

# **ESCUELA POLITÉCNICA NACIONAL**

## **ESCUELA DE FORMACIÓN DE TECNÓLOGOS**

### **DISEÑO Y CONSTRUCCIÓN DE UN SISTEMA DE AUTOMATIZACIÓN DE APERTURA Y CIERRE DE PERSIANAS VERTICALES**

**PROYECTO PREVIO A LA OBTENCIÓN DEL TÍTULO DE TECNÓLOGO EN  
ELECTRÓNICA Y TELECOMUNICACIONES**

**GALARZA JÁCOME ANGEL DANIEL(danyg\_larza@hotmail.es)**

**MEDINA LÒPEZ SOFÍA KATHERINE (sofikathe\_conmigo@yahoo.com)**

**DIRECTOR: ING. CARLOS FLORES(carlos.flores@epn.edu.ec)**

**QUITO, OCTUBRE 2008**

## DECLARACIÓN

Nosotros, Galarza Jácome Ángel Daniel, Medina López Sofía Katherine, declaramos bajo juramento que el trabajo aquí descrito es de nuestra autoría; que no ha sido previamente presentado para ningún grado o calificación profesional, y que hemos consultado las referencias bibliográficas que se incluyen en este documento.

A través de la presente declaración cedemos nuestros derechos de propiedad intelectual correspondiente a este trabajo, a la Escuela Politécnica Nacional, según lo establecido por la Ley de Propiedad Intelectual, por su Reglamento y por la normativa institucional vigente.

-----  
Daniel Galarza

-----  
Sofía Medina

## **CERTIFICACIÓN**

Certifico que el presente trabajo fue desarrollado por los alumnos: Medina López Sofía Katherine, Galarza Jácome Ángel Daniel, bajo mi supervisión.

-----

Ing. Carlos Flores  
DIRECTOR DEL PROYECTO

## **AGRADECIMIENTO**

Con toda sinceridad queremos agradecer por la ayuda desinteresada en el desarrollo y consecución del presente proyecto.

A Dios por bendecir nuestros pasos en la vida que hemos dado junto a nuestros padres y hermanos, ya que sin la ayuda y apoyo incondicional que nos han dado siempre no habiéramos podido alcanzar un sueño y una etapa más en nuestras vidas.

Nuestra gratitud al Ing. Carlos Flores por la acertada dirección y por el apoyo brindado en el desarrollo de este proyecto.

Muchas gracias a todos, los llevamos en el corazón.

SOFY Y DANIEL.

## **DEDICATORIA**

Este trabajo lo he dedicado con la más extensa gratitud a nuestros padres, por todo su amor y por el gran esfuerzo que han invertido en provecho de nuestra preparación personal y profesional.

Habiendo cumplido con una de nuestras metas, compartimos esta inmensa alegría con nuestros hermanos, amigos y con el señor Jesucristo, la fuente y el gozo de nuestras vidas.

Para todos ustedes nuestro esfuerzo y trabajo plasmado en este proyecto.

**SOFY Y DANIEL.**

## CONTENIDO

CONTENIDO.....	1
INTRODUCCIÓN.....	5
RESUMEN.....	7
CAPÍTULO I: FUNDAMNETOS TEÓRICOS.....	9
1.1. EL CLIMA EN EL ECUADOR.....	9
1.1.1. Generalidades.....	9
1.2. CARACTERÍSTICAS DEL MICROCONTROLADOR.....	13
1.2.1. ¿Qué es el microcontrolador?.....	13
1.2.2. ¿Qué es el microprocesador?.....	14
1.2.3. Diferencias entre microprocesador y microcontrolador.....	15
1.2.4. Los recursos comunes de los microcontroladores .....	15
1.2.4.1. Arquitectura Básica.....	16
1.2.4.2. El procesador o CPU.....	17
1.2.4.3. Procesaor RISC con arquitectura Harvard.....	18
1.2.5. Memoria.....	19
1.2.5.1. Memoria de programa.....	19
1.2.5.2. Memoria de datos.....	19
1.2.5.3. Líneas de Entrda /Salida .....	19
1.2.5.4. Reloj principal.....	20
1.2.6. Recursos especiales.....	21
1.2.6.1. Temporizadores o "Timers".....	22
1.2.6.2. Perro guardián o "Wacthdog" (WDT).....	22
1.2.6.3. Protección ante fallo de alimentación o "Brownout" .....	22
1.2.6.4. Estado de reposos o de bajo consumo.....	23
1.2.6.5. Conversor A/D (CAD).....	23
1.2.6.6. Conversor D/A (CDA).....	24
1.2.6.7. Comparador analógico.....	24
1.2.6.8. Modulador de anchura de impulsos o PWM.....	24
1.2.6.9. Puertos de Comunicación.....	24
1.2.7. Microcontrolador 16F877A.....	25
1.2.7.1. Estructura y características.....	25

1.2.7.2. La Familia de los PIC.....	25
1.2.7.2.1. Gama Enana .....	26
1.2.7.2.2. Gama Baja o Básica.....	26
1.2.7.2.3. Gama Media.....	27
1.2.7.2.4. Gama Alta.....	27
1.2.7.3. Como recursos fundamentales.....	27
1.2.7.4. Como dispositivos Periféricos.....	28
1.2.7.5. Aspecto Externo .....	29
1.2.7.6. Organización de la memoria.....	30
1.2.7.7. Puertos de Entrada/Salida.....	32
1.2.7.8. El reloj .....	35
1.2.7.9. Reinicialización o reset.....	36
1.2.8.0. Interrupciones.....	37
1.2.8.0.1. Causas de la interrupción.....	37
1.3. MOTOR PAP O PASO A PASO.....	38
1.3.1. Introducción.....	38
1.3.1.1. Definición.....	39
1.3.2. Parámetros del motor paso a paso.....	39
1.3.2.1. Características mecánicas.....	39
1.3.2.2. Características físicas de un motor paso a paso.....	42
1.3.3. Tipos de motores paso a paso.....	42
1.3.3.1. Los de imán permanente .....	43
1.3.3.2. Los motores del tipo de reluctancia variable.....	43
1.3.3.3. Los motores híbridos .....	44
1.3.4. Motores paso a paso de imán permanente .....	44
1.3.4.1. Principio de funcionamiento.....	44
1.3.4.2. Clasificación de los motores de imán permanente.....	46
1.3.4.2.1. Motores paso a paso unipolares.....	46
1.3.4.2.2. Motores paso a paso bipolares.....	51
1.3.5. Identificación de las bobinas de un motor paso a paso bipolar.....	52
1.4. PANTALLA DE CRISTAL Líquido (LCD).....	53
1.4.1. Introducción.....	53
1.4.2. Descripción del LCD 2X16.....	54

1.4.3. Descripción de pines.....	55
1.4.4. Características principales .....	56
1.4.5. Funcionamiento .....	56
1.4.6. Declaración de instrucciones para el LCD en PBP.....	57
1.5. EL AMPLIFICADOR OPERACIONAL (AOP) .....	57
1.5.1. Introducción.....	57
1.5.1.1. Definición.....	58
1.5.2. El modelo ideal de un AOP .....	58
1.5.3. Simbología del AOP.....	59
1.5.4. Circuitos lineales básicos con AOP.....	59
1.5.4.1. Amplificador no inversor.....	59
1.5.4.2. Circuito seguidor.....	60
1.5.5. Circuitos no lineales básicos con AOP.....	61
1.5.5.1. Comparadores.....	61
1.5.6. Circuito Integrado LM358.....	63
1.6. SENSORES.....	63
1.6.1. Light Dependent Resistor (LDR).....	64
1.6.2. Sensor de Temperatura LM35.....	66
1.7. OPTO - ACOPLADOR .....	68
1.7.1. Funcionamiento del Opto - acoplador .....	68
CAPÍTULO II: DISEÑO Y CONSTRUCCIÓN DEL SISTEMA DE PAUTOMATIZACIÓN DE APERTURA Y CIERRE DE PERSIANAS VERTICALES.....	
2.1. INTRODUCCIÓN.....	70
2.2. CONSIDERACIONES BÁSICAS A LA HORA DE MONTAR UN PROYECTO.....	70
2.3. DESCRIPCIÓN DEL HARDWARE UTILIZADO.....	72
2.3.1. Circuito de alimentación.....	73
2.3.2. Conexión del microcontrolador.....	74
2.3.3. Termómetro digital.....	75
2.3.4. Teclas e Interruptores.....	76
2.3.5. Funcionamiento del circuito sensor de iluminación.....	78
2.3.6. Motor paso a paso unipolar de 5 hilos.....	79



2.3.7. Pantalla de cristal Líquido (LCD).....	80
2.3.8. Opto - acopladores.....	81
2.4. DESARROLLO DELSOFTWARE.....	82
2.4.1. Lenguaje de programación.....	83
2.4.1.1. Lenguaje de bajo nivel.....	83
2.4.1.2. Lenguaje de alto nivel.....	83
2.5. LENGUAJE DE PROGRAMACIÓN PBP.....	84
2.6. DESCRIPCIÓN DEL PROGRAMA DISEÑADO Y REALIZADO EN MICRO CODE STUDIO.....	91
2.7. FUNCIONAMIENTO DEL SISTEMA DE AUTOMATIZACIÓN DE APERTURA Y CIERRE DE PERSIANAS VERTICALES.....	106
2.8. DISEÑO DE LAS TARJETAS DEL SISTEMA DE AUTOMATIZACIÓN.....	110
CAPÍTULO III: CONSTRUCCIÓN Y ACOPLAMIENTO DEL SISTEMA DE AUTOMATIZACIÓN DE APERTURA Y CIERRE DE PERSIANAS VERTICALES.....	114
3.1. CONSTRUCCIÓN Y MODIFICACIONES.....	114
3.2. ELABORACIÓN DE LOS CIRCUITOS IMPRESOS.....	116
3.2.1. Acabado final de las placas electrónicas .....	119
3.2.2. Elaboración de las cajas donde serán colocadas las placas.....	120
3.3. EMPOTRAMIENTO DE LAS CAJAS DE LOS MOTORES PASO A PASO.....	122
3.4. CABLEADO DEL SISTEMA DE AUTOMATIZACIÓN DE APERTURA Y CIERRE DE PERSIANAS VERTICALES.....	122
3.5. ACABADO FINAL DEL SISTEMA DE APERTURA Y CIERRE DE PERSIANAS VERTICALES.....	123
3.6. PRUEBAS Y RESULTADOS.....	124
CAPÍTULO IV: CONCLUSIONES Y RECOMENDACIONES .....	126
4.1. CONCLUSIONES .....	126
4.2. RECOMENDACIONES.....	127
BIBLIOGRAFÍA .....	130
ANEXOS.....	132

## INTRODUCCIÓN

El presente documento trata sobre el diseño y la construcción del sistema de automatización de apertura y cierre de persianas verticales para que trabaje bajo las siguientes condiciones del medio:

Temperatura ambiental, condiciones de claridad /oscuridad, y que trabaje de forma manual con la ayuda de un panel de control.

Empleando al microcontrolador PIC16F877A el mismo que se ha programado con el lenguaje PBP (Pic Basic Pro.), se utilizó este microcontrolador porque resultó ser el más apropiado para el desarrollo de este proyecto y además se lo puede conseguir localmente.

Para la construcción del proyecto se emplea un sensor de temperatura LM35 para poder desarrollar el termómetro digital, este dispositivo presenta en su pin out una variación de 10mV por grado centígrado su alimentación puede ser de 4 a 30 Voltios, y su rango de temperatura a sensor entre -55 °C hasta 150 °C.

Su funcionamiento es muy simple, el voltaje del pin out del LM35, lo duplicamos a través de un juego de amplificadores operacionales (LM358), el cuál esta configurado como amplificador no inversor, la salida del Amplificador Operacional lo conectamos al Puerto A0 del PIC, el mismo que está configurado como conversor A/D de 8 bits, este dato se almacena en una variable, la cual dividimos para dos, con la finalidad de estabilizar la señal que ingresa del pin out LM35.

Si la temperatura actual permanece dentro del rango establecido, por ejemplo si la temperatura mínima es 16°C, la temperatura actual es 22°C y la temperatura máxima 26°C, los motores de paso, no se activarán ya sea para cerrar o abrir la persiana, además de colocarle en forma lateral o frontal independientemente del estado que esta haya estado anteriormente.

Si la temperatura actual es menor a la temperatura mínima, los motores paso a paso realizarán los giros correspondientes para cerrar la persiana vertical, y si la temperatura actual es mayor a la temperatura máxima se abrirá.

Cuando la temperatura actual está dentro del rango, los motores de paso no se accionarán, para esto se ha implementado un teclado muy fácil de utilizar, donde se presiona las teclas varias veces ya sea abrir, cerrar, frontal y lateral para abrir o cerrar las persianas verticales, según la necesidad y también para poder operar manualmente.

Si se desea modificar los rangos de temperatura, se presiona la tecla E con las otras dos teclas, bajar y subir se aumenta o disminuye la temperatura mínima a comparar, y una vez que se esté de acuerdo con la temperatura elegida se presiona la tecla E nuevamente, luego se pide programar la temperatura máxima, se procede igual que en el caso anterior y cuando se presione la tecla E, parpadeará cuatro veces el led, indicando que los nuevos valores ya fueron guardados en la memoria no volátil.

Los datos de la temperatura mínima, actual y máxima a visualizarse se la hacen por medio de un LCD 2 x 16, el mismo que se conecta al PIC 16F877A con un bus de 4 bits, esta comunicación se realiza enviando primero los 4 bits más altos y luego los 4 bits más bajos.

En el riel se colocó un opto-acoplador, al inicio y uno al final, su función es enviar una señal al PIC cuando la persiana vertical llegue a cerrarse o abrirse completamente, y de esta manera el motor paso a paso no continúe girando y se detenga.

Por medio de un interruptor se cambia la modalidad para que trabaje la LDR, según las condiciones de claridad /oscuridad (apertura o cierre de la persianas), trabajando con las mismas funciones expuestas anteriormente

## RESUMEN

Actualmente, en hogares, lugares de trabajo, etc. existen persianas verticales y horizontales, pero nosotros nos enfocaremos en el de tipo vertical, a la misma que realizamos modificaciones para el desarrollo de este proyecto, con la finalidad de brindarle comodidad al usuario en las tareas cotidianas que realiza diariamente.

El país no es inmune al problema del robo así que se la puede emplear como un simulador de presencia si el usuario no se encuentra en el lugar donde está instalado el proyecto.

El presente documento está dividido en cuatro capítulos logrando una organización adecuada de la información.

En el capítulo I: Se especifica acerca de las generalidades del clima de Quito, así también se describe las principales ventajas del microcontrolador usado, como sus aspectos técnicos que deben ser tomados en cuenta para la programación del mismo y sus características externas e internas.

Igualmente se define los diferentes dispositivos electrónicos que hemos empleado como amplificadores operacionales, sensores, LCD, etc.

En el capítulo II: Se describe las funciones de los dispositivos que estarán en el proyecto, las etapas que conforman el mismo con sus respectivos diagramas, se explica detalladamente en el programa del microcontrolador que hace cada una de las instrucciones requeridas para el buen funcionamiento del proyecto. Así como el funcionamiento respectivo de Hardware y el Software empleado para el desarrollo de este proyecto.

En el capítulo III: Se describe los pasos utilizados para el diseño y construcción del circuito impreso, además se muestran las pruebas realizadas al sistema de

automatización y sus resultados, de la misma forma las correcciones y modificaciones que se realizaron para su perfecto funcionamiento.

En el capítulo IV: Se indican las conclusiones más importantes del presente proyecto que se han obtenido durante el proceso de elaboración y las recomendaciones necesarias para que funcione adecuadamente.

Los anexos contienen información para el usuario.

# CAPÍTULO I: FUNDAMENTOS TEÓRICOS

## 1.1. EL CLIMA EN EL ECUADOR <sup>1</sup>

### 1.1.1. Generalidades:

El territorio del Ecuador está dividido en cuatro regiones naturales claramente definidas entre sí, ya sea por su topografía, clima, vegetación y población. Estas cuatro regiones son: costa, sierra, oriente e insular.

A 600 millas de la costa ecuatoriana hacia el oeste se encuentra el Archipiélago de Colón o Islas Galápagos, como su nombre lo indica está integrado por varias islas siendo la mayor de ellas La Isabela.

El suelo de la región litoral es generalmente bajo, con pequeñas elevaciones que no sobrepasan los 800 m de altura sobre el nivel del mar.

La región sierra está atravesada por la cordillera de los Andes que la recorre de norte a sur. En la sierra las altitudes varían desde los 1200m hasta los 6000m.

En la región oriental las condiciones naturales son semejantes a todas las regiones tropicales del mundo. Debido a su posición geográfica y a la diversidad de alturas impuesta por la cordillera de los Andes, el Ecuador presenta una gran variedad de climas y cambios considerables a cortas distancias.

El país está ubicado dentro del cinturón de bajas presiones atmosféricas donde se sitúa la Zona de Convergencia Intertropical (ZCIT), por esta razón, ciertas áreas del Ecuador reciben la influencia alternativa de masas de aire con diferentes características de temperatura y humedad.

---

<sup>1</sup> www. [http://www.ecuaworld.ec/clima\\_ecuador.htm](http://www.ecuaworld.ec/clima_ecuador.htm).  
<http://www.vivecuador.com/html2/esp/clima.htm>.

Se cuentan con climas tropicales y templados, regiones con características subtropicales, situadas principalmente en las estribaciones de las dos cordilleras; también encontramos zonas desérticas, semi-desérticas, estepas frías y cálidas, etc.

El Ecuador por su situación astronómica en el centro de la Zona Tórrida debiera tener un clima completamente cálido de manera general. No obstante, no es así siempre ni en todos los lugares, debido a la influencia de otros factores que modifican el clima.

La altitud del suelo es sin duda, el factor que más contribuye a modificar el clima en el país. Si se considera que partiendo del nivel del mar la temperatura desciende un grado por cada 200 metros de altura, nuestro clima tiene una fluctuación de aproximadamente 31°C, ya que el nivel de sus tierras va desde 0 metros al nivel del mar hasta 6310 metros que es su máxima altura en las cumbres del Chimborazo.

Esto ha hecho que el país goce del privilegio de poseer todos los tipos de clima, desde el cálido del litoral hasta el glacial de las alturas andinas.

Donde existe mayor vegetación, como en el litoral y el oriente, se produce mayor evaporación del suelo y de las plantas lo que contribuye al aumento de las precipitaciones, modificando así el clima en dichas regiones.

La altura de las cordilleras occidental y oriental del sistema montañoso de los Andes impide la penetración de los vientos cálidos y húmedos del occidente y del oriente al interior de las hoyas de nuestra región Andina, modificando el clima de esta región.

La insolación se entiende por, el número de horas en que el sol se hace presente en un lugar determinado. En la región interandina, la insolación fluctúa entre las 1200 y 2000 horas anuales con ciertas excepciones de lugares muy lluviosos.

La temperatura está vinculada estrechamente con la altura, entre los 1500 y 3000 metros los valores medios varían entre los 10°C y 16°C.

A excepción de unos valles abrigados, puede afirmarse que todo el surco interandino, a causa del relieve, goza de temperaturas primaverales durante todos los meses del año (Quito 13,50 °C de temperatura media anual; Ibarra, más al norte, 15 °C; Riobamba en la zona central 13,5 °C; Cuenca, en el sur, 14,2 °C), pero las lluvias varían mucho de una "hoya" a otra.

En la zona regada por el río Guayllabamba, que corresponde al sector de Quito, las precipitaciones medias son de 1.041 mm al año; en la del Chota (Ibarra) bajan drásticamente a 480 mm; en la de Chimbo (Riobamba) descienden a 420 mm y en la de Paute (Cuenca) la cifra se eleva en cambio a 738 mm.

Son explicables las características climáticas de Vilcabamba, un valle localizado a poca distancia del extremo meridional y famoso en todo el mundo por la longevidad de sus habitantes: la temperatura media es de 19,6 °C y las precipitaciones de 2.000 mm al año, en una zona situada a 1.700 m sobre el nivel del mar.

En ciertos valles las lluvias suelen escasear (Chota 347 mm, Puéllaro 354 mm) y más arriba, sobre los páramos estériles, las temperaturas son decididamente frías y pueden situarse incluso a valores bajo cero.

A poca distancia de la línea equinoccial pero a 4.095 m de altura, el mes más cálido registra una temperatura media de 6 °C y el más frío de 3,3 °C; las precipitaciones, de 1.000 a 2.000 mm, caen en forma de nieve o granizo.

En toda la llanura litoral hasta una altura de 500 m en la ladera de la cordillera Occidental, el promedio anual de horas de brillo solar fluctúa entre las 600 y 1700 horas, siendo las más favorables de este número las zonas más secas.



Pese a la poca información existente, en la región amazónica se ha determinado que la insolación se ubica entre las 1000 y 1400 horas anuales.

En el Archipiélago de Colón, el promedio anual de insolación se ubica alrededor de las 2000 horas anuales.

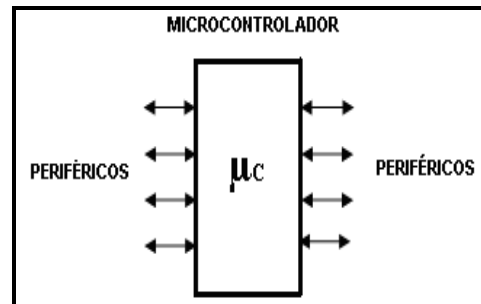
En estas tres últimas regiones la media de temperatura anual se establece entre los 24°C y 26°C, con extremos que raramente sobrepasan los 36°C o bajan a menos de los 14°C.

Por lo ya expuesto, el presente proyecto de automatización de apertura y cierre de persianas verticales posee un teclado el cual permite ajustar la temperatura de cierre y la temperatura de apertura manualmente, en otras palabras el usuario podrá fijar estas temperaturas a su preferencia.

Gracias a la implementación del teclado podemos superar el inconveniente de la gran diversidad de climas que poseemos en el Ecuador.

## 1.2. CARACTERÍSTICAS DEL MICROCONTROLADOR

### 1.2.1. ¿Qué es un microcontrolador?<sup>2</sup>



**Figura 1.1** El microcontrolador es un sistema cerrado. Todas las partes del computador están contenidas en su interior y sólo salen al exterior las líneas que gobiernan los periféricos.<sup>2</sup>

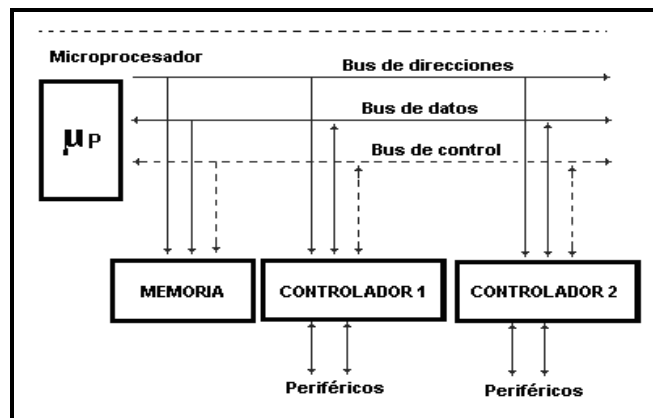
Es un circuito integrado programable con contiene todos los componentes de un computador, el cual se emplea para controlar el funcionamiento de una tarea asignada.

El microcontrolador en su memoria sólo reside un programa destinado a gobernar una aplicación determinada; sus líneas de entrada/salida soportan el conexionado de los sensores y actuadores del dispositivo a controlar y todos los recursos complementarios disponibles tienen como única finalidad atender sus requerimientos como se indica en la *Figura 1.1*.

Una vez programado y configurado el microcontrolador solamente sirve para gobernar la tarea específica.

<sup>2</sup> ANGULO, José Angulo, Ignacio. Microcontroladores PIC, Diseño Práctico de Aplicaciones, Editora McGrawHill, Págs. 1-4.

### 1.2.2. ¿Qué es un microprocesador? <sup>2</sup>



**Figura 1.2 Estructura de un sistema abierto basado en un microprocesador. La disponibilidad de los buses en el exterior permite que se configure a la medida de la aplicación. <sup>2</sup>**

El microprocesador es un circuito integrado que contiene la “**Unidad Central de Proceso**” (UCP) como indica la *Figura 1.2*. También llamada procesador, de un computador, la UCP está formada por la Unidad de Control, que interpreta las instrucciones, y el camino de datos, que las ejecuta.

Los pines de un microprocesador sacan al exterior las líneas de sus buses de direcciones, datos y control, para permitir conectarle con la Memoria y los Módulos de E/S y configura un computador implementando por varios circuitos integrados. Se dice que un microprocesador es un “**sistema abierto**” porque su configuración es variable de acuerdo con la aplicación a la que se destine.

<sup>2</sup> ANGULO, José Angulo, Ignacio. Microcontroladores PIC, Diseño Práctico de Aplicaciones, Editora McGrawHill, Págs. 1-4

### 1.2.3. Diferencias entre microprocesador y microcontrolador <sup>2</sup>

- Un microprocesador es un **sistema abierto** con el que puede construirse un computador con las características que se desee, acoplándole los módulos necesarios.
- Un microcontrolador es un **sistema cerrado** que contiene un computador completo y de prestaciones limitadas que no se pueden modificar.

Las ventajas más sobresalientes que podemos citar de los dispositivos microcontroladores son las siguientes:

- Aumento de prestaciones.
- Aumento de la fiabilidad.
- Reducción de tamaño en el producto acabado.
- Mayor flexibilidad.

### 1.2.4. Los recursos comunes de los microcontroladores <sup>3</sup>

Todos los microcontroladores están integrados en un chip, su estructura y sus características básicas son muy parecidas. Todos deben disponer de los bloques esenciales: procesador, memoria de datos y de instrucciones, líneas de E/S, oscilador externo y módulos controladores de periféricos.

Sin embargo, cada fabricante intenta enfatizar los recursos más idóneos para las aplicaciones a las que se destinan preferentemente.

---

<sup>2</sup> ANGULO, José Angulo, Ignacio. Microcontroladores PIC, Diseño Práctico de Aplicaciones, Editora McGrawHill, Págs. 1-4.

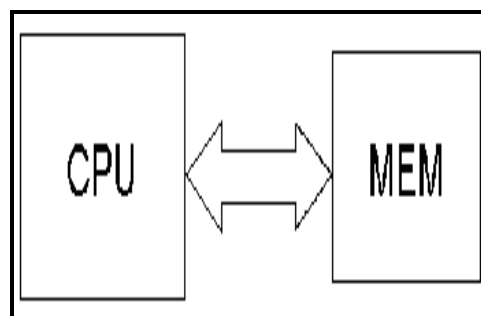
<sup>3</sup> [www.monografias.com/trabajos12/microco/microco.shtml](http://www.monografias.com/trabajos12/microco/microco.shtml)

A continuación se detallan los recursos que se hallan en los microcontroladores describiendo las diversas alternativas y opciones que pueden encontrarse según el modelo seleccionado.

#### 1.2.4.1. Arquitectura Básica<sup>3</sup>

La necesidad de conseguir elevados rendimientos en el procesamiento de las instrucciones ha desembocado en el empleo generalizado de procesadores de arquitecturas Harvard frente a las tradicionales que seguían la arquitectura de Von Neuman.

La arquitectura de Von Neumann *Figura 1.3*, se caracteriza por disponer de una sola memoria principal donde se almacenan datos e instrucciones de forma indistinta. A dicha memoria se accede a través de un sistema de buses único (direcciones, datos y control).

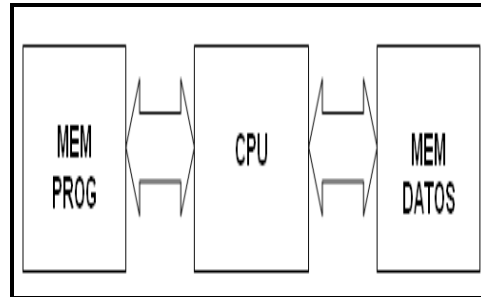


*Figura 1.3 Arquitectura Von Neumann.*<sup>4</sup>

La arquitectura Harvard *Figura 1.4*, dispone de dos memorias independientes una, que contiene sólo instrucciones y otra, sólo datos. Cada una dispone de su propio sistema de buses para el acceso y es posible realizar operaciones de acceso (lectura o escritura) simultáneamente en ambas memorias.

<sup>3</sup> [www.monografias.com/trabajos12/microco/microco.shtml](http://www.monografias.com/trabajos12/microco/microco.shtml).

<sup>4</sup> <http://usuarios.lycos.es/sfriswolker7pic7uno.htm>.



**Figura1.4 Arquitectura Harvard.** <sup>4</sup>

Esta dualidad además de propiciar el paralelismo, permite la adecuación del tamaño de las palabras y los buses a los requerimientos específicos de las instrucciones y de los datos.

También la capacidad de cada memoria es diferente al igual que sus respectivos sistemas de buses de acceso.

Los microcontroladores PIC (Peripheral Interface Controller) responden a la arquitectura Harvard. En las dos configuraciones anteriores, la CPU contiene a la ALU (Unidad Aritmética Lógica) la cual realiza las operaciones lógicas y aritméticas.

#### **1.2.4.2. El procesador o CPU** <sup>5</sup>

Es el elemento más importante del microcontrolador y determina sus principales características, tanto a nivel hardware como software. Se encarga de direccionar la memoria de instrucciones, recibir el código de la instrucción en curso, su decodificación y la ejecución de la operación que implica la instrucción, así como la búsqueda de los operandos y el almacenamiento del resultado.

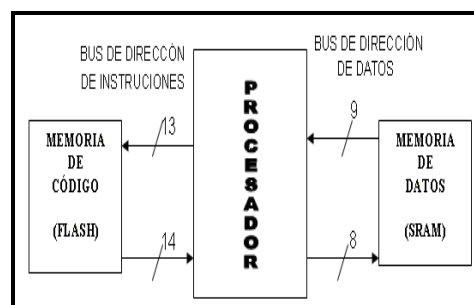
<sup>4</sup> <http://usuarios.lycos.es/sfriswolker7pic7uno.htm>

<sup>5</sup> COSTALES, Alcívar, Apuntes de microcontroladores

### 1.2.4.3. Procesador RISC con arquitectura Harvard <sup>6</sup>

El procesador de los modernos microcontroladores responde a la arquitectura RISC (Computadores de Juego de Instrucciones Reducido) en la *Figura 1.5*. En esta arquitectura se caracteriza por la independencia entre la memoria de código y la memoria de datos, facilitando el trabajo, en paralelo de las dos memorias, es decir cada una dispone de su propio sistema de buses para el acceso, lo que permite obtener altas cotas de rendimiento.

La filosofía RISC se refleja en el reducido número de instrucciones que forman su repertorio. Solo constan de 35 instrucciones que se ejecutan en un ciclo de instrucción equivalente a 4 ciclos de reloj, excepto las de salto que necesitan dos como es el caso en el PIC16F877A.



**Figura 1.5 Arquitectura Harvard <sup>6</sup>**

Además se introduce una segmentación en el procesador, dividiendo la ejecución de una instrucción en varias etapas. De esta forma se puede trabajar sobre varias instrucciones simultáneamente cada una en una etapa distinta.

<sup>6</sup> ANGULO, José Angulo, Ignacio. Microcontroladores PIC, Diseño Práctico de Aplicaciones, Tomoll, Editora McGrawHill, Págs. 22-23.

### **1.2.5. Memoria <sup>3</sup>**

#### **1.2.5.1. Memoria de programa <sup>3</sup>**

Es una memoria de almacenamiento no volátil (ROM, EPROM, EEPROM, FLASH), en la que se almacena el programa que gobierna la aplicación a la que está destinado el microcontrolador. No existen dispositivos de almacenamiento masivo por lo que todo el código debe estar almacenado en esta memoria. Por otro lado, al ser un circuito dedicado a una sola tarea debe almacenar un único programa.

#### **1.2.5.2. Memoria de datos <sup>6</sup>**

La memoria para almacenar datos debe ser de lectura y escritura, por lo que en general se usa memoria SRAM, aunque algunos microcontroladores llevan memoria EEPROM para evitar la pérdida de los datos en caso de corte en el suministro de corriente.

Los tamaños son mucho más reducidos que la memoria de programa; por ejemplo: el PIC16F877A consta de una memoria RAM de 368 bytes y 256 bytes de EEPROM.

#### **1.2.5.3. Líneas de Entrada/Salida (E/S) <sup>3</sup>**

La principal utilidad de los pines que posee el encapsulado que contiene un microcontrolador es soportar las líneas de E/S que comunican al computador interno con los periféricos exteriores.

Según los controladores de periféricos que posea cada modelo de microcontrolador, las líneas de E/S se destinan a proporcionar el soporte a las señales de entrada, salida y control.

---

<sup>3</sup> [www.monografias.com/trabajos12/microco/microco.shtml](http://www.monografias.com/trabajos12/microco/microco.shtml)

<sup>6</sup> ANGULO, José Angulo, Ignacio. Microcontroladores PIC, Diseño Práctico de Aplicaciones, Tomo II, Editora McGrawHill, Pág. 26.



Manejan la información en paralelo y se agrupan en conjuntos que reciben el nombre de puertas, pórticos o puertos.

Los pines de los puertos pueden ser todos configurados de acuerdo a la necesidad de la aplicación, es decir, que los pines de un mismo puerto pueden ser usados unos como entradas y otros como salidas.

Además, algunos pines E/S de los puertos son multiplexados a una función alternativa de características periféricas. En general, cuando una función de estas es habilitada, ese pin tal vez no pueda ser usado como un pin de propósito de E/S.

#### **1.2.5.4. Reloj principal <sup>3</sup>**

Todos los microcontroladores disponen de un circuito oscilador que genera una onda cuadrada de alta frecuencia, que configura los impulsos de reloj usados en la sincronización de todas las operaciones del sistema.

Generalmente, el circuito de reloj está incorporado en el microcontrolador y sólo se necesitan unos pocos componentes exteriores para seleccionar y estabilizar la frecuencia de trabajo.

Los microcontroladores admiten cuatro tipos de osciladores:

**Oscilador tipo “RC” <sup>6</sup>:** se trata de un oscilador de bajo costo formado por una simple resistencia y un condensador proporciona una estabilidad mediocre de la frecuencia, cuyo valor depende de la tolerancia de los dos elementos de la red R–C.

**Oscilador tipo “HS” <sup>6</sup>:** se trata de un oscilador que alcanza una frecuencia comprendida entre 4 y 12MHz y está basado en un cristal de cuarzo o un resonador cerámico.

---

<sup>3</sup> [www.monografias.com/trabajos12/microco/microco.shtml](http://www.monografias.com/trabajos12/microco/microco.shtml)

**Oscilador tipo "XT"**<sup>6</sup>: es un oscilador de cristal de cuarzo o resonador para frecuencias estándar comprendidas entre 100KHz y 4MHz.

**Oscilador tipo "LP"**<sup>6</sup>: es un oscilador de bajo consumo con cristal de cuarzo o resonador cerámico diseñado para trabajar en un rango de frecuencias de 35 a 200KHz.

En cualquier aplicación descrita anteriormente, el cristal de cuarzo o el resonador cerámico se coloca entre las pines OSC1 Y OSC2 del PIC, al aumentar la frecuencia de reloj supone disminuir el tiempo en que se ejecutan las instrucciones pero lleva aparejado un incremento del consumo de energía.

### 1.2.6. Recursos especiales<sup>3</sup>

Cada fabricante oferta numerosas versiones de una arquitectura básica de microcontrolador. En algunas extienden las capacidades de las memorias, en otras incorpora nuevos recursos, en otras reduce las prestaciones al mínimo para aplicaciones muy simples, etc.

La labor del diseñador es encontrar el modelo mínimo que satisfaga todos los requerimientos de su aplicación. De esta forma, minimizará el costo, el Hardware y el Software.

Los principales recursos específicos que incorporan los microcontroladores son:

- Temporizadores o "Timers".
- Perro guardián o "Watchdog" (WDT).
- Protección ante fallo de alimentación o "Brownout".
- Estado de reposo o de bajo consumo.
- Conversor A/D.
- Conversor D/A.

---

<sup>3</sup> [www.monografias.com/trabajos12/microco/microco.shtml](http://www.monografias.com/trabajos12/microco/microco.shtml)

<sup>6</sup> ANGULO, José Angulo, Ignacio. Microcontroladores PIC, Diseño Práctico de Aplicaciones, Tomo II, Editora McGrawHill, Pág. 55-56.

- Comparador analógico.
- Modulador de anchura de impulsos o PWM.
- Puertos de comunicación.

#### **1.2.6.1. Temporizadores o "Timers" <sup>3</sup>**

Se emplean para controlar periodos de tiempo (temporizadores) y para llevar la cuenta de acontecimientos que suceden en el exterior (contadores).

Para la medida de tiempos se carga un registro con el valor adecuado y a continuación dicho valor se va incrementando o decreciendo al ritmo de los impulsos de reloj o algún múltiplo hasta que se desborde y llegue a 0, momento en el que se produce un aviso.

Cuando se desean contar acontecimientos que se materializan por cambios de nivel o flancos en alguno de los pines del microcontrolador, el mencionado registro va incrementando o decreciendo al ritmo de dichos impulsos.

#### **1.2.6.2. Perro guardián o "Watchdog" (WDT) <sup>3</sup>**

El Perro guardián consiste en un temporizador que cuando se desborda y pasa por 0, provoca un reset automáticamente en el sistema. Se debe diseñar el programa de trabajo que controla la tarea de forma que refresque o inicialice al Perro guardián antes de que provoque el reset.

Si falla el programa o se bloquea, no se refrescará el Perro guardián y al completar su temporización, "ladrará y ladrará" hasta provocar el reset.

#### **1.2.6.3. Protección ante fallo de alimentación o "Brownout" <sup>3</sup>**

Se trata de un circuito que resetea al microcontrolador cuando el voltaje de alimentación ( $V_{DD}$ ) es inferior a un voltaje mínimo ("brownout").

---

<sup>3</sup> [www.monografias.com/trabajos12/microco/microco.shtml](http://www.monografias.com/trabajos12/microco/microco.shtml)

Mientras el voltaje de alimentación sea inferior al de brownout el dispositivo se mantiene reseteado, comenzando a funcionar normalmente cuando sobrepasa dicho valor.

#### **1.2.6.4. Estado de reposo o de bajo consumo <sup>3</sup>**

Son abundantes las situaciones reales de trabajo en que el microcontrolador debe esperar, sin hacer nada, a que se produzca algún acontecimiento externo que le ponga de nuevo en funcionamiento.

Para ahorrar energía (factor clave en los aparatos portátiles) los microcontroladores disponen de una instrucción especial (SLEEP en los PIC), que les pasa al estado de reposo o de bajo consumo, en el cual los requerimientos de potencia son mínimos.

En dicho estado se detiene el reloj principal y se "congelan" sus circuitos asociados, quedando el microcontrolador sumido en un profundo "sueño". Al activarse una interrupción ocasionada por el acontecimiento esperado, el microcontrolador se despierta y reanuda su trabajo.

#### **1.2.6.5. Conversor A/D (CAD) <sup>3</sup>**

Los microcontroladores que incorporan un conversor A/D (Analógico/Digital) pueden procesar señales analógicas, Ejemplo:

Temperatura, cantidad de luz, humedad, presión y otros.

Suelen disponer de un multiplexor que permite aplicar a la entrada del conversor análogo / digital (CAD) diversas señales analógicas desde los pines del circuito integrado.

---

<sup>3</sup> [www.monografias.com/trabajos12/microco/microco.shtml](http://www.monografias.com/trabajos12/microco/microco.shtml)

#### **1.2.6.6. Conversor D/A (CDA) <sup>3</sup>**

Transforma los datos digitales obtenidos del procesamiento de un dispositivo en su correspondiente señal analógica que se envía al exterior por uno de los pines del circuito integrado.

#### **1.2.6.7. Comparador analógico <sup>3</sup>**

Algunos modelos de microcontroladores disponen internamente de un amplificador operacional que actúa como comparador entre una señal fija de referencia y otra variable que se aplica por uno de los pines del circuito integrado.

La salida del comparador proporciona un nivel lógico 1 o 0 según una señal sea mayor o menor que la otra entrada.

También hay modelos de microcontroladores con un módulo de tensión de referencia que proporciona diversas tensiones que se pueden aplicar en los comparadores.

#### **1.2.6.8. Modulador de anchura de impulsos o PWM <sup>3</sup>**

Son circuitos que proporcionan en su salida pulsos de ancho variable, que se ofrecen al exterior a través de los pines del circuito integrado.

#### **1.2.6.9. Puertos de comunicación <sup>3</sup>**

Con objeto de dotar al microcontrolador de la posibilidad de comunicarse con otros dispositivos externos, otros buses de microprocesadores, buses de sistemas, buses de redes y poder adaptarlos con otros elementos bajo otras normas y protocolos.

---

<sup>3</sup> [www.monografias.com/trabajos12/microco/microco.shtml](http://www.monografias.com/trabajos12/microco/microco.shtml)

Algunos modelos como el PIC16F877A disponen de algunos de estos recursos que permiten directamente esta tarea, entre los que se destacan:

- **UART:** adaptador de comunicación serie asincrónica.
- **USART:** adaptador de comunicación serie sincrónica y asincrónica. Puerto paralelo esclavo para poder conectarse con los buses de otros microcontroladores.
- **USB (Universal Serial Bus):** es moderno bus serie para los PC.
- **Bus I2C:** interfaz serie de dos hilos desarrollado por Philips.

### 1.2.7. Microcontrolador 16F877A

#### 1.2.7.1. Estructura y características <sup>6</sup>

En este capítulo se justifica la elección del microcontrolador PIC16F877A, con el cual se procederá a revisar su estructura y sus características, procurando determinar si el mismo es suficiente para el diseño del “SISTEMA DE AUTOMATIZACIÓN DE APERTURA Y CIERRE DE PERSIANAS VERTICALES”.

La elección del Microcontrolador se basó en las siguientes ventajas que nos ofrece:

- Más resistencia a condiciones físicas externas.
- Recursos incorporados como conversores A/D, salida PWM, memoria de datos volátiles, etc.
- Protección ante fallos de alimentación.
- Código de protección programable.
- Programación de la memoria de programa más sencilla.
- Costos vs. prestaciones más accesibles.

---

<sup>6</sup>ANGULO, José Angulo, Ignacio. Microcontroladores PIC, Diseño Práctico de Aplicaciones, Tomo II, Editorial McGrawHill, Pág.4-5.

Entre más de cincuenta fabricantes de microcontroladores que se encuentran en el mundo, es muy difícil seleccionar “el mejor”, en realidad no existe, porque en cada aplicación son sus características específicas las que determinan el más conveniente.

Sin embargo, los factores siguientes son determinantes en su elección: sencillez de manejo, buena información, precio, buen promedio de parámetros (velocidad, consumo, tamaño, alimentación, código compacto), entre otros.

Para el caso de este proyecto se escogió uno de los microcontroladores del fabricante MicroChip, basándonos en la utilización más común en este tipo de proyectos, y por ser el más idóneo para la realización del mismo.

### **1.2.7.2. La Familia de los PIC <sup>2</sup>**

Microchip dispone de cuatro familias o gamas de microcontroladores de donde escoger, estas son:

*1.2.7.2.1. Gama Enana:* PIC16F(C) XXX con instrucciones de 12 o 14 bits, su característica principal es su reducido tamaño, al disponer todos sus componentes de 8 pines, trabaja con una alimentación en DC que va desde 2.5VDC a 5.5 VDC, consumen menos de 2mA y puede trabajar hasta 2MHz.

*1.2.7.2.2. Gama Baja o Básica:* PIC 16C5X, con instrucciones de 12 bits, se trata de una serie de PIC con recursos limitados, solo tienen manejo de instrucciones a nivel de bits. No tienen ningún otro recurso auxiliar.

*1.2.7.2.3. Gama Media:* PIC16F(C) XXX con instrucciones de 14 bits. Es la gama más variada y completa de los PIC, tienen encapsulados desde 18 pines hasta 68 e integran abundantes periféricos externos.

---

<sup>2</sup>ANGULO, José Angulo, Ignacio. Microcontroladores PIC, Diseño Práctico de Aplicaciones, Editora McGrawHill, Págs. 29-33

1.2.7.2.4. *Gama Alta:* PIC17CXXX con instrucciones de 16 bits. Tienen un sistema de gestión de interrupciones vectorizadas muy potentes, además de diversos periféricos externos y un multiplicador de Hardware a gran velocidad.

Dentro de la gama media, la serie del PIC16F87X y PIC 16F87XA, en general reúnen las mejores características de todas las gamas de familias, el que más recursos posee y mejor se adapta a las necesidades de este proyecto es el PIC16F877A cuyas características más sobresalientes son:

### 1.2.7.3. Como recursos fundamentales <sup>6</sup>

- Juego de 35 instrucciones de 14 bits de longitud, todas ellas se ejecutan en un ciclo de instrucción, menos las instrucciones de salto.
- Frecuencia de trabajo hasta 20 Mhz.
- Hasta 8k palabras de 14 bits para la memoria de código tipo flash.
- Hasta 368 bytes de memoria de datos RAM.
- Hasta 256 bytes de memoria de datos no volátil EEPROM.
- Hasta 14 fuentes de interrupción interna y externa.
- Pila con 8 niveles.
- Modo de direccionamiento directo, indirecto y relativo.
- Perro guardián (WDT).
- Código de protección programable.
- Modo SLEEP de bajo consumo.
- Voltaje de alimentación comprendido entre 2VDC y 5.5VDC.
- Bajo consumo (menos de 2mA a 5VDC y 4MHz).

---

<sup>6</sup>ANGULO, José Angulo, Ignacio. Microcontroladores PIC, Diseño Práctico de Aplicaciones, Tomo II, Editora McGrawHill, Pág.22



#### 1.2.7.4. Como dispositivos Periféricos <sup>6</sup>

- Timer 0: temporizador- contador de 8bits con predivisor.
- Timer 1: temporizador-contador de 16 bits, con predivisor.
- Timer 2: temporizador de 8bits, con predivisor y postdivisor.
- Dos módulos de PWM, captura y comparación.
- Conversor A/D de 10 bits.
- Puerto serie síncrono (SSP) con SPI e I2C.
- Puerto serie asíncrono (USART).
- Puerto Paralelo Esclavo (PSP).
- Una interrupción externa.

En la *Tabla 1.1* presentada a continuación se puede resumir las características principales del PIC 16F877A. La memoria FLASH del PIC soporta hasta 1.000 operaciones de escritura / borrado, mediante un proceso eléctrico que no precisa sacar el microcontrolador de su zócalo.

La memoria EEPROM para datos que tiene los PIC soportan 100.000 operaciones de grabado / borrado.

---

<sup>6</sup> ANGULO, José Angulo, Ignacio. Microcontroladores PIC, Diseño Práctico de Aplicaciones, Tomo II, Editora McGrawHill, Pág. 22.

<b>Característica</b>		<b>Descripción</b>
<b>Memoria de Programa (FLASH)</b>	<b>de BYTES</b>	14336
	<b>PALABRAS</b>	8192 X 14 bits
<b>Memoria de Datos</b>	<b>Bytes EEPROM</b>	256
	<b>BYTES DE RAM</b>	368
<b>Frecuencia de operación hasta los (Mhz)</b>		20
<b>Líneas de Entrada/Salida (E/S)</b>		33
<b>Fuentes de Interrupción</b>		14
<b>Convertor A/D</b>		8 (10 bits)
<b>Comunicación serial</b>		USART/ MSSP
<b>Comunicación paralelo</b>		SI
<b>ICSP (programación serie en circuito)</b>		SI
<b>Temporizadores</b>		TMR0,TMR1,TMR2
<b>Módulos de Captura/Comparación/PWM</b>		DOS
<b>Voltaje interno de referencia</b>		SI
<b>Voltaje de alimentación (Vdc)</b>		2.0 – 5.5
<b>Corriente máxima de entrada y salida (mA)</b>		25
<b>Conjunto de instrucciones</b>		35 de 14 bits

**Tabla 1.1 Características del Microcontrolador 16F877A.** <sup>6</sup>

#### **1.2.7.5. Aspecto Externo** <sup>6</sup>

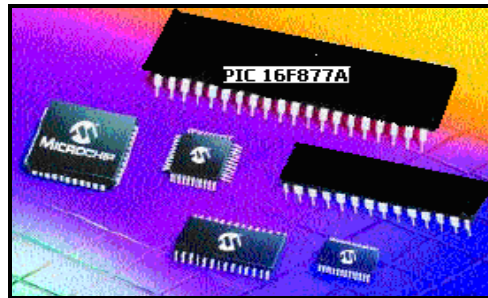
El PIC16F877A está fabricado con una tecnología CMOS, es decir que consume muy poca corriente pero a la vez es susceptible a daños por estática ( por lo que es muy recomendable utilizar pinzas para manipular o una manilla antiestática y así poder transportar desde el grabador al protoboard o viceversa), su encapsulado es de tipo DIP de 40 pines, cuatro de ellos soportan la tensión de alimentación ( $2V_{DD}, 2V_{SS}$ ), otros dos reciben la señal del oscilador externo (cristal

<sup>6</sup> ANGULO, José Angulo, Ignacio. Microcontroladores PIC, Diseño Práctico de Aplicaciones, Tomo II, Editora McGrawHill, Pág.4-6.

<sup>6</sup> ANGULO, José Angulo, Ignacio. Microcontroladores PIC, Diseño Práctico de Aplicaciones, Tomo II, Editora McGrawHill, Pág.25-28.

de cuarzo) y otro se utiliza para generar un reset (MCRL) o entrada de voltaje de programación verificación en el PIN 1 y los 33 pines restantes funcionan como líneas de E/S.

En la *Figura 1.6*, se presenta el aspecto externo del microcontrolador 16F877A.



*Figura 1.6* Microcontrolador 16F877A.<sup>5</sup>

#### 1.2.7.6. Organización de la memoria<sup>6</sup>

El PIC 16F877A tienen dos tipos de memoria; Memoria de Programa y Memoria de Datos, cada bloque con su propio bus: Bus de Datos y Bus de Programa.

**Memoria de programa<sup>6</sup>:** conocida también como memoria de instrucciones, aquí se escribe las órdenes para que el CPU las ejecute. El PIC16F877A tiene una memoria de programa no volátil tipo FLASH con una capacidad de 8k palabras de 14 bits cada una, dicha memoria está dividida en dos páginas de 4k cada una y está direccionada con el PC, que tiene un tamaño de 13 bits.

La pila tiene 8 niveles de profundidad y funciona automáticamente, no necesita instrucciones para guardar o sacar información de ella. El vector RESET ocupa la dirección 0000h, y el vector interrupción la posición 00004h como se indica en la *Figura 1.7*.

<sup>5</sup> COSTALES, Alcívar, Apuntes de microcontroladores.

<sup>6</sup> ANGULO, José Angulo, Ignacio. Microcontroladores PIC, Diseño Práctico de Aplicaciones, Tomo II, Editora McGrawHill, Pág.25-28.

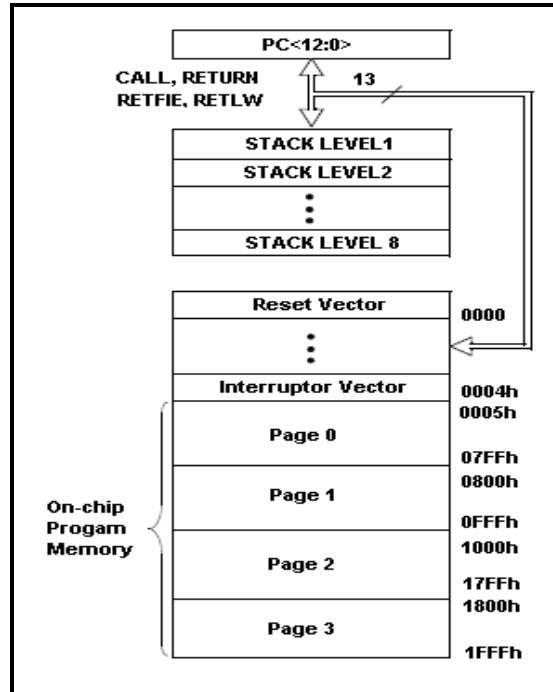


Figura 1.7 Mapa de la memoria de programa y pila.<sup>8</sup>

## Memoria de Datos<sup>6</sup>

**RAM:** se alojan los registros operativos fundamentales en el funcionamiento del procesador y en el manejo de todos sus periféricos, además de registros que el programador puede usar para información de trabajo propia de la aplicación. La memoria de datos está dividida en 4 bancos con 128 bytes cada uno.

En las posiciones iniciales de cada banco se encuentran los siguientes registros específicos que gobiernan al procesador y sus recursos. El total de la memoria es de 368x8 bytes como se indica en la *Figura 1.8*.

**EEPROM:** tiene 256 bytes de memoria, donde se puede almacenar datos y variables que interesa que no se pierdan cuando se desconecta la alimentación al sistema, es una memoria no volátil.

<sup>6</sup> ANGULO, José Angulo, Ignacio. Microcontroladores PIC, Diseño Práctico de Aplicaciones, Tomo II, Editora McGrawHill, Pág.25-28.

<sup>8</sup> <http://www.microchip.com/downloads/en/DeviceDoc/39582b.pdf>

INDF	00h	INDF	80h	INDF	100h	INDF	180h
TMRO	01h	OPTION_REG	81h	TMRO	101h	OPTION_REG	181h
PCL	02h	PCL	82h	PCL	102h	PCL	182h
STATUS	03h	STATUS	83h	STATUS	103h	STATUS	183h
FSR	04h	FSR	84h	FSR	104h	FSR	184h
PORTA	05h	TRISA	85h		105h		185h
PORTB	06h	TRISB	86h	PORTB	106h	TRISB	186h
PORTC	07h	TRISC	87h		107h		187h
PORTD (1)	08h	TRISD (1)	88h		108h		188h
PORTE (1)	09h	TRISE (1)	89h		109h		189h
PCLATH	0Ah	PCLATH	8Ah	PCLATH	10Ah	PCLATH	18Ah
INTCON	0Bh	INTCON	8Bh	INTCON	10Bh	INTCON	18Bh
PIR1	0Ch	PIE1	8Ch	EEDATA	10Ch	EECON1	18Ch
PIR2	0Dh	PIE 2	8Dh	EEADR	10Dh	EECON2	18Dh
TMR1L	0Eh	PCON	8Eh	EEDATH	10Eh	Resercado	18Eh
TMR1H	0Fh		8Fh	EEADRH	10Fh	Reservado	18Fh
T1CON	10h		90h		110h		190h
TMR2	11h	SSPCON2	91h	Registro de Propósito General 16 Bytes	111h	Registro de Propósito General 16 Bytes	191h
T2CON	12h	PR2	92h		112h		192h
SSPBUF	13h	SSPADD	93h		113h		193h
SSPCON	14h	SSPSTAT	94h		114h		194h
CCPR1L	15h		95h		115h		195h
CCPR1H	16h		96h		116h		196h
CCP1CON	17h		97h		117h		197h
RCSTA	18h	TXSTA	98h		118h		198h
TXREG	19h	SPBRG	99h		119h		199h
RCREG	1Ah		9Ah		11Ah		19Ah
CCPR2L	1Bh		9Bh		11Bh		19Bh
CCPR2H	1Ch		9Ch		11Ch		19Ch
CCP2CON	1Dh		9Dh		11Dh		19Dh
ADRESH	1Eh	ADRESL	9Eh		11Eh		19Eh
ADCON0	1Fh	ADCON1	9Fh		11Fh		19Fh
Registros de Propósito General 96 Bytes	20h	Registro de Propósito General 80 Bytes	A0h	Registro de Propósito General 80 Bytes	120h	Registro de Propósito General 80 Bytes	1A0h
			EFh		16Fh		1EFh
		Mapeados con 70h-7Fh	F0h	Mapeados con 70h-7Fh	170h	Mapeados con 70h-7Fh	1F0h
	7Fh		FFh		17Fh		1FFh
Banco 0		Banco 1		Banco2		Banco3	

Figura 1.8 División de la memoria de datos del microcontrolador. <sup>8</sup>

### 1.2.7.7. Puertos de Entrada/Salida <sup>6</sup>

El PIC 16F877A, tiene 33 líneas para comunicarse con el medio externo, todos estos puertos son multifuncionales, es decir, realizan diversas funciones según estén programadas. Sin embargo todas ellas tienen la capacidad de trabajar como líneas de E/S digitales. En la *Figura 1.9* se indica la distribución de cada uno de los puertos multifuncionales

<sup>6</sup> ANGULO, José Angulo, Ignacio. Microcontroladores PIC, Diseño Práctico de Aplicaciones, Tomo II, Editora McGrawHill, Pág.67-72.

<sup>8</sup> <http://www.microchip.com/downloads/en/DeviceDoc/39582b.pdf>.

## Puerto A

Sólo dispone de 6 líneas denominadas RA0 – RA5, son bidireccionales y su sentido queda configurado según la programación realizada.

**RA0/AN0, RA1/AN1Y y RA2/AN2/Vref-/CVref:** además de líneas de E/S digitales también pueden actuar como los canales 0, 1 y 2 por lo que se puede aplicar una señal analógica al convertor A/D.

**RA3/AN3/Vref +:** también puede actuar como entrada de la tensión de referencia para los periféricos que la precisan.

**RA4/TOCK1/C1OUT:** actúa como E/S digital y como entrada de señal de reloj para el timer 0.

**RA5/AN4/SS/C2OUT:** tiene multiplexada tres funciones, E/S digital, canal 4 para el convertor A/D y selección del modo esclavo cuando se trabaja con la comunicación serie síncrona.

## Puerto B

Dispone de 8 líneas bidireccionales cuya función se elige mediante la programación.

**RB0/INT, RB1, RB2, RB3/PGM, RB4, RB5:** son líneas de E/S digitales del puerto B, RB0/INT también sirve como entrada a una petición de interrupción externa.

**RB6/PGC:** es una línea de E/S digitales, también se introducen los pulsos de reloj, cuando se introduce la programación síncronamente en serie.

**RB7/PGD:** es una línea de E/S digitales, también se introducen los bits de datos serie, cuando se introduce la programación síncronamente en serie.

## Puerto C

Consta de 8 líneas bidireccionales cuyo sentido se configura mediante su programación, todas los pines tienen multiplexados diferentes funciones.

**RC0/T1OSO/T1CKI:** esta línea puede actuar como E/S digital, como salida del timer 1 o como entrada de impulsos para el timer 1.

**RC1/T1OSI/CCP2:** esta línea puede actuar como E/S digital, entrada al oscilador del timer 1, entrada del módulo de captura 2 , salida del comparador 2 , salida del PWM 2.

**RC2/CCP1:** esta línea puede actuar como E/S digital, entrada captura1, salida comparador 1, salida PWM1.

**RC3/SCK/SCL:** esta línea puede actuar como E/S digital, señal de reloj en modo SPI, señal de reloj en modo I2C.

**RC4/SDI/SDA:** esta línea puede actuar como E/S digital, entrada de datos en modo SPI, línea de datos en modo I2C.

**RC5/SDO:** esta línea puede actuar como E/S digital, salida de datos en modo SPI

**RC6/TX/CK:** esta línea puede actuar como E/S digital, línea de transmisión en USART, señal de reloj sincrónica en transmisión serie.

**RC7/RX/DT:** esta línea puede actuar como E/S digital, línea de recepción del USART, línea de datos en transmisión serie sincrónica.

## Puerto D

Consta de 8 líneas bidireccionales.

**RD0/PSP–RD7/PSP7:** además de usarse como líneas de E/S digitales normales, implementan un puerto paralelo esclavo de 8 líneas (PSP), que permiten la comunicación en paralelo con otros elementos del sistema.

## Puerto E

Consta de tres líneas bidireccionales, cuyo sentido se configura mediante su programación.

**RE0/RD/AN5:** además de usarse como líneas de E/S digitales, señal de lectura en el modo de puerta paralela esclava, canal 5 del conversor A/D.

**RE1/WR/AN6:** además de usarse como líneas de E/S digitales, señal de escritura en modo PSP, canal 6 del conversor A/D.

**RE2/CS/AN7** además de usarse como líneas de E/S digitales, selección de chip en el modo PSP, canal 7 del conversor A/D.

## Pines de polarización <sup>8</sup>

$V_{DD}$ : tensión de alimentación positiva (pines 11 y 32).

$V_{SS}$ : tierra o negativo de la alimentación (pines 12 y 31).

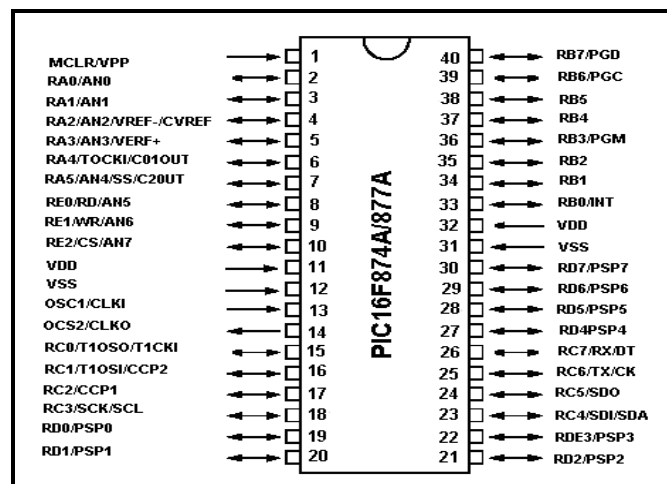


Figura 1.9 Distribución de pines del 16F877A. <sup>8</sup>

### 1.2.7.8. El reloj <sup>8</sup>

Este es el parámetro fundamental a la hora de establecer la velocidad de ejecución de las instrucciones y el consumo de energía.

<sup>8</sup> <http://www.microchip.com/downloads/en/DeviceDoc/39582b.pdf>



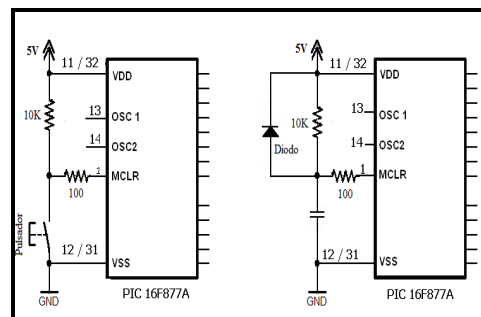
El microcontrolador 16F877A posee un oscilador interno RC de 4MHz, permite también utilizar un oscilador externo de hasta 20 MHz.

### 1.2.7.9 Reinicialización o reset <sup>6</sup>

El PIC 16F877A dispone de diversas maneras de reiniciarse que se citan a continuación:

- Reset por conexión de la alimentación (POR: Power – on Reset). El valor de la tensión de alimentación VDD sube entre 1,2 VDC a 1,7 VDC.
- Activación del pin MCLR, por un nivel bajo en dicho pin durante una operación normal.
- Activación del pin MCLR estando en el PIC trabajando en modo Reposo o SLEEP.
- Reset provocado por el desbordamiento del perro guardián en una operación normal.
- Reset provocado por el desbordamiento del perro guardián durante el estado de reposo.
- Reset provocado por una caída de voltaje (BORN: Brown – out- Reset) en VDD baja entre 3,8 VDC y 4,2VDC.

Existen dos circuitos de conexión usados para el reset o reinicialización como se indica en la *Figura 1.10*



**Figura 1.10 Circuitos usados para el Reset. <sup>5</sup>**

<sup>5</sup> COSTALES, Alcívar, Apuntes de microcontroladores.

<sup>6</sup> ANGULO, José Angulo, Ignacio. Microcontroladores PIC, Diseño Práctico de Aplicaciones, Tomo II, Editora McGrawHill, Pág.45-46.

### 1.2.8.0. Interrupciones <sup>6</sup>

Las interrupciones son desviaciones asincrónicas del flujo de control del programa originadas por diversos sucesos que no están bajo el control de las instrucciones del programa. Estos sucesos pueden ser internos o externos al sistema y en diseños industriales son un recurso muy importante para atender acontecimientos físicos en tiempo real. Cuando se produce una interrupción se detiene la ejecución del programa en curso, se salva el valor del PC en la pila y se carga aquel con el valor 0004h que es el vector de interrupción.

#### 1.2.8.0.1. Causas de la interrupción <sup>6</sup>

Existen catorce posibles causas de interrupción.

- Activación del pin RB0/INT.
- Desbordamiento del temporizador TMR0.
- Desbordamiento del temporizador TMR1.
- Desbordamiento del temporizador TMR2.
- Captura o comparación en el módulo CCP1.
- Captura o comparación en el módulo CCP2.
- Transferencia en el puerto serie síncrono.
- Transferencia en el puerto paralelo esclavo.
- Colisión de bus en el puerto serie síncrono.
- Cambio de estado de una de las entradas RB4 - RB7 del puerto B.
- Finalización de la escritura en la EEPROM de datos.
- Fin de la transmisión en el USART.
- Fin de la recepción en el USART.
- Fin de la conversión en el conversor A/D.

---

<sup>6</sup> ANGULO, José Angulo, Ignacio. Microcontroladores PIC, Diseño Práctico de Aplicaciones, Tomo II, Editora McGrawHill, Pág.74-75.

## 1.3. MOTOR PAPA O PASO A PASO <sup>9</sup>

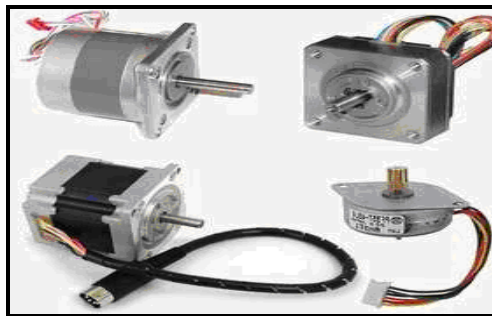
### 1.3.1. Introducción

En las diferentes aplicaciones de proyectos, es necesario convertir la energía eléctrica en energía mecánica, normalmente se utilizan los motores de corriente continua C.C y/o los de corriente alterna C.A.

Pero cuando lo que se requiere, es posicionamiento con un elevado grado de exactitud, una muy buena regulación de la velocidad, movimiento, enclavamiento y giros como en este proyecto, la solución es utilizar un motor paso a paso los cuales requieren para su funcionamiento una señal que es de naturaleza digital.

Nos vamos a centrar en los motores paso a paso (*stepping motor*), en el cual se realizará un estudio sobre su funcionamiento, los motores paso a paso se clasifican en dos tipos, según su diseño y fabricación pueden ser Bipolares o Unipolares, y las ventajas que este ofrece.

En la *Figura 1.11* se ve algunos modelos de motores paso a paso que existen.



**Figura 1.11** Diferentes modelos de los motores paso a paso bipolares y unipolares. <sup>10</sup>

<sup>9</sup> [http://www.emc.uji.es/d/IngMecDoc/344AmpIDisMaq/Curso\\_0203/webs\\_Alumnos/Web13\\_0203/MotoresPasoPaso.htm](http://www.emc.uji.es/d/IngMecDoc/344AmpIDisMaq/Curso_0203/webs_Alumnos/Web13_0203/MotoresPasoPaso.htm).

<sup>10</sup> [http://es.wikipedia.org/wiki/Motor\\_paso\\_a\\_paso](http://es.wikipedia.org/wiki/Motor_paso_a_paso) clases de motores.

Sus principales aplicaciones se pueden encontrar en robótica, los mismos que son ideales para la construcción de mecanismos en donde se requieren movimientos muy precisos, en la tecnología aeroespacial, en computadores (discos duros, unidades de CD-ROM o de DVD e impresoras), maquinarias (tornos, fresadoras, bordadoras), etc.

Estos motores paso a paso una de sus características más comunes es la habilidad de poder quedar enclavados en una posición o bien totalmente libres.

Si una o más de sus bobinas están energizadas, el motor estará enclavado en la posición correspondiente y por el contrario quedará completamente libre si no circula corriente por ninguna de sus bobinas.

**1.3.1.1. Definición:** un motor paso a paso se diferencia de un motor convencional en que en éste se puede posicionar su eje en posiciones fijas o pasos, pudiendo mantener la posición.

El motor paso a paso gira con movimientos discretos, llamados pasos. Después de que el rotor da un paso, deja de girar hasta que recibe el siguiente comando.

### **1.3.2. Parámetros del motor paso a paso <sup>9</sup>**

#### **1.3.2.1. Características mecánicas.**

**Par dinámico o de trabajo:** es el momento máximo que el motor es capaz de desarrollar sin perder paso, es decir, sin dejar de responder a algún impulso de excitación del estator y dependiendo, de la carga. El fabricante ofrece las curvas denominadas de arranque sin error y que relaciona el par en función del número de pasos.

---

<sup>9</sup>[http://www.emc.uji.es/d/IngMecDoc/344AmpIDisMaq/Curso\\_0203/webs\\_Alumnos/Web13\\_0203/MotoresPasoPaso.htm](http://www.emc.uji.es/d/IngMecDoc/344AmpIDisMaq/Curso_0203/webs_Alumnos/Web13_0203/MotoresPasoPaso.htm)

**Par de mantenimiento:** es el par requerido para desviar, en régimen de excitación, un paso el rotor cuando la posición anterior es estable; es mayor que el par dinámico y actúa como freno para mantener el rotor en una posición estable dada.

**Par de detención:** es un par de freno que, siendo propio de los motores de imán permanente, es debido a la acción del rotor cuando los devanados estáticos están desactivados. Estos momentos se expresan en miliNewton por metro.

**Momento de inercia del rotor:** es su momento de inercia asociado que se expresa en gramos por centímetro cuadrado.

**Ángulo de paso:** se define como el avance angular que se produce en el motor paso a paso por cada impulso de excitación, el cual se mide en *grados*; siendo este factor importante al momento de elegir un motor paso a paso, para un uso determinado. Este define la cantidad de grados que rotará el eje para cada paso completo. Una operación de medio-paso o semi-paso (half step) del motor duplicará la cantidad de pasos por revolución al reducir la cantidad de grados por paso. De acuerdo a la aplicación que deben realizar los motores paso a paso tiene diferentes grados de precisión como muestra en la *Tabla 1.2*:

Grados que gira por Impulso de excitación	Número de pasos para Llegar a 360°
0,72°	500
1,8°	200
3,75°	96
7,5°	48
15°	24
90°	4

**Tabla 1.2** Número de pasos que debe dar un motor PAP para llegar a dar una vuelta completa, según su ángulo de giro.<sup>9</sup>

<sup>9</sup> [http://www.emc.uji.es/d/IngMecDoc/344AmplIDisMaq/Curso\\_0203/webs\\_Alumnos/Web13\\_0203/MotoresPasoaPaso.htm](http://www.emc.uji.es/d/IngMecDoc/344AmplIDisMaq/Curso_0203/webs_Alumnos/Web13_0203/MotoresPasoaPaso.htm)

Cuando el valor de grados por paso no está indicado en el motor, es posible contar a mano la cantidad de pasos por vuelta, haciendo girar el motor y sintiendo por el tacto cada "diente" magnético. Los grados por paso se calculan dividiendo 360 (una vuelta completa) por la cantidad de pasos que se contaron, las cantidades más comunes de grados se muestran en la tabla anterior, a este valor de grados por paso usualmente se le llama la resolución del motor.

En el caso de que un motor no indique los grados por paso en su carcasa, pero sí la cantidad de pasos por revolución, al dividir 360 por ese valor se obtiene la cantidad de grados de paso por vuelta, siendo de la siguiente manera:

**Número de pasos por vuelta:** se define como la cantidad de pasos que ha de efectuar el rotor para realizar una revolución completa.

$$NP = \frac{360}{\alpha}$$

Donde:

**NP** = es el número de pasos

**$\alpha$**  = es el ángulo de paso.

Por ejemplo:

Un motor de 200 pasos por vuelta, tendrá una resolución de 1,8° por paso.

**Velocidad máxima de paso:** es el número máximo de pasos por segundo. Inversamente proporcional a la masa (inercia) del rotor. Cuando la masa y la inercia aumentan la velocidad máxima de paso decrece.

**Frecuencia de paso máxima:** es el máximo número de pasos por segundo que el rotor puede efectuar obedeciendo a los impulsos de control.

**Voltaje:** los motores paso a paso tienen una tensión eléctrica de trabajo. Este valor viene impreso en su carcasa o por lo menos se especifica en su hoja de datos. Algunas veces puede ser necesario aplicar un voltaje superior para lograr

que un determinado motor cumpla con el torque deseado, pero esto producirá un calentamiento excesivo y/o acortará la vida útil del motor.

**Resistencia eléctrica:** Otra característica de un motor paso a paso es la resistencia de los bobinados. Esta resistencia determinará la corriente que consumirá el motor, y su valor afecta la curva de torque del motor y su velocidad máxima de operación.

### 1.3.2.1. Características físicas de un motor paso a paso

En la mayoría de los motores paso a paso vienen sus características más importantes en una placa, en nuestro caso se utilizó un motor paso a paso de 5 hilos unipolar y su placa dice:

<p><b>Stepping MOTOR CBKS-12</b></p>
<p><b>VOLT = 5V COIL = 3.3 Ω Deg/step = 1,8°</b></p>

**Tabla 1.3 Placa del Motor paso a paso unipolar de 5 hilos.** <sup>12</sup>

### 1.3.3. Tipos de motores paso a paso <sup>9</sup>

Los motores paso a paso se dividen en dos categorías principales: de imán permanente y de reluctancia variable. También existe una combinación de ambos, a los que se les llama híbridos.

<sup>12</sup> Creadores y Autores del Proyecto.

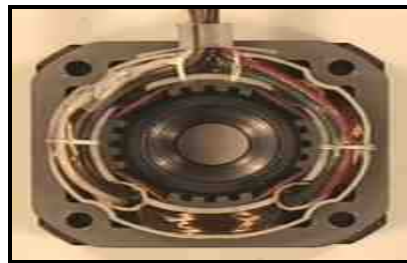
<sup>9</sup>[http://www.emc.uji.es/d/IngMecDoc/344AmpIDisMaq/Curso\\_0203/webs\\_Alumnos/Web13\\_0203/motoresPasoaPaso.htm](http://www.emc.uji.es/d/IngMecDoc/344AmpIDisMaq/Curso_0203/webs_Alumnos/Web13_0203/motoresPasoaPaso.htm)

**1.3.3.1. Los de imán permanente:** son los más comunes, se los utiliza, en el avance de papel y del cabezal de impresión de las impresoras, en el movimiento del cabezal de las disketteras, etc.

Su rotor es un imán permanente que está ranurado (como se indica en la *Figura 1.12*) en toda su longitud y el estator está formado por una serie de bobinas enrolladas alrededor de un núcleo o polo como se muestra en la *Figura 1.13*. Como su nombre indica, poseen un imán que aporta el campo magnético para la operación.



*Figura 1.12 Rotor.*<sup>10</sup>



*Figura 1.13 Estator de 4 bobinas.*<sup>10</sup>

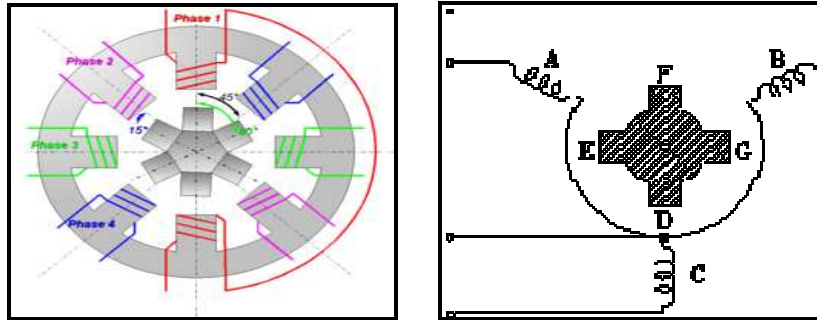
**1.3.3.2. Los motores del tipo de reluctancia variable:** en cambio, poseen un rotor de hierro dulce que en condiciones de excitación del estator, y bajo la acción de su campo magnético, ofrece menor resistencia a ser atravesado por su flujo en la posición de equilibrio.

Su mecanización es similar a los de imán permanente y su principal inconveniente radica en que en condiciones de reposo (sin excitación) el rotor queda en libertad de girar y, por lo tanto, su posicionamiento de régimen de carga dependerá de su inercia y no será posible predecir el punto exacto de reposo.

El tipo de motor de reluctancia variable consiste en un rotor y un estator cada uno con un número diferente de dientes, el rotor no tienen un imán permanente, y gira libremente, porque no presenta un torque de detección En la *Figura 1.14a* se indica su estructura interna, y en la *Figura 1.14 b* el diagrama de conexión

<sup>10</sup> [http://es.wikipedia.org/wiki/Motor\\_paso\\_a\\_pasoclasas](http://es.wikipedia.org/wiki/Motor_paso_a_pasoclasas) de motores.





3 **Figura 1.14.a Estructura interna.**<sup>10</sup> **Figura 14b Diagrama interno de conexión.**<sup>10</sup>

**1.3.3.3. Los motores híbridos:** combinan las mejores características de los de reluctancia variable y de imán permanente. Se construyen con estatores multidentados y un rotor de imán permanente. Los motores híbridos estándar tienen 200 dientes en el rotor y giran en pasos de  $1,8^\circ$ .

Existen motores híbridos con configuraciones de  $0,9^\circ$  y  $3,6^\circ$ . Dado que poseen alto torque estático y dinámico y se mueven a muy altas velocidades de pulso, se los utiliza en una amplia variedad de aplicaciones industriales.

En éste proyecto se utilizará un motor paso a paso de imán permanente, por tal motivo daremos más énfasis en su estudio de funcionamiento.

### 1.3.4. Motores paso a paso de imán permanente<sup>9</sup>

#### 1.3.4.1. Principio de funcionamiento

Básicamente estos motores están constituidos normalmente por un rotor sobre el que van aplicados distintos imanes permanentes y por un cierto número de bobinas excitadoras bobinadas en su estator. Las bobinas son parte del estator y el rotor es un imán permanente.

Toda la conmutación (o excitación de las bobinas) deber ser externamente manejada por un controlador electrónico y, típicamente, los motores y sus

<sup>10</sup> [http://es.wikipedia.org/wiki/Motor\\_paso\\_a\\_paso](http://es.wikipedia.org/wiki/Motor_paso_a_paso) clases de motores.

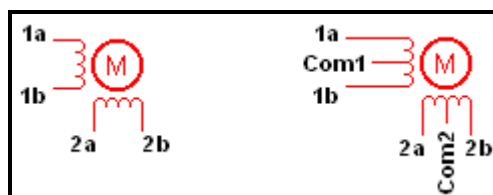
<sup>9</sup> [http://www.emc.uji.es/d/IngMecDoc/344AmpIDisMaq/Curso\\_0203/webs\\_Alumnos/Web13\\_0203/MotoresPasoPaso.htm](http://www.emc.uji.es/d/IngMecDoc/344AmpIDisMaq/Curso_0203/webs_Alumnos/Web13_0203/MotoresPasoPaso.htm).

controladores se diseñan de manera que el motor se pueda mantener en una posición fija y también para que se lo pueda hacer girar en un sentido y en el otro. La mayoría de los motores paso a paso conocidos avanzan a frecuencias de audio, lo que les permite girar muy velozmente.

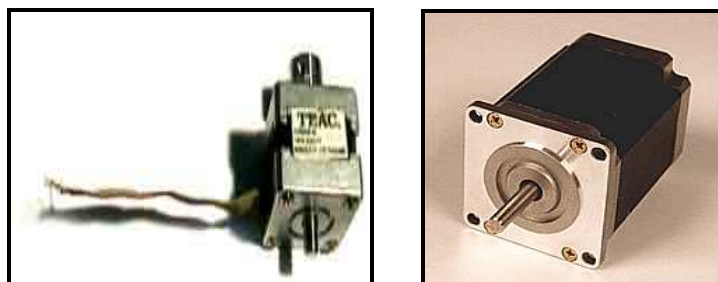
Con un controlador apropiado, se los puede hacer arrancar y detenerse en un instante en posiciones controladas. Los motores paso a paso de imán permanente se dividen a su vez en distintos tipos, diferenciados por el tipo de bobinado.

Existen entonces motores paso a paso de imán permanente **unipolares** (también llamados unifilares), **bipolares** (también llamados bifilares).

En la *Figura 1.15* se diferencian las conexiones internas que existen entre los dos motores de imán permanente, en las *Figuras 1.16a* y *1.16b* se pueden observar como son físicamente los dos motores paso a paso.



*Figura 1.15 Motor permanente unipolar y bipolar.*<sup>13</sup>



*Figura 1.16a Motor paso a paso unipolar de 4 hilos. Figura 1.16b Motor paso a paso bipolar.*<sup>14</sup>

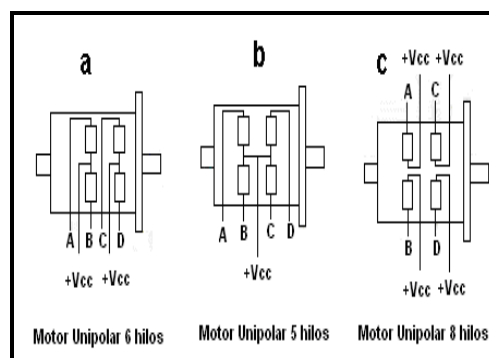
<sup>13</sup> <http://www.forosdeelectronica.com/about22164.html>.

<sup>14</sup> [http://www.sapiensman.com/motores\\_por\\_pasos/](http://www.sapiensman.com/motores_por_pasos/).

### 1.3.4.2. Clasificación de los motores de imán permanente

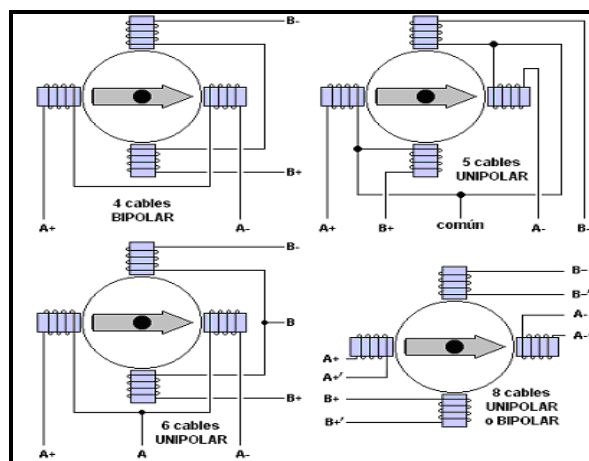
#### 1.3.4.2.1. Motores paso a paso unipolares:

Los motores unipolares son relativamente fáciles de controlar, gracias a que poseen devanados duplicados, estos motores suelen tener 6, 5 o 8 cables de salida, dependiendo si el común está unido internamente o no como se muestra en la siguiente *Figura 1.17*.



**Figura 1.17. Disposición de las bobinas de motores paso a paso a) unipolar a 6 hilos b) unipolar a 5 hilos c) unipolar a 8 hilos.**<sup>9</sup>

En la *Figura 1.18* se indica los diagramas de conexión de los diferentes motores paso a paso de imán permanente.

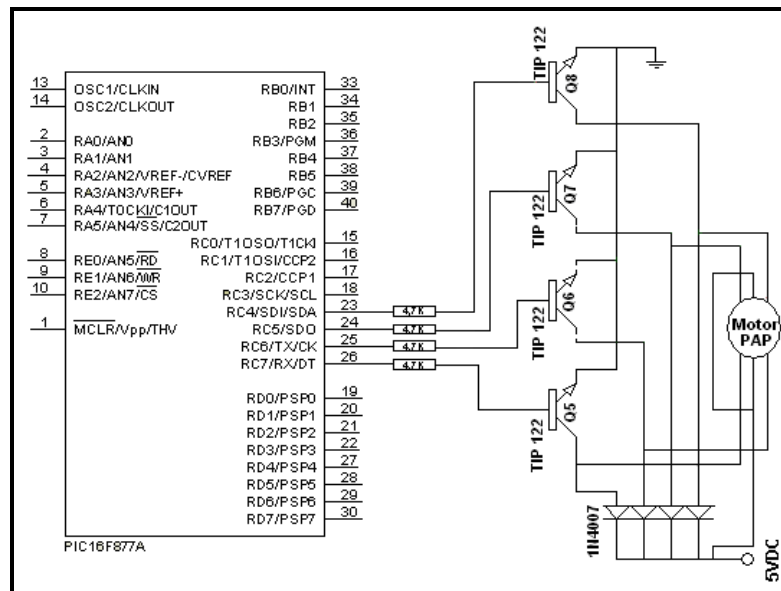


**Figura 1.18 Las diferentes conexiones internas de los motores de imán permanente.**<sup>14</sup>

<sup>9</sup>[http://www.emc.uji.es/d/IngMecDoc/344AmpIDisMaq/Curso\\_0203/webs\\_Alumnos/Web13\\_0203/MotoresPasoAPaso.htm](http://www.emc.uji.es/d/IngMecDoc/344AmpIDisMaq/Curso_0203/webs_Alumnos/Web13_0203/MotoresPasoAPaso.htm).

<sup>14</sup>[http://www.sapiensman.com/motores\\_por\\_pasos/](http://www.sapiensman.com/motores_por_pasos/).

En la *Figura 1.19* se muestra el diagrama de control utilizado para el control del motor unipolar. Este circuito consta de transistores TIP 22 para soportar la corriente que demanda el motor.



**Figura 1.19** Conexión de un motor paso a paso unipolar de 5 hilos a las salidas del PIC. <sup>12</sup>

Para controlar los motores paso a paso unipolares existen secuencias que son las siguientes:

### Paso simple <sup>15</sup>

Esta secuencia de pasos es la más simple de todas y consiste en activar cada bobina una a una y por separado, con esta secuencia de encendido de bobinas no se obtiene mucha fuerza ya que solo es una bobina cada vez la que arrastra y sujeta el rotor del eje del motor. Como se indica en la *Figura 1.20* y *Figura 1.21* la secuencia de pasos en el diagrama de tiempo y energizado del motor paso a paso simple.

<sup>12</sup> Creadores y Autores del Proyecto.

<sup>15</sup> <http://www.todorobot.com.ar/informacion/tutorial%20stepper/stepper-tutorial.htm>.

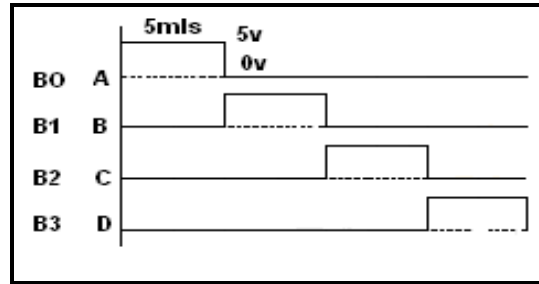


Figura 1.20 Diagrama de tiempo.<sup>7</sup>

PASO	Bobina A	Bobina B	Bobina C	Bobina D	
1	ON	OFF	OFF	OFF	
2	OFF	ON	OFF	OFF	
3	OFF	OFF	ON	OFF	
4	OFF	OFF	OFF	ON	

Figura 1.21 Energizado de bobinas para la secuencia de paso simple.<sup>15</sup>

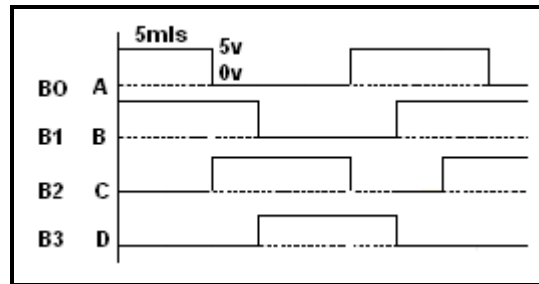
### Paso doble<sup>15</sup>

Esta es la secuencia más usada y la que generalmente recomienda el fabricante, su funcionamiento es el siguiente se va activando las bobinas de dos en dos con lo que hacemos un campo magnético más potente que atraerá con más fuerza y retendrá el rotor en el mismo sitio. Los pasos también serán algo más bruscos debidos a que la acción del campo magnético es más poderosa que en la secuencia anterior, cabe recordar que en esta modalidad el motor consume el

<sup>7</sup> REYES, Carlos, Aprenda rápidamente a programar microcontroladores, Pág. 121.

<sup>15</sup> <http://www.todorobot.com.ar/informacion/tutorial%20stepper/stepper-tutorial.htm>.

40% más de corriente que el caso anterior. Dado por el mejor torque que presenta esta configuración es la que emplearemos para el desarrollo del proyecto como se indica en la *Figura 1.22* y la *Figura 1.23* la secuencia de pasos en el diagrama de tiempo y energizado para el manejo del motor a medio paso.



*Figura 1.22 Diagrama de tiempo.*<sup>7</sup>

PASO	Bobina A	Bobina B	Bobina C	Bobina D	
1	ON	ON	OFF	OFF	
2	OFF	ON	ON	OFF	
3	OFF	OFF	ON	ON	
4	ON	OFF	OFF	ON	

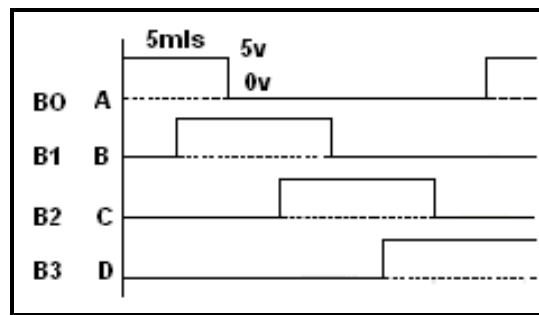
*Figura 1.23 Energizado de bobinas para la secuencia de paso doble.*<sup>15</sup>

<sup>15</sup> <http://www.todorobot.com.ar/informacion/tutorial%20stepper/stepper-tutorial.htm>.

<sup>7</sup> REYES, Carlos, Aprende rápidamente a programar microcontroladores, Pág. 124.

## Medio Paso <sup>15</sup>

Esta es una combinación de las dos secuencias anteriores, se energiza dos bobinas, luego una bobina, luego nuevamente otras dos bobinas y así alternadamente como se indica en la *Figura 1.24*, la secuencia de pasos en el diagrama de tiempo como resultado el rotor avanza medio paso por cada pulso de excitación, la ventaja de esta secuencia es la disminución del avance angular, de  $1.8^\circ$  a  $0.9^\circ$ , por consiguiente para girar una vuelta completa ( $360^\circ$ ) se requiere el doble de pasos, en el caso del proyecto sería 400 pasos. En la *Figura 1.25* podemos ver como se van energizando las bobinas del motor internamente.



*Figura 1.24 Diagrama de tiempo.* <sup>7</sup>

PASO	Bobina A	Bobina B	Bobina C	Bobina D	
1	ON	OFF	OFF	OFF	
2	ON	ON	OFF	OFF	
3	OFF	ON	OFF	OFF	

<sup>7</sup> REYES, Carlos, Aprenda rápidamente a programar microcontroladores, Pág. 125.

<sup>15</sup> <http://www.todorobot.com.ar/informacion/tutorial%20stepper/stepper-tutorial.htm>.

4	OFF	ON	ON	OFF	
5	OFF	OFF	ON	OFF	
6	OFF	OFF	ON	ON	
7	OFF	OFF	OFF	ON	
8	ON	OFF	OFF	ON	

**Figura 1.25 Energizado de bobinas para la secuencia de medio paso.** <sup>15</sup>

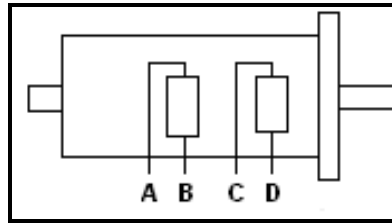
#### 1.3.4.2.2. Motores paso a paso bipolares <sup>16</sup>:

Los motores bipolares tienen generalmente 4 cables de salida como se muestra en la *Figura 1.26*, requieren circuitos de control y de potencia más complejos. Pero en la actualidad esto no es problema, ya que estos circuitos se suelen implementar en un integrado, que soluciona esta complejidad en un solo componente.

<sup>15</sup> <http://www.todorobot.com.ar/informacion/tutorial%20stepper/stepper-tutorial.htm>.

<sup>16</sup> <http://www.x-robotics.com/lcd.htm>.





**Figura1.26 Motor Bipolar con 4 cables de salida.**<sup>9</sup>

Como mucho se deben agregar algunos componentes de potencia, como transistores y diodos para las contracorrientes, aunque esto no es necesario en motores pequeños y medianos.

Como no tienen el doble bobinado de los unipolares ya que en éstos todo el tiempo se está utilizando sólo una de las bobinas duplicadas, mientras la otra queda desactivada y sin ninguna utilidad.

### **1.3.5. Identificación de la bobinas de un motor paso a paso bipolar**<sup>12</sup>

Uno de los mayores inconvenientes a la hora de trabajar con los motores paso a paso en especial los unipolares es la de poder identificar cuál es la bobina A, B, C y D para esto simplemente medimos las resistencias de cada una de las bobinas, (para nuestro aplicación utilizaremos un motor paso a paso de 5 hilos).

Las características del motor paso a paso del proyecto es de  $1,8^\circ$  de movimiento con una alimentación de 5 voltios, cuyas bobinas tienen una resistencia de  $3,3 \Omega$  cada una. En algunos motores de paso vienen las bobinas identificadas con una gama de colores, en este caso no tenía, para poder identificar las resistencias de las bobinas, se mide un cable con otro cable, si marca  $3,3 \Omega$  se está midiendo una bobina con el común, y si marca el doble  $6,3 \Omega$  se está midiendo bobina con bobina.

Para poder identificar la secuencias de las bobinas A, B, C, y D se procede de la siguiente manera, una vez identificado el común del motor lo conectamos a

<sup>9</sup>[http://www.emc.uji.es/d/IngMecDoc/344AmpIDisMaq/Curso\\_0203/webs\\_Alumnos/Web13\\_0203/MotoresPasoPaso.htm](http://www.emc.uji.es/d/IngMecDoc/344AmpIDisMaq/Curso_0203/webs_Alumnos/Web13_0203/MotoresPasoPaso.htm)

<sup>12</sup> Creadores y Autores del Proyecto.

Vcc = 5VDC, con el terminal negativo de la fuente se comienza a hacer pruebas con los 4 cables restantes del motor, puede presentarse casos como por ejemplo la secuencia escogida aleatoriamente no es la correcta y el motor comienza a moverse en sentido horario y antihorario, o el motor se queda temblando en el mismo lugar, se debe seguir probando y cambiando las secuencias hasta que el motor empiece a girar correctamente, por último se identifica las bobinas A, B, C y D.

## **1.4. PANTALLA DE CRISTAL LÍQUIDO (LCD) <sup>7</sup>**

### **1.4.1. Introducción**

La pantalla de cristal líquido es uno de los visualizadores más utilizados en la actualidad debido a las importantes ventajas que ofrece, por ejemplo permite mostrar mensajes que indican al operario el estado de la máquina, o para dar instrucciones de manejo, mostrar valores, etc.

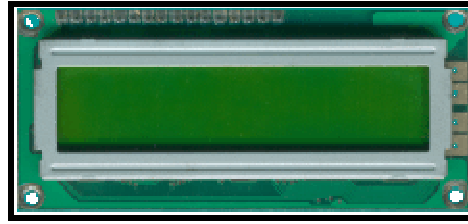
Es decir que permite la comunicación entre máquinas y humanos, ya que los mensajes que se visualizan en la pantalla del LCD pueden mostrar cualquier código ASCII, introduciendo el código correspondiente de cada uno de los caracteres a visualizar.

En el mercado existen varias presentaciones por ejemplo de 2 líneas por 8 caracteres, 2x16, 2x20, 4x20, etc. A pesar de que la variedad de modelos de LCD es muy grande, las líneas necesarias para su conexión y control son prácticamente las mismas.

Para la visualización del proyecto se utiliza un LCD 2x16, como se muestra en la *Figura 1.27* y a continuación se indica sus aspectos más importantes.

---

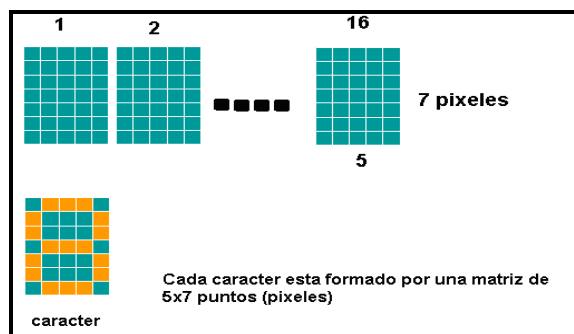
<sup>7</sup> Reyes Carlos, Aprenda rápidamente a programar microcontroladores, Págs. 79 - 82.



**Figura 1.27 LCD 2 X16.** <sup>16</sup>

#### 1.4.2. Descripción del LCD 2X16 <sup>5</sup>

La pantalla de cristal líquido o LCD (Liquid Crystal Display) es un dispositivo microcontrolador de visualización gráfico para la presentación de caracteres, símbolos o incluso dibujos (en algunos modelos), es este caso dispone de 2 filas de 16 caracteres cada una y cada carácter dispone de una matriz de 5x7 puntos (pixels), aunque los hay de otro número de filas y caracteres, como se indica en la *Figura 1.28*



**Figura 1.28 Matriz de puntos.** <sup>5</sup>

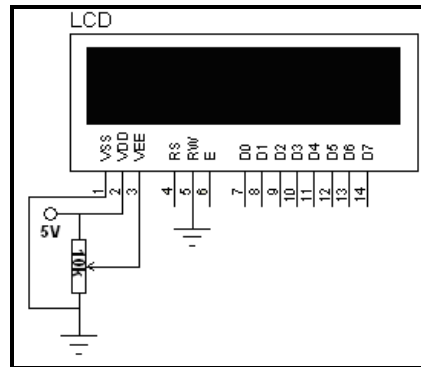
Este dispositivo está gobernado internamente por un microcontrolador Hitachi 44780 y regula todos los parámetros de presentación, este modelo es el más comúnmente usado.

<sup>5</sup> COSTALES, Alcívar, Apuntes de microcontroladores.

<sup>16</sup> <http://www.x-robotics.com/lcd.htm>.

### 1.4.3. Descripción de pines <sup>16</sup>

En la *Figura 1.29* se puede visualizar la distribución del pines del LCD 2x16, y en la *Tabla 1.4* la función que cumple cada pin.



**Figura 1.29** Distribución de pines del LCD. <sup>12</sup>

Pin	Símbolo	Descripción
1	Vss	Tierra de alimentación GND
2	Vdd	Alimentación de +5 Vcc
3	Vo	Ajuste del contraste del cristal líquido (0 a+5 V)
4	Rs	Selección del registro control/datos Rs=0 reg. control RS= 1 reg. datos
5	R/W	Lectura / Escritura en LCD R/W =0 escritura R/W =1 lectura
6	E	Habilitación
7	D0	Bit menos significativo ( bus de datos bidireccional)
8	D1	
9	D2	
10	D3	
11	D4	
12	D5	
13	D6	
14	D7	Bit más significativo ( bus de datos bidireccional)

**Tabla 1.4** Función de cada pin del LCD. <sup>7</sup>

<sup>7</sup> Reyes Carlos, Aprenda rápidamente a programar microcontroladores, Pág. 79 - 80.

<sup>12</sup> Creadores y Autores del Proyecto.

<sup>16</sup> <http://www.x-robotics.com/lcd.htm>.

#### 1.4.4. Características principales <sup>16</sup>

- Pantalla de caracteres ASCII.
- Desplazamiento de los caracteres hacia la izquierda o la derecha.
- Proporciona la dirección de la posición absoluta o relativa del carácter.
- Memoria de 40 caracteres por línea de pantalla.
- Movimiento del cursor y cambio de su aspecto.
- Permite que el usuario pueda programar 8 caracteres.
- Conexión a un procesador usando un interfaz de 4 u 8 bits.

#### 1.4.5. Funcionamiento <sup>7</sup>

Para comunicarse con la pantalla LCD podemos hacerlo por medio de sus pines de entrada de dos maneras posibles, con bus de 4 bits o con bus de 8 bits, estos dos se diferencian en el tiempo de retardo, pues la comunicación a 4bits, primero envía los 4 bits más altos y luego los 4 bits más bajos, mientras que la de 8 bits envía todo al mismo tiempo, esto no es un inconveniente si consideramos que el LCD trabaja en microsegundos ( $\mu$ s). Para la aplicación del proyecto se utiliza la conexión a 4 bits, debido a que se deben conectar pocos cables , sólo se debe conectar el bit de registro, el Enable y los 4 bits más altos de LCD, con esto es suficiente para enviar los mensajes. El compilador PBP soporta módulos LCD con controlador Hitachi 44780 o equivalentes, se conecta el LCD al PIC16F877A de la siguiente manera, en B3 para el registro R/S, en B2 el Enable, en el puerto B desde el B4 hasta el B7, los bits más altos del LCD, también se utiliza una resistencia de  $10\Omega$  conectado a la alimentación del backlight, sirve para evitar altas temperaturas, además el R/W del LCD se encuentra conectada a tierra, esto de debe porque la declaración de LCDOUT, es de escritura solamente.

---

<sup>7</sup> Reyes Carlos, Aprenda rápidamente a programar microcontroladores, Pág. 79.-80.

<sup>16</sup> <http://www.x-robotics.com/lcd.htm>

### 1.4.6. Declaración de instrucciones para el LCD en PBP <sup>7</sup>

La declaración LCDOUT sirve para mostrar items en una pantalla de cristal líquido, se utiliza escribiendo: LCDOUT seguido por \$fe, esto se lo escribe siempre, junto a esto el comando a utilizar. La *Tabla 1.5* muestra los comandos para la programación del PIC 16F877A en el compilador PBP.

<b>Comando</b>	<b>Operación</b>
\$FE,1	Limpia el visor del LCD
\$FE,2	Vuelve al inicio (comienzo de la primera línea)
\$FE,\$0C	Apagar el cursor
\$FE,\$0E	Subrayado del cursor activo
\$FE,\$0F	Parpadeo del cursor activo
\$FE,\$10	Mover el cursor una posición a la izquierda
\$FE,\$14	Mover el cursor una posición a la derecha
\$FE,\$80	Mover el cursor al comienzo de la primera línea
\$FE,\$C0	Mover el cursor al comienzo de la segunda línea
\$FE,\$94	Mover el cursor al comienzo de la tercera línea
\$FE,\$D4	Mover el cursor al comienzo de la cuarta línea

**Tabla 1.5 Los comandos más utilizados para manejar el LCD. <sup>7</sup>**

## 1.5. EL AMPLIFICADOR OPERACIONAL (AOP) <sup>17</sup>

### 1.5.1. Introducción

Un amplificador operacional es un circuito integrado de extrema eficacia y amplia aplicación, por lo general en los diferentes sistemas electrónicos analógicos, se emplea el amplificador operacional, con él podremos amplificar señales, atenuarlas, filtrarlas, etc.

<sup>7</sup> Reyes Carlos, Aprenda rápidamente a programar microcontroladores, Pág. 79.-80.

<sup>17</sup> [http://html.wikipedia.com/amplificadores-operacionales\\_4.html](http://html.wikipedia.com/amplificadores-operacionales_4.html).

El concepto original del AOP (amplificador operacional) procede del campo de los computadores analógicos, en los que comenzaron a usarse técnicas operacionales en una época tan temprana como en los años 40.

El nombre de amplificador operacional deriva del concepto de un amplificador DC (amplificador acoplado en continua) con una entrada diferencial y ganancia extremadamente alta, cuyas características de operación estaban determinadas por los elementos de realimentación utilizados. Además, el mismo amplificador era capaz de realizar diversas operaciones, y el desarrollo gradual de los amplificadores operacionales dio lugar al surgimiento de una nueva era en los conceptos de diseño de circuitos.

#### **1.5.1.1. Definición <sup>18</sup>**

Un amplificador operacional o AOP, es un circuito que recibe una señal en su entrada y produce una versión más grande sin distorsión de la señal recibida en su salida, es decir, tiene una alta ganancia ya que posee una alta impedancia de entrada y baja impedancia de salida. Por lo general el amplificador operacional se utiliza para proporcionar cambios en la amplitud de voltaje (amplitud y polaridad).

#### **1.5.2. El modelo ideal de un AOP <sup>18</sup>**

Un amplificador operacional es un dispositivo electrónico activo siendo capaz de ofrecer una tensión de salida en función de una tensión de entrada.

Vamos a considerar única y exclusivamente el amplificador operacional ideal, que a pesar de no existir en la vida real, es una aproximación muy precisa y perfectamente válida para el análisis de sistemas reales. A continuación la descripción de las características de un amplificador operacional.

- Tiempo de conmutación nulo.
- Ancho de banda infinito.

---

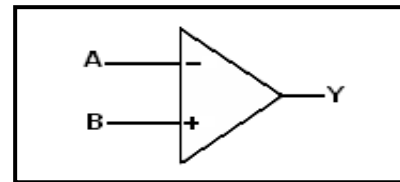
<sup>18</sup> BOYLESTAD, Robertl, Nashelsky Louis, Electrónica: Teoría de Circuitos y Dispositivos Electrónicos, Editorial Pearson Educación de Mexico, S.A, Págs. 686-687.

- Ganancia de tensión infinita.
- Impedancia de entrada infinita.
- Impedancia de salida nula.
- Corrientes de polarización nulas.
- Tensión de desplazamiento nula (la diferencia de potencial entre las entradas inversora y no inversora nula).
- Margen dinámico  $\pm V_{cc}$  (la tensión de salida puede a nivel teórico alcanzar el valor de la tensión de alimentación, en la práctica se aproxima pero no puede ser igual ya que se producen saturaciones en el dispositivo).

### 1.5.3. Simbología del AOP <sup>18</sup>

En la *Figura 1.30* se puede ver el símbolo más utilizado para representar a un AOP

- A: entrada inversora.
- B: entrada no inversora.
- Y: salida.



*Figura 1.30 Amplificador Operacional.* <sup>18</sup>

### 1.5.4. Circuitos lineales básicos con AOP <sup>18</sup>

#### 1.5.4.1. Amplificador no inversor

En la conexión de la *Figura 1.31a* se muestra un circuito amplificador no inversor su característica más destacable es su capacidad para mantener la fase de la señal, es decir no presenta desfase alguno. El análisis se realiza de la siguiente forma para que el lector de este documento lo entienda de una forma fácil.

Para determinar la ganancia de voltaje del circuito, se puede utilizar la representación equivalente que se muestra en la *Figura 1.31b*. Se puede observar

<sup>18</sup> BOYLESTAD, Robertl, Nashelsky Louis, *Electrónica: Teoría de Circuitos y Dispositivos Electrónicos*, Editorial Pearson Educación de Mexico, S.A, Págs. 686-687.



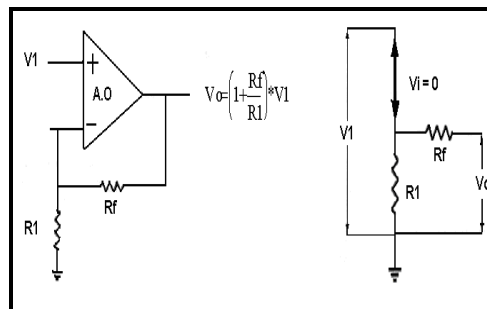
que el voltaje a través de  $R_1$  es  $V_1$  dado que  $V_i \approx 0V$ . Este debe ser igual al voltaje de salida, a través de un divisor de voltaje de  $R_1$  y  $R_f$  de forma que:

$$V_1 = \frac{R_1}{R_1 + R_f} V_o$$

La cual da por resultado:

$$A_v = \frac{V_o}{V_{in}} = \frac{R_1 + R_f}{R_1} = 1 + \frac{R_f}{R_1}$$

Se puede apreciar como no existe signo negativo en la expresión (no se invierte la señal), siendo además la ganancia siempre superior a la unidad. Este circuito no permite por consiguiente atenuar señales.



**Figura 1.31a Amplificador no Inversor.**<sup>18</sup> **Figura 1.31b Representación equivalente.**<sup>18</sup>

#### 1.5.4.2. Circuito seguidor<sup>18</sup>

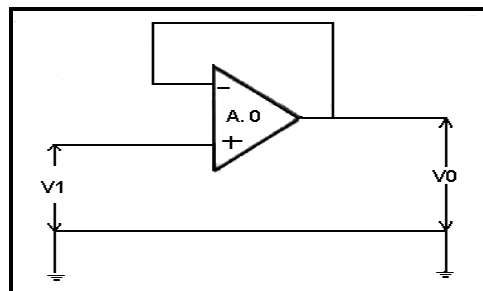
Esta sencilla configuración ofrece una tensión de salida igual a la tensión de entrada, no produciéndose ganancia alguna, es decir este circuito presenta una impedancia de entrada muy alta al mismo tiempo obteniendo, una impedancia de salida muy pequeña.

<sup>18</sup> BOYLESTAD, Robertl, Nashelsky Louis, Electrónica: Teoría de Circuitos y Dispositivos Electrónicos, Editorial Pearson Educación de Mexico, S.A, Pág. 686.

El circuito seguidor, como el que aparece en la *Figura 1.32*, proporciona una ganancia unitaria (1) sin inversión de polaridad o fase a partir del circuito equivalente ver la figura 1.27, de forma que:

$$V_0 = V_1$$

Y que la salida tiene la misma polaridad y magnitud que la entrada.



*Figura 1.32 Seguidor Unitario.*<sup>18</sup>

### 1.5.5. Circuitos no lineales básicos con AOP<sup>20</sup>

Los circuitos no lineales son los comparadores, conformadores de onda y circuitos con diodos activos. La salida de un circuito no lineal con amplificador operacional normalmente tiene una forma diferente que la señal de entrada, porque el amplificador operacional se satura durante parte del ciclo de entrada. Por esta razón, se tiene que analizar dos modos de funcionamiento distintos para ver qué ocurre durante un ciclo completo.

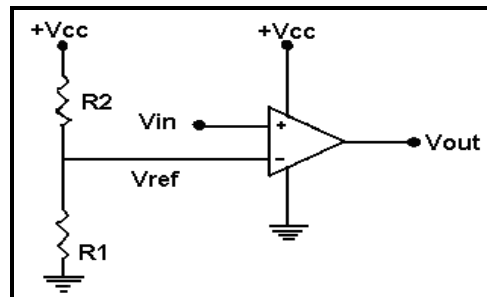
#### 1.5.5.1. Comparadores<sup>18</sup>:

El comparador es un amplificador operacional en lazo abierto (sin realimentación entre su salida y su entrada) y suele usarse para comparar una tensión variable con otra tensión fija que se utiliza como referencia. Como se indica en la *Figura 1.33*.

<sup>18</sup> BOYLESTAD, Robertl, Nashelsky Louis, *Electrónica: Teoría de Circuitos y Dispositivos Electrónicos*, Editorial Pearson Educación de Mexico, S.A, Pág. 686-687.

<sup>20</sup> W, García López, J.L. Gutiérrez Iglesias, *Amplificadores Operacionales*, Editorial ALCO, artes gráficas, Jaspe.34 -28026-MADRID , Pág. 796.

Como todo amplificador operacional, un comparador estará alimentado por dos fuentes de corriente continua ( $+V_{cc}$ ,  $-V_{cc}$ ). El comparador hace que, si la tensión de la entrada no inversora ( $V_{in}$ ) es mayor que la tensión en la entrada inversora ( $V_{ref}$ ), la salida ( $V_{out}$ ) será igual a  $+V_{cc}$ . En caso contrario, la salida tendrá una tensión  $-V_{cc}$ .



**Figura 1.33 Comparador.**<sup>18</sup>

Es decir:

Cuando la tensión de entrada  $V_{in}$  sea mayor al voltaje de referencia, dará un voltaje positivo y por lo tanto la tensión de salida estará a nivel alto (cercano a  $V_{cc}$ )

Cuando la tensión de entrada  $V_{in}$  sea menor al voltaje de referencia, dará un voltaje negativo y por lo tanto la salida estará a nivel bajo (cercano a 0 voltios)

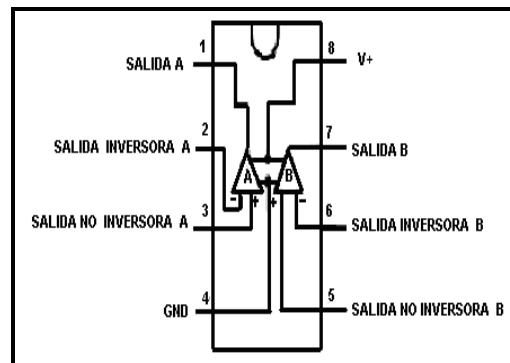
En la aplicación del proyecto hemos empleado el LM358, dando a conocer una breve descripción del mismo.

<sup>18</sup> BOYLESTAD, Robertl, Nashelsky Louis, *Electrónica: Teoría de Circuitos y Dispositivos Electrónicos*, Editorial Pearson Educación de Mexico, S.A, Pág. 686-687.

### 1.5.6. Circuito Integrado LM358 <sup>21</sup>

Es un circuito integrado que en su interior se encuentran dos amplificadores operacionales que trabaja desde 3v a 32v a 500 $\mu$ A, una particularidad de este circuito integrado, es que puede ser alimentado con una sola fuente positiva.

En algunos proyectos se necesita un amplificador donde no se cuenta con fuentes +V -V y este integrado es una buena opción. En la *Figura 1.34* se muestra la distribución de pines.



*Figura 1.34 Distribución de pines de LM358.* <sup>22</sup>

### 1.6. SENSORES <sup>23</sup>

Un sensor o captador, es un dispositivo diseñado para recibir información de una magnitud del exterior y transformarla en otra magnitud, normalmente eléctrica, que seamos capaces de cuantificar y manipular.

Normalmente estos dispositivos se encuentran realizados mediante la utilización de componentes pasivos (resistencias variables, PTC, LDR, etc. todos aquellos componentes que varían su magnitud en función de alguna variable), y la utilización de componentes activos.

A continuación se indican algunos tipos y ejemplos de sensores electrónicos:

<sup>21</sup> [http://www.unicrom.com/tut\\_comparadores.asp](http://www.unicrom.com/tut_comparadores.asp).

<sup>22</sup> [www.national.com/datasheet-lm358/](http://www.national.com/datasheet-lm358/). <http://es.wikipedia.org/wiki/Sensor>

<sup>23</sup> <http://es.wikipedia.org/wiki/Sensor>

- Sensores de temperatura: Termopar, Termistor, RTD.
- Sensores de deformación: Galga extensiométrica.
- Sensores de acidez: IsFET.
- Sensores de luz: fotodiodo, fotorresistencia, fototransistor.
- Sensores de sonido: micrófono.
- Sensores de imagen digital (fotografía): CCD o CMOS.
- Sensores de proximidad: detectores capacitivos inductivos y fotoeléctricos

Por lo general la señal de salida de estos sensores no es apta para su lectura directa y a veces tampoco para su procesado, por lo que se usa un circuito de acondicionamiento.

#### **1.6.1. Light Dependent Resistor (LDR) <sup>24</sup>**

La LDR es un componente electrónico que viene de la expresión inglesa Light Dependent Resistor la cual es una resistencia que varía su valor dependiendo de la cantidad de luz que la ilumina. Puede también ser llamado fotoresistor, fotoconductor, célula fotoeléctrica o resistor dependiente de la luz.

Desde el punto de vista constructivo, las LDR están fabricadas con materiales de estructura cristalina, siendo los más utilizados el sulfuro de cadmio y el seleniuro de cadmio, aprovechando sus propiedades fotoconductoras.

El sulfuro de cadmio, convenientemente tratado, contiene pocos o ningún electrón libre, si se mantiene en completa oscuridad. En estas condiciones, su resistencia es elevada. Si absorbe luz, se libera cierto número de electrones, y esto hace aumentar la conductividad del material, justificando la denominación de fotoconductor que recibe. Los electrones permanecen en libertad por un tiempo limitado, ya que al cesar la iluminación, son recapturados a sus posiciones originales y el material se convierte de nuevo en aislador.

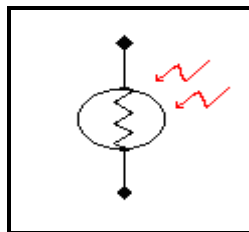
---

<sup>24</sup> <http://es.wikipedia.org/wiki/LDR>

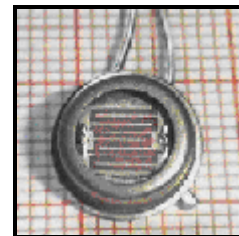
Una cuestión a tener en cuenta cuando diseñamos circuitos que usan LDR es que su valor (en Ohmios) no variará de forma instantánea cuando pasa de estar expuesta de la luz a la oscuridad, o viceversa, y el tiempo que dura este proceso no siempre es igual, si se pasa de oscuro a iluminado o si se pasa de iluminado a oscuro. Igualmente, estos tiempos son cortos, generalmente del orden de una décima de segundo.

Si bien los valores que puede tomar una LDR en total oscuridad y a plena luz puede variar un poco de un modelo a otro, en general oscilan entre unos  $50\Omega$  a  $1K\Omega$  cuando están iluminadas (por ejemplo, con luz solar) y valores comprendidos entre  $50K\Omega$  y varios megohmios cuando está a oscuras.

La *Figura 1.35a* nos muestra el símbolo utilizado para representar las LDR en los esquemas electrónicos y en la *Figura 1.35b* su representación física, aunque a veces pueden ser ligeramente diferentes pero siempre tomando como base el símbolo de una resistencia común con alguna(s) flecha(s) que simbolizan la incidencia de la luz. Los resistores LDR cuentan con gran número de aplicaciones en toda clase de circuitos para control, recuento, medida, alarma y conmutación.



**Figura 1.35a Símbolo de una LDR.** <sup>24</sup>



**Figura 1.35b LDR Físicamente.** <sup>24</sup>

<sup>24</sup> <http://es.wikipedia.org/wiki/LDR>



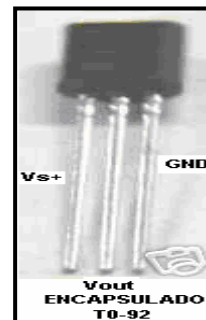
La baja impedancia de salida, su salida lineal y su precisa calibración inherente hace posible una fácil instalación en un circuito de control. Debido a su baja corriente de alimentación (60uA), se produce un efecto de autocalentamiento reducido, menos de 0.1 °C en situación de aire estacionario.

### Encapsulado <sup>24</sup>

El sensor LM35 se presenta en diferentes tipos de encapsulados, pero el más común es el T0-92 (siendo el más idóneo para nuestra aplicación), de igual forma que un típico transistor con 3 pines, dos de ellas para alimentarlo y la tercera nos entrega un valor de tensión proporcional a la temperatura medida por el dispositivo.

Como se muestra en la *Figura 1.37*, el LM35 viendo las letras del encapsulado hacia arriba tenemos de izquierda a derecha los pines:

- VCC
- Vout
- GND.



*Figura 1.37 Distribución de pines del LM35.* <sup>25</sup>

### Usos:

El sensor de temperatura puede usarse para compensar un dispositivo de medida sensible a la temperatura ambiente o refrigerar partes delicadas del robot

<sup>24</sup> <http://es.wikipedia.org/wiki/LDR>

<sup>25</sup> <http://cgi.ebay.es/sensor-control-de-temperatura-lm35-to92.>

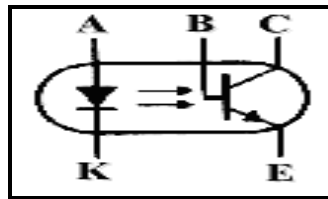


## 1.7. OPTO - ACOPLADOR <sup>27</sup>

Un opto-acoplador combina un dispositivo semiconductor formado por un fotoemisor, un fotoreceptor y entre ambos hay un camino por donde se transmite la luz.

### 1.7.1. Funcionamiento del Opto - acoplador

La señal de entrada se aplica al fotoemisor y la salida es tomada del fotoreceptor, los opto-acopladores son capaces de convertir una señal eléctrica en una señal luminosa modulada y volver a convertirla en una señal eléctrica. La gran ventaja de un opto-acoplador reside en el aislamiento eléctrico que puede establecerse entre los circuitos de entrada y salida. En la *Figura 1.38* se puede observar como está constituido internamente



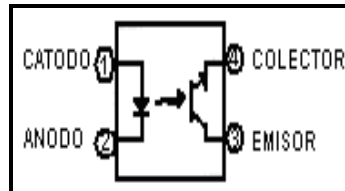
*Figura1.38 Constitución interna del Opto-acoplador.* <sup>27</sup>

Los fotoemisores que se emplean en los opto-acopladores son diodos que emiten rayos infrarrojos (IRED) y los fotoreceptores pueden ser tiristores, transistores o compuertas lógicas.

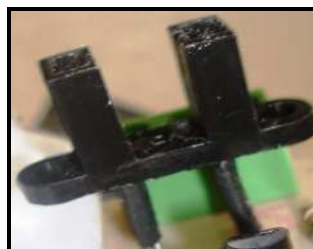
Cuando aparece una tensión sobre los terminales del diodo IRED, este emite un haz de rayos infrarrojo que se transmite a través de una pequeña guía-ondas de plástico o cristal hacia el fotoreceptor. La energía luminosa que incide sobre el fotoreceptor hace que este genere una tensión eléctrica a su salida. Este responde a las señales de entrada, que podrían ser pulsos de tensión.

<sup>27</sup> <http://www.uv.es/marinjl/electro/opto.html>

Hemos empleado un opto-acoplador, el cual podemos encontrarlo fácilmente en el mercado como es el **ECG 3100** su estructura interna se observa en la *Figura 1.35a* y en la *Figura 1.35b* su estructura externa.



*Figura 1.35a Estructura Interna.*<sup>28</sup>



*Figura 1.35b Estructura externa del ECG 3100.*<sup>12</sup>

<sup>12</sup> Creadores y Autores del Proyecto.

<sup>28</sup> ECG, Master Replacement Guide, Edition 19, Pág.1-160.

## **CAPÍTULO II: DISEÑO Y CONSTRUCCIÓN DEL SISTEMA DE AUTOMATIZACIÓN DE APERTURA Y CIERRE DE PERSIANAS VERTICALES**

### **2.1. INTRODUCCIÓN**

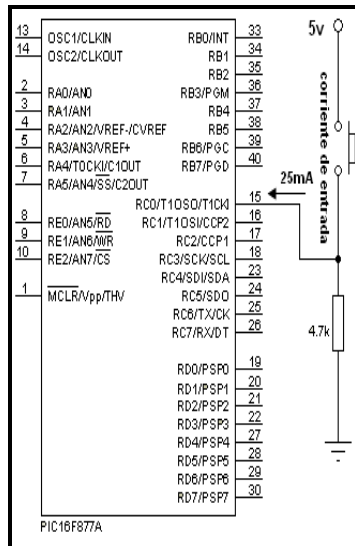
El sistema de automatización de apertura y cierre de persianas verticales es activado automáticamente, por medio de la claridad / oscuridad, temperatura o interactuando por medio de un panel de control constituido por teclas que realizan las siguientes funciones; cambiar los intervalos de temperatura mínima y máxima, cambiar la modalidad para que trabaje en función de claridad / oscuridad, abrir o cerrar las persianas verticales (desplazarlas horizontalmente), colocar las persianas en la posición frontal o lateral y además se lo puede emplear como un método de seguridad, ya que nos otorga una simulación de presencia.

### **2.2. CONSIDERACIONES BÁSICAS A LA HORA DE MONTAR UN PROYECTO**

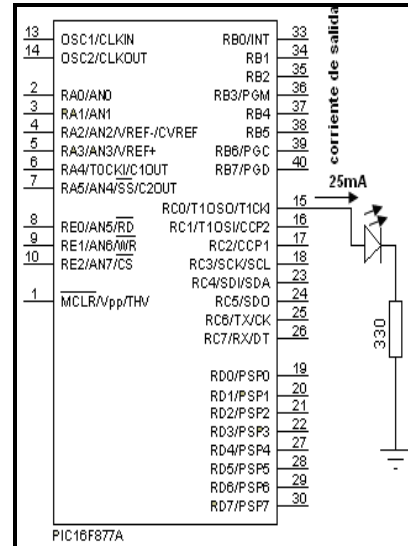
Es muy importante tomar en cuentas estas recomendaciones ya que si no se las sigue podría correr el riesgo de dañar el PIC:

1. El PIC 16F877A es de tecnología CMOS, esto quiere decir que consume muy poca corriente pero a la vez es muy susceptible a daños por estática, se recomienda utilizar pinzas o a su vez utilizar una manilla antiestática y así poder transportar desde el grabador al protoboard y viceversa.
2. Se emplea un regulador de voltaje como el L7805C-V que entrega exactamente 5VDC, y no emplear algún cargador de batería ya sea de celular o de algún otro dispositivo, ya que el voltaje de salida no siempre es el mismo que especifica sus características técnicas.

3. No se debe sobrepasar los niveles de corriente, tanto de entrada como de salida, el PIC puede entregar por cada uno de sus pines una corriente máxima de 25 mA. Así mismo soporta una corriente máxima de entrada de 25mA. Como se ven en las *Figuras 2.1a* y *2.1b*.



**Figura 2.1a** Corriente de entrada.



**Figura 2.1b** Corriente de salida.<sup>12</sup>

Para encender un led con la resistencia adecuada se debe considerar los siguientes valores:

El voltaje de salida del PIC, si es alimentado con 5VDC= 5VDC out.

Corriente que requiere un led par un encendido normal= 15 mA.

$$V = R \times I$$

$$R = \frac{V}{I}$$

$$R = \frac{5V}{0,015A}$$

$$R = 333,33\Omega = 330\Omega$$

Para encender un led sin correr el riesgo de dañarlo se necesitará colocar una resistencia de 330Ω.

<sup>12</sup> Creadores y Autores del Proyecto.

Para el correcto funcionamiento de la tecla se debe tomar en cuenta la siguiente condición:

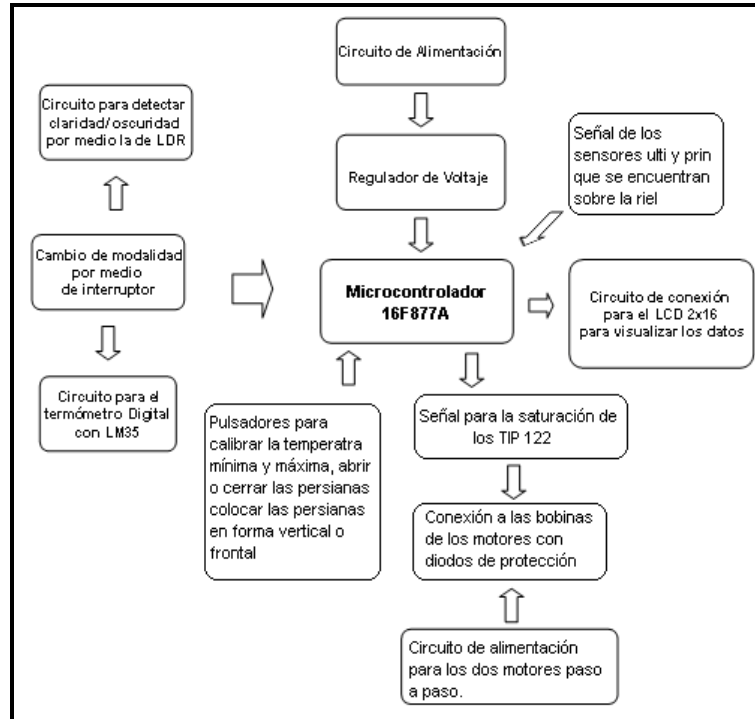
El diagrama de la *Figura 2.1a* es un circuito básico para generar pulsos (1L y 0L) de entrada al PIC, que consta de una tecla y una resistencia la cual se coloca para que no produzca un corto circuito al momento de pulsar la tecla.

La resistencia mínima a colocarse es de  $220\Omega$  para estar al límite de la capacidad que soporta el PIC, pero no es muy aconsejable colocar la resistencia justa, por lo que se recomienda utilizar una resistencia de  $1k\Omega$  a  $10k\Omega$ , así el PIC estaría trabajando tranquilamente con una corriente de entrada de  $5mA$  o  $0,5mA$  respectivamente. Pero en la aplicación hemos trabajado con resistencias de  $4,7K\Omega$  con una corriente de  $0,00106$  Amperios o  $1,06 mA$ .

4. Cuando se necesite precisión en el trabajo del PIC 16F877A, se recomienda utilizar un oscilador externo de  $4MHz$  (de configuración HS) debido a que estos cristales son muy precisos en cuanto a la frecuencia que entregan, ya que el oscilador interno RC no posee una buena precisión.

### **2.3. DESCRIPCIÓN DEL HARDWARE UTILIZADO.**

El sistema de automatización de apertura y cierre de persianas verticales se compone de varias etapas ilustradas en la *Figura 2.1*, formadas por: circuitos integrados (dos LM358) , un regulador de voltaje L7805C-V, un microcontrolador 16F877A, elementos discretos como dos capacitores de cerámica de  $22pF$  y un electrolítico de  $22\mu$ , 18 resistencias de  $4.7K\Omega$  dos de  $330\Omega$  una de  $3.9K\Omega$  tres de  $10K\Omega$ , dos potenciómetros tipo araña uno de  $10K\Omega$  y el otro de  $5K\Omega$ , dos optoacopladores ECG 3100, ocho diodos 1N4007, ocho TIP 122, un led, un sensor LM35, una LDR de  $5mm$  de radio, ocho teclas de dos pines, un interruptor y un oscilador de cuarzo de  $4MHz$  que complementan el Hardware.



**Figura 2. Diagrama de bloques.** <sup>12</sup>

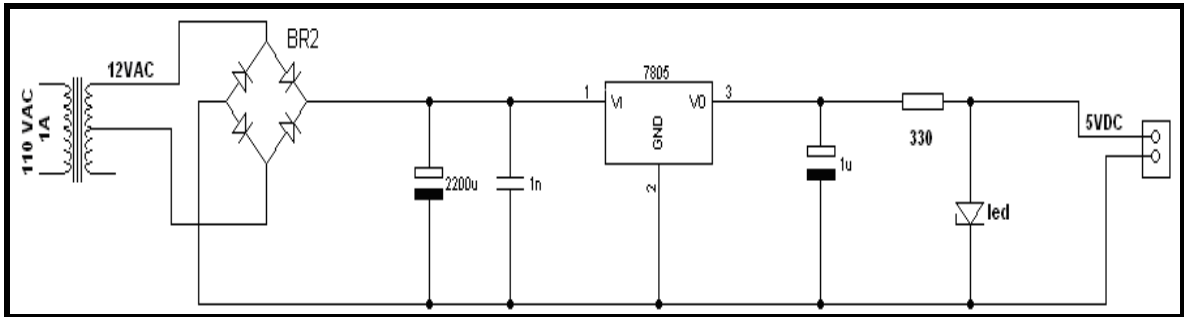
A continuación se detalla cada una de las etapas que constituyen el sistema de automatización de apertura y cierre de persianas verticales.

### 2.3.1. Circuito de alimentación

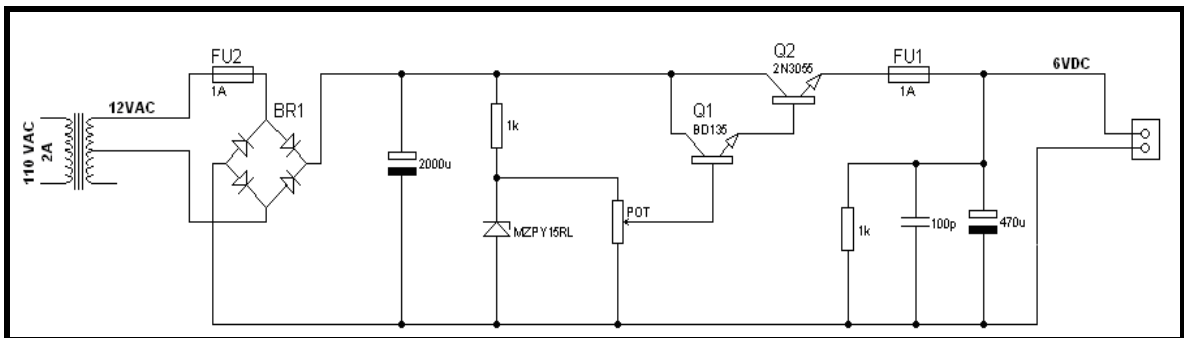
Para la alimentación del microcontrolador y demás elementos del circuito se ha implementado una fuente de voltaje de 5VDC fijos a 1Amperio como máxima corriente que nos entregará el transformador, suficiente para el funcionamiento del circuito, se emplea un regulador de voltaje L7805C-V que nos proporciona 5VDC exactos (Figura 2.3a).

Para el funcionamiento de los motores paso a paso unipolares de 5 hilos se necesita otra fuente de alimentación variable de 0VDC a 12VDC a 2 Amperios, a dicha fuente le fijaremos el valor a 6VDC a 2 Amperios (Figura 2.3b) debido que a plena carga trabaja con 1,3 Amperios.

<sup>12</sup> Creadores y Autores del Proyecto.



**Figura 2.3a Fuente de voltaje para el PIC con sus elementos.**

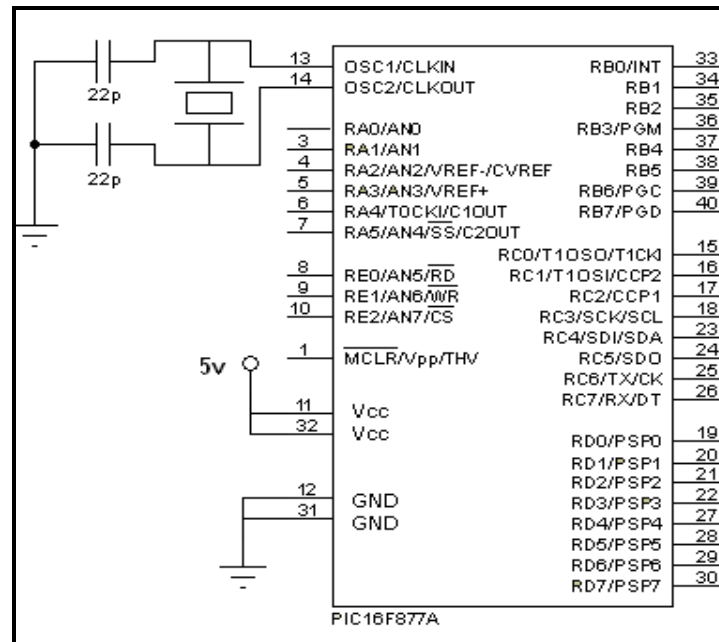


**Figura 2.3b Fuente de alimentación variable para los motores paso a paso.**

### 2.3.2. Conexión del microcontrolador

Para la alimentación de voltaje del microcontrolador se empleó un regulador L7805C-V, (en la salida siempre tendremos 5VDC), para establecer la velocidad de ejecución de las instrucciones, el microcontrolador, utiliza un oscilador externo HS que es un cristal de cuarzo de 4 Mhz.

En la *Figura 2.4*, se observa la conexión básica de los elementos utilizados para el funcionamiento del microcontrolador.



**Figura 2.4** Circuito de conexión básica del Microcontrolador 16F877A.

### 2.3.3. Termómetro digital

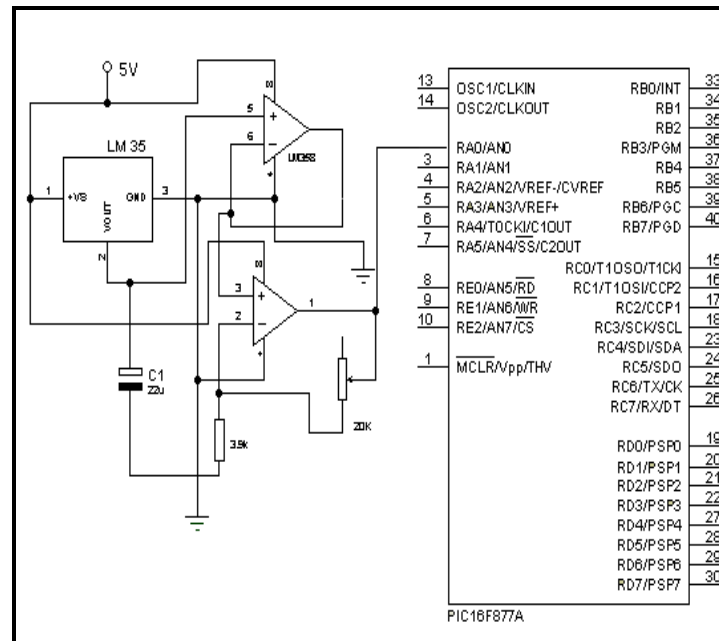
Empleamos un sensor de temperatura LM35, el cual en su pin de out presenta una variación de 10mV por grado centígrado, trabaja en un rango de temperatura de -5°C hasta 150°C.

El voltaje que nos proporciona el LM35, lo duplicamos por medio de un amplificador operacional (LM358), que está configurado como amplificador no inversor, el voltaje duplicado lo conectamos al puerto A.0 que está programado como un conversor A/D de 8bits, este valor se almacena en la variable llamada dato, el cual lo dividimos para dos, con el fin de estabilizar la señal que ingresa del pin out de LM35.

Se ha programado el PIC de tal manera que posea dos rangos de temperatura un máximo y un mínimo que están grabados directamente en la EEPROM del PIC, estos rangos son programables a gusto del usuario las veces que estime necesario. Si la temperatura actual del ambiente sobrepasa cualquiera de los dos rangos, el PIC emite una orden a las salidas del puerto C, las cuales están conectadas a cada una de las bobinas del motor paso a paso (para abrir o cerrar).



La conexión del sensor LM35 se indica en al *Figura 2.5*.



**Figura 2.5 Circuito del termómetro digital.**

### 2.3.4. Teclas e Interruptores

Las teclas son la interfase que utiliza el operador para poder calibrar y ajustar en este caso el rango de la temperatura que más se ajuste a su conveniencia, avance y retroceso del motor y cambio de modalidad de funcionamiento.

La descripción para la manipulación de las teclas es la siguiente:

Si se desea modificar los rangos de temperatura, se presiona la tecla enter y el led parpadeará una vez, en el LCD aparecerá programar temperatura baja y con la tecla subir se aumenta la temperatura mínima en una unidad por cada pulso y de igual manera si se presiona la tecla bajar se disminuye la temperatura mínima en una unidad por cada pulso. Al momento de estar de acuerdo con la temperatura mínima se presiona una vez la tecla enter y el led parpadeará una vez indicándonos que se ha grabado en la EEPROM, acto seguido el mensaje del LCD cambiará a programar temperatura alta y el procedimiento es idéntico que ajustar la temperatura mínima cuando se este de acuerdo con los valores de

temperatura programados se presiona la tecla enter y el led parpadeará tres veces indicándonos que se ha grabado en la EEPROM y el LCD vuelve a su mensaje principal.

Las teclas abrir y cerrar serán utilizados cuando el motor no esté en funcionamiento para abrir o cerrar la persiana (recorrer izquierda a derecha o viceversa) una determinada distancia por cada pulsación que se ejecute en las mismas.

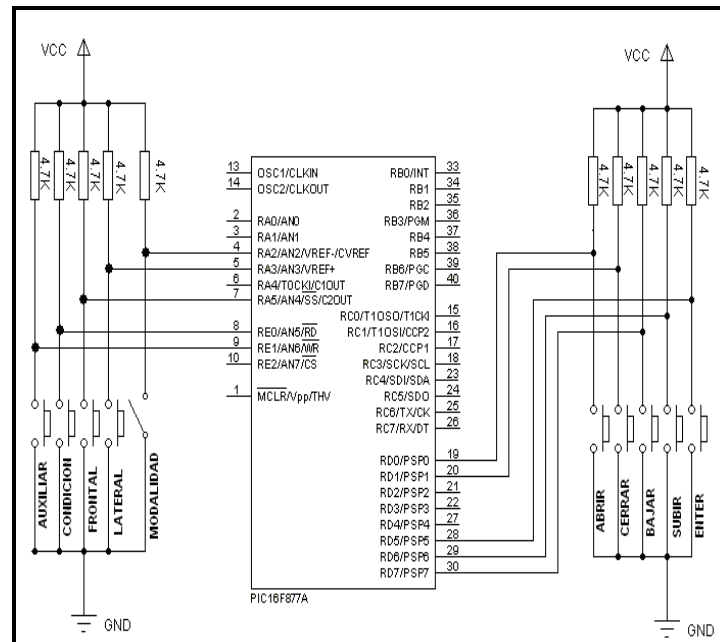
Las teclas frontal y lateral son las encargadas de hacer girar al segundo motor paso a paso unipolar de 5 hilos, para que así coloque a las persianas en posición lateral para el desplazamiento izquierda derecha o viceversa.

Hay que recalcar que las teclas enter, bajar, subir, abrir y cerrar solo trabajan en modo temperatura y en modo claridad / oscuridad no desempeñan ninguna función. Mientras que las teclas lateral y frontal funcionan en ambas modalidades.

La función que desempeña el interruptor es el controlar la modalidad de apertura y cierre de las persianas verticales, de modo temperatura a modo claridad / oscuridad o viceversa.

La finalidad del micro-switch llamado bcondi en el programa, es el de censar cuando las persianas se encuentran completamente en forma lateral para así proceder con la apertura o cierre de la persiana.

Este micro-switch es muy importante ya que si las persianas se desplazan de izquierda a derecha o viceversa sin que estén en forma lateral, el riel o una parte de él se dañe y por ende la persiana ya no tendrá la misma facilidad para abrirse o cerrarse. El micro-switch es el que envía la orden al PIC para que el motor paso a paso deje de moverse. El circuito de conexión de las teclas se muestra en la *Figura 2.6*.



**Figura 2.6 Conexión de las teclas.**

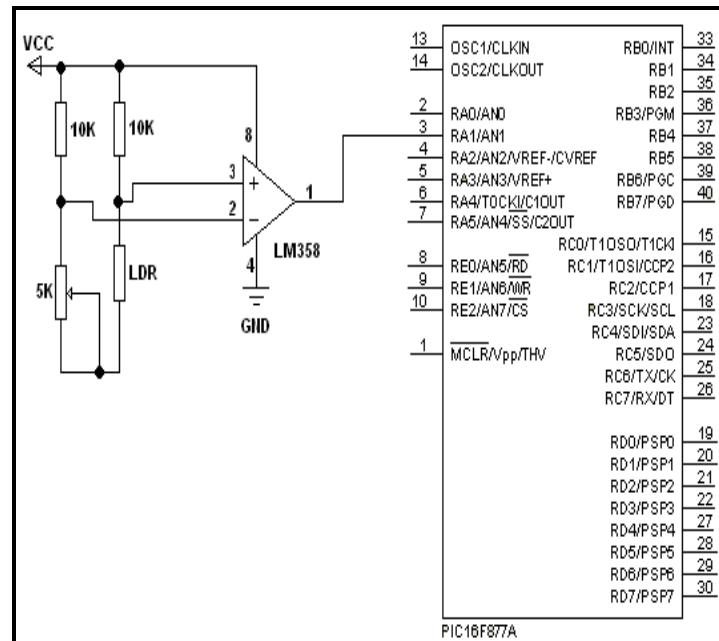
### 2.3.5. Funcionamiento del circuito sensor de iluminación

La LDR (light depending resistor) es el sensor que detecta el nivel de luz, en realidad es una resistencia que varía su valor dependiendo a la cantidad de luz que incide en ella, su comportamiento es el siguiente: a mayor iluminación menor resistencia, y a menor iluminación mayor resistencia. Se debe evitar que la LDR no reciba directamente otro tipo de luz que no sea la luz proveniente del medio ambiente, ya que se trabaja en base al grado de iluminación del ambiente.

Está diseñado con un amplificador operacional LM358 que funciona como comparador entre la tensión variable que proviene de la LDR y una tensión de referencia ajustable mediante el potenciómetro de 5KΩ.

El potenciómetro de 5KΩ sirve para ajustar la sensibilidad del circuito, así regulamos en qué momento debe girar el motor para abrir o cerrar la persiana vertical. Como se indica en la *Figura 2.7*.

El comportamiento de la persiana cuando se encuentra en modalidad claridad / oscuridad es el siguiente: al instante en que el voltaje que emita la LDR sobrepase o disminuya el valor de voltaje fijo que nos proporciona el potenciómetro, la persiana se abrirá o se cerrará por completo y nunca tendremos la persiana a medio abrir.



**Figura 2.7** Circuito de funcionamiento del sensor de iluminación.

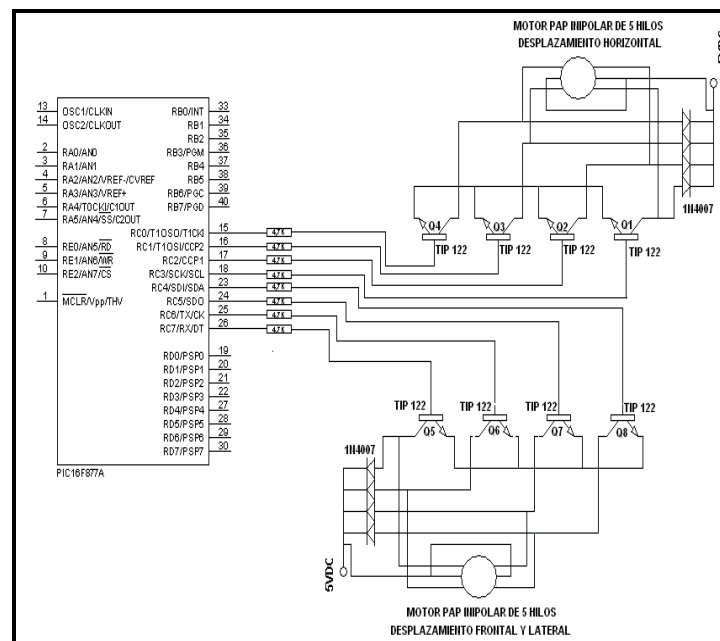
### 2.3.6. Motor paso a paso unipolar de 5 hilos

El motor paso a paso unipolar de 5 hilos, funciona con una fuente de alimentación variable muy sencilla, porque proporciona tensiones de 0VDC a 12VDC y corrientes hasta 2 Amp, para ello se debe tomar en cuenta estas recomendaciones:

- El transistor Q2 (transistor de potencia 2N3055) debe estar colocado sobre un buen disipador térmico.
- Para evitar las sobrecargas se debe colocar un fusible de 2 Amperios en serie con el emisor del transistor Q2.

Para proteger al PIC del efecto inductivo que genera el motor, se ha colocado diodos de protección 1N4007, cuyos ánodos están conectados al colector de cada TIP122 los cuales funcionan como interruptores y también como amplificadores de corriente. Cada base de los TIP 122 está conectada a una resistencia de 4,7k $\Omega$  la misma que protege al PIC, mientras que el emisor está conectado a tierra y por último cada colector va conectado a cada una de las bobinas del motor paso a paso.

El funcionamiento es sencillo, cuando el PIC envíe 1L a cualquiera de los pines del puerto C estos saturan a cada uno de los 8 transistores, activando de manera secuencial cada una de las bobinas (A, B, C, D) de los motores paso a paso, generando así, la apertura o cierre de la persiana vertical (*Figura 2.8*).

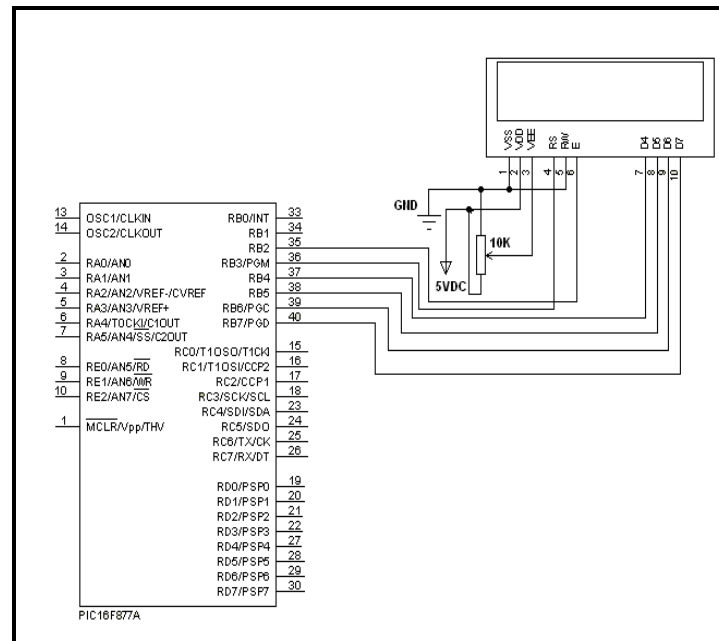


**Figura 2.8** Circuito del motor paso a paso unipolar de 5 hilos a las salidas del PIC 16F877A.

### 2.3.7. Pantalla de Cristal Líquido (LCD)

Para la comunicación entre el LCD y el PIC 16F877A, se empleó un bus de 4 bits, su funcionamiento se realiza enviando primero los 4 bits más altos y luego los 4 bits más bajos (*Figura 2.9*) de esa manera nos evitamos el uso de 4 pines del PIC y de 4 cables extras y su configuración es la siguiente:

- B3 para el registro R/S.
- B2 Enable.
- En el puerto B desde el B4 hasta el B7, los bits más altos del LCD.
- R/W del LCD se encuentra conectada a tierra, esto se debe a que la declaración de LCDOUT, es de escritura solamente.



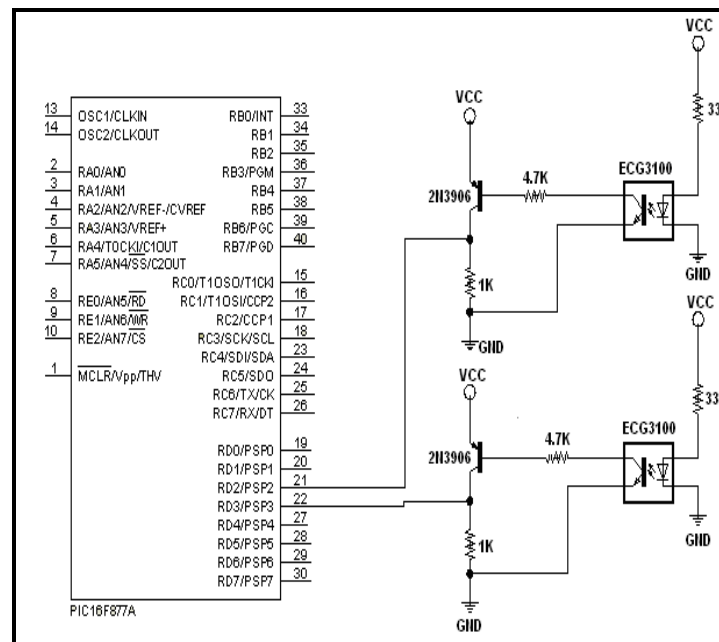
**Figura 2.9** Circuito de conexión básica para el LCD 2x16.

### 2.3.8. Opto - acopladores

Los opto-acopladores son muy útiles para el proyecto y utilizaremos dos (ulti y prin llamados así en el programa). Son los encargados de informar al PIC que la persiana ya está cerrada o abierta en su totalidad impidiendo así que se ejecute nuevamente la instrucción del programa de abrir o cerrar la persiana y también de cortar el movimiento tanto del motor de desplazamiento horizontal como condicionar el movimiento del motor de desplazamiento frontal y lateral (Figura 2.10).

Su funcionamiento es el siguiente:

- Cuando el obstáculo que está sujeto en el último seguro donde ingresa la tira de la persiana obstaculiza a cualquiera de los opto-acopladores (ulti o prin) éste pasa del estado de saturación al estado de corte dando como resultado que el transistor PNP 2N3906 se sature y el pin donde se encuentra conectado (D2 o D3) cambie de estado provocando así que el motor se detenga.
- Una vez que el motor se haya detenido el opto-acoplador queda obstaculizado, para explicar mejor haremos cuenta que la persiana se ha cerrado por completo y el opto-acoplador se queda obstaculizado, esto le indica al PIC 16F877A que la persiana se encuentra cerrada y no mande la orden de continuar cerrando en una próxima ocasión.



**Figura 2.10** Circuito de conexión de los opto-acopladores ulti y prin

## 2.4. DESARROLLO DEL SOFTWARE

Antes de iniciar el desarrollo del Software se debe tener claro cada una de las actividades que el sistema va a realizar.

### 2.4.1. Lenguaje de programación

Para la elaboración del software del microcontrolador se utilizó el compilador PIC Basic Pro (PBP), es un lenguaje de programación de alto nivel debido a que es fácil y rápido para la programación de microcontroladores. El lenguaje Pic Basic Pro es mucho más fácil de leer y escribir que el lenguaje ensamblador.

La diferencia entre el lenguaje de Pic Basic Pro y el ensamblador, es que uno es un lenguaje de alto nivel y el otro de bajo nivel.

**2.4.1.1. Lenguajes de bajo nivel**<sup>29</sup>: se llaman de bajo nivel porque están muy cercanos al Hardware del ordenador. Este es el primer lenguaje que se utilizó para el lenguaje máquina, que consiste en un conjunto de instrucciones en binario, con ceros y unos, con los cuales se indica al ordenador qué hacer.

Este lenguaje es muy complicado y la posibilidad de cometer errores es muy alta, para solventar estas dificultades apareció el lenguaje ensamblador, que consiste en asignar una abreviatura a cada instrucción en binario, de forma que sea más fácil recordarla y menos probable equivocarse. Sin embargo, con este lenguaje sigue siendo necesario conocer muy bien el Hardware del ordenador.

**2.4.1.2. Lenguajes de alto nivel**: este lenguaje se encuentran más cercanos al lenguaje natural (lenguaje humano) que al lenguaje máquina. Se trata de lenguajes independientes de la arquitectura del ordenador. Por lo que, en principio, un programa escrito en un lenguaje de alto nivel, lo puedes enviar de una máquina a otra sin ningún tipo de problema.

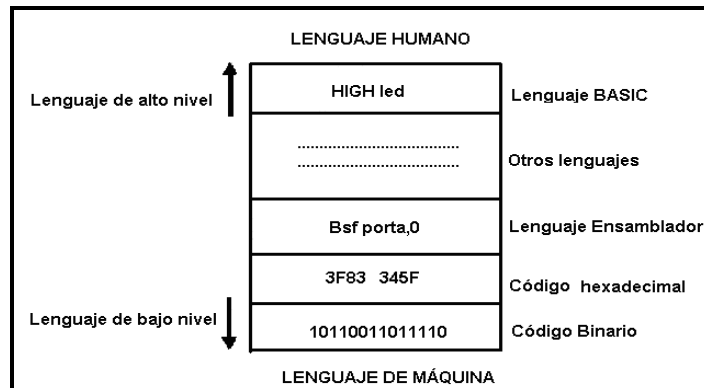
Estos lenguajes permiten al programador olvidarse por completo del funcionamiento interno de las máquinas para la que están diseñando el programa. Tan solo necesitan un traductor que entiendan el código fuente como las características de la máquina.

En la *Figura 2.11*, se puede observar la diferencia entre los dos tipos de lenguaje.

---

<sup>29</sup> <http://www.desarrolloweb.com/articulos/2358.php>.





**Figura 2.11** Cuadro de los niveles de programación el lenguaje que más se acerca a los humanos es el de más alto nivel, el lenguaje más próximo al tipo de datos que entiende el microcontroladores es un lenguaje de bajo nivel.<sup>7</sup>

## 2.5. LENGUAJE DE PROGRAMACIÓN PBP<sup>30</sup>

Se utilizó el programador PBP (Pic Basic Pro) que tiene muchas librerías y funciones como un compilador real por lo que los programas se ejecutan mucho más rápido.

El PBP tiene una variedad de instrucciones utilizadas para programar microcontroladores. A continuación se describen las instrucciones utilizadas en el desarrollo del programa para el sistema de automatización de apertura y cierre de persianas verticales.

**DEFINE:** Algunos elementos, como el oscilador y las ubicaciones de los pines del LCD, están predefinidos en PBP. DEFINE, le permite al programa PBP cambiar estas definiciones si así se desea.

**Ejemplo:**

```
DEFINE LCD_DREG PORT B
DEFINE LCD_DBIT 4
```

La primera instrucción define los pines del LCD desde el pín B4 hasta B7, la segunda instrucción nos indica que va a empezar desde el pín B4 hasta el B7.

<sup>7</sup> Reyes Carlos, Aprende rápidamente a programar microcontroladores, Pág. 33.

<sup>30</sup> [http://www.todopic.com.ar/pbp\\_sp.html](http://www.todopic.com.ar/pbp_sp.html)

**IDENTIFICADORES:** Un identificador es simplemente un nombre, son usados en PBP como etiquetas de líneas y nombres de variables. Un identificador es cualquier secuencia de letras, dígitos y símbolos, aunque no deben comenzar con un dígito.

Los identificadores no distinguen las letras mayúsculas de las minúsculas, por lo que la etiqueta se puede escribir de las siguientes formas: etiqueta, ETIQUETA, Etiqueta estos ejemplos son tratados como equivalentes. Aunque las etiquetas pueden tener cualquier número de caracteres de longitud PBP solamente reconoce los primeros 32.

**VARIABLES:** Es donde se guardan datos en forma temporal en el programa PBP, son creadas usando la palabra clave **VAR**, pueden ser bits, bytes o word. El formato para crear una variable es el siguiente:

Etiqueta **VAR** tamaño

Etiqueta es cualquier identificador. Sus tamaños son: bit (valores 0 o 1), byte (enteros de 0 a 255) o word (enteros de 0 a 65535). Crean una variable y le asignan un tamaño ya sea BIT, BYTE O WORD.

**Ejemplo:** x **VAR BYTE**

**PINES:** A los pines se puede acceder de diferentes modos, el mejor camino para especificar un pin para una operación, es usando simplemente sus nombres PORT (A o B) y un número de bit.

**Ejemplo:** PORTD.4=1

Esta instrucción indica, que al pín D4 debe sacar 1L.

Para recordar fácilmente el uso de un pin, puede asignársele un nombre usando el comando **VAR**, de esta manera, el nombre puede ser utilizado luego en cualquier operación.

**Ejemplo:** Led VAR PORTD.4

**ETIQUETAS DE LÍNEA (LABELS):** Para marcar líneas que el programa puede desear referenciar con comandos como el GOTO, PBP usa etiquetas de línea. Cualquier línea PBP puede comenzar con una etiqueta de línea que es simplemente un identificador seguido por dos puntos (:).

**Ejemplo:** sensor:

**PAUSE** 1000

**GOTO** sensor

**HIGH:** Coloca el Pin especificado en valor alto (5 voltios) y lo convierte automáticamente en salida.

**Ejemplo:** led VAR PORTD.4

**HIGH** led

**LOW:** Coloca el pin especificado en valor bajo (0 voltios) y automáticamente lo convierte en salida.

**Ejemplo:** Led VAR PORTD.4

**LOW** led

**GOTO:** La ejecución del programa continúa en la declaración de la etiqueta.

**Ejemplo:** **GOTO** sensor

sensor:

**HIGH** led

**PAUSE:** Detiene el programa por el tiempo señalado en milisegundos por "Periodo". El periodo tiene 16 bits, por lo que los retardos pueden ser de hasta 65.535 milisegundos. No coloca al microcontrolador en modo de baja potencia,

inclusive, consume mayor potencia, pero es muy exacto ya que tiene la misma precisión que el reloj.

**Ejemplo: PAUSE 1000**

Este pause de 1000 nos indica que el programa, espera un segundo para ejecutar la siguiente instrucción.

**IF...THEN** puede operar de dos maneras:

De una forma, el **THEN** en un **IF...THEN** es esencialmente un GOTO. Si la condición es cierta, el programa irá hacia la etiqueta que sigue al **THEN**, si la condición es falsa, el programa va a continuar hacia la próxima línea después del **IF...THEN**. Otra declaración no puede ser puesta después del **THEN**, sino que debe ser una etiqueta. Todas las comparaciones no tienen signo, ya que PBP solo soporta operaciones sin signo.

**Ejemplo: IF PORTD.4=1 THEN espere**

En la segunda forma, **IF...THEN** puede ejecutar condicionalmente un grupo de declaraciones que sigan al **THEN**. Las declaraciones deben estar seguidas por un **ELSE** o un **ENDIF** para completar la estructura.

```
Ejemplo: IF PORTD.4=1 THEN  
           PAUSE 1000  
           ELSE  
           PAUSE 5000  
  
           ENDIF
```

**EEPROM, READ Y WRITE** estas declaraciones las explicaremos con un ejemplo.

**EEPROM 5, [3,"K", 9,12]:** quiere decir colocar en la memoria EEPROM, dirección 5 el número 3, en la dirección 6 el carácter ASCII de K, es decir el número 75, y en la dirección 7 se guardará el número 9 y así sucesivamente.

La declaración **EEPROM** define el contenido inicial en un chip EEPROM.

**READ 5, pepe:** quiere decir leer la dirección 5 de la memoria **EEPROM** y guardar en la variable pepe, por consiguiente pepe se carga con el número 3. Además la declaración READ lee byte de un chip EEPROM

**WRITE 8,137:** quiere decir guardar en la dirección 8 de la memoria EEPROM el número 137, el dato anterior en este caso el 12 automáticamente se borra y se reescribe el número 137.

**WRITE 7, pepe:** en este caso la variable estaba cargada con el número 3 por consiguiente la dirección 7 de la memoria EEPROM se borra y se carga con el número 3. La declaración WRITE graba bytes en un chip EEPROM.

**Operadores de comparación:** se usan en declaraciones **IF... THEN** para comparar una expresión con otra. Los operadores soportados se muestran en la *Tabla 2.3*.

Operador	Descripción
= o ==	Igual
<> o !=	No igual
<	Menor
>	Mayor
<=	Menor o igual
>=	Mayor o igual

**Tabla 2.1 Operadores.**

**FOR ... NEXT:** el lazo **FOR ... NEXT** permite a los programas ejecutar un número de declaraciones (Body), un número de veces, usando una variable como contador.

```
FOR Count = start TO end {step {-} inc}
{Body}
NEXT {count}
```

El valor de start se asigna a la variable índice, count, puede ser una variable de cualquier tipo. Se ejecuta el body, que es opcional y puede ser omitido (quizás por un lazo de demora).

El valor de inc es sumado a (o restado si se especifica "-") count. Si no se define una cláusula step, se incrementa count en uno.

**Ejemplo:** FOR contador = 1 TO 10

```
        PAUSE 1000
NEXT
```

**RETURN y GOSUB:** Estos sirven cuando se tiene muchas repeticiones de una línea o grupo de líneas de programa.

La declaración **RETURN**, envía de regreso al inicio del programa para continuar después del **GOSUB** ya que este tiene una subrutina de cualquier nombre que le hayamos asignado, en donde se especifica el tiempo con un **PAUSE** con su tiempo respectivo, lo cual no es necesario escribir este tiempo en el programa varias veces sino solo una vez en la subrutina.

**Ejemplo:** Inicio:

```
portb=%00001
GOSUB x
portb=%00010
GOSUB x
```

```
GOTO inicio
x:
    Pause 1000
RETURN
```

**END:** Detiene la ejecución del proceso y entra en modo de baja potencia. Todos los pines de E/S permanecen en el estado en que se encuentran, **END** trabaja ejecutando una instrucción **SLEEP** continua dentro de un lazo.

Un **END**, **GOTO** deben ser colocados al final de un programa para evitar pasar del límite de la misma y comience nuevamente.

**Ejemplo:** END

**TRIS( A,B,C,D,E):** Esta declaración nos permite indicar al PIC cuales pines serán salida y cuales serán entradas.

**Ejemplo:**

```
TRISA = %00001111
```

Esto nos dice que los cuatro bits menos significativos son entradas y los cuatro bits más significativos son salidas.

**ADCON1:** Sirve para especificar cuál o cuáles pines del puerto (A o E) serán conversores análogos / digitales.

## 2.6. DESCRIPCIÓN DEL PROGRAMA DISEÑADO Y REALIZADO EN MICRO CODE STUDIO

**DEFINE** LCD\_DREG PORTB

Esta primera línea de instrucción nos permite definir el puerto del PIC que utilizaremos para la conexión del bus de datos del LCD (Puerto B).

**DEFINE** LCD\_DBIT 4

Esta instrucción nos indica desde que pin del puerto B vamos a conectar el bus de datos del LCD ( B.4 al B.7).

**DEFINE** LCD\_RSREG PORTB

Define en que puerto va a ser conectado el bit del registro del LCD (Puerto B).

**DEFINE** LCD\_RSBIT 3

El bit de registro va a estar conectado en el pin B.3

**DEFINE** LCD\_EREG PORTB

Define en que puerto va a ser conectado el bit de enable del LCD (Puerto B).

**DEFINE** LCD\_EBIT 2

El bit de enable va a estar conectado en el pin B.2

**DEFINE** ADC\_BITS 8

Fija el número de BITS del resultado.

**DEFINE** ADC\_CLOCK 3

Fije EL CLOCK (rc = 3)

**DEFINE** ADC\_SAMPLEUS 50

Fija el tiempo de muestreo en  $\mu$ s. ADC\_SAMPLEUS es el número de microsegundos que el programa espera entre fijar el canal y comenzar la conversión análoga/digital.

TRISA= %00101111                      Indica que A.0, A.1, A.2, A.3 Y A.5 son entradas  
y el A.4 es salida.



TRISC= %00000000	Indica que todo el p�rtico C es salida.
TRISE= %00000001	Indica que solo E.0 es entrada, el resto es salida.
ADCON1 = %00001110	Solo el pin A.0 es conversor A/D, los dem�s Digitales.

dato	<b>VAR BYTE</b>	Crea una variable y asigna un tama�o de 8 bits.
tempbaj	<b>VAR BYTE</b>	Crea una variable y asigna un tama�o de 8 bits.
tempalt	<b>VAR BYTE</b>	Crea una variable y asigna un tama�o de 8 bits.
x	<b>VAR BYTE</b>	Crea una variable y asigna un tama�o de 8 bits.
y	<b>VAR BYTE</b>	Crea una variable y asigna un tama�o de 8 bits.
w	<b>VAR BYTE</b>	Crea una variable y asigna un tama�o de 8 bits.
z	<b>VAR BYTE</b>	Crea una variable y asigna un tama�o de 8 bits.

up	<b>VAR</b>	portD.0	Asignamos un nombre al puerto D.0
down	<b>VAR</b>	portD.1	Asignamos un nombre al puerto D.1
ulti	<b>VAR</b>	portD.2	Asignamos un nombre al puerto D.2
prin	<b>VAR</b>	portD.3	Asignamos un nombre al puerto D.3
led	<b>VAR</b>	portD.4	Asignamos un nombre al puerto D.4
enter	<b>VAR</b>	portD.5	Asignamos un nombre al puerto D.5
bsubir	<b>VAR</b>	portD.6	Asignamos un nombre al puerto D.6
bbajar	<b>VAR</b>	portD.7	Asignamos un nombre al puerto D.7
bcelda	<b>VAR</b>	portA.1	Asignamos un nombre al puerto A.1
bmodo	<b>VAR</b>	portA.2	Asignamos un nombre al puerto A.2
blateral	<b>VAR</b>	portA.3	Asignamos un nombre al puerto A.3
bfrontal	<b>VAR</b>	portA.5	Asignamos un nombre al puerto A.5
bcondi	<b>VAR</b>	portE.0	Asignamos un nombre al puerto E.0

**EEPROM** 0, [10,2] Guardamos desde la direcci n 0 de la EEPROM el n mero 10 y en la direcci n 1 el n mero 24.

**ON INTERRUPT GOTO** parada En caso de existir una interrupci n saltar a parada.

<b>INTCON = %10010000</b>	Habilita la interrupción al puerto B0.
<b>inicio:</b>	Nos indica el inicio del programa.
<b>For x =1 TO 3</b>	Crear un lazo de tres repeticiones.
<b>HIGH led</b>	Sacar 1L en led.
<b>PAUSE 200</b>	Led permanezca encendido 200 ms.
<b>LOW led</b>	Sacar 0L en led.
<b>PAUSE 200</b>	Led permanezca apagado 200ms.
<b>NEXT</b>	Terminado el lazo, continúa con el resto del programa.
<b>READ 0, tempbaj</b>	Lee la EEPROM 0 y lo guarda en la variable tempbaj.
<b>READ 1, tempalt</b>	Lee la EEPROM 1 y lo guarda en la variable tempalt.
<b>sensor:</b>	Nombre de subrutina.
<b>IF bmodo =0 THEN luz</b>	Si la condición es verdadera salte a luz caso contrario continúe con el programa. Este interruptor nos permite cambiar de modo la automatización de temperatura a condición de luz/oscuridad o viceversa.
<b>ADCIN 0, dato</b>	Leer el canal 0 (A0) y guarde en la variable dato.
<b>LCDOUT \$fe, 1, "T.mi T.act T.max"</b>	Limpiar LCD y sacar texto.
<b>dato = dato /2</b>	El valor en la variable dato dividirlo para 2.
<b>LCDOUT \$fe,\$c6,DEC dato,"oC"</b>	Sacar por el LCD el valor de dato en decimal.
<b>LCDOUT \$fe,\$c0,DEC tempbaj,"oC"</b>	Sacar por el LCD el valor de tempbaj en decimal.
<b>LCDOUT \$fe,\$cc,DEC tempalt,"oC"</b>	Sacar por el LCD el valor de dato en decimal.

<b>FOR</b> x = 1 <b>TO</b> 50	Crear un lazo de 50 repeticiones.
<b>IF</b> enter =0 <b>THEN</b> grabar1a	Si la condición es verdadera saltar a grabar1a, caso contrario continuar con el programa.
<b>PAUSE</b> 10	Detiene el programa 10 ms.
<b>NEXT</b>	Siguiente repetición hasta que x=50, luego continúa con el resto del programa
<b>IF</b> up =0 <b>THEN</b> derecha	Si las condiciones son verdaderas ir a sus respectivas subrutinas, caso contrario
<b>IF</b> down =0 <b>THEN</b> izquierda	continuar con el programa
<b>IF</b> blateral=0 <b>THEN</b> lateral	
<b>IF</b> bfrontal=0 <b>THEN</b> frontal	
<b>IF</b> dato < tempbaj <b>THEN</b> cerrarse	
<b>IF</b> dato > tempalt <b>THEN</b> abrirse	
portc= %00000000	Enviar 0L a todo el puerto C.
<b>GOTO</b> sensor	Regresar a la subrutina sensor.
cerrarse:	Nombre de la subrutina.
<b>IF</b> ulti =0 <b>THEN</b> sensor	Si la condición es verdadera saltar a sensor, caso contrario continuar con el programa.
<b>IF</b> bcondi=0 <b>THEN</b> cerrar	Si la condición es verdadera saltar a cerrar, caso contrario continuar con el programa
<b>GOSUB</b> lateral1	Saltar a lateral1 y regresar para continuar con el resto del programa.
cerrar:	Nombre de la subrutina.
portc=%00001001	Enviar 1L a los puerto C.0 y C.3
<b>GOSUB</b> timer	Saltar a timer y regresar para continuar con el resto del programa.
portc=%00001100	Enviar 1L a los puerto C.3 y C.2

<b>GOSUB</b> timer	Saltar a timer y regresar para continuar con el resto del programa
portc=%00000110	Enviar 1L a los puertos C.2 y C.1
<b>GOSUB</b> timer	Saltar a timer y regresar para continuar con el resto del programa
portc=%00000011	Enviar 1L a los puertos C.1 y C.0
<b>GOSUB</b> timer	Saltar a timer y regresar para continuar con el resto del programa.
portc =%00000000	Enviar 0L a todo el puerto C.
<b>IF</b> ulti =0 <b>THEN</b> san	Si la condición es verdadera saltar a close, caso contrario continuar con el programa.
<b>GOTO</b> cerrar	Regresar a la subrutina cerrar.
close:	Nombre de la subrutina
portc=%00001001	Enviar 1L a los puertos C.0 y C.3.
<b>GOSUB</b> timer	Saltar a timer y regresar para continuar con el resto del programa.
portc=%00001100	Enviar 1L a los puertos C.3 y C.2.
<b>GOSUB</b> timer	Saltar a timer y regresar para continuar con el resto del programa.
portc=%00000110	Enviar 1L a los puertos C.2 y C.13.
<b>GOSUB</b> timer	Saltar a timer y regresar para continuar con el resto del programa.
portc =%00000000	Enviar 0L a todo el puerto C.
<b>GOTO</b> frontal1	Ir a la subrutina frontal1.
abrirse:	Nombre de la subrutina.
<b>IF</b> prin =0 <b>THEN</b> sensor	Si la condición es verdadera saltar a sensor, caso contrario continuar con el programa.
<b>IF</b> bcondi =0 <b>THEN</b> abrir	Si la condición es verdadera saltar a abrir, caso contrario continuar con el programa.

<b>GOSUB</b> lateral1	Saltar a lateral1 y regresar para continuar con el resto del programa
abrir:	Nombre de la subrutina.
portc=%00000011	Enviar 1L a los puertos C.0 y C.1
<b>GOSUB</b> timer	Saltar a timer y regresar para continuar con el resto del programa
portc=%00000110	Enviar 1L a los puertos C.1 y C.2
<b>GOSUB</b> timer	Saltar a timer y regresar para continuar con el resto del programa
portc=%00001100	Enviar 1L a los puertos C.2 y C.3
<b>GOSUB</b> timer	Saltar a timer y regresar para continuar con el resto del programa
portc=%00001001	Enviar 1L a los puertos C.3 y C.0
<b>GOSUB</b> timer	Saltar a timer y regresar para continuar con el resto del programa
portc =%00000000	Enviar 0L a todo el puerto C.
<b>IF</b> prin =0 <b>THEN</b> open	Si la condición es verdadera saltar a open, caso contrario continuar con el programa.
<b>GOTO</b> abrir	Regresar a la subrutina abrir.
open:	Nombre de la subrutina.
portc=%00000011	Enviar 1L a los puertos C.1 y C.0
<b>GOSUB</b> timer	Saltar a timer y regresar para continuar con el resto del programa.
portc=%00000110	Enviar 1L a los puertos C.1 y C.2
<b>GOSUB</b> timer	Saltar a timer y regresar para continuar con el resto del programa.
portc=%00001100	Enviar 1L a los puertos C.2 y C.3
<b>GOSUB</b> timer	Saltar a timer y regresar para continuar con el resto del programa.
portc =%00000000	Enviar 0L a todo el puerto C.

<b>GOTO</b> sensor	Regresar a la subrutina sensor.
derecha:	Nombre de la subrutina.
<b>GOSUB</b> soltar	Saltar a soltar (antirrebote) y regresar para continuar con el resto del programa.
<b>IF</b> ulti =0 <b>THEN</b> sensor	Si la condición es verdadera saltar a sensor, caso contrario continuar con el programa.
<b>IF</b> bcondi=0 <b>THEN</b> primero	Si la condición es verdadera saltar a primero, caso contrario continuar con el programa.
<b>GOSUB</b> lateral1	Saltar a lateral1 y regresar para continuar con el resto del programa.
primero:	Nombre de la subrutina
<b>FOR</b> w = 1 <b>TO</b> 500	Crear lazo de repetición de 500 veces.
portc=%00001001	Enviar 1L a los puertos C.0 y C.3
<b>GOSUB</b> timer	Saltar a timer y regresar para continuar con el resto del programa
portc=%00001100	Enviar 1L a los puertos C.3 y C.2
<b>GOSUB</b> timer	Saltar a timer y regresar para continuar con el resto del programa.
portc=%00000110	Enviar 1L a los puertos C.2 y C.1
<b>GOSUB</b> timer	Saltar a timer y regresar para continuar con el resto del programa
portc=%00000011	Enviar 1L a los puertos C.1 y C.0
<b>GOSUB</b> timer	Saltar a timer y regresar para continuar con el resto del programa.
<b>IF</b> ulti =0 <b>THEN</b> sensor	Si la condición es verdadera saltar a sensor, caso contrario continuar con el programa.

<b>NEXT</b>	Siguiente repetición hasta que w=500 luego continúa con el resto del programa
portc=%00000000	Enviar 0L a todo el puerto C
<b>GOTO</b> sensar	Regresar a la subrutina sensar
izquierda:	Nombre de la subrutina.
<b>GOSUB</b> soltar	Saltar a soltar (antirrebote) y regresar.
<b>IF</b> prin =0 <b>THEN</b> sensar	Si la condición es verdadera saltar a sensar, caso contrario continuar con el programa.
<b>IF</b> bcondi=0 <b>THEN</b> segundo	Si la condición es verdadera saltar a segundo, caso contrario continuar con el programa.
<b>GOSUB</b> lateral1	Saltar a lateral1 y regresar para continuar con el resto del programa.
segundo:	Nombre de la subrutina
<b>FOR</b> w = 1 <b>TO</b> 500	Crear lazo de repetición de 500 veces.
portc=%00000011	Enviar 1L a los puertos C.0 y C.1
<b>GOSUB</b> timer	Saltar a timer y regresar para continuar con el resto del programa
portc=%00000110	Enviar 1L a los puertos C.1 y C.2
<b>GOSUB</b> timer	Saltar a timer y regresar para continuar con el resto del programa.
portc=%00001100	Enviar 1L a los puertos C.2 y C.3
<b>GOSUB</b> timer	Saltar a timer y regresar para continuar con el resto del programa
portc=%00001001	Enviar 1L a los puertos C.3 y C.0
<b>GOSUB</b> timer	Saltar a timer y regresar para continuar con el resto del programa
<b>IF</b> prin =0 <b>THEN</b> sensar	Si la condición es verdadera saltar a sensar, caso contrario continuar con el programa.





Lateral2:

portc= %10010000

**GOSUB** timer

portc= %11000000

**GOSUB** timer

portc= %01100000

**GOSUB** timer

portc= %00110000

**GOSUB** timer

portc= %00000000

**IF** bcondi =1 **THEN** lateral2

**PAUSE** 1000

**GOTO** sensar

Nombre de la subrutina

Enviar 1L a los puertos C.4 y C.5

Saltar a timer y regresar para continuar con el resto del programa

Enviar 1L a los puertos C.5 y C.6

Saltar a timer y regresar para continuar con el resto del programa

Enviar 1L a los puertos C.6 y C.7

Saltar a timer y regresar para continuar con el resto del programa

Enviar 1L a los puertos C.7 y C.4

Saltar a timer y regresar para continuar con el resto del programa.

Enviar 0L a todo el puerto C

Regresar a la subrutina sensar

frontal1:

**IF** ulti=0 **THEN** frontal2

**GOTO** sensar

Nombre de la subrutina

Si la condición es verdadera saltar a frontal2, caso contrario continuar con el programa.

Regresar a la subrutina sensar.

Frontal2:

**FOR** z=1 **TO** 700

portc= %00110000

**GOSUB** timer

portc= %01100000

**GOSUB** timer

portc= %11000000

Nombre de la subrutina.

Crear lazo de repetición de 700 veces

Enviar 1L a los puertos C.4 y C.7

Saltar a timer y regresar para continuar con el resto del programa

Enviar 1L a los puertos C.7 y C.6

Saltar a timer y regresar para continuar con el resto del programa

Enviar 1L a los puertos C.6 y C.5

<b>GOSUB</b> timer	Saltar a timer y regresar para continuar con el resto del programa
portc= %10010000	Enviar 1L a los puertos C.4 y C.3
<b>GOSUB</b> timer	Saltar a timer y regresar para continuar con el resto del programa.
<b>NEXT</b>	Siguiente repetición hasta que z=700 luego continúa con el resto del programa
portc= %00000000	Enviar 0L a todo el puerto C
<b>GOTO</b> sensor	Regresar a la subrutina sensor
frontal:	Nombre de la subrutina.
<b>GOSUB</b> soltar	Saltar a soltar (antirrebote) y regresar
<b>IF</b> ulti =0 <b>AND</b> bcondi =0 <b>THEN</b> voy	Si la condición es verdadera saltar a voy, caso contrario continuar con el programa.
<b>GOTO</b> sensor	Regresar a la subrutina sensor.
voy:	Nombre de la subrutina.
<b>FOR</b> z = 1 <b>TO</b> 700	Crear lazo de repetición de 700 veces
portc=%00110000	Enviar 1L a los puertos C.4 y C.5
<b>GOSUB</b> timer	Saltar a timer y regresar para continuar con el resto del programa.
portc=%01100000	Enviar 1L a los puertos C.5 y C.6
<b>GOSUB</b> timer	Saltar a timer y regresar para continuar con el resto del programa.
portc=%11000000	Enviar 1L a los puertos C.6 y C.7
<b>GOSUB</b> timer	Saltar a timer y regresar para continuar con el resto del programa.
portc=%10010000	Enviar 1L a los puertos C.7 y C.4
<b>GOSUB</b> timer	Saltar a timer y regresar para continuar con el resto del programa.

<b>NEXT</b>	Si siguiente repetición hasta que z=700 luego continúa con el resto del programa
portc =%00000000	Enviar 0L a todo el p <sup>o</sup> rtico C.
<b>GOTO</b> sensar	Regresar a la subrutina sensar.
grabar1a:	Nombre de la subrutina.
<b>GOSUB</b> soltar	Saltar a soltar (antirrebote) y regresar.
grabar1:	Nombre de la subrutina.
<b>LCDOUT</b> \$fe, 1 , "Programar temp."	
<b>LCDOUT</b> \$fe,\$c0, "baja= ",dec tempbaj , " oC"	
<b>PAUSE</b> 100	
<b>IF</b> bbajar=0 <b>THEN</b> restar1	Si una condición es verdadera saltar a sus respectiva subrutina caso contrario continuar con el programa.
<b>IF</b> bsubir=0 <b>THEN</b> sumar1	
<b>IF</b> enter=0 <b>THEN</b> grabarA	
<b>GOTO</b> grabar1	
restar1:	Nombre de la subrutina.
<b>GOSUB</b> soltar	Saltar a soltar (antirrebote) y regresar
<b>IF</b> tempbaj < 1 <b>THEN</b> grabar1	Si la condición es verdadera saltar a grabar1, caso contrario continuar con el programa.
tempbaj= tempbaj -1	Disminuir 1 a la variable tempbaj y grabar el resultado de la resta en tempbaj.
<b>GOTO</b> grabar1	Regresar a la subrutina grabar1
sumar1:	Nombre de la subrutina
<b>GOSUB</b> soltar	Saltar a soltar (antirrebote) y regresar
<b>IF</b> tempbaj > 40 <b>THEN</b> grabar1	Si la condición es verdadera saltar a sensar, caso contrario continuar con el programa.

tempbaj= tempbaj + 1	Sumar 1 a la variable tempbaj y guardar el resultado de la suma en tempbaj.
<b>GOTO</b> grabar1	Regresar a la subrutina grabar1.
grabarA:	Nombre de la subrutina
<b>GOSUB</b> soltar	Saltar a soltar (antirrebote) y regresar.
<b>WRITE</b> 0,tempbaj	Escribir en la dirección 0 de la EEPROM el valor de la variable tempbaj.
grabar2:	Nombre de la subrutina
<b>LCDOUT</b> \$fe, 1 ,"Programar temp."	
<b>LCDOUT</b> \$fe,\$c0,"alta= ",dec tempalt ," oC"	
<b>PAUSE</b> 100	
<b>IF</b> bbajar=0 <b>THEN</b> restar2	Si una condición es verdadera saltar a su respectiva subrutina, caso contrario
<b>IF</b> bsubir=0 <b>THEN</b> sumar2a	continuar con el programa.
<b>IF</b> enter=0 <b>THEN</b> grabarB	
<b>GOTO</b> grabar2	Regresar a la subrutina grabar2
restar2:	Nombre de la subrutina
<b>GOSUB</b> soltar	Saltar a soltar (antirrebote) y regresar
<b>IF</b> tempalt < 5 <b>THEN</b> grabar2	Si la condición es verdadera saltar a sensar, caso contrario continuar con el programa.
tempalt= tempalt -1	Restar 1 a la variable tempalt y guardar el resultado de la resta en tempalt.
<b>GOTO</b> grabar2	Regresar a la subrutina grabar2
sumar2:	Nombre de la subrutina
<b>GOSUB</b> soltar	Saltar a soltar (antirrebote) y regresar

<p><b>IF</b> tempalt &gt; 50 <b>THEN</b> grabar2</p> <p>tempalt= tempalt + 1</p> <p><b>GOTO</b> grabar2</p> <p>grabarB:</p> <p><b>GOSUB</b> soltar</p> <p><b>WRITE</b> 1,tempalt</p> <p><b>GOTO</b> inicio</p> <p>luz:</p> <p><b>PAUSE</b> 50</p> <p><b>LCDOUT</b> \$fe, 1,"modo claridad:"</p> <p><b>IF</b> bcelda =0 <b>THEN</b> dia</p> <p>noche:</p> <p><b>LCDOUT</b> \$fe,\$c6, "noche."</p> <p><b>GOTO</b> cerrarse</p> <p>dia:</p> <p><b>LCDOUT</b> \$fe, \$c6, "dia."</p> <p><b>GOTO</b> abrirse</p> <p>timer:</p> <p><b>PAUSE</b> 8</p>	<p>Si la condición es verdadera saltar a grabar, caso contrario continuar con el programa.</p> <p>Sumar 1 a la variable tempalt y grabar el resultado de la suma en tempalt.</p> <p>Regresar a la subrutina grabar2</p> <p>Nombre de la subrutina</p> <p>Saltar a soltar (antirrebote) y regresar.</p> <p>Escribir en la dirección 1 de la EEPROM el valor de la variable tempalt.</p> <p>Regresar a la subrutina inicio.</p> <p>Nombre de la subrutina</p> <p>Detener el programa 50 ms.</p> <p>Si la condición es verdadera saltar a día, caso contrario continuar con el programa.</p> <p>Nombre de la subrutina</p> <p>Regresar a la subrutina cerrarse.</p> <p>Nombre de la subrutina</p> <p>Regresar a la subrutina abrirse</p> <p>Nombre de la subrutina</p> <p>El programa se detiene 8 ms</p>
--	--

**RETURN**

Regresa una línea debajo del gosub que le envió a esta subrutina.

soltar:

Nombre de la subrutina del antirrebote.

**HIGH** led

El programa provoca un parpadeo del led.

**PAUSE** 150

**LOW** led

soltar2:

Nombre de la subrutina

**IF** bbajar=0 **THEN** soltar2

Si una condición es verdadera saltar a su respectiva subrutina.

**IF** bsubir=0 **THEN** soltar2

Esta subrutina nos ayuda a evitar el rebote que provocan las teclas al ser presionadas.

**IF** enter=0 **THEN** soltar2

**IF** up= 0 **THEN** soltar2

**IF** down=0 **THEN** soltar2

**IF** blateral= 0 **THEN** soltar2

**IF** bfrontal =0 **THEN** soltar2

**PAUSE** 100

**RETURN**

Regresa una línea debajo del gosub que le envió a esta subrutina.

**END**

nos indica la terminación del programa.

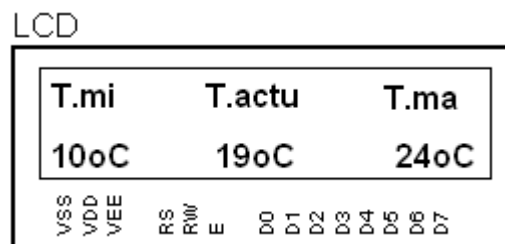
## 2.7. FUNCIONAMIENTO DEL SISTEMA DE AUTOMATIZACIÓN DE APERTURA Y CIERRE DE PERSIANAS VERTICALES.

Como se ha explicado en los temas anteriores el funcionamiento de cada elemento utilizado en el proyecto por separado y también se ha descrito el funcionamiento del programa que es grabado en el PIC, es necesario explicar el funcionamiento en conjunto.

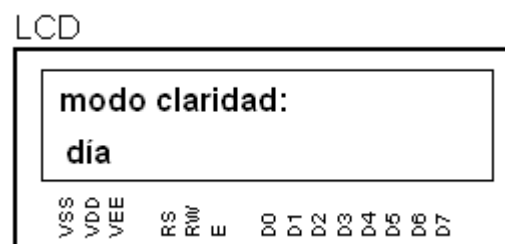
Lo primero que se observa al prender el sistema de automatización, es que el led titila tres veces seguidas indicando que el PIC está trabajando perfectamente.

A continuación el PIC verifica en qué estado se encuentra el interruptor de modalidad, si se encuentra en posición 0 en el LCD aparecerá lo que se indica en la *Figura 2.12*, tomando en cuenta que la temperatura actual es la que está censando el sensor LM35 en tiempo real.

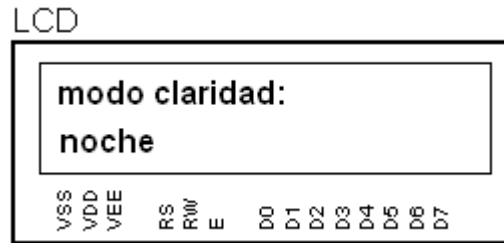
Si el mismo interruptor se encuentra en la posición 1 en el LCD aparecerá lo que se indica en la *Figura 2.13a* cuando existe un alto grado de iluminación sobre la LDR. Si el grado de iluminación sobre la LDR es baja muestra lo de la *Figura 2.13b*.



**Figura 2.12** El sistema se automatización se encuentra en modalidad temperatura.



**Figura 2.13a** El sistema se automatización se encuentra en modalidad claridad.

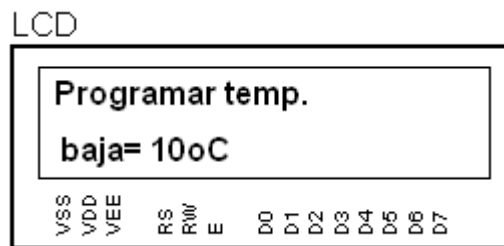


**Figura 2.13b** El sistema de automatización se encuentra en modalidad oscuridad.

El funcionamiento en modalidad temperatura es el siguiente:

La temperatura actual es la que el PIC está censando a cada instante con las variaciones de voltaje que proporciona el LM35 por el cambio de temperatura que existe en el ambiente donde se encuentra dicho sensor.

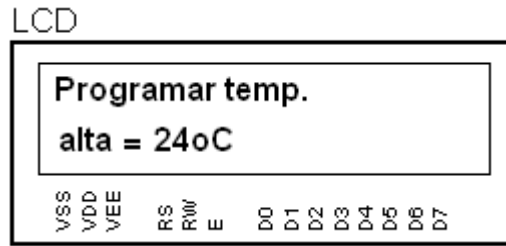
La temperatura mínima y temperatura máxima que aparece en el LCD son las que están grabadas previamente en la EEPROM del PIC, las mismas que se pueden variar con el uso de las teclas enter, subir y bajar a conveniencia del usuario. Al presionar la primera vez la tecla enter tenemos en el LCD lo que indica en la *Figura 2.14* mientras que con el uso de las teclas subir y bajar variamos aquel valor sin bajar de 10°C o exceder de 40°C.



**Figura 2.14** Pantalla para programar la temperatura baja.

Presionando por segunda ocasión la tecla enter el led titila una sola vez y el LCD cambia como se indica en la *Figura 2.15*, de igual manera al presionar las teclas subir o bajar variamos aquel valor sin bajar de 50°C o exceder de 50°C.





**Figura 2.15 Pantalla para programar la temperatura alta.**

Al presionar por tercera ocasión la tecla enter el led parpadeará cuatro veces confirmando de esa manera que los datos calibrados se han guardado en la EEPROM del PIC.

Si la temperatura actual llega a disminuir por debajo del valor de la temperatura mínima lo primero que realiza el PIC es confirmar en que estado se encuentra el sensor ulti, si éste se encuentra en 0L el PIC no realiza ninguna función y envía nuevamente a censar la temperatura. Pero si el sensor ulti se encuentra en 1L el PIC confirma en qué estado se encuentra el micro-switch (bcondi) si éste está en 0L (presionado) ordena al motor de desplazamiento horizontal que gire en sentido horario hasta que el sensor ulti cambie de estado deteniendo el movimiento de motor al instante, acto seguido el PIC ordena que el motor frontal/lateral gire en sentido antihorario un determinado número de vueltas mientras que el micro-switch ha cambiado de estado a 1L.

Debemos tener en cuenta que el micro-switch puede estar en 1L solo cuando la persiana esté completamente cerrada.

La otra probabilidad es que la temperatura actual sobrepase la temperatura máxima y lo primero en realizar el PIC es confirmar el estado del sensor prin, si este se encuentra en 0L el PIC no realiza ninguna función y envía nuevamente a censar la temperatura. Caso contrario el PIC confirma en qué estado se encuentra el micro-switch (bcondi) si éste está en 0L ordena al motor de desplazamiento horizontal que gire en sentido antihorario hasta que el sensor prin cambie de estado a 0L deteniendo el movimiento del motor al instante.

Si el micro-switch (bcondi) se encuentra en 1L el PIC ordena que el motor lateral/frontal gire en sentido horario hasta que el micro-switch cambie su estado a 0L deteniendo su movimiento al instante y provocando el movimiento del motor de desplazamiento horizontal que gire en sentido antihorario hasta que el sensor ulti cambie a 0L y detenga el movimiento del motor.

El uso de las teclas cerrar y abrir es exclusivo en esta modalidad para hacer girar en sentido horario si el sensor ulti  $\neq 0$  o en sentido antihorario si el sensor prin  $\neq 0$  respectivamente.

El uso de las teclas frontal y lateral es exclusivamente cuando el sensor ulti está en 0L, de cumplirse esta condición al pulsar la tecla frontal el motor frontal/lateral gira en sentido antihorario un determinado número de vueltas. Mientras que al pulsar la tecla lateral el motor frontal/lateral gira en sentido horario hasta que el micro-switch cambia a 0L cortando el movimiento del motor al instante.

El funcionamiento en modalidad claridad / oscuridad es el siguiente:

Si el grado de iluminación en la LDR es alta (día) el comportamiento del circuito será igual que cuando la temperatura actual sobrepasa el valor de la temperatura máxima.

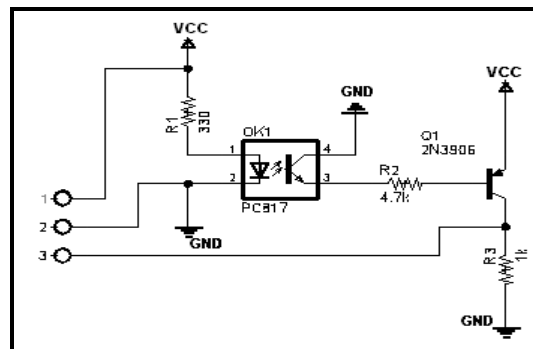
Mientras que si el grado de iluminación en la LDR es baja (noche) el comportamiento del circuito será igual que cuando la temperatura actual disminuye por debajo del valor de la temperatura mínima, del mismo modo es con las teclas frontal y lateral.

Se recuerda que las teclas abrir, cerrar, enter, subir y bajar no desempeñan ninguna función en la modalidad claridad / oscuridad.

## 2.8. DISEÑO DE LAS TARJETAS DEL SISTEMA DE AUTOMATIZACIÓN

El diseño del ruteado de las tarjetas se llevó a cabo con la ayuda de los programas Eagle 4.11 y Proteus V7.1 SP2 (ARES e ISIS). El diagrama esquemático de la tarjeta principal se lo elaboró en Proteus ISIS (*Figura 2.17*), mientras que el esquemático de la placa de los opto-acopladores se lo ha elaborado en Eagle (*Figura 2.16*).

Los diagramas esquemáticos es la parte más importante para la elaboración del ruteado puesto que ahí se coloca los elementos a ser utilizados y en algunos casos cambiar la forma de sus encapsulados o crear encapsulados nuevos.



**Figura 2.16** Diagrama esquemático de la placa de los opto-acopladores.

El ruteado de la placa principal se logró con Proteus ARES (*Figura 2.14*) con la creación de 11 puentes para poder unir todas las pistas de la placa, el ruteado de la placa de los sensores se la realizó en Eagle (*Figura 2.18*).

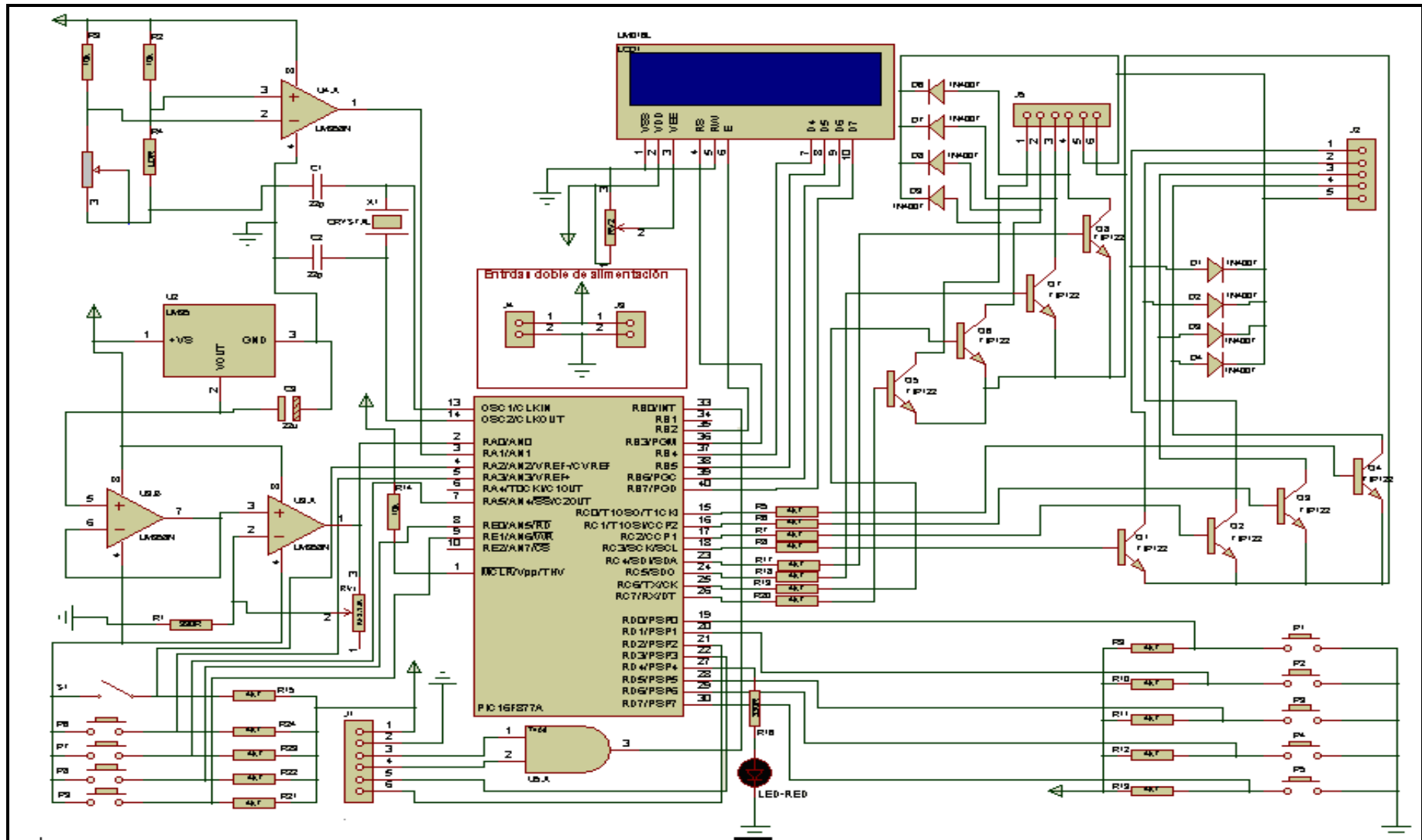
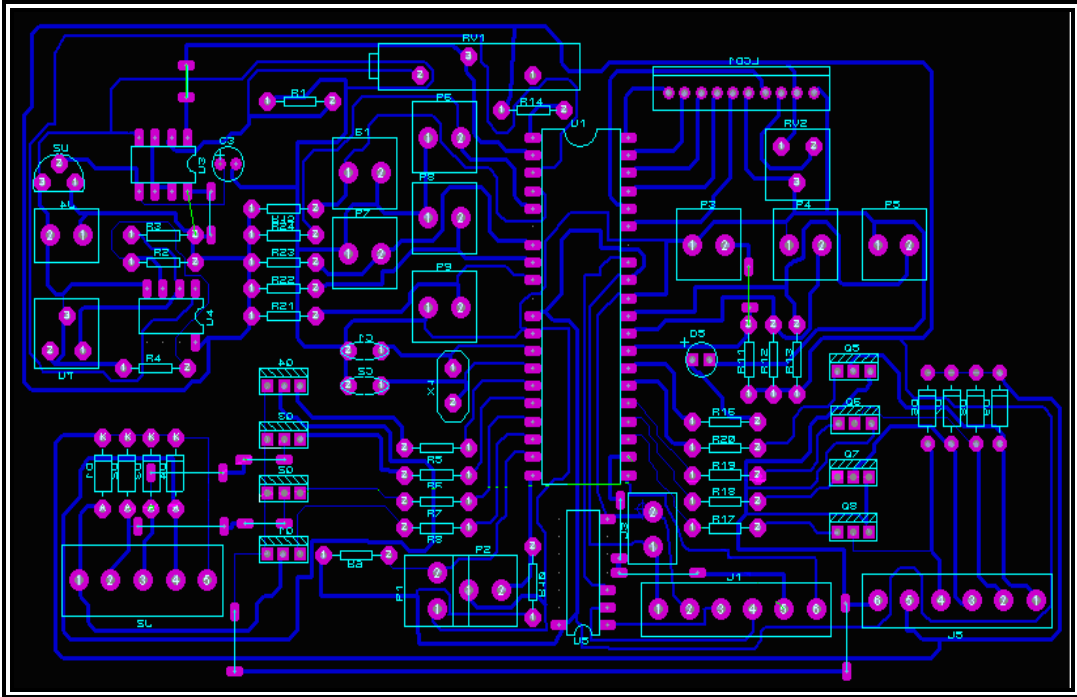
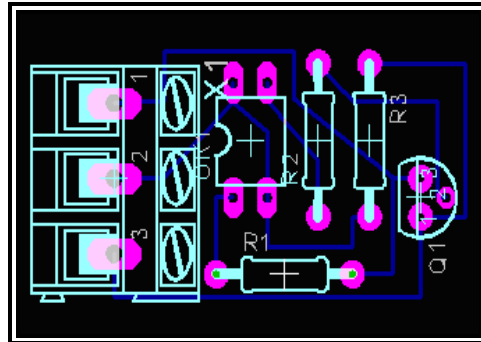


Figura 2.17 Diagrama esquemático de la placa principal.



**Figura 2.18 Ruteado de la placa principal en Proteus ARES.**



**Figura 2.18 Ruteado de la placa del opto-acoplador en Eagle**

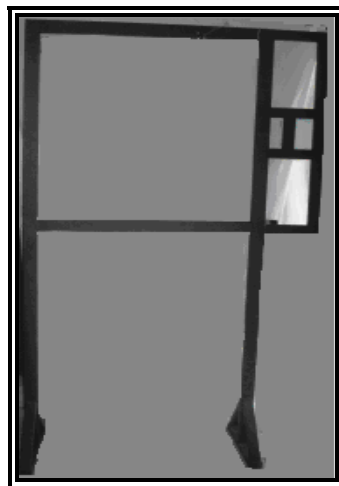
La *Figura 2.19* es una imagen en 3D que nos proporciona el programa Proteus ARES de cómo será el resultado final de la placa principal.



## **CAPÍTULO III: CONSTRUCCIÓN Y ACOPLAMIENTO DEL SISTEMA DE AUTOMATIZACIÓN DE APERTURA Y CIERRE DE PERSIANAS VERTICALES**

### **3.1. CONSTRUCCIÓN Y MODIFICACIONES**

El primer paso para realizar el sistema de automatización de apertura y cierre de persianas verticales, es la construcción de un marco en forma de ventana elaborado en hierro (*Figura 3.1*), en el cual está montado el riel y la persiana vertical.



*Figura 3.1 Marco de la persiana.*

Las modificaciones que se realizaron en la estructura de la persiana vertical son las siguientes:

Se debe tener en cuenta que el riel de la persiana vertical que se utiliza en este proyecto, tiene dos tiras que cuelgan en uno de sus extremos, la una es una cadena de bolas cuya función es la de colocar las persianas en forma frontal o lateral, mientras que el cordón de nylon, tiene la función de desplazar las persianas hacia la derecha o hacia la izquierda.

La modificación que se realizó, en las dos tiras que cuelgan del riel de la persiana, hizo que adquirieran una forma física mixta, es decir tanto la cadena de bolas como el cordón de nylon se les acopló a cada una de ellas una banda dentada de caucho, con la finalidad que solo la banda dentada circule por el motor paso a paso de 5 hilos unipolar y provoque el movimiento deseado y con la condición de que esta misma banda nunca entre a la parte interna del riel (*Figura 3.2*).



**Figura 3.2 Cadenas mixtas para el movimiento de las persianas.**

Después a las dos tiras se las colocó en su respectivo lugar, y luego se instaló el riel de la persiana en el marco de hierro como que se indica en la *Figura 3.3*



**Figura 3.3 Acoplamiento de la persiana en la placa.**

Otra modificación que se realizó, es la colocación de un micro-switch, mientras que el primer seguro donde se introduce la tira de la persiana se ha instalado una tira de metal cuya finalidad es la de pulsar el micro-switch cuando las tiras de la persiana se encuentran en posición lateral (*Figura 3.4*).





**Figura 3.4** *Micro-switch con su mecanismo de accionamiento sobre la persiana.*

En el seguro donde se introduce la última persiana se ha colocado una segunda tira metálica para provocar la obstrucción de los opto-acopladores (Figura 3.5), cuando la persiana se ha cerrado o se ha abierto en su totalidad.



**Figura 3.5** *Mecanismo de accionamiento de los opto-acopladores.*

### **3.2. ELABORACIÓN DE LOS CIRCUITOS IMPRESOS**

Hechos los cambios ya mencionados en la persiana, procedemos a empotrar la misma en el marco de hierro, luego se procede a la fabricación del circuito impreso para la fuente de voltaje, la placa principal y las placas de los opto-acopladores.

Los pasos a seguir son los siguientes:

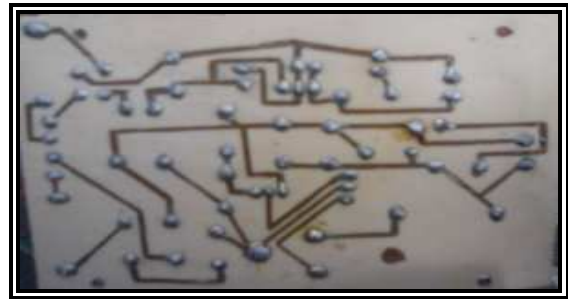
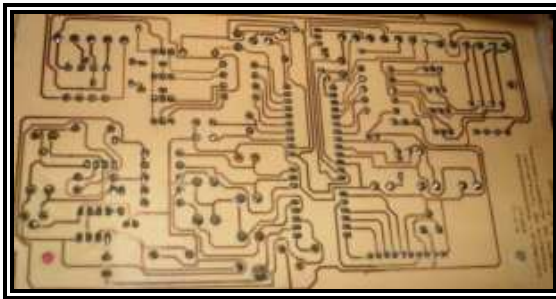
- Teniendo ya el diseño de las pistas (ruteado) con la ayuda de los programas Eagle y ARES (Proteus) se procedió a la impresión con láser en papel termo transferi.

- En una baquelita de cobre con las dimensiones de 10x17cm para la placa principal, de 6x8cm para la fuente de voltaje y de 3x5cm para las placas de los optoacopladores, se procede a la adhesión del papel térmico sobre las mismas, lo cual se realiza con una plancha casera a máxima temperatura durante 50 minutos para la placa principal mientras que para la fuente de voltaje y las placas de los optoacopladores 23 minutos cada una.
- Terminado ya el planchado de las placas se retocó con marcador donde el papel térmico no se haya adherido correctamente.
- Luego se prepara el cloruro ferrico disolviéndolo con un poco de agua, para colocar la placa de cobre con el diseño del circuito impreso hacia arriba. Mientras la placa de cobre se encuentre sumergida en la solución de cloruro férrico se debe agitar el recipiente donde se encuentra inclinándolo de lado a lado para que de esta manera el químico pueda disolver más rápido el cobre de la placa.
- Después de que el cloruro férrico haya consumido todo el cobre, se procede a sacar la placa del recipiente y a retirar la tinta con thinner y un trapo. Para dar mayor presentación al circuito impreso se lo debe lavar y secar con papel de cocina, de ser necesario pulir suavemente con viruta de acero.
- Comenzamos a perforar en los lugares indicados, para esto se debe emplear una broca de 0,75mm de diámetro, para los orificios de resistencias, capacitores y semiconductores, para los orificios de bornera, pines de los diferentes integrados y del PIC una broca de 1mm de diámetro es la más adecuada. Aquí será de suma utilidad acertar al orificio central de la isla para que quede la hilera de perforaciones lo más pareja posible.
- Colocados los elementos en su lugar y posiciones respectivas se procede a soldar con un cautín de 40w, procurando no dejar pistas cristalinas o también llamadas sueldas frías. Tomando en cuenta que siempre se debe montar primero los componentes de menor espesor, comenzando por los puentes de

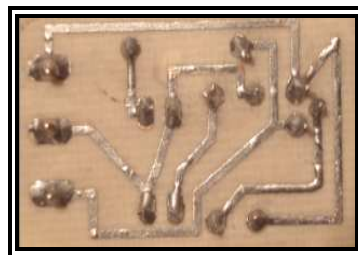
alambre, diodos, resistencias, pequeños capacitores, transistores, pines de conexión y los zócalos de circuitos integrados. Lo más adecuado es montar zócalos para los circuitos integrados, para cuando sea necesario reemplazarlos en futuras reparaciones será simple quitar uno y colocar otro sin siquiera usar el caudín. Además, el desoldar y soldar una plaqueta hace que la pista vaya perdiendo adherencia a la baquelita.

- Terminado ya de soldar todos los elementos se debe limpiar la placa del excedente de pomada con un cepillo y thinner, para terminar con el acabado final de las placas se debe lacar las pistas.

Las placas resultantes luego de todos los pasos mencionados anteriormente se pueden ver en la *Figura 3.6a*, *Figura 3.6b*, *Figura 3.6c*.



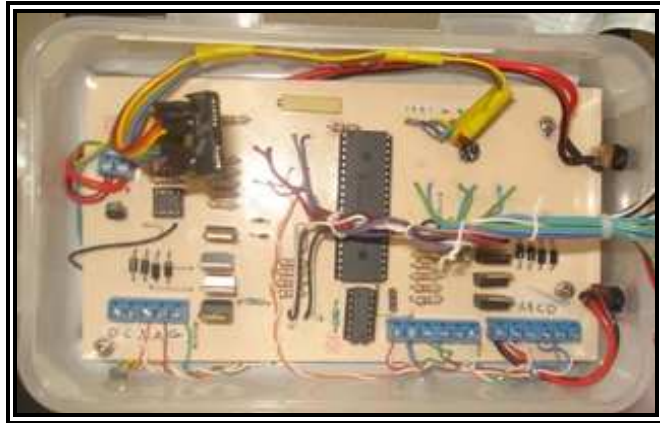
**Figura 3.6a** Acabado de la placa principal. **Figura 3.6b** Acabado de la placa de las fuentes.



**Figura 3.6c** Acabado de la placa de los opto-acopladores.

### 3.2.1. Acabado final de las placas electrónicas

En la *Figura 3.7* se muestra el resultado final de la placa principal, se puede comparar con el dibujo en 3D que nos proporcionó el programa Proteus ARES (Capítulo II), es muy parecido a nuestra placa final.



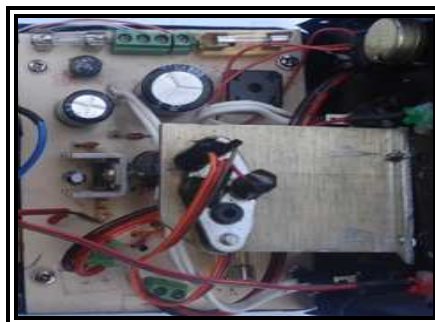
**Figura 3.7** Acabado final del panel de control.

La *Figura 3.8* se muestra la placa resultante de los opto-acopladores.



**Figura 3.8** Placa de los opto-acopladores.

La *Figura 3.9* muestra la placa resultante de la fuente de voltaje.



**Figura 3.9** Placa de las fuentes de voltaje y la fuente de alimentación variable-.

### 3.2.2. Elaboración de las cajas donde serán colocadas las placas.

La caja donde está colocada la fuente de voltaje es de tool apropiada para esta aplicación. En el interior de la caja a un extremo irán colocados dos transformadores, de 1A para la alimentación del PIC 16F877A y todos sus componentes electrónicos y de 2A para la alimentación de las bobinas de los motores pasa a paso y al otro extremo irá la placa de la fuente.

La caja donde está colocada la placa principal está elaborada en plástico. La tapa posee cuatro seguros que se colocan a presión, de esta forma la placa estará bien protegida y en caso de tener que manipular los potenciómetros que se encuentran en la placa dicha tapa se abre sin mayores problemas como se muestra en la *Figura 3.10*



***Figura 3.10*** Caja donde se coloca la placa principal y servirá para el panel de control.

En la *Figura 3.11* se observa el resultado final de la fuente de voltaje, donde se instaló las dos placas tanto para el funcionamiento del PIC como para los dos motores de pasos, que será ocupada para el sistema de automatización.



*Figura 3.11 Resultado final de la fuente de voltaje y la fuente de alimentación variable.*

### **3.3. EMPOTRAMIENTO DE LAS CAJAS DE LOS MOTORES PASO A PASO**

La caja de la placa principal está sujeta al marco mediante cuatro tornillos los mismos que hacen presión en cuatro postes que están colocados en el marco.

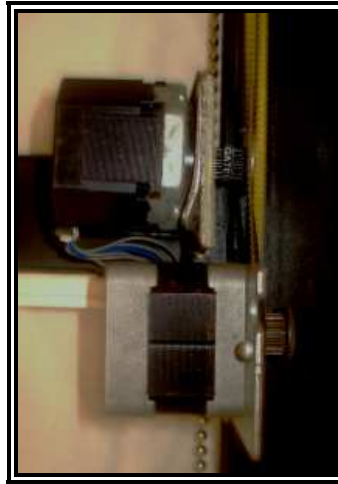
Las cajas donde están los opto - acopladores son sujetas con pegamento extra-fuerte en el riel de la persiana (*Figura 3.12*)



*Figura 3.12 Acoplamiento de la caja de los opto-acopladores en la persiana.*

Los motores van colocados como se indica en la *Figura 3.13* de tal modo que las dos cadenas de las persianas no se estorben una a la otra como tampoco moleste al movimiento de las tiras de las persianas.

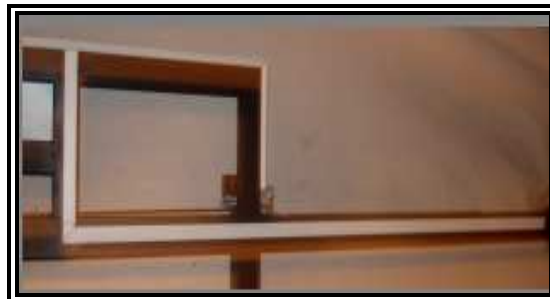
El motor que va en la parte superior es el de desplazamiento frontal/lateral y el motor que va debajo es el motor de desplazamiento horizontal.



*Figura 3.13 Acoplamiento de los motores de pasos en el marco.*

### **3.4. CABLEADO DEL SISTEMA DE AUTOMATIZACIÓN DE APERTURA Y CIERRE DE PERSIANAS VERTICALES.**

El cableado está constituido de dos cables UTP de 12 pares cada uno, las extensiones para la LDR y del LM35 y dos cables de alimentación. Todo el cableado está dentro de canaletas evitando así que los cables estén sueltos y expuestos a daños (*Figura 3.14*).



*Figura 3.14 Canaletas usadas para el cableado del sistema de automatización.*

Asimismo en la parte inferior de la caja se encuentran dos conectores que son las entradas de alimentación que de 5V (para el PIC) y de 6V (para los motores paso a paso). Como se indica en la *Figura 3.15*.



*Figura 3.15 Entradas de alimentación para el PIC y motores paso a paso.*

En la parte lateral izquierda de la caja existen tres orificios por donde ingresan los cables que provienen de los motores, de los sensores que están colocados en el riel de la persiana y las extensiones para el LM35 y la LDR respectivamente (*Figura 3.16*).



*Figura 3.16 Cables que ingresan al panel de control.*

### **3.5. ACABADO FINAL DEL SISTEMA DE APERTURA Y CIERRE DE PERSIANAS VERTICALES.**

En la *Figura 3.17* se puede observar como es el resultado final del proyecto. Listo para su perfecto desempeño funcional.





*Figura 3.17 Resultado final del sistema de apertura y cierre de persianas verticales.*

### **3.6. PRUEBAS Y RESULTADOS**

- Las primeras pruebas en el sistema de automatización fue la correcta ubicación de los motores paso a paso ya que según eso se tenía que programar el PIC para que los motores de pasos giren en sentido horario abriendo las persianas y en sentido antihorario cerrando las persianas.
- Otra prueba fue establecer un determinado recorrido de desplazamiento horizontal para cuando se presionen las teclas abrir o cerrar la persiana, siempre que la temperatura esté dentro del rango de medición.
- Una prueba difícil fue como detener el motor cuando la persiana ya se ha abierto o cerrado por completo. En un principio se colocó dos opto-acopladores a cada extremo de la persiana, el uno para provocar una interrupción y detener al motor mientras que el otro era para informar al PIC que la persiana está en su extremo respectivo, tomando en cuenta que si el uno estaba tapado el otro estaba descubierto. El problema que surgió es que el opto-acoplador encargado de la interrupción se quedaba tapado provocando que el PIC no ejecute ninguna otra opción y dando como resultado que el sistema de automatización quede fuera de servicio.

Debido al problema que se suscitó, fue necesario hacer cambios en el programa para no alterar físicamente la tarjeta principal. Y la solución fue simple, se sacó provecho a la exactitud del movimiento que poseen los motores PAP, para que por un determinado giro del motor el PIC cense al opto-acoplador que informa si la persiana se encuentra abierta o cerrada, si ninguno de estos opto-acopladores se encuentran obstaculizados por la tira de metal el motor paso a paso continua girando y censando a la vez. Mientras que en el programa se eliminó los comandos de interrupción, puesto que ya no se necesitan los opto-acopladores de interrupción.

De esta manera ahora se ocupa solo un opto-acoplador a cada extremo de la persiana, ahorrando espacio y mejorando la estética al mismo tiempo.

Con el cambio realizado el circuito ahora posee dos entradas en paralelo al puerto B.0 (compuerta lógica 7404) disponibles para cualquier implemento que se desee añadir.

- Después de haber buscado la ubicación idónea del micro-switch que es el que determina la apertura y cierre en forma frontal / lateral de la persiana, se llegó a la conclusión que debe ir en el primer seguro donde ingresa la tira de la persiana, puesto que este seguro no se desplaza y es firme en su puesto. El resultado obtenido es que cuando la persiana gira 90° en sentido antihorario el micro-switch es presionado, deteniendo inmediatamente el movimiento del motor.
- Se realizó una fuente de alimentación ocupando un solo transformador de 2 Amperios, el resultado fue que el PIC se reiniciaba al instante en que el motor paso a paso arrancaba, debido a que el motor consumía casi la totalidad de corriente que suministraba el transformador. La solución de este problema fue que cada fuente posea su propio transformador. Para el PIC un transformador de 1 Amperio y para los motores de pasos un transformador de 2 Amperios ambos a 12 voltios.

## **CAPÍTULO IV: CONCLUSIONES Y RECOMENDACIONES**

### **4.1. CONCLUSIONES:**

- Para controlar el avance y retroceso de la persiana vertical se usa motores paso a paso porque presentan un alto grado de exactitud de movimiento tanto en su arranque y en su parada.
- Con la precisión otorgada por el motor paso a paso no es necesario utilizar interrupciones para detener el movimiento del motor, cuando las tiras de la persiana lleguen al inicio o final de la misma.
- Si el valor del oscilador incrementa, la velocidad de ejecución por cada instrucción aumenta, en otras palabras el PIC se hará más rápido.
- El teclado del panel de control no ejecuta ninguna instrucción, cuando uno de los dos motores paso a paso esté en movimiento, ya que el programa se encuentra en un lazo cerrado de activación secuencial de las bobinas del motor.
- Para aumentar la velocidad de giro del motor paso a paso, se debe disminuir el tiempo de activación de bobina a bobina.
- Para que el motor posea más fuerza en su torque se configura en una secuencia de paso completo, pero se debe tener en cuenta que consume más corriente que en la secuencia de paso simple.

## 4.2. RECOMENDACIONES:

- La recomendación más relevante es tener cuidado al momento de conectar la alimentación a la placa principal ya que el voltaje de alimentación del motor es de 6VDC y si se lo conecta al PIC éste se dañará.
- Para el ahorro de pines del PIC16F877A se conecta solo 4 bits para la comunicación entre el PIC y el LCD 2X16, en lugar de utilizar los 8 bits de comunicación, con esta configuración el resultado es el mismo.
- No manipular las tiras de metal que se han colocado para presionar al micro-switch y obstaculizar al opto-acoplador, ya que estas tiras están calibradas para que desempeñen sus funciones correctamente.
- Como es un sistema automático en lo posible no manipular las persianas manualmente ya que se puede alterar su calibración.
- Evitar que el sensor de temperatura LM35 y la LDR estén expuestos al contacto del agua para que éstos no sufran daños físicos.
- Tener como repuesto básico un fusible de 1 Amperio y otro de 2 Amperios, si la fuente sufre un corto circuito o una sobre carga.

## BIBLIOGRAFÍA

### Libros:

1. Angulo, José Angulo, Ignacio.  
Microcontroladores PIC, Diseño Práctico de Aplicaciones, Tomo II  
Editora McGrawHill
2. BOYLESTAD, Robert L, Nashelsky Louis,  
Electrónica: Teoría de Circuitos y Dispositivos Electrónicos.  
Editorial Pearson Educación de México, S.A.,
3. COSTALES, Alcívar  
Apuntes de microcontroladores.
4. REYES, Carlos  
Aprenda rápidamente a programar microcontroladores.
5. ROBERT, F. Coughlin, Frederick F. Driscoll,  
Amplificadores operacionales y circuitos integrados lineales.  
Editorial Prentice Hall Hispanoamericana S.A.
6. W, García López, J. L. Gutiérrez Iglesias.  
Amplificadores Operacionales.  
Editorial ALCO, artes gráficas, Jaspe.34 -28026-MADRID.

### Direcciones de Internet:

7. Amplificador Operacional  
[http://html.wikipedia.com/amplificadores-operacionales\\_4.html](http://html.wikipedia.com/amplificadores-operacionales_4.html).  
[http://www.unicrom.com/tut\\_comparadores.asp](http://www.unicrom.com/tut_comparadores.asp).

## 8. Circuito impreso

[http://es.wikipedia.org/wiki/Circuito\\_impreso](http://es.wikipedia.org/wiki/Circuito_impreso)

<http://www.pablin.com.ar/electron/cursos/pcb/>

## 9. Clima en el Ecuador

[www. http://www.ecuaworld.ec/clima\\_ecuador.htm](http://www.ecuaworld.ec/clima_ecuador.htm)

<http://www.vivecuador.com/html2/esp/clima.htm>.

## 10. Imágenes del Amplificador operacional LM358

[www.national.com/datasheet-lm358/](http://www.national.com/datasheet-lm358/).

## 11. Light Dependent Resistor (LDR)

<http://es.wikipedia.org/wiki/LDR>.

## 12. Microcontroladores y Microprocesadores

[www. monografias.com/trabajos12/microco/microco.shtml](http://www.monografias.com/trabajos12/microco/microco.shtml)

<http://usuarios.lycos.es/sfriswolker7pic7uno.htm>.

## 13. Motor Paso a Paso

[http://www.emc.uji.es/d/IngMecDoc/344AmplDisMaq/Curso\\_02-](http://www.emc.uji.es/d/IngMecDoc/344AmplDisMaq/Curso_02-)

[03/webs\\_Alumnos/Web13\\_0203/MotoresPasoPaso.htm](http://www.emc.uji.es/d/IngMecDoc/344AmplDisMaq/Curso_02-03/webs_Alumnos/Web13_0203/MotoresPasoPaso.htm).

[http://72.14.203.104/translate\\_c?hl=es&u=http://www.eminebea.com/content/html/en/motor\\_list/pm\\_motor/pg251024.shtml&prev=/search%3Fq%3DMI%20NEBEA%2BCO.,%20BLTD.%20MOTORES%20BDE%20PASO%20THAILAND%26hl%3Des%26rlz%3D1G1GGLQ\\_ESEC254](http://72.14.203.104/translate_c?hl=es&u=http://www.eminebea.com/content/html/en/motor_list/pm_motor/pg251024.shtml&prev=/search%3Fq%3DMI%20NEBEA%2BCO.,%20BLTD.%20MOTORES%20BDE%20PASO%20THAILAND%26hl%3Des%26rlz%3D1G1GGLQ_ESEC254)

<http://www.todorobot.com.ar/informacion/tutorial%20stepper/stepper-tutorial.htm>.

## 14. Optoacoplador

<http://www.uv.es/marinjl/electro/opto.html>.

## 15. Pantalla de cristal líquido (LCD)

<http://www.x-robotics.com/lcd.htm>

## 16. PIC 16F877A

<http://ww1.microchip.com/downloads/en/DeviceDoc/39582b.pdf>.

## 17. Sensores

<http://es.wikipedia.org/wiki/Sensor>.

## 18. Sensor de temperatura LM35

<http://www.ucontrol.com.ar/wiki/index.php?title=LM35>.

<http://cgi.ebay.es/SENSOR-CONTROL-DE-TEMPERATURA-LM35-TO92>.

# **ANEXOS**



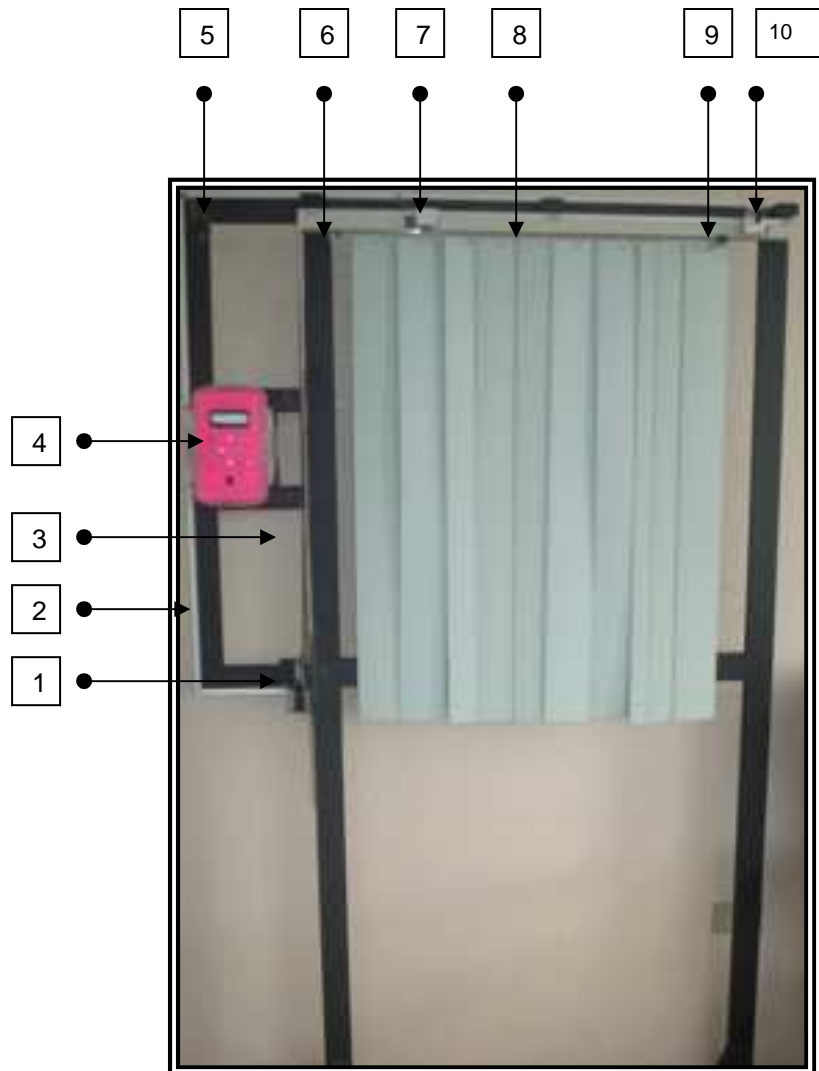
## Guía de Usuario

Esta guía de usuario explica de manera clara y sencilla, el principio de funcionamiento del sistema de automatización de apertura y cierre de persianas verticales, de tal modo que cualquier usuario pueda operarlo de manera correcta.

Tiene las siguientes características:

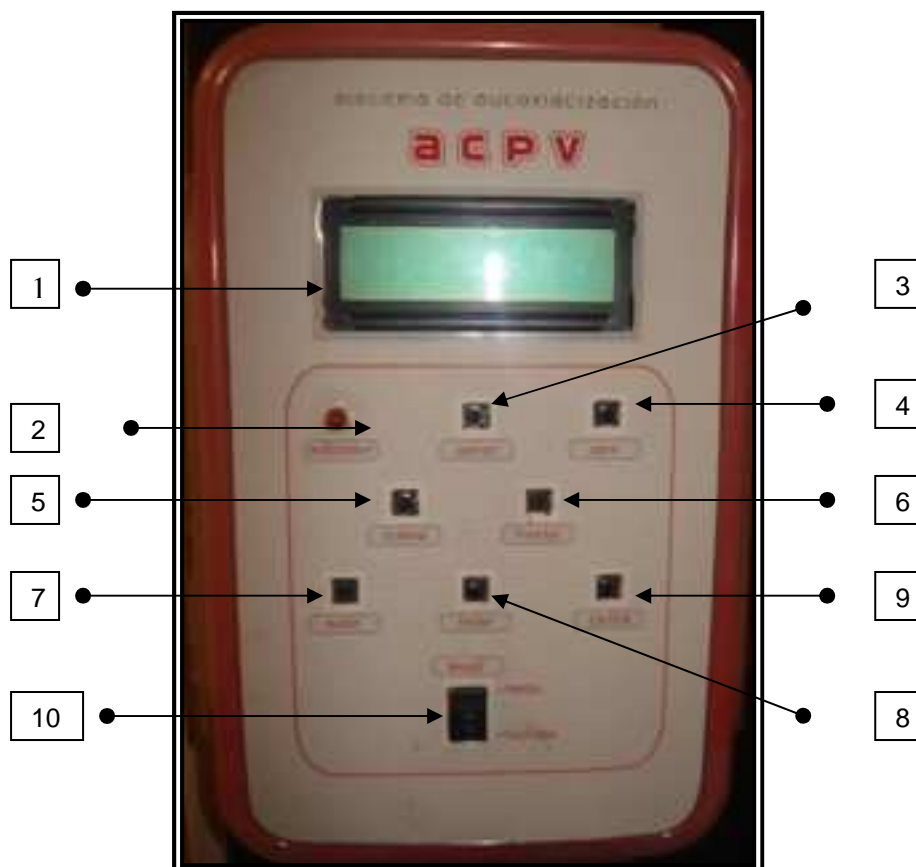
- Diseño compacto y liviano.
- Diseñado para que funcione automáticamente.
- Fácil de instalar y usar.
- Led indicador para las diferentes funciones.
- Los datos de la temperatura mínima, actual y máxima a visualizarse se la hacen por medio de un LCD 2 x 16.
- Por medio de un interruptor podemos cambiar la modalidad es decir, de modalidad de temperatura a modalidad LDR y viceversa.
- Además se lo puede emplear como un método de seguridad, ya que nos otorga una simulación de presencia.

## PARTES DEL SISTEMA DE AUTOMATIZACIÓN DE APERTURA Y CIERRE DE PERSIANAS VERTICALES



- 1: Dos motores paso a paso unipolares de 5 hilos.
- 2: Canaletas para el cableado.
- 3: Cadenas mixtas para el movimiento de las persianas.
- 4: Panel de control.
- 5: Sensores de temperatura (LM35) y la LDR.
- 6: Tira metálica de accionamiento del micro-switch.
- 7 y 10: Opto-acopladores prin y ulti respectivamente.
- 8: Riel de la persiana.
- 9: Tira metálica de accionamiento de ambos opto-acopladores.

## PARTES DEL PANEL DE CONTROL DE SISTEMA DE AUTOMATIZACIÓN A.C.P.V.



1: Pantalla LCD 2X16

2: Led indicador.

3: Tecla cerrar.

4: Tecla abrir.

5: Tecla lateral.

6: Tecla frontal.

7: Tecla subir.

8: Tecla bajar.

9: Tecla enter.

10: Tecla de modalidad Temperatura o Luz / oscuridad.

A continuación se describe las partes de panel de control:

- Tecla de modalidad: determina el modo de funcionamiento claridad / oscuridad o temperatura.
- Tecla enter: confirma el dato de temperatura mínima y máximo , a ser programado.
- Tecla subir: permite la variación del valor de la temperatura mínima. como también de la temperatura máxima, una unidad más por cada pulso.
- Tecla bajar: permite la variación del valor de la temperatura mínima. como también de la temperatura máxima, una unidad menos por cada pulso.
- Tecla frontal: permite colocar las persianas en posición frontal, evitando el ingreso de luz.
- Tecla lateral: permite colocar las persianas en posición lateral, permitiendo el ingreso de luz sin recorrer las persianas horizontalmente.
- Tecla cerrar: por cada pulso que se ejecute, la persiana se cierra una determinada distancia.
- Tecla abrir: por cada pulso que se ejecute, la persiana se abre una determinada distancia.
- Potenciómetro1: sirve para ajustar la sensibilidad de la LDR.
- Potenciómetro2: sirve para ajustar el contraste del LCD.
- Led indicador: indica el encendido del equipo, la ejecución de una orden al presionar una tecla.
- Entradas de alimentación: posee dos entradas de alimentación la de 5VDC es para la tarjeta electrónica, y la otra entrada es de 6VDC para la alimentación de los dos motores paso a paso.
- Sensor de temperatura: es el encargado de medir la temperatura ambiental, en el lugar donde se encuentre instalado.
- LDR: es el encargado de detectar si existe luz y/o oscuridad en el lugar donde está instalado.

**Calibración:**

Para el movimiento frontal o lateral de la persiana se debe tener en cuenta la tira de metal que se encuentra ubicada fija en el primer seguro de la izquierda, ésta debe activar el micro-switch en el momento que ha recorrido 90°, de no suceder este movimiento la cadena dentada ingresará a los engranajes internos del riel provocando el daño de los mismos. La calibración de esta tira de metal se la constata así:

- Cerrar la persiana en su totalidad.
- Fijarse que la tira de metal gire en el mismo sentido que las tiras de la persiana.
- Cerciorarse que la tira de metal esté alineada con el filo superior de las tiras de la persiana.

De darse con estas condiciones se concluye que la tira metálica está calibrada, de no ser así se procede a calibrar la tira metálica de la siguiente manera:

- Cerrada completamente la persiana, se forza suavemente intentando cerrar más la persiana. Con este procedimiento la tira metálica empieza a alinearse con los filos de las tiras de la persiana.

**Mantenimiento preventivo y correctivo:**

El mantenimiento preventivo lo puede realizar un técnico o una persona que no posea conocimientos electrónicos. Los pasos a seguir son los siguientes:

- Limpieza de los sensores de temperatura y de luz, como también de los opto-acopladores.
- Verificación de la calibración de la tira de metal.
- Evitar que en algún momento las cadenas mixtas se enreden entre sí.

Para el mantenimiento correctivo debe ser estrictamente realizado por personas autorizadas por el fabricante. Las posibles causas que pueden presentarse para dar un mantenimiento correctivo son los siguientes:

- Algún cable se haya roto.

La solución más adecuada es medir la continuidad en los cables de las dos fuentes tanto de la 5VDC que alimenta la placa principal y 6VDC que alimentan los dos motores de paso a paso, en los cables de la LDR y el sensor de temperatura, si están en corto o en mal estado lo más adecuado es cambiarlo por unos nuevos.

- Que la temperatura censada no sea la correcta y que el resto del sistema esté funcionando correctamente.

Lo más probable es que el sensor de temperatura esté dañado y sus características se hayan desvalorizado, o el potenciómetro que calibra la sensibilidad no está ajustado apropiadamente.

- Que el Led indicador muestre que el equipo se ha prendido pero que en el LCD no se visualice ningún dato.

Lo que se debe hacer es verificar si el cable de conexión del LCD se ha desconectado, si no funciona verificar si hay continuidad en cada uno de sus pines, o que el contraste que permite la visualización de datos esté bajado. Se debe ajustar el potenciómetro de manera que se pueda visualizar nuevamente los datos, pero si el problema persiste se debe reemplazar el LCD con otro que tenga las mismas características.

- Que el equipo no responda al momento de encenderlo.

Primeramente se debe verificar los fusibles de la fuente, si no están quemados por un sobrevoltaje, de ser así se los debe reemplazar por unos nuevos, y verificar que el voltaje de salida que alimenta la placa sea el correcto. Otra posibilidad es que el PIC se haya desconfigurado, lo mejor es llamar al técnico especializado.

- Que el movimiento de los motores paso a paso no sea continuo y se quede en un estado de vibración.

Este problema es provocado, debido a que uno de los transistores esté dañado, se lo debe reemplazar por uno nuevo.

- Que una tecla del panel de control no realice su función asignada.

Se debe revisar que la tecla esté funcionando, en las dos posiciones de abierto y cerrado, y verificar la resistencia que está en serie con la tecla no esté abierta o desvalorizada, comprobar que los cables que están soldados en los dos terminales de la tecla no estén rotos. Si algunos de los problemas mencionados anteriormente se presentan se los deben sustituir por elementos nuevos.

### **Recomendaciones:**

- Si los cables de alimentación están en mal estado o en corto se los debe reemplazar por otro cable nuevo.
- Una recomendación es limpiar el polvo, suciedad y humedad de los dos optoacopladores que se encuentran en la parte superior del riel, ya que esto puede obstaculizar el paso de la luz y no pueda ejecutar correctamente su función.
- No es muy recomendable manipular la persiana manualmente, pero si el caso lo amerita, se lo debe hacer muy delicadamente para no descalibrar la tira de metal que hace accionar al micro-switch o las cadenas mixtas que provocan el movimiento de la persiana.
- Preferible no manipular la calibración que se ha establecido en los potenciómetros del LCD y la LDR.

## Análisis Económico

En la construcción del equipo del sistema de automatización de apertura y cierre de persianas verticales se empleo dispositivos conocidos y accesibles en el medio, relativamente económicos para el usuario si desea adquirirlos.

En la *Tabla 4.1* se puede observar los correspondientes costos de los dispositivos y elementos empleados en la construcción del sistema, así como el valor de la mano de obra y el costro total.

<u>Item</u>	<u>Cantidad</u>	<u>Descripción</u>	<u>Total USD</u>
1	1	Amplificador operacional LM358	0.35
2	1	Amplificador operacional LM741	0.35
3	2	Bandas de caucho	3.00
4	1	Baquelita 10X20cm Fibra de vidrio	1.50
5	10	Borneras	3.15
6	1	Caja de plástica para la placa de principal	2.00
7	1	Caja de proyecto para la fuente	3.50
8		Cable de poder	1.60
9	4	Condensadores de cerámica	0.12
10	5	Condensadores electrolíticos	0.65
11	1	Cristal de cuarzo de 4 MHz	0.80
12	2	Cloruro férrico	1.20
13	8	Diodos 1N4007	0.48
14	1	Diodo zener de 15VDC a 400MW	0.12
15	2	ECG 3100	2.40
16	3 metros	Espagueti	0.60
17	2	Fusible 2 Amperios a 250VAC	0.10
18	2	Fusible 2.5 Amperios a 250 VAC	0.10
19	1	Interruptor 2.5 Amperios a 250 VAC	1.05



20	3	Leds	0.45
21	1	LDR	0.80
22	1	LCD 2X16	16.00
23	1	Marco de Hierro	40.00
24	2	Motores paso a paso unipolar de 5 hilos	10.00
25	1	Papel azul termo transferible	1.80
26	1	Perrilla para la fuente regulada	0.35
27	1	PIC 16F877A	6.75
28	1	Potenciómetro de 50 K $\Omega$	0.35
29	3	Potenciómetros de arañas	1.05
30	3	Porta-fusibles	0.15
31	2	Puentes de Diodos	2.40
32	1	Regulador de Voltaje L7805C-V	0.80
33	1	Riel de Persiana	15.00
34	33	Resistencias de ¼ w	0.93
35	1	Sensor de temperatura LM35	2.25
36	1	Transformador de 12VDC a 1 Amperio	4.50
37	1	Transformador de 12VDC a 2 Amperios	5.50
38	7	Teclas normalmente abiertas	0.56
39	1	Transistor ECG 373	0.50
40	1	Transistor ECG 130	0.80
41	8	Transistores TIP122	4.80
42	2	Transistor 2N3906	0.40
43		Mano de Obra	100.00
		<b>TOTAL:</b>	<b>239.21</b>

**Tabla 4.1 Costos parciales y totales del sistema de utomatización de apertura y cierre de persianas verticales.**

De acuerdo a los costos totales y los beneficios que proporciona el sistema de automatización de apertura y cierre de persianas verticales se concluye que es accesible para el personal técnico y público en general que lo desea fabricar.