

UNIVERSIDAD DE EL SALVADOR
FACULTAD DE INGENIERIA Y ARQUITECTURA
ESCUELA DE INGENIERIA ELECTRICA



“Propuesta de diseño y construcción de un controlador de mando tipo Mouse (touch pad) para una silla de ruedas”

PRESENTADO POR:

VICTOR BARRERA CUBIAS

PARA OPTAR AL TITULO DE:

INGENIERO ELECTRICISTA

CIUDAD UNIVERSITARIA, OCTUBRE DE 2006

UNIVERSIDAD DE EL SALVADOR
FACULTAD DE INGENIERIA Y ARQUITECTURA
ESCUELA DE INGENIERIA ELECTRICA

Trabajo de Graduación previo a la opción al Grado de:

INGENIERO ELECTRICISTA

Título :

“Propuesta de diseño y construcción de un controlador de mando tipo Mouse (touch pad) para una silla de ruedas”

Presentado por :

VICTOR BARRERA CUBIAS

Trabajo de Graduación Aprobado por:

Docente Director :
Ing. Hugo Miguel Colato Rodríguez

San Salvador, octubre de 2006

UNIVERSIDAD DE EL SALVADOR

RECTORA :

DRA. MARÍA ISABEL RODRÍGUEZ

SECRETARIA GENERAL :

LICDA. ALICIA MARGARITA RIVAS DE RECINOS

FACULTAD DE INGENIERIA Y ARQUITECTURA

DECANO :

ING. MARIO ROBERTO NIETO LOVO

SECRETARIO :

ING. OSCAR EDUARDO MARROQUÍN HERNÁNDEZ

ESCUELA DE INGENIERIA ELECTRICA

DIRECTOR :

ING. LUIS ROBERTO CHÉVEZ PAZ

Trabajo de Graduación Aprobado por:

Docente Director :

Ing. Hugo Miguel Colato Rodríguez

PREFACIO.

Las sillas manuales son impulsadas por dos aros colocados en las ruedas principales de la silla, en algunos casos, debido a la postura adoptada o al sobreesfuerzo hecho para su propulsión puede provocar sobrecargas musculares que llevarían a reacciones anómalas asociadas y a favorecer deformidades.

En cambio una silla de ruedas eléctrica proporciona autonomía de movimiento al usuario, ya que hace uso de dos motores eléctricos para mover la silla de un lugar a otro, pero su alto costo la hace inaccesible para la mayor parte de la población con impedimentos físicos.

No obstante el nivel de incapacidad motriz en muchas personas es de tal grado o magnitud que les impide poder manipular el “joystick” utilizado para maniobrar la silla de ruedas eléctrica, por tal motivo el presente trabajo propone diseñar e implementar un controlador de dirección y movimiento para silla de ruedas que utilice un touchpad como entrada de mando. También el diseño del controlador contempla el uso de sensores medidores de distancia para detectar posibles objetos en la trayectoria de recorrido de la silla de ruedas.

La construcción del controlador localmente en el país, bajaría los costos de adquisición de la misma y la utilización de un “touchpad” como entrada de mando, beneficiaría a personas que le es imposible utilizar el “joystick” tradicional de las sillas eléctricas comerciales, ya que el esfuerzo físico realizado es mínimo, comparado con el empleado en el uso del “joystick”.

Además, ante la falta de trabajos de investigación en la rama de la Ingeniería Eléctrica, orientados a resolver algunas necesidades de personas con discapacidades físicas, se hace necesario dar un aporte a esta parte de la población salvadoreña.

Finalmente se comprobó la validez en el diseño y construcción del controlador, fabricando un prototipo mecánico para poder demostrar la operación del controlador diseñado.

RESUMEN DEL TRABAJO.

En este documento se presenta el diseño y construcción de un controlador de dirección y movimiento para una silla de ruedas, con la principal característica de tener como entrada de mando un touchpad. Con el “touchpad” el usuario podrá realizar cinco tipos de movimientos: avanzar, retroceder, giro izquierdo, giro derecho y avance de recorrido largo. El controlador contará con dos sensores medidores de distancia, uno de los cuales estará colocado al frente y otro en la parte posterior del prototipo construido, con esto el controlador será capaz de tomar decisiones en lo referente a reducción de velocidad cuando detecte un obstáculo o incremento de velocidad cuando no lo haya, todo esto sin la intervención del usuario.

El capítulo I, trata sobre el diseño del controlador de manera general, para establecer las bases sobre las cuales se desarrolle tanto el hardware como el software. Se especifican las entradas y salidas que tendrá el controlador, la operación de cada una de las etapas que lo conforman y como se interrelacionan unas con otras, también se hace una descripción de los movimientos que el usuario de la silla de ruedas debe hacer sobre un touchpad para mover la silla y luego se examinan los posibles casos de uso de la silla de ruedas, esto nos permitirá definir el comportamiento de la silla y el usuario ante diferentes situaciones.

El capítulo II, Se realiza la selección de los componentes electrónicos utilizados en el diseño y su principio de operación. Además se presenta la utilización de los módulos del microcontrolador que se utilizaran en el diseño del controlador y finalmente se presenta el diseño completo del circuito realizado.

En el capítulo III, se presenta el diseño de un programa desarrollado en lenguaje ensamblador, el cual implementa el protocolo PS/2² para leer las coordenadas de movimiento enviadas por el touchpad, esta información será utilizada para tomar decisiones de control de velocidad, dirección y que sensor leer, todo esto desarrollado en base a las especificaciones de los casos de usos.

² PS/2: Protocolo de comunicación utilizado por algunos dispositivos de apuntamiento y desarrollado por IBM.

TABLA DE CONTENIDOS

Capítulo	Página
Prefacio	i
Resumen del trabajo	ii
Lista de tablas	vi
Lista de figuras	vii
I DISEÑO DEL CONTROLADOR	1
INTRODUCCION	1
1.1.0 Planteamiento del problema	1
1.1.1 Definición de entradas y salidas del controlador	1
1.1.2 Habilidad de sensores medidores de distancia	2
1.1.3 Determinación de la distancia hasta un	3
obstáculo y su relación con la modificación PWM.	
1.1.4 Detección de usuario.....	3
1.1.5 Inhabilitación del controlador	3
1.1.6 Reinicialización del controlador	4
1.1.7 Señales de control de velocidad y de dirección	4
1.1.7.1 Modulación de ancho de pulso	4
1.2.0 Diagrama de bloques	6
1.2.1 Etapa de control	6
1.2.2 Etapa de entrada de mando	8
1.2.2.1 Principio de operación del touchpad	8
1.2.2.2 Reporte de datos del touchpad	9
1.2.2.2.1 Modo relativo	9
1.2.2.2.2 Modo absoluto	10
1.2.3 Etapa de medición de distancias	11
1.2.4 Etapa de alimentación	11
1.3.0 Acciones definidas del usuario sobre el touchpad	12
para controlar el movimiento de la silla	
1.3.1 Avance para desplazamiento corto	12
1.3.2 Avance para desplazamiento largo	13
1.3.3 Retroceso para desplazamiento corto	13
1.3.4 Retroceso para desplazamiento largo	14
1.3.5 Giro izquierdo	14
1.3.6 Giro derecho	15
1.3.7 Movimientos no validos	15
1.4.0 Casos de usos de la silla de ruedas	15
1.4.1 Caso de uso avance y sus escenarios	16
1.4.2 Caso de uso retroceso y sus escenario	18
1.4.3 Caso de uso giro hacia la izquierda y su escenario	19
1.4.4 Caso de uso giro hacia la derecha y su escenario	21
1.4.5 Caso de uso detección de usuario y sus escenarios.....	23
CONCLUSIONES DEL CAPITULO I	25

REFERENCIAS BIBLIOGRAFICAS	26
II. DISEÑO E IMPLEMENTACION DEL HARDWARE	27
2.1.0 Etapa de control	27
2.1.1 Selección del microcontrolador	27
2.1.2 Funcionamiento y programación de los módulos Timer2 y CCP (en modo PWM)	28
2.1.2.1 Pasos necesarios para configuración y operación PWM del modulo CCP1..	28
2.1.2.2 Cálculos realizados referentes a los módulos CCP	31
2.1.3 Detalle de conexión de las salidas de control de dirección y velocidad	31
2.2.0 Etapa de entrada de mando	32
2.2.1 Selección del touchpad	32
2.2.2 Características del touchpad Alps modelo 56AAA1965B	33
2.2.3 Tipo de transmisión /recepción.....	34
2.2.3.1 Recurso de interrupción por flanco	35
2.3.0 Etapa de medición de distancias	36
2.3.1 Selección del sensor medidor de distancias por infrarrojo	36
2.3.2 Características del sensor Sharp modelo GP2Y0A02YK.....	36
2.3.3 Salida no lineal del sensor GP2Y0A02YK	38
2.3.4 Principio de operación del sensor GP2Y0A02YK	38
2.3.5 Funcionamiento y programación del modulo A/D del microcontrolador PIC16f877A	39
2.3.5.1.Convertidor analógico digital (ADC)	39
2.3.5.2 Referencia de voltaje externa	40
2.3.5.3 Registros del modulo A/D	40
2.3.5.4 Pasos necesarios para realizar una conversión A/D	42
2.3.5.5 Cálculos realizados referentes a los módulos AD, TMR0 y TMR1	43
2.3.6 Asignación de duty cycle en base a las lecturas del sensor medidor de distancia.....	44
2.4.0 Etapa de alimentación.....	45
2.5.0 Esquemas del controlador	46
2.6.0 Costo de materiales	51
CONCLUSIONES DEL CAPITULO II	52
REFERENCIA BIBLIOGRAFICAS	53
III. DISEÑO E IMPLEMENTACION DEL SOFTWARE	54
3.1 .0 Programa principal	57
3.1.1 Transmisión de datos del microcontrolador al touchpad.....	55
3.1.2 Transmisión de datos del touchpad al microcontrolador.....	57
3.1.3 Conteo de estados.....	59
3.1.4 Determinación de la tendencia.....	60
CONCLUSIONES DEL CAPITULO III	66
REFERENCIAS BIBLIOGRAFICAS	67
CONCLUSIONES GENERALES Y RECOMENDACIONES	68
ANEXOS	69

LISTA DE TABLAS.

Tabla		Página
1.1	Cuadro sinóptico de las entradas del controlador.....	2
1.2	Cuadro sinóptico de las salidas del controlador.....	2
1.3	Detalle de la habilitación/inhabilitación de los sensores para cada movimiento.....	3
1.4	Valores de la señal de salida de control de dirección para establecer el sentido de rotación en los motores.....	4
1.5	Contenido de paquete relativo PS/2.....	9
1.6	Contenido de paquete absoluto.....	11
1.7	Detalle del escenario ES 1.1.....	16
1.8	Detalle del escenario ES 1.2.....	16
1.9	Detalle del caso de uso avance.....	16
1.10	Detalle del escenario ES 2.1.....	18
1.11	Detalle del caso de uso retroceso.....	18
1.12	Detalle del escenario ES 3.1.....	19
1.13	Detalle del caso de uso giro hacia la izquierda.....	20
1.14	Detalle del escenario ES 4.1.....	21
1.15	Detalle del caso de uso giro hacia la derecha.....	22
1.16	Detalle del escenario ES 5.1.....	23
1.17	Detalle del escenario ES 5.2.....	23
1.18	Detalle del caso de uso detección de usuario.....	24
2.1	Lista de touchpad ofrecidos por tres fabricantes.....	33
2.2	Sensores Sharp medidores de distancias por IR.....	36
2.3	Relación entre el voltaje de salida del sensor y la asignación de duty cycle...	45
2.4	Costo del controlador.....	51
2.5	Costo fuente de alimentación.....	51

LISTA DE FIGURAS.

Figura	Página
1.1 Señal modulada con un duty cycle del 50%.....	5
1.2 Señal modulada con un duty cycle del 10%.....	6
1.3 Diagrama de bloques del controlador.....	7
1.4 Disposición de la capa aislante y la grilla de conductores del touchpad.....	9
1.5 Desplazamiento para avance.....	12
1.6 Trayectoria de movimiento tipo avance.....	12
1.7 Disposición del botón utilizado para desplazamientos largos.....	13
1.8 Desplazamiento para retroceso.....	13
1.9 Desplazamiento para giro izquierdo.....	14
1.10 Trayectoria de movimiento tipo avance.....	14
1.11 Desplazamiento para giro derecho.....	15
2.1 Diagrama de encapsulado del PIC16F877A.....	28
2.2 Salida PWM.....	30
2.3 Conexión de las salidas de control de velocidad y dirección.....	32
2.4 Touchpad ALPS modelo 56AAA1965B.....	33
2.5 Dimensiones y detalle del conector del touchpad.....	34
2.6 Conexión del touchpad con el microcontrolador.....	34
2.7 Apariencia y dimensiones físicas del sensor Sharp GP2Y0A02YK.....	37
2.8 Conexión de los sensores con el microcontrolador.....	37
2.9 Curva de Voltaje de Salida Análogo vs. Distancia a objeto reflectivo.....	38
2.10 Sistema óptico de medición de distancias.....	39
2.11 Diagrama de bloque del modulo A/D.....	40
2.12 Selección de justificación derecha e izquierda del resultado.....	41
2.13 Secuencia de conversión A/D.....	42
2.14 Diagrama de la fuente de alimentación.....	46
2.15 Diagrama esquemático del controlador.....	47
2.16 Esquema del circuito impreso.....	48
2.17 Vista trasera del prototipo construido.....	49
2.18 Vista frontal del prototipo construido.....	49
2.19 Vista lateral del prototipo construido.....	50
2.20 Vista frontal del touchpad.....	50
3.1 Figura 3.1 Diagrama de flujo principal	54
3.2 Diagrama de flujo para transmisión de datos del microcontrolador al touchpad.....	56
3.3 Continuación del diagrama de flujo para transmisión de datos del microcontrolador al touchpad.....	57
3.4 Diagrama de flujo para tratamiento de interrupción por flanco en RB0, cuando se va a recibir un byte enviado por el touchpad.....	58
3.5 Diagrama de conteo de estados.....	59
3.6 Determinación de la tendencia de los bytes leídos.....	60
3.7 Diagrama de flujo para movimiento tipo avance.....	61

3.8	Diagrama de flujo para movimiento tipo retroceso.....	62
3.9	Diagrama de flujo para movimiento tipo giro derecho.....	63
3.10	Diagrama de flujo para movimiento tipo giro izquierdo.....	64
3.11	Diagrama de flujo para tratamiento de la interrupción A/D.....	65

CAPITULO I

DISEÑO DEL CONTROLADOR.

Introducción:

En este capítulo se establecen las bases del diseño del controlador, esto comprende el establecimiento de las características, controles y comportamiento que se pretende darle al controlador, de manera que el diseño tenga una base real y no se tengan que tomar decisiones de prueba y error en las etapas siguientes.

El diseño comprende el planteamiento del problema, determinación de las funciones del controlador y los casos de uso.

Esta información servirá de base para el desarrollo de los próximos capítulos.

1.1.0 Planteamiento del problema.

El propósito principal del presente trabajo, es diseñar un controlador para silla de ruedas que utilice como entrada de mando un “touchpad” (dispositivo utilizado para apuntamiento en computadoras), es decir que por medio de este, se pueda controlar el movimiento de la silla de ruedas (adelante, atrás, giro izquierdo, giro derecho y otros).

Además el controlador debe hacer uso de sensores y otros componentes electrónicos de entrada para brindar mayor seguridad al usuario durante la operación del controlador.

1.1.1 Definición de entradas y salidas del controlador.

Para el diseño del controlador, es necesario establecer las diferentes entradas y salidas que conforman el controlador, esto permitirá conocer los recursos con que debe contar el microcontrolador.

Entradas:

Botón de reinicio.

Botón de inhabilitación.

“Touchpad”.

Sensor medidor de distancias delantero.

Sensor medidor de distancias trasero.

Detector de usuario.

Salidas:

Dos señales de control de velocidad.

Dos señales de control de dirección de giro.

A continuación se muestra un cuadro sinóptico de las entradas y salidas del controlador.

Tabla 1.1 Cuadro sinóptico de las entradas del controlador.

ENTRADA	FUNCION	DESCRIPCION	TIPO DE SEÑAL
Botón de reinicio.	Inicializar el controlador.	Entrada manual para inicializar el controlador y este abandone el estado activo de freno de emergencia.	Digital.
Botón de inhabilitación.	Inhabilitar la operación del controlador.	Entrada manual para detener la operación del controlador.	Digital.
Touchpad.	Control de movimiento.	Entrada de mando para controlar las señales de dirección de giro del controlador	Digital.
Sensor medidor de distancias delantero.	- Detectar obstáculos frontales. - Referencia para asignación de duty cycle.	Sensor colocado en la parte frontal del controlador para detectar posibles objetos de colisión en la trayectoria de recorrido hacia adelante.	Análoga.
Sensor medidor de distancias trasero.	- Detectar obstáculos traseros. - Referencia para asignación de duty cycle.	Sensor colocado en la parte trasera del controlador para detectar posibles objetos de colisión en la trayectoria de recorrido hacia atrás.	Análoga.
Detector de usuario.	Indicar que el usuario se encuentra sentado en la silla.	Interruptor que se activa cuando el usuario se sienta en la silla de ruedas y se desactiva cuando este la abandona.	Digital.

Tabla 1.2 Cuadro sinóptico de las salidas del controlador.

SALIDA	FUNCION	DESCRIPCION	TIPO DE SEÑAL
Señales de control de velocidad.	Control de velocidad	Salidas PWM (Pulse Width Modulation) del controlador	Digital.
Señales de control de dirección.	Control de dirección de giro	Salidas digitales para establecer el sentido de giro.	Digital.

1.1.2 Habilitación de sensores medidores de distancia.

Como los sensores se colocaran en la parte frontal y trasera del prototipo, estos únicamente detectaran objetos que se encuentren en la trayectoria de avance o retroceso.

El sensor delantero se habilitara únicamente cuando el usuario realice el movimiento tipo avance sobre el “touchpad”, el sensor trasero en este caso quedara deshabilitado.

El sensor trasero se habilitara únicamente cuando el usuario realice el movimiento tipo retroceso sobre el “touchpad”, el sensor delantero en este caso quedara deshabilitado.

Para los movimientos giro izquierdo y derecho, no se habilitara ningún sensor, ya que la trayectoria de giro del prototipo no coincide con el campo de rango de los sensores, debido a la ubicación en que estos se encuentran colocados sobre el prototipo.

Tabla 1.3 Detalle de la habilitación/inhabilitación de los sensores para cada movimiento.

MOVIMIENTO	SENSOR HABILITADO	SENSOR INHABILITADO
Avance recorrido corto	Delantero	Trasero
Avance recorrido largo	Delantero	Trasero
Retroceso recorrido corto	Trasero	Delantero
Giro Izquierdo	-	Delantero Trasero
Giro Derecho	-	Delantero Trasero

1.1.3 Determinación de la distancia hasta un obstáculo y su relación con la modificación PWM³.

Los sensores que se emplearan serán infrarrojos, estos emiten un haz de luz y si existe un objeto dentro de su rango de detección, la luz es reflejada hacia el sensor, luego el este genera un voltaje de salida proporcional a la distancia en que se encuentra el objeto de detección.

Para obtener datos experimentales que relacionen la distancia con el voltaje de salida del sensor, se coloca un obstáculo y se empieza a variar la distancia entre el sensor y el obstáculo, para cada distancia se toma la medición con un multímetro. Conociendo estos valores es posible hacer una partición en el rango de detección del sensor para asignar un PWM correspondiente, que se empleara como referencia de velocidad.

1.1.4 Detección de usuario.

El controlador necesita saber si el usuario se encuentra sentado en la silla de ruedas para ejecutar una operación normal o quedarse en espera del estado activo del botón de reinicio. Para esto se cuenta con un botón pulsador que se activa cuando el usuario se sienta en la silla de ruedas y se desactiva cuando este la abandona.

Debido a que no se cuenta con una silla de ruedas, el botón pulsador será colocado en el chasis del controlador y será activado/desactivado manualmente para simular la presencia de usuario sentado en la silla de ruedas.

1.1.5 Inhabilitación del controlador.

El usuario en cualquier momento puede detener la operación del controlador y así inhibir las señales de control de velocidad y de dirección. Para realizar esta función el usuario contara con un botón pulsador que estará colocado en la parte frontal del chasis del

³ PWM: Pulse Width Modulation (Modulación por ancho de pulso).

controlador, al ser accionado, causara una interrupción en el microcontrolador, esta detendrá el flujo normal de operación del controlador y quedara a la espera de la activación del botón de reinicio por parte del usuario.

1.1.6 Reinicialización del controlador.

Cuando se haya activado el botón de inhabilitación o el detector de usuario se encuentra inactivo, la única forma de restablecer la operación normal del controlador es a través del botón de reinicio. Este botón pulsador debe colocarse en la parte trasera del chasis del controlador, de modo que el usuario no lo active accidentalmente, sin embargo para efectos de demostración se colocara en la parte frontal.

1.1.7 Señales de control de velocidad y de dirección.

Como el controlador de la silla emplea dos motores de corriente continua, se necesitan dos señales de control de velocidad y dos de control de dirección. La señal de control de velocidad, es una señal PWM (ver sección 1.1.7.1) y opera de manera conjunta con la señal de control de dirección para establecer la velocidad y sentido de rotación del motor. Las salidas de control de velocidad y de dirección serán diseñadas para ser conectadas a dos controladores de velocidad puente H, uno para cada motor.

Tabla 1.4 Valores de la señal de salida de control de dirección para establecer el sentido de rotación en los motores.

MOVIMIENTO	SALIDA DE CONTROL DE DIRECCION MOTOR IZQUIERDO	SALIDA DE CONTROL DE DIRECCION MOTOR DERECHO	SENTIDO DE ROTACION MOTOR IZQUIERDO	SENTIDO DE ROTACION MOTOR DERECHO
Avance	Bajo	Alto	SCMR	SMR
Retroceso	Alto	Bajo	SMR	SCMR
Giro Izquierdo	Alto	Alto	SMR	SMR
Giro Derecho	Bajo	Bajo	SCMR	SCMR

1.1.7.1 Modulación de ancho de pulso (PWM).

La modulación de ancho de pulso consiste en generar una señal cuadrada de frecuencia fija a la cual se le varia el tiempo en que dicha señal permanece en estado alto, para proveer un voltaje variable promedio de corriente continua.

En otras palabras, el período de la señal se mantendrá constante, pero la cantidad de tiempo que permanezca en estado alto o bajo dentro del período puede variar.

El “duty cycle⁴” es definido como el porcentaje de la señal que permanece en estado alto durante el periodo PWM.

Para controlar la velocidad de un motor de corriente continua, se varía el “duty cycle” de la señal que se le suministra al motor de modo que el voltaje promedio entre sus terminales varíe.

El voltaje promedio para una señal PWM se calcula de la siguiente forma:

$$\text{Voltaje promedio} = \frac{1}{T} \int_0^T v(t) dt \quad \text{Ec. (1.1)}$$

Donde:

$v(t)$: Señal PWM en el dominio del tiempo.

T : Periodo de la señal PWM.

A continuación se muestra el cálculo del voltaje promedio para una señal PWM con dos ciclos de trabajo diferentes:

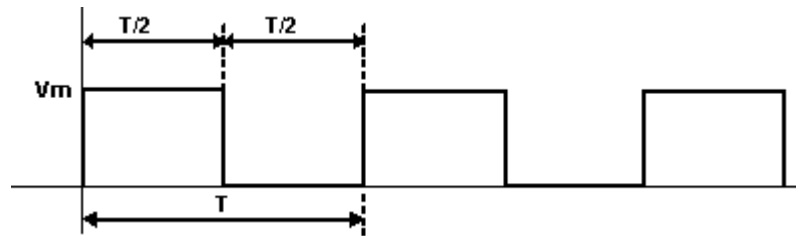


Figura 1.1. Señal modulada con un ciclo de trabajo del 50%.

Mediante la ecuación (1.1) el voltaje promedio para esta señal es:

$$\text{Voltaje promedio} = \frac{1}{T} \int_0^{T/2} V_m dt + \frac{1}{T} \int_{T/2}^T 0 dt$$

$$\boxed{\text{Voltaje promedio} = \frac{V_m}{2}} \quad \text{Para un ciclo de trabajo del 50\%}$$

Donde:

V_m : Valor máximo de voltaje de la señal PWM.

La siguiente figura muestra una señal PWM con un ciclo de trabajo del 10%.

⁴ Duty cycle: ciclo de trabajo.

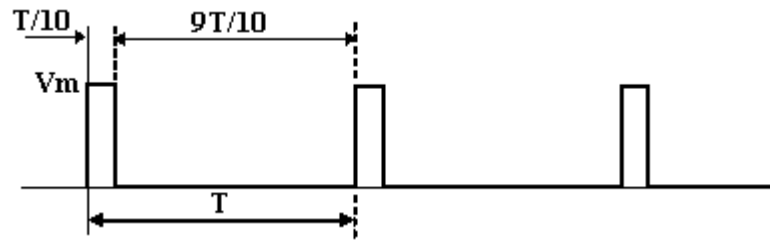


Figura 1.2. Señal modulada con un “duty cycle” del 10%.

Mediante la ecuación (1.1) el voltaje promedio para esta señal es:

$$\text{Voltaje promedio} = \frac{1}{T} \int_0^{T/10} V_m dt + \frac{1}{T} \int_{T/10}^T 0 dt$$

$$\boxed{\text{Voltaje promedio} = \frac{V_m}{10}} \quad \text{Para un ciclo de trabajo del 10\%}$$

Para un caso general, el voltaje promedio en función del ciclo de trabajo queda expresado de la siguiente forma:

$$\boxed{\text{Voltaje promedio} = \frac{V_m}{K}} \quad \text{Ec. (1.2)}$$

Donde:

V_m: Valor máximo de voltaje de la señal PWM.

K: Duty cycle de la señal PWM.

Cuando se incrementa o decrementa el ciclo de trabajo de una señal PWM, el voltaje de salida promedio se incrementa o decrementa respectivamente.

1.2.0 Diagrama de bloques.

El diagrama de bloques permite una fácil comprensión de la operación de un circuito, ya que su funcionamiento está dividido por etapas que realizan una operación específica, además de mostrar la interrelación de unas etapas con otras.

En la figura 1.3. Se muestra el diagrama de bloques del controlador de mando tipo Mouse (touchpad) para una silla de ruedas. A continuación se explicará la operación de cada uno de estos bloques.

1.2.1 Etapa de control.

El componente principal de esta etapa es el microcontrolador, este es un circuito integrado programable que se emplea para controlar el funcionamiento de una tarea determinada, por medio de un programa escrito en su memoria.

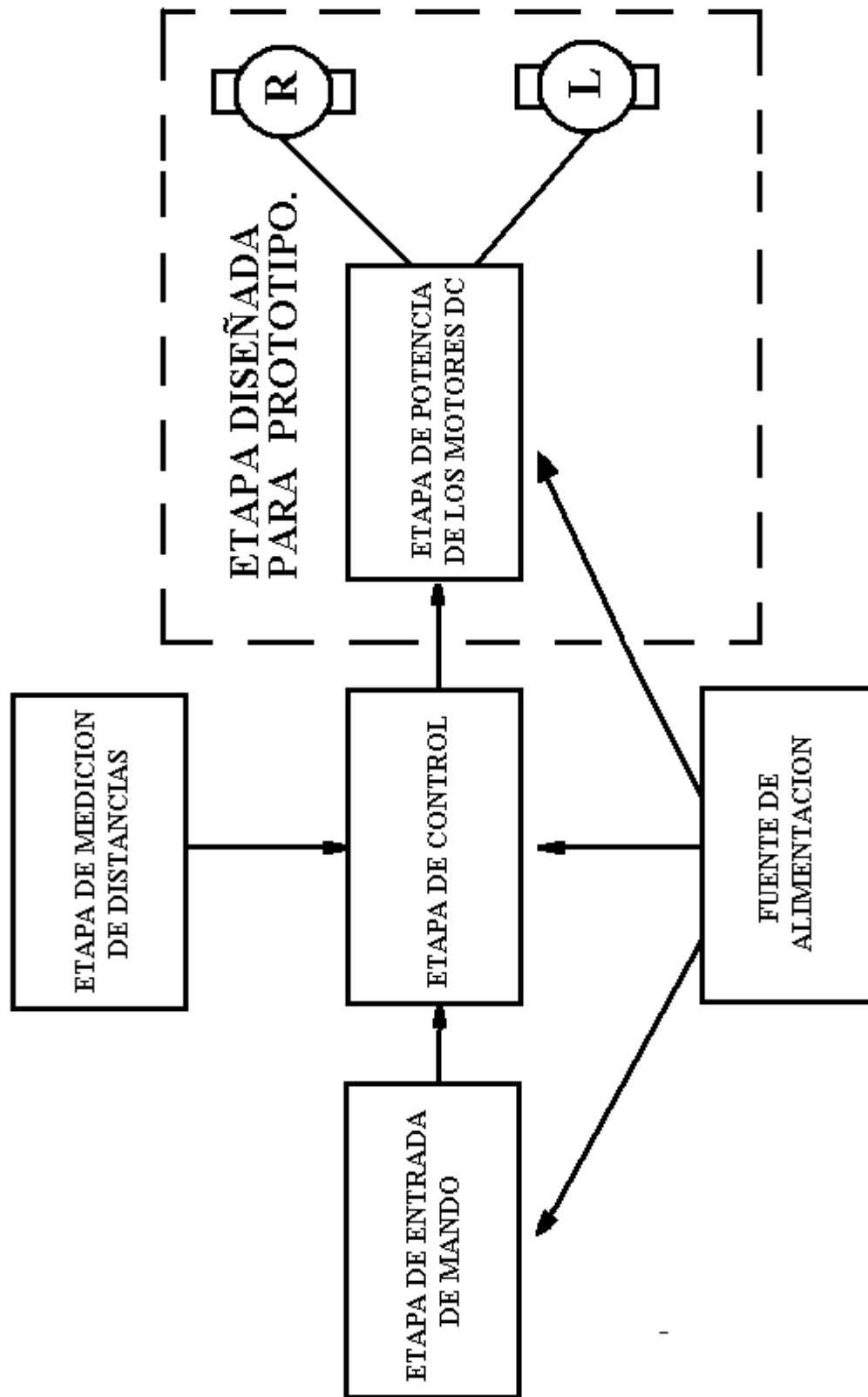


Figura 1.3. Diagrama de bloques del controlador de mando tipo touchpad para una silla de ruedas.

La silla de ruedas requiere de un sistema que le permita controlar la recepción de la información enviada por el “touchpad” (formato digital) y por los sensores medidores de distancia por infrarrojo (formato analógico), debiendo transformar esta información en señales de control de movimiento para la silla. Estas señales de control de movimiento corresponden a señales moduladas por ancho de pulso PWM, las cuales controlaran los cambios de velocidad en la silla.

El control de dirección será realizado haciendo girar los motores en un sentido establecido, ver tabla 1.4.

1.2.2 Etapa de entrada de mando.

Como dispositivo de entrada de mando se utilizara un “touchpad”, este se encargará de sensar la posición del dedo de una persona sobre una superficie de navegación, esta información expresada en coordenadas espaciales (x,y) serán enviadas a la etapa de control para decidir el movimiento que tomara la silla: giro izquierdo, giro derecho, avance y retroceso.

El “touchpad” se utiliza principalmente como apuntador para computadoras, no obstante por su tamaño compacto, bajo costo y su carencia de partes móviles lo hace ideal para ser empleado en el diseño de dispositivos portátiles como es el caso de la silla de ruedas.

Las sillas eléctricas comerciales cuentan como entrada de mando un “joystick” para controlar la dirección del movimiento, el diseño realizado pretende sustituir este joystick por un “touchpad”. La idea de utilizar un “touchpad” y no un “joystick” es que algunas personas usuarias de sillas de ruedas tienen problemas de movilidad de sus manos dificultándoseles el poder manipular un “joystick”, por lo que el “touchpad” representa una alternativa viable para controlar la dirección de movimiento de la silla.

1.2.2.1 Principio de operación del “touchpad”.

El “touchpad” está formado por una rejilla de dos capas de tiras de electrodos, una vertical y otra horizontal, separadas por un aislante y conectadas a un sofisticado circuito, ver figura 1.4. El circuito se encarga de medir la capacidad mutua entre cada electrodo vertical y horizontal. Cuando un dedo es situado cerca de la intersección de dos electrodos modifica la capacidad mutua entre ellos. (Recordar que el cuerpo humano es un buen conductor de la electricidad).

La posición del dedo se calcula con precisión basándose en las variaciones de la capacidad mutua en varios puntos hasta determinar el centroide de la superficie de contacto. La resolución de este sistema es notable, hasta 1/40 mm (0.025 mm). Además se puede medir también la presión que se hace con el dedo.

No se pueden usar lápices u otros materiales no conductores, aunque en algunos modelos más sofisticados si es posible. Es muy resistente al entorno, soporta perfectamente el polvo, la humedad y la electricidad estática.

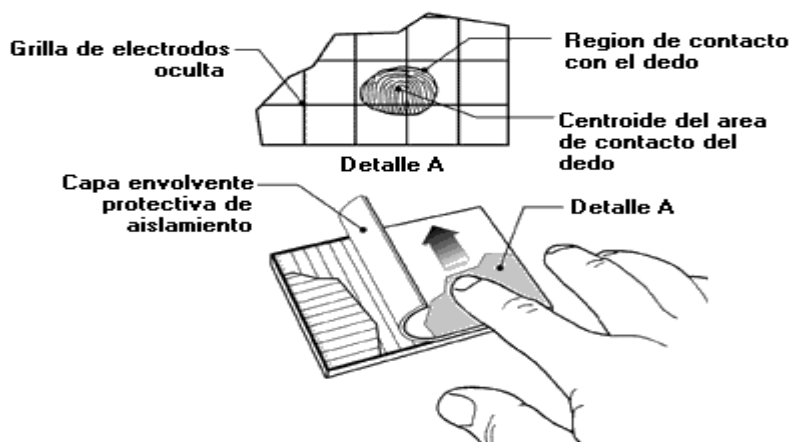


Figura 1.4. Disposición de la capa aislante y la grilla de conductores del touchpad.

1.2.2.2 Reporte de datos del “touchpad”.

El “touchpad” soporta dos modos para expresar el movimiento de los paquetes. Estos modos son: relativo y absoluto.

1.2.2.2.1 Modo relativo.

Es llamado así debido a que las acciones de los dedos sobre el “touchpad” son reportadas en términos de movimientos relativos parecidos al de un ratón normal a través del pad. El “touchpad” reporta este movimiento relativo en paquetes compatibles a los de un ratón. El “touchpad” genera aproximadamente 40-80 paquetes por segundo. Cada paquete reporta la cantidad de movimiento en las direcciones horizontal (X) y vertical (Y) que han ocurrido desde el paquete previo. Estas cantidades de movimientos son llamados *deltas*, y son escritas “ ΔX ” y “ ΔY ”. El paquete también reporta información sobre el estado del botón izquierdo y derecho.

El paquete relativo PS/2 contiene la siguiente información:

Tabla 1.5. Contenido de paquete relativo PS/2⁵.

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Byte 1	Sobre flujo Y	Sobre flujo X	Signo Y	Signo X	1	Medio	Derecho	Izquierdo
Byte 2	Delta X							
Byte 3	Delta Y							

Sobre flujo Y.

- 1: Se ha excedido el rango del contador de movimiento en Y (-255, 255).
- 0: No se ha excedido el rango.

⁵ Synaptics Touchpad Interfacing Guide.

Sobre flujo X.

1: Se ha excedido el rango del contador de movimiento en X (-255, 255).

0: No se ha excedido el rango.

Signo Y.

1: Valor delta Y es negativo.

0: Valor delta Y es cero o positivo.

Signo X.

1: Valor delta X es negativo.

0: Valor delta X es cero o positivo.

Medio. Estado físico del botón medio (En algunos “touchpad” no esta implementado)

Derecho. Estado físico del botón derecho.

1: Botón derecho presionado.

0: Botón derecho liberado.

Izquierdo. Estado físico del botón izquierdo.

1: Botón derecho presionado (gesto de golpe o arrastre).

0: Botón derecho liberado.

Delta X. Representa la cantidad de movimiento que ha ocurrido en la dirección X (horizontal) desde el último reporte de movimiento de datos.

Delta Y. Representa la cantidad de movimiento que ha ocurrido en la dirección Y (vertical) desde el último reporte de movimiento de datos.

1.2.2.2.2 Modo absoluto.

En el modo absoluto, el “touchpad” transmite un paquete extendido que reporta la posición absoluta del dedo en el pad (X,Y), la presión del dedo o área de contacto (Z), y otras informaciones tales como el estado de los botones. Los paquetes son reportados continuamente en la velocidad especificada, 40 u 80 paquetes por segundo, siempre que el dedo este sobre o cercano al pad.

Específicamente el “touchpad” comienza transmitiendo paquetes cuando Z (variable inherente del “touchpad” para denotar la presión u área de contacto) tiene valor de 8 o mayor.

El “touchpad” también comienza enviando paquetes siempre que cualquier botón sea presionado o liberado. Una vez que comienza a transmitir, continua enviando paquetes por un segundo después que Z cae debajo de 8 y los botones cambian de estado.

El paquete estándar absoluto contiene la siguiente información:

Tabla 1.6. Contenido de paquete absoluto.

Campo	Tamaño (bits)	Rango	Significado
X	13	0 - 6143	Posición horizontal del dedo. 0 = mas alejado a la izquierda.
Y	13	0 - 6143	Posición vertical del dedo. 0 = mas alejado en el fondo.
Z	8	0 - 255	Presión u área de contacto, 0 = no contacto
W	4	0 - 15	Ancho del dedo u otro estado de información.
Izquierdo	1	0 o 1	Estado físico del botón izquierdo 0 = no presionado, 1 = presionado
Derecho	1	0 o 1	Estado físico del botón derecho. 0 = no presionado, 1 = presionado
Gestión	1	0 o 1	Gestión de golpe/arrastre en progreso 0 = no gestión, 1 = gestión.
dedo	1	0 o 1	Presencia del dedo, 0 = no dedo, 1 = dedo.

Para el modo de reporte de datos del touchpad se empleara el modo relativo, ya que la información contenida en el paquete relativo es suficiente para los fines requeridos del controlador (variación en cantidad de movimiento (X,Y) y estado del botón derecho).

1.2.3 Etapa de medición de distancias.

Esta etapa sensorá la distancia del objeto más próximo a la silla de ruedas, que se encuentre en su trayectoria de recorrido (dentro del rango de detección), con el fin de evitar posibles choques que pudieran ocasionar daño físico al usuario de la silla de ruedas. La señal de salida de esta etapa se enviara a la etapa de control para su proceso.

En el mercado existen diversos sensores especializados para realizar esta tarea, pudiéndose agrupar en dos grandes grupos a saber:

- Sensores que utilizan el principio óptico de la luz.
- Sensores que utilizan el principio de ultrasonido.

En el diseño propuesto se emplearan los primeros, por ser de menor costo y fácil utilización.

El sensor de medición de distancias por infrarrojos, es un dispositivo que proporciona una lectura continua de la distancia medida como una tensión analógica o salida digital dentro de un rango determinado generalmente menor a 150 cm.

1.2.4 Etapa de alimentación.

Todos los dispositivos electrónicos tienen requisitos específicos de voltajes y corrientes de operación. La fuente de alimentación esta diseñada para proporcionar estos de una fuente externa como lo es una Batería de automóvil de 12 VDC, no obstante algunos componentes requieren un voltaje menor de operación y será necesario reducir el nivel de voltaje para estos componentes utilizando reguladores de voltaje.

1.3.0 Acciones definidas del usuario sobre el touchpad para controlar el movimiento de la silla.

A continuación se describirán los movimientos realizados sobre el touchpad para definir el movimiento de la silla de ruedas.

1.3.1 Avance para desplazamiento corto.

El usuario de la silla de ruedas coloca uno de sus dedos en el *touchpad* y lo desplaza cierta distancia tal como se indica en la figura 1.5.

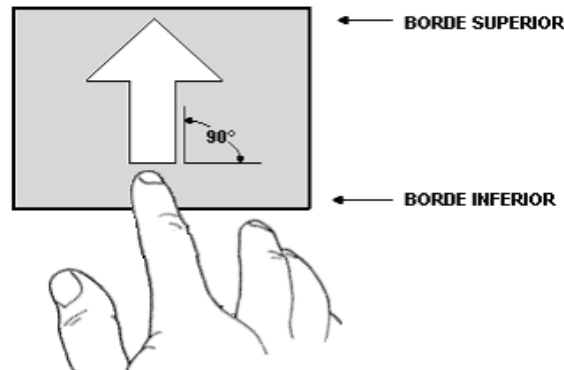


Figura 1.5 Desplazamiento para avance.

Este desplazamiento debe formar un ángulo aproximado de 90° con la base del *touchpad*, permitiéndose un margen de tolerancia de $\pm 45^\circ$ entre dos puntos consecutivos en la trayectoria recorrida por el dedo. Esto implica que Fuera de este margen, el movimiento no será tomado como uno de tipo avance para desplazamientos cortos. Ver figura 1.6.

El punto de inicio del movimiento del dedo sobre el touchpad no esta restringido a un área específica del *touchpad*, puede ser en cualquier coordenada dentro de sus límites. La silla avanzará hacia adelante con el inicio del movimiento descrito y comenzara a desacelerar cuando el usuario detenga el movimiento de su dedo sobre el *touchpad* o levante su dedo de su superficie.

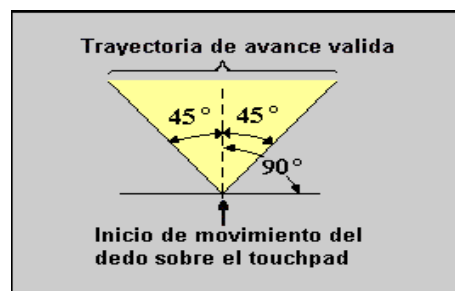


Figura 1.6 Trayectoria de movimiento tipo avance.

El usuario debe esperar a que la silla se encuentre detenida antes de efectuar un nuevo movimiento, si se intenta hacer un cambio de movimiento brusco, la silla de ruedas no responderá inmediatamente sino que reducirá gradualmente su velocidad hasta detenerse y luego efectuará el movimiento indicado por el usuario.

Esto es aplicable para los cuatro movimientos básicos de la silla: **avance, retroceso, giro izquierda y giro derecha.**

1.3.2 Avance para desplazamiento largo.

El usuario de la silla de ruedas presiona el botón que se encuentra localizado en la parte inferior derecha, próximo a la base del touchpad, tal como se indica en la figura 1.7 En ese instante, la silla comenzará a moverse hacia adelante y comenzará a desacelerar hasta que el usuario deje de presionar dicho botón.

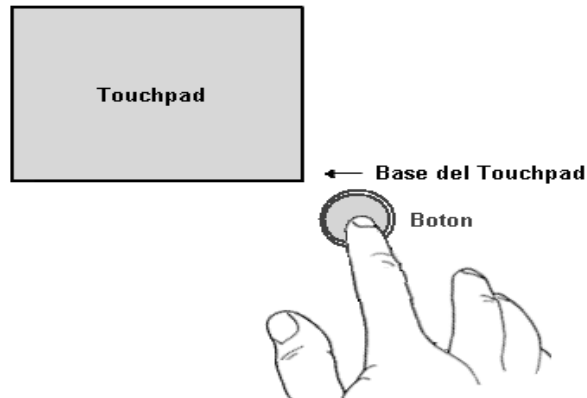


Figura 1.7 Disposición del botón utilizado para desplazamientos largos.

1.3.3 Retroceso para desplazamiento corto.

El usuario de la silla de ruedas coloca uno de sus dedos en el *touchpad* y lo desplaza cierta distancia como se indica en la figura 1.8.

Este desplazamiento debe formar un ángulo aproximado de 90° con el borde superior del *touchpad*, permitiéndose un margen de tolerancia de $\pm 45^\circ$ con la trayectoria recorrida por el dedo. Fuera de este margen, el movimiento no será tomado como uno de tipo retroceder.

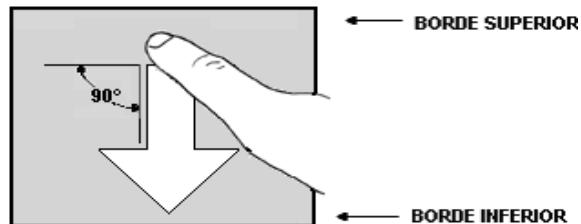


Figura 1.8 Desplazamiento para retroceso.

La silla comenzará a retroceder con el inicio del movimiento descrito y terminará de retroceder hasta que el usuario detenga el movimiento de su dedo sobre el *touchpad* o levante su dedo de la superficie del *touchpad*.

1.3.4 Retroceso para desplazamiento largo.

No existe un botón para hacer retroceder la silla en largos recorridos, tal como existe para el caso de avanzar. El usuario de la silla de ruedas debe efectuar el retroceso para desplazamientos cortos expuesto anteriormente. También tiene la opción de hacer girar la silla hasta la dirección deseada y luego presionar el botón para desplazamientos largos.

1.3.5 Giro izquierdo.

El usuario de la silla de ruedas coloca uno de sus dedos en el *touchpad* y lo desplaza cierta distancia como se indica en la figura 1.9.

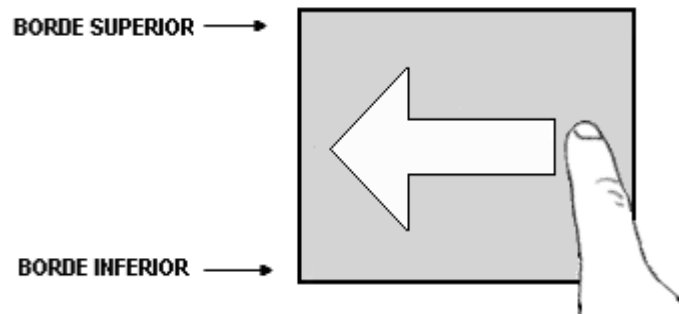


Figura 1.9. Desplazamiento para giro izquierdo.

Este desplazamiento debe formar un ángulo aproximado de 0° con la base del *touchpad*, permitiéndose un margen de tolerancia de $\pm 45^\circ$ con la trayectoria recorrida por el dedo. Fuera de este margen, el movimiento no será tomado como uno de tipo giro a la izquierda, ver figura 1.10.

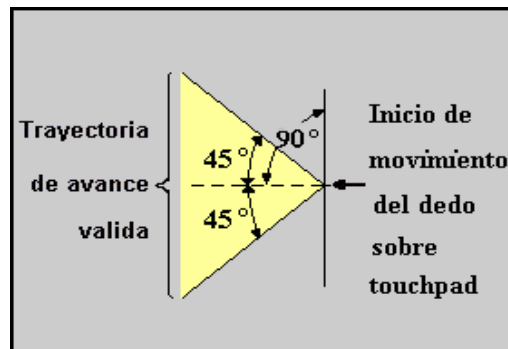


Figura 1.10 Trayectoria de movimiento tipo avance.

La silla comenzará a girar a la izquierda con el inicio del movimiento descrito y comenzará a desacelerar hasta que el usuario detenga el movimiento de su dedo sobre el *touchpad* o levante su dedo de la superficie del mismo.

1.3.6 Giro derecho.

El usuario de la silla de ruedas coloca uno de sus dedos en el *touchpad* y lo desplaza cierta distancia como se indica en la figura 1.11.

Este desplazamiento debe formar un ángulo aproximado de 0° con la base del *touchpad*, permitiéndose un margen de tolerancia de $\pm 45^\circ$ con la trayectoria recorrida por el dedo. Fuera de este margen, el movimiento no será tomado como uno de tipo giro a la derecha.

La silla comenzará a girar a la derecha con el inicio del movimiento descrito y comenzará a desacelerar hasta que el usuario detenga el movimiento de su dedo sobre el *touchpad* o levante su dedo de la superficie del mismo.

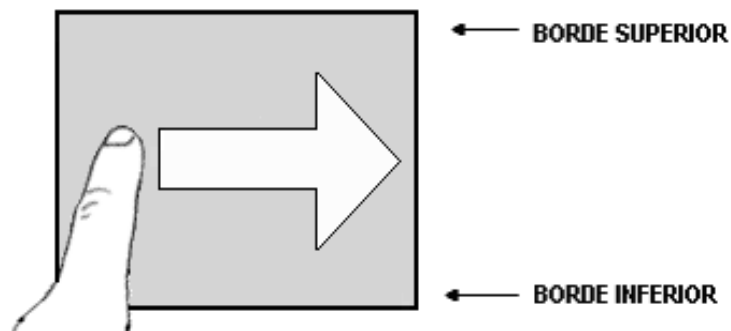


Figura 1.11 Desplazamiento para giro derecho.

1.3.7 Movimientos no validos.

Todo movimiento que el usuario de la silla de ruedas haga con cualquiera de sus dedos sobre el *touchpad* que no sea de avance, retroceso, giro izquierda o giro derecha será tomado como no válido, por lo que la silla no responderá ante tales movimientos.

1.4.0 Casos de usos de la silla de ruedas.

Los casos de uso representan lo que hace el sistema desde el punto de vista del usuario. Es decir, describen un uso del sistema y cómo este interactúa con el usuario.

A continuación describiremos los posibles casos de uso de la silla de ruedas.

1.4.1 Caso de uso avance y sus escenarios.

A continuación se muestra el desarrollo del caso de uso avance y sus escenarios:

Tabla 1.7 Detalle del escenario ES 1.1

Identificación escenario	ES 1.1
Nombre del escenario	Avance para desplazamiento corto.
Precondición	El usuario debe estar sentado en la silla de ruedas y la alimentación eléctrica debe ser suficiente para operar el controlador.
Poscondición:	La silla ha sido desplazada hacia adelante.

Tabla 1.8 Detalle del escenario ES 1.2

Identificación escenario	ES 1.2
Nombre del escenario	Avance para desplazamiento largo.
Precondición	El usuario debe estar sentado en la silla de ruedas y la alimentación eléctrica debe ser suficiente para operar el controlador.
Poscondición:	La silla ha sido desplazada hacia adelante.

Tabla 1.9 Detalle del caso de uso avance.

CU 1	Avance.
Descripción.	El controlador deberá permitir al usuario de la silla de ruedas, desplazarse hacia adelante siempre y cuando no haya ningún obstáculo en su trayectoria de recorrido a menos de 20 cm, para esto el controlador dispondrá de un sensor colocado en la parte frontal del prototipo. Para desplazamientos cortos el usuario deberá utilizar el <i>touchpad</i> y para desplazamientos largos, un botón situado próximo al <i>touchpad</i> .
Actor:	Usuario de la silla de ruedas.

Tabla 1.9 (continuación).

Secuencia Normal.	1a-	El usuario hace el movimiento definido para avanzar sobre el <i>touchpad</i> con cualquier dedo de su mano izquierda o derecha.
	1b	El usuario de la silla de ruedas presiona el botón de avance de recorrido largo con cualquier dedo de su mano izquierda o derecha.
	2	El controlador lee las secuencias de coordenadas de movimiento enviadas por el <i>touchpad</i> y establece la tendencia de estas.
	3	Si la tendencia indica que el usuario desea avanzar, el controlador determina si la silla de ruedas se encuentra actualmente en un movimiento tipo avance.
	4	El controlador lee el registro que almacena el duty cycle y lo incrementa un 3%.
	5	El controlador lee el sensor medidor de distancia delantero y de acuerdo a su lectura asigna un duty cycle.
	6	El controlador asigna a las salidas PWM el menor de los duty cycle obtenidos en los pasos 4 y 5.
Excepciones.	2'	Error en la tendencia de las secuencias de coordenadas (primera y segunda vez). El controlador permitirá al usuario volver a presionar el botón de avance de recorrido largo o hacer el movimiento definido para avanzar sobre el <i>touchpad</i> (paso 1b o 1a).
	2''	Error en la tendencia de las secuencias de coordenadas (tercera vez). El controlador comenzara a reducir gradualmente la velocidad de la silla, abandonara esta condición hasta que haya una nueva secuencia de movimiento del <i>touchpad</i> o la velocidad de la silla sea cero, en cualquier caso el controlador permitirá al usuario volver a presionar el botón o hacer el movimiento definido para avanzar sobre el <i>touchpad</i> (paso 1b o 1a).
	3	La silla de ruedas no se encuentra en un movimiento tipo avance. El controlador deberá reducir gradualmente la velocidad de la silla en la dirección en que se encuentre hasta detener su marcha y luego ejecutar la nueva orden de movimiento.
	5	Detección de un objeto cercano en un rango menor de 20 cm. La silla no podrá desplazarse hacia adelante aunque reciba la orden de avanzar.
	1..6	Inhabilitación del controlador. En cualquier momento el usuario puede inhabilitar el controlador de la silla de ruedas.

1.4.2 Caso de uso retroceso y sus escenarios.

A continuación se muestra el desarrollo del caso de uso retroceso y sus escenarios:

Tabla 1.10 Detalle del escenario ES 2.1

Identificación escenario	ES 2.1
Nombre del escenario	Retroceso para desplazamiento corto.
Precondición	El usuario debe estar sentado en la silla de ruedas y la alimentación eléctrica debe ser suficiente para operar el controlador.
Poscondición:	La silla ha sido desplazada hacia atrás.

Tabla 1.11 Detalle del caso de uso retroceso.

CU 2	Retroceso.	
Descripción.	El controlador deberá permitir al usuario de la silla de ruedas, desplazarse hacia atrás siempre y cuando no haya ningún obstáculo en su trayectoria de recorrido a menos de 20 cm, para esto el controlador dispondrá de un sensor colocado en la parte frontal del prototipo. Para desplazamientos cortos el usuario deberá utilizar el <i>touchpad</i> y para desplazamientos largos, un botón situado próximo al <i>touchpad</i> .	
Actor:	Usuario de la silla de ruedas.	
Secuencia Normal.	1	El usuario hace el movimiento definido para retroceder sobre el <i>touchpad</i> con cualquier dedo de su mano izquierda o derecha.
	2	El controlador lee las secuencias de coordenadas de movimiento enviadas por el <i>touchpad</i> y establece la tendencia de estas.
	3	Si la tendencia indica que el usuario desea retroceder, el controlador determina si la silla de ruedas se encuentra actualmente en un movimiento tipo retroceso.
	4	El controlador lee el registro que almacena el duty cycle y lo incrementa un 3 %.
	5	El controlador lee el sensor medidor de distancia trasero y de acuerdo a su lectura asigna un duty cycle.
	6	El controlador asigna a las salidas PWM el menor de los duty cycle obtenidos en los pasos 4 y 5.

Tabla 1.11 (continuación).

Excepciones.	2'	Error en la tendencia de las secuencias de coordenadas (primera y segunda vez). El controlador permitirá al usuario volver a presionar el botón o hacer el movimiento definido para avanzar sobre el touchpad (paso 1b o 1a).
	2''	Error en la tendencia de las secuencias de coordenadas (tercera vez). El controlador comenzara a reducir gradualmente la velocidad de la silla, abandonara esta condición hasta que haya una nueva secuencia de movimiento del <i>touchpad</i> o la velocidad de la silla sea cero, en cualquier caso el controlador permitirá al usuario volver a hacer el movimiento definido para retroceder sobre el touchpad (paso 1).
	3	La silla de ruedas no se encuentra en un movimiento tipo retroceso. El controlador deberá reducir gradualmente la velocidad de la silla en la dirección en que se encuentre hasta detener su marcha y luego ejecutar la nueva orden de movimiento.
	5	Detección de un objeto cercano en un rango menor de 20 cm. La silla no podrá desplazarse hacia atrás aunque reciba la orden de avanzar por parte del usuario.
	1..6	Inhabilitación del controlador. En cualquier momento el usuario puede inhabilitar el controlador de la silla de ruedas.

1.4.3 Caso de uso giro hacia la izquierda y su escenario.

A continuación se muestra el desarrollo del caso de giro hacia la izquierda y sus escenarios:

Tabla 1.12 Detalle del escenario ES 3.1

Identificación escenario	ES 3.1
Nombre del escenario	Giro izquierdo corto.
Precondición	El usuario debe estar sentado en la silla de ruedas y la alimentación eléctrica debe ser suficiente para operar el controlador.
Poscondición:	La silla girado a la izquierda.

Tabla 1.13 Detalle del caso de uso giro hacia la izquierda.

CU 3	Giro hacia la izquierda.	
Descripción.	El controlador deberá permitir al usuario de la silla de ruedas girar hacia la izquierda, cuando el usuario haga el movimiento establecido para ello sobre el touchpad. Para el movimiento giro a la izquierda no se habilita ningún sensor que detecte objetos en la trayectoria de recorrido. La distancia recorrida en el giro dependerá de la velocidad de desplazamiento del dedo sobre el touchpad y el tiempo que permanezca éste sobre el, debido a las dimensiones del touchpad solo podrán hacerse giros cortos.	
Actor:	Usuario de la silla de ruedas.	
Secuencia Normal.	1	El usuario debe verificar visualmente que no haya obstáculos en la trayectoria de giro que pretende hacer.
	2a	Si no hubiera obstáculo, el usuario debe hacer el movimiento definido para girar a la izquierda sobre el <i>touchpad</i> con cualquier dedo de su mano izquierda o derecha.
	2b	Si hubiera obstáculo el usuario debe hacer retroceder o avanzar la silla hasta que no haya obstáculo en la trayectoria de giro. Luego debe hacer el movimiento definido para girar a la izquierda sobre el <i>touchpad</i> con cualquier dedo de su mano izquierda o derecha.
	3	El controlador lee las secuencias de coordenadas de movimiento enviadas por el <i>touchpad</i> y establece la tendencia de estas.
	4	Si la tendencia indica que el usuario desea girar hacia la izquierda, el controlador determina si la silla de ruedas se encuentra actualmente en un movimiento tipo giro izquierdo.
	5	El controlador lee el registro que almacena el duty cycle y lo incrementa un 5 %.
	6	El controlador asigna a las salidas PWM el duty cycle obtenido en paso 5, si éste es menor que el 10% de lo contrario les asigna el 10%.
Excepciones.	3'	Error en la tendencia de las secuencias de coordenadas (primera y segunda vez). El controlador permitirá al usuario volver a hacer el movimiento definido para girar a la izquierda sobre el touchpad (paso 2a).

Tabla 1.13 (continuación).

	3”	Error en la tendencia de las secuencias de coordenadas (tercera vez). El controlador comenzara a reducir gradualmente la velocidad de la silla, abandonara esta condición hasta que haya una nueva secuencia de movimiento del <i>touchpad</i> o la velocidad de la silla sea cero, en cualquier caso el controlador permitirá al usuario volver a hacer el movimiento definido para girar a la izquierda sobre el touchpad (paso 2a).
	4	La silla de ruedas no se encuentra en un movimiento tipo giro izquierdo. El controlador deberá reducir gradualmente la velocidad de la silla en la dirección en que se encuentre hasta detener su marcha y luego ejecutar la nueva orden de movimiento
	1..6	Inhabilitación del controlador. En cualquier momento el usuario puede inhabilitar el controlador de la silla de ruedas.

1.4.4 Caso de uso giro hacia la derecha y su escenario.

A continuación se muestra el desarrollo del caso de giro hacia la derecha y su escenario:

Tabla 1.14 Detalle del escenario ES 4.1

Identificación escenario	ES 4.1
Nombre del escenario	Giro derecho corto.
Precondición	El usuario debe estar sentado en la silla de ruedas y la alimentación eléctrica debe ser suficiente para operar el controlador.
Poscondición:	La silla girado a la derecha.

Tabla 1.15 Detalle del caso de uso giro hacia la derecha.

CU 4	Giro hacia la derecha.	
Descripción.	El controlador deberá permitir al usuario de la silla de ruedas girar hacia la derecha, cuando el usuario haga el movimiento establecido para ello sobre el touchpad. Para el movimiento giro a la derecha no se habilita ningún sensor que detecte objetos en la trayectoria de recorrido. La distancia recorrida en el giro dependerá de la velocidad de desplazamiento del dedo sobre el touchpad y el tiempo que permanezca éste sobre el, debido a las dimensiones del touchpad solo podrán hacerse giros cortos.	
Actor:	Usuario de la silla de ruedas.	
Secuencia Normal.	1	El usuario debe verificar visualmente que no haya obstáculos en la trayectoria de giro que pretende hacer.
	2a	Si no hubiera obstáculo, el usuario debe hacer el movimiento definido para girar a la derecha sobre el <i>touchpad</i> con cualquier dedo de su mano izquierda o derecha.
	2b	Si hubiera obstáculo el usuario debe hacer retroceder o avanzar la silla hasta que no haya obstáculo en la trayectoria de giro. Luego debe hacer el movimiento definido para girar a la derecha sobre el <i>touchpad</i> con cualquier dedo de su mano izquierda o derecha.
	3	El controlador lee las secuencias de coordenadas de movimiento enviadas por el <i>touchpad</i> y establece la tendencia de estas.
	4	Si la tendencia indica que el usuario desea girar hacia la derecha, el controlador determina si la silla de ruedas se encuentra actualmente en un movimiento tipo giro derecho.
	5	El controlador lee el registro que guarda el duty cycle y lo incrementa un 5 %.
	6	El controlador asigna a las salidas PWM el duty cycle obtenido en paso 5, si éste es menor que el 10% de lo contrario les asigna el 10%.

Tabla 1.15 (continuación).

Excepciones.	3'	Error en la tendencia de las secuencias de coordenadas (primera y segunda vez). El controlador permitirá al usuario volver a hacer el movimiento definido para girar a la derecha sobre el touchpad (paso 2a).
	3''	Error en la tendencia de las secuencias de coordenadas (tercera vez). El controlador comenzara a reducir gradualmente la velocidad de la silla, abandonara esta condición hasta que haya una nueva secuencia de movimiento del <i>touchpad</i> o la velocidad de la silla sea cero, en cualquier caso el controlador permitirá al usuario volver a hacer el movimiento definido para girar a la derecha sobre el touchpad (paso 2a).
	4	La silla de ruedas no se encuentra en un movimiento tipo giro derecho. El controlador deberá reducir gradualmente la velocidad de la silla en la dirección en que se encuentre hasta detener su marcha y luego ejecutar la nueva orden de movimiento.
	1.6	Inhabilitación del controlador. En cualquier momento el usuario puede inhabilitar el controlador de la silla de ruedas.

1.4.5 Caso de uso detección de presencia de usuario y sus escenarios.

A continuación se muestra el desarrollo del caso de uso detectar presencia de usuario y sus escenarios:

Tabla 1.16 Detalle del escenario ES 5.1

Identificación escenario	ES 5.1
Nombre del escenario	Detección de presencia de usuario con usuario presente.
Precondición	El usuario debe estar sentado en la silla de ruedas y la alimentación eléctrica debe ser suficiente para operar el controlador.
Poscondición:	El controlador permite que el usuario efectúe cualquier tipo de movimiento valido para el desplazamiento de la silla.

Tabla 1.17 Detalle del escenario ES 5.2

Identificación escenario	ES 5.2
Nombre del escenario	Detección de presencia de usuario con usuario ausente.
Precondición	El usuario debe estar sentado en la silla de ruedas y la alimentación eléctrica debe ser suficiente para operar el controlador.
Poscondición:	El controlador no permite que el usuario de la silla de ruedas efectúe ningún tipo de movimiento valido para el desplazamiento de la silla.

Tabla 1.18 Detalle del caso de uso detectar presencia de usuario.

CU 5	Detectar presencia de usuario.	
Descripción.	El controlador detectara la presencia del usuario sobre la silla de ruedas para poder operar normalmente, de lo contrario inhibirá cualquier movimiento que se intente hacer. Para ello el controlador contara con un detector colocado en el asiento de la silla de ruedas, de modo que el usuario al sentarse, active el detector. Para mostrar que el usuario esta presente, se activara un led colocado en el modulo de control del prototipo. Cuando el controlador inhiba los movimientos de la silla, el usuario podrá restablecer el controlador presionando el botón de reinicio.	
Actor:	Usuario de la silla de ruedas.	
Secuencia Normal.	1	El usuario activa el interruptor principal de la fuente de alimentación.
	2	El controlador lee el estado del sensor detector de presencia y determina si el usuario se encuentra sentado sobre la silla de ruedas.
	3	Si el usuario estuviera presente, el controlador activa el LED correspondiente para indicar la presencia del usuario, de lo contrario no lo activa.
	4a	El controlador espera la orden de movimiento que el usuario haga.
	4b	El controlador inhibirá cualquier movimiento que se intentara hacer.
Excepciones.	4	Abandono de la silla de ruedas sin apagarla. El controlador inhibirá cualquier movimiento que se intentara hacer, aun si el usuario de volviera a sentarse.
	1..4	Reiniciar. En cualquier momento el usuario podrá re-inicializar el controlador.

CONCLUSIONES DEL CAPITULO I.

- Con el método de los casos y usos se desarrolló un análisis descriptivo de las diversas situaciones en la que se encuentra la silla de ruedas, permitiendo anticipar los resultados que deben esperarse entre el usuario de la silla de ruedas y el controlador, haciendo mas versátil el diseño
- Se lograron definir todas las entradas y salidas que tendrá el controlador, esto permitió definir completamente las necesidades o requerimientos del microcontrolador empleado en el diseño, de manera que este no sea sobredimensionado.
- El diseño propuesto en este capítulo servirá de base para la implementación tanto del software como del hardware a realizar, ya que quedan descritos aspectos de comportamiento del usuario, acciones del controlador y la determinación de los requerimientos para el diseño.

REFERENCIAS BIBLIOGRAFICAS.

Recursos encontrados en internet⁶:

www.cirque.com/pages/?section=3&page=19

www.synaptics.com/decaf/utilities/ACF126.pdf

www.infor.uva.es/~mlaguna/is2/apuntes/3-casos.pdf

www.ingenierosoftware.com/analisisydiseno/casosdeuso.php

⁶ También disponible en CD que acompaña a este documento.

CAPITULO II.

DISEÑO E IMPLEMENTACION DEL HARDWARE.

Introducción:

Como parte del diseño del hardware, se hace una selección de los componentes electrónicos que serán utilizados en el circuito del controlador, de manera que el diseño sea óptimo y viable económicamente. El capítulo también describe el principio de operación de algunos componentes y ciertos módulos del microcontrolador utilizado. Específicamente se necesitan dos módulos PWM (Modulación por ancho de pulso), un módulo ADC (Convertidor analógico digital) con referencia externa de voltaje, módulos temporizadores y recursos de interrupción.

2.1.0 Etapa de control.

La etapa de control la constituye básicamente el microcontrolador, los optoacopladores utilizados solo brindan aislamiento entre la etapa de potencia y la de control.

2.1.1 Selección del microcontrolador.

Para la etapa de control se ha optado por emplear el PIC16F877A [ver anexo A.1] porque tiene dos módulos de salida PWM, cada uno para controlar uno de los motores, además de contar con un convertidor analógico – digital de 8 canales y varios puertos digitales. A continuación se listan las características y periféricos del microcontrolador PIC16F877A que son utilizados en este proyecto.

Características:

- Procesador de arquitectura RISC avanzada
- Juego de solo 35 instrucciones con 14 bits de longitud. (Todas ellas se ejecutan en un ciclo de instrucción, menos las de salto que tardan dos).
- Hasta 8K palabras de 14 bits para la Memoria de Programa.
- Hasta 368 Bytes de memoria de Datos RAM.
- Hasta 256 Bytes de memoria de Datos EEPROM.
- Hasta 14 fuentes de interrupción internas y externas.
- Pila de 8 niveles.
- Power-on Reset.
- Programación serie en circuito (ICSP)

Periféricos:

- Timer0: Temporizador-contador de 8 bits con pre-escalador de 8 bits
- Timer1: Temporizador-contador de 16 bits.
- Timer2: Temporizador-contador de 8 bits con pre-escalador y post-escalador.
- Dos módulos CCP (modo PWM)
- Convertidor A/D de 10 bits (8 canales).

La siguiente figura muestra el diagrama de encapsulado del PIC16F877A.

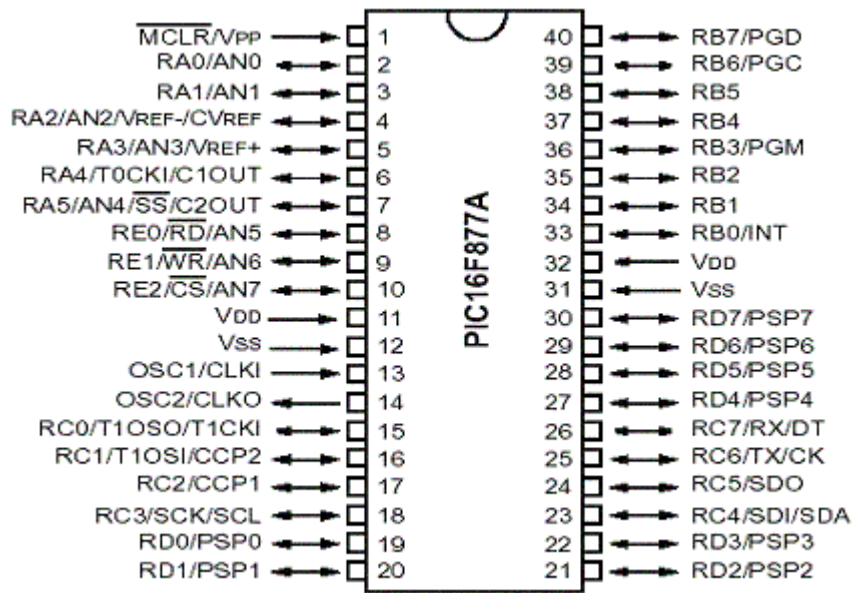


Figura 2.1 Diagrama de encapsulado para el PIC16F877A.

2.1.2 Funcionamiento y programación de los módulos Timer2 y CCP (en modo PWM).

Para generar las señales PWM se hará uso de dos módulos internos con que cuenta el PIC 16F877A, estos son: CCP1 y CCP2. En conjunto a estos se utilizarán dos salidas digitales del puerto C para controlar la dirección de giro de los motores DC, específicamente RC0 y RC3. De este modo es posible controlar la dirección y movimiento de la silla de ruedas.

Las señales de salida PWM en el microcontrolador PIC16F877A corresponden a los pines 16 y 17, nótese la función multiplexada de estos pines en la figura 2.1.

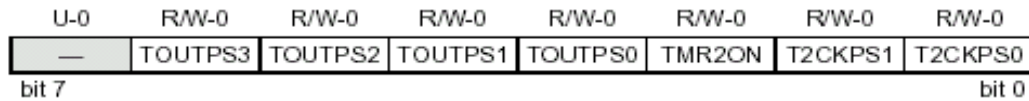
La operación del módulo CCP2 en modo PWM es idéntica a CCP1, por lo que solo se mostrarán los registros asociados al módulo CCP1.

2.1.2.1 Pasos necesarios para configuración y operación PWM del módulo CCP1.

- 1) Coloque el periodo PWM escribiendo al registro PR2.
- 2) Coloque el Duty Cycle escribiendo al registro CCPR1L y a los bits <5:4> de CCP1CON.
- 3) Hacer el pin CCP1 una salida, inicializando el bit 2 del registro TRISC.
- 4) Coloque el valor del pre-escalador de TMR2 y habilite TIMER2 escribiendo a T2CON.
- 5) Configurar módulo CCP1 para operación PWM escribiendo a CCP1CON

TIMER2: es un temporizador de 8 bits que se emplea como la base de tiempo (periodo) PWM de los módulos CCP. Este modulo dispone de los siguientes tres registros:

T2CON: Registro de control del Timer2.

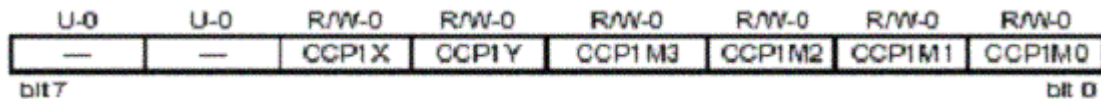


- Bit 7 No implementado.
- Bit 6-3 Bits de selección de post-escalador de salida de Timer2.
- Bit 2 Bit de encendido de Timer2.
- Bit 1-0 Bits de selección del pre-escalador de reloj de Timer2.

PR2: Registro de periodo de Timer2 (8 bits).

TMR2: Registro del modulo Timer2. (8 bits)

Como se mencionó anteriormente, la configuración del modulo CCP1 en modo PWM se hace a través del registro CCP1CON.



Registro CCP1CON.

Los bits CCP1M3: CCP1M0 controlan el modo de operación del modulo CCP1. Para modo PWM los bits CCP1M3 y CCP1M2 deben ser puestos a 1, el estado de los bits CCP1M1 y CCP1M0 no interesan en este modo.

El periodo PWM es especificado escribiendo al registro PR2 de TIMER2. Las hojas técnicas del microcontrolador especifican una formula para calcular el periodo PWM de la siguiente forma:

$$\text{Periodo PWM} = [(\text{PR2} + 1)] * 4 * \text{TOSC} * \text{Valor de pre-escalador de TMR2. Ec.(2.1)}$$

Donde:

PR2: Registro de periodo.

TOSC: Periodo de la señal de reloj del microcontrolador.

pre-escalador de TMR2: escalamiento de la señal de reloj.

Generalmente tanto el periodo PWM como el Valor del pre-escalador de TMR2 son asumidos y la incógnita es el valor del registro de periodo PR2 de TIMER2.

El duty Cycle es especificado escribiendo al registro CCPR1L y a los bits <5:4> del registro CCP1CON.

El registro CCPR1L contiene los 8 bits más significativos y CCP1CON <5:4> los 2 bits menos significativos.

El registro CCPR1H y 2 bits internos son usados para salvaguardar doblemente el duty cycle PWM.

Cuando el temporizador TMR2 es igual a PR2 ocurren los siguientes tres eventos:

- TMR2 es inicializado.
- El pin CCP1 tendrá un estado lógico de 1 (excepción: si el duty cycle es cero, el pin CCP1 no podrá ser 1).
- El duty Cycle PWM es enganchado desde el registro CCPR1L hasta CCPR1H.

Cuando el valor concatenado de 10 bits formado por el registro TMR2 y 2 bits Q del reloj interno es igual al valor en el registro CCPR1H y 2 bits internos, el pin CCP1 se pone en estado lógico de 0, terminando con el duty cycle, ver figura 2.2

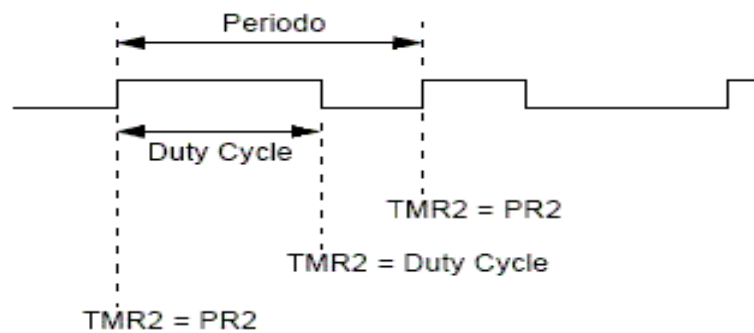


Figura 2.2 Salida PWM.

Para determinar el “duty cycle” se utilizan las siguientes expresiones:

$$\text{Duty cycle} = (\text{Valor decimal concatenado de CCPR1L y bits CCP1CON} < 5 : 4 > * \text{Tosc} * (\text{valor pre-escalador de TMR2}), \text{ en unidades de tiempo} \quad \text{Ec.}(2.2)$$

$$\text{Duty cycle \%} = \frac{\text{Duty cycle}}{\text{Periodo PWM}} \times 100\% \quad \text{Ec.}(2.3)$$

La resolución del “duty cycle” puede ser hasta 10 bits o menos. Este parámetro indica con cuantos bits puede ser especificado el valor del “duty cycle”. La siguiente relación muestra como se determina:

$$\text{Resolución} = \frac{\log_{10}(F_{osc} / F_{pwm})}{\log_{10}(2)} \text{ bits} \quad \text{Ec.}(2.4)$$

Donde:

Fosc: Frecuencia del cristal utilizado por el microcontrolador.

Fpwm: Frecuencia de la señal PWM.

Cuando la resolución es de 10 bits se tiene el siguiente rango de valores para escribir al registro del duty cycle: (0x000: 0x3FF).

Cuando la resolución es de 8 bits se tiene el siguiente rango de valores para escribir al registro del duty cycle: (0x00: 0xFF). En este caso cualquier valor mayor que 0xFF dará siempre un duty cycle del 100%.

2.1.2.2 Cálculos realizados referentes a los módulos CCP.

Resolución duty cycle:

Mediante la ecuación (2.4).

Resolución = $\log_{10}(16E6/15625) / \log_{10}(2)$ bits

Resolución = 10 bits.

Registro de periodo.

Despejando para el registro PR2 (registro de periodo) de ecuación (2.1) se obtiene:

$$PR2 = [(1/FPWM)/(4 * T_{osc} * \text{Valor pre-escalador de TMR2})] - 1 \quad \text{Ec. (2.5)}$$

FPWM = 15.625 KHz (este valor es de diseño⁷).

Pre-escalador de TMR2 = 1 (este valor es de diseño).

$$PR2 = (1/15625)/(4 * 1/16.00E+06 * 1) - 1$$

$$PR2 = 255 = 0xFF$$

2.1.3 Detalle de conexión de las salidas de control de dirección y velocidad.

RC0:	Salida digital para control de dirección motor derecho.
RC1/CCP1:	Salida PWM para control de velocidad motor derecho.
RC3:	Salida digital para control de dirección motor izquierdo.
RC2/CCP1:	Salida PWM para control de velocidad motor izquierdo.

Para aislar eléctricamente la etapa de potencia de la de control se han utilizado optoacopladores, esto permite proteger al microcontrolador de un exceso de corriente o de voltaje que este fuera de sus límites. La siguiente figura muestra la conexión de estos elementos con el microcontrolador.

⁷ Obtenida para una resolución de 10 bits, con un cristal de 16 MHz.

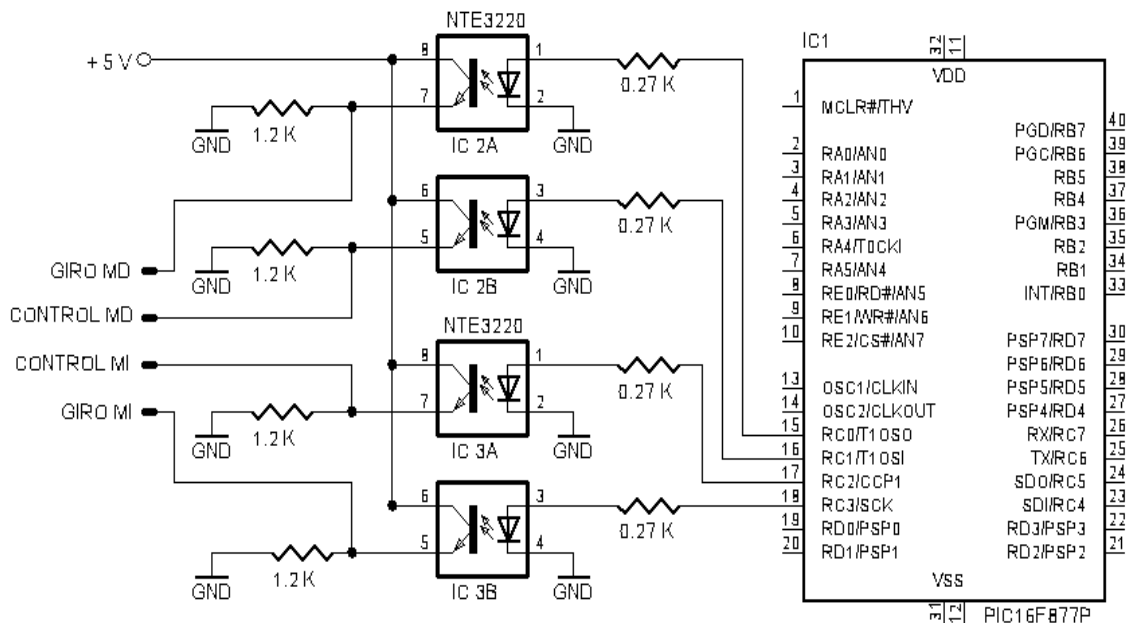


Figura 2.3. Conexión de las salidas de control de velocidad y dirección.

2.2.0 Etapa de entrada de mando.

Como se estableció en el capítulo I, como entrada de mando se utilizara un touchpad para el diseño del controlador realizado.

2.2.1 Selección del touchpad.

La mayoría de touchpads son empleados en computadoras portátiles (LAPTOPS), aunque se encontró que existen opciones para computadoras de escritorio (DESKTOP) y otras aplicaciones que no son para PC.

Existen varios fabricantes de touchpad entre los cuales podemos mencionar: Cirque, ALPS, SYNAPTICS, TOSHIBA, DELL y otros.

En la Tabla siguiente se muestran las principales características de algunos modelos de touchpad ofrecidos por tres fabricantes.

Tabla 2.1 Lista de touchpad ofrecidos por tres fabricantes.

Fabricante	Parte Numero	Area de superficie (mm)	Area activa (mm)	Profundidad (mm)	Puerto		
					PS/2	Serie	USB
Cirque	TSM9910	34.8 x 28.3	33.0 x 26.4	4.0	X		
	TSM9920	65.0 x 49.0	62.5 x 46.5	5.5	X	X	
	TSM9925	65.0 x 49.0	62.5 x 46.5	5.5	X		X
	TSM9946	117 x 68.8	77.0 x 54.0	5.5	X	X	
	TSM9957	64.8 x 48.8	62.5 x 46.5	5.0	X		
	TSM9778	96.9 x 78.3	89.9 x 71.3	5.5	X	X	
Synactics	TM41PUG134	65.0 x 49.0	62.5 x 46.5	1.9	X		
ALPS	56AAA1965B	65.0 x 49.0	62.5 x 46.5	1.9	X		

Se escogió el protocolo de comunicación PS/2 como principal criterio de selección para del touchpad por las siguientes razones:

- Fácil implementación vía software.
- No requiere circuitería externa para adaptar los niveles de tensión de las señales utilizadas para comunicación.
- Mayor información publicada sobre el protocolo.

De esta manera se escogió el touchpad ALPS modelo 56AAA1965B.

A continuación se muestra la imagen de la parte inferior del touchpad (lado de los componentes).

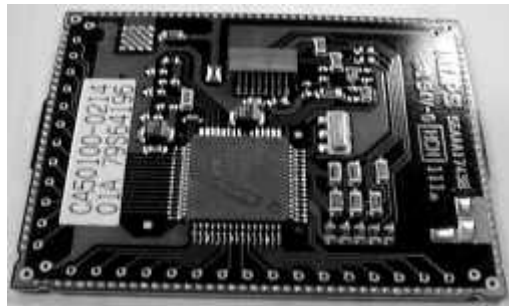


Figura 2.4. Touchpad ALPS modelo 56AAA1965B.

2.2.2 Características del touchpad Alps modelo 56AAA1965B.

- Delgado, liviano e instalación simple.
- No requiere mantenimiento.
- Superior navegación, alta respuesta.
- Inmune a la humedad.
- Reducido consumo de potencia.
- Soporta hasta dos botones mecánicos.

Como se mencionó el protocolo de comunicación del touchpad es el PS/2, este hace uso de cuatro señales para su operación, en el anexo A se encuentra la información relativa a este protocolo de comunicación. En la siguiente figura se puede apreciar la disposición de estas señales en el conector del touchpad así como sus dimensiones físicas.

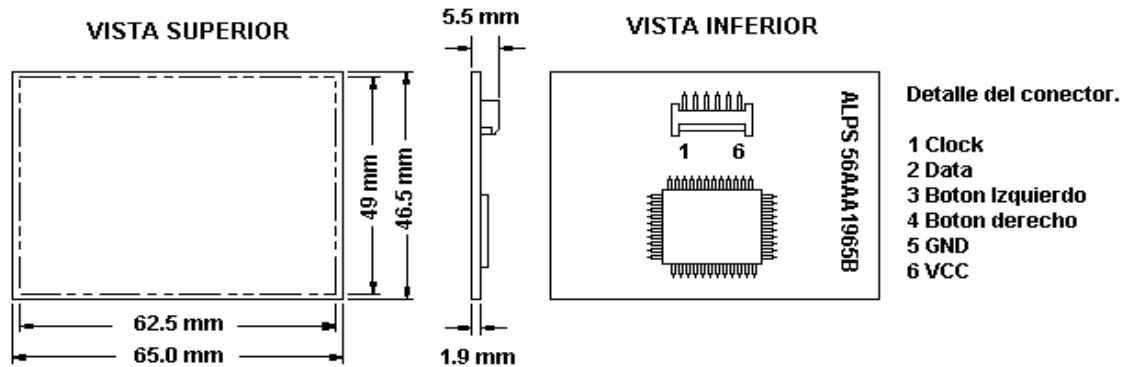


Figura 2.5. Dimensiones y detalle del conector del touchpad.

La conexión del touchpad con el microcontrolador es como se muestra a continuación:

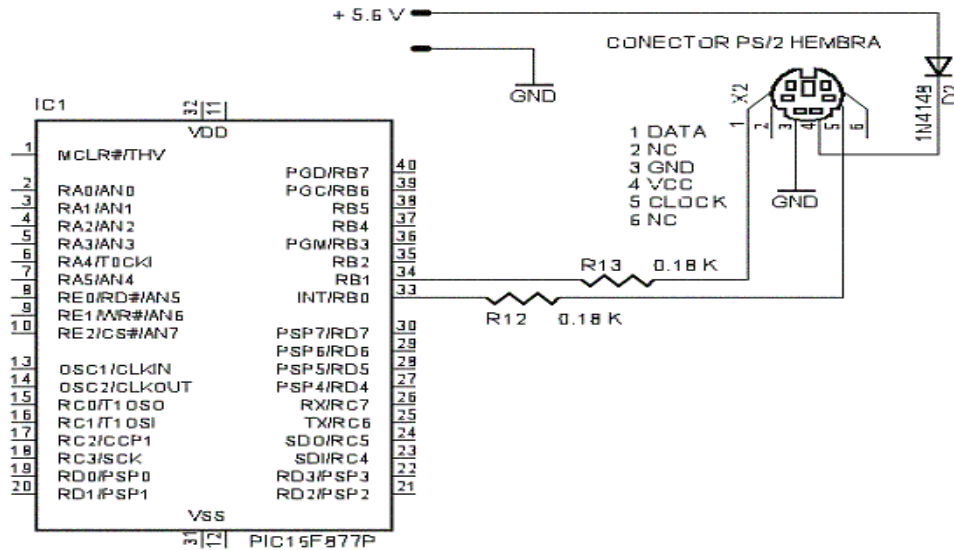


Figura 2.6. Conexión del touchpad con el microcontrolador.

2.2.3 TIPO DE TRANSMISIÓN/RECEPCIÓN.

Una transmisión/recepción síncrona no necesita emplear bits de principio y final para delimitar cada uno de los datos. La señal de reloj empleada por el receptor para sincronizar los bits se obtiene a partir de una señal independiente que debe acompañar a los datos desde el transmisor hasta el receptor.

La transmisión/recepción asíncrona requiere que cada uno de los datos vaya acompañado de un bit de arranque o inicio con una polaridad opuesta (0) al bit de paro (1).

Las transiciones de 1 a 0 se emplean para arrancar los temporizadores internos, que se usarán para identificar los bits de datos que siguen. El bit de paro que sigue a los bits de datos tiene la misma polaridad que la empleada para señalar la condición de reposo de la línea.

Con lo anteriormente expuesto podemos expresar que la transmisión/recepción de datos en el protocolo PS/2 es una combinación de tipo síncrona y asíncrona, porque utiliza una señal de reloj y se emplean bits de marca para delimitar los bits de datos.

La información encontrada referente a este protocolo establece que es un protocolo serial Síncrono.

Revisando la hoja técnica del microcontrolador, no se encontró ningún modulo interno que operara de esta manera, por lo que se opto por utilizar el recurso de interrupción por flanco a través de RB0. La señal que provocara la interrupción por flanco será la señal de reloj generada por el touchpad.

2.2.3.1 Recurso de interrupción por flanco.

Una interrupción es un evento que obliga al llamado de una subrutina que usualmente es llamada rutina de servicio a la interrupción.

La fuente de interrupción INT es sumamente importante para atender eventos externos en tiempo real, en nuestro caso serán los flancos de bajada de la señal de reloj del touchpad.

A continuación se detallan los pasos necesarios para programar este tipo de interrupción:

- a) Activar el bit de habilitación global de interrupción (GIE) del registro INTCON.
- b) Activar el bit de habilitación de interrupción externa (INTE) del registro INTCON.
- c) Seleccionar el flanco activo en RB0/INT por medio del bit de selección de flanco de interrupción (INTDEG) del registro OPTION_REG.

INTDEG =1, Selecciona interrupción por flanco de subida.

INTDEG =0, Selecciona interrupción por flanco de bajada.

En la sección de anexo C se pueden encontrar la descripción detallada de los registros mencionados.

Cuando en la línea RB0/INT se hace una petición de interrupción, entonces, de forma automática, el bit INTF del registro INTCON se pone a 1, este bit representa la bandera de interrupción INT y se utiliza para indicar que ha ocurrido una interrupción debido a la ocurrencia del flanco seleccionado, de este modo en la rutina de servicio a la interrupción se ejecuta el segmento de programa apropiado para atender esa interrupción.

Antes de regresar al programa principal, el segmento del programa de atención a la interrupción INT, debe borrar la bandera INTF, puesto que en caso contrario al ejecutar la instrucción de retorno de interrupción RETFIE se volverá a desarrollar el mismo proceso de interrupción, esto es aplicable para todas las interrupciones del PIC.

2.3.0 Etapa de medición de distancias.

2.3.1 Selección del sensor medidor de distancias por infrarrojo.

Existen varios fabricantes de sensores entre los cuales se encuentra Sharp, en la tabla 2.2 se muestran algunos modelos de sensores medidores de distancias por infrarrojo diseñados por este fabricante.

Tabla 2.2. Sensores Sharp medidores de distancia por infrarrojo.

Modelo	Salida	Distancia mínima	Distancia máxima	Corriente en estado activo	Corriente en estado inactivo
GP2D02	Digital	10 cm.	80 cm	-22mA	-3mA
GP2D05	Digital	Ajustable	Fija en 24 cm	-10mA	-3mA
GP2D12	Análoga	10 cm.	80 cm	-33mA	-33mA
GP2D15	Digital	-	Fija en 24 cm ± 3 cm	-33mA	-33mA
GP2D120	Análoga	4 cm.	30 cm	-33mA	-33mA
GP2Y0A02YK	Análoga	15 cm.	150 cm	-33mA	-33mA
GP2Y0D02YK	Digital	-	Fija en 80 cm	-33mA	-33mA

De los sensores anteriores mostrados en la tabla 2.2, podemos seleccionar el sensor Sharp modelo GP2Y0A02YK por tener mayor distancia de medición (15 cm - 150 cm).

2.3.2 Características del sensor Sharp modelo GP2Y0A02YK.

- Menor influencia en los colores de los objetos reflejados y su reflectividad, debido al método de medición del triángulo óptico.
- Rango de detección 15 a 150 cm.
- No es necesario un circuito de control externo, su salida puede ser conectada directamente al microcontrolador⁸.

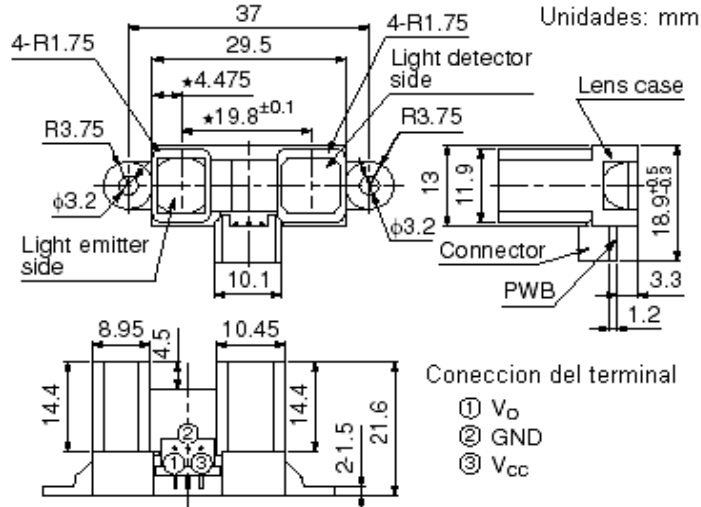
En la figura 2.7 se muestra la imagen física del sensor, así como sus dimensiones.

El sensor medidor de distancia frontal se conectara a la entrada análoga AN0 del microcontrolador. Mientras que el sensor trasero se conectara a la entrada análoga AN1.

⁸ El microcontrolador requiere referencia externa de voltaje, debido al valor máximo de salida del sensor.



(a) Apariencia física.



(b) Dimensiones físicas

Figura 2.7. Apariencia y dimensiones físicas del sensor Sharp GP2Y0A02YK

La conexión del touchpad con el microcontrolador es como se muestra a continuación:

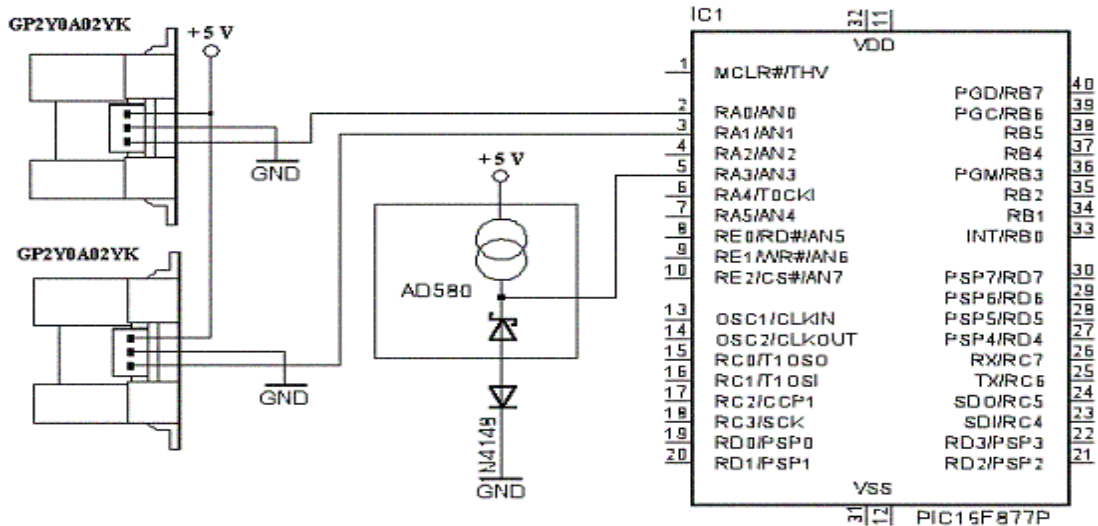
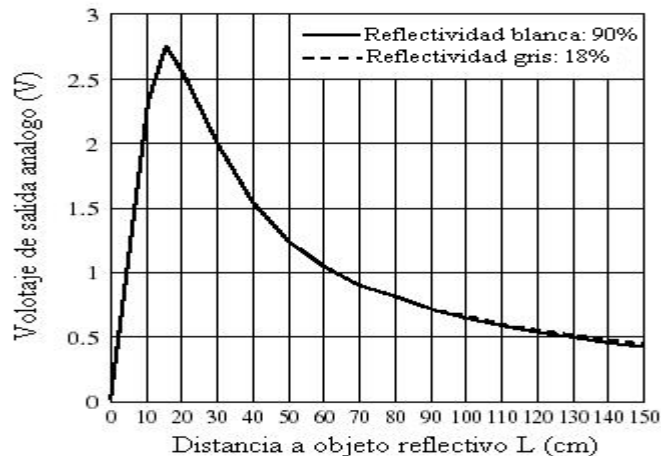


Figura 2.8. Conexión de los sensores con el microcontrolador.

2.3.3 Salida no lineal del sensor GP2Y0A02YK.

La salida de este sensor es no lineal con respecto a la distancia medida. En la figura 2.9 se muestra la salida típica, en la cual se puede apreciar que la salida del sensor dentro del rango válido (15 cm – 150 cm) tiene tendencia un tanto logarítmica, mientras que en el rango no válido (menor que 15 cm) la salida cae rápidamente y comienza a verse parecida a una lectura de larga distancia.

Figura 2.9 Curva de Voltaje de Salida Análogo vs. Distancia a objeto reflectivo.



Este voltaje de salida no lineal se envía a la etapa de control, donde es convertido a formato digital por un convertidor analógico digital (recurso auxiliar del microcontrolador). Será esta etapa quien se encargue de interpretar esta información.

2.3.4 Principio de operación del sensor GP2Y0A02YK.

El sensor medidor de distancia GP2Y0A02YK es un dispositivo óptico que aplica el principio de triangulación, mostrado en figura 2.10. La luz emitida por un LED infrarrojo pasa a través de lentes de proyección para entregar una luz con direccionalidad excepcionalmente aguda, e irradiarla al objeto de detección. La luz es difusamente reflejada desde el objeto de detección, y si la luz incidente es concentrada sobre un elemento de recepción de luz a través de unos lentes, entonces la posición del punto de luz incidente sobre el elemento de recepción variara de acuerdo a la distancia hasta el objeto. La distancia hasta el objeto de detección puede ser encontrada detectando este punto de luz incidente. Un detector sensitivo de posición (PSD), es usado como elemento de recepción de luz para detectar eléctricamente la posición del punto de luz incidente.

El PSD es una aplicación de fotodiodos de silicio.

En el diseño se propone utilizar dos sensores de este tipo, uno en la parte de enfrente y el otro colocado en la parte posterior de la silla de ruedas.

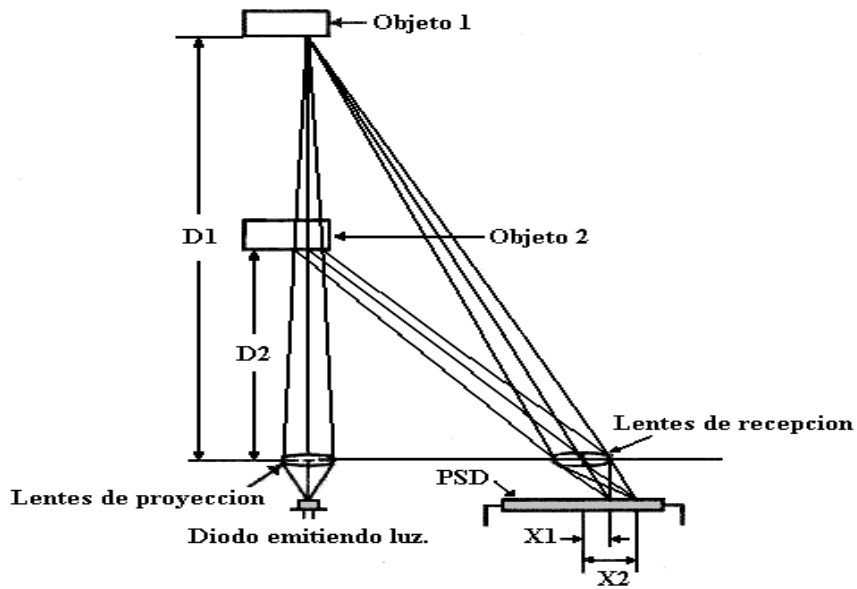


Figura 2.10. Sistema óptico de medición de distancias.

La figura anterior muestra como varía el punto de luz incidente sobre el PSD dependiendo de la distancia del sensor al objeto reflejado. En el caso del objeto 1, la posición X_1 del punto de luz incidente sobre el PSD es menor comparada con la posición X_2 producida por el objeto 2.

2.3.5 Funcionamiento y programación del módulo A/D del microcontrolador PIC 16f877A.

El controlador de la silla de ruedas requiere el empleo de un convertidor análogo-digital para digitalizar la señal proveniente de los sensores medidores de distancia y así esta pueda ser procesada por la etapa de control, convenientemente el microcontrolador ya cuenta con convertidor analógico digital (ADC). A continuación se explicará la operación de módulo ADC y los registros involucrados para su programación.

2.3.5.1 Convertidor analógico-digital (ADC).

El microcontrolador PIC16F876A cuenta con este módulo para la conversión de señales análogas en formato digital, entre sus características más importantes podemos mencionar las siguientes:

- Cinco canales de entrada análogos.
- Selección de referencia de voltaje interna o externa.
- Diez bits de resolución del resultado de la conversión.
- Interrupción por finalización de conversión.

Aunque existan cinco entradas análogas, se debe hacer notar que no es posible seleccionar más de una a la vez, ya que solamente existe un ADC para los cinco canales de entrada análogos. En la figura 2.11 se muestra claramente esta situación.

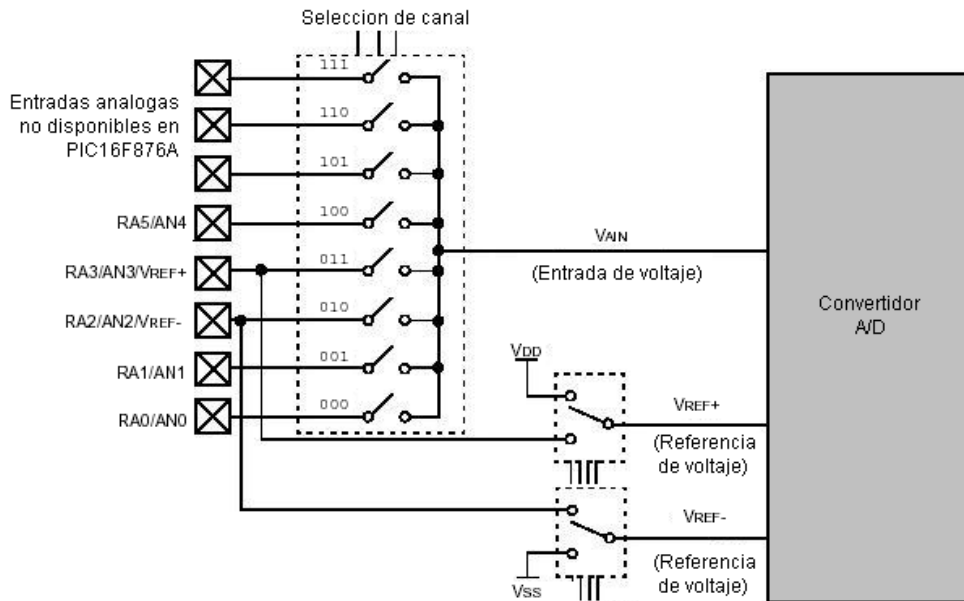


Figura 2.11. Diagrama de bloque del módulo A/D.

2.3.5.2 Referencia de voltaje externa.

Esta establece un límite de voltaje alto para el convertidor, es decir el máximo valor posible de conversión será el voltaje de referencia (este no debe sobrepasar 5 Vdc).

Cuando se aplique un voltaje a una entrada analógica y este sea mayor que el de referencia, el ADC no convertirá este sobrepaso de voltaje, sino que siempre dará una lectura equivalente al valor de la referencia de voltaje.

En la figura 2.9 se muestra la curva de voltaje de salida del sensor medidor de distancias, en esta se puede apreciar que el máximo voltaje de salida posible es de 2.7V aproximadamente, por lo tanto se diseñara una referencia de voltaje con ese valor.

Por facilidad en el diseño se ha utilizado el AD580, esta referencia de voltaje de precisión proporciona +2.5 V, para aumentar este voltaje se conecto un diodo 1N4148 al terminal común del AD580, elevando así el nivel de voltaje.

El voltaje de referencia será: $V_{REF+} = 3.2 \text{ V}$. En la figura 2.8 se puede observar la conexión de estos elementos.

2.3.5.3 Registros del módulo A/D.

Para su operación el módulo ADC cuenta con cuatro registros, estos son: **ADRESH**, **ADRESL**, **ADCON1** y **ADCON0**.

ADRESH: Registro alto del resultado A/D.

ADRESL: Registro bajo del resultado A/D.

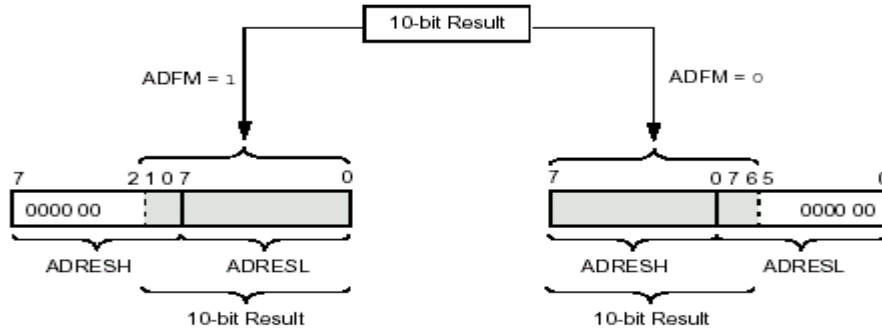


Figura 2.12. Selección de justificación derecha e izquierda del resultado.

El par de registros ADRESH-ADRESL contienen los 10 bits del resultado de la conversión. Por medio del bit ADFM de ADCON1 se puede seleccionar justificación izquierda o derecha para este resultado.

Cuando el bit ADFM de ADCON1 vale 1, se selecciona justificación derecha y cuando vale cero habilita justificación izquierda del resultado, ver figura 2.12.

ADCON1: Registro de control 1 A/D.

Configura entradas análogas/voltaje de referencia y Entradas/Salidas digitales.

R/W-0	R/W-0	U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0
ADFM	ADCS2	—	—	PCFG3	PCFG2	PCFG1	PCFG0
bit 7							bit 0

Bit 7 ADFM: Bit de selección de formato de resultado A/D.

Bit 6 ADCS2: Bit de selección de reloj de conversión. Se utiliza en conjunto con los bits ADCS1 y ADCS0 de ADCON0 para realizar esta función.

Bit 5-4 No implementados, leídos como '0'.

Bit 3-0 Bits de control de configuración del puerto A/D.

ADCON0: Registro de control 0 A/D.

Controla la operación del modulo A/D.

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	U-0	R/W-0
ADCS1	ADCS0	CHS2	CHS1	CHS0	GO/DONE	—	ADON
bit 7						bit 0	

Bit 7-6 ADCS1:ADCS0 Bits de selección de selección de reloj de conversión. Se utilizan en conjunto con el bit ADCS2 de ADCON1 para realizar esta función.

Bit 5-3 CHS2:CHS0 Bits de selección de canal análogo.

Bit 2 Bit de estado de conversión A/D.

Bit 1 No implementado, leído como '0'.

Bit 0 Bits de encendido A/D.

Para una completa descripción de estos registros ver anexo C.

2.3.5.4 Pasos necesarios para realizar una conversión A/D:

- 1) Configuración del modulo A/D.
 - Configurar entradas análogas/voltaje de referencia y Entradas/Salidas digitales (ADCON1).
 - Seleccionar canal de entrada (ADCON0).
 - Seleccionar reloj de conversión A/D.
 - Encender modulo A/D.
- 2) Configurar interrupción A/D.
 - Inicializar bit de bandera de interrupción del convertidor ADIF (Bit 6 de registro PIR1).
 - Activar bit de habilitación de interrupción del convertidor ADIE (Bit 6 de registro PIE1).
 - Activar bit de habilitación de interrupción periférica PEIE (Bit 6 de registro INTCON).
 - Activar bit de habilitación de interrupción global GIE (Bit 7 de registro INTCON).
- 3) Esperar el tiempo de adquisición (Tiempo que una entrada debe permanecer seleccionada antes de comenzar la conversión.)
- 4) Iniciar conversión.
 - Active bit GO/DONE (ADCON0).
- 5) Esperar finalización de conversión.
 - Esperar para interrupción del modulo A/D.
- 6) Leer el resultado A/D del par de registros (ADRESH:ADRESL), inicializar bit ADIF (PIR1).

La siguiente figura muestra los secuencia de los tiempos de adquisición y conversión.

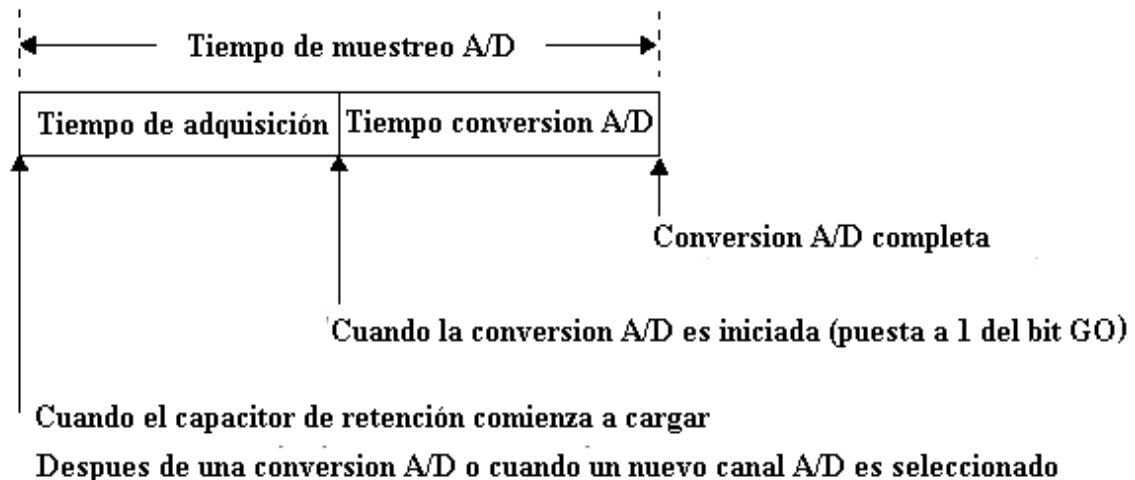


Figura 2.13 Secuencia de conversión A/D.

2.3.5.5 Cálculos realizados referentes a los módulos AD, TMR0 y TMR1.

Frecuencia de cristal utilizado = 16.00 MHz.

MODULO A/D.

Tiempo de adquisición.

$$TACQ = TAMP + TC + TCOFF. \quad \text{Ec. (2.6)}$$

Donde:

TACQ: Tiempo de adquisición.

TAMP: Tiempo de asentamiento del amplificador.

TC: Tiempo de carga del capacitor de retención.

TCOFF: Coeficiente de temperatura.

TAMP = 2 μ seg. (Dato de hoja tecnica del microcontrolador)

$$TC = -CHOLD (RIC + RSS + RS) \ln(1/2047) \quad \text{Ec. (2.7)}$$

Donde:

CHOLD: Capacitor de retención de carga.

RIC: Resistencia de interconexión.

RSS: Resistencia interna del interruptor de muestreo.

RS: Resistencia interna de la fuente externa.

$TC = (-120e-12) * (1000 + 7000 + 10000) * \ln(1/2047) = 16.47 \mu\text{seg.}$

$TCOFF = [(Temperatura - 25^\circ\text{C})(0.05 \text{ us}/^\circ\text{C})]$

$TCOFF = [(50^\circ\text{C} - 25^\circ\text{C})(0.05 \text{ us}/^\circ\text{C})] = 1.25 \text{ useg}$

$TACQ = TAMP + TC + TCOFF.$

$TACQ = 2 \mu\text{seg} + 16.47 \mu\text{seg} + 1.25 \text{ useg} = 19.72 \mu\text{seg.}$

Tiempo de adquisición (programa) = 30us

Para una correcta conversión A/D, el reloj de conversión A/D debe ser seleccionado para asegurar un tiempo mínimo de conversión por bit (TAD) de 1.6 μ seg.

$TAD = 64 * TOSC$

$TAD = 64 * (1 / 16.00 \text{ E}+06) = 4.0 \mu\text{seg}$

El bit GO (bit 2 de ADCON0) que inicia la conversión se activa cada 5 ms, este tiempo es controlado por el temporizador TMR0.

MODULO TMR0.

Temporización TMR0: 5 ms

Pre-escalador TMR0 = 128;

$$\text{Temporización} = 4 * TOSC * (256 - \text{Valor a cargar en TMR0}) * \text{pre-escalador TMR0} \quad \text{Ec. (2.8)}$$

Despejando para valor a cargar en TMR0 se obtiene:

$$\text{Valor a cargar en TMR0} = 256 - \text{Temporización} / (4 * \text{Tosc} * \text{pre-escalador TMR0}) \quad \text{Ec. (2.9)}$$

$$\text{TMR0} = 256 - 0.005 / (4 * 1 / 16.00E+06 * 128) = 100 = 0x64.$$

MODULO TMR1.

El temporizador TMR1 controlara el tiempo máximo en que se determine la ocurrencia de los siguientes estados:

- Envío de las coordenadas de movimiento del touchpad.
- Botón derecho presionado (movimiento avance recorrido largo).
- Ninguna coordenada de movimiento enviada por el touchpad.

Temporizacion TMR1: 100 ms

Prescaler = 8

$$\text{Temporización} = 4 * \text{TOSC} * (65535 - \text{Valor a cargar en TMR1}) * \text{pre-escalador TMR1} \quad \text{Ec. (2.10)}$$

Despejando para valor a cargar en TMR1 se obtiene:

$$\text{Valor a cargar en TMR1} = 65535 - \text{Temporización} / (4 * \text{Tosc} * \text{pre-escalador TMR1}) \quad \text{Ec. (2.10)}$$

$$\text{TMR1} = 65535 - 0.1 / (4 * 1 / 16.00E+06 * 8) = 15535$$

$$\text{TMR1} = 0x3CAF$$

2.3.6 Asignación de duty cycle en base a las lecturas del sensor medidor de distancia.

Como se describió en la sección 1.4, se obtuvo mediciones del voltaje de salida del sensor, variando la distancia entre un obstáculo y el sensor, dentro del rango de detección menor a 150cm. Luego se dividió el 100% del “duty cycle” en 25 particiones, para asignarlas a ciertas distancias del rango de detección del sensor, de modo que los valores mayores de “duty cycle” fueron asignados a los casos en que el obstáculo este mas separado del sensor (dentro del rango de detección).

A continuación se muestran los resultados obtenidos en la Tabla 2.3:

Tabla 2.3 Relación entre el voltaje de salida del sensor y la asignación de “duty cycle” empleado para referencia.

MODULO A/D				DUTY CYCLE		
Distancia (cm)	Voltaje (voltios) ⁹	Valor hexa	Valor binario	Duty cycle %	Valor hexa	Valor binario
150	0.4009	084	0010000100	100	3FF	1111111111
145	0.4222	08B	0010001011	95	3CD	1111001101
140	0.4404	091	0010010001	90	39A	1110011010
135	0.4708	09B	0010011011	85	367	1101100111
130	0.4920	0A2	0010100010	80	334	1100110100
125	0.5102	0A8	0010101000	75	300	1100000000
120	0.5224	0AC	0010101100	70	2CD	1011001101
115	0.5315	0AF	0010101111	65	29A	1010011010
110	0.5619	0B9	0010111001	60	267	1001100111
105	0.5801	0BF	0010111111	55	234	1000110100
100	0.5922	0C3	0011000011	50	200	1000000000
95	0.6621	0DA	0011011010	45	1CD	0111001101
90	0.7107	0EA	0011101010	40	19A	0110011010
85	0.7411	0F4	0011110100	35	167	0101100111
80	0.7714	0FE	0011111110	30	134	0100110100
75	0.8413	115	0100010101	25	100	0100000000
70	0.8717	11F	0100011111	20	0CD	0011001101
65	0.8929	126	0100100110	18	0B9	0010111001
60	1.0205	150	0101010000	16	0A4	0010100100
55	1.0903	167	0101100111	14	090	0010010000
50	1.2118	18F	0110001111	12	07B	0001111011
40	1.4700	1E4	0111100100	10	067	0001100111
30	1.9620	286	1010000110	8	052	0001010010
22	2.3295	2FF	1011111111	5	034	0000110100
0	2.7425	387	1110000111	0	000	0000000000

2.4.0 Etapa de alimentación.

Para la fuente de alimentación se ocupara una batería de automóvil con las siguientes características:

Tensión nominal: 12 VDC.

Capacidad: 35 A/H.

La capacidad es medida en amperios-hora, representa el numero de horas que la batería puede suplir una cierta cantidad de corriente antes de que el voltaje caiga abajo de su valor nominal.

Por ejemplo la batería con capacidad 35 A/H significa que la batería puede suplir continuamente una corriente de 35A por una hora antes de descargarse.

⁹ Valores de voltaje obtenidos experimentalmente.

La tensión de 12 V se ocupara para la etapa de potencia de los motores DC. Además de este voltaje se necesita una fuente secundaria que proporcione un voltaje de 5.6 V y 5V. Estos voltajes se emplearan para alimentar el controlador construido y parte de la lógica de control del circuito de potencia.

El regulador de voltaje utilizado para tal propósito es el 7805, que tiene una capacidad máxima de corriente de 1A.

A continuación se muestra el diagrama de la fuente de alimentación:

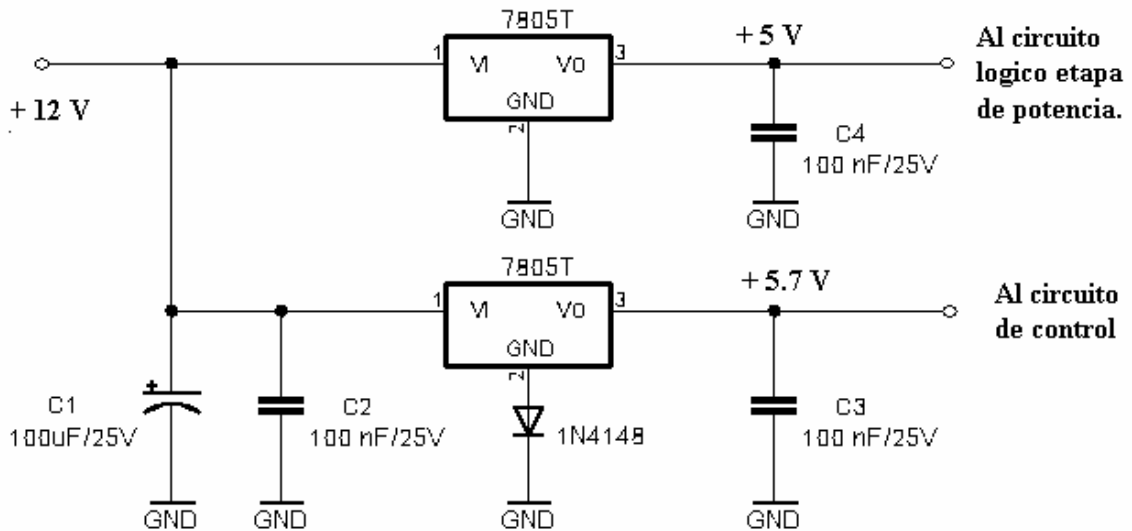


Figura 2.14. Diagrama de la fuente de alimentación.

2.5.0 Esquemas del controlador.

A continuación se muestra el diagrama esquemático del controlador diseñado, vista de la placa de circuito impreso (creadas a partir del programa EAGLE versión 4.11) e imágenes del prototipo del controlador construido.

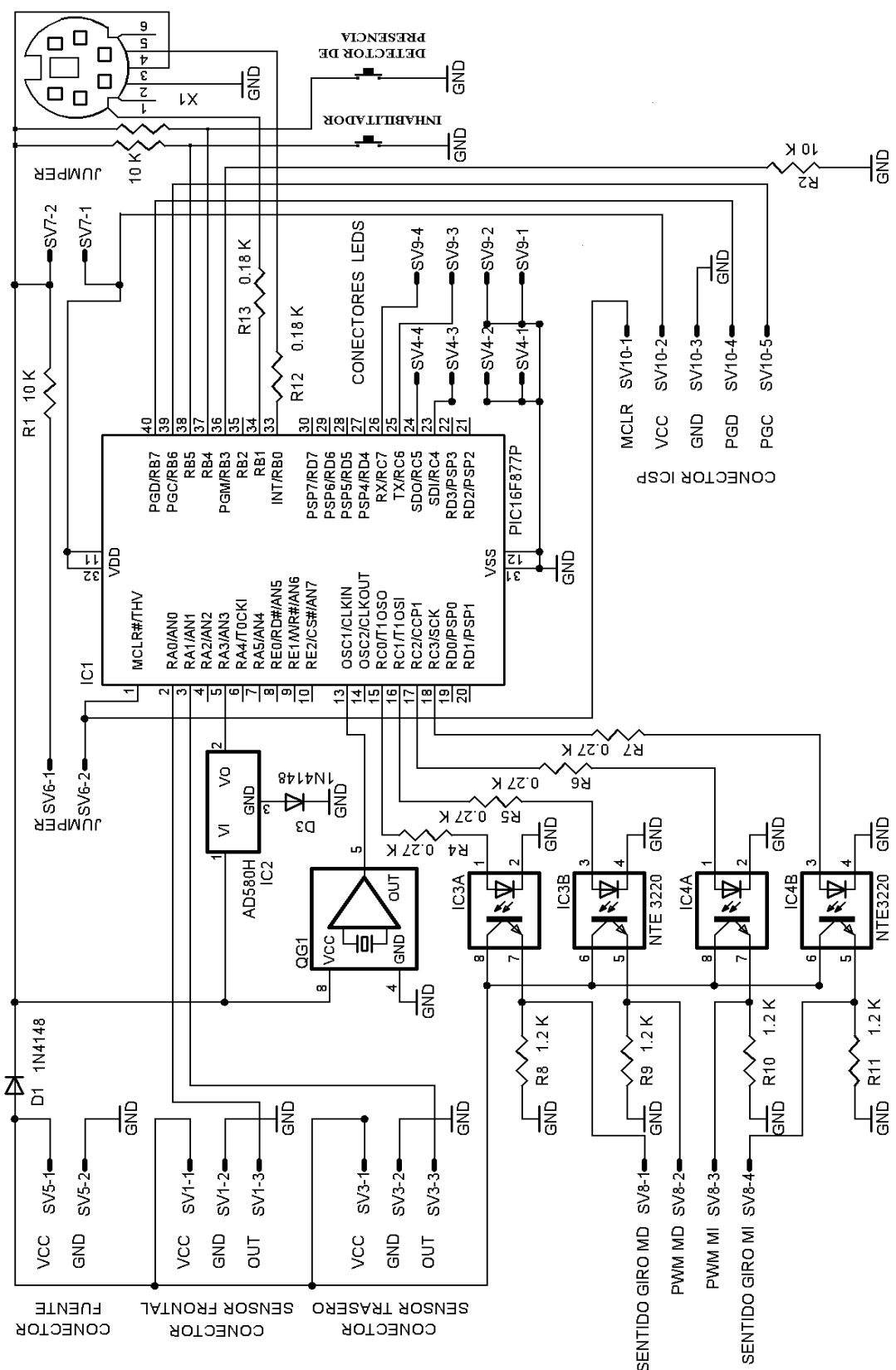


Figura 2.15 Diagrama esquemático del controlador.

Esquemáticos del circuito impreso

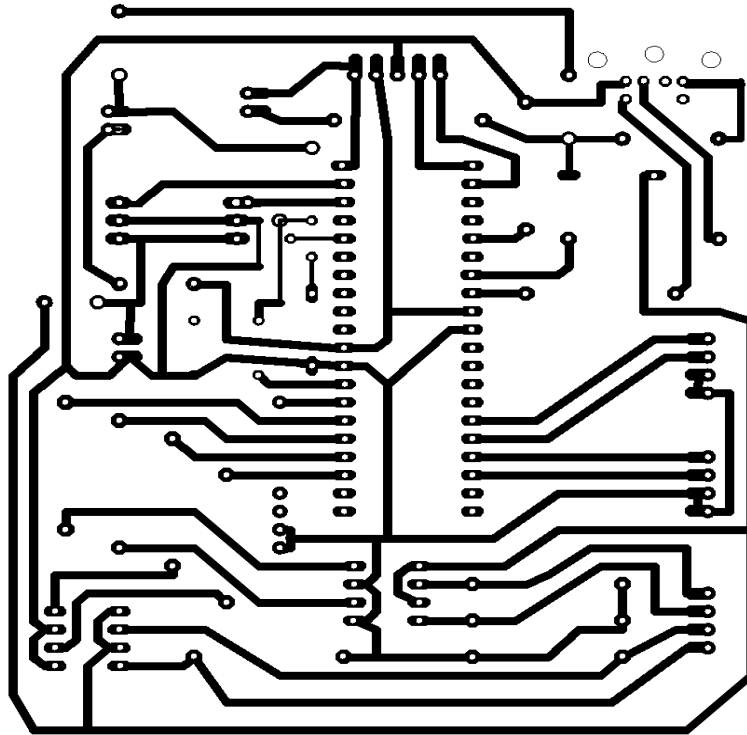


Figura 2.16(A) Vista del circuito impreso lado de las pistas.

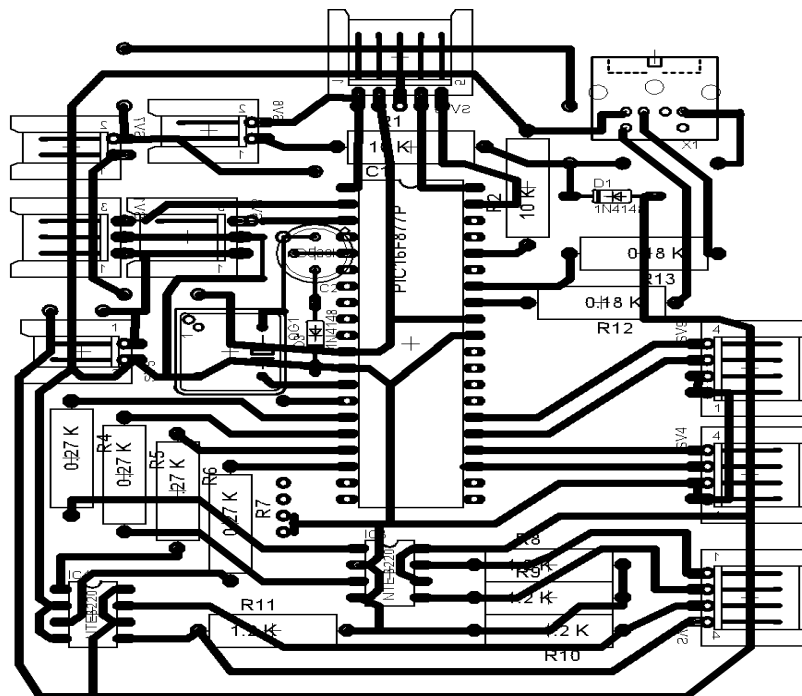


Figura 2.16(B) Vista del circuito impreso del lado de los componentes.

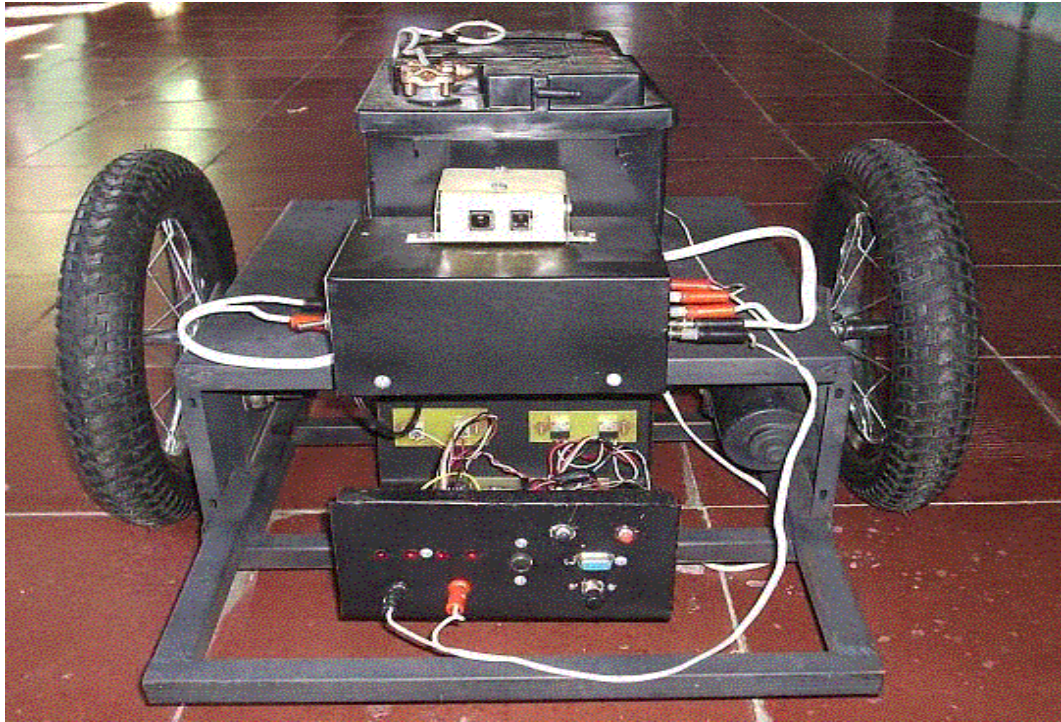


Figura 2.17 Vista trasera del prototipo construido.

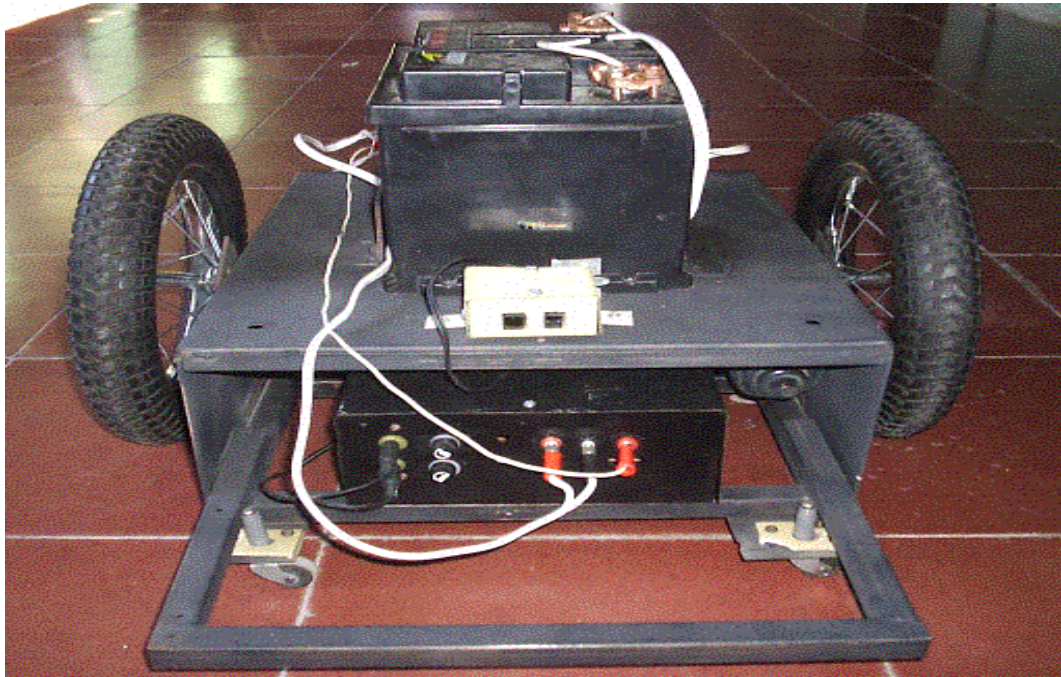


Figura 2.18 Vista frontal del prototipo construido



Figura 2.19 Vista lateral del prototipo construido.

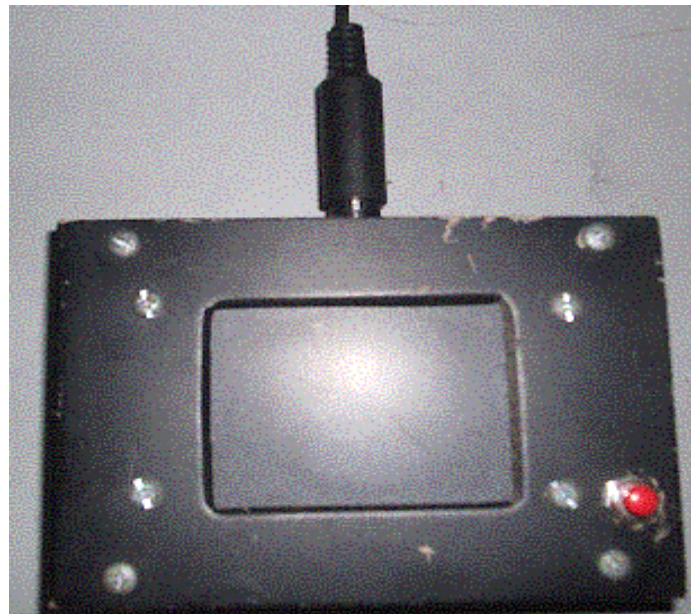


Figura 2.20 Vista frontal del touchpad.

2.6.0 Costo de materiales.

Tabla 2.4 Costo del controlador.

Cantidad	Descripción	Precio unitario (US\$)	Total (US\$)
1	Microcontrolador PIC16F877A	\$18.09	\$18.09
2	Optoacopladores NTE3220	\$1.15	\$2.30
1	AD580	-	-
2	Diodo 1N4148	\$0.15	\$0.30
1	Touchpad ALPS 56AAA1965B	\$28.95	\$28.95
2	Sensor Sharp GP2YA02YK	\$25.47	\$50.94
1	Resonador 16MHz.	\$3.50	\$3.50
12	Resistencias ¼ W	\$0.17	\$2.04
4	Leds rojos	\$0.17	\$0.68
2	Conector mini Din 6 pines hembra	\$0.64	\$1.28
2	Conector mini Din 6 pines macho	\$0.64	\$1.28
1	Conector hembra DB9	\$0.35	\$0.35
2	Conector hembra para fuente	\$0.20	\$0.40
4	Jack 3.5 Stereo de chasis	\$0.31	\$1.24
2	Jack Mono de chasis	\$0.33	\$0.66
4	Espigas Mono	\$0.12	\$0.48
4	Espigas Stereo	\$0.14	\$0.56
4	Espigas banana para fuente	\$0.20	\$0.80
1	Base de 40 pines	\$0.45	\$0.45
2	Base de 8 pines	\$0.23	\$0.46
1	Placa de circuito impreso 10x15 cm	\$0.68	\$0.68
1	Onza de percloruro de hierro.	\$1.00	\$1.00
	Costo de fabricación circuito impreso.	\$5.00	\$5.00
10	Yardas de estaño	\$0.17	\$1.70
	Total		\$123.14

Tabla 2.5 Costo fuente de alimentación.

Cantidad	Descripción	Precio unitario (US\$)	Total (US\$)
2	Regulador de voltaje positivo UA-7805	\$0.51	\$1.10
1	Diodo 1N4148	\$0.15	\$0.15
1	Led rojo	\$0.17	\$0.17
1	Interruptor EC-1503	\$0.51	\$0.51
7	Conector hembra para fuente	\$0.20	\$1.40
2	Espigas banana para fuente	\$0.20	\$0.40
1	Batería para automóvil 12 V, 35 A/H	\$44.95	\$44.95
2	Terminal de cobre para batería	\$1.63	\$3.26
	Total		\$51.94

Total Costo del controlador y fuente de alimentación = US\$ 175.08

CONCLUSIONES DEL CAPITULO II.

- El touchpad puede ser perfectamente utilizado como entrada de mando, ya que permite efectuar los movimientos previstos a través de su empleo: avance, retroceso, giro izquierdo y giro derecho.
- En la selección de componentes, se tomo en cuenta las características y desempeño de los mismos, así como también el costo, para que el proyecto sea competitivo económicamente.
- Se utilizo el microcontrolador PIC 16F877A porque cuenta con los recursos mínimos necesarios para la implementación del controlador, de modo que la utilización de sus recursos son aprovechados al máximo.
- Actualmente el campo de rango de los sensores medidores de distancia es bastante estrecho, esto limita la capacidad de detección de ciertos obstáculos por parte del controlador.

REFERENCIAS BIBLIOGRAFICAS.

Angulo Usategui, Jose Maria. MICROCONTROLADORES PIC DISEÑO PRACTICO DE APLICACIONES.
Segunda Edición, Mc Graw Hill
España [1999].

Manual de Referencia de la Familia de Microcontroladores de Rango Medio PIC micro¹⁰.
USA Microchip [2002].

Recursos encontrados en internet¹¹:

www.cirque.com/pages/?section=3&page=19

www.synaptics.com/decaf/utilities/ACF126.pdf

http://document.sharpsma.com/files/Optical_Sensors_AN.pdf

<http://info.hobbyengineering.com/specs/SHARP-GP2Y0A02YK.pdf>

www.nteinc.com/specs/3200to3299/pdf/nte3220.pdf

¹⁰ Disponible en el CD adjunto a este documento con el nombre DS31002A

¹¹ También disponible en CD que acompaña a este documento.

CAPITULO III

DISEÑO E IMPLEMENTACION DEL SOFTWARE.

Introducción.

El presente capítulo de este trabajo trata sobre el software desarrollado para el controlador de la silla de ruedas. Como se ha comentado al principio del trabajo, se implementó el protocolo PS/2 para poder establecer comunicación con el touchpad, utilizando el recurso de interrupción por flanco en RB0, además el programa es capaz de tomar decisiones de reducción de velocidad cuando los sensores detecten un obstáculo en la trayectoria de recorrido. El programa fue escrito completamente en lenguaje ensamblador.

3.1 Programa principal.

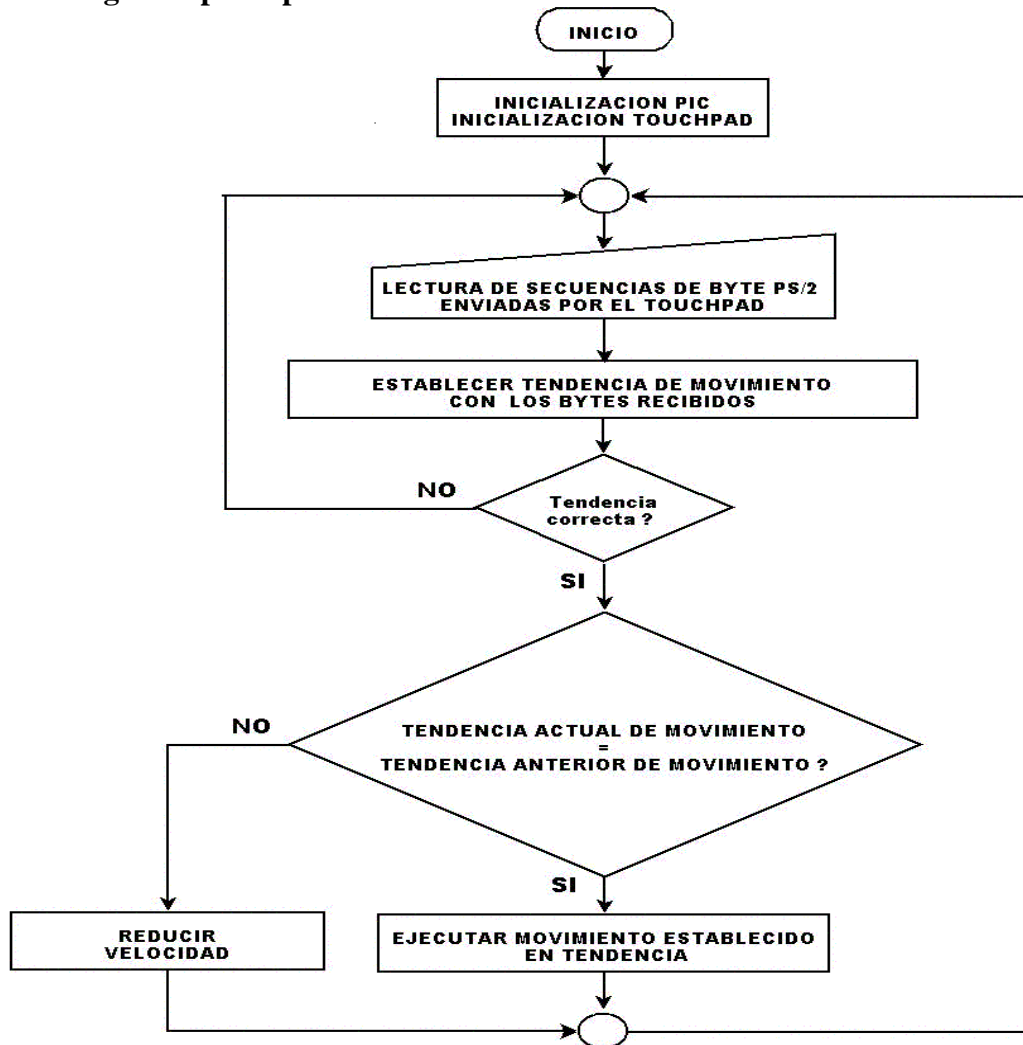


Figura 3.1.0 Diagrama de flujo principal.

La inicialización PIC, consiste en configurar los módulos CCP1 y CCP2 en modo PWM, configuración de puertos, entradas análogas e inicialización de variables utilizadas.

En inicialización del touchpad, se ejecuta una rutina de retardo de aproximadamente 1 seg para dar tiempo al touchpad de autocalibrarse, posteriormente son enviados los comandos PS/2 RESET y ENABLE para inicializar el touchpad y habilitar la transmisión de datos en modo relativo respectivamente.

Después se leen las secuencias de byte PS/2 enviadas por el touchpad para establecer una tendencia de movimiento, si esta tendencia actual coincide con la establecida en un movimiento anterior, se ejecuta el movimiento actual de lo contrario, se reduce la velocidad (señal PWM).

El proceso se repite obteniendo nuevas secuencias de coordenadas de movimiento.

3.1.1 Transmisión de datos del microcontrolador al touchpad.

Como se mencionó en la sección 2.2.3.1, la línea de reloj del touchpad se conecta al pin RB0, este pin se configura para provocar una interrupción INT en el microcontrolador cada vez que ocurra un flanco ya sea de bajada o de subida.

Cuando se quiere transmitir un comando PS/2, el microcontrolador pone en estado bajo la línea de reloj (estado de inhibición) durante un tiempo de 100 useg aproximadamente, luego pone en estado bajo la línea de datos y libera la línea de reloj (estado de solicitud de envío). En este punto se carga el valor del comando en la variable BYTE0 (8 bits) y se coloca una bandera para indicar que se está en modo de transmisión y se espera a que ocurra la interrupción, en esta rutina la línea de datos se coloca en estado lógico cero o uno dependiendo del valor de los bits de BYTE0, también se lleva un conteo de los flancos para transmitir el bit correspondiente y se debe abandonar la interrupción antes de que ocurra el próximo flanco de bajada.

En las figuras 3.2 y 3.3 se muestran los diagramas de flujo de la interrupción por flanco en modo transmisión.

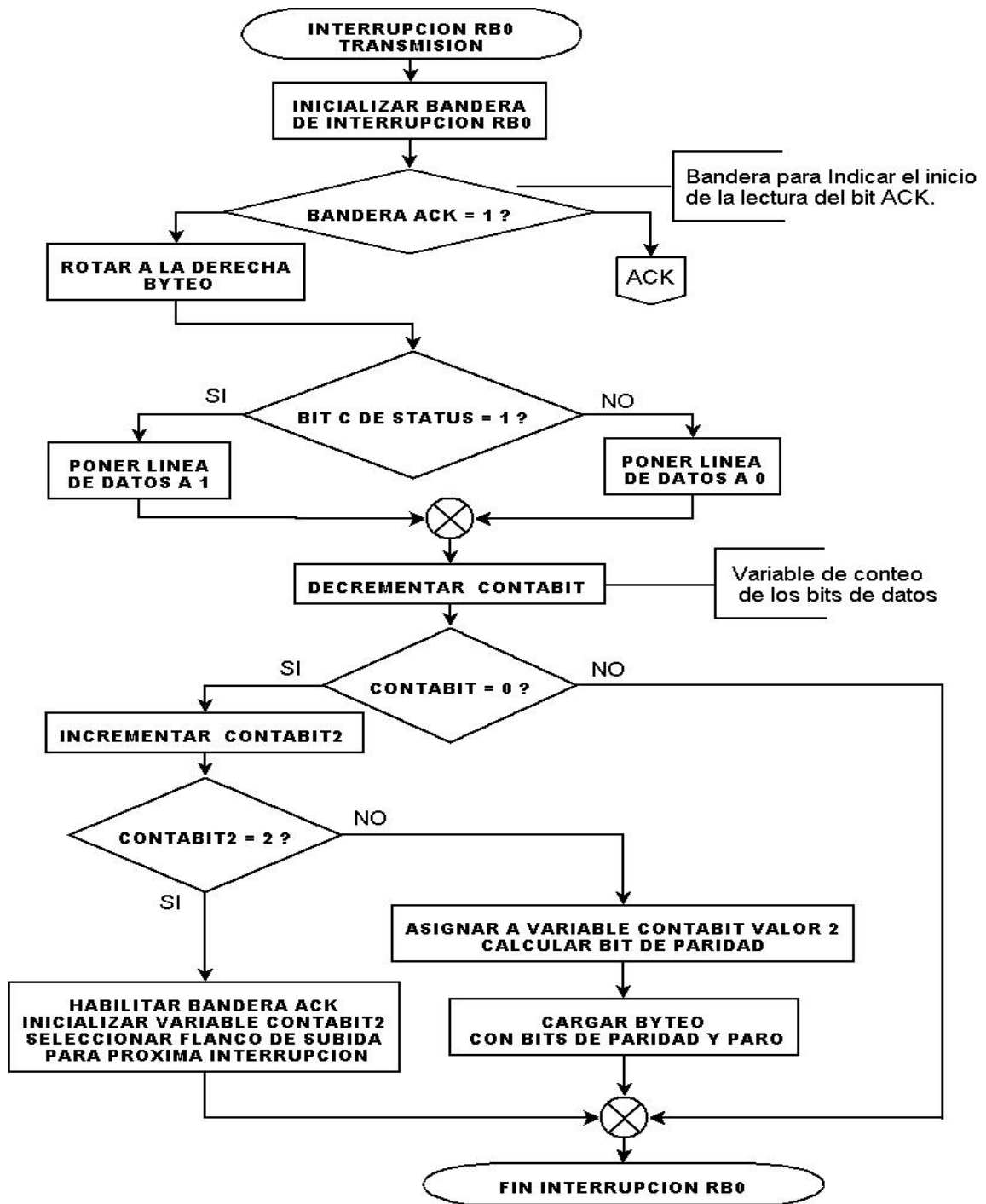


Figura 3.2 Diagrama de flujo para transmisión de datos del microcontrolador al touchpad.

Completada la transmisión, el touchpad envía un bit de reconocimiento para indicar que ha recibido el comando, este bit también se lee dentro de la rutina para chequear su valor.

En la continuación del diagrama figura 3.2 se muestra esta situación.

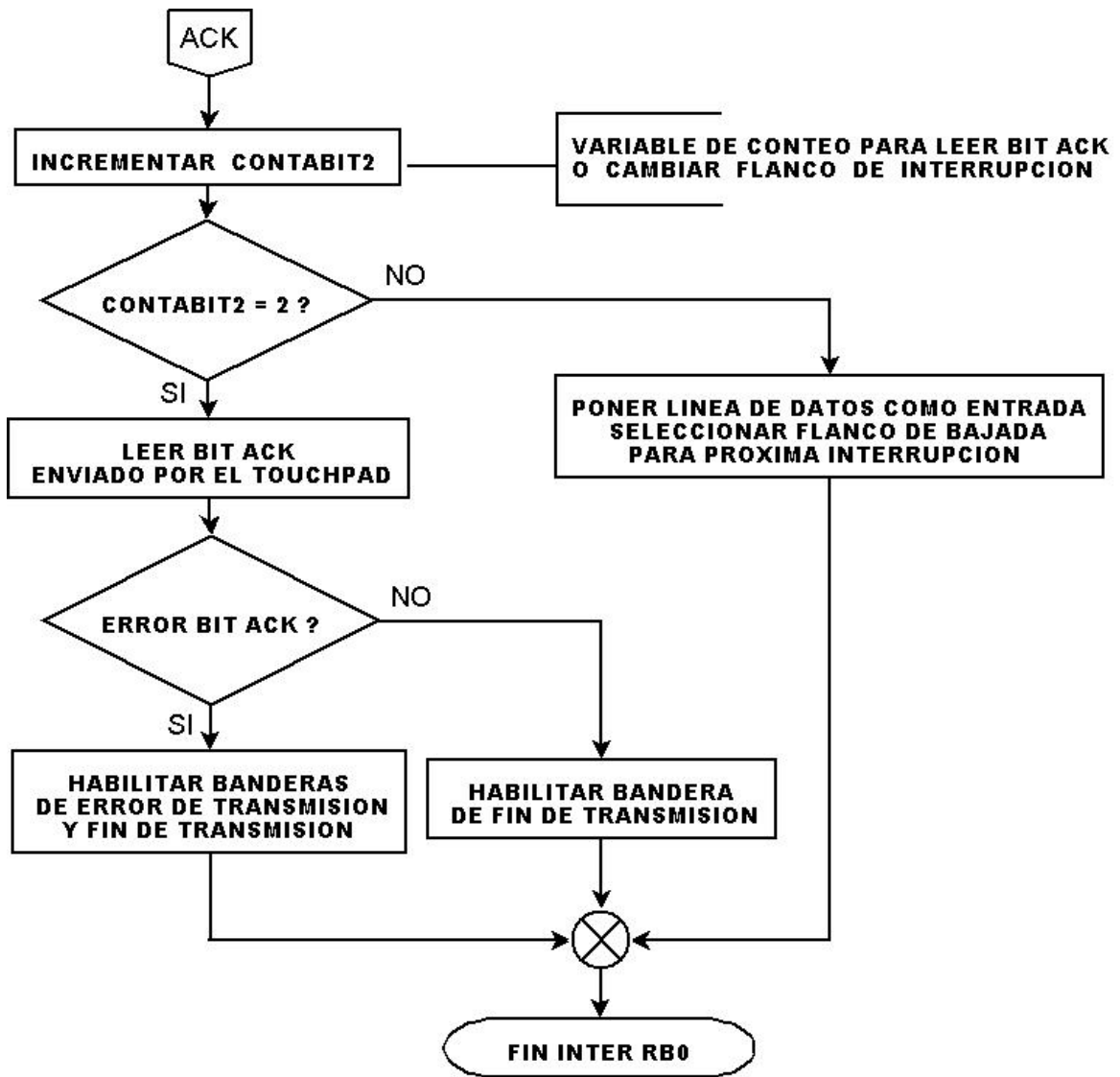


Figura 3.3 Continuación del diagrama de flujo para transmisión de datos del microcontrolador al touchpad.

3.1.2 Transmisión de datos del touchpad al microcontrolador.

La recepción de un byte PS/2 por parte del microcontrolador se produce de manera similar, se coloca una bandera para indicar que se esta en modo de recepción.

Dentro de la rutina de interrupción, se lee el estado de la línea de datos para obtener el valor del bit, si este es un bit de control se comprueba su valor para detectar un posible error de recepción, pero si es de datos se salva.

Los flancos de bajada se cuentan, para llevar un control de la recepción de los bits y se utiliza una variable adicional para contar el número de bytes que son recibidos y poder así salvarlos en el byte correspondiente.

En las figuras 3.4 se muestra el diagrama de flujo de la interrupción por flanco en modo recepción.

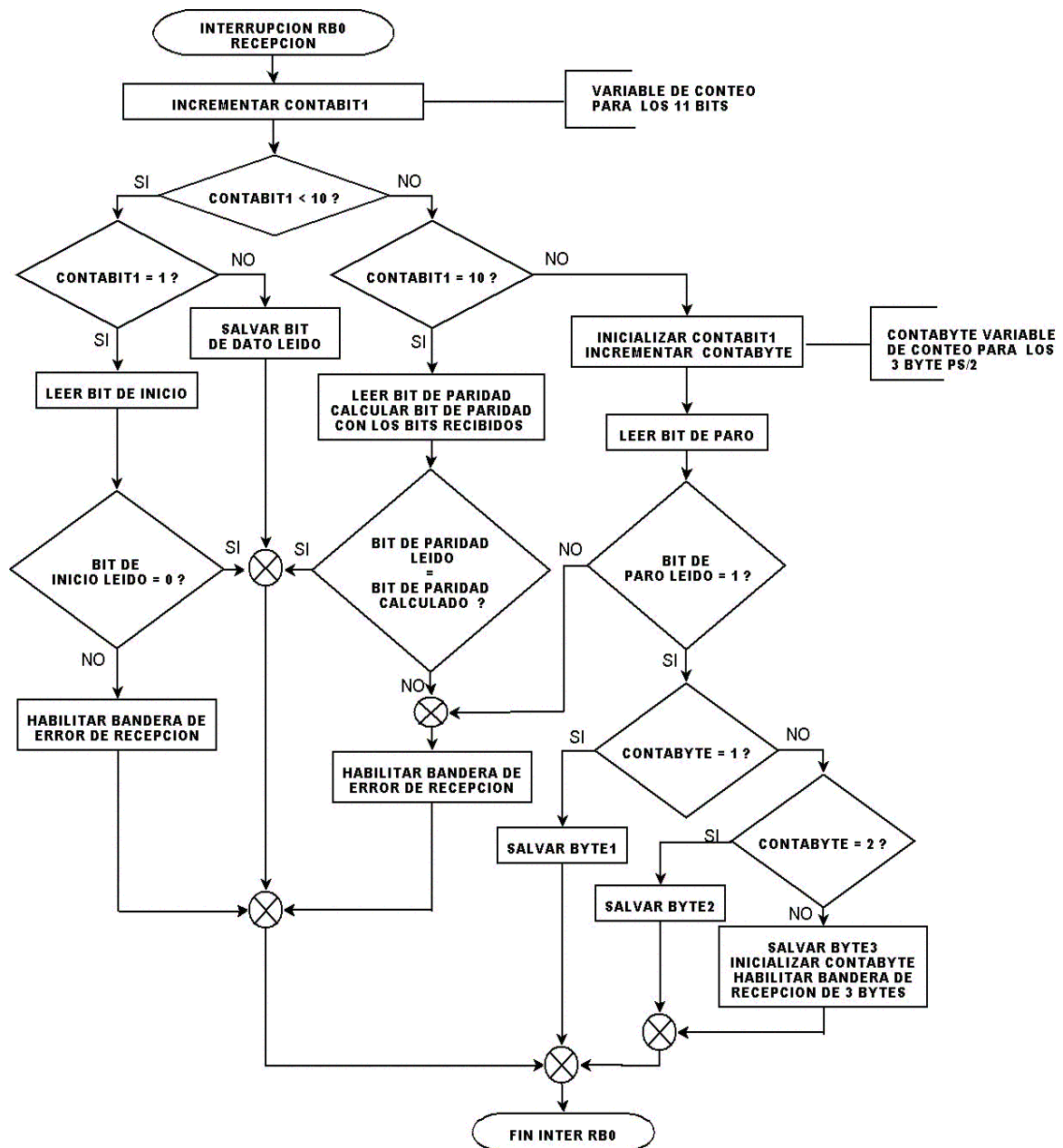


Figura 3.4 Diagrama de flujo para tratamiento de interrupción por flanco en RB0, cuando se va a recibir un byte enviado por el touchpad.

3.1.3 Conteo de estados.

Para establecer la tendencia de las coordenadas de movimiento enviadas por el touchpad se hace un conteo de la ocurrencia de los siguientes estados:

Botón avance presionado, recepción de BYTES PS/2 y ningún BYTE PS/2 recibido.

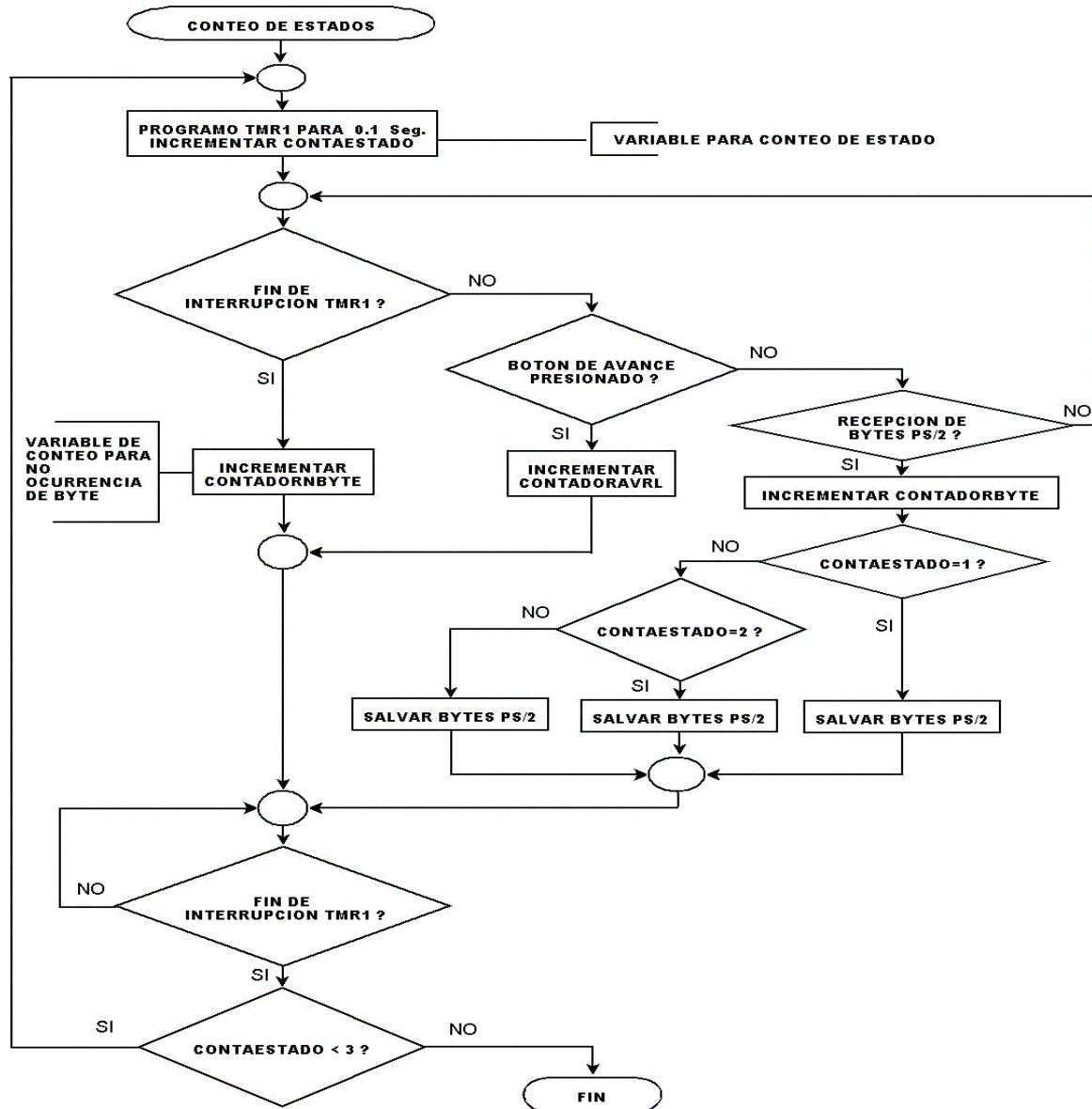


Figura 3. 5 Diagrama de conteo de estados.

En un lapso de 0.1 seg. se establece que estado ocurre y se incrementa el contador apropiado para cada estado.

Si ocurre un byte PS/2, también se salva. Esto se repite tres veces y luego se encuestan los contadores. Esta información será empleada en la determinación de la tendencia.

Solo si la tendencia es correcta, se procede a obtener una cuarta secuencia de coordenadas.

Para que se ejecute un movimiento, debe haber coincidencia en la tendencia de la cuarta secuencia, movimiento anterior y las tres secuencias actuales.

A continuación se muestran las rutinas de avance, retroceso, giro izquierdo y giro derecho.

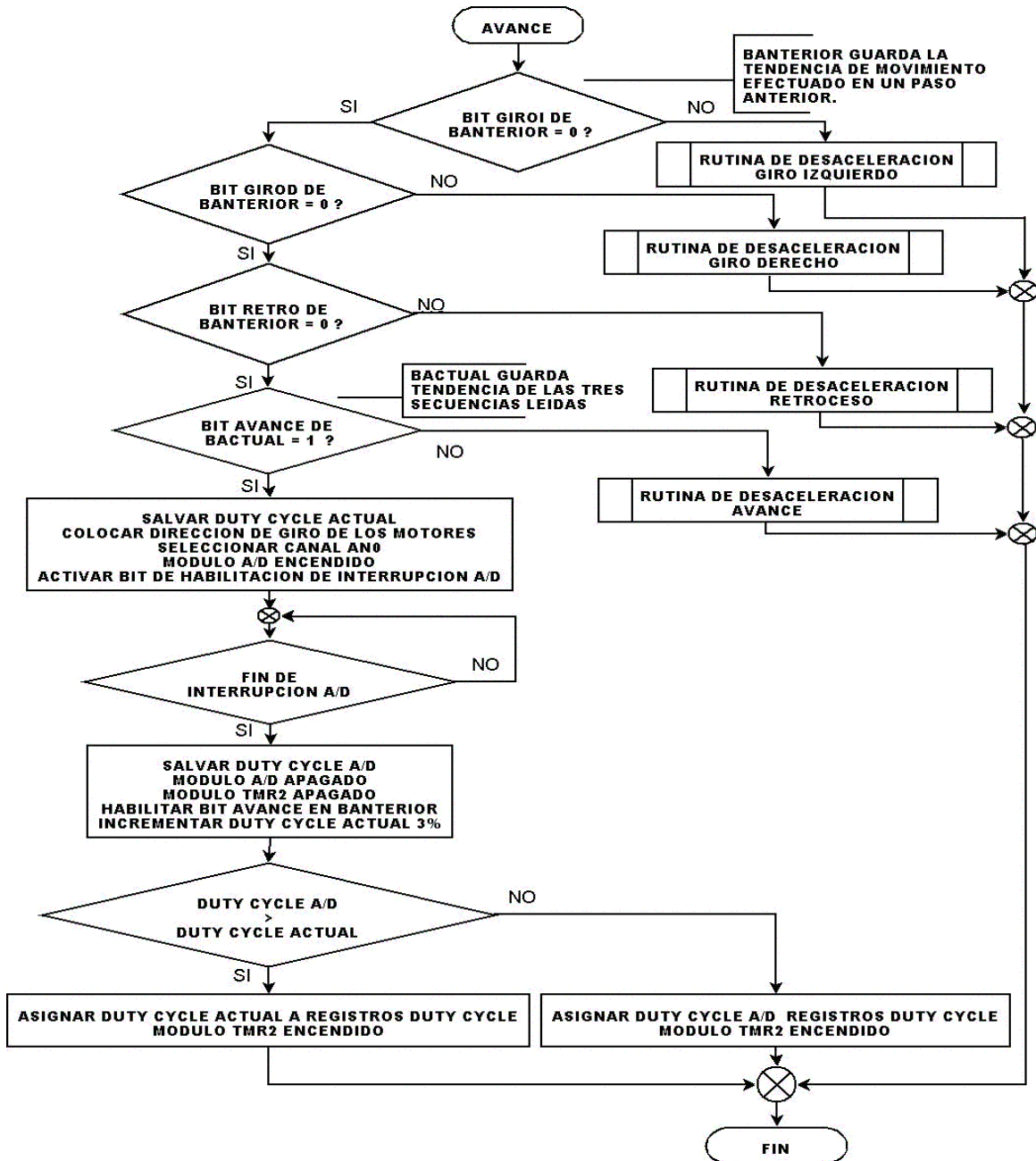


Figura 3.7 Diagrama de flujo para movimiento tipo avance.

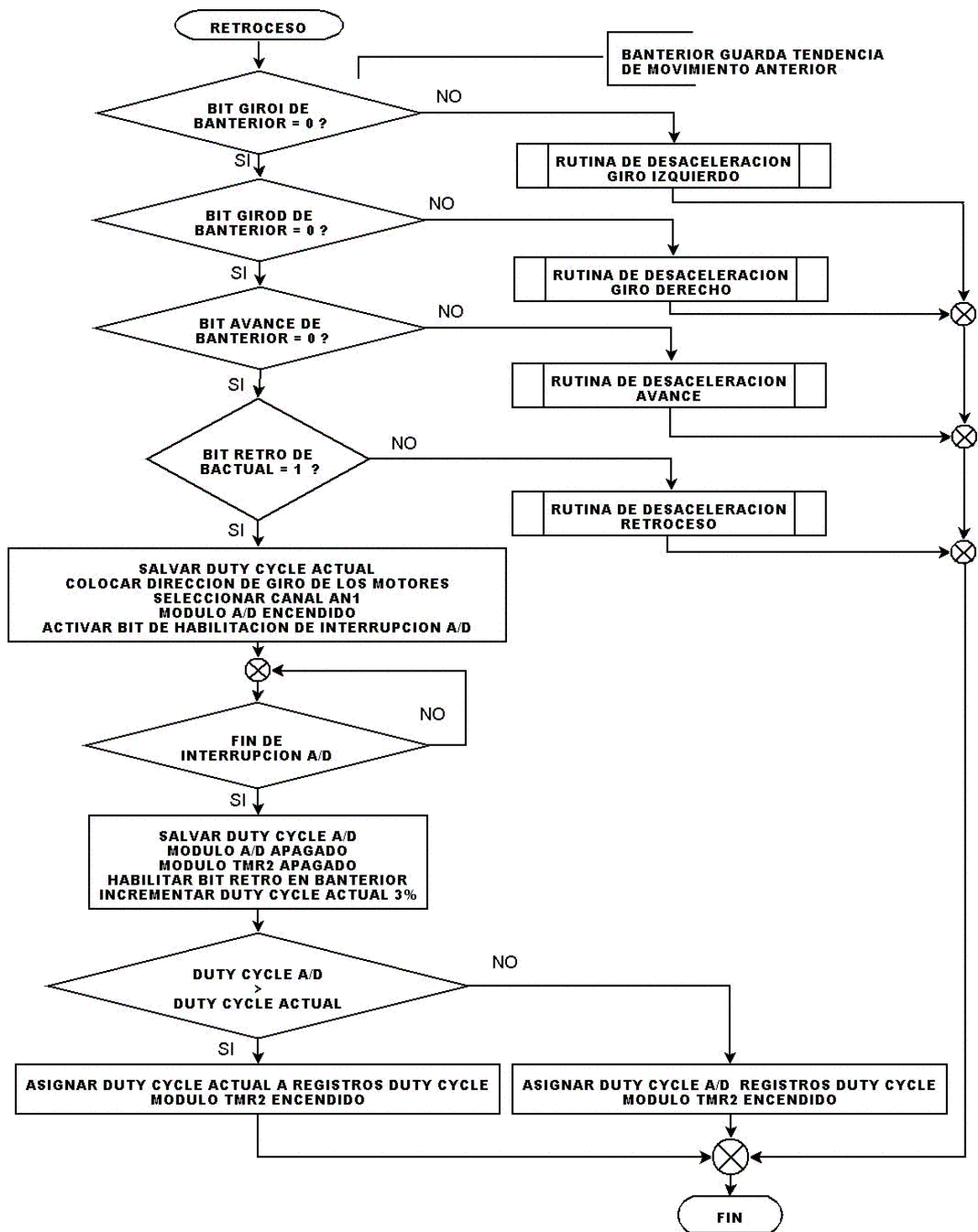


Figura 3.8 Diagrama de flujo para movimiento tipo retroceso.

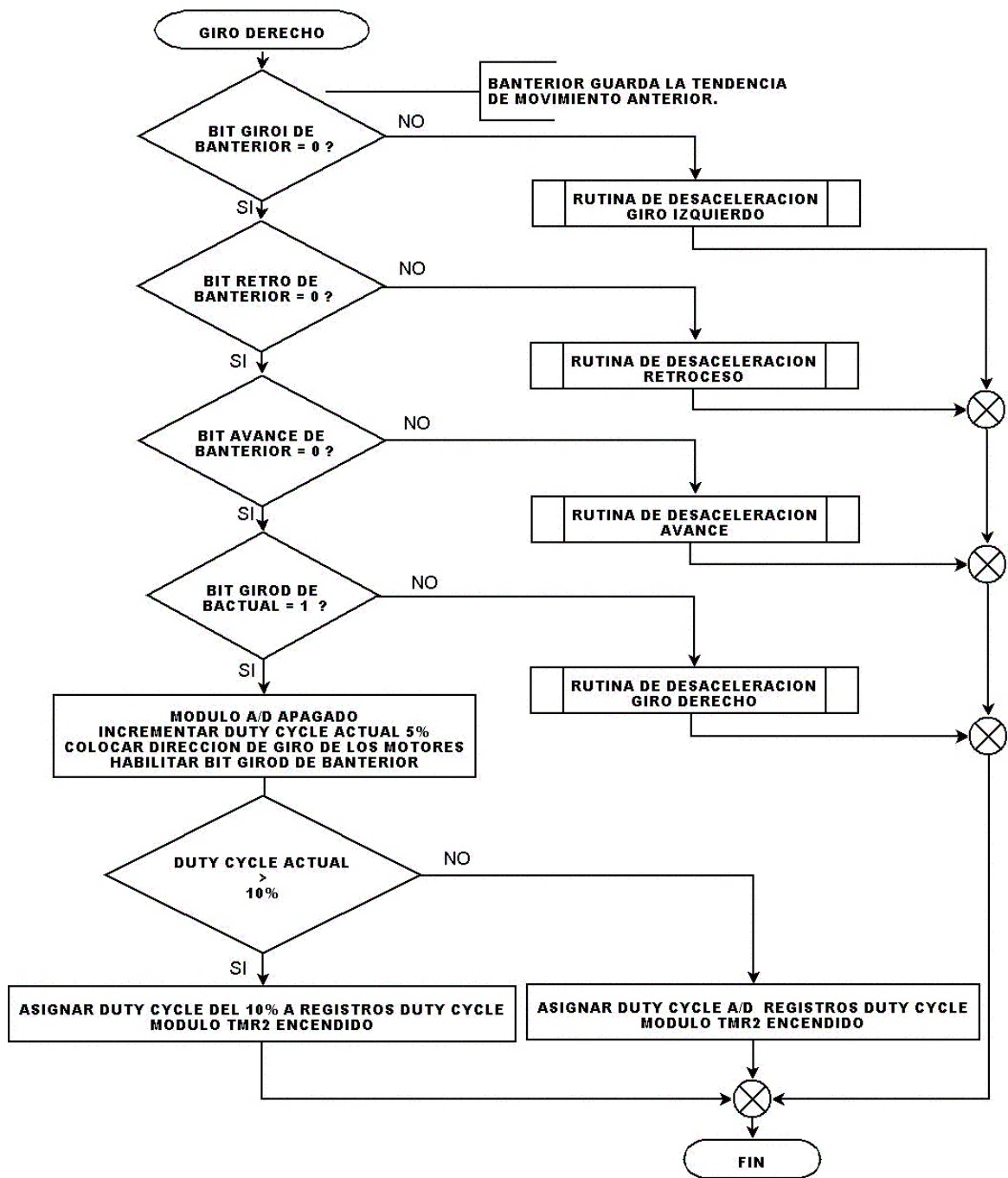


Figura 3.9 Diagrama de flujo para movimiento tipo giro derecho.

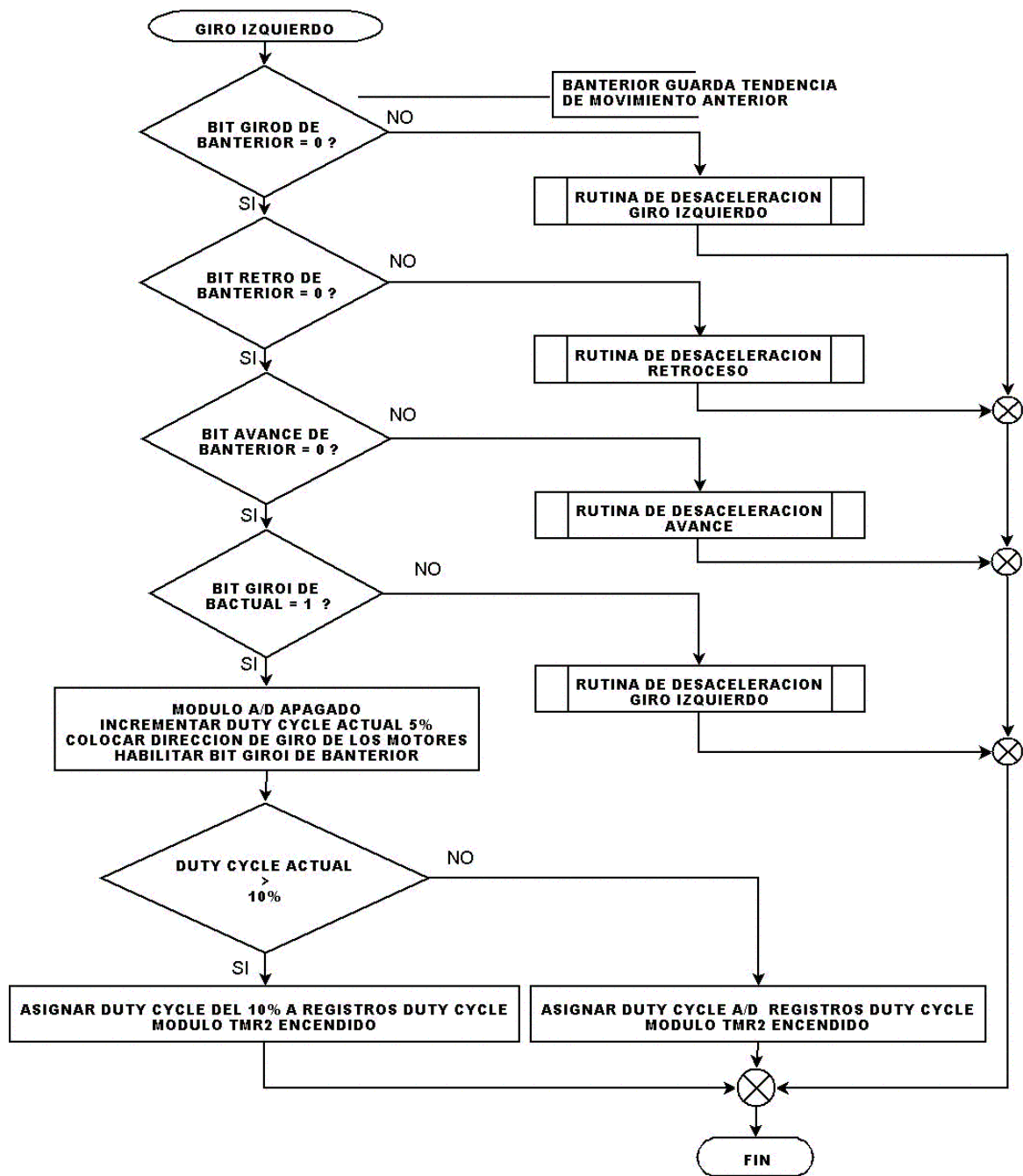


Figura 3.10 Diagrama de flujo para movimiento tipo giro izquierdo.

A continuación se muestra la rutina de servicio a la interrupción cuando ocurre una interrupción por causa del modulo A/D

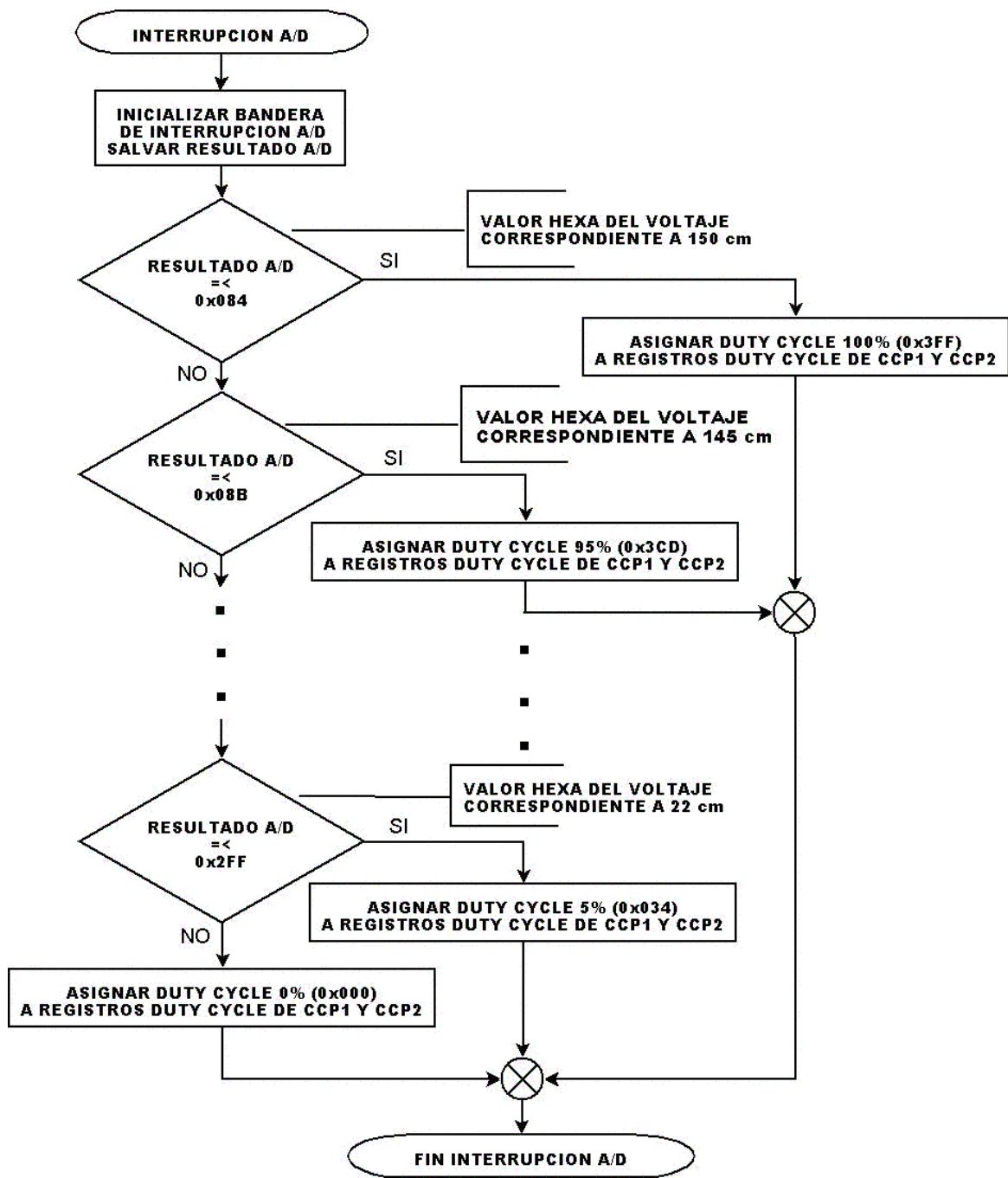


Figura 3.11 Diagrama de flujo para tratamiento de la interrupción A/D.

CONCLUSIONES DEL CAPITULO III.

- Debido a que no existe ningún modulo de comunicación apto en el microcontrolador para manejar la trama de bytes del protocolo PS/2, se tuvo que utilizar el recurso de interrupción por flanco (Interrupción INT).
- Algunos valores de tiempo de retardo utilizados en las rutinas de desaceleración, están definidas en función del prototipo construido, por lo tanto deben ser redimensionadas para futuros desarrollo del proyecto.
- El programa es capaz de manejar la comunicación entre el microcontrolador y el touchpad, lo cual es fundamental para el desarrollo del trabajo realizado.
- El programa cuenta con rutinas de desaceleración que se ejecutan con los siguientes eventos: detección de obstáculos, movimientos bruscos e inactividad del touchpad después de realizar un movimiento.
- El programa escrito en lenguaje ensamblador permite mayor control de las operaciones de los registros internos del microcontrolador.

REFERENCIAS BIBLIOGRAFICAS.

Angulo Usategui Jose Maria. MICROCONTROLADORES PIC DISEÑO PRACTICO DE APLICACIONES.

Segunda Edición. Mc Graw Hill

España 1999.

Manual de Referencia de la Familia de Microcontroladores de Rango Medio PIC micro. USA Microchip [2002].

Recursos encontrados en internet¹²:

www.fiacopetti.it

www.synaptics.com/decaf/utilities/ACF126.pdf

www.panda.cs.ndsu.nodak.edu/~achapwes/

www.gnome.org/projects/dia/home.html

¹² Disponible también en el CD adjunto a este documento.

CONCLUSIONES GENERALES Y RECOMENDACIONES.

- En el capítulo I, se desarrollaron aspectos generales de diseño del controlador de la silla de ruedas, con el fin de definir aspectos de comportamiento y requerimiento de componentes para el desarrollo de las etapas de Hardware y software.
- En el capítulo II, se diseñó e implementó un controlador de mando tipo touchpad. Para lo cual se realizó una exhaustiva selección de componentes y así obtener un diseño óptimo y a la vez económicamente viable.
- En el capítulo III, se realizó un programa en lenguaje ensamblador, que tomando como base los casos de uso desarrollados en el capítulo I, permitió obtener resultados satisfactorios en el desempeño del controlador.
- Se hace imprescindible medir la velocidad de giro de los motores, para que el controlador coloque el duty cycle más adecuadamente.
- Se necesita emplear por lo menos dos sensores medidores de distancia para la parte frontal del prototipo y otros dos para parte trasera, para que el controlador tenga más capacidad de detección de objetos, al cubrir una mayor campo del rango del sensor.

ANEXO A.

PROTOCOLO DE COMUNICACION PS/2

El puerto PS/2 fue desarrollado por IBM para conectar teclados y dispositivos de apuntamiento tales como ratones, “trackballs” y “touchpads”.

En esta sección se describirá la interconexión eléctrica, el protocolo de comunicación y los comandos utilizados para transmitir/recibir información.

A.1.0 INTERCONEXIÓN ELÉCTRICA.

El protocolo PS/2 incluye cuatro señales de las cuales dos son utilizadas para alimentación y las otras dos para comunicación.

VCC: Es una señal simple de +5 Vdc para dar potencia al touchpad.

GND: Señal de tierra común para referencia cero de voltaje.

CLOCK: Señal de reloj generada por el touchpad para sincronizar los datos que son transmitidos o recibidos por el touchpad.

DATA: Es una señal bidireccional en la cual los paquetes de datos son enviados ya sea del microcontrolador al touchpad o viceversa.

Las líneas de Clock y Data son ambas de lógica de colector–abierto por lo que presentan dos estados posibles: baja y alta impedancia. En el estado de baja impedancia la línea (Clock o Data) es llevada a nivel de tierra. En el estado de alta impedancia la línea es dejada sin manejo teniendo un estado flotante alto.

Un puerto PS/2 utiliza un conector mini DIN-6 con la siguiente distribución de pines:

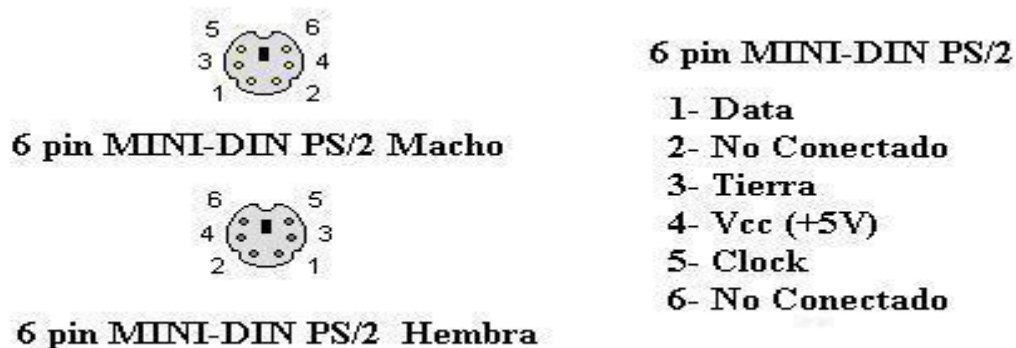


Figura A.1 Distribución de pines del conector PS/2 Macho y Hembra.

A.2.0 COMUNICACIÓN: DESCRIPCIÓN GENERAL.

El PS/2 implementa un protocolo serial síncrono que permite la comunicación en ambas direcciones, aunque no es posible transmitir y recibir datos simultáneamente.

El bus conformado por las señales Clock y Data puede estar en uno de los siguientes estados:

Tabla A.1 Estados del Bus PS/2.

Condición	Estado
Data = alto, Clock = Alto	Ocioso
Data = alto, Clock = Bajo	Comunicación Inhibida
Data = bajo, Clock = Alto	Solicitud de envío

El estado ocioso del bus se refiere a que tanto el touchpad como el microcontrolador no están transmitiendo o recibiendo ningún dato, el bus se encuentra inactivo.

El microcontrolador tiene el control final sobre el bus y puede inhibir la transmisión de datos del touchpad en cualquier momento, para lo cual el microcontrolador coloca la línea de reloj en estado bajo.

Si el microcontrolador está manteniendo la línea de datos en estado bajo y permitiendo a la línea de reloj estar en estado flotante alto, significa que el microcontrolador está próximo a transmitir datos al touchpad.

Los datos transmitidos consisten en una trama de 11-12 bits, estos bits son los siguientes:

Bit de inicio.

Este siempre es cero.

Bits de datos.

8 Bits de datos, el menos significativo debe ser enviado primero.

Bit de paridad.

Paridad impar (El número de unos en los bits de datos más el bit de paridad debe ser impar)

Bit de paro.

Este siempre es uno.

Bit de reconocimiento.

Este es cero. (Este bit ocurre solamente cuando la comunicación es del microcontrolador al touchpad).

A.2.1 COMUNICACIÓN: TRANSMISIÓN DE DATOS DEL TOUCHPAD AL MICROCONTROLADOR.

El touchpad puede transmitir un byte al microcontrolador siempre que el bus este en estado ocioso. El touchpad no puede transmitir si el bus esta en estado de solicitud de envío o inhibido.

Si el bus esta inhibido, el touchpad espera que el bus deje el estado inhibido antes de transmitir. Después que el estado de inhibición finaliza, el touchpad espera aproximadamente 50 μ seg antes de colocar la línea de CLOCK en estado bajo para comenzar el bit de inicio.

Si el bus esta en el estado de solicitud de envío, el touchpad descarta su transmisión pendiente y se dispone a recibir y procesar el byte de argumento o comando enviado por microcontrolador.

El touchpad transmite un byte de datos pulsando la línea de CLOCK a estado bajo y luego a alto un total de 11 veces, mientras transmitiendo el bit de inicio, bits de datos, bit de paridad y bit de paro sobre la línea de datos. El microcontrolador muestrea la línea de datos cada vez que la señal de CLOCK se encuentre en estado bajo. El touchpad cambia el estado de la línea de datos durante el periodo alto de la señal de CLOCK.

Si el microcontrolador inhibe el bus por al menos 100 μ seg durante una transmisión del touchpad, antes del flanco de subida del décimo pulso de reloj la transmisión es abortada y el byte interrumpido se reenviara tan pronto el estado de inhibición finalice.

Si la inhibición ocurre después del décimo pulso de reloj, la transmisión es considerada completa y el microcontrolador debe tomar como valido el byte transmitido por el touchpad.

El estado de inhibición es reconocido por el touchpad notando que la señal de CLOCK se mantiene en estado bajo durante la porción alta del ciclo de reloj.

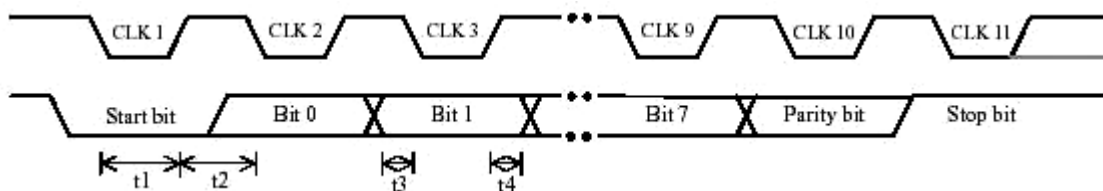


Figura A.2 Forma de onda para la transmisión del touchpad al microcontrolador.

En la figura A.2, el tiempo de duración para los flancos de subida y bajada debe ser al menos 5 μ seg (t_3 , t_4). La señal de reloj es baja o alta durante un tiempo de 30-50 μ seg (t_1 , t_2)

A.2.2 COMUNICACIÓN: TRANSMISIÓN DE DATOS DEL MICROCONTROLADOR AL TOUCHPAD.

Cuando el microcontrolador quiere enviar datos al touchpad hace lo siguiente: Inhibe la comunicación colocando la línea de reloj en estado bajo por lo menos 100 μ seg. Después coloca al bus en estado solicitud de envío, esto lo hace poniendo la línea de datos en estado bajo y liberando el reloj.

Cuando el touchpad detecta el estado de solicitud de envío pulsa la línea de CLOCK a estado bajo y luego a alto un total de 11 veces para recibir un byte. El microcontrolador cambia la línea de datos mientras la señal de reloj es baja y el touchpad muestrea la línea de datos cuando es la señal de reloj es alta. El microcontrolador puede abortar la transmisión en cualquier momento forzando a línea de reloj al estado bajo por al menos 100 μ seg antes del onceavo pulso de reloj.

Después del décimo pulso de reloj, el touchpad chequea para un valido bit de paro y responde poniendo la línea de datos baja (bit de reconocimiento transmitido por el touchpad) y generando un último pulso de reloj. Inmediatamente después de recibir el bit de reconocimiento, el microcontrolador puede inhibir el bus mientras procesa los datos.

Si el touchpad encuentra en estado bajo la línea de datos durante el bit de paro, un error de trama ocurre y genera pulsos de reloj hasta que la línea de datos cambie a estado alto.

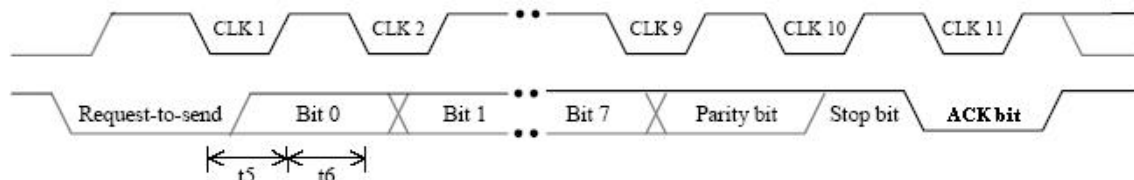


Figura A.3 Forma de onda para la transmisión del microcontrolador al touchpad.

En la figura A.3 El tiempo de duración para los flancos de subida y bajada debe ser al menos 5 μ seg. La señal de reloj es baja o alta durante un tiempo de 30-50 μ seg (t_5 , t_6).

A.3.0 CONJUNTO DE COMANDOS DEL PROTOCOLO PS/2.

El touchpad acepta el conjunto completo de comandos estándar de un ratón PS/2, no obstante en esta sección solamente se describirán los comandos utilizados en el programa realizado.

Comando RESET. (0xFF)

Cuando se envía este comando al touchpad este se reinicia y recalibra, este proceso le toma 300-500 mseg aproximadamente, después de este retardo el touchpad responde con los siguientes tres byte: ACK (0xFA), 0xAA y 0x00. Si el touchpad falla en auto calibrarse, transmite un código de error 0xFC y uno de identificación ID 0x00.

El microcontrolador no debe intentar enviar comandos al touchpad hasta que el proceso de auto calibración haya finalizado.

Cabe mencionar que este proceso de inicialización y calibración también se ejecuta cada vez que se energiza al touchpad, no obstante al enviar el comando RESET se tiene la certeza de que la inicialización y calibración ocurrieron satisfactoriamente al comprobar los tres byte enviados por el touchpad en respuesta a dicho comando.

Después de una inicialización exitosa el touchpad se encuentra con las siguientes características por defecto:

Razón de muestreo reportado: 100 muestras por segundo.

Resolución reportada: 4 cuentas por mm.

Escalamiento: 1:1

Modo STREAM seleccionado.

Reporte de datos deshabilitado

Modo absoluto deshabilitado.

Comando ENABLE. (0XF4)

Este comando se envía para habilitar el reporte de datos. El touchpad responde con un byte de reconocimiento (0xFA)

A.4.0 BYTE MODO DEL TOUCHPAD.

El touchpad cuenta con un pequeño conjunto de características configurables que se encuentran encapsuladas en el *byte modo*, este es un byte de 8 bits al cual el microcontrolador puede colocar un valor utilizando una secuencia de comando especial (ver sección A.4.1).

El byte modo esta dispuesto de la siguiente forma:

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Absoluto	Razón	-	-	Baud/Sleep	DisGest	Packsize	Modo W

Figura A.4 Byte modo del touchpad.

En la figura A.4, los bits marcados con “-“son reservados para uso futuro y deben ser puestos a cero. A continuación se muestra la descripción de estos bits:

Bit 7 Absoluto.

Este bit es 0 para modo Relativo o 1 para seleccionar el modo Absoluto.

Bit 6 Razón.

Este bit es 0 para seleccionar una baja velocidad de paquete de aproximadamente 40 paquetes por segundo o 1 para seleccionar una alta velocidad de paquete de aproximadamente 80 paquetes por segundo

Bit 3 Baud/Sleep.

Para el protocolo serie, este es el bit de Baudio. 0: 1200 bps, 1:9600 bps.

Para PS/2, es el bit de Sleep. 0: Normal operación, 1: Modo Sleep habilitado.

Bit 2 DisGest.

Este bit es 0 para habilitar el procesamiento de los gestos “golpe” y “arrastre” o 1 para deshabilitarlos. En modo absoluto este bit siempre es cero.

Bit 1 PackSize.

Para el protocolo serie, este bit es 0 para seleccionar paquetes absolutos de seis bytes o 1 para seleccionar paquetes de 7-8 bytes (por el bit Modo W)

Para PS/2 este bit es reservado.

Bit 0 Modo W.

En modo absoluto este bit es 1 para seleccionar paquetes que contienen el valor “W” (variable que define el ancho de un dedo) y es 0 para paquetes normales sin el valor de “W”.

En modo relativo, este bit es reservado y debería ser dejado a un valor de cero.

A.4.1 SECUENCIA DE COLOCACION DEL BYTE MODO.

El protocolo PS/2 no permite enviar comandos arbitrarios a un dispositivo de apuntamiento PS/2. A fin de ser compatible con el protocolo PS/2 el touchpad debe expresar toda la información de interrogación (Estado de registros internos o información de las características en las cuales opera el touchpad) y otras operaciones usando solamente combinaciones de aquellos comandos que son soportados por el protocolo PS/2.

Estas combinaciones de comandos son llamadas secuencias de comando especial. Cada secuencia consiste de cuatro comandos Set Resolution (0xE8) con un valor codificado de argumento de 8 bits, seguido inmediatamente por un comando Set Sample Rate (0xF3) o Status Request (0xE9).

Para cambiar el valor del byte modo se envía la siguiente secuencia de comandos:

0XE8 rr 0XE8 ss 0XE8 tt 0XE8 uu 0xF3 0x14

Donde rr, ss, tt y uu son números en el rango 0x00-0x03.

Luego el completo argumento de 8 bits para la secuencia de comando especial es:

$$(rr \times 40) + (ss \times 10) + (tt \times 4) + uu = \text{ARGUMENTO}$$

Ejemplo de cálculo:

Seleccionar modo absoluto, alta velocidad de paquete, no Sleep, Modo W habilitado.

Byte Modo = 1 1 0 0 0 0 1 = 0xC1

1	1	0	0	0	0	0	1	=	0xC1
↓		↓		↓		↓			
0x03		0x00		0x00		0x01			

rr = 0x03, ss = 0x00, tt = 0x00 , uu = 0x01
 $(0x03 \times 40) + (0x00 \times 10) + (0x00 \times 4) + 0x01 = 0xC1$

La secuencia de comando especial para ese caso sería:
0XE8 0x03 0XE8 0x00 0XE8 0x00 0XE8 0x01 0xF3 0x14

ANEXO B

HOJAS DE DATOS TECNICOS DE LOS COMPONENTES UTILIZADOS

B.1.0 Microcontrolador PIC 16F877A.



PIC16F87XA

28/40/44-Pin Enhanced Flash Microcontrollers

Devices Included in this Data Sheet:

- PIC16F873A • PIC16F876A
- PIC16F874A • PIC16F877A

High-Performance RISC CPU:

- Only 35 single-word instructions to learn
- All single-cycle instructions except for program branches, which are two-cycle
- Operating speed: DC – 20 MHz clock input
DC – 200 ns instruction cycle
- Up to 8K x 14 words of Flash Program Memory, Up to 368 x 8 bytes of Data Memory (RAM), Up to 256 x 8 bytes of EEPROM Data Memory
- Pinout compatible to other 28-pin or 40/44-pin PIC16CXXX and PIC16FXXX microcontrollers

Peripheral Features:

- Timer0: 8-bit timer/counter with 8-bit prescaler
- Timer1: 16-bit timer/counter with prescaler, can be incremented during Sleep via external crystal/clock
- Timer2: 8-bit timer/counter with 8-bit period register, prescaler and postscaler
- Two Capture, Compare, PWM modules
 - Capture is 16-bit, max. resolution is 12.5 ns
 - Compare is 16-bit, max. resolution is 200 ns
 - PWM max. resolution is 10-bit
- Synchronous Serial Port (SSP) with SPI™ (Master mode) and I²C™ (Master/Slave)
- Universal Synchronous Asynchronous Receiver Transmitter (USART/SCI) with 9-bit address detection
- Parallel Slave Port (PSP) – 8 bits wide with external RD, WR and CS controls (40/44-pin only)
- Brown-out detection circuitry for Brown-out Reset (BOR)

Analog Features:

- 10-bit, up to 8-channel Analog-to-Digital Converter (A/D)
- Brown-out Reset (BOR)
- Analog Comparator module with:
 - Two analog comparators
 - Programmable on-chip voltage reference (VREF) module
 - Programmable input multiplexing from device inputs and internal voltage reference
 - Comparator outputs are externally accessible

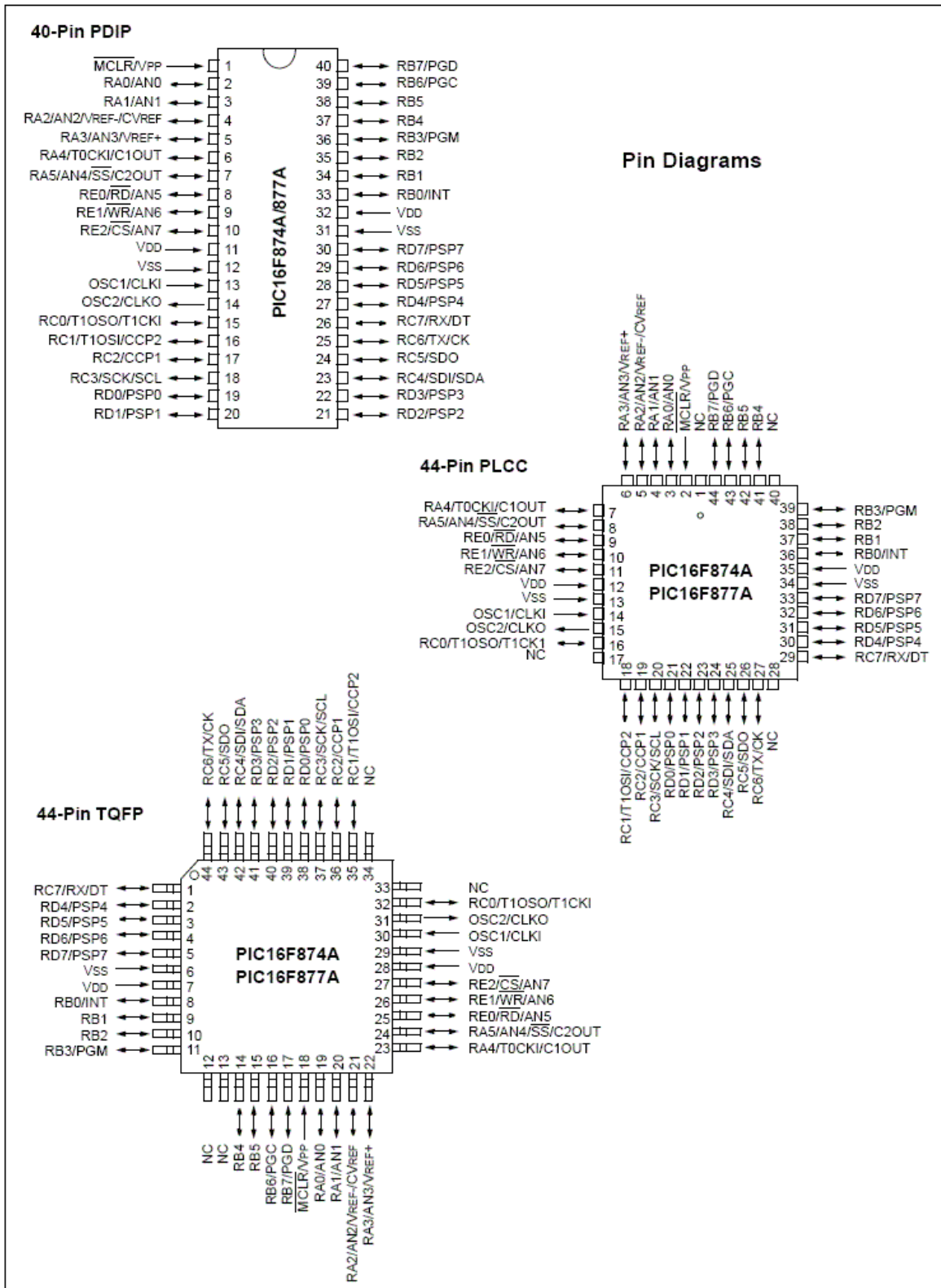
Special Microcontroller Features:

- 100,000 erase/write cycle Enhanced Flash program memory typical
- 1,000,000 erase/write cycle Data EEPROM memory typical
- Data EEPROM Retention > 40 years
- Self-reprogrammable under software control
- In-Circuit Serial Programming™ (ICSP™) via two pins
- Single-supply 5V In-Circuit Serial Programming
- Watchdog Timer (WDT) with its own on-chip RC oscillator for reliable operation
- Programmable code protection
- Power saving Sleep mode
- Selectable oscillator options
- In-Circuit Debug (ICD) via two pins

CMOS Technology:

- Low-power, high-speed Flash/EEPROM technology
- Fully static design
- Wide operating voltage range (2.0V to 5.5V)
- Commercial and Industrial temperature ranges
- Low-power consumption

Device	Program Memory		Data SRAM (Bytes)	EEPROM (Bytes)	I/O	10-bit A/D (ch)	CCP (PWM)	MSSP		USART	Timers 8/16-bit	Comparators
	Bytes	# Single Word Instructions						SPI	Master I ² C			
PIC16F877A	14.3K	8192	368	256	33	8	2	Yes	Yes	Yes	2/1	2



B.2.0 Sensor medidor de distancia.

SHARP

GP2Y0A02YK

GP2Y0A02YK

Long Distance Measuring Sensor

■ Features

1. Less influence on the colors of reflected objects and their reflectivity, due to optical triangle measuring method
2. Distance output type
(Detection range:20 to 150cm)
3. An external control circuit is not necessary
Output can be connected directly to a microcomputer

■ Applications

1. For detection of human body and various types of objects in home appliances, OA equipment, etc

■ Absolute Maximum Ratings (T_a=25°C)

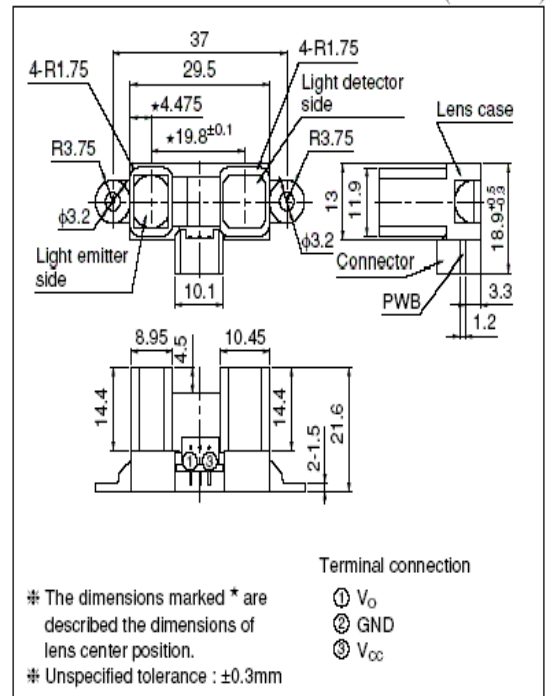
Parameter	Symbol	Rating	Unit
Supply voltage	V _{CC}	-0.3 to +7	V
*1 Output terminal voltage	V _O	-0.3 to V _{CC} +0.3	V
Operating temperature	T _{opr}	-10 to +60	°C
Storage temperature	T _{stg}	-40 to +70	°C

*1 Open collector output

■ Recommended Operating Conditions

Parameter	Symbol	Rating	Unit
Operating Supply voltage	V _{CC}	4.5 to 5.5	V

■ Outline Dimensions (Unit : mm)



Electro-optical Characteristics

($T_a=25^{\circ}\text{C}$, $V_{CC}=5\text{V}$)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Distance measuring range	ΔL	*2 *3	20	-	150	cm
Output terminal voltage	V_O	*2 $L=150\text{cm}$	0.25	0.4	0.55	V
Difference of output voltage	ΔV_O	*2 Output change at $L=150\text{cm}$ to 20cm	1.8	2.05	2.3	V
Average dissipation current	I_{CC}	-	-	33	50	mA

Note) L: Distance to reflective object

*2 Using reflective object: White paper (Made by Kodak Co. Ltd. gray cards R-27 · white face, reflective ratio; 90%)

*3 Distance measuring range of the optical sensor system

Fig.1 Internal Block Diagram

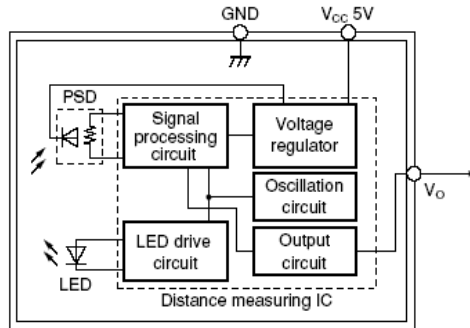


Fig.2 Timing Chart

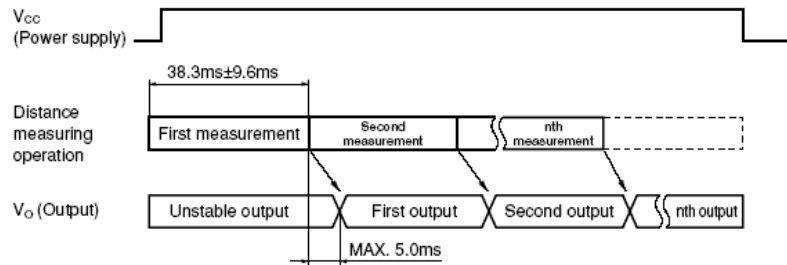
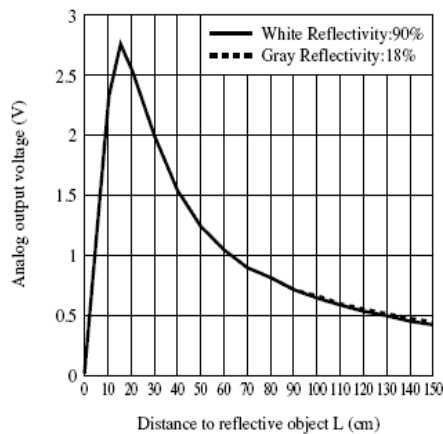


Fig.3 Analog Output Voltage vs. Distance to Reflective Object



B.3.0 Optoacoplador NTE3220



NTE3220 & NTE3221 Optoisolator NPN Transistor Output

Features:

- Current Transfer Ratio: CTR: 50% Min @ $I_F = 5\text{mA}$, $V_{CE} = 5\text{V}$
- High Input-Output Isolation Voltage: $V_{ISO} = 5000V_{rms}$
- Compact DIP Package:
 - NTE3220: 2-Channel Type (8-Lead DIP)
 - NTE3221: 4-Channel Type (16-Lead DIP)

Applications:

- Computer Terminals
- System Appliances, Measuring Instruments
- Registers, Copiers, Automatic Vending Machines
- Electric Home Appliances such as Fan Heaters, Etc.
- Medical Instruments, Physical and Chemical Equipment
- Signal Transmission between Circuits of Different Potentials and Impedances

Absolute Maximum Ratings: ($T_A = +25^\circ\text{C}$ unless otherwise specified)

Input

Forward Current, I_F	50mA
Peak Forward Current (Note 1), I_{FM}	1A
Reverse Voltage, V_R	6V
Power Dissipation, P_D	70mW

Output

Collector-Emitter Voltage, V_{CEO}	35V
Emitter-Collector Voltage, V_{ECO}	6V
Collector Current, I_C	50mA
Collector Power Dissipation, P_C	150mW

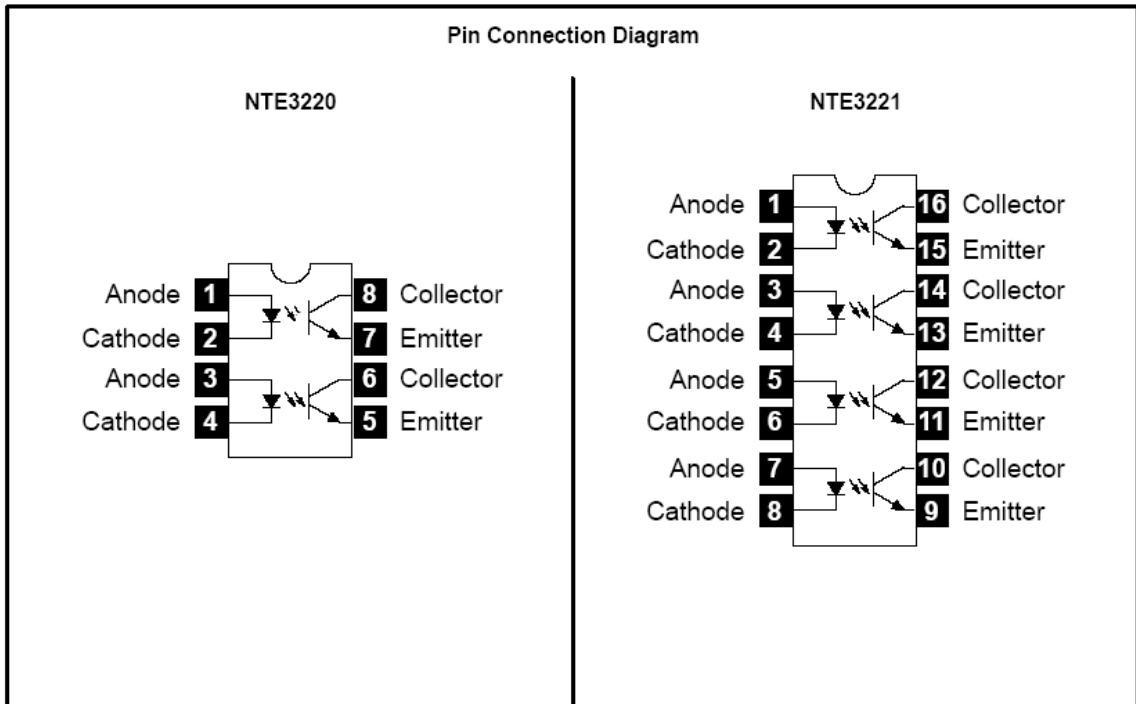
Total Device

Isolation Voltage (Note 2), V_{ISO}	5000 V_{rms}
Total Power Dissipation, P_{tot}	200mW
Operating Temperature Range, T_{opr}	-30° to $+100^\circ\text{C}$
Storage Temperature Range, T_{stg}	-55° to $+125^\circ\text{C}$
Lead Temperature (During Soldering, 10sec), T_L	$+260^\circ\text{C}$

Electro-Optical Characteristics: ($T_A = +25^\circ\text{C}$ unless otherwise specified)

Parameter	Symbol	Test Conditions	Min	Typ	Max	Unit
Input						
Forward Voltage	V_F	$I_F = 20\text{mA}$	–	1.2	1.4	V
Peak Forward Voltage	V_{FM}	$I_{FM} = 500\text{mA}$	–	–	3	V
Reverse Current	I_R	$V_R = 4\text{V}$	–	–	10	μA
Terminal Capacitance	C_t	$V = 0, f = 1\text{kHz}$	–	30	250	pF
Output						
Collector Dark Current	I_{CEO}	$V_{CE} = 20\text{V}, I_F = 0, R_{BE} = \infty$	–	–	100	nA
Collector-Emitter Breakdown Voltage	$V_{(BR)CEO}$	$I_C = 0.1\text{mA}, I_F = 0$	35	–	–	V
Emitter-Collector Breakdown Voltage	$V_{(BR)ECO}$	$I_E = 10\mu\text{A}, I_F = 0$	6	–	–	V
Transfer Characteristics						
Collector Current	I_C	$I_F = 5\text{mA}, V_{CE} = 5\text{V}, \text{Note 3}$	2.5	–	–	mA
Collector-Emitter Saturation Voltage	$V_{CE(sat)}$	$I_F = 20\text{mA}, I_C = 1\text{mA}$	–	0.1	0.2	V
Isolation Resistance	R_{ISO}	500V DC, 40% to 60% R.H.	5×10^{10}	10^{11}	–	Ω
Floating Capacitance	C_f	$V = 0, f = 1\text{MHz}$	–	0.6	1.0	pF
Cut-Off Frequency	f_c	$V_{CE} = 5\text{V}, I_C = 2\text{mA}, R_L = 100\Omega, -3\text{dB}$	–	80	–	kHz
Response Time (Rise)	t_r	$V_{CE} = 2\text{V}, I_C = 2\text{mA}, R_L = 100\Omega$	–	4	18	μs
Response Time (Fall)	t_f		–	3	18	μs

Note 3. $\text{CTR} = \frac{I_C}{I_F} \times 100\%$



ANEXO C

REGISTROS DEL MICROCONTROLADOR UTILIZADOS.

C.1.0 TIMER0

Timer0

Control Register

The OPTION_REG register is a readable and writable register which contains various control bits to configure the TMR0/WDT prescaler, the External INT Interrupt, TMR0, and the weak pull-ups on PORTB.

Note: To achieve a 1:1 prescaler assignment for the TMR0 register, assign the prescaler to the Watchdog Timer.

Register 11-1: OPTION_REG Register

$\overline{\text{RBP}}\text{U}^{(1)}$	INTEDG	T0CS	T0SE	PSA	PS2	PS1	PS0
bit 7							bit 0

- bit 7 **$\overline{\text{RBP}}\text{U}^{(1)}$** : Weak Pull-up Enable bit
 1 = Weak pull-ups are disabled
 0 = Weak pull-ups are enabled by individual port latch values
- bit 6 **INTEDG**: Interrupt Edge Select bit
 1 = Interrupt on rising edge of INT pin
 0 = Interrupt on falling edge of INT pin
- bit 5 **T0CS**: TMR0 Clock Source Select bit
 1 = Transition on T0CKI pin
 0 = Internal instruction cycle clock (CLKOUT)
- bit 4 **T0SE**: TMR0 Source Edge Select bit
 1 = Increment on high-to-low transition on T0CKI pin
 0 = Increment on low-to-high transition on T0CKI pin
- bit 3 **PSA**: Prescaler Assignment bit
 1 = Prescaler is assigned to the WDT
 0 = Prescaler is assigned to the Timer0 module
- bit 2:0 **PS2:PS0**: Prescaler Rate Select bits

Bit Value	TMR0 Rate	WDT Rate
000	1 : 2	1 : 1
001	1 : 4	1 : 2
010	1 : 8	1 : 4
011	1 : 16	1 : 8
100	1 : 32	1 : 16
101	1 : 64	1 : 32
110	1 : 128	1 : 64
111	1 : 256	1 : 128

Legend
 R = Readable bit W = Writable bit
 U = Unimplemented bit, read as '0' - n = Value at POR reset

Note 1: Some devices call this bit $\overline{\text{GPP}}\text{U}$. Devices that have the $\overline{\text{RBP}}\text{U}$ bit, have the weak pull-ups on PORTB, while devices that have the $\overline{\text{GPP}}\text{U}$ have the weak pull-ups on the GPIO Port.

Timer1

Control Register

Register 12-1 shows the Timer1 control register.

Register 12-1: T1CON: Timer1 Control Register

U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	
—	—	T1CKPS1	T1CKPS0	T1OSCEN	$\overline{T1SYNC}$	TMR1CS	TMR1ON	
bit 7								bit 0

bit 7:6 **Unimplemented:** Read as '0'

bit 5:4 **T1CKPS1:T1CKPS0:** Timer1 Input Clock Prescale Select bits

- 11 = 1:8 Prescale value
- 10 = 1:4 Prescale value
- 01 = 1:2 Prescale value
- 00 = 1:1 Prescale value

bit 3 **T1OSCEN:** Timer1 Oscillator Enable bit

- 1 = Oscillator is enabled
- 0 = Oscillator is shut off. The oscillator inverter and feedback resistor are turned off to eliminate power drain

bit 2 **$\overline{T1SYNC}$:** Timer1 External Clock Input Synchronization Select bit

When TMR1CS = 1:

- 1 = Do not synchronize external clock input
- 0 = Synchronize external clock input

When TMR1CS = 0:

This bit is ignored. Timer1 uses the internal clock when TMR1CS = 0.

bit 1 **TMR1CS:** Timer1 Clock Source Select bit

- 1 = External clock from pin T1OSO/T1CKI (on the rising edge)
- 0 = Internal clock (FOSC/4)

bit 0 **TMR1ON:** Timer1 On bit

- 1 = Enables Timer1
- 0 = Stops Timer1

Legend	
R = Readable bit	W = Writable bit
U = Unimplemented bit, read as '0'	- n = Value at POR reset

C.3.0 TIMER2.

Timer2

Control Register

Register 13-1 shows the Timer2 control register.

Register 13-1: 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

Legend	
R = Readable bit	W = Writable bit
U = Unimplemented bit, read as '0'	- n = Value at POR reset

10-bit A/D Converter

Control Register

Register 23-1: ADCON0 Register

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	U-0	R/W-0
ADCS1	ADCS0	CHS2	CHS1	CHS0	GO/DONE	—	ADON
bit 7							bit 0

bit 7:6 **ADCS1:ADCS0:** A/D Conversion Clock Select bits

- 00 = FOSC/2
- 01 = FOSC/8
- 10 = FOSC/32
- 11 = FRC (clock derived from the internal A/D RC oscillator)

bit 5:3 **CHS2:CHS0:** Analog Channel Select bits

- 000 = channel 0, (AN0)
- 001 = channel 1, (AN1)
- 010 = channel 2, (AN2)
- 011 = channel 3, (AN3)
- 100 = channel 4, (AN4)
- 101 = channel 5, (AN5)
- 110 = channel 6, (AN6)
- 111 = channel 7, (AN7)

Note: For devices that do not implement the full 8 A/D channels, the unimplemented selections are reserved. Do not select any unimplemented channel.

bit 2 **GO/DONE:** A/D Conversion Status bit

When ADON = 1

- 1 = A/D conversion in progress (setting this bit starts the A/D conversion which is automatically cleared by hardware when the A/D conversion is complete)
- 0 = A/D conversion not in progress

bit 1 **Unimplemented:** Read as '0'

bit 0 **ADON:** A/D On bit

- 1 = A/D converter module is powered up
- 0 = A/D converter module is shut off and consumes no operating current

Legend

R = Readable bit W = Writable bit
 U = Unimplemented bit, read as '0' - n = Value at POR reset

ADCON1 Register

U-0	U-0	R/W-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	ADFM	—	PCFG3	PCFG2	PCFG1	PCFG0
bit 7				bit 0			

bit 7:6 **Unimplemented: Read as '0'**

bit 5 **ADFM: A/D Result format select** (also see [Figure 23-6](#)).

1 = Right justified. 6 Most Significant bits of ADRESH are read as '0'.

0 = Left justified. 6 Least Significant bits of ADRESL are read as '0'.

bit 4 **Unimplemented: Read as '0'**

bit 3:0 **PCFG3:PCFG0: A/D Port Configuration Control bits**

PCFG	AN7	AN6	AN5	AN4	AN3	AN2	AN1	AN0	VREF+	VREF-	C / R
0000	A	A	A	A	A	A	A	A	AVDD	AVSS	8 / 0
0001	A	A	A	A	VREF+	A	A	A	AN3	AVSS	7 / 1
0010	D	D	D	A	A	A	A	A	AVDD	AVSS	5 / 0
0011	D	D	D	A	VREF+	A	A	A	AN3	AVSS	4 / 1
0100	D	D	D	D	A	D	A	A	AVDD	AVSS	3 / 0
0101	D	D	D	D	VREF+	D	A	A	AN3	AVSS	2 / 1
011x	D	D	D	D	D	D	D	D	—	—	0 / 0
1000	A	A	A	A	VREF+	VREF-	A	A	AN3	AN2	6 / 2
1001	D	D	A	A	A	A	A	A	AVDD	AVSS	6 / 0
1010	D	D	A	A	VREF+	A	A	A	AN3	AVSS	5 / 1
1011	D	D	A	A	VREF+	VREF-	A	A	AN3	AN2	4 / 2
1100	D	D	D	A	VREF+	VREF-	A	A	AN3	AN2	3 / 2
1101	D	D	D	D	VREF+	VREF-	A	A	AN3	AN2	2 / 2
1110	D	D	D	D	D	D	A	A	AVDD	AVSS	1 / 0
1111	D	D	D	D	VREF+	VREF-	D	A	AN3	AN2	1 / 2

A = Analog input D = Digital I/O

C/R = # of analog input channels / # of A/D voltage references

<p>Legend</p> <p>R = Readable bit W = Writable bit</p> <p>U = Unimplemented bit, read as '0' - n = Value at POR reset</p>

Note 1: On any device reset, the port pins that are multiplexed with analog functions (ANx) are forced to be an analog input.

Control Register

Register 14-1: CCPxCON Register

U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	
—	—	DCxB1	DCxB0	CCPxM3	CCPxM2	CCPxM1	CCPxM0	
bit 7								bit 0

bit 7:6 **Unimplemented:** Read as '0'

bit 5:4 **DCxB1:DCxB0:** PWM Duty Cycle bit1 and bit0

Capture Mode:
Unused

Compare Mode:
Unused

PWM Mode:

These bits are the two LSBs (bit1 and bit0) of the 10-bit PWM duty cycle. The upper eight bits (DCx9:DCx2) of the duty cycle are found in CCPxL.

bit 3:0 **CCPxM3:CCPxM0:** CCPx Mode Select bits

0000 = Capture/Compare/PWM off (resets CCPx module)

0100 = Capture mode, every falling edge

0101 = Capture mode, every rising edge

0110 = Capture mode, every 4th rising edge

0111 = Capture mode, every 16th rising edge

1000 = Compare mode,

Initialize CCP pin Low, on compare match force CCP pin High (CCPIF bit is set)

1001 = Compare mode,

Initialize CCP pin High, on compare match force CCP pin Low (CCPIF bit is set)

1010 = Compare mode,

Generate software interrupt on compare match

(CCPIF bit is set, CCP pin is unaffected)

1011 = Compare mode,

Trigger special event (CCPIF bit is set)

11xx = PWM mode

Legend

R = Readable bit

W = Writable bit

U = Unimplemented bit, read as '0'

- n = Value at POR reset