



**ENTRENADOR DIGITAL USB CON  
INTERFAZ EN LABVIEW**

CARLOS AUGUSTO RIOS ARDILA  
RICARDO RODRIGUEZ GUEVARA

**UNIVERSIDAD PONTIFICIA BOLIVARIANA  
Seccional BUCARAMANGA  
ESCUELA DE INGENIERÍAS Y ADMINISTRACION  
INGENIERIA ELECTRONICA  
BUCARAMANGA  
2007**



**ENTRENADOR DIGITAL USB CON  
INTERFAZ EN LABVIEW**

CARLOS AUGUSTO RIOS ARDILA  
RICARDO RODRIGUEZ GUEVARA

Trabajo de Grado para optar al título de  
Ingeniero Electrónico

**Director**  
ADOLFO TORRES CALDERON  
Ing. Electricista Ing. Sistemas

**UNIVERSIDAD PONTIFICIA BOLIVARIANA  
Seccional BUCARAMANGA  
ESCUELA DE INGENIERÍAS Y ADMINISTRACION  
INGENIERIA ELECTRONICA  
BUCARAMANGA  
2007**

**Nota de aceptación**

---

---

---

---

---

**Presidente del Jurado**

---

**Jurado**

---

**Jurado**

Ciudad y fecha:

A nuestras familias, por el acompañamiento y apoyo incondicional, que siempre nos brindaron.

A Dios por habernos iluminado en los momentos difíciles de este proyecto.

## **AGRADECIMIENTOS**

A Dios, por habernos otorgado la fuerza de voluntad necesaria, para seguir adelante en este proyecto.

A nuestras familias, por el apoyo incondicional prestado, así como su voz de aliento y apoyo en momentos de desesperación.

A nuestro Director de Proyecto, por haber confiado en nosotros y habernos brindado la oportunidad de realizar este gran proyecto.

## RESUMEN

Desde hace tiempo y en la actualidad ha surgido la necesidad de buscar el desarrollo de herramientas académicas capaces de proporcionar a la comunidad universitaria (estudiantes de ingeniería), los espacios de crecimiento integral y desarrollo profesional. Inicialmente, la innovación tecnológica que introduce la Electrónica vista desde el ángulo de la educación, es una oportunidad y un reto. Teniendo presente lo anterior, el proyecto de grado, persigue, desarrollar un entrenador digital para prácticas de laboratorios de ingeniería electrónica, con interfaz software en Labview, y comunicado a un PC por el Bus Serial Universal U.S.B.

Mediante programas como "Aspire", "TUSB3410 Windows VCP" y "Labview", se realiza la programación, la transmisión y la recepción de datos. Este software programa las funciones del microcontrolador, genera el driver del dispositivo y aplicaciones de acceso de hardware, y comunica el usuario con el dispositivo. Es importante aclarar que el entrenador digital es un dispositivo de plataforma abierta, es decir, se puede implementar bajo otros ambientes de trabajo disponibles en la universidad, como por ejemplo MATLAB.

El trabajo central consiste en el desarrollo de un entrenador digital U.S.B. con interfaz Labview. Se emplea el algoritmo propuesto por el ADUC841 la arquitectura 8052 que con la ayuda de los registros y el set de instrucciones programan las diferentes tareas que realiza el entrenador digital donde están caracterizadas en los módulos de I/O, AD, DA, Comunicación, PWM, y el contador de eventos.

Los datos recibidos o enviados por el usuario serán canalizados por el dispositivo TUSB3410 el cual es un puente entre el entrenador y el PC donde el dispositivo contiene en la board un microcódigo de ROM que le permite al MCU que enumere el dispositivo como un periférico USB y así permita el flujo de datos. Estos Datos se configuran por el usuario en la interfaz de Labview permitiéndole el desarrollo libre de la aplicación que desee trabajar. Esta comunicación se logra con la ayuda del recurso de Labview VISA.

## **ABSTRACT**

At present there is a need to pursue development of tools to provide the academic community of engineering students, spaces comprehensive growth and professional development. Technological innovation introduced by the electronic viewed from the angle of education, is an opportunity and a challenge. With this in mind, the proposed project, aims to develop a digital trainer laboratory practices for electronic engineering, software interface with Labview, and communicated to a PC by USB Universal Serial Bus.

Through programs like "Aspire", "TUSB3410 Windows VCP" and "Labview", is performed programming, transmission and reception of data. This software program functions of the microcontroller, the device generates the driver and hardware access applications, and communicates the user with the device. It is important to clarify that the trainer is a digital device open platform; this can be implemented under other work environments, such as MATLAB.

The central work is the development of a digital USB trainer with Labview interface. It uses the algorithm proposed by the ADUC841 architecture 8052 with the help of the records and the set of instructions programmed the various tasks carried out by the coach where digital modules are characterized in the I / O, DP, DA, Communication, PWM, and the events counter.

The data received or sent by the user will be channeled by the TUSB3410 device which is a bridge between the trainer and the PC where the device contains a board in the microcode ROM that allows the MCU to list the device as a USB peripheral and thus allowing the flow of data. These data are configured by the user interface Labview allowing the free development of the application you want to work.

## OBJETIVOS

### OBJETIVO GENERAL

- Diseñar y construir un entrenador digital con puerto U.S.B e interfaz de configuración y programación en “Labview”.

### OBJETIVOS ESPECIFICOS

- Diseñar y construir el circuito con interfaces para los puertos disponibles en el Microcontrolador ADUC841 del fabricante Analog Devices.
- Diseñar y construir el circuito y software de control que permita comunicar el Microcontrolador ADUC841 con el puerto USB del PC utilizando un microcontrolador TUSB3410.
- Diseñar e Implementar un software en Labview que permita configurar y programar las diferentes funciones del entrenador digital a través del puerto USB.
- Desarrollar una práctica “demo” y la guía de laboratorio de la misma que permita mostrar la funcionalidad de cada uno de los módulos disponibles en el entrenador digital.

## TABLA DE CONTENIDO

	<b>Pág.</b>
<b>INTRODUCCION.....</b>	<b>17</b>
<b>1. MARCO TEORICO.....</b>	<b>19</b>
<b>1.1 FUNDAMENTOS DE LOS MICROCONTROLADORES.....</b>	<b>19</b>
<b>1.2. BUS SERIAL UNIVERSAL USB.....</b>	<b>29</b>
<b>1.2.1 Funcionamiento.....</b>	<b>29</b>
<b>1.2.2 Características Generales.....</b>	<b>30</b>
<b>1.2.3 Características de Transmisión.....</b>	<b>30</b>
<b>1.2.4 Conectores USB Serie "A" Y Serie "B".....</b>	<b>32</b>
<b>1.2.5 Requerimientos para Desarrollar una Aplicación USB.....</b>	<b>33</b>
<b>1.3 LENGUAJE GRAFICO (Lenguaje G).....</b>	<b>34</b>
<b>2. DESARROLLO DEL ENTRENADOR DIGITAL USB.....</b>	<b>35</b>
<b>2.1 Descripción General.....</b>	<b>36</b>
<b>2.2. HARDWARE.....</b>	<b>37</b>
<b>2.2.1 Descripción.....</b>	<b>37</b>
<b>2.2.2 Modulo de Fuentes.....</b>	<b>37</b>
<b>2.2.2.1 Diseño del Circuito.....</b>	<b>38</b>
<b>2.2.3 Modulo del Microcontrolador ADUC 841.....</b>	<b>40</b>
<b>2.2.4 Periféricos.....</b>	<b>41</b>
<b>2.2.4.1 Convertidor Análogo/Digital (ADC).....</b>	<b>41</b>
<b>2.2.4.1.1 Modo DMA.....</b>	<b>43</b>

2.2.4.2 Convertidor Digital/Análogo (DAC).....	43
2.2.4.3 Modulación por Amplitud de Pulso (PWM).....	44
2.2.4.3.1 Modos de Operación.....	45
2.2.4.4 Puertos.....	47
2.2.4.5 Contador de Eventos.....	48
2.2.5. Diseño del Circuito del Entrenador Digital USB.....	50
2.2.5.1 Diseño del Circuito Impreso.....	51
2.2.6 Modulo USB (Microcontrolador TUSB3410).....	52
2.2.6.1 Diseño del Circuito del Módulo USB.....	54
2.2.6.2 Diseño del Circuito Impreso.....	55
2.3 SOFTWARE.....	55
2.3.1 Aspire.....	56
2.3.2 Labview.....	57
2.3.3 FIRMWARE (Software del ADUC y TUSB3410).....	60
2.3.3.1 Diagramas de Bloques de la Programación.....	61
2.3.3.2 Interfaz de Usuario en LABVIEW.....	68
3. RESULTADOS OBTENIDOS.....	74
3.1 Capa Software.....	74
3.2 Capa Hardware.....	92
3.3 Etapas en el Desarrollo del Entrenador Digital USB.....	94
4. CONCLUSIONES Y PERSPECTIVAS.....	98
BIBLIOGRAFÍA.....	99

## INDICE DE FIGURAS

	<b>Pág.</b>
Figura 1. Estructura básica de un microcontrolador.....	19
Figura 2. Configuraciones de Memoria en un Microcontrolador.....	23
Figura 3. Conectores USB.....	32
Figura 4. Diagrama de Bloques de Hardware del Entrenador Digital.....	38
Figura 5. Fuente CC3 – 1205SS-E.....	38
Figura 6. Diagrama del Circuito de la Fuente CC3 – 1205SS-E.....	38
Figura 7. Diagrama de Bloques de Alimentación por el Puerto USB.....	39
Figura 8. Diagrama del Circuito de Alimentación por el Puerto USB.....	39
Figura 9. Diagrama de Bloques Funcional.....	40
Figura 10. Estructura Interna del ADC.....	42
Figura 11. Arquitectura del DAC.....	44
Figura 12. Diagrama de Bloques del PWM.....	45
Figura 13. PWM Modo 1.....	46
Figura 14. PWM Modo 2.....	46
Figura 15. PWM Modo 3.....	47
Figura 16. PWM Modo 4.....	47
Figura17. Diagrama de Bloques del Contador de 16 Bits.....	49
Figura 18. Diagrama del circuito del Microcontrolador ADUC 841.....	50
Figura 19. Cara Superior del Circuito Impreso (Bloque Microcontrolador ADUC 841 y Fuente de Alimentación).....	51

Figura 20. Cara Inferior del Circuito Impreso (Bloque Microcontrolador ADUC 841 y Fuente de Alimentación).....	51
Figura 21. Cara Superior e Inferior del Circuito Impreso (Bloque Microcontrolador ADUC 841 y Fuente de Alimentación).....	52
Figura 22. Flujo de Datos a través del Microcontrolador TUSB3410.....	53
Figura 23. Diagrama del Circuito del Microcontrolador TUSB3410 (Bloque USB).....	54
Figura 24. Cara Superior del Circuito Impreso (Bloque Microcontrolador TUSB3410 y Fuente de Alimentación por el Puerto USB).....	55
Figura 25. Cara Inferior del Circuito Impreso (Bloque Microcontrolador TUSB3410 y Fuente de Alimentación por el Puerto USB).....	55
Figura 26. Ejemplo de panel frontal.....	57
Figura 27. Ejemplo de Diagrama de Bloque y su correspondiente Panel Frontal.....	58
Figura 28. Ejemplo de un Terminal.....	58
Figura 29. Ejemplo de Icono.....	59
Figura 30. Ejemplo de Panel Conector.....	60
Figura31. Diagrama de Bloques de ADC DMA.....	61
Figura32. Diagrama de Bloques DAC.....	62
Figura33. Diagrama de Bloques de Contador de Eventos.....	63
Figura34. Diagrama de Bloques Escritura del Puerto 8 Bits.....	64
Figura35. Diagrama de Bloques Escritura del Puerto 16 Bits.....	65
Figura36. Diagrama de Bloques Lectura del Puerto.....	66
Figura37. Diagrama de Bloques PWM.....	67
Figura38. Interfaz de Menú Principal.....	68

Figura39. Interfaz de Usuario PWM Modo 1.....	69
Figura40. Ejemplo Mensaje de Error PWM Modo 1.....	70
Figura41. Interfaz de Usuario PWM Modo 2.....	71
Figura42. Ejemplo Mensaje de Error PWM Modo 2.....	73
Figura43. Interfaz de Usuario PWM Modo 3.....	74
Figura44. Ejemplo Mensaje de Error PWM Modo 3.....	75
Figura45. Interfaz de Usuario PWM Modo 4.....	76
Figura46. Ejemplo Mensaje de Error PWM Modo 4.....	77
Figura47. Interfaz de Usuario ADC en modo DMA.....	78
Figura48. Interfaz de Usuario de Puertos 8 Bits.....	79
Figura49. Interfaz de Usuario de Puertos 16 Bits.....	80
Figura50. Interfaz de Usuario de Contador de Eventos 16 Bits.....	81
Figura51. Interfaz de Usuario de Lectura del Puerto.....	82
Figura52. Interfaz de Usuario de DAC 12 Bits.....	82
Figura53. Tabla de Muestras1 Onda Seno.....	85
Figura54. Tabla de Muestras2 Onda Seno.....	86
Figura55. Muestras Onda Diente de Sierra.....	86
Figura56. Muestras Onda Triangular.....	87
Figura57. Tabla de Función de Transferencia ADC.....	88
Figura58. Tabla de Frecuencias PWM COUNTER con Preescaler de 1.....	89
Figura59. Tabla de Frecuencias PWM COUNTER con Preescaler de 4.....	90

Figura60. Tabla de Frecuencias PWM COUNTER con Preescaler de 16.....	90
Figura61. Tabla de Frecuencias PWM COUNTER con Preescaler de 64.....	91
Figura62. Primer Prototipo del Entrenador Digital USB.....	94
Figura63. Módulo del Microcontrolador ADuC841.....	95
Figura64. Módulo de Comunicación del Microcontrolador TUSB3410 USB –UART.....	95
Figura65. Módulos del Entrenador Digital USB.....	95
Figura66. Pruebas del Entrenador.....	96
Figura67. Señales de Prueba del DAC del Entrenador Digital USB.....	96
Figura68. Entrenador Digital USB Final.....	97

## INDICE DE TABLAS

	<b>Pág.</b>
Tabla 1. Pines de Cable USB.....	31
Tabla 2. Características Generales de la Fuente CC3 – 1205SS-E.....	38

## ANEXOS

	<b>Pág.</b>
A. Manual de Usuario para Instalar el Entrenador Digital USB.....	100
B. Características del Nucleo8052.....	107
C. Driver/Firmware Virtual Com Port (VCP).....	153
D. Tabla de Excel – Función de Transferencia ADC.....	154
E. Tabla de Excel PWM COUNTER.....	154
F. Guía de Laboratorio Práctica “Demo”.....	155

## INTRODUCCION

La electrónica avanza muy rápidamente al igual que el interés de los estudiantes de la ingeniería electrónica, por ello en la actualidad se han innovado muchas formas de incentivar al estudiante, entre ellas el entrenador digital. El propósito de los entrenadores es familiarizar al estudiante con el comportamiento de las variables para las cuales fueron desarrollados. En este proyecto se incorporan dos microcontroladores que nos permiten programar las funciones del entrenador y comunicar al usuario con el mismo, estos son el ADUC841 y el TUSB3410.

Con el Microcontrolador ADUC 841 se realizan las operaciones de los módulos del entrenador, el que se encarga de procesar y analizar los datos ingresados u obtenidos por el usuario, y el Microcontrolador TUSB3410 se encarga de transmitir o recibir la información requerida por el usuario; como resultado se obtiene una herramienta con el cual el estudiante interactúa con el entrenador a través del interfaz de Labview ingresando u obteniendo datos de circuitos o aplicaciones electrónicas que el estudiante este trabajando.

El capítulo 1 muestra un marco teórico con los principales conceptos y términos utilizados para la realización de la tesis. Este marco teórico contiene desde fundamentos de los microcontroladores (características principales, estructura básica y composición), teoría de USB (Universal serie bus), conceptos y características del lenguaje G usado en Labview.

El capítulo 2 muestra la descripción de la tesis realizada, además de contiene las características de hardware y software del entrenador, sus componentes y la explicación del funcionamiento de cada uno. Además contiene los diagrama circuitales y de impreso del entrenador

El capítulo también contiene la explicación de las diferentes rutinas del programa de las diferentes funciones del entrenador así como también los pasos para la configuración del puerto, instalación de la aplicación del TUSB3410 para el entrenador.

Este capítulo contiene la Interfaz Gráfica del Usuario, allí se especifica la funcionalidad de cada componente de la GUI diseñada para el análisis de los datos del entrenador digital.

El capítulo 3 contiene los resultados obtenidos de las implementaciones del hardware. Se muestran los eventos especiales ocurridos durante el montaje del entrenador.

En el capítulo 4 se muestran las conclusiones y las perspectivas a futuro del trabajo realizado. Este capítulo contiene un análisis detallado para los módulos del

entrenador trabajado en esta tesis, así como también unas recomendaciones para el continuo desarrollo y mejoramiento del entrenador digital.

## CAPITULO 1. MARCO TEORICO

### 1.1 FUNDAMENTOS DE LOS MICROCONTROLADORES

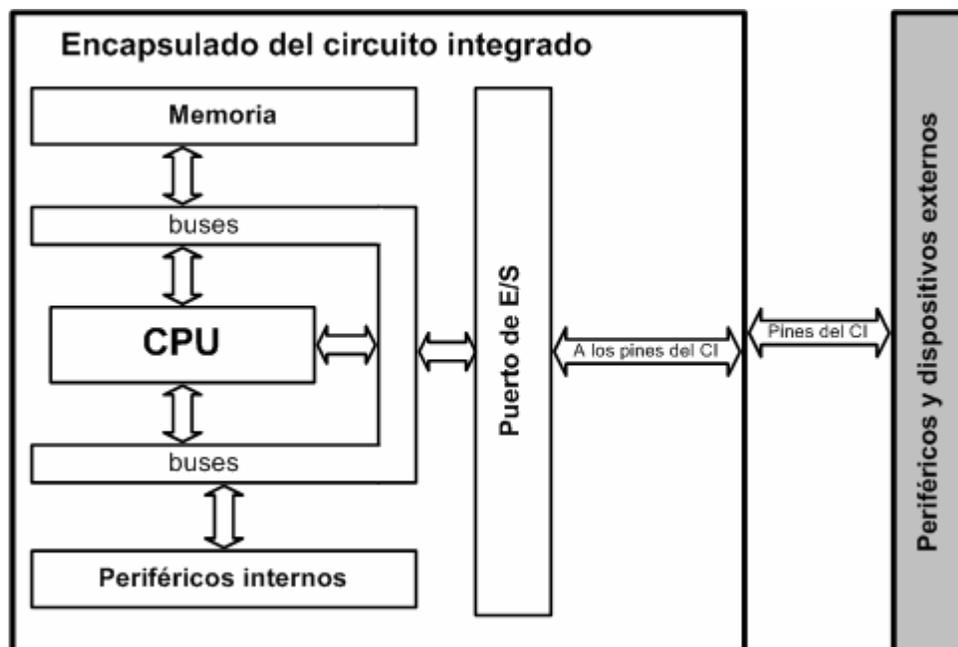
El microcontrolador no es más que una computadora completa, con procesador, memoria, puertos de entrada/salida (E/S) y periféricos, todo encapsulado o embebido dentro de un circuito integrado

La estructura básica de un microcontrolador consta:

- Procesador
- Buses
- Memoria de programas
- Memoria de datos
- Periféricos
- Puertos de E/S

Como se muestra en la figura:

Figura: 1. Estructura básica de un microcontrolador



Fuente: Autores

En los microcontroladores, existen dos tipos de memoria bien definidas: memoria de datos (típicamente algún tipo de SRAM) y memoria de programas (ROM, PROM, EEPROM, FLASH u de otro tipo no volátil). La memoria está segregada y el acceso a cada tipo de memoria depende de las instrucciones del procesador. También son usados los registros, una unidad de control, buses, conjunto de instrucciones, interrupciones y los periféricos.

#### *Núcleo de un microcontrolador*

Aún cuando el microcontrolador es circuito integrado, se compone de un núcleo y un conjunto de circuitos adicionales. Dentro del núcleo se encuentran el procesador y la memoria.

#### *Procesador*

El procesador es el encargado de ejecutar las instrucciones y procesar los datos que son necesarios para todas las funciones del microcontrolador.

#### *Registros*

Son un espacio de memoria muy reducido de aquí se toman los datos para varias operaciones que debe realizar el resto de los circuitos del procesador. Los registros sirven para almacenar los resultados de la ejecución de instrucciones, cargar datos desde la memoria externa o almacenarlos en ella.

#### *Unidad de control*

Esta unidad es de las más importantes en el procesador, en ella recae la lógica necesaria para la decodificación y ejecución de las instrucciones, el control de los registros, la ALU, los buses y cuanta más información se quiera guardar dentro del procesador.

La unidad de control es uno de los elementos fundamentales que determinan las prestaciones del procesador, ya que su tipo y estructura, determina parámetros tales como el tipo de conjunto de instrucciones, velocidad de ejecución, tiempo del ciclo de máquina, tipo de buses que puede tener el sistema, manejo de interrupciones y un buen número de cosas más que en cualquier procesador van a parar a este bloque.

#### *Buses*

Son el medio de comunicación que utilizan los diferentes componentes del procesador para intercambiar información entre sí.

Existen tres tipos de buses:

- Dirección: Se utiliza para seleccionar al dispositivo con el cual se quiere trabajar o en el caso de las memorias, seleccionar el dato que se desea leer o escribir.
- Datos
- Control: Se utiliza para gestionar los distintos procesos de escritura lectura y controlar la operación de los dispositivos del sistema.

### *Conjunto de instrucciones*

Define las operaciones básicas que puede realizar el procesador, que conjugadas y organizadas forman el software; el elemento básico del lenguaje, que organizadas adecuadamente permiten escribir palabras, oraciones y cualquier programa que se desee realizar.

### *Memoria*

Como se nombro anteriormente existen dos tipos de memoria la memoria de datos y la memoria de programas.

La memoria RAM está destinada al almacenamiento de información temporal (memoria de datos) que será utilizada por el procesador para realizar cálculos u otro tipo de operaciones lógicas. En el espacio de direcciones de memoria RAM se ubican además los registros de trabajo del procesador y los de configuración y trabajo de los distintos periféricos del microcontrolador. En la mayoría de los casos, aunque se tenga un espacio de direcciones de un tamaño determinado, la cantidad de memoria RAM de que dispone el programador para almacenar sus datos es menor que la que puede direccionar el procesador.

El tipo de memoria utilizada en las memorias RAM de los microcontroladores es SRAM, lo que evita tener que implementar sistemas de refrescamiento como en el caso de las computadoras personales, que utilizan gran cantidad de memoria, típicamente alguna tecnología DRAM. A pesar de que la memoria SRAM es más costosa que la DRAM, es el tipo adecuado para los microcontroladores porque éstos poseen pequeñas cantidades de memoria RAM.

En el caso de la memoria de programas se utilizan diferentes tecnologías, y el uso de una u otra depende de las características de la aplicación a desarrollar, a continuación se describen algunas tecnologías existentes.

- ROM con máscara

Es una memoria no volátil de sólo lectura cuyo contenido se graba durante la fabricación del chip. El elevado coste del diseño de la máscara sólo hace aconsejable el empleo de los microcontroladores con este tipo de memoria cuando se precisan cantidades superiores a varios miles de unidades.

- OTP

El microcontrolador contiene una memoria no volátil de sólo lectura "programable una sola vez" por el usuario. OTP (One Time Programmable). Es el usuario quien puede escribir el programa en el chip mediante un sencillo grabador controlado por un programa desde un PC.

La versión OTP es recomendable cuando es muy corto el ciclo de diseño del producto, o bien, en la construcción de prototipos y series muy pequeñas.

Tanto en este tipo de memoria como en la EPROM, se suele usar la encriptación mediante fusibles para proteger el código contenido.

- EPROM Erasable Programmable Read Only Memory. Los microcontroladores con este tipo de memoria son muy fáciles de identificar porque su encapsulado es de cerámica y llevan encima una ventanita de vidrio desde la cual puede verse la oblea de silicio del microcontrolador.

Se fabrican así porque la memoria EPROM es reprogramable, pero antes debe borrarse, y para ello hay que exponerla a una fuente de luz ultravioleta, el proceso de grabación es similar al empleado para las memorias OTP. Al aparecer tecnologías menos costosas y más flexibles, como las memorias EEPROM y FLASH, este tipo de memoria han caído en desuso, se utilizaban en sistemas que requieren actualizaciones del programa y para los procesos de desarrollo y puesta a punto.

- EEPROM Electrical Erasable Programmable Read Only Memory. Fueron el sustituto natural de las memorias EPROM, la diferencia fundamental es que pueden ser borradas eléctricamente, por lo que la ventanilla de cristal de cuarzo y los encapsulados cerámicos no son necesarios.

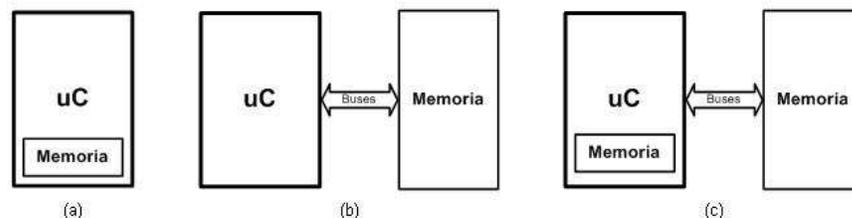
Al disminuir los costos de los encapsulados, los microcontroladores con este tipo de memoria se hicieron más baratos y cómodos para trabajar que sus equivalentes con memoria EPROM. Otra característica destacable de este tipo de microcontrolador es que fue en ellos donde comenzaron a utilizarse los sistemas

de programación en circuito o ICSP (In Circuit Serial Programming) que evitan tener que sacar el microcontrolador de la tarjeta que lo aloja para hacer actualizaciones al programa.

- FLASH. En el campo de las memorias reprogramables para microcontroladores, son el último avance tecnológico en uso a gran escala, y han sustituido a los microcontroladores con memoria EEPROM.

A las ventajas de las memorias FLASH se le adicionan su gran densidad respecto a sus predecesoras lo que permite incrementar la cantidad de memoria de programas a un costo muy bajo. Pueden además ser programadas con las mismas tensiones de alimentación del microcontrolador, el acceso en lectura y la velocidad de programación es superior, disminución de los costos de producción, entre otras. Lo más habitual es encontrar que la memoria de programas y datos está ubicada toda dentro del microcontrolador, de hecho, actualmente son pocos los microcontroladores que permiten conectar memoria de programas en el exterior del encapsulado. Las razones para estas "limitaciones" están dadas porque el objetivo fundamental es obtener la mayor integración posible y conectar memorias externas consume líneas de E/S que son uno de los recursos más preciados de los microcontroladores. A pesar de lo anterior existen familias como la INTEL 51 cuyos microcontroladores tienen la capacidad de ser expandidos en una variada gama de configuraciones para el uso de memoria de programas externa. La Figura 2 muestra algunas de las configuraciones para memoria de programa que podemos encontrar en los microcontroladores. La configuración (a) es la típica y podemos encontrarla casi en el 100% de los microcontroladores. La configuración (b) es poco frecuente y generalmente se logra configurando al microcontrolador para sacrificar la memoria de programas interna, sin embargo el 8031 de INTEL es un microcontrolador sin memoria de programas interna. La configuración (c) es la que se encuentra habitualmente en los microcontroladores que tienen posibilidades de expandir su memoria de programas como algunos PIC de gama alta.

Figura 2. Configuraciones de Memoria en un Microcontrolador



Fuente: Autores

Cuando se requiere aumentar la cantidad de memoria de datos, lo más frecuente es colocar dispositivos de memoria externa en forma de periféricos, de esta forma se pueden utilizar memorias RAM, FLASH o incluso discos duros como los de las PC, mientras que para los cálculos y demás operaciones que requieran almacenamiento temporal de datos se utiliza la memoria RAM interna del microcontrolador. Esta forma de expandir la memoria de datos está determinada, en la mayoría de los casos, por el tipo de repertorio de instrucciones del procesador y porque permite un elevado número de configuraciones distintas, además del consiguiente ahorro de líneas de E/S que se logra con el uso de memorias con buses de comunicación serie.

### *Interrupciones*

- En términos generales, un proceso de interrupción y su atención por parte del procesador, tiene la siguiente secuencia de acciones.
- En el mundo real se produce el evento para el cual queremos que el procesador ejecute un programa especial, este proceso tiene la característica de que no puede esperar mucho tiempo antes de ser atendido o no sabemos en qué momento debe ser atendido.
- El circuito encargado de detectar la ocurrencia del evento se activa, y como consecuencia, activa la entrada de interrupción del procesador.
- La unidad de control detecta que se ha producido una interrupción y “levanta” una bandera para registrar esta situación; de esta forma si las condiciones que provocaron el evento desaparecen y el circuito encargado de detectarlo desactiva la entrada de interrupción del procesador, ésta se producirá de cualquier modo, porque ha sido registrada.
- La unidad de ejecución termina con la instrucción en curso y justo antes de comenzar a ejecutar la siguiente, comprueba que se ha registrado una interrupción.
- Se realiza un proceso que permite guardar el estado actual del programa en ejecución y salta a una dirección especial de memoria de programas, donde está la primera instrucción de la subrutina de atención a interrupción.
- Se ejecuta el código de atención a interrupción, esta es la parte “consciente” de todo el proceso porque es donde se realizan las acciones propias de la atención a la interrupción y el programador juega su papel.

- Cuando en la subrutina de atención a interrupción se ejecuta la instrucción de retorno, se desencadena el proceso de restauración del procesador al estado en que estaba antes de la atención a la interrupción.

El manejo de las interrupciones es bastante complejo, sin embargo tiene dos ventajas que obligan a su implementación: la velocidad y su capacidad de ser asincrónico. Ambas en conjunto permiten que aprovechemos al máximo las capacidades de trabajo de nuestro procesador.

Los mecanismos de interrupción no solo se utilizan para atender eventos ligados a procesos que requieren atención inmediata sino que se utilizan además para atender eventos de procesos asincrónicos.

Las interrupciones son tan eficaces que permiten que el procesador actúe como si estuviese haciendo varias cosas a la vez cuando en realidad se dedica a la misma rutina de siempre, ejecutar instrucciones una detrás de la otra.

### *Periféricos*

En la organización básica de un microcontrolador, se ubican un conjunto de periféricos, cuyas salidas están reflejadas en los pines del microcontrolador. A continuación se describen algunos de los periféricos que con mayor frecuencia encontraremos en los microcontroladores.

- Entradas salidas de propósito general

También llamados como puertos de E/S, generalmente agrupadas en puertos de 8 bits de longitud, permiten leer datos del exterior o escribir en ellos desde el interior del microcontrolador, el destino habitual es el trabajo con dispositivos simples como relés, LED, o cualquier otra cosa.

Algunos puertos de E/S tienen características especiales que le permiten manejar salidas con determinados requerimientos de corriente, o incorporan mecanismos especiales de interrupción para el procesador.

Típicamente cualquier pin de E/S puede ser considerada E/S de propósito general, pero como los microcontroladores no pueden tener infinitos pines, ni siquiera todos los pines que queramos, las E/S de propósito general comparten los pines con otros periféricos. Para usar un pin con cualquiera de las características a él asignadas debemos configurarlo mediante los registros destinados a ello.

- Temporizadores y contadores

Son circuitos sincrónicos para el conteo de los pulsos que llegan a su entrada de reloj. Si la fuente de conteo es el oscilador interno del microcontrolador es común que no tengan un pin asociado, y en este caso trabajan como temporizadores. Por otra parte, cuando la fuente de conteo es externa, entonces tienen asociado un pin configurado como entrada, este es el modo contador.

Los temporizadores son uno de los periféricos más habituales en los microcontroladores y se utilizan para muchas tareas, como por ejemplo, la medición de frecuencia, implementación de relojes, para el trabajo de conjunto con otros periféricos que requieren una base estable de tiempo entre otras funcionalidades. Es frecuente que un microcontrolador típico incorpore más de un temporizador/contador e incluso algunos tienen arreglos de contadores. Los tamaños típicos de los registros de conteo son 8 y 16 bits, pudiendo encontrar dispositivos que solo tienen temporizadores de un tamaño o con más frecuencia con ambos tipos de registro de conteo.

- Conversor A/D

Se utiliza para tomar datos de varias entradas diferentes que se seleccionan mediante un multiplexor.

Las resoluciones más frecuentes son 8 y 10bits, aunque hay microcontroladores con convertidores de 11,12 y de 24 bits, para resoluciones mayores es preciso utilizar convertidores A/D externos. Los convertidores A/D son uno de los periféricos más codiciados en el mundo de los microcontroladores y es por ello que muchísimos PIC los incorporan, siendo esta una de las características más destacables de los dispositivos que fabrica Microchip.

- Puerto serie

Este periférico está presente en casi cualquier microcontrolador, normalmente en forma de UART (Universal Asynchronous Receiver Transmitter) o USART (Universal Synchronous Asynchronous Receiver Transmitter) dependiendo de si permiten o no el modo sincrónico de comunicación.

El destino común de este periférico es la comunicación con otro microcontrolador o con una PC y en la mayoría de los casos hay que agregar circuitos externos para completar la interfaz de comunicación. La forma más común de completar el puerto serie es para comunicarlo con una PC mediante la interfaz EIA-232 (más conocida como RS-232), es por ello que muchas personas se refieren a la UART o USART como puerto serie RS-232, pero esto constituye un error, puesto que este periférico se puede utilizar para interconectar dispositivos mediante otros estándares de comunicación.

- Puerto serie sincrónico

Este tipo de periférico se utiliza para comunicar al microcontrolador con otros microcontroladores o con periféricos externos conectados a él, mediante las interfaces SPI (Serial Peripheral Interface) o I2C (Inter-Integrated Circuit).

A pesar de que es también un tipo de puerto serie, es costumbre tratarlo de forma diferenciada respecto a la UART/USART porque las interfaces SPI e I2C aparecieron mucho después que la UART/USART, su carácter es únicamente sincrónico y no están diseñadas para interconectar al sistema con otros dispositivos independientes como una PC, sino para conectar al microcontrolador dispositivos tales como memorias, pantallas LCD, conversores A/D o D/A.

- Otros puertos de comunicación

En un mundo cada vez más orientado a la interconexión de dispositivos, han aparecido muchas interfaces de comunicación y los microcontroladores no se han quedado atrás para incorporarlas, es por ello que podemos encontrar algunos modelos con puertos USB (Universal Serial Bus), CAN (Controller Área Network), Ethernet, puerto paralelo entre otros.

- Comparadores

Son circuitos analógicos basados en amplificadores operacionales que tienen la característica de comparar dos señales analógicas y dar como salida los niveles lógicos '0' o '1' en dependencia del resultado de la comparación. Es un periférico muy útil para detectar cambios en señales de entrada de las que solamente nos interesa conocer cuando está en un rango determinado de valores.

- Modulador por ancho de pulsos

Los PWM (Modulación por Amplitud de Pulsos) son periféricos muy útiles sobre todo para el control de motores, sin embargo hay un grupo de aplicaciones que pueden realizarse con este periférico, dentro de las cuales podemos citar: la conversión digital analógica D/A, el control regulado de luz (dimming) entre otras.

- Memoria de datos no volátil

Muchos microcontroladores han incorporado este tipo de memoria como un periférico más, para el almacenamiento de datos de configuración o de los procesos que se controlan. Esta memoria es independiente de la memoria de

datos tipo RAM o la memoria de programas, en la que se almacena el código del programa a ejecutar por el procesador del microcontrolador.

Muchos de los microcontroladores PIC, incluyen este tipo de memoria, típicamente en forma de memoria EEPROM, incluso algunos de ellos permiten utilizar parte de la memoria de programas como memoria de datos no volátil, por lo que el procesador tiene la capacidad de escribir en la memoria de programas como si ésta fuese un periférico más.

- Protección ante fallo de alimentación o "Brownout"

Se trata de un circuito que resetea al microcontrolador cuando el voltaje de alimentación (VDD) es inferior a un voltaje mínimo ("brownout"). Mientras el voltaje de alimentación sea inferior al de brownout el dispositivo se mantiene reseteado, comenzando a funcionar normalmente cuando sobrepasa dicho valor.

- Perro guardián o "Watchdog"

Cuando el computador personal se bloquea por un fallo del software u otra causa, se pulsa el botón del reset y se reinicializa el sistema. Pero un microcontrolador funciona sin el control de un supervisor y de forma continuada las 24 horas del día. El Perro guardián consiste en un temporizador que, cuando se desborda y pasa por 0, provoca un reset automáticamente en el sistema.

Se debe diseñar el programa de trabajo que controla la tarea de forma que refresque o inicialice al Perro guardián antes de que provoque el reset. Si falla el programa o se bloquea, no se refrescará al Perro guardián y, al completar su temporización, "ladrará y ladrará" hasta provocar el reset.

- Estado de reposo ó de bajo consumo

Son abundantes las situaciones reales de trabajo en que el microcontrolador debe esperar, sin hacer nada, a que se produzca algún acontecimiento externo que le ponga de nuevo en funcionamiento. Para ahorrar energía, (factor clave en los aparatos portátiles), los microcontroladores disponen de una instrucción especial, que les pasa al estado de reposo o de bajo consumo, en el cual los requerimientos de potencia son mínimos. En dicho estado se detiene el reloj principal y se "congelan" sus circuitos asociados, quedando sumido en un profundo "sueño" el microcontrolador. Al activarse una interrupción ocasionada por el acontecimiento esperado, el microcontrolador se despierta y reanuda su trabajo.

## **1.2. BUS SERIAL UNIVERSAL USB**

En un principio teníamos la interfaz serie y paralelo, pero era necesario unificar todos los conectores creando uno más sencillo y de mayores prestaciones. Así nació el USB (Universal Serial Bus) con una velocidad de 12Mb/seg. y como su evolución, USB 2.0, y muy pronto el lanzamiento de USB 3.0 apodado USB de alta velocidad, con velocidades en este momento de hasta 480Mb/seg., es decir, 40 veces más rápido que las conexiones mediante cables USB 1.1.

El Universal Serial Bus (bus universal en serie) fue creado en 1996 por siete empresas: IBM, Intel, Northern Telecom, Compaq, Microsoft, Digital Equipment Corporation y NEC.

USB Universal Serial Bus es una interfaz plug&play entre la PC y ciertos dispositivos tales como teclados, mouses, scanner, impresoras, módems, placas de sonido, cámaras, etc.

Una característica importante es que permite a los dispositivos trabajar a velocidades mayores, en promedio a unos 12 Mbps, esto es más o menos de 3 a 5 veces más rápido que un dispositivo de puerto paralelo y de 20 a 40 veces más rápido que un dispositivo de puerto serial.

### **1.2.1 Funcionamiento**

Trabaja como un interfaz para transmisión de datos y distribución de energía, que ha sido introducida en el mercado de PC's y periféricos para mejorar las lentas interfaces serie (RS-232) y paralelo. Esta interfaz de 4 hilos, 12 Mbps y "plug and play", distribuye 5V para alimentación, transmite datos y está siendo adoptada rápidamente por la industria informática.

Es un bus basado en el paso de un dato, semejante a otros buses como los de las redes locales. El controlador USB distribuye datos por el bus, el dispositivo cuya dirección coincide con la que porta el dato responde aceptando o enviando datos al controlador. Este también gestiona la distribución de energía a los periféricos que lo requieran.

Emplea una topología de estrellas apiladas que permite el funcionamiento simultáneo de 127 dispositivos a la vez. En la raíz o vértice de las capas, está el controlador anfitrión o host que controla todo el tráfico que circula por el bus. Esta topología permite a muchos dispositivos conectarse a un único bus lógico sin que los dispositivos que se encuentran más abajo en la pirámide sufran retardo. A diferencia de otras arquitecturas, USB no es un bus de almacenamiento y envío, de forma que no se produce retardo en el envío de un paquete de datos hacia capas inferiores.

Como detalle sorprendente es que cada puerto utiliza una única solicitud de interrupción (IRQ) independientemente de los periféricos que tenga conectados

(sea 1 ó 127) por lo tanto no hay riesgo de conflictos entre una cantidad de dispositivos que de otra forma no podrían ser conectados por falta de recursos; de la misma manera tampoco utilizan DMA (asignación de memoria).

### **1.2.2 Características Generales**

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

1. El ordenador debe identificar automáticamente un dispositivo agregado mientras trabaja y configurarlo.
2. Los dispositivos pueden ser también desconectados mientras el ordenador está en uso.
3. Deben poder compartir un mismo bus tanto dispositivos que requieren de unos pocos Kbps como los que requieren varios Mbps.
4. Más de 127 dispositivos diferentes pueden estar conectados simultáneamente y operando con una misma computadora sobre el Bus Serial Universal.
5. El bus debe permitir periféricos multifunción, es decir aquellos que pueden realizar varias tareas a la vez, como lo son algunas impresoras que adicionalmente son fotocopadoras y máquinas de fax.
6. Capacidad para manejo y recuperación de errores producidos por un dispositivo cualquiera.
7. Soporte para la arquitectura Conectar y Operar (Plug&Play).
8. Bajo costo.

### **1.2.3 Características de Transmisión**

El cable USB soporta tres velocidades de transferencia de datos:

- Baja Velocidad (1.0): Bitrate de 1.5Mbit/s (192KB/s). Utilizado en su mayor parte por Dispositivos de Interfaz Humana (HID) como los teclados, los ratones y los joysticks.

- Velocidad Completa (1.1): Bitrate de 12Mbit/s (1.5MB/s). Esta fue la más rápida antes de que se especificara la USB 2.0 y muchos dispositivos fabricados en la actualidad trabajan a esta velocidad. Estos dispositivos, dividen el ancho de banda de la conexión USB entre ellos basados en un algoritmo FIFO.
- Alta Velocidad (2.0): Bitrate de 480Mbit/s (60MB/s).
- Súper Velocidad (3.0) Actualmente en fase experimental. Bitrate de 4.8Gbit/s (600MB/s). Esta especificación será lanzada a mediados de 2008 por la compañía Intel, de acuerdo a información recabada de Internet. Las velocidades de los buses serán 10 veces más rápidas que la de USB 2.0 debido a la inclusión de un enlace de fibra óptica que trabaja con los conectores tradicionales de cobre. Se espera que los productos fabricados con esta tecnología lleguen al consumidor en 2009 o 2010.

Las señales del USB son transmitidas en un cable de data par trenzado con impedancia de  $90\Omega \pm 15\%$  llamados D+ y D- . Éstos, colectivamente utilizan señalización diferencial en half dúplex para combatir los efectos del ruido electromagnético en enlaces largos. D+ y D- usualmente operan en conjunto y no son conexiones simples. Los niveles de transmisión de la señal varían de 0 a 0.3V para bajos (ceros) y de 2.8 a 3.6V para altos (unos) en las versiones 1.0 y 1.1, y en  $\pm 400\text{mV}$  en Alta Velocidad (2.0). En las primeras versiones, los alambres de los cables no están conectados a masa, pero en el modo de alta velocidad se tiene una terminación de  $45\ \Omega$  a tierra o un diferencial de  $90\ \Omega$  para acoplar la impedancia del cable.

### Cuadro 1. Pines de Cable USB

PIN	NOMBRE	COLOR DEL CABLE	DESCRIPCION
1	VCC	ROJO	+5V
2	D-	BLANCO	Data -
3	D+	VERDE	Data +
4	GND	NEGRO	Tierra

Fuente: Autores

### 1.2.4 Conectores USB Serie "A" Y Serie "B"

Existen dos tipos de conectores dentro del Bus Serial Universal. El conector Serie A está pensado para todos los dispositivos USB que trabajen sobre plataformas de PCs. Serán bastante comunes dentro de los dispositivos listos para ser empleados con host PCs, y lo más probables es que tengan sus propios cables con su conector serie A. Sin embargo, esto no se dará en todos los casos, existirán dispositivos USB que no posean cable incorporado, para los cuales el conector Serie B será una característica. Sin embargo este no es un problema, ya que ambos conectores son estructuralmente diferentes e insertarlos de forma equívoca será imposible por la forma de las ranuras.

La serie de conectores tipo B en su mayoría, son implementados en periféricos que tienen cables desmontables.

Figura 3. Conectores USB



Fuente: AXELSON, Jan. USB COMPLETE: Electrical Interface. 3 Edición. Madison WI: Lakeview Research LLC. Año 2005, Pag. 522

Los tipos de conectores USB también pueden ser Micro USB, mini USB y Hembra USB.

Los cables USB son de 5Vdc y pueden ser usados para suministrar energía a cualquier periférico. El cable provee alrededor desde unos 100mA hasta 500mA de corriente. Algunos periféricos deben incluir fuentes internas o externas, ya que

en ocasiones es necesario mayor consumo de corriente según en lo que se este trabajando.

### **1.2.5 Requerimientos para Desarrollar una Aplicación USB**

Para el desarrollo de una aplicación se deben tener en cuenta los siguientes puntos como son: Un osciloscopio, una computadora con puerto USB, borrador de memorias, algún kit de desarrollo, etc.

### **1.3. LENGUAJE GRAFICO (Lenguaje G)**

G es un lenguaje de programación, aunque a diferencia de la gran mayoría no es basado en texto, es gráfico, que a través de diagramas se representa la lógica del programa. El lenguaje G es el lenguaje que se utiliza para desarrollar programas en LabVIEW. **LabVIEW** es una herramienta gráfica de test, control y diseño mediante la programación.

Antes se utilizaba en dos productos de NI, LabVIEW y BridgeVIEW, posteriormente BridgeVIEW se convirtió en un modulo de LabVIEW (DSC) y ahora es LabVIEW el único programa que utiliza G.

Los programas hechos con LabVIEW se llaman VI (Virtual Instrument), lo que da una idea de uno de sus principales usos: el control de instrumentos.

*Principales usos:*

- Adquisición de datos.
- Control de instrumentos.
- Automatización industrial o PAC (Controlador de Automatización Programable).
- Diseño de control.

*Principales características*

Su principal característica es la facilidad de uso, personas con pocos conocimientos en programación pueden hacer programas relativamente complejos. También es muy rápido hacer programas con LabVIEW y cualquier programador, por experimentado que sea, puede beneficiarse de él. En LabVIEW se puede crear programas de miles de VIs (páginas de código) para aplicaciones complejas, programas de automatizaciones de decenas de miles de puntos de entradas/salidas, etc.

*Presenta facilidades para el manejo de:*

- Interfaces de comunicaciones como son puerto serie, puerto paralelo, GPIB, PXI, VXI, TCP/IP, UDP, IrDA, OPC, Bluetooth, USB, entre otras.
- Con la característica de poder interactuar con otras aplicaciones como son: dll, ActiveX, Matlab, y Simulink.
- Herramientas para el procesado digital de señales.
- Visualización y manejo de gráficas con datos dinámicos.
- Adquisición y tratamiento de imágenes.
- Control de movimiento.
- Tiempo Real estrictamente hablando.
- Programación de FPGAs.
- Sincronización.

## CAPITULO 2. DESARROLLO DEL ENTRENADOR DIGITAL USB

### 2.1 Descripción General

Para el desarrollo del proyecto, la programación fue realizada en lenguaje assembler de la herramienta *Aspire (simulador)* para el Microcontrolador ADUC 841 y en el lenguaje G de la herramienta Labview. Cabe resaltar que a lo largo del presente proyecto se hablara de las características de hardware y funcionamiento, también de las principales rutinas el cual hacen posible la realización de los múltiples módulos del entrenador, y la explicación del interfaz realizada en Labview.

El entrenador digital USB, es una herramienta dirigida hacia el desarrollo de prácticas de laboratorio de ingeniería electrónica; está conformado por módulos y posee una interfaz de comunicación que implementa el intercambio de datos entre los protocolos USB y serial, sin comprometer la integridad de los datos.

El entrenador digital consta de un modulo ADC en modo DMA (Acceso Directo a Memoria), de un modulo DAC, con una resolución de salida de 12 bits; el modulo PWM, con varios modos de funcionamiento y configuración, de un modulo de puertos de I/O (Entradas y Salidas) y el contador de eventos.

El método de comunicación funciona en conjunto con driver/firmware que se encuentra distribuido de la siguiente manera: El Firmware está almacenado en la memoria EEPROM que está ubicada en el circuito impreso del modulo de comunicación, y el driver, el Virtual Com Port (VCP) está almacenado en el Host (computador). Para información adicional véase anexo **C. Driver/Firmware Virtual Com Port (VCP)**.

El entrenador digital emplea una interfaz grafica, con lenguaje de programación en Labview. Dicha interfaz permite al usuario configurar y manipular los distintos módulos que posee el entrenador digital USB.

Los módulos son los siguientes:

El módulo de PWM, permite al estudiante configurar la señal de salida PWM, en uno de los cuatro diferentes modos de operación, así como programar el ciclo de trabajo y la frecuencia de la señal de salida PWM.

El módulo de conversión análogo / digital incorpora un ADC de ocho canales, a una resolución de 12 Bits, de una sola fuente. El conversor ADC consiste de un convertidor convencional de aproximaciones sucesivas.

El modo de conversión DMA permite escribir los resultados directamente a la memoria estática externa de 2 Mbytes, sin cualquier interacción del núcleo.

El módulo de conversión digital / análogo DAC proporciona al estudiante la opción de convertir una señal en formato digital, a su equivalente análogo. La señal digitalizada puede originarse desde el computador, por medio del software de aplicación, siendo más específico desde el software LABVIEW, ó desde el módulo de conversión análogo / digital que posee el mismo entrenador.

El módulo de puertos digitales I/O permite intercambiar datos, con dispositivos externos al microprocesador usando lógica TTL. El entrenador ha destinado dos puertos bidireccionales de ocho bits, para realizar operaciones de propósito general, el puerto cero y el puerto dos. El puerto cero, es de colector abierto mientras el puerto dos posee resistencias internas de “pull up”.

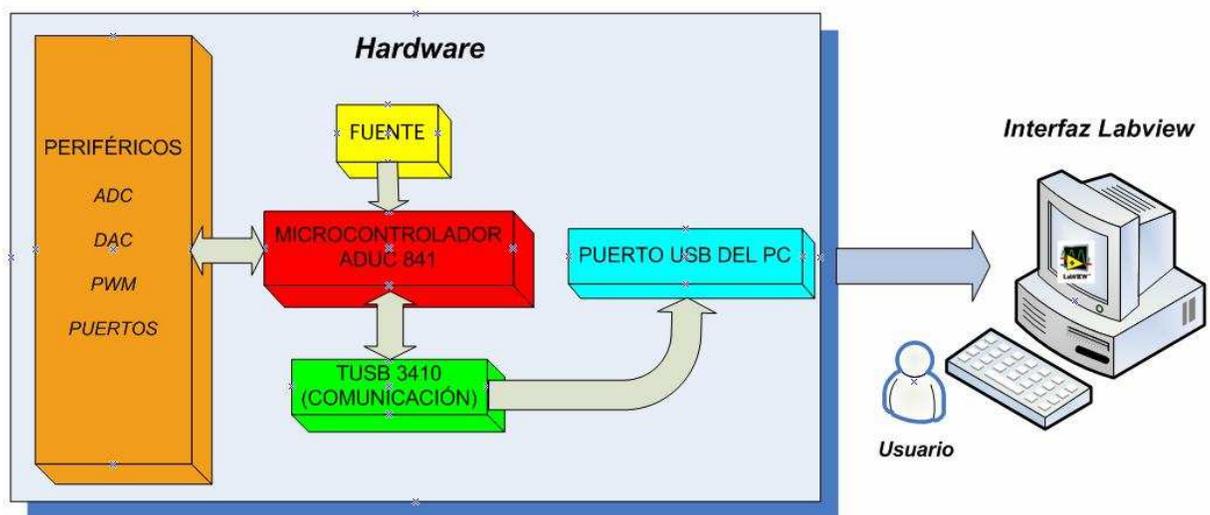
El módulo contador de eventos, permite al estudiante realizar un conteo de un tren de pulsos, permitiendo configurar un pre-escalamiento que determina, la razón de incremento del contador.

## 2.2 HARDWARE

### 2.2.1 Descripción

En esta sección se describe el hardware del entrenador digital USB. El diagrama de bloques del hardware se muestra en la figura 4 con la detallada descripción de cada bloque.

Figura 4. Diagrama de Bloques de Hardware del Entrenador Digital



Fuente: Autores

### 2.2.2 Modulo de Fuentes

El entrenador digital USB tiene 2 fuentes de alimentación:

- 1 Fuente de CC3 – 1205SS-E
- 1 Fuente de alimentación por el puerto usb.

Se utiliza 2 fuentes de alimentación debido a que el integrado SN75LV4737A maneja diferentes valores de voltaje y corriente, por ello, para asegurar un funcionamiento efectivo se disponen de un CC3 – 1205SS-E y la alimentación por el puerto USB, para garantizar la suficiente corriente en toda la tarjeta y no se caiga el voltaje.

La fuente CC3 – 1205SS-E que se muestra en la siguiente figura 5 alimenta al Microcontrolador ADUC841, a los operacionales, al max232, SN75LV4737A y suministra 5Vs externos a la bornera del entrenador.

Figura 5. Fuente CC3 – 1205SS-E



Fuente: Datasheet CC3 – 1205SS-E

Las características principales de esta fuente se presentan en la siguiente tabla.

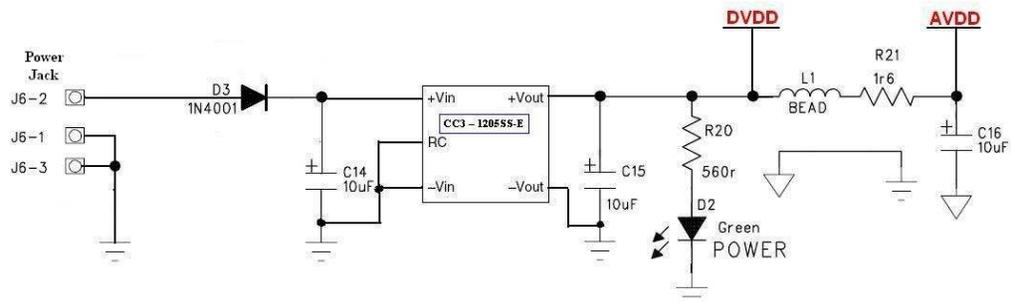
**Cuadro 2. Características Generales de la Fuente CC3 – 1205SS-E**

Voltaje de Entrada (V)	Voltaje de Salida (V)		Corriente de salida (A)	Eficiencia (%)
9 - 18	5.0	+/-3%	0.6	79

Fuente: Autores

### 2.2.2.1 Diseño del Circuito

Figura 6. Diagrama del Circuito de la Fuente CC3 – 1205SS-E

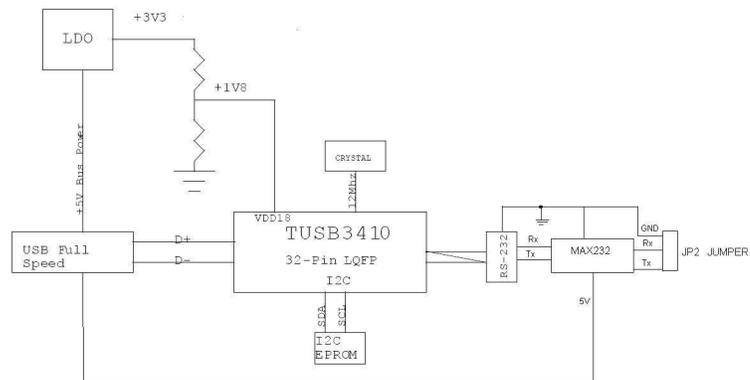


Fuente: Autores

La figura 6 muestra el diagrama de circuitual de la fuente CC3 – 1205SS-E es un convertidor DC/DC el cual es alimentado por 9V y da una salida de 5V de manera análoga y digital como lo permite el arreglo circuitual de la fuente.

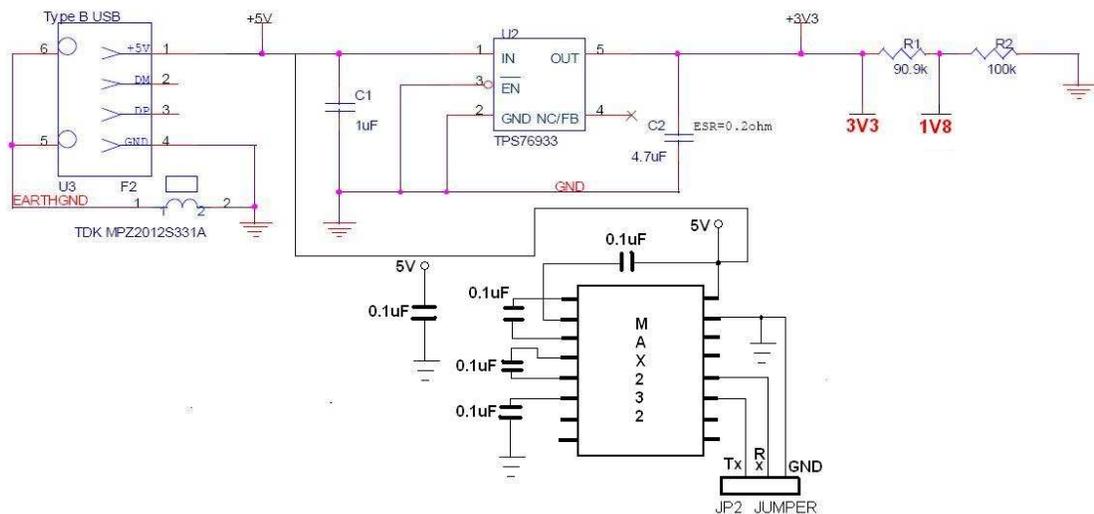
El siguiente diagrama de bloques figura 7 y el diagrama del circuito figura 8, muestra la alimentación por parte del puerto USB, el cual suministra 5V al TPS76P33 y al MAX232, donde el TPS76P33 regula el voltaje suministrado por el puerto USB 3.3V y por medio de un divisor de voltaje también suministra 1.8V.

Figura 7. Diagrama de Bloques de Alimentación por el Puerto USB.



Fuentes: Autores

Figura 8. Diagrama del Circuito de Alimentación por el Puerto USB.



Fuentes: Autores

### 2.2.3 Modulo del Microcontrolador ADUC 841

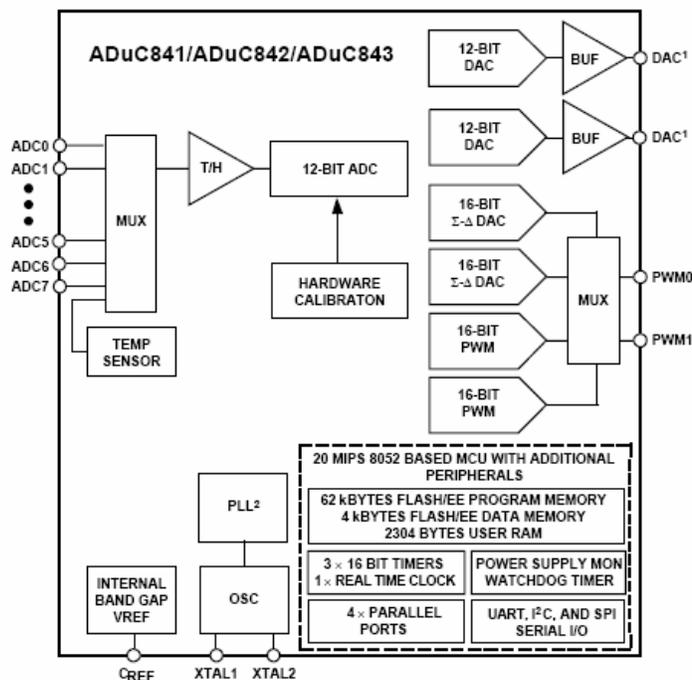
El ADuC841 es un Microcontrolador que integra un alto rendimiento y autocalibración multicanal ADC, un DAC dual, y un óptimo ciclo de 20MHz 8-bit MCU.

El ADuC841 trabaja con una frecuencia de reloj de 20MHz con el cristal externo. El microcontrolador es un núcleo optimizado 8052 con 20MIPS (Millones Instrucciones Por Segundo) de rendimiento pico. Contiene memorias disponibles ofreciendo de 62 Kbytes de memoria de programa no volátil Flash/EE, 4 KBytes de memoria de datos no volátil Flash/EE. 256 bytes RAM, y 2KBytes de memoria RAM extendida, también integrada en el microcontrolador.

Tiene funcionalidad análoga con 2 DAC's de 12 bit; incluye dos DAC's de 16 bits, una salida dual PWM de 16 bit, un watchdog timer, un contador de intervalo de tiempo, tres timers o contadores y tres puertos seriales I/O (SPI, I2C, y UART).

La interfaz SPI opera separadamente en los puertos P3.3, P3.4, P3.5 mientras la I2C usa pines estándares. El firmware soporta descarga en circuito serial, modo de depuración (por medio de la UART), así como también pin de emulación por medio del pin EA. Un diagrama de bloques de funcionalidad de las partes se muestra en la figura.

Figura 9. Diagrama de Bloques Funcional



Fuente: Datasheet ADuC841

### *Características:*

- Alto rendimiento con un ciclo de hasta 20 MIPS en el 8052, alta velocidad en el núcleo de 420 KSPS y con un ADC de 12 bits.
- Memoria de alta capacidad.
- Circuito reprogramable, Flash/EE con 100 años de retención, 100Kciclos de duración, 2304 en el chip de RAM de datos.
- Encapsulado pequeño LSCSP-56 8mm x 8mm.
- I/O Análogas con 8 canales, 420 KSPS de alta precisión, 12 bit en el ADC integrado, control DMA, alta velocidad de captura de ADC a la RAM, dos DAC's de 12 bits.
- Dual output PWM  $\Sigma$ - $\Delta$  DAC's.
- Alto rendimiento en un ciclo en el núcleo, PLL programable dentro del integrado, 12 fuentes de interrupción, 2 niveles de prioridad en el apuntador dual de datos.
- Periféricos integrados como el contador de eventos, UART, I2C®, y SPI® Serial I/O.
- Alimentación normal de 4.5 mA a 3 V (núcleo CLK = 2.098 MHz) y en modo de ahorro de energía de 10  $\mu$ A a 3V con cristal externo.

## **2.2.4 Periféricos**

### **2.2.4.1 Convertidor Análogo/Digital (ADC)**

El ADC incorpora 8 canales, a una resolución de 12 Bits, de una sola fuente. El conversor ADC consiste de un convertidor convencional de aproximaciones sucesivas, basados en un capacitor DAC como se muestra en la figura 18.

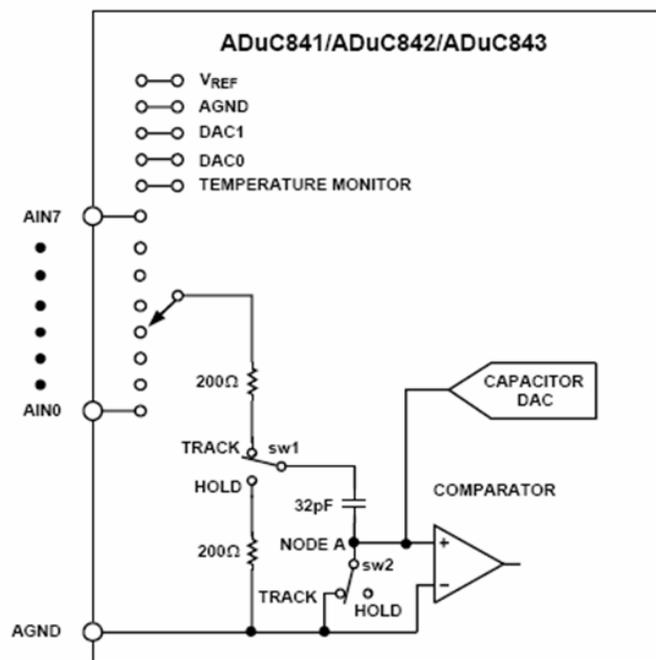
Al ADC incorpora una estructura de aproximaciones sucesivas (SAR) que involucra el estado de la entrada de la carga-muestra. En la figura 18 se muestra el circuito equivalente de la sección de la entrada análoga. Cada conversión del ADC está dividida en dos distintas fases, definida por la posición de los interruptores como se muestra en la figura 18. Durante la fase de muestreo, una carga proporcional al voltaje de la entrada análoga se desarrolla a través de la entrada del capacitor de muestreo. Durante la fase de conversión el capacitor del DAC ajusta internamente el SAR lógico hasta que el voltaje del nodo A sea cero indicando que la carga muestreada a la entrada del capacitor ha sido balanceada a la salida del condensador del DAC. El valor final digital contenido en el SAR entonces se cierra a la salida, como resultado de la conversión del ADC. El control del SAR para programar el tiempo de adquisición y el modo de muestreo se

maneja automáticamente, por la construcción del control lógico del ADC. El tiempo de adquisición y conversión son también configurables por el usuario.

Una vez configurado el ADCCON 1 - 3 SFR's, al ADC convierte la entrada análoga y provee como resultado al ADC una palabra de 12 bit en el ADCDATAH/L SFR's. Los 4 bit de parada del ADCDATAH/L SFR's son escritos en el canal de selección de bit para identificar el canal resultante. Para mayor información véase en anexo:

## B. Características del Nucleo8052.

Figura 10. Estructura Interna del ADC.



Fuente: Datasheet ADuC841

Las partes del ADC de una aproximación sucesiva se conducen por una división descendente del reloj maestro. Para asegurar la operación del ADC, el reloj debe estar entre los 400KHz y los 8.38MHz. Las frecuencias dentro de este rango pueden conseguirse con el reloj maestro desde frecuencias de 400KHz hasta sobre los 16 MHz. En modo de conversión continua una nueva conversión empieza cada una vez previamente una finaliza. La tasa de muestreo es simplemente al inverso del tiempo total de conversión.

### **2.2.4.1.1 Modo DMA**

El modo de conversión DMA escribe los resultados directamente a la memoria estática externa de 2 Mbytes (Interna en el microcontrolador ADUC841), sin cualquier interacción del núcleo.

Opera de desde el reloj ADC y usa “pipeling” para llevar a cabo la conversión ADC y para acceder la memoria al mismo tiempo. El tiempo requerido para llevar a cabo una conversión es llamado un ciclo DMA. Las acciones realizadas durante este ciclo son las siguientes:

1. Una conversión ADC es realizada sobre un canal, cuyo ID fue leído durante el ciclo previo.
2. El resultado de 12 bits y el ID del canal de la conversión llevada a cabo en el ciclo anterior, es escrito a la memoria externa.
3. El ID del próximo canal a ser convertido, es leído desde la memoria externa.

Ya que la lógica DMA usa “pipeling”, toma tres ciclos antes de que el primer resultado correcto sea escrito al espacio en memoria.

Cuando la conversión DMA ha finalizado, se establece una interrupción y la SRAM externa contiene los resultados de la nueva conversión ADC.

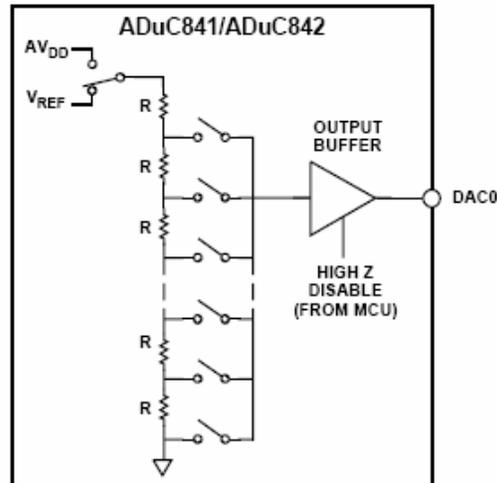
### **2.2.4.2 Convertidor Digital/Análogo (DAC)**

Básicamente el DAC funciona en la conversión de una tensión en un número binario a través de una cadena de resistencias del DAC seguidas por un amplificador buffer de salida donde la arquitectura del DAC posee características de monotocidad y una excelente linealidad diferencial. Ver figura 19.

La señal digitalizada puede originarse desde el computador, por medio del software de aplicación, siendo más específico desde el software LABVIEW, ó desde el módulo de conversión análogo / digital que posee el mismo entrenador.

Entre sus características específicas, el conversor digital / análogo está dotado de dos salidas de conversión independientes y una resolución máxima de salida de 12 bits. Cada una de las salidas posee un buffer capaz de manejar una carga de 10 K $\Omega$  / 100 pF. El rango de trabajo del conversor está entre 0 [V] y 2.5 [V].

Figura 11. Arquitectura del DAC



Fuente: Datasheet ADuC841

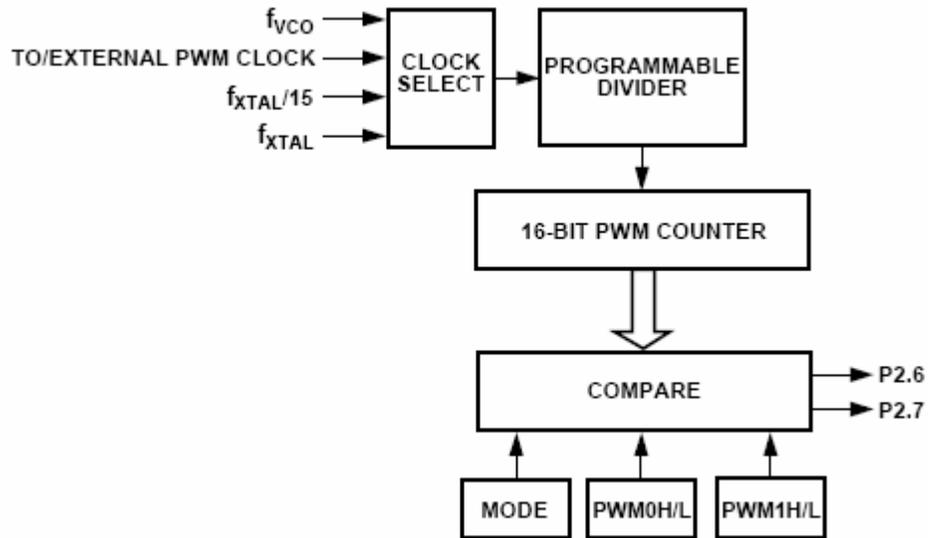
Ambos DAC's parten de un registro de control, DACCON, y cuatro registros de datos DAC1H/L, DAC0/L. En modo de 12 bit asíncrono la salida del voltaje del DAC es actualizada tan pronto como el dato DACL ha sido escrito en el SFR. Además, el registro de datos del DAC deberá ser actualizado como DACH primero, seguido por el DACL. Véase en anexos:

## B. Características del Nucleo8052

### 2.2.4.3 Modulación por Amplitud de Pulso (PWM)

El PWM funciona básicamente modificando el ciclo de trabajo de la señal periódica, éste ofrece una alta resolución programable, una entrada de reloj, y puede ser configurado de cualquiera de los cuatro modos de operación. Un diagrama de bloques se muestra en la figura 12.

Figura 12. Diagrama de Bloques del PWM



Fuente: Datasheet ADuC841

Según el modo seleccionado, el PWM está en la disponibilidad de generar una o dos señales de salida PWM, de las cuales, parámetros de la señal tales como el período y el ciclo de trabajo, son completamente configurables, de forma independiente.

El módulo PWM posee una resolución variable, la cual se encuentra limitada según el módulo seleccionado. La resolución máxima posible es de 16 Bits. Al ser independiente sus salidas, permite establecer distintas frecuencias y ciclos de trabajo, para cada señal de salida PWM.

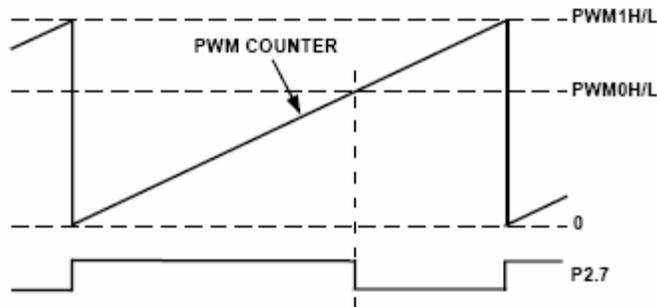
El PWM del microcontrolador maneja cinco registros uno de control (PWMCON) y cuatro registros de datos (PWM0H, PWM0L, PWM1H, y PWM1L). Véase anexo **B. Características del Nucleo8052**.

### 2.2.4.3.1 Modos de Operación

- *Modo uno: Único PWM Resolución Variable*

En el modo uno, la longitud de pulso y la duración del ciclo (período) son programables por código de usuario, permitiendo la resolución variable del PWM.

Figura 13. PWM Modo 1



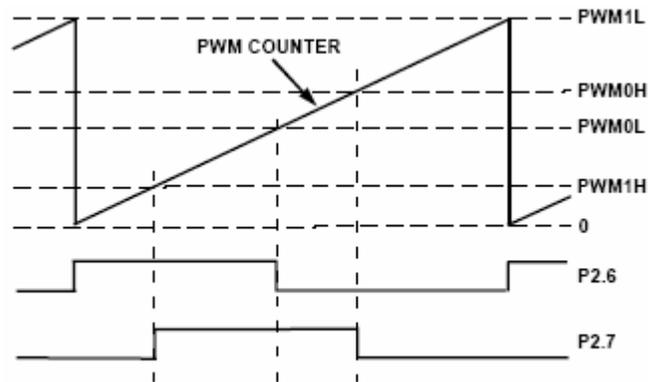
Fuente: Datasheet ADuC841

- *Modo dos: Doble PWM 8 bits*

En el modo 2, la duración del ciclo de la salida del PWM y la resolución de la salida del PWM son ambas programables. La máxima resolución de salida del PWM es de 8 bits.

Las salidas del PWM son P2.6 y P2.7 como se muestran en la figura 14.

Figura 14. PWM Modo 2

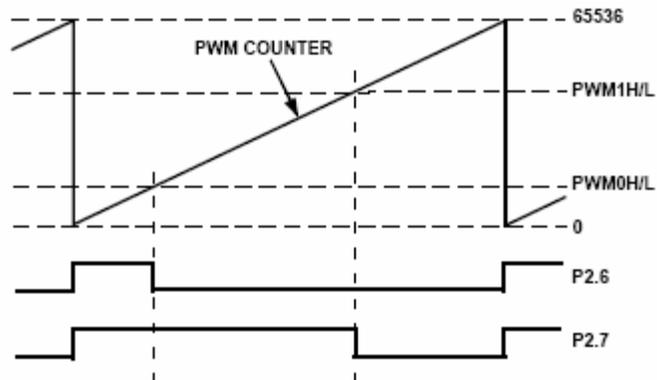


Fuente: Datasheet ADuC841

- *Modo tres: Doble PWM 16 bits*

En el modo 3, el contador PWM es fijo para un conteo de 0 a 65536, ofreciendo un PWM fijo de 16-bit. Opera desde 16.777MHz el reloj del núcleo da lugar a un índice de salida de PWM de 256 Hz. La duración del ciclo de las salidas del PWM a P2.6 y P2.7 es independiente programable.

Figura 15. PWM Modo 3

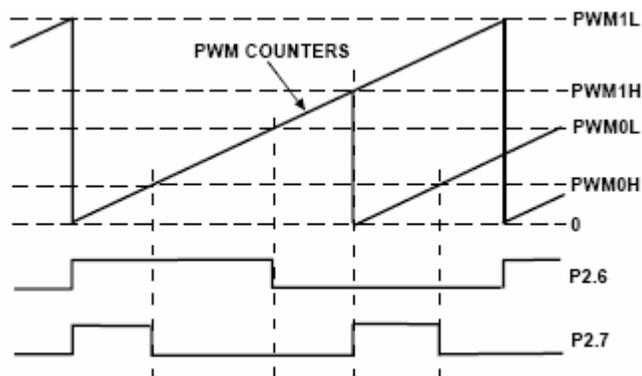


Fuente: Datasheet ADuC841

- *Modo cuatro: Doble PWM 8 bits*

En el modo 5, la duración del ciclo en las salidas PWM y la resolución de las salidas PWM son individualmente programables. La máxima resolución de las salidas PWM es de 8 bits. Ambos PWM's tienen la misma fuente de reloj y divisor de reloj.

Figura 16. PWM Modo 4



Fuente: Datasheet ADuC841

#### 2.2.4.4 Puertos

El módulo de puertos permite intercambiar datos, con dispositivos externos al microprocesador usando lógica TTL. El entrenador ha destinado dos puertos

bidireccionales de ocho bits, para realizar operaciones de propósito general, el puerto cero y el puerto dos.

- *El Puerto Cero*

El puerto es de 8 bit y es de colector abierto y es directamente controlado por el registro PORT0 del SFR. El puerto 0 también es multiplexado bajo la orden de la dirección y de los datos del bus durante el acceso programa externo o la memoria de datos.

Los pines del puerto 0 pueden ser usados como entradas de alta impedancia y también como salidas de propósito general.

- *El Puerto 2*

Posee resistencia pull-up directamente controladas por el P2 del SFR. También emite byte de dirección de alto orden durante la ejecución de la memoria de programa externa, y medio y alto dirección de byte durante al acceso al espacio en memoria externa de datos de 24 bits.

#### **2.2.4.5 Contador de Eventos.**

El Timer puede ser controlado individualmente, y puede ser usado en tres funciones generales.

- 1) Generando retardos o calculando tiempos entre dos eventos. (Modo Timer).
- 2) Contando eventos (Modo Contador).
- 3) Generador de razón de baudios para la comunicación serial.

En el modo Timer el contador se incrementa cada ciclo de máquina. Un ciclo de máquina consiste en 1 pulsos de cristal.

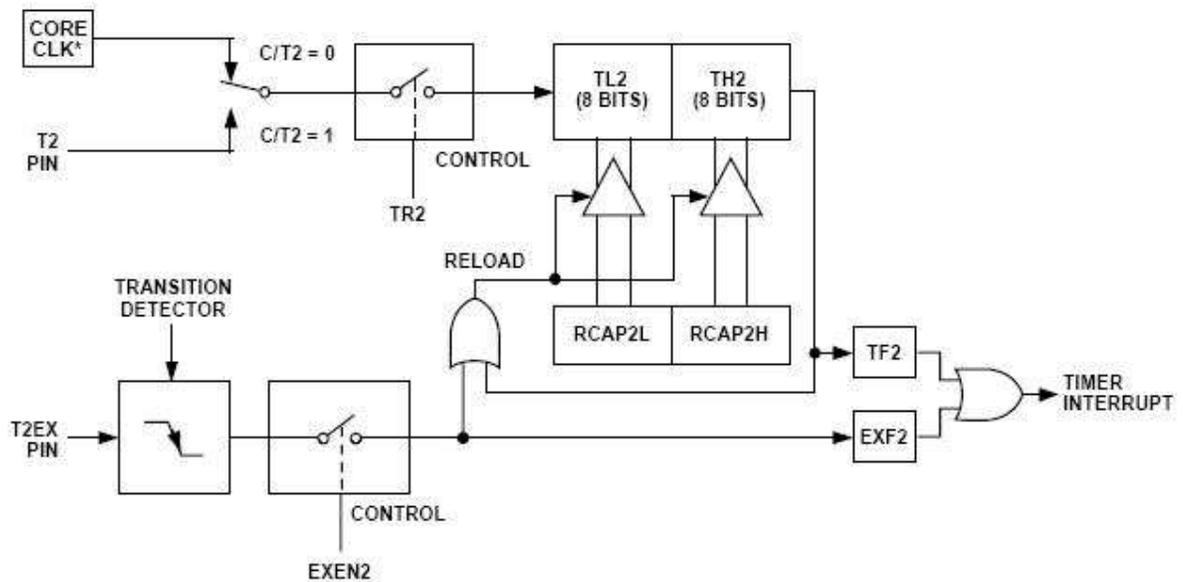
1 Ciclo de Maquina = Periodo del Cristal  
Periodo del cristal =  $1 / \text{Frecuencia del Cristal}$

El Timer siempre cuenta ascendente, no importa en qué modo este siendo usado el Timer el siempre cuenta en forma ascendente y se desborda cuando el conteo va de 0ffffh a 0000h.

Los Timer son divididos en 2 registros de 8 bits del (SFR), llamados Timer LOW (TL0, TL1) y Timer HIGH (TH0, TH1). Estos registros el ultimo valor de conteo del

Timer. La acción del Timer es controlada por dos registros del SFR Timer: Modo de Control de Registro (TMOD) y Timer/Counter Control Registro (TCON).

Figura17. Diagrama de Bloques del Contador de 16 Bits



Fuente: Datasheet ADuC841

### Timer/Contador 16 Bits En Modo Autorecarga:

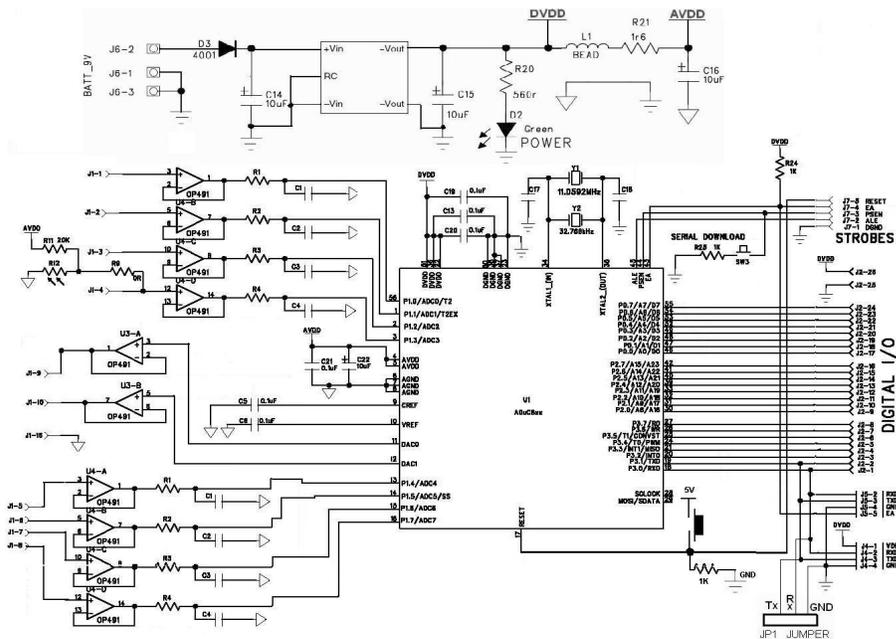
El modo de autorecarga tiene dos opciones que son seleccionadas por medio del bit EXEN2 en el registro de funciones especiales (SFR) T2CON. El Timer/Contador2 está configurado por defecto de tal manera que el bit EXEN2= 0, el Timer/contador2 se desborde desde 65535 a 0, no solamente establece la bandera de interrupción del timer2 TF2 cuya interrupción se encuentra habilitada, sino que también causa que los registro del Timer/contador2 TH2, TL2 sean recargados con el valor de 16 bits que está guardado en los registros RCAP2L y RCAP2H, los cuales son preestablecidos por software.

Para mayor información véase anexos:

### B. Características del Nucleo8052.

## 2.2.5. Diseño del Circuito del Entrenador Digital USB

Figura 18. Diagrama del circuito del Microcontrolador ADUC 841



Fuente: Autores

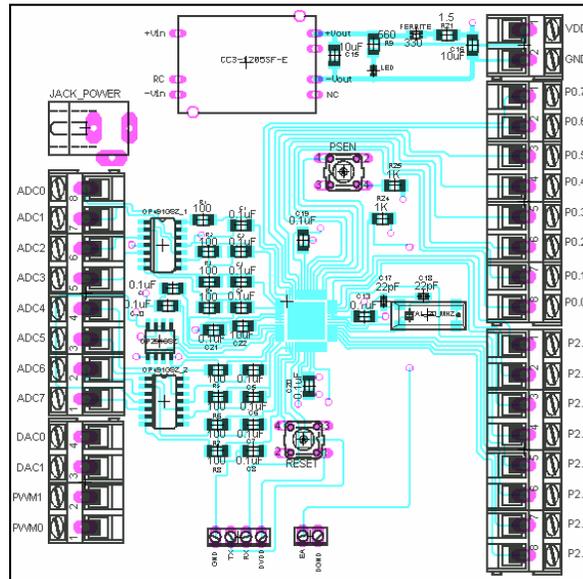
La figura 18 muestra el diagrama del circuito del microcontrolador ADUC841, el cual contiene los módulos de PWM, los módulos de conversión ADC Y DAC, conformado por los operacionales, el modulo PWM, la comunicación entre el microcontrolador ADUC841 y el microcontrolador TUSB3410 a través de los jumper JP1 Y JP2 y su fuente de alimentación por CC3 – 1205SS-E, como fue nombrada anteriormente.

El microcontrolador ADUC841 es el encargado de realizar todas las funciones por los módulos y es quien interactúa con el microcontrolador TUSB3410 para el flujo de datos.

A continuación los diseños del circuito impreso del microcontrolador junto con la fuente CC3 – 1205SS-E por ambas caras.

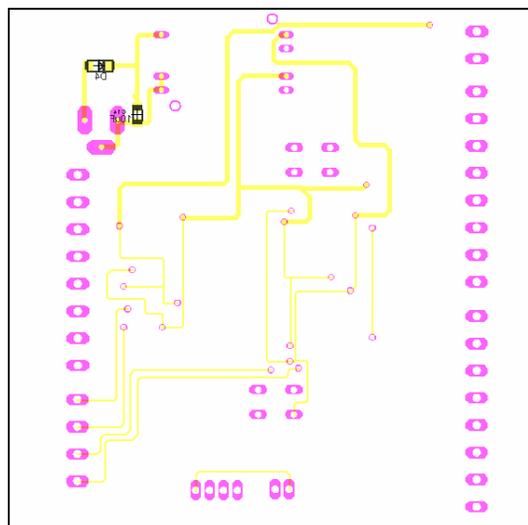
### 2.2.5.1 Diseño del Circuito Impreso

Figura 19. Cara Superior del Circuito Impreso (Bloque Microcontrolador ADUC 841 y Fuente de Alimentación):



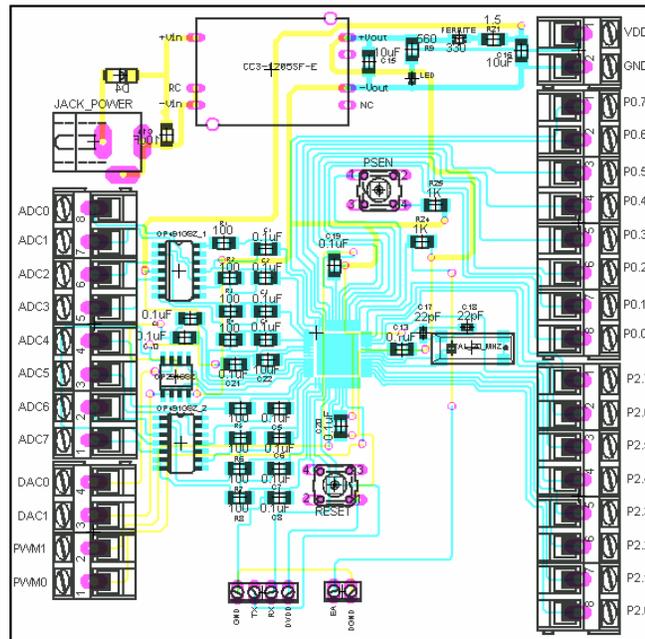
Fuente: Autores

Figura 20. Cara Inferior del Circuito Impreso (Bloque Microcontrolador ADUC 841 y Fuente de Alimentación):



Fuente: Autores

Figura 21. Cara Superior e Inferior del Circuito Impreso (Bloque Microcontrolador ADUC 841 y Fuente de Alimentación):



Fuente: Autores

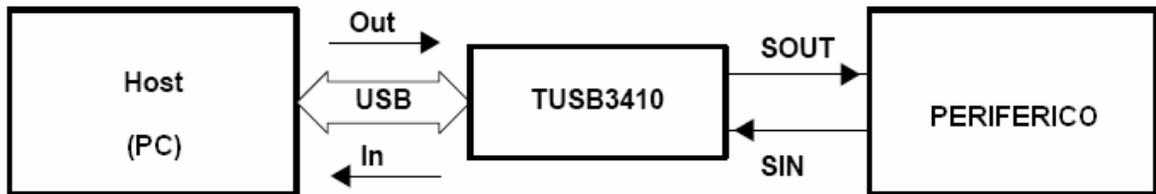
## 2.2.6 Modulo USB (Microcontrolador TUSB3410)

El TUSB3410 establece un puente entre el puerto USB y un puerto serie UART. El TUSB3410 contiene la lógica por medio del firmware que se encuentra almacenado en la memoria serial, que permite comunicar al computador con un dispositivo de puerto serial empleando el bus USB.

Incluye un microcontrolador 8052 unidad (MCU) con 16Kbyte de memoria RAM, 10 Kbyte de la ROM que permite a la unidad configurar el puerto USB en el momento de arranque.

El TUSB3410 se usa como interfaz entre un dispositivo periférico serial y un Puerto USB del PC. El siguiente diagrama muestra el flujo de datos a través del Microcontrolador.

Figura 22. Flujo de Datos a través del Microcontrolador TUSB3410



Fuente: Autores

El microcontrolador TUSB3410 funciona con el driver VCP para Windows, que se explicara más adelante el modo de instalación.

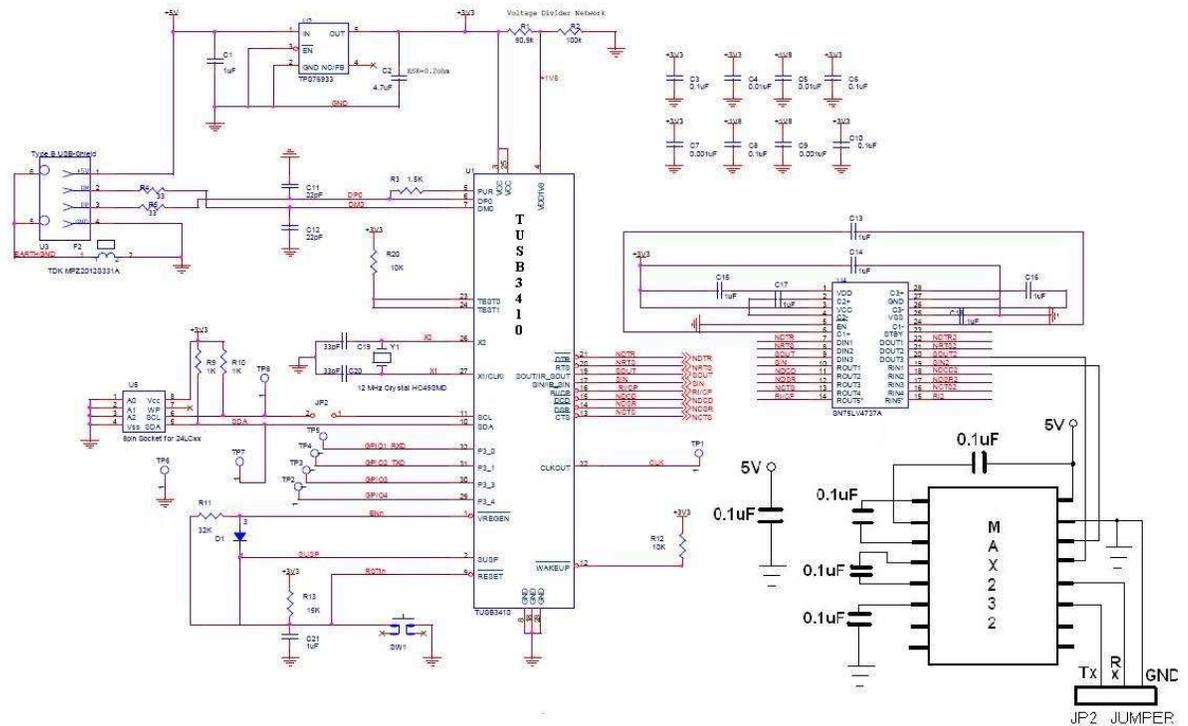
El microcontrolador trabaja con un cristal de 12MHz el cual se encarga de la fuente de reloj del dispositivo, además también trabaja con la memoria EEPROM que contiene el código para arrancar el TUSB3410's interno 8052 micro controlador y otros parámetros de configuración.

Entre las características principales están:

- Compatible con USB 2.0 full velocidad.
- Soporta 12 Mbps en la tasa de datos USB.
- Puede soportar un total de 3 entradas y 3 salidas endpoints.
- Opera con un cristal de 12MHz.
- Trabaja a 3.3V con 1.8V con el funcionamiento básico, el voltaje de funcionamiento proporcionado por el integrado 1.8V es un voltaje regulado.

### 2.2.6.1 Diseño del Circuito del Módulo USB

Figura 23. Diagrama del Circuito del Microcontrolador TUSB3410 (Bloque USB)



Fuente: Autores

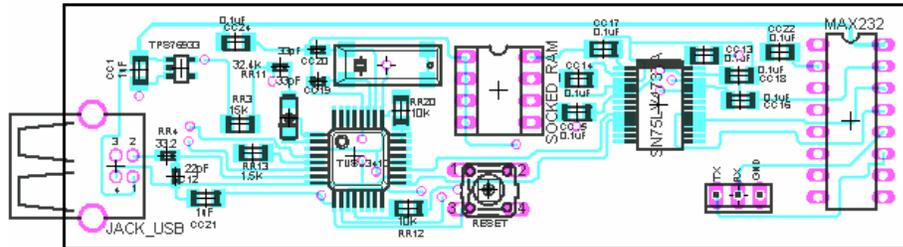
El microcontrolador TUSB3410 recibe los datos de la interfaz de Labview por medio del puerto USB e internamente realiza transferencias DMA (Acceso Directo Memoria) para transmitir los datos bajo el protocolo UART. El manejador/receptor multicanal de línea (SN75LV4737A) eleva los niveles de voltaje a RS-232 (+8V,-8V) y viceversa, ya que el TUSB3410 se encuentra polarizado a 3.3V. Este proceso aplica de igual manera tanto para transmisión como para recepción de datos.

El MAX232 recibe los datos en niveles RS-232, y los convierte a niveles lógicos TTL y viceversa. El JP2 (Jumper) es el “puente” de conexión entre la tarjeta del modulo de comunicación USB y el microcontrolador ADUC841.

A continuación los diseños del circuito impreso del microcontrolador TUSB3410 por ambas caras.

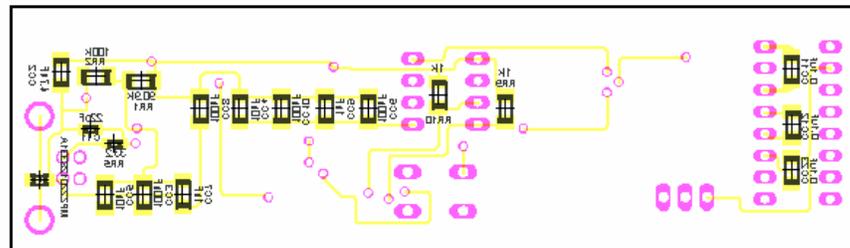
### 2.2.6.2 Diseño del Circuito Impreso

Figura 24. Cara Superior del Circuito Impreso (Bloque Microcontrolador TUSB3410 y Fuente de Alimentación por el Puerto USB):



Fuente: Autores

Figura 25. Cara Inferior del Circuito Impreso (Bloque Microcontrolador TUSB3410 y Fuente de Alimentación por el Puerto USB):



Fuente: Autores

## 2.3 SOFTWARE

Las Herramientas de software utilizadas para el desarrollo de la tesis fueron el *Aspire* y *Labview* (Lenguaje G). El *Aspire* fue utilizado para la realización del programa del microcontrolador ADUC 841 y el lenguaje *Labview* para realizar el interfaz de usuario.

### 2.3.1 Aspire

El ambiente de desarrollo integrado *Aspire* integra todas las herramientas necesarias para editar, ensamblar y depurar código. El sistema de desarrollo integrado Quickstart soporta solamente depuración en lenguaje ensamblador.

El *Aspire* tiene incorporado el ensamblador Metalink 8051 Cross Assembler, que toma un archivo fuente creado con un editor de texto y guardado en lenguaje ensamblador (extensión .asm) y crea dos archivos, un archivo de lista de salida (.lst) y un archivo de objeto de lenguaje de máquina en formato estándar Hexadecimal de Intel. (.Hex).

El archivo de lista de salida muestra los resultados de la operación del ensamblador incluyendo cualquier error de sintaxis u otro error presente en el código fuente original.

El archivo en formato estándar hexadecimal de Intel es usado para programar el microcontrolador por medio del modo de descarga serial que contiene el *Aspire*.

Para el desarrollo de cualquier aplicación el *aspire* incluye soporte para proyectos, ventanas convenientes para ver y modificar los datos, ventanas de interfaz múltiples con editor de texto.

El *Aspire* en conjunto con la familia de microcontroladores ADuC8XX posee la valiosa ventaja de “programming in-circuit”, lo que se traduce como depuración de código en tiempo real. Esta importante característica permite monitorear registro de funciones especiales (SFR), bloques de memoria, estado de bits importantes, variables internas y periféricos, permitiendo ahorrar tiempo y esfuerzo en el desarrollo de aplicaciones.

### 2.3.2 Labview

Labview es un lenguaje de programación gráfica que utiliza iconos en lugar de líneas de texto para crear aplicaciones. En contraste con la programación basada en el lenguaje de programación en texto, donde las instrucciones determinan la ejecución del programa, LabVIEW utiliza programación de flujo de datos, donde el flujo de datos determina la ejecución.

En Labview, se construye una interfaz de usuario con un conjunto de herramientas y objetos. Labview maneja una interfaz de usuario que es también conocida como *panel frontal*. Se puede agregar código usando representaciones gráficas de funciones para controlar los objetos del panel frontal, donde el diagrama de bloques contiene este código. Dicho de otra manera, el diagrama de bloques se parece al un diagrama de flujo.

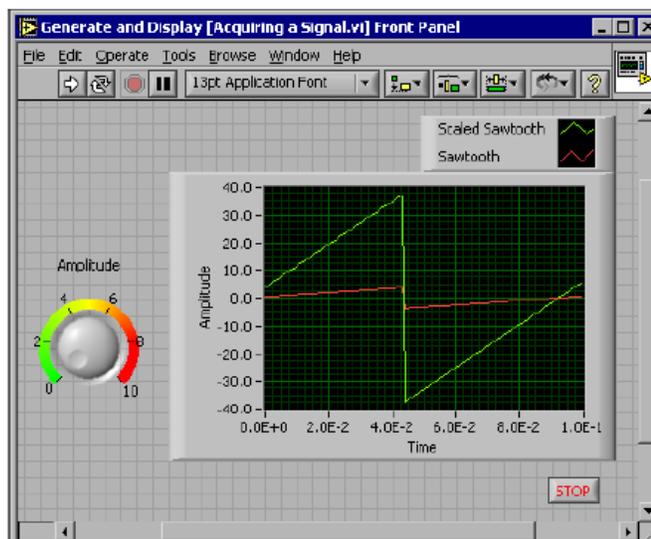
Labview usa unas plantillas llamadas VIs (Instrumento Virtual), llamadas así, debido a su apariencia física y al funcionamiento de imitar instrumentos, como son los osciloscopios y multímetros.

Cada VI utiliza funciones que manipulan la entrada desde la interfaz de usuario o de otras fuentes y mostrar la información o moverla a otros archivos o a otros equipos.

Una VI contiene los siguientes 3 componentes:

- *Panel Frontal:* Cumple la función de la de interfaz de usuario.
- *Diagrama de Bloques:* Contiene código fuente que define la funcionalidad de la VI.
- *Icono y Panel conector:* Identifica la VI a fin de que puede utilizar la VI en otro VI. Un VI dentro de otro VI que se denomina una subVI. Un subVI Corresponde a una subrutina en el texto basado en los lenguajes de programación.

Figura 26. Ejemplo de panel frontal



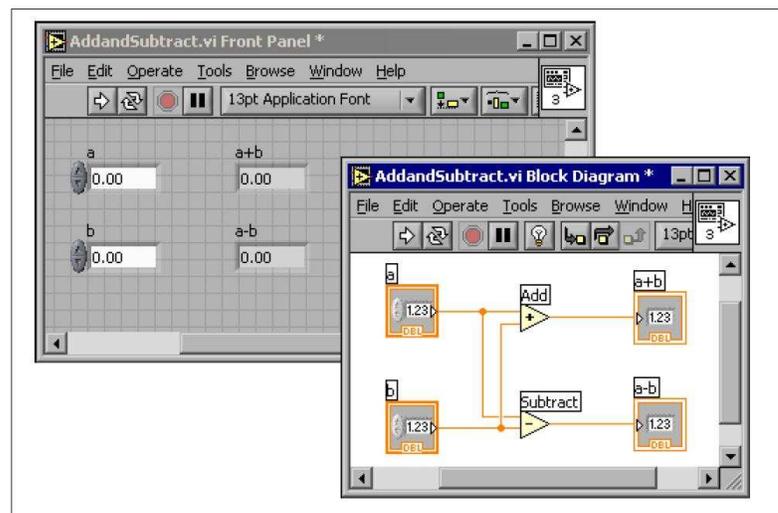
Fuente: Datasheet Labview User Manual

Un panel frontal se puede construir con controles e indicadores, que son los terminales interactivos de entrada y salida de la VI, respectivamente. Los controles son perillas, botones, diales, y otros dispositivos de entrada. Los indicadores son gráficos, LED, y otras pantallas. Los instrumentos de control simulan la entrada de datos al dispositivo para el diagrama de bloque de la VI. Los Instrumentos

indicadores simulan la salida de datos del dispositivo y lo muestran en el diagrama de bloques.

El diagrama de bloques contiene el código fuente. En el panel frontal aparecen los objetos como terminales sobre el diagrama de bloques.

Figura 27. Ejemplo de Diagrama de Bloque y su correspondiente Panel Frontal.



Fuente: Datasheet Labview User Manual

- Los terminales representan el tipo de dato del control o el indicador. Se Puede configurar los controles del panel frontal, los indicadores que aparecen como iconos, o los terminales del tipo de dato que esta sobre el diagrama de bloque.  
Los terminales son entradas y salidas de puertos que intercambian información entre el panel frontal y el diagrama de bloques que adquiere o genera.

Figura 28. Ejemplo de un Terminal.



Fuente: Datasheet Labview User Manual

- Los nodos son objetos sobre el diagrama de bloques que tienen entradas y / O salidas y realizan operaciones cuando la VI se está ejecutando. En la figura 26 Los puntos que están en la función de suma y resta son nodos.
- Para transferir los datos entre los objetos del diagrama de bloques se hacen a través de los cables. En la figura 26 los cables se conectan a los controladores y a los terminales de los indicadores de la función de suma y resta. Los cables pueden ser de distintos colores o tipo de acuerdo al tipo de dato que se esté manejando.
- Las estructuras son representaciones gráficas de bucles y casos de lenguaje de programación basado en texto. Las estructuras se usan sobre los diagramas de bloques para repetir bloques de código y para ejecutar códigos condicionales o una orden específica.

Los iconos son representaciones gráficas de una VI. Estas pueden contener texto, imágenes o combinaciones de ambas. Si se usa una VI como una subVI el icono identifica la subVI en el diagrama de bloques de la VI. Al dar doble click en el icono se puede personalizar o editar éste.

Figura 29. Ejemplo de Icono.



Fuente: Datasheet Labview User Manual

El panel conector es un conjunto de terminales que corresponden a los controles y a los indicadores de las VIs que, de forma similar a una lista de parámetros de una función llamada por un lenguaje de programación basada en texto.

El panel conector define las entradas y las salidas que se pueden cablear a la VI de modo que puede usarse como una subVI.

Un panel conector recibe los datos en sus terminales de entrada y pasa los datos a él diagrama de bloques de código a través de los controles del panel frontal y recibe los resultados en sus terminales de salida desde el indicador de panel frontal.

Al abrir el panel conector por primera vez, vera un conector patrón, se puede seleccionar un patrón diferente si el usuario lo desea. El panel conector por lo general, tiene un terminal para cada control o indicador en la parte del panel frontal. Se puede asignar hasta 28 terminales a un panel conector.

Figura 30. Ejemplo de Panel Conector



Fuente: Datasheet Labview User Manual

Después de construir una VI y crear un icono y un panel conector, el usuario puede usarla como subVI si solo desea. Además el usuario también puede guardarlas las VIs como archivos individuales, o como un grupo de VIs juntas y guardarlas como una librería

El usuario también puede personalizar la apariencia y el comportamiento de una VI. También se puede crear menús personalizados de varias VIs construidas y también se podrá configurar según como lo requiera la persona.

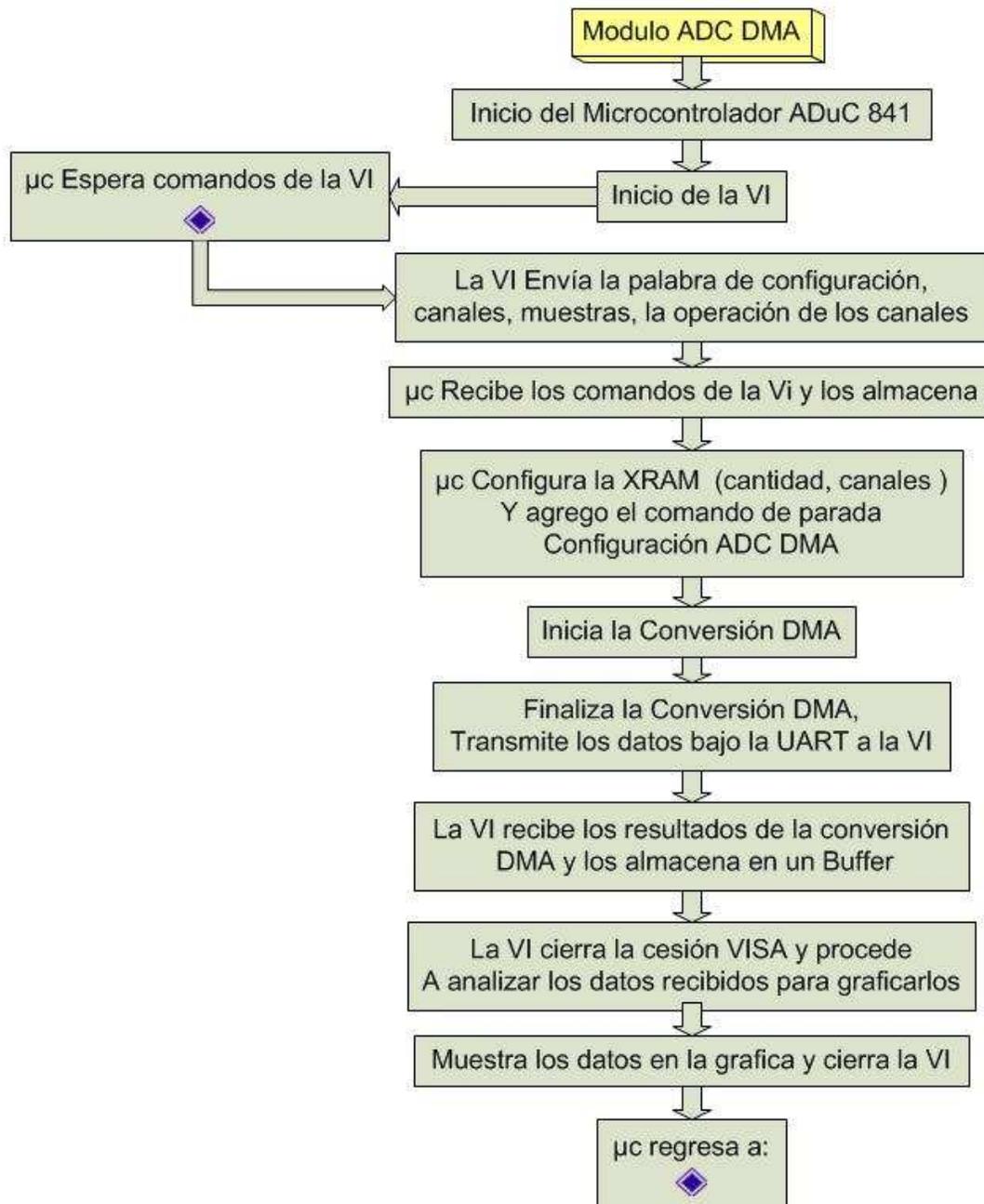
### **2.3.3 FIRMWARE (Software del ADUC y TUSB3410)**

Para las Aplicaciones del microcontrolador TUSB3410 se necesita driver/firmware y el VCP (Puerto Virtual COM). El driver/firmware VCP junto con el TUSB3410 proporciona un “puente” USB/serial para una aplicación USB /UART (Como interfaz de un microcontrolador).Al descargar el firmware en la EEPROM en la board del modulo USB, se procede a la instalación del VCP Driver/Firmware del TUSB3410.

Para mayor información ver anexo **A. Manual de Usuario para Instalar el Entrenador Digital USB.**

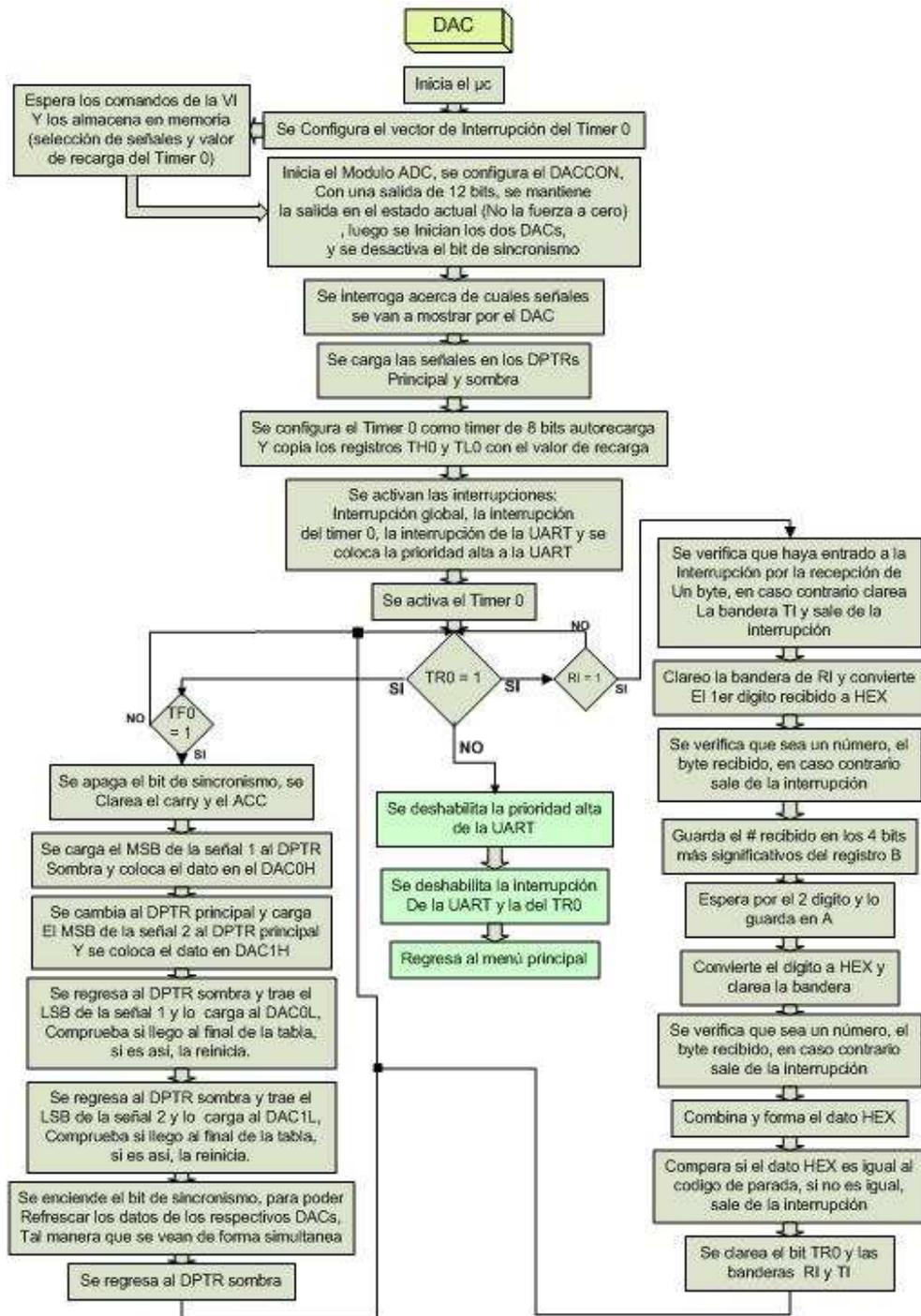
### 2.3.3.1 Diagramas de Bloques de la Programación

Figura31. Diagrama de Bloques de ADC DMA



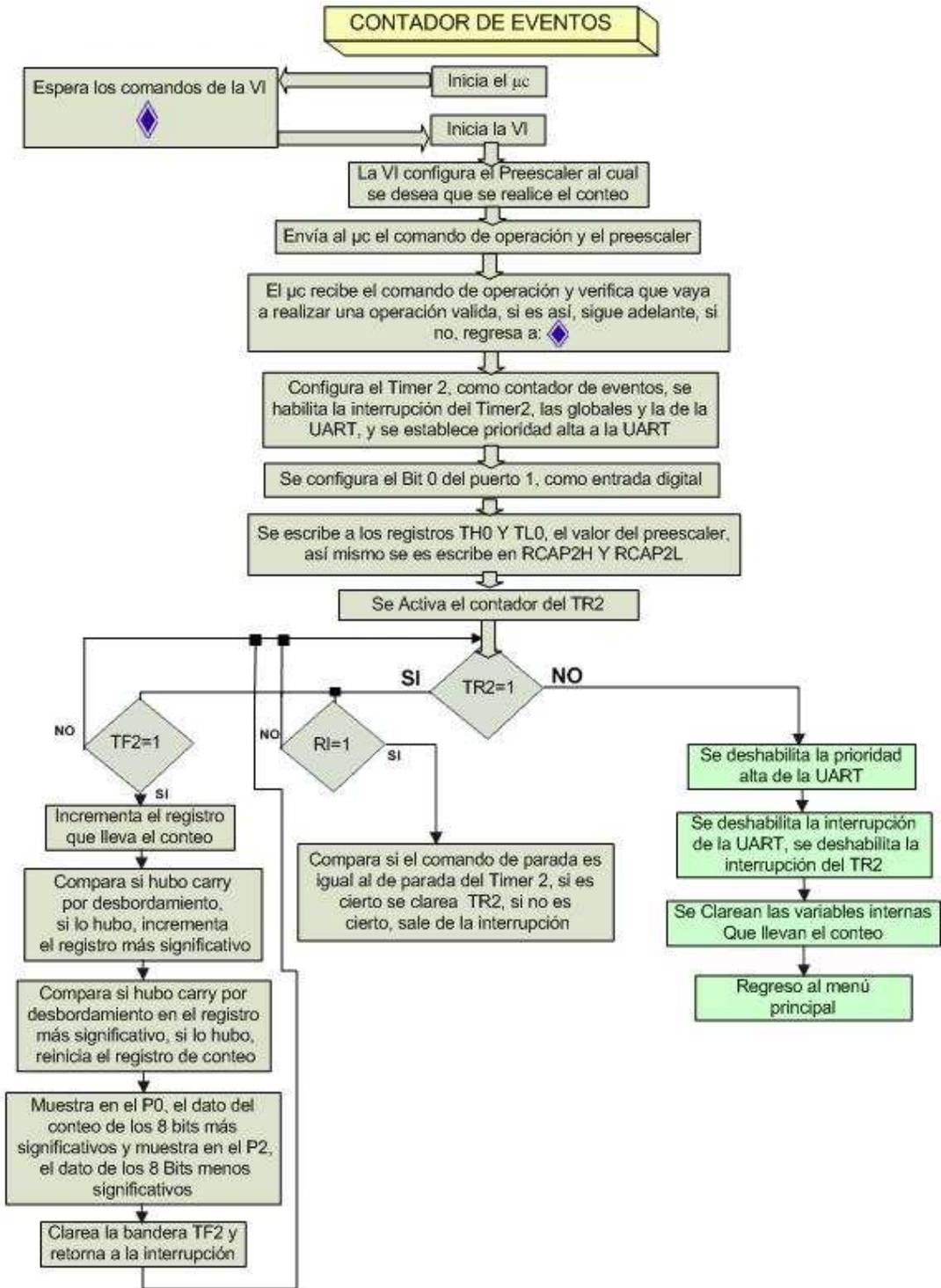
Fuente: Autores

Figura32. Diagrama de Bloques DAC



Fuente: Autores

Figura33. Diagrama de Bloques de Contador de Eventos.



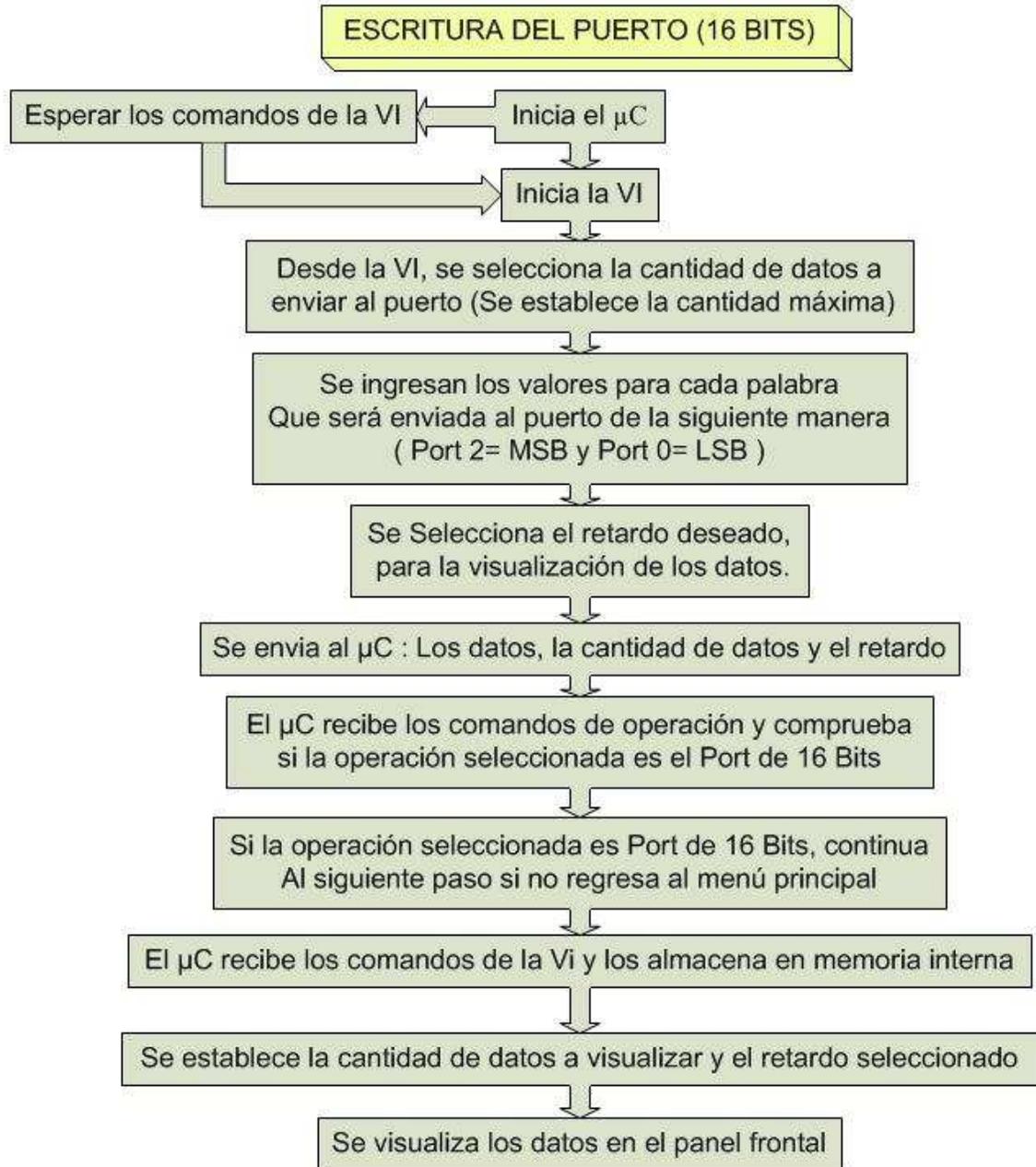
Fuente: Autores

Figura34. Diagrama de Bloques Escritura del Puerto 8 Bits.



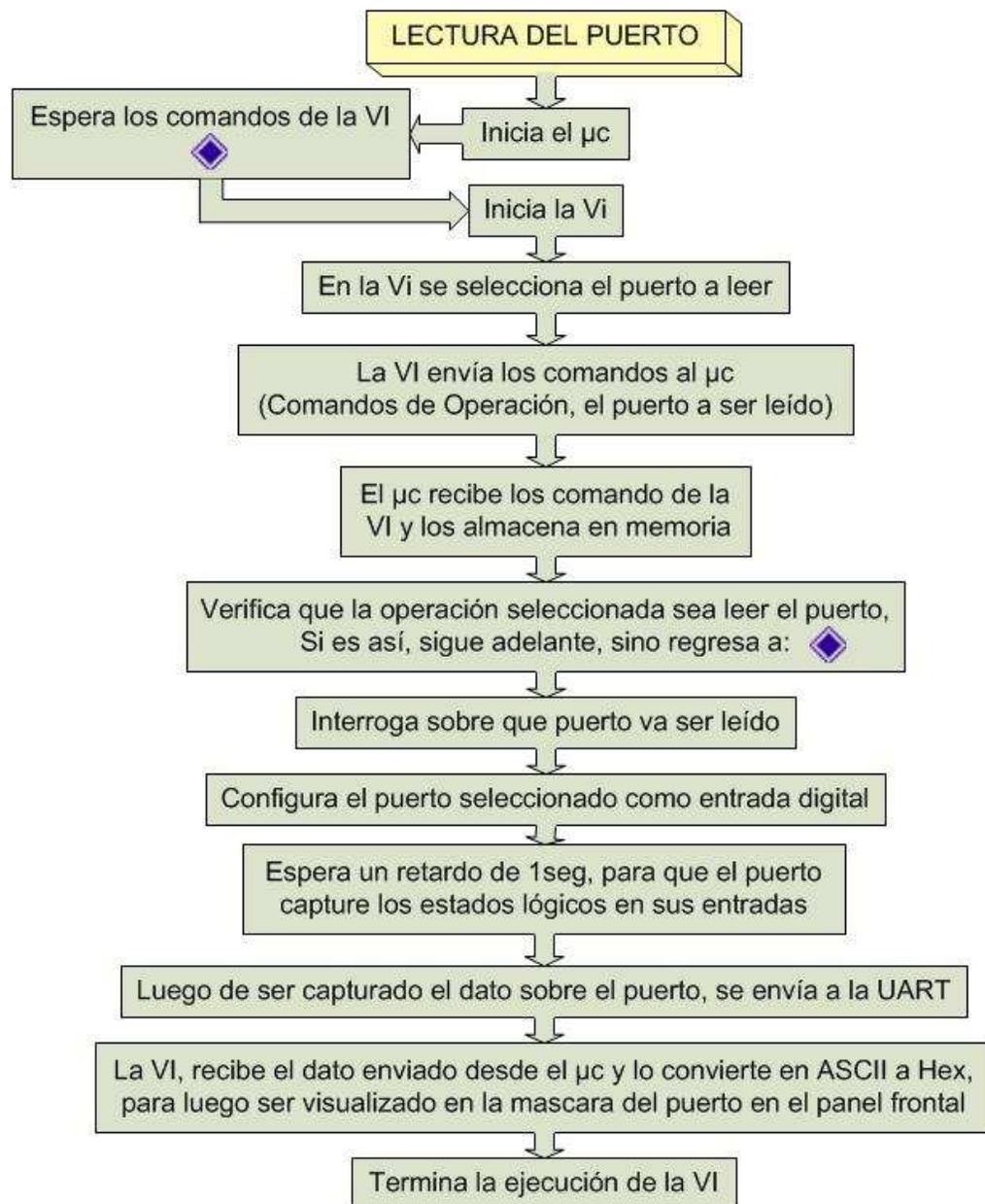
Fuente: Autores

Figura35. Diagrama de Bloques Escritura del Puerto 16 Bits.



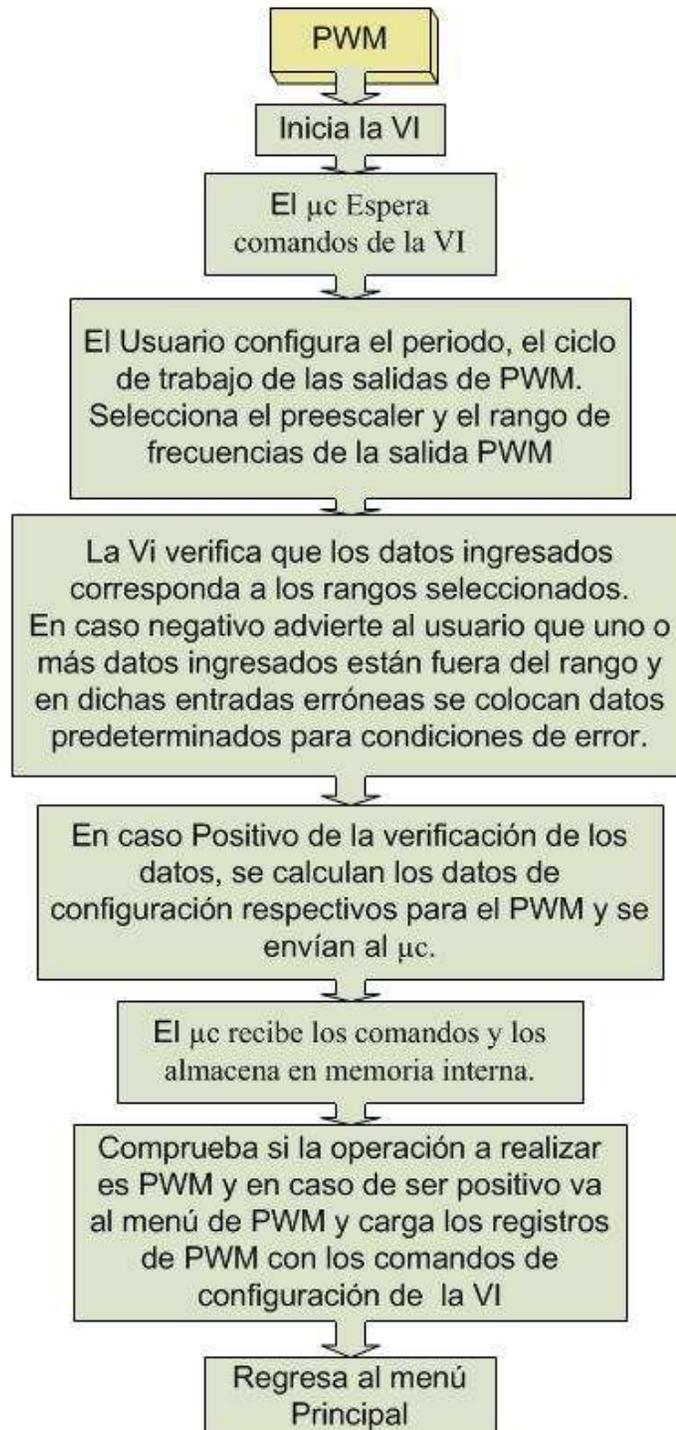
Fuente: Autores

Figura36. Diagrama de Bloques Lectura del Puerto.



Fuente: Autores

Figura37. Diagrama de Bloques PWM.



Fuente: Autores

### 2.3.3.2 Interfaz de Usuario en LABVIEW

Figura38. Interfaz de Menú Principal



Fuente: Autores

En el menú desplegable “**Menú Principal**” se encuentran los distintos módulos que componen al entrenador digital. El usuario puede seleccionar cada módulo desde el menú desplegable “**Menú Principal**”.

En el menú desplegable “**Seleccione Puerto Serie**” se escoge el puerto virtual al cual el sistema operativo asocia el driver del TUSB3410.

Para ver la asociación que se le asignó al driver del TUSB3410 por parte del sistema operativo anexo **A. Manual de Usuario para Instalar el Entrenador Digital USB.**

Comportamiento del interruptor “**START**”:

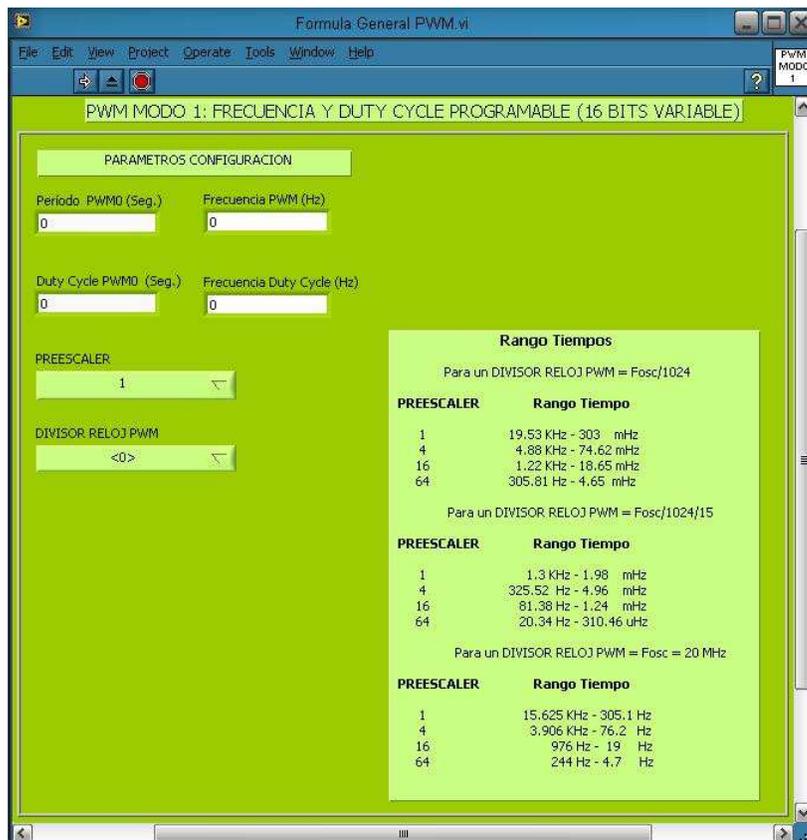
Cuando el interruptor se encuentra en la posición “**START**” La VI del menú principal ejecuta el módulo seleccionado en el menú desplegable “**Menú Principal**”.

Comportamiento del interruptor “**STOP**”:

Cuando el interruptor se encuentra en la posición “**STOP**”, la Vi del menú principal envía al microcontrolador el comando de parada necesario para finalizar el lazo infinito en que se encuentra el microcontrolador en los módulos de contador de eventos y DAC.

Esto se realiza con el fin de retornar el microcontrolador al menú principal del programa interno.

Figura39. Interfaz de Usuario PWM Modo 1



Fuente: Autores

En el control numérico “**Periodo PWM0**” el usuario ingresa el valor del periodo deseado.

En el control numérico “**Duty Cycle PWM0**” el usuario ingresa el valor del ciclo de trabajo deseado.

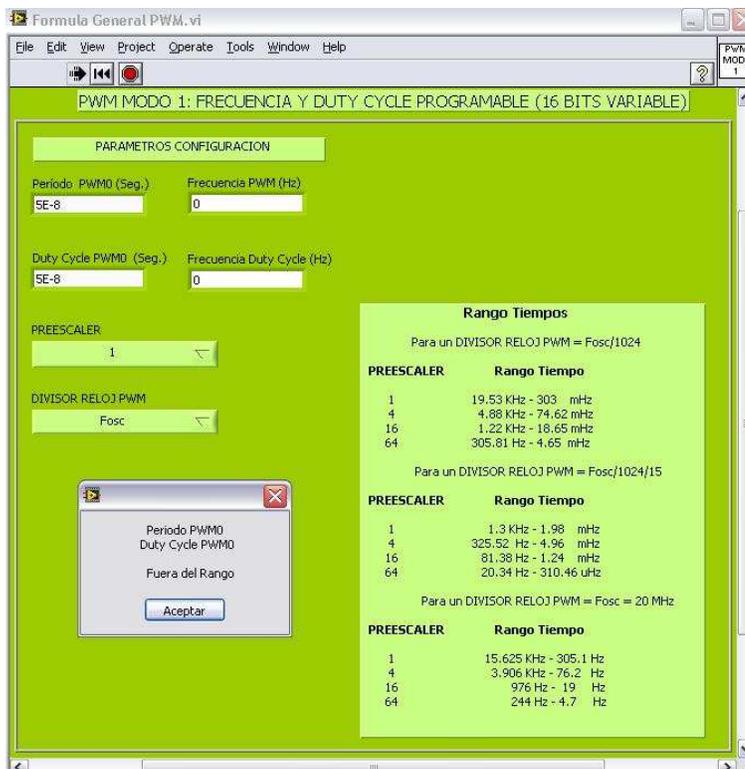
En el menú desplegable “**PREESCALER**” el usuario puede seleccionar las distintas opciones de posibles frecuencias alcanzadas por el entrenador.

En el menú desplegable “**DIVISOR DE RELOJ PWM**” el usuario puede seleccionar los diferentes rangos de frecuencia alcanzados por el entrenador y mostrados en la tabla adjunta.

En los indicadores numéricos “**Frecuencia PWM0**” y “**Frecuencia Duty Cycle**” el usuario visualiza el valor de la frecuencia, luego de haberse ajustado al rango de frecuencias más cercano que el modo de PWM puede alcanzar.

Mensajes posibles de error:

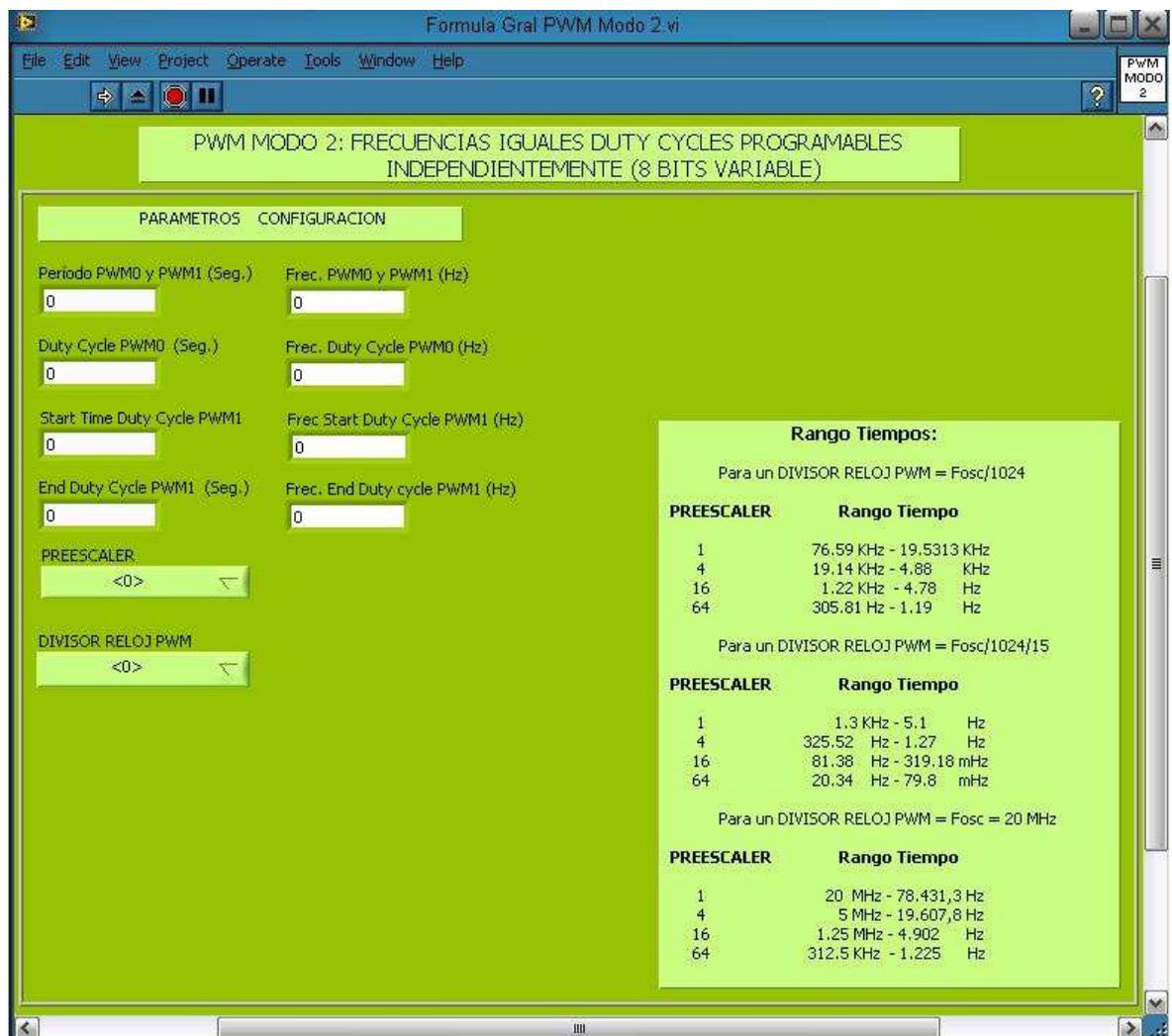
Figura40. Ejemplo Mensaje de Error PWM Modo 1.



Fuente: Autores

Ante una entrada en los controles numéricos “**Periodo PWM0**” y “**Duty Cycle PWM0**” que se encuentre por fuera de los rangos permitidos por la tabla, el usuario será advertido con una ventana de error emergente la cual le señalará el control o los controles que contienen valores por fuera del rango permitido. Adicional a lo anterior, la VI preestablecerá el valor límite para el respectivo control.

Figura41. Interfaz de Usuario PWM Modo 2



Fuente: Autores

En el control numérico “**Periodo PWM0 y PWM1**” el usuario ingresa el valor del periodo deseado para los dos PWMs.

En el control numérico “**Duty Cycle PWM0**” el usuario ingresa el valor del ciclo de trabajo deseado.

En el control numérico “**Start Time Duty Cycle PWM1**” el usuario ingresa el valor de inicio del ciclo de trabajo deseado.

En el control numérico “**End Duty Cycle PWM1**” el usuario ingresa el valor de parada del ciclo de trabajo deseado.

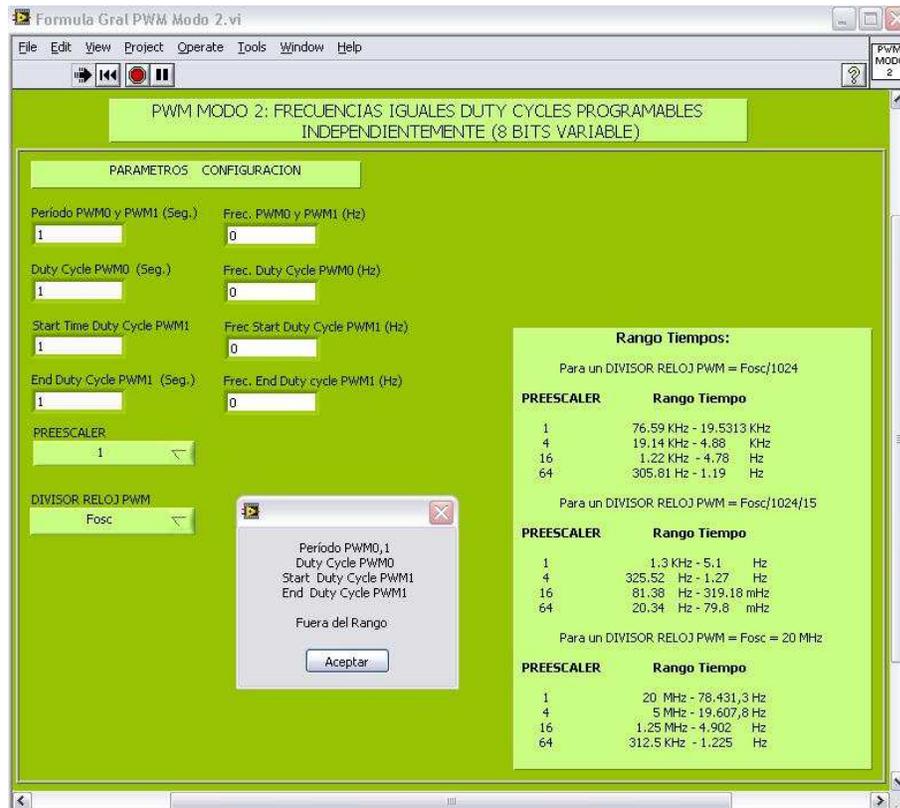
En el menú desplegable “**PREESCALER**” el usuario puede seleccionar las distintas opciones de posibles frecuencias alcanzadas por el entrenador.

En el menú desplegable “**DIVISOR DE RELOJ PWM**” el usuario puede seleccionar los diferentes rangos de frecuencia alcanzados por el entrenador y mostrados en la tabla adjunta.

En los indicadores numéricos “**Frecuencia PWM0 y PWM1**”, “**Frecuencia Duty Cycle PWM0**” “**Frec. Start Time Duty Cycle PWM1**” y “**Frec. End Duty Cycle PWM1**” el usuario visualiza el valor de la frecuencia, luego de haberse ajustado al rango de frecuencias más cercano que el modo de PWM puede alcanzar.

Mensajes posibles de error:

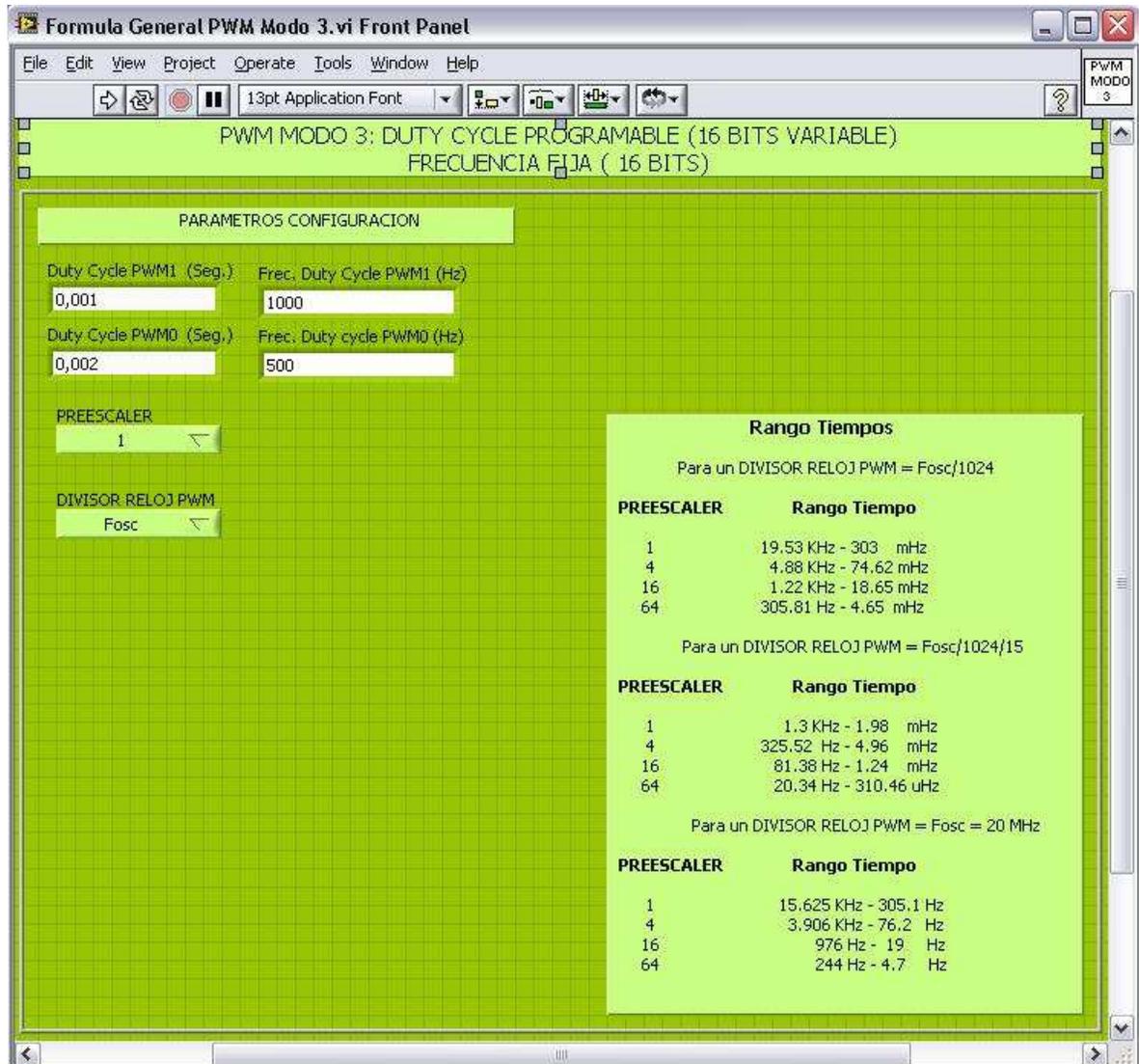
Figura42. Ejemplo Mensaje de Error PWM Modo 2.



Fuente: Autores

Ante una entrada en los controles numéricos “**Periodo PWM0 y PWM1**”, “**Duty Cycle PWM0**”, “**Start Time Duty Cycle PWM1**” y “**End Duty Cycle PWM1**” que se encuentre por fuera de los rangos permitidos por la tabla, el usuario será advertido con una ventana de error emergente la cual le señalará el control o los controles que contienen valores por fuera del rango permitido. Adicional a lo anterior, la VI preestablecerá el valor límite para el respectivo control.

Figura43. Interfaz de Usuario PWM Modo 3



Fuente: Autores

En el control numérico “**Duty Cycle PWM1**” el usuario ingresa el valor del ciclo de trabajo deseado.

En el control numérico “**Duty Cycle PWM0**” el usuario ingresa el valor del ciclo de trabajo deseado.

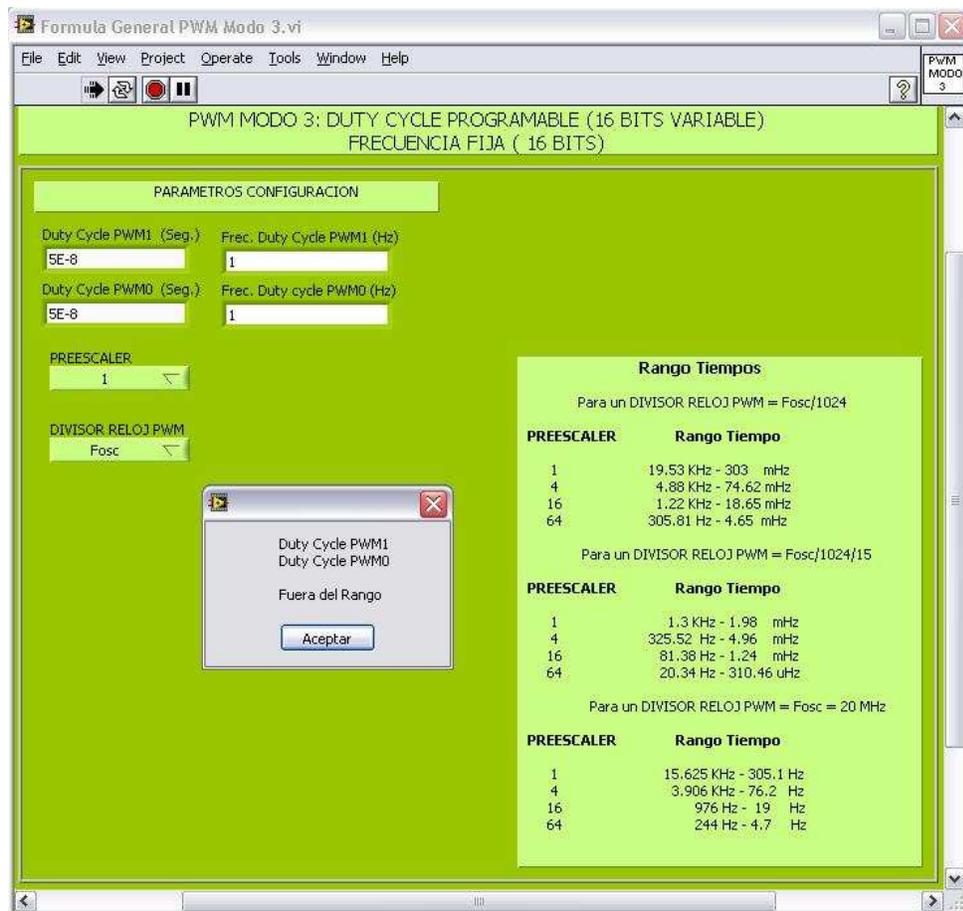
En el menú desplegable “**PREESCALER**” el usuario puede seleccionar las distintas opciones de posibles frecuencias alcanzadas por el entrenador.

En el menú desplegable “**DIVISOR DE RELOJ PWM**” el usuario puede seleccionar los diferentes rangos de frecuencia alcanzados por el entrenador y mostrados en la tabla adjunta.

En los indicadores numéricos “**Frec. Duty Cycle PWM1**” y “**Frec. Duty Cycle PWM0**” el usuario visualiza el valor de la frecuencia, luego de haberse ajustado al rango de frecuencias más cercano que el modo de PWM puede alcanzar.

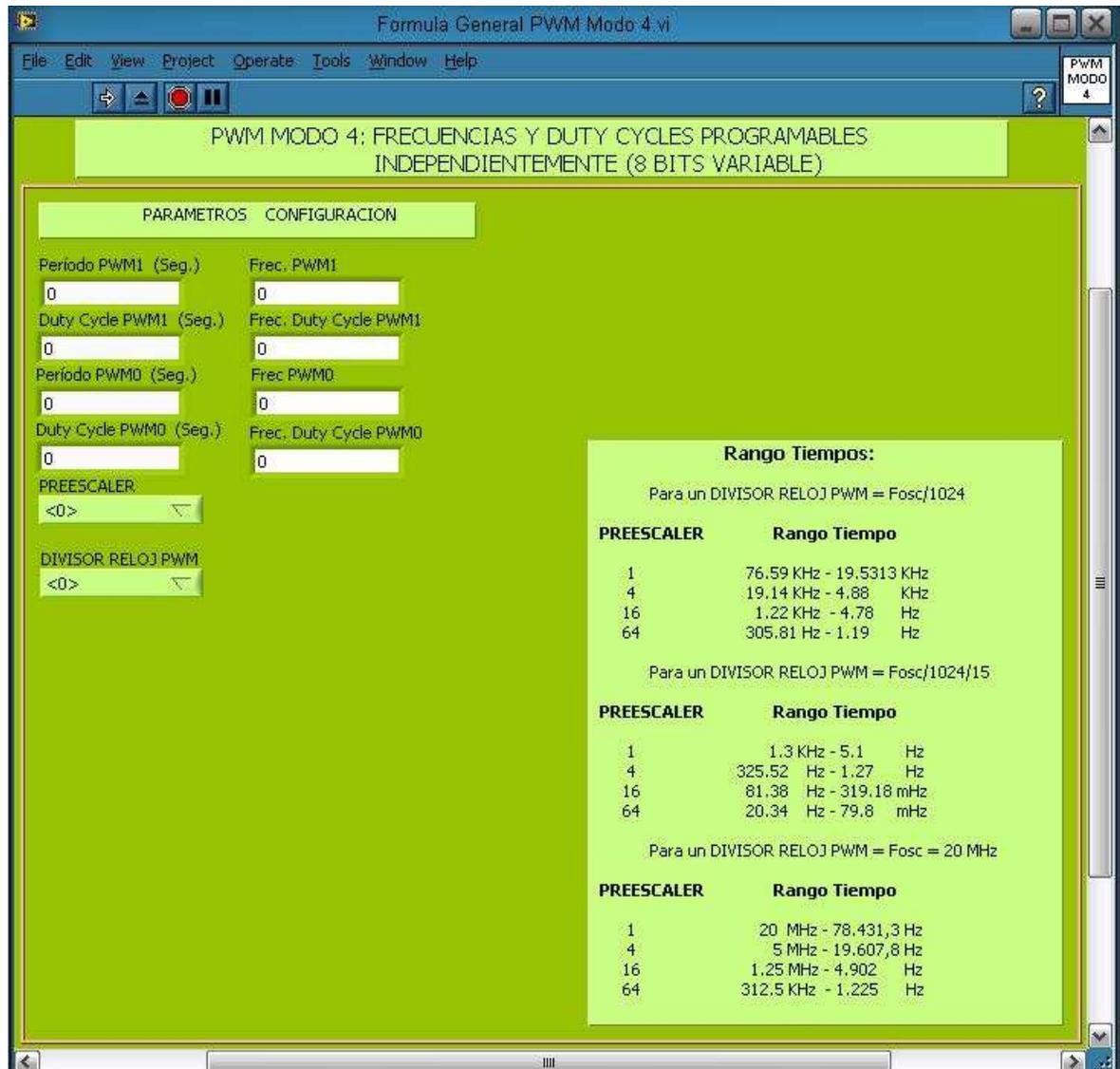
Mensajes posibles de error:

Figura44. Ejemplo Mensaje de Error PWM Modo 3.



Fuente: Autores

Figura45. Interfaz de Usuario PWM Modo 4



Fuente: Autores

En el control numérico “**Periodo PWM1**” el usuario ingresa el valor del periodo deseado para el PWM1.

En el control numérico “**Duty Cycle PWM1**” el usuario ingresa el valor del ciclo de trabajo deseado.

En el control numérico “**Periodo PWM0**” el usuario ingresa el valor del periodo deseado para el PWM0.

En el control numérico “**Duty Cycle PWM0**” el usuario ingresa el valor del ciclo de trabajo deseado.

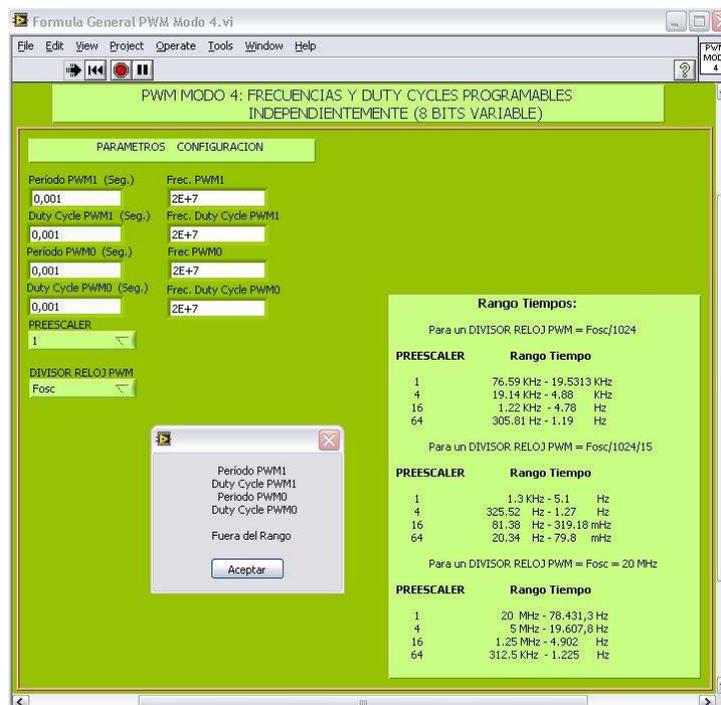
En el menú desplegable “**PREESCALER**” el usuario puede seleccionar las distintas opciones de posibles frecuencias alcanzadas por el entrenador.

En el menú desplegable “**DIVISOR DE RELOJ PWM**” el usuario puede seleccionar los diferentes rangos de frecuencia alcanzados por el entrenador y mostrados en la tabla adjunta.

En los indicadores numéricos “**Frec. PWM1**”, “**Frec. Duty Cycle PWM1**”, “**Frec. PWM0**” y “**Frec. Duty Cycle PWM0**” el usuario visualiza el valor de la frecuencia, luego de haberse ajustado al rango de frecuencias más cercano que el modo de PWM puede alcanzar.

Mensajes posibles de error:

Figura46. Ejemplo Mensaje de Error PWM Modo 4.



Fuente: Autores

Figura47. Interfaz de Usuario ADC en modo DMA



Fuente: Autores

En el indicador numérico “**Frecuencia Muestreo**” indica la cantidad de muestras por segundos a la cual se muestrea cada canal seleccionado.

En el menú desplegable “**MCLK Divider**” se puede seleccionar los distintos periodos de muestreo disponibles.

En el menú desplegable “**Nº ADC Clock**” se puede seleccionar el número de tiempos de adquisición para el amplificador de muestreo y retención de la señal de entrada. Una adquisición de 3 ó más (ADC Clock’s) es recomendada por el fabricante.

En los Controles Booleanos “**Canal 0... 7 OFF/ON**” se seleccionan los canales a muestrear.

Figura48. Interfaz de Usuario de Puertos 8 Bits



Fuente: Autores

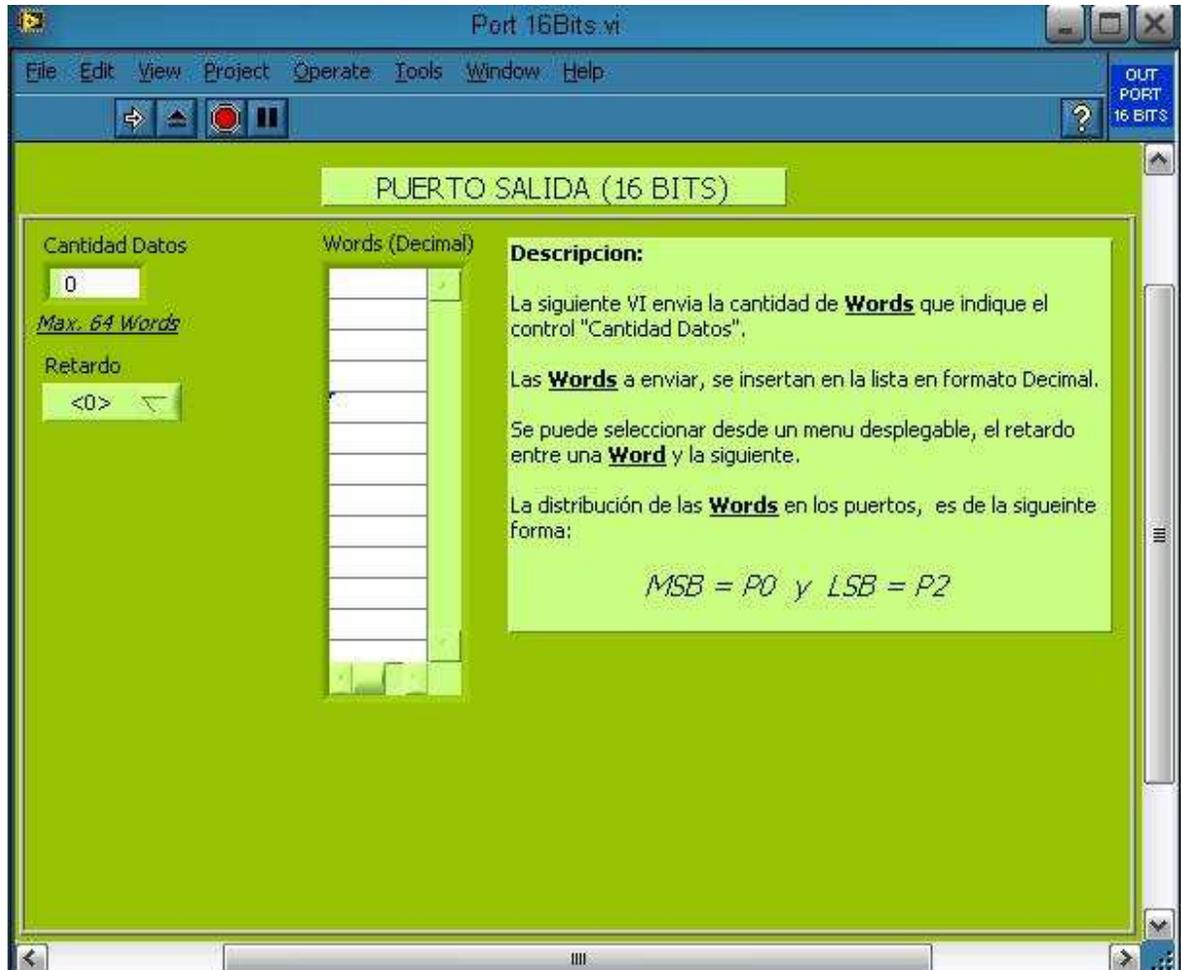
En el control numérico “**Cantidad Datos**” se ingresa el número de datos que van a ser enviados al puerto I/O. La cantidad máxima de datos a ser enviados es de 128 bytes.

En el menú desplegable “**Retardo**” se selecciona el tiempo de retardo deseado para visualizar los datos en el puerto I/O.

En el menú desplegable “**Puerto Salida**” se selecciona el puerto al cual los datos van a ser enviados para posteriormente ser visualizados.

En la lista de control “**Datos (Decimal)**” se ingresan los datos que van a ser visualizados en el puerto seleccionado. Estos datos deben ser ingresados en notación decimal.

Figura49. Interfaz de Usuario de Puertos 16 Bits



Fuente: Autores

En el control numérico “**Cantidad Datos**” se ingresa el número de datos que van a ser enviados al puerto I/O. La cantidad máxima de datos a ser enviados es de 64 bytes.

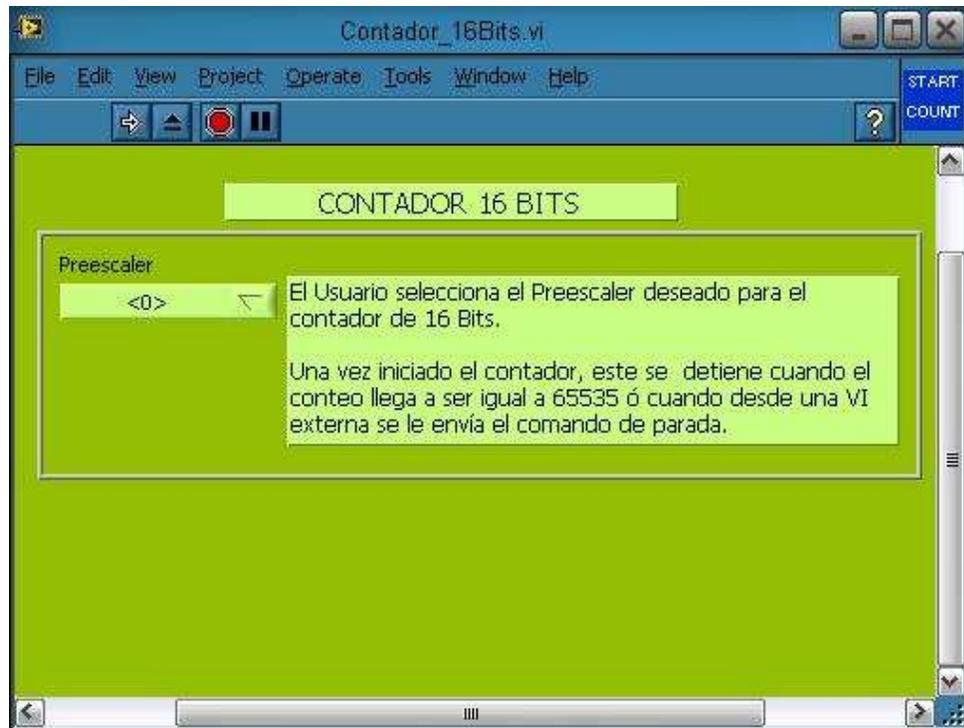
En el menú desplegable “**Retardo**” se selecciona el tiempo de retardo deseado para visualizar los datos en el puerto I/O.

En la lista de control “**Words (Decimal)**” se ingresan las palabras que van a ser visualizadas de la siguiente manera:

- Los 8 bits menos significativos de la palabra se visualizan en el puerto 2

- Los 8 bits más significativos de la palabra se visualizan en el puerto 0.

Figura50. Interfaz de Usuario de Contador de Eventos 16 Bits.

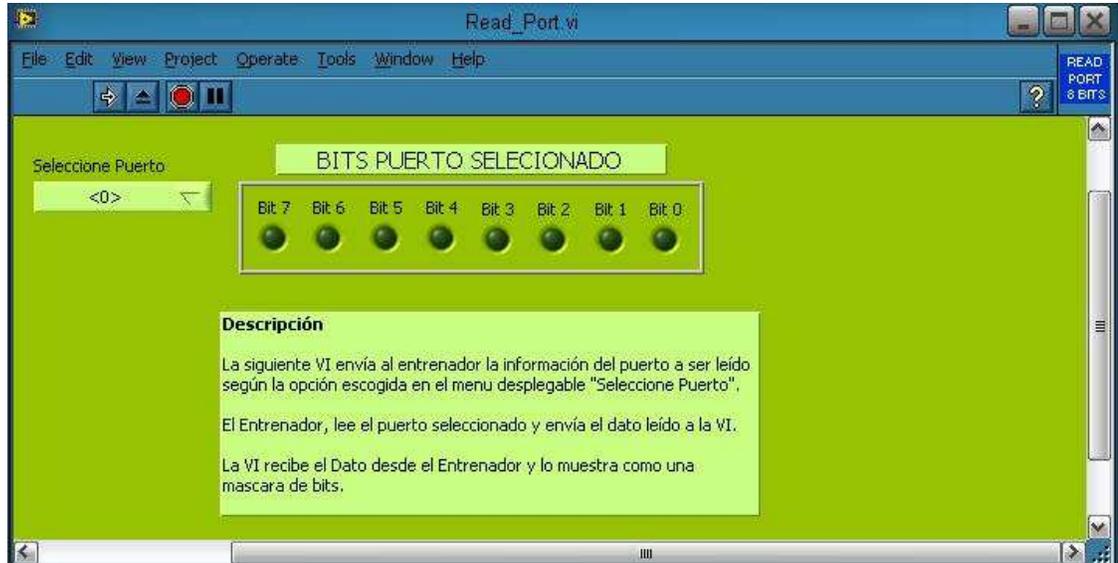


Fuente: Autores

En el menú desplegable “**Preescaler**” se selecciona cada cuantas transiciones del estado lógico *alto* al estado lógico *bajo* se incrementa el contador, y se visualiza el conteo realizado en los puertos I/O con el mismo formato de configuración que se visualizan los datos de 16 bits.

Cuando el contador llega al máximo (65535) se detiene el conteo y espera que el usuario reinicie el módulo del contador. Esto se realiza enviando el comando de parada desde la VI del menú principal.

Figura51. Interfaz de Usuario de Lectura del Puerto.



Fuente: Autores

En el menú desplegable “**Seleccione Puerto**” se escoge el puerto que va a ser leído. El dato leído en el puerto es enviado del microcontrolador a la VI para ser mostrado en la máscara de bits.

Figura52. Interfaz de Usuario de DAC 12 Bits



Fuente: Autores

En el menú desplegable “**Selección Ondas**” se escoge el par de ondas a ser visualizadas en las salidas de los DAC’s.

En el menú desplegable “**Frecuencia Salida DAC**” se escoge la frecuencia a la cual el par de ondas seleccionadas van a ser generadas.

## **CAPITULO 3. RESULTADOS OBTENIDOS**

### **3.1 Capa Software**

Se implemento un modulo de conversión ADC en modo DMA (acceso Directo a memoria), es decir, la conversión realiza un muestreo de la Señal a una frecuencia determinada y con una cantidad de muestras preestablecidas.

El modulo posee las siguientes características:

- Ofrece al usuario la selección de la cantidad de muestras por segundo a la cual desee muestrear los canales deseados.
- Ofrece al usuario la selección de distintos periodos de muestreo disponibles.
- Ofrece al usuario la selección del número de tiempos de adquisición para la conversión de la señal de entrada.
- Posee hasta ocho canales de muestreo para las necesidades del usuario.

A lo anterior a dichas características, éstas se ven reflejadas en la interfaz realizada a dicho modulo en Labview para cualquiera sea la necesidad del usuario.

El microcontrolador se programó de tal manera que realiza la conversión ADC en modo DMA. Finalizada la conversión los resultados son enviados a la vi por medio de la UART, para finalmente regresar al menú principal del programa interno del microcontrolador. La vi envía los comandos de configuración de la conversión y espera un tiempo prudente antes de abrir el puerto serie para permitir al microcontrolador ejecutar la acción, posteriormente abre el puerto serie y espera los resultados de la conversión dentro de los cuales esta insertado el byte de terminación de carácter que finaliza la comunicación y cierra el puerto serie. Luego de haberse cerrado el puerto serie se procesan los datos para ser visualizados en las graficas de la vi.

Para la implementación del módulo ADC se probó los distintos modos de operación los cuales son: conversión ADC en modo continuo y conversión en modo DMA (acceso directo a memoria). Además de los modos anteriores existen el modo de conversión única, modo de conversión por medio de una señal externa de disparo y modo de conversión por medio del timer 2.

En el desarrollo de la interfaz gráfica se descubrió un factor limitante en la implementación del módulo ADC, dicha limitación hace referencia a la función de Labview empleada que administra el puerto serie. Esta función determina un tiempo máximo de apertura del puerto que impide al usuario mantener una conversión continua por medio del puerto serie de manera indefinida. Debido a esta limitación se recurrió a la implementación del modo DMA. La cantidad de muestras preestablecidas es de 1022 y se reparte en la cantidad de canales seleccionados afectando la frecuencia de muestreo. Esto se evidencia de la siguiente manera:

- 1 canal: 1022 muestras de la señal para el canal.
- 2 canales: 511 muestras de la señal por canal.
- 3 canales: 340 muestras de la señal por canal.
- 4 canales: 255 muestras de la señal por canal.
- 5 canales: 204 muestras de la señal por canal.
- 6 canales: 170 muestras de la señal por canal.
- 7 canales: 146 muestras de la señal por canal.
- 8 canales: 127 muestras de la señal por canal.

Figura53. Tabla de Muestras1 Onda Seno.

A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P
DIVISION/CUADRANTE	PUNTOS/CUADRANTE	ANGULO MINIMO		3 Muestras	5 Muestras	9 Muestras	17 Muestras	33 Muestras	65 Muestras	3 Muestras	5 Muestras	9 Muestras	17 Muestras	33 Muestras	65 Muestras
0	1	90		90º	45º	22,5º	11,25º	5,625º	2,8125º	90º	45º	22,5º	11,25º	5,625º	2,8125º
1	2	45		0	0	0	0	0	0	0801	0801	0801	0801	0801	0801
2	4	22,5		45	22,5	11,25	5,625	2,8125	1,40625	0A44	093A	08A1	0851	0829	0815
3	8	11,25		90	45	22,5	11,25	5,625	2,8125	0B34	0A44	093A	08A1	0851	0829
4	16	5,625			90	45	22,5	11,25	5,625		0A44	093A	08A1	0851	0829
5	32	2,8125				90	45	22,5	11,25			0A44	093A	08A1	0851
6	64	1,40625					90	45	22,5				0A44	093A	08A1
7	128	0,703125						90	45					0A44	093A
8	256	0,3515625							90						0A44
9															
10															
11															
12															
13															
14															
15															
16															
17															
18															
19															
20															
21															
22															
23															
24															
25															
26															
27															
28															
29															
30															
31															
32															
33															
34															
35															
36															
37															

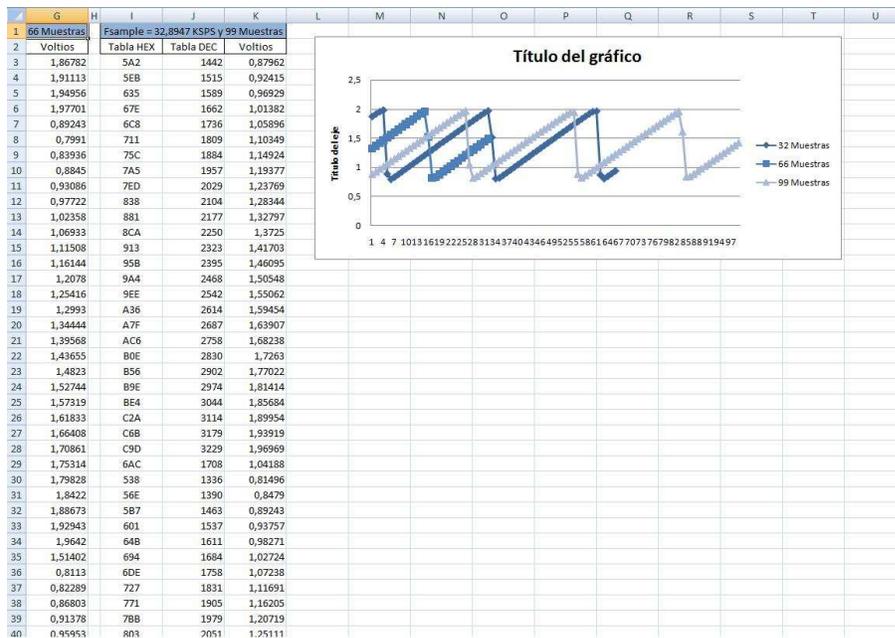
Fuente: Autores

Figura54. Tabla de Muestras2 Onda Seno.

Q	R	S	T	U	V	W	X	Y	Z	AA	AB
3 Muestras	5 Muestras	9 Muestras	17 Muestras	33 Muestras	65 Muestras	3 Muestras	5 Muestras	9 Muestras	17 Muestras	33 Muestras	65 Muestras
-90	-45	-22,5	-11,25	-5,625	-2,8125	-90	-45	-22,5	-11,25	-5,625	-2,8125
0	0	0	0	0	0	0801	0801	0801	0801	0801	0801
315	337,5	348,75	354,375	357,1875	358,59375	05BD	06C7	0761	0780	07D8	07ED
270	315	337,5	348,75	354,375	357,1875	04CD	05BD	06C7	0761	0780	07D8
	292,5	326,25	343,125	351,5625	355,78125		050B	0639	0713	0788	07C4
	270	315	337,5	348,75	354,375		04CD	05BD	06C7	0761	0780
		303,75	331,875	345,9375	352,96875			0557	067E	073A	079C
		292,5	326,25	343,125	351,5625			050B	0639	0713	0788
		281,25	320,625	340,3125	350,15625			04DD	05F9	06E6	0775
		270	315	337,5	348,75			04CD	05BD	06C7	0761
			309,375	334,6875	347,34375				0587	06A2	074D
			303,75	331,875	345,9375				0557	067E	073A
			298,125	329,0625	344,53125				052E	0658	0726
			292,5	326,25	343,125				050B	0639	0713
			286,875	323,4375	341,71875				04F0	0618	0700
			281,25	320,625	340,3125				04DD	05F9	06ED
			275,625	317,8125	338,90625				04D1	05DA	06DA
			270	315	337,5				04CD	05BD	06C7
				312,1875	336,09375					05A1	06B5
				309,375	334,6875					0587	06A2
				306,5625	333,28125					056E	0690
				303,75	331,875					0557	067E
				300,9375	330,46875					0542	066D
				298,125	329,0625					052E	0658
				295,3125	327,65625					051C	064A
				292,5	326,25					050B	0639
				289,6875	324,84375					04FD	0629
				286,875	323,4375					04F0	0618
				284,0625	322,03125					04E6	0608
				281,25	320,625					04DD	05F9
				278,4375	319,21875					04DE	05E9
				275,625	317,8125					04D1	05DA
				272,8125	316,40625					04CE	05CB
				270	315					04CD	05BD
					313,59375						05AF
					312,1875						05A1
					310,78125						0594
					309,375						0587
					307,96875						057A

Fuente: Autores

Figura55. Muestras Onda Diente de Sierra.



Fuente: Autores

Figura56. Muestras Onda Triangular.

	A	B	C	D	E	F	G	H	I	J	K
				Y1	Y2	Y3				YY1	YY2
1											
2		0		0801	0B34	04CD		0		04CD	0B34
3		1		081A	0B1B	04E7		1		04DB	0A4A
4		2		0834	0B01	0500		2		04E8	0960
5		3		084E	0AE8	051A		3		04F6	0876
6		4		0867	0ACE	0533		4		0503	078C
7		5		0881	0AB4	054D		5		0511	06A1
8		6		089A	0A9B	0567		6		051F	05B7
9		7		08B4	0A81	0580		7		052C	04CD
10		8		08CE	0A67	059A		8		053A	
11		9		08E7	0A4E	05B4		9		0547	
12		10		0901	0A34	05CD		10		0555	
13		11		091A	0A1B	05E7		11		0563	
14		12		0934	0A01	0600		12		0570	
15		13		094E	09E7	061A		13		057E	
16		14		0967	09CE	0634		14		058C	
17		15		0981	09B4	064D		15		0599	
18		16		099B	099B	0667		16		05A7	
19		17		09B4	0981	0680		17		05B4	
20		18		09CE	0967	069A		18		05C2	
21		19		09E7	094E	06B4		19		05D0	
22		20		0A01	0934	06CD		20		05DD	
23		21		0A1B	091A	06E7		21		05EB	
24		22		0A34	0901	0701		22		05F8	
25		23		0A4E	08E7	071A		23		0606	
26		24		0A67	08CE	0734		24		0614	
27		25		0A81	08B4	074D		25		0621	
28		26		0A9B	089A	0767		26		062F	
29		27		0AB4	0881	0781		27		063C	
30		28		0ACE	0867	079A		28		064A	
31		29		0AE8	084E	07B4		29		0658	
32		30		0B01	0834	07CD		30		0665	
33		31		0B1B	081A	07E7		31		0673	
34		32		0B34	0801	0801		32		0680	
35		33			07E7			33		068E	
36		34			07CD			34		069C	
37		35			07B4			35		06A9	
38		36			079A			36		06B7	
39		37			0781			37		06C4	
40		38			0767			38		06D2	

Fuente: Autores

Figura57. Tabla de Función de Transferencia ADC

angulo	NUMERO	HEX	VOLTAGE	NUMERO	HEX	VOLTAGE	y(x)=FUNCION
0,0000	0	000	0	46	02E	0,02806	1,25000
2,8125	1	001	0,00061	47	02F	0,02867	1,31133
5,6250	2	002	0,00122	48	030	0,02928	1,37252
8,4375	3	003	0,00183	49	031	0,02989	1,43341
11,2500	4	004	0,00244	50	032	0,0305	1,49386
14,0625	5	005	0,00305	51	033	0,03111	1,55373
16,8750	6	006	0,00366	52	034	0,03172	1,61286
19,6875	7	007	0,00427	53	035	0,03233	1,67111
22,5000	8	008	0,00488	54	036	0,03294	1,72835
25,3125	9	009	0,00549	55	037	0,03355	1,78444
28,1250	10	00A	0,0061	56	038	0,03416	1,83925
30,9375	11	00B	0,00671	57	039	0,03477	1,89263
33,7500	12	00C	0,00732	58	03A	0,03538	1,94446
36,5625	13	00D	0,00793	59	03B	0,03599	1,99462
39,3750	14	00E	0,00854	60	03C	0,0366	2,04299
42,1875	15	00F	0,00915	61	03D	0,03721	2,08945
45,0000	16	010	0,00976	62	03E	0,03782	2,13388
47,8125	17	011	0,01037	63	03F	0,03843	2,17619
50,6250	18	012	0,01098	64	040	0,03904	2,21626
53,4375	19	013	0,01159	65	041	0,03965	2,25401
56,2500	20	014	0,0122	66	042	0,04026	2,28934
59,0625	21	015	0,01281	67	043	0,04087	2,32216
61,8750	22	016	0,01342	68	044	0,04148	2,35240
64,6875	23	017	0,01403	69	045	0,04209	2,37999
67,5000	24	018	0,01464	70	046	0,0427	2,40485
70,3125	25	019	0,01525	71	047	0,04331	2,42693
73,1250	26	01A	0,01586	72	048	0,04392	2,44618
75,9375	27	01B	0,01647	73	049	0,04453	2,46254
78,7500	28	01C	0,01708	74	04A	0,04514	2,47598
81,5625	29	01D	0,01769	75	04B	0,04575	2,48647
84,3750	30	01E	0,0183	76	04C	0,04636	2,49398
87,1875	31	01F	0,01891	77	04D	0,04697	2,49849
357,1880	32	020	0,01952	78	04E	0,04758	1,18868
354,3650	33	021	0,02013	79	04F	0,04819	1,12726
351,5630	34	022	0,02074	80	050	0,0488	1,06660
348,7500	35	023	0,02135	81	051	0,04941	1,00614
345,9380	36	024	0,02196	82	052	0,05002	0,94629
343,1250	37	025	0,02257	83	053	0,05063	0,88714
340,3130	38	026	0,02318	84	054	0,05124	0,82890
337,5000	39	027	0,02379	85	055	0,05185	0,77165
334,6880	40	028	0,0244	86	056	0,05246	0,71557
331,8750	41	029	0,02501	87	057	0,05307	0,66075
329,0630	42	02A	0,02562	88	058	0,05368	0,60738

tabla HEX	NUMERO	HEX	VOLTAGE	NUMERO	HEX	VOLTAGE
801	92	05C	0,05612	138	08A	0,08418
866	93	05D	0,05673	139	08B	0,08479
8CA	94	05E	0,05734	140	08C	0,0854
92E	95	05F	0,05795	141	08D	0,08601
990	96	060	0,05856	142	08E	0,08662
9F2	97	061	0,05917	143	08F	0,08723
A53	98	062	0,05978	144	090	0,08784
A24	99	063	0,06039	145	091	0,08845
B11	100	064	0,061	146	092	0,08906
B6D	101	065	0,06161	147	093	0,08967
BC7	102	066	0,06222	148	094	0,09028
C1E	103	067	0,06283	149	095	0,09089
C73	104	068	0,06344	150	096	0,0915
CC5	105	069	0,06405	151	097	0,09211
D14	106	06A	0,06466	152	098	0,09272
D61	107	06B	0,06527	153	099	0,09333
DA9	108	06C	0,06588	154	09A	0,09394
DF0	109	06D	0,06649	155	09B	0,09455
E32	110	06E	0,0671	156	09C	0,09516
E70	111	06F	0,06771	157	09D	0,09577
EA9	112	070	0,06832	158	09E	0,09638
EDF	113	071	0,06893	159	09F	0,09699
F10	114	072	0,06954	160	0A0	0,0976
F3D	115	073	0,07015	161	0A1	0,09821
F66	116	074	0,07076	162	0A2	0,09882
F8A	117	075	0,07137	163	0A3	0,09943
FAA	118	076	0,07198	164	0A4	0,10004
FCS	119	077	0,07259	165	0A5	0,10065
FDB	120	078	0,0732	166	0A6	0,10126
FEC	121	079	0,07381	167	0A7	0,10187
FF8	122	07A	0,07442	168	0A8	0,10248
FFF	123	07B	0,07503	169	0A9	0,10309
79C	124	07C	0,07564	170	0AA	0,1037
738	125	07D	0,07625	171	0AB	0,10431
6D4	126	07E	0,07686	172	0AC	0,10492
672	127	07F	0,07747	173	0AD	0,10553
60F	128	080	0,07808	174	0AE	0,10614
5AF	129	081	0,07869	175	0AF	0,10675
54F	130	082	0,0793	176	0B0	0,10736
4F1	131	083	0,07991	177	0B1	0,10797
495	132	084	0,08052	178	0B2	0,10858
43B	133	085	0,08113	179	0B3	0,10919
3E4	134	086	0,08174	180	0B4	0,1098

Fuente: Autores

Por ello, según la necesidad del usuario, el dispondrá de la calidad de la señal y de la cantidad de señales que desee trabajar según las características del modulo ya nombradas. Ver Anexo D. **Tabla de Excel – Función de Transferencia ADC.**

En la etapa de implementación y comprobación del módulo de PWM se constató lo siguiente:

Para modos de PWM con una resolución de salida de 8 bits presenta un error de precisión en la salida de PWM debido a que el tiempo de subida y el tiempo de caída del cristal llegan a ser críticos en la medida en que la frecuencia de salida de PWM aumenta. La generación de la señal de salida PWM posee una relación directamente proporcional al cristal del microcontrolador. De lo anterior se comprobó que a medida que la frecuencia de PWM al acercarse a sus valores limites, la precisión de la misma se degrada progresivamente.

Para modos PWM con una resolución de salida de 16 bits se encontró de manera empírica un punto de inflexión a partir del cual se elimina el error de precisión nombrado en el modo de 8 bits y se garantiza una precisión desde 1% hasta el 99% en el ciclo de trabajo con respecto a la salida.

A continuación se presenta unas tablas que muestran los rangos de frecuencias según el preescaler seleccionado en las frecuencias de PWM.  
Para mayor detalle de la tabla ver anexo **E. Tabla de Excel PWM COUNTER.**

Figura58. Tabla de Frecuencias PWM COUNTER con Preescaler de 1

A1	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
	PWM COUNTER	FRECUENCIA	PWM COUNTER	FRECUENCIA	PWM COUNTER	FRECUENCIA	PWM COUNTER	FRECUENCIA	PWM COUNTER	FRECUENCIA	PWM COUNTER	FRECUENCIA	PWM COUNTER	FRECUENCIA	PWM COUNTER
1	1	20.000.000,000000	2001	19.880.019800	2001	9.985.020499	3001	6.664.445185	4001	4.988.750112	5001	3.989.200560	6001	3.332.717870	
2	2	10.000.000,000000	3002	19.940.079840	2002	9.990.020990	3002	6.662.225183	4002	4.997.501249	5002	3.998.400560	6002	3.331.225920	
3	3	6.666.666,666667	3003	19.940.179462	2003	9.985.022466	3003	6.660.006660	4003	4.996.252810	5003	3.997.601459	6003	3.331.607500	
4	4	5.000.000,000000	3004	19.920.318725	2004	9.980.039920	3004	6.657.789654	4004	4.995.004995	5004	3.996.802558	6004	3.331.112592	
5	5	4.000.000,000000	3005	19.900.497512	2005	9.975.062344	3005	6.655.574043	4005	4.993.757803	5005	3.996.003996	6005	3.330.557868	
6	6	3.333.333,333333	3006	19.880.715706	2006	9.970.089711	3006	6.653.359947	4006	4.992.511233	5006	3.995.205753	6006	3.330.003300	
7	7	2.857.142,857143	3007	19.860.973188	2007	9.965.122073	3007	6.651.147323	4007	4.991.265286	5007	3.994.407829	6007	3.329.448976	
8	8	2.500.000,000000	3008	19.841.309641	2008	9.960.159363	3008	6.648.936170	4008	4.990.019960	5008	3.993.610224	6008	3.328.894807	
9	9	2.222.222,222222	3009	19.821.605550	2009	9.955.201593	3009	6.646.726487	4009	4.988.775256	5009	3.992.812937	6009	3.328.340822	
10	10	2.000.000,000000	3010	19.801.980198	2010	9.950.248756	3010	6.644.518272	4010	4.987.531172	5010	3.992.015968	6010	3.327.787022	
11	11	1.818.181,818182	3011	19.782.339670	2011	9.945.300845	3011	6.642.311524	4011	4.986.287709	5011	3.991.219318	6011	3.327.233405	
12	12	1.666.666,666667	3012	19.764.849850	2012	9.940.337853	3012	6.640.106242	4012	4.985.044865	5012	3.990.422965	6012	3.326.679973	
13	13	1.538.461,538462	3013	19.749.336624	2013	9.935.437771	3013	6.637.902423	4013	4.983.802641	5013	3.989.626970	6013	3.326.126735	
14	14	1.428.571,428571	3014	19.723.865878	2014	9.930.486594	3014	6.635.700666	4014	4.982.561036	5014	3.988.831272	6014	3.325.579661	
15	15	1.333.333,333333	3015	19.704.434968	2015	9.925.538313	3015	6.633.499171	4015	4.981.320050	5015	3.988.035892	6015	3.325.020781	
16	16	1.250.000,000000	3016	19.685.039370	2016	9.920.604921	3016	6.631.299735	4016	4.980.079681	5016	3.987.240829	6016	3.324.468085	
17	17	1.176.470,588235	3017	19.665.683382	2017	9.915.715411	3017	6.629.107377	4017	4.978.839950	5017	3.986.446083	6017	3.323.915573	
18	18	1.111.111,111111	3018	19.646.365422	2018	9.910.802775	3018	6.626.905235	4018	4.977.600796	5018	3.985.651654	6018	3.323.363244	
19	19	1.052.631,578947	3019	19.627.085378	2019	9.905.894007	3019	6.624.710169	4019	4.976.362279	5019	3.984.857541	6019	3.322.811098	
20	20	1.000.000,000000	3020	19.607.848137	2020	9.900.990999	3020	6.622.516556	4020	4.975.124378	5020	3.984.063745	6020	3.322.259136	
21	21	952.380,952381	3021	19.588.638950	2021	9.896.010444	3021	6.620.324396	4021	4.973.887093	5021	3.983.270285	6021	3.321.707398	
22	22	909.090,909091	3022	19.569.471624	2022	9.891.046855	3022	6.618.138866	4022	4.972.650423	5022	3.982.477101	6022	3.321.155762	
23	23	869.565,217391	3023	19.550.342131	2023	9.886.074664	3023	6.615.944426	4023	4.971.414387	5023	3.981.684252	6023	3.320.604300	
24	24	833.333,333333	3024	19.531.250000	2024	9.881.122925	3024	6.613.756614	4024	4.970.178926	5024	3.980.891720	6024	3.320.053121	
25	25	800.000,000000	3025	19.512.195122	2025	9.876.343210	3025	6.611.570248	4025	4.968.944099	5025	3.980.099502	6025	3.319.502075	
26	26	769.230,769231	3026	19.493.177388	2026	9.871.688312	3026	6.609.383527	4026	4.967.709886	5026	3.979.307600	6026	3.318.951211	
27	27	740.740,740741	3027	19.474.196689	2027	9.866.788234	3027	6.607.201800	4027	4.966.476285	5027	3.978.516014	6027	3.318.400581	
28	28	714.285,714286	3028	19.455.252918	2028	9.861.932939	3028	6.605.019815	4028	4.965.243297	5028	3.977.724741	6028	3.317.850033	
29	29	689.655,172414	3029	19.436.345967	2029	9.857.072449	3029	6.602.839221	4029	4.964.010921	5029	3.976.933784	6029	3.317.299718	
30	30	666.666,666667	3030	19.417.475728	2030	9.852.267749	3030	6.600.660066	4030	4.962.779156	5030	3.976.143141	6030	3.316.749485	
31	31	645.161,290323	3031	19.398.642095	2031	9.847.365880	3031	6.598.482499	4031	4.961.540033	5031	3.975.352833	6031	3.316.199655	
32	32	625.000,000000	3032	19.379.844961	2032	9.842.519685	3032	6.596.306669	4032	4.960.317460	5032	3.974.562798	6032	3.315.649867	
33	33	606.060,606061	3033	19.361.084221	2033	9.837.578308	3033	6.594.131223	4033	4.959.087528	5033	3.973.773098	6033	3.315.100282	
34	34	588.235,294118	3034	19.342.259788	2034	9.832.841691	3034	6.591.957811	4034	4.957.858205	5034	3.972.983711	6034	3.314.550878	
35	35	571.428,571429	3035	19.323.871496	2035	9.828.009828	3035	6.589.785882	4035	4.956.624042	5035	3.972.194658	6035	3.314.001657	
36	36	555.555,555556	3036	19.305.019305	2036	9.823.182711	3036	6.587.615283	4036	4.955.401388	5036	3.971.405878	6036	3.313.452618	
37	37	540.540,540541	3037	19.286.403086	2037	9.818.360344	3037	6.585.446164	4037	4.954.178892	5037	3.970.617451	6037	3.312.903760	
38	38	526.315,789474	3038	19.267.822736	2038	9.813.542489	3038	6.583.278473	4038	4.952.947003	5038	3.969.829297	6038	3.312.355084	
39	39	512.520,512522	3039	19.249.278152	2039	9.808.729769	3039	6.581.112206	4039	4.951.707213	5039	3.969.041476	6039	3.311.806590	
40	40	500.000,000000	3040	19.230.769331	2040	9.803.932169	3040	6.578.947368	4040	4.950.495500	5040	3.968.253968	6040	3.311.258278	
41	41	487.804,878049	3041	19.212.295869	2041	9.799.180739	3041	6.576.783953	4041	4.949.299983	5041	3.967.466772	6041	3.310.710147	
42	42	476.190,476190	3042	19.193.857965	2042	9.794.392925	3042	6.574.621959	4042	4.948.045522	5042	3.966.678989	6042	3.310.162198	
43	43	465.116,279070	3043	19.175.455417	2043	9.789.525208	3043	6.572.461387	4043	4.946.816667	5043	3.965.893317	6043	3.309.614430	
44	44	454.545,454545	3044	19.157.088123	2044	9.784.735812	3044	6.570.302234	4044	4.945.598417	5044	3.965.107058	6044	3.309.066843	
45	45	444.444,444444	3045	19.138.755981	2045	9.779.911000	3045	6.568.144499	4045	4.944.375713	5045	3.964.321110	6045	3.308.519438	
46	46	434.782,608696	3046	19.120.458891	2046	9.775.171065	3046	6.565.988811	4046	4.943.153732	5046	3.963.535474	6046	3.307.972213	
47	47	425.531,914894	3047	19.102.190753	2047	9.770.395701	3047	6.563.833279	4047	4.941.932296	5047	3.962.750149	6047	3.307.425170	

Fuente: Autores

Figura59. Tabla de Frecuencias PWM COUNTER con Preescalador de 4.

A1	f <sub>s</sub> = 1/(500000)											
A	B	C	D	E	F	G	H	I	J	K	L	M
200.0E-9	PWM COUNTER	FRECUENCIA	PWM COUNTER	FRECUENCIA	PWM COUNTER	FRECUENCIA	PWM COUNTER	FRECUENCIA	PWM COUNTER	FRECUENCIA	PWM COUNTER	FRECUENCIA
1	1	5.000.000.00000	1001	4.995.004995	2001	2.498.750625	3001	1.666.111296	4001	1.249.687578	5001	999.800040
2	2	2.500.000.00000	1002	4.990.019960	2002	2.497.502498	3002	1.665.556296	4002	1.249.375512	5002	999.400160
3	3	1.666.666.66667	1003	4.985.044865	2003	2.496.255617	3003	1.665.001665	4003	1.249.063203	5003	999.000360
4	4	1.250.000.00000	1004	4.980.079681	2004	2.495.009980	3004	1.664.447403	4004	1.248.751249	5004	998.200639
5	5	1.000.000.00000	1005	4.975.124378	2005	2.493.765586	3005	1.663.893511	4005	1.248.439451	5005	999.000999
6	6	833.333.33333	1006	4.970.178926	2006	2.492.522453	3006	1.663.339987	4006	1.248.127808	5006	998.801438
7	7	714.285.714286	1007	4.965.245297	2007	2.491.280518	3007	1.662.786851	4007	1.247.816321	5007	998.601957
8	8	625.000.00000	1008	4.960.317460	2008	2.490.039841	3008	1.662.234043	4008	1.247.504980	5008	998.402556
9	9	555.555.55556	1009	4.955.401388	2009	2.488.800398	3009	1.661.681622	4009	1.247.193814	5009	998.203234
10	10	500.000.00000	1010	4.950.495050	2010	2.487.567189	3010	1.661.129568	4010	1.246.882793	5010	998.003992
11	11	454.545.45455	1011	4.945.598417	2011	2.486.325211	3011	1.660.577881	4011	1.246.571927	5011	997.804829
12	12	416.666.66667	1012	4.940.711462	2012	2.485.089463	3012	1.660.026560	4012	1.246.261216	5012	997.605746
13	13	384.615.384615	1013	4.935.834156	2013	2.483.854943	3013	1.659.475606	4013	1.245.950660	5013	997.406742
14	14	357.142.857143	1014	4.930.964469	2014	2.482.621648	3014	1.658.925017	4014	1.245.640259	5014	997.207818
15	15	333.333.33333	1015	4.926.108374	2015	2.481.389578	3015	1.658.374793	4015	1.245.330012	5015	997.008973
16	16	312.500.00000	1016	4.921.259843	2016	2.480.158730	3016	1.657.824934	4016	1.245.019920	5016	996.810207
17	17	294.117.647059	1017	4.916.420846	2017	2.478.929103	3017	1.657.274539	4017	1.244.709983	5017	996.611521
18	18	277.777.77778	1018	4.911.591356	2018	2.477.700694	3018	1.656.726309	4018	1.244.400199	5018	996.412914
19	19	263.157.894737	1019	4.906.771344	2019	2.476.479302	3019	1.656.177542	4019	1.244.090570	5019	996.214385
20	20	250.000.00000	1020	4.901.960784	2020	2.475.247525	3020	1.655.629138	4020	1.243.781095	5020	996.015936
21	21	238.095.238095	1021	4.897.159647	2021	2.474.022761	3021	1.655.081099	4021	1.243.471773	5021	995.817566
22	22	227.272.727273	1022	4.892.367906	2022	2.472.799209	3022	1.654.533422	4022	1.243.162606	5022	995.619275
23	23	217.391.304348	1023	4.887.585533	2023	2.471.576866	3023	1.653.986107	4023	1.242.855592	5023	995.421063
24	24	208.333.33333	1024	4.882.812500	2024	2.470.355791	3024	1.653.439153	4024	1.242.544732	5024	995.222930
25	25	200.000.00000	1025	4.878.049780	2025	2.469.135862	3025	1.652.892562	4025	1.242.236025	5025	995.024876
26	26	192.307.692308	1026	4.873.294847	2026	2.467.917078	3026	1.652.346332	4026	1.241.927471	5026	994.826900
27	27	185.185.185185	1027	4.868.549172	2027	2.466.699556	3027	1.651.800463	4027	1.241.619071	5027	994.629003
28	28	178.571.428571	1028	4.863.813230	2028	2.465.483235	3028	1.651.254954	4028	1.241.310824	5028	994.431185
29	29	172.413.791105	1029	4.859.086492	2029	2.464.268112	3029	1.650.709805	4029	1.241.002730	5029	994.233446
30	30	166.666.66667	1030	4.854.368932	2030	2.463.054187	3030	1.650.165017	4030	1.240.694789	5030	994.035785
31	31	161.290.321581	1031	4.849.660524	2031	2.461.841457	3031	1.649.620587	4031	1.240.387001	5031	993.838203
32	32	156.250.00000	1032	4.844.961240	2032	2.460.629921	3032	1.649.076517	4032	1.240.079565	5032	993.640700
33	33	151.515.151515	1033	4.840.271055	2033	2.459.419577	3033	1.648.532806	4033	1.239.771882	5033	993.443274
34	34	147.058.823529	1034	4.835.589942	2034	2.458.210423	3034	1.647.989453	4034	1.239.464551	5034	993.245928
35	35	142.857.142857	1035	4.830.917874	2035	2.457.002457	3035	1.647.446458	4035	1.239.157373	5035	993.048659
36	36	138.888.888889	1036	4.826.254826	2036	2.455.795678	3036	1.646.903821	4036	1.238.850347	5036	992.851469
37	37	135.135.135135	1037	4.821.600771	2037	2.454.590083	3037	1.646.361841	4037	1.238.543478	5037	992.654358
38	38	131.578.947368	1038	4.816.955684	2038	2.453.385672	3038	1.645.820618	4038	1.238.236751	5038	992.457324
39	39	128.205.128205	1039	4.812.319538	2039	2.452.182442	3039	1.645.279052	4039	1.237.930181	5039	992.260369
40	40	125.000.00000	1040	4.807.692308	2040	2.450.980393	3040	1.644.738842	4040	1.237.623762	5040	992.063492
41	41	121.951.219512	1041	4.803.073967	2041	2.449.779520	3041	1.644.199588	4041	1.237.317496	5041	991.866693
42	42	119.047.619048	1042	4.798.464491	2042	2.448.578924	3042	1.643.655490	4042	1.237.011381	5042	991.669972

Fuente: Autores

Figura60. Tabla de Frecuencias PWM COUNTER con Preescalador de 16.

A1	f <sub>s</sub> = 1/(1250000)										
A	B	C	D	E	F	G	H	I	J	K	
800.0E-9	PWM COUNTER	FRECUENCIA	PWM COUNTER	FRECUENCIA	PWM COUNTER	FRECUENCIA	PWM COUNTER	FRECUENCIA	PWM COUNTER	FRECUENCIA	
1	1	1.250.000.00000	1001	1.248.751249	2001	624.687656	3001	416.527824	4001	312.421895	
2	2	625.000.00000	1002	1.247.504990	2002	624.375624	3002	416.389074	4002	312.343828	
3	3	416.666.66667	1003	1.246.261216	2003	624.063904	3003	416.250416	4003	312.265801	
4	4	312.500.00000	1004	1.245.019220	2004	623.752495	3004	416.111851	4004	312.187812	
5	5	250.000.00000	1005	1.243.781095	2005	623.441397	3005	415.973378	4005	312.109863	
6	6	208.333.33333	1006	1.242.544732	2006	623.130608	3006	415.834997	4006	312.031952	
7	7	178.571.428571	1007	1.241.310824	2007	622.820130	3007	415.696708	4007	311.954080	
8	8	156.250.00000	1008	1.240.079365	2008	622.509960	3008	415.558511	4008	311.876248	
9	9	138.888.88889	1009	1.238.803477	2009	622.200100	3009	415.420405	4009	311.798453	
10	10	125.000.00000	1010	1.237.623762	2010	621.890547	3010	415.282392	4010	311.720698	
11	11	113.636.363636	1011	1.236.399604	2011	621.581303	3011	415.144470	4011	311.642982	
12	12	104.166.66667	1012	1.235.177866	2012	621.272366	3012	415.006640	4012	311.565304	
13	13	96.153.846154	1013	1.233.958539	2013	620.963736	3013	414.868901	4013	311.487665	
14	14	89.285.714286	1014	1.232.741617	2014	620.655412	3014	414.731254	4014	311.410065	
15	15	83.333.33333	1015	1.231.527094	2015	620.347395	3015	414.593698	4015	311.332503	
16	16	78.125.00000	1016	1.230.314961	2016	620.039683	3016	414.456233	4016	311.254980	
17	17	73.529.411765	1017	1.229.105211	2017	619.732276	3017	414.318860	4017	311.177496	
18	18	69.444.444444	1018	1.227.897839	2018	619.425173	3018	414.181577	4018	311.100050	
19	19	65.789.473684	1019	1.226.692836	2019	619.118375	3019	414.044386	4019	311.022642	
20	20	62.500.00000	1020	1.225.490196	2020	618.811881	3020	413.907285	4020	310.945274	
21	21	59.523.809524	1021	1.224.289912	2021	618.505690	3021	413.770275	4021	310.867943	
22	22	56.818.181818	1022	1.223.091977	2022	618.199802	3022	413.633355	4022	310.790651	
23	23	54.347.826083	1023	1.221.896383	2023	617.894217	3023	413.496527	4023	310.713398	
24	24	52.083.33333	1024	1.220.710325	2024	617.588933	3024	413.359788	4024	310.636183	
25	25	50.000.00000	1025	1.219.512195	2025	617.283951	3025	413.223140	4025	310.559006	
26	26	48.076.923077	1026	1.218.323587	2026	616.979269	3026	413.086583	4026	310.481868	
27	27	46.296.296296	1027	1.217.137293	2027	616.674889	3027	412.950116	4027	310.404768	
28	28	44.642.857143	1028	1.215.953307	2028	616.370809	3028	412.813738	4028	310.327706	
29	29	43.103.448276	1029	1.214.771623	2029	616.067028	3029	412.677451	4029	310.250683	
30	30	41.666.66667	1030	1.213.592233	2030	615.763547	3030	412.541254	4030	310.173697	
31	31	40.322.580645	1031	1.212.415131	2031	615.460364	3031	412.405147	4031	310.096750	
32	32	39.062.500000	1032	1.211.240310	2032	615.157480	3032	412.269129	4032	310.019841	
33	33	37.878.787879	1033	1.210.067764	2033	614.854894	3033	412.133201	4033	309.942970	
34	34	36.764.705882	1034	1.208.897485	2034	614.552606	3034	411.997363	4034	309.866138	
35	35	35.714.285714	1035	1.207.729469	2035	614.250614	3035	411.861614	4035	309.78934	

Figura61. Tabla de Frecuencias PWM COUNTER con Preescalador de 64.

A	B	C	D	E	F	G	H	I	J	K	L	M
3.2E-6	PWM COUNTER	FRECUENCIA	PWM COUNTER	FRECUENCIA	PWM COUNTER	FRECUENCIA	PWM COUNTER	FRECUENCIA	PWM COUNTER	FRECUENCIA	PWM COUNTER	FRECUENCIA
2	1	312.500,000000	1001	312.187812	2001	156.171914	3001	104.131956	4001	78.105474	5001	62.487502
3	2	156.250,000000	1002	311.876248	2002	156.093906	3002	104.097268	4002	78.089557	5002	62.475010
4	3	104.166,666667	1003	311.565304	2003	156.015976	3003	104.062604	4003	78.064650	5003	62.462522
5	4	78.125,000000	1004	311.254980	2004	155.938124	3004	104.027963	4004	78.049953	5004	62.450040
6	5	62.500,000000	1005	310.945274	2005	155.860349	3005	103.993344	4005	78.027466	5005	62.437562
7	6	52.083,333333	1006	310.636188	2006	155.782652	3006	103.958749	4006	78.007988	5006	62.425090
8	7	44.642,857143	1007	310.327706	2007	155.705052	3007	103.924177	4007	77.988520	5007	62.412622
9	8	39.062,500000	1008	310.019841	2008	155.627490	3008	103.889628	4008	77.969062	5008	62.400160
10	9	34.722,222222	1009	309.712587	2009	155.550025	3009	103.855101	4009	77.949613	5009	62.387702
11	10	31.250,000000	1010	309.405941	2010	155.472637	3010	103.820598	4010	77.930175	5010	62.375250
12	11	28.409,090909	1011	309.099901	2011	155.395326	3011	103.786118	4011	77.910745	5011	62.362802
13	12	26.041,666667	1012	308.794466	2012	155.318091	3012	103.751660	4012	77.891326	5012	62.350359
14	13	24.038,461538	1013	308.489635	2013	155.240994	3013	103.717225	4013	77.871916	5013	62.337921
15	14	22.321,428571	1014	308.185404	2014	155.163853	3014	103.682814	4014	77.852516	5014	62.325489
16	15	20.833,333333	1015	307.881773	2015	155.086849	3015	103.648425	4015	77.833126	5015	62.313061
17	16	19.531,250000	1016	307.578740	2016	155.009921	3016	103.614058	4016	77.813745	5016	62.300638
18	17	18.382,352941	1017	307.276303	2017	154.933069	3017	103.579715	4017	77.794374	5017	62.288210
19	18	17.361,111111	1018	306.974460	2018	154.856293	3018	103.545394	4018	77.775012	5018	62.275787
20	19	16.447,368421	1019	306.673209	2019	154.779594	3019	103.511096	4019	77.755661	5019	62.263399
21	20	15.625,000000	1020	306.372549	2020	154.702970	3020	103.476821	4020	77.736318	5020	62.250996
22	21	14.880,952381	1021	306.072478	2021	154.626423	3021	103.442569	4021	77.716986	5021	62.238598
23	22	14.204,545455	1022	305.772994	2022	154.549951	3022	103.408339	4022	77.697663	5022	62.226205
24	23	13.586,956522	1023	305.474096	2023	154.473554	3023	103.374132	4023	77.678349	5023	62.213816
25	24	13.020,833333	1024	305.175781	2024	154.397233	3024	103.339947	4024	77.659046	5024	62.201433
26	25	12.500,000000	1025	304.878049	2025	154.320988	3025	103.305785	4025	77.639752	5025	62.189055
27	26	12.019,230769	1026	304.580897	2026	154.244817	3026	103.271646	4026	77.620467	5026	62.176681
28	27	11.574,074074	1027	304.284323	2027	154.168725	3027	103.237529	4027	77.601192	5027	62.164313
29	28	11.160,714286	1028	303.988327	2028	154.092702	3028	103.203435	4028	77.581927	5028	62.151949
30	29	10.775,862069	1029	303.692906	2029	154.016757	3029	103.169363	4029	77.562671	5029	62.139590
31	30	10.416,666667	1030	303.398058	2030	153.940887	3030	103.135314	4030	77.543424	5030	62.127237
32	31	10.080,645161	1031	303.103783	2031	153.865091	3031	103.101287	4031	77.524188	5031	62.114888
33	32	9.765,625000	1032	302.810078	2032	153.789370	3032	103.067282	4032	77.504960	5032	62.102544
34	33	9.469,699570	1033	302.516941	2033	153.713724	3033	103.033300	4033	77.485743	5033	62.090205
35	34	9.191,76471	1034	302.224371	2034	153.638151	3034	102.999341	4034	77.466534	5034	62.077870
36	35	8.928,571429	1035	301.932367	2035	153.562654	3035	102.965404	4035	77.447336	5035	62.065541
37	36	8.680,555556	1036	301.640927	2036	153.487230	3036	102.931489	4036	77.428147	5036	62.053217
38	37	8.445,945946	1037	301.350448	2037	153.411880	3037	102.897596	4037	77.408967	5037	62.040897
39	38	8.223,684211	1038	301.059730	2038	153.336605	3038	102.863726	4038	77.389797	5038	62.028583
40	39	8.012,820513	1039	300.769971	2039	153.261403	3039	102.829878	4039	77.370636	5039	62.016273
41	40	7.812,500000	1040	300.480769	2040	153.186275	3040	102.796055	4040	77.351485	5040	62.003968
42	41	7.621,951220	1041	300.192123	2041	153.111220	3041	102.762249	4041	77.332343	5041	61.991668
43	42	7.440,476190	1042	299.904031	2042	153.036239	3042	102.728468	4042	77.313211	5042	61.979373
44	43	7.267,441860	1043	299.616491	2043	152.961331	3043	102.694709	4043	77.294089	5043	61.967083

Fuente: Autores

En el módulo de contador de eventos de 16 bits el microcontrolador entra a un bucle infinito en el cual espera a que suceda un evento para llevar un conteo y visualizarlo en los puertos bidireccionales. Para interrumpir este bucle infinito se activó la interrupción serial y debido a que el módulo del contador también funciona por medio de interrupciones, se estableció la prioridad alta a la interrupción serial, lo que significa que dicha interrupción (serial) interrumpe a cualquier interrupción que se esté ejecutando en ese momento, ejecuta la interrupción serial y devuelve el control a la interrupción o programa principal donde haya sido llamado.

Dentro de la interrupción serial se compara el comando recibido con el comando preestablecido, en caso de ser iguales, interrumpe el bucle infinito, de lo contrario, permanece en el bucle.

Cuando el contador alcanza su valor máximo se detiene y espera a recibir el comando de parada evitando que se reinicie el conteo automáticamente y se pierda el conteo realizado.

En el módulo del DAC las muestras que conforman las ondas se encuentran previamente almacenadas en una tabla. Dichas muestras son enviadas al DAC

por medio de la interrupción del Timer 0. El Timer 0 establece la frecuencia de las ondas y actúa por medio de la interrupción del Timer 0.

Con el objetivo de visualizar las ondas en forma periódica, el Timer es mantenido en un bucle infinito. Al igual que en el módulo del contador, la interrupción serial es quien libera al microcontrolador del bucle y lo retorna al programa principal.

La frecuencia de las ondas de salida se encuentran relacionadas con el número de muestras que conforman la onda. Debido a que el Timer 0 establece la frecuencia de salida y dicho Timer tiene una resolución de 8 bits, se estableció una frecuencia de salida para cada valor de desbordamiento del Timer 0.

Para efectos de frecuencias más altas se reduciría el número de muestras que conforman la onda.

En los módulos de contador y DAC se empleó la interrupción de la UART con prioridad alta con el propósito de retornar al programa principal los módulos que permanecen en un bucle infinito.

En la parte análoga (ADC, DAC) el voltaje de referencia se estableció de manera interna a 2.5V por motivos de confiabilidad y facilidad en la implementación del diseño.

En la implementación del módulo de comunicación se utilizó el Firmware que proporciona el Kit de desarrollo del TUSB3410UARTPDK que posee su propio VID y PID que es propiedad de Texas Instrument.

### **3.2. Capa Hardware**

El entrenador digital consta de dos circuitos impresos mencionados a continuación:

En el circuito impreso del microcontrolador ADuC841 se encuentra la fuente de alimentación, la circuitería análoga-digital, las borneras para la circuitería externa y los conectores que comunican al microcontrolador con el módulo de comunicación.

Este modulo se realizo de manera separada debido a inconvenientes de conectividad presentados en la primera tarjeta realizada. Esta decisión de diseño fue tomada con el fin de ir comprobando etapas de funcionamiento del sistema.

En el circuito impreso del microcontrolador TUSB3410 se encuentra los componentes que permiten establecer la comunicación entre el protocolo USB-RS232 UART. Este módulo se realizó de manera separada para comprobar el correcto funcionamiento.

El entrenador digital se encuentra energizado por medio de un regulador de 5V octoacoplado. La razón por la cual se empleo este tipo de fuente octoacoplada se debe en principio, al deseo de proveer al entrenador digital de puertos I/O octoacoplados. En el proceso de desarrollo se presentó inconsistencia en el diseño del acoplamiento debido a problemas de tierra comunes. Dado que los componentes empleados son de carácter superficial y son importados, se optó por suprimir esta etapa ya que era un valor agregado y exigía mayor tiempo y costo en el proceso de desarrollo.

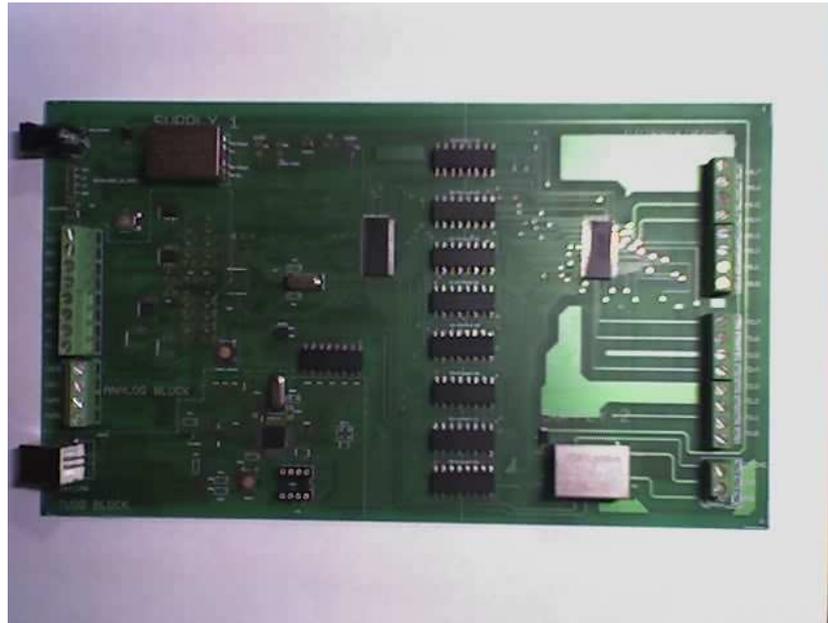
El entrenador digital USB posee una gran ventaja respecto a las demás herramientas de su género, en el sentido que permite control del entrenador digital y todos sus módulos así como también permite modificar el código fuente del microcontrolador a través del mismo protocolo USB, es decir permite programar, depurar y corregir código fuente en tiempo real, evitando implementación de un circuito quemador y la necesidad de remover el microcontrolador de la placa.

El entrenador digital USB funciona con un computador, es decir no se puede conectar dos o más entrenadores al mismo computador debido a que firmware contenido en la EEPROM no se encuentra serializada.

Serializar el firmware significa que el sistema operativo asocia un adaptador serial de puertos múltiples a cada entrenador. Véase **anexo A. Manual de Usuario para Instalar el Entrenador Digital USB.**

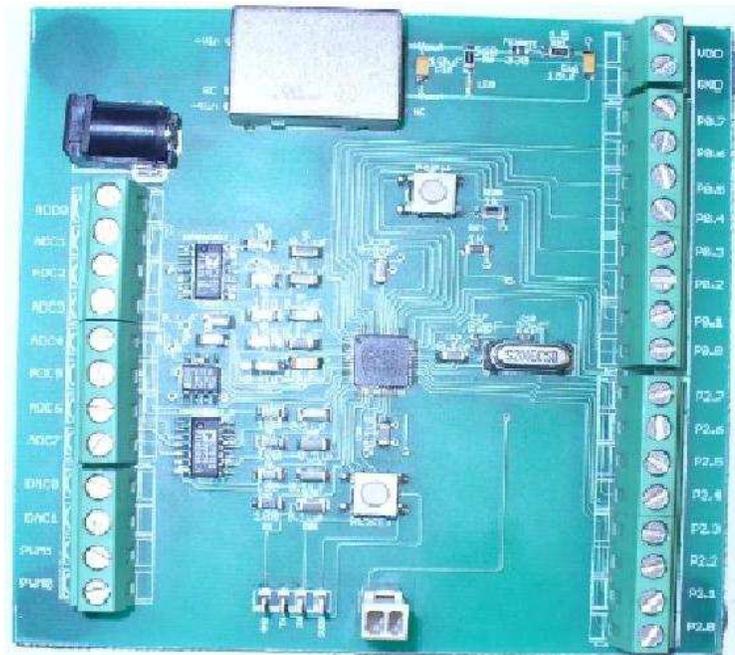
### 3.3 Etapas en el Desarrollo del Entrenador Digital USB

Figura62. Primer Prototipo del Entrenador Digital USB



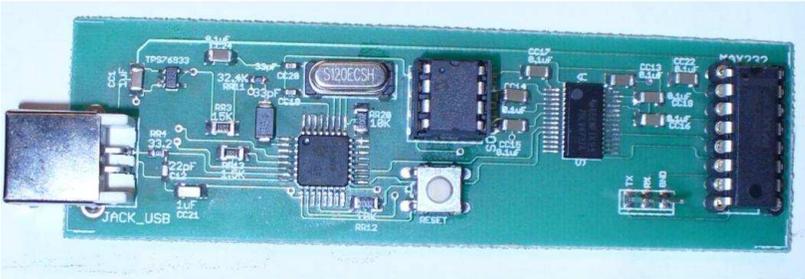
Fuente: Autores

Figura63. Módulo del Microcontrolador ADuC841



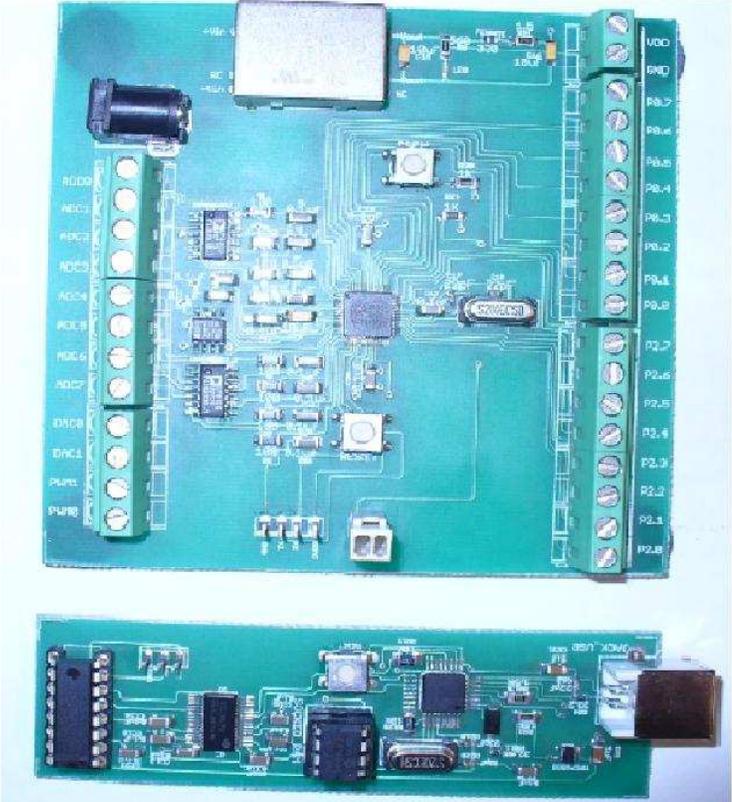
Fuente: Autores

Figura64. Módulo de Comunicación del Microcontrolador TUSB3410 USB -UART



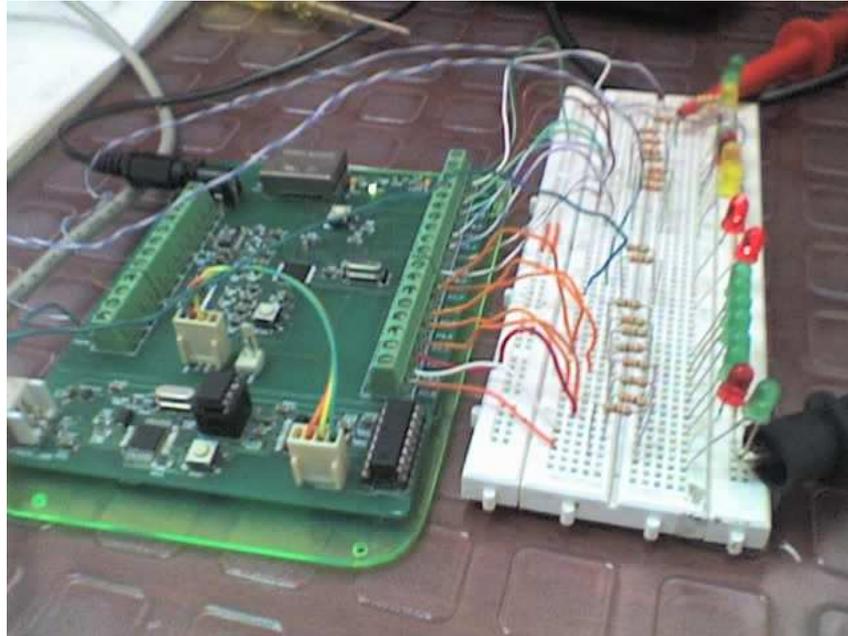
Fuente: Autores

Figura65. Módulos del Entrenador Digital USB



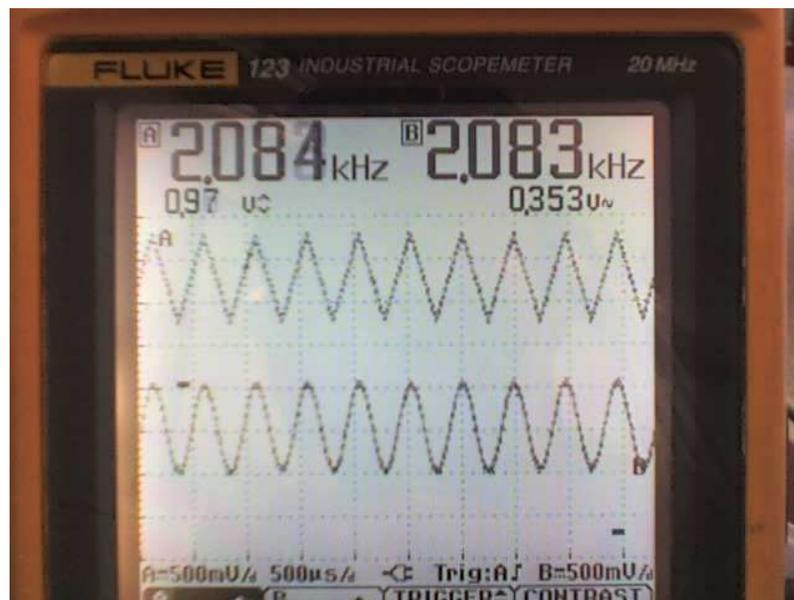
Fuente: Autores

Figura66. Pruebas del Entrenador



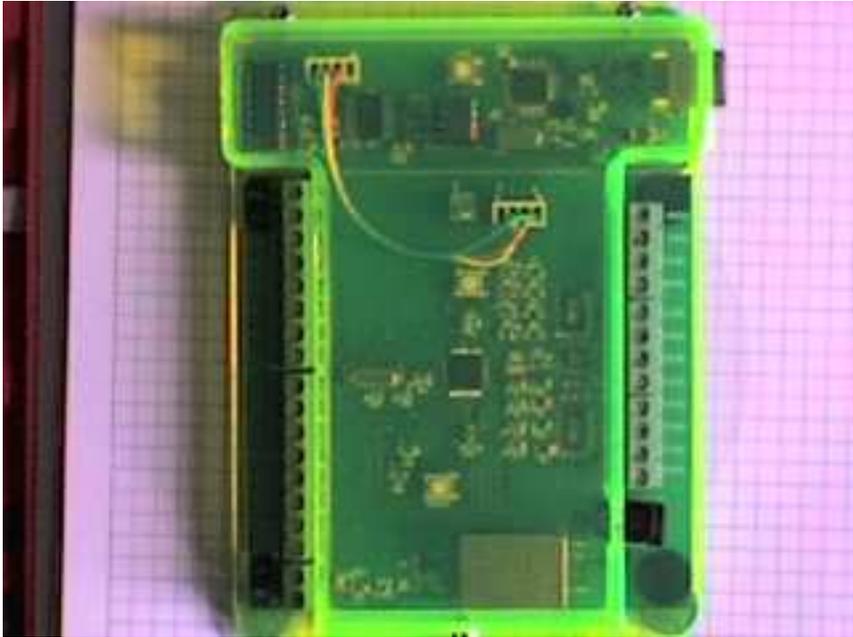
Fuente: Autores

Figura67. Señales de Prueba del DAC del Entrenador Digital USB.



Fuente: Autores

Figura68. Entrenador Digital USB Final



Fuente: Autores

#### 4. CONCLUSIONES Y PERSPECTIVAS

- Se elaboro un entrenador digital con conectividad USB e interfaz grafica en Labview que permiten controlar los diversos módulos el entrenador.
- Se desarrollo la capa de hardware y software necesaria en la operación del entrenador y actualización del entrenador. Se provee al entrenador con la capacidad de mejorar, ajustar y agregar nuevos módulos según las necesidades y aplicaciones del usuario.
- El entrenador Digital USB posee la versatilidad de emplear el hardware con distintos ambientes de desarrollo de programación que permitan interactuar con periféricos que utilizan el protocolo serial.
- Se diseño la interfaz grafica en Labview de forma intuitiva y amigable al usuario.
- Para futuras mejoras del entrenador, en el caso del modulo del ADC, se sugiere una implementación de una memoria externa de hasta 16 Mbytes, permitiendo así, una mayor cantidad de muestras durante la conversión.
- Para la construcción de futuros entrenadores, ya habiendo sido probadas las tarjetas de los módulos del microcontrolador ADuC841 y de comunicación con el microcontrolador TUSB3410, se puede diseñar la tarjeta con ambos módulos en conjunto para efectos de reducción de costos y compactibilidad.
- Para efectos de mejorar la resolución del ADC se puede implementar el microcontrolador ADuC845 que posee una resolución de 24 bits y es compatible en gran medida con las características del ADuC841.

## Bibliografía

COWAN, Frederick. 8051 Microcontrollers: An Applications-based Introduction. Ediciones Elseiver. Publicado 2004. 416 p. ISBN 0750657596.

AYALA, Kenneth. The 8051 Microcontroller. Ediciones Thomson Delmar Learning. Publicado 2004. 432 p. ISBN 140186158X.

AXELSON, Jan. Serial Port Complete: COM PORTS, USB VIRTUAL COM PORTS AND PORTS FOR EMBEDDED SYSTEM. Ediciones Lakeview Research LLC. Publicado 2007. 400 p. ISBN 978-1931448-07-9

NATIONAL INSTRUMENT. Labview user Manual. [En línea].  
< [www.ni.com/pdf/manuals/320999e.pdf](http://www.ni.com/pdf/manuals/320999e.pdf) > [citado 3 de julio del 2007].

METALINK CORPORATION. 8051 CROSS ASSEMBLER USER'S MANUAL MetaLink Corporation. [En línea]. < [www.k6ese.com/data/asm51.pdf](http://www.k6ese.com/data/asm51.pdf) > [citado 9 de julio del 2007].

ANALOG DEVICES. ADuC841 MicroConverter 12-Bit ADCs and DACs. [En línea].  
< [www.analog.com/UploadedFiles/Data\\_Sheets/ADUC841\\_842\\_843.pdf](http://www.analog.com/UploadedFiles/Data_Sheets/ADUC841_842_843.pdf) > [citado 10 de octubre del 2006].

TEXAS INSTRUMENT. TUSB3410 USB-to-Serial Port Controller (Rev. F). [En línea]. < [www.kamami.pl/dl/tusb3410.pdf](http://www.kamami.pl/dl/tusb3410.pdf) > [citado 15 de marzo del 2007].

TEXAS INSTRUMENT. USB/Serial Applications Using TUSB3410/5052 and the VCP Software. [En línea]. < [focus.ti.com/lit/an/slla170d/slla170d.pdf](http://focus.ti.com/lit/an/slla170d/slla170d.pdf) > [citado 15 de marzo del 2007].

TDK LAMBDA. DC to DC Converters Insulated Type, Ultra Compact Size, 1.5 to 25W. [En línea]. < [www.tdk.co.jp/tefe02/ea335\\_cc\\_e.pdf](http://www.tdk.co.jp/tefe02/ea335_cc_e.pdf) > [citado 11 de mayo del 2007].

ANALOG DEVICES. OP191/OP291/OP491 Micropower Single-Supply Rail-to-Rail Input. [En línea].  
< [www.analog.com/UploadedFiles/Data\\_Sheets/OP191\\_291\\_491.pdf](http://www.analog.com/UploadedFiles/Data_Sheets/OP191_291_491.pdf) > [citado 13 de mayo del 2007].

CADSOFT EAGLE. Manual EAGLE 4.1. [En línea].  
< [web.mit.edu/eaglecad\\_v4.16/manual-eng.pdf](http://web.mit.edu/eaglecad_v4.16/manual-eng.pdf) > [citado 2 de abril del 2007].

## ANEXOS

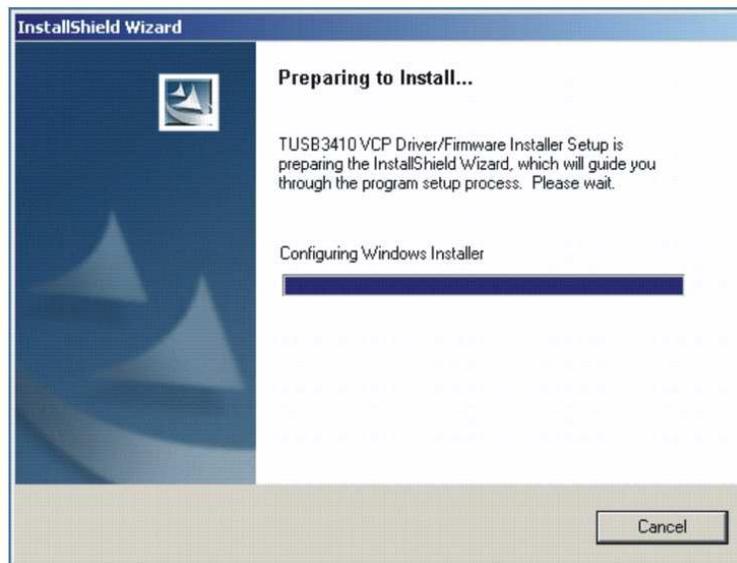
### A. Manual de Usuario para Instalar el Entrenador Digital USB.

Para la instalación de la aplicación y el manejo del entrenador el usuario debe realizar los siguientes pasos antes de trabajar con el Entrenador Digital USB.

- *Paso 1:*

Se ejecuta "setup.exe" del instalador del driver del TUSB3410, el cual instala los archivos necesarios para la utilización del dispositivo.

**Figura. TUSB3410 VCP Driver/Firmware Instalación Progreso**

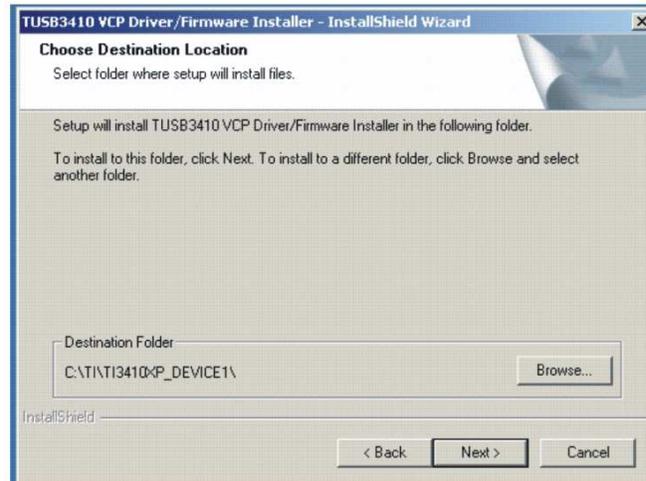


Fuente: Autores

- *Paso 2:*

Se selecciona la localización del directorio. Por defecto Windows selecciona C:\TI\TI3410XP\_DEVICE.

**Figura. TUSB3410 VCP Driver/Firmware Instalación del Directorio**

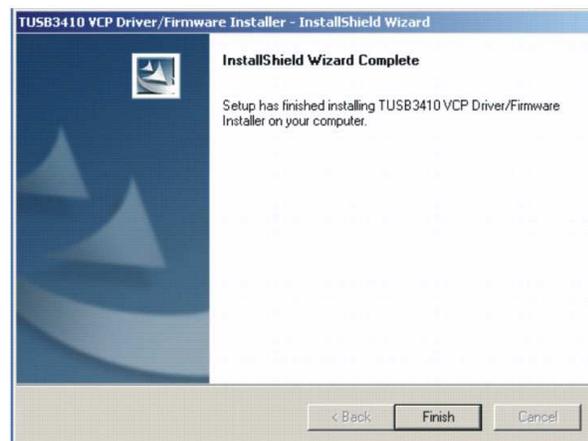


Fuente: Autores

- *Paso 3:*

Finalización de la instalación del Drivers/Firmware VCP del TUSB3410.

**Figura. TUSB3410 VCP Driver/Firmware Instalación Completada**

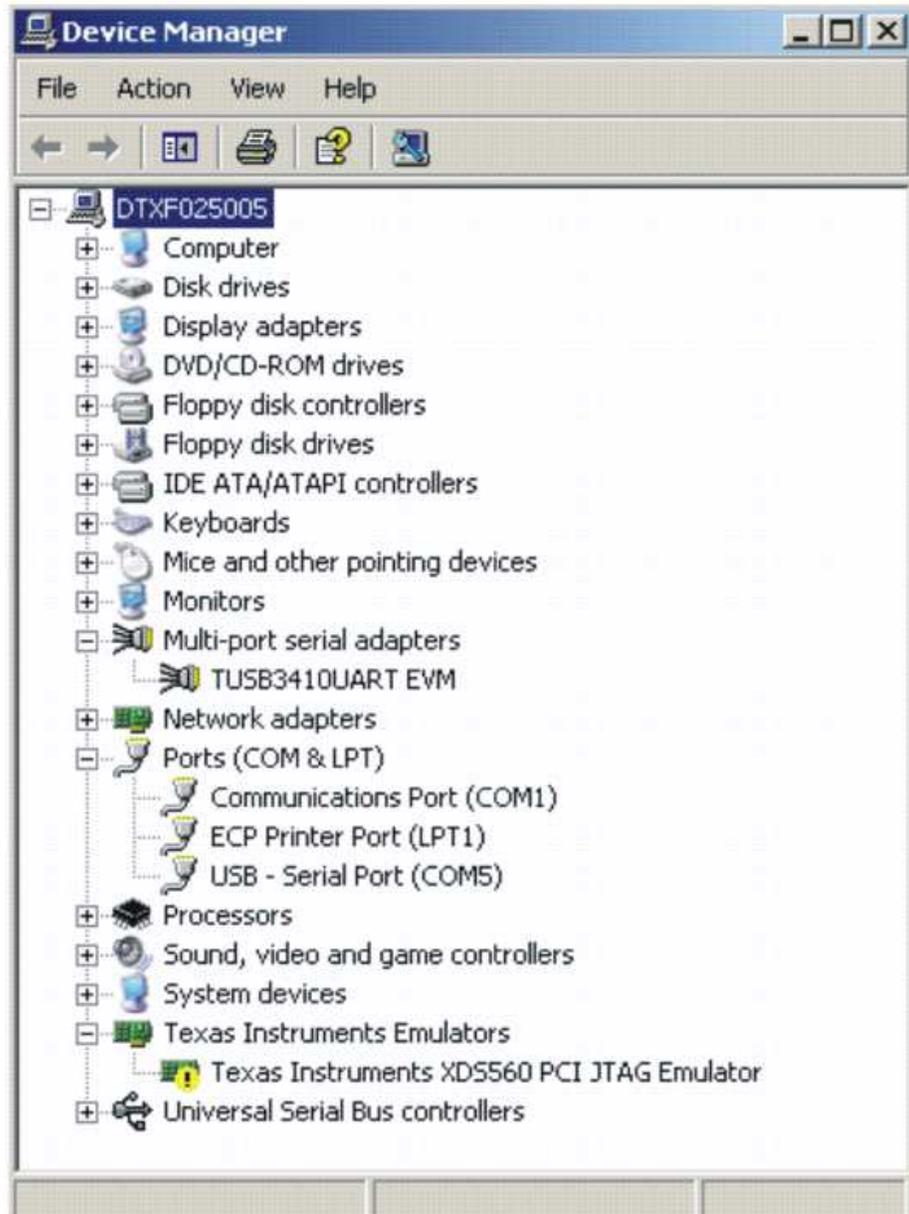


- Fuente: Autores

- *Paso 4:*

Conecte el entrenador digital al PC. Windows instalará los drivers automáticamente como se muestra en la siguiente figura.

**Figura. Visualización del administrador de Dispositivos**

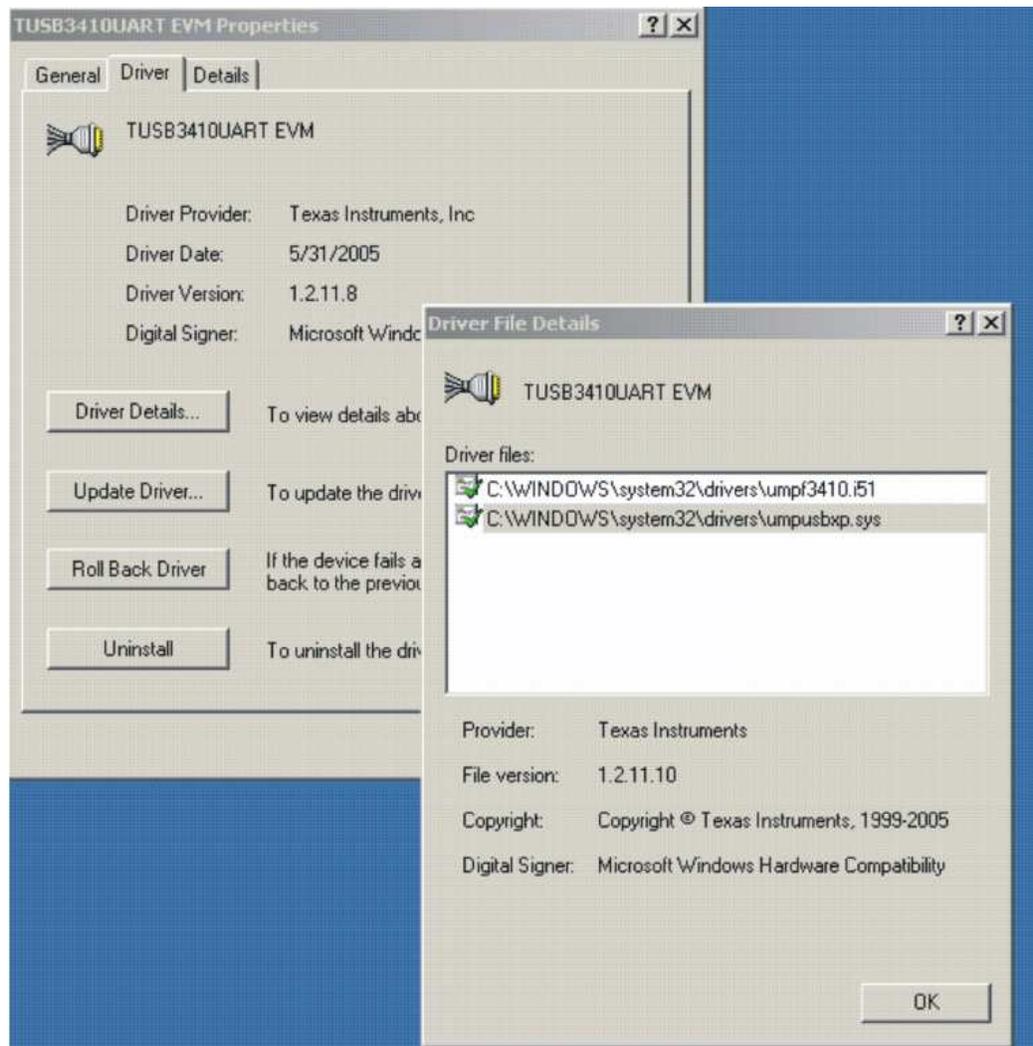


Fuente: Autores

- *Paso 5:*

Para observar las propiedades del driver, se da doble click al dispositivo TUSB3410UART EVM, selecciona driver y luego detalles del driver. Como se ilustra en la figura siguiente.

**Figura. Propiedades del driver del Dispositivo**

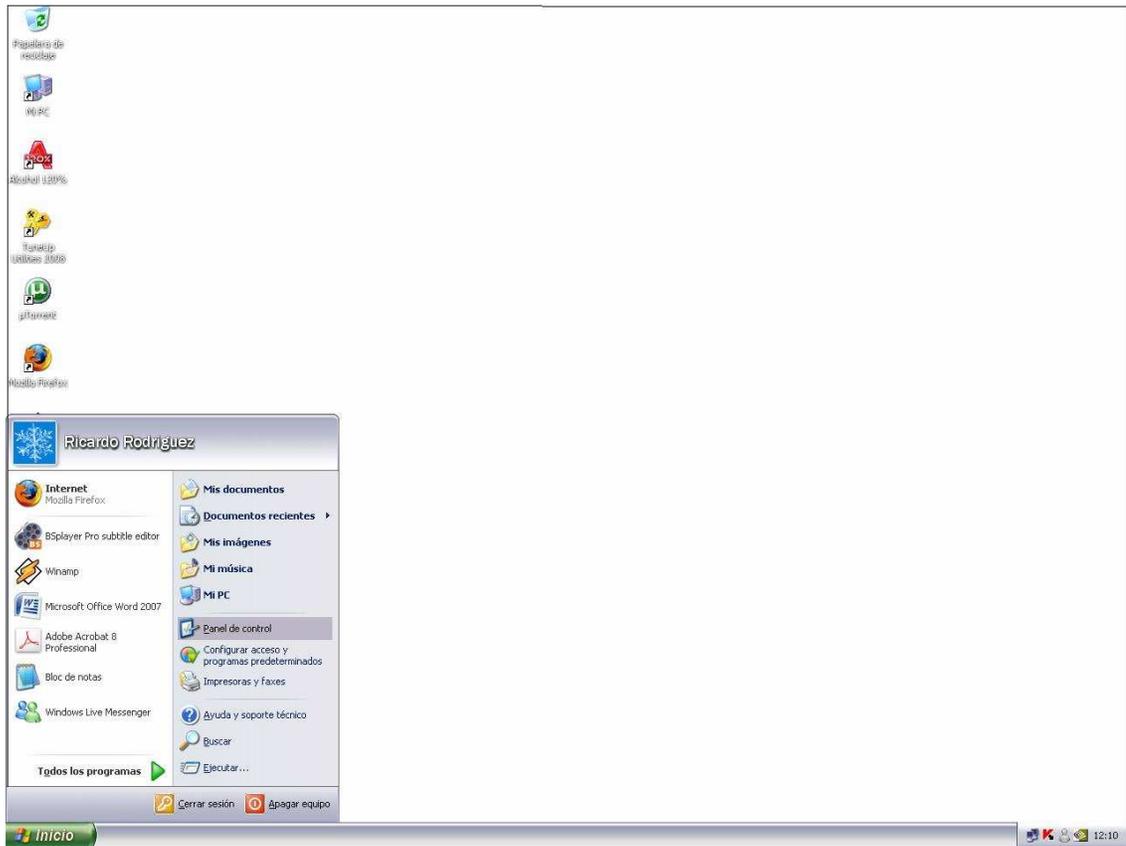


Fuente: Autores

Luego de seguir los pasos anteriores para la instalación del driver, configuramos el puerto, para finalmente trabajar con el entrenador digital.

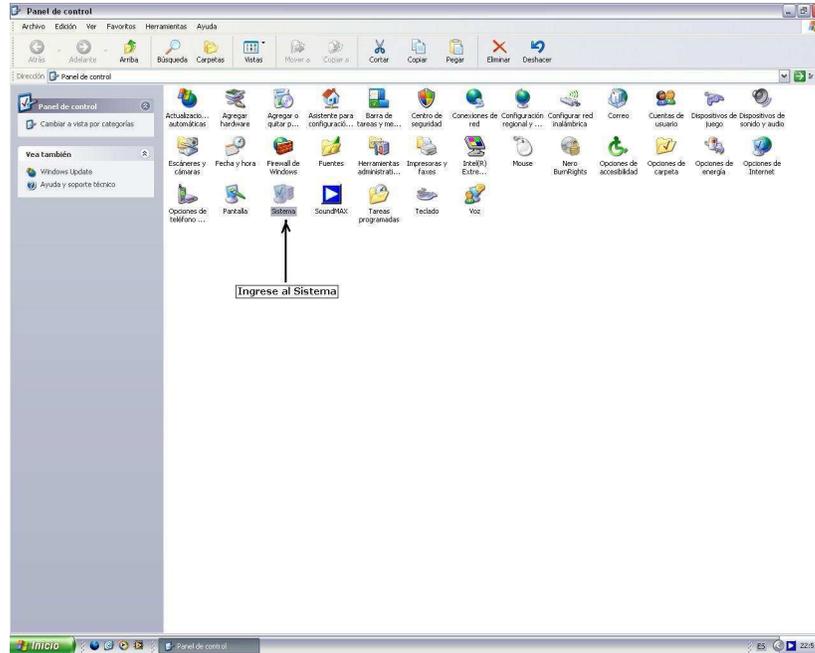
A continuación se muestra las siguientes figuras con su respectiva descripción para la configuración del puerto.

**Figura. Inicio-Panel de Control**



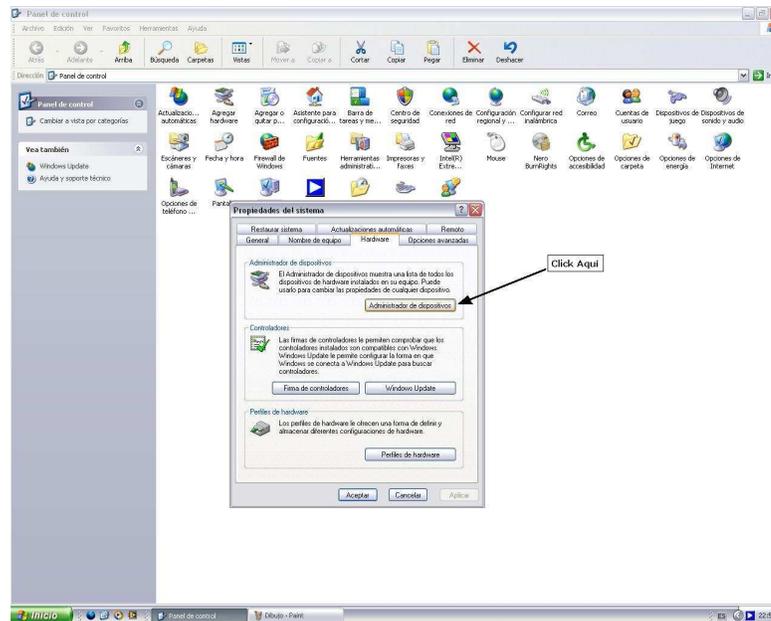
Fuente: Autores

**Figura. Ingreso al Sistema**



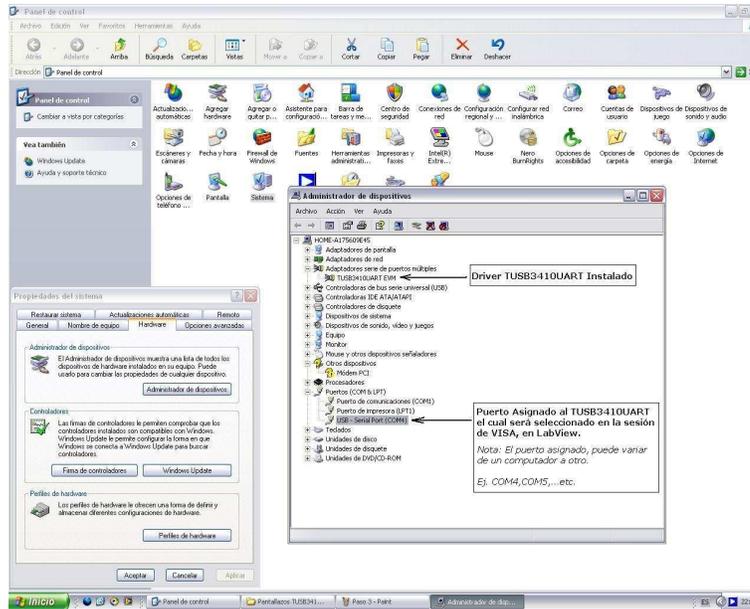
Fuente: Autores

**Figura. Administrador de Dispositivos**



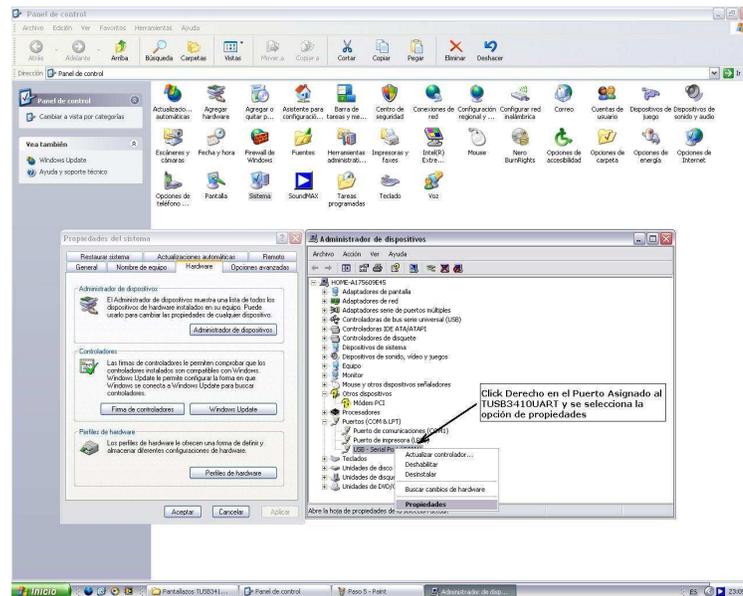
Fuente: Autores

Figura. Asignación de puerto TUSB3410UART



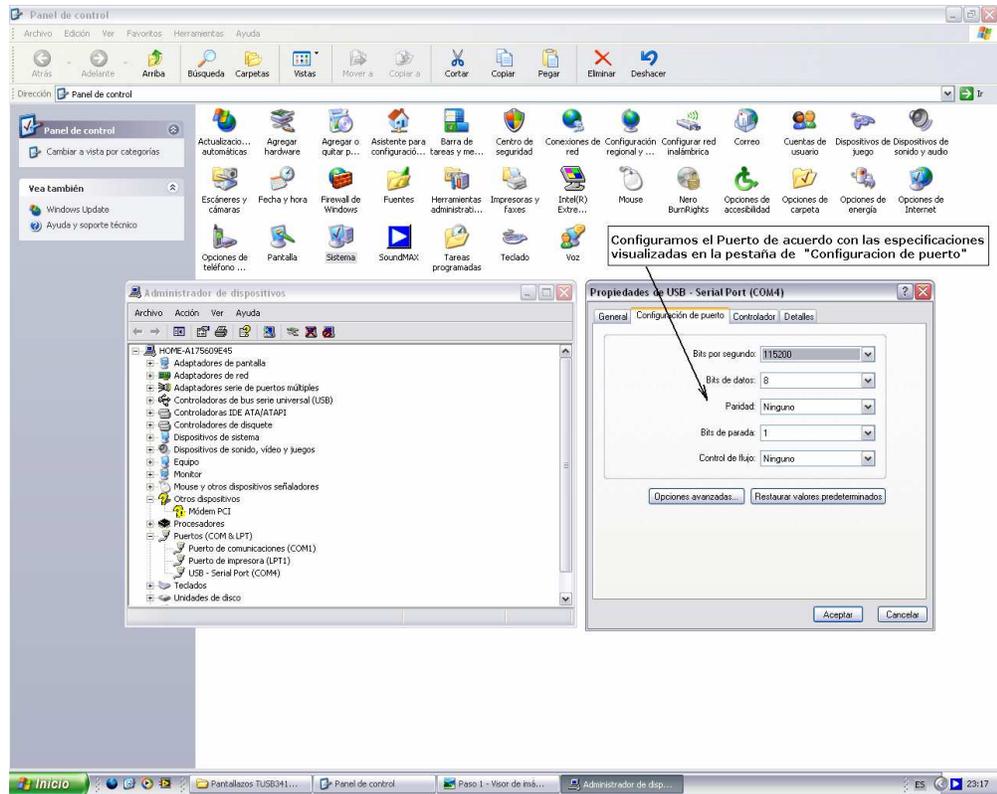
Fuente: Autores

Figura. Propiedades del Puerto Asignado TUSB3410UART



Fuente: Autores

Figura. Configuración del Puerto.



Fuente: Autores

## **B. Características del Nucleo8052**

El núcleo 8052 del microcontrolador es muy óptimo, ofrece hasta un rendimiento pico de 20 MIPS (Millones de Instrucciones por Segundo) para un máximo rendimiento. Tres diferentes memorias habilitadas que ofrecen hasta 62 Kbytes de memoria de programa no-volátil Flash/EE. 4 Kbytes de no-volátil de memoria de datos Flash/EE. 256 bytes de RAM, y 2Kbytes de RAM extendida que están integradas en el integrado.

### **Memoria de Programa**

El ADuC841 tiene 4 bloques de memoria:

- Hasta 62 kBytes en el chip Flash/EE de la memoria de programa.
- 4 kBytes en el chip Flash/EE en la memoria de datos
- 256 bytes de RAM de propósito general.
- 2 kBytes of internal XRAM.

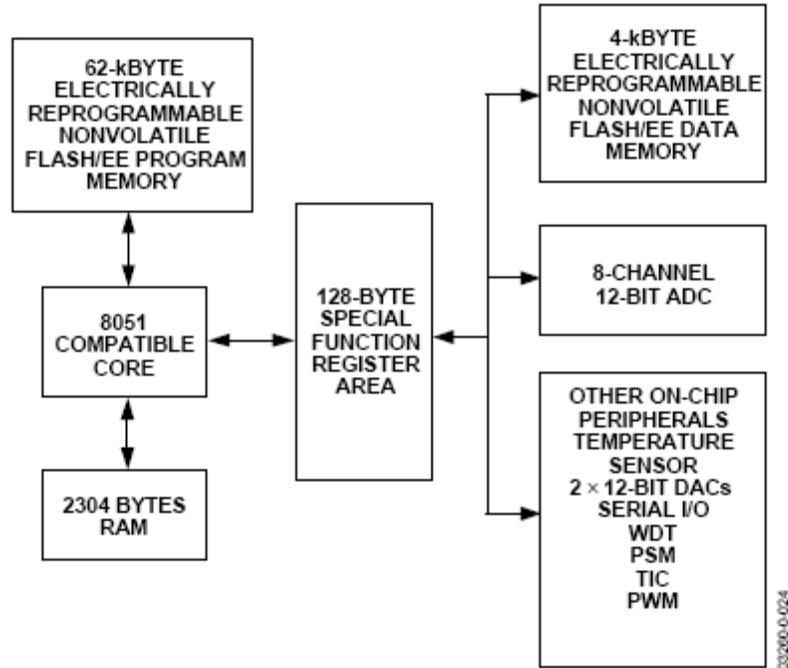
### **Memoria de Programa Flash/EE**

Esta parte provee hasta 62 kBytes de memoria de programa Flash/EE para código del usuario. El usuario solo puede correr el código desde la memoria interna solamente. Esta parte no soporta “roll-over” desde F7FFH hasta el espacio de código interno. F800H espacio en código externo. Desde 2048 bytes entre F800H y FFFFH aparece como instrucciones por código de usuario NOP. El espacio de código interno puede ser descargado por el puerto serial UART mientras el dispositivo esta en el circuito. 56 kBytes de memoria de programa que pueden ser reprogramada durante el tiempo transcurrido. Este espacio en código puede ser actualizado en el campo por un protocolo definido por el usuario, o puede ser usado como memoria de datos. Para un modelo de memoria de 32 kBytes, el tope es de 8 kBytes de funcionamiento como espacio en la ULOAD.

### **Mapa de Memoria de Registros de Funciones Especiales**

El espacio de SFR esta mapeado dentro de los 128 bytes superior del espacio en la memoria interna de datos y se accesa solamente direccionando directamente. Este provee una interfaz entre la CPU y todos los chips periféricos. Un diagrama de bloques muestra el modelo de programación de las áreas del SFR en la figura. Todos los registros, el “program counter” (PC) y los cuatro registros del banco de propósito general están en el área de SFR.- Los registros de SFR incluyen control, configuración, y registro de datos, el cual provee una interfaz entre la CPU y todos sus periféricos.

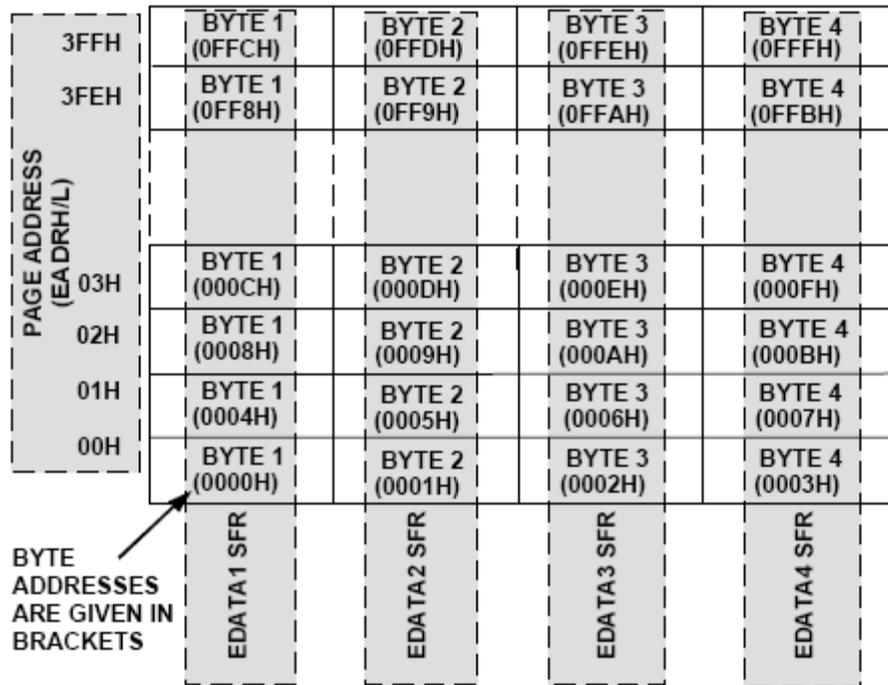
## Diagrama de Bloques: Modelo de Programación de las Áreas del SFR.



### Memoria de datos Flash/EE

4 kBytes de memoria de datos Flash/EE habilitadas para que el usuario pueda acceder indirectamente vía a al grupo de registros de control mapeados dentro del área de función de registros especiales (SFR). Un grupo de 4 registros de datos (EDATA1–4) es usado para reservar 4 bytes de datos a cada página. Un diagrama de bloques del interfaz del SFR para la tabla de la memoria de datos Flash/EE como se muestra en la figura.

**Diagrama de Bloques del Interfaz del SFR para la tabla de la memoria de datos Flash/EE.**



**ALE**

La salida del pin ALE estándar en el 8052 es un 1/6 de la frecuencia del reloj de operación del núcleo. El ADuC841, el pin ALE opera de la siguiente forma. Para un ciclo de instrucción de máquina, ALE es alto para el primer medio ciclo de máquina y luego es bajo para la segunda mitad del ciclo de máquina. La salida ALE es el núcleo de la frecuencia de operación. Para dos o más ciclos de máquinas de instrucción, ALE es alto para el primer medio ciclo de máquina y bajo para el resto ciclos de máquinas.

**Memoria de datos externa XRAM**

Compatible con un núcleo estándar 8051, el ADUC841 puede acceder a la memoria externa de datos usando la instrucción MOVX. La instrucción MOVX automáticamente controla varias salidas para el acceso de la memoria de datos. Esta parte, sin embargo, pueda acceder hasta 16 MBytes de memoria externa de

datos. Esto es una mejora de los 64KBytes del espacio de la memoria de datos externa habilitada por el núcleo 8051

### **A.2.5 Memoria de datos interna XRAM**

Contiene 2 KBytes de memoria de datos. Esta memoria, aunque es interna en el integrado, también se puede tener acceso por la instrucción. Los 2 KBytes la XRAM interna están mapeados dentro de la parte inferior de los 2 KBytes del espacio de direccionamiento externo si los bits CFG841/CFG842 son altos. Por otro lado el acceso a la memoria externa de datos solo ocurre en el 8051. Cuando se usa la memoria interna XRAM, los puertos 0 y 2 son libres para propósito general para I/O.

### **Registros de Funciones Especiales (SFR)**

#### **Registro Acumulador (ACC)**

ACC es el registro acumulador y es usado para operaciones matemáticas incluyendo la suma, la resta, multiplicación y división de números enteros, y manipulación de bits Booleanos. El neumónico para la instrucción específica del acumulador se refiere al acumulador como A.

#### **Registro B**

El registro B es usado con el acumulador (ACC) para operaciones de división y multiplicación. Para otras operaciones ésta se puede tratar como registro de plataforma forzada de propósito general.

#### **Stack Pointer**

El SP del SFR es el stack pointer y es usado para controlar una dirección interna de RAM que es llamada la parada del stack. El registro SP es incrementado antes del almacenar el dato durante la pulsación y el llamado de la ejecución. Mientras el stack pueda estar en cualquier lugar del chip en la RAM, el registro SP es inicializado en 07H después se resetea, el cual causa que el stack empiece en la localidad de 08H.

#### **Data Pointer**

El Data Pointer es un registro de 3 hasta 8 bits llamado DPP (page byte), DHP (high byte), y DPL (low byte). Estos son usados para proporcionar direcciones de memoria para código de acceso interno y externo y acceso externo de datos.

Estos pueden ser manipulados como registros de 16 bits (DPTR = DPH, DPL), aunque la instrucción INC DPTR automáticamente coloca un carry a DPP, o como tres registros independientes de 8 bits (DPP, DPH, DPL).

### Program Status Word (PSW)

El SFR del PSW contiene varios bits que reflejan el estado actual de la CPU, como se muestra en la siguiente tabla.

Dirección del SFR	D0H
Power-On por defecto	00H
Bit Direccionable	Yes

### Tabla de Designación de Bits del PSW.

Bit	Name	Description
7	CY	Carry Flag.
6	AC	Auxiliary Carry Flag.
5	F0	General-Purpose Flag.
4	RS1	Register Bank Select Bits.
3	RS0	RS1      RS0      Selected Bank
		0      0      0
		0      1      1
		1      0      2
		1      1      3
2	OV	Overflow Flag.
1	F1	General-Purpose Flag.
0	P	Parity Bit.

### Power Control (PCON)

El PCON del SFR contiene bits para opciones power-saving y para los estados de las banderas de propósito general como se muestra en la tabla.

Dirección del SFR	87H
Power-On por defecto	00H
Bit Direccionable	NO

## Tabla de Designación de Bits del PCON.

Bit No.	Name	Description
7	SMOD	Double UART Baud Rate.
6	SERIPD	I <sup>2</sup> C/SPI Power-Down Interrupt Enable.
5	INT0PD	INT0 Power-Down Interrupt Enable.
4	ALEOFF	Disable ALE Output.
3	GF1	General-Purpose Flag Bit.
2	GF0	General-Purpose Flag Bit.
1	PD	Power-Down Mode Enable.
0	IDL	Idle Mode Enable.

## Set de Instrucciones

### Tabla de Instrucciones de Operaciones Aritméticas.

Mnemonic	Description	Bytes	Cycles
<b>Arithmetic</b>			
ADD A,Rn	Add register to A	1	1
ADD A,@Ri	Add indirect memory to A	1	2
ADD A,dir	Add direct byte to A	2	2
ADD A,#data	Add immediate to A	2	2
ADDC A,Rn	Add register to A with carry	1	1
ADDC A,@Ri	Add indirect memory to A with carry	1	2
ADDC A,dir	Add direct byte to A with carry	2	2
ADDC A,#data	Add immediate to A with carry	2	2
SUBB A,Rn	Subtract register from A with borrow	1	1
SUBB A,@Ri	Subtract indirect memory from A with borrow	1	2
SUBB A,dir	Subtract direct from A with borrow	2	2
SUBB A,#data	Subtract immediate from A with borrow	2	2
INC A	Increment A	1	1
INC Rn	Increment register	1	1
INC @Ri	Increment indirect memory	1	2
INC dir	Increment direct byte	2	2
INC DPTR	Increment data pointer	1	3
DEC A	Decrement A	1	1
DEC Rn	Decrement register	1	1
DEC @Ri	Decrement indirect memory	1	2
DEC dir	Decrement direct byte	2	2
MUL AB	Multiply A by B	1	9
DIV AB	Divide A by B	1	9
DA A	Decimal adjust A	1	2

## Tabla de Instrucciones de Operaciones Lógicas.

Mnemonic	Description	Bytes	Cycles
<b>Logic</b>			
ANL A,Rn	AND register to A	1	1
ANL A,@Ri	AND indirect memory to A	1	2
ANL A,dir	AND direct byte to A	2	2
ANL A,#data	AND immediate to A	2	2
ANL dir,A	AND A to direct byte	2	2
ANL dir,#data	AND immediate data to direct byte	3	3
ORL A,Rn	OR register to A	1	1
ORL A,@Ri	OR indirect memory to A	1	2
ORL A,dir	OR direct byte to A	2	2
ORL A,#data	OR immediate to A	2	2
ORL dir,A	OR A to direct byte	2	2
ORL dir,#data	OR immediate data to direct byte	3	3
XRL A,Rn	Exclusive-OR register to A	1	1
XRL A,@Ri	Exclusive-OR indirect memory to A	2	2
XRL A,#data	Exclusive-OR immediate to A	2	2
XRL dir,A	Exclusive-OR A to direct byte	2	2
XRL A,dir	Exclusive-OR indirect memory to A	2	2
XRL dir,#data	Exclusive-OR immediate data to direct	3	3
CLR A	Clear A	1	1
CPL A	Complement A	1	1
SWAP A	Swap nibbles of A	1	1
RL A	Rotate A left	1	1
RLC A	Rotate A left through carry	1	1
RR A	Rotate A right	1	1
RRC A	Rotate A right through carry	1	1

## Tabla de Instrucciones de Operaciones de Transferencia Datos.

Mnemonic	Description	Bytes	Cycles
<b>Data Transfer</b>			
MOV A,Rn	Move register to A	1	1
MOV A,@Ri	Move indirect memory to A	1	2
MOV Rn,A	Move A to register	1	1
MOV @Ri,A	Move A to indirect memory	1	2
MOV A,dir	Move direct byte to A	2	2
MOV A,#data	Move immediate to A	2	2
MOV Rn,#data	Move register to immediate	2	2
MOV dir,A	Move A to direct byte	2	2
MOV Rn,dir	Move register to direct byte	2	2
MOV dir,Rn	Move direct to register	2	2
MOV @Ri,#data	Move immediate to indirect memory	2	2
MOV dir,@Ri	Move indirect to direct memory	2	2
MOV @Ri,dir	Move direct to indirect memory	2	2
MOV dir,dir	Move direct byte to direct byte	3	3
MOV dir,#data	Move immediate to direct byte	3	3
MOV DPTR,#data	Move immediate to data pointer	3	3
MOVC A,@A+DPTR	Move code byte relative DPTR to A	1	4
MOVC A,@A+PC	Move code byte relative PC to A	1	4
MOVX A,@Ri	Move external (A8) data to A	1	4
MOVX A,@DPTR	Move external (A16) data to A	1	4
MOVX @Ri,A	Move A to external data (A8)	1	4
MOVX @DPTR,A	Move A to external data (A16)	1	4
PUSH dir	Push direct byte onto stack	2	2
POP dir	Pop direct byte from stack	2	2
XCH A,Rn	Exchange A and register	1	1
XCH A,@Ri	Exchange A and indirect memory	1	2
XCHD A,@Ri	Exchange A and indirect memory nibble	1	2
XCH A,dir	Exchange A and direct byte	2	2

## Tabla de Instrucciones de Operaciones Booleanas.

Mnemonic	Description	Bytes	Cycles
<b>Boolean</b>			
CLR C	Clear carry	1	1
CLR bit	Clear direct bit	2	2
SETB C	Set carry	1	1
SETB bit	Set direct bit	2	2
CPL C	Complement carry	1	1
CPL bit	Complement direct bit	2	2
ANL C,bit	AND direct bit and carry	2	2
ANL C,/bit	AND direct bit inverse to carry	2	2
ORL C,bit	OR direct bit and carry	2	2
ORL C,/bit	OR direct bit inverse to carry	2	2
MOV C,bit	Move direct bit to carry	2	2
MOV bit,C	Move carry to direct bit	2	2

## Tabla de Instrucciones de Operaciones de Bifurcación.

Mnemonic	Description	Bytes	Cycles
<b>Bifurcación</b>			
JMP @A+DPTR	Jump indirect relative to DPTR	1	3
RET	Return from subroutine	1	4
RETI	Return from interrupt	1	4
ACALL addr11	Absolute jump to subroutine	2	3
AJMP addr11	Absolute jump unconditional	2	3
SJMP rel	Short jump (relative address)	2	3
JC rel	Jump on carry equal to 1	2	3
JNC rel	Jump on carry equal to 0	2	3
JZ rel	Jump on accumulator = 0	2	3
JNZ rel	Jump on accumulator not equal to 0	2	3
DJNZ Rn,rel	Decrement register, JNZ relative	2	3
LJMP	Long jump unconditional	3	4
LCALL addr16	Long jump to subroutine	3	4
JB bit,rel	Jump on direct bit = 1	3	4
JNB bit,rel	Jump on direct bit = 0	3	4
JBC bit,rel	Jump on direct bit = 1 and clear	3	4
CJNE A,dir,rel	Compare A, direct JNE relative	3	4
CJNE A,#data,rel	Compare A, immediate JNE relative	3	4
CJNE Rn,#data,rel	Compare register, immediate JNE relative	3	4
CJNE @Ri,#data,rel	Compare indirect, immediate JNE relative	3	4
DJNZ dir,rel	Decrement direct byte, JNZ relative	3	4

## Tabla de Instrucciones de Operaciones de Misceláneas.

Mnemonic	Description	Bytes	Cycles
<b>Miscellaneous</b>			
NOP	No operation	1	1

## Estructura de Interrupción

El ADuC841 provee un total de nueve fuentes de interrupción con dos niveles de prioridad. El control y la configuración del sistema de interrupción se llevan a cabo por las interrupciones relacionadas con el SFRs:

IE                    Registro de Interrupción Permitida  
IP                    Registro de Prioridad de Interrupción  
IEIP2                Registro de Interrupciones Secundarias Permitida

## Activación de las Interrupciones

Se realiza de acuerdo a los registros de interrupciones IE IP y IEIP2 de la siguiente forma:

IE                    *Registro de Interrupción Permitida*  
SFR Address                    A8H  
Power-On Default                00H  
Bit Direccionable                Yes

## Tabla Designación de Bit en el SFR del Registro de Interrupción Permitida (IE).

Bit No.	Name	Description
7	EA	Set by the user to enable, or cleared to disable all interrupt sources.
6	EADC	Set by the user to enable, or cleared to disable ADC interrupts.
5	ET2	Set by the user to enable, or cleared to disable Timer 2 interrupts.
4	ES	Set by the user to enable, or cleared to disable UART serial port interrupts.
3	ET1	Set by the user to enable, or cleared to disable 0 Timer 1 interrupts.
2	EX1	Set by the user to enable, or cleared to disable External Interrupt 1.
1	ET0	Set by the user to enable, or cleared to disable Timer 0 interrupts.
0	EX0	Set by the user to enable, or cleared to disable External Interrupt 0 .

IP                    *Registro de Prioridad de Interrupción*  
SFR Address                    B8H  
Power-On Default                00H  
Bit Direccionable                Yes



interrupciones del mismo nivel de prioridad ocurren simultáneamente, se obtiene una secuencia como es mostrada en la tabla.

**Tabla Nivel de Prioridad dentro de una Interrupción.**

Source	Priority	Description
PSMI	1 (Highest)	Power Supply Monitor Interrupt.
WDS	2	Watchdog Timer Interrupt.
IE0	2	External Interrupt 0.
ADCI	3	ADC Interrupt.
TF0	4	Timer/Counter 0 Interrupt.
IE1	5	External Interrupt 1.
TF1	6	Timer/Counter 1 Interrupt.
ISPI/I2CI	7	SPI Interrupt/I <sup>2</sup> C Interrupt.
RI + TI	8	Serial Interrupt.
TF2 + EXF2	9	Timer/Counter 2 Interrupt.
TII	11 (Lowest)	Time Interval Counter Interrupt.

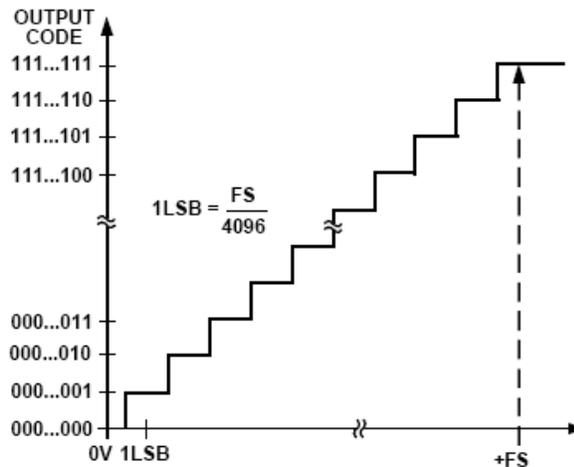
## Conversor Análogo/Digital (ADC)

### Función de Transferencia

El rango de entrada análoga para el ADC es 0 V a VREF. Para este rango, el código designado de transición ocurre a medio camino entre los valores de entero sucesivo LSB, 0.5 LSB, 1.5 LSB, 2.5 LSB...FS -1.5 LSB. La codificación de la salida es directamente binaria con 1 LSB = FS/4096 ó 2.5 V/4096 = 0.61 mV cuando VREF = 2.5 V.

La característica ideal de entrada/salida para 0 V a VREF esta en el rango como se muestra en la figura.

## Figura Función de Transferencia ADC



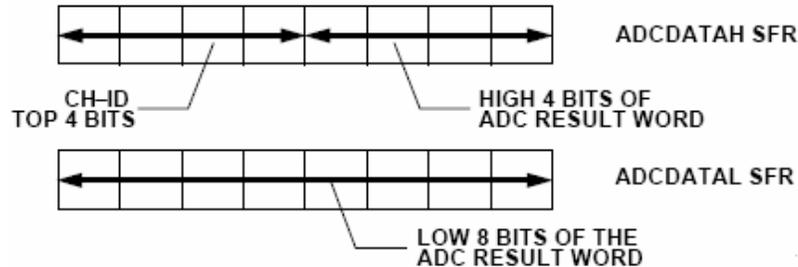
## Configuración del ADC

Las partes del ADC de una aproximación sucesiva se conducen por una división descendente del master clock. Para asegurar la adecuada operación del ADC, el reloj del ADC debe estar entre los 400KHz y los 8.38MHz. Las frecuencias dentro de este rango pueden conseguirse fácilmente con el master clock desde frecuencias de 400KHz hasta sobre los 16 MHz. En modo de conversión continua una nueva conversión empieza cada una vez previamente una finaliza. La tasa de muestreo es simplemente al inverso del tiempo total de conversión descrito previamente.

## Operación Típica

Una vez configurado los registros ADCCON 1-3 en el SFR, el ADC convierte la entrada analógica y proporciona una palabra en el ADCCON de 12-bit como resultado en el ADCDATAH/L en el SFR. El tope de 4 bits del ADCDATAH del SFR son escritos por el canal de selección de bits para identificar el resultado. El formato del ADC de 12 bits del resultado de la palabra se muestra en la siguiente figura.

## Formato del resultado de la palabra.



- Los registros del ADC utilizados en el SFR son los siguientes:

### ADCCON1— (ADC Control SFR 1)

El registro ADCCON1 controla la conversión y el tiempo de adquisición, el modo de conversión por hardware, y el modo de apagado como se muestra detalladamente a continuación.

SFR Address	EFH
SFR Power-On Default	40H
Bit Direccionable	No

### Tabla de Designación de Bit en el Registro del ADCCON1 del SFR.

Bit No.	Name	Description															
7	MD1	The mode bit selects the active operating mode of the ADC. Set by the user to power up the ADC. Cleared by the user to power down the ADC.															
6	EXT_REF	Set by the user to select an external reference. Cleared by the user to use the internal reference.															
5	CK1	The ADC clock divide bits (CK1, CK0) select the divide ratio for the PLL master clock (ADuC842/ADuC843) or the external crystal (ADuC841) used to generate the ADC clock. To ensure correct ADC operation, the divider ratio must be chosen to reduce the ADC clock to 8.38 MHz or lower. A typical ADC conversion requires 16 ADC clocks plus the selected acquisition time. The divider ratio is selected as follows: <table> <tr> <td>CK1</td> <td>CK0</td> <td>MCLK Divider</td> </tr> <tr> <td>0</td> <td>0</td> <td>32</td> </tr> <tr> <td>0</td> <td>1</td> <td>4 (Do not use with a CD setting of 0)</td> </tr> <tr> <td>1</td> <td>0</td> <td>8</td> </tr> <tr> <td>1</td> <td>1</td> <td>2</td> </tr> </table>	CK1	CK0	MCLK Divider	0	0	32	0	1	4 (Do not use with a CD setting of 0)	1	0	8	1	1	2
CK1	CK0		MCLK Divider														
0	0		32														
0	1		4 (Do not use with a CD setting of 0)														
1	0	8															
1	1	2															
4	CK0																
3	AQ1	The ADC acquisition select bits (AQ1, AQ0) select the time provided for the input track-and-hold amplifier to acquire the input signal. An acquisition of three or more ADC clocks is recommended; clocks are as follows: <table> <tr> <td>AQ1</td> <td>AQ0</td> <td>No. ADC Clks</td> </tr> <tr> <td>0</td> <td>0</td> <td>1</td> </tr> <tr> <td>0</td> <td>1</td> <td>2</td> </tr> <tr> <td>1</td> <td>0</td> <td>3</td> </tr> <tr> <td>1</td> <td>1</td> <td>4</td> </tr> </table>	AQ1	AQ0	No. ADC Clks	0	0	1	0	1	2	1	0	3	1	1	4
AQ1	AQ0		No. ADC Clks														
0	0		1														
0	1		2														
1	0	3															
1	1	4															
2	AQ0																
1	T2C	The Timer 2 conversion bit (T2C) is set by the user to enable the Timer 2 overflow bit to be used as the ADC conversion start trigger input.															
0	EXC	The external trigger enable bit (EXC) is set by the user to allow the external Pin P3.5 ( $\overline{\text{CONVST}}$ ) to be used as the active low convert start input. This input should be an active low pulse (minimum pulse width >100 ns) at the required sample rate.															

## ADCCON2— (ADC Control SFR 2)

El registro ADCCON2 controla el canal de selección del ADC y el modo de conversión como se muestra detalladamente a continuación.

SFR Address	D8H
SFR Power-On Default	00H
Bit Direccionable	Yes

### Tabla de Designación de Bit en el Registro del ADCCON2 del SFR.

Bit No.	Name	Description				
7	ADCI	ADC Interrupt Bit. Set by hardware at the end of a single ADC conversion cycle or at the end of a DMA block conversion. Cleared by hardware when the PC vectors to the ADC interrupt service routine. Otherwise, the ADCI bit is cleared by user code.				
6	DMA	DMA Mode Enable Bit. Set by the user to enable a preconfigured ADC DMA mode operation. A more detailed description of this mode is given in the ADC DMA Mode section. The DMA bit is automatically set to 0 at the end of a DMA cycle. Setting this bit causes the ALE output to cease; it will start again when DMA is started and will operate correctly after DMA is complete.				
5	CCONV	Continuous Conversion Bit. Set by the user to initiate the ADC into a continuous mode of conversion. In this mode, the ADC starts converting based on the timing and channel configuration already set up in the ADCCON SFRs; the ADC automatically starts another conversion once a previous conversion has completed.				
4	SCONV	Single Conversion Bit. Set to initiate a single conversion cycle. The SCONV bit is automatically reset to 0 on completion of the single conversion cycle.				
3	CS3	Channel Selection Bits. Allow the user to program the ADC channel selection under software control. When a conversion is initiated, the converted channel is the one pointed to by these channel selection bits. In DMA mode, the channel selection is derived from the channel ID written to the external memory.				
2	CS2					
1	CS1					
0	CS0					
	CS3		CS2	CS1	CS0	CH#
	0		0	0	0	0
	0		0	0	1	1
	0		0	1	0	2
	0	0	1	1	3	
	0	1	0	0	4	
	0	1	0	1	5	
	0	1	1	0	6	
	0	1	1	1	7	
	1	0	0	0	Temp Monitor	Requires minimum of 1 $\mu$ s to acquire.
	1	0	0	1	DAC0	Only use with internal DAC output buffer on.
	1	0	1	0	DAC1	Only use with internal DAC output buffer on.
	1	0	1	1	AGND	
	1	1	0	0	V <sub>REF</sub>	
	1	1	1	1	DMA STOP	Place in XRAM location to finish DMA sequence; refer to the ADC DMA Mode section.
						All other combinations reserved.



## Configuración Típica del ADC en modo DMA

El modo de configuración DMA consiste en los siguientes pasos:

1. El ADC deberá estar en modo apagado. Esto es para asegurar que MD1 y MD0 sean puestos en 0 en ADCCON1.
2. El apuntador de direccionamiento del DMA deberá ser puesto para empezar en la dirección donde serán escritos los resultados del ADC. Esto es para escribir en el modo DMA la dirección que apunta a DMAL DMAH y DMAP. DMAL deberá ser escrito primero luego DMAH y por consiguiente DMAP.
3. La memoria externa deberá ser preconfigurada. Esto consiste en escribir el IDs requerido para el canal ADC dentro los 4 bits de parada en la segunda ubicación en memoria externa SRAM, empezando a primero especificando la dirección para el apuntador del DMA. Porque el modo DMA del ADC opera independientemente del núcleo del ADuC841, y si es necesario provee un comando de parada. Esto se hace repitiendo el ultimo ID del canal para ser convertido a 1111 en la siguiente selección de canal. Una típica preconfiguración de la memoria externa se muestra en la figura.

### Preconfiguración típica de la memoria externa DMA.

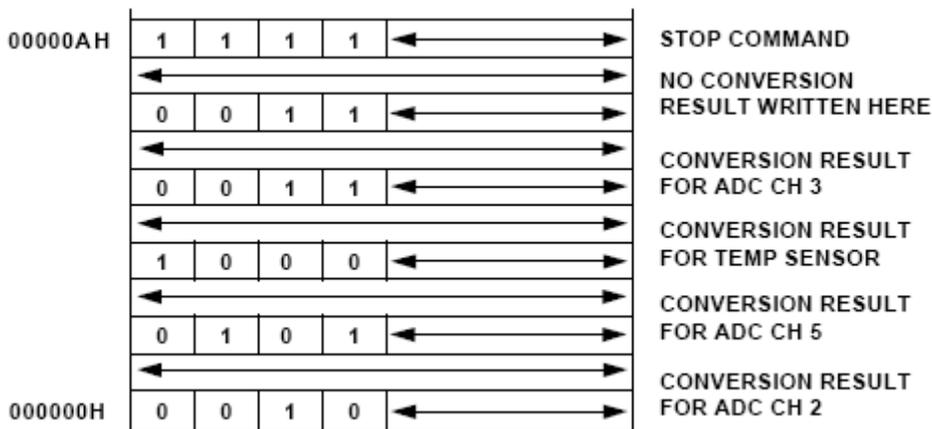
00000AH	1	1	1	1		STOP COMMAND
						REPEAT LAST CHANNEL FOR A VALID STOP CONDITION
	0	0	1	1		
						CONVERT ADC CH 3
	1	0	0	0		CONVERT TEMP SENSOR
	0	1	0	1		CONVERT ADC CH 5
000000H	0	0	1	0		CONVERT ADC CH 2

4. El modo DMA se inicializa cuando se escribe al SFR del ADC la siguiente secuencia:
  - a) ADCCON2 es escrito para habilitar el modo DMA, i.e.,  
MOV ADCCON2, #40H; habilita el modo DMA.

- b) ADCCON1 se escribe para configurar el tiempo de conversión y el power-up del ADC. También se puede habilitar el Timer2 para correr las conversiones o provocarlas las conversiones externamente si se requiere.
- c) Las conversiones ADC son iniciadas.

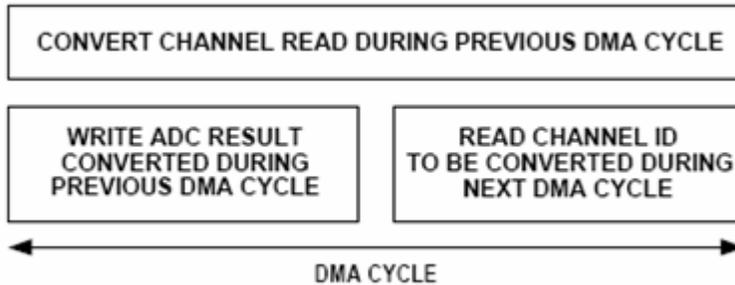
Al realizar esto se inicia una simple conversión, empezando con Timer2, corre la conversión en el Timer2, o recibe una interrupción externa. Cuando la conversión DMA esta completada, el bit de interrupción del ADC ADCI, se coloca alto por hardware, y la SRAM externa contiene el nuevo resultado de conversión del ADC como se muestras en la figura “ ” . Note que el resultado no se escribe en loas dos últimas localizaciones de memoria. Cuando el modo lógico DMA esa activo, toma la responsabilidad de almacenar los resultados del ADC fuera, tanto del usuario como de la parte lógica del núcleo. Como la interfaz DMA escribe los resultados del ADC en la memoria externa, encargándose de la interfaz de la memoria externa desde el núcleo. Además cualquier instrucción del núcleo tiene acceso a la memoria externa mientras que el modo DMA cuando es habilitado no tiene acceso a la memoria externa.

### Configuración Típica de la Memoria Externa después de la Operación ADC DMA



La operación lógica DMA se aplica al reloj ADC y usa la pipelining para llevar a cabo las conversiones ADC y acceder a la memoria externa al mismo tiempo. El tiempo que toma realizar una sola conversión del ADC es llamada ciclo DMA. LA acción se lleva a cabo durante un ciclo lógico DMA típico como se muestra en la figura.

## Ciclo DMA



Durante un ciclo DMA las siguientes acciones son realizadas lógicamente por el DMA:

1. Una conversión ADC es realizada en canal el cual el ID fue leído durante un ciclo previo
2. El resultado de 12-bit y el canal ID de conversión realizado por el ciclo previamente son escritos en memoria externa.
3. El ID del siguiente canal para ser convertido es leído desde la memoria externa.

## Conversor Digital/Análogo (DAC)

El ADuC841 incorpora dos DAC's de 12-bit de voltaje a la salida. Cada carril del DAC a la salida el voltaje es capaz de conducir 10 k $\Omega$ /100 pF. Cada DAC tiene dos rangos seleccionables de 0 V a VREF y de 0 V a AVDD. Cada uno puede operar en un modo de 12 bit o de 8 bit.

Ambos DAC's parten de un registro de control, DACCON, y cuatro registros de datos DAC1H/L, DAC0/L. En modo de 12 bit asíncrono la salida del voltaje del DAC es actualizada tan pronto como el dato DACL ha sido escrito en el SFR. Además, el registro de datos del DAC deberá ser actualizado como DACH primero, seguido por el DACL.

## Registros

En la siguiente tabla se muestra uno de los registros del DAC con las siguientes características:

SFR Address	AFH
Power-On Default	10H
Bit Direccionable	No

### Tabla de Designación de Bit del Registro CFG841 en el SFR

Bit No.	Name	Description
7	EXSP	Extended SP Enable. When set to 1 by the user, the stack rolls over from SPH/SP = 00FFH to 0100H. When set to 0 by the user, the stack rolls over from SP = FFH to SP = 00H.
6	PWPO	PWM Pin Out Selection. Set to 1 by the user to select P3.4 and P3.3 as the PWM output pins. Set to 0 by the user to select P2.6 and P2.7 as the PWM output pins.
5	DBUF	DAC Output Buffer. Set to 1 by the user to bypass the DAC output buffer. Set to 0 by the user to enable the DAC output buffer.
4	EPM2	Flash/EE Controller and PWM Clock Frequency Configuration Bits. Frequency should be configured such that $F_{osc}/Divide\ Factor = 32\ kHz + 50\%$ .
3	EPM1	EPM2 EPM1 EPM0 Divide Factor
2	EPM0	0 0 0 32 0 0 1 64 0 1 0 128 0 1 1 256 1 0 0 512 1 0 1 1024
1	MSPI	Set to 1 by the user to move the SPI functionality of MISO, MOSI, and SCLOCK to P3.3, P3.4, and P3.5, respectively. Set to 0 by the user to leave the SPI functionality as usual on MISO, MOSI, and SCLOCK pins.
0	XRAMEN	XRAM Enable Bit. When set to 1 by the user, the internal XRAM is mapped into the lower two kBytes of the external address space. When set to 0 by the user, the internal XRAM is not accessible, and the external data memory is mapped into the lower two kBytes of external data memory.

El siguiente registro del DAC es el DACCON con las siguientes características:

SFR Address	FDH
Power-On Default	04H
Bit Direccionable	No

## Tabla de Designación de Bit del Registro DACCON en el SFR.

Bit No.	Name	Description
7	MODE	The DAC MODE bit sets the overriding operating mode for both DACs. Set to 1 by the user to select 8-bit mode (write 8 bits to DACxL SFR). Set to 0 by the user to select 12-bit mode.
6	RNG1	DAC1 Range Select Bit. Set to 1 by the user to select the range for DAC1 as 0 V to $V_{DD}$ . Set to 0 by the user to select the range for DAC1 as 0 V to $V_{REF}$ .
5	RNG0	DAC0 Range Select Bit. Set to 1 by the user to select the range for DAC0 as 0 V to $V_{DD}$ . Set to 0 by the user to select the range for DAC0 as 0 V to $V_{REF}$ .
4	CLR1	DAC1 Clear Bit. Set to 1 by the user to leave the output of DAC1 at its normal level. Set to 0 by the user to force the output of DAC1 to 0 V.
3	CLR0	DAC0 Clear Bit. Set to 1 by the user to leave the output of DAC0 at its normal level. Set to 0 by the user to force the output of DAC0 to 0 V.
2	SYNC	DAC0/1 Update Synchronization Bit. When set to 1, the DAC outputs update as soon as DACxL SFRs are written. The user can simultaneously update both DACs by first updating the DACxL/H SFRs while SYNC is 0. Both DACs then update simultaneously when the SYNC bit is set to 1.
1	PD1	DAC1 Power-Down Bit. Set to 1 by the user to power on DAC1. Set to 0 by the user to power off DAC1.
0	PD0	DAC0 Power-Down Bit. Set to 1 by the user to power on DAC0. Set to 0 by the user to power off DAC0.

## DACxH/L Registro de datos del DAC

Función	Registro de datos DAC, Escrito por el usuario para actualizar la salida del DAC
Dirección SFR	DAC0L (DAC0 Data Low Byte) -> F9H; DAC1L (DAC1 Data Low Byte) -> FBH DACH (DAC0 Data High Byte) -> FAH; DAC1H (DAC1 Data High Byte) -> FCH
Power-On Default	00H todos los 4 registros.
Bit Direccionable	No todos los 4 registros.

Los datos de 12 Bit del DAC deberán ser escritos dentro de DACxH/L tal como está justificado a la derecha tal como DACxL que contiene los 8 bit menos significativos y el DACxH que contiene los 4 bit más significativos.

## Consideraciones Hardware

### *Consumo de Potencia*

El consumo de corriente por las varias secciones del microcontrolador son mostrados en la tabla”. El Valor que representa el núcleo de corriente esta dado por DVDD mientras que el resto (ADC, DAC, y voltaje de referencia) son dadas por el pin de AVDD y puede ser deshabilitado por software cuando no se usa. Otros periféricos del microcontrolador (como lo son el watchdogtimer y power suplí monitor) tienen un consumo insignificante de corriente. Por supuesto el usuario deberá adicionar cualquier fuente de corriente en los pines de I/O por paralelo y serial, y adicionar una fuente para el DAC, en un orden para determinar el total de corriente que necesita los pines de energía. También la corriente por la fuente de DVDD aumenta cerca aproximadamente 10 mA durante el borrado de la Flash/EE y los ciclos de programa.

### **Tabla de Corriente Típica I<sub>DD</sub> del núcleo y los periféricos.**

	<b>V<sub>DD</sub> = 5 V</b>	<b>V<sub>DD</sub> = 3 V</b>
Core (Normal Mode)	(2.2 nA × M <sub>CLK</sub> )	(1.4 nA × M <sub>CLK</sub> )
ADC	1.7 mA	1.7 mA
DAC (Each)	250 μA	200 μA
Voltage Ref	200 μA	150 μA

La corriente de operación DVDD tiene como principal función la velocidad del reloj, la expresión para el núcleo de la fuente de corriente en la tabla anterior” ” están dadas por MCLK, la frecuencia del reloj del núcleo. Ponga un valor para el MCLK en hertz para determinar la corriente consumida por el núcleo a esta frecuencia de oscilación. Entonces el ADC y los DAC’s pueden ser habilitados o deshabilitados por software, además solamente la corrientes de los periféricos podrás esperar utilizar, Además no olvides incluir la corriente de fuente por los pines de I/O, pines del puerto serial, salidas del DAC, y además la corriente adicional utilizada durante el borrado de la Flash/EE y ciclos de programa. Un software siempre enciende y apaga el integrado para activar y apagar en modo normal dentro el modo oculto, y también dentro del modo apagado.

## PLL

### Funcionamiento

El ADuC841 opera directamente desde un cristal externo. El núcleo puede operar a esta frecuencia (16.78 DMZ) o en los submúltiplos binarios de ella permitiendo el ahorro de energía en caso de funcionamiento máximo del núcleo cuando no es requerido. El reloj del núcleo por defecto esta en el reloj del PLL dividido por 8 o 2.097152 MHz. El reloj del ADC también se deriva del reloj del PLL, con la tasa del modulador empieza la misma frecuencia de oscilación del cristal. El procedimiento de selección de frecuencias se asegura que el modulador y el núcleo sean síncronos, sin importar la tasa del reloj del núcleo. El registro de control del PLL es el PLLCON.

A 5 V el reloj del núcleo puede ser puesto en alto a un máximo de 16.78MHz, mientras que a 3 V la máxima configuración de reloj es de 8.38MHz. Note que el ADuC841, cambiando los bits del divisor del reloj (CD) en el PLLCON causa que la velocidad del núcleo cambie. La velocidad del núcleo es dada por  $\text{frec Cristal} / 2\text{CD}$ .

### Registro

El registro PLLCON posee las siguientes características:

SFR Address	D7H
Power-On Default	53H
Bit Direccionable	No

## Tabla de Designación de Bit del Registro PLLCON en el SFR.

Bit No.	Name	Description																																				
7	OSC_PD	Oscillator Power-Down Bit. Set by the user to halt the 32 kHz oscillator in power-down mode. Cleared by the user to enable the 32 kHz oscillator in power-down mode. This feature allows the TIC to continue counting even in power-down mode.																																				
6	LOCK	PLL Lock Bit. This is a read-only bit. Set automatically at power-on to indicate that the PLL loop is correctly tracking the crystal clock. If the external crystal subsequently becomes disconnected, the PLL will rail. Cleared automatically at power-on to indicate that the PLL is not correctly tracking the crystal clock. This may be due to the absence of a crystal clock or an external crystal at power-on. In this mode, the PLL output can be 16.78 MHz $\pm 20\%$ .																																				
5	----	Reserved. Should be written with 0.																																				
4	----	Reserved. Should be written with 0.																																				
3	FINT	Fast Interrupt Response Bit. Set by the user enabling the response to any interrupt to be executed at the fastest core clock frequency, regardless of the configuration of the CD2-0 bits (see below). Once user code has returned from an interrupt, the core resumes code execution at the core clock selected by the CD2-0 bits. Cleared by the user to disable the fast interrupt response feature.																																				
2	CD2	CPU (Core Clock) Divider Bits.																																				
1	CD1	This number determines the frequency at which the microcontroller core operates.																																				
0	CD0	<table border="1"> <thead> <tr> <th>CD2</th> <th>CD1</th> <th>CD0</th> <th>Core Clock Frequency (MHz)</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>0</td> <td>0</td> <td>16.777216</td> </tr> <tr> <td>0</td> <td>0</td> <td>1</td> <td>8.388608</td> </tr> <tr> <td>0</td> <td>1</td> <td>0</td> <td>4.194304</td> </tr> <tr> <td>0</td> <td>1</td> <td>1</td> <td>2.097152 (Default Core Clock Frequency)</td> </tr> <tr> <td>1</td> <td>0</td> <td>0</td> <td>1.048576</td> </tr> <tr> <td>1</td> <td>0</td> <td>1</td> <td>0.524288</td> </tr> <tr> <td>1</td> <td>1</td> <td>0</td> <td>0.262144</td> </tr> <tr> <td>1</td> <td>1</td> <td>1</td> <td>0.131072</td> </tr> </tbody> </table>	CD2	CD1	CD0	Core Clock Frequency (MHz)	0	0	0	16.777216	0	0	1	8.388608	0	1	0	4.194304	0	1	1	2.097152 (Default Core Clock Frequency)	1	0	0	1.048576	1	0	1	0.524288	1	1	0	0.262144	1	1	1	0.131072
CD2	CD1	CD0	Core Clock Frequency (MHz)																																			
0	0	0	16.777216																																			
0	0	1	8.388608																																			
0	1	0	4.194304																																			
0	1	1	2.097152 (Default Core Clock Frequency)																																			
1	0	0	1.048576																																			
1	0	1	0.524288																																			
1	1	0	0.262144																																			
1	1	1	0.131072																																			

## Modulación de Amplitud de Pulsos (PWM)

El PWM usa cinco SFR's: El de control SFR (PWMCON) y los cuatro de datos SFR's (PWM0H, PWM0L, PWM1H, y PWM1L).

PWMCON, como se describirá en la siguiente sección, controla los diferentes modos de operación del PWM, como también la frecuencia de reloj del PWM.

PWM0H/L y PWM1H/L son registros de datos que determinan el tiempo de ciclo de las salidas del PWM. Los pines de salida que usa el PWM son determinados por el registro CFG841 y pueden ser P2.6 y P2.7 o P3.4 y P3.3.

Para usar el software del usuario de PWM, primero se escribe al PWMCON, para seleccionar el modo de operación del PWM y la entrada del reloj del PWM. Escribiendo en el PWMCON también se resetea el contador PWM. Si cualquiera de los modos de operación de 16 bit (modos 1, 3), el usuario por software deberá escribir al PWM0L o PWM1L SFRs primero. Este valor escrito está oculto en el SFR. Escribiendo en el PWM0H o PWM1H los SFRs actualiza ambos el PWMxH y el PWMxL SFRs pero no cambia las salidas hasta terminar el ciclo en progreso

del PWM. Los valores escritos en estos registros de 16 bit's son también usados en el siguiente ciclo PWM.

## Registros

El registro PWMCON posee las siguientes características:

SFR Address                    AEH  
 Power-On Default            00H  
 Bit Direccionable            No

### Tabla de Designación de Bit del Registro PWMCON en el SFR.

Bit No.	Name	Description
7	SNGL	Turns off PMW output at P2.6 or P3.4, leaving the port pin free for digital I/O.
6	MD2	PWM Mode Bits.
5	MD1	The MD2/1/0 bits choose the PWM mode as follows:
4	MD0	
	MD2      MD1      MD0      Mode	
	0          0          0          Mode 0: PWM Disabled	
	0          0          1          Mode 1: Single variable resolution PWM on P2.7 or P3.3	
	0          1          0          Mode 2: Twin 8-bit PWM	
	0          1          1          Mode 3: Twin 16-bit PWM	
	1          0          0          Mode 4: Dual NRZ 16-bit $\Sigma$ - $\Delta$ DAC	
	1          0          1          Mode 5: Dual 8-bit PWM	
	1          1          0          Mode 6: Dual RZ 16-bit $\Sigma$ - $\Delta$ DAC	
	1          1          1          Reserved	
3	CDIV1	PWM Clock Divider.
2	CDIV0	Scale the clock source for the PWM counter as follows:
	CDIV1      CDIV0      Description	
	0          0          PWM Counter = Selected Clock/1	
	0          1          PWM Counter = Selected Clock/4	
	1          0          PWM Counter = Selected Clock/16	
	1          1          PWM Counter = Selected Clock/64	
1	CSEL1	PWM Clock Divider.
0	CSEL0	Select the clock source for the PWM as follows:
	CSEL1      CSEL0      Description	
	0          0          PWM Clock = $f_{XTAL}/15$ , ADuC841 = $f_{OCS}/DIVIDE FACTOR /15$ (see the CFG841 register)	
	0          1          PWM Clock = $f_{XTAL}$ , ADuC841 = $f_{OCS}/DIVIDE FACTOR$ (see the CFG841 register)	
	1          0          PWM Clock = External input at P3.4/TO	
	1          1          PWM Clock = $f_{VCO} = 16.777216$ MHz, ADuC841 = $f_{osc}$	

## Tabla de Designación de Bit del Registro TIMECON en el SFR.

Bit No.	Name	Description															
7	---	Reserved.															
6	TFH	Twenty-Four Hour Select Bit. Set by the user to enable the hour counter to count from 0 to 23. Cleared by the user to enable the hour counter to count from 0 to 255.															
5	ITS1	Interval Timebase Selection Bits. Written by user to determine the interval counter update rate. <table border="1" style="margin-left: 20px;"> <thead> <tr> <th>ITS1</th> <th>ITS0</th> <th>Interval Timebase</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>0</td> <td>1/128 Second</td> </tr> <tr> <td>0</td> <td>1</td> <td>Seconds</td> </tr> <tr> <td>1</td> <td>0</td> <td>Minutes</td> </tr> <tr> <td>1</td> <td>1</td> <td>Hours</td> </tr> </tbody> </table>	ITS1	ITS0	Interval Timebase	0	0	1/128 Second	0	1	Seconds	1	0	Minutes	1	1	Hours
ITS1	ITS0		Interval Timebase														
0	0		1/128 Second														
0	1		Seconds														
1	0		Minutes														
1	1	Hours															
4	ITS0																
3	STI	Single Time Interval Bit. Set by the user to generate a single interval timeout. If set, a timeout clears the TIEN bit. Cleared by the user to allow the interval counter to be automatically reloaded and start counting again at each interval timeout.															
2	TII	TIC Interrupt Bit. Set when the 8-bit interval counter matches the value in the INTVAL SFR. Cleared by user software.															
1	TIEN	Time Interval Enable Bit. Set by the user to enable the 8-bit time interval counter. Cleared by the user to disable the interval counter.															
0	TCEN	Time Clock Enable Bit. Set by the user to enable the time clock to the time interval counters. Cleared by the user to disable the clock to the time interval counters and reset the time interval SFRs to the last value written to them by the user. The time registers (HTHSEC, SEC, MIN, and HOUR) can be written while TCEN is low.															

## Puerto Digitales Bidireccionales

### Visión Global

El ADuC841 tiene puertos de entrada y salida para intercambiar datos con dispositivos externos. Además de la ejecución de propósito generales de los puertos de entrada y salida, algunos puertos son capaces de hacer operaciones con memoria externa mientras otros son multiplexados con funciones alternas para periféricos con características del dispositivo. En general, cuando un periférico es habilitado, estos pines no deben ser usados como propósito general.

### Puerto 0

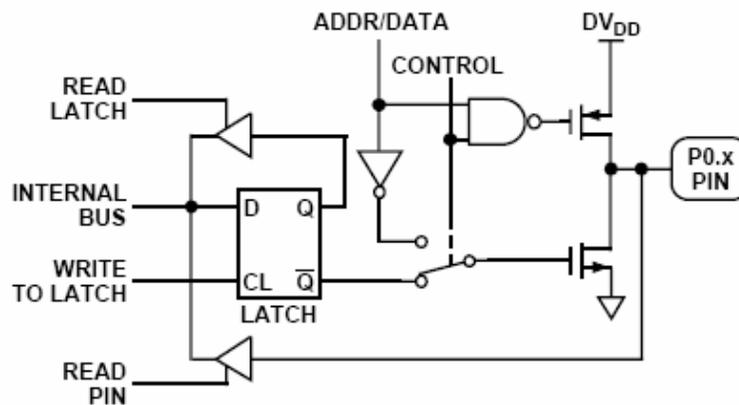
#### Arquitectura interna

El puerto 0 es un puerto bidireccional de 8 bits que es directamente controlado vía por el PORT0 del SFR. El puerto 0 es también un bus de orden inferior

multiplexado de la dirección de datos durante a accesos a memoria externa de programa o de datos.

En la figura se muestra un bit “latcheado” (un bit en los puertos SFR) y es representado como un flip-flop tipo D, el cual el reloj es un valor interno del bus en respuesta a escribir a la señal del cierre de la CPU. La salida Q del flip-flop es colocada en el bus interno en respuesta a la lectura de la señal del cierre de la CPU. Algunas instrucciones que leen un puerto activan la señal leída del cierre, y otras activan el pin leído de la señal.

**Figura del Puerto 0 del Bit Latch de entrada y salida del Buffer.**



Como se muestra en la figura anterior, la salida son conducidas de los pines del puerto 0 que son switchable (cambiables) para el bus interno ADDR y ADDE/DATA en el control interno de la señal para el uso de acceso externo en memoria. Durante el acceso a la memoria externa, el puerto 0 del SFR tiene un 1seg escrito, es decir, que todos los bis cerrados vuelven a ser 1. Cuando se accesa a la memoria externa, la señal de control en la figura anterior es alta, permitiendo la operación push-pull a la salida del pin desde la dirección interna o el bus de datos (línea ADDR/DATA).

El puerto en modo de propósito general I/O, los pines del puerto tiene 1seg para enviar al PORT 0 del SFR y configurarlos como colector abierto y además como flotante. En este estado, los pines del puerto 0 pueden ser usados como entradas de alta impedancia. Esto se representa en la figura por la compuerta NAND lo cual la salida siguen siendo alta mientras que la señal de control sea baja, de tal modo inhabilitando la parada del FET. Las resistencias externas de pull-up son requeridas cuando los pines del puerto 0 son usados a la salida como propósito general.

## Puerto 2

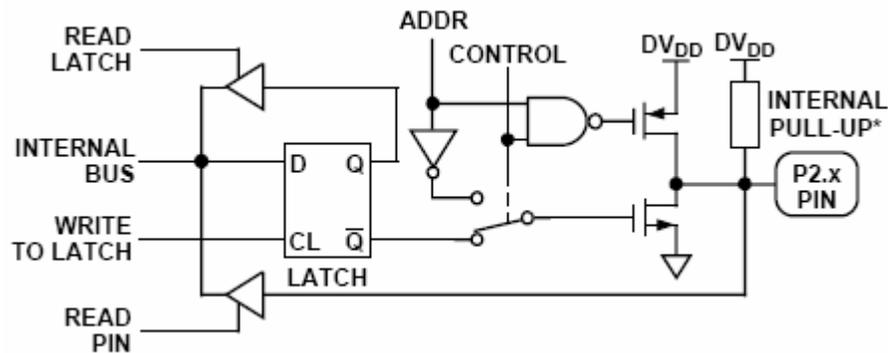
El puerto 2 es bidireccional con resistencias pull-up directamente controlado por el P2 del SFR. El puerto 2 también emite una orden alta al bit de dirección durante la búsqueda desde la memoria de programa externa, y una orden media-alta al bit de dirección durante el acceso a los 24 bit en el espacio de memoria de datos externa.

Como se muestra en la figura las salidas conducidas del puerto 2 son switchables a un bus interno de ADDR y ADDR/DATA por una señal interna de control para el uso en acceso a la memoria externa. En modo de direccionamiento externo de memoria (CONTROL = 1), los pines del puerto operan con la característica pull-up, controladas por la dirección interna del bus (Línea ADDR).

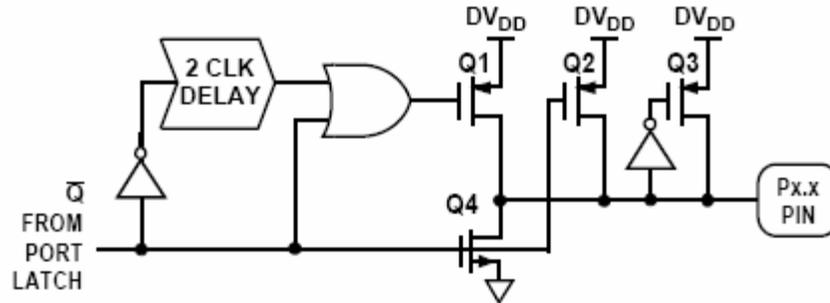
Sin embargo, a diferencia del P0 SFR durante accesos de memoria externos, el P2 SFR permanece sin cambiar. En el modo de propósito general los pines del puerto 2 como se muestra en la figura siguiente, en este estado, pueden ser usados como entradas.

P2.6 y P2.7 se pueden también utilizar como salidas de PWM. Cuando son seleccionadas como las salidas de PWM vía el CFG841/CFG842 SFR, Las salidas de PWM sobrescriben cualquier cosa escrita a P2.6 o a P2.7.

**Figura del Puerto 2 del Bit Latch de entrada y salida del Buffer.**



**Figura de Configuración interna Pull-Up.**



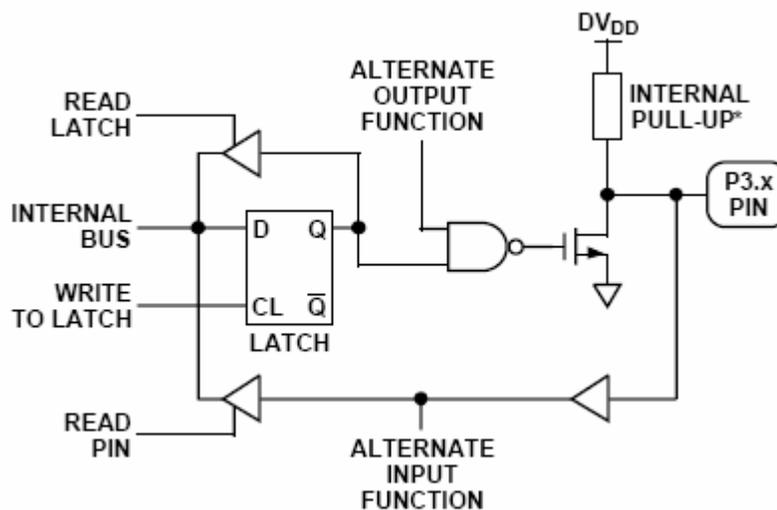
### Puerto 3

#### Arquitectura

El puerto 3 es bidireccional con pull-up interno, y directamente controlado por el P3 del SFR. Los pines del puerto 3 tienen 1seg escrito en ellos un altos por el pull-up interno, en este estado pueden usarse como entradas. Como entradas, los pines del puerto 3 son “pulled” bajo externamente por la fuente de corriente debido al pull-up interno.

Los pines del puerto 3 con 0's escritos en ellos conducirán un voltaje lógico bajo y capaces de trabajar 4mA.

**Figura del Puerto 3 del Bit Latch de entrada y salida del Buffer.**



## Funciones Alternas

El puerto 3 fija también tiene varias funciones secundarias según lo descrito en la tabla siguiente.

Las funciones alternas de los pines del puerto 3 se pueden activar solamente si el cierre correspondiente del bit en el P3 SFR contiene un 1. Si no, el pin del puerto se coloca en 0.

P3.3 y P3.4 se pueden también utilizar como salidas de PWM. Cuando son seleccionados como las salidas de PWM vía el CFG841/CFG842 SFR, las salidas de PWM sobrescriben cualquier cosa escrita a P3.4 o a P3.3.

**Tabla del Puerto 3 de las Funciones Alternas de los pines.**

Pin No.	Alternate Function
P3.0	RxD (UART Input Pin) (or Serial Data I/O in Mode 0)
P3.1	TxD (UART Output Pin) (or Serial Clock Output in Mode 0)
P3.2	$\overline{\text{INT0}}$ (External Interrupt 0)
P3.3	$\overline{\text{INT1}}$ (External Interrupt 1)/PWM 1/MISO
P3.4	T0 (Timer/Counter 0 External Input) PWM External Clock/PWM 0
P3.5	T1 (Timer/Counter 1 External Input)
P3.6	$\overline{\text{WR}}$ (External Data Memory Write Strobe)
P3.7	$\overline{\text{RD}}$ (External Data Memory Read Strobe)

## Timers/Contadores

El ADuC841 tiene tres Timer/Contador de 16 bits: Timer 0, Timer 1, y Timer 2. El Timer/Contador es un hardware incluido dentro del integrado del microcontrolador para ayudar al núcleo del procesador sin importar la ejecución de la funcionalidad del timer/contador en software. Cada timer/contador consiste en dos registros de 8 bits THx y TLx (x = 0, 1, y 2). Todos pueden ser configurados para operar como timers o como contadores.

### Modo Timer

En función de timer, el registro TLx se incrementa cada ciclo de máquina. Así, se puede pensar que como el contador de ciclos de máquina. Puesto que un ciclo de

maquina en un simple ciclo del núcleo consiste en un periodo de reloj del núcleo, la máxima tasa de conteo es la frecuencia de reloj.

### **Modo Contador**

En modo de de contador, el registro TLx se incrementa por transición de 1-0 en el pin externo correspondiente a la entrada: T0, T1, o T2. Cuando las muestras muestran un alto en un ciclo o un bajo en el siguiente ciclo, el contador es incrementado. Entonces toma dos ciclos de máquina, para organizar la transición de 1 hasta 0, la máxima tasa de conteo es media frecuencia de reloj del núcleo.

### **Registros**

El timer/contador posee los siguientes registros para la configuración y control de los modos de operación.

*TMOD, TCON* Control y configuración para los timers 0 y 1.

*T2CON* Control y configuración para el timer 2

Las características del registro TMOD se pueden apreciar en la tabla siguiente.

**TMOD** Modo de Registro del Timer/Counter 0 y 1

SFR Address	89H
Power-On Default	00H
Bit Direccionable	No

## Tabla de Designación de bit del Registro TMOD en el SFR.

Bit No.	Name	Description
7	Gate	Timer 1 Gating Control. Set by software to enable Timer/Counter 1 only while the $\overline{\text{INT1}}$ pin is high and the TR1 control bit is set. Cleared by software to enable Timer 1 whenever the TR1 control bit is set.
6	C/T	Timer 1 Timer or Counter Select Bit. Set by software to select counter operation (input from T1 pin). Cleared by software to select timer operation (input from internal system clock).
5	M1	Timer 1 Mode Select Bit 1 (Used with M0 Bit).
4	M0	Timer 1 Mode Select Bit 0. M1 M0 0 0 TH1 operates as an 8-bit timer/counter. TL1 serves as 5-bit prescaler. 0 1 16-Bit Timer/Counter. TH1 and TL1 are cascaded; there is no prescaler. 1 0 8-Bit Autoreload Timer/Counter. TH1 holds a value that is to be reloaded into TL1 each time it overflows. 1 1 Timer/Counter 1 Stopped.
3	Gate	Timer 0 Gating Control. Set by software to enable Timer/Counter 0 only while the $\overline{\text{INT0}}$ pin is high and the TR0 control bit is set. Cleared by software to enable Timer 0 whenever the TR0 control bit is set.
2	C/T	Timer 0 Timer or Counter Select Bit. Set by software to select counter operation (input from T0 pin). Cleared by software to select timer operation (input from internal system clock).
1	M1	Timer 0 Mode Select Bit 1.
0	M0	Timer 0 Mode Select Bit 0. M1 M0 0 0 TH0 operates as an 8-bit timer/counter. TL0 serves as a 5-bit prescaler. 0 1 16-Bit Timer/Counter. TH0 and TL0 are cascaded; there is no prescaler. 1 0 8-Bit Autoreload Timer/Counter. TH0 holds a value that is to be reloaded into TL0 each time it overflows. 1 1 TL0 is an 8-bit timer/counter controlled by the standard Timer 0 control bits. TH0 is an 8-bit timer only, controlled by Timer 1 control bits.

## TCON

### Timer/Counter 0 and 1

Registro de Control	
SFR Address	88H
Power-On Default	00H
Bit Direccionable	SI

## Tabla de Designación de bit del Registro TCON en el SFR.

Bit No.	Name	Description
7	TF1	Timer 1 Overflow Flag. Set by hardware on a Timer/Counter 1 overflow. Cleared by hardware when the program counter (PC) vectors to the interrupt service routine.
6	TR1	Timer 1 Run Control Bit. Set by the user to turn on Timer/Counter 1. Cleared by the user to turn off Timer/Counter 1.
5	TF0	Timer 0 Overflow Flag. Set by hardware on a Timer/Counter 0 overflow. Cleared by hardware when the PC vectors to the interrupt service routine.
4	TR0	Timer 0 Run Control Bit. Set by the user to turn on Timer/Counter 0. Cleared by the user to turn off Timer/Counter 0.
3	IE1 <sup>1</sup>	External Interrupt 1 ( $\overline{INT1}$ ) Flag. Set by hardware by a falling edge or by a zero level being applied to the external interrupt pin, $\overline{INT1}$ , depending on the state of Bit IT1. Cleared by hardware when the PC vectors to the interrupt service routine only if the interrupt was transition-activated. If level-activated, the external requesting source controls the request flag, rather than the on-chip hardware.
2	IT1 <sup>1</sup>	External Interrupt 1 (IE1) Trigger Type. Set by software to specify edge-sensitive detection, i.e., 1-to-0 transition. Cleared by software to specify level-sensitive detection, i.e., zero level.
1	IE0 <sup>1</sup>	External Interrupt 0 ( $\overline{INT0}$ ) Flag. Set by hardware by a falling edge or by a zero level being applied to external interrupt pin $\overline{INT0}$ , depending on the state of Bit IT0. Cleared by hardware when the PC vectors to the interrupt service routine only if the interrupt was transition-activated. If level-activated, the external requesting source controls the request flag, rather than the on-chip hardware.
0	IT0 <sup>1</sup>	External Interrupt 0 (IE0) Trigger Type. Set by software to specify edge-sensitive detection, i.e., 1-to-0 transition. Cleared by software to specify level-sensitive detection, i.e., zero level.

## Registros de Datos del Timer/Counter 0 y 1

Cada timer consiste en un registro de 8 bits. Estos pueden ser usados como registros independientes o registros combinados de 16 bits dependiendo del modo de configuración del timer.

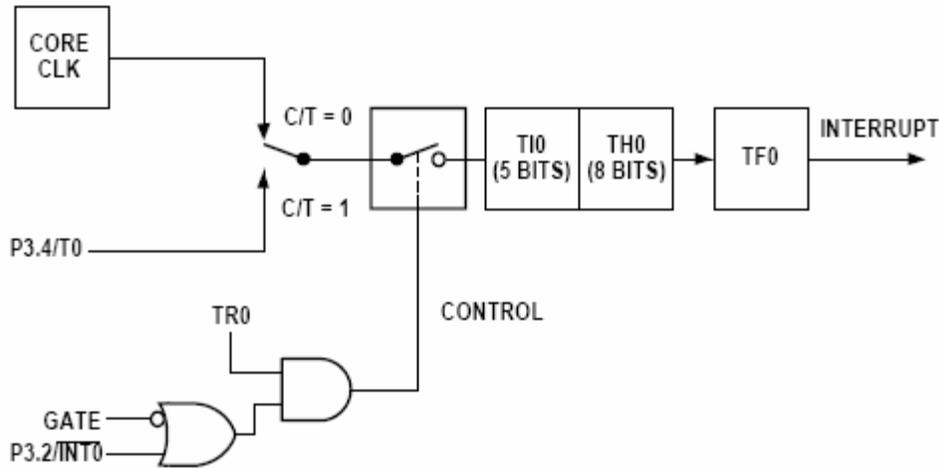
## Modos de operación Timer/Contador 0 y 1

En las siguientes secciones se describen los modos de operación del Timer/Contador 0 y 1. A menos que se indique lo contrario, se debe asumir que los modos de operación para los Timer 0 y 1 son los mismos.

### Modo cero: 13 bits

En modo cero se configura el timer/contador en 8 bits. La siguiente figura muestra el modo de operación 0.

**Figura del Modo 0 del Timer/Contador 0.**

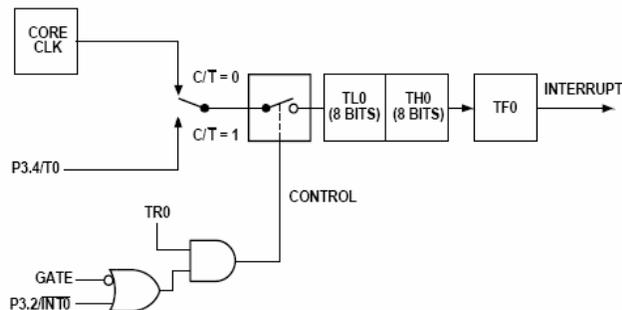


En este modo, el registro del Timer es configurado como un registro de 13 bits. TF0 puede ser usado para solicitar una interrupción. La entrada del contador es habilitada del Timer cuando TR0=1 y la otra compuerta es =0 o INT0 =1. Configurando la compuerta (Gate)=1 el timer siempre va ser controlado por la entrada de INT0 para facilitar la medida del ancho de pulso. El TR0 es el bit de control en el SFR del TCON. La compuerta (Gate) es en el TMOD. El registro de 13 bit consiste en los 8 bits del TH0 y cinco bits menos significativos del TL0. Los 3 bits más significativos del TL0 son indeterminados y deber ser ignorados.

**Modo uno: 16 bits**

El modo 1 es el mismo modo 2, excepto que en el modo1 el registro del Timer corre hasta con 16 bits. El modo 1 se muestra en la figura.

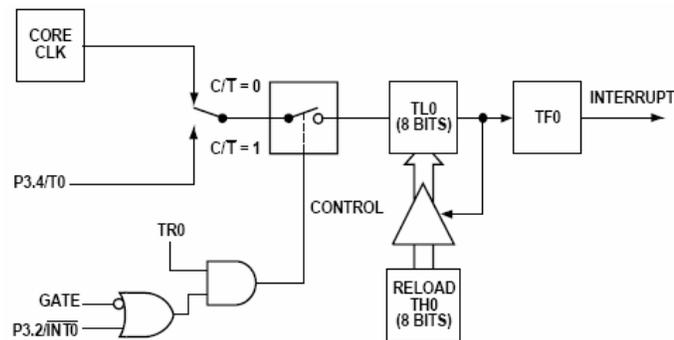
**Figura del Modo 1 del Timer/Contador 0.**



## Modo dos: 8 bits con Autorecarga

En el modo 2 se configura el registro del timer como un contador de 8 bits (TL0) con recarga automática, como se muestra en la siguiente figura. El desbordamiento de TL0 no solamente fija a TF0, también recarga a TL0 con el contenido de TH0, el cual es preestablecido por software. La recarga sale de TH0 sin cambios

Figura del Modo 2 del Timer/Contador 0.



## Timer/Contador 2

El Timer/Contador 2 tiene dos pares de registro de datos de 8 bits asociados con éste. Ambos son usados como registros de datos y como registros de captura/recarga del timer.

## Registros

El timer/Contador 2 posee las siguientes características como se muestran en la tabla:

T2CON Timer/Counter	2 Control Register
SFR Address	C8H
Power-On Default	00H
Bit Direccionable	SI

## Tabla de Designación de bit del Registro T2CON en el SFR.

Bit No.	Name	Description
7	TF2	Timer 2 Overflow Flag. Set by hardware on a Timer 2 overflow. TF2 cannot be set when either RCLK = 1 or TCLK = 1. Cleared by user software.
6	EXF2	Timer 2 External Flag. Set by hardware when either a capture or reload is caused by a negative transition on T2EX and EXEN2 = 1. Cleared by user software.
5	RCLK	Receive Clock Enable Bit. Set by the user to enable the serial port to use Timer 2 overflow pulses for its receive clock in serial port Modes 1 and 3. Cleared by the user to enable Timer 1 overflow to be used for the receive clock.
4	TCLK	Transmit Clock Enable Bit. Set by the user to enable the serial port to use Timer 2 overflow pulses for its transmit clock in serial port Modes 1 and 3. Cleared by the user to enable Timer 1 overflow to be used for the transmit clock.
3	EXEN2	Timer 2 External Enable Flag. Set by the user to enable a capture or reload to occur as a result of a negative transition on T2EX if Timer 2 is not being used to clock the serial port. Cleared by the user for Timer 2 to ignore events at T2EX.
2	TR2	Timer 2 Start/Stop Control Bit. Set by the user to start Timer 2. Cleared by the user to stop Timer 2.
1	CNT2	Timer 2 Timer or Counter Function Select Bit. Set by the user to select counter function (input from external T2 pin). Cleared by the user to select timer function (input from on-chip core clock).
0	CAP2	Timer 2 Capture/Reload Select Bit. Set by the user to enable captures on negative transitions at T2EX if EXEN2 = 1. Cleared by the user to enable autoreloads with Timer 2 overflows or negative transitions at T2EX when EXEN2 = 1. When either RCLK = 1 or TCLK = 1, this bit is ignored and the timer is forced to autoreload on Timer 2 overflow.

### Modos de operación Timer/Contador 2

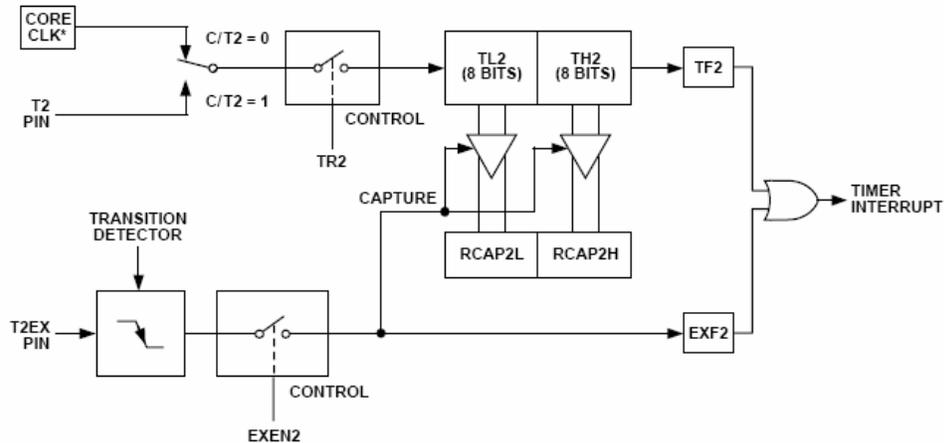
Los modos de operación del timer/contador 2 son:

- Modo cero: 16 bits con Autorecarga.
- Modo uno: 16 bits captura.

### Modo de Autorecarga de 16 bits Timer/Contador 2.

El modo de autorecarga tiene dos opciones que son seleccionadas por el bit de EXEN2 en T2CON. Si EXEN2=0 cuando el timer 2 está en rolls over, no solo pone en alto el TF2, causa también que el registro del timer 2 se recargue con un valor de 16 bits en el registro RCAP2L y RCAP2H, el cual es preconfigurado por software. Si EXEN2=1 entonces el timer 2 realiza la operación contraria, pero además con la característica de que la transición de 1 a 0 de la entrada externa de T2EX también accionara los 16 bits de recarga y configurar el EXF2.

## Modo uno: 16 bits Captura



El modo de captura también tiene dos opciones que son seleccionadas por el bit de EXEN2 en T2CON. Si EXEN2=0, entonces el Timer 2 es un temporizador de 16 bits o un contador que, hasta que se desborde, el bit TF2 se coloca en alto, el bit de desbordamiento del Timer2, podrá ser usado para generar una interrupción. Si EXEN2=1, entonces el Timer 2, realiza lo contrario, pero la transición de 1 a 0 de la entrada externa de T2EX causa la actualización del valor del registro en el Timer2, TL2 y TH2, capturando en los registros RCAP2L y RCAP2H respectivamente. Además la transición a T2EX causa que el bit de EXF2 en T2CON se coloque en alto. El modo de captura se muestra en la figura siguiente. El modo de generador de la tasa de baudio es seleccionado por RCLK=1 y/o TCLK=1.

En cualquier caso, si el Timer2 se está utilizando para generar la tasa de baudio, la bandera de interrupción TF2 no ocurrirá. Por lo tanto, las interrupciones del Timer 2 no ocurrirán, por tanto no tienen que ser deshabilitadas. En este modo, la bandera EXF2, sin embargo, puede todavía causar interrupciones, cuál se puede utilizar como tercera interrupción externa. La generación de la tasa de baudio se describe como parte del puerto serial del UART.

## Interfaz Serial UART

### Funcionamiento

El puerto serial es Full-dúplex, significa que puede transmitir y recibir simultáneamente. Significa que puede empezar a recibir un segundo byte antes de recibir previamente el byte que se ha leído en el registro de recepción. Sin embargo si el primer byte aun no sido leído para el momento en que la recepción del segundo byte es completada, el primer byte se pierde. La Interfaz física para la

conexión serial de datos es vía por el pin de RxD (P3.0) y TxD (P3.1), mientras que interfaz del SFR para la UART lo abarca SBUF y SCON como se describirá a continuación.

## SBUF

Ambos puertos el serial recibe y transmite los registros que son accedados por el SBUF del SFR (Dirección del SFR = 99H). Al escribir a SBUF carga el registro de transmisión, y la lectura de SBUF tiene acceso físicamente al recibir el registro separado.

### Tabla de Designación de bit del Registro SCON en el SFR.

Bit No.	Name	Description															
7	SM0	UART Serial Mode Select Bits.															
6	SM1	These bits select the serial port operating mode as follows: <table border="0" style="margin-left: 20px;"> <tr> <td>SM0</td> <td>SM1</td> <td>Selected Operating Mode.</td> </tr> <tr> <td>0</td> <td>0</td> <td>Mode 0: Shift Register, fixed baud rate (Core_Clk/2).</td> </tr> <tr> <td>0</td> <td>1</td> <td>Mode 1: 8-bit UART, variable baud rate.</td> </tr> <tr> <td>1</td> <td>0</td> <td>Mode 2: 9-bit UART, fixed baud rate (Core_Clk/32) or (Core_Clk/16).</td> </tr> <tr> <td>1</td> <td>1</td> <td>Mode 3: 9-bit UART, variable baud rate.</td> </tr> </table>	SM0	SM1	Selected Operating Mode.	0	0	Mode 0: Shift Register, fixed baud rate (Core_Clk/2).	0	1	Mode 1: 8-bit UART, variable baud rate.	1	0	Mode 2: 9-bit UART, fixed baud rate (Core_Clk/32) or (Core_Clk/16).	1	1	Mode 3: 9-bit UART, variable baud rate.
SM0	SM1	Selected Operating Mode.															
0	0	Mode 0: Shift Register, fixed baud rate (Core_Clk/2).															
0	1	Mode 1: 8-bit UART, variable baud rate.															
1	0	Mode 2: 9-bit UART, fixed baud rate (Core_Clk/32) or (Core_Clk/16).															
1	1	Mode 3: 9-bit UART, variable baud rate.															
5	SM2	Multiprocessor Communication Enable Bit. Enables multiprocessor communication in Modes 2 and 3. In Mode 0, SM2 must be cleared. In Mode 1, if SM2 is set, RI is not activated if a valid stop bit was not received. If SM2 is cleared, RI is set as soon as the byte of data has been received. In Modes 2 or 3, if SM2 is set, RI is not activated if the received 9th data bit in RB8 is 0. If SM2 is cleared, RI is set as soon as the byte of data has been received.															
4	REN	Serial Port Receive Enable Bit. Set by user software to enable serial port reception. Cleared by user software to disable serial port reception.															
3	TB8	Serial Port Transmit (Bit 9). The data loaded into TB8 is the 9th data bit transmitted in Modes 2 and 3.															
2	RB8	Serial Port Receiver Bit 9. The 9th data bit received in Modes 2 and 3 is latched into RB8. For Mode 1, the stop bit is latched into RB8.															
1	TI	Serial Port Transmit Interrupt Flag. Set by hardware at the end of the 8th bit in Mode 0, or at the beginning of the stop bit in Modes 1, 2, and 3. TI must be cleared by user software.															
0	RI	Serial Port Receive Interrupt Flag. Set by hardware at the end of the 8th bit in Mode 0, or halfway through the stop bit in Modes 1, 2, and 3. RI must be cleared by software.															

## Modos de operación

### Modo cero: Modo registro de corrimiento de 8 bits

El modo 0 es seleccionado para clarear ambos bits SM0 y SM1 en el SCON del SFR. La entrada serial de datos y la salida es por el RxD. La salida TxD cambia de reloj.

Cada bit de dato es transmitido o recibido. La transmisión es iniciada por cualquier instrucción que se escriba a SBUF. Se cambian de puesto los datos a la salida de la línea de RxD. Los 8 bits son transmitidos con el bit menos significativo (LSB) primero.

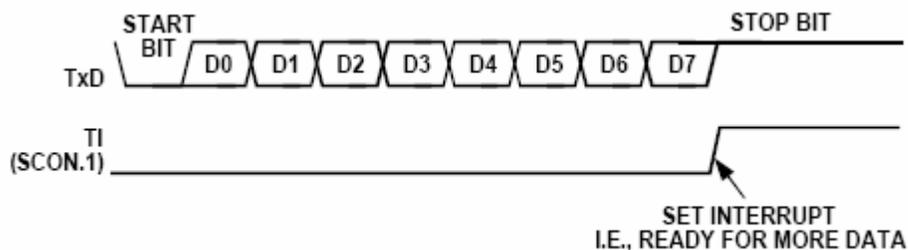
La recepción es iniciada cuando la recepción, el bit habilitado (REN) es 1 y recibe la interrupción del bit (RI) es cero. Cuando (RI) es clareado los datos se registran en la línea de RxD y los pulsos de reloj son la salida de la línea de TxD.

### Modo uno: UART 8 bit, razón de baudios variable

El modo 1 es seleccionado clareando SM0 y colocando en alto SM1. Cada bit de dato (Primer LSB) es precedido por el bit de inicio (0) y seguido por el bit de parada (1). Además 10 bits son transmitidos en TxD o son recibidos en RxD. La razón de baudio es puesta en alto por el Timer 1 o por el desbordamiento del Timer 2, o de la combinación de los dos (uno por transmisión y el otro por recepción).

La transmisión es iniciada escribiendo al SBUF. Al escribir al SBUF la señal carga un 1 (bit de parada) dentro de los 9 bits de posición de la transmisión de cambio de registro. LA salida del dato bit a bit es hasta que el bit de parada aparece en TxD y la transmisión hace una bandera de interrupción (TI) que es automáticamente puesta en alto como se muestra en la siguiente figura.

Figura del Puerto de Transmisión Serial UART modo 1.



La recepción es iniciada cuando la transmisión de 1-0 es detectada en RxD. Asumiendo que el bit de inicio es detectado, el carácter de recepción continua. El bit de inicio se salta y los 8 bits de datos son puestos en el registro de cambio del puerto serial. Cuando los 8 bits están dentro del registro, ocurren los siguientes eventos:

- Los 8 bits en la recepción de cambio de registro son latched dentro del SBUF.
- Los 9 bits están dentro del registro del RB8 en SCON.
- La bandera de interrupción de recepción (RI) está en 1.

En este caso, si y solo si, todas las siguientes condiciones son realizadas cuando se genera el pulso final de cambio:

- RI = 0
- Tampoco SM2 = 0 SM2 = 1
- El bit de parada de recepción es = 1

Si cualquiera de estas condiciones no se realiza, la estructura recibida es una pérdida irreversible, y RI no es puesto en uno (1).

### **Modo dos: UART 9 bit, razón de baudios fija**

El modo 2 es seleccionado colocando en alto SM0 y clareando SM1. En este modo la UART opera en modo de 9 bits con una razón de baudio fija. La razón de baudio es fija en el núcleo Clk/32 por defecto, aunque fijando en 1 el bit de SMOD en PCON, la frecuencia puede ser doblada. Once bits son transmitidos o recibidos. Un bit de inicio (0), 8 bits de datos, un bit programable (9th bit) y un bit de parada (1). El bit programable es el más usado como bit de parada, aunque puede ser usado para cualquier otra cosa, incluso como bit de dato si es requerido.

Para transmitir, los 8 bits de datos deben ser escritos en SBUF. El 9th bit debe ser escrito TB8 en SCON. Cuando la transmisión es iniciada, los 8 bits de datos son cargados en la transmisión de cambio de registro. El contenido de TB8 es cargado en el 9th bit de posición de la transmisión de cambio de registro. La transmisión empieza a la siguiente razón de baudio válida. La bandera TI es puesta en alto tan pronto el bit de parada aparece en TxD.

La recepción en el modo 2 es similar al modo 1. Los 8 bits de datos son entradas de la RxD y cargados dentro de la recepción de registro de cambio. Cuando los 8 bits están dentro del registro ocurren los siguientes eventos:

- Los 8 bits en la recepción en el registro de cambio son latched dentro de SBUF.
- El 9th bit de dato es latched dentro de RB8 en SCON.
- La bandera de interrupción de recepción (RI) es alto (1).

En este caso, si y solo si, todas las siguientes condiciones son realizadas cuando se genera el pulso final de cambio:

- RI = 0
- Tampoco SM2 = 0 SM2 = 1
- El bit de parada de recepción es = 1

Si cualquiera de estas condiciones no se realiza, la estructura recibida es una pérdida irreversible, y RI no es puesto en uno (1).

### **Modo tres: UART 9 bit, razón de baudios variable**

El modo 3 es seleccionado configurando ambos SM0 y SM1 en uno. En este modo el puerto serial 8051 UART opera en modo de 9 bits con una razón de baudio variable determinada por el Timer 1 o el Timer 2. La operación de la UART de 9 bits es la misma que en el modo 2, pero la razón de baudio puede variar como en el modo 1. En todos los modos, la transmisión para todos los modos, es iniciada por cualquier instrucción que usa SBUF como registro de destinación.

### **Generación de Razón de Baudios del Puerto Serial UART**

#### **Formula de Generación de razón de baudios modo cero**

$$\text{Mode 0 Baud Rate} = (\text{Core Clock Frequency}/12)$$

La razón de baudio en modo cero es fija.

#### **Fórmula de Generación de razón de baudios modo dos**

$$\text{Mode 2 Baud Rate} = (2^{\text{SMOD}}/32 \times [\text{Core Clock Frequency}])$$

La razón de baudio en modo 2 depende del valor del bit SMOD en el PCON del SFR. Si SMOD = 0, la razón de baudio es 1/32 del reloj del núcleo. Si SMOD = 1 la razón de baudio es 1/16 del reloj del núcleo.

#### **Generación de razón de baudios modos uno y tres**

La razón de baudio en modos 1 y 3 son determinadas por el desbordamiento en Timer 1 o en Timer 2, o en ambos.

#### **Fórmula de Generación de razón de baudios Timer uno**

$$\text{Modes 1 and 3 Baud Rate} = (2^{\text{SMOD}}/32 \times (\text{Timer 1 Overflow Rate}))$$

Cuando el Timer 1 es usado como generador de razón de baudio, la razón de baudio en modo 1 y 3 son determinadas por la razón de desbordamiento del Timer 1 y el valor de SMOD.

La interrupción del Timer 1 deberá ser deshabilitada en esta aplicación. El Timer mismo puede ser configurado por otro Timer o una operación de contador y en cualquiera de sus tres modos correrá. En la aplicación más típica, éste es configurado en modo de operación de Timer en modo de autorecarga. En este caso, la razón de baudio está dada por la siguiente fórmula:

**Fórmula de Generación de razón de baudios Timer 1 en modo de Autorecarga.**

$$\text{Modes 1 and 3 Baud Rate} = (2^{\text{SMOD}}/32) \times (\text{Core Clock} / [256 - \text{TH1}])$$

**Fórmula de Generación de razón de baudios Timer 2**

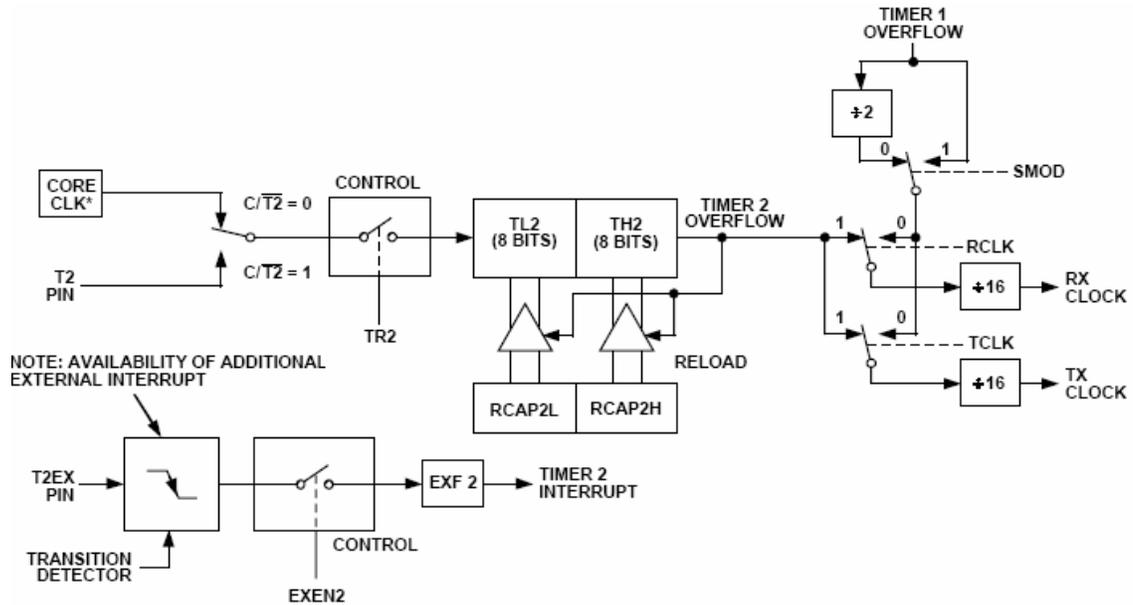
$$\text{Modes 1 and 2 Baud Rate} = (1/16) \times (\text{Timer 2 Overflow Rate})$$

La razón de baudio también puede ser generada usando el Timer 2. Usando el Timer 2 es similar que usando el Timer 1 en donde el Timer deberá desbordarse 16 veces antes de que un bit sea transmitido o recibido. Esto debido a que el Timer 2 tiene un modo de autorecarga de 16 bits, un rango mas amplio de razón de baudio es posible usando el Timer 2.

Además, cuando el Timer 2 es usado para generar razón de baudios, el Timer incrementa siempre dos ciclos de reloj, más que un ciclo de maquina como anteriormente. Además este incremento es seis veces más rápido que con el Timer 1, y la razón de baudio es seis veces más rápida.

El Timer 2 es seleccionado como generador de razón de baudio colocando en alto TCLK y/o RCLK en T2CON. Para transmitir y recibir la razón de baudio puede ser simultáneamente diferente. Fijando RCLK y/o TCLK colocando el Timer 2 como modo de generador de baudios como se muestra en la siguiente figura.

## Figura del Timer 2 Razón de Baudios de la UART

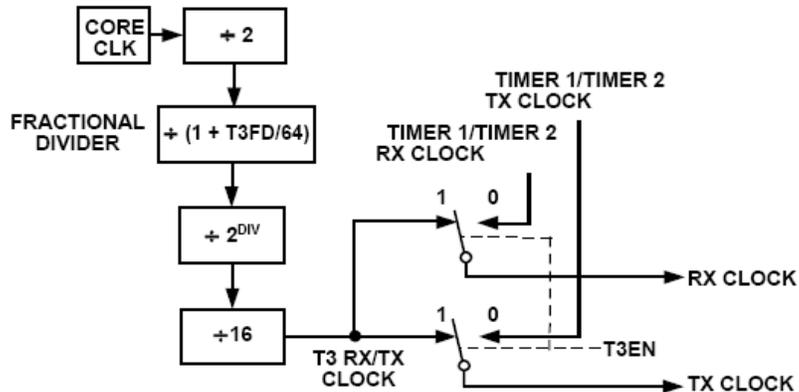


## Generación de razón de baudios Timer tres

Las divisiones de números enteros altos en un bloque de la UART significan que la razón de baudio a alta velocidad no siempre es posible usando cristales particulares.

El Timer 3 puede ser usado en vez del Timer 1 o Timer 2 para generar con exactitud altas velocidades de razón de baudio en la UART incluyendo a 115200 y 230400. El Timer 3 también permite una gama amplia de razón de baudio que puede ser obtenida. De hecho, cada razón de bit deseada desde los 12 bit/s a 393216 bit/s puede ser generada dentro de un error de  $\pm 0.8\%$ . El Timer 3 libera también otros tres Timers, permitiendo que sean utilizados para diferentes usos. Un diagrama de de bloques del Timer 3 se muestra en la siguiente figura.

**Figura del Timer 3, Razón de Baudios UART.**



Dos SFRs (T3CON y T3FD) son usados para el control del Timer 3. El T3CON es el control de la razón de baudio SFR, permitiendo que el Timer 3 pueda ser usado para configurar la razón de baudio de la UART, y el divisor binario (DIV).

El valor apropiado para escribir en el DIV2-1-0 bits puede ser calculado usando la siguiente fórmula  $f_{CORE}$  que esta definida por el PLLCON de SFR. Note que el valor del DIV debe ser redondeado al valor de abajo.

**Fórmula del Divisor Binario.**

$$DIV = \frac{\log\left(\frac{f_{CORE}}{16 \times \text{Baud Rate}}\right)}{\log(2)}$$

El T3FD es el divisor de radio fraccional requerido para alcanzar la razón de baudio requerida. El valor apropiado para T3FD puede ser calculado con la siguiente fórmula:

**Fórmula del Divisor de Radio Fraccional.**

$$T3FD = \frac{2 \times f_{CORE}}{2^{DIV-1} \times \text{Baud Rate}} - 64$$

Note que T3FD deberá ser redondeado al entero más cercano. Una vez que el valor para DIV y el T3FD son calculados, la razón de baudio actual puede ser calculada con la siguiente fórmula:

**Fórmula de Razón de Baudios Actual.**

$$Actual\ Baud\ Rate = \frac{2 \times f_{CORE}}{2^{DIV-1} \times (T3FD + 64)}$$

**Tabla de Designación de bit del Registro T3CON en el SFR.**

Bit No.	Name	Description
7	T3BAUDEN	T3UARTBAUD Enable. Set to enable Timer 3 to generate the baud rate. When set, PCON.7, T2CON.4, and T2CON.5 are ignored. Cleared to let the baud rate be generated as per a standard 8052.
6		Reserved.
5		Reserved.
4		Reserved.
3		Reserved.
2	DIV2	Binary Divider Factor.
1	DIV1	DIV2      DIV1      DIV0      Bin Divider
0	DIV0	0      0      0      1
		0      0      1      1
		0      1      0      1
		0      1      1      1
		1      0      0      1
		1      0      1      1
		1      1      0      1
		1      1      1      1

**Tabla Razón de Baudio comúnmente usada con el Timer 3 con 16.777216Mhz de reloj en el PLL.**

Ideal Baud	CD	DIV	T3CON	T3FD	% Error
230400	0	2	82H	09H	0.25
115200	0	3	83H	09H	0.25
115200	1	2	82H	09H	0.25
115200	2	1	81H	09H	0.25
57600	0	4	84H	09H	0.25
57600	1	3	83H	09H	0.25
57600	2	2	82H	09H	0.25
57600	3	1	81H	09H	0.25
38400	0	4	84H	2DH	0.2
38400	1	3	83H	2DH	0.2
38400	2	2	82H	2DH	0.2
38400	3	1	81H	2DH	0.2
19200	0	5	85H	2DH	0.2
19200	1	4	84H	2DH	0.2
19200	2	3	83H	2DH	0.2
19200	3	2	82H	2DH	0.2
19200	4	1	81H	2DH	0.2
9600	0	6	86H	2DH	0.2
9600	1	5	85H	2DH	0.2
9600	2	4	84H	2DH	0.2
9600	3	3	83H	2DH	0.2
9600	4	2	82H	2DH	0.2
9600	5	1	81H	2DH	0.2

### **C. Driver/Firmware Virtual Com Port (VCP)**

El dispositivo de Texas Instruments (TI) TUSB3410 puede ser usado para implementaciones flexibles de full-speed en aplicaciones de puente USB/Serial.

TI provee un driver/firmware como solución a las aplicaciones USB/Serial usando TUSB3410 y el software VCP, ambos en conjunto, proveen una llave de solución de software para aplicaciones de puente USB/Serial:

- USB/RS232
- USB/RS485
- USB/UART (Para interfaz de un Microcontrolador)
- USB/IrDA

Cuando diseñamos una solución basada en USB/Serial con el TUSB3410, se deben tener en cuenta los siguientes pasos:

1. Obtener el VID del vendedor y el PID del producto.
2. Determinar la localización del firmware (En el PC o en el dispositivo EEPROM).
3. Configurar y programar la EEPROM.
4. Configurar el instalador.

Para el cumplimiento USB, todo producto debe tener un ID de vendedor y de producto. El VID y PID para el entrenador digital fueron suministrados por TI, ya que fueron dados por la compra del KIT TUSB3410.

El Firmware en la aplicación del TUSB3410 puede ser almacenado en el host USB del PC o en la memoria EEPROM.

Cuando el firmware es almacenado en la memoria EEPROM, 28Kbit deberán ser suficiente para cualquier aplicación, El firmware es de aproximadamente de 14K, dejando aproximadamente 2k para la información del descriptor header.

Luego se procede a la instalación de driver del TUSB3410, el cual instala los archivos necesarios para la utilización del dispositivo.

#### **D. Tabla de Excel – Función de Transferencia ADC**

Ver archivo de Excel anexo en el CD.

#### **E. Tabla de Excel PWM COUNTER**

Ver archivo de Excel anexo en el CD.

## F. Guía de Laboratorio Práctica “Demo”

En la siguiente guía de laboratorio el estudiante empezará a entender y a familiarizarse con el entrenador digital USB, para que en las futuras prácticas de laboratorio que vaya a desarrollar, el estudiante sea capaz de sacar el máximo provecho al entrenador y pueda realizar sus aplicaciones.

### 1. Manejo del Módulo ADC DMA

- a. Llevar previamente al laboratorio en la protoboard un circuito generador de señales como son: Onda seno, Onda triangular, Onda diente de sierra con un nivel DC preestablecido (1V) y la amplitud de la señal 1Vpp.
- b. Iniciar el PC de trabajo, conectar el entrenador digital USB para iniciar el funcionamiento en conjunto.
- c. Abrir el programa de interfaz del entrenador digital USB en el computador.
- d. Conectar las diferentes señales a las borneras del entrenador digital USB con los respectivos canales que el estudiante desee trabajar.
- e. En el menú principal del interfaz, seleccionar en el menú desplegable, ADC DMA, y luego seleccionamos el puerto serie (COM1, COM2, COM3, COM4...) según el puerto que haya escogido el PC para esta aplicación. Ver en administrador de dispositivos en la sección *Ports (COM %LPT)* y se ejecuta la VI.
- f. Al ejecutarse la VI principal, se abre una nueva VI, el Módulo de ADC DMA, donde el estudiante seleccionará los parámetros para el tratamiento de las señales que el desee trabajar.
- g. Ingresar los comandos a la VI del módulo: La cantidad de canales a muestrear, la frecuencia de muestreo, el “**MCLK Divider**” que permite seleccionar los periodos de muestreo disponibles y el “**Nº ADC Clock**” que permite seleccionar el número de tiempos de adquisición para la conversión de la señal de entrada.
- h. Ejecute la VI.
- i. El entrenador digital USB recibe los comandos de la VI para proceder con la conversión ADC DMA, luego de realizarla, la envía de nuevo a la VI para mostrar los datos en las graficas respectivas y cierra la VI.

- j. El entrenador digital USB esperara por nuevos comandos para un nuevo módulo.

## 2. Manejo del Módulo DAC

- a. Retornamos al menú principal del entrenador digital USB.
- b. En el menú principal del interfaz, seleccionar en el menú desplegable, DAC, luego seleccionamos el puerto serie (COM1, COM2, COM3, COM4...) Ver en administrador de dispositivos en la sección *Ports (COM %LPT)* el puerto según haya escogido el PC para esta aplicación; y luego iniciamos el modulo DAC oprimiendo el botón **START**, para así finalmente ejecutar la VI.
- c. Al ejecutarse la VI principal, se abre una nueva VI, el Módulo DAC, donde el estudiante seleccionará los parámetros de configuración.
- d. Seleccionar en el menú desplegable "**Selección Ondas**" el par de señales a trabajar, para esté laboratorio se mostrará la señal seno y la señal triangular. Luego seleccionará en el menú desplegable "**Frecuencia Salida DAC**" el valor de la frecuencia a la cual van salir las señales (610Hz).
- e. Ejecute la VI.
- f. El entrenador digital USB recibe los comandos de la VI para proceder con la conversión DAC, luego de realizarla, se envían los datos a las borneras del módulo DAC del entrenador digital USB (DAC0 y DAC1).
- g. Con la ayuda de un osciloscopio o un "Fluke" observe y mide para comprobar la salida de las señales con el respectivo valor de frecuencia y voltaje de salida.
- h. Detener el módulo DAC, retorne al menú principal del entrenador digital USB, y proceda a oprimir el botón de **STOP**. Esto permitirá trabajar en los demás módulos, ya que cuando se trabaja con el DAC, los demás módulos quedan inhabilitados.
- i. El entrenador digital USB esperará por nuevos comandos para un nuevo módulo.

### 3. Manejo del Módulo Contador de Eventos

- a. Retornamos al menú principal del entrenador digital USB.
- b. Conectar en la Bornera ADC0 el “evento” que se desee contar. Para el desarrollo de esta práctica se utilizará un PWM con un periodo de un 1seg. (El estudiante es libre de implementar el sensor de conteo de eventos el cual debe entregar a su salida un nivel alto o bajo en lógica TTL.). Conecte un juego de leds en ambos puertos P0 y P2 para la visualización del contador de Eventos. (Recuerde que el puerto P0 es de colector abierto, “lógica inversa”).
- c. En el menú principal del interfaz, seleccionar en el menú desplegable, Contador de Eventos, luego seleccionamos el puerto serie (COM1, COM2, COM3, COM4...) Ver en administrador de dispositivos en la sección *Ports (COM %LPT)* el puerto según haya escogido el PC para esta aplicación; y luego iniciamos el modulo de Contador oprimiendo el botón **START**, para así finalmente ejecutar la VI.
- d. Al ejecutarse la VI principal, se abre una nueva VI, el Módulo Contador, donde el estudiante seleccionará los parámetros de configuración.
- e. Seleccionar en el menú desplegable “**Preescaler**” el modo de conteo (1).
- f. Ejecute la VI.
- g. El entrenador digital USB visualiza el conteo en los puertos del entrenador digital USB donde el P2 mostrará el dato de conteo de los 8 bits menos significativos y el P0 mostrará los 8 bits más significativos.
- h. Detener el módulo Contador de Eventos (16bits), se retorna al menú principal del entrenador digital USB, y se procederá a oprimir el botón de **STOP**. El módulo también se detendrá si el contador llega a su límite (65535), es decir cuando todos los leds de ambos puertos estén encendidos, si este caso ocurre, retorne al menú principal y oprima el botón **STOP**. Esto permitirá seguir trabajando en los demás módulos.
- i. El entrenador digital USB esperará por nuevos comandos para un nuevo módulo.

#### 4. Manejo del Módulo Puerto de Salida 8bits

- a. Retornamos al menú principal del entrenador digital USB.
- b. Conecte un juego de leds en ambos puertos P0 y P2 para la visualización de la cantidad de datos que se va a trabajar. (Recuerde que el puerto P0 es de colector abierto, es decir lógica inversa).
- c. En el menú principal del interfaz, seleccionar en el menú desplegable, Escritura al Puerto 8Bits, y luego seleccionamos el puerto serie (COM1, COM2, COM3, COM4...) según el puerto que haya escogido el PC para esta aplicación. Ver en administrador de dispositivos en la sección *Ports (COM %LPT)* y se ejecuta la VI.
- d. Al ejecutarse la VI principal, se abre una nueva VI, el Módulo Puerto Salida 8bits, donde el estudiante seleccionará los parámetros de configuración.
- e. Seleccionar en el control numérico “**Cantidad Datos**” la cantidad de datos (3) para ser enviados al puerto I/O seleccionado. (Recuerde que la cantidad máxima de datos a ser enviados es de 128 bytes).
- f. Seleccionar en el menú desplegable “**Retardo**” el tiempo de retardo de 1seg para visualizar los datos en el puerto I/O seleccionado.
- g. Seleccionar en el menú desplegable “**Puerto Salida**” el puerto P0, para visualizar los datos en este puerto.
- h. Ingrese en la lista de control “**Datos (Decimal)**” (En notación decimal) el 1, 3, 15,63 para ser visualizado en el puerto seleccionado.
- i. Ejecute la VI.
- j. Se Visualiza los datos en el puerto seleccionado con el retardo de 1seg.
- k. Repetir los pasos del numeral (e) hasta el numeral (i), seleccionando el puerto P2 y con un retardo de 0.5seg.
- l. Retorne al menú principal y el entrenador digital USB esperará por nuevos comandos para un nuevo módulo.

## 5. Manejo del Módulo Puerto de Salida 16bits

- a. Retornamos al menú principal del entrenador digital USB.
- b. Conecte un juego de leds en ambos puertos P0 y P2 para la visualización de los datos que se va a trabajar. (Recuerde que el puerto P0 es de colector abierto).
- c. En el menú principal del interfaz, seleccionar en el menú desplegable, Escritura al Puerto 16Bits, y luego seleccionamos el puerto serie (COM1, COM2, COM3, COM4...) según el puerto que haya escogido el PC para esta aplicación. Ver en administrador de dispositivos en la sección *Ports (COM %LPT)* y se ejecuta la VI.
- d. Al ejecutarse la VI principal, se abre una nueva VI, el Módulo Puerto Salida 16bits, donde el estudiante seleccionará los parámetros de configuración.

El entrenador digital USB visualiza el conteo en los puertos del entrenador digital USB donde el P2 mostrará el dato de conteo de los 8 bits menos significativos y el P0 mostrará los 8 bits más significativos.

- e. Seleccionar en el control numérico “**Cantidad Datos**” la cantidad de datos (3) para ser enviados al puerto I/O seleccionado. (Recuerde que la cantidad máxima de datos a ser enviados es de 64 bytes).
- f. Seleccionar en el menú desplegable “**Retardo**” el tiempo de retardo de 1seg para visualizar los datos en el puerto I/O seleccionado.
- g. Ingrese en la lista de control “**Words (Decimal)**” (En notación decimal) el 1, 3, 15, 63, 127, 255, para ser visualizado los puertos.
- h. Ejecute la VI.
- i. Se visualiza los datos en los puertos con el retardo de 1seg.
- j. Retorne al menú principal y el entrenador digital USB esperará por nuevos comandos para un nuevo módulo.

## 6. Manejo del Módulo Lectura del Puerto

- a. Retornamos al menú principal del entrenador digital USB.
- b. Conecte un DIP SWITCH en ambos puertos P0 y P2 para la escritura de los datos que se va a trabajar. (Recuerde que el puerto P0 es de colector abierto).
- c. En el menú principal del interfaz, seleccionar en el menú desplegable, Escritura al Puerto 16Bits, y luego seleccionamos el puerto serie (COM1, COM2, COM3, COM4...) según el puerto que haya escogido el PC para esta aplicación. Ver en administrador de dispositivos en la sección *Ports (COM %LPT)* y se ejecuta la VI.
- d. Al ejecutarse la VI principal, se abre una nueva VI, el Módulo de Lectura del Puerto “Bits Puerto Seleccionado”, donde el estudiante seleccionará los parámetros de configuración.

Ingrese el dato por medio del dip swicht la siguiente palabra: 11001100.

- e. Seleccione en el menú desplegable “**Seleccione Puerto**” el Puerto P2.
- f. Ejecute la VI.
- g. Repita los pasos del numeral (d) hasta el (f), con la palabra 11111111 y el puerto P0.
- h. Retorne al menú principal y el entrenador digital USB esperará por nuevos comandos para un nuevo módulo.

## 7. Manejo del Módulo PWM MODO 1

- a. Retornamos al menú principal del entrenador digital USB.
- b. En el menú principal del interfaz, seleccionar en el menú desplegable, PWM Modo 1, y luego seleccionamos el puerto serie (COM1, COM2, COM3, COM4...) según el puerto que haya escogido el PC para esta aplicación. Ver en administrador de dispositivos en la sección *Ports (COM %LPT)* y se ejecuta la VI.

- c. Al ejecutarse la VI principal, se abre una nueva VI, el Módulo de PWM Modo 1, donde el estudiante seleccionará los parámetros de configuración.
- d. Seleccione en el control numérico “**Periodo PWM0**” un periodo de 0.0001429seg (7000HZ).
- e. Seleccione en el control numérico “**Duty Cycle PWM0**” un ciclo de trabajo de 0.0000714seg (14000Hz).

Observar la tabla guía, mostrada al lado derecho de la interfaz gráfica de Labview.

- f. Seleccione en el menú desplegable “**PREESCALER**” 1.
- g. Seleccione en el menú desplegable “**DIVISOR DE RELOJ PWM**” Fosc (Opción de rango de frecuencia alcanzado por el entrenador).
- h. Ejecutar la VI.

En los indicadores numéricos “**Frecuencia PWM0**” y “**Frecuencia Duty Cycle**” el usuario visualiza el valor de la frecuencia, luego de haberse ajustado al rango de frecuencias más cercano que el modo de PWM puede alcanzar.

- i. Mediante un osciloscopio o un “Fluke” compruebe el valor de la señal PWM generada por el entrenador digital USB.
- j. Retorne al menú principal y el entrenador digital USB esperará por nuevos comandos para un nuevo módulo.

## 8. Manejo del Módulo PWM MODO 2

- a. Retornamos al menú principal del entrenador digital USB.
- b. En el menú principal del interfaz, seleccionar en el menú desplegable, PWM Modo 2, y luego seleccionamos el puerto serie (COM1, COM2, COM3, COM4...) según el puerto que haya escogido el PC para esta aplicación. Ver en administrador de dispositivos en la sección *Ports (COM %LPT)* y se ejecuta la VI.
- c. Al ejecutarse la VI principal, se abre una nueva VI, el Módulo de PWM Modo 2, donde el estudiante seleccionará los parámetros de configuración.

- d. Seleccione en el control numérico “**Periodo PWM0 y PWM1**” un periodo de 0.0002049seg (4880HZ) para los dos PWM's.
- e. Seleccione en el control numérico “**Duty Cycle PWM0**” un ciclo de trabajo de 0.0001025seg (9760Hz).
- f. Seleccione en el control numérico “**Start Time Duty Cycle PWM1**” un inicio de 0.0002seg (5000HZ) de ciclo de trabajo.
- g. Seleccione en el control numérico “**End Duty Cycle PWM1**” un final de 0.0000667seg (15000HZ) de ciclo de trabajo.
- h. Seleccione en el menú desplegable “**PREESCALER**” 4.
- i. Seleccione en el menú desplegable “**DIVISOR DE RELOJ PWM**” Fosc/1024 (Opción de rango de frecuencia alcanzado por el entrenador).

Observar la tabla guía, mostrada al lado derecho de la interfaz gráfica de Labview.

- j. Ejecute la VI.

En los indicadores numéricos “**Frecuencia PWM0 y PWM1**”, “**Frecuencia Duty Cycle PWM0**” “**Frec. Start Time Duty Cycle PWM1**” y “**Frec. End Duty Cycle PWM1**” el usuario visualiza el valor de la frecuencia, luego de haberse ajustado al rango de frecuencias más cercano que el modo de PWM puede alcanzar.

- k. Mediante un osciloscopio o un “Fluke” compruebe el valor de la señal PWM generada por el entrenador digital USB.
- l. Retorne al menú principal y el entrenador digital USB esperará por nuevos comandos para un nuevo módulo.

## 9. Manejo del Módulo PWM MODO 3

- a. Retornamos al menú principal del entrenador digital USB.
- b. En el menú principal del interfaz, seleccionar en el menú desplegable, PWM Modo 3, y luego seleccionamos el puerto serie (COM1, COM2, COM3,

COM4...) según el puerto que haya escogido el PC para esta aplicación. Ver en administrador de dispositivos en la sección *Ports (COM %LPT)* y se ejecuta la VI.

- c. Al ejecutarse la VI principal, se abre una nueva VI, el Módulo de PWM Modo 3, donde el estudiante seleccionará los parámetros de configuración.
- d. Seleccione en el control numérico **“Duty Cycle PWM1”** un ciclo de trabajo de 0.0016388seg (610.2HZ).
- e. Seleccione en el control numérico **“Duty Cycle PWM0”** un ciclo de trabajo de 0.0010925seg (915.3Hz).

Observar la tabla guía, mostrada al lado derecho de la interfaz gráfica de Labview.

- f. Seleccione en el menú desplegable **“PREESCALER”** 1.
- g. Seleccione en el menú desplegable **“DIVISOR DE RELOJ PWM”** Fosc (Opción de rango de frecuencia alcanzado por el entrenador).
- h. Ejecutar la VI.

En los indicadores numéricos **“Frec. Duty Cycle PWM1”** y **“Frec. Duty Cycle PWM0”** el usuario visualiza el valor de la frecuencia, luego de haberse ajustado al rango de frecuencias más cercano que el modo de PWM puede alcanzar.

- i. Mediante un osciloscopio o un “Fluke” compruebe el valor de la señal PWM generada por el entrenador digital USB.
- j. Retorne al menú principal y el entrenador digital USB esperará por nuevos comandos para un nuevo módulo.

## 10. Manejo del Módulo PWM MODO 4

- a. Retornamos al menú principal del entrenador digital USB.
- b. En el menú principal del interfaz, seleccionar en el menú desplegable, PWM Modo 4, y luego seleccionamos el puerto serie (COM1, COM2, COM3, COM4...) según el puerto que haya escogido el PC para esta aplicación.

Ver en administrador de dispositivos en la sección *Ports (COM %LPT)* y se ejecuta la VI.

- c. Al ejecutarse la VI principal, se abre una nueva VI, el Módulo de PWM Modo 4, donde el estudiante seleccionará los parámetros de configuración.
- d. Seleccione en el control numérico **“Periodo PWM1”** un periodo de 0.0166667seg (60HZ) para el PWM.
- e. Seleccione en el control numérico **“Duty Cycle PWM1”** un ciclo de trabajo de 0.00833333seg (120Hz).
- f. Seleccione en el control numérico **“Periodo PWM0”** un periodo de 0.0066667seg (150HZ).
- g. Seleccione en el control numérico **“Duty Cycle PWM0”** un ciclo de trabajo de 0.00333333seg (300HZ).
- h. Seleccione en el menú desplegable **“PREESCALER”** 1.
- i. Seleccione en el menú desplegable **“DIVISOR DE RELOJ PWM”** Fosc/1024/15 (Opción de rango de frecuencia alcanzado por el entrenador).

Observar la tabla guía, mostrada al lado derecho de la interfaz gráfica de Labview.

- j. Ejecute la Vi.

En los indicadores numéricos **“Frec. PWM1”**, **“Frec. Duty Cycle PWM1”**, **“Frec. PWM0”** y **“Frec. Duty Cycle PWM0”** el usuario visualiza el valor de la frecuencia, luego de haberse ajustado al rango de frecuencias más cercano que el modo de PWM puede alcanzar.

- k. Mediante un osciloscopio o un “Fluke” compruebe el valor de la señal PWM generada por el entrenador digital USB.
- l. Retorne al menú principal y el entrenador digital USB esperará por nuevos comandos para un nuevo módulo.

