



UNIVERSIDAD TECNOLÓGICA DE LA MIXTECA

Tarjeta de Adquisición de Datos con interfaz USB, empleando al Microcontrolador AT90USB646 de ATMEL

Tesis para obtener el título de:

Ingeniero en Electrónica

Presenta:

Magdiel Pascual García Juárez

Director de Tesis:

M.C. Felipe Santiago Espinosa

Huajuapán de León, Oaxaca; Mayo de 2013

Dedicatoria

Con especial afecto y cariño a:

Mi madre

Juliana

Mis hermanos

Ignacio, Dolores, Griselda, Dolly y Francisco

Mi hija

Carolina

Mi esposa

Lorena

Agradecimientos

A mi director de tesis el M.C. Felipe Santiago Espinosa por su amistad, el conocimiento y tiempo invertido en la realización de esta tesis.

A los profesores Fermín Hugo Ramírez Leyva, Heriberto I. Hernández Martínez, Hugo Suarez Onofre cuyas observaciones y sugerencias contribuyeron al mejoramiento de esta tesis.

A mi madre Juliana Juárez Meza porque gracias a sus consejos y palabras de aliento me fue posible concluir este trabajo de tesis.

A mi esposa Lorena Barragán por su apoyo y comprensión durante todo el tiempo que duró el desarrollo del presente trabajo de tesis.

A mi padre José por su apoyo económico.

A la Universidad Tecnológica de la Mixteca por permitirme desarrollar el presente trabajo de tesis dentro de sus instalaciones.

Índice General

Dedicatoria	iii	
Agradecimientos	v	
Índice General	vii	
Índice de Figuras	xi	
Índice de Tablas	xvii	
Resumen	xix	
Capítulo 1	Introducción	1
1.1	Planteamiento del Problema	5
1.2	Justificación	5
1.3	Objetivos	6
1.3.1	Objetivo General	6
1.3.2	Objetivos particulares	6
1.4	Organización del documento	7
Capítulo 2	Marco teórico	9
2.1	Microcontroladores	9
2.1.1	Organización de los Microcontroladores	9
2.1.1.1	La Unidad Central de Procesamiento	10
2.1.1.2	Sistema de memoria	12
2.1.1.3	Oscilador	12
2.1.1.4	Temporizador/Contador	12
2.1.1.5	Perro Guardián	12

2.1.1.6	Puerto serie	12
2.1.1.7	Entradas/Salidas Digitales	13
2.1.1.8	Entradas/Salidas Analógicas	13
2.1.2	Clasificación de los Microcontroladores.....	13
2.1.3	Organización de los Microcontroladores AVR de Atmel.	14
2.1.3.1	El Núcleo AVR	14
2.2	Bus Universal Serial.....	16
2.2.1	Aspectos generales del USB	16
2.2.2	Historia y evolución del USB	16
2.2.3	Áreas de aplicación	17
2.2.4	Protocolo de comunicación.....	17
2.2.4.1	Especificaciones Eléctricas	21
2.2.4.2	Especificaciones Mecánicas	21
2.2.4.3	Especificaciones Funcionales.....	22
2.2.4.4	Estados de los Dispositivos USB	23
2.2.4.5	Descriptores USB.....	25
2.3	Entorno de desarrollo LabVIEW	27
2.3.1	Llamadas a las funciones de una biblioteca	28
2.3.2	Detalles de un CIN.....	29
2.3.3	Biblioteca de Enlace Dinámico para un HID de Atmel	29
Capítulo 3	Definición del sistema	33
3.1	Metodología de desarrollo.....	33
3.2	Especificación del producto	35
3.3	Definición de Hardware y Software.....	35
3.3.1	Definición de Hardware	36
3.3.1.1	El Microcontrolador AT90USB646.....	37
3.3.1.1.1	Controlador USB	39
3.3.1.1.2	Puertos de Entrada/Salida digitales	41
3.3.1.1.3	Convertidor Analógico Digital	42
3.3.1.1.4	Temporizadores	44
3.3.1.1.5	Modulación por Ancho de Pulso	45
3.3.1.1.6	Comunicación I2C.....	46
3.3.1.1.7	Comunicación serial por SPI	47
3.3.1.2	Acondicionamiento de Señal.....	48
3.3.1.3	Convertido Digital-Analógico.....	50
3.3.2	Definición del Software	50
3.3.2.1	Software del Microcontrolador	50

3.3.2.2	Interfaz Gráfica de Usuario.....	51
Capítulo 4	Diseño del sistema	53
4.1	Diseño del Hardware.....	53
4.2	Diseño del Software.....	56
4.2.1	Interfaz Gráfica de Usuario en la PC	56
4.2.2	Software del Microcontrolador.....	64
Capítulo 5	Integración y pruebas	69
5.1	Prueba comunicación micro-PC y viceversa	70
5.2	Evaluación de los Módulos.....	71
5.2.1	Prueba del módulo ADC	71
5.2.2	Prueba del módulo PWM.....	72
5.2.3	Prueba del módulo Contador de Eventos.....	73
5.2.4	Prueba del módulo Entradas/Salidas Digitales	74
5.2.5	Prueba del módulo Convertidor Digital Analógico	75
5.2.6	Prueba del módulo I2C	76
5.2.7	Medición del tiempo de conversión del DAC y ADC	78
5.2.8	Prueba de dos módulos en un mismo VI	79
Capítulo 6	Conclusiones y trabajos futuros	81
6.1	Trabajos futuros.	82
Bibliografía	85
Apéndice A	Tramas de configuración y monitoreo de los recursos ...	A-1
Apéndice B	Manual de Usuario	B-1
Apéndice C	Circuitos y componentes empleados.....	C-1

Índice de Figuras

Figura 1.1 Diagrama de un sistema de adquisición de datos.	2
Figura 1.2 Ejemplos de Tarjetas de Adquisición de Datos [18,19].	4
Figura 2.1. Organización de un microcontrolador [21].	10
Figura 2.2. Elementos comunes en una CPU [21].	11
Figura 2.3 Diagrama a bloques del núcleo AVR [10].	15
Figura 2.4 Modelo de la capa de comunicación [7].	18
Figura 2.5 Formato del paquete PID.	19
Figura 2.6 Formato del campo de dirección.	19
Figura 2.7 Formato para el campo Endpoint.	19
Figura 2.8 Cable USB [27].	21
Figura 2.9 Conectores USB [7].	22
Figura 2.10 Topología estándar USB [3].	23
Figura 2.11 Estados de la unidad USB [7].	24
Figura 2.12 Descriptor USB [3].	26
Figura 2.13 Módulo para llamadas a funciones de una biblioteca.	29
Figura 2.14 Configuración de parámetros para las llamadas a funciones de biblioteca.	29
Figura 2.15 Funciones DLL [25].	30

Figura 3.1 Diagrama de las fases para el diseño de un sistema embebido [2].	34
Figura 3.2 Diagrama a bloques de la especificación del sistema.	36
Figura 3.3 Aspecto externo del microcontrolador AT90USB646 [10].	38
Figura 3.4 Diagrama a bloques del Controlador USB [10].	40
Figura 3.5 Alimentación del BUS para Unidades USB [10].	41
Figura 3.6 Esquemático para un pin de Entrada/Salida [10].	42
Figura 3.7 Pre-escalador del ADC [10].	43
Figura 3.8 Hardware para el voltaje de referencia [10].	43
Figura 3.9 Definición del ciclo de trabajo.	46
Figura 3.10 Bus de interconexión TWI [10].	47
Figura 3.11 Conexión entre 2 dispositivos vía SPI [21].	48
Figura 3.12 Circuito integrado MCP4922 [12].	50
Figura 3.13 Diagrama de la arquitectura del software.	51
Figura 4.1 Conexión de los puertos Digitales y salidas PWM.	54
Figura 4.2 Circuito acondicionador de señales para una entrada de 0 a 10V.	55
Figura 4.3 Circuito acondicionador de señales de entrada de -10 a 10V.	55
Figura 4.4 Diseño del Convertidor Analógico Digital.	56
Figura 4.5 Módulo para establecer la comunicación.	57
Figura 4.6 Módulo para probar la comunicación.	58
Figura 4.7 Módulo para la configuración de E/S digitales.	59
Figura 4.8 Módulo para la configuración de las salidas PWM.	60
Figura 4.9 Módulo para el manejo del ADC.	61
Figura 4.10 Módulo para la configuración del DAC.	62
Figura 4.11 Módulo para la configuración del contador de Eventos.	63
Figura 4.12 Módulo para la configuración de la comunicación por I2C.	64
Figura 4.13 Arquitectura del software del microcontrolador [19].	65
Figura 4.14 Diagrama de flujo del software para el manejo de los recursos de la TAD.	67

Figura 5.1 Integración del Hardware y Software.....	69
Figura 5.2 Interfaz de usuario para la configuración de la conexión.....	70
Figura 5.3 Interfaz del módulo para probar la comunicación.....	70
Figura 5.4 Pruebas de la comunicación de la TAD con la PC.....	70
Figura 5.5 Señal convertida de 0 a 5V.....	71
Figura 5.6 Señal de entrada al ADC.	71
Figura 5.7 Señal convertida de 0 a 10 V.....	72
Figura 5.8 Señal de entrada al ADC de 0 a 10 V.....	72
Figura 5.9 Interfaz para la configuración de las salidas en PWM.	72
Figura 5.10 Medición de la frecuencia de la señal PWM generada por la TAD.	73
Figura 5.11 Medición del ciclo de trabajo.	73
Figura 5.12 Configuración del contador de eventos.	73
Figura 5.13 Señal de prueba para el modulo contador de eventos.	73
Figura 5.14 Interfaz del módulo E/S digitales.	74
Figura 5.15 Conexión física de las entradas al puerto B.	74
Figura 5.16 Configuración del puerto digital B como salida.....	75
Figura 5.17 Salida del puerto digital B.....	75
Figura 5.18 Interfaz de configuración del módulo DAC y señal de prueba generada.....	75
Figura 5.19 Señal Convertida recuperada del osciloscopio.....	75
Figura 5.20 Medición del tiempo de conversión.	76
Figura 5.21 Diagrama a bloques del sistema I2C.	76
Figura 5.22 Sistema de prueba para el módulo I2C.....	77
Figura 5.23 Configuración de la GUI para la lectura del nodo Esclavo.....	77
Figura 5.24 Configuración del dato de entrada del nodo Esclavo 1.	77
Figura 5.25 Configuración de la GUI para la escritura del nodo Esclavo 1.	78
Figura 5.26 Datos de salida del nodo Esclavo 2.	78
Figura 5.27 Señal de entrada del ADC y señal de salida del DAC.....	79

Figura 5.28 Generación de una señal PWM a partir de una señal analógica.....	79
Figura A.1 Trama Escribir-Led y Lectura-Led.....	A-1
Figura A.2 Trama Inicializar-Pdigital.....	A-2
Figura A.3 Trama Lectura-Pdigital.....	A-2
Figura A.4 Trama Escribir-Pdigital.....	A-2
Figura A.5 Trama Inicializa-PWM.....	A-3
Figura A.6 Trama Escritura-PWM.....	A-3
Figura A.7 Trama Inicializa-ADC.....	A-4
Figura A.8 Trama Lectura-ADC.....	A-4
Figura A.9 Trama Inicializa-DAC.....	A-4
Figura A.10 Trama Escribir-DAC.....	A-5
Figura A.11 Inicialización-Evento.....	A-5
Figura A.12 Trama Escribe-Evento.....	A-5
Figura A.13 Trama Lectura-Evento.....	A-6
Figura A.14 Trama Inicializa_I2C.....	A-6
Figura A.15 Trama Escritura_I2C.....	A-7
Figura A.16 Trama Petición Lectura_I2C.....	A-7
Figura A.17 Trama de Lectura_I2C.....	A-7
Figura B.1 Mensajes de Windows mientras instala los drivers de la TAD.....	B-1
Figura B.2 Fin de la instalación del dispositivo.....	B-1
Figura B.3 Ventana del administrador de dispositivos.....	B-2
Figura B.4 Propiedades de la TAD.....	B-2
Figura B.5 Paleta de funciones para buscar los módulos de los recursos.....	B-3
Figura B.6 Ayuda del módulo PWM.....	B-3
Figura B.7 Código para el módulo PWM.....	B-4
Figura B.8 Interfaz de usuario para el módulo PWM.....	B-5
Figura C.1 Diagrama esquemático de la placa base.....	C-2

Figura C.2 Diagrama de la placa base de la TAD.	C-5
Figura C.3 Montaje de los componentes de la placa base.	C-5
Figura C.4 Diagrama esquemático para el microcontrolador.	C-6
Figura C. 5 PCB del microntrolador.	C-7
Figura C.6 Diagrama esquemático del circuito para el acondicionamiento de señales analógicas.	C-7
Figura C.7 PCB del lado opuesto al de los componentes-	C-8
Figura C.8 PCB del lado de los componentes.	C-9
Figura C.9 Diagrama de montaje de componentes.	C-9

Índice de Tablas

Tabla 1.1 Tabla de comparación en precio de las TAD.....	5
Tabla 2.1 Versiones del USB.....	17
Tabla 2.2 Paquetes PID [7].	20
Tabla 3.1 Asignación de los puertos del microcontrolador a los recursos de la TAD.....	39
Tabla 3.2 Opciones para voltaje de referencia del ADC [10].....	44
Tabla 3.3 Configuraciones típicas del Amplificador Operacional.....	49
Tabla 4.1 Terminales del módulo para inicializar la comunicación.	57
Tabla 4.2 Terminales del módulo para probar la comunicación.....	58
Tabla 4.3 Terminales del módulo de E/S Digitales.	59
Tabla 4.4 Terminales del módulo PWM.....	60
Tabla 4.5 Terminales del módulo ADC.....	61
Tabla 4.6 Terminales del módulo DAC.....	62
Tabla 4.7 Descripción de las Entradas/Salidas del módulo de contador de Eventos.....	63
Tabla 4.8 Descripción de las Entradas/Salidas del módulo de comunicación I2C.....	64
Tabla C.1 Descripción de las terminales de entrada/salida.	C-3
Tabla C. 2 Componentes de la placa base.	C-4
Tabla C. 3 Lista de componentes de la placa del microcontrolador.....	C-6

Tabla C. 4 Lista de componentes de la placa para el acondicionamiento de señal.....	C-7
---	-----

Resumen

En el presente trabajo de tesis se muestra el diseño y construcción del hardware y software necesarios para la implementación de una Tarjeta de Adquisición de Datos con conexión USB a una computadora, bajo el sistema operativo Windows.

Se utilizó el microcontrolador AT90USB646 de ATMEL, ya que este cuenta con los recursos necesarios para implementar el protocolo de comunicación USB. Este microcontrolador es el encargado de controlar la inicialización y ejecución de los recursos así como la comunicación con la computadora.

El software que permite manejar los recursos de la tarjeta y la interfaz gráfica con el usuario se desarrolló en la plataforma de LabView. Se desarrolló un VI para cada recurso con los que cuenta la tarjeta, los cuales son:

- ADC con 8 canales de 10 bits.
- DAC con 2 canales de 12bits.
- Contador de Eventos.
- PWM con 3 canales.
- 2 puertos de E/S digitales de 8 bits.
- Puerto de comunicación I2C.

Capítulo 1 Introducción

La instrumentación electrónica es el área de la tecnología que estudia los equipos realizados mediante circuitos y sistemas electrónicos, destinados a la medición, visualización, generación y conversión de señales eléctricas, así como los dispositivos o circuitos eléctricos que convierten una señal no eléctrica (temperatura, presión, etc.) en una señal eléctrica, que suelen recibir el nombre de sensores. De acuerdo con esto, los sistemas de instrumentación electrónica se pueden clasificar en tres grandes grupos [16]:

- a) *Instrumentos de medida y visualización*, son sistemas electrónicos que realizan la evaluación de uno o varios parámetros de una señal eléctrica y los presentan de forma gráfica, numérica o alfanumérica.
- b) *Instrumentos generadores de señal*, son sistemas electrónicos que tienen como misión generar señales eléctricas con características determinadas. Un ejemplo de este tipo de instrumento es el generador de señales.
- c) *Instrumentos convertidores de señal*, son dispositivos o circuitos electrónicos que convierten una señal eléctrica o no eléctrica, en otra señal eléctrica con características y rango determinados, sus parámetros contienen la información relevante de la primera señal.

En la actualidad existen diversos sistemas de instrumentación que combinan la instrumentación de medida, instrumentos generadores de señal e instrumentos convertidores de señal, por ejemplo un Sistema de Adquisición de Datos (SAD), puede ser definido como el conjunto de dispositivos lineales e interfaces que realizan la conexión entre los sensores de medición y una computadora, la cual realiza el procesado y almacenamiento de la información [15].

En la Figura 1.1 se muestra un diagrama a bloques de las partes que conforman un SAD.



Figura 1.1 Diagrama de un sistema de adquisición de datos.

El transductor permite tomar muestras de una señal física (temperatura, presión, intensidad luminosa, etc.) para convertirla en una señal eléctrica. Ciertos tipos de transductores requieren de un circuito adicional para ajustar los niveles de voltaje a los parámetros de entrada de la Tarjeta de Adquisición de Datos (TAD) y que ésta pueda interpretar los datos correctamente. Algunas TAD cuentan con este tipo de circuitos y el usuario debe tener en cuenta las características de la señal a introducir para evitar dañarla. La PC permite manipular o desplegar las señales digitales proporcionadas por la TAD, realiza algunas acciones de control y manda una respuesta a la TAD para que ésta genere señales de salida (analógicas o digitales) con ciertos parámetros de amplitud y frecuencia.

Una TAD es un dispositivo que digitaliza las señales eléctricas de manera que puedan ser procesadas y manipuladas a través de una interfaz de computadora. Los principales parámetros de una tarjeta de adquisición de datos son:

- El número de E/S analógicas,
- Número de E/S digitales,
- Contadores/temporizadores,
- Velocidad de muestreo,
- Puerto de comunicación,
- Precio.

La comunicación entre una TAD con una PC puede llevarse a cabo a través de diferentes puertos, como:

- Paralelo (LPT, Line Print Terminal),
- Puerto serial (RS-232, Recommended Standard),
- Componentes de conexión periférica (PCI, Peripheral Component Interconnect),
- Bus universal (USB, Universal Serial Bus),
- Ethernet.

En el mercado existe una gran variedad de modelos de TAD y su precio depende de las características con que cuente. En la Figura 1.2 se muestran dos ejemplos de TAD comerciales.

El gran avance en la tecnología ha propiciado que los sistemas electrónicos sean cada vez más compactos y fáciles de manipular por el usuario. Por su parte, las TAD han venido evolucionado y se han ajustado a las nuevas tecnologías, han podido migrar de conexiones internas de una PC, como el bus PCI, que si bien brinda una gran velocidad en la transmisión de datos, no es muy útil cuando se requiere que el sistema sea portable; posteriormente se popularizaron sistemas con conexiones a puertos externos como LPT y RS-232, los cuales dotaban al sistema de portabilidad.



- 8 entradas analógicas.
- 12 entradas/salidas digitales.
- 2 salidas analógicas.
- Bus de conexión USB 2.0 (12 Mb/s).
- Precio \$1,405 MN.
- Fabricada por National Instruments.
- Modelo NI-6008.



- 8 entradas analógicas.
- 16 entradas/salidas digitales,
- 2 salidas analógicas.
- Bus de conexión USB 2.0 (12 Mb/s).
- Precio \$130.00 USD.
- Fabricada por Measurement Computing
- Modelo PDM-1208FS.

Figura 1.2 Ejemplos de Tarjetas de Adquisición de Datos [18,19].

Actualmente en la mayoría de los sistemas electrónicos se ha estandarizado el puerto de comunicación USB, por su alta velocidad de transmisión de datos y su portabilidad. Por ello, las TAD que no trabajan en tiempo real, utilizan este puerto como medio de comunicación.

Por otro lado, el avance en el desarrollo de la arquitectura de los microcontroladores ha permitido incorporarle una mayor cantidad de recursos y mecanismos que permiten al usuario configurarlos con relativa facilidad.

Existe una gran gama de microcontroladores y cada uno cuenta con diferentes recursos que pueden ser usados para diversas aplicaciones.

Recientemente la compañía Atmel ha ampliado la familia AVR para incluir al microcontrolador AT90USB646, el cual cuenta con un módulo con la interfaz USB para realizar una comunicación con alguna PC. Esto lo hace factible para ser la base en el desarrollo de TAD.

Por ello, en esta tesis se desarrollará una TAD utilizando al microcontrolador AT90USB646, ya que los microcontroladores AVR son dispositivos de alto rendimiento ampliamente estudiados y usados en la UTM. Con esto se pretende profundizar en el conocimiento y evaluación de la familia AVR, permitiendo así comprender y manipular al

protocolo de comunicación USB y en un futuro inmediato poder desarrollar sistemas que utilicen esta interfaz en lugar de la interfaz serial estándar RS-232.

1.1 Planteamiento del Problema

La interfaz de comunicación USB es ampliamente usada en una TAD porque permite que un sistema sea portable y de fácil conexión con la PC, brindándole mayor comodidad al usuario. Debido a esto, se plantea desarrollar una tarjeta de adquisición de datos basada en el microcontrolador AT90USB646, utilizando una comunicación vía USB y la clase para Dispositivos de Interfaz Humana (HID) proporcionada por Atmel Corporation.

La interfaz USB se vuelve muy compleja cuando los datos del usuario no se ajustan a las clases de la norma USB (almacenamiento masivo, audio, vídeo, etc.), deben desarrollarse controladores específicos, esto requiere una cantidad significativa de tiempo. Debido a esto, Atmel ha desarrollado una clase para Dispositivos de Interfaz Humana (HID), la cual asegura una transferencia dúplex completa entre el dispositivo y la PC, así ahorrar tiempo y esfuerzos de desarrollo [4].

Si además se cuenta con un ambiente de evaluación y desarrollo basado en LabView, se tendrá una herramienta eficiente para la implementación de diversas aplicaciones.

1.2 Justificación

El desarrollo de una tarjeta de adquisición de datos representaría un costo de inversión bajo en comparación de las tarjetas más baratas que actualmente se encuentran en el mercado. La Tabla 1.1 Tabla de comparación en precio de las TAD. Tabla 1.1 muestra un comparativo en precio de las tarjetas de adquisición de datos comerciales de menor costo, con un estimado para la tarjeta que se planea desarrollar en el presente trabajo.

Tabla 1.1 Tabla de comparación en precio de las TAD.

Fabricante de TAD	Costo
National Instrument	\$1,405MN
Measurent Computing	\$139 dólares
Tarjeta bajo desarrollo	\$700 MN (costos materiales)

Una vez que se cuente con la TAD, podrá ser empleada en prácticas de laboratorio de diferentes cursos, como Metrología e Instrumentación, Sensores y Transductores, Control, etc. Y al ser esta de arquitectura abierta, los alumnos podrán agregar o quitar características para adecuarla a su aplicación.

Por otro lado, los estudiantes de las ingenierías de Electrónica y Mecatrónica de la UTM están familiarizados con los microcontroladores ATmega8 y ATmega16 de Atmel Corporation. El presente trabajo permitirá profundizar en el conocimiento y evaluación de la familia AVR y así poder migrar algunas aplicaciones al microcontrolador AT90USB646.

En consecuencia, las aplicaciones que actualmente se desarrollan utilizando la interfaz de comunicación RS-232, se podrán realizar con la interfaz de comunicación USB.

1.3 Objetivos

1.3.1 Objetivo General

Desarrollar y construir una Tarjeta de Adquisición de Datos con interfaz USB, empleando el microcontrolador AT90USB646 de ATMEL.

1.3.2 Objetivos particulares

- Evaluar la organización del microcontrolador AT90USB646 y comprender el funcionamiento de sus recursos internos orientados al manejo de la interfaz USB.
- Diseñar el hardware necesario para los recursos de la tarjeta.
- Desarrollar un programa para una PC bajo el entorno de desarrollo de LabVIEW para Windows.
- Evaluar la interacción de la tarjeta con la interfaz en la PC a través de la interfaz de comunicación USB.
- Documentar los resultados obtenidos, haciendo énfasis en el manejo de la interfaz USB del microcontrolador AT90USB646.

1.4 Organización del documento

El documento comprende seis capítulos y tres apéndices, los cuales se describen a continuación.

Capítulo 1. Se muestra una introducción del documento, se plantean el problema, la justificación así como el objetivo general y los objetivos particulares.

Capítulo 2. Se da una descripción general sobre la arquitectura de los microcontroladores AVR y particularmente se describen los recursos del microcontrolador AT90USB646, seguidamente se exponen los aspectos generales del protocolo de comunicación USB y finalmente se describe la plataforma de desarrollo de LabVIEW.

Capítulo 3. Se explica la metodología de diseño empleada para el desarrollo de los módulos de entradas/salidas digitales y analógicas, I2C, contadores de eventos y PWM, para su implementación en el microcontrolador y en la interfaz gráfica de LabVIEW. Se detalla la organización del sistema.

Capítulo 4. Describe el diseño de los módulos con las herramientas de desarrollo, su estructura individual y posteriormente la integración del sistema.

Capítulo 5. Se muestra la interacción Software-Hardware. Se realizan las pruebas pertinentes para verificar la funcionalidad de los módulos.

Capítulo 6. Conclusiones obtenidas con los ejemplos de evaluación (adquisición y generación de señales digitales y analógicas, comunicación I2C, contador de eventos, prueba de PWM y verificación de comunicación). También se consideran posibles trabajos futuros.

Apéndice A. Contiene la descripción de las tramas que se utilizan para la comunicación de la GUI con la TAD.

Apéndice B y C. Muestran los manuales de software y de hardware respectivamente. Brinda al usuario información de consideraciones técnicas necesarias para el uso de la TAD.

Capítulo 2 Marco teórico

Este capítulo contiene una breve descripción de los conceptos fundamentales para la realización de una tarjeta de adquisición de datos, la cual se comunica con una PC a través del puerto USB. Se describen las partes que conforman al sistema. La parte principal de una TAD es el microcontrolador que realiza las operaciones de adquisición y transferencia de datos, por eso es importante comprender su arquitectura. También se describe el protocolo de comunicación USB y el entorno de desarrollo de LabVIEW.

2.1 Microcontroladores

Un microcontrolador es un Circuito Integrado con muy alta escala de integración (VLSI, *Very Long Scale Integration*), cuenta con más de 10,000 transistores. Internamente contiene una Unidad Central de Procesamiento (CPU, *Central Processing Unit*), memoria para código, memoria para datos, temporizadores, fuentes de interrupción y otros recursos necesarios para el desarrollo de aplicaciones, por lo general con un propósito específico [21].

2.1.1 Organización de los Microcontroladores

Existen diferentes fabricantes de microcontroladores y cada uno maneja diferentes familias con una variedad de modelos, sin embargo, hay bloques que son comunes a todos, en la Figura 2.1 se muestra la organización típica de un microcontrolador y en las siguientes secciones se describe cada uno de los bloques.

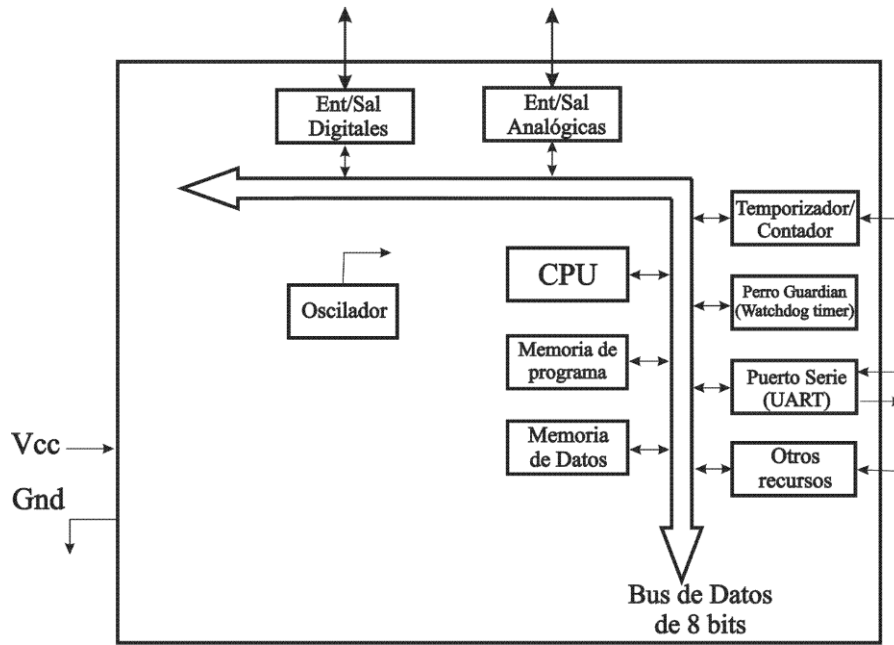


Figura 2.1. Organización de un microcontrolador [21].

2.1.1.1 La Unidad Central de Procesamiento

La CPU administra todas las actividades del sistema y ejecuta todas las operaciones sobre los datos, mediante las instrucciones ubicadas en la memoria de código, con las cuales se determina el comportamiento del sistema.

El trabajo de la CPU puede resumirse en tres tareas fundamentales: captura de una instrucción, decodificación de la misma y ejecución. Las CPU pueden estar organizadas de diferentes maneras, de acuerdo con ello, pueden clasificarse como CISC o RISC. Con CISC se hace referencia a computadoras con un Repertorio de Instrucciones Complejo (CISC, *Complex Instruction Set Computer*) y RISC es para referir a computadoras con un Repertorio de Instrucciones Reducido (RISC, *Reduced Instruction Set Computers*) [21].

En la Figura 2.2 se muestran los bloques típicos de una CPU, los cuales se pueden clasificar en dos grupos; por una parte el *Camino de Datos* y la *Unidad de Decodificación* que involucra los elementos en los que puede fluir la información cuando se ejecuta una instrucción y por la otra tenemos el *Control*, el cual determina qué elementos se activan dentro del camino de datos para la correcta ejecución de una instrucción.

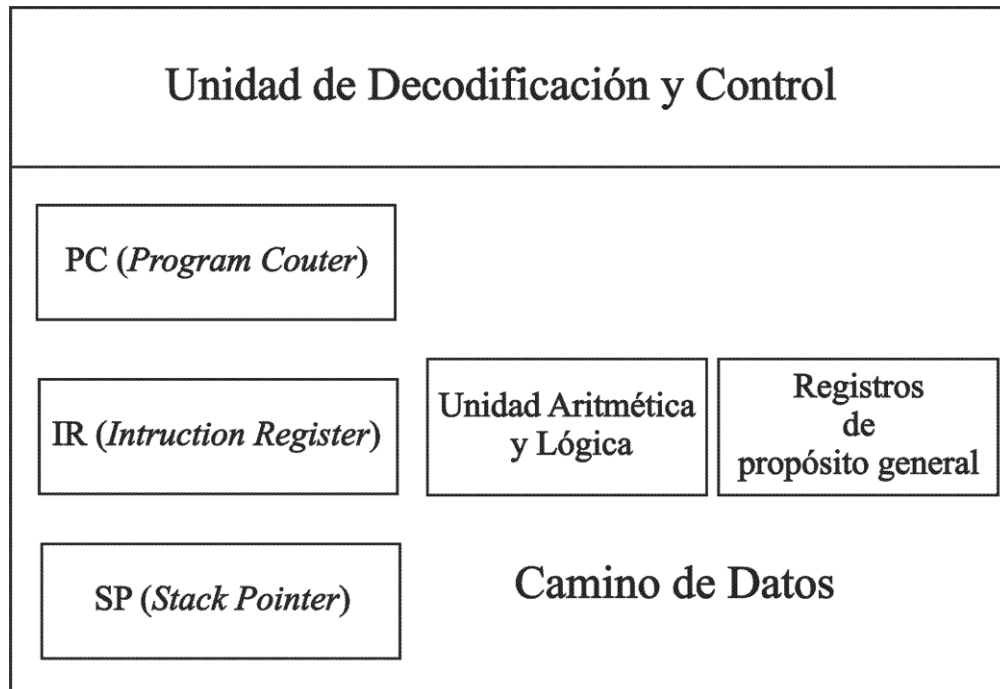


Figura 2.2. Elementos comunes en una CPU [21].

El Contador del Programa (PC, *Program Counter*), el Registro de Instrucción (IR, *Instruction Register*) y el Apuntador de Pila (SP, *Stack Pointer*), son registros con una función específica en una CPU.

El PC contiene la dirección de la instrucción que se va a ejecutar en un instante de tiempo determinado y mientras esa instrucción se ejecuta, el PC automáticamente actualiza su valor para apuntar a la siguiente instrucción a ejecutar. El registro IR contiene la cadena de bits que conforman la instrucción bajo ejecución, de esa cadena, la unidad de control considera el campo del opcode para determinar la activación de las señales en los demás elementos de la CPU. El SP contiene la dirección del tope de la pila, que es un espacio de almacenamiento utilizado durante la invocación de las rutinas. La Unidad Aritmética y Lógica (ALU, *Arithmetic Logic Unit*) es el bloque que se encarga de realizar las operaciones aritméticas y lógicas con los datos, también opera sobre direcciones para calcular el destino de un salto o la ubicación de una localidad a la que se tendrá acceso para una transferencia de memoria a registro o viceversa [21].

2.1.1.2 Sistema de memoria

Una computadora (y por lo tanto, también un microcontrolador) debe contar con espacios de memoria para almacenar los programas (código) y los datos. En relación a cómo se organizan estos espacios, se tienen dos modelos de computadoras, un modelo en donde el código y datos comparten el mismo espacio de memoria (arquitectura Von Neumann) y el otro en donde se tienen memorias separadas, una para código y otra para datos (Arquitectura Harvard).

2.1.1.3 Oscilador

La frecuencia a la que la CPU toma las instrucciones de la memoria del programa para su posterior ejecución, está determinada por el circuito de oscilación. El cual genera la frecuencia de trabajo a partir de un circuito RC, un resonador cerámico o un cristal de cuarzo externo.

2.1.1.4 Temporizador/Contador

Es un registro de n bits que se incrementa con una señal de reloj interna o con eventos externos. En el primer caso funciona como temporizador y en el segundo como contador de eventos. Como temporizador es utilizado para manejar intervalos de tiempo y como contador es la base para programar alguna tarea cada que ocurra una cantidad predeterminada de eventos externos al microcontrolador.

2.1.1.5 Perro Guardián

El Perro Guardián (WDT, *Watch Dog Timer*) es un temporizador que se compone de un registro de n-bits, cuando el WDT se desborda ocasiona un reinicio del sistema (*reset*). Su principal función es evitar que el microcontrolador se cicle en estados no contemplados.

2.1.1.6 Puerto serie

La mayoría de los microcontroladores cuentan con un receptor/transmisor universal asíncrono (UART, *Universal Asynchronous Receiver Transceiver*), para comunicación serial

con dispositivos o sistemas externos, bajo protocolos y razones de transmisiones estándares. La comunicación serial puede ser de dos tipos:

- Síncrona: Además de la línea de datos se utiliza una línea de reloj.
- Asíncrona: Sólo hay líneas para los datos, el transmisor y el receptor se deben configurar con la misma velocidad de transferencia (bits/segundos, *Baud Rate*), además, se debe definir el mismo formato para cada trama.

2.1.1.7 Entradas/Salidas Digitales

Los microcontroladores usan componentes de Entrada/Salida digitales para intercambiar datos con otros dispositivos. A diferencia de un puerto serie, en donde se transfiere un bit a la vez, en los puertos digitales es posible realizar un intercambio de bytes. Todos los microcontroladores cuentan con puertos digitales, aunque el número puede variar entre dispositivos.

2.1.1.8 Entradas/Salidas Analógicas

Para entradas analógicas algunos microcontroladores incorporan Convertidores Analógico a Digital (ADC, *Analogic-Digital Converter*) o comparadores analógicos. Estos recursos permiten obtener información analógica del exterior, para monitorear parámetros como temperatura, velocidad, humedad, etc. Para salidas analógicas no es común que un microcontrolador incluya un Convertidor Digital a Analógico (DAC, *Digital-Analogic Converter*), para compensar esto, algunos microcontroladores incluyen salidas con Modulación por Ancho de Pulso (PWM, *Pulse Width Modulation*).

2.1.2 Clasificación de los Microcontroladores

Existen diferentes formas de clasificar a los microcontroladores, a continuación se describen las clasificaciones típicas.

Por la arquitectura de la CPU, los microcontroladores pueden clasificarse como RISC o CISC, aunque casi todos los nuevos microcontroladores son RISC. Por el tamaño de los datos, se tienen microcontroladores de 4, 8, 16 y 32 bits, por el tamaño de los datos debe entenderse

el tamaño de los registros de trabajo y por lo tanto, corresponde con el número de bits de los operandos en la ALU. Tomando como base la organización y el acceso a la memoria de código y datos, se tienen 2 modelos: Arquitectura Von Neumann y Arquitectura Harvard. Considerando la forma en cómo se acceden a los datos para realizar las operaciones, se tienen cuatro modelos: Arquitectura tipo Pila, tipo Acumulador, tipo Registro a Memoria y tipo Registro-Registro [21].

2.1.3 Organización de los Microcontroladores AVR de Atmel.

Los microcontroladores AVR incluyen un procesador **RISC** de **8 bits**, su arquitectura es del tipo **harvard** y sus operaciones se realizan bajo un esquema **Registro-Registro**. El núcleo AVR combina un conjunto de instrucciones con 32 registros de propósito general. Todos ellos están directamente conectados a la ALU, permitiendo que se tenga acceso a dos registros independientes en una sola instrucción ejecutada en un ciclo de reloj. La arquitectura resultante procesa al código de forma más eficiente, pues alcanza rendimientos hasta diez veces más rápido que los microcontroladores convencionales con arquitectura CISC [21,11].

2.1.3.1 El Núcleo AVR

La organización interna de los microcontroladores de ATMEL se fundamenta en el núcleo AVR, el núcleo es la unidad central de procesamiento (CPU), es decir, es el hardware encargado de la captura, decodificación y ejecución de instrucciones, su organización se muestra en la Figura 2.3.

En torno al núcleo se encuentra un bus de 8 bits al cual están conectados los diferentes recursos del microcontrolador, estos recursos pueden diferir entre dispositivos. La principal función de la CPU es asegurar la correcta ejecución de programas. La CPU debe tener acceso a los datos, realizar cálculos, controlar periféricos y manejar interrupciones.

Para maximizar el rendimiento y paralelismo, el AVR usa una arquitectura Harvard con memorias y buses separados para el programa y para los datos. El programa se ubica en la memoria flash y los datos están en 3 espacios diferentes: En el archivo de registros (32 registros de 8 bits), en la SRAM y en la EEPROM [21].

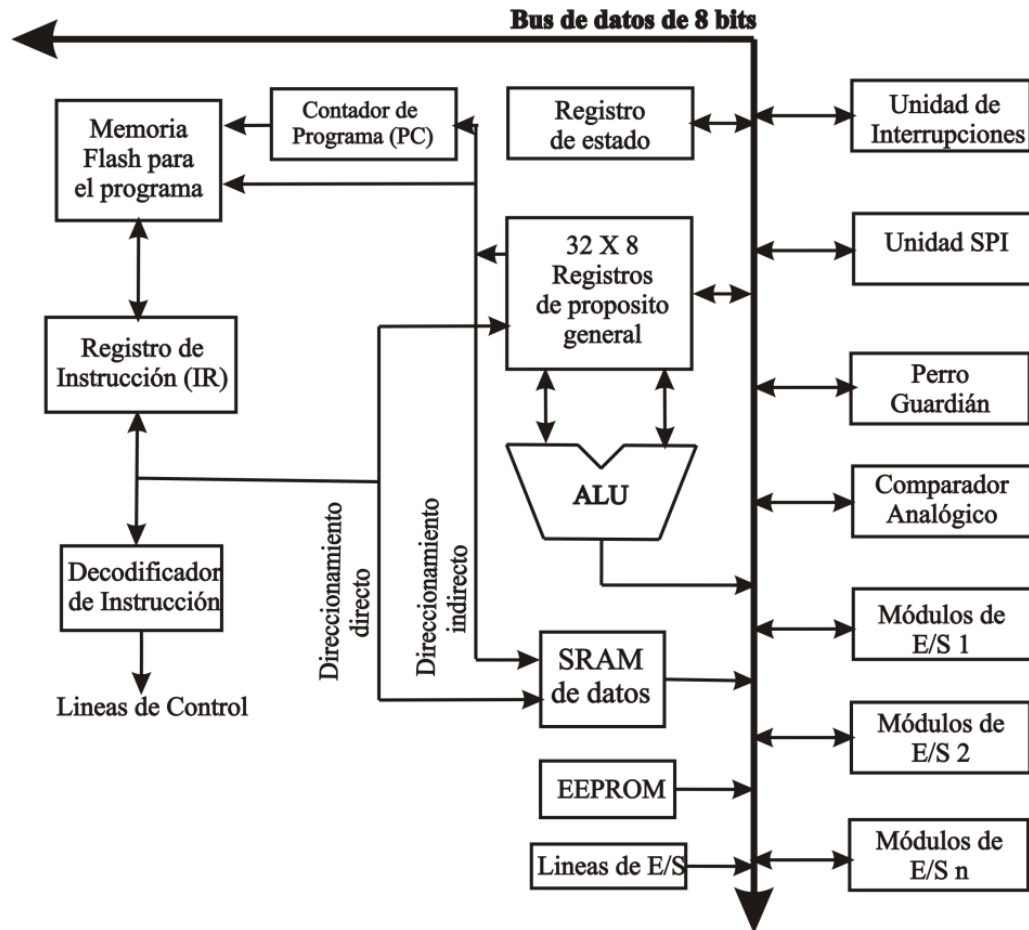


Figura 2.3 Diagrama a bloques del núcleo AVR [10].

De la memoria flash se obtiene cada instrucción del programa y se coloca en el registro IR para su decodificación y ejecución. La memoria flash es direccionada por el registro PC o bien, por uno de los registros de propósito general. El PC es el registro que indica la ubicación de la instrucción a ejecutar, sin embargo, es posible que un registro de propósito general proporcione esta dirección a modo de que funcione como apuntador y se haga un acceso utilizando direccionamiento indirecto [21].

La ALU soporta operaciones aritméticas y lógicas entre los 32 registros de propósito general o entre un registro y una constante, para cualquier operación, al menos uno de los operandos es uno de los registros. Los 32 registros son la base para el procesamiento de datos porque la arquitectura es del tipo registro-registro, esto implica que si un dato de SRAM o de EEPROM va ser modificado, primeramente deberá ser llevado a cualquiera de los 32 registros de 8 bits, dado que todos tienen la misma jerarquía [21].

2.2 Bus Universal Serial

En esta sección se describe el proceso evolutivo y las áreas de aplicación del protocolo de comunicación del Bus Universal Serial (USB, *Universal Serial Bus*), así mismo, se describe brevemente la especificación USB 2.0.

2.2.1 Aspectos generales del USB

El USB es un estándar de comunicaciones industrial que define los cables, conectores y protocolos usados en un bus para el intercambio de datos entre una computadora y una amplia gama de periféricos. Permite conectar, comunicar y proveer de alimentación eléctrica, desde el mismo bus.

Los periféricos conectados comparten el ancho de banda USB a través de un protocolo basada en mensajes (*tokens*) [11]. El bus permite que los periféricos puedan ser conectados, configurados, utilizados y desconectados, mientras que el Host y otros periféricos se encuentran en operación [17].

En la actualidad, prácticamente todos los dispositivos portátiles utilizan la interfaz USB para conectarse con una PC. Como estos productos han aumentado su popularidad, hay una creciente necesidad de comunicación entre los dispositivos portátiles y una computadora vía USB y directamente entre dispositivos cuando una computadora no está disponible [17].

2.2.2 Historia y evolución del USB

En 1995 las compañías *Intel*, *HP*, *Microsoft*, *Renesas Electronics* y *ST-Ericsson* crearon y estandarizaron el USB bajo el auspicio de una organización llamada Foro de Implementadores del Bus Universal Serial (USB-IF, *Universal Serial Bus Implementers Forum*). El objetivo fue crear un bus serial de baja y alta velocidad que permitiera el acceso compartido, disponibilidad, robustez, auto configuración y facilidad de uso [1].

Un hecho importante que permitió que sistemas con conexiones USB evolucionaran para ser más eficientes fue que las computadoras han alcanzado rendimientos cada vez mayores y son capaces de procesar grandes cantidades de datos. Además, nuevos tipos de dispositivos, formatos de medios de comunicación y de almacenamiento de bajo costo están convergiendo en brindar a los usuarios finales una mayor funcionalidad. En la Tabla 2.1 se muestran las diferentes versiones de la interfaz USB y sus principales características [23,27].

Tabla 2.1 Versiones del USB.

Version USB	Velocidad de Transmisión	Tipo de Transmisión	Año de publicación
USB 1.1	Baja (1.5 Mbps) Completa (12Mbps)	Half duplex	1996
USB 2.0	Baja (1.5Mbps) Completa (12Mbps) Alta (480Mbps)	Flujo de datos unidireccional	2000
USB 3.0	Baja (1.5Mbps) Completa (12Mbps) Alta (480Mbps) Súper (5 Gbps)	Dual-simplex	2009

2.2.3 Áreas de aplicación

El USB se ha convertido en una interfaz de comunicación muy popular, debido a su simplicidad para las aplicaciones de usuario final (*Plug and Play*, sin necesidad de reiniciar).

Puesto que el USB es un estándar para la conexión de periféricos, dispositivos como ratones, teclados, joysticks, escáneres, cámaras digitales, teléfonos móviles, reproductores multimedia, impresoras, sistemas de adquisición de datos, módems, tarjetas de red, tarjetas de sonido, discos duros externos, gadgets y demás dispositivos de interfaz humana usan este bus para comunicarse con la PC [7].

2.2.4 Protocolo de comunicación

Como se muestra en la Figura 2.4, la conexión de un dispositivo USB con el Host requiere la interacción entre un número de capas y entidades. La capa de la interfaz del bus USB proporciona una conexión física, señalización de la conexión y conectividad de los paquetes entre el Host y un dispositivo USB. En esta capa se encuentra el controlador host USB, el cual integra hardware y software que permite a los dispositivos USB ser conectados a un host [7].

En la capa de dispositivo USB, el software del sistema USB (software que soporta el estándar USB en un sistema operativo en particular) utiliza el controlador Host para gestionar las transferencias de datos entre el host y los dispositivos USB.

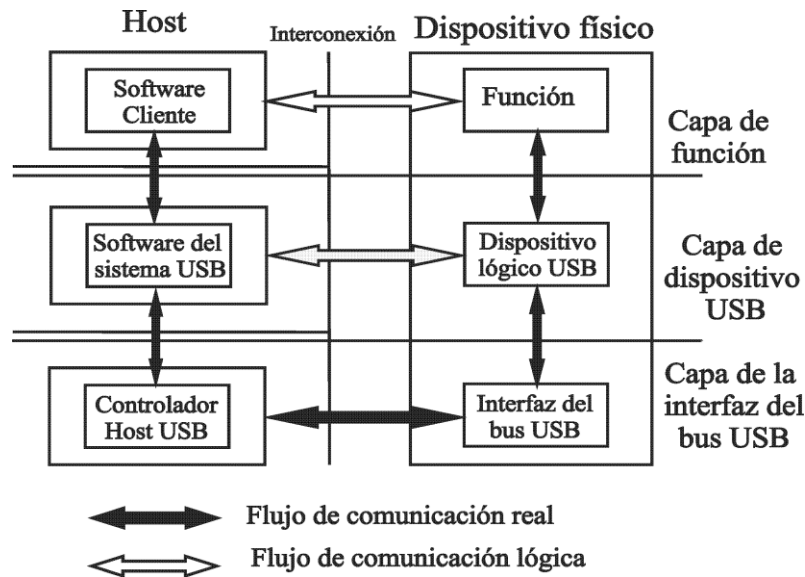


Figura 2.4 Modelo de la capa de comunicación [7].

Los dispositivos USB que proveen de funcionalidad adicional al Host son conocidos como funciones. El cliente describe todas las entidades de software que son responsables de la interacción directa con los dispositivos USB. Cuando los dispositivos están conectados al sistema, estos clientes pueden interactuar directamente con el hardware periférico [7].

El protocolo de acceso al canal de comunicación (*media access protocol*) es de tipo *polled* (con sondeos continuos sobre la configuración). En el USB, los datos se envían de manera serial, agrupados en paquetes. Los paquetes se agrupan para formar las Transacciones y las Transacciones se agrupan para formar las Transferencias. Las Transferencias son las estructuras de datos que tienen sentido para el Software que se ejecuta en el Host (Maestro en un sistema USB) [7].

Cada transacción se inicia cuando el controlador Host envía un paquete denominado *token*, describiendo el tipo y la dirección de la transacción, la dirección del dispositivo USB y el número del finalizador (*endpoint*). El dispositivo USB que es direccionado se encarga de decodificar apropiadamente el campo de dirección. En una determinada transacción, los datos se transfieren ya sea desde el host a un dispositivo o desde un dispositivo a la máquina. La fuente de la transacción posteriormente envía un paquete de datos o indica que no tiene datos que enviar. El destinatario, por lo general, responde con un paquete llamado *handshake* para que ambas partes se reconozcan [7].

Todos los paquetes comienzan con un campo de sincronización (SYNC), utilizado por el hardware de entrada para alinear los datos entrantes con el reloj local. Un identificador de paquete (PID, *Identifier Packet*) sigue inmediatamente después del campo de sincronización en cada paquete de USB. Consta de un campo de cuatro bits para el tipo de paquete seguido por un campo de cuatro bits de verificación como se muestra en la Figura 2.5 [7].

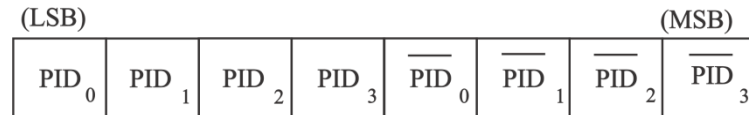


Figura 2.5 Formato del paquete PID.

El PID indica el tipo de paquete y, por inferencia, el formato del paquete y el tipo de detección de errores aplicados al paquete. Un error del PID existe si los cuatro bits de verificación PID no se complementan de sus respectivos bits de paquetes de identificación. En la Tabla 2.2 se muestra los tipos de paquetes PID y una breve descripción [7].

Seguido del campo PID está el campo de dirección (*ADDR*) para indicar el destino del paquete. Consta de 7 bits, por lo que permite que puedan ser direccionados hasta 127 dispositivos. En la Figura 2.6 se muestra el formato del campo de dirección [7].

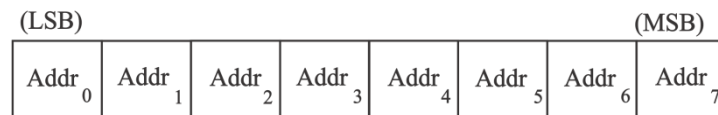


Figura 2.6 Formato del campo de dirección.

En seguida está el campo del *Endpoint* (ENDP) que consta de 4 bits, en la Figura 2.7 se muestra el formato de este campo. Permite mayor flexibilidad direccionando a las funciones en las que se requiere más de un *Endpoint*. Excepto por el *Endpoint* cero, los números del *Endpoint* corresponden a funciones específicas [7].

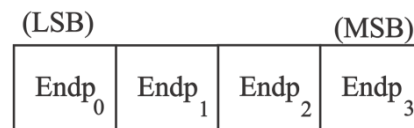


Figura 2.7 Formato para el campo Endpoint.

Posteriormente se encuentra el campo de número de muestras (*Frame number field*) compuesto por 11 bits y es incrementado por el Host en cada muestra. A continuación está el campo de datos, el cual varía entre 0 y 1024 bytes. Después del campo de datos se encuentra el

verificador de redundancia cíclica (CRC, *Cyclic Redundancy Checks*) considerado como campo de protección [7].

Tabla 2.2 Paquetes PID [7].

Tipo de paquete PID	Nombre	Descripción
<i>Token</i>	OUT	Informa al dispositivo que el Host va a enviar información
	IN	Informa al dispositivo que el Host va a leer información
	SOF	Marcador de inicio de trama y número de muestras (<i>frame</i>)
	SETUP	Se usa para comenzar el control de transferencia
<i>Data</i>	DATA 0 y 1	Paquete de datos par e impar
	DATA2	Paquete de datos PID de alta velocidad, gran ancho de banda para transacciones isócronas de micro-muestras (<i>microframes</i>)
	MDATA	Paquete de datos PID de alta velocidad para la división y gran ancho de banda de transacciones isócronas.
<i>Handshake</i>	ACK	Permite saber que el paquete se recibió con éxito
	NAK	El Receptor no puede aceptar los datos o el Transmisor no puede enviar datos
	STALL	El <i>Endpoint</i> está detenido o una solicitud en un segmento de control está detenida.
	NYET	No hay respuesta del receptor
<i>Special</i>	PRE	(<i>Token</i>) Habilita el tráfico del bus a dispositivos de baja velocidad
	ERR	(<i>Handshake</i>) Error de transacción
	SPLIT	Provee transacciones adicionales
	PING	(<i>Token</i>) Sondeo del control de flujo de alta velocidad para un <i>Endpoint</i> tipo <i>bulk/control</i>
	Reserved	Reservado

2.2.4.1 Especificaciones Eléctricas

Los datos son transmitidos en forma serie, a través de 2 líneas denominadas V+ y V-. Además, la interfaz USB cuenta con 2 líneas para establecer los niveles de voltaje (VCC y GND), toman la alimentación del Host o de una fuente externa (5 V, 500 mA máximo). En la Figura 2.8 se muestra la organización del cable USB [7].

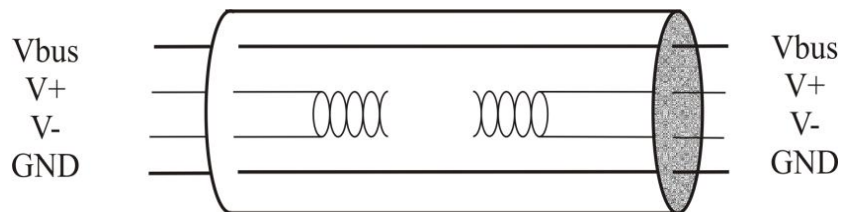


Figura 2.8 Cable USB [27].

La comunicación en un sistema USB se hace punto a punto a tres diferentes velocidades [7, 6].

- Baja velocidad (*Low-Speed*) es de 1.5 Mb/s.
- Máxima velocidad (*Full-Speed*) es de 12 Mb/s.
- Alta velocidad (*High-Speed*) es de 480 Mb/s

El nivel de voltaje que define a cada estado varía dependiendo de la velocidad que se esté empleando. Para baja velocidad, se toma un 1 diferencial cuando D+ es menor a 2.8V y D- no es mayor que 0.3V respecto a la señal de tierra. Se toma como un 0 diferencial cuando D- es menor a 2.8V y D+ no es mayor que 0.3V con respecto a tierra.

Para velocidad completa, se toma un 1 diferencial cuando D+ es menor a 2 V respecto a tierra y la diferencia entre D+ y D- es mayor a 200mV. Se toma como 0 diferencial cuando D- es menor a 2V y la diferencia entre D+ y D- es mayor a 200 mV.

Para alta velocidad, se toma un 1 diferencial cuando D+ es menor a 0.36V y D- no es mayor a 0.01V. Un 0 diferencial existe cuando D- es menor que 0.36V y D+ no es mayor que 0.01V [7].

2.2.4.2 Especificaciones Mecánicas

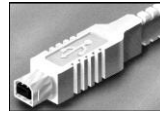
Los dispositivos USB que operan en alta velocidad (480 Mbps) y velocidad completa (12 Mbps) requieren el uso de un cable blindado con dos conductores de potencia y conductores de par trenzados de señal. Para baja velocidad (1.5 Mbps) se recomienda, pero no

es necesario, el uso de un cable con conductores de par trenzado de señal. El icono USB en los conectores es un indicador visual para obtener la orientación correcta [7].

En la Figura 2.9 se muestran los tipos de conectores USB. Los conectores tipo A son el principal medio de conexión entre los dispositivos USB y un Host o el puerto de un hub o un dispositivo USB. Todos los dispositivos USB tienen el conector estándar macho tipo A. Los conectores tipo B permiten que los dispositivos USB cuenten con un cable estándar desmontable y el usuario final pueda reemplazarlo fácilmente [7].



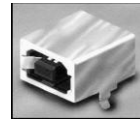
Conector macho
tipo A



Conector macho
tipo
B



Conector hembra
tipo A



Conector hembra
tipo B

Figura 2.9 Conectores USB [7].

2.2.4.3 Especificaciones Funcionales

Hay sólo un nodo Maestro (*Host*) en cualquier sistema con interfaz USB y éste es el operador maestro del bus, su interfaz USB puede ser implementada a partir de una combinación de hardware, firmware o software. El Host USB interactúa con los dispositivos USB a través del controlador Host [3].

Existen dos tipos de clases de dispositivos USB: hubs y unidades. Sólo los hubs tienen la capacidad de proveer puntos adicionales para conexión.

La topología física del bus USB es de tipo estrella. Cada segmento de cable es una conexión punto a punto entre el Host USB y un hub o un dispositivo USB. En la Figura 2.10

se muestra la topología de la interfaz USB. Debido a las limitaciones de tiempo de propagación en el hub y el cable, el número máximo de niveles es 7 (incluyendo el nivel del Host). El Host direcciona a los dispositivos Esclavos USB mediante un número de 7 bits, en consecuencia, es posible direccionar hasta 128 dispositivos por hub [7, 3].

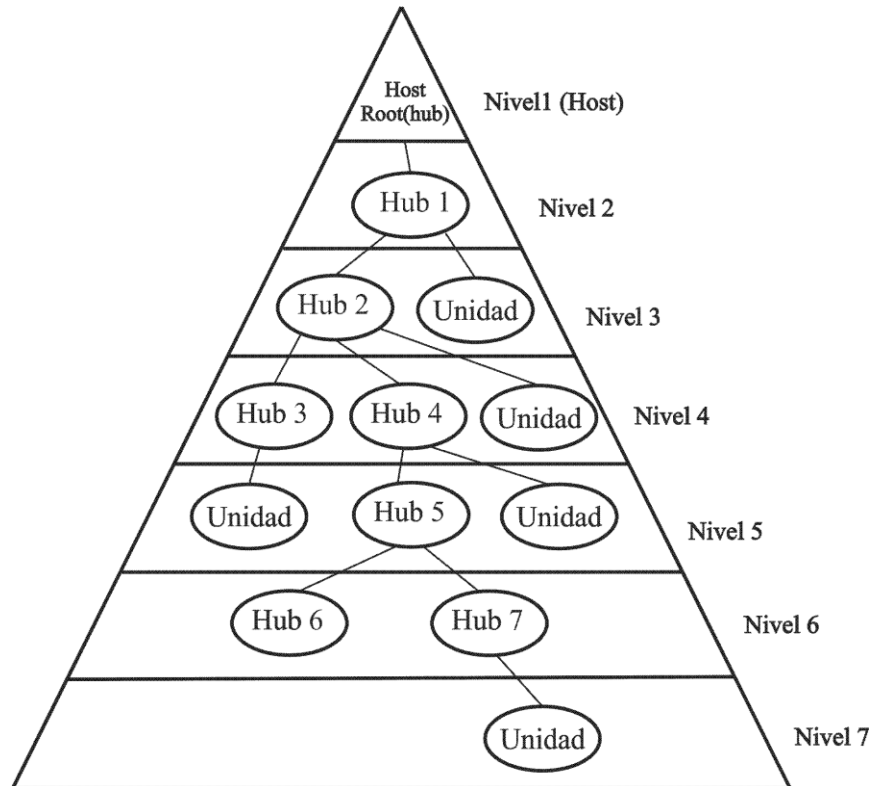


Figura 2.10 Topología estándar USB [3].

Algunos ejemplos de dispositivos USB son:

- Dispositivos de interfaz humana, tales como ratones, teclados, controles de juegos, etc.
- Dispositivos de imagen como cámaras, escáner e impresoras.
- Dispositivos de almacenamiento masivo como drivers de CD-ROM, floppy y DVD.

2.2.4.4 Estados de los Dispositivos USB

En la Figura 2.11 se muestran los estados de una unidad USB. Algunos son visibles para el Host USB y otros son internos de la unidad USB [7].

Cuando una unidad USB es conectada o desconectada del USB, el Host ejecuta un proceso llamado enumeración para identificar y manejar los estados de la unidad. Durante este

proceso, el servidor pregunta a la unidad algunos valores para identificar y cargar los controladores correctamente [7].

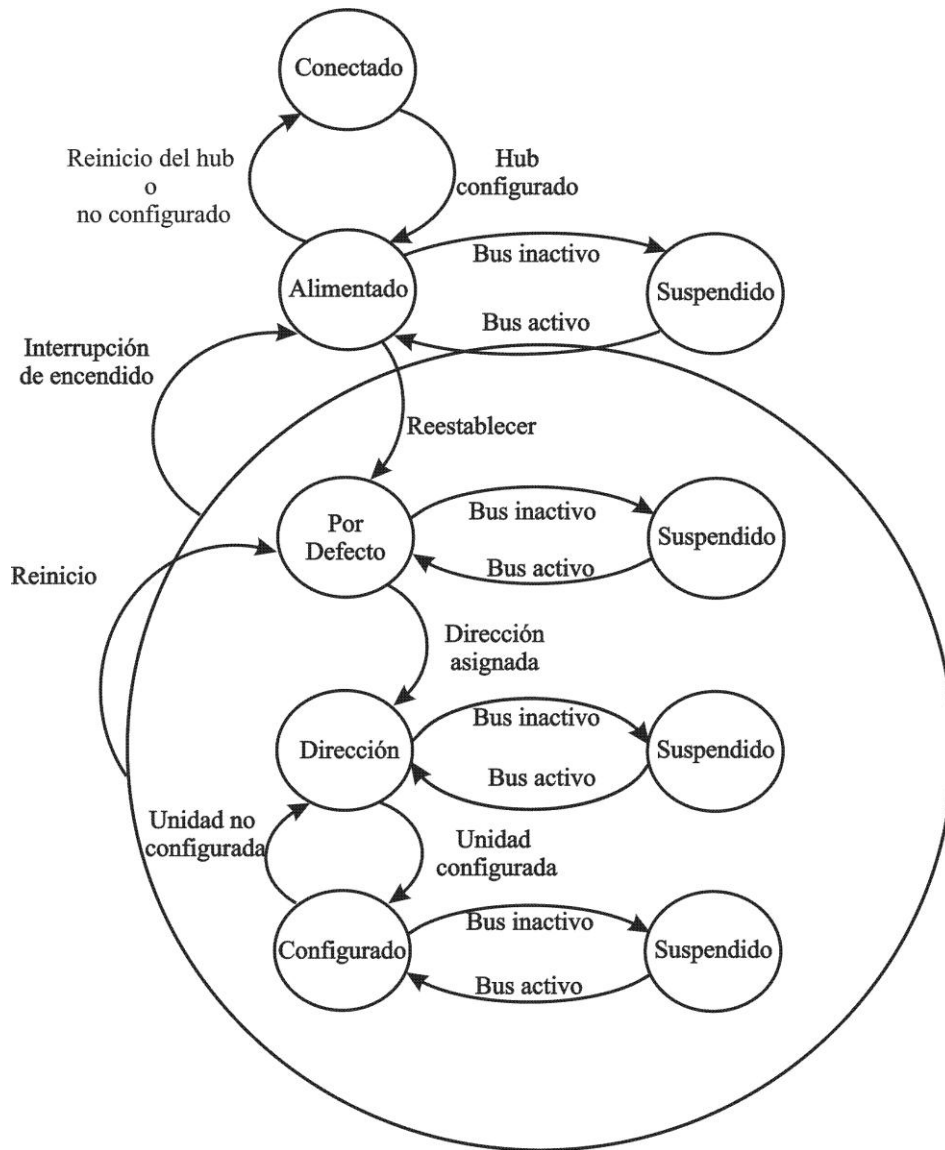


Figura 2.11 Estados de la unidad USB [7].

Cuando la unidad es conectada, se realizan las siguientes acciones [7]:

- El hub al que se conecta el dispositivo USB informa al Host del evento. En este punto, el dispositivo está en el estado de encendido y el puerto al cual se conectó está deshabilitado.
- El Host determina el motivo del cambio mediante la consulta al hub.

- El Host sabe a qué puerto se ha conectado el dispositivo USB, espera entonces por lo menos 100 ms para permitir la terminación del proceso de inserción y que se establezca la conexión eléctrica. El Host emite un comando para activar y restaurar el puerto.
- Cuando la señal de reset (reinicio) se ejecuta, el puerto es habilitado. La unidad USB está en el estado de Default (por defecto) y no puede utilizar más de 100mA del bus, todos sus registros se han reiniciado y la unidad USB responde a la dirección por defecto (cero).
- El Host asigna una dirección única a la unidad USB, cambiando a la unidad USB al estado *dirección*.
- Antes de que la unidad USB reciba una dirección única, el Host lee los descriptores de la unidad USB para determinar cuál es el tamaño máximo de los datos que esta unidad va a utilizar.
- El Host lee toda la información de configuración de la unidad USB. Este proceso lleva algunos milisegundos para que se complete.
- Basado en la información de configuración y en cómo la unidad USB puede ser usada, el Host asigna un valor de configuración a la unidad USB. La unidad USB pasa al estado de configuración y puede señalar la cantidad de energía que utilizará. Desde el punto de vista de la unidad USB, está lista para usarse.

Cuando una unidad USB es removida, el hub envía una notificación al Host. Al recibir la información, el Host actualiza la información topológica.

2.2.4.5 Descriptores USB

Los descriptores son utilizados por las Unidades USB para informar al Host de sus requerimientos. Un descriptor es una estructura de datos con un formato definido. Cada descriptor comienza con un campo de un byte para identificar el tipo de descriptor. Cada unidad USB debe tener por lo menos los descriptores que se muestran en la Figura 2.12 para que pueda ser reconocida por el servidor [3, 5, 22].

Unidad Descriptor: El dispositivo USB sólo puede tener un descriptor. Este descriptor muestra información sobre la versión del dispositivo USB, el tamaño máximo de los paquetes,

el identificador del fabricante, el identificador del producto, la versión del producto o el número de las posibles configuraciones que el dispositivo puede tener.

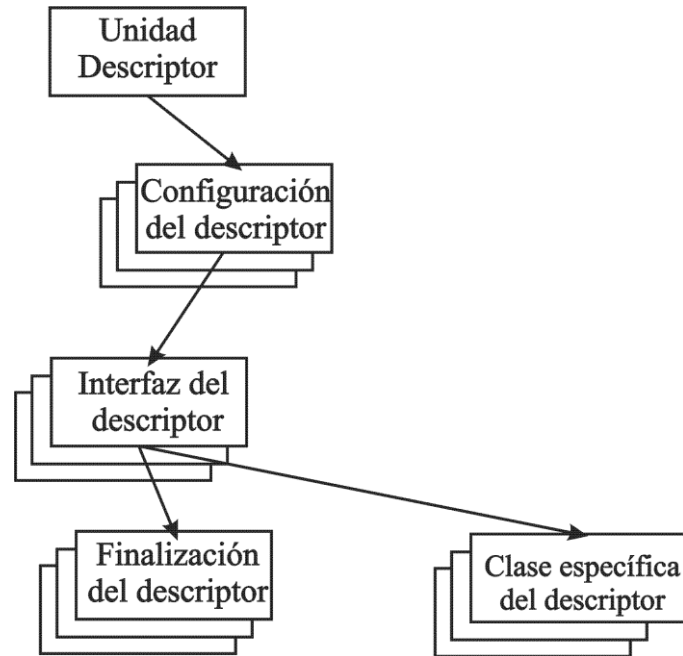


Figura 2.12 Descriptor USB [3].

Configuración del descriptor: El dispositivo USB puede tener más de una configuración, aunque la mayoría de los dispositivos usan sólo una. Este descriptor especifica la forma en que será alimentado (alimentación a través del bus o alimentación externa), el máximo voltaje que soporta, las interfaces que pertenecen al dispositivo, el tamaño total de todos los descriptors de datos, etc. Por ejemplo, un dispositivo puede tener dos configuraciones, una para cuando es alimentado a través del bus y la otra cuando es alimentado por una fuente externa.

Interfaz del descriptor: Un dispositivo puede tener más de una interfaz. La principal información dada por este descriptor es el número de criterios de valoración utilizados por esta interfaz, la clase y la subclase USB.

Finalización del descriptor: Este descriptor incluye los parámetros de la dirección de entrada/salida (IN/OUT), el tipo de transferencia soportada (control, interrupción, bulk e isócrona) y el intervalo de los datos en caso de una transferencia por interrupción.

La arquitectura USB comprende cuatro tipos básicos de transferencia de datos [22, 24]:

- *Control:* Usada para configurar un dispositivo en el momento de conectarlo y puede ser usada por dispositivos de propósito específico.

- *Interrupción*: Se utiliza para cuando se trabaja en tiempo real y se requiere seguridad en la transmisión.
- *Bulk*: Se usa cuando se manejan grandes cantidades de datos por lo que requiere de un ancho de banda suficiente.
- *Isócrona*: Ocupa una cantidad del ancho de banda USB configurando la latencia de entrada de los datos. También llamada transferencia de flujo continuo en tiempo real.

2.3 Entorno de desarrollo LabVIEW

LabVIEW es un entorno de programación gráfica usado por ingenieros e investigadores para desarrollar sistemas sofisticados de medición, pruebas y control. Esta plataforma de desarrollo fue creada por National Instruments y actualmente funciona sobre sistemas operativos como MAC, UNIX, Windows y GNU/Linux [13].

Los programas en LabVIEW no se escriben en forma textual, en lugar de ello se utilizan íconos gráficos y cables en una especie de diagrama de flujo. El entorno de desarrollo cuenta con una gran cantidad de bloques prediseñados con funciones específicas que proporcionan al usuario facilidad en el desarrollo de programas. Los programas en LabVIEW son llamados instrumentos virtuales (VIs, *Virtual Instruments*), porque con su apariencia y funcionamiento imitan a los instrumentos reales. Cada VI consta de dos partes diferentes [14]:

- El *Panel Frontal* es la interfaz con el usuario, permite interactuar con el usuario cuando el programa se está ejecutando. Los usuarios podrán observar los datos del programa actualizados en tiempo real. En esta interfaz se definen las entradas por medio de botones, campos de texto, listas numéricas, selectores booleanos, etc. También se definen las salidas con indicadores booleanos, textuales, numéricos o gráficos.
- En la interfaz del *Diagrama de Bloques* se colocan e interconectan los íconos para realizar una determinada función.

Gracias a su entorno gráfico, LabVIEW facilita la programación para desarrollar sistemas que requieren la comunicación de la PC con hardware a través de módulos con

puertos RS-232, USB y Ethernet. Ofrece una integración de diversos dispositivos de hardware y brinda bibliotecas integradas para un análisis avanzado y visualización de datos [13].

LabVIEW cuenta con métodos para la comunicación de software desarrollado en otros lenguajes, estos métodos implican una de las siguientes acciones [26]:

- Utilizar protocolos específicos de la plataforma.
- Usar llamadas a funciones de los siguientes tipos de bibliotecas compartidas:
 - Bibliotecas de enlace dinámicos (DLL) en Windows.
 - Fragmentos de códigos en Macintosh.
 - Bibliotecas compartidas en UNIX.
- Crear un código de interfaz de nodo (CIN, Code Interfaz Node) para llamar a código escrito específicamente para vincular VIs.

Las llamadas de funciones de biblioteca de LabVIEW y del CIN se realizan mediante objetos gráficos escritos en lenguaje convencional de LabVIEW. Aparecen en el diagrama a bloques como iconos con terminales de entrada y salida. La vinculación de código externo a LabView incluye los siguientes pasos [13]:

- Compilar el código fuente y vincular al código ejecutable. Si ya se cuenta con un archivo DLL compilado, este paso no es necesario.
- LabVIEW llama al código ejecutable cuando el nodo se ejecuta.
- LabVIEW pasa los datos de entrada desde el diagrama de bloques al código ejecutable.
- LabVIEW retorna los datos del código ejecutable al diagrama a bloques.

2.3.1 Llamadas a las funciones de una biblioteca

Las llamadas a las funciones de una biblioteca son más apropiadas cuando se tiene el código al que se desea llamar, o si se tiene conocimiento del proceso de creación de normas de bibliotecas compartidas. Debido a que una biblioteca utiliza un formato estándar entre varios entornos de desarrollo, se puede usar casi cualquier entorno de desarrollo para crear una biblioteca que LabVIEW pueda llamar. En la Figura 2.13 se muestra el módulo de LabVIEW que permite llamar a las funciones de una biblioteca [26].

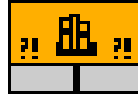


Figura 2.13 Módulo para llamadas a funciones de una biblioteca.

En la configuración de este módulo se debe especificar la ruta de la DLL, la función a la que se quiera tener acceso, así como los parámetros de entrada y valor de retorno de ésta. En la Figura 2.14 se muestra la ventana en la que se pueden configurar estos parámetros.

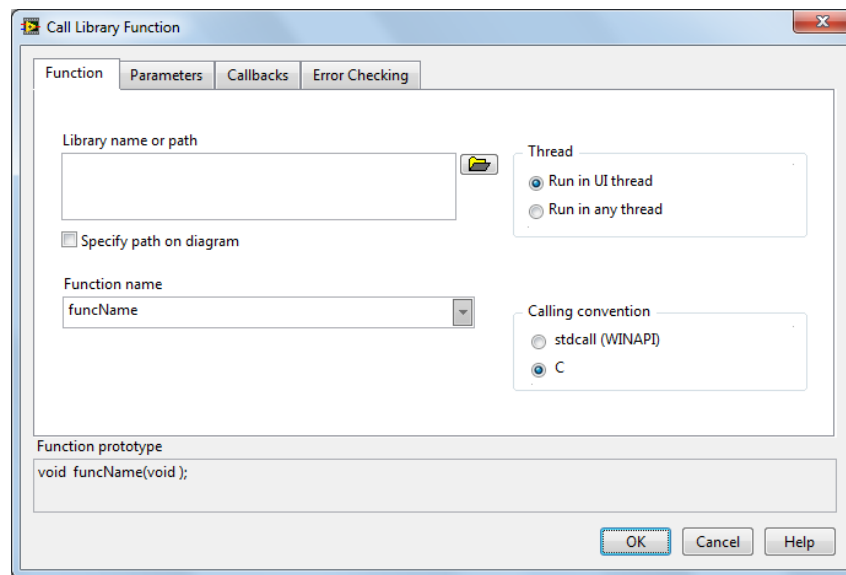


Figura 2.14 Configuración de parámetros para las llamadas a funciones de biblioteca.

2.3.2 Detalles de un CIN

El CIN es un método general para ejecutar código C a partir de LabVIEW. En algunos casos se tiene mejor rendimiento usando CINs, porque la estructura de datos de la CIN está en el mismo formato que se almacena en LabVIEW [26].

2.3.3 Biblioteca de Enlace Dinámico para un HID de Atmel

Para que un dispositivo USB sea reconocido por una computadora se requiere de una biblioteca de enlace dinámico (DLL, *Dynamic Link Library*). Una DLL es un archivo ejecutable que permite a los programas compartir código y otros recursos necesarios para realizar determinadas tareas. Microsoft Windows proporciona DLLs con funciones y recursos para que los programas basados en Windows funcionen adecuadamente [9].

Las aplicaciones de Dispositivos de Interfaz Humana (HID, *Human Interface Device*) usan diferentes tipos de reportes (in/out) para enviar y recibir datos. La longitud de cada reporte es asignada en el Firmware y automáticamente detectada por la DLL después de las configuraciones del Firmware. En la Figura 2.15 se muestra un diagrama de la DLL proporcionada por Atmel, la cual permite comunicarse sólo con un dispositivo HID a la vez. Varios dispositivos no pueden ser manejados al mismo tiempo usando la misma DLL [25, 4].

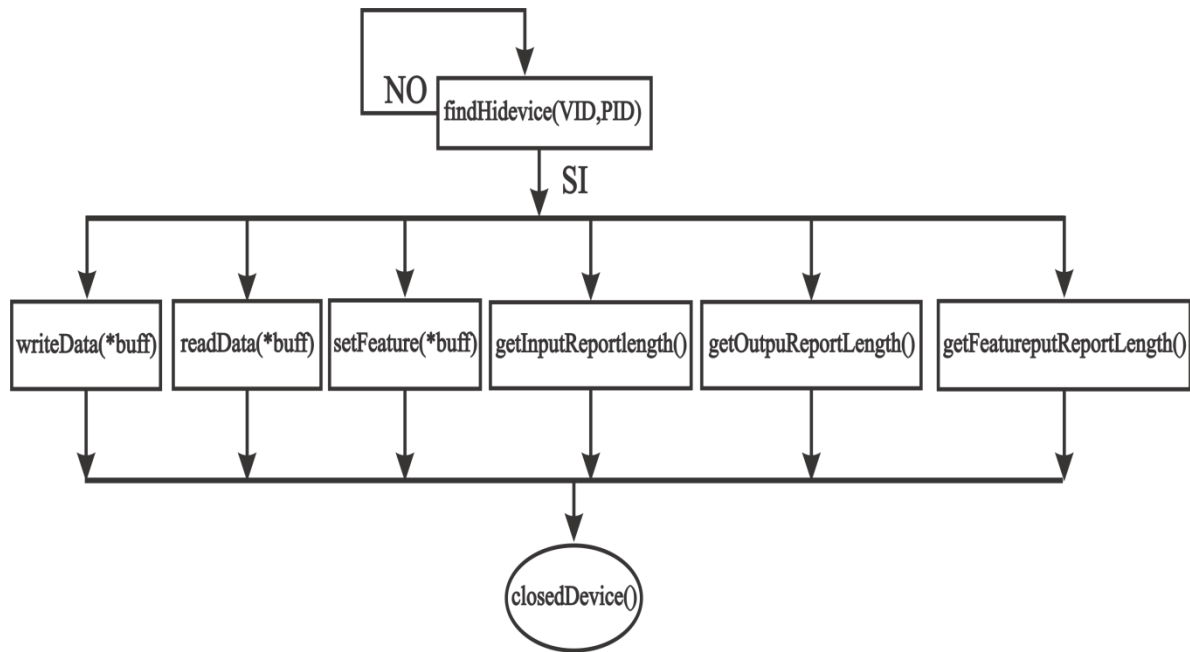


Figura 2.15 Funciones DLL [25].

El papel de cada una de las funciones es el siguiente [25]:

- La función *findHidDevice* permite buscar unidades genéricas HID, usando el identificador del fabricante VID y el identificador del producto PID, abre un manejador si la unidad es identificada.
- La función *ClosedDevice* termina la comunicación con todas las unidades y los manejadores.
- La función *writeData* envía datos a la unidad. La longitud máxima del reporte de datos es proporcionado por la función *getOutputReportLength*. Si la longitud del paquete de datos excede el tamaño especificado por el firmware, el usuario tiene que enviar los datos en varios paquetes. Por el contrario, cuando la longitud de un paquete es menor a la especificada, la función completa con ceros las localidades vacías.
- La función *readData* lee los paquetes de datos enviados por la unidad.

- La función *setFeature* permite al usuario enviar un comando para controlar la unidad.
- La función *getOutputReportLength* permite al usuario obtener la longitud de los paquetes de datos de salida (paquetes de datos enviados de la PC a la unidad).
- La función *getInputReportLength* permite al usuario obtener la longitud de los paquetes de datos de entrada (paquetes de datos enviados de la unidad a la PC).
- La función *getFeatureReportLength* permite obtener la longitud del reporte de configuración (paquete de datos de control enviados de la PC a la unidad).

Capítulo 3 Definición del sistema

En este capítulo se describe el diseño del hardware de la tarjeta de adquisición de datos con interfaz USB y del software que se encargará de inicializar y atender los recursos de Entradas/Salidas Digitales, Contador de Eventos, Entradas Analógicas, Salidas Analógicas, Salidas con PWM y manejo del protocolo I2C del microcontrolador AT90USB646. También se explica el diseño del software en la PC que permitirá monitorear y atender los recursos de la tarjeta de adquisición de datos e interactuar con el usuario.

3.1 Metodología de desarrollo

Para desarrollar un sistema de instrumentación electrónica es necesario realizar su planificación bajo alguna metodología. En el presente trabajo se utiliza la metodología de desarrollo para sistemas embebidos.

Un sistema embebido es un sistema que tiene un software embebido y hardware, lo que lo convierte en un sistema dedicado para una aplicación o parte específica de una aplicación o un producto [20].

En la Figura 3.1 se muestra un diagrama que contiene las diferentes etapas en el desarrollo de un sistema embebido [2].

En la etapa de *Especificación del producto* se define de forma clara y precisa la función del sistema, así como las entradas y salidas reales de éste y su interfaz con el

operador. Consiste en una descripción de los requerimientos técnicos y funcionales del sistema que se va a implementar para asegurar un producto robusto.

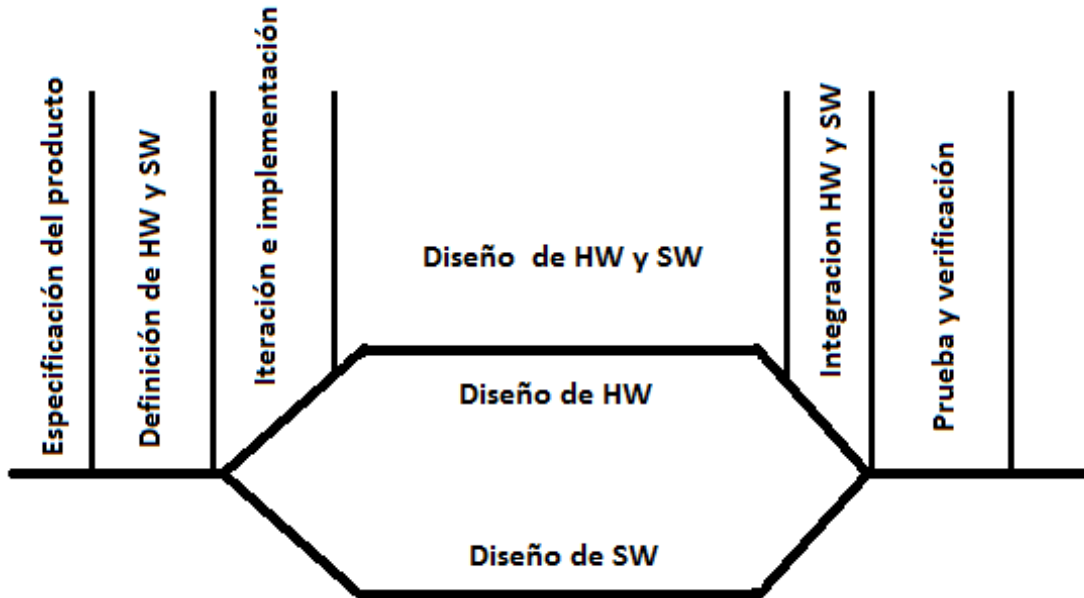


Figura 3.1 Diagrama de las fases para el diseño de un sistema embebido [2].

En la etapa de *Definición de HW y SW* se debe definir el hardware que se utilizará así como sus características principales. En la definición del software se especifican los requisitos, con base en la definición del hardware, en las especificaciones de ingeniería y en la definición general de requisitos del sistema.

En la etapa de *Iteración e implementación*, los componentes hardware y software son simulados o probados con la ayuda de herramientas especializadas, emuladores, para conocer con anticipación el comportamiento del sistema que se está implementando, pudiendo de esta forma anticipar y si fuera necesario corregir su funcionalidad.

En la etapa de *Diseño paralelo HW y SW* se diseñan detalladamente todos los componentes HW y SW especificados en los requerimientos, poniendo especial atención en mantener la coherencia entre los componentes en el producto final.

En la fase de *Integración de HW y SW* se deben integrar los componentes desarrollados para verificar que interaccionan correctamente entre sí.

En la etapa de *Prueba y verificación del producto* se realizan las pruebas del producto, si es preciso se realizan modificaciones y/o correcciones para adaptarlo a las necesidades del usuario.

3.2 Especificación del producto

El producto es una Tarjeta de Adquisición de Datos con interfaz USB y la interfaz gráfica de usuario para interactuar con la tarjeta. Las especificaciones iniciales del sistema son:

- Debe ser un sistema reconfigurable.
- Se comunicará a través del puerto USB.
- Debe contar con diversos recursos que permitan iniciar y terminar la comunicación.
- Se podrán inicializar los recursos que permitan el uso de Entradas/Salidas digitales, Entradas/Salidas analógicas, puerto I2C, salidas en PWM y contador de eventos.
- Se debe configurar y monitorear los diversos recursos de la tarjeta a través de la interfaz de una computadora desarrollada bajo la plataforma de LabVIEW.
- Dicha interfaz debe de cumplir con los siguientes requisitos:
 - Debe inicializar y finalizar la comunicación con la tarjeta.
 - Debe proporcionar al usuario las herramientas para poder configurar los diversos recursos de la tarjeta.
 - Contará con módulos que permitan al usuario conocer el estado de la conexión de la tarjeta.
 - Contará con un espacio de trabajo para que el usuario pueda desarrollar aplicaciones, utilizando los módulos de la TAD.
 - Debe presentar una interfaz amigable con el usuario final.

3.3 Definición de Hardware y Software

El diagrama a bloques mostrado en la Figura 3.2 describe las especificaciones del sistema. Este diagrama auxilia en la definición de los componentes de hardware y las especificaciones de los componentes de software que integran el sistema.

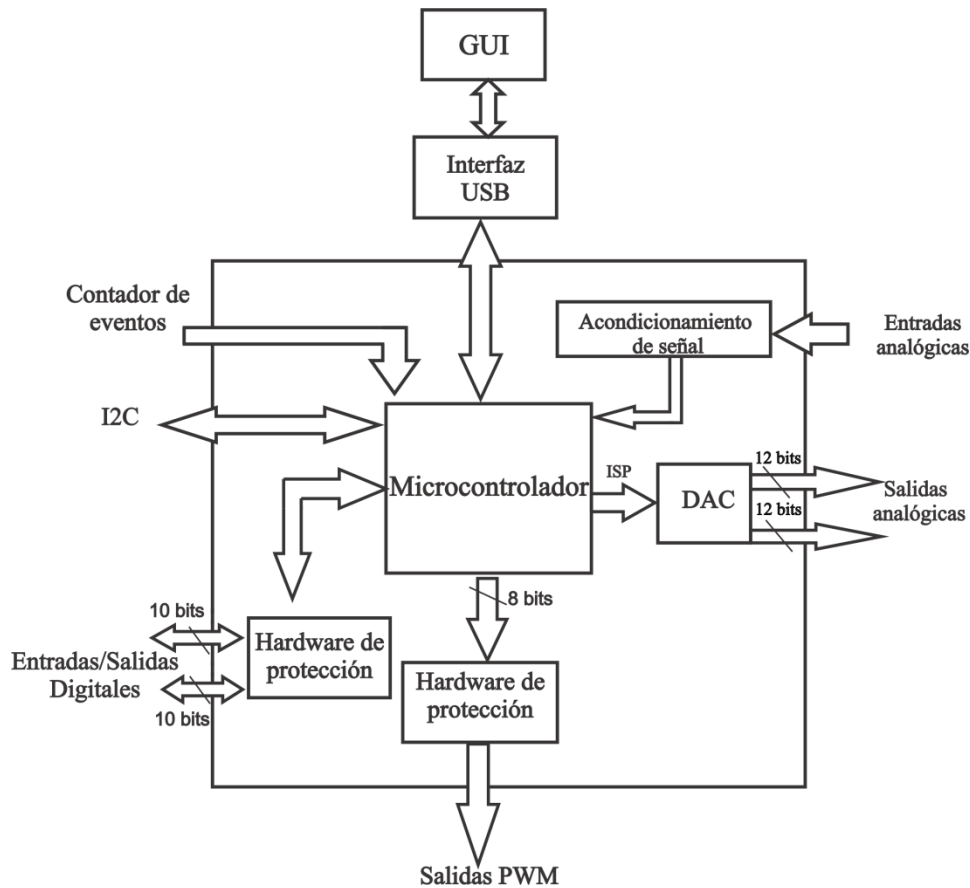


Figura 3.2 Diagrama a bloques de la especificación del sistema.

- Requerimientos de Hardware
 - 16 pines digitales que podrán configurarse como entrada o salida.
 - Puerto I²C.
 - 2 Salidas analógicas de 12 bits.
 - 2 contadores de eventos.
 - 8 entradas analógicas de 10 bits referenciadas.
 - 3 salidas PWM de 8 bits.
 - Puerto de comunicación USB.
- Requerimientos de Software
 - Módulos de configuración y atención de los recursos.
 - Interfaz Gráfica de Usuario (GUI, *Graphical User Interface*).

3.3.1 Definición de Hardware

Según las especificaciones del producto, un microcontrolador es la parte principal de la tarjeta y se encargará de atender los diferentes recursos, ya que cuenta con los elementos necesarios para la adquisición y generación de señales analógicas y digitales.

3.3.1.1 El Microcontrolador AT90USB646

El microcontrolador AT90USB646 es un circuito integrado fabricado con tecnología CMOS (*Complementary metal-oxide-semiconductor*) de bajo consumo, con núcleo AVR de 8 bits, arquitectura RISC y con un rendimiento cercano a 1 MIPS por MHz. A continuación se muestran algunos de los recursos con los que cuenta este microcontrolador [10].

- 64 KB de memoria de programa.
- 2 KB en EEPROM.
- 4 KB SRAM.
- 48 pines para I/O de propósito general.
- 32 registros de propósito general.
- 4 Temporizadores/Contadores con modos de comparación y generación de señales moduladas en ancho de pulso (PWM).
- Receptor / Transmisor Universal Sincrónico /Asincrónico (USART).
- Convertidor Analógico-Digital de 10 bits (*Analog Converter Digital*, ADC), con 8 canales.
- Perro guardián (*Watchdog Timer*) programable con oscilador interno.
- Puerto SPI.
- Controlador USB de alta velocidad (12 Mbps).
- Interfaz JTAG.

Comercialmente el microcontrolador AT90USB646 se encuentra en un encapsulado SMD de 64 pines. En la Figura 3.3 se muestra el aspecto externo del dispositivo, puede verse que cuenta con 6 puertos de 8 bits [10].

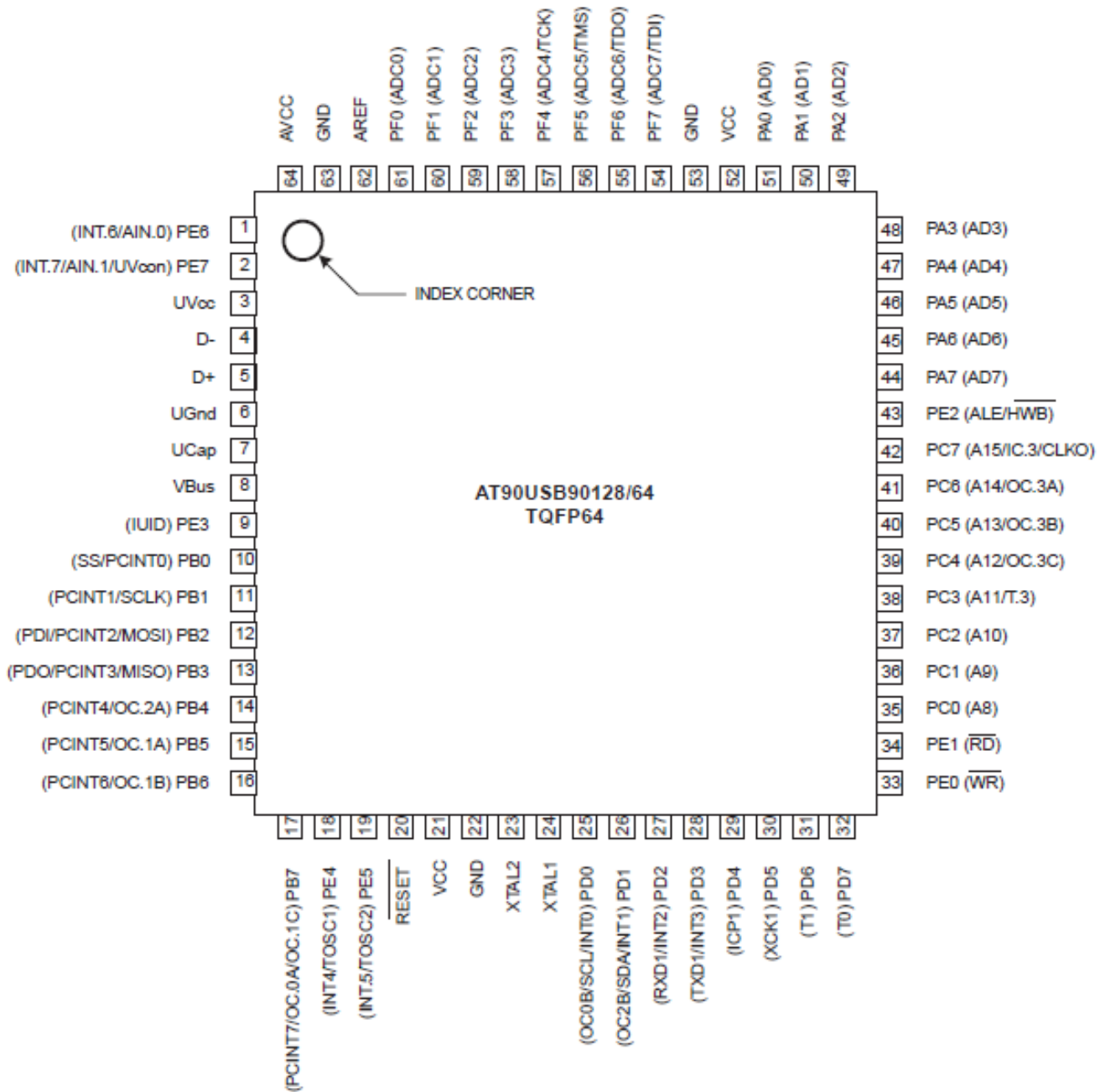


Figura 3.3 Aspecto externo del microcontrolador AT90USB646 [10].

En la Tabla 3.1 se muestra la distribución de los puertos del microcontrolador para cada recurso de la TAD.

Tabla 3.1 Asignación de los puertos del microcontrolador a los recursos de la TAD.

Puerto en el microcontrolador	Uso en la tarjeta
Puerto A y E	Entrada/Salidas Digitales
Puerto B	Salidas analógicas
Puerto C	Salidas PWM y Control de buffer
Puerto D	Comunicación I2C y contador de eventos
Puerto F	Entradas analógicas

En las siguientes secciones se describen los recursos del microcontrolador que se emplearon para cubrir con las especificaciones de la tarjeta bajo desarrollo.

3.3.1.1.1 Controlador USB

El Microcontrolador AT90USB646 cuenta con un controlador USB que proporciona el hardware para interconectar un enlace USB a un flujo de datos almacenados en una memoria de puerto doble (DPRAM).

Este controlador USB requiere una frecuencia de reloj de 48 MHz, la cual es obtenida de un Lazo de Seguimiento de Fase (PLL, *Phase Locked Loop*) interno. El PLL genera una señal de reloj con una frecuencia interna para la interfaz USB a partir de una señal externa con una frecuencia inferior, generada mediante un cristal de cuarzo, en la Figura 3.4 se muestra un diagrama a bloques del controlador USB [10].

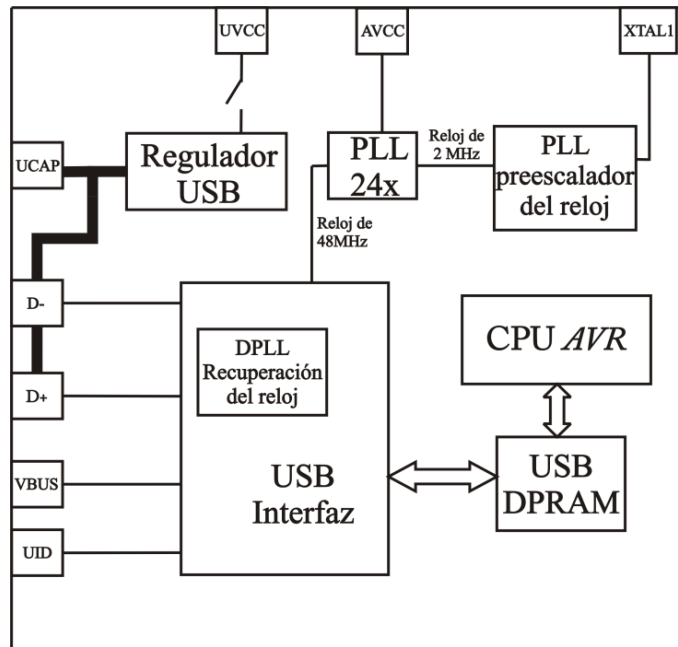


Figura 3.4 Diagrama a bloques del Controlador USB [10].

El microcontrolador AT90USB646 se puede configurar para operar como Unidad USB, como Servidor (Host) o como Unidad USB de doble función, según sea la aplicación a desarrollar. Aunque en realidad sería un servidor reducido, porque su puerto USB está diseñado para manejar unidades con una conexión punto a punto. Esta configuración de servidor reducido soporta aplicaciones de dispositivos con VID y PID (VID: identificador USB del fabricante, PID: identificador USB del producto) [10].

El modo de operación Dispositivo USB de doble función permite apoyar los modos Unidad USB y Servidor reducido USB, se puede cambiar entre uno y otro mediante un pin identificador. Si el pin se conecta a tierra el microcontrolador se configura como servidor USB reducido, pero si no está conectado, opera en el modo de unidad.

En la Figura 3.5 se muestra la conexión física para cuando el microcontrolador AT90USB646 es configurado para que opere como Unidad USB y sea alimentado con un nivel de voltaje de 5 Volts proporcionados por el bus [10].

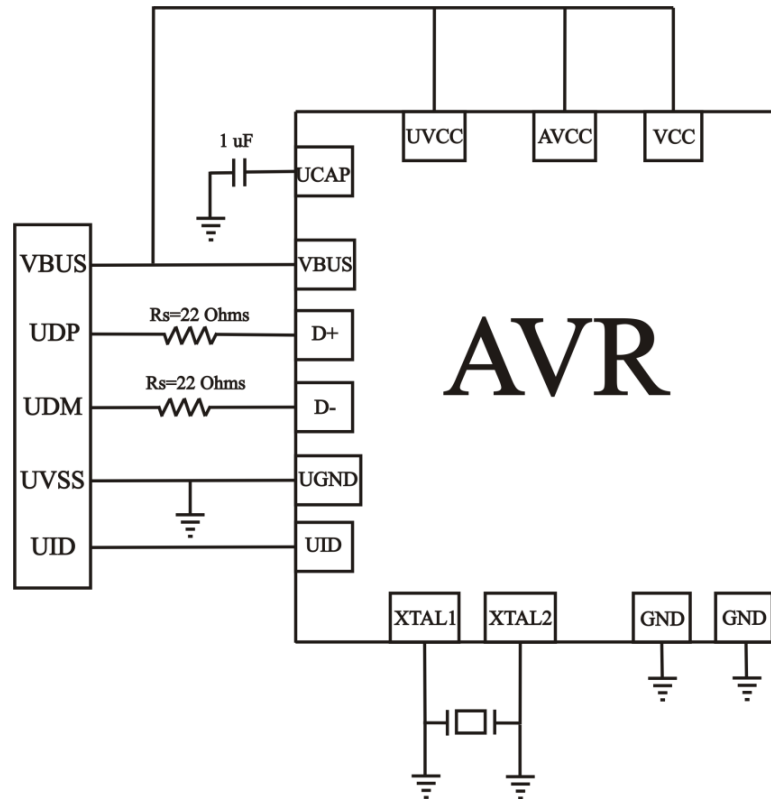


Figura 3.5 Alimentación del BUS para Unidades USB [10].

El controlador USB incluido en el AT90USB646 es fundamental para el desarrollo de la TAD, dado que esta interfaz se emplea para llevar a cabo la transferencia de datos entre el microcontrolador y una PC. Es a través del puerto USB que se configuran y manipulan los diversos recursos de la TAD.

3.3.1.1.2 Puertos de Entrada/Salida digitales

Todos los puertos de los microcontroladores AVR tienen la función de leer o escribir cuando se utilizan como puertos digitales de Entrada/Salida. La dirección de un pin de un puerto se puede cambiar sin tener que cambiar al resto. Lo mismo se aplica cuando se cambia el valor de la unidad (si está configurado como salida) o la activación / desactivación de los resistores *pull-up* (si está configurado como entrada) [21].

Todos los pines del puerto tienen resistores de *pull-up* cuya conexión se puede configurar de forma individual, también cuentan con diodos de protección tanto a VCC y tierra como se indica en la Figura 3.6.

A cada puerto se le asignan tres localidades de memoria de E/S, PORTx para el registro de datos, DDRx para el registro de dirección del puerto y PINx para los pines de entrada del puerto [21].

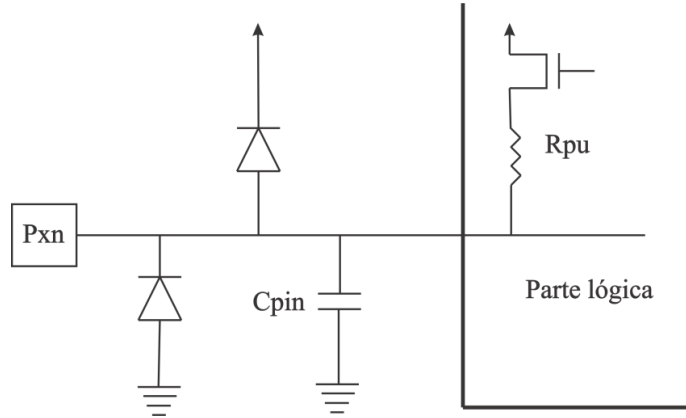


Figura 3.6 Esquemático para un pin de Entrada/Salida [10].

Los 16 pines digitales para entrada o salida en la TAD son manejados directamente por los puertos de entrada/salida que permitirán la lectura y escritura de niveles lógicos de voltaje. Estos puertos están protegidos por un circuito externo al microcontrolador para evitar que corrientes parásitas causen algún desperfecto.

3.3.1.1.3 *Convertidor Analógico Digital*

El microcontrolador AT90USB646 cuenta con un convertidor analógico a digital de aproximaciones sucesivas de 10 bits, el cual está conectado a un multiplexor analógico que permite seleccionar 1 de 8 canales externos (ADC0, ADC1,..., ADC7). Las principales características con las que cuenta este ADC son [21]:

- 7 canales de entrada diferencial
- Tiempos de conversión de 65 a 260 us
- Hasta 15 kmuestras/segundo cuando se emplea la máxima resolución
- Ajuste opcional del resultado de la conversión a la izquierda para su lectura con resolución de 8 bits
- Rango de voltaje de entrada entre 0 - Vcc
- Interrupción del ADC por Conversión Completa
- Supresor de ruido en la espera

Por defecto, el circuito de aproximaciones sucesivas requiere una frecuencia de reloj de entrada entre 50 kHz y 200 kHz para obtener la máxima resolución. Si se requiere de una resolución inferior a 10 bits, la frecuencia de reloj de entrada para el ADC puede ser superior a 200 kHz para obtener una frecuencia de muestreo más alta [21].

El módulo ADC contiene un pre-escalador de 7 bits que genera su frecuencia de trabajo a partir de la frecuencia de operación del microcontrolador, éste se muestra en la Figura 3.7.

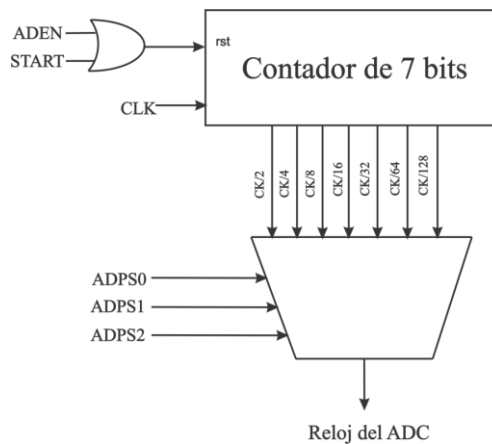


Figura 3.7 Pre-escalador del ADC [10].

Debido a que el ADC lleva a cabo las conversiones por aproximaciones sucesivas, requiere de un voltaje de referencia (V_{ref}), el cual puede ser proporcionado por diferentes fuentes tal como se muestra en la Figura 3.8. La selección de V_{ref} se realiza con los bits RefS1 y RefS2, como se muestra en la

Tabla 3.2, el voltaje de referencia determina el rango de conversión del ADC, si el voltaje analógico excede a V_{ref} será codificado como 0x3FF (1023) [21].

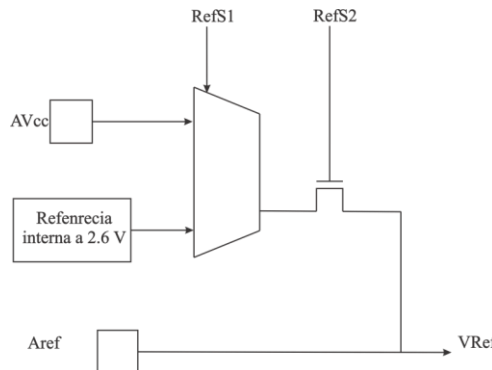


Figura 3.8 Hardware para el voltaje de referencia [10].

Tabla 3.2 Opciones para voltaje de referencia del ADC [10].

Refs1	Refs2	Voltaje de referencia
0	0	Voltaje externo en Aref, referencia interna apagada
0	1	Voltaje externo en AVcc
1	0	Reservado
1	1	Voltaje interno de 2.6V

El ADC permite a la TAD leer señales analógicas provenientes de algún sensor para digitalizarlas y de esta manera poder enviarlas a través del puerto USB. Una vez que los datos son recibidos por la PC, la interfaz gráfica se encarga de mostrarlos para que el usuario pueda llevar a cabo alguna acción de control.

El voltaje de referencia para el ADC es de 5 V proporcionado por el bus USB. Debido a esto, para garantizar la correcta funcionalidad del recurso, fue necesario acondicionar las señales de entrada mediante circuitos externos, los cuales se explican en la sección 2.3.1.2.

3.3.1.1.4 Temporizadores

El microcontrolador AT90USB646 cuenta con 4 temporizadores. Los temporizadores 0 y 2 son de 8 bits, mientras que el 1 y 3 son de 16 bits. Los eventos que se pueden generar con los temporizadores son: desbordamiento, coincidencia por comparación y captura de entrada.

El evento de desbordamiento ocurre cuando el valor de alguno de los temporizadores (TCNTn) alcanza su valor máximo (MAXVAL) y se reinicia con 0. Esto provoca que una bandera (TOVn) sea puesta en alto. El valor máximo de los temporizadores queda determinado por la siguiente expresión [10].

$$MAXVAL = 2^{Tamaño(TCNT)} - 1$$

Con respecto a los eventos de coincidencia por comparación, en el hardware se dedica un registro para comparaciones continuas (OCRn, *Output Compare Register*), en este registro se puede cargar un valor entre 0 y MAXVAL. En cada ciclo de reloj se compara al registro del temporizador con el registro de comparación, una coincidencia produce un evento que se indica con la puesta en alto de una bandera (OCFn, *Output Compare Flag*) [10].

Para el evento de captura de entrada, se tiene una terminal dedicada a capturar eventos externos (ICP, *Input Capture Pin*), un cambio en esta terminal provocará que el registro del temporizador sea leído y almacenado en el registro de captura de entrada (ICR, *Input Capture Register*). El tipo de transición en ICP que provoca las capturas es configurable, puede ser un flanco de subida o un flanco bajada.

Existen 3 formas para detectar los eventos producidos por los temporizadores y actuar ante ellos [10]:

- Sondeo (*polling*), el programa evalúa frecuentemente el estado de las banderas.
- Uso de interrupciones, se deben de activar en el registro de enmascaramiento de interrupciones (TIMSK, *Timer Interrupt Mask*).
- Respuesta automática, se modifican las terminales de comparación de salida automáticamente, en respuesta a coincidencias por comparación (OC, *Output Compare*).

La TAD cuenta con 2 contadores de eventos implementados con el temporizador 3 y 0 de 16 y 8 bits respectivamente. Con el temporizador 3 se puede tener una cuenta máxima de 65 536 eventos y con el temporizador 0 se pueden contar hasta 1023 eventos [10].

Para poder usar el temporizador como contador de eventos, el registro del temporizador se configura para que se incremente en cada flanco de subida o bajada de una señal externa, la cual indicará que un evento ha ocurrido y por lo tanto se debe incrementar el conteo. Esta señal se conecta a un pin específico del microcontrolador, el cual corresponde al temporizador deseado.

3.3.1.1.5 Modulación por Ancho de Pulso

La Modulación por Ancho de Pulso (PWM, *Pulse Width Modulation*) es una técnica para generar señales analógicas en alguna salida de un sistema digital. La base de PWM es la variación del ciclo de trabajo (*duty cycle*) de una señal cuadrada, en la Figura 3.9 se muestra una señal cuadrada con la definición de ciclo útil.

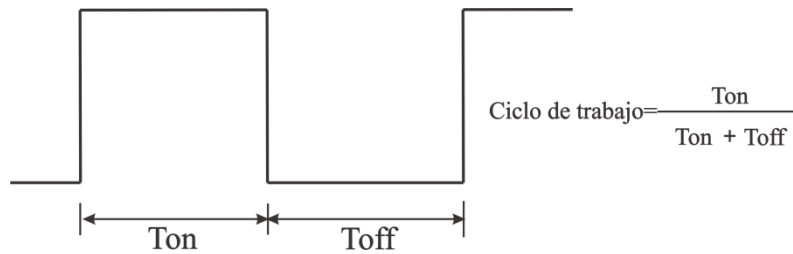


Figura 3.9 Definición del ciclo de trabajo.

Las señales PWM se generan con los recursos de los temporizadores. Los temporizadores del microcontrolador AT90USB646 pueden generar señales PWM en 3 diferentes modos [10]:

- PWM rápido (*fast PWM*).
- PWM con fase correcta (*phase correct PWM*).
- PWM con fase y frecuencia correcta (*phase and frequency correct PWM*).

La TAD cuenta con 3 salidas PWM de 16 bits. Para ello se hace uso del temporizador 1 y se configura para que trabaje en el modo PWM con fase correcta. En este modo el temporizador cuenta de manera ascendente de 0 a su valor máximo para después contar de manera descendente, la salida se conmuta para generar el ciclo de trabajo en la coincidencia por comparación con el registro comparación de salida (OCRx, *Output Compare Register*) [10].

La frecuencia de la señal generada está dada por $f_{pwm} = \frac{f_{clk}}{2(MAX+1)}$.

Las 3 salidas PWM tendrán un buffer de protección para evitar que alguna corriente parasita cause un desperfecto en el sistema.

3.3.1.1.6 Comunicación I2C

La interfaz Serial a Dos Hilos (TWI, *Two Wire Serial Interface*) es un recurso del microcontrolador AT90USB646 que le permite comunicarse por medio de un bus direccional de 2 líneas, una para reloj (SCL) y la otra para datos (SDA). La interfaz TWI es compatible con el estándar I2C desarrollada por Philips Semiconductors (actualmente NXP). En la Figura 3.10 se muestra la forma en que los diferentes dispositivos se conectan, como hardware externo únicamente se requieren dos resistores conectados a V_{cc} (*pull-up*). Mediante este protocolo se pueden conectar hasta 128 dispositivos diferentes [21].

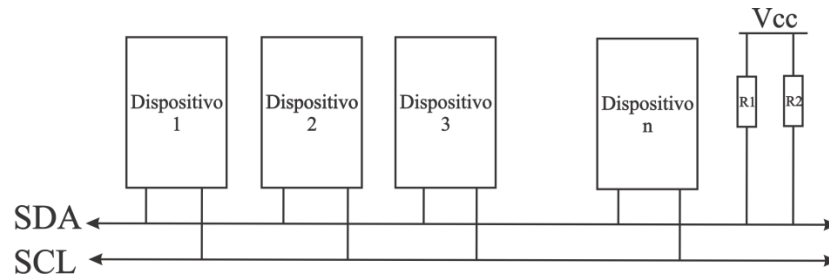


Figura 3.10 Bus de interconexión TWI [10].

El protocolo TWI maneja un esquema Maestro-Esclavo, sin embargo, cualquier dispositivo puede transmitir en el bus. Cada bit transferido en la línea SDA deberá acompañarse de un pulso en la línea SCL.

El nivel del dato debe de estar estable cuando la línea de reloj está en alto. El nodo Maestro es el que se encarga de iniciar y terminar una transmisión, por lo tanto, es éste quien genera las condiciones de inicio y paro, estas condiciones son señalizadas cambiando el nivel en la línea SDA cuando SCL está en alto. Entre estas condiciones el bus se considera ocupado y ningún otro nodo Maestro podrá tomar control sobre él [21].

La interfaz TWI disponible en el AT90USB646 hace posible la disposición de un puerto con el protocolo I2C en la TAD, para que sirva como herramienta y fácilmente se puedan evaluar sensores, memorias u otros dispositivos que utilicen esta interfaz.

3.3.1.1.7 Comunicación serial por SPI

La Interfaz Serial Periférica (SPI, *Serial Peripheral Interface*) establece un protocolo estándar de comunicaciones, usado para transferir paquetes de información de 8 bits entre circuitos integrados. Permite una transferencia síncrona de datos a muy alta velocidad.

En la conexión de 2 dispositivos siempre ocurre que uno funciona como nodo Maestro y otro como nodo Esclavo. El nodo Maestro es quien determina cuando se enviarán los datos y quien genera la señal de reloj. En la Figura 3.11 se muestra una conexión entre 2 dispositivos vía SPI [10].

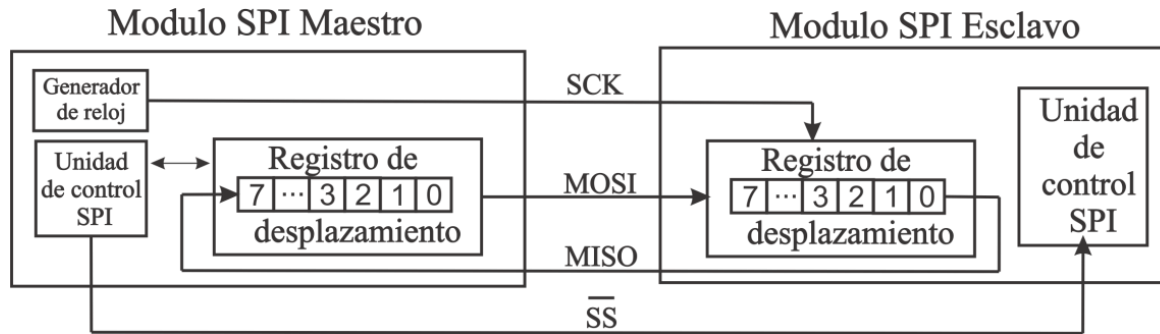


Figura 3.11 Conexión entre 2 dispositivos vía SPI [21].

- *MOSI*: Señal de salida del nodo Maestro y entrada del nodo Esclavo.
- *MISO*: Señal de entrada al nodo Maestro y salida en el nodo Esclavo. Proporciona el mecanismo para que el nodo Esclavo pueda dar respuesta al nodo Maestro.
- *SCK*: Es la señal de reloj para sincronizar la transferencia.
- *SS*: Señal de selección de un nodo Esclavo, esta señal es útil para el manejo de múltiples nodos Esclavos.

El envío y recepción se realizan de manera simultánea, mientras el nodo Maestro envía datos por MOSI recibe una respuesta del nodo Esclavo por MISO. Sincronizando el envío y recepción de información con la señal de reloj, en el mismo ciclo de reloj se transmite y recibe 1 bit.

La interfaz SPI del AT90USB646 se utiliza para generar señales analógicas, para ello se hace uso del circuito integrado MCP4922, el cual es un DAC que maneja una comunicación vía SPI con el microcontrolador para recibir los datos de forma digital y producir la señal analógica que le corresponda.

3.3.1.2 Acondicionamiento de Señal

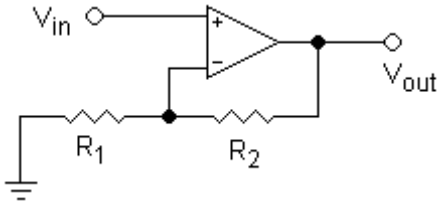
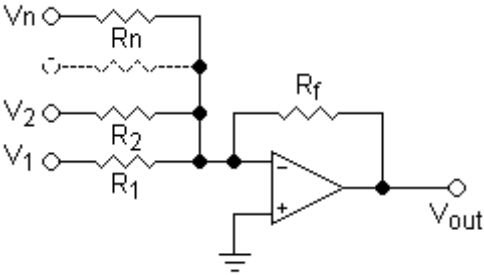
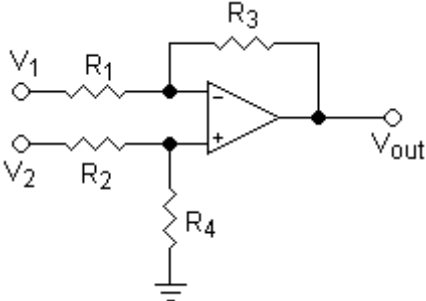
Para poder acoplar dos sistemas electrónicos, generalmente es necesario el uso de un circuito que permita acondicionar las señales de tal forma que ambos sistemas puedan interpretarlas correctamente. A estos circuitos se les conoce como circuitos para el acondicionamiento de señales.

Debido a que la TAD cuenta con señales de entrada y salidas analógicas que serán usadas o proporcionadas por otro circuito, es necesario el uso de una etapa que permita acondicionar los niveles de tensión.

La TAD utilizará el voltaje de referencia de 5 V proporcionado por el bus USB para el módulo del ADC. En consecuencia, cuando la señal de entrada tenga una amplitud de -10 V a 10 V o de 0 a 10 V, se acondicionará esta señal mediante el uso de amplificadores operacionales para que el módulo ADC del microcontrolador pueda interpretarlas correctamente.

Un amplificador operacional es un circuito integrado que permite la manipulación de la amplitud de las señales analógicas. En la Tabla 3.3 se muestran las configuraciones típicas del amplificador operacional y la ecuación matemática que lo describe [8].

Tabla 3.3 Configuraciones típicas del Amplificador Operacional.

Nombre	Circuito	Ecuación Matemática
Amplificador no inversor		$V_{out} = V_{in} \left(1 + \frac{R_2}{R_1} \right)$
Sumador inversor		$V_{out} = -R_f \left(\frac{V_1}{R_1} + \frac{V_2}{R_2} + \frac{V_3}{R_3} + \dots + \frac{V_n}{R_n} \right)$
Restador inversor		$V_{out} = V_2 \left(\frac{(R_3 + R_1)R_4}{(R_4 + R_2)R_1} \right) - V_1 \left(\frac{R_3}{R_1} \right)$

3.3.1.3 Convertido Digital-Analógico

Para poder llevar a cabo la conversión de datos digitales a analógicos es necesario el uso de un circuito integrado externo ya que el microcontrolador AT90USB646 no cuenta con recursos que permitan realizar esta tarea.

Se utiliza el circuito integrado MCP4922 de la firma Microchip, debido a que cuenta con un bus de comunicaciones SPI, permitiendo así reducir el número de pines del microcontrolador. La resolución del convertidor digital a analógico es de 12 bits y tiene 2 canales de salida. En la Figura 3.12 se muestra el encapsulado del circuito integrado MCP4922 [12].

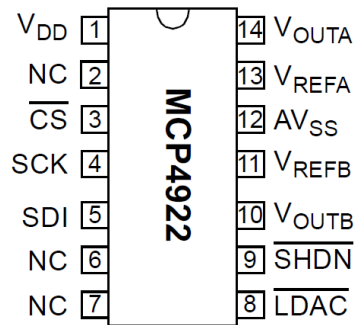


Figura 3.12 Circuito integrado MCP4922 [12].

3.3.2 Definición del Software

El software a desarrollar es, por una parte, la aplicación implementada en el microcontrolador, que se encarga de generar y procesar las señales que el usuario requiere y, por otra, la interfaz gráfica de usuario desarrollada en LabVIEW para una PC, que permita al usuario configurar los diversos recursos del microcontrolador así como la visualización de las señales adquiridas.

3.3.2.1 Software del Microcontrolador

El software que está implementado en el microcontrolador está dividido en el módulo de comunicación y el módulo de la aplicación.

El módulo de comunicación se encarga de configurar al controlador USB del microcontrolador, para que se pueda establecer la comunicación con una PC. La velocidad de la comunicación es de 12 Mbps, el micro es configurado para que opere como un dispositivo

de interfaz humana y de esta forma hace uso del controlador proporcionado por ATMEL para que el Sistema Operativo de la PC reconozca al microcontrolador.

El módulo de la aplicación se refiere al software que se encarga de decodificar y generar las tramas recibidas y enviadas por una PC.

3.3.2.2 Interfaz Gráfica de Usuario

El software para la interfaz con el usuario se implementó en una PC en la plataforma de desarrollo de LabVIEW. Esta parte del sistema se encarga de mostrar al usuario los diversos recursos que puede emplear en la TAD.

Una vez que el usuario gráficamente configura un recurso, internamente la aplicación genera la trama que es enviada por el bus USB hacia la tarjeta. Cuando se ha configurado un recurso que permite la entrada de datos desde la tarjeta, la aplicación se encarga de decodificar la trama enviada y muestra los resultados al usuario de forma gráfica o numérica, según sea el caso.

Los módulos que integran la interfaz gráfica de usuario son los siguientes:

- Módulo para conocer el estado de conexión de la tarjeta
- Módulo para configurar la conexión y desconexión de la tarjeta
- Módulo para configurar los recursos del ADC, el DAC, la interfaz I2C, las Entradas/Salidas digitales, el contador de eventos y las salidas PWM.

En la Figura 3.13 se muestra el diagrama de la arquitectura del software.

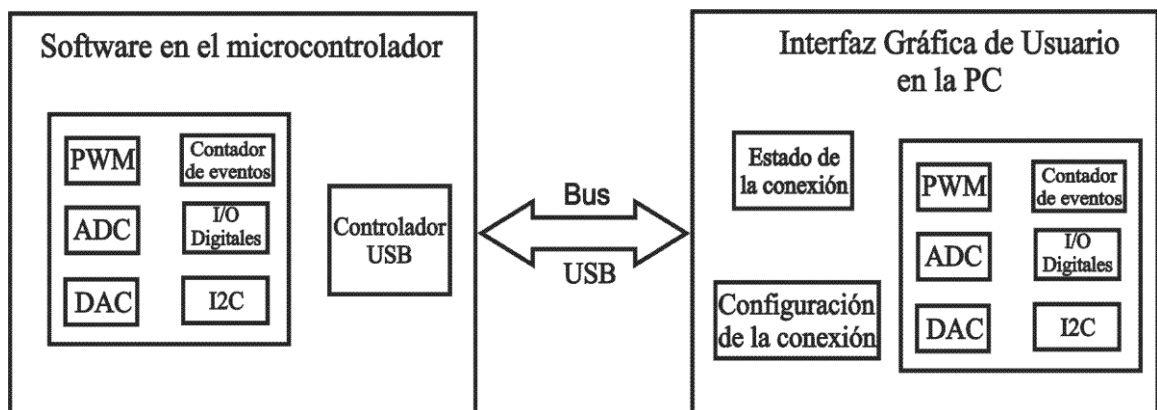


Figura 3.13 Diagrama de la arquitectura del software.

Capítulo 4 Diseño del sistema

4.1 Diseño del Hardware

El microcontrolador AT90USB646 es la parte principal de la TAD e internamente cuenta con el hardware necesario para configurar los recursos de Entradas/Salidas Digitales, generación de señales PWM, Contador de Eventos, ADC y generación de señales mediante el protocolo de comunicación I2C y SPI. Estos módulos no requieren de algún circuito externo para su funcionamiento.

Los puertos del microcontrolador que proporcionan las E/S Digitales (Puerto A y E) y Salidas PWM (Puerto C), se conectan a un circuito de protección que permite asegurar su correcto funcionamiento en caso de que exista una mala conexión y para evitar que alguna corriente parásita cause algún desperfecto. El circuito de protección que se usa es el 74LS245, el cual permite un flujo de datos bidireccional para el caso de las E/S Digitales controlado por el puerto C. Por la estructura del 74LS245, todo el puerto debe de ser configurado como entrada o salida.

Para los contadores de eventos se usan los temporizadores 0 y 1 del AT90USB646 y los recursos se configuran con señales de reloj externas para poder detectar los eventos. En la Figura 4.1 se muestra la conexión de los buffers de protección con el microcontrolador y las terminales de la TAD.

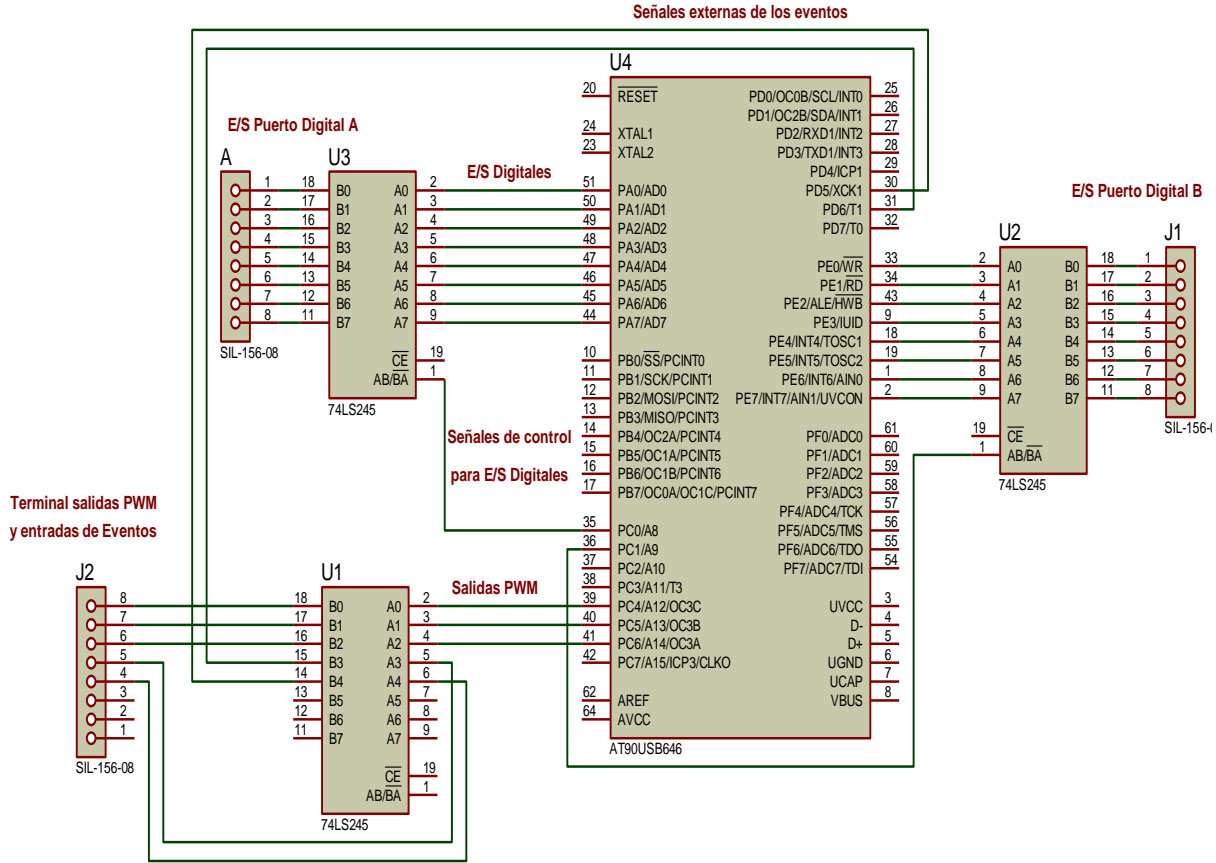


Figura 4.1 Conexión de los puertos Digitales y salidas PWM.

Las entradas para el ADC están en el puerto F del microcontrolador y provienen de circuitos que permiten acondicionar la señal para que el hardware del ADC pueda interpretarlas correctamente. El ADC no cuenta con hardware de protección contra voltajes fuera del rango de operación, por lo que el usuario sólo debe usar señales con estas características.

La TAD cuenta con 8 entradas para señales analógicas que hacen uso del ADC, 3 de esos 8 canales aceptan señales de entrada de 0 a 5 V de amplitud, por lo que no requieren una etapa de acondicionamiento. Otros tres canales son para señales en el rango de 0 a 10 V de amplitud y para este caso es necesario el uso de amplificadores operacionales configurados en modo *amplificador inversor*. En la Figura 4.2 se muestra el circuito que permite atenuar una señal de entrada de 0 a 10V a una señal de salida de 0 a 5V y de esta forma asegurar que el micro pueda leerla correctamente sin que haya pérdidas de información. La configuración del primer operacional atenúa la señal de entrada a la mitad de su valor en amplitud y la invierte,

posteriormente el segundo operacional sólo la invierte y de esta forma se asegura que en la salida habrá una señal en el rango de 0 a 5 V de amplitud.

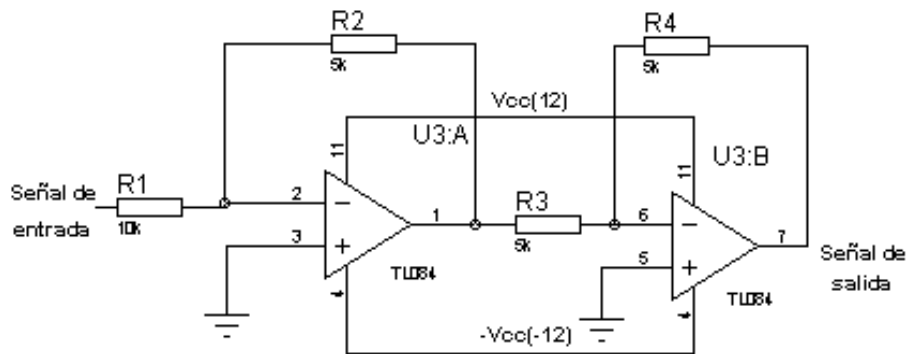


Figura 4.2 Circuito acondicionador de señales para una entrada de 0 a 10V.

Los 2 canales analógicos restantes en la TAD permiten entradas de -10 a 10 V de amplitud. Para poder acondicionar la señal se hace uso del circuito mostrado en la Figura 4.3. El OPAM1 está configurado como amplificador inversor, atenúa la señal de entrada a cuatro veces su valor y la invierte. El OPAM2 duplica el valor de la señal de entrada y le suma 5 V. Por último el OPAM3 atenúa la señal de entrada a la mitad de su valor y la invierte, así la señal de salida tendrá un valor en el rango de 0 a 5 V.

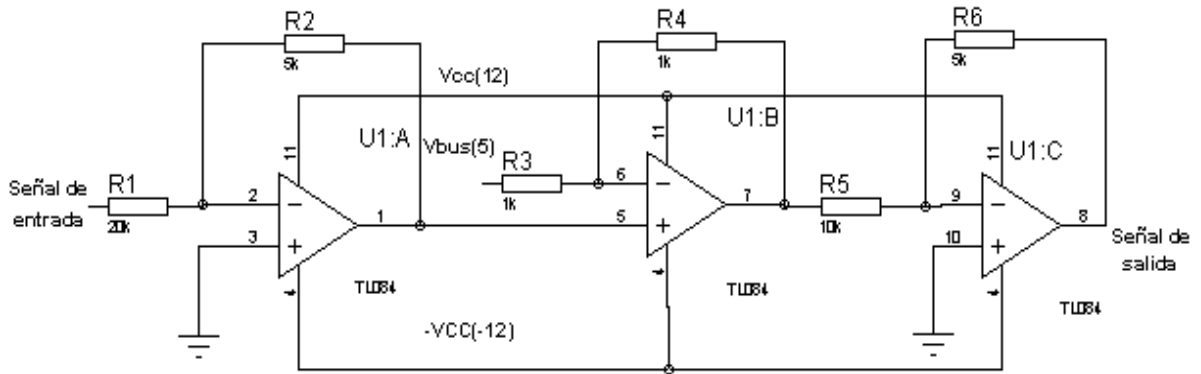


Figura 4.3 Circuito acondicionador de señales de entrada de -10 a 10V.

Debido a que la TAD solo usa el voltaje del bus USB, fue necesario el uso del integrado AM1D-N-Z para elevar los niveles de tensión y que los operacionales trabajen con voltajes de 15 y -15V.

Para la implementación de los Convertidores Digital a Analógico en la TAD se hace uso del circuito integrado MCP4922, dado que el microcontrolador no cuenta con los recursos internos que permitan llevar a cabo esta tarea. La comunicación entre los dispositivos se

realiza por medio del protocolo de comunicación SPI. En la Figura 4.4 se muestra la conexión del microcontrolador con el circuito integrado MCP4922.

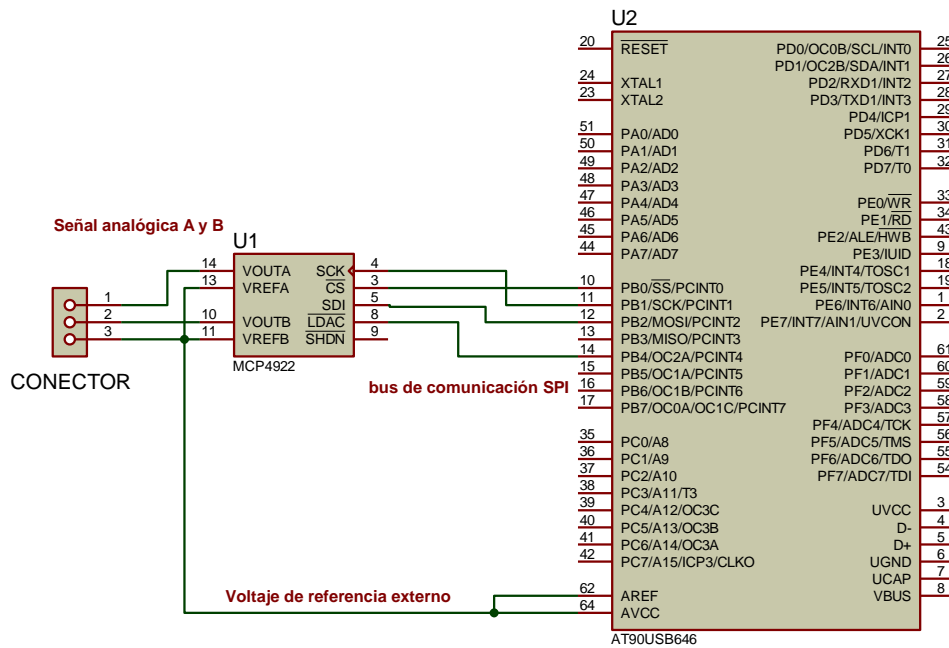


Figura 4.4 Diseño del Convertidor Analógico Digital.

La interfaz del puerto para la comunicación TWI no requiere de Hardware adicional por lo que los dispositivos TWI que se deseen controlar con la TAD, se conectan directamente con esta.

4.2 Diseño del Software

De acuerdo con la Figura 3.13, el software se divide en la interfaz gráfica de usuario en la PC y el software del microcontrolador. En esta sección se describe el diseño de estas partes.

El diseño del software referente a la GUI se desarrolla en el entorno de LabVIEW de National Instruments y para el software del microcontrolador se utiliza al AVR Studio versión 4.14, que es una plataforma de desarrollo proporcionada gratuitamente por Atmel Corporation.

4.2.1 Interfaz Gráfica de Usuario en la PC

EL software de la GUI en la PC comprende la configuración de la conexión, el estado de la conexión y los módulos para la configuración y monitoreo de los recursos.

Para que la TAD pueda comunicarse con la PC, es necesario que el usuario configure la comunicación. En la Figura 4.5 se muestra el módulo realizado para establecer la

comunicación, en el que el usuario debe especificar la ruta de la DLL, y en la Tabla 4.1 se describen sus terminales.

El módulo se comunica con la DLL para leer el VID y el PID proporcionados por la TAD, si estos parámetros coinciden con los especificados en el módulo, en la salida se tiene un estado lógico *Verdadero*, que indica que se reconoce a la TAD y se estableció correctamente la comunicación, en caso contrario resulta en un *Falso* indicando que la comunicación no se estableció.



Figura 4.5 Módulo para establecer la comunicación.

Tabla 4.1 Terminales del módulo para inicializar la comunicación.

Nombre	Flujo	Descripción
Ruta	Entrada	Se establece la ruta actual de la DLL
Estado	Salida	Indica si se ha inicializado la comunicación con la TAD

El usuario puede saber en algún determinado momento si se está llevando a cabo la comunicación entre la TAD y la PC por medio de un módulo que se encarga de monitorear el estado de la conexión. En la Figura 4.6 se muestra el módulo para probar la comunicación y en la Tabla 4.2 se describen sus terminales.

Si el interruptor está en la posición de Enviar, el módulo genera la trama Escribe-Led y la envía a la TAD a través del puerto USB con la instrucción para prender o apagar un LED. La descripción de ésta y las demás tramas que se requieren para la comunicación de la aplicación de la TAD con la GUI se describen en el apéndice A.

Si el interruptor está en la posición Recibir, el módulo lee la trama Lectura-Led proveniente de la TAD y decodifica la instrucción para que se realice la misma operación en la interfaz gráfica. De esta forma se tiene una comunicación bidireccional y se puede comprobar la comunicación de la TAD a la PC y viceversa

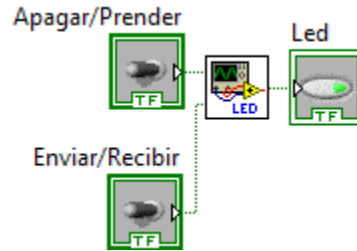


Figura 4.6 Módulo para probar la comunicación.

Tabla 4.2 Terminales del módulo para probar la comunicación.

Nombre	Flujo	Descripción
Apagar/Prender	Entrada	'1' lógico para prender el LED y '0' lógico apagarlo
Enviar/Recibir	Entrada	'1' lógico para enviar la instrucción a la TAD y el '0' indica que se leerá la instrucción de la TAD
LED	Salida	Muestra la configuración seleccionada

Para cuando la aplicación del usuario finalice un módulo se encarga de enviar una instrucción de reinicio a la TAD para que todos los recursos que la aplicación del usuario haya habilitado, se deshabiliten y se pueda ejecutar el software de la TAD sin necesidad de desconectarla.

El software de los módulos para los recursos de E/S Digitales, PWM, E/S Analógicas, comunicación I2C y contador de eventos está dividido en dos partes, una se encarga de generar una trama para indicar a la TAD que se desea inicializar al recurso y la otra se encarga de actualizar o mostrar los valores del recurso que se está utilizando.

El módulo de E/S Digitales permite al usuario seleccionar uno de dos puertos digitales (A o B) y especificar si se usará como entrada o salida. En caso de que se configure como salida, es posible indicar el valor con el que se desea inicializar al puerto. A partir de estas configuraciones se genera la trama Inicializa-Pdigital y se envía por el puerto USB a la TAD para que el recurso se inicialice.

Cada vez que se ejecuta el módulo en la aplicación del usuario, se leen o escriben los valores del puerto según se haya configurado, mediante el uso de las tramas Lectura-Pdigital y Escritura-Pdigital. En la Figura 4.7 se muestra el módulo para E/S digitales y en la Tabla 4.3 se describen sus entradas y salidas.

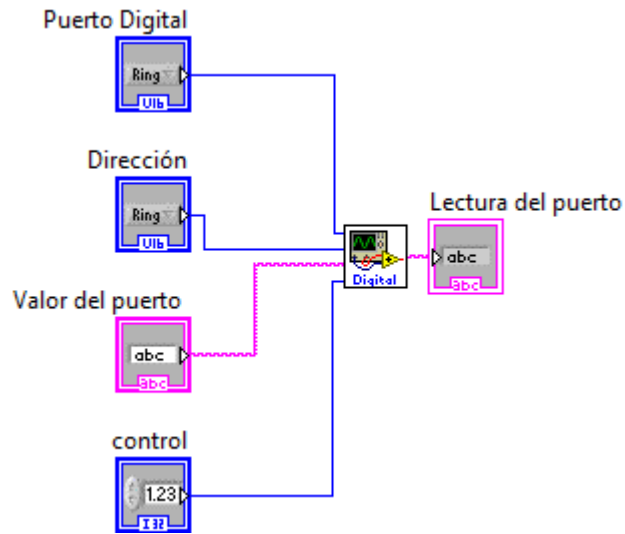


Figura 4.7 Módulo para la configuración de E/S digitales.

Tabla 4.3 Terminales del módulo de E/S Digitales.

Nombre	Flujo	Descripción
Puerto Digital	Entrada	Selecciona el puerto A o B
Dirección	Entrada	Selecciona si es Entrada o salida
Valor del puerto	Entrada	Valor del puerto en Hexadecimal
Control	Entrada	Control interno del módulo, indica que está en modo de configuración o en modo de operación
Lectura del puerto	Salida	Valor del puerto de la tarjeta

Las salidas PWM se pueden configurar mediante un módulo que permite seleccionar una de tres salidas PWM (A, B o C), la frecuencia y el ciclo de trabajo. A partir de estas configuraciones se genera la trama Inicializa-PWM y se envía a la TAD para que el recurso sea inicializado.

El valor del ciclo de trabajo con el que se va a inicializar se debe especificar en el rango de 0 a 100%. Es posible modificar el valor del ciclo de trabajo en tiempo de ejecución, para ello, cada vez que el módulo es ejecutado, se lee el nuevo valor y se envía en la trama Escritura-DC.

La frecuencia no puede ser modificada en tiempo de ejecución y el rango de operación es de 100 Hz a 1 MHz para todas las salidas. En la Figura 4.8 se muestra el módulo para la configuración de las salidas PWM y en la Tabla 4.4 se describen sus entradas y salidas.

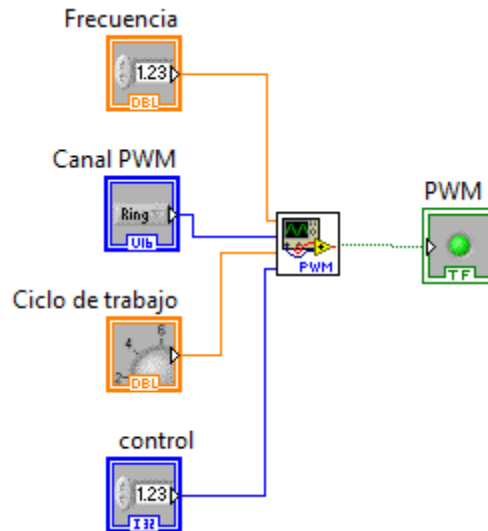


Figura 4.8 Módulo para la configuración de las salidas PWM.

Tabla 4.4 Terminales del módulo PWM.

Nombre	Flujo	Descripción
Frecuencia	Entrada	Entrada numérica de la frecuencia
Canal PWM	Entrada	Selecciona el canal A, B o C
Ciclo de Trabajo	Entrada	Entrada numérica del ciclo de trabajo
Control	Entrada	Control interno del módulo, indica que está en modo de configuración o en modo de operación
PWM	Salida	Indica que el módulo está en operación

Las entradas analógicas se habilitan mediante un módulo que permite seleccionar uno de ocho canales de entrada, de los cuales tres son para señales de 0 a 5 V, tres para señales de 0 a 10 V y dos para señales de -10 a 10 V, a partir de esto se genera la trama Inicializa-ADC y se envía a través del puerto USB a la TAD para que el recurso sea inicializado.

Cada vez que esté módulo se ejecuta en la aplicación del usuario, se lee la trama Lectura-ADC que envía a la TAD con el valor digital correspondiente a la señal analógica y se muestra al usuario de forma gráfica. En la Figura 4.9 se muestra el módulo para la manejo del ADC y en la Tabla 4.5 se describen sus entradas y salidas.

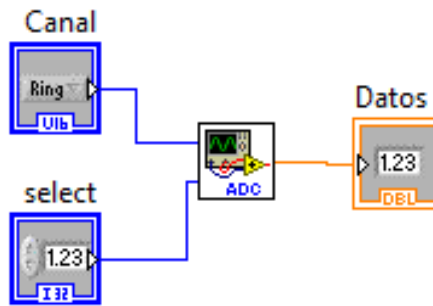


Figura 4.9 Módulo para el manejo del ADC.

Tabla 4.5 Terminales del módulo ADC.

Entrada	Flujo	Descripción
Canal	Entrada	Selecciona el canal (1 al 8)
Control	Entrada	Control interno del módulo, indica que esta en modo de configuración o en modo de operación
Datos	Salida	Muestra el dato convertido

Para la salidas Analógicas se diseñó un módulo que permite al usuario seleccionar uno de dos canales (A o B) y a partir de esto el módulo envía la trama Inicializa-DAC, la cual permite inicializar el recurso.

Cada vez que se ejecuta en la aplicación del usuario, se actualiza el valor digital y se crea la trama Escritura-ADC, la cual se envía a la TAD para que ésta haga la conversión digital analógica. En la Figura 4.10 se muestra el módulo del DAC y en la Tabla 4.6 se describen sus entradas y salidas.

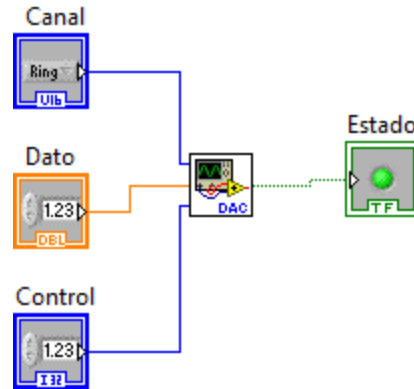


Figura 4.10 Módulo para la configuración del DAC.

Tabla 4.6 Terminales del módulo DAC.

Nombre	Flujo	Descripción
Canal	Entrada	Selecciona el canal A o B
Dato	Entrada	Entrada numérica del dato a enviar
Control	Entrada	Control interno del módulo, indica que esta en modo de configuración o en modo de operación
Estado	Salida	Indica el estado de operación del módulo

Para el contador de eventos se diseñó un módulo que permite al usuario seleccionar uno de dos canales (A o B), se debe indicar si se tomará la lectura del evento por flanco de bajada o por flanco de subida. A partir de esto el módulo envía la trama Inicializa-DAC, la cual permite inicializar el recurso. En la Figura 4.11 se muestra el diseño del módulo del DAC.

Cada vez que el módulo se ejecuta en la aplicación del usuario se verifica el estado del botón Reiniciar, si está activado, se debe de reiniciar la cuenta de eventos. El botón Tomar Lectura indica que se debe tomar la lectura del canal configurado. Si ambos botones de Reiniciar y Tomar Lectura están activados, se va a reiniciar la cuenta y posteriormente se toma la lectura. Tomando en cuenta la configuración del canal y del detector de flancos se genera la trama Escritura-Evento y se envía a la TAD para que configure el recurso. La cuenta del contador se lee de la trama Lectura-Evento que envía la TAD.

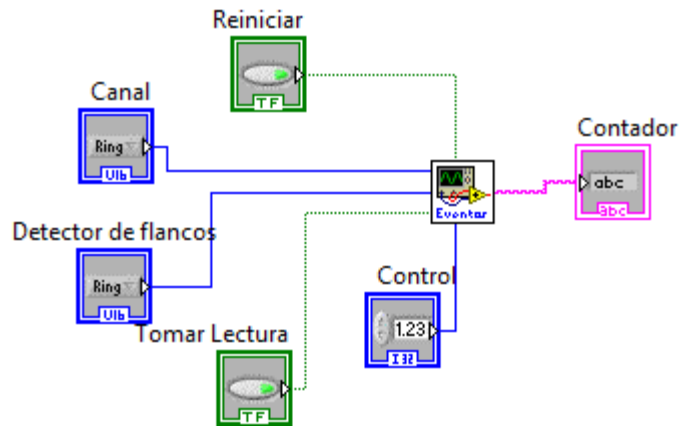


Figura 4.11 Módulo para la configuración del contador de Eventos.

En la Tabla 4.7 se muestran las entradas y salidas para el módulo de configuración para los contadores de eventos

Tabla 4.7 Descripción de las Entradas/Salidas del módulo de contador de Eventos.

Nombre	Flujo	Descripción
Reinicia	Entrada	Reinicia la cuenta
Canal	Entrada	Selecciona el canal A o B
Detector de Flancos	Entrada	Selecciona el detector de flancos de subida o de bajada
Tomar Lectura	Entrada	Con '1' lógico toma la lectura
Control	Entrada	Control interno del módulo, indica que está en modo de configuración o en modo de operación
Contador	Salida	Muestra la cuenta

Y por último, se diseñó un módulo para la comunicación bajo el protocolo I2C el cual permite a la TAD operar como nodo Maestro y así poder controlar hasta 128 dispositivos conectados como nodos Esclavo. Desde la interfaz en la PC se puede seleccionar la operación que se desee realizar, siendo estas las de Escritura o Lectura y a qué dispositivo se desea direccionar. En la Figura 4.12 se muestra el módulo para la configuración del protocolo I2C. Inicialmente cuando se ejecuta el módulo, se envía la trama Inicializa-I2C con lo que se indica

a la TAD que se desea habilitar dicho recurso. Posteriormente el módulo permitirá enviar las tramas Escritura_I2C o Lectura_I2C según se desee enviar o recibir datos.

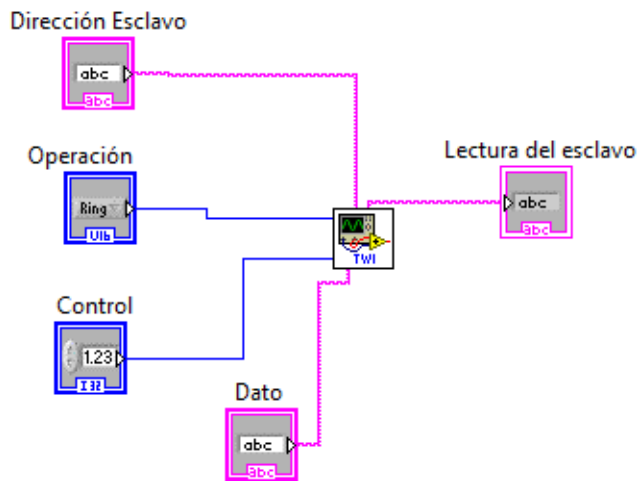


Figura 4.12 Módulo para la configuración de la comunicación por I2C.

En la Tabla 4.8 se muestra la descripción de las entradas y salidas del módulo para la comunicación por I2C.

Tabla 4.8 Descripción de las Entradas/Salidas del módulo de comunicación I2C.

Nombre	Flujo	Descripción
Dirección Esclavo	Entrada	Indica la dirección del nodo Esclavo, se expresa en decimal
Operación	Entrada	Permite seleccionar la operación que se desee realizar, Escritura o Lectura.
Dato	Entrada	Se indica el dato que se desea enviar al nodo Esclavo en caso de que se seleccione la operación de Escritura
Lectura del nodo Esclavo	Salida	Se muestra el dato leído del nodo Esclavo direccionado cuando se haya seleccionado la operación de Lectura
Control	Entrada	Control interno del módulo, indica que está en modo de configuración o en modo de operación

4.2.2 Software del Microcontrolador

En la Figura 4.13 se muestra la arquitectura del software del microcontrolador que permite configurar la comunicación con la PC bajo el protocolo de comunicación USB y la

aplicación de usuario que permite habilitar los recursos de la TAD y leer o escribir los valores de estos recursos.

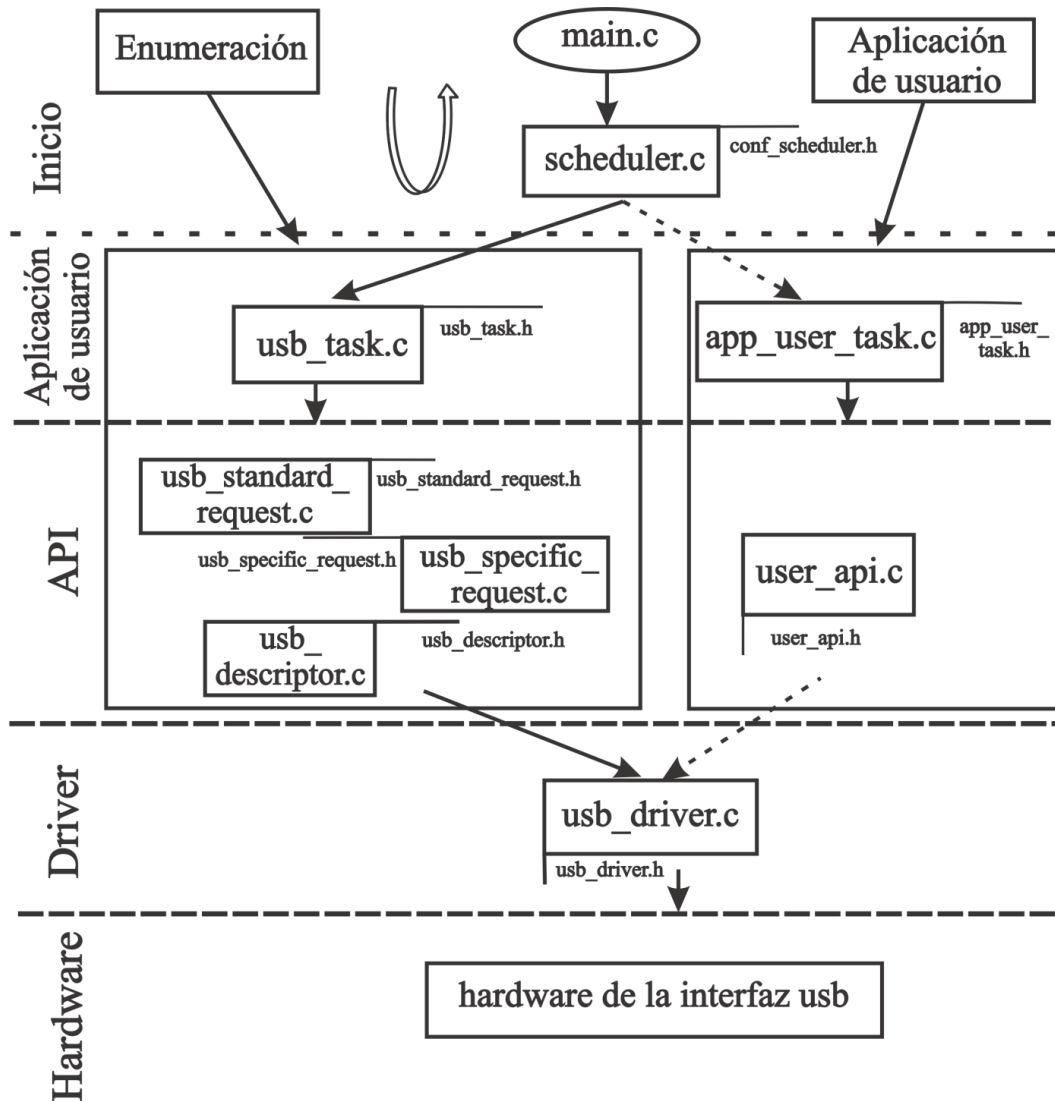


Figura 4.13 Arquitectura del software del microcontrolador [19].

En la parte correspondiente al inicio se tiene la función main, este código es el elemento principal de la arquitectura y permanece en un ciclo infinito. El código del archivo scheduler es ejecutado dentro del ciclo infinito y permite llamar a las tareas definidas en el archivo `conf_scheduler.h`.

La aplicación está basada en dos diferentes tareas:

- En la tarea USB (asociada al archivo fuente `usb_task.c`) se realiza el proceso de enumeración. Una vez que la tarea detecta que la unidad USB ha sido conectada, actualiza el estado de las banderas.
- La tarea de la aplicación (`app_user_task.c`) se encarga de atender la aplicación y establecer la interfaz con el usuario. En esta tarea se verifica que se reciban o envíen los datos.

La capa API permite la interconexión entre la capa de aplicación de usuario y la capa del driver. El archivo `usb_standard_request.c` maneja todos los requerimientos estándar, estos requerimientos son necesarios en cualquier clase de enumeración USB. En el archivo `usb_standard_request.h` se define el valor de los parámetros relacionados a los requerimientos estándar.

En el archivo `usb_specific_request.c` se definen los requerimientos específicos de la aplicación, estos requerimientos son propios de la aplicación e incluyen las funciones necesarias para su manejo. En el archivo `usb_specific_request.h` se define el valor de los requerimientos específicos.

En el archivo `usb_descriptor.c` se inicializan las variables y parámetros para la estructura del descriptor de la enumeración definidos en el archivo `usb_descriptor.h`.

El archivo `user_api.c` contiene la aplicación en donde se hace uso de los recursos del microcontrolador y se maneja la transferencia de datos con la PC.

En la Figura 4.14 se muestra un diagrama con el flujo que sigue del software para cumplir con los requerimientos de la TAD, este diagrama está implementado en el archivo `hid_task.c` y se encuentra en la capa de aplicación de usuario, el control de flujo proviene de la etapa de configuración del puerto de comunicación USB y una vez que se ejecuta la aplicación el control del flujo permanece en la parte del monitoreo del puerto USB.

Cuando se lee la trama almacenada en el buffer USB proveniente de la aplicación en la PC, se decodifica para saber a qué recurso hace referencia esta trama.

Si el recurso no ha sido inicializado, el software se encarga de configurar los registros necesarios para poder inicializarlo con los parámetros definidos por el usuario en la GUI.

Por el contrario, si el recurso ya ha sido inicializado, el software se encarga de leer o escribir los valores actuales de los recursos que previamente se han activado, al término de

esto si es necesario se crea una trama con el recurso que se atendió para que se escriba en el buffer USB y posteriormente se envíe a la GUI.

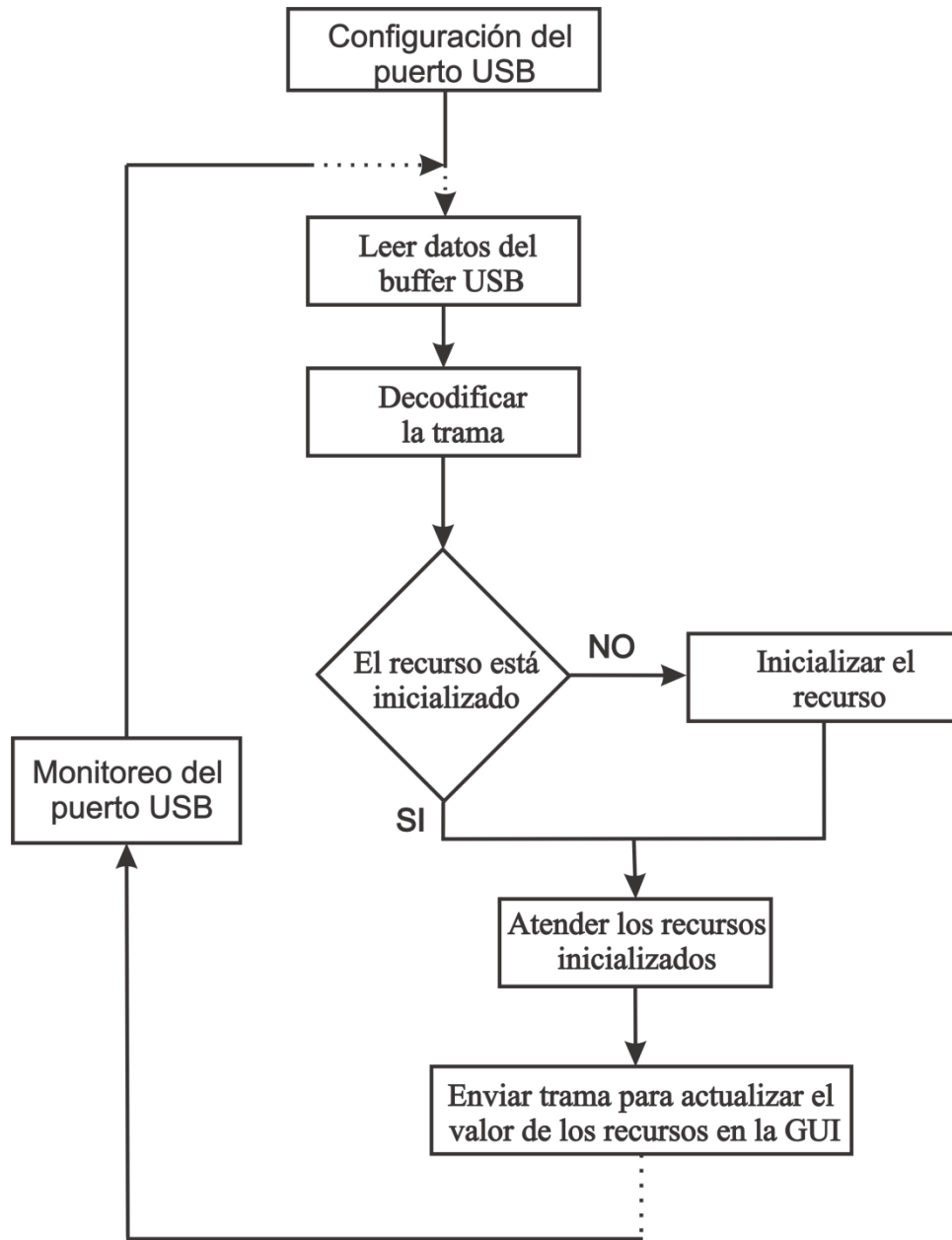


Figura 4.14 Diagrama de flujo del software para el manejo de los recursos de la TAD.

En la capa del driver se manejan los recursos del micro que permiten configurar al controlador USB. El archivo `usb_driver.c` presenta la interfaz entre el usuario y el hardware USB, contiene todas las rutinas para manejar el controlador USB.

Por su parte, el archivo `usb_driver.h` contiene la definición de los registros referentes al controlador USB.

Capítulo 5 Integración y pruebas

Una vez que el diseño del sistema se llevó a cabo, fue necesario hacer las pruebas pertinentes para corroborar que la integración del hardware y software de la Figura 5.1 funciona correctamente. Para ello se realizaron las siguientes pruebas:

- Verificación de la comunicación de la tarjeta con la PC, realizando pruebas de comunicación de la PC a la TAD y viceversa.
- Evaluación de los módulos ADC, PWM, Contador de Eventos, E/S Digitales, DAC e I2C para comprobar su correcto funcionamiento.
- Medición del tiempo de respuesta del ADC y del DAC.
- Ejecución de dos módulos en un mismo VI.

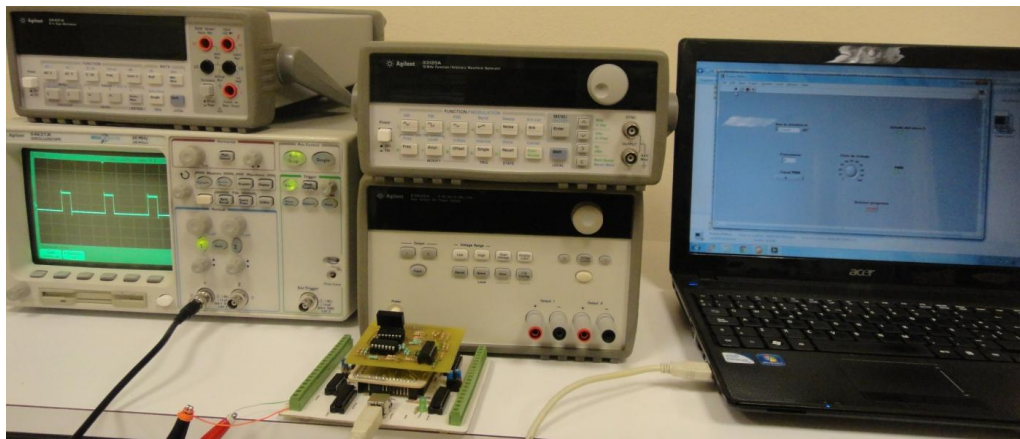


Figura 5.1 Integración del Hardware y Software

5.1 Prueba comunicación micro-PC y viceversa

Una vez que la TAD se conecta a la PC, se debe de configurar la comunicación, para ello se hace uso del módulo de la Figura 4.5. Este módulo tiene como parámetro de entrada la ruta del archivo DLL y como parámetro de salida al estado actual de la comunicación. En la Figura 5.2 se muestra una imagen de la interfaz del módulo con el usuario, el LED prende cuando la comunicación se establece correctamente, en caso contrario permanecerá apagado.



Figura 5.2 Interfaz de usuario para la configuración de la conexión.

Para el caso en que se desee saber si se llevó a cabo correctamente la conexión entre la TAD y la PC, se hace uso del módulo que permite enviar y recibir una instrucción de la TAD para prender y apagar un LED en la interfaz de la PC y en la TAD. En la Figura 5.3 se muestra la interfaz de usuario para llevar a cabo esta tarea. Se puede observar que la posición del interruptor “Apagar/Encender” está en la posición de Prender con lo cual se ha prendido el LED de la TAD y el estado del indicador “switch” señala que en la TAD el interruptor está conectado a tierra como se muestra en la Figura 5.4.



Figura 5.3 Interfaz del módulo para probar la comunicación.

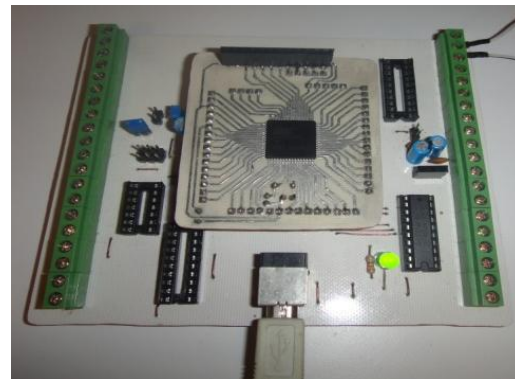


Figura 5.4 Pruebas de la comunicación de la TAD con la PC.

5.2 Evaluación de los Módulos

5.2.1 Prueba del módulo ADC

Las pruebas para el módulo del ADC se hicieron utilizando señales senoidales provenientes de un generador de funciones. La primera prueba se hizo para los canales que permiten entradas en el rango de 0 a 5 V, estas señales no pasan por ningún circuito de la TAD para su acondicionamiento, por lo que están conectadas directamente con el microcontrolador. En la Figura 5.5 se muestra una imagen de la señal de entrada recuperada en la interfaz de usuario y en la Figura 5.6 se muestra la señal de entrada a la TAD recuperada de un osciloscopio.

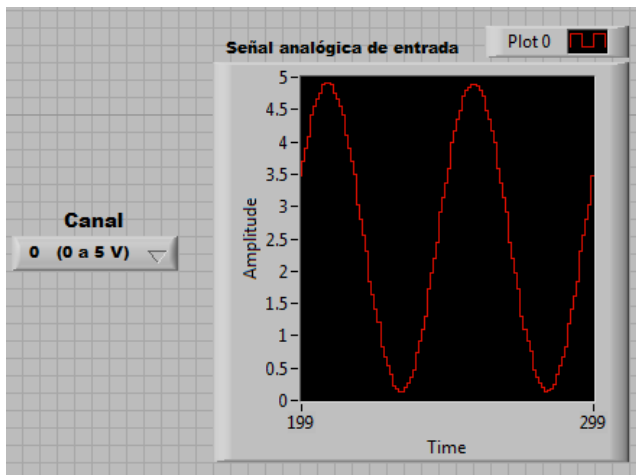


Figura 5.5 Señal convertida de 0 a 5V.

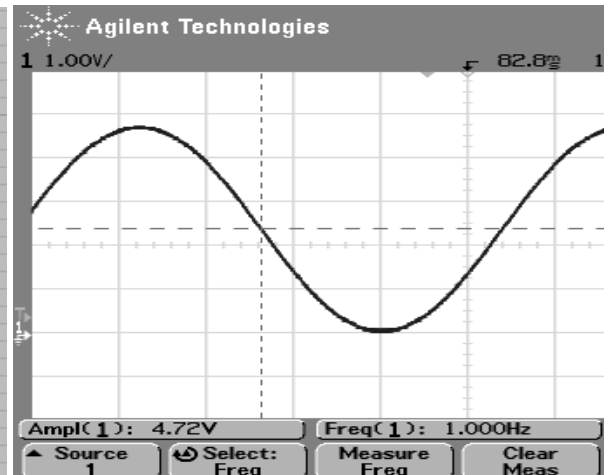


Figura 5.6 Señal de entrada al ADC.

Otra prueba que se realizó para el módulo del ADC, fue la de habilitar el canal de entrada que permite señales de entrada de 0 a 10 V. En la Figura 5.7 se muestra una imagen de la GUI en la cual se puede observar la configuración del canal y la señal convertida proveniente de la TAD. En la Figura 5.8 se muestra la señal de entrada a la TAD recuperada de un osciloscopio.

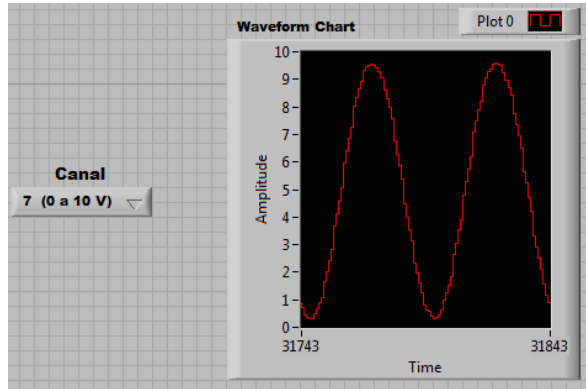


Figura 5.7 Señal convertida de 0 a 10 V.

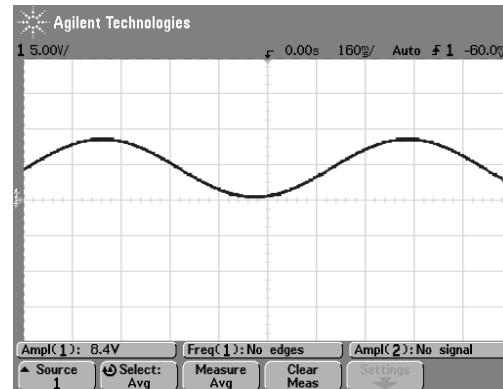


Figura 5.8 Señal de entrada al ADC de 0 a 10 V

5.2.2 Prueba del módulo PWM

Para comprobar que el módulo para la generación de señales PWM funciona correctamente según lo esperado, se habilita el canal B de la salida PWM, a una frecuencia de 100 kHz y un ciclo de trabajo de 60%.

En la Figura 5.9 se muestra la captura de la interfaz del módulo para la configuración del recurso.

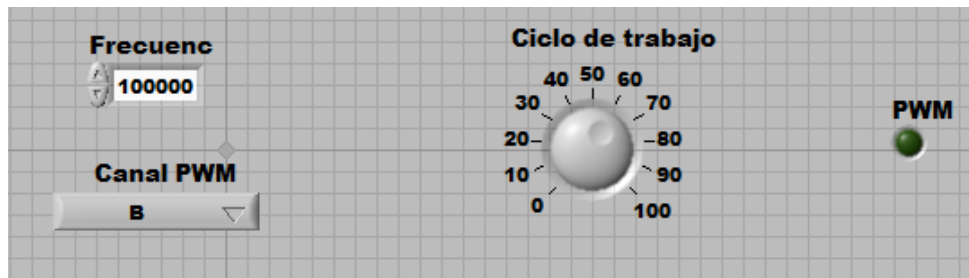


Figura 5.9 Interfaz para la configuración de las salidas en PWM.

En la imagen de la Figura 5.10 se muestra la señal de salida de la TAD, en la que se puede comprobar que la frecuencia de 100 kHz corresponde a la configurada en la interfaz de usuario. La imagen se tomó de un osciloscopio modelo 54600 del fabricante *Agilent Technologies*.

En la Figura 5.11 se observa que el tiempo en alto de la señal es de 6 uSeg, correspondiente al 60% del periodo de la señal que es de 10 uSeg.

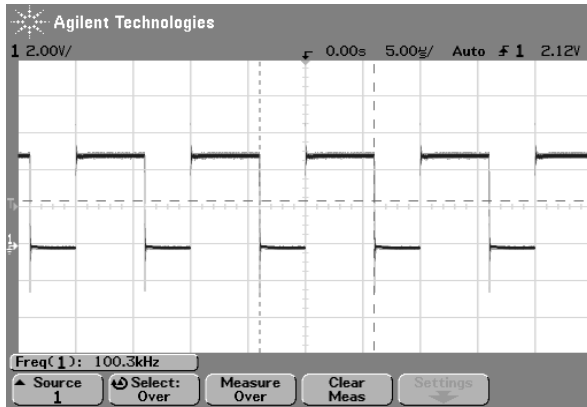


Figura 5.10 Medición de la frecuencia de la señal PWM generada por la TAD.

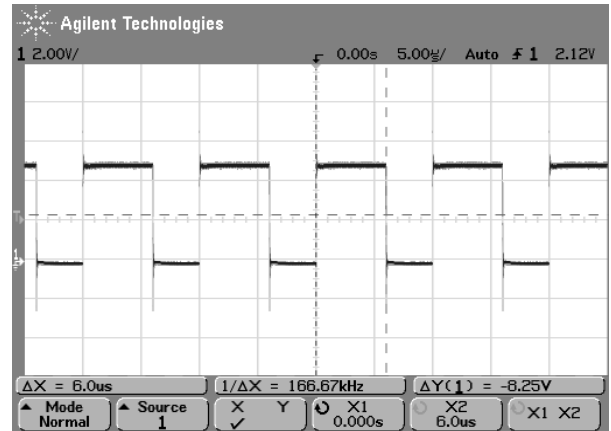


Figura 5.11 Medición del ciclo de trabajo.

5.2.3 Prueba del módulo Contador de Eventos

Para verificar el correcto funcionamiento del módulo que permite contar eventos externos se configuró el módulo como se muestra en la Figura 5.12.

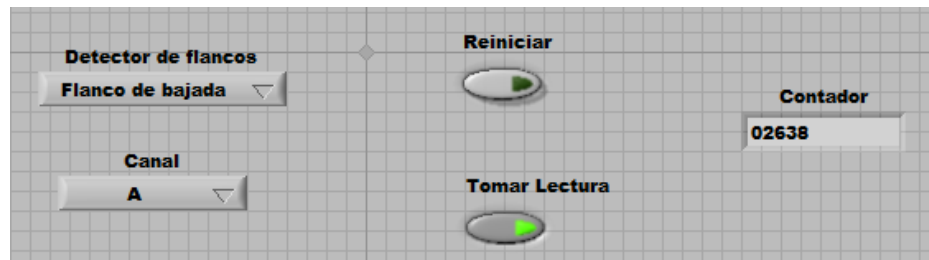


Figura 5.12 Configuración del contador de eventos.

Se hicieron las pruebas con la señal que se muestra en la Figura 5.13, la cual es proporcionada por un generador de funciones a una frecuencia de 1Hz, una amplitud de 5Vpp y un offset de 2.5V, todo esto para que la TAD pueda interpretar el '1' y '0' lógico.

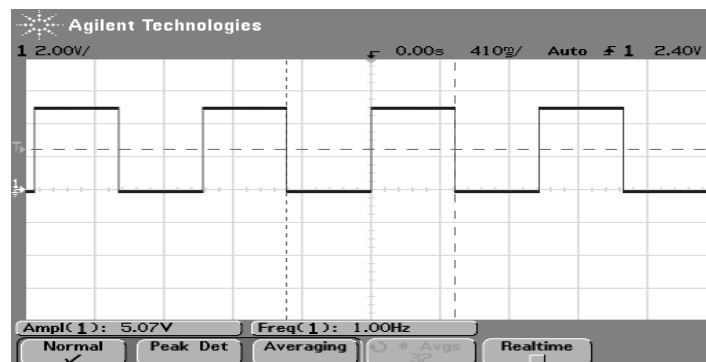


Figura 5.13 Señal de prueba para el modulo contador de eventos.

Se seleccionó el canal A, el cual es de 16 bits, por lo que la cuenta máxima es de 65536 eventos y se configuró para que detectara los flancos de bajada. En el recuadro con la etiqueta de Contador se muestra la cuenta en el momento en que el botón tomar lectura se habilita, el botón Reiniciar permite reiniciar la cuenta a cero en cualquier momento.

5.2.4 Prueba del módulo Entradas/Salidas Digitales

De la interfaz del módulo para las Entradas/Salidas digitales se seleccionó el puerto A para que operara como entrada según se muestra en la Figura 5.14. En el recuadro de lectura del puerto se indica el valor leído físicamente del puerto en formato hexadecimal. En la Figura 5.15 se muestra la conexión física de los interruptores que permiten introducir el valor del puerto digital. Debido a que en el puerto digital seleccionado se habilitan las resistencias de pull-up en el caso de que se configure como entrada, sólo es necesario indicar el valor de ‘0’ lógico poniendo el pin a tierra, por lo que se puede observar que la lectura del puerto según la configuración de los interruptores es de ‘F1’ en hexadecimal.

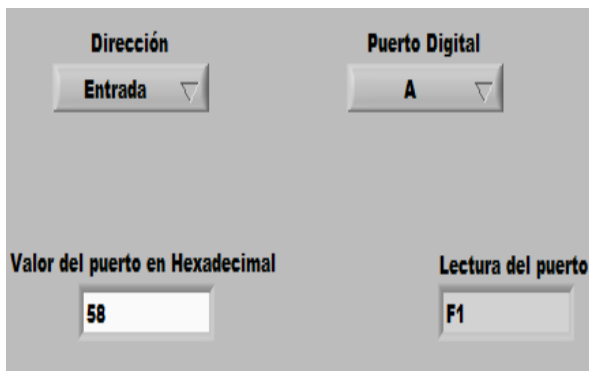


Figura 5.14 Interfaz del módulo E/S digitales.

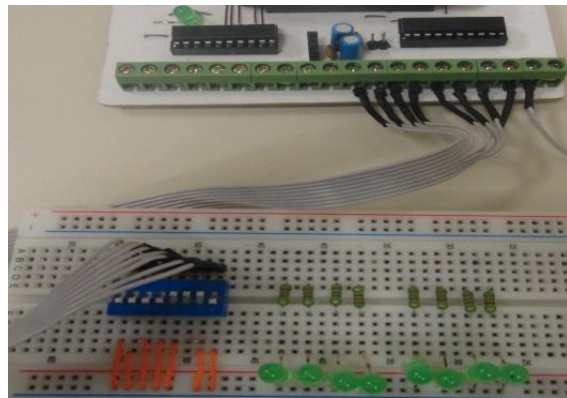


Figura 5.15 Conexión física de las entradas al puerto B.

En la Figura 5.16 se muestra el puerto B configurado como salida y en el recuadro “Valor del puerto en Hexadecimal” se indica que el valor que será escrito en el puerto es el 58 en hexadecimal. En la Figura 5.177 se puede observar que los LEDs encendidos representan dicho valor. Se toma la lectura de izquierda a derecha del LSB hasta el MSB.



Figura 5.16 Configuración del puerto digital B como salida.

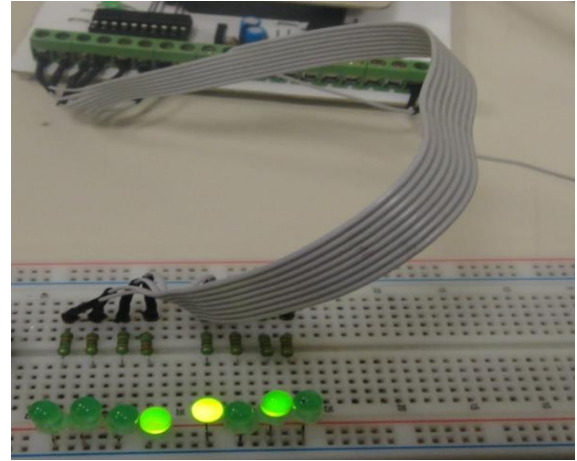


Figura 5.17 Salida del puerto digital B.

5.2.5 Prueba del módulo Convertidor Digital Analógico

El módulo de conversión de datos digitales a analógicos se probó generando una señal en el canal A como la que se muestra en la Figura 5.188 generada en LabVIEW, la cual corresponde a una señal tipo dientes de sierra. Los valores de dicha señal están expresados en enteros y van de 0 a 100, así al enviar estos datos a la TAD, ésta puede interpretarlos correctamente ya que tiene una resolución de 10 bits para el DAC.

En la Figura 5.199 se muestra la señal convertida por la TAD y recuperada de un osciloscopio.

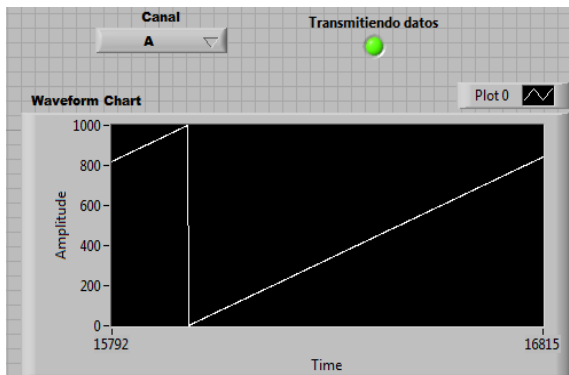


Figura 5.18 Interfaz de configuración del módulo DAC y señal de prueba generada.

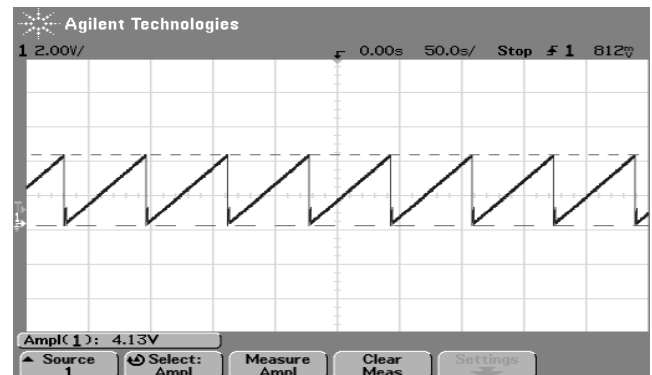


Figura 5.19 Señal Convertida recuperada del osciloscopio.

En la Figura 5.20 se puede observar la medición del tiempo de demora desde que se convierte un dato hasta que se obtiene la siguiente, el tiempo aproximadamente es de 15.96 ms.

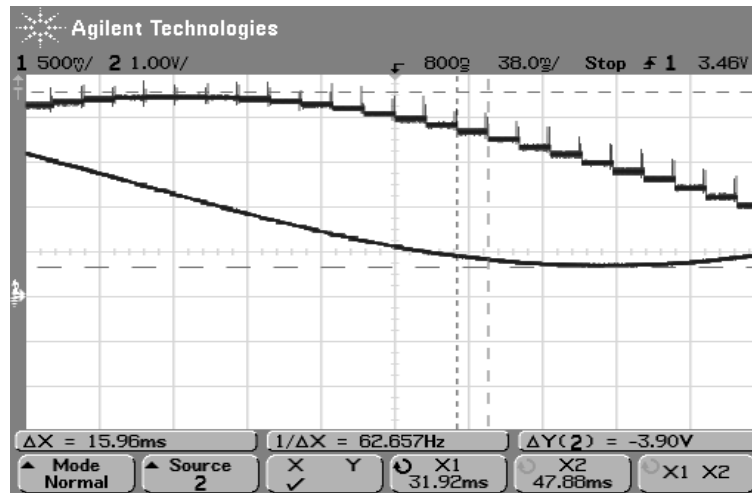


Figura 5.20 Medición del tiempo de conversión.

5.2.6 Prueba del módulo I2C

Las pruebas para el módulo I2C se realizaron con un sistema I2C externo, en la Figura 5.21 se muestra un diagrama a bloques.

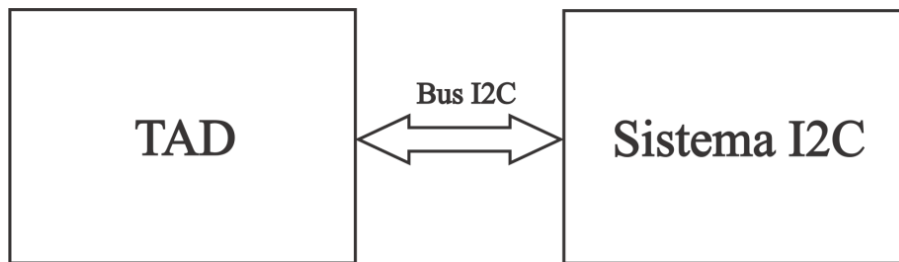


Figura 5.21 Diagrama a bloques del sistema I2C.

El sistema cuenta con 2 nodos Esclavo como el que se muestra en la Figura 5.22. Los nodos Esclavo son microcontroladores ATmega8 con dirección 1 para el de la parte de arriba y 2 para el de la parte de debajo de la imagen.

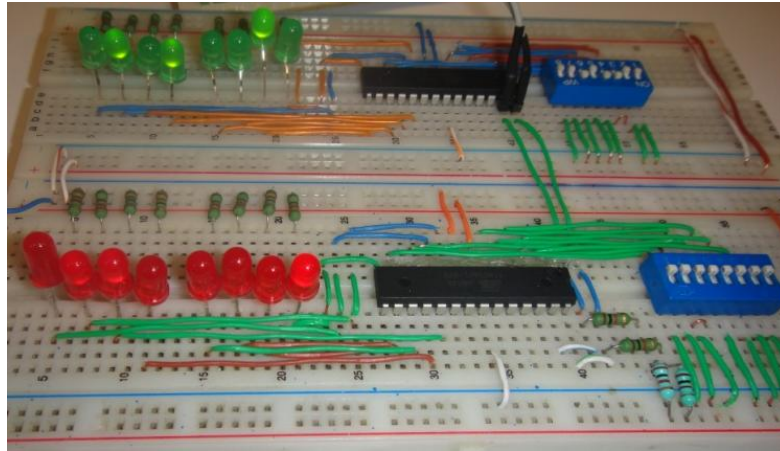


Figura 5.22 Sistema de prueba para el módulo I2C.

En la Figura 5.233 se muestra la GUI para el módulo I2C, en la que se indica un proceso de lectura del nodo Esclavo con dirección 2.

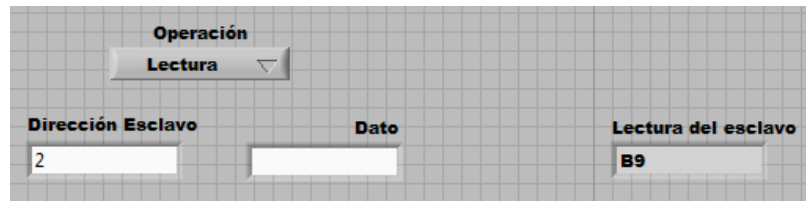


Figura 5.23 Configuración de la GUI para la lectura del nodo Esclavo.

En la Figura 5.24 se muestra el nodo Esclavo con dirección 2, el cual cuenta con un dip-switch que permite introducir el dato que el nodo Maestro (TAD) va a leer del nodo Esclavo, el interruptor en la posición hacia arriba de la imagen indica ‘1’ lógico y en la posición hacia debajo de la imagen indica ‘0’ lógico, el bit más significativo es el que se muestra a la izquierda de la imagen. Se puede apreciar que el dato introducido en formato hexadecimal es B9 y que éste coincide al que la GUI está desplegando en el recuadro de Lectura de nodo Esclavo en la Figura 5.23.

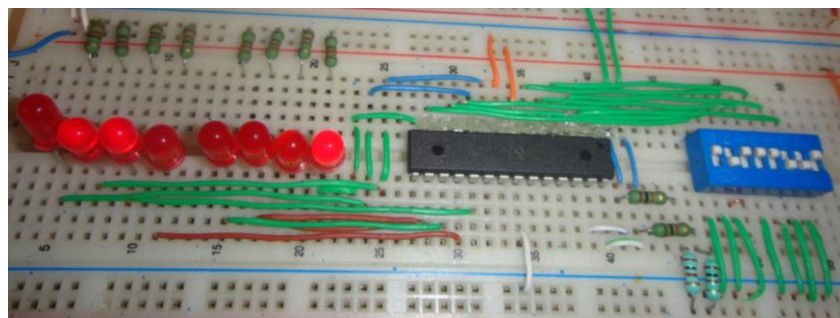


Figura 5.24 Configuración del dato de entrada del nodo Esclavo 1.

En la Figura 5.25 se muestra la configuración en la GUI para enviar a la TAD la instrucción de escritura. Se indica la dirección del nodo Esclavo 1 y se va a enviar el dato contenido en el recuadro Dato, el cual se expresa en formato hexadecimal y corresponde a 52.



Figura 5.25 Configuración de la GUI para la escritura del nodo Esclavo 1.

En la Figura 5.26 se puede observar que el nodo Esclavo con dirección 1 indica en los LEDs el valor que se le envía desde la GUI. El dato se lee de izquierda a derecha de la imagen, así el bit más significativo es el que se encuentra más a la izquierda de la imagen.

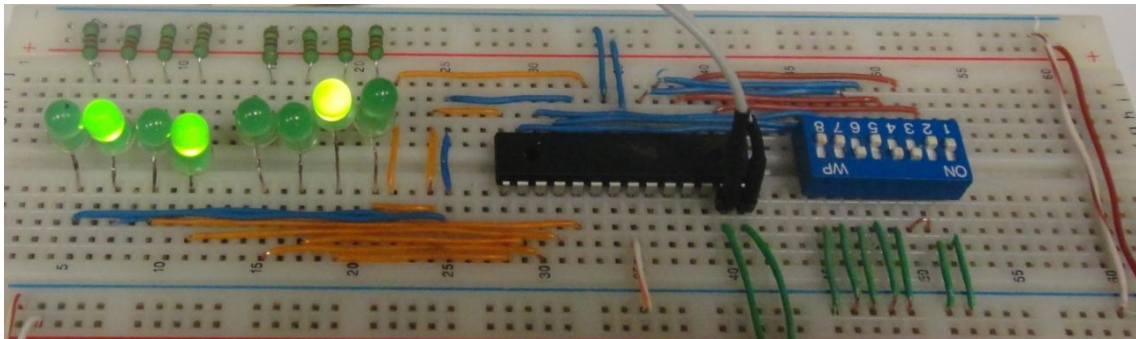


Figura 5.26 Datos de salida del nodo Esclavo 2.

5.2.7 Medición del tiempo de conversión del DAC y ADC

Para poder medir el tiempo que tarda el sistema desde que convierte un dato de una entrada analógica, la envía a la PC para que ésta lo regrese a la TAD y finalmente sea convertido a formato analógico, se utilizaron los módulos ADC y DAC.

En la Figura 5.27 se observa que la señal más definida corresponde a la señal de entrada al ADC y la señal correspondiente a la salida del DAC es la que presenta pequeñas distorsiones generadas por el tiempo de conversión. Se puede apreciar también que existe un desfase entre las señales de aproximadamente 288ms, lo cual indica el tiempo de propagación del dato a través del ADC, la PC y el DAC.

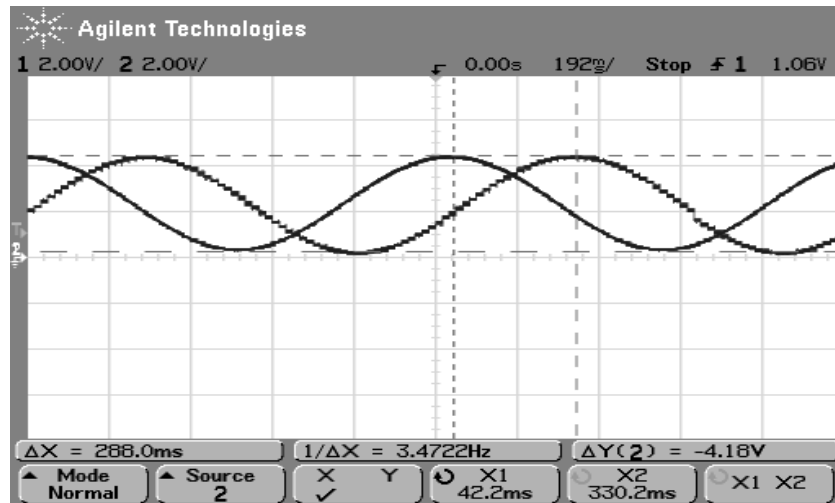


Figura 5.27 Señal de entrada del ADC y señal de salida del DAC.

5.2.8 Prueba de dos módulos en un mismo VI

Para esta prueba se hace uso del módulo de entrada ADC con el canal 0 y salida PWM con el canal A. Cuando el valor de conversión del ADC es mayor que 2V, correspondiente a la mitad del valor máximo, el valor de la salida PWM es de 90% del ciclo de trabajo a una frecuencia de 100Hz. Y cuando el valor de conversión está por debajo de los 2V, la salida PWM tiene 20% del ciclo de trabajo, pero estos pueden variar según la aplicación los requiera. En la Figura 5.288 se puede observar esta relación, la señal ascendente corresponde a la señal senoidal de entrada al ADC a una frecuencia de 1 Hz y la señal cuadrada corresponde a la señal PWM.

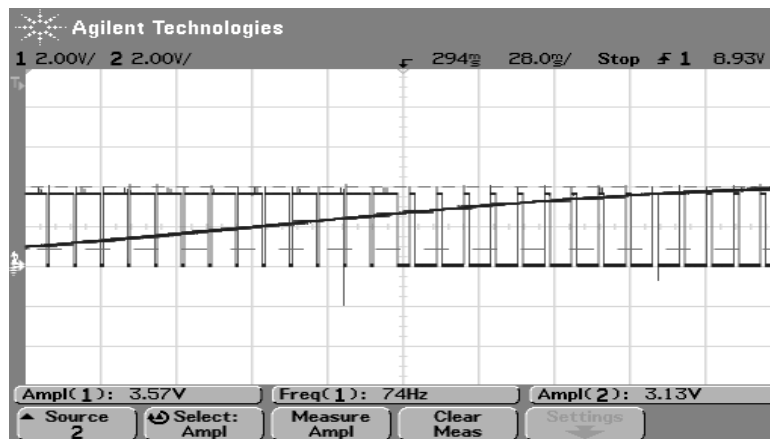


Figura 5.28 Generación de una señal PWM a partir de una señal analógica.

Capítulo 6 Conclusiones y trabajos futuros

Del presente trabajo se concluye que se alcanzaron satisfactoriamente los objetivos, ya que se logró diseñar e implementar una TAD que cuenta con los recursos de ADC, PWM, Contador de Eventos, E/S Digitales, DAC e I2C.

La TAD puede ser reconocida por cualquier sistema operativo debido a que utiliza los drivers de un Dispositivo de Interfaz Humana (HID) el cual es estándar para los sistemas operativos Windows, Macintosh y Linux. En el presente trabajo todas las pruebas realizadas se hicieron bajo el sistema operativo Windows 7. A pesar de que existen versiones de LabVIEW para Linux y para Macintosh, no se hicieron pruebas en esas plataformas.

Los módulos desarrollados en LabVIEW cuentan con una ayuda rápida, la cual puede ser visualizada al seleccionar el módulo seguidamente del comando “ctrl + h”. En dicha ayuda se detallan la función del módulo así como el tipo de dato para las entradas y salidas. La validación de las entradas y salidas de los módulos se realizó definiendo el tipo de datos y el formato en que debían de ser introducidos, los módulos no son capaces de manejar el dato si se indica de otra forma a lo definido y no será capaz de entregar los resultados esperados.

Es posible utilizar varios módulos del mismo recurso, por ejemplo para las salidas PWM que cuentan con 3 canales A, B y C, es válido poner en un VI de la interfaz de LabVIEW 3 módulos PWM, pero en cada módulo se debe indicar el canal que se desee usar para que puedan ser controlados independientemente.

El algoritmo que permite inicializar y atender a los recursos en la TAD está diseñado para que cada recurso sea inicializado y atendido en el mismo orden en el que se va recibiendo

la petición desde la GUI. Todos los recursos inicializados se insertan en una estructura tipo FIFO, en la que la primera petición en llegar es la primera en atenderse, no se maneja ningún tipo de prioridades y para algunos casos se usan variables temporales para almacenar datos hasta que sean enviados a la GUI.

Cada módulo en la interfaz de LabVIEW cuenta con un mecanismo que permite indicarle a la TAD que desea inicializarse o que desea enviar o recibir datos. El programa del usuario en la GUI puede detenerse en cualquier momento, ya sea por una entrada de control o por el botón de detener. Una vez que el programa de aplicación está detenido, los recursos que se hayan usado se quedarán inicializados y seguirán estando disponibles para usarse en la TAD a pesar de que la aplicación esté detenida, siempre que no se haya interrumpido la alimentación. Para utilizar nuevos o los mismos recursos, si fuera necesario en la aplicación, se debe aplicar un *reset* a la TAD, el cual puede ser por hardware o por software. Cada vez que se aplica el *reset* la comunicación USB se pierde y la PC reinicia el mecanismo que permite establecer nuevamente la comunicación USB.

La placa que permite el acondicionamiento de señales analógicas de 0 a 10V y de -10 a 10V para los canales del ADC, cuenta con dispositivos electrónicos que consumen corriente a pesar de que no se estén usando dichos canales, debido a esto, el diseño de la TAD permite desconectarlo físicamente si en la aplicación el usuario no hace uso de estos canales, sin que esto afecte al resto de los recursos.

6.1 Trabajos futuros.

A partir de los resultados obtenidos, se proponen los siguientes trabajos de investigación:

- Migrar la plataforma a otros Sistemas Operativos.
- Programar la validación de las entradas para los módulos desarrollados en LabVIEW.
- Asignar prioridades a cada recurso y diseñar un algoritmo que permita atender cada recurso inicializado en la TAD con base en su prioridad.
- Diseñar una carcasa que permita cubrir la PCB (*Printed Circuit Board*, Placa de Circuito Impreso) para poder protegerla durante su utilización y mejorar la presentación con el usuario.

- Implementar un algoritmo que maneje los errores que se generen en la comunicación de la PC con la TAD.
- Diseñar un algoritmo y acondicionar una memoria externa al MCU, para poder almacenar muestras y enviarlas de la TAD a la PC y viceversa, esto para dotar a la TAD de la capacidad de manipular señales con una frecuencia mayor a la que actualmente se puede emplear por el ADC y el DAC.

Bibliografía

- [1] About USB Implementers Forum, Inc. recuperado de <http://www.usb.org>. Fecha de acceso 5 mayo de 2011.
- [2] Arnold, S. B. Embedded Systems Design: An Introduction to Processes, Tools, and Techniques. United States of America: CMP Books, 2002.
- [3] AVR276: USB Software Library for AT90USBxxx Microcontrollers. Hojas de especificaciones. Atmel Corporation. 2007.
- [4] AVR328: USB Generic HID Implementation. Hojas de especificaciones. Atmel Corporation. 2008.
- [5] AVR329: USB Firmware Architecture. Hoja de especificaciones. Atmel Corporation. 2006.
- [6] Axelson J. USB Complete. Lakeview Research. USA. 2001.
- [7] Compaq Computer Corporation, Hewlett-Packard Company, Intel Corporation, Lucent Technologies Inc, Microsoft Corporation, NEC Corporation, Koninklijke Philips Electronics N.V. Universal Serial Bus Specification. 2000.
- [8] Coughlin F.R, Driscoll F.F. Amplificadores Operacionales y Circuitos Integrados Lineales. Prentice Hall, 1999.
- [9] Definición y explicación de un archivo DLL. Recuperado de <http://support.microsoft.com/kb/87934/es>, fecha de acceso: 11 de febrero de 2011.
- [10] El microcontrolador AT90USB646. Hojas de especificaciones .Atmel Corporation. 2009.
- [11] Firmware para dispositivo esclavo USB de Clase HID. Emanuel G. Aguirre, Pablo A. Di Giulio, Universidad Tecnológica Nacional, Facultad Regional San Francisco, pp. 2.
- [12] Hoja de especificaciones del circuito integrado MCP4921. Microchip. 2004.
- [13] Información de Producto: ¿Qué es NI LabVIEW? Recuperado de <http://www.ni.com/labview/whatis/esa/>, fecha de acceso: 29 de enero de 2011.
- [14] Introducción a LabVIEW Curso de Seis-Horas. National Instruments Corporation. Septiembre 2003.
- [15] Josep, B., y José, L.R. Ordenadores industriales compatibles PC. En Balcells, J., y Romeral, J.L. 1997.
- [16] Mandado, P.E., y Mariño, E.P, y Lago, F.A. Instrumentación electrónica. Barcelona España: Marcombo, 1995.
- [17] McDowell, S. y Seyer, D.M. USB explained. United States of America, Prentice-Hall,

1999.

- [18] NI USB-6008 Recuperado de <http://sine.ni.com/nips/cds/view/p/lang/es/nid/201986>, fecha de acceso: 29 de enero de 2011.
- [19] PMD-1208FS. Hojas de especificaciones. Measurement Computing Corporation. 2005.
- [20] Raj, K. Embedded System. McGraw-Hill. 2008.
- [21] Santiago, E.F. Los Microcontroladores AVR de Atmel. Universidad Tecnológica de la Mixteca, ISBN-978-607-95222-7-8, 2012.
- [22] USB Class Definitions for Communication Devices. USB Implementers' Forum. 19 de junio de 1999.
- [23] USB-IF eStore Details. Recuperado de <http://www.usb.org/developers/estoreinfo/>. Fecha de acceso 10 de mayo de 2011.
- [24] USB MASS Storage Class CBI Transport Specification. USB Implementers Forum. 2003.
- [25] USB PC Drivers Based on Generic HID Class. Hojas de especificaciones. Atmel Corporation. 2008.
- [26] Using external code in LabVIEW, National Instrument, Julio del 2000.
- [27] USB 3.0 Specification. Recuperado de <http://www.usb.org/developers/docs/>. Fecha de acceso 10 de mayo de 2011.

Apéndice A Tramas de configuración y monitoreo de los recursos

Para inicializar y monitorear los recursos de la TAD, se diseñaron tramas de comunicación que la TAD y la GUI deben interpretar correctamente. Cada recurso cuenta con sus propias tramas tanto de inicialización del recurso como de atención o actualización de los valores del recurso según sea el caso. A continuación se muestran las tramas de cada recurso y se explica brevemente cada uno de sus campos. Se hace referencia a cada trama como trama de lectura o escritura desde el punto de vista de la GUI.

El módulo que permite comprobar que se estableció la comunicación entre la TAD y la GUI emplea 2 tramas. Los campos de las tramas Escribir-Led y Lectura-Led son similares y se muestran en la Figura A.1.

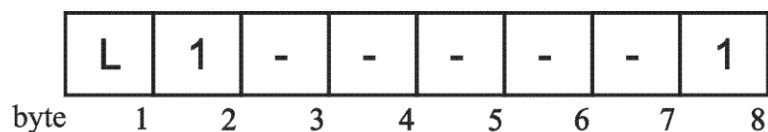


Figura A.1 Trama Escribir-Led y Lectura-Led.

- **byte 1:** Indica el recurso al que se hace referencia, se usa L para indicar esta trama.
- **byte 2:** Se usa para indicar el estado del LED o el interruptor, '1' prendido y '0' apagado para cuando sea de escritura y cuando se use como lectura el '1' indica un 1 lógico y el '0' indica un 0 lógico.
- **byte 3-7:** Sin uso.
- **byte 8:** El 1 indica que es una trama de escritura a la TAD.

Para inicializar el recurso de Entradas/Salidas digitales se usa la trama Inicializa-Pdigital y sus campos se muestran en la Figura A.2.

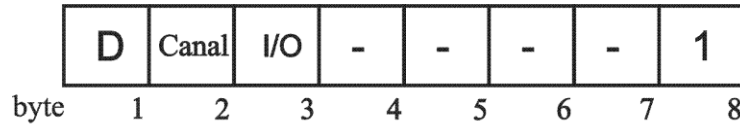


Figura A.2 Trama Inicializar-Pdigital.

- **byte 1:** Indica el recurso al que se hace referencia, se usa D para indicar que es una trama del recurso digital.
- **byte 2:** Se usa para indicar el puerto digital que se desea usar, A o B.
- **byte 3:** Se usa para indicar si el puerto seleccionado es de entrada o salida. El '1' salida y '0' entrada.
- **byte 4-7:** Sin uso.
- **byte 8:** El 1 indica que es una trama de inicialización de un recurso.

En la Figura A.3 se muestra la trama Lectura-Pdigital, la cual permite leer el valor digital del puerto seleccionado.

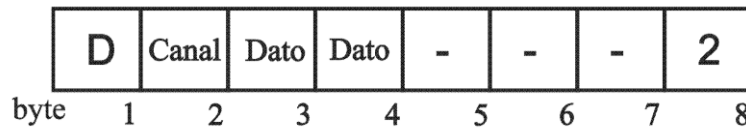


Figura A.3 Trama Lectura-Pdigital.

- **byte 1:** Indica el recurso al que se hace referencia, se usa D para indicar que es una trama del recurso digital.
- **byte 2:** Se usa para indicar el puerto, A o B.
- **byte 3-4:** Contiene el valor del puerto en formato hexadecimal, el byte 3 indica la parte alta y el byte 4 la parte baja.
- **byte 5-7:** Sin uso.
- **byte 8:** El 2 indica que es una trama de lectura.

En la Figura A.4 se muestra la trama Escritura-Pdigital, la cual permite actualizar el valor del puerto digital que se esté usando.

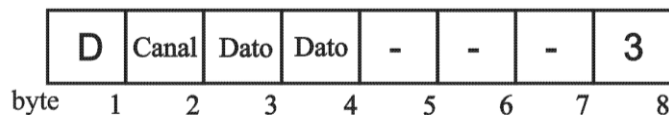


Figura A.4 Trama Escribir-Pdigital.

- **byte 1:** Indica el recurso al que se hace referencia, se usa D para indicar que es una trama del recurso digital.
- **byte 2:** Se usa para indicar el puerto, A o B.
- **byte 3-4:** Contiene el valor del puerto en formato hexadecimal, el byte 3 indica la parte alta y el byte 4 la parte baja.
- **byte 5-7:** Sin uso.
- **byte 8:** El 3 indica que es una trama de escritura.

La inicialización del recurso PWM se lleva a cabo a partir de la trama Inicializa-PWM,

En la Figura A.5 se muestran los campos de dicha trama.

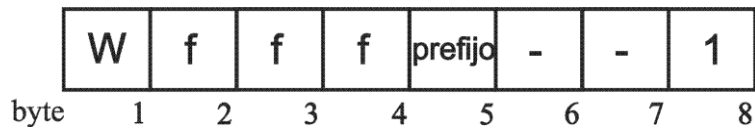


Figura A.5 Trama Inicializa-PWM.

- **byte 1:** Indica el recurso al que se hace referencia, se usa W para indicar que es una trama del recurso PWM.
- **byte 2-4:** Se usa para indicar la frecuencia de operación, el valor esta en decimal.
- **byte 5:** Indica el prefijo de la frecuencia, M (MegaHertz) y K (KiloHertz).
- **byte 6-7:** Sin uso.
- **byte 8:** El 1 indica que es una trama de inicialización.

Para actualizar el valor del PWM en los canales A, B o C, se usa la trama Escritura-DC mostrada en la Figura A.6.

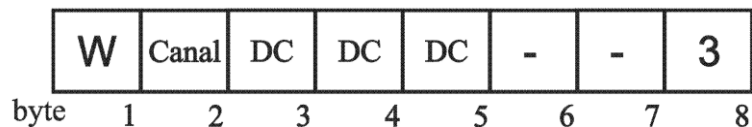


Figura A.6 Trama Escritura-PWM.

- **byte 1:** Indica el recurso al que se hace referencia, se usa W para indicar que es una trama del recurso PWM.
- **byte 2:** Se usa para indicar el canal A, B o C.
- **byte 3-5:** Indica el valor del ciclo de trabajo, 0 a 100 %.
- **byte 6-7:** Sin uso.
- **byte 8:** El 3 indica que es una trama de escritura.

El recurso del ADC se inicializa con la trama Inicializa-ADC y los campos de dicha trama se muestran en la Figura A.7.

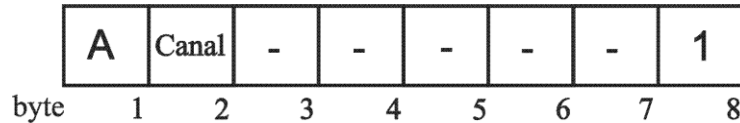


Figura A.7 Trama Inicializa-ADC.

- **byte 1:** Indica el recurso al que se hace referencia, se usa A para indicar que es una trama del recurso ADC.
- **byte 2:** Se usa para seleccionar el canal. El '0' indica que se usará el canal 0, '1' para el canal 1 y así sucesivamente hasta el canal 7.
- **byte 3-7:** Sin uso.
- **byte 8:** El '1' indica que es una trama de inicialización.

Para poder leer el valor del canal se usa la trama Lectura-ADC, la cual se conforma por los campos que se muestran en la Figura A.8.

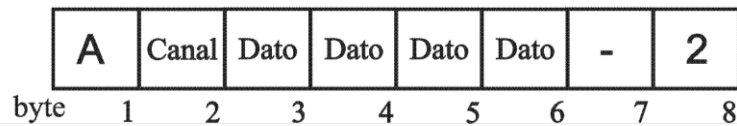


Figura A.8 Trama Lectura-ADC.

- **byte 1:** Indica el recurso al que se hace referencia, se usa A para indicar que es una trama del recurso ADC.
- **byte 2:** Se usa para indicar el canal al que se hace referencia, '0' para indica el canal 0, '1' para el canal 1 y así sucesivamente hasta el canal 7.
- **byte 3-6:** Indican el dato de la conversión, están en formato decimal.
- **byte 7:** Sin uso.
- **byte 8:** El 2 indica que es una trama de lectura.

Para configurar el recurso DAC se hace uso de la trama Inicializa-DAC. Los campos se muestran a continuación en la Figura A.9.

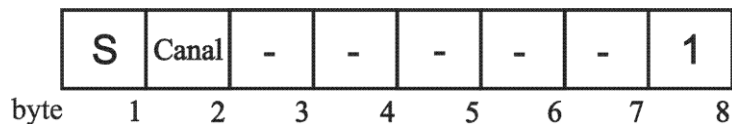


Figura A.9 Trama Inicializa-DAC.

- **byte 1:** Indica el recurso al que se hace referencia, se usa S para indicar que es una trama del recurso ADC.
- **byte 2:** Se usa para indicar el canal A o B.
- **byte 3-7:** Sin uso.

- **byte 8:** El '1' indica que es una trama de inicialización.

Para poder enviar el dato digital y que sea convertido se usa la trama Escribir-DAC, los campos de esta trama se muestran en la Figura A.10.

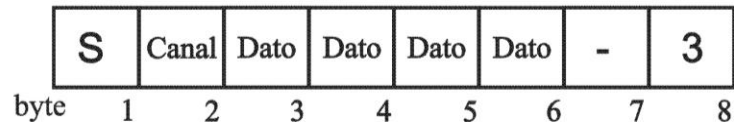


Figura A.10 Trama Escribir-DAC.

- **byte 1:** Indica el recurso al que se hace referencia, se usa S para indicar que es una trama del recurso ADC.
- **byte 2:** Se usa para indicar el canal A o B que se va a emplear para la conversión de datos.
- **byte 3-6:** Contiene el dato en formato digital.
- **byte 8:** El 3 indica que es una trama de escritura.

El recurso del contador de eventos se configura mediante la trama Inicializa-Evento. Los campos de esta trama se muestran en la Figura A.11.

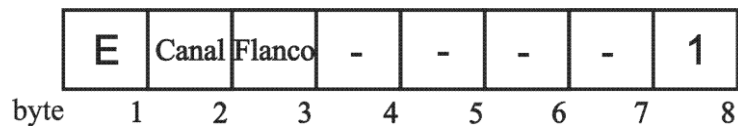


Figura A.11 Inicialización-Evento.

- **byte 1:** Indica el recurso al que se hace referencia, se usa E para indicar que es una trama del recurso Eventos.
- **byte 2:** Se usa para indicar el canal A o B que se desea usar.
- **byte 3:** Indica el flanco para la detección de eventos. Con el '6' se indica flanco de bajada y el '7' para indicar flanco de subida.
- **byte 4-7:** Sin uso.
- **byte 8:** El '1' indica que es una trama de inicialización.

En la Figura A.12 se muestra la trama Escribe-Evento, la cual se emplea para hacer la petición de lectura del contador o indicar que se desea reiniciar el contador.

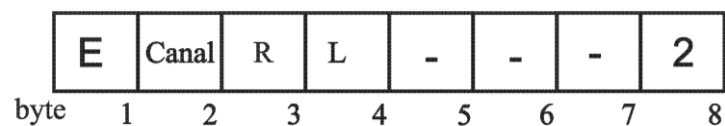


Figura A.12 Trama Escribe-Evento.

- **byte 1:** Indica el recurso al que se hace referencia, se usa E para indicar que es una trama del recurso Eventos.
- **byte 2:** Se usa para indicar el canal A o B.
- **byte 3:** '1' indica que se debe reiniciar el contador y '0' permanece la cuenta.
- **byte 4:** '1' indica que se desea leer la cuenta de eventos y '0' no realiza ninguna tarea.
- **byte 5-7:** Sin uso.
- **byte 8:** El '2' indica que es una trama de escritura.

Una vez que se hace la petición de lectura en la GUI, se lee la trama Lectura-Evento, la cual contiene la cuenta de los eventos en la Figura A.13 se muestra dicha trama.

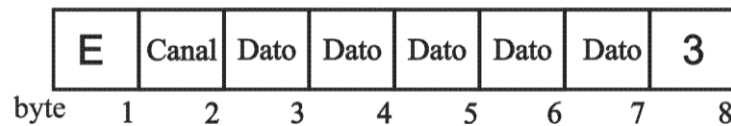


Figura A.13 Trama Lectura-Evento.

- **byte 1:** Indica el recurso al que se hace referencia, se usa E para indicar que es una trama del recurso Eventos.
- **byte 2:** Se usa para indicar el canal A o B.
- **byte 3-7:** Contiene la cuenta de los eventos en formato decimal
- **byte 8:** El '3' indica que es una trama de lectura.

En la Figura A.14 se muestra la trama que genera la GUI para indicarle a la TAD que se desea inicializar el recurso I2C.

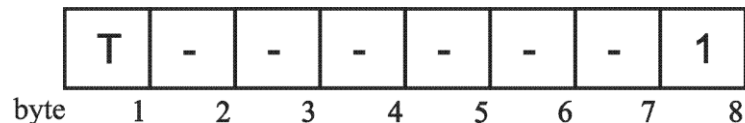


Figura A.14 Trama Inicializa_I2C.

- **byte 1:** Indica el recurso al que se hace referencia, se usa T para indicar que es una trama del recurso I2C.
- **byte 2-7:** Sin uso.
- **byte 8:** Indica que es una trama de inicialización.

En la Figura A.15 se muestra la Trama que genera la GUI para hacer la petición a la TAD que se desea escribir en el bus I2C.

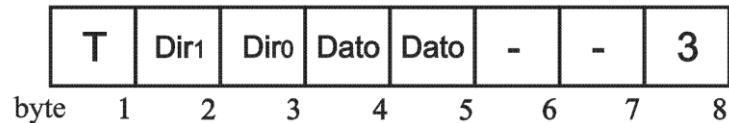


Figura A.15 Trama Escritura_I2C.

- **byte 1:** Indica el recurso al que se hace referencia, se usa T para indicar que es una trama del recurso I2C.
- **byte 2-3:** Indica la dirección del nodo Esclavo en formato Hexadecimal.
- **byte 4-5:** Indica el dato que se va a escribir, en formato Hexadecimal.
- **byte 6-7:** Sin uso.
- **byte 8:** Indica que es una trama de escritura.

En la Figura A.16 se muestra la trama que se envía a la TAD para hacer la petición de Lectura.

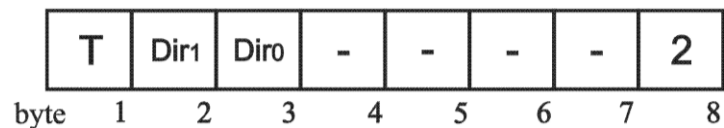


Figura A.16 Trama Petición Lectura_I2C.

- **byte 1:** Indica el recurso al que se hace referencia, se usa T para indicar que es una trama del recurso I2C.
- **byte 2-3:** Indica la dirección del nodo Esclavo en formato Hexadecimal.
- **byte 4-7:** Sin uso.
- **byte 8:** Indica que es una trama de Petición de Lectura_I2C.

Una vez que se hace la petición de lectura, la TAD responde con la trama Lectura_I2C en la que la GUI puede leer el dato proveniente del nodo Esclavo al que solicito la lectura. En la Figura A. 17 se muestra la Trama Lectura_I2C.

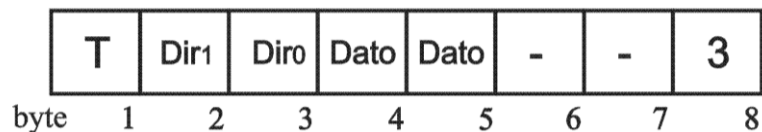


Figura A. 17 Trama de Lectura_I2C.

- **byte 1:** Indica el recurso al que se hace referencia, se usa T para indicar que es una trama del recurso I2C.
- **byte 2-3:** Indica la dirección del nodo Esclavo en formato Hexadecimal.
- **byte 4-5:** Dato que se lee del nodo Esclavo direccionado, en formato Hexadecimal.

- **byte 6-7:** Sin uso.
- **byte 8:** Indica que es una trama de Petición de Lectura_I2C.

Apéndice B Manual de Usuario

Cuando se conecta la TAD a la PC, aparece un mensaje del sistema operativo como el que se muestra en la Figura B.1, el cual indica que la tarjeta se está comunicando con la PC y que tomará unos momentos en instalar el software del controlador del dispositivos

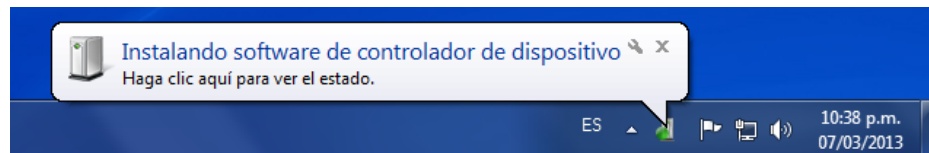


Figura B.1 Mensajes de Windows mientras instala los drivers de la TAD.

Una vez que este proceso termina, Windows manda otro mensaje en el que se indica que el proceso de instalación ha terminado correctamente tal como se muestra en la Figura B. 2.

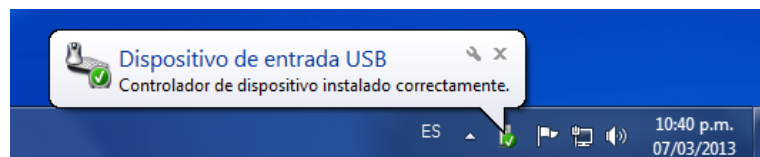


Figura B. 2 Fin de la instalación del dispositivo.

Para comprobar que la TAD se ha instalado correctamente, se abre la ventana del administrador de dispositivos y en la pestaña Dispositivos de interfaz de usuario (HID) se

desplegará la opción de Dispositivo compatible con HID, tal como se muestra en la Figura B. 3.

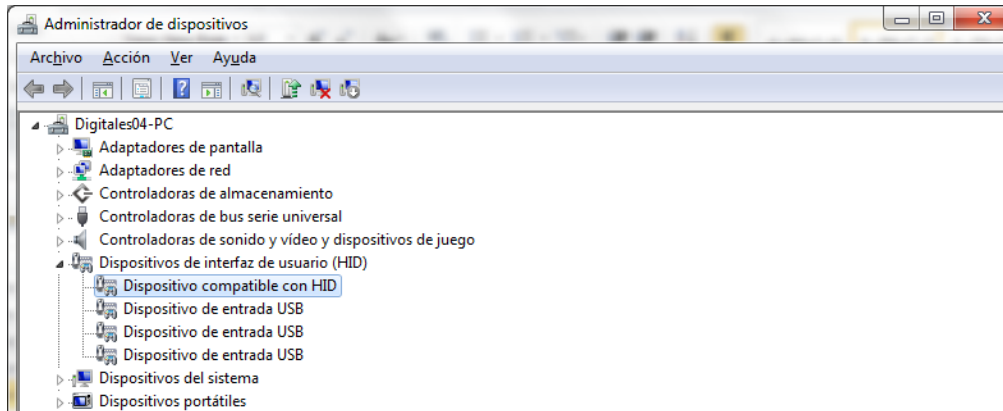


Figura B. 3 Ventana del administrador de dispositivos.

Con un doble click a esta pestaña mostrará una ventana como la de la Figura B. 4, en la que se muestra las características de la TAD. Se puede ver que el VID (03EB) y PID (2FF9) corresponden a los proporcionados por el fabricante del microcontrolador AT90USB646.

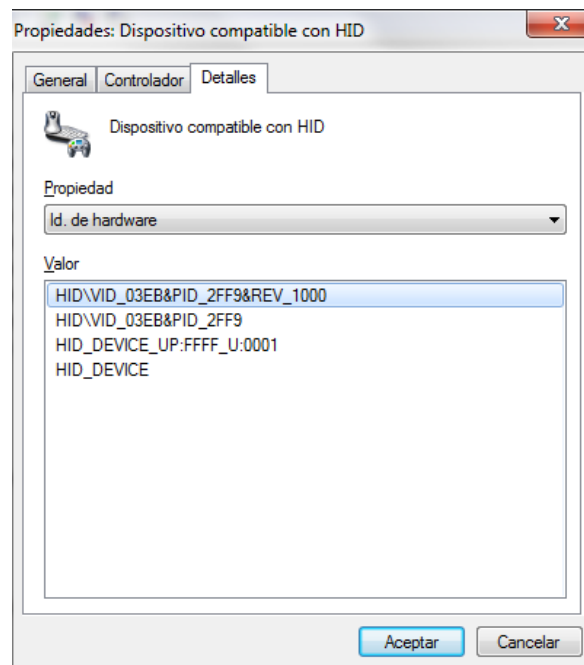


Figura B. 4 Propiedades de la TAD.

Una vez que se ha establecido la comunicación de la TAD con la PC, es posible usar los módulos en LabVIEW. Los módulos se encuentran alojados en un directorio y para agregarlos al VI es necesario abrir la paleta de funciones y seleccionamos la pestaña “Select a

VI...” como se muestra en la Figura B.5. A continuación se abrirá el explorador de Windows para buscar el módulo que se desee usar y agregarlo al VI.

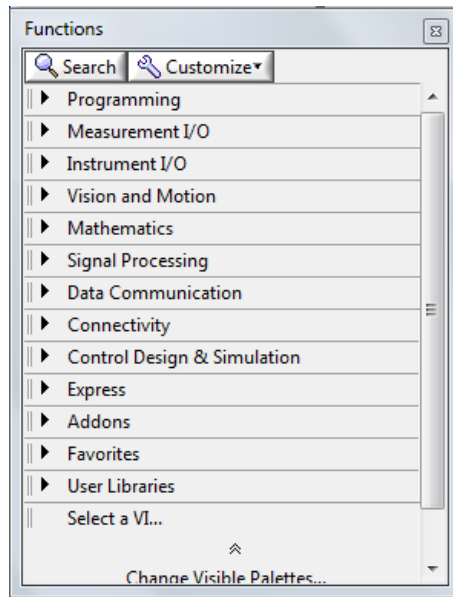


Figura B.5 Paleta de funciones para buscar los módulos de los recursos.

En la Figura B. 6 se muestra el bloque del módulo PWM y la ayuda de dicho módulo.



Figura B. 6 Ayuda del módulo PWM.

Al agregar el módulo al VI, se tendrá una interfaz como la que se muestra en la Figura B.7, la cual contiene un programa para utilizar el módulo PWM.

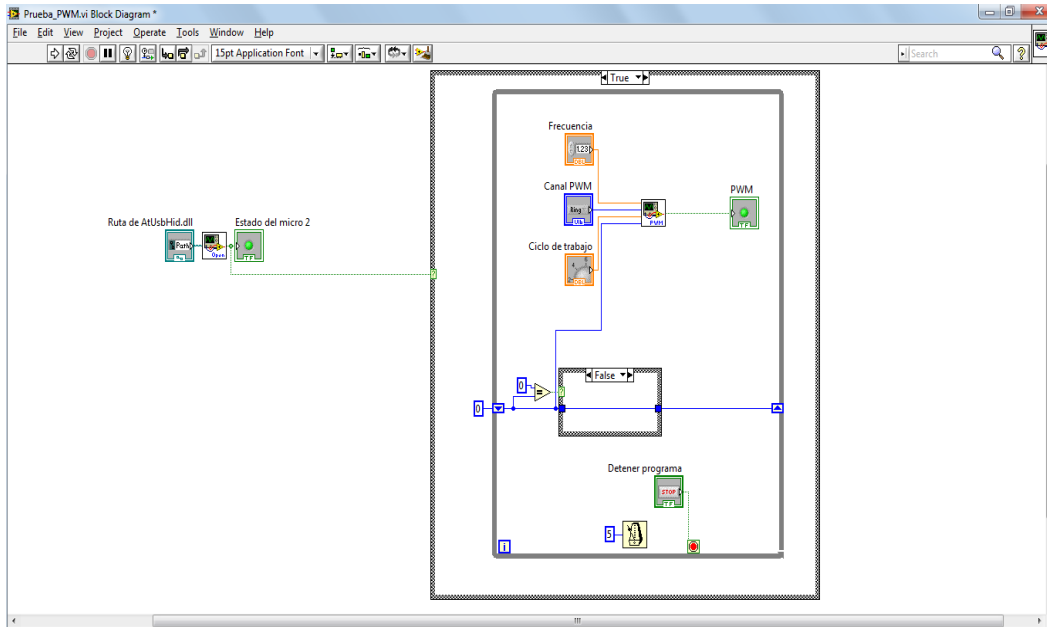


Figura B.7 Código para el módulo PWM.

En la Figura B.8 se muestra la interfaz de usuario del programa para generar salidas PWM en la TAD. Las entradas son:

- Ruta de ATUSBHID.dll, se debe colocar la ruta del archivo ATUSBHID.dll para la comunicación de LabVIEW con el puerto USB.
- La entrada Frecuencia para indicar numéricamente el valor de la frecuencia.
- En la lista Canal PWM se indica el canal que se va a usar A, B o C.
- Para indicar el ciclo de trabajo de la señal PWM se usa una perilla con valores de 0 a 100.
- Se puede detener el programa con el botón STOP.
- Los LEDs Estado del micro y PWM indican si se estableció la comunicación con el puerto y si el PWM está en uso respectivamente.

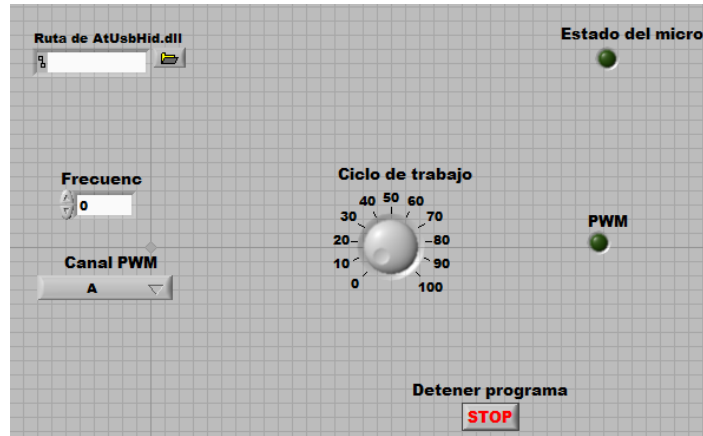


Figura B.8 Interfaz de usuario para el módulo PWM.

Apéndice C Circuitos y componentes empleados

En este apéndice se describe el hardware de la TAD mostrada en la Figura 3.2. El sistema está constituido por 3 PCB's (*Printed Circuit Board*, Placa de Circuito Impresa), las cuales son:

- PCB con las terminales para entrada y salida de las señales
- PCB del microcontrolador AT90USB646
- PCB para el acondicionamiento de la señal

En la Figura C.1 se muestra el diagrama esquemático de la placa base que contiene los componentes para las entradas y salidas de señales.

ADIG, EDIG, PWM y ENTALO son los conectores para la interfaz de salida y entrada de la TAD. El conector VREFDAC es para seleccionar si se desea usar un Voltaje de referencia externo o utilizar el voltaje de la TAD de 5 V.

El conector j1 y j9 es para seleccionar el voltaje de alimentación de la TAD, ya sea que se tome del bus USB o que se tome externo a través de las terminales de entrada. Los integrados U4, U5, y U6 con serie 74LS245, son los buffer de protección de los puertos de entrada/salida digitales. El integrado U3 con serie MCP4922, es el DAC por SPI. El conector J2 contiene las terminales para la programación del micro mediante el protocolo SPI.

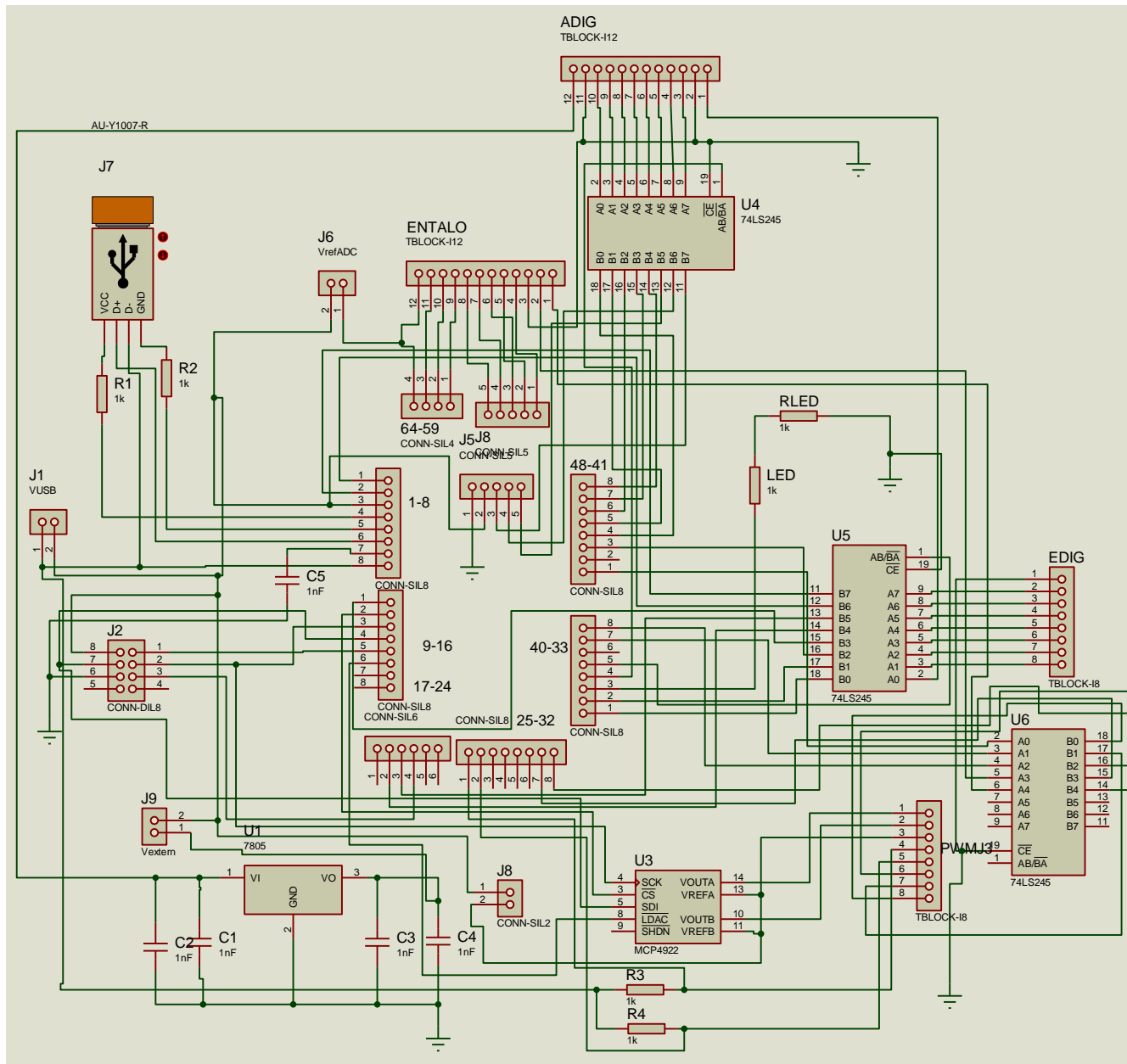


Figura C.1 Diagrama esquemático de la placa base.

En la Tabla C.1 se muestra la descripción de las terminales para la entrada y salida de datos de la placa base.

Tabla C.1 Descripción de las terminales de entrada/salida.

Conector	Número	descripción
EDIG	1	Puerto digital B0
	2,11	GND
	3-10	Puerto digital A
	12	Vcc externo
EDIG	1	GND
	2-8	Puerto digital B7,...,B1
ENALO	1,2	Contador de eventos T0 y T1
	3	GND
	4,5	Entradas analógicas de 0 a 10V
	6,7 y 8	Entradas analógicas de -10 a 10V
	9,10 y 11	Entradas analógicas de 0 a 5V
	12	Vcc
PWM	1 y 2	Salidas DAC A y B
	3	Voltaje de referencia DAC
	4	SCL
	5	SDA
	6,7 y 8	PWM B, C y A

En la Tabla C. 2 se muestra la lista de componentes que se utilizaron en la placa base.

Tabla C. 2 Componentes de la placa base.

Cantidad	Referencia	Descripción	Modelo	Precio por unidad[\$]	Proveedor
4	Con1,Con2,Con3,Con4,Con5,Con6, Con7 y Con8	Conectores hembra para PCB	HOUSING-40	8.6	AG Electrónica
4	ADIG, EDIG, PWM y ENTALO	Conectores de 2 tornillos para PCB	TRTG-02	4.5	AG Electrónica
3	U4,U5 yU6	Buffers de protección	HD74LS245P	19	AG Electrónica
2	J1,J6,J8,J9	Tira de pines con 36 terminales	HEADER-1	1.7	AG Electrónica
1	J2	Tira de pines con 36 terminales, 2 líneas	HEADER-2	6	AG Electrónica
1	J7	Conector USB tipo B	USBB/F90_DIP	8.6	AG Electrónica
2	R1, R2	Resistencia de Carbón 5% Tol ½ Watt 22Ω	RC-22E/1/2	.86	AG Electrónica
2	R3,R4	Resistencia de Carbón 5% Tol ½ Watt 2KΩ	RC-2K2/1/2	.86	AG Electrónica
1	Rled	Resistencia de Carbón 5% Tol ½ Watt 330Ω	RC-330E/1/2	.86	AG Electrónica
1	Led	LED 3MM	LED-B3D-AC-R30	1.72	AG Electrónica
1	U3	Convertidor Digital Analógico con interfaz SPI	MCP4922-E/P	1	Newark
1	U1	Regulador de voltaje 5V 1A	L7805CV	7	AG Electrónica
4	C1,C2,C3 y C4	Capacitor Cerámico .001μF=1nF	CC-.001/50V	2	AG Electrónica

En la Figura C.2 y Figura C.3 se muestra el PCB de la tarjeta base de la TAD y el montaje de los componentes respectivamente.

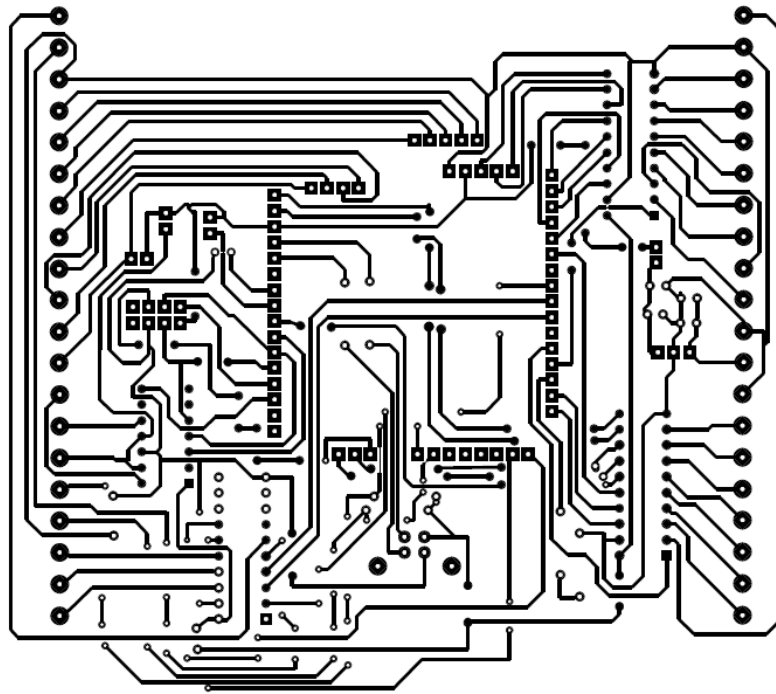


Figura C.2 Diagrama de la placa base de la TAD.

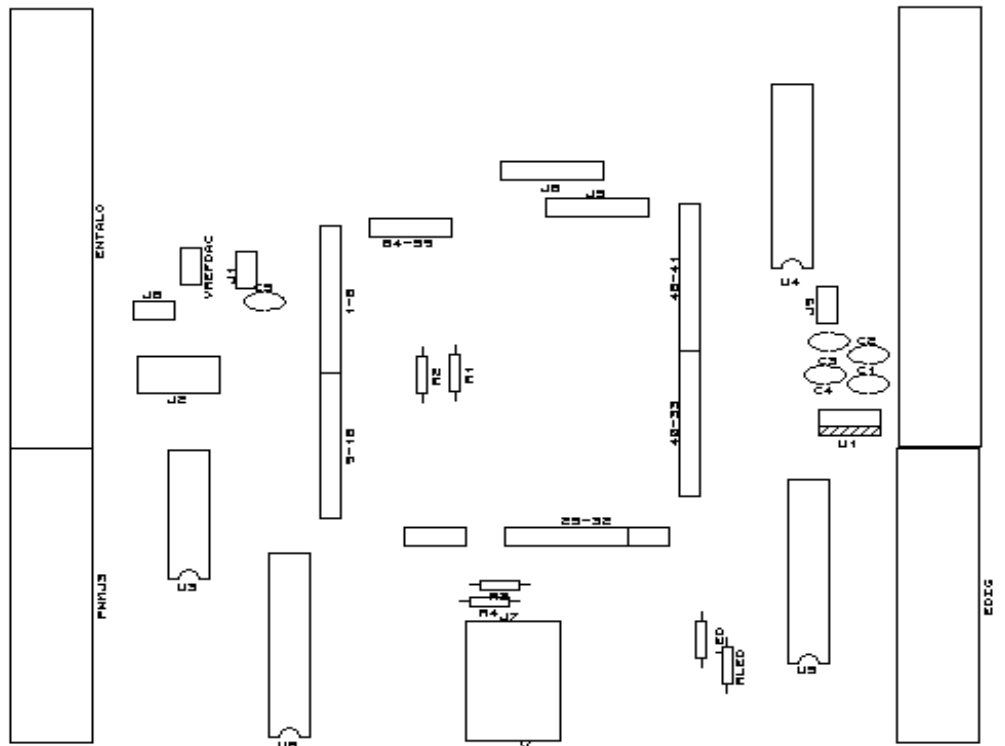


Figura C.3 Montaje de los componentes de la placa base.

En la Figura C. 4 se muestra el diagrama esquemático de la tarjeta del microcontrolador, esta placa va montada sobre la placa base de la Figura C.2.

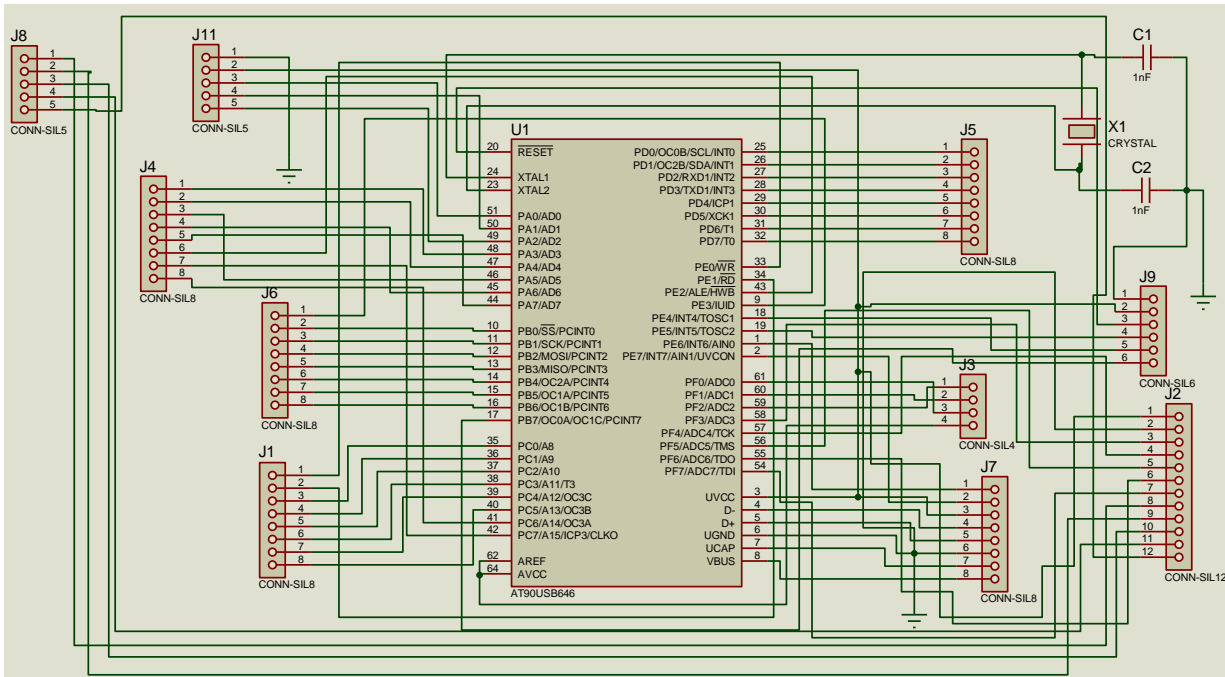


Figura C. 4 Diagrama esquemático para el microcontrolador.

En la Tabla C.3 se muestra la lista de componentes de la placa del microcontrolador AT90USB646.

Tabla C.3 Lista de componentes de la placa del microcontrolador.

Cantidad	Referencia	Descripción	Modelo	Precio por Unidad [\$]	Proveedr
9	J1,J3,J4,J5,J6, J7,J8,J9,J10	Tira de pines con 36 terminales	HEADER-1	1.7	AG Electrónica
1	J2	Conectores hembra para PCB	HOUSING-40	8.6	AG Electrónica
2	C1,C2	Capacitor Cerámico .001µF=1nF	CC-.001/50V	2	AG Electrónica
1	Cristal	OSCILADOR DE 8MHZ	OSC-8MHZ	25.8	AG Electrónica

En la Figura C.5 se muestra el PCB de la tarjeta del microcontrolador AT90USB646

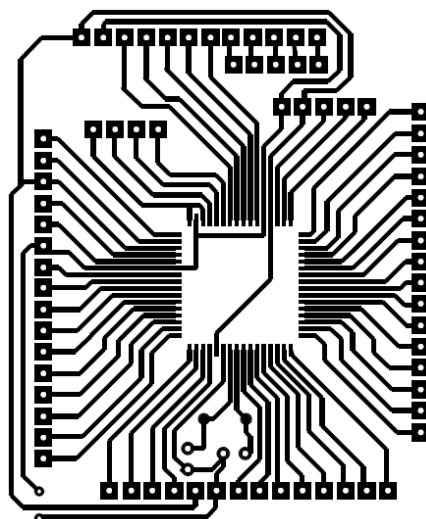


Figura C.5 PCB del microcontrolador.

El diagrama esquemático del circuito para el acondicionamiento de las señales analógicas de entrada se muestra en la Figura C.6 y en la Tabla C. 4 se muestra la lista de los componentes utilizados.

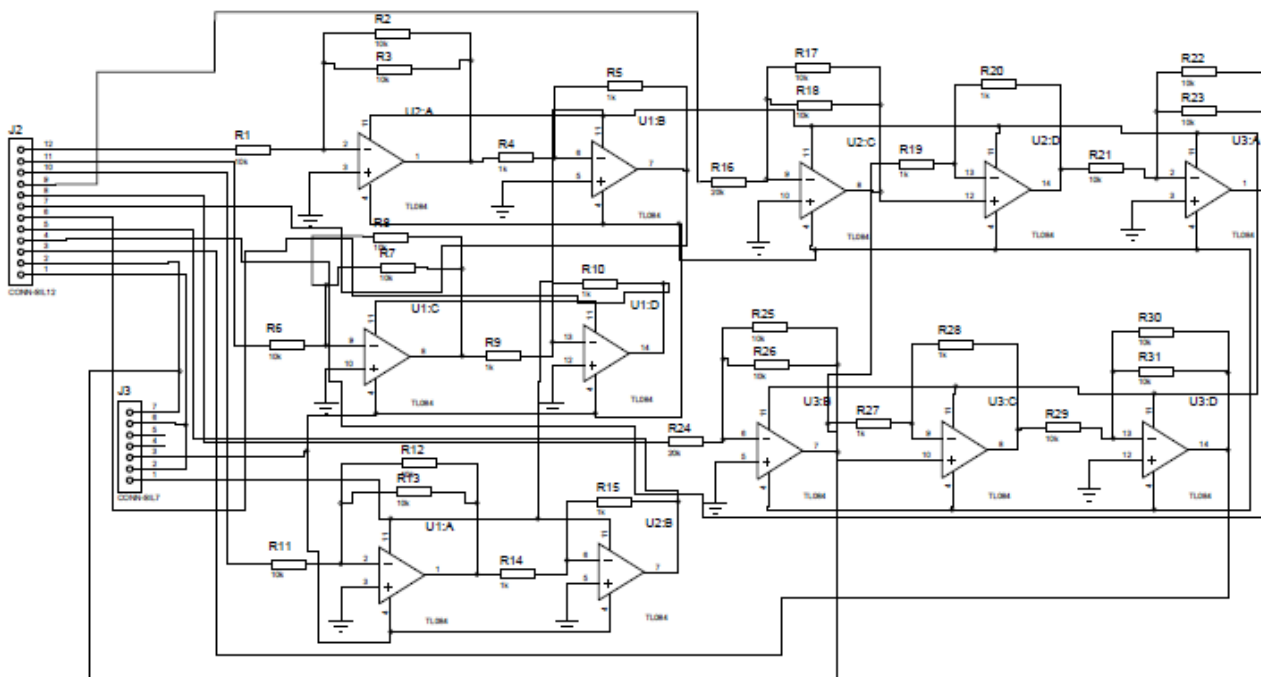


Figura C.6 Diagrama esquemático del circuito para el acondicionamiento de señales analógicas.

Tabla C. 4 Lista de componentes de la placa para el acondicionamiento de señal.

Cantidad	Referencia	Descripción	Modelo	Precio por unidad[\$]	Proveedor
3	U1, U2 y U3	Amplificador Operacional cuádruple.	TL084CD	5.17	AG Electrónica
17	R(1,2,3,6,7,8,11,12,13,17,18,22,23,25,26,30,31)	Resistencia de Carbón 5% Tol ½ Watt 10KΩ	RC-10K/1/2	.86	AG Electrónica
10	R(4,5,9,10,14,15,19,20,27,28)	Resistencia de Carbón 5% Tol ½ Watt 1KΩ	RC-1K/1/2	.86	AG Electrónica
4	R(16,21,24,29)	Resistencia de Carbón 5% Tol ½ Watt 20KΩ	RC-20K/1/2	.86	AG Electrónica
2	J2,j3	Tira de pines con 36 terminales	HEADER-1	1.7	AG Electrónica

En la Figura C.7 y en la Figura C.8 se muestra el diagrama de pistas del circuito para el acondicionamiento de señales analógicas.

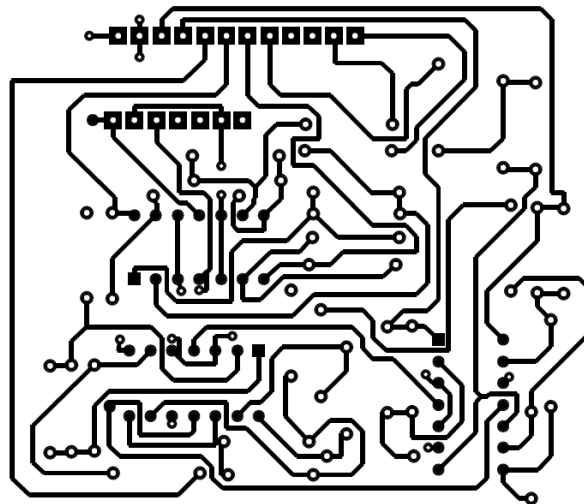


Figura C.7 PCB del lado opuesto al de los componentes-

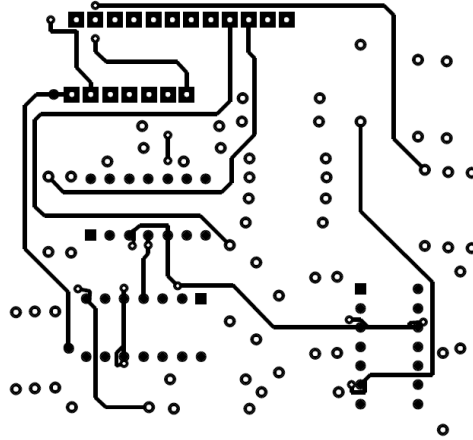


Figura C.8 PCB del lado de los componentes.

En la Figura C.9 se muestra el diagrama de montajes de componentes del circuito de acondicionamiento de señal

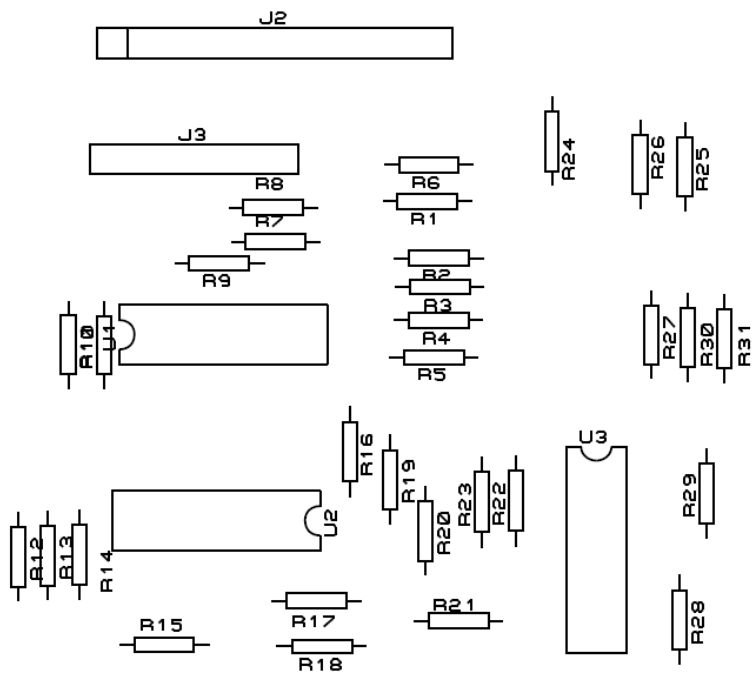


Figura C.9 Diagrama de montaje de componentes.

