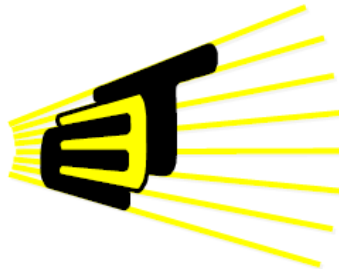


**REDISEÑO DE LA INTERFAZ GRÁFICA PARA EL EQUIPO DE EVALUACIÓN
ULTRASÓNICA SONIC 137**



**EDWARD ANDRÉS GARCÍA REMOLINA
PEDRO SIZA MORENO**



**UNIVERSIDAD INDUSTRIAL DE SANTANDER
FACULTAD DE INGENIERÍAS FÍSICO – MECÁNICAS
ESCUELA DE INGENIERÍAS ELÉCTRICA, ELECTRÓNICA Y
TELECOMUNICACIONES
BUCARAMANGA
2009**

**REDISEÑO DE LA INTERFAZ GRÁFICA PARA EL EQUIPO DE EVALUACIÓN
ULTRASÓNICA SONIC 137**

**EDWARD ANDRÉS GARCÍA REMOLINA
PEDRO SIZA MORENO**

Trabajo de grado para optar el título de Ingeniero Electrónico

**Director
MPE. JAIME BARRERO PEREZ**

**Codirectora
MSC. LUZ AMPARO QUINTERO ORTIZ**

**UNIVERSIDAD INDUSTRIAL DE SANTANDER
FACULTAD DE INGENIERÍAS FÍSICO – MECÁNICAS
ESCUELA DE INGENIERÍAS ELÉCTRICA, ELECTRÓNICA Y
TELECOMUNICACIONES
BUCARAMANGA
2009**

A mis padres y hermanas

PEDRO SIZA MORENO

A mis padres y a mi tía

EDWARD ANDRÉS GARCÍA REMOLINA

AGRADECIMIENTOS

Agradezco a Dios y a mis padres, que siempre están conmigo y me han brindado la oportunidad de contar con una excelente educación, este apoyo se convirtió en la razón por la cual plantee mis metas, que ahora son una realidad.

Agradezco a Msc. Luz Amparo Quintero, por brindarnos su colaboración y a Mpe. Jaime Barrero Pérez por la asesoría que permitió llevar a cabo el presente proyecto, a todos mis compañeros y demás personas que siempre me acompañan brindándome su apoyo.

PEDRO SIZA MORENO

Agradezco a mis padres y a mi tía que desde pequeño creyeron en mis capacidades, brindándome su apoyo en los momentos más difíciles de mi vida universitaria, colaborándome e incentivándome para cumplir mis sueños.

EDWARD ANDRÉS GARCÍA REMOLINA

TABLA DE CONTENIDO

	pág.
INTRODUCCIÓN	6
1. ONDAS DE ULTRASONIDO	8
1.1 Definición	8
1.2 Principio de Medición por Ondas de Ultrasonido	10
1.3 Angulo de Reflexión y Refracción	11
1.4 Método de Detección de Fallas por Ultrasonido	12
1.5 Transductores Ultrasónicos	12
1.5.1 Materiales piezoeléctricos	13
1.5.1.1 Cuarzo	13
1.5.1.2 Sulfato de Litio	13
1.5.1.3 Cerámicos polarizados	13
1.5.2 Transductores Utilizados en la Detección de Fallas	13
1.5.2.1 Transductores de contacto	13
1.5.2.2 Transductores de Haz Angular	14
1.5.2.3 Transductores de Línea de Contacto	14
1.5.2.4 Transductor de Inmersión	14
1.5.2.5 Transductor de Doble Elemento	14
1.5.3 Material Acoplante	15
2. SONIC 137 Y PRIMERA VERSION DE LA INTERFAZ	16
2.1 Descripción del Sonic 137	16
2.2 Primera Versión de la Interfaz	18
3. DESCRIPCIÓN DE COMPONENTES Y FUNCIONES DEL SONIC	19
3.1 Componentes para la Visualización	19
3.2 Comandos de Ejecución Inmediata	22
3.2.1 Vista Ampliada “View”	22
3.2.2 Aumento de Ganancia “+dB”	22
3.2.3 Guardar Valor de Espesor “Save Thick”	22
3.2.4 Guardar Gráfica “Save Ascan”	22
3.2.5 Opción del DAC “Profile”	23
3.3 Comandos del Menú General	23
3.3.1 Configuración del Pulso Eco “Pulser”	23
3.3.2 Ganancia “Gain”	23
3.3.3 Compuertas para Detección de Fallas “Gate”	24
3.3.4 Configura Rango, Retardos y Velocidad del Material “Range”	24
3.3.5 Configuración de Alarmas y Medición de Espesor “Thick”	24
3.3.6 Haz Angular “Angle Beam”	25
3.3.7 Corrección de Distancia de Amplitud “DAC”	25

3.3.8 Almacenamiento e Informes “Setup”	25
3.3.9 Recuperar Datos Almacenados “Recall”	26
3.3.10 Configuración de Hora y Fecha “Spcl”	26
4. DISEÑO INTERFAZ EN LABVIEW	27
4.1 Identificación de Requisitos	27
4.2 Interesados	28
4.3 Descripción de Requisitos	28
4.3.1 Comunicación en Labview	28
4.3.1.1 Visa	28
4.3.1.1.1 Visa Serial	29
4.3.1.1.2 Visa Write	29
4.3.1.1.3 Visa Read	29
4.3.1.1.4 Visa Close	30
4.3.2 Lectura de Datos del Sonic	30
4.3.3 Escritura de Datos	31
4.3.4 Revisión de Cambios del Menú	32
4.3.5 Navegación en el Menú del Sonic	33
4.3.6 Visualización de la Gráfica	34
4.3.7 Determinación de la Ubicación de los Ecos	35
4.3.8 Revisión de Comandos de Ejecución Inmediata	36
4.3.9 Activación de la compuerta “GATE 2”	37
4.3.10 Activación del Display de Espesor del Sonic (“T-GAUGE”)	38
4.4 Organización del Sistema	38
4.5 Resultado del Diseño	40
5. CONVERTOR RS232-USB	41
5.1 Teoría de Comunicación USB	41
5.1.1 Bus Serial Universal (USB)	42
5.1.2 Origen	42
5.1.3 Ventajas	43
5.1.4 Aplicación	43
5.1.5 Características	44
5.1.6 Dispositivos	44
5.1.7 Transmisión	45
5.2 Circuito integrado FT232BM USB UART (usb-serial)	46
5.3 Comunicación Serial	46
5.3.1 Puerto serie RS232	48
5.3.2 Conector DB9	49
5.4 Circuito Integrado MAX232	51
5.4.1 Funcionamiento	52
5.5 Diagrama Esquemático y PCB conversor RS232-USB	53

5.6	Instalación de los Drives para Conversor	53
6.	PRUEBAS FINALES	54
6.1	Calibración del SONIC 137	54
6.2	Pruebas con detección de fallas	57
6.3	Medición con haz Angular	
	60	
7.	TARJETA DE ADQUISICIÓN DE DATOS	61
7.1	Programación del PIC 18F2550	61
7.1.1	Compilador PIC	61
7.1.2	Herramienta Para el Programador	63
7.1.3	Grabación al PIC	64
7.2	Características Generales	65
7.2.1	Sistema Operativo	65
7.2.2	Aplicación Para el Control de la Tarjeta TAD_EP	65
7.3	Funcionamiento General de la Aplicación	66
7.3.1	Conexión	66
7.3.2	Interfaz	66
7.3.3	Descripción Del Programa En Labview	66
7.3.4	Manejo De La Interfaz	71
8.	CONCLUSIONES Y OBSERVACIONES	77
	BIBLIOGRAFÍA	82
	ANEXOS	

LISTA DE TABLAS

Tabla 1. Datos Técnicos para Configurar el Puerto Serial	17
Tabla 2. Descripción de pines USB	42
Tabla 3. Descripción de pines conector DB9	50
Tabla 4. Valores de espesor de la escalerilla	56
Tabla 5. Valores de Referencia Ofrecidos por el Fabricante	58
Tabla 6. Valores Tomados con la Calibración Realizada en la Interfaz	58
Tabla 7. Valores Medidos con el Sonic	58

LISTA DE FIGURAS

Figura 1. Ondas longitudinales y Ondas Transversales	9
Figura 2. Dispersión de las Emisiones del transductor	10
Figura 3. Haz de incidencia y reflacción	11
Figura 4. Ejemplo de Detección de Fallas	12
Figura 5. Transductor de contacto	13
Figura 6. Transductor de Haz Angular	14
Figura 7. Fotografía del Sonic 137	16
Figura 8. Primera Versión de Interfaz para el Sonic 137	18
Figura 9. Pantalla CRT	20
Figura 10. Menú Display	20
Figura 11. Indicador de Espesor <i>"Thickness Display"</i>	20
Figura 12. Esquema de Programación para la Lectura de Datos	30
Figura 13. Vector Tipo String de Datos Recibidos	31
Figura 14. Esquema de Programación para Lectura y Actualización de Perillas	31
Figura 15. Esquema de Programación para la Escritura de valores al Sonic	32
Figura 16. Esquema de Programación para la Visualización del Menú Pulser	33
Figura 17. Botones de Navegación y Perilla de Variación	33
Figura 18. Menú e indicador de comando seleccionado	34
Figura 19. Esquema de Programación Visualización de la Gráfica	35
Figura 20. Indicador del Espesor Determinado por la ubicación de los ecos	35
Figura 21. Esquema de Programación Para Determinar la Ubicación de los Ecos	36
Figura 22. Esquema de Programación para Detectar la Pulsación de los Comandos de Ejecución Inmediata	36
Figura 23. Visualización de la Compuerta GATE 2 y BLOCKING GATE	37
Figura 24. Esquema de Programación Activación de la Compuerta "GATE 2"	37
Figura 25. Diagrama de Flujo del Diseño de la Interfaz	39
Figura 26. Imagen Final del Diseño de la Segunda Versión de la Interfaz	40
Figura 27. Fotografía del Conversor Serial- USB	41
Figura 28. Típicas funciones de un sistema USB	45
Figura 29. Esquema de la transmisión asíncrona	45
Figura 30. Transmisión sincrónica	46
Figura 31. Conector DB9 macho y hembra	49
Figura 32. Integrado MAX232 con sus Condensadores Externos	51
Figura 33. Niveles norma RS-232	52
Figura 34. Imagen Pruebas de Calibración	54
Figura 35. Visualización del Espesor de Menor Valor	54
Figura 36. Visualización del Espesor de Mayor Valor	55
Figura 37. Ubicación de la Compuerta en el Menor Espesor	55
Figura 38. Ubicación de la Compuerta en el Mayor Espesor	55
Figura 39. Visualización del Espesor del Material	56
A) Sonic 137	

B) Interfaz Gráfica

Figura 40. Escalerilla utilizada para la calibración	56
Figura 41. Detección de Falla en Placa de Acero	57
Figura 42. Esquema de las Fallas en la Placa de Acero	57
Figura 43. Escalerilla de Acero no Caracterizada	58
Figura 44. Detección de Falla con Escalerilla	59
Figura 45. Gráfica del Sonic sin presencia de falla	59
Figura 46. Gráfica del Sonic Cuando se presenta la falla	59
Figura 47. Decaimiento Exponencial de la Amplitud de la falla	60
Figura 48. Pruebas con Transductor de Haz Angular	60
Figura 49. Grafica Haz Angular	60
Figura 50. Secuencia de programación	61
Figura 51. Entorno programa WinPic800	63
Figura 52. Instaladores y Archivos para hacer Uso de Tarjeta	66
Figura 53. <i>SubVI</i> para Leer y Escribir por USB	67
Figura 54. Secuencia de Escritura al Micro entrada digital	68
Figura 55. Secuencia de Lectura del Micro Entrada Digital	68
Figura 56. Representacion de las entradas	68
Figura 57. Secuencia de Escritura al Micro Entrada Analógica	69
Figura 58. Secuencia de Lectura del Micro Entrada Analógica	70
Figura 59. Secuencia de Escritura de Datos al microcontrolador	70
Figura 60. Contenido Carpeta del Ejecutable	71
Figura 61. Vista Principal Interfaz en Labview 8.2 (Licencia Evaluación)	71
Figura 62. Indicador de Entradas Digitales Labview 8.2 (Licencia Evaluación)	72
Figura 63. Visualizador Entrada Analógica Labview 8.2 (Licencia Evaluación)	72
Figura 64. Control de Salidas Digitales Labview 8.2 (Evaluación)	73
Figura 65. Control Para Detener la Aplicación	73
Figura 66. Vista Principal Interfaz en Labview 7.1	74
Figura 67. Indicador Entradas Digitales Labview 7.1	75
Figura 68. Visualizador Entrada Analógica Labview 7.1	75
Figura 69. Control de Salidas Digitales Labview 7.1	76
Figura 70. Visualización del valor de Espesor a partir de la Ubicación de los Ecos	84
Figura 71. Ejemplo de la Compuerta en el Menor Espesor	85
Figura 72. Ejemplo de la Compuerta en el Mayor Espesor	85
Figura 73. Indicador de Espesor	86
Figura 74. Icono de Ejecución del la Interfaz Sonic 137	87
Figura 75. Botón para Cambiar a Detección de Fallas	87
Figura 76. Perilla de Ajuste del Rango	87
Figura 77. Perillas de Ajuste de la Compuerta	87
Figura 78. Visualización Compuerta	88
Figura 79. Visualización Compuerta y Falla Detectada	88
Figura 80. Indicador de la Profundidad de la Falla	88
Figura 81. Diagrama esquemático conversor	93

Figura 82. Primera Capa Layout del Conversor	93
Figura 83. Segunda Capa Layout del Conversor	94
Figura 84. Simulación en 3D del conversor	94
Figura 85. Mensaje de Windows de reconocimiento de hardware, Comunicación USB	95
Figura 86. Asistente de instalación	95
Figura 87. Asistente de instalación, Instalar desde una lista o ubicación específica	95
Figura 88. Asistente de instalación, opción de examinar la carpeta o dirección donde se encuentran los drivers	96
Figura 89. Proceso de instalación de drivers	96
Figura 90. Asistente de instalación terminado	96
Figura 91. Mensaje de verificación del nuevo Hardware instalado	97
Figura 92. Asistente de instalación, emulación puerto COM	97
Figura 93. Mensaje de Windows de reconocimiento de hardware, puerto COM virtual creado	97
Figura 94. Opción para identificar el puerto COM creado	97
Figura 95. Asistente de propiedades del sistema	98
Figura 96. Administrador de dispositivos de Windows para identificar nuevos puertos	98
Figura 97. Reconocimiento de puerto	99
Figura 98. Entorno programa <i>Pic C Compiler</i>	100
Figura 99. Esquemático en Eagle del programador	108
Figura 100. Layout del programador	109
Figura 101. Simulación en 3D del programador	109
Figura 102. Reconocimiento del hardware	110
Figura 103. Asistente para hardware	110
Figura 104. Carpeta que contiene los drives	111
Figura 105. Detección de drives e instalación	111
Figura 106. Ventana de seguridad de windows	112
Figura 107. Mensaje de correcta instalación	112
Figura 108. Extracción de componentes del run time	114
Figura 109. Inicio de extracción	114
Figura 110. Instalación	115
Figura 111. Asistente para guardar programa instalado	115
Figura 112. Componentes instalados	116
Figura 113. Fin de la instalación	116
Figura 114. Diagrama esquemático TAD	117
Figura 115. Layout TAD	117
Figura 116. Simulación en 3D de LA TAD	118
Figura 117. Hardware TAD	118
Figura 118. Hardware TAD con su Estuche	118
Figura 119. Hardware TAD Y CABLE	118

LISTA DE ANEXOS

ANEXO A. GUIA DE CALIBRACIÓN SONIC 137	84
ANEXO B. GUIA DETECCIÓN DE FALLAS	87
ANEXO C. CIRCUITO INTEGRADO FT232BM UTILIZADO EN CONVERSOR SERIAL-USB	89
ANEXO D. DIAGRAMA ESQUEMATICO Y PCB CONVERSOR RS232-USB	93
ANEXO E. INTALACION DE LOS DRIVES PARA EL CONVERSOR	95
ANEXO F. COMPILADOR PIC	100
ANEXO G. PROGRAMACIÓN PIC 18F2550	102
ANEXO H. ESQUEMATICO LAYOUT E IMPRESO DEL PROGRAMADOR	108
ANEXO I. CONEXIÓN TARJETA DE ADQUISICIÓN DE DATOS E INSTALACIÓN DEL RUN TIME	110
ANEXO J. ESQUEMÁTICO PCB Y HARDWARE DE LA TARJETA DE ADQUISICIÓN DE DATOS	117

RESUMEN

Título:

REDISEÑO DE LA INTERFAZ GRÁFICA PARA EL EQUIPO DE EVALUACIÓN ULTRASÓNICA SONIC 137.*

Autores:

PEDRO SIZA MORENO, EDWARD ANDRÉS GARCÍA REMOLINA**

Palabras Claves: Labview, VISA, Sonic 137, Calibración, Espesores, RS-232, USB.

CONTENIDO:

La escuela de ingeniería Metalúrgica cuenta con equipos de medición de espesores por ultrasonido, entre ellos el equipo Sonic 137, había sufrido daños en su pantalla y no era posible su visualización de señales, en el año 2002 se desarrolló e implementó una interfaz gráfica que puso en funcionamiento parcial el equipo.

Para un mayor aprovechamiento del equipo se realizó el rediseño de la interfaz, desde Labview se obtuvo una imagen y configuración de los submenús similar al Sonic 137, la actualización de los datos al Sonic con un retardo del orden de los 700ms, permite la calibración completa lo que genera la activación del indicador del espesor, las alarmas que incluye el equipo permiten configurar límites superior e inferior, la activación de la alarma se realiza cuando el espesor se encuentra por fuera de este rango. Es posible realizar pruebas con haz angular, estas pruebas no destructivas se caracterizan por enviar el haz de ultrasonido a un ángulo determinado por el usuario.

Con el fin de utilizar el rediseño de la interfaz con computadores actuales, que en su mayoría solo contienen puertos USB y considerando que la comunicación del Sonic 137 es RS-232, se desarrolló un conversor RS232-USB, proyectar la utilización por más tiempo en los laboratorios de Ingeniería Metalúrgica.

Con los conocimientos previos obtenidos en el desarrollo del conversor y la información sobre estándares de comunicación con Labview, se implementó una tarjeta de adquisición de datos como aporte para los laboratorios de automatización de procesos de la escuela de Ingeniería Eléctrica, Electrónica y Telecomunicaciones.

* Proyecto de grado

** Facultad de ingenierías físico- mecánicas. Escuela de Ingenierías Eléctrica, Electrónica y Telecomunicaciones. Director: MPE. Jaime Barrero Pérez. Codirector: MSC. Luz Amparo Quintero Ortiz

ABSTRACT

Title:

RE-DESIGN OF THE GRAPH INTERFACE FOR THE SONIC 137 ULTRASONIC EVALUATION EQUIPMENT. *

Authors:

PEDRO SIZA MORENO AND EDWARD ANDRÉS GARCÍA REMOLINA **

Keywords: Labview, VISA, Sonic 137, Calibration, Thickness, RS-232, USB.

CONTENT:

The school of Metallurgy Engineering has equipment to measure thickness based on ultrasound. One of them, the Sonic 137 equipment exhibited screen damages and it was not possible to see signals. A graph interface was designed and set up in the year 2002 in order to make this equipment work in a partial manner.

Redesigning of the interface was completed to use the equipment better. Image and configuration of Submenus were obtained from the Labview program and this process was similar to the Sonic 137. This updating of data had a retardation of 700 ms with regard to the Sonic equipment which, in turn, allows the activation of thickness indicator and modify the upper and lower limits. The alarm activation is completed when the thickness is out of this rank. Tests are possible to be conducted with an angular beam; these non – destructive tests are characterized by sending the ultrasonic beam at an angle determined by the user.

An RS232-USB converter was developed in order to use interface redesign with new technology computers since the majority of them utilize USB ports, considering that the communication offered by Sonic 137 is RS-232. Longer use in laboratories of Metallurgy Engineering is attained by this means.

A data – acquisition board was implemented based on the previous knowledge gained during the converter development process and the information on communication standards with Labview. This is a contribution for the Automation process Laboratories of the Electrical, Electronic and Telecommunication Engineering schools.

* Graduation project

** Faculty of physical and Mechanical Engineering. Electrical, Electronic and Telecommunication Engineering schools. Director: MPE. Jaime Barrero Pérez. Director Assistant: MSC. Luz Amparo Quintero Ortiz

INTRODUCCIÓN

El ultrasonido en la industria permite la medición de distancias, caracterizaciones internas del material y en general la realización de ensayos no destructivos. El equipo Sonic 137 es un medidor de espesores que utiliza ondas de ultrasonido, en el primer capítulo se encuentran las características de la ondas y se especifica los rangos de frecuencias de trabajo para la medición de espesores en metales, las aplicaciones en detección de fallas y los diferentes tipos de transductores utilizados para las pruebas no destructivas.

El Sonic 137, presenta un daño en la pantalla de rayos catódicos, impidiendo la aparición de los Ecos de ultrasonido y las compuertas utilizadas en la calibración. La reparación para este daño genera unos costos elevados, orientando a que la mejor solución sea realizar una interfaz que además de las características de la primera versión, genere una mejor proyección del Sonic para la ejecución de una completa calibración, posterior medición de espesor y utilización de alarmas propias del equipo.

La descripción de los componentes y las funciones del Sonic, muestra características del equipo que son tomadas como información necesaria para el diseño, identificando las funciones que serán implementadas en la segunda versión de la interfaz.

Junto con la pantalla de visualización de la gráfica, menú e indicador de espesor, las funciones que se implementaron en la segunda versión, contribuyen a la calibración completa del Sonic, mediante la aparición de una compuerta y la activación del indicador de espesor, procedimiento de calibración presentado en el manual de usuario.

En el cuarto capítulo se presenta el diseño de la Segunda versión de la interfaz, partiendo de los requisitos del sistema, donde se tratan las necesidades de los estudiantes de Ingeniería metalúrgica y las recomendaciones por parte de la

docente Msc. Luz Amparo Quintero, así como el desarrollo del diseño se muestra la programación en LABVIEW y finalmente el resultado de la segunda versión.

El Sonic 137 presenta una comunicación con la norma RS232 para interactuar con el PC. A partir de esto se vio la necesidad de desarrollar un conversor RS232-USB que permitiera el acople directo en computadores de última tecnología y equipos portátiles que en su mayoría solo contienen puertos USB. Dentro del quinto capítulo se presenta el diagrama esquemático del conversor, los componentes utilizados, la descripción de los mismos y la instalación de los drives para su adecuado funcionamiento.

En el capítulo seis se presenta el resultado final del rediseño, donde se desarrollan una serie de pruebas no destructivas con diferentes patrones de medida para comprobar el correcto funcionamiento de la segunda versión.

Además de proyectar la utilización del Sonic por más tiempo gracias al rediseño de la interfaz, se creó una tarjeta de adquisición de datos que funciona desde Labview, para ser utilizada en los laboratorios de automatización de procesos en conjunto con los estudiantes de Ingeniería Electrónica, donde pueden recibir tanto señales analógicas como digitales y simular salidas digitales para algún tipo de control. En el capítulo final se presenta el circuito de la tarjeta, la programación en Labview, la programación del microcontrolador utilizado y la descripción detallada de su utilización además de su previa instalación.

1. ONDAS DE ULTRASONIDO

1.1 Definición

Los sonidos que alcanzan a ser captados por el oído humano se encuentra en el rango de frecuencias entre los 15 a los 20KHz, por debajo de este rango se encuentran los infrasonidos, algunos animales para comunicarse a grandes distancias envían estas ondas, como por ejemplo el elefante. Los desastres naturales como las erupciones volcánicas, terremotos y tornados, producen sonidos que se encuentran por debajo de los 20 Hz. [3]

Por encima del rango de frecuencia audible por el ser humano se encuentra el ultrasonido, con aplicaciones industriales en las cuales encontramos la medición de distancias, ensayos no destructivos y en la medicina en ecografías y ultrasonoterapia.

Una onda de ultrasonido es un conjunto de vibraciones mecánicas que viajan a través de un medio, puede ser solido, líquido o un gas, a una velocidad determinada que al encontrar un material diferente, la onda es reflejada o transmitida según las reglas básicas de física. En ensayos no destructivos, para la medición de espesor en metales, el rango de frecuencias trabajado es de 250KHz a 25MHz, con una longitud de onda mucho menor a las ondas de frecuencia audible. [9]

Las vibraciones mecánicas tienen forma de onda similar a las ondas luminosas pero requieren de un medio elástico para la propagación, como líquidos o sólidos. Si la temperatura y la tensión en el material permanecen constantes la relación de longitud de onda se define como;

$$\lambda = \frac{c}{f}$$

λ = Longitud de onda

c = Velocidad del Sonido del material

f = Frecuencia

Los modos de propagación más utilizados en la detección de fallas por ultrasonido son de ondas longitudinales y de ondas transversales. Las longitudinales se caracterizan por el movimiento de las partículas a la misma dirección de la propagación de la onda y las ondas transversales el movimiento de las partículas es perpendicular a la dirección de propagación de la onda. [10]

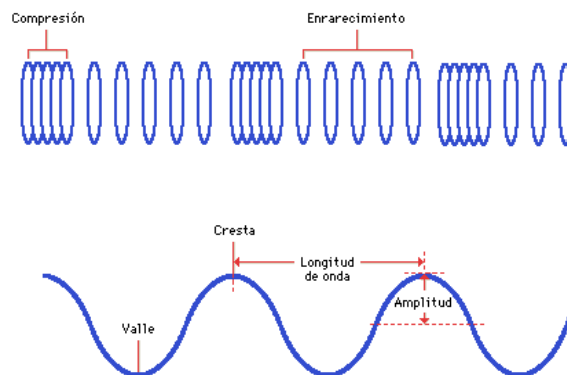


Figura 1. Ondas longitudinales y Ondas Transversales [6]

La velocidad de propagación depende del material que se tenga para las pruebas no destructivas.

Debe tenerse en cuenta dos factores básicos para los ensayos no destructivos, el tiempo que tarda en viajar la onda en el material y la amplitud de la señal recibida, mediante la relación de velocidad y distancia es posible determinar el espesor del material;

$$T = \frac{ct_s}{2}$$

T = Espesor del material

c = Velocidad del sonido en el material

t_s = Tiempo de viaje de la señal (ida y vuelta)

Todas las emisiones de ultrasonido divergen, es decir los transductores dispersan las emisiones, el transductor plano mostrado en la **Figura 2**, es un ejemplo que muestra el ángulo formado, este puede reducirse seleccionando un transductor de mayor frecuencia o de mayor diámetro. [3]

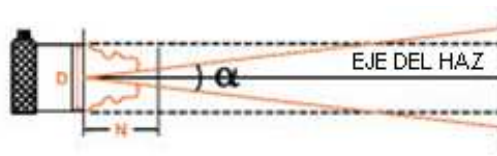


Figura 2. Dispersión de las Emisiones del transductor [3]

1.2 Principio de Medición por Ondas de Ultrasonido

Los ensayos no destructivos por ultrasonidos, son utilizados para la caracterización del espesor de un material por medio de ondas sonoras a altas frecuencias, que utiliza un transductor que se compone de un elemento activo que convierte energía eléctrica en un pulso de excitación en energía ultrasónica y un atenuador que absorbe la energía radiada para controlar las vibraciones del transductor. [3]

Las mediciones por ultrasonido pueden ser utilizadas en la mayoría de materiales industriales, incluyendo gran cantidad de metales. El Sonic cuenta con una alta precisión, utilizado a nivel industrial para el control de calidad. [4]

1.3 Angulo de Reflexión y Refracción

En los casos donde la onda de ultrasonido encuentra un límite, el ángulo de reflexión es igual al ángulo de incidencia, si un haz incide perpendicularmente a una superficie se refleja directamente de vuelta, si se trasmite de un material a otro se basa en la ley de Snell de refracción (Ver **Figura 3**), que consiste en la relación del seno de los ángulos de incidencia, refracción y las velocidades de cada material [10],

$$\frac{\sin \phi_1}{\sin \phi_2} = \frac{V_1}{V_2}$$

ϕ_1 = ángulo de incidencia en el primer material

ϕ_2 = ángulo refractado en el segundo material

V_1 = la velocidad de sonido en el primer material

V_2 = velocidad del sonido en el segundo material

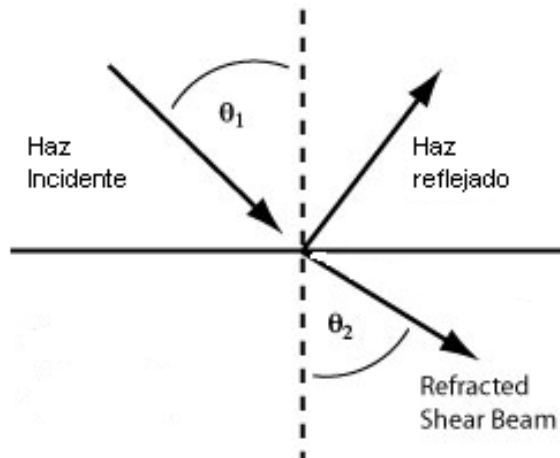


Figura 3. Haz de incidencia y refracción [10]

1.4 Método de Detección de Fallas por Ultrasonido

Es una de las aplicaciones de ultrasonido más antiguas y utilizadas, desde la década de los 40 según las leyes físicas orientadas a la propagación de ondas de sonido a través de materiales sólidos utilizados para detectar grietas, huecos, porosidad y otras discontinuidades. [10]

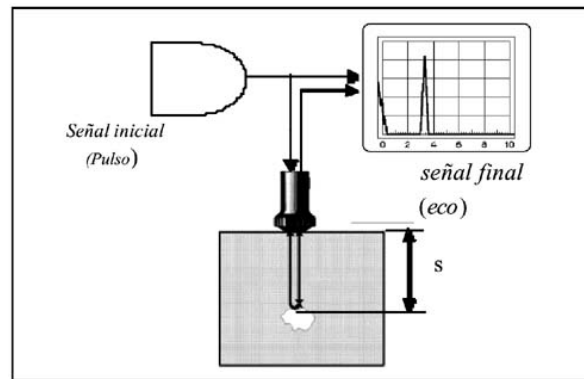


Figura 4. Ejemplo de Detección de Fallas [3]

Son consideradas pruebas no destructivas porque se basan en la aplicación de ondas de alta frecuencia sobre un objeto sin alternarlo ni dañarlo. Estas pruebas son importantes para el continuo desarrollo industrial y pueden ser utilizadas como prevención en casos donde una falla en metales pueda generar catástrofes o que alteren la integridad de sus empleados. [3]

1.5 Transductores Ultrasónicos

Un transductor es un dispositivo que convierte una energía de una forma a otra por medio del efecto del piezoeléctrico, los transductores utilizados para las pruebas de medición y detección de fallas parte de una excitación con un alto voltaje de impulso eléctrico, que vibra a través de un espectro de frecuencias y genera una ráfaga de ondas de ultrasonido, es necesario una capa delgada de acoplamiento de liquido o de grasa entre el transductor y la pieza de ensayo. [3]

1.5.1 Materiales Piezoeléctricos

1.5.1.1 Cuarzo

Es obtenido en cristales naturales, con buenas características de estabilidad eléctrica, química y resistente al desgaste, requiere un alto voltaje para el manejo a bajas frecuencias y se debe utilizar a temperaturas menores a 500°C. [5]

1.5.1.2 Sulfato de Litio

Considerado uno de los receptores más eficientes, no envejece y es poco afectado por interferencias en conversión, debe emplearse a temperaturas menores a los 75°C. [5]

1.5.1.3 Cerámicos Polarizados

Es obtenido por sintetización y se polarizan durante el proceso de fabricación, opera a bajos voltajes de polarización, no son afectados por la humedad y soportan temperatura de hasta 300°C, en algunos casos se presentan interferencia en conversión. [5]

1.5.2 Transductores Utilizados en la Detección de Fallas

1.5.2.1 Transductores de contacto



Figura 5. Transductor de contacto [10]

Se utilizan en contacto directo con el material a evaluar con un material de acoplamiento, incide el haz perpendicular a la superficie, se utiliza para la localización de huecos, porosidad y grietas, ideal para imperfecciones paralelas a la superficie.

1.5.2.2 Transductores de Haz Angular

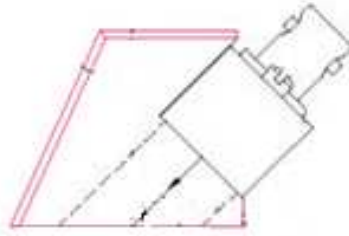


Figura 6. Transductor de Haz Angular [3]

Se construye a partir del acople de un transductor de contacto (Haz recto) a una de las caras de una zapata de plástico, el cual presenta un determinado ángulo, es utilizado para introducir ondas longitudinales, normalmente se utiliza para la inspección de soldadura o imperfecciones perpendiculares a la superficie.

1.5.2.3 Transductores de Línea de Contacto

Se utiliza para pruebas en altas temperaturas, la línea de retardo protege el elemento de daños térmicos.

1.5.2.4 Transductor de Inmersión

Se utiliza para ensayos en agua.

1.5.2.5 Transductor de Doble Elemento

Es utilizado por separado el transmisor y receptor en un solo montaje, generalmente utilizado en materiales ásperos, para la detección de picaduras o porosidad.

1.5.3 Material Acoplante

Es un medio viscoso que permite el paso de las ondas del transductor a la pieza examinada, el objetivo es evitar la presencia del aire, para obtener buena propagación de la onda.

Se debe considerar que tal acoplante debe tener humectabilidad, viscosidad adecuada, baja atenuación, no ser corrosivo y fácilmente removible, generalmente los materiales utilizados son el aceite, glicerina, grasa y vaselina. [3]

2. SONIC 137 Y PRIMERA VERSION DE LA INTERFAZ



Figura 7. Fotografía del Sonic 137

2.1 Descripción del Sonic 137

El Sonic 137, es un equipo que permite la medición del espesor de materiales metálicos y ofrece la aplicación de detección de fallas, la visualización de los ecos se realiza mediante la emisión de rayos catódicos a la pantalla de cristal líquido (CRT), los ecos representan la onda de ultrasonido que viaja a través del material, esta pantalla sufrió un daño que impide que se presenten los Ecos.

La pieza averiada que impide la visualización resulta algo costosa para la escuela de ingeniería metalúrgica, por lo cual se decidió implementar una interfaz realizada en Labview que permite la visualización de la gráfica y la configuración del Sonic.

Este daño en la pantalla CRT limita la utilización del Sonic 137, en cuanto a la calibración y la posterior medición de espesores.

El equipo en su parte posterior cuenta con un puerto serial y permite la comunicación al PC, mediante comandos especificados en el manual de usuario, es posible realizar la escritura y lectura de los valores contenidos en los submenús, la obtención la gráfica y el envío del PC al Sonic 137 de comandos adicionales.

Parámetros RS-232	Configuración del Sonic
Rata de baudios	9600 bps
Tamaño (Bits)	8
Paridad	Ninguno
Bits de Parada	1

Tabla 1. Datos Técnicos para Configurar el Puerto Serial [2]

El módulo adicional de corrección de distancia de amplitud (DAC) ofrecido por los fabricantes, no fue adquirido, por esta razón no es posible hacer uso de esta función.

La pantalla de visualización del menú, el indicador de espesor y demás botones presentan un buen funcionamiento.

2.2 Primera Versión de la Interfaz

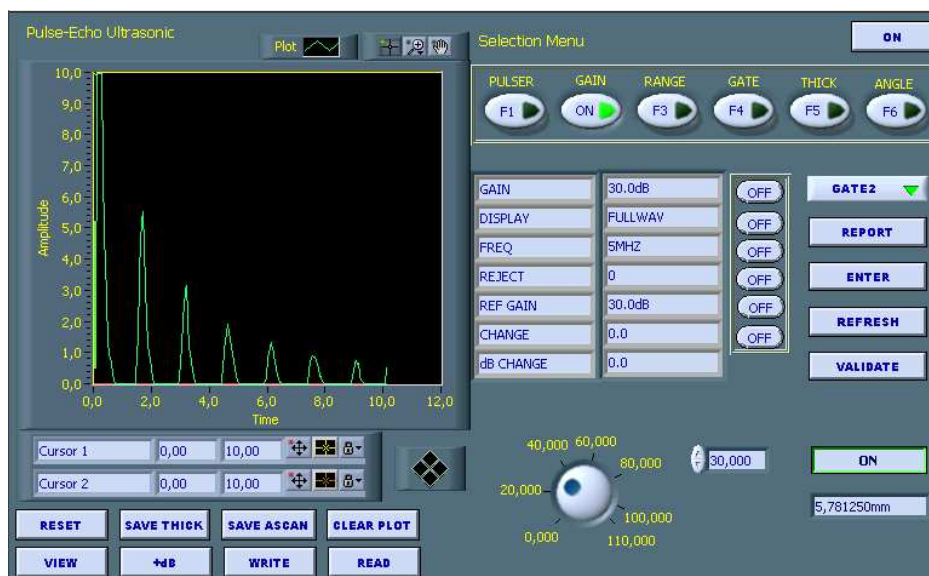


Figura 8. Primera Versión de Interfaz para el Sonic 137 [2]

La interfaz permite el despliegue rápido de los menús y la modificación de los submenús se realiza mediante los botones “OFF” y después de modificado en necesario oprimir “VALIDATE”, lo que genera solo una escritura al Sonic, la lectura es realizada mediante “REFRESH” y permite la actualización de los valores cambiados directamente del Sonic, la gráfica es tomada por cortos intervalos de tiempo lo que facilita observar los cambios de espesores.

La separación de los ecos presentados en la **Figura 8** representa la distancia que recorre la onda de ultrasonido, después de ubicar en la posición correcta los ecos del espesor mayor y menor a medir, es posible obtener para otros valores de espesor muy aproximados a los reales.

Hasta este punto no termina la calibración del equipo, pero es posible obtener el valor de espesor mediante cálculos matemáticos, la primera versión de la interfaz lo realiza a partir de la grafica.

La calibración continúa con la activación de la compuerta *GATE2*, una barra de referencia del Sonic para detectar el pulso mayor e internamente realizar el cálculo necesario para obtener el valor de espesor, que posteriormente será mostrado en el indicador del equipo, a partir de este momento se genera una saturación del Sonic, pues el equipo internamente realiza varias ejecuciones que superan el límite de trabajo optimo, esta saturación es reflejada en la lectura de valores erróneos de la gráfica y parpadeos continuos en el indicador de espesor. Esta activación no es posible realizarla con la primera versión.

La función de Haz angular utiliza palpadores (transductores) diseñados para transmitir la onda de ultrasonido a un ángulo determinado, esto permite detectar fallas que resultan difíciles para observar con el haz de ángulo recto, la interfaz no cuenta con detección de fallas mediante la identificación de ecos en diferente posición.

El diseño de la interfaz se realizó en Labview 6.1, este software ofrece SubVÍ's para la comunicación serial, herramienta que ejecuta gran cantidad de funciones con pequeños retardos.

3. DESCRIPCIÓN DE COMPONENTES Y FUNCIONES DEL SONIC

3.1 Componentes para la Visualización

Pantalla CRT: Permite visualizar la señal de ultrasonido que proviene del transductor.[1]



Figura 9. Pantalla CRT [Autores]

Menú “*menu display*”: Muestra los parámetros de operación del instrumento, indican la selección y permite el ajuste. También muestra anuncios o mensajes de error.



Figura 10. Menú Display [Autores]

Indicador de Espesor “*thickness display*”: Muestra la lectura del espesor, para la inspección por haz angular puede ser programado para mostrar la distancia o la profundidad de la lectura.



Figura 11. Indicador de Espesor “*Thickness Display*” [Autores]

Indicador de estado “*status indicators*”: Muestra el estado de operación del instrumento y permite identificar cual característica del instrumento esta seleccionada, se compone de: [1]



LOCK: Los parámetros del instrumento están bloqueados y no es posible cambiarlos hasta que se desbloquee.

VIEW: La pantalla está en modo de vista ampliada.

INTF: El instrumento está en la interfaz de sincronización.

+dB: El instrumento tiene 6, 12, 14 o 20 dB de ganancia añadida a la muestra.

DAC: La función opcional de corrección de distancia de amplitud esta activa.

150V: La parte inferior de impulsos de tensión se ha seleccionado.

DUAL: El pulso/receptor esta funcionando en modo doble elemento.

REJ: El rechazo lineal se aplica para la señal recibida.

GATE: Parpadea alternadamente con el indicador de alarma (ALARM) cuando se presenta una falla.

ALARM: Parpadea alternadamente con el GATE cuando hay una falla y con THICK cuando hay un espesor fuera de la tolerancia de alarma.

THICK: Parpadea alternadamente con ALARM indicando cuando el espesor esta fuera de la tolerancia de alarma.

BAT: La batería se encuentra casi descargada.

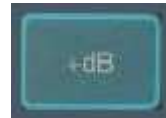
3.2 Comandos de Ejecución Inmediata

3.2.1 Vista Ampliada “*View*”



Esta función es utilizada para expandir la señal en el CRT, sirve para elegir el eco deseado.

3.2.2 Aumento de Ganancia “+dB”



Permite incrementar la ganancia del instrumento a 6, 12, 14 o 20 dB.

3.2.3 Guardar Valor de Espesor “*Save Thick*”



Guarda la lectura de espesor, dispone una ubicación de almacenamiento.

3.2.4 Guardar Gráfica “*Save Ascan*”



Activa el digitalizador para convertir la muestra de forma de onda de ultrasonido a su equivalente digital y la guarda en una ubicación de almacenamiento.

3.2.5 Opción del DAC “*Profile*”



Si la opción DAC está instalada, este activa el digitalizador para convertir la forma de onda de ultrasonido a su equivalente digital. [1]

3.3 Comandos del Menú General

3.3.1 Configuración del Pulso Eco “*Pulser*”



Configura el pulso enviado al transductor para la medición y análisis, como el ajuste del espesor de señal cuadrada, control de la carga eléctrica aplicada al transductor, modo de operación como opciones pulso eco, dual o transmisión directa, ajuste de la frecuencia de repetición del pulso ultrasónico, voltaje de la onda cuadrada y la función de limitar la frecuencia de repetición del pulso.

3.3.2 Ganancia “*Gain*”



Permite la configuración de especificaciones relacionadas con las ganancias y datos del sensor, frecuencia de operación de la circuitería interna, ajuste de nivel de rechazo lineal, así como el cambio en el tipo de detección aplicada a la visualización, media onda, onda completa y sin detección.

3.3.3 Compuertas para Detección de Fallas “Gate”



Se establecen los parámetros que permiten la determinación de fallas presentadas en el material a evaluar cuando la amplitud supera o se baja del nivel de umbral establecido.

El ajuste de la compuerta GATE 2 permite la calibración del equipo, representa una barra definida por el nivel (LEVEL), posición (POSN) y ancho (WIDTH), que permite internamente determinar la posición y el equivalente espesor

3.3.4 Configura Rango, Retardos y Velocidad del Material “Range”



Se configuran los parámetros de trabajo del Sonic, el rango (RANGE) determina el valor de espesor mayor a medir, retardos utilizados en la calibración así como la velocidad que depende del material utilizado para las pruebas, permite cambiar las unidades de trabajo.

3.3.5 Configuración de Alarmas y Medición de Espesor “Thick”



Activación del indicador del espesor del Sonic (T-GAUGE=ON) después de activada la compuerta GATE 2, permite la calibración, especificando el espesor mayor y menor que será medido, se configuran los límites de tolerancia de

espesor y la activación de alarmas, con el fin de determinar cuando el material presenta fallas.

3.3.6 Haz Angular “Angle Beam”



Da la posibilidad al Sonic de detectar fallas con un haz incidente con ángulo determinado, permite detectar fallas con diferentes orientaciones a las que son posibles captar con palpadores de haz recto.

Es posible modificar el valor mostrado en el indicador de espesor mediante, para determinar la distancia recorrida o la profundidad de la falla mediante “SHOW”.

3.3.7 Corrección de Distancia de Amplitud “DAC”



Función destinada para la corrección de distancia de amplitud, es un modulo adicional, el Sonic 137 de la escuela de Ingeniería metalúrgica no cuenta con esta función.

3.3.8 Almacenamiento e Informes “Setup”



Se utiliza para fijar los parámetros de almacenamiento, editar un informe programado, y utilizar los códigos de acceso.

3.3.9 Recuperar Datos Almacenados “Recall”



Es utilizado para recuperar los programas almacenados, A-scan y lecturas de espesor. Permite la selección del tipo de información a recordar, y recuerda el tipo de datos seleccionado.

3.3.10 Configuración de Hora y Fecha “Spcl”



Incluyen menús de panel frontal, como control de fondo, la hora, fecha, forma de onda de procesamiento y comunicaciones.

4. DISEÑO INTERFAZ EN LABVIEW

4.1 Identificación de Requisitos

Obtener una mejoría en la utilización del Sonic desde el PC y dar la posibilidad de realizar pruebas que ofrece el equipo.

Facilitar el manejo y asimilar un poco más la configuración del Sonic, agregar un navegador de menú, para dejar atrás los botones de seleccionar comando (“OFF” y “ON”) y “VALIDAR”, utilizados en la primera versión de la interfaz para permitir la escritura al Sonic.

Después de visualizada la gráfica, determinar la ubicación de los pulsos, procedimiento propio de la calibración donde se ubicará el eco en la posición correcta según el espesor medido, de la misma forma como era realizado en los laboratorios por los estudiantes de ingeniería metalúrgica, donde determinaban de manera manual y aproximada la distancia entre el eco inicial y el siguiente eco, obteniendo a partir del recorrido de la onda de ultrasonido equivalente al espesor del material.

Generar una visualización constante de la ubicación del eco, esto permite facilitar el ajuste del extremo inferior y superior de la medida en el metal utilizado, conociendo su espesor y con la modificación de algunos comandos del Sonic permite ubicar el valor más acertado.

La aparición de la compuerta en la pantalla permite configurarla para que la barra corte el eco mayor (diferente al eco inicial) y de esta manera internamente el Sonic genera el valor que será mostrado en el indicador del espesor y en la interfaz.

La activación del indicador de espesor del Sonic se realiza mediante el comando “T-GAUGE”, permitir activar el submenú de CAL-MODE mediante la interfaz con el fin de completar la calibración del equipo.

Permitir la visualización de alarmas en presencia de una falla, para las pruebas realizadas con haz recto y haz angular.

4.2 Interesados

Estudiantes Ingeniería Metalúrgica, son los usuarios directos del sistema, el cumplimiento de los objetivos del sistema, genera que cuenten con una buena formación en la utilización de medidores de espesor.

Docente Luz Amparo Quintero, cuenta con excelente habilidad para el manejo del equipo y establece las necesidades que son tomadas como los objetivos del sistema.

4.3 Descripción de Requisitos

4.3.1 Comunicación en Labview

4.3.1.1 Visa

Este módulo facilita la comunicación mediante el puerto serial, en capacidad de enviar una cadena de caracteres ASCII para obtener las operaciones que realiza el instrumento, esto conlleva a la ventaja de generar una interfaz independiente y mediante su configuración poder realizar modificaciones que generan la programación de cualquier instrumento.

VISA es un lenguaje orientado a objetos y facilita la adaptación de nuevas interfaces que se desarrollaran en un futuro. Si una sesión no se cierra cuando se ejecuta en un periodo, esta permanecerá abierta y es posible que se presenten

problemas con los recursos del sistema, hay casos donde dejando abierto el puerto puede ser útil.

Tomamos la configuración que presenta un ejemplo básico de comunicación serial en Labview de lectura y escritura, utilizando los siguientes elementos

4.3.1.1.1 Visa Serial



Permite la configuración del puerto donde podemos especificar la tasa de baudios, puerto para utilizar la comunicación, numero de bits de datos y bit de paridad.

La tasa de baudios especifica que tan rápido se mueven los datos en el instrumento que se usan mediante la comunicación serial. La señal de salida por lo general oscila entre 12V y -12V. El orden de transmisión es lógica invertida y del bit menos significativo (LSB) al más significativo (MSB).

El bit de paridad es opcional, si está presente también maneja la lógica invertida, (1 para voltaje negativo y 0 para voltaje positivo), este bit se incluye como medio de gestión de errores. [19]

La velocidad de transmisión determina el número de caracteres por segundo, por ejemplo para una tasa de transmisión de 9600 baudios, se obtiene 827 caracteres por segundo, lo que resulta algo positivo para los cambios que se realizaron en la programación de la interfaz para el Sonic.

4.3.1.1.2 Visa Write



Permite la escritura de una cadena de caracteres en el instrumento.

4.3.1.1.3 Visa Read



Permite la lectura de datos, según el numero de bytes especificado ofrece los datos en un buffer de salida, es posible seleccionar la lectura de datos síncrona y asíncrona independiente de la plataforma. [19]

4.3.1.1.4 Visa Close



Esta función cierra la sesión con el instrumento y las emisiones de cualquier sistema de recursos que se utilizaron.

A partir de estos componentes se realizaron Sub VÍ's, para la escritura y toma de datos del Sonic, y se busco la manera de poder configurar otros puertos, aparte del COM1, puertos que permite la conexión por medio del Bus Universal en Serie (USB).

4.3.2 Lectura de Datos del Sonic

Considerando que la ejecución de los programas en Labview, realiza inicialmente las estructuras "sequence", fue utilizada para la lectura de los valores iniciales del Sonic, realizando la actualización de los valores contenidos en los "Know" o las llamadas "perillas" utilizadas en la interfaz, para la lectura de los valores del Sonic se realizo un subVI, que ofrece la posibilidad de cambiar el puerto, desde el COM1 al COM9, para tomar los llamados "puertos virtuales" que configura Windows para los USB.

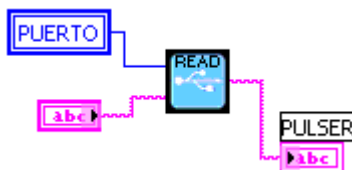


Figura 12. Esquema de Programación para la Lectura de Datos [Autores]

Con base en el manual del equipo son tomadas las instrucciones necesarias para la lectura, este SubVI entrega un vector de tipo string, incluyendo los valores de los submenús de tres menús del Sonic, en la **figura 13** muestra los valores de la función GAIN.

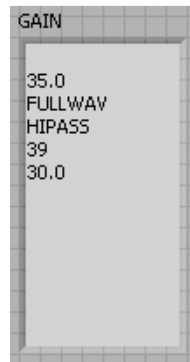


Figura 13. Vector Tipo String de Datos Recibidos [Autores]

Estos valores son obtenidos por separado por medio de un “For Loop” que crea un vector y son tomados para actualizar las perillas. [2]

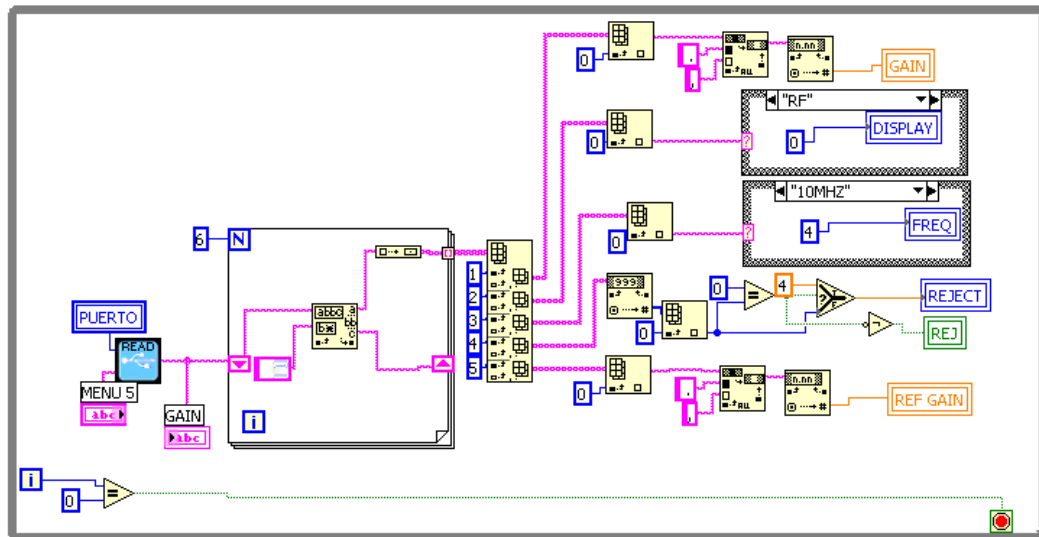


Figura 14. Esquema de Programación para Lectura y Actualización de Perillas [Autores]

4.3.3 Escritura de Datos

La función y el comando inicial corresponde a PULSER y su primer comando que relaciona el ancho del pulso (PULSE) de esta manera después de leídos los datos iniciales, se habilita la escritura constante de este menú, para ello realizamos un SubVI de escritura;

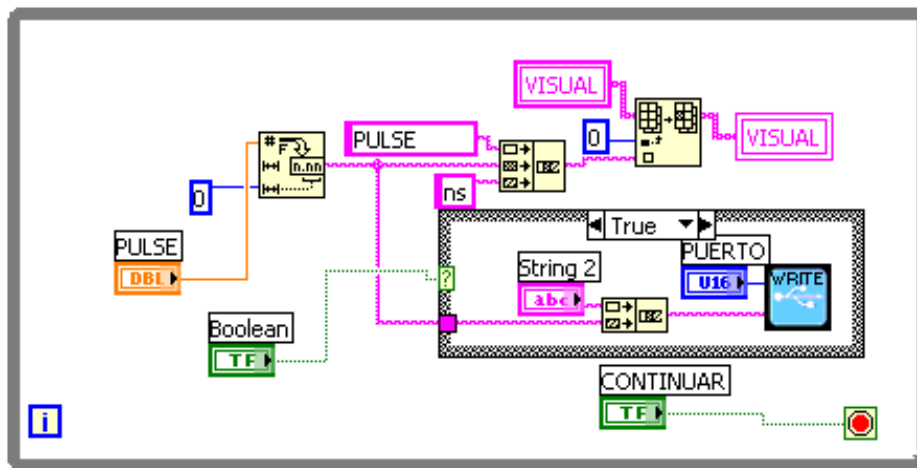


Figura 15. Esquema de Programación para la Escritura de valores al Sonic
[Autores]

El “Boolean”, impide la escritura cuando se obtienen los valores de la gráfica en una duración aproximada de 800ms, esto con el fin de evitar interferencia tanto en la toma como en la escritura de datos, “CONTINUAR”, es activado cuando se realizan cambios en selección de menú, en función o en comando.

4.3.4 Revisión de Cambios en el menú

Al detectar una pulsación en una función diferente, se visualiza de todo el menú y se envía al Sonic la instrucción que permiten visualizar en el “Menú Display” del Sonic, mediante la variable “FUNCIÓN” se almacena el valor y se realiza la escritura según la ubicación especificada.

Esta parte se realiza en un estado de la estructura “Sequence”, que constantemente esta en ejecución, compone de un “case structure” como se ilustra en la **Figura 16**, para el ejemplo se seleccionó la función “PULSER”.

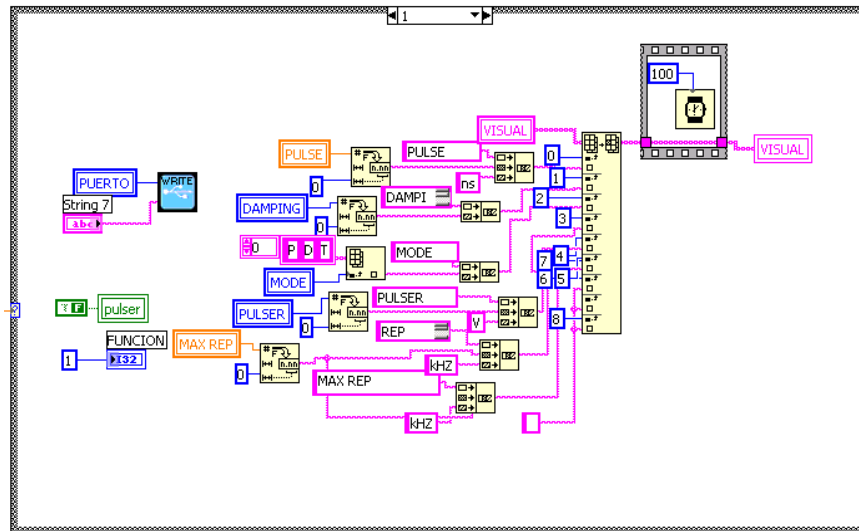


Figura 16. Esquema de Programación para la Visualización del Menú Pulser [autores]

4.3.5 Navegación en el menú del Sonic

El navegador de menú utiliza un “Shift Register”, que a medida que se presiona hacia arriba genera una disminución en el número asignado para la subfunción (Comando) y presionando hacia abajo produce un aumento, de esta manera se permite fácilmente hacer un recorrido por los comandos del menú del Sonic y de manera sencilla cambiar los valores y con muy pequeños retardos (orden de los 700ms) ver estos cambios en el Equipo de medición.

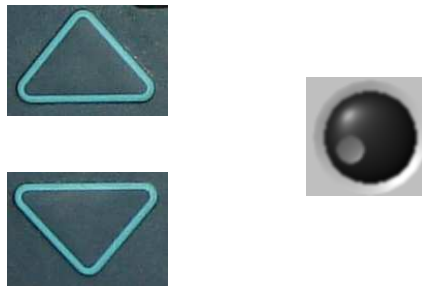


Figura 17. Botones de Navegación y Perilla de Variación [autores]

Adicionamos un indicador del comando seleccionado, el cual se compone de un conjunto de led's organizados, ubicado junto al menú y permite identificar la subfunción que será modificada por el usuario (ver **Figura 18**)

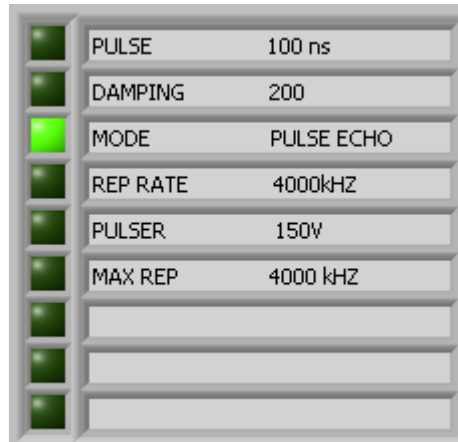


Figura 18. Menú e indicador de comando seleccionado [Autores]

Por cortos intervalos de tiempo se realiza la visualización de la grafica junto al cambio constante de los valores de comandos en el Sonic y la revisión de los cambios de estado de los comandos de ejecución inmediata.

4.3.6 Visualización de la Gráfica

Para evitar el tráfico de datos en el puerto es necesario deshabilitar la escritura de datos al Sonic, de esta manera la gráfica se obtiene constante por la ejecución de la estructura "Stacked Sequence", en otro estado de esta estructura se realizan los cálculos que determinan la distancia de los ecos, que serán la especificación del espesor del material medido.

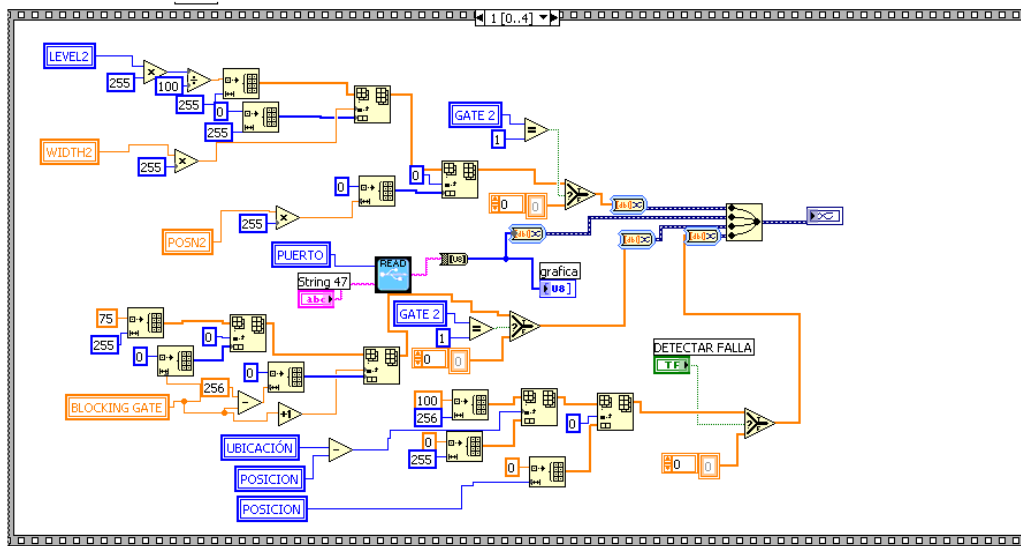


Figura 19. Esquema de Programación Visualización de la Gráfica. [Autores]

4.3.7 Determinación de la Ubicación de los Ecos

A partir de la gráfica se determinan la ubicación de los ecos y se determina el valor de espesor equivalente y es mostrado constantemente en el indicador ubicado en la parte superior de la gráfica (ver **Figura 20**).

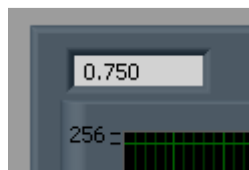


Figura 20. Indicador del Espesor Determinado por la ubicación de los ecos
[Autores]

La programación se realizó con la determinación del valor mayor mediante comparaciones de los 256 valores que entrega el Sonic correspondiente a la gráfica, el valor mayor es tomado sin incluir el eco inicial. Mediante cálculos matemáticos se determino el espesor equivalente. (Ver **Figura 21**)

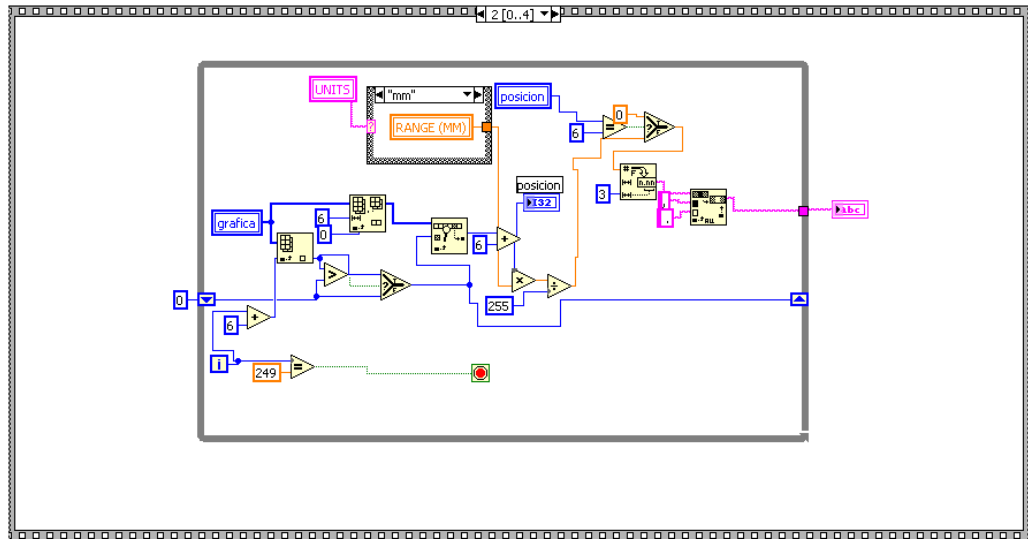


Figura 21. Esquema de Programación Para Determinar la Ubicación de los Ecos.

[Autores]

4.3.8 Revisión de Comandos de Ejecución Inmediata

Para los comandos de ejecución inmediata fue necesaria una estructura "Case Structure" que al ser pulsado realiza la escritura del comando respectivo;

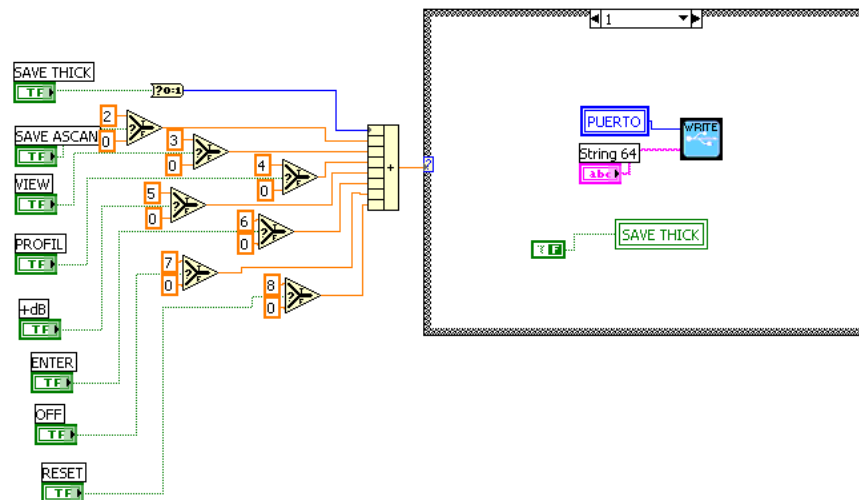


Figura 22. Esquema de Programación para Detectar la Pulsación de los Comandos de Ejecución Inmediata [Autores]

4.3.9 Activación de la compuerta “GATE 2”

La activación de la compuerta genera una gráfica adicional, que corresponde a una línea, con características de nivel, posición y ancho, esta compuerta debe cortar el eco mayor (diferente al eco inicial), ver **figura 23**

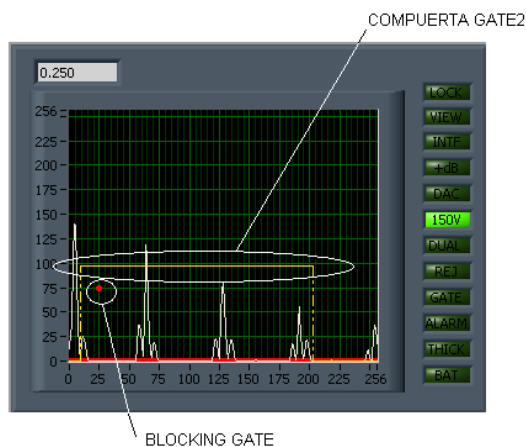


Figura 23. Visualización de la Compuerta GATE 2 y BLOCKING GATE [Autores]

La idea en la programación se basó en la generación de la gráfica mediante los valores modificados como “LEVEL 2”, “WIDTH 2” y “POSN 2”, de la misma manera se muestra constantemente, los cambios en la compuerta se ven en cortos intervalos de tiempo (700ms).

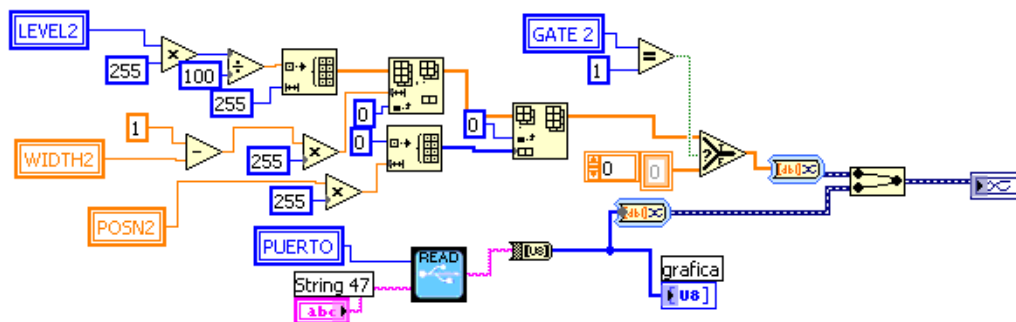


Figura 24. Esquema de Programación Activación de la Compuerta “GATE 2”
[Autores]

4.3.10 Activación del Display de Espesor del Sonic (“T-GAUGE”)

La activación del “T-GAUGE” genera saturación del Sonic, para ello, aumentamos el tiempo en visualizar la gráfica, a partir de este momento los cambios de la gráfica presentan un retardo significativo, orden de 1.5s, se calibra el valor mayor y el menor de manera que sea posible visualizar el valor en el Sonic, de igual manera aparecerá este valor en la interfaz.

Para que la interfaz ejecutara normalmente a excepción de la gráfica fue necesario un “Shift Register” para que cumplido dos ejecuciones, permita la visualización en Labview de la gráfica y del valor de espesor presentado en el Sonic.

4.4 Organización del sistema

Ver diagrama de flujo **Figura 25**.

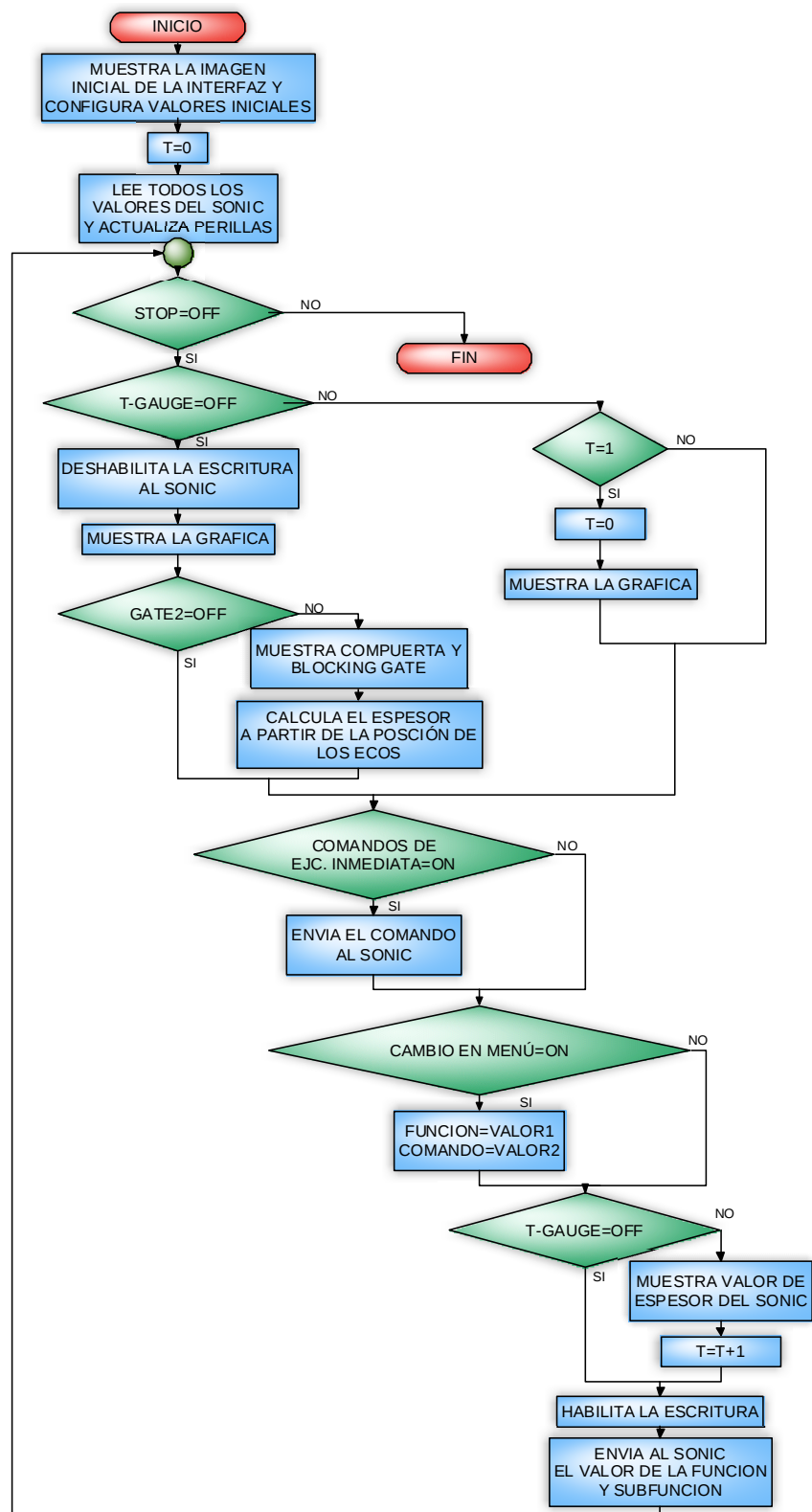


Figura 25. Diagrama de Flujo del Diseño de la Interfaz [Autores]

4.5 Resultado del Diseño



Figura 26. Imagen Final del Diseño de la Segunda Versión de la Interfaz
[Autores]

La segunda versión de la interfaz para el Sonic, cumple con los objetivos planteados, orientados a la completa utilización del equipo, cuenta con la posibilidad del despliegue rápido en el menú del Sonic y genera que el usuario piense que se encuentra manejando el equipo desde LABVIEW, se tomaron imágenes propias del Sonic para que su apariencia fuera similar, su programación bien estructurada y con verificación de posibles fallas en el futuro, hacen del diseño un buen aporte para la escuela de Ingeniería Metalúrgica y aumenta la importancia del Sonic 137 en los próximos laboratorios de pruebas no destructivas.

5. CONVERTOR RS232-USB

Los ordenadores modernos, no poseen puerto serial, lo cual es un problema a la hora de poner en práctica muchos de los temas tratados en Electrónica e incorporar diversos dispositivos que aun funcionan con la norma RS232.

Se construyó un dispositivo que convierta las señales USB a la norma RS232 (Ver **Figura 27**). Esto se logra con un dispositivo que emplea el circuito integrado FT232BM que genera un puerto serial virtual a una computadora, es instalado en la mayoría de los sistemas operativos como Windows y Linux.

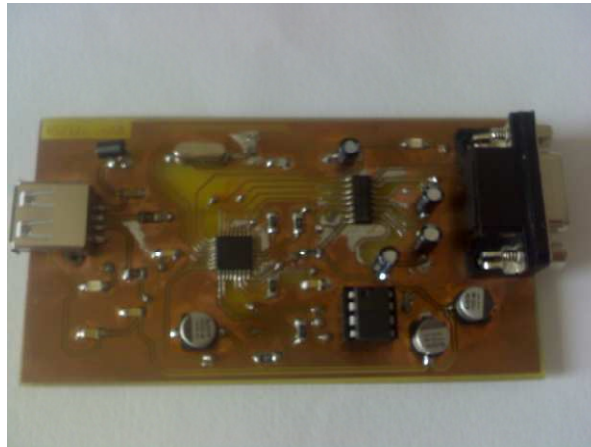


Figura 27. Fotografía del Conversor Serial- USB

La principal característica radica en que es posible seleccionar el puerto COM asociado al puerto USB donde se ha colocado este adaptador, permitiendo así, la conexión del Sonic y demás aplicaciones que necesitan este tipo de conversión.

5.1 Teoría de Comunicación USB

El desarrollo tecnológico de los ordenadores ha permitido que la mayoría de los periféricos necesarios para su adecuada utilización dispongan de comunicación USB, debido a su facilidad de adaptación, configuración e instalación.

5.1.1 Bus Serial Universal (USB)

El Bus serial universal se introdujo en la industria informática para mejorar el rendimiento de la interfaz serial y paralelo, tanto en velocidad de transferencia de datos como en adaptabilidad.

La interfaz USB está compuesta por 4 hilos: Vcc, D-, D+ y GND además de distribuir 5v para la alimentación.

Pin	Señal	Descripción
1	Vcc	+5v
2	D-	Datos -
3	D+	Datos +
4	GND	Tierra

Tabla 2. Descripción de pines USB [Autores]

Actualmente existen diversos componentes para el PC que funcionan con interfaz USB (teclado, mouse, monitor, modem, webcam, etc.) debido a su facilidad de instalación, disponibilidad de puertos y detección automática. Con USB es posible la conexión de 127 periféricos a un solo puerto de un PC.

Todos los dispositivos USB deben tener el mismo tipo de cable y el mismo tipo de conector, sin depender de la función que cumplan. [7]

5.1.2 Origen

Para el origen del Bus Serial Universal se tuvieron en cuenta pautas importantes que se relacionan entre sí. La conexión de un teléfono al PC. Debido a que se necesitaban transmitir señales de audio y de voz el USB proporcionaba el ancho de banda adecuado para esta función. El uso sencillo de utilizar un solo puerto de

las mismas características para cualquier componente externo que se fuera a conectar al PC Y la facilidad de expandir los puertos para conectar automáticamente numerosos componentes con detección automática. La mayoría de los componentes que existen con comunicación USB brindan los drives adecuados para su instalación. [8]

5.1.3 Ventajas

- Bajo costo para aplicaciones que demandan por encima de los 12Mbps.
- Cómoda integración de dispositivos de tecnologías y fabricantes diferentes.
- Mayor velocidad de transferencia de datos.
- Adaptación permanente de cualquier dispositivo extra de nueva tecnología.

5.1.4 Aplicación

El USB permite acceder a casi todos los dispositivos que requieran como máximo una velocidad de 12Mbps (velocidades altas), al igual que velocidades medias de cualquier componente externo esencial (teclado, modem, webcam, etc.), pero su característica primordial es que todos los dispositivos cuentan con el mismo conector ya sea macho o hembra lo que facilita su adaptación.

Las velocidades medias dentro de este bus son del tipo isocrónicas, y las velocidades bajas son asincrónicas.

Dentro de sus grandes aplicaciones esta el suplir aquellas comunicaciones tanto serial como paralelo que en su momento fueron acogidas en todos los componentes externos. Además de lograr sobrepasar sus ventajas. [8]

5.1.5 Características

Todos los dispositivos USB deben tener el mismo tipo de cable y el mismo tipo de conector, sin depender de la función que cumplan.[8]

- El ordenador debe identificar automáticamente un dispositivo agregado mientras trabaja y configurarlo. De no ser así cada componente cuenta con sus driver de funcionamiento.
- Los dispositivos pueden ser también desconectados mientras el ordenador está en uso.
- Deben poder compartir un mismo bus tanto dispositivos que requieren de unos pocos Kbps como los que requieren varios Mbps.
- Más de 127 dispositivos diferentes pueden estar conectados simultáneamente y operando con una misma computadora sobre el Bus Serial Universal.
- El bus debe permitir periféricos multifunción, es decir aquellos que pueden realizar varias tareas a la vez, como lo son algunas impresoras que adicionalmente son fotocopadoras y máquinas de fax.
- Capacidad para manejo y recuperación de errores producidos por un dispositivo cualquiera.
- Bajo costo.

5.1.6 Dispositivos

Una gran variedad de dispositivos pueden ser conectados a un PC, debido a que la mayoría de ellos tiene algo en común, como lo son el cable y el conector USB los cuales son iguales para una numerosa cantidad de aplicaciones.

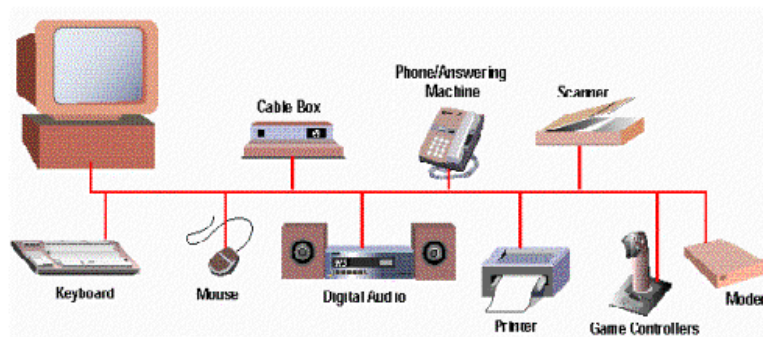


Figura 28. Típicas funciones de un sistema USB [8]

La aparición del USB no implica el fin de la interfaz RS232 y paralelo ya que los PC que estarán disponibles en el mercado seguirán manteniendo estos puertos con sus características normales, sin embargo más adelante desaparezcan poco a poco con el transcurrir del tiempo. De todas formas ya existen conversores RS232 y paralelos a USB.

5.1.7 Transmisión

Asincrónica: Esta transmisión consiste en que el receptor y el transmisor dispongan de relojes que funcionen a la misma frecuencia para posibilitar una transmisión exitosa. por lo cual, cuando uno de ellos desea transmitir, prepara un grupo de bits encabezados por un bit conocido como de arranque, un conjunto de 7 u 8 bits de datos, un bit de paridad (para control de errores), y uno o dos bits de parada.

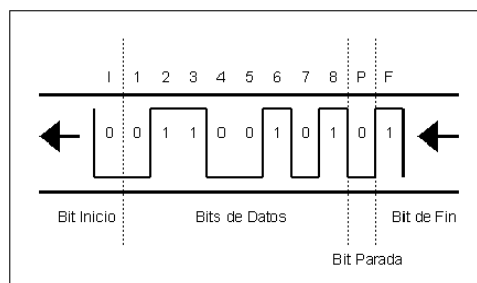


Figura 29. Esquema de la transmisión asíncrona [8]

Sincrónica: En este tipo de transmisión, el sincronismo viaja en la misma señal, de esta forma la transmisión puede alcanzar distancias mucho mayores como también un mejor aprovechamiento de canal. En la transmisión sincrónica, los grupos de datos o paquetes están compuestos por 128 bytes, 1024 bytes o más, dependiendo de la calidad del canal de comunicaciones.

Las transmisiones sincrónicas ocupan en la actualidad gran parte del mundo de las comunicaciones seriales, especialmente las que emplean el canal telefónico.

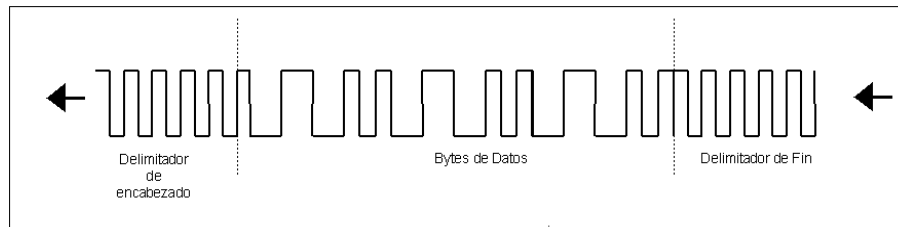


Figura 30. Transmisión sincrónica [8]

5.2 Circuito Integrado FT232BM USB UART (USB-Serial)

EL FT232BM es la 2ª generación del circuito integrado USB UART de FTDI. Este dispositivo no sólo agrega funcionalidad extra a su predecesor FT8U232AM y reduce la cantidad de componentes externos, sino que también mantiene un alto grado de compatibilidad de patas con el original, haciéndolo fácil de mejorar o reducir costos de los diseños existentes y aumentar el potencial para usar el dispositivo en nuevas áreas de aplicación.¹

VER **ANEXO C.CIRCUITO INTEGRADO FT232BM UTILIZADO EN CONVERTOR SERIAL-USB**

¹ Tarjeta Adaptadora para Puerto Serial desde Puerto USB. (s.f.). Recuperado el 3 de marzo de 2009 de <http://www.clubse.com.ar/DIEGO/NOTAS/3notas/nota01-1.htm>

5.3 Comunicación Serial

La comunicación serial es un protocolo muy común para la interacción entre dispositivos que se incluye de manera estándar en prácticamente cualquier computadora. La mayoría de las computadoras incluyen dos puertos seriales RS-232. La comunicación serial es también un protocolo común utilizado por varios dispositivos para instrumentación; existen varios dispositivos compatibles con GPIB que incluyen un puerto RS-232. Además, la comunicación serial puede ser utilizada para adquisición de datos si se usa en conjunto con un dispositivo remoto de muestreo. La mayoría de los equipos que requieren de un control por computadora manejan este tipo de interacción.

El concepto de comunicación serial es sencillo. El puerto serial envía y recibe bytes de información un bit a la vez. Aun y cuando esto es más lento que la comunicación en paralelo, que permite la transmisión de un byte completo por vez, este método de comunicación es más sencillo y puede alcanzar mayores distancias.

Típicamente, la comunicación serial se utiliza para transmitir datos en formato ASCII. Para realizar la comunicación se utilizan 3 líneas de transmisión: (1) Tierra (o referencia), (2) Transmitir, (3) Recibir. Debido a que la transmisión es asíncrona, es posible enviar datos por una línea mientras se reciben datos por otra. Existen otras líneas disponibles para realizar *handshaking*, o intercambio de pulsos de sincronización, pero no son requeridas. Las características más importantes de la comunicación serial son la velocidad de transmisión, los bits de datos, los bits de parada, y la paridad. Para que dos puertos se puedan comunicar, es necesario que las características sean iguales.

Velocidad de transmisión (*baud rate*): Indica el número de bits por segundo que se transfieren, y se mide en baudios (*bauds*). Por ejemplo, 300 baudios representa 300 bits por segundo. Cuando se hace referencia a los ciclos de reloj se está hablando de la velocidad de transmisión.

Bits de datos: Es la cantidad de bits en la transmisión. Cuando la computadora envía un paquete de información, el tamaño de ese paquete no necesariamente será de 8 bits. Las cantidades más comunes de bits por paquete son 5, 7 y 8 bits. El número de bits que se envía depende en el tipo de información que se transfiere. Un paquete se refiere a una transferencia de byte, incluyendo los bits de inicio/parada, bits de datos, y paridad.

Bits de parada: Usado para indicar el fin de la comunicación de un solo paquete. Los valores típicos son 1, 1.5 o 2 bits. Debido a la manera como se transfiere la información a través de las líneas de comunicación y que cada dispositivo tiene su propio reloj, es posible que los dos dispositivos no estén sincronizados. Por lo tanto, los bits de parada no sólo indican el fin de la transmisión sino además dan un margen de tolerancia para esa diferencia de los relojes.

Paridad: Es una forma sencilla de verificar si hay errores en la transmisión serial. Existen cuatro tipos de paridad: par, impar, marcada y espaciada. La opción de no usar paridad alguna también está disponible. Para paridad par e impar, el puerto serial fijará el bit de paridad (el último bit después de los bits de datos) a un valor para asegurarse que la transmisión tenga un número par o impar de bits en estado alto lógico. La paridad marcada y espaciada en realidad no verifican el estado de los bits de datos; simplemente fija el bit de paridad en estado lógico alto para la marcada, y en estado lógico bajo para la espaciada. Esto permite al dispositivo receptor conocer de antemano el estado de un bit, lo que serviría para determinar si hay ruido que esté afectando de manera negativa la transmisión de los datos, o si los relojes de los dispositivos no están sincronizados.²

²Comunicación Serial: Conceptos Generales.(s.f.). Recuperado el 3 de marzo de 2009 de <http://digital.ni.com/public.nsf/allkb/039001258CEF8FB686256E0F005888D1>

5.3.1 Puerto serie RS232

RS-232 (Estándar ANSI/EIA-232) es el conector serial hallado en las PCs IBM y compatibles. Es utilizado para una gran variedad de propósitos, como conectar un ratón, impresora o modem, así como instrumentación industrial. Gracias a las mejoras que se han ido desarrollando en las líneas de transmisión y en los cables, existen aplicaciones en las que se aumenta el desempeño de RS-232 en lo que respecta a la distancia y velocidad del estándar.

SCI es un enlace serie asíncrono de 1962. También es conocido como UART (Universal Asynchronous receiver transmitter). Es full-duplex y sólo admite la transmisión-recepción entre dos elementos (punto a punto). La velocidad máxima de transmisión suele ser 256kbps con cables de 15m (50 pies). El protocolo es el de la norma RS-232 pero los niveles eléctricos difieren, por lo que es necesario un integrado adaptador de niveles tipo MAX232 para poder utilizar el SCI como puerto RS-232. Dicho puerto es también conocido como puerto COM o puerto serie.

5.3.2 Conector DB9

El conector DB9 (originalmente *DE-9*) es un conector analógico de 9 clavijas de la familia de conectores D-Subminiature (D-Sub o Sub-D).

El conector DB9 se utiliza principalmente para conexiones en serie, ya que permite una transmisión asíncrona de datos según lo establecido en la norma RS-232 (RS-232C).³



Figura 31. Conector DB9 macho y hembra [11]

³ Conector DB9. (s.f). Recuperado el 4 de marzo de 2009 de <http://es.kioskea.net/contents/elec/connecteur-prise-db9.php3>

Pines	Nombre
1	CD: Detector de transmisión
2	RXD: Recibir datos
3	TXD: Transmitir datos
4	DTR: Terminal de datos lista
5	GND: Señal de tierra
6	DSR: Ajuste de datos listo
7	RTS: Permiso para transmitir
8	CTS: Listo para enviar
9	RI: Indicador de llamada
	Protección

Tabla 3. Descripción de pines conector DB9 [11]

El conector macho (plug) y el hembra (socket) son algo diferentes. En el primero, la patilla 2 corresponde a la línea de recepción, y la 3 a la línea de transmisión. En el segundo ocurre al revés. De este modo, al interconectar dos dispositivos, si uno tiene conector macho y otro hembra, el cable a utilizar será un cable no cruzado. Si los dos dispositivos tienen el mismo tipo de conector entonces será necesario un cable cruzado, en el que el pin 3 de un extremo está conectado al 2 del otro y viceversa.

El conector RS-232 de nuestro PC es macho. Dado que los cables más comunes y fáciles de encontrar son los macho-hembra. Debido a esto el conector que se vaya a utilizar en el dispositivo conversor debe ser hembra.

5.4 Circuito integrado MAX232

El SONIC 137 utiliza la norma RS232, por lo que los niveles de tensión de los pines están comprendidos entre +15 y -15 voltios (o +9 y -9 voltios). El FT232BM normalmente trabaja con niveles TTL (0-5v). Es necesario por tanto incorporar un circuito que adapte los niveles:

El MAX232 dispone internamente de 4 conversores de niveles TTL al bus standard RS232 y viceversa, para comunicación serie como los usados en los ordenadores, el Com1 y Com2.

EL circuito integrado lleva internamente 2 conversores de nivel de TTL a rs232 y otros 2 de rs232 a TTL con lo que en total podremos manejar 4 señales del puerto serie del PC, por lo general las más usadas son; TX, RX, RTS, CTS, estas dos últimas son las usadas para el protocolo handshaking pero no es imprescindible su uso. Para que el MAX232 funcione correctamente es necesario poner unos condensadores externos. (Ver **Figura 32**)

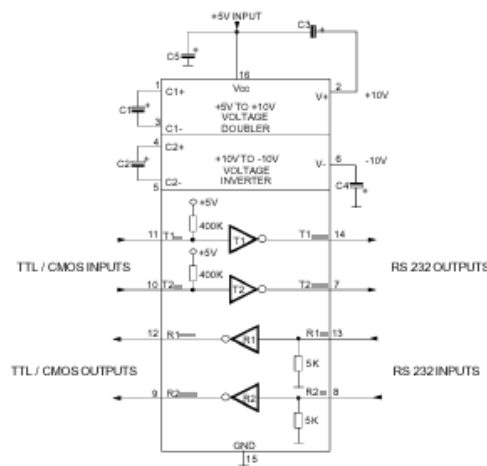


Figura 32. Integrado max232 con sus Condensadores Externos [12]

5.4.1 Funcionamiento

El estado dominante de la línea es '1', por lo que se dice que es un protocolo NRZ (No Return to Zero). Eléctricamente un '1' corresponde a un valor entre -3 y -15V, y un '0' con un valor entre +3 y +15V. (Ver **Figura 33**)

En primer lugar se manda un bit de start (0) que sirve al receptor para la sincronización. Posteriormente se envía el mensaje (1 byte) y un bit de stop (1). Para enviar el siguiente byte se repetiría el proceso. A este tipo de transmisión se le denomina 8N1.

A veces se añade un bit de paridad para comprobar si el dato se ha recibido correctamente. Su valor depende del byte enviado y del tipo de paridad.

- Paridad par: Su valor es 0 si el número de ceros enviados es par
- Paridad impar: Su valor es 0 si el número de ceros enviados es impar

Existen otras variaciones en el protocolo tales como la utilización de dos bits de stop, transmisión de datos de siete bits, etc. por lo que tendremos que verificar que las dos unidades utilizan el mismo protocolo. El 8N1 es el más utilizado.

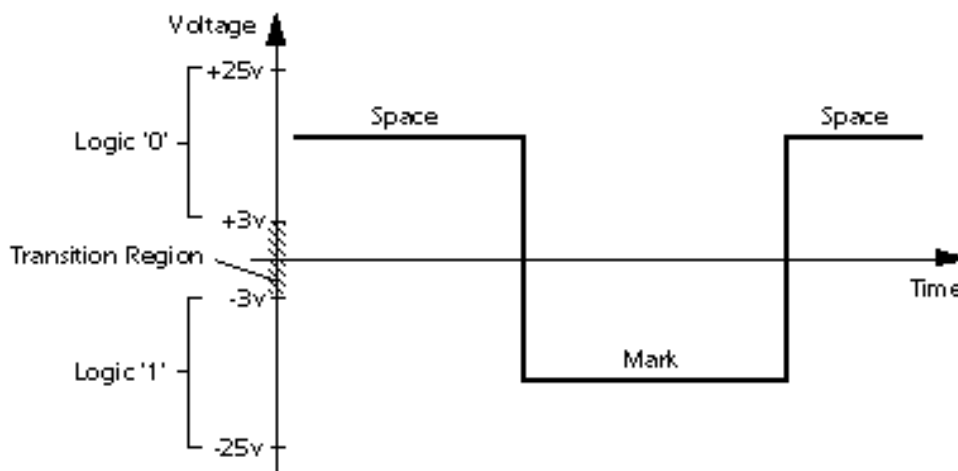


Figura 33.Niveles norma RS-232 [14]

Transmisión vía RS232 en modo 7 bits

A la hora de visualizar la transmisión con el osciloscopio hay que tener en cuenta dos cosas:

- Dado que primero se envía el bit de menos peso, el byte debe ser leído de derecha a izquierda.
- Los valores negativos de tensión corresponden a un '1', y los positivos a un '0'. Para no confundir los datos lo mejor es invertir el canal para ver los unos como tensiones positivas y los ceros como tensiones negativas.

El conversor debe contar con este integrado para poder transformar los niveles que arroja el FT232BM en el envío y la recepción de datos, en tensiones que el SONIC pueda reconocer con el fin de que devuelva una respuesta y así exista una comunicación opima.⁴

5.5 Diagrama Esquemático y PCB conversor RS232-USB

Ver **ANEXO D. DIAGRAMA ESQUEMATICO Y PCB CONVERTOR RS232-USB**

5.6 Instalación de los Drives para Conversor

Ver **ANEXO E. INTALACION DE LOS DRIVES PARA EL CONVERTOR**

⁴ Circuito MAX232. (s.f.). Recuperado el 4 de marzo de 2009 de http://www.geocities.com/alva_cesar/rs232/max232.html

6. PRUEBAS FINALES

6.1 Calibración del SONIC 137

La calibración se realiza configurando los valores contenidos en el manual, se conecta el palpador (Transductor) al Sonic, con suficiente acoplante sobre el material a medir, se visualizan los ecos en la pantalla de la interfaz del Sonic y modificando algunos submenús se busca obtener la mejor presentación de los ecos.



Figura 34. Imagen Pruebas de Calibración [Autores]

La interfaz permite la ubicación inicial de los ecos, el indicador mostrado en la **Figura 35** representa el valor de espesor del menor valor calibrado, con “DELAY” se modifica la posición del eco, hasta obtener un valor acertado del espesor real.

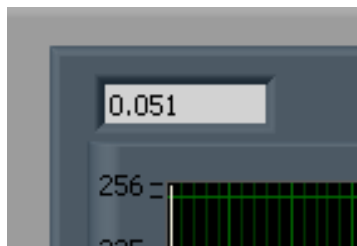


Figura 35. Visualización del Espesor de Menor Valor [Autores]

El valor mayor de espesor se modifica por el cambio de los valores en “VELOCIDAD” (Ver **Figura 36**)

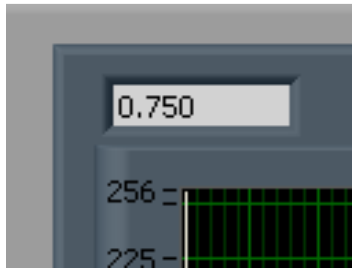


Figura 36. Visualización del Espesor de Mayor Valor [Autores]

La interfaz permite la modificación de la compuerta GATE2, para que la calibración del Sonic se realice bien esta barra debe cortar el eco de mayor amplitud (diferente al eco inicial), se debe considerar que para las variaciones de amplitud siempre se debe cortar la gráfica.

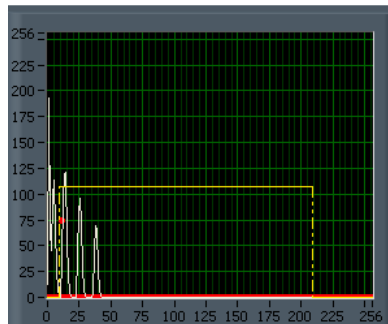


Figura 37. Ubicación de la Compuerta en el Menor Espesor [Autores]

Cuando el palpador se ubica en el espesor mayor el ancho debe ser modificado para que la compuerta corte el eco, en ocasiones es necesario modificar el nivel.

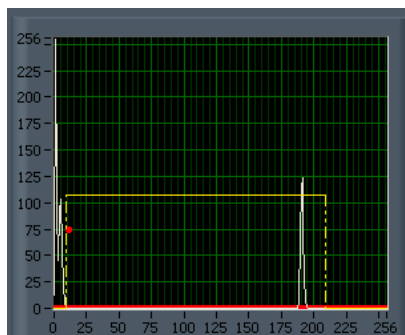


Figura 38. Ubicación de la Compuerta en el Mayor Espesor [Autores]

Después de la activación del indicador de espesor, se presenta una leve saturación del Sonic, para corregir los errores en la interfaz fue necesario aumentar el tiempo en actualizar la grafica, el valor de espesor que muestra el Sonic también es visualizado en la interfaz. (Ver **figura 39**)



Figura 39. Visualización del Espesor del Material [Autores]

La calibración y verificación de los valores de espesor se realizaron con la escalerilla presentada en la **Figura 40**.

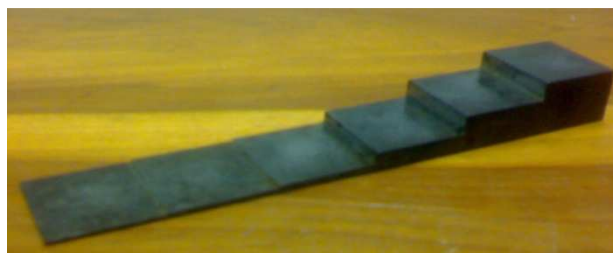


Figura 40. Escalerilla utilizada para la calibración [Autores]

	A	B	C	D	E	F
ESPESOR(in)	0.750	0.500	0.250	0.100	0.070	0.050

Tabla 4. Valores de espesor de la escalerilla [Autores]

Después de verificar la calibración del equipo es posible realizar las pruebas directamente con el Sonic, independiente de la interfaz, el daño en la pantalla CRT limita la calibración, pero no la activación de las alarmas del Sonic para determinar las variaciones del material por problemas en corrosión.

6.2 Pruebas con detección de fallas

Para estas pruebas determinamos las fallas presentadas en la placa de acero (Ver **Figura 41**), estas son perforaciones que fueron realizadas a diferentes distancias y atraviesan la placa.



Figura 41. Detección de Falla en Placa de Acero [Autores]

Después de realizar la calibración, se caracterizó la placa y permitió determinar las distancias en fallas y medidas propias del material.

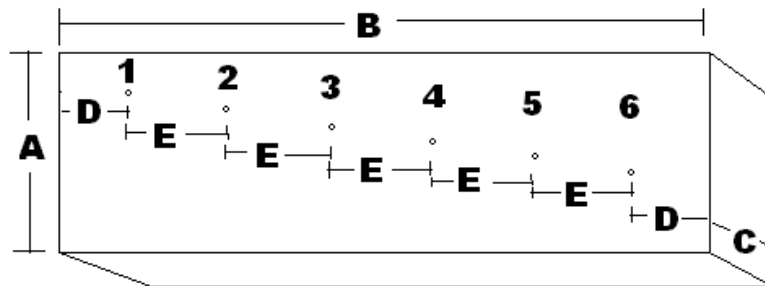


Figura 42. Esquema de las Fallas en la Placa de Acero [Autores]

Las perforaciones se realizaron tomando como referencia el centro de la circunferencia para dar distancias en valores estándar de pulgadas (ver **Tabla 4**), la diferencia presentada en los valores de profundidad de falla en la **Tabla 5**, es el equivalente al radio de la perforación, el Sonic mide la distancia hasta límite del material, lo que equivale a la parte superior de la circunferencia, para la placa de acero de la Escuela de Ingeniería Metalúrgica, las perforaciones tienen un radio de 0.020 in.

DISTANCIAS (in)					PROFUNDIDAD DE LA FALLA (in)					
A	B	C	D	E	1	2	3	4	5	6
3.00	12.00	1.250	1.00	2.00	0.250	0.500	0.750	1.00	1.250	1.500

Tabla 5. Valores de Referencia Ofrecidos por el Fabricante

DISTANCIAS (in)					PROFUNDIDAD DE LA FALLA (in)					
A	B	C	D	E	1	2	3	4	5	6
2.945	11.836	1.241	0.980	2.0	0.230	0.480	0.730	0.980	1.230	1.480

Tabla 6. Valores Tomados con la Calibración Realizada en la Interfaz [Autores]

También se utilizó una escalera de acero, con porosidad y cada escalón presenta una falla, se determinaron los valores de espesor y de la falla en la **Tabla 6**, este material impide que se obtengan ecos con buena amplitud, porque la onda de ultrasonido no atraviesa con facilidad.



Figura 43. Escalera de Acero no Caracterizada [Autores]

	A	B	C	D	E	F	G
ESPESOR(in)	1.172	0.984	0.797	0.602	0.367	0.250	0.133
FALLA(in)	0.591	0.487	0.375	0.248	0.150	0.109	0.073

Tabla 7. Valores Medidos con el Sonic. [Autores]



Figura 44. Detección de Falla con Escalerilla [Autores]

Para la determinación de la falla es necesario configurar una compuerta, debe ubicarse entre el eco inicial y antes del eco que representa el espesor. (Ver **Figura 45**)

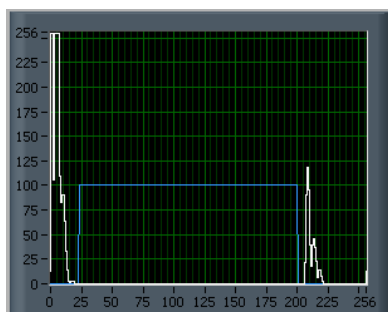


Figura 45. Gráfica del Sonic sin presencia de falla [Autores]

Cuando se presenta la falla la gráfica muestra un pequeño eco ubicado dentro de la compuerta destinada para detectarla (ver **Figura 46**)

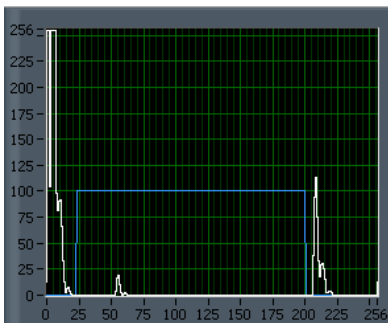


Figura 46. Gráfica del Sonic Cuando se presenta la falla [Autores]

Todos los ecos presentan decaimiento exponencial debido a que decrece la densidad de energía en la onda de ultrasonido (ver **Figura 47**) se presenta en el eco del espesor y el eco de la falla.

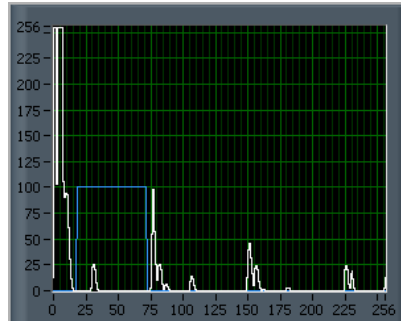


Figura 47. Decaimiento Exponencial de la Amplitud de la falla [Autores]

6.3 Medición con haz Angular



Figura 48. Pruebas con Transductor de Haz Angular [Autores]

El indicador de espesor permite visualizar la distancia que recorre la onda de ultrasonido o la profundidad, cuando se quiere detectar una falla.

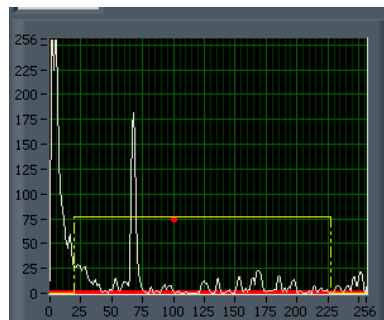


Figura 49. Grafica Haz Angular [Autores]

7. TARJETA DE ADQUISICIÓN DE DATOS

Una Tarjeta De Adquisición de Datos (TAD) es un equipo que permite conectar un computador con el mundo externo para poder así interactuar con él y poder realizar mediciones, análisis y control de un sin número de variables análogas que se encuentran en el mundo real como son la temperatura, humedad, presión, etc. Además se pueden conectar sensores y actuadores para generar una gama de aplicaciones que van desde el área industrial, la biometría, medicina, monitoreo, seguridad, etc.

De este modo se convierte en un equipo indispensable en cualquier aplicación industrial, tecnológica, científica y educacional.

7.1 Programación del PIC 18F2550

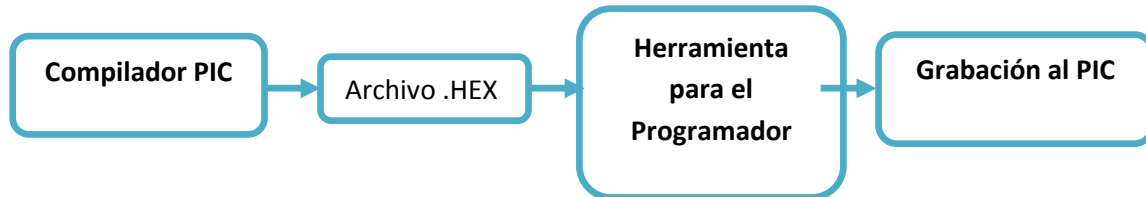


Figura 50. Secuencia de programación [Autores]

7.1.1 Compilador PIC

Se optó por la utilización de este software debido a su facilidad de uso, sus diversos archivos que proporciona, por ser una versión gratuita y sobre todo porque permite compilar programación en C para los microcontroladores, en este caso PIC18F2550.

Las características principales de la programación del PIC fueron: La asignación de los pines para el RS232 (tanto la entrada como la salida), la configuración USB

y la descripción del dispositivo, y por ultimo las especificaciones para la programación en LABVIEW.

La programación del PIC consiste en dos partes, un archivo .C que como su extensión lo dice esta en lenguaje C donde se establecen las acciones que el micro va a realizar, como el envío de datos, la recepción de datos, asignación de puertos, etc. Y un archivo .h que se encuentra en lenguaje de ensamblador donde se ajusta la configuración USB, las características del dispositivo y donde se establece el reconocimiento para la interacción con Windows. Este ultimo tiene que ser incluido por el archivo .C a la hora de la compilación para asegura el adecuado funcionamiento del dispositivo.

Ver más información en **ANEXO F. COMPILADOR PIC**

- Descripción del código

Consultar **ANEXO G. PROGRAMACIÓN PIC 18F2550**

```
#include <18F2550.h>
```

Añade las funciones y variables para poder compilar nuestro microcontrolador, en realidad no es más que un archivo que traduce nuestro código en C a un nivel ensamblador.

```
#fuses
```

```
HSPLL,NOWDT,NOPROTECT,NOLVP,NODEBUG,USBDIV,PLL5,CPUDIV1,VRE  
GEN
```

Constantes definidas por CCS, como no activar *watchdog*, no proteger el código, no usar voltaje bajo de 3.3V., CCS tiene muchas constantes para definir diferentes funciones y estados del microcontrolador.

```
#use delay(clock=48000000)
```

Define la velocidad del reloj del microcontrolador en Hz.

```
#use rs232(baud=9600, xmit=PIN_C6, rcv=PIN_C7)
```

Declaración para utilizar la UART del micro, donde se define la velocidad, patillas de transmisión y recepción y periférico que se comunica. Es importante saber que una de las ventajas de CCS es que puede simular por código diferentes UART con lo que podemos comunicarnos con un mismo PIC con diferentes periféricos.

7.1.2 Herramienta para el programador



winPic800

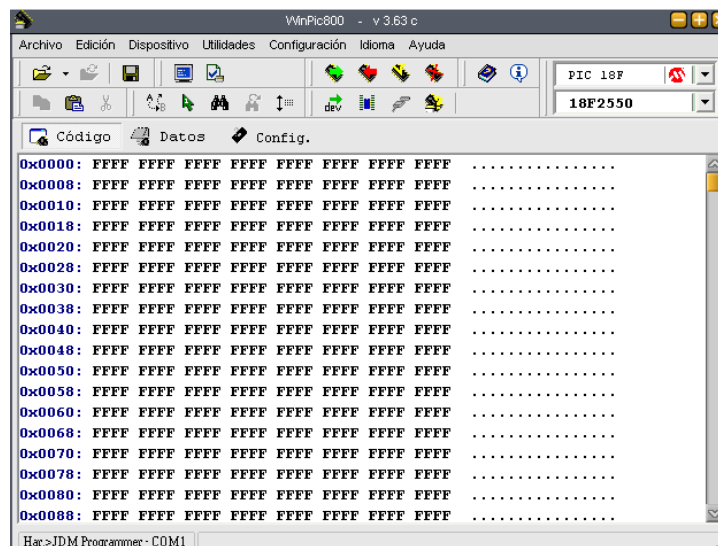


Figura 51. Entorno programa WinPic800 [Autores]

El programador *WINPIC800* es un software desarrollado por *Sisco Benach*, y está diseñado para programar microcontroladores y memorias de tipo *EEPROM* de varias categorías y marcas, entre ellas destacan *MICROCHIP*, *ATMEL*, etc.

Este software se actualiza con bastante frecuencia y se le amplían sus características, así como el incremento del número de dispositivos soportados.

Para programar el micro es necesario contar con el archivo “*.HEX*” que este caso no lo suministra el *C compiler*. Antes que nada debemos ajustar el *winpic800* para que reconozca el programador (*JDM*) y el puerto por el cual se va hacer la

comunicación. El software también cuenta con una prueba de hardware para verificar que el programador está arrojando las tensiones necesarias para la grabación.

7.1.3 Grabación al PIC

- *Programador JDM*

Los microcontroladores en general, y los de la empresa *Microchip* en particular, necesitan de un circuito electrónico auxiliar que nos permita transferirles desde el ordenador el programa que hemos escrito para ellos.

Así es que en el mercado es posible conseguir “programadores” de PICs con conexión para puerto USB, paralelo o serie (RS-232). Por razones de simplicidad en el diseño y por ser seguramente una de las alternativas más económicas, en este proyecto se construyó el *programador JDM*, con conexión serial.

No solo nos permitirá grabar datos en los microcontroladores más comunes, de 8, 18, 28 y 40 pines, sino que también lo podremos utilizar para grabar y leer varios tipos de memorias.

Los microcontroladores de *Microchip* (PICs) se programan mediante un protocolo tipo serie. Se necesitan dos tensiones de alimentación para poder llevar a cabo la programación: una de 4.5v a 5.5v (Vdd) y otra comprendida entre 12v y 14v (MCLR), que es la que indica al PIC que va a ser programado, para que cambie la función que realizan los pines I/O implicados en la programación.

VER **ANEXO H.** ESQUEMATICO LAYOUT E IMPRESO DEL PROGRAMADOR

7.2 Características generales

La tarjeta *TAD_EP* está constituida por un microcontrolador *PIC 18F2550* que cuenta con un entorno USB para su comunicación con el PC. Se ha escogido una comunicación bidireccional masiva, bulk transfers, entre el PIC y el Software del PC vía USB 2.0 a full speed, para aprovechar al máximo la velocidad de transferencia.

La tarjeta cuenta con cuatro entradas digitales, cuatro salidas digitales y una entrada analógica con resolución de 8 bits.

7.2.1 Sistema operativo

La *TAD_EP* está diseñada para que opere en sistemas operativos tales como Windows XP/2000 e inferiores ya que se desarrollo un código en PicC C Compiler y un software en labview para lograr la comunicación de la tarjeta con el PC y permitir la interacción del usuario. Como elementos adicionales se utilizaron las herramientas suministradas por la página de *microchip* tales como los “drivers” (instaladores) para Windows y la “dll” para labview.

7.2.2 Aplicación para el control de la tarjeta TAD_EP

La aplicación ha sido desarrollada usando *Labview Versión 8.2 (Evaluación)* y *Labview Versión 7.1 de National Instruments* para ser usada en equipos ejecutando el sistema operativo Windows XP.

7.3 Funcionamiento general de la aplicación

Dentro del CD que se encuentra con la *TAD_EP* existen tres carpetas que son necesarias para el adecuado funcionamiento tanto del hardware como del software. Estas tres carpetas funcionan igual para las dos versiones de labview en que se de desarrollo la aplicación.



Figura 52. Instaladores y Archivos para hacer Uso de Tarjeta [Autores]

7.3.1 Conexión

VER **ANEXO I.** CONEXIÓN DE LA TARJETA DE ADQUISICIÓN DE DATOS E INSTALACIÓN DEL RUN TIME

7.3.2 Interfaz

7.3.3 Descripción Del Programa En Labview

Fue necesario hacer dos *subVI*, uno de lectura y otro de escritura que permitiera el envío y la recepción de datos vía USB. (Ver **Figura 53**)

Dentro de estos *subVI* se establecen las normas de comunicación para poder interactuar desde labview con la tarjeta de adquisición de datos. Lo primero que se necesita para el adecuado funcionamiento del programa es instalar o llamar desde labview los *drivers* para Windows, que proporciona la pagina de *microchip* (*mpusbapi.dll*), los *subVI* fueron configurados para realizar esta instalación. Estos *drivers* establecen los valores de *VID&PID* los cuales representan la licencia o el permiso para que Windows reconozca el hardware conectado al PC.

Los valores de *VID&PID* deben coincidir con los establecido en el programa del microcontrolador, de eso depende el adecuado funcionamiento tanto del hardware como del software.



Figura 53. *SubVI* para Leer y Escribir por USB [Autores]

Cuando se desarrollo la programación del micro se tuvo en cuenta que debería ser similar a la programar en labview, ya que de ello dependía la interacción correcta entre los dos software.

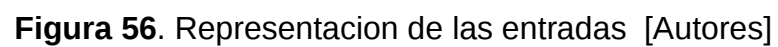
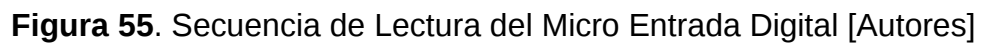
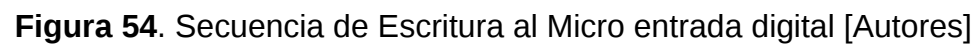
Se plantearon tres posibles casos para la toma o recepción de datos con la tarjeta,

- Entradas digitales
- Entrada analógica
- Salidas digitales.

Los casos fueron establecidos con la estructura “case” desde labview al igual que en la programación en C del micro.

Entrada digital

En este primer caso lo que se hace es escribir desde labview una sentencia junto con el valor de *VID&PID*, para que después sean devuelto los datos de entrada que se necesitan leer, seguido de esto se muestra dicha lectura con el encendido y apagado de una serie de leds creados para la interfaz. Debido a que no se puede leer y escribir al mismo tiempo fue necesario colocar estos *subVI* por separado dentro de un “sequence” (Ver **Figura 54**) el cual me organiza el orden de ejecución, es decir primero escribir y luego leer.



Entrada analógica

En el segundo caso se aplica la misma metodología que se practico en las entradas digitales, debido a que también se trata de una entrada. La única diferencia importante es que la representación de una entrada analógica es el cambio de una señal de 0 a 255, esto se debe a la conversión que realiza el micro para adquirir este tipo de entrada.

Para la visualización fue necesario hacer una conversión que permitiera eliminar valores altos que enviaba el microcontrolador, ya estos datos los envía cada 8 bits que es la resolución del conversor interno. (Ver **Figura 57**)

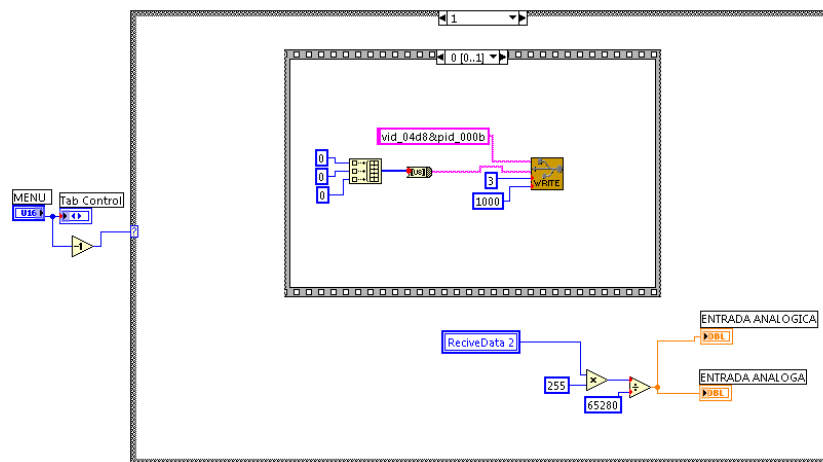


Figura 57. Secuencia de Escritura al Micro Entrada Analógica [Autores]

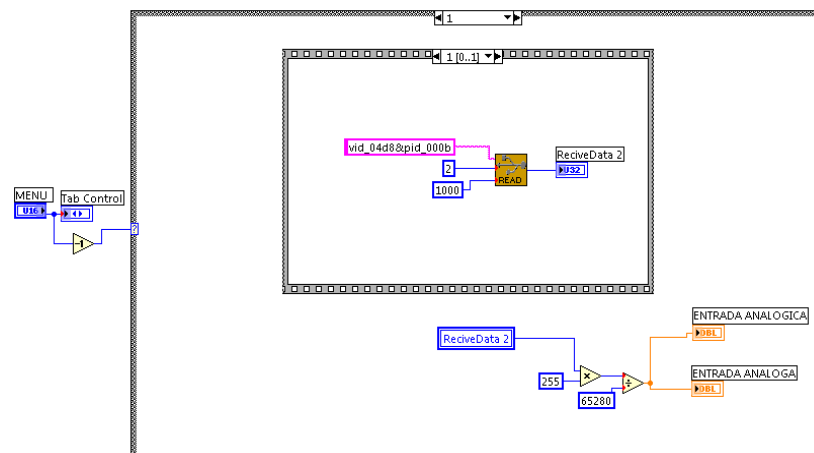


Figura 58. Secuencia de Lectura del Micro Entrada Analógica [Autores]

Para el tercer caso fue un poco diferente la metodología ya que se trata de una salida, por ello solo era necesario escribir desde labview para que el microcontrolador reconociera la salida y la mostrara en una serie de leds.

The screenshot shows a Proteus simulation environment with a circuit diagram for a 4-bit LED display. The circuit includes a 'MENU' button, a 'Tab Control' block, a 4-bit counter (0-15), four 7-segment displays (led 1 to led 4), a 4-bit adder, a 4-bit register, a 4-bit shift register, and a 4-bit output register. The circuit is connected to a 5V supply and a 1000Hz clock source.

7.3.4 Manejo de la interfaz

70



Figura 60.Contenido Carpeta del Ejecutable [Autores]

Tarjeta con Labview 8.2 (licencia Evaluación)

Dentro de la interfaz existe un menú desplegable donde se puede seleccionar la función que se quiera hacer.

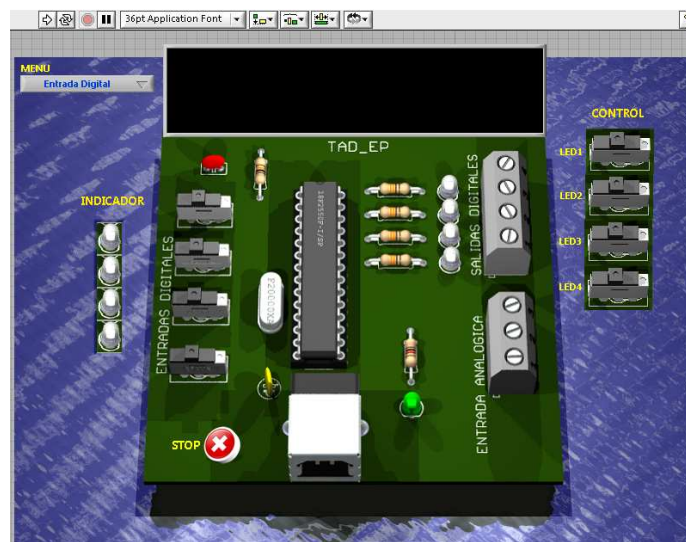


Figura 61.Vista Principal Interfaz en Labview 8.2 (Licencia Evaluación) [Autores]

El menú entrada digital cuenta con una serie de leds que van a ser accionados desde la *TAD_EP* con unos interruptores, existen 4 interruptores que me permiten visualizar cuatro entradas digitales.



Figura 62. Indicador de Entradas Digitales Labview 8.2 (Licencia Evaluación)
[Autores]

El menú entrada analógica cuenta con una pantalla donde se visualiza las variaciones que se ejecutan desde la TAD_EP con un potenciómetro, existen un solo potenciómetro para la entrada analógica y este cuenta con una resolución de 8 bits.



Figura 63. Visualizador Entrada Analógica Labview 8.2 (Licencia Evaluación)
[Autores]

El menú salida digital cuenta con una serie de leds que accionaran dándole click sobre cada uno de ellos los leds que se encuentran en la TAD_EP, existen cuatro leds que representan las cuatro salidas digitales.

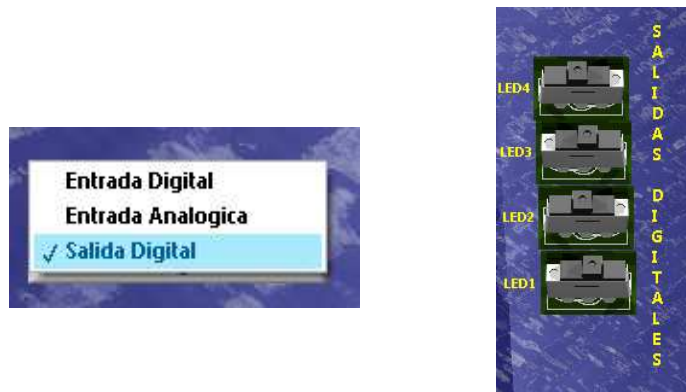


Figura 64.Control de Salidas Digitales Labview 8.2 (Evaluación) [Autores]

Para parar la aplicación debe darle clic al botón llamado stop o simplemente en la barra de labview darle clic al botón rojo (ver **Figura 65**), estas opciones terminan la aplicación inmediatamente.



Figura 65. Control Para Detener la Aplicación [Autores]

Tarjeta con Labview 7.1

La interfaz grafica desarrollada en Labview cuenta con una imagen en 3D de la TAD_EP para hacer más didáctica la interacción con el usuario.

Esta interfaz cuenta con cuatro menús para su utilización, en cada uno de ellos existen unas imágenes que permiten que el usuario pueda entender lo que se va a desarrollar.

El primer menú es *TAD_EP*, donde se puede ver la imagen en 3D de la tarjeta de adquisición de datos junto con los nombres de los fabricantes.

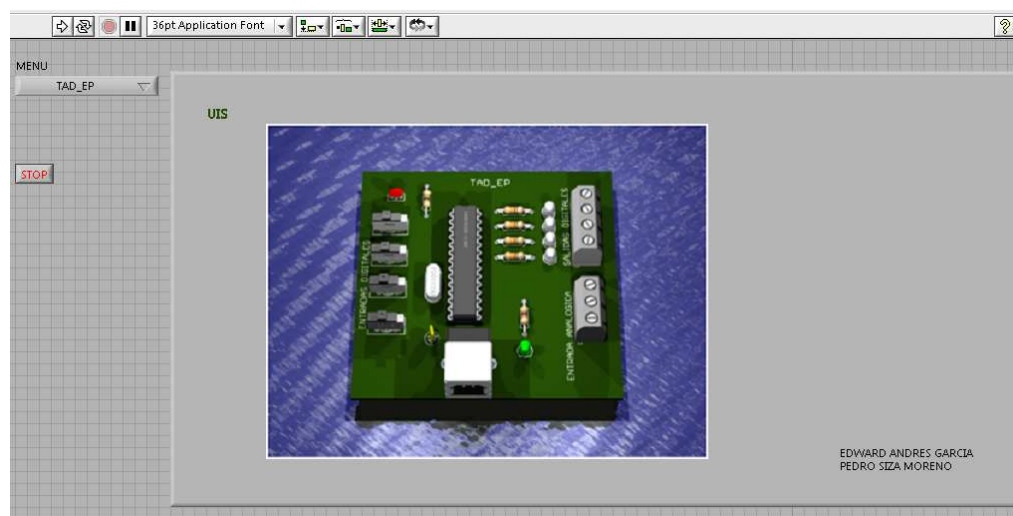


Figura 66. Vista Principal Interfaz en Labview 7.1 [Autores]

El segundo menú es ENTRADAS DIGITALES, aquí se especifica con la imagen en 3D de la TAD_EP donde deben accionarse dichas entradas (interruptores en este caso) desde el hardware. Este entorno cuenta con cuatro leds que se prenderán y apagaran dependiendo de la acción del usuario a su vez se visualizara el cambio en los interruptores que se aprecian en la imagen.

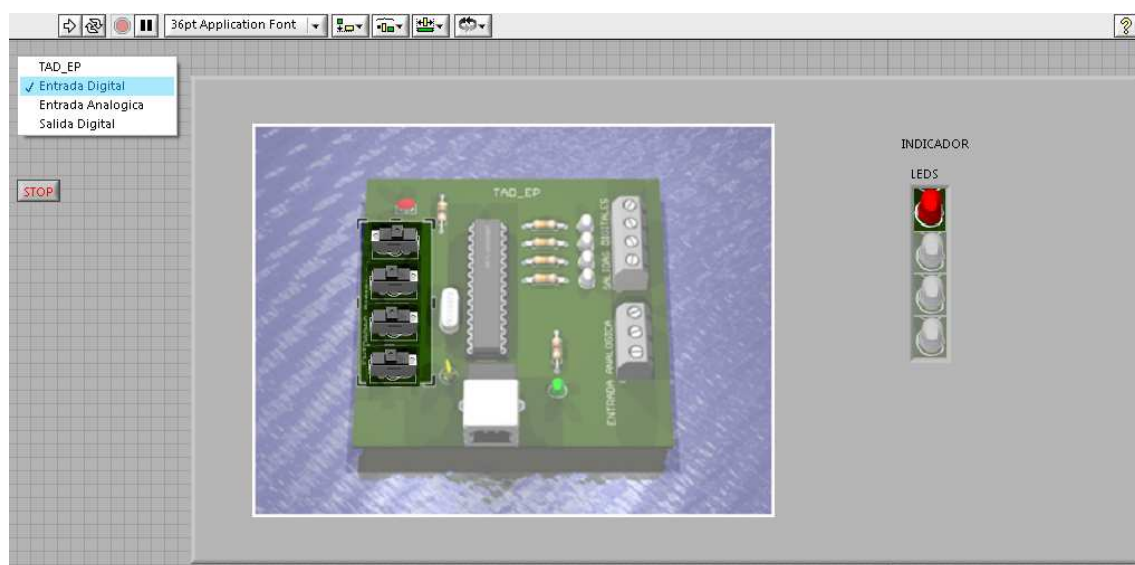


Figura 67. Indicador Entradas Digitales Labview 7.1 [Autores]

El tercer menú es ENTRADA ANALÓGICA, al igual que en el anterior menú aparecerá una imagen que especifica la ubicación de esta entrada en el hardware. Este entorno cuenta con una grafica en donde se podrá ver cualquier cambio que se efectuó en el potenciómetro o cualquier otra entrada análoga que se adecue, también existe un indicador que va de 0 a 255 para rectificar el cambio que ejerce la entrada análoga. (ver **Figura 68**)

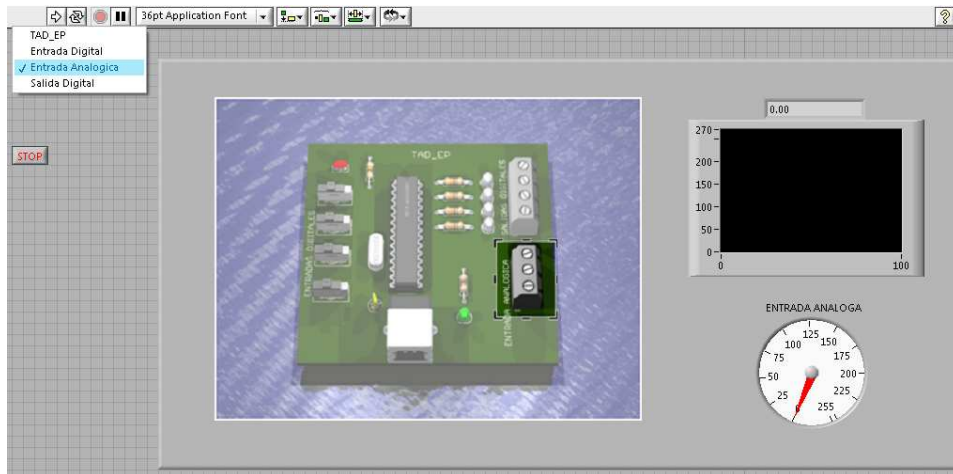


Figura 68. Visualizador Entrada Analógica Labview 7.1 [Autores]

El cuarto menú es SALIDAS DIGITALES, en este entorno además de indicar donde se van a ver reflejadas esas salidas en el hardware con el cual el usuario está interactuando también cuenta con un control (para este caso se escogió la imagen de un interruptor para accionar los leds) que permitirá encender o apagar los leds que se encuentran en la *TAD_EP*. (ver **Figura 69**)

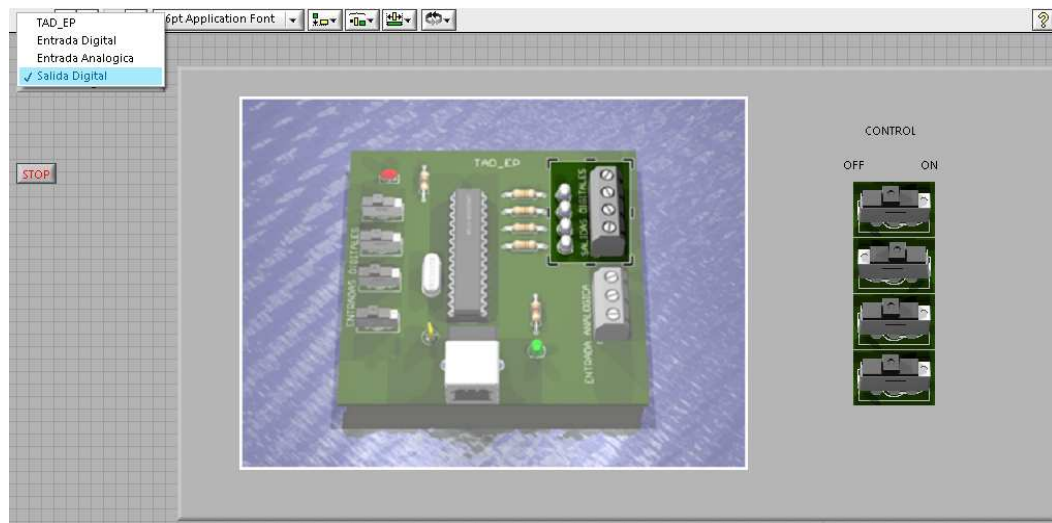


Figura 69. Control de Salidas Digitales Labview 7.1 [Autores]

VER **ANEXO J.** ESQUEMÁTICO PCB Y HARDWARE DE LA TARJETA DE ADQUISICIÓN DE DATOS

8. CONCLUSIONES Y OBSERVACIONES

El Sonic 137 permite la comunicación mediante puerto serial, cuenta con comandos de tres letras que son utilizados para la escritura y lectura de datos, en el presente rediseño fueron utilizados para la escritura desde el PC y la actualización de los valores iniciales del Sonic.

La velocidad de envío de información que ofrece el Sonic, limita la cantidad de ejecuciones en LABVIEW, para ello fue necesario la utilización de algunos retardos de manera que el Sonic no produjera problemas internos como el envío erróneo de la gráfica y el parpadeo continuo del indicador de espesor del Sonic.

La ejecución alternada de lectura y escritura se realizó con el conocimiento de las estructuras según la prioridad para LABVIEW, se pensó en la organización de tal forma que tanto el Sonic como la interfaz funcionaran de la manera más eficiente y reducir los bloqueos que presenta el Sonic.

El Submenú “ANGLE”, configura el ángulo de envío de la onda de ultrasonido, no permite la modificación mediante Labview, el comando de lecto-escritura mostrado en el manual no concuerda con el necesario para la comunicación, para este caso se agregó un mensaje de advertencia, para que el usuario realice la configuración desde el Sonic.

El Rediseño se oriento en asemejar el manejo del Sonic, pero realizado desde LABVIEW y permite que los estudiantes de Ingeniería Metalúrgica calibren el equipo, de la misma forma que se realizaba cuando no se tenía el problema en la pantalla CRT, permitir la realización de pruebas en el Sonic que anteriormente no eran posibles y contribuir a su formación en manejo de estos equipos utilizados a

nivel industrial, para la detección de discontinuidades por corrosión, grietas y defectos en soldaduras.

El programador del PIC se desarrolló con comunicación serial debido a su fácil adaptación y simplicidad de uso, este cuenta con un led que indica el inicio y la finalización de la transferencia de datos a la memoria del *microcontrolador*. Se utilizó el programador JDM que permite programar el *PIC18F2550* con la asistencia de *winpic800*.

Para la programación del micro se crearon dos códigos, uno en *Assembler* y otro en *lenguaje C*. El primero establece el nombre del dispositivo cuando lo reconoce el sistema operativo que se esté usando (Windows o Linux), y el valor de *VID&PID* para el reconocimiento de drives. El segundo proporciona las funciones que desempeña el micro con sus adecuados comandos de ejecución. Las estructuras utilizadas en la programación del PIC debe ser las mismas empleadas en Labview, es decir deben concordar los comandos adaptados para el acople del sistema. (Estructuras tales como *WHILE*, *FOR*, *IF*, *CASE*, etc).

El *PIC 18F2550* cuenta con un modulo de comunicación USB, esto hace mas fácil la adaptación al PC, además de ser más pequeño que muchos micros de su familia que cumplen las mismas funciones. El micro tiene los comandos necesarios para la configuración del modulo USB de tal forma que se pueden ajustar a las necesidades del programa que se quiera desarrollar.

Para el funcionamiento correcto de la comunicación entre el micro y el PC los valores de *VID&PID* establecidos en el código *Assembler* deben ser idénticos a los especificados en el software que controla la aplicación (En nuestro caso Labview), ya que estos valores proporcionan la licencia o permiso del funcionamiento del sistema que se está instalando.

Para evitar la pérdida de datos y la sobreposición de los mismos, se separó por casos las funciones de ejecución del micro. Dentro del primer caso se configuró las entradas digitales, en el cual es necesario leer los puertos correspondientes a las entradas disponibles. En el segundo caso se leen las entradas analógicas, para ello es necesario hacer uso del conversor análogo digital interno del micro. Y en el tercer caso están las salidas digitales, donde se escriben niveles altos y bajos en los puertos de salida.

Se crearon dos SubVi de lectura y escritura únicamente para la transferencia de datos USB, desde estos SubVi se llaman o se instalan los drives de Windows (mpusbapi.dll) suministrados por Microchip. Estos drives adecuan la programación en Labview para el reconocimiento de datos provenientes del hardware externo (TAD).

Los SubVi desarrollados no se pueden poner en funcionamiento simultáneamente, ya que puede existir sobreposición y posteriormente perdida de datos, por ello se utilizó una estructura llamada "Sequence" que permite organizar la secuencia adecuada de lectura y escritura para los diferentes casos establecidos.

Se diseñó una interfaz para controlar e indicar los cambios ocurridos en la tarjeta (TAD), en dos versiones diferentes (Labview 8.2 y Labview 7.1), para permitir a los estudiantes de automatización de procesos una variedad de opciones para ejecutar el programa.

La interfaz se diseñó lo más didáctica posible, ya que cuenta con una imagen en 3D de la tarjeta (TAD), los indicadores y controles se asemejan a los especificados en el hardware, además los botones que simulan el encendido y apagado de los

leds y el cambio de los interruptores fueron modificados para que cumplieran la misma función que los vistos en la tarjeta física (hardware).

Los ejecutables contruidos para la interfaz también están en las dos versiones de Labview, esto permite su utilización desde cualquier PC que no tenga instalado Labview, pero para su correcto funcionamiento es necesario instalar los Run Time adecuados a la versión correspondiente. Esta función permite ejecutar la interfaz en casi cualquier PC sin importar que tipo de programas tenga instalados.

La tarjeta (TAD) cuenta con dos drives que son necesarios para su adaptación. El primero es instalado desde los SubVi de escritura y lectura. Y el segundo es aquel que reconoce la familia del micro desde el sistema operativo que se esté utilizando (para esta instalación Windows proporciona una reproducción automática). Sin la adecuada instalación de estos dos drives el PC no reconocerá la tarjeta ni mucho menos funcionara la aplicación. Para asegurar la detección de los drives se colocó dentro del hardware un led indicador.

El conversor implementado permite que el Sonic pueda ser conectado a cualquier PC que utilice puerto USB (computadores de última tecnología y portátiles) lo que permite un amplio rango de utilización del rediseño en conjunto con el Sonic

BIBLIOGRAFÍA

[1] Staveley technologies inc, *User manual Sonic 137*.

[2] Alexander Pinzón y Edwin A. Silva, *Diseño de la interfaz para el equipo de evaluación ultrasónica Sonic 137*, Bucaramanga, Universidad Industrial de Santander, 2002.

[3] Jose M. Lorente Gassó, *Transductores de Ultrasonido*, Industrial Informatics and Real-Time Systems Group, 2006, disponible:

<http://www.gii.upv.es/personal/gbenet/treballs%20cursos%20anteriors-TIM-IIN-INYP-AYPD/ultrasonidos/trabajo%20ultrasonidos.pdf>

[4] Olympus Corporation, 2009, *Medición de espesor*, disponible:

<http://www.olympus-ims.com/es/thickness-measurement/>

[5] Instituto Politécnico Nacional, centro de estudios científicos y tecnológicos, *Equipo de Ultrasonido, Transductores*, disponible:

http://www.cecyl7.ipn.mx/recursos/polilibros/inspeccion%20de%20soldadura/equipo_de_ultrasonido.html

[6] Sistemas Espectroscópicos, *Movimiento Ondulatorio*, disponible:

<http://rabfis15.uco.es/lvct/tutorial/21/Movimiento%20Ondulatorio.html>

[7] Universidad de Sevilla, *Universal Serial Bus*, Disponible:

http://www.forpas.us.es/aula/hardware/dia2_USB.pdf

[8] 2009, disponible: docente.uco.es/al993734/public_html/tinfo.doc

[9] Olympus Corporation, 2009, *Teoría y aplicación de la medición de precisión del espesor por ultrasonidos*, disponible:

<http://www.olympus-ims.com/es/applications-and-solutions/ndt-theory/theory/>

[10] Olympus Corporation, 2009, *An Introduction to Ultrasonic Flaw Detection*, disponible:

<http://www.olympus-ims.com/en/applications-and-solutions/introductory-ultrasonics/introduction-flaw-detection/>

[11] Conector DB9, disponible:

<http://es.kioskea.net/contents/elec/connecteur-prise-db9.php3>

[12] Ucontrol, *MAX232*, disponible:

<http://www.ucontrol.com.ar/wiki/index.php/MAX232>

[13] *Programador JDM*, disponible:

<http://ittz.blogspot.com/2008/04/programador-jdm.html>

[14] *The RS232 standard*, Massachusetts, 1993-2006, disponible:

http://www.camiresearch.com/Data_Com_Basics/RS232_standard.html

[15] Radio Determination Satellite Service, *Programador JDM*, disponible:

http://www.rdss.com.ar/datasheets/programador_jdm_plus_sc.pdf

[16] Micros y más micros, disponible:

<http://www.freewebs.com/glafebre/>

[17] Future Technology Devices International Ltd, *FT232BM*, disponible:

<http://www.ftdichip.com/Products/FT232BM.htm>

[18] System Consultation, 2005, *USB*, disponible:

<http://www.compsys1.com/support/docs/usbp3.pdf>

[19] LABVIEW 7.1 HELP, *Visa*.

ANEXO A. GUÍA DE CALIBRACIÓN SONIC 137

1. Encender el equipo y conectar el palpador.
2. Seleccionar dos valores conocidos de espesor, mayor y menor para la medición.
3. Configurar los valores iniciales;

MENÚ **PULSER**

DAMPING	50
MODE	PULSE-ECHO
MAX REP	4KHz

MENÚ **GAIN**

DISPLAY	FULLWAV
FREQ	HIPASS
REJECT	OFF

MENÚ **RANGE**

RANGE	Escoja un valor apropiado del rango
DELAY	0.020 seg.
VELOCIDAD	0.231 in/seg.

4. Colocar el palpador en el espesor más pequeño y obtenga la mejor presentación modificando DISPLAY en HALF+ o HALF- y modificando el REJECT hasta obtener ecos bien definidos, ubique el eco de tal manera que ocupe el 50% de amplitud en la pantalla modificando GAIN.
5. Ubicar los ecos en la posición correcta utilizando DELAY y comparando el valor de espesor mostrado en el display en el PC. (ver **Figura 70**).

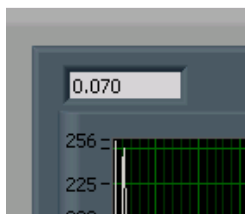


Figura 70. Visualización del valor de Espesor a partir de la Ubicación de los Ecos (Espesor)

6. Colocar el palpador en el espesor mayor y ubique el valor del espesor que corresponda modificando el comando VELOCIDAD (Ver **Figura 70**).

7. Configure los comandos;

HORN (GATE)	OFF
ANGLE BEAM (ANGLE)	ON

8. Active la compuerta GATE 2 "+", ubique el palpador en el espesor menor y ajuste el nivel (LEVEL) y la posición (POSN), de manera que corte el eco y la posición para que la compuerta no incluya el eco inicial (ver **Figura 71**)

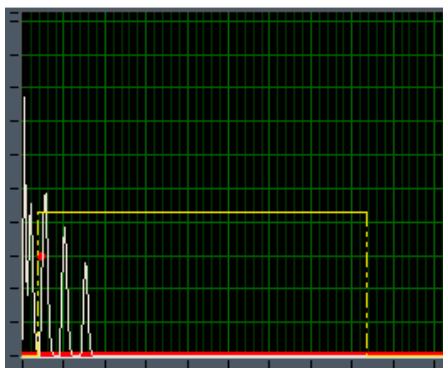


Figura 71. Ejemplo de la Compuerta en el Menor Espesor

NOTA: Si los ecos presentan variaciones en la amplitud se debe escoger el valor de LEVEL que siempre corte solo un eco tanto del menor como del mayor espesor.

9. Ajustar el ancho (WIDTH) es determinado por la ubicación del eco de espesor mayor, este debe ser cortado por la compuerta (ver **Fig. 72**).

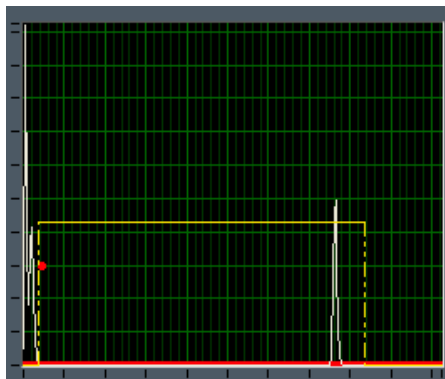


Figura 72. Ejemplo de la Compuerta en el Mayor Espesor

10. Ajustar el BLOCKING GATE (THICK) justo al lado derecho del eco inicial.

11. T-GAUGE (THICK) en “ON” encenderá el display de espesor del Sonic.



Figura 73. Indicador de Espesor

12. Ubicar el palpador en el mayor espesor, y configurar CAL MODE (THICK) en “HIGH” y presionar “ENTER”, configurar el valor de espesor conocido, y a continuación presione nuevamente “ENTER”.

NOTA: Verifique que el valor que muestra el indicador de espesor del Sonic equivale al valor mayor calibrado, de lo contrario es necesario repetir este punto.

13. Ubicar el palpador en el menor espesor, y configurar CAL MODE en “LOW” y presionar “ENTER”, configurar el valor de espesor, y a continuación presione de nuevo “ENTER”.

NOTA: Verifique que el valor que muestra el indicador de espesor del Sonic equivale al valor menor calibrado, de lo contrario es necesario repetir este punto.

14. Comprobar otras medidas de espesores, si la calibración no se logra se debe deshabilitar T-GAUGE y repetir los pasos a partir del punto 8.

15. Si desea una utilización óptima del Sonic, cierre el programa y utilice el equipo independiente del computador.

ANEXO B. GUÍA DETECCIÓN DE FALLAS

1. Realice la calibración del Sonic 137.
2. Ejecute nuevamente la interfaz del Sonic 137.



Figura 74. Icono de Ejecución del la Interfaz Sonic 137

3. Configure los siguientes valores

MENÚ **GAIN**

GAIN	35 dB
DISPLAY	FULLWAV
FREQ	HIPASS
REJECT	OFF

4. Escoja la opción “ir a detección de fallas”



Figura 75. Botón para Cambiar a Detección de Fallas

5. Para una mejor visualización de los ecos es posible variar el comando RANGE (RANGE) para aumentar la separación.



Figura 76. Perilla de Ajuste del Rango

6. Con las perillas de inicio y final configure la compuerta de manera que deje pasar el eco inicial y ubicada justo antes del siguiente eco. (ver **Figura 77**)



Figura 77. Perillas de Ajuste de la Compuerta

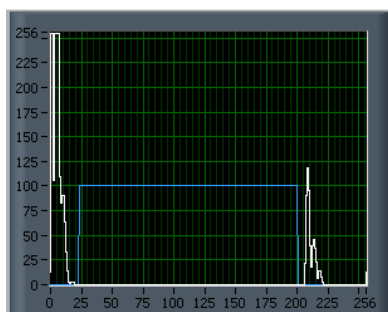


Figura 78. Visualización Compuerta

7. En la presencia de la falla la interfaz detecta su profundidad.

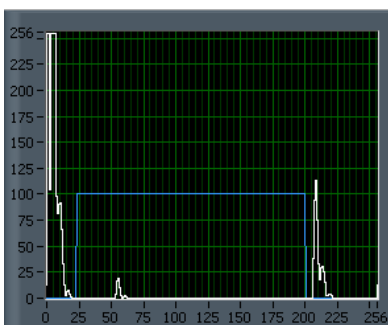


Figura 79. Visualización Compuerta y Falla Detectada



Figura 80. Indicador de la Profundidad de la Falla

ANEXO C. CIRCUITO INTEGRADO FT232BM UTILIZADO EN CONVERTOR SERIAL-USB

C1. Características del Hardware

- Transferencia USB =Asincrónica de Datos en Serie, en un solo chip
- Señales Completas de Protocolo
- UART I/F soporta datos de 7/8 bits, 1-2 bits de detención y paridad impar/par/guión/espacio/ninguna
- Velocidad de datos 300 = > 3M Baud (TTL)
- Velocidad de datos 300 = > 1M Baud (RS232)
- Buffer (Almacenamiento temporario) de recepción de 384 bytes / buffer de transmisión de 128 bytes para una alta generación de datos
- Tiempo de exclusión ajustable del buffer RX
- Hardware plenamente asistido o Protocolo X-encendido/X-apagado
- Soporte incorporado para caracteres de eventos y condición de ruptura de línea
- Soporte para la Suspensión/Reactivación de USB a través de las patas SLEEP# y RI#
- Soporte para dispositivos de alta potencia alimentados por el Bus USB a través de la pata PWREN#
- Conversor de nivel integrado en UART y señales de control para “comunicar” con la lógica de 5V y de 3,3V
- Regulador integrado de 3,3V para USB IO
- Circuito integrado de reinicialización de encendido
- Multiplicador integrado PLL de reloj de 6MHz-48MHz
- Modos de transferencia de datos isocrónico o masivo USB
- Funcionamiento con fuente de alimentación única de 4,35V a 5,25V

- Compatibilidad con controlador anfitrión UHCI/OHCI/EHCI
- Compatibilidad con USB 1.1 y USB 2.0
- USB VID, PID, Número Serie y cadenas de descripción de Producto en EEPROM externa
- EEPROM programable a través de USB
- Encapsulado compacto de 32 patas

C2. Drivers para Puerto Virtual COM (VCP)

La empresa provee drivers para manejar a este circuito integrado con los siguientes sistemas operativos:

Windows 98 y Windows 98 SE

Windows 2000/ME/XP

Windows CE 4.2

MAC OS-8 y OS-9

MAC OS-X

Linux 2.40 y superior

C3. D2XX (Drivers Directos USB + Interfaz DLL S/W)

Windows 98 y Windows 98 SE

Windows 2000/ME/XP

Windows CE 4.2

Linux 2.4 y superior

C4. Áreas de Aplicación

- Conversores USB <=> RS232
- Conversores USB <=> RS422/RS485
- Mejora de Periféricos RS232 hacia USB

- Interfaces y cables USB de transferencia de datos de teléfonos celulares e inalámbricos
- Interfaz de diseños basados en MCU (Unidad microcontroladora) a USB
- Transferencia de datos de video de banda angosta y de audio USB
- Transferencia de datos PDA <=>USB
- Lectoras inteligentes USB de tarjetas
- Interfaz USB - PC Set Top Box
- Módems USB
- Módems inalámbricos USB
- Instrumentación USB
- Lectoras USB de Código de Barras

C5. UART

La UART (**U**niversal **A**synchronous **R**eceiver-**T**ransmitter) realiza la conversión asincrónica paralela a serie y serie a paralela de 7/8 bits de los datos en la interfaz RS232 (RS422 y RS485). Las señales de control soportadas por la UART incluyen RTS, CTS, DSR, DTR, DCD y RI. La UART provee una señal de control de habilitación del transmisor (TXDEN) para ayudar en la interfaz con transceptores RS485. La UART soporta opciones de protocolo RTS/CTS, DSR/DTR y X-On/X-Off. El protocolo se maneja por hardware (donde se requiera) para asegurar tiempos de respuesta rápidos. La UART también soporta la posición BREAK de RS232 y las condiciones de detección.

C6. Interfaz de EEPROM

Aunque FT232BM trabajará sin la EEPROM opcional, se puede usar una EEPROM externa 93C46 (93C56 o 93C66) para adaptar a medida el USB VID, PID, Número de Serie, Cadenas de Descripción del Producto y el valor del

Descriptor de Potencia de la FT232BM para aplicaciones de OEM. Otros parámetros controlados por la EEPROM incluyen Despertar Remoto, Modo de Transferencia Isócrono, Descenso Suave de Tensión en el Apagado y modos del descriptor USB 2.0. La EEPROM debe ser de una configuración de 16 bits de ancho tal como MicroChip 93LC46B o equivalente capaz de una frecuencia de reloj de 1Mb/s con $V_{cc} = 4,35V$ a $5,25V$. La EEPROM se programa en la plaqueta a través de USB usando una utilidad disponible de la página de internet de FTDI (www.ftdichip.com). Si no se conecta ninguna EEPROM (o la EEPROM está en blanco), FT232BM usará su VID, Descripción de Producto PID y el valor del Descriptor de Potencia por defecto incorporados. En este caso, el dispositivo no tendrá un número de serie como parte del descriptor de USB.

Características:

- 1K de capacidad de memoria.
- 128 x 8 bits de tabla de organización.
- Bajo consumo gracias a la tecnología CMOS.⁵

Esta memoria es un componente opcional que se emplea en caso de querer un dispositivo privado donde se establece la descripción del producto para una posterior comercialización.

⁵ Conversor USB-RS232.(s.f.). Recuperado el 3 de marzo de 2009 de <http://www.learobotics.com/proyectos/usb-rs232/usb-rs232.htm>

ANEXO D. DIAGRAMA ESQUEMATICO Y PCB CONVERTOR RS232-USB

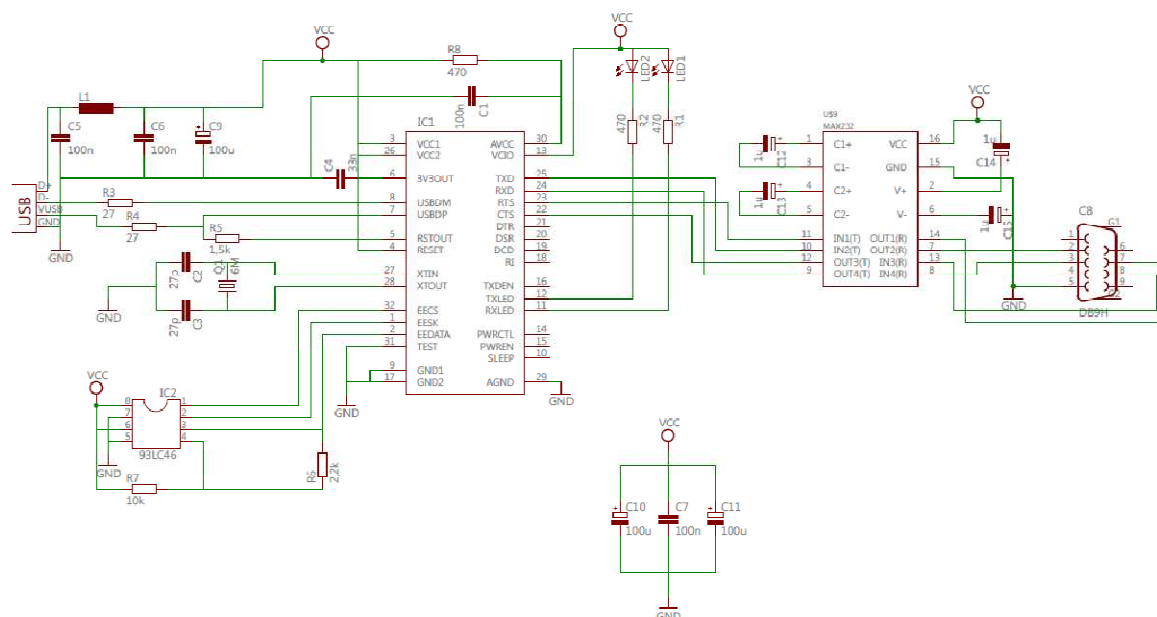


Figura 81 .Diagrama esquemático convertor [Autores]

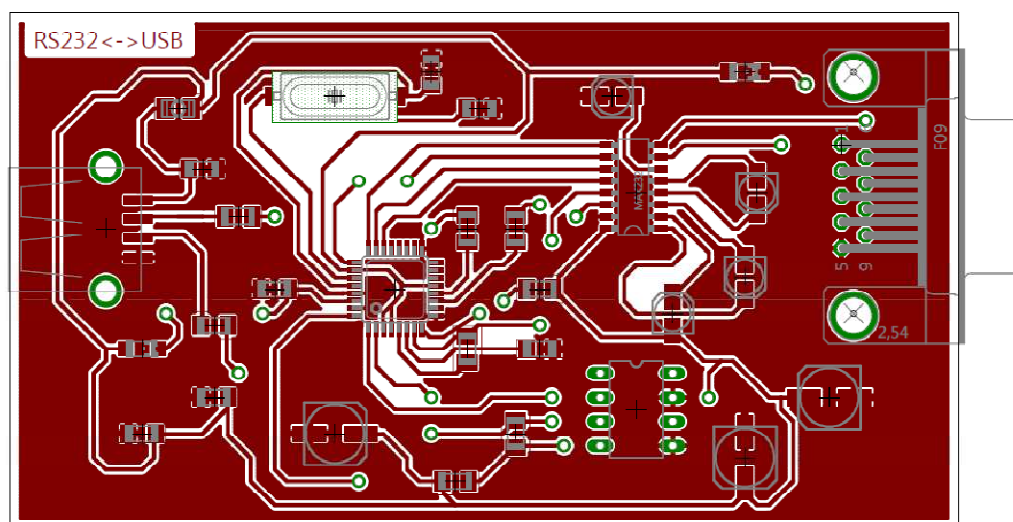


Figura 82. Primera Capa Layout del Convertor [Autores]

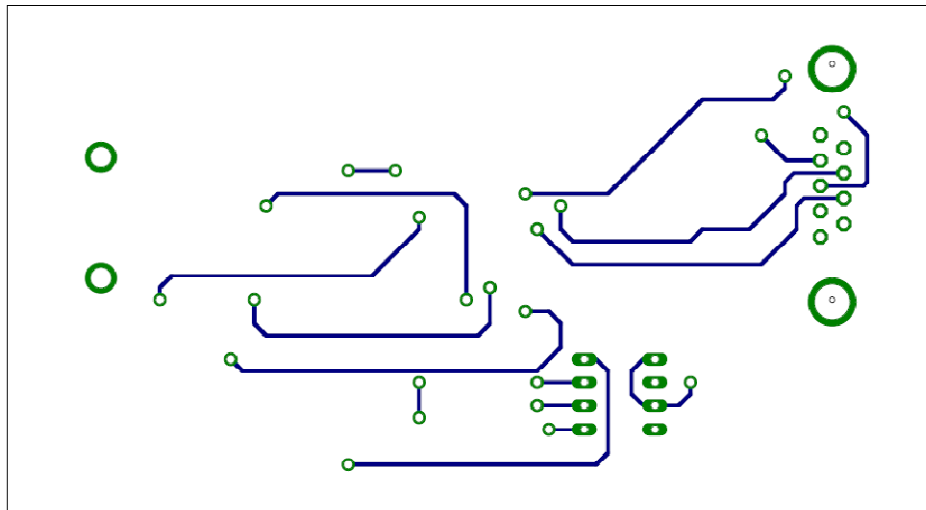


Figura 83. Segunda Capa Layout del Conversor [Autores]

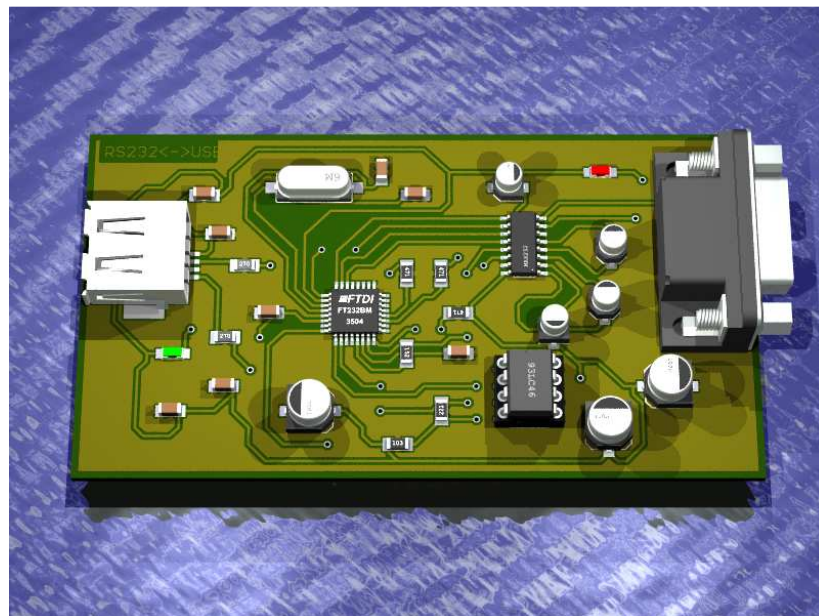


Figura 84. Simulación en 3D del conversor [Autores]

ANEXO E INSTALACIÓN DE LOS DRIVES PARA EL CONVERTOR

Al conectar el convertor al computador de inmediato saldrá un letreo que indicara el reconocimiento del nuevo hardware encontrado. Ver figura 85

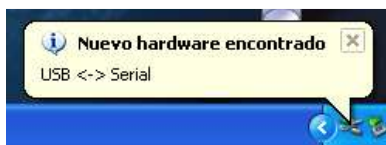


Figura 85. Mensaje de Windows de reconocimiento De hardware, comunicación USB [Autores]

Seguido de esto se abrirá un asistente donde debemos poner: Instalar desde una lista o ubicación específica, para buscar los drives del nuevo hardware. Ver figuras 86 y 87

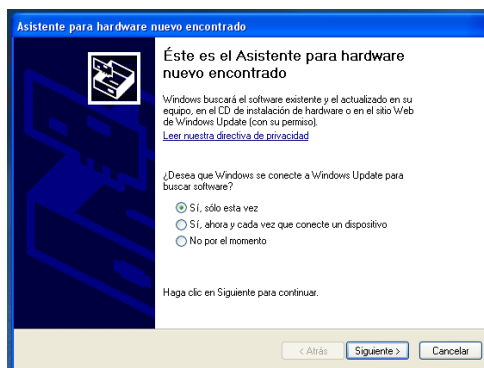


Figura 86. Asistente de instalación [Autores]

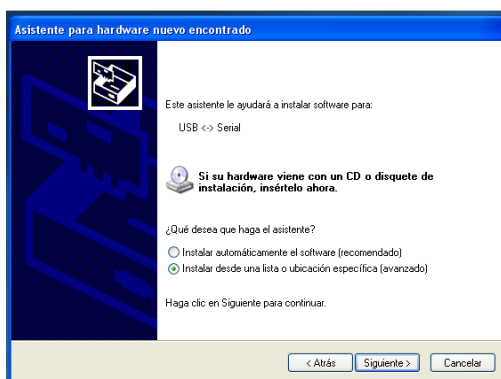


Figura 87. Asistente de instalación, Instalar desde una lista o ubicación específica [Autores]

En la opción examinar buscamos la carpeta donde se encuentran los drives (CD proporcionado por los autores), y luego presionamos siguientes. El asistente empezará a buscar los archivos necesarios e instalar los permisos.

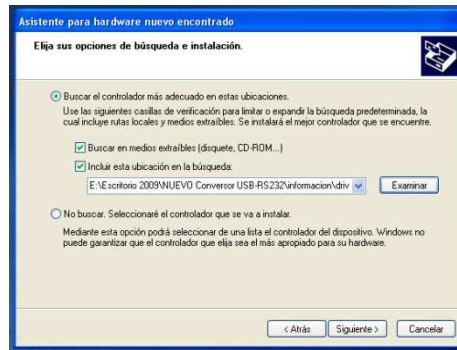


Figura 88. Asistente de instalación, opción de examinar la carpeta o dirección donde se encuentran los drivers [Autores]



Figura 89. Proceso de instalación de drivers [Autores]

Finalmente el asistente termina e indica la correcta instalación del hardware (USB serial port).



Figura 90. Asistente de instalación terminado [Autores]



Figura 91. Mensaje de verificación del nuevo Hardware instalado [Autores]

Después de esto nuevamente saldrá el asistente para instalar o crear la emulación del puerto COM, para ello se repiten los pasos anteriores para buscar los drives en la misma carpeta, una vez terminada la instalación saldrá un letrero que indica que el hardware puede ser usado.

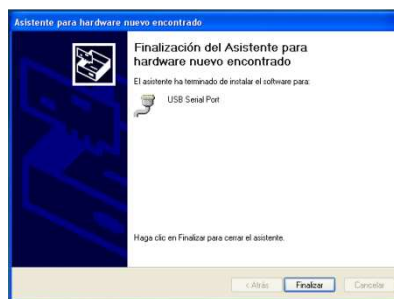


Figura 92. Asistente de instalación, emulación puerto COM [Autores]



Figura 93. Mensaje de Windows de reconocimiento De hardware, puerto COM virtual creado [Autores]

Para saber el número del puerto COM creado es necesario ir a MiPC, click derecho propiedades.



Figura 94. Opción para identificar el puerto COM creado [Autores]

Luego a hardware y le damos click a administrador de dispositivos.

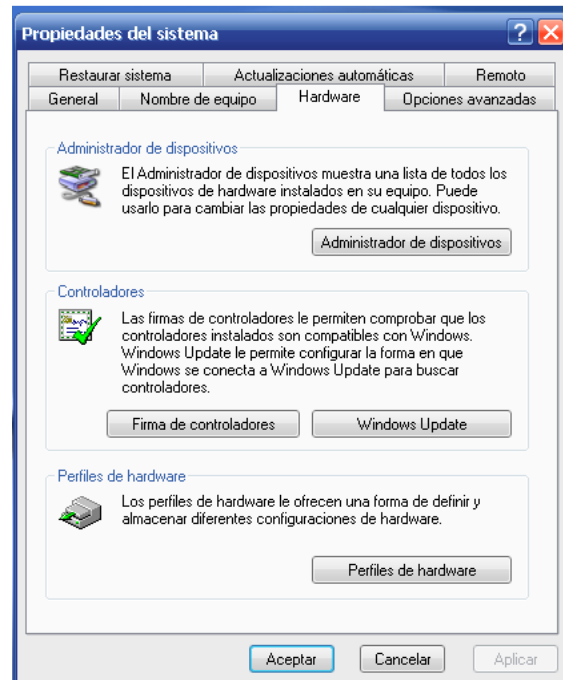


Figura 95. Asistente de propiedades del sistema [Autores]

Aquí saldrán todos los dispositivos presentes en el computador al igual que el puerto COM emulado anteriormente con el conversor.

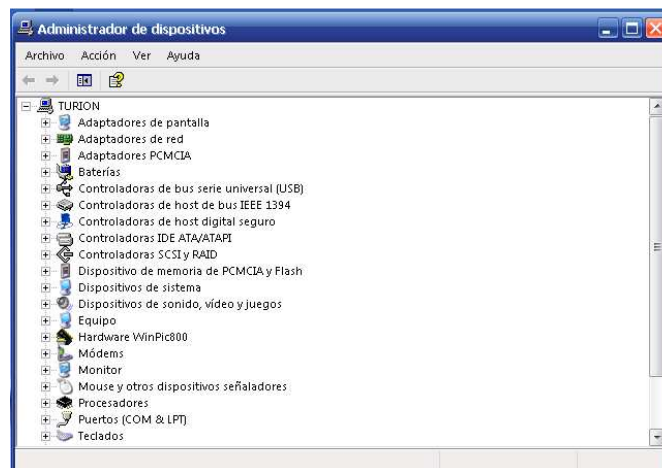


Figura 96. Administrador de dispositivos de Windows para identificar nuevos puertos [Autores]

Para verificar la instalación damos click en: Puertos (COM & LPT) y ahí aparecera nuestro puerto COM con su respectivo numero de asignacion (USB serial port (COM 3)).

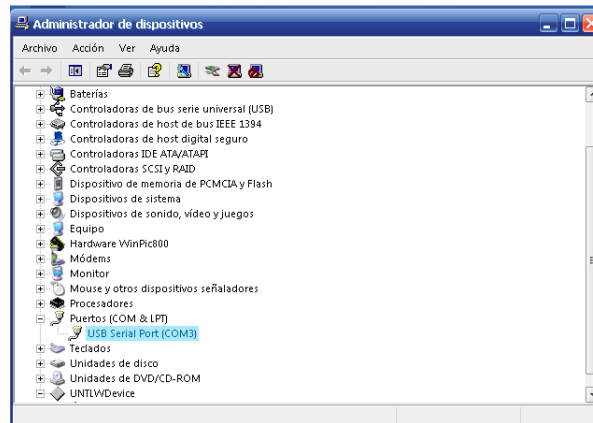


Figura 97. Reconocimiento de puerto [Autores]

En este caso aparecio COM 3 pero eso depende de la cantidad de dispositivos instalados y del numero de puertos USB en donde se instale el conversor.

ANEXO F. COMPILADOR PIC



PicC C Compiler

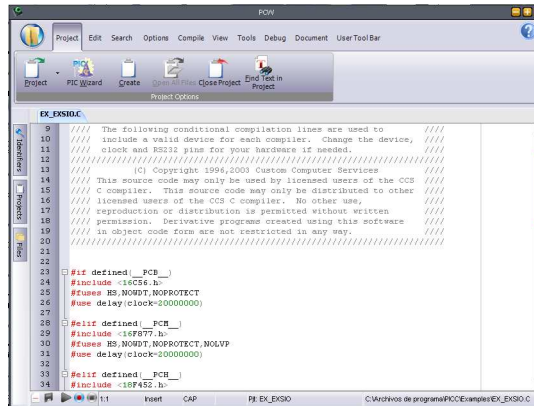


Figura 98. Entorno programa Pic C Compiler [Autores]

El PicC C Compiler fue desarrollado para cumplir con las especificaciones del lenguaje ANSI C. El compilador produce tres tipos de archivos. Archivos con extensión .hex que le permitirá grabar el programa ejecutable en el PIC por medio del uso de un programador. El archivo con extensión .asm contendrá un listado en assembler del programa compilado con la información del mapeo de memoria. Estos archivos son muy útiles para el debugging de los programas y para determinar la cantidad de pasos de programas (ciclos de ejecución) tiene la aplicación.

Beneficios

- Está basado en el ANSI C.
- Soporte completo de la familia de microcontroladores PIC de 14 bits.
- Salida Assembly .
- Industry standard Intel Hex 8 bit Merged format (INHX8M) .
- Soporta interrupciones.
- Tipos de datos 8 y 16 bit - int, char, long, pointers, unsigned, etc.

- Inserción de código ensamblador - asm();
- Todos los operadores aritméticos - incluyendo multiplicación, división, modulo y otros.
- Las variables y funciones no utilizadas son borradas automáticamente.
- Reutilización de RAM.
- Instrucciones simples en castellano.
- Dispositivos soportados: 16F84, 16C83, 16C554, 16C556, 16C558, 16C61, 16C62, 16C620, 16C621, 16C622, 16C63, 16C64, 16C641, 16C642, 16C65, 16C66, 16C661, 16C662, 16C67, 16C71, 16C710, 16C711, 16C715, 16C72, 16C73, 16C74, 16C76, 16C77, 16C9xx, 14C000
- **Nuevos** : 16CE623, 16CE624, 16CE625, 12C671, 12C672, 12C673, 12C674, 16F873, 16F874, 16F876, 16F877

Se optó por la utilización de este software debido a su facilidad de uso, sus diversos archivos que proporciona, por ser una versión gratuita y sobre todo por que permite compilar programación en C para los microcontroladores, en este caso PIC18F2550. Las características principales de la programación del PIC fueron: La asignación de los pines para el RS232 (tanto la entrada como la salida), la configuración USB y la descripción del dispositivo, y por último las especificaciones para la programación en LABVIEW.

La programación del PIC consiste en dos partes, un archivo .C que como su extensión lo dice esta en lenguaje C donde se establecen las acciones que el micro va a realizar, como el envío de datos, la recepción de datos, asignación de puertos, etc. Y un archivo .h que se encuentra en lenguaje de ensamblador donde se ajusta la configuración USB, las características del dispositivo y donde se establece el reconocimiento para la interacción con Windows. Este último tiene que ser incluido por el archivo .C a la hora de la compilación para asegurar el adecuado funcionamiento del dispositivo.

ANEXO G. PROGRAMACIÓN PIC 18F2550

PROGRAMACIÓN EN C

```
////////////////////////////////////
///      TARJETA DE ADQUISICION DE DATOS      ///
////////////////////////////////////
#include <18F2550.h>
#fuses HSPLL,NOWDT,NOPROTECT,NOLVP,NODEBUG,USBDIV,PLL5,CPUDIV1,VREGEN
#use delay(clock=48000000)
#use rs232(baud=9600, xmit=PIN_C6, rcv=PIN_C7)
////////////////////////////////////
#define USB_HID_DEVICE FALSE
#define USB_EP1_TX_ENABLE USB_ENABLE_BULK
#define USB_EP1_RX_ENABLE USB_ENABLE_BULK
#define USB_EP1_TX_SIZE 32
#define USB_EP1_RX_SIZE 32
////////////////////////////////////
#include <pic18_usb.h>
#include ".\include\configuracion.h"
#include <usb.c>
////////////////////////////////////
#define LED1 PIN_B7
#define LED2 PIN_B6
#define LED3 PIN_B5
#define LED4 PIN_B4
#define LED_ON output_high
#define LED_OFF output_low
#define modo recibe[0]
#define param1 recibe[1]
#define resultado envia[0]
#define resultado1 envia[1]

void main(void) {

    int8 recibe[2];          //declaramos variables
    int8 envia[2];

    usb_init();              //inicializamos el USB

    setup_adc_ports(AN0_TO_AN1);
    setup_adc(ADC_CLOCK_INTERNAL);
    set_adc_channel(0);

    usb_task();              //habilita periferico usb e interrupciones
    usb_wait_for_enumeration(); //esperamos hasta que el PicUSB sea configurado por el host

    LED_OFF(LED1);
```

```

while (TRUE)
{
  if(usb_enumerated())
  {
    if (usb_kbhit(1))
    {
      usb_get_packet(1, recibe, 3);

      switch (modo){

        case 0: //entrada digital (listo)

          if(input(PIN_A2)== 0 & input(PIN_A3)==0 & input(PIN_A4)==0 & input(PIN_A5)==0){ resultado=0;
            usb_put_packet(1, envia, 2, USB_DTS_TOGGLE);}

          if(input(PIN_A2)==1 & input(PIN_A3)==0 & input(PIN_A4)== 0 & input(PIN_A5)==0){ resultado=1;
            usb_put_packet(1, envia, 2, USB_DTS_TOGGLE);}

          if(input(PIN_A2)==0 & input(PIN_A3)==1 & input(PIN_A4)== 0 & input(PIN_A5)==0){ resultado=2;
            usb_put_packet(1, envia, 2, USB_DTS_TOGGLE);}

          if(input(PIN_A2)==1 & input(PIN_A3)==1 & input(PIN_A4)== 0 & input(PIN_A5)==0){ resultado=3;
            usb_put_packet(1, envia, 2, USB_DTS_TOGGLE);}

          if(input(PIN_A2)==0 & input(PIN_A3)==0 & input(PIN_A4)== 1 & input(PIN_A5)==0){ resultado=4;
            usb_put_packet(1, envia, 2, USB_DTS_TOGGLE);}

          if(input(PIN_A2)==1 & input(PIN_A3)==0 & input(PIN_A4)== 1 & input(PIN_A5)==0){ resultado=5;
            usb_put_packet(1, envia, 2, USB_DTS_TOGGLE);}

          if(input(PIN_A2)==0 & input(PIN_A3)==1 & input(PIN_A4)== 1 & input(PIN_A5)==0){ resultado=6;
            usb_put_packet(1, envia, 2, USB_DTS_TOGGLE);}

          if(input(PIN_A2)==1 & input(PIN_A3)==1 & input(PIN_A4)== 1 & input(PIN_A5)==0){ resultado=7;
            usb_put_packet(1, envia, 2, USB_DTS_TOGGLE);}

          if(input(PIN_A2)==0 & input(PIN_A3)==0 & input(PIN_A4)== 0 & input(PIN_A5)==1){ resultado=8;
            usb_put_packet(1, envia, 2, USB_DTS_TOGGLE);}

          if(input(PIN_A2)==1 & input(PIN_A3)==0 & input(PIN_A4)== 0 & input(PIN_A5)==1){ resultado=9;
            usb_put_packet(1, envia, 2, USB_DTS_TOGGLE);}

          if(input(PIN_A2)==0 & input(PIN_A3)==1 & input(PIN_A4)== 0 & input(PIN_A5)==1){ resultado=10;
            usb_put_packet(1, envia, 2, USB_DTS_TOGGLE);}

          if(input(PIN_A2)==1 & input(PIN_A3)==1 & input(PIN_A4)== 0 & input(PIN_A5)==1){ resultado=11;
            usb_put_packet(1, envia, 2, USB_DTS_TOGGLE);}

          if(input(PIN_A2)==0 & input(PIN_A3)==0 & input(PIN_A4)== 1 & input(PIN_A5)==1){ resultado=12;
            usb_put_packet(1, envia, 2, USB_DTS_TOGGLE);}

```

```
if(input(PIN_A2)==1 & input(PIN_A3)==0 & input(PIN_A4)== 1 & input(PIN_A5)==1){ resultado=13;
usb_put_packet(1, envia, 2, USB_DTS_TOGGLE);}
```

```
if(input(PIN_A2)==0 & input(PIN_A3)==1 & input(PIN_A4)== 1 & input(PIN_A5)==1){ resultado=14;
usb_put_packet(1, envia, 2, USB_DTS_TOGGLE);}
```

```
if(input(PIN_A2)==1 & input(PIN_A3)==1 & input(PIN_A4)== 1 & input(PIN_A5)==1){ resultado=15;
usb_put_packet(1, envia, 2, USB_DTS_TOGGLE);}
```

case 1: // salidad analoga (listo)

```
resultado1 = read_adc();
resultado =0;
usb_put_packet(1, envia, 2, USB_DTS_TOGGLE);
```

case 2: // salida digital

```
if (param1 == 0) {LED_OFF(LED1); LED_OFF(LED2);LED_OFF(LED3);LED_OFF(LED4);}
if (param1 == 1) {LED_ON(LED1); LED_OFF(LED2);LED_OFF(LED3);LED_OFF(LED4);}
if (param1 == 2) {LED_OFF(LED1); LED_ON(LED2);LED_OFF(LED3);LED_OFF(LED4);}
if (param1 == 3) {LED_ON(LED1); LED_ON(LED2);LED_OFF(LED3);LED_OFF(LED4);}
if (param1 == 4) {LED_OFF(LED1); LED_OFF(LED2);LED_ON(LED3);LED_OFF(LED4);}
if (param1 == 5) {LED_ON(LED1); LED_OFF(LED2);LED_ON(LED3);LED_OFF(LED4);}
if (param1 == 6) {LED_OFF(LED1); LED_ON(LED2);LED_ON(LED3);LED_OFF(LED4);}
if (param1 == 7) {LED_ON(LED1); LED_ON(LED2);LED_ON(LED3);LED_OFF(LED4);}
if (param1 == 8) {LED_OFF(LED1); LED_OFF(LED2);LED_OFF(LED3);LED_ON(LED4);}
if (param1 == 9) {LED_ON(LED1); LED_OFF(LED2);LED_OFF(LED3);LED_ON(LED4);}
if (param1 == 10) {LED_OFF(LED1); LED_ON(LED2);LED_OFF(LED3);LED_ON(LED4);}
if (param1 == 11) {LED_ON(LED1); LED_ON(LED2);LED_OFF(LED3);LED_ON(LED4);}
if (param1 == 12) {LED_OFF(LED1); LED_OFF(LED2);LED_ON(LED3);LED_ON(LED4);}
if (param1 == 13) {LED_ON(LED1); LED_OFF(LED2);LED_ON(LED3);LED_ON(LED4);}
if (param1 == 14) {LED_OFF(LED1); LED_ON(LED2);LED_ON(LED3);LED_ON(LED4);}
if (param1 == 15) {LED_ON(LED1); LED_ON(LED2);LED_ON(LED3);LED_ON(LED4);}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

PROGRAMACIÓN EN ASEMBLER

```
////////////////////////////////////  
///          CONFIGURACION.h          ///  
///          ///  
////////////////////////////////////
```

```
#ifndef __USB_DESCRIPTOR__  
#define __USB_DESCRIPTOR__
```

```
#include <usb.h>
```

```
////////////////////////////////////  
///  
////////////////////////////////////
```

```
#define USB_TOTAL_CONFIG_LEN 32
```

```
char const USB_CONFIG_DESC[] = {
```

```
    USB_DESC_CONFIG_LEN,  
    USB_DESC_CONFIG_TYPE,  
    USB_TOTAL_CONFIG_LEN,0,  
    1,  
    0x01,  
    0x00,  
    0xC0,  
    0x32,  
    USB_DESC_INTERFACE_LEN,  
    USB_DESC_INTERFACE_TYPE,  
    0x00,  
    0x00,  
    2,  
    0xFF,  
    0xFF,  
    0xFF,  
    0x00,
```

```
//endpoint descriptor
```

```
    USB_DESC_ENDPOINT_LEN,  
    USB_DESC_ENDPOINT_TYPE,  
    0x81,  
    0x02,  
    USB_EP1_TX_SIZE,0x00,  
    0x01,
```

```
//endpoint descriptor
```

```
    USB_DESC_ENDPOINT_LEN,  
    USB_DESC_ENDPOINT_TYPE,  
    0x01,  
    0x02,
```

```

        USB_EP1_RX_SIZE,0x00,
        0x01,

};

//***** BEGIN CONFIG DESCRIPTOR LOOKUP TABLES *****
#define USB_NUM_HID_INTERFACES 0
#define USB_MAX_NUM_INTERFACES 1

const char USB_NUM_INTERFACES[USB_NUM_CONFIGURATIONS]={1};

#if (sizeof(USB_CONFIG_DESC) != USB_TOTAL_CONFIG_LEN)
    #error USB_TOTAL_CONFIG_LEN not defined correctly
#endif

////////////////////////////////////
///
///  start device descriptors
///
////////////////////////////////////

//device descriptor
char const USB_DEVICE_DESC[]={
    USB_DESC_DEVICE_LEN,
    0x01,
    0x10,0x01,
    0x00,
    0x00,
    0x00,
    USB_MAX_EP0_PACKET_LENGTH,
    0xD8,0x04,    //vendor id (0x04D8 is Microchip)
    0x0B,0x00,    //product id
    0x01,0x00,
    0x01,
    0x02,        //index of string descriptor of the product
    0x00,        //index of string descriptor of serial number
    USB_NUM_CONFIGURATIONS //number of possible configurations
};

////////////////////////////////////
///
///  start string descriptors
///
////////////////////////////////////

const char USB_STRING_DESC_OFFSET[]={0,4,12};

#define USB_STRING_DESC_COUNT sizeof(USB_STRING_DESC_OFFSET)

char const USB_STRING_DESC[]={

```

```
//string 0
4,
USB_DESC_STRING_TYPE,
0x09,0x04,
//string 1 --> la compañía del producto ???
8,
USB_DESC_STRING_TYPE,
'R',0,
'R',0,
'2',0,
//string 2 --> nombre del dispositivo
22,
USB_DESC_STRING_TYPE,
'S',0,
'O',0,
'F',0,
'T',0,
'I',0,
'N',0,
'G',0,
'U',0,
'I',0,
'S',0,

};

#endif
```

ANEXO H. ESQUEMATICO LAYOUT E IMPRESO DEL PROGRAMADOR

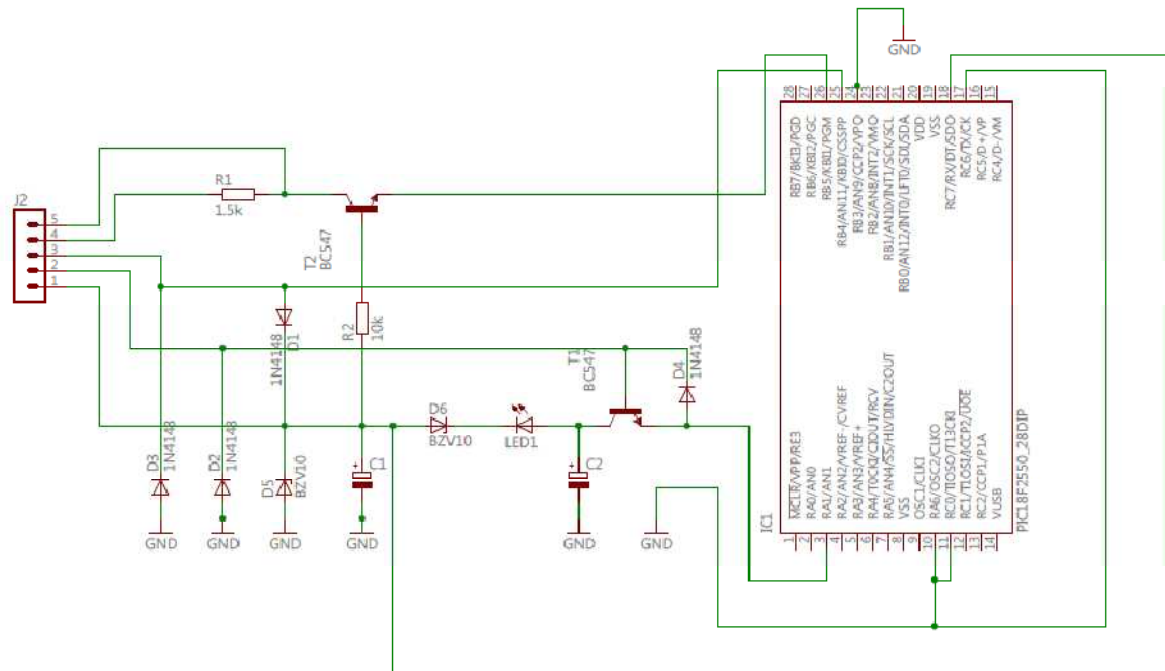


Figura 99. Esquemático en Eagle del programador [Autores]

- Componentes del programador

R1 = 1.5 K

R2 = 10 K

D1 a D4 = Diodos 1N4148

DZ1 = Zener 5.1 volt 1/2 watts

DZ2 = Zener 6.2 volt 1/2 watts

Q1 - Q2 = BC547

L1 = Led rojo

EL1 = Electrolítico 47 uf/16 volt

EL2 = Electrolítico 100 uf/16 volt

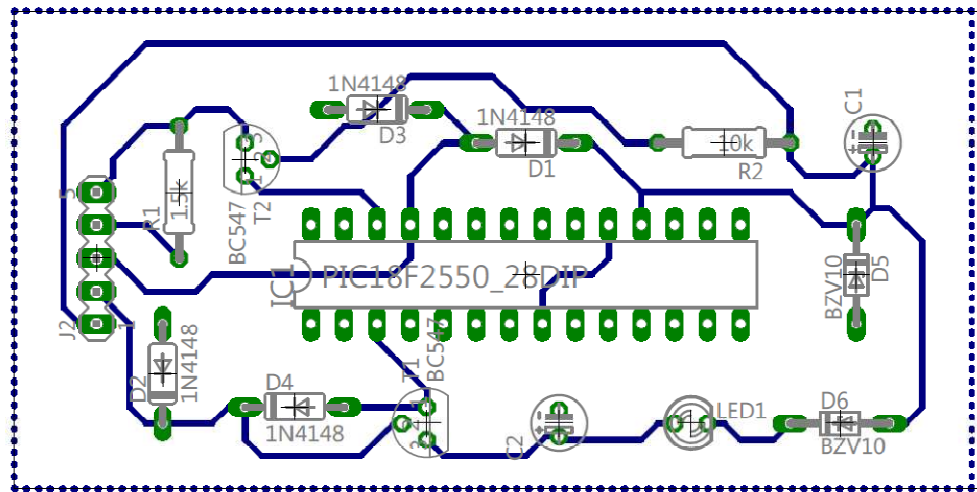


Figura 100. Layout del programador [Autores]

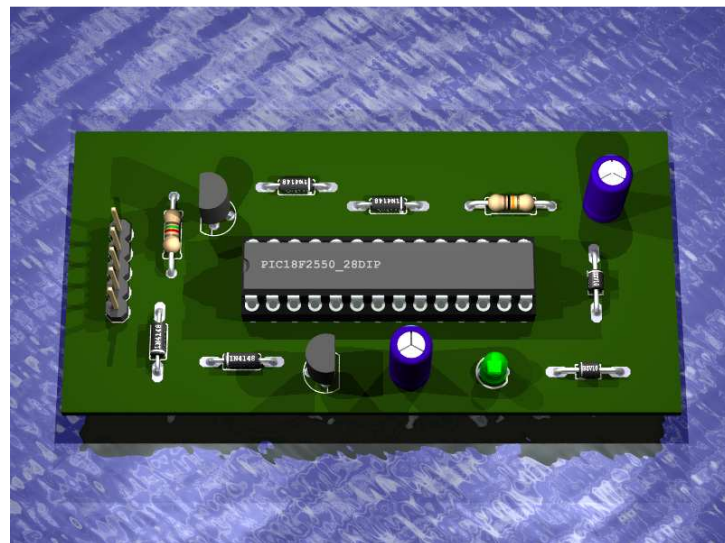


Figura 101. Simulación en 3D del programador [Autores]

ANEXO I. CONEXIÓN DE LA TARJETA DE ADQUISICIÓN DE DATOS E INSTALACIÓN DEL RUN TIME

Cuando la TAD_EP es conectada mediante el puerto USB el computador reconoce un dispositivo y de inmediato pide los drivers para poder dejar funcionar el nuevo hardware encontrado.



Figura 102. Reconocimiento del hardware [Autores]

Los drivers son los proporcionados por la página de MICROCHIP los cuales permiten reconocer cualquier dispositivo que tenga la interfaz USB. (Tales como 18F2550 y 18F4550 entre otros).

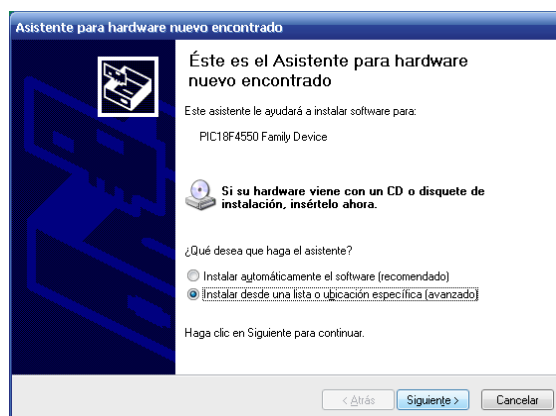


Figura 103. Asistente para hardware [Autores]

Seleccionamos la opción de instalar de una lista o ubicación específica, y buscamos la carpeta adjunta donde se encuentran los drivers (los drivers se

encuentran en el CD que viene con la tarjeta de adquisición), seleccionamos PicUSB_driver.



Figura 104. Carpeta que contiene los drives [Autores]

De inmediato con base en estos archivos el PC inicia su instalación para el correcto funcionamiento de la tarjeta de adquisición.



Figura 105. Deteccion de drives e instalacion [Autores]

En algunas oportunidades saldrá una ventana que preguntará ¿Desea continuar?, solo se le da continuar para que Windows permita instalar todo sin ningún problema, Windows realiza esta pregunta por seguridad del funcionamiento del

sistema, debido a que el software no es de total conocimiento para el sistema operativo.

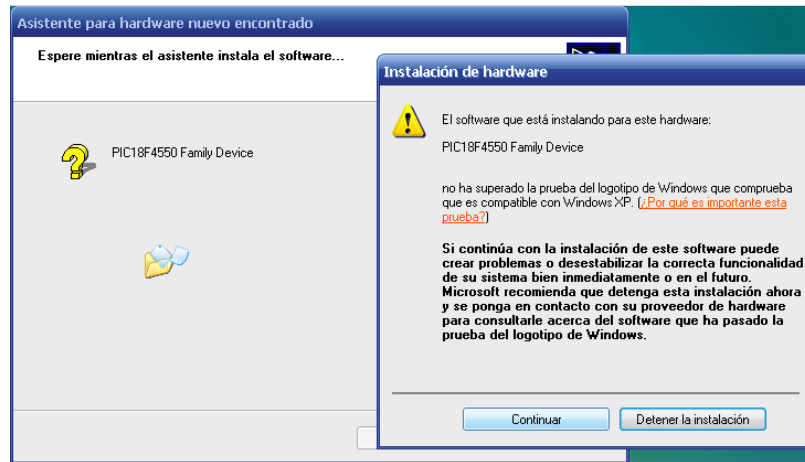


Figura 106. Ventana de seguridad de windows [Autores]

Una vez terminada la instalación de los drivers aparecerá una ventana que indica que el dispositivo ya puede ser usado desde el PC.



Figura 107. Mensaje de correcta instalación [Autores]

11. Instalación del RunTime

Este software es una aplicación de National Instruments que sirve para ejecutar proyectos realizados en Labview en computadores en donde no esté instalado Labview, es decir es un componente extra para posibilitar la utilización de archivos de Labview ejecutables.

Este componente es necesario para que funcione el ejecutable que se creó de la interfaz en Labview, contiene librerías, accesos y componentes para la adecuada ejecución de la interfaz. Para instalarlo es necesario buscar en el CD adjunto la carpeta llamada **Run time para Labview 8.2 WINDOWS XP, 2000** o **Run time para Labview 7.1 WINDOWS XP, 2000** dependiendo de la versión que en la que quiera ejecutar el programa.



En muchos computadores es necesario instalar los dos software que se encuentran en esa carpeta.

Una vez se de doble clic en **LabVIEW82RuntimeEngine** aparecerá una venta indicando si quiero extraer los componentes necesarios, en esta opción es necesario darle clic en aceptar para continuar con la ejecución del mismo.

De igual forma con el run time para Labview 7.1

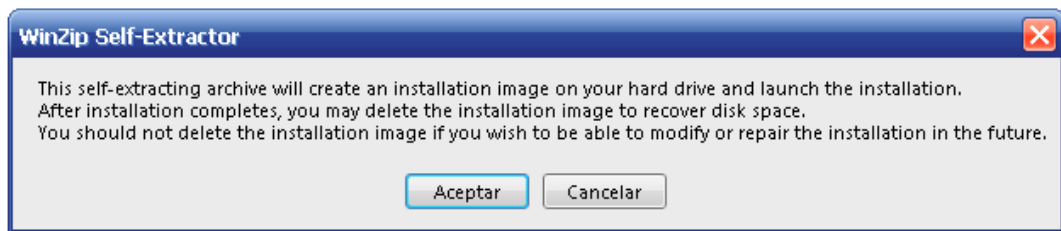


Figura 108. Extracion de componentes del run time [Autores]

A continuación escogemos la opción Unzip para que extraiga los archivos necesarios.

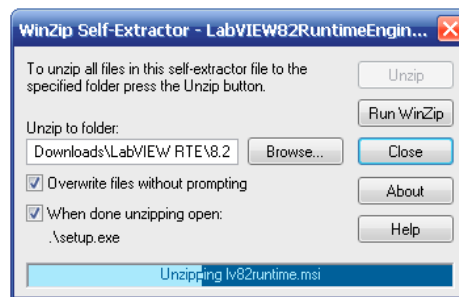


Figura 109. Inicio de extracion [Autores]

Seguido de eso le damos aceptar y continuamos con la instalación de la aplicación dándole la opción de siguiente (next).

La aplicación se empieza a instalar como un componente fundamental de Labview 8.2

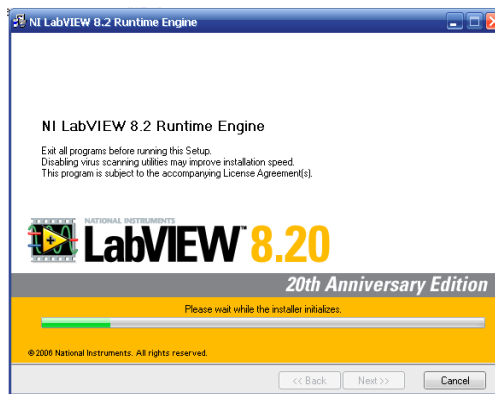


Figura 110. Instalacion [Autores]

En esta ventana le da la opción de guardar el componente en cualquier dirección que crea conveniente, pero por recomendación es mejor dejarlo en la dirección que aparezca por defecto.

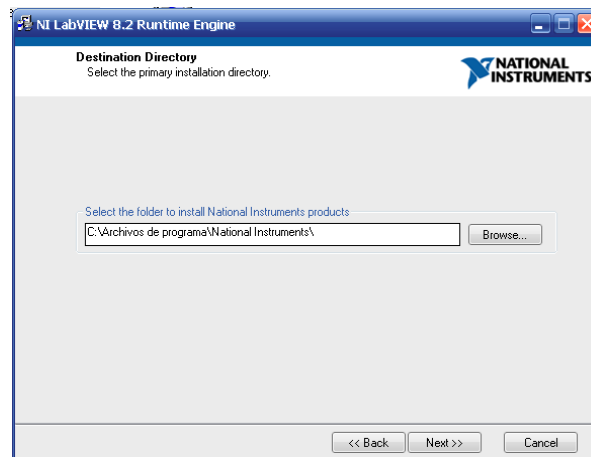


Figura 111. Asistente para guardar programa instalado [Autores]

Es esta ventana aparece la opción de seleccionar los componentes que se instalaran, es recomendable escoger los se ofrecen por defecto, pues estos permiten una buena ejecución de la aplicación.

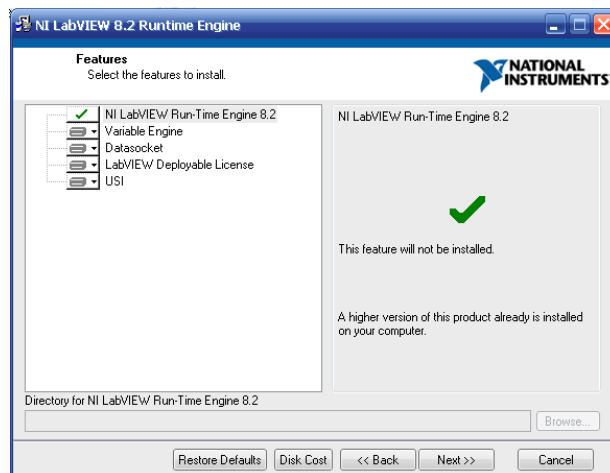


Figura 112. Componentes instalados [Autores]

Cuando termina la instalación de dicho componente es necesario reiniciar el sistema para que se actualicen las librerías y demás funciones adicionales.

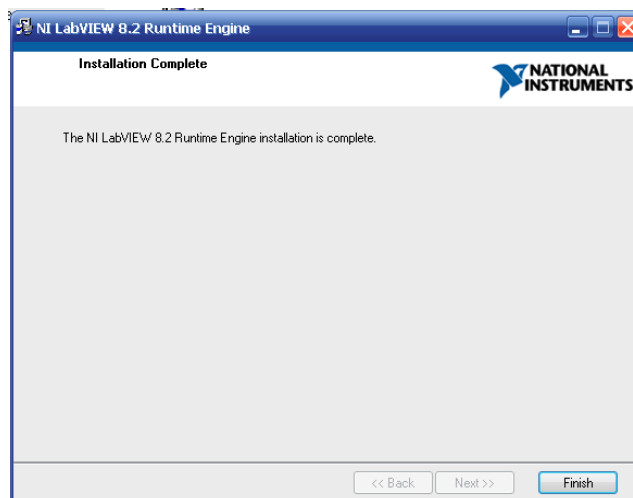


Figura 113. Fin de la instalacion [Autores]

ANEXO J. ESQUEMÁTICO PCB Y HARDWARE DE LA TARJETA DE ADQUISICIÓN DE DATOS

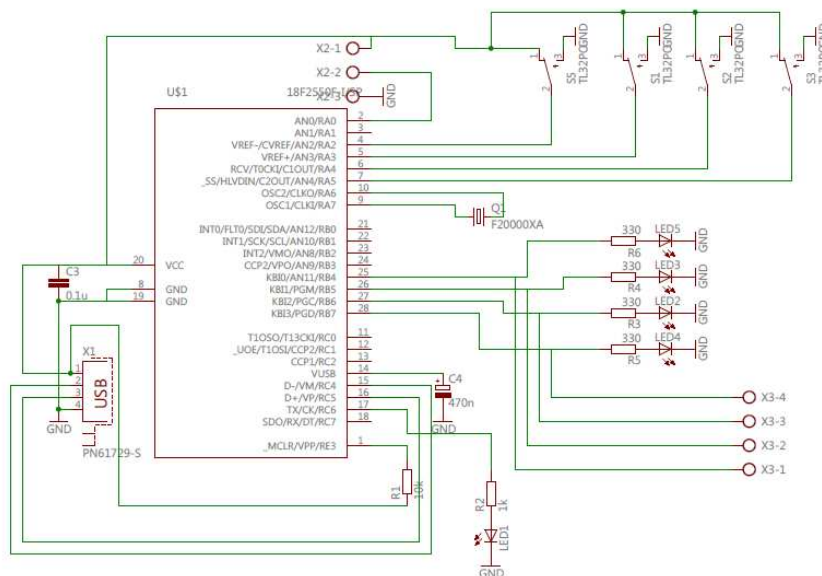


Figura 114. Diagrama esquemático TAD [Autores]

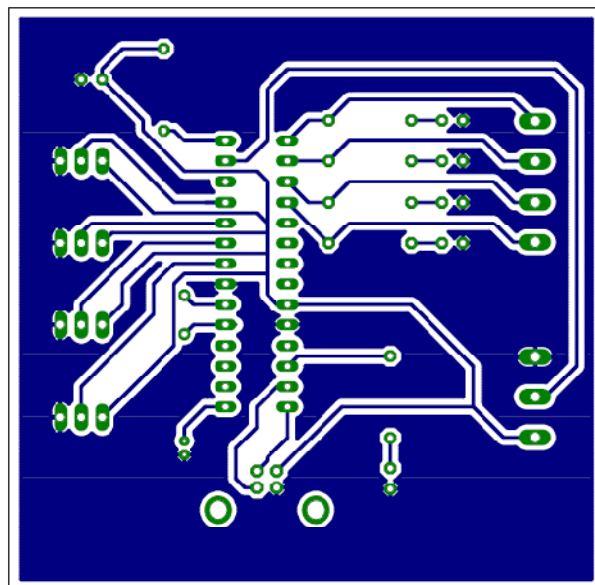


Figura 115. Layout TAD [Autores]



Figura 116. Simulación en 3D de LA TAD [Autores]



Figura 117. Hardware TAD [Autores]



Figura 118. Hardware TAD con su Estuche [autores]



Figura 119. Hardware TAD Y CABLE [Autores]