



UNIVERSIDAD NACIONAL DE LOJA

**AREA DE LA ENERGÍA, LAS INDUSTRIAS Y LOS RECURSOS NATURALES NO
RENOVABLES**

**“IMPLEMENTACIÓN DEL LABORATORIO DE SISTEMAS
MICROPROCESADOS”**

*Memoria del trabajo práctico previo a
la obtención del título de Tecnólogos
en Electrónica.*

AUTORES:

**CRISTIAN ULPIANO ORTEGA GUAMÁN
JORGE FABIAN QUEZADA MALDONADO
FRANKLIN EDUARDO VILLALTA ALVARADO**

DIRECTOR:

ING. DIEGO VINICIO ORELLANA VILLAVICENCIO

ASESORES:

ING. JAIME REMIGIO JARAMILLO FEBRES

LOJA - ECUADOR

2009

CERTIFICACIÓN**Ing. DIEGO VINICIO ORELLANA VILLAVICENCIO****CERTIFICA:**

Haber dirigido y asesorado la tesis titulada "Implementación del Laboratorio de Sistemas Microprocesados", en la carrera de Tecnología en Electrónica, Área de energía, las Industrias y los Recursos Naturales no Renovables de la Universidad Nacional de Loja, realizada por los señores Cristian Ulpiano Ortega, Jorge Fabián Quezada y Franklin Eduardo Villalta, el mismo que se realizó bajo mi supervisión teórica y práctica, por lo cual autorizo su presentación y sustentación.

Loja, Octubre del 2009

Ing. DIEGO VINICIO ORELLANA VILLAVICENCIO
DIRECTOR

AUTORÍA

El contenido de este trabajo es de total responsabilidad de los autores y su copia total o parcial debe ser autorizada por los mismos.

Cristian Ulpiano Ortega Guamán

Jorge Fabián Quezada Maldonado

Franklin Eduardo Villalta Alvarado

AGRADECIMIENTO

A la Universidad Nacional de Loja y especialmente al Área de Energía, las Industrias y los Recursos Naturales no Renovables por haber generado y ejecutado el presente trabajo de investigación. Implementación del Laboratorio de Sistemas Microprocesados.

Al Director y Asesores de la Investigación, por sus orientaciones y tiempo proporcionado para la elaboración del presente trabajo.

A los directivos, docentes y estudiantes de la Carrera de Tecnología en Electrónica del Área de Energía, Industrias y los Recursos Naturales no Renovables, por la colaboración brindada en el proceso de investigación.

DEDICATORIA

Cristian

A Dios, a mis padres y todas las personas quienes de una u otra forma aportaron con un granito de arena para que este trabajo llegue a tener su culminación satisfactoria.

Jorge

Este trabajo esta dedicado a mi familia quienes me han apoyado siempre y gracias a ellos he podido alcanzar algunos retos que me he propuesto.

Franklin

El presente trabajo esta dedicado a mi querida madre quien con su apoyo incondicional supo darme ánimos para alcanzar este reto.

RESUMEN

El microprocesador es uno de los logros más sobresalientes del siglo XX, debido a su amplia aplicación en el mundo de la electrónica, razón por la cual se ha creído conveniente implementar el laboratorio de microcontroladores. El presente trabajo abarca todo lo relativo al proceso de programación de las familias de microcontroladores PIC (familias 16F84A y 16F877A). Este programador está constituido por el hardware y su respectivo software, el enlace entre ambos es mediante el puerto serial. Además se aprovecha la energía que provee el puerto, eliminándose así la necesidad de una fuente externa. El microcodestudio es un programa editor de texto está hecho para facilitar la programación de los microcontroladores PIC, luego se adjunta al lc-prog el mismo que es el lenguaje ensamblador o de máquina.

ABSTRACT

The microprocessor is one of the most outstanding achievements of the twentieth century, due to its wide application in the electronics world, which is why it proved necessary to implement the laboratory of microcontrollers. This work covers all aspects relating to the process of programming PIC microcontroller families (families 16F84A and 16F877A). This program is made by the respective hardware and software, the link between them is through the serial port. Also use the energy provided by the port, thus eliminating the need for an external source. The microcod-studio is a text editor program is meant to facilitate the programming of PIC microcontrollers, and then attached to the IC-Prog is the same as assembly language or machine.

INDICE

TEMA	INDICE
CERTIFICACIÓN	ii
AUTORÍA	iii
AGRADECIMIENTO	iv
DEDICATORIA	v
INTRODUCCIÓN	1
 CAPITULO I	
1. Estudio de los Pic's	2
1.1 Estudio General de un Pic	2
1.2 Estudio General del Pic16f84A	5
1.2.1 Descripción de sus elementos y pines	6
1.2.2 Diagrama de bloques	9
1.3 Estudio General del Pic16f877A	9
1.3.1 Descripción de sus elementos y pines	11
1.3.2 Diagrama de bloques	16
 CAPITULO II	
2. Software de Programación.	17
2.1 Instalación y manejo del Software	17
2.2 Manejo del Micro-Code-Studio	20
2.3 Declaraciones disponibles en el programa PBP2.4	23
 CAPITULO III	
3. Programación de las Prácticas.	25
3.1 Manejo del Puerto b.	25
3.1.1 Diagrama de flujo	25
3.1.2 Programación en micro-code-studio	25
3.2 Juego de Luces	26
3.2.1 Diagrama de Flujo	26
3.2.2 Programación en micro-code-studio	26
3.3 Contador con Displays de 7 segmentos	27
3.3.1 Diagrama de flujo	27
3.3.2 Programación en micro-code-studio	27
3.4 Modulación por Ancho de Pulsos	29
3.4.1 Diagrama de Flujo	29
3.4.2 Programación en micro-code-studio	31

3.5 Manejo de un LCD	32
3.5.1 Diagrama de flujo	32
3.5.2 Programación en micro-code-studio	33
3.6 Manejo del teclado hexadecimal	34
3.6.1 Diagrama de flujo	34
3.6.2 Programación en micro-code-studio	36
3.7 Manejo de un motor de paso	39
3.7.1 Diagrama de flujo	39
3.7.2 Programación en micro-code-studio	39
3.8 Comunicación Serial Pic a Pc y Pc a Pic	41
3.8.1 Diagrama de Flujo	41
3.8.2 Programación en micro-code-studio	41
3.9 Termómetro Digital	43
3.9.1 Diagrama de Flujo	43
3.9.2 Programación en micro-code-studio	44

CAPITULO IV

4 Elaboración de los PCB'S	45
4.1 Diseño para los displays de 7 segmentos y barra de leds	45
4.2 Diseño para el motor en DC	45
4.3 Diseño para el motor de paso	46
4.4 Diseño para la comunicación serial	46
4.5 Diseño para el teclado hexadecimal y el LCD	47
4.6 Diseño para el termómetro digital	47

CAPITULO V

5 Guía de las Prácticas.	48
5.1Preparatorio de la Práctica # 1	48
5.2 Práctica # 1	48
5.3 Preparatorio de la Práctica # 2	51
5.4 Práctica # 2	51
5.5 Preparatorio de la Práctica # 3	54
5.6 Práctica # 3	54
5.7 Preparatorio de la Práctica # 4	57
5.8 Práctica # 4	57
5.9 Preparatorio de la Práctica # 5	60
5.10 Práctica # 5	60
5.11Preparatorio de la Práctica # 6	63
5.12 Práctica # 6	63
5.13 Preparatorio de la Práctica # 7	67
5.14 Práctica # 7	67
5.15Preparatorio de la Práctica # 8	69

5.16 Práctica # 8	69
5.17Preparatorio de la Práctica # 9	72
5.18 Práctica # 9	72
CAPITULO VI	
6.1 Conclusiones	75
6.2 Recomendaciones	76
BIBLIOGRAFIA	77
ANEXOS	78

INTRODUCCIÓN

Se denomina microcontrolador a un dispositivo programable capaz de realizar diferentes actividades que requieran del procesamiento de datos digitales y del control y comunicación digital de diferentes dispositivos. Por ejemplo, el controlador que regula el funcionamiento de un horno dispone de un sensor que mide constantemente su temperatura interna y cuando traspasa los límites prefijados, genera las señales adecuadas que accionan los efectores que intentan llevar el valor de la temperatura dentro del rango estipulado.

Los microcontroladores poseen una memoria interna que almacena dos tipos de datos; las instrucciones, que corresponden al programa que se ejecuta, y los registros, es decir, los datos que el usuario maneja, así como registros especiales para el control de las diferentes funciones del microcontrolador.

El presente trabajo abarca todo lo relativo al proceso de programación de las familias de los microcontroladores PIC (familias 16F84A y 16F877A). Este proceso está constituido por hardware y su respectivo software, el enlace entre ambos se lo realiza mediante el puerto serial, aprovechando de esta manera la energía que provee el puerto, eliminando así, la necesidad de una fuente externa.

Para cubrir todas las áreas concernientes al proceso de programación de un determinado grupo de elementos de las familias de microcontroladores PIC y poder lograr este fin, se ha desarrollado un programador cuya interfaz de comunicación es a través del puerto Serial DB9, el cual constituye todo un protocolo de comunicaciones, y por lo tanto involucra un cierto grado de complejidad.

Así, la estructura del presente trabajo incluye: En el Capítulo 1 se ha visto la necesidad de poner énfasis en la explicación de la descripción de cada uno de los pines de los PICs en estudio. En el Capítulo 2 comprende los argumentos de programación de un microprocesador, que incluye aspectos como el conocer debidamente la forma como se han de escribir, leer y borrar sus respectivas declaraciones, así como, el manejo e instalación del software utilizado. Luego, en el Capítulo 3 se considera todo lo relativo al diseño y programación de prácticas orientadas a la aplicación y manejo de microprocesadores y sus componentes, las mismas que han sido comprobadas y ejecutadas. Dentro del Capítulo 4 se presenta el diseño y elaboración de los Circuitos y PCBS y las prácticas respectivas que faciliten su conocimiento, manejo y aplicación. En el Capítulo 5, se presentan conclusiones y recomendaciones obtenidas en base a la aplicación práctica en lo concerniente a este proyecto de titulación. Para finalmente, en el Capítulo 6 dar a conocer la guía de prácticas donde el estudiante con o sin conocimiento o experiencia tenga un apoyo adicional al momento de realizar las prácticas.

CAPITULO I

1. ESTUDIO DE LOS PIC'S

1.1 Estudio General de un PIC.

Los microcontroladores poseen principalmente una ALU (Unidad Lógico Aritmética), memoria del programa, memoria de registros, y pines I/O (entrada y/o salida). La ALU es la encargada de procesar los datos dependiendo de las instrucciones que se ejecuten (ADD, OR, AND), mientras que los pines son los que se encargan de comunicar al microcontrolador con el medio externo; la función de los pines puede ser de transmisión de datos, alimentación de corriente para el funcionamiento de este o pines de control específico.

Algunas de estas características se muestran a continuación:

- Soporta modo de comunicación serial, posee dos pines para ello.
- Amplia memoria para datos y programa.
- Memoria reprogramable: La memoria en este PIC es la que se denomina FLASH; este tipo de memoria se puede borrar electrónicamente (esto corresponde a la "F" en el modelo).
- Set de instrucciones reducidas (tipo RISC), pero con las instrucciones necesarias para facilitar su manejo.

Un microcontrolador es un circuito integrado de alta escala de integración que incorpora la mayor parte de los elementos que configuran un controlador.

Un microcontrolador dispone normalmente de los siguientes componentes:

- Procesador o UCP (Unidad Central de Proceso).
- Memoria RAM para almacenar los datos.
- Memoria para el programa tipo ROM (Memoria de solo Lectura)/EPROM (Memoria de solo lectura programable y borrrable eléctricamente).
- Líneas de E/S para comunicarse con el exterior.
- Diversos módulos para el control de periféricos (temporizadores, Puertas Serie y Paralelo, CAD: Conversores Analógico/Digital, CDA: Conversores Digital/Analógico, etc.).
- Generador de impulsos de reloj que sincronizan el funcionamiento de todo el sistema.

Los PIC que para su regulación incorporan un microcontrolador disponen de las siguientes ventajas:

Aumento de prestaciones: un mayor control sobre un determinado elemento representa una mejora considerable en el mismo.

Aumento de la fiabilidad: al reemplazar el microcontrolador por un elevado número de elementos disminuye el riesgo de averías y se precisan menos ajustes.

Reducción del tamaño en el producto acabado: La integración del microcontrolador en un chip disminuye el volumen, la mano de obra y los stocks.

Mayor flexibilidad: las características de control están programadas por lo que su modificación sólo necesita cambios en el programa de instrucciones.

El microcontrolador es en definitiva un circuito integrado que incluye todos los componentes de un computador. Debido a su reducido tamaño es posible montar el controlador en el propio dispositivo al que gobierna. En este caso el controlador recibe el nombre de controlador empotrado (embedded controller).

Organización de la memoria

En primer lugar tenemos que distinguir claramente entre tres tipos de memoria:

- Uno: la memoria EEPROM (Memoria de solo lectura programable y borrrable eléctricamente) donde almacenaremos el programa que haremos, esta memoria solo podrá ser leída por el PIC (el PIC va leyendo las instrucciones del programa almacenado en esta memoria y las va ejecutando). Al apagar el PIC, esta memoria no se borra.
- Dos: la memoria RAM en cuyos registros se irán almacenando los valores de las variables que nosotros queramos y cuando nosotros queramos (por programa), al apagar el PIC esta memoria se borra.
- Tres: la memoria programable para datos, es un espacio de memoria EEPROM en la que se pueden guardar variables que queremos conservar aunque se apague el PIC. No ha sido empleada en este trabajo por su dificultad de uso.

La memoria EEPROM o memoria de programa

El espacio de memoria donde irá nuestro programa, comprende las direcciones de memoria desde la 0000h hasta 3FFh que hacen 1024 direcciones, es decir, 1Kbyte)

"Reset Vector" es la primera dirección a la que se dirige el pic al encenderlo o al resetearlo.

"PC" y los "Stack Level" son empleados por el pic y nosotros no tenemos acceso a ellos.

registro 0Ch al 0Dh no podremos hacerlo directamente, deberemos usar una instrucción para cargar el valor del registro 0Ch en el acumulador y después otra instrucción para cargar el valor del acumulador en el registro 0Bh.

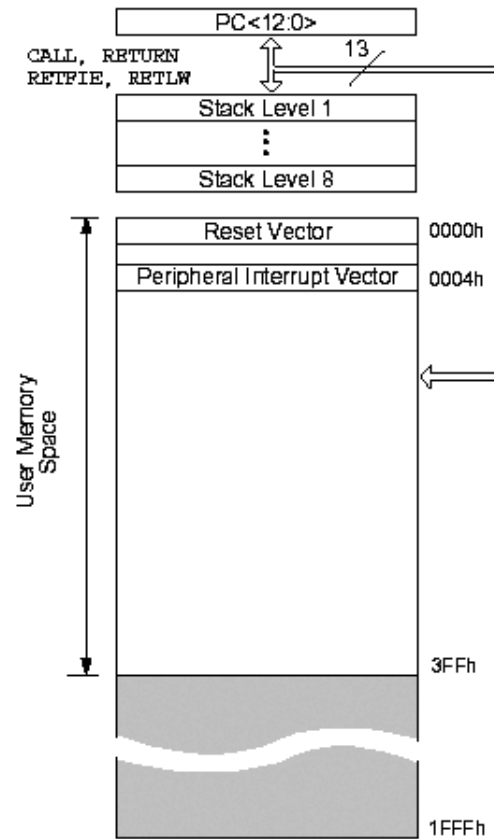


Fig.1.2 División de la memoria RAM

1.2 Estudio General del PIC 16F84A

El pic16F84 es un microcontrolador de dimensiones muy reducidas, consta solamente de 18 pines y además se puede programar fácilmente todas las veces que se necesite con un sencillo circuito conectado al puerto serial, paralelo o USB del PC.

El programa se escribe en el bloc del programa Micro-Code-Studio con extensión .ASM, después hay que ensamblarlo para obtener el fichero con extensión .HEX, que es el que hay que transferir al PIC. El programa que esta adjuntado con el Micro-Code-Studio es el Ic-Prog, también podemos adjuntarlo con el EduPlc, MPLAB y otros.

1.2.1 Descripción de sus elementos y pines

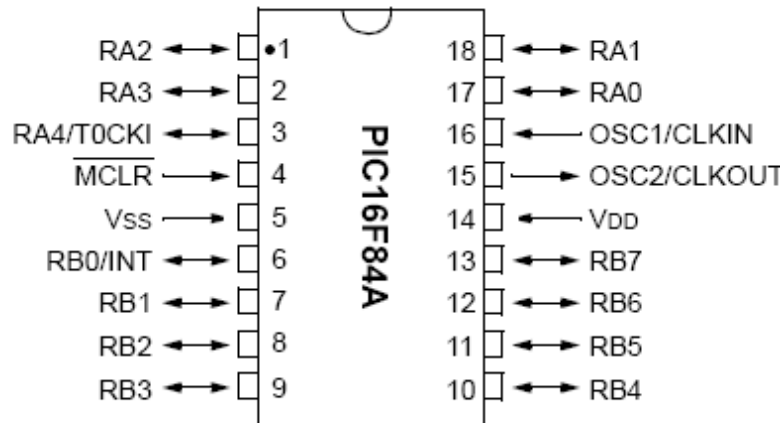


Fig.1.3 Descripción de los pines del Pic16f84A

Se trata de un microcontrolador de 8 bits. Es un PIC de gama baja, cuyas características podemos resumir en:

- Memoria de 1K x 14 de tipo Flash
- Memoria de datos EEPROM de 64 bytes
- -13 Líneas de E/S con control individual
- Frecuencia de funcionamiento máxima de 10 Mhz.
- Cuatro fuentes de interrupción
- Activación del puerto RBO/INT
- Desbordamiento del TMRO
- Cambio de estado en el puerto RB4-RB7
- Fin de la escritura de la EEPROM de datos
- Temporizador/contador TMRO programable de 8 bits
- Perro Guardián o WatchDog

Generalmente se encuentra encapsulado en formato DIP18.

A continuación veremos una breve descripción de cada uno de los pines:

- ◆ VDD: alimentación
- ◆ VSS: masa.
- ◆ OSC1/CLKIN-OSC2/CLKOUT: conexión del oscilador
- ◆ VPP/MCLR: tensión de programación y reset
- ◆ RAO-RA3: líneas de E/S del puerto A
- ◆ RA4; Línea de E/S del puerto A4 o entrada de impulsos de reloj para TMRO
- ◆ RBO/INT: línea de E/S del puerto B o petición de interrupción
- ◆ RB1-RB7; líneas de E/S del puerto B

Organización de la memoria

A continuación podemos ver la organización de la memoria del Pic16F84A:

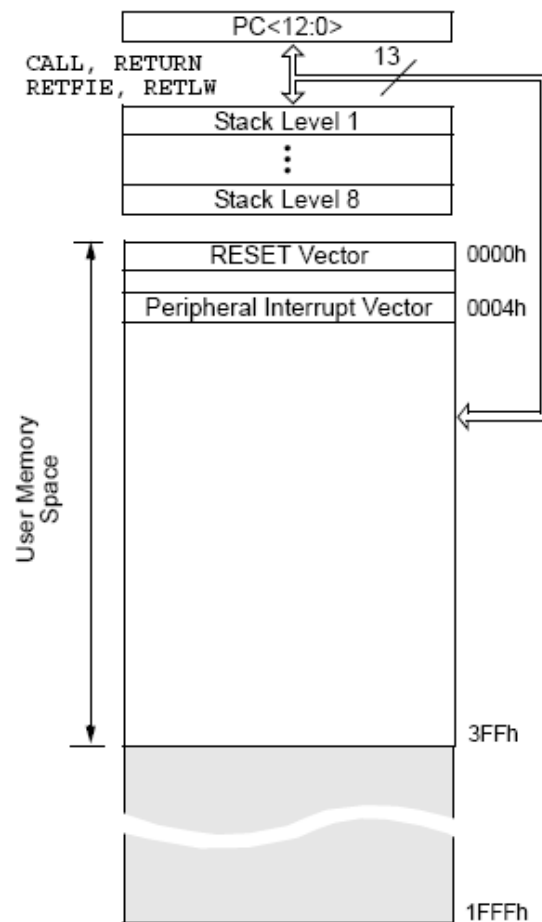


Fig.1.4. Organización de la memoria del Pic16f84A

Podemos comprobar en la Fig.1.4 como la memoria esta dividida en dos bancos (cada una de las columnas): banco 0 y banco 1. Las primeras 12 posiciones de cada banco (00h-0Bh y 80h-8Bh) están ocupadas por los Registros de Propósito Especial (Special Purpose Registers). Estos registros, en los que entraremos en detalle mas adelante, son los encargados de controlar ciertas funciones específicas del PIC.

Las 68 posiciones siguientes (0Ch-4Fh y 8Ch-CFh) son los denominados Registros de Propósito General, del inglés (General Purpose Registers). Estos son empleados para guardar cualquier dato que necesitemos durante la ejecución del programa.

BANCO 0

- TMRO: es un temporizador/contador de 8 bits. Puede operar de dos modos distintos:
 - Temporizador: el registro incrementa su valor en cada ciclo de instrucción ($F_{osc}/4$).

- Contador: el registro incrementa su valor con cada impulse introducido en el pin RA4/TOSKI. En ambos casos, cuando el registro se desborda, es decir, llega a su valor máximo (en este caso $2^8 = 256$. Como el 0 también se cuenta, el máximo valor sería 255), empieza de nuevo a contar a partir del 0. No sin antes informar de este evento a través de la activación de un flag y/o una interrupción.
- PCL: es el contador del programa. Indica la dirección de memoria que se leerá a continuación. En algunas ocasiones, como el empleo de las tablas, el uso de este registro es imprescindible.
- STATUS: registro de 8 bits que sirve para configurar ciertos aspectos del PIC. En la tabla 1.1 se aprecia la disposición de los bits de dicho registro:

Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
IRP		RPO	TO#	PD#	Z	DC	C

Tabla.1.1. Registro de 8 bits

- RPO: indica el banco de memoria con el que se está trabajando. Cuando vale 0 se selecciona el Banco 0 y cuando vale 1 el Banco 1.
- PORTA: representación de la puerta A. Cada bit representa una línea de E/S de la puerta A:

Bit	Bit6	Bit5	Bit 4	Bit3	Bit2	Bit1	Bit0
7							
			RA4	RA3	RA2	RA1	RA0

Tabla.1.2. Bits de E/S para la puerta A

Como se puede comprobar, los tres bits de mayor peso no representan ninguna línea de E/S, ya que la puerta A solo tiene 5 Líneas de E/S

- PORTB: lo mismo que la puerta A, pero en este caso con 8 líneas de E/S BANCO 1
- TRISA: registro de 8 bits de configuración de la puerta A. Si un bit encuentra en 1, esa línea de E/S se configura como entrada: si, en cambio, se encuentra a 0, se configura como salida
- TRISB: lo mismo que TRISA, pero referente a la puerta B.

Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
RB7	RB6	RB5	RB4	RB3	RB2	RB1	RB0

Tabla.1.3. Registro de 8 bits de configuración de la puerta B

Finalmente cabe destacar el registro W, también conocido como registro de trabajo (del inglés work) o acumulador. Es de vital importancia ya que, entre otras, deberemos usarlo de registro puente para llevar a cabo ciertas operaciones.

1.2.2 Diagrama de Bloques:

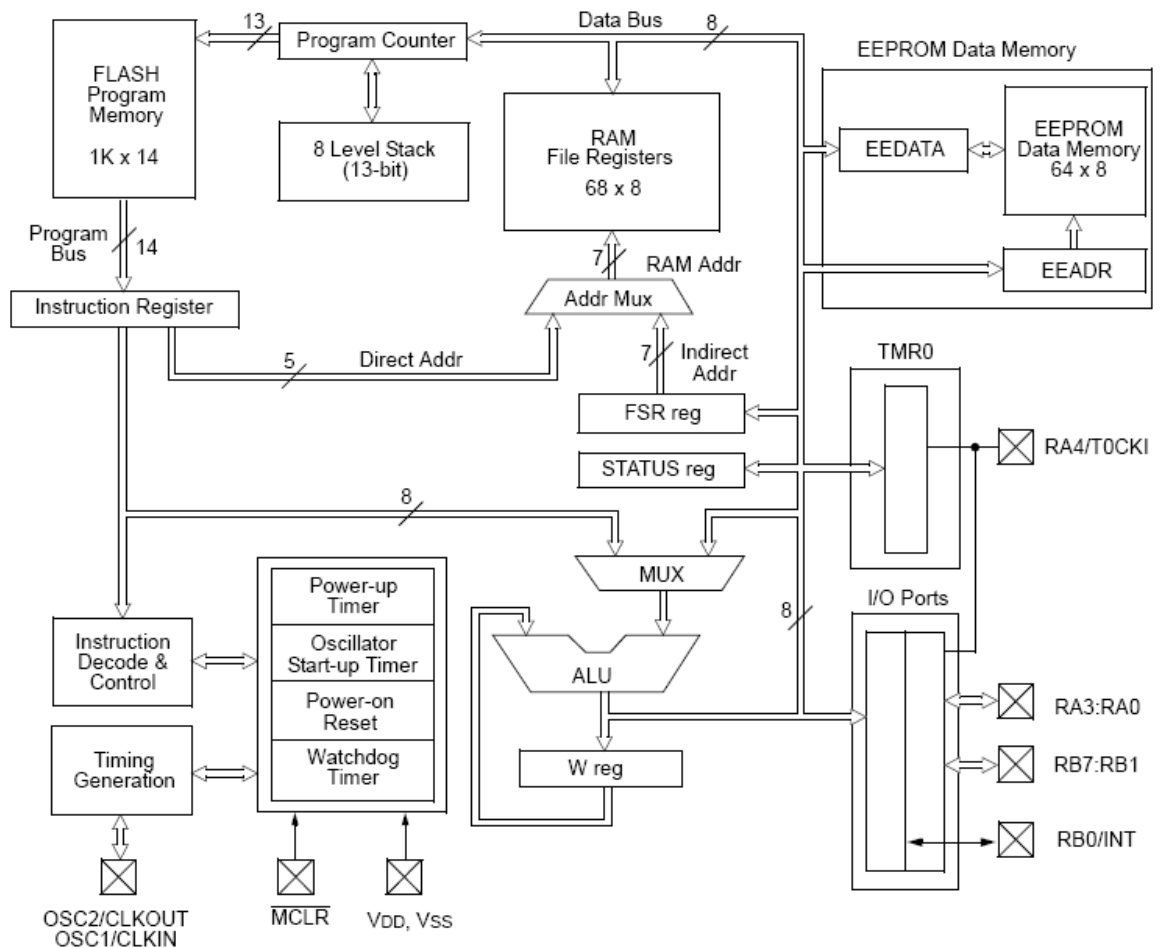


Fig.1.5 Diagrama de bloques del Pic16f84A

1.3 Estudio General del PIC 16F877A

El PIC 16f877A es un microcontralador de dimensiones muy amplias, consta de 40 pines, de igual manera se puede programar todas las veces que se desee, su programador va conectado al puerto serial, paralelo o USB.

El programa se escribe en el bloc del programa Micro-Code-Studio con extensión .ASM al mismo tenemos que ensamblarlo para obtener la extensión .HEX que es el lenguaje de máquina.

CARACTERÍSTICAS	16F877
Frecuencia máxima	DX-20MHz
Memoria de programa flash palabra de 14 bits	8KB
Posiciones RAM de datos	368
Posiciones EEPROM de datos	256
Puertos E/S	A,B,C,D,E
Número de pines	40
Interrupciones	14
Timers	3
Módulos CCP	2
Comunicaciones Serie	MSSP, USART
Comunicaciones paralelo	PSP
Líneas de entrada de CAD de 10 bits	8
Juego de instrucciones	35 Instrucciones
Longitud de la instrucción	14 bits
Arquitectura	Harvard
CPU	Risc
Canales Pwm	2
Pila Hardware	-
Ejecución En 1 Ciclo Máquina	-

Tabla.1.4. Características comunes del PIC16F877A

1.3.1 Descripción de sus elementos y pines

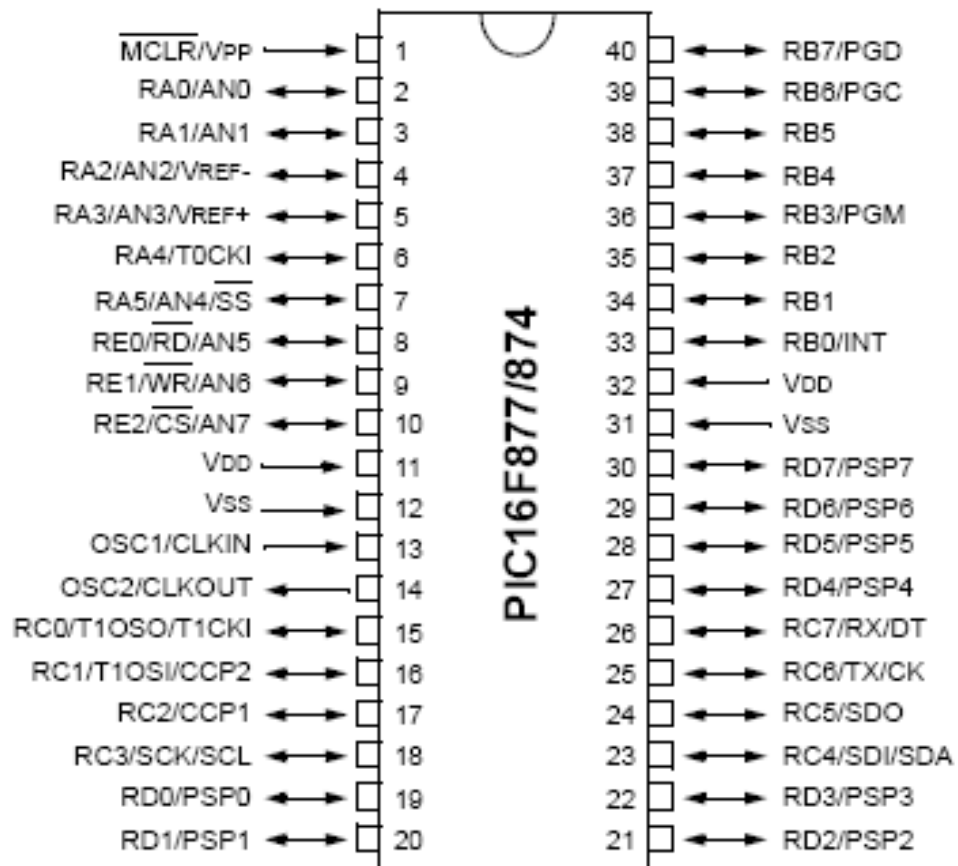


Fig.1.6. Descripción de los pines del Pic 16f877A

Descripción de los puertos:

Puerto A:

- Puerto de e/s de 6 pines
- RA0 è RA0 y AN0
- RA1 è RA1 y AN1
- RA2 è RA2, AN2 y Vref-
- RA3 è RA3, AN3 y Vref+
- RA4 è RA4 (Salida en colector abierto) y T0CKI(Entrada de reloj del modulo Timer0)
- RA5 è RA5, AN4 y SS (Selección esclavo para el puerto serie síncrono)

Puerto B:

- Puerto e/s 8 pines
- Resistencias pull-up programables
- RB0 è Interrupción externa

- RB4-7 è Interrupción por cambio de flanco
- RB5-RB7 y RB3 è programación y debugger in circuit

Puerto C:

- Puerto e/s de 8 pines
- RC0 è RC0, T1OSO (Timer1 salida oscilador) y T1CKI (Entrada de reloj del modulo Timer1).
- RC1-RC2 è PWM/COMP/CAPT
- RC1 è T1OSI (entrada osc timer1)
- RC3-4 è IIC
- RC3-5 è SPI
- RC6-7 è USART

Puerto D:

- Puerto e/s de 8 pines
- Bus de datos en PPS (Puerto paralelo esclavo)

Puerto E:

- Puerto de e/s de 3 pines
- RE0 è RE0 y AN5 y Read de PPS
- RE1 è RE1 y AN6 y Write de PPS
- RE2 è RE2 y AN7 y CS de PPS

Dispositivos periféricos:

- Timer0: Temporizador-contador de 8 bits con preescaler de 8 bits
- Timer1: Temporizador-contador de 16 bits con preescaler que puede incrementarse en modo sleep de forma externa por un cristal/clock.
- Timer2: Temporizador-contador de 8 bits con preescaler y postescaler.
- Dos módulos de Captura, Comparación, PWM (Modulación de Anchura de Impulsos).
- Conversor A/D de 10 bits.
- Puerto Serie Síncrono Master (MSSP) con SPI e I²C (Master/Slave).
- USART/SCI (Universal Synchronous Asynchronous Receiver Transmitter) con 9 bit.
- Puerta Paralela Esclava (PSP) solo en encapsulados con 40 pines

NOMBRE DEL PIN	PIN	TIPO	TIPO DE BUFFER	DESCRIPCIÓN
OSC1/CLKIN	13	I	ST/MOS	Entrada del oscilador de crystal / Entrada de señal de reloj externa
OSC2/CLKOUT	14	O	-	Salida del oscilador de crystal
MCLR/V _{pp} /THV	1	I/P	ST	Entrada del Master clear (Reset) o entrada de voltaje de programación o modo de control high voltage test
RA0/AN0 RA1/AN1 RA2/AN2/ Vref- RA3/AN3/Vref+ RA4/T0CKI RA5/SS/AN4	2 3 4 5 6 7	I/O I/O I/O I/O I/O I/O	TTL TTL TTL TTL ST TTL	PORTA es un puerto I/O bidireccional RA0: puede ser salida analógica 0 RA1: puede ser salida analógica 1 RA2: puede ser salida analógica 2 o referencia negativa de voltaje RA3: puede ser salida analógica 3 o referencia positiva de voltaje RA4: puede ser entrada de reloj el timer0. RA5: puede ser salida analógica 4 o el esclavo seleccionado por el puerto serial síncrono.
RBO/INT RB1 RB2 RB3/PGM RB4 RB5 RB6/PGC RB7/PGD	33 34 35 36 37 38 39 40	I/O I/O I/O I/O I/O I/O I/O I/O	TTL/ST TTL TTL TTL TTL TTL/ST TTL/ST	PORTB es un puerto I/O bidireccional. Puede ser programado todo como entradas RB0 puede ser pin de interrupción externo. RB3: puede ser la entrada de programación de bajo voltaje Pin de interrupción Pin de interrupción Pin de interrupción. Reloj de programación serial
RCO/T1OSO/T1CKI RC1/T1OS1/CCP2 RC2/CCP1 RC3/SCK/SCL RC4/SD1/SDA RC5/SD0 RC6/Tx/CK RC7/RX/DT	15 16 17 18 23 24 25 26	I/O I/O I/O I/O I/O I/O I/O I/O	ST ST ST ST ST ST ST ST	PORTC es un puerto I/O bidireccional RCO puede ser la salida del oscilador timer1 o la entrada de reloj del timer1 RC1 puede ser la entrada del oscilador timer1 o salida PMW 2 RC2 puede ser una entrada de captura y comparación o salida PWN RC3 puede ser la entrada o salida serial de reloj síncrono para modos SPI e I2C RC4 puede ser la entrada de datos SPI y modo I2C RC5 puede ser la salida de datos SPI RC6 puede ser el transmisor asíncrono USART o el reloj síncrono. RC7 puede ser el receptor asíncrono USART o datos síncronos

RD0/PSP0 RD1/PSP1 RD2/PSP2 RD3/PSP3 RD4/PSP4 RD5/PSP5 RD6/PSP6 RD7/PSP7	19 20 21 22 27 28 29 30	I/O I/O I/O I/O I/O	I/O I/O I/O I/O	ST/TTL ST/TTL ST/TTL ST/TTL ST/TTL ST/TTL ST/TTL ST/TTL	PORTD es un puerto bidireccional paralelo
REO/RD/AN5 RE1/WR/AN RE2/CS/AN7	8 9 10	I/O I/O I/O	ST/TTL ST/TTL ST/TTL	PORTE es un puerto I/O bidireccional REO: puede ser control de lectura para el puerto esclavo paralelo o entrada analógica 5 RE1: puede ser escritura de control para el puerto paralelo esclavo o entrada analógica 6 RE2: puede ser el selector de control para el puerto paralelo esclavo o la entrada analógica 7.	
Vss	12.31	P	-	Referencia de tierra para los pines lógicos y de I/O	
Vdd	11.32	P	-	Fuente positiva para los pines lógicos y de I/O	
NC	-	-	-	No está conectado internamente	

Tabla.1.5 Descripción de pines del PIC 16F877A

Organización de la Memoria:

A continuación podemos ver la organización de la memoria del 16f877A

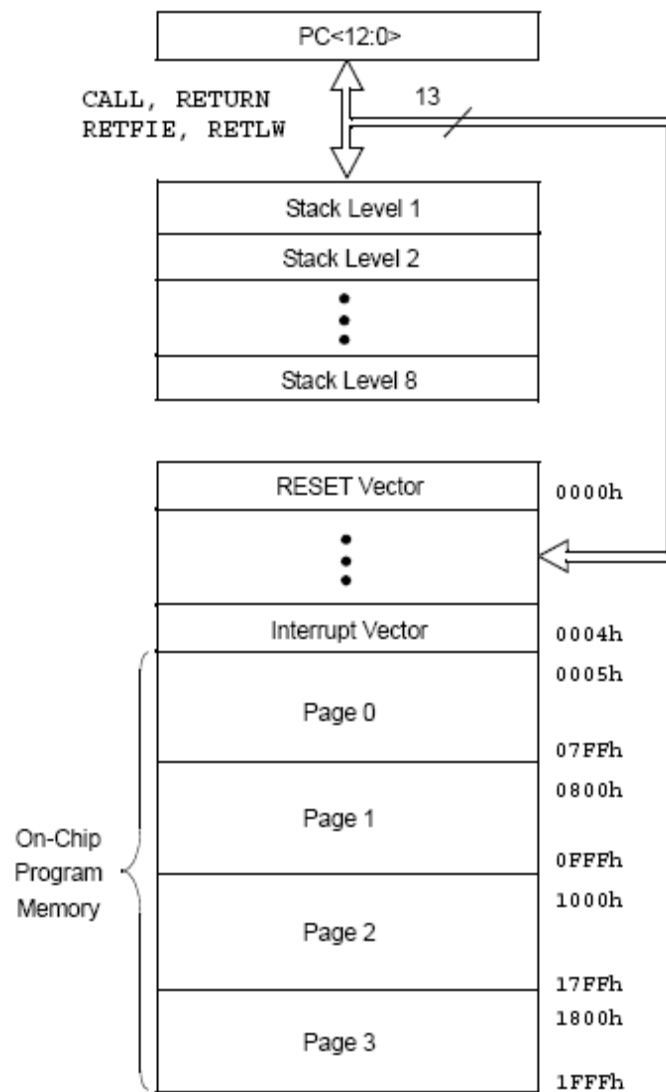


Fig.1.7 Organización de la memoria del PIC 16f877A

1.3.2 Diagrama de Bloques:

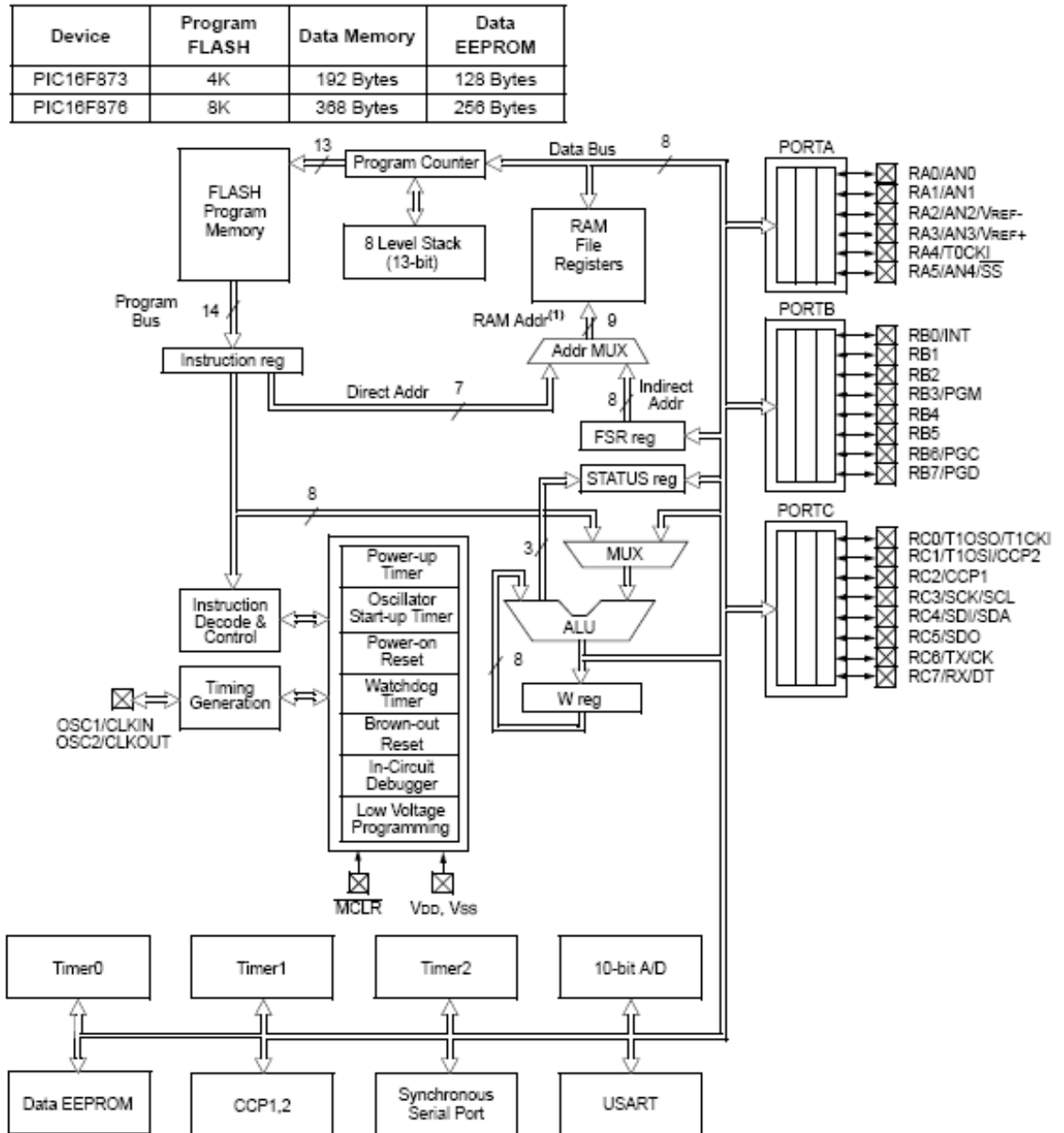


Fig.1.8 Diagrama de bloques del PIC 16f877A

CAPÍTULO II

2. SOFTWARE DE PROGRAMACIÓN DE LOS MICROCONTROLADORES

2.1 Instalación y manejo del Software:

La instalación del programa se lo hará en el disco duro siguiendo los pasos que se nos indica al momento de instalarlo lo que vamos a hacer es agrupar los programas dentro de la carpeta mecanique los programas pbp2.44 y el icprog105c, como se muestra en la figura 2.1

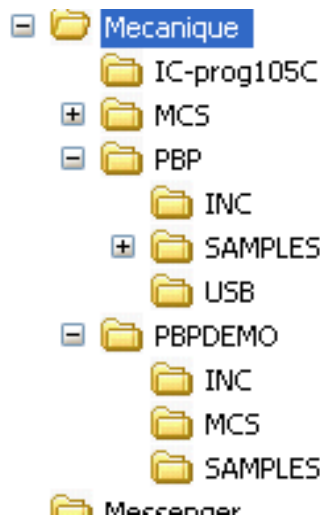


Fig.2.1 Instalación del software

Luego creamos un acceso directo para los dos programas el micro-code – studio y el ic-prog.



Fig.2.2 Acceso directo a micro-code-studio y al ic-prog

Ya ingresado al programa nos presenta la pantalla de la siguiente manera:

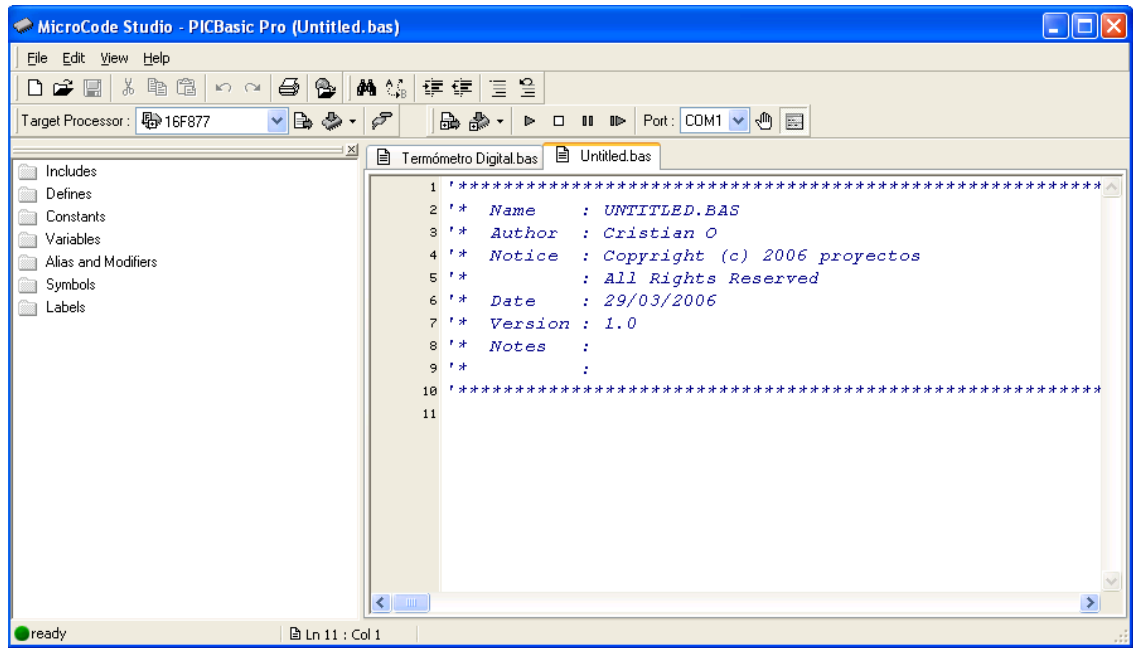


Fig.2.3 Menú principal del micro-code-studio

una vez que ingresemos al programa y buscamos **View** y damos un clic y nos ubicamos en **Editor Options**, marcamos en **Show line numbers in left gutter** esto nos servirá para que aparezca el número de línea que estamos programando esto es útil al momento de encontrar errores.

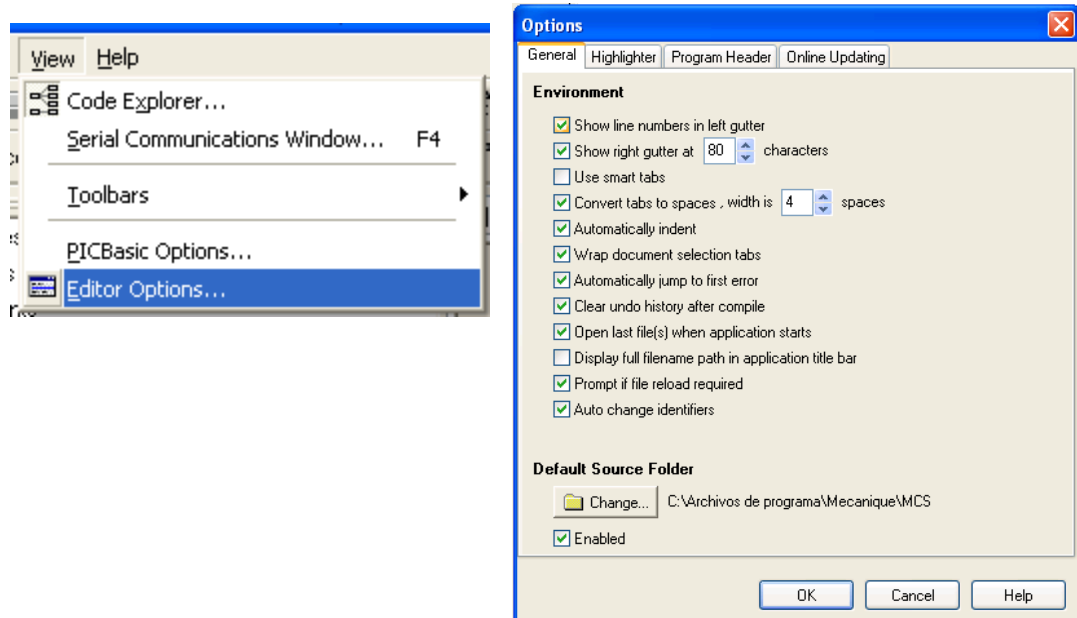


Fig.2.4 Menú de opciones

Nuevamente nos ubicamos en **View** y damos clic en **PicBasicOptions** pestañamos en **programmer** buscamos el icprog caso contrario nos ubicamos en **Add New Programmer**.

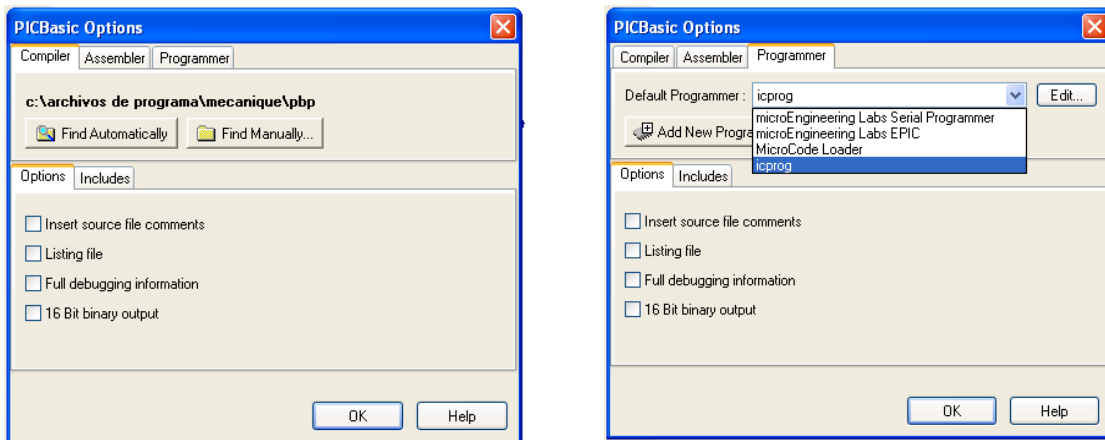


Fig.2.5. Adaptación del Ic-prog

Inmediatamente aparece una pantalla más pequeña nos ubicamos en **create a custom programmer entry**, luego presionamos **next** en la siguiente pantalla escribimos el nombre icprog luego **next**, y nos pedirá el nombre del archivo ejecutable que es **icprog.exe**, presionamos **next** y luego **Find automatically** en esta pantalla nos pide parámetros pero para el icprog no hace falta así que solo presionamos en **Finished** luego sale la pantalla de **PicBasicOptions** presionamos **Ok**. Y ya está configurado. Si disponemos de cualquier de estos sistemas operativos es necesario que se instale un driver para que funcione correctamente el programador **lc-prog105C** primero encontramos el archivo **lc-prog_driver.zip** y procedemos a descomprimirlo de igual manera como se lo hace y dentro de esta carpeta tendremos el archivo **icprog.sys**, el cual debemos moverlo dentro de la carpeta **lcprog105C**, junto al archivo ejecutable **icprog.exe**,

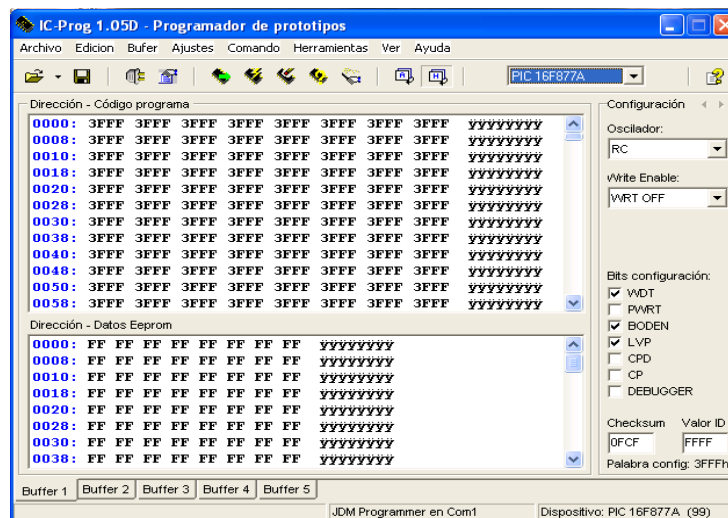


Fig.2.6 Menú de programación del ic-prog

La programación se la realiza por el puerto serial no aseguramos que el programador se encuentre en **JDM Programmer**, una vez realizado ese cambio nos colocamos en **Settings**, luego damos clic en **Options**, cambiamos el lenguaje y ponemos Ok, más tarde hacemos clic en **Miscelánea** para habilitar el driver de Windows **NT/2000/XP**, vez marcado el casillero damos clic en Ok.

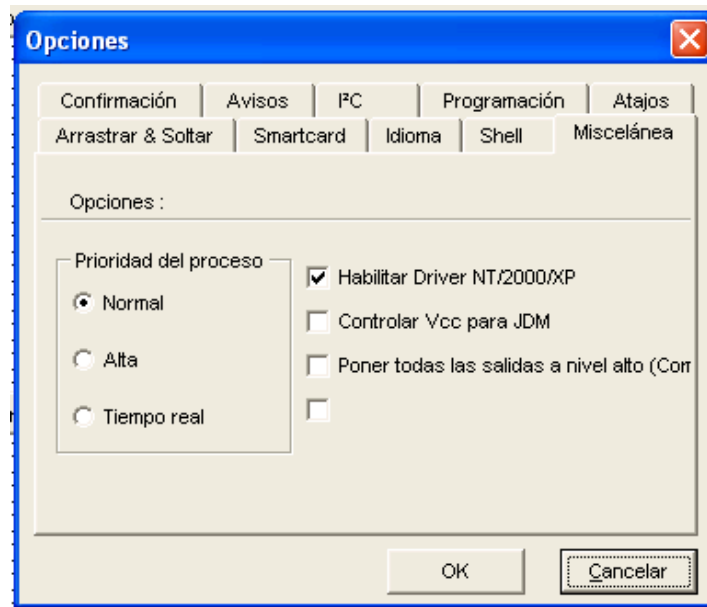


Fig.2.7 Habilitador del driver

2.2 Manejo del Micro-Code-Studio.

El microcodestudio es un programa editor de texto esta hecho para facilitar la programación de los microcontroladores PIC, los mismos que consta de las siguientes partes:

1. Identificador de modelo de pic:



Fig.2.8 Modelos de PIC

2. Buscador de códigos:

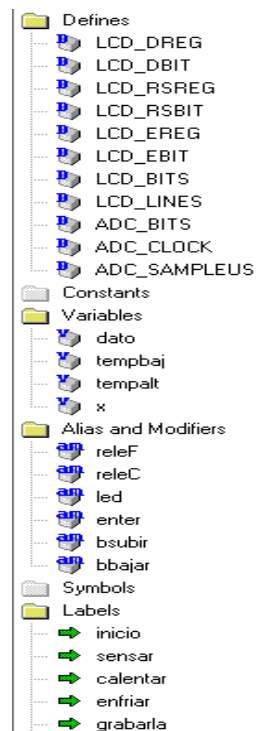


Fig.2.9 Buscador de códigos

3. Número de línea del programa:

```

11 PAUSE 200
12 DEFINE LCD_DREG PORTB
13 DEFINE LCD_DBIT 4
14 DEFINE LCD_RSREG PORTB
15 DEFINE LCD_RSBIT 3
16 DEFINE LCD_EREG PORTB
17 DEFINE LCD_EBIT 2
18 DEFINE LCD_BITS 4
19 DEFINE LCD_LINES 2
20

```

Fig.2.10 Numerador del programa

4. Espacio que ocupa el Pic:

● success : 793 words used.

Fig.2.11 Espacio utilizado en el programa

5. Programa del microcontrolador:

```

DEFINE ADC_BITS 8
DEFINE ADC_CLOCK 3
DEFINE ADC_SAMPLEUS 50
        ;ADC_SAMPLEUS es
        ;entre fi;

TRISA= %1
ADCON1= %00001110

dato VAR BYTE
tempbaj VAR BYTE
tempalt VAR BYTE

```

Fig.2.12 Programa del microcontrolador

6. Comentarios:

```

;fije número de bits del resultado (5,8,10)
;fije El CLOcK(re= 3)
;fije el tiempo de muestro en Us
el número de microsegundos que el programa espera
r el canal y comenzar lña conversión analógica/dig
;el puerto A es de entrada
;el puerto A.0 es conversor los demas digitales

;crear variable dato para guardar

```

7. Encabezado del programa:

```

Name      : UNTITLED.BAS
Author    : Cristian O
Notice    : Copyright (c) 2006 proyectos
          : All Rights Reserved
Date      : 19/02/2006
Version   : 1.0
Notes     :
          :

```

8. Copilador:



Fig.2.13 Compilador del programa

2.3 Declaraciones disponibles en el programa PBP2.44

DECLARACIÓN	APLICACIÓN
@	Inserta una línea de código ensamblador
ADCIN	Lee el conversor analógico
ASM...ENDASM	Insertar una sección de código ensamblador
BRANCH	GOTO computado
BRANCHL	BRANCH fuera de página
BUTTON	Anti-rebote y auto repetición
CALL	Llamada a subrutina de ensamblador
CLEAR	Hace cero todas las variables
CLEARWDT	Hace cero el contador del Watchdog timer
COUNT	Cuenta el número de pulsos de un pin
DATA	Define el contenido inicial en un chip EEPROM
DEBUG	Señal asincrónica de salida en un pin fijo y baud
DEBUGIN	Señal asincrónica de entrada en un pin fijo y baud
DISABLE	Deshabilita el procesamiento de ON INTERRUPT
DISABLE DEBUG	Deshabilita el procesamiento de ON DEBUG
DISABLE INTERRUPT	Deshabilita el procesamiento ON INTERRUPT
DTMFOUT	Produce tonos telefónicos en un pin
EEPROM	Define el contenido inicial en un chip EEPROM
ENABLE	Habilita el procesamiento de ON INTERRUPT
ENABLE DEBUG	Habilita el procesamiento de ON DEBUG
ENABLE INTERRUPT	Habilita el procesamiento de ON INTERRUPT
END	Detiene la ejecución e ingresa en modo de baja potencia
FOR...NEXT	Ejecuta declaraciones en forma repetitiva
FREQOUT	Produce hasta 2 frecuencias en un pin
GOSUB	Llama a una subrutina BASIC en la línea
GOTO	Continúa la ejecución en la línea
HIGH	Saca un 1 lógico (5V) por pin
HPWM	Salida de hardware con ancho de pulsos modul.
HSERIN	Entrada serial asincrónica (hardware)
HSEROUT	Salida serial asincrónica (hardware)
I2CREAD	Lee bytes de dispositivos I2C
I2CWRITE	Graba bytes de dispositivos I2C
IF..THEN..ELSE..END	Ejecuta declaraciones en forma condicional
INPUT	Convierte un pin en entrada
LCDIN	Lee caracteres desde una RAM de un LCD
LCDOUT	Muestra caracteres en un LCD
LET	Asigna el resultado de una expresión a un var.
LOOKDOWN	Busca un valor en una tabla de constantes
LOOKDOWN2	Busca un valor en una tabla de constantes
LOOKUP	Obtiene un valor constante de una tabla
LOOKUP2	Obtiene un valor constante o variable
LOW	Hace 0 lógico (0V) en un pin específico
NAP	Apagar el procesador por un corto periodo
ON DEBUG	Ejecuta un debug en BASIC

ON INTERRUPT	Ejecuta una subrutina BASIC en un interruptor
OUTPUT	Convierte un pin en salida
OWIN	Entrada de dispositivos un-alambre
OWOUT	Salida de dispositivos un-alambre
PAUSE	Demora con resolución de 1 milisegundo
PAUSEUS	Demora con resolución de 1 microsegundo
PEEK	Lee un byte del registro
POKE	Graba un byte en el registro
POT	Lee el potenciómetro en el pin especificado
PULSIN	Mide el ancho de pulso en un pin
PULSOUT	Genera pulso hacia un pin
PWM	Salida modulada en ancho de pulso por un pin
RANDOM	Genera número pseudos-aleatoria
RCTIME	Mide el ancho de pulso en un pin
READ	Lee byte de un chip EEPROM
READCODE	Lee palabra desde un código de memoria
RESUME	Continúa la ejecución después de interrupción
RETURN	Continúa en la declaración que sigue al GOSUB
REVERSE	Convierte un pin de salida en entrada
SELECT CASE	Compara una variable con diferentes valores
SERIN	Entrada serial asincrónica (tipo BASIC Stamp1)
SERIN2	Entrada serial asincrónica (tipo BASIC Stamp2)
SEROUT	Salida serial asincrónica
SEROUT2	Salida serial asincrónica
SHIFTIN	Entrada serial sincrónica
SHIFTOUT	Salida serial sincrónica
SLEEP	Apagar el procesador por un periodo de tiempo
SOUND	Genera un tono o ruido blanco en un pin
STOP	Detiene la ejecución del programa
SWAP	Intercambia los valores de dos variables
TOGGLF	Hace salida a un pin y cambia su estado
USBIN	Entrada ESB
USBINIT	Inicializar USB
USBOUT	Salida USB
WHILE...WEND	Ejecuta declaraciones mientras sea cierta
WRITE	Graba bytes en un chip EEPROM
WRITECODE	Escribe palabra en código de memoria
XIN	Entrada X -10
XOUT	Salida X-10
TRIS	Declara salidas todo el puerto elegido

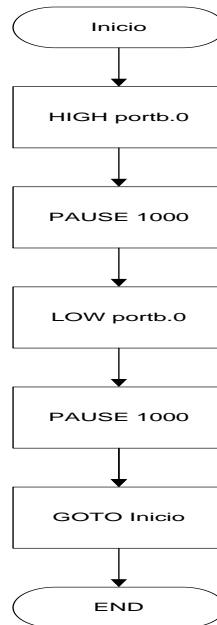
Tabla.2.1 Declaraciones disponibles en el PBP2.4

CAPITULO III

3. PROGRAMACIÓN DE LAS PRÁCTICAS

3.1 Manejo del Puerto b.

3.1.1 Diagrama de Flujo:

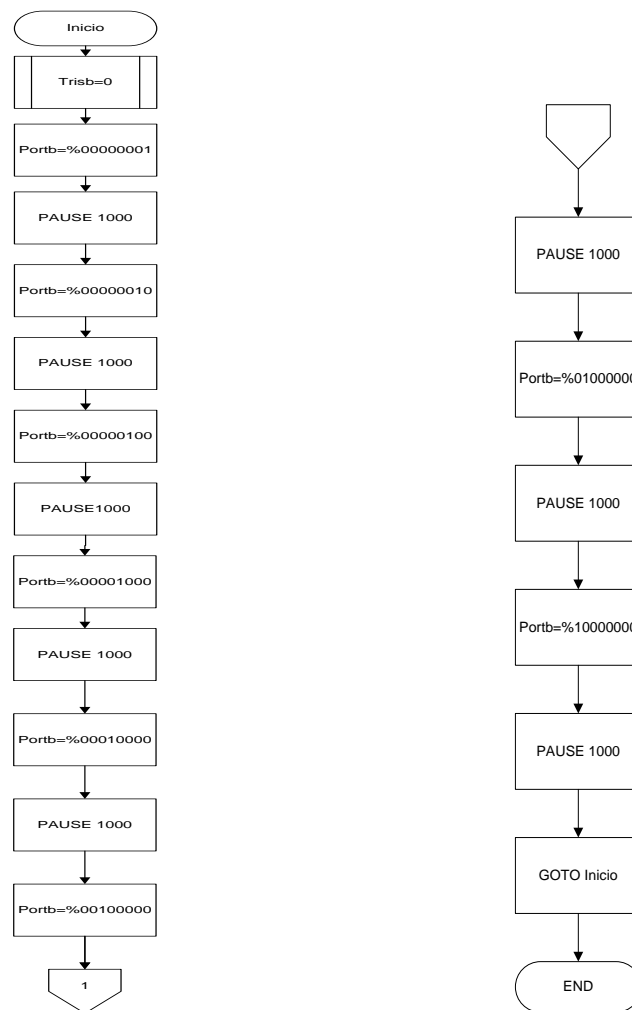


3.1.2 Programación en micro-code-studio:

Inicio:	<i>;etiqueta de inicio</i>
HIGH portb.0	<i>;sacar un alto en el puertob.0</i>
PAUSE 1000	<i>;hacer pausa 1 segundo</i>
LOW portb.0	<i>;sacar un bajo en el puertob.0</i>
PAUSE 1000	<i>;hacer pausa 1 segundo</i>
GOTO inicio	<i>;regresa a la etiqueta de inicio</i>
END	<i>;fin de la rutina</i>

3.2 Juego de Luces:

3.2.1 Diagrama de Flujo:



3.2.2 Programación en el micro-code-studio:

```

trisb=0           ;hacer bajo todas las salidas del puerto b
inicio:           ;etiqueta de inicio
portb= %00000001 ;poner en alto el puertob.0
pause 1000        ;hacer pausa 1 segundo
portb= %00000010 ; poner en alto el puertob.1
pause 1000        ;hacer pausa 1 segundo
portb= %00000100 ; poner en alto el puertob.2
pause 1000        ;hacer pausa 1 segundo
portb= %00001000 ; poner en alto el puertob.3
pause 1000        ;hacer pausa 1 segundo
portb= %00010000 ; poner en alto el puertob.4
pause 1000        ;hacer pausa 1 segundo
portb= %00100000 ; poner en alto el puertob.5
pause 1000        ;hacer pausa 1 segundo
portb= %01000000 ; poner en alto el puertob.6
  
```

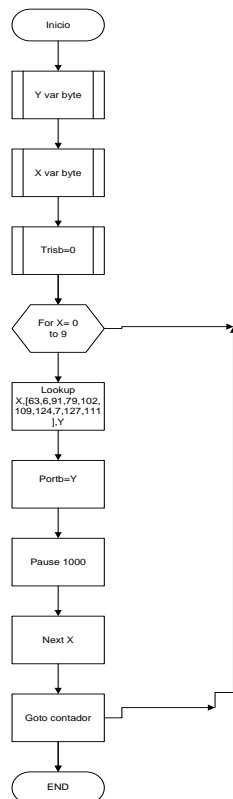
```

pause 1000           ;hacer pausa 1 segundo
portb= %10000000    ; poner en alto el puertob.7
pause 1000           ;hacer pausa 1 segundo
goto inicio          ;regresa a la etiqueta de inicio
end                  ;fin del programa

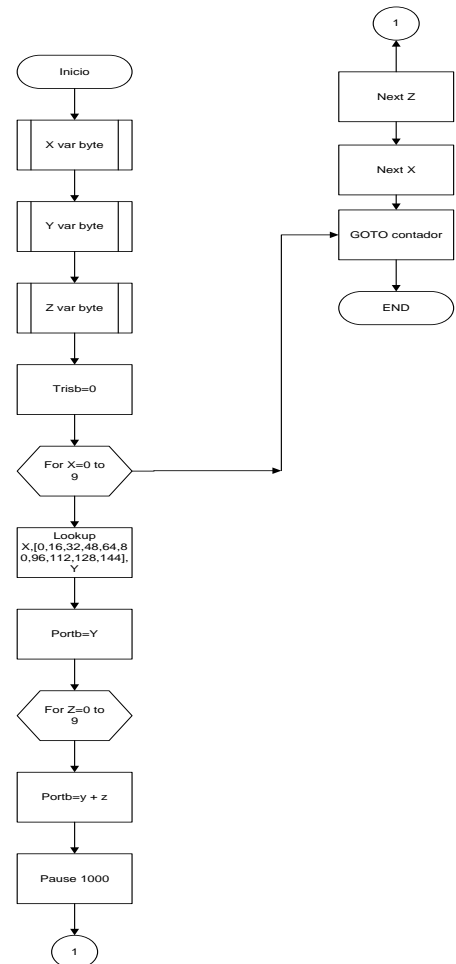
```

3.3 Contador con displays de 7 segmentos:

3.3.1 Diagrama de Flujo:



Contador de 0 a 9



Contador de 0 a 99

3.3.2 Programación en micro-code-studio:

3.3.2.1 contador de 0 a 9.

```

y var byte           ;crea la variable y de un tamaño de 255
x var byte           ;crea la variable x de un tamaño de 255
trisb= 0             ;poner en bajo las salidas del puerto b
contador:            ;etiqueta del programa
for x= 0 to 9         ;contador de 0 a 9
    lookup x,[63,6,91,79,102,109,124,7,127,111],y
    portb= y
    pause 1000        ;hacer pausa 1 segundo
next x
goto contador         ;regresa a la etiqueta contador

```

end

3.3.2.2 contador de 0 a 99.

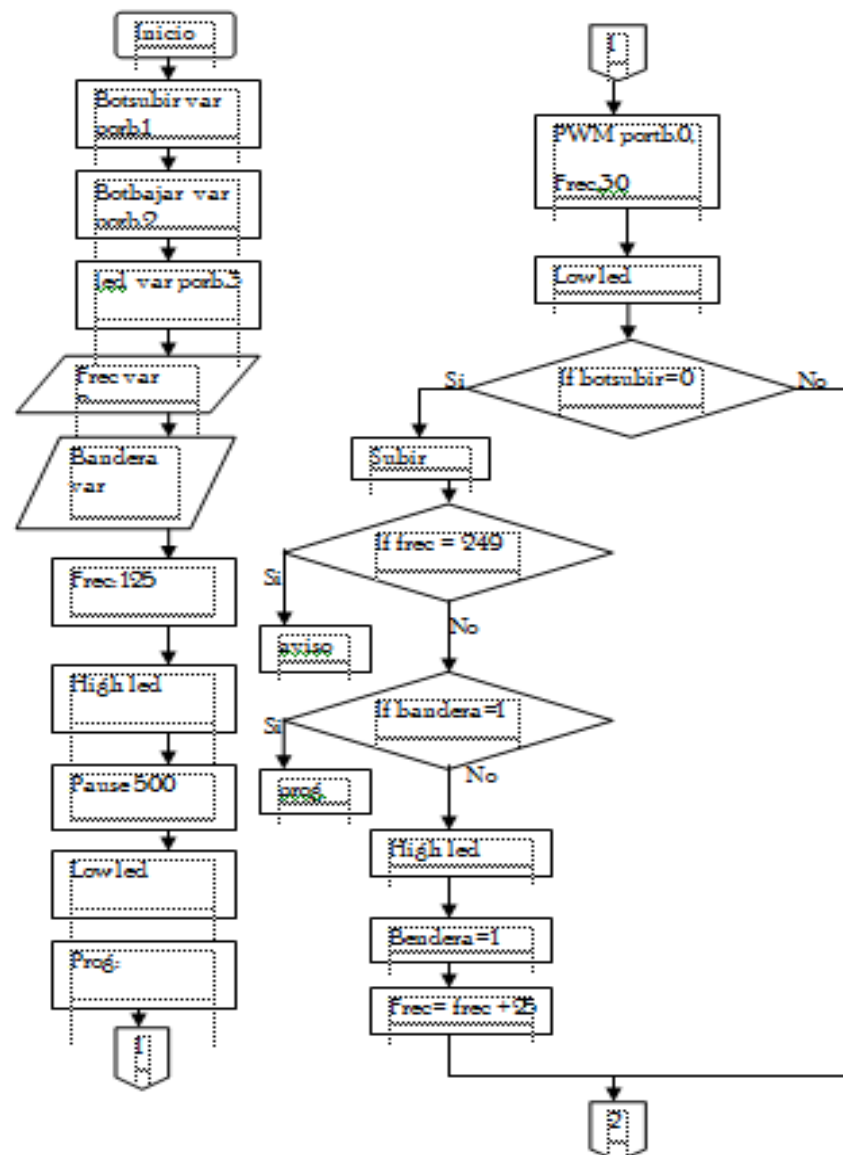
```

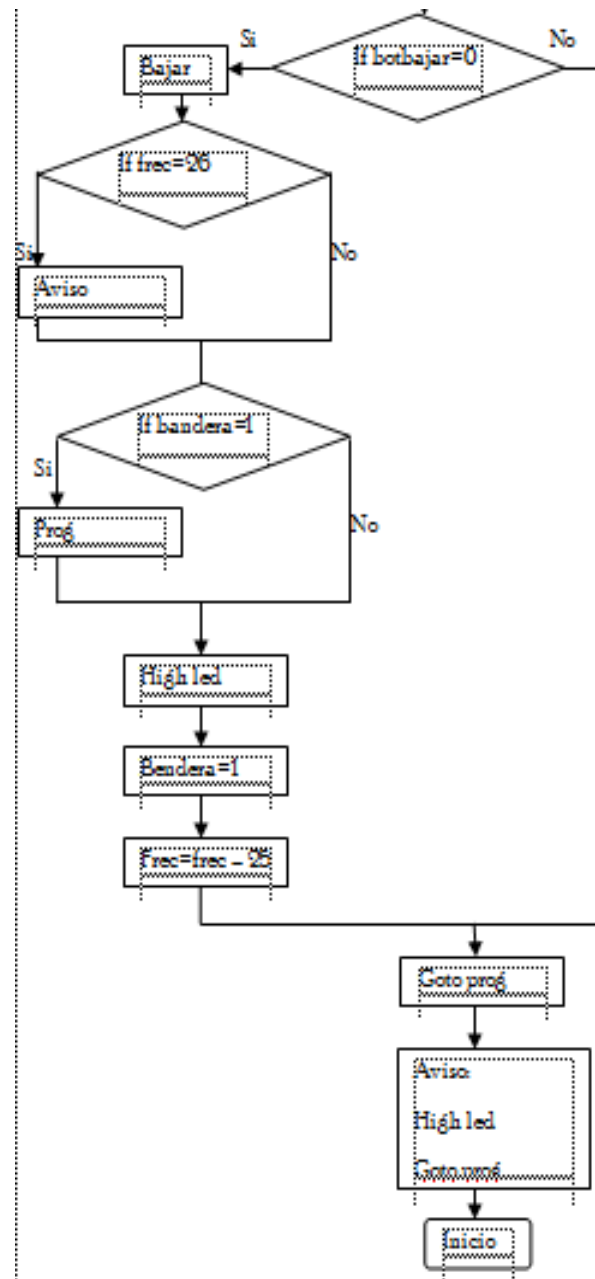
x var byte           ;crea la variable x de un tamaño de 255
y var byte           ;crea la variable y de un tamaño de 255
z var byte           ;crea la variable z de un tamaño de 255
trisb=0              ;poner en bajo todas las salidas del puerto b
contador:            ;etiqueta del programa
  for x=0 to 9 ;      contador de 0 a 9
    lookup x,[0,16,32,48,64,80,96,112,128,144],y      ;contador de decenas
    portb=y
  for z=0 to 9
    portb=y+z      ;contador de unidades
    pause 1000      ; hacer pausa en 1 segundo
  next z            ;regresa al contador z
next x              ;regresa al contador x
goto contador       ;regresa a la etiqueta contador
end

```

3.4 Modulación por Ancho de Pulsos:

3.4.1 Diagrama de flujo:





3.4.2 Programación en micro-code-studio:

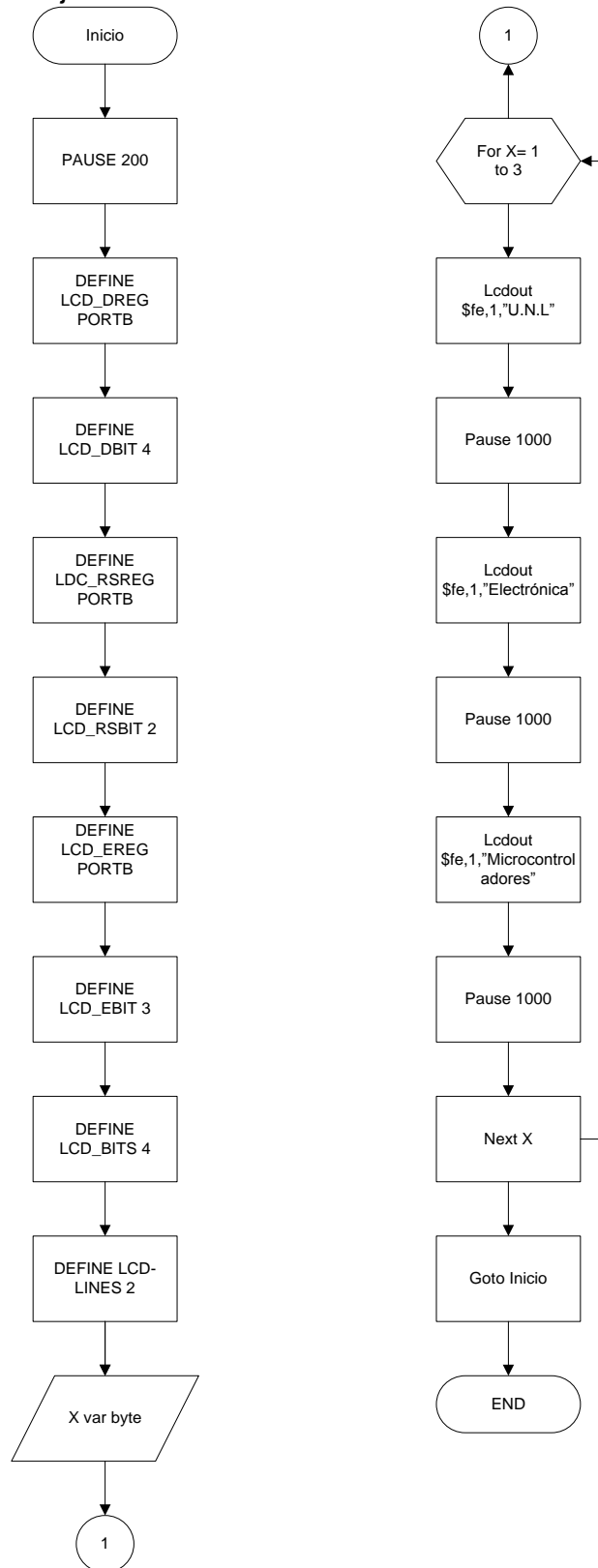
```

botsubir var portb.1      ;crear variable en el puerto b.1
botbajar var portb.2      ;crear variable en el puerto b.2
LED var portb.3           ;crear variable en el Puerto b.3
FREC var byte             ;crea la variable FREC de un tamaño de 255
bandera var bit           ;crea la variable bandera de un tamaño de 1bit
FREC= 125
high LED                  ;encender el led para saber que ya arranco
pause 500                 ;por medio segundo
low LED                   ; apagar el led
PROG:
pwm portb.0,FREC,30       ;sacar PWM 30 ciclos de 125 señal util
low led                   ;apagar el led
if botsubir=0 then subir  ;si se pulsa el boton s ir a subir
if botbajar=0 then bajar  ;si se pulsa el boton b bajar
bandera=0                 ;bandera cargado con 0
goto PROG
Subir:
if FREC > 249 then aviso   ;si supera 249 ir a aviso
if bandera= 1 then prog    ;si la bandera esta en 1 salir
high LED                   ;encender el led
bandera=1                  ;cargar la bandera como uno
FREC= FREC + 25            ;sumar 25 la variable FREC
goto PROG                  ;ir al programa
Bajar:
if Frec < 26 then aviso    ;si baja de 26 ir a aviso
if bandera= 1 then prog    ;si la bandera esta en uno salir
high LED                   ;encender el led
bandera=1                  ;cargar la bandera con uno
FREC= FREC-25              ;restar 25 a la variable FREC
goto prog                  ;ir a programa
aviso:
high LED                   ;encender el led
goto prog                  ;regresa a programa
end

```

3.5 Manejo de un LCD:

3.5.1 Diagrama de flujo:



3.5.2 Programación en micro-code-studio:

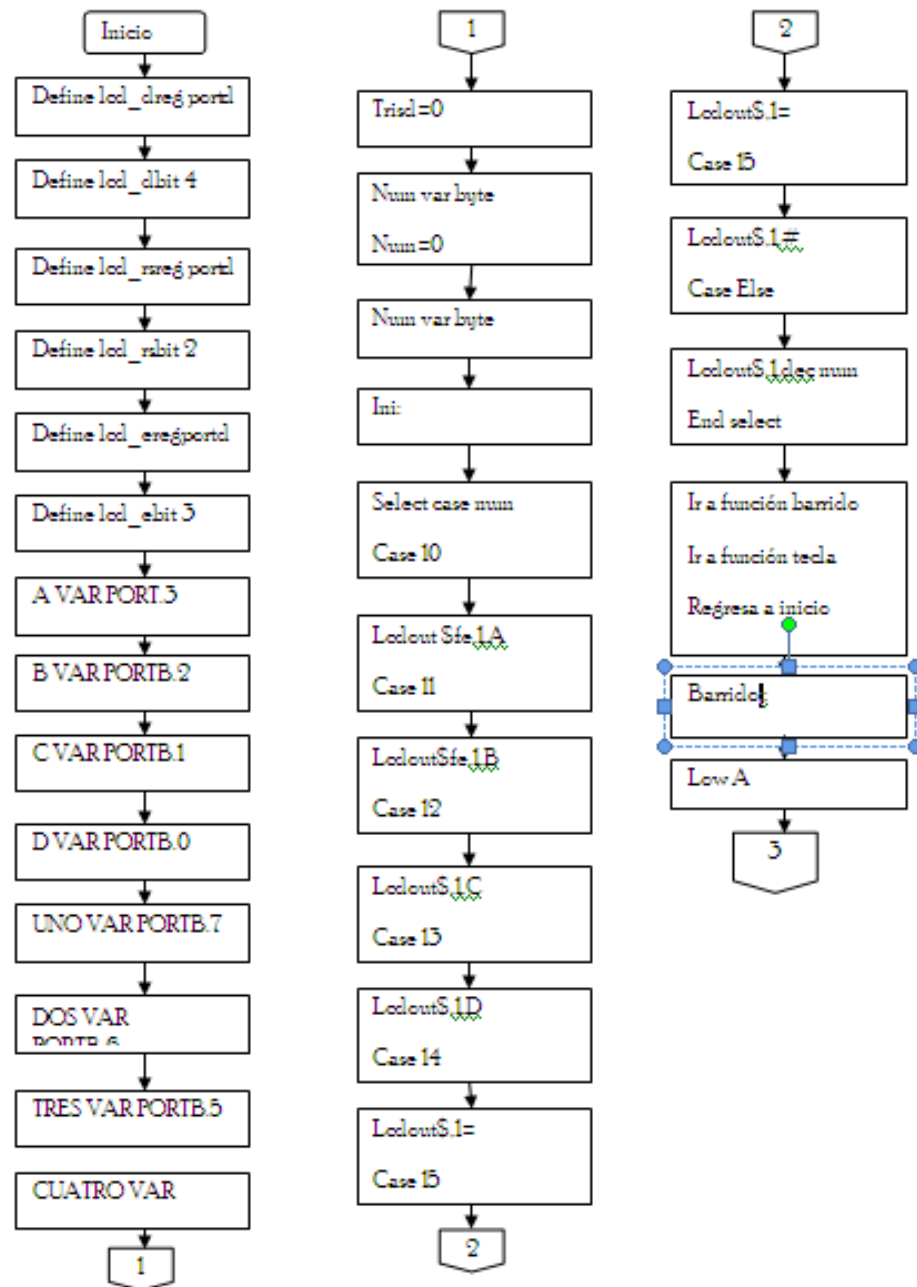
```

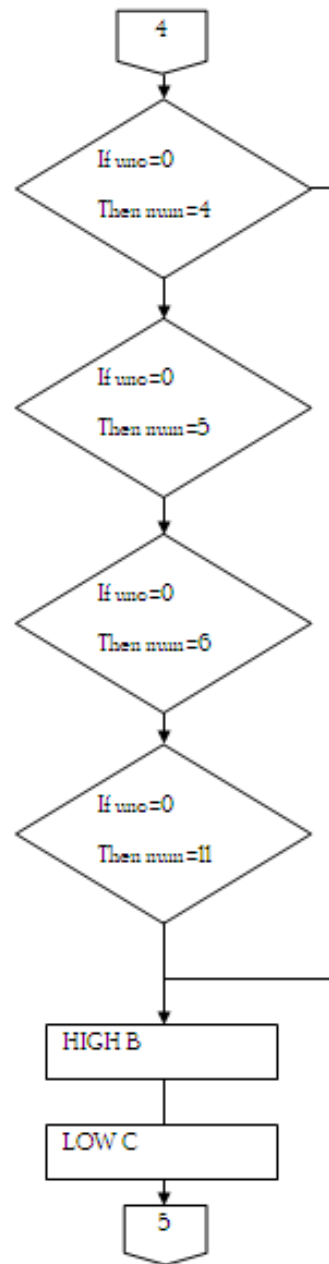
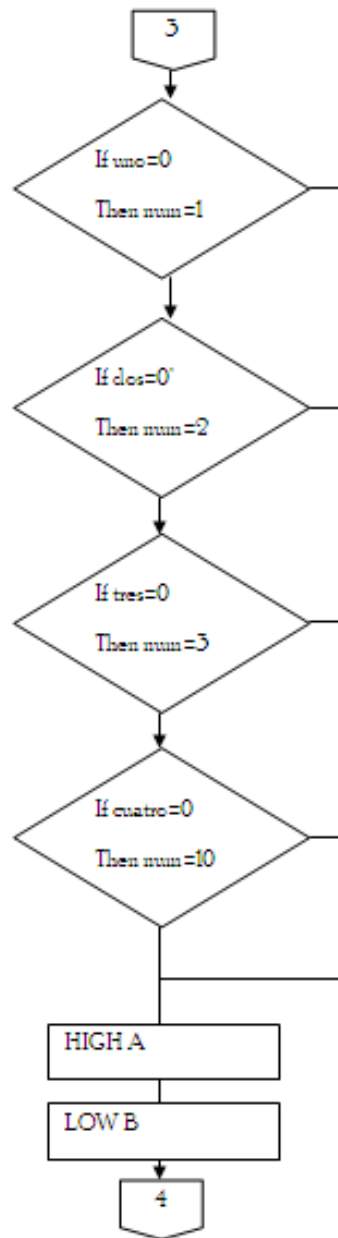
pause 200
DEFINE LCD_DREG PORTB
DEFINE LCD_DBIT 4
DEFINE LCD_RSREG PORTB
DEFINE LCD_RSBIT 2
DEFINE LCD_EREG PORTB
DEFINE LCD_EBIT 3
DEFINE LCD_BITS 4
DEFINE LCD_LINES 2
x var byte                                ;crea la variable x de un tamaño de 255
inicio:
for x= 1 to 3                             ;repite tres veces las instrucciones
  lcdout $fe,1,"U.N.L."                   ;limpiar pantalla y sacar el texto U.N.L.
  pause 1000
  lcdout $fe,1,"Eletrônica"               ;limpiar pantalla y sacar el texto Electrónica.
  pause 1000
  lcdout $fe,1,"Sexto-Modulo"             ;limpiar pantalla y sacar el texto Sexto Módulo.
  pause 1000
  LCDOUT $FE,$C0,"Microcontrolador";limpiar pantalla y sacar el texto Micro.
  pause 1000
next x
goto inicio
end

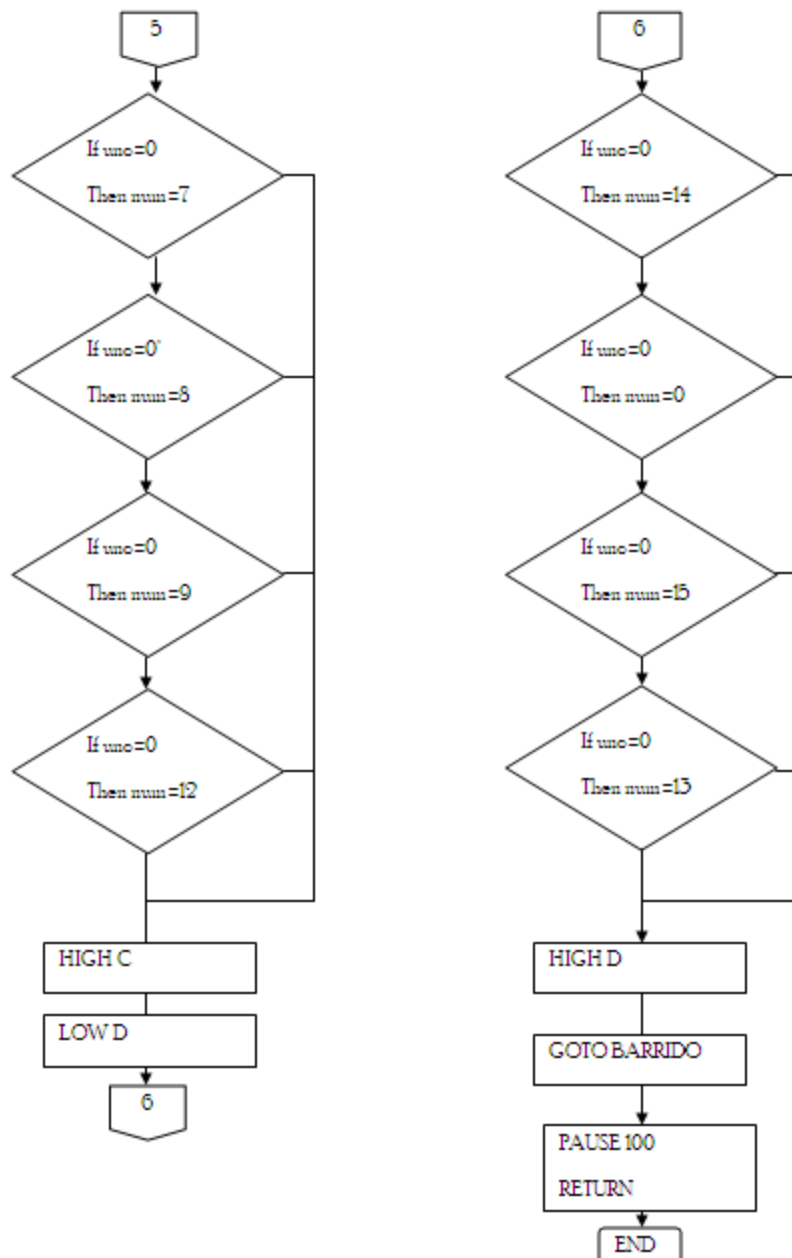
```

3.6 Manejo del Teclado Hexadecimal:

3.6.1 Diagrama de Flujo:







3.6.2 Programación en micro-code-studio:

```

DEFINE LCD_DREG PORTD
DEFINE LCD_DBIT 4
DEFINE LCD_RSREG PORTD
DEFINE LCD_RSBIT 2
DEFINE LCD_EREG PORTD
DEFINE LCD_EBIT 3
  
```

```

A VAR PORTB.3
B VAR PORTB.2
C VAR PORTB.1
  
```

```

;define pines del LCD D4, a D7
;empezando desde el Puerto D4 al D7
;define pin para conectar al bit RS
;en el Puerto D2
;define pin para conectar el bit enable
;en el puerto D3
  
```

```

;nombres para los pines de las filas
  
```

D VAR PORTB.0

UNO VAR PORTB.7 ;nombres para los pines de las columnas
 DOS VAR PORTB.6
 TRES VAR PORTB.5
 CUATRO VAR PORTB.4

TRISD = 0

NUM VAR BYTE
 NUM = 0

INI:

```

SELECT CASE NUM
CASE 10
LCDOUT $FE,1,"A" ; sacar A en el lcd
CASE 11
LCDOUT $FE,1,"B" ;sacar B en el lcd
CASE 12
LCDOUT $FE,1,"C" ,sacar C en el lcd
CASE 13
LCDOUT $FE,1,"D" ;sacar D en el lcd
CASE 14
LCDOUT $FE,1,"*" ;sacar * en el lcd
CASE 15
LCDOUT $FE,1,"#" ;sacar # en el lcd
CASE ELSE
LCDOUT $FE,1,DEC NUM
END SELECT
GOSUB BARRIDO
GOSUB TECLA
GOTO INI

```

BARRIDO:

```

LOW A ;hacer bajo la 1era fila
IF UNO = 0 THEN NUM = 1: return ;si la 1ra tecla es presionada sacar 1
IF DOS = 0 THEN NUM = 2: return ;si la 2da tecla es presionada sacar 2
IF TRES = 0 THEN NUM = 3: return ;si la 3ra tecla es presionada sacar 3
IF CUATRO = 0 THEN NUM = 10: return ;si la 4ta tecla es presionada
sacar 10
HIGH A ;poner en alto la 1era fila
LOW B ;hacer bajo la 2da fila
IF UNO = 0 THEN NUM = 4: return
IF DOS = 0 THEN NUM = 5: return

```

```

IF TRES = 0 THEN NUM = 6: return
IF CUATRO = 0 THEN NUM = 11: return
  HIGH B                                ;poner en alto la 2da fila
  LOW C                                 ;hacer bajo la 3ra fila
IF UNO = 0 THEN NUM = 7: return
IF DOS = 0 THEN NUM = 8: return
IF TRES = 0 THEN NUM = 9: return
IF CUATRO = 0 THEN NUM = 12: return
  HIGH C                                ;poner en alto la 3ra fila
  LOW D                                 ;hacer bajo la 4ta fila
IF UNO = 0 THEN NUM = 14: return
IF DOS = 0 THEN NUM = 0: return
IF TRES = 0 THEN NUM = 15: return
IF CUATRO = 0 THEN NUM = 13: return
  HIGH D                                ;poner en alto la 4ta fila
                                          ;pausa de 10 milisegundos

```

goto barrido

TECLA:

```

IF UNO = 0 THEN TECLA
IF DOS = 0 THEN TECLA
IF TRES = 0 THEN TECLA
IF CUATRO = 0 THEN TECLA
PAUSE 100
RETURN

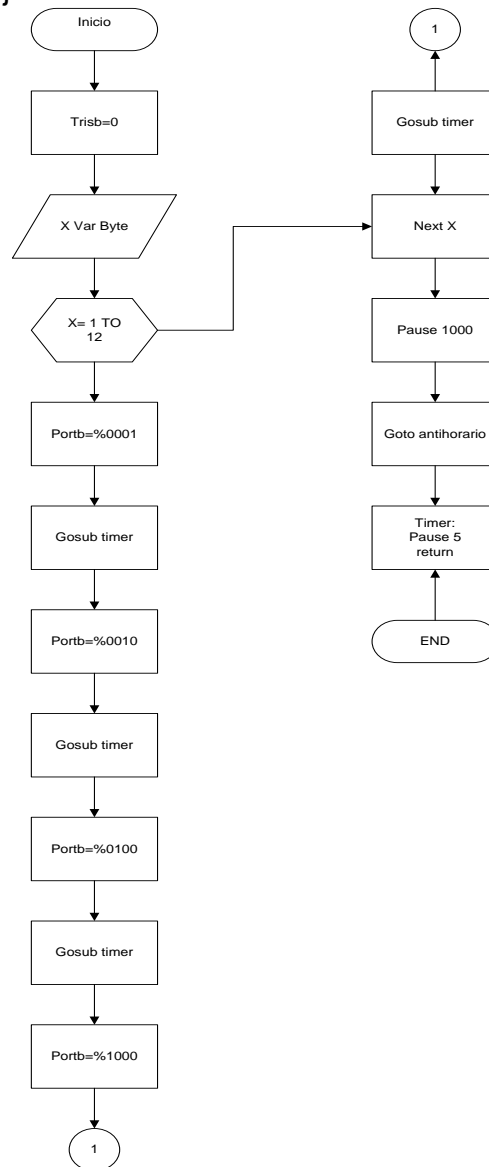
```

;continuar con el barrido de teclas

END

3.7 Manejo de un Motor de Paso:

3.7.1 Diagrama de Flujo:



3.7.2 Programación en micro-code-studio:

```

trisb=0      ;declara salidas para el puerto b
x var byte   ;variable x de 255
antihorario:
for x= 1 to 12 ;12 veces repetir secuencia de giro antihorario
portb=%0001 ;energiza bobina A
gosub timer ;espera 5mls
portb=%0010 ;energiza bobina C
gosub timer ;espera 5mls
portb=%0100 ;energiza bobina B
gosub timer ;espera 5mls
portb=%1000 ;energiza bobina D
  
```

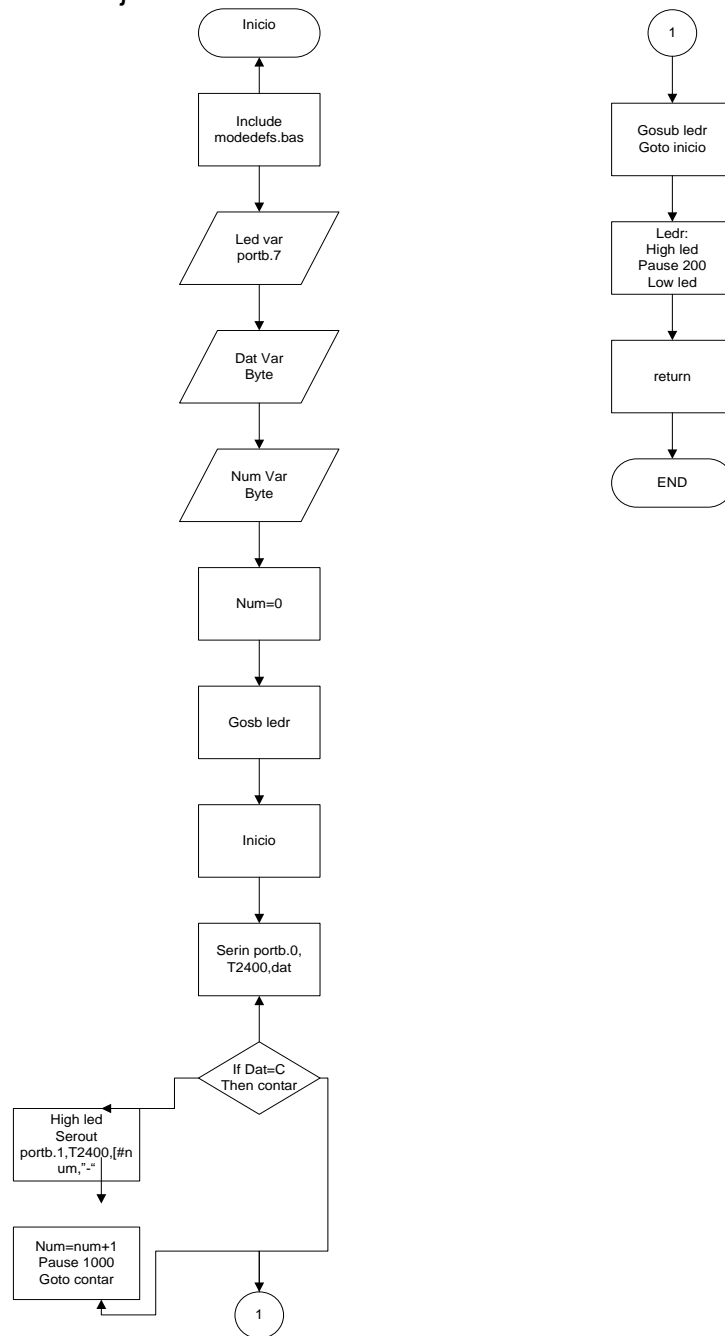
```
gosub timer    ;espera 5 mls  
next x  
pause 1000
```

```
Next x  
pause 1000
```

```
goto antihorario  
timer:  
pause 5        ;pausa de 5 mls  
return  
end
```

3.8 Comunicación Serial PC a PIC y PIC a PC:

3.8.1 Diagrama de Flujo:



3.8.2 Programación en micro-code-studio:

```

include "modedefs.bas" ;incluyen los modos de comunicación
led var portb.7         ;nombre led al puerto b7
dat var byte            ;variable de almacenamiento de 255
num var byte            ;variable para almacenar el contador
num=0                  ;contenido inicial para la variable num
gosub ledr              ;ir a leds para saber si ya arranco el pic
Inicio:

```

```

serin portb.0,T2400,dad ;esperar el dato y guardarlo en dad
if dad= "C" then contar ;si dad es empezar contar
gosub ledr
goto inicio ;ir a esperar el siguiente caracter
contar:
high led ;led solo encendido indica enviando datos
serout portb.1,T2400,[#num,"-"] ;enviar el contenido decimal de la variable num
num= num+1 ;incrementar la variable 1*1
pause 1000 ;espera de 1S
Goto contar ;ir a subrutina contar
ledr:
high led ;Encender led
pause 200
low led ;apagar led
return ;regresa a la etiqueta inicio
end

```

3.9 Termómetro Digital:

3.9.1 Diagrama de flujo:



3.9.2 Programación en micro-code-studio:

```

pause 200
DEFINE LCD_DREG PORTB ;define pines del LCD B4 y B7
DEFINE LCD_DBIT 4 ;empezando desde el puerto B4 al B7
DEFINE LCD_RSREG PORTB ;define pin para conectar el RS
DEFINE LCD_RSBIT 3 ;en el puerto B3
DEFINE LCD_EREG PORTB ;define pin para conectar el bit enable
DEFINE LCD_EBIT 2 ;en el puerto B2
DEFINE LCD_BITS 4 ;define comunicación en 4 bits con el LCD
DEFINE LCD_LINES 2 ;define un LCD de 2 Líneas

define ADC_BITS 8 ;fije número de bits del resultado (5,8,10)
define ADC_CLOCK 3 ;fije El CLOCK(re= 3)
define ADC_SAMPLEUS 50 ;fije el tiempo de muestro en Us
;ADC_SAMPLEUS es el número de microsegundos que el programa
espera
;entre fijar el canal y comenzar la conversión analógica/digital
TRISA= %1 ;el puerto A es de entrada
ADCON1= %00001110 ;el puerto A.0 es conversor los demas digitales

datos var byte ;crear variable dato para guardar
temp var byte ;crear variable dato para guardar
dectemp varbyte ;crear variable dato para guardar
pause 500 ;esperar 0,5 seg.

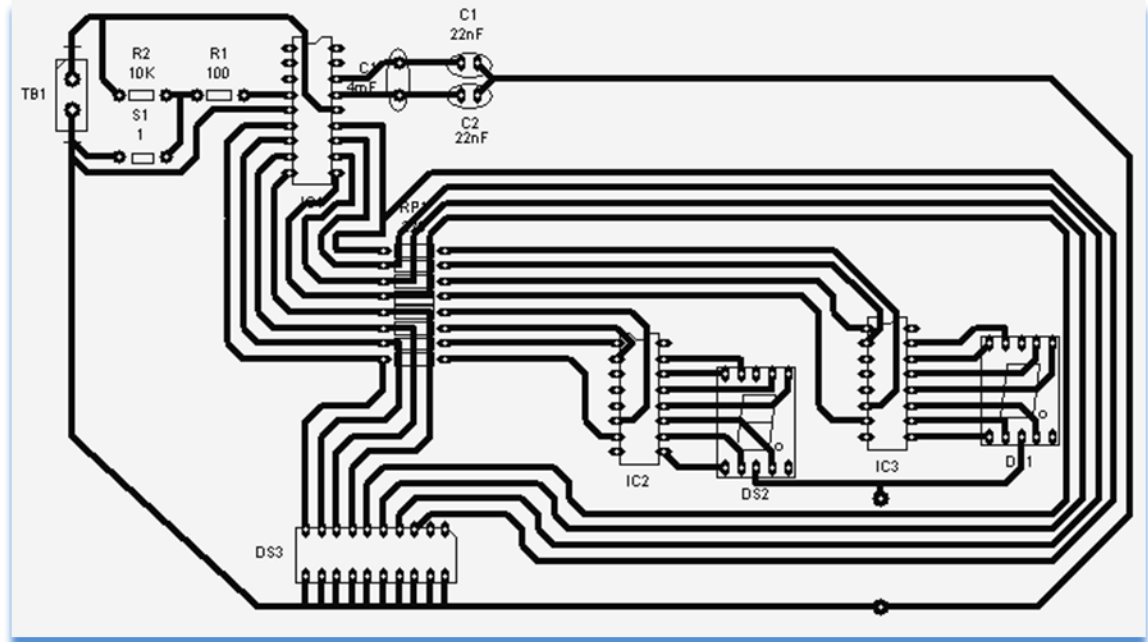
inicio:
adcin 0,datos ;leer el canal 0 y guardarlo en datos
temp= (datos*50)/255
lcdout $fe,1,"temperatura="
lcdout $fe,$c7,dec temp ;desplegar el valor de datos en decimal
pause 300
goto inicio ;volver a medir el conversor A/D
end

```

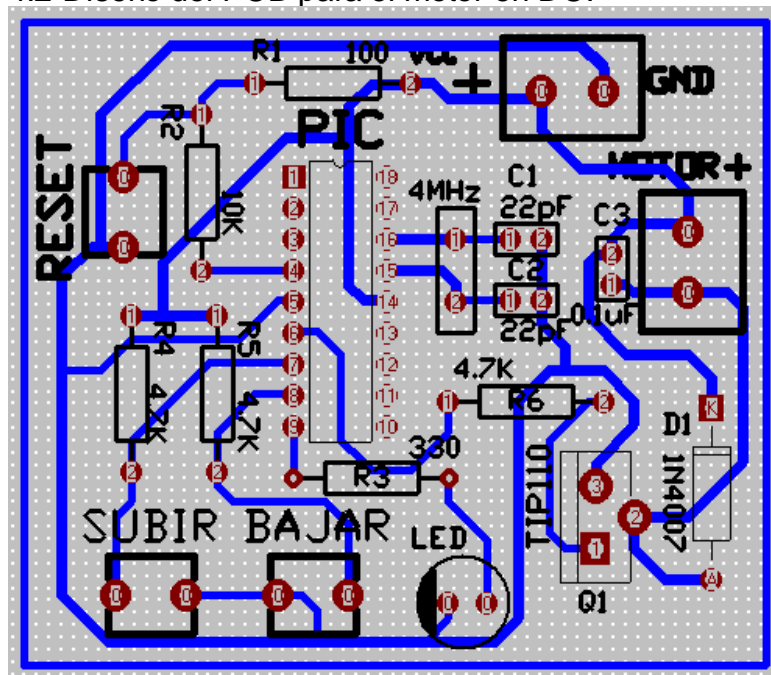
CAPITULO IV

4. ELABORACIÓN DE LOS PCB'S

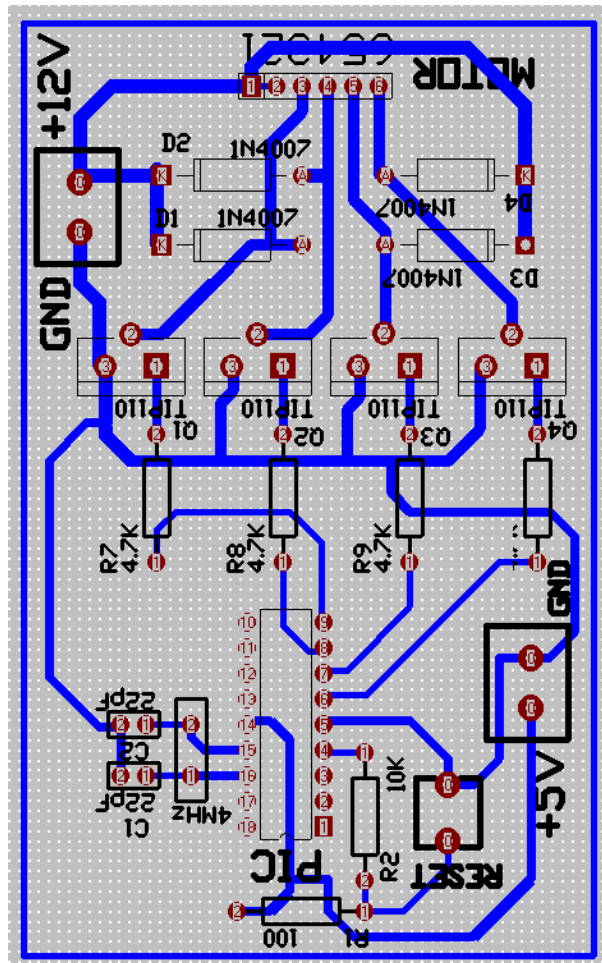
4.1 Diseño del PCB para Display de 7 segmentos y barra de leds:



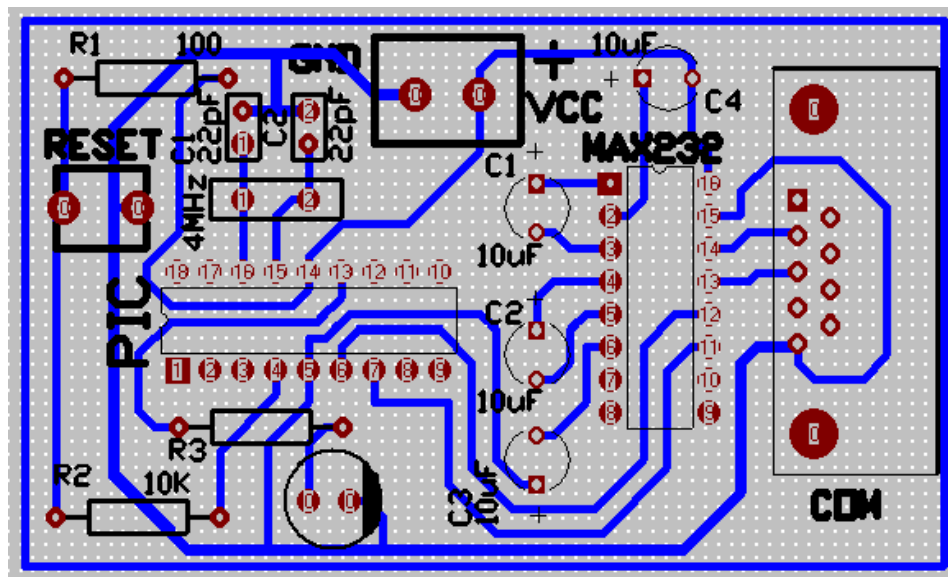
4.2 Diseño del PCB para el motor en DC:



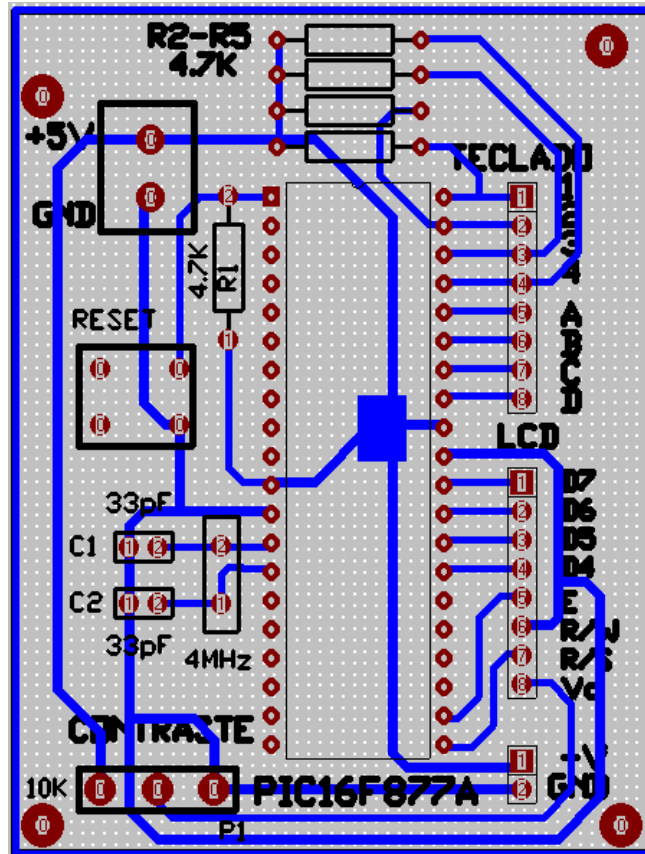
4.3 Diseño del PCB para el motor paso a paso:



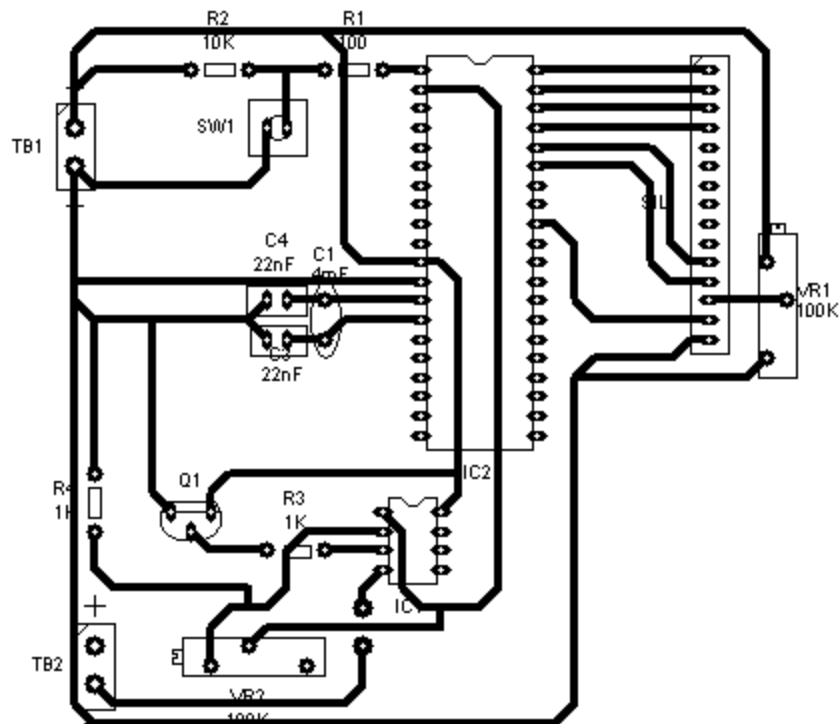
4.4 Diseño del PCB para la comunicación serial:



4.5 Diseño del PCB para el manejo del teclado y LCD:



4.6 Diseño del PCB para el termómetro digital:



CAPITULO V

5. GUIA DE PRÁCTICAS

5.1 Preparatorio de la Práctica #1

1. Defina qué es un PIC?

.....

2. Data sheet del Pic 16F84

.....

3. ¿Qué es el programa Micro-Code-Studio?

.....

4. Qué son las sentencias, Inicio, HIGH, PAUSE, LOW, GOTO, END?

.....

5.2 Práctica # 1

Tema: Manejo del Puerto B

Objetivos:

- ✧ Aprender a familiarizarse con el Pic.
- ✧ Definir cuales son Los pines del puerto B
- ✧ Encender un led encendido en el puerto B0
- ✧ Apagar un led en el puerto B0

Materiales:

Cantidad	Descripción	Características
1	Grabador de Pics	Puerto Serial
1	Pc	WindowsXP
1	Pic16f84	Microchip
1	Micro-Code-Studio	Lenguaje de Programa
3	Resistencias	100ohms,1K y 10K
1	Cristal	4Mhz
1	Diodo	Led
1	Pulsador	Reseteador
1	Protoboard	Entrenador

Esquema:

Desarrollo:

Para realizar esta práctica necesitamos grabar el Pic en el lenguaje de programación Micro-Code-Studio, una vez echo esto lo copilamos y grabamos si no sale ningún mensaje de error lo ensamblamos en el programa Ic-Prog que esta adjuntado al anterior mencionado, necesitamos tener un ligero conocimiento de las sentencias que se utilizaran en esta práctica, y la programación de la práctica quedaría de la siguiente manera:

```

Inicio:           ; etiqueta del programa
HIGH portb.0      ; sacar un 1 por el puerto b0
PAUSE 1000        ; demora 1 segundo
LOW portb.0       ; sacar un 0 por el puerto b0
PAUSE 1000        ; demora 1 segundo
GOTO inicio       ; repite la etiqueta inicio
END               ; finaliza el programa

```

Una vez programado el Pic16f84, necesitamos armar el circuito para eso nos vamos a guiar en el esquema, primeramente armaremos el oscilador que consta de 2 capacitores de 22pf y un Cristal de 4Mhz, estos se conectan en serie en los pines 15 y 16, se unen los 2 extremos de los condensadores y van a tierra, luego partimos del pin4 (MCLR) que se conecta a una resistencia de 100ohms y este a su vez se une en serie a una resistencia de 10Kohms el extremo sobrante de esta se conecta a Vcc= 5V; partimos del medio de las dos resistencias con un pulsador y el otro extremo del pulsador va a tierra.

Utilizaremos el puerto B0 el cual corresponde al pin 6 se conecta al diodo de ánodo a cátodo y este va en serie a una resistencia de 1Kohm y el otro extremo de la resistencia va a tierra, los pines 14 van a Vcc y 5 van a GND o tierra.

Por ultimo lo alimentamos con un voltaje continuo de 5V y lo probamos.

Conclusiones:

- ☆ El puerto B.0 se puede utilizar para encender y apagar un led .
- ☆ El micro code_studio nos permite programar diferentes rutinas

Recomendaciones:

- ☆ Tener muy en cuenta la numeración de los pines al momento de ensamblarlo.
- ☆ La fuente de poder debe estar regulada a 5V de corriente continua.
- ☆ El micro code_studio se lo puede utilizar en cualquier tipo de circuitos. .

5.3 Preparatorio de la Práctica # 2

1. ¿Qué significa la sentencia tris?

.....

2. ¿Qué significa la etiqueta Inicio:?

.....

3. ¿Qué es un diodo led?

.....

5.4 Práctica # 2

Tema: Juego de luces

Objetivos:

- ☆ Utilizar todos los pines del puerto B como salidas
- ☆ Encender y apagar los leds en forma secuencial
- ☆ Comprobar el funcionamiento del reset

Materiales:

Cantidad	Descripción	Características
1	Grabador de Pics	Puerto Serial
1	Pc	WindowsXP
1	PIC16f84	Microchip
1	Micro-Code-Studio	Lenguaje de Programa
2	Resistencias	100ohms, y 10K
1	Cristal	4Mhz
10	Diodo	Led
1	Pulsador	Reseteador
1	Protoboard	Entrenador
10	Resistencias	1Kohm

Esquema:

Desarrollo:

Para efectuar esta práctica utilizaremos una nueva sentencia que declarará a todos los pines del puerto B como salidas, seguiremos casi todos los pasos de la práctica anterior con la diferencia que ahora utilizaremos todas las salidas del puerto B, la programación en Micro-Code-Studio quedará de la siguiente forma:

```

trisb=0          ;todos los pines del puerto B son de salida
inicio:          ;nombre de subrutina inicio
portb= %00000001 ;solo enciende el b0
pause 1000       ;espera de tiempo
portb= %00000010 ;solo enciende el b1
pause 1000       ;espera de tiempo
portb= %00000100 ;solo enciende el b2
pause 1000       ;espera de tiempo
portb= %00001000 ;solo enciende el b3
pause 1000       ;espera de tiempo
portb= %00010000 ;solo enciende el b4
pause 1000       ;espera de tiempo
portb= %00100000 ;solo enciende el b5
pause 1000       ;espera de tiempo
portb= %01000000 ;solo enciende el b6
pause 1000       ;espera de tiempo
portb= %10000000 ;solo enciende el b7
pause 1000       ;espera de tiempo
goto inicio      ;repite la etiqueta inicio
end              ;fin del programa

```

Una vez programado estas sentencias en el Pic procedemos a armar de la misma forma como en la práctica anterior, con la diferencia de que aumentamos más leds a la salida.

Esto implica que no podemos equivocarnos en las salidas de los pines y en la programación ya que sino no obtendremos lo propuesto.

Conclusiones:

- ☆ Los pines utilizados nos permiten encender los leds de izquierda a derecha o viceversa.
- ☆ El funcionamiento del programa simula los leds de forma secuencial y repetitiva.

Recomendaciones:

- ☆ Tener bien definida la estructura de los leds de ánodo y cátodo.
- ☆ Tener un Cristal de 4 Mhz de repuesto y un Pic.

5.5 Preparatorio de la Práctica # 3

1. ¿Qué es un Display de 7 Segmentos de Cátodo común?

.....

2. ¿Para qué sirve el Circuito integrado IC74LS48?

.....

3. ¿Para qué sirven las sentencias FOR, NEXT, VAR, BYTE?

.....

5.6 Práctica # 3

Tema: Contadores con displays de 7 segmentos

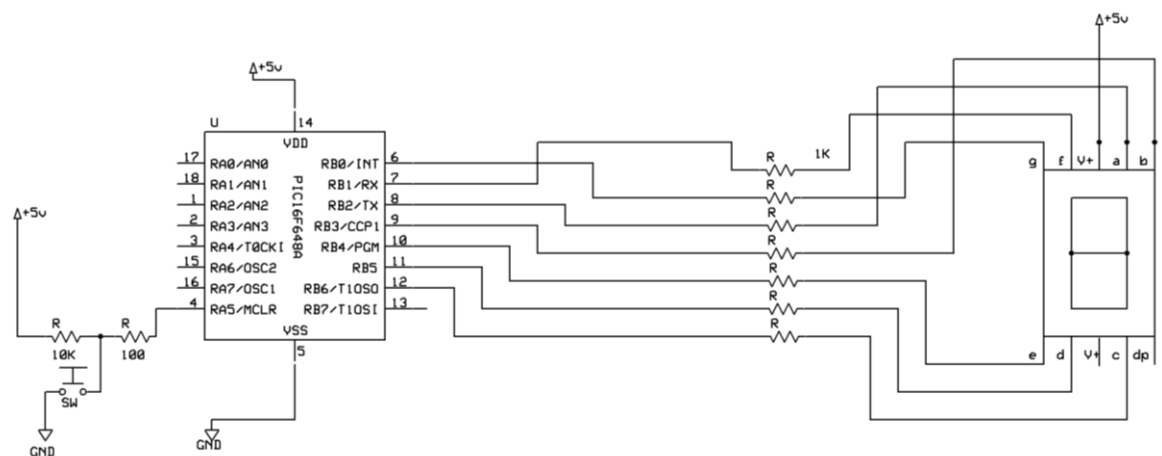
Objetivos:

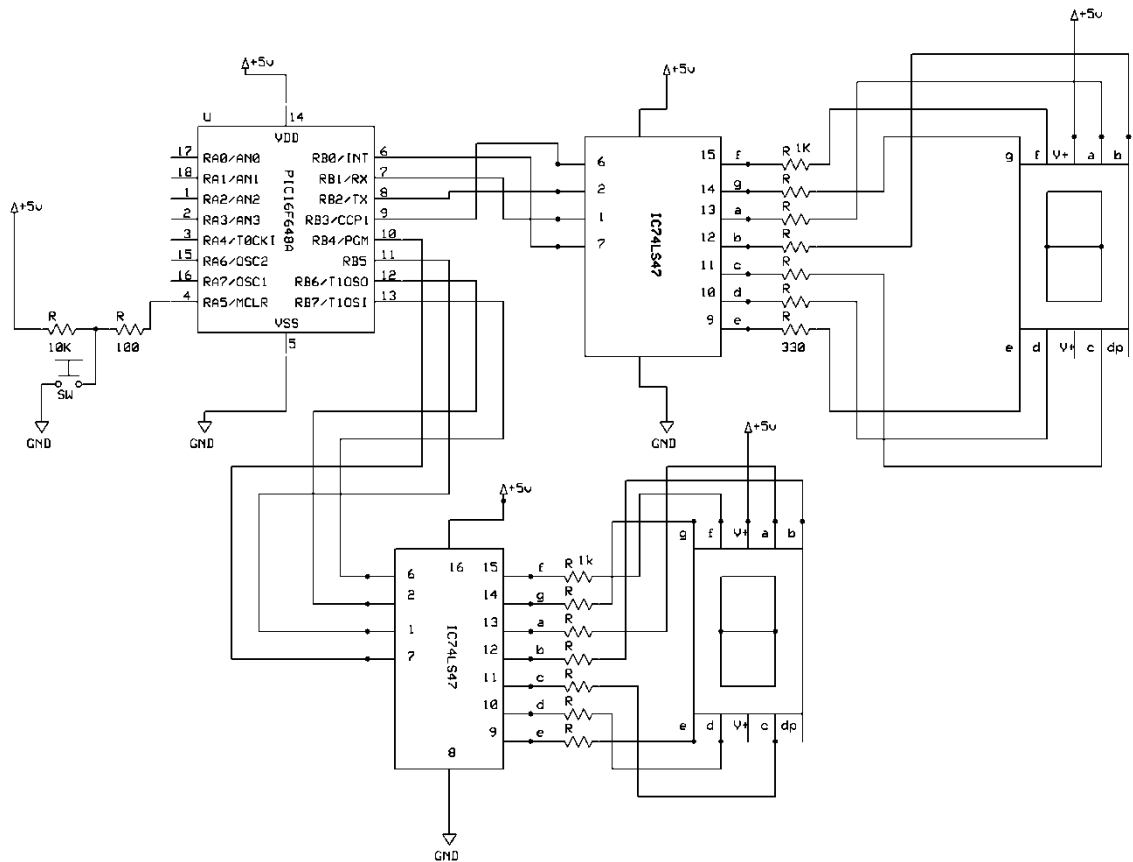
1. Programar un contador de 0 a 99
2. Programar un contador de 0 a 9 suprimiendo los integrados IC74LS48

Materiales:

Cantidad	Descripción	Características
1	Grabador de Pics	Puerto Serial
1	Pc	WindowsXP
1	Pic16f84	Microchip
1	Micro-Code-Studio	Lenguaje de Programa
2	Resistencias	100ohms, y 10K
1	Cristal	4Mhz
2	Displays	7 Segmentos
1	Pulsador	Reseteador
1	Protoboard	Entrenador
10	Resistencias	1Kohm

Esquemas:





Desarrollo:

En la presente práctica se va a utilizar varios displays de 7 segmentos los cuales nos permitirán visualizar numeración entre 0 a 9, para ello nos guiaremos en el gráfico el cual nos permitirá realizar las conexiones respectivas en el protoboard se debe realizar este trabajo con cuidado, una vez echo esto no dirigimos a programar en el Microcode-studio de la siguiente:

contador de 0 a 9

y var byte

x var byte

trisb= 0

contador:

for x= 0 to 9

lookup x,[63,6,91,79,102,109,124,7,127,111],y

portb= y

pause 1000

next x

goto contador

end

contador de 0 a 99

```

x var byte
y var byte
z var byte
trisb=0
contador:
  for x=0 to 9
    lookup x,[0,16,32,48,64,80,96,112,128,144],y      ;contador de decenas
    portb=y
  for z=0 to 9
    portb=y+z      ;contador de unidades
    pause 1000
  next z
next x
goto contador
end

```

Una vez programado colocamos el Pic en el Protho y nos disponemos a probarlo.

Conclusiones:

- ☆ Se puede observar que el contador de 0 a 9 mantiene una duración de 1 segundo, para cambiar el número todo depende del tiempo que nosotros lo asignemos.
- ☆ El contador de 0 a 99 utiliza unos decodificadores que se encargan de sacar mas salidas para el display.

Recomendaciones:

- ☆ Se recomienda utilizar un decodificador 74LS47
- ☆ Los decodificadores soportan un voltaje máximo de 5V.
- ☆ Hay que tomar en cuenta que los displays son de Cátodo común por lo que funcionan solamente con tierra.

5.7 Preparatorio de la Práctica # 4

1. ¿Qué significan las siglas PWM?

.....

2. ¿Qué son las sentencias If, PWM?

.....

5.8 Práctica #4

Tema: Modulación por Ancho de Pulsos

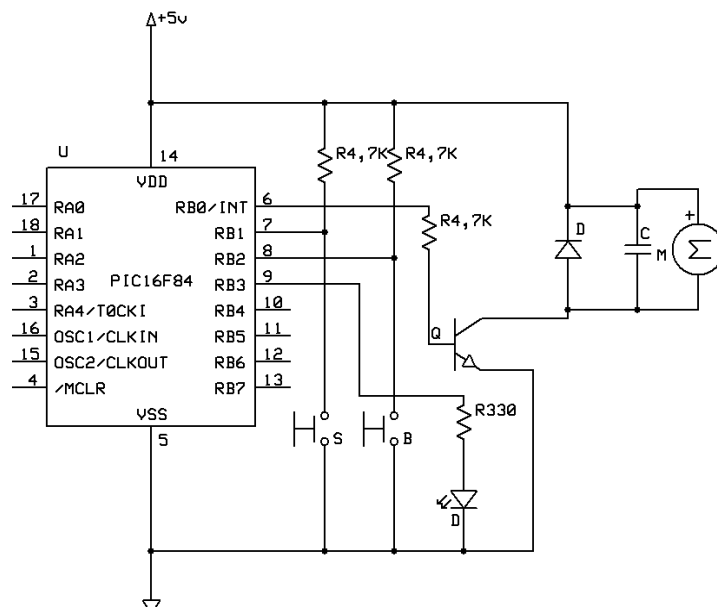
Objetivos:

- ✧ Analizar las formas de onda en un osciloscopio
- ✧ Determinar el porcentaje del ancho del pulso
- ✧ Armar un circuito educativo con el PIC.

Materiales:

Cantidad	Descripción	Características
1	Grabador de Pics	Puerto Serial
1	Pc	WindowsXP
1	Pic16f84	Microchip
1	Micro-Code-Studio	Lenguaje de Programa
2	Resistencias	100ohms, y 10K
1	Cristal	4Mhz
1	Pulsador	Reseteador
1	Protoboard	Entrenador
4	Resistencias	1Kohm, 4.7K, 330ohms
1	Transistor	TIP 110
2	Pulsadores	Abiertos
1	Motor	DC

Esquema:



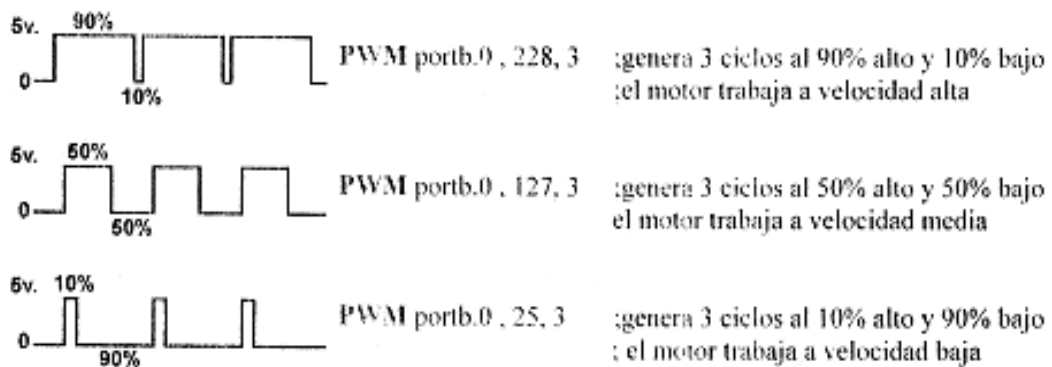
Desarrollo:

En la presente práctica nos proponemos a armar un circuito manejado por ancho de pulsos para lo cual nos fijaremos en la grafica, luego programaremos en Pic en el Micro-code-studio, para luego ser grabado.

El PWM (Pulse Width Modulation) o modulación en ancho del pulso, tiene muchas aplicaciones, por ejemplo para atenuar la iluminación de un led, para variar la velocidad de un motor en DC.

PWM portB.0, 127, 60 ; quiere decir sacar 60 pulsos PWM por el puerto B.0 al 50% en alto
; aproximadamente

La forma de la señal que sale por el PIC es similar a los siguientes gráficos:



Por consiguiente 0 representa 0% de ciclo útil y 255 el 100% de nivel alto, el largo de cada ciclo para un oscilador de 4MHZ es de **5 milisegundos** y para un oscilador de 20MHZ es de 1 milisegundo

Fig. 5.1 Representación de un ciclo cuadrado en PWM

Esta es la programación en el Micro-code:

```

botsubir var portb.1      ;nombre para el pin B1
botbajar var portb.2      ;nombre para el pin B2
LED var portb.3           ;nombre para el pin B3
FREC var byte             ;variable frecuencia tamaño 255
bandera var bit           ;variable bandera de 1 bit
FREC= 125                 ;valor inicial para variable FREC
high LED                  ;ecender led para saber que ya arranco
pause 500                 ;por medio segundo
low LED                   ;apagar led
PROG:
pwm portb.0,FREC,30       ;sacar PWM 30 ciclos de 125 señal útil
low led                   ;apagar el led
if botsubir=0 then subir  ;si se pulsa el boton S ir a subir
if botbajar=0 then bajar  ;si se pulsa el boton B bajar
bandera=0                 ;bandera cargado con 0
  
```

```

goto PROG
Subir:
if FREC > 249 then aviso      ;si supera a 249 ir a aviso
if bandera= 1 then prog      ;si la bandera esta en 1 salir
high LED                     ;encender el led
bandera=1                    ;cargar la bandera como uno
FREC= FREC + 25               ;sumar 25 a la variable FREC
goto PROG                     ;ir a prog
Bajar:
if Frec < 26 then aviso      ;si baja de 26 ir a aviso
if bandera= 1 then prog      ;si la bandera está en 1 salir
high LED                     ;encender el led
bandera=1                    ;cargar la bandera como 1
FREC= FREC-25                 ;restar 25 a la variable FREC
goto prog                     ; ir a prog
aviso:
high LED                     ;encender el led
goto prog                     ;regresa a PROG
end

```

Nos fijaremos que al momento de partir el motor el led se encenderá automáticamente y al momento de pulsar el cambio de pulso también se encenderá el led para demostrar el cambio en el motor.

Conclusiones:

- ☆ Se puede ver que el motor empieza a girar con una frecuencia de 125, al pulsar el botón S se incrementa la variable en múltiplos de 25 y sube la frecuencia hasta llegar a 250 que es el valor máximo del motor.
- ☆ Al momento de pulsar el botón B el motor disminuye la frecuencia en periodos de 25.

Recomendaciones:

- ☆ Tener cuidado al momento de manejar el motor ya que es muy frágil y se pueden dañar con facilidad.
- ☆ El transistor TIP110 es de configuración NPN y debemos tomar en cuenta dicha configuración al momento de conectarlo.

5.9 Preparatorio de la práctica # 5

1. ¿Qué es un LCD?

.....

2. Data Sheet del LCD

.....

3. ¿Qué son las sentencias LCDOUT, DEFINE, VAR BYTE, FOR?

.....

5.10 Práctica # 5

Tema: Manejo de un LCD

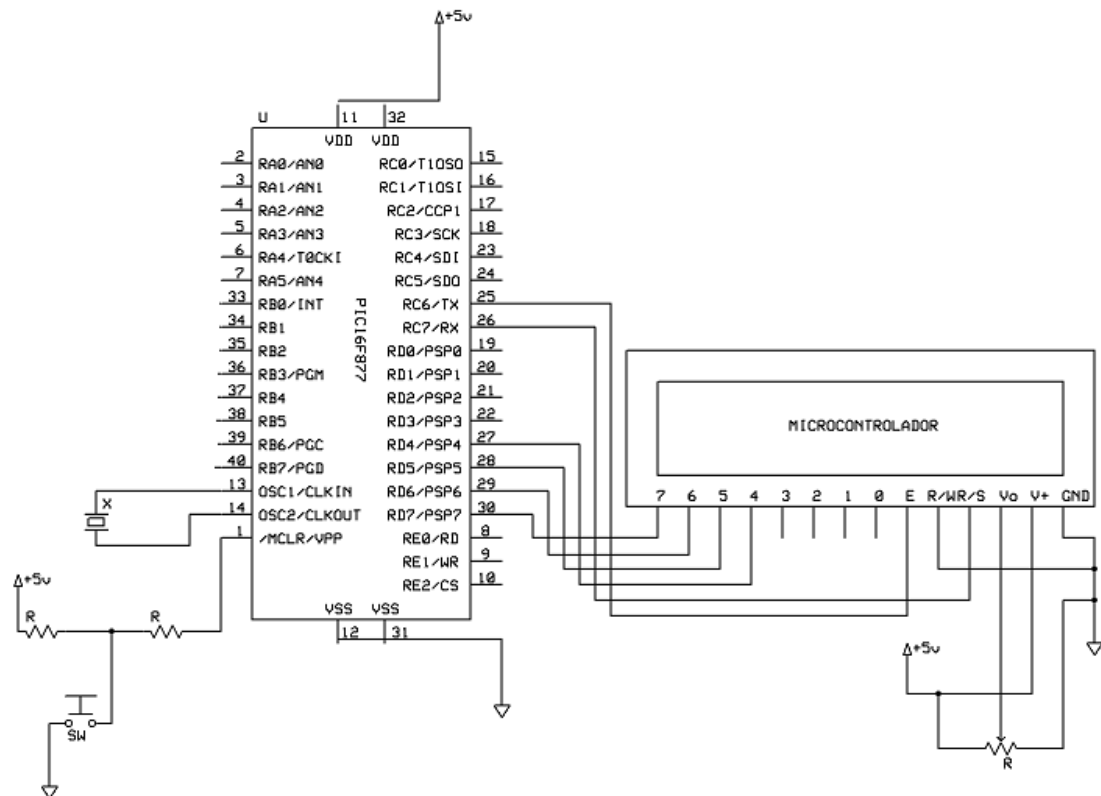
Objetivos:

- ☆ Visualizar datos o letras por medio de un LCD.
- ☆ Aprender a conectar un LCD controlado por el PIC

Materiales:

Cantidad	Descripción	Características
1	Grabador de Pics	Puerto Serial
1	Pc	WindowsXP
1	Pic16f877	Microchip
1	Micro-Code-Studio	Lenguaje de Programa
2	Resistencias	100ohms, y 10K
1	Cristal	4Mhz
1	LCD	2*16 líneas
1	Fuente	5V DC

Esquema:



Desarrollo:

La utilización de un LCD es muy importante ya que gracias a esta podemos visualizar Datos, letras; para realizar esta práctica debemos seguir la siguiente estructura de código en Micro-code-studio de la siguiente manera:

pause 200

```
DEFINE LCD_DREG PORTB
```

```
DEFINE LCD_DBIT 4
```

```
DEFINE LCD_RSREG PORTB
```

```
DEFINE LCD_RSBIT 2
```

```
DEFINE LCD_EREG PORTB
```

```
DEFINE LCD_EBIT 3
```

```
DEFINE LCD_BITS 4
```

```
DEFINE LCD_LINES 2
```

```
x var byte ;crea la variable x de un tamaño de 255
```

inicio:

```
for x= 1 to 3 ;repite tres veces las instrucciones
```

```
lcdout $fe,1,"U.N.L." ;limpiar pantalla y sacar el texto U.N.L.
```

```
pause 1000
```

```
lcdout $fe,1,"Eletrônica"
```

```
pause 1000
```

```
lcdout $fe,1,"Sexto-Modulo"  
pause 1000  
LCDOUT $FE,$C0,"Microcontrolador"  
pause 1000  
next x  
goto inicio  
end
```

Conclusiones:

- ☆ Podemos darnos cuenta que en la pantalla LCD se muestra caracteres de acuerdo a la programación que se ha hecho.
- ☆ La pantalla LCD consta de 14 pines los mismos que tienen su respectiva descripción o numeración.

Recomendaciones:

- ☆ Se recomienda conectar la pantalla LCD en un solo puerto del Pic que se utilice.
- ☆ Se puede utilizar una pantalla LCD de 14 pines que es la más común que se encuentra a la venta.

5.11 Preparatorio de la Práctica # 6

1. Qué es un Teclado Hexadecimal
.....
2. Data sheet
.....
3. Qué son la sentencias VAR, LCDOUT, IF
.....

5.12 Práctica # 6

Tema: Manejo del teclado Hexadecimal

Objetivos:

- ☆ Utilizar un teclado hexadecimal que nos permitirá sacar números en el LCD.
- ☆ Aprender el funcionamiento del teclado.

Materiales:

Cantidad	Descripción	Características
1	Grabador de Pics	Puerto Serial
1	Pc	WindowsXP
1	Pic16f877	Microchip
1	Micro-Code-Studio	Lenguaje de Programa
2	Resistencias	100ohms, y 10K
1	Cristal	4Mhz
1	LCD	2*16 lineas
1	Fuente	5V DC
1	Teclado	Hexadecimal

Esquema:**Desarrollo:**

En la presente práctica utilizaremos los puertos B, C y D los mismos que serán configurados para manejar un teclado hexadecimal de 4 filas X 4 columnas el mismo que nos permitirá observar un dato respectivo en el LCD, la programación se la verá a continuación:

```

DEFINE LCD_DREG PORTD
DEFINE LCD_DBIT 4
DEFINE LCD_RSREG PORTD
DEFINE LCD_RSBIT 2
DEFINE LCD_EREG PORTD
DEFINE LCD_EBIT 3

```

```

A VAR PORTB.3
B VAR PORTB.2
C VAR PORTB.1
D VAR PORTB.0

```

```

UNO VAR PORTB.7
DOS VAR PORTB.6
TRES VAR PORTB.5
CUATRO VAR PORTB.4

```

```

TRISD = 0

```

```

NUM VAR BYTE
NUM = 0

```

INI:

```

SELECT CASE NUM
CASE 10
LCDOUT $FE,1,"A"
CASE 11
LCDOUT $FE,1,"B"
CASE 12
LCDOUT $FE,1,"C"
CASE 13
LCDOUT $FE,1,"D"
CASE 14
LCDOUT $FE,1,"*"
CASE 15
LCDOUT $FE,1,"#"
CASE ELSE
LCDOUT $FE,1,DEC NUM
END SELECT
GOSUB BARRIDO
GOSUB TECLA
GOTO INI

```

BARRIDO:

```

LOW A                                ;hacer bajo la 1era fila
IF UNO = 0 THEN NUM = 1: return      ;si la 1ra tecla es presionada sacar 1
IF DOS = 0 THEN NUM = 2: return      ;si la 2da tecla es presionada sacar 2
IF TRES = 0 THEN NUM = 3: return     ;si la 3ra tecla es presionada sacar 3
IF CUATRO = 0 THEN NUM = 10: return  ;si la 4ta tecla es presionada sacar 10
HIGH A                              ;poner en alto la 1era fila
LOW B                                ;hacer bajo la 2da fila
IF UNO = 0 THEN NUM = 4: return
IF DOS = 0 THEN NUM = 5: return
IF TRES = 0 THEN NUM = 6: return
IF CUATRO = 0 THEN NUM = 11: return
HIGH B                              ;poner en alto la 2da fila
LOW C                                ;hacer bajo la 3ra fila
IF UNO = 0 THEN NUM = 7: return
IF DOS = 0 THEN NUM = 8: return
IF TRES = 0 THEN NUM = 9: return
IF CUATRO = 0 THEN NUM = 12: return
HIGH C                              ;poner en alto la 3ra fila
LOW D                                ;hacer bajo la 4ta fila
IF UNO = 0 THEN NUM = 14: return
IF DOS = 0 THEN NUM = 0: return
IF TRES = 0 THEN NUM = 15: return
IF CUATRO = 0 THEN NUM = 13: return

```

```

HIGH D                                ;poner en alto la 4ta fila
                                        ;pausa de 10 milisegundos
goto barrido

TECLA:
IF UNO = 0 THEN TECLA
IF DOS = 0 THEN TECLA
IF TRES = 0 THEN TECLA
IF CUATRO = 0 THEN TECLA
PAUSE 100
RETURN                                ;continuar con el barrido de teclas

END

```

Conclusiones:

- ☆ Se puede mirar que al momento de pulsar una tecla podemos visualizar un número pulsado en la pantalla.
- ☆ La utilización del teclado es muy importante en la elaboración de cerraduras de seguridad.

Recomendaciones:

- ☆ La utilización del LCD es muy delicado debemos fijarnos muy bien en la numeración de sus pines.
- ☆ Existen LCD de 16 pines estos dos últimos son para encender un blaking o luz de fondo.

5.13 Preparatorio de la Práctica # 7

1. ¿Qué es un motor paso a paso?

.....

2. Data sheet del motor paso a paso

.....

3. Descripción del transistor tip110

.....

4. Descripción de un diodo.....

5.14 Práctica # 7

Tema: Manejo de un motor paso a paso

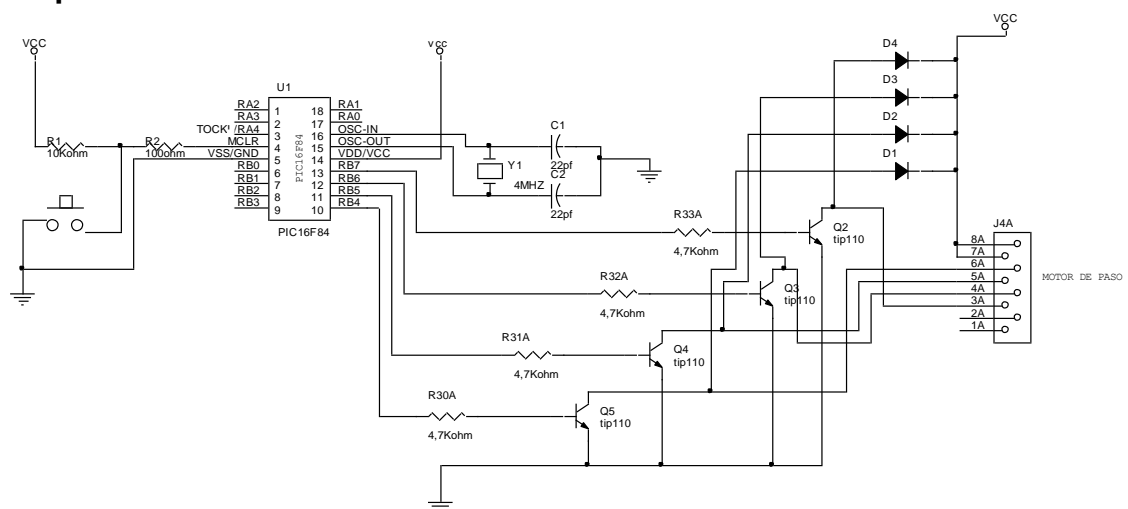
Objetivos:

- ✧ Conocer el funcionamiento de un motor paso a paso
- ✧ Demostrar los movimientos exactos del motor

Materiales:

Cantidad	Descripción	Características
1	Grabador de Pícs	Puerto Serial
1	Pc	WindowsXP
1	Pic16f84	Microchip
1	Micro-Code-Studio	Lenguaje de Programa
2	Resistencias	100ohms, y 10K
1	Cristal	4Mhz
1	LCD	2*16 líneas
1	Fuente	5V DC
1	Transistor tip110	Potencia

Esquema:



Desarrollo:

Los motores paso a paso (PAP) son ideales para la construcción de mecanismos en donde se requieren movimientos muy precisos, uno de los inconvenientes de estos motores a la hora de trabajar con ellos es poder identificar sus bobinas A,B,C,D; para esto simplemente medimos las resistencias de cada una de ellas nos guiaremos gracias al grafico al momento de conectarlo la programación es la siguiente:

```

trsb=0           ;hacer salidas el puerto b
x var byte       ;variable x de 255
antihorario:
for x= 1 to 12   ;12 veces repetir secuencia de giro antihorario
portb=%0001      ;energiza bobina A
gosub timer      ;espera 5mls
portb=%0010      ;energiza bobina C
gosub timer      ;espera 5mls
portb=%0100      ;energiza bobina B
gosub timer      ;espera 5mls
portb=%1000      ;energiza bobina D
gosub timer      ;espera 5 mls
next x
pause 1000

Next x
pause 1000

goto antihorario
timer:
pause 5          ;pausa de 5 mls
return
end

```

Conclusiones:

- ☆ Podemos darnos cuenta que el motor PAP de 6 hilos da un giro de 360° en 4 tiempos equivalentes a 90°
- ☆ Los motores PAP tienen una precisión exacta al momento de girar

Recomendaciones:

- ☆ Identificar los 2 hilos comunes para las demás bobinas
- ☆ Hay que tener muy en claro el orden de las bobinas A, B, C, D.

5.15 Preparatorio de la Práctica # 8

1. Descripción del Puerto Serial
2. Data sheet del puerto serial y pic16f628A
.....
3. Data sheet del max232
.....

5.16 Práctica #8

Tema: Comunicación serial Pc a Pic y Pic a Pc.

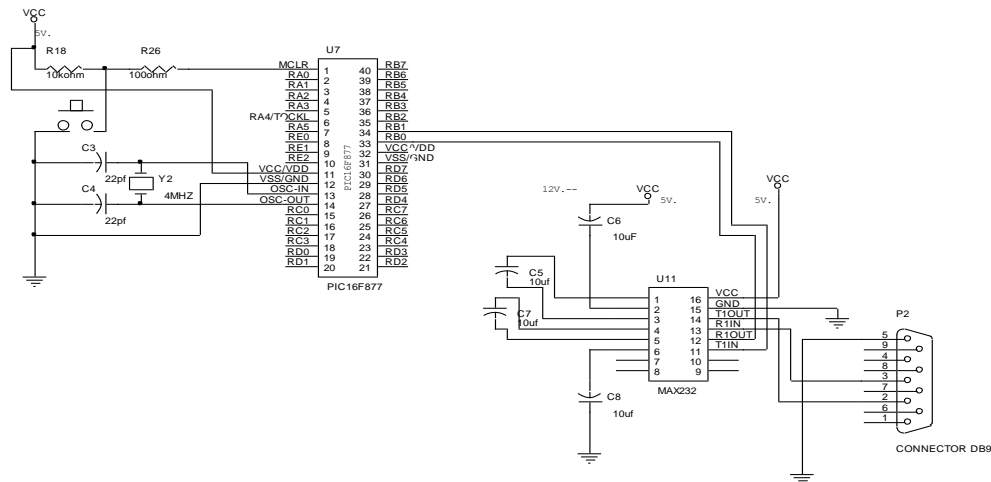
Objetivos:

- ☆ Comprobar el funcionamiento del programa en micro-code-studio.
- ☆ Analizar una vez más la comunicación half-duplex.

Materiales:

Cantidad	Descripción	Características
1	Grabador de Pics	Puerto Serial
1	Pc	WindowsXP
1	Pic16f877	Microchip
1	Micro-Code-Studio	Lenguaje de Programa
2	Resistencias	100ohms, y 10K
1	Cristal	4Mhz
1	LCD	2*16 líneas
1	Fuente	5V DC
1	Circuito Integrado	Max232
4	Capacitares	10uf

Esquema:



Desarrollo:

El CI max232 da solución al problema de transmisión a larga distancia ya que incrementa los niveles de voltaje mediante un juego de capacitores que doblan el voltaje el funcionamiento de este programa es que al momento de ingresar una letra desde la PC al Pic este automáticamente empezará a contar en forma indefinida, la programación en micro-code-studio es:

```
include "modedefs.bas" ;incluyen los modos de comunicación
led var portb.7         ;nombre led al puerto b7
dat var byte            ;variable de almacenamiento de 255
num var byte            ;variable para almacenar el contador
num=0                   ;contenido inicial para la variable num
gosub ledr              ;ir a leds para saber si ya arranco el pic
Inicio:
  serin portb.0,T2400,dat ;esperar el dato y guardarlo en dat
  if dat= "C" then contar ;si dat es empezar contar
  gosub ledr
  goto inicio           ;ir a esperar el siguiente caracter
contar:
  high led              ;led solo encendido indica enviando datos
  serout portb.1,T2400,[#num,"-"] ;enviar el contenido decimal de la variable num
  num= num+1            ;incrementar la variable 1*1
  pause 1000            ;espera de 1S
  Goto contar           ;ir a subrutina contar
ledr:
  high led
  pause 200
  low led
  return
end
```


Conclusiones:

- ✧ Al momento de enviar un dato determinado al Pic este automáticamente empieza a contar.
- ✧ Si se ingresa un dato que no este dentro de los parámetros de la programación el Pic no realizará ningún trabajo.

Recomendaciones:

- ✧ Tener muy presente la numeración del puerto serial DB9 al momento de conectarlo.
- ✧ Verificar cuantas veces sea necesaria la conexión de la práctica.

5.17 Preparatorio de la práctica #9

1. ¿Qué es conversión analógica digital?
.....
2. ¿Qué es un sensor de temperatura?
.....
3. Data sheet del CI LM35
.....
4. Data sheet del LM741 amplificador operacional
.....

5.18 Práctica # 9

Tema: Termómetro Digital:

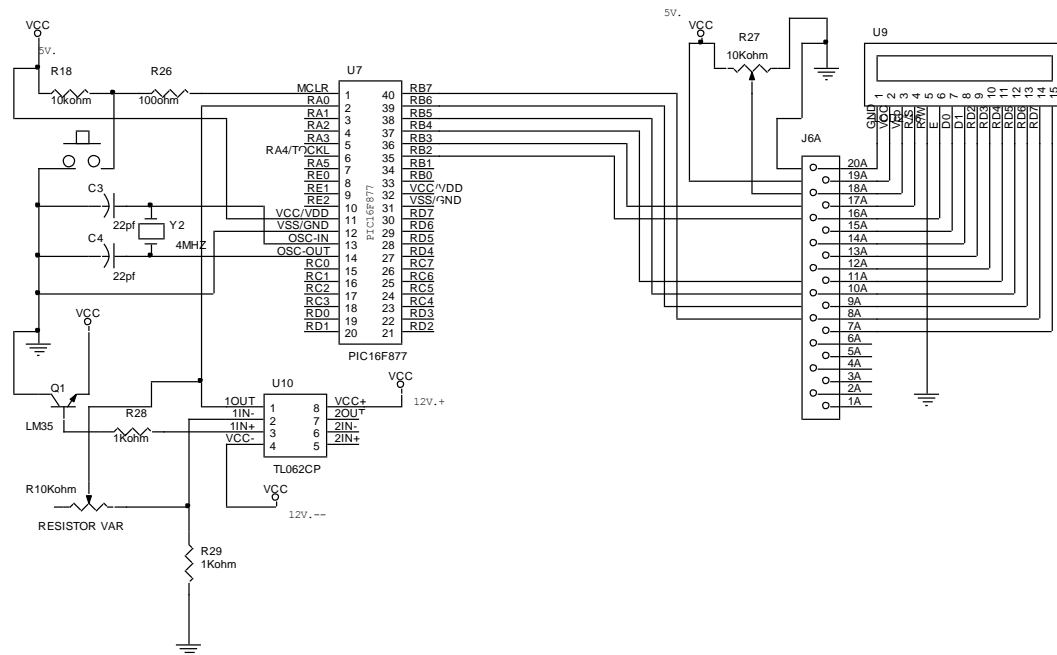
Objetivos:

- ☆ Determinar el valor de la temperatura ambiente
- ☆ Censar la temperatura en diferentes circunstancias

Materiales:

Cantidad	Descripción	Características
1	Grabador de Pics	Puerto Serial
1	Pc	WindowsXP
1	Pic16f877	Microchip
1	Micro-Code-Studio	Lenguaje de Programa
2	Resistencias	100ohms, y 10K
1	Cristal	4Mhz
1	LCD	2*16 lineas
1	Fuente	5V DC
1	Circuito Integrado	LM35
4	Capacitares	10uf

Esquema:



Desarrollo:

En esta práctica utilizaremos el LM35 este dispositivo presenta un Pin OUT con una variación de 10mV por grado centígrado su alimentación puede ser de 4 a 30Voltios y rango de temperatura es de -55°C hasta 150°C debido a que su señal de salida es muy débil la vamos a duplicar utilizando un juego de amplificadores el cual esta configurado como amplificador no inversor su programación es la siguiente:

```

pause 200
DEFINE LCD_DREG PORTB
DEFINE LCD_DBIT 4
DEFINE LCD_RSREG PORTB
DEFINE LCD_RSBIT 3
DEFINE LCD_EREG PORTB
DEFINE LCD_EBIT 2
DEFINE LCD_BITS 4
DEFINE LCD_LINES 2

```

```

define ADC_BITS 8           ;fije número de bits del resultado (5,8,10)
define ADC_CLOCK 3          ;fije El CLOCK(re= 3)
define ADC_SAMPLEUS 50      ;fije el tiempo de muestra en Us
                             ;ADC_SAMPLEUS es el número de microsegundos que el programa
                             ;espera
                             ;entre fijar el canal y comenzar la conversión analógica/digital
TRISA= %1                   ;el puerto A es de entrada

```

ADCON1= %00001110 ;el puerto A.0 es conversor, los demás digitales

```
datos var byte      ;crear variable dato para guardar
temp var byte       ;crear variable dato para guardar
dectemp varbyte     ;crear variable dato para guardar
pause 500           ;esperar 0,5 seg.
```

inicio:

```
adcin 0,datos        ;leer el canal 0 y guardarlo en datos
temp= (datos*50)/255
lcdout $fe,1,"temperatura="
lcdout $fe,$c7,dec temp ;deplegar el valor de datos en decimal
pause 300
goto inicio          ;volver a medir el conversor A/D
end
```

Conclusiones:

- ☆ Podemos observar la variación de temperatura al momento de someterlo a varios cambios climatológicos
- ☆ El termómetro digital es muy efectivo al momento de medir temperatura.

Recomendaciones:

- ☆ Observar que los voltajes de salida se encuentren entre los rangos aceptables tanto para el Pic como para el sensor de temperatura.
- ☆ Tener muy bien definida las Entradas de alimentación para el amplificador operacional

CAPÍTULO VI

CONCLUSIONES Y RECOMENDACIONES

6.1 CONCLUSIONES:

- ✧ De acuerdo a los resultados obtenidos en la fase de pruebas y programación, es posible confirmar el buen funcionamiento del programador, para ambos tipos de familias de PICs, por lo que se recomienda su uso en el desarrollo de cualquier proyecto.
- ✧ El diseño de los circuitos impresos están hechos exclusivamente con fines didácticos para que el estudiante pueda realizar diversos proyectos.
- ✧ Los PIC nos permiten programar y borrar sus aplicaciones cuantas veces sean.
- ✧ La aplicación de los PIC es muy extensa ya que podemos hacer cualquier tipo de proyecto.
- ✧ El funcionamiento del LCD nos permitió observar gráficamente los valores exactos del conversor A/D el cual se encuentra configurado junto al sensor de temperatura LM35.
- ✧ El circuito integrado max232 nos permite tener un alcance de comunicación de datos mayor a 10 metros.

6.2 RECOMENDACIONES:

- ☆ Con el objetivo de que el dispositivo que maneja al puerto serial DB9, no sufra daños, cuando se disponga programar algún Pic, el dispositivo automáticamente detecta errores en la programación.
- ☆ Tener muy en cuenta la polarización de los PIC al alimentarlos con voltaje.
- ☆ Asegúrese que cada proyecto a ejecutar se encuentre muy bien identificado al declarar las sentencias.
- ☆ Identifique la utilización de los puertos en cada práctica.
- ☆ Utilice la preparatoria de cada práctica para hacer más fácil la programación.
- ☆ Tenga mucha precaución al momento de configurar el amplificador operacional ya que funciona a 12 voltios uno positivo y otro negativo.
- ☆ Utilice una fuente de alimentación regulada la misma que deberá contar con un fusible de seguridad en caso de algún corto circuito.

BIBLIOGRAFIA

Libros:

- ☆ REYES, CARLOS A., Aprenda rápidamente a programar Microcontroladores PIC, edición 2004, GRÁFICAS AYERVE C.A., Quito, EC.

Sitios Web:

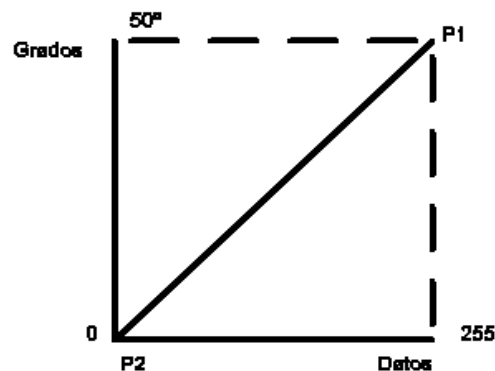
- ☆ <http://www.ic-prog.com> descarga de programa lc-prog105.zip.
- ☆ <http://www.mecanique.co.uk> descarga de programas Microcode y Pbp demo.
- ☆ <http://www.monografias.com> todo tipo de temas relacionados al PIC.

ANEXOS

Anexo 1

Conversor Analógico Digital

Milivoltios	Voltaje	Temperatura	Bits
0mv	0v	0°	0bits
250mv	2,5v	25°	127,5bits
500mv	5v	50°	255bits



Datos:

$$X2=255$$

$$X1=0$$

$$Y2=50$$

$$Y1=0$$

$$T=m \text{ datos}$$

$$m= Y2-Y1/X2-X1$$

$$m= 50-0/255-0$$

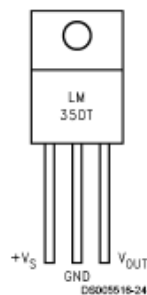
$$m= 50/255$$

$$T= 50*\text{datos}/255$$

$$T= 50*100/255$$

$$T=10,6^{\circ}\text{C}$$

TO-220
Plastic Package*



El LM35 es un sensor de temperatura con una precisión calibrada de 1°C y un rango que abarca desde -55° a $+150^{\circ}\text{C}$.

El sensor se presenta en diferentes encapsulados pero el más común es el to-92 de igual forma que un típico transistor con 3 patas, dos de ellas para alimentarlo y la tercera nos entrega un valor de tensión proporcional a la temperatura medida por el dispositivo. Con el LM35 sobre la mesa las patillas hacia nosotros y las letras del encapsulado hacia arriba tenemos que de izquierda a derecha los pines son: VCC - Vout - GND.

La salida es lineal y equivale a $10\text{mV}/^{\circ}\text{C}$ por lo tanto: $+1500\text{mV} = 150^{\circ}\text{C}$

- $+250\text{mV} = 25^{\circ}\text{C}$
- $-550\text{mV} = -55^{\circ}\text{C}$

Funcionamiento: Para hacernos un termómetro lo único que necesitamos es un voltímetro bien calibrado y en la escala correcta para que nos muestre el voltaje equivalente a temperatura. El LM35 funciona en el rango de alimentación comprendido entre 4 y 30 voltios.

Podemos conectarlo a un conversor Analógico/Digital y tratar la medida digitalmente, almacenarla o procesarla con un microcontrolador o similar.

Usos: El sensor de temperatura puede usarse para compensar un dispositivo de medida sensible a la temperatura ambiente, refrigerar partes delicadas del robot o bien para loggear temperaturas en el transcurso de un trayecto de exploración.

Configuración de un Amplificador no Inversor

Data sheet del amplificador TL062CP:

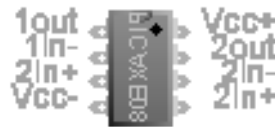
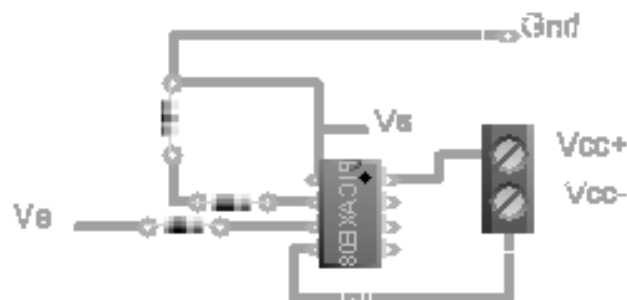


Diagrama:



Datos:

$$R1 = 1\text{K}$$

$$R2 = \text{¿}$$

$$1 + R2/R1 = 10$$

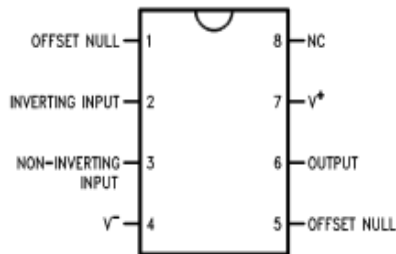
$$R1 + R2 = 10R1$$

$$R2 = 9R1$$

$$Vs = (1 + R2/R1)Ve$$

$$V_s = (1 + 3,9K/2,2K)2V$$

$$V_s = 5,55V$$



El Amplificador operacional LM741

La Descripción general

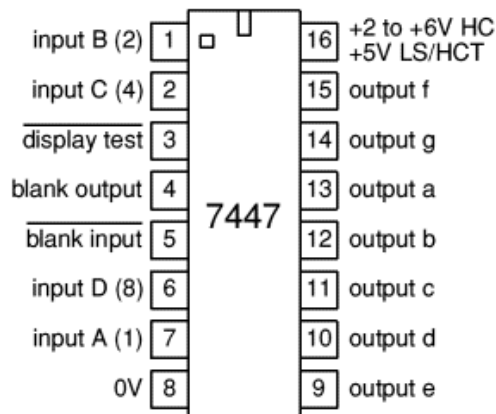
La serie de LM741 es el propósito general amplificadores operacionales que el rasgo mejoró la actuación encima de las normas de industria como el LM709. Ellos son directos, tapón-en los reemplazos para los 709C, LM201, MC1439 y

748 en la mayoría de las aplicaciones.

Los amplificadores ofrecen muchos rasgos que hacen su aplicación casi seguro: cargue excesivamente protección en la entrada y rendimiento, ningún pestillo-a cuando el rango del modo común se excede, así como la libertad de las oscilaciones.

El LM741C es idéntico al LM741/LM741A sólo que el LM741C tiene su actuación garantizada encima de un 0°C a +70°C rango de temperatura, en lugar de -55°C a +125°C.

Descripción general del CI74ls47

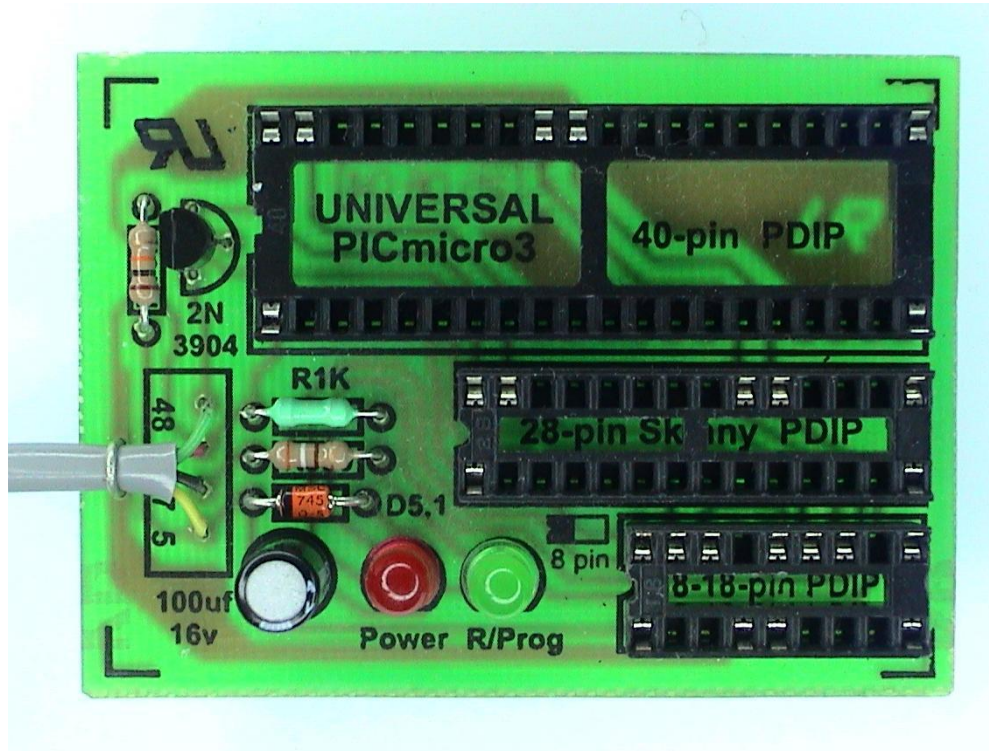


El DM74LS47 acepta cuatro líneas de BCD (8421) la entrada los datos, genera sus complementos internamente y descifra los datos con siete Y/O verjas que tienen, el abrir-coleccionista los rendimientos para manejar el indicador segmentan directamente. Cada segmento el rendimiento se garantiza penetrar 24 MA el EN (BAJO) el estado y resiste 15V en el FUERA DE (ALTO) el estado con un máximo el goteo actual de 250 MA. Las entradas auxiliares proporcionaron borrando, la prueba de la lámpara y funciones de cero-supresión de cascable.

Los 7447 quieren BCD (el decimal codificado binario) qué se entra los valores 0 a 9 (0000 a 1001 en el binario). las Entradas de 10 a 15 (1010 a 1111 en el binario) encenderá los segmentos del despliegue impares pero no hará el daño.

Anexo 2

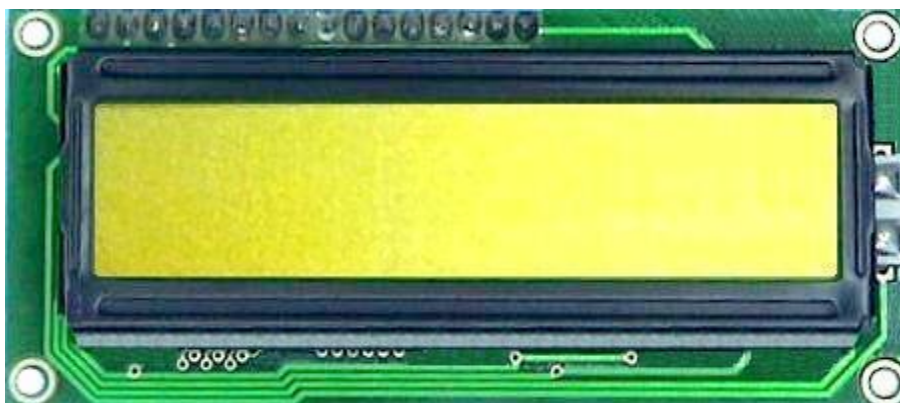
Grabador de Pics:



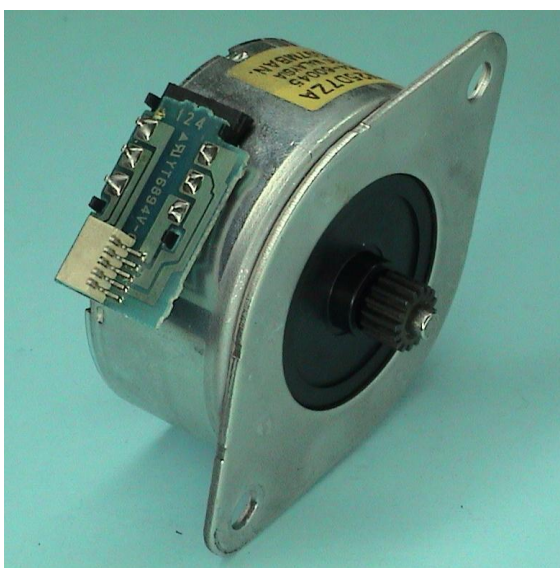
Teclado Hexadecimal:



Pantalla LCD 2*16:



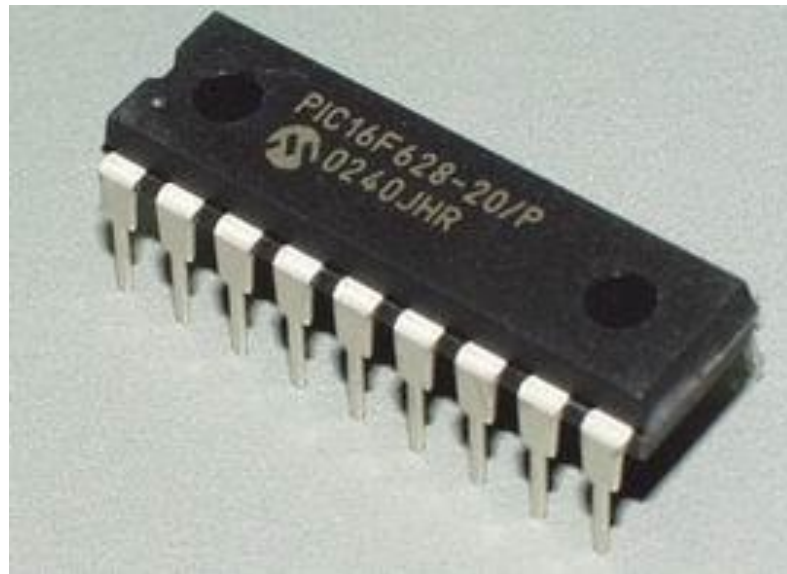
Motor de paso:



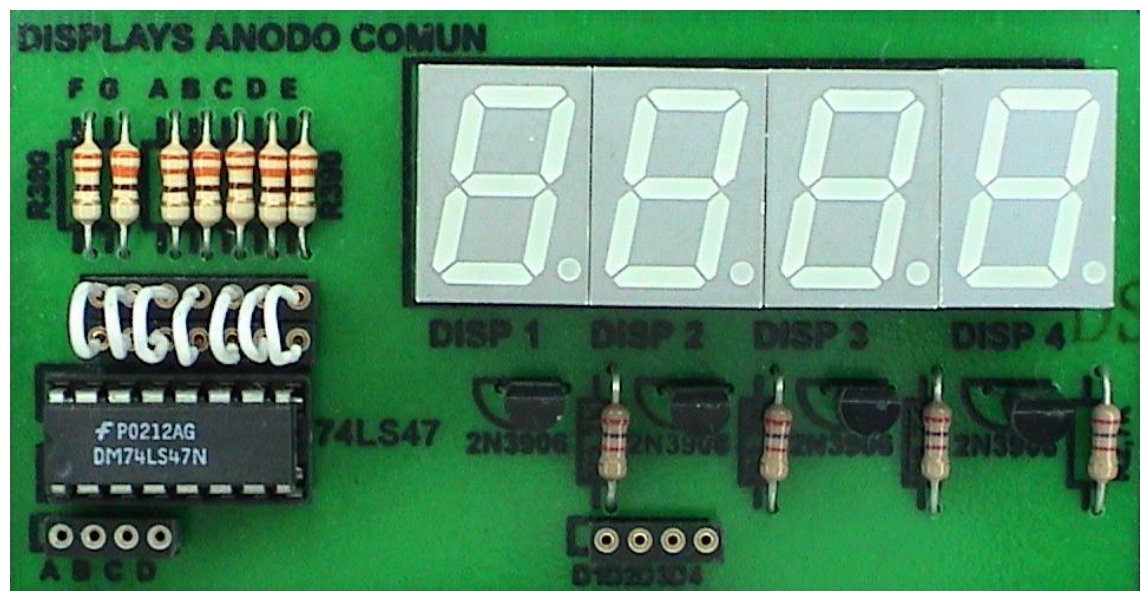
Funcionamiento del LCD:



PIC:



Displays de 7 segmentos:



Anexo 3



Área de la Energía, las Industrias y los Recursos Naturales no Renovables

Proyecto de Tesis

Carrera de tecnología en electrónica

I. TEMA:

“Implementación del Laboratorio de Sistemas Microprocesados”

AUTORES:

Cristian Ulpiano Ortega Guamán
Jorge Fabián Quezada Maldonado
Franklin Eduardo Villalta Alvarado

DIRECTOR SUGERIDO:

Ing. Diego Vinicio Orellana Villavicencio

LOJA – Ecuador

2009

II. PLANTEAMIENTO DEL PROBLEMA

Uno de los grandes sueños del hombre ha sido siempre mejorar su calidad de vida en todos los aspectos, como la explotación de los recursos ya sean estos renovables o no, pero para lograr esto requiere de tecnología para aprovechar al máximo estos recursos, es por eso que en el siglo XVIII se abre el camino a la revolución industrial y al comercio que alcanzan un desarrollo prodigioso.

Los avances tecnológicos juegan un papel cada vez más importante ya que da solución muchos problemas, uno de estos avances es el surgimiento de la electrónica

El computador tiene sus inicios en el siglo XVII en donde Blaise Pascal crea una máquina que permitía realizar sumas y restas.

En 1950 se crea la primera generación de las computadoras en 1960 la segunda generación que funcionaba a base de transistores, en 1965 aparecen las computadoras con circuitos integrados que es la tercera generación, luego en la cuarta generación funcionaban a base de microchips.

*Los Circuitos Integrados ("IC= Integrated Circuits"), al principio de la década de los sesenta se denominó "integración en pequeña escala ("SSI= Small Scale Integration") estando compuesto de unas pocas compuertas lógicas. Después, sobrevino la Integración a Mediana Escala ("MSI= Médium Scale Integration") con un circuito contador Integrado o posiblemente un registro. Los refinamientos y cambios modernos en la tecnología han dado como resultado los componentes denominados de Integración a Gran Escala ("LSI= Large Scale Integration") que tienen bajos costos y que incorporan alrededor de 10.000 transistores en una sola pastilla de silicio, cota necesaria para incluir en un chip a todos los elementos que conforman la Unidad Central de Proceso ("CPU= Central Processor Unit"), dicho circuito integrado recibe el nombre de microprocesador y dio origen a un computador pequeño y de bajo costo denominado

microcomputador. Posteriormente a finales de la década de los setenta se emprendió la integración a gran escala ("VLSI= Very Large Scale Integration") que han incorporado un millón de elementos en una sola pastilla ("chip") de silicio, consiguiéndose integrar todos los elementos que componen un microcomputador.*

El microprocesador es la culminación del desarrollo que ha tenido lugar en la tecnología de los semiconductores, desde que se produjo el primer transistor en los primeros años de la década de los cincuenta.

El PIC es el microcontrolador que fabrica la compañía microchip, aunque no son los microcontroladores que más prestaciones ofrecen, en los últimos años han ganado mucho mercado, debido al bajo precio de éstos, lo sencillo de su manejo y programación, un PIC al ser un microcontrolador programable puede llevar a cabo cualquier tarea para la cual haya sido programado además son materiales fáciles de conseguir ya que actualmente en la industria han permitido la automatización de muchas máquinas permitiendo de esta manera la agilitación de sus productos y aumentando considerablemente sus ganancias y obteniendo mejor calidad de producción, no obstante debemos ser conscientes de las limitaciones de cada PIC.

III. MARCO TEÓRICO

INTRODUCCION

El microprocesador es uno de los logros más sobresalientes del siglo XX. Esas son palabras atrevidas, y hace un cuarto de siglo tal afirmación habría parecido absurda. Pero cada año, el microprocesador se acerca más al centro de nuestras vidas, forjándose un sitio en el núcleo de una máquina tras otra. Su presencia ha comenzado a cambiar la forma en que percibimos el mundo e incluso a nosotros mismos. Cada vez se hace más difícil pasar por alto el

microprocesador como otro simple producto en una larga línea de innovaciones tecnológicas.

Ninguna otra invención en la historia se ha diseminado tan aprisa por todo el mundo o ha tocado tan profundamente tantos aspectos de la existencia humana. Hoy existen casi 15,000 millones de microchips de alguna clase en uso (el equivalente de dos computadoras poderosas para cada hombre, mujer y niño del planeta). De cara a esa realidad, ¿quién puede dudar que el microprocesador no sólo está transformando los productos que usamos, sino también nuestra forma de vivir y, por último, la forma en que percibimos la realidad?

No obstante que reconocemos la penetración del microprocesador en nuestras vidas, ya estamos creciendo indiferentes a la presencia de esos miles de máquinas diminutas que nos encontramos sin saberlo todos los días. Así que, antes de que se integre de manera demasiado imperceptible en nuestra diaria existencia, es el momento de celebrar al microprocesador y la revolución que ha originado, para apreciar el milagro que es en realidad cada uno de esos chips de silicio diminutos y meditar acerca de su significado para nuestras vidas y las de nuestros descendientes.

1. HISTORIA DE LOS MICROPROCESADORES

Han pasado más de 25 años desde que Intel diseñara el primer microprocesador, siendo la compañía pionera en el campo de la fabricación de estos productos, y que actualmente cuenta con más del 90 por ciento del mercado. Un tiempo en el que todo ha cambiado enormemente, y en el que desde aquel 4004 hasta el actual Pentium II hemos visto pasar varias generaciones de máquinas que nos han entretenido y nos han ayudado en el trabajo diario.

Dicen que es natural en el ser humano querer mirar constantemente hacia el futuro, buscando información de hacia dónde vamos, en lugar de en dónde hemos estado. Por ello, no podemos menos que asombrarnos de las previsiones

que los científicos barajan para dentro de unos quince años. Según el Dr. Albert Yu, vicepresidente de Intel y responsable del desarrollo de los procesadores desde el año 1984, para el año 2011 utilizaremos procesadores cuyo reloj irá a una velocidad de 10 GHz (10.000 MHz), contendrán mil millones de transistores y será capaz de procesar cerca de 100 mil millones de instrucciones por segundo. Un futuro prometedor, que permitirá realizar tareas nunca antes pensadas.

♦ Los inicios

Sin embargo, para que esto llegue, la historia de los procesadores ha pasado por diferentes situaciones, siguiendo la lógica evolución de este mundo. Desde aquel primer procesador 4004 del año 1971 hasta el actual Pentium II del presente año ha llovido mucho en el campo de los procesadores. Tanto, que no estamos seguros si las cifras que se barajan en Intel se pueden, incluso, quedar cortas. Aquel primer procesador 4004, presentado en el mercado el día 15 de noviembre de 1971, poseía unas características únicas para su tiempo. Para empezar, la velocidad de reloj sobrepasaba por poco los 100 KHz (sí, habéis leído bien, kilohertzios), disponía de un ancho de bus de 4 bits y podía manejar un máximo de 640 bytes de memoria. Realmente una auténtica joya que para entonces podía realizar gran cantidad de tareas, pero que por desgracia no tiene punto de comparación con los actuales micros. Entre sus aplicaciones, podemos destacar su presencia en la calculadora Busicom, así como dotar de los primeros tintes de inteligencia a objetos inanimados.

Poco tiempo después, sin embargo, el 1 de abril de 1972, Intel anunciaba una versión mejorada de su procesador. Se trataba del 8008, que contaba como principal novedad con un bus de 8 bits, y la memoria direccionable se ampliaba a los 16 Kb. Además, llegaba a la cifra de los 3500 transistores, casi el doble que su predecesor, y se le puede considerar como el antecedente del procesador que serviría de corazón al primer ordenador personal. Justo dos años después, Intel anunciaba ese tan esperado primer ordenador personal, de nombre Altair, cuyo nombre proviene de un destino de la nave Enterprise en uno de los

capítulos de la popular serie de televisión Star Trek la semana en la que se creó el ordenador. Este ordenador tenía un coste de entorno a los 400 dólares de la época, y el procesador suponía multiplicar por 10 el rendimiento del anterior, gracias a sus 2 MHz de velocidad (por primera vez se utiliza esta medida), con una memoria de 64 Kb. En unos meses, logró vender decenas de miles de unidades, en lo que suponía la aparición del primer ordenador que la gente podía comprar, y no ya simplemente utilizar.

♦ Tipos de conexión

El rendimiento que dan los microprocesadores no sólo dependen de ellos mismos, sino de la placa donde se instalan. Los diferentes micros no se conectan de igual manera a las placas:

- **Socket**, con mecanismo ZIF (Zero Insertion Force). En ellas el procesador se inserta y se retire sin necesidad de ejercer alguna presión sobre él. Al levantar la palanquita que hay al lado se libera el microprocesador, siendo extremadamente sencilla su extracción. Estos zócalos aseguran la actualización del microprocesador. Hay de diferentes tipos:
- **Slot A / Slot 1 /Slot 2**. Es donde se conectan respectivamente los procesadores Athlon antiguos de AMD / los procesadores Pentium II y antiguos Pentium III / los procesadores Xeon de Intel dedicados a servidores de red. Todos ellos son cada vez más obsoletos. El modo de insertarlos es a similar a una tarjeta gráfica o de sonido, ayudándonos de dos guías de plástico insertadas en la placa base.
- En las placas base más antiguas, el micro iba soldado, de forma que no podía actualizarse (486 a 50 MHz hacia atrás). Hoy día esto no se ve en lo referente a los microprocesadores de PC.

LA UNIDAD DE PROCESAMIENTO CENTRAL

La unidad de procesamiento central (CPU) es donde se manipulan los datos. En una microcomputadora, el CPU completo está contenido en un chip muy pequeño llamado *microprocesador*. Todas las CPU tienen por lo menos dos

partes básicas, la *unidad de control* y la *unidad aritméticológica*. Todos los recursos de la computadora son administrados desde la **unidad de control**, cuya función es coordinar todas las actividades de la computadora. La unidad de control contiene las instrucciones de la CPU para llevar a cabo comandos. El **conjunto de instrucciones**, que está incluido dentro de los circuitos de la unidad de control, es una lista de todas las operaciones que realiza la CPU. Cada instrucción en el conjunto de instrucciones es acompañado por un **microcódigo**, que son instrucciones muy básicas que le dicen a la CPU cómo ejecutar las instrucciones. Cuando la computadora corre un programa, busca los comandos del programa dentro del conjunto de instrucciones de la CPU y las ejecuta en orden. La *unidad aritmético-lógico (ALU)* es cuando la unidad de control encuentra una instrucción que involucra aritmética o lógica, le pasa el control al segundo componente de la CPU. La ALU incluye un grupo de registros, es decir, memoria construida directamente en la CPU que se usa para guardar datos que están siendo procesados por la instrucción actual.

El microprocesador moderno contiene unos 20 millones de transistores y cada chip terminado es el producto de procesos más complicados que los que se utilizaron en el Proyecto Manhattan para construir la bomba atómica. Y no obstante, pese a un proceso de manufactura extraordinariamente refinado, los microchips se producen en volumen a razón de más de 1,000 millones de unidades por año. Para poner esta complejidad en perspectiva, imagínese que dentro de cada microprocesador diminuto existe una estructura tan compleja como una ciudad de tamaño mediano, incluidas todas sus líneas de energía eléctrica, líneas telefónicas, líneas de drenaje, edificios, calles y casas. Ahora imagine que en esa misma ciudad, millones de personas se desplazan a la velocidad de la luz y con la sincronización perfecta en una danza de coreografía muy complicada.

Y eso es tan sólo un chip. De todas las estadísticas asombrosas que se utilizan para describir el mundo del microprocesador, ninguna es más extraordinaria que ésta: el número total de transistores que integran todos los microchips que se

producirán en el mundo este año es equivalente al número de gotas de lluvia que caerán en California durante ese mismo periodo.

AVANCE TECNOLÓGICO DE LOS PROCESADORES.

♦ 8088 / 8086

En 1975 Intel decidió ponerse manos a la obra para construir su primer micro de 16 bits que salió al mercado en 1978. Se trataba del 8086, que definió el inicio de su gama de productos más famosa, la familia de microprocesadores x86.

La longitud de los registros del 8086 era de 16 bits (de ahí su denominación de 16 bits), había versiones que funcionaban a 4.77 y 8 MHz, tenía un bus de datos de 16 bits y un bus de direcciones de 20 bits, lo que le permitía acceder a un máximo de memoria de 1 Mb según el cálculo 2^{20} . Seguidamente, en 1979, Intel sacó el 8088, que en contra de lo que podamos pensar no es mejor que el 8086. La diferencia era sustancial; el bus de datos era de 8 bits (la mitad). Este paso hacia atrás estuvo provocado por el estado de la industria de la época. Utilizar un bus de datos de 16 bits suponía forzar al mercado a desarrollar para 16 bits lo que implicaba un incremento en los costes de desarrollo de controladores de periféricos y memorias. El síntoma fue que Intel se había adelantado a su época.

Los japoneses, aprovechando la ocasión y dando fe a su fama de copiones fabricaron unos clónicos de los 8086 y 8088. Fue NEC la que, por procedimientos de ingeniería inversa, sacó el diseño de estos micros y creó sus modelos V20 y V30, que incrementaban el rendimiento respecto a los de Intel en un 10-30 por ciento. Como os podréis imaginar Intel demandó a NEC, pero perdió el juicio. La Justicia determinó que el microcódigo del chip podía registrarse pero Intel no había marcado en el chip el símbolo del copyright, con lo cual perdía los derechos de copia.

Se encuentra un error de diseño en el 8088 que afecta a la gestión de interrupciones y al registro de pila SS (*Stack Segment*).

Intel fabricó variaciones de estos modelos, sacando al mercado los 80C86, 80C88, 80186 y 80188 cuyas modificaciones fueron el ahorro de energía en las versiones C para su instalación en portátiles, y el rediseño interno para la optimización en las versiones 1.

♦ 80286

En 1984 aparece el 80286 como base para una nueva generación de ordenadores de IBM, el IBM AT (*Advanced Technology*). Supone un nuevo salto tecnológico. Además de incrementar el bus de direcciones de 20 bits a 24, lo que permitía acceder hasta los 16 Mb de RAM, se incrementaba la velocidad, llegando a ser hasta un 25 por ciento más rápidos que los 8086 y 8088 originales.

La novedad más importante que se introdujo fue la gestión de memoria virtual. La memoria virtual es una extensión de memoria en disco (o dispositivo de almacenamiento secundario) añadida a la memoria física instalada. Así, el 80286 es capaz de tratar hasta un total de 1 Gb, desglosado en 16 Mb de memoria física más 1008 Mb de memoria virtual. La memoria virtual solamente la utilizan los programas que están preparados para ello.

Aparece también un nuevo modo de operación del microprocesador. Aparte del modo real (el normal de operación) que direcciona hasta 1 Mb de memoria física y asegura la compatibilidad para aplicaciones diseñadas por los 8086/8088, se tiene el modo protegido que no es compatible con estos programas desarrollados para los micros antes mencionados. El modo protegido es el que permite acceder a los 1.008 Mb de memoria virtual.

El 80286 trabaja en su arranque en modo real. El cambio a modo protegido, lo que se conoce técnicamente como upshift, no es reversible (downshift), siendo necesario hacer un reset del microprocesador para volver al modo real; sin duda un gran fallo de diseño.

El 80286 se presentó con velocidades de reloj de 2, 8, 10, 12, 16 y 20 MHz.

♦ 80386 DX y SX

Introducido en 1985, el 80386 DX supera un nuevo escalón en el avance tecnológico en microprocesadores. Se incorpora una nueva ampliación y surge el número mágico, el 32. Los buses de datos y de direcciones se amplían hasta 32 líneas de datos, ocurriendo lo mismo con el tamaño de los registros. Esta ampliación supone un incremento en la memoria RAM física instalada. Puede direccionar 4 GB de memoria física (DX significa *Double word eXternal*) y 64 Tb de memoria virtual, una cifra que en la actualidad está aún muy por encima de las posibilidades económicas de los usuarios (a ver quién instala 4.000 Megabytes de RAM, unos 20 millones de pesetas).

Arranca en modo real, al igual que el 80286, e incorpora un nuevo modo de operación: el modo real virtual del 8086, que permite tener varias sesiones 8086 trabajando simultáneamente simulando una especie de pseudomultitarea.

En los microprocesadores anteriores la gestión de memoria se realizaba en segmentos de 16 Kb. Con el 80286 este tamaño de los segmentos de la memoria se hacían muy pequeños y el programador tenía que trabajar más para adaptarse a una gran cantidad de segmentos. El 80386 permite la definición de segmentos de memoria de tamaño variable. Aparte, Intel corrigió la deficiencia del downshifting, pudiéndose realizar por software.

Otra de las innovaciones en la inclusión de una memoria cache interna en el chip destinado a almacenar instrucciones provenientes de memoria sin necesidad de que la unidad de ejecución intervenga. Intel comete un nuevo error en el diseño del micro que genera inexactitudes en el cálculo de 32 bits, que se presentan en los micros lanzados al mercado hasta mayo del 1987. Los modelos corregidos van etiquetados con una doble sigma mayúscula o con el identificativo DX. Este error afectaba a las operaciones de multiplicación de 32 bits. Ocurría bajo las siguientes circunstancias:

Se detecta un segundo *bug* denominado POPAD bug. Su efecto es el vaciado del registro acumulador EAX cuando se ejecuta una instrucción de acceso a memoria inmediatamente después de la ejecución de la instrucción POPAD.

Aparecen variaciones que afectan al consumo de energía pensadas para portátiles, se trataba de los 80386SL (Slow Low power) y 80386SLC (Slow Low power Cache), que es propiedad de IBM aunque lo fabrique Intel. Las frecuencias de funcionamiento eran de 12, 20, 25 y 33 MHz.

El 80386SX (SX significa *Simple word eXternal*) tiene las mismas características que el 80386DX, salvo que el bus de direcciones externo se reduce a 16 bits. Introducido en 1988 daba la potencia de un 80386 a precio de un 80286. Durante mucho tiempo se rumoreó que el P9 podría ser compatible con los zócalos 80286, pero al final no fue así. La razón es que el 80286 multiplexa todos sus buses para conseguir con menos líneas el mismo resultado (menor coste) El 80386SX sólo multiplexa el bus de direcciones. Las frecuencias de funcionamiento eran de 16, 20, 25 y 33 MHz.

◆ 80486DX 80486SX

El 80486DX salió al mercado en 1989. La estructura interna hablando en términos numéricos es igual a la de un 80386. El tamaño de los registros y de los buses son de 32 bits. Mantiene los tres modos de operación: real, protegido y real virtual. Las diferencias reales con los 80386DX son que tiene un flag más, un estado de excepción más, 2 bits más en la tabla de entrada de páginas, 6 instrucciones y los registros de control tienen una longitud de 9 bits.

Se realizan también cambios en la arquitectura interna. Se crea un mayor número de líneas hardware lo que implica un incremento en la velocidad. Se imponen reglas de diseño más estrictas, lo que supone una reducción del tamaño del chip. Al reducirse el tamaño se reduce también el consumo y consiguientemente la temperatura que alcanza el chip, con lo cual lo podemos hacer funcionar a un mayor número de ciclos de reloj, lo que supondrá la aparición de los Overdrives.

Se incluye un coprocesador matemático interno que dobla las prestaciones de un 80387 trabajando a la misma velocidad. Se logra un diseño mejor y la comunicación entre el chip principal y el coprocesador matemático es interna, lo que mejora la velocidad en las transferencias y unas sincronizaciones más estrechas.

La memoria cache (8 Kb) del microprocesador está dividida en 4 caches de 2 Kb cada una. Esto agiliza la ejecución de algunas aplicaciones. Si se asigna una memoria cache secundaria (L2) el rendimiento del micro puede aumentar hasta un 30 por ciento más.

El 80486SX es igual que un 80486DX, sólo que el coprocesador matemático está inhabilitado. El coprocesador matemático 80487SX es en realidad un 80486DX puro que desactiva por completo el 80486SX, sin que podamos retirarlo de la placa. Las velocidades a las que funcionan son de 25, 33, 40 y 50 MHz. Hay versión SL para portátiles.

♦ **80486DX2, 80486DX4, 80486SX2**

Estos modelos de microprocesadores en realidad son iguales que sus hermanos menores. Internamente duplican la velocidad del reloj del sistema. Es igual que revolucionar el motor de un coche para que corra más. Las consecuencias son obvias: un sobrecalentamiento del micro con una reducción de potencia. Por este motivo se recomienda utilizar un método de disipación de calor para que el rendimiento no se vea reducido (laminillas disipadoras o ventiladores). Las velocidades a las que trabajan son: 50, 66, 75 y 100 (sólo para el DX4) MHz para los DX2 y 40 y 50 MHz para los SX2.

Los 80486 tampoco están libres del pecado original. En el 80486 cuando el coprocesador matemático detecta un error de límite del tamaño de segmento, algunas veces la CPU falla al generar la excepción 13. Este error se genera bajo las siguientes condiciones:

Un segundo bug aparece en el cálculo de la instrucción dedicada a calcular arcotangentes.

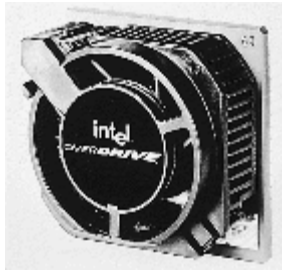
OVERDRIVES

Intel comenzó una nueva política con la salida de los microprocesadores con la denominación Overdrive. Los Overdrive eran actualizaciones para los microprocesadores instalados en los sistemas que dispusieran de un segundo zócalo para tal propósito. En esta primera generación de Overdrives los chips disponían de un duplicador de frecuencia interno y tenían un pin más, el número 169. Este pin se encargaba de inhabilitar el 80486 instalado en la placa dejando como único micro funcionando el Overdrive. No era posible la retirada del micro anterior, puesto que el sistema dejaba de funcionar. La política de actualización era buena, lo que no era tan bueno era la trampa para los usuarios. Si se puede retirar el micro anterior se puede vender a usuarios que necesiten menos potencia, no siendo tan gravosa la inversión de más de 80.000 pesetas que costaban cuando salieron los primeros en 1991.

En la segunda generación de Overdrives se olvidó el pin 169, teniendo los 168 que los hacían compatibles con los zócalos de los 80486. En estos casos la actualización es sencillísima: quitar el que estaba y poner el nuevo en el mismo lugar.

La tercera generación de Overdrives trabaja con un consumo menor para reducir de este modo su alta temperatura. El voltaje se reduce a 3.3 voltios de los 5 que necesitaban los anteriores. Si vas a comprar un DX2 o DX4 fíjate bien qué voltaje utiliza vuestra placa. los Overdrives etiquetados como tal funcionan a 5V, los etiquetados directamente como DX4, sin disipador, funcionan a 3.3V.

♦ Pentium Overdrive



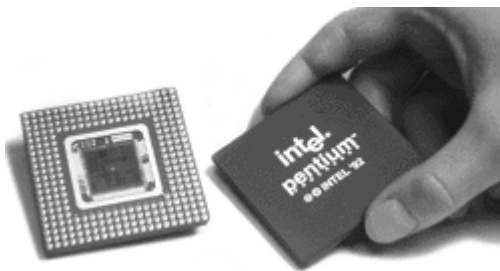
¡Cómo no íbamos a disponer de una versión "light" del Pentium compatible pin a pin con nuestros microprocesadores 80486DX y DX2!

El 18 de septiembre de 1995 Intel anuncia la disponibilidad de un nuevo modelo procesador de mejora Pentium Overdrive a 83 MHz (además del modelo a 63 MHz que ya existía) que permite la actualización de los microprocesadores 80486 DX, DX2 y SX. Compatible pin a pin con estos microprocesadores en Pentium Overdrive integra la tecnología del Pentium en 3,3 voltios y 0'6 micras.

Aparte de ser como un Pentium genuino cabe destacar la presencia de una memoria cache de 32 Kb, un regulador de tensión para reducir la tensión de 5 a 3,3 voltios, disipador y ventilador integrado y una circuitería interna que incrementa en dos veces y media la frecuencia del bus del sistema ($33 \text{ MHz} * 2'5 = 82'5 \text{ MHz}$). El incremento medio en las prestaciones respecto a un 80486 a 66 MHz es de un 50 por ciento aunque en aplicaciones puntuales (AutoCAD 13) puede ser de un 96 por ciento, lo que lo sitúa en las prestaciones de un Pentium genuino a 75 MHz. El precio de venta al público es de unas 40.000 pesetas más IVA.

Antes de comprarlo os recomiendo que llevéis el ordenador a la tienda para que sepan qué zócalo tiene para su actualización.

EL PENTIUM



La quinta generación de microprocesadores Intel tomó el nombre de Pentium. Aparecido en marzo de 1993 en frecuencias de trabajo de 60 y 66 MHz llega a ser cinco veces más potente que un 80486 a 33 MHz. Fabricados con un

proceso BiCMOS de geometría de 8 micras y con una arquitectura superescalar, los microprocesadores Pentium se encuadran en un concepto RISC. Mientras que el 80386 y el 80486 tienen una unidad de ejecución, el Pentium tiene dos, pudiendo ejecutar dos instrucciones por ciclo de reloj con sus correspondientes cálculos, ya que también tiene dos unidades aritmético-lógicas (ALU). El 80386 (CISC) ejecuta una instrucción en varios ciclos de reloj y el 80486 ejecuta una instrucción por ciclo de reloj (en términos medios).

Intel toma como modelo la estructura separada para la memoria cache interna del microprocesador. Consta de dos bloques de 8 Kb, uno para las instrucciones y otro para los datos que funcionan bajo una estructura de asociación de conjuntos bidireccional. Para los extremadamente curiosos el algoritmo de sustitución de datos en la cache es el LRU (*Least Recently Used*, el menos utilizado recientemente).

El coprocesador matemático incluido utiliza algoritmos mejorados y añade instrucciones de suma, multiplicación y división de números en punto flotante integradas en el silicio, además de incorporar un pipeline de 8 niveles para lograr ejecutar operaciones en punto flotante en un solo ciclo de reloj.

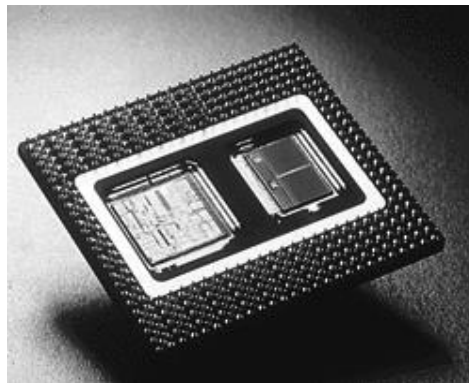
Se integran nuevos avances tecnológicos, además de los ya comentados, como por ejemplo la predicción de ramificaciones, buses de datos internos de 256 bits, bus de datos externo de 64 bits (que soporta transferencias de 258 Kbytes por segundo) y memorias cache de escritura diferida.

La tecnología de bus PCI se presenta junto con el Pentium, que incorporaba una mejor implementación del bus local. Permite tener hasta 10 conectores PCI en un primer nivel, conectados al procesador a través de la circuitería controladora PCI (conexión no directa).

Recientemente han aparecido versiones del Pentium a 75, 90, 100 y 133 MHz, siendo el último, presentado el 23 de octubre, uno a 120 MHz diseñado especialmente para ordenadores portátiles.

♦ Pentium pro y Pentium II

La aparición, el 27 de marzo de 1995, del procesador Pentium Pro supuso para los servidores de red y las estaciones de trabajo un aire nuevo, tal y como ocurriera con el Pentium en el ámbito doméstico. La potencia de este nuevo procesador no tenía comparación hasta entonces, gracias a la arquitectura de 64 bits y el empleo de una tecnología revolucionaria como es la de .32 micras, lo que permitía la inclusión de cinco millones y medio de transistores en su interior. El procesador contaba con un segundo chip en el mismo encapsulado, que se encargaba de mejorar la velocidad de la memoria caché, lo que resultaba en un incremento del rendimiento sustancioso. Las frecuencias de reloj se mantenían como límite por arriba en 200 MHz, partiendo de un mínimo de 150 MHz. Un procesador que en principio no tiene muchos visos de saltar al mercado doméstico, puesto que los procesadores Pentium MMX parecen cubrir de momento todas las necesidades en este campo. No podemos asegurar que en un futuro cercano esto no acabe ocurriendo, pues en el mundo de la informática han sucedido las cosas más extrañas, y nunca se sabe por dónde puede tirar un mercado en constante evolución.



Una evolución que demostró Intel hace muy poco con un nuevo procesador, denominado Pentium II, que viene a ser simplemente un nuevo ingenio que suma las tecnologías del Pentium Pro con el MMX. Como resultado, el Pentium II es el procesador más rápido de cuantos ha comercializado Intel. Por el momento únicamente se dispone de las versiones a 233 y 266 MHz, pero después de este verano podremos disfrutar de la versión de 300 MHz, que

supondrá un nuevo récord de velocidad de reloj. El Pentium II, cuyas características fueron tratadas con detalle en el artículo de portada del pasado mes de la revista, es hoy (por poco tiempo) el extremo de la cadena evolutiva de Intel.

Fabricado en una geometría de 0'6 micras, Intel está realizando sus desarrollos con vistas a reducirla a 0'35 micras como la de los Pentium actuales a 133 MHz, lo que reducirá su temperatura y podrá elevarse la frecuencia de reloj hasta los 200 MHz.

Intel ha puesto mucho esfuerzo en probar el Pentium Pro para intentar salvarse de los numerosos bugs que manchan su gran prestigio. Intel nos ofreció participar en las pruebas de sus Pentium Pro, petición a la cual respondimos afirmativamente, pero al final, la drástica reducción de unidades nos hicieron quedar fuera de los elegidos (sólo se probaron 100 unidades que estarían más que asignadas de las más de 10.000 peticiones que recibió Intel a través de Internet).

El Pentium Pro no es compatible con las placas que existen en el mercado. El motivo principal es la inclusión de la memoria cache secundaria dentro del chip. Se utiliza un bus interno que está optimizado para trabajar con las temporizaciones de conexión directa, lo cual imposibilita la conexión de la memoria cache externa (a mi entender no tiene la suficiente justificación, puesto que a nivel de SETUP la memoria cache secundaria se puede desactivar e incluso anular retirando los integrados de sus zócalos. El tema de la sincronización ya es otro cantar y probablemente sea el motivo real).

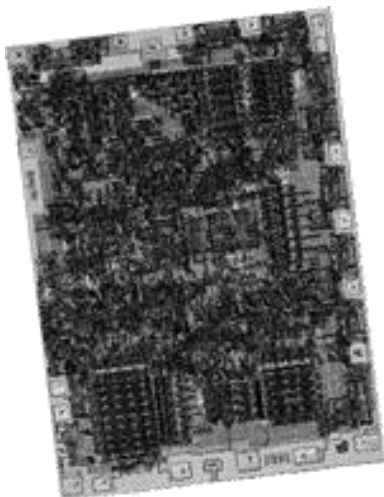
Este nuevo producto tiene un bus que ha sido diseñado para conectar varios Pentium Pro en paralelo que soporta el protocolo MESI, es un microprocesador de 32 bits que incorpora una instrucción más (mover datos condicionalmente) que supone una mayor predicción de ramificaciones en la ejecución. Tiene 21 millones de transistores, 5'5 millones en el núcleo y 15'5 millones en la memoria cache secundaria. La CPU consta de dos chips colocados en cavidades independientes conectadas internamente. El chip correspondiente a la memoria

cache es más pequeño que el del chip del núcleo, ya que la disposición de los transistores permite una mayor concentración.

LA TECNOLOGÍA MMX

Aunque no podamos considerar la tecnología MMX como un procesador en sí mismo, sería injusto no hablar de ella en un informe como éste. Es uno de los mayores pasos que ha dado Intel en la presente década, y según ellos mismos, todos los procesadores que fabriquen a partir de mediados del próximo año llevarán incorporada esta arquitectura. Para su desarrollo se analizaron un amplio rango de programas para determinar el funcionamiento de diferentes tareas: algoritmos de descompresión de vídeo, audio o gráficos, formas de reconocimiento del habla o proceso de imágenes, etc. El análisis dio como resultado que numerosos algoritmos usaban ciclos repetitivos que ocupaban menos del 10% del código del programa, pero que en la práctica suponían el 90% del tiempo de ejecución. De modo que nació la tecnología MMX, compuesta por 57 instrucciones y 4 tipos de datos nuevos, que se encargan de realizar esos trabajos cíclicos consumiendo mucho menos tiempo de ejecución. Antes, para manipular 8 bytes de datos gráficos requería 8 repeticiones de la misma instrucción; ahora, con la nueva tecnología, se puede utilizar una única instrucción aplicada a los 8 bytes simultáneamente, obteniendo de este modo un incremento del rendimiento de 8x.

EL FUTURO DE LOS MICROPROCESADORES



La evolución que están sufriendo los procesadores es algo que no parece escapar a la atención de millones de personas, cuyo trabajo depende de hasta dónde sean capaces de llegar los ingenieros de Intel a la hora de desarrollar nuevos chips. El último paso conocido ha sido la implementación de la nueva arquitectura de 0.25 micras, que viene a

sustituir de forma rotunda la empleada hasta el momento, de 0.35 micras en los últimos modelos de procesador. Esto va a significar varias cosas en un futuro no muy lejano. Para empezar, la velocidad se incrementará una media del 33 por ciento con respecto a la generación de anterior. Es decir, el mismo procesador usando esta nueva tecnología puede ir un 33 por ciento más rápido que el anterior. Para que os podáis hacer una idea del tamaño de esta tecnología, deciros que el valor de 0.25 micras es unas 400 veces más pequeño que un pelo de cualquier persona. Y este tamaño es el que tienen los transistores que componen el procesador. El transistor, como muchos sabréis, permite el paso de la corriente eléctrica, de modo que en función de en qué transistores haya corriente, el ordenador realiza las cosas (esto es una simplificación de la realidad, pero se ajusta a ella más o menos). Dicha corriente eléctrica circula entre dos puntos, de modo que cuanto menor sea esta distancia, más cantidad de veces podrá pasar pues el tiempo de paso es menor. Aunque estamos hablando de millonésimas de segundo, tened en cuenta que un procesador está trabajando continuamente, de modo que ese tiempo que parece insignificante cuando es sumado a lo largo de las miles de millones de instrucciones que realizar, nos puede dar una cantidad de tiempo bastante importante. De modo que la tecnología que se utilice puede dar resultados totalmente distintos incluso utilizando el mismo procesador. Por el momento, en un futuro cercano además de contar con la arquitectura de 0.25 micras, podremos disfrutar de una de 0.07 para el año 2011, lo que supondrá la introducción en el procesador de mil millones de transistores y alcanzando una velocidad de reloj cercana a los 10000 MHz, es decir, 10 GHz.

2. MICROCONTROLADORES

Los microcontroladores están conquistando el mundo. Están presentes en nuestro trabajo, en nuestra casa y en nuestra vida, en general. Se pueden encontrar controlando el funcionamiento de los ratones y teclados de los computadores, en los teléfonos, en los hornos microondas y los televisores de nuestro hogar. Pero la invasión acaba de comenzar y el nacimiento del siglo XXI será testigo de la conquista masiva de estos diminutos computadores, que

gobernarán la mayor parte de los aparatos que fabricaremos y usamos los humanos.

CONTROLADOR Y MICROCONTROLADOR.

Recibe el nombre de controlador el dispositivo que se emplea para el gobierno de uno o varios procesos. Por ejemplo, el controlador que regula el funcionamiento de un horno dispone de un sensor que mide constantemente su temperatura interna y, cuando traspasa los límites prefijados, genera las señales adecuadas que accionan los efectores que intentan llevar el valor de la temperatura dentro del rango estipulado.

Aunque el concepto de controlador ha permanecido invariable a través del tiempo, su implementación física ha variado frecuentemente. Hace tres décadas, los controladores se construían exclusivamente con componentes de lógica discreta, posteriormente se emplearon los microprocesadores, que se rodeaban con chips de memoria y E/S sobre una tarjeta de circuito impreso. En la actualidad, todos los elementos del controlador se han podido incluir en un chip, el cual recibe el nombre de microcontrolador. Realmente consiste en un sencillo pero completo computador contenido en el corazón (chip) de un circuito integrado.

Un microcontrolador es un circuito integrado de alta escala de integración que incorpora la mayor parte de los elementos que configuran un controlador.

Un microcontrolador dispone normalmente de los siguientes componentes:

Procesador o UCP (Unidad Central de Proceso).

Memoria RAM para Contener los datos.

Memoria para el programa tipo ROM/PROM/EPROM.

Líneas de E/S para comunicarse con el exterior.

Diversos módulos para el control de periféricos (temporizadores, Puertas Serie y Paralelo, CAD: Conversores Analógico/Digital, CDA: Conversores Digital/Analógico, etc.).

Generador de impulsos de reloj que sincronizan el funcionamiento de todo el sistema.

Los productos que para su regulación incorporan un microcontrolador disponen de las siguientes ventajas:

Aumento de prestaciones: un mayor control sobre un determinado elemento representa una mejora considerable en el mismo.

Aumento de la fiabilidad: al reemplazar el microcontrolador por un elevado número de elementos disminuye el riesgo de averías y se precisan menos ajustes.

Reducción del tamaño en el producto acabado: La integración del microcontrolador en un chip disminuye el volumen, la mano de obra y los stocks.

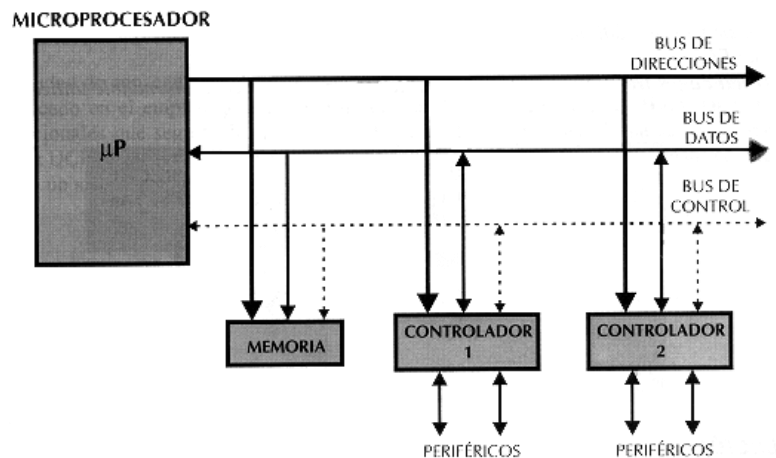
Mayor flexibilidad: las características de control están programadas por lo que su modificación sólo necesita cambios en el programa de instrucciones.

El microcontrolador es en definitiva un circuito integrado que incluye todos los componentes de un computador. Debido a su reducido tamaño es posible montar el controlador en el propio dispositivo al que gobierna. En este caso el controlador recibe el nombre de controlador empotrado (embedded controller).

DIFERENCIA ENTRE MICROPROCESADOR Y MICROCONTROLADOR.

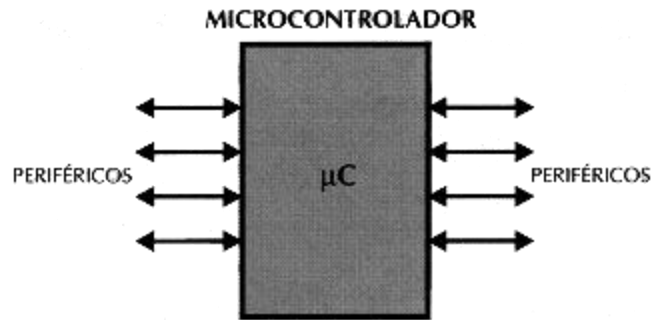
El microprocesador es un circuito integrado que contiene la Unidad Central de Proceso (UCP), también llamada procesador, de un computador. La UCP está formada por la Unidad de Control, que interpreta las instrucciones, y el Camino de Datos, que las ejecuta.

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



Estructura de un sistema abierto basado en un microprocesador. La disponibilidad de los buses en el exterior permite que se configure a la medida de la aplicación.

Si sólo se dispusiese de un modelo de microcontrolador, éste debería tener muy potenciados todos sus recursos para poderse adaptar a las exigencias de las diferentes aplicaciones. Esta potenciación supondría en muchos casos un despilfarro. En la práctica cada fabricante de microcontroladores oferta un elevado número de modelos diferentes, desde los más sencillos hasta los más poderosos. Es posible seleccionar la capacidad de las memorias, el número de líneas de E/S, la cantidad y potencia de los elementos auxiliares, la velocidad de funcionamiento, etc. Por todo ello, un aspecto muy destacado del diseño es la selección del microcontrolador a utilizar.



El microcontrolador es un sistema cerrado. Todas las partes del computador están contenidas en su interior y sólo salen al exterior las líneas que gobiernan los periféricos.

APLICACIONES DE LOS MICROCONTROLADORES.

Cada vez existen más productos que incorporan un microcontrolador con el fin de aumentar sustancialmente sus prestaciones, reducir su tamaño y coste, mejorar su fiabilidad y disminuir el consumo.

Algunos fabricantes de microcontroladores superan el millón de unidades de un modelo determinado producidas en una semana. Este dato puede dar una idea de la masiva utilización de estos componentes.

Los microcontroladores están siendo empleados en multitud de sistemas presentes en nuestra vida diaria, como pueden ser juguetes, horno microondas, frigoríficos, televisores, computadoras, impresoras, módems, el sistema de arranque de nuestro coche, etc. Y otras aplicaciones con las que seguramente no estaremos tan familiarizados como instrumentación electrónica, control de sistemas en una nave espacial, etc. Una aplicación típica podría emplear varios microcontroladores para controlar pequeñas partes del sistema. Estos pequeños controladores podrían comunicarse entre ellos y con un procesador central, probablemente más potente, para compartir la información y coordinar sus acciones, como, de hecho, ocurre ya habitualmente en cualquier PC.

EL MERCADO DE LOS MICROCONTROLADORES.

Aunque en el mercado de la microinformática la mayor atención la acaparan los desarrollos de los microprocesadores, lo cierto es que se venden cientos de microcontroladores por cada uno de aquéllos.

Existe una gran diversidad de microcontroladores. Quizá la clasificación más importante sea entre microcontroladores de 4, 8, 16 ó 32 bits. Aunque las prestaciones de los microcontroladores de 16 y 32 bits son superiores a los de 4 y 8 bits, la realidad es que los microcontroladores de 8 bits dominan el mercado y los de 4 bits se resisten a desaparecer. La razón de esta tendencia es que los microcontroladores de 4 y 8 bits son apropiados para la gran mayoría de las aplicaciones, lo que hace absurdo emplear micros más potentes y consecuentemente más caros. Uno de los sectores que más tira del mercado del microcontrolador es el mercado automovilístico. De hecho, algunas de las familias de microcontroladores actuales se desarrollaron pensando en este sector, siendo modificadas posteriormente para adaptarse a sistemas más genéricos. El mercado del automóvil es además uno de los más exigentes: los componentes electrónicos deben operar bajo condiciones extremas de vibraciones, choques, ruido, etc. y seguir siendo fiables. El fallo de cualquier componente en un automóvil puede ser el origen de un accidente.

En cuanto a las técnicas de fabricación, cabe decir que prácticamente la totalidad de los microcontroladores actuales se fabrican con tecnología CMOS 4 (Complementary Metal Oxide Semiconductor). Esta tecnología supera a las técnicas anteriores por su bajo consumo y alta inmunidad al ruido.

La distribución de las ventas según su aplicación es la siguiente:

Una tercera parte se absorbe en las aplicaciones relacionadas con los computadores y sus periféricos.

La cuarta parte se utiliza en las aplicaciones de consumo (electrodomésticos, juegos, TV, vídeo, etc.)

El 16% de las ventas mundiales se destinó al área de las comunicaciones.

Otro 16% fue empleado en aplicaciones industriales.

El resto de los microcontroladores vendidos en el mundo, aproximadamente un 10% fueron adquiridos por las industrias de automoción.

También los modernos microcontroladores de 32 bits van afianzando sus posiciones en el mercado, siendo las áreas de más interés el procesamiento de imágenes, las comunicaciones, las aplicaciones militares, los procesos industriales y el control de los dispositivos de almacenamiento masivo de datos.

¿Qué microcontrolador emplear?

A la hora de escoger el microcontrolador a emplear en un diseño concreto hay que tener en cuenta multitud de factores, como la documentación y herramientas de desarrollo disponibles y su precio, la cantidad de fabricantes que lo producen y por supuesto las características del microcontrolador (tipo de memoria de programa, número de temporizadores, interrupciones, etc.):

Costes. Como es lógico, los fabricantes de microcontroladores compiten duramente para vender sus productos. Y no les va demasiado mal ya que sin hacer demasiado ruido venden 10 veces más microcontroladores que microprocesadores.

Para que nos hagamos una idea, para el fabricante que usa el microcontrolador en su producto una diferencia de precio en el microcontrolador de algunas pesetas es importante (el consumidor deberá pagar además el coste del empaquetado, el de los otros componentes, el diseño del hardware y el desarrollo del software). Si el fabricante desea reducir costes debe tener en cuenta las herramientas de apoyo con que va a contar: emuladores, simuladores, ensambladores, compiladores, etc. Es habitual que muchos de ellos siempre se decanten por microcontroladores pertenecientes a una única familia.

Aplicación. Antes de seleccionar un microcontrolador es imprescindible analizar los requisitos de la aplicación:

- **Procesamiento de datos:** puede ser necesario que el microcontrolador realice cálculos críticos en un tiempo limitado. En ese caso debemos asegurarnos de seleccionar un dispositivo suficientemente rápido para ello. Por otro lado, habrá que tener en cuenta la precisión de los datos a manejar: si no es suficiente con un microcontrolador de 8 bits, puede ser necesario acudir a microcontroladores de 16 ó 32 bits, o incluso a hardware de coma flotante. Una alternativa más barata y quizá suficiente es usar librerías para manejar los datos de alta precisión. -
- **Entrada Salida:** para determinar las necesidades de Entrada/Salida del sistema es conveniente dibujar un diagrama de bloques del mismo, de tal forma que sea sencillo identificar la cantidad y tipo de señales a controlar. Una vez realizado este análisis puede ser necesario añadir periféricos hardware externos o cambiar a otro microcontrolador más adecuado a ese sistema.
- **Consumo:** algunos productos que incorporan microcontroladores están alimentados con baterías y su funcionamiento puede ser tan vital como activar una alarma antirrobo. Lo más conveniente en un caso como éste puede ser que el microcontrolador esté en estado de bajo consumo pero que despierte ante la activación de una señal (una interrupción) y ejecute el programa adecuado para procesarla.
- **Memoria:** para detectar las necesidades de memoria de nuestra aplicación debemos separarla en memoria volátil (RAM), memoria no volátil (ROM, EPROM, etc.) y memoria no volátil modificable (EEPROM). Este último tipo de memoria puede ser útil para incluir información específica de la aplicación como un número de serie o parámetros de calibración.

El tipo de memoria a emplear vendrá determinado por el volumen de ventas previsto del producto: de menor a mayor volumen será conveniente emplear EPROM, OTP y ROM. En cuanto a la cantidad de memoria necesaria puede ser

imprescindible realizar una versión preliminar, aunque sea en pseudo-código, de la aplicación y a partir de ella hacer una estimación de cuánta memoria volátil y no volátil es necesaria y si es conveniente disponer de memoria no volátil modificable.

- Ancho de palabra: el criterio de diseño debe ser seleccionar el microcontrolador de menor ancho de palabra que satisfaga los requerimientos de la aplicación. Usar un microcontrolador de 4 bits supondrá una reducción en los costes importante, mientras que uno de 8 bits puede ser el más adecuado si el ancho de los datos es de un byte. Los microcontroladores de 16 y 32 bits, debido a su elevado coste, deben reservarse para aplicaciones que requieran sus altas prestaciones (Entrada/Salida potente o espacio de direccionamiento muy elevado).

- Diseño de la placa: la selección de un microcontrolador concreto condicionará el diseño de la placa de circuitos. Debe tenerse en cuenta que quizá usar un microcontrolador barato encarezca el resto de componentes del diseño.

Los microcontroladores más populares se encuentran, sin duda, entre las mejores elecciones:

8048 (Intel). Es el padre de los microcontroladores actuales, el primero de todos. Su precio, disponibilidad y herramientas de desarrollo hacen que todavía sea muy popular.

8051 (Intel y otros). Es sin duda el microcontrolador más popular. Fácil de programar, pero potente. Está bien documentado y posee cientos de variantes e incontables herramientas de desarrollo.

80186, 80188 y 80386 EX (Intel). Versiones en microcontrolador de los populares microprocesadores 8086 y 8088. Su principal ventaja es que permiten aprovechar las herramientas de desarrollo para PC.

68HC11 (Motorola y Toshiba). Es un microcontrolador de 8 bits potente y popular con gran cantidad de variantes.

683xx (Motorola). Surgido a partir de la popular familia 68k, a la que se incorporan algunos periféricos. Son microcontroladores de altísimas prestaciones.

PIC (MicroChip). Familia de microcontroladores que gana popularidad día a día. Fueron los primeros microcontroladores RISC.

Es preciso resaltar en este punto que existen innumerables familias de microcontroladores, cada una de las cuales posee un gran número de variantes.

RECURSOS COMUNES A TODOS LOS MICROCONTROLADORES.

Al estar todos los microcontroladores integrados en un chip, su estructura fundamental y sus características básicas son muy parecidas. Todos deben disponer de los bloques esenciales Procesador, memoria de datos y de instrucciones, líneas de E/S, oscilador de reloj y módulos controladores de periféricos. Sin embargo, cada fabricante intenta enfatizar los recursos más idóneos para las aplicaciones a las que se destinan preferentemente.

En este apartado se hace un recorrido de todos los recursos que se hallan en todos los microcontroladores describiendo las diversas alternativas y opciones que pueden encontrarse según el modelo seleccionado.

♦ Arquitectura básica

Aunque inicialmente todos los microcontroladores adoptaron la arquitectura clásica de von Neumann, en el momento presente se impone la arquitectura Harvard. La arquitectura de von Neumann se caracteriza por disponer de una sola memoria principal donde se almacenan datos e instrucciones de forma indistinta. A dicha memoria se accede a través de un sistema de buses único (direcciones, datos y control).

La arquitectura Harvard dispone de dos memorias independientes una, que contiene sólo instrucciones y otra, sólo datos. Ambas disponen de sus

respectivos sistemas de buses de acceso y es posible realizar operaciones de acceso (lectura o escritura) simultáneamente en ambas memorias. Figura 1.3.

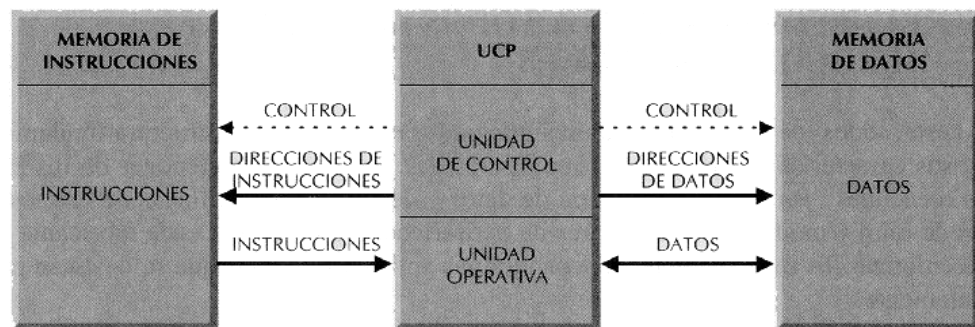


Figura 1.3. La arquitectura Harvard dispone de dos memorias independientes para datos y para instrucciones, permitiendo accesos simultáneos.

Los microcontroladores PIC responden a la arquitectura Harvard.

♦ El procesador o UCP

Es el elemento más importante del microcontrolador y determina sus principales características, tanto a nivel hardware como software.

Se encarga de direccionar la memoria de instrucciones, recibir el código OP de la instrucción en curso, su decodificación y la ejecución de la operación que implica la instrucción, así como la búsqueda de los operandos y el almacenamiento del resultado.

Existen tres orientaciones en cuanto a la arquitectura y funcionalidad de los procesadores actuales.

CISC: Un gran número de procesadores usados en los microcontroladores están basados en la filosofía CISC (Computadores de Juego de Instrucciones Complejo). Disponen de más de 80 instrucciones máquina en su repertorio,

algunas de las cuales son muy sofisticadas y potentes, requiriendo muchos ciclos para su ejecución.

Una ventaja de los procesadores CISC es que ofrecen al programador instrucciones complejas que actúan como macros.

RISC: Tanto la industria de los computadores comerciales como la de los microcontroladores están decantándose hacia la filosofía RISC (Computadores de Juego de Instrucciones Reducido). En estos procesadores el repertorio de instrucciones máquina es muy reducido y las instrucciones son simples y, generalmente, se ejecutan en un ciclo.

La sencillez y rapidez de las instrucciones permiten optimizar el hardware y el software del procesador.

SISC: En los microcontroladores destinados a aplicaciones muy concretas, el juego de instrucciones, además de ser reducido, es "específico", o sea, las instrucciones se adaptan a las necesidades de la aplicación prevista. Esta filosofía se ha bautizado con el nombre de SISC (Computadores de Juego de Instrucciones Específico).

♦ **Memorias**

En los microcontroladores la memoria de instrucciones y datos está integrada en el propio chip. Una parte debe ser no volátil, tipo ROM, y se destina a contener el programa de instrucciones que gobierna la aplicación. Otra parte de memoria será tipo RAM, volátil, y se destina a guardar las variables y los datos.

Hay dos peculiaridades que diferencian a los microcontroladores de los computadores personales:

No existen sistemas de almacenamiento masivo como disco duro o disquetes.

Como el microcontrolador sólo se destina a una tarea en la memoria ROM, sólo hay que almacenar un único programa de trabajo.

RAM en estos dispositivos es de poca capacidad pues sólo debe contener las variables y los cambios de información que se produzcan en el transcurso del programa. Por otra parte, como sólo existe un programa activo, no se requiere guardar una copia del mismo en la RAM pues se ejecuta directamente desde la ROM.

Los usuarios de computadores personales están habituados a manejar Megabytes de memoria, pero, los diseñadores con microcontroladores trabajan con capacidades de ROM comprendidas entre 512 bytes y 8 k bytes y de RAM comprendidas entre 20 y 512 bytes.

Según el tipo de memoria ROM que dispongan los microcontroladores, la aplicación y utilización de los mismos es diferente. Se describen las cinco versiones de memoria no volátil que se pueden encontrar en los microcontroladores del mercado.

ROM con máscara. Es una memoria no volátil de sólo lectura cuyo contenido se graba durante la fabricación del chip. El elevado coste del diseño de la máscara sólo hace aconsejable el empleo de los microcontroladores con este tipo de memoria cuando se precisan cantidades superiores a varios miles de unidades.

OTP el microcontrolador contiene una memoria no volátil de sólo lectura "programable una sola vez" por el usuario. OTP (One Time Programmable). Es el usuario quien puede escribir el programa en el chip mediante un sencillo grabador controlado por un programa desde un PC.

La versión OTP es recomendable cuando es muy corto el ciclo de diseño del producto, o bien, en la construcción de prototipos y series muy pequeñas.

Tanto en este tipo de memoria como en la EPROM, se suele usar la encriptación mediante fusibles para proteger el código contenido.

EPROM los microcontroladores que disponen de memoria EPROM (Erasable Programmable Read Only Memory) pueden borrarse y grabarse muchas veces. La grabación se realiza, como en el caso de los OTP, con un grabador

gobernado desde un PC. Si, posteriormente, se desea borrar el contenido, disponen de una ventana de cristal en su superficie por la que se somete a la EPROM a rayos ultravioleta durante varios minutos. Las cápsulas son de material cerámico y son más caros que los microcontroladores con memoria OTP que están hechos con material plástico.

EEPROM se trata de memorias de sólo lectura, programables y borrables eléctricamente EEPROM (Electrical Erasable Programmable Read Only Memory). Tanto la programación como el borrado, se realizan eléctricamente desde el propio grabador y bajo el control programado de un PC. Es muy cómoda y rápida la operación de grabado y la de borrado. No disponen de ventana de cristal en la superficie.

Los microcontroladores dotados de memoria EEPROM una vez instalados en el circuito, pueden grabarse y borrarse cuantas veces se quiera sin ser retirados de dicho circuito. Para ello se usan "grabadores en circuito" que confieren una gran flexibilidad y rapidez a la hora de realizar modificaciones en el programa de trabajo.

El número de veces que puede grabarse y borrarse una memoria EEPROM es finito, por lo que no es recomendable una reprogramación continua. Son muy idóneos para la enseñanza y la Ingeniería de diseño.

Se va extendiendo en los fabricantes la tendencia de incluir una pequeña zona de memoria EEPROM en los circuitos programables para guardar y modificar cómodamente una serie de parámetros que adecuan el dispositivo a las condiciones del entorno. Este tipo de memoria es relativamente lenta.

FLASH se trata de una memoria no volátil, de bajo consumo, que se puede escribir y borrar. Funciona como una ROM y una RAM pero consume menos y es más pequeña.

A diferencia de la ROM, la memoria FLASH es programable en el circuito. Es más rápida y de mayor densidad que la EEPROM.

La alternativa FLASH está recomendada frente a la EEPROM cuando se precisa gran cantidad de memoria de programa no volátil. Es más veloz y tolera más ciclos de escritura/borrado.

Las memorias EEPROM y FLASH son muy útiles al permitir que los microcontroladores que las incorporan puedan ser reprogramados "en circuito", es decir, sin tener que sacar el circuito integrado de la tarjeta. Así, un dispositivo con este tipo de memoria incorporado al control del motor de un automóvil permite que pueda modificarse el programa durante la rutina de mantenimiento periódico, compensando los desgastes y otros factores tales como la compresión, la instalación de nuevas piezas, etc. La reprogramación del microcontrolador puede convertirse en una labor rutinaria dentro de la puesta a punto.

♦ Puertas de Entrada y Salida

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

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

♦ Reloj principal

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

Generalmente, el circuito de reloj está incorporado en el microcontrolador y sólo se necesitan unos pocos componentes exteriores para seleccionar y estabilizar la frecuencia de trabajo. Dichos componentes suelen consistir en un cristal de cuarzo junto a elementos pasivos o bien un resonador cerámico o una red R-C.

Aumentar la frecuencia de reloj supone disminuir el tiempo en que se ejecutan las instrucciones pero lleva aparejado un incremento del consumo de energía.

RECURSOS ESPECIALES

Cada fabricante oferta numerosas versiones de una arquitectura básica de microcontrolador. En algunas amplía las capacidades de las memorias, en otras incorpora nuevos recursos, en otras reduce las prestaciones al mínimo para aplicaciones muy simples, etc. La labor del diseñador es encontrar el modelo mínimo que satisfaga todos los requerimientos de su aplicación. De esta forma, minimizará el coste, el hardware y el software.

Los principales recursos específicos que incorporan los microcontroladores son:

- Temporizadores o "Timers".
- Perro guardián o "Watchdog".
- Protección ante fallo de alimentación o "Brownout".
- Estado de reposo o de bajo consumo.
- Conversor A/D.
- Conversor D/A.
- Comparador analógico.
- Modulador de anchura de impulsos o PWM.
- Puertas de E/S digitales.
- Puertas de comunicación.

♦ Temporizadores o "Timers"

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

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

Cuando se desean contar acontecimientos que se materializan por cambios de nivel o flancos en alguna de las patitas del microcontrolador, el mencionado registro se va incrementando o decrementando al ritmo de dichos impulsos.

♦ Perro guardián o "Watchdog"

Cuando el computador personal se bloquea por un fallo del software u otra causa, se pulsa el botón del reset y se reinicializa el sistema. Pero un microcontrolador funciona sin el control de un supervisor y de forma continuada las 24 horas del día. El Perro guardián consiste en un temporizador que, cuando se desborda y pasa por 0, provoca un reset automáticamente en el sistema.

Se debe diseñar el programa de trabajo que controla la tarea de forma que refresque o inicialice al Perro guardián antes de que provoque el reset. Si falla el programa o se bloquea, no se refrescará al Perro guardián y, al completar su temporización, "ladrará y ladrará" hasta provocar el reset.

♦ Protección ante fallo de alimentación o "Brownout"

Se trata de un circuito que resetea al microcontrolador cuando el voltaje de alimentación (VDD) es inferior a un voltaje mínimo ("brownout"). Mientras el voltaje de alimentación sea inferior al de brownout el dispositivo se mantiene reseteado, comenzando a funcionar normalmente cuando sobrepasa dicho valor.

♦ Estado de reposo ó de bajo consumo

Son abundantes las situaciones reales de trabajo en que el microcontrolador debe esperar, sin hacer nada, a que se produzca algún acontecimiento externo que le ponga de nuevo en funcionamiento. Para ahorrar energía, (factor clave en los aparatos portátiles), los microcontroladores disponen de una instrucción especial (SLEEP en los PIC), que les pasa al estado de reposo o de bajo consumo, en el cual los requerimientos de potencia son mínimos. En dicho estado se detiene el reloj principal y se "congelan" sus circuitos asociados, quedando sumido en un profundo "sueño" el microcontrolador. Al activarse una interrupción ocasionada por el acontecimiento esperado, el microcontrolador se despierta y reanuda su trabajo.

♦ Conversor A/D (CAD)

Los microcontroladores que incorporan un Conversor A/D (Analógico/Digital) pueden procesar señales analógicas, tan abundantes en las aplicaciones. Suelen disponer de un multiplexor que permite aplicar a la entrada del CAD diversas señales analógicas desde las patitas del circuito integrado.

♦ Conversor D/A (CDA)

Transforma los datos digitales obtenidos del procesamiento del computador en su correspondiente señal analógica que saca al exterior por una de las patitas de la cápsula. Existen muchos efectores que trabajan con señales analógicas.

♦ Comparador analógico

Algunos modelos de microcontroladores disponen internamente de un Amplificador Operacional que actúa como comparador entre una señal fija de referencia y otra variable que se aplica por una de las patitas de la cápsula. La salida del comparador proporciona un nivel lógico 1 ó 0 según una señal sea mayor o menor que la otra.

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

♦ **Modulador de anchura de impulsos o PWM**

Son circuitos que proporcionan en su salida impulsos de anchura variable, que se ofrecen al exterior a través de las patitas del encapsulado.

♦ **Puertos de E/S digitales**

Todos los microcontroladores destinan algunas de sus patitas a soportar líneas de E/S digitales. Por lo general, estas líneas se agrupan de ocho en ocho formando Puertos.

Las líneas digitales de los Puertos pueden configurarse como Entrada o como Salida cargando un 1 ó un 0 en el bit correspondiente de un registro destinado a su configuración.

♦ **Puertos de comunicación**

Con objeto de dotar al microcontrolador de la posibilidad de comunicarse con otros dispositivos externos, otros buses de microprocesadores, buses de sistemas, buses de redes y poder adaptarlos con otros elementos bajo otras normas y protocolos. Algunos modelos disponen de recursos que permiten directamente esta tarea, entre los que destacan:

UART, adaptador de comunicación serie asíncrona.

USART, adaptador de comunicación serie síncrona y asíncrona

Puerta paralela esclava para poder conectarse con los buses de otros microprocesadores.

USB (Universal Serial Bus), que es un moderno bus serie para los PC.

Bus I²C, que es un interfaz serie de dos hilos desarrollado por Philips.

CAN (Controller Area Network), para permitir la adaptación con redes de conexión multiplexado desarrollado conjuntamente por Bosch e Intel para el cableado de dispositivos en automóviles. En EE.UU. se usa el J1850.

DESARROLLO DEL SOFTWARE:

Ensamblador. La programación en lenguaje ensamblador puede resultar un tanto ardua para el principiante, pero permite desarrollar programas muy eficientes, ya que otorga al programador el dominio absoluto del sistema. Los fabricantes suelen proporcionar el programa ensamblador de forma gratuita y en cualquier caso siempre se puede encontrar una versión gratuita para los microcontroladores más populares.

Compilador. La programación en un lenguaje de alto nivel (como el C ó el Basic) permite disminuir el tiempo de desarrollo de un producto. No obstante, si no se programa con cuidado, el código resultante puede ser mucho más ineficiente que el programado en ensamblador. Las versiones más potentes suelen ser muy caras, aunque para los microcontroladores más populares pueden encontrarse versiones demo limitadas e incluso compiladores gratuitos.

Depuración: debido a que los microcontroladores van a controlar dispositivos físicos, los desarrolladores necesitan herramientas que les permitan comprobar el buen funcionamiento del microcontrolador cuando es conectado al resto de circuitos.

Simulador. Son capaces de ejecutar en un PC programas realizados para el microcontrolador. Los simuladores permiten tener un control absoluto sobre la ejecución de un programa, siendo ideales para la depuración de los mismos. Su gran inconveniente es que es difícil simular la entrada y salida de datos del microcontrolador. Tampoco cuentan con los posibles ruidos en las entradas, pero, al menos, permiten el paso físico de la implementación de un modo más

seguro y menos costoso, puesto que ahorraremos en grabaciones de chips para la prueba in-situ.

Placas de evaluación. Se trata de pequeños sistemas con un microcontrolador ya montado y que suelen conectarse a un PC desde el que se cargan los programas que se ejecutan en el microcontrolador. Las placas suelen incluir visualizadores LCD, teclados, LEDs, fácil acceso a los pines de E/S, etc. El sistema operativo de la placa recibe el nombre de programa monitor. El programa monitor de algunas placas de evaluación, aparte de permitir cargar programas y datos en la memoria del microcontrolador, puede permitir en cualquier momento realizar ejecución paso a paso, monitorizar el estado del microcontrolador o modificar los valores almacenados los registros o en la memoria.

Emuladores en circuito. Se trata de un instrumento que se coloca entre el PC anfitrión y el zócalo de la tarjeta de circuito impreso donde se alojará el microcontrolador definitivo. El programa es ejecutado desde el PC, pero para la tarjeta de aplicación es como si lo hiciese el mismo microcontrolador que luego irá en el zócalo. Presenta en pantalla toda la información tal y como luego sucederá cuando se coloque la cápsula.

IV. JUSTIFICACIÓN

Con el Sistema Académico Modular por Objeto de Transformación (SAMOT), el mismo que fue implementado por la Universidad Nacional de Loja en Octubre de 1990 con el único propósito de romper el sistema académico tradicionalista, para mejorar la enseñanza aprendizaje de docentes y estudiantes.

Por lo que es importante la educación ya que ella impacta en todos los ámbitos de la vida, en la productividad laboral, en la participación ciudadana y en general en el mejoramiento de la calidad de vida.

El trabajo práctico que se propone realizar en este módulo es: “Implementación del laboratorio de Sistemas Microprocesados”.

Los microprocesadores y microcontroladores por su tamaño han permitido computarizar relojes, calculadoras, televisores, juegos electrónicos, por ejemplo las máquinas registradoras que leen los códigos de barras de un producto supermercado, registran el código del producto, su costo totalizan y proveen de una salida impresa del valor a pagar son basadas en microprocesadores.

Razón por la cual se justifica la ejecución de este trabajo desde el punto de vista académico, porque va a permitir adquirir nuevos conocimientos, que por ende benefician al desarrollo de la ciudad de Loja y la región sur del país, porque la falta inversión en nuestra institución obliga a dar solución a ciertas necesidades que deben ser atendidas urgentemente, por que se facilita la adquisición de sus materiales son fáciles de adquirir en nuestro medio, además resulta más económico construir dicho laboratorio.

Todo esto con la finalidad de aportar con alguna alternativa de solución al problema que estamos viviendo, siguiendo las fases del método científico y la investigación aportando de esta manera al desarrollo tecnológico de nuestra carrera, para ofrecer un laboratorio de prácticas que beneficiara a los futuros aspirantes de esta carrera.

Muchas industrias utilizan a los microprocesadores para monitorear y controlar sus líneas de ensamblaje y cumplir con los requisitos de control de calidad. Los microprocesadores también son utilizados en sistemas telefónicos brindando un mayor beneficio a la sociedad.

V. OBJETIVOS

General:

- ♦ Implementar el laboratorio de Sistemas Microprocesados.

Específicos:

- ♦ Proporcionar alternativas de solución para una mejor capacitación práctica del estudiante en el área de sistemas microprocesados.
- ♦ Poner en práctica los conocimientos adquiridos durante el periodo académico de la carrera.
- ♦ Conocer de mejor manera los elementos que van a ser utilizados en la implementación del laboratorio esto es su procedencia, adquisición, funcionamiento y programación.
- ♦ Contribuir con el desarrollo académico de la carrera en electrónica.

VI. METODOLOGÍA

La metodología es un procedimiento general para lograr de una manera precisa el objetivo de la investigación, de allí que la metodología nos permite conocer los métodos y técnicas más recomendables para realizar el presente trabajo práctico.

El trabajo práctico está dirigido a implementar el laboratorio de microprocesadores y microcontroladores.

Para llevar a cabo la implementación, se estructura la recolección de la información que fue revisada en Internet, libros y folletos.

Luego se procede a clasificar la información de acuerdo a los microprocesadores y microcontroladores y a lo que son los PIC`S y obteniendo así lo que es el marco teórico, de esta manera se abre paso al diseño del prototipo aplicando lo que es el dibujo técnico, que nos servirá como una guía para poder construir el

VIII. BIBLIOGRAFÍA

Libros:

- REYES, CARLOS A., Aprenda rápidamente a programar Microcontroladores PIC, edición 2004, GRÁFICAS AYERVE C.A., Quito, EC.

Sitios web:

- <http://www.intel.co.nz/espanol.com>
- <http://www.Laboratorio> de Microcontroladores-mografías.com
- <http://members.tripod.com/~alpertron.com>