



**UNIVERSIDAD CATÓLICA DE
SANTIAGO DE GUAYAQUIL**

**FACULTAD DE EDUCACIÓN
TÉCNICA PARA EL DESARROLLO**

**TESIS DE GRADO
PREVIO A LA OBTENCIÓN DEL TÍTULO DE:
INGENIERO EN TELECOMUNICACIONES CON
MENCIÓN EN GESTIÓN EMPRESARIAL**

**TEMA:
ANÁLISIS DESCRIPTIVO DEL SISTEMA MODULAR
EASYPIC6 E IMPLEMENTACIÓN DE PRÁCTICAS PARA
LA MATERIA DE MICROCONTROLADORES**

**ALUMNOS:
HUAYAMAVE JÚPITER IVÁN JOSÉ
MONSERRATE COELLO RICARDO JOSÉ
PONCE ORELLANA ANDRÉS FERNANDO
ROBALINO CRUZ LISSETTE GEOMAR
RUIZ CASTRO JEAN CARLOS**

**DIRECTOR:
MSC. PALACIOS MELÉNDEZ EDWIN, ING.**



TESIS DE GRADO

**“ANÁLISIS DESCRIPTIVO DEL SISTEMA MODULAR EASYPIC6 E
IMPLEMENTACIÓN DE PRÁCTICAS PARA LA MATERIA DE
MICROCONTROLADORES”**

Presentado a la Facultad Técnica, Carrera de Ingeniería en Telecomunicaciones de la
Universidad Católica de Santiago de Guayaquil, por:

**HUAYAMAVE JÚPITER IVÁN JOSÉ
MONSERRATE COELLO RICARDO JOSÉ
PONCE ORELLANA ANDRÉS FERNANDO
ROBALINO CRUZ LISSETTE GEOMAR
RUIZ CASTRO JEAN CARLOS**

Para dar cumplimiento con uno de los requisitos para optar con el título de:

**INGENIERO EN TELECOMUNICACIONES
Mención: Gestión Empresarial en Telecomunicaciones**

Tribunal de Sustentación

MsC. Edwin F. Palacios Meléndez, Ing.

**MsC. Héctor Cedeño Abad, Ing.
Decano de la Facultad Técnica**

**Ing. Pedro Tutivén López
Director de Carrera**

**Ing. Víctor del Valle Ramos
Coordinador Académico**

Profesor Revisor

CERTIFICACIÓN

Certifico que el proyecto de grado titulado **“Análisis Descriptivo del Sistema Modular EasyPic6 e implementación de prácticas para la materia de Microcontroladores”**, desarrollado por los estudiantes Huayamave Júpiter Iván José, Monserrate Coello Iván José, Ponce Orellana Andrés Fernando, Robalino Cruz Lissette Geomar y Ruiz Castillo Jean Carlos, fue realizado, corregido y terminado, razón por la cual está apto para su presentación y sustentación.

MsC. Palacios Meléndez Edwin, Ing.

DIRECTOR DE TESIS

AGRADECIMIENTO

Agradecer en primer lugar a Dios por habernos dado la vida y sus bendiciones que nos han permitido haber desarrollado este trabajo, a nuestras familias ya que sin su apoyo incondicional no seríamos lo que ahora somos.

Son muchas las personas a las que nos gustaría agradecer, profesores amigos y compañeros que a lo largo de nuestra vida universitaria nos ayudado con sus criticas para ser mejores profesionales, por su amistad, apoyo, ánimo y compañía en las diferentes etapas de nuestras vidas.

Agradecer al Decano de la Facultad Técnica, Ing. Héctor Cedeño Abad, por permitirnos que esta tesis se desarrollara en el marco de un proyecto de colaboración para nuestra Facultad. Debemos agradecer también a las diferentes autoridades que han pasado durante nuestra estancia como estudiantes en la facultad, ya que sin su apoyo como docentes y directivos pudimos alcanzar los objetivos planteados en nuestra vida universitaria.

Agradecer de manera especial al MsC. Palacios Meléndez Edwin, Ing. por aceptarnos para realizar este trabajo de tesis bajo su tutela. Su apoyo en nuestro trabajo y su capacidad para guiar nuestras ideas ha sido un gran aporte, no solamente en el desarrollo de esta tesis, sino también en las diferentes materias en las que ha incurrido. Le agradecemos también el habernos facilitado siempre los instrumentos necesarios para llevar a cabo todas las actividades propuestas durante el desarrollo de este proyecto. Muchas gracias.

DEDICATORIA

Dedicamos esta tesis a Dios y a nuestros padres. A Dios por habernos guiado en cada paso que damos, cuidándonos, dándonos fortaleza y sus bendiciones para poder continuar, a nuestros padres, quienes a lo largo de nuestras vidas se han preocupado por nuestro bienestar y educación siendo nuestro apoyo en cada momento. Brindándonos su entera confianza y apoyo incondicional en todos los retos que nos da la vida, aunque en el camino hayamos tropezado varias veces.

INDICE

CERTIFICACION.....	2
AGRADECIMIENTO.....	3
DEDICATORIA.....	4
INDICE.....	5
CAPITULO I: INTRODUCCION.....	9
1.1 JUSTIFICACION DEL PROBLEMA.....	12
1.2 DIFINICION DEL PROBLEMA.....	13
1.3 OBJETIVO.....	13
1.4 HIPOTISIS (O IDEA A DEFENDER).....	14
1.5 METODOLOGIA DE LA INVESTIGACION.....	14
CAPITULO II: SISTEMAS DE ENTRENAMIENTO DE MICROCONTROLADORES.....	16
2.1 INTRODUCCION.....	16
2.2 ENTRENADOR DE MICROCONTROLADORES PICAX.....	17
2.3 ANALISIS DE PRESENTACION DEL ENTRENADOR DE MICROCONTROLADOR PICAXE.....	24
2.4 ENTRENADOR DE MICROCONTROLADOR EASYPIC6.....	24
2.5 ANALISIS DE PRESENTACION DEL ENTRENADOR DE MICROCONTROLADORES V EASYPIC6.....	28
2.6 ENTRENADOR DE MICROCONTROLADORES PIC EDUTRONICA JJTSA.....	28
2.7 ANALISIS DE PRESENTACIONES DEL ENTRENADOR DE MICROCONTROLADORES PIC EDUTRONICA JJTSA.....	31
2.8 CONSIDERACION FINALES DE LOS SISTEMAS DE LOS SISTEMA DE ENTRENAMIENTOS DE MICROCONTROLADORES.....	31
CAPITULO III: SISTEMA MODULAR DE ENTRENAMIENTO EASYPI6.....	32
3.1 INTRODUCCION DE EASYPI6.....	32
3.2 CONEXIÓN DEL SISTEMA AL PC O NOTEBOOK.....	33
3.3 MICROCONTROLADORES SOPORTADOS.....	35

3.4 EL PROGRAMADOR INCORPORADO USB 2.0 PICFLASH	38
3.5 DEPURADOR DE CIRCUITOS A NIVEL DE HARWARE (MIKROLCD	41
3.6 FUENTE DE ALIMENTACION DEL ENTRENADOR EASYPI6.....	43
3.7 EL MODULO DE COMUNICACIÓN RS-232	44
3.8 MODULO DE COMUNICACIÓN PS/2	46
3.9 CONECTOR ICD.....	47
3.10 MODULO DE COMUNICACIÓN USB.....	48
3.11 SENSOR DE TEMPERATURA DS1820.....	50
3.12 ENTRADAS DE PRUEBA DEL CONVERTIDOR A/D 11.0	52
3.13 DIODOS LED	54
3.14 BOTON DE PRESION	56
3.15 TECLADOS	58
3.16 VISUALIZADOR LCD 2x16	60
3.17 VISUALIZADOR LCD INCORPORADO 2x16	62
3.18 VISUALIZADOR GRAFICO LCD 128x64.....	63
3.19 PANEL TACTIL	64
3.20 PUERTOS DE ENTRADA/SALIDA	66
3.21 ESTENSOR DE PUERTOS.....	70
CAPITULO IV: DESARROLLO EXPERIMENTAL DE LAS	
PRÁCTICAS DE LABORATORIO	72
4.1 PRÁCTICA #1: ESCRIBIR EN EL PUERTO B EL VALOR 55H	72
4.2 PRÁCTICA# 2: ESCRIBIR EN EL PUERTO B EL VALOR AAHH	74
4.3 PRÁCTICA# 3: ESCRIBIR EN EL PUERTO C EL VALOR FO	75
4.4 PRÁCTICA# 4: LEER EN EL PUERTO C Y ESCRIBIR EL PUERTO B.....	77
4.5 PRÁCTICA# 5: ESCRIBIR EN UN DISPLAY DE 7 SEGMENTOS SIN UTILIZAR DECODIFICADORES, UN VALOR ASENDENTES ENTRE O Y F	78

4.6 PRÁCTICA# 6: MEDIANTE UN DISPLAY ALFANUMERICO MOSTRAR EN ORDEN DECENDENTE LAS LETRAS DEL ALFABETO DE LA Z A LA A.....	81
4.7 PRÁCTICA# 7: LOGICA BOOLEANA CON MICROCONTROLADOR.....	88
4.8 PRÁCTICA# 8: UTILIZACIONES DE DECLARACIONES CON EL MICROCONTROLADOR PIC	89
4.9 PRÁCTICA# 9: CONTADOR ASCENDETE DE 0 A 9	94
4.10. PRACTICA#10: CONTADOR DESENDENTE DE 0 A9.....	94
4.11. PRÁCTICA#11: CONTADOR ASCENDENTE DE DOS DIGITOS	97
4.12. PRÁCTICA#12: TEMPORIZADOR-1	100
4.13. PRÁCTICA#13: TEMPORIZADOR-2	102
4.14. PRÁCTICA#14: TEMPORIZADOR-3	104
CAPITULO V: CONCLUSIONES Y RECOMENDACIONES	107
5.1 CONCLUSIONES.....	107
RECOMENDACIONES	108
BIBLIOGRAFIA	109
ANEXO	

RESUMEN

El desarrollo de la presente Tesis previo a la obtención del Título de Ingeniero en Telecomunicaciones, se encargó de realizar el estudio en mercado nacional e internaciones de los Sistemas Modulares de Entrenamiento de los Microcontroladores para así agilizar y facilitar el aprendizaje y enseñanza de los Microcontroladores en el Laboratorio de Electrónica.

La presente investigación consistió en la realidad existente en el mercado, analizando de manera muy eficiente y en beneficio de los estudiantes de la Carrera de Ingeniería en Telecomunicaciones en la Universidad Católica de Santiago de Guayaquil y hacerlo extensivo al estudiante ecuatoriano en general.

Así también a partir de la selección del Sistema Modular EasyPic6 se procedió a realizar 14 prácticas ilustrativas tanto en la programación utilizando el software **mikroBasic** y la simulación en **Isis Proteus**, para obtener un sistema basado en el programa de estudios de nuestra unidad académica y como institución educativa nos permita formar y fortalecer nuevas capacidades en cada uno de los alumnos que estudian la asignatura de Microcontroladores en beneficio de su futuro profesional.

CAPÍTULO 1: INTRODUCCIÓN

En el año de 1976, gracias al aumento en la densidad de integración, salió a la luz la primera computadora en un chip, es decir, se integraron junto con el microprocesador los subsistemas que anteriormente formaban unidades especializadas e independientes, pero unidas por las pistas de circuitos impresos con el microprocesador formando lo que se conoce como sistema mínimo. A este nuevo circuito integrado se le denominó **MICROCONTROLADOR**¹. No hay duda que disponer de un chip que cuente con tanto subsistemas, es un gran avance y consecuentemente muy rentable [Sánchez, 07].

Los Microcontroladores son uno de los componentes cruciales que más están experimentando un gran aumento de potencia debido al imparable desarrollo tecnológico. La consiguiente tendencia es a aumentar la complejidad del software que llevan incorporado. Este se concibió como un dispositivo programable que puede ejecutar un sinnúmero de tareas y procesos. Desde este momento, el diseño de productos electrónicos cambió radicalmente, circuitos lógicos, manejo de periféricos, temporizadores y estructura de computadores, todo programable y alojado en un solo integrado; es decir, un pequeño computador para todas las aplicaciones [Duque, 97].

La labor principal del microcontrolador es la de control sobre el sistema al que está adscrito. La utilización de este dispositivo frente a la lógica convencional tiene varias ventajas claras, entre las que podemos destacar la reducción de tamaño y costes del sistema al tener integrados varios periféricos y, por otro lado, la programabilidad, que permite utilizar un mismo

¹ **MICROCONTROLADOR:** es un circuito integrado diseñado especialmente para controlar sistemas electrónicos, que consta de todos los elementos de una computadora, como memoria de programa, memoria RAM, memoria EEPROM, puertos de entrada y salida, además de contadores, temporizadores, convertidores de analógico a digital, comparadores, etc.

componente, el microcontrolador, para multitud de aplicaciones sin más que cambiar el programa que lleva grabado.

Muchos de nosotros sabemos qué apariencia tiene una computadora. Usualmente tiene teclado, monitor, CPU², impresora y mouse. Este tipo de computadoras, como la PC³ o Mac⁴, son diseñadas principalmente para comunicarse con humanos. Manejo de base de datos, análisis financieros o incluso procesadores de textos, se encuentran todos dentro de la "gran caja", que contiene CPU, la memoria, el disco rígido, etc. El verdadero "cómputo", sin embargo, tiene lugar dentro de la CPU.

Si piensa sobre esto, el único propósito del monitor, teclado, mouse e incluso la impresora, es "conectar" a la CPU con el mundo exterior. ¿Pero usted sabía que hay computadoras alrededor de nosotros, corriendo programas y haciendo cálculos silenciosamente sin interactuar con ningún humano? Estas computadoras están en su auto, en el transbordador espacial, en un juguete, e incluso puede haber uno en su secador de pelo.

Los microcontroladores están presentes en muchos de los productos electrónicos que empleamos en nuestra vida cotidiana. Su enseñanza es un reto debido a la variedad de aplicaciones posibles. Sin embargo, a pesar de su diversidad, hay unidad en los principios de funcionamiento y en las arquitecturas de muchos microcontroladores [Valdés, 07]. Hay un número infinito de aplicaciones para los microcontroladores. ¡Su imaginación es el único factor limitante! Cientos (sino miles) de variaciones diferentes de microcontroladores están disponibles.

² **CPU:** La **unidad central de procesamiento** o **CPU** (por el acrónimo en inglés de *central processing unit*), o simplemente el **procesador** o **microprocesador**, es el componente del computador y otros dispositivos programables, que interpreta las instrucciones contenidas en los programas y procesa los datos.

³ **PC:** Una computadora personal u ordenador personal, también conocida como *PC* (sigla en inglés de *personal computer*)

⁴ **MAC:** *Mac* puede referirse a: Macintosh, una serie de computadoras de Apple Inc.

Algunos son programados una vez y producidos para aplicaciones específicas, otros son "reprogramables", que quiere decir que pueden ser usados una y varias veces para diferentes aplicaciones. Los Microcontroladores son increíblemente versátiles, se utilizan para la realización de sistemas electrónicos empotrados en otros sistemas (eléctricos, mecánicos, etc.) como por ejemplo electrodomésticos (televisor, lavadora, microondas, etc.), sistemas informáticos (ratón, impresora, etc.), sistemas de telecomunicaciones (teléfono móvil, circuito de control de una red, etc.) sistemas e control de maquinaria (circuito de control del brazo de un robot, etc.) o sistemas de automoción (circuito de control de frenado, circuito de control de climatización, etc.). en todas estas aplicaciones los microcontroladores emulan a numerosos sistemas digitales que antes se realizaban con circuitos integrados MSI⁵ y LSI⁶ como por ejemplo circuitos contadores, comparadores, etc. [Mandado, 07]

Prácticamente todo diseño electrónico contiene en la actualidad uno o más microcontroladores, el enorme avance de la electrónica, proporcionan cada vez más potentes microcontroladores industriales, a un precio muy reducido, lo que permite la simplificación del diseño del hardware que tradicionalmente llevaba a cabo un Ingeniero en Electrónica o Telecomunicaciones, aumentando sin embargo la complejidad del software que éste deberá programar.

Otro buen ejemplo es el gran desarrollo que han experimentado los teléfonos móviles. Cada vez son más pequeños en tamaño, pero tienen un sistema de menú cada vez más completo que incluye funcionalidades

⁵ **MSI**- Significa *Medium Scale Integration* (integración en mediana escala), y comprende los chips que contienen de 13 a 100 compuertas. Ejemplos: codificadores, registros, contadores, multiplexores, decodificadores y demultiplexores. los Circuitos Integrados MSI se fabrican empleando tecnologías TTL, CMOS, y ECL.

⁶ **LSI**- Significa *Large-Scale Integration* (integración en alta escala) y comprende los chips que contienen de 100 a 1000 compuertas. ejemplos: memorias, unidades aritméticas y lógicas (alu's), microprocesadores de 8 y 16 bits. Los Circuitos Integrados LSI se fabrican principalmente empleando tecnologías i2l, NMOS y PMOSs.

adicionales, como juegos o acceso a Internet, que nos puede dar una idea de la complejidad del software incorporado. Hoy en día el desarrollo del software es una parte crucial en el diseño de sistemas que puede llegar a ocupar un 70% del tiempo total dedicado al proyecto.

1.1. JUSTIFICACIÓN DEL PROBLEMA

Desde la docencia se debe apostar por adaptar la enseñanza de Microcontroladores a estos cambios, aumentando la importancia de la fase de construcción del software y dotándola de una metodología eficaz. Se propone un método concreto, con fases de análisis, diseño, desarrollo, pruebas y documentación, que facilitará la resolución de problemas complejos. Los resultados serán una programación de calidad y que tendrá en cuenta otros aspectos como correcciones o modificaciones posteriores, trabajo en equipo y generación progresiva de la documentación.

El por qué de utilizar un sistema modular de microcontroladores, se debe pensar que lo ofrecido en el mercado tiene variados propósitos y la misión de los docentes de electrónica es la de impartir conocimientos de manera más sencilla y explicativa. Los cuales serán presentados con principios claros que concuerden y se dispongan unos sobre otros con claridad y que estos puedan ser comprendidos en la asignatura de microcontroladores con naturalidad y facilidad.

Es importante hacer hincapié que lo expresado anteriormente no es solamente teoría sino que en realidad son sucesos que son llevados a la práctica real y que contribuyen con la noble tarea de la formación profesional de los futuros Ingenieros en Telecomunicaciones de la Facultad Técnica para el Desarrollo en la Universidad Católica de Santiago de Guayaquil en cada uno de los diferentes ocupaciones laborales en los que lograrán desempeñarse en el transcurso de su vida profesional.

1.2. DEFINICIÓN DEL PROBLEMA

Nuestro contacto con la empresa nos permite apreciar las carencias formativas de los Ingenieros en Telecomunicaciones, y una de las más importantes que observamos es no emplear un método para la programación de Microcontroladores. Sin duda hay que rechazar la programación "artesanal" y la figura del programador "gurú". Desde la docencia universitaria debemos rechazar esta figura como alguien que hace lo que nadie más puede, sin un método. De esta manera evitaríamos proyectos sin documentación o cuya única explicación son los comentarios insertados en el programa (que si el tiempo apremiaba, ni estaban) o confusos y extensos diagramas de flujo de muchas páginas y que ha medida que iban creciendo en tamaño tendían a desaparecer o a estar incompletos.

1.3. OBJETIVOS

Objetivo Principal:

Evaluar la operatividad del Sistema Modular EasyPic6 y la de sus aplicaciones con el PIC16F877 para el fortalecimiento del proceso enseñanza aprendizaje en la asignatura de Microcontroladores.

Objetivos Específicos:

- a) Diagnosticar el estado actual del arte o marco teórico en los sistemas modulares para Microcontroladores.
- b) Diseñar las prácticas de la materia de Microcontroladores para mejorar el proceso enseñanza aprendizaje.
- c) Evaluar el sistema modular EasyPic6 a través de las prácticas desarrolladas usando los Microcontroladores Microchip.
- d) Implementar los módulos EasyPic6 en el Laboratorio de Electrónica.

1.4. HIPÓTESIS (O IDEA A DEFENDER)

La operatividad del Sistema Modular EasyPic6 con distintas aplicaciones de la familia de los Microcontroladores, permitirá mejorar el proceso de enseñanza – aprendizaje, considerado también como aplicación en las tecnologías de la información y las comunicaciones (TICS) que se están incorporando a la enseñanza de la ingeniería a nivel mundial.

1.5. METODOLOGÍA DE INVESTIGACIÓN.

Es un estudio de carácter **Explicativo**, pues se pretende familiarizar a los estudiantes de la carrera de Ingeniería en Telecomunicaciones utilizando el entrenador de microcontroladores EsayPic6 que originan el fenómeno en cuestión, también interesa explicar por qué ocurre el fenómeno (uso de interfaz de programación en ensamblador, BASIC o C++) en conjunto con los microcontroladores. El **paradigma** es el Empírico-Analítico, con un enfoque **Cuantitativo**. El método es el **Ex post facto**, puesto que se pretenderá evidenciar las posibles relaciones de causa efecto entre los sistemas modulares de entrenamiento y los PICS (de Microchip).

El diseño de la Investigación es **No experimental Transversal**, puesto que no se manipularán deliberadamente las variables de estudio, se procederá a la observación directa del fenómeno tal y como se da en su contexto natural, y luego se procederá a su análisis respectivo. El diseño es transeccional, porque obedece a una realidad en un momento determinado, lo que significa que los datos serán recogidos en un tiempo único, puesto que el propósito es describir las variables, analizar su incidencia e interrelación en momento definido y único.

En el **Capítulo 1** se presenta la concepción metodológica de la tesis de pregrado: se definen los objetivos generales y específicos, los antecedentes,

el planteamiento y justificación del problema y los métodos de investigación utilizados.

En el **Capítulo 2** se presenta el marco teórico de los Sistemas de Entrenamiento de Microcontroladores; para analizar y describir las ventajas del sistema a implementar en el Laboratorio de Electrónica de la Facultad Técnica para el Desarrollo en la Universidad Católica de Santiago de Guayaquil.

En el **Capítulo 3** se presenta el marco teórico del Sistema Modular de Entrenamiento EasyPic6 (hardware) seleccionado en el capítulo 2, y el software que permite realizar la programación y simulación de diferentes aplicaciones con los microcontroladores PIC16F628A y PIC18F77A de la familia Microchip.

En el **Capítulo 4** se muestra el desarrollo de los experimentos o prácticas de laboratorio de los microcontroladores tanto en la programación como en simulación.

En el **Capítulo 5** se exponen las conclusiones y recomendaciones de la tesis.

CAPÍTULO 2

SISTEMAS DE ENTRENAMIENTO DE MICROCONTROLADORES

2.1 Introducción

El gran éxito de los microcontroladores es que nos permite diseñar sistemas programables de control mínimos de bajo coste, flexibles y muy fiables. En el mercado existen herramientas como software y hardware que facilitan el trabajo de diseñar, compilar y poner a punto las distintas aplicaciones en el ámbito profesional e inclusive en el área de la medicina.

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 [Valdivieso, 09]. Estas herramientas son empleadas en los laboratorios, centros de enseñanza e investigación en electrónica específicamente en la asignatura de microcontroladores como plataformas que simplifican a los estudiantes el diseño de aplicaciones.

Permiten trabajar con diferentes dispositivos PIC, aportan una serie de periféricos relevantes que permiten probar y simular el funcionamiento de aplicaciones, así como también se puede grabar (la programación al PIC) el microcontrolador sin sacar del equipo (entrenador). A continuación se detallan los diferentes entrenadores de microcontroladores para luego analizar las características de cada una de las plataformas propuestas y finalmente realizar un estudio más profundo de la plataforma escogida para la implementación en el laboratorio de electrónica:

1. Entrenador de Microcontroladores PICAXE⁷
2. Entrenador de Microcontroladores EasyPic6
3. Tarjeta PVK40

2.2 Entrenador de Microcontroladores PICAXE

El diseño del circuito del entrenador y programador de Microcontroladores **Picaxe** se basa en el Manual de uso, y el software Programming Editor de la página www.picaxe.co.uk. Consta de un conjunto de circuitos, que son independientes entre sí. Cada uno de los módulos tiene listadas las conexiones a la fuente de alimentación y a los diferentes elementos que lo componen, tiene terminales de conexión, donde, utilizando cable (RJ-45⁸) telefónico, se pueden transmitir señales de control de un bloque a otro. Ahora dependerá del circuito que se desea probar, se realiza la configuración necesaria.

En la figura 2.1 se muestra el circuito del entrenador por bloques el cual consta de una fuente de alimentación sencilla, donde la tensión que viene del transformador llega rectificadas y filtradas mediante el IC⁹ 7805, el mismo que entrega +5V (Vcc), que es la alimentación necesaria usada por los dispositivos a programar. El LED Verde (ver figura 2.2) encendido nos indica que el programador está alimentado y las resistencias sirven de acoplamiento de las señales del puerto serial a la señal del Microcontrolador.

⁷ **PICAXE** es el nombre de un sistema microcontrolador de origen británico basado en una gama de circuitos integrados PIC. Inicialmente comercializado para su uso en la educación y para aficionados a la electrónica, también son utilizados en ámbitos técnicos y comerciales, incluyendo el desarrollo rápido de prototipos.

⁸ **RJ-45** es una interfaz física comúnmente usada para conectar redes de cableado estructurado, (categorías 4, 5, 5e, 6 y 6a). *RJ* es un acrónimo inglés de *Registered Jack* que a su vez es parte del Código Federal de Regulaciones de Estados Unidos. Posee ocho "pines" o conexiones eléctricas, que normalmente se usan como extremos de cables de par trenzado.

⁹ **IC** es un **circuito integrado (CI)**, también conocido como *chip* o *microchip*, es una pastilla pequeña de material semiconductor, de algunos milímetros cuadrados de área, sobre la que se fabrican circuitos electrónicos generalmente mediante fotolitografía y que está protegida dentro de un encapsulado de plástico o cerámica. El encapsulado posee conductores metálicos apropiados para hacer conexión entre la pastilla y un circuito impreso

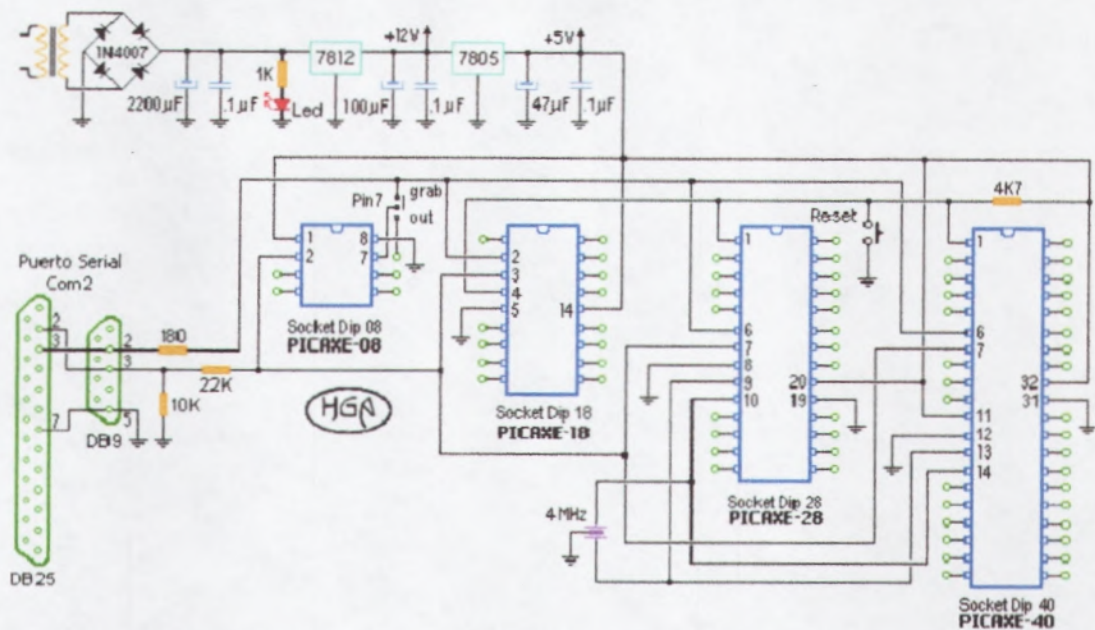


Figura 2.1: Diagrama por bloques del entrenador PICAXE. [Fuente: PICAXE]

En la figura 2.1 las señales del puerto serial (**DB25**) utilizadas son 2, 3, y 7, que permiten la conexión del circuito con la PC. El pin 2 (**Serial in**) es el encargado de transmitir los datos desde la PC hacia el Microcontrolador que se está programando. El pin 3 (**Serial out**) permite a la PC leer los datos enviados por el Microcontrolador desde el programador. En tanto el pin 7 (**Gnd**) es la conexión a tierra. El microcontrolador PICAXE posee internamente un pequeño programa (firmware) para que su utilización y programación sea mucho más fácil. Este programa interno, permite que los PICAXE puedan programarse en **BASIC** y en **DIAGRAMA DE FLUJO**, no requieren de cargador externo y se programan sin ser quitados del circuito donde están funcionando.

El microcontrolador PICAXE almacena todos sus programas en su memoria FLASH¹⁰ (no volátil). Es imposible que el código se pueda leer de

¹⁰ **Memoria FLASH** es una manera desarrollada de la memoria EEPROM que permite que múltiples posiciones de memoria sean escritas o borradas en una misma operación de programación mediante impulsos eléctricos, frente a las anteriores que sólo permite escribir o borrar una única celda cada vez.

un programa grabado en el microcontrolador PICAXE, por lo tanto debe guardar el código en el ordenador para después ser utilizado en el programador. Reprogramar el PICAXE, es muy sencillo, para lo cual grabe un nuevo programa en el microcontrolador, esta acción permitirá borrar el viejo código (programa) almacenado en la memoria y almacena el nuevo código en la memoria. La memoria sólo permite el almacenamiento de un programa a la vez.

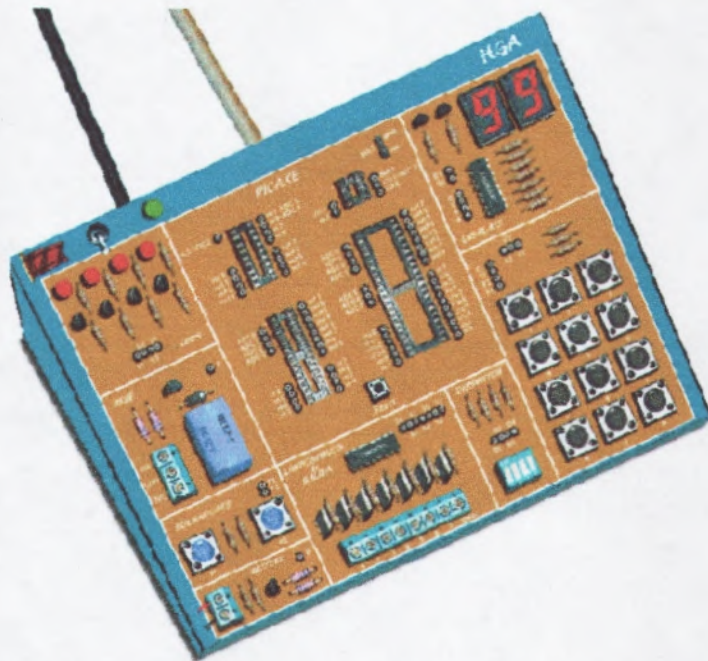


Figura 2.2: Entrenador de Microcontroladores PICAXE. [Fuente: PICAXE]

Durante la programación de microcontroladores es necesario que se conecte el cable del entrenador al programador (ordenador). En este bloque hay un zócalo independiente para diferentes tamaños (número de pines) de microcontroladores, un pulsador (**P1**), para reiniciar (resetear), adicional cuenta con conectores que sirven para extender los pines del microcontrolador, de esta forma, las señales de entrada y salida se pueden llevar hasta los otros bloques del circuito. El Jumper (**Pin 7**), es usado únicamente por el microcontrolador Picaxe-08. Se lo coloca en la posición Grab, para grabar el

Por ello, flash permite funcionar a velocidades muy superiores cuando los sistemas emplean lectura y escritura en diferentes puntos de esta memoria al mismo tiempo.

programa en el microcontrolador, y en la posición Out, para probar el programa.

En la figura 2.3 muestra el circuito esquemático de los DISPLAYS¹¹. El sistema dispone de 2 displays (D1, D2) de ánodo-común, conectados sobre el mismo bus de datos y con los ánodos manejados independientemente. Se emplea el decodificador **SN74LS47** para la conversión de datos que entrega el microcontrolador (A, B, C, D). La habilitación del display correspondiente se hace conectando un nivel lógico bajo en la base del transistor que maneja el ánodo de dicho display.

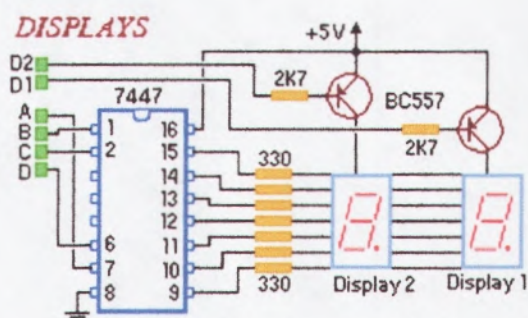


Figura 2.3: Circuito esquemático de la conexión de los 2 displays. Fuente [PICAXE]

En la figura 2.4 muestra el circuito esquemático de los DIODOS LED, el cual dispones de 4 diodos LED¹² (L1, L2, L3, L4). Para encenderlos se necesita un nivel lógico alto en su respectivo acceso.

¹¹ **DISPLAYS** se llama **visualizador**, *display* en inglés, a un dispositivo de ciertos aparatos electrónicos que permite mostrar información al usuario, creado a partir de la aparición de calculadoras, cajas registradoras e instrumentos de medida electrónicos en los que era necesario hacerlo.

¹² **LED: diodo emisor de luz**, el nombre español proviene del acrónimo inglés (*Light-Emitting Diode*: 'diodo emisor de luz') es un dispositivo semiconductor (diodo) que emite luz incoherente de espectro reducido cuando se polariza de forma directa la unión PN del mismo y circula por él una corriente eléctrica. Este fenómeno es una forma de electroluminiscencia. El color, depende del material semiconductor empleado en la construcción del diodo y puede variar desde el ultravioleta, pasando por el visible, hasta el infrarrojo. Los diodos emisores de luz que emiten luz ultravioleta también reciben el nombre de led UV (*ultraviolet light*: 'luz ultravioleta') y los que emiten luz infrarroja se llaman IRED (*InfraRed Emitting Diode*).

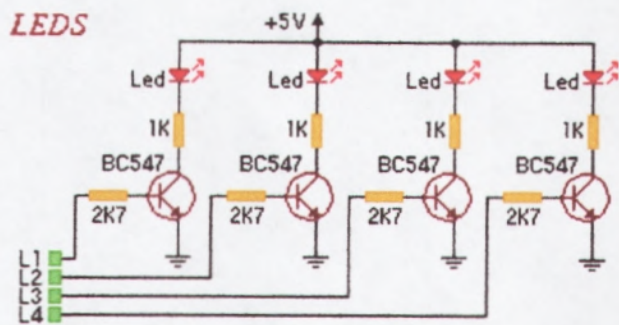


Figura 2.4: Circuito esquemático de la conexión de 4 LED's. Fuente [PICAXE]

En la figura 2.5 se muestra la SALIDA CON RELE, donde se utiliza un relé para manejar cargas de potencia desde el microcontrolador. Para su activación, se requiere de un nivel lógico alto en el pin R.

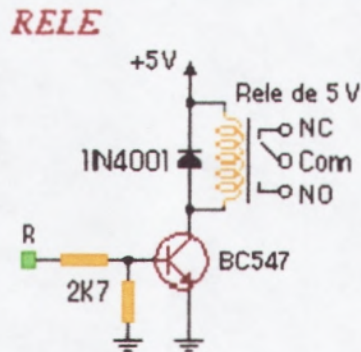


Figura 2.5: Circuito esquemático de la salida con relé. Fuente [PICAXE]

En la figura 2.6 se muestra la conexión de PULSADORES¹³, el cual consta de dos interruptores (pulsadores) normalmente abiertos (**P1**, **P2**), conectados a tierra con resistencias de 10K. En los pines de conexión se tiene normalmente un nivel lógico (0V) bajo, pero cuando se oprimen los botones se obtiene un nivel lógico (5V) alto.

¹³ **PULSADORES:** es un **interruptor eléctrico** es un dispositivo utilizado para desviar o interrumpir el curso de una corriente eléctrica. En el mundo moderno las aplicaciones son innumerables, van desde un simple interruptor que apaga o enciende un bombillo, hasta un complicado selector de transferencia automático de múltiples capas controlado por computadora. Su expresión más sencilla consiste en dos contactos de metal inoxidable y el actuante. Los contactos, normalmente separados, se unen para permitir que la corriente circule. El actuante es la parte móvil que en una de sus posiciones hace presión sobre los contactos para mantenerlos unidos.

PULSADORES



Figura 2.6: Circuito esquemático de pulsadores. Fuente [PICAXE]

En la figura 2.7 se muestra la conexión del DIPSWITCH¹⁴, el mismo tiene cuatro interruptores (**S1, S2, S3, S4**), los cuales se conectan a tierra con resistencias de 10K Ω . Los pines de conexión del dipswitch se tiene un nivel lógico bajo si el interruptor está abierto, pero cuando el interruptor se cierra, se obtiene un nivel lógico alto.

DIPSWITCH

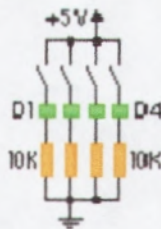


Figura 2.7: Circuito esquemático del Dipswitch. Fuente [PICAXE]

En la figura 2.8 se ilustra la conexión del TECLADO MATRICIAL¹⁵, este tiene un teclado con una matriz de 3x4, con caracteres de 0 a 9, *, y #. Las columnas (**C1, C2, C3**) del teclado tienen conectadas unas resistencias de 2,7K Ω , que sirven para fijar un nivel lógico alto cuando no se está oprimiendo ninguna tecla.

¹⁴ **DIPSWITCH**: es un interruptor conjunto de interruptores manuales eléctricas que están empaquetados en un grupo en un paquete estándar dual en línea (DIP) (la unidad paquete completo también puede ser denominado como un interruptor DIP en singular). Este tipo de interruptor está diseñado para ser utilizado en un tablero de circuito impreso, junto con otros componentes electrónicos y se utiliza para personalizar el comportamiento de un dispositivo electrónico para situaciones específicas.

¹⁵ **TECLADO MATRICIAL**: es un simple *arreglo de botones conectados en filas y columnas*, de modo que se pueden leer varios botones con el mínimo número de pines requeridos.

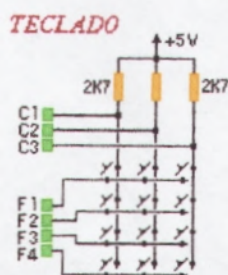


Figura 2.8: Circuito esquemático del Teclado Matricial. Fuente [PICAXE]

En la figura 2.9 se observa el circuito de conexión BUZZER¹⁶, el cual puede crear automáticamente sonidos de diferentes frecuencias utilizando el comando sound.

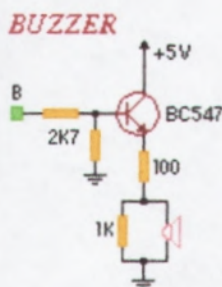


Figura 2.9: Circuito esquemático del Buzzer. Fuente [PICAXE]

En el circuito de la figura 2.10 se ilustra el circuito para los DISPOSITIVOS DE SALIDA, que provee de siete salidas (+5 V_{cc} o +12 V_{cc}) con óptima capacidad de corriente. Para amplificar la corriente de salida del microcontrolador se utiliza el IC ULN2003 como driver de los transistores de salida. La habilitación del transistor de salida correspondiente se hace conectando un nivel lógico alto en los pines de conexión (**S1, S2, S3, S4, S5, S6, S7**). El transistor de salida conmuta a tierra, por lo tanto la carga hay que colocarla entre las salidas (**1, 2, 3, 4, 5, 6, 7**), y la alimentación +5 V_{cc} , o +12 V_{cc} .

¹⁶ **BUZZER:** en inglés, es un dispositivo electrónico que produce un sonido o zumbido continuo o intermitente de un mismo tono. Sirve como mecanismo de señalización o aviso, y son utilizados en múltiples sistemas como en automóviles o en electrodomésticos. Inicialmente este dispositivo estaba basado en un sistema [electromecánico](#) que era similar a una campana eléctrica pero sin el badajo metálico, el cual imitaba el sonido de una campana.

DISPOSITIVOS de SALIDA

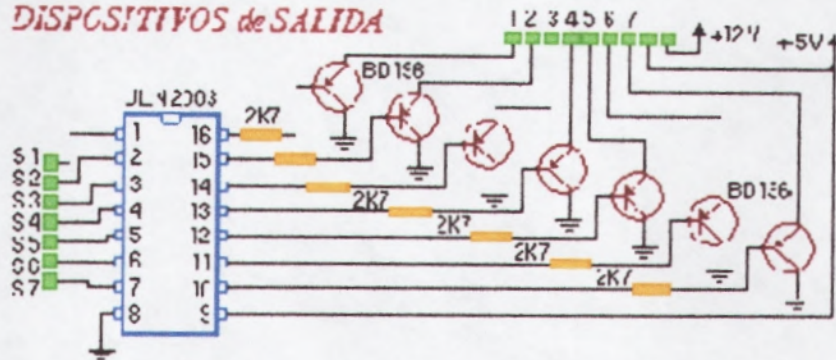


Figura 2.10: Circuito esquemático de los dispositivos de salida. Fuente [PICAXE]

2.3 Análisis de prestaciones del entrenador de microcontroladores PICAXE

Con este entrenador de microcontroladores PICAXE, existe la posibilidad de armar circuitos básicos que son necesarios para cubrir el programa de estudios de Microcontroladores. Debiendo indicarse que se dedica exclusivamente a los microcontroladores PICAXE (ver Anexo 1). Para los μC de Intel y Microchip es necesario utilizar otras tarjetas para cada una de ellas. La desventaja de estos μC es que no son parte del programa de estudio de la asignatura. Por lo tanto no es gran ventaja el uso de este sistema modular de microcontroladores.

2.4 Entrenador de Microcontroladores EasyPic6.

La tarjeta EasyPic6 es producida por la empresa **Mikroelektronika** 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 2.11, en la que se han conservado números referenciales para destacar las características de la tarjeta de una mejor manera.

El Sistema Modular EasyPic6 desarrolla todas las funciones para casi todos los microcontroladores PIC de la familia Microchip. Está diseñado para permitir a los estudiantes y docentes para probar fácilmente y explorar las capacidades de los microcontroladores PIC. También permite a los microcontroladores PIC una interfaz con circuitos externos y una amplia gama de dispositivos periféricos. Los estudiantes pueden, por tanto centrarse en el desarrollo de software solamente.

Hay marcas de identificación a cada componente en una serigrafía, tanto en la parte superior e inferior. Estas marcas describen la conexión con el microcontrolador, los modos de operación y proporcionar información útil adicional de modo que casi no hay necesidad de esquemas adicionales.

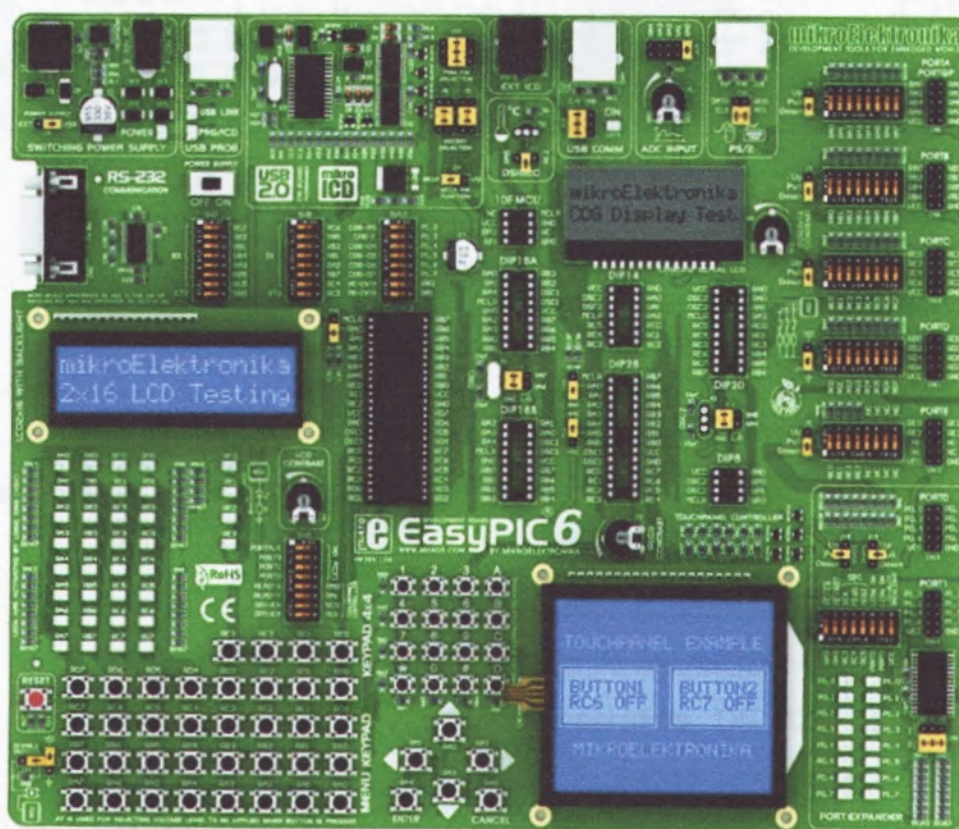


Figura 2.11: Sistema Modular EasyPic6. Fuente [Mikroelektronika]

El Sistema Modular EasyPic6 desarrolla todas las funciones para casi todos los microcontroladores PIC de la familia Microchip. Está diseñado para permitir a los estudiantes y docentes para probar fácilmente y explorar las capacidades de los microcontroladores PIC. También permite a los microcontroladores PIC una interfaz con circuitos externos y una amplia gama de dispositivos periféricos. Los estudiantes pueden, por tanto centrarse en el desarrollo de software solamente.

Hay marcas de identificación a cada componente en una serigrafía, tanto en la parte superior e inferior. Estas marcas describen la conexión con el microcontrolador, los modos de operación y proporcionar información útil adicional de modo que casi no hay necesidad de esquemas adicionales.

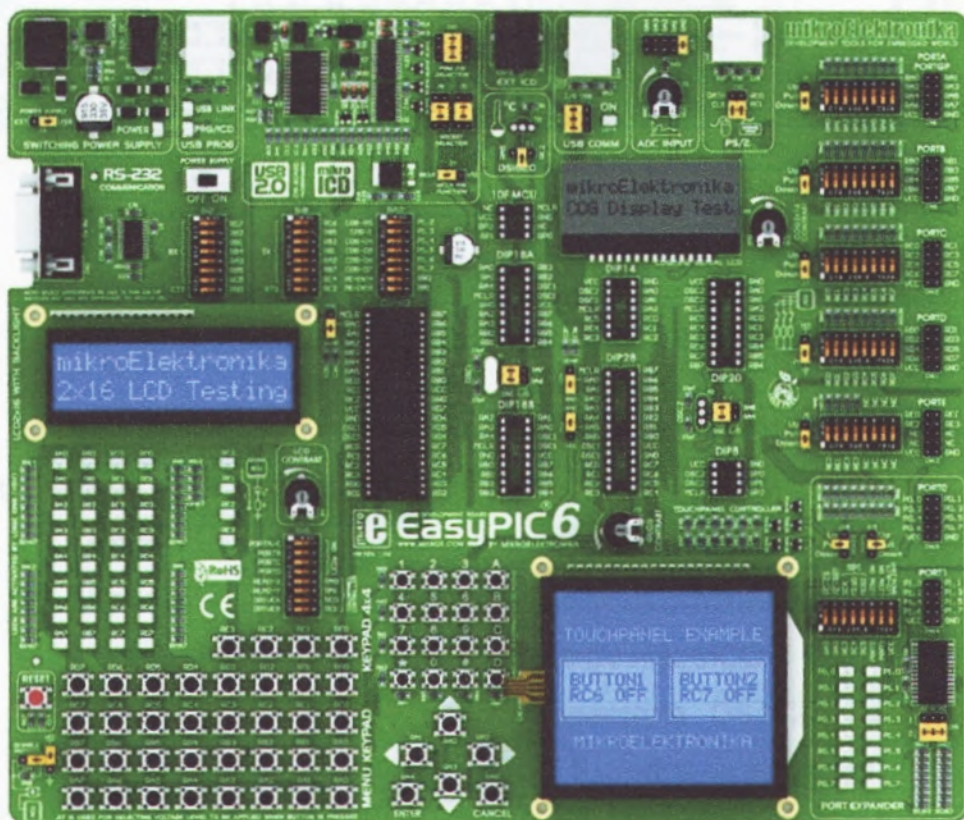


Figura 2.11: Sistema Modular EasyPic6. Fuente [Mikroelektronika]

El Sistema Modular EasyPic6 desarrolla todas las funciones para casi todos los microcontroladores PIC de la familia Microchip. Está diseñado para permitir a los estudiantes y docentes para probar fácilmente y explorar las capacidades de los microcontroladores PIC. También permite a los microcontroladores PIC una interfaz con circuitos externos y una amplia gama de dispositivos periféricos. Los estudiantes pueden, por tanto centrarse en el desarrollo de software solamente.

Hay marcas de identificación a cada componente en una serigrafía, tanto en la parte superior e inferior. Estas marcas describen la conexión con el microcontrolador, los modos de operación y proporcionar información útil adicional de modo que casi no hay necesidad de esquemas adicionales.

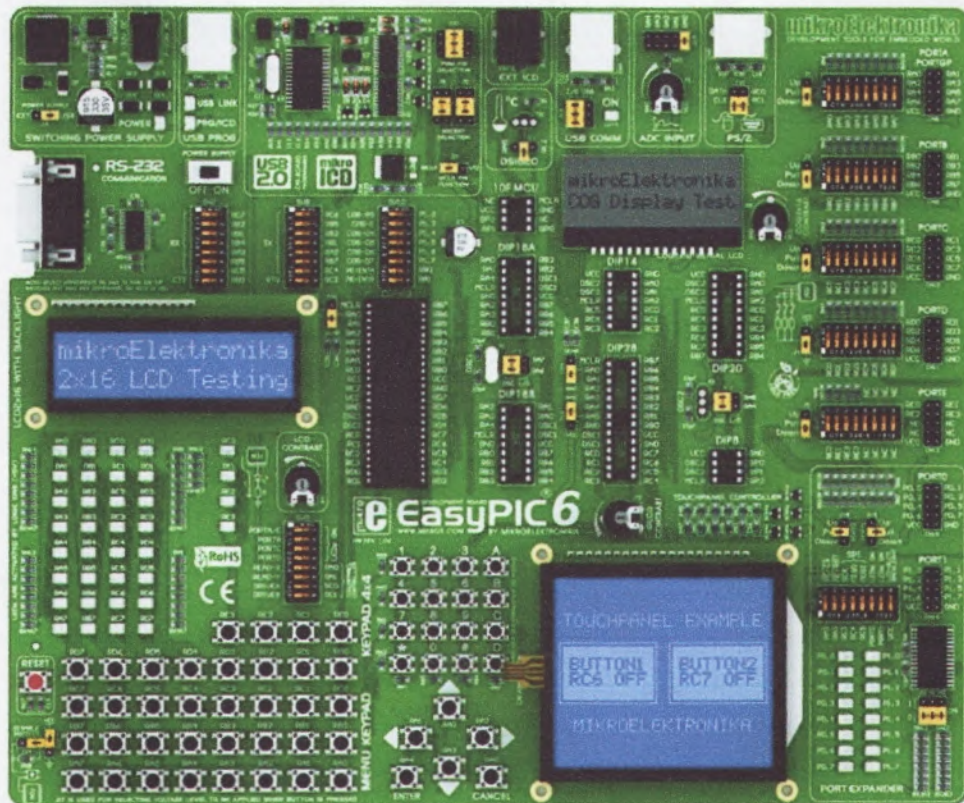


Figura 2.11: Sistema Modular EasyPic6. Fuente [Mikroelektronika]

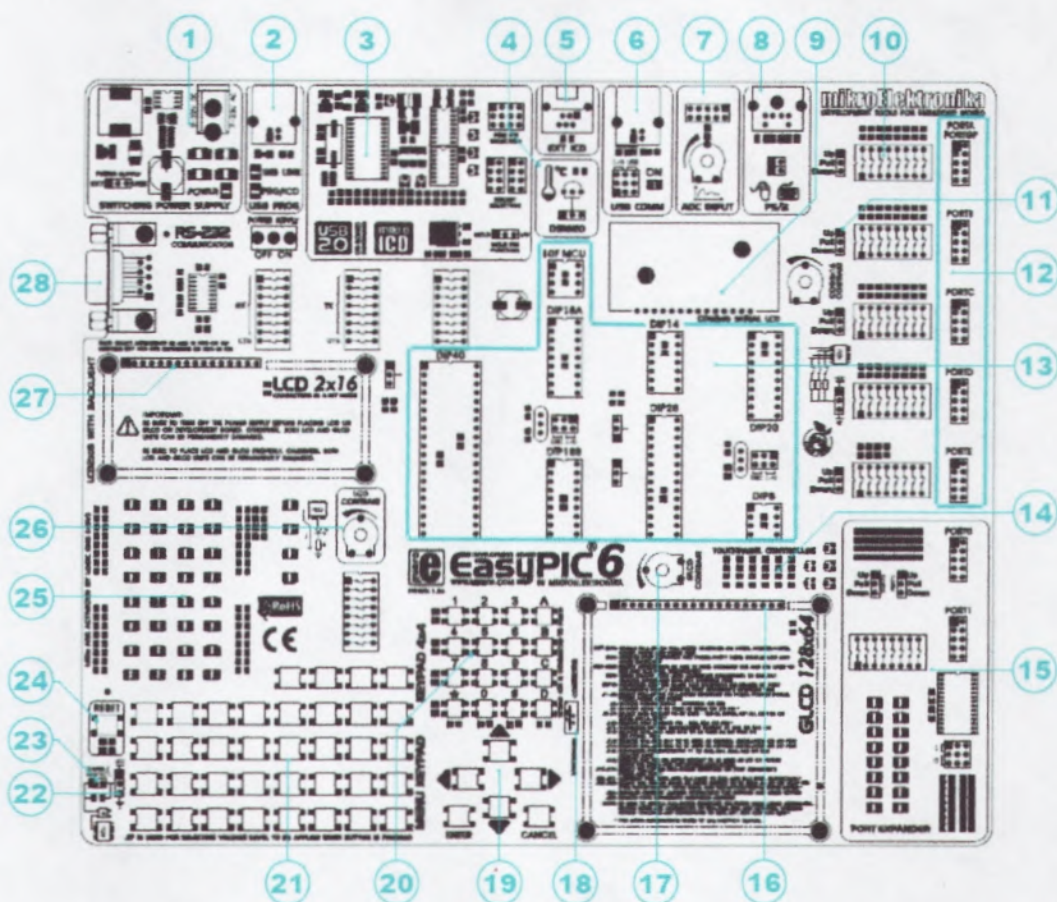


Figura 2.12: Diagrama esquemático del sistema modular EasyPic6. Fuente [Mikroelektronika]

En la figura 2.1.2 se observa con mayor detalle las siguientes características de la numeración presentada:

- 1) Regulador del voltaje de la fuente de alimentación.
- 2) Conector USB¹⁷ para el programador incorporado.
- 3) Programador USB 2.0 con soporte de mikroICD.
- 4) Zócalo para el sensor de temperatura DS1820 que permite mediciones de temperatura con 0.5 grados de exactitud.
- 5) Conector para el depurador externo (ICD2 o ICD3) de Microchip.
- 6) Conector para la comunicación USB.

¹⁷ **USB: Universal Serial Bus** (bus universal en serie) o **Conductor Universal en Serie (CUS)**, abreviado comúnmente **USB**, es un puerto que sirve para conectar periféricos a un ordenador. Fue creado en 1996 por siete empresas (que actualmente forman el consejo directivo): IBM, Intel, Northern Telecom, Compaq, Microsoft, Digital Equipment Corporation y NEC.

- 7) Entradas para probar el convertidor A/D.
- 8) Conector PS/2.
- 9) Visualizador LCD 2x16 incorporado.
- 10) Interruptor DIP que permiten el funcionamiento de las resistencias pull-up/pull-down.
- 11) Puente para seleccionar las resistencias pull-up/pull-down.
- 12) Conectores de los puertos de E/S.
- 13) Zócalo para colocar el microcontrolador PIC.
- 14) Controlador del panel táctil. microcontrolador.
- 15) Extensor de puertos.
- 16) Conector del visualizador LCD gráfico 128x64.
- 17) Potenciómetro de contraste del visualizador LCD gráfico 128x64.
- 18) Conector de panel táctil.
- 19) Teclado menú.
- 20) Teclado 4x4.
- 21) Botones de presión para simular las entradas digitales.
- 22) Puente para seleccionar el estado lógico de los botones de presión.
- 23) Puente para poner en cortocircuito una resistencia de protección.
- 24) Botón para resetear el microcontrolador.
- 25) 36 diodos LED que indican el estado lógico en los pines.
- 26) Ajuste de contraste del visualizador LCD alfanumérico.
- 27) Conector del visualizador LCD alfanumérico.
- 28) Conector para la comunicación RS-232.

EasyPic6 fue diseñado como una herramienta de apoyo a la enseñanza de los microcontroladores, cuyas características son 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.

microcontrolador que se pueden utilizar en este sistema son los modelos de PIC 16F84, 16F627, 16F628, 16F877, 16F874 y 18F452 y aquellos otros que sean compatibles en PINOUT con estos modelos.

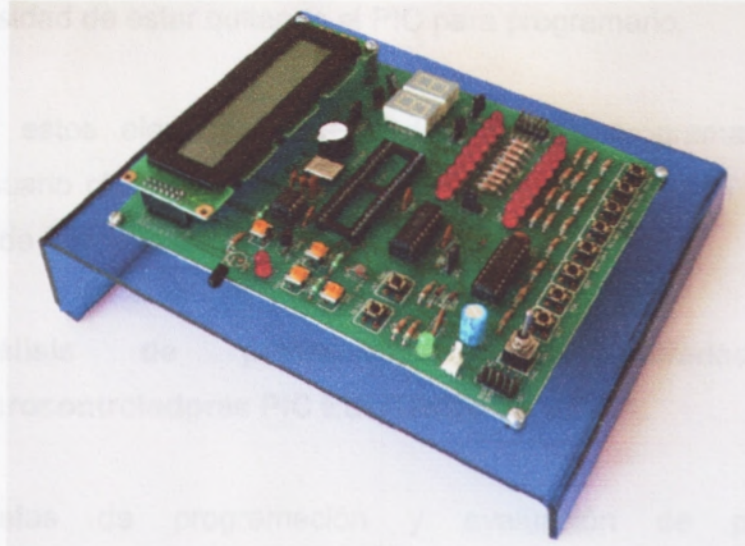


Figura 2.14: Tarjeta de la etapa de evaluación de programas. Fuente [EDUTRÓNICA]

Para este sistema de evaluación de programas se detalla sus componentes:

- 1) Display de Cristal Líquido de 2x20 caracteres (LCD).
- 2) Doble Display de 7 segmentos.
- 3) Puertos con salida de LED's.
- 4) Un puerto con Interruptor Táctil.
- 5) Interruptor táctil para interrupción externa (RB0/INT).
- 6) Buzzer Piezoeléctrico.
- 7) Sensor analógico de luz (Fotorresistencia).
- 8) Sensor de Temperatura (LM35DZ).
- 9) Dos resistencias variables para voltaje variable (0-5 VCD).
- 10) Sensor infrarrojo para decodificación de señales infrarrojas. (Control Remoto).
- 11) Interruptor de Reset.

- 12) Conector "header" macho del Puerto B para cargas externas al sistema.
- 13) Conector "header" macho para programación en línea del PIC, sin necesidad de estar quitando el PIC para programarlo.

Mediante estos elementos se pueden realizar programas que le permiten al usuario rápidamente elevar el nivel y habilidad en el desarrollo de programas de interés para sus aplicaciones.

2.7 Análisis de prestaciones del Entrenador de Microcontroladores PIC EDUTRONICA JJTSA.

Las tarjetas de programación y evaluación de programas EDUTRÓNICA presentan un entorno similar de trabajo que del PICAXE en el sentido de su apariencia con display LCD, LED's, potenciómetros, área para prototipos. Sus diferencias fundamentales radican en que trabajan con microcontroladores PIC (de Microchip) y PICAXE, así como también la cantidad de pulsadores que es muy inferior. La fortaleza de la tarjeta radica en la presentación de la utilización de características avanzadas en microcontroladores.

2.8 Consideraciones finales de los sistemas de entrenamiento de microcontroladores.

Una vez realizado el análisis de las características generales de las tres tarjetas, es evidente que para el curso actual de Microcontroladores la que mejor cubre las necesidades prácticas es la tarjeta del entrenador de microcontroladores EasyPic6. En el capítulo 3 se presenta el estado del arte del EasyPic6.

CAPÍTULO 3

SISTEMA MODULAR DE ENTRENAMIENTO EASYPIC6

3.1 Introducción de EasyPic6

El sistema de desarrollo EasyPic6 es una herramienta de desarrollo extraordinaria, adecuada para la programación y la experimentación con los microcontroladores PIC® de la compañía Microchip. El sistema dispone de un programador incorporado con el soporte de mikroICD® (depurador en circuito a nivel de hardware) que proporciona un interfaz entre el microcontrolador y la PC. Se espera del docente y alumnos que escriban un código en alguno de nuestros compiladores, que genere un fichero **.hex** y que programe el microcontrolador utilizando el programador PICflash™.

Los módulos numerosos, como visualizador gráfico LCD de 128x64 píxeles, visualizador LCD de 2x16 caracteres, visualizador LCD incorporado de 2x16 caracteres, teclado 4x4, extensor de puertos etc., le permiten simular con facilidad el funcionamiento del dispositivo destino. En el capítulo 2 se analizó brevemente la tarjeta completa del EasyPic6 (Figura 2.11), la misma que está compuesta por bloques detallados a continuación:



Placa de desarrollo para microcontroladores PIC con todas las prestaciones y de uso amigable.



Programador USB 2.0 de alto rendimiento en la placa.



Depurador Hardware en circuito para una depuración paso a paso a nivel de hardware.



Extensor de puertos proporciona una fácil expansión de E/S (2 puertos adicionales) utilizando interfaz serie.



Visualizador LCD serie de 2x16 caracteres.



Visualizador LCD gráfico con iluminación de fondo

3.2 Conexión del Sistema al PC o Notebook.

El sistema EasyPic6 es conectado a la PC o Notebook a través de un cable USB. Una punta del cable USB proporcionado con un conector de tipo "USB-B" se conecta al sistema de desarrollo como se muestra en la Figura 3.1. La otra punta del cable con un conector USB de tipo A se conecta a la PC. Al establecer la conexión, asegúrese de que el puente J6 se coloque en la posición USB como se muestra en la Figura 3.2.

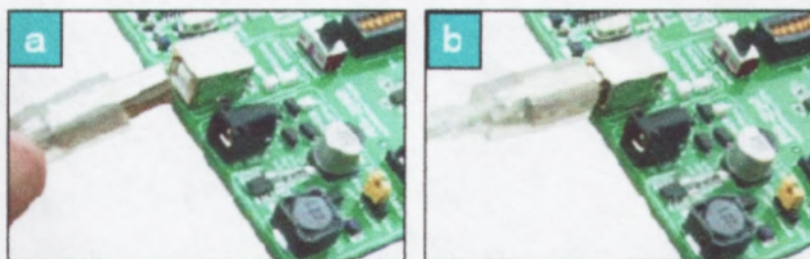


Figura 3.1: Conexión USB del EasyPic6 al PC o Notebook. a) Verificar la forma de colocar el conector USB. b) Finalmente la conexión a la tarjeta. Fuente [Mikroelektronika]

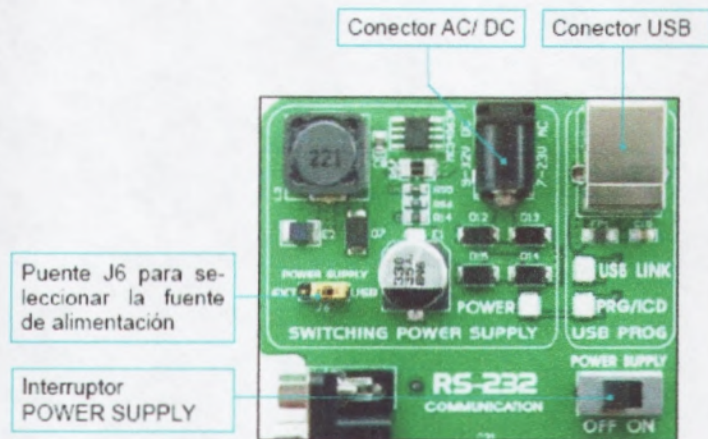


Figura 3.2: Fuente de alimentación. Fuente [Mikroelektronika]

Es importante seguir con las instrucciones proporcionadas en los manuales pertinentes para instalar los controladores USB y el programa **PICflash** desde el CD del entrenador EasyPic6. Estos controladores USB son imprescindibles para el funcionamiento adecuado del programador incorporado. En caso de que ya tenga un compilador PIC de Mikroelektronika instalado en la PC, no es necesario reinstalar los controladores ya que se van a instalar automáticamente junto con el compilador.

Posterior a esto es necesario el encendido del sistema de desarrollo al poner el interruptor *POWER SUPPLY*¹⁹ en la posición *ON*. Dos diodos *LED* etiquetados con *POWER* y *USB LINK* se activan automáticamente para indicar que el sistema de desarrollo está listo para su uso. La ventaja de este sistema es que incluye un programador incorporado y de la interfaz (**PICflash**) para cargar un código **hex** (sistema hexadecimal) en el microcontrolador. Una vez realizado este proceso, se puede utilizar la placa para probar y desarrollar proyectos de aplicación con los microcontroladores (ver Anexo 2).

¹⁹ **POWER SUPPLY**, Dispositivo o sistema que provee electricidad u otros tipos de energía, aunque el término es especialmente usado para referirse a energía eléctrica.

El sistema puede funcionar con módulos adicionales, tales como LCD, GLCD, placas adicionales etc., es necesario colocarlos apropiadamente en el sistema de desarrollo antes de encender el sistema. De lo contrario, pueden quedarse dañadas permanentemente (Ver Figura 3.3)

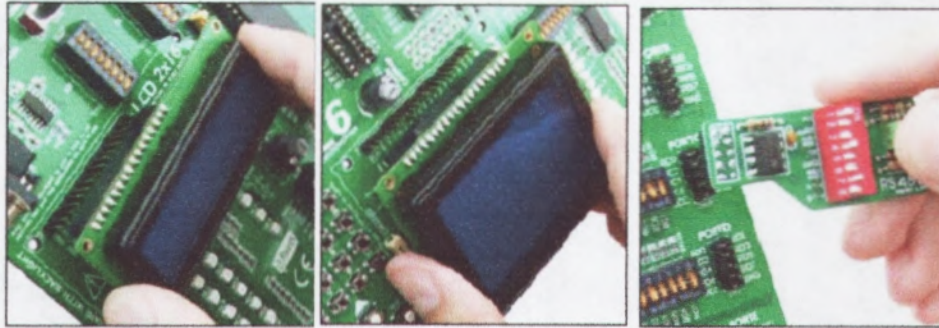


Figura 3.3: Correcta conexión de los módulos adicionales a la tarjeta EasyPic6. Fuente [Mikroelektronika]

3.3 Microcontroladores soportados.

El sistema modular de entrenamiento EasyPic6 tiene ocho zócalos (ver figura 3.4) separados para la inserción de microcontroladores en los encapsulados DIP40 (PIC de 40 pines, ejemplo: PIC16F877A), DIP28 (PIC de 28 pines, ejemplo: PIC16F870), DIP20 (PIC de 20 pines, ejemplo: PIC16f86), DIP18 (PIC de 18 pines, ejemplo PIC16F628A), DIP14 (PIC de 14 pines, ejemplo: PIC16F505) y DIP8 (PIC de 8 pines, ejemplo: PIC12F519).

Los 8 zócalos permiten que los dispositivos soportados en los DIP (encapsulados) estén conectados directamente a la placa de desarrollo. En la placa se proporcionan dos zócalos para microcontroladores PIC en el encapsulado DIP 18. El zócalo a utilizar depende exclusivamente de la disposición de los pines en el microcontrolador utilizado. El entrenador EasyPIC6 viene con el microcontrolador en el encapsulado DIP 40.

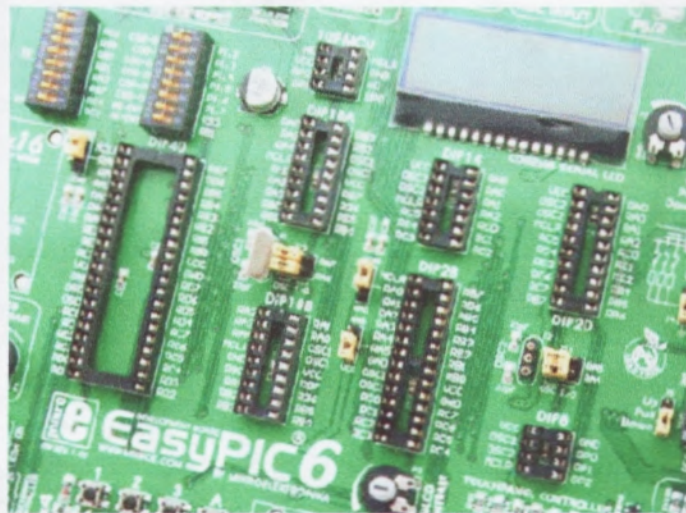


Figura 3.4: Placa que muestra los 8 zócalos de la tarjeta EasyPic6. Fuente [Mikroelektronika]

Los puentes junto a los zócalos se utilizan para seleccionar las funciones de los pines del microcontrolador mostrados en la Tabla 3.1:

Puente	Posición
J22	RA0 – Pin de E/S (Entradas/Salidas) VCAP – Capacitor de filtro (para el PIC16F724/727)
J23	VCAP – Capacitor de filtro (para el PIC16F722/723) RA0 – Pin de E/S (Entradas/Salidas)
J16	RA5 – Pin de E/S (Entradas/Salidas) VCC – Fuente de alimentación para microcontroladores PIC18F2331/2431.
J13	OSC – RA6, RA7 son pines de OSC. I/O – RA6, RA7 son pines de E/S (Entrada/Salida)
J14	OSC – RA4, RA5 son pines de OSC. I/O – RA4, RA5 son pines de E/S (Entrada/Salida)

Tabla 3.1: Funciones de los puentes.

Los microcontroladores PIC utilizan con frecuencia un oscilador (cristal de cuarzo) para estabilizar la frecuencia de reloj. El **EasyPic6** adicional cuenta con dos zócalos para colocar el oscilador. En los encapsulados DIP18A, DIP18B, DIP28 y DIP40 utilizan el zócalo (OSC₁) X₁ para el cristal

de cuarzo. En caso de que se requiera el uso de los encapsulados DIP8, DIP14 y DIP20 es necesario mover el oscilador del zócalo X_1 al zócalo X_2 (OSC_2). Además, el oscilador es reemplazable por cualquier otro cristal de cuarzo. El valor del oscilador depende de la frecuencia de reloj máxima permitida. Los microcontroladores colocados en el zócalo 10F utilizan un oscilador interno. No están conectados a ningún zócalo del cristal de cuarzo anteriormente mencionado.

Antes de colocar el microcontrolador en el zócalo adecuado, asegúrese de que la fuente de alimentación esté apagada. En la Figura 3.5 muestra cómo colocar correctamente un microcontrolador. La Figura 3.5a) muestra el zócalo DIP 40 vacío. Coloque una punta del microcontrolador en el zócalo como se muestra en la Figura 3.5b). Entonces ponga lentamente el microcontrolador más abajo hasta que los pines encajen en el zócalo, como se muestra en la Figura 3.5c).

Compruebe una vez más si todo está colocado correctamente y presione el microcontrolador lentamente hasta que encaje en el zócalo completamente, como se muestra en la Figura 3.5d).

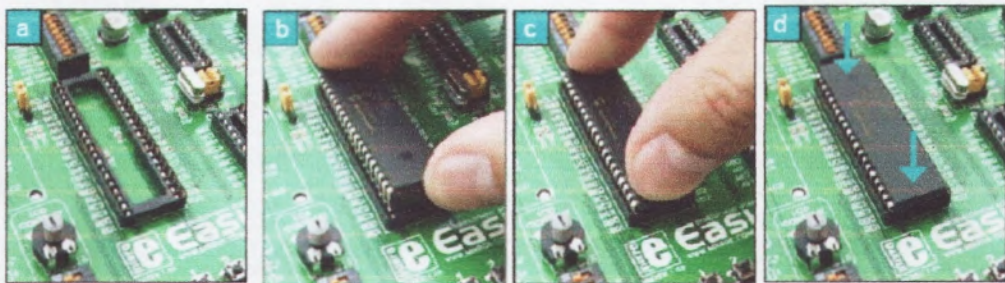


Figura 3.5: Pasos para colocar correctamente el Microcontrolador en el zócalo apropiado. a) Zócalo de 40 pines; b) Inserción inicial del PIC; c) Inserción en la parte final del encapsulado (PIC); d) Correcta inserción del PIC, caso contrario no funcionaría. Fuente [Mikroelektronika]

3.4 El programador incorporado USB 2.0 PICflash.

El programador **PICflash** es una herramienta obligatoria al trabajar con microcontroladores. El sistema de desarrollo **EasyPic6** dispone del programador **PICflash** integrado con soporte de **mikroICD** (ver figura 3.6). El programador **PICflash** permite establecer una conexión entre el PC y el microcontrolador. Utilice el programador **PICflash** para cargar un fichero .hex en el microcontrolador. En la Figura 3.7 se muestra la conexión entre el compilador, el programador PICflash y el microcontrolador.

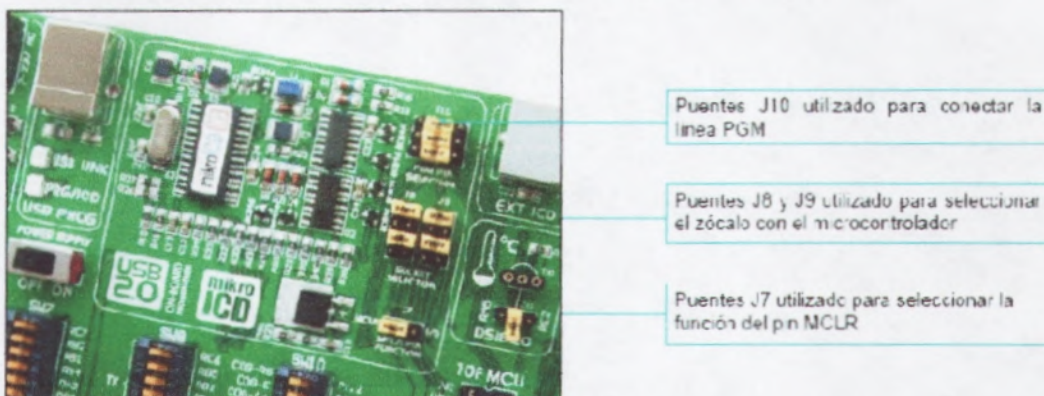


Figura 3.6: Programador PICflash con mikroICD. Fuente [Mikroelektronika]

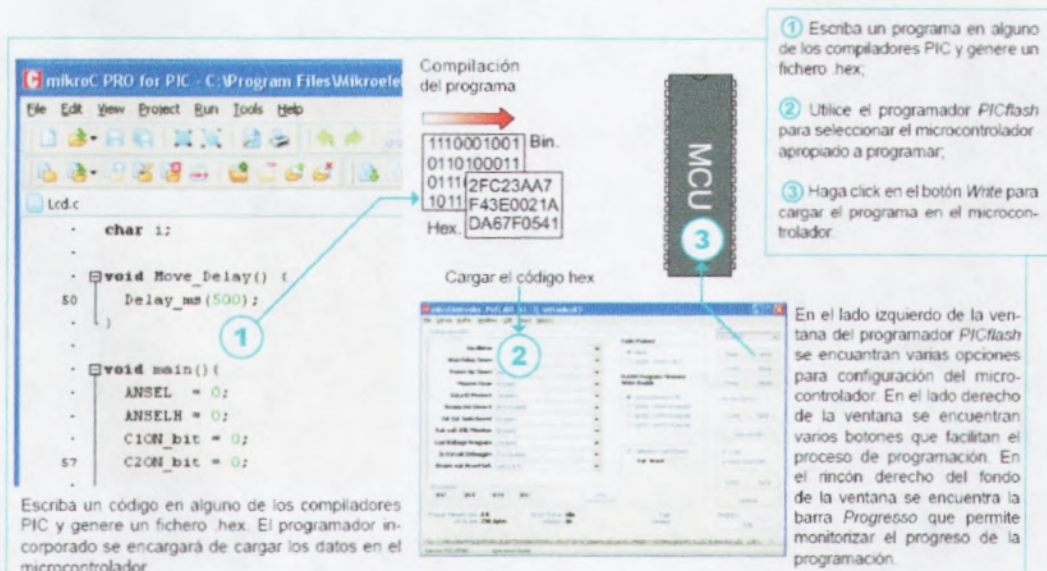


Figura 3.7: Detalle básico de funcionamiento del programador. Fuente [Mikroelektronika]

Los microcontroladores PIC se pueden programar de dos formas: en modo de programación de alto o bajo voltaje.

- ✓ **El modo de programación de bajo voltaje**, se puede habilitar/deshabilitar al utilizar los bits de configuración del microcontrolador. Cuando el modo de programación de bajo voltaje está habilitado, el proceso de programación se inicia al aplicar un uno lógico (1) al pin PGM.
- ✓ **El modo de programación de alto voltaje** está siempre habilitado y el proceso de programación se inicia al aplicar un voltaje alto al pin MCLR/VPP.

El programador **PICflash** utiliza únicamente el modo de alto voltaje. Este modo requiere llevar un voltaje más alto que el voltaje de la fuente de alimentación del microcontrolador (el rango de 8 a 14V, lo que depende de tipo del microcontrolador utilizado) al pin MCLR/pp para realizar el proceso de la programación. Para programar el microcontrolador en modo de alto voltaje es necesario aplicar un cero lógico (0) al pin PGM, lo que deshabilita que el microcontrolador entre en modo de programación de bajo voltaje. Dependiendo del microcontrolador utilizado es posible seleccionar uno de los siguientes pines RB3, RB4 y RB5 para utilizarlo como el pin PGM. El puente J10 se utiliza para seleccionar el pin PGM como se muestra en la Figura 3.8.

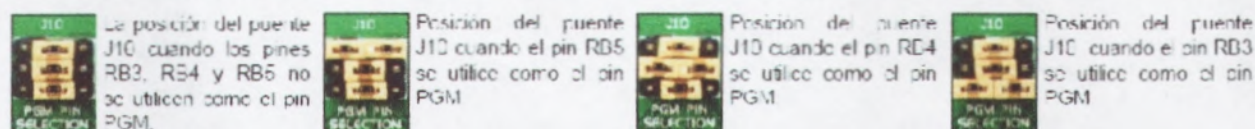


Figura 3.8: Diversas posiciones del puente J10. Fuente [Mikroelektronika]

Durante la programación, el multiplexor mostrado en la figura 3.9 desconecta los pines del microcontrolador, utilizados para la programación, del resto de la placa y los conecta al programador **PICflash**. Una vez

terminada la programación, los pines se desconectan del programador y se pueden utilizar como los pines de E/S.

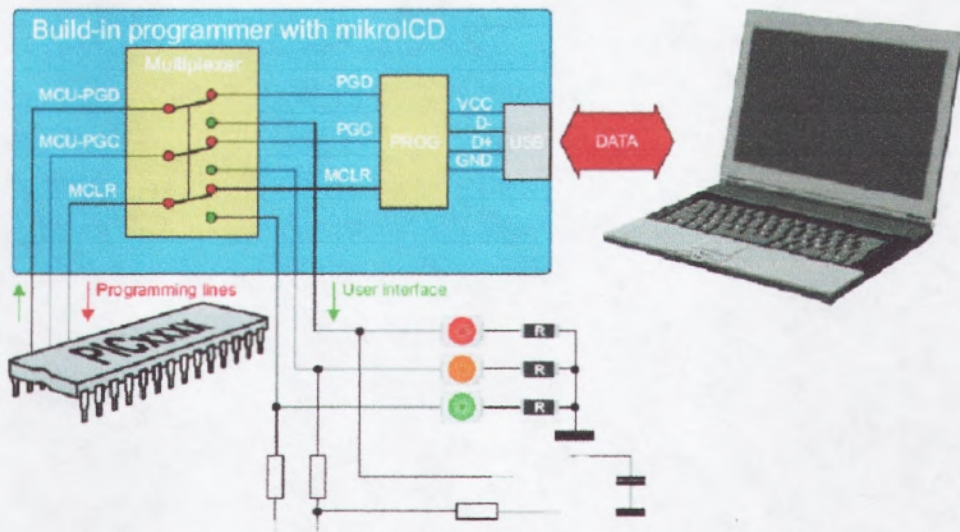
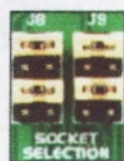


Figura 3.9: Esquema del programador EasyPic6. Fuente [Mikroelektronika]

En la Figura 3.10 se muestran cada una de las posiciones de los puentes J8 y J9 que dependen de los zócalos DIP utilizados. Los puentes J8 y J9 son usados para seleccionar el zócalo a recibir una señal de programación.



Microcontrolador se coloca en uno de los siguientes zócalos: DIP40, DIP28, DIP18A o DIP18B.



Microcontrolador se coloca en uno de los siguientes zócalos: DIP20, DIP14 o DIP8.

Figura 3.10: Posición de los puentes J8 y J9. Fuente [Mikroelektronika]

El pin MCLR (*Master Clear*, literalmente borrado maestro) depende de la posición del puente J7 (ver figura 3.11). Al colocarlo en la posición MCRL se utiliza como el pin MCRL /Vpp. De lo contrario, al colocarlo en la posición E/S, está disponible como un pin de E/S.



Pin MCLR utilizado como el pin MCLR/Vpp



Pin MCLR utilizado como un pin de I/O

Figura 3.11: Posición del puente J7. Fuente [Mikroelektronika]

3.5 Depurador en circuito a nivel de hardware (mikroICD).

El *mikroICD* (depurador en circuito) es una parte integral del programador incorporado. Se utiliza con el propósito de probar y depurar programas en tiempo real. El proceso de probar y depurar se realiza al monitorizar los estados de todos los registros dentro del microcontrolador durante su funcionamiento en entorno real. El software *mikroICD* está incorporado en todos los compiladores PIC diseñados por MikroElektronika (**mikroBASIC PRO** for PIC, **mikroC PRO** for PIC and **mikroPASCAL PRO** for PIC).

Tan pronto como se inicie el depurador mikroICD, aparecerá una ventana, como se muestra en la figura más abajo. El depurador mikroICD se comunica con el PC por los pines de programación que por eso no se pueden utilizar como pines de E/S durante la ejecución de la depuración de programa.

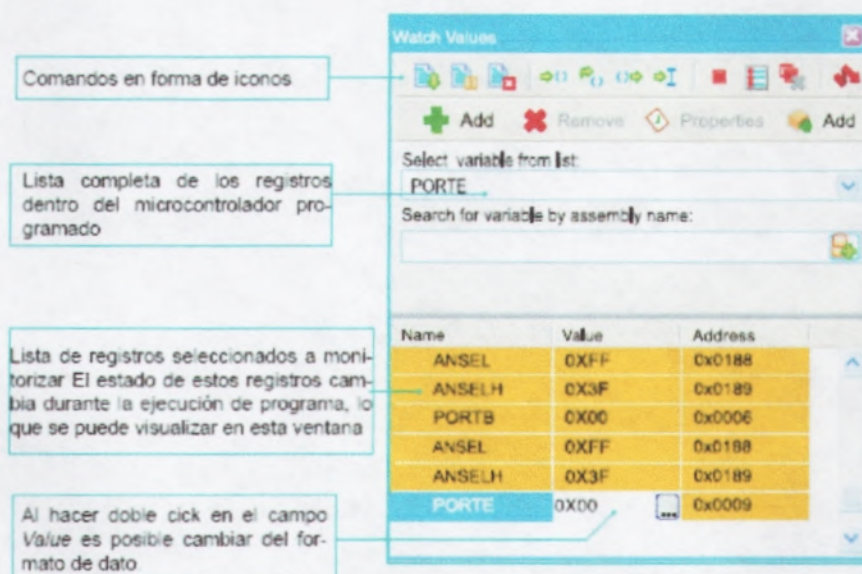


Figura 3.12: Ventana *Watch Values* del *mikroICD*. Fuente [Mikroelektronika]

El depurador mikroICD también ofrece funciones tales como ejecutar el programa paso a paso, detener la ejecución de programa para monitorizar el estado de los registros actualmente activos por medio de los puntos de

ruptura, monitorizar los valores de algunas variables etc. El siguiente ejemplo muestra una ejecución de programa paso a paso utilizando el comando *Step Over*.

Ejemplo:

- ✓ En la figura 3.13a muestra la línea de programa 41 la cual es marcada por el color azul, lo que quiere decir que es la siguiente en ser ejecutada. El estado actual de todos los registros dentro del microcontrolador se puede visualizar en la ventana *Watch Values* del *mikroICD*
- ✓ Después de ejecutar el comando *Step Over* el microcontrolador ejecutará la línea de programa 41 (ver figura 3.13b). La siguiente línea en ser ejecutada está marcada por el azul. El estado de los registros cambiados durante la ejecución de esta instrucción se puede visualizar en la ventana *Watch Values*.

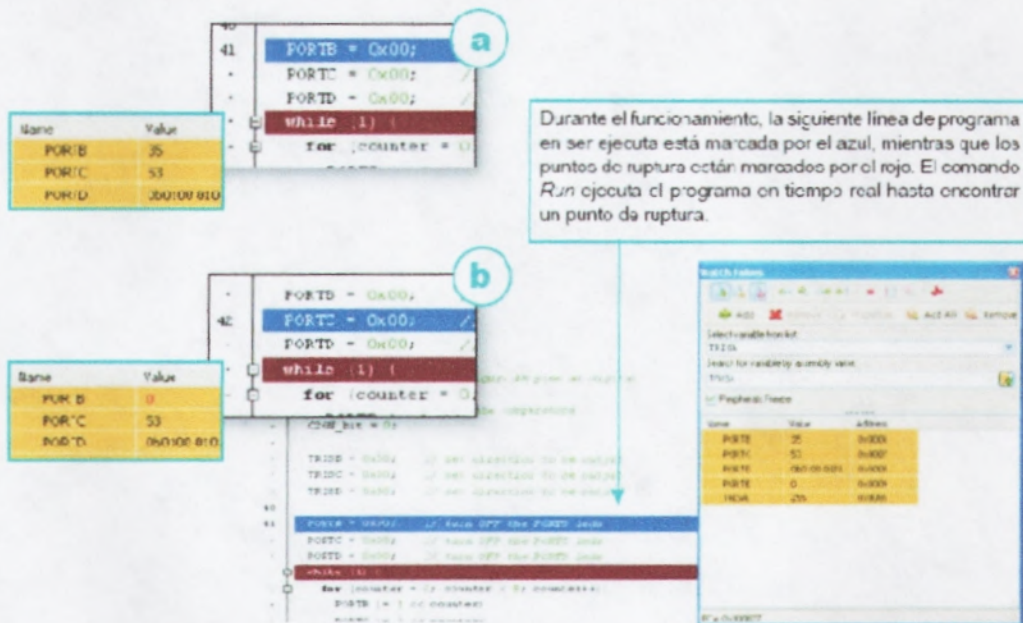


Figura 3.13: Ejecución del comando *Step Over*. a) Ejecución de línea 41 de programación, b) Ejecución de la instrucción en la ventana *Watch Values*. Fuente [Mikroelektronika]

3.6 Fuente de alimentación del entrenador EasyPic6.

El sistema modular de entrenamiento EasyPic6 puede utilizar una de las dos fuentes de alimentación que viene incorporada al sistema como se muestra en la figura 3.14:

1. La Fuente de alimentación de +5V desde el PC por el cable USB de programación;
2. La fuente de alimentación externa conectada a un conector AC/DC proporcionado por el entrenador.

El regulador del voltaje de la fuente de alimentación MC34063A y el rectificador *Gretz* permiten que el voltaje de la fuente de alimentación externa sea AC (en el rango de 7V a 23V) o DC (en el rango de 9V a 32V). El puente J6 se utiliza como selector de la fuente de alimentación. Cuando se utilice la fuente de alimentación USB, el puente J6 debe estar en la posición USB. Cuando se utilice la fuente de alimentación externa, el puente J6 debe estar en la posición EXT.

El sistema de desarrollo se enciende/apaga al cambiar de posición del interruptor de POWER SUPPLY.

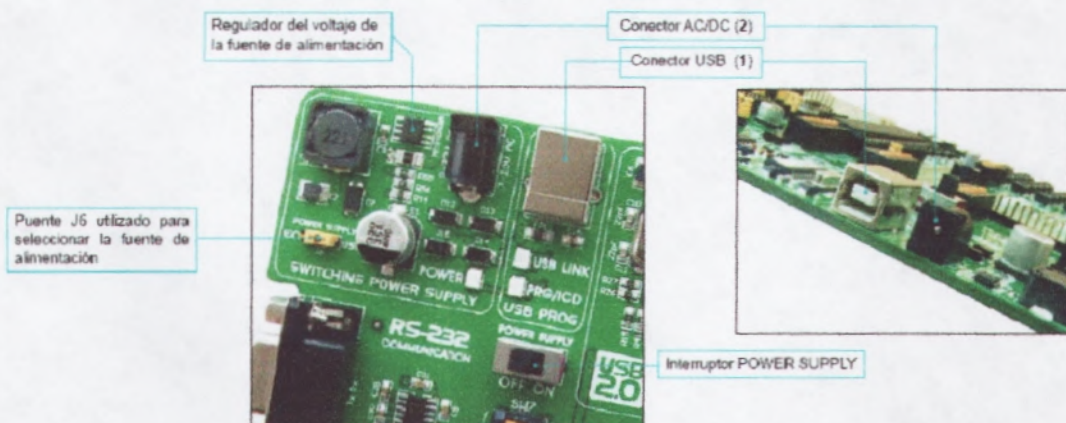


Figura 3.14: Fuente de alimentación del entrenador EasyPic6. Fuente [Mikroelektronika]

En la ilustración de la figura 3.15, el programador utiliza el interruptor MOSFET²⁰ (IRFR9024N) para detener la fuente de alimentación en el sistema de desarrollo durante la programación. Al acabar el proceso de la programación, el programador habilita la alimentación del sistema de desarrollo.

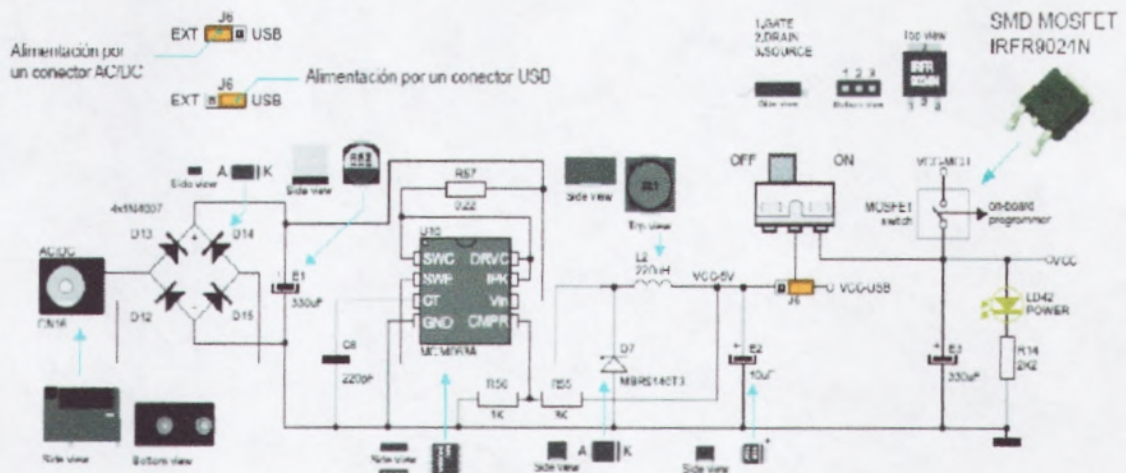


Figura 3.15: Esquemático de la conexión de la fuente de alimentación del entrenador EasyPic6. Fuente [Mikroelektronika]

3.7 El módulo de comunicación RS-232

La comunicación serial RS232 (ver figura 3.16) se realiza por medio de un conector SUB-D de 9 pines y el módulo USART²¹ del microcontrolador. Para habilitar tal comunicación, es necesario establecer una conexión entre las líneas de comunicación Rx (recibir datos) y Tx (transmitir datos) y los pines del microcontrolador proporcionados con el módulo USART por medio de un interruptor DIP. Las líneas utilizadas en tal comunicación serial están marcados de la siguiente manera: Rx (*receive data*); Tx (*transmit data*);

²⁰ **MOSFET** son las siglas de *Metal Oxide Semiconductor Field Effect Transistor*. Consiste en un transistor de efecto de campo basado en la estructura MOS. Es el transistor más utilizado en la industria microelectrónica. Prácticamente la totalidad de los circuitos integrados de uso comercial están basados en transistores MOSFET.

²¹ **USART** (*universal synchronous/asynchronous receiver/transmitter*) - transmisor/receptor síncrono/asíncrono universal es una de las formas más frecuentes de intercambiar los datos entre la PC y los periféricos.

CTS (clear to send) - listo para transmitir; RTS (request to send) - listo para enviar.

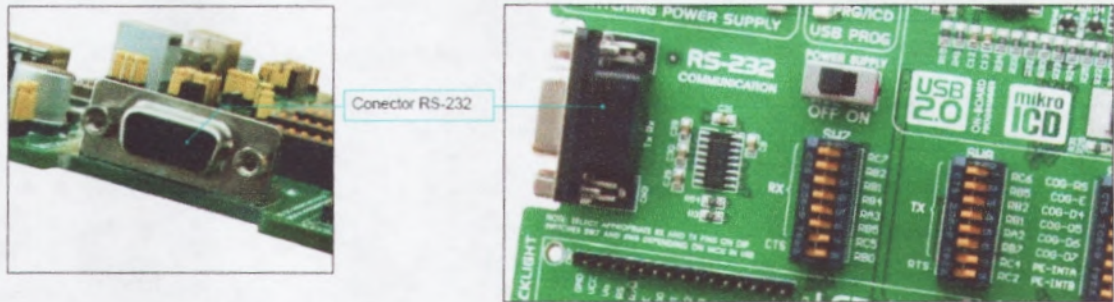


Figura 3.16: Módulo RS-232 del EasyPic6. Fuente [Mikroelektronika]

La velocidad de transmisión en baudios es hasta 115 kbps. Para habilitar el módulo USART del microcontrolador y pueda recibir las señales de entrada de acuerdo con el estándar RS-232 es necesario ajustar los niveles de voltaje por medio de un IC tal como MAX202C. La función de los interruptores DIP SW7 y SW8 (ver figura 3.17) es de determinar cuáles pines del microcontrolador se utilizarán como líneas Rx y Tx. La disposición de los pines difiere dependiendo del tipo de microcontrolador. Se sugiere revisión de que su microcontrolador este proporcionado por el módulo USART, ya que no todos los microcontroladores están integrado a él.

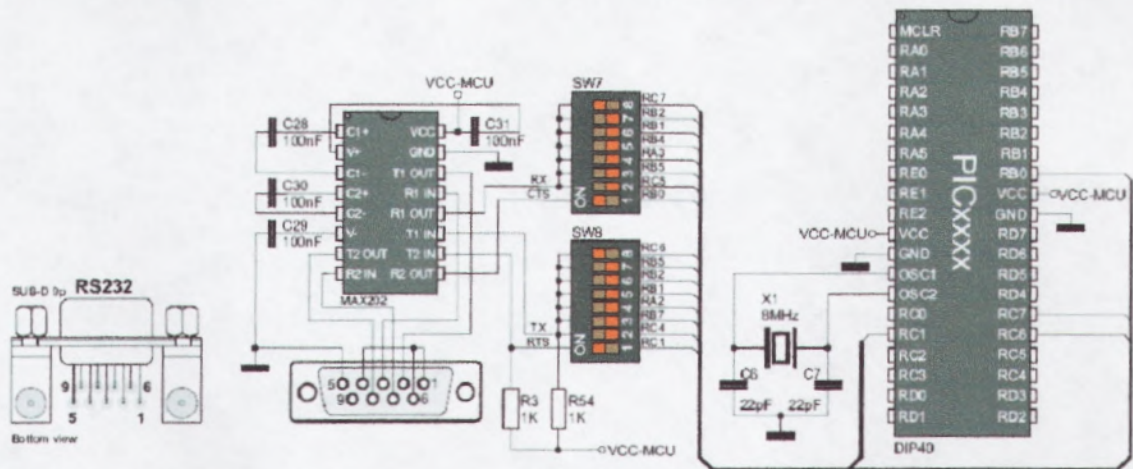


Figura 3.17: Esquema de conexión del módulo RS-232. Fuente [Mikroelektronika]

3.8 Módulo de comunicación PS/2

El PS/2 (ver el esquemático en la figura 3.18) habilita conectar los dispositivos de entrada, tales como teclado y ratón al sistema modular EasyPic6. Para habilitar la comunicación PS/2 es necesario colocar apropiadamente los puentes J20 y J21 como se muestra en la figura 3.19, de manera que las líneas *DATA* y *CLK* estén conectadas a los pines del microcontrolador RC0 y RC1.

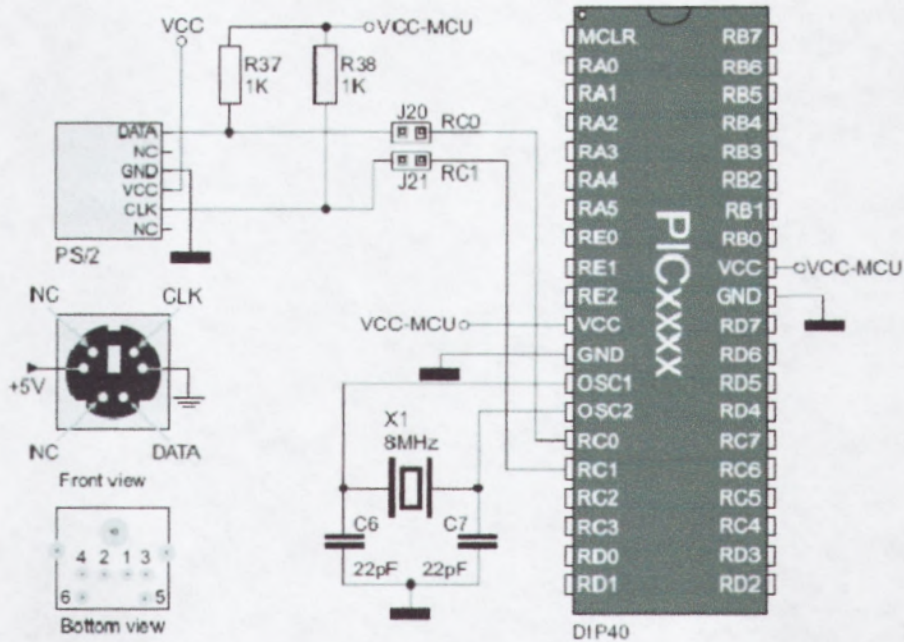


Figura 3.18: Esquemático de conexión del conector PS/2. Fuente [Mikroelektronika]



Figura 3.19: Conector PS/2 (J20 y J21 puentes colocados). Fuente [Mikroelektronika]

En la figura 3.20 se puede apreciar que no están colocados los puentes J20 y J21. No se debe conectar/desconectar los dispositivos de entrada al conector PS/2 mientras que la tarjeta EasyPic6 se encuentre encendido ya que puede dañar el microcontrolador permanentemente.



Figura 3.20: Conector PS/2 (J20 y J21 puentes no colocados). Fuente [Mikroelektronika]

La figura 3.21 muestra la conexión entre el sistema modular de entrenamiento EasyPic6 con el teclado.



Figura 3.21: EasyPic6 conectado al teclado. Fuente [Mikroelektronika]

3.9 Conector ICD

La figura 3.22 muestra el conector ICD (Depurador en circuito) permite que el microcontrolador se comunice con el depurador ICD (ver el etiquetado y posición en la figura 3.23) externo (ICD2 o ICD3) de la compañía Microchip. Los puentes J8 y J9 se colocan de la misma manera que al utilizar el programador *PICflash* con *mikroICD* diseñado por Mikroelektronika.



Figura 3.22: Conector ICD. Fuente [Mikroelektronika]

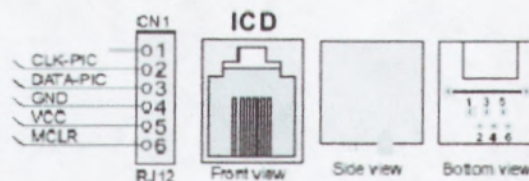


Figura 3.23: Etiquetas y posición de los pines en el conector ICD. Fuente [Mikroelektronika]

3.10 Módulo de comunicación USB

El conector USB habilita la conexión con los microcontroladores PIC, que disponen de un módulo de comunicación USB incorporado, a los dispositivos periféricos. Para establecer la comunicación USB, es necesario cambiar de posición del puente J12 de la posición I/O a la USB (ver figura 3.24). De este modo se conectan las líneas USB DATA (D+ y D-) a los pines del microcontrolador RC4 y RC5, mientras que el pin RC3/USB se conecta

a los capacitores C16 y C17. En caso de que no se utilice la comunicación USB, los puentes J12 deben quedarse en la posición I/O (ver figura 3.25).

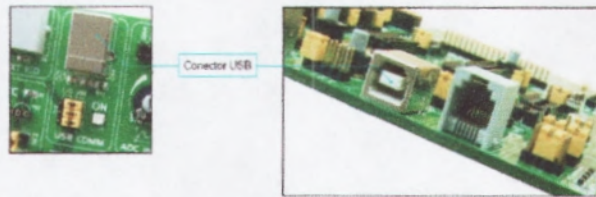


Figura 3.24: Habilitada la comunicación USB. Fuente [Mikroelektronika]



Figura 3.25: Deshabilitada la comunicación USB. Fuente [Mikroelektronika]

El estado de la comunicación USB (encendido/apagado) está indicado por el LED apropiado. En las Figuras 3.26a y 3.26b se muestran los esquemas de los microcontroladores utilizados con más frecuencia con el módulo USB integrado.

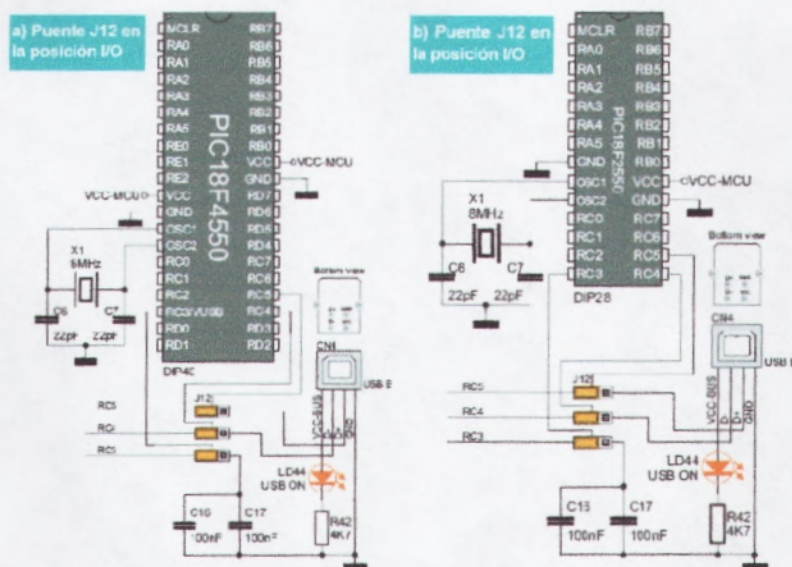


Figura 3.26: a) Esquema de conexión entre el USB y PIC18F4550, b) Esquema de conexión entre el USB y PIC18F2550. Fuente [Mikroelektronika]

3.11 Sensor de temperatura DS1820

La comunicación serie 1-wire® permite la transmisión de datos por medio de una sólo línea de comunicación, mientras que el mismo proceso está bajo el control del microcontrolador maestro. La ventaja de tal comunicación es que se utiliza sólo un pin de microcontrolador. Todos los dispositivos esclavos por defecto disponen de un código ID único, lo que permite que el dispositivo maestro identifique fácilmente los dispositivos que comparten la misma interfaz.



Asegúrese de que el semi-círculo en la placa coincida con la parte redonda del DS1820.

Figura 3.27: Sensor DS1820 colocado en el zócalo apropiado.

El DS1820 (ver figura 3.27) es un sensor de temperatura que utiliza la comunicación 1-wire, cuando no se requiera esta comunicación se coloca según la figura 3.28. El DS1820 es capaz de medir las temperaturas dentro del rango de -55 a 125°C y proporcionar la exactitud de medición de $\pm 0.5^{\circ}\text{C}$ para las temperaturas dentro del rango de -10 a 85°C . Para su funcionamiento el DS1820 requiere un voltaje de alimentación de 3 a 5.5V. El DS1820 tarda como máximo 750 ms en calcular la temperatura con una resolución de 9 bits.



Figura 3.28: Conector DS1820 (no se utiliza la comunicación 1-wire). Fuente [Mikroelektronika]

El sistema modular EasyPIC6 proporciona un zócalo separado para la comunicación del DS1820. Se puede utilizar uno de los pines, RA5 (ver figura 3.29) o RE2 (ver figura 3.30), para la comunicación con el microcontrolador, lo que depende de la posición del puente J11.

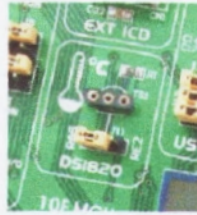


Figura 3.29: Puente J11 en la posición RA5 (comunicación 1-wire realizada por el pin RA5). Fuente [Mikroelektronika]



Figura 3.30: Puente J11 en la posición RA2 (comunicación 1-wire realizada por el pin RE2). Fuente [Mikroelektronika]

En la Figura 3.31 se muestra la comunicación 1-wire con el microcontrolador por el pin RA5.

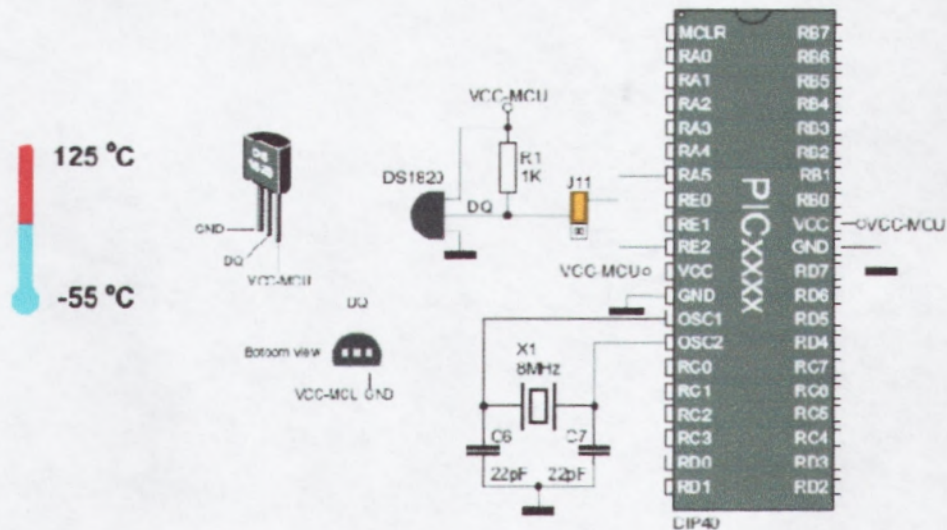


Figura 3.31: Esquemático de la conexión del sensor de temperatura DS1820. Fuente [Mikroelektronika]

3.12 Entradas de prueba del convertidor A/D 11.0.

Un convertidor A/D (analógico-digital) se utiliza con el propósito de convertir una señal analógica a un valor digital apropiado. El convertidor A/D es lineal, lo que quiere decir que el número convertido es linealmente dependiente del valor del voltaje de entrada. El convertidor A/D dentro del microcontrolador convierte un valor de voltaje analógico a un número de 10 bits. Los voltajes que varían de 0 a 5V DC se pueden suministrar al microcontrolador por las entradas de prueba del convertidor A/D.

En la figura 3.32 muestra el puente J15 utilizado para seleccionar uno de los pines para la conversión A/D: RA0 (ver figura 3.33), RA1, RA2, RA3 o RA4.



Figura 3.32: ADC (posición por defecto del puente). Fuente [Mikroelektronika]



Figura 3.33: Pin RA0 utilizado como una entrada para la conversión A/D.

Fuente [Mikroelektronika]

La resistencia R63 sirve de protección, empleada para limitar el flujo de corriente por el potenciómetro o por el pin del microcontrolador, en las figuras 3.34, 3.35 y 3.36 muestran los esquemas de las conexiones de PIC de 18, 28 y 40 pines respectivamente para la conversión A/D. El valor del voltaje analógico de entrada puede cambiar linealmente utilizando el potenciómetro P1.

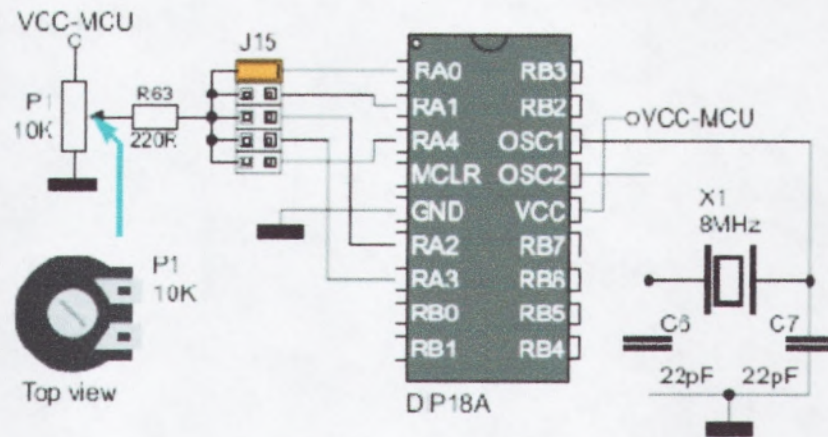


Figura 3.34: Esquema de conexión de las entradas de prueba del convertidor A/D al microcontrolador en el encapsulado DIP18A. Fuente [Mikroelektronika]

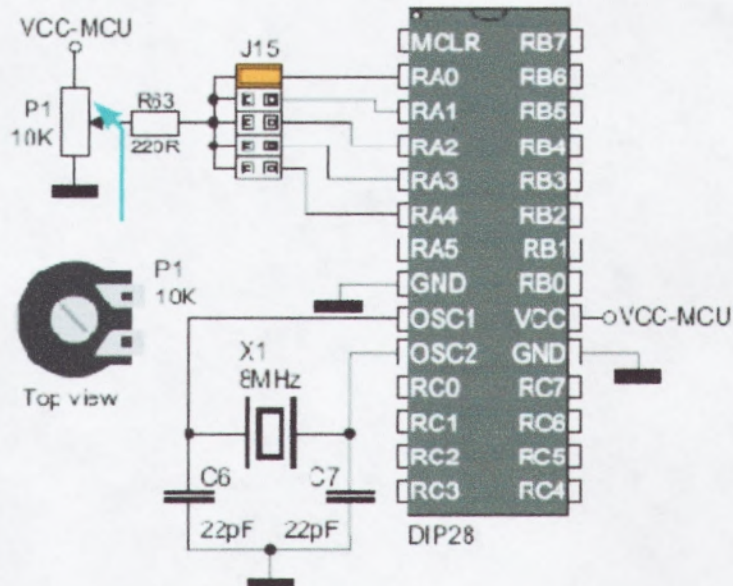


Figura 3.35: Esquema de conexión de las entradas de prueba del convertidor A/D al microcontrolador en el encapsulado DIP28. Fuente [Mikroelektronika]

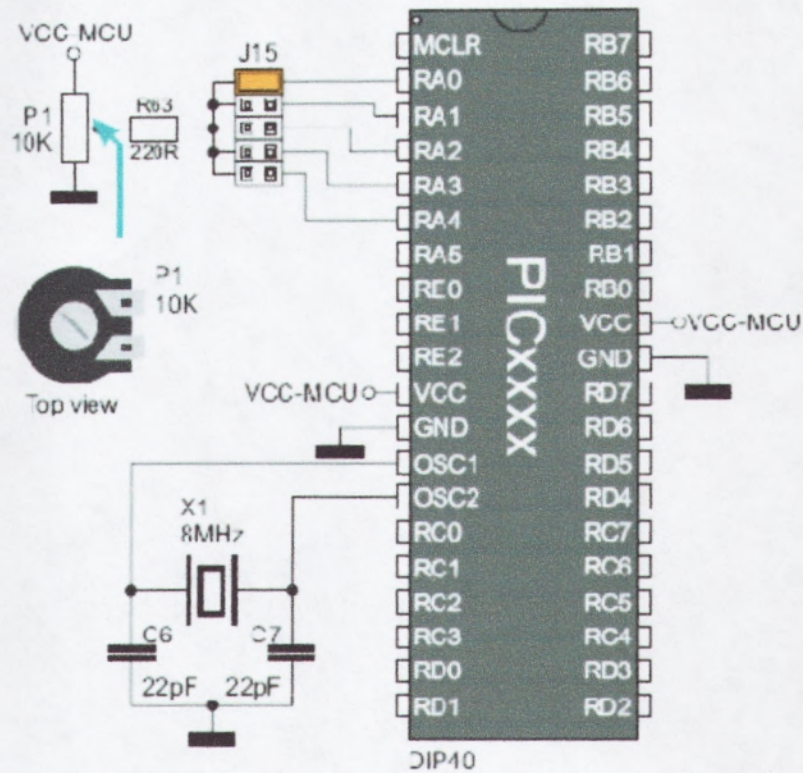


Figura 3.36: Esquema de conexión de las entradas de prueba del convertidor A/D al microcontrolador en el encapsulado DIP40. Fuente [Mikroelektronika]

3.13 Diodos LED

El diodo LED (Light-Emitting Diode), representa una fuente electrónica de luz de muy alta eficacia. Al conectar los LED's (ver figura 3.37) es necesario colocar una resistencia para limitar la corriente. El voltaje común de LED es aproximadamente 2.5V, mientras que la intensidad de corriente varía de 1 a 20mA dependiendo del tipo del diodo LED. El sistema de desarrollo EasyPIC6 utiliza los LED's con la corriente $I=1\text{mA}$).

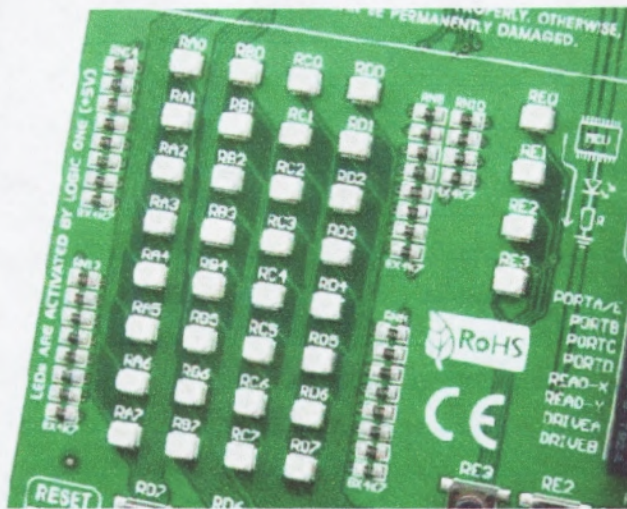


Figura 3.37: Diodos LED. Fuente [Mikroelektronika]

El sistema modular EasyPIC6 tiene 36 LED's que indican visualmente el estado lógico en cada pin de E/S del microcontrolador (ver ejemplo de conexión en la figura 3.38). Un diodo LED activo indica la presencia de un uno lógico (1) en el pin.



Figura 3.38: Indicador del cátodo y flujo de corriente por un LED. Fuente [Mikroelektronika]

Para habilitar que se muestre el estado de los pines, es necesario seleccionar el puerto apropiado PORTA/E, PORTB (ver ejemplo de conexión en la figura 3.39), PORTC o PORTD utilizando el interruptor DIP SW9.

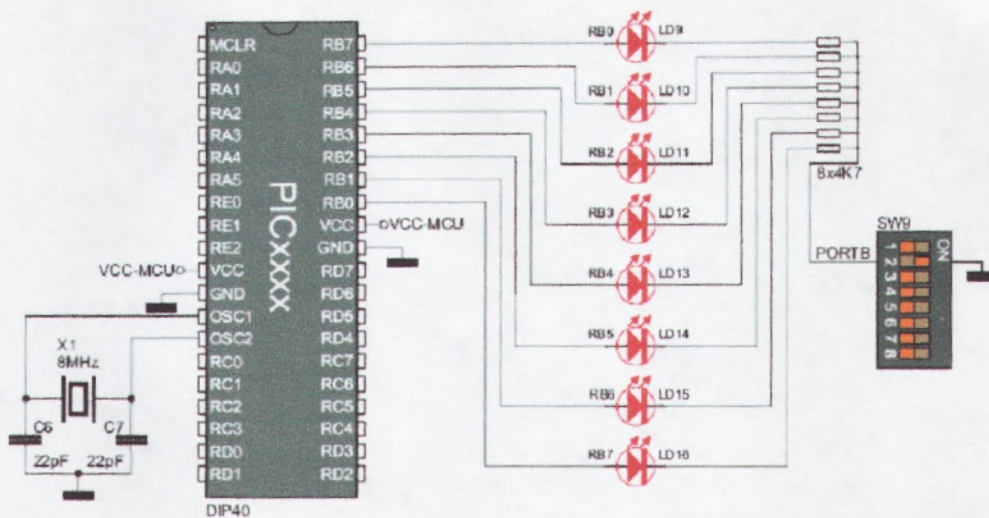


Figura 3.39: Esquema de conexión de los LED's y del puerto PORTB. Fuente [Mikroelektronika]

3.14 Botones de presión

El estado lógico de cada una de las entradas digitales del microcontrolador puede variar al utilizar los botones de presión (ver figura 3.40) también llamados pulsadores. El puente J17 se utiliza para determinar el estado lógico que será aplicado al pin deseado del microcontrolador al pulsar el botón apropiado. El propósito de la resistencia de protección es de limitar la máxima corriente, para impedir dañar el sistema de desarrollo y los módulos periféricos en caso de que ocurra un corto circuito. Si es necesario, los usuarios con más experiencia pueden, pueden poner en cortocircuito esta resistencia por medio del puente J24. Justamente junto a los botones de presión, se encuentra un botón RESET utilizado suministrar una señal de reset al pin MCLR por medio del programador incorporado.



Figura 3.40: Botones de presión. Fuente [Mikroelektronika]

Al pulsar cualquier botón de presión (RB0-RB7) cuando el puente J17 esté en la posición VCC-MCU, un 1 lógico (5V) será aplicado al pin apropiado del microcontrolador como se muestra en la Figura 3.41.

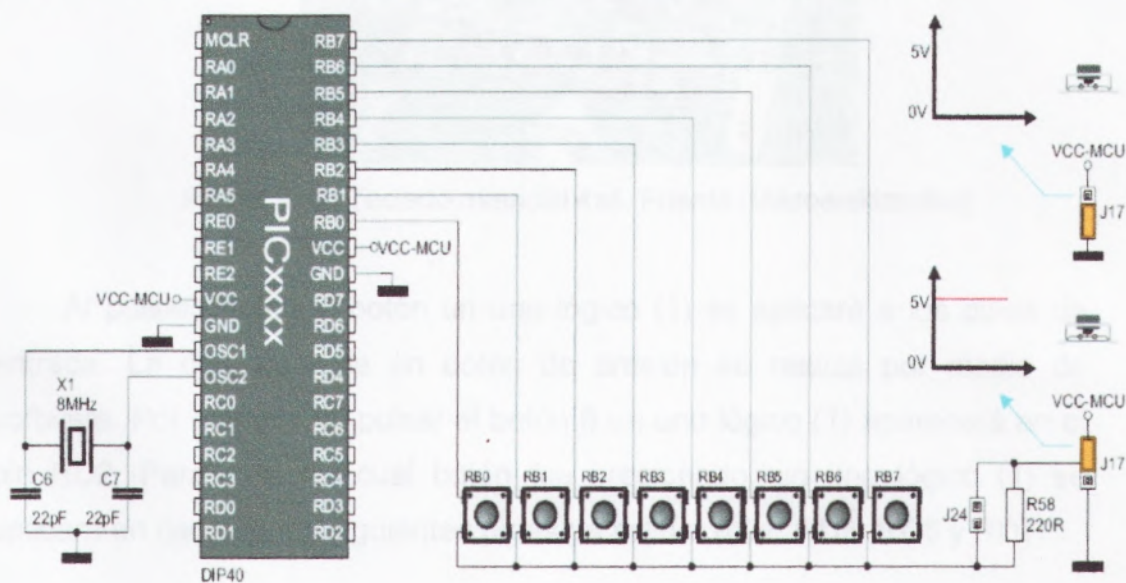


Figura 3.41: Esquema de conexión de los botones de presión al puerto PORTB. Fuente [Mikroelektronika]

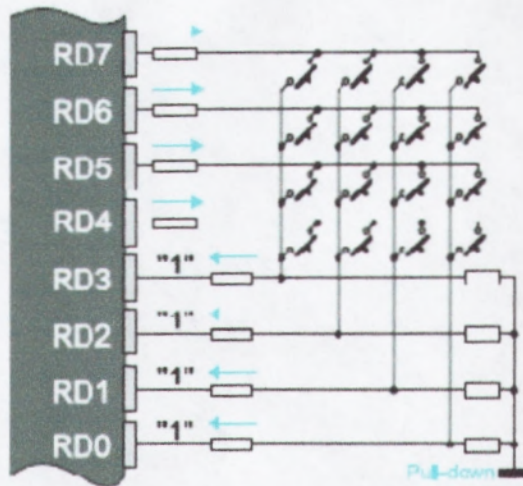


Figura 3.43: Rendimiento del teclado 4x4. Fuente [Mikroelektronika]

Los botones del teclado MENU están conectados de manera similar a los botones en el puerto PORTA. La única diferencia es la disposición de los botones. Los botones del teclado MENU están colocados de tal manera que proporcionen una fácil navegación por menús. La figura 3.45 muestra en forma conjunta la conexión y operación de los teclados matricial y menú al microcontrolador

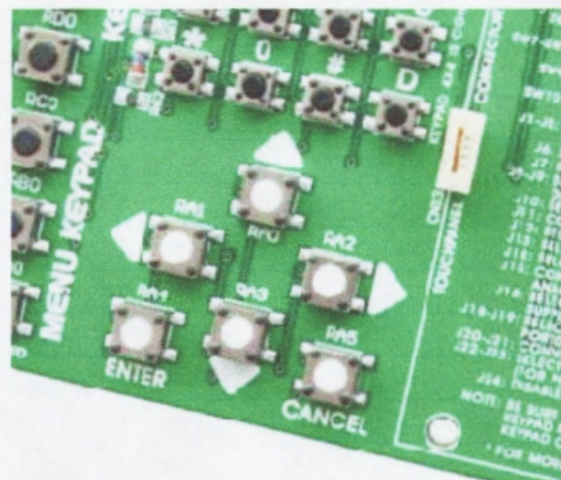


Figura 3.44: Teclado MENU. Fuente [Mikroelektronika]

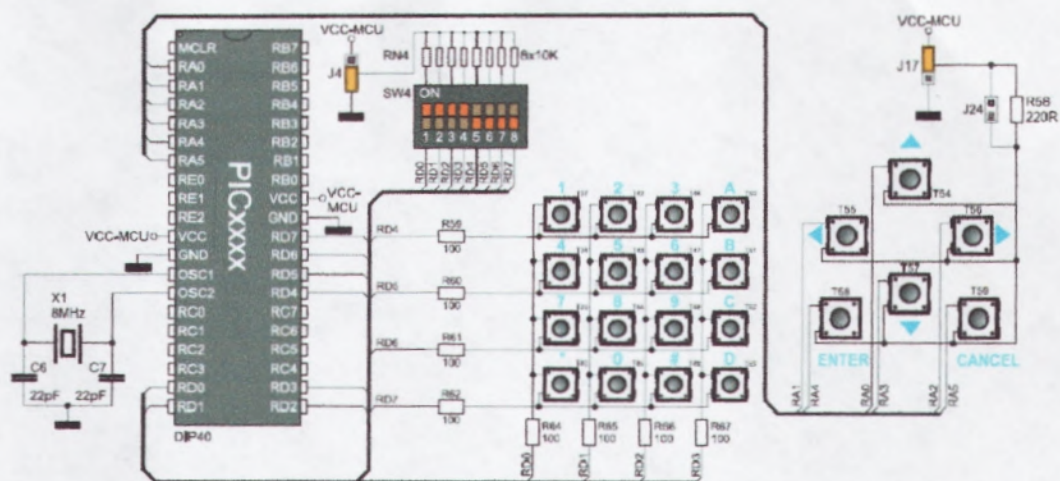


Figura 3.45: Esquema de conexión de los teclados (4x4 y MENU) al microcontrolador. Fuente [Mikroelektronika]

3.16 Visualizador LCD 2x16

El sistema modular EasyPic6 dispone de un conector (ver figura 3.46) incorporado en el que se coloca el visualizador LCD alfanumérico 2x16. Este conector está conectado al microcontrolador por el puerto PORTB. El potenciómetro P4 se utiliza para ajustar el contraste del visualizador mostrada por la figura 3.47.



Figura 3.46: Conector del visualizador LCD alfanumérico 2x16. Fuente [Mikroelektronika]

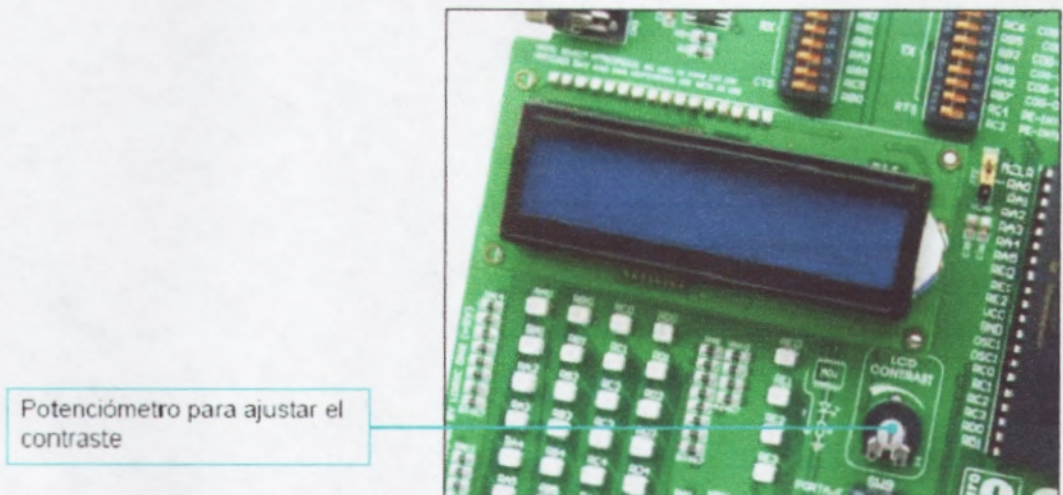


Figura 3.47: Visualizador LCD alfanumérico 2x16. Fuente [Mikroelektronika]

El interruptor con la etiqueta LCD - GLCD BACKLIGHT (ver figura 3.48) en el interruptor DIP SW6 se utiliza para encender/apagar luz de fondo del visualizador. La comunicación entre un visualizador LCD y el microcontrolador se establece utilizando el modo de 4 bits. Los dígitos alfanuméricos se visualizan en dos líneas de las que cada una contiene hasta 16 caracteres de 7x5 píxeles.

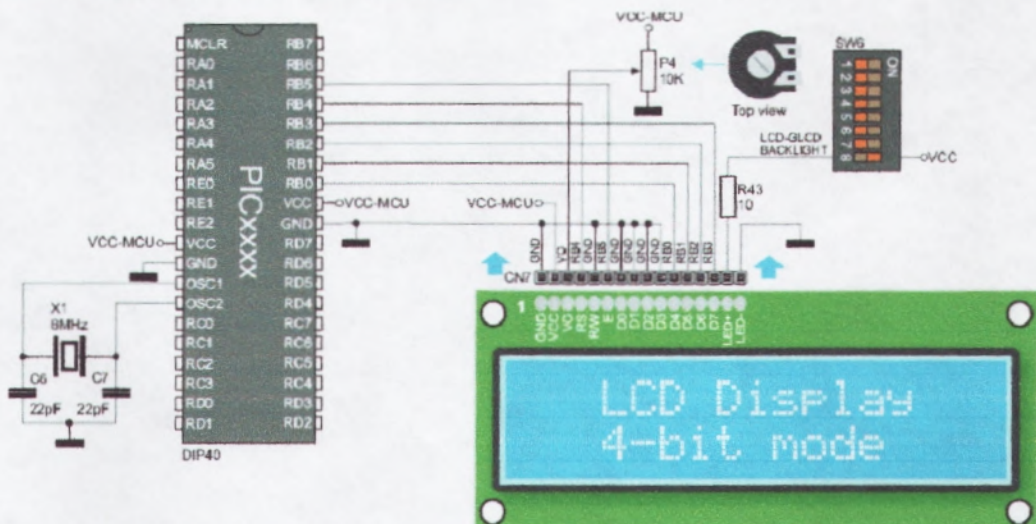


Figura 3.48: Esquema de conexión del visualizador LCD 2x16. Fuente [Mikroelektronika]

3.18 Visualizador gráfico LCD 128x64

El visualizador LCD gráfico 128x64 (GLCD, ver figura 3.51) proporciona un método avanzado de visualizar los mensajes gráficos. Está conectado al microcontrolador por los puertos PORTB y PORTD. El visualizador GLCD dispone de la resolución de pantalla de 128x64 píxeles que permite visualizar diagramas, tablas u otros contenidos gráficos. Puesto que el puerto PORTB es utilizado por el visualizador LCD alfanumérico 2x16 también, no es posible utilizar los dos visualizadores simultáneamente.

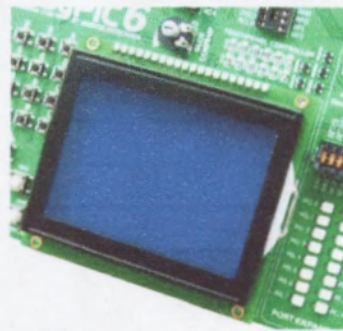


Figura 3.51: Visualizador GLCD. Fuente [Mikroelektronika]

La figura 3.52 muestra al potenciómetro P3 que se utiliza para ajustar el contraste del visualizador GLCD.

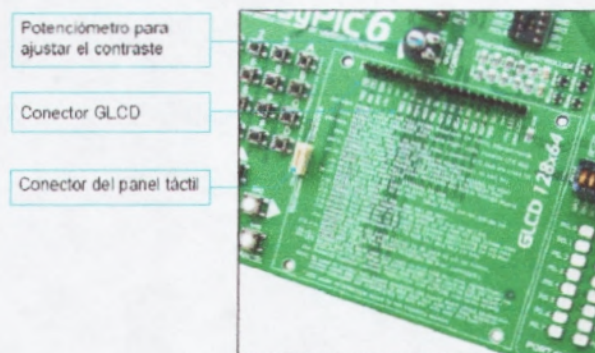


Figura 3.52: Conector GLCD. Fuente [Mikroelektronika]

En la figura 3.53 se muestra el esquemático de las conexiones del GLCD, donde el interruptor 8 en el interruptor DIP SW6 se utiliza para encender/apagar la luz de fondo del visualizador.

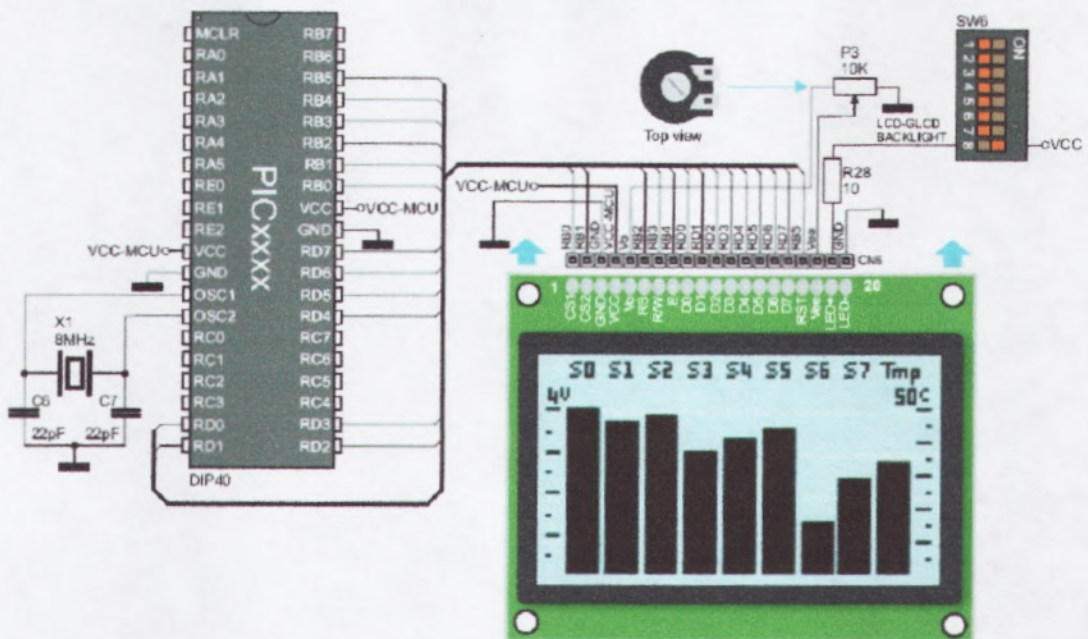


Figura 3.53: Esquema de conexión del visualizador GLCD. Fuente [Mikroelektronika]

3.19 Panel táctil

Un panel táctil es un panel no, autoadhesivo, transparente y toque sensitivo. Se coloca sobre el visualizador GLCD. El propósito principal de este panel es de registrar la presión en un punto específico del visualizador y enviar sus coordenadas en la forma del voltaje analógico al microcontrolador. La Figura 3.53 muestra cómo colocar un panel táctil sobre un visualizador GLCD. Asegúrese de que el cable plano esté a la izquierda del visualizador GLCD como se muestra en la Figura 3.54d.

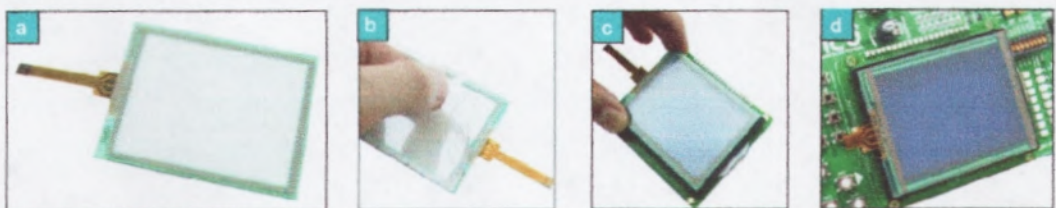


Figura 3.54: Colocación del panel táctil sobre un visualizador GLCD. a) panel táctil con protección, b) retiro del protector, c) forma de tomar al GLCD y d) panel táctil colocado al sistema modular EasyPic6. Fuente [Mikroelektronika].

Los interruptores 5, 6, 7 y 8 en el interruptor DIP SW9 se utilizan para conectar el panel táctil al microcontrolador, tal y como se muestra en la figura 3.55.

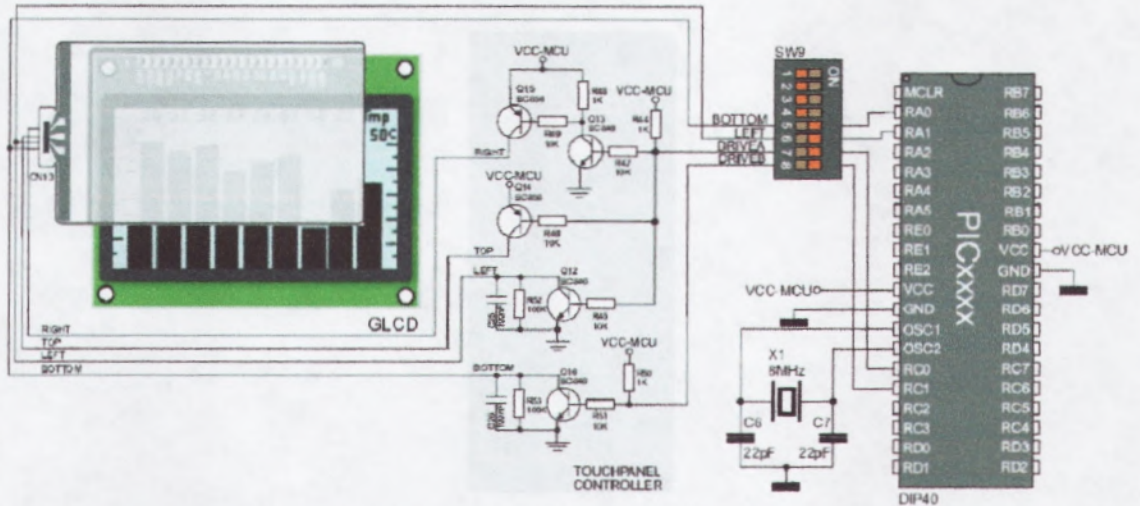


Figura 3.55: Esquema de conexión del panel táctil. Fuente [Mikroelektronika]

La Figura 3.56 muestra cómo conectar un panel táctil al microcontrolador. Acerque la punta del cable plano al conector C13 (ver figura 3.56a). Inserte el cable en el conector (ver figura 3.56b), y presiónelo lentamente de modo que la punta del cable encaje en el conector completamente según la Figura 3.56c. Luego inserte el visualizador GLCD en el conector apropiado como se muestra en la Figura 3.56d.

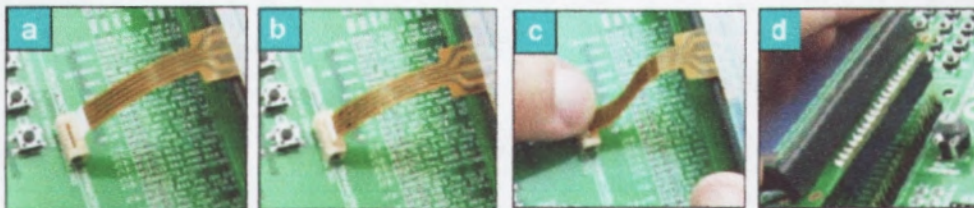


Figura 3.56: Colocación del panel táctil. Fuente [Mikroelektronika]

Los LED's y las resistencias pull-up/pull-down en los pines RA0 y RA1 del puerto PORTA tienen que estar apagados al utilizar un panel táctil.

3.20 Puertos de Entrada/Salida

A lo largo de la parte izquierda del sistema modular EasyPic6 están siete conectores de 10 pines que están conectados a los puertos de E/S del microcontrolador, mostrado en la figura 3.57. Unos están conectados directamente a los pines del microcontrolador, mientras que otros están conectados por medio de los puentes.

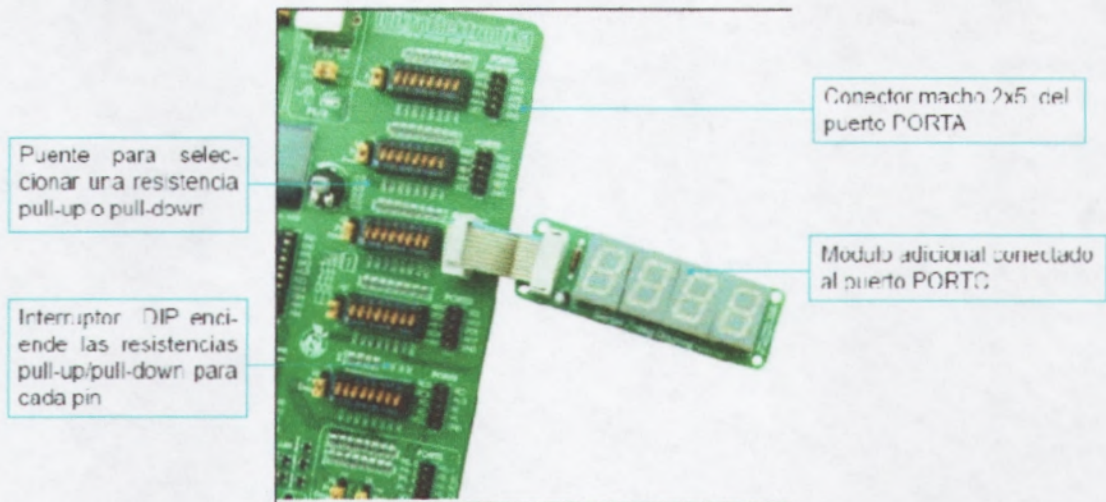


Figura 3.57: Puertos de E/S. Fuente [Mikroelektronika]

Los interruptores DIP SW1-SW5 permiten que cada pin de conector esté conectado a una resistencia pull up/pull down. Si los pines del puerto están conectados a una resistencia pull-up o pull-down depende de la posición de los puentes J1-J5, en la figura 3.58 y 3.59 muestra al puente J2 en posición pull-down y pull-up respectivamente.

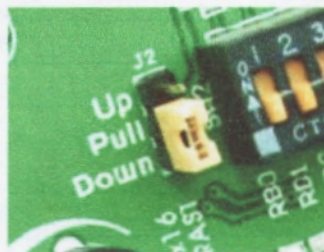


Figura 3.58: Puente J2 en la posición Pull-down. Fuente [Mikroelektronika]

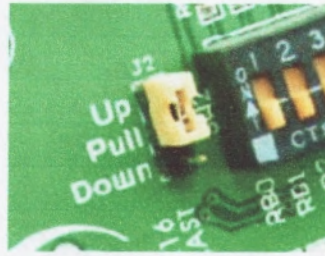


Figura 3.59: Puente J2 en la posición Pull-up. Fuente [Mikroelektronika]

En la figura 3.60, muestra las resistencias pull-up/pull-down (cuando están habilitadas) determinar el nivel lógico en todos los pines de entrada del microcontrolador cuando estén en el estado inactivo. El nivel lógico depende si el puente está en la posición pull-up o pull-down. El pin RB0 con la resistencia DIP SW2 pertinente, el puente J2 y el botón de presión RB0 con el puente J17 se utilizan con el propósito de explicar el funcionamiento de las resistencias pull-up/pull-down. El principio de su funcionamiento es idéntico para todos los pines del microcontrolador.

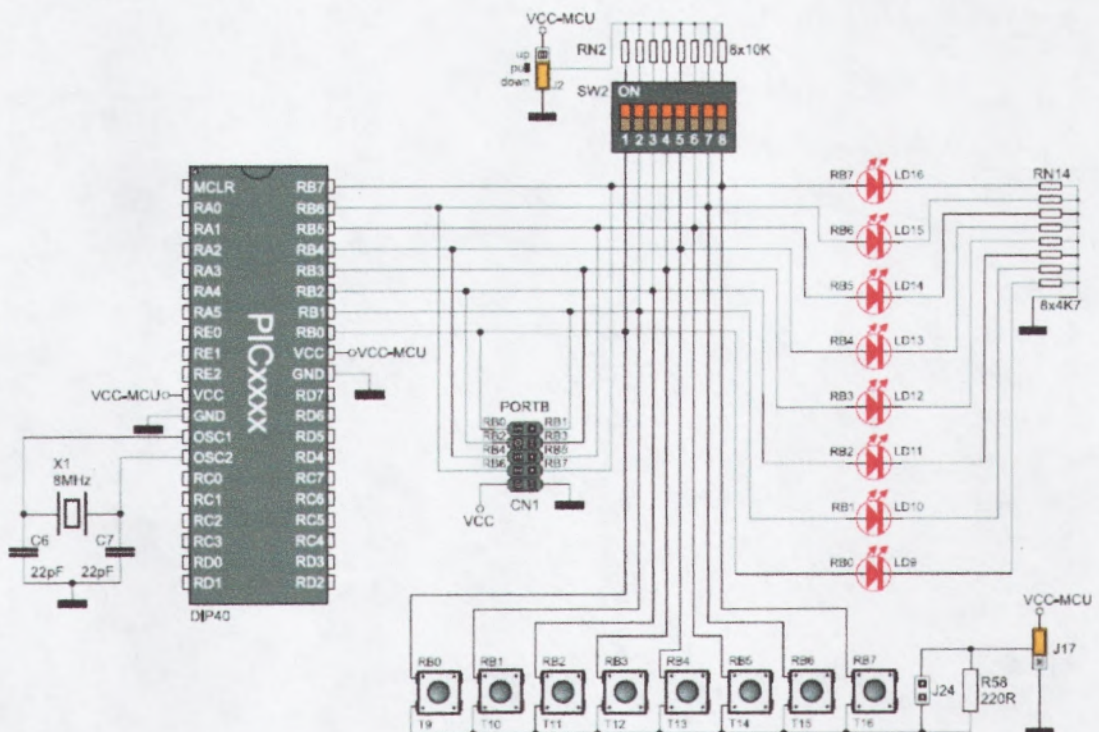


Figura 3.60: Esquema de conexión del puerto PORTB. Fuente [Mikroelektronika]

Para conectar los pines en el puerto PORTB a las resistencias pull-down, es necesario poner el puente J2 en la posición Down, tal y como se muestra en la figura 3.61. Esto permite que se lleve un cero lógico (0) a cualquier pin del puerto PORTB en el estado inactivo por medio del puente J12 y de la red de resistencias de 8x10K. Para llevar esta señal al pin RB0, es necesario poner el interruptor RB0 en el interruptor DIP SW2 en la posición ON. Por consiguiente, cada vez que se presiona el botón RB0 un uno lógico (1) aparecerá en el pin RB0, con tal de que el puente J17 esté colocado en la posición VCC-MCU.

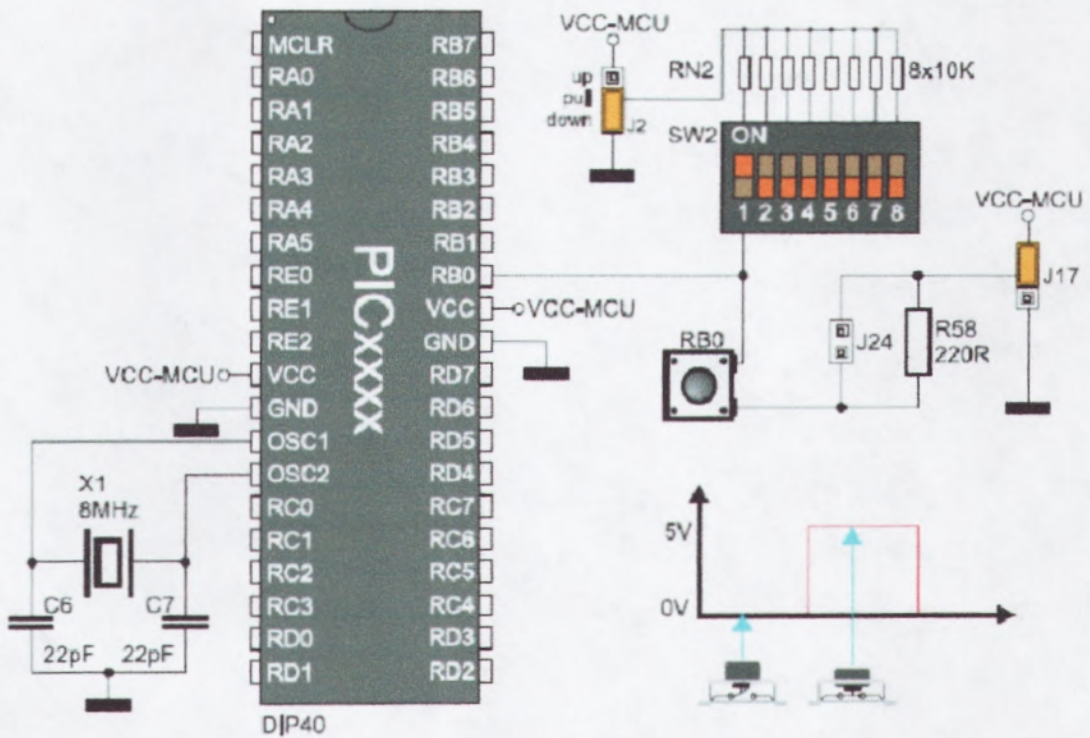


Figura 3.61: Puente J2 en la posición pull-down y el puente J17 en la posición pull-up. Fuente [Mikroelektronika]

Para conectar los pines en el puerto PORTB a las resistencias pull-up, y para llevar un cero lógico (0) a los pines de entrada en el puerto, es necesario poner el puente J2 en la posición Up (5V) y el puente J17 en la posición GND (0V) mostrada por la figura 3.62. Además, el pin RB0 en el interruptor DIP se debe colocar en la posición ON. Esto permite llevar un uno

lógico (1) a todos los pines de entrada del puerto PORTB en el estado inactivo por medio de la resistencia de 10k. El voltaje se lleva al pin RB0 por medio de la resistencia de 10k y el interruptor RB0. Por consiguiente, cada vez que se presiona el botón RB0, un cero lógico (0) aparecerá en el pin RB0.

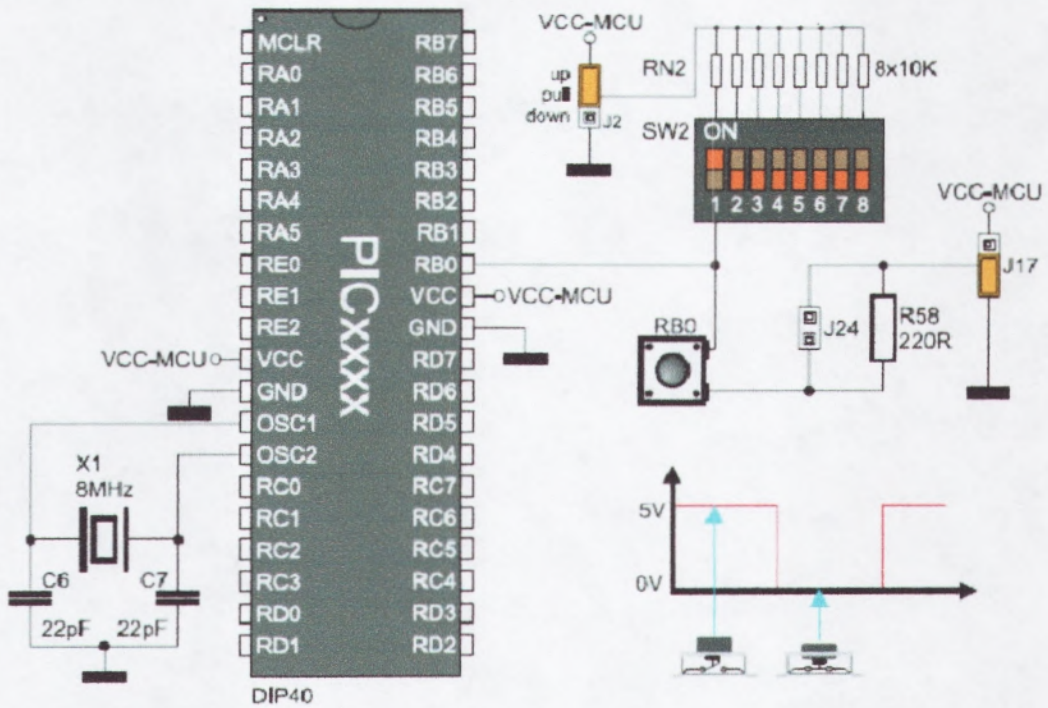


Figura 3.62: Puente J2 en la posición pull-up y puente J17 en la posición pull-down. Fuente [Mikroelektronika]

En caso de que los puentes J2 y J17 (ver figura 3.63) tengan el mismo estado lógico, al presionar cualquier botón de presión no cambia de estado lógico de los pines de entrada.

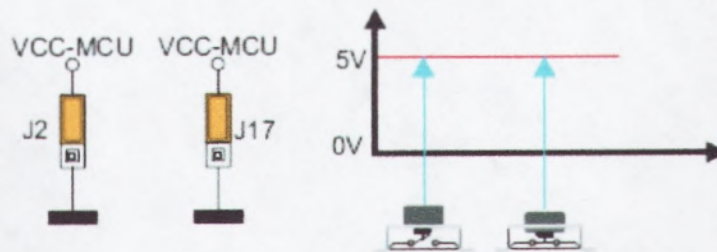


Figura 3.63: Puentes J2 y J17 en las mismas posiciones. Fuente [Mikroelektronika]

3.21 Extensor de puertos

En la figura 3.64 se muestra las líneas de comunicación SPI y el circuito MCP23S1 proporcionan el sistema de desarrollo EasyPIC6 con recursos de incrementar en dos el número de los puertos de E/S.

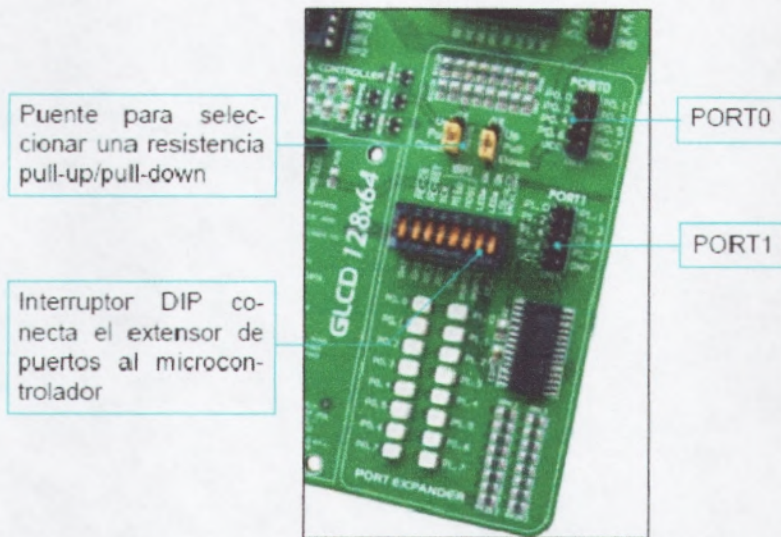


Figura 3.64: Extensor de puertos. Fuente [Mikroelektronika]

Si el extensor de puertos está conectado al microcontrolador por medio del interruptor SW6 (ver figura 3.65), los siguientes pines RA2, RA3, RC3, RC4 y RC5 se utilizarán para la comunicación SPI, por lo que no se pueden utilizar como los pines de E/S.

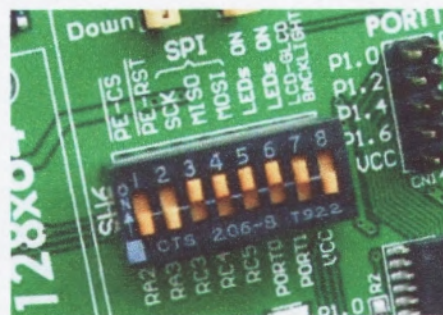


Figura 3.65: Posición del interruptor DIP SW6 cuando el extensor de puertos está habilitado. Fuente [Mikroelektronika]

Los interruptores INTA e INTB en el interruptor DIP SW10 habilitan una interrupción como se muestra en la figura 3.66. MCP23S17 habilita la expansión paralela de 16 bits y se puede configurar para funcionar en el modo de 16 o 8 bits.

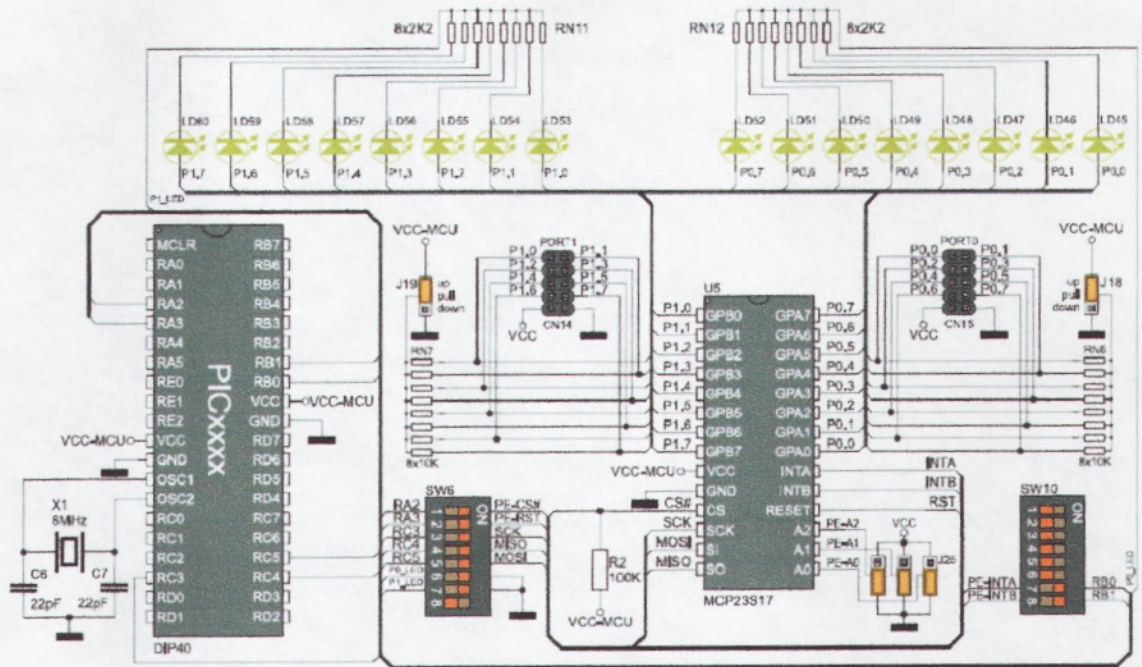


Figura 3.66: Esquema del extensor de puertos. Fuente [Mikroelektronika]

CAPÍTULO 4

DESARROLLO EXPERIMENTAL DE LAS PRÁCTICAS DE LABORATORIO

En el desarrollo del presente capítulo se expondrán 6 prácticas de microcontroladores que servirán para la enseñanza práctica en el Laboratorio de Electrónica en relación a los microcontroladores PIC (MicroChip). Estas prácticas están desarrolladas de manera que los alumnos logren una mejor comprensión para futuras investigaciones que permita un mejor desenvolvimiento en el ámbito laboral. A continuación se desarrollarán las prácticas de Laboratorio en la asignatura de Microcontroladores.

Prácticas del Laboratorio de Microcontroladores

4.1 Práctica # 1: Escribir en el puerto B el valor 55h

Objetivo: Utilizar los puertos del microcontrolador PIC como entradas y salidas, como interface digital al mundo exterior (el mismo objetivo se utiliza para las prácticas 1 al 7).

Entradas:

Número 55h

Salidas:

Código binario del número 55h (01010101) en el PortB del micro controlador.

Procesos:

- ✓ Escribir el valor 55h en el PortB.
- ✓ Retardo de 1 segundo.

Diagrama de flujo:

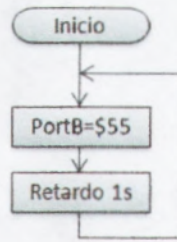


Figura 4.1: Diagrama ASM del desarrollo experimental 1. Fuente: Propia del Autor.

Programación en MikroBasic:

```
program practica1_1
```

```
trisb=0 'Habilita el puerto B como Salida
```

```
main: 'Parte principal del programa
```

```
portb=$55 'Escribe en el puerto B el valor 55h=01010101
```

```
delay_ms(1000) 'Retardo de 1 segundo
```

```
end. 'Fin del programa
```

Simulación

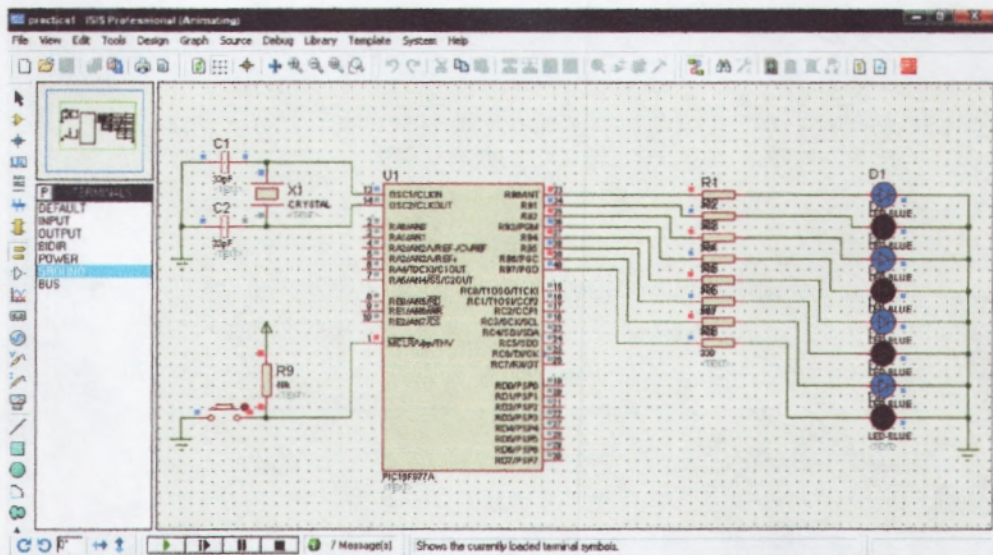


Figura 4.2: Simulación del programa de la primera práctica²².

²² Simulación del circuito de la práctica 1 en ISIS de Proteus. ISIS Professional v7.6 SP2. © Labcenter Electronics 1989-2010

4.2 Práctica # 2: Escribir en el puerto B el valor AAh

Entradas:

Número AAh

Salidas:

Código binario del número AAh (10101010) en el PortB del micro controlador.

Procesos:

- ✓ Escribir el valor AAh en el PortB.
- ✓ Retardo de 1 segundo.

Diagrama de flujo:

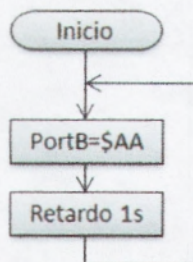


Figura 4.3: Diagrama ASM de práctica 2. Fuente: Propia del Autor.

Programación en MikroBasic:

```
program practica1_2 trisb=0 main:  
portb=$AA 'Escribe en el puerto B el valor AAh=10101010  
delay_ms(1000)  
end.
```

Simulación

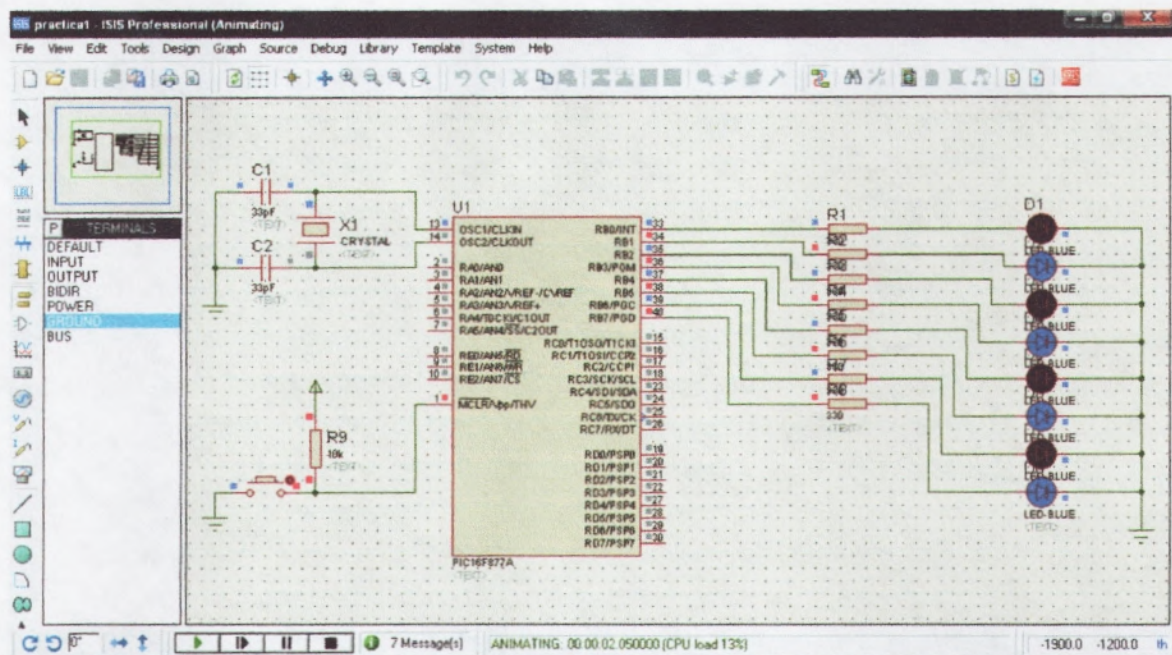


Figura 4.4: Simulación del programa de la segunda práctica²³.

4.3 Práctica # 3: Escribir en el puerto C el valor F0

Entradas:

Número F0h

Salidas:

Código binario del número F0h (11110000) en el PortB del micro controlador.

Procesos:

- ✓ Escribir el valor F0h en el PortB.
- ✓ Retardo de 1 segundo.

Diagrama de flujo:

²³ Simulación del circuito de la práctica 2 en ISIS de Proteus. ISIS Professional v7.6 SP2. © Labcenter Electronics 1989-2010

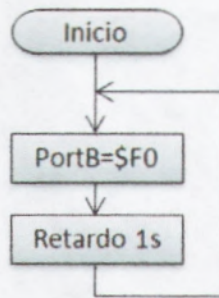


Figura 4.5: Diagrama ASM de práctica 3. Fuente: Propia del Autor.

Programación en MikroBasic:

```

program practica1_3 trisc=0 'habilita el puerto C como salida main:
portc=$F0 'Escribe en el puerto C el valor F0h=11110000
delay_ms(1000)
end.
  
```

Simulación

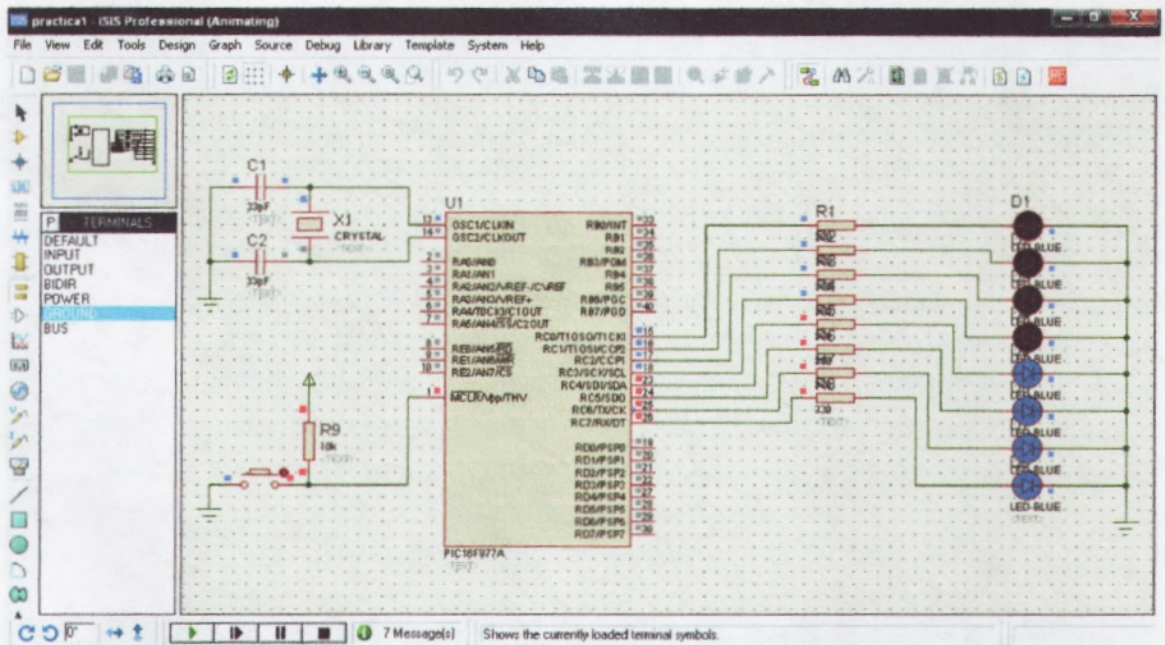


Figura 4.6: Simulación del programa de la segunda práctica²⁴.

²⁴ Simulación del circuito de la práctica 3 en ISIS de Proteus. ISIS Professional v7.6 SP2. © Labcenter Electronics 1989-2010

4.4 Práctica # 4: Leer en el puerto C y escribir en el puerto B.

Entradas:

Valores de entrada en el PortC.

Salidas:

Bits que ingresan por el PortC salen en las patitas del PortB.

Procesos:

- ✓ Definir variable para almacenar los datos ingresados por el PortC.
- ✓ Habilitar el PortC como entrada y el PortB como salida.
- ✓ Leer bits con el PortC.
- ✓ Guardar los bits que ingresan por el PortC en la variable.
- ✓ Escribir el valor de la variable en el PortB.

Diagrama de flujo:

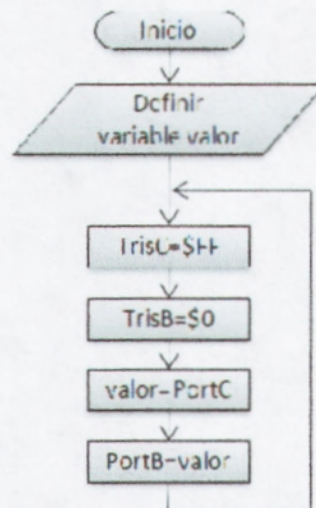


Figura 4.7: Diagrama ASM de práctica 4. Fuente: Propia del Autor.

Programación en MikroBasic:

```
program practica1_4
```

```
dim valor as byte 'Dimensiona una variable (valor) tipo byte
```

```
main: lazo:
```

trisc=\$FF 'Habilita el puerto C como Entrada

trisb=\$0 'Habilita el puerto B como Salida

valor=portc

portb=valor

goto lazo

end.

Simulación

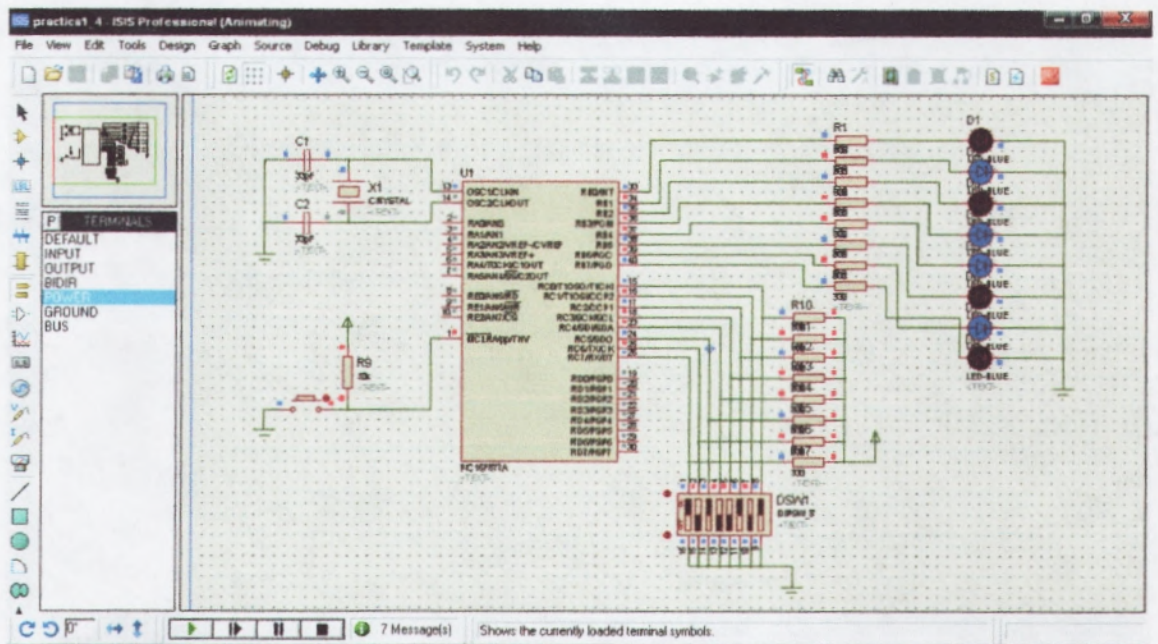


Figura 4.8: Simulación del programa de la cuarta práctica ²⁵.

4.5 Práctica # 5: Escribir en un display de 7 segmentos sin utilizar decodificador, un valor ascendente entre 0 y F.

Entradas:

Valores de 0 a F

Salidas:

Código binario entre 0 y F en el PortB del micro controlador.

²⁵ Simulación del circuito de la práctica 4 en ISIS de Proteus. ISIS Professional v7.6 SP2. © Labcenter Electronics 1989-2010

Procesos:

- ✓ Definir código binario para cada valor.
- ✓ Escribir el código binario establecido para cada valor en el PortB
- ✓ Retardo 2 segundos.
- ✓ Repetir los tres pasos anteriores hasta llegar a la F

Diagrama de flujo:

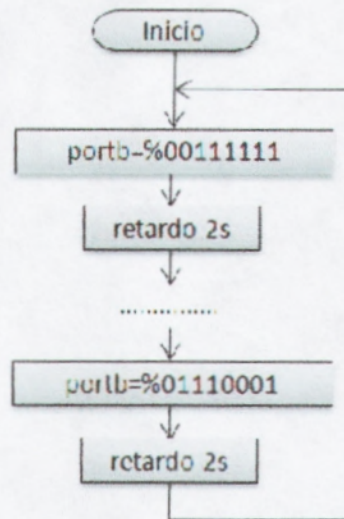


Figura 4.9: Diagrama ASM de práctica 5. Fuente: Propia del Autor.

Programación en MikroBasic:

```
program practica1_5
```

```
trisb=$0
```

```
'a=portb.0, b=portb.1, c=portb.2, d=portb.3, e=portb.4, f=portb.5,
```

```
g=portb.6
```

```
'el display debe ser cátodo común
```

```
main:
```

```
lazo:
```

```
portb=%00111111 formato para trabajar en binario
```

```
delay_ms(2000)
```

```
portb=%00000110
```

```
delay_ms(2000)
```

```
portb=%01011011
delay_ms(2000)
portb=%01001111
delay_ms(2000)
portb=%01100110
delay_ms(2000)
portb=%01101101
delay_ms(2000)
portb=%01111101
delay_ms(2000)
portb=%00000111
delay_ms(2000)
portb=%01111111
delay_ms(2000)
portb=%01100111
delay_ms(2000)
portb=%01110111
delay_ms(2000)
portb=%01111100 delay_ms(2000)
portb=%01011000
delay_ms(2000)
portb=%01011110
delay_ms(2000)
portb=%01111001
delay_ms(2000)
portb=%01110001
delay_ms(2000)
goto lazo
end.
```


Simulación

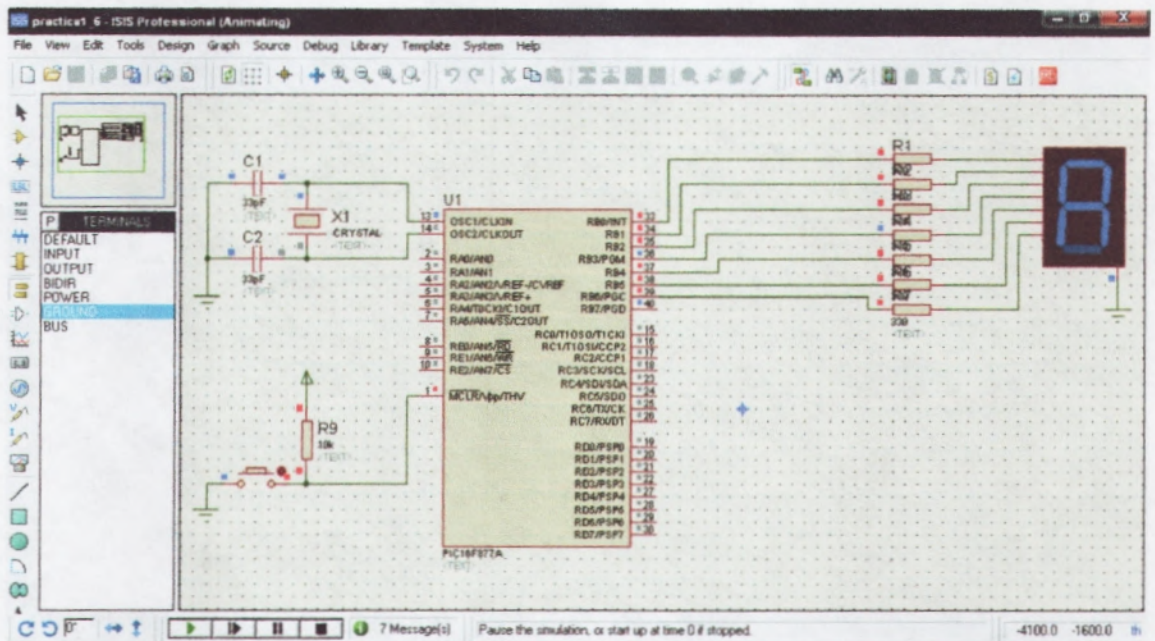


Figura 4.10: Simulación del programa de la quinta práctica²⁶.

4.6 Práctica # 6: Mediante un display alfanumérico mostrar en orden descendentes las letras del alfabeto de la Z a la A

Entradas:

Valores alfanuméricos de la Z a la A

Salidas:

Código binario desde la Z a la A en el PortB del micro controlador.

Procesos:

- ✓ Definir código binario para cada valor.
- ✓ Escribir los 8 primeros bits (a-h) del código binario establecido para cada valor en el PortB.

²⁶ Simulación del circuito de la práctica 5 en ISIS de Proteus. ISIS Professional v7.6 SP2. © Labcenter Electronics 1989-2010

- ✓ Escribir los últimos bits (i-n) del código binario establecido para cada valor en el PortC.
- ✓ Retardo 1 segundo
- ✓ Repetir los cuatro pasos anteriores hasta llegar a la A

Diagrama de flujo:

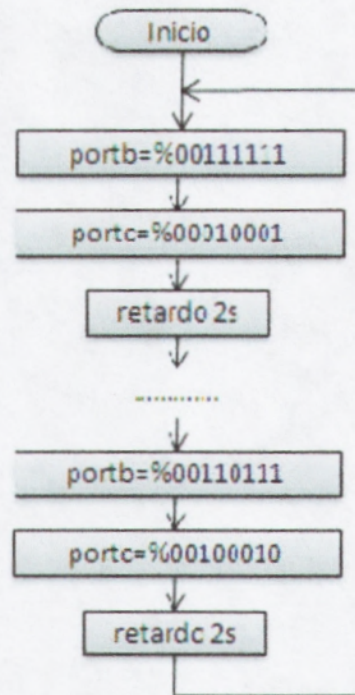


Figura 4.11: Diagrama ASM de práctica 6. Fuente: Propia del Autor.

Tabla de valores para encender los leds del display

	o	n	m	l	k	j	i	h	g	f	e	d	c	b	a		
Z	0	0	0	1	0	0	0	1	0	0	0	0	1	0	0	1	
Y	0	0	0	0	1	0	0	1	0	1	0	0	0	0	0	0	
X	0	0	0	1	0	1	0	1	0	1	0	0	0	0	0	0	
W	0	0	0	1	0	1	0	0	0	0	1	1	0	1	1	0	
V	0	0	0	1	0	0	0	1	0	0	1	1	0	0	0	0	
U	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	0
T	0	0	0	0	1	0	0	0	1	0	0	0	0	0	0	0	1
S	0	0	1	0	0	0	1	0	0	0	1	0	1	1	0	1	
R	0	0	1	0	0	1	1	0	0	0	1	1	0	0	1	1	
Q	0	0	1	0	0	0	1	0	0	0	1	0	0	1	1	1	
P	0	0	1	0	0	0	1	0	0	0	1	1	0	0	1	1	
O	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	
N	0	0	0	0	0	1	0	0	0	1	1	1	0	1	1	0	
M	0	0	0	0	0	0	0	1	0	1	1	1	0	1	1	0	
L	0	0	0	0	0	0	0	0	0	0	1	1	1	1	0	0	0
K	0	0	1	0	0	1	0	1	0	0	1	1	0	0	0	0	0
J	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	0
I	0	0	0	0	1	0	0	0	1	0	0	0	1	0	0	0	1
H	0	0	1	0	0	0	1	0	0	0	1	1	0	1	1	0	1
G	0	0	0	0	0	0	1	0	0	0	1	1	1	1	0	1	1
F	0	0	1	0	0	0	1	0	0	0	1	1	0	0	0	0	1
E	0	0	1	0	0	0	1	0	0	0	1	1	1	0	0	0	1
D	0	0	1	0	0	0	1	0	0	0	0	1	1	1	1	1	0
C	0	0	0	0	0	0	0	0	0	0	1	1	1	0	0	0	1
B	0	0	1	0	0	0	1	0	0	0	1	1	1	1	1	0	0
A	0	0	0	0	0	0	0	0	0	0	1	1	0	1	1	1	1

Tabla 4.1: Configuración de leds del display mostrado en la figura 4.12.

Fuente: Propia del Autor.

Los leds del display mostrado por la figura 4.12, son distribuidos de la siguiente manera:

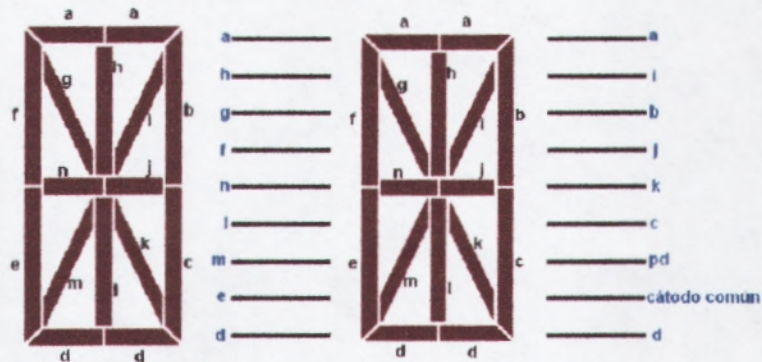


Figura 4.12: Distribución de los pines del display CD 8103 A/B cátodo común

Fuente: Propia del Autor.

Los pines del PIC se distribuyen de la siguiente manera:

- ✓ Los valores correspondientes a los leds desde la a hasta la h, serán controlados por los pines del puerto B del PIC, así:

PortB.7	PortB.6	PortB.5	PortB.4	PortB.3	PortB.2	PortB.1	PortB.0
h	g	f	e	d	c	b	a

Tabla 4.2: Configuración del puerto B del PIC. Fuente: Propia del Autor.

- ✓ Los valores correspondientes a los leds desde la i hasta la m, serán controlados por los pines del puerto C del PIC, así:

PortC.7	PortC.6	PortC.5	PortC.4	PortC.3	PortC.2	PortC.1	PortC.0
o	o	r	m	l	k	j	i

Tabla 4.3: Configuración del puerto B del PIC. Fuente: Propia del Autor.

Nótese que el puerto C sólo debería controlar los últimos 6 leds del display y que los puertos PortC.6 y PortC.7 estarían inutilizados, es por esta razón que estos 2 pines estarán constantemente en 0. (Véase Tabla 4.3).

Programación en MikroBasic:

```

program practica1_6
trisa=$0
trisc=$0
main:
lazo:
portb=%00001001
portc=%00010001
delay_ms(1000)
portb=%01000000
portc=%00001001
delay_ms(1000)
portb=%01000000
    
```

```
portc=%00010101
delay_ms(1000)
portb=%00110110
portc=%00010100
delay_ms(1000)
portb=%00110000
portc=%00010001
delay_ms(1000)
portb=%00111110
portc=%00000000
delay_ms(1000)
portb=%10000001
portc=%00001000
delay_ms(1000)
portb=%00101101
portc=%00100010
delay_ms(1000)
portb=%00110011
portc=%00100110
delay_ms(1000)
portb=%00100111
portc=%00100010
delay_ms(1000)
portb=%00110011
portc=%00100010
delay_ms(1000)
portb=%00111111
portc=%00000000
delay_ms(1000)
portb=%01110110
portc=%00000100
```

```
delay_ms(1000)
portb=%01110110
portc=%00000001
delay_ms(1000)
portb=%00111000
portc=%00000000
delay_ms(1000)
portb=%00110000
portc=%00100101
delay_ms(1000)
portb=%00011110
portc=%00000000
delay_ms(1000)
portb=%10001001
portc=%00001000
delay_ms(1000)
portb=%00110110
portc=%00100010
delay_ms(1000)
portb=%00111101
portc=%00000010
delay_ms(1000)
portb=%00110001
portc=%00100010
delay_ms(1000)
portb=%00111001
portc=%00100010
delay_ms(1000)
portb=%00011110
portc=%00100010
delay_ms(1000)
```

```

portb=%00111001
portc=%00000000
delay_ms(1000)
portb=%00111100
portc=%00100010
delay_ms(1000)
portb=%00110111
portc=%00100010
delay_ms(1000)
goto lazo
end.

```

Simulación

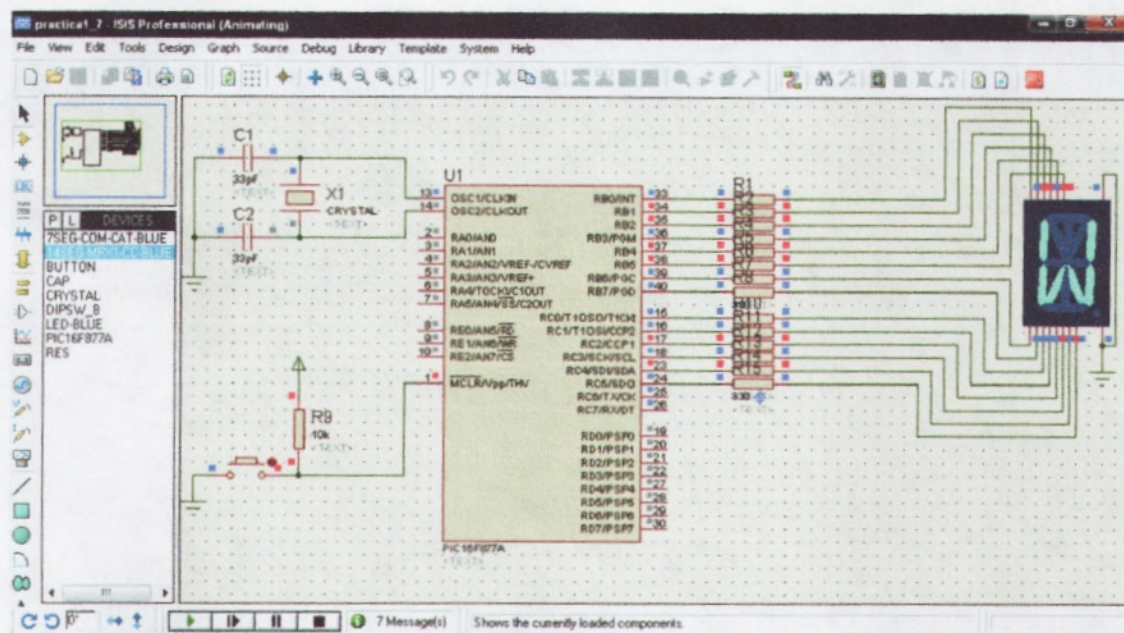


Figura 4.13: Simulación del programa de la sexta práctica²⁷.

²⁷ Simulación del circuito de la práctica 6 en ISIS de Proteus. ISIS Professional v7.6 SP2. © Labcenter Electronics 1989-2010

4.7 Práctica # 7: Lógica booleana con micro controlador

Entradas:

Valores en el PortC

Salidas:

Código binario en el PortB del micro controlador.

Procesos:

- ✓ Definir código binario para cada valor.
- ✓ Escribir en el PortB.0 PortC.0 and PortC.1
- ✓ Escribir en el PortB.1 PortC.0 or PortC.1
- ✓ Escribir en el PortB.2 PortC.0 xor PortC.1

Diagrama de flujo:

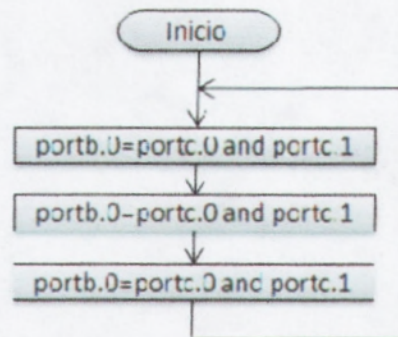


Figura 4.14: Diagrama ASM de práctica 7. Fuente: Propia del Autor.

Programación en MikroBasic:

```
program practica1__7
trisc=$FF 'Puerto C como entradas
trib=$0 'Puerto B como salidas
main:
portb.0=portc.0 and portc.1
portb.1=portc.2 or portc.3
portb.2=portc.4 xor portc.5
```


goto main
end.

Simulación

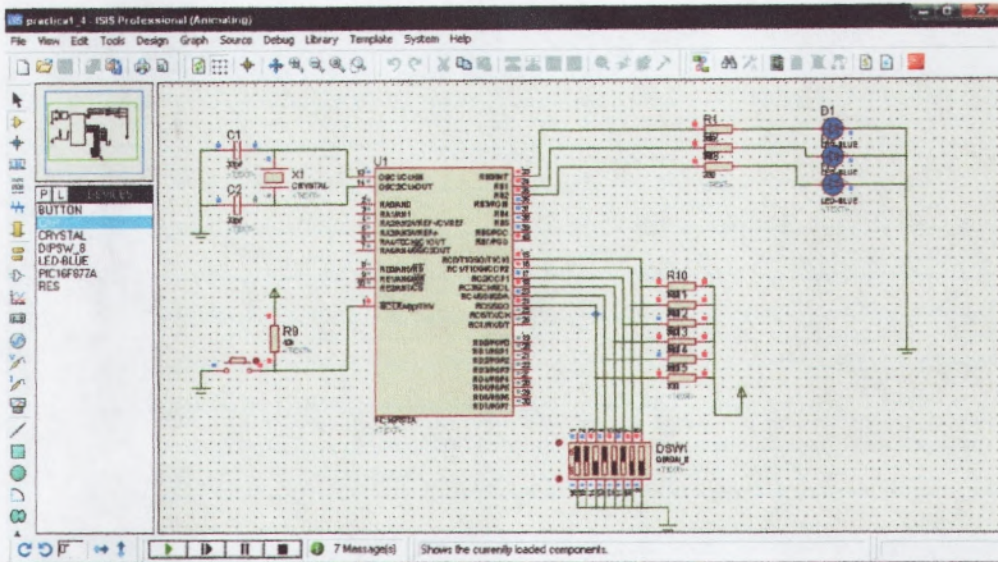


Figura 4.15: Simulación del programa de la séptima práctica²⁸.

4.8 Práctica # 8: Utilización de declaraciones con el Microcontrolador PIC

Objetivo: utilización de declaraciones con el microcontrolador PIC.

Breve Fundamentación Teórica de la sentencias

Sentencias

Las sentencias especifican y controlan el flujo de ejecución del programa. En ausencia de las sentencias de salto y de selección, las sentencias se ejecutan en el orden de su aparición en el código de programa.

²⁸ Simulación del circuito de la práctica 7 en ISIS de Proteus. ISIS Professional v7.6 SP2. © Labcenter Electronics 1989-2010

Sentencias de asignación

Las sentencias de asignación evalúan la expresión y le asigna el valor de la expresión a una variable aplicando las reglas de la conversión implícita.

Variable = expression

El especificador **variable** puede ser cualquier variable declarada, mientras que el especificador **expression** representa la expresión cuyo valor corresponde a la variable dada.

Sentencias condicionales

Las sentencias condicionales o las sentencias de selección pueden decidir entre varios cursos de acción distintos en función de ciertos valores.

Sentencias *if*

La sentencia ***if*** es una sentencia condicional. La sintaxis de la sentencia ***if*** es la siguiente:

```
if expression then
    Statement1
    [else
    Statement2]
End if
```

Si la expresión se evalúa como cierta, ***statement1*** se ejecuta. Si ***expression*** se evalúa como falso, ***statement2*** se ejecuta. La rama ***else*** compuesta de la palabra clave ***else*** y la sentencia ***statement2*** es opcional.

Sentencia *Select Case*

La sentencia ***Select case*** es una sentencia condicional de ramificaciones múltiple. Consiste en una sentencia de control (selector) y una lista de los valores posibles de la expresión.

Sentencia *For*

La sentencia ***for*** se utiliza para implementación del bucle iterativo cuando el número de iteraciones está especificado.

Sentencia *While*

La sentencia ***While*** se utiliza para implementación del bucle iterativo cuando el número de iteraciones no está especificado. Es necesario comprobar la condición de iteración antes de la ejecución del bucle. La sintaxis de la sentencia ***While*** es la siguiente:

```
While expression  
    Statement  
Wend
```

La sentencia ***Statement*** se ejecuta repetidamente siempre que el valor de la expresión ***expression*** sea cierto, el valor de la expresión se comprueba antes de que se ejecute la siguiente iteración. Si el valor de la expresión es falso antes de entrar al bucle, no se ejecuta ninguna iteración.

Sentencia *Do*

La sentencia ***Do*** se utiliza para implementación de bucle iterativo cuando el número de iteraciones no es especificado. La sentencia se ejecuta repetitivamente hasta que la expresión sea cierta. Es necesario chequear la condición de iteración al final del bucle. La sintaxis de la sentencia ***Do*** es la siguiente:

```
Do  
    Statement  
Loop until expression
```

La sentencia **Statement** se ejecuta repetidamente hasta que el valor de la expresión **expression** llegue a ser cierta. La expresión se evalúa después de cada iteración así que la sentencia se ejecutará por lo menos una vez.

Entradas:

Un pulsador

Salidas:

Foco

Procesos:

Si se apaga el botón el foco se enciende por 3 segundos, luego se apaga 1 segundo y se vuelve a encender el foco por 3 segundos, caso contrario el foco permanece prendido.

Diagrama de Flujo:

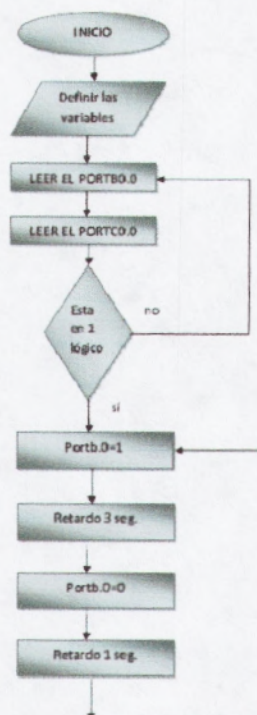


Figura 4.16: Diagrama ASM de práctica 8. Fuente: Propia del Autor.

Programación en MikroBasic:

```
program practica1_8
trisc=$FF 'Puerto C como entradas
trisb=$0 'Puerto B como salidas
main:
if portc.0=1 then
  portb.0=1
  delay_ms(3000)
  portb.0=0
  delay_ms(1000)
  portb.0=1
  delay_ms(3000)
  portb.0=0
else
  portb.0=1
end if
goto main
end
```

Simulación:

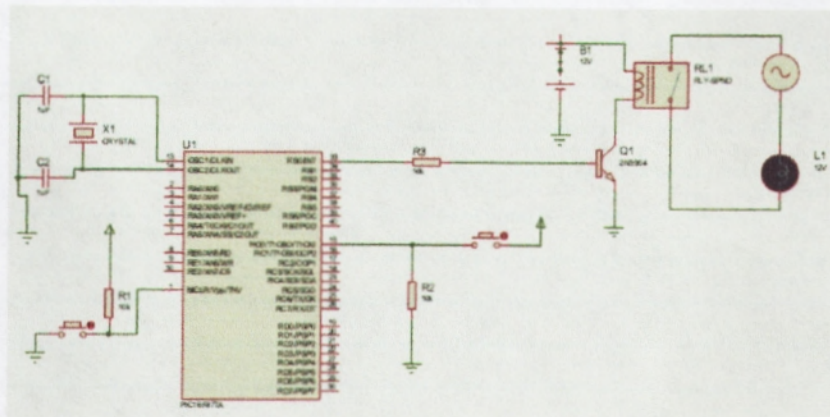


Figura 4.17: Simulación del programa de la octava práctica²⁹.

²⁹ Simulación del circuito de la práctica 8 en ISIS de Proteus. ISIS Professional v7.6 SP2. © Labcenter Electronics 1989-2010

4.9 Práctica # 9: Contador ascendente de 0 a 9.

Entradas:

Utilización de declaraciones con el microcontrolador PIC.

Salidas:

Display de 7 segmentos

Procesos:

Utilizando la sentencia *For-next*, elaborar un contador ascendente de 0 a 9, que realice repeticiones sucesivas.

Diagrama de Flujo:

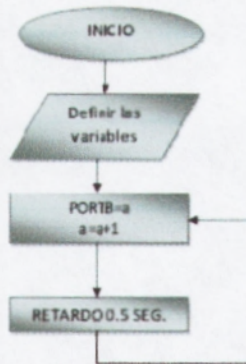


Figura 4.18: Diagrama ASM de práctica 9. Fuente: Propia del Autor.

Programación en MikroBasic:

```
program practica1_9
```

```
dim i,a as byte
```

```
main:
```

```
trisb=0
```

```
i=1
```

```
a=0
```

```
for i=1 to 10
```

```
portb=a
```

```
a=a+1
```

```

delay_ms(500)
next i
if a=10 then
a=0
else
end if
delay_ms(500)
goto main
end

```

Simulación:

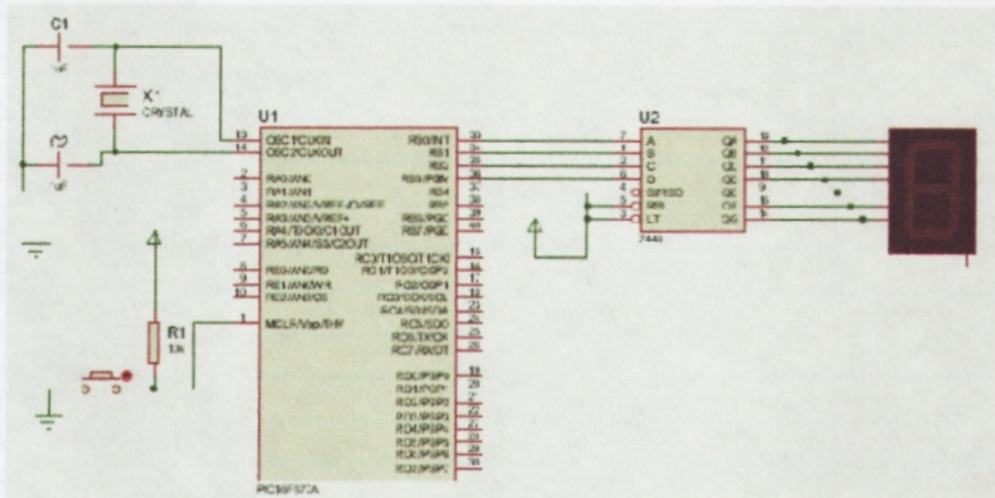


Figura 4.19: Simulación del programa de la novena práctica³⁰.

4.10 Práctica # 10: Contador descendente de 0 a 9.

Entradas:

Ninguna.

Salidas:

Display de 7 segmentos

³⁰ Simulación del circuito de la práctica 9 en ISIS de Proteus. ISIS Professional v7.6 SP2. © Labcenter Electronics 1989-2010

Procesos:

Utilizando la sentencia *For-next*, elaborar un contador descendente de 0 a 9, que realice repeticiones sucesivas.

Diagrama de Flujo:

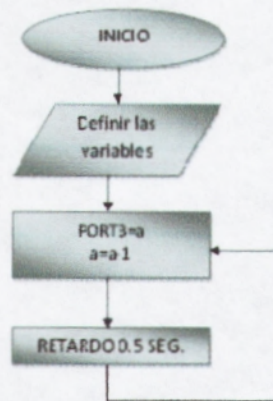


Figura 4.20: Diagrama ASM de práctica 10. Fuente: Propia del Autor.

Programación en MikroBasic:

```
program practica1_10
```

```
dim i,a as byte
```

```
main:
```

```
trisb=0
```

```
i=1
```

```
a=9
```

```
for i=0 to 9
```

```
portb=a
```

```
a=a-1
```

```
delay_ms(500)
```

```
next i
```

```
if a=10 then
```

```
a=9
```

```
else
```

```
end if
```



```

delay_ms(500)
goto main
end

```

Simulación:

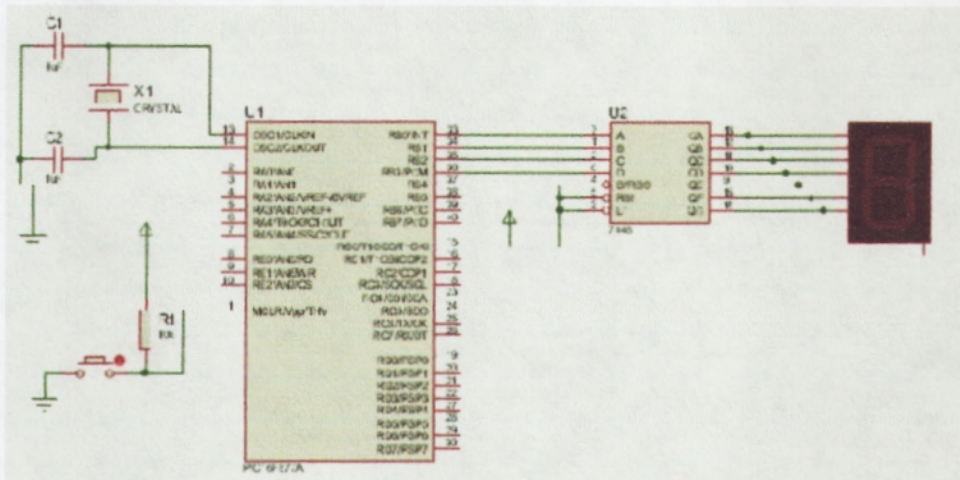


Figura 4.21: Simulación del programa de la décima práctica³¹.

4.11 Práctica # 11: Contador ascendente de dos dígitos.

Entradas:

Ninguna.

Salidas:

Dos display de 7 segmentos

Procesos:

Utilizando la sentencia **while-wend**, elaborar un contador ascendente dos dígitos.

³¹ Simulación del circuito de la práctica 10 en ISIS de Proteus. ISIS Professional v7.6 SP2. © Labcenter Electronics 1989-2010

Diagrama de Flujo:

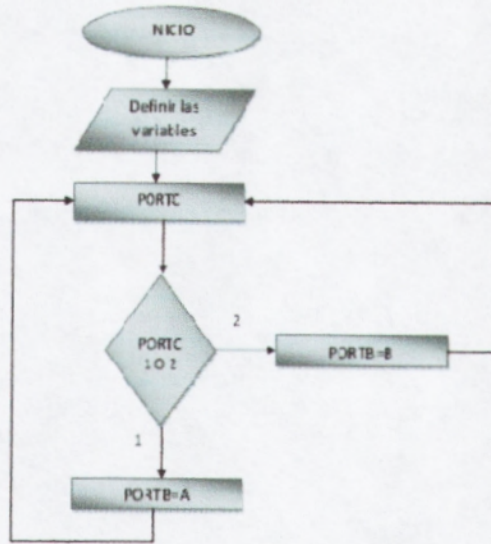


Figura 4.22: Diagrama ASM de práctica 11. Fuente: Propia del Autor.

Programación en MikroBasic:

```
program practica1_11
```

```
dim is A,B as byte
```

```
dim I as word
```

```
A=0
```

```
B=0
```

```
I=0
```

```
main:
```

```
TRISB=0
```

```
TRISC=0
```

```
While true.
```

```
portc=1
```

```
portb=A
```

```
delay_us(248)
```

```
portc=2
```

```
portb=B
```

```
delay_us(248)
```

```

inc (I)
if I=2000 then
inc (B)
        if B=10 then
                B=0
                Inc (A)
                else
                if A=10 then
                        A=0
                        end if
                end if
        end if
I=0
else
end if
wend
end

```

Simulación:

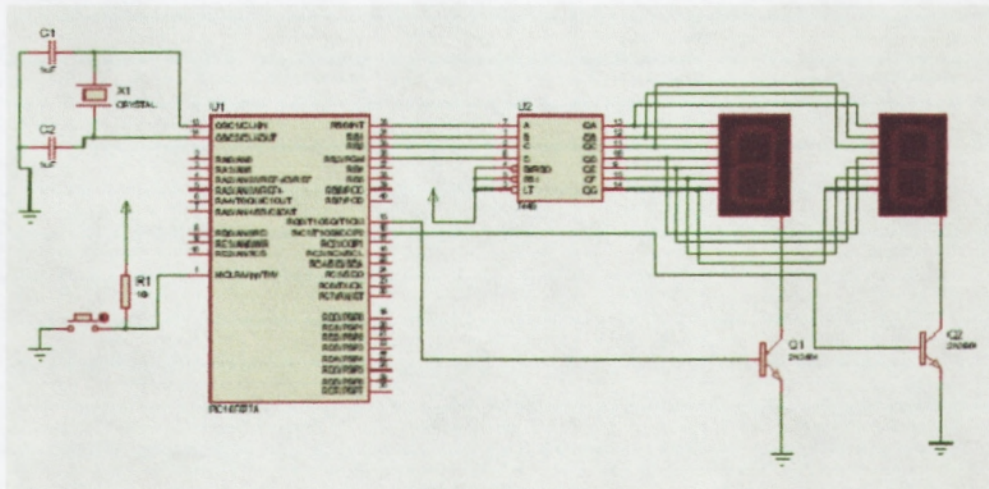


Figura 4.23: Simulación del programa de la onceava práctica³².

³² Simulación del circuito de la práctica11 en ISIS de Proteus. ISIS Professional v7.6 SP2. © Labcenter Electronics 1989-2010

4.12 Práctica # 12: Temporizadores - 1.

Objetivo: Utilizar los temporizadores con el microcontrolador PIC.

Entradas:

Ninguna.

Salidas:

Display de 7 segmentos.

Procesos:

Utilizar el temporizador para elaborar un contador de 1 segundo, cuyo valor se observa en un display de 7 segmentos.

Diagrama de Flujo:

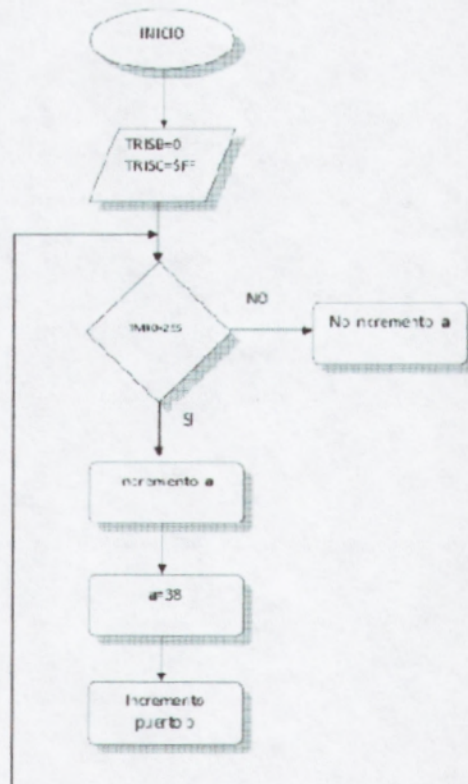


Figura 4.24: Diagrama ASM de práctica 12. Fuente: Propia del Autor.

Programación en MikroBasic:

```
program practica1_12
```

```
dim a as byte
```

```
a=0
```

```
main:
```

```
TRISA=$FF
```

```
TRISB=$0
```

```
Option_reg=%10000111
```

```
Intcon=$0
```

```
TMR0=0
```

```
portb=$0
```

```
LAZO:
```

```
if tmr0=255 then
```

```
    inc(a)
```

```
    if a=38 then
```

```
        inc(portb)
```

```
    end if
```

```
    end if
```

```
GOTO LAZO
```

```
end.
```

Simulación:

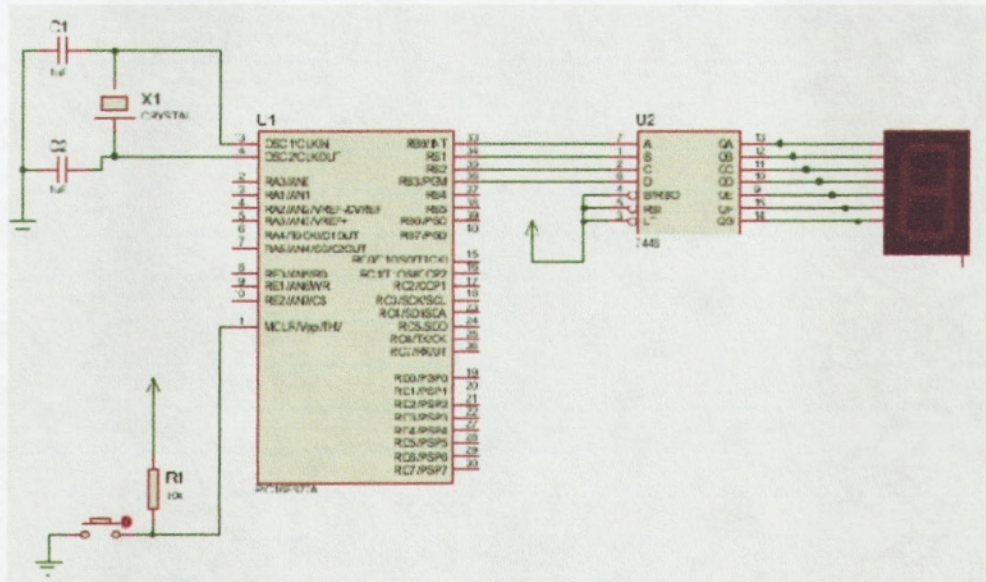


Figura 4.25: Simulación del programa de la doceava práctica ³³.

4.13 Práctica # 13: Temporizadores - 2.

Entradas:

Ninguna.

Salidas:

Display de 7 segmentos.

Procesos:

Elabore un programa que realice las operaciones del numeral 1, pero con TMR0=96, TMR0=155.

Diagrama de Flujo:

³³ Simulación del circuito de la práctica12 en ISIS de Proteus. ISIS Professional v7.6 SP2. © Labcenter Electronics 1989-2010

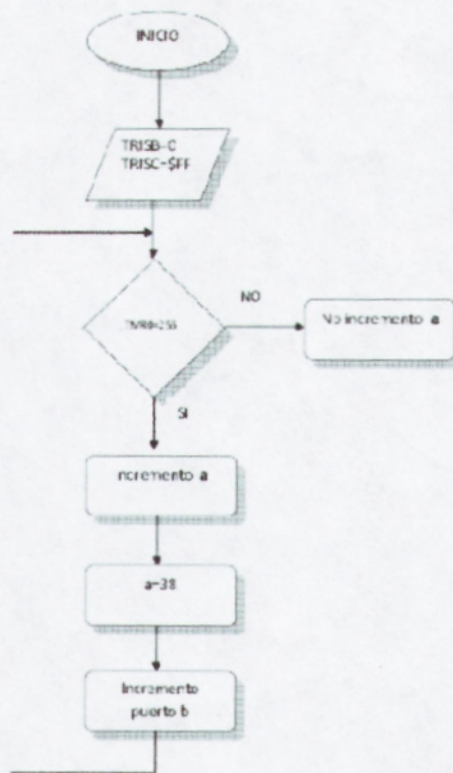


Figura 4.26: Diagrama ASM de práctica 13. Fuente: Propia del Autor.

Programación en MikroBasic:

```
program practica1_13
```

```
dim a as byte
```

```
a=0
```

```
main:
```

```
TRISA=$FF
```

```
TRISB=$0
```

```
Option_reg=%10000111
```

```
Intcon=$0
```

```
TMR0=155
```

```
portb=$0
```

```
LAZO:
```

```
if tmr0=255 then
```

```

inc(a)
if a=38 then
  inc(portb)
end if
end if
GOTO LAZO
end.

```

Simulación:

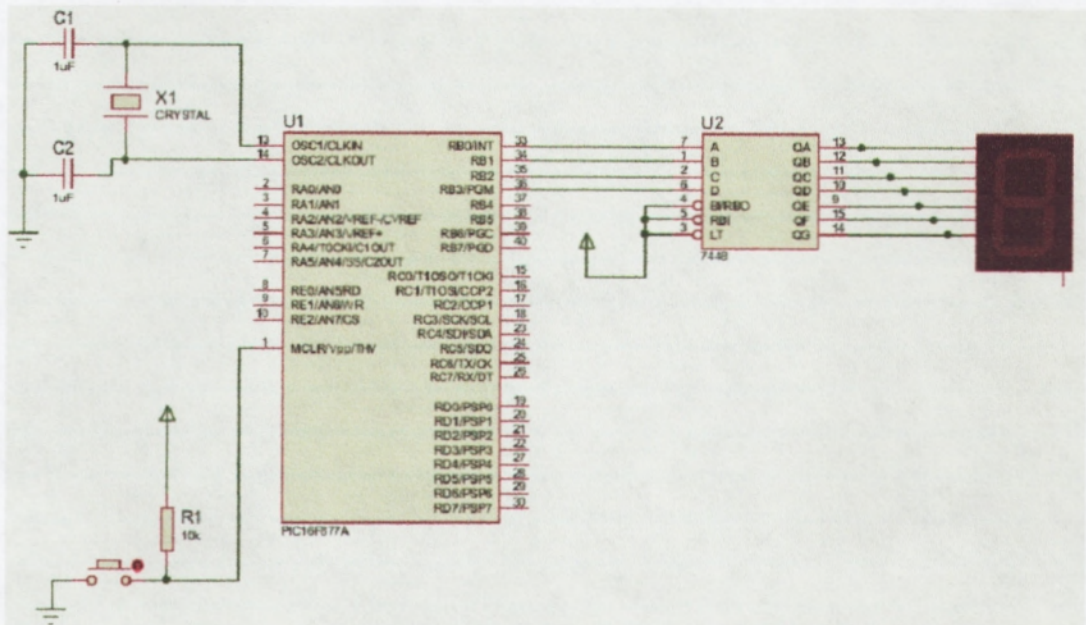


Figura 4.27: Simulación del programa de la treceava práctica³⁴.

4.14 Práctica # 14: Temporizadores - 3.

Entradas:

Pulsos.

Salidas:

Incrementar en un display de 7 segmentos.

³⁴ Simulación del circuito de la práctica13 en ISIS de Proteus. ISIS Professional v7.6 SP2. © Labcenter Electronics 1989-2010

Procesos:

Elabore un programa que utilice el **timer** en modo contador externo, y visualice al incremento cada cuatro pulsos en un display de 7 segmentos.

Diagrama de Flujo:

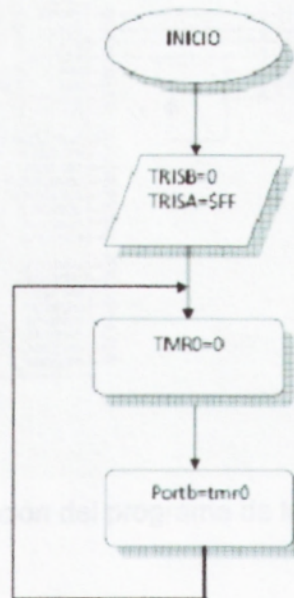


Figura 4.28: Diagrama ASM de práctica 14. Fuente: Propia del Autor.

Programación en MikroBasic:

```
program practica1_14
main:
TRISA=$FF
TRISB=$0
Option_reg=%10100010
Intcon=$0
TMR0=0

LAZO:
Portb=tmr0
GOTO LAZO
```

CAPÍTULO 5

CONCLUSIONES Y RECOMENDACIONES

5.1 CONCLUSIONES

- i. Definición clara de las habilidades de los estudiantes para formar futuros profesionales comprometidos con la sociedad y el conocimiento, en su recorrido y estudio de la asignatura de Microcontroladores, tanto en práctica como en la teoría, a partir de los objetivos que persigue la asignatura.
- ii. El IDE de mikroBasic resulta amigable para programadores que han tenido experiencia usando lenguaje de programación BASIC.
- iii. Se conocen cada unos de los recursos con los que debe contar o tener los diferentes Sistemas Modulares de entrenamiento de Microcontroladores, para manejarse de acuerdo al conjunto de acciones a emprender en el desarrollo de las prácticas por los estudiantes en el laboratorio de electrónica.
- iv. Los microcontroladores PIC nos permite controlar de una manera fácil los diferentes tipos de displays y arreglos de leds gracias a la considerable cantidad de pines bidireccionales que este contiene.
- v. El Sistema Modular EasyPic6 propuesto en el capítulo se implementa como una plataforma o entrenador que es abierto, dinámico y que evoluciona en forma constante, para así integrar nuevos módulos ya existentes en el mercado nacional e internacional de dispositivos electrónicos.

- vi. El Sistema Modular EasyPic6 instalado en el Laboratorio de Electrónica será utilizado en la materia de microcontroladores y materias afines que aparecen la malla curricular de la Carrera de Ingeniería en Telecomunicaciones con muy pocas modificaciones, como lo es los sistemas microprocesadores.

5.2 RECOMENDACIONES

- i. Tener a disposición toda la ayuda bibliográfica y de software necesario acerca de la programación en mikroBasic para lograr optimizar y profundizar la programación.
- ii. Contar con un manual de la familia de los microcontroladores PIC (MicroChip), para conocer la adecuada conexión de cada uno de los pines, aunque se puede obtener de internet.
- iii. A partir del Sistema Modular EasyPic6, incentivar a los alumnos a diseñar y construir nuevos sistemas modulares de entrenamiento o maquetas que permitan trabajar en forma similar al Sistema Modular implementado, para así mejorar la enseñanza y el conocimiento adquirido.
- iv. Que la Universidad Católica de Santiago de Guayaquil a través de la Facultad Técnica para el Desarrollo para el desarrollo de concursos y que sean incentivados los estudiantes, para así apreciar sus diseños y creatividad en beneficio de la colectividad y sociedad.
- v. Así mismo sería necesario el despliegue de esta herramienta muy necesaria para su enseñanza en posgrado en la Maestría de Telecomunicaciones, específicamente en materias como Diseño Electrónico para extender el uso del Sistema Modular EasyPic6.

BIBLIOGRAFÍA

Libros, Tesis y otros.

[Ángulo, 03a] Ángulo U, J. *Diseño Práctico de Aplicaciones. Primera Parte PIC16F84*. Mc Graw Hill, 2003.

[Ángulo, 03b] Ángulo J, Romero S, Ángulo I. *Diseño Práctico de Aplicaciones. Segunda Parte PIC16F87x*. Mc Graw Hill, 2003.

[Duque, 1997] Duque, Edison. *Curso Básico de Microcontroladores PIC*. Cekit, Compañía Editorial Electrónica. Colombia, 1997.

[Mandado, 07] Mandado E., Menéndez L., Fernández L., López E. *Microcontroladores PIC. Sistema integrado para el autoaprendizaje*. Marcombo Ediciones Técnicas, S.A. España, 2007.

[Sánchez, 2007] Eduardo Sánchez A, MsC., *Curso de Microcontroladores I*, [Online] W.W.W.:
<http://www.itnogales.edu.mx/Microprocesadores%20I/Apuntes-%20Micros/Curso%20Microcontroladores.pdf>

[Valdés, 07] Fernando Valdés P, Ramon Pallàs Areny. *Microcontroladores: Fundamentos y Aplicaciones con PIC*. Marcombo Ediciones Técnicas, S.A. España, 2007.

[Valdivieso, 09] Carlos E. Valdivieso A. *Sistema Modular para el Laboratorio de Microcontroladores de la ESPOL*. Escuela Superior Politécnica, 2009.

ANEXOS

EASYPIC 6

Diagrama esquemático

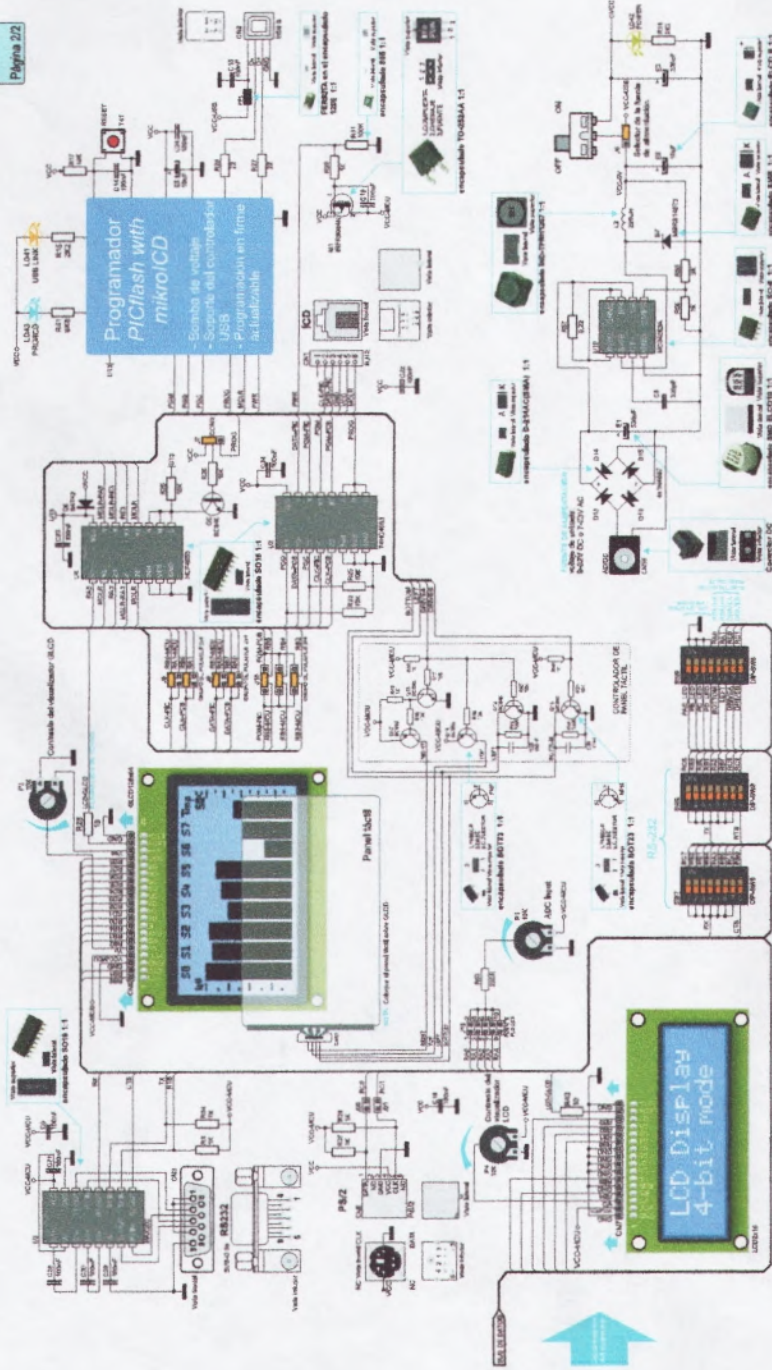
Este documento proporciona un esquema eléctrico del sistema de desarrollo EasyPIC6. El dispositivo ha sido diseñado con cuidado y se ha prestado la atención debida a elegir los componentes que se utilizarán en la fabricación para las partes más importantes del sistema como el PIC, el convertidor de voltaje, el regulador de voltaje, el controlador de motor y el LCD. Además, los componentes son montados en un módulo de desarrollo en superficie, lo que facilita la generación de prototipos para soldadura. Lo que tenemos ahora como resultado es un producto fiable de alta calidad que logra los más altos estándares del mundo aplicados en la industria electrónica y en la protección del medio ambiente.

Sistema de desarrollo

MikroElektronika

SOFTWARE AND HARDWARE SOLUTIONS FOR EMBEDDED WORLD

...making it simple.



Manufactured in accordance with the ISO 9001 standard. The components used are MikroElektronika products or products of companies that are ISO 9001 certified.

Supported Microcontrollers

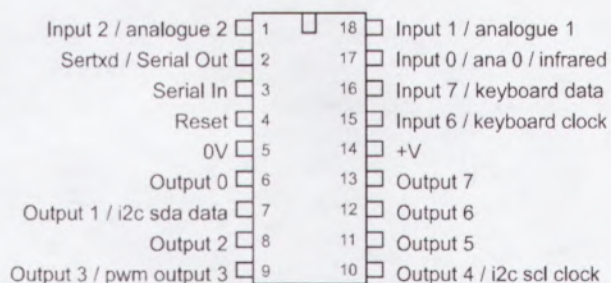
The EasyPIC6 development system supports a wide range of microcontrollers including over **160 PIC microcontrollers**. Newly released PIC microcontrollers will be supported by a new version of PICFlash software that is updated regularly. Here is a list of all microcontrollers supported by the EasyPIC6 development system:

PIC10 AND PIC12 MICROCONTROLLERS FAMILY								
PIC10F200	PIC10F202	PIC10F204	PIC10F206	PIC10F220	PIC10F222	PIC12F508	PIC12F509	PIC12F510
PIC12F519	PIC12F609	PIC12F615	PIC12F629	PIC12F635	PIC12F675	PIC12F683	PIC12HV609	PIC12HV615
PIC16 MICROCONTROLLERS FAMILY								
PIC16F505	PIC16F506	PIC16F526	PIC16F54	PIC16F57	PIC16F59	PIC16F610	PIC16F616	PIC16F627
PIC16F627A	PIC16F628	PIC16F628A	PIC16F630	PIC16F631	PIC16F636	PIC16F639	PIC16F648A	PIC16F676
PIC16F677	PIC16F684	PIC16F685	PIC16F687	PIC16F688	PIC16F689	PIC16F690	PIC16F716	PIC16F72
PIC16F722	PIC16F723	PIC16F724	PIC16F726	PIC16F727	PIC16F73	PIC16F737	PIC16F74	PIC16F747
PIC16F76	PIC16F767	PIC16F77	PIC16F777	PIC16F785	PIC16F818	PIC16F819	PIC16F83	PIC16F84
PIC16F84A	PIC16F87	PIC16F870	PIC16F871	PIC16F872	PIC16F873	PIC16F873A	PIC16F874	PIC16F874A
PIC16F876	PIC16F876A	PIC16F877	PIC16F877A	PIC16F88	PIC16F882	PIC16F883	PIC16F884	PIC16F886
PIC16F887	PIC16F913	PIC16F914	PIC16F916	PIC16F917	PIC16F946	PIC16HV610	PIC16HV616	PIC16HV785
PIC18 MICROCONTROLLERS FAMILY								
PIC18F1220	PIC18F1230	PIC18F1320	PIC18F1330	PIC18F2220	PIC18F2221	PIC18F2320	PIC18F2321	PIC18F2331
PIC18F2410	PIC18F242	PIC18F2420	PIC18F2423	PIC18F2431	PIC18F2439	PIC18F2450	PIC18F2455	PIC18F2458
PIC18F248	PIC18F2480	PIC18F2510	PIC18F2515	PIC18F252	PIC18F2520	PIC18F2523	PIC18F2525	PIC18F2539
PIC18F2550	PIC18F2553	PIC18F258	PIC18F2580	PIC18F2585	PIC18F2610	PIC18F2620	PIC18F2680	PIC18F2682
PIC18F2685	PIC18F4220	PIC18F4221	PIC18F4320	PIC18F4321	PIC18F4331	PIC18F4410	PIC18F442	PIC18F4420
PIC18F4423	PIC18F4431	PIC18F4439	PIC18F4450	PIC18F4455	PIC18F4458	PIC18F448	PIC18F4480	PIC18F4510
PIC18F4515	PIC18F452	PIC18F4520	PIC18F4523	PIC18F4525	PIC18F4539	PIC18F4550	PIC18F4553	PIC18F458
PIC18F4580	PIC18F4585	PIC18F4610	PIC18F4620	PIC18F4680	PIC18F4682	PIC18F4685		

PICAXE 18X/28X/40X Extended Features...

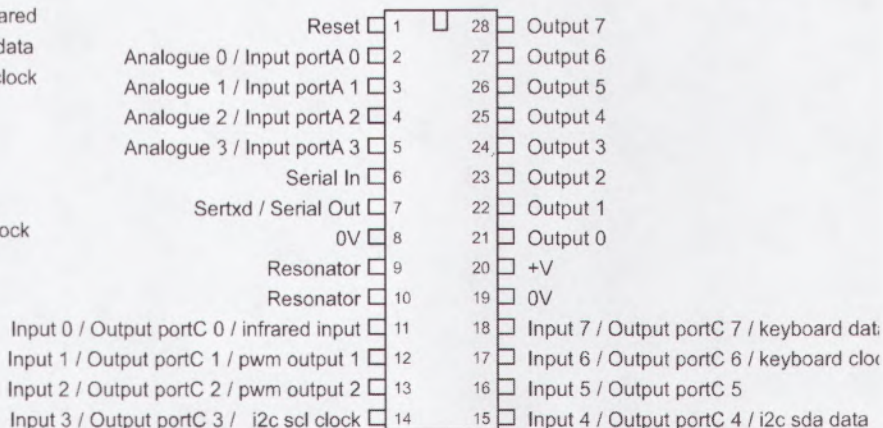
PICAXE Type	IC Size	Memory (lines)	I/O Pins	Outputs	Inputs	ADC (L =low)	Data Memory	Polled Interrupt
PICAXE-08	8	40	5	1-4	1-4	1L	128 - prog	-
PICAXE-18	18	40	13	8	5	3L	128 - prog	-
PICAXE-18A	18	80	13	8	5	3	256	Yes
PICAXE-18X	18	600	14	9	5	3	256 + i2c	Yes
PICAXE-28	28	80	20	8	8	4	64 + 256	-
PICAXE-28A	28	80	20	8	8	4	64 + 256	Yes
PICAXE-28X	28	600	21	9-17	0-12	0-4	128 + i2c	Yes
PICAXE-40X	40	600	32	9-17	8-20	3-7	128 + i2c	Yes

PICAXE-18X



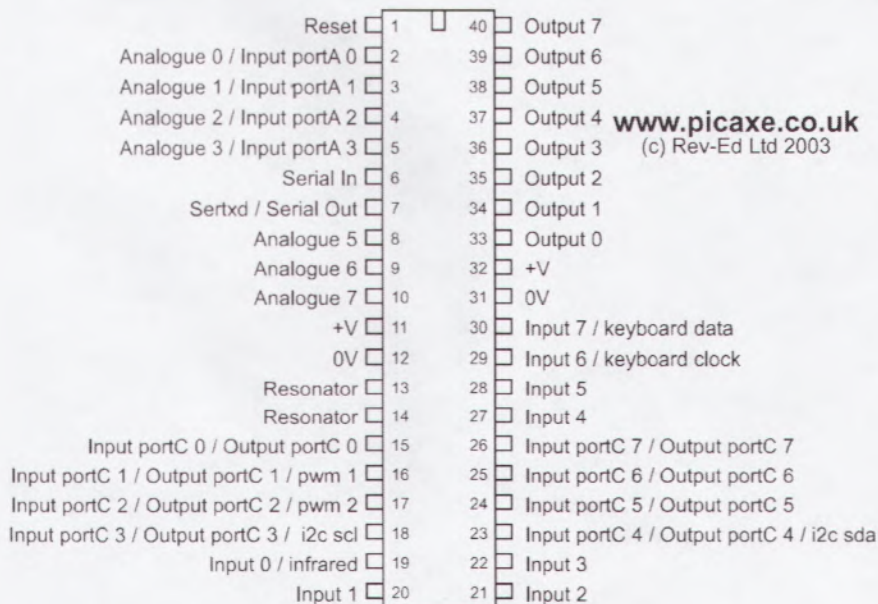
www.picaxe.co.uk
(c) Rev-Ed Ltd 2003

PICAXE-28X



www.picaxe.co.uk
(c) Rev-Ed Ltd 2003

PICAXE-40X



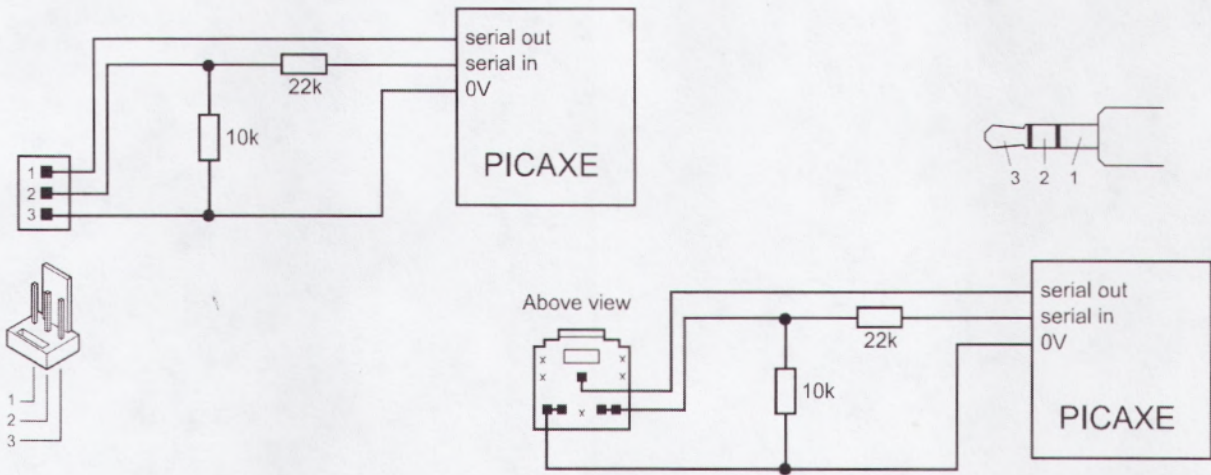
www.picaxe.co.uk
(c) Rev-Ed Ltd 2003

In this datasheet...

- Section 1 - PICAXE Commands
- Section 2 - What's New
- Section 3a - PICAXE-28X input/output pins
- Section 3b - PICAXE-40X input/output pins
- Section 4 - Resonator Frequency and Overclocking

Serial Download Circuit:

The serial download circuit for all PICAXE microcontrollers is (straight or 'stereo plug' cable connections):



SECTION 1 - PICAXE Commands: (new X part commands in bold)

Output -	high, low, toggle, pulsout, let pins =	Please see the BASIC Commands
Sound -	sound	help file for more detailed syntax help
Input -	if...then, readadc, readac10 , pulsin, button	and information about each command.
Serial -	serin, serout, sertxd	
Program Flow -	goto, gosub, return, branch	
Loops -	for...next	
Mathematics -	let... (+, -, *, **, /, //, max, min, &, , ^, &/, /, ^/)	
Variables -	if...then, random, lookdown, lookup	
Data memory -	eeprom, write, read	
Delays -	pause, wait, nap, sleep, end	
Miscellaneous -	symbol, debug	
RAM -	peek, poke	
Servo Control -	servo	
Infrared -	infrain	
Interrupt -	setint	
Temperature -	readtemp , readtemp12	
Keyboard -	keyin , keyled	
1-wire Serial No -	readown	
I2C -	readi2c , writei2c , i2cslave	
PWM -	pwmout	
Counting -	count	

SECTION 2 - What's new in the PICAXE-18X, 28X, 40X?

The extended X parts support all the standard commands and features, with the following enhancements:

- Program memory 8x as long (approx. 600 lines rather than 80), with intelligent download
- Continuously driven pwm motor drive outputs (pwmout command)
- Count high frequency pulses within a set time period (count command)
- Large data memory (128 or 256 bytes) (read/write commands)
- i2c bus support for EEPROMs and other devices (i2cslave/writei2c/readi2c commands)
- Interrupt feature on inputs (setint command)
- Accurate digital temperature sensor interface (readtemp/readtemp12 commands)
- 10 bit and 8 bit adc option (readadc10/readadc commands)
- User serial output via the serout pin / programming cable (sertxd command)
- 4800 baud rate option (and faster at higher clock frequencies) (serin/serout commands)
- Read serial number from any Dallas 1-wire device (e.g. iButton) (readowsn command)
- Computer keyboard interface on inputs 6 and 7 (keyin, keyed command)
- Software support for increased clock frequency (see section 3 of this datasheet).

See the BASIC Commands datasheet (v3.5 or greater) for further information on each command.

In addition the PICAXE-28X and 40X have a more **flexible i/o pin** layout to allow the user to select more inputs and/or outputs than the standard configuration. See section 2 of this datasheet.

Memory Size

The X parts have a memory size 8x larger than the A parts (2048 bytes rather than 256 bytes). This means it can store a program of approximately **500-700 lines** of BASIC code (depending on commands used).

To reduce download times the X parts only download the appropriate (used) pages of memory. Therefore a shorter program will download quicker than a longer program

PICAXE-40X

The PICAXE-40X is electronically configured as a 'special' version of the PICAXE-28X (with additional pins) Therefore when using the Programming Editor software the 'PICAXE-28X' mode is used for **programming both** the PICAXE-28X and the PICAXE-40X microcontrollers.

SECTION 3a - PICAXE-28X Input/Output Pins

To provide greater flexibility, the input/output pin configuration of the PICAXE-28X can be varied by the user. The default power up settings are the same as the other PICAXE-28 parts (8 in, 8 out, 4 analogue).

PORTA (legs 2 to 5) provide 4 analogue inputs (default) or up to 4 digital inputs.

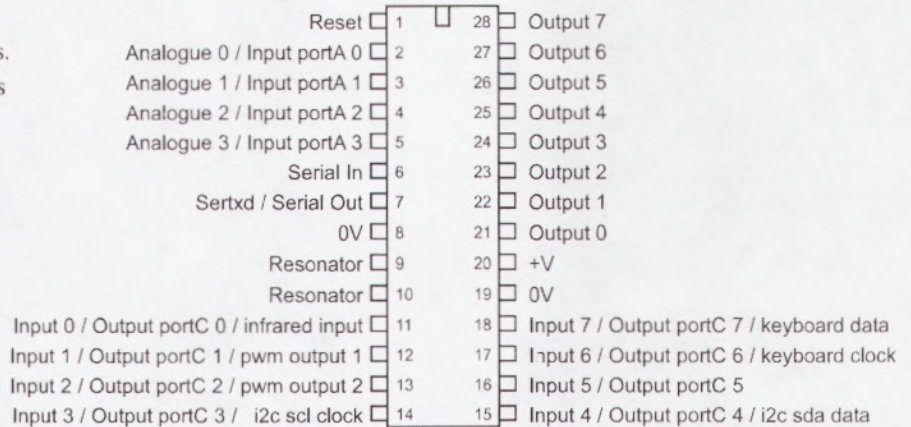
PORTB (leg 21 to 28) provide 8 fixed outputs.

PORTC (leg 11 to 18) provide 8 digital inputs (default) or up to 8 outputs.

This gives a maximum of:

- 12 digital inputs
- 16 outputs
- 4 analogue inputs

PICAXE-28X



PORTA Functions

Leg	Default Function	Second Function
2	analogue 0	porta input 0
3	analogue 1	porta input 1
4	analogue 2	porta input 2
5	analogue	porta input 3

www.picaxe.co.uk
(c) Rev-Ed Ltd 2003

PORTB Functions

PORTB pins are fixed as outputs and cannot be altered.

PORTC Functions

Leg	Default Function	Second Function	Special Function
11	input 0	output portc 0	infrared (input)
12	input 1	output portc 1	pwm 1 (output)
13	input 2	output portc 2	pwm 2 (output)
14	input 3	output portc 3	i2c scl clock (input)
15	input 4	output portc 4	i2c sda data (input)
16	input 5	output portc 5	
17	input 6	output portc 6	keyboard clock (input)
18	input 7	output portc 7	keyboard data (input)

The portC pins can be used as the default inputs, changed to outputs, or used with their special function via use of the infrain, keyin, i2cslave, or pwmout command as appropriate.

The second or special function of the pins are selected by modified commands as explained in the next section.

SECTION 3b - PICAXE-40X Input/Output Pins

To provide greater flexibility, the input/output pin configuration of the PICAXE-40X can be varied by the user.

PORTA (legs 2 to 5) provide 4 analogue inputs (default) or up to 4 digital inputs.

PORTB (leg 32 to 40) provide 8 fixed outputs.

PORTC (leg 15-18, 23-26) provide 8 digital inputs (default) or up to 8 outputs.

PORTD (leg 19-22, 27-30) provide 8 digital inputs

PORTE (leg 8 to 10) provide 3 analogue inputs

This gives a maximum of :

- 20 digital inputs
- 16 outputs
- 7 analogue inputs

PORTA Functions

Leg	Default Function	Second Function
2	analogue 0	porta input 0
3	analogue 1	porta input 1
4	analogue 2	porta input 2
5	analogue	porta input 3

PORTB Functions

PORTB pins are fixed as outputs and cannot be altered.

PORTC Functions

Leg	Default Function	Second Function	Special Function
15	input portc 0	output portc 0	
16	input portc 1	output portc 1	pwm 1 (output)
17	input portc 2	output portc 2	pwm 2 (output)
18	input portc 3	output portc 3	i2c scl clock (input)
23	input portc 4	output portc 4	i2c sda data (input)
24	input portc 5	output portc 5	
25	input portc 6	output portc 6	
26	input portc 7	output portc 7	

The portC pins can be used as the default inputs, changed to outputs, or used with their special function via use of the i2cslave or pwmout command as appropriate.

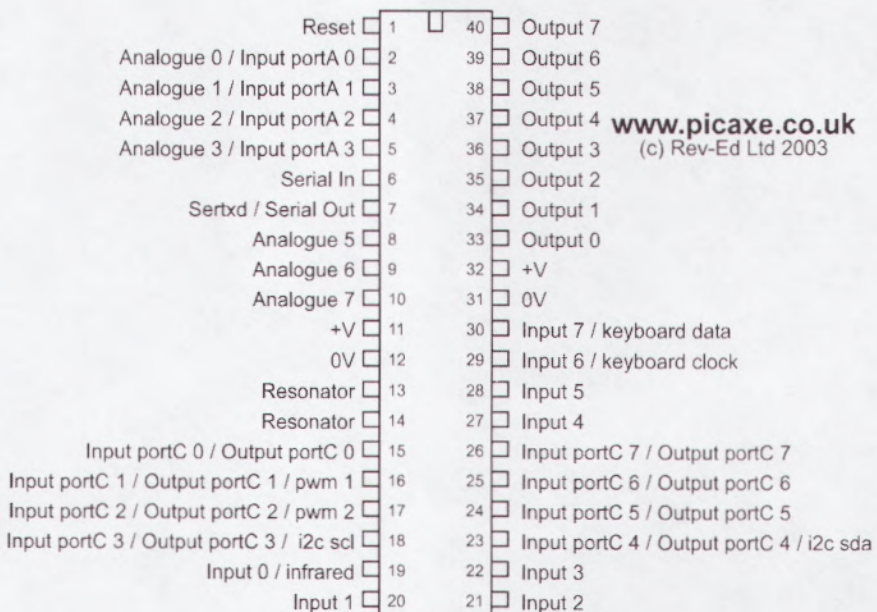
PORTD Functions

Leg	Default Function	Special Function
19	input 0	infrared (input)
20	input 1	
21	input 2	
22	input 3	
27	input 4	
28	input 5	
29	input 6	keyboard clock (input)
30	input 7	keyboard data (input)

PORTE Functions

PORTE pins are fixed as analogue inputs and cannot be altered.

PICAXE-40X



www.picaxe.co.uk
(c) Rev-Ed Ltd 2003

SECTION 3c - Using portA analogue inputs as digital inputs (28X, 40X)

The portA pins 0 to 3 (legs 2 to 5) are, by default, configured as analogue inputs. However with the PICAXE-28X and -40X they can also be used as simple digital inputs.

The following syntax is used to test the input condition:

```
if portA pin0 = 1 then jump
```

i.e. the additional keyword 'portA' is inserted after the 'if' command.

to test if two (or more) portA inputs are on

```
if portA pin0 = 1 AND pin1 = 1 then jump
```

to test if either of two (or more) portA inputs are on

```
if portA pin0 = 1 OR pin1 = 1 then jump
```

Note the portA command is only required once after the 'if' command.

It is not possible to test inputs on two different ports within the same if...then statement.

It is not possible to access the portA pins with any other 'input' type commands (count, pulsIn etc).

Therefore these pins should be reserved as simple on/off switches.

SECTION 3d - Using portC as inputs (40X)

On the PICAXE-28X portC are the standard input pins and addressed by the standard `if pin0 =` command.

On the PICAXE-40X portD are the standard inputs, and hence use the standard `if pin0 =` command. Therefore for portC inputs the extra keyword `portC` must be used (as in the `if portA pin0 =` example above).

SECTION 3e - Using portC as outputs (28X, 40X)

The portC pins are, by default, digital input pins.

However with the PICAXE-28X and -40X they can also be configured to be used as digital outputs.

To convert the pin to output and make it high

```
high portC 1
```

To convert the pin to output and make it low

```
low portC 1
```

To convert all the pins to outputs

```
let dirsc = %11111111
```

To convert all the pins to inputs

```
let dirsc = %00000000
```

Note that 'dirsc' uses the common BASIC notation 0 for input and 1 for output. (Advanced - If you are more familiar with assembler code programming you may prefer to use the command 'let trisc =' instead, as this uses the inverted assembler notation - 1 for input and 0 for output. Do not attempt to directly poke the trisc register (poke command) as the PICAXE bootstrap refreshes the register setting regularly).

To switch all the outputs on portC high

```
let pinsc = %11111111
```

(or) `let portC = %11111111`

To switch all the outputs on portC low

```
let pinsc = %00000000
```

(or) `let portC = %00000000`

To use portC 1 and portC 2 as pwm controlled outputs use the `pwmout` command (see the BASIC Commands help file for further information). The pwm output is maintained continuously in the background, making these pins ideal for controlling motors etc.

It is not possible to access the portc pins with any other 'output' type commands (sound, serout, pulsout etc). Therefore these pins should be reserved as simple on/off outputs (apart from the pwm control on 1 and 2).

When using the special input functions (infrared sensor (0), or an i2c device (3, 4), or a keyboard (6, 7)) you must take care to ensure that the appropriate pins are maintained as inputs. Converting these pins to outputs may damage the external device and/or the microcontroller.

SECTION 4 - Resonator Frequency and Overclocking.

All PICAXE functions are based upon a 4MHz resonator frequency. This is the only frequency recommended. However the user may choose to 'overclock' the X parts if desired, although this is not recommended unless absolutely necessary for a particular project (e.g. when using the count command).

With the -08, -18, -18A the internal resonator is fixed at 4MHz and cannot be altered.

With the -18X the internal resonator has a default value of 4MHz. However it can be increased by the user to 8MHz via use of the 'setfreq' command.

With the -28 and -28A an external 4MHz resonator must be used.

With the -28X / -40X an external 4MHz 3 pin ceramic resonator is normally used, but it is also possible to use a faster resonator (8 or 16Mhz), although this will affect the operation of some of the commands.

NB PICAXE-28X firmware version 7.0 can be used at 4 or 8 MHz

PICAXE-28X or -40X firmware version 7.1 (or greater) can be used at 4, 8 or 16 MHz

The Programming Editor software supports resonator frequencies of 4, 8 and 16MHz only. No other frequencies are supported. If any other frequency is used it will not be possible to download a new program into the PICAXE microcontroller.

To change the frequency:

PICAXE-18X

Download a program containing the command `setfreq m4` (for 4 MHz) or `setfreq m8` (for 8MHz). If no `setfreq` command is used in a program the frequency will default to 4MHz. Note the new frequency occurs immediately after the command is run. When downloading new programs, you must ensure the correct frequency (View>Options>Mode) is used to match the last program running in the PICAXE-18X chip. If in doubt perform a 'hard-reset' at 4Hz.

PICAXE-28X and PICAXE-40X

Solder the appropriate external 3pin ceramic resonator into the project board.

Downloading programs at 4, 8, 16MHz

After changing frequency you must select the correct frequency via the View>Options>Mode software menu. If the wrong frequency is selected the program will not download.

Commands affected by resonator frequency.

Many of the commands are affected by a change in resonator frequency. A summary of the important commands affected are given below (see BASIC Commands datasheet for detailed command syntax).

count

The base unit of count is

1ms at 4MHz
0.5ms at 8 MHz
0.25ms at 16 MHz

The pin is checked every

20us at 4MHz (max. 25kHz pulse rate)
10us at 8MHz (max. 50kHz pulse rate)
5us at 16MHz (max. 100kHz pulse rate)

i2slave

The bus speed within i2slave must be adjusted by use of the appropriate frequency keyword

i2cfast / i2cslow at 4Mz
i2cfast8 / i2cslow8 at 8Mz
i2cfast16 / i2cslow16 at 16Mz

If the incorrect keyword is used the i2c function may not work.

pause / wait

The base unit of pause is:

1ms at 4MHz
0.5ms at 8 MHz
0.25ms at 16 MHz

The base unit of wait is:

1s at 4MHz
0.5s at 8 MHz
0.25s at 16 MHz

pulsout / pulsln

The base unit of pulsout/pulsln is:

10us at 4Mhz
5us at 8Mhz
2.5us at 16Mhz

pwmout

The period and duty cycle should be calculated using 4MHz, 8MHz or 16Mhz as appropriate.

serin / serout / sertxd

Due to the sensitive nature of serial communication no guarantee is given that serin or serout commands will work at any frequency other than 4MHz. However the theoretical baud rates at the higher clock frequencies are as follows:

Baudmode	4MHz	8MHz	16MHz
600	600	1200	2400
1200	1200	2400	4800
2400	2400	4800	9600
4800	4800	9600	19200 (also sertxd baud rate)

A maximum of 4800 is recommended for complicated serial transactions.

sound

The note of sound will be multiplied by 2 (8Mhz) or 4 (16MHz).

The duration of sound (12ms at 4MHz) will be reduced to 6ms (8MHz) or 3ms (16MHz)

Commands that do not work at 8 or 16MHz

The following commands will not work at 8 or 16MHz due to timing issues with the external device listed:

- infrain (infrared remote)
- keyin (keyboard)
- keyled (keyboard)
- readtemp / readtemp12 (DS18B20 temperature sensor)
- readown (1-wire device)
- servo (servo)

Commands that are not affected by frequency changes.

The following timing commands are NOT affected as they use a separate internal r/c timer:

- nap and sleep

Análisis Descriptivo del Sistema Modular EasyPic6 e implementación de prácticas para la materia de Microcontroladores

Por: José Huayamave Júpiter, Ricardo Monserrate Coello, Andrés Ponce Orellana, Lissette Robalino Cruz y Jean Carlos Ruiz Castro

Previa a la obtención del título de: Ingeniero en Telecomunicaciones con Mención en Gestión Empresarial

Facultad de Educación Técnica para el Desarrollo
Universidad Católica de Santiago de Guayaquil

Resumen: El desarrollo de la presente Tesis previo a la obtención del Título de Ingeniero en Telecomunicaciones, se encargó de realizar el estudio en mercado nacional e internaciones de los Sistemas Modulares de Entrenamiento de los Microcontroladores para así agilizar y facilitar el aprendizaje y enseñanza de los Microcontroladores en el Laboratorio de Electrónica.

Instalar tanto el software y el hardware del Sistema Modular EasyPic6 en el Laboratorio de Electrónica de la Facultad Técnica para el Desarrollo, para el uso de los docentes y los estudiantes en la materia de microcontroladores, para que puedan poner en práctica sus conocimientos en dichas materia a partir de los manuales adquiridos en este sistema.

Abstract: The development of this thesis before obtaining Engineering Degree in Telecommunications, was commissioned to conduct the study in national and international market for Training Modular Microcontrollers order to expedite and facilitate learning and teaching in Microcontrollers Laboratory of Electronics. Install both software and hardware EasyPic6 Modular System in the Laboratory of Electronics at the Technical Faculty Development for use by teachers and students in the field of microcontrollers, so they can put their knowledge into practice in such matters from textbooks purchased in this system.

Introducción: Los Microcontroladores son uno de los componentes cruciales que más están experimentando un gran aumento de potencia debido al imparable desarrollo tecnológico. La consiguiente tendencia es a aumentar la complejidad del software que llevan incorporado.

Objetivo General: Evaluar la operatividad del Sistema Modular EasyPic6 y la de sus aplicaciones con el PIC16F877 para el fortalecimiento del proceso enseñanza aprendizaje en la asignatura de Microcontroladores.

Objetivos Específicos:

1. Diagnosticar el estado actual del arte o marco teórico en los sistemas modulares para Microcontroladores.
2. Diseñar las prácticas de la materia de
3. Microcontroladores para mejorar el proceso enseñanza aprendizaje.
4. Evaluar el sistema modular EasyPic6 a través de las prácticas desarrolladas usando los Microcontroladores Microchip.
5. Implementar los módulos EasyPic6 en el Laboratorio de Electrónica.

ANTECEDENTES

La operatividad del Sistema Modular EasyPic6 con distintas aplicaciones de la familia de los Microcontroladores, permitirá mejorar el proceso de enseñanza – aprendizaje, considerado también como aplicación en las tecnologías de la información y las comunicaciones (TICS) que se están incorporando a la enseñanza de la ingeniería a nivel mundial.

Entrenador de Microcontroladores EasyPic6

La tarjeta EasyPic6 es producida por la empresa **Mikroelektronika** 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 2.11, en la que se han conservado números referenciales para destacar las características de la tarjeta de una mejor manera.

El Sistema Modular EasyPic6 desarrolla todas las funciones para casi todos los microcontroladores PIC de la familia Microchip. Está diseñado para permitir a los estudiantes y docentes para probar fácilmente y explorar las capacidades de los microcontroladores PIC. También permite a los microcontroladores PIC una interfaz con circuitos externos y una amplia gama de dispositivos

periféricos. Los estudiantes pueden, por tanto centrarse en el desarrollo de software solamente.

Es importante seguir con las instrucciones proporcionadas en los manuales pertinentes para instalar los controladores USB y el programa **PICflash** desde el CD del entrenador EasyPic6. Estos controladores USB son imprescindibles para el funcionamiento adecuado del programador incorporado. En caso de que ya tenga un compilador PIC de Mikroelektronika instalado en la PC, no es necesario reinstalar los controladores ya que se van a instalar automáticamente junto con el compilador.



Figura 2.11: Sistema Modular EasyPic6.

Tiene ocho zócalos (ver figura 3.4) separados para la inserción de microcontroladores en los encapsulados DIP40 (PIC de 40 pines), DIP28 (PIC de 28 pines), DIP20 (PIC de 20 pines), DIP18 (PIC de 18 pines), DIP14 (PIC de 14 pines) y DIP8 (PIC de 8 pines).



Figura 3.4: Placa que muestra los 8 zócalos de la tarjeta EasyPic6.

CONCLUSIONES

El Sistema Modular EasyPic6 propuesto en el capítulo se implementa como una plataforma o entrenador que es abierto, dinámico y que evoluciona en forma constante, para así integrar nuevos módulos ya existentes en el mercado nacional e internacional de dispositivos electrónicos.

El Sistema Modular EasyPic6 instalado en el

Laboratorio de Electrónica será utilizado en la materia de microcontroladores y materias afines que aparecen la malla curricular de la Carrera de Ingeniería en Telecomunicaciones con muy pocas modificaciones, como lo es los sistemas microprocesadores.

RECOMENDACIONES

Tener a disposición toda la ayuda bibliográfica y de software necesario acerca de la programación en mikroBasic para lograr optimizar y profundizar la programación.

Contar con un manual de la familia de los microcontroladores PIC (MicroChip), para conocer la adecuada conexión de cada uno de los pines, aunque se puede obtener de internet.

A partir del Sistema Modular EasyPic6, incentivar a los alumnos a diseñar y construir nuevos sistemas modulares de entrenamiento o maquetas que permitan trabajar en forma similar al Sistema Modular implementado, para así mejorar la enseñanza y el conocimiento adquirido.

Bibliografía:

Libros, Tesis y otros.

[Ángulo, 03a] Ángulo U, J. Diseño Práctico de Aplicaciones. Primera Parte PIC16F84. Mc Graw Hill, 2003.

[Ángulo, 03b] Ángulo J, Romero S, Ángulo I. Diseño Práctico de Aplicaciones. Segunda Parte PIC16F87x. Mc Graw Hill, 2003.

[Duque, 1997] Duque, Edison. Curso Básico de Microcontroladores PIC. Cedit, Compañía Editorial Electrónica. Colombia, 1997.

[Mandado, 07] Mandado E., Menéndez L., Fernández L., López E. Microcontroladores PIC. Sistema integrado para el autoaprendizaje. Marcombo Ediciones Técnicas, S.A. España, 2007.

[Sánchez, 2007] Eduardo Sánchez A, MsC., Curso de Microcontroladores I, [Online] W.W.W.:

<http://www.itnogales.edu.mx/Microprocesadores/%201/Apuntes-%20Micros/Curso%20Microcontroladores.pdf>

[Valdés, 07] Fernando Valdés P, Ramon Pallás Areny. Microcontroladores: Fundamentos y Aplicaciones con PIC. Marcombo Ediciones Técnicas, S.A. España, 2007.

[Valdivieso, 09] Carlos E. Valdivieso A. Sistema Modular para el Laboratorio de Microcontroladores de la ESPOL. Escuela Superior Politécnica, 2009.