

88



UNIVERSIDAD NACIONAL AUTONOMA DE MEXICO

ESCUELA NACIONAL DE ESTUDIOS PROFESIONALES "CAMPUS ARAGÓN"

"IMPLEMENTACIÓN DE UN CIRCUITO DE CONTROL DE TEMPERATURA CON EL PIC16F84"

T E S I S

QUE PARA OBTENER EL TITULO DE:
INGENIERO MECANICO ELECTRICISTA
AREA: ELECTRICA Y ELECTRONICA
P R E S E N T A N:
VÍCTOR MESSINA LÓPEZ
MANUEL ANTONIO NAVARRO VITE

ASESOR: ING. ELEAZAR MARGARITO PINEDA DIAZ



MEXICO

TESIS CON FALLA DE ORIGEN

2002



Universidad Nacional
Autónoma de México

Dirección General de Bibliotecas de la UNAM

Biblioteca Central



UNAM – Dirección General de Bibliotecas
Tesis Digitales
Restricciones de uso

DERECHOS RESERVADOS ©
PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL

Todo el material contenido en esta tesis esta protegido por la Ley Federal del Derecho de Autor (LFDA) de los Estados Unidos Mexicanos (México).

El uso de imágenes, fragmentos de videos, y demás material que sea objeto de protección de los derechos de autor, será exclusivamente para fines educativos e informativos y deberá citar la fuente donde la obtuvo mencionando el autor o autores. Cualquier uso distinto como el lucro, reproducción, edición o modificación, será perseguido y sancionado por el respectivo titular de los Derechos de Autor.

**A nuestros Padres y Madres
con eterno agradecimiento.**

**A nuestros hermanos
con cariño y respeto.**

A la Universidad Nacional Autónoma de México.

**A los Ingenieros Eleazar M. Pineda Díaz y Enrique Herrera Hidalgo por su
valiosa y decisiva colaboración en este proyecto de tesis.**

Gracias a los maestros por su labor de educación y formación profesional.

CAPITULADO.

-INTRODUCCIÓN.	1
-CAPÍTULO I. CARACTERÍSTICAS DE LOS MICROCONTROLADORES.	4
-1.1. QUE ES UN MICROCONTROLADOR.	4
-1.2. DIFERENCIAS ENTRE MICROPROCESADOR Y MICROCONTROLADOR.	6
-1.3. ARQUITECTURA INTERNA.	7
1.3.1. EL PROCESADOR.	8
1.3.2. MEMORIA DE PROGRAMA.	9
1.3.3. MEMORIA DE DATOS.	11
1.3.4. LÍNEAS DE ENTRADA Y SALIDA.	11
1.3.5. RECURSOS AUXILIARES.	11
-1.4. DIAGRAMA DE CONEXIONES.	12
1.4.1. FUNCIÓN DE LOS PINES DEL PIC16X84.	13
1.4.2. MEMORIA DE PROGRAMA.	13
1.4.3. MEMORIA DE DATOS RAM.	15
1.4.4. MEMORIA DE DATOS EEPROM.	17
1.4.5. PUERTOS DE ENTRADA Y SALIDA.	17
1.4.6. REGISTRO DE CONTROL.	18
-1.5. DATOS DE FABRICANTE.	23
1.5.1. FRECUENCIA DE FUNCIONAMIENTO.	25
-CAPÍTULO II. PROGRAMACIÓN DEL MICROCONTROLADOR.	28
-2.1. PROGRAMAS Y LENGUAJES DE PROGRAMACIÓN.	28
-2.2. SECUENCIA PARA EL DESARROLLO DE PROGRAMAS.	29
-2.3. EL CÓDIGO FUENTE.	29
-2.3.1. ORGANIZACIÓN DEL CÓDIGO FUENTE.	30
-2.3.2. FORMATOS DE INSTRUCCIÓN.	31
-2.4. DIRECTIVAS.	32
-2.5. ETIQUETAS.	35
-2.6. JUEGO DE INSTRUCCIONES.	35
2.6.1. DISEÑO DE LAS INSTRUCCIONES.	36
2.6.2. INSTRUCCIONES QUE MANEJAN REGISTROS.	37
2.6.3. INSTRUCCIONES QUE MANEJAN BITS.	42

2.6.4. INSTRUCCIONES DE SALTO.	43
2.6.5. INSTRUCCIONES QUE MANEJAN OPERANDOS INMEDIATOS.	45
2.6.6. INSTRUCCIONES DE CONTROL Y ESPECIALES.	47
-2.7. ENSAMBLADO DE PROGRAMAS.	49
2.7.1. ENSAMBLADO DE DOS PASOS.	50
CAPÍTULO III. DISEÑO E IMPLEMENTACIÓN DEL CIRCUITO DE CONTROL DE TEMPERATURA.	53
-3.1. INTRODUCCIÓN.	53
-3.2. DIAGRAMA A BLOQUES Y FUNCIONES.	53
-3.3. SELECCIÓN DE COMPONENTES Y OPCIONES.	55
3.3.1. SELECCIÓN DEL SENSOR.	55
3.3.2. SELECCIÓN DEL COMPARADOR.	59
3.3.3. SELECCIÓN DEL ACOPLAMIENTO DE ENTRADA.	62
3.3.4. SELECCIÓN DEL CONTROL.	65
3.3.5. SELECCIÓN DEL ACOPLAMIENTO DE SALIDA.	67
3.3.6. SELECCIÓN DE LA ETAPA DE POTENCIA.	70
3.3.7. SELECCIÓN DE LA CARGA.	73
-3.4. SEÑAL DE ENTRADA.	74
3.4.1. ESCALAS DE TEMPERATURA.	74
-3.5. ETAPA SENSORA.	76
3.5.1. SENSOR DE TEMPERATURA LM335.	76
3.5.2. FUNCIONAMIENTO.	77
3.5.3. DIAGRAMA.	78
-3.6. ETAPA COMPARADORA.	78
3.6.1. AMPLIFICADOR OPERACIONAL TL082.	78
3.6.2. FUNCIONAMIENTO.	78
3.6.3. DIAGRAMA.	80
-3.7. ETAPA DE ACOPLAMIENTO DE ENTRADA.	80
3.7.1. OPTOACOPLADOR 4N33.	80
3.7.2. FUNCIONAMIENTO.	80
3.7.3. DIAGRAMA.	81
-3.8. ETAPA DE CONTROL.	82
3.8.1. MICROCONTROLADOR PIC16F84.	82
3.8.2. FUNCIONAMIENTO.	82
3.8.3. DIAGRAMA.	84

-3.9. ETAPA DE ACOPLAMIENTO DE SALIDAS.	85
3.9.1. OPTOACOPLADOR MOC 3022.	85
3.9.2. FUNCIONAMIENTO.	85
3.9.3. DIAGRAMA.	85
-3.10. ETAPA DE POTENCIA.	86
3.10.1. TRIAC MAC218.	86
3.10.2. FUNCIONAMIENTO.	86
3.10.3. DIAGRAMA.	87
-3.11. CARGA.	87
3.11.1. MOTOR MONOFÁSICO DE RELUCTANCIA.	87
3.11.2. FUNCIONAMIENTO.	88
3.11.3. DIAGRAMA.	89
-3.12. DIAGRAMA GENERAL.	90
-3.13. PRUEBAS.	91
3.13.1. PRUEBA DE LA ETAPA SENSORA.	91
3.13.2. PRUEBA DE LA ETAPA COMPARADORA.	91
3.13.3. PRUEBA DE LA ETAPA ACOPLADORA DE ENTRADA.	92
3.13.4. PRUEBA DE LA ETAPA DE CONTROL.	93
3.13.5. PRUEBA DE LA ETAPA ACOPLADORA DE SALIDA.	94
3.13.6. PRUEBA DE LA ETAPA DE POTENCIA.	95
 CAPÍTULO IV. PROGRAMA Y GRABACIÓN.	 96
-4.1. PROGRAMA.	96
4.1.1. ALGORITMO.	97
4.1.2. LISTADO DEL CÓDIGO FUENTE.	98
4.1.3. LISTADO DE LA LIBRERÍA PIC16F84.	103
4.1.4. ENSAMBLADO DEL PROGRAMA.	107
4.1.5. EL ARCHIVO DE ERRORES.	108
4.1.6. LISTADO DEL ARCHIVO HEXADECIMAL.	109
-4.2. GRABACIÓN.	109
4.2.1. CIRCUITO GRABADOR.	110
4.2.2. PROGRAMA GRABADOR.	111
 CONCLUSIONES.	 122
 BIBLIOGRAFIA	 124

INTRODUCCIÓN.

Actualmente la presencia de los microcontroladores se encuentra en la gran mayoría de los sectores industriales, comerciales y en el hogar. Pero su empleo masivo dentro de la electrónica apenas comienza.

Estos dispositivos de bajo costo, alta confiabilidad y tamaño reducido, se han introducido de una manera exponencial ya que se encuentran en los hornos de microondas, en los teclados y ratones de las computadoras, en las alarmas, en los televisores, en los teléfonos, en fin en una gran variedad de artículos. Las áreas existentes para su aplicación son ilimitadas y en un futuro gobernarán la mayoría de los aparatos que utilizamos.

Dentro de la problemática existente en espacios cerrados dentro de escuelas, oficinas, locales comerciales e industriales por la gran cantidad de personas que se encuentran y además del factor climático cambiante en las diferentes etapas del año; los cuales están relacionados directamente con el desempeño de las personas al realizar sus actividades e influyendo en la vida útil de una amplia variedad de productos y servicios ya sean perecederos (temperatura de almacenamiento de alimentos), eléctricos (temperatura de motores), electrónicos (temperatura de operación de los circuitos integrados), mecánicos (temperatura de materiales), químicos (temperatura de reacciones químicas), agrícolas (temperatura de invernaderos), médicos (temperatura de incubadoras) es imprescindible el uso de sistemas de control de temperatura.

Los problemas anteriormente descritos pueden ser resueltos adaptando el controlador de temperatura según las necesidades de cada aplicación, esto gracias a que el rango de temperatura que el circuito maneja, puede ser muy pequeño (de 1°C) ó muy grande (de 140°C), la amplitud del rango se debe a las características del sensor empleado. Dadas las reducidas dimensiones del controlador de temperatura, su instalación no representa ningún inconveniente al usuario ya que puede ser colocado en cualquier espacio disponible.

La aplicación que se tuvo en mente al inicio del diseño fue que el circuito funcionara como un control de temperatura de espacios cerrados, donde la variable física a controlar es la temperatura ambiente; por ejemplo, que regule la temperatura de cuartos, habitaciones, salones, etc.

Este trabajo de tesis tiene como objetivo: diseñar un circuito electrónico que emplee un microcontrolador PIC16F84, este circuito tiene como finalidad sensar la temperatura y a la vez mantenerla constante dentro de un determinado rango, este rango debe ser previamente establecido por el usuario.

La tesis trata sobre el diseño e implementación del circuito antes descrito, con el objetivo fundamental de que este tenga un bajo costo y que además cumpla con las características siguientes: pocos componentes, implementación sencilla, tamaño reducido, facilidad de operación y una buena precisión.

La metodología a seguir para el desarrollo del tema de tesis consistirá en elaborar un diagrama a bloques, dentro de este figuran cada una de las etapas que conformarán al circuito; una vez que se tiene el diagrama, cada bloque es diseñado e implementado según la función que deba realizar dentro del circuito.

En el primer capítulo se expone el funcionamiento, arquitectura y recursos fundamentales del microcontrolador, como son los temporizadores, puertos de entrada y salida, registros y la memoria de datos de los que están constituidos los microcontroladores PIC, centrándonos fundamentalmente en la gama mediana a la cual pertenece el PIC16F84, que es el empleado en esta tesis por diversas razones: sencillez, confiabilidad, rapidez, costo y la función de escritura y borrado del programa, el cual soporta un innumerable ciclo de estas dos funciones.

El segundo capítulo es referido a la programación del microcontrolador, abordando temas tales como: que es un programa y un lenguaje de programación, secuencia para el desarrollo de un programa, elaboración del código fuente, empleo de directivas y etiquetas, el juego de instrucciones con el que cuenta el PIC16F84 y finalizando con una breve explicación sobre el ensamblado del programa.

En el tercer capítulo se trata el diseño e implementación del circuito de control de temperatura, aquí se encuentra el diagrama a bloques del circuito, selección de componentes para cada una de las etapas dentro de una variedad de opciones y el diseño de cada etapa, terminando el capítulo con el diagrama general del circuito y las pruebas respectivas.

El último capítulo se refiere al programa, en su formato de código fuente (lenguaje ensamblador) y código máquina (lenguaje de máquina), además de presentar como se realiza paso a paso el ensamblado del programa y la grabación del mismo.

CAPÍTULO I.

CARACTERÍSTICAS DE LOS MICROCONTROLADORES.

1.1. QUE ES UN MICROCONTROLADOR.

Los circuitos integrados programables PIC (Programmable Integrated Circuits) son en si, una computadora, ya que cuentan con todos los elementos que esta contiene. Se emplean para el control y funcionamiento de una tarea determinada y, gracias a su pequeño tamaño, están incorporados en el propio dispositivo al que gobiernan. Por esta ultima razón, también se le denomina controlador incrustado (embedded controller).

Su memoria consta de un programa destinado a realizar una aplicación determinada; sus líneas de entrada / salida (I / O) soportan la conexión de los sensores y actuadores del dispositivo a controlar y sus demás componentes complementarios de que dispone tienen la finalidad de atender dichos requerimientos. Una vez que se le configura y programa, el microcontrolador solo sirve para desarrollar la tarea asignada.

Los aparatos que funcionan en base a uno o varios microcontroladores están en amplio y rápido crecimiento. Con lo que es posible pronosticar que en el siglo XXI habrá muy pocos elementos electrónicos que no cuenten con un microcontrolador.

Los periféricos de la computadora; el ratón, el teclado y la impresora, son controlados por el programa de un microcontrolador.

Los electrodomésticos de línea blanca (hornos, lavavajillas, lavadoras, etc.) y los aparatos electrónicos (televisiones, estereos, videos, etc.) constan de numerosos microcontroladores. Igualmente, los sistemas de vigilancia y alarma en edificios constan de estos chips en la optimización y rendimiento de ascensores, aire acondicionado, calefacción, alarmas de incendio o robo, etc.

En las comunicaciones y sus sistemas de transferencia de información son profundamente utilizados, incorporándolos en los grandes automatismos y en los modernos teléfonos.

Una industria de suma importancia para estos chips es la automotriz, la cual los emplea en el control de funciones tan populares como la climatización, la seguridad interior y los frenos ABS.

Las diferentes aplicaciones que se le puede dar a un microcontrolador están: telefonía, hogar, sector automotriz, computación y en la industria; donde cada uno de ellos ocupa cierto porcentaje de la producción total. Las aplicaciones y los valores de porcentaje se muestran en la Figura 1.1

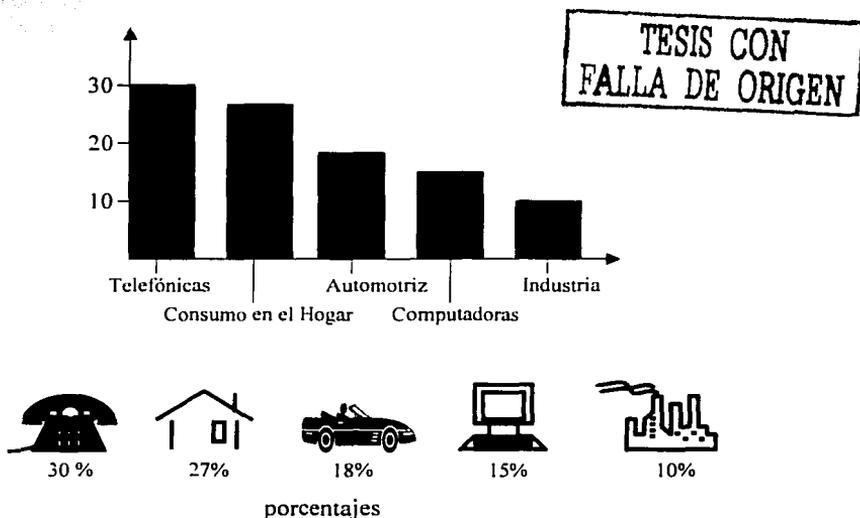


Figura 1.1. Grafica que ilustra la distribución por sectores de la producción mundial de microcontroladores en las diversas áreas de aplicación.

Como se observa en la grafica las comunicaciones y los productos de consumo en el hogar consumen mas de la mitad de la producción de microcontroladores. El resto es distribuido entre la industria, las computadoras y el sector automotriz.

1.2. DIFERENCIAS ENTRE MICROPROCESADOR Y MICROCONTROLADOR.

El microprocesador es un circuito integrado IC (integrated circuit) que contiene la Unidad Central de Procesos CPU (central processing unit), conocida como procesador, de la computadora. El CPU esta conformada por la unidad de control, que interpreta las instrucciones, y el cambio de datos, que las ejecuta.

Los pines de un microprocesador mandan al exterior las líneas de sus buses de dirección, datos y control; para poder conectarse con la memoria y los módulos I/O logrando la configuración de la computadora implementada con varios circuitos integrados.(Figura 1.2.)

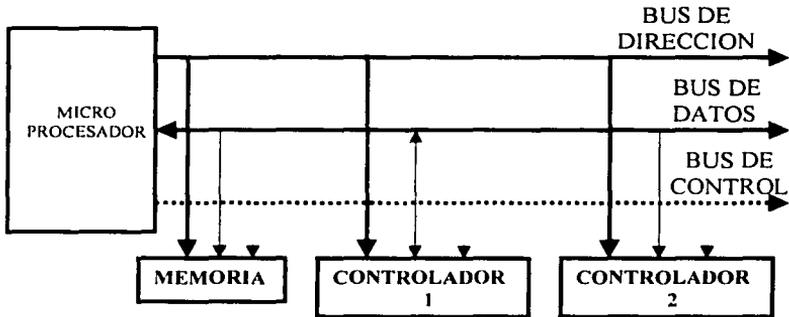


Figura 1.2.Estructura interna de un microprocesador.

Por lo tanto un microprocesador es solamente la Unidad Central de Procesos (CPU), en la cual todos sus periféricos son externos. El microcontrolador tiene integrado el CPU y todos los periféricos en un solo chip. Con lo cual la programación es el aspecto importante a realizar, al no interesarse por una gran cantidad de dispositivos exteriores. (Figura 1.3.)



Figura 1.3. Estructura interna de un microcontrolador.

1.3. ARQUITECTURA INTERNA.

Como ya se explico, el microcontrolador posee todos los componentes de una computadora, pero sus instrucciones son fijas y no pueden ser alteradas.

Las partes principales de un microcontrolador son:

1. Procesador
2. Memoria no volátil para almacenar el programa principal
3. Memoria de lectura y escritura para guardar los datos de la aplicación
4. Pines de I/O para los controladores periféricos:
 - a) Comunicación en paralelo
 - b) Comunicación en serie
 - c) Diversas puertos de comunicación con protocolos
5. Recursos auxiliares
 - a) Circuito de reloj
 - b) Temporizadores
 - c) Perro Guardián (Wachtdog)
 - d) Conversores AD y DA (analógico/digital, digital/analógico)
 - e) Comparadores analógicos
 - f) Protección ante fallas de la alimentación
 - g) Estado de reposo o de bajo consumo

A continuación se describirán las partes mas importantes de un microcontrolador.

1.3.1. EL PROCESADOR.

Existen dos tipos de arquitectura en los procesadores. La arquitectura Harvard y la Von Neumann. Con la necesidad de elevar los rendimientos en el procesamiento de las instrucciones, se ha generalizado el empleo del primero.

La arquitectura del Von Neumann se caracteriza porque el CPU se conecta con una memoria única, donde datos e instrucciones coexisten por medio de un sistema de buses. (Figura 1.4.)

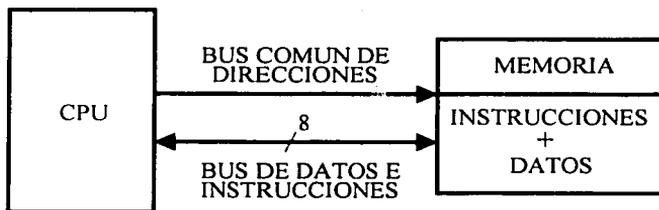


Figura 1.4. En la arquitectura de Von Neumann el CPU se comunica por medio de un sistema de buses con la memoria, lugar donde se guardan las instrucciones y datos.

En la arquitectura Harvard la memoria de instrucciones y la memoria de datos son independientes y cada una dispone de su propio sistema de buses para el acceso. Con esta configuración existe un paralelismo, el cual permite la adecuación del tamaño de las palabras y los buses a los requerimientos específicos de las instrucciones y los datos. Por lo tanto la capacidad de memoria también es diferente. (Figura 1.5.)

Otra característica de los microcontroladores es el uso de un conjunto de instrucciones del tipo de Computadoras de Juego de Instrucciones Reducido o RISC (reduced instruction set computer), el cual posee un número de instrucciones máquina simple y pequeño, de esta forma la mayor parte de las instrucciones se ejecuta en un ciclo de instrucción.

El paralelismo implícito aumenta el rendimiento de la computadora, el cual consiste en la segmentación del procesador (pipe line), descomponiéndolo en etapas para poder procesar una instrucción diferente, con una sola o con varias a la vez.

Este alto rendimiento y velocidad que logran los procesadores, como el que contiene los microcontroladores PIC, se debe a las tres técnicas en conjunción.

- Arquitectura Harvard
- Arquitectura RISC
- Segmentación

1.3.2. MEMORIA DE PROGRAMA.

La memoria de programa de un microcontrolador está diseñada para almacenar todas las instrucciones del programa control. La posibilidad de utilizar memorias externas de ampliación queda anulada.

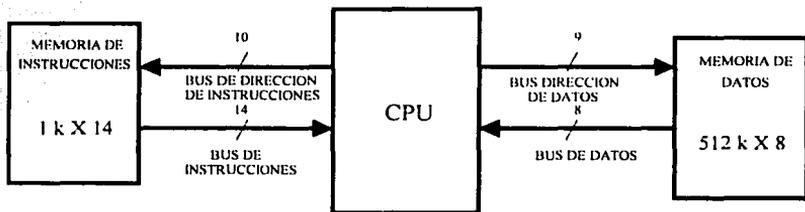


Figura 1.5. En la arquitectura Harvard la memoria de instrucciones y de datos son independientes, lo que hace optimizar sus características y proporciona el paralelismo. La figura muestra la memoria de instrucciones tiene 1024 de posiciones de 14 bits cada una, mientras que la de datos solo es de 512 posiciones de 1 byte.

Como el programa a ejecutar siempre será el mismo, es grabado en forma permanente. Existen cinco versiones de memoria, que son adecuados para soportar esta función.

1) Memoria de Solo Lectura ROM (read only memory) con máscara.

En este tipo de memoria, el programa es grabado durante el último proceso de producción de la unidad. Este procedimiento es costoso por lo que el productor carga un costo extra al cliente.

2) Memoria de Solo Lectura Programable y Borrable EPROM (erasable programmable read only memory).

Esta memoria puede reestructurarse a su valor inicial (todos 0 o todos 1) aun cuando se hayan cambiado previamente. Cuando una EPROM se coloca bajo luz ultravioleta especial por un período de tiempo, la radiación de onda corta descarga las compuertas internas que sirven como contactos. Después de borrar la memoria esta puede volver a emplearse.

3) Programable Una Vez OTP (one time programing).

Esta memoria como su nombre lo indica solo puede ser grabada una vez por el usuario, y utiliza el mismo método que la EPROM para su borrado. Una vez realizado este procedimiento no se podrá volver a borrar. Su precio es bajo y la sencillez de la grabación es recomendada para prototipos finales.

4) Memoria de solo Lectura Programable y Borrable Eléctricamente EEPROM (electrical erasable programmable read only memory)

La grabación es parecida a la de la OTP y la EPROM, pero el borrado es mas sencillo al efectuarse de la misma forma que el grabado, o sea eléctricamente. Esta puede ser grabada y borrada tantas veces como se quiera.

5) FLASH.

Esta es una memoria no volátil, de bajo consumo, que se puede escribir y borrar en circuito igual a la EEPROM, pero dispone de mayor capacidad que estas últimas. El borrado es posible solo con bloques completos y no se puede realizar sobre posiciones concretas. Esta memoria esta sustituyendo a la memoria EEPROM debido a sus mejores prestaciones para contener información.

En la actualidad el fabricante Microchip esta comercializando dos microcontroladores prácticamente iguales, que se diferencian en que la memoria de programa en uno de ellos es de tipo EEPROM y la del otro tipo FLASH. Se trata del PIC16C84 y el PIC16F84, respectivamente. El primero esta actualmente descontinuado en México.

1.3.3. MEMORIA DE DATOS.

En los programas, los datos que se manejan varían continuamente y por esto se le exige a la memoria que los contiene leerlos y escribirlos, por lo que la memoria RAM estática (SRAM) es la mas adecuada, aunque sea volátil.

Existen microcontroladores que cuentan con memoria de datos que se divide en una de lectura y una de escritura no volátil, del tipo EEPROM. De esta manera, al haber un corte en suministro de la alimentación no ocasiona la perdida de la información, que esta disponible al reiniciarse el programa. El PIC16C84, el PIC16F83 y el PIC16F84 cuentan con 64 bytes de memoria EEPROM.

1.3.4. LINEAS DE ENTRADA Y SALIDA PARA LOS CONTROLADORES DE PERIFÉRICOS.

Todos los pines de un microcontrolador a excepción de dos que están destinadas a la alimentación, otros dos para el reloj que regula la frecuencia de trabajo, y una mas para el Reset; sirven para soportar su comunicación con los periféricos externos que controla.

En los microcontroladores las líneas I/O que se adaptan con los periféricos manejan información en paralelo, las cuales se agrupan en conjuntos de ocho, que reciben el nombre de Puertas. Hay modelos con líneas que soportan la comunicación en serie; otros conjuntos disponen de líneas que implementan puertas de comunicación para distintos protocolos, como el I²C, el USB, etc.

1.3.5. RECURSOS AUXILIARES.

El microcontrolador según el fabricante, orienta un sin numero de complementos, que refuerzan la potencia y flexibilidad del dispositivo. Entre los recursos mas comunes se citan los siguientes:

- a) Circuito de reloj, es el encargado de generar los impulsos que sincronizan el funcionamiento de todo el sistema.
- b) Temporizadores, orientados a controlar tiempos.
- c) Perro Guardián, (watchdog), provoca una reinicialización cuando el programa queda bloqueado.
- d) Convertidores AD y DA, pueden recibir y enviar señales analógicas.
- e) Comparadores analógicos, verifican el valor de una señal analógica.
- f) Sistema de protección ante fallos de la alimentación.
- g) Estado de reposo, en este estado el sistema queda “congelado” y el consumo de energía se reduce al mínimo.

1.4. DIAGRAMA DE CONEXIONES.

La distribución y denominación de las patas del PIC16X84, que es el modelo de microcontrolador que se utilizara en esta tesis se presenta en la Figura 1.6. y serán analizados en los siguientes subtemas.

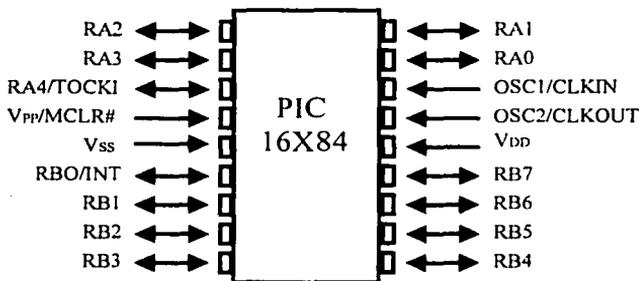


Figura. 1.6. Distribución y denominación de las patas del PIC16X84.

1.4.1. FUNCION DE LOS PINES DEL PIC16X84

OSC1/CLK: Entrada externa de los impulsos de reloj o conexión con el cristal de cuarzo.

OSC2/CLKOUT: Salida de Fosc/4 en modo oscilador R-C o conexión con el cristal de cuarzo.

VPP/MCCLR#: En modo de grabación, se introduce por este pin la tensión.

VPP (12 – 14 V DC). En su funcionamiento normal este pin soporta la función de Reset.

RA0 – RA3: Líneas de I/O del puerto A.

RA4/T0CKI: Líneas de I/O del puerto A o entrada de impulsos de reloj para TMRO.

RBO/INT: Línea de I/O del puerto B o petición de interrupción.

RBI – RB7: Líneas de I/O del puerto B.

V_{DD}: Pin conectado al Voltaje de Alimentación.

V_{DD}: Pin conectado a tierra o negativo de la alimentación.

1.4.2. MEMORIA DE PROGRAMA.

Esta memoria tiene implementadas 1024 palabras de 14 bits cada una, ocupando las direcciones comprendidas entre 0000 y la 03FF. Estos modelos llamados de línea media alcanzan 8 K de capacidad máxima. Figura 1.7.

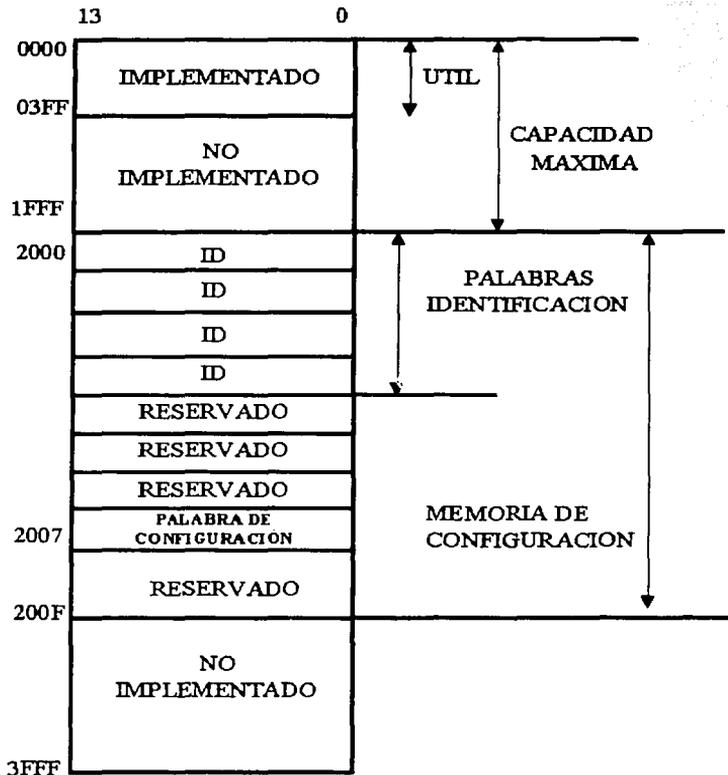


Figura 1.7. Organización de la memoria de instrucciones de los PIC16X84.

Las posiciones de identificación ID sirven para que el usuario grave en sus 4 bits de menos peso los códigos que se deseen.

La palabra de configuración tiene solo 5 bits validos y sus simbolos y funciones se muestran en la Figura 1.8.

13	4	3	2	1	0
NO DEFINIDOS	CP	PWRTE	WDTE	FOSC1	FOSC0
<p>CP: Protección de código (CP = 0 activado)</p> <p>PWRTE: Control de temporizador "POWER-UP" (En el PIC16C84 PWRTE = 1, activado, en el PIC16F84 al revés).</p> <p>WDTE: Control del Perro Guardián (WDTE = 1, activado)</p> <p>FOSC1-FOSC0: Selección oscilador:00:LP, 10:HS, 11:R-C</p>					

Figura 1.8. Palabra de configuración.

1.4.3. MEMORIA DE DATOS RAM.

La memoria tiene dos bancos de 128 bytes cada uno. En el PIC 16C84 tan solo están implementadas las primeras 48 direcciones de cada banco. En las primeras 12 se encuentran los registros de propósito específico (SFR) y en las 36 siguientes los registros de propósito general (GPR).

No obstante los 36 registros de propósito general del banco 1 se mapean sobre el banco 0, dejando reducidos a 36 los registros operativos de este tipo. En el PIC16F84 la memoria RAM es muy parecida, pero los registros de propósito general pueden ser hasta 68 en cada banco, siendo efectivos solo 68, con lo que los del banco 1 se mapean sobre el banco 0. Figura 1.9.

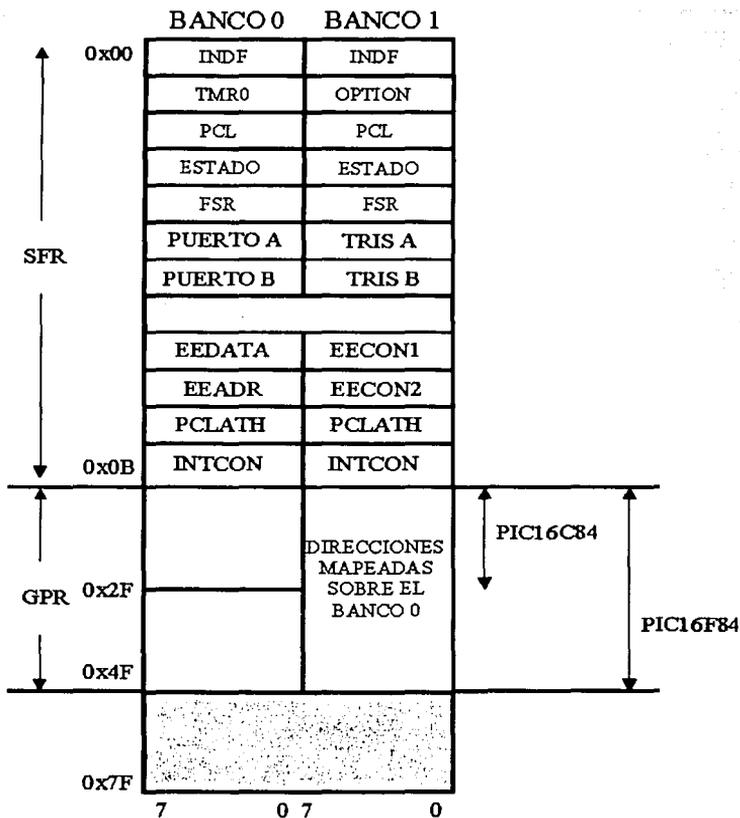


Figura 1.9. Organización de la memoria de datos RAM. En el PIC16C84 hay 36 registros de propósito general, en tanto que en el PIC16F84 existen 68.

1.4.4. MEMORIA DE DATOS EEPROM.

Los datos que se guardan en esta memoria no se pierden al quitar la alimentación. Se compone de 64 bytes que ocupan direcciones que van desde la 0x00 y la 0x3F.

Para entrar a la memoria en lectura o escritura hay que utilizar los registros de control: EEDATA, EEADR, EECON1 y EECON2.

El proceso de escritura en esta memoria de una posición es lento y dura aproximadamente 10 ms. Cuando termina se activa un señalizador y se puede ocasionar una interrupción.

1.4.5. PUERTOS DE ENTRADA Y SALIDA.

Son dos puertos de I/O: puerto A y puerto B.

PUERTO A.

- Contiene 5 líneas denominadas RA4-RA0.
- RA3-RA0 actúan como líneas de I/O digitales configurables de forma independiente.
- RA4/T0CKI, tienen dos funciones, como líneas de I/O, o como entrada de impulsos de reloj para el TMRO.
- Si a un bit de TRISA se pone en 1, la línea de I/O correspondiente de la puerto A sirve como entrada y si se le pone a 0 sirve como salida.
- Cuando se lee una entrada se lee el estado actual de la pata.
- Las líneas de salida están lacheadas y en estas se mantiene el último valor que se ha sacado.
- Después de un Reset todos los bits de TRISA se ponen en 1 y las líneas del puerto A quedan configuradas como entradas.

- En el puerto A cada línea suministra una corriente máxima de 20 mA y puede soportar hasta 25 mA. La corriente máxima que puede soportar la puerto A es de 80 mA y la que puede dar es de 50 mA.

PUERTO B.

- Las líneas del puerto B se pueden configurar como entradas, con esto es posible acoplarlas a una resistencia "pull-up" al dispositivo, poniendo a cero el bit numero 7 (RBPU) del registro OPTION.
- RB0/INT estas líneas también pueden actuar como líneas de petición de interrupción.
- RB7-RB4 se programan para soportar una tarea especial cuando están configuradas como entradas. Si se cambia el estado lógico de una de ellas se provoca una interrupción.
- RB6 esta terminal se usa para la grabación en serie del PIC, para los impulsos de reloj y la RB7 para la entrada de datos.

1.4.6. REGISTROS DE CONTROL.

En las siguientes figuras (1.10a y 1.10b) se muestra gráficamente la distribución y tares de los bits de los registros de control (SFR) mas importantes:

IRP	RP1	RP0	TO#	PD#	Z	DC	C
-----	-----	-----	-----	-----	---	----	---

C:	Señalizador de acarreo en el octavo bit 1=Acarreo en la suma y no en la resta 0=No acarreo en la suma y si en la resta						
DC:	Señalizador de acarreo en el 4º bit de menos peso 1=Acarreo en la suma 0=No acarreo en la suma Ela resta lo contrario						
Z:	Señalizador de cero 1=El resultado de una operación es cero 0=El resultado de una operación es distinto de cero						
PD#:	Power Down 1=Tras conectar V_{DD} o ejecutar "CLRWDT" 0=Al ejecutar la instrucción "SLEEP"						
TO#:	Timer Out 1=Tras conectar V_{DD} o ejecutar "CLRWDT" o "SLEEP" 0=Al rebasar el tiempo del "WDT"						
RP<1:0>:	Selección de banco para el direccionamiento directo 11: Banco 3 (180-1Fh) 10: Banco 2 (100-17Fh) 01: Banco 1 (800-FFh) 00: Banco 0 (00-7Fh) Cada banco es de 128 bytes. El PIC16C84 solo usa el RP0 y RP1 debe mantenerse en cero						
IRP:	Selección de bancos para el direccionamiento indirecto 0=Bancos 0 y 1 (00-FFh) 1=Bancos 2 y 3 (100-1Fh) El IRP no se usa en el PIC16C84 y debe mantenerse en cero						
GIE:	Activación global de interrupciones 1=Concedido el permiso de interrupciones 0=No hay posibilidad de interrupciones						

Figura 1.10 a, Registros de ESTADO.

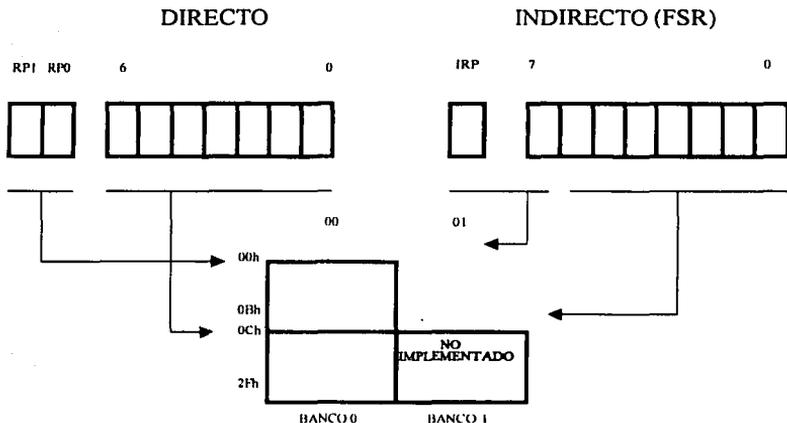
TESIS CON
FALLA DE ORIGEN

RBPO#	INTEDG	TOCS	TOSE	PSA	PS2	PS1	PS0
-------	--------	------	------	-----	-----	-----	-----

PS2:PS0		Rango en el que actúa el Divisor de Frecuencia				
PS2	PS1	PS0	División del TMR0	División del WDT		
0	0	0	1:2	1:1		
0	0	1	1:4	1:2		
0	1	0	1:8	1:4		
0	1	1	1:16	1:8		
1	0	0	1:32	1:16		
1	0	1	1:64	1:32		
1	1	0	1:128	1:64		
1	1	1	1:256	1:128		
PSA:		Asignación del Divisor de Frecuencias				
1=		El Divisor de frecuencia se le asigna al WDT				
0=		El divisor de frecuencia se le asigna al TMR0				
TOSE:		Tipo de flanco en T=CKI				
1=		Incremento de TMR0 cada flanco descendente				
0=		Incremento de TMR0 cada flanco ascendente				
TOCS:		Tipo de reloj para el TMR0				
1=		Pulsos introducidos a través de TOCKI (contador)				
0=		Pulsos de reloj interno Fosc/4 (temporizador)				
INTEDG:		Flanco activo interrupción externa				
1=		Flanco ascendente				
0=		Flanco descendente				
RBPO#:		Resistencias Pull-up Puerta B				
1=		Desactivadas				
0=		Activadas				

Figura. 1.10 b, Registro OPTION.

En las siguientes Figuras (1.11a, 1.11b y 1.11c) aparece el Registro de Direccionamiento Directo e Indirecto, Registro de control EECON1 para la EEPROM y Registro de Control de las Interrupciones INTCON respectivamente.



DIRECTO

Los 7 bits menos significativos que conforman el código OP de la instrucción, seleccionan una entre 128 posiciones diferentes de la memoria de datos. Los bits <6:5> del registro de estado (RP1:RP0) seleccionan el banco. Para acceder al banco 1 es obligatorio poner previamente a 1 el bit RP0.

INDIRECTO

Si como operando de una instrucción se accede al registro INDF (posición 0 área de datos), se accede realmente a la posición que en ese momento indique el registro FSR (posición 4 del área de datos). Los 7 bits de menos peso del <6:0>, seleccionan la posición; el bit <7> del FSR, junto con el bit IRP del registro de Estado (bit <7>), seleccionan el banco. En el PIC16C84 solo están implementados los bancos 0 y 1, por lo que el bit IRP debe valer "0".

Figura 1.11a, Registro de Direccionamiento Directo e Indirecto.

TESIS CON
FALLA DE ORIGEN

--	--	---	EEIF	WRER	WREN	WR	RD
<p>Bits 7-5: No implementados. Se leen como 0</p> <p>EEIFE: Señalizador de interrupción de la operación de escritura 1=La escritura ha terminado 0=La escritura no se ha terminado o no ha comenzado</p> <p>WRERR: Señalizador de error 1=La escritura ha terminado prematuramente 0=La escritura se ha completado</p> <p>WREN: Activación de escritura 1=Permite escribir 0=Prohíbe escribir en la memoria de datos EEPROM</p> <p>WR: Control de escritura 1=Inicia un ciclo 0=Prohibición de comienzo de ciclo de escritura en la memoria de la EEPROM</p> <p>RD: Control de lectura 1=Comienzo de lectura 0=Prohibición de comienzo de lectura</p>							

Figura 1.11b, Registros de control EECON1 para la EEPROM.

GIE	EEIE	TOIE	INTE	RBIE	TOIF	INTF	RBIF
<p>RBIF: Señalizador de estado de la puerta B 1=Cuando cambia de estado cualquier línea de PB (RB<7:4>) se borra por software 0=Ninguna entrada de PB ha cambiado</p> <p>INTF: Señalizador de estado de interrupción externa 1=La entrada de la interrupción se ha activado se borra por software 0=No hay interrupción externa</p> <p>TOIF: Señalizador de rebosamiento del TMR0 1=El TMR0 se ha rebosado. Se borra por software 0=El TMR0 no se ha rebosado</p> <p>RBIE: Activación de la interrupción de la PB 1=Interrupción activada 0=Interrupción desactivada</p> <p>INTE: Activación de la interrupción EXTERNA 1=Interrupción activada 0=Interrupción desactivada</p>							

Continúa.

TOIE:	Activación de la interrupción del TMR0
	1=Interrupción activada
	0=Interrupción desactivada
EEIE:	Activación de la interrupción de la memoria EEPROM
	1=Interrupción activada
	0=Interrupción desactivada
GIE:	Activación global de interrupciones
	1=Concedido el permiso de interrupciones
	0=No hay posibilidad de interrupciones

Figura 1.11c. Registro de control de las interrupciones INTCON.

1.5. DATOS DE FABRICANTE.

De los diferentes modelos que actualmente se comercializan, dentro de la familia genérica PIC16X8X fabricados por la compañía Microchip, se tiene que observar sus diversas características en función de escoger el PIC adecuado para una aplicación en particular. En este caso se describe las características del PIC16F84.

- 1) Margen del voltaje de alimentación.
- 2) Tipo de oscilador externo usado para generar la frecuencia de reloj.
- 3) Frecuencia máxima de funcionamiento

Nuestro PIC tiene impresa una F en su parte intermedia (PIC16F84), lo cual nos dice de que tipo de memoria interna esta constituido, en este caso la memoria es de tipo FLASH.

Las siguientes características eléctricas de funcionamiento que a continuación se enumeran son proporcionadas por su fabricante Microchip, en la tabla 1.1. se muestran las características eléctricas de operación.

CARACTERÍSTICAS ELECTRICAS DEL PIC16F84

Valores Máximos Absolutos.

Temperatura Ambiente bajo disponibilidad.....	-55°C a, + 125°C
Temperatura de Almacenaje.....	-65°C a, + 150°C
Voltaje V_{DD} con respecto a V_{SS}	-0.3, a + 7.5V
Voltaje MCLR con respecto a V_{SS}	-0.3, a + 14V
Voltaje en cualquier pin con respecto a V_{SS} (excepto V_{DD} y MCLR).....	-0.6 a ($V_{DD} + 0.6V$)
Potencia Total de Disipación.....	800 mW
Corriente Máxima de salida en el pin V_{SS}	150 mA
Corriente Máxima de entrada en el pin V_{DD}	100 mA
Corriente Máxima Absorbida por la línea.....	25 mA
Corriente Máxima Suministrada por la línea.....	20 mA
Corriente Máxima Absorbida en la puerto A.....	80 mA
Corriente Máxima Suministrada en la puerto A.....	50mA
Corriente Máxima Absorbida en la puerto B.....	150 mA
Corriente Máxima Suministrada en la puerto B.....	100 mA
Voltaje de Alimentación (V_{DD}).....	2 a, 6V DC
Voltaje de Grabación (V_{PP}).....	12 a, 14V DC

Tabla 1.1. Características eléctricas del PIC16F84.

En la Figura 1.13. se muestran las principales características de algunos modelos de PIC16F84, en relación a su frecuencia máxima y al tipo de oscilador a utilizar.

OSCILADOR	PIC16F84-04	PIC16F84-10
RC	VDD: 4.0 V a 6.0 V IDD: 4.5mA max. a 5.5 V IPD: 14µA max. a 4 V WDT dis Frec. 4.0 MHz max.	VDD: 4.5 V a 5.5 V IDD: 1.8mA tip. a 5.5 V IPD: 1.0µA tip. a 5.5 V WDT dis Frec. 4.0 MHz max.
XT	VDD: 4.0 V a 6.0 V IDD: 4.5 mA max. a 5.5 V IPD: 14µA max. A 4 V WDT dis Frec. 4.0 MHz max.	VDD: 4.5 V a 5.5 V IDD: 1.8mA tip. a 5.5 V IPD: 1.0µA tip a 5.5 V WDT dis Frec. 4.0 MHz max.
HS	VDD: 4.5 V a 6.0 V IDD: 4.5 mA tip a 5.5 V IPD: 1.0µA tip. a 4.5 V WDT dis Frec. 4.0 MHz max.	VDD: 4.5 V a 5.5 V IDD: 10mA max. a 5.5 V tip. IPD: 1.0µA tip. a 4.5 V WDT dis Frec. 10 MHz max.
LP	VDD: 4.0 V a 6.0 V IDD: 4.5 mA max. a 5.5 V IPD: 14µA max. A 4V WDT dis Frec. 4.0 MHz max.	NO SE USA EN EL MODULO LP

Figura 1.13. Principales características del modelo PIC16F84.

1.5.1. FRECUENCIA DE FUNCIONAMIENTO.

Una característica de trabajo fundamental en un microcontrolador es la frecuencia, debido a que se requiere establecer la velocidad en la ejecución de instrucciones y consumo de energía.

Si un PIC16F84 tiene una frecuencia de 10 MHz, que es su frecuencia máxima, le corresponde un ciclo de instrucción de 400 ns, debido a que cada instrucción tarda en ejecutarse 4 periodos de reloj, o sea $4 \times 100 \text{ ns} = 400 \text{ ns}$.

En un ciclo de instrucción se realizan todas las instrucciones del PIC, menos la de salto, que tardan el doble. Los impulsos del reloj entran por el pin OSC1/CLKIN y se dividen en 4 internamente, dando lugar a las señales Q1, Q2, Q3 y Q4, que se muestran en la figura 1.14. Durante un ciclo de instrucción, que contiene las cuatro señales mencionadas, tienen las siguientes funciones.

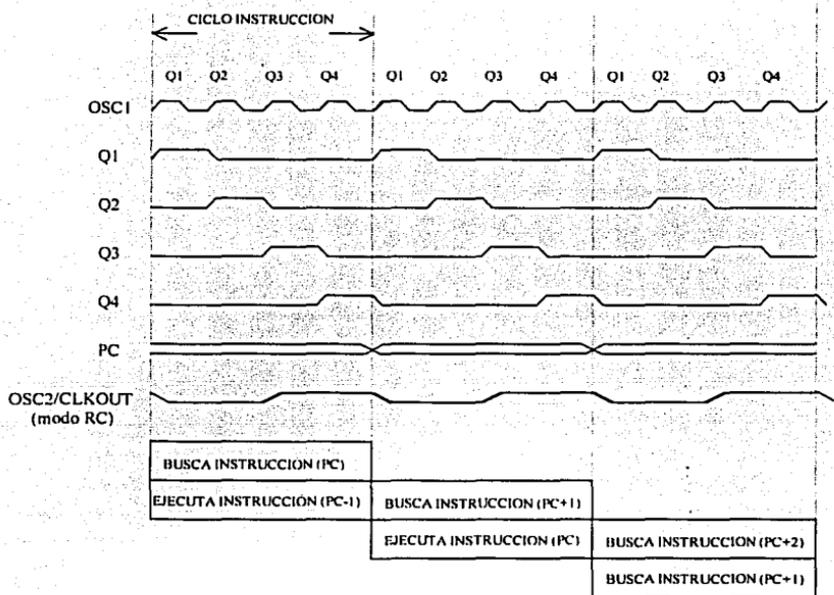


Figura 1.14. Impulsos de reloj externo (OSC1).

Q1: Durante este impulso se incrementa el contador de programa.

Q2-Q3: Durante la activación de estas dos señales se produce la decodificación y la ejecución de la instrucción.

Q4: Durante este impulso se busca el código de la instrucción en la memoria del programa y se carga en el registro de instrucción.

Para poder conseguir que se ejecute cada instrucción en un ciclo de instrucción, (excepto la de salto que tardan dos), se usa la técnica de segmentación o (pipe line), la cual consiste en realizar en paralelo las dos fases de las que consta cada instrucción.

En si cada instrucción se ejecuta en dos ciclos: el primero lleva a cabo la fase de búsqueda del código de la instrucción en la memoria de programa, y el segundo se decodifica y se ejecuta. Por la estructura segmentada del procesador se puede realizar al mismo tiempo la fase de ejecución de una instrucción y la búsqueda de la siguiente. Si la instrucción ejecutada corresponde a un salto no se sabe cual será la siguiente hasta que se complete, por esta situación se sustituye la fase de búsqueda de la siguiente instrucción por un ciclo (vacío), lo que origina que las instrucciones de salto tarden en realizarse dos ciclos de instrucción. Figura 1.15.

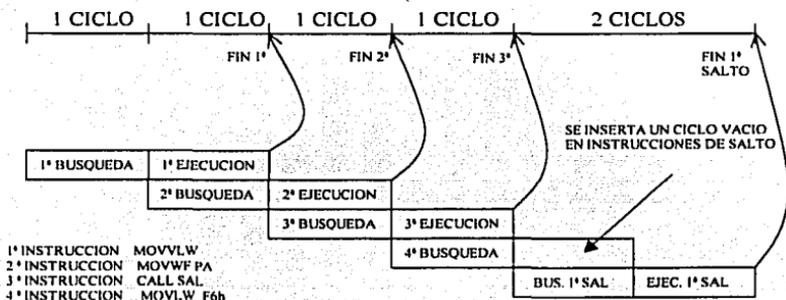


Figura 1.15. Segmentación.

CAPÍTULO II.

PROGRAMACIÓN DEL MICROCONTROLADOR.

2.1. PROGRAMAS Y LENGUAJES DE PROGRAMACIÓN.

Un programa es una serie de instrucciones estructuradas de tal manera que le indican al microcontrolador que hacer y en que momento, para que de esta manera el microcontrolador pueda realizar tareas o funciones específicas.

El programador es el encargado de estructurar dichas instrucciones, para esto debe de conocer perfectamente uno o varios lenguajes de programación, ya que con ellos el programador se comunica con el microcontrolador.

Los lenguajes de programación se pueden dividir en dos categorías: los de alto nivel y los de bajo nivel.

Un lenguaje de programación de alto nivel es mucho más parecido a un lenguaje cotidiano, escrito de forma natural y semejante al entendimiento humano, por ejemplo, el lenguaje C.

Un lenguaje de programación de bajo nivel es aquél que se acerca más a un lenguaje de máquina, por ejemplo, el lenguaje ensamblador.

Es indispensable que el programador conozca cuando menos un lenguaje de programación ya sea de alto o bajo nivel, ya que éste es el único medio por el cual el programador le indica al microcontrolador las instrucciones a realizar, esto es debido a que el microcontrolador maneja un lenguaje propio y exclusivo referido como lenguaje de máquina o código máquina.

El lenguaje de máquina o código máquina es el que únicamente entiende el microcontrolador a la hora de ejecutar un programa, este lenguaje es binario ya que consta exclusivamente de unos y ceros.

2.2. SECUENCIA PARA EL DESARROLLO DE PROGRAMAS.

El primer paso para desarrollar programas en lenguaje ensamblador consiste en escribir el llamado código fuente, el cual consta de un conjunto de mnemónicos (nombres de las instrucciones) y operandos (referencias de datos) seleccionados por el programador para que el microcontrolador pueda realizar las funciones deseadas; el código fuente se escribe con la ayuda de un editor de textos.

El segundo paso es el de traducir ese código fuente al llamado código máquina, también conocido como código objeto, el cual contendrá todos los códigos numéricos reales para que el microcontrolador ejecute el programa; este proceso es llevado a cabo por el programa ensamblador.

El tercer paso consiste en cargar el código máquina en la memoria de programa del microcontrolador, esto se logra por medio de un programa especial y de un circuito que sirve de interfase, a estos dos elementos en conjunto se les conoce como cargador o grabador.

El cuarto paso es la ejecución del programa, en donde se verifica que el programa corra (se ejecute) de la manera planeada, es decir, sin errores.

El quinto paso consiste en la depuración del programa, esto no es más que la corrección de los posibles errores, defectos o problemas que pudiera presentar el programa durante el paso anterior.

2.3. EL CÓDIGO FUENTE.

Para escribir el código fuente se puede usar cualquier editor de textos que emplee caracteres ASCII, generalmente el más empleado es el propio del sistema operativo DOS (EDIT.COM).

El uso de mayúsculas y minúsculas en el código fuente obedece a una serie de reglas o normas de estilo, las cuales son comunes entre los programadores a la hora de programar en lenguaje ensamblador que aunque no son obligatorias, le facilitan mucho al programador la lectura y/o comprensión del código fuente.

Algunas de las reglas más comunes o utilizadas son:

- 1) Directivas, parámetros, etiquetas y rótulos en mayúsculas.
- 2) Mnemónicos, operandos y comentarios en minúsculas.
- 3) Programa muy bien tabulado.

El texto que esté precedido de punto y coma únicamente tiene la función de servir como comentario, por lo que es ignorado por el ensamblador cuando éste traduce el código fuente a código máquina.

Las diversas formas de representar o expresar números en lenguaje ensamblador dependiendo de su base, se muestran en la tabla 2.1., en donde se puede observar que existen varias formas de representar números hexadecimales, aunque la más utilizada es la de anteponer 0x al número hexadecimal.

BASE.	REPRESENTACIÓN.
Binaria.	b'1010'
Decimal.	d'12'
Hexadecimal.	h'0c', 0x0c, 0ch ó 0c

Tabla 2.1. Representación de números según su base.

Además, no hay ningún problema si se olvida representar o como representar números hexadecimales ya que en estos casos el ensamblador asume que se utiliza o se utilizará la base hexadecimal.

Al terminar de escribir el código fuente éste se debe de guardar con un nombre de archivo que puede ser cualquiera, pero es imprescindible que la extensión sea ASM, por ejemplo: TESIS.ASM, esto es una forma de identificación para que así el programador y el ensamblador sepan que se trata de un archivo que contiene el código fuente de un programa escrito en lenguaje ensamblador.

2.3.1. ORGANIZACIÓN DEL CÓDIGO FUENTE.

Generalmente los códigos fuente en lenguaje ensamblador se organizan en cuatro columnas cuando se están escribiendo, esto con el fin de facilitar su visualización ya sea en la pantalla de un monitor o en una hoja impresa.

Estas columnas contienen los siguientes elementos:

La primer columna contiene las etiquetas que defina el programador y los rótulos que indican posiciones de memoria relativa, utilizados para los saltos y llamadas a subrutinas.

La segunda columna contiene a las directivas que usará el ensamblador y a los mnemónicos, es decir, los nombres de las instrucciones que se ejecutarán.

La tercer columna contiene los parámetros de las directivas y a los operandos, o sea, las direcciones o datos sobre los cuales actuarán los mnemónicos.

La cuarta columna se reserva única y exclusivamente para los comentarios, éstos sirven para aclarar, explicar o indicar algo con respecto al programa.

2.3.2. FORMATOS DE INSTRUCCIÓN.

Un formato de instrucción en lenguaje ensamblador tiene dos componentes principales: la instrucción y los operandos. Por medio de este formato se le indica al microcontrolador que operación realizar y sobre que información se realiza esta operación.

Instrucción.

La instrucción indica la operación que se va a efectuar y las diferentes instrucciones se distinguen entre sí por un nombre único, el cual proviene de la contracción o abreviatura de la palabra en inglés que denota la operación que realiza la instrucción. Por ejemplo: ADD=sumar, CLR=limpiar (clear), MOV=mover (move).

Cuando la instrucción es traducida por el ensamblador se convierte en el llamado código de operación. El conjunto de todas las instrucciones que pueden ser ejecutadas directamente por el microcontrolador recibe el nombre de juego de instrucciones.

Operandos.

Los operandos son valores o direcciones de la información sobre la cual el microcontrolador realiza las operaciones. Cuando los operandos son traducidos por el ensamblador éstos se convierten en los llamados campos de dirección. Los operandos generalmente se indican por medio de un valor numérico o también por una etiqueta; el valor numérico puede ser binario, decimal o hexadecimal y en el caso de la etiqueta, ésta debe ser previamente definida con un valor numérico.

Todos los juegos de instrucciones se caracterizan por el número de operandos que se permiten para cada instrucción, en el caso del juego de instrucciones del PIC16F84 pueden ser uno o dos.

2.4. DIRECTIVAS.

Una directiva no es más que una disposición dentro del código fuente, la cual va dirigida al programa ensamblador para que durante el proceso de ensamble la considere, es decir, que la tome y tenga en cuenta cuando ensamble el código fuente del programa; esta disposición se indica con los parámetros que lleve la directiva.

Existen diversas directivas, pero las más importantes y básicas cuando se realiza un programa en lenguaje ensamblador para el microcontrolador PIC16F84 son las relacionadas con: la inclusión de un archivo en el programa (INCLUDE), la configuración del microcontrolador (`_CONFIG`) y el fin del programa (END).

A continuación se explica con más detalle a cada una de las directivas antes mencionadas:

INCLUDE.

Lo que hace esta directiva es incluir o anexar al programa un archivo, cuando éste es ensamblado. Generalmente este archivo contiene una librería o en otros casos pudiera contener el código fuente de otro programa para así complementar al que se está realizando.

Ahora bien, una librería es un archivo creado por el fabricante del microcontrolador para cada uno de sus modelos de microcontroladores ya que existe una gran variedad de modelos con diversas características.

Por otra parte, existen varios tipos de librerías para diversos fines como por ejemplo: encabezados de programa, realización de operaciones matemáticas, lectura de teclados matriciales, manejo de displays de cristal líquido, etc.

Es recomendable utilizar la librería que sirve como encabezado de programa por que facilita y agiliza la elaboración de programas ya que contiene las etiquetas que definen configuraciones, registros y bits de más utilidad e importancia. Al denominar a este tipo de librería "encabezado de programa" es claro que debe ser el primer elemento en el código fuente del programa.

A este tipo de librerías se les identifica por que tienen como nombre de archivo a uno que hace referencia al modelo de microcontrolador para el cual fueron creadas.

El parámetro de la directiva INCLUDE es el nombre y extensión del archivo a incluir, la sintaxis de esta directiva es la siguiente:

```
INCLUDE <nombre y extensión del archivo>
```

```
__CONFIG.
```

Esta directiva se coloca inmediatamente después de que se anexa la librería de encabezado de programa con la directiva INCLUDE y su función es la de seleccionar y habilitar o deshabilitar las siguientes características operativas del microcontrolador:

- Oscilador (OSC).
 - LP (_LP_OSC).
 - XT (_XT_OSC).
 - HS (_HS_OSC).

- RC (`_RC_OSC`).
- Perro guardián (WDT).
 - Habilitado (`_WDT_ON`).
 - Deshabilitado (`_WDT_OFF`).
- Retardo para la estabilización de VDD (PWRTE).
 - Habilitado (`_PWRTE_ON`).
 - Deshabilitado (`_PWRTE_OFF`).
- Protección contra copia (CP).
 - Habilitada (`_CP_ON`).
 - Deshabilitada (`_CP_OFF`).

Los parámetros de la directiva `__CONFIG` son las características operativas a seleccionar y habilitar o deshabilitar, la sintaxis de esta directiva es la siguiente:

```
__CONFIG      OSC & WDT & PWRTE & CP
                END.
```

La función de esta directiva es la de indicarle al ensamblador el final de un programa, por lo tanto esta directiva debe ser el último elemento en el código fuente.

Esta directiva no tiene parámetros y su sintaxis es:

```
END
```

2.5. ETIQUETAS.

Cuando un programa se está elaborando el programador trabaja con varios números, la mayoría de ellos hexadecimales, los cuales están relacionados directamente con el microcontrolador ya sea que representen a un registro, una localidad de memoria, la posición de un bit o un dato.

Estos números pueden llegar a abrumar y hasta confundir en un momento dado al programador, por que no es tan fácil relacionar a varios números, sobre todo hexadecimales, con los elementos antes mencionados de un microcontrolador.

Debido a lo anterior es que se crean y se usan las etiquetas ya que ellas definen una equivalencia o igualdad entre un número y una palabra, ya que es más fácil relacionar cosas con palabras que con números. Esta equivalencia o igualdad se define mediante el uso de una peculiar directiva llamada EQU; el parámetro de esta directiva es el valor numérico con el que se define a la etiqueta.

Las etiquetas funcionan de la siguiente manera: uno crea una etiqueta en donde a un valor numérico, por ejemplo de un registro, se le define una igualdad con una palabra que lo identifique o relacione y a lo largo de la elaboración del código fuente se utiliza esta palabra en lugar del valor numérico del registro, cuando se ensambila el código fuente el ensamblador sustituye la palabra utilizada por el valor numérico del registro previamente definido.

La sintaxis para definir una etiqueta es la siguiente:

Etiqueta EQU Valor numérico

2.6. JUEGO DE INSTRUCCIONES.

Todos los modelos de microcontroladores PIC se basan en la arquitectura RISC (Reduced Instruction Set Computer), que significa: Computadora con Juego de Instrucciones Reducido. Esto no solamente implica que el número de instrucciones que es capaz de interpretar y ejecutar el microcontrolador es pequeño, como ocurre con el PIC16F84 que consta de 35, sino que también posee las características siguientes:

1) Las instrucciones son rápidas y simples.- La carencia de complejidad en la operación que realizan las instrucciones de los procesadores RISC da como resultado que la mayoría de éstas sean ejecutadas en sólo un ciclo de instrucción (4 ciclos de reloj), excepto las de salto ya que tardan dos ciclos de instrucción (8 ciclos de reloj).

2) Las instrucciones son ortogonales.- Cualquier instrucción puede usar cualquier operando, por lo que éstas apenas tienen restricciones en el uso de operandos.

3) La longitud de las instrucciones y los datos es constante.- Todas las instrucciones tienen la misma longitud (14 bits) y todos los datos también (8 bits); la arquitectura Harvard del procesador permite separar la memoria de instrucciones de la de datos, pudiendo tener diferente tamaño sus palabras.

2.6.1. DISEÑO DE LAS INSTRUCCIONES.

Las instrucciones de los PIC16F84 tienen un diseño de 14 bits de longitud, este diseño se divide en diferentes campos y cada campo hace referencia a operandos o elementos que maneja la instrucción en la operación que realiza el microcontrolador; dichos campos son los siguientes:

Campo del código de operación.- Los bits de este campo sirven para definir la operación que realiza la instrucción.

Campo de los operandos fuente (f) y destino (d).- Los bits de este campo definen los registros que sirven como operandos en la instrucción, suelen dar referencia a la dirección que ocupan en la memoria de datos.

Campo del operando inmediato o literal (k).- Los bits de este campo contienen el valor de un operando inmediato.

Campo que hace referencia a un bit (b).- Generalmente es un campo de 3 bits que indica la posición de un bit en particular dentro de un registro de 8 bits.

Campo de la dirección del salto.- En las instrucciones de salto CALL y GOTO existe un campo de bits que contiene la dirección de la siguiente instrucción a ejecutar; este campo de bits se carga en el contador de programa (PC) en las instrucciones de salto incondicional.

Los diversos diseños que aceptan las instrucciones del PIC16F84 se clasifican atendiendo al tipo de operación que realizan en cinco grupos:

- Instrucciones que manejan registros.
- Instrucciones que manejan bits.
- Instrucciones de salto.
- Instrucciones que manejan operandos inmediatos.
- Instrucciones de control y especiales.

2.6.2. INSTRUCCIONES QUE MANEJAN REGISTROS.

ADDWF.

Sintaxis: addwf f,d

Acción que realiza la instrucción: Suma los contenidos de los registros W y f, el resultado de esta suma se almacena en el registro W si d=0 o en el registro f si d=1.

Ciclos de reloj que tarda en ejecutarse la instrucción: cuatro.

Diseño de la instrucción: 000111dffffff.

Banderas que activa la instrucción: C, DC, Z.

ANDWF.

Sintaxis: andwf f,d

Acción que realiza la instrucción: Realiza la operación lógica AND entre los contenidos de los registros W y f, el resultado de esta operación se almacena en el registro W si d=0 o en el registro f si d=1.

Ciclos de reloj que tarda en ejecutarse la instrucción: cuatro.

Diseño de la instrucción: 000101dfffff.

Bandera que activa la instrucción: Z.

CLRF.

Sintaxis: clrf f

Acción que realiza la instrucción: Borra el contenido del registro f, es decir, pone a todos los bits del registro f a cero.

Ciclos de reloj que tarda en ejecutarse la instrucción: cuatro.

Diseño de la instrucción: 0000011fffff.

Bandera que activa la instrucción: Z.

CLRW.

Sintaxis: clrw

Acción que realiza la instrucción: Borra el contenido del registro W, es decir, pone a todos los bits del registro W a cero.

Ciclos de reloj que tarda en ejecutarse la instrucción: cuatro.

Diseño de la instrucción: 0000010XXXXXXXX.

Bandera que activa la instrucción: Z.

COMF.

Sintaxis: comf f,d

Acción que realiza la instrucción: Realiza la operación lógica NOT al contenido del registro f, el resultado de esta operación se almacena en el registro W si d=0 o en el registro f si d=1.

Ciclos de reloj que tarda en ejecutarse la instrucción: cuatro.

Diseño de la instrucción: 001001dfffff.

Bandera que activa la instrucción: Z.

DECF.

Sintaxis: decf f,d

Acción que realiza la instrucción: Decrementa en uno el contenido del registro f, el resultado de dicho decremento se almacena en el registro W si d=0 o en el registro f si d=1.

Ciclos de reloj que tarda en ejecutarse la instrucción: cuatro.

Diseño de la instrucción: 000011dfffff.

Bandera que activa la instrucción: Z.

INCF.

Sintaxis: incf f,d

Acción que realiza la instrucción: Incrementa en uno el contenido del registro f, el resultado de dicho incremento se almacena en el registro W si d=0 o en el registro f si d=1.

Ciclos de reloj que tarda en ejecutarse la instrucción: cuatro.

Diseño de la instrucción: 001010dfffff.

Bandera que activa la instrucción: Z.

IORWF.

Sintaxis: iorwf f,d

Acción que realiza la instrucción: Realiza la operación lógica OR entre los contenidos de los registros W y f, el resultado de esta operación se almacena en el registro W si d=0 o en el registro f si d=1.

Ciclos de reloj que tarda en ejecutarse la instrucción: cuatro.

Diseño de la instrucción: 000100dfffff.

Bandera que activa la instrucción: Z.

MOVf.

Sintaxis: movf f,d

Acción que realiza la instrucción: Mueve el contenido del registro f hacia el registro W si d=0 o hacia el propio registro f si d=1.

Ciclos de reloj que tarda en ejecutarse la instrucción: cuatro.

Diseño de la instrucción: 001000dfffff.

Bandera que activa la instrucción: Z.

MOVWF.

Sintaxis: movwf f

Acción que realiza la instrucción: Mueve el contenido del registro W hacia el registro f.

Ciclos de reloj que tarda en ejecutarse la instrucción: cuatro.

Diseño de la instrucción: 0000001fffff.

Banderas que activa la instrucción: ninguna.

NOP.

Sintaxis: nop

Acción que realiza la instrucción: Esta instrucción durante su ejecución no realiza ninguna acción.

Ciclos de reloj que tarda en ejecutarse la instrucción: cuatro.

Diseño de la instrucción: 0000000XX00000.

Banderas que activa la instrucción: ninguna.

RLF.

Sintaxis: rlf f,d

Acción que realiza la instrucción: Rota una posición hacia la izquierda a todos los bits del registro f, el resultado de esta rotación se almacena en el registro W si d=0 o en el registro f si d=1.

Ciclos de reloj que tarda en ejecutarse la instrucción: cuatro.

Diseño de la instrucción: 001101dffffff.

Bandera que activa la instrucción: C.

RRF.

Sintaxis: rrf f,d

Acción que realiza la instrucción: Rota una posición hacia la derecha a todos los bits del registro f, el resultado de esta rotación se almacena en el registro W si d=0 o en el registro f si d=1.

Ciclos de reloj que tarda en ejecutarse la instrucción: cuatro.

Diseño de la instrucción: 001100dffffff.

Bandera que activa la instrucción: C.

SUBWF.

Sintaxis: subwf f,d

Acción que realiza la instrucción: Resta los contenidos de los registros W y f, el resultado de esta resta se almacena en el registro W si d=0 o en el registro f si d=1.

Ciclos de reloj que tarda en ejecutarse la instrucción: cuatro.

Diseño de la instrucción: 000010dfffff.

Banderas que activa la instrucción: C, DC, Z.

SWAPF.

Sintaxis: swapf f,d

Acción que realiza la instrucción: Intercambia los cuatro primeros bits con los últimos cuatro bits del registro f, el resultado de este intercambio se almacena en el registro W si d=0 o en el registro f si d=1.

Ciclos de reloj que tarda en ejecutarse la instrucción: cuatro.

Diseño de la instrucción: 001110dfffff

Banderas que activa la instrucción: ninguna.

XORWF.

Sintaxis: xorwf f,d

Acción que realiza la instrucción: Realiza la operación lógica XOR entre los contenidos de los registros W y f, el resultado de esta operación se almacena en el registro W si d=0 o en el registro f si d=1.

Ciclos de reloj que tarda en ejecutarse la instrucción: cuatro.

Diseño de la instrucción: 000110dfffff.

Bandera que activa la instrucción: Z.

2.6.3. INSTRUCCIONES QUE MANEJAN BITS.

BCF.

Sintaxis: bcf f,b

Acción que realiza la instrucción: Pone a cero (borra) al bit b del registro f.

Ciclos de reloj que tarda en ejecutarse la instrucción: cuatro.

Diseño de la instrucción: 0100bbbfffff.

Banderas que activa la instrucción: ninguna.

BSF.

Sintaxis: bsf f,b

Acción que realiza la instrucción: Pone a uno al bit b del registro f.

Ciclos de reloj que tarda en ejecutarse la instrucción: cuatro.

Diseño de la instrucción: 0101bbbfffff.

Banderas que activa la instrucción: ninguna.

2.6.4. INSTRUCCIONES DE SALTO.

BTFSC.

Sintaxis: btfsc f,b

Acción que realiza la instrucción: Examina al bit b del registro f y se realiza un salto si este bit vale cero.

Ciclos de reloj que tarda en ejecutarse la instrucción: cuatro u ocho.

Diseño de la instrucción: 0110bbbfffff.

Banderas que activa la instrucción: ninguna.

BTFSS.

Sintaxis: btfss f,b

Acción que realiza la instrucción: Examina al bit b del registro f y se realiza un salto si este bit vale uno.

Ciclos de reloj que tarda en ejecutarse la instrucción: cuatro u ocho.

Diseño de la instrucción: 0111bbbfffff.

Banderas que activa la instrucción: ninguna.

DECFSZ.

Sintaxis: decfsz f,d

Acción que realiza la instrucción: Decrementa en uno el contenido del registro f y se realiza un salto si el resultado de este decremento hace que todos los bits del registro f estén a cero; además, el resultado de dicho decremento se almacena en el registro W si d=0 o en el registro f si d=1.

Ciclos de reloj que tarda en ejecutarse la instrucción: cuatro u ocho.

Diseño de la instrucción: 001011dfffff.

Banderas que activa la instrucción: ninguna.

INCFSZ.

Sintaxis: incfsz f,d

Acción que realiza la instrucción: Incrementa en uno el contenido del registro f y se realiza un salto si el resultado de este incremento hace que todos los bits del registro f estén a uno; además, el resultado de dicho incremento se almacena en el registro W si d=0 o en el registro f si d=1.

Ciclos de reloj que tarda en ejecutarse la instrucción: cuatro u ocho.

Diseño de la instrucción: 001111dfffff.

Banderas que activa la instrucción: ninguna.

2.6.5. INSTRUCCIONES QUE MANEJAN OPERANDOS INMEDIATOS.

ADDLW.

Sintaxis: addlw k

Acción que realiza la instrucción: Suma el contenido del registro W con el valor k, el resultado de esta suma se almacena en el registro W.

Ciclos de reloj que tarda en ejecutarse la instrucción: cuatro.

Diseño de la instrucción: 11111Xkkkkkkkk.

Banderas que activa la instrucción: C, DC, Z.

ANDLW.

Sintaxis: andlw k

Acción que realiza la instrucción: Realiza la operación lógica AND entre el contenido del registro W y el valor k, el resultado de esta operación se almacena en el registro W.

Ciclos de reloj que tarda en ejecutarse la instrucción: cuatro.

Diseño de la instrucción: 111001kkkkkkkk.

Bandera que activa la instrucción: Z.

IORLW.

Sintaxis: iorlw k

Acción que realiza la instrucción: Realiza la operación lógica OR entre el contenido del registro W y el valor k, el resultado de esta operación se almacena en el registro W.

Ciclos de reloj que tarda en ejecutarse la instrucción: cuatro.

Diseño de la instrucción: 111000kkkkkkkk.

Bandera que activa la instrucción: Z.

MOVLW.

Sintaxis: movlw k

Acción que realiza la instrucción: Carga al registro W con el valor k.

Ciclos de reloj que tarda en ejecutarse la instrucción: cuatro.

Diseño de la instrucción: 1100XXkkkkkkkkk.

Banderas que activa la instrucción: ninguna.

SUBLW.

Sintaxis: sublw k

Acción que realiza la instrucción: Resta el contenido del registro W con el valor k, el resultado de esta resta se almacena en el registro W.

Ciclos de reloj que tarda en ejecutarse la instrucción: cuatro.

Diseño de la instrucción: 11110Xkkkkkkkkk.

Banderas que activa la instrucción: C, DC, Z.

XORLW.

Sintaxis: xorlw k

Acción que realiza la instrucción: Realiza la operación lógica XOR entre el contenido del registro W y el valor k, el resultado de esta operación se almacena en el registro W.

Ciclos de reloj que tarda en ejecutarse la instrucción: cuatro.

Diseño de la instrucción: 111010kkkkkkkkk.

Banderas que activa la instrucción: ninguna.

2.6.6. INSTRUCCIONES DE CONTROL Y ESPECIALES.

CALL.

Sintaxis: call k

Acción que realiza la instrucción: Llama a la subrutina con el rótulo de nombre k.

Ciclos de reloj que tarda en ejecutarse la instrucción: ocho.

Diseño de la instrucción: 100kkkkkkkkkkk.

Banderas que activa la instrucción: TO#, PD#.

CLRWDT.

Sintaxis: clrwdt

Acción que realiza la instrucción: Pone a cero (resetea) el contador interno WDT.

Ciclos de reloj que tarda en ejecutarse la instrucción: cuatro.

Diseño de la instrucción: 00000001100100.

Banderas que activa la instrucción: ninguna.

GOTO.

Sintaxis: goto k

Acción que realiza la instrucción: Realiza un salto incondicional hacia el rótulo de nombre k.

Ciclos de reloj que tarda en ejecutarse la instrucción: ocho.

Diseño de la instrucción: 101kkkkkkkkkk.

Banderas que activa la instrucción: ninguna.

RETFIE.

Sintaxis: retfie

Acción que realiza la instrucción: Retorna de una interrupción y al bit GIE del registro INTCON lo pone a uno.

Ciclos de reloj que tarda en ejecutarse la instrucción: ocho.

Diseño de la instrucción: 000000000(00).

Banderas que activa la instrucción: ninguna.

RETLW.

Sintaxis: retlw k

Acción que realiza la instrucción: Retorna de una subrutina y carga al registro W con el valor k.

Ciclos de reloj que tarda en ejecutarse la instrucción: ocho.

Diseño de la instrucción: 1101XXkkkkkkkk.

Banderas que activa la instrucción: ninguna.

RETURN.

Sintaxis: return

Acción que realiza la instrucción: Retorna de una subrutina.

Ciclos de reloj que tarda en ejecutarse la instrucción: ocho.

Diseño de la instrucción: 00000000001000.

Banderas que activa la instrucción: ninguna.

SLEEP.

Sintaxis: sleep

Acción que realiza la instrucción: El microcontrolador entra en el modo de reposo.

Ciclos de reloj que tarda en ejecutarse la instrucción: cuatro.

Diseño de la instrucción: 00000001100011.

Banderas que activa la instrucción: TO#, PD#.

2.7. ENSAMBLADO DE PROGRAMAS.

El lenguaje de máquina es poco amistoso, confuso y extremadamente difícil de modificar, es por esta razón que los traductores de lenguaje se encuentran entre las herramientas de apoyo en software más importantes de que se disponen.

En este grupo se incluyen a los ensambladores, compiladores, emuladores e intérpretes; los cuales le permiten al usuario escribir programas directamente en un lenguaje de alto o bajo nivel que luego entonces traducen a código máquina para que el microcontrolador pueda ejecutar el programa.

El ensamblador es un programa muy especial ya que debe de guardar una estrecha relación con la arquitectura interna del microcontrolador; por ejemplo, en el caso particular de esta tesis el ensamblador lo suministró el propio fabricante del microcontrolador.

El ensamblado de programas es un proceso muy sencillo para el programador ya que sólo tiene que indicarle al ensamblador que archivo contiene el código fuente del programa, no hay que olvidar que este archivo debe de tener la extensión ASM, para que así el ensamblador se encargue de crear un archivo con el mismo nombre que el que contiene el código fuente pero éste con la extensión HEX.

Este nuevo archivo es el resultado del proceso de ensamble y contiene todos los códigos numéricos necesarios para que el microcontrolador ejecute el programa desarrollado. Este archivo es el que hay que cargar a través del cargador o grabador a la memoria de programa del microcontrolador.

Adicionalmente el ensamblador crea otros archivos igual con el mismo nombre que los de extensión ASM y HEX, siendo los más importantes los de extensión ERR y LST.

El de extensión ERR contiene la localización y una breve descripción de los errores que el ensamblador encontró en el código fuente, generalmente estos errores son de sintaxis u ortográficos, por lo que hay que reeditar el código fuente para corregir los errores detectados y volver a ensamblarlo hasta que no se registren errores por parte del ensamblador.

El de extensión LST no es tan importante o indispensable como los anteriores pero si es muy interesante su contenido ya que tiene información que puede llegar a ser útil y que no aparece en los demás archivos.

2.7.1. ENSAMBLADOR DE DOS PASOS.

Los ensambladores se pueden llevar a la práctica como ensambladores de uno, uno y medio o de dos pasos, estos últimos son los más comunes y se componen de los siguientes subsistemas:

El Explorador.

El explorador es el subsistema del ensamblador que separa, aísla y retorna los distintivos de cada instrucción en lenguaje ensamblador. Un distintivo es simplemente un elemento del lenguaje que se trata como una entidad singular e indivisible y no se separa ni simplifica más; todo lo que encuentra caracter por caracter lo clasifica para ver si está permitido o se puede usar por el ensamblador.

Paso Uno.

La función principal del paso uno es la de asignar un valor a todos los símbolos que se emplean en el código fuente. El valor de un símbolo durante el proceso de ensamblado es el valor del contador de localidades (LC) en el momento en el que el ensamblador define el símbolo.

Las responsabilidades mínimas del ensamblador durante el paso uno se reducen a las operaciones siguientes:

- 1) Manejo del compilador de posiciones.
- 2) Definición de todos los símbolos que se usan en el código fuente.

Estas dos responsabilidades a su vez, se reducen a una sola operación que es la creación de una tabla de símbolos. La tabla de símbolos es temporal y sirve al propio programa ensamblador a la hora de efectuar la traducción; una vez ensamblado el programa la tabla no tiene otra utilidad que la de su posible consulta por parte del programador durante el proceso de depuración.

Paso Dos.

La función principal del paso dos consiste en un análisis gramatical y en la generación de código.

Sus cuatro responsabilidades básicas son:

- 1) Analizar gramaticalmente los distintivos.
- 2) Generar código, es decir, pasar las instrucciones a código máquina.
- 3) Imprimir el listado de:
 - a) Señales de error.
 - b) Contador de localidades.
 - c) Código binario.
 - d) Código fuente.
- 4) Crear un módulo de salida para el proceso de ensamble, este módulo puede contener:
 - a) Dirección de la localidad donde se empieza a cargar el programa.
 - b) La extensión del programa.

- c) La dirección de la localidad donde se inicia la ejecución.**
- d) Información de referencias externas.**
- e) Información de relocalización.**

CAPÍTULO III

DISEÑO E IMPLEMENTACION DEL CIRCUITO DE CONTROL DE TEMPERATURA.

3.1. INTRODUCCION.

El capítulo comienza mostrando el diagrama a bloques y sus funciones, para continuar con una serie de opciones de componentes a seleccionar en base a las hojas de datos de los fabricantes. Así implementaremos cada uno de los bloques que componen el circuito.

También se describe el concepto de temperatura ambiente, escalas de medición y equivalencias entre dos de las escalas existentes: una la más utilizada en nuestro país y en el mundo (Centígrada o Celsius), y la otra la usada para propósitos científicos y tecnológicos (Absoluta o Kelvin).

De acuerdo al diagrama a bloques y siguiendo su secuencia, se describe para cada una de las etapas: el dispositivo base del cual se compone cada etapa, el funcionamiento de la misma y por último su diagrama electrónico.

El capítulo finaliza mostrando el diagrama electrónico general, es decir, donde se interconectan todas las etapas previamente diseñadas.

3.2. DIAGRAMA A BLOQUES Y FUNCIONES.

Este diagrama a bloques enmarca lo que el sistema debe realizar en cada etapa y se muestra en la Figura 3.1. donde se observa que la señal de entrada es la temperatura ambiente y luego pasa a la etapa sensora, en la cual se mide y se convierte en señal eléctrica para que en la etapa siguiente (comparador) se haga una comparación entre el voltaje de referencia con el voltaje del sensor y determine un voltaje de salida para una temperatura alta y otro para una temperatura baja. La quinta etapa dentro del sistema es la etapa de control, en la cual se deben procesar los diferentes voltajes de cada una de las temperaturas sensadas y el comportamiento de la carga será directamente proporcional al proceso de esta etapa.

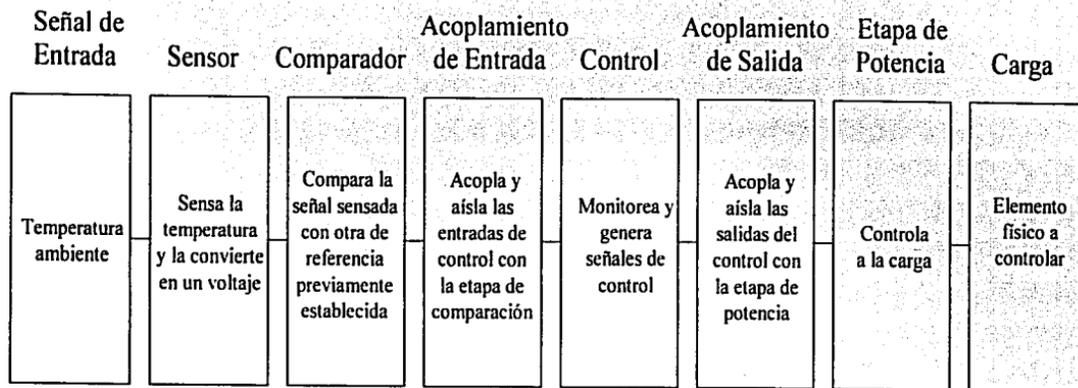


Figura. 3.1. Diagrama a bloques del sistema

3.3. SELECCIÓN DE COMPONENTES Y OPCIONES

3.3.1. SELECCIÓN DEL SENSOR .

DESCRIPCION GENERAL.

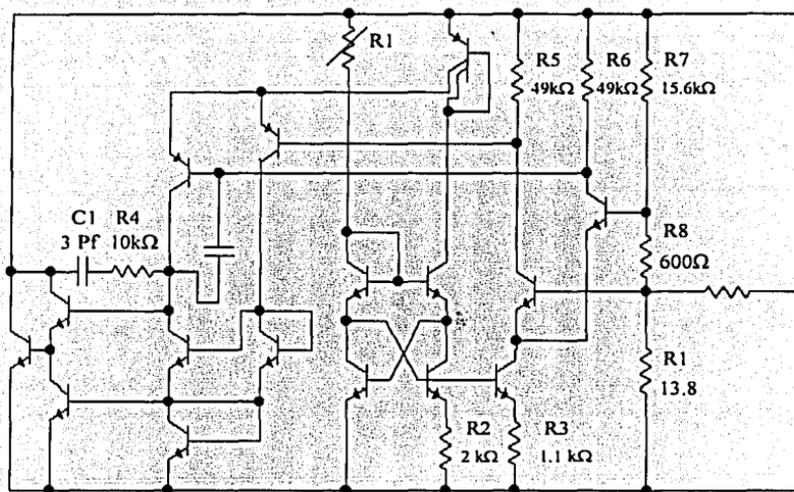
Debido a que la señal de entrada es la temperatura ambiente que puede existir en un tiempo determinado, se requiere que el sensor detecte a esta temperatura.

La serie de componentes de precisión LM con su facilidad de calibración son circuitos integrados sensores de temperatura, que operan como un diodo Zener y por esto llevan su símbolo, el cual se muestra en la figura 3.2.

El LM135 tiene una caída de voltaje directamente proporcional en temperatura absoluta de $+10 \text{ mV/}^\circ\text{K}$. Con menos de 1Ω de impedancia dinámica de salida, el dispositivo opera bajo un rango de corriente de $400 \mu\text{A}$ a 5 mA sin cambio virtual en su estructura. Cuando se calibra a 25°C que es la típica el LM135 tendrá menos de 1°C de error sobre 100°C de rango de temperatura. A diferencia de otros sensores el LM135 tiene una salida lineal.

Las aplicaciones para el LM135 incluye casi cualquier tipo de temperatura a sensor, desde un rango bajo los -55°C hasta $+150^\circ\text{C}$ de rango de temperatura, mientras que el LM235 opera desde unos -40°C hasta $+125^\circ\text{C}$ de rango de temperatura.

DIAGRAMA ESQUEMATICO.



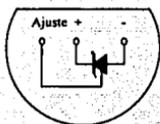
CARACTERISTICAS

- Calibración Directa en $^{\circ}\text{K}$
- Dispone de 1°C inicial de precisión
- Opera desde los 400 μA hasta los 5mA
- Menos de 1Ω de impedancia dinámica
- Fácil calibración
- Extenso rango de temperatura de operación
- Bajo costo

Los componentes que forman a esta serie son:

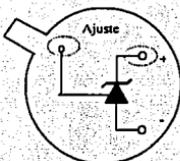
El LM135/LM235/LM335 están disponibles en empaques herméticos de transistor TO-46 mientras que el LM335 solo esta disponible en empaque plástico TO-92. estos se muestran en la figura 3.2.

TO-92
Empaque Plástico



Vista Inferior
Numero de Serie LM335Z
o LM335AZ

TO-46
Empaque de Metal



Vista Inferior
Numero de Serie LM135H,
LM235 o LM335H

Figura 3.2. Empaques de los componentes LM.

En las siguientes tablas 3.1-a, 3.1-b y 3.1-c se enuncian los rangos de temperatura y las características eléctricas de los sensores.

Parámetros	Condiciones	LM135/LM235A			LM135/LM235			Unidades
		Min.	Típica	Max.	Min.	Típica	Max.	
Voltaje de operación a la salida	$T_c = 25^\circ\text{C}$, $I_R = 1\text{mA}$	2.97	2.98	2.99	2.95	2.98	3.01	V
Error de temperatura sin calibrar	$T_c = 25^\circ\text{C}$, $I_R = 1\text{mA}$		0.5	1		1	3	$^\circ\text{C}$
Error de temperatura sin calibrar	$T_{\min} \leq T_c \leq T_{\max}$, $I_R = 1\text{mA}$		1.3	2.7		2	5	$^\circ\text{C}$
Error de temperatura al calibrar con 25°C	$T_{\min} \leq T_c \leq T_{\max}$, $I_R = 1\text{mA}$		0.3	1		0.5	1.5	$^\circ\text{C}$
Error al extenderse la temperatura	$T_c = T_{\max}$ (intermitente)		2			2		$^\circ\text{C}$
No lineal	$I_R = 1\text{mA}$		0.3	0.5		0.3	1	$^\circ\text{C}$

Tabla 3.1-a. Rangos de Temperatura del LM135/235, LM135A/LM235A.

Parámetros	Condiciones	LM135/LM235A			LM135/LM235			Unidades
		Min.	Tipica	Max.	Min.	Tipica	Max.	
Voltaje de operación a la salida	$T_c = 25^\circ\text{C}$, $I_R = 1\text{mA}$	2.95	2.98	3.01	2.92	2.98	3.04	V
Error de temperatura sin calibrar	$T_c = 25^\circ\text{C}$, $I_R = 1\text{mA}$		1	3		2	6	$^\circ\text{C}$
Error de temperatura sin calibrar	$T_{\min} \leq T_c \leq T_{\max}$, $I_R = 1\text{mA}$		2	5		4	9	$^\circ\text{C}$
Error de temperatura al calibrar con 25°C	$T_{\min} \leq T_c \leq T_{\max}$, $I_R = 1\text{mA}$		0.5	1		1	2	$^\circ\text{C}$
Error al extenderse la temperatura	$T_c = T_{\max}$ (intermitente)		2			2		$^\circ\text{C}$
No lineal	$I_R = 1\text{mA}$		0.3	1.5		0.3	1.5	$^\circ\text{C}$

Tabla 3.1-b. Rangos de Temperatura del LM335, LM335A.

Parámetros	Condiciones	LM135/LM235A			LM335A			Unidades
		Min.	Tipica	Max.	Min.	Tipica	Max.	
Voltaje de operación a la salida con cambio en la corriente	$400\ \mu\text{A} \leq I_R \leq 5\text{mA}$		2.5			3	14	mV
Impedancia dinámica	$I_R = 1\ \text{mA}$		0.5			0.6		Ω
Coefficiente de temperatura del voltaje a la salida			10			10		$\text{mV}/^\circ\text{C}$
Constante de tiempo	En vacío		80			80		seg
	100 ft/aire mínimo		10			10		seg
	En aceite		1			1		seg
Estabilidad	$T_c = 125^\circ\text{C}$		0.2			0.2		

Tabla 3.1-c. Características eléctricas.

CALIBRACION.

Este chip incluye un fácil método de calibración, para una mayor precisión del dispositivo. Un potenciómetro conectado a través del LM135 con el cursor a la terminal de ajuste permite la calibración del sensor para así evitar imprecisiones sobre amplios rangos de temperatura.

La salida del LM135 es proporcional a la temperatura absoluta, presentando en el sensor un voltaje de salida de 0V a 0°K (-2373.15). La salida del dispositivo calibrado o descalibrado (con potenciómetro o sin el) es expresada como.

$$V_{salT} = \left[V_{salT_0} \right] \times \left(\frac{T}{T_0} \right)$$

Donde T es la temperatura desconocida y T_0 es la temperatura de referencia, nominalmente la salida tiene una variación de 10 mV por °K

3.3.2. SELECCIÓN DEL COMPARADOR.

DESCRIPCIÓN GENERAL.

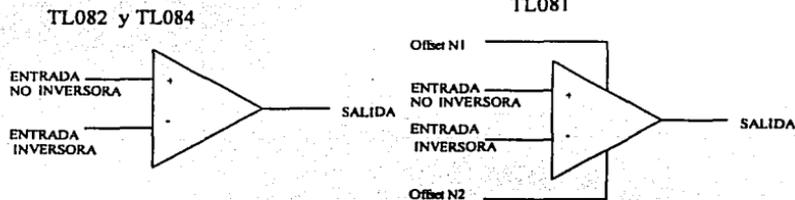
Dentro de los componentes que pueden realizar la comparación de voltajes, se encuentran los amplificadores operacionales.

La serie TL de amplificadores Operacionales con entrada JFET está diseñada para ofrecer una amplia selección de operaciones que cualquier otra familia desarrollada anteriormente.

Cada uno de estos amplificadores operacionales incorpora un alto voltaje JFET y un transistor bipolar en un circuito integrado monolítico. Las características del dispositivo son alta velocidad de respuesta, entrada de offset, vías de baja corriente y un bajo coeficiente de temperatura de voltaje de offset. El ajuste de offset y las opciones de compensación externa, están disponibles dentro de la familia TL08x. Los componentes que forman esta serie son:

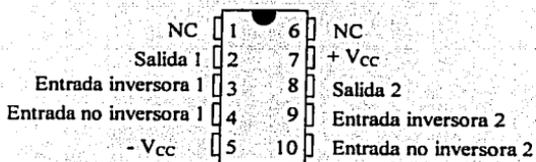
En los dispositivos el sufijo-C, esta caracterizado para operar desde 0 °C hasta 70 °C. El los dispositivos de sufijo-I, esta caracterizado para operar desde los - 40 °C hasta 85 °C. El sufijo-Q, denota una operación desde los - 40 °C hasta los 125 °C. En el sufijo-M, se caracteriza por operar por completo en temperaturas dentro del rango militar, desde - 55 °C hasta 125°C.

SÍMBOLO.

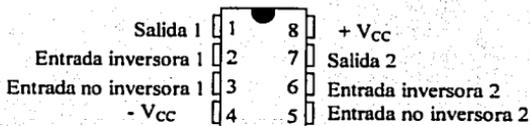


ENCAPSULADO.

Vista Superior
TL082M
EMPAQUE U



Vista Superior
TL082, TL082A, TL082B
EMPAQUE, D, JG, P, o PW.



Vista Superior
TL082M
EMPAQUE FK

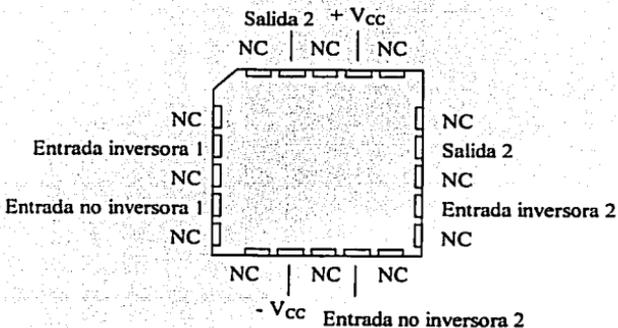
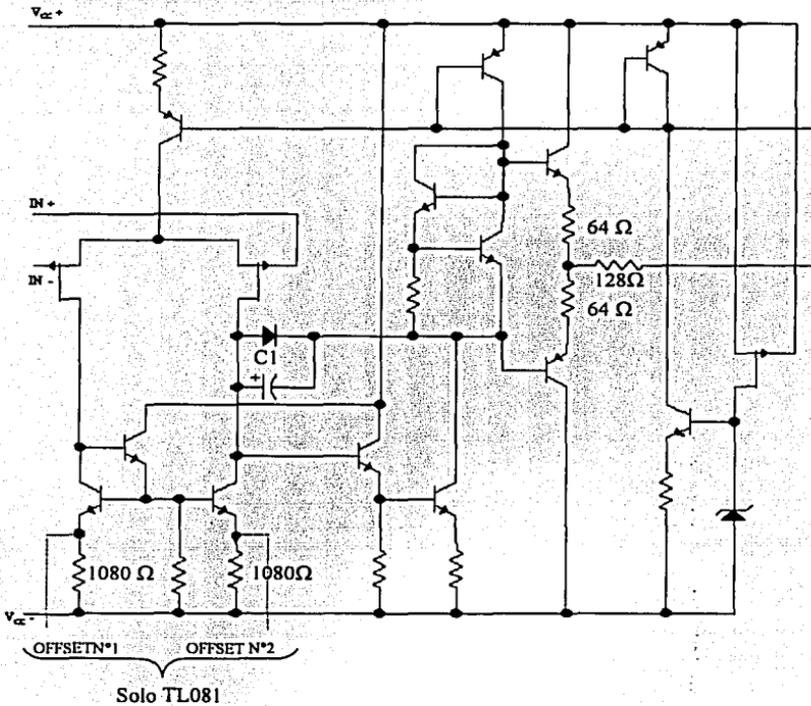


DIAGRAMA ESQUEMATICO.



CARACTERÍSTICAS

- Bajo consumo de energía
- Baja distorsión por armónicos (0.003 %), típica
- Protección de corto circuito a la salida
- Alta impedancia de entrada
- En modo común el rango de voltaje de entrada puede ser hasta Vcc
- Alta velocidad de respuesta

	TL08_C TL08_AC TL08_BC	TL08_I	TL084_Q	TL08_M	UNIT
Fuente de Voltaje Vcc +	18	18			V
Fuente de Voltaje Vcc -	-18	-18			V
Entrada diferencial de Voltaje, Vid	±30	±30	±30	±30	V
Entrada de Voltaje, Vi	±15	±15	±15	±15	V
Duración de cortocircuito a la salida	ilimitado	ilimitado	ilimitado	ilimitado	
Total de potencia continua disipada	Ver la tabla para los rangos de disipación				
Rango de operación de temperatura en vacío, TA	0 hasta 70	-40 hasta 85	-40 hasta 125	-55 hasta 125	°C
Rango de temperatura de almacenaje	-65 hasta 150	-65 hasta 150	-65 hasta 150	-65 hasta 150	°C
Temperatura para casos de 60 seg Tc	Empaque FK			260	°C
Conductor de temperatura de 1,6 mm para el caso de 60 seg	Empaque J o JG			300	°C
Conductor de temperatura de 1,6 mm para el caso de 10 seg	Empaque D, N, P o PW	260	260	260	°C

Tabla 3.4. principales características eléctricas (TL081, TL081A, TL082B, TL082, TL082A, TL082B, TL082Y, TL084, TL084A, TL084B y TL084Y.

3.3.3. SELECCIÓN DEL ACOPLAMIENTO DE ENTRADA.

DESCRIPCIÓN GENERAL.

La serie 4N de acopladores ópticos puede realizar la función de acoplamiento de una manera eficiente.

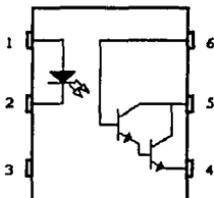
El optoacoplador 4N32 y 4N33 consisten de un transistor fotodarlington acoplado ópticamente con un diodo emisor infrarrojo de arseniuro de galio en un empaque plástico DIP (dual inline package).

Los elementos son montados de esquina a esquina usando una técnica coplanar, que provee una distancia fija entre la entrada y salida para grandes requerimientos de seguridad.

CARACTERÍSTICAS.

- Alta resistencia de aislamiento
- Baja capacitancia de acoplamiento, típica 0.3 μ pF
- Alto radio de transferencia de corriente (CTR)
- Bajo coeficiente de temperatura de CTR
- Sistema de acoplamiento directo

DIAGRAMA.



En las siguientes tablas 3.5-a y 3.5-b se describen los rangos de operación y las características eléctricas de los optoacopladores 4N32-33.

Entrada (Emisor)

Parámetros	Condiciones	Símbolo	Valor	Unidad
Voltaje inverso		V_R	6	V
Corriente en adelanto		I_F	60	mA
Corriente en adelanto		I_{FSM}	3	A
Disipación de potencia	$t_p \leq 10\mu s$	P_V	100	mW
Temperatura de unión	$T_{amb} \leq 25^\circ C$	T_J	125	$^\circ C$

Salida (Detector)

Parámetros	Condiciones	Símbolo	Valor	Unidad
Voltaje colector-base		V_{CBO}	50	V
Voltaje colector-emisor		V_{CEO}	30	V
Voltaje emisor-colector		V_{ECO}	5	V
Corriente de colector		I_C	150	mA
Corriente pico en el colector	$t_p/T = 0.5, t_p \leq 10ms$	I_{CM}	200	mA
Disipación de potencia	$T_{amb} \leq 25^\circ C$	P_V	150	mW
Temperatura de unión		T_J	125	$^\circ C$

Acopladas

Parámetros	Condiciones	Símbolo	Valor	Unidad
Voltaje eficaz (RMS)	$t = 1 \text{ min}$	V_{IO}	3.75	kV
Total de potencia disipada	$T_{amb} \leq 25^\circ\text{C}$	P_{tot}	250	mW
Rango de temperatura amb.		T_{amb}	-55 a 100	$^\circ\text{C}$
Rango de temperatura de almacenaje		T_{stg}	-55 a 100	$^\circ\text{C}$
Temperatura de soldadura	$2 \text{ mm}, t \leq 10 \text{ s}$	T_{SD}	260	$^\circ\text{C}$

3.5-a. Rangos de temperatura y operación.

Entrada (Emisor)

Parámetros	Condiciones	Símbolo	Min.	Tip.	Max.	Unidad
Voltaje en Adelanto	$I_F = 50 \text{ mA}$	V_F		1.25	1.5	V
capacitancia	$V_R = 0, f = 1 \text{ MHz}$	C_J		50		pF

Salida (Detector)

Parámetros	Condiciones	Símbolo	Min.	Tip.	Max.	Unidad
Voltaje colector-base	$I_C = 100 \mu\text{A}$	V_{CBO}	50			V
Voltaje colector-emisor	$I_C = 1 \text{ mA}$	V_{CEO}	30			V
Voltaje emisor-colector	$I_C = 100 \mu\text{A}$	V_{ECO}	5			V
Corriente de colector	$V_{CE} = 10 \text{ V},$ $I_F = 0$ $E = 0$	I_{CEO}			100	nA

Acopladas

Parámetros	Condiciones	Símbolo	Min.	Tip.	Max.	Unidad
Voltaje eficaz (RMS)	$f = 50 \text{ HZ}, t = 2 \text{ s}$	V_{IO}	3.75			KV
Resistencia de aislamiento	$V_{IO} = 1000 \text{ V}$ 40% humedad relativa	R_{IO}		10		Ω
Voltaje de saturación de colector-emisor	$I_F = 8 \text{ mA}$ $I_C = 2 \text{ mA}$	V_{CEsat}			1	V
Interrupción de frecuencia	$I_F = 2 \text{ mA}$ $V_{CE} = 10 \text{ V}$ $R_L = 100 \Omega$	f_C		30		kHz
Capacitancia de acoplamiento	$f = 1 \text{ MHz}$	C_K		0.3		pF

Corriente proporcional de transferencia (CTR)

Parámetros	Condiciones	Tipo	Símbolo	Min.	Tip.	Max.	Unidad
I_C/I_F	$V_{CE}=10\text{ V}$ $I_F=10\text{ mA}$ $t_p/T=0.01$ $t_r=0.3\text{ ms}$	4N32, 4N33	CTR	2			

3.5-b. Características eléctricas.

3.3.4. SELECCIÓN DEL CONTROL

De los diferentes componentes que pueden realizar operaciones de control, existe una gran variedad de microcontroladores que actualmente están a la venta con diferentes modelos y cuya nomenclatura genérica es PIC16X8X, con las siguientes características.

1. Frecuencia máxima de funcionamiento.
2. Tipo de oscilador externo usado para la generación de la frecuencia de reloj.
3. Margen del voltaje de alimentación.
4. Puertos de control de entrada/salida.

En los microcontroladores de la familia de la gama o serie mediana se encuentran diferentes tipos códigos como C, F o CR, los cuales admiten un rango de voltaje de alimentación estándar, que oscila de 4 a 4.2 V DC como mínimo, y como máximo 5.5 a 6 V DC. Otros modelos con códigos intermedios LC, LF o LCR. Soportan un voltaje que va desde los 2 V a los 6 V DC.

En la siguiente figura 3.5a. y 3.5b. se indican las principales características de algunos modelos, en relación a su frecuencia máxima, tipo de oscilador empleado y las características de la Gama 16X8X.

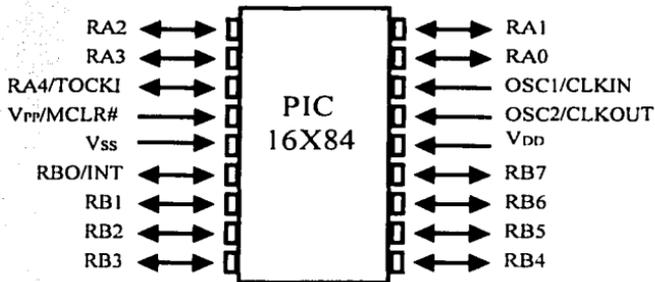
Oscilador	PIC 16F84-04 PIC 16F83-04	PIC 16F84-10 PIC 16F83-10	PIC 16LF84-04 PIC 16LF83-04
RC	VDD: 4.0 a 6.0 V IDD: 4.5 mA max. a 5.5V IPD: 14µA max. a 4V WDT dis Frecuencia 4.0 MHz max.	VDD: 4.5 a 5.5 V IDD: 1.8 mA tip. a 5.5V IPD: 1.0µA tip. a 5.5V WDT dis Frecuencia 4.0 MHz max.	VDD: 2.0 a 6.0 V IDD: 4.5 mA max. a 5.5V IPD: 7.0µA max. a 4V WDT dis Frecuencia 2.0 MHz max.
XT	VDD: 4.0 a 6.0 V IDD: 4.5 mA max. a 5.5V IPD: 14µA max. a 4V WDT dis Frecuencia 4.0 MHz max.	VDD: 4.5 a 5.5 V IDD: 1.8 mA tip. a 5.5V IPD: 1.0µA tip. a 5.5V WDT dis Frecuencia 4.0 MHz max.	VDD: 2.0 a 6.0 V IDD: 4.5 mA max. a 5.5V IPD: 7.0µA max. a 4V WDT dis Frecuencia 2.0 MHz max.
HS	VDD: 4.5 a 6.0 V IDD: 4.5 mA típica a 5.5V IPD: 1.0 µA tip. a 4.5V WDT dis Frecuencia 4.0 MHz max.	VDD: 4.5 a 5.5 V IDD: 10 mA max. a 5.5V tip. IPD: 1.0µA tip. a 4.5V WDT dis Frecuencia 10 MHz max.	NO SE USA EN EL MODO HS
LP	VDD: 4.0 a 6.0 V IDD: 35µA tip. a 32 kHz, 3.0V IPD: 0.6µA tip. a 3.0V WDT dis Frecuencia 200 kHz max.	NO SE USA EN EL MODO LP	VDD: 2.0 a 6.0 V IDD: 32µA max. A 32 kHz, 3.0V IPD: 7.0µA max. a 2V WDT dis Frecuencia 200 kHz max.

Tabla.3.6-a. Principales características de los PIC16X8X.

	Reloj Frec. Max.	Memoria					Características			Encapsulado
		Memoria de Prog. Flash	EEPROM	ROM	RAM de Datos Bytes	Datos EEPROM Bytes	Fuentes de Interrupción	Pines I/O	Rango de Voltaje	
PIC16C84	10	—	1024	—	36	64	4	13	2.0-6.0	18 pines DIP, SOIC
PIC16F84	10	1024	—	—	68	64	4	13	2.0-6.0	18 pines DIP, SOIC
PIC16CR84	10	—	—	1024	68	64	4	13	2.0-6.0	18 pines DIP, SOIC
PIC16F83	10	512	—	—	36	64	4	13	2.0-6.0	18 pines DIP, SOIC
PIC16CR83	10	—	—	512	68	64	4	13	2.0-6.0	18 pines DIP, SOIC

Tabla. 3.6-b. Variantes del PIC16X8X.

DIAGRAMA.



3.3.5. SELECCIÓN DEL ACOPLAMIENTO DE SALIDA.

DESCRIPCIÓN GENERAL

Para poder aislar etapas de potencia del resto del circuito y con esto evitar daños mayores, se puede usar optoacopladores.

La serie MOC consiste de un diodo emisor infrarrojo de arseniuro de galio, ópticamente acoplado a un interruptor.

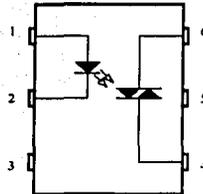
Dentro de la serie MOC existe el 3022 cuya salida es un triodo de corriente alterna TRIAC (triodo altern courant),

APLICACIONES

- Controles de Solenoide/válvula
- Balastra de lámpara
- Interfase de microprocesador hacia periféricos de 115 Vac,
- Control de motor
- Interruptor de potencia estático
- Relés de estado sólido
- Dimmers para lámparas incandescentes

DIAGRAMA DEL MOC3022

TESIS CON
FALLA DE ORIGEN



Los rangos de operación y las características eléctricas, se muestran en las tablas. 3.7-a y 3.7-b.

Diodo emisor infrarrojo

Parámetros	Símbolo	Valor	Unidad
Voltaje inverso	V_R	3	V
Corriente en adelante	I_F	60	mA
Total de potencia disipada potencia mínima en el triac cerca de 25°C	P_D	100 1.33	mW

Salida

Parámetros	Símbolo	Valor	Unidad
Apagado a la salida de la terminal de voltaje	V_{DRM}	400	V
Corriente pico (PW = 1 ms, 120 pps)	I_{TSM}	1	A
Total de potencia disipada cerca de 25°C	P_D	300 4	mW mW/°C

Total

Parámetros	Símbolo	Valor	Unidad
Aislamiento de voltaje (Voltaje pico AC, 60 Hz, 1segundo de duración)	V_{ISO}	7500	Vac(pk)
Total de potencia disipada TA = 25°C Cerca de 25°C	PD	330 4.4	mW mW/°C

Continúa

Rango de temperatura de almacenaje	TJ	-40 a 100	°C
Rango de operación a temperatura ambiente	TA	-40 a 85	°C
Rango de temperatura de almacenaje	Tstg	-40 a 150	°C
Temperatura de soldadura (10 s)	TL	260	°C

3.7-a. Rangos de operación.

Led de entrada

Características	Símbolos	Min.	Tip.	Max.	Unidad
Corriente inversa de escape ($V_R = 3 \text{ V}$)	I_R		0.05	100	μA
Voltaje en adelanto ($I_F = 10 \text{ mA}$)	V_F		1.15	1.5	V

Detector de salida.

Características	Símbolos	Min.	Tip.	Max.	Unidad
Bloqueo de corriente pico, en otra dirección	I_{DRM}		10	100	nA
Encendido de voltaje pico, en otra dirección	V_{TM}		1.8	3	V
Proporción crítica de voltaje de rizo Cuando esta apagado	dv/dt		10		V/ μs

Acopladas

Características	Símbolos	Min.	Tip.	Max.	Unidad
Led Terminal principal de voltaje = 3V MOC3021 MOC3022 MOC3023	I_{FT}		8	15 10 5	mA
Corriente, en otra dirección	I_{II}		100		μA

3.7-b. Características eléctricas.

3.3.6. SELECCIÓN DE LA ETAPA DE POTENCIA.

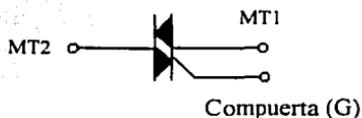
DESCRIPCIÓN GENERAL.

La serie MAC218 y MAC218-A, están diseñadas principalmente para: aplicaciones de corriente alterna de onda completa, dimmers para lámparas, control de motores, control de calefacciones y fuentes de poder.

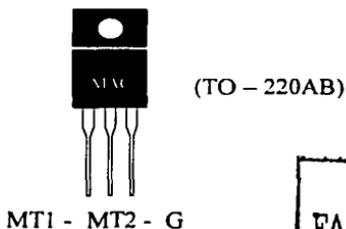
CARACTERÍSTICAS

- Bloqueo de voltaje desde 200 hasta 800 V. (según el modelo).
- Uniones pasivas de vidrio para un gran parámetro de uniformidad y estabilidad.
- Construcción en encapsulado TO-220, para una baja resistencia térmica, alta disipación de calor y durabilidad.
- Disparo garantizado de la compuerta en tres modos (serie MAC218) ó cuatro modos (serie MAC218-A).

DIAGRAMA.



ENCAPSULADO.



TESIS CON
FALLA DE ORIGEN

En las tablas 3.8-a, 3.8-b y 3.8-c, se muestran los rangos máximos de operación, las características térmicas y las características eléctricas de la serie de TRIA's MAC.

Rangos	Símbolo	Valor	Unidad
Voltaje apagado de pico repetitivo compuerta abierta, $T_J=25$ Hasta 125°C MAC218-4, MAC218A4 MAC218-6, MAC218A6 MAC218-8, MAC218A8 MAC218-10, MAC218A10	V_{DRM}	200 400 600 800	V.
Corriente de encendido RMS ángulo de conducción = 360° , $T_c = +80^\circ\text{C}$	$I_{\text{T(RMS)}}$	8	A
Pico no repetitivo de corriente un ciclo completo, 60 Hz, $T_c = 80^\circ\text{C}$ precedido y seguido por	I_{TSM}	100	A
Corriente de fusión ($T = 6.3$ ms)	I^2t	40	A^2s
Potencia pico de compuerta ($T_c = +80^\circ\text{C}$, ancho de pulso = μs)	P_{GM}	16	W
Potencia de promedio de compuerta ($T_c = +80^\circ\text{C}$, Tiempo = 8.3 ms)	$P_{\text{G(av)}}$	0.35	W
Corriente de disparo pico de compuerta (Ancho de pulso = $1\mu\text{s}$)	I_{GTM}	4	A
Rango de temperatura de unión	T_J	-40 hasta +125	$^\circ\text{C}$
Rango de temperatura de almacenaje	T_{stg}	-40 hasta +150	$^\circ\text{C}$

Tabla 3.8-a. Rangos máximos de operación.

Características	Símbolo	Max.	Unidad
Resistencia térmica en caso de unión	$R_{\theta\text{JC}}$	2.2	$^\circ\text{C/W}$

Tabla 3.8-b. características térmicas.

Características	Símbolo	Min.	Típica	Max.	Unidad
Corriente de bloqueo pico ($V_D = \text{rated } V_{DRM}$, compuerta abierta) $T_J = 25^\circ\text{C}$ $T_J = 125^\circ\text{C}$	I_{DRM}			10 2	μA mA
Voltaje pico de encendido en cualquier dirección ($I_{TM} = 11.3 \text{ A}$ pico: ancho de pulso = 1 hasta 2 ms, exento de ciclo < 2%)	V_{TM}		1.7	2	V
Corriente de disparo de compuerta (DC continua) ($V_D = 12 \text{ Vdc}$, $R_L = 12 \Omega$) Modo de disparo MT2(+), GATE(+); MT2(+), GATE (-); MT2(-), GATE(-) MT2(-), GATE(+) sufijo "A" solamente	I_{GT}			50 75	MA
Compuerta de disparo de voltaje (Terminal principal de voltaje = 12 Vdc, $R_L = 100 \Omega$) MT2(+), G(+) MT2(+), G(-) MT2(-), G(-) MT2(-), G(+) sufijo "A" Solamente (Terminal principal de voltaje = V_{DRM} , $R_L = 10 \text{ k}\Omega$, $T_J = 125^\circ\text{C}$) MT2(+), G(+); MT2(-), G(-); MT2(+), G(-) MT2(-), G(+) sufijo "A" solamente	V_{GT}		0.9 0.9 1.1 1.4	2 2 2 2.5	Volts
Corriente de sostenimiento (cualquier dirección) ($V_D = 24 \text{ Vdc}$ puerta abierta al inicio de la corriente = 200 mA)	IH	0.2 0.2		20	MA

Tabla 3.8-c. Características eléctricas

3.3.7. SELECCIÓN DE LA CARGA.

DESCRIPCION GENERAL.

Al no contar la mayoría de los hogares y pequeños comercios con energía trifásica para operar maquinas eléctricas, estas deben operar con energía monofásica. En estos casos, todos los motores deben de operar con fuentes de potencia monofásicas.

Los motores monofásicos que cumplen con los requisitos para el diseño dentro de esta tesis son, el motor universal y los motores de inducción de uso especial que son el motor de reluctancia y el motor de histéresis.

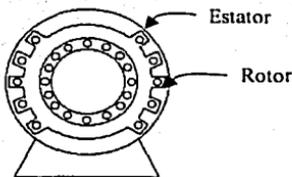
El motor universal es quizá el motor mas utilizado en cualquier fuente de potencia monofásica. La simplicidad de este motor consiste en tomar una maquina de dc y accionarla desde una fuente de ac.

El otro es el motor monofásico inducción con rotor de jaula de ardilla y estator monofásico los cuales presentan una gran desventaja al existir una sola fase en el devanado del estator, el campo magnético no rota y en su lugar, primero pulsa con gran intensidad, después con menor intensidad pero en la misma dirección. Este motor al no tener campo magnético rotacional en le estator, el motor monofásico de inducción no tiene par de arranque.

CARACTERÍSTICAS.

- Velocidad sincrónica sin devanados de campo del rotor.
- Aceleración hasta la velocidad sincrónica por si solos.
- Características par/velocidad semejante a motores trifásicos.

DIAGRAMA.



3.4. SEÑAL DE ENTRADA.

El concepto común de temperatura se relaciona con las sensaciones fisiológicas de calor y de frío, al igual que el de fuerza se relaciona con el esfuerzo muscular; sin embargo, las determinaciones obtenidas por este método corresponden a percepciones relacionadas con el sentido del tacto que, por su carácter subjetivo, carecen de significación científica.

El análisis de la estructura molecular de los cuerpos permite definir la temperatura como una medida de la energía cinética media de las moléculas que los forman, y la mecánica estadística, al relacionarla con las restantes magnitudes termodinámicas, proporciona una definición de la misma mediante la ecuación $T = (\delta U / \delta S)V$; donde T = temperatura, U = energía interna, S = entropía, V = volumen. según la cual la temperatura de un cuerpo es la derivada parcial a volumen constante de la energía interna con respecto a la entropía.

Es posible definir parcialmente la temperatura a partir del principio cero de la termodinámica, que establece que dos sistemas están en equilibrio térmico con un tercero están también en equilibrio entre si. Se dice entonces de los tres sistemas en equilibrio que están a la misma temperatura.

Las variaciones de la temperatura T de los cuerpos van acompañadas de cambios de otras magnitudes físicas del mismo, tales como el volumen, el índice de refracción, la resistividad eléctrica específica, etc. Pudiendo establecerse una relación denominada función de estado [$T = F(\alpha)$], entre T y cualquiera de las magnitudes variables con la misma, α . Además, dado que dos cuerpos a distinta temperatura puestos en contacto equilibran ésta hasta igualarla, puede utilizarse uno de ellos como termómetro para determinar la temperatura del otro.

3.4.1. ESCALAS DE TEMPERATURA.

Una escala termométrica se define, una vez establecida la relación entre la magnitud variable α elegida y la temperatura, fijando arbitrariamente dos puntos a los que se atribuyen determinados valores de la temperatura.

En la escala CENTÍGRADA O ESCALA CELSIUS, se toman los puntos fijos de manera que el cero de la misma corresponden a la temperatura de fusión del hielo, y la división numero 100, a la de ebullición del agua, a 760 mm de presión y a 45° de latitud.

La escala REAUMUR utiliza los mismos puntos fijos que la centigrada, pero asignando para la ebullición del agua la división 80 de la escala.

La escala FAHRENHEIT utiliza como puntos fijos los de fusión y ebullición de una disolución de cloruro amonico en agua; la división 32 de la misma corresponde al 0 de las escalas centigrada y de Reaumur y la 212 a la temperatura de ebullición del agua. La relación entre los grados de las tres escalas viene dada por las ecuaciones:

$$\frac{^{\circ}\text{C}}{100} = \frac{^{\circ}\text{R}}{80} = \frac{^{\circ}\text{F} - 32}{180}$$

Donde °C = grados Centígrados, °R = grados Reaumur, °F = grados Fahrenheit. La escala mas utilizada es la Centígrada, la Reaumur está completamente en desuso y la Fahrenheit se utiliza en los países anglosajones y Japón, tendiendo esta ultima a cambiar por la centigrada de mas facilidad de manejo.

Otra escala que se emplea universalmente, sobre todo en los medios científicos, fue la propuesta por el fisico Ingles LORD KELVIN, a la cual se le ha dado su nombre, escala KELVIN o escala absoluta. Esta escala surgió de las discusiones relacionadas con las temperaturas máximas y mínimas que puede alcanzar un cuerpo.

Se comprobó, que, teóricamente, no hay un limite superior para la temperatura que puede alcanzar un objeto. Pero se observo que hay un limite natural cuando se intenta bajar la temperatura. Los estudios realizados demuestran que es imposible obtener una temperatura inferior a -273.15°C. Esta temperatura se denomina cero absoluto. En realidad, el cero absoluto es una temperatura es una temperatura limite que no se puede alcanzar, y por ello solo se han obtenido valores muy próximos a ella.

Kelvin propuso como origen de su escala (representado por 0° K), la temperatura del cero absoluto, y un intervalo unitario igual al intervalo de 1°C, es decir $\Delta(1^{\circ}\text{K}) = \Delta(1^{\circ}\text{C})$, de esta manera se obtiene:

$$0^{\circ}\text{K} \Rightarrow -273^{\circ}\text{C}$$

$$1^{\circ}\text{K} \Rightarrow -272^{\circ}\text{C}$$

$$2^{\circ}\text{K} \Rightarrow -271^{\circ}\text{C}$$

$$273^{\circ}\text{K} \Rightarrow 0^{\circ}\text{C}$$

$$373^{\circ}\text{K} \Rightarrow 100^{\circ}\text{C}$$

De modo general, designado por T la temperatura Kelvin, Y por t_c la temperatura Celsius correspondiente, es fácil concluir, que:

$$T = t_c + 273$$

Entonces, para expresar en la escala Kelvin una temperatura dada en grados Celsius, basta sumar 273 a este valor.

3.5. ETAPA SENSORA.

3.5.1.SENSOR DE TEMPERATURA LM335.

Se empleara un sensor de temperatura LM335, ya que este es un sensor de temperatura de tipo semiconductor y convierte la cantidad de grados que percibe en su superficie un voltaje de corriente continua.

Entonces el segundo bloque de este proyecto de tesis puede estar formado por un circuito integrado diseñado por National Semiconductor para trabajar como sensor de temperatura.

Su referencia es LM335 y contiene en su interior 16 transistores, 2 condensadores y 10 resistencia, presentadas en un empaque similar al de un transistor tipo TO-92. (Figura 3.2)

Este sensor trabaja como un diodo zener cuyo voltaje de ruptura es directamente proporcional a la temperatura que le llega a su superficie.

Por cada grado de temperatura ya sea kelvin (K) o centígrados (C), el LM335 varia su voltaje en 10mV, presentando teóricamente una lectura de 0V a la temperatura de 0°K, (cero absoluto) es decir a -273 °C, en la practica este circuito trabaja en un rango comprendido entre - 40 °C y 100°C.

3.5.2.FUNCIONAMIENTO.

La etapa se puede polarizar con 12 V, se debe usar una resistencia de 10 kΩ para limitar la corriente en la terminal positiva del sensor (pin N°2), por lo que éste al ser polarizado presenta entre sus terminales una diferencia de voltaje, el cual es proporcional a la temperatura.

El sensor cuenta con una terminal de ajuste (pin N°1) con el fin de calibrar el mismo con la relación proporcional voltaje-temperatura deseado. Esta relación de voltaje-temperatura tiene un incremento de 10 mV por cada 1°C, por lo que en la tabla 3.9.se muestra su relación voltaje-temperatura promedio.

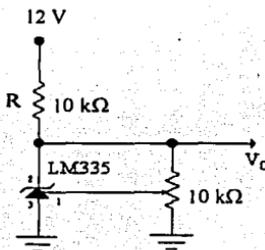
VOLTAJE	TEMPERATURA
2.932 V	20°C
2.952 V	22°C
2.962 V	23°C
2.972 V	24°C
2.982 V	25°C
2.992 V	26°C
3.002 V	26°C
3.012 V	27°C
3.032 V	30°C
3.082 V	35°C

Tabla 3.9. Relación voltaje-temperatura.

Por lo tanto esta etapa es de suma importancia debido a que sirve de sensor y transductor al mismo tiempo: Sensor al evaluar la temperatura ambiente y transductor al convertir temperatura en voltaje.

Después de obtener la señal eléctrica V_O , ésta ira directamente al comparador por terminal 2. La salida de este circuito se encuentra en paralelo con un potenciómetro de $10\text{ k}\Omega$, que aprovechará el voltaje V_O para que a través de su pin central se le aplique un voltaje al pin 1 de ajuste al sensor.

3.5.3. DIAGRAMA.



3.6. ETAPA COMPARADORA.

3.6.1. AMPLIFICADOR OPERACIONAL TL082.

Se eligió el comparador diseñado por Texas Instruments TL082, ya que se adecua a la perfección a las necesidades del proyecto de tesis. Decimos que se adecua ya que tiene entradas J-FET, lo que hace que tenga una impedancia de entrada mucho mayor, opera con un muy bajo ruido, tiene un bajo costo y no precisa de fuentes simétrica para su funcionamiento, como sería el caso del OPAM 741. Debido a que se desea sensar una temperatura alta y una baja, se van a utilizar dos circuitos comparadores

3.6.2. FUNCIONAMIENTO.

Al obtener la señal eléctrica V_O proveniente del sensor de temperatura LM335, cada comparador tiene en su entrada a V_O y en la segunda terminal de entrada mediante una calibración previamente establecida, se tendrá a las señales eléctricas de referencia (V_{ref}) a comparar con la señal V_O .

La etapa de referencia esta alimentada con 12 V para aprovechar el voltaje de polarización de los operacionales, y esta formada por dos resistencias y un potenciómetro de ajuste para determinar una temperatura alta y otra baja respectivamente. En estos dos arreglos se puede determinar su voltaje de referencia mediante la ecuación para un divisor de voltaje y con los valores indicados en el diagrama se tendrán los siguientes valores.

a) para el extremo superior de R_2 se encuentra el valor V_{ref} máximo

$$V_{Ref} = \frac{V_{CC}(R_2 + R_3)}{(R_1 + R_2 + R_3)} = 4.067 \text{ V, valor máximo}$$

b) para el extremo inferior de R_2 se encuentra el valor mínimo de V_{ref} .

$$V_{Ref}' = \frac{V_{CC}(R'_2)}{(R'_1 + R'_2 + R'_3)} = 2.033 \text{ V, valor mínimo}$$

Donde; R_1 , R_2 , R_3 , R'_1 , R'_2 , y R'_3 son las resistencias y los potenciómetros de cada arreglo respectivamente. La variación máxima y mínima de V_{ref} es de 2V aproximadamente.

Al obtener el voltaje V_{ref} el amplificador operacional TL082 lo comparara con V_O a través de la relación para un circuito comparador, la cual es:

$$V_{SAL} = [V(+) - V(-)] Aol$$

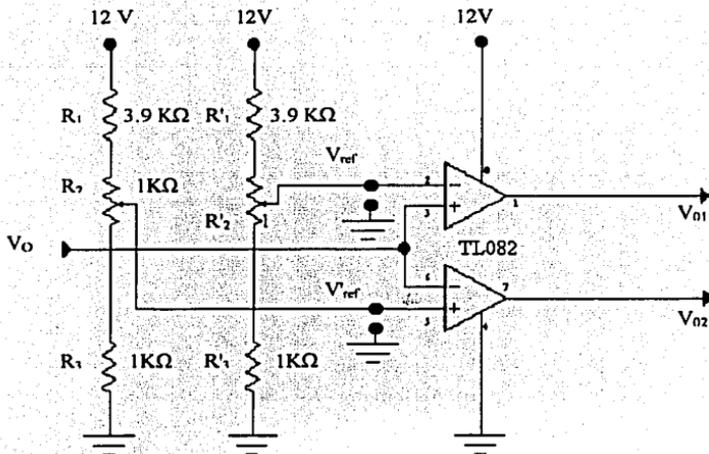
Donde: $V(+)$ es la entrada no inversora.

$V(-)$ es la entrada inversora.

Aol es la ganancia de voltaje en lazo abierto y tiene un valor típico de 200,000 (dato de fabricante).

Al comparar la señal proveniente del sensor de temperatura y la señal V_{ref} , se tendrá posibles dos señales a la salida (V_{01} y V_{02}), que tomadas como valores lógicos o como voltajes serian: 1 y 0 lógicos ó 12V y 0V, respectivamente.

3.6.3. DIAGRAMA.



3.7. ETAPA DE ACOPLAMIENTO DE ENTRADAS

3.7.1. OPTOACOPLADOR 4N33.

El 4N33 se escogió por ser un optoacoplador con salida a darlington, ya que con este tipo de salida se obtiene una mejor relación CTR (corriente proporcional de transferencia) lo cual asegura el corte y saturación del par darlington.

3.7.2. FUNCIONAMIENTO.

Al presentarse el voltaje de salida V_{01} , ó V_{02} del operacional con valor 1 lógico circula una corriente que es limitada por la resistencia de $2.7 \text{ k}\Omega$, esta misma corriente es la que circula a través del diodo infrarrojo, por lo que emite un haz de luz que es captada por el arreglo fotodarlington con lo cual el transistor pasa a la región de saturación, y se presenta una caída de voltaje máxima en la R_1 que esta en el emisor.

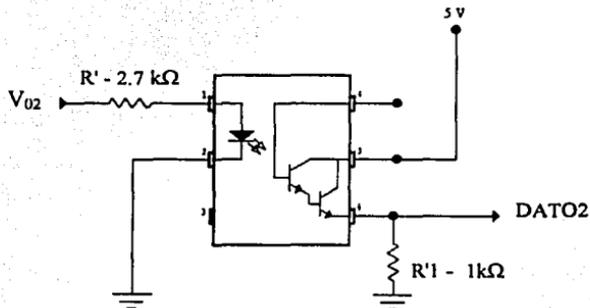
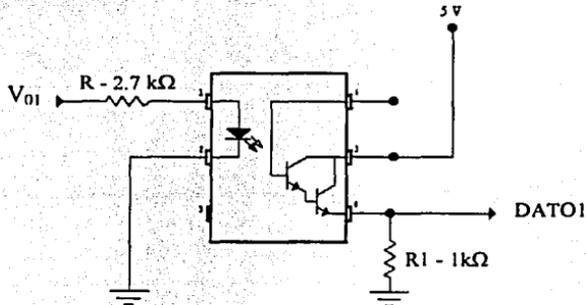
Cuando se presenta un voltaje en V_{01} ó V_{02} igual a 0 volts, no circula corriente en el diodo infrarrojo, por lo que el fotodarlington se encuentra en la región de corte y se presenta una caída de voltaje mínimo en el emisor.

Por lo anterior el voltaje D_1 y D_2 que se encuentra en la salida del acoplamiento de entradas puede tomar uno de los siguientes valores:

a) valor máximo $D_1 = 5V - 0.2V = 4.8V$

b) valor minino $D_2 = 5V - 5V = 0V$

3.7.3. DIAGRAMA



3.8. ETAPA DE CONTROL.

3.8.1. MICROCONTROLADOR PIC16F84.

La decisión de elegir el PIC16F84 es porque cumple a la perfección con los requisitos para el rápido aprendizaje de los microcontroladores y comenzar a diseñar proyectos nuevos. Es sencillo de programar, es practico y de bajo costo, en comparación con otros de distinta gama.

Pero la mayor razón para su utilización, es, el tipo de memoria de programa con la que cuenta. En el caso del PIC16C84 existe una memoria EEPROM de 1024 palabras de 14 bits cada una. Este PIC es el que nosotros utilizamos en esta tesis tiene la misma capacidad de memoria de instrucciones, pero esta es de tipo FLASH. Los dos constan de 64 bytes de EEPROM como memoria de datos auxiliares y opcional.

Tanto la memoria EEPROM como la FLASH, son eléctricamente grabables y borrables, lo cual permite escribir y borrar el programa para su operación, así se mantiene el PIC en su zócalo, usando el mismo dispositivo para grabación y borrado. Esta propiedad es la que hace aventajar a éstos, de los demás tipos, que tienen memoria de tipo EPROM, la cual se graba eléctricamente, pero para borrarla se somete a una luz ultravioleta por un periodo de tiempo, lo que implica sacar al PIC de su zócalo y colocarlo en un borrador de EPROM. Su fabricante Microchip introdujo la memoria FLASH debido a su gran posibilidad de aumentar su capacidad con relación a la EEPROM. Tiene también una mayor velocidad y menor consumo de energía, no obstante, la EEPROM soporta 1,000,000 de ciclos de escritura / borrado, con tan solo 1000 ciclos de la FLASH.

3.8.2. FUNCIONAMIENTO.

Lo primero que se observa dentro de esta etapa es un saludo inicial en el display con la leyenda HOLA, precedido de un punto que marca el final del mismo para empezar a mostrar la temperatura ambiente que existe en el momento del encendido.

Las señales de entrada en los pines 1 y 2 del pic, contienen las señales eléctricas ya sensadas previamente y aisladas, con lo que, el microcontrolador desempeña las funciones preestablecidas dentro de su programa interno, por lo que este solo obedece las funciones para las que esta programado y es mostrar si existe una temperatura baja (BA), normal (nor) alta (AL) y falla (FALLA), la cual se describe en la tabla 3.9.

D ₁	D ₂	ESTADO
0	0	FALLA
0	1	BAJA
1	0	ALTA
1	1	NORMAL

Tabla 3.9. Tabla de verdad.

El pin 3 esta alimentado con 5V y tiene en paralelo un swich con el que se despliega el numero de cuenta en el display de los dos integrantes de esta tesis. Los dos siguientes pines: 4 y 5 se encuentran conectados con un reset, el cual funciona en caso de exista una falla ó al mostrar el numero de cuenta y se quiera reestablecer la medición.

Todos los pines subsiguientes desde el pin 6, hasta el pin 13, están conectados al display donde se muestra el saludo inicial, las temperaturas sensadas, fallas y numero de cuenta de los integrantes con los siguientes desplegados precedidos de un punto que anuncia el final y con cada letra subsecuentemente para cada función descrita.



Temperatura
Baja



Temperatura
Normal



Temperatura
Alta



Falla al desconectarse la alimentación del sensor.



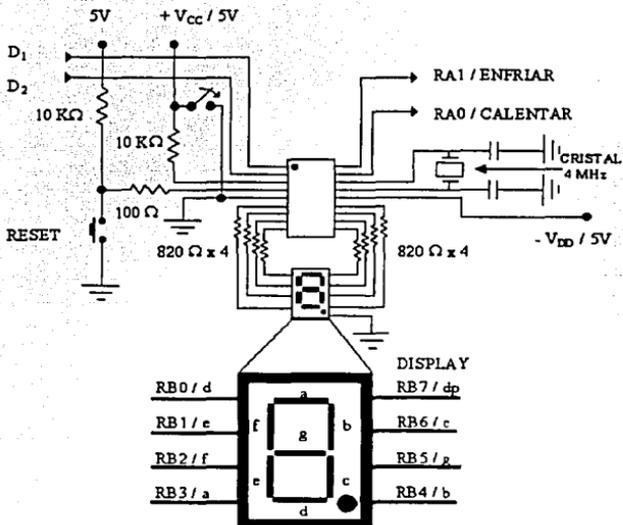
Identificación del circuito.

El pin 14 por donde se aplica la tensión de alimentación de 5V, los pines 15 y 16 admiten osciladores externos para la aplicación de la frecuencia de funcionamiento.

Aquí se encuentra el oscilador de tipo XT, el cual es un oscilador de cristal o resonador para frecuencias comprendidas entre los 100 KHz y 4 MHz.

Los dos pines restantes son el 17 y 18, estos dos pines están configurados por medio del programa para que actúen como salidas donde tendremos las dos señal eléctrica activen las cargas dependiendo de la temperatura, y muestran los mismos valores que a la entrada.

3.8.3. DIAGRAMA.



3.9. ETAPA DE ACOPLAMIENTO DE SALIDAS.

3.9.1. OPTOACOPLADOR MOC3022.

El MOC3022 se escogió por ser un optoacoplador con salida a TRIAC, ya que puede conducir en cualquier dirección una vez que ha excedido su voltaje de transición conductiva; este TRIAC es el elemento que permite activar a la etapa de potencia.

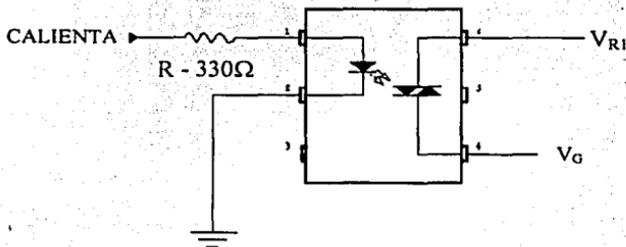
3.9.2. FUNCIONAMIENTO.

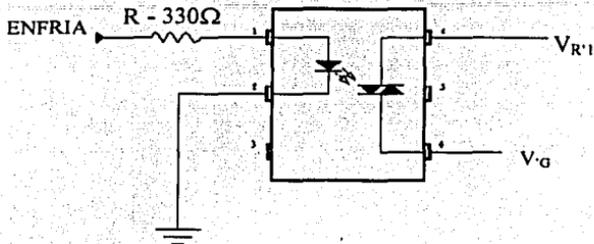
Se escogió un optoacoplador con salida a TRIAC, con la finalidad de aislar la etapa de baja tensión (control) con la etapa de alta tensión (potencia), por cuestión de seguridad

El funcionamiento de esta etapa es muy sencilla, la entrada se conecta a la salida del PIC (D_1 Y D_2), en esta salida tendremos ya sea un nivel lógico alto (5 V) ó un nivel lógico bajo (0V); ahora bien, cuando se presenta un nivel lógico alto circula una corriente a través del diodo lo cual, emite un rayo de flujo radiante y hace que el fototriac conduzca. Caso contrario, cuando se presenta un nivel lógico bajo no circula corriente, por lo tanto el diodo infrarrojo no conduce, haciendo que el fototriac permanezca apagado.

Cuando el fototriac se activa, la terminal 4 y 6 prácticamente están en corto, por lo que el voltaje V_{R1} y V'_{R1} aparecen en las terminales V_G y V'_G respectivamente.

3.9.3. DIAGRAMA.





3.10. ETAPA DE POTENCIA.

3.10.1. TRIAC MAC 218.

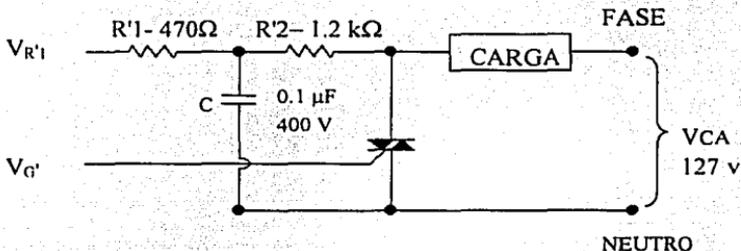
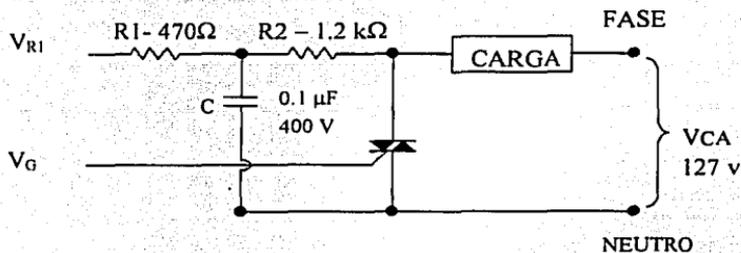
Se escogió el MAC218A6 por sus características eléctricas ya que maneja un voltaje pico a pico de 400 V y una corriente máxima de 8 Amperes, lo cual lo hace ideal para la aplicación de baja tensión (127 V) con la que se excita a la carga.

3.10.2. FUNCIONAMIENTO.

La operación de estas etapas es muy sencilla, porque se basa fundamentalmente en que el fototriac del MOC3022 conduzca o no; cuando este conduce se presenta teóricamente un corto circuito entre V_{R1} y V_G haciendo que circule una corriente I_G , que es la que dispara al TRIAC dando como resultado que la carga se active.

Caso contrario, cuando el fototriac del MOC3022 no conduce se presenta un circuito abierto entre V_{R1} y V_G por lo que no circula corriente alguna que dispare al TRIAC MAC 218, debido a lo anterior, la carga se encuentra inactiva.

3.10.3. DIAGRAMA.



3.11. CARGA.

3.11.1. MOTOR MONOFÁSICO DE RELUCTANCIA.

Este tipo de motor es el que se utiliza comúnmente en los ventiladores de los hogares. Los motores de reluctancia dependen del par de reluctancia para operar. El par de reluctancia es el par inducido en un objeto de hierro en presencia de un campo magnético externo, el cual obliga a que el objeto se alinee con dicho campo.

Este par ocurre cuando el campo externo induce un campo magnético interno en el hierro del objeto, ocasionando que aparezca un par entre los dos campos, con lo cual gira el objeto hasta alinearlos con el campo externo.

Para que se realice el par de reluctancia en un objeto, este debe de ser alargado axialmente en ángulos que correspondan a los ángulos entre polos adyacentes en el campo magnético.

3.11.2. FUNCIONAMIENTO.

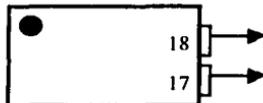
El funcionamiento del motor de reluctancia es sencillo, al ser un motor sincrónico el rotor está ligado a los campos magnéticos del estator hasta que se exceda el par máximo del motor. Así como todo motor sincrónico normal, este no cuenta con par de arranque por lo que no puede funcionar por sí solo. Este tipo de motor se puede implementar para que funcione como un motor de reluctancia de autoarranque, el cual operará a velocidad sincrónica hasta que exceda su par de reluctancia máximo al modificar el rotor de un motor de inducción. El rotor cuenta con polos salientes para operación en estado estacionario como un motor de reluctancia y cuenta también con rotor jaula de ardilla ó devanados de amortiguación para el arranque.

Al estar alimentado con una fuente de potencia monofásica y estar ligado a la operación del triac, deben esperar la operación del microcontrolador

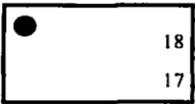
La operación de la carga es muy sencilla al ser dependiente del disparo del triac para su funcionamiento. La carga conectada a la línea monofásica de 127 Vca permanece en reposo hasta que la señal proveniente de los pines 17 y 18 del PIC llegan a la etapa acopladora de salida, la cual actúa como un interruptor al obtener teóricamente niveles lógicos de 1 y 0 volts.

De los pines 17 y 18 del PIC depende que el motor vaya a funcionar para las temperaturas sensadas;

- Temperatura baja.

		ESTADO LOGICO	NIVEL DE VOLTAJE	
	18	RA1/ENFRIAR	1	4.479
	17	RA1/CALENTAR	0	0.002

- Temperatura normal.

		ESTADO LOGICO	NIVEL DE VOLTAJE
	18	RAI/ENFRIAR	0
	17	RAI/CALENTAR	0

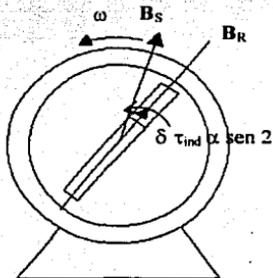
- Temperatura Alta.

		ESTADO LOGICO	NIVEL DE VOLTAJE
	18	RAI/ENFRIAR	0
	17	RAI/CALENTAR	1

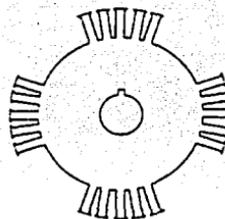
Con esta combinación de señales eléctricas cada uno de los dos motores recibe la orden para poder accionar y mantener una temperatura ambiente promedio.

En resumen los dos motores al ser del mismo tipo, con la única diferencia que uno cuenta con una resistencia encargada de calentar el aire a su salida actuando como un calefactor.

3.11.3. DIAGRAMA



Estator monofásico



Rotor

TESIS CON
FALLA DE ORIGEN

3.13. PRUEBAS.

Debido a que el circuito se diseña a partir de un diagrama a bloques, éste puede ser probado etapa por etapa, por lo que facilita la localización de fallas en los componentes o malas conexiones. A continuación se presentan tablas con los voltajes importantes de cada una de las etapas.

3.13.1. PRUEBA DE LA ETAPA SENSORA.

Los voltajes de esta etapa varían en función de la temperatura, los datos que aparecen en la tabla 3.10. corresponden a los voltajes de los pines del sensor LM335 y fueron tomados a una temperatura de 25°C.

PIN.	1	2	3
VOLTAJE	1.475	2.982	0

Tabla 3.10. Voltajes en la etapa sensora.

3.13.2. PRUEBA DE LA ETAPA COMPARADORA.

Los cursores de los potenciómetros que determinan el rango de temperatura se ajustan para efectos de prueba a una temperatura de 20°C y 30°C, correspondiéndoles un voltaje de 2.932 V y 3.032 V respectivamente. En la tabla 3.11-a, 3.11-b y 3.11-c se muestran los voltajes del amplificador operacional TL082 cuando la temperatura sensada tiene un valor bajo (menor a 20°C), normal (entre 20°C y 30°C) y alto (mayor a 30°C).

PIN	1	2	3	4	5	6	7	8
VOLTAJE	1.362	2.932	<2.982	0	3.032	<2.982	10.333	12

Tabla 3.11-a. Voltajes con una temperatura baja.

PIN	1	2	3	4	5	6	7	8
VOLTAJE	10.333	2.932	2.982	0	3.032	2.982	10.333	12

Tabla 3.11-b. Voltajes con una temperatura normal.

PIN	1	2	3	4	5	6	7	8
VOLTAJE	10.333	2.932	>2.982	0	3.032	>2.982	1.385	12

Tabla 3.11-c. Voltajes con una temperatura alta.

3.13.3. PRUEBA DE LA ETAPA ACOPLADORA DE ENTRADA.

Las tablas 3.12-a, 3.12-b y 3.12-c. presentan los voltajes en los dos optoacopladores de entrada (1 y 2), para las diferentes condiciones de temperatura sendadas y comparadas en las etapas anteriores.

PIN DEL OPTOACOPLADOR 1	1	2	3	4	5	6
VOLTAJE	1.362	0.388	0	0.194	5	0
PIN DEL OPTOACOPLADOR 2	1	2	3	4	5	6
VOLTAJE	10.333	9.252	0	4.307	5	5

Tabla 3.12-a. Voltajes con temperatura baja.

PIN DEL OPTOACOPLADOR 1	1	2	3	4	5	6
VOLTAJE	10.333	9.252	0	4.320	5	5
PIN DEL OPTOACOPLADOR 2	1	2	3	4	5	6
VOLTAJE	10.333	9.252	0	4.320	5	5

Tabla 3.12-b. Voltajes con temperatura normal.

PIN DEL OPTOACOPLADOR 1	1	2	3	4	5	6
VOLTAJE	10.333	9.252	0	4.320	5	5
PIN DEL OPTOACOPLADOR 2	1	2	3	4	5	6
VOLTAJE	1.385	0.388	0	0.219	5	0

Tabla 3.12-c. Voltajes con temperatura alta.

3.13.4. PRUEBA DE LA ETAPA DE CONTROL.

En la tabla 3.13. se muestran los voltajes en cada uno de los pines del microcontrolador, del pin 6 al 13 se da el rango de valores que toman estos, ya que su estado lógico varía dependiendo del mensaje.

BAJA		NORMAL		ALTA	
PIN	VOLTAJE	PIN	VOLTAJE	PIN	VOLTAJE
1	0.194	1	4.320	1	4.320
2	4.307	2	4.320	2	0.219
3	5	3	5	3	5
4	5	4	5	4	5
5	0	5	0	5	0
6	0.001-4.907	6	0.001-4.907	6	0.001-4.907
7	0.001-4.907	7	0.001-4.907	7	0.001-4.907
8	0.001-4.907	8	0.001-4.907	8	0.001-4.907
9	0.001-4.907	9	0.001-4.907	9	0.001-4.907
10	0.001-4.907	10	0.001-4.907	10	0.001-4.907
11	0.001-4.907	11	0.001-4.907	11	0.001-4.907
12	0.001-4.907	12	0.001-4.907	12	0.001-4.907
13	0.001-4.907	13	0.001-4.907	13	0.001-4.907
14	5	14	5	14	5
15	1.983	15	1.983	15	1.983
16	1.983	16	1.983	16	1.983
17	4.425	17	0.001	17	0.001
18	0.001	18	0.001	18	4.425

Tabla 3.13. Voltajes en el microcontrolador.

3.13.5. PRUEBA DE LA ETAPA ACOPLADORA DE SALIDA.

Las tablas 3.14-a, 3.14-b y 3.14-c. presentan los voltajes en los dos optoacopladores de salida (1 y 2), para las diferentes condiciones de temperatura a controlar.

PIN DEL OPTOACOPLADOR 1	1	2	3	4	5	6
VOLTAJE	5	3.265	0	0.574	1.041	0.915
PIN DEL OPTOACOPLADOR 2	1	2	3	4	5	6
VOLTAJE	0	0	0	0	6.275	115.54

Tabla 3.14-a. Voltajes con temperatura baja.

PIN DEL OPTOACOPLADOR 1	1	2	3	4	5	6
VOLTAJE	0	0	0	0	63.304	115.97
PIN DEL OPTOACOPLADOR 2	1	2	3	4	5	6
VOLTAJE	0	0	0	0	6.32	115.85

Tabla 3.14-b. Voltajes con temperatura normal.

PIN DEL OPTOACOPLADOR 1	1	2	3	4	5	6
VOLTAJE	0	0	0	0	62.261	113.399
PIN DEL OPTOACOPLADOR 2	1	2	3	4	5	6
VOLTAJE	5	3.260	0	0.715	0.524	0.992

Tabla 3.14-c. Voltajes con temperatura alta.

3.13.6. PRUEBA DE LA ETAPA DE POTENCIA.

El contenido de las tablas 3.15-a, 3.15-b y 3.15-c indica los voltajes en los triacs (1 y 2) para las diferentes condiciones de temperatura a controlar.

PIN DEL TRIAC 1	1	2	3
VOLTAJE	0.008	1.104	0.578
PIN DEL TRIAC 2	1	2	3
VOLTAJE	0.047	117.214	0.045

Tabla 3.15-a. Voltajes con una temperatura baja.

PIN DEL TRIAC 1	1	2	3
VOLTAJE	0	117.214	0
PIN DEL TRIAC 2	1	2	3
VOLTAJE	0	117.457	0

Tabla 3.15-a. Voltajes con una temperatura normal.

PIN DEL TRIAC 1	1	2	3
VOLTAJE	0	117.123	0
PIN DEL TRIAC 2	1	2	3
VOLTAJE	0	1.241	0.750

Tabla 3.15-a. Voltajes con una temperatura alta.

CAPÍTULO IV.

PROGRAMA Y GRABACIÓN.

4.1. PROGRAMA.

El programa se diseñó con el fin de monitorear y controlar temperatura, dadas las características del sensor de temperatura empleado esta temperatura es la temperatura ambiente.

El monitoreo se lleva a cabo por medio de mensajes que son mostrados en un display de siete segmentos que es manejado por el propio microcontrolador, para esto se utilizan ocho de los trece pines de E/S con los que cuenta el PIC16F84.

El control se realiza a través de dos señales que genera también el propio microcontrolador, para esto se utilizan dos de los trece pines de E/S con los que cuenta el PIC16F84. Nombraremos simbólicamente a cada una de estos dos pines como: CALIENTA y ENFRÍA.

Tanto el monitoreo como el control son el resultado del procesamiento que el microcontrolador efectúa sobre datos que constantemente recibe; para que el microcontrolador reciba datos se utilizan tres de los trece pines de E/S con los que cuenta el PIC16F84. Simbólicamente nombraremos a cada uno de estos pines como: MODO, DATO1 y DATO2.

Anteriormente se mencionó que el monitoreo y el control son resultado de un procesamiento; ahora bien, este procesamiento se basa en una tabla que fue elaborada teniendo en mente que es lo que se quiere que haga el microcontrolador y en que momento, a esta tabla se le llama "Tabla de Operación" (tabla 4.1.).

En esta tabla básicamente se expone la relación entre las entradas con las salidas, es decir, la relación entre MODO, DATO1 y DATO2 (las entradas) con CALIENTA, ENFRÍA y el mensaje (las salidas).

Otra manera de comprender el contenido de esta tabla es que en esta se indica que combinación de entradas generan determinadas salidas. A continuación se muestra dicha tabla.

MODO	DATO1	DATO2	CALIENTA	ENFRÍA	MENSAJE
0	0	0	DESACTIVADA	DESACTIVADA	IDENTIFICACION
0	0	1	DESACTIVADA	DESACTIVADA	IDENTIFICACION
0	1	0	DESACTIVADA	DESACTIVADA	IDENTIFICACION
0	1	1	DESACTIVADA	DESACTIVADA	IDENTIFICACION
1	0	0	DESACTIVADA	DESACTIVADA	FALLA.
1	0	1	ACTIVADA.	DESACTIVADA	BAJA.
1	1	0	DESACTIVADA	ACTIVADA.	ALTA.
1	1	1	DESACTIVADA	DESACTIVADA	NORMAL.

Tabla 4.1. "Tabla de Operación"

4.1.1. ALGORITMO.

El algoritmo con el cual el programa se elaboró consta de cuatro pasos y constituyen lo que sería la base o núcleo del programa. Estos cuatro pasos son efectuados secuencialmente por el microcontrolador cuando éste ejecuta el programa, dando como resultado que el microcontrolador efectúe la tarea o función para la cual fue programado.

El algoritmo es el siguiente:

1) Lectura y análisis de datos.- En este primer paso el microcontrolador recibe los datos y los analiza uno por uno, o sea, examina si hay un uno o un cero lógicos en cada una de los pines denominados MODO, DATO1 y DATO2.

2) Toma de decisiones.- Aquí el microcontrolador toma decisiones en base a los datos leídos y analizados en función de la tabla de operación, con lo que se prepara a mostrar el mensaje adecuado y a colocar un uno o un cero en los pines CALIENTA y ENFRÍA.

3) Monitoreo y control.- En este paso el microcontrolador muestra el mensaje correspondiente en el display de siete segmentos y pone un uno o un cero en los pines CALIENTA y ENFRÍA.

4) Iteración del algoritmo.- Una vez que se ha efectuado el paso anterior el microcontrolador vuelve a efectuar el primer paso del algoritmo ocasionando que este se repita una y otra vez; por lo tanto, el microcontrolador siempre está efectuando su tarea o función asignada.

4.1.2. LISTADO DEL CÓDIGO FUENTE.

El código fuente del programa fue estructurado de tal manera que fusiona y plasma en el dos ideas muy importantes: la tabla de operación y el algoritmo; además de estas dos ideas se anexan algunos elementos extras para complementar el programa.

A continuación se muestra el listado del código fuente en donde cada uno de los elementos (directivas, etiquetas e instrucciones) empleados en el código fuente contiene un comentario; el contenido de este comentario trata sobre la función o acción que desempeña el elemento en cuestión dentro del programa.

```

INCLUDE      <P16F84.INC>      ;anexa el archivo P16F84.INC
;al programa.
__CONFIG    _XT_OSC & _WDT_OFF & _PWRTE_ON & _CP_OFF
;configura al microcontrolador
;para usar un oscilador XT,
;habilitar PWRTE y
;deshabilitar WDT y CP.
TOYON      EQU      d'1'      ;etiqueta a 1 con TOYON.
EO        EQU      d'3'      ;etiqueta a 3 con EO.
PAYTA     EQU      d'5'      ;etiqueta a 5 con PAYTA.
PBYTB    EQU      d'6'      ;etiqueta a 6 con PBYTB.
TIEMPO    EQU      d'12'     ;etiqueta a 12 con TIEMPO.
bsf       EO,d'5'          ;accesa al banco de registros
;1.
movlw     b'11010110'      ;carga la configuración del
;registro OPTION.
movwf     TOYON            ;configura al registro OPTION
;para desconectar las
;resistencias de pull-up del
;puerto B, activar por flanco
;ascendente las interrupciones
;externas, activar como
;temporizador a TMR0, activar
;por flanco descendente los
;impulsos externos, asignar el
;divisor de frecuencias a
;TMR0 y ponerlo como divisor
;por 128.
movlw     b'11100'        ;carga la configuración del
;registro TRISA.
movwf     PAYTA           ;configura al registro TRISA
;para poner como entradas a
;RA2, RA3 y RA4 y como salidas
;a RA0 y RA1.
movlw     b'00000000'     ;carga la configuración del
;registro TRISB.
movwf     PBYTB          ;configura al registro TRISB
;para poner como salidas a
;RB0, RB1, RB2, RB3, RB4, RB5,
;RB6 y RB7.
bcf       EO,d'5'        ;regresa al banco de registros
;0.
bcf       PAYTA,d'0'     ;RA0 (CALIENTA) asume un cero.
bcf       PAYTA,d'1'     ;RA1 (ENFRIA) asume un cero.
movlw     b'11100110'    ;carga el caracter "H".
call     DISPLAY         ;llama a la subrutina DISPLAY.
movlw     b'10101111'    ;carga el caracter "O".
call     DISPLAY         ;llama a la subrutina DISPLAY.
movlw     b'00001110'    ;carga el caracter "L".
call     DISPLAY         ;llama a la subrutina DISPLAY.
movlw     b'11110111'    ;carga el caracter "A.".
call     DISPLAY         ;llama a la subrutina DISPLAY.
btfs     PAYTA,d'4'      ;lee y examina a RA4 (MOD0).
goto     DNTFCCN        ;se detectó un cero y salta a

```

**TESIS CON
FALLA DE ORIGEN**

INICIO

```

;DNTFCCN.
      btpsc      PAYTA,d'2'      ;se detectó un uno por lo que
      goto      ALNOR           ;lee y examina a RA2 (DATO1).
                                   ;se detectó un uno y salta a
                                   ;ALNOR.
      goto      BAFA           ;se detectó un cero por lo que
                                   ;salta a BAFA.
ALNOR  btffs     PAYTA,d'3'      ;lee y examina a RA3 (DATO2).
      goto      ALTA           ;se detectó un cero y salta a
                                   ;ALTA.
      goto      NORMAL        ;se detectó un uno por lo que
                                   ;salta a NORMAL.
BAFA   btpsc     PAYTA,d'3'      ;lee y examina a RA3 (DATO2).
      goto      BAJA          ;se detectó un uno y salta a
                                   ;BAJA.
      goto      FALLA         ;se detectó un cero por lo que
                                   ;salta a FALLA.
ALTA   movlw     b'11100111'     ;carga el caracter "A".
      call      DISPLAY         ;llama a la subrutina DISPLAY.
      movlw     b'00011110'     ;carga el caracter "L.".
      call      DISPLAY         ;llama a la subrutina DISPLAY.
      bcf       PAYTA,d'0'      ;RA0 (CALIENTA) asume un cero.
      bsf       PAYTA,d'1'      ;RA1 (ENFRIA) asume un uno.
      goto      INICIO         ;salta a INICIO.
BAJA   movlw     b'11101111'     ;carga el caracter "B".
      call      DISPLAY         ;llama a la subrutina DISPLAY.
      movlw     b'11110111'     ;carga el caracter "A.".
      call      DISPLAY         ;llama a la subrutina DISPLAY.
      bsf       PAYTA,d'0'      ;RA0 (CALIENTA) asume un uno.
      bcf       PAYTA,d'1'      ;RA1 (ENFRIA) asume un cero.
      goto      INICIO         ;salta a INICIO.
FALLA  bcf       PAYTA,d'0'      ;RA0 (CALIENTA) asume un cero.
      bcf       PAYTA,d'1'      ;RA1 (ENFRIA) asume un cero.
      movlw     b'01000111'     ;carga el caracter "F".
      call      DISPLAY         ;llama a la subrutina DISPLAY.
      movlw     b'11100111'     ;carga el caracter "A".
      call      DISPLAY         ;llama a la subrutina DISPLAY.
      movlw     b'00001110'     ;carga el caracter "L".
      call      DISPLAY         ;llama a la subrutina DISPLAY.
      movlw     b'00001110'     ;carga el caracter "L".
      call      DISPLAY         ;llama a la subrutina DISPLAY.
      movlw     b'11110111'     ;carga el caracter "A.".
      call      DISPLAY         ;llama a la subrutina DISPLAY.
      goto      INICIO         ;salta a inicio.
DNTFCCN bcf       PAYTA,d'0'      ;RA0 (CALIENTA) asume un cero.
      bcf       PAYTA,d'1'      ;RA1 (ENFRIA) asume un cero.
      movlw     b'11101011'     ;carga el caracter "9".
      call      DISPLAY         ;llama a la subrutina DISPLAY.
      movlw     b'10100000'     ;carga el caracter "1".
      call      DISPLAY         ;llama a la subrutina DISPLAY.
      movlw     b'10100000'     ;carga el caracter "1".
      call      DISPLAY         ;llama a la subrutina DISPLAY.
      movlw     b'10100000'     ;carga el caracter "1".
      call      DISPLAY         ;llama a la subrutina DISPLAY.

```

	movlw	b'11001101'	;carga el caracter "2".
	call	DISPLAY	;llama a la subrutina DISPLAY.
	movlw	b'10101111'	;carga el caracter "0".
	call	DISPLAY	;llama a la subrutina DISPLAY.
	movlw	b'11101111'	;carga el caracter "8".
	call	DISPLAY	;llama a la subrutina DISPLAY.
	movlw	b'01000000'	;carga el caracter "-".
	call	DISPLAY	;llama a la subrutina DISPLAY.
	movlw	b'01101111'	;carga el caracter "6".
	call	DISPLAY	;llama a la subrutina DISPLAY.
	movlw	b'01001100'	;carga el caracter "c".
	call	DISPLAY	;llama a la subrutina DISPLAY.
	movlw	b'01101100'	;carga el caracte "o".
	call	DISPLAY	;llama a la subrutina DISPLAY.
	movlw	b'01100100'	;carga el caracter "n".
	call	DISPLAY	;llama a la subrutina DISPLAY.
	movlw	b'11101011'	;carga el caracter "9".
	call	DISPLAY	;llama a la subrutina DISPLAY.
	movlw	b'10100000'	;carga el caracter "1".
	call	DISPLAY	;llama a la subrutina DISPLAY.
	movlw	b'11101001'	;carga el caracter "3".
	call	DISPLAY	;llama a la subrutina DISPLAY.
	movlw	b'11001101'	;carga el caracter "2".
	call	DISPLAY	;llama a la subrutina DISPLAY.
	movlw	b'01101111'	;carga el caracter "6".
	call	DISPLAY	;llama a la subrutina DISPLAY.
	movlw	b'11001101'	;carga el caracter "2".
	call	DISPLAY	;llama a la subrutina DISPLAY.
	movlw	b'10100011'	;carga el caracter "7".
	call	DISPLAY	;llama a la subrutina DISPLAY.
	movlw	b'01000000'	;carga el caracter "-".
	call	DISPLAY	;llama a la subrutina DISPLAY.
	movlw	b'10111111'	;carga el caracter "0".
	call	DISPLAY	;llama a la subrutina DISPLAY.
	goto	INICIO	;salta a INICIO.
NORMAL	bcf	PAYTA,d'0'	;RAO (CALIENTA) asume un cero.
	bcf	PAYTA,d'1'	;RAI (ENFRIA) asume un cero.
	movlw	b'01100100'	;carga el caracter "n".
	call	DISPLAY	;llama a la subrutina DISPLAY.
	movlw	b'01101100'	;carga el caracter "o".
	call	DISPLAY	;llama a la subrutina DISPLAY.
	movlw	b'01010100'	;carga el caracter "r".
	call	DISPLAY	;llama a la subrutina DISPLAY.
DISPLAY	goto	INICIO	;salta a INICIO.
	movwf	PBYTB	;muestra el caracter cargado.
	clrf	TIEMPO	;limpia el registro 12.
MSEGUNDO	clrf	TOYON	;limpia el registro TMR0 para ;reiniciar el conteo del ;temporizador.
MEDIO	btfs	TOYON,d'6'	;examina al sexto bit del ;registro TMR0.
	goto	MEDIO	;se detectó un cero y salta a ;MEDIO.
	incf	TIEMPO,1	;se detectó un uno por lo que

```

;han transcurrido 8.192E-3
;segundos e incrementa en uno
;al registro 12.
;examina al septimo bit del
;registro 12.
;se detecto un cero y salta a
;MSEGUNDO.
;se detectó un uno por lo que
;han transcurrido 1.048
;segundos y borra el caracter
;mostrado.
;limpia el registro 12.
;limpia el registro TMR0 para
;reiniciar el conteo del
;temporizador.
;examina al sexto bit del
;registro TMR0.
;se detectó un cero y salta a
;CUARTO.
;se detectó un uno por lo que
;han transcurrido 8.192E-3
;segundos e incrementa en uno
;al registro 12.
;examina al quinto bit del
;registro 12.
;se detecto un cero y salta a
;CSEGUNDO.
;se detecto un uno por lo que
;han transcurrido 262.144E-3
;segundos y regresa a donde se
;realizo el llamado.
;indica el fin del programa

                btfss      TIEMPO,d'7'
                goto       MSEGUNDO
                clrf       PBYTB
                clrf       TIEMPO
                clrf       TOYON
CSEGUNDO
                clrf       TOYON,d'6'
                goto       CUARTO
                incf       TIEMPO,1
                btfss      TIEMPO,d'5'
                goto       CSEGUNDO
                return
END

```

**TESIS CON
FALLA DE ORIGEN**

4.1.3. LISTADO DE LA LIBRERÍA P16F84.INC.

Esta librería sirve como encabezado de programa y la proporcionó el fabricante del microcontrolador, por lo que es la que se incluye dentro del código fuente.

La librería consta básicamente de etiquetas que sirven para definir configuraciones, registros y bits importantes o de más utilidad. Su función dentro del programa es la de proporcionar los valores numéricos necesarios en forma de de etiquetas (`_XT_OSC`, `_WDT_OFF`, `_PWRTE_ON` y `_CP_OFF`) a la directiva `__CONFIG`, para así configurar al microcontrolador.

A continuación se muestra el listado de la librería P16F84.INC tal y como fue proporcionada por el fabricante, lo anterior con el propósito de no alterar en nada el contenido de dicha librería.

```
LIST
; P16F84.INC Standard Header File, Version 2.00 Microchip Technology, Inc.
NOLIST
```

```
; This header file defines configurations, registers, and other useful bits of
; information for the PIC16F84 microcontroller. These names are taken to match
; the data sheets as closely as possible.
```

```
; Note that the processor must be selected before this file is
; included. The processor may be selected the following ways:
```

- ```
; 1. Command line switch:
; C:\MPASM MYFILE.ASM /PIC16F84
; 2. LIST directive in the source file
; LIST P=PIC16F84
; 3. Processor Type entry in the MPASM full-screen interface
```

```

;
; Revision History
;
;-----
```

```
;Rev: Date: Reason:
;2.00 07/24/96 Renamed to reflect the name change to PIC16F84.
;1.01 05/17/96 Corrected BADRAM map
;1.00 10/31/95 Initial Release
```

```

;
; Verify Processor
;
;-----
```

```
IFNDEF __16F84
MESSG "Processor-header file mismatch. Verify selected processor."
ENDIF
```

```

;
; Register Definitions
;
;-----
```

```
W EQU H'0000'
F EQU H'0001'
```

```
----- Register Files -----
```

```
INDF EQU H'0000'
TMRO EQU H'0001'
PCL EQU H'0002'
STATUS EQU H'0003'
```

TESIS CON  
FALLA DE ORIGEN

```

FSR EQU H'0004'
PORTA EQU H'0005'
PORTB EQU H'0006'
EEDATA EQU H'0008'
EEADR EQU H'0009'
PCLATH EQU H'000A'
INTCON EQU H'000B'

```

```

OPTION_REG EQU H'0081'
TRISA EQU H'0085'
TRISB EQU H'0086'
EECON1 EQU H'0088'
EECON2 EQU H'0089'

```

----- STATUS Bits -----

```

IRP EQU H'0007'
RP1 EQU H'0006'
RP0 EQU H'0005'
NOT_TO EQU H'0004'
NOT_PD EQU H'0003'
Z EQU H'0002'
DC EQU H'0001'
C EQU H'0000'

```

----- INTCON Bits -----

```

GIE EQU H'0007'
EEIE EQU H'0006'
T0IE EQU H'0005'
INTE EQU H'0004'
RBIE EQU H'0003'
T0IF EQU H'0002'
INTF EQU H'0001'
RBIF EQU H'0000'

```

----- OPTION Bits -----

```

NOT_RBPU EQU H'0007'
INTEDG EQU H'0006'
TOCS EQU H'0005'
TOSE EQU H'0004'
PSA EQU H'0003'
PS2 EQU H'0002'
PS1 EQU H'0001'
PS0 EQU H'0000'

```

----- EECON1 Bits -----

```

EEIF EQU H'0004'
WRERR EQU H'0003'
WRN EQU H'0002'
WR EQU H'0001'
RD EQU H'0000'

```

```

;
;
; RAM Definition
;

```

```
 _MAXRAM H'CF'
 _BADRAM H'07', H'50'-H'7F', H'87'
```

```

;
;
; Configuration Bits
;

```

```
_CP_ON EQU H'000F'
_CP_OFF EQU H'3FFF'
_PWRTE_ON EQU H'3FFF'
_PWRTE_OFF EQU H'3FFF'
_WDT_ON EQU H'3FFF'
_WDT_OFF EQU H'3FFB'
_LP_OSC EQU H'3FFC'
_XT_OSC EQU H'3FFD'
_HS_OSC EQU H'3FFE'
_RC_OSC EQU H'3FFF'
```

LIST

#### 4.1.4. ENSAMBLADO DEL PROGRAMA.

El programa se ensambló con el ensamblador MPASM versión 01.50, este ensamblador es desarrollado por el fabricante del microcontrolador y lo pone a disposición de los usuarios que utilizan sus productos.

El uso de este ensamblador es muy sencillo y sólo hay que asegurarse de tener en el mismo directorio o carpeta a los archivos que contienen al ensamblador y al código fuente del programa, a continuación se presentan las pantallas que aparecen cuando se ensambló el programa.

Esta es la primer pantalla (figura 4.1.) y aparece cuando se ejecuta el ensamblador (MPASM.EXE). Aquí lo primero que se hace es introducir el nombre y extensión del archivo que contiene el código fuente del programa, en este caso es TESIS.ASM; después hay que seleccionar el modelo de microcontrolador para el cual fue diseñado el programa, en este caso es el 16F84; las demás opciones que aparecen y que por lo tanto pueden ser alteradas se dejan tal y como están ya que vienen configuradas por default y no es necesario modificarlas, una vez hecho lo anterior se le indica al ensamblador que inicie el proceso de ensamblado.

TESIS CON  
FALLA DE ORIGEN

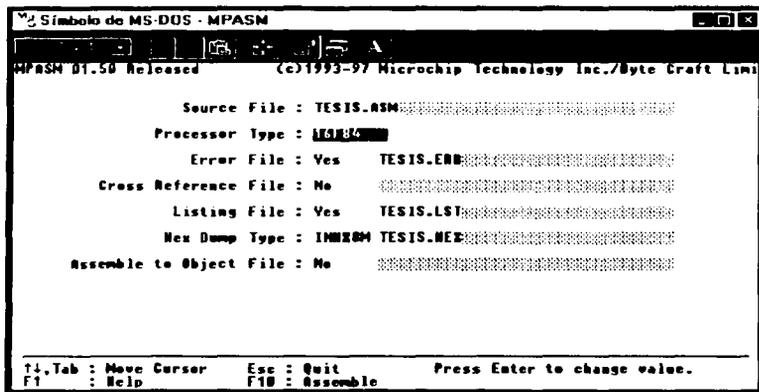


Figura 4.1. Menú del ensamblador.

Esta segunda y última pantalla (figura 4.2.) aparece cuando finaliza el proceso de ensamblado; en ella el ensamblador reporta cuantos errores (Errors) encontró en el código fuente e indica el número de líneas que ensambló, además señala la cantidad de advertencias (Warnings) y mensajes (Messages) que generó al terminar de ensamblar el programa. Como se puede observar, el ensamblador no encontró ningún error ni generó ninguna advertencia o mensaje.

```
Símbolo de MS-DOS - MPASM
MPASM 01.50 Released (c)1993-97 Microchip Technology Inc./Byte Craft Ltd
Checking C:\MPASM\TESIS.ASM for symbols...
Assembling...
TESIS.ASM 195
Building files...

Errors : 0
Warnings : 0 reported, 0 suppressed
Messages : 0 reported, 0 suppressed
Lines Assembled : 330

Press any key to continue.
```

Figura 4.2. Resultados del ensamblado.

#### 4.1.5 EL ARCHIVO DE ERRORES.

Al no reportar el ensamblador ningún error, advertencia o mensaje referentes con el código fuente del programa este archivo se encuentra vacío, es decir, el ensamblador lo crea de todas maneras aún cuando no vaya a contener nada.

TESIS CON  
FALLA DE ORIGEN

#### 4.1.6. LISTADO DEL ARCHIVO HEXADECIMAL.

El contenido de este archivo es el resultado del proceso de ensamblado del código fuente, por lo que este archivo ya contiene los códigos numéricos (en forma de números hexadecimales) para que el microcontrolador pueda ejecutar el programa.

A continuación se muestra el listado del archivo hexadecimal y como se puede observar únicamente hay números hexadecimales los cuales sólo entiende e interpreta el microcontrolador.

```
:100000008316D63081001C308500003086008312B4
:1000100005108510E6306E20AF306E200E306E2059
:10002000F7306E20051E3828051917281A28851D57
:100030001D286528851924282B28E7306E201E30BE
:100040006E20051085141228EF306E20F7306E20D8
:100050000514851012280510851047306E20E730F2
:100060006E200E306E200E306E20F7306E2012287B
:1000700005108510EB306E20A0306E20A0306E2071
:10008000A0306E20CD306E20AF306E20EF306E206D
:1000900040306E206F306E204C306E206C306E2001
:1000A00064306E20EB306E20A0306E20E9306E2080
:1000B000CD306E206F306E20CD306E20A3306E209C
:1000C00040306E20BF306E2012280510851064303D
:1000D0006E206C306E2054306E20122886008C0109
:1000E0008101011F71288C0A8C1F702886018C01E8
:0E00F0008101011F79288C0A8C1E78280800D7
:02400E00F13FB0
:00000001FF
```

#### 4.2. GRABACIÓN.

Una vez que se ha ensamblado el código fuente y por consecuencia creado el archivo hexadecimal, lo que sigue es grabar en la memoria de programa del microcontrolador el contenido de este archivo.

La manera más común de grabar un programa en un microcontrolador PIC16F84 es la que utiliza lo siguiente:

- 1) El puerto paralelo de una PC.
- 2) Un programa especial conocido como programa grabador.

3) Un circuito especialmente diseñado para operar en conjunto con el programa grabador, denominado usualmente circuito grabador.

Por el puerto paralelo de la PC entran y salen datos, salen cuando se está grabando el programa y entran cuando se verifica que el programa se grabó correctamente. El programa grabador establece la comunicación entre el microcontrolador y el puerto paralelo de la PC, esta comunicación hace posible la grabación del programa y la verificación de la grabación de este; el circuito grabador sirve como interfase entre el puerto paralelo de la PC y el microcontrolador.

#### 4.2.1. CIRCUITO GRABADOR.

La característica principal de este circuito grabador es la de que los componentes con los que cuenta son muy pocos, tan sólo son: un transistor, dos diodos, dos capacitores, cinco resistencias y un conector DB25 macho.

En la figura 4.3. se muestra el diagrama del circuito grabador y como se puede observar consta de escasos elementos, por lo que su implementación es sumamente sencilla. Es muy importante notar que el circuito necesita de dos fuentes de voltaje para su funcionamiento: una de entre 12 y 14 Volts y otra de 5 Volts.

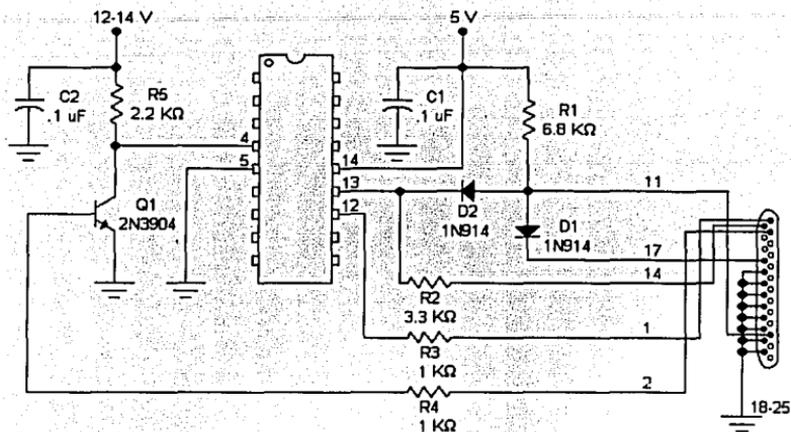


Figura 4.3. Circuito grabador.

#### 4.2.2. PROGRAMA GRABADOR.

El programa grabador empleado es uno llamado NOPPP, iniciales de No Parts PIC Programmer, que quiere decir Programador de PIC sin Partes, esto debido a que su circuito grabador dispone de muy pocos componentes; la versión utilizada fue la del 16 de Septiembre de 1998 a las 21:48:44 y Michael A. Covington es el desarrollador de este programa grabador.

El uso de este programa grabador es muy sencillo y sólo hay que asegurarse de tener en el mismo directorio o carpeta al archivo hexadecimal y al programa grabador, a continuación se presentan las pantallas que aparecen cuando se grabó el programa.

Antes que nada se tiene que conectar la PC con el circuito grabador, éste último sin microcontrolador y sin tener aplicado ninguno de los dos voltajes con los que opera.

Esta es la primer pantalla (figura 4.4.) y aparece al ejecutar el programa grabador (NOPPP.EXE). Aquí se debe de escoger el puerto paralelo que se va a utilizar para la grabación; generalmente las PC cuentan sólo con un puerto paralelo que es el LPT1, por lo que se selecciona éste.

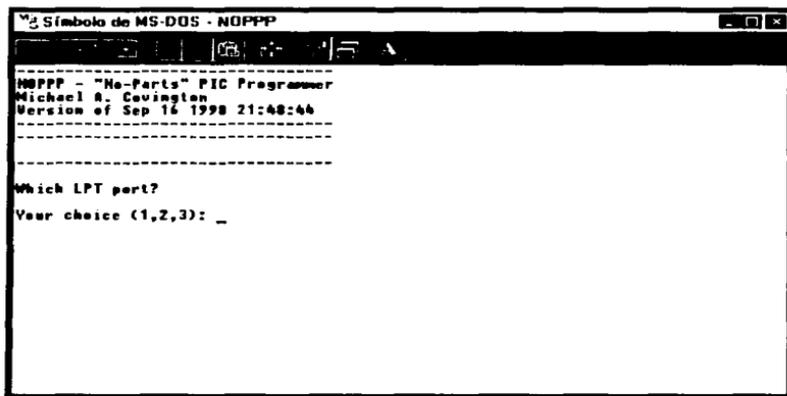


Figura 4.4. Selección de puerto.

La siguiente pantalla (figura 4.5.) avisa que se deben de aplicar los voltajes correspondientes al circuito grabador, pero sin colocarle aún el microcontrolador.

TESIS CON  
FALLA DE ORIGEN

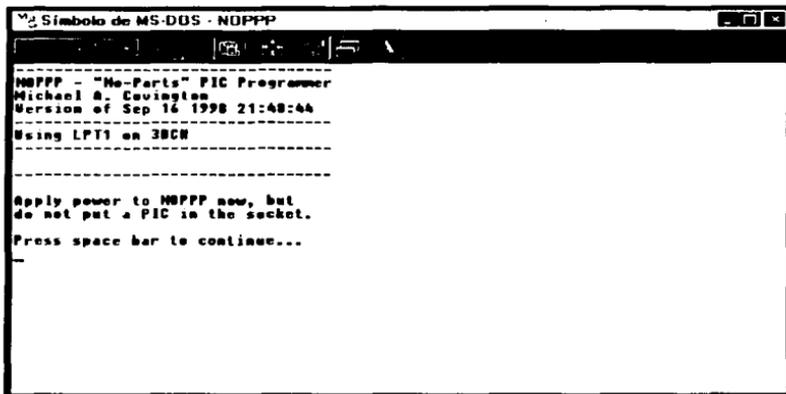


Figura 4.5. Aplicación de voltajes.

Dentro de esta pantalla (figura 4.6.) se puede seleccionar el modelo de microcontrolador a grabar o entrar a la prueba del circuito grabador, en este caso se selecciona el PIC16F84 que es el modelo a grabar.

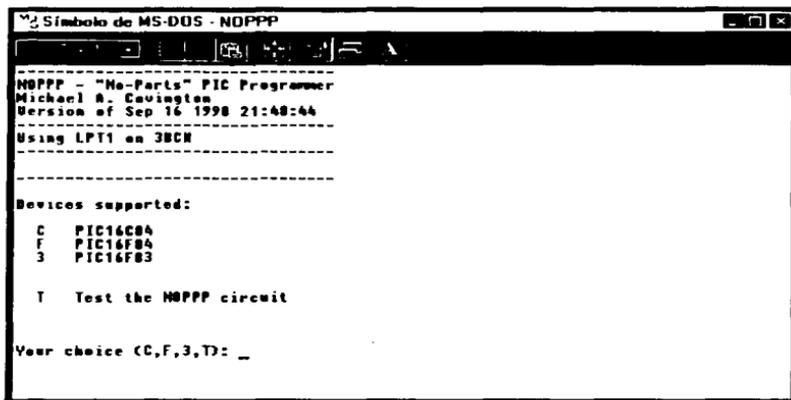


Figura 4.6. Menú primario.

TESIS CON  
FALLA DE ORIGEN

Aquí la pantalla (figura 4.7.) muestra un aviso de que ya se puede colocar el microcontrolador en el circuito grabador.

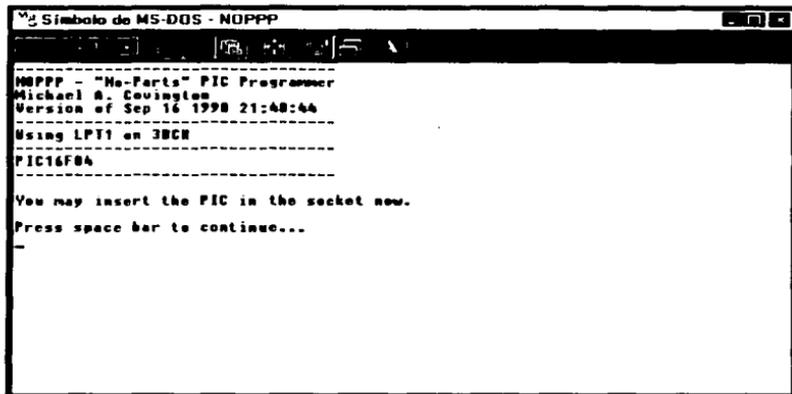
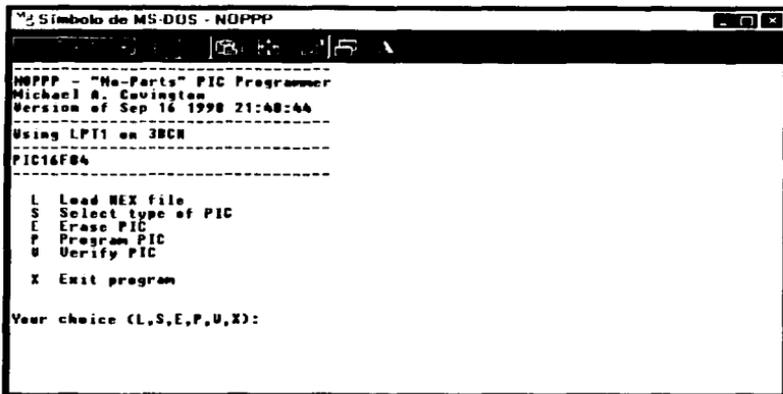


Figura 4.7. Inserción del microcontrolador.

Ahora en esta pantalla (figura 4.8.), a la cual llamaremos menú principal, se presentan opciones tales como: carga del archivo hexadecimal, selección del modelo de microcontrolador, borrado del microcontrolador, grabación del microcontrolador, verificación de la grabación y por último salida del programa grabador. Lo primero que se hace es seleccionar la carga del archivo hexadecimal.

TESIS CON  
FALLA DE ORIGEN



```

M$ Símbolo de MS-DOS - NOPPP

NOPPP - "No-Parts" PIC Programmer
Michael A. Covington
Version of Sep 16 1998 21:48:44

Using LPT1 on 3BC8

PIC16F84

L Load HEX file
S Select type of PIC
E Erase PIC
P Program PIC
V Verify PIC

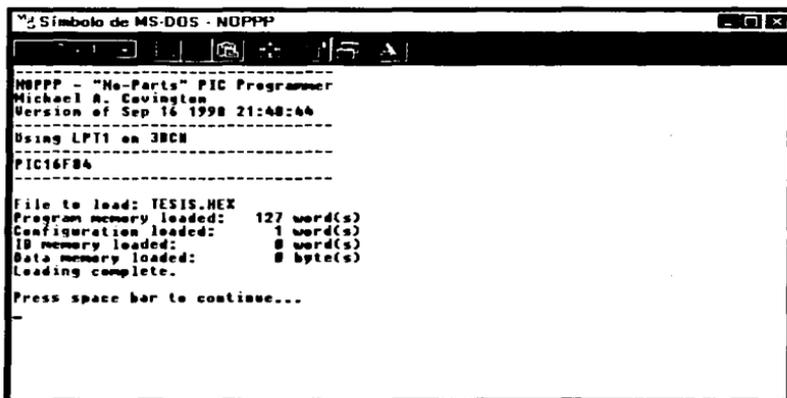
X Exit program

Your choice (L,S,E,P,U,X):

```

Figura 4.8. Menú principal.

Aquí se introduce el nombre y extensión del archivo hexadecimal (figura 4.9.), en este caso es TESIS.HEX, al hacer esto se indican las palabras cargadas de: programa, configuración, identificación y bytes de datos.



```

M$ Símbolo de MS-DOS - NOPPP

NOPPP - "No-Parts" PIC Programmer
Michael A. Covington
Version of Sep 16 1998 21:48:44

Using LPT1 on 3BC8

PIC16F84

File to load: TESIS.HEX
Program memory loaded: 127 word(s)
Configuration loaded: 1 word(s)
ID memory loaded: 8 word(s)
Data memory loaded: 8 byte(s)
Loading complete.

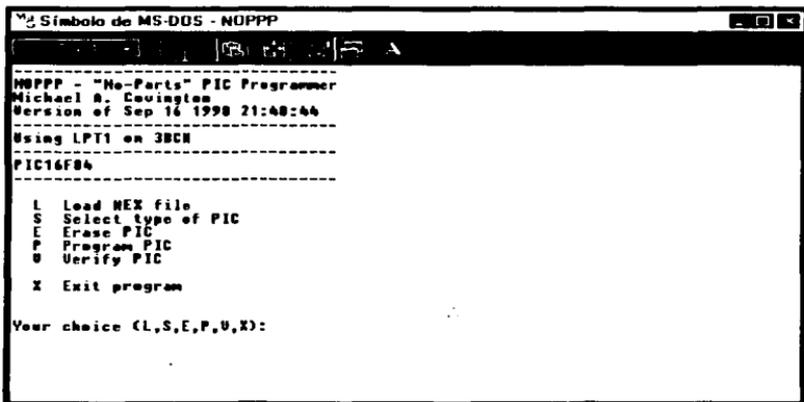
Press space bar to continue...
-

```

Figura 4.9. Carga del archivo hexadecimal.

TESIS CON  
FALLA DE ORIGEN

Al completarse la carga del archivo hexadecimal se regresa al menú principal (figura 4.10.) y lo que sigue es seleccionar borrar el microcontrolador.



```
Símbolo de MS-DOS - NUPPP

NUPPP - "No-Parts" PIC Programmer
Michael B. Covington
Version of Sep 16 1998 21:48:44

Using LPT1 on 3BC8

PIC16F84

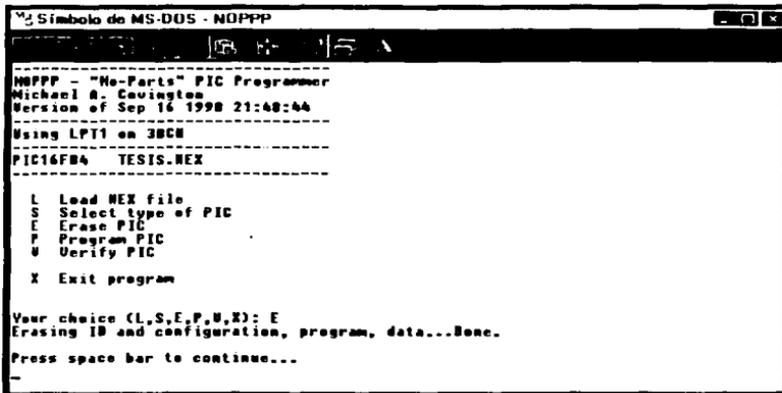
L Load HEX file
S Select type of PIC
E Erase PIC
P Program PIC
V Verify PIC
X Exit program

Your choice (L,S,E,P,V,X):
```

Figura 4.10. Menú principal.

En la parte inferior de la pantalla principal aparece un mensaje (figura 4.11.), este indica que se completó el borrado de la identificación, configuración, programa y datos del microcontrolador.

TESIS CON  
FALLA DE ORIGEN



```

Simbolo de MS-DOS - NOPPP

NOPPP - "No-Parts" PIC Programmer
Michael A. Covington
Version of Sep 16 1998 21:48:44

Using LPT1 on 386H

PIC16F84 TESIS.NEX

L Load HEX file
S Select type of PIC
E Erase PIC
P Program PIC
V Verify PIC

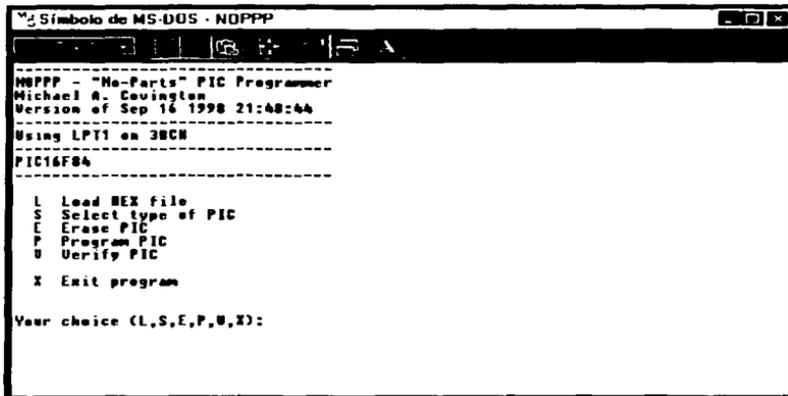
X Exit program

Your choice (L,S,E,P,U,X): E
Erasing ID and configuration, program, data...Done.
Press space bar to continue...

```

Figura 4.11. Borrado.

Al completarse la operación de borrado del microcontrolador nuevamente se regresa al menú principal (figura 4.12.). A continuación se selecciona la opción de grabar, el programa grabador usa el término Program PIC.



```

Simbolo de MS-DOS - NOPPP

NOPPP - "No-Parts" PIC Programmer
Michael A. Covington
Version of Sep 16 1998 21:48:44

Using LPT1 on 386H

PIC16F84

L Load HEX file
S Select type of PIC
E Erase PIC
P Program PIC
V Verify PIC

X Exit program

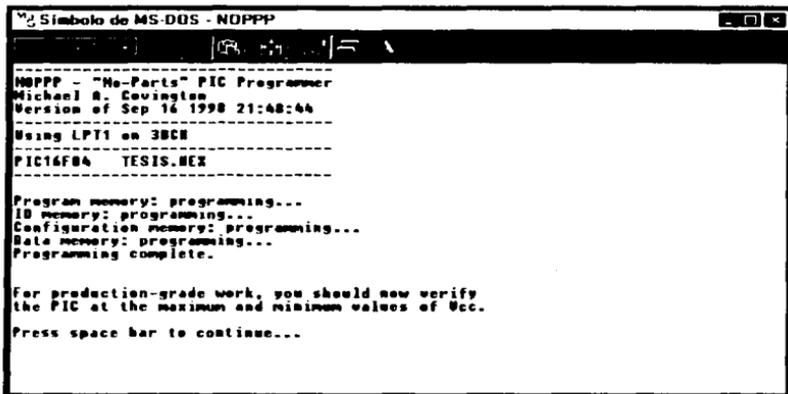
Your choice (L,S,E,P,U,X):

```

Figura 4.12. Menú principal.

TESIS CON  
FALLA DE ORIGEN

El proceso de grabación se inicia y al terminarse, el programa grabador reporta las palabras (programa, identificación y/o configuración) y datos que grabó (figura 4.13.); además, recomienda verificar la grabación a diferentes voltajes de operación del microcontrolador.



```

Símbolo de MS-DOS - NOPPP

NOPPP - "No-Parts" PIC Programmer
Michael A. Covington
Version of Sep 16 1998 21:48:44

Using LPT1 on 3BC8

PIC16F84 TESIS.HEX

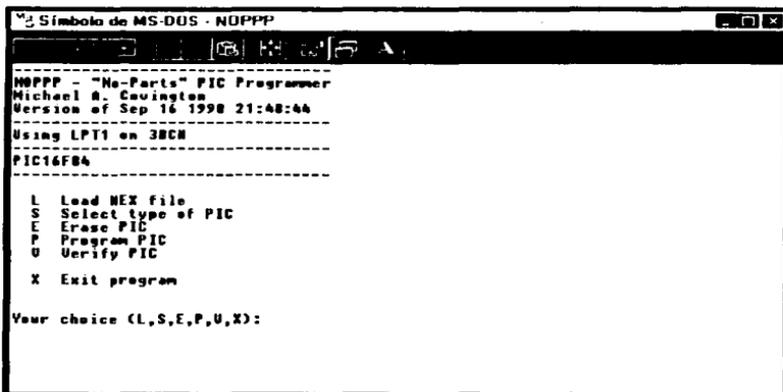
Program memory: programming...
ID memory: programming...
Configuration memory: programming...
Data memory: programming...
Programming complete.

For production-grade work, you should now verify
the PIC at the maximum and minimum values of Vcc.
Press space bar to continue...

```

Figura 4.13. Grabación.

Al completarse la grabación se retorna nuevamente al menú principal (figura 4.14.) y sólo hace falta verificar la grabación del microcontrolador, el programa grabador utiliza el término Verify PIC, por lo que se escoge esta opción.



```
MS-DOS - NOPPP

NOPPP - "No-Parts" PIC Programmer
Michael B. Covington
Version of Sep 16 1998 21:48:44

Using LPT1 on 3BC8

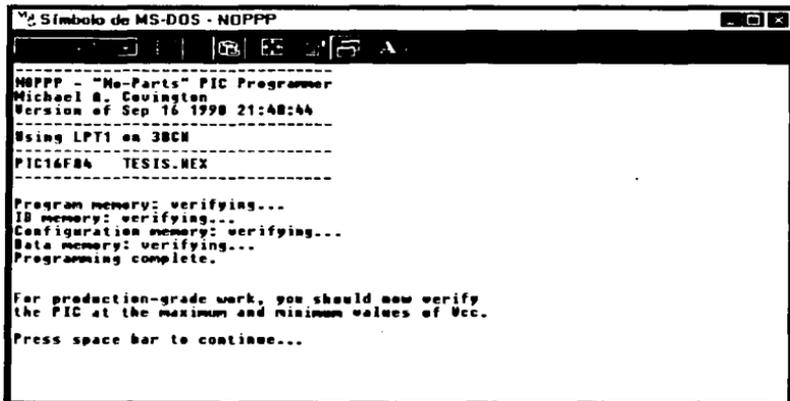
PIC16F84

L Load HEX file
S Select type of PIC
E Erase PIC
P Program PIC
U Verify PIC
X Exit program

Your choice (L,S,E,P,U,X):
```

Figura 4.14. Menú principal.

La verificación da inicio y termina presentando un mensaje (figura 4.15.) en donde indica que verificó la memoria de programa, las palabras de identificación y configuración, además de los bytes de datos.



```
MS-DOS - NOPPP

NOPPP - "No-Parts" PIC Programmer
Michael B. Covington
Version of Sep 16 1998 21:48:44

Using LPT1 on 3BC8

PIC16F84 TESIS.HEX

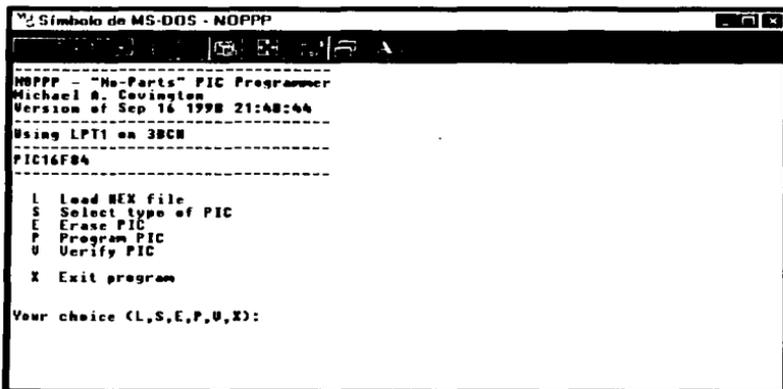
Program memory: verifying...
ID memory: verifying...
Configuration memory: verifying...
Data memory: verifying...
Programming complete.

For production-grade work, you should now verify
the PIC at the maximum and minimum values of Vcc.

Press space bar to continue...
```

Figura 4.15. Verificación.

Cuando se finaliza la verificación otra vez se regresa al menú principal (figura 4.16.) y se elige la opción salir.



```
Símbolo de MS-DOS - NOPPP

NOPPP - "No-Parts" PIC Programmer
Michael A. Covington
Version of Sep 16 1998 21:48:44
Using LPT1 on 3BC8

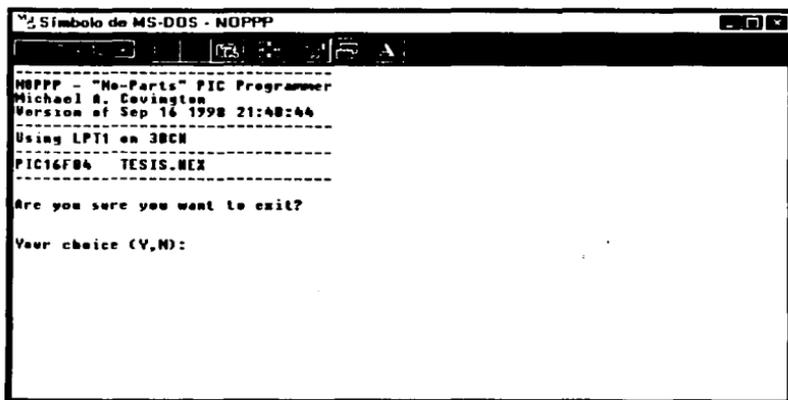
PIC16F04

L Load HEX file
S Select type of PIC
E Erase PIC
P Program PIC
U Verify PIC
X Exit program

Your choice (L,S,E,P,U,X):
```

Figura 4.16. Menú principal.

Ahora aparece la pregunta de si en verdad se desea salir (figura 4.17.).



```
Símbolo de MS-DOS - NOPPP

NOPPP - "No-Parts" PIC Programmer
Michael A. Covington
Version of Sep 16 1998 21:48:44
Using LPT1 on 3BC8

PIC16F04 TESIS.HEX

Are you sure you want to exit?

Your choice (Y,N):
```

Figura 4.17. Confirmación.

Finalmente aparece un mensaje (figura 4.18.) que indica que ya se puede quitar el microcontrolador del circuito grabador.

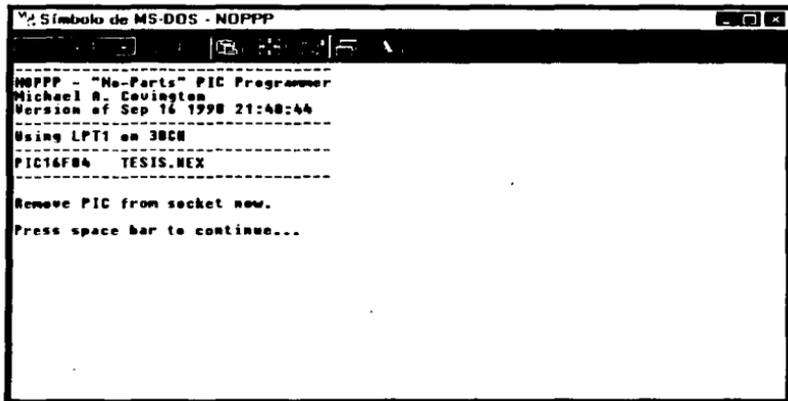


Figura 4.18. Retiro del microcontrolador.

Una vez que ya ha sido grabado el programa únicamente hace falta conectar el microcontrolador a la etapa correspondiente, de esta manera ya se tiene completo el circuito de control de temperatura con el PIC16F84.

## CONCLUSIONES.

El controlar una temperatura a través de un circuito que tenga como componente principal a un microcontrolador implicó resolver tres problemas fundamentales. El primero fue medir y regular una variable física como lo es la temperatura ambiente; el segundo fue el buscar, seleccionar y adecuar una serie de dispositivos para que asistan y/o complementen al elemento base del circuito y el tercero fue conocer y programar al microcontrolador.

Los dos primeros problemas fueron resueltos gracias a que en la actualidad existe una amplia variedad de componentes electrónicos en el mercado que cumplen con las necesidades del circuito diseñado (sensores, amplificadores operacionales, optoacopladores y triacs), por lo que solo fue necesario seleccionarlos de acuerdo a las características operativas y eléctricas requeridas.

El tercer problema, el más difícil de resolver, requirió de investigar todo lo referente al microcontrolador como: arquitectura interna (registros, memoria de datos y de programa, puertos de entrada y salida, temporizadores, etc.), características eléctricas (voltaje de alimentación, frecuencias de operación, corrientes suministradas y soportadas en los puertos, etc.) y lo más importante, como se programa.

El microcontrolador puede ser programado con dos tipos de lenguajes; uno de alto nivel (BASIC o C) y otro de bajo nivel (lenguaje ensamblador), cada uno de ellos tiene ventajas y desventajas que pueden llegar a ser críticas cuando se desarrolla un programa, como lo son: tiempo y herramientas de desarrollo, velocidad de ejecución, facilidad de programación, costo y dimensión del programa, etc.

Por lo anterior se optó por programar en lenguaje ensamblador, dando como ventajas una velocidad de ejecución óptima ya que el programa es compacto, debido a que las instrucciones en lenguaje ensamblador son concretas y específicas.

La desventaja de programarlo en lenguaje ensamblador, fue que consumió un tiempo de desarrollo relativamente largo ya que se tuvo que indagar y estudiar sobre: el juego de instrucciones del microcontrolador, sintaxis de las instrucciones, organización del código fuente, etc.

El circuito de control de temperatura funciona acorde a lo planeado, que es regular la temperatura de un espacio cerrado; la desventaja de este circuito es que emplea como medio visual un display de siete segmentos, lo cual limita en demasía el despliegue de los caracteres que componen los mensajes que indican el estado de la temperatura o una falla en el circuito.

En un momento dado se pensó en utilizar un display LCD (Liquid Cristal Display), pero esta medida implicaría un incremento considerable en el costo global del circuito; además de que el microcontrolador PIC16F84 emplea en este circuito todos sus pines de entrada/salida, por lo que le harían falta pines de este tipo para manejar un display LCD.

El costo total del circuito no supera los \$30.00 usd, siendo obviamente el componente de mayor costo el microcontrolador (aproximadamente unos \$6.00 usd), esto sin contar con la carga, ya que esta puede ser sustituida por el usuario según la necesidad de la aplicación.

Las herramientas de desarrollo para el programa (el ensamblador y el programa grabador) no implicaron ningún costo, lo cual representa una ventaja esencial dentro del costo total del circuito; estas dos herramientas se obtuvieron de Internet.

Al estar relacionada la temperatura directa e indirectamente en una infinidad de procesos y/o aplicaciones de diversa índole, este circuito tiene la posibilidad de adaptarse a otras aplicaciones; las cuales pueden servir como trabajos de tesis.

## **BIBLIOGRAFIA.**

### **AMPLIFICADORES OPERACIONALES Y CIRCUITOS INTEGRADOS LINEALES.**

Robert F. Coughlin.  
Frederick F. Driscoll.  
Prentice Hall.

### **ELECTRÓNICA: TEORÍA DE CIRCUITOS.**

Robert L. Boylestad.  
Louis Nashelsky.  
Prentice Hall.

### **LÓGICA DIGITAL Y DISEÑO DE COMPUTADORES.**

M. Morris Mano.  
Prentice Hall.

### **MÁQUINAS ELÉCTRICAS.**

Stephen J. Chapman.  
Mc Graw Hill.

### **MICROCONTROLADORES PIC, DISEÑO PRÁCTICO DE APLICACIONES.**

José Maria Angulo Usategui.  
Ignacio Angulo Martínez.  
Mc Graw Hill.

### **CONSULTAS A INTERNET.**

[www.nationalsemiconductor.com](http://www.nationalsemiconductor.com)  
[www.microchip.com](http://www.microchip.com)  
[www.mindspring.com/~covington/noppp](http://www.mindspring.com/~covington/noppp)  
[www.motorola.com](http://www.motorola.com)  
[www.telefunken.com](http://www.telefunken.com)  
[www.ti.com](http://www.ti.com)