



ESCUELA SUPERIOR POLITÉCNICA DEL LITORAL
FACULTAD DE INGENIERÍA EN ELECTRICIDAD Y COMPUTACIÓN

TESIS DE GRADO:

***Sistema Modular para el Laboratorio de
Microcontroladores de la ESPOL***

PREVIA A LA OBTENCIÓN DEL TÍTULO DE:

MAGISTER EN AUTOMATIZACION Y CONTROL INDUSTRIAL

PRESENTADA POR:

CARLOS ENRIQUE VALDIVIESO ARMENDARIZ

GUAYAQUIL – ECUADOR

2009

AGRADECIMIENTO

Ante todo a Dios, luego a mis padres que observarán con agrado esta pequeña contribución a la sociedad desde la paz del señor. A mis hijas que estoy seguro superarán con creces los ejemplos de su padre. A mis hermanas por su cariño y respaldo. A mi esposa y compañera por su apoyo y comprensión. A mis compañeros de maestría y colegas de largas jornadas educativas. A las autoridades de la FIEC en especial al Ing. Carlos Monsalve, quien me permitió formar parte del grupo de profesores que nos beneficiamos con la Maestría en Automatización Industrial. A mis alumnos que me han ayudado a probar la eficacia del sistema planteado en esta Tesis. A todos quienes de alguna manera me facilitaron información y ayuda a través de libros, publicaciones u otros medios.

DEDICATORIA

A mis alumnos quienes han sido el principal motivo de inspiración para dotarlos de un nuevo sistema de aprendizaje de los microcontroladores en forma ordenada y escalonada. Sistema que es muy dinámico y se va enriqueciendo día a día mediante experiencias adquiridas: en el dictado de un Tópico de Graduación, según la antigua modalidad; en la conducción de dos versiones de la nueva forma Materia de Graduación; y, en la Cátedra por varios años en la materia de Microcontroladores.

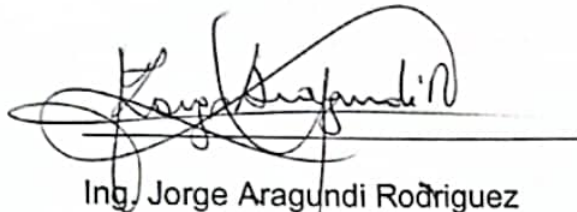
DECLARACIÓN EXPRESA

"La responsabilidad del contenido de este trabajo, me corresponde exclusivamente; y el patrimonio intelectual del mismo a la ESCUELA SUPERIOR POLITÉCNICA DEL LITORAL".

(Reglamento de exámenes y títulos profesionales de la ESPOL)

CARLOS ENRIQUE VALDIVIESO ARMENDARIZ

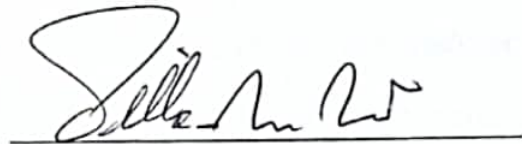
TRIBUNAL DE GRADUACIÓN



Ing. Jorge Aragundi Rodríguez

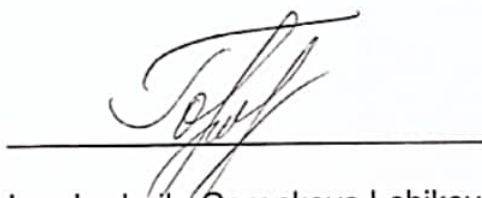
SUB-DECANO DE LA FACULTAD

PRESIDENTE



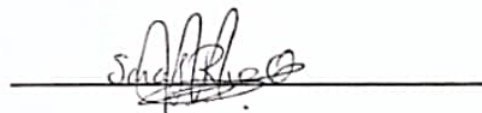
Ing. Hugo Villavicencio Villavicencio

DIRECTOR DE TESIS



Ing. Ludmila Gorenkova Labikova

MIEMBRO PRINCIPAL



Ing. Sara Ríos Orellana

MIEMBRO PRINCIPAL

RESUMEN

La presente Tesis corresponde a la creación de un Sistema Modular para facilitar la enseñanza de los microcontroladores en el ámbito de prácticas de laboratorio. Para lo cual se hizo un estudio de la realidad existente en el mercado, analizando la manera más eficiente en la que se podría beneficiar al estudiante de la ESPOL y hacerlo extensivo al estudiante ecuatoriano en general. Lográndose visualizar un sistema basado en nuestra realidad como país y como institución educativa que nos permita la formación de nuevas capacidades en los alumnos que cursan la materia de Microcontroladores que van a ser de gran beneficio en su futuro profesional.

ABREVIATURAS

A	Amperio
ADC	Convertidor analógico digital
bit	Unidad de medida de información equivalente a la elección entre dos posibilidades igualmente probables
dec.	Decremento
dsPIC	Microcontrolador de Microchip con soporte para procesamiento de señales
EEPROM	Memoria de escritura y lectura por medios eléctricos
I2C	Protocolo de comunicación serial
inc.	Incremento
LCD	Display de cristal líquido
MHz	Mega hertz
ohm	Ohmio
PWM	Modulación de ancho de pulso
PID	Proporcional Integral Derivativo
RPM	Revoluciones por minuto

INDICE DE IMÁGENES

FIGURA 1: TARJETA LAB-X1	9
FIGURA 2: TARJETA PICPROTO	10
FIGURA 3: PARTES PARA TARJETA PICPRO	11
FIGURA 4: TARJETA EASYPIC5	14
FIGURA 5: TARJETA PVK40	18
FIGURA 6: TARJETA DEPURADORA INSIDER	20
FIGURA 7: TARJETA PICDEM 4 CONECTADA CON EL ICD 2 DE MICROCHIP	22
FIGURA 8: ELEMENTOS RELACIONADOS CON EL SISTEMA MODULAR	39

INDICE DE TABLAS

TABLA 1: HABILIDADES QUE DEBEN FORMARSE	38
TABLA 2: PROYECTOS QUE REQUIEREN DE IMPLEMENTACIÓN FÍSICA	49
TABLA 3: SISTEMA MODULAR PROPUESTO	50

INDICE GENERAL

CAPÍTULO 1

ESTADO DEL ARTE DE LAS PLATAFORMAS DE DESARROLLADO EXISTENTES EN LA ACTUALIDAD. BÚSQUEDA BIBLIOGRÁFICA CON BREVE RESEÑA DE LOS MATERIALES ESTUDIADOS Y EL ANÁLISIS CRÍTICO DE LOS MISMOS	1
1.1 INTRODUCCIÓN	1
1.2 LAS PLATAFORMAS DE TRABAJO PROPUESTAS POR LAS EMPRESAS DEDICADAS AL DESARROLLO DE SISTEMAS DE ENSEÑANZA EN ESTA TEMÁTICA	8
1.2.1 TARJETA LAB-X1	8
1.2.1.1 ANALISIS DE SUS PRESTACIONES	11
1.2.2 TARJETA EASYPIC5	14
1.2.2.1 ANALISIS DE SUS PRESTACIONES	16
1.2.3 TARJETA PVK40	17
1.2.3.1 ANALISIS DE SUS PRESTACIONES	20
1.3 PLATAFORMA DE TRABAJO PROPUESTA POR EL PRINCIPAL FABRICANTE DE MICROCONTROLADORES PIC .	21
1.3.1 ANALISIS DE SUS PRESTACIONES	23

CAPÍTULO 2

PROPUESTA DE CONCEPCIÓN DEL SISTEMA MODULAR	25
---	----

2.1 INTRODUCCIÓN

2.2 ELEMENTOS DE HARDWARE MÁS UTILIZADOS EN LA

INDUSTRIA	29
-----------------	----

2.3 HABILIDADES PROFESIONALES A FORMAR EN EL LABORATORIO DE LA MICROCONTROLADORES	33
--	----

2.4 EXIGENCIAS QUE DEBE CUMPLIR EL SISTEMA MODULAR PROPUESTO	38
---	----

CAPÍTULO 3

DESARROLLO DEL SISTEMA MODULAR	45
--------------------------------------	----

3.1 DIFERENTES ELEMENTOS QUE CONFORMAN LA

PLATAFORMA	45
------------------	----

3.2 PRESENTACION DE TARJETAS USADAS EN EL SISTEMA MODULAR.....	50
---	----

3.3 SOFTWARE Y SIMULADORES UTILIZADOS	51
---	----

3.4 FUNCIONAMIENTO DEL SISTEMA MODULAR	56
--	----

CAPITULO 4

DESARROLO DE PRÁCTICAS PARA EL SISTEMA MODULAR	61
--	----

4.1 INTRODUCCION	61
------------------------	----

4.2 PRACTICAS PARA PREGRADO	62
4.2.1 PRÁCTICA 1: ENCENDIDO DE LEDS MEDIANTE INTERRUPTORES	62
4.2.2 PRÁCTICA 2: ACCIONAMIENTO DE UN DISPLAY DE 7 SEGMENTOS MEDIANTE INTERRUPTORES	81
4.2.3 PRÁCTICA 3: SEMAFORO PARA EL CONTROL DE INTERSECCION DE DOS VIAS UNIDIRECCIONALES	100
4.2.4 CONTADOR DE 10 CAJAS CON 8 ARTÍCULOS EN UNA LINEA DE PRODUCCION MEDIANTE EL USO DE LA INTERRUPCION EXTERNA	104
4.2.5 USO DE TECLADO MATRICIAL 4x4 CON VISUALIZACIÓN EN DISPLAY DE 7 SEGMENTOS ...	112
4.2.6 SISTEMA DE ACCESO MEDIANTE CLAVE CON LECTURA Y GRABACION DE MEMORIA EEPROM ..	128
4.2.7 MANEJO DE DISPLAY LCD Y SUS PRINCIPALES FUNCIONES	138
4.2.8 VELOCIDAD DE MOTOR DC CONTROLADA MEDIANTE PWM CUYO CICLO DE TRABAJO ES EL RESULTADO DE UNA CONVERSIÓN ANALÓGICO DIGITAL (ADC).....	147
4.2.9 COMUNICACIÓN I2C MAESTRO ESCLAVO ENTRE DOS MICROCONTROLADORES	159
4.3 PRACTICAS PARA LA MAESTRÍA EN AUTOMATIZACION INDUSTRIAL	173

4.3.1 CONSTRUCCION DE TARJETA PROTOTIPO BASICA USBPOLI, CARGA DE BOOTLOADER Y EJERCICIO DE APLICACIÓN	173
4.3.2 ALMACENAMIENTO DE DATOS ADQUIRIDOS EN LA MEMORIA EEPROM PARA SU POSTERIOR VISUALIZACIÓN Y ANALISIS MEDIANTE LABVIEW O VISUAL BASIC	189
4.3.3 CONTROL EN LAZO CERRADO	193
4.3.4 CONTROL PID APLICADO AL CONTROL DE VELOCIDAD DE UN MOTOR DC	198
CONCLUSIONES Y RECOMENDACIONES	203
ANEXOS	205
BIBLIOGRAFIA	209

INTRODUCCION

En este trabajo presenta un sistema modular para la enseñanza práctica de microcontroladores tomando en cuenta las limitaciones de nuestro medio y el claro objetivo de sacar el mayor provecho posible del mismo.

En el capítulo uno se hace un estudio de una muestra de lo que se ofrece en el mercado globalizado para trabajar en el laboratorio de microcontroladores y de esta manera estar en capacidad de diseñar en base a experiencias y requerimientos propios la plataforma que a criterio del autor sea considerada la que más facilite el aprendizaje práctico de los microcontroladores en una entidad educativa.

En el capítulo dos se abordará el tema de de la plataforma de trabajo a utilizarse, partiendo de la sencillez de los circuitos más simples que facilitan el entendimiento de los conceptos básicos para ir aumentando su complejidad a medida que se avanza en el estudio. Es decir que se va aprendiendo mediante módulos que se construyen uno en base a otro hasta alcanzar un grado satisfactorio de conocimientos y formación de habilidades que permitan enfrentar con solvencia nuevos sistemas con microcontroladores.

En el capítulo tres se presentan en conjunto los elementos del Sistema Modular como un ente coherente que permite la formación de hábitos y

habilidades para el diseño con microcontroladores en los estudiantes que cursan la materia a niveles de pregrado o maestría.

En el cuarto capítulo se presenta las prácticas recomendadas para pregrado en la primera parte y las prácticas recomendadas para el nivel de maestría en la segunda parte.

CAPÍTULO 1

ESTADO DEL ARTE DE LAS PLATAFORMAS DE DESARROLLO EXISTENTES EN LA ACTUALIDAD. BÚSQUEDA BIBLIOGRÁFICA CON BREVE RESEÑA DE LOS MATERIALES ESTUDIADOS Y EL ANÁLISIS CRÍTICO DE LOS MISMOS.

1.1 INTRODUCCIÓN

Existe una gran variedad de entornos de desarrollo utilizados en la enseñanza de los microcontroladores. El problema radica en encontrar un sistema que se adapte de la mejor manera con el programa de estudios de la materia de Microcontroladores que se dicta en la Escuela Superior Politécnica del Litoral. El sistema deberá considerar además las limitaciones de equipos y presupuestos para su mantenimiento y adquisición.

Planteado el problema se consideró de vital importancia analizar lo que se ofrece por parte de los principales proveedores y fabricantes de microcontroladores a nivel global. Para lo cual se procederá a hacer un estudio de tres proveedores escogidos al azar de entre los más importantes que presentan su información en la Web. Se habla solamente de una muestra de proveedores y fabricantes porque hay tanto material que sería demasiado extenso abarcar todo lo que hay sobre la temática. Considerándose las limitaciones del estudio propuesto se procedió a tomar una muestra, en base a la experiencia de entre las empresas y fabricantes más importantes

dedicados a esta tarea. Se los escogió al azar porque sería extremadamente difícil indicar cuales son los mejores de cada grupo por la cantidad de parámetros en juego. Cabe agregar que, resultaría extremadamente difícil y a lo mejor injusto clasificarlos por sus prestaciones dentro de una categoría definida. Con estos elementos y la experiencia de varios años de enseñanza teórico-práctica de Microcontroladores, se podrá tener una visión más amplia que permitirá presentar a la comunidad politécnica un sistema modular para Laboratorios de Microcontroladores adecuado al programa de estudios vigente.

El por qué se busca un sistema modular a la medida, se puede razonar indicando que lo que se ofrece en el mercado tiene variados propósitos y la misión del docente es la de impartir conocimientos de la manera más sencilla y explicativa. Presentando principios claros que se armonicen y se construyan unos sobre otros con tal claridad que permitan la comprensión de la materia objeto del curso con fluidez y sencillez.

Si se navega en la Web, la cantidad de información es tan grande que es muy difícil decidir qué escoger para enseñar. No es tan simple como para adoptar de buenas a primeras un sistema de un fabricante x sin reflexionar sobre las posibles consecuencias de esta decisión. ¿Se estará por el camino correcto?; ¿Se estará usando el hardware más apropiado?; ¿ El software que propone el fabricante

será el que se debe usar?; ¿Se está utilizando un sistema que permita y estimule el trabajo en grupo?. Estas y muchas otras interrogantes no pueden ser contestadas inmediatamente es necesario un análisis detallado donde influyen muchos factores entre los que juegan un papel preponderante el conocimiento previo de la materia, de los entornos de desarrollo existentes y propuestos, de la temática de los programas de estudio, de las limitaciones presentes, es decir variados parámetros que van matizándose con la experiencia del educador, su visión y perspectiva, hasta se podría decir por su particular arte de enseñar.

Los tiempos cambian y los requerimientos del profesional en ciencias técnicas son cada vez más exigentes debido en parte al desarrollo vertiginoso de la ciencia y tecnología, pudiéndose apreciar que estos aspectos se contraponen a los métodos tradicionales de enseñanza. Es necesario orientar la enseñanza desde nuevas perspectivas que faciliten el aprendizaje del Laboratorio de Microcontroladores, que pueda adaptarse en forma rápida a los avances tecnológicos y permitan impartir conocimiento en forma escalonada pero segura. Esta tarea se facilita mediante la armonización de Prácticas de Laboratorio que van de la mano con el marco teórico impartido en la asignatura.

Dentro del Laboratorio se estimula el trabajo grupal que es un reflejo de situaciones reales a las que se verán enfrentados los futuros

profesionales. El sistema empleado en la enseñanza deberá ser capaz de tomar en cuenta estos aspectos, posibilitando la interacción entre las personas que desarrollan una práctica. El número de participantes por grupo es también importante. La experiencia en trabajos de laboratorio docente, indica que un buen número es el de dos integrantes por grupo. ¿Porqué dos y no tres?; porque con tres es muy común ver que no trabajan todos y se pueden dar variadas situaciones: que sólo uno trabaje y los otros dos se distraigan; que dos trabajen y el otro se sienta fuera de la integración grupal por pereza, timidez o cualquier otro factor. Inclusive el control se vuelve un problema para el profesor mientras que el manejo de un grupo de dos estudiantes es una figura estándar que se ha venido utilizando a través de varios laboratorios dentro de la ESPOL.

El sistema a utilizarse deberá disponerse en forma tal que se adapte a la modalidad de prácticas de laboratorio estipuladas en el programa de estudio. Es decir que deberá seleccionarse con mucho cuidado el material para que cumpla con los objetivos de cada práctica y que además el hardware empleado deberá permitir cambiar aspectos de la práctica y poder seguir manteniendo el espíritu de la misma.

Muy importante por cierto que el hardware mantenga la sencillez necesaria para entender con facilidad las conexiones de los diversos

elementos. Cosa que es difícil lograrlo con los entornos provenientes de diversos fabricantes ya que tratan de presentar muchas cosas a la vez y lo que logran es confundir de entrada al principiante. Cuando quienes conocen de microcontroladores, saben que con muy pocas conexiones ya es posible energizar un micro. Básicamente dotándolo de una fuente de alimentación y de un circuito de reloj, que es la configuración básica sobre la que empezarán a construirse circuitos más complejos. Es decir con muy pocos elementos presentados en un sencillo diagrama de conexiones es posible presentar la poderosa potencialidad básica de un microcontrolador. Lo cual impacta beneficiosamente en los estudiantes que pueden observar tanto poder en circuitos tan simples.

Qué importante es el factor confianza y la satisfacción de ver que es posible hacer trabajar un circuito básico con muy pocos elementos e instrucciones. Esto definitivamente engancha la voluntad de muchos estudiantes que inmediatamente se ven atraídos por el estudio de los microcontroladores. Hay quienes quieren desbordar las aulas y realizar sus propias investigaciones. Esto es totalmente positivo y estimulante, una vez que el estudiante se va adentrando en el estudio crece su interés por nuevos temas. En este punto es muy importante el poder guiarlo adecuadamente para que avance positivamente. Pues el quedarse en rutinas sin solución puede ser frustrante, ya que el estudio de los micros demanda de mucho

tiempo, paciencia y capacidad de investigación para encontrar lo más rápidamente posible soluciones a los problemas abordados.

Por qué no hablar del ambiente de trabajo que debe acompañar al Sistema Modular de Enseñanza cuando esto es un complemento importantísimo que puede motivar o desmotivar al estudiantado. Por todos es conocido ambientes agradables con armonía, claridad y comodidad son propicios para el desarrollo de ciertas plantas no se diga del ser humano que es otro ser vivo que se influencia fácilmente por su entorno. Si trabaja en un Laboratorio limpio, cómodo, con sillas confortables y ambiente agradable es fácil comprobar que los estudiantes se van a sentir a gusto y en condiciones favorables tendrán como consecuencia un mejor desempeño.

Con el Sistema Modular no se busca uno o varios tableros aislados si no un conjunto de herramientas que combinan diversos factores como: el estímulo del trabajo en grupo, un entorno de trabajo agradable, programas de apoyo, hardware apropiado, programadores, conducción del profesor como experto en la materia, apoyo de ayudantes de laboratorio con el debido entrenamiento, más horas de apoyo en las que el o los estudiantes podrán trabajar en el Laboratorio fuera de las horas de clase convencionales.

Al estudiante de Laboratorio siempre se los estimula para que use los recursos disponibles. Es muy común verlos trabajando en horas

fuera de clase en proyectos que pueden no ser de la asignatura de Laboratorio de Microcontroladores, sino de cualquier otra materia que estén cursando y muchas veces en proyectos propios que han despertado su interés al conocer las potencialidades de los microcontroladores.

El interés mostrado por los estudiantes en el aprendizaje de esta materia ha motivado el deseo del cuerpo docente de la ESPOL, a cargo del área de Microcontroladores, de estudiar con detenimiento y preocuparse permanentemente por la mejora de la enseñanza y del rendimiento estudiantil. Producto de este esfuerzo constituye el modesto aporte presentado en este documento cuyas enseñanzas se las ha venido poniendo en práctica con buenos resultados en los últimos cuatro semestres en que se ha dictado la materia de Laboratorio de Microcontroladores. Por lo que vale indicar que lo que se expresa aquí no es pura teoría sino que son planteamientos llevados a la realidad y que contribuyen con la loable tarea de la formación profesional de futuros ingenieros politécnicos que representarán a la ESPOL desde los diferentes puestos de trabajo en los que lograrán desempeñarse en el transcurso de su vida profesional.

1.2 LAS PLATAFORMAS DE TRABAJO PROPUESTAS POR LAS EMPRESAS DEDICADAS AL DESARROLLO DE SISTEMAS DE ENSEÑANZA EN ESTA TEMÁTICA

Múltiples son las ofertas en este sentido y como se mencionó con anterioridad como no se puede abarcar todo el universo de los microcontroladores, en este estudio se tratará sólo sobre las propuestas de tres empresas que enfocan con mayor énfasis el estudio de los microcontroladores de la gama media.

A continuación se enumeran para luego analizar las características de cada una de las plataformas propuestas.

- 1) Tarjeta LAB-X1
- 2) Tarjeta EasyPic5
- 3) Tarjeta PVK40

1.2.1 TARJETA LAB-X1

Lab-X1 [1] es una plataforma para experimentación preensamblada. La cual contiene los circuitos que permiten el funcionamiento de los microcontroladores permitidos: fuente de poder de 5 voltios, cristal, reset y circuitos de aplicación. El circuito del oscilador de cristal incluye relojes de 4MHz, 8MHz, 10MHz, 12MHz, 13.32MHz, 16MHz y 20MHz, seleccionables mediante jumpers.

Los circuitos de aplicación incluyen una matriz de interruptores, potenciómetros, un bargraph de LEDs (módulo de 10 LEDs), módulo

LCD, interfase RS232 e interfase RS485. Opcionalmente se le pueden agregar otros circuitos como: EEPROMS seriales, reloj de tiempo real, sensores de temperatura, conectores para servos e interfaz infrarroja. Adicionalmente se incluye un espacio para construcción de prototipos en caso de que faltara alguno. Ver figura 1.

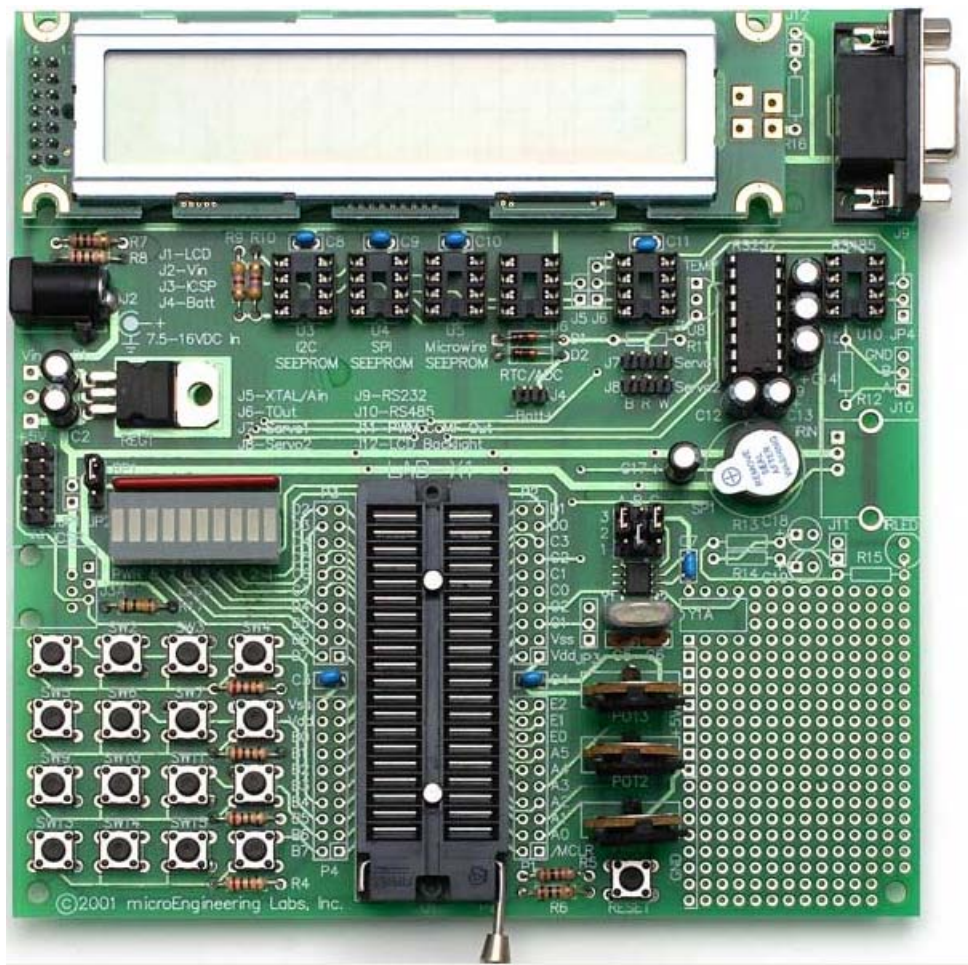


FIGURA 1: TARJETA LAB-X1

Muchos circuitos han sido creados usando Lab-X1 tales como: calculadoras, relojes digitales, termómetros digitales, transformación de LCD paralela a serial, secuenciador de tonos, control remoto para TV y muchos más. Con Lab-X1 pueden utilizarse programas escritos en lenguaje ensamblador, lenguaje C, picbasic y picbasic pro.

En la tarjeta se incluye un conector para la programación serial en circuito. Lab-X1 no es un programador y necesita de un programador externo que soporte esta posibilidad. Los microcontroladores a programar serán del tipo flash. Programadores soportados son el Melab USB Programmer y el Pickit 2 entre otros. Para este último se construyó un conector especial para su adaptación permitiendo la programación de los dispositivos flash colocados en la tarjeta Lab-X1 sin necesidad de sacarlos del conector.

La empresa que ofrece esta plataforma, recomienda adicionalmente la adquisición de la tarjeta PICPROTO con una cantidad de elementos básicos que permite la infraestructura necesaria para empezar a trabajar en un proyecto con microcontroladores PIC. Ver figura 2.

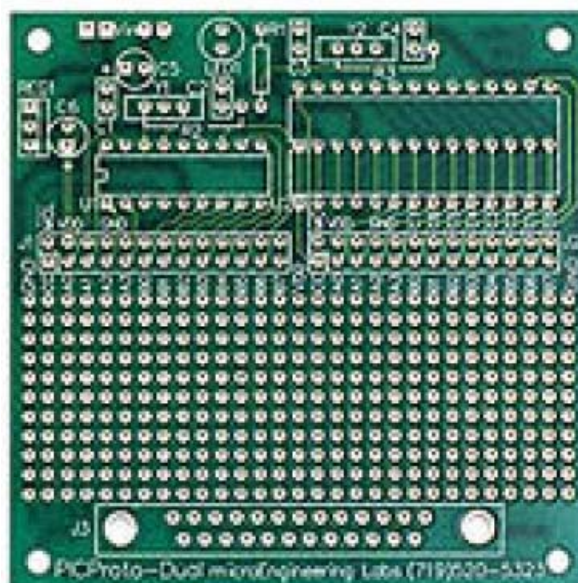


FIGURA 2: TARJETA PICPROTO

El kit de partes recomendado incluye:

1 regulador de 5 voltios

1 cristal de 4MHz

2 capacitores de 22pF

1 capacitor de 10uF

1 capacitor de 1uF

1 LED rojo

1 resistencia de 470 ohm

1 resistencia de 1k

1 conector de batería de 9 voltios

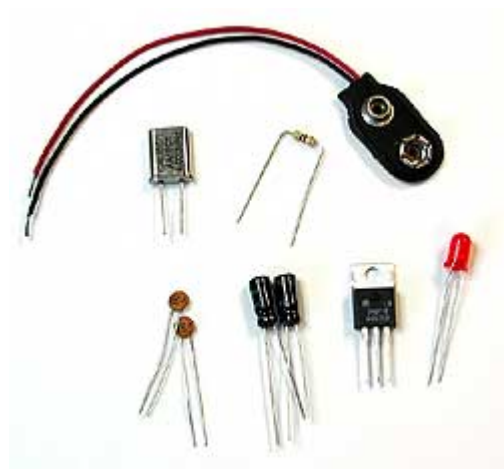


FIGURA 3: PARTES PARA TARJETA PICPRO

1.2.1.1 ANALISIS DE SUS PRESTACIONES

Con esta tarjeta existe la posibilidad de armar todos los circuitos básicos necesarios para cubrir el programa de estudios de Microcontroladores. Debiendo indicarse que se dedica exclusivamente a los microcontroladores de Microchip. Para los de Intel será necesario utilizar otra tarjeta. Ventajosamente en el curso de Microcontroladores sólo se compara el comportamiento de los microcontroladores PIC de Microchip con los de Intel. A partir del Segundo Término del período lectivo 2008-2009 se hizo realidad el proyecto de empezar a estudiar en la materia de Microprocesadores a los microcontroladores de Intel. Lo cual permitirá a los estudiantes una sólida formación en el área básica en lo que se refiere a

microcontroladores porque se amplían las posibilidades para el entendimiento de las aplicaciones relativas a dos grandes campos de estudio.

Las ventajas de la tarjeta radican en que permiten al docente la rápida implementación de aplicaciones sin tener que partir desde cero cada vez. Es conocido en este tipo de circuito que la fuente y el reloj son indispensables como punto de partida. De allí en adelante los elementos que se emplearán dependen de la aplicación que se quiera abordar. En el caso por ejemplo de un reloj digital habrá que poner atención en la provisión de displays de 7 segmentos, pulsadores, LEDs etc.

Se menciona el lado del docente porque es bajo esta perspectiva que se justifica la adquisición de una tarjeta de este tipo. Para el estudiante de nivel 400 que todavía no tiene la suficiente práctica en el manejo de circuitos electrónicos hay que preparar otro tipo de circuito. No se le puede entregar una tarjeta Lab-X1 porque la tarjeta corre el riesgo de quedar inhabilitada por manos inexpertas y además el hecho de estar todo en el mismo tablero no permite visualizar con claridad el énfasis sobre uno u otro circuito. Debe ser ese el motivo por el cual los fabricantes de esta plataforma recomiendan el uso de la tarjeta adicional PICPROTO con su conjunto de elementos básicos.

En el proceso de enseñanza de la materia de Microcontroladores se han experimentado varias modalidades: en un principio se le entregaba al estudiante un kit de partes básicas para cada proyecto, las mismas que deberían armarse de acuerdo con la práctica preestablecida. Si bien esto permitía al estudiantado poner las manos sobre los elementos y sentir su manipulación lo cual es muy beneficioso, se veía contrastado por el alto deterioro de los componentes utilizados y por la estrechez del horario de clase para cumplir etapas de diseño, simulación, programación y ensamblado en la misma sesión de laboratorio.

Posteriormente se procedió a dividir las prácticas de laboratorio de la materia teórica. Para enfrentar esta nueva modalidad y facilitar el dictado del laboratorio en base a experiencias anteriores se trabajó en la presente tesis aprovechando la coyuntura de cursar la MACI (Maestría en Automatización y Control Industrial) y la necesidad de presentar una Tesis de grado. La idea fue la de combinar la experiencia docente con el objetivo curricular de la MACI para dar solución a un problema profesional real que se presenta en cada término en los estudios de pregrado y aportar además con prácticas sugeridas para la Maestría en Automatización Industrial que podrán utilizarse, sobre el andamiaje de apoyo construido hasta el momento.

1.2.2 TARJETA EASYPIC5

La tarjeta EasyPic5 [2] es producida por la empresa Microelectrónica la cual presenta una propuesta muy interesante de desarrolladores y compiladores en lenguaje Basic, Ensamblador, Pascal, Lenguaje C entre otros. Se puede observar una imagen de la tarjeta en la figura 4, en la que se han conservado números referenciales para destacar las características de la tarjeta de una mejor manera.

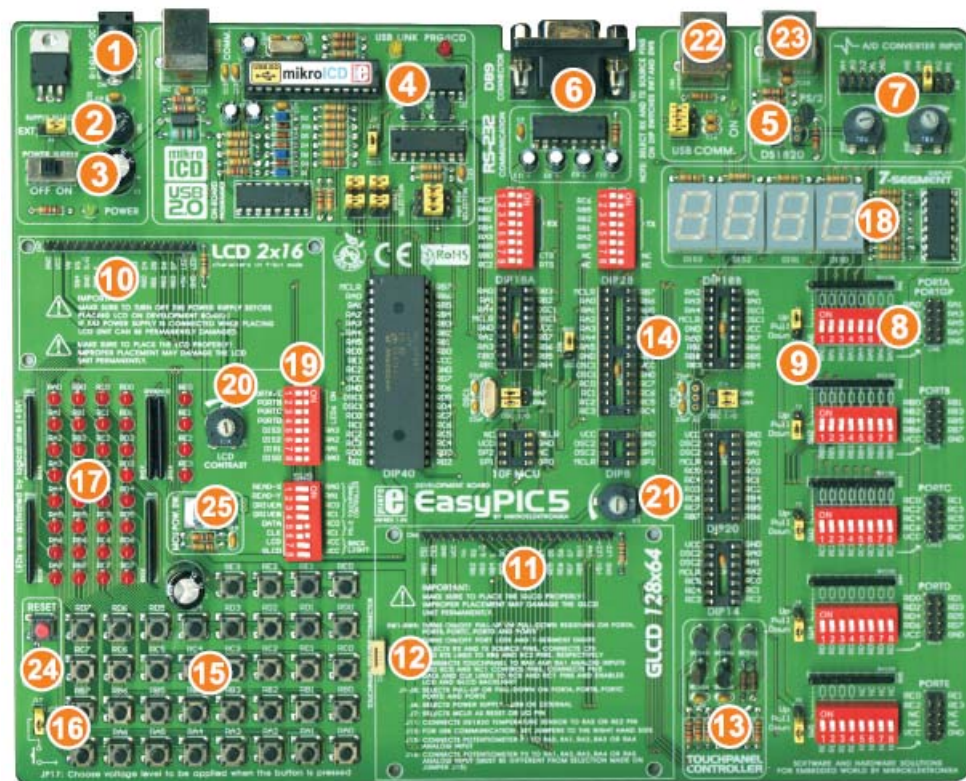


FIGURA 4: TARJETA EASYPIC5

Observando la numeración presentada sobre el gráfico de la figura 4 se destacan las siguientes características:

- 1) Fuente de poder externa de 8 -16 voltios AC/DC
- 2) Cuando se usa el puerto USB se puede escoger entre fuente de poder interna o externa

- 3) Interruptor de encendido y apagado de la fuente de poder
- 4) Sensor de temperatura DS1820 que permite mediciones de temperatura con 0.5 grados de exactitud.
- 5) Comunicación RS232 a través de Rx y Tx
- 6) Los pines RA0–RA5 pueden conectarse a los potenciómetros P1 y P2. Los cuales pueden ser usados como voltajes analógicos referenciales.
- 7) Por medio del grupo de interruptores en A1 se pueden conectar resistencias al puerto A. En caso de estar desconectados el puerto A puede ser usado como entradas analógicas.
- 8) Jumper para determinar si al puerto A se le colocan resistencias de pull up o de pull down.
- 9) Conector de 16 pines para insertar el LCD
- 10) Conector de 20 pines para la colocación de un LCD gráfico.
- 11) Conector para panel sensible al tacto.
- 12) Controlador de panel sensible al tacto.
- 13) Soporte para micros DIP8, DIP14, DIP18, DIP20, DIP28 Y DIP40. Por tanto la mayor parte de microcontroladores de Microchip pueden soportados.
- 14) 36 interruptores permiten el control de cada pin del microcontrolador.
- 15) Jumper que permite seleccionar estado de alto o bajo del pin que se va presionar.
- 16) 36 LEDs conectadas a los pines de entrada/salida
- 17) Displays de 7 segmentos en modo múltiple.

- 18) Prede y apaga LEDs en puertos A, B,C,D y E. También estos switches conectan y desconectan dígitos de los display de 7 segmentos. Finalmente estos interruptores permiten la desconexión de todos los LEDs de las patitas de los microcontroladores.
- 19) Potenciómetro para el contraste del LCD
- 20) Potenciómetro para el contraste del LCD gráfico.
- 21) Puerto USB para microcontroladores que lo soportan.
- 22) Conector para teclado.
- 23) Circuito de reset.
- 24) Interruptor de encendido de la tarjeta.

El fabricante de la tarjeta la publicita características de 3 en 1, es decir que permite: 1) Prueba de instrucciones a través de un ICD incorporado; 2) Programación en circuito 3) El desarrollo de aplicaciones.

1.2.2.1 ANALISIS DE SUS PRESTACIONES

La tarjeta EasyPic5 es una tarjeta bastante completa que incorpora una importante herramienta en el trabajo con microcontroladores que es un programador. En este caso un ICD que permite la prueba paso a paso y la depuración de instrucciones. Trae también incorporados un panel sensible al tacto y un panel gráfico. Dispositivos que se emplean con frecuencia en sistemas de automatización.

Lo anterior unido con la gran flexibilidad para manejar pines de entrada y salida con o sin el uso de pull ups, pull downs, interruptores o LEDs, lo hacen muy atractivo para una fácil implementación de aplicaciones.

Una vez más se debe añadir que una tarjeta de estas entregadas a un estudiante para las prácticas cotidianas de la clase de laboratorio constituiría un desastre seguro por la poca experiencia de los estudiantes de este nivel en el manejo de circuitos electrónicos. Tomado desde el punto de vista didáctico constituye una gran herramienta que aporta flexibilidad a la enseñanza y que el docente puede utilizar para afinar sus prácticas antes de exponerlas al estudiantado.

1.2.3 TARJETA PVK40

La tarjeta PVK40 [3] permite la conexión con microcontroladores tipo flash de las series PIC16F87x, PIC16F7x, PIC18Fxx. Incluye los circuitos necesarios para un estudio detallado de los periféricos del microcontrolador y su interconexión con el mundo exterior.

Los microcontroladores de 40 pines contienen más periféricos que los microcontroladores de menor cantidad de pines, manteniendo una compatibilidad en el mapeo de registros. Es por eso que todo el conocimiento y experiencias ganadas con los microcontroladores de

40 pines se aplican en forma similar a los de menor cantidad de pines. Ver figura 5.

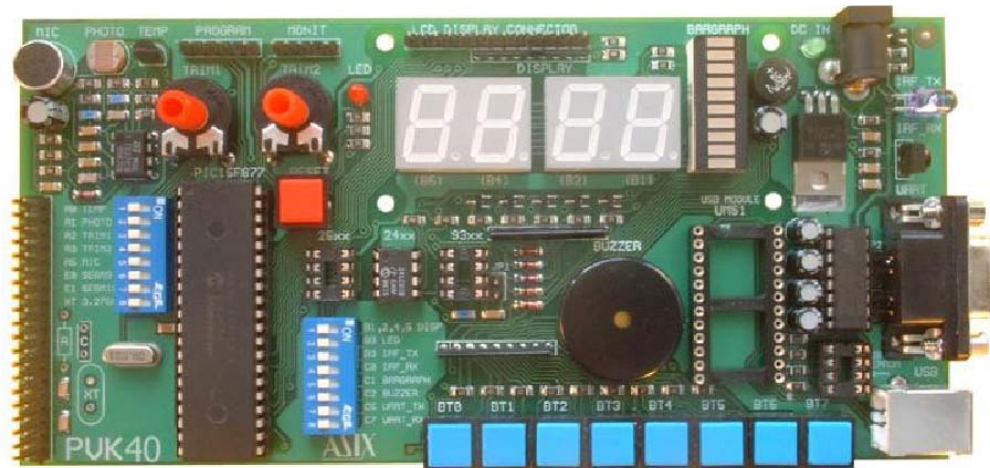


FIGURA 5: TARJETA PVK40

Entra las características de la tarjeta PVK40 se encuentran:

- Dos potenciómetros conectados a las entradas A/D (análoga-digital).
- Fotorresistor conectado a la entrada A/D.
- Micrófono con preamplificador conectado a la entrada A/D
- Zócalos para tres tipos de memorias seriales EEPROM (I2C – 24xx, MicroWire – 93xx, SPI – 25xx), 24LC01.
- Interfase RS232 (9 pines) con conversión de nivel.
- Módulo de conversión USB a serial (con zócalo para FTQDI FT232)
- Transmisor y receptor IR (infrarrojos).
- Display de 7 segmentos de 4 dígitos.

- Arreglo de LEDES (10 posiciones).
- Interfaz con display LCD
- 1 LED individual no multiplexado
- 8 pulsadores
- Resonador piezoeléctrico.
- Oscilador de cristal de 3.2768 MHz.
- Circuitos para osciladores de cristal y RC adicionales.
- Pulsador de reset.
- Fuente de 5 voltios estabilizada.
- Todos los pines de salida están accesibles a través de conectores para su interconexión con el mundo exterior.
- Soporte para la programación en circuito utilizando el programador PRESTO u otro programador compatible.

Como herramienta adicional presenta a la tarjeta INSIDER la cual permite la depuración de circuitos no sólo de microcontroladores de Microchip sino también de otros fabricantes como Atmel, Dallas, Phillips, Motorota, y más aún, FPGAs (Xilinx, Altera...). La principal ventaja de la tarjeta "INSIDER" su alto grado de efectividad en la depuración de aplicaciones en tiempo real sin un emulador o cualquier otro Depurador. Ver figura 6,

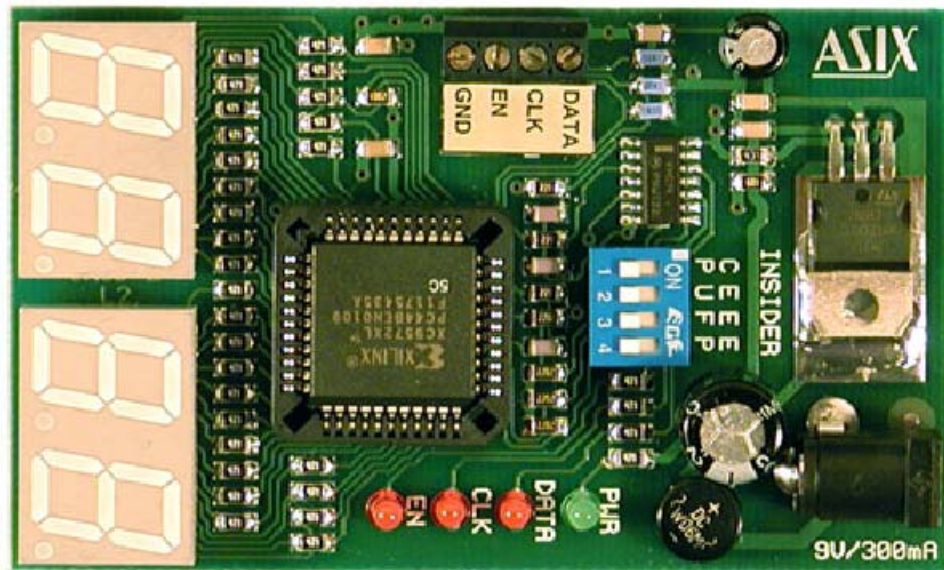


FIGURA 6: TARJETA DEPURADORA “INSIDER”

1.2.3.1 ANALISIS DE SUS PRESTACIONES

La tarjeta PVK40 presenta una disposición familiar a los otros entornos de las otras tarjetas estudiadas. Añadiendo ingredientes particulares como su programador en circuito PRESTO y su tarjeta depuradora INSIDER. Los fabricantes son checos y esta presentación checa avaliza la universalidad del conocimiento que de una u otra forma tiene sus principios básicos que deben ser cubiertos durante la enseñanza y son el punto de partida hacia infinidad de caminos que se abren en un abanico de aplicaciones.

No se puede calificar esta tarjeta como mejor o peor que alguna de las otras analizadas. Todas están en aproximadamente el mismo rango de precio entre los 100 – 300 dólares y todas presentan similares circuitos básicos con énfasis en uno u otro periférico. Lo importante es poder asimilar la diversidad para poder construir bases

sólidas que contribuyan a la formación del conocimiento requerido en los alumnos que cursen la materia de Microcontroladores.

1.3 PLATAFORMA DE TRABAJO PROPUESTA POR EL PRINCIPAL FABRICANTE DE MICROCONTROLADORES PIC

La tarjeta PIDE4 [4] presenta facilidades para microcontroladores tipo flash de 8, 14 y 18 pines de las familias 12, 16 y de la familia 18 incluyendo aquellos que poseen como característica la tecnología nanowatt utilizada para la optimización del uso de la energía. Una operación de bajo consumo de energía es posible mediante la utilización de un condensador de gran capacidad y conmutadores que permiten deshabilitar parte del hardware para eliminar innecesarios consumos de energía. La tarjeta incluye también un zócalo para un oscilador, cristal o circuito RC. Tiene además un regulador que obtiene 5 voltios a partir de una entrada de 9 voltios. También posee la capacidad de conectarse a una fuente regulada de 100 miliamperios a 5 voltios.

Entre otras de sus características se puede apreciar un conector ICD que permite la programación serial en el propio circuito sin necesidad de sacar el componente su de su zócalo; para lo cual deberá colocarse un programador externo el recomendado es el Programador en Circuito ICD2 de Microchip. También posee en su arquitectura un conector RS232, un display LCD de cristal líquido

2x16, y diferentes espacios para la incorporación de memorias EEPROM, un puente H, transceptor LIN y un área de prototipos. Ver la figura 7.

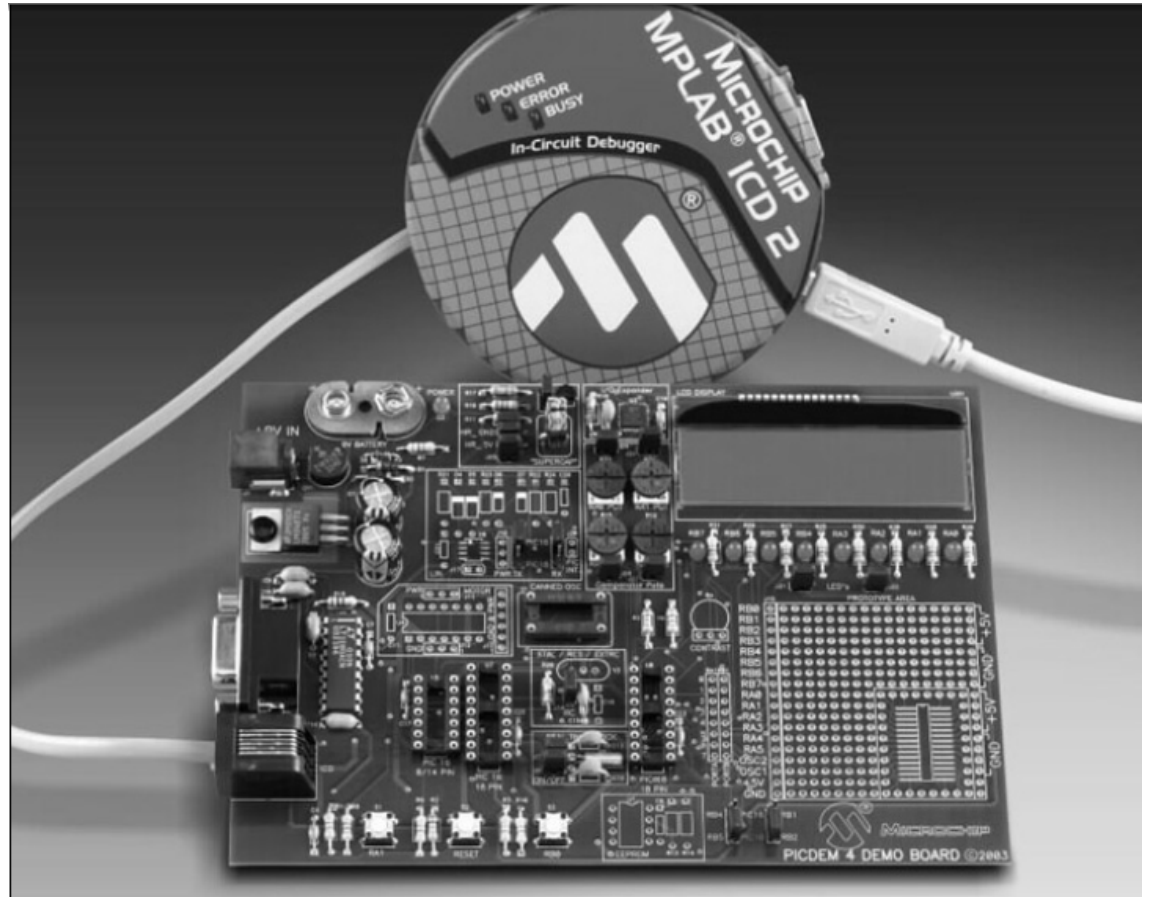


FIGURA 7: TARJETA PICDEM 4 CONECTADA CON EL ICD 2 DE MICROCHIP

Sus principales prestaciones pueden resumirse en las siguientes líneas:

- Interfase RS232
- LCD 2x16
- Conector ICD para programación en serial en circuito.
- 8 LEDs, 4 potenciómetros y 3 pulsadores,
- Acondicionamiento listo para alojar EEPROM, puente H para manejo de motores y transceptor LIN.

- Osciladores de cristal o RC
- Soporta adaptador de 8 voltios o fuente regulada de 5 voltios a 100mA,
- Área para prototipo
- Conectores de expansión.

1.3.1 ANALISIS DE SUS PRESTACIONES

La tarjeta PICDEM 4 presenta un entorno de trabajo similar al del Lab-X1 en el sentido de su apariencia con display LCD, LEDs, potenciómetros, área para prototipos. Sus diferencias fundamentales radican en la cantidad de pulsadores que es muy inferior. Aquí se presentan varios zócalos para microcontroladores mientras que en el Lab-X1 hay un solo conector Ziff que soporta micros de hasta 40 pines. Lo cual no es posible en esta nueva tarjeta.

La fortaleza de la tarjeta PICDEM 4 radica en la presentación de la utilización de características avanzadas en microcontroladores como las de la tecnología nanowat. Así como también el análisis de redes LIN muy utilizadas en el ambiente automotriz para interconectar distintos sensores y dispositivos de control de una forma sencilla y económica mediante un solo cable. Pudiendo este tipo de comunicación implementarse por software en patitas disponibles de cualquier microcontrolador.

Observando las características generales de ambas tarjetas, es evidente que para el curso actual de Microcontroladores la que mejor cubre las necesidades prácticas es la tarjeta Lab-X1.

CAPÍTULO 2

PROPUESTA DE CONCEPCIÓN DEL SISTEMA MODULAR

2.1 INTRODUCCIÓN

El análisis de la diversidad de entorno de trabajo, induce a pensar en una alternativa viable tanto en lo económico como en lo práctico para la enseñanza de microcontroladores PIC. Lo que este autor concibe como un sistema modular es todo un proceso adaptable al incesable avance tecnológico, que permite a la universidad o centro de estudios ir de la mano con los adelantos en este campo de estudio. Sin restringirse a un solo grupo de prácticas sino que será posible diversificar dentro de la rigidez de un programa de estudio. Por qué se habla de rigidez porque el programa de estudios especifica una práctica sobre un tópico general y sobre ese tópico es posible realizar múltiples experiencias que se verían beneficiadas con la existencia de un hardware específico para cada una de ellas. Un entorno de trabajo que focalice los conceptos empleados y que no se disgregue o apantalle la funcionalidad de los dispositivos.

Como se ha podido observar las plataformas de trabajo propuestas por diversas empresas dedicadas a construir entornos de desarrollo tratan de hacer todo con el mismo hardware. Esto es beneficioso como se ha expresado anteriormente para el docente quien ante la estrechez de tiempo opta por la vía rápida para la elaboración de su circuito con microcontroladores Pic. Aquí la tarjeta multipropósito

viene bien y ayuda enormemente. Para el estudiante que se inicia es más adecuado algo más sencillo en donde pueda apreciar el trabajo dedicado de un microcontrolador a una tarea específica.

De aquí nace la idea de la modularidad circuitos más sencillos, dentro de la complejidad de un entorno de desarrollo comercial, que permitan enfatizar las características del diseño estudiado. Aquí va a tener un papel preponderante el nuevo equipamiento para la fabricación de circuitos impresos con el que contará la ESPOL en este año, será posible ir enriqueciendo la plataforma modular que crecerá al ritmo del avance de la ciencia que a su vez deberá irse reflejando en los programas de estudios para dotar al estudiantado de los conocimientos básicos necesarios para su adecuado desempeño en su futuro profesional.

La plataforma que se concibe no es un grupo de maquetas fijo inflexible sino módulos que en forma escalonada van formando al estudiante a través de experiencias didácticas enriquecedoras en las que utilizarán sus manos para tocar componentes su inteligencia para visualizar una aplicación, realizar un diseño, plasmarlo en un diagrama, probarlo en un simulador, llevarlo a una placa de circuitos impresos y analizar el comportamiento de su diseño, hecho con sus manos y cerebro. Esto es una experiencia inolvidable que perdura y gratifica con el conocimiento de lo vivido y experimentado que servirá como punto de partida hacia una fructífera carrera.

Actualmente podemos observar a muchos de nuestros ex-estudiantes en los que ha quedado sembrada la inquietud del conocimiento de los microcontroladores. Quienes no dejan de investigar y desarrollar sus propios circuitos, ya sea por propio placer, por cumplir con proyectos de otras materias o por alguna aplicación comercial con lo que se ayudan económicamente en sus estudios y vida personal. Con el pasar del tiempo se ha visto también que el interés de los estudiantes en lugar de decrecer aumenta por lo menos entre aquellos que todavía cursan materias es la ESPOL.

Esto ha dado como resultado un fenómeno interesante y es el hecho que por la facilidad de su implementación y por la amplitud de su campo de acción los estudiantes prefieran la utilización de microcontroladores en sus trabajos de tesis o en proyectos relativos a las materias que están tomando. Dada esta circunstancia se hace muy importante en la formación del futuro profesional el diseño y utilización de circuitos impresos. Observada esta tendencia, se pensó en la adquisición de una máquina para la fabricación de circuitos impresos. Lo cual es ya una realidad que empezará a funcionar dentro de muy poco tiempo. Esto contribuirá positivamente incrementando el interés de los estudiantes y la necesidad de muchos de los profesores de la FIEC de poder utilizar en sus proyectos circuitos prácticos.

Con el propósito de englobar lo anteriormente expuesto se enmarcó el programa de estudios en temas de trabajo lo suficientemente amplios que permitan bajo un título general abarcar varias prácticas. Este sistema permite la construcción por parte de los estudiantes de los módulos de trabajo que el autor Viene mencionando a los largo de esta tesis. Es decir los módulos no es algo fijo como muchos podrían pensar sino que son muchas partes en las que intervienen activamente los estudiantes con la conducción de sus maestros.

Este tipo de enseñanza es la que difícilmente se olvida y perdura dentro del bagaje de conocimientos adquiridos. Cuando en la vida profesional tengan necesidad de realizar circuitos impresos podrán desempeñarse con solvencia conociendo además que en la ESPOL cuentan con la capacidad instalada para la realización de placas de excelente calidad, lo cual hasta el momento no ha sido posible obtener en la ciudad de Guayaquil. Esto que se dice es muy cierto y para construir una placa decente ha habido necesidad de recurrir a empresas instaladas en las ciudades de Quito y Cuenca. La idea detrás con respecto a la facilidad del nuevo equipamiento en la ESPOL es la de poder ir avanzando en la tecnología y poder ir dotando de más opciones como por ejemplo la confección de placas de tres o más capas y ya no tan solo de dos como será al inicio. También se puede avanzar en el montaje de circuitos impresos y en

mecanismos para la adquisición de componentes en forma oportuna mediante una adecuada planificación.

2.2 ELEMENTOS DE HARDWARE MÁS UTILIZADOS EN LA INDUSTRIA

Con referencia al tema es necesario precisar que el enfoque es respecto a lo relacionado con los microcontroladores. De lo contrario la cobertura sería muy amplia y se saldría del ámbito de esta tesis. Tampoco se quiere hacer referencia a todos los elementos discretos sino a ciertos circuitos específicos y elementos que se encuentran con mayor frecuencia cuando se trabaja con microcontroladores. Su identificación ayudará a definir aquello sobre lo cual se deberá hacer énfasis en los planes de estudio.

Tampoco es sencillo mencionar los circuitos con microcontroladores que funcionan en la industria pero si se podría hablar en forma general que dichos circuitos contienen: sensores, temporizadores, interruptores, botoneras, displays de siete segmentos, displays de leds, teclados displays LCDs, memorias EEPROM, etc.

Sensores que permiten la captación de señales que luego de pasar por un circuito de acoplamiento es posible ingresarlas a los microcontroladores para su procesamiento.

Temporizadores que permiten realizar tareas según un orden determinado. En los microcontroladores de la gama media el timer TMRO es un excelente ejemplo de temporizador y se hace imprescindible su estudio. Cuando se temporiza con el TMRO hay que aprender el uso del registro OPTION_REG con el significado de cada uno de sus bits. Es decir que se topan los conceptos de preescalador, selección del TMRO o del watch dog timer, habilitador del ingreso de pulsos externos, indicación de su flanco, del flanco de las interrupciones y de la habilitación o no las resistencias de pull up en el puerto B.

Los circuitos con interrupciones están presentes en la mayoría de aplicaciones. Con ellos se relacionan íntimamente las subrutinas que deben de generarse para atenderlas. En ese momento entra también en juego el concepto de stack o pila para poder llevar las direcciones de retorno una vez atendida la interrupción.

Cuando se da servicio a los teclados se involucra el manejo de retardos para circuitos antirrebotes que no permitan el ingreso de falsos datos. El manejo del software destaca también su importancia porque hay que utilizar una gran cantidad de algoritmos que permitan la selección de filas y columnas para poder realizar un escaneo de cada una de las teclas.

El manejo de los LCDs requiere el aprendizaje de circuitos con otros microcontroladores propios que mantienen un sistema operativo diferente al del microcontrolador principal. Esto permite el estudio detallado de las instrucciones del microcontrolador manejador LCDs para poder utilizarlo adecuadamente obedeciendo y siguiendo sus instrucciones y comandos. Las configuraciones que se utilizan para conectar los LCDs son diversas pudiendo ser en paralelo o en serie. Los LCDs a su vez pueden conectarse a cuatro hilos u ocho hilos. Los LCDs que se estudiarán en el laboratorio contienen un máximo de diez y seis caracteres visibles (de un total de 40) distribuidos en dos líneas (2 x 16).

El uso de botoneras e interruptores así como el de LEDs y lámparas de siete segmentos se ha generalizado para múltiples circuitos. Para los pulsadores hay que tener especial cuidado de aprender a utilizarlos con circuitos antirrebotes ya sea por software o por hardware. Cuando se los hace por software el tiempo aconsejado es de aproximadamente unos quince milisegundos. Cuando se trata de los display es importante tener en cuenta los tipos de conversión que se hagan para poder presentar en forma adecuada. El manejo de tablas con las instrucciones adecuadas que permiten una cantidad de transformaciones facilita el manejo de la información.

El empleo de las memorias EEPROM reviste gran importancia ya que su uso es generalizado para el almacenamiento de información

que quiere ser mantenida aún en caso de corte de energía. Se emplean con frecuencia en circuitos para activación de claves o mantener datos críticos de algún proceso o valores de seteos iniciales de un equipo. Es importante destacar que para la grabación de memorias de EEPROM se utiliza un proceso especial indicado en las hojas de especificaciones de los microcontroladores, para evitar falla en este delicado proceso. La lectura y escritura de las memorias EEPROM toma varios milisegundos lo que contrasta con la velocidad habitual de pocos microsegundos a la que corren los microcontroladores. Los microcontroladores para su memoria de programa utilizan memorias tipo FLASH que son memorias muy rápidas que puede mantener la información cuando se apaga el circuito.

Otro aspecto destacable de los circuitos con microcontroladores son sus diversas interfases. La más básica y que se ha venido usando por mucho tiempo la interfase RS232. La mencionada interfase actualmente está cayendo en desuso, a tal punto que las nuevas computadoras ya no traen un conector para alojarla. La interfase paralela es otra que va en camino a la extinción y que también ha sido utilizada por mucho tiempo . Ambas interfases se las puede seguir utilizando mediante el empleo de un convertidor de USB a serial o de USB a puerto paralelo. La nueva interfase USB ha tomado un rol preponderante especialmente por su rápido manejo de la información. Resumiendo recalamos que las nuevas

computadoras ya no vienen con puertos seriales, ni puertos paralelos y en su lugar traen varios puertos USB. Si se quiere reproducir un puerto paralelo o un puerto serial, habrá que utilizar uno de los cables convertidores mencionados con anterioridad. Se destaca entonces la importancia del estudio de la interfase USB cuyos principios deben ser entendidos para poder explicarse el flujo de la información.

Vale destacar que las comunicaciones con microcontroladores se las puede hacer de múltiples maneras. Siendo las mas utilizadas la comunicación serial, ya sea a través de la interfase RS232 o vía USB. Igualmente existen muchos otros protocolos que permiten comunicaciones entre diversos dispositivos destacándose el I2C, el CAN y el SPI.

2.3 HABILIDADES PROFESIONALES A FORMAR EN EL LABORATORIO DE MICROCONTROLADORES

Muy importante de tener en cuenta son las habilidades que se quieren formar en los estudiantes a su paso por el laboratorio de Microcontroladores. Para un acertado enfoque de la enseñanza en cualquier carrera resulta significativo el tomar en cuenta las habilidades profesionales a formar en el ingeniero y expresarlas dentro del modelo profesional. Múltiples especialistas han dedicado esfuerzos al estudio de la formación de habilidades profesionales en estudiantes de diversos niveles de aprendizaje, abordando estos

estudios desde diversos planos como el psicológico, didáctico y metodológico.

En el tratamiento de estos temas han surgido una variedad de criterios proporcionando una multiplicidad de acepciones que definen las habilidades profesionales.

Una de las concepciones más cortas y claras que tratan de definir la habilidad profesional está dada por Danilov que la define como "conocimiento en acción" [5].

Para el Doctor en Ciencias Carlos Álvarez Zayas "la habilidad es un elemento del contenido y expresa en un lenguaje didáctico un sistema de acciones y operaciones para alcanzar un objetivo" [6].

Para los Doctores en Ciencias: Homero Fuentes González, Ulises Mestre Gómez y Faustino Repilado Ramírez, "la habilidad es el modo de interacción del sujeto con el objeto, es el contenido de las acciones que el sujeto realiza, integradas por un conjunto de operaciones que tienen un objetivo y que se asimila en el propio proceso" [7].

Cada individuo asimila el conocimiento de acuerdo a su capacidad y su éxito en el desarrollo de sus actividades depende de la forma en

que las haya asimilado. Las habilidades y los hábitos son formas diferentes de asimilación de la actividad.

En el proceso docente-educativo, constituye un propósito fundamental la formación y desarrollo de habilidades y hábitos en los estudiantes. Pero hay que tener en cuenta que existen una gran divergencia acerca de la naturaleza de estos temas, así que en esta tesis, tomando como referencia bases conceptuales de diversos autores y basados en la experiencia propia aplicada al tema específico, se destacarán la formación de habilidades derivadas al aprendizaje dentro del Laboratorio de Microcontroladores.

El desarrollo de habilidades contribuye a la formación psicológica de la personalidad. Así como se habla de las habilidades de los estudiantes también debe de hablarse de las habilidades docentes. Cuando nos referimos a los maestros sus habilidades docentes son elementos inseparables de su actividad profesional. Es decir las habilidades constituyen una componente importante dentro del proceso docente-educativo tanto en la formación de nuevos profesionales como en la propia habilidad profesional del docente para transmitir, enseñar, formar influenciar en la formación de habilidades hábitos específicos en el estudiantado que vayan forjando adecuadamente su personalidad para enfrentar problemas relativos a su futura profesión.

El término habilidad independientemente de las diferentes acepciones que se encuentran en la literatura psicológica y pedagógica moderna y contemporánea se puede identificar con la expresión “saber hacer”.

La determinación de habilidades, como se comprenderá es anterior a la definición de cualquier plataforma de trabajo, esto obligó en esta investigación a encuadrar en este estudio aquellas que se enmarquen dentro del programa de estudios de la ESPOL y con las características propias de las actividades que los estudiantes llevan a cabo dentro del Laboratorio como actividad académica.

En este caso particular para la asignatura de Microcontroladores, el objeto de estudio son los microcontroladores y la capacidad que deben adquirir los estudiantes es la de utilizar al microcontrolador para dar solución a problemas inherentes a la profesión.

Al hablar de capacidad se está hablando de un conjunto de habilidades y conocimientos que deben ser adquiridos por parte de los estudiantes para que en un futuro le permitan: diseñar circuitos, simular y armar circuitos con microcontroladores que puedan conectarse con dispositivos tanto digitales como analógicos, estudiados en diversas asignaturas.

Este proceso de formación del estudiante estaría inconcluso sino de se desarrollan en él las habilidades propias de la construcción y puesta en marcha de los circuitos diseñados.

En la actualidad la mayor parte de especialistas en pedagogía coinciden en la importancia del papel de los objetivos planteados dentro de una asignatura en la dirección del proceso enseñanza aprendizaje. Para lo cual dichos objetivos deben de expresar con precisión las aspiraciones deseadas en el proceso educativo en marcha.

El objetivo principal del Laboratorio de Microcontroladores es que los estudiantes diseñen, simulen y construyan circuitos con microcontroladores, basados tanto en software como en hardware, que les permita dar solución a problemas relacionados con la automatización de procesos y al mismo tiempo ser capaces de la puesta en marcha y seguimiento de su correcta operación.

Tomando en cuenta las reflexiones expuestas y el conjunto de actividades dentro del proceso educativo-formativo del estudiante pasando por etapas que van desde la presentación de un problema a solucionar, el diseño de alternativas de solución, la selección de componentes, la simulación, la confección de la placa de circuitos impresos, la prueba de funcionamiento hasta la puesta en marcha y prueba final. Se deben de formar en los estudiantes habilidades

profesionales que les permitan llevar a cabo las actividades mencionadas. Las mismas que han sido resumidas en la tabla siguiente (Ver la tabla 1):

No.	HABILIDADES
1	Habilidad para diseñar circuitos con microcontroladores basados tanto en hardware como en software
2	Habilidad para simular este tipo de circuitos
3	Habilidad para la construcción de prototipos
4	Habilidad para la puesta a punto y operación sostenida de sus circuitos

TABLA 1: HABILIDADES QUE DEBEN FORMARSE

2.4 EXIGENCIAS QUE DEBE CUMPLIR EL SISTEMA MODULAR PROPUESTO

Como Sistema Modular se definirá al conjunto de elementos que servirán de apoyo en la tarea docente con el objetivo específico de la enseñanza práctica en el Laboratorio de Microcontroladores. Observando en forma ordenada la concepción sobre la formación de

habilidades en los estudiantes, siguiendo el esquema práctico establecido en el programa del laboratorio de la materia donde los estudiantes tomarán acciones tendientes al cumplimiento de sus tareas asignadas. Para lo cual utilizarán según la concepción de esta tesis un sistema modular estrechamente ligado con el propósito de formación de habilidades profesionales. En un armonioso ambiente en donde concurren no sólo elementos de hardware y software sino donde también se ha tomado en cuenta la parte humana para dotar al estudiantado de la comodidad, claridad, buen trato, atención a sus preguntas, acceso a material complementario, que hacen del paso por el Laboratorio de Microcontroladores una experiencia grata que forja habilidades que lo acompañarán en su desempeño profesional. Lo anteriormente expresado se lo esquematiza en la figura 8. Donde se prioriza la formación de habilidades mediante acciones tomadas algunas de las cuales se van convirtiendo en hábitos y todo esto con el sistema modular propuesto en esta tesis.

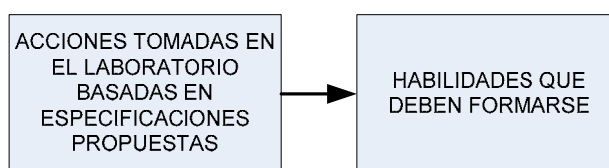


FIGURA 8: ELEMENTOS RELACIONADOS CON EL SISTEMA MODULAR

Los elementos de la figura 8 se los pudo haber enmarcado dentro de un cuadro más grande con el nombre de ambiente de trabajo

sobre el que el autor ha tenido especial cuidado y para lo cual ha recibido el debido apoyo de las autoridades de la FIEC.

Sobre la importancia de la formación de habilidades en el estudiantado ya se ha hecho un planteamiento claro incluso en el tema anterior se presentó un cuadro resumen (Tabla 1) de la principales habilidades que deben impulsarse. En la figura 8 que se acabó de mostrar se presentan un cuadro de acciones que inducen al estudiantado mediante el uso del sistema modular que se propone en esta tesis a la formación de hábitos y habilidades que lo acompañarán en su desempeño futuro.

Es decir el modo de actuar o el comportamiento ordenado del estudiantado dentro del Laboratorio son parte de las acciones para lograr la formación de habilidades profesionales. Este comportamiento se lo induce a través del desarrollo de cada práctica que se la elabora con el cuidado necesario para cumplir con su objetivo. Las tareas o acciones emprendidas por el estudiante a medida que avanza en el cumplimiento de sus metas se va subconscientemente automatizando, llegando a construir hábitos de trabajo cuando enfrenta problemas similares.

Acompañando a su accionar, el estudiante deberá contar con las herramientas de trabajo tanto en software como en hardware que se necesitan para el cumplimiento de las tareas enfrentadas en la

práctica respectiva. Una vez más se puede apreciar la estrecha relación de cada elemento en la correcta formación de habilidades utilizando las características del sistema modular que se propone.

En la práctica de laboratorio se presenta al estudiante una problemática que sabe debe ser resuelta mediante el uso de microcontroladores como elemento central de la solución. A partir de esta situación el estudiante debe ser capaz con su razonamiento electrónico de definir los recursos que necesitará para encontrar la solución correcta. Entendiéndose por recursos las subrutinas, los circuitos y componentes electrónicos del entorno del microcontrolador. Determinados los recursos deberá observar los recursos con los que cuenta y cuales deberá obtener y agregar a su diseño.

Esta primera etapa es la de preparación mental para enfrentar el problema a resolver. En este punto es importante que el estudiante sepa relacionar los recursos propios del microcontrolador con la utilización que podrá darle en la solución de problemas planteados en el Laboratorio.

En la siguiente etapa el estudiante acomete la tarea del diseño del circuito que solucionará el problema propuesto. Por las características expresadas los circuitos en base a microcontroladores están conformados por un hardware del que el

microcontrolador es el actor principal acompañado de un software que debe ser desarrollado de acuerdo con las exigencias del problema planteado. El software puede ser desarrollado en Lenguaje Ensamblador o en Lenguaje de nivel alto (como el Lenguaje C o el Basic).

A este punto el estudiante deberá efectuar las siguientes tareas:

- Analizar comparativamente las diferentes soluciones a la tarea propuesta
- Seleccionar la mejor solución

Para poder realizar su tarea dentro del tiempo asignado, es condición imprescindible de que los estudiantes puedan tener acceso a la información de manera completa y oportuna. Punto en el que se han extremado los cuidados para que esto sea posible en el desarrollo de cada práctica de laboratorio.

En lo que se refiere a libros, monografías, tutoriales, notas de aplicación, hojas de especificaciones del fabricante, son elementos que se hacen disponibles desde la clase de la materia teórica pero que sin embargo están siempre disponibles en red como material de consulta. Pero en cada práctica debido a circunstancias de tiempo se optimiza el proceso de la información para que esta esté disponible en forma oportuna y no tenga que perderse el tiempo que está ya dosificado hacia el objetivo específico de cada tarea.

Para escribir el programa se utiliza un editor de texto teniendo en cuenta al momento de la grabación del programa de hacerlo con la extensión correcta, .asm para el caso de un programa en Lenguaje Ensamblador y .c para el Lenguaje C.

Para revisar el programa fuente escrito deberá hacerse uso del compilador respectivo con los programas de simulación adecuados hasta que esté libre de problemas de compilación (usando compiladores) y de fallas en la ejecución (mediante el uso de simuladores). Entendiéndose que los simuladores no son la ejecución real sino una aproximación que puede tener limitaciones inesperadas particular sobre el cual hay que estar consciente al emprender en el desarrollo de una aplicación.

Existen simuladores gráficos que tienen la facultad de presentar un ambiente en el que se pueden colocar los elementos constitutivos del circuito, conectarlos mediante cables y simular su comportamiento. El editor gráfico permite observar el comportamiento del diseño bajo ciertas restricciones y limitaciones. Sin lugar a dudas estas herramientas constituyen un apoyo muy valioso en la implementación de circuitos. Hay que estar conscientes de las limitaciones que tienen estos programas ya que no es posible conectar todos los elementos que uno quiera ni realizar todas las conexiones con dispositivos externos que se quisieran. Es decir hay

que aprender a usar correctamente las herramientas de desarrollo, las mismas que están en un constante avance evolutivo que obliga a la docencia a un permanente esfuerzo de actualización para mantener la pertinencia de la materia tratada.

Una vez concluida la etapa de simulación se deberá proceder al uso de los módulos preparados para la realización de cada práctica. Para lo cual se procederá a grabar el microcontrolador mediante el uso de un programador apropiado, para luego insertarlo en el zócalo del módulo de hardware respectivo y observar su comportamiento real. Cualquier discrepancia con el objetivo final será objeto del debido análisis y evaluación hasta lograr su corrección. Entendiéndose que para hacer cambios en la programación habrá que recorrer nuevamente varias de las etapas mencionadas con anterioridad.

Para lograr el correcto funcionamiento o para observar el comportamiento del circuito diseñado el estudiante podrá hacer uso de las herramientas normales del laboratorio como osciloscopio, multímetro, fuentes de poder, etc.

CAPÍTULO 3

DESARROLLO DEL SISTEMA MODULAR

3.1 DIFERENTES ELEMENTOS QUE CONFORMAN LA PLATAFORMA

Hasta el momento se han analizado algunos de los elementos del Sistema Modular, ahora es el momento de presentarlo en su conjunto. La forma como fue ideado como producto de la investigación realizada y la experiencia docente.

El Sistema modular que se describe, no es un conjunto de tarjetas aisladas, tampoco un software o un hardware específico. Es un conjunto de elementos que permiten cumplir con el objetivo específico de formar hábitos y habilidades de trabajo relativos al diseño de circuitos con Microcontroladores en los estudiantes que cursan el Laboratorio. La presentación que hoy se hace del Sistema Modular en mención bajo ningún punto de vista es una tarea terminada, porque siempre surgen nuevas ideas y habrá algo nuevo que incorporar. Es decir que no es un sistema rígido, inflexible, al contrario es totalmente abierto, actualizable de acuerdo con los avances científicos ya sea incorporando nuevos circuitos o nuevos sistemas de enseñanza.

Una de las dificultades con las que se ha enfrentado el estudiante politécnico es la de confeccionar placas de circuitos impresos. Problema que no sólo es de los estudiantes sino también que se presenta a nivel profesional. Son muchas las oportunidades de trabajo que se pierden por no contar con un laboratorio adecuado para la confección de circuitos impresos profesionales, de manera rápida, con una facilidad cercana y sobre todo con mínimos márgenes de calidad. Los sitios más próximos en la ciudad de Guayaquil son talleres artesanales que han venido funcionando durante varios años pero sin mejorar sus estándares de calidad. En tal forma que una placa de este tipo es dudosamente confiable ya que no cuentan ni con los procesos ni los materiales que permitan obtener la calidad requerida en rigurosos ambientes industriales.

Por otro lado en las ciudades de Quito y Cuenca si hay facilidades que prestan estos servicios con buena calidad y también desde hace varios años. Surge aquí la pregunta ¿porqué en Guayaquil no se ha desarrollado adecuadamente esta actividad, sabiendo que se encuentra en plena vigencia en dos de las tres ciudades más influyentes del Ecuador? La misma pregunta surgió con mayor fuerza hace aproximadamente cinco años entre profesores de varias áreas de la FIEC y se pensó en un plan a largo plazo para adquirir un equipo que permitiera realizar esta actividad en forma profesional. El mismo que llenaría un vacío que la actividad privada no ha sido capaz de cubrir. Posiblemente uno de los motivos por los

que no se ha desarrollado esta actividad se deba al mismo sistema actual que por dificultades propias del proceso no le ha impregnado el dinamismo necesario.

Se decía que la tarea del Sistema Modular no está terminada en el sentido de que el trabajo continúa inclusive multiplicándose en forma muy dinámica. Ya con el equipo para la realización de circuitos impresos que estará en servicio en pocos días, deberá empezar una estrategia en conjunto de todos los profesores de la ESPOL para dinamizar su utilización en beneficio de la comunidad politécnica y nacional. Este equipo aparte de llenar un vacío en el ámbito politécnico deberá servir a la comunidad en general. Sus servicios deberán estar asequibles a cualquiera que los requiera con el único objetivo de dinamizar esta actividad en la ciudad de Guayaquil.

Son muchos ya varios cientos de estudiantes los que han cursado el laboratorio de Microcontroladores, en los que se han forjado habilidades que podrían alcanzar una mayor expresión al saber que se cuenta con el equipamiento adecuado. A este punto muchos se preguntarán ¿Para qué un equipo sofisticado para fabricar circuitos impresos si los artesanales han venido trabajando? La respuesta es cierto que han venido trabajando pero con serias limitaciones. Su durabilidad tanto mecánica como eléctrica está muy limitada. Además no se pueden cumplir con las exigencias de modernos diseños en cuanto a tamaño de pistas y componentes, no se diga de

velocidad. El equipo que está adquiriendo la ESPOL no es tampoco el más óptimo de todos. Pero si uno con especificaciones básicas de calidad, que servirá de punto de partida con posibilidades abiertas de crecimiento de acuerdo al desarrollo que se logre con el esfuerzo politécnico y de la comunidad en general.

Se hace esta introducción para dejar ver que la presentación del Sistema Modular motivo de esta tesis, no es una tarea aislada sino que requiere de mucho trabajo y preocupación sobre varios detalles para que pueda funcionar en forma armónica. Formando habilidades en los estudiantes politécnicos para lograr un dinamismo en el diseño electrónico con microcontroladores encaminados a la resolución rápida de problemas que se presentan en el sector industrial, en la actividad económica diaria, en servicios varios. El desarrollo que se impulse generará un efecto positivo multiplicador de trabajo y empleo que beneficiará a la comunidad y potenciará la prestación de servicios.

Y el sistema modular, ¿Qué tiene que ver el equipo para la fabricación de circuitos impresos con el Sistema Modular? Muy sencillo fue pensado como parte de su propia dinámica. Como parte de su actividad curricular el estudiante deberá realizar dos proyectos que requerirán de dicho equipo los mismos que se pueden apreciar en la tabla 2.

PRÁCTICA No.	DESCRIPCIÓN
6	Presentación de Proyecto 1 (en Lenguaje ensamblador)
12	Presentación de Proyecto 2 (en Lenguaje ANSI C)

**TABLA 2: PROYECTOS QUE REQUIEREN DE
IMPLEMENTACIÓN FÍSICA EN EL LABORATORIO DE
MICROCONTROLADORES**

Retomando lo que se manifestaba con anterioridad el Sistema Modular se compone de varios elementos entre los que destacamos:

1) Un ambiente de trabajo apropiado. 2) Las acciones que deben tomarse en el Laboratorio para la formación de habilidades inducidas en un adecuado diseño de Prácticas de Laboratorio (que serán tratadas en el capítulo cuatro). 3) Las herramientas de trabajo tanto de software como de hardware. 4) Los diseños implementados por los estudiantes en sus dos proyectos obligatorios haciendo uso de un equipo de fabricación de circuitos impresos.

Claro está que estos esfuerzos serían imposibles si no se contara con el apoyo de las autoridades universitarias, del personal docente, de los ayudantes de laboratorio y de los estudiantes que cursan la

materia. El Sistema Modular en su conjunto puede ser apreciado en la tabla 3.

	SISTEMA MODULAR
1	Un ambiente de trabajo apropiado
2	Acciones inducidas mediante Prácticas de Laboratorio apropiadas para el desarrollo de habilidades profesionales
3	Herramientas de trabajo
4	Diseños hechos por los estudiantes haciendo uso del Laboratorio de Circuitos Impresos

TABLA 3: SISTEMA MODULAR PROPUESTO

3.2 PRESENTACION DE TARJETAS USADAS EN EL SISTEMA MODULAR

Las tarjetas que se presentan en este capítulo fueron producto del Sistema Modular utilizado, donde con el apoyo de los estudiantes se van generando las tarjetas básicas que se utilizarán en futuras prácticas de aprendizaje de nuevos grupos de estudiantes. El costo de fabricación del circuito impreso de las tarjetas oscila entre los cinco a diez dólares, excepto por una que fueron hechas en Quito y Cuenca que sobrepasan los veinticinco dólares. Los

microcontroladores utilizados que son los elementos más costosos y oscilan aproximadamente de cuatro a quince dólares. Los demás elementos discretos son de menor valor la mayoría en el orden de varias decenas de centavos de dólar. En el Apéndice A se muestran fotografías tarjetas realizadas bajo la modalidad del Sistema Modular.

3.3 SOFTWARE Y SIMULADORES UTILIZADOS

Uno de los elementos componentes del Sistema Modular son el software y los simuladores utilizados. Al respecto se hace el énfasis respectivo en el desarrollo de cada práctica, lo cual podrá apreciarse a cabalidad en el próximo capítulo.

Vale enfatizar que la validez de un Sistema Modular no está únicamente en el software utilizado, el cual indudablemente contribuye en su medida a facilitar la consecución de objetivos, sino en el conjunto de elementos que contribuyen con el todo. Habrá circunstancias que determinen en una nueva etapa de desarrollo que el software utilizado hasta el momento ha quedado atrás en prestaciones comparado con algún otro programa. Es decir hay que entender bien la dinámica del Sistema Modular que permite la adaptación, incorporación o sustitución de herramientas en un continuo estado de mejoramiento acorde con los adelantos científicos.

En el caso del compilador como se trata del estudio de los microcontroladores PIC hay que necesariamente utilizar los recomendados por su único fabricante que es Microchip. El fabricante está permanentemente actualizando su entorno de trabajo llamado MPLAB lo que obliga a un seguimiento continuo, con el correspondiente estudio e incorporación de las mejoras en beneficio del proceso educativo. El MPLAB es una plataforma de trabajo con la cual se pueden entrelazar diversos compiladores. Microchip presenta su compilador gratuito Mpasm que se utiliza para compilar los microcontroladores de la gama básica y media. Puede albergar además otros compiladores como el C18 y el C30 que no son gratuitos pero a los que se puede acceder mediante versiones de evaluación.

Hablando ahora de simuladores el propio MPLAB tiene un excelente simulador llamado MPLAB SIM que es apropiado para muchas tareas pero que se queda corto en otras por ejemplo cuando se incluyen antirrebotes para botoneras dificulta la operación paso a paso, especialmente en procesos largos de depuración. Igualmente interacciones con teclados son difíciles de manejar ya que la lógica del teclado demanda un escaneo de teclas, lo cual cuando se trabaja paso a paso es tedioso y de extremo cuidado para llevar una secuencia y no equivocarse.

Aparte del MPLAB SIM existen otros simulador que funcionan en forma independiente del MPLAB, uno de estos es el Proteus, aunque puede tener cierta vinculación de operación por medio de un pequeño programa de los conocidos como "plug in". Lo interesante del Proteus es su gran manejo gráfico con una rica librería de componentes, comprendiendo los microcontroladores más usados en la docencia como son el 16F84, 16F628, 16F877, 18F452 y el 18F4550. Este último dispositivo es un manejador del puerto USB que fue incorporado recientemente, al entorno de simulación, por la firma Labcenter Electronics en su programa Proteus.

Lo atractivo de Proteus es que en sus páginas para albergar circuitos pueden agregarse componentes con apariencia similar a los reales y conectárselos entre sí de tal manera que es muy fácil seguir su comportamiento en circuito pequeños, aumentando su complejidad para circuitos mayores. Más atractivo aún resulta el Proteus cuando se utilizan los modos de animación donde se puede apreciar en colores las representaciones de los estados binarios de los pines de los componentes. Cuando se trata de elementos interactivos como pulsadores es posible dirigir el Mouse sobre el botón y mediante un clic simular su pulsación. En el Caso de interruptores también mediante un clic es posible cambiar su estado. Para el caso de lámparas o LEDs interactivos es posible que se enciendan o apaguen de acuerdo a la lógica de programación.

Más interesante aún resulta cuando en un mismo circuito intervienen dos o más microcontroladores. Cada uno puede albergar su propio programa y desempeñar su función independiente pero coordinada en comunicación con el resto del circuito.

Por cierto el Proteus es tan sólo una pantalla gráfica que tiene como el MPLAB que conectarse a los compiladores para manejar un programa. En este sentido el ambiente del MPLAB es más cómodo de operar y lo que generalmente se hace es compilar en el MPLAB y luego simular en el Proteus, con excelentes resultados didácticos.

No se quiso decir que el Proteus no puede albergar compiladores, si puede hacerlo pero su manejo no es tan simple como el del MPLAB y su interfaz para este tipo de operación no es tan agradable. Para poder utilizar estas capacidades hay que seguir procedimientos adicionales que no son tan intuitivos como los seguidos en el caso del MPLAB. Lo importante es que conociendo la manera es posible sobrevivir sin el MPLAB.

El otro caso el de MPLAB sin Proteus también es posible sobrevivir. Aunque debemos estar concientes de las limitaciones que se nos presentarían una vez que se conoce la operatividad de ambos programas. Lo cierto es que en las prácticas de laboratorio se destaca lo meritorio de uno y otro programa y se trabaja con las mejores prestaciones de cada uno habiéndose logrado una

adecuada combinación que contribuyen en forma eficiente a los objetivos planteados en cada práctica.

Otro de los compiladores utilizados es el PICC de la empresa CCS que sigue los estándares ANSI C. El PICC tiene su propio entorno de trabajo el cual presenta la facilidad de tener a la mano los archivos de ayuda. Cuando el PICC es invocado por el MPLAB para compilar desde allí, no es posible acceder a los archivos de ayuda. Siendo el lenguaje C rico en variantes y parámetros dentro de sus funciones vale la pena trabajar desde el ambiente mismo del PICC y tener la ayuda a la mano.

Hay situaciones especiales en las que se ha desarrollado un programa con PICC y sólo se cuenta con un programador de Microchip. En este caso será necesario utilizar los recursos propios del programador si los tuviera. En el caso del programador Pickit2, este viene con un programa actualizable por medio de la Web para ampliar su cobertura de componentes, y que permite la programación en circuito (ICSP) de los microcontroladores soportados. Lo único que requiere es importar un archivo .hex y está listo para programar el dispositivo previamente reconocido por el hardware.

Mediante el uso del MPLAB también se puede importar archivos .hex desde programas como el PICC para grabar dispositivos soportados en los programadores que estén disponibles. Lo contrario en cambio no es posible es decir un programa compilado en C30 de Microchip por ejemplo, no puede ser grabado con un programador usando el PICC para importar el archivo .hex debido a que CCS, que es el fabricante del compilador, utiliza en sus programadores formatos propietarios. En tal forma que los programadores de CCS sólo pueden utilizarse para programar códigos elaborados con la herramienta propietaria PICC.

Desafortunadamente para que un investigador pueda darse cuenta de estos inconvenientes, deberá realizar una cantidad de pruebas hasta convencerse de que lo que quiere hacer no es posible y comprender que las luchas sobre un mercado es la que provocan estas aparentes incompatibilidades que dificultan la tarea del usuario. Como es lógico al fabricante no le interesa divulgar este tipo de detalles porque son impopulares y podrían ocasionar la pérdida de potenciales clientes.

3.4 FUNCIONAMIENTO DEL SISTEMA MODULAR

El Sistema Modular fue ideado como respuesta al cumplimiento de la actividad curricular demandada por la Maestría en Automatización y Control Industrial (MACI). Aproximadamente dos años se empezó a

trabajar en su diseño y se lo ha venido aplicando en la medida de las posibilidades con relativo éxito.

Su funcionamiento es sencillo. Durante el semestre en cada clase se imparte la práctica correspondiente mediante un documento en línea, donde se detalla todo lo necesario en el transcurso de la misma. Se da al estudiante una ligera explicación verbal especialmente en las primeras prácticas hasta que se acostumbre al método de trabajo.

Se pone mucho énfasis en el control de asistencias, ya que las prácticas han sido diseñadas en forma escalonada y progresiva, construyendo en cada una conocimientos que se aplicarán en las siguientes. En tal forma que la actual depende de las anteriores. Si un estudiante no asiste a una práctica se le exige que la realice para llenar su vacío de conocimiento aunque no se le acredite puntaje alguno.

El tiempo para realizar la práctica está distribuido en tal forma que las dos horas reglamentarias son utilizadas al máximo. Durante toda la clase los estudiantes que trabajan en grupos de dos pasan concentrados en su trabajo. En caso de cualquier duda está el profesor para contestarla. Como el tiempo es limitado las repuestas son siempre concretas enfocándose al tema central de la práctica. Es importante mencionar esto porque es muy fácil distraerse cuando

se trabaja con microcontroladores por la infinidad de variantes que se le pueden dar a los programas.

Cada equipo del Laboratorio está en red y tiene los servicios de Internet, lo cual permite al estudiante tener información a la mano. Ya sea consultando la Web por alguna página recomendada o accediendo a notas de clase publicadas por sus profesores.

La disposición de las mesas de trabajo está hecha para que el profesor pueda colocarse estratégicamente y divisar cada una de las pantallas de trabajo. Al inicio del semestre se entrega a cada estudiante el programa del curso y las políticas de trabajo, donde se indican las reglas que regirán durante su paso por el Laboratorio, incluyendo el sistema de calificaciones.

Muy importante es la presentación de reglas claras porque los argumentos de los estudiantes son muy buenos tratando de justificar atrasos faltas o malos rendimientos. Tampoco se puede ser inflexible y en casos de fuerza mayor se acepta justificación y se permite la recuperación de una práctica en un horario establecido por el profesor permitiéndose al estudiante optar por un ciento por ciento de la calificación para esa práctica.

Las prácticas ejecutadas en cada clase obedecen a un esquema preestablecido en el Programa de Estudio. Esto permite al profesor

el poder cambiar los detalles de su práctica para evitar ser repetitivos de un semestre a otro e ir formando una base de diferentes prácticas que se podrían utilizar sobre un mismo tema.

Recordemos que se mencionó, en el capítulo anterior, sobre dos proyectos que se encargan a los estudiantes. La idea es que estos proyectos sean implementados en hardware implicando la construcción física del circuito. Como la ESPOL no cuenta todavía con facilidades para este tipo de actividad se ha tratado de que por lo menos uno de esos proyectos sea implementado en hardware.

La situación cambiará cuando el equipo para la fabricación de circuitos impresos esté ya instalado y funcionando adecuadamente. Cada estudiante deberá diseñar su circuito, simularlo, construirlo y probarlo, ya sea en grupo o en forma individual. Esto sin duda mejorará el actual sistema de formación de habilidades profesionales en los estudiantes.

Como se puede apreciar el Sistema Modular planteado es muy dinámico, está en constante revisión y mejoramiento. Uno de los elementos que se pensó desde un principio como fundamental para su funcionamiento fue el equipo para la fabricación de circuitos impresos que será una realidad a muy corto plazo. Este hecho más una adecuada aplicación del Sistema Modular de enseñanza en el Laboratorio de Microcontroladores se espera, aunando esfuerzos

con la comunidad en general, contribuyan a un mejor desarrollo de la actividad de diseño con microcontroladores como solución a múltiples problemas que se presentan no sólo en el campo industrial sino también en la vida diaria.

CAPÍTULO 4

DESARROLLO DE PRÁCTICAS PARA EL SISTEMA MODULAR

4.1 INTRODUCCIÓN

Las prácticas a utilizarse en el sistema modular se las ha venido preparando desde hace varios semestres. El primer documento al respecto estuvo listo el 20 de Mayo del 2006. Fue en el primer término del mismo año cuando la dirección de la FIEC ordeñó la separación de la materia teórica de la práctica. Como esta medida se anunció con la debida anticipación, se aprovechó el momento para llevar adelante el proyecto de esta tesis cuya idea había surgido con anterioridad dentro del ambiente de la MACI.

Cada práctica sigue un ordenamiento preestablecido y uniforme en forma tal que cuando se termina con la primera práctica y se empieza con la siguiente, se siente la sensación de continuidad. El estudiante se va familiarizando con los procedimientos de tal manera que llegará el momento en que los va seguir de manera intuitiva.

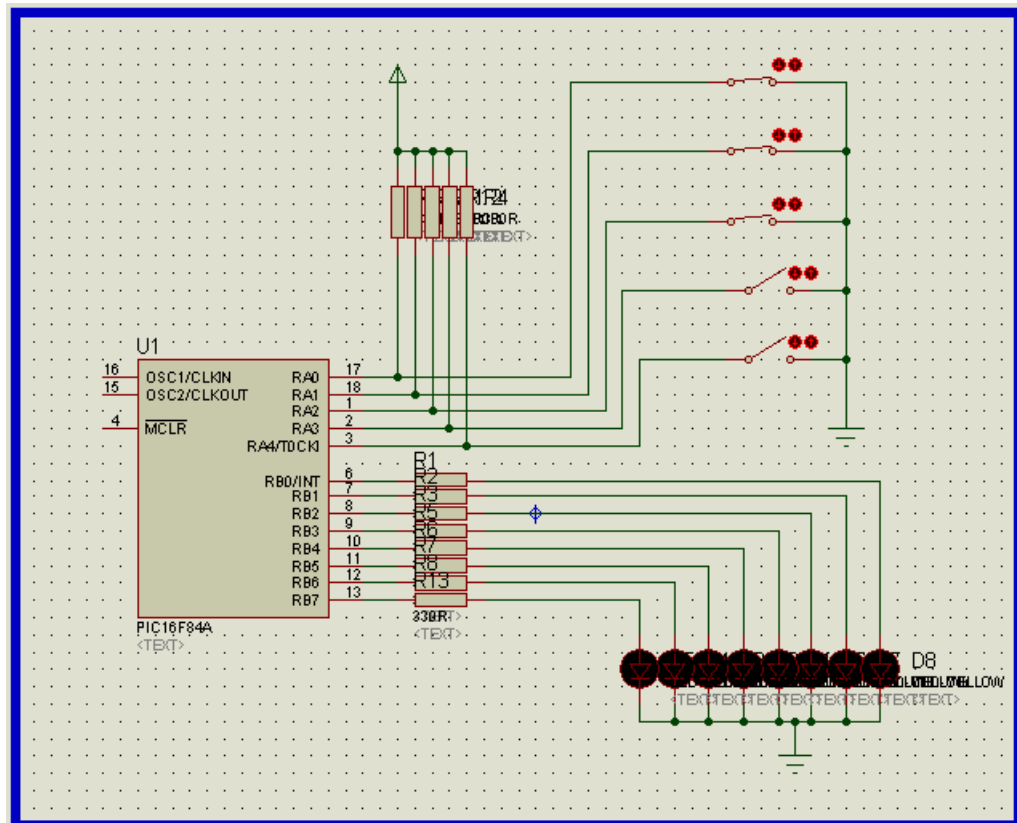
El orden, contenido y correlación con la materia teórica convierten al Laboratorio de Microcontroladores en una experiencia agradable y gratificante para el estudiante, ampliando su visión en cuando a posibles aplicaciones en el mundo práctico.

4.2 PRÁCTICAS PARA PREGRADO

4.2.1 ENCENDIDO DE LEDS MEDIANTE INTERRUPTORES

PRACTICA 1

ENCENDIDO DE LEDS MEDIANTE INTERRUPTORES



OBJETIVO:

Familiarizarse con el funcionamiento del **MPLAB** y del **Proteus**, para lo cual se utilizará un programa sencillo que permite encender LEDS en el Puerto B mediante interruptores colocados en el Puerto A.

CONTENIDO:

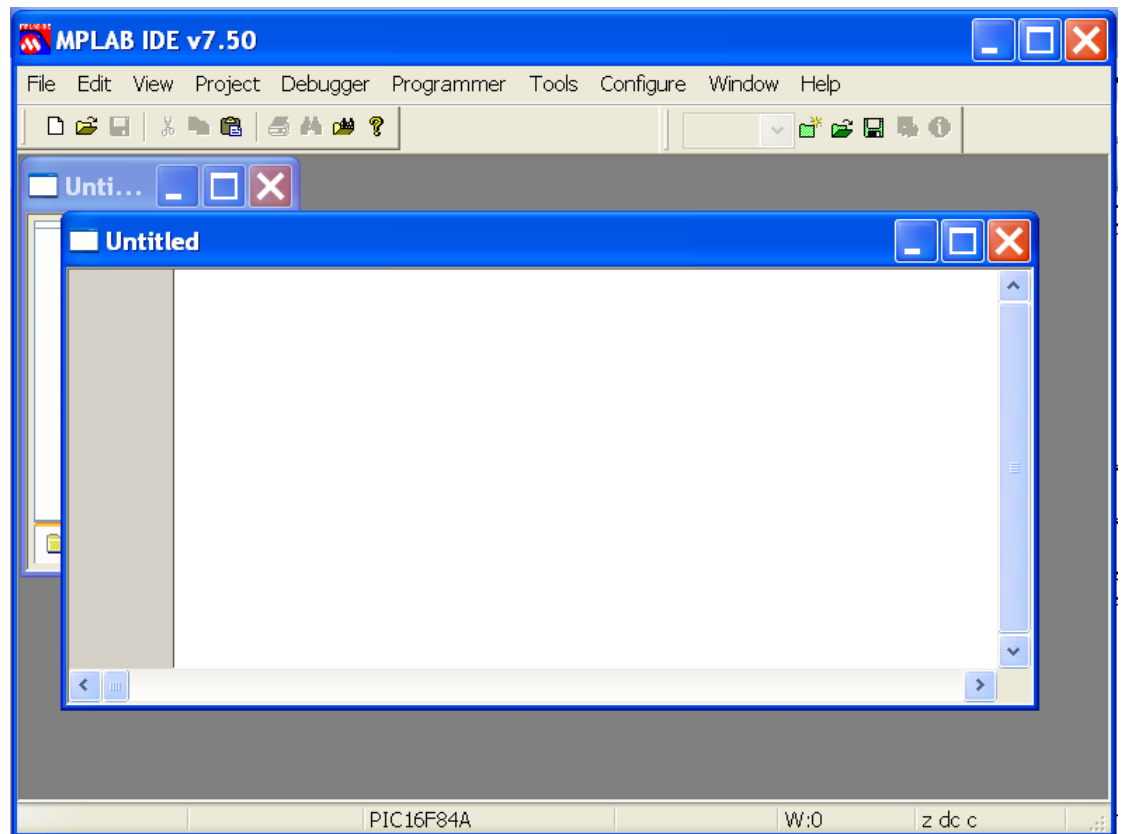
- a) CREACIÓN DE UN PROYECTO NUEVO CON EL MPLAB
- b) EJECUCION PASO A PASO
- c) OBSERVACION DE REGISTROS VARIOS
- d) GRABACION DE PIC
- e) SIMULACION EN PROTEUS

a) CREACIÓN DE UN PROYECTO NUEVO

Lo primero que se necesita es un archivo **.asm** que puede ser creado en cualquier procesador de texto o en el ambiente del **MPLAB**. Esta segunda alternativa combinada con el **Project Wizard** es la que se sigue a continuación:

1) En el directorio **C:** cree una carpeta de trabajo **G#NN**. Donde G es por Grupo # debe indicar el número de grupo y NN son las iniciales de los nombres de los integrantes del grupo. Ejemplo: G5JM pertenece al Grupo #5 de Jose y Maritza. Esta será su carpeta temporal de trabajo para la práctica del día. La misma que deberá ser borrada al finalizar la práctica.

- 2) Abra el programa MPLAB y seleccione **FILE→NEW** y aparecerá una ventana sin título.



- 3) En **Untitled** digite el siguiente programa:

```
;PROGRAMA A1
```

```
;*****
```

```
; LECTURA DE PORTA Y ENCENDIDO DE LEDS EN
```

```
; PORTB
```

```
*****
;
```

```
;
```

```
; Controla cinco interruptores conectados a RA0-RA4 y
; enciende una serie de leds en las puertas RB0-RB4
; dependiendo de las entradas introducidas por la
; puerta A. Con un 0 en la puerta A se enciende
; el led correspondiente de la puerta B.
```

```
LIST p=16F84A ; Se usa el PIC16F84
```

```
INCLUDE "P16F84A.INC"
```

```
;INICIO DEL PROGRAMA -----
```

```
ORG 0 ; Comienzo del programa (Vector de Reset)
```

```
;SETEO DE PUERTOS -----
```

```
BSF STATUS,5 ; Acceso al Banco 1
```

```
CLRF TRISB ; Puerto B como salidas
```

```
BCF STATUS,5 ; Regreso al al Banco 0
```

```
;INICIALIZACION -----
```

```
CLRF PORTB
```

```
;PARTE PRINCIPAL DEL PROGRAMA -----
```

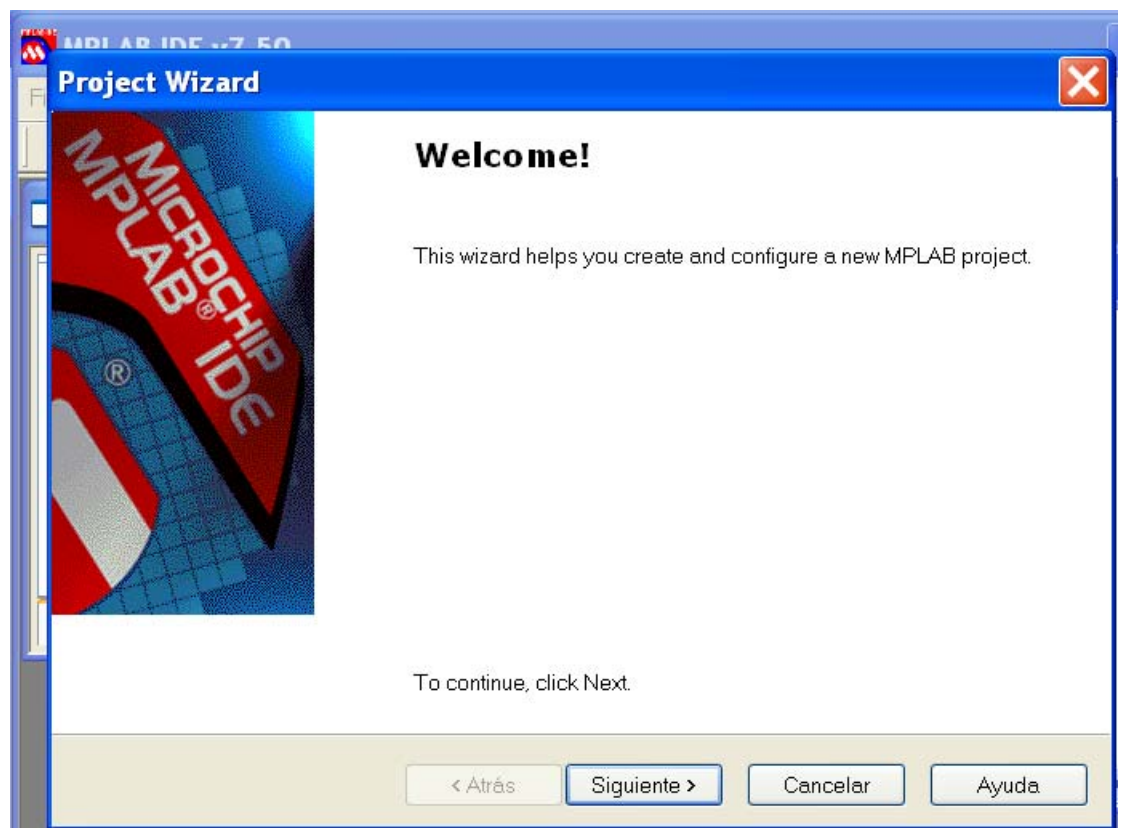
```
BUCLE
```



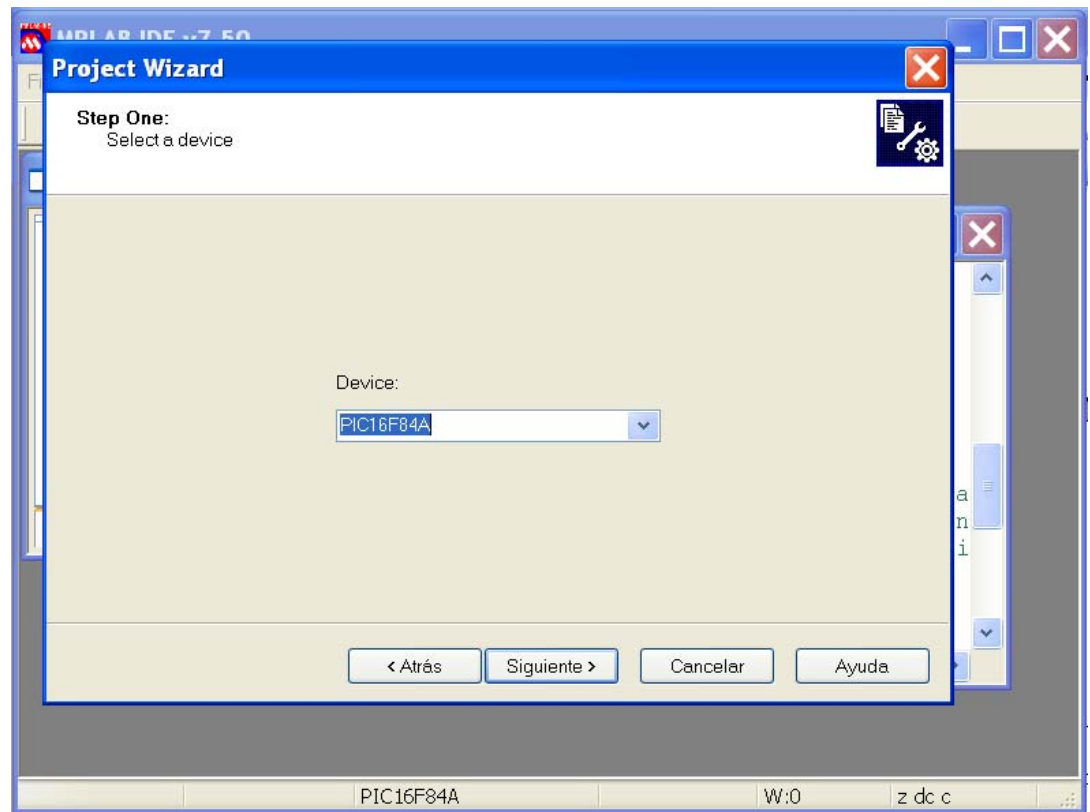
```
COMF PORTA,W      ; Complementa la entrada y la  
deposita en W  
  
MOVWF PORTB      ; El contenido de W se publica en Port B  
  
GOTO BUCLE       ; Se crea un bucle cerrado e infinito  
  
END              ; Fin del programa
```

4) Guarde el documento creado con su carpeta de trabajo con el A1NN,asm (Donde NN son las iniciales de los integrantes del Grupo).

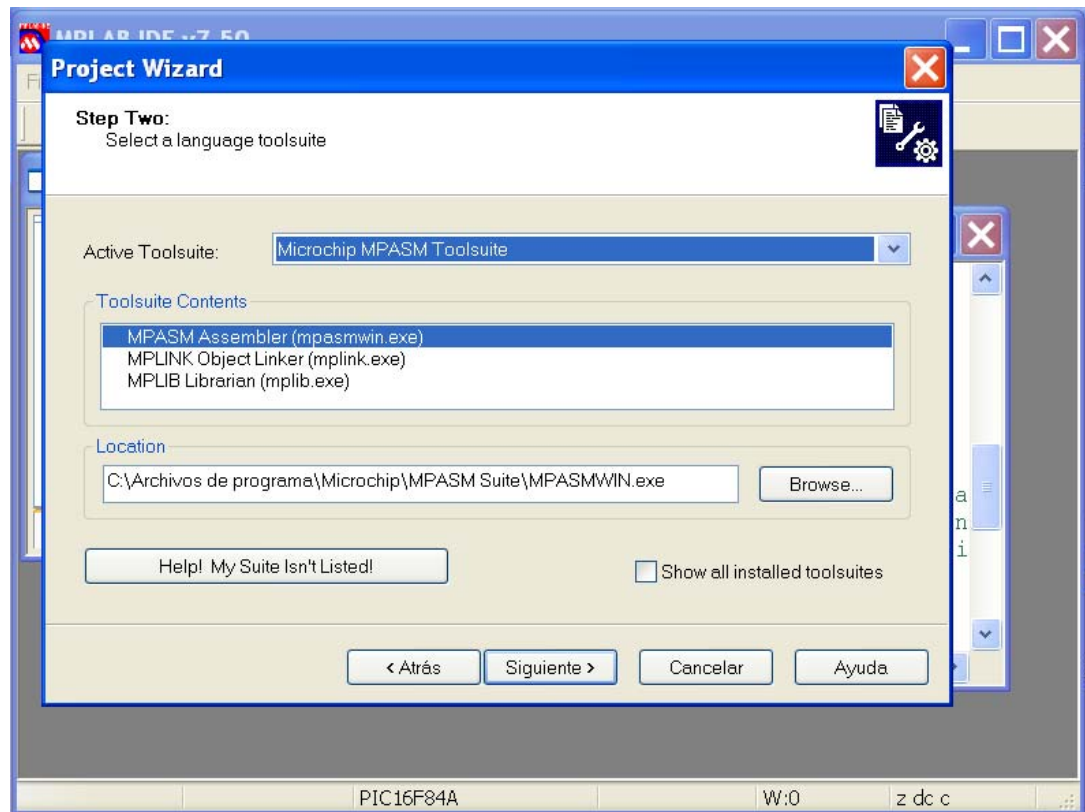
5) Seleccione **Project**→**Wizard**



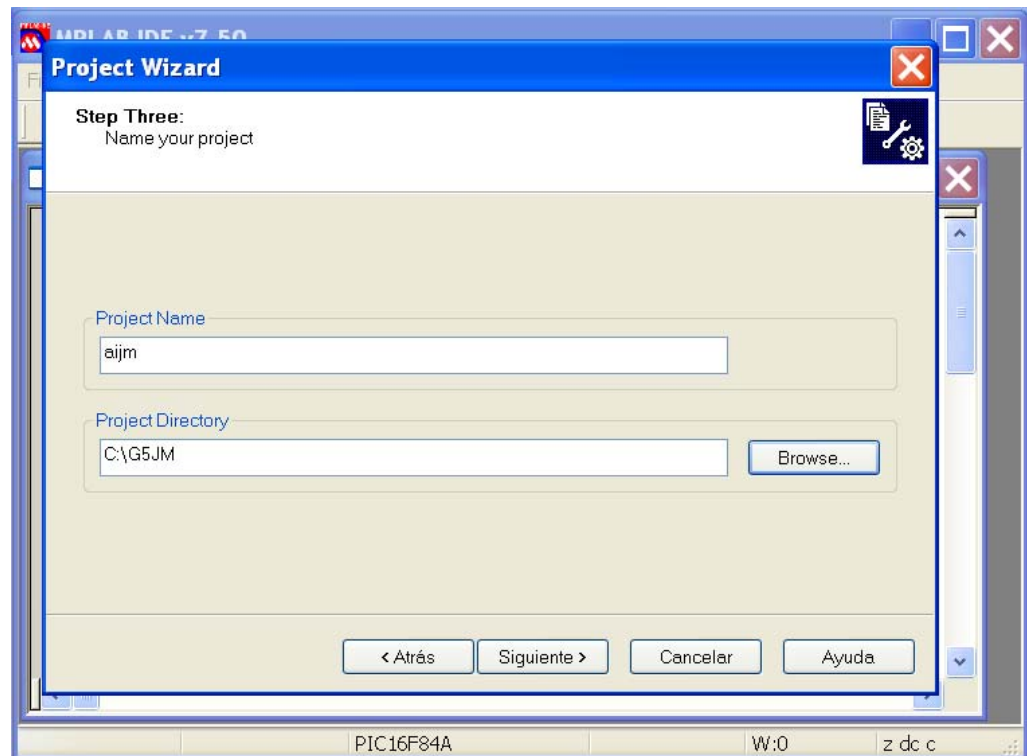
6) Presione Siguiete.



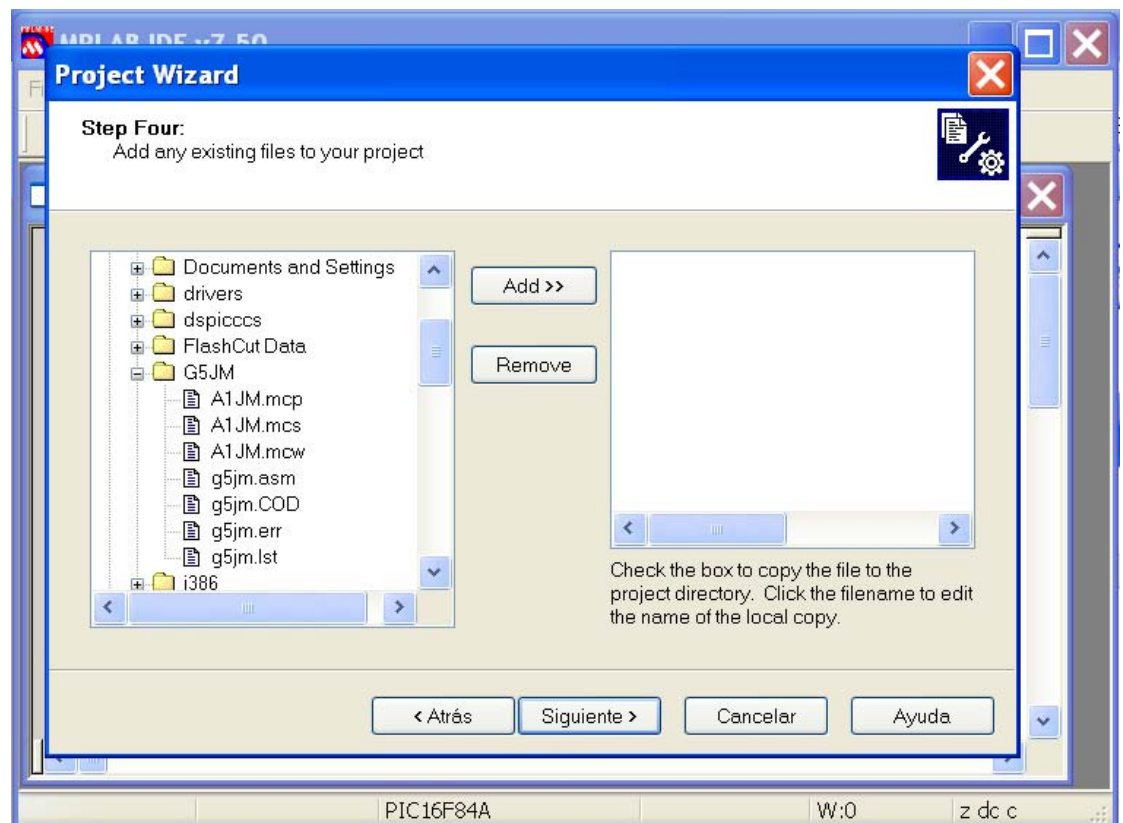
7) De la ventana plegable seleccione el microcontrolador 16F84A y presione Siguiete.



8) Si tiene seleccionado **Microchip MPASM Toolsuite** (En caso de no tenerlo seleccionado tendrá que hacerlo) presione siguiente. Ingrese el nombre del proyecto A1 adjuntando las iniciales de su grupo (Ejemplo **A1JM.asm** para el Grupo de Jose y Maritza). Seleccione con **Browse** su directorio de trabajo.



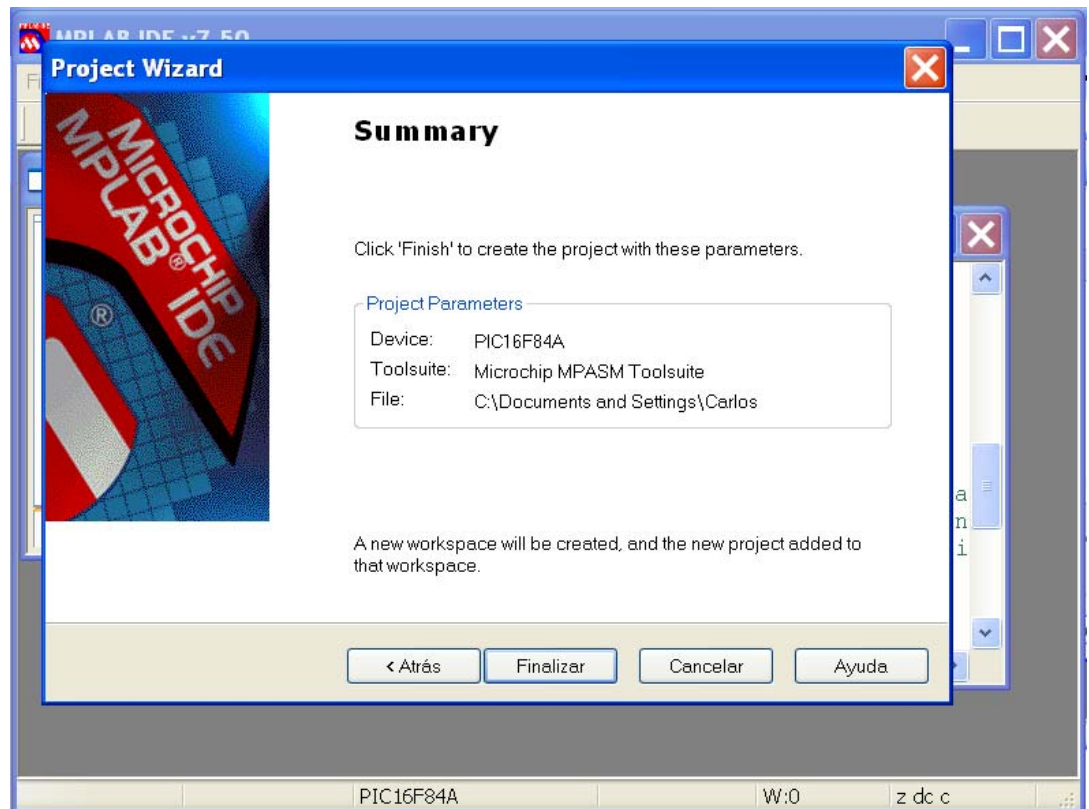
9) Presione siguiente.



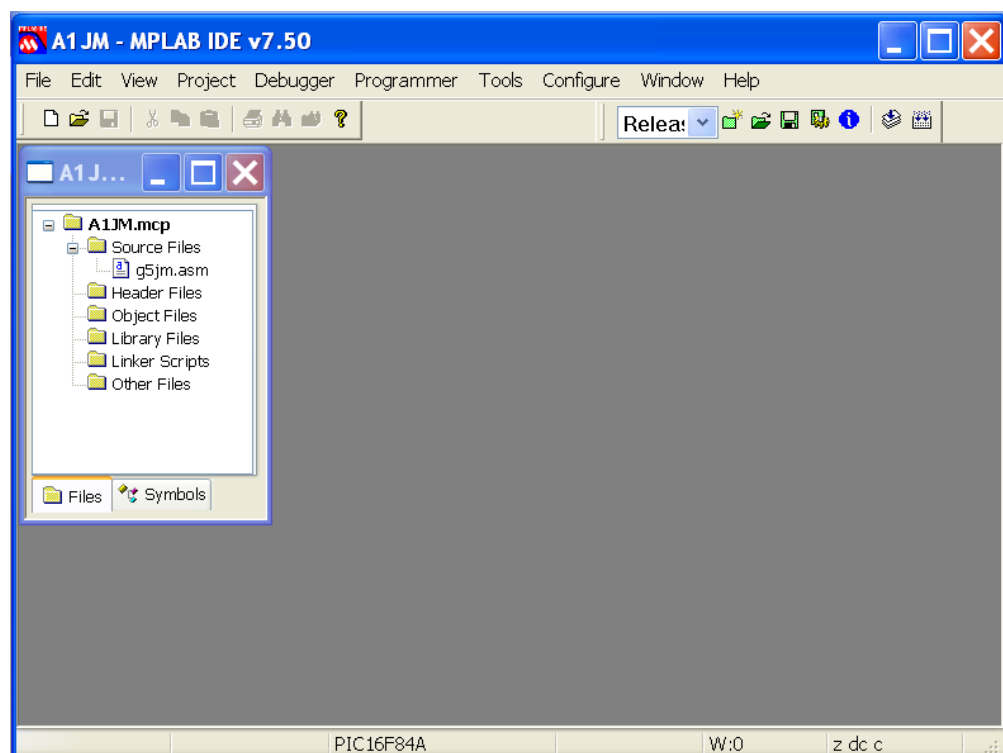
10) Seleccione su programa .asm (Por ejemplo: g5jm.asm) y pulse el botón **ADD**.



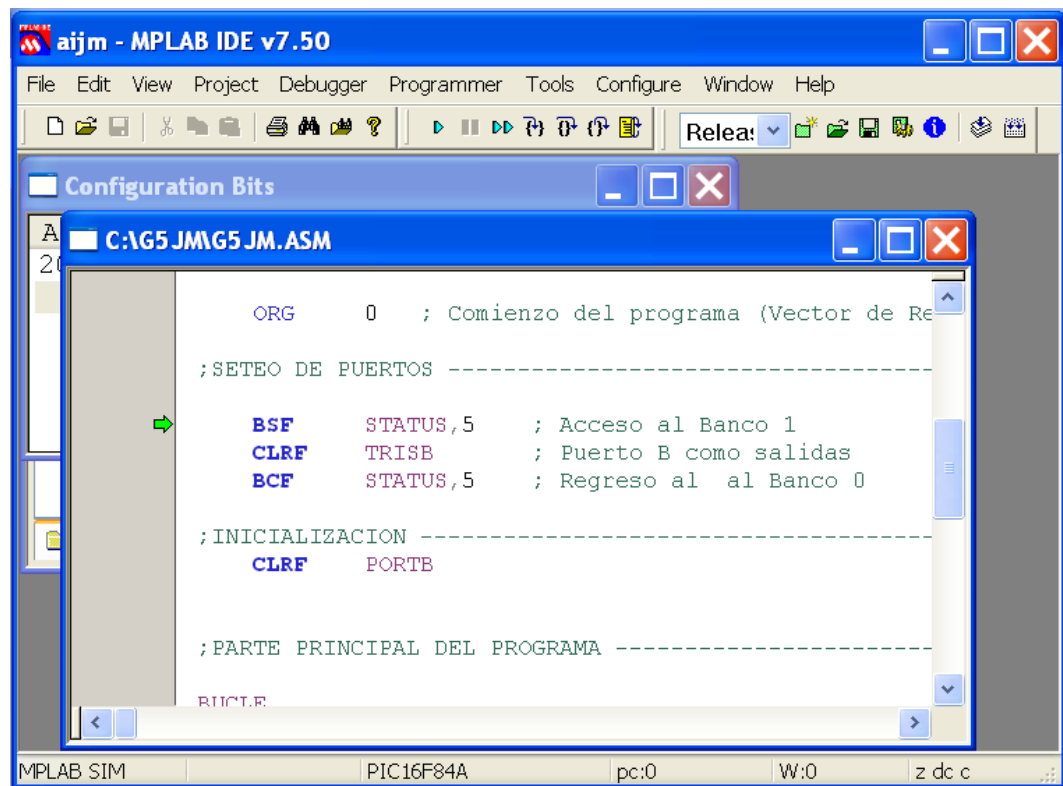
11) Luego presione siguiente.



11) Presione Finalizar.



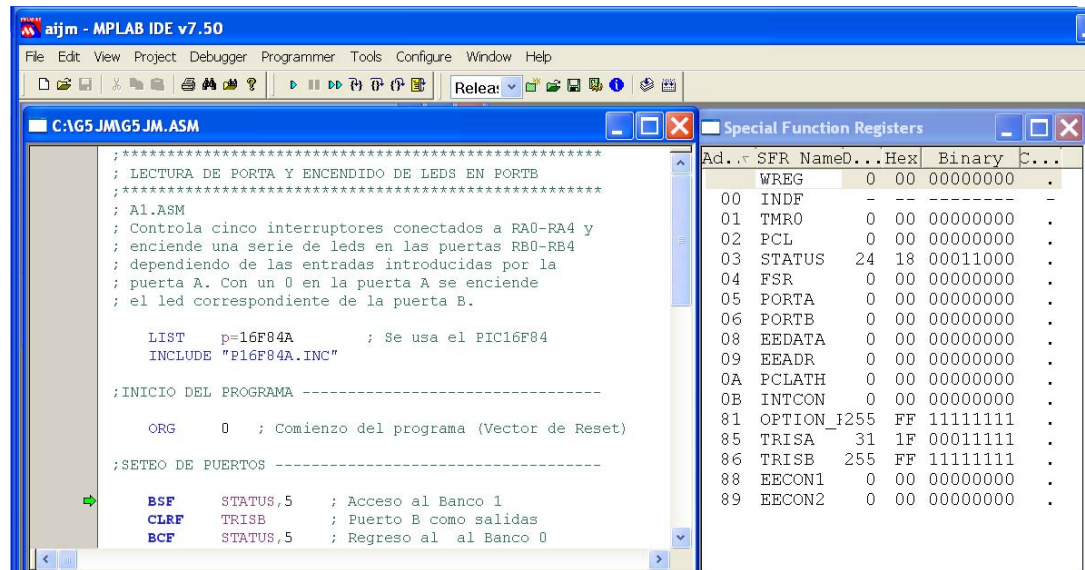
Haciendo doble clic en el error lo conduce al sitio del error.



EJECUCION PASO A PASO

Una vez compilado el programa se podrá ejecutar paso a paso. Para lo cual es recomendable ordenar el espacio de trabajo según las siguientes recomendaciones:

- 1) Con **View** → **Special Function Registers** abra los registros de funciones especiales SFR y colóquelos del lado derecho tal como se muestra en la gráfica siguiente:



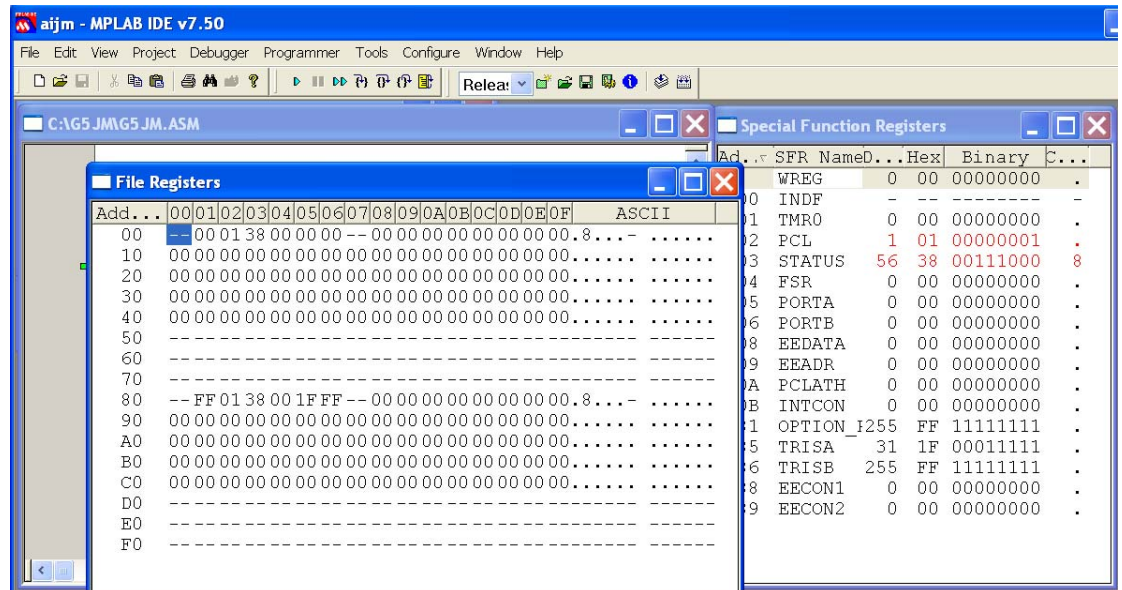
2) Ahora podrá ejecutar paso a paso el código con **Debugger** → **Step into**

O su ícono equivalente ó **F8**. En los SFR register se podrán observar los cambios que sufren los registros de acuerdo con la ejecución de cada instrucción.

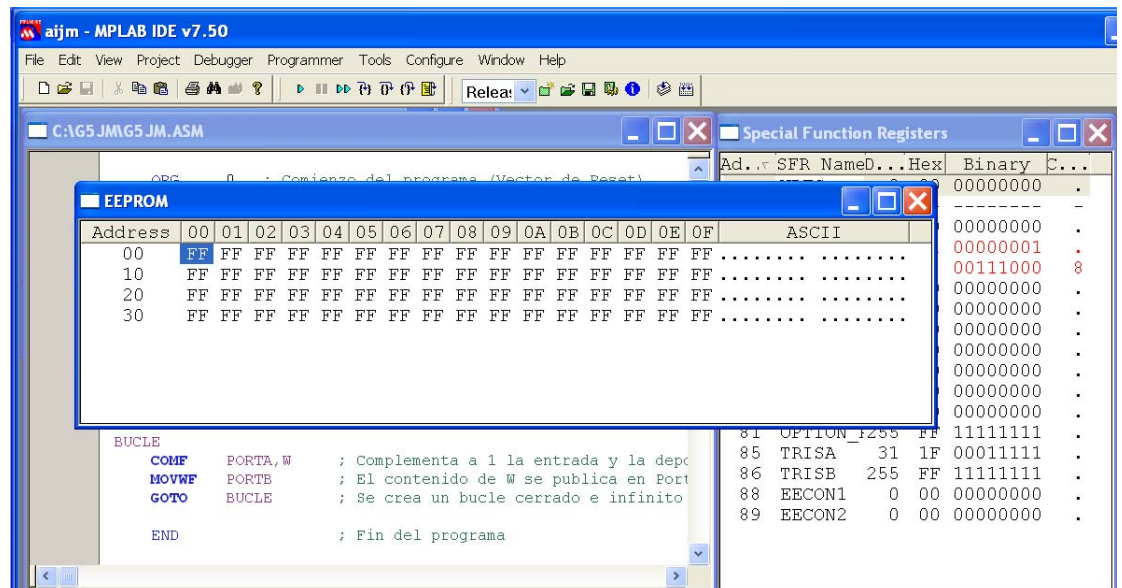
3) Analice el comportamiento de cada uno de los íconos del MPLAB SIM (Que son aquellos que aparecen cuando se selecciona **Debugger** → **Select Tool** → **MPLAB SIM** y en el ANEXO DE RESPUESTAS A PRACTICA 1, indique el funcionamiento de cada uno de ellos

OBSERVACION DE REGISTROS VARIOS

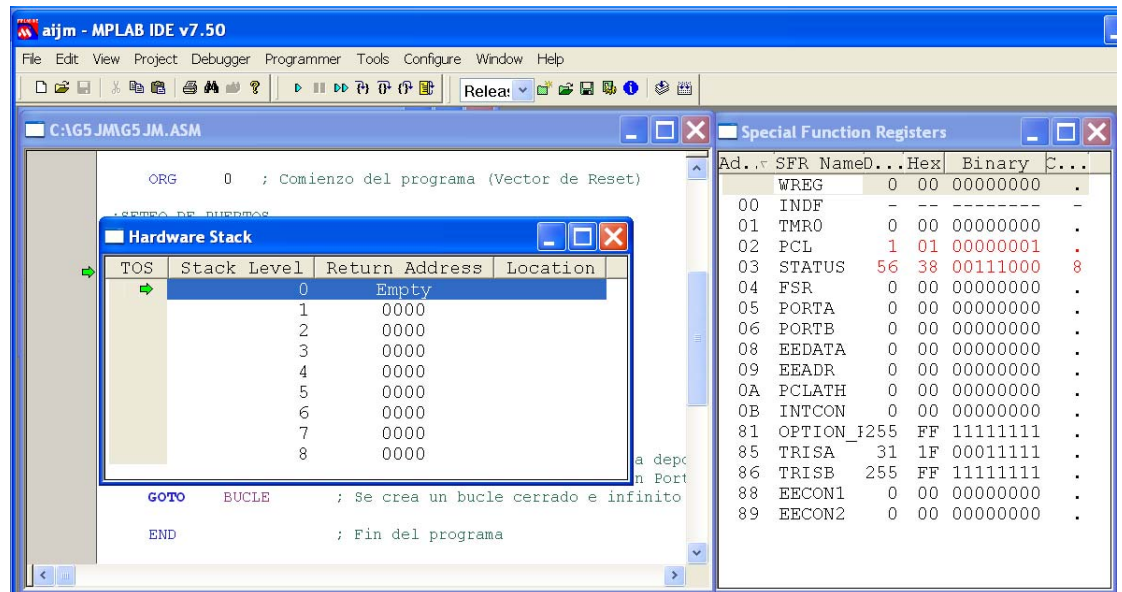
1) Con **View** → **File Registers** se pueden observar los contenidos de los dos bancos de datos del 16F84A



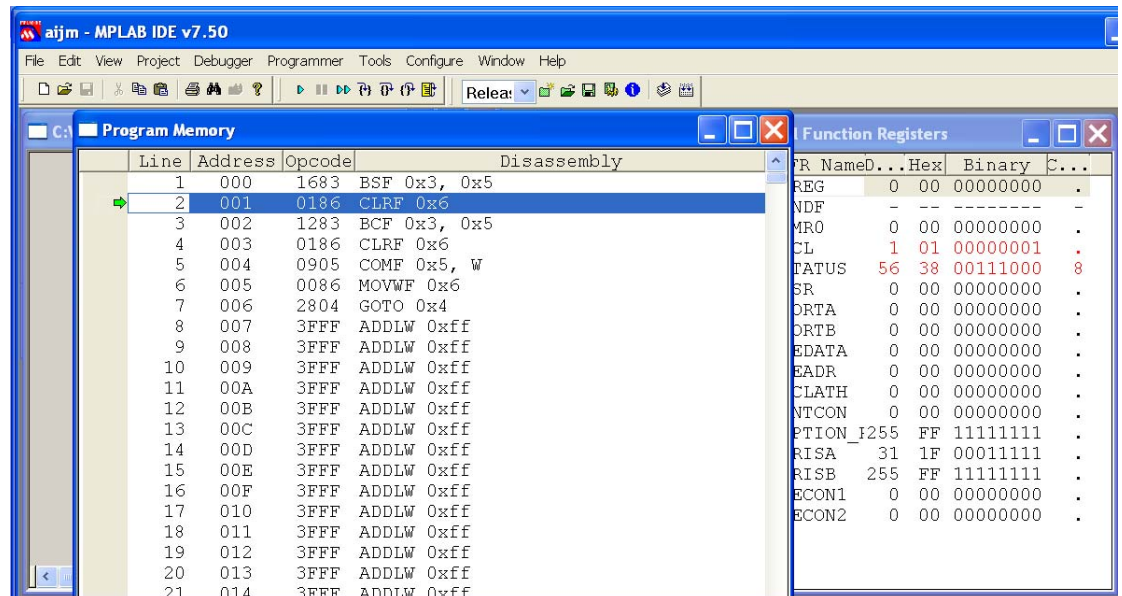
2) Con **View**→**EEPROM** observará el contenido de las 64 posiciones de memoria EEPROM del 16F84A.



3) Con **View**→**Hardware Stack** observará los 8 niveles de stack disponibles en los microcontroladores de la gama media.



4) Con **View** → **Program Memory** pueden mirarse detalles de cada una de las instrucciones en la memoria de programa.



c) PROGRAMACION

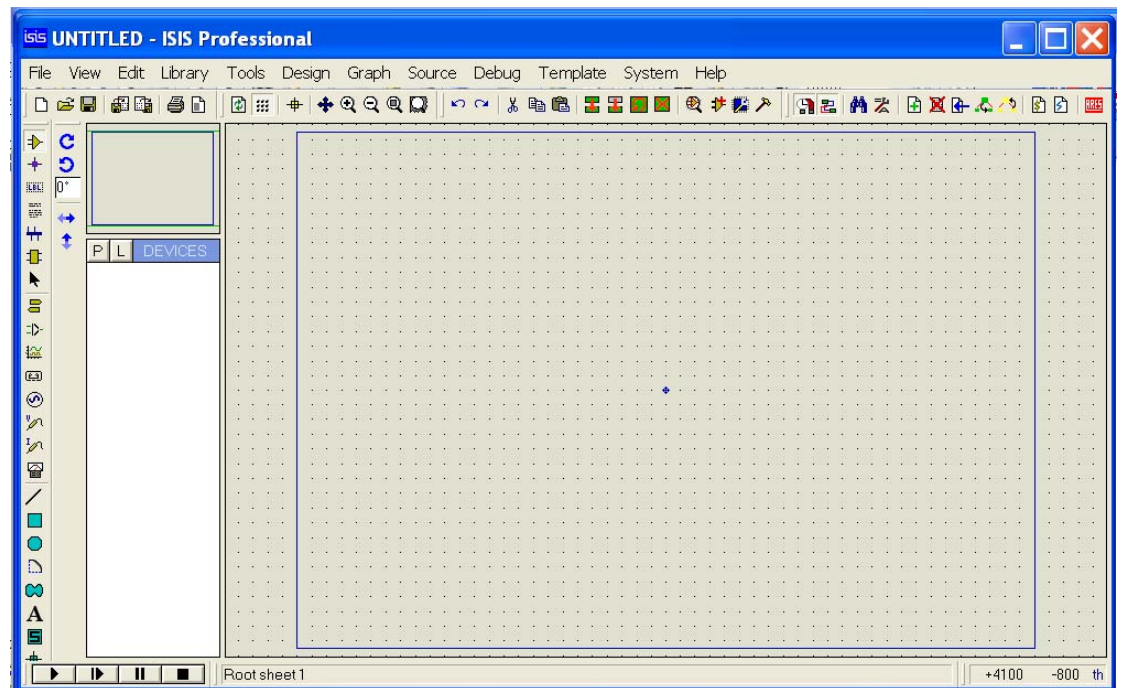
Como resultado de la compilación debió haberse generado un archivo .hex que es el ejecutable que se usa para programar el PIC.

- 1) Conecte el Programador **Pic Start Plus** en el puerto serial.
- 2) En **MPLAB** seleccione **PROGRAMMER→Select Programmer→PICSTART Plus**
- 3) Seleccione **Programmer→Enable Programmer** y se activará el programador.
- 4) Con **Programmer→Program** se programará el PIC (Se puede observar el titileo del LED del programador mientras se realiza la programación del PIC).

d) SIMULACION EN PROTEUS

La simulación en el Programa PROTEUS nos permite una excelente visualización del comportamiento del PIC programado dentro del circuito utilizado. Siga los siguientes pasos:

- 1) En el Programa PROTEUS abra ISIS:



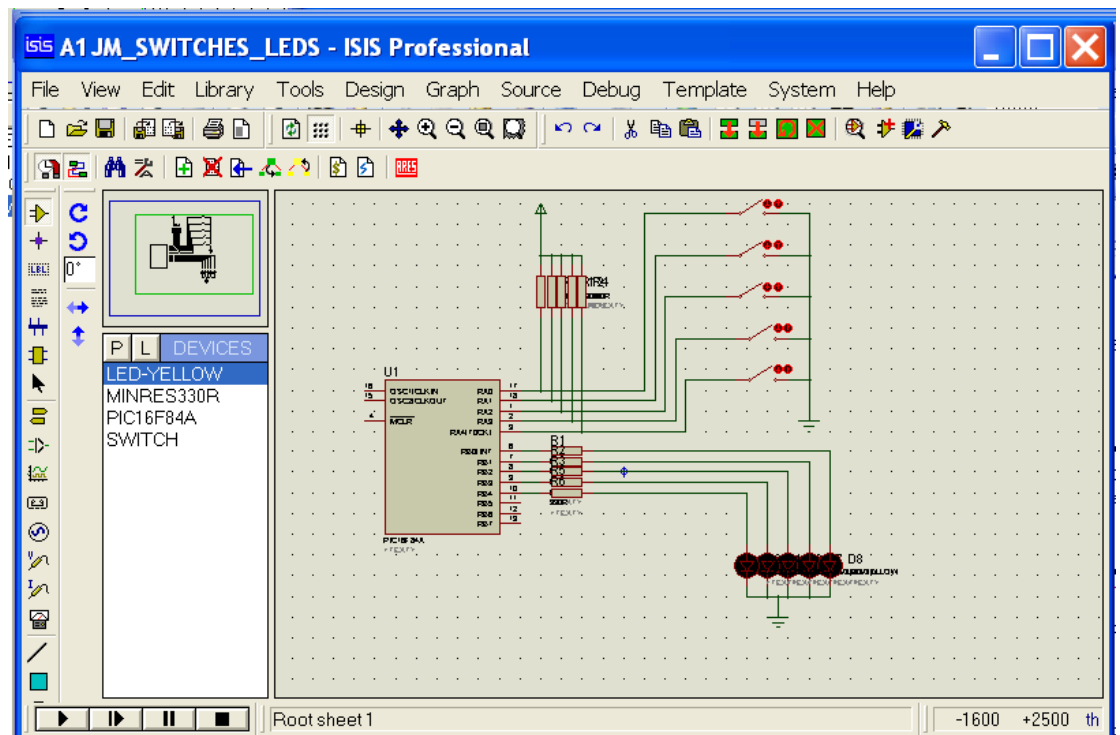
2) Seleccione **Component** y luego **P**. Esto abrirá una ventana de selección de componentes en donde escogeremos los que vamos a utilizar

3) En **Keywords** ingresamos **16F84A** y hacemos doble clic en el elemento respectivo de la derecha, esto transfiere el componente a nuestro ambiente de trabajo. Hacemos lo mismo con **switch**, **res 330R** y **led yellow**, para que también se transfieran.

4) Guarde la hoja de ISIS con el nombre **AlNN_SWITCHES_LEDS**

5) Con estos elementos se puede empezar a dibujar el circuito arrastrando y pegando elementos del espacio de trabajo al plano de dibujo.

6) Acercando el Mouse a los terminales de los componentes se los va uniendo hasta obtener la siguiente figura:



7) Haga un clic derecho en el centro del Microcontrolador y en la ventana que se abre seleccione en la línea de Program File el ícono de la carpeta y aparecerá un directorio de búsqueda que tenemos que hacerlo apuntar al archivo AlNN.hex que se formó como resultado de la compilación de Aljm.asm y que debe de estar en su carpeta de trabajo. Luego presione OK y está listo para la simulación

8) Presione PLAY en la parte inferior izquierda de la pantalla de ISIS y empezará la simulación. Observe que haciendo clic sobre los switches causará que estos se abran o cierren y afecten el valor de prendido o apagado de los LEDs.

ANEXO DE RESPUESTAS A PRACTICA 1

Run .- compila el programa -asm, y aparece una flecha verde en la primera fila del programa, para poder hacer una depuración paso a paso.

Halt.- sirve para hacer una pausa de la depuración.

Animate.- depura el programa de manera continua

Step Into.- al ejecutar el run, depura la línea donde está la flecha verde y entra a las subrutinas

Step Over.- lo mismo que el step into, pero este no entra a las subrutinas, las ejecuta sin entrar a ellas

Step Out.- cuando está ejecutando la subrutina, y ya no se quiere seguir ejecutando paso a paso, este lo ejecuta rápidamente, y sale de la subrutina

Reset.- vuelve a las condiciones iniciales del programa que se estaba ejecutando

OBJETIVO:

Aprender el manejo de DISPLAYS de LEDS binarios,

DISPLAYS de 7 segmentos

Combinar rutinas útiles de diversos programas

Entender la aplicación de las tablas conversión

CONTENIDO:

- a) PROYECTO A3_0
- b) PROYECTO A3
- c) PROYECTO A3_1 CON SUS ETIQUETAS Y SUBRUTINAS
- d) SUBRUTINA PARA RETARDO_200_MS
- e) GRABACION DE PIC
- f) PRUEBA DE DISEÑO EN HARDWARE

DESCRIPCION DEL CONTENIDO:

Esta práctica consta de tres partes. En la primera parte se trabaja con el archivo A3_0 y se observa una salida binaria en LEDS. En la segunda parte se utiliza el archivo A3 que nos permite visualizar la salida en un DISPLAY de 7 segmentos y la tercera parte en la que debemos realizar cambios que nos permitan incorporar un control MANUAL/AUTO para visualizar dos comportamientos diferentes en un DISPLAY de 7 segmentos.

DESARROLLO DE LA PRACTICA:**a) PROYECTO A3_0**

Con el archivo A3_0 cree un programa utilizando las técnicas aprendidas en la Práctica 1. Deberá compilar el programa con el MPLAB y armarlo en PROTEUS. A continuación se presenta el listado del programa y el gráfico de la simulación en PROTEUS:

```

;*****
;
;      USO DE UNA TABLA PARA EL
; ENCENDIDO DE UN DYPLAY DE LEDS EN BINARIO
;*****
; A3_0.ASM
; Programa que visualiza en binario un número del
; 1 al 7 en ocho LEDS colocados en el puerto B
; dependiendo del valor binario de tres
; interruptores situados en la puerta A

; Se usa el mismo programa A3 pero sin el llamado
; a la tabla de conversión a código de 7 segmentos

      LIST      P=16F84A
      INCLUDE  "P16F84A.INC"

;ASIGNACION DE VARIABLES-----

VALOR EQU    0X0C          ; Registro auxiliar

;INICIO DEL PROGRAMA -----

```

```

ORG    0

GOTO  INICIO

;SETEO DE PUERTOS -----

INICIO

    BSF    STATUS,5    ; Selección del banco 1

    CLRF  PORTB        ; Puerta B como salida

    BCF    STATUS,5    ; Regreso al banco 0

;PARTE PRINCIPAL DEL PROGRAMA -----

EXPLORA

    MOVF  PORTA,W      ; Se cargan los switches
en W

    MOVWF VALOR        ; Se usa un registro auxiliar

    CLRW

;SE COMPRUEBA EL VALOR INGRESADO POR LOS SWITCHES PARA
;ASIGNAR EL PESO CORRESPONDIENTE

bit_0

    BTFSS VALOR,0      ; Si RA0 = 1, brinca
GOTO  bit_1            ; Salta a explorar el bit_1
    ADDLW 0x01         ; Si RA0 =1, se suma 1

bit_1

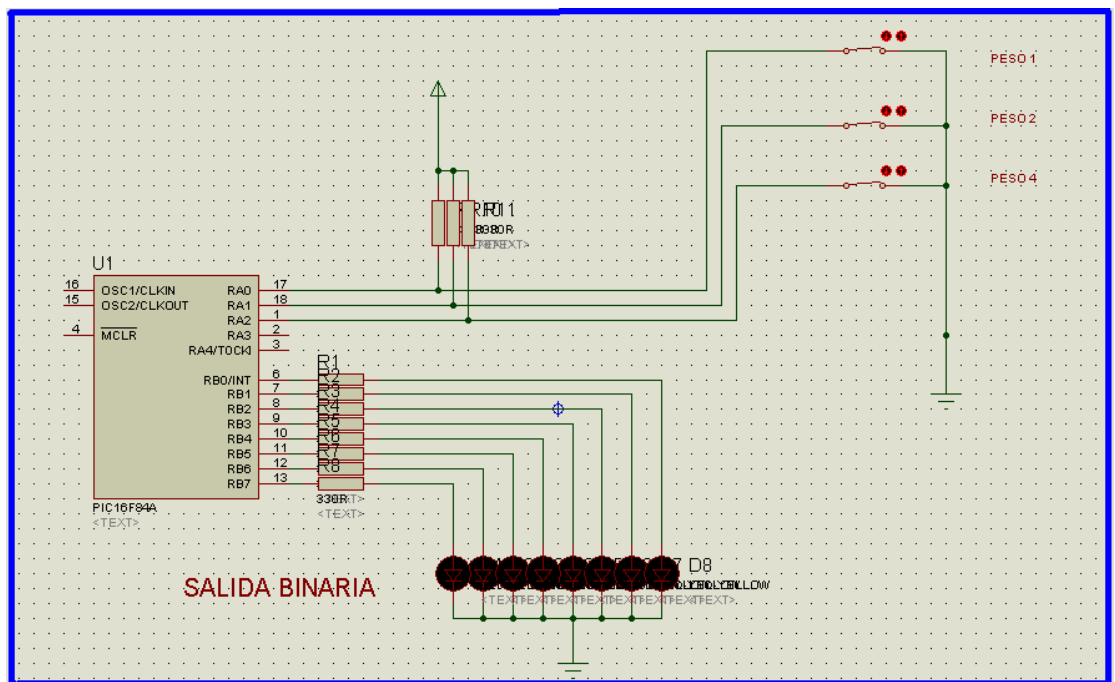
    BTFSS VALOR,1      ; Si RA1 = 1, brinca
GOTO  bit_2            ; Salta a explorar el bit_2
    ADDLW 0x02         ; Si RA1 = 1, se suman 2

bit_2

```

```

BTFSS VALOR,2           ; Si RA2 = 1, brinca
GOTO VISUAL             ; Salta al módulo del
display
ADDLW 0x04              ; Si RA2 0 1, se suman 4
VISUAL
MOVWF PORTB             ; Se publica en PORTB el valor
de W
GOTO EXPLORA           ; bucle infinito
FIN
END
    
```



b) PROYECTO A3

Con el archivo A3_0 cree un programa utilizando las técnicas aprendidas en la Práctica 1. Deberá compilar el programa con el MPLAB y armarlo en PROTEUS. A continuación se presenta el listado del programa y el gráfico de la simulación en PROTEUS:

```

;*****
;
;          USO DE UNA TABLA PARA EL
;
;          ENCENDIDO DEL DIPLAY DE 7 SEGMENTOS
;*****
; A3.ASM
; Programa que visualiza un número del
; 1 al 7 en un display de siete segmentos en la
; puerta B dependiendo del valor binario de tres
; interruptores situados en la puerta A

          LIST      P=16F84A
          INCLUDE  "P16F84A.INC"

;ASIGNACION DE VARIABLES-----

VALOR EQU   0X0C           ; Registro auxiliar

;INICIO DEL PROGRAMA -----

          ORG      0
          GOTO    INICIO

;SETEO DE PUERTOS -----

INICIO

```

```

BSF    STATUS,5    ; Selección del banco 1
CLRF   PORTB       ; Puerta B como salida
BCF    STATUS,5    ; Regreso al banco 0

```

```

;PARTE PRINCIPAL DEL PROGRAMA -----

```

```

EXPLORA

```

```

MOVWF  PORTA,W     ; Se cargan los switches en W
MOVWF  VALOR       ; Se usa un registro auxiliar
COMF   VALOR,F     ; Invertir niveles de switches
CLRWF  VALOR       ;

```

```

;SE COMPRUEBA EL VALOR INGRESADOS POR LOS SWITCHES PARA
;ASIGNAR EL PESO CORRESPONDIENTE

```

```

bit_0

```

```

BTFSS  VALOR,0     ; Si RA0 = 1, brinca
GOTO   bit_1       ; Salta a explorar el bit_1
ADDLW  0x01        ; Si RA0 =1, se suma 1

```

```

bit_1

```

```

BTFSS  VALOR,1     ; Si RA1 = 1, brinca
GOTO   bit_2       ; Salta a explorar el bit_2
ADDLW  0x02        ; Si RA1 = 1, se suman 2

```

```

bit_2

```

```

BTFSS  VALOR,2     ; Si RA2 = 1, brinca
GOTO   VISUAL      ; Salta al módulo del display

```

```

ADDLW  0x04        ; Si RA2 0 1, se suman 4

```

```

VISUAL

```

```

CALL   DISPLAY     ; Salta al módulo del
                    ; display

```

;DISPLAY: Rutina de visualización del acumulador en el display de 7 segmentos

MOVWF PORTB ; Se publica en PORTB el valor de W

GOTO EXPLORA ; bucle infinito

; TABLA DE CONVERSION-----

DISPLAY

ADDWF PCL,F ; PCL + W -> W
; El PCL se incrementa con el
; valor de W proporcionando un
; salto

RETLW 0x3F ; Retorna con el código del 0

RETLW 0x06 ; Retorna con el código del 1

RETLW 0x5B ; Retorna con el código del 2

RETLW 0x4F ; Retorna con el código del 3

RETLW 0x66 ; Retorna con el código del 4

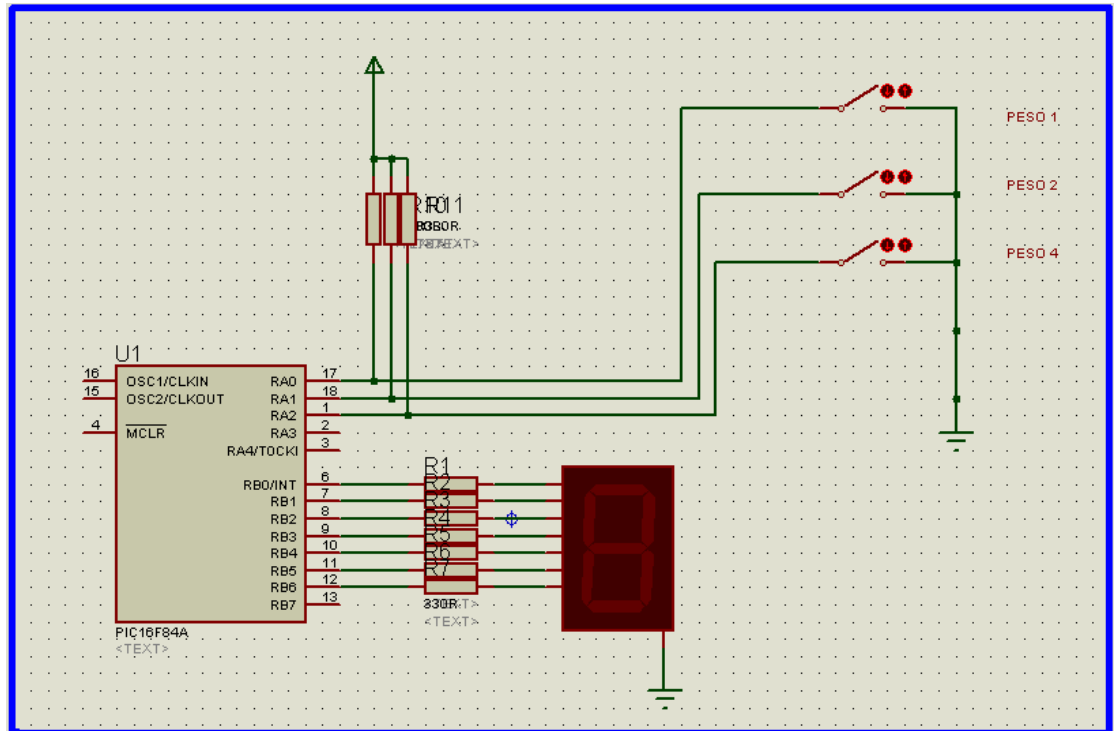
RETLW 0x6D ; Retorna con el código del 5

RETLW 0x7D ; Retorna con el código del 6

RETLW 0x07 ; Retorna con el código del 7

FIN

END



c) PROYECTO A3_1

Deberá crear un nuevo proyecto que cumpla con las especificaciones estipuladas en la siguiente cabecera de programa y gráfico en PROTEUS. Se proporcionan todas las Rutinas y Subrutinas, Vaya combinándolas en un solo archivo y probando por partes hasta que el todo funcione.

```

;*****
;
;          USO DE UNA TABLA PARA EL
;
;          ENCENDIDO DEL DIPLAY DE 7 SEGMENTOS
;*****
; A3_1.ASM
; Programa que en MODO MANUAL visualiza un número del
; 1 al 7 en un display de siete segmentos en la
; puerta B dependiendo del valor binario de tres

```



```

; interruptores situados en el puerto A
; Además permite mediante un SWITCH con las
; funciones MANUAL/AUTOMATICO colocado en RA3
; encender un LED en RB4 que indica modo
; AUTO cuando está encendido un LED en RA4 y
; modo MANUAL cuando está apagado.
; En modo AUTO el DISPLAY de 7 segmentos deberá
; mostrar la salida de un contador que va desde 0 a 9
; regresa a 0 y continúa contando hasta que se regrese
; el SWITCH a modo manual. El contador deberá contar
; cada 200 milisegundos para lo cual es necesario
; incorporar un retardo. (Adjunto encontrará el código
; para el retardo de 200 milisegundos).
;*****
***

```

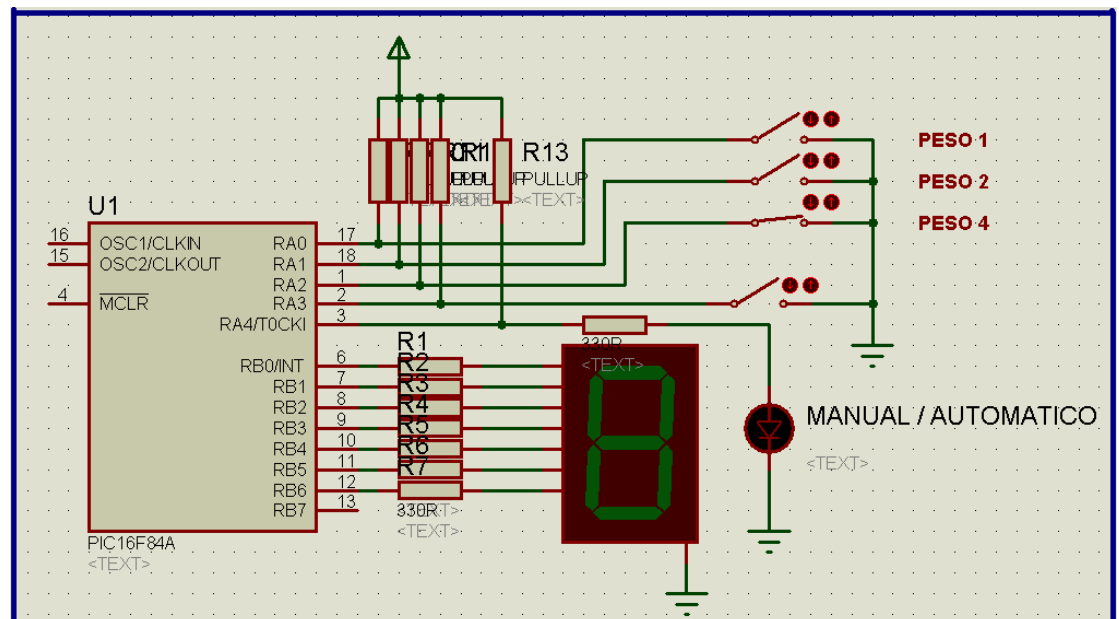
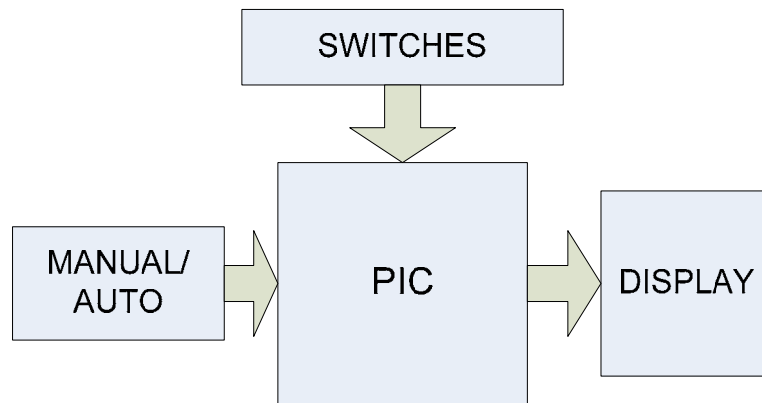
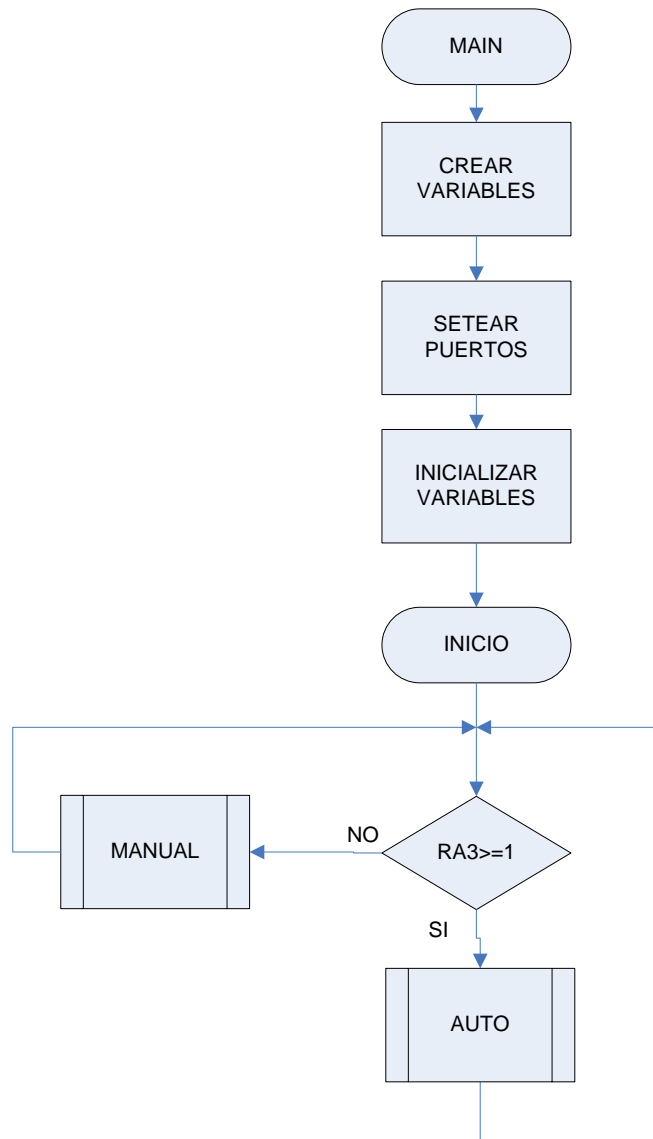


DIAGRAMA DE BLOQUES

RUTINA PRINCIPALRUTINA MAIN

```

LIST    P=16F84A

INCLUDE "P16F84A.INC"

```

```

;ASIGNACION DE VARIABLES-----

```

```

VALOR      EQU    0X0C  ; Registro auxiliar
D0         EQU    0X0D
CONTADOR1  EQU    0X0E
CONTADOR2  EQU    0X0F

```

```
CONTADOR3 EQU 0X10

;INICIO DEL PROGRAMA -----

    ORG 0

    GOTO INICIO

;SETEO DE PUERTOS -----

INICIO

    BSF STATUS,5 ; Selección del banco 1
    CLRF PORTB ; Puerta B como salida
    BCF TRISA,4
    BCF STATUS,5 ; Regreso al banco 0

;PRUEBA MANUAL/AUTO

;(RA3=1 --> AUTO)

;(RA3=0 --> MANUAL)

PRUEBA

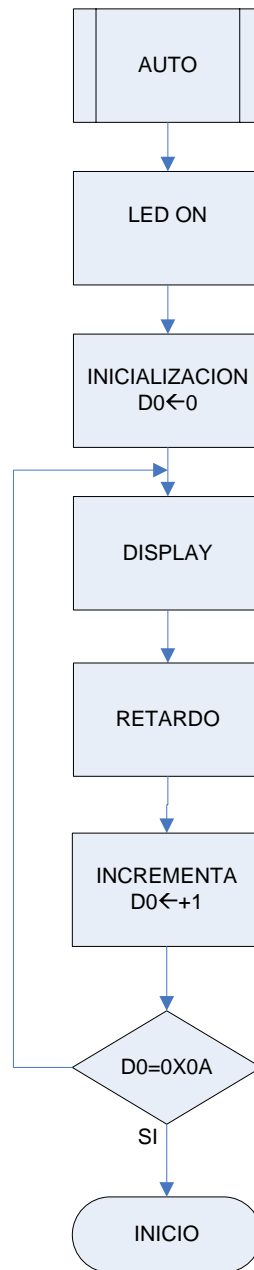
    BTFSS PORTA,3

    GOTO MANUAL

    GOTO AUTO

; (Colocar aquí Rutina MANUAL)
; (Colocar aquí Subrutina DISPLAY)
; (Colocar aquí Rutina AUTO)
; (Colocar aquí Subrutina RETARDO_200_MS)

    END
```

RUTINA AUTO

AUTO

BSF PORTA, 4

CLRF D0

INCREMENTA

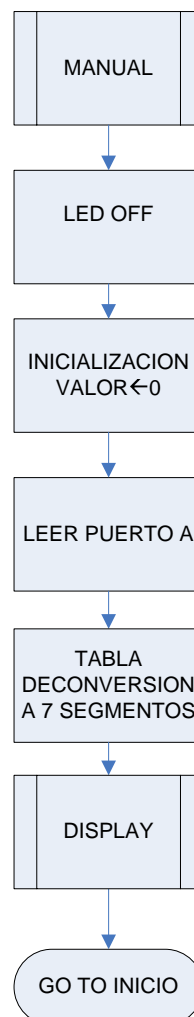
MOVF D0, W

CALL DISPLAY

NOP

```
CALL RETARDO_200MS  
NOP  
INCF D0 ,F  
MOVLW 0X0A  
SUBWF D0 ,W  
BTFSS STATUS,Z ;STATUS,2 (Es lo mismo) Flag de  
cero  
GOTO INCREMENTA  
GOTO PRUEBA
```

RUTINA MANUAL



MANUAL

```

BCF    PORTA,4
MOVWF  PORTA,W    ; Se cargan los switches en W
MOVWF  VALOR      ; Se usa un registro auxiliar
CLRWF  VALOR

```

```

;SE COMPRUEBA EL VALOR INGRESADOS POR LOS SWITCHES PARA
;ASIGNAR EL PESO CORRESPONDIENTE

```

bit_0

```

BTFSS  VALOR,0    ; Si RA0 = 1, brinca
GOTO   bit_1      ; Salta a explorar el bit_1
ADDLW  0x01       ; Si RA0 =1, se suma 1

```

bit_1

```

BTFSS  VALOR,1    ; Si RA1 = 1, brinca
GOTO   bit_2      ; Salta a explorar el bit_2
ADDLW  0x02       ; Si RA1 = 1, se suman 2

```

bit_2

```

BTFSS  VALOR,2    ; Si RA2 = 1, brinca
GOTO   VISUAL     ; Salta al módulo del display

ADDLW  0x04       ; Si RA2 0 1, se suman 4

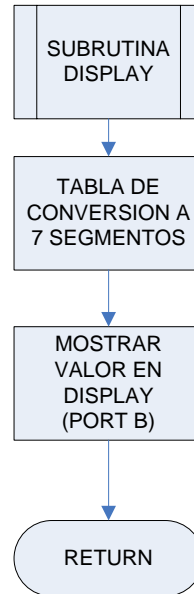
```

VISUAL

```

CALL  DISPLAY
CALL  RETARDO_200MS
GOTO  PRUEBA

```

SUBROUTINA DISPLAY

DISPLAY

```

; Rutina de visualización del valor equivalente
; en código de 7 segmentos del registro W
; Se accede a esta subrutina con la instrucción:
; CALL DISPLAY

        CALL  CONVERSION_7SEG    ; Va a la Tabla de
                                   ; Conversión a 7 segmentos

        MOVWF PORTB              ; Se publica código en PORTB

        NOP

        RETURN

; TABLA DE CONVERSION-----
CONVERSION_ 7SEG

        ADDWF PCL,F              ; PCL + W -> W
                                   ; El PCL se incrementa con el
                                   ; valor de W proporcionando un
                                   ; salto
  
```



```

RETLW 0x3F      ; Retorna con el código del 0
RETLW 0x06      ; Retorna con el código del 1
RETLW 0x5B      ; Retorna con el código del 2
RETLW 0x4F      ; Retorna con el código del 3
RETLW 0x66      ; Retorna con el código del 4
RETLW 0x6D      ; Retorna con el código del 5
RETLW 0x7D      ; Retorna con el código del 6
RETLW 0x07      ; Retorna con el código del 7

RETLW _____ ; Completar con el código del 8

RETLW _____ ; Completar con el código del 9

```

d) SUBROUTINA PARA RETARDO 200 MS

```

;*****
*****

;                               RETARDO DE 200 MILISEGUNDOS
;*****
*****

; ACCEDER A RETARDO CON CALL RETARDO_200MS
; COLOCAR EN ZONA DE ETIQUETAS:
; CONTADOR1 EQU 0X0E ; Use localidades de memoria que no
; CONTADOR2 EQU 0X0F ; se repitan con las definidas para
; CONTADOR3 EQU 0X10 ; otras variables a partir de 0x0C

RETARDO

    MOVLW    0X0
    MOVWF   CONTADOR1
    MOVLW    0X0

```

```
MOVWF    CONTADOR2
MOVLW    0X01
MOVWF    CONTADOR3
```

RETARDO1

```
DECFSZ   CONTADOR1 , F
GOTO     RETARDO1
DECFSZ   CONTADOR2 , F
GOTO     RETARDO1
DECFSZ   CONTADOR3 , F
GOTO     RETARDO1
RETURN
END
```

e) GRABACION DE PIC

Con el programa A3_1 deberá programar el PIC siguiendo las técnicas de la Práctica 1.

f) PRUEBA DE DISEÑO EN HARDWARE

El chip programado será probado en hardware mediante un PROTOBOARD armado por esta Práctica.

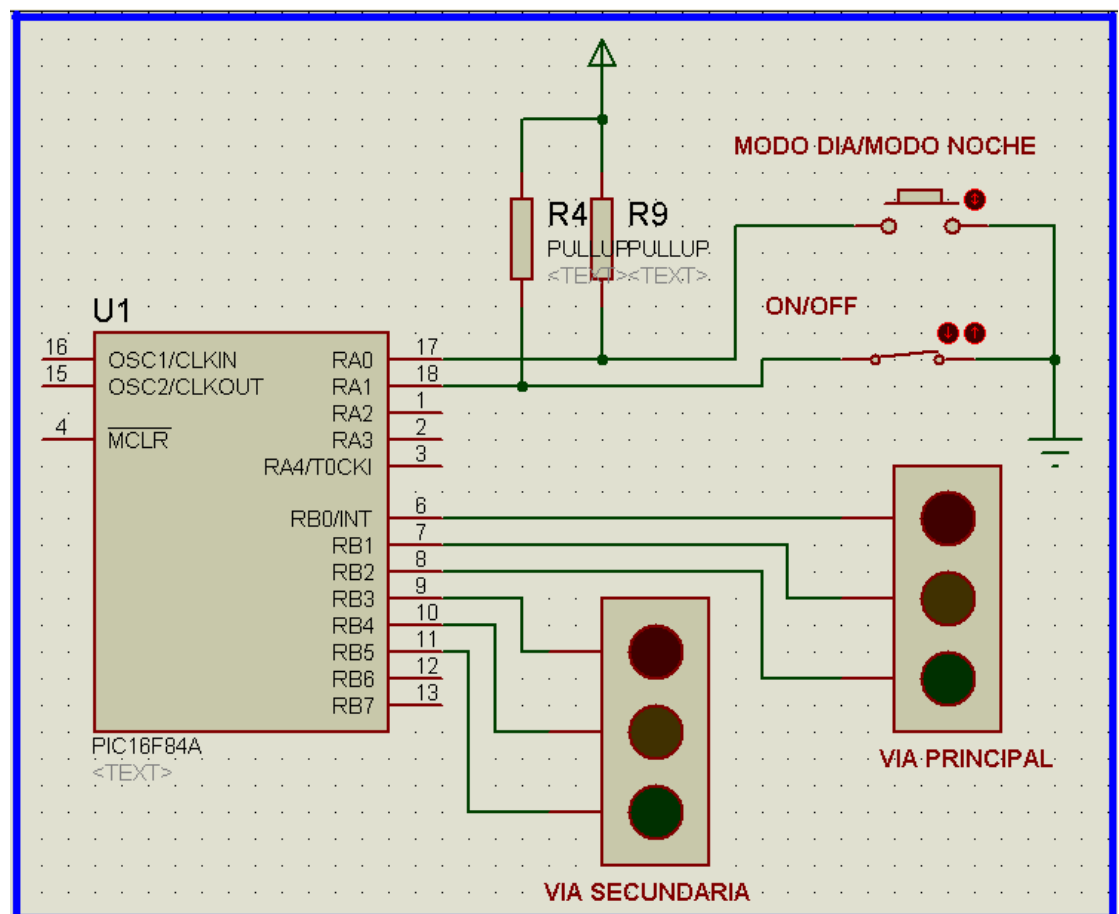
Una vez completado su trabajo revíselo y asegúrese de entender todas las instrucciones y diagramas de flujo.

Fin de práctica

4.2.3 SEMAFORO PARA EL CONTROL DE INTERSECCION DE DOS VIAS UNIDIRECCIONALES

PRACTICA 3

SEMAFORO PARA EL CONTROL DE INTERSECCION DE DOS VIAS UNIDIRECCIONALES



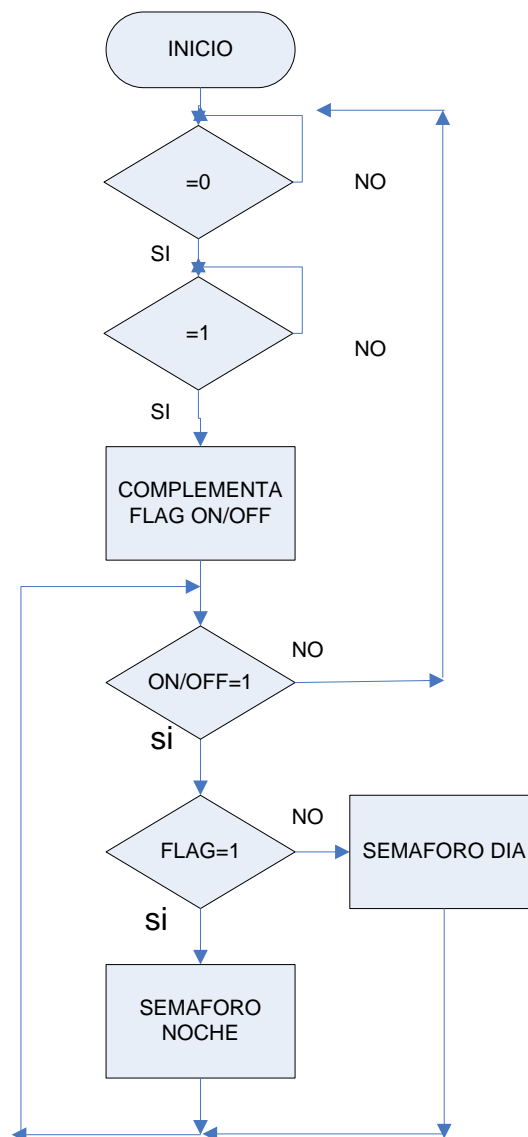
OBJETIVO:

Desarrollar destrezas en la creación de programas siguiendo un diagrama de flujo y ciertas especificaciones

CONTENIDO:

- a) DIAGRAMA DE FLUJO A NIVEL ALTO
- b) ESPECIFICACIONES
- c) SET DE INSTRUCCIONES DEL PIC 16F84A
- d) GRABACION DE PIC
- e) PRUEBA DE DISEÑO EN HARDWARE

a) DIAGRAMA DE FLUJO



b) Especificaciones

Se trata del diseño de un semáforo basado en el microcontrolador PIC 16F84A de acuerdo al diagrama de flujo presentado y de acuerdo con las siguientes especificaciones:

El semáforo controla la intersección de dos vías unidireccionales.

El sistema tendrá un SWITCH de ON/OFF que será necesario cerrar para que se accione. Luego de lo cual se puede presionar el botón MODO DIA/MODO NOCHE el mismo que se iniciará en el MODO DIA.

Para cambiar de modo será necesario desactivar el sistema colocando el SWITCH ON/OFF en OFF y presionar el botón MODO DIA/MODO NOCHE (Con esto se regresa al inicio que es un nivel superior)

El MODO DIA encenderá luces cada segundo en la siguiente secuencia R-R/A-V-A en la calle principal y V-A-R-R/A

R=rojo

R/A=rojo y amarillo

V=verde

A=amarillo

R/A=rojo amarillo

En el modo noche la principal prenderá intermitentemente las luces rojas y la secundaria las luces amarillas en intervalos de 400 milisegundos.

El semáforo se lo obtiene en PROTEUS con el nombre de TRAFFIC LIGHTS.

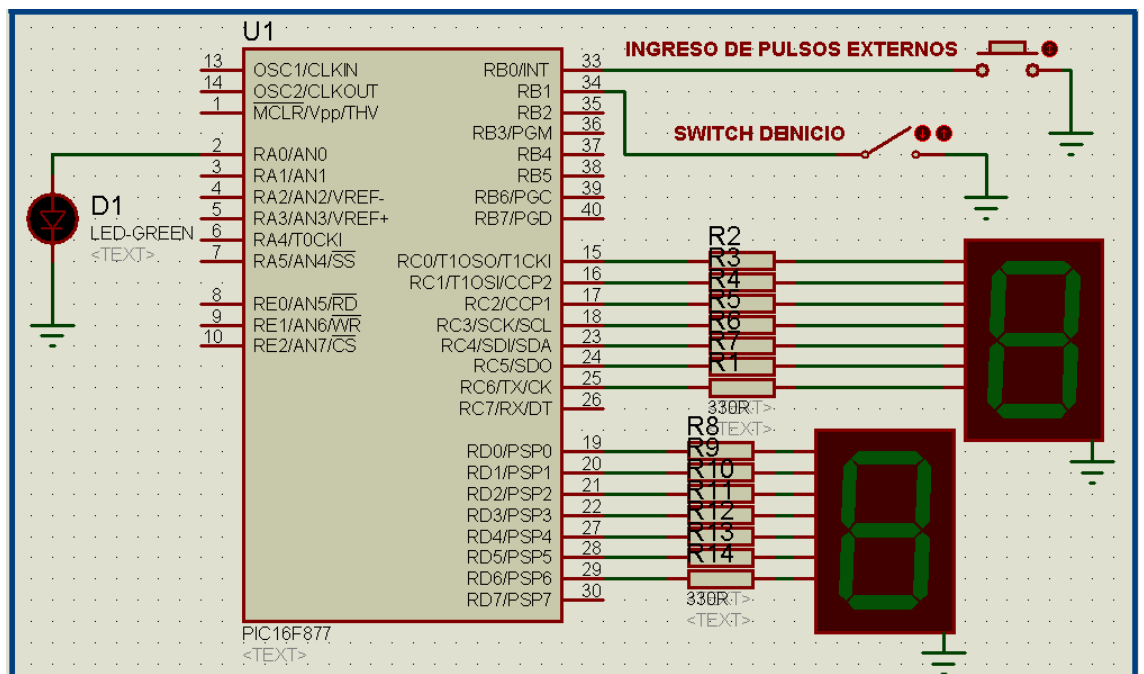
C) SET DE INSTRUCCIONES DEL PIC 16F84A

Se adjunta PDF “ Instrucciones assembler para 16F84”

4.2.4 CONTADOR DE 10 CAJAS CON 8 ARTÍCULOS EN UNA LINEA DE PRODUCCION MEDIANTE EL USO DE LA INTERRUPCION EXTERNA

PRACTICA 4

CONTADOR DE 10 CAJAS CON 8 ARTÍCULOS EN UNA LINEA DE PRODUCCION MEDIANTE EL USO DE LA INTERRUPCION EXTERNA



OBJETIVO:

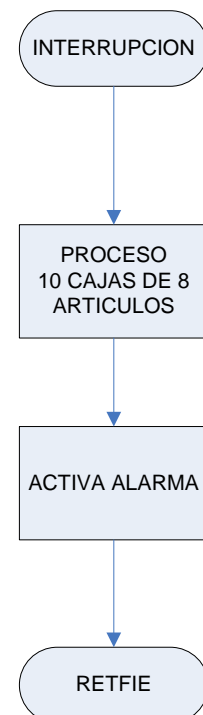
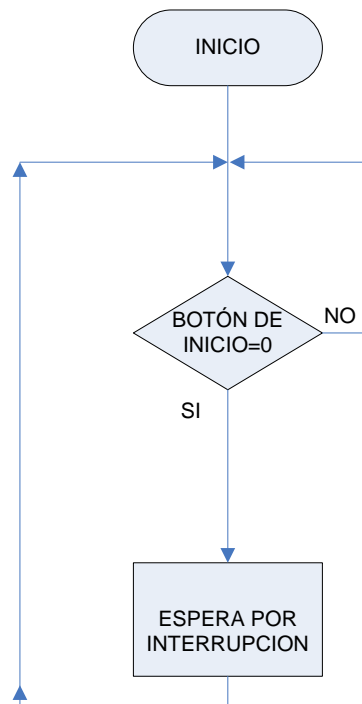
Familiarizarse con el manejo de las interrupciones

Hacer uso de la interrupción externa

Utilizar el PIC 16F877A

CONTENIDO:

- a) DIAGRAMA DE FLUJO A NIVEL ALTO
- b) ESPECIFICACIONES
- c) GRABACION DE PIC
- d) PRUEBA DE DISEÑO EN HARDWARE

a) DIAGRAMA DE FLUJO

a) ESPECIFICACIONES

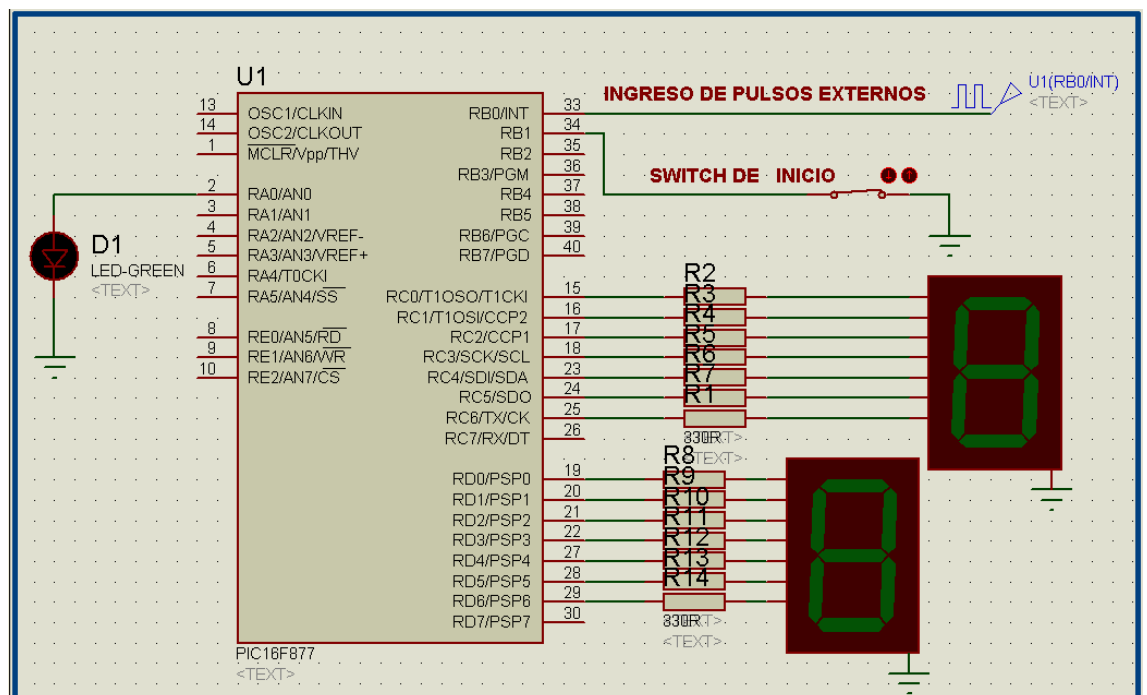
Se requiere contar el número de artículos producidos en una línea de producción así como también el número de cajas. En cada caja entran ocho artículos. Para realizar este trabajo deberá utilizar un PIC 16F877A que tiene dos DISPLAYS de 7 segmentos. Uno en el puerto C y otro en el D.

El funcionamiento es el siguiente:

El operador cierra el switch de inicio y comienza el proceso de conteo. El número de artículos por caja se va acumulando en un DISPLAY de 7 segmentos al llegar a ocho cuenta una caja y regresa a cero, Cuenta nuevamente hasta ocho e incrementa el número de cajas a dos y así sucesivamente hasta completar diez cajas. Terminadas las diez cajas se emite una alarma que consiste en prender un LED.

Los artículos van pasando uno a uno por una cinta transportadora. Un sensor óptico determina su presencia y envía un pulso por artículo a la entrada de interrupción externa del PIC. Es Decir que al detectarse un artículo se produce una interrupción externa y el programa irá a atender esta subrutina, terminada la cual retorna al programa principal mediante la instrucción RETFIE. El programa principal contiene un lazo infinito en espera de la interrupción.

Una vez que su programe funcione, en Proteus seleccione el ícono
 Generador→DPULSE y colóquelo en lugar del pulsador en RB0 y
 prográmelo para que funcione a 10 hertz y podrá observar que su diseño
 funciona en forma continua.



NOTAS:

1) Para que el PIC 16F877A compile correctamente deberá hacer al registro
 ADCON1=7 en el BANCO 1. Esto debido a que por default el PIC 16F877
 viene seteado para 8 entradas analógicas y si lo que se necesitan son
 entradas y salidas digitales debemos desactivar esas entradas analógicas
 ingresando el valor que se indica en el registro ADCON

2) No olvide habilitar las resistencias de pullup internas del puerto B cargando cero en el TMR0; puede hacerlo con la instrucción CLRF OPTION_REG en el BANCO 1 ó también con la instrucción BCF OPTION_REG,7

3) Los registros SFR del PIC 16F877A ocupan hasta la posición 1F, así que las variables podrán colocarse a partir de la posición 0x20 a diferencia del PIC 16F84 en donde lo hacíamos a partir de 0x0C

4) Es una buena costumbre colocar un NOP después de actualizar un puerto para que se establezca su valor. Ejemplo:

```
MOVWF    PORTC
NOP
```

5) Cuando termine de contar las 10 cajas deberá encender RA0 y mostrar cero en ambos contadores.

6) Antes de cargar datos en el registro INTCON utilice la instrucción CLRF PORTB

Esto evita que al cargarse INCON se produzca una interrupción. Ejemplo:

```
CLRF    PORTB
MOVLW  B'10010000'
MOVWF  INTCON
```

7) El programa Principal testea el SWITCH DE INICIO; si está cerrado se queda en un LOOP y si está abierto se va a END. Ejemplo:

LOOP

BTFSC PORTB,1

GOTO FIN

GOTO LOOP

.....

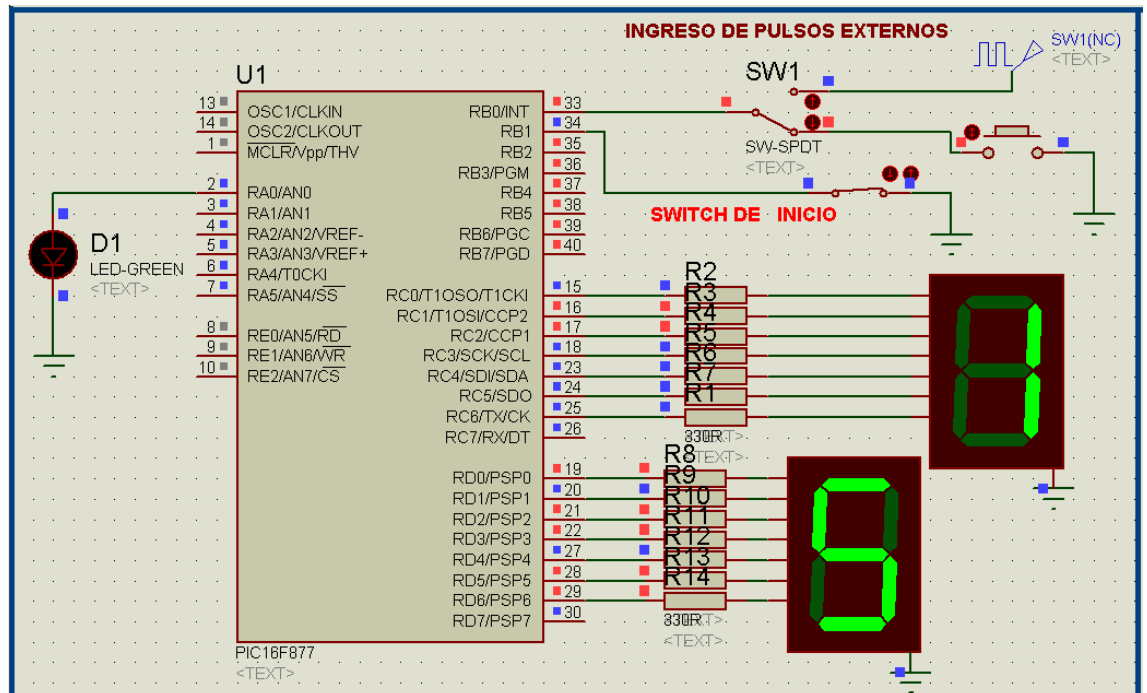
.....

.....

FIN

END

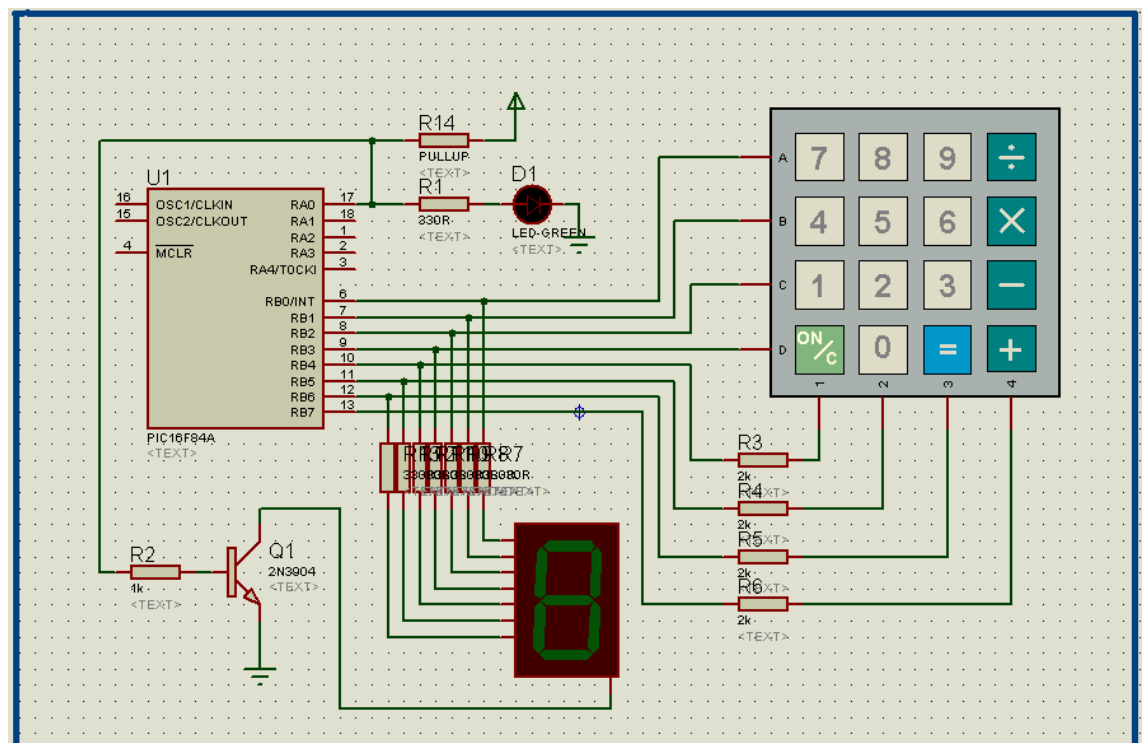
ALTERNATIVA: En la gráfica siguiente se puede apreciar el uso del pulsador manual o del reloj de pulsos seleccionados mediante un SWITCH.



4.2.5 USO DE TECLADO MATRICIAL 4x4 CON VISUALIZACIÓN EN DISPLAY DE 7 SEGMENTOS

PRACTICA 5

USO DE TECLADO MATRICIAL 4x4 CON VISUALIZACIÓN EN DISPLAY DE 7 SEGMENTOS



OBJETIVO:

Aprender el manejo del teclado con el PIC 16F84A y con el PIC 16F877

Utilizar rutinas para la eliminación de rebotes mecánicos en pulsadores

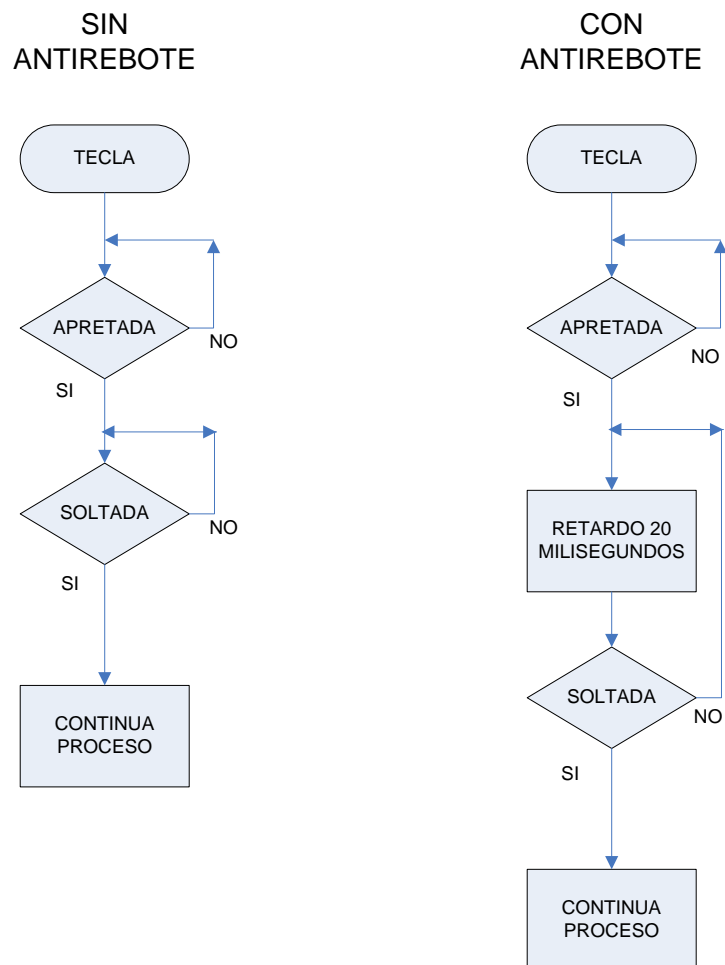
Entender el uso de las directivas del ensamblador especificadas

CONTENIDO:

- a) INTRODUCCION
- b) ESPECIFICACIONES
- c) DIAGRAMA DE FLUJO DEL TECLADO
- d) PROGRAMA PRINCIPAL
- e) ARCHIVOS INCLUDE
- f) GRABACION DE PIC
- g) PRUEBA DE DISEÑO EN HARDWARE

a) INTRODUCCION

En la Práctica anterior habrá notado que, al implementar el hardware, al pulsar la tecla se producían falsos conteos. Esto es debido a que no se eliminaron los rebotes que ocurren por el accionamiento mecánico del pulsador. Para eliminar los rebotes es necesario introducir un retardo por software de aproximadamente 20 milisegundos cuando se pulsa una tecla de la siguiente manera:

FLUJO ANTIREBOTE CON RETARDO DE 20 MILISEGUNDOS

De igual manera cuando se utiliza un teclado para evitar molestias de falsos accionamientos se emplea un antirebote por software para cada tecla.

Observe en la rutina TECLADO.ASM el uso de un retardo de 20 milisegundos.

En el diagrama de flujo del teclado no se detalló el flujo del antirrebote por cada tecla. Ud. si lo deberá hacer en su reporte.

b) ESPECIFICACIONES

a. Guarde cada uno de los archivos INCLUDE que aparecen en el siguiente programa en el mismo directorio donde creará su programa principal para que puedan ser detectados por el ensamblador.

b. Observe la forma como se colocan los archivos INCLUDE. Cada archivo include se lo guarda con extensión **.asm**

c. Comprenda la función de las directivas:

```
__CONFIG _CP_OFF&_WDT_OFF&_XT_OSC
```

```
ERRORLEVEL -302
```

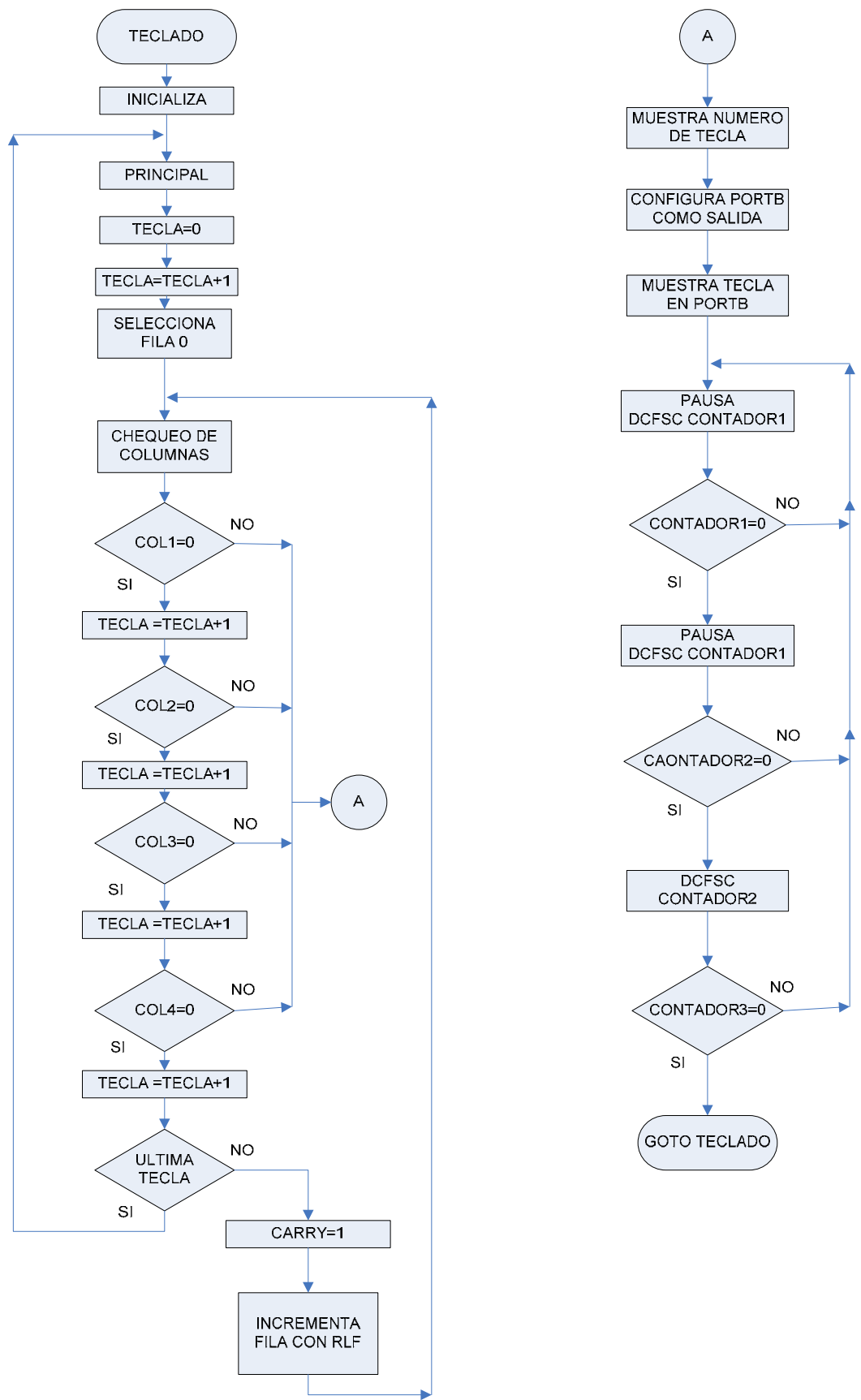
```
CBLOCK
```

```
INCLUDE
```

d. Construya el mismo circuito pero ahora con el PIC16F877

e. Es su obligación entender cada detalle de la práctica

c) DIAGRAMA DE FLUJO DEL TECLADO



e) PROGRAMA PRINCIPAL

```

;*****
;
;       TECLADO SIN REBOTES
;
;   TRATA A CADA TECLA COMO UN PULSADOR
;
;   ELIMINANDO LOS REBOTES PRODUCIDOS
;
;   POR EL ACCIONAMIENTO MECÁNICO
;*****
;
; A7.ASM
;
; Programa funciona con un teclado y un DISPLAY de 7
; segmentos en el puerto B mostrando el valor de
; la tecla apretada por el lapso aproximado de 200
; milisegundos.
;
;
;       LIST P=16F84A           ; Se usa el PIC16F84
;
;       INCLUDE "P16F84A.INC"
;
;
;       __CONFIG _CP_OFF&_WDT_OFF&_XT_OSC ;Deshabilita la
;
;               ;protección de código. Apaga el watch dog
;
;               ;Habilita el oscilador modo cristal XT
;
;
;       ERRORLEVEL -302 ;Elimina warnings por
;
;               ;cambio de banco

```

```
;CBLOCK 0X30 Indica que las variables que se  
;declaran en archivos INCLUDE comienzan en  
;la posición 0x30
```

```
CBLOCK 0X30
```

```
ENDC
```

```
ORG 0
```

```
;SETEO DE PUERTOS
```

```
BSF STATUS,RP0
```

```
CLRF TRISA
```

```
CLRF OPTION_REG
```

```
BCF STATUS,RP0
```

```
CLRF PORTA
```

```
GOTO MAIN
```

```
;COLOCACION DE ARCHIVOS INCLUDE ENTRE
```

```
;INSTRUCCIONES GOTO MAIN Y MAIN
```

```
INCLUDE PAUSA_200MS ;Para visualizar Display
```

```
INCLUDE PAUSA_20MS ;Para eliminar rebotes
```

```
INCLUDE KEY_REB.ASM
```



```
GOTO    MAIN      ;No hubo tecla apretada
BSF     PORTA,0   ;Si se apretó tecla
                        ;Habilita Display de LEDS
CALL    CONVERSION_7SEG
MOVWF   PORTB
NOP
CALL    PAUSA_200MS ;Permite ver la tecla presionada
                        ;por 200 milisegundos
BCF     PORTA,0
GOTO    MAIN

DISPLAY
BSF     STATUS,RP0 ;Habilita Display
CLRF    TRISB
CLRF    TRISA
BCF     STATUS,RP0;Apaga Display
MOVF    TECLA,W

RETURN

FIN
END
```

f) **ARCHIVOS INCLUDE**1) **PAUSA_200MS.ASM**

;Con reloj de 4Mhz produce una

;pausa de 200 milisegundos

CBLOCK

CONTADOR1

CONTADOR2

CONTADOR3

ENDC

PAUSA_200MS

MOVLW 0X0

MOVWF CONTADOR1

MOVLW 0X0

MOVWF CONTADOR2

MOVLW 0X01

MOVWF CONTADOR3

PAUSA1

DECFSZ CONTADOR1,F

GOTO PAUSA1

DECFSZ CONTADOR2,F

GOTO PAUSA1


```
DECFSZ    CONTADOR3,F
GOTO      PAUSA1
RETURN
```

2) PAUSA_20MS.ASM

;Con reloj de 4Mhz produce una
;pausa de 20 milisegundos

```
CBLOCK
CONTADOR111
CONTADOR222
ENDC
```

PAUSA_20MS

```
MOVLW    0X0
MOVWF    CONTADOR111
MOVLW    0X1A
MOVWF    CONTADOR222
```

PAUSA2

```
DECFSZ    CONTADOR111,F
GOTO      PAUSA2
DECFSZ    CONTADOR222,F
```

```
GOTO    PAUSA2  
  
RETURN
```

3) **KEY_REB.ASM**

```
*****  
;  
; SUBRRUTINAS DE TECLADO  
;  
; DEBE DEVOLVER EN TECLA  
;  
; EL NUMERO DE LA TECLA  
;  
; APRETADA  
*****
```

```
CBLOCK
```

```
TECLA
```

```
ENDC
```

```
TECLADO
```

```
CALL    KB_PORT
```

```
CALL    KB_SCAN
```

```
RETURN
```

```
KB_PORT
```

```
BSF     STATUS,RP0
```

```
MOVLW   0XF0
```

```
MOVWF TRISB
BCF STATUS,RP0
RETURN
```

KB_SCAN

```
CLRF TECLA
INCF TECLA,F
MOVLW 0X0E ;Selección de primera fila
MOVWF PORTB
NOP
```

CHEQ_COL

```
BTFSS PORTB,4
GOTO TESTB4
INCF TECLA,F
BTFSS PORTB,5
GOTO TESTB5
INCF TECLA,F
BTFSS PORTB,6
GOTO TESTB6
INCF TECLA,F
BTFSS PORTB,7
GOTO TESTB7
INCF TECLA,F
```

ULTIMA_TECLA

```
        MOVLW    0X11
        SUBWF   TECLA,W
        BTFSC   STATUS,Z
        GOTO    NTECLAS
        BSF     STATUS,C
        RLF     PORTB,F
        GOTO    CHEQ_COL

NTECLAS
        CLRF    TECLA
        RETURN

;*****
TESTB4
        CALL    PAUSA_20MS
        BTFSC   PORTB,4
        RETURN
        GOTO    TESTB4

TESTB5
        CALL    PAUSA_20MS
        BTFSC   PORTB,5
        RETURN
        GOTO    TESTB5

TESTB6
        CALL    PAUSA_20MS
        BTFSC   PORTB,6
```

```

RETURN
GOTO    TESTB6
TESTB7
CALL    PAUSA_20MS
BTFSC  PORTB,7
RETURN
GOTO    TESTB7

```

4) **CONVERSION_7SEG.ASM**

CONVERSION_7SEG

```

ADDWF  PCL,F      ; PCL + W → W
RETLW  0x3F      ; Retorna en W Código 7SEG del 0
RETLW  0x07      ; Retorna en W Código 7SEG del 7
RETLW  0x7F      ; Retorna en W Código 7SEG del 8
RETLW  0x67      ; Retorna en W Código 7SEG del 9
RETLW  0x71      ; Retorna en W Código 7SEG del F
RETLW  0x66      ; Retorna en W Código 7SEG del 4
RETLW  0x6d      ; Retorna en W Código 7SEG del 5
RETLW  0x7d      ; Retorna en W Código 7SEG del 6
RETLW  0x79      ; Retorna en W Código 7SEG del E
RETLW  0x06      ; Retorna en W Código 7SEG del 1
RETLW  0x5b      ; Retorna en W Código 7SEG del 2
RETLW  0x4f      ; Retorna en W Código 7SEG del 3

```

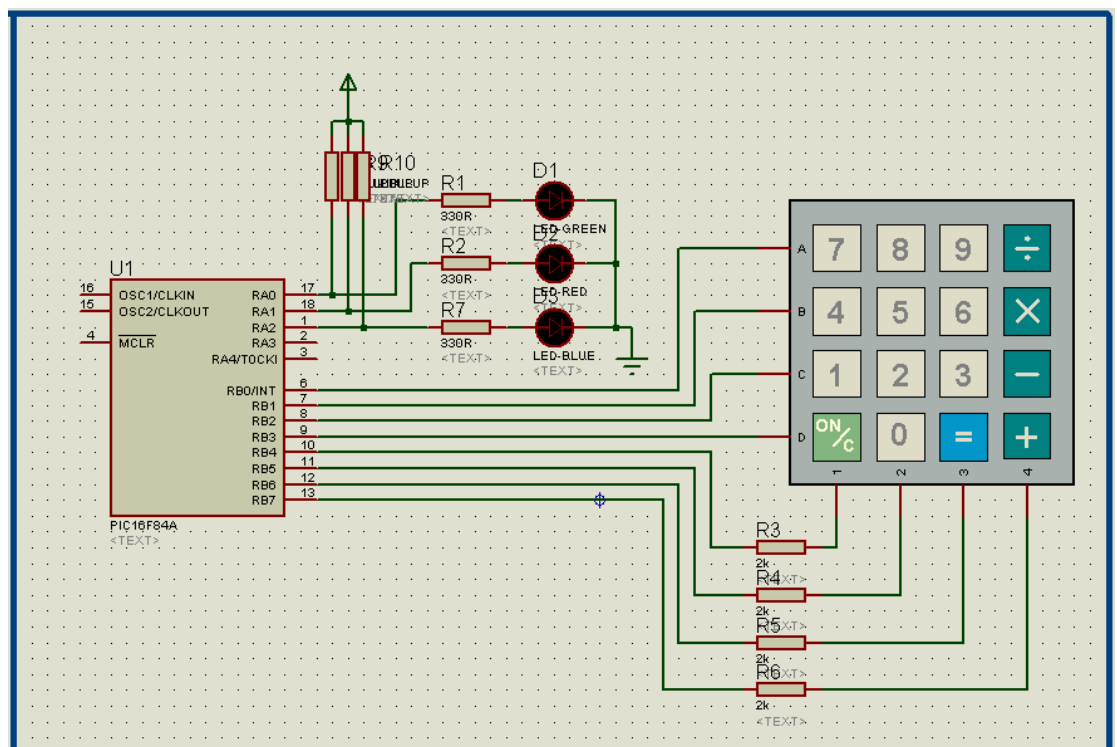
RETLW	0x5E	; Retorna en W Código 7SEG del D
RETLW	0x77	; Retorna en W Código 7SEG del A
RETLW	0x3F	; Retorna en W Código 7SEG del 0
RETLW	0x7C	; Retorna en W Código 7SEG del B
RETLW	0x39	; Retorna en W Código 7SEG del C

NOTA: Observe el archivo A7.lst y comente lo observado bajo el título de **OBSERVACION 1** en su reporte.

4.2.6 SISTEMA DE ACCESO MEDIANTE CLAVE CON LECTURA Y GRABACION DE MEMORIA EEPROM

PRACTICA 6

SISTEMA DE ACCESO MEDIANTE CLAVE CON LECTURA Y GRABACION DE MEMORIA EEPROM



OBJETIVO:

Familiarizarse con la utilización de la memoria EEPROM

CONTENIDO:

- a) ESPECIFICACIONES
- b) DIAGRAMA DE FLUJO A NIVEL ALTO
- c) PROGRAMA QUE GRABA CLAVE DE CUATRO
DIGITOS EN EEPROM
- d) ARCHIVO INCLUDE
- e) COMANDOS PARA LECTURA DE EEPROM
- f) DIAGRAMA DE FLUJO OPCIONAL
- g) GRABACION DE PIC
- h) PRUEBA DE DISEÑO EN HARDWARE

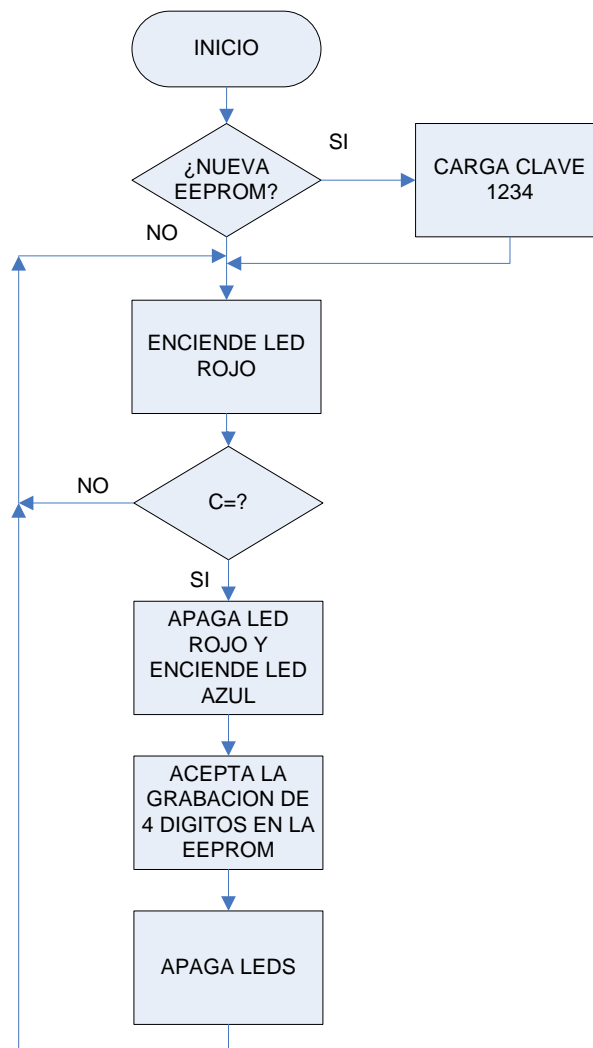
a) ESPECIFICACIONES

Se trata del diseño de un sistema digital de acceso mediante clave, basado en el microcontrolador PIC16F84A

Para la realización de este sistema se seguirán siguientes pasos:

- a) Se comprobará si la EEPROM es nueva leyendo la posición 00H de la memoria EEPROM. Si el contenido es FF quiere decir que está en blanco, entonces, automáticamente se guardará la clave 1234 en las cuatro primeras posiciones 00H, 01H, 02H y 03H de la EEPROM de datos y se encenderá un LED rojo en RA1.
- b) Si el contenido de 00H es diferente de FF quiere decir que ya tiene una clave de cuatro dígitos grabada y se encenderá un LED rojo en RA1.
- c) Al apretar la tecla C, se permite el cambio de clave, apagándose el LED rojo y encendiéndose un LED azul en RA2. Al terminar de grabar la nueva clave de cuatro dígitos se apagará el LED azul y se volverá a encender el LED rojo.

b) DIAGRAMA DE FLUJO



c) PROGRAMA QUE GRABA CLAVE DE CUATRO DIGITOS EN EEPROM

; PROGRAMA QUE PERMITE GRABAR UNA CLAVE DE 4 DIGITOS
 ; EN C1, C2, C3 Y C4. ADEMAS SE LE ADICIONO LA
 ; CAPACIDAD DE GRABAR LA MEMORIA EEPROM UTILIZANDO
 ; EL ARCHIVO INCLUDE GRABA_EEPROM

```

LIST p=16F84A           ; Se usa el PIC16F84
INCLUDE P16F84A.INC
  
```

```
__CONFIG _CP_OFF&_WDT_OFF&_XT_OSC
```

```
CONTADO1      EQU      0X0C
CONTADO2      EQU      0X0D
CONTADO3      EQU      0X0E
```

```
;NUEVAS VARIABLES
```

```
C1      EQU      0X10
C2      EQU      0X11
C3      EQU      0X12
C4      EQU      0X13
```

```
CBLOCK 0X30
```

```
ENDC
```

```
ORG      0
```

```
BSF      STATUS,RP0
```

```
CLRF    TRISA
```

```
CLRF    OPTION_REG
```

```
BCF      STATUS,RP0
```

```
CLRF    PORTA
```

```
GOTO P1
```

```
INCLUDE  "key_reb.asm"
```

```
INCLUDE  PAUSA_20MS ;Pausa antirrebote
```

```
INCLUDE  PAUSA_200MS
```

```
INCLUDE  GRABA_EEPROM
```



```

    BTFSC STATUS, Z
    GOTO P3

    MOVF  TECLA, W
    MOVWF C3
    MOVWF DATA_L
    MOVLW 0X03
    MOVWF ADDR_L
    CALL  GRABA_EEPROM

P4
    CALL  TECLADO
    NOP                                     ;Número de tecla apretada en
TECLA
    BTFSC STATUS, Z
    GOTO P4
    MOVF  TECLA, W
    MOVWF C4
    MOVWF DATA_L
    MOVLW 0X04
    MOVWF ADDR_L
    CALL  GRABA_EEPROM

    BSF          PORTA, 0
    CALL  PAUSA
    BCF          PORTA, 0
    GOTO P1

PAUSA
    MOVLW          0X0
    MOVWF CONTADO1
    MOVLW          0X0
    MOVWF CONTADO2

```

```

        MOVLW      0X01
        MOVWF     CONTADO3
PAUS1
        CLRWDI
        DECFSZ    CONTADO1 ,F
        GOTO     PAUS1
        DECFSZ    CONTADO2 ,F
        GOTO     PAUS1
        DECFSZ    CONTADO3 ,F
        GOTO     PAUS1
        RETURN
FIN
        END

```

d) ARCHIVO INCLUDE GRABA_EEPROM.ASM

```

;Graba EEPROM indicar dirección y dato
        CBLOCK
        ADDR_L
        DATA_L
        ENDC
GRABA_EEPROM
        MOVF     ADDR_L ,W
        MOVWF   EEADR
        MOVF     DATA_L ,W
        MOVWF   EEDATA
        BSF     STATUS ,RP0
        BSF     EECON1 ,WREN
        BCF     INTCON ,GIE
        MOVLW   0X55

```

```
MOVWF EECON2

MOVLW 0XAA

MOVWF EECON2

BSF      EECON1,WR

BSF      INTCON,GIE

CALL    PAUSA_200MS

BCF      EECON1,EEIF

BCF      EECON1,WREN

BCF      STATUS,RP0

RETURN
```

e)COMANDOS PARA LECTURA DE EEPROM

;LECTURA DE EEPROM

```
LEE_EEPROM

BSF    STATUS,RP1

BCF    STATUS,RP0

MOVF   ADDR_L,W

MOVWF  EEADR

BSF    STATUS,RP0

BSF    EECON1,EEPGD

BSF    EECON1,RD

NOP

NOP

BCF    STATUS,RP0

MOVF   EEDATA,W

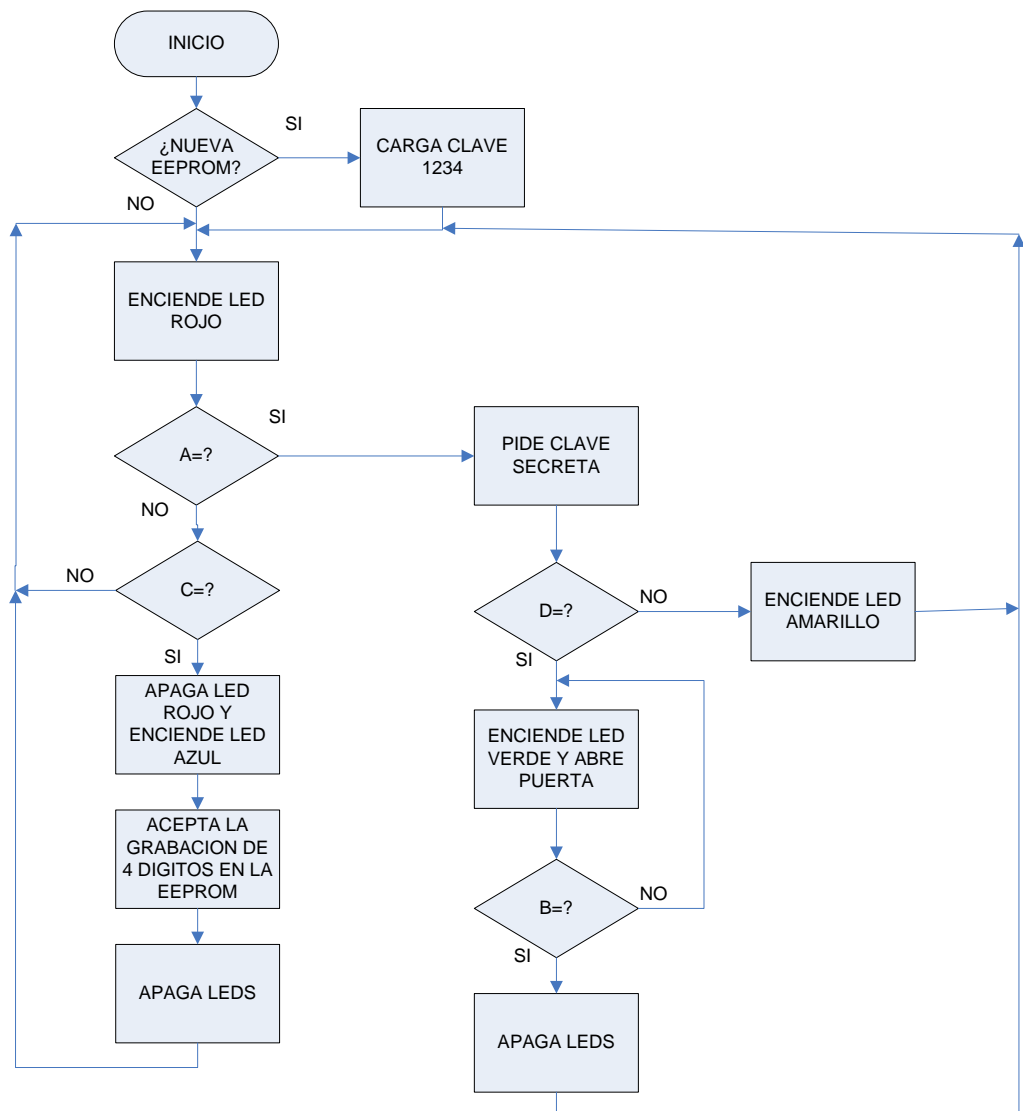
MOVWF  DATA_L

RETURN
```

f) DIAGRAMA DE FLUJO OPCIONAL

El diagrama de flujo que se encuentra a continuación se utilizará en esta práctica de manera opcional, en el se muestra un sistema de control de acceso a una oficina.

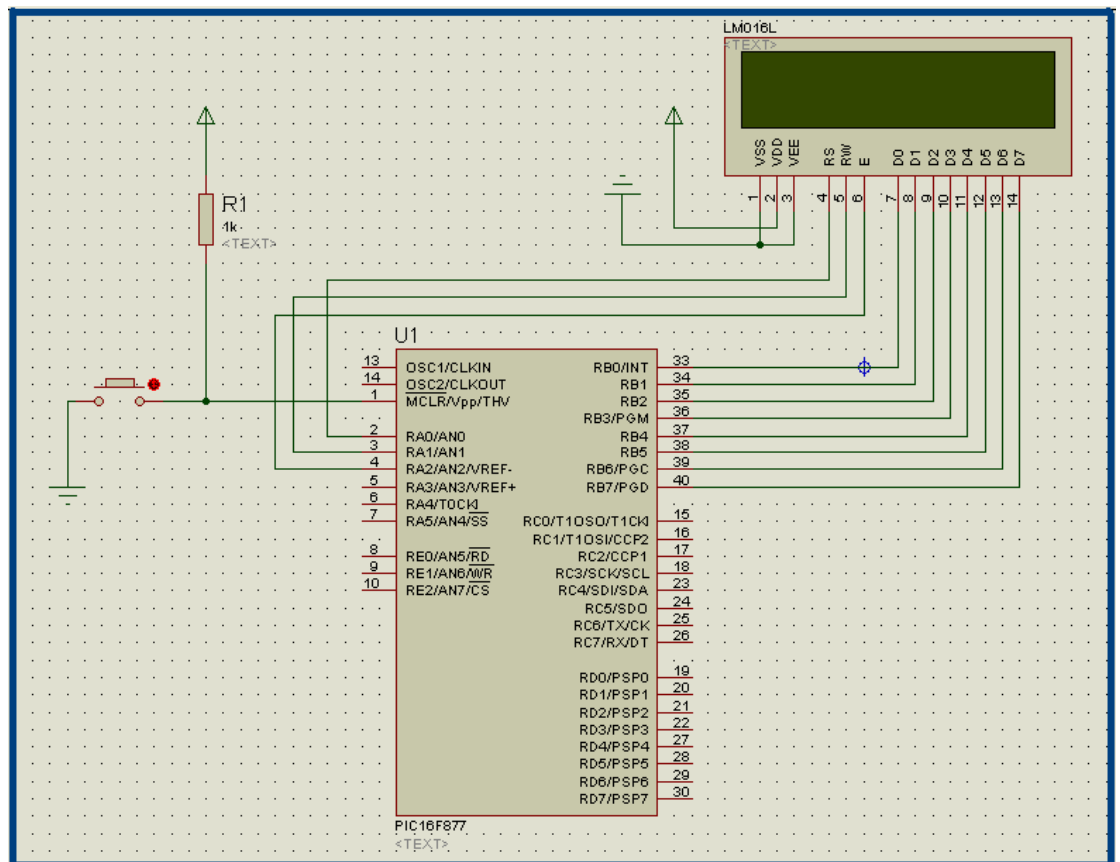
También como trabajo opcional se plantea que el sistema pida el ingreso de la clave anterior para permitir el cambio de clave.



4.2.7 MANEJO DE DISPLAY LCD Y SUS PRINCIPALES FUNCIONES

PRACTICA 7

MANEJO DE DISPLAY LCD Y SUS PRINCIPALES FUNCIONES



OBJETIVO:

Familiarizarse con la utilización del DISPLAY LCD

CONTENIDO:

- a) ESPECIFICACIONES
- b) PROGRAMA QUE PERMITE LA VISUALIZACION DE MENSAJES SOBRE UN LCD
- c) ARCHIVO INCLUDE
- d) COMANDOS DEL LCD
- e) TAREA OPCIONAL
- f) GRABACION DE PIC

a) ESPECIFICACIONES

Especificaciones: Se trata del diseño de un sistema digital basado en el microcontrolador PIC16F877 con las especificaciones de funcionamiento siguientes:

- Se proporciona un programa que permite la visualización de caracteres en dos líneas de un DISPLAY LCD
- Corra el programa
- Usando el MPLAB revise paso a paso todas las rutinas del DISPLAY LCD con sus comentarios.
- Reemplace los mensajes por:

INGRESE LA CLAVE CORRECTA (1ª. línea)

..... (2ª. línea)

- Mediante un SWITCH colocado en RC0 y utilizando los comandos del DISPLAY LCD haga: Con SWITCH abierto, que encienda un LED amarillo en RC1 y roten los mensajes hacia la izquierda. Con SWITCH cerrado, que encienda un LED rojo en RC2 y roten los mensajes hacia la derecha.
- Haga que la configuración anterior funcione utilizando un bus de datos de 4 bits (RB4 – RB7). Ver ejemplo de Proteus, en el menú HELP --> Simple Designs --> MICROPROCESSORS SIMULATION SAMPLES --> PIC 16 AND LCD IN 4 BIIT MODE.

Opcional: Adicione un teclado matricial de 4x4 en puerto B de forma multiplexada con el LCD y haga que los caracteres digitados en el teclado aparezcan en la LCD.

b)PROGRAMA QUE PERMITE LA VISUALIZACION DE MENSAJES SOBRE UN

LCD

```
*****
;
```

```
;          LCD CON EL 16F877
```

```
*****
```

```
;
```

```
;PROGRAMA QUE MUESTRA TEXTO EN LA LINEA 1 Y
```

```
; EN LA LINEA 2 DE UN LCD 2X16
```

```
          LIST p=16F877          ;Indica al ensamblador el pic a
usarse
```

```
          include "P16F877.inc"   ;incluye definiciones del chip
          __CONFIG _CP_OFF&_WDT_OFF&_XT_OSC
```

```
LCD_VAR      EQU      0X32
```

```
TEMP         EQU      0X33
```

```
*****
```

```
          cblock 0x20             ;inicio de registros de propósito general
```

```
          count                   ;Igual a: count EQU 0x20
```

```
          endc
```

```
*****
```

```
ORG      0

GOTO INICIO

INCLUDE "LX1.ASM"

INICIO   call   UP_LCD_877

         call   LCD_INI           ;inicializa LCD

         call   DISPLAY_ON_CUR_OFF

linea1   call   LCD_LINE1

         clrf   count           ;contador a cero, offset TEXTO1

mensaje1 movf   count,w          ;valor de contador a w

         call   TEXT1          ;obtener caracter desde tabla

         xorlw  0x00           ;es igual a 0?

         btfsc STATUS,Z

         goto  linea2

         call   LCD_DATO

         incf   count,f

         goto  mensaje1

linea2   call   LCD_LINE2       ;mover a segunda fila, primera
columna
```

```

                                clrf    count                ;contador a 0, offset de
TEXTO2
mensaje2  movf    count,w                ;contador en w
                                call    TEXT2                ;obtener caracter desde
tabla
                                xorlw   0x00                ;es igual a cero?
                                btfsc  STATUS,Z
                                goto   fin_mensaje
                                call   LCD_DATO
                                incf   count,f
                                goto   mensaje2

fin_mensaje goto line1

```

```

;*****
;

```

```

; TABLAS DE TEXTO

```

```

HEX_TABLE  ADDWF  PCL,f
                                RETLW  0x30
                                RETLW  0x31
                                RETLW  0x32
                                RETLW  0x33
                                RETLW  0x34

```

```
RETLW 0x35
RETLW 0x36
RETLW 0x37
RETLW 0x38
RETLW 0x39
RETLW 0x41
RETLW 0x42
RETLW 0x43
RETLW 0x44
RETLW 0x45
RETLW 0x46
```

```
TEXT1                                addwf PCL, f
                                     retlw 'h'
                                     retlw 'o'
                                     retlw 'l'
                                     retlw 'a'
                                     retlw ''
                                     retlw 'E'
                                     retlw 's'
                                     retlw 'p'
                                     retlw 'o'
                                     retlw 'l'
                                     retlw 0x00
```


e) TAREA OPCIONAL

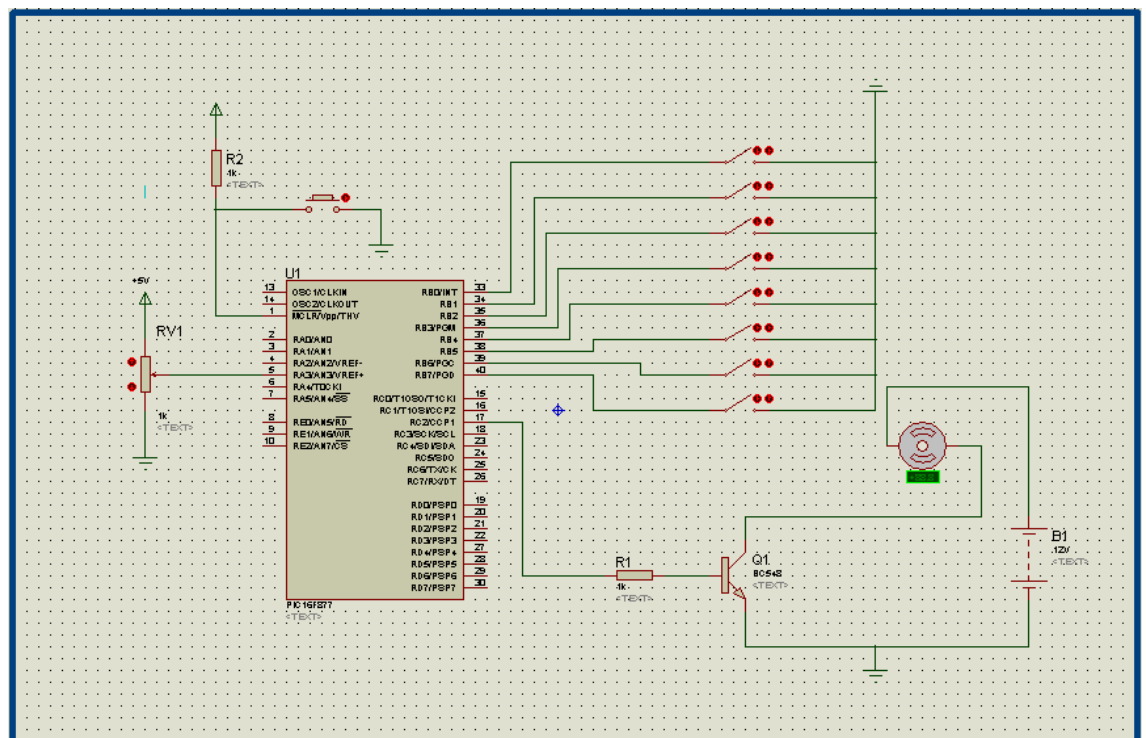
Conecte un teclado matricial 4x4 multiplexado con el DISPLAY LCD y haga que se muestre en el LCD los caracteres ingresados por el teclado

f) GRABACION DE PIC

4.2.8 VELOCIDAD DE MOTOR DC CONTROLADA MEDIANTE PWM CUYO CICLO DE TRABAJO ES EL RESULTADO DE UNA CONVERSIÓN ANALÓGICO DIGITAL (ADC)

PRACTICA 8

VELOCIDAD DE MOTOR DC CONTROLADA MEDIANTE PWM CUYO CICLO DE TRABAJO ES EL RESULTADO DE UNA CONVERSIÓN ANALÓGICO DIGITAL (ADC)



OBJETIVO:

Combinar el modo de trabajo PWM de los módulos CCP del PIC16F877 con el módulo de Conversión Analógica Digital (ADC) para regular, mediante PWM, la velocidad de un motor DC con ayuda de un potenciómetro analógico.

CONTENIDO:

- a) ESPECIFICACIONES
- b) PROGRAMA QUE PERMITE LA VISUALIZACION DEL EQUIVALENTE BINARIO DEL VOLTAJE APLICADO EN LA ENTRADA DEL ADC EN UN LCD
- c) PROGRAMA QUE CONTROLA UN MOTOR DC MEDIANTE LA TECNICA PWM
- d) TAREA OPCIONAL
- e) GRABACION DE PIC

a) **ESPECIFICACIONES**

Use la rutina proporcionada del módulo de Conversión Analógico Digital (ADC) hágalo trabajar y estudie detenidamente cada uno de los comandos

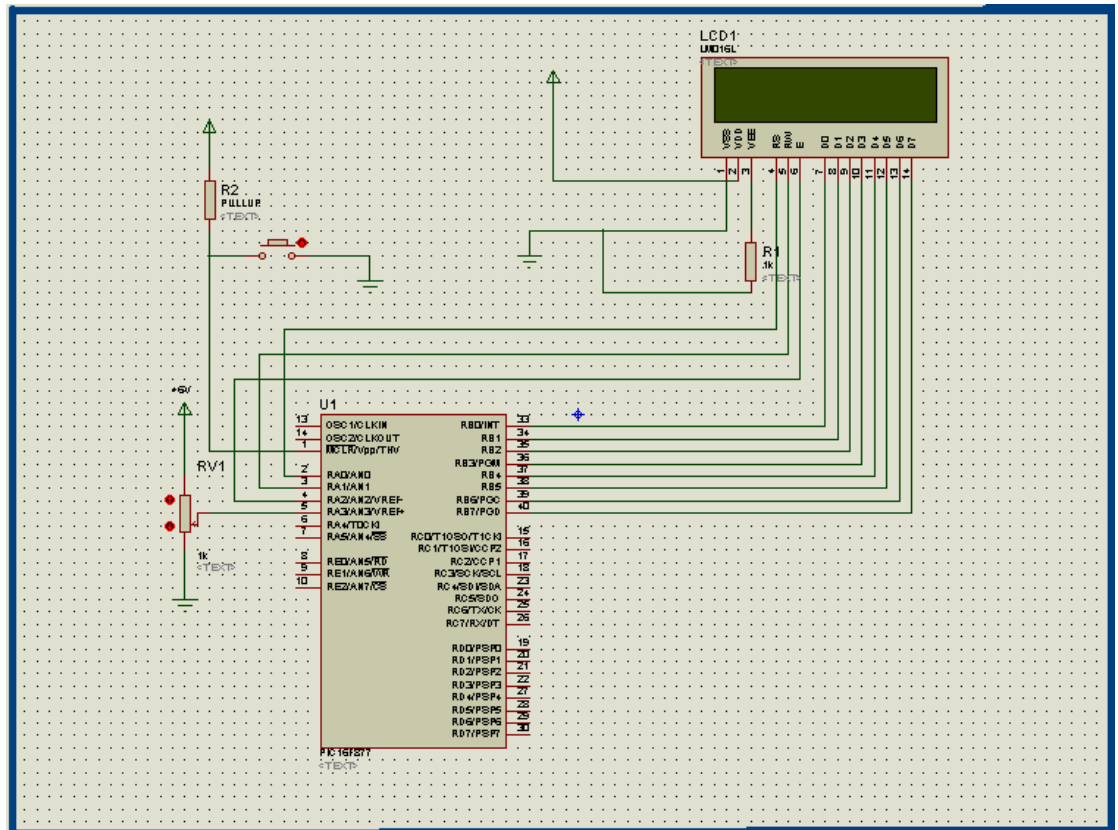
Use la rutina proporcionada del módulo PWM hágalo trabajar y estudie detenidamente cada uno de los comandos

Combine los módulos ADC y PWM de la siguiente manera:

El módulo CCP1 se empleará como salida de señal PWM (patita RC2) para regular la velocidad del motor DC. El período de esta señal es de 8 milisegundos (el ejercicio resuelto trabaja con período de 4 milisegundos) y el ciclo de trabajo (Duty Cycle) variará en función del valor analógico suministrado por un potenciómetro variable conectado a la patita RA3. Para la precisión o resolución del ciclo de trabajo se emplean 8 bits.

El resultado de la conversión análogo digital se almacena en el registro CCPR1L para establecer el ciclo de trabajo y por lo tanto controlar la velocidad del motor (Usar solo 8 BITS).

b)PROGRAMA QUE PERMITE LA VISUALIZACION DEL EQUIVALENTE BINARIO DEL VOLTAJE APLICADO EN LA ENTRADA DEL ADC EN UN LCD



```

;*****
;
;          CONVERTIDOR ADC CON EL 16F877
;*****
;
;A15.asm
;
;Los dispositivos PIC16F87X disponen de un convertidor
;A/D de 10 bits de resolución y 5 u 8 canales de
;entrada analógica. La tensión de referencia determina
;la resolución por bit:(Resolución = Vref/1024).
;Con Vref=5V Resol=4.88 mV/Bit
;con 2.5 Vref=2.5 V Resol= 2.44 mV/Bit
    
```

```

;
;Este ejemplo realiza la conversión de la tensión
;presente en el canal RA3/AN3, que se puede variar
;con un potenciómetro de 1K. El resultado de la
;conversión se visualiza en binario sobre la pantalla
;LCD. El reloj de conversión se fija en Fosc/32.

                                LIST  p=16F877           ;Tipo de procesador
                                INCLUDE  "P16F877.INC"     ;Registros
internos

                                __CONFIG _CP_OFF&_WDT_ON&_XT_OSC

LCD_VAR          EQU    0x20           ;Variables de las rutinas
LCD
NUMCHAR          EQU    0x23           ;Contador de caracteres
                                                ;enviados a LCD

                                ORG    0x00           ;Vector de Reset
                                GOTO  Inicio

                                ORG    0x05           ;Evita el vector de
interrupción

                                INCLUDE  "LX1.ASM"       ;Incluye rutinas de manejo del
LCD

;Visualiza: esta subrutina justifica a la izquierda
;los 10 bits resultantes de la conversión en ADRESH
;y ADRESL. Los convierte a caracteres ASCII (0 o 1)
;y los visualiza sobre el LCD.

;NOTA: Perro guardian activo

```

```

;-----
;Programa principal

Inicio

    BSF        STATUS,RP0    ;Banco 1
    BCF        STATUS,RP1    ;
    MOVLW     b'00001111'
    MOVWF     OPTION_REG    ;Preescaler de 128 asociado al
WDT
    BCF        STATUS,RP0    ;Regreso a banco 0

;Se activa el ADC y se selecciona el canal RA3/AN3.
Frecuencia de trabajo Fosc/32

    MOVLW     b'10011001'
    MOVWF     ADCON0        ;ADC en On, selecciona canal AN3

Loop

    CLRWDT
    BSF        STATUS,RP0    ;Banco 1
    BCF        STATUS,RP1    ;
    CLRWF     ADCON1        ;Puerta A entradas analógicas,
Vref=Vcc
    BCF        STATUS,RP0    ;Banco 0
    BCF        PIR1,ADIF    ;Restaura el flag del conversor
AD
    BSF        ADCON0,GO    ;Inicia la conversión

ADC_Wait

    BTFSS    PIR1,ADIF    ;Fin de conversión ??

```

```

        GOTO  ADC_Wait      ;Todavía no
        BSF   STATUS,RP0   ;Banco 1

;El resultado de la conversión debe estar en ADRESH y ADRESL
;Ahora es necesario publicar el resultado en el LCD
;Las líneas de la Puerta A se reconfiguran como salidas
;digitales

        MOVLW b'00000110'
        MOVWF ADCON1      ;Puerta A digital
        BCF   STATUS,RP0   ;Banco 0

;Se juntaron las subrutinas de activación del LCD

        CALL  LCD
        CALL  Visualiza   ;Visualiza el resultado de la
conversión
        GOTO  Loop

LCD
        CALL  UP_LCD_877  ;Configura E/S para el LCD
        CALL  LCD_INI     ;Inicializar la pantalla LCD
        MOVLW b'00001100'
        CALL  LCD_REG     ;LCD On, cursor y blink Off
        RETURN

Visualiza:
        MOVLW 0x80
        CALL  POS_CURSOR  ;Cursor en primera fila primera
columna

```



```

        MOVLW d'10'          ;Diez caracteres(10 bits)
        MOVWF NUMCHAR        ;a visualizar en el LCD

Visual_loop

        BSF   STATUS,RP0     ;Voy a banco 1
        RLF   ADRESL,F       ;Rotamos ADRESL a la izquierda a
través de CF
        BCF   STATUS,RP0     ;Regreso abanco 0
        RLF   ADRESH,F       ;Rotación izquierda de ADRESH a
través de CF
        BTFSC STATUS,C      ;Prueba el bit en CF a visualizar
        GOTO  Bit_1         ;Está a 1
        MOVLW '0'          ;Está a 0
        GOTO  Visu_1

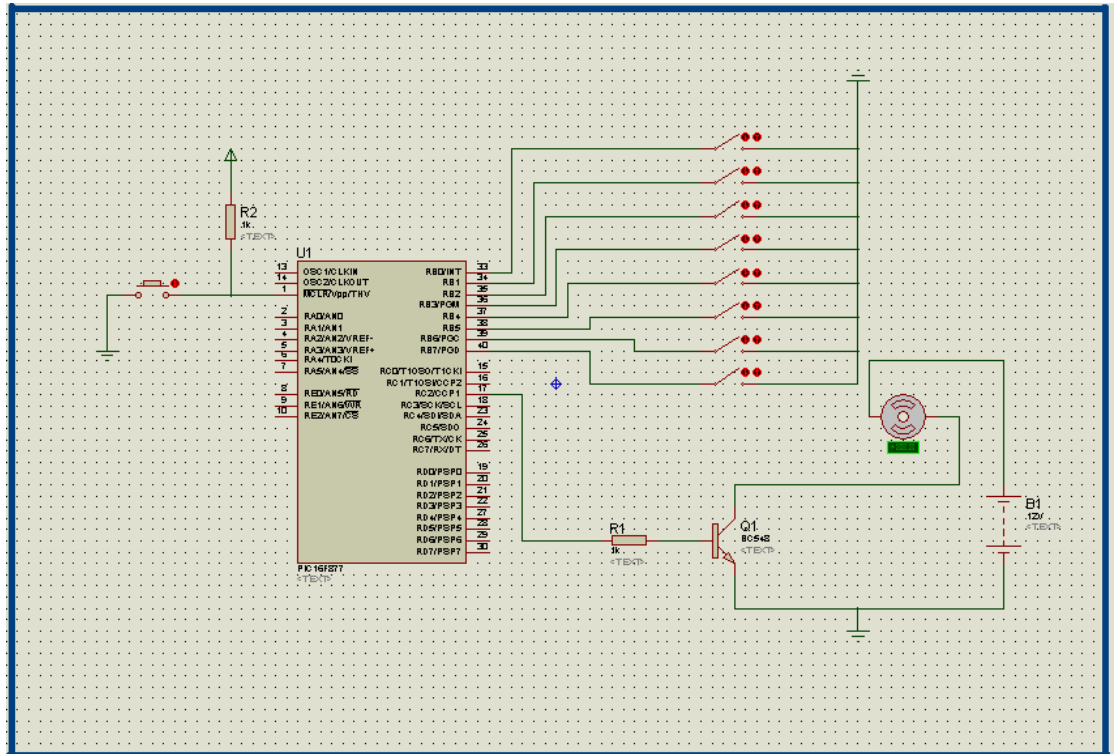
Bit_1
        MOVLW '1'

Visu_1
        CALL   LCD_DATO     ;Visualiza el "0" o el "1"
sobre el LCD
        DECFSZ NUMCHAR,F    ;Siguiete caracter
        GOTO   Visual_loop
        RETURN

        END                ;Fin del programa fuente
;-----

```

c) PROGRAMA QUE CONTROLA UN MOTOR DC MEDIANTE LA TECNICA PWM



```

;*****
; MODULACION POR ANCHO DE PULSO "PWM"
; CONTROL DE VELOCIDAD DE MOTOR DC
;*****
; A13.ASM
;
;Uso de Los módulos CCPx en modo PWM
;Modulación de ancho de pulsos
;
;Este ejercicio consiste en generar una señal
;de onda cuadrada por la línea RCx/CCPx
;cuyo periodo y ancho de pulso (Duty Cycle)
;pueden ser modificados
;El periodo se determina según la fórmula
    
```

```

; T(us)=(PR2+1)(4*Tosc)(Preescalador del TMR2)

;La duración del ancho del pulso (Duty Cycle)
;se determina según:

;d=(CCPR1L:CCPCON1<5:4>)(Tosc)(Preescalador del TMR2)

;Este ejemplo emplea al módulo CCP1 con salida
;de señal por la línea RC2/CCP1 y un preescaler
;de 16. La señal de salida tiene un periodo de
;4 mseg. La anchura del pulso es variable
;y, se determina, según el valor binario de los
;interruptores RB0-RB7.

;NOTA: En este programa habilitamos el Perro
;guardián (WDT)porque se lo usa dentro de un
;lazo

LIST p=16F877 ;Tipo de procesador
INCLUDE "P16F877.INC" ;Definiciones de registros
internos
__CONFIG __CP_OFF&_WDT_ON&_XT_OSC

;DEFINICION DE VARIABLE

TEMPORAL EQU 0x20 ;Variable temporal

;DEFINICION DE CONSTANTE EN DECIMAL

PER EQU d'250' ;Periodo de 250. Para reloj de 4Mhz

```

```
;T=(250)(Preescalador de 16)=4mS
```

```

ORG    0x00          ;Vector de Reset

GOTO  INICIO

;PROGRAMA PRINCIPAL

INICIO

    BSF    STATUS,RP0

    BCF    STATUS,RP1 ;Banco 1

    MOVLW b'00000110'

    MOVWF  ADCON1     ;Puerta A E/S digitales

    BCF    TRISC,2    ;RC2 salida PWM

    MOVLW b'00001111'

    MOVWF  OPTION_REG

    MOVLW PER-1

    MOVWF  PR2

    BCF    STATUS,RP0 ;Selecciona banco 0

;Módulo CCP1 en modo PWM

    MOVLW b'00001100'

    MOVWF  CCP1CON

;Preescaler TMR2 1:16 (BITS 0 Y 1) y TMR2ON (BIT 2)

    MOVLW b'00000111'

    MOVWF  T2CON

LOOP

    CLRWDT            ;Refresca el WDT

```

```

MOVWF PORTB,W      ;Lee valor de switches
MOVWF CCPR1L      ;Carga ancho de pulso (n*Prescaler de
16)
GOTO LOOP          ;Bucle infinito

END                ;Fin del programa fuente

;*****

```

d) TAREA OPCIONAL

La pantalla LCD deberá mostrar en términos de tanto por ciento, la anchura del ciclo de trabajo que se aplica en todo momento al motor.

Ejemplo: **MOTOR DC**

C. TRABAJO: 60%

Variando la posición del potenciómetro el porcentaje debe cambiar, por ejemplo a:

MOTOR DC

C. TRABAJO: 80%

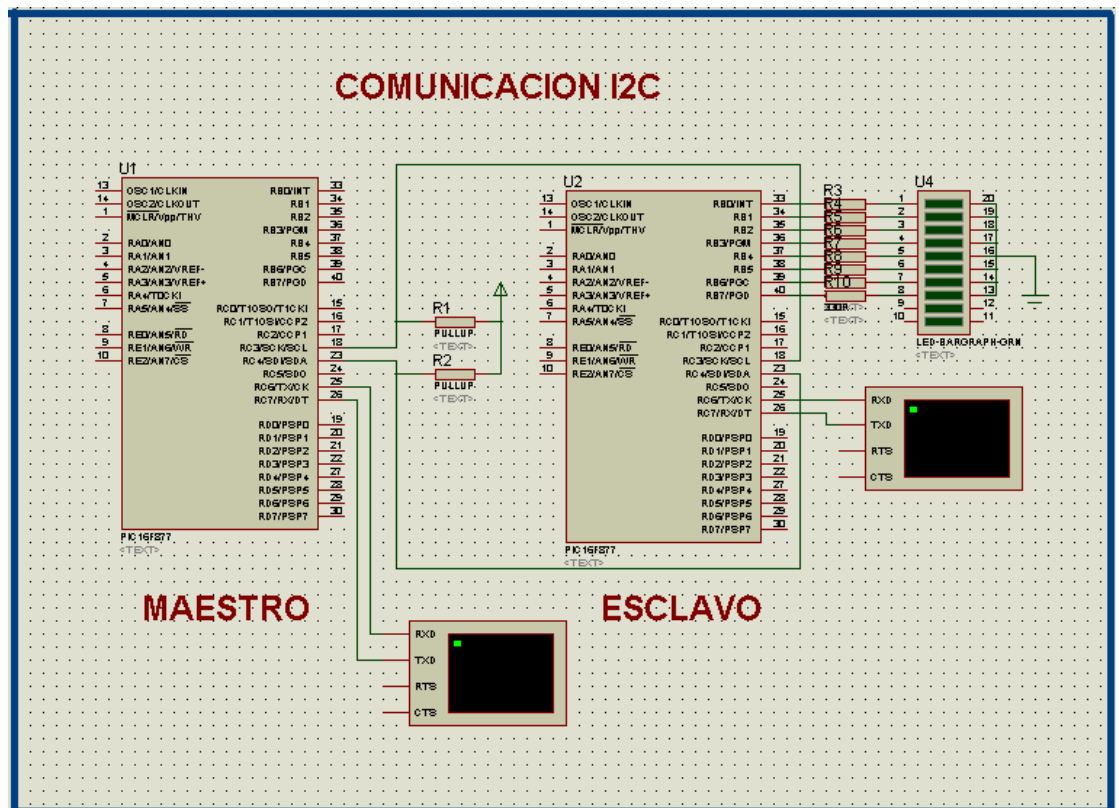
En esta parte al trabajar con el DISPLAY LCD, la puerta B se emplea para enviar comandos y datos a la pantalla LCD, por lo tanto debe configurarse como salida. Para el control del módulo LCD 2x16 use lo siguiente: RA0 controla RS, RA1 controla R/W y RA2 controla E.

e) GRABACION DEL PIC

4.2.9 COMUNICACIÓN I2C MAESTRO ESCLAVO ENTRE DOS MICROCONTROLADORES

PRACTICA 9

COMUNICACIÓN I2C MAESTRO ESCLAVO ENTRE DOS MICROCONTROLADORES



OBJETIVO:

Familiarizarse con las herramientas de software para programar

PICs en Lenguaje C

Aprender a comunicar PICs usando la comunicación serial I2C

CONTENIDO:

a) ESPECIFICACIONES

b) INSTRUCCIONES A SEGUIR

c) PROGRAMA FUENTE DEL MAESTRO: i2C1.C

d) PROGRAMA FUENTE DEL ESCLAVO: i2C2.C

e) GRABACION DE PIC

f) PRUEBA DE DISEÑO EN HARDWARE

a) ESPECIFICACIONES

Se usarán dos PICs 16F877 conectados por medio de dos cables más tierra.

El primer PIC tendrá un teclado y una pantalla virtual para visualizar los caracteres enviados a través de la interfaz. El segundo PIC tendrá una pantalla virtual y un BARGRAPH para visualizar los caracteres recibidos.

El estudiante deberá compilar los programas i2c1.c e i2c2.c.

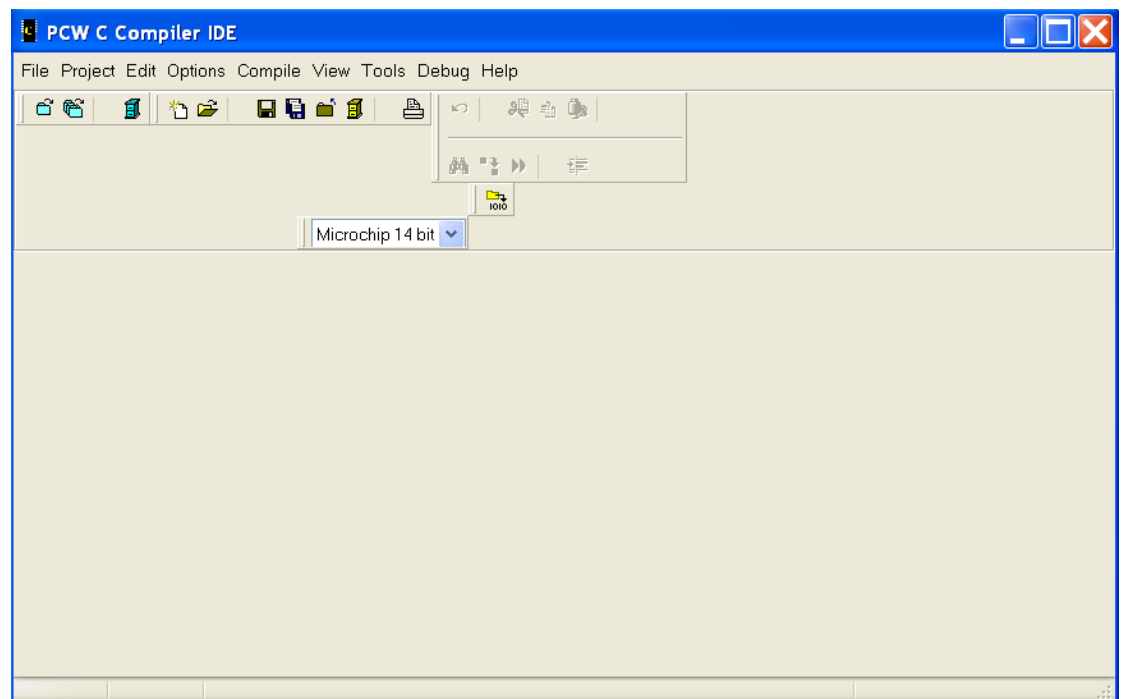
Luego incorporarlos en el archivo de PROTEUS y hacerlo trabajar.

Lo más importante de esta práctica es la familiarización con el software y el entender la comunicación I2C.

b) PROCEDIMIENTO A SEGUIR:

1) Por el momento lo que hará es seleccionar **Todos los Programas → PIC**

C Compiler obteniendo la ventana PCW C Compiler IDE:



2) Desde FILE→OPEN abra el programa adjunto i2c1 que consta de dos partes necesarias para su compilación:

i2c1.c

i2c1.h

En **i2c1.c** en el archive `#include` deberá arreglar la ruta de directorios que permita llegar a su archivo **i2c1.h**

3) Utilizando **Compile→Compile F9** va a compilar el programa **i2c1** el mismo que creará el archivo **i2c1.hex** en el directorio escogido para trabajar

4) Repita los procedimientos 2 y 3 anteriores para el archivo **i2c2** y obtenga el archivo **i2c2.hex**

2) Los archivos **i2c1.hex** e **i2c2.hex** (obtenidos anteriormente) serán usados en los microcontroladores maestro y esclavo del archivo **i2c** de Proteus

3) Desde el ISIS de Proteus ejecute el programa **i2c** y observe su comportamiento para diversos caracteres ingresados

4) Revise el menú **HELP** especialmente:

HELP→Index

HELP→Build in functions

Con **HELP→Index** se ayudará para encontrar la información que se busque

5) Asegúrese de entender cada sentencia de la programación en lenguaje C.

Use el **HELP** de Proteus

6) En clase estudiaremos la forma de utilizar el Projet→New→Pic Wizard para la creación de nuevos proyectos. Si se siente capaz de hacerlo o investigarlo hágalo ahora.

c) PROGRAMA FUENTE DEL MAESTRO: i2c1.c e i2c1.h

// PROGRAMA 12c1.c

```
#include "C:\Documents and Settings\cv\Mis documentos\Ej_C\i2c1.h"
```

```
// TRANSMISION I2C
```

```
// PROGRAMACION DEL MAESTRO
```

```
// SE NECESITA, EN EL WIZARD, SETEO COMO MAESTRO
```

```
// Y HABILITAR LA COMUNICACION I2C.
```

```
// SI SE QUIERE RS-232 TAMBIEN SE LA PUEDE HABILITAR
```

```
// TAL COMO OCURRE EN ESTE EJEMPLO.
```

```
// c SE DEFININE COMO char PARA RECOGER EL DATO RECIBIDO
```

```
// DEL TECLADO MEDIANTE EL COMANDO c=getchar().
```

```
// RECIBIDO EL DATO SE LO ENVIA CON i2c_write(c)
```

```
void main()
```

```
{
```

```
char c,d;
```

```
printf("Press any key to start:\r");
```

```
// EL SIGUIENTE GRUPO DE PARAMETROS SE CARGAN AL INICIO
```

```
// DURANTE LA CONFIGURACION DEL PROGRAMA MEDIANTE EL WIZARD
```

```
port_b_pullups(TRUE);
setup_adc_ports(NO_ANALOGS);
setup_adc(ADC_OFF);
setup_psp(PSP_DISABLED);
setup_timer_0(RTCC_INTERNAL|RTCC_DIV_1);
setup_timer_1(T1_DISABLED);
setup_timer_2(T2_DISABLED,0,1);

// FIN DE PARAMETROS

while (1)
{
    c=getchar();
    printf("caracter escrito es: ");
    putchar(c);

    i2c_start();
    i2c_write(c); // Se envía dato al esclavo
    i2c_stop();

    printf(" --> enviado exitosamente\r");
}

}

// PROGRAMA i2c1.h

#include <16F877.h>
```

```

#device adc=8

#FUSES NOWDT           //No Watch Dog Timer
#FUSES XT              //Crystal osc <= 4mhz
#FUSES NOPUT          //No Power Up Timer
#FUSES NOPROTECT      //Code not protected from reading
#FUSES BROWNOUT       //Reset when brownout detected
#FUSES LVP            //Low Voltage Programming on B3(PIC16) or B5(PIC18)
#FUSES NOCPD          //No EE protection
#FUSES NOWRT          //Program memory not write protected
#FUSES NODEBUG        //No Debug mode for ICD

#use delay(clock=4000000)

#use rs232(baud=9600,parity=N,xmit=PIN_C6,rcv=PIN_C7,bits=9)

#use i2c(Master,Slow,sda=PIN_C4,scl=PIN_C3)

```

d) PROGRAMA FUENTE DEL ESCLAVO: i2c2.c e i2c2.h

// PROGRAMA i2c1.c

```
#include "C:\Documents and Settings\cv\Mis documentos\Ej_C\i2c2.h"
```

// TRANSMISION I2C

// PROGRAMACION DEL ESCLAVO

// SE NECESITA, EN EL WIZARD, DEFINIR UNA DIRECCION

// Y HABILITAR LA COMUNICACION I2C.

// SI SE QUIERE RS-232 TAMBIEN SE LA PUEDE HABILITAR

// TAL COMO OCURRE EN ESTE EJEMPLO.

// c SE DEFININE COMO char PARA RECOGER EL DATO RECIBIDO

```
// MEDIANTE EL COMANDO c=I2C_READ().

// UNA VEZ RECIBIDO EL DATO SE LO PUBLICA EN EL PUERTO B
// AL CUAL, EN EL WIZARD, SE LE HABILITÓ LAS RESISTENCIAS
// DE PULLUP Y SE LAS DESIGNO COMO SALIDA. /*

void main()

{

char c;

printf("ok\r");

while(1)

{

printf("Listo para recibir: ");

c=I2C_READ();

putchar(c);

printf(" recibido\r");

output_B(c);

}

// EL SIGUIENTE GRUPO DE PARAMETROS SE CARGAN AL INICIO
// DURANTE LA CONFIGURACION DEL PROGRAMA MEDIANTE EL WIZARD

port_b_pullups(TRUE);

setup_adc_ports(NO_ANALOGS);

setup_adc(ADC_OFF);

setup_psp(PSP_DISABLED);

setup_timer_0(RTCC_INTERNAL|RTCC_DIV_1);

setup_timer_1(T1_DISABLED);

setup_timer_2(T2_DISABLED,0,1);

}
```

```
// PROGRAMA i2c.h
```

```
#include <16F877.h>
```

```
#device adc=8
```

```
#FUSES NOWDT           //No Watch Dog Timer
```

```
#FUSES XT              //Crystal osc <= 4mhz
```

```
#FUSES NOPUT          //No Power Up Timer
```

```
#FUSES NOPROTECT      //Code not protected from reading
```

```
#FUSES BROWNOUT       //Reset when brownout detected
```

```
#FUSES LVP             //Low Voltage Programming on B3(PIC16) or  
B5(PIC18)
```

```
#FUSES NOCPD          //No EE protection
```

```
#FUSES NOWRT          //Program memory not write protected
```

```
#FUSES NODEBUG        //No Debug mode for ICD
```

```
#use delay(clock=4000000)
```

```
#use rs232(baud=9600,parity=N,xmit=PIN_C6,rcv=PIN_C7,bits=9)
```

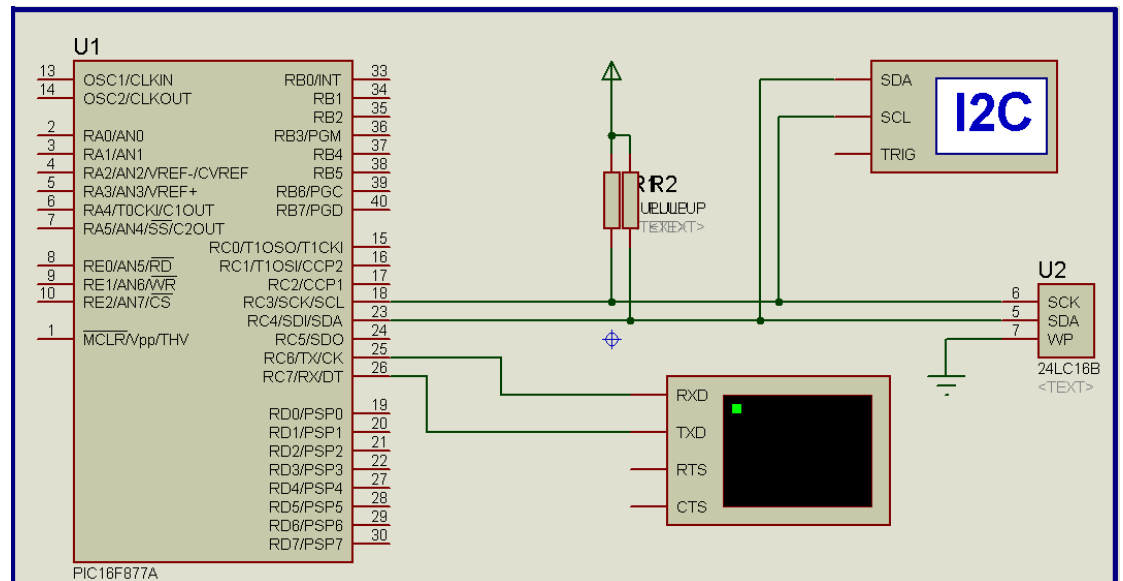
```
#use i2c(Slave,Slow,sda=PIN_C4,scl=PIN_C3,address=0x1)
```

g) GRABACION DE PIC

h) PRUEBA DE DISEÑO EN HARDWARE

PRACTICA ADICIONAL

Comunicación I2C con una memoria EEPROM



OBJETIVO:

Aprender a programar una interfaz de comunicación I2C entre la memoria EEPROM 24LC16 y un microcontrolador PIC utilizando lenguaje C de CCS

Comprender el código de las librerías correspondientes que facilitan el uso de los componentes utilizados

Utilizar tanto el PIC16F877A de la gama media como el PIC18F452 de la gama alta para entender el uso de las librerías que permiten su utilización con las memorias EEPROM 24LC16

CONTENIDO:

- a) ESPECIFICACIONES
- b) INSTRUCCIONES A SEGUIR
- c) PROGRAMA FUENTE DEL PIC16F877A: EX_EEPROM.C
- d) DESARROLLO DEL MISMO PROGRAMA UTILIZANDO EL PIC 18F452
- e) GRABACION DE PIC
- f) PRUEBA DE DISEÑO EN HARDWARE

a) ESPECIFICACIONES:

Se usará un PIC16F877A y una EEPROM 24LC16 conectados por medio de dos cables más tierra. El PIC tendrá una pantalla virtual en la que aparecerán los siguientes mensajes que deberán seguirse para el correcto funcionamiento del diseño:

- 1) Un mensaje pregunta si se trata de “LECTURA / ESCRITURA”. Para indicar lectura ingrese ‘R’ y para indicar escritura ingrese ‘W’.

- 2) Luego aparece el mensaje “DIRECCION:” y se deberá ingresar una dirección de tres posiciones hexadecimales.
- 3) En el caso de que se hubiera ingresado “R” automáticamente se proporcionará el “VALOR” leído de esa dirección EEPROM.
- 4) Si en cambio se hubiera ingresado “W” automáticamente hubiese aparecido el mensaje: “NUEVO VALOR:” y se tendría que ingresar el valor que se quiere escribir en la dirección señalada.

b) PROCEDIMIENTO A SEGUIR:

El estudiante deberá compilar el programa proporcionado en EX_EEPROM.c para lo cual deberá primero leerlo con el programa PIC C de CCS desde el directorio PICC/EXAMPLES. A continuación deberá incorporar el ejecutable en el archivo de PROTEUS y corregir librería antes de hacerlo trabajar (ver parte c).

En el diagrama podrá observar un visualizador I2C que permite capturar lo que se transmite por esta interfaz.

Trate grabando diferentes localidades de la memoria EEPROM y leyéndolas de vuelta para verificar su correcto funcionamiento. Para visualizar la memoria EEPROM I2C luego de arrancar Proteus deberá colocarlo en pausa y seleccionarla en el menú “Debug”.

Analice el programa con cada uno de sus archivos incluidos porque es allí donde encontrará las funciones utilizadas en la programación del archivo principal. Esto permite comprender mejor los alcances y el poder de la programación en un lenguaje de nivel alto cuando se cuenta con las librerías apropiadas.

c) PROGRAMA FUENTE DEL PIC16F877A: EX_EEPROM.C

Este programa está en la ruta:

C:\Archivos de programa\PICC\Examples

Para que funcione correctamente con el hardware propuesto deberá estudiarlo y modificar el archivo incluido:

```
#include "2416.c"
```

Debe de revisar con atención este archivo y modificar la posición de los pines utilizados para la comunicación I2C que aparecen definidos en el puerto E ya que el diseño propuesto utiliza los pines RC3/SCK/SCL y RC4/SDI/SDA del puerto C.

Indique la modificación que se tendría que hacer en el hardware para no modificar el software:

(coloque aquí su respuesta e inclúyala en su reporte)

Investigue lo anterior estudiando las funciones de la comunicación I2C del PIC C de CCS y explique su respuesta a la pregunta anterior.

(coloque aquí su respuesta e inclúyala en su reporte

d) DESARROLLO DEL MISMO PROGRAMA UTILIZANDO EL PIC 18F452

En esta parte deberá repetir el ejercicio anterior cambiando el PIC16F877 por el PIC18F452. Deberá observar la hoja de especificaciones del nuevo PIC para ver la forma de conexión. Cómo habrá observado la librería de la memoria EEPROM 24LC16 contempla la posibilidad de usar micros de la familia 18 cuyo conjunto de instrucciones ya no es de 14 bits sino de 16 bits. En el PIC C hay que escoger “PCH16 bits” como compilador en lugar de “PCM14 bits”.

e) GRABACION DE PIC

f) PRUEBA DE DISEÑO EN HARDWARE

4.3 PRACTICAS PARA LA MAESTRÍA EN AUTOMATIZACION INDUSTRIAL

4.3.1 CONSTRUCCION DE TARJETA PROTOTIPO BASICA USBPOLI, CARGA DE BOOTLOADER Y EJERCICIO DE APLICACIÓN

PRACTICA 1

CONSTRUCCIÓN DE TARJETA PROTOTIPO BASICA USBPOLI CARGA DE BOOTLOADER Y EJERCICIO DE APLICACION

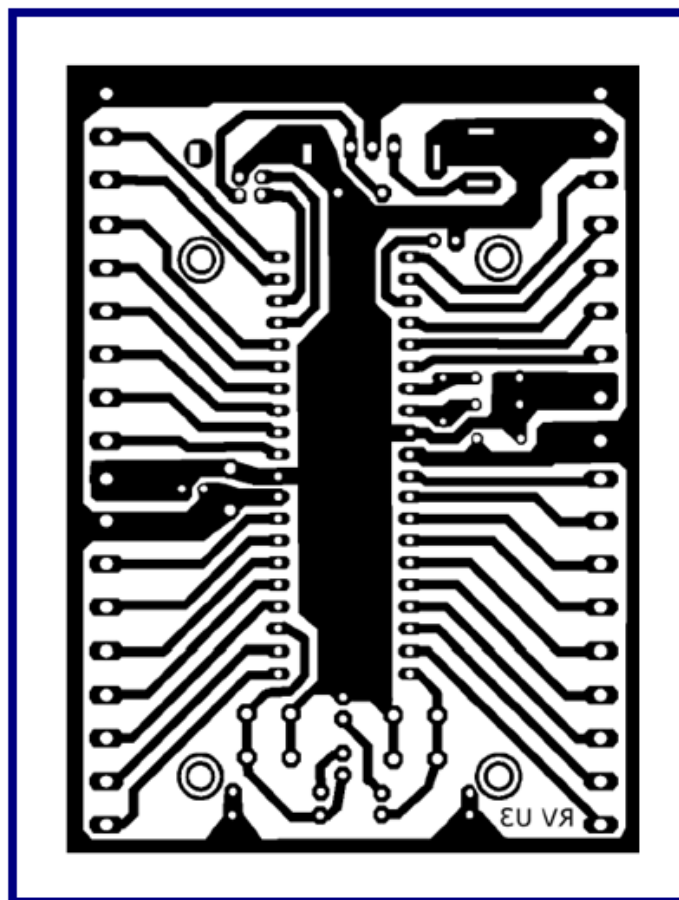


FIGURA No. TARJETA USBPOLI

OBJETIVO:

- Desarrollar destrezas en la creación de tarjetas prototipo educativas
- Enseñar la utilización de bootloaders para la programación de microcontroladores sin necesidad de un programador
- Introducir al estudiante en el manejo de la programación en Basic utilizado como herramienta de trabajo el compilador Mikrobasic

CONTENIDO:

- A) DIAGRAMA DE LA TARJETA A CONSTRUIR
- B) ESPECIFICACIONES DE LA TARJETA A CONSTRUIR
- C) TUTORIAL PARA LA CREACION DEL DISPOSITIVO DE INTERFACE HUMANA HID PARA EL MANEJO DEL PUERTO USB
- D) USO DEL SOFTWARE DEL BOOTLOADER

- Se utilizará un oscilador de cristal externo de 20MHz.
- Será capaz de escoger su alimentación de 5 voltios, mediante la selección de un jumper, del conector USB de la PC o mediante una fuente externa.
- Tendrá un pulsador para RESET y otro de propósito general
- Habrán dos LEDS una para indicar la existencia de la alimentación y otra de uso general en RB5 que es el mismo pin donde se encuentra conectado el pulsador de propósito general
- El PIC18F4550 se colocará sobre un zócalo de 40 pines
- Todos los pines del microcontrolador serán accesibles mediante conectores de tornillo colocados en los bordes de la tarjeta.

c) Tutorial para la creación del dispositivo de interface humana HID para el manejo del puerto USB con el programa PicBasicPro

Con el programa PicBasicPro abierto se selecciona **View-> EasyHID USB Wizard** y se abre la siguiente ventana:

EasyHID Wizard

Introduction

EasyHID is designed to provide a simple solution to the problems normally associated with implementing USB communications between a PIC microcontroller and Personal Computer (PC).

EasyHID is used to create two program templates, ready for compiling. One program is targetted for your PIC microcontroller (the USB device), the other is used on your PC (the USB host).

Device Information

Please enter your company and product name. The company and product name strings are mandatory fields. The device serial number is an optional string value and can be omitted.

Company Name:

Product Name:

Serial Number:

Buttons: Help, < Back, Next >, Cancel

En donde se ingresa ESPOL en Name y UsbPoli en Product

Name y se continúa con siguiente, apareciendo:

EasyHID Wizard

Vendor and Product ID

Please enter a valid Vendor ID (VID) and Product ID (PID).

Vendor ID:

Product ID:

If you intend to ship a USB device you need an official USB Vendor ID, which is unique throughout the world. Vendor ID's are assigned by the USB implementers forum at www.usb.org. Use the default values above FOR TESTING ONLY.

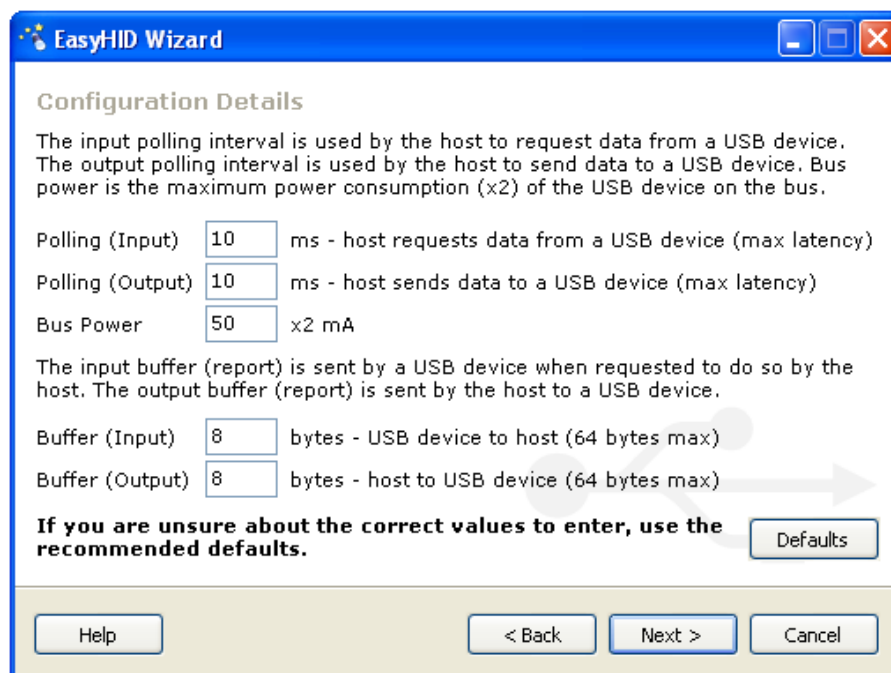
Alternatively, Mecanique own a USB Vendor ID and can provide an individual or a company with a set of product ID's at very low cost. This means that your product can be shipped world wide with a guaranteed unique and unambiguous VID and PID combination.

Purchase a unique set of Products IDs so that I can distribute my product...

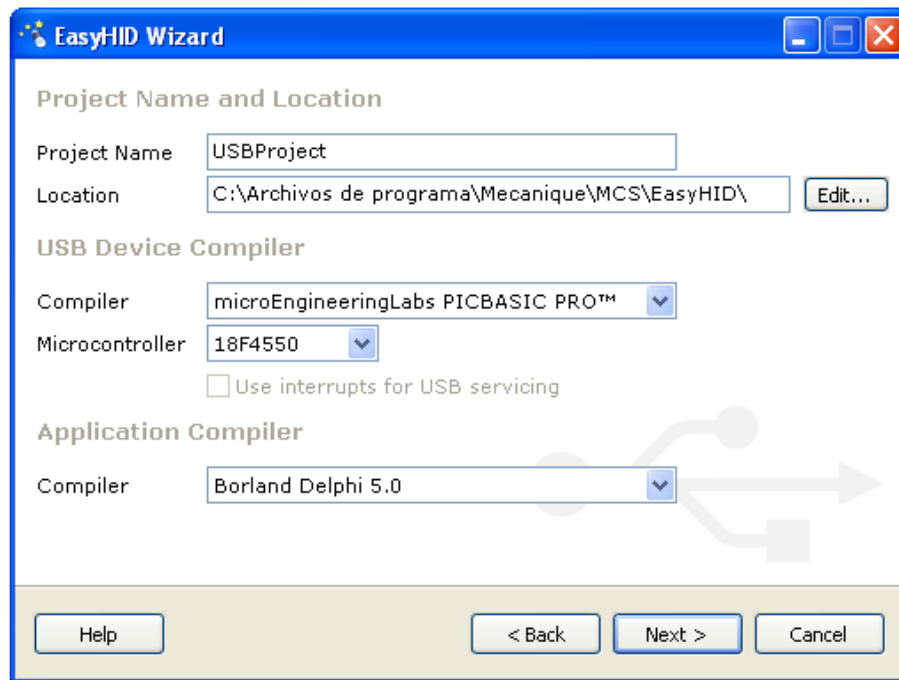
[Tell me more about using Vendor and Product ID's...](#)

Buttons: Help, < Back, Next >, Cancel

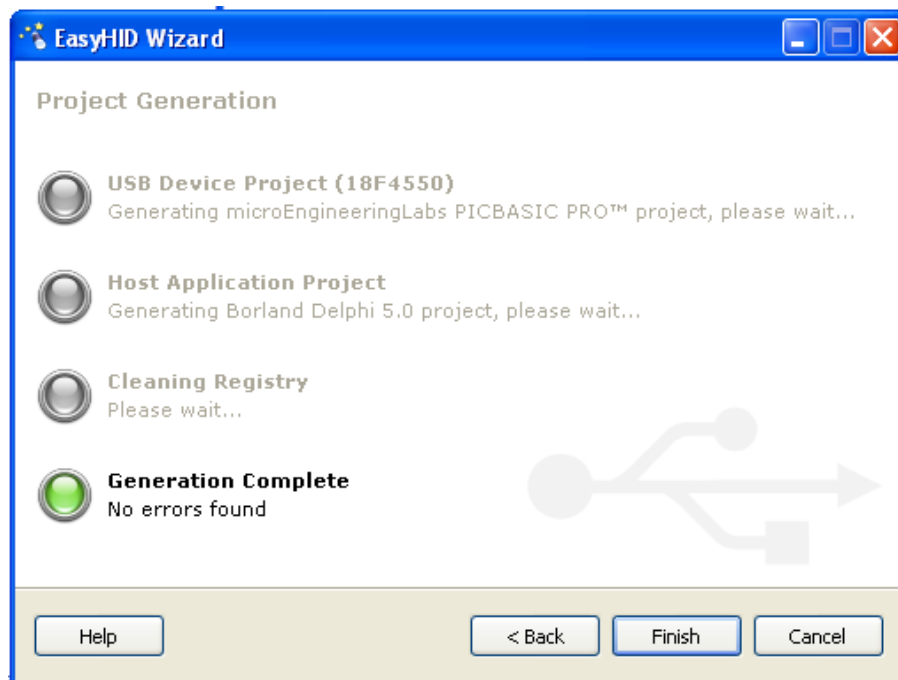
En esta ventana se ingresa el Vendor ID y el Product ID estos campos se dejan como están. Cada creador de un nuevo dispositivo deberá pagar derechos mínimos por el uso de dispositivos USB a la organización www.usb.org (lea el mensaje presentado en la pantalla anterior). Adelante con Next y aparece:



En la pantalla anterior se escogen los tiempos de comunicación y el número de bytes que se transmite y recibe. Para el caso actual se aceptan los valores presentados por default y se continúa con Next. Esto genera la siguiente ventana:

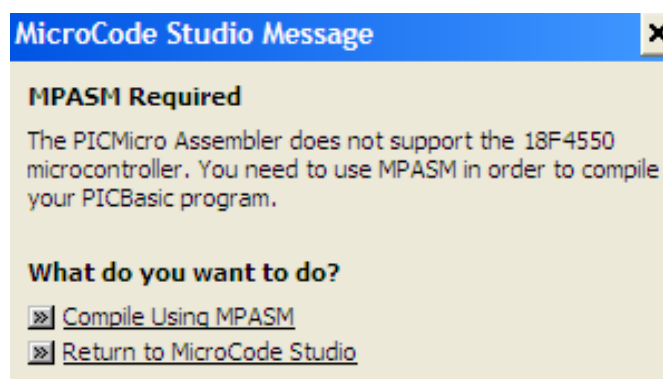


En la que lo único que se cambia es el compilador a Visual Basic 5.0. Al digitar Next aparece la siguiente ventana luego de pasos intermedios automáticos que se suceden uno tras otro, culminando el proceso al apretar Finish.

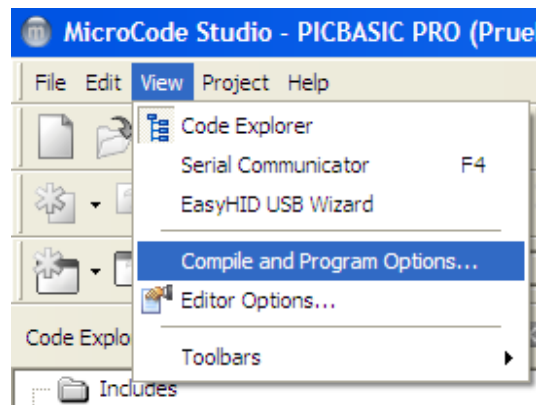


A partir de este punto el dispositivo USB está listo para su manipulación.

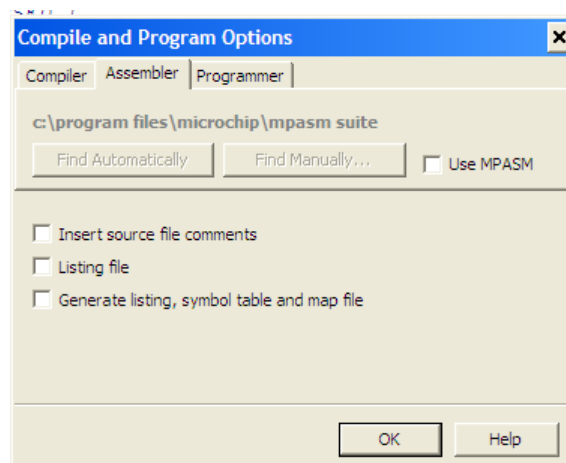
Ahora se procede a la activación del C18 en el PicBasicPro. Al compilar un código basado en los PIC18 se generan errores por la incorrecta asignación del ensamblador, entonces aparece la siguiente ventana:



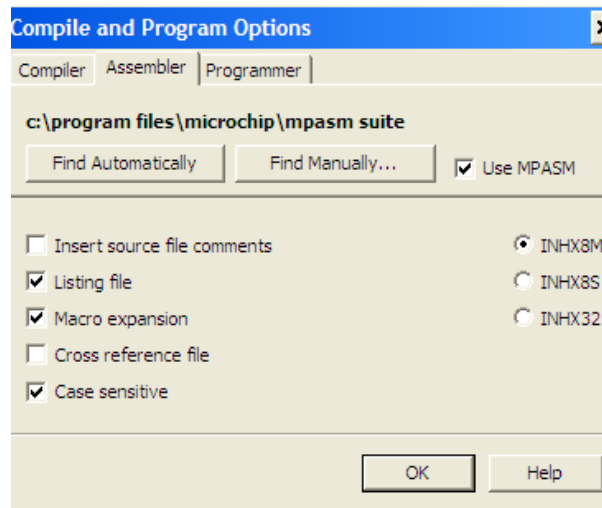
Al compilar debido a que el el PBP no soporta al PIC 18F4550 hay que escoger Compile using MPASM. Como la búsqueda tarda se cancela y se localiza el ensamblador en forma manual. Cancelada la búsqueda en la barra de menú se selecciona la opción **View -> Compile and Program Options**, tal como en la siguiente gráfica:



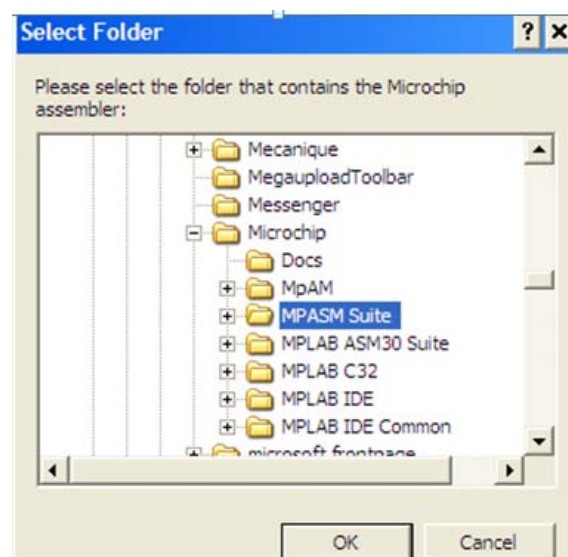
Luego de la selección aparece:



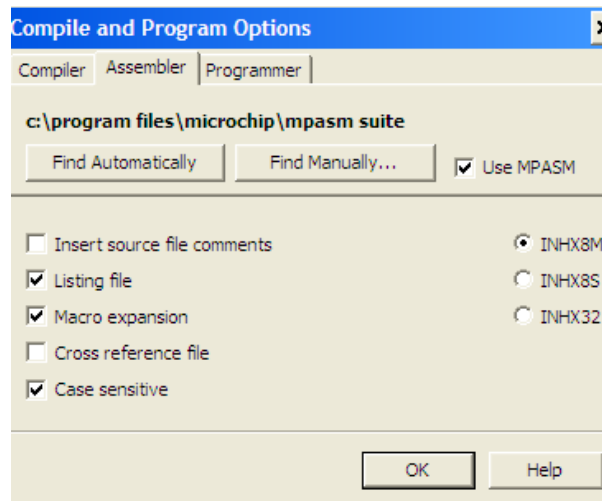
Donde se selecciona la opción de **Use MPASM**. Luego se presiona **Find Manually**.



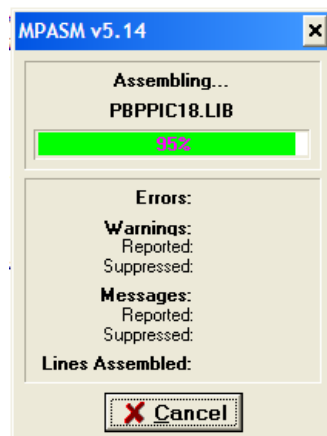
Apareciendo una ventana que nos permite ubicar la ruta adecuada tal como se muestra en la siguiente figura:



Seleccionada la ruta debe escogerse el siguiente cuadro:



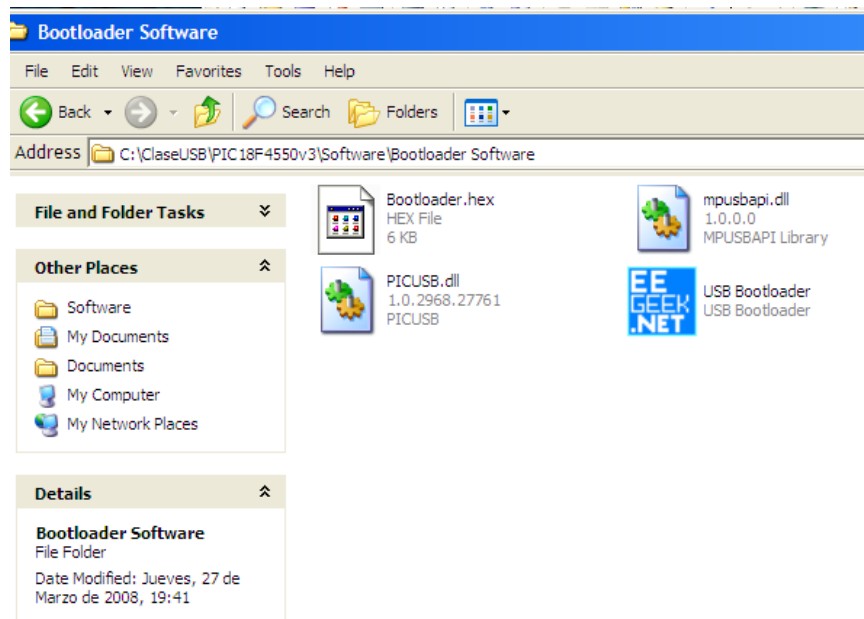
A partir de este momento empieza a compilarse mostrando:



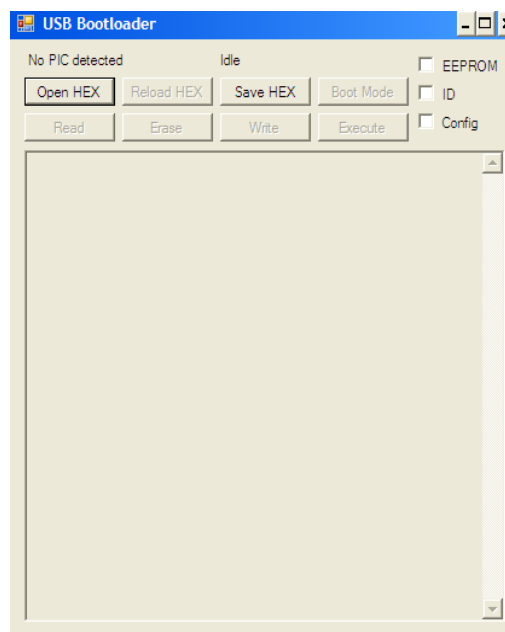
Al final se observará el mensaje de compilación completa.

d) **Uso del software Bootloader**

En la carpeta de Software Bootloader se selecciona el icono **USB Bootloader**.

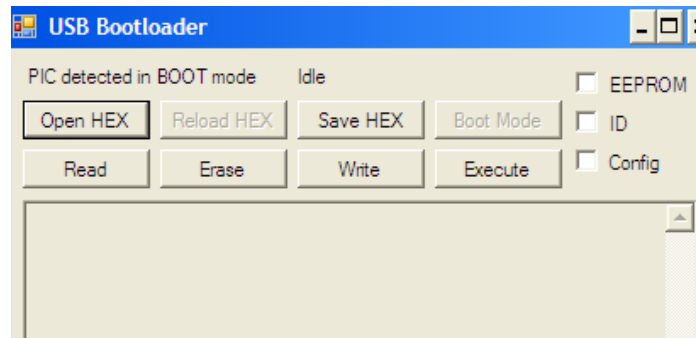


Con doble click se espera a que se abra el programa.

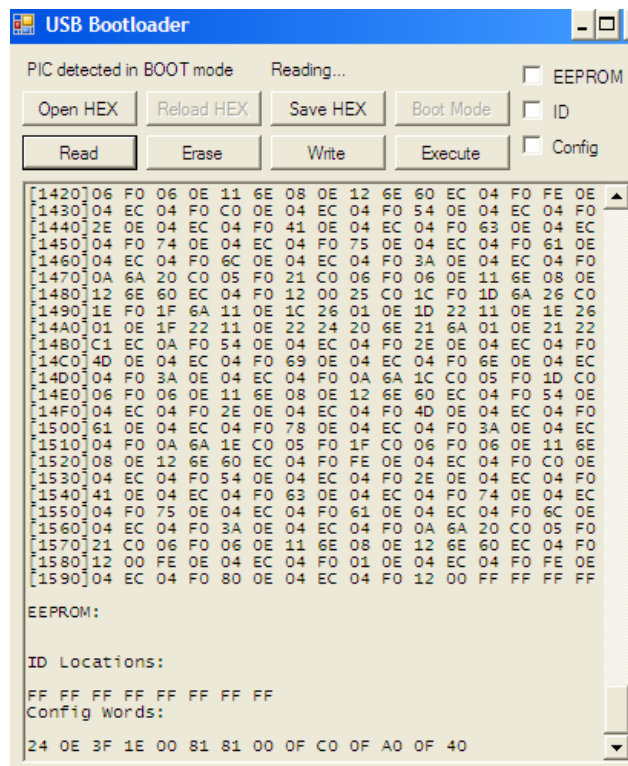


Para el siguiente paso la tarjeta USbPoli debe estar conectada al Puerto USB. Recuerde que hay dos pulsadores uno de RESET y otro de propósito general que sera por ahora el de

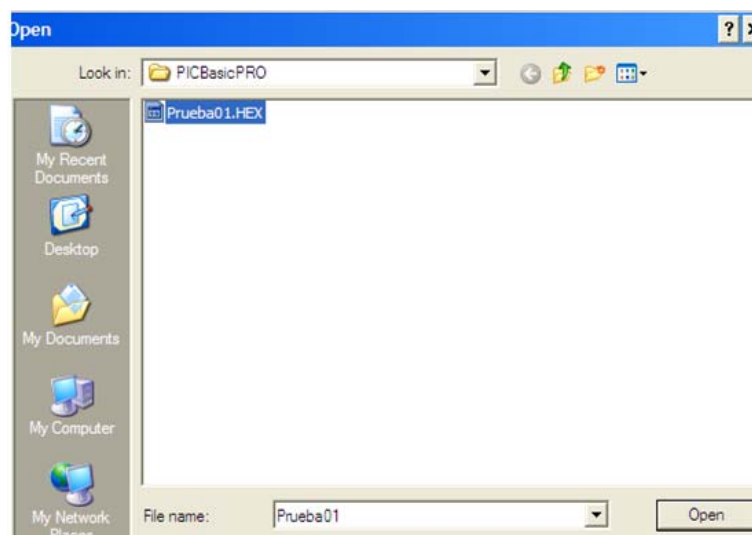
PROGRAMACIÓN. Para entrar al modo de programación se presiona primero el pulsador PROGRAMACION y luego el de RESET. El bootloader se mostrará de la siguiente manera:



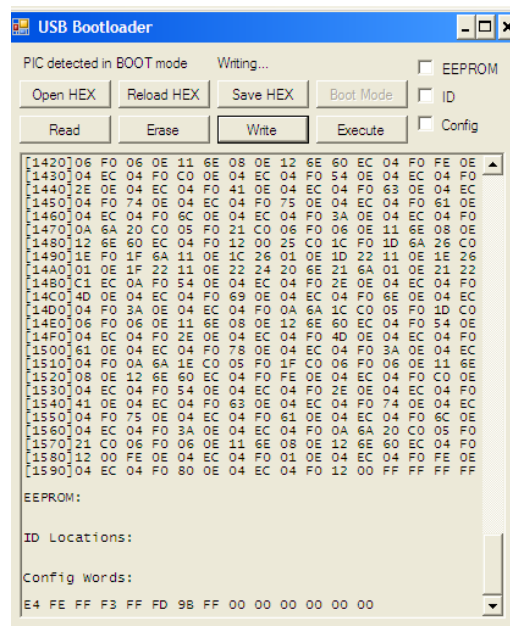
Que nos permite entrar de lleno al modo de programación. Con READ obtenemos el código hexadecimal grabado en el PIC.



Con Open Hex se puede invocar a un Nuevo código .hex que se desee grabar en el PIC. Aparecerá una ventana que le permite escoger la ruta de acceso de la siguiente manera.



Escogido el código a grabar, este se mostrará en la pantalla en forma similar a lo que se muestra a continuación:



De aquí lo que resta es grabarlo al PIC mediante el botón Write. Terminada la grabación se presiona RESET en la tarjeta. Ya se la puede desconectar del Puerto USB. El PIC estará cargado con el Nuevo programa y estará listo para probarse.

e) ESPECIFICACIONES DEL EJERCICIO A DESARROLLAR PARA COMPROBAR EL FUNCIONAMIENTO BASICO DE LA TARJETA USBPOLI

Cada estudiante escogerá un ejercicio que involucre el uso de por lo menos un pulsador y tres LEDS. Recuerde que la tarjeta solo cuenta con un LED y un pulsador de propósito general, así que cualquier hardware adicional puede ser añadido colocándolo en

un proto board que se conectará mediante cables con la tarjeta UsbPoli o directamente en los conectores de la tarjeta.

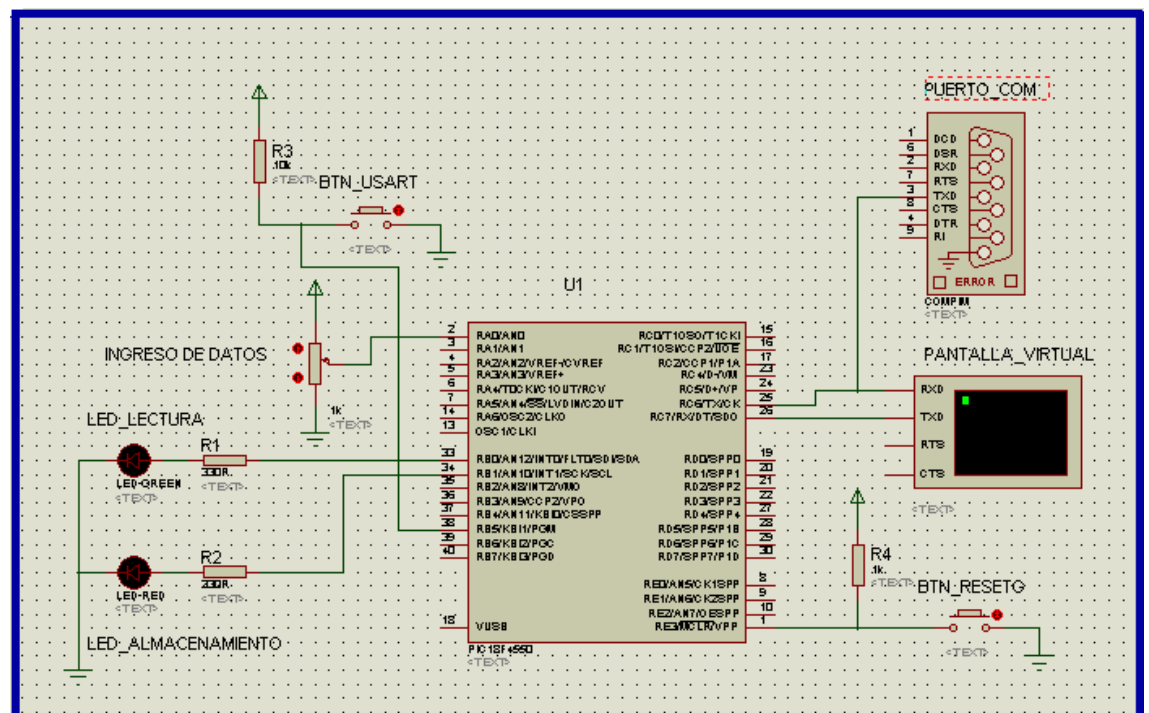
El trabajo será individual y se lo presentará en un CD siguiendo el siguiente formato:

- a) Especificaciones del proyecto
- b) Diagrama de bloques
- c) Diagrama de flujo del programa principal
- d) Diagrama de flujo de las subrutinas
- e) Código en PicbasicPro
- f) Simulación en Proteus
- g) Conclusiones (por lo menos tres)
- h) Recomendaciones (por lo menos tres)

4.3.2 ALMACENAMIENTO DE DATOS ADQUIRIDOS EN LA MEMORIA EEPROM PARA SU POSTERIOR VISUALIZACIÓN Y ANALISIS MEDIANTE LABVIEW O VISUAL BASIC

PRACTICA 2

ALMACENAMIENTO DE DATOS ADQUIRIDOS EN LA MEMORIA EEPROM PARA SU POSTERIOR VISUALIZACIÓN Y ANALISIS EN LABVIEW O VISUAL BASIC



OBJETIVO:

- Desarrollar destrezas en la adquisición de datos analógicos
- Aprender el manejo de la memoria EEPROM para almacenar datos importantes
- Introducir al estudiante al manejo del compilador Mikrobasic
- Involucrar al estudiante en el uso de interfaces con Visual Basic o LabView

CONTENIDO:

- a) ESPECIFICACIONES DEL PROGRAMA
- b) ESPECIFICACIONES PARA LA PRESENTACION DEL EJERCICIO A DESARROLLAR

a) Especificaciones del programa

Construir el hardware propuesto utilizando la tarjeta UsbPoli y cumplir con las siguientes especificaciones:

- Realizar la lectura de 10 datos mediante el uso de los canales analógicos del PIC18F4550, simulando el ingreso del dato con un potenciómetro de 10K; el cual va a variar entre 0 y 5 Voltios. Cada lectura será tomada en un tiempo estimado de 500 ms.

- Almacenar los datos leídos en la EEPROM INTERNA del PIC18F4550.

- Encender por cada dato de lectura un led de color verde, y se encenderá un led de color rojo al final del almacenamiento de los datos.

- Mediante una botonera se enviarán los datos almacenados en la EEPROM INTERNA del PIC18F4550 por el USART y se comunicará con un programa creado en VISUAL BASIC el mismo que receptorá los datos y los almacenará en un archivo de datos con extensión (.xls) para su posterior graficación.

b) Especificaciones para la presentación del ejercicio a desarrollar

El trabajo será individual y se lo presentará en un CD siguiendo el siguiente formato:

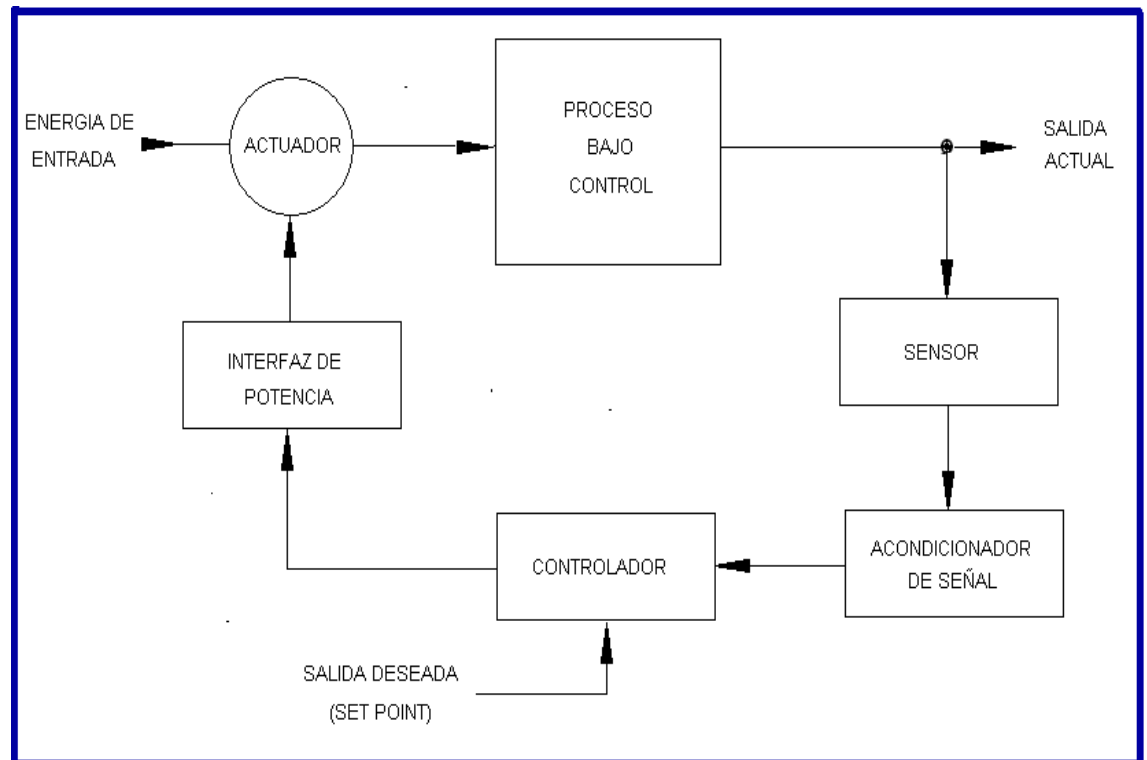
- a) Especificaciones del proyecto
- b) Diagrama de bloques
- c) Diagrama de flujo del programa principal
- d) Diagrama de flujo de las subrutinas

- e) Código en PicbasicPro
- f) Simulación en Proteus
- g) Conclusiones (por lo menos tres)
- h) Recomendaciones (por lo menos tres)

4.3.3 CONTROL EN LAZO CERRADO

PRACTICA 3

CONTROL EN LAZO CERRADO



OBJETIVO:

- Comprender la programación de microcontroladores en tareas de control
- Mejorar destrezas en el manejo de la programación Basic para microcontroladores

CONTENIDO:**c) Especificaciones del programa**

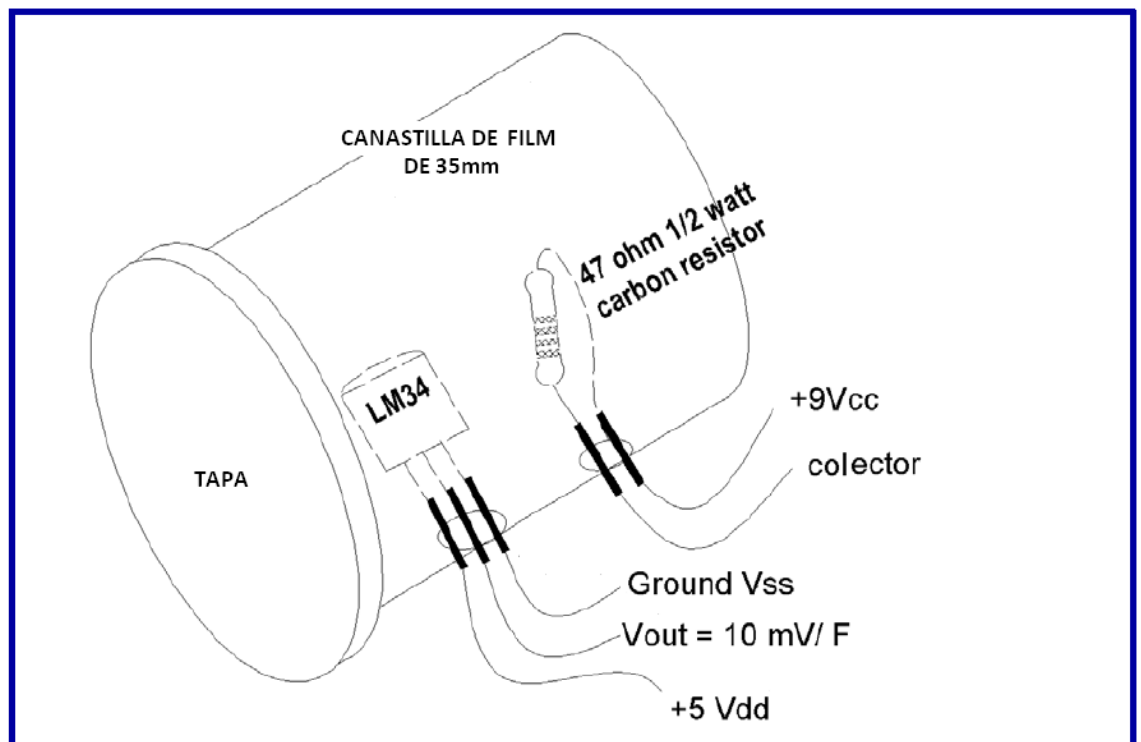
Con un control en lazo abierto se puede obtener una salida deseada si se tiene una buena comprensión del proceso y si las condiciones que lo afectan se mantienen constantes. Pero un proceso en lazo abierto no garantiza la salida deseada de un proceso sometido a perturbaciones aunque estas sean pequeñas.

En un sistema de control a lazo abierto no hay un mecanismo que permita reaccionar ante perturbaciones.

En esta práctica se construirá y programará un controlador On-off que controlará la temperatura de un proceso que tendrá un rango de valores entre un valor de temperatura máximo y un mínimo.

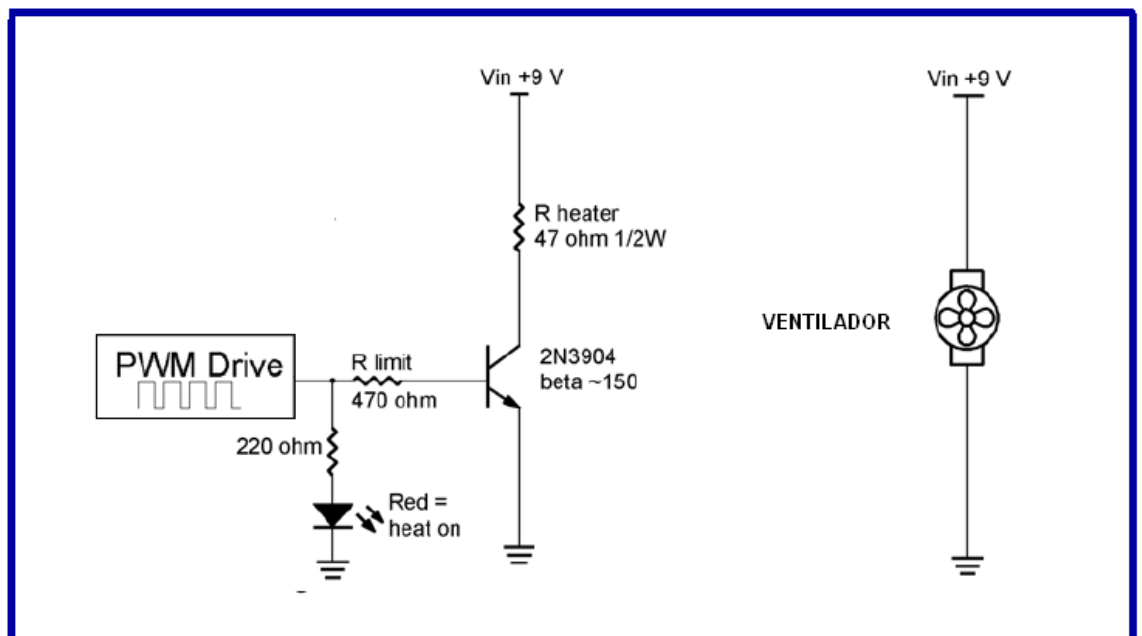
El controlador se lo programará en un PIC18F450 utilizando los siguientes elementos:

- Tarjeta UsbPoli
- Un teclado desde el que se podrá ingresar el Set Point y los límites superior e inferior del rango de temperatura
- Un display LCD 2x16 que permitirá mostrar mensajes y valores del proceso
- La planta del proceso se la construirá de acuerdo al gráfico siguiente:



Puede observarse en la figura anterior una resistencia y un sensor de temperatura LM34 colocados dentro de una canastilla cerrada con dos perforaciones, a través de las cuales se han insertado y asegurado la resistencia y el control de temperatura.

Para alimentar la resistencia se ha considerado un voltaje desregulado de 9 voltios a través de un transistor que recibirá del microcontrolador una señal PWM para energizar la resistencia. El mismo voltaje será utilizado para prender un ventilador que servirá para perturbar el sistema tal como se muestra en el siguiente gráfico:



El ejercicio planteado se lo hará inicialmente para un Set point de 101.5 grados fahrenheit que es la temperatura de una incubadora de polluelos. El valor de temperatura máximo será de 102.5 y el mínimo de 100.5 grados fahrenheit.

d) Especificaciones para la presentación del Reporte de la Práctica

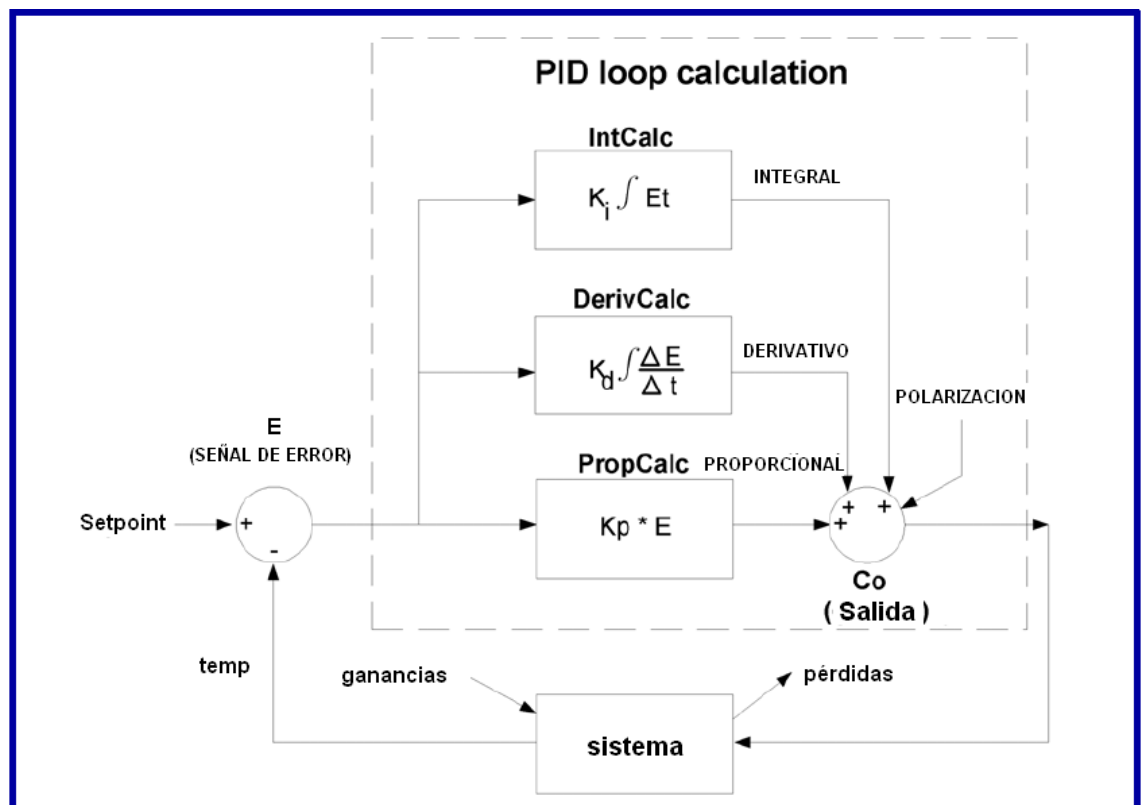
El trabajo se lo presentará en un CD siguiendo el siguiente formato:

- a) Especificaciones del proyecto
- b) Diagrama de bloques
- c) Diagrama de flujo del programa principal
- d) Diagrama de flujo de las subrutinas
- e) Código en PicbasicPro
- f) Simulación en Proteus
- g) Conclusiones (por lo menos tres)
- h) Recomendaciones (por lo menos tres)

4.3.4 CONTROL PID APLICADO AL CONTROL DE VELOCIDAD DE UN MOTOR DC

PRACTICA 4

CONROL PID APLICADO AL CONTROL DE VELOCIDAD DE UN MOTOR DC



OBJETIVO:

- Aprender la programación de controladores PID embebidos en microcontroladores
- Reforzar destrezas en el manejo de la programación Basic para microcontroladores

CONTENIDO:**a) Especificaciones del programa**

El objetivo de un control de procesos es mantener el proceso constante. En la práctica anterior creamos una banda diferencial para el ajuste de temperatura entre un valor máximo y un mínimo, causando que la temperatura oscilara alrededor del Setpoint. Si bien es cierto ese tipo de control tiene sus ventajas y desventajas, el control PID permite un mejor control de un sistema pero puede ser más difícil de implementar y sintonizar para un óptimo comportamiento.

Mantener un sistema constante implica que continuamente se le debe agregar energía para compensar exactamente las pérdidas

del sistema. Si las pérdidas del sistema fuesen constantes, el control del proceso se lo lograría añadiendo un drive constante. Sin embargo, los factores que afectan un proceso cambian en magnitudes y velocidades impredecibles, complicando el problema. Debe entenderse que el sistema tiene retardos de reacción y un instante de cambios en pérdidas debido a una perturbación, no es sentido inmediatamente, el cambio de drive del sistema no ocurre instantáneamente.

En esta práctica se construirá y programará un controlador PID de 8 bits que controlará la velocidad de un motor de acuerdo a un Setpoint que se ingresará mediante un teclado.

El controlador se lo programará en un PIC18F450 usando como referencia la Nota de Aplicación AN937 de Microchip que se puede obtener de la dirección www.microchip.com. En el diseño se emplearán los siguientes elementos:

- Tarjeta UsbPoli
- Un teclado desde el que se podrá ingresar el Set Point y los parámetros que se requieran para operar el sistema
- Un display LCD 2x16 que permitirá mostrar mensajes y valores del proceso

- La planta del proceso será un motor DC. Para su manejo se empleará un puente H
- La medición de la velocidad se la hará mediante un sistema óptico que permitirá contar las revoluciones por minuto del motor. En el eje del motor se acoplarán discos con franjas blancas y negras que darán un valor de conteo por vuelta, de acuerdo al número de franjas que se utilicen.

En esta ocasión los datos no se grabarán en la EEPROM del microcontrolador sino que se enviarán directamente a una PC y se mostrarán las gráficas del proceso para diferentes situaciones de sintonización PID indicando en cada caso los valores de ganancia proporcional k_p , ganancia integral k_i y ganancia derivativa k_d en gráficas similares a las de la nota de aplicación.

El ejercicio en 8 bits simplifica la tarea planteada en el documento AN937. Además no es necesario el manejo de los registros para signos porque con los tipos de variables disponibles en Mikrobasic se los puede manejar.

La visualización de las curvas de comportamiento de la variable controlada velocidad, se la podrá hacer empleando Visual Basic o LabVIEW. Los datos serán enviados directamente a través del RS232.

b) Especificaciones para la presentación del Reporte de la Práctica

El trabajo se lo presentará en un CD siguiendo el siguiente formato:

- a) Especificaciones del proyecto
- b) Diagrama de bloques
- c) Diagrama de flujo del programa principal
- d) Diagrama de flujo de las subrutinas
- e) Código en PicbasicPro
- f) Simulación en Proteus
- g) Conclusiones (por lo menos tres)
- h) Recomendaciones (por lo menos tres)

CONCLUSIONES Y RECOMENDACIONES

CONCLUSIONES

1. Se definen las habilidades profesionales a formar, a su paso por el Laboratorio de la asignatura Microcontroladores, en quienes cursan el pregrado o la Maestría en Automatización Industrial, a partir del objetivo rector de la asignatura.
2. Se declara un sistema de acciones que define el modo de actuación del estudiante en el laboratorio, tomando como punto de partida las habilidades profesionales a formarse en esta actividad docente.
3. Se presentan los recursos con los que debe contar el Sistema Modular propuesto para que pueda soportar el conjunto de acciones a desarrollar por los estudiantes en el laboratorio.
4. Con el Sistema Modular propuesto se implementa una plataforma de trabajo que es abierta, dinámica y que se encuentra en constante crecimiento, permitiendo agregar nuevos elementos a los módulos ya existentes.
6. El Sistema Modular propuesto puede ser empleado en los laboratorios de asignaturas afines que aparecen en el currículo de las carreras de perfil eléctrico con muy pocas modificaciones.

RECOMENDACIONES

1. Construir nuevas maquetas de procesos que permitan extender la plataforma de trabajo del Sistema Modular enriqueciendo el conocimiento adquirido e incorporando nuevos recursos para la enseñanza
2. Crear concursos para que los estudiantes puedan competir con sus diseños y creatividad en beneficio de la colectividad
3. Incrementar el equipamiento del Laboratorio para dotar al estudiante de mejores recursos para su formación
4. Extender el uso del Sistema Modular al nivel de enseñanza de Postgrado.

ANEXOS

ANEXO I

CARACTERES ASCII

Los caracteres ASCII son de gran utilidad en la representación de información. A menudo se necesita trabajar con interfaces y es necesario trabajar con diferentes tipos de datos que emplean su conversión en ASCII que es parte de una representación estándar que se utiliza con mucha frecuencia. Los LCD trabajan con caracteres ASCII si se quiere publicar algún valor en sus pantallas hay que pasar necesariamente por la representación ASCII.

El código ASCII reserva los primeros 32 primeros códigos como caracteres de control para dispositivos como impresoras que utilizaban ASCII. Por ejemplo el carácter 13 representa "retorno de carro" o RETURN. El carácter 10 representa "nueva línea" o LINE FEED. El código 127 (los siete bits a uno, equivale a "suprimir" DELETE. Muchos de los caracteres de control ASCII ya no se usan porque fueron ideados para el control de cintas perforadas. Actualmente los usuarios de ASCII usan menos los caracteres de control, con algunas excepciones como "retorno de carro" o "nueva línea".

Caracteres de control ASCII

Binario	Decimal	Hex	Abreviatura	Repr	AT	Nombre/Significado
0000 0000	0	00	NUL	NUL	^@	Carácter Nulo
0000 0001	1	01	SOH	SOH	^A	Inicio de Encabezado
0000 0010	2	02	STX	STX	^B	Inicio de Texto
0000 0011	3	03	ETX	ETX	^C	Fin de Texto
0000 0100	4	04	EOT	EOT	^D	Fin de Transmisión
0000 0101	5	05	ENQ	ENQ	^E	Enquiry
0000 0110	6	06	ACK	ACK	^F	Acknowledgement
0000 0111	7	07	BEL	BEL	^G	Timbre
0000 1000	8	08	BS	BS	^H	Retroceso

0000 1001	9	09	HT	HT	^I	Tabulación horizontal
0000 1010	10	0A	LF	LF	^J	Line feed
0000 1011	11	0B	VT	VT	^K	Tabulación Vertical
0000 1100	12	0C	FF	FF	^L	Form feed
0000 1101	13	0D	CR	CR	^M	Carriage return
0000 1110	14	0E	SO	SO	^N	Shift Out
0000 1111	15	0F	SI	SI	^O	Shift In
0001 0000	16	10	DLE	DLE	^P	Data Link Escape
0001 0001	17	11	DC1	DC1	^Q	Device Control 1 — oft. XON
0001 0010	18	12	DC2	DC2	^R	Device Control 2
0001 0011	19	13	DC3	DC3	^S	Device Control 3 — oft. XOFF
0001 0100	20	14	DC4	DC4	^T	Device Control 4
0001 0101	21	15	NAK	NAK	^U	Negative Acknowledgement
0001 0110	22	16	SYN	SYN	^V	Synchronous Idle
0001 0111	23	17	ETB	ETB	^W	End of Trans. Block
0001 1000	24	18	CAN	CAN	^X	Cancel
0001 1001	25	19	EM	EM	^Y	End of Medium
0001 1010	26	1A	SUB	SUB	^Z	Substitute
0001 1011	27	1B	ESC	ESC	^[or ESC	Escape
0001 1100	28	1C	FS	FS	^\ ^_	File Separator
0001 1101	29	1D	GS	GS	^]	Group Separator
0001 1110	30	1E	RS	RS	^^	Record Separator
0001 1111	31	1F	US	US	^_	Unit Separator
0111 1111	127	7F	DEL	DEL	^?, Delete , or Backspace	Delete

Caracteres imprimibles ASCII

Binar io	D ec	H ex	Represen tación	Binar io	D ec	H ex	Represen tación	Binar io	D ec	H ex	Represen tación
0010 0000	32	20	espacio ()	0100 0000	64	40	@	0110 0000	96	60	`
0010 0001	33	21	!	0100 0001	65	41	A	0110 0001	97	61	a
0010 0010	34	22	"	0100 0010	66	42	B	0110 0010	98	62	b
0010 0011	35	23	#	0100 0011	67	43	C	0110 0011	99	63	c
0010 0100	36	24	\$	0100 0100	68	44	D	0110 0100	100	64	d
0010 0101	37	25	%	0100 0101	69	45	E	0110 0101	101	65	e
0010 0110	38	26	&	0100 0110	70	46	F	0110 0110	102	66	f
0010 0111	39	27	:	0100 0111	71	47	G	0110 0111	103	67	g
0010 1000	40	28	(0100 1000	72	48	H	0110 1000	104	68	h
0010 1001	41	29)	0100 1001	73	49	I	0110 1001	105	69	i
0010 1010	42	2A	*	0100 1010	74	4A	J	0110 1010	106	6A	j
0010 1011	43	2B	±	0100 1011	75	4B	K	0110 1011	107	6B	k
0010 1100	44	2C	^	0100 1100	76	4C	L	0110 1100	108	6C	l
0010 1101	45	2D	=	0100 1101	77	4D	M	0110 1101	109	6D	m
0010 1110	46	2E	^	0100 1110	78	4E	N	0110 1110	110	6E	n

0010 1111	47	2F	l	0100 1111	79	4F	O	0110 1111	11 1	6F	o
0011 0000	48	30	0	0101 0000	80	50	P	0111 0000	11 2	70	p
0011 0001	49	31	1	0101 0001	81	51	Q	0111 0001	11 3	71	q
0011 0010	50	32	2	0101 0010	82	52	R	0111 0010	11 4	72	r
0011 0011	51	33	3	0101 0011	83	53	S	0111 0011	11 5	73	s
0011 0100	52	34	4	0101 0100	84	54	T	0111 0100	11 6	74	t
0011 0101	53	35	5	0101 0101	85	55	U	0111 0101	11 7	75	u
0011 0110	54	36	6	0101 0110	86	56	V	0111 0110	11 8	76	v
0011 0111	55	37	7	0101 0111	87	57	W	0111 0111	11 9	77	w
0011 1000	56	38	8	0101 1000	88	58	X	0111 1000	12 0	78	x
0011 1001	57	39	9	0101 1001	89	59	Y	0111 1001	12 1	79	y
0011 1010	58	3 A	:	0101 1010	90	5 A	Z	0111 1010	12 2	7 A	z
0011 1011	59	3 B	:	0101 1011	91	5 B	[0111 1011	12 3	7 B	l
0011 1100	60	3 C	≤	0101 1100	92	5 C	\	0111 1100	12 4	7 C	l
0011 1101	61	3 D	≡	0101 1101	93	5 D	l	0111 1101	12 5	7 D	l
0011 1110	62	3 E	≥	0101 1110	94	5 E	^	0111 1110	12 6	7 E	z
0011 1111	63	3F	?	0101 1111	95	5F	-				

BIBLIOGRAFIA

- 1) Diseño Práctico de Aplicaciones. Primera Parte PIC16F84. José María Angulo Useátegui, Ignacio Angulo Martínez. Mc Graw Hill, 2003.
- 2) Diseño Práctico de Aplicaciones. Segunda Parte PIC16F87x. José María Angulo Useátegui, Susana Romero Yesa. Ignacio Angulo Martínez. Mc Graw Hill, 2000
- 3) Microcontroladores. CD conteniendo eBook. Dra. Nancy Blanco, 2002.
- 4) Embedded Design with the PIC18F452 Microcontroller. John B. Peatman. Pearson Education, Prentice Hall, 2003.
- 5) Aprenda Rápidamente a Programar Microcontroladores. Carlos A. Reyes. Gráficas Ayerve C. A., 2004.
- 6) Curso Avanzado de Microcontroladores PIC. Edison Duque C.. CEKIT. Editado por CEKIT, 1998.
- 7) PICmicro MCU C. An Introduction to Programming the Microchip PIC in CCS C. Nygel Gardner. Publicado por CCS Inc., 2002.
- 8) PIC MCU Compiler Reference Manual. Publicado por CCS Inc., 2003.
- 9) Industrial Control. Publicado por Paralallax Educational Materials.
- 10) Automating Science and Engineering Laboratories with Visual Basic. Mark F. Russo, Martin M. Echols. John Wiley and Sons Inc. 1999.
- 11) PIC in Practice a Project Based Approach. D. W. Smith. Elsevier, Impreso por Newnes de Elsevier, 2003.
- 12) Microcontroller Programming the Microchip PIC. Julio Sanchez. María P. Canton. CRC Press Taylor and Francis Group, 2007.
- 13) Interfacing PIC Microcontrollers Embedded Design by Interactive Simulation. Martin Bates. Elsevier, Impreso por Newnes de Elsevier, 2006.
- 14) The Quintessential PIC Microcontroller. Springer-Verlag, 2000.
- 15) Designing Embedded Systems with PIC Microcontrollers Principles and Applications. Tim Wilmshurst. Elsevier, Impreso por Newnes de Elsevier, 2007.
- 16) Programming 16 bit Microcontrollers in C. Lucio Di Jasio. Elsevier, Impreso por Newnes de Elsevier, 2007.

- 17) PIC16F87X Tutorial by Example Copyright, Peter H. Anderson, Baltimore, MD, Jan, '01
- 18) Embedded C Language Development Kit For the PICmicro® MCU EXERCISE BOOK. Custom Computer Services, Inc. Brookfield, Wisconsin, USA, 2002.
- 19) Serial Port Complete. Programming and Circuits for RS-232 y RS-485 Links and Networks. John Axelson. Publicado por Lakeview Reasearch, 2000.
- 20) Serial Port Complete. COM Ports, USB Virtual COM Ports and Ports for Embedded Systems. Jan Axelson. Publicado por Lakeview Reasearch, 2007.