



Pontificia Universidad Católica Argentina  
Facultad de Ciencias Fisicomatemáticas e Ingeniería

Carrera de Ingeniería Electrónica y Comunicaciones

## **Trabajo Práctico:**

### **Silla de ruedas controlada por movimientos oculares**

**Alumno: Eduardo Luengo**

**Profesor: Ing. Norberto Heyaca**

**Tutor: Ing. Ricardo Vecchio**

## Resumen

Este trabajo trata el diseño, construcción y medición de una silla de ruedas. Se pretende comandar la misma mediante dispositivos electrónicos, como también electro-mecánicos, los cuales brindaran movimientos laterales, como también hacia adelante. Su finalidad es mejorar la calidad de vida de personas discapacitadas debido a sus extremidades (piernas y brazos-manos). Si bien se trata de un modelo a escala, es perfectamente viable de adaptación en la práctica para casos particulares de discapacidad que así lo requieran.

El trabajo se complementa con una sección de Anexos con el motivo de profundizar conceptos que irán apareciendo a lo largo de las explicaciones técnicas de funcionamiento.

# Índice

## 1 – INTRODUCCIÓN

### 1.1 – Generalidades

### 1.2 – Origen, Historia y tipos de sillas de ruedas

### 1.3 – La Electrooculografia aplicada a un fin específico

## 2 – DESARROLLO

### 2.1 – Herramientas de Programación involucradas

### 2.2 – Funcionamiento de los electrodos y control de los motores

### 2.3 – Detección de señal de audio (silbido)

#### 2.3.1 – MIC-electret y obtención de señal alterna

#### 2.3.2 – Amplificación

#### 2.3.3 – Filtro pasabanda

#### 2.3.4 – Rectificación de señal

#### 2.3.5 – Schmitt trigger

### 2.4 – Descripción del microcontrolador en la implementación

### 2.5 – Programa en C del microcontrolador

## 3 – PRUEBAS EN CAMPO

## 4 – FOTOS DEL PROTOTIPO

## 5 – CONCLUSIONES

## 6 – REFERENCIAS BIBLIOGRÁFICAS

## 7 – ANEXOS

# 1 – INTRODUCCIÓN

## 1.1 – Generalidades

El trabajo práctico se subdividió en dos partes las cuales fueron elaboradas en forma separada. La primera parte fue presentada por el alumno Juan José Oubiña durante el primer cuatrimestre del año 2007 como Trabajo Final de la carrera de Ingeniería Electrónica y Comunicaciones. Allí se obtuvo una señal electrooculográfica de valores del orden del volt a través de 3 electrodos de medicina conectados en el rostro de una persona, y mediante un circuito compuesto de filtros de bajo ruido y amplificadores de instrumentación.

La segunda parte del trabajo se realizó durante el segundo semestre del año 2007 y es una aplicación práctica a partir de la obtención de señales electrooculográficas. Dado que se explicara la interacción de ambas partes, es recomendable recurrir al material escrito de la primera parte para obtener más información.

Hoy en día, debido al avance tecnológico de las últimas cuatro décadas la electrónica se ve inmersa cada vez más en soluciones de diferente índole. Muchas de estas soluciones responden a necesidades humanas de carácter urgente. La discapacidad es una de ellas. Claro que al hablar de discapacidad existen diferentes tipos y por ende numerosas maneras de lograr soluciones. Existen los siguientes tipos de discapacidad: Física, Psíquica, Sensorial y Intelectual o Mental. En las próximas páginas se pretende dar una solución a una discapacidad física, en particular, la de personas minusválidas hasta el punto de cuadripléjicas.

## 1.2 – Origen, Historia y tipos de sillas de ruedas

La silla de ruedas supone una solución a las limitaciones de desplazamiento que tienen algunas personas con discapacidad. Las primeras sillas de ruedas aparecieron en los EE.UU. en la década del 50. A pesar que las sillas actuales se podrían llamar electrónicas, las primeras eran verdaderamente eléctricas. La E&J 840 era una máquina simple que no tenía tableros de circuitos ni controles suaves ni controlables como los actuales y en vez de control manual único se controlaba prendiendo y apagando cuatro interruptores que harían que la silla saltara al comenzar a andar, parar o cambiar de dirección.

Algo de la brusquedad se podía suavizar con dos pesados motores de bobina de partida lenta, factor demasiado ineficiente, por no decir penosamente lento. La silla, llanamente, tenía dos velocidades básicas, alta y baja; o como la llamaban los mecánicos de sillas de ruedas de ese entonces, lenta y horriblemente lenta. La silla tenía dos baterías de seis voltios conectadas en paralelo para las velocidades bajas y, para la velocidad alta, en serie y se debía parar la silla para cambiar de velocidad. A pesar de todos los inconvenientes, se trataba de movilidad independiente.

El siguiente gran avance en el diseño de sillas de ruedas fue la integración de un circuito electrónico y control de manejo gradual. Esto permitía que el usuario tuviera mayor control en la operación de la silla. Ahora, mientras más se moviera el bastón de control hacia la dirección deseada, más rápido se movería la silla en esa dirección. El circuito electrónico también permitió que se reemplazaran los motores de bobina con motores de magneto permanente más livianos y eficientes. Hoy en día, en vez de estar limitadas a un comienzo lento, el operador puede partir a la velocidad que sea capaz.



Los años 1990 vieron un notable aumento de la investigación en Sillas de Ruedas Autónomas (silla de ruedas) en todo el mundo en un intento de aplicar tecnologías, ya experimentadas en los múltiples campos de la robótica móvil, al campo específico de la Asistencia a la Movilidad.

Pueden distinguirse dos niveles en las diferentes arquitecturas de silla de ruedas existentes: el nivel físico (hardware) y el nivel lógico (software), siendo éste último parcialmente independiente del primero. No obstante, las posibilidades funcionales y el costo del sistema está más ligado al nivel físico que al lógico, por lo que el primer criterio de decisión es la definición del hardware del sistema. A grandes rasgos, las arquitecturas a nivel físico de los diferentes sistemas de silla de ruedas responden a alguna de las siguientes líneas:

- Control centralizado, monoprocesador.
- Control centralizado, multiprocesador.
- Control distribuido, multiprocesador.

#### Control centralizado, monoprocesador

La solución más común parte de una silla motorizada comercial básica a la que se le añade una unidad de control basada en un PC comercial.

Una gran ventaja derivada del uso de recursos ligados al sistema PC son bajo costo y alta potencia de cálculo en la unidad central, así como un amplio repertorio de recursos software. Pero el inconveniente estriba en su propia estructura centralizada. La conexión entre las unidades sensoras y los actuadores del sistema (motores, etc.) se realiza mediante cableado dedicado usando puertos de Entrada/Salida, o mediante tarjetas periféricas diseñadas al uso; de esta forma resulta difícil modificar o actualizar el hardware del sistema para adaptarse a nuevas necesidades, nuevos sensores o a cambios en la configuración del mismo.

#### Control centralizado, multiprocesador

El principio de diseño coincide en principio con el anterior: se parte de una plataforma estándar a la que se le añaden las interfaces de E/S necesarias para acceder a las informaciones de los sensores y elementos de mando y poder actuar sobre las unidades motrices.

Su máxima ventaja reside en la utilización de componentes estándar en robótica industrial y en la paralelización de procesos mediante el uso de varios procesadores independientes. Sin embargo, el uso de esta arquitectura se liga, casi de forma exclusiva, con la experimentación de estructuras software de control, pues se trata de una solución desventajosa frente a la anterior, principalmente en términos de costo, compartiendo además sus inconvenientes: escasa flexibilidad y compleja configuración hardware.

#### Control distribuido, multiprocesador

Las soluciones más flexibles y con mejor relación prestaciones/costo se basan en sistemas distribuidos en los que la inteligencia del sistema reside en múltiples Unidades de Proceso Dedicadas, intercomunicadas por uno o varios canales serie.

Las arquitecturas distribuidas tienen como principal ventaja la de operar vía mensajes; esto es: las informaciones de configuración, estado y comandos entre los elementos del sistema, se realizan mediante el intercambio de datos siguiendo un protocolo prefijado. La actualización o reconfiguración de los diversos subsistemas se facilita al no ser necesario realizar recableados de conexiones físicas sino sólo lógicas: añadir, quitar o cambiar una determinada prestación del sistema se logra insertando un nuevo elemento en el Bus y reconfigurando la programación, simplemente.

El aumento objetivo de las prestaciones de una silla de ruedas que siga esta estructura justifica la elección de esta configuración como la óptima de entre las enumeradas anteriormente, tanto a nivel hardware como a nivel software.

### 1.3 – La Electrooculografía aplicada a un fin específico

La ingeniería biomédica ha desarrollado diferentes técnicas de obtención de señales eléctricas (biopotenciales) que emiten las diferentes zonas de nuestro cuerpo. Un biopotencial es una diferencia de tensión eléctrica, causada por un flujo de corriente iónica a través del cuerpo. Muchas de ellas se utilizan cotidianamente para realizar chequeos cardiacos, ópticos, etc. A continuación, se muestran algunos de los valores típicos para diferentes señales bioeléctricas:

SEÑAL	MAGNITUD	ANCHO DE BANDA (Hz)
ECG (Electrocardiograma)	0,5 – 4 mV	0,01 – 250
EEG (Electroencefalograma)	5 – 300 $\mu$ V	DC - 150
EKG (Electrogastrograma)	10 – 1000 $\mu$ V	DC - 1
EMG (Electromiograma)	0,1 – 5 mV	DC - 10000
EOG (Electrooculograma)	50 – 3500 $\mu$ V	DC - 50
ERG (Electroretinograma)	0 – 900 $\mu$ V	DC - 50

NOTA: La Electrooculografía, es utilizada en el campo de la medicina para detectar los movimientos oculares y mediante un transductor registrar dichos movimientos en papel milimetrado usando un graficador.

Para más información sobre ingeniería biomédica dirigirse a la sección Anexos.

La electrooculografía (EOG) no solo se implementa para la adquisición de datos en estudios médicos sino que puede ser aplicada para diferentes fines.

En cuanto a los trabajos desarrollados se puede mencionar un mouse comandado por los ojos basado en Electrooculografía, desarrollado por los alumnos Sevillano y Dos Santos, egresados de la carrera de Bioingeniería de la Universidad Nacional de Entre Ríos. También existe un mouse para discapacitados que utiliza una cámara de video instalada en la cabeza del usuario, desarrollado por docentes y alumnos de la Escuela Técnica ORT en la Ciudad de Buenos Aires. Este último proyecto no está basado en movimientos oculares, sino en movimientos de la cabeza. Otro ejemplo en el cual puede aplicarse esta técnica es en la locomoción de silla de ruedas. En la actualidad existen multitud de sistemas de ayuda al guiado de silla. Todos estos sistemas intentan ayudar a las personas con minusvalías en el control de su guiado de forma que éste se haga más sencillo y por lo tanto más fácil de llevar a cabo. Estos sistemas pueden funcionar de forma autónoma o como sistemas de referencia o información de las personas que los controlan. No obstante, en los últimos años, han aumentado considerablemente las aplicaciones encaminadas a desarrollar sistemas de ayuda de personas con algún tipo de discapacidad y por lo tanto los sistemas tradicionales de control no son válidos en estos casos.

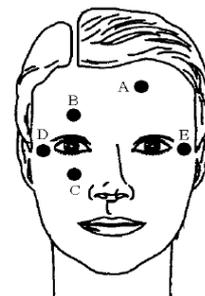
Siguiendo esta línea, son muchos los sistemas desarrollados con tal fin: sistemas de videooculografía (VOG) u oculografía infrarroja (IROG) que se basan en detectar la posición del ojo utilizando una cámara; existen aplicaciones basadas en técnicas de reconocimiento de voz con el objetivo de reconocer ciertas ordenes básicas para el control de cualquier instrumento o móvil, aunque la más popular de todas estas técnicas consiste en utilizar un joystick o pantalla táctil con el objeto de poder realizar las diferentes órdenes de control.

Estas técnicas de ayuda pueden aplicarse a diferentes personas según su grado de discapacidad, de forma que siempre se utilizará aquella o aquellas que resulten más fáciles para cada persona.

El objetivo de este trabajo práctico es realizar el guiado de una silla de ruedas mediante la posición ocular utilizando electrooculografía.

El principio de funcionamiento consiste en detectar las diferencias de potenciales eléctricos (biopotenciales) que se generan en la periferia del ojo cuando este se mueve utilizando la técnica mencionada anteriormente. Por medio de electrodos externos que se apoyan en la piel del operador, se detecta esta variación de potencial.

Al tener colocados un electrodo como referencia (A), más otros dos en la parte lateral de la cavidad ocular (D Y E), podemos obtener la información necesaria para mover los motores de continua.



En cuanto a su movilidad, el dispositivo se regirá por dos estados de locomoción: GIRO y PARADA.

Para pasar de un estado a otro, la persona que se encuentre comandando dicho dispositivo deberá generar un silbido de corta duración (aprox. 2 seg.), el cual servirá de entrada para conmutar entre ambos estados, y así poder vincularlos para obtener los movimientos deseados.

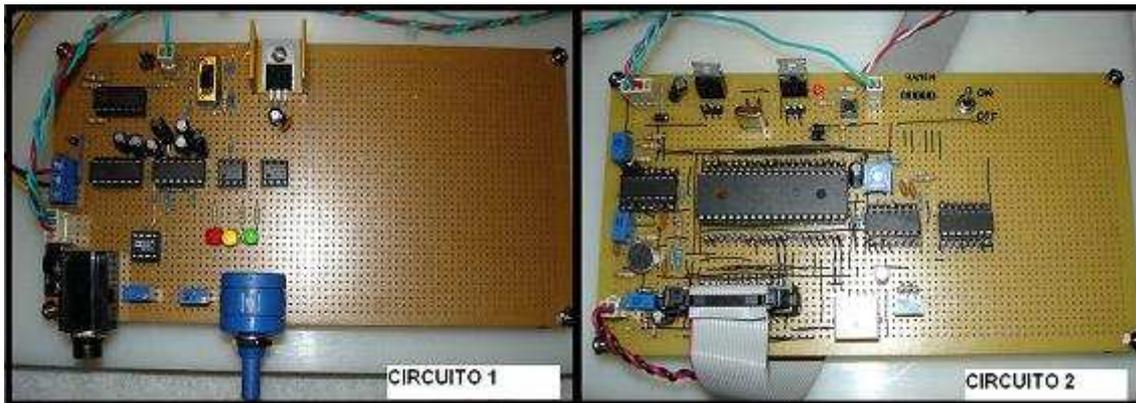
En el estado GIRO los motores obedecen al movimiento de los ojos. Con la mirada hacia el centro la silla avanza a una velocidad fija.

En el estado PARADA los motores se encontraran detenidos hasta que el operador vuelva al estado GIRO.

## 2 – Desarrollo

En esta sección se describe el diseño del circuito como también las herramientas que formaron parte de la implementación. El trabajo se divide en dos circuitos de los cuales el segundo es el que se explicara a detalle:

- **Circuito 1:** Circuito de amplificación y filtrado de la señal de los electrodos (Alumno: Juan Oubiña, ver Anexos Circuito 1)
- **Circuito 2:** Circuito de manipuleo de las señales de entrada y obtención de los valores de salida para comandar los motores de CC. (Alumno: Eduardo Luengo)



En el **circuito 2**, el microcontrolador PIC18F452 (fabricante Microchip) fue escogido para llevar a cabo todas las acciones acordes a cada entrada. Los 'PIC' son una familia de microcontroladores tipo RISC fabricados por Microchip Technology Inc. y derivados del PIC1650, originalmente desarrollado por la división de microelectrónica de General Instruments. Hoy en día multitud de PICs vienen con varios periféricos incluidos (módulos de comunicación serie, UARTs, núcleos de control de motores, etc.) y con memoria de programa desde 512 a 32.000 palabras (una *palabra* corresponde a una instrucción en ensamblador, y puede ser 12, 14 o 16 bits, dependiendo de la familia específica de PICmicro).

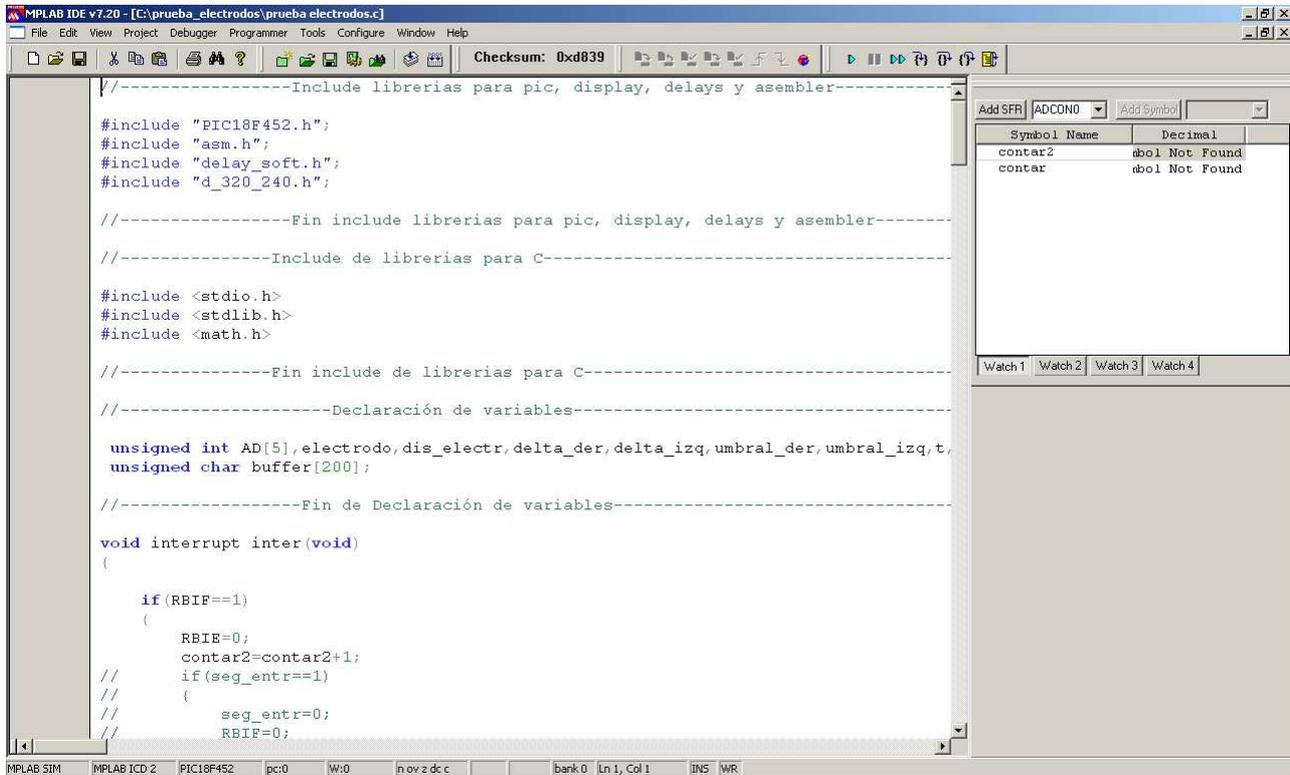
El PIC usa un juego de instrucciones tipo RISC, cuyo número puede variar desde 35 para PICs de gama baja a 70 para los de gama alta. Las instrucciones se clasifican entre las que realizan operaciones entre el acumulador y una constante, entre el acumulador y una posición de memoria, instrucciones de condicionamiento y de salto/retorno, implementación de interrupciones y una para pasar a modo de bajo consumo llamada *sleep*.



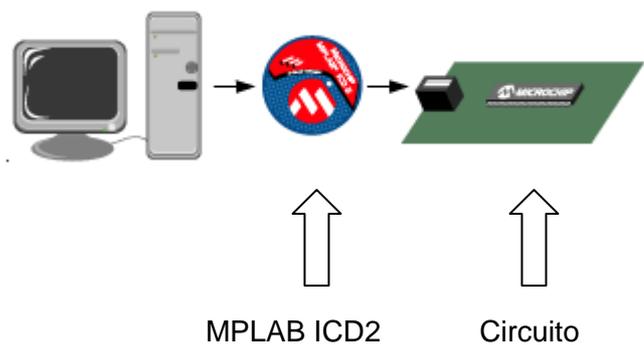
### 2.1 – Herramientas de Programación involucradas

Microchip proporciona un entorno de desarrollo freeware llamado MPLAB que incluye un simulador software y un ensamblador.

La programación del PIC se realiza en lenguaje de programación C utilizando la aplicación MPLAB (fabricante Microchip) y modulo HI-TECH PICC18 TOOLSUITE.



Ambos permiten programar en C, compilar, debuggear y enviar el código .hex al microcontrolador (MPLAB ICD2). Como se ve en la figura el MPLAB ICD2 vincula la PC por USB al circuito a programar con un conector RJ-45. Esta técnica se conoce como programación IN-CIRCUIT, por ende, no hay necesidad de tener que retirar el PIC del circuito para realizar cambios en el código.

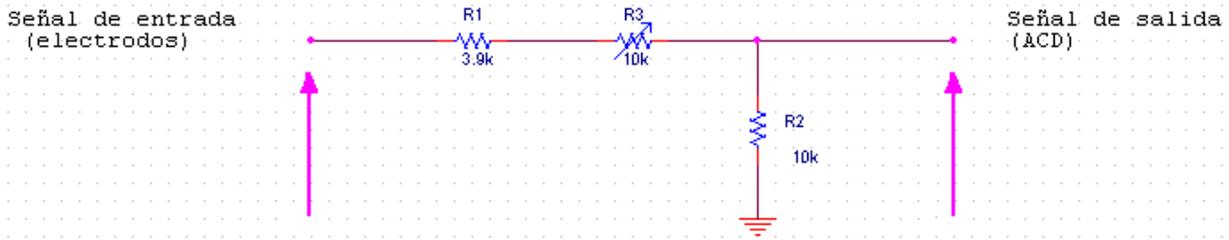


Nota: Para transferir el código de una PC al PIC normalmente se usa un dispositivo llamado programador (en este caso es el ICD2).

### 2.2 – Funcionamiento de los electrodos y control de los motores

La señal proveniente de los electrodos, tiene como máxima excursión los valores  $\pm 12$ volts. Antes de poner en funcionamiento la silla debe realizarse una calibración del **circuito 1** de manera de obtener una tensión entre 0 y 10volt. Esa señal al entrar al **circuito 2**, primeramente es atenuada 2 veces aproximadamente, para luego poder ser introducida al ADC del microcontrolador.

La atenuación de la señal del **circuito 1** en el **circuito 2** se realizo a partir de un filtro pasivo con los valores que se ven en la grafica.



$$V_{sal} = \frac{R2 * V_{entr}}{(R1 + R3) + R2}$$

Si bien R1+ R3 debe valer 10k para atenuar dos veces, la idea de colocar un R3 variable es para poder ajustar mejor la atenuación y no depender de valores fijos.

Volt de electrodos	Volt de electrodos con G=1/2
-2	-1
-1	-0,5
0	0
1	0,5
2	1
3	1,5
4	2
5	2,5
6	3
7	3,5
8	4
9	4,5

En rojo se detalla el umbral central cuando la mirada del operador se encuentra al centro.

Luego de atenuar la señal de los electrodos se coloco un BUFFER para separar etapas de impedancia entre el divisor resistivo y el pin de entrada del microcontrolador.

Los valores tomados de los electrodos son muestreados cada vez que se produce una interrupción por timer en el programa principal del microcontrolador, y se pueden visualizar en el display.

En materia de programación, el timer 2 fue elegido para realizar esta aplicación y es el único que puede utilizarse en el modulo PWM.

Una vez obtenida la señal analógica de los electrodos y muestreada a valores discretos, el siguiente paso es comandar los motores mediante PWM.

Se supusieron 2 umbrales (derecha e izquierdo) para poder dividir los valores provenientes de los electrodos. Valores mayores a 160 se entenderán como *girar hacia la derecha*, valores menores a 100 como *girar hacia la izquierda* y valores entre 100 y 160 como centro, o sea, se avanzara a una velocidad fija (ambos motores en funcionamiento).

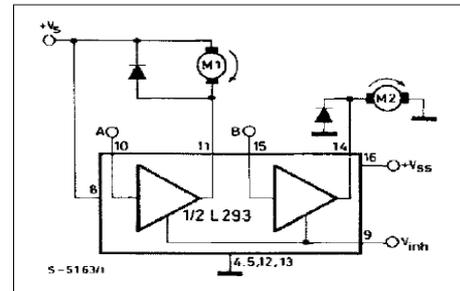
Como valor central de referencia se tomo 130, entonces realizando una resta entre este valor y el tomado por los electrodos se puede obtener un 'delta de giro'. Ese delta modula los anchos de los pulsos PWM, que es un modulo del PIC18F452 para controlar las revoluciones de los motores de corriente continua. Esta técnica hace variar la velocidad de rotación de un motor.

Con un 'duty cycle' pequeño durante varios periodos de timer, la velocidad del motor es menor que con 'duty cycle' mayores.

En el extremo cuando el 'duty cycle' sea igual al periodo del timer, la señal aplicada a los motores es la misma que al aplicar una tensión continua. La señal de 5v generada por el modulo PWM es enviada a la silla de ruedas mediante 5 cables de cobre de diámetro 0,20mm (tensión 12v, masa, enable chip, input1, input2). A bordo de la silla se encuentra el circuito integrado L293E que se utiliza como driver para los motores de continua. Ya que los motores son de 12v, la señal PWM debe ser amplificada para poder controlar los mismos. En la siguiente figura, tomada de la hoja de datos del L293E, se muestra como se realizo el circuito y la conexión de los motores al driver.

**APPLICATION INFORMATION**

**Figure 8 :** DC Motor Controls (with connection to ground and to the supply voltage)



V <sub>inH</sub>	A	M1	B	M2
H	H	Fast Motor Stop	H	Run
H	L	Run	L	Fast Motor Stop
L	X	Free Running Motor Stop	X	Free Running Motor Stop

L = Low      H = High      X = Don't Care

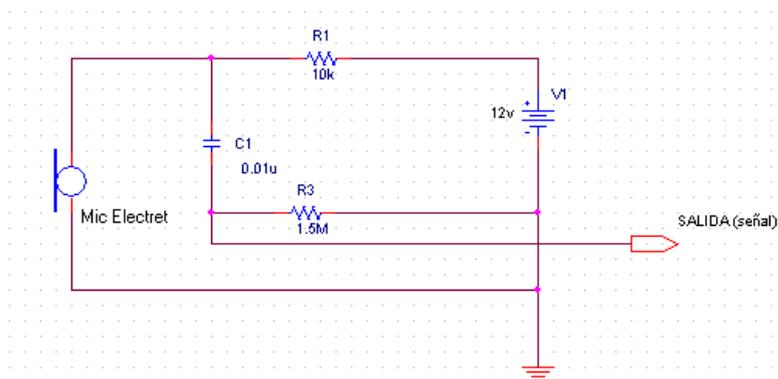
**2.3 – Detección de señal de audio (silbido)**

El circuito de captación señal de frecuencia 1khz aprox. se realizo en varias etapas que se detallan a continuación.

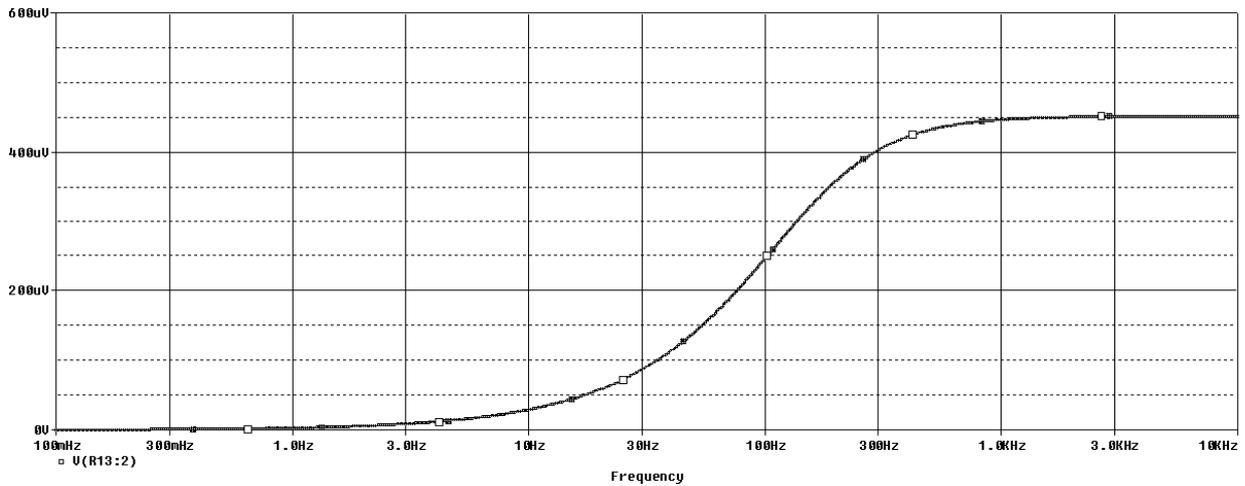


**2.3.1 – MIC-electret y obtención de señal alterna**

Para alimentar el micrófono y a su vez obtener la señal de alterna generada por el mismo se realizo el circuito como se ve en la figura.

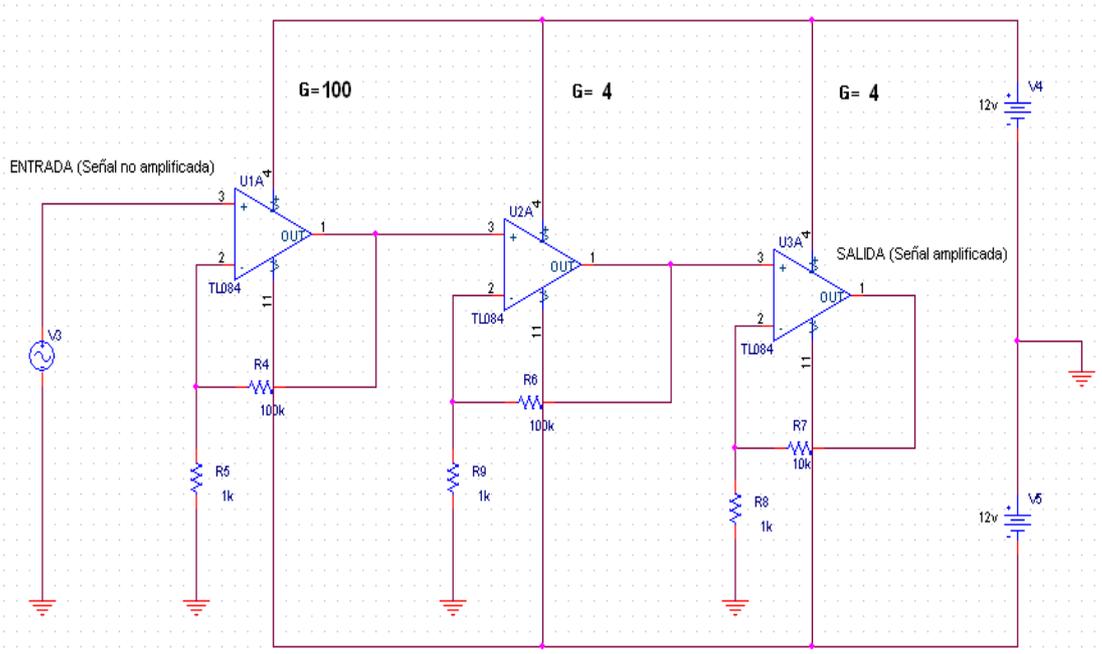


Se eligió  $R1=10K$  ya que la corriente máxima de alimentación no debe superar los 5.5mA. El capacitor  $C1=0.01\mu F$  filtra las frecuencias bajas que no interesan para esta aplicación. A continuación se detalla la respuesta en frecuencia en ORCAD para este circuito sobre la SALIDA (señal).



### 2.3.2 – Amplificación

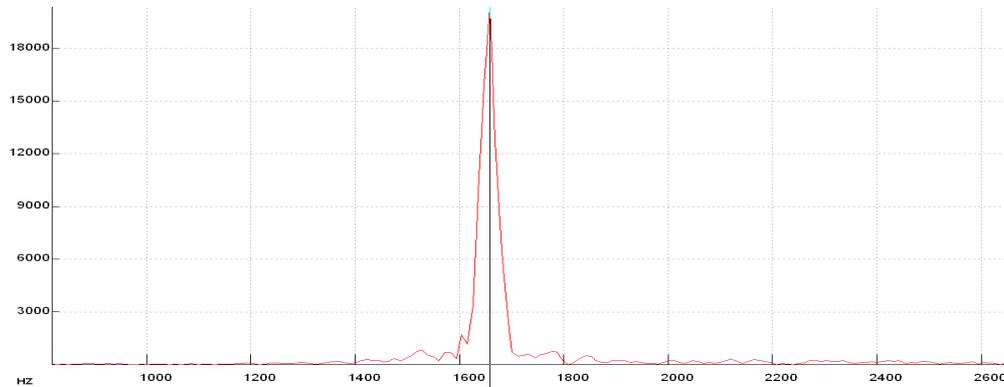
Se realizo en tres etapas de las cuales la primera gana 100, la segunda y tercera 4 veces. La razón se debe a que con un solo amplificador operacional la ganancia total es demasiado grande y el dispositivo deja de trabajar en su zona lineal. Además la inestabilidad y el ruido que inserta el amplificador con alta ganancia degradan la señal notablemente. Se utilizo el amplificador operacional de bajo ruido TL084CN (también para la etapa de filtrado) en configuración no inversor.



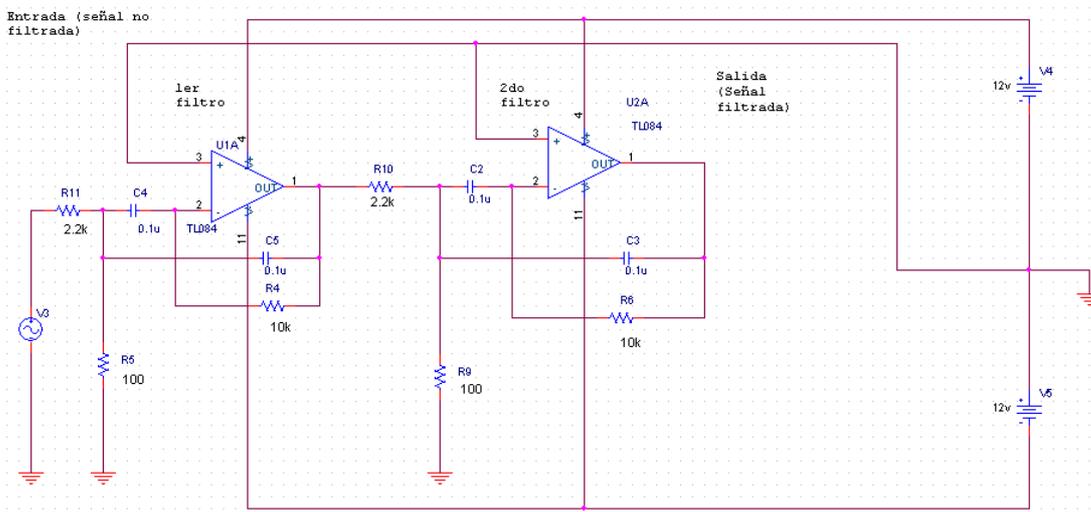
$$\frac{V_{salida}}{V_{entrada}} = 1 + \frac{R_{reali}}{R_{masa}}$$

### 2.3.3 – Filtro pasabanda

Esta etapa consta de dos filtros pasabanda los cuales dejan pasar la banda de frecuencias útiles, o sea, aquellas que hagan posible la detección de un silbido ente 1500 y 1800 hz.



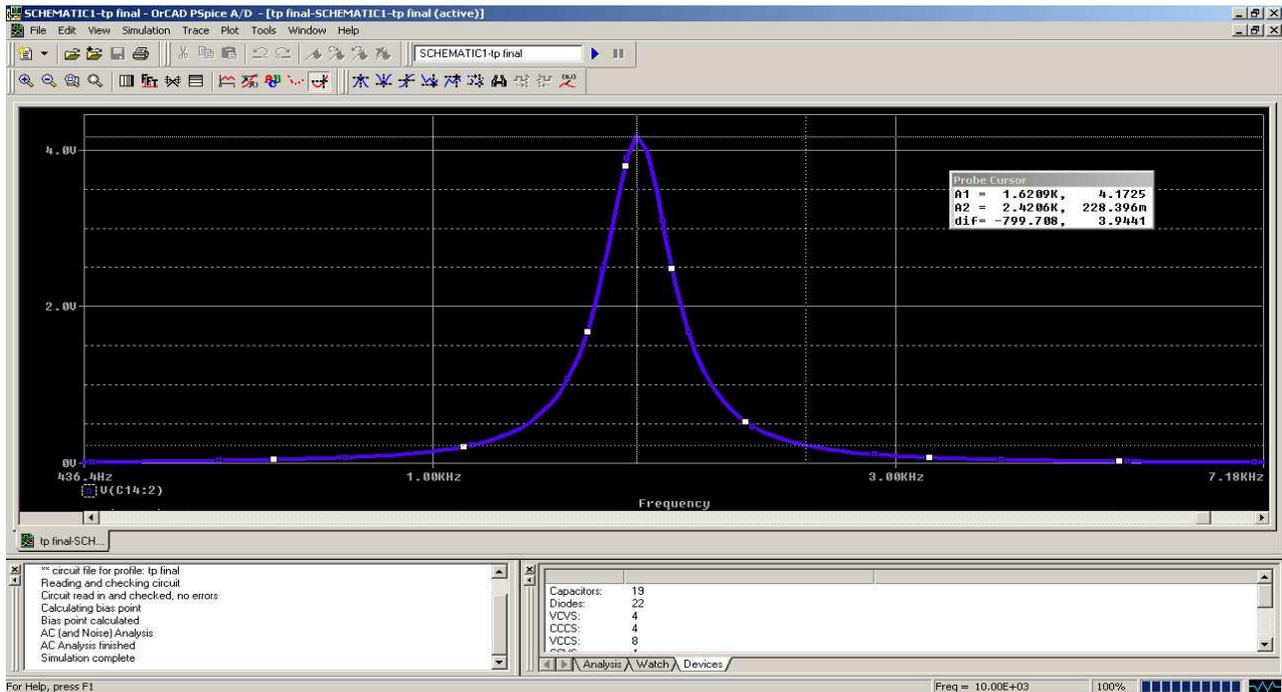
Hay que destacar que esta etapa se complicó aun más de lo pensado previamente, ya que la frecuencia utilizada para PWM es de 2.44kHz afectando la detección del silbido generado por el operador. Esta frecuencia es la mínima que se puede obtener con el timer 2 para el modulo PWM, una frecuencia mayor que esta dificulta el manejo de los motores de continua por lo que se debió hacer un filtro pasabanda que solo deje pasar la frecuencia cercanas a 1600hz y atenúe la frecuencia de 2.44khz.



Características de los filtros:

- Frecuencia central en 1627hz
- BW = 318 hz
- G = 2.3
- Q = 5.11

A continuación se muestra una simulación en Orcad de la señal rectificada. (color Azul)

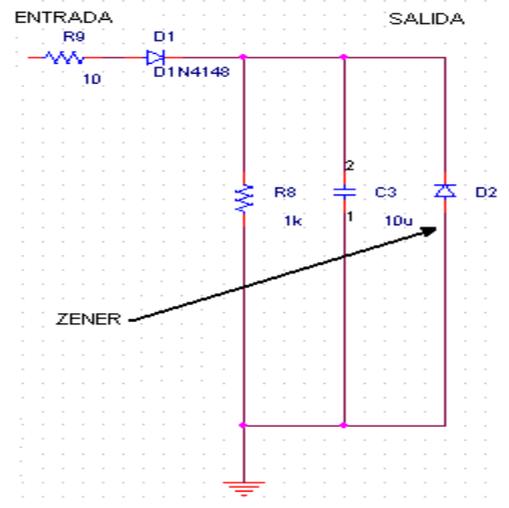


Nótese que para una frecuencia de 2.44khz la señal se atenúa considerablemente.

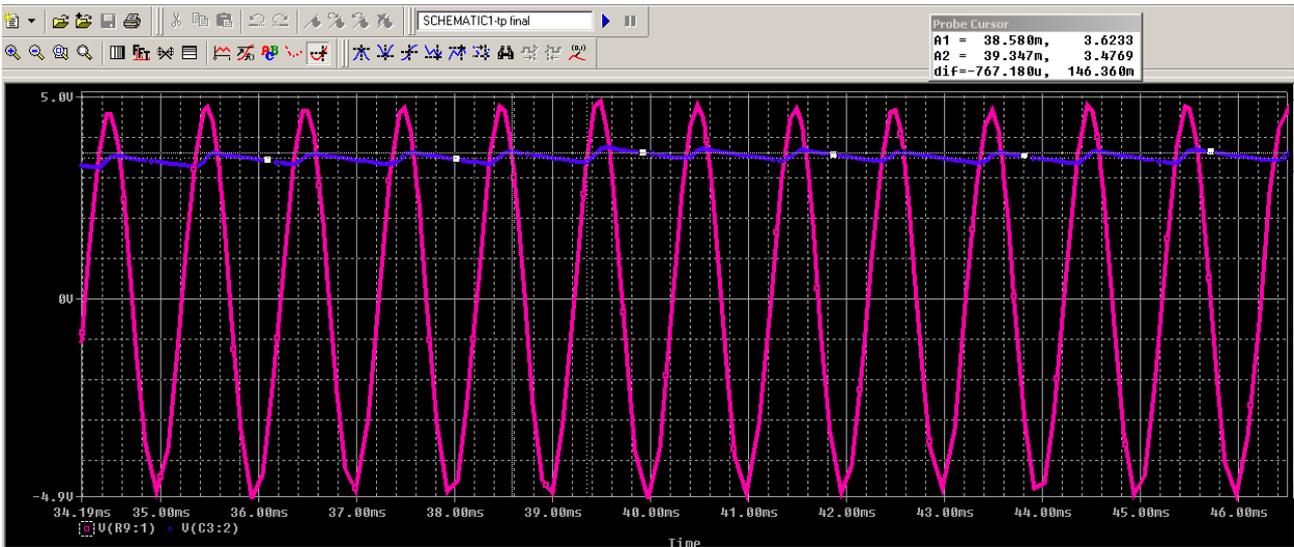
### 2.3.4 – Rectificación de señal

La señal alterna de salida de la etapa anterior no podría introducirse al PIC sin realizar la rectificación para obtener de ella una señal de tensión continua entre 0 y 5 volts. Para ello se debe rectificar los ciclos de la señal alterna con un diodo y un capacitor (rectificador de media onda) En la figura se muestra el circuito.

La señal proveniente de la salida de la etapa de filtrado tendrá una frecuencia aprox de 1khz, o lo que es lo mismo, un  $T=1\text{ms}$ . El tiempo descarga del circuito RC ( $\tau = r \times c$ ) se determino que sea 10 veces mas lento que el T de la señal (10ms). S  
Suponiendo una  $r=1\text{k}\Omega$  el capacitor debe ser de 10uF. El diodo zener 1N5231B se coloco para fijar una tensión de 5 volts a la salida.



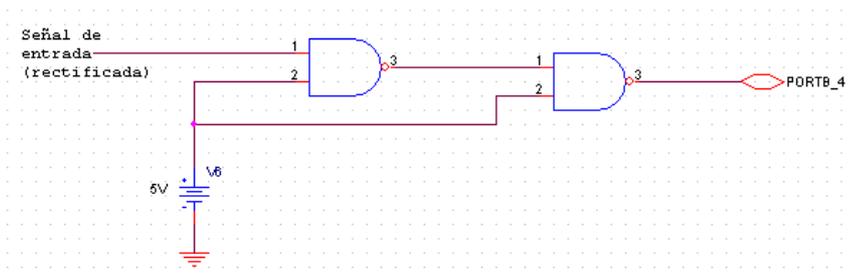
A continuación se muestra una simulación en Orcad de la señal rectificada. (color Azul)



2.3.5–Schmitt trigger (disparador)

La señal alterna convertida a un pulso entre 0 y 5 volts debe ingresar como entrada a un schmitt trigger, de manera de obtener valores acordes a la lógica TTL utilizada por el PORTB\_4 en modo interrupt-on-change.

En el supuesto de que no se coloque el disparador, un silbido de 1khz produciría un tren de pulsos que harían que el PORTB\_4 sea interrumpido a cada instante dado los flancos ascendentes y descendentes que lo componen. Como se quiere generar un único pulso cada vez que se genere un silbido, debe colocarse un dispositivo que discretice esos pulsos. De manera que cuando la tensión de entrada sea menor que un umbral de referencia, la tensión a la salida del mismo sea 0 volt, y 5 volt en caso contrario. El dispositivo utilizado es el CD4093 disparador 2 entradas NAND.



Compuerta NAND de 2 entradas:

A	B	X
0	0	1
0	1	1
1	0	1
1	1	0

En la siguiente figura se muestra la hoja principal de especificaciones del CD4093:

# MC14093B

## Quad 2-Input "NAND" Schmitt Trigger

The MC14093B Schmitt trigger is constructed with MOS P-channel and N-channel enhancement mode devices in a single monolithic structure. These devices find primary use where low power dissipation and/or high noise immunity is desired. The MC14093B may be used in place of the MC14011B quad 2-input NAND gate for enhanced noise immunity or to "square up" slowly changing waveforms.

### Features

- Supply Voltage Range = 3.0 Vdc to 18 Vdc
- Capable of Driving Two Low-Power TTL Loads or One Low-Power Schottky TTL Load Over the Rated Temperature Range
- Triple Diode Protection on All Inputs
- Pin-for-Pin Compatible with CD4093
- Can be Used to Replace MC14011B
- Independent Schmitt-Trigger at each Input
- Pb-Free Packages are Available

### MAXIMUM RATINGS (Voltages Referenced to V<sub>SS</sub>)

Symbol	Parameter	Value	Unit
V <sub>DD</sub>	DC Supply Voltage Range	-0.5 to +18.0	V
V <sub>in</sub> , V <sub>out</sub>	Input or Output Voltage Range (DC or Transient)	-0.5 to V <sub>DD</sub> + 0.5	V
I <sub>in</sub> , I <sub>out</sub>	Input or Output Current (DC or Transient) per Pin	±10	mA
P <sub>D</sub>	Power Dissipation, per Package (Note 1)	500	mW
T <sub>A</sub>	Ambient Temperature Range	-55 to +125	°C
T <sub>stg</sub>	Storage Temperature Range	-65 to +150	°C
T <sub>L</sub>	Lead Temperature (8-Second Soldering)	260	°C

Stresses exceeding Maximum Ratings may damage the device. Maximum Ratings are stress ratings only. Functional operation above the Recommended Operating Conditions is not implied. Extended exposure to stresses above the Recommended Operating Conditions may affect device reliability.

#### 1. Temperature Derating:

Plastic "P and D/DW" Packages: - 7.0 mW/°C From 65° C To 125° C

This device contains protection circuitry to guard against damage due to high static voltages or electric fields. However, precautions must be taken to avoid applications of any voltage higher than maximum rated voltages to this high-impedance circuit. For proper operation, V<sub>in</sub> and V<sub>out</sub> should be constrained to the range V<sub>SS</sub> ≤ (V<sub>in</sub> or V<sub>out</sub>) ≤ V<sub>DD</sub>.

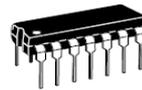
Unused inputs must always be tied to an appropriate logic voltage level (e.g., either V<sub>SS</sub> or V<sub>DD</sub>). Unused outputs must be left open.



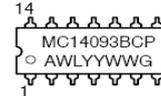
ON Semiconductor®

http://onsemi.com

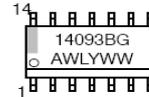
### MARKING DIAGRAMS



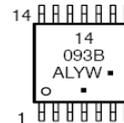
PDIP-14  
P SUFFIX  
CASE 646



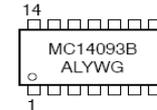
SOIC-14  
D SUFFIX  
CASE 751A



TSSOP-14  
DT SUFFIX  
CASE 948G



SOEIAJ-14  
F SUFFIX  
CASE 965



- A = Assembly Location
- WL, L = Wafer Lot
- YY, Y = Year
- WW, W = Work Week
- G or ■ = Pb-Free Package

(Note: Microdot may be in either location)

### ORDERING INFORMATION

See detailed ordering and shipping information in the package dimensions section on page 2 of this data sheet.

El PORTB\_4 se seteo como entrada en modo Interrupt-on-change y presenta un gran impedancia para la señal en cuestión. La programación del mismo se pensó de manera de detectar solamente el flanco ascendente (1er flanco) del pulso. O sea se pretende ignorar el flanco descendente en cuanto a acciones a tomar, solo se tomaran acciones generadas por la detección del flanco ascendente.

# PIC18FXX2

## 8.6 INT0 Interrupt

External interrupts on the RB0/INT0, RB1/INT1 and RB2/INT2 pins are edge triggered: either rising, if the corresponding INTEDGx bit is set in the INTCON2 register, or falling, if the INTEDGx bit is clear. When a valid edge appears on the RBx/INTx pin, the corresponding flag bit INTxF is set. This interrupt can be disabled by clearing the corresponding enable bit INTxE. Flag bit INTxF must be cleared in software in the Interrupt Service Routine before re-enabling the interrupt. All external interrupts (INT0, INT1 and INT2) can wake-up the processor from SLEEP, if bit INTxE was set prior to going into SLEEP. If the global interrupt enable bit GIE is set, the processor will branch to the interrupt vector following wake-up.

Interrupt priority for INT1 and INT2 is determined by the value contained in the interrupt priority bits, INT1IP (INTCON3<6>) and INT2IP (INTCON3<7>). There is no priority bit associated with INT0. It is always a high priority interrupt source.

## 8.7 TMR0 Interrupt

In 8-bit mode (which is the default), an overflow (FFh → 00h) in the TMR0 register will set flag bit TMR0IF. In 16-bit mode, an overflow (FFFFh → 0000h) in the TMR0H:TMR0L registers will set flag bit TMR0IF. The interrupt can be enabled/disabled by setting/clearing enable bit TOIE (INTCON<5>). Interrupt priority for Timer0 is determined by the value contained in the interrupt priority bit TMR0IP (INTCON2<2>). See Section 10.0 for further details on the Timer0 module.

## 8.8 PORTB Interrupt-on-Change

An input change on PORTB<7:4> sets flag bit RBIF (INTCON<0>). The interrupt can be enabled/disabled by setting/clearing enable bit, RBIE (INTCON<3>). Interrupt priority for PORTB interrupt-on-change is determined by the value contained in the interrupt priority bit, RBIP (INTCON2<0>).

## 8.9 Context Saving During Interrupts

During an interrupt, the return PC value is saved on the stack. Additionally, the WREG, STATUS and BSR registers are saved on the fast return stack. If a fast return from interrupt is not used (See Section 4.3), the user may need to save the WREG, STATUS and BSR registers in software. Depending on the user's application, other registers may also need to be saved. Equation 8-1 saves and restores the WREG, STATUS and BSR registers during an Interrupt Service Routine.

### EXAMPLE 8-1: SAVING STATUS, WREG AND BSR REGISTERS IN RAM

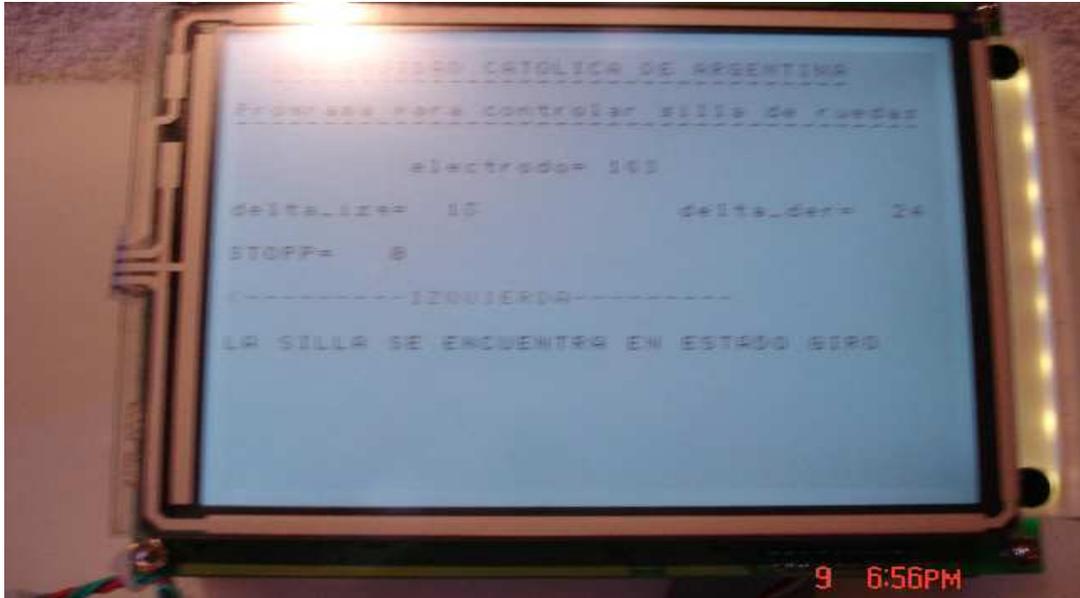
```

MOVWF  W_TEMP                ; W_TEMP is in virtual bank
MOVFF  STATUS, STATUS_TEMP   ; STATUS_TEMP located anywhere
MOVFF  BSR,    BSR_TEMP      ; BSR located anywhere
;
; USER ISR CODE
;
MOVFF  BSR_TEMP, BSR         ; Restore BSR
MOVFF  W_TEMP,  W           ; Restore WREG
MOVFF  STATUS_TEMP, STATUS  ; Restore STATUS

```

## 2.4 – Descripción del microcontrolador

El código de programación consta de una rutina inicial (main) y una interrupción donde se realizan los cambios para comandar la salida (motores de continua). También se muestran algunos resultados en un display Hitachi de 320 x 240.



# PIC18FXX2

## 12.0 TIMER2 MODULE

The Timer2 module timer has the following features:

- 8-bit timer (TMR2 register)
- 8-bit period register (PR2)
- Readable and writable (both registers)
- Software programmable prescaler (1:1, 1:4, 1:16)
- Software programmable postscaler (1:1 to 1:16)
- Interrupt on TMR2 match of PR2
- SSP module optional use of TMR2 output to generate clock shift

Timer2 has a control register shown in Register 12-1. Timer2 can be shut-off by clearing control bit TMR2ON (T2CON<2>) to minimize power consumption. Figure 12-1 is a simplified block diagram of the Timer2 module. Register 12-1 shows the Timer2 control register. The prescaler and postscaler selection of Timer2 are controlled by this register.

## 12.1 Timer2 Operation

Timer2 can be used as the PWM time-base for the PWM mode of the CCP module. The TMR2 register is readable and writable, and is cleared on any device RESET. The input clock ( $F_{osc}/4$ ) has a prescale option of 1:1, 1:4 or 1:16, selected by control bits T2CKPS1:T2CKPS0 (T2CON<1:0>). The match output of TMR2 goes through a 4-bit postscaler (which gives a 1:1 to 1:16 scaling inclusive) to generate a TMR2 interrupt (latched in flag bit TMR2IF, (PIR1<1>)).

The prescaler and postscaler counters are cleared when any of the following occurs:

- a write to the TMR2 register
- a write to the T2CON register
- any device RESET (Power-on Reset,  $\overline{MCLR}$  Reset, Watchdog Timer Reset, or Brown-out Reset)

TMR2 is not cleared when T2CON is written.

### REGISTER 12-1: T2CON: TIMER2 CONTROL REGISTER

U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	TOUTPS3	TOUTPS2	TOUTPS1	TOUTPS0	TMR2ON	T2CKPS1	T2CKPS0
bit 7							bit 0

- bit 7 **Unimplemented:** Read as '0'
- bit 6-3 **TOUTPS3:TOUTPS0:** Timer2 Output Postscale Select bits
  - 0000 = 1:1 Postscale
  - 0001 = 1:2 Postscale
  - 
  - 
  - 
  - 1111 = 1:16 Postscale
- bit 2 **TMR2ON:** Timer2 On bit
  - 1 = Timer2 is on
  - 0 = Timer2 is off
- bit 1-0 **T2CKPS1:T2CKPS0:** Timer2 Clock Prescale Select bits
  - 00 = Prescaler is 1
  - 01 = Prescaler is 4
  - 1x = Prescaler is 16

<b>Legend:</b>			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'	
- n = Value at POR	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown

# PIC18FXX2

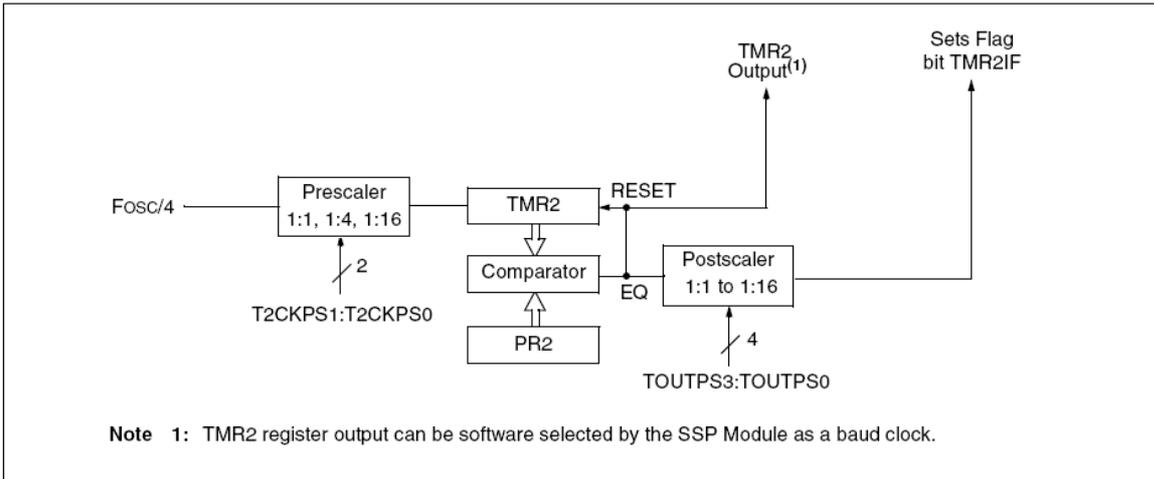
## 12.2 Timer2 Interrupt

The Timer2 module has an 8-bit period register, PR2. Timer2 increments from 00h until it matches PR2 and then resets to 00h on the next increment cycle. PR2 is a readable and writable register. The PR2 register is initialized to FFh upon RESET.

## 12.3 Output of TMR2

The output of TMR2 (before the postscaler) is fed to the Synchronous Serial Port module, which optionally uses it to generate the shift clock.

**FIGURE 12-1: TIMER2 BLOCK DIAGRAM**



**TABLE 12-1: REGISTERS ASSOCIATED WITH TIMER2 AS A TIMER/COUNTER**

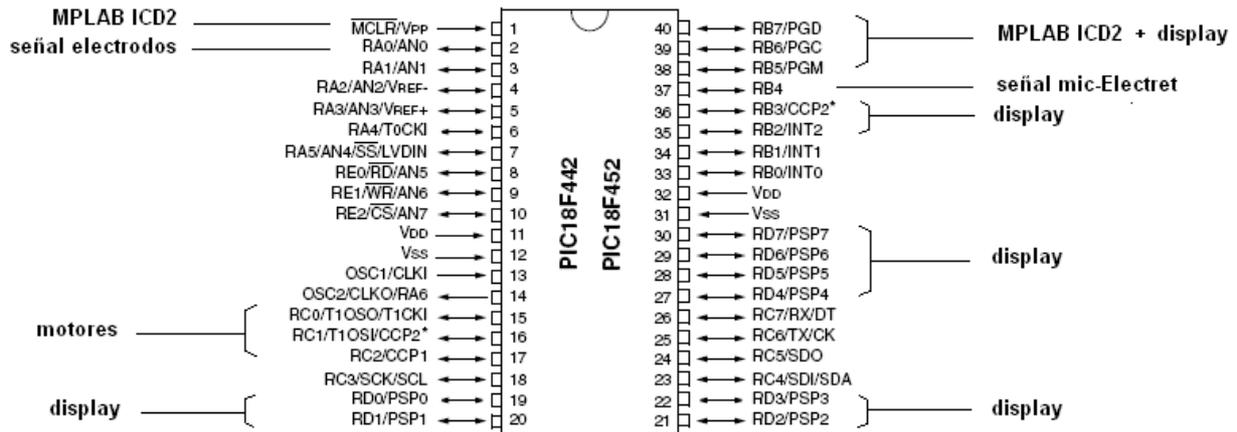
Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on POR, BOR	Value on All Other RESETS
INTCON	GIE/GIEH	PEIE/GIEL	TMR0IE	INT0IE	RBIE	TMR0IF	INT0IF	RBIF	0000 000x	0000 000u
PIR1	PSPIF <sup>(1)</sup>	ADIF	RCIF	TXIF	SSPIF	CCP1IF	TMR2IF	TMR1IF	0000 0000	0000 0000
PIE1	PSPIE <sup>(1)</sup>	ADIE	RCIE	TXIE	SSPIE	CCP1IE	TMR2IE	TMR1IE	0000 0000	0000 0000
IPR1	PSPIP <sup>(1)</sup>	ADIP	RCIP	TXIP	SSPIP	CCP1IP	TMR2IP	TMR1IP	0000 0000	0000 0000
TMR2	Timer2 Module Register								0000 0000	0000 0000
T2CON	—	TOUTPS3	TOUTPS2	TOUTPS1	TOUTPS0	TMR2ON	T2CKPS1	T2CKPS0	-000 0000	-000 0000
PR2	Timer2 Period Register								1111 1111	1111 1111

Legend: x = unknown, u = unchanged, - = unimplemented read as '0'. Shaded cells are not used by the Timer2 module.

**Note 1:** The PSPIF, PSPIE and PSPIP bits are reserved on the PIC18F2X2 devices; always maintain these bits clear.

A continuación se describen las señales de entrada al microcontrolador. La señal proveniente de los electrodos entra por el PORTA\_0 al conversor analógico-digital (ADC) .EL PORTB\_4 recibe un pulso de 5v cuando el mic-Electret capta un silbido. Esto se menciona con mayor detalle en páginas posteriores.

Para programación (ICD2) se utilizan: PORTB\_5, PORTB\_6 y PORTB\_7.  
 Los puertos de salida del microcontrolador se describen a continuación. El PORTC\_0, PORTC\_1 (PWM) y PORTC\_2 (PWM) comandan los motores. Estos últimos dos puertos se utilizan en modo



PWM que es un modulo del PIC18F452 para controlar las revoluciones de los motores de corriente continua, lo cual se explicara luego en forma detallada. Por ultimo los puertos PORTB\_2, PORTB\_3, PORTB\_5, PORTB\_6, PORTB\_7 y todo el PORTD se destina para la conexión de un display.

### 2.5 – Programa en C del microcontrolador

Aquí se incluye el programa completo con comentarios de manera de lograr una mejor comprensión del mismo.

```
//-----Include librerias para pic, display, delays y assembler-----

#include "PIC18F452.h";
#include "asm.h";
#include "delay_soft.h";
#include "d_320_240.h";

//-----Fin include librerias para pic, display, delays y assembler-----

//-----Include de librerias para C-----

#include <stdio.h>
#include <stdlib.h>
#include <math.h>

//-----Fin include de librerias para C-----

//-----Declaración de variables-----

unsigned int
AD[5],electrodo,dis_electr,delta_der,delta_izq,umbral_der,umbral_izq,centro,t,l,STOPP,motores,segunda,derecha,izquie
rda;
unsigned char buffer[200];
```

```

//-----Fin de Declaración de variables-----

void interrupt inter(void)
{
//-----Rutina de interrupt on change-----
    if(RBIF==1)
    {
        PORTB=PORTB;
        RBIE=0;
        RBIF=0;
        if(segunda<2)
        {
            entro a este if estoy en flanco ascendente
                segunda=segunda++;
                RBIE=1;
        }
        else
        {
            permanecer detenido
                if(STOPP==1)
                {
                    STOPP=0;
                    segunda=0;
                    RBIE=1;
                }
            else
            {
                descendente del pulso
                    if(segunda==2)
                    {
                        CCPR1L=0b00000000;
                        CCPR2L=0b00000000;
                        STOPP=1;
                        segunda=0;
                        RBIE=1;
                    }
                }
            }
        }
    }
//-----Fin Rutina de interrupt on change-----

    else
    {
        if(STOPP==1)
        {
            TMR2IF=0;    //limpio flag del timer2
            TMR2ON=0;    //apago timer 2

            CCPR1L=0b00000000;
            CCPR2L=0b00000000;
            //detiene los motores

            PR2=255;
            //sin postcaler lo mas lento que se

            puede hacer es 408useg
            TMR2ON=1;    //disparo timer 2
        }
        else
        {
            TMR2IF=0;    //limpio flag del timer2
            TMR2ON=0;    //apago timer 2
        }
    }
}
//-----Algoritmo de control de motores-----

```

```

        if(dis_electr>=umbral_der)
        {
            derecha=0;
            izquierda=1;
            centro=0;
            delta_izq=dis_electr-130;

            //mover motor izquierdo
            CCPR2L=delta_izq;
            CCPR1L=0;
        }
        if(dis_electr<=umbral_izq)
        {
            if(dis_electr<5)
            {
                dis_electr==1;
            }
            derecha=1;
            izquierda=0;
            centro=0;
            delta_der=130-dis_electr;

            //mover motor derecho
            CCPR1L=delta_der;
            CCPR2L=0;
        }
        if(dis_electr<umbral_der)
        {
            if(dis_electr>umbral_izq)
            {
                derecha=0;
                izquierda=0;
                centro=1;
                CCPR1L=100;

                //detiene los motores
                CCPR2L=100;
            }
        }

        //-----Fin de algoritmo de control de motores-----

        //-----Adquisición de señal electrodos-----
        ADCON0=0b10000001; //AN0; Fosc/64

        t=20;
        while(t--);
        GODONE=1;
        while(GODONE);

        AD[4]=ADRESH;
        AD[4]=AD[4]<<8;
        AD[4]=AD[4]|ADRESL;
        electrodo=AD[4];
        dis_electr=electrodo>>2;

        //-----Fin de Adquisición de señal electrodos-----

        PR2=255; //sin postcaler lo mas lento que

        TMR2ON=1; //disparo timer 2
    }
}

void main(void)

```

```

{
//-----Inicialización de puertos y Variables-----
//-----PUERTOS-----
TRISA=0b11111111;
PORTA=0;
TRISC=0b00000000;
PORTC=0b00000001;           //el bit 0 lo pongo en 1 para habilitar el L293E
TRISB=0b00010000;         //el bit 4 en 1 para interrupt on change (electrec)
PORTB=0b00000000;

//-----VARIABLES-----
AD[5]=0;
buffer[200]=0;
electrodo=0;
dis_electr=0;
umbral_der=170;
umbral_izq=90;
delta_der=0;
delta_izq=0;
l=0;
STOPP=0;
motores=0;
segunda=0;
derecha=0;
centro=0;
izquierda=0;

//-----Fin de Inicialización de puerta y Variables-----

//-----Inicializacion del display-----
delay_soft_ms(400); // importante, esperar que VCC se estabilice !!!
DIS320240_SYS_SET(); // OKI
DIS_SCROLL(); // OKI
DIS_HDOT_SCR();
DIS_OVLAY();

DIS_CSRW();
DIS_CSR_DIR(); // Cursor hacia derecha OKI

DIS_cl_graf(); // limpio display grafico
DIS_cl_text(); // limpio display texto

DIS_CSRW_FORM(); // Tipo de cursor
DIS_ON(); // Display ON

DIS_LED (1);

//-----Fin de inicializacion del display-----

//-----Seteo Canales Analógicos e Interrupciones-----
ADCON1=0b11000000;
ADIF=0;
ADIE=0;

//-----Fin seteo Canales Analógicos e Interrupciones-----

//-----Seteo de interrupciones-----
GIE=1;
PEIE=1;
RBPU=1;
IPEN=1;           //habilitar las prioridades para las interrupciones
RBIP=1;          //alta prioridad para interrupcion por PORTB
RBIF=0;          //limpio flag de interrupcion po PORTB

```

```

RBIE=1; //habilita interrupciones por PORTB

//-----Fin seteo de interrupciones-----

//-----Adquisición de señal electrodos-----
ADCON0=0b10000001; //ANO; Fosc/64

t=20;
while(t--);
GODONE=1;
while(GODONE);

AD[4]=ADRESH;
AD[4]=AD[4]<<8;
AD[4]=AD[4]|ADRESL;
electrodo=AD[4];
dis_electr=electrodo>>2;

//-----Fin de Adquisición de señal electrodos-----

//-----Seteo Timer 2-----

TMR2IF=0;
TMR2IE=1;
T2CON=0b00000011; //postcaler en 1 y prescaler en 16
PR2=255; //sin postcaler lo mas lento que se puede hacer es 408useg

//-----Fin Seteo Timer 2-----

//-----Seteo de PWM-----

CCP1CON=0b11001111; //los menos significativos en 0
CCPR1L=0b00000000; //el duty cycle es 0%

CCP2CON=0b11001111; //los menos significativos en 0
CCPR2L=0b00000000; //el duty cycle es 0%

//-----Fin de seteo de PWM-----

TMR2ON=1; //disparo timer 2

while(1)
{
ei();
sprintf(buffer,"UNIVERSIDAD CATOLICA DE ARGENTINA");
print_at(1,2,buffer);
di();
delay_soft_ms(10);

ei();
sprintf(buffer,"-----");
print_at(2,2,buffer);
di();
delay_soft_ms(10);

ei();
sprintf(buffer,"Programa para controlar silla de ruedas");
print_at(4,0,buffer);
di();
delay_soft_ms(10);

ei();
sprintf(buffer,"-----");
print_at(5,0,buffer);
di();
}

```

```

delay_soft_ms(10);

ei();
sprintf(buffer,"electrodo= %3d",dis_electr);
print_at(8,10,buffer);
di();
delay_soft_ms(10);

ei();
sprintf(buffer,"delta_izq= %3d",delta_izq);
print_at(11,0,buffer);
di();
delay_soft_ms(10);

ei();
sprintf(buffer,"delta_der= %3d",delta_der);
print_at(11,25,buffer);
di();
delay_soft_ms(10);

ei();
sprintf(buffer,"STOPP= %3d",STOPP);
print_at(14,0,buffer);
di();
delay_soft_ms(10);

        if(centro)
        {
                di();
                sprintf(buffer,"                ");
                print_at(17,0,buffer);
                ei();
                delay_soft_ms(1);

        }

        if(derecha)
        {
                di();
                sprintf(buffer,"                ");
                print_at(17,0,buffer);
                ei();
                delay_soft_ms(1);

                di();
                sprintf(buffer,"-----DERECHA----->");
                print_at(17,0,buffer);
                ei();
                delay_soft_ms(1);

        }

        if(izquierda)
        {
                di();
                sprintf(buffer,"                ");
                print_at(17,0,buffer);
                ei();
                delay_soft_ms(1);

                di();
                sprintf(buffer,"<-----IZQUIERDA-----");
                print_at(17,0,buffer);
                ei();
                delay_soft_ms(1);

        }

        if(STOPP)

```

```
    {
        ;
        di();
        sprintf(buffer,"                ");
        print_at(17,0,buffer);
        ei();
        delay_soft_ms(1);

        di();
        sprintf(buffer,"                ");
        print_at(20,0,buffer);
        ei();
        delay_soft_ms(1);

        di();
        sprintf(buffer,"LA SILLA SE ENCUENTRA EN ESTADO PARADO");
        print_at(19,0,buffer);
        ei();
        delay_soft_ms(1);
    }
    else
    {
        if (STOPP==0)

        {

            di();
            sprintf(buffer,"                ");
            print_at(19,0,buffer);
            delay_soft_ms(1);
            ei();

            di();
            sprintf(buffer,"                ");
            print_at(28,0,buffer);
            ei();
            delay_soft_ms(1);

            di();
            sprintf(buffer,"LA SILLA SE ENCUENTRA EN ESTADO GIRO");
            print_at(20,0,buffer);
            delay_soft_ms(1);
            ei();

        }
    }
}
}
//-----Fin de programa Principal-----
```

### 3 – Pruebas en campo

La silla se probó sobre una superficie liza para visualizar el avance sin perturbaciones debidas al contacto, y hacer más fácil el control de giro de la misma.

Antes de comenzar se requirió de una correcta calibración del offset, con la mirada al centro y los motores de continua apagados, hasta obtener valores cercanos a 130 en el display. (en el display se debe visualizar aprox. electrodo=130)

Una vez logrado esto, se encendieron los motores y se realizaron varios giros. Respondió en forma correcta hacia la dirección solicitada, y además al generar el silbido la silla se detuvo sin problemas. Al volver generar el silbido, se retomó el estado anterior para seguir con las pruebas de guiado mediante los electrodos.

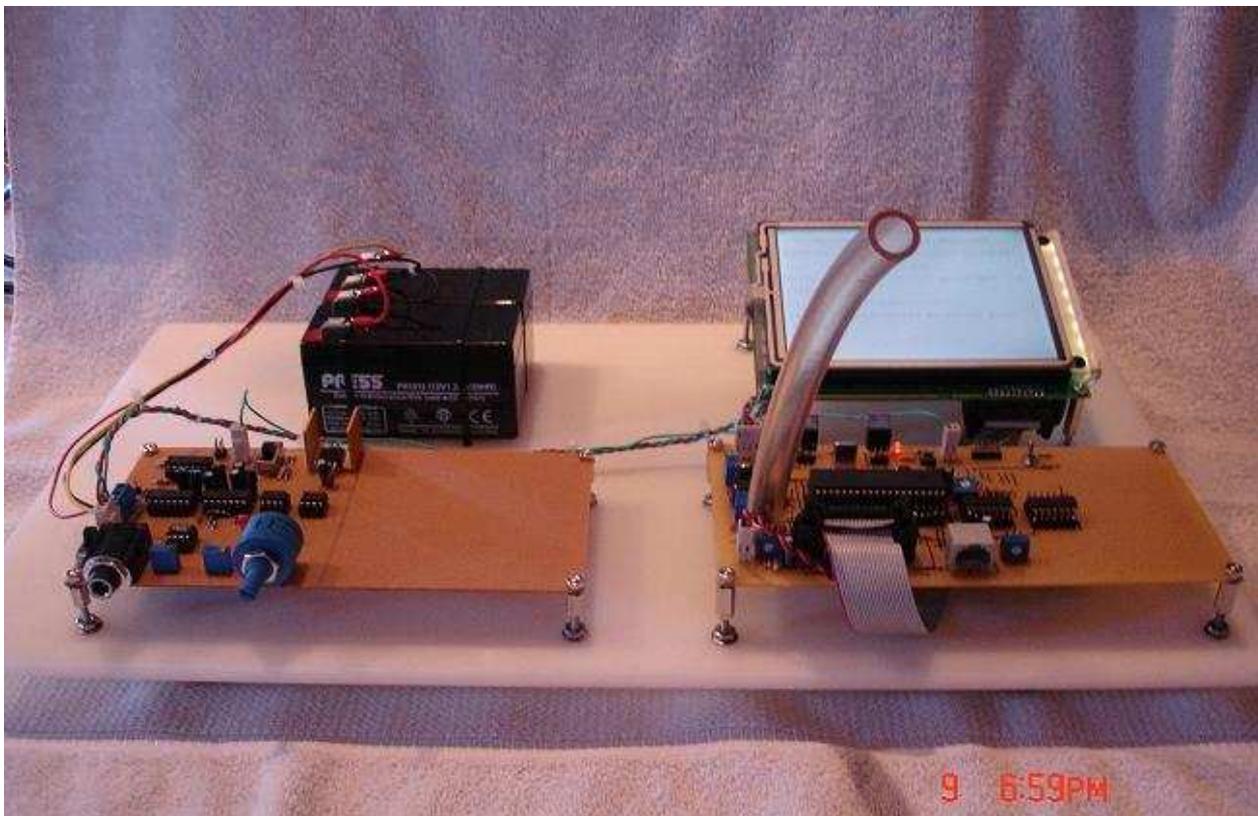
Con respecto al MIC-Electret, se probó recubrirlo con una manguera de goma transparente y que a su vez esté direccionada hacia la boca del operador de manera de lograr una mejor captación. Esto trajo aparejado una mejora de la señal útil de audio, ya que permitió bajar la sensibilidad del Mic-Electret a los valores de amplificación que figuran en este trabajo e ignorar aun más las frecuencias que no resultan de interés.

La carga de las baterías de 12 volt duro aprox. 30 minutos, debido a la fricción, el peso de la silla y la energía consumida por los motores de continua en contacto con el suelo.

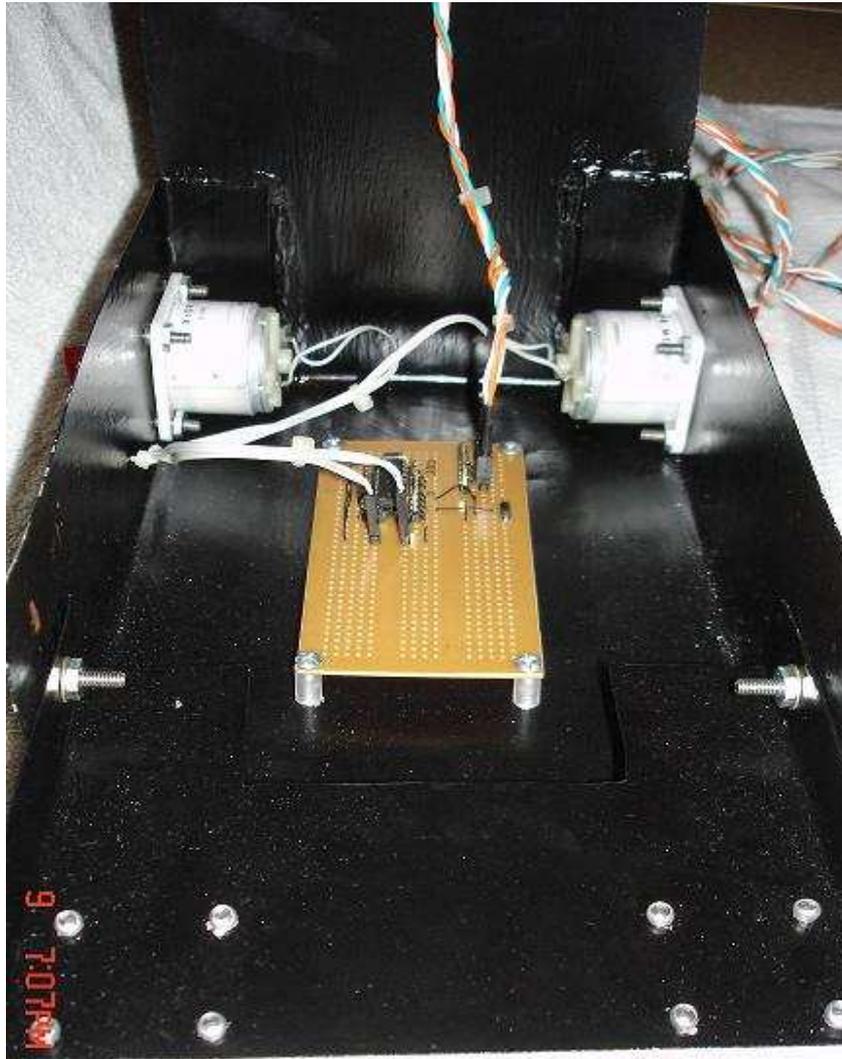
También se agregaron disipadores a los reguladores de tensión en forma preventiva.

### 4 – Fotos del prototipo

Tablero de comandos:



Placa experimental con L293E y motores de continua a bordo de la silla:



## 5 – Conclusiones

El trabajo práctico realizado contiene varios de los temas vistos en la carrera de Ingeniería Electrónica y Comunicaciones. Desde filtros activos y sus cálculos correspondientes hasta programación de un microcontrolador de la serie PIC. Este emprendimiento tuvo los siguientes buenos resultados:

- mayor experiencia práctica de diseño de sistemas electrónico-mecánicos
- mayor práctica en programación de los mismos.
- tener en cuenta diferentes alternativas de implementación para el fin deseado
- armado físico de los circuitos.

Principalmente se ha logrado realizar una aplicación práctica de las señales electrooculograficas se forma relativamente sencilla

Se debe aclarar que si bien la silla funciona correctamente según lo pensado inicialmente, a medida que se fue construyendo también aparecieron nuevos desafíos. Algunos de estos se pudieron resolver fácilmente, por ejemplo, la decisión de emplear un filtro pasabanda que atenúe las frecuencias del PWM dado que de lo contrario traería ruido al Mic-electret. Otros como la necesidad de tener que calibrar los electrodos antes de comenzar a usar la silla, resulta incomodo en la practica y escapan al trabajo practico, lo que no quita que, deberá tenerse en cuenta a la hora de implementarlo en una silla de ruedas de tamaño real.

## 6 – Referencias

[1] <http://es.wikipedia.org>

[2] <http://www.disabilityworld.org/June-July2000/Spanish/Tecnologia/Smithsonian.htm>

[3] [http://www.pcpaudio.com/pcpfiles/doc\\_altavoces/medicionCHE/preampPanasonic/](http://www.pcpaudio.com/pcpfiles/doc_altavoces/medicionCHE/preampPanasonic/)

[4] [http://www.epanorama.net/circuits/microphone\\_powering.html](http://www.epanorama.net/circuits/microphone_powering.html)

[5] [http://www.unicrom.com/Tut\\_filt\\_act\\_PasoBanda.asp](http://www.unicrom.com/Tut_filt_act_PasoBanda.asp)

[6] [http://www.disabilityworld.org/09-10\\_02/spanish/acceso/ergonomics.shtml](http://www.disabilityworld.org/09-10_02/spanish/acceso/ergonomics.shtml)

[7] [http://www.integrando.org.ar/investigando/silla\\_ibot.htm](http://www.integrando.org.ar/investigando/silla_ibot.htm)

[8] Mark Palmer : AN566, "Using the PORTB Interrupt on Change as an External Interrupt", Nota de aplicacion Microchip,1997

[9] R. Barea, L. Boquete, M. Mazo, E. López, L.M. Bergasa : "Aplicación de electrooculografía para ayuda a minusválidos", Universidad de Alcalá, Madrid.

[10] Microchip Technology Inc.: "Manual PIC18FXX2", 2002

[11] Hitech Software: "Manual PICC18 ansi C compiler", 2005

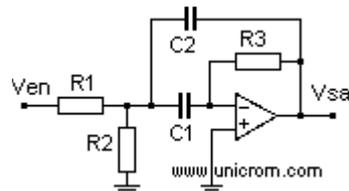
[12] Texas Instruments, "Manual TL084CN J-fet input operacional amplifier", Febrero 1977 – revisado en Septiembre, 2004

[13] Semiconductor Components Industries, "Manual MC14093B", Octubre, 2006

## 7 – Anexos

- Calculo del filtro pasabanda

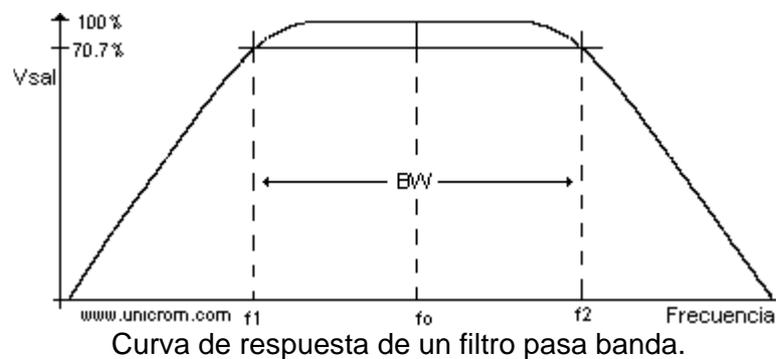
Los filtros activos se diferencian de los filtros comunes, en que estos últimos son sólo combinación de resistencias, capacitores e inductores. En un filtro común, la salida es de menor magnitud que la entrada. En cambio los filtros activos se componen de resistores, capacitores y dispositivos activos como Amplificadores Operacionales o transistores. En un filtro activo la salida puede ser de igual o de mayor magnitud que la entrada.



Filtro activo pasa banda con Amplificador Operacional

El filtro Pasa Banda tiene la siguiente curva de respuesta de frecuencia. Dejará pasar todas las tensiones de la señal de entrada que tengan frecuencias entre la frecuencia de corte inferior  $f_1$  y la de corte superior  $f_2$ . Las tensiones fuera de este rango de frecuencias serán atenuadas y serán menores al 70.7 % de la tensión de entrada. La frecuencia central de este tipo de filtro se obtiene con la siguiente fórmula:

$$f_0 = 1 / [ 2\pi C \times (R_3 R)^{1/2} ]$$



Si se seleccionan los capacitores y resistores de modo que:

$$C_1 = C_2 = C \text{ y } R_1 = R_2 = R$$

El ancho de banda será:  $BW = f_2 - f_1 = 1.41 R / [ C R_3 (R_3 R)^{1/2} ]$

El factor de calidad  $Q = f_0 / BW$ .

Las líneas discontinuas verticales sobre  $f_1$  y  $f_2$  y la línea horizontal del 70.7% representan la respuesta de un filtro pasa banda ideal.

Nota:  $f_1$  y  $f_2$  (frecuencias de corte) son puntos en la curva de transferencia en que salida ha caído 3 dB (decibeles) desde su valor máximo.

- Ingeniería biomédica

La ingeniería biomédica es la ingeniería que se dedica al diseño y construcción de equipos médicos, prótesis, dispositivos médicos, dispositivos de diagnóstico (imagenología médica) y de terapia. También interviene en la gestión o administración de los recursos técnicos ligados a un sistema de hospitales. Combina la experiencia de la ingeniería con necesidades médicas para obtener beneficios en el cuidado de la salud. Puede ser definida como "la aplicación de técnicas y principios de ingeniería al campo médico".

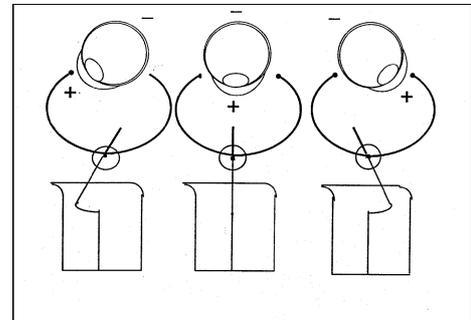
*El inicio de la Ingeniería Biomédica:* en sus inicios, esta disciplina estuvo ligada fundamentalmente a la aplicación de técnicas de ingeniería eléctrica y electrónica para la construcción de equipos médicos (instrumentación médica), así como al diseño de prótesis y ortesis (biomecánica y rehabilitación). Posteriormente, una parte muy importante de las aplicaciones de la ingeniería a la medicina fue la instrumentación para la adquisición de imágenes del cuerpo humano (imagenología médica). A partir del desarrollo de los ordenadores, la importancia de la instrumentación fue disminuyendo, mientras que el procesamiento de las señales adquiridas cobró mayor ímpetu debido a que fue posible obtener información adicional a partir de las señales que la instrumentación proporcionaba, y que no era visible directamente a partir de los trazos puros (procesamiento de señales biomédicas). En la actualidad la disciplina está ligada también a otras como la genómica y proteómica (biología computacional).

Hay autores que indican que existe la ingeniería biomédica desde que se aplicaron remedios a problemas particulares del individuo como una prótesis del dedo gordo del pie que fue descubierta en una tumba egipcia con una antigüedad de más de 3000 años. Otros autores mencionan a los dibujos anatómicos de Leonardo Da Vinci y sus aproximaciones a brazos de palanca o los trabajos de Luigi Galvani y de Lord Kelvin sobre la conducción eléctrica en los seres vivos. Sin embargo, el desarrollo de la instrumentación eléctrica y electrónica produjo una explosión de resultados y se puede considerar como uno de los orígenes más cercanos de la ingeniería biomédica. Esto se da principalmente entre los años de 1890 y 1930. Ejemplos de esto son los diseños para el registro de señales electrofisiológicas, comenzando por los registros de A.D. Waller en corazones de humanos (1887), el refinamiento de la técnica por parte de W. Einthoven al desarrollar un galvanómetro de cuerda (1901) y la aplicación de este al registro de señales electroencefalográficas en humanos por Berger (1924). La instrumentación electrónica a partir de tubos de vacío se empleó por E. Lovett Garceau para amplificar estas señales eléctricas y el primer sistema de electroencefalógrafo comercial de tres canales fue construido por Alfred Grass en 1935. Otro ejemplo es el desarrollo de la instrumentación en imagenología. Desde el descubrimiento de los rayos-X por Röntgen en 1895 hasta su primera aplicación en biomedicina pasó una semana. Desde 1896, Siemens y General Electric ya vendían estos sistemas. En la actualidad, los nuevos desarrollos en imagenología han tomado mucho más tiempo en lograr su aplicación clínica. El principio de resonancia magnética se descubrió en 1946, pero no fue sino hasta 30 años después, que se pudo desarrollar un sistema para uso en humanos.

- Electrooculografía

La retina tiene, en condiciones basales, un potencial bioeléctrico de reposo, de carácter electronegativo respecto a la córnea. De este modo, los giros del globo ocular provocan cambios en la dirección del vector correspondiente a este dipolo eléctrico. La electrooculografía se basa en el registro neurofisiológico de este fenómeno, en condiciones de oscuridad (escotópicas) y de luminosidad.

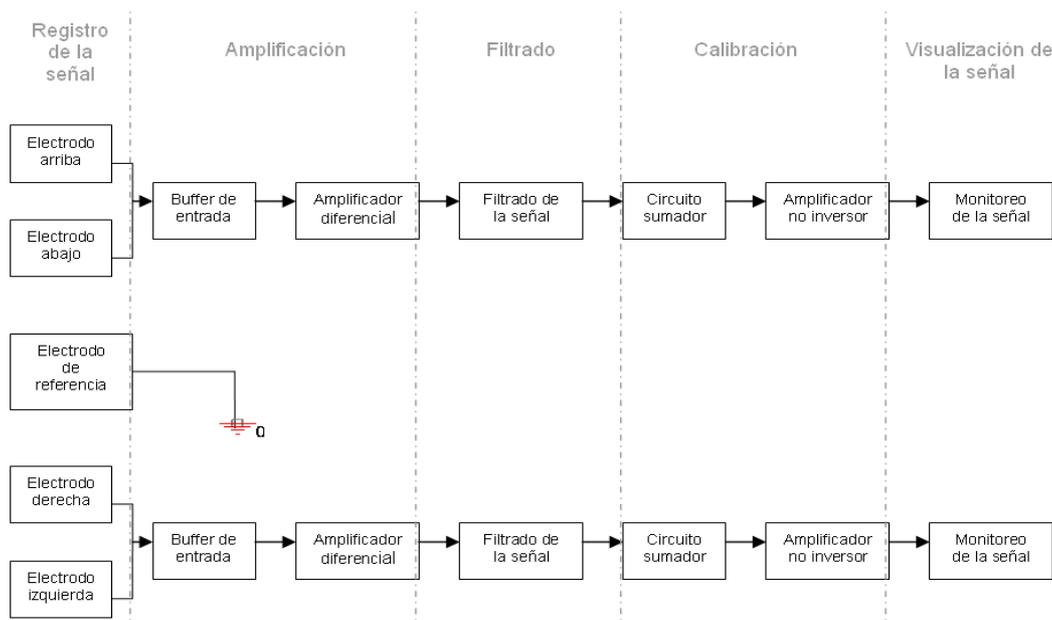
Observamos cómo en la retina normal, la amplitud de la respuesta en condiciones escotópicas es aproximadamente el doble que en condiciones fotópicas. A la amplitud de la respuesta en condiciones escotópicas sobre la amplitud en condiciones fotópicas, se denomina índice de Arden. El electrooculograma refleja la actividad bioeléctrica del epitelio pigmentario.



Registro de las variaciones del voltaje de reposo de la retina durante los giros oculares a uno y otro lado, en condiciones escotópicas y fotópicas. Se registra el electrooculograma durante 12 minutos en condiciones escotópicas y a continuación durante otros 12 minutos en condiciones fotópicas. En condiciones fotópicas, la amplitud de la respuesta es mayor que en condiciones escotópicas. El índice de Arden es la proporción entre la amplitud en condiciones fotópicas y la amplitud en condiciones escotópicas. Se considera netamente patológico un índice inferior a 1.65

- Circuito 1

Diagrama en bloques del diseño:



Para más información recurrir al Trabajo Final realizado por el alumno Juan José Oubiña Agosto del 2007, Carrera de Ingeniería Electrónica y Comunicaciones, Universidad Católica Argentina.

- Control por Ancho de Pulso (PWM)

La Regulación por Ancho de Pulso (o PWM, *pulse-width modulation* en inglés) de un motor de CC está basada en el hecho de que si se recorta la CC de alimentación en forma de una onda cuadrada, la energía que recibe el motor disminuirá de manera proporcional a la relación entre la parte alta (habilita corriente) y baja (cero corriente) del ciclo de la onda cuadrada. Controlando esta relación se logra variar la velocidad del motor de una manera bastante aceptable.

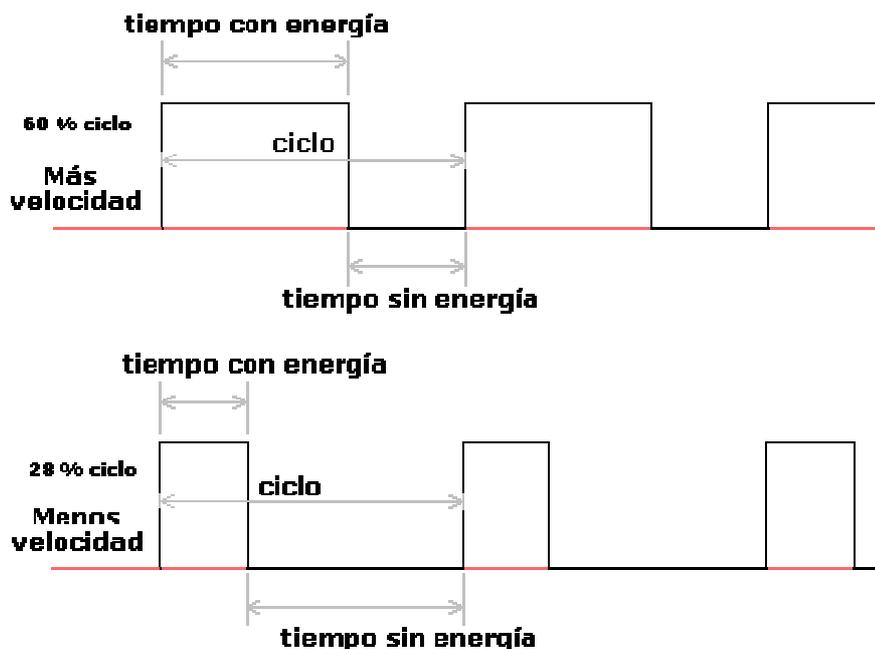
El ciclo de trabajo de una señal periódica es el ancho relativo de su parte positiva en relación al período. Matemáticamente:

$D$  es el ciclo de trabajo  
 $\tau$  es el tiempo en que la función es positiva (ancho del pulso)  
 $T$  es el período de la función.

$$D = \frac{\tau}{T}$$

Algunos parámetros importantes de un PWM son:

- La relación de amplitudes entre la señal portadora y la moduladora, siendo recomendable que la última no supere el valor pico de la portadora y esté centrada en el valor medio de ésta.
- La relación de frecuencias, donde en general se recomienda que la relación entre la frecuencia de la portadora y la de señal sea de 10 a 1.



A continuación se cita el seteo del PWM para el PIC185452.

# PIC18FXX2

## 14.5 PWM Mode

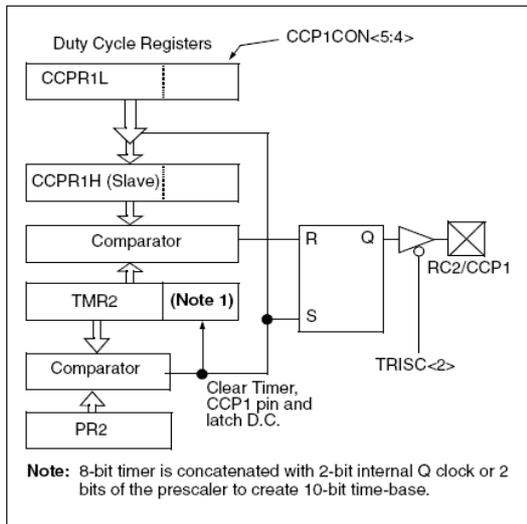
In Pulse Width Modulation (PWM) mode, the CCP1 pin produces up to a 10-bit resolution PWM output. Since the CCP1 pin is multiplexed with the PORTC data latch, the TRISC<2> bit must be cleared to make the CCP1 pin an output.

**Note:** Clearing the CCP1CON register will force the CCP1 PWM output latch to the default low level. This is not the PORTC I/O data latch.

Figure 14-3 shows a simplified block diagram of the CCP module in PWM mode.

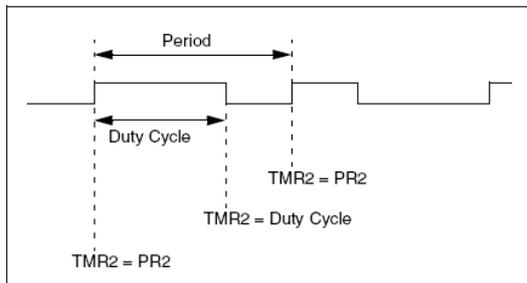
For a step-by-step procedure on how to set up the CCP module for PWM operation, see Section 14.5.3.

**FIGURE 14-3: SIMPLIFIED PWM BLOCK DIAGRAM**



A PWM output (Figure 14-4) has a time-base (period) and a time that the output stays high (duty cycle). The frequency of the PWM is the inverse of the period (1/period).

**FIGURE 14-4: PWM OUTPUT**



### 14.5.1 PWM PERIOD

The PWM period is specified by writing to the PR2 register. The PWM period can be calculated using the following formula:

$$\text{PWM period} = (\text{PR2} + 1) \cdot 4 \cdot \text{Tosc} \cdot (\text{TMR2 prescale value})$$

PWM frequency is defined as  $1 / [\text{PWM period}]$ .

When TMR2 is equal to PR2, the following three events occur on the next increment cycle:

- TMR2 is cleared
- The CCP1 pin is set (exception: if PWM duty cycle = 0%, the CCP1 pin will not be set)
- The PWM duty cycle is latched from CCPR1L into CCPR1H

**Note:** The Timer2 postscaler (see Section 12.0) is not used in the determination of the PWM frequency. The postscaler could be used to have a servo update rate at a different frequency than the PWM output.

### 14.5.2 PWM DUTY CYCLE

The PWM duty cycle is specified by writing to the CCPR1L register and to the CCP1CON<5:4> bits. Up to 10-bit resolution is available. The CCPR1L contains the eight MSBs and the CCP1CON<5:4> contains the two LSBs. This 10-bit value is represented by CCPR1L:CCP1CON<5:4>. The following equation is used to calculate the PWM duty cycle in time:

$$\text{PWM duty cycle} = (\text{CCPR1L:CCP1CON<5:4>}) \cdot \text{Tosc} \cdot (\text{TMR2 prescale value})$$

CCPR1L and CCP1CON<5:4> can be written to at any time, but the duty cycle value is not latched into CCPR1H until after a match between PR2 and TMR2 occurs (i.e., the period is complete). In PWM mode, CCPR1H is a read only register.

The CCPR1H register and a 2-bit internal latch are used to double buffer the PWM duty cycle. This double buffering is essential for glitchless PWM operation.

When the CCPR1H and 2-bit latch match TMR2 concatenated with an internal 2-bit Q clock or 2 bits of the TMR2 prescaler, the CCP1 pin is cleared.

The maximum PWM resolution (bits) for a given PWM frequency is given by the equation:

$$\text{PWM Resolution (max)} = \frac{\log\left(\frac{F_{\text{OSC}}}{F_{\text{PWM}}}\right)}{\log(2)} \text{ bits}$$

**Note:** If the PWM duty cycle value is longer than the PWM period, the CCP1 pin will not be cleared.

# PIC18FXX2

### 14.5.3 SETUP FOR PWM OPERATION

The following steps should be taken when configuring the CCP module for PWM operation:

1. Set the PWM period by writing to the PR2 register.
2. Set the PWM duty cycle by writing to the CCPR1L register and CCP1CON<5:4> bits.
3. Make the CCP1 pin an output by clearing the TRISC<2> bit.
4. Set the TMR2 prescale value and enable Timer2 by writing to T2CON.
5. Configure the CCP1 module for PWM operation.

**TABLE 14-4: EXAMPLE PWM FREQUENCIES AND RESOLUTIONS AT 40 MHz**

PWM Frequency	2.44 kHz	9.77 kHz	39.06 kHz	156.25 kHz	312.50 kHz	416.67 kHz
Timer Prescaler (1, 4, 16)	16	4	1	1	1	1
PR2 Value	0xFF	0xFF	0xFF	0x3F	0x1F	0x17
Maximum Resolution (bits)	14	12	10	8	7	6.58

**TABLE 14-5: REGISTERS ASSOCIATED WITH PWM AND TIMER2**

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on POR, BOR	Value on All Other RESETS
INTCON	GIE/GIEH	PEIE/GIEL	TMR0IE	INT0IE	RBIE	TMR0IF	INT0IF	RBF	0000 000x	0000 000u
PIR1	PSPIF <sup>(1)</sup>	ADIF	RCIF	TXIF	SSPIF	CCP1IF	TMR2IF	TMR1IF	0000 0000	0000 0000
PIE1	PSPIE <sup>(1)</sup>	ADIE	RCIE	TXIE	SSPIE	CCP1IE	TMR2IE	TMR1IE	0000 0000	0000 0000
IPR1	PSPIP <sup>(1)</sup>	ADIP	RCIP	TXIP	SSPIP	CCP1IP	TMR2IP	TMR1IP	0000 0000	0000 0000
TRISC	PORTC Data Direction Register								1111 1111	1111 1111
TMR2	Timer2 Module Register								0000 0000	0000 0000
PR2	Timer2 Module Period Register								1111 1111	1111 1111
T2CON	—	TOUTPS3	TOUTPS2	TOUTPS1	TOUTPS0	TMR2ON	T2CKPS1	T2CKPS0	-000 0000	-000 0000
CCPR1L	Capture/Compare/PWM Register1 (LSB)								xxxx xxxx	uuuu uuuu
CCPR1H	Capture/Compare/PWM Register1 (MSB)								xxxx xxxx	uuuu uuuu
CCP1CON	—	—	DC1B1	DC1B0	CCP1M3	CCP1M2	CCP1M1	CCP1M0	--00 0000	--00 0000
CCPR2L	Capture/Compare/PWM Register2 (LSB)								xxxx xxxx	uuuu uuuu
CCPR2H	Capture/Compare/PWM Register2 (MSB)								xxxx xxxx	uuuu uuuu
CCP2CON	—	—	DC2B1	DC2B0	CCP2M3	CCP2M2	CCP2M1	CCP2M0	--00 0000	--00 0000

Legend: x = unknown, u = unchanged, - = unimplemented, read as '0'. Shaded cells are not used by PWM and Timer2.

**Note 1:** The PSPIF, PSPIE and PSPIP bits are reserved on the PIC18F2X2 devices; always maintain these bits clear.