

# Sistemas temporizados implementados con microcontroladores Atmel, construcción de plataforma básica para explicar el uso detallado del temporizador Timer 2.

Christian Blanco, Raúl Bejarano, Carlos Valdivieso  
Facultad de Ingeniería en Electricidad y Computación  
Escuela Superior Politécnica del Litoral (ESPOL)  
Campus Gustavo Galindo, Km 30.5 Via Perimetral, Apartado 09-01-5863, Guayaquil-Ecuador  
[cblanco@fiec.espol.edu.ec](mailto:cblanco@fiec.espol.edu.ec), [ibejaran@fiec.espol.edu.ec](mailto:ibejaran@fiec.espol.edu.ec), [cvaldiv@fiec.espol.edu.ec](mailto:cvaldiv@fiec.espol.edu.ec)

## Resumen

*El proyecto consiste en el diseño e implementación de una plataforma entrenadora conteniendo el KIT AVR BUTTERFLY, con la finalidad de explicar el uso detallado del temporizador timer2.*

*Los sistemas temporizados implementados en la plataforma constan de dos ejemplos desarrollados en lenguaje ensamblador y tres ejemplos desarrollados en lenguaje C, y de esta manera contribuir al desarrollo teórico-práctico de la programación de los microcontroladores Atmel.*

*Se utiliza el microcontrolador atmega169 programado en Avr studio4 para el control de los dispositivos que posee el Avr Butterfly y la interacción con los dispositivos externos usados en cada uno de los ejemplos tales como: Leds, potenciómetro, motor DC 12v, entre otros.*

*La simulación de cada ejemplo es hecha mediante el software de simulación Proteus con la finalidad de verificar el correcto funcionamiento de cada sistema.*

**Palabras claves:** AVR BUTTERFLY, JOYSTICK, AVR STUDIO 4

## Abstract

*The project consists in the design and implementation of a platform trainer containing the AVR BUTTERFLY KIT, in order to explain the use of Tmr2 timer in detail.*

*Timed systems implemented on the platform have two examples developed in assembly language and three examples developed in C language, contribute to theoretical and practical programming of Atmel microcontrollers.*

*The microcontroller atmega169 programmed in Avr Studio4 is used to control the Avr Butterfly device in interaction with external devices used in the examples such as LEDs, potentiometer, 12V DC motor and others.*

*Simulations are performed using Proteus simulation software to verify the correct operation of each system.*

**Keywords:** AVR BUTTERFLY, JOYSTICK, AVR STUDIO 4

## 1. Introducción

Con el presente trabajo se implementará una plataforma para explicar el uso detallado del temporizador Timer 2, a través del desarrollo de una serie de ejercicios los cuales permiten conocer el funcionamiento, características y configuración del temporizador Timer2. Para el desarrollo de cada ejercicio se han elaborado los diagramas de bloques, diagramas de flujo y simulaciones, permitiendo así una mayor comprensión del desarrollo e implementación de cada ejercicio.

### 1.1 Descripción de Timer2

El Timer 2 del Atmega169 es un temporizador/contador de 8 bits. Su registro de conteo

conocido como TCNT2 es controlado por la unidad de control lógico la cual se encarga de indicar el sentido del conteo es decir si cambia de ascendente a descendente o viceversa, dependiendo del modo de operación que este configurado. Este Timer 2 posee dos tipos de interrupciones que son: interrupción por desborde y la interrupción por comparación. Posee un Pre-escalador, el cual puede ser asignado con ocho posibles valores.

El timer2 posee 4 modos de operación conocidos como: Modo Normal, Modo CTC, Modo PWM rápido y Modo PWM fase correcta. [1,3]

## 2. Herramientas de Hardware y Software

El presente capítulo contiene información sobre las herramientas de Software y Hardware utilizadas. Desde el punto de vista del Software se realizará una

descripción del entorno de trabajo de cada uno de ellos y así entender su utilidad en la elaboración de cada uno de los ejercicios mientras que desde el punto de vista del Hardware se detallara sus características importantes con el fin de entender los recursos disponibles y limitaciones en cada uno de los dispositivos utilizados.

## 2.1 AVR Butterfly

Este Hardware es un módulo de soporte que puede ser utilizado en numerosas aplicaciones. El AVR Butterfly contiene un micro controlador ATmega169, el cual va a realizarlas diferentes funciones de las que es capaz este kit. [1,3].

Las características del Butterfly son:

- Diseño de bajo poder.
- El tipo de paquete MLF
- Controlador de LCD.
- Memorias Flash, EEPROM, SRAM, DATAFLASH externos.
- Interfaces de comunicación UART, SPI, USI.
- Convertidor analógico a digital (ADC).
- Temporizadores / contadores.
- Reloj en tiempo real (RTC).
- Modulación por impulsos (PWM)



Figura 2.1 AVR Butterfly

## 2.2 Protues

El software PROTEUS, perteneciente a Labcenter Electronics, es un entorno integrado diseñado para la realización de proyectos de equipos electrónicos en todas sus etapas: diseño, simulación, depuración y construcción. Este emulador consta de dos programas principales: Ares e Isis, y los módulos VSM y Electra.



Figura 2.2 Proteus

## 2.3 AVR Studio 4

El software AVR studio incorpora un depurador que permite el control de ejecución con fuente y nivel de instrucción, paso a paso y puntos de interrupción, el registro, la memoria y E/S puntos y configuración y gestión, y apoyo a la programación completa para los programadores independientes, además permite crear archivos assembler (asm) y archivos .C

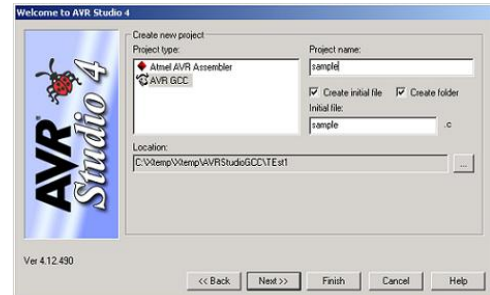


Figura 2.3 AVR Studio 4

## 3. Descripción e implementación del Proyecto

A continuación se ponen de manifiesto las etapas de diseño, implementación para la elaboración de este proyecto, sus diagramas de bloques, algoritmos y los códigos que serán cargados en el atmega169 del avr butterfly.

### 3.1 Implementación Plataforma AVR Butterfly

Para la implementación de la Plataforma se necesitaron de los siguientes elementos:

**Protoboard o placa de pruebas:** nos permite construir los prototipos de los circuitos electrónicos de los ejemplos elaborados.

**Cuatro pilas doble A:** se utilizan como fuente de poder del Avr Butterfly para poder programarlo.

**Socket para Pilas:** se utiliza para mantenerlas fijas y adicionalmente darnos la facilidad de cambiarlas en caso de que estén descargadas.

**Tarjeta Butterfly:** la cual será programada para la implementación física de los ejemplos.

**Bus de datos:** los cuales van conectados a los puertos B, D, F de la tarjeta Butterfly según el requerimiento del ejemplo a desarrollar.

**Cable USB a DB9:** se lo utiliza para poder cargar el código de cada ejemplo en el microcontrolador atmega169 del Avr Butterfly.

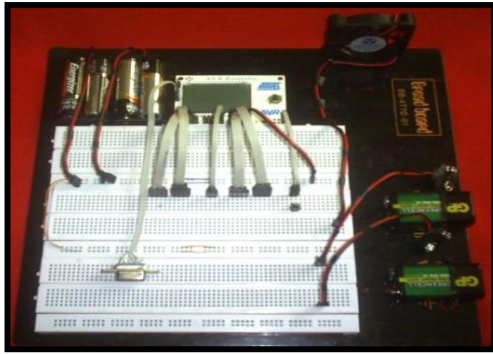


Figura 3.1 Plataforma AVR Butterfly

### 3.2 Descripción Teórica de los ejemplos

El proyecto se basa en la implementación de sistemas temporizados para mostrar el uso del Timer2, por lo que se han desarrollado una serie de ejemplos implementados tanto en lenguaje ensamblador como en lenguaje C, los cuales serán detallados a continuación.

#### 3.2.1 Contador up/down

El fundamento teórico se enfoca en la configuración del Timer2 en modo de operación Normal para realizar el conteo cada vez que ocurre la interrupción por desborde (overflow) del Timer2. Adicionalmente se configura el Timer2 para funcionar con fuente de reloj externa de 32,768 Hz y el respectivo pre-escalador de 128 para obtener un tiempo de temporización por desborde igual a 1Hz, calculado mediante la siguiente fórmula:

$$t(\text{over-flow}) = (f_{\text{clk}}/N)/256$$

Donde  $f_{\text{clk}}$  es la frecuencia de reloj en este caso externo y N es el valor del preescalador. Finalmente el conteo será mostrado en el LCD de Butterfly [1,3].

#### 3.2.2 Motor de 3 velocidades

El fundamento teórico se enfoca en la configuración del Timer2 en modo de Operación PWM Phase Correct para generar una onda cuadrada en el pin OC2A al ocurrir la igualdad entre el valor cargado en el registro de comparación OCR2A con el valor del registro de conteo del Timer TCNT2. Para poder generar el cambio de velocidad en el motor, se debe cambiar el valor del registro de comparación OCR2A lo que da como resultado la variación del ancho del pulso de la onda cuadrada pero el periodo de la misma no cambia, esto es lo ocurre cada vez que se usa el Joystick para hacer el respectivo cambio de velocidad. Finalmente mediante los leds se indica en que velocidad se encuentra el motor es decir si en velocidad 1, 2 o 3. [5]

#### 3.2.3 Encender o Apagar led mediante Interrupción

El fundamento Teórico se enfoca en la configuración del Timer2 en modo Normal para encender o apagar un LED cada 1segundo.

Para generar el estado de encendido del Led, se hace uso de la interrupción por Comparación es decir, cuando el valor del Registro OCR2A es igual al valor de registro de conteo del Timer2 TCNT2 entonces se procede a encender el Led y a mostrar en el LCD del Butterfly el mensaje "ON".

Para generar el estado de apagado del Led, se hace uso de la interrupción por desborde es decir, cuando el valor del registro de conteo del Timer2 TCNT2 alcanza su valor máximo 0xFF entonces se procede a apagar el Led y a mostrar en el LCD del Butterfly el mensaje "OFF".

#### 3.2.4 Onda cuadrada de frecuencia variable

El fundamento Teórico se enfoca en la configuración del Timer2 en modo CTC para generar una onda cuadrada mediante el Pin OC2A al ocurrir la igualdad entre el valor cargado en el registro de comparación OCR2A con el valor del registro de conteo del Timer2 TCNT2. Para poder generar el cambio de la frecuencia en la onda cuadrada, se debe cambiar el valor del registro OCR2A lo que genera una variación del periodo de la onda cuadrada, esto se logra cargando continuamente OCR2A con el valor del registro ADCH generado de la conversión ADC del voltaje inyectado en el Pin ADC4 al variar la resistencia del potenciómetro, dicha conversión ADC está configurada con ajuste hacia la izquierda para obtener una resolución de 8 bits los cuales quedan almacenados en ADCH. Finalmente en el LCD del Butterfly se presenta el valor BCD de OCR2A para poder realizar el respectivo cálculo de la frecuencia y así comprobar lo obtenido en la simulación.

#### 3.2.5 Control de velocidad de motor DC 12v

El fundamento teórico se enfoca en la configuración del Timer2 en modo FAST PWM para generar una onda cuadrada mediante el Pin OC2A al ocurrir la igualdad entre el valor cargado en el registro de comparación OCR2A con el valor del registro de conteo del Timer2 TCNT2. Para poder generar la variación de la velocidad del motor, se debe cambiar el valor del registro OCR2A lo que genera una variación en el ancho del pulso de la onda cuadrada pero su periodo no cambia, esto se logra cargando continuamente OCR2A con el valor del registro ADCH generado de la conversión ADC del voltaje inyectado en el Pin ADC4 al variar la resistencia del potenciómetro, dicha conversión ADC está configurada con ajuste hacia la izquierda para obtener una resolución de 8 bits los cuales quedan almacenados en ADCH. Finalmente en el LCD del Butterfly se presenta continuamente el mensaje "TIMER2 MODO FAST PWM".

#### 4. Simulación y Prueba

En esta sección se desarrolla el diagrama de bloques, el diagrama de flujo y la simulación de cada ejemplo, para finalmente poder ser implementado físicamente en el Protoboard.

##### 4.1 Ejemplo 1: Contador up/down

El siguiente programa esta implementado en lenguaje de ensamblador y consiste en un contador up / down, el sentido del conteo se cambia mediante el joystick del butterfly, donde la dirección hacia arriba indica conteo ascendente y la dirección hacia abajo indica conteo descendente. [11]

##### 4.1.1 Diagrama de bloques

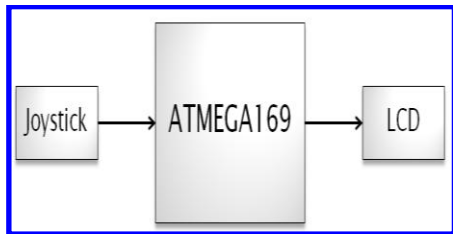
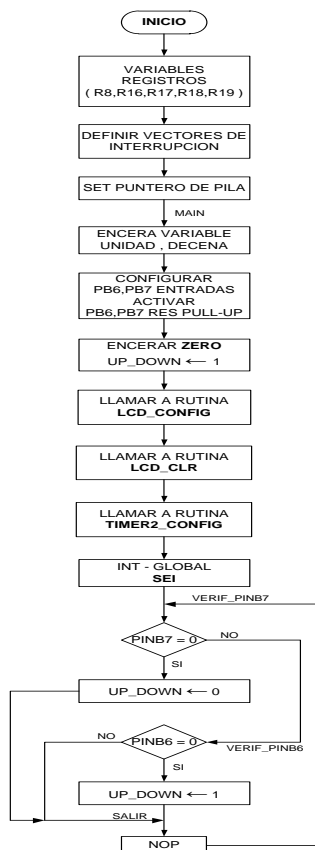


Figura 4.1.1 diagrama de bloques ejemplo 1

##### 4.1.2 Diagrama de flujo



##### 4.1.3 Simulación proteus

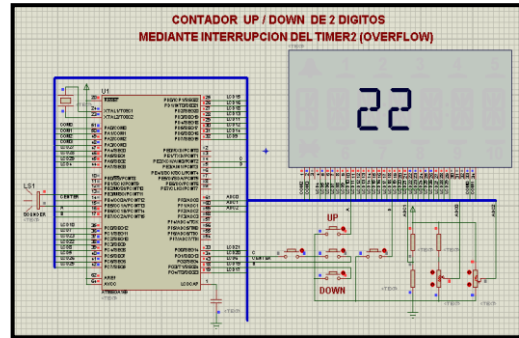


Figura 4.1.3 simulación ejemplo 1

##### 4.2 Ejemplo 2: Motor de 3 velocidades

El siguiente programa esta implementado en lenguaje de ensamblador y consiste en un motor de 3 velocidades, la velocidad del motor es controlada por el joystick, cuando se presiona hacia arriba se incrementa la velocidad y si se presiona en el medio la velocidad disminuye. [5]

##### 4.2.1 Diagrama de bloques

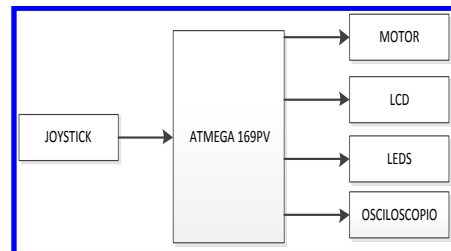
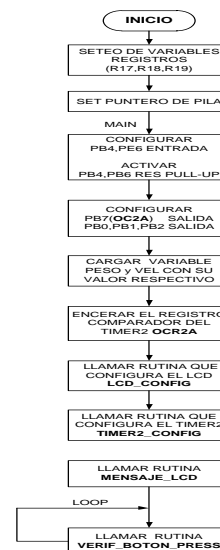


Figura 4.2.1 Diagrama de bloques ejemplo 2

##### 4.2.2 Diagrama de flujo



### 4.2.3 Simulación proteus

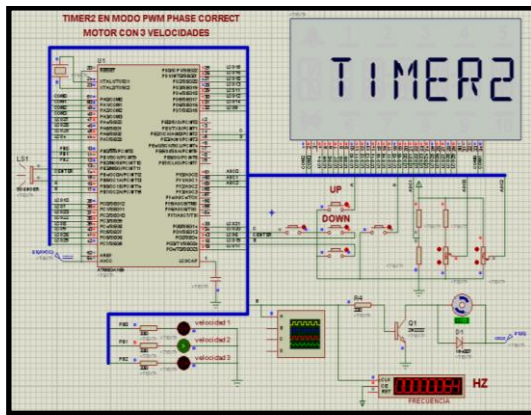


Figura 4.2.3 simulación ejemplo 2

### 4.3 Ejemplo 3: encender /apagar led mediante Interrupción

El siguiente programa esta implementado en lenguaje de C y consiste en encender o apagar un led cada 1 segundo haciendo uso de las interrupciones por comparación y por desborde del timer2. [4,11]

#### 4.3.1 Diagrama de bloques

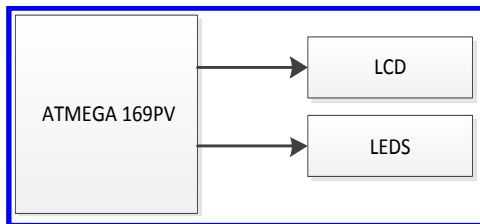
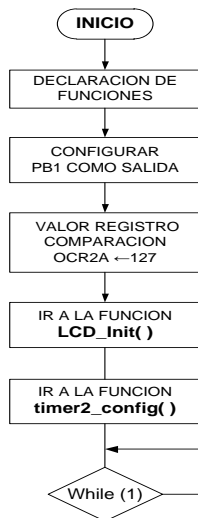


Figura 4.3.1 Diagrama de bloques ejemplo 3

#### 4.3.2 Diagrama de flujo



### 4.3.3 Simulación proteus

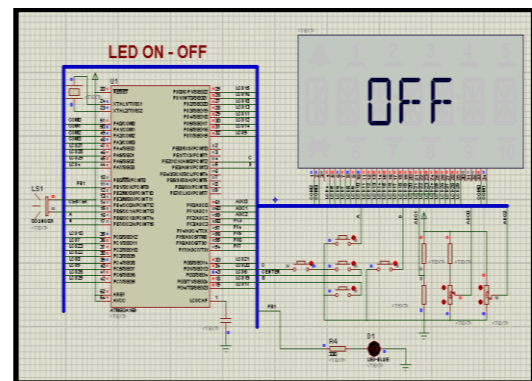
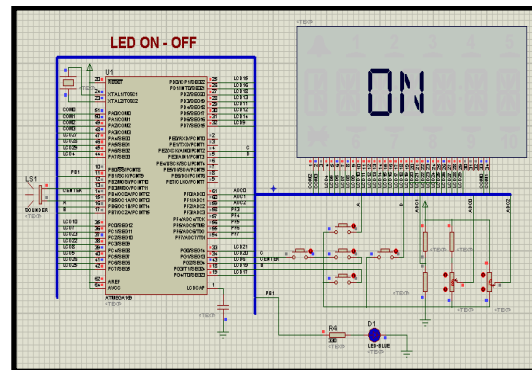


Figura 4.3.3 simulación ejemplo 3

### 4.4 Ejemplo 4: Onda Cuadrada de frecuencia variable

El ejemplo Consiste en la generación de una onda cuadrada mediante la configuración del Timer2 en modo CTC y cuya frecuencia se la hace variar en función del voltaje inyectado en el pin ADC4 el cual es controlado por un potenciómetro. Adicionalmente se muestra en el LCD el valor del Registro OCR2A. [1,6]

#### 4.4.1 Diagrama de bloques

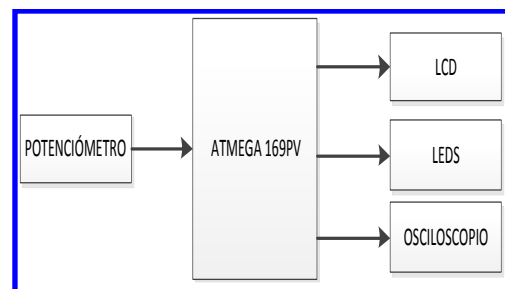
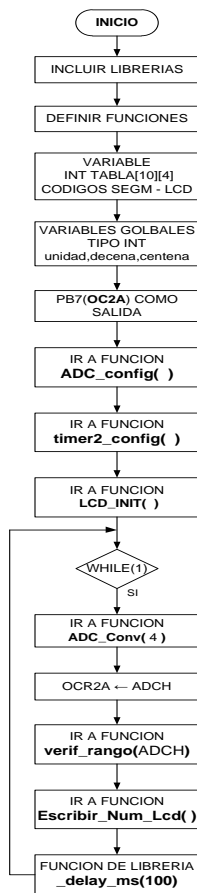


Figura 4.4.1 Diagrama de bloques ejemplo 4

#### 4.4.2 Diagrama de flujo



#### 4.4.3 Simulación proteus

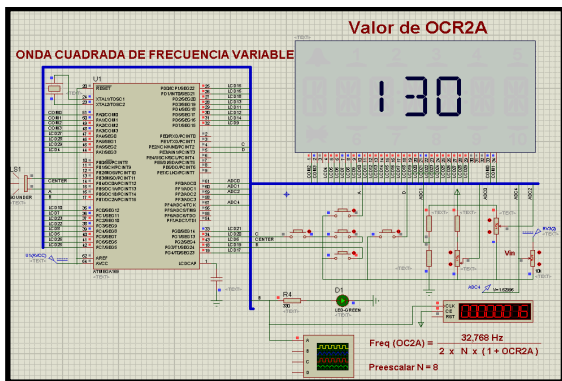


Figura 4.4.3 simulación ejemplo 4

#### 4.5 Ejemplo 5: Control de velocidad de motor DC

El ejemplo consiste en el control de la velocidad de un motor DC de 12v mediante la configuración del Timer2 en modo PWM rápido y haciendo uso de un potenciómetro con el cual se lleva el control de la velocidad del motor en función del voltaje inyectado en el pin ADC4. Adicionalmente se muestra en el LCD del Butterfly el mensaje TIMER2 FAST PWM. [1,2,6]

#### 4.5.1 Diagrama de bloques

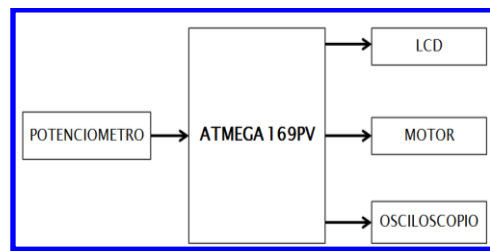
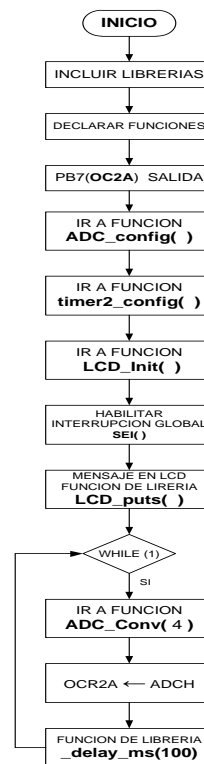


Figura 4.5.1 diagrama de bloques ejemplo 5

#### 4.5.2 Diagrama de flujo



#### 4.5.3 Simulación proteus

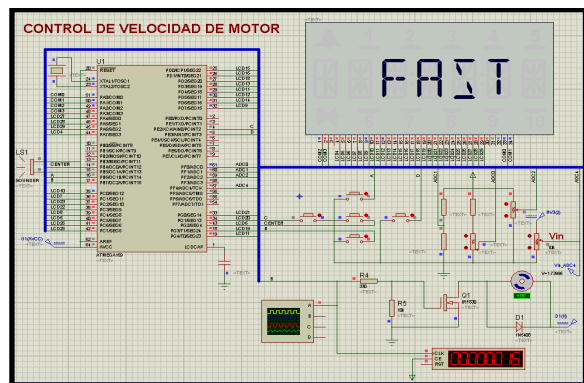


Figura 4.5.3 simulación ejemplo 5

## 5. Conclusiones

Al realizar el proyecto se logró implementar una plataforma entrenadora para el Kit AVR Butterfly de la familia ATMEL con lo cual los futuros estudiantes del laboratorio de microcontroladores contarán con esta versátil herramienta para el desarrollo de sus prácticas, además los ejercicios elaborados contribuirán a reforzar la comprensión de la configuración del Timer2.

La realización de los ejercicios de este proyecto nos ayudó a comprender la correcta manera de configurar el Timer2 en sus diferentes modos de operación y sus diferentes tipos de interrupciones y de igual manera la diferencia de programar usando lenguaje de ensamblador o lenguaje C.

La plataforma Avr Butterfly implementada se convierte en una buena herramienta didáctica para la visualización de los diferentes modos de operación del timer2 además de la flexibilidad que esta ofrece para la implementación de nuevos ejemplos a futuro.

## 6. Recomendaciones

Al momento de escribir un programa, es recomendable segmentarlo en funciones que luego serán utilizadas por el programa principal, con el fin de llevar un orden progresivo y así evitar confusiones al momento de revisar o corregir el código.

Se recomienda no asentar el Kit AVR Butterfly en superficies conductivas, ya que se pueden producir cortocircuitos, los cuales provocan daños en el dispositivo.

Es necesario revisar las hojas de especificaciones antes de trabajar con los dispositivos y en el caso del AVR Butterfly el de cada uno de los dispositivos que este posee tales como el atmega169, joystick, Lcd, sensores, etc.

## 7. Agradecimientos

A nuestro director, el Ing. Carlos Valdivieso Moreno de por su valiosa colaboración para poder desarrollar con éxito el presente proyecto de graduación.

A mis padres Luís Blanco y Teresa Salazar de quienes siempre tuve su apoyo constante e incondicional.

A mis tíos quienes me dijeron que sea constante y que lograría concluir mi tesina.

Christian Blanco Salazar.

A mis padres Raúl Bejarano y Graciela Saritama quienes me brindan el apoyo para alcanzar mis metas.

Finalmente un agradecimiento especial a las personas que siempre nos brindaron su apoyo y ayuda incondicional.

Raúl Bejarano Saritama.

## 8. Referencias

[1] Atmel, "8-bit microcontroller with 16k bytes in-system programmable flash", [http://www.atmel.com/dyn/resources/prod\\_documents/doc2514.pdf](http://www.atmel.com/dyn/resources/prod_documents/doc2514.pdf),

Fecha de consulta: 22 de agosto del 2011.

[2] Butterfly logger, "Esquemático del avr butterfly", <http://www.brokentoaster.com/butterflylogger/hw.html>,

Fecha de consulta: 22 de agosto del 2011.

[3] Vico Daniel, "timer 2 del avr atmel y sus registros", <http://vicodg.blogspot.com/2011/03/timer-2.html#!2011/03/timer-2.html>,

Fecha de consulta: 4 de septiembre del 2011.

[4] Calderon Johan, "Lenguaje c sobre como configurar el timer del avr atmel", <http://www.micro2c.com/categoria-Atmel-AVR-c2p0.html>,

Fecha de consulta: 4 de septiembre del 2011.

[5] Gonzales Daniel, "Códigos en ensamblador sobre el display y joystick", <http://sites.google.com/site/avrasminro/home/5-butterfly-lcd-joystick>,

Fecha de consulta: 13 septiembre del 2011.

[6] engineersgarage, "código de adc", <http://www.engineersgarage.com/embedded/avr-microcontroller-projects/adc-circuit>,

Fecha de consulta: 13 septiembre del 2011.

[7] Todo Topic, "Código de timer en assembler y tiempo de instrucción", <http://www.todopic.com.ar/foros/index.php?topic=27420.100>,

Fecha de consulta: 13 septiembre del 2011.

[8] Boellmann Werner, "avr lib", [http://www.nongnu.org/avr-libc/user-manual/group\\_avr\\_interrupts.html](http://www.nongnu.org/avr-libc/user-manual/group_avr_interrupts.html),

Fecha de consulta: 13 septiembre del 2011.

[9] Guerrero Richard, "kit de desarrollo avr butterfly, desarrollo de guía de prácticas de laboratorio y tutoriales", <http://www3.espe.edu.ec:8700/bitstream/21000/424/.../T-ESPE-014271.pdf>,

Fecha de consulta: 27 de octubre del 2011.

[10] Universidad de Oviedo, "modulos de temporizacion del pic", [http://www.ate.uniovi.es/fernando/Doc2006/Sed\\_06/Presentaciones/Timers\\_v2.pdf](http://www.ate.uniovi.es/fernando/Doc2006/Sed_06/Presentaciones/Timers_v2.pdf),

Fecha de consulta: 27 de octubre del 2011.

[11] Kui Attie, "tiempo de temporización por sobreflujo igual a 1hz", <http://www.sparkfun.com/products/540>,

Fecha de consulta: 27 de octubre del 2011.