

**DISEÑO E IMPLEMENTACIÓN DEL PROTOTIPO DE UN OSCILOSCOPIO
DIGITAL BASADO EN EL SISTEMA EMBEBIDO MINI2440**

**CESAR DAVID REVELO APRAEZ
JOSE LUIS ORDOÑEZ REVELO**

**UNIVERSIDAD DE NARIÑO
FACULTAD DE INGENIERÍA
INGENIERÍA ELECTRÓNICA
SAN JUAN DE PASTO
2012**

**DISEÑO E IMPLEMENTACIÓN DEL PROTOTIPO DE UN OSCILOSCOPIO
DIGITAL BASADO EN EL SISTEMA EMBEBIDO MINI2440**

**CESAR DAVID REVELO APRAEZ
JOSE LUIS ORDOÑEZ REVELO**

Trabajo de grado para optar al título de Ingeniero Electrónico

**DIRECTOR
DARÍO FERNANDO FAJARDO FAJARDO
INGENIERO ELECTRÓNICO**

**ASESOR
MARIO FERNANDO JOJOA ACOSTA
INGENIERO ELECTRÓNICO**

**UNIVERSIDAD DE NARIÑO
FACULTAD DE INGENIERÍA
INGENIERÍA ELECTRÓNICA
SAN JUAN DE PASTO
2012**

Nota de Responsabilidad

“La Universidad de Nariño no se hace responsable por las opiniones o resultados obtenidos en el presente trabajo y para su publicación priman las normas sobre el derecho de autor”.

Acuerdo 1. Artículo 324. Octubre 11 de 1966. Emanado del honorable Consejo Directivo de la Universidad de Nariño.

Nota de aceptación

Firma del presidente del Jurado

Firma del jurado

Firma del jurado

A Dios, por habernos permitido culminar con éxito la realización de este proyecto, y a nuestros padres por brindarnos la oportunidad de hacer realidad este sueño compartido, gracias por su incondicional apoyo y formidable ejemplo.

AGRADECIMIENTOS

A todas las personas que participaron e hicieron posible la realización de este proyecto, muchas gracias por su apoyo y enseñanza:

Darío Fernando Fajardo Fajardo, Director
Mario Fernando Jojoa Acosta, Asesor
Andrey Fernando Salas
Jaime Andrés Rodríguez
Jimmy Velásquez

A quienes participaron en la fase de investigación y experimento:

Álvaro José Cervelión C.
Jhon Alejandro Botina
Jhon Alexander Yela

Al Departamento de Ingeniería Electrónica.
Por apoyar nuestra idea y permitirnos la realización de este proyecto

TABLA DE CONTENIDO

	Pág.
INTRODUCCIÓN.....	19
JUSTIFICACIÓN.....	20
1. OBJETIVOS	21
1.1 OBJETIVO GENERAL	21
1.2 OBJETIVOS ESPECIFICOS.....	21
2. MARCO TEÓRICO.....	23
2.1 ANTECEDENTES	23
2.2 SISTEMAS EMBEDIDOS	24
2.2.1 ¿Qué es un sistema embebido?	25
2.2.2 Algunos ejemplos.....	26
2.2.3 Características de los sistemas embebidos.....	27
2.2.3.1 Arquitectura de los sistemas embebidos	28
2.2.3.2 Software en los sistemas embebidos.....	33
2.2.4 Historia y situación actual de los sistemas embebidos.....	37
2.3 PLACA DE DESARROLLO MINI2440	38
2.3.1 Arquitectura interna de la MINI2440	40
2.3.1.1 Microprocesador ARM S3C2440AL.....	42
2.3.1.2 Otros subsistemas importantes en la MINI2440	48
2.3.2 Software en la MINI2440.....	57
2.3.2.1 Sistemas operativos	57
2.3.2.2 Bootloaders o gestores de arranque.....	58
2.4 EL OSCILOSCOPIO	59
2.4.1 ¿Qué es un Osciloscopio?	59
2.4.2 ¿Qué muestra un Osciloscopio?.....	60
2.4.3 Funcionamiento del Osciloscopio	61
2.4.3.1 Osciloscopio Análogo.....	61
2.4.3.2 Osciloscopio Digital	62
2.4.4 Tipos de Osciloscopio Digital	63
2.4.4.1 Osciloscopio de memoria digital	63
2.4.4.2 Osciloscopio de fósforo digital	64
2.4.4.3 Osciloscopio de muestreo.....	65
2.4.5 Factores que influyen en la calidad de un Osciloscopio	66

2.4.5.1	Ancho de banda	66
2.4.5.2	Precisión de la ganancia	67
2.4.5.3	Relación señal a ruido (SNR).....	67
2.4.5.4	Errores de cuantizacion.....	67
3.	DISEÑO Y DESARROLLO DEL DISPOSITIVO	72
3.1	HARDWARE	73
3.1.1	Selección de la arquitectura.....	73
3.1.2	Funcionamiento y diagrama de bloques	73
3.1.3	El Microcontrolador PIC18F2553	74
3.1.3.1	Módulo ADC del Microcontrolador	77
3.1.3.2	Módulo de comunicación serial EUSART	86
3.1.4	Etapa de entrada y acondicionamiento de la señal	94
3.1.4.1	Sección vertical	95
3.1.4.2	Bloque comparador de tensión	103
3.1.4.3	Bloque sumador DC	108
3.1.5	Esquema final de la placa de adquisición de datos implementada	110
3.1.6	Fuente de alimentación	112
3.2	FIRMWARE.....	112
3.2.1	Herramienta de trabajo	113
3.2.2	Algoritmo principal del Microcontrolador	114
3.2.3	Algoritmo para la ejecución de ordenes del sistema embebido MINI2440.....	115
3.2.4	Algoritmo para el control del disparo y conversión de la señal de entrada	118
3.2.5	Algoritmo para el muestreo de señales de baja frecuencia.....	120
3.3	SOFTWARE	121
3.3.1	Selección de herramientas.....	121
3.3.1.1	Lenguaje de programación.....	121
3.3.1.2	La plataforma de desarrollo .NET Framework	122
3.3.1.3	Entorno de desarrollo	124
3.3.1.4	Visual Studio 2005 y .NET Compact Framework.....	126
3.3.2	Estructura del software	127
3.3.2.1	Biblioteca de clases de .NET Compact Framework utilizadas	127
3.3.2.2	Clases y métodos utilizados en el desarrollo de la aplicación	129
3.3.3	Funcionamiento del software	130
3.3.3.1	Control de la interfaz gráfica	130
3.4	CONSTRUCCION DEL DISPOSITIVO.....	135
3.4.1	Materiales y componentes utilizados	135
3.4.2	Diseño de las placas PCB	135
3.4.3	Diseño y construcción de la carcasa.....	139

3.4.4	Presupuesto total del equipo.....	141
4.	RESULTADOS	143
4.1	CARACTERIZACIÓN DEL EQUIPO.....	143
4.1.1	Características eléctricas	143
4.1.2	Linealidad	143
4.1.3	Ruido	144
4.2	PRUEBA DE FUNCIONAMIENTO DEL EQUIPO	145
5.	CONCLUSIONES.....	149
	BIBLIOGRAFIA.....	150

LISTA DE FIGURAS

	Pág.
Figura 1. Esquema sencillo de una PC o de un sistema embebido.	29
Figura 2. Diagrama de bloques de una computadora genérica.	30
Figura 3. Diagrama de bloques de un sistema embebido arbitrario.	30
Figura 4. Componentes de un router/modem ADSL Natgear DG632.	33
Figura 5. Arquitectura del sistema operativo OSfree.	34
Figura 6. Kit MINI2440 de FriendlyARM.	40
Figura 7. Configuración de los componentes hardware en la placa de desarrollo MINI2440.	42
Figura 8. Esquema de los registros del microprocesador S3C2440AL.	46
Figura 9. Esquema del registro CPSR/SPSR del S3C2440AL.	46
Figura 10. Esquema de las 2 memorias Flash incorporadas en la MINI2440.	49
Figura 11. Sistema de almacenamiento SDRAM en la MINI2440.	50
Figura 12. Disposición de la memoria dependiendo del modo de arranque.	50
Figura 13. Esquema interno de las interfaces USB con las cuales cuenta la MINI2440.	52
Figura 14. Aspecto físico de los 3 puertos seriales (UART0, UART1, UART2), incorporados en la MINI2440.	52
Figura 15. Esquema de chip LCD41P, para la interfaz con el display LCD.	53
Figura 16. Conexión física del pin AIN0 al potenciómetro de 10K para testear el ADC con el que cuenta la MINI2440.	54
Figura 17. Chip DM9000 10/100M LAN con transformadores de red y un conector RJ-45.	55
Figura 18. Interfaz JTAG incorporada en la MINI2440.	56
Figura 19. Aspecto físico del CON8, presente en la placa MINI2440.	56
Figura 20. Reguladores de voltaje incorporado en la MINI2440, encargado de suministrar tensiones de 3.3v, 1.8v y 1.25v.	57
Figura 21. Conexión del botón RESET al chip MAX811.	57
Figura 22. Onda senoidal.	60
Figura 23. Onda triangular.	60
Figura 24. Onda cuadrada o rectangular.	61
Figura 25. Diagrama de bloques Osciloscopio análogo.	61
Figura 26. Diagrama de bloques Osciloscopio digital.	62
Figura 27. Bloques osciloscopio de memoria digital.	64
Figura 28. Bloques osciloscopio de fósforo digital.	65
Figura 29. Bloques osciloscopio de muestreo.	66
Figura 30. Error de compensación.	68
Figura 31. Error de ganancia.	69
Figura 32. Error de no-linealidad diferencial.	70
Figura 33. Error de cuantización.	71
Figura 34. Diagrama de bloques del Osciloscopio Digital implementado.	74
Figura 35. Pin-out del microcontrolador PIC18F2553.	76
Figura 36. Multiplexación de canales análogo-digital del PIC18F2553.	80
Figura 37. Función de transferencia del conversor análogo-digital del PIC18F2553.	81
Figura 38. Modelo del conversor análogo-digital del PIC18F2553.	82
Figura 39. Operación del conversor A/D (ACQT <2:0>=001, TACQ = 2TAD).	86
Figura 40. Diagrama de bloques del transmisor de la EUSART.	89
Figura 41. Diagrama de bloques del receptor de la EUSART.	91
Figura 42. Selector de acoplamiento de entrada (AC-GND-DC).	96
Figura 43. Esquema del atenuador de entrada implementado (Divisor de Voltaje).	97
Figura 44. Circuito equivalente de una sonda compensada.	99

Figura 45. Esquema del amplificador inversor de ganancia programable implementado.....	100
Figura 46. Esquema del A.O. configurado como comparador de tensión.....	104
Figura 47. Método de muestreo en tiempo real.....	105
Figura 48. Interpolación lineal y seno x/x	106
Figura 49. Presentación de una señal muestreada en tiempo equivalente aleatorio y secuencial respectivamente.....	107
Figura 50. Formas de onda correspondiente al muestreo en tiempo equivalente secuencial.	108
Figura 51. Esquema del bloque sumador DC, incorporado previo a la entrada del ADC interno en el microcontrolador.....	109
Figura 52. Esquema final de la placa de adquisición de datos incorporada en el dispositivo.....	111
Figura 53. Diagrama de conexiones de la fuente de alimentación eléctrica.....	112
Figura 54. Captura de Pantalla de MPLAB V.8.70.....	113
Figura 55. Algoritmo principal del Microcontrolador.....	114
Figura 56. Algoritmo para la ejecución de órdenes del sistema embebido MINI2440.....	116
Figura 57. Algoritmo para el control de disparo de la conversión de la señal de entrada.....	118
Figura 58. Algoritmo para el muestreo de señales de baja frecuencia.....	120
Figura 59. Arquitectura de la infraestructura de desarrollo.....	123
Figura 60. Organización de un proyecto en Visual Studio 2005.....	125
Figura 61. Organización de una solución en Visual Studio 2005.....	126
Figura 62. Compilación de un proyecto.....	126
Figura 63. GUI desplegada en la pantalla del sistema embebido MINI2440.....	131
Figura 64. Algoritmo principal de la aplicación ejecutada por el sistema embebido MINI2440.....	132
Figura 65. Algoritmo del subproceso ejecutado por la aplicación gráfica implementada en el sistema embebido MINI2440.....	134
Figura 66. Método de planchado.....	136
Figura 67. Placa adherida al papel satinado e introducida en un recipiente con agua.....	136
Figura 68. Placa después de retirar el papel.....	137
Figura 69. Placa de la tarjeta de adquisición de datos después de retirar el cobre.....	137
Figura 70. Placa fuente de alimentación después de retirar el cobre.....	138
Figura 71. Placa de la tarjeta de adquisición de datos soldada.....	138
Figura 72. Placa de la fuente de alimentación soldada.....	139
Figura 73. Vista interna de la carcasa del dispositivo.....	140
Figura 74. Vista frontal de la carcasa del dispositivo.....	140
Figura 75. Vista posterior de la carcasa del dispositivo.....	141
Figura 76. Señal triangular (comprobación de linealidad en el dispositivo).....	144
Figura 77. Señal de ruido interno del dispositivo.....	144
Figura 78. Señal sinusoidal visualizada en los 2 equipos.....	145
Figura 79. Señal triangular visualizada en los 2 equipos.....	146
Figura 80. Señal cuadrada o rectangular visualizada en los 2 equipos.....	146
Figura 81. Señal sinusoidal, de baja frecuencia (101Hz) y con Amplitud de 12Vpp.....	147
Figura 82. Señal sinusoidal, de alta frecuencia (212.2 KHz) y con Amplitud de 3Vpp.....	148

LISTA DE TABLAS

	Pág.
Tabla 1. Sistemas operativos embebidos vigentes hasta el año 2009.....	35
Tabla 2. Modos de ejecución	44
Tabla 3. Tipos y Modos de excepción	48
Tabla 4. Características Microcontrolador PIC18F2553.....	75
Tabla 5. Registro ADCON0 (FC2h): Registro 0 DE Control A/D	77
Tabla 6. CHS3:CHS0: Bits de selección del canal análogo.	77
Tabla 7. GO/DONE: Bit de inicio y monitorización del estado de la Conversión A/D.....	78
Tabla 8. ADON: Bit de habilitación del Conversor A/D.....	78
Tabla 9. Registro ADCON1 (FC1h): Registro 1 de Control A/D.	78
Tabla 10. VCFG1: Bit de configuración de la tensión de referencia VREF-.....	78
Tabla 11. VCFG0: Bit de configuración de la tensión de referencia VREF+.....	78
Tabla 12. PCFG3:PCFG0: Bits de configuración de los puertos de conversión A/D.	78
Tabla 13. Registro ADCON2 (FC0h): Registro 2 de control A/D.	79
Tabla 14. ADFM: Bit de justificación del resultado de la conversión.	79
Tabla 15. ACQT2:ACQT0: Bits de configuración del tiempo de adquisición para la Conversión.	79
Tabla 16. ADCS2:ADCS0: Bits de configuración de la señal de reloj para el Conversor A/D.	79
Tabla 17. TAD vs Frecuencia de operación del dispositivo.....	85
Tabla 18. Registro TXSTA (FACH): Registro de estado y control de transmisión.....	88
Tabla 19. CSRC: Bit de selección de la fuente de señal de reloj para el modo síncrono.	88
Tabla 20. TX9: Bit de habilitación del modo de 9 bits en transmisión.	88
Tabla 21. TXEN: Bit de habilitación de la transmisión.....	88
Tabla 22. SYNC: Bit de selección del modo de trabajo de la EUSART.	88
Tabla 23. SENDB: Bit de envío del carácter de BREAK en modo asíncrono.....	88
Tabla 24. BRGH: Bit de selección del modo de alta velocidad en modo asíncrono.	88
Tabla 25. TRMT: Bit de estado del registro de desplazamiento de transmisión.	88
Tabla 26. Registro RCSTA (FABh): Registro de estado y control de recepción.	90
Tabla 27. SPEN: Bit de activación del puerto serie.	90
Tabla 28. RX9: Bit de habilitación del modo de 9 bits en transmisión.....	90
Tabla 29. SREN: Bit de habilitación de la recepción simple en modo síncrono Maestro.....	90
Tabla 30. CREN: Bit de habilitación de la recepción.	90
Tabla 31. ADDEN: Bit de habilitación de la dirección para el modo asíncrono de 9 bits.	90
Tabla 32. FERR: Bit de error de formato (solo lectura).	90
Tabla 33. OERR: Bit de error de sobrescritura (solo lectura).....	91
Tabla 34. Registro BAUDCON (FB8h): Registro de estado y control de los baudios.	92
Tabla 35. ABDOVF: Bit de desbordamiento de auto-detección de la velocidad de comunicación.	92
Tabla 36. RCIDL: Bit de estado de la operación de recepción.....	92
Tabla 37. SCKP: Bit de selección de polaridad de la señal de reloj en modo síncrono.....	92
Tabla 38. BRG16: Bit de habilitación del generador de velocidad de 16 bits (SPBRG y SPBRGH).92	92
Tabla 39. WUE: Bit de habilitación del modo de auto-activación en modo asíncrono.	93
Tabla 40. ABDEN: Bit de habilitación del modo de auto-detección de velocidad de comunicación.	93
Tabla 41. Calculo de velocidades de transmisión.	93
Tabla 42. Parámetros de trabajo del multiplexor CD4051BC.....	100
Tabla 43. Parámetros para el cálculo de las ganancias en el PGA.....	102
Tabla 44. Ganacias calculadas para cada factor de escala	103
Tabla 45. Resistencia de realimntación calculada para cada factor de escala seleccionado.	103
Tabla 46. Presupuesto total del dispositivo	141
Tabla 47. Características electricas del Osciloscopio.....	143

LISTA DE ANEXOS

ANEXO A: Manual de usuario para manejo del prototipo del Osciloscopio Digital.

ANEXO B: CD-ROM con los siguientes archivos:

- Manual de usuario para la construcción del Entorno de Desarrollo Integrado (IDE), necesario para trabajar con el sistema embebido MINI2440.
- Software de la interfaz gráfica de usuario (GUI) desarrollada bajo Microsoft Visual Studio 2005.
- Carpeta con los programas (código fuente) de control del PIC y visualización en la MINI2440 (software) desarrollados.
- Planos de la estructura externa y circuitos impresos diseñados.
- Fotografías y videos de la construcción del prototipo.

GLOSARIO

ALIASING: Corresponde al efecto que causa que señales continuas distintas se tornen indistinguibles cuando se muestrean digitalmente.

ARM: (Advanced RISC Machines). Arquitectura RISC de 32 bits desarrollada por ARM Holdings, la relativa simplicidad de los procesadores ARM los hace ideales para aplicaciones de baja potencia.

BAUDIO: Unidad de medida, utilizada en telecomunicaciones, que representa la cantidad de veces que cambia el estado de una señal en un periodo de tiempo, tanto para señales digitales como para señales analógicas.

CARCASA: Conjunto de piezas duras y resistentes, que dan soporte (internas) o protegen (externas) a otras partes de un equipo.

CODIFICACIÓN: Proceso que consiste en la traducción de los valores de tensión eléctrica analógicos que ya han sido previamente cuantificados (ponderados) al sistema binario, mediante códigos preestablecidos.

CUANTIFICACIÓN: Etapa del proceso de digitalización de señales, que consiste en convertir una sucesión de muestras de amplitud continua en una sucesión de valores discretos preestablecidos según el código utilizado.

DIGITALIZACIÓN: Consiste básicamente en realizar de forma periódica medidas de la amplitud (tensión) de una señal, redondear sus valores a un conjunto finito de niveles preestablecidos de tensión y registrarlos como números enteros en cualquier tipo de memoria o soporte.

DRAM: Memoria dinámica de acceso aleatorio que se usa principalmente en los módulos de memoria RAM como memoria principal del sistema.

FIRMWARE: Bloque de instrucciones de máquina para propios específicos, grabado en una memoria de tipo de solo lectura (ROM, EEPROM, flash, etc), que establece la lógica de más bajo nivel que controla los circuitos electrónicos de un dispositivo de cualquier tipo.

GPIO: (General Purpose Input/Output), puertos de expansión de Entrada y Salida, presentes en cualquier tipo de Hardware.

GUI: Programa informático que actúa de interfaz de usuario, utilizando un conjunto de imágenes y objetos gráficos para representar la información y acciones disponibles en la interfaz.

MUESTREO: Etapa del proceso de digitalización de señales, que consiste en tomar muestras de una señal analógica a una frecuencia o tasa de muestreo constante, para cuantificarlas posteriormente.

PAGER: (También llamado dispositivo de radio búsqueda o radio mensajería), es un dispositivo de telecomunicaciones muy simple utilizado para la recepción de mensajes de texto cortos.

RISC: (Reduced Instruction Set Computer). Conjunto de Instrucciones Reducido. Corresponde a un tipo de arquitectura de computadora que promueve conjuntos pequeños y simples de instrucciones, los cuales toman poco tiempo para ejecutarse.

SDRAM: Memoria de acceso aleatorio DRAM que tiene una interfaz síncrona.

RESUMEN

El presente proyecto consiste en el desarrollo del prototipo de un dispositivo digital de captura de señales eléctricas (conocido con el nombre de osciloscopio), implementado sobre el sistema embebido MINI2440.

El osciloscopio posee 2 canales de 12 bit de resolución y un ancho de banda de 250KHz, su diseño está basado en una placa que adquiere y transmite de manera serial los datos hacia el sistema embebido MINI2440 a través del protocolo RS-232, para su posterior procesamiento y visualización sobre la pantalla táctil con la cual cuenta el dispositivo.

El diseño de la placa de adquisición de datos está basado en un microcontrolador central de referencia PIC18F2553 y varios componentes (controlados por dicho microcontrolador) con el fin de llevar a cabo la tarea de captura de la señal de entrada.

Además del diseño y construcción del hardware (placa de adquisición de datos), el proyecto contempla el desarrollo del firmware (programa que corre en el PIC para controlar los componentes) y el software (interfaz gráfica que corre sobre el sistema embebido MINI2440 para controlar el osciloscopio).

ABSTRACT

This project involves the development of the prototype of an electrical signal acquisition device (also known as oscilloscope) implemented on the MINI2440 Embedded System.

The oscilloscope has two 12-bits resolution channels and a bandwidth of 250 KHz, it's design is based on a board that acquires and transmits the data serially to the MINI2440 Embedded System through the RS-232 protocol, for subsequent processing and visualization on the touchscreen that has the device.

The design of the data acquisition board is based on a central microcontroller PIC18F2553 core and multiple components (controlled by such processor) which take of the whole acquisition task.

Besides design and building of the hardware (acquisition data board), this project includes the development of firmware (program which runs on the microcontroller) and software (graphical interface that runs on the MINI2440 Embedded System to control the oscilloscope).

MARCAS REGISTRADAS

Windows Embedded CE 6.0 y sus respectivas actualizaciones R2 y R3, al igual que Windows Mobile, Windows Server, Windows Vista, SQL Server, Visual Basic, Visual C#, Visual Studio, Windows XP y Win7, son marcas registradas de Microsoft Corporation® en Estados Unidos y en otros países.

MPLAB IDE, es una marca registrada de MICROCHIP®.

PROTEUS, es una marca registrada de Proteus–Hubor®.

AutoCAD, es una marca registrada de Autodesk®.

MINI2440, es una marca registrada de FriendlyARM®.

PIC18F2553, es una marca registrada por MICROCHIP®.

CD4051BC Single 8-Channel Analog Mux, es una marca registrada de National Semiconductor®.

A.O. LM311, A.O. TL082, son una marca registrada de STMicroelectronics®

INTRODUCCIÓN

Los niveles de integración actuales permiten construir sistemas cada vez más pequeños, veloces y de menor consumo de potencia, lo cual ha favorecido su difusión en casi todas las actividades humanas, a medida que se extiende el campo de aplicación de estos sistemas digitales, lo hacen las exigencias de funcionamiento a ellos impuestas; nuevos retos de diseño se presentan a medida que los sistemas embebidos se integran a las actividades humanas.

Un sistema embebido consiste en un sistema de computación cuyo hardware y software están específicamente diseñados y optimizados para resolver un problema concreto eficientemente, además pueden hacer parte de un dispositivo o máquina mayor, que cumple funciones de monitoreo o control. Algunos sistemas embebidos incluyen un sistema operativo, pero muchos son tan especializados que toda la lógica puede implementarse en un solo programa.

La característica principal que diferencia a los “embebidos” de los demás sistemas electrónicos es que al estar insertados dentro del dispositivo que controlan, están sujetos en mayor medida a cumplir requisitos de tamaño, fiabilidad, consumo energético y costo, y su existencia puede no ser aparente. Algunos ejemplos de Sistemas embebidos son los sistemas de información integrados en automóviles, trenes o aviones, y controladores de procesos en sistemas de producción industrial.

La gran aplicabilidad de los Sistemas embebidos en cualquier ámbito sectorial, así como el valor añadido que aportan los mismos a los equipos que los contienen, hacen que el desarrollo de aplicaciones que den solución a problemas de ingeniería llevadas a cabo sobre estos sistemas se convierta en una alternativa innovadora a nivel institucional, que fácilmente podría abrir una nueva línea de investigación enfocada al desarrollo de aplicaciones en este tipo de sistemas.

JUSTIFICACIÓN

Fundaciones para el desarrollo tecnológico¹, al igual que Instituciones Universitarias a través de grupos de investigación a nivel de Iberoamérica², llevan actualmente a cabo importantes proyectos enfocados en el desarrollo de Sistemas embebidos, generando de esta manera una apertura en cuanto a la transferencia tecnológica y de conocimiento, no solo en el desarrollo de estos sistemas, sino también en las aplicaciones que es posible implementar sobre estos, permitiendo así el diseño y construcción de equipos que cuenten con un valor agregado en cuanto al tamaño, costo, consumo de potencia y velocidad de procesamiento.

Sumado a lo anterior, la disminución dramática en el costo de los dispositivos semiconductores (generalmente llamadas placas de desarrollo) necesarios para implementar cualquier tipo de aplicación, son factores de peso que constituyen una razón fundamental por la cual se decide llevar a cabo el desarrollo de una aplicación sobre el sistema embebido MINI2440 en particular.

El proceso de investigación en el sistema embebido MINI2440, está enfocado principalmente a ofrecer un conjunto de conocimientos en cuanto al manejo de la placa de desarrollo, con el fin de poder implementar cualquier tipo de aplicación orientada a resolver un problema específico de ingeniería, y de esta manera contar con un equipo liviano, de menor tamaño, bajo consumo de potencia y una alta velocidad de procesamiento en cuanto a la ejecución de tareas.

Se pretende que los estudiantes del programa de Ingeniería Electrónica de la Universidad de Nariño cuenten con una alternativa innovadora en el desarrollo de sus proyectos, basadas particularmente en el sistema embebido MINI2440, aprovechando así el gran aporte que actualmente ofrece el acelerado desarrollo de los sistemas embebido.

¹ Fundación OPTI. Fundación ASCAMM. Tendencias y aplicaciones de los sistemas embebidos Septiembre 2009 [En línea].

<<http://www.opti.org/publicaciones/pdf/texto131.pdf>>

² Camargo B, Carlos Iván. Transferencia tecnológica y de comunicaciones en el diseño de sistemas embebidos.

Agosto 13 2011 [En línea].

<http://www.bdigital.unal.edu.com/5696/1/carlosivancamargobare%C3%B1o_2011.pdf>

1. OBJETIVOS

1.1 OBJETIVO GENERAL

Desarrollar un dispositivo de medición electrónico que permita la visualización y el control en los parámetros de medida de señales eléctricas en tiempo real, a través de una aplicación gráfica implementada sobre el sistema embebido MINI2440.

1.2 OBJETIVOS ESPECIFICOS

- Adquirir conocimientos básicos en cuanto al funcionamiento y la arquitectura interna de los sistemas embebidos en general, así como también al papel que actualmente juegan estos sistemas de computación en el desarrollo de dispositivos electrónicos de propósito específico.
- Apropiarse del conocimiento suficiente acerca del funcionamiento, la arquitectura interna y los recursos de hardware y software con los cuales cuenta el sistema embebido MINI2440.
- Llevar a cabo la construcción de un entorno apropiado para el desarrollo (edición, compilación y depuración del código fuente) de aplicaciones basadas en la plataforma .NET de Microsoft, las cuales posteriormente serán ejecutadas sobre el sistema embebido MINI2440.
- Implementar un Sistema Operativo dedicado, basado en Microsoft Windows, sobre el sistema embebido MINI2440, que permita ejecutar aplicaciones basadas en el .NET Compact Framework de la plataforma .NET de Microsoft.
- Utilizar un lenguaje de alto nivel (HLL), en el desarrollo de una aplicación gráfica para el sistema embebido MINI2440, la cual despliegue en la pantalla del dispositivo una amigable GUI (Interfaz Gráfica de Usuario), para el control de los diferentes parámetros de medida del dispositivo y la visualización de las señales eléctricas de entrada.
- Diseñar e implementar una etapa previa al sistema embebido MINI2440, que permita la adquisición, adaptación y conversión digital de las señales análogas de entrada.

- Verificar el correcto funcionamiento del prototipo final, utilizando un dispositivo electrónico de laboratorio, denominado generador de formas de onda.

2. MARCO TEÓRICO

2.1 ANTECEDENTES

En el documento final del trabajo de grado “Implementación de un sistema teleoperado con realimentación visual para evasión de obstáculos de un robot móvil”³, presentado en el año 2012 por Luis Miguel Betancourt y Nelson Alfonso Boton Gómez, en la Universidad Militar Nueva Granada. Se implementa un sistema teleoperado con realimentación visual, utilizando el sistema embebido MINI2440 montado sobre el robot móvil con el propósito de controlarlo desde un computador distante, que se comunica con el sistema (MINI2440-Robot Móvil) por medio del protocolo TCP. El usuario recibe la imagen enviada por el sistema y será capaz de controlar el movimiento del robot al enviar señales de control por medio del protocolo TCP.

En el año 2011, en la Universidad Tecnológica de Malasya, fue llevado a cabo el proyecto de investigación y desarrollo, “Development of a Portable Heart Beat Monitor Unit Using Friendly ARM MINI2440”⁴, desarrollado por Mat Taib Binti Norshazwana, en el cual se diseña y construye una unidad portátil para monitorear el latido del corazón usando la Friendly ARM MINI2440, básicamente consiste en un electrocardiograma personal, implementado sobre el sistema embebido MINI2440, capaz de monitorear el ritmo cardiaco.

En el año 2011, fue presentado el informe final de la tesis “Desarrollo de una interfaz gráfica con pantalla táctil para el manejo de un sistema robótico”⁵, desarrollado por Javier Echeverría Senosiain, en este proyecto se logra sustituir un ordenador de sobremesa que controla un sistema robótico por un equipo de menor tamaño, portátil y manejable.

³ Betancourt, Luis Miguel. Boton G, Nelson Alfonso. Implementación de un sistema teleoperado con realimentación visual para evasión de obstáculos de un robot móvil. Junio 12 2012 [En línea].

< <http://repository.unimilitar.edu.co/handle/10654/3299> >

⁴ Norshazwana B, Mat Taib. Development of a Portable Heart Beat Monitor Unit Using Friendly ARM MINI2440. 2011[En línea].

< <http://norlaili.fke.utm.my/human-capital-1> >

⁵ Echeverría Senosiain, Javier. Desarrollo de una interfaz gráfica con pantalla táctil para el manejo de un sistema robótico. 2011[En línea].

< <http://academica-e.unavarra.es/handle/2454/4623> >

En el documento “Instrucción en el diseño de sistemas empotrados. Aplicación al control de un brazo robot”⁶ escrito por Houcine Hassan, Javier Soriano, Javier Montagudy Carlos Domínguez. En el año 2003, en la Universidad Politécnica de Valencia (UPV), se presenta una metodología de aprendizaje para el diseño de sistemas empotrados (SE), inspirada en el proyecto Europa de la Universidad Politécnica de Valencia, en el artículo se describe la implementación de dicha metodología en la asignatura de Diseño de Sistemas con Microprocesadores de la Escuela Técnica Superior de Informática Aplicada.

En el documento final del trabajo de grado “Representación de variables eléctricas en el sistema embebido MINI2440”⁷, presentado en el año 2011 por Fernando Marcotegui Zabalza, en la Escuela Técnica Superior de Ingenieros Industriales y de Telecomunicaciones de Navarra. Se describe el proceso, que fue llevado a cabo para el monitoreo de una señal análoga en la entrada del módulo ADC del sistema embebido MINI2440.

En el trabajo de grado presentado en el año 2006, denominado “Osciloscopio USB”⁸, desarrollado por Pablo Hoffman y Martin Szmulewicz, en la Universidad ORT de Uruguay, el cual consiste en el desarrollo del prototipo de un osciloscopio digital de captura de señales eléctricas (también conocido con el nombre de osciloscopio) con conexión a la PC a través del puerto USB.

2.2 SISTEMAS EMBEBIDOS

Los sistemas de computación se encuentran en cualquier parte. Es probable que nadie se sorprenda al saber que millones de sistemas de computación son producidos y vendidos cada año, como por ejemplo computadores de escritorio o computadores personales. Sin embargo, sorprende saber que billones de sistemas de computación de propósito específico son construidos y vendidos cada año; los sistemas embebidos se encuentran alrededor de nuestras vidas, en forma de teléfonos móviles, equipos médicos, sistemas de navegación aérea, reproductores MP3, impresoras, automóviles, etc.

⁶ Houcine Hassan, Javier Doriano. Montagudy Carlos Domínguez. Instrucción en el diseño de un sistema empotrado. Aplicación al control de un brazo robot. 2003[En línea].

< <http://www.ceautomatica.es/old/actividades/jornadas/XXIV/documentos/econ/90.pdf> >

⁷ Marcotegui Zabalza, Fernando. Representación de variables eléctricas en el sistema embebido Mini2440. 2011[En línea].

< <http://academica-e.unavarra.es/bitstream/handle/2454/4283/577623.pdf?sequence=1> >

⁸ Hoffman, Pablo. Szmulewicz, Martin. Osciloscopio USB.

Septiembre 13 2006[En línea].

< <http://pablohoffman.com/cgi-bin/twiki/bin/view/Oscusb/WebHome> >

Cada vez que se mira alrededor es posible identificar un dispositivo que contiene un microprocesador, y probablemente se ha encontrado un sistema embebido. Por estas razones resulta conveniente y pertinente estudiarlos.

2.2.1 ¿Qué es un sistema embebido?

Dar una descripción precisa de qué es un sistema embebido puede resultar una tarea complicada. El experto en software embebido Michael Barr definía, no sin cierta ambigüedad, un sistema embebido o como: “Una combinación de hardware y software de computadora, sumado tal vez a algunas piezas mecánicas o de otro tipo, diseñada para tener una función específica”⁹

Esta definición se centra en la idea de que los sistemas embebidos en principio son diseñados específicamente para llevar a cabo de manera eficiente una tarea en particular, de tal forma que los recursos utilizados son optimizados para resolver dicho problema en concreto. Un sistema embebido posee hardware de computador junto con software embebido como uno de sus componentes más importantes, por lo tanto debe existir un sistema de computación gobernado por un microprocesador o similares.

Así, un sistema embebido se encontraría en un punto intermedio entre una computadora de propósito general -como por ejemplo una computadora personal (PC), que está diseñada para permitir realizar, idealmente, la mayor variedad posible de tareas- y un sistema que es diseñado por hardware específicamente para una tarea o un conjunto reducido de ellas, como por ejemplo los sistemas tradicionales de control, los dispositivos de lógica programable (PLDs) o las *field-programmable gate arrays* (FPGAs), que son programados directamente en hardware.

No obstante, en los últimos años han cobrado importancia algunos tipos de dispositivos que, si bien no encajan estrictamente en la definición anterior, comparten ciertas características con el resto de sistemas embebidos y por tanto resulta razonable incluirlos en este grupo. En concreto, los *smartphones* y las computadoras *tablets*, así como las *personal digital assistants* (PDAs), son dispositivos que podríamos situar entre las PCs y los sistemas embebidos, pues aunque no están diseñados para realizar una única función específica, presentan muchas características típicas de estos últimos, como las restricciones en los recursos computacionales disponibles o la manera de desarrollar el software.

⁹ Barr, Michael. Programming Embedded System in C and C++. En: O'Reilly [En línea] N° 1, Enero, 1999.

<http://www.embeonline.com/uploads/O'Reilly-programming_embedded_systems_in_C_and_C++.pdf>

Ante esta cuestión, el citado Michael Barr, desde su perspectiva de desarrollador de software, en la actualidad es partidario de incluir estos últimos dispositivos mencionados en el conjunto de los sistemas embebidos. Siguiendo el mismo razonamiento, en este trabajo se considera que la placa de desarrollo MINI2440 de FriendlyARM, con la cual se llevó a cabo este proyecto, es también un sistema embebido a todos los efectos, ya que su arquitectura es muy similar a una Tablet o una PDA.

2.2.2 Algunos ejemplos

Es posible encontrar infinidad de ejemplos de dispositivos que contienen sistemas embebidos y que son empleados en la actualidad a diario en el llamado “primer mundo”, en la esfera doméstica, profesional, industrial, científica, militar, etc. Además de los ya mencionados smartphones, tablets y PDAs, a continuación se citan algunos otros:

Relojes digitales: En cualquier dispositivo donde haya un reloj digital existe un sistema embebido, tanto en un reloj de muñeca o en un horno o termómetro digital, por ejemplo. Típicamente, un reloj digital contiene un procesador sencillo de 4 u 8 bits (empleado típicamente para permitir dar soporte con el mismo hardware a una variedad de modelos y características) y su propia ROM integrada.

Electrodomésticos: En prácticamente cualquier electrodoméstico fabricado en las últimas décadas se encuentran sistemas embebidos, desde una batidora a una lavadora o a una tostadora. Cada tipo de electrodoméstico dispondrá de un sistema específico que le permita realizar su función e interactuar con el usuario final.

Automóviles: El control eléctrico de los automóviles fabricados hoy en día es cada vez más complejo y precisa de numerosos dispositivos embebidos, para los sistemas de ignición, transmisión, frenado, suspensión, control de tracción, dirección asistida, seguridad, localización geográfica o control de emisiones. Un automóvil puede tener hasta un centenar de microprocesadores y microcontroladores, muchos de ellos comunicados entre sí.

Máquinas de producción industrial: En los procesos industriales, el control de motores, hornos, maquinaria, etc. Es manejado por sistemas embebidos, los cuales a menudo ofrecen una interfaz hombre-máquina para ser dirigidos por un operario e informarle al mismo de la marcha del proceso.

Videoconsolas: En muchos casos, este tipo de máquinas tienen más potencia de procesamiento que los PCs de su generación y su procesador está altamente

especializado para las demandas del tipo de videojuegos que va a permitir jugar. En ocasiones el diseño de cada procesador corre a cargo del propio fabricante de la videoconsola.

Sistemas radar de aviones: Los dispositivos para el procesado de la señal recibida o reflejada de los subsistemas radar embarcados en aviones requieren alta potencia de cálculo además de ocupar poco espacio, pesar poco y soportar condiciones extremas de funcionamiento (temperatura, presión atmosférica, vibraciones, etc.).

Así, se podría enumerar un sinfín más de ejemplos: teléfonos móviles, reproductores de audio o video, periféricos de una PC, navegadores GPS, controles remotos, equipos médicos, cajeros automáticos, máquinas expendedoras, etc.

2.2.3 Características de los sistemas embebidos

Los sistemas embebidos poseen ciertas características que los distinguen de otros sistemas de cómputo, a continuación se describen las más importantes:

Funcionamiento específico: Un sistema embebido usualmente ejecuta un programa específico de forma repetitiva. Por ejemplo un *pager*, siempre es un pager. En contraste, un sistema de escritorio ejecuta una amplia variedad de programas, como hojas de cálculo, juegos, etc. Además nuevos programas son añadidos frecuentemente. Por supuesto puede haber excepciones, podría ocurrir que el programa del sistema embebido fuese actualizado a una nueva versión. Por ejemplo, un teléfono celular podría actualizarse de alguna manera.

Fuertes limitaciones: Todos los sistemas de computación poseen limitaciones en sus métricas de diseño, pero en los sistemas embebidos son muy fuertes. Una métrica de diseño es una medida de algunas características de implementación, como: costo, tamaño, desempeño, y consumo de energía. Los sistemas embebidos generalmente deben ser poco costosos, poseer un tamaño reducido, tener un buen desempeño para procesar datos en tiempo real, y además consumir un mínimo de energía para así extender el tiempo de vida de las baterías o prevenir la necesidad de elementos adicionales de enfriamiento.

Reactivos y tiempo real: Muchos sistemas embebidos deben ser reactivos o reaccionar ante cambios en el ambiente, además de realizar algunos cálculos en tiempo real sin ningún retraso, es decir, se deben tener resultados en tiempos fijos ante cualquier eventualidad. Por ejemplo, el módulo de control de viaje de un automóvil continuamente monitorea la velocidad y los sensores de frenos, reaccionando ante cualquier eventualidad. Ante un estímulo anormal, el módulo de

control debe realizar los cálculos de forma precisa y acelerada para garantizar la entrega de los resultados dentro de un tiempo límite, una violación en este tiempo podría ocasionar la pérdida del control del automóvil. En contraste, un sistema de escritorio se enfoca en realizar cálculos con una frecuencia no determinada y la demora de los mismos no produce fallas en el sistema.

Los sistemas embebidos resultan más flexibles y versátiles que los dispositivos diseñados totalmente por hardware, pero disponen de recursos más limitados que los sistemas de propósito general; estas limitaciones pueden afectar en mayor o menor medida sobre las características del producto final y depende de las especificaciones del problema y de la solución en concreto.

2.2.3.1 Arquitectura de los sistemas embebidos¹⁰

De manera general, un sistema embebido y una PC comparten una arquitectura semejante, tal y como está representada de forma sencilla en la Figura 1.

En la parte central se encuentra un microprocesador, un microcontrolador o un procesador digital se señal (DSP) que conforma la unidad central de proceso (CPU), es decir, la que aporta capacidad de cómputo al sistema, pudiendo incluir memoria interna o externa, o en principio cualquier arquitectura específica, según los requerimientos. Los subsistemas de memoria y de entrada/salida (E/S) se interconectan mediante los buses del sistema (compuestos a su vez por el bus de control, el bus de direcciones y el bus de datos).

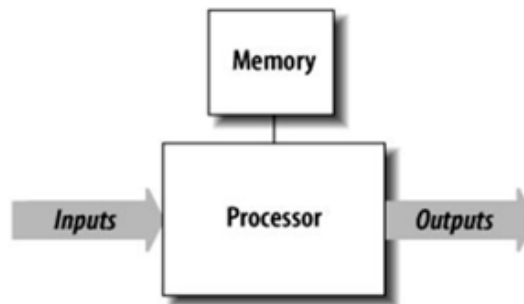
El subsistema de entrada acepta datos del exterior para ser procesados mientras que el subsistema de salida transfiere los resultados hacia el exterior. Lo más habitual es que haya varios subsistemas de entrada y varios de salida. A estos subsistemas se les reconoce habitualmente como periféricos de E/S.

El subsistema de memoria almacena las instrucciones que controlan el funcionamiento del sistema. Estas instrucciones forman el programa que ejecuta el sistema. La memoria también almacena varios tipos de datos: datos de entrada que aún no han sido procesados, resultados intermedios del procesado y resultados finales en espera de salida al exterior.

¹⁰ Marcotegui Zabalza, Fernando. Representación de variables eléctricas en el sistema embebido Mini2440. 2011[En línea].

< <http://academica-e.unavarra.es/bitstream/handle/2454/4283/577623.pdf?sequence=1>>

Figura 1. Esquema sencillo de una PC o de un sistema embebido



Fuente: Marcotegui Zabalza, Fernando. Representación de variables eléctricas en el sistema embebido Mini2440. [En línea].

< <http://ac.ademica-e.unavarra.es/bitstream/handle/2454/4283/577623.pdf?sequence=1> >

No obstante, como ya se ha comentado, las restricciones que presentan los sistemas embebidos hacen que su hardware presente una configuración diferente que las PCs, normalmente presenta menos recursos que estas últimas. Tomando como ejemplo la PC arbitraria representada en la Figura 2, la cual cuenta con una gran memoria principal para albergar al sistema operativo, aplicaciones y datos, y una interfaz para dispositivos de almacenamiento masivo (discos duros y DVD/CD-ROMs), posee además varios dispositivos de E/S para el usuario (teclado, ratón, micrófono, pantalla gráfica y audio) así como de conectividad (periféricos y red). El rápido procesador requerirá de un gestor del sistema para monitorizar la temperatura del núcleo y las tensiones de alimentación, así como para poder generar un *reset*.

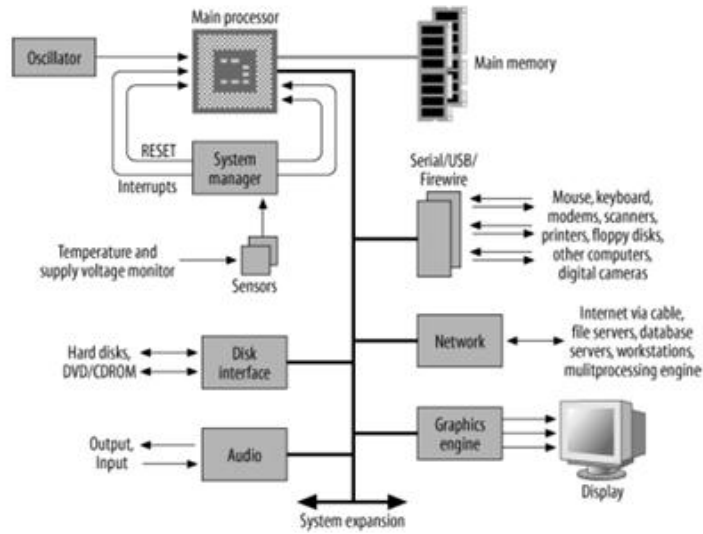
Algunos tipos de sistemas embebidos, los de más alto rendimiento, pueden presentar un aspecto semejante a este. Por ejemplo, si funcionan como router o gateway, necesitarán una o varias interfaces de red, amplia memoria y alta velocidad de operación. Puede que también requieran algún tipo de interfaz de usuario como parte de su aplicación embebida, y en el caso extremo, simplemente son computadoras convencionales dedicadas a una tarea específica.

Sin embargo, dependiendo de las especificaciones que tengan que cumplir, los sistemas embebidos irán reduciendo los recursos disponibles.

Los sistemas embebidos más pequeños utilizan microcontroladores como procesador, con la ventaja de que este procesador incorporara gran parte de las funcionalidades de la computadora en un solo chip.

En la Figura 3 se muestra un sistema embebido arbitrario basado en un microcontrolador genérico.

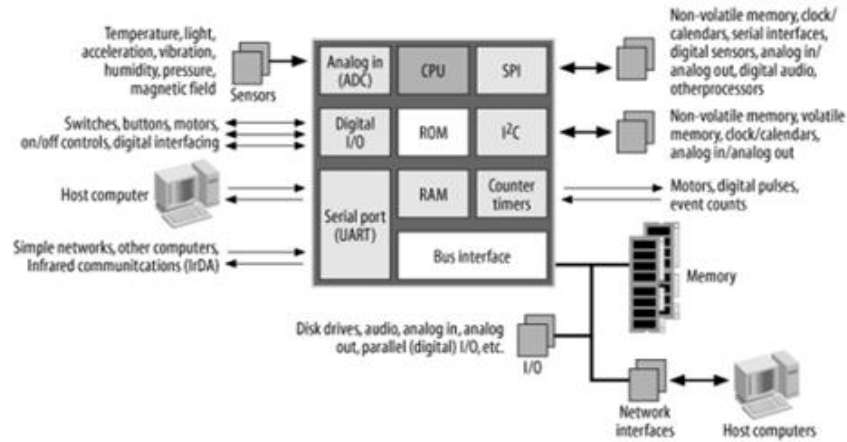
Figura 2. Diagrama de bloques de una computadora genérica



Fuente: Marcotegui Zabalza, Fernando. Representación de variables eléctricas en el sistema embebido Mini2440. [En línea].

< <http://academica-e.unavarra.es/bitstream/handle/2454/4283/577623.pdf?sequence=1> >

Figura 3. Diagrama de bloques de un sistema embebido arbitrario



Fuente: Marcotegui Zabalza, Fernando. Representación de variables eléctricas en el sistema embebido Mini2440. [En línea].

< <http://academica-e.unavarra.es/bitstream/handle/2454/4283/577623.pdf?sequence=1> >

Un microcontrolador es un circuito integrado que incluye una CPU, memoria y circuitos de E/S. Entre los subsistemas de E/S que incluyen los microcontroladores se encuentran los temporizadores, los conversores análogo-digital (ADC) y digital-análogo (DAC), así como también los canales de comunicación serial.

Estos subsistemas de E/S se suelen optimizar para aplicaciones específicas (por ejemplo audio, video, procesos industriales, comunicaciones, etc.) y se implementan dentro del microcontrolador como bloques de subsistemas.

A continuación se describen los elementos que con más frecuencia están presentes en las arquitecturas embebidas, aunque, como se ha comentado, pueden aparecer o no en un dispositivo en concreto. Estos elementos son muchas veces el objeto de las propias restricciones, siendo los más críticos el procesador y la memoria.

Microprocesador: Es el encargado de realizar las operaciones de cálculo principales del sistema y el que aporta capacidad de cómputo. Ejecuta código (conjunto de instrucciones) para realizar una determinada tarea y dirige el funcionamiento de los demás elementos que integran el sistema. La potencia de procesamiento se puede medir en términos del número de instrucciones que es capaz de ejecutar por unidad de tiempo, generalmente en MIPS (millones de instrucciones por segundo). Otra característica importante es el ancho de palabra del registro, que puede variar típicamente de 8 y 64 bits. Las PCs utilizadas hoy en día utilizan ya exclusivamente procesadores de 32 y 64 bits, sin embargo todavía es común en los sistemas embebidos utilizar procesadores más económicos de 8 o 16 bits.

Memoria RAM o principal: En ella se encuentra almacenado el código de los programas que el sistema puede ejecutar así como los datos. Su característica principal es que debe tener un acceso de lectura y escritura lo más rápido posible para que el microprocesador no pierda tiempo en tareas que no sean meramente de cálculo. Al ser volátil el sistema requiere de un soporte donde se almacenen los datos incluso sin disponer de alimentación o energía. La cantidad de memoria debe ser dimensionada correctamente por el diseñador y también puede afectar a la elección del procesador. En general, el ancho de palabra del registro del procesador establece el límite superior de la cantidad de memoria a la que pueda acceder. Por ejemplo, un registro de direcciones de 8 bits solo puede seleccionar una de entre 256 posiciones de memoria únicas.

Cache: Es una memoria más rápida que la principal en la que se almacenan los datos y el código accedido últimamente. Dado que el sistema realiza microtareas, muchas veces repetitivas, la caché permite ahorrar tiempo ya que no hará falta ir a memoria principal si el dato o la instrucción ya se encuentra en la caché. Dado su alto precio tiene un tamaño muy inferior con respecto a la principal. En el interior del chip de microprocesador se encuentra una pequeña caché, pero normalmente se tiene una mayor en otro chip de la placa madre.

Disco Duro: En él la información no es volátil y además puede conseguir capacidades muy elevadas. A diferencia de la memoria que es de estado sólido este suele ser magnético. Pero su exclusivo tamaño a veces lo hace no viable

para PCs embebidas, con lo que se requieren soluciones como unidades de estado sólido. Otro problema que presentan los dispositivos magnéticos, a la hora de integrarlos en sistemas embebidos, es que llevan partes mecánicas móviles, lo que los hace inadecuados para entornos donde estos estarán expuestos a ciertas condiciones de vibración. Existen en el mercado varias soluciones de esta clase (DiskOnChip, CompactFlash, IDE Flash Drive, etc.) con capacidades suficientes para la mayoría de sistemas embebidos (desde 2 MB hasta más de 1 GB).

BIOS-ROM (Basic Input & Output System-Read Only Memory): *Memoria de solo lectura para el sistema básico de entrada y salida*, es un chip de memoria específico para albergar un código fijo, necesario para inicializar la computadora y para poner en comunicación los distintos elementos de la placa madre. En muchos sistemas embebidos no existe tal memoria BIOS y el proceso de arranque lo realizan programas instalados en otro tipo de memorias, como por ejemplo Flash, a los que se conoce como *bootloaders*.

CMOS-RAM: Es un chip de memoria de lectura y escritura alimentado con una pila donde se almacena el tipo y ubicación de los dispositivos conectados a la placa madre (disco duro, puertos de entrada y salida, etc.). Además contiene un reloj en permanente funcionamiento que ofrece al sistema la fecha y la hora.

Memoria Flash: Es una tecnología de almacenamiento (derivada de la memoria E²PROM), que permite la lectura/escritura de múltiples posiciones de memoria en la misma operación. Existen dos tecnologías de memoria Flash (tipo NAND y tipo NOR, según el tipo de puertas lógicas utilizadas), cada una con sus características de densidad de almacenamiento, costo, velocidad de lectura/escritura/borrado, fiabilidad, etc.

Chipset: Chip que se encarga de controlar las interrupciones dirigidas al microprocesador, el acceso directo a memoria (DMA) y al bus ISA, además de ofrecer temporizadores y otras funcionalidades. Es frecuente encontrar la CMOS-RAM y el reloj de tiempo real en el interior del chip.

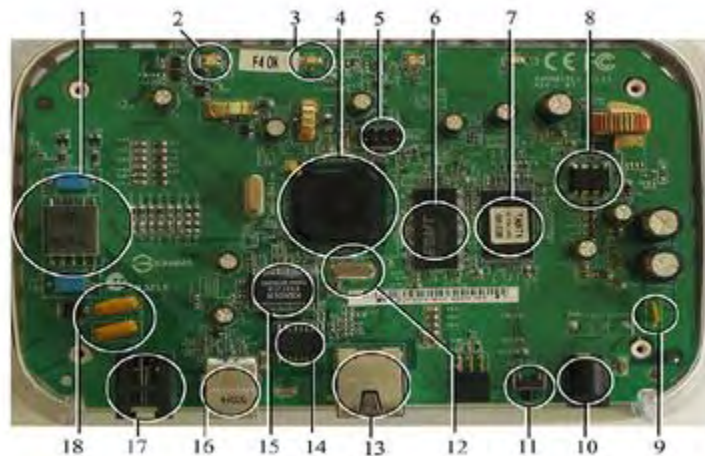
Periféricos de E/S: Los sistemas embebidos pueden tener una gran variedad de subsistemas de entrada/salida, tanto de control como de datos. Dependiendo de su función, pueden disponer de una o varias interfaces para interactuar con el usuario, tales como teclado, monitores, botones, dispositivos táctiles, etc. En otras ocasiones, sin embargo, su tarea se ejecuta en un segundo plano y su existencia puede no ser aparente para el usuario, como por ejemplo el sistema antibloqueo de frenos de un automóvil. Uno de los subsistemas de entrada/salida más comunes es el llamado general-purpose input-output (GPIO) que está basado en la lectura o escritura de valores digitales binarios en ciertos pines del circuito. Se puede utilizar para leer el estado de interrupciones, botones u otros dispositivos externos, o bien para activar o desactivar acciones, procesos, etc. Aunque la mayoría de los microcontroladores tienen otros subsistemas de E/S además de

GPIO, se suele proporcionar la posibilidad de convertir los otros subsistemas a GPIO si las funcionalidades de aquellos no son requeridas.

Muchos sistemas embebidos necesitan además entradas analógicas, por ejemplo procedentes de sensores de variables físicas, como temperatura, aceleración, presión acústica, intensidad lumínica, campo magnético, etc. para registrarlas, monitorizarlas o procesarlas. En tal caso el sistema deberá disponer también de un conversor ADC para digitalizar las señales.

Algunos sistemas embebidos disponen de puertos serie, que permiten la comunicación bien con una computadora anfitriona o *host*, con un modem (Figura 4), con otro sistema embebido o quizás acceder a una sencilla red. Existen formas especializadas de interfaces serie, tales como SPI e I²C, que permiten al procesador tener acceso a dispositivos periféricos tales como memorias externas, chips de reloj para sincronización, sensores digitales o analógicos u otros procesadores. Algunos sistemas embebidos también incluyen interfaces de red como Ethernet, USB o incluso *wireless*.

Figura 4. Componentes de un router/modem ADSL Natgear DG632. Las partes marcadas incluyen un microprocesador (4), una memoria RAM (6) y una Flash (7)



Fuente: Wikipedia. sistema embebido: Componentes de un router ADSL Natgear. [En Linea].
< http://es.wikipedia.org/wiki/Sistema_embebido>

2.2.3.2 Software en los sistemas embebidos

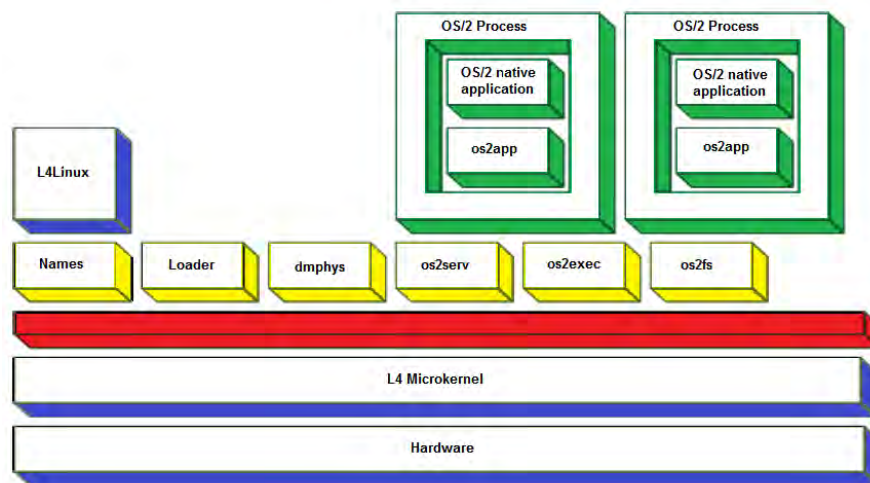
Como se mencionó en la sección 2.2.3.1, no existe una única arquitectura hardware para todos los dispositivos embebidos, sino que hay mucha variedad y flexibilidad en las configuraciones y cada una está condicionada por sus necesidades particulares. De la misma manera, el software que se ejecuta sobre los dispositivos no es el mismo para todos, sino que depende en

gran medida de la aplicación concreta, de la configuración hardware en particular, del número de capas que estén dando soporte por debajo a ese software, etc.

El concepto de software en el ámbito embebido es similar al del de las computadoras convencionales, aunque tiene ciertas peculiaridades que afectan a los programadores.

Básicamente, la idea que debería tener en mente una persona (con experiencia desarrollando aplicaciones para una PC) a la hora de desarrollar software embebido es que su ámbito de actuación va a estar ahora más cerca del hardware, tomando como referencia por ejemplo, el modelo de capas del sistema operativo osFree, mostrado en la Figura 5, significara que el programador va a trabajar más abajo en el diagrama, con menos capas de abstracción (librerías, interfaces de programación de aplicaciones (APIs), dependencias, etc.) en las que apoyarse. En definitiva, va a programar a más bajo nivel. Por otra parte, dado que los recursos son más limitados, el código debe hacer uso de ellos de forma especialmente eficiente. Como consecuencia, la programación se vuelve más específica para cada hardware, lo cual hace que las aplicaciones embebidas sean, en general, menos portables que en el mundo de las PCs.

Figura 5. Arquitectura del sistema operativo OSfree



Fuente: Apesol. Arquitectura de sistema operativo OSFree: [En línea].
 < http://stone-head.org/talks/2006/SENCICO-Soltech-SistemaOperativo_Rudy.pdf >

Sistemas operativos embebidos: Se dan casos como el del reloj digital en los que la estructura es tan simple y las tareas a realizar son tan sencillas, que el desarrollo de software conlleva en sí mismo desde el nivel de aplicación hasta el sistema operativo, dando como producto una única imagen ejecutable que contiene todas las tareas para el sistema.

En otros casos en cambio, el software se ejecutara sobre una compleja estructura estratificada de librerías, APIs, dependencias, etc, tales como las aplicaciones para smartphones que tan de moda están hoy en día, Este software a su vez reposa sobre un sistema operativo, el cual toma una arquitectura adecuada al dispositivo.

El sistema operativo funciona de manera semejante al de una PC, gestionando los procesos básicos del sistema, aunque en este caso estará diseñado para ser más compacto y eficiente, y por tanto carecerá de algunas funcionalidades que ofrecen los sistemas operativos mayores.

Entre los dispositivos que cuentan con una estructura, la cual puede considerarse como un sistema operativo”, destaca el conjunto de los mencionados smartphones, PDAs, tablets, teléfonos móviles convencionales y localizadores GPS.

Existen en el mercado varias empresas que están desarrollando sistemas operativos diferentes para este tipo de dispositivos a un nivel muy competitivo entre ellas, presentando diferentes filosofías tecnológicas y estrategias empresariales. En la Tabla 1 se muestra un resumen de algunos sistemas operativos embebidos vigentes hasta el año 2009.

Tabla 1. Sistemas operativos embebidos vigentes hasta el año 2009

Nombre	Gratuito	Código Abierto	Última Actualización	Uso Principal
<i>VxWorks</i>)	No	No	Feb, 2009	Modems, Firewalls, WirlessRouters
<i>QNX</i>	No	No	May, 2009	Dispositivos de tiempo real
<i>Windows CE</i>	No	No	Nov, 2007	PDAs, Smartphones
<i>Windows Mobile</i>	No	No	May, 2009	PDAs, Smartphones, Dispositivos multimedia
<i>Ecos</i>	Si	Si	Mar, 2009	Dispositivos de tiempo real
<i>LynxOS</i>	No	No	Sep, 2006	Aeronaves, automóviles
<i>TinyOS</i>	No	Si	Ago, 2008	Redes de sensores, motores
<i>PalmOS</i>	No	No	Sep, 2006	PDAs, Smartphones
<i>Garnet OS</i>	No	No	May, 2009	PDAs, Smartphones

<i>Linux Embebido</i>	Si	Si	Ago, 2006	PDA's, Smartphones, WirlessRouters
<i>Symbian</i>	No	No	Oct, 2008	Smartphones
<i>Inferno</i>	No	Si	Feb, 2007	Sistemas distribuidos de tiempo real

En cualquier caso, éste es un sector en auge y con buenas perspectivas de mercado. Esta cuestión se discutirá en mayor profundidad más adelante.

Lenguajes de programación: Como se ve, la programación embebida se hace a más bajo nivel, pero esto no significa que el programador tenga que descender todo el tiempo al lenguaje ensamblador. El lenguaje utilizado en la mayoría de las aplicaciones embebidas es C. Se trata de un lenguaje simple y relativamente fácil de aprender, que dispone de las estructuras típicas de los lenguajes de alto nivel pero que, a su vez, dispone de construcciones del lenguaje que permiten un control a muy bajo nivel.

Existen compiladores, prácticamente para cualquier procesador, que suelen ofrecer también, extensiones al lenguaje de programación utilizado, lo cual posibilita mezclar código en lenguaje ensamblador con código en lenguaje C o acceder directamente a memoria o dispositivos periféricos. En ocasiones, si la arquitectura así lo permite, es posible hacer una programación orientada a objetos en C++ o C#, aunque la programación embebida es tradicionalmente estructurada.

Otros lenguajes de programación que se vienen utilizando, sobre todo en los dispositivos que cuentan con una mayor capacidad para añadir nuevas capas, son Java, Pearl, Python, PHP, etc.

Sistemas en tiempo real: Una subclase especial dentro de los sistemas embebidos son los sistemas de tiempo real. Dentro de las especificaciones que deben cumplir estos sistemas, resulta crítico el tiempo de respuesta del sistema, debido a que un sistema de tiempo real debe responder dentro de un intervalo restringido de tiempo a eventos externos mediante la ejecución de la tarea asociada con cada evento.

Los sistemas de tiempo real se pueden caracterizar como blandos o duros. Si un sistema de tiempo real blando no cumple con sus restricciones de tiempo, simplemente se degrada el rendimiento del sistema, pero si el sistema es de tiempo real duro y no cumple con sus restricciones de tiempo, el sistema fallará, generando consecuencias catastróficas, por ejemplo en el caso del sistema de control del vuelo de un avión.

2.2.4 Historia y situación actual de los sistemas embebidos

Tal y como se definieron los sistemas embebidos en la sección 2.2.1, resulta imposible que hubieran podido existir antes de 1971, ya que fue en ese año cuando la compañía Norteamericana Intel lanzó al mercado el primer microprocesador de la historia, el 4004, diseñado para ser utilizado en una línea de calculadoras producida por la compañía japonesa Busicom.

Dos años antes, en 1969, Busicom había solicitado a Intel diseñar y fabricar un conjunto de circuitos integrados para una serie de modelos de calculadoras para negocios, de manera que cada calculadora tuviera su propio circuito específico.

Sin embargo, la propuesta de Intel fue diseñar un circuito de propósito general que pudiera ser utilizado en la línea entera de calculadoras. Este procesador de propósito general, el 4004, fue diseñado para leer y ejecutar un conjunto de instrucciones que estaban almacenadas como software en un chip externo de memoria. La idea de Intel era que sería el software lo que proporcionaría a cada calculadora su conjunto único de características.

Aquel microprocesador tuvo un éxito instantáneo y su uso creció progresivamente durante la siguiente década. Las primeras aplicaciones embebidas incluyeron pruebas espaciales no tripuladas, semáforos de tráfico computarizados y sistemas de control para la aviación.

En la década de 1980, los sistemas embebidos fueron ganando espacios conforme se instauraba la era de las microcomputadoras, y acabaron llegando a todos los rincones de la vida personal y profesional de los países desarrollados. Como ya se dijo, en estos países, la mayoría de los equipos electrónicos que se encuentran en las viviendas o en los puestos de trabajo hoy en día son sistemas embebidos. En la última década la tendencia hacia la reducción de tamaño de las computadoras personales y su transición hacia otro tipo de dispositivos con nuevas funcionalidades como los Smartphones y demás, ha hecho aumentar aún más la repercusión de este tipo de sistemas.

Actualmente, ante los retos que presenta la globalización y la fuerte presión de los mercados emergentes, casi todos los sectores están haciendo esfuerzos que les permitan aumentar sus cuotas de competitividad. La gran aplicabilidad de los sistemas embebidos en cualquier ámbito sectorial, así como el valor añadido que aportan los mismos a los productos que los contienen, hace que el desarrollo de estos sistemas sea un área estratégica preferente para muchas empresas que buscan precisamente este aumento de su competitividad. Los sistemas embebidos están en todos lados, en las casas, oficinas, automóviles, fabricas, hospitales y electrónicos. Físicamente los sistemas embebidos pueden ser desde dispositivos portátiles como teléfonos, relojes digitales y reproductores mp3, hasta luces de

tráfico, controladores de fábricas o los sistemas que controlan las plantas nucleares.

De esta manera, los sistemas embebidos van a jugar un papel vital en nuestra sociedad y se supone revolucionarán los sectores de actividad, como son el sector médico, el de medios de transporte o el de automatización industrial, entre otros. Como dato que refleja la importancia que este tipo de sistemas ha adquirido, se estima que el volumen de mercado global en 2013 ronda los 120.000 millones de dólares. De acuerdo con un estudio de la fundación OPTI y la Fundación ASCAMM, en 2009 Europa era el máximo representante en el campo de los sistemas embebidos a nivel mundial, donde se esperaba que para el año 2010, el porcentaje de inversión en investigación y desarrollo (I+D) en sistemas embebidos, sobre el total de la inversión en I+D, fuese del 14%¹¹. Aun así, esta posición ventajosa podría perderse a favor de los Estados Unidos o de algunos países emergentes, sobre todo los asiáticos. Los Estados Unidos tienden a utilizar los recursos de los sistemas embebidos obtenidos en las aplicaciones militares e industriales y los países asiáticos poseen un amplio mercado nacional y el *know-how* tecnológico en fabricación que ponen en peligro la posición de liderazgo Europeo.

Europa, por otro lado, posee una elevada cualificación profesional en este ámbito, unos mercados desarrollados y una buena infraestructura. Una parte de esta infraestructura la conforman las plataformas Europeas relacionadas con los sistemas embebidos. Las más destacadas son la Plataforma ARTEMIS y la Plataforma ENIAC, a través de las cuales se ha creado una gran red de contactos que facilita la transferencia de conocimiento y potencia la colaboración a nivel Europeo. Su importancia es tal que llega a influir en la toma de decisiones de las líneas de investigación en el marco Europeo.

2.3 PLACA DE DESARROLLO MINI2440¹²

En este capítulo se da una descripción detallada del sistema embebido MINI2440, que constituye la herramienta central de este proyecto, ya que sobre él se ha desarrollado la aplicación que permite el control general de todos los parámetros de medida del osciloscopio digital implementado.

¹¹ Arilla, Cristina. Arribas, Laura. Tendencias y aplicaciones de los sistemas embebidos en España. Estudio de Prospectiva. [En línea].

< <http://www.opti.org/publicaciones/pdf/texto131.pdf>>

¹² FriendlyARM MINI2440. Manual de usuario. [En línea]

< http://www.friendlyarm.net/dl.php?file=mini2440_manual_20110421.pdf>

[Citado junio 20 de 2012].

Este dispositivo embebido fabricado en China y distribuido por compañías como FriendlyARM, Industrial ARMworks, Andahammer, Hiteg o ThaiEasyElec, es un sistema de computación versátil y de propósito múltiple, con un tamaño de bolsillo, aunque también con notables limitaciones de recursos con respecto a una PC actual. En un hipotético eje que tiene en un extremo a las PCs y en el otro a los dispositivos hardware-monotarea, la MINI2440 se ubicaría en una zona intermedia, compartiendo espacio con smartphones, tablets y similares.

La MINI2440, al ser concebida como una placa de desarrollo, permite que programadores desarrollen y prueben en esta sus aplicaciones, las cuales generalmente son de carácter muy diferente entre sí, debido a la amplia variedad de posibilidades que presenta el dispositivo; permite la conexión de una cámara web, un módulo GPS o una tarjeta de televisión, entre otras. Además de su versatilidad, otra característica como su robustez y su precio reducido hacen que sea de gran utilidad en el campo de la investigación.

Es comercializada en distintas versiones, presentando ligeras diferencias en cuanto a sus recursos y presentaciones. Encontramos por ejemplo, a través de FriendlyARM sin display gráfico, o con un display LCD táctil fabricado bien por Sony o bien por Toppoly, en versiones de 3.5" o de 7". La memoria NAND Flash puede ser de 64, 128 o 256 MB.

El kit concreto sobre el cual se llevo a cabo el desarrollo del proyecto cuenta con un display gráfico LCD táctil de 3.5" 320x240 de Toppoly, un microprocesador Samsung S3C2440AL (ARM920T), el cual trabaja a una frecuencia máxima de 533MHz, memoria NAND Flash de 256 MB, memoria RAM de 64 MB, memoria NOR para la BIOS de 2 MB, conector USB Host y USB Device, conector RS232, incluye además un adaptador de alimentación eléctrica, cables de conexión serie RS232 y USB, un cable de red Ethernet cruzado y un adaptador JTAG, cabe resaltar también, que se cuenta con el web oficial¹³ de FriendlyARM en el cual podremos encontrar material de consulta, paquetes de software, tales como sistemas operativos Qtopia, WinCE 5.0 y 6.0, aplicaciones test para los subsistemas, esquemáticos y datasheets de los módulos físicos (Microprocesador, RAM, Flash, Display, etc) con los cuales cuenta la placa.

El kit fue adquirido desde la página web oficial quickembedded¹⁴. Un pack similar se muestra en la Figura 6, aunque en este caso el kit contiene un DVD con material idéntico al que se encuentra en el web oficial de FriendlyARM.

¹³FriendlyARM MINI2440. Manual de Usuario. [En línea].

< http://www.friendlyarm.net/dl.php?file=mini2440_manual_20110421.pdf>

¹⁴Quickembedded. Kit FriendlyARM MINI2440. [En línea].

< <http://www.quickembed.com/Tools/Shop/ARM/200905/4.html>>

Cabe resaltar, en el aspecto negativo, la falta de manuales y material explicativo para esta placa, que sean enfocados para principiantes en el mundo de la programación embebida y que estén estructurados apropiadamente y traducidos al menos al inglés, ya que la información contenida en el Web oficial de FriendlyARM es escasa, dispersa y en su mayor parte se encuentra solo disponible en chino.

Aunque la comunidad de usuarios en internet de ésta y otras placas de desarrollo realizan un gran aporte en este sentido (foros, blogs, sitios ftp, etc.), los cuales constituyeron un gran y valioso aporte en el desarrollo del proyecto.

Figura 6. Kit MINI2440 de FriendlyARM



Quickembedded. Kit FriendlyARM MINI2440. [En línea].
< <http://www.quickembed.com/Tools/Shop/ARM/200905/4.html>>

2.3.1 Arquitectura interna de la MINI2440

La MINI2440 está construida sobre una placa de 10 x 10 cm, y cuenta con una gran variedad de interfaces y sistemas de almacenamiento que están montados sobre ella, tal y como se puede observar en la Figura 7.

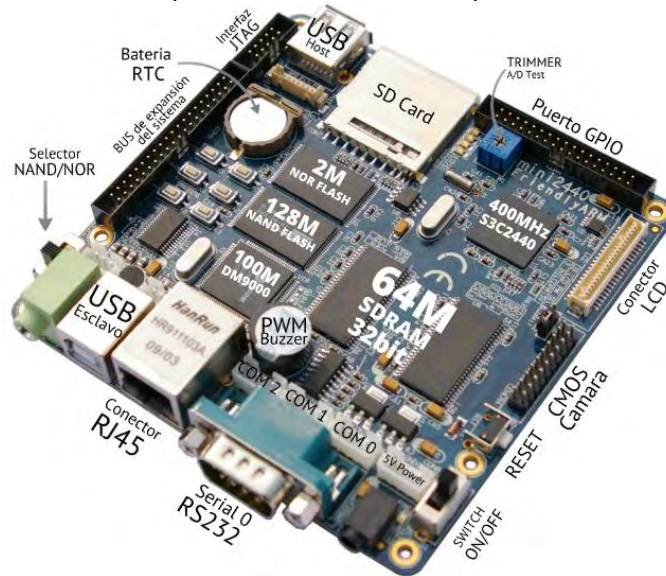
- **Microprocesador.**
 - S3C2440AL de arquitectura ARM920T de 32 bits.
 - Frecuencia de 450 MHz (Max. 533 MHz).
 - 200 MIPS efectivas a 180 MHz.

- **Memoria RAM.**
 - SDRAM 64 MB montada sobre la placa.

- Bus de datos de 32 bits.
- Frecuencia de reloj hasta 100 MHz.
- **Memoria Flash.**
 - NAND Flash de 256 MB.
 - NOR Flash de 2 MB.
- **Interfaz LCD.**
 - Interfaz táctil resistiva de 4 canales conectada al ADC de 10 bits de resolución.
 - Soporte para LCD STN escala de grises de 4 y 16 niveles, 256 y 4096 colores.
 - Soporte para LCD TFT escala de grises de 4 y 16 niveles, 256, 64k colores y True Color.
 - Pantalla de 3.5", aunque existen versiones de hasta 12.1".
 - Resolución máxima de pantalla de 1024 x 768 pixeles.
 - Configuración estándar para el LCD táctil NEC 320 x 240/3,5" TFT True color.
 - Conectada a una interfaz de alimentación on-board de 12V, para la alimentación del módulo backlight LCD TFT 12V CCFL.
- **Otras interfaces y accesorios externos.**
 - Una interfaz 1/100M Ethernet RJ-45 (chip de red DM9000).
 - 3 puertos serie, uno configurado para RS-232 (COM0).
 - Un USB host.
 - Una interfaz USB de tipo Slave B.
 - Una interfaz para tarjetas de almacenamiento SD, sin límite de tamaño.
 - Una interfaz de salida de audio stereo.
 - Un micrófono integrado.
 - Una entrada de micrófono.
 - Una interfaz JTAG de 10 pines de 2.0 mm de pitch.
 - 4 LEDs de usuario.
 - 6 botones de usuario (userbuttons), con conexión a conectores GPIO y de usuario de 8 pines.
 - Un botón de reinicio (reset), conectado a un chip de reset.
 - Un interruptor de selección de arranque desde Flash NAND/NOR.
 - Un potenciómetro, para test del ADC.
 - Un timer PWM (buzzer).
 - Un chip de bus IIC AT24C08 para datos de configuración o prueba, abarca 256 bytes.
 - Una interfaz de cámara de 20 pines de 2.0 mm de pitch.
 - Una interfaz de alimentación de 5V, con un interruptor de encendido y luz indicadora.
- **Fuente de reloj del sistema.**
 - De cristal pasivo.
 - Reloj interno de tiempo real (con batería de litio de respaldo).

- **Interfaz de expansión.**
 - Una interfaz GPIO de 34 pines de 2.0 mm de pitch.
 - Una interfaz de bus del sistema de 40 pines de 2.0 mm de pitch.

Figura 7. Configuración de los componentes hardware en la placa de desarrollo MINI2440



Fuente: FriendlyARM. Manual MINI2440. [En línea].
 <http://www.friendlyarm.net/dl.php?file=mini2440_manual_20110421.pdf>

2.3.1.1 Microprocesador ARM S3C2440AL

ARM¹⁵ (Advanced RISC Machines): Es la denominación de un conjunto de microprocesadores con una arquitectura y un juego de instrucciones RISC (Reduced Instruction Set Computer) de 32 bits que fueron diseñados por Acorn Computers y desarrollados por ARM Holdings Ltda, ambas compañías británicas.

Fue originalmente concebida por Acorn Computers como procesador para PCs de sobremesa, un mercado dominado por la familia x86 que era usada por los computadores IBM PC. Sin embargo, la relativa sencillez de los procesadores ARM los hacían apropiados para aplicaciones de bajo consumo eléctrico. Esto los ha hecho liderar el mercado de la electrónica móvil y de los dispositivos embebidos como microprocesadores y microcontroladores pequeños y asequibles. Atendiendo a datos de producción, la arquitectura ARM es el repertorio de instrucciones de 32 bits más extendido actualmente. Hasta el año 2007 el 98% de los más de mil millones de teléfonos móviles vendidos anualmente usaban al

¹⁵ Wikipedia. Arquitectura ARM. [En línea].
 < http://es.wikipedia.org/wiki/Arquitectura_ARM>

menos un procesador ARM. En 2009 aproximadamente el 90% de los procesadores RISC de 32 bits empleados en dispositivos empotrados eran ARM.

Estos procesadores son ampliamente usados en electrónica de consumo, incluyendo PDAs, iPods y otros reproductores, videoconsolas, portátiles, calculadoras y periféricos de computador como discos duros y routers. La arquitectura ARM es licenciable. Algunas compañías que son o han sido licenciadas incluyen: Alcatel, Atmel, Broadcom, Cirrus Logic, Digital Equipment Corporation, Freescale, Intel (a través de DEC), LG, Marvell Technology Group, NEC, NVIDIA, NXP (Previamente Philips), Oki, Qualcomm, Samsung, Sharp, ST Microelectronics, SimbiosLogic, Texas Instruments, VLSI Technology, Yamaha y ZiiLABS.

S3C2440AL: El núcleo de la MINI2440 es el microprocesador S3C2440AL, de arquitectura ARM920T de 32 bits, fabricado por la empresa Surcoreana Samsung. La arquitectura ARM920T está compuesta por un núcleo ARM9TDMI más una unidad de gestión de memoria (MMU, memory management unit) y memoria cache. El núcleo es un dispositivo de arquitectura Harvard, implementado utilizando un pipeline de cinco etapas: fetch, decode, execute, memory, y wire. Las caches de instrucciones y datos tienen 16 KB cada una, con una longitud de 8 palabras. El procesador implementa una arquitectura ARM enriquecida v4 MMU para proporcionar traducción y para permitir chequeos de permiso para la direcciones de datos e instrucciones.

La interfaz del ARM920T con el resto del sistema se hace sobre buses unificados de datos y direcciones. Esta interfaz habilita la implementación de un esquema de bus tipo Advanced High-performance Bus Architecture (AMBA), Advanced High-performance Bus (AHB). El ARM920T además puede soportar una macrocelda ETM (Embedded Trace Macrocell) para el rastreo en tiempo real de datos e instrucciones.

Tipos de datos: Los tipos de datos soportados por el S3C2440AL son:

- Byte 8 bits
- Halfword 16 bits
- Word 32 bits
- Double word 64 bits.

Modos de ejecución: El microprocesador S3C2440AL soporta los siete modos de ejecución que se muestran en la Tabla 2.

Tabla 2. Modos de ejecución

Modo de ejecución	Numero de modo	Descripción
<i>User (usr)</i>	0b10000	Modo de ejecución normal
<i>FIQ (fiq)</i>	0b10001	Interrupciones rápidas
<i>IRQ (irq)</i>	0b10010	Interrupciones de propósito general
<i>Supervisor (svc)</i>	0b10011	Modo protegido para el sistema operativo
<i>Abort (abt)</i>	0b10111	Permite implementar memoria virtual y/o protección de memoria
<i>Undefined (und)</i>	0b11011	Soporte para emulación software de coprocesadores
<i>System (sys)</i>	0b11111	Ejecución de tareas privilegiadas del sistema operativo

Fuente: FriendlyARM. Manual MINI2440. [En línea].

< http://www.friendlyarm.net/dl.php?file=mini2440_manual_20110421.pdf>

La mayoría de las aplicaciones se ejecutan en modo User. En este modo el programa en ejecución no puede acceder a ciertos recursos (memoria, coprocesadores, periféricos, etc) ni cambiar de modo, a menos que tenga lugar una excepción (esta se produce mediante la instrucción SWI). Esto permite al sistema operativo controlar el uso de los recursos.

Todos los modos excepto el modo User son modos privilegiados. Tienen total acceso a los recursos del sistema y pueden cambiar el modo de ejecución libremente. Cinco de ellos son los modos de excepción: FIQ, IRQ, Supervisor, Abort y Undefined.

Se accede automáticamente a estos modos cuando se produce una excepción. Cada uno de ellos posee una serie de registros adicionales para evitar corromper el estado del modo User cuando se produce la excepción.

El modo restante es el modo System, al cual no se accede mediante una excepción y emplea los mismos registros que el modo User. Al ser un modo privilegiado carece de las restricciones del modo User. Su uso está indicado para tareas del sistema operativo que necesitan acceder a los recursos del sistema, pero necesitan evitar usar los registros adicionales asociados a los modos de excepción. De este modo se asegura que el estado de la tarea no se altera al ocurrir una excepción.

Registros: En la arquitectura ARM920T, el procesador dispone de 37 registros de 32 bits, cuya distribución se muestra en la Figura 8, y de los cuales se tiene que:

- 6 son registros de estado, aunque no se usan todos los bits.

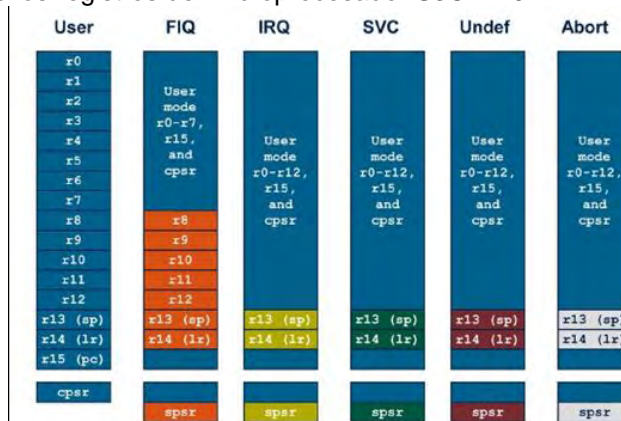
- 31 son de propósito general, incluyendo el contador del programa. De ellos 16 son visibles en todo momento. El resto se emplean para acelerar el procesamiento de excepciones.

Este banco principal de 16 registros (R0 a R15) puede ser direccionado por el código sin necesidad de privilegios, o sea, en modo User. Los registros de este banco, tal y como se muestra en la Figura 8, están dispuestos en bancos parcialmente solapados, siendo el modo de ejecución actual el que controla que banco está actualmente visible. Estos registros pueden clasificarse en función de la forma en que se solapan y en sus propósitos específicos:

- **Registros no solapados, de R0 a R7:** no se solapan, esto significa que cada uno de ellos se refiere al mismo registro físico de 32 bits en todos los modos de ejecución. Son registros de propósito general puros, sin usos especiales impuestos por la arquitectura.
- **Registros solapados, de R8 a R12:** el registro físico al que se refiere cada uno de ellos depende del modo de ejecución actual. Los registros R8 a R12 disponen de dos bancos de registros físicos cada uno. Uno de los bancos está reservado para el modo FIQ y el otro es común para el resto de modos. Estos registros no tienen un cometido especial en la arquitectura, sin embargo, para interrupciones que pueden ser tratadas solamente con los registros R8 a R12, la existencia de un banco aislado para el modo FIQ permite un procesamiento muy rápido de las interrupciones.
- **El registro R13:** es utilizado por el software como puntero de pila (stack pointer o SP).
- **El registro R14:** es el Registro de enlace, (link register o LR). Tiene dos usos especiales en la arquitectura:
 1. Dentro de cada modo, LR contiene la dirección de retorno de la subrutina. El retorno de la subrutina se realiza cargando LR en el contador de programa.
 2. Cuando tiene lugar una excepción el LR del modo de ejecución apropiado se carga con la dirección de retorno de la excepción (sumado a un offset, dependiendo de la excepción). El retorno se realiza de forma similar al retorno de la subrutina, teniendo en cuenta que es preciso restaurar completamente el estado del programa que se estaba ejecutando.

- **El registro R15:** es el contador de programa o program counter “PC”. Contiene la dirección de la siguiente instrucción que se va a ejecutar. Lógicamente, las instrucciones tienen ciertas limitaciones al acceder a él. Leer el PC se suele emplear para direccionar instrucciones y datos cercanos independientes de la posición dentro de un programa. Escribir en el PC supone saltar a la dirección escrita en él.

Figura 8. Esquema de los registros del microprocesador S3C2440AL



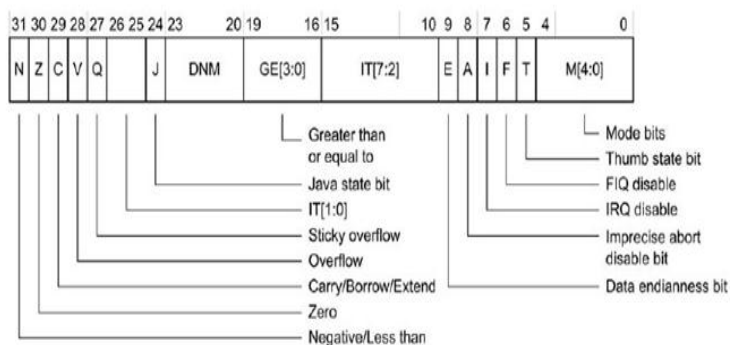
Fuente: FriendlyARM. Manual MINI2440. [En línea].

< http://www.friendlyarm.net/dl.php?file=mini2440_manual_20110421.pdf>

Entre el resto de registros que no son siempre visibles, destacan los registros de estado de programa. El current program status register (CPSR) es accesible en todos los modos de ejecución.

Contiene los bits de condición, el bit para habilitar/inhabilitar las interrupciones, el modo de ejecución y otra información de estado y control. Cada modo de excepción dispone, además, del saved program status register (SPSR), que contiene el CPSR del programa interrumpido. El formato del CPSR y del SPSR se muestra en la Figura 9, y a continuación se explican los bits que lo conforman.

Figura 9. Esquema del registro CPSR/SPSR del S3C2440AL



Fuente: FriendlyARM. Manual Mini2440. [En línea].

http://www.friendlyarm.net/dl.php?file=mini2440_manual_20110421.pdf

- **Bits de condición:** Los bits N, Z, C y V (negativo, cero, acarreo y desbordamiento) son los bits de condición o flags. Se comprueban para saber si una instrucción condicional debe ser ejecutada.
- **Bit Q:** Se usa para indicar que se ha producido un desbordamiento y/o saturación en ciertas instrucciones DSP.
- **Bits GE [3:0]:** Se usan para comprobar relaciones de igualdad y de orden en instrucciones SIMD.
- **Bit E:** controla el endianness del manejo de datos.
- **Bits de inhabilitación de interrupciones:** El bit A deshabilita los Data Aborts imprecisos cuando vale 1. El bit I deshabilita las interrupciones IRQ cuando vale 1, mientras que el bit F hace lo propio con las interrupciones FIQ.
- **Bits de modo:** indican el modo de ejecución del núcleo.
- **Bits T y J:** Se utilizan para seleccionar el juego de instrucciones.

El resto de bits están reservados para futuras expansiones. Es recomendable que el código se escriba de modo que estos bits no sean modificados para prevenir efectos indeseados al ejecutar ese código en versiones futuras de la arquitectura.

Excepciones: La arquitectura ARM920T soporta seis tipos de excepción, cada una de las cuales se procesa mediante un modo privilegiado de ejecución, como se muestra en la Tabla 3. Los siete tipos son:

- Reset, reinicio.
- Intento de ejecutar una instrucción indefinida.
- Interrupciones software (SWI), pueden usarse para implementar llamadas al sistema operativo.
- Data abort, error al acceder a datos en memoria.
- IRQ interrupciones normales.
- FIQ, interrupciones rápidas.

Las excepciones son generadas por fuentes internas o externas para que el procesador trate un evento, como una interrupción externa o el intento de ejecutar una instrucción indefinida. Pueden producirse múltiples excepciones al mismo tiempo.

Cuando ocurre una excepción, se fuerza la ejecución de código desde unas posiciones de memoria fijas, denominadas vectores de excepción y algunos registros comunes son remplazados por otros específicos del modo.

Todos los modos de excepción disponen de versiones privadas de R13 y R14. El modo FIQ, además, dispone de más registros privados adicionales para un procesamiento rápido de las interrupciones. Normalmente el estado del procesador (el contenido de CPSR y el PC) es guardado en el momento que se produce la excepción para poder recuperar el programa original, una vez el tratamiento de la excepción ha concluido.

Cuando se entra en el manejador de la excepción se guarda en R14 la dirección de retorno. Esto sirve para retornar una vez la excepción ha sido tratada y para conocer la dirección de la instrucción causante de la excepción. El registro R13 es privado en cada modo para permitir punteros de pila independientes. Los registros R8-R12 en el modo FIQ son privados para evitar tener que salvarlos y restaurarlos. Reset comparte el modo privilegiado con SWI.

Tabla 3. Tipos y Modos de excepción

Tipo de excepción	Modo	Dirección normal	Dirección High Vector
<i>Reset</i>	Supervisor	0x00000000	0xFFFF0000
<i>Undefined instructions</i>	Undefined	0x00000004	0xFFFF0004
<i>Software interrupt (SWI)</i>	Supervisor	0x00000008	0xFFFF0008
<i>Prefetch Abort (Instruction fetch memory abort)</i>	Abort	0x0000000C	0xFFFF000C
<i>Data Abort (data Access memory abort)</i>	Abort	0x00000010	0xFFFF0010
<i>IRQ (interrupt)</i>	IRQ	0x00000018	0xFFFF0018
<i>System (sys)</i>	FIQ	0x0000001C	0xFFFF001C

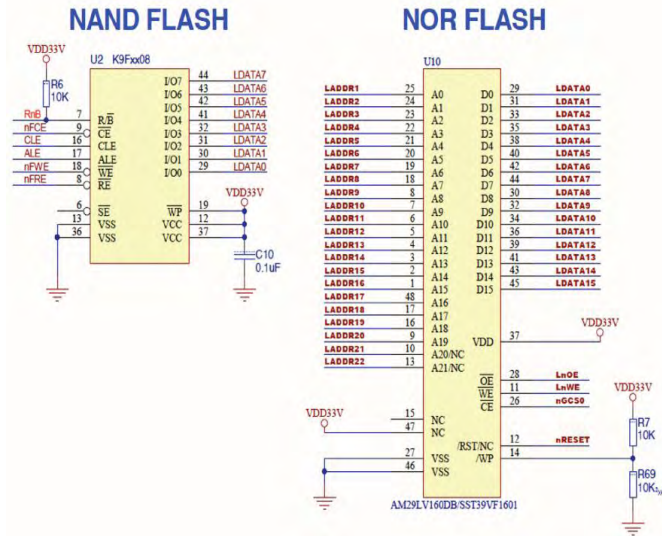
2.3.1.2 Otros subsistemas importantes en la MINI2440

- **Sistemas de almacenamiento Flash:** La MINI2440 cuenta con 2 memorias Flash diferentes. Una es la NOR Flash de 2MB, y la otra es la NAND Flash de 256 MB (Versión empleada en este proyecto). El microprocesador S3C2440AL interno en la MINI2440 permite el arranque desde cualquiera de estas 2 memorias a través del interruptor S2 incorporada en la placa.

La memoria NAND Flash albergara la imagen binaria del Sistema Operativo ejecutado por la MINI2440, mientras que la memoria NOR Flash de 2MB, alberga el gestor de arranque o bootloader (supervivi), específico para los microprocesadores de Samsung. El arranque desde la NOR Flash se hará

para transferir a la NAND Flash tanto la imagen binaria del sistema operativo que ejecutara la placa como el gestor de arranque de este, también permitirá llevar a cabo tareas de recuperación, en el caso de que el bootloader de la memoria NAND estuviese dañado, configurar un arranque automático desde distintas fuentes, particionar la NAND, configurar opciones de arranque para el sistema operativo, etc.

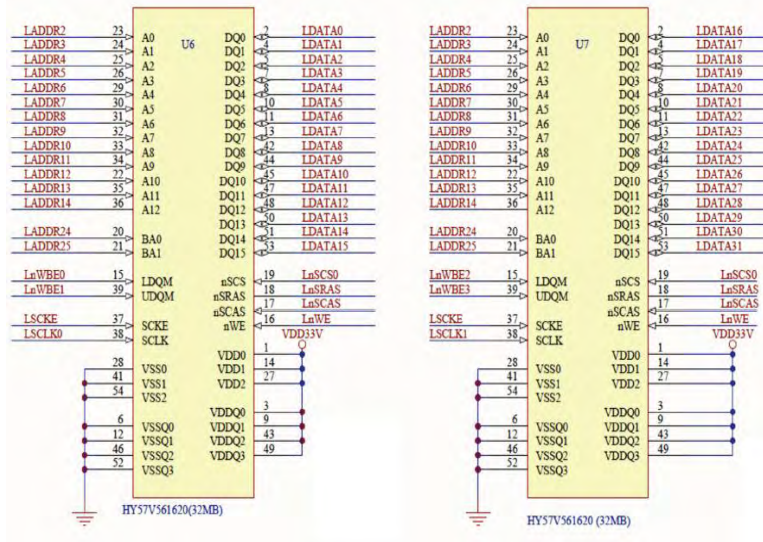
Figura 10. Esquema de las 2 memorias Flash incorporadas en La MINI2440



Fuente: FriendlyARM. Manual MINI2440. [En línea].
http://www.friendlyarm.net/dl.php?file=mini2440_manual_20110421.pdf

- **Sistemas de almacenamiento SDRAM:** La Mini2440 cuenta con dos memorias SDRAM de 32MB cada una, permitiéndole contar con un total de 64MB en memoria, con un bus de datos de 32 bits de ancho, logrando así velocidades de hasta los 100MHz. La selección del espacio nGCS0 por parte del µP, es usada para la SDRAM, la cual se ubica en la dirección de memoria 0x30000000.

Figura 11. Sistema de almacenamiento SDRAM en la Mini2440

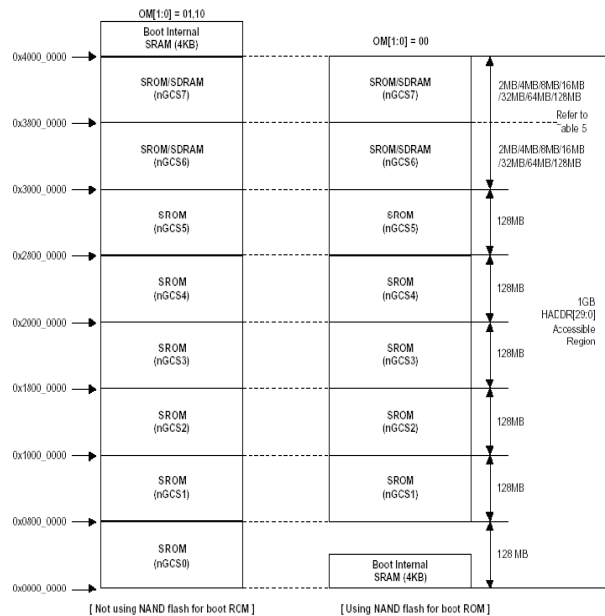


Fuente: FriendlyARM. Manual MINI2440. [En línea].

http://www.friendlyarm.net/dl.php?file=mini2440_manual_20110421.pdf

- **Modos de arranque de la MINI2440:** El μ P S3C2440AL soporta dos modos de arranque: desde la NAND Flash o desde la NOR Flash. El modo de arranque determina la configuración en la disposición de memoria, como se muestra en la Figura 12.

Figura 12. Disposición de la memoria dependiendo del modo de arranque.



Fuente: FriendlyARM. Manual MINI2440. [En línea].

http://www.friendlyarm.net/dl.php?file=mini2440_manual_20110421.pdf

Cuando el switch de selección NOR/NAND se encuentra en la posición NOR, el sistema se ajusta a la disposición de memoria ubicado a la izquierda. En la posición NAND, la disposición de memoria a la derecha representa la configuración de arranque.

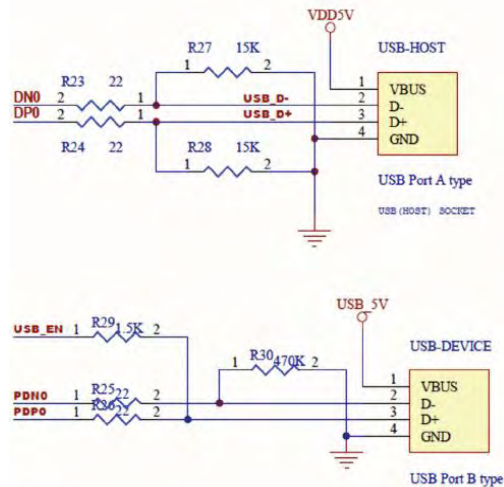
En el modo de arranque desde la NAND Flash, la sección interna BootSram de 4 KB, es designada al espacio nGCS0, en cambio en el modo de arranque desde la NOR Flash nGCS0 está conectado a una memoria externa. La NOR Flash ha sido dispuesta para seleccionar el espacio nGCS0 del μ P. La dirección de espacio para la SDRAM será: 0x30000000 ~ 0x34000000.

- **Interfaz USB:** La Mini2440 cuenta con dos interfaces USB (Figura 13). Un puerto USB, como el que encontramos en una PC convencional, el cual puede ser utilizado para conectar una cámara USB, un teclado, un mouse, una tarjeta de red wifi, etc. Un segundo conector USB esclavo más pequeño (como el que se encuentra en las impresoras), se utiliza generalmente para descargar la imagen del sistema operativo a la tarjeta de destino, cuando la Mini2440 ejecuta un sistema operativo WinCE, es posible instalar ActiveSync en el host, con el fin de sincronizar la placa con este último y así poder tener acceso a los archivos que esta contiene.

Cuando la Mini2440 ejecuta un sistema operativo Linux, no será posible la transferencia de datos a través del puerto USB esclavo, debido a que no existe un controlador “*driver*” que permita hacerlo, en este caso la interfaz ethernet con la cual cuenta la placa será utilizada para dicha tarea.

El puerto USB esclavo puede ser controlado a través del registro GBC5, seteando el bit USB_EN, o poniéndolo a cero, con el fin de deshabilitar el puerto y de esta manera conservar los recursos de la CPU.

Figura 13. Esquema interno de las interfaces USB con las cuales cuenta la Mini2440

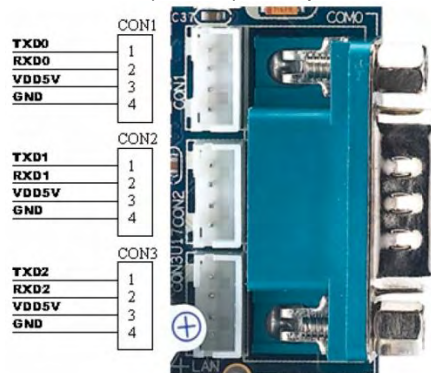


Fuente: FriendlyARM. Manual MINI2440. [En línea].

<http://www.friendlyarm.net/dl.php?file=mini2440_manual_20110421.pdf>

- Puerto serial:** El μ P Samsung S3C2440 soporta 3 puertos seriales UART0, UART1, UART2. Cada uno de los cuales posee un conector físico implementado en la placa Mini2440 (Figura 14) los cuales soportan niveles de señal TTL, las señales de los puertos se encuentran conectadas también al CON4 (Conector GPIO de la placa). El puerto serial UART0, tiene incorporado además un conversor TTL-RS232, ya que el puerto cuenta con una conexión física a un Max232, permitiendo usar un conector DB9 “COM0” para trabajar con cualquier dispositivo RS232. El COM0 trabaja a una velocidad máxima de 115.200 Baudios, y es totalmente soportado en sistemas operativos Linux y WinCE.

Figura 14. Aspecto físico de los 3 puertos seriales (UART0, UART1 y UART2) incorporados en la Mini2440, además se observa, el conector DB9 (COM0), compartido con el puerto UART0



Fuente: FriendlyARM. Manual MINI2440. [En línea].

<http://www.friendlyarm.net/dl.php?file=mini2440_manual_20110421.pdf>

- **Interfaz para el display grafico LCD táctil:** El chip de interfaz para el display LCD, modelo LCD41P, tiene un conector blanco de 41 pines de 0.5 mm de pitch, que contiene algunas señales de control típicas de los LCDs (reloj, activación, campo de escaneo, etc.) y la señal de salida RGB de 8:8:8, que puede soportar un máximo de 16 millones de colores. La iluminación de fondo (backlight) es dada a través de LEDs y se puede apagar y encender (y ser regulada su intensidad) a través de GPB1. Los pines 37, 38, 39 y 40 son utilizados para la interfaz táctil de 4 canales y están conectados directamente a ella. Como todas las interfaces LCD actuales, utiliza una alimentación de 5 Volts.
- **Jumper para la alimentación del LCD:** La placa de desarrollo cuenta únicamente con un jumper J2, el cual es utilizado para seleccionar el voltaje de alimentación para el controlador del LCD mostrado en la Figura 15.

Para pantallas LCDs NEC y Toppoly de 3.5", así como también para las Innolux de 7", se seleccionan voltajes de alimentación de 5v, mediante el jumper J2.

Figura 15. Esquema de chip LCD41P, para la interfaz con el display LCD

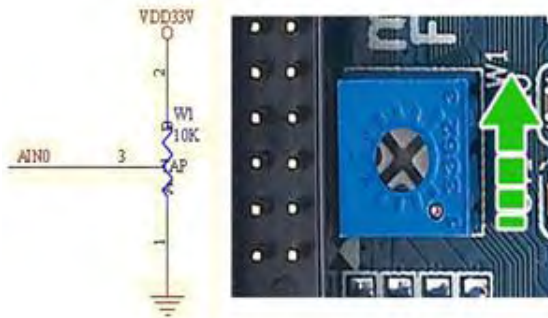


Fuente: FriendlyARM. Manual MINI2440. [En línea].
http://www.friendlyarm.net/dl.php?file=mini2440_manual_20110421.pdf

- **Interfaz GPIO:** El conector CON4 de la MINI2440 tiene una interfaz para GPIO con 34 pines de 2.0 mm de pitch. Además de los pines de GPIO, en CON4 se encuentran otros pines para la CPU, como AIN0-AIN3, CLKOUT, interfaces SPI e I2C, GPB0 Y GPB1.
- **Convertor análogo-digital (ADC):** La MINI2440 cuenta con un convertor ADC, modelo UDA1341TS, de 8 canales y 10 bits de resolución, del tipo

muestreo y retención (*sample and hold*). La frecuencia de muestreo por defecto es de 44.1 KHz. Cuando se está usando el display LCD táctil, 4 canales se utilizan para la conversión de las señales táctiles de entrada. Los otros 4 son de propósito general y son accesibles a través del conector CON4, es decir, el del GPIO. Los pines 5, 6, 7 y 8 de este conector corresponden respectivamente a las señales analógicas de entrada AIN0, AIN1, AIN2 y AIN3. El circuito impreso (PCB) de la placa salida de fábrica tiene AIN0 conectado a un potenciómetro para permitir testear el ADC, como se puede observar en la Figura 16. Es posible utilizar AIN0 como GPIO; retirando el potenciómetro o cortando la pista que se encuentra en la parte de abajo del PCB con bastante espacio libre a su alrededor.

Figura 16. Conexión física del pin AIN0 al potenciómetro de 10K para testear el ADC con el que cuenta la MINI2440.

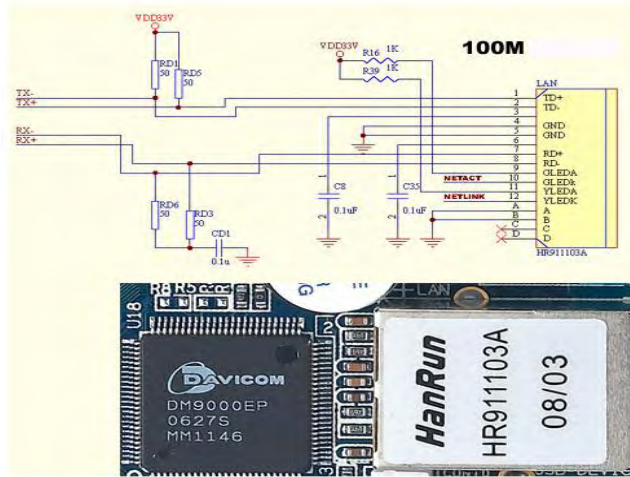


Fuente: FriendlyARM. Manual MINI2440. [En línea].
<http://www.friendlyarm.net/dl.php?file=mini2440_manual_20110421.pdf>

- **Interfaz de red Ethernet:** La placa MINI2440 posee un chip DM9000 10/100M LAN con transformadores de red y un conector RJ-45, tal como se muestra en la Figura 17.

Se puede utilizar un cable Ethernet Cat5E para conectar la placa a un PC, router, switch u otros dispositivos. Las direcciones MAC e IP por defecto son idénticas en todos los dispositivos MINI2440 (IP: 192.168.1.230, MAC: 08:08:11:18:12:27) y pueden ser configuradas por software.

Figura 17. Chip DM9000 10/100M LAN con transformadores de red y un conector RJ-45.



Fuente: FriendlyARM. Manual MINI2440. [En línea].

< http://www.friendlyarm.net/dl.php?file=mini2440_manual_20110421.pdf>

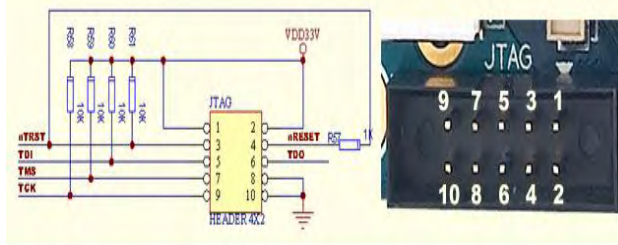
- **Interfaz JTAG:** La memoria interna flash EPROM de la MINI2440 viene en blanco por defecto de fabricación, por lo tanto se hace necesario utilizar una interfaz JTAG para programar la memoria NOR flash con el BIOS (*basic input/output system*) Supervivi, el cual es un tipo de firmware que una vez cargado sobre la placa permitirá utilizar el puerto USB para descargar sistemas aún más complejos. En caso de borrarse el BIOS, será necesario utilizar la interfaz JTAG para arrancar la placa. La interfaz JTAG es utilizada también en tareas de depuración con características como: depuración paso a paso, testeo de memoria, depuración software, etc.

El estándar para la interfaz JTAG utiliza cuatro líneas: TMS, TCK, TDI, TDO, o conocidos también como modo de selección, señal de reloj, datos de entrada, datos de salida, estas señales sumadas a la alimentación y un total de 6 líneas más son suficientes para trabajar con la interfaz, aunque para facilitar la depuración, la mayoría de los simuladores también proporcionan una señal de reset.

Por tanto la interfaz JTAG, incorporara conectores de 20 y 10 pines, la MINI2440 ofrece un completo conjunto de señales para trabajar con la interfaz JTAG, ya que cuenta con un conector de 10 pines destinados para la interfaz (Figura 18).

La interfaz JTAG en la MINI2440 funciona con el Windows H-JTAG, y es utilizada para: cargar la NAND flash de la placa, el borrado del BIOS en la NOR flash, la depuración paso a paso de las instrucciones de un código muy básico, mientras se observa el cambio en algunos de los registros o alguna señal externa.

Figura 18. Interfaz JTAG incorporada en la MINI2440

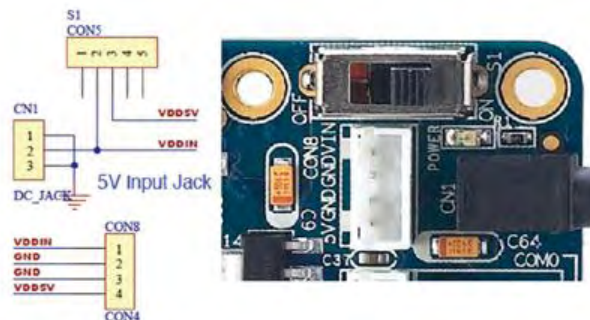


Fuente: FriendlyARM. Manual MINI2440. [En línea].
<http://www.friendlyarm.net/dl.php?file=mini2440_manual_20110421.pdf>

- **Sistema de alimentación:** La MINI2440 requiere una alimentación externa de 5v, suministrados a través del CON8 (Figura 19). Mediante 3 reguladores (Figura 20) son generadas 3 tensiones de alimentación interna de 3.3v 1.8v, y 1.25v, necesarias para el suministro de voltaje a los diferentes chips con los cuales cuenta la placa. La MINI2440 no está diseñada para ser un dispositivo móvil, debido a que no cuenta con un sistema de suministro eléctrico lo suficientemente potente de acuerdo a los requerimientos de los circuitos internos en la placa. El interruptor DIP S1 controla la alimentación del sistema para la placa, lo cual no es posible por software.

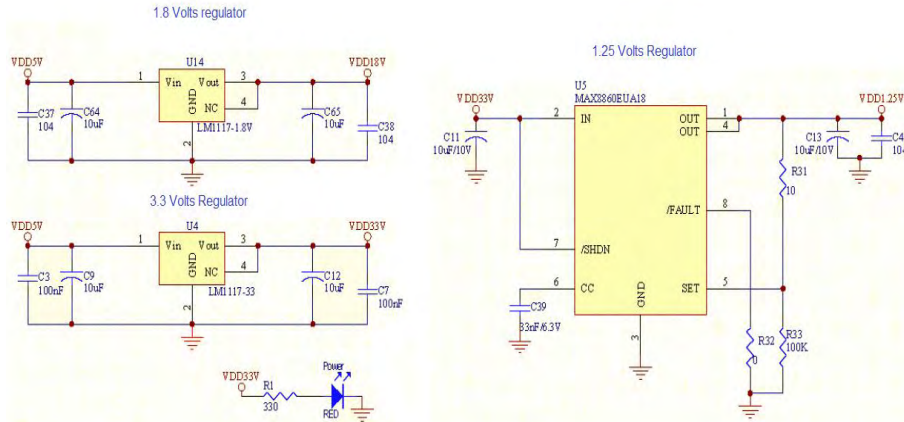
La placa MINI2440, brinda la posibilidad de contar con una fuente de alimentación, la cual puede suministrar 5v a cualquier dispositivo externo (Microcontroladores, Circuitos Integrados, Conversores ADC o DAC, memorias seriales EEPROMs, etc), mediante los conectores de 2 mm de pitch del CON8.

Figura 19. Aspecto físico del CON8, presente en la placa MINI2440



Fuente: FriendlyARM. Manual MINI2440. [En línea].
http://www.friendlyarm.net/dl.php?file=mini2440_manual_20110421.pdf

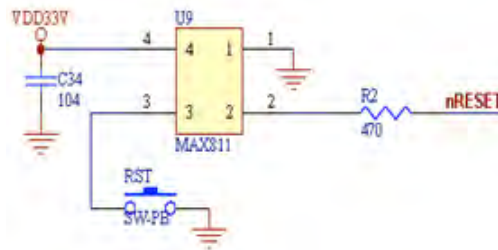
Figura 20. Reguladores de voltaje incorporado en la MINI2440, encargado de suministrar tensiones de 3.3v, 1.8v y 1.25v.



Fuente: FriendlyARM. Manual MINI2440. [En línea].
 < http://www.friendlyarm.net/dl.php?file=mini2440_manual_20110421.pdf>

- **Reinicio del sistema (Reset):** La MINI2440 dispone de un chip de reset modelo MAX811 que se muestra en la Figura 21, que proporciona un reinicio de la CPU limpio y de bajo nivel cuando se pulsa el botón RESET.

Figura 21. Conexión del botón RESET al chip MAX811



Fuente: FriendlyARM. Manual MINI2440. [En línea].
 < http://www.friendlyarm.net/dl.php?file=mini2440_manual_20110421.pdf>

2.3.2 Software en la MINI2440

2.3.2.1 Sistemas operativos

Debido a la complejidad que presenta la configuración hardware de la placa de desarrollo, está diseñada para ser gobernada por un sistema operativo que gestione los procesos básicos del sistema.

La configuración hardware de la placa cuenta con suficiente capacidad de procesamiento y memoria como para albergar y ejecutar un sistema operativo

multiusuario/multitarea con una interfaz gráfica de usuario (GUI, graphic user interface) completa.

Es por ello que la primera tarea fue instalar un sistema operativo previamente compilado para la arquitectura ARM920T de 32 bits con la cual cuenta el Microprocesador Samsung S3C2440AL presente en la placa de desarrollo, además el sistema operativo también incluirá los controladores o drivers de los subsistemas. Posteriormente sobre el sistema operativo instalado se podrá implementar aplicaciones, haciendo uso de las plataformas y librerías disponibles que les puedan dar soporte.

Con respecto al sistema operativo para la FriendlyARM MINI2440, se pudo observar que tiene una gran posibilidad de configuraciones. Se puede elegir varios sistemas operativos, teniendo cada uno sus particularidades y quedando la elección a libertad del desarrollador, entre las posibilidades se puede encontrar Android, Linux (Linux 2.6 + Qtopia, Embedian, Angstrom), Windows CE (5.0, 6.0), uCos (sistema operativo para aplicaciones en tiempo real).

Para las versiones de Android y Linux se puede descargar las imágenes del sistema operativo directamente desde la página web del dispositivo. En cambio, para el sistema operativo Windows CE, se debió descargar un fichero de configuración para adaptarlo al dispositivo, el cual permite elegir múltiples opciones con las cuales se puede crear la imagen final (sistema operativo) que posteriormente correrá sobre la placa de desarrollo.

La razón por la cual se eligió “Windows CE 6.0” como sistema operativo de la placa de desarrollo fue principalmente a que este permite modificar su código de software, ya que está escrito en C#, y este lenguaje es compatible únicamente con Windows.

Para poder crear la imagen del sistema operativo que correrá sobre la placa MINI2440, y de esta manera desarrollar aplicaciones de prueba que paso a paso irán llevando al desarrollo de la aplicación final que es en si el objetivo del proyecto, fue necesario construir un entorno de desarrollo adecuado (Explicado paso a paso en el primer archivo del anexo B). Para ello se llevó a cabo ciertas configuraciones en los dos dispositivos involucrados (host y target), así como también se estableció una comunicación entre ellos mediante protocolos convenientes, que permitieron la transferencia de archivos del uno al otro.

2.3.2.2 Bootloaders o gestores de arranque

La MINI2440, no dispone de un chip BIOS-ROM, por tanto necesita al menos un programa bootloader “*Gestor de arranque*” instalado en la memoria NAND Flash,

que permita inicializar el hardware, proporcionándole los parámetros de arranque (especialmente al controlador de memoria) e “iniciar” por decirlo de alguna manera, el sistema operativo.

Otra función del bootloader es interactuar con el usuario a través de una consola de comandos para transferir nuevas imágenes binarias a la RAM vía Ethernet o puerto serie, copiar imágenes binarias desde la RAM a la Flash, recuperar el sistema operativo, etc. Para ello es necesario tener un bootloader instalado también en la memoria NOR Flash.

Dentro de este aspecto cabe destacar que existe una gran variedad de posibilidades. La MIN2440 trae por defecto en ambas Flash el Bootloader Supervivi, que es específico para ARM, pero existen otros Bootloaders universales como U-boot, RedBoot o uMon, que también pueden usarse en la placa. Por sencillez, se optó por mantener la configuración de fábrica.

2.4 EL OSCILOSCOPIO

En este apartado se definirá que es un osciloscopio, sus funcionalidades, los tipos de osciloscopios que hay en el mercado y cuál de estos se va a diseñar, teniendo en cuenta los requerimientos del equipo y los medios de los cuales se disponía.

2.4.1 ¿Qué es un Osciloscopio¹⁶?

Un osciloscopio es un dispositivo electrónico de medición que representa gráficamente señales eléctricas variables en el tiempo.

Este dispositivo está provisto de una pantalla dividida en 8 divisiones verticales y 10 horizontales. Estas divisiones existen para poder ver los valores de tensión y tiempo de una forma detallada, funcionando así como una especie de voltímetro visual. Los osciloscopios actuales, permiten incluso calcular valores de tensión eficaces, valores medios e incluso desfases entre señales eléctricas.

El eje horizontal, denominado X, representa el tiempo en el que queremos visualizar la señal. Normalmente, todos los osciloscopios poseen un potenciómetro para cambiar la escala de tiempo y tensión. El eje vertical Y, nos indica la amplitud

¹⁶ Cuesta, Cristóbal. Osciloscopio Virtual: Muestreo en tiempo real interpolación. 2003 [En línea].
< <http://rabfis15.uco.es/lvct/tutorial/22/ayuda/introduccion.htm> >

con la que queremos ver la señal. En el caso que se tuviera que analizar una señal estas divisiones permitirán saber su amplitud en tensión y su frecuencia.

Existen dos clasificaciones básicas de osciloscopios: análogos y digitales. Los análogos son aquellos que trabajan con señales continuas de voltaje y los digitales los que trabajan con señales discretas de voltajes.

Un osciloscopio análogo trabaja modificando un haz de electrones que chocan directamente sobre una pantalla, mediante el voltaje inducido. En cambio, un osciloscopio digital debe muestrear la señal a medir usando un conversor análogo-digital (ADC) para, posteriormente, reconstruir la señal en pantalla.

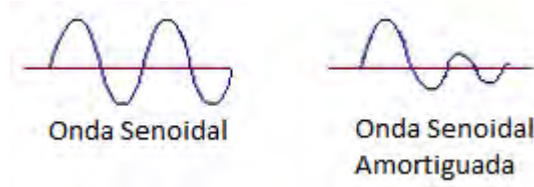
2.4.2 ¿Qué muestra un Osciloscopio?

En este apartado se explicara las distintas señales que se pueden ver en la pantalla de un osciloscopio.

Un osciloscopio permite visualizar todo tipo de señales periódicas o no periódicas.

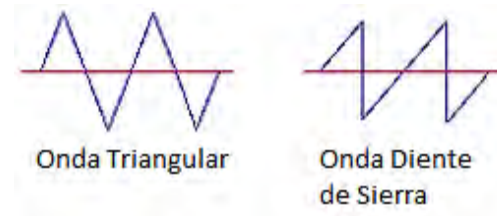
- **Onda senoidal:** Es la onda fundamental de cualquier señal (Figura 22). Es decir, sumando señales senoidales y variándolas tanto en amplitud como frecuencias, se obtiene señales como podría ser una onda cuadrada.

Figura 22. Onda senoidal



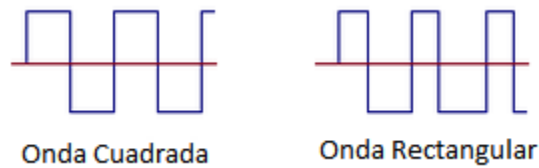
- **Onda triangular:** Corresponde a una señal que aumenta y disminuye constantemente en el tiempo (Figura 23), produciendo un fenómeno que normalmente es conocido como rampa, el cual se utiliza generalmente en el control lineal de voltaje. En el momento en que el tiempo de bajada o subida de la señal disminuye, la señal resultante toma el nombre de diente de sierra.

Figura 23. Onda triangular



- **Onda cuadrada o rectangular:** Este tipo de onda básicamente corresponde a una señal que cambia de un estado a otro de tensión en un periodo de tiempo supremamente pequeño. Este tipo de señales es muy frecuente en el ámbito digital, como por ejemplo, ordenadores, calculadoras, electrodomésticos, e inclusive la televisión con la inclusión de la TDT (Televisión Digital Terrestre).

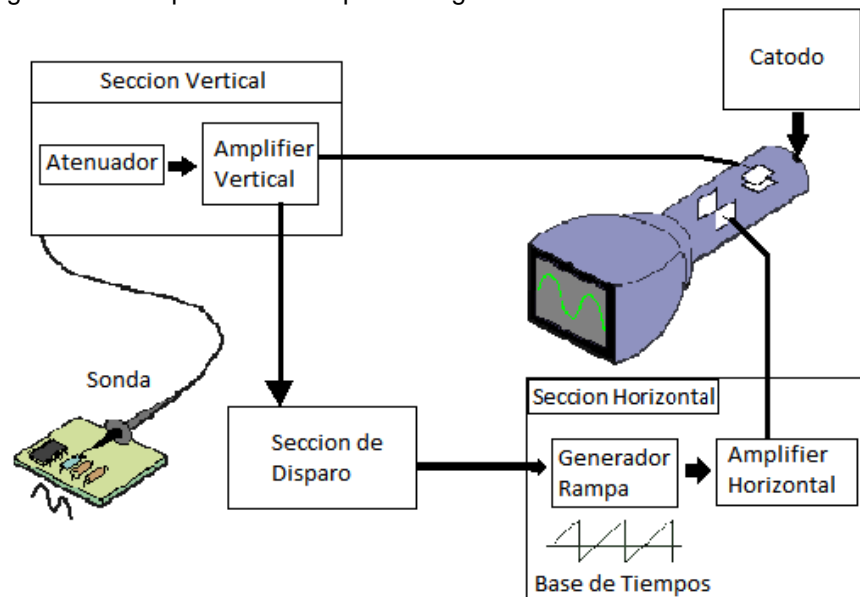
Figura 24. Onda cuadrada o rectangular



2.4.3 Funcionamiento del Osciloscopio

2.4.3.1 Osciloscopio Análogo

Figura 25. Diagrama de bloques Osciloscopio análogo



Fuente: Cuesta, Cristóbal. Osciloscopio Virtual: subsistemas del osciloscopio análogo. [En línea]. < <http://rabfis15.uco.es/lvct/tutorial/22/ayuda/introduccion.htm> >

En esquema de la Figura 25 primero vemos el circuito a analizar y la sonda por donde se introducirá la señal al osciloscopio. Esta señal se dirige hacia un atenuador-amplificador de la sección vertical. Que sea atenuador o amplificador depende de cómo tengamos el mando de control del amplificador, el cual variará

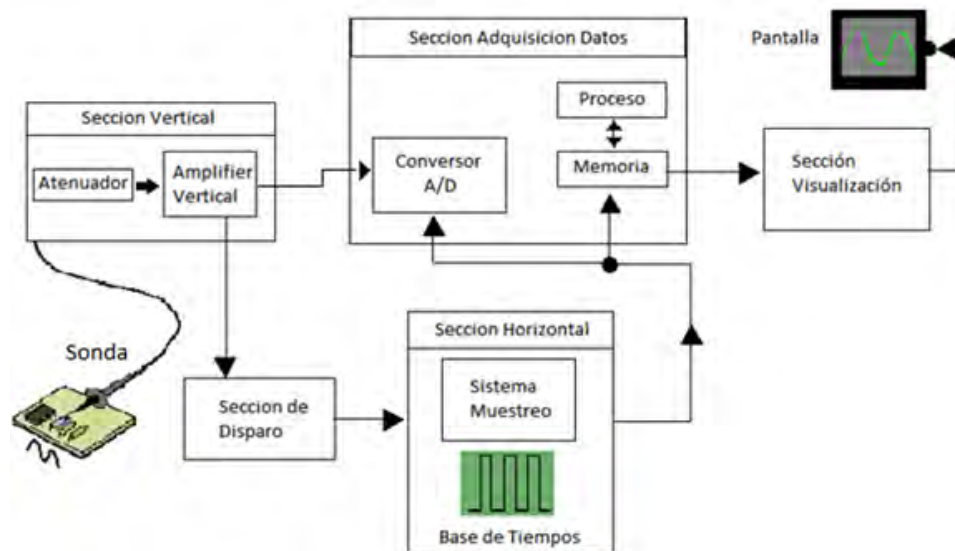
la ganancia del amplificador operacional. A la salida de este bloque, nos dirigimos directamente a las placas de deflexión verticales, que son las encargadas de desviar el haz de electrones que representa la onda. Este haz de electrones viene del cátodo y se ve gracias a una capa fluorescente que hay en la pantalla. Las placas se moverán hacia arriba cuando la señal sea positiva y hacia abajo cuando esta sea negativa.

La señal amplificada o atenuada, también se dirigirá al bloque de disparo, el cual realiza el barrido horizontal. Este se encarga de mover de izquierda a derecha las placas horizontales en un tiempo determinado. Este movimiento de izquierda a derecha se consigue aplicando la rampa ascendente de un diente de sierra a dicha placa de deflexión horizontal. El tiempo de esta rampa viene configurado por el mando regulador del tiempo que queramos por división. El movimiento de derecha a izquierda se consigue con un tiempo mucho menor, ya que es la bajada de este diente de sierra.

Esta es la forma en la que trabaja un osciloscopio análogo. Podemos ver que es bastante sencillo, pero esto también hace que las funciones de este sean limitadas

2.4.3.2 Osciloscopio Digital

Figura 26. Diagrama de bloques Osciloscopio digital



Fuente: Cuesta, Cristóbal. Osciloscopio Virtual: subsistemas del osciloscopio Digital. [En línea].
 < <http://rabfis15.uco.es/lvct/tutorial/22/ayuda/introduccion.htm> >

En el esquema de la Figura 26, se observa que cuando se conecta la sonda a un circuito, la señal se dirige a la sección vertical donde, al igual que en los

osciloscopios análogos, es atenuada o amplificada de dependiendo de donde se sitúe el control vertical VOLTS/DIV.

La sección de disparo es necesaria para estabilizar las señales periódicas. Para lograr la imagen estable, es necesario que el trazado de la onda comience siempre en el mismo punto.

Posterior a la etapa de entrada, encontramos el sistema de adquisición de datos, basado en un ADC, que convertirá la señal analógica en información digital. Con este conversor se muestra la señal a intervalos de tiempo determinados y convierte la señal de tensión continua en una serie de valores digitales llamados muestras que son almacenados en memoria como puntos de señal. El número de los puntos de señal utilizados para reconstruir la señal en pantalla se denomina registro y dependiendo de las prestaciones del modelo de osciloscopio, esta información se puede procesar adicionalmente para realizar varias operaciones matemáticas y gráficas, mostrando el resultado posteriormente en pantalla.

En la sección horizontal una señal de reloj determina el conversor A/D toma las muestras. La velocidad de este reloj se denomina velocidad de muestreo y se mide en muestras por segundo. Por otro lado, mediante el control SEC/DIV es posible ajustar la duración de cada división horizontal de la pantalla.

Finalmente, la sección de visualización recibe los puntos del registro y genera la imagen en la pantalla que típicamente son pantallas de cristal (LCD).

2.4.4 Tipos de Osciloscopio Digital

2.4.4.1 Osciloscopio de memoria digital

En el osciloscopio digital convencional también denominado DSO. Típicamente, depende de una pantalla de barrido en lugar de una de fosforo luminoso típico en el osciloscopio análogo. Los DSO permiten la captura y visualización de eventos que ocurren solamente una vez y a los que se conoce como transitorios. En este tipo de osciloscopios la señal puede ser analizada, archivada e imprimida, debido a que la información de la forma de onda ha sido digitalizada y almacenada. Proporcionan un almacenamiento permanente de la señal. Sin embargo, los DSO no tienen típicamente graduación de intensidad en el tiempo real, por tanto, no puede mostrar niveles de variación de intensidad de una señal “viva”.

Los subsistemas principales que componen un DSO son similares a los de los osciloscopios analógicos. Aunque los DSO no tienen además, otros subsistemas adicionales de procesamiento de datos que se utilizan para recompensar y mostrar los

datos de la forma de onda completa. La arquitectura de procesado de este tipo de osciloscopio es en serie, tal como se muestra en la Figura 27.

Figura 27. Bloques osciloscopio de memoria digital



Fuente: Palacios, Enrique Alonso. Lopez, Luis Bolivar. Diseño e implementación de un osciloscopio de memoria digital de bajo coste y su comunicación con un PC. [En línea].
< <http://upcommons.upc.edu/pfc/bitstream/2099.1/11082/2/Mem%C3%B2ria.pdf>>

La primera etapa de un DSO corresponde a un amplificador operacional, el cual permite ajustar la amplitud y el rango de posición en esta etapa.

A continuación, se encuentra el convertor análogo-digital del sistema, el cual muestrea la señal en puntos aislados en el tiempo y convierte el voltaje de la señal en estos puntos, en valores digitales. El reloj de muestreo determina la velocidad de muestreo expresada en muestras por segundo.

Las muestras del ADC se almacenan en la memoria de adquisición como puntos de la forma de onda. Todos los puntos de la forma de onda en conjunto conforman el registro de forma de onda.

La ruta de la señal del DSO incluye un microprocesador a través del cual pasa la señal medida en su camino hacia la pantalla. Este microprocesador procesa la señal, coordina las actividades de presentación, gestiona los controles del panel frontal y todas las demás funciones que tenga el osciloscopio.

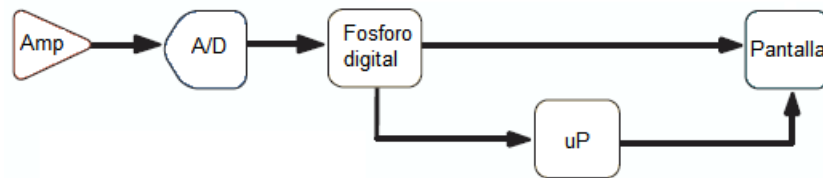
2.4.4.2 Osciloscopio de fósforo digital

Este tipo de osciloscopios (DPO) ofrece un nuevo concepto de arquitectura. Esta arquitectura permite el osciloscopio alcanzar capacidades de adquisición y presentación sin igual para reconstruir una señal con precisión.

Mientras que un DSO utiliza una arquitectura de procesado en serie para capturar, presentar y analizar señales, un DPO utiliza una arquitectura de procesado en paralelo para llevar a cabo estas funciones.

La arquitectura de un DPO se muestra en la Figura 28.

Figura 28. Bloques osciloscopio de fósforo digital



Fuente: Palacios, Enrique Alonso. Lopez, Luis Bolivar. Diseño e implementación de un osciloscopio de memoria digital de bajo coste y su comunicación con un PC. [En línea].
< <http://upcommons.upc.edu/pfc/bitstream/2099.1/11082/2/Mem%C3%B2ria.pdf>>

La etapa de entrada de un DPO es similar a la de un osciloscopio análogo (Amplificador vertical), y su segunda etapa es similar a la de un DSO, con el ADC. Pero la principal diferencia llega a partir de esta conversión.

El DPO explora los datos digitalizados de la forma de onda sobre una base de datos de fósforo digital. La principal característica de este osciloscopio es la gran velocidad que posee, ya que es capaz de realizar una instantánea de la imagen de la señal cada 1/30 segundos y envía al sistema de presentación. Este osciloscopio, respecto a los de otros tipos tiene una mejora del tiempo de actividad del osciloscopio, y un a viva actualización de la presentación. Los detalles de la señal, los eventos intermitentes y las características dinámicas de la señal, se capturan en tiempo real. El microprocesador del DPO trabaja en paralelo con este sistema de adquisición integrado para la gestión de la presentación, la automatización de las medidas y el control del instrumento, para que ello no afecte a la velocidad de adquisición del osciloscopio.

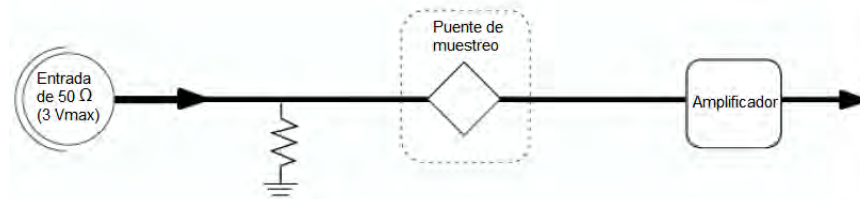
2.4.4.3 Osciloscopio de muestreo

Cuando se están midiendo señales de alta frecuencia, es posible que el osciloscopio no sea capaz de recoger suficientes muestras en un solo barrido. Un osciloscopio de muestreo es una herramienta ideal para la captura precisa de señales cuyas componentes de frecuencia son mucho más elevadas que la velocidad de muestreo del osciloscopio.

Este osciloscopio es capaz de medir señales de hasta el orden de magnitud más rápida que cualquier otro osciloscopio. Puede alcanzar anchos de banda y tiempos de alta velocidad diez veces más elevados que otros osciloscopios para señales repetitivas. Se dispone de osciloscopios de muestreo secuencial en tiempo equivalente con anchos de banda de hasta 50 GHz.

La arquitectura de este tipo de osciloscopio se muestra en la Figura 29.

Figura 29. Bloques osciloscopio de muestreo



Fuente: Palacios, Enrique Alonso. Lopez, Luis Bolivar. Diseño e implementación de un osciloscopio de memoria digital de bajo coste y su comunicación con un PC. [En línea].
< <http://upcommons.upc.edu/pfc/bitstream/2099.1/11082/2/Mem%C3%B2ria.pdf>>

En contraste con las arquitecturas de memoria digital y de los osciloscopios de fósforo digital, la arquitectura del osciloscopio de muestreo invierte la posición del atenuador/amplificador y del puente de muestreo. El muestreo se realiza antes de la atenuación o amplificación de la señal. Este permite tener un ancho de banda muy elevado, pero a la vez, limita mucho el rango dinámico de muestreo del osciloscopio. Al no haber ningún atenuador ni amplificador en la entrada desaparece la función para escalar la entrada. Por tanto, el rango dinámico de este tipo de osciloscopios queda limitado a 1Vpp, bastante más pequeña que el de los otros tipos, que pueden manejar de 50 a 100 Volts.

Después de realizar el estudio de todos los tipos de osciloscopios disponibles en el mercado, finalmente, se ha decidido diseñar un osciloscopio de memoria digital. Las razones por las cuales se tomó esta decisión saltan a la vista, ya que el osciloscopio de fosforo digital precisa de una etapa análoga muy potente para poder realizar muchas muestras rápidamente para, de esta forma, ir resaltando los puntos que se quiere dibujar en la pantalla. Esta etapa análoga se sale del presupuesto sabiendo que se quiere montar un osciloscopio de bajo coste; y el osciloscopio de muestreo es realmente mucho más rápido que el DSO, pero no podríamos observar toda la señal y el rango de tensiones quedaría muy limitado. Por tanto, el tipo más completo, que tiene un ancho de banda más que aceptable y un rango de tensiones correcto es el osciloscopio de memoria digital.

2.4.5 Factores que influyen en la calidad de un Osciloscopio

2.4.5.1 Ancho de banda

El ancho de banda determina la capacidad básica de un osciloscopio para medir una señal. Conforme aumenta la frecuencia de la señal, disminuye la capacidad del osciloscopio para presentar la señal con exactitud.

Esta especificación indica el rango de frecuencias que el osciloscopio puede medir con precisión.

El ancho de banda de un osciloscopio se define como la frecuencia a la cual una señal sinusoidal se presenta atenuada a un 70,7 % respecto a la amplitud real de la señal. Este punto se conoce como el punto a -3 dB, termino basado en una escala logarítmica.

Sin un ancho de banda adecuado, el osciloscopio no podrá resolver los cambios de alta frecuencia, distorsionando la amplitud y características de la señal.

2.4.5.2 Precisión de la ganancia

La precisión de la ganancia indica la exactitud con que el sistema vertical atenúa o amplifica una señal, habitualmente se indicada como un porcentaje de error.

2.4.5.3 Relación señal a ruido (SNR)

La relación señal a ruido es una razón de la señal deseada a indeseado de la señal (ruido) en el promedio de nivel de potencia de una transmisión.

Si hay demasiado ruido en un circuito, la relación señal/ruido es baja. Si el circuito es de buena calidad, la relación señal/ruido será alto.

2.4.5.4 Errores de cuantizacion

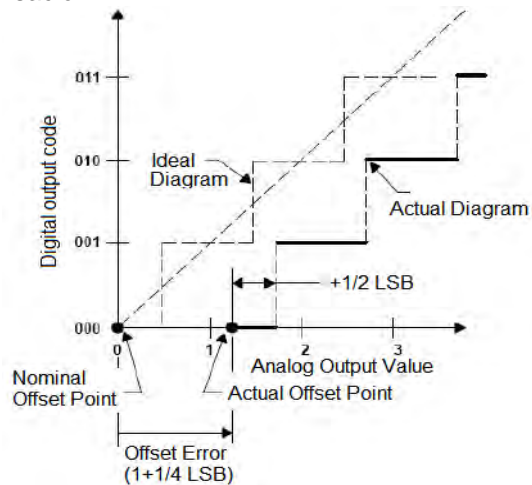
Estos errores son parte de la etapa de conversión de la señal análoga en una representación digital de la misma. A continuación se presenta los diferentes tipos de errores que encontramos en este proceso¹⁷.

- **Error de compensación:** Está definido como la diferencia entre la compensación nominal y la que realmente se tiene. Para el caso de un conversor análogo-digital, el punto de compensación nominal es el valor que se tiene en la entrada para obtener una salida igual a cero según muestra la Figura 30.

¹⁷ Steven W. Smith. "The Scientist and Engineer's Guide to Digital Signal Processing". En: DSP. 2^o edition – California Technical. [En línea] <www.dspguide.com>

Cuando en la entrada se coloca este supuesto valor y en su salida se obtiene un valor diferente de cero es, justamente, cuando existe un error de compensación. Esto se puede corregir ajustando los valores de compensación, y si no es posible ajustarlos, entonces se puede realizar en la etapa de análisis de los datos obtenidos, haciendo los ajustes necesarios en estos valores.

Figura 30. Error de compensación

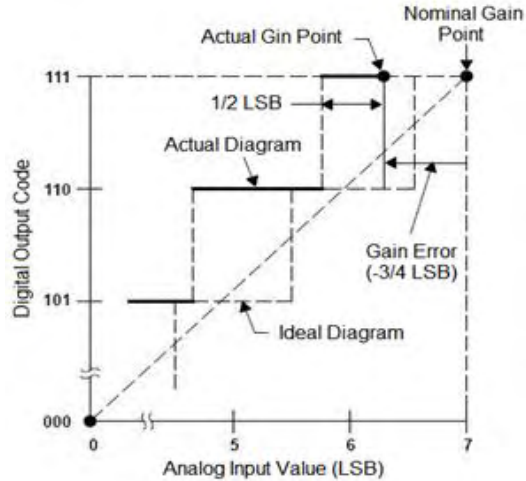


Fuente: Hoffman, pablo. OscUsb: Investigación y análisis. [En línea].
 < <http://pablohoffman.com/cgi-bin/twiki/bin/view/Oscusb/DocCap02Investigacion> >

- **Error de ganancia:** Se define como la diferencia entre el valor de ganancia nominal (ganancia esperada, calculada) y la ganancia real. Este valor se calcula cuando el error por compensación es nulo, caso contrario se obtendrá un error en un cálculo generado por otro error.

La Figura 31 muestra de forma simple el significado de este error. Cuando se tiene un valor conocido a la entrada del convertidor, se espera un valor de salida determinado. Cuando la diferencia entre el valor esperado y el obtenido es constante e igual sin importar cuál sea el valor en su entrada, entonces se trata de un error de ganancia. Este error es lineal y constante en todo el rango posible de entrada. Este error es posible solucionarlo, aunque también se puede corregir en etapas posteriores.

Figura 31. Error de ganancia



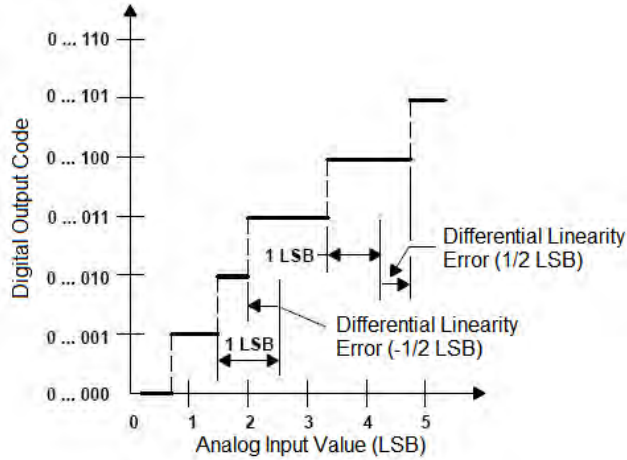
Fuente: Hoffman, pablo. OscUsb: Investigación y análisis. [En línea].
< <http://pablohoffman.com/cgi-bin/twiki/bin/view/Oscusb/DocCap02Investigacion> >

- **Error de apertura:** Es un tipo de error generado en los propios componentes y no es corregible. La incertidumbre en la señal de entrada en el momento en que se muestra y retiene dicha señal (sample and hold) está dado desde el instante en que se muestrea hasta que se retiene la señal, antes de pasar al proceso de conversión. El error de apertura puede ser generado por ruido o también o también por variaciones en la señal de reloj. Este error también influye en las características finales del sistema.

Este error puede ser reducido si el tiempo de muestreo y retención disminuye. Si este valor es lo más pequeño posible, entonces se tiene mayor certeza de que la conversión realizada es correcta.

- **Error de no-linealidad diferencial:** Es la diferencia entre la variación de tensión nominal que genera un cambio en un bit a la salida del convertidor de 1 LBT (Least Significant Byte) y la variación real que debe ocurrir. Es decir, que si el cambio de tensión que debe ocurrir a la entrada para que haya un cambio en la salida de un bit es extremadamente el esperado, entonces el error DNL es cero. En cambio, si para que haya un cambio en la salida, el diferencial de tensión a la entrada debe ser mayor (o menor) al que se especifica, entonces existe este tipo de error. Incluso este tipo de error puede generar que existan valores binarios de salida que nunca se lleguen a dar, debido a que el rango de entrada tiene como respuesta una menor cantidad de valores debido al error citado (cuando el diferencial de tensión de entrada es mayor que el especificado para 1 LSB).

Figura 32. Error de no-linealidad diferencial



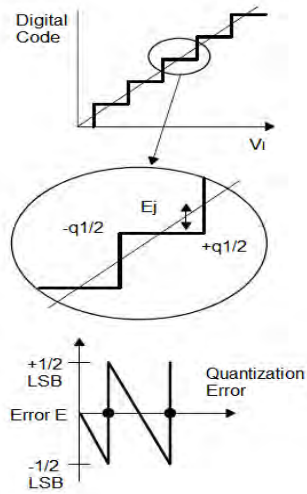
Fuente: Hoffman, pablo. OscUsb: Investigación y análisis. [En línea].
< <http://pablohoffman.com/cgi-bin/twiki/bin/view/Oscusb/DocCap02Investigacion> >

- **Error de Cuantización:** Esta basado en la naturaleza de las señales análogas y digitales. Una señal análoga es continua y puede tener infinitos valores (por más que la señal este acotada), mientras que una señal digital es discreta, tiene una cantidad finita de valores posibles. Aquí entonces radica este error. Al intentar convertir una señal que puede tener infinitos valores en otra que solo puede tener valores finitos, está claro que se pierde información. La cantidad de valores posibles (o estados posibles) que puede tener la señal digital está relacionada con la cantidad de bits con la cual esta se representa.

Sin embargo la cantidad de bits es una cantidad finita. La única forma de hacer que este error sea nulo, es haciendo que la cantidad de bits con la que se representa el valor digital sea infinito. Esto implica que cierto valor digital representa a muchos (de hecho, infinitos) valores análogos posibles. De esta forma, un valor análogo produce una salida digital, y esa salida digital si se vuelve a convertir a un valor análogo, puede que no corresponda con el valor original. Cierta información se ha perdido en este proceso, y esto se conoce como error de cuantización.

En el caso de un conversor ideal, donde la salida se puede entenderse como una escalera perfecta, el error entre la señal real de entrada y su correspondencia con la salida digital, tiene una función de densidad de probabilidad uniforme en el caso que se asume que la entrada es totalmente aleatoria. Este error varía en el rango de $\pm 1/2$ LSB, o bien, $\pm q/2$, donde q es el ancho de un escalón, tal como se muestra en la Figura 33.

Figura 33. Error de cuantización



Fuente: Hoffman, pablo. OscUsb: Investigación y análisis. [En línea].
< <http://pablohoffman.com/cgi-bin/twiki/bin/view/Oscusb/DocCap02Investigacion> >

3. DISEÑO Y DESARROLLO DEL DISPOSITIVO

El dispositivo desarrollado permite visualizar (sobre una grilla de 10x8 divisiones), señales eléctricas, variables en el tiempo, de manera que sobre el eje vertical denominado Y es representado el voltaje de la señal de entrada, mientras que el tiempo de la señal se representa sobre el eje horizontal X. Las señales se visualizan sobre una pantalla LCD táctil de 3.5", a través de la cual es posible llevar a cabo un innovador e intuitivo control sobre los parámetros de medida del dispositivo, utilizando los diferentes elementos (menús, botones y barras) incorporados en la GUI desplegada en la pantalla del equipo.

El osciloscopio permite ser conectado a través del puerto USB con cualquier laptop o PC de mesa que corra un sistema operativo Windows, trabajando en este caso como una tarjeta de adquisición de datos para el equipo previamente conectado, el cual ejecutara la aplicación desarrollada de igual forma que se ejecuta sobre el sistema embebido MINI2440, aunque en este caso se trata de una aplicación de escritorio para Windows. Esto fue posible gracias al Entorno de Desarrollo Integrado para Windows (IDE, por sus siglas en inglés) conocido con el nombre de Microsoft Visual Studio 2005 utilizado en el desarrollo de la aplicación.

Este IDE en particular permite entre otras crear aplicaciones tanto para dispositivos móviles como también aplicaciones de escritorio para cualquier entorno que soporte la plataforma .NET de Microsoft.

La interfaz gráfica de usuario desplegada en la pantalla táctil del osciloscopio, dispone de 2 barras de control, que permiten seleccionar el factor de escala de medición en el dispositivo, tanto en tiempo (10 divisiones en total TIME/DIV) como en amplitud (8 divisiones en total VOLTS/DIV), cuenta además con 3 botones (CH1, CH2, CH1y2) que permiten seleccionar uno de los 2 canales de entrada o los 2 simultáneamente si la medición así lo requiere. La interfaz incorpora también 4 controles adicionales denominados U "up", D "down" (Y POS), L "left" y D "down" (X POS), que permitirán respectivamente ajustar la posición vertical del canal de entrada y desplazar un eje paralelo al eje vertical de origen, con el fin de lograr una medición más exacta en cuanto al periodo de la señal de entrada.

El dispositivo se energiza a través de un adaptador de voltaje AC/DC, que se encarga de suministrar 16V a la fuente de alimentación dual incorporada en el dispositivo, la cual se encarga de transformarlos en un voltaje de +/-8V DC, para el suministro de los diferentes componentes en el dispositivo. Cabe resaltar también que el diseño de la fuente dual incorpora un regulador de voltaje a 5V, debido a que la alimentación de algunos componentes así lo requiere.

El desarrollo del dispositivo se ha logrado mediante la finalización satisfactoria de las diferentes fases definidas en el cronograma de actividades. Para obtener un control más eficiente del cumplimiento de los objetivos, el proyecto se desarrolló de forma modular, fragmentando el diseño e implementación general del dispositivo en tres partes fundamentales; el diseño y construcción de la placa de adquisición de datos, que en conjunto con el sistema embebido MINI2440 conforman el Hardware del dispositivo, el desarrollo del Firmware, programa que es ejecutado internamente por el microcontrolador para implementar la lógica de captura a través del control de los diferentes bloques incorporados en la placa de adquisición de datos, y finalmente el Software del dispositivo, que corresponde al desarrollo de la aplicación (interfaz gráfica de usuario) que ejecuta el sistema embebido MINI2440, la cual permite controlar cada uno de los diferentes parámetros y funcionalidades del equipo.

3.1 HARDWARE

El siguiente subcapítulo describe el diseño y desarrollo de la placa de adquisición de datos incorporada en el dispositivo, la cual se acopla al sistema embebido MINI2440 para conformar así, el hardware del equipo.

3.1.1 Selección de la arquitectura

La arquitectura implementada consiste en un microcontrolador central y varios componentes separados, los cuales en conjunto conforman la placa de adquisición de datos del dispositivo. De esta manera el microcontrolador se encargara de implementar la lógica de captura, a través del control tanto en la sección vertical como horizontal del dispositivo.

El microcontrolador PIC18F2553, cuenta con un conversor ADC interno, cuyas características de funcionamiento fueron suficientes de acuerdo a los requerimientos en cuanto al ancho de banda que debía presentar el dispositivo, el microcontrolador permite además controlar la sección vertical de la placa de adquisición de datos, a través de los pines físicos E/S de propósito general con los cuales cuenta externamente.

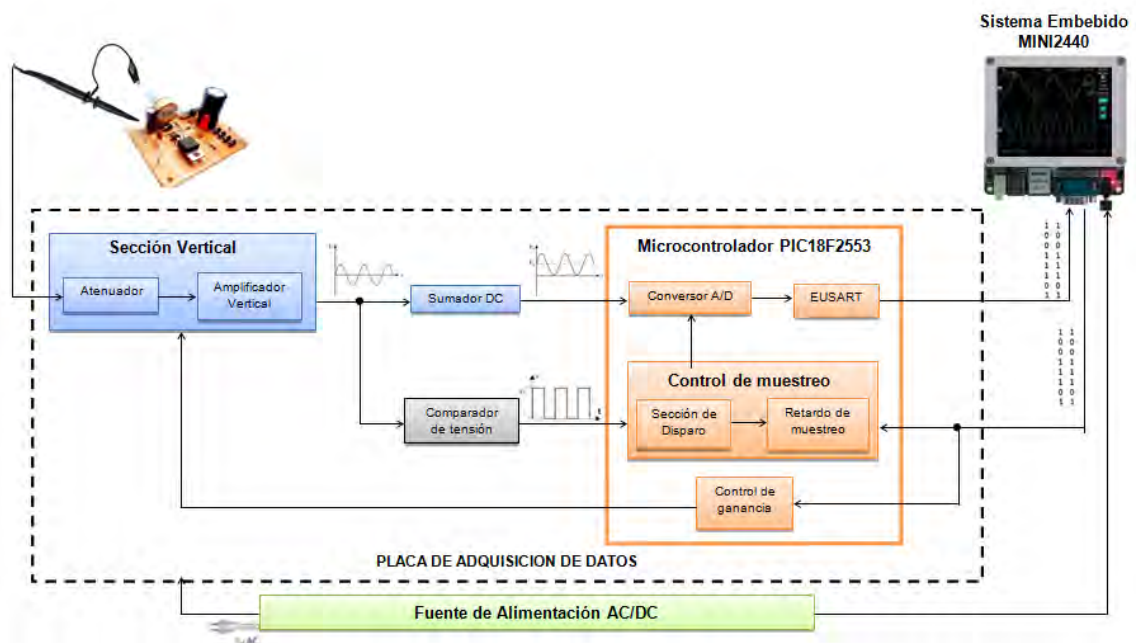
3.1.2 Funcionamiento y diagrama de bloques

De acuerdo a la arquitectura seleccionada, la placa de adquisición de datos del dispositivo incorpora los siguientes elementos:

- 1 Microcontrolador central, encargado de implementar la lógica de captura, a través del control sobre los bloques de la sección vertical y horizontal del dispositivo, por medio de los multiplexores y la base de tiempos del sistema de muestreo del convertor A/D interno en él.
- 2 Multiplexores, encargados de seleccionar las diferentes ganancias para cada uno de los canales de entrada, de acuerdo al factor de escala seleccionado. Estos multiplexores hacen parte de los amplificadores de ganancia programable implementados en la sección vertical del dispositivo.
- Divisores de tensión y amplificadores operacionales en diferentes configuraciones con el fin de adaptar, atenuar y amplificar las señales provenientes de cada uno de los canales de entrada.

En la Figura 34 se identifican los bloques principales de la tarjeta de adquisición de datos del dispositivo, la cual en conjunto con el sistema embebido MINI2440 conforman el hardware del equipo.

Figura 34. Diagrama de bloques del Osciloscopio Digital implementado



3.1.3 El Microcontrolador PIC18F2553

Teniendo en cuenta que el sistema embebido MINI2440 debía realizar absoluto control sobre los diferentes parámetros de medición del dispositivo, a través de una interfaz gráfica de usuario (software); se decidió incorporar un

microcontrolador cuya función consiste en enlazar el software (GUI, desplegada en la pantalla del dispositivo) y el hardware (etapa de adecuación y adquisición de las señales de voltaje) del equipo; llevando a cabo las siguientes tareas: conversión Análoga/Digital de las señales previamente adecuadas, control de las diferentes ganancias para cada uno de los canales, control de los retardos en el muestreo, de acuerdo al factor de escala horizontal seleccionado en pantalla y transmisión digital de los datos (correspondientes a las señales análogas de voltaje), hacia el sistema embebido MINI2440, fue por ello que microcontrolador seleccionado debía cumplir como mínimo con los siguientes requisitos:

- Alta velocidad en la frecuencia de operación.
- Incorporar un módulo de comunicación digital con periféricos.
- Incorporar un módulo de Captura/Comparación.
- ADC interno de mínimo 10 Bits.
- Memoria de programa Flash de mínimo 16KB.
- Un puerto GPIO con al menos 6 líneas físicas.

A partir de los requerimientos expuestos, se llevó a cabo una búsqueda en la Web y se encontró que el Microcontrolador PIC18F2553 de la Compañía Microchip era el más apropiado para la aplicación, al incorporar en un pequeño encapsulado DIP28 de alto desempeño una memoria Flash mejorada con un conversor A/D interno de 12 Bits además de trabajar con tecnología nanoWatt, la cual implementa diferentes funciones que permiten controlar los componentes de energía, incluidos los de consumo de potencia estática, activa y media.

En la Tabla 4 se presentan algunas de las características con las cuales cuenta el Microcontrolador PIC18F2553.

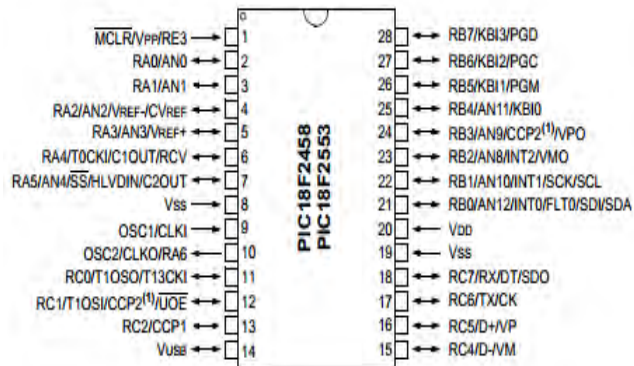
Tabla 4. Características Microcontrolador PIC18F2553

CARACTERISTICAS	PIC 18F2553
Frecuencia de operación	DC – 48 MHz
Memoria de programa (Bytes)	24576
Memoria de programa (Instrucciones)	12288
Memoria de datos (Bytes)	2048
Memoria de datos EEPROM (Bytes)	256
Fuentes de interrupción	19
Puertos de E/S	Puertos A,B,C
Temporizadores	4
Módulos de captura/comparación/PWM	2
Comunicaciones serie	MSSP,USART mejorada
Módulo USB	1
Módulo ADC de 12 Bits	10 canales de entrada
Comparadores	2

Resets (y retrasos)	POR, BOR, WDT, Instrucción RESET, desbordamiento de la STACK, MCLR, PWRT, OST
Programación por detección de alto/bajo voltaje	Si
Set de instrucciones	75 Instrucciones, 83 habilitando el set de instrucciones extendido

Como se puede observar, las prestaciones proporcionadas por el Microcontrolador seleccionado son más que suficientes de acuerdo a los requerimientos planteados con anterioridad, ya que entre otras presenta una frecuencia de operación muy elevada de 48 MHz, lo que permite una gran velocidad en la ejecución del programa (Firmware), también cuenta con una memoria de programa de 32KB, la cual es más que suficiente para las instrucciones que se va a utilizar en el desarrollo del programa principal que contendrá el microcontrolador, aunque la característica más importante quizás consiste en que incorpora un módulo ADC de 12 Bits de resolución, factor que influye directamente en una mayor resolución vertical, en cuanto a la caracterización del prototipo final.

Figura 35. Pin-out del microcontrolador PIC18F2553



Fuente: Microchip. Hoja de especificaciones PIC18F2553. p. 2 [en línea].
<http://pdf1.alldatasheet.com/datasheet-pdf/view/197486/MICROCHIP/PIC18F2553.html>

Cabe resaltar también el juego de 75 instrucciones y el sistema de memoria lineal con los que cuentan la familia de los PIC 18F, factores que proporcionan una mayor facilidad en el desarrollo del programa principal y comodidad en el momento de manejar buffers lineales.

3.1.3.1 Módulo ADC del Microcontrolador

El Conversor A/D interno del microcontrolador, cuenta con 10 entradas controladas a través de un multiplexor, este módulo permite convertir una señal análoga de voltaje en un correspondiente número digital de 12 bits.

El conversor ADC interno del microcontrolador cuenta con 5 registros:

- Registro alto del resultado de la conversión A/D (ADRESH)
- Registro bajo del resultado de la conversión A/D (ADRESL)
- Registro de control 0 de la conversión A/D (ADCON0)
- Registro de control 1 de la conversión A/D (ADCON1)
- Registro de control 2 de la conversión A/D (ADCON2)

El registro ADCON0 controla la operación del módulo A/D, el ADCON1 configura las funciones de los pines físicos conectados al conversor y el registro ADCON2 permite configurar la fuente de reloj para el conversor A/D, establecer el tiempo de adquisición del mismo y configurar la justificación del resultado de la conversión.

Tabla 5. Registro ADCON0 (FC2h): Registro 0 DE Control A/D

U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
-	-	CHS3	CHS2	CHS1	CHS0	GO-DONE	ADON
Bit 7						bit 0	

Tabla 6. CHS3:CHS0: Bits de selección del canal análogo

Bits de Configuración	Canal Análogo
0000	Canal 0 (AN0)
0001	Canal 1 (AN1)
0010	Canal 2 (AN2)
0011	Canal 3 (AN3)
0100	Canal 4 (AN4)
0101	No implementado
0110	No implementado
0111	No implementado
1000	Canal 8 (AN8)
1001	Canal 9 (AN9)
1010	Canal 10 (AN10)
1011	Canal 11 (AN11)
1100	Canal 12 (AN12)
1101	No implementado
1110	No implementado
1111	No implementado

Tabla 7. GO/DONE: Bit de inicio y monitorización del estado de la Conversión A/D

0	Proceso de Conversión A/D ha finalizado
1	Proceso de Conversión A/D esta en progreso

Tabla 8. ADON: Bit de habilitación del Conversor A/D

0	Conversor A/D deshabilitado
1	Conversor A/D Habilitado

Tabla 9. Registro ADCON1 (FC1h): Registro 1 de Control A/D

U-0	U-0	R/W-0	R/W-0	R/W-0	R/W ⁽¹⁾	R/W ⁽¹⁾	R/W ⁽¹⁾
-	-	VCFG1	VCFG0	PCFG3	PCFG2	PCFG1	PCFG0
Bit 7							bit 0

Tabla 10. VCFG1: Bit de configuración de la tensión de referencia VREF-

0	VREF+ se conecta a Vss
1	VREF+ se conecta a la línea física RA2

Tabla 11. VCFG0: Bit de configuración de la tensión de referencia VREF+

0	VREF+ se conecta a VDD
1	VREF+ se conecta a la línea física RA3

Tabla 12. PCFG3:PCFG0: Bits de configuración de los puertos de conversión A/D. Mediante estos bits se establecen las líneas físicas (RA0, RA1, RA2, RA3, RA5, RB4, RB3, RB2, RB1, RB0) que van a trabajar como entradas al ADC interno del microcontrolador.

PCFG3	AN12	AN11	AN10	AN9	AN8	AN4	AN3	AN2	AN1	AN0
0000	A	A	A	A	A	A	A	A	A	A
0001	A	A	A	A	A	A	A	A	A	A
0010	A	A	A	A	A	A	A	A	A	A
0011	D	A	A	A	A	A	A	A	A	A
0100	D	D	A	A	A	A	A	A	A	A
0101	D	D	D	A	A	A	A	A	A	A
0110	D	D	D	D	A	A	A	A	A	A
0111	D	D	D	D	D	A	A	A	A	A
1000	D	D	D	D	D	A	A	A	A	A
1001	D	D	D	D	D	A	A	A	A	A
1010	D	D	D	D	D	A	A	A	A	A
1011	D	D	D	D	D	D	A	A	A	A
1100	D	D	D	D	D	D	D	A	A	A
1101	D	D	D	D	D	D	D	D	A	A
1110	D	D	D	D	D	D	D	D	D	A
1111	D	D	D	D	D	D	D	D	D	D

Tabla 13. Registro ADCON2 (FC0h): Registro 2 de control A/D

R/W-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
ADFM	-	ACQT2	ACQT1	ACQT0	ADCS2	ADCS1	ADCS0
Bit 7							bit 0

Tabla 14. ADFM: Bit de justificación del resultado de la conversión.

0	El resultado de la conversión se almacena con justificación a la derecha
1	El resultado de la conversión se almacena con justificación a la izquierda

Tabla 15. ACQT2:ACQT0: Bits de configuración del tiempo de adquisición para la Conversión

000	0 TAD (Se agregara un tiempo equivalente a un ciclo de instrucción, antes de iniciar la conversión)
001	2 TAD
010	4 TAD
011	6 TAD
100	8 TAD
101	12 TAD
110	16 TAD
111	20 TAD

Tabla 16. ADCS2:ADCS0: Bits de configuración de la señal de reloj para el Conversor A/D

000	FOSC/2
001	FOSC/8
010	FOSC/32
011	FRC (Señal de Reloj tomada del oscilador RC)
100	FOSC/4
101	FOSC/16
110	FOSC/64
111	FRC (Señal de Reloj tomada del oscilador RC)

La fuente de tensión análoga de referencia del conversor es seleccionada por software y fue configurada de tal forma que corresponda a la alimentación del microcontrolador (VDD y VSS), de esta manera VREF+ = 5V y VREF- = 0V, y por tanto la resolución en el módulo Conversor A/D estará dada por:

$$Resolucion\ ADC = \frac{(VREF+) - (VREF-)}{2^n - 1} \quad Ecuación\ 1$$

$$\text{Resolución ADC} = \frac{5V - 0V}{2^{12} - 1}$$

$$\text{Resolución ADC} = \frac{5V}{4.095}$$

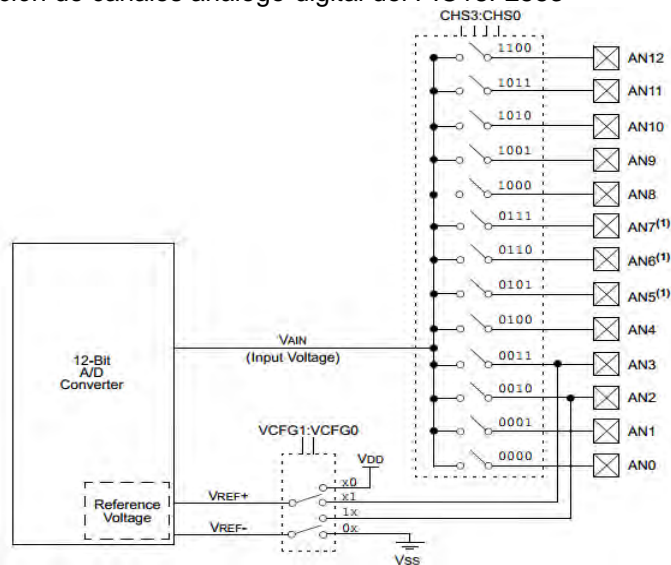
$$\text{Resolución ADC} = 1,221mV$$

Este parámetro indica la mínima variación de tensión en la señal de entrada que el conversor podrá detectar y como se mencionó anteriormente, influye directamente en la resolución vertical del prototipo final desarrollado.

El conversor toma la señal procedente del bloque de muestreo y retención y utilizando el método de aproximaciones sucesivas genera un resultado binario equivalente a la señal análoga de entrada.

Los pines de los puertos asociados al módulo conversor A/D del microcontrolador pueden ser configurados como entradas análogas o como E/S digitales. Los registros ADRESH y ADRESL contienen el resultado de la conversión; cuando la conversión A/D ha sido completada, el resultado se carga en esta pareja de registros, y automáticamente el bit GO/DONE del registro ADCON0 se pone en 0, indicando que el proceso de conversión a finalizado. En la Figura 36, se muestra el diagrama de bloques del módulo A/D interno del microcontrolador.

Figura 36. Multiplexación de canales análogo-digital del PIC18F2553



Note 1: Channels AN5 through AN7 are not available on 28-pin devices.

Fuente: Microchip. Hoja de especificaciones PIC18F2553. p. 22 [en línea].
<http://pdf1.alldatasheet.com/datasheet-pdf/view/197486/MICROCHIP/PIC18F2553.html>

Una vez configurado adecuadamente el módulo A/D, el canal seleccionado lleva a cabo el proceso de adquisición, antes de comenzar la conversión como tal, este

tiempo de adquisición, como se verá más adelante, puede ser programado por software para que ocurra automáticamente entre la activación del bit GO/DONE y el comienzo real de la conversión.

Con el fin de que el módulo conversor A/D trabaje adecuadamente, fue necesario llevar a cabo los siguientes pasos:

1. Configurar el módulo A/D:

- Configurar los pines análogos de entrada, la fuente del voltaje de referencia y las E/S digitales (ADCON1).
- Seleccionar el canal A/D de entrada (ADCON0).
- Seleccionar el tiempo de adquisición y el reloj de conversión A/D (ADCON2).
- Activar el módulo A/D (ADCON0).

2. Adicionar un pequeño retardo de al menos 2 μ s, requerido entre la activación del módulo A/D y el inicio de la conversión.

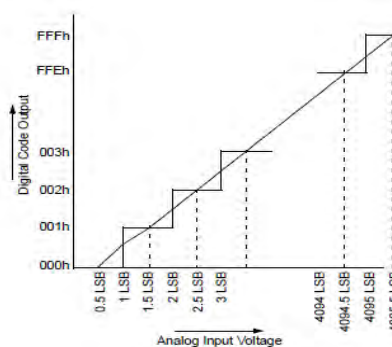
3. Iniciar la conversión:

- Poner en alto el bit GO/DONE del registro ADCON0
- Esperar que el proceso de conversión A/D finalice, a través de un continuo testeo del bit GO/DONE, el cual se pondrá automáticamente en bajo al finalizar el proceso.

4. Finalmente se lee el resultado de la conversión A/D que ha sido cargado en la pareja de registros ADRESH:ADRESL

En la Figura 37 se muestra la función de transferencia del conversor A/D interno del microcontrolador.

Figura 37. Función de transferencia del conversor análogo-digital del PIC18F2553



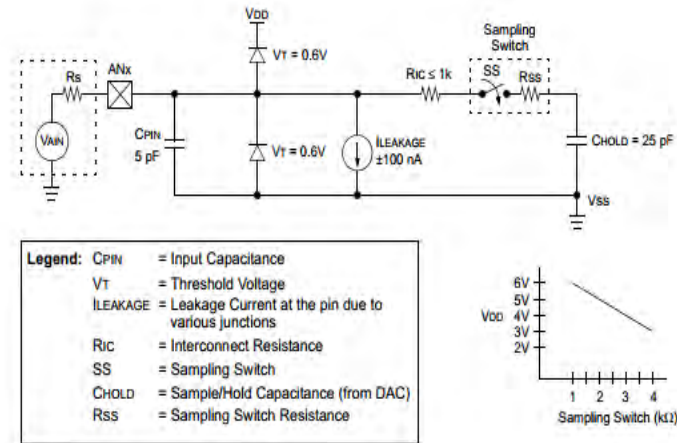
Fuente: Microchip. Hoja de especificaciones PIC18F2553. p. 23 [en línea].

<http://pdf1.alldatasheet.com/datasheet-pdf/view/197486/MICROCHIP/PIC18F2553.html>

1. Requerimientos de adquisición en el Módulo A/D

Con el fin de lograr mayor precisión en la medida por parte del conversor, la carga de retención del capacitor (CHOLD) debe permitir una completa carga del nivel de voltaje presente en el canal de entrada. El modelo de la entrada análoga, se muestra en la Figura 38.

Figura 38. Modelo del conversor análogo-digital del PIC18F2553



Fuente: Microchip. Hoja de especificaciones PIC18F2553. p. 23 [en línea].

<<http://pdf1.alldatasheet.com/datasheet-pdf/view/197486/MICROCHIP/PIC18F2553.html>>

La impedancia de la fuente (R_s) al igual que la impedancia del switch interno de muestreo (R_{ss}), afectan directamente el tiempo requerido para la carga del capacitor CHOLD. La impedancia del switch interno de muestreo, varía el voltaje de alimentación (VDD) en el dispositivo. Mientras que la impedancia de la fuente afecta el voltaje de offset a la señal análoga de entrada (debido a la corriente de fuga en el pin). La máxima impedancia recomendada para las fuentes de entrada análogas es de 2.5kΩ. Una vez seleccionado el canal análogo de entrada, este debe ser muestreado un tiempo mínimo de adquisición antes de comenzar la conversión (una vez iniciada la conversión, el capacitor de carga CHOLD es desconectado del pin de entrada).

Con el fin de establecer el tiempo mínimo de adquisición, se debe tener en cuenta la siguiente ecuación:

$$TACQ = TAMP + TC + TCOFF$$

Ecuación 2

*TACQ = Tiempo de establecimiento del amplificador
+ Tiempo de carga del capacitor
+ Tiempo de coeficiente de temperatura*

En la cual, de acuerdo a la Tabla de datos del microcontrolador se establece que:

$$TAMP = 0,2us$$

$$TC = -(CHOLD)(RIC + RSS + RS) \ln\left(\frac{1}{4096}\right)us \quad \text{Ecuación 3}$$

$$TCOFF = (Temp - 25^{\circ}C) \left(\frac{0,02us}{^{\circ}C}\right) \quad \text{Ecuación 4}$$

Esta ecuación también asume un error de $\frac{1}{2}$ LSb (4096 pasos para el A/D de 12 bits). El $\frac{1}{2}$ LSb error es el error máximo permitido por el conversor A/D para conseguir su resolución específica.

Con el fin de calcular el tiempo mínimo de adquisición, TACQ. Se asume los siguientes datos de aplicación:

CHOLD = 25Pf
 RS = 2.5k Ω
 Error de conversión = $\frac{1}{2}$ LSb
 VDD = 3V \rightarrow RSS = 4k Ω
 Temperatura = 60 $^{\circ}$ C (máxima)

De esta manera:

$$TACQ = TAMP + TC + TCOFF$$

$$TAMP = 0,2us$$

$$TCOFF = (Temp - 25^{\circ}C) \left(\frac{0,02us}{^{\circ}C}\right)$$

$$TCOFF = (60^{\circ}C - 25^{\circ}C) \left(\frac{0,02us}{^{\circ}C}\right)$$

$$TCOFF = 0,7us$$

El tiempo de coeficiente de temperatura es requerido únicamente para temperaturas mayores a 25 $^{\circ}$ C, por encima de esta magnitud TCOFF = 0us.

$$TC = -(CHOLD)(RIC + RSS + RS) \ln\left(\frac{1}{4096}\right)us$$

$$TC = -(25pF)(1k\Omega + 4k\Omega + 2.5k\Omega) \ln\left(\frac{1}{4096}\right)us$$

$$TC = -(25pF)(1k\Omega + 4k\Omega + 2.5k\Omega) \ln(0.0002441)us$$

$$TC = 1.56\mu s$$

Finalmente, se tiene que el tiempo mínimo de adquisición para el conversor A/D del microcontrolador será:

$$TACQ = 0.2\mu s + 1.56\mu s + 0.7\mu s$$

$$TACQ = 2.46\mu s$$

2. Selección y configuración del tiempo de adquisición

A través del registro de control ADCON2, es posible seleccionar el tiempo de adquisición que ocurrirá automáticamente, cada vez que el bit GO/DONE es puesto en alto.

El tiempo de adquisición puede ser establecido a través de los bits ACQT2:ACQT0 (ADCON2, bits 5:3), los cuales proporcionan un rango de 2 a 20 TAD. Cuando el bit GO/DONE es puesto en alto, el módulo A/D procede a muestrear la señal de entrada durante el tiempo de adquisición previamente establecido, luego del cual se da inicio automáticamente al proceso de conversión.

Una vez finalizada la conversión, el bit GO/DONE se limpia automáticamente, y el conversor A/D comienza nuevamente a muestrear el canal seleccionado.

3. Selección del reloj para el Módulo A/D

El tiempo de conversión por cada bit se define como TAD, por lo tanto el conversor A/D del microcontrolador requiere 13 TAD para la conversión de 12 bits. La fuente de reloj del conversor se selecciona por software, y existen 7 posibles opciones para establecer el TAD:

- 2 TOSC
- 4 TOSC
- 8 TOSC
- 16 TOSC
- 32 TOSC
- 64 TOSC
- Oscilador interno RC

Para una correcta conversión A/D, el reloj de conversión A/D (TAD) debe ser tan pequeño como sea posible, pero al mismo tiempo tan grande como el mínimo TAD

especificado en la hoja de datos del microcontrolador (0.8us). Con el fin de dar una mayor explicación con respecto a la correcta configuración del parámetro TAD, se presenta la Tabla 17, la cual muestra los diferentes valores de TAD de acuerdo a la frecuencia de operación del microcontrolador y a la fuente de reloj para el conversor A/D seleccionada.

Tabla 17. TAD vs Frecuencia de operación del dispositivo

Fuente de reloj para el conversor A/D (TAD)		Se asume un TAD Min = 0.8us
Operación	ADCS2:ADCS0	Máxima FOSC
2 TOSC	000	2.50 MHz
4TOSC	100	5.00 MHz
8 TOSC	001	10.00 MHz
16 TOSC	101	20.00 MHz
32 TOSC	010	40.00 MHz
64 TOSC	110	48.00 MHz
RC	x11	1.00 MHz

Particularmente en el desarrollo del proyecto, la frecuencia de operación del microcontrolador fue configurada en 48MHz, lo cual establece de acuerdo a la anterior Tabla un valor de TAD igual a 64 TOSC, es decir:

$$TAD = 64TOSC \quad \text{Ecuación 5}$$

$$TAD = \frac{64}{FOSC} \quad \text{Ecuación 6}$$

$$TAD = 64/(48 * 10^6 \text{Hz})$$

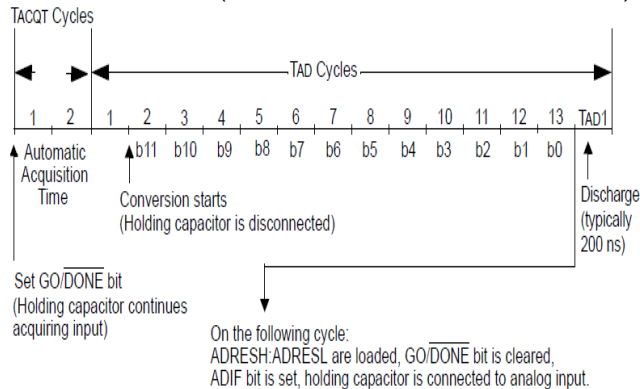
$$TAD = 1.33us$$

De esta manera se configuro el tiempo de adquisición en 2TAD, estableciendo así un TACQ de 2.66us, el cual estará por encima del tiempo mínimo requerido por el módulo para trabajar correctamente.

4. Proceso de conversión del Módulo A/D

En la Figura 39, se muestra la operación del conversor A/D de acuerdo a la configuración establecida.

Figura 39. Operación del conversor A/D (ACQT <2:0>=001, TACQ = 2TAD)



Fuente: Microchip. Hoja de especificaciones PIC18F2553. p. 24 [en línea].
 <<http://pdf1.alldatasheet.com/datasheet-pdf/view/197486/MICROCHIP/PIC18F2553.html>>

Una vez el bit GO/DONE (ADCON0, bit 1), es puesto en alto y se ha establecido un TACQ igual a 2TAD (ACQT2:ACQT0 = 001), el proceso de adquisición empieza y transcurrirá un tiempo igual a 2TAD antes de que la conversión inicie.

Si por alguna razón el bit GO/DONE es limpiado durante la conversión, el proceso será abortado inmediatamente y el par de registros ADRESH:ADRESL no será actualizado, manteniendo el resultado de la última conversión completada.

Después de que el proceso de conversión ha sido completado o abortado, un tiempo denominado TCY es requerido antes de continuar con la siguiente adquisición. Una vez transcurrido el tiempo mencionado, automáticamente empezara una nueva adquisición en el canal seleccionado.

Se adiciono además un tiempo de retardo de 2us entre la habilitación del módulo conversor (poner en alto el bit ADON del registro ADCON0), y el inicio de la conversión (poner en alto el bit GO/DONE del registro ADCON0), de acuerdo a los requerimientos de funcionamiento establecidos en la hoja de datos del Microcontrolador.

3.1.3.2 Módulo de comunicación serial EUSART

El módulo de Transmisión y Recepción Universal Síncrono Asíncrono Mejorado (EUSART) que incorpora el Microcontrolador PIC18F2553, conocido también con el nombre de Interfaz de Comunicación Serial o SCI, permite ser configurado como un sistema asíncrono de comunicación full-duplex, (comunicando al microcontrolador con dispositivos periféricos tales como terminales de computadora o computadoras personales), o como un sistema síncrono de comunicación half-duplex (comunicando al microcontrolador con dispositivos como

convertidores A/D o D/A, circuitos integrados o memorias EEPROM con comunicación serial).

El módulo EUSART incorpora características adicionales, entre las cuales se encuentran: detección automática de velocidad de comunicación (baudrate), Auto-activación por detección de dato recibido y también permite la transmisión y detección de carácter de BREAK (bus LIN).

De esta manera, el módulo EUSART permitirá ser configurado en los siguientes modos:

- Asíncrono (full-duplex) con:
 - Detección automática de velocidad de comunicación (baudrate).
 - Auto-activación por detección de dato recibido.
 - Transmisión y detección de carácter de BREAK de 12 bits.
- Síncrono en modo Master (half-duplex) con selección en la polaridad del reloj.
- Síncrono en modo Esclavo (half-duplex) con selección en la polaridad del reloj.

Cabe mencionar también que la operación y el funcionamiento del módulo es controlado a través de los siguientes registros:

- Estado y Control de Transmisión (TXSTA).
- Estado y Control de Recepción (RCSTA).
- Control de la Rata de Baudios (BAUDCON).

Teniendo en cuenta que el sistema embebido MINI2440, incorpora 3 puertos seriales (UART0, UART1, UART2) totalmente soportados en sistemas operativos Linux y WinCE, que manejan además niveles de señal TTL y permiten trabajar a una velocidad máxima de 115.200 Baudios, se decidió configurar la EUSART interna del Microcontrolador PIC18F2553 en modo asíncrono (full-duplex), estableciendo de este modo una comunicación serial y asíncrona entre el PIC y el sistema embebido MINI2440. Para ello fue necesario llevar a cabo ciertas configuraciones en cada uno de los registros involucrados, con el fin de que el módulo EUSART del microcontrolador trabaje en el modo previamente establecido.

A continuación se describe la manera en que trabaja el módulo EUSART interno del microcontrolador, teniendo en cuenta la configuración para cada uno de los registros involucrados en el proceso, con el fin de establecer una comunicación serial asíncrona con el sistema embebido MINI2440.

En el modo de comunicación serial asíncrona (full-duplex), la EUSART usa un formato estándar NRZ asíncrono, el cual utiliza para la sincronización 1 bit de inicio, 8 o 9 bits de datos y 1 bit de paro. Mientras no se estén transmitiendo datos la EUSART envía continuamente un bit de marca en alto.

Tabla 18. Registro TXSTA (FACH): Registro de estado y control de transmisión

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R-1	R/W-0
CSRC	TX9	TXEN ⁽¹⁾	SYNC	SENDB	BRGH	TRMT	TX9D
Bit 7						bit 0	

Tabla 19. CSRC: Bit de selección de la fuente de señal de reloj para el modo síncrono

0	Modo Esclavo (señal de reloj externa)
1	Modo Maestro (señal de reloj generada internamente)

Tabla 20. TX9: Bit de habilitación del modo de 9 bits en transmisión

0	Se habilita el modo de 8 bits en transmisión
1	Se habilita el modo de 9 bits en transmisión

Tabla 21. TXEN: Bit de habilitación de la transmisión

0	Transmisión deshabilitada
1	Transmisión habilitada

Tabla 22. SYNC: Bit de selección del modo de trabajo de la EUSART

0	Selección del modo asíncrono
1	Selección del modo síncrono

Tabla 23. SENDB: Bit de envío del carácter de BREAK en modo asíncrono

0	Envío del carácter de transmisión BREAK completado
1	Se enviará un carácter BREAK en la próxima transmisión (se pone a '0' por hardware cuando finalizo el envío)

Tabla 24. BRGH: Bit de selección del modo de alta velocidad en modo asíncrono

0	Se selecciona baja velocidad de comunicación
1	Se selecciona alta velocidad de comunicación

Tabla 25. TRMT: Bit de estado del registro de desplazamiento de transmisión

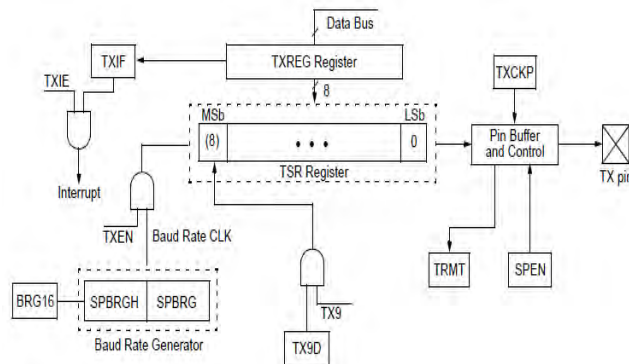
0	Registro de desplazamiento de transmisión ocupado
1	Registro de desplazamiento de transmisión vacío
TX9D: Valor del 9º Bit en transmisión (puede utilizarse como bit de paridad o para distinguir entre dirección o dato en los buses maestro-esclavo)	

El modo asíncrono se seleccionó limpiando el bit SYNC del anterior registro, además se habilito el modo de 8 bits de transmisión limpiando el bit TX9, y se seleccionó el modo de alta velocidad de comunicación poniendo en alto el bit BRGH del registro TXSTA (esto con el fin de lograr una mayor precisión en el momento de realizar el cálculo para la velocidad de comunicación del módulo EUSART).

Cabe resaltar que el bit TRMT de este registro va a ser fundamental en el desarrollo del Firmware que ejecutara el Microcontrolador, ya que este será testeado continuamente entre la transmisión de dato y dato, con el fin de no sobrescribir el registro de desplazamiento de transmisión de la EUSART y de este modo evitar la transmisión errónea de datos.

En la Figura 40 se muestra el diagrama de bloques del transmisor de la EUSART. El corazón de este módulo es el registro de corrimiento (transmit shift register, TSR).

Figura 40. Diagrama de bloques del transmisor de la EUSART



Fuente: Microchip. Hoja de especificaciones PIC18F2553. p. 26 [en línea].
<http://pdf1.alldatasheet.com/datasheet-pdf/view/197486/MICROCHIP/PIC18F2553.html>

Para transmitir un dato, el programa principal del microcontrolador escribe el dato en el registro TXREG, en cuanto el TSR termina de enviar el dato anterior (incluyendo el bit de parada, de haber sido configurado así previamente) lee el dato actual contenido en TXREG (si lo hay) esto ocurre en un ciclo TCY, en cuanto el dato de TXREG es transferido al TSR, el TXREG queda vacío, condición que es indicada a través del bit bandera TXIF (PIR1, bit 4), el cual se pone automáticamente en alto.

Existe otro bit, ya mencionado anteriormente llamado TRMT (TXSTA, bit 1), el cual muestra el estado del TSR, este bit se pone automáticamente en alto cuando TSR está vacío, y cambia su estado en el momento que TSR se encuentra transmitiendo un dato. Mientras que el bit TXIF puede generar una interrupción, el bit TRMT no lo puede hacer, ya que está diseñado para ser testeado en aplicaciones que requieran una transmisión continua de datos a través de la

EUSART. Cabe resaltar que para habilitar el módulo de transmisión es necesario poner en alto el bit TXEN (TXSTA, bit 5), de lo contrario, el pin de transmisión (RC6/TX/CK) se mantendrá en alta impedancia.

Por otra parte, cabe mencionar que el funcionamiento del receptor serial asíncrono que implementa el módulo EUSART del microcontrolador, estará gobernado de acuerdo a la configuración en el registro de estado y control de recepción RCSTA (registro ubicado en la dirección FABh de la memoria de datos interna del microcontrolador), descrito a continuación.

Tabla 26. Registro RCSTA (FABh): Registro de estado y control de recepción

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R-0	R-0	R-x
SPEN	RX9	SREN	CREN	ADDEN	FERR	OERR	TX9D
Bit 7						bit 0	

Tabla 27. SPEN: Bit de activación del puerto serie

0	Puerto serie desactivado
1	Puerto serie activado (Se configuran las líneas RC6/TX/CK y RC7/RX/DT como líneas del puerto serie)

Tabla 28. RX9: Bit de habilitación del modo de 9 bits en transmisión

0	Se habilita el modo de 8 bits en recepción
1	Se habilita el modo de 9 bits en recepción

Tabla 29. SREN: Bit de habilitación de la recepción simple en modo síncrono Maestro

0	Se deshabilita la recepción simple
1	Se habilita la recepción simple

Tabla 30. CREN: Bit de habilitación de la recepción

0	Recepción deshabilitada
1	Recepción habilitada

Tabla 31. ADDEN: Bit de habilitación de la dirección para el modo asíncrono de 9 bits

0	Detección de dirección deshabilitada (todos los bytes recibidos en el registro de desplazamiento de recepción son procesados independientemente del valor del 9° bit recibido)
1	Detección de dirección habilitada (solo se procesa un byte recibido en el registro de desplazamiento de recepción si el 9° bit está a '1')

Tabla 32. FERR: Bit de error de formato (solo lectura)

0	No se ha producido error de formato
1	Se ha producido error de formato

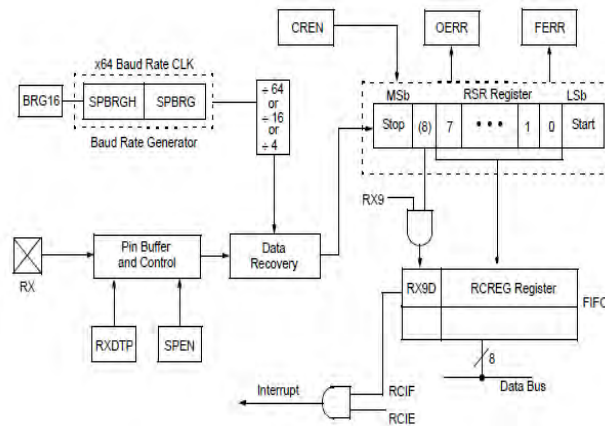
Tabla 33. OERR: Bit de error de sobrescritura (solo lectura)

0	No se ha producido error de sobrescritura
1	Se ha producido error de sobrescritura
RX9D: Valor del 9° Bit recibido	

En cuanto a la configuración del registro RCSTA, se habilito tanto el puerto serial, como la recepción continua de datos, poniendo en alto los bits SPEN y SREN respectivamente, además se estableció el modo de 8 bits de recepción, poniendo en bajo el bit RX9, ya que en la comunicación serial asíncrona entre el microcontrolador y el sistema embebido no se tendrá en cuenta el estado del 9° bit para el posterior procesamiento de los datos. El registro RCSTA cuenta además con dos bits de señalización muy importantes (FERR, OERR), los cuales indican que se ha producido en la recepción un error de formato o de sobrescritura, muy útiles en el desarrollo del firmware, ya que serán testeados constantemente a medida que el registro de recepción va recibiendo un nuevo dato.

En la Figura 41 se muestra el diagrama de bloques del receptor de la EUSART. El bloque principal de este módulo corresponde al registro de corrimiento (receive shift register, RSR).

Figura 41. Diagrama de bloques del receptor de la EUSART



Fuente: Microchip. Hoja de especificaciones PIC18F2553. p. 27 [en línea].
<http://pdf1.alldatasheet.com/datasheet-pdf/view/197486/MICROCHIP/PIC18F2553.html>

Los datos entran en serie por el pin RC7/RX y son muestreados por el bloque de lectura de bits (el cual trabaja a una frecuencia de 16 veces el baudrate). Si el bit de habilitación de recepción CREN está en alto, el bloque de lectura de bits identifica los bits recibidos (Start Bit, Bits de datos, 9° bit y Stop bit) y los va transmitiendo de manera serial al RSR. Una vez se han recibido todos los bits correspondientes a un byte el valor recibido se transmite en paralelo desde el RSR al registro RCREG. De esta manera el registro RSR queda preparado para recibir un nuevo dato.

Si por alguna circunstancia, durante el proceso de recepción se produce un error de formato (el valor del Stop bit el '0' en lugar de '1') o de sobreescritura (se recibe un nuevo byte antes de que el registro RSR se haya volcado en el RCREG) se pondrá en alto el bit correspondiente (FERR/OERR).

Cada vez que es recibido un nuevo dato, el bit RCIF (PIR1, bit 5) se pone en alto, y de este modo si el bit de habilitación de la interrupción por recepción en el módulo EUSART está activado, se generará una interrupción. El bit RCIF es únicamente de lectura y se pone automáticamente en alto en el momento que un nuevo dato es recibido en el registro RCREG, el cual al ser leído permitirá que el bit RCIF retorne a su estado inicial bajo. Si el modo de 9 bits ha sido activado (RCSTA, bit 0) el valor de 9° bit recibido quedará almacenado en el bit RX9D.

Como se mencionó anteriormente, la configuración del módulo EUSART contempla 3 registros; Estado y Control de Transmisión (TXSTA), Estado y Control de Recepción (RCSTA) y un último e importante registro conocido como Control de los Baudios o BAUDCON (registro ubicado en la dirección FB8h de la memoria de datos interna del microcontrolador), el cual permite la configuración de los diferentes parámetros en cuanto a la velocidad y el formato de los datos recibidos y transmitidos a través de la EUSART.

Tabla 34. Registro BAUDCON (FB8h): Registro de estado y control de los baudios

R/W-0	R-1	R/W-0	R/W-0	R/W-0	U-0	R/W-0	R/W-0
ABDOVF	RCIDL	RXDTP	TXCKP	BRG16	-	WUE	ABDEN
Bit 7							bit 0

Tabla 35. ABDOVF: Bit de desbordamiento de auto-detección de la velocidad de comunicación

0	No se ha producido desbordamiento durante la auto-detección
1	Se ha producido desbordamiento durante la auto-detección (debe ponerse a '0' por software)

Tabla 36. RCIDL: Bit de estado de la operación de recepción

0	Hay una operación de recepción en marcha
1	No hay ninguna operación de recepción en marcha

Tabla 37. SCKP: Bit de selección de polaridad de la señal de reloj en modo síncrono

0	El dato está disponible en el nivel bajo de la señal de reloj (después del flanco de bajada)
1	El dato está disponible en el nivel alto de la señal de reloj (después del flanco de subida)

Tabla 38. BRG16: Bit de habilitación del generador de velocidad de 16 bits (SPBRG y SPBRGH)

0	Generador de velocidad de comunicación de 8 bits (solo SPBRG)
1	Generador de velocidad de comunicación de 16 bits

	(SPBRG y SPBRGH)
--	------------------

Tabla 39. WUE: Bit de habilitación del modo de auto-activación en modo asíncrono

0	Modo de auto-activación deshabilitado
1	Modo de auto-activación habilitado (cuando se detecta un flanco de bajada en la línea RX se pone en alto el flag RCIF)

Tabla 40. ABDEN: Bit de habilitación del modo de auto-detección de velocidad de comunicación

0	Modo de auto-detección deshabilitado
1	Modo de auto-detección habilitado en la siguiente recepción

La única configuración, llevada a cabo con este registro consistió en poner en alto el bit BRG16, habilitando de esta manera el generador de velocidad de comunicación de 16 bits, lo cual permitirá en conjunto con la activación del bit BRGH (TXSTA, bit 2), establecer de una manera más exacta la magnitud en la tasa de baudios de la comunicación, manteniendo de este modo sincronizado al Microcontrolador con el sistema embebido MINI2440.

El Microcontrolador PIC 18F2553 dispone de un generador de velocidad de comunicación, conocido también como BRG (Baud Rate Generator), el cual consiste en un temporizador cuya función se controla mediante un registro único de escritura. Este registro puede ser de 8 (SPBRG) o 16 bits ([SPBRGH:SPBRG]).

El generador BRG, cuenta además con dos velocidades, las cuales son seleccionadas mediante el bit BRGH (TXSTA, bit 2). En la siguiente Tabla se establecen las fórmulas que se deben tener en cuenta en el momento de calcular la velocidad de comunicación para el modo de comunicación asíncrona en función de los valores de los bits BRGH y BRG16 y de los registros SPBRGH y SPBRG.

Tabla 41. Calculo de velocidades de transmisión

BRG16	BRGH	CALCULO DE LA VELOCIDAD DE COMUNICACIÓN
0	0	$\frac{F_{osc}}{64 * (SPBRG + 1)}$
0	1	$\frac{F_{osc}}{16 * (SPBRG + 1)}$
1	0	$\frac{F_{osc}}{16 * ([SPBRGH: SPBRG] + 1)}$
1	1	$\frac{F_{osc}}{4 * ([SPBRGH: SPBRG] + 1)}$

Teniendo en cuenta que la frecuencia de operación del microcontrolador (Fosc) se estableció en 48MHz y que además los bits BRGH y BRG16 se configuraron en un nivel lógico alto, se debió calcular el valor correspondiente de los registros

SPBRGH (FB0h) y SPBRG (FAFh), sin olvidar también que la velocidad de comunicación entre el microcontrolador y el sistema embebido MINI2440 será de 115.200 baudios, al ser la velocidad máxima a la cual trabajan los puertos seriales del sistema embebido MINI2440, de este modo se procedió a realizar el cálculo, a partir de la relación presentada en la última fila de la Tabla anterior.

$$115.200 = \frac{F_{osc}}{4 * ([SPBRGH:SPBRG] + 1)} \quad \text{Ecuación 7}$$

$$[SPBRGH:SPBRG] = \frac{48 * 10^6}{(4 * 115.200)} - 1 \quad \text{Ecuación 8}$$

$$[SPBRGH:SPBRG] = 103,1667 = \mathbf{103}$$

Por tanto, SPBRGH=0x00 y SPBRG = 0x67, de esta forma el valor real de la velocidad de comunicación corresponde a:

$$\text{Velocidad de comunicacion} = \frac{F_{osc}}{4 * ([SPBRGH:SPBRG] + 1)} \quad \text{Ecuación 9}$$

$$\text{Velocidad de comunicacion} = \frac{48 * 10^6}{4 * (103 + 1)} = \mathbf{115.384,6154}$$

Puede observarse, que la velocidad de comunicación no es del todo exacta, ya que presenta un pequeño porcentaje de error dado por:

$$\text{Error} = \frac{\text{Velocidad en baudios calculada} - \text{Velocidad en baudios deseada}}{\text{Velocidad en baudios deseada}} \quad \text{Ecuación 10}$$

$$\text{Error} = \frac{115.384,6154 - 115.200}{115.200} = 0,16\%$$

Este pequeño porcentaje de error, no influye en la comunicación del microcontrolador con el sistema embebido MINI2440, debido a que la velocidad en el puerto serial de este último, fue configurada exactamente a 115.384 baudios, gracias a las herramientas de programación que incorpora el entorno de desarrollo integrado de Visual Studio 2005, bajo el cual fue llevado a cabo el desarrollo de la aplicación para el sistema embebido MINI2440.

3.1.4 Etapa de entrada y acondicionamiento de la señal

Este apartado describe el diseño y funcionamiento de la etapa previa al microcontrolador, cuya función consiste en adaptar la señal de entrada de acuerdo a los requerimientos de funcionamiento del PIC y a las escalas de medida

seleccionadas a través de los diferentes controles presentes en la GUI del dispositivo.

La etapa esta conformada por 1 sección principal y 2 bloques secundarios.

- **Sección vertical.** Encargada de acoplar, atenuar y amplificar la señal de entrada de acuerdo al factor de escala vertical seleccionado a través del control VOLTS/DIV incorporado en la GUI del dispositivo. La señal de salida de esta sección ira a cada uno de los bloques secundarios que conforman la etapa, con el fin de llevar a cabo procesos independientes, de acuerdo a la función que deba cumplir cada bloque.
- **Bloque comparador de tensión.** Encargado de generar una señal digital (en sincronización con la señal de entrada al ADC), cuyos flancos de subida indicaran al microcontrolador que la señal entrante al ADC hace un cruce ascendente por cero en ese preciso instante. Este evento permitirá al microcontrolador llevar a cabo un muestreo secuencial en tiempo equivalente sobre la señal de entrada al conversor, y de esta modo lograr que el dispositivo sea capaz de medir señales cuya frecuencia sea mucho mayor a la frecuencia de muestreo del ADC interno en el PIC.
- **Bloque sumador DC.** Como su nombre lo indica, se encargara de sumar un nivel DC a la señal proveniente de la sección vertical, con el fin de entregar al ADC del microcontrolador una señal cuya amplitud se encuentre dentro de su rango de referencia de trabajo.

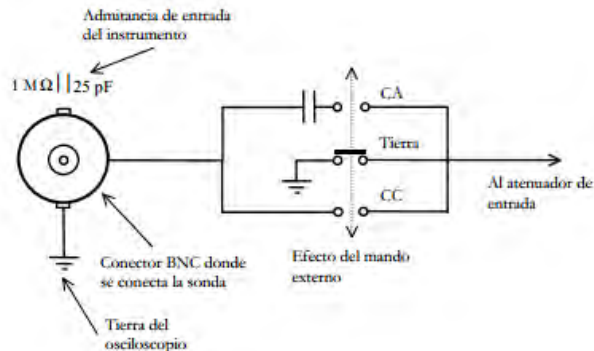
El diseño de la etapa contempla dos canales físicos idénticos e independientes, conformados por subetapas compuestas de O.P. (Amplificadores Operacionales), en diferentes configuraciones [1], de acuerdo a los requerimientos específicos que deba cumplir cada una de ellas.

3.1.4.1 Sección vertical

1. Selector de acoplamiento de entrada (AC-GND-DC)

Esta subetapa proporciona flexibilidad al dispositivo, ya que permite visualizar una señal con o sin su nivel de acoplo de CC (posiciones DC y AC respectivamente), o visualizar una línea horizontal en la pantalla con el fin de calibrar el instrumento o de restablecer el origen de coordenadas original, después de haber desplazado verticalmente una señal. La Figura 42 muestra el esquema eléctrico del selector de acoplamiento de entrada.

Figura 42. Selector de acoplamiento de entrada (AC-GND-DC)



Fuente: JJGDR-UCA. Instrumentación Electrónica. Juan José González de la Rosa. p. 17 [en línea].

http://www2.uca.es/grupinvest/instrument_electro/ppjjgdr/Electronics_Instrum/Electronics_Instrum_Files/temas/T2_oscilos_por.pdf

A menudo es necesario visualizar una señal sin su nivel de continua (CC o DC), ya que este puede ser elevado y poco relevante, como por ejemplo el nivel de acoplo de la señal de rizado de fuentes de alimentación, si se esta interesado en medir el rizado. La posición AC, de bloqueo de continua, logra este objetivo, proporcionando consecuentemente mayor resolución a las medidas, ya que el control del factor de escala vertical podrá situarse en posiciones bajas, menos voltios en una división. Sin embargo, las señales de baja frecuencia pueden ser atenuadas de manera no deseada, debido a que la señal de entrada pasa por un condensador, cuya reactancia a estas frecuencias es elevada y absorbe gran parte de la tensión de entrada.

En la posición DC la señal de entrada se conecta directamente al atenuador, y la señal es mostrada en pantalla con su nivel de acoplamiento.

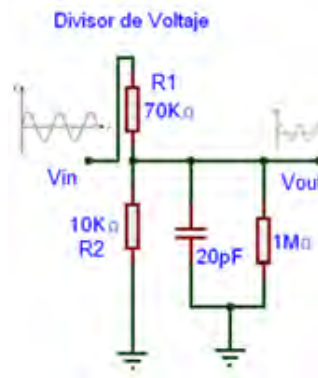
La posición GND, se encargara de conectar a tierra la entrada de la sección vertical del dispositivo, de manera que en la pantalla del osciloscopio no se desplegará ninguna medición dada por la sonda de prueba, ya que únicamente permite observar la referencia (0 voltios).

2. Atenuador compensado

Posterior al selector de acoplamiento, se implemento una subetapa, encargada de atenuar la señal de entrada, de tal forma que su amplitud pico a pico, se encuentre dentro del rango de tensión de referencia del conversor interno en el microcontrolador, el cual se configuro mediante software en el intervalo de 0 a 5 Voltios.

De acuerdo a lo anterior y teniendo en cuenta que el dispositivo debía ser capaz de medir señales eléctricas cuya amplitud no supere los 40Vpp (aunque en el caso de utilizar una sonda atenuada de alta impedancia en la medición se lograría medir hasta 400Vpp, como se vera mas adelante), se diseño el divisor de tensión mostrado en la Figura 43.

Figura 43. Esquema del atenuador de entrada implementado (Divisor de Voltaje).



El cual se encarga de atenuar la señal de entrada V_{in} de acuerdo a la siguiente relación.

$$V_{out} = \frac{V_{in} * R_2}{(R_1 + R_2)} \quad \text{Ecuación 11}$$

Los valores de R_1 y R_2 se establecieron, teniendo en cuenta el caso particular en el cual la señal de entrada presente la máxima amplitud, establecida para el dispositivo (40Vpp), lo que implica que la amplitud en la señal de salida del atenuador deberá ser de 5Vpp, debido a que es esta la magnitud máxima de la señal de entrada al conversor del microcontrolador (de acuerdo al rango de referencia en el cual trabaja), de modo que la señal visualizada en pantalla no sobrepase el fondo de escala para cada una de las posiciones de sensibilidad en VOLTS/DIV del dispositivo.

Por tanto, el divisor de voltaje deberá atenuar la señal de entrada, de tal forma que cuando ingrese en el, una señal de 40Vpp, su señal de salida presentara una amplitud de 5Vpp (atenuándola en un factor de 1/8). Teniendo en cuenta la Ecuación 8 tendremos que:

$$5Vpp = \frac{40Vpp * R_2}{(R_1 + R_2)}$$

Obteniendo que:

$$R_1 = 7R_2$$

Se utilizaron resistencias de $10\text{K}\Omega$ y $70\text{K}\Omega$, debido a que son valores comerciales y además generan bajas corrientes (en el orden de los mA) a la entrada del dispositivo, aunque son prácticamente anuladas gracias al acople de alta impedancia en el dispositivo.

La atenuación debe ser la misma para cualquier frecuencia de entrada dentro del ancho de banda establecido para el dispositivo (la señal de entrada será atenuada siempre en un factor de $1/8$), de hay el atributo compensado, para esta subetapa.

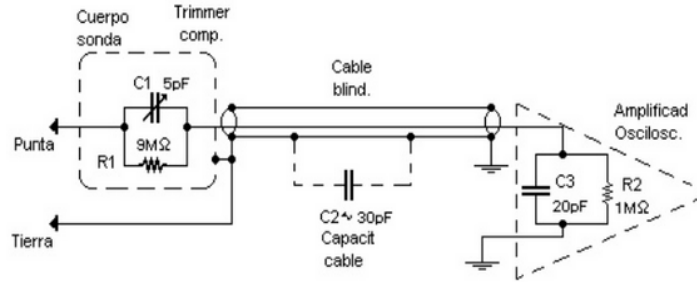
En la Figura 43, se puede observar un resistor de $1\text{M}\Omega$ en paralelo con un capacitor cerámico de 20pF , que en conjunto con una red resistencia/capacidad en paralelo interna en la sonda atenuada de alta impedancia utilizada en la medición de la señal, determina las posibles atenuaciones, consiguiendo así una atenuación constante en todo el ancho de banda del osciloscopio.

3. Sonda atenuada de alta impedancia

La capacidad de entrada del dispositivo desarrollado vista por un circuito bajo prueba, es bastante alta, especialmente cuando se incluye el cable blindado que es indispensable en estas mediciones. La impedancia de entrada resultante ($1\text{M}\Omega$ en paralelo generalmente con 100 pF) es a menudo muy baja para circuitos sensibles, y los “carga” exageradamente debido a la conocida acción del divisor de tensión. Peor aún, la capacitancia del circuito equivalente de entrada, puede hacer que algunos circuitos presenten un comportamiento erróneo, algunas veces hasta el punto de hacerlos entrar en oscilación.

La solución a este problema, consiste en utilizar “sondas” de alta impedancia. La sonda más popular, conocida como 10X, basa su operación en el circuito ilustrado en la Figura 44, en el cual para una señal de entrada DC, la sonda actúa simplemente como un divisor de voltaje por 10, y ajustando C1 (un trimmer incorporado en la propia sonda) a $1/9$ de la capacidad en paralelo de C2 y C3 (capacidades del cable blindado y del amplificador vertical del osciloscopio), el circuito se convierte en un divisor por 10 a cualquier frecuencia, con una impedancia de $10\text{M}\Omega$ en paralelo con unos pocos pF.

Figura 44. Circuito equivalente de una sonda compensada.



Fuente: Universidad Nacional de Colombia. Departamento de Ing. Eléctrica. OSCILOSCOPIO, GENERADORES DE FUNCIONES, MULTIMETROS. p. 7 [en línea].
<http://es.slideshare.net/guest07963/osciloscopio-presentation>

4. Acople de impedancias atenuador-amplificador

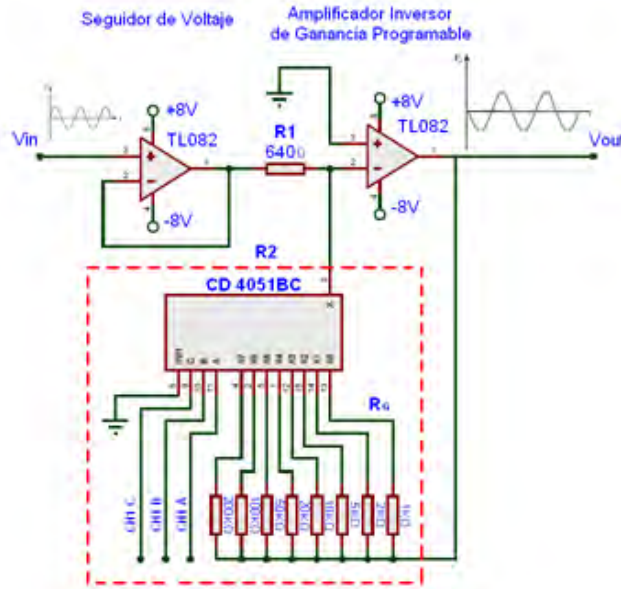
Posterior a la subetapa encargada de atenuar la señal de entrada, se implementó un seguidor de tensión, el cual permite un acople de impedancias (presenta una impedancia de entrada muy elevada, mientras que la de salida es prácticamente nula), entre el atenuador de voltaje y la subetapa posterior, encargada de amplificar la señal, permitiendo así un menor consumo en cuanto a la cantidad de corriente por parte del amplificador en la siguiente etapa y una mayor transferencia de potencia de la señal, lo cual permitirá obtener una medición en el dispositivo mucho más exacta.

5. Amplificador de ganancia programable (PGA)

Los amplificadores de ganancia programables se basan en el uso de interruptores que conmutan las resistencias de la red de realimentación del Amplificador Operacional, con el fin de obtener ganancias distintas dependiendo del interruptor que se encuentre cerrado. Los interruptores son controlados a través de señales digitales, cuyos niveles eléctricos se adaptan convenientemente a los requerimientos del interruptor.

El amplificador de ganancia programable diseñado e implementado durante el desarrollo del proyecto se muestra en la Figura 45, este utiliza el multiplexor CD4051, para conmutar cada una de las 8 resistencias de la red de realimentación del A.O., de acuerdo a las señales de control (A, B, C), las cuales son gobernadas por el Microcontrolador PIC18F2553.

Figura 45. Esquema del amplificador inversor de ganancia programable implementado.



En cuanto al multiplexor CD4051BC, se debe mencionar que presenta una resistencia interna debida a la acción del interruptor, la cual equivale aproximadamente 80Ω (varia de acuerdo al voltaje de operación y a la temperatura de trabajo, como se muestra en la Tabla 42), y además una alta impedancia cuando esta desenergizado (funciona como circuito abierto), al igual que una baja disipación de potencia (1 micro Watt)¹⁸.

Tabla 42. Parámetros de trabajo del multiplexor CD4051BC

Symbol	Parameter	Conditions	-55°C		+25°			+125°C		Units
			Min	Max	Min	Typ	Max	Min	Max	
I_{DD}	Quiescent Device Current	$V_{DD}=5V$ $V_{DD}=10V$ $V_{DD}=15V$		5 10 20			5 10 20		150 300 600	μA μA μA
R_{ON}	"ON" Resistance (Peak for $V_{EE} \leq V_{IS} \leq V_{DD}$)	$R_L = 10\text{ k}\Omega$ (any channel selected)								
		$V_{DD}=2.5V$, $V_{EE}=-2.5V$ or $V_{DD}=5V$, $V_{EE}=0V$		800		270	1050		1300	Ω
		$V_{DD}=5V$ $V_{EE}=-5V$ or $V_{DD}=10V$, $V_{EE}=0V$		310		120	400		550	Ω
		$V_{DD}=7.5V$, $V_{EE}=-7.5V$ or $V_{DD}=15V$, $V_{EE}=0V$		200		80	240		320	Ω

Fuente: National semiconductor. Hoja de especificaciones CD4051BC. p. 2[en línea].
<http://www.datasheetcatalog.org/datasheets/208/109138_DS.pdf>

¹⁸ National semiconductor. Hoja de especificaciones CD4051BC. p. 3[en línea].
http://www.datasheetcatalog.org/datasheets/208/109138_DS.pdf

De acuerdo a la Figura 45, el circuito equivalente al amplificador de ganancia programable, no es más que un A.O. en configuración inversora, cuya ganancia estará dada por la siguiente relación.

$$G = -\frac{V_{out}}{V_{in}} = -\frac{R_2}{R_1} \quad \text{Ecuación 12}$$

En donde R2 corresponderá a la resistencia interna del multiplexor (80Ω), sumada con la resistencia de realimentación seleccionada por el microcontrolador (la cual denominaremos en adelante como R_G), a través de los pines de control del multiplexor.

De acuerdo a lo anterior y teniendo en cuenta que el valor de R1 se estableció en 640Ω , obtendremos la siguiente relación.

$$G = \frac{80\Omega + R_G}{640\Omega}$$

En donde:

$$R_G = (640\Omega * G) - 80\Omega \quad \text{Ecuación 13}$$

Esta última ecuación, será fundamental en el calculo de los valores de cada resistencia en la red de realimentación del amplificador inversor de ganancia programable, de acuerdo a la ganancia establecida para cada uno de los factores de escala del selector de voltaje sobre división (VOLTS/DIV) implementado en la pantalla del dispositivo.

6. Calculo de ganancias para la red de realimentación del PGA

El calculo de ganancias para cada uno de los factores de escala seleccionado a través del control VOLTS/DIV desplegado en la pantalla del osciloscopio, dependerá de los siguientes parámetros.

- La magnitud máxima de tensión pico a pico que puede representarse en la pantalla del osciloscopio, de acuerdo al factor de escala seleccionado, teniendo en cuenta que la cuadrícula de ploteo del dispositivo utiliza un total de 8 divisiones verticales, para representar la amplitud de las señales de entrada. Por ejemplo si nos encontramos en una escala de 1V/DIV, tendremos que la máxima tensión pico a pico que se podrá visualizar en la pantalla del osciloscopio será de 8V.

- La magnitud de tensión pico a pico de la señal en la salida del atenuador, cuando en su entrada ingresa cada una de las diferentes señales obtenidas en el anterior ítem, considerando que el factor de atenuación para cualquier señal de entrada al divisor de voltaje será de 1/8, como ya ha sido mencionado anteriormente.

La Tabla 43, muestra la magnitud de los diferentes parámetros para cada uno de los factores de escala seleccionado.

Tabla 43. Parámetros para el cálculo de las ganancias en el PGA.

Factor de escala seleccionado [VOLTS/DIV]	Amplitud máxima de la señal de entrada representada en pantalla [Vpp]	Amplitud en la señal de salida del atenuador [Vpp]
5V	40V	5V
2V	16V	2V
1V	8V	1V
500mV	4V	500v
200mV	1.6V	200mV
100mV	800Mv	100mV
50mV	400mV	50mV
20mV	160mV	20mV

La segunda columna, indica la máxima amplitud que puede presentar la señal de entrada al dispositivo, para ser representada de forma adecuada en la pantalla del mismo, de acuerdo al factor de escala en el cual nos encontremos. Cuando la señal de entrada alcanza el límite establecido por el factor de escala seleccionado, la magnitud pico a pico de la señal de entrada al ADC del microcontrolador, estará oscilando al borde de los límites de referencia establecidos para el convertidor (la amplitud de la señal estará cerca de los 5Vpp).

Lo que significa que la ganancia para cada uno de los factores de escala del osciloscopio, debe permitir al PGA, obtener una señal de salida de 5Vpp, en el evento que su señal de entrada alcance el límite establecido por las magnitudes consignados en la tercera columna de la Tabla 43.

De acuerdo a lo anterior y teniendo en cuenta la Ecuación 12 tendremos que la ganancia del amplificador inversor de ganancia programable estará dada por.

$$G_{FE} = -\frac{5V_{pp}}{V_{AM}}$$

En donde G_{FE} , corresponde a la ganancia del PGA, para cada uno de los factores de escala seleccionado y V_{AM} , corresponde a la amplitud máxima de la señal de entrada atenuada que podrá ser representada en la pantalla del dispositivo igualmente para cada factor de escala seleccionado.

De esta manera, se realizó cálculo de las diferentes ganancias, cuyas magnitudes son consignadas en la Tabla 44.

Tabla 44. Ganancias calculadas para cada factor de escala.

Factor de escala seleccionado [VOLTS/DIV]	Amplitud máxima de la señal de entrada representada en pantalla [Vpp]	Ganancia del PGA
5V	40V	1
2V	16V	2.5
1V	8V	5
500mV	4V	10
200mV	1.6V	25
100mV	800mV	50
50mV	400mV	100
20mV	160mV	250

A partir de las magnitudes de ganancia calculadas y teniendo en cuenta además la Ecuación 13 presentada en el apartado anterior, se realizó el cálculo de las diferentes resistencias de realimentación R_G , que seleccionara el microcontrolador, a través del multiplexor, de acuerdo al factor de escala seleccionado en la pantalla del dispositivo.

Tabla 45. Resistencia de realimentación calculada para cada factor de escala seleccionado.

Factor de escala seleccionado [VOLTS/DIV]	Ganancia del PGA	Resistencia de realimentación R_G [Ω]	Valor del trimmer utilizado [Ω]
5V	1	560	1K
2V	2.5	1.52K	2K
1V	5	3.12K	5K
500mV	10	6.32K	10K
200mV	25	15.92K	20K
100mV	50	31.92K	50K
50mV	100	63.92K	100K
20mV	250	159.92K	200K

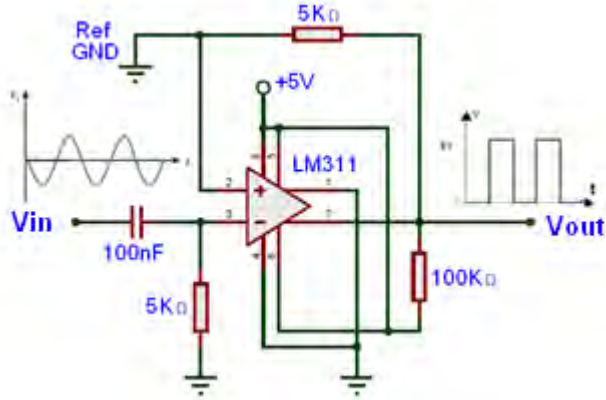
Con el fin de lograr una mayor precisión de ganancia en el sistema vertical del dispositivo, se implementó un arreglo en la red de realimentación del PGA, compuesto por varios trimmer (de valores comerciales, consignados en la cuarta columna de la Tabla 45), que permiten ajustar su resistencia de manera exacta, de acuerdo a la magnitud R_G , requerida para cada factor de escala seleccionado.

3.1.4.2 Bloque comparador de tensión

Este bloque está conformado por un A. O. configurado como comparador de tensión (Figura 46), se encargará de generar una onda cuadrada a partir de la señal proveniente de la sección vertical del dispositivo, la cual es utilizada por el microcontrolador como señal de reloj para llevar a cabo un proceso denominado muestreo en tiempo equivalente secuencial, que permitirá al dispositivo ser capaz

de medir señales eléctricas, cuya frecuencia sea mucho mayor a la frecuencia de muestreo del ADC interno en el microcontrolador.

Figura 46. Esquema del A.O. configurado como comparador de tensión.



Los flancos de subida en la señal de salida del comparador, serán registrados por el microcontrolador, con el fin de establecer una sincronización entre la señal de entrada al ADC y el instante en el cual la señal será muestreada.

1. Métodos de muestreo del dispositivo

El muestreo es el proceso de convertir una porción de una señal de entrada en un número de valores eléctricos individualizados con fines de almacenamiento, procesamiento o presentación de los mismos. La magnitud de cada punto muestreado es igual a la amplitud de la señal de entrada en el instante del muestreo de la señal en el tiempo.

El muestreo podría definirse como una toma de instantáneas, cada una de las cuales corresponderá a un punto específico en el tiempo de la forma de onda de entrada al dispositivo. Estas instantáneas pueden ser luego dispuestas en un orden específico en el tiempo con el fin de poder reconstruir la señal en la pantalla del equipo.

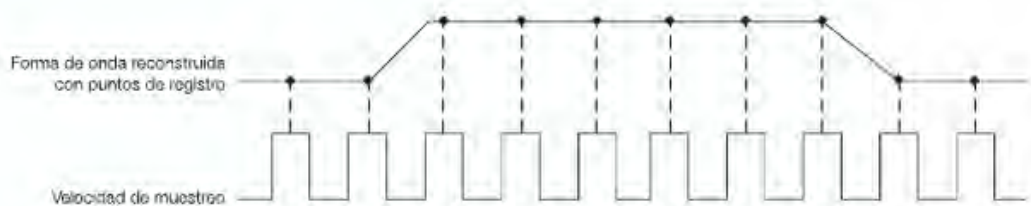
Aunque existen un número determinado de diferentes formas de aplicar la tecnología de muestreo, el dispositivo desarrollado utiliza dos métodos básicos: muestreo en tiempo real con interpolación lineal, y muestreo en tiempo equivalente secuencial. Cada método presentara diferentes ventajas, dependiendo del tipo de medidas que se desee realizar.

- Muestreo en tiempo real con interpolación

El método de muestreo en tiempo real, resulta ideal para señales cuyo rango de frecuencia es menor que la mitad de la velocidad máxima de muestreo del ADC interno en el microcontrolador (cuya frecuencia de muestreo es aproximadamente de $22\mu\text{s}$, de acuerdo a los requerimientos de adquisición establecidos en la sección 3.1.3.1).

En este caso, el dispositivo puede adquirir puntos más que suficientes con un solo “barrido” de la forma de onda de entrada como para conformar una imagen precisa, como se indica en la Figura 47

Figura 47. Método de muestreo en tiempo real.



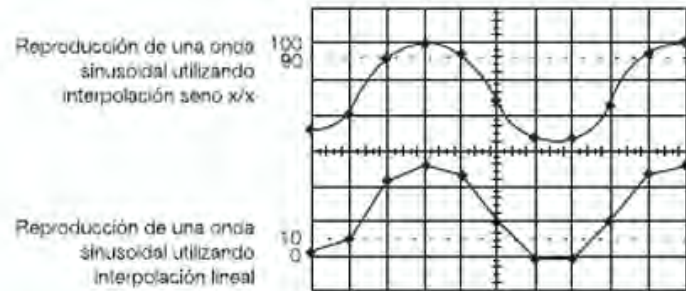
Fuente: Tectronix. El XYZ de los osciloscopios. Conceptos básicos. p. 26 [en línea].
<http://es.scribd.com/doc/7630284/39/Muestreo-en-tiempo-equivalente>.

Sin embargo, podrá resultar difícil para el dispositivo poder visualizar la señal representada por pocos puntos, los cuales en este caso estarán representando porciones de alta frecuencia de la señal de entrada. Con el fin de poder visualizar estas señales de mayor frecuencia, se utilizan métodos de presentación con interpolación.

La interpolación “conecta los puntos” para que una señal muestreada pocas veces en cada ciclo pueda ser representada de forma precisa sobre la pantalla del dispositivo. El osciloscopio recoge unos pocos puntos de muestreo de la señal en un solo barrido y posteriormente utiliza la interpolación para rellenar los espacios intermedios. Esta técnica de procesamiento es utilizada para estimar el aspecto de la forma de onda, basándose tan solo en unos pocos puntos.

La interpolación lineal conecta los puntos de las muestras mediante líneas rectas, por lo tanto este método está limitado a la reconstrucción de señales de flancos rectos, tales como las ondas cuadradas, según se ilustra en la Figura 48.

Figura 48. Interpolación lineal y seno x/x



Fuente: Tectronix. El XYZ de los osciloscopios. Conceptos básicos. p. 27 [en línea].
<http://es.scribd.com/doc/7630284/39/Muestreo-en-tiempo-equivalente>.

La interpolación seno x/x es más versátil y conecta los puntos de las muestras mediante curvas, como se muestra en la Figura 48. La interpolación seno x/x es un proceso matemático en el cual se calculan los puntos que rellenan el espacio entre muestras reales.

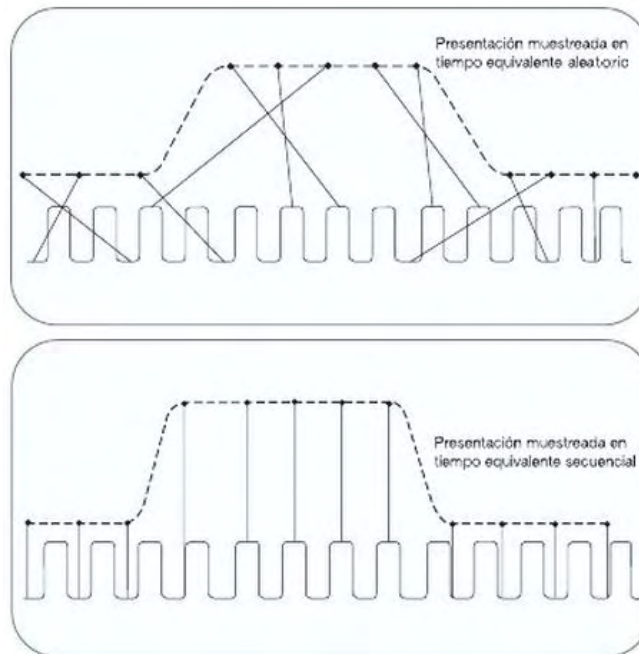
- **Muestreo en tiempo equivalente**

Esta técnica de muestreo se emplea para representar señales eléctricas de alta frecuencia, las cuales superan los límites del ADC interno en el microcontrolador.

El muestreo en tiempo equivalente permite reconstruir en pantalla, la imagen de una señal repetitiva, capturando tan solo una parte de la información en cada repetición. La forma de onda se va visualizando como una cadena de puntos interpolados linealmente, permitiendo al osciloscopio capturar con precisión señales cuyas componentes de frecuencia son mucho más elevadas que la velocidad de muestreo del convertidor interno en el microcontrolador.

Existen dos tipos de muestreo en tiempo equivalente: aleatorio y secuencial (Figura 49). Brindando cada uno diferentes ventajas. El muestreo en tiempo equivalente aleatorio permite la presentación de la señal en entrada antes del punto de disparo, sin la utilización de una línea de retardo. El muestreo en tiempo equivalente secuencial proporciona una resolución en tiempos y una precisión mucho mayores. Ambos métodos requieren que la señal de entrada sea repetitiva.

Figura 49. Presentación de una señal muestreada en tiempo equivalente aleatorio y secuencial respectivamente.



Fuente: Tectronix. El XYZ de los osciloscopios. Conceptos básicos. p. 27 [en línea].
<http://es.scribd.com/doc/7630284/39/Muestreo-en-tiempo-equivalente>.

- **Muestreo en tiempo equivalente secuencial**

El sistema de muestreo en tiempo equivalente secuencial adquiere una muestra por disparo, independientemente del factor en la escala horizontal VOLTS/DIV seleccionado, o de la velocidad de barrido. Cuando se detecta un disparo, se adquiere una muestra después de un retardo muy corto, pero muy bien definido. Cuando se produce el próximo disparo, se añade un pequeño incremento de tiempo (Δt) a este retardo, y el conversor adquiere otra muestra. Este proceso se repite hasta completar el barrido horizontal de la pantalla en el dispositivo, añadiendo un “ Δt ” en cada adquisición previa.

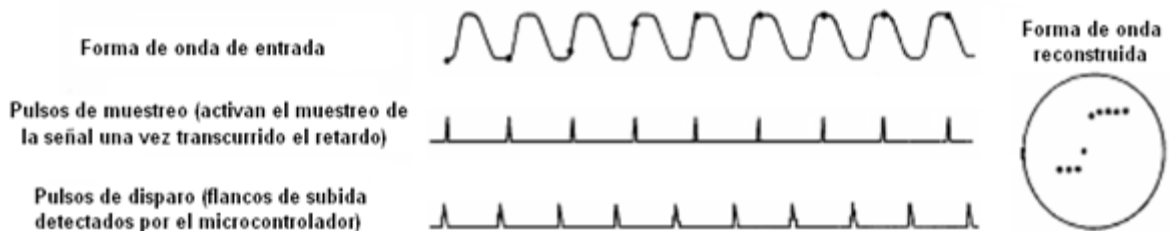
Los puntos de las muestras aparecen de izquierda a derecha de forma secuencial a lo largo de la forma de onda en la pantalla del dispositivo.

En la reconstrucción de la onda (Figura 50), los flancos de subida de la señal proveniente del comparador de tensión, serán los encargados de activar el pequeño retardo Δt (de acuerdo al factor de escala horizontal seleccionado en GUI del dispositivo), antes de dar inicio a la adquisición de la primera muestra (para su posterior cuantificación y codificación) de la señal de entrada, llevado a cabo por el conversor del microcontrolador. El retardo entre el disparo y el

muestreo de la señal se ira incrementando en factores del numero de muestras adquiridas hasta ese preciso instante. Cada muestra es transmitida al sistema embebido MINI2440, previamente codificada con un número de 12 bits, dado por la resolución del ADC interno en el microcontrolador, y corresponderá a uno de los puntos utilizados en la reconstrucción digital de la señal de entrada, que será visualizada en la pantalla del dispositivo.

El número de puntos utilizados en un barrido horizontal de pantalla, será siempre el mismo (255 muestras), por tanto, lo que se modifica es únicamente la magnitud del retardo entre el disparo y el muestreo de la señal (Δt), de acuerdo al factor de escala horizontal seleccionado.

Figura 50. Formas de onda correspondiente al muestreo en tiempo equivalente secuencial.



Fuente: Universidad Nacional de Quilmes. Ingeniería en Automatización y Control Industrial. Cátedra de Instrumentos y Mediciones. Osciloscopios. p. 23 [en línea]. http://iaci.unq.edu.ar/materias/ins_med/archivos/Instymed_t8.pdf

3.1.4.3 Bloque sumador DC

Este bloque esta conformado por un A. O. configurado como sumador no inversor, el cual se encarga de agregar un offset de DC (2.5Volts), a la señal proveniente de la sección vertical del dispositivo, de tal forma que la señal de salida, estará desplazada verticalmente un nivel DC de 2.5V.

De esta manera se asegura que el valor de la señal de entrada al ADC del microcontrolador, caiga dentro del rango de referencia de trabajo previamente establecido, evitando así el ingreso de tensiones negativas al conversor.

En el esquema del bloque sumador DC, mostrado en la Figura 51, se observa, que previo al sumador de tensión, la señal ingresa a un A. O. configurado como inversor de voltaje, con el fin de poner en fase la señal de entrada al dispositivo con la señal de entrada al conversor (debido al desfase de la señal generado por el amplificador inversor de ganancia programable implementado en la sección

vertical del dispositivo), cabe destacar que la ganancia de este inversor de voltaje debe ser unitaria, evitando así que la señal de entrada sea amplificada.

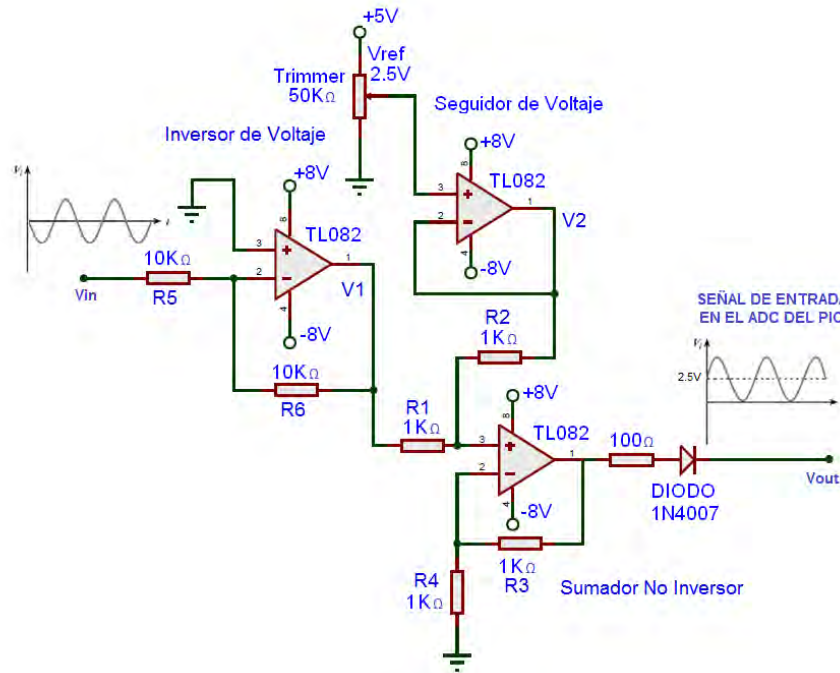
Para ello se tuvo en cuenta la relación entre la entrada y salida de la configuración en particular de este amplificador operacional, la cual esta dada por la siguiente relación:

$$V1 = -\left(\frac{R6}{R5}\right) * Vin \quad \text{Ecuación 14}$$

De la cual se concluye que para lograr una ganancia unitaria, las resistencias R5 y R6 deben presentar el mismo valor.

Se incluye también un A. O. configurado como seguidor de voltaje, el cual permite acoplar impedancias entre el divisor de tensión y el sumador de tensión. Este acople evitara que el nivel de tensión DC (suministrado por el divisor de tensión, a través de un trimmer), se “caiga”, asegurando siempre un offset de 2.5V en la señal de entrada al conversor del microcontrolador.

Figura 51. Esquema del bloque sumador DC, incorporado previo a la entrada del ADC interno en el microcontrolador.



Teniendo en cuenta la siguiente ecuación:

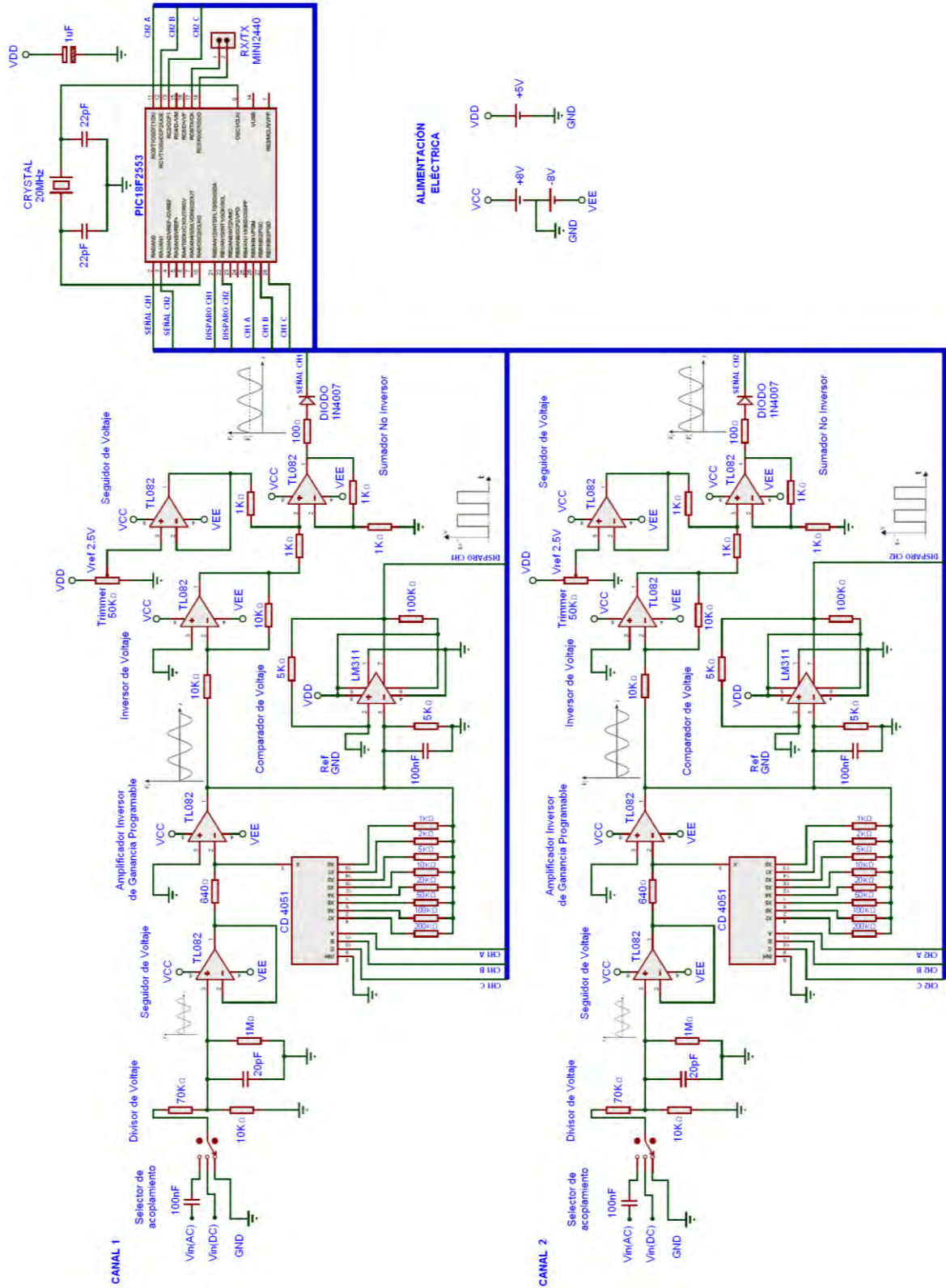
$$Vout = \left(\frac{R3}{R1}\right) * V1 + \left(\frac{R3}{R2}\right) * V2 \quad \text{Ecuación 15}$$

La cual relaciona la salida V_{out} , con las 2 señales de entrada V_1 y V_2 en el amplificador sumador no inversor mostrado en la Figura 51, se utilizaron resistencias de valores idénticos (R_1 , R_2 y R_3), evitando así que la señal proveniente de la sección vertical sea amplificada, ya que de lo contrario se obtendría una medida errónea de la señal de entrada en el dispositivo.

3.1.5 Esquema final de la placa de adquisición de datos implementada

Una vez descrito tanto el diseño como el funcionamiento de cada una de las secciones y bloques que conforman la etapa de entrada y acondicionamiento de la señal, así como también la configuración y funcionamiento de cada uno de los módulos utilizados en el microcontrolador, se presenta a continuación (Figura 52), el esquema final de la placa de adquisición de datos implementada en el dispositivo, la cual cuenta con 2 canales de entrada idénticos e independientes, conectados cada uno a diferentes pines físicos del microcontrolador, los cuales han sido previamente configurados como entradas análogas al ADC interno del mismo.

Figura 52. Esquema final de la placa de adquisición de datos incorporada en el dispositivo



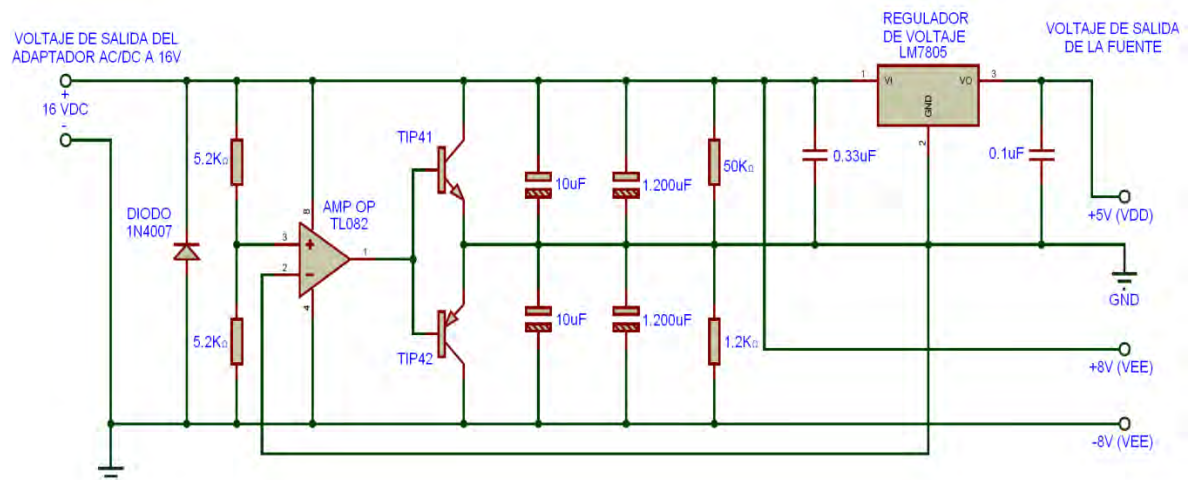
3.1.6 Fuente de alimentación

El osciloscopio digital dispone de un adaptador AC/DC, con entrada de 110V AC y salida de 16V DC, dispuesto en una carcasa independiente para reducir el efecto de las corrientes parasitas sobre el osciloscopio digital, debido al campo magnético generado por el transformador.

Teniendo en cuenta que el cargador suministra una salida de 16V DC, se diseñó una fuente de alimentación dual de $\pm 8V$ DC mostrada en la Figura 53, la cual se encarga de alimentar los diferentes componentes que conforman la tarjeta de adquisición de datos del dispositivo, incluyendo los amplificadores operacionales en sus diferentes configuraciones así como también los multiplexores.

El microcontrolador PIC18F2553 al igual que el sistema embebido MINI2440, se alimentan a través de un regulador de voltaje a 5V, debido a que suministra un voltaje muy estable, indispensable para el correcto funcionamiento tanto del sistema embebido como del PIC.

Figura 53. Diagrama de conexiones de la fuente de alimentación eléctrica



3.2 FIRMWARE

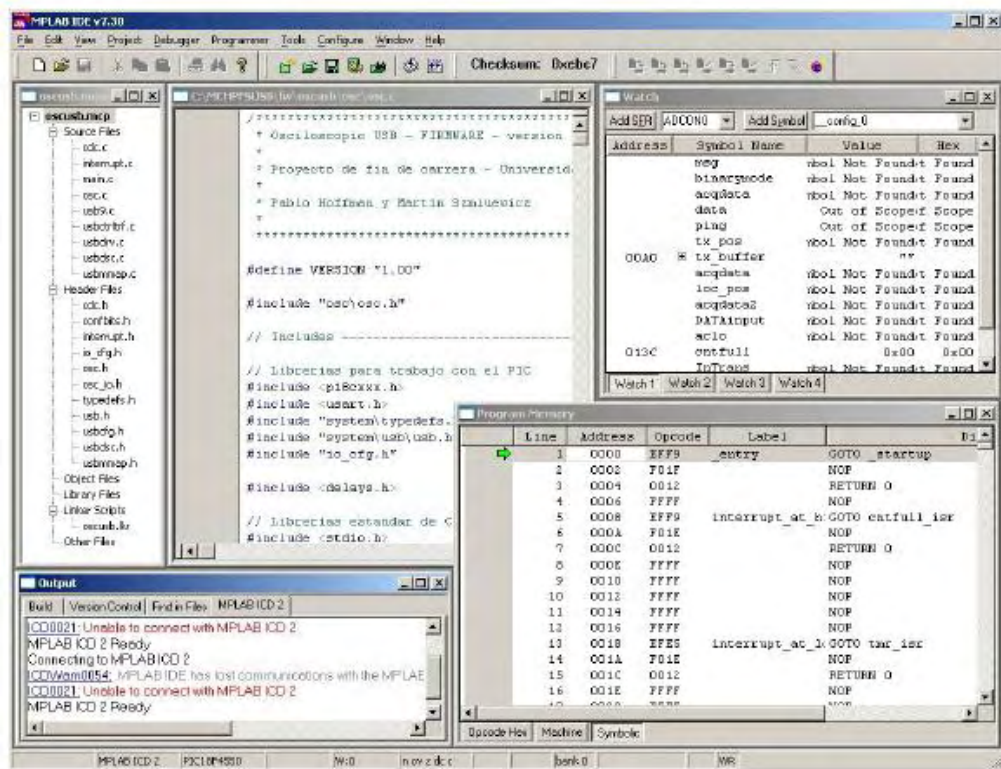
El firmware es el programa desarrollado en lenguaje ensamblador, que es ejecutado internamente por el microcontrolador para controlar los diferentes parámetros del osciloscopio. Este programa se encarga de mantener la sincronización entre el software y el hardware del equipo, ejecutando diferentes tareas de acuerdo, a las órdenes o requerimientos solicitados por el sistema

embebido MINI2440 a través de la interfaz grafica de usuario implementada sobre este último.

3.2.1 Herramienta de trabajo

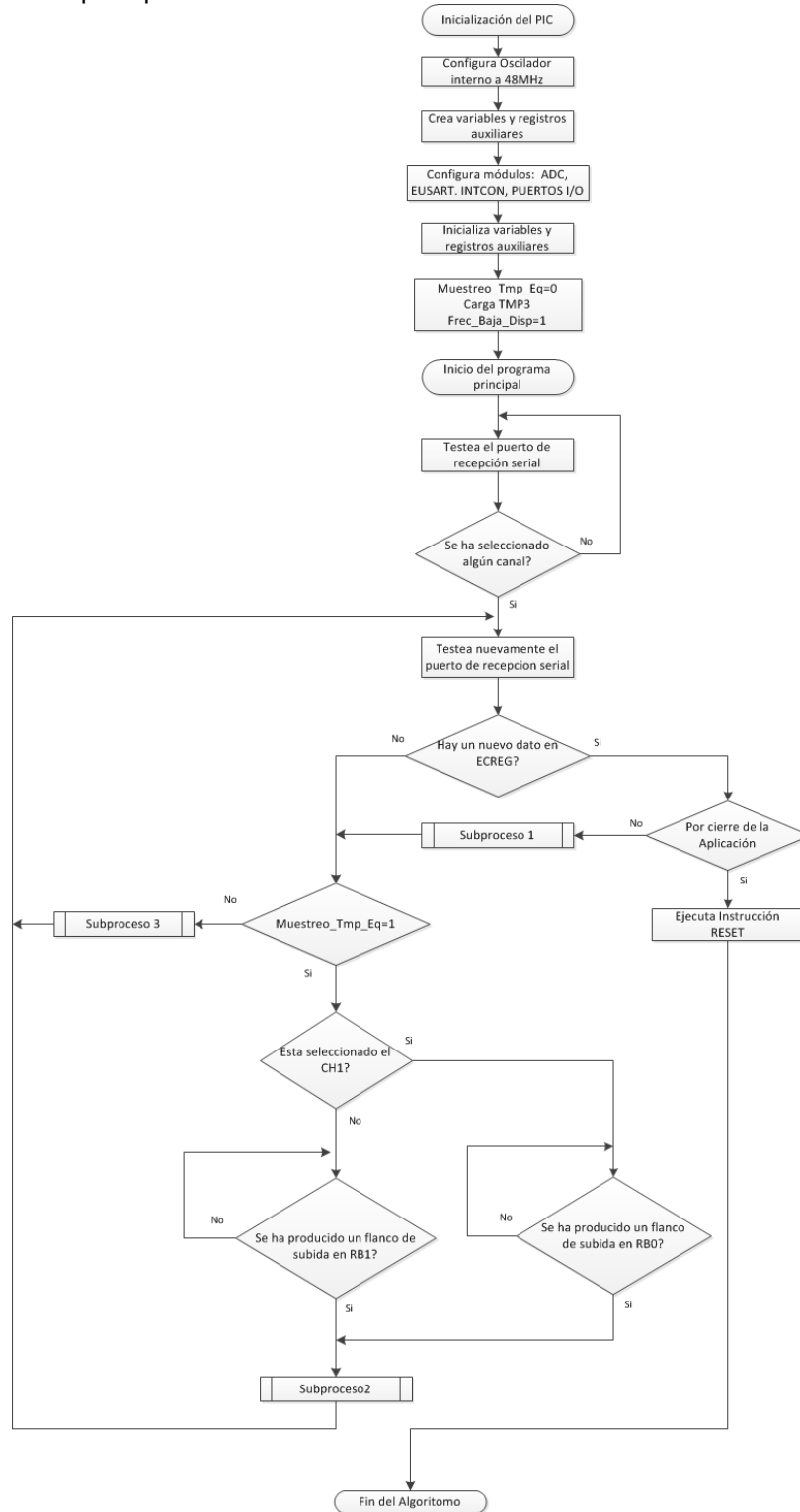
Este firmware fue editado, compilado y depurado con la herramienta software de distribución gratuita MPLAB IDE v8.70, el cual es un conjunto integrado de herramientas gratuitas para el desarrollo de aplicaciones, que emplean los Microcontroladores PIC y dsPIC de la marca MICROCHIP, este programa incluye funciones para acceder al hardware del microcontrolador, y además el compilador que incorpora es capaz de detectar posibles errores en el código fuente del firmware desarrollado. En la Figura 54 se muestra una captura de pantalla del software MPLAB IDE v.8.70.

Figura 54. Captura de Pantalla de MPLAB IDE V.8.70



3.2.2 Algoritmo principal del Microcontrolador

Figura 55. Algoritmo principal del Microcontrolador



El algoritmo principal del firmware, básicamente contempla dos acciones fundamentales que debe llevar a cabo el microcontrolador:

1. Realizar un continuo testeo en el puerto de recepción serial, con el fin de verificar si el sistema embebido MINI2440 ha solicitado algún tipo de requerimiento en cuanto a:
 - El canal seleccionado.
 - El control de los MUX; encargados de establecer la ganancia para cada canal, de acuerdo a la escala de amplitud seleccionada (V/División), a través de la GUI implementada en la pantalla TFT del sistema embebido MINI2440.
 - El control en el tiempo de muestreo de la señal de entrada, de acuerdo a la escala de tiempo seleccionada (ms/División).
2. Llevar a cabo la conversión A/D de la señal análoga de entrada para el canal seleccionado y posteriormente transmitir el resultado hacia el sistema embebido MINI2440, a través del puerto de transmisión serial.

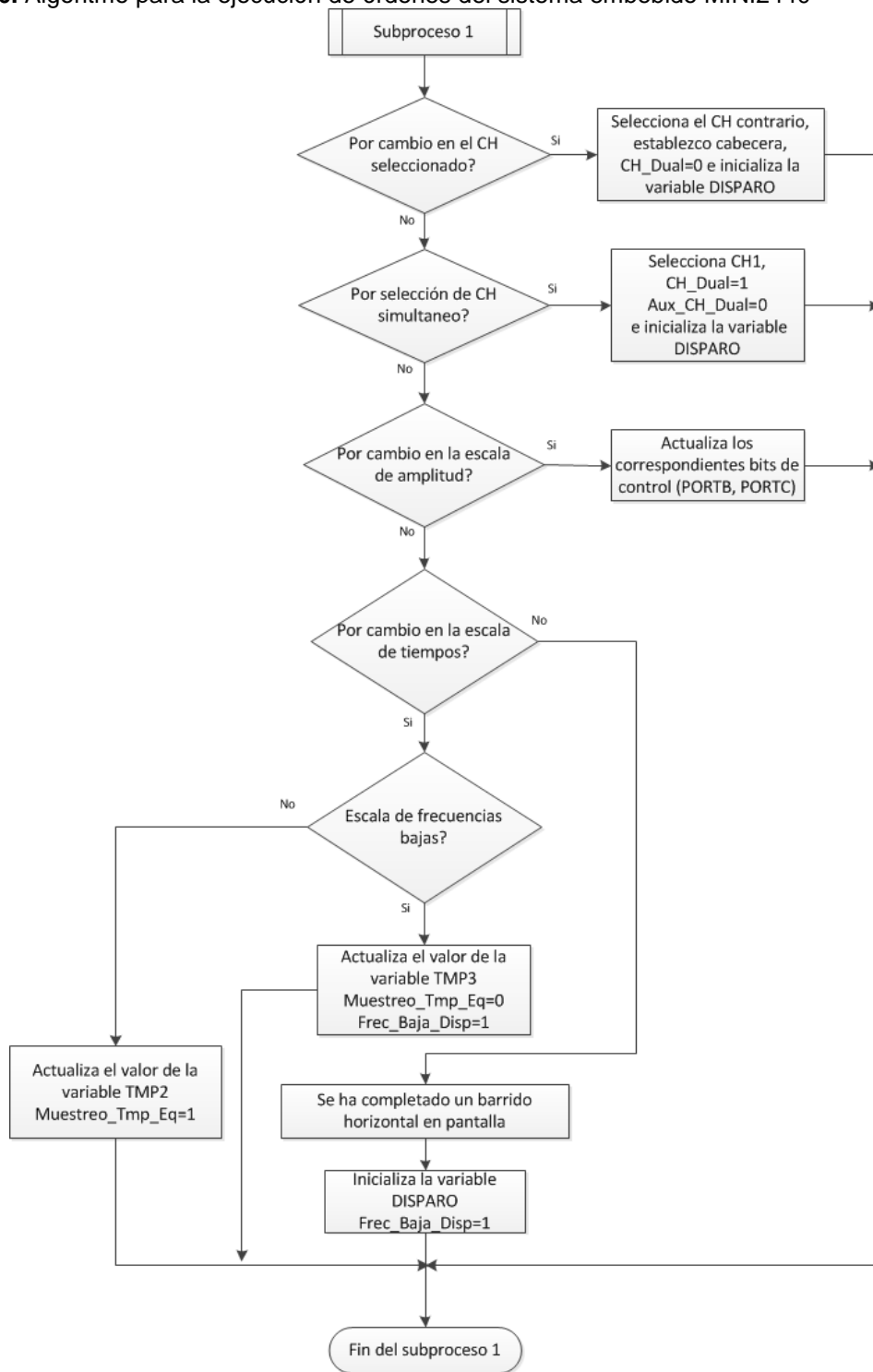
El algoritmo principal, presentado en la Figura 55, contempla además la configuración de los diferentes módulos utilizados en el desarrollo del firmware que ejecuta el microcontrolador, de igual manera se establece una frecuencia de 48MHz para el oscilador interno y se crean variables auxiliares que serán de gran ayuda en la ejecución de retardos dentro del programa.

3.2.3 Algoritmo para la ejecución de ordenes del sistema embebido MINI2440

El siguiente algoritmo (Figura 56), describe en forma general el subproceso 1 indicado en el diagrama de flujo principal del PIC. Básicamente contempla la lectura del puerto de recepción serial, a través del registro RCREG del microcontrolador y de acuerdo al dato recibido, se procederá a ejecutar el requerimiento solicitado por el sistema embebido MINI2440.

Si se ha solicitado un cambio en el canal de medida, el programa del microcontrolador modificara la configuración de los bits CHS3:CHS0 del registro de control ADCON0, y limpiara además el registro auxiliar DISPARO.

Figura 56. Algoritmo para la ejecución de órdenes del sistema embebido MINI2440



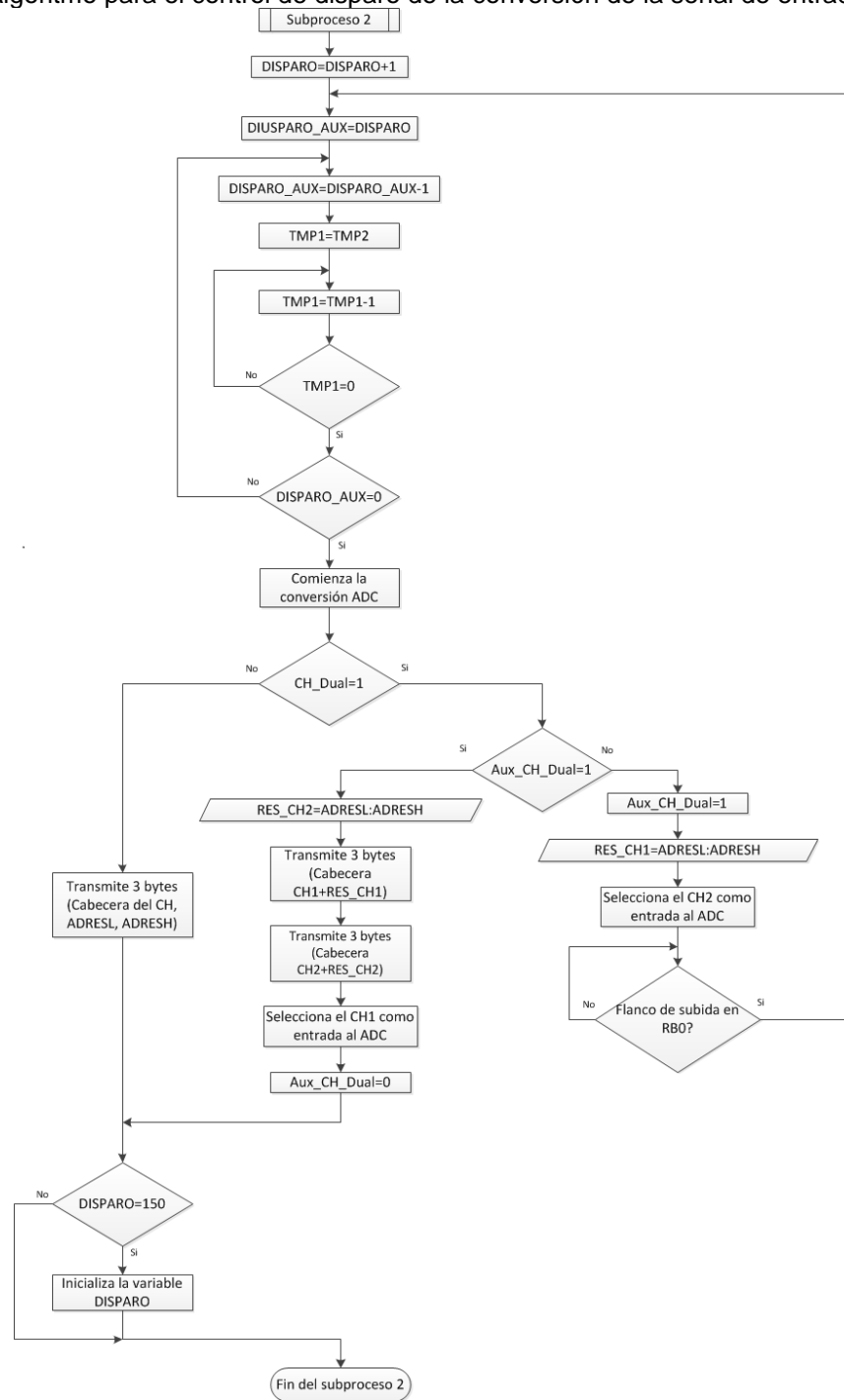
Por otro lado, si la solicitud se ha producido por cambio en la escala de amplitud de medida seleccionada, el microcontrolador actualizará los bits de control de uno de los multiplexores, dependiendo si se trata del CH1 (actualizará los bits 7:5 del registro PORTB), o del CH2 (actualizará los bits 2:0 del registro PORTC), de esta manera permite al usuario contar con 8 diferentes escalas de amplitud, para cada uno de los canales del osciloscopio.

En cambio, si lo que se ha producido es una variación en la escala de tiempos del osciloscopio, el microcontrolador se encargará de actualizar el valor del registro auxiliar TMP2, lo que permitirá indirectamente modificar la duración que debe transcurrir entre la señal de disparo (flanco de subida en el puerto RB0 o RB1, dependiendo del canal seleccionado) y el inicio de adquisición de la señal análoga de entrada en el canal seleccionado.

Finalmente si la solicitud no corresponde a una de las anteriormente descritas, el Microcontrolador deberá limpiar el registro auxiliar DISPARO (lo que indica que la señal visualizada ha llegado al límite de la pantalla) estableciendo así un control automático en la sincronización del disparo para una adecuada visualización de la señal análoga de entrada en el canal seleccionado.

3.2.4 Algoritmo para el control del disparo y conversión de la señal de entrada.

Figura 57. Algoritmo para el control de disparo de la conversión de la señal de entrada



En la Figura 57 se muestra el algoritmo de control correspondiente al subproceso 2 contemplado en el diagrama de flujo principal.

Básicamente en este subproceso, se presenta la forma en que el microcontrolador coordina el proceso de adquisición de la señal análoga de entrada de acuerdo a la señal de disparo detectada.

En el momento que se produce un flanco ascendente en uno de los puertos RB0 o RB1 (dependiendo del canal seleccionado), la variable auxiliar DISPARO incrementa en 1 su valor actual y el resultado se guarda en la variable DISPARO_AUX.

El valor de la variable auxiliar DISPARO_AUX, indicara al microcontrolador, el número de veces que debe ejecutar el retardo del bucle interno en el algoritmo (establecido por la variable TMP2), antes de llevar a cabo la adquisición de la señal análoga presente en el canal seleccionado.

De esta manera la adquisición en la señal análoga de entrada se llevara a cabo una vez se ejecute el retardo dado por:

$$TR = (DISPARO) * (RETARDO ESTABLECIDO POR TMP2) \quad \text{Ecuación 16}$$

Posterior a ello se llevara a cabo el proceso de conversión A/D de la señal adquirida y el resultado será transmitido hacia el sistema embebido MINI2440 a través de una trama de 3 bytes que contendrá: el resultado de la conversión (dividido en 2 bytes ADRESH:ASRESL) y un byte de cabecera que indicara al sistema embebido el canal al cual corresponde ese resultado.

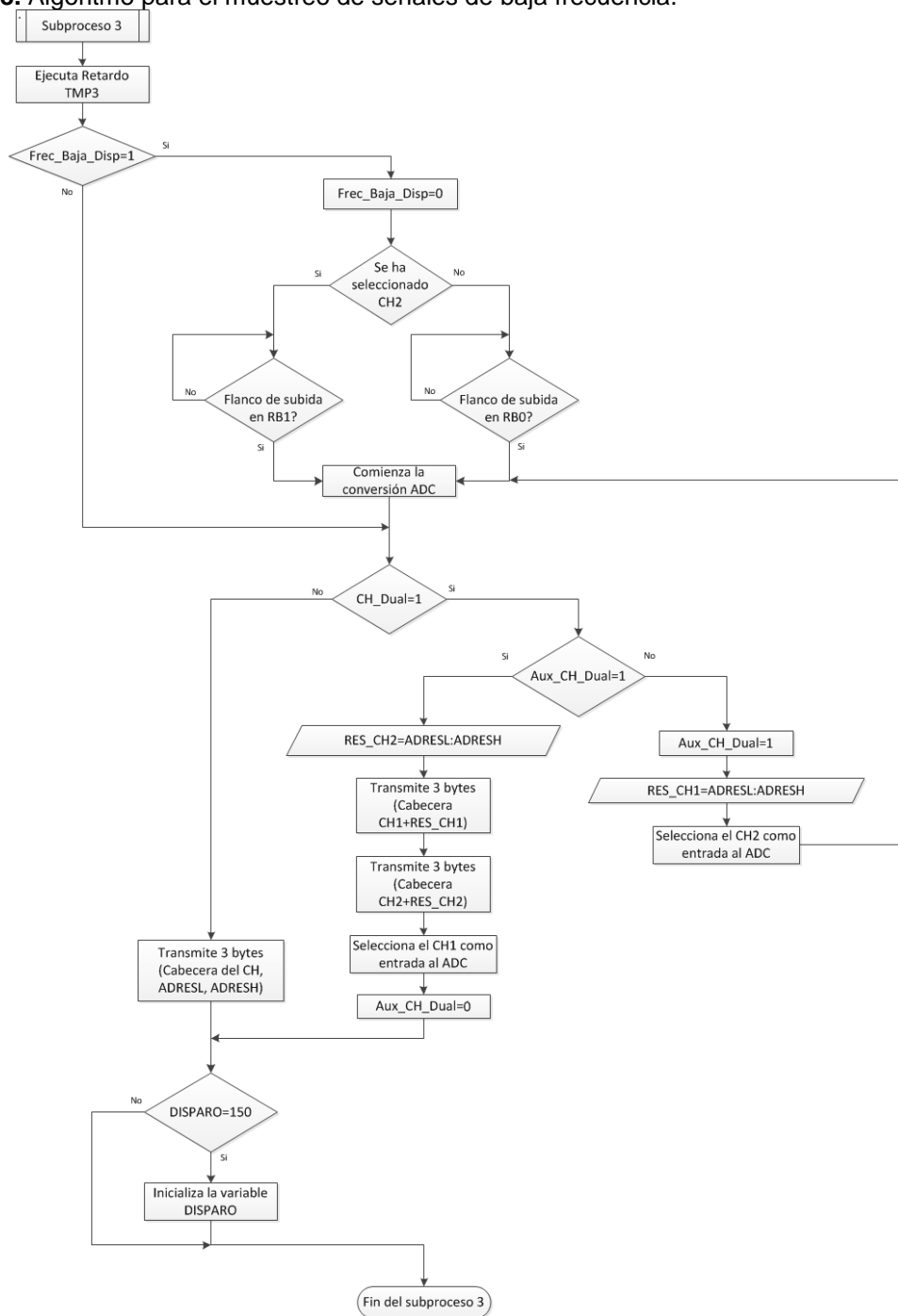
En la parte inferior del diagrama de flujo se contempla un bucle para verificar si ya han sido transmitidas las 150 muestras suficientes para cubrir un completo ploteo de la señal a lo ancho de la pantalla del equipo, sin perder claridad en la señal visualizada.

De esta manera cada vez que el ploteo de la señal llega al final de la pantalla del equipo, el microcontrolador inicializara nuevamente la variable DISPARO, y transmitirá otra nueva secuencia de 150 muestras.

Cabe resaltar que dependiendo del valor cargado en la variable TMP2 a través del subproceso 1, el equipo será capaz de muestrear señales en un rango de frecuencia determinado. A medida que el valor cargado es más pequeño, el equipo será capaz de medir señales de mayor frecuencia.

3.2.5 Algoritmo para el muestreo de señales de baja frecuencia (Método de muestreo en tiempo real con interpolación lineal).

Figura 58. Algoritmo para el muestreo de señales de baja frecuencia.



En la Figura 58 se muestra el algoritmo de control correspondiente al subproceso 3 contemplado en el diagrama de flujo principal del Microcontrolador.

Básicamente en este subproceso, se presenta la forma en que el microcontrolador muestrea señales de baja frecuencia; este subproceso es ejecutado cuando el usuario selecciona una de las 2 primeras escalas en la barra de control TIME/DIV de la GUI. En este subproceso, el microcontrolador no ejecuta retardo alguno antes de muestrear la señal, únicamente coordina el primer muestreo en cada barrido horizontal de pantalla, utilizando el sistema de disparo incorporado en la tarjeta de adquisición de datos del dispositivo, esto con el fin de que la señal de entrada, pueda ser visualizada de forma estática sobre la pantalla del equipo.

3.3 SOFTWARE

Este capítulo describe la implementación del software, el cual corresponde a la aplicación implementada en el sistema embebido MINI2440 para controlar el osciloscopio, se incluye información sobre el diseño, herramientas utilizadas, métodos de programación empleados, estructura del código y funcionamiento del software.

3.3.1 Selección de herramientas

La primera decisión en el momento de desarrollar la aplicación para controlar el osciloscopio fue la selección de las herramientas a utilizar, particularmente:

- El lenguaje de programación a usar.
- El entorno de desarrollo a utilizar para crear la aplicación.
- Las librerías a utilizar para controlar diversos aspectos de la aplicación.

3.3.1.1 Lenguaje de programación

Para tomar la decisión del lenguaje a utilizar para la elaboración del software gráfico que se implementara en el sistema embebido MINI2440 para el control del osciloscopio desarrollado se han tenido en cuenta los siguientes requisitos:

- Debe incorporar interfaz gráfica.
- Debe ser de rápido desarrollo.
- Debe poder comunicarse con puertos virtuales de Windows de forma sencilla.

El lenguaje de propósito general seleccionado para programar la aplicación fue C#, al incorporar los mejores elementos de múltiples lenguajes de amplia difusión como C++, Java, Visual Basic o Delphi, permitiendo así el desarrollo de aplicaciones que combinen un control a bajo nivel, de lenguajes como C con la velocidad y sencillez de programación, de lenguajes como Visual Basic.

Fundamentalmente, las características que permitieron establecer a C# como lenguaje de programación en el desarrollo de la aplicación para el sistema embebido MINI2440 fueron:

- Ser un lenguaje moderno y orientado a objetos, con una sintaxis muy similar a la de C++ y Java. Combina la alta productividad de Visual Basic con el poder y la flexibilidad de C++.
- La misma aplicación que se ejecuta bajo Windows podría funcionar en un dispositivo móvil de tipo PDA.
- Es posible crear una gran variedad de aplicaciones en C#: aplicaciones de consola, aplicaciones para Windows con ventanas y controles, aplicaciones para la Web, etc.
- Mediante la plataforma .NET desde la cual se ejecuta es posible interactuar con otros componentes realizados en otros lenguajes .NET de manera muy sencilla.
- Es posible interactuar con otros componentes no gestionados fuera de la plataforma .NET.
- Desde C# es posible acceder a una librería de clases muy completa y muy bien diseñada, que permitió disminuir en gran medida el tiempo de desarrollo de la aplicación.

3.3.1.2 La plataforma de desarrollo .NET Framework¹⁹

El .NET Framework es la plataforma de desarrollo de código administrativo de Microsoft Corporation. Está formado por una serie de herramientas y bibliotecas con las que se pueden crear todo tipo de aplicaciones, desde las tradicionales aplicaciones de escritorio (WPF o Windows Forms) hasta aplicaciones para XBOX

¹⁹ Microsoft Corporation. Microsoft .NET Framework 3.5
Noviembre 20 2007[En línea].
< <http://www.microsoft.com/es-es/download/details.aspx?id=21>>

(XNA) pasando por desarrollo Web (ASP.NET), desarrollo para móviles (compact framework), aplicaciones de servidor (WPF, WCF), etc.

Dentro de los diversos recursos que componen la infraestructura (Figura 59), del .NET Framework, el más importante quizás es una maquina virtual conocida como CLR (*Common Language Runtime*), sobre la cual se ejecutan las aplicaciones. De este modo, los programas no contendrán código nativo de ningún microprocesador en particular, sino instrucciones MSIL (*Microsoft Intermediate Language*) que serán traducidas a código nativo en el momento de su ejecución (por medio de un compilador *Just In Time*).

De igual manera el Framework define una biblioteca de clases, BCL (*Base Class Library*), a la cual es posible acceder desde cualquier lenguaje desarrollado para la plataforma.

Por encima de la infraestructura se ubica un conjunto de reglas básicas que debe implementar un lenguaje para poder ser parte de la familia .NET. Esta especificación es conocida como CLS (*Common Language Specification*).

Finalmente, se encuentra el conjunto de lenguajes que cumplan con las especificaciones CLS, como el C#, el VB.NET, Managed C++, etc.

Es posible crear recursos en cualquiera de estos lenguajes que hagan uso de recursos escritos en otros; de hecho, es posible crear una clase en C# que herede de una clase creada en Managed C++.

Figura 59. Arquitectura de la infraestructura de desarrollo



3.3.1.3 Entorno de desarrollo

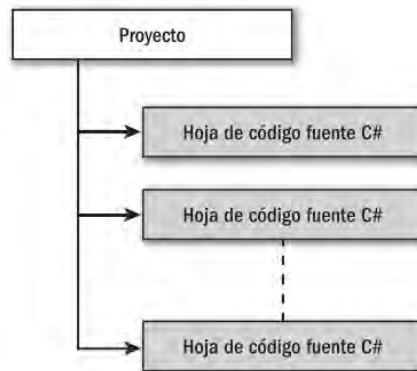
El entorno de desarrollo integrado utilizado para crear la aplicación grafica fue Visual Studio 2005, al ser un completo conjunto de herramientas para el desarrollo de aplicaciones Web ASP.NET, Servicios Web XML, aplicaciones de escritorio y aplicaciones móviles, las cuales podrán contener uno o más proyectos de diversos lenguajes entre los que se encuentran: Visual Basic, Visual C++, Visual C# y Visual J#, de igual forma estos lenguajes aprovechan las funciones de .NET Framework, lo cual proporciona un acceso a tecnologías clave para simplificar el desarrollo de aplicaciones.

En Particular, fue posible desarrollar la aplicación en C# gracias al .NET Framework 2.0, el cual permite compilar el código fuente C# para crear la aplicación como tal. Visual Studio 2005 hace uso del .NET Framework y funciona como *frontend*, evitando tener que interactuar con herramientas en el comando de línea. Sin embargo Visual Studio 2005, es mucho más que un simple *frontend*, a continuación se muestran algunas de las tareas que es posible llevar a cabo con este entorno de desarrollo:

- Navegar fácilmente por las clases por medio del visor de clases.
- Navegar por los archivos del proyecto, a través del explorador de soluciones.
- Extender más rápidamente el código escrito gracias a que el editor colorea las palabras reservadas y los tipos de datos conocidos.
- Organizar múltiples proyectos y editar fácilmente sus propiedades.
- Configurar el entorno para que ejecute herramientas externas (como precompiladores).
- Depurar fácilmente los proyectos y consultar valores de objetos de modo interactivo, así como realizar depuraciones remotas desde otras computadoras.
- Acceder a facilidades de búsqueda y remplazo por hoja de código fuente activo y en archivos.
- Editar recursos (bitmaps, iconos, archivos binarios, etc.) por medio de herramientas integradas, y navegar por ellos por medio del visor de recursos.
- Agregar tareas a realizar haciendo uso de la lista de tareas pendientes, que además inserta automáticamente tareas a partir de comentarios prefijados.
- Posibilidad de integrar herramientas al entorno por medio de un sistema de plug-ins.

Haciendo uso del entorno de programación Visual Studio 2005 es posible crear proyectos. Un proyecto contendrá, básicamente, hojas de código fuente en C#, tal y como se puede apreciar en la Figura 60.

Figura 60. Organización de un proyecto en Visual Studio 2005

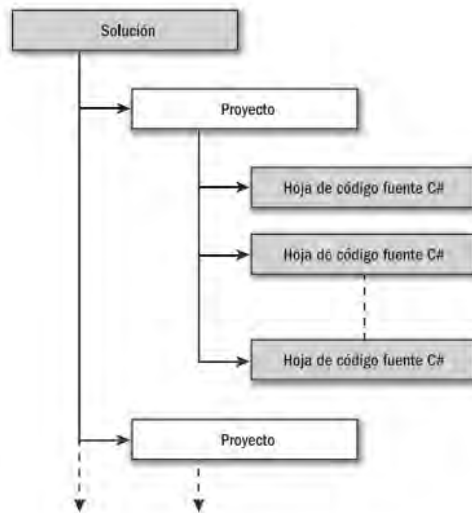


Un proyecto posee ciertas propiedades, las cuales indican de que manera deberán cumplirse las fuentes y los recursos que incluye. La idea de “proyecto” como organización de fuentes se encuentra muy extendida, y casi todo el entorno de programación la maneja. Usualmente, un proyecto construido tendrá como salida una aplicación ejecutable (archivo.EXE) o una librería (archivo.DLL). Sin embargo, es posible (y, de hecho, muy común) que el sistema en el cual se esté trabajando se encuentre integrado por más de un componente; por ejemplo, una aplicación y una librería de enlace dinámico que sea utilizada por la aplicación. En este caso, no es solo un proyecto el que se debe crear, sino dos.

Afortunadamente, Visual Studio 2005 soporta más de un proyecto abierto de manera simultánea en el mismo espacio de trabajo. Estos proyectos pueden relacionarse entre si por medio de dependencias; de modo que si la aplicación es reconstruida, el entorno automáticamente reconstruirá todos los proyectos dependientes que se hayan creado (por ejemplo, la biblioteca).

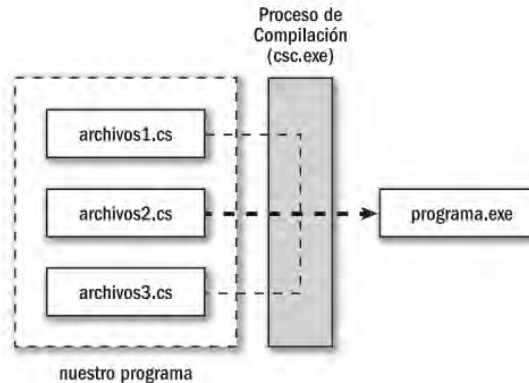
Visual Studio denomina solución a esta agrupación de proyectos (Figura 61). Todo proyecto debe estar construido en una solución; por tanto, cuando se crea un proyecto nuevo, Visual Studio automáticamente crea una solución que lo contiene.

Figura 61. Organización de una solución en Visual Studio 2005



Luego, cuando se construye la solución (compilación de un proyecto, Figura 62), automáticamente se crean los proyectos (que pueden estar relacionados directamente o no), es decir procesan todos los elementos de cada proyecto para generar los archivos de salida que correspondan.

Figura 62. Compilación de un proyecto



3.3.1.4 Visual Studio 2005 y .NET Compact Framework

Microsoft Visual Studio 2005 proporciona el tipo de proyecto de Smart Device para desarrollar aplicaciones para Pocket PC, Smartphone y otras plataformas basadas en Windows CE, además permite crear y probar las aplicaciones desarrolladas para este tipo de dispositivos mediante la tecnología de emulación sin dejar el entorno de desarrollo integrado (IDE) de Visual Studio 2005.

Es posible utilizar Visual Studio 2005 en el desarrollo de aplicaciones de .NET Compact Framework para dispositivos basados en las siguientes versiones de plataforma:

- Windows Mobile 2003 para Pocket PC.
- Windows Mobile 2003 para Smartphone.
- Windows CE 5.0 y 6.0.

De esta manera, el Entorno de Desarrollo Integrado Visual Studio 2005 permitió llevar a cabo el desarrollo de la aplicación para el sistema embebido MINI2440, teniendo en cuenta que este último soporta una plataforma Windows CE 6.0, la cual fue previamente montada (proceso descrito paso a paso en el primer documento del Anexo B, contemplado en la lista de anexos).

Cabe resaltar también que el .NET Compact Framework utilizado en el desarrollo de la aplicación ofrece funciones tales como:

- Ejecutar programas independientes del hardware y el sistema operativo.
- Admite protocolos de red comunes y se conecta perfectamente con servicios XML Web.
- Proporciona un modelo para orientar las aplicaciones y componentes ya sea a una amplia gama de dispositivos o a una categoría específica de estos.
- Facilita el diseño y la optimización de los recursos de sistema limitados.
- Obtiene un rendimiento óptimo en la generación de código nativo cuando se utiliza la compilación *Just-InTime* (JIT).

De esta manera .NET Compact Framework se convierte en un entorno independiente del hardware para la ejecución de aplicaciones en dispositivos de computación con limitaciones de recursos, entre los cuales se encuentran los PDA o Asistentes de Datos Personales, como los Pocket PC, teléfonos móviles, decodificadores de televisión, dispositivos de computación para automóviles y dispositivos embebidos de diseño personalizado, que estén integrados en el sistema operativo Windows CE .NET.

3.3.2 Estructura del software

3.3.2.1 Biblioteca de clases de .NET Compact Framework utilizadas

La biblioteca de clases de .NET Framework corresponde a una biblioteca de clases, interfaces y tipos de valor que se incluyen en el Kit de desarrollo de Software de Windows (SDK). Esta biblioteca brinda acceso a la funcionalidad del

sistema y es la base sobre la cual se crean las aplicaciones, los componentes y los controles de .NET Framework. Dentro de la amplia biblioteca de clases que proporciona el .NET Compact Framework, a continuación se mencionara únicamente las utilizadas en el desarrollo de la aplicación:

Using.System: El espacio de nombres System contiene clases fundamentales y clases base que definen tipos de datos de referencia y de valor de uso frecuente, eventos y controladores de eventos, interfaces, atributos y excepciones de procesamiento.

Using.System.Data: Proporciona acceso a las clases que representan la arquitectura de ADO.NET, la cual permite generar componentes que administran datos de varios orígenes de datos con eficiencia.

Using.System.IO.Ports: Contiene clases para controlar puertos serie, el espacio de nombres incluye enumeraciones que simplifican el control de los puertos serie, como Handshake, Parity, SerialPinChange y StopBits.

Using.System.Drawing: Proporciona acceso a funcionalidad de gráficos básica de GDI+, la clase Graphics proporciona métodos para dibujar en el dispositivo de pantalla. Clases como Rectangle y Point encapsulan primitivos de GDI+. La clase Pen se utiliza para dibujar líneas y curvas, mientras que las clases derivadas de la clase abstracta Brush se utilizan para rellenar el interior de las formas.

Using.System.Text: Contiene clases que representan codificaciones de caracteres ASCII, Unicode, UTF-7 y UTF-8; clases base abstractas para la conversión de bloques de caracteres en bloques de bytes y viceversa; y una clase auxiliar que manipula y da formato a objetos String sin necesidad de crear instancias intermedias de String.

Using.System.ComponentModel: Proporciona clases que se utilizan para implementar el comportamiento en tiempo de ejecución y tiempo de diseño de los componentes y controles. Este espacio de nombres incluye interfaces y clases base para implementar atributos y convertidores de tipos, enlazar a orígenes de datos y dotar de licencia a componentes.

Using.System.Collections.Generic: Contiene interfaces y clases que definen colecciones genéricas, permitiendo crear colecciones con establecimiento inflexible de tipos para proporcionar una mayor seguridad de tipos y un mejor rendimiento con respecto a las colecciones no genéricas

Using.System.Windows.Forms: Contiene clases para crear aplicaciones para Windows que aprovechan todas las ventajas de las características de la interfaz de usuario en el sistema operativo Microsoft Windows.

Using.System.Runtime.InteropServices: Proporciona una gran variedad de miembros que admiten la interoperabilidad COM y los servicios de invocación de plataforma.

3.3.2.2 Clases y métodos utilizados en el desarrollo de la aplicación

Dado que C# es un lenguaje preferentemente orientado a objetos, el software de la aplicación desarrollada estará compuesto por varias clases y métodos los cuales corresponden a las funciones que permiten llevar a cabo los diferentes procesos a través de la aplicación.

A continuación se presenta un resumen de las clases y métodos (contenidos en la biblioteca de clases descrita anteriormente), utilizadas en el desarrollo de la aplicación.

Serial Port: Proporciona un marco de trabajo para E/S síncrona y orientada a eventos, el acceso a los estados de punto de conexión e interrupción y el acceso a las propiedades del controlador serie. Se puede usar para ajustar objetos Stream, permitiendo que las clases que utilizan secuencias obtengan acceso al puerto serie.

Canvas: Define un área en la que pueden colocarse explícitamente objetos secundarios utilizando coordenadas, las cuales son relativas al área de trabajo.

Graphics: Esta clase proporciona métodos para dibujar objetos en el dispositivo de pantalla. Graphics se asocia a un determinado contexto de dispositivo.

Pen: Esta clase define un objeto utilizado para dibujar líneas y curvas. Un Pen dibuja una línea con el ancho y el estilo especificados. La línea dibujada por un Pen se puede rellenar con diversos estilos de relleno, como colores sólidos y texturas. El estilo de relleno depende del pincel o la textura que se utilice como objeto de relleno.

Brush: Define los objetos que se utilizan para rellenar formas de gráficos, por ejemplo rectángulos, elipses, gráficos circulares y rutas.

Font: Define un formato concreto para el texto, incluidos el nombre de fuente, el tamaño y los atributos de estilo. No se puede heredar esta clase.

Form: Representa una ventana o un cuadro de diálogo que constituye la interfaz de usuario de la aplicación, la clase Form se puede utilizar para crear ventanas estándar, sin bordes y flotantes, también puede utilizarse la clase Form para crear las ventanas modales como un cuadro de diálogo.

3.3.3 Funcionamiento del software

Un programa gráfico desarrollado en Visual Studio 2005, consta de una Aplicación y varios Formularios, que pertenecen a ella. Estos formularios son justamente las diferentes ventanas de la aplicación. Como en el caso del software desarrollado se tiene una sola ventana, por lo tanto tendrá un solo Formulario.

Al disparar la aplicación (JODORO_2440.exe), esta abre automáticamente el formulario por defecto (JODORO_2440_Form4.cs) que corresponde a la ventana que se observa al ejecutar el programa en el sistema embebido MINI2440.

3.3.3.1 Control de la interfaz gráfica

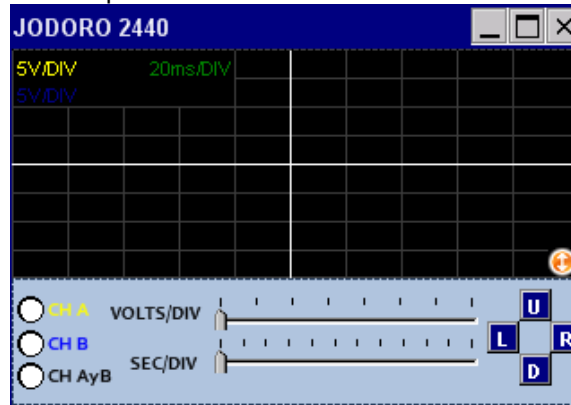
El desarrollo de la ventana se realiza (al igual que en cualquier aplicación grafica) creando un formulario, el cual es el encargado de controlar el funcionamiento de la ventana de la aplicación gráfica. En este formulario se ubican controles (botones, etiquetas, cuadro para ingresar texto, etc) en coordenadas específicas. Las coordenadas pueden darse en pixeles o en proporciones, lo cual permite que la ventana pueda mantener su aspecto al ser maximizada o cambiada de tamaño.

A los diferentes controles (por ejemplo, botones) se les define un comportamiento a través de eventos que son disparados cuando se realiza una acción sobre ellos (por ejemplo, pulsar un botón).

Al ser disparados, dichos eventos llaman a una función específica predefinida al crear el formulario.

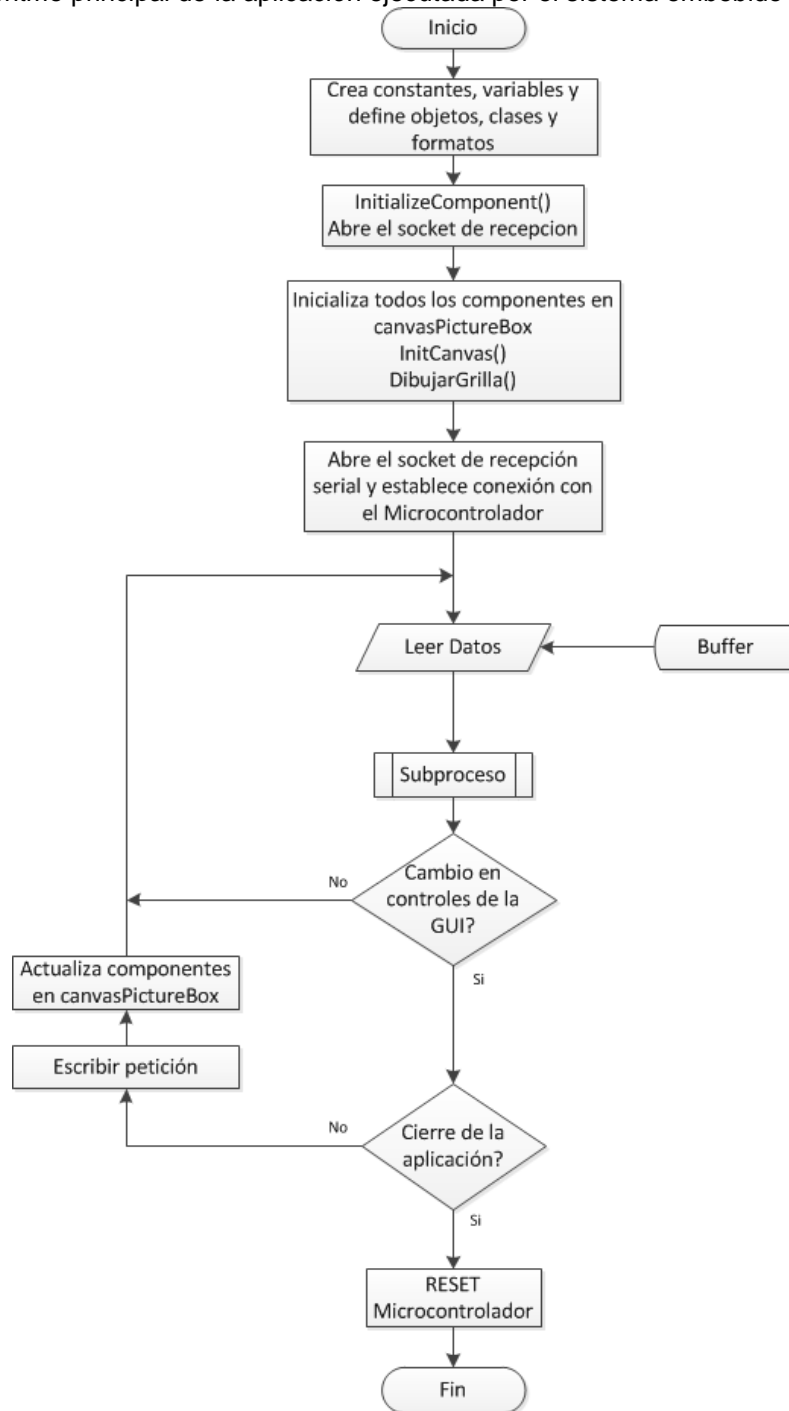
Asimismo, otros controles son de salida (por ejemplo, el display del osciloscopio), los cuales pueden ser modificados arbitrariamente desde el código aplicación a través de métodos que estos proveen (Por ejemplo, dibujar una línea, cambiar el color de fondo, etc.), en la Figura 63 se muestra un pantallazo de la interfaz grafica desarrollada para el sistema embebido MINI2440.

Figura 63. GUI desplegada en la pantalla del sistema embebido MINI2440



De igual forma, en la Figura 64 se muestra de manera general el algoritmo principal que ejecuta la aplicación desarrollada para el sistema embebido MINI2440, cuya función será llevar a cabo el procesamiento de los datos provenientes del microcontrolador, para posteriormente visualizar en pantalla la forma de onda reconstruida a partir de la señal análoga de entrada.

Figura 64. Algoritmo principal de la aplicación ejecutada por el sistema embebido MINI2440



El algoritmo de la aplicación ejecutada por el sistema embebido MINI2440, básicamente describe lo siguiente:

Se inicializan las variables y posteriormente se abre el socket que se encarga de recibir los datos provenientes de la placa de adquisición del dispositivo.

Una vez el socket establece conexión, se inicia la lectura asíncrona de datos por el puerto COM y se almacena la información en un buffer interno.

Posteriormente, el algoritmo ejecuta un subproceso (Figura 65), que se encargará de procesar y validar los datos previamente almacenados, dividiendo la secuencia de bits y convirtiendo los datos en coordenadas que serán representadas de forma gráfica.

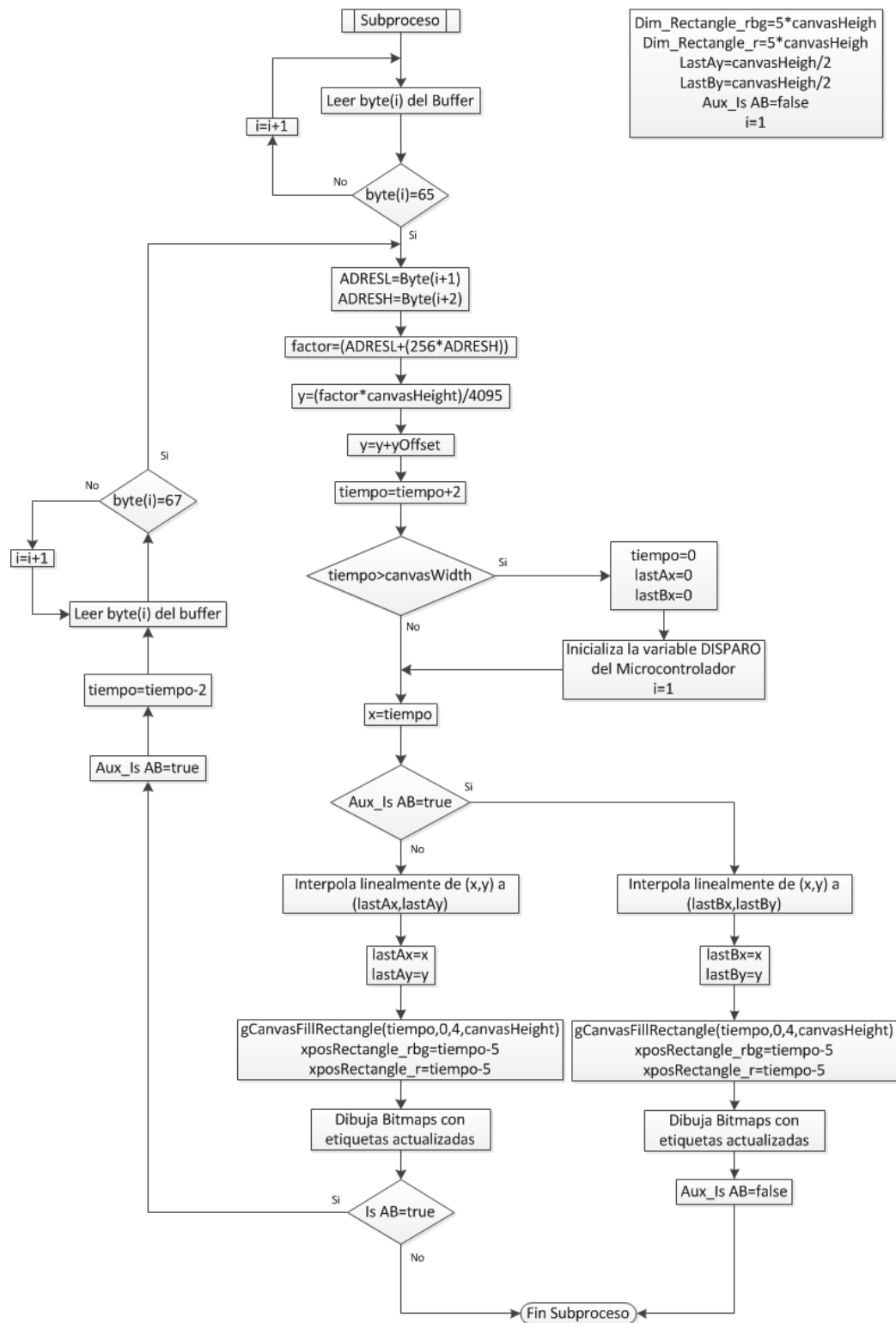
Antes de mostrar los puntos en pantalla, dentro del subproceso, se hace una comprobación para verificar si las nuevas coordenadas están fuera de los límites de la pantalla, en cuyo caso se resetee las coordenadas al origen.

En el proceso de dibujo, se renderizan las coordenadas correspondientes a los datos procesados al igual que los valores de la frecuencia y amplitud en la esquina superior izquierda.

Después de ejecutar el subproceso, se verifica si existe algún tipo de solicitud en los componentes de interfaz grafica de usuario, tales como: cambios en la escala de amplitud, cambios en la escala de tiempos, cambio de canal o cierre de la aplicación, los cuales luego de ser validados se procede a enviar dicha solicitud al socket.

Si no existe solicitud de cierre de la aplicación, se renueva el ciclo de petición de datos al socket, en caso contrario se transmitirá un comando que inicializa al Microcontrolador y posteriormente se envía una solicitud de cierre al socket, que daría por terminado el algoritmo.

Figura 65. Algoritmo del subproceso ejecutado por la aplicación gráfica implementada en el sistema embebido MINI2440.



3.4 CONSTRUCCION DEL DISPOSITIVO

3.4.1 Materiales y componentes utilizados

En este apartado se nombrara cada uno de los componentes que han sido utilizados para construir el osciloscopio.

- Protoboard: Utilizada para realizar las pruebas de hardware.
- Zócalos: utilizados para la inserción del microcontrolador PIC, multiplexor y amplificadores operacionales.
- Estaño: utilizado para soldar los componentes en la placa de topes.
- Resistencias y condensadores: Utilizados Principalmente en la implementación de la etapa análoga.
- 1 x Microcontrolador PIC18F4553.
- 4 x TL082CN: Amplificadores operacionales de alta precisión.
- 2 x CD4051: Multiplexor.
- Pertinax cobreado para impresos.
- Conectores macho y hembra tipo Molex.
- TRIMMERS: Resistencias variables del 1% de tolerancia.
- Cristal de cuarzo de 20Mhz.
- Cables Robbon.
- Cables de tipo banana y BNC-Banana.
- PC con Visual Studio 2005.
- PIC C Compiler para compilar programa hechos en assembler.
- MPLAB para programar el PIC.

3.4.2 Diseño de las placas PCB

Placa de fibra de vidrio (PCB): Esta placa, también llamada de circuito impreso, esta constituida de una placa aislante, de una o dos caras, de conductores planos metalizados cuya función es asegurar las conexiones eléctricas entre los componentes eléctricos conectados en la superficie.

Para realizar un montaje en placa impresa, se debe primero realizar, mediante un programa de ordenador (ARES en nuestro caso), el diseño del circuito, pensando donde poner cada componente para que las pistas que conecten dichos componentes no se entrecrucen. Una vez realizado todo el diseño y revisado cuidadosamente, se procede a imprimirlo (en impresora laser), sobre en una hoja de papel satinado al tamaño de la plaqueta. Posteriormente se fija a la plaqueta con pequeños cortes de cinta, la hoja de papel impresa, de manera que quede

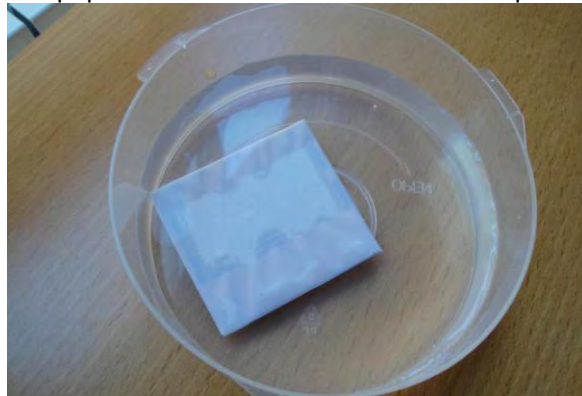
enfrentada al cobre, ya que la cara sin impresión es la que permite la transferencia de calor de la plancha. La cinta es necesaria para evitar que el papel se mueva durante el planchado (Figura 66).

Figura 66. Método de planchado



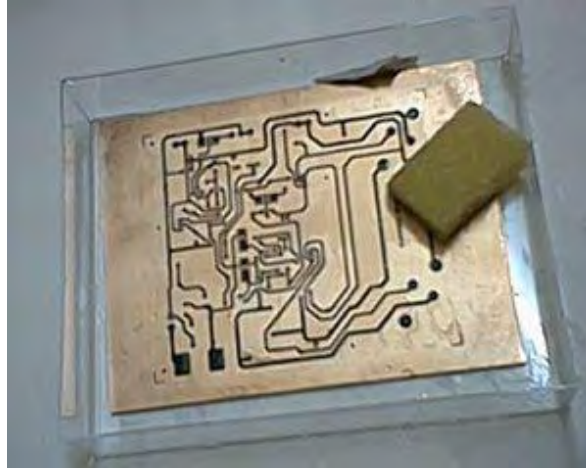
Una vez que las pistas están marcadas en el papel, se introduce inmediatamente la placa con el papel satinado pegado a un recipiente con agua (Figura 67).

Figura 67. Placa adherida al papel satinado e introducida en un recipiente con agua



Al cabo de unos 20 minutos, el papel ya húmedo formara arrugas entre pistas, indicando que se debe proceder a despegarlo, presionando suavemente con una esponja con el fin de deshacerlo. Una vez retirado todo el papel (Figura 68), se procede a enjuagar la placa con abundante agua, eliminando los restos de papel que puedan haber quedado entre pistas y de este modo la plaqueta ya estará lista para ser atacada con el cloruro Férrico.

Figura 68. Placa después de retirar el papel



Este sistema de hacer pistas es un poco lento, y quizá no tan profesional, pero tiene a su favor que no se necesita ningún tipo de dispositivo para el proceso. Además, los resultados han sido realmente buenos.

Las Figuras presentadas a continuación (Figura 69, 70, 71 y 72), corresponden a las fotografías de las placas finales, antes y después de soldar los diferentes componentes.

Figura 69. Placa de la tarjeta de adquisición de datos, después de retirar el cobre

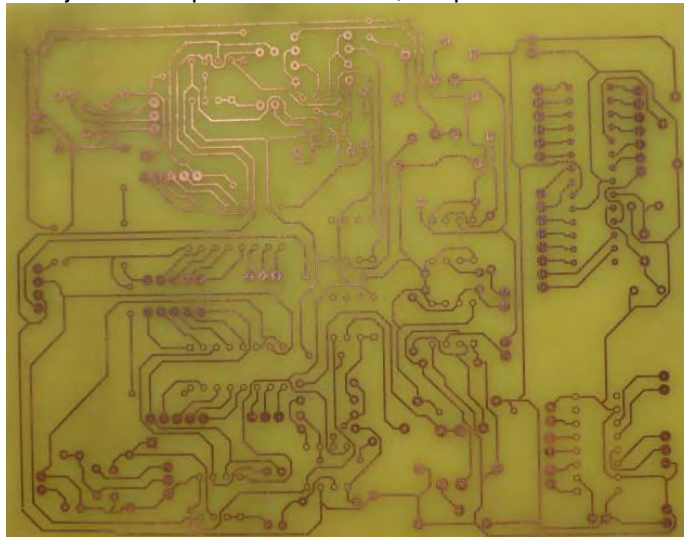


Figura 70. Placa de la fuente de alimentación después de retirar el cobre

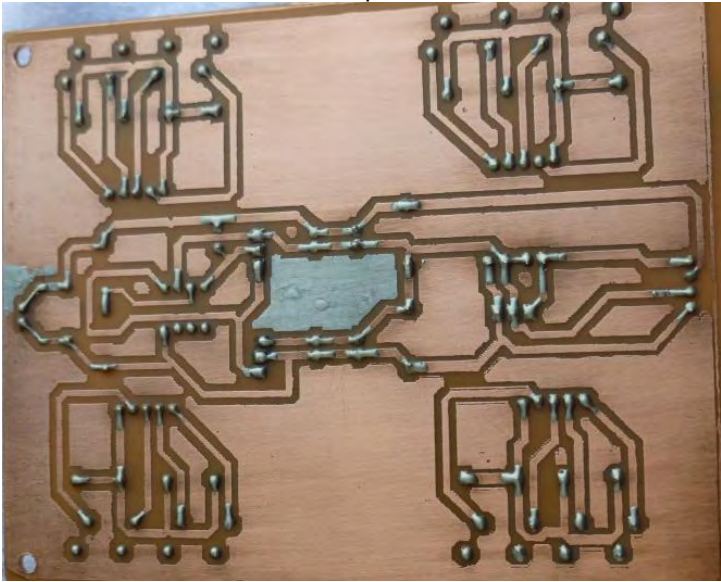


Figura 71. Placa tarjeta de adquisición de datos soldada.

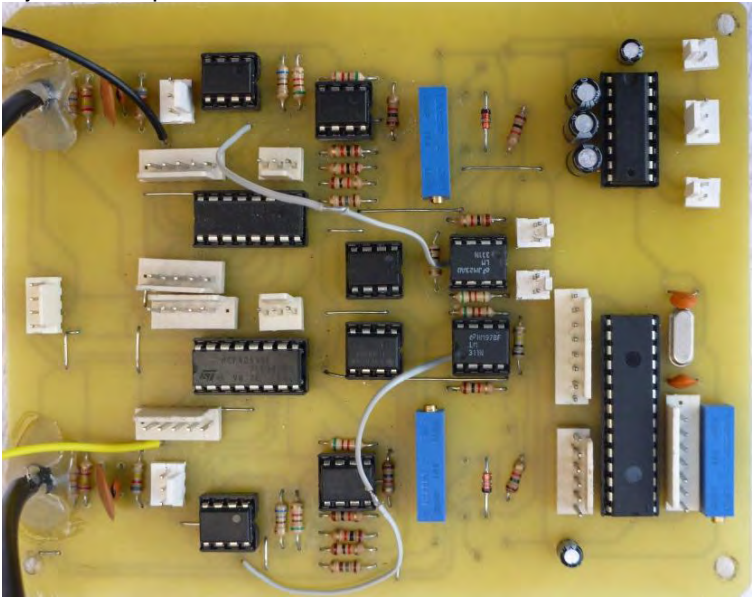
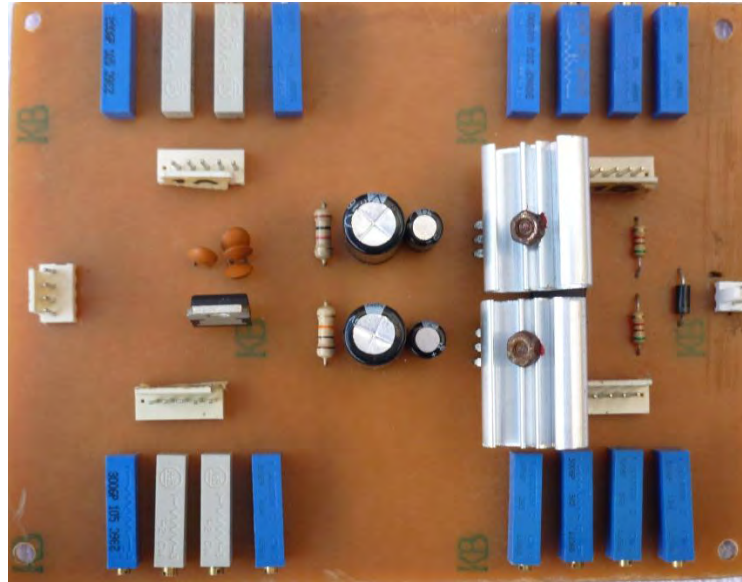


Figura 72. Placa de la fuente de alimentación soldada



3.4.3 Diseño y construcción de la carcasa

El diseño de la carcasa ha sido modelado mediante la herramienta AutoCAD, logrando así ubicar de manera correcta cada una de las placas y piezas internas con las cuales cuenta el dispositivo final (conectores, PCB tarjeta de adquisición de datos, PCB fuente de alimentación eléctrica y el sistema embebido MINI 2440), posterior a ello se realizó un montaje en cartón paja de dicho modelado, para observar la disposición de los elementos en una forma tangible, logrando así depurar el diseño inicial.

La estructura de la carcasa final (Figura 73, 74 y 75), esta conformada en su totalidad por láminas de plástico, debido a que este material permite al dispositivo contar con mejores características en cuanto a flexibilidad, bajo costo, liviano y maleable.

Figura 73. Vista interna de la carcasa del dispositivo



Figura 74. Vista frontal de la carcasa del dispositivo

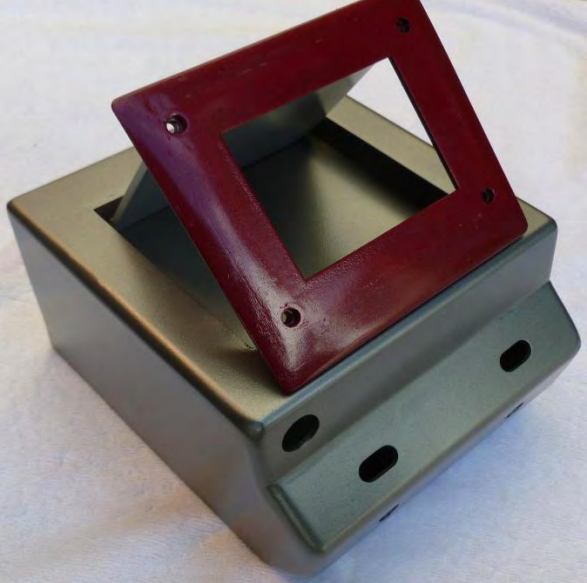


Figura 75. Vista posterior de la carcasa del dispositivo



3.4.4 Presupuesto total del equipo

En la Tabla 46 se presenta el presupuesto total para el desarrollo del dispositivo.

Tabla 46. Presupuesto total del dispositivo

Componente	Modelo	Cantidad	Precio/Unidad(\$)	Total
Amplificadores Operacionales	TL082	8	1200	\$9.600
Multiplexor Digital	CD4051	2	800	\$1.600
Microcontrolador	PIC 18F4553	1	15.000	\$15.000
Trimmers		18	1.300	\$23.400
Diodo	1N4007	3	200	\$600
Resistencias	CARBON ¼ w Tolerancia del 1%	30	100	\$3.000
Condensador Electrolítico	COND ELEC RAD 105°C 10UF 50V	3	200	\$600
Condensador Electrolítico	COND ELEC RAD 105°C 2200UF 50V	2	700	\$1.400

Condensador Electrolítico	COND ELEC RAD 105°C 4,7UF 50V	2	350	\$700
Condensadores cerámicos	COND CER NP PRECISION 10PF	8	100	\$800
LM7805	POSITIVE VOLTAGE REGUALTOR 5V 1A TO-220	1	750	\$750
Cristal 48Mhz	XTAL 48000K	1	3.500	\$3.500
Placa de fibra de vidrio	PFP23(140X240 mm)	1	20.000	\$20.000
Conector Macho	Molex	23	250	\$5.750
Conector Hembra	Molex	23	250	\$5.750
Transistor NPN	TIP41	1	700	\$700
Transistor PNP	TIP42	1	700	\$700
Conector Macho	BNC	2	550	\$550
Conector Hembra	BNC	2	550	\$550
Cable Ribbon	Ribbon	2(mts)	2.200	\$2.200
Kit Placa de desarrollo MINI2440	FriendlyARM MINI2440	1	220.000	\$220.000

TOTAL	\$316.4500
--------------	-------------------

4. RESULTADOS

4.1 CARACTERIZACIÓN DEL EQUIPO

En este capítulo se expone la caracterización del osciloscopio, que incluye una Tabla de las características eléctricas y otras características típicas de los osciloscopios, como es la frecuencia de muestreo que utiliza. La finalidad de este capítulo es poder apreciar fácilmente las características del osciloscopio y hacer algún tipo de comparación con los osciloscopios que se encuentran en el mercado.

4.1.1 Características eléctricas

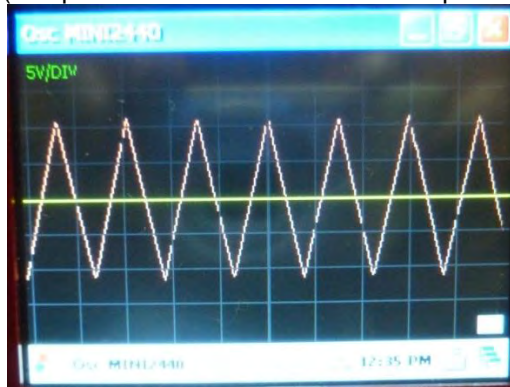
Tabla 47. Características eléctricas del Osciloscopio

CARACTERISTICAS ELECTRICAS OSCILOSCOPIO	
Ancho de banda	250 KHz
Canales	2
Trigger Externo	No
Velocidad de respuesta	1V/0.4uS
Resolución Vertical	12 Bits
Sensibilidad Vertical	1.22mV
Precisión vertical en nivel DC	5% en 5V/div
Entrada Máxima de Voltaje	40Vpp
Entrada Mínima de Voltaje	20mVpp
Rango de Posición	20mV a 5V
Entrada de Acople	AC, DC, GND
Impedancia de entrada	1M Ω 20pF

4.1.2 Linealidad

Una característica importante para conocer el comportamiento de nuestro osciloscopio es la linealidad. Para comprobar que la linealidad es correcta, se introduce una señal triangular y se visualiza en el osciloscopio (Figura 76). De esta forma, se puede ver si la señal es totalmente lineal o no.

Figura 76. Señal triangular (comprobación de linealidad en el dispositivo)

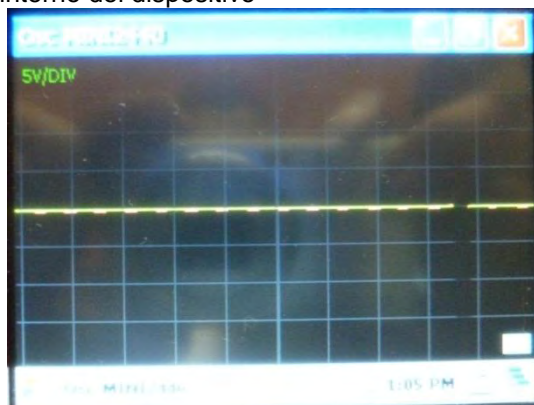


Como se puede observar en la imagen anterior, la señal es suficientemente lineal, ya que la señal en los montes y valles no ha sido deformada.

4.1.3 Ruido

El ruido introducido por el propio dispositivo es también un aspecto importante a tener en cuenta. En este sentido, un buen osciloscopio presentara un nivel de ruido interno mínimo o en algunos casos hasta nulo, ya que de lo contrario influiría en la medición de las señales de entrada, visualizando así formas de onda erróneas en pantalla. Con el fin de verificar el nivel de ruido del propio dispositivo, se conecto uno de los 2 canales de entrada a tierra del dispositivo, visualizando en pantalla la señal presentada en la Figura 77, de la cual se puede concluir que el nivel de ruido interno del dispositivo es mínimo.

Figura 77. Señal de ruido interno del dispositivo



4.2 PRUEBA DE FUNCIONAMIENTO DEL DISPOSITIVO

Para poner a prueba el funcionamiento del dispositivo, se utilizó un generador de señales de referencia Protek-9205C, previamente configurado en cuanto a la amplitud, frecuencia y forma de onda de su señal de salida. Posterior a ello se conectó dicha señal tanto al osciloscopio digital de referencia UNIT-UT2202C como el prototipo de osciloscopio digital desarrollado.

A continuación se muestran imágenes de una señal sinusoidal (Figura 78), triangular (Figura 79) y cuadrada (Figura 80) visualizadas simultáneamente en los dos equipos.

Figura 78. Señal sinusoidal visualizada en los 2 equipos

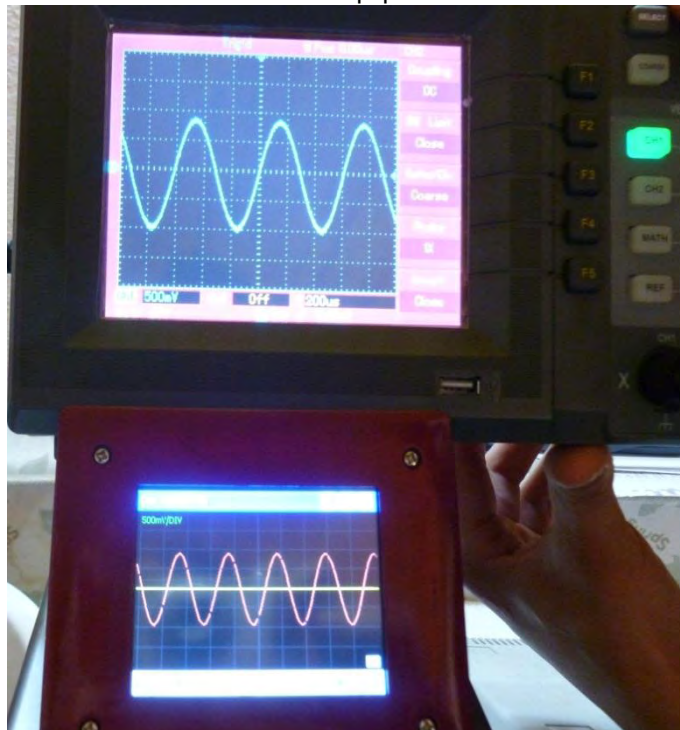


Figura 79. Señal triangular visualizada en los 2 equipos

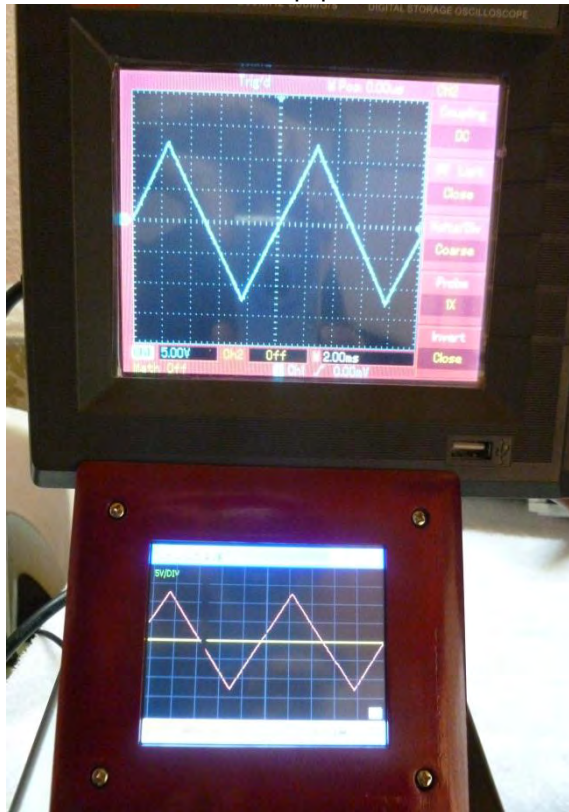
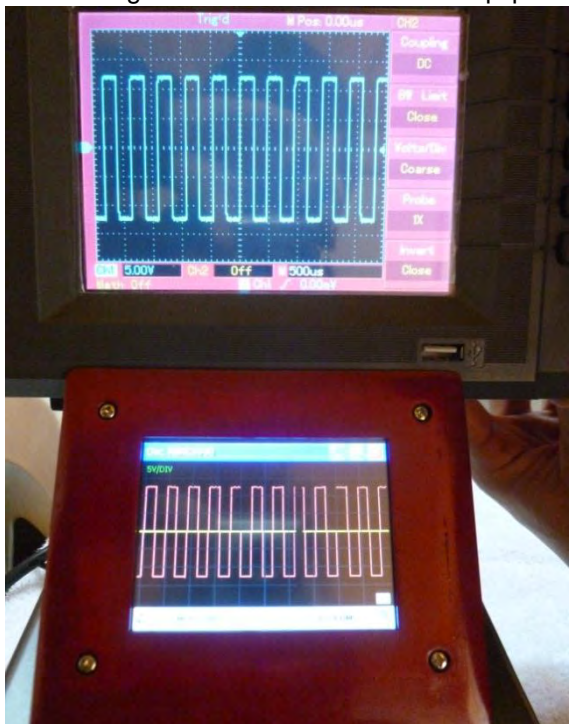


Figura 80. Señal cuadrada o rectangular visualizada en los dos equipos



Durante las pruebas de funcionamiento, también se realizaron mediciones simultáneas de señales de diferente frecuencia y amplitud, provenientes de un generador de señales, previamente configurado.

En este sentido, la Figura 81 muestra la medición simultánea por parte de los 2 dispositivos, de una señal sinusoidal de baja frecuencia y gran amplitud, mientras que en la Figura 82 se observa la medición simultánea de la misma forma de onda que la anterior pero esta vez con una mayor frecuencia y menor amplitud.

Figura 81. Señal sinusoidal, de baja frecuencia (101Hz) y con Amplitud de 12Vpp

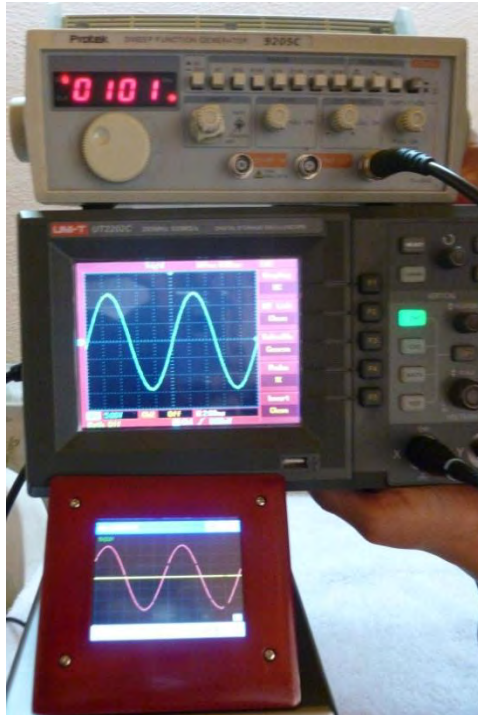
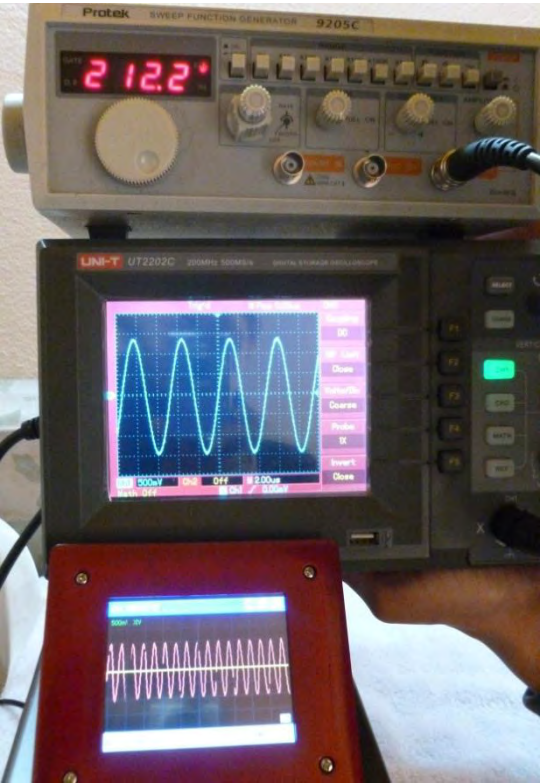


Figura 82. Señal sinusoidal, de alta frecuencia (212.2 KHz) y con Amplitud de 3Vpp



5. CONCLUSIONES

- Se desarrolló un dispositivo innovador, denominado “JODORO 2440”, el cual incorpora una pantalla LCD touchscreen de 3,5”, para la visualización de señales eléctricas en tiempo real, mediante una Interfaz Grafica de Usuario (GUI), que permite también llevar a cabo un control y configuración interactivo sobre los diferentes parámetros de medida del dispositivo.
- Utilizando un lenguaje de programación en Visual C#, se desarrolló una aplicación gráfica para el sistema embebido MINI2440, la cual corresponde a la GUI desplegada en la pantalla del dispositivo final.
- La Facultad de Ingeniería Electrónica de la Universidad de Nariño, cuenta con un documento que describe detalladamente los pasos para lograr construir un entorno de desarrollo adecuado que permita poner en marcha el diseño y posterior implementación de cualquier tipo de aplicación sobre el sistema embebido MINI2440, basada en el .NET CF de Microsoft.
- Se logró diseñar e implementar una etapa previa al sistema embebido MINI2440, el cual consiste básicamente en 2 canales independientes que permiten la adquisición, adaptación y conversión de las señales eléctricas de entrada, cuya amplitud y frecuencia no superen los 40 Vpp y 250 KHz respectivamente.
- El desarrollo de este dispositivo, brinda a la comunidad académica del programa de Ingeniería Electrónica, un elemento didáctico, que puede ser utilizado en diferentes asignaturas como: Circuitos eléctricos, Circuitos digitales, Electrónica de potencia, entre otras, contribuyendo así al avance en la calidad académica del programa y por ende al de la Universidad de Nariño.
- El desarrollo del proyecto conllevó a la adaptación de conceptos en cuanto al uso de nuevas tecnologías a través del sistema embebido MINI2440, debido a que se logró llevar a cabo un procesamiento digital de señales utilizando ciertos recursos con los cuales cuenta el sistema embebido en particular.

BIBLIOGRAFIA

AnalogDevices. Hoja de datos AD 620. [En línea]
< http://www.stanford.edu/class/ee122/Parts_Info/datasheets/ad620.pdf>
[Citado junio 30 de 2012].

Camargo B. Carlos Iván. Transferencia tecnológica y de conocimientos en el diseño de sistemas embebidos. [En línea]
<http://www.bdigital.unal.edu.co/5696/1/carlosivancamargobare%C3%B1o_2011.pdf>
[Citado junio 12 de 2012].

Echeverria Senosian, Javier. Desarrollo de una interfaz grafica con pantalla táctil para el manejo de un sistema robótico. [En línea]
<<http://academicae.unavarra.es/bitstream/handle/2454/4623/577669.pdf?sequence=1>>
[Citado julio 15 2012].

FRANCO, Sergio Design with Operational Amplifiers and Analog Integrated Circuits, 3a. ed. New York: McGraw-Hill, 2002. P. 17-19.

FriendlyARM MINI2440. Manual de usuario. [En línea]
< http://www.friendlyarm.net/dl.php?file=mini2440_manual_20110421.pdf>
[Citado junio 20 de 2012].

Fundación OPTI. Fundación ASCAMM. Tendencias y aplicaciones de los Sistemas Embebidos en España “Estudio de prospectiva”. España, Septiembre 2009. [En línea]
<<http://www.opti.org/publicaciones/pdf/texto131.pdf>>
[Citado 11 mayo 2012].

Hassan Houcine, Dorian Javier, Montagud Javier, Domínguez Carlos. Instrucción en el diseño de sistemas empotrados. Aplicación al control de un brazo robot. Departamento de Informática de Sistemas y Computadores (DISCA), Universidad Politécnica de Valencia (UPV). Valencia, España 2003. [En línea]
<<http://www.ceautomatica.es/old/actividades/jornadas/XXIV/documentos/econ/90.pdf>>
[Citado febrero 28 de 2012].

LAPLANTE, A. Real-time systems design and analysis. An engineer's handbook. IEEE Press. 1992.

Mark Weiser. The Computer for the 21st Century. [En línea]
<<http://www.ubiq.com/hyertext/weiser/Sci-AmDraft3.html>>
[Citado marzo 13 de 2012].

Mat Taib, Norshazwana Binti. Portable Heart Beat Monitor Unit Using FriendlyArm MINI2440, Universidad Tecnológica de Malasya, Salesiana 2011. [En línea].
<http://psm.fke.utm.my/libraryfke/files/1563_NORSHAZWANABINTIMATTAIB2011.pdf>
[Citado junio 16 2012].

Microchip. Hoja de datos Microcontrolador PIC18F2553. [En línea]
<<http://pdf1.alldatasheet.com/datasheet-pdf/view/197486/MICROCHIP/PIC18F2553.html>>
[Citado junio 10 de 2012].

Microchip. Hoja de datos Microcontrolador PIC16F877A. [En línea]
<<http://ww1.microchip.com/downloads/en/devicedoc/39582b.pdf>>
[Citado febrero 10 de 2012].

Programa Gacela-Secretaria de Economía de México. Fundación FUMEC. Programa de innovación en el sector de los sistemas embebidos. México, Agosto 2009[En línea]
<<http://fumec.org.mx/v5/htdocs/embebidos.pdf>>
[Citado 12 abril de 2012].

Ugelstadt, Thomas. Op Amps for Everyone: Active Filter Design Techniques. Texas Instruments. p.6 [en línea].
<<http://www.ti.com/lit/ml/sloa088/sloa088.pdf>>
[Citado mayo de 2012].

W.A, Halang & K.M. Sacha.Real-time systems: Implementation of industrial computerised process automation. Ed. World Scientific, 1992.

Zabalza M, Fernando. Representación de variables eléctricas en el sistema embebido MINI2440. Escuela Técnica Superior de Ingenieros Industriales y de Telecomunicaciones, Universidad Pública de Navarra, Pmplona 2011. [En línea].
<<http://academicae.unavarra.es/bitstream/handle/2454/4283/577623.pdf?sequence=1>>
[Citado junio 13 de 2012].