



UNIVERSIDAD NACIONAL DE LOJA

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

CARRERA DE TECNOLOGÍA EN
ELECTRICIDAD Y CONTROL INDUSTRIAL

TEMA:

“SIMULACIÓN DE MICROCONTROLADORES, (PIC 16F877A) CON LA
APLICACIÓN DEL SOFTWARE MICROCODE STUDIO, PROTEUS, EN EL
LABORATORIO DE ELECTRONICA”

INFORME TÉCNICO PREVIO ALGRADO
DE TECNÓLOGO EN ELECTRICIDAD Y CONTROL
INDUSTRIAL

AUTOR:

Pedro Clemente Martínez Jiménez

DIRECTOR:

ING. Luis Alberto Yunga Herrera Mg. Sc.

LOJA - ECUADOR

2012



CERTIFICACIÓN

Ing. Luis Alberto Yunga Herrera Mg. Sc.

DOCENTE DEL ÁREA DE LA ENERGÍA, LAS INDUSTRIAS Y LOS RECURSOS NATURALES NO RENOVABLES DE LA UNIVERSIDAD NACIONAL DE LOJA; Y DIRECTOR DEL INFORME TÉCNICO.

CERTIFICA:

Que el trabajo de investigación titulado **“SIMULACIÓN DE MICROCONTROLADORES, (PIC 16F877A) CON LA APLICACIÓN DEL SOFTWARE MICROCODE STUDIO,PROTEUS, EN EL LABORATORIO DE ELECTRONICA”**, desarrollado por el señor Pedro Clemente Martínez Jiménez, previo a optar el grado de ***Tecnólogo en Electricidad y Control Industrial*** ha sido realizado bajo mi dirección, mismo que cumple con los requisitos de grado exigidos en las Normas de graduación, por lo que autorizo su presentación ante el tribunal de grado.

Loja, Julio del 2012

Ing. Luis Alberto Yunga Herrera Mg. Sc.
DIRECTOR DEL INFORME TÉCNICO.

AUTORÍA

Todos los conceptos, opiniones, ideas, cálculos y resultado vertido en el siguiente trabajo de investigación son de absoluta responsabilidad del autor.

PEDRO C. MARTÍNEZ J.

AUTOR

AGRADECIMIENTO

Agradezco a la Universidad Nacional de Loja, a los docentes de la carrera de Tecnología Eléctrica y Control Industrial, quienes formaron parte integral en el desarrollo personal e intelectual de mi vida profesional.

Así como también de manera especial al Ing. Luis Alberto Yunga Herrera Mg. Sc., por el apoyo y la guía continúa brindada durante la ejecución y desarrollo del presente trabajo investigativo.

Y como olvidarme de darle gracias a mi querido DIOS y a mi familia, por darme las fuerzas necesarias para poder levantarme siempre y seguir adelante con la misma Fe y dedicación, para convertirme en un hombre de bien y un profesional éticamente responsable y respetuoso.

A ellos y a todas las personas que han colaborado directa e indirectamente para culminar mi formación universitaria, mis más sinceros agradecimientos y respeto.

PEDRO C. MARTÍNEZ J.



DEDICATORIA

El presente trabajo representa el esfuerzo, dedicación y trabajo duro que día a día mis padres, hermanos y familia entera han inculcado y hecho parte de mí, siendo para todos ellos dedicado la presente.

PEDRO CLEMENTE MARTÍNEZ JIMÉNEZ

RESUMEN

Con el afán de que las generaciones futuras de compañeros estudiantes, puedan tener una guía básica que les permita desarrollar proyectos con Microcontroladores, en la presente se plantean prácticas que contienen y mantienen, un nivel de complejidad que va desde el más fácil hasta un avanzado, con la respectiva guía paso a paso, para que puedan ser implementadas de manera correcta y adecuada, sin ningún tipo de errores.

En el desarrollo e implementación de las prácticas planteadas se hará uso del PIC 16F877A, por la gran cantidad de pines programables que posee y su robustez ante cortos.

Microcode Studio es el software de programación, que se utilizará para el desarrollo de los distintos códigos de las prácticas planteadas, por su facilidad y compatibilidad con el software de simulación de circuitos ISIS de Proteus Profesional.

ISIS de Proteus Profesional es un simulador de circuitos analógicos y digitales, que nos permite analizar el correcto desempeño y la corrección de errores de un circuito antes de su implementación práctica.

La implementación de las prácticas planteadas en el PIC 16F877A, se hará a través del software PICKit 2, que permite programar en cualquier PIC; las líneas de código desarrolladas en los programas de Microcode.

Con el afán de verificar el funcionamiento real de cada práctica planteada, se diseñó un MÓDULO, que contiene todos y cada uno de los elementos requeridos, para nuestros o cualquier proyecto que se desee implementar.

SUMMARY

With the desire that the future generations of partners students, they can have a basic guide that allows them to develop projects with Microcontroller's, in the present they think about practical that they contain and they maintain, a level of complexity that goes from the easiest to an advanced one, with the respective guide step to step, so that they can be to implement them in a correct and appropriate way, without any type of errors.

In the development and implementation of the outlined practices will be made use of the PIC 16F877A, for the great quantity of programmable pines that possesses and its robustness before short.

Microcode Studio is the programming software that will be used for the development of the different codes of the outlined practices, for its easiness and compatibility with the software of simulation of circuits ISIS of Professional Proteus.

ISIS of Professional Proteus is a pretender of analogical and digital circuits that allows us to analyze the correct acting and the correction of errors of a circuit before its implementation he/she practices.

The implementation of the practices outlined in the PIC 16F877A, will be made through the software PICKit 2 that it allows to program in any PIC; the code lines developed in the programs of Microcode.

With the desire of verifying the real operation of each outlined practice, a MODULE was designed that contains all and each one of the required elements, for our or any project that it is wanted to implement.

ÍNDICE GENERAL

PORTADA.....	i
CERTIFICACIÓN	ii
AUTORÍA	iii
AGRADECIMIENTO	iv
DEDICATORIA.....	v
RESUMEN.....	vi
SUMMARY.....	vii
ÍNDICE GENERAL.....	viii
I. INTRODUCCIÓN.....	2
II. DESCRIPCIÓN TECNICA Y UTILIDAD	4
2.1. CONTROLADORES Y MICROCONTROLADORES.....	5
2.1.1. Microcontrolador.....	5
2.1.2. Diferenciasentre Microprocesador y Microcontrolador.	6
2.1.3. Ventajasde un Microcontrolador frente a un Microprocesador.	7
2.1.4. Los Microcontroladores Hoy Día.....	8
2.1.5. Tipos de Arquitecturas de Microcontroladores	9
2.1.5. CaracterísticasRelevantesde los PIC	11
2.1.6. Microcontroladores Disponibles.....	12
2.1.7. MicrocontroladorPIC 16F877A.....	12
2.2. DISPOSITIVOS ELECTRÓNICOS.....	18
2.2.1. Capacitores.....	18
2.2.2. Resistores	21
2.2.3. Relé, Relay - Relevador.....	24
2.2.4. Pulsador	25
2.2.5. Cristales de Cuarzo.....	26
2.2.6. Teclado Matricial 4X4	28
2.2.7. Displaysde 7 Segmentos.....	30
2.2.8. Módulo LCD.....	33
2.2.9. Matriz de Leds	35
2.3. SOFTWARE MICROCODE STUDIO.....	37
2.3.1. Características Microcode Studio	37

2.3.2.	PIC que Soporta Microcode Studio	38
2.3.3.	Configuracionesdel Hardware.....	38
2.4.	SOFTWARE PROTEUS PROFESIONAL	39
2.4.1.	Descripción	39
2.4.2.	Aspectos Básicos	40
2.4.3.	Características.....	40
2.4.4.	Elementos	41
2.4.5.	ISIS.....	41
2.4.5.1.	HerramientaGráficapara la Gestiónde los Encapsulados.....	43
2.4.5.2.	Completa Gestión de Diseños Jerárquicos.....	44
2.4.5.3.	UtilizacióndeBuses	44
2.4.5.4.	Administrador de las Propiedades de los Componentes.....	45
2.5.	PICKIT2 CLONE VERSIÓN 2.61 PARA PUERTO USB	46
2.5.1.	Pasos para la descarga e instalación de PICKit2 v2.61.....	46
2.6.	FUENTE DE ALIMENTACIÓN DEL MÓDULO DE PRÁCTICAS.....	52
2.6.1.	Estructura Básica	53
2.6.2.	Conexión Red Eléctrica	53
2.6.3.	Fusible.....	54
2.6.4.	Filtro de Red	54
2.6.5.	Transformador	55
2.6.6.	Rectificador	59
2.6.7.	Filtro.....	61
2.6.8.	Regulador de Tensión.....	62
III.	MATERIALES	69
3.1.	MATERIALES DEL MÓDULO	70
IV.	PROCESO METODOLÓGICO EMPLEADO	72
4.1.	PROCESO METODOLÓGICO.....	73
4.2.	MICROCODE STUDIO.....	73
4.2.1.	Configuraciónde Microcode Studio (IDE).....	73
4.3.	ISIS.....	75
4.3.1.	Selección de un Componente	75
4.3.2.	Títulos y Etiquetado de Componentes	77
4.3.3.	Selección de Instrumentos de Medida	78
4.3.4.	SeleccióndeFuentesde Alimentación	79



4.4.	PICKIT 2 VERSIÓN 2.61.....	80
4.4.1.	Pasos para la programación con el PICKit2 Clone.....	80
4.5.	MÓDULO DE PRÁCTICAS PARA MICROCONTROLADORES.....	83
4.5.1.	DiseñoPCBde la Fuente de Alimentación.....	83
4.5.2.	Diseño del Módulo de Prácticas.....	84
4.1.3.	Diseño PCB del Módulo de Prácticas.....	85
V.	RESULTADOS.....	89
5.1.	PRÁCTICASPROPUESTAS PARA EL MÓDULO.....	90
	PRACTICA 1.....	91
	<i>Parpadeo de Leds en Intervalos de un Segundo.....</i>	91
	PRACTICA 2.....	99
	<i>Desplazamiento de Leds Tipo Auto Fantástico.....</i>	99
	PRACTICA 3.....	109
	<i>Semáforode Tres Intersecciones.....</i>	109
	PRACTICA 4.....	118
	<i>Parpadeo Alternado de Leds con Pulsadores.....</i>	118
	PRACTICA 5.....	127
	<i>Encender un Foco de 110V-AC con un Pulsador.....</i>	127
	PRACTICA 6.....	136
	<i>Decodificación, Arreglo y Manejo de Displays.....</i>	136
	PRACTICA 7.....	146
	<i>Manejo de Elementos Externos de Audio (Parlante de 8Ω).....</i>	146
	PRACTICA 8.....	154
	<i>DECODIFICACIÓN Y MANEJO DE UNA LCD.....</i>	154
	PRACTICA 9.....	162
	<i>Decodificacióny Manejode una Matriz de Leds de 5X7.....</i>	162
	PRACTICA 10.....	171
	<i>Decodificación y Manejode un Teclado 4x4 con Relés de 110V.....</i>	171
VI.	CONCLUSIONES.....	180
6.1.	CONCLUSIONES:.....	181
VII.	RECOMENDACIONES.....	182
7.1.	RECOMENDACIONES.....	183
VIII.	BIBLIOGRAFÍA.....	184
8.1.	BIBLIOGRAFÍA:.....	185

IX. ANEXOS.....	187
ANEXO A	188
CÓDIGOS DE ACCESO RÁPIDO PARA COMPONENTES DE ISIS	188
ANEXO B	189
DECLARACIONES DE MICROCODE.....	189
ANEXO C.....	192
FOTOS DE LAS PRÁCTICAS IMPLEMENTADAS.....	192
PRÁCTICA 1:.....	192
PRÁCTICA 2:.....	192
PRÁCTICA 3:.....	193
PRÁCTICA 4:.....	193
PRÁCTICA 5:.....	194
PRÁCTICA 6:.....	194
PRÁCTICA 7:.....	195
PRÁCTICA 8:.....	195
PRÁCTICA 9:.....	196
PRÁCTICA 10:.....	196

**“SIMULACIÓN DE MICROCONTROLADORES, (PIC
16F877A) CON LA APLICACIÓN DEL SOTFWARE
MICROCODE STUDIO,PROTEUS, EN EL
LABORATORIO DE ELECTRONICA”**

I. INTRODUCCIÓN

Un microcontrolador es todo un “sistema mínimo” dentro de un solo dispositivo lo cual ofrece enormes beneficios tecnológicos y un enorme panorama hacia el mundo de la compatibilidad. En estos circuitos nosotros los usuarios podemos programar en un solo “chip”, para desarrollar e implementar varias aplicaciones y funciones con menos componentes que la circueteria tradicional de compuertas, optimizando mucho espacio físico de los mismos, son mas económicos y sin la limitación de que un cambio de lógica de el usuario, ocasione un cambio en el diseño del hardware, ya que los cambios ampliaciones futuras del sistema son casi nulos(solo se requieren cambios en el software).

Los microcontroladores PIC, pueden ser programados tanto en lenguaje ensamblador como en Basic y principalmente en lenguaje C, para el que existen numerosos compiladores como es el caso del software de programación Microcode Studio.

En la actualidad podemos encontrar algunas aplicaciones de los microcontroladores PIC 16F877A, desarrollados para controlar diferentes dispositivos electrónicos, dispositivos eléctricos, manejo de sistemas como base de datos, donde se puede acceder de forma local o remota. Por lo que es conveniente implementar aplicaciones que permitan realizar tareas difíciles para los seres humanos.

En nuestro país el tema de los microcontroladores esta dando buenos resultados,estando precentes en nuestra casa y nuestra vida, como por

ejemplo controlando el funcionamiento de los mouse y teclados de la computadora, en los hornos, en los microondas y los televisores, etc.

Razón por la que se creyó conveniente la implementación a través de software de 10 proyectos, con el afán de que las generaciones futuras de compañeros estudiantes, puedan tener una guía básica que les permita con mayor facilidad:

- Diseñar circuitos en ISIS de PROTEUS.
- Diseñar algoritmos de programación.
- Programar en MicroCode Studio.
- Utilización de la interfaz entre ISIS de PROTEUS y MicroCode Studio.
- Lectura de Datasheets.

Permitiéndoles de esta manera estar en la capacidad de realizar cualquier tipo de proyecto que se propongan con la utilización del microcontrolador PIC 16F877A, u otro de la misma familia.

Con el entusiasmo de que los compañeros estudiantes puedan comprobar físicamente el funcionamiento de los proyectos propuestos en la presente investigación, adicionalmente se diseñó un hardware (Módulo de prácticas). Y en éste no solo podrán verificar los presentes proyectos sino también implementar cualquier otro proyecto según las necesidades que requieran.

II. DESCRIPCIÓN TÉCNICA Y UTILIDAD

2.1. CONTROLADORES Y MICROCONTROLADORES

Un controlador es un dispositivo electrónico encargado de controlar uno o varios procesos. Por ejemplo, el controlador del aire acondicionado, recogerá la información de los sensores de temperatura, la procesará y actuará en consecuencia.

Al principio, los controladores estaban formados exclusivamente por componentes discretos. Más tarde, se emplearon procesadores rodeados de memorias, circuitos de E/S, sobre una placa de circuito impreso (PCB).

En la actualidad, los controladores integran todos los dispositivos antes mencionados en un pequeño chip. Esto es lo que hoy conocemos con el nombre de microcontrolador.

2.1.1. Microcontrolador

Un microcontrolador es un dispositivo electrónico capaz de llevar a cabo procesos lógicos. Estos procesos o acciones son programados en lenguaje ensamblador por el usuario, y son introducidos en este a través de un programador.

Inicialmente cuando no existían los microprocesadores las personas se ingeniaban en diseñar sus circuitos electrónicos y los resultados estaban expresados en diseños que implicaban muchos componentes electrónicos y cálculos matemáticos. Un circuito lógico básico requería de muchos elementos electrónicos basados en transistores, resistencias, etc., lo cual desembocaba en circuitos con muchos ajustes y fallos; pero en el año 1971 apareció el primer microprocesador el cual originó un cambio decisivo en las técnicas de diseño

de la mayoría de los equipos. Al principio se creía que el manejo de un microprocesador era para aquellas personas con un coeficiente intelectual muy alto; Por lo contrario con la aparición de este circuito integrado todo sería mucho más fácil de entender y los diseños electrónicos serian mucho más pequeños y simplificados. Entre los microprocesadores más conocidos tenemos el popular Z-80 y el 8085. Los diseñadores de equipos electrónicos ahora tenían equipos que podían realizar mayor cantidad de tareas en menos tiempo y su tamaño se redujo considerablemente; Sin embargo, después de cierto tiempo aparece una nueva tecnología llamada microcontrolador que simplifica aunmás el diseño electrónico.

2.1.2. Diferencias entre Microprocesador y Microcontrolador.

Si se ha tenido la oportunidad de realizar un diseño con un microprocesador se puede observar que dependiendo del circuito se requerían algunos circuitos integrados adicionales además del microprocesador como por ejemplo: memorias RAM para almacenar los datos temporalmente y memorias ROM para almacenar el programa que se encargaría del proceso del equipo, un circuito integrado para los puertos de entrada y salida, y finalmente un decodificador de direcciones.

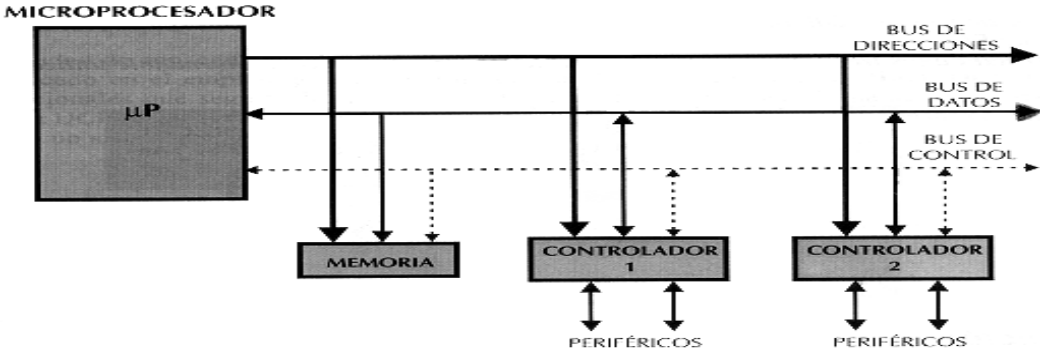


Figura 2.1. Estructura de un sistema abierto basado en un microprocesador

Un microcontrolador es un solo circuito integrado que contiene todos los elementos electrónicos que se utilizaban para hacer funcionar un sistema basado con un microprocesador; Es decir contiene en un solo integrado la Unidad de Proceso, la memoria RAM, memoria ROM, puertos de entrada, salidas y otros periféricos, con la consiguiente reducción de espacio.

El microcontrolador es en definitiva un circuito integrado que incluye todos los componentes de un computador. Debido a su reducido tamaño es posible montar el controlador en el propio dispositivo al que gobierna.

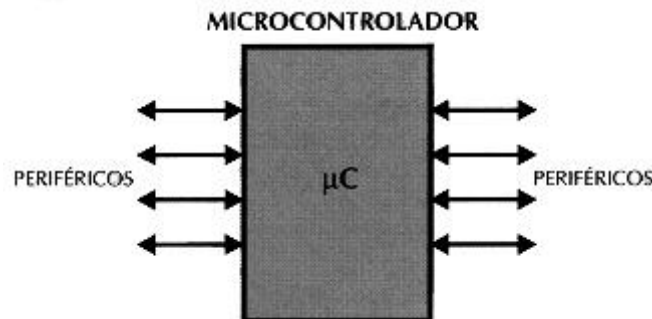


Figura 2.2. El Microcontrolador es un sistema cerrado.

2.1.3. Ventajas de un Microcontrolador frente a un Microprocesador.

Estas ventajas son reconocidas inmediatamente para aquellas personas que han trabajado con los microprocesadores y después pasaron a trabajar con los microcontroladores. Estas son las diferencias más importantes:

- La configuración mínima básica de un microprocesador estaba constituida por un Micro de 40 Pines, Una memoria RAM de 28 Pines, una memoria ROM de 28 Pines y un decodificador de direcciones de 18 pines.
- Un microcontrolador incluye todos estos elementos en un solo Circuito Integrado por lo que implica una gran ventaja en varios factores: En el

circuito impreso por su amplia simplificación de circuitería, el costo para un sistema basado en microcontrolador es mucho menor y, lo mejor de todo, el tiempo de desarrollo de su proyecto electrónico se disminuye considerablemente.

2.1.4. Los Microcontroladores Hoy Día

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

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

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

Los microcontroladores están siendo empleados en multitud de sistemas presentes en nuestra vida diaria, como pueden ser juguetes, horno microondas, frigoríficos, televisores, computadoras, impresoras, módems, el sistema de arranque de nuestro vehículo, etc. Y otras aplicaciones con las que seguramente no estaremos tan familiarizados como instrumentación

electrónica, control de sistemas en una nave espacial, etc. Una aplicación típica podría emplear varios microcontroladores para controlar pequeñas partes del sistema. Estos pequeños controladores podrían comunicarse entre ellos y con un procesador central, probablemente más potente, para compartir la información y coordinar sus acciones, como, de hecho, ocurre ya habitualmente en cualquier PC.

2.1.5. Tipos de Arquitecturas de Microcontroladores

Arquitectura Von Neumann

La arquitectura tradicional de computadoras y microprocesadores está basada en la arquitectura Von Neumann, en la cual la unidad central de proceso (CPU), está conectada a una memoria única donde se guardan las instrucciones del programa y los datos.

El tamaño de la unidad de datos o instrucciones está fijado por el ancho del bus que comunica la memoria con la CPU. Así un microprocesador de 8 bits con un bus de 8 bits, tendrá que manejar datos e instrucciones de una o más unidades de 8 bits (bytes) de longitud. Si tiene que acceder a una instrucción o dato de más de un byte de longitud, tendrá que realizar más de un acceso a la memoria.

Y el tener un único bus hace que el microprocesador sea más lento en su respuesta, ya que no puede buscar en memoria una nueva instrucción mientras no finalicen las transferencias de datos de la instrucción anterior.

Resumiendo todo lo anterior, las principales limitaciones que nos encontramos con la arquitectura Von Neumann son:

- La limitación de la longitud de las instrucciones por el bus de datos, que hace que el microprocesador tenga que realizar varios accesos a memoria para buscar instrucciones complejas.
- La limitación de la velocidad de operación a causa del bus único para datos e instrucciones que no deja acceder simultáneamente a unos y otras, lo cual impide superponer ambos tiempos de acceso.

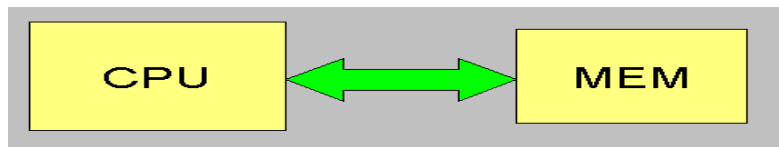


Figura 2.3.Arquitectura Von Neumann

Arquitectura Harvard

La arquitectura Harvard tiene la unidad central de proceso (CPU) conectada a dos memorias (una con las instrucciones y otra con los datos) por medio de dos buses diferentes.

Una de las memorias contiene solamente las instrucciones del programa (Memoria de Programa), y la otra, sólo almacena datos (Memoria de Datos).

Ambos buses son totalmente independientes y pueden ser de distintos anchos. El set de instrucciones y el bus de memoria de programa pueden diseñarse de tal manera que todas las instrucciones tengan una sola posición de memoria de programa de longitud.

Además, al ser los buses independientes, la CPU puede acceder a los datos para completar la ejecución de una instrucción, y al mismo tiempo leer la siguiente instrucción a ejecutar.

- El tamaño de las instrucciones no está relacionado con el de los datos, y por lo tanto puede ser optimizado para que cualquier instrucción ocupe una sola posición de memoria de programa, logrando así mayor velocidad y menor longitud de programa.
- El tiempo de acceso a las instrucciones puede superponerse con el de los datos, logrando una mayor velocidad en cada operación.

Una pequeña desventaja de los procesadores con arquitectura Harvard, es que deben poseer instrucciones especiales para acceder a tablas de valores constantes que pueda ser necesario incluir en los programas, ya que estas tablas se encontrarán físicamente en la memoria de programa (por ejemplo en la EPROM de un microprocesador).

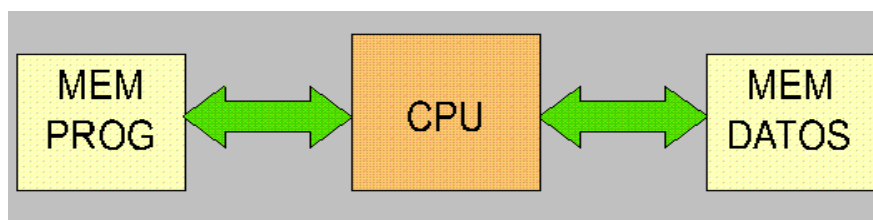


Figura 2.4. Arquitectura Harvard

El microcontrolador PIC 16F84 posee arquitectura Harvard, con una memoria de datos de 8 bits, y una memoria de programa de 14 bits.

2.1.5. Características Relevantes de los PIC

- La arquitectura del procesador sigue el modelo Harvard.
- Se aplica la técnica de segmentación ("pipe-line") en la ejecución de las instrucciones.
- El formato de todas las instrucciones tiene la misma longitud.

- Procesador RISC (Computador de Juego de Instrucciones Reducido).
- Todas las instrucciones son ortogonales.
- Arquitectura basada en un banco de registros.

Prácticamente todos los PIC se caracterizan por poseer unos mismos, recursos mínimos.

- Modelos de arquitectura cerrada y de arquitectura abierta.
- Diversidad de modelos de microcontroladores.
- Amplio margen de alimentación y corrientes de salida elevadas.
- Herramientas de soporte potentes y económicas.

2.1.6. Microcontroladores Disponibles

Los microcontroladores más utilizados para el desarrollo de aplicaciones electrónicas son:

- PIC 16F628A
- PIC 16F877A

2.1.7. Microcontrolador PIC 16F877A

- 14 KBytes de memoria Flash
- 368 bytes de RAM interna
- 2 Canales PWM de 10 bits
- 256 bytes de EEPROM
- 33 Líneas I/O Programables
- 8 canales ADC de bits
- 2 temporizador/contador de 8 bits
- 1 temporizador/contador de 16 bits

Características del PIC 16F877A

Tabla 2.1. Características del Pic 16F877A

CARACTERÍSTICAS	16F877A
Frecuencia máxima	DX-20MHz
Memoria de programa flash palabra de 14 bits	8KB
Posiciones RAM de datos	368
Posiciones EEPROM de datos	256
Puertos E/S	A, B, C, D, E
Número de pines	40
Interrupciones	14
Timers	3
Módulos CCP	2
Comunicaciones Serie	MSSP, USART
Comunicaciones Paralelo	PSP
Líneas de entrada de CAD de 10 bits	8
Juego de instrucciones	35 Instrucciones
Longitud de instrucción	14 bits
Arquitectura	Harvard
CPU	Risc
Canales PWM	2
Pila Hardware	-

Descripción de puertos

Puerto A:

- Puertos de E/S de 6 pines
- RA0/ANO – PIN 2
- RA1/AN1 – PIN 3
- RA2/AN2 y Vref(-) – PIN 4
- RA3/AN3 y Vref(+) – PIN 5
- RA4(salida de colector abierto) y TOCKI(Entrada de reloj del módulo Timer0) – PIN 6
- RA5/AN4 y SS(Selección esclavo para el puerto serie síncrono) – PIN 7.

Puerto B:

- Puertos de E/S de 8 pines
- Resistencias pull-up programables
- RB0(Interrupción interna) – PIN 33
- RB4-7(Interrupción por cambio de flanco)
- RB4, RB7 y RB3(programación y debugger in circuit)

Puerto C:

- Puertos de E/S de 8 pines
- RC0/T1OSO(Timer1 salida oscilador) y T1CKI(Entrada de reloj del modulo Timer1) – PIN 15
- RC1-RC2(PWM/COMP/CAPT)
- RC1/T1OSI(Estrada osc timer1) – PIN 16
- RC3-4/IIC

- RC3-5/SPI
- RC6-7/USART

Puerto D:

- Puertos de E/S de 8 pines
- Bus de datos en PPS(Puerto paralelo esclavo)

Puerto E:

- Puertos de E/S de 3 pines
- RE0/AN5 y Read de PPS – PIN 8
- RE1/AN6 y Write de PPS – PIN 9
- RE2/AN7 y CS de PPS – PIN 10

El puerto serie usart

La USAR (Universal SyncheronusAsynchronous Receiver Transmitter) es uno de los dos periféricos contenidos en el PIC que nos permiten realizar comunicación en serie. El otro el MSSP (Master Synchronous serial port)

La USART, también conocida como SCI (Serial Communications Interface), puede configurarse como una unidad de comunicación en serie para la transmisión de datos asíncrona con dispositivos tales como terminales de computadora o bien para comunicación síncrona con dispositivos tales como convertidores A/D o D/A, circuitos integrados o memorias EEPROM con comunicación serie,etc. La gran mayoría de comunicación de datos digitales actuales utiliza la comunicación serie, debido a las grandes ventajas que representa esta manera de comunicar datos.

Dispositivos periféricos

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

Diagrama de bloques

El PIC 16F877A, utilizado en esta tesis cuenta con memoria de datos de 368 bytes y una memoria EEPROM de 256 bytes, con una memoria flash de 8k, cuenta con 5 puertos, este microcontrolador por su gran potencial permite realizar proyectos que necesiten gran consumo de memoria.

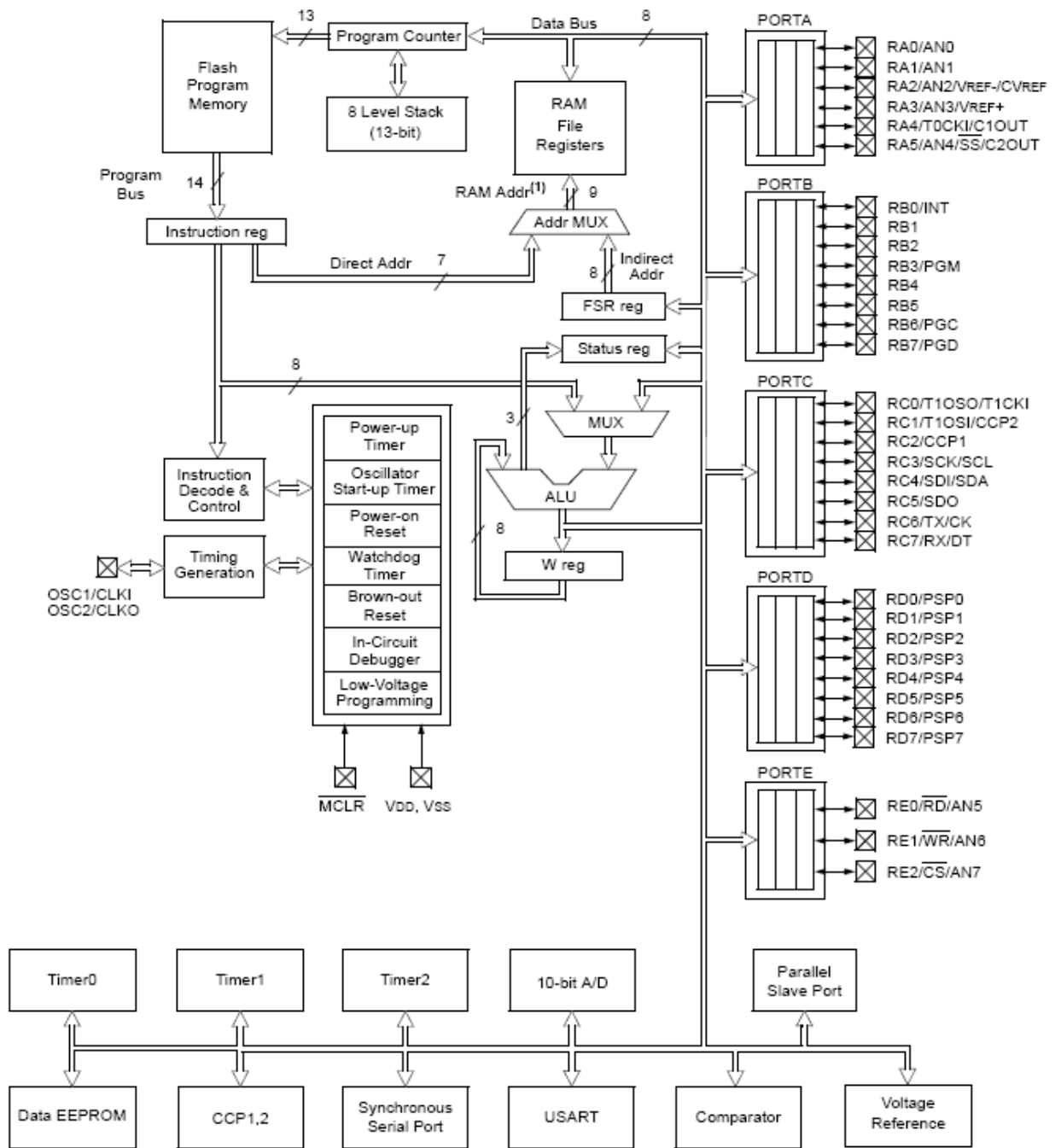


Figura 2.5. Diagrama de Bloques Pic 16F628A

Configuración de pines

La figura 2.6 muestra la rotulación o configuración de los pines, distribución de puertos, los pines se pueden utilizar de acuerdo a las necesidades que se requieran.

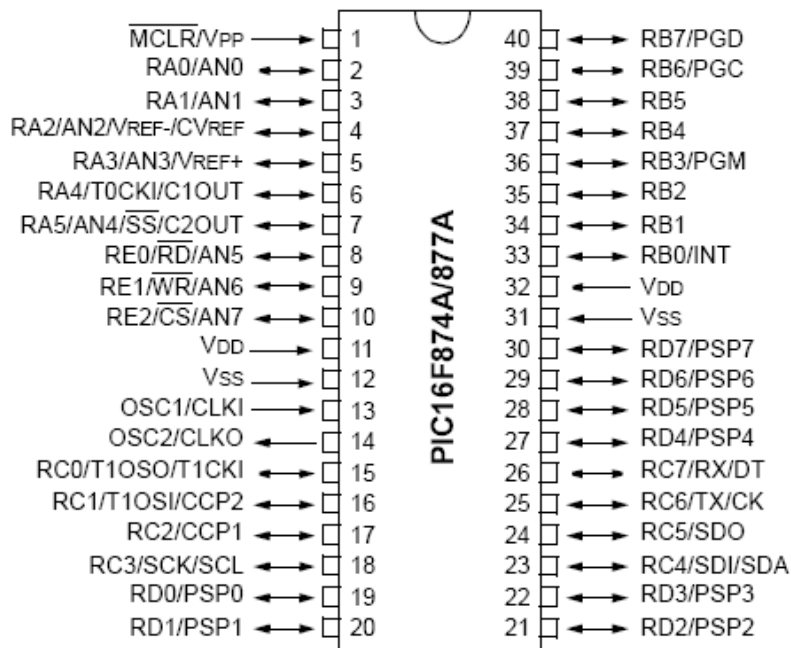


Figura 2.6. Configuración de Pines Pic 16F877A

2.2. DISPOSITIVOS ELECTRÓNICOS.

2.2.1. Capacitores

Estos capacitores tienen una capacidad fija determinada por el fabricante y su valor no se puede modificar. Sus características dependen principalmente del tipo de dieléctrico utilizado, de tal forma que los nombres de los diversos tipos se corresponden con los nombres del dieléctrico usado.

De esta forma podemos distinguir los siguientes tipos:

- Cerámicos.
- Plástico.
- Mica.
- Electrolíticos.
- De doble capa eléctrica.

Capacitores cerámicos

El dieléctrico utilizado por estos capacitores es la cerámica, siendo el material más utilizado el dióxido de titanio.

Capacitores electrolíticos

En estos capacitores una de las armaduras es de metal mientras que la otra está constituida por un conductor iónico o electrolito. Presentan unos altos valores capacitivos en relación al tamaño y en la mayoría de los casos aparecen polarizados.

Identificación de capacitores

Vamos a disponer de un código de colores, cuya lectura varía según el tipo de condensador, y un código de marcas, particularizado en los mismos. Primero determinaremos el tipo de condensador (fijo o variable) y el tipo concreto dentro de estos.

Las principales características que nos vamos a encontrar en los capacitores van a ser la capacidad nominal, tolerancia, tensión y coeficiente de temperatura, aunque dependiendo de cada tipo traerán unas características u otras.

- Capacitores cerámicos tipo placa, grupo 1 y 2.

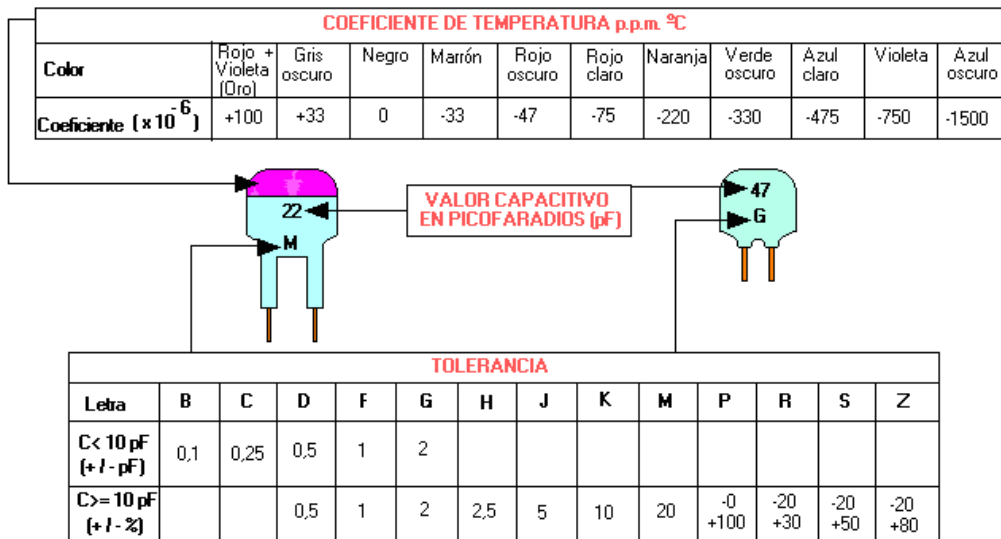


Figura 2.7. Capacitores cerámicos Grupo 1

➤ código de colores

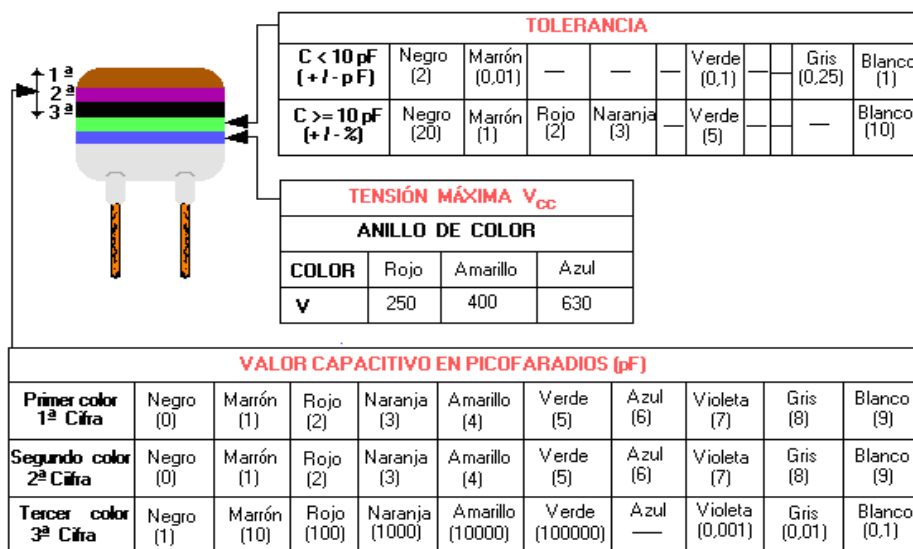


Figura 2.8. Capacitores cerámicos Grupo 2

➤ **Capacitores electrolíticos**

Estos capacitores siempre indican la capacidad en microfaradios y la máxima tensión de trabajo en voltios. Dependiendo del fabricante también

pueden venir indicados otros parámetros como la temperatura y la máxima frecuencia a la que pueden trabajar.

Las formas más usuales de indicación por parte de los fabricantes son las siguientes:



Figura 2.9. Capacitores electrolíticos

2.2.2. Resistores

Los resistores son componentes eléctricos pasivos en los que la tensión instantánea aplicada es proporcional a la intensidad de corriente que circula por ellos. Su unidad de medida es el ohmio (Ω).

Se pueden dividir en tres grupos:

- **Resistores lineales fijos:** su valor de resistencia es constante y está predeterminado por el fabricante.



Figura 2.10. Resistor fijo

- **Resistores variables:** su valor de resistencia puede variar dentro de unos límites.
- **Resistores no lineales:** su valor de resistencia varía de forma no lineal dependiendo de distintas magnitudes físicas (temperatura, luminosidad, etc.).

Resistores variables

Estos resistores pueden variar su valor dentro de unos límites. Para ello se les ha añadido un tercer terminal unido a un contacto móvil que puede desplazarse sobre el elemento resistivo proporcionando variaciones en el valor de la resistencia. Este tercer terminal puede tener un desplazamiento angular (giratorio) o longitudinal (deslizante). Según su función en el circuito éstas resistores se denominan:

- **Potenciómetros:** se aplican en circuitos donde la variación de resistencia la efectúa el usuario desde el exterior (controles de audio, video, etc.).
- **Trimmers, o resistores ajustables:** se diferencian de las anteriores en que su ajuste es definitivo en el circuito donde van aplicadas. Su acceso está limitado al personal técnico (controles de ganancia, polarización, etc.).

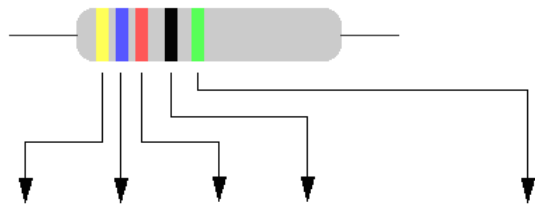
Identificación de resistores

En primer lugar se debe determinar el grupo al que pertenecen, es decir, si son lineales fijas, variables, o no lineales, y el tipo concreto al que pertenecen dentro de cada grupo.

➤ **Código de colores**

Es el código con el que se regula el marcado del valor nominal y tolerancia para resistores fijos de carbón y metálicas de capa fundamentalmente.

Tabla 2. *Código de colores para tres o cuatro bandas*



COLOR	1ª CIFRA	2ª CIFRA	3ª CIFRA	Nº DE CEROS	TOLERANCIA (+/-%)
PLATA	-	-	-	0,01	-
ORO	-	-	-	0,1	-
NEGRO	-	0	0	-	-
MARRÓN	1	1	1	0	1%
ROJO	2	2	2	00	2%
NARANJA	3	3	3	000	-
AMARILLO	4	4	4	0000	-
VERDE	5	5	5	00000	0,5%
AZUL	6	6	6	000000	-
VIOLETA	7	7	7	-	-
GRIS	8	8	8	-	-
BLANCO	9	9	9	-	-

Tolerancia: sin indicación +/- 20%

Para determinar el valor de la resistencia comenzaremos por determinar la banda de la tolerancia: oro, plata, rojo, marrón, o ningún color. Si las bandas son de color oro o plata, está claro que son las correspondientes a la tolerancia y debemos comenzar la lectura por el extremo contrario. Si son de color rojo o marrón, suelen estar separadas de las otras tres o cuatro bandas, y así comenzaremos la lectura por el extremo opuesto, 1ª cifra, 2ª cifra, número de ceros o factor multiplicador y tolerancia, aunque en algunos casos existe una tercera cifra significativa. **En caso de existir sólo tres bandas con color, la tolerancia será de +/- 20%.** La falta de esta banda dejará un hueco grande en uno de los extremos y se empezará la lectura por el contrario. Suele ser característico que la separación entre la banda de tolerancia y el factor multiplicativo sea mayor que la que existe entre las demás bandas.

2.2.3. Relé, Relay - Relevador

El **Relé** es un interruptor operado magnéticamente. El **relé** se activa o desactiva (dependiendo de la conexión) cuando el electroimán (que forma parte del **relé**) es energizado (le ponemos un voltaje para que funcione). Esta operación causa que exista conexión o no, entre dos o más terminales del dispositivo (el **relé**). Esta conexión se logra con la atracción o repulsión de un pequeño brazo, llamado armadura, por el electroimán. Este pequeño brazo conecta o desconecta los terminales antes mencionados.

Funcionamiento del Relé:

Si el electroimán está activo jala el brazo (armadura) y conecta los puntos C y D. Si el electroimán se desactiva, conecta los puntos D y E. De esta manera se puede conectar algo, cuando el electroimán está activo, y otra cosa conectada, cuando está inactivo. Es importante saber cual es la resistencia del

bobinado del electroimán (lo que está entre los terminales A y B) que activa el **relé** y con cuanto voltajeeste se activa. Este voltaje y esta resistencia nos informan que magnitud debe de tener la señal que activará el **relé** y cuanta corriente se debe suministrar a éste. La corrientese obtiene con ayuda de la Ley de Ohm: $I = V / R$.

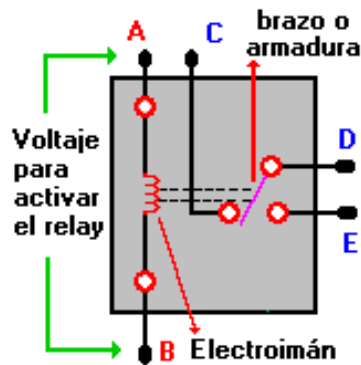


Figura 2.11. Estructura interna del relé

2.2.4. Pulsador

Un pulsador es simplemente un interruptor o switch cuya función es permitir o interrumpir el paso de la corriente eléctrica, a diferencia de un switch común, un pulsador solo realiza su trabajo mientras lo tengas presionado.

Existen pulsadores **NC** y **NA**, es decir **normalmente abiertos** y **normalmente cerrados**, cuando pulsas uno de tipo NC, se abre mientras lo presiones es decir no permite le pasode la corriente, y en un NA, cuando lo presionas permites el paso, es decir lo contrario, generalmente se usa para enviar pulsos o para activar algo. Como ejemplo el teclado de tu PC cada tecla es un pulsador, cuando presionas uno permite el paso de una señal eléctrica que le dice a la CPU que estas presionando esa tecla.

Pero en uso común un pulsador comercial para hacer algún proyecto de electrónica tiene este aspecto en varios tipos:

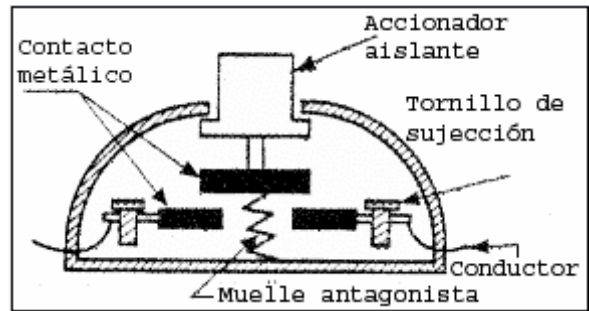


Figura 2.12. Estructura interna de un pulsador



Figura 2.13. Pulsadores en el mercado

2.2.5. Cristales de Cuarzo

Los cristales de cuarzo son ampliamente utilizados en los equipos que utilizan los radioaficionados. Basta dar una rápida ojeada al esquema de cualquiera de nuestros transceptores para encontrar varios de ellos.

Estos dispositivos están formados por una fina lámina de cuarzo situada entre dos electrodos. Como es sabido, el cuarzo, también llamado cristal de roca, es un mineral compuesto por silicio y oxígeno, (óxido anhídrido de silicio, bióxido de silicio o anhídrido silícico, SiO₂) cuyos cristales tienen forma de prisma hexagonal terminado por dos romboedros que parecen una bipirámide

hexagonal. El cuarzo es el mineral más difundido en la corteza terrestre, bien en forma de cristales o formando parte otras rocas, como el granito (cuarzo, feldespato y mica). En la figura número uno se puede ver un grupo de cristales de cuarzo tal como aparecen en la naturaleza.



Figura 2.14. Mineral de cuarzo

En la figura de abajo podremos observar dos de los distintos tipos de cristales de cuarzo empleados en comunicaciones. Este dispositivo es un oscilador completo, ya que basta aplicar una tensión de 5 voltios para que el dispositivo genere la señal, que en el caso del que se muestra en la figura, es de 5 MHz.

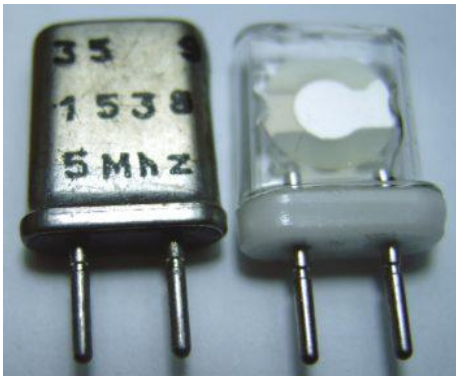


Figura 2.15. Cristal de cuarzo

En los cristales de cuarzo se produce un fenómeno llamado piezoelectricidad. Este fenómeno consiste en que la aplicación de una tensión

eléctrica produce una deformación del cristal, mientras que la deformación del cristal genera una tensión eléctrica. Esta característica se aprovecha en electrónica para producir tensiones alternas con una gran estabilidad de frecuencia ya que ésta depende directamente del grueso de la lámina de cuarzo.

Cómo es natural, un simple cristal de cuarzo no es capaz por si sólo de generar ninguna tensión alterna. Es preciso que el cristal de cuarzo forme parte de un circuito electrónico llamado oscilador que no es sino un amplificador donde una parte de la energía de salida, se realimenta con la misma fase a la entrada. A esto se le llama realimentación positiva.

Si en el camino de la realimentación positiva se encuentra un cristal de cuarzo, entonces el oscilador generará una tensión alterna cuya frecuencia vendrá determinada por las características mecánicas del cristal utilizado y tendrá la característica de que su frecuencia será sumamente estable y precisa. De ahí su empleo en equipos de telecomunicaciones.

Como ya se ha indicado, la frecuencia de oscilación de un cristal viene determinada por el grueso de la lámina de cuarzo y la dirección en que se dio el corte del cristal original para obtener la lámina.

2.2.6. Teclado Matricial 4X4

Dispositivo de entrada de datos que consta de 16 teclas o pulsadores, dispuestos e interconectados en filas y columnas. Dispone de un conector SIL (Single In Line) macho de 8 pines que se corresponden con las 4 filas y las cuatro columnas de las que dispone.



Figura 2.16. Teclado matricial 4x4

Como se ve en la figura las filas del teclado se conectan a las líneas RB0, RB1, RB2, RB3 configuradas como salidas mientras que las columnas se conectan a las líneas RB4, RB5, RB6, RB7 configuradas como entradas.

Para detectar si hay una tecla pulsada se aplica el siguiente procedimiento: Se hace un barrido por cada una de las filas, inicialmente se aplica a la primera fila un nivel bajo y a las otras tres un nivel alto si en este instante se pulsa alguna tecla de esta fila entonces el nivel bajo aparecerá en la columna en la que se encontraba la tecla, seguidamente se pasa a verificar la segunda fila enviando un nivel bajo a esta y a las otras un nivel alto y así sucesivamente. Verificando los niveles de los terminales de las columnas se podrá deducir si hay alguna tecla pulsada en la fila donde se esta aplicando el nivel bajo, considerando que si no hay alguna tecla pulsada en las entradas del PIC RB4, RB5, RB6, RB7 se encontrara un nivel alto.

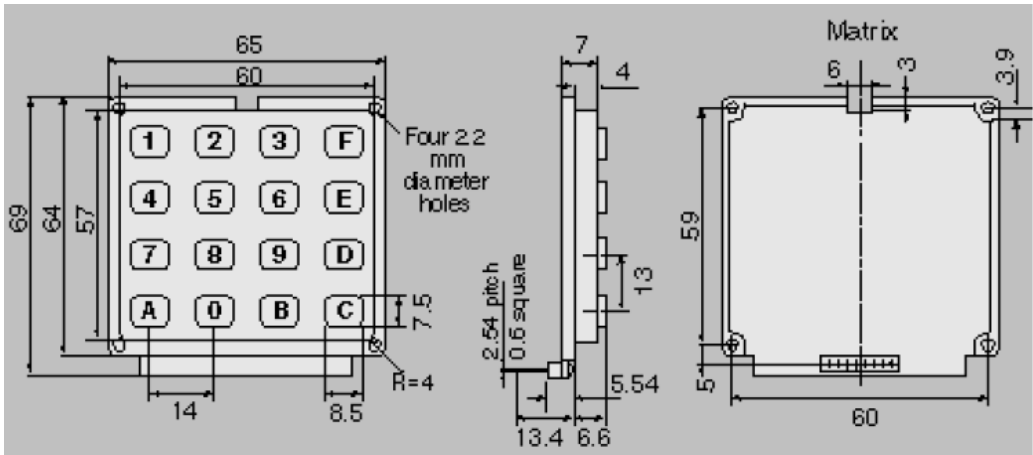


Figura 2.17. Estructura de un teclado matricial 4x4

2.2.7. Displays de 7 Segmentos

Muchos equipos electrónicos proporcionan información al usuario mediante la utilización de señales luminosas, como la emisora sintonizada en un equipo de radio o la lectura de tensión en un voltímetro digital.

Para representar las cifras numéricas se agrupan siete diodos en de segmentos. Estos diodos tienen conectados entre si todos los ánodos.

Un Display de este tipo está compuesto por siete u ocho leds de diferentes formas especiales y dispuestas sobre una base de manera que puedan representarse todos los símbolos numéricos y algunas letras. Los primeros siete segmentos son los encargados de formar el símbolo y con el octavo podemos encender y apagar el punto decimal.

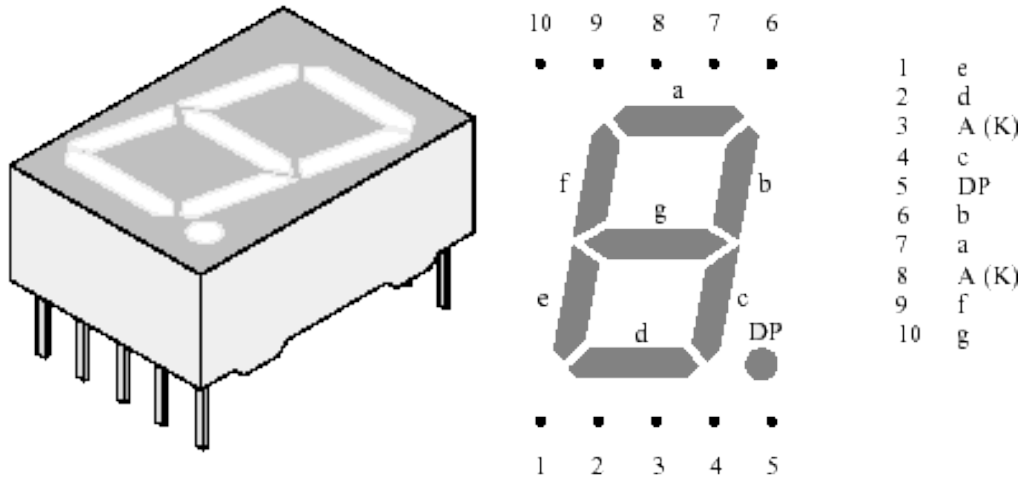


Figura 2.18. Denominación de los segmentos de Display Comercial

Esta es la denominación de los 7 segmentos en los modelos comerciales.

Esquema eléctrico del visualizador:

Se ha realizado el esquema de tres segmentos de los 7 que lo componen ya que el resto es idéntico.

Se ha optado por una configuración de ánodo común. La decisión ha sido totalmente aleatoria, por lo que serviría exactamente igual una configuración de cátodo común.

Como se puede deducir del esquema, el número de cables entre la placa controladora y el visualizador digital es de 8. (7 para los segmentos y 1 para el negativo la alimentación).

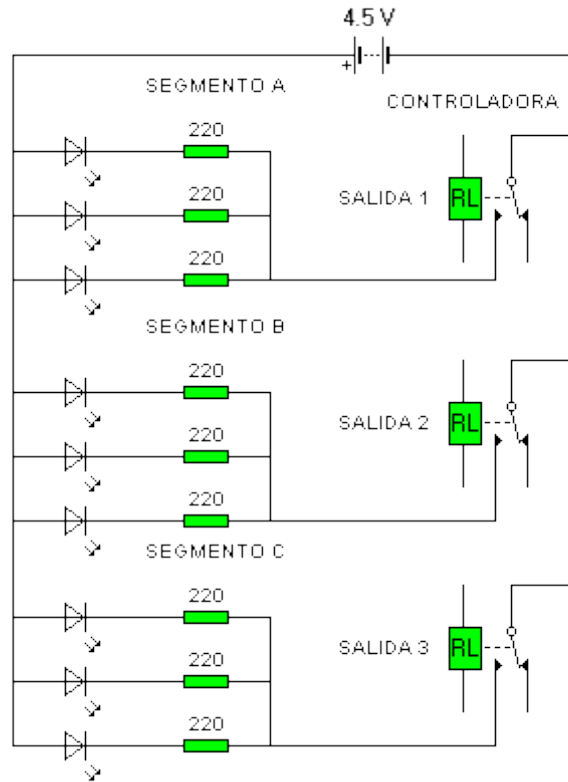


Figura 2.19. Denominación de los segmentos de Display

Características

- Bajo consumo
- Alto contraste
- Conector con pines
- Angulo de visión ancho
- Rápida respuesta

Una aplicación de los LEDS es el Display de 7 segmentos. Una de las aplicaciones más populares de los LEDS es la de señalización. Quizás la más utilizada sea la de 7 LEDS colocados en forma de ocho tal y como se indica en la figura.

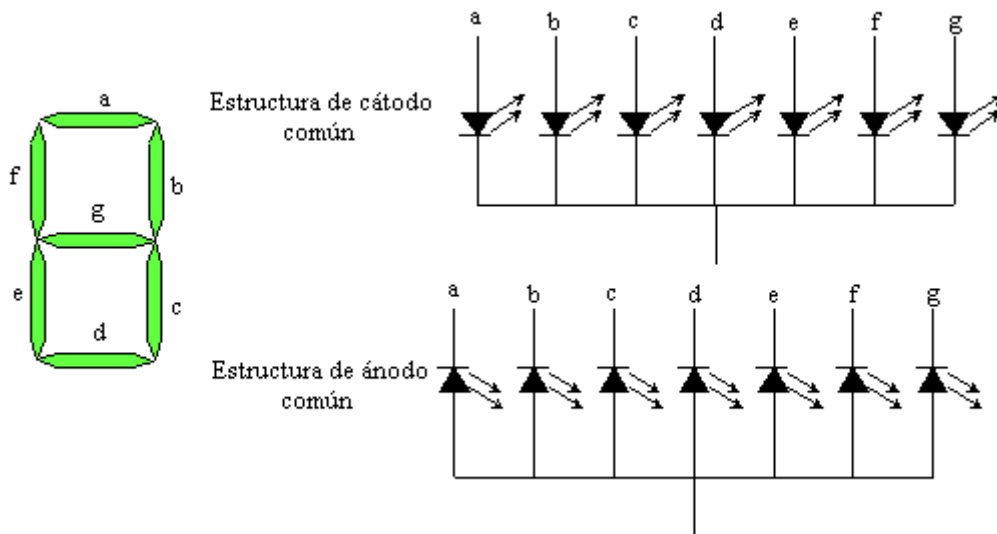


Figura 2.20. Estructura de un Display ánodo y cátodo común

2.2.8. Módulo LCD

Las pantallas de cristal líquido LCD o Display LCD para mensajes (*LiquidCrystalDisplay*) tienen la capacidad de mostrar cualquier carácter alfanumérico, permitiendo representar la información que genera cualquier equipo electrónico de una forma fácil y económica.

La pantalla consta de una matriz de caracteres (normalmente de 5x7 o 5x8 puntos) distribuidos en una, dos, tres o cuatro líneas de 16 hasta 40 caracteres cada línea.

El proceso de visualización es gobernado por un microcontrolador incorporado a la pantalla, siendo el Hitachi 44780 el modelo de controlador más utilizado.

El módulo LCD ejecuta automáticamente una secuencia de inicio interna en el instante de aplicarle la tensión de alimentación si se cumplen los requisitos de alimentación expuestos en su manual.

Dichos requisitos consisten en que el tiempo que tarde en estabilizarse latensión desde 0.2 V hasta los 4.5V mínimos necesario sea entre 0.1 ms y 10ms. Igualmente el tiempo de desconexión debe ser como mínimo de 1 msantes de volver a conectar.

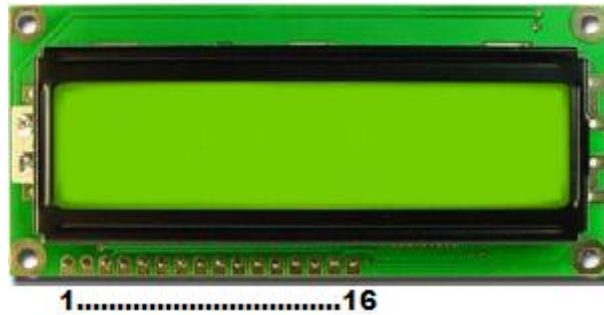


Figura 2.21. LCD 2x16: está compuesto por 2 líneas de 16 caracteres

El módulo LCD posee una zona de memoria RAM llamada DDRAM (Data Display RAM) donde se almacenan los caracteres que se van a mostrar en la pantalla.

Tiene una capacidad de 80 bytes, 40 por cada línea, de los cuales sólo 32 se pueden visualizar a la vez (16 bytes por línea).

El LCD dispone de una zona de memoria interna no volátil llamada CGROM donde se almacena una tabla con los 192 caracteres que pueden ser visualizados.

Cada uno de los caracteres tiene su representación binaria de 8 bits. Para visualizar un carácter debe recibir por el bus de datos el código correspondiente.

En la siguiente tabla podremos observar la descripción de cada Pin de conexión de la LCD.

Tabla 3: Descripción Pines de Conexión de una LCD 2x16

SEÑAL	DEFINICIÓN	PINES	FUNCIÓN
DB0...DB7	Bus de datos	7...14	bus de datos
E	Enable	6	E=0 , LCD no habilitado E=1 , LCD habilitado
R/W	Lectura/escritura	5	R/W=0, escribe en LCD R/W=1, lee del LCD
RS	Register Select	4	R/S=0 , modo comando R/S=1 , modo carácter
Vlc	Liquid Crystal driving Voltaje	3	Tensión para ajustar el contraste
Vdd	Power supply voltaje	2	Tensión de alimentación
Vss	Ground	1	Masa

2.2.9. Matriz de Leds

Una matriz de Leds consiste en un arreglo de Leds que pueden ser encendidos y apagados individualmente desde un microcontrolador. Pueden pensar en ella como una pantalla de pocos pixeles en los cuales pueden presentar gráficos y textos, tanto estáticos como en movimiento.

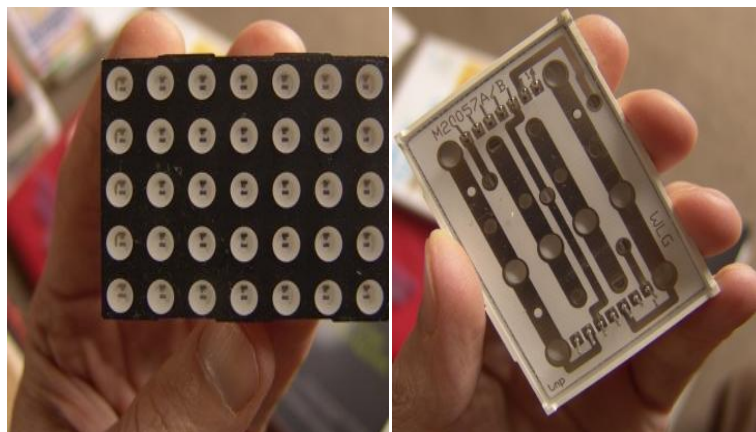


Figura 2.22. Forma real y pines de conexión de una matriz leds de 5x7

En la siguiente figura se presenta una matriz de 5x7 Leds, donde las columnas corresponden a los cátodos (deben ir conectados a tierra) y las filas corresponden a los ánodos (deben ir conectados a voltaje).

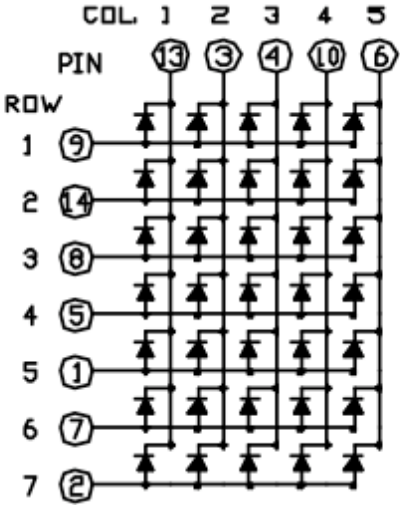


Figura 2.23. Estructura interna de conexión de una matriz leds de 5x7

Para encender un LED específico debe ubicarse la intersección entre la columna y la línea correspondiente. Por ejemplo, para encender el LED que se muestra en la figura siguiente, la columna 3 (C3) debe ser conectada a tierra (a través de una resistencia de 220 ohm) mientras que la fila 5 (R5) es conectada a voltaje (5v).

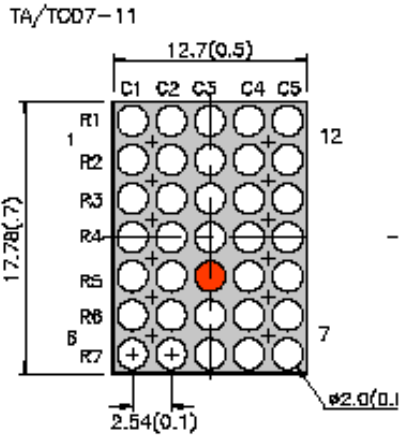


Figura 2.24. Como encender un led de una matriz de 5x7

2.3. SOFTWARE MICROCODE STUDIO.

Microcode Studio es un interface en el cual se puede escribir el código del programa y el corrige los errores de sintaxis, ordena visualmente las subrutinas. El Microcode queda enlazado con PIC BASIC y el IC-PROG, de manera que una vez que se termina el programa compilamos y generamos el archivo. Los programas se los guarda en formato Pic-basic BAS.

El explorador de código nos permite automáticamente saltar a incluir los archivos, definen: constantes variables altas y modificadoras, los símbolos y las etiquetas que están contenidos dentro de su código fuente. Permite cortar, copiar pegar y deshacer, se proporciona junto con las características de búsqueda y remplazo.

Microcode Studio incluye también EasyHIDWisard, una herramienta de generación de código que permite al usuario implementar rápidamente la comunicación bidireccional entre un microcontrolador integrado y el computador.

2.3.1. Características Microcode Studio

- Completo resultado de sintaxis de su código de fuente.
- Saltar rápidamente para incluir, los archivos los símbolos, definir las variables y las etiquetas utilizando la ventana de explorador de código.
- Identificar y corregir la compilación y el ensamblador de errores.
- Ver la salida de serie del microcontrolador.
- Palabra clave basada en el contexto de ayuda sensible al soporte para MPSAM.

- Es fácil de configurar su compilador, ensamblador y las opciones de programador o se puede dejar que Microcode Studio lo haga.
- Compilación y errores en ensamblador puede ser fácilmente identificada y corregida mediante la ventana de error.
- Para corregir el error simplemente hay que hacer clic en un error de compilación y Microcode Studio lo llevara automáticamente a la línea de error.
- Microcode Studio viene incluido también con una ventana de comunicaciones que nos permite depurar y ver la salida de serie del microcontrolador.

2.3.2. PIC que Soporta Microcode Studio

Microcode Studio soporta actualmente los siguientes dispositivos de microcontroladores PIC de las diferentes gamas existentes:

16F627(A), 16F628(A), 16F73, 16F74, 16F76, 16F77, 16F870, 16F871, 16F873(A), 16F874(A), 16F876(A), 16F877(A), 16F87, 16F88, 18F242, 18F248, 18F252, 18F258, 18F442, 18F448, 18F452, 18F458, 18F1220, 18F1320, 18F220, 18F2320, 18F4220, 18F4320, 18F6620, 18F8620, 18F2220.

2.3.3. Configuraciones del Hardware.

MicrocodeStudio ICD no requiere ningún hardware especial para albergar el microcontrolador. Sin embargo se necesita una tarjeta de acogida que soporta la comunicación serial RS232 O USB. Por ejemplo un programador de PIC que es ideal es la comunicación entre PIC y computador.

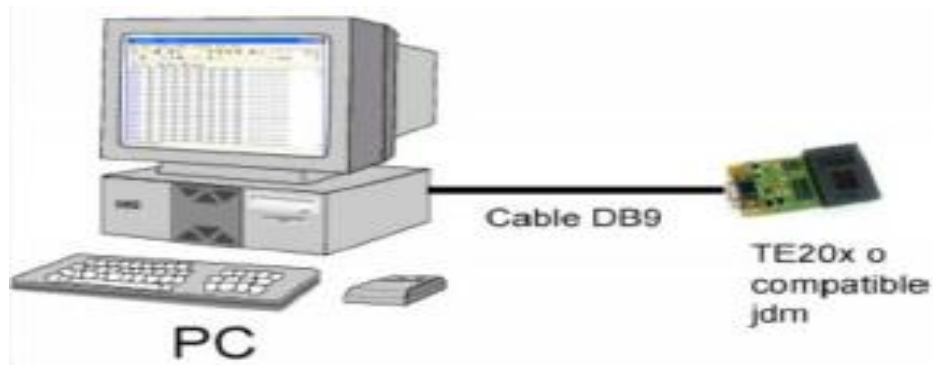


Figura2.25. Comunicación entre el PIC y el PC

Microcode Studio ICD se comunica con el microcontrolador utilizando su hardware USART, WinPic ó IC-prog. Nos debemos remitir a la ficha de datos de los microchips el microcontrolador que se utiliza.

En la actualidad las velocidades soportadas incluyen 4, 8, 10,12,16,20,24, 25,32,33y 40 MHz. Sin embargo es necesario que el código PICBasic tenga el oscilador de ajuste definido. Si el dispositivo de destino se está ejecutando a una velocidad diferente, hay que asegurarse de la adecuada definición que se utiliza con el código PICBasic.

2.4. SOFTWARE PROTEUS PROFESIONAL

2.4.1. Descripción

Proteus es una compilación de programas de diseño y simulación electrónica, que consta de los siguientes programas principales:

- **ARES** e **ISIS**, y los módulos VSMY Electra.

Proteus es un entorno integrado diseñado para la realización completa de proyectos de construcción de equipos electrónicos en todas sus etapas: diseño, simulación depuración y construcción

2.4.2. Aspectos Básicos

Proteus es un programa para simular circuitos electrónicos complejos integrando inclusive desarrollos realizados con microcontroladores de varios tipos, es una herramienta de alto desempeño con unas capacidades graficas impresionantes.

Presenta una filosofía de trabajo semejante al SPICE, arrastrando componentes de una barra e incrustándolos en la aplicación.

2.4.3. Características

- Entorno de diseño grafico de esquemas electrónicos (**ISIS**) fácil de utilizar y dotado de poderosas herramientas para facilitar el trabajo del diseñador.
- Entorno de diseño de placas de circuito impreso (**ARES**) de ultra-altas, prestaciones con bases de datos de 32 bits posicionador automático de elementos y generación automática de pistas con tecnologías de auto corte y regeneración.
- Moderno y atractivo interface de usuario estandarizado a lo largo de todas las herramientas que componen el entorno **PROTEUS**.
- La mayor parte de los módulos que componen **PROTEUS** han sido escritos por el mismo equipo, garantizado al máximo nivel posible, la compatibilidad e inter-operatividad asegurando su sensibilidad al máximo.

- Ejecutable en los diferentes entornos Windows 98, 2000, XP.
- Herramienta de máximas prestaciones, basadas en los más de los 15 años de continuo desarrollo y presencia en el mercado.
- Miles de instalaciones vendida a mas de 35 países a todo lo largo de el mundo

2.4.4. Elementos

Principalmente se compone de cuatro elementos, perfectamente integrados:

- ISIS
- ARES
- VSM
- Electra

2.4.5. ISIS

El programa ISIS (sistema de Enrutado de Esquemas inteligentes), permite diseñar el plano eléctrico del circuito que se desea realizar con componentes muy variados. ISIS se suministra con una extensa librería, compuesta por más de 8.000 dispositivos. Esta librería incluye elementos estándar como, resistencias, condensadores, transistores, diodos, válvulas, TTL, CMOS, ECL, microprocesadores, memorias, PLDs, ICs análogos y amplificadores operacionales, sócalos, conectores de Tierra, positivo y otros componentes con prestaciones diferentes. ISIS es la herramienta ideal para una rápida realización de complejos diseños de esquemas electrónicos destinados tanto a la construcción de equipos electrónicos, como a la

realización de tareas de Simulación y prueba. Además ISIS es una herramienta excepcional para la realización de atractivos esquemas electrónicos destinados a su publicación en libros, manuales documentos técnicos, e inclusive, antes de hacer el plano eléctrico, se pueden elegir rotulados y tamaños para la impresión desde tamaños portátiles (A5, A4, A3), hasta tamaños estilo plotter (grande): A2, A1 e inclusive A0.

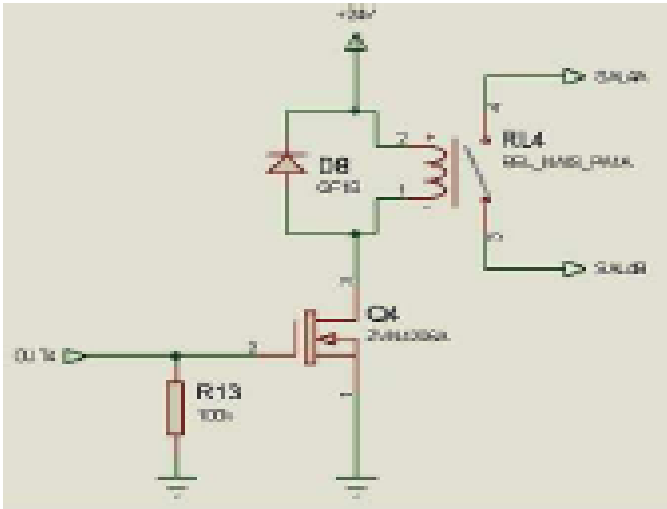


Figura 2.26.Esquema de ISIS

ISIS viene provisto de las habituales herramientas de cortar y copiar y pegar desde el portapapeles de Windows. De esta manera podemos copiar de manera muy sencilla. La totalidad o parte esquema electrónico y pegarla en cualquier otra herramienta que utilice el estándar del portapapeles de Windows.

Además la librería incorpora dispositivos específicos de los fabricantes más importantes como Philips, National semiconductor, Motorola, Teccor, Texas y Zetex.

Un proceso de actualización permanente posibilita que cada cuatro meses la librería se incremente con nuevos productos aparecidos en el mercado. Los

2.4.5.2. Completa Gestión de Diseños Jerárquicos.

En la actualidad los circuitos electrónicos han aumentado su complejidad incluso en los proyectos más pequeños. Para facilitar la gestión de los esquemas complejos, ISIS incorpora la gestión de diseños multihojas, es decir permite repartir un esquema electrónico en varias hojas de papel de forma rápida y cómoda.

Por añadidura la potencia de ISIS para el tratamiento de esquemas complejos se pone manifiesto con su gestión de los diseños jerárquicos. El diseño jerárquico consiste en que un particular componente puede ser definido como un modulo que representa un determinado esquema electrónico y que se comporta de acuerdo con él.

2.4.5.3. Utilización de Buses

Un bus es una serie de conductores en paralelo. Su uso más extendido es en la gestión de las líneas de datos y direcciones en el diseño de microprocesadores. La figura 2.28 muestra la disposición de buses en Isis.

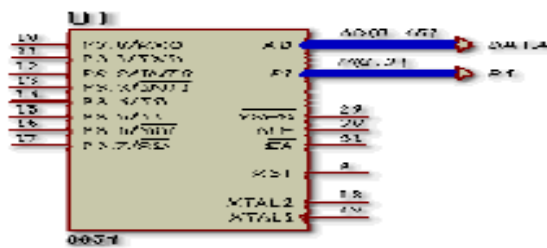


Figura 2.28. Buses de Isis

2.4.5.4. Administrador de las Propiedades de los Componentes

Cada uno de los componentes que se incluyen en un diseño electrónico posee una lista de propiedades. Algunas de ellas se refieren de forma específica a su empaquetado o al comportamiento del modelo utilizado para la simulación de los circuitos electrónicos.

ISIS además le permite añadir las propiedades que usted desee para sus propias necesidades, incluye una completa colección de herramientas para permitir la selección de grupos, de componentes de acuerdo a sus propiedades o la manipulación de las propiedades de un determinado grupo de componentes. La herramienta para la importación de datos de formato ASCII (ADI) que incorpora ISIS posibilita la importación de propiedades desde ficheros ASCII, utilizando reglas muy simples para determinar a qué tipo de componentes se deben asignar.

La figura 2.29 muestra los componentes de uno de los diferentes dispositivos para el diseño de circuitos

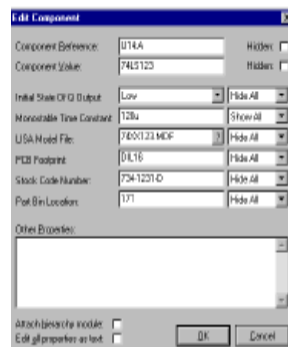


Figura 2.29. Diagrama de componentes de Isis

Cuando se crea un nuevo elemento de librería, se pueden definir las propiedades de ese componente y sus valores por defecto. Además cada propiedad debe ser definida totalmente tanto en lo referente a su comportamiento (visible, editable, etc.).

2.5. PICKIT2 CLONE VERSIÓN 2.61 PARA PUERTO USB

Es un quemador de PIC USB, siendo éste una versión simplificada del original PICKit2 de Microchip. Trabaja con el programa de aplicación PICKit2 v2.61 de Microchip, por lo tanto su buen funcionamiento está garantizado y asegurado.

Microchip en su página oficial informa que PICKit2 trabaja correctamente en Windows XP y Windows Vista y probado exitosamente en Windows 7 (Home Premium, Ultimate y Professional).

La lista de dispositivos soportados por este grabador de PIC es muy extensa, y se puede ver ingresando a *Help* → *ReadMe* en el programa de aplicación PICKit2 v2.61.

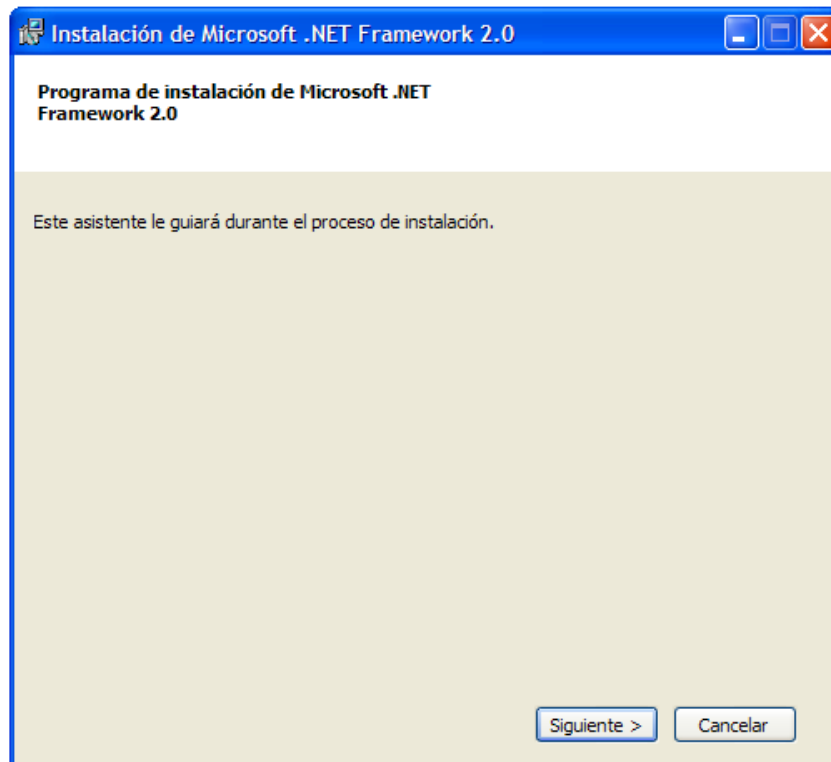
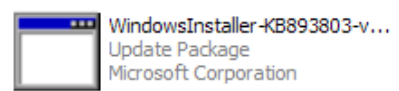
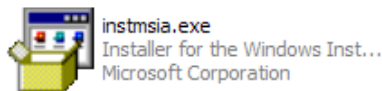
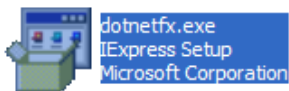
La conexión se realiza a través de los pines VPP (voltaje de programación), VDD (voltaje de alimentación), VSS (referencia), PGD (datos) y PGC (reloj) del programador y los pines correspondientes del microcontrolador.

2.5.1. Pasos para la descarga e instalación de PICKit2 v2.61

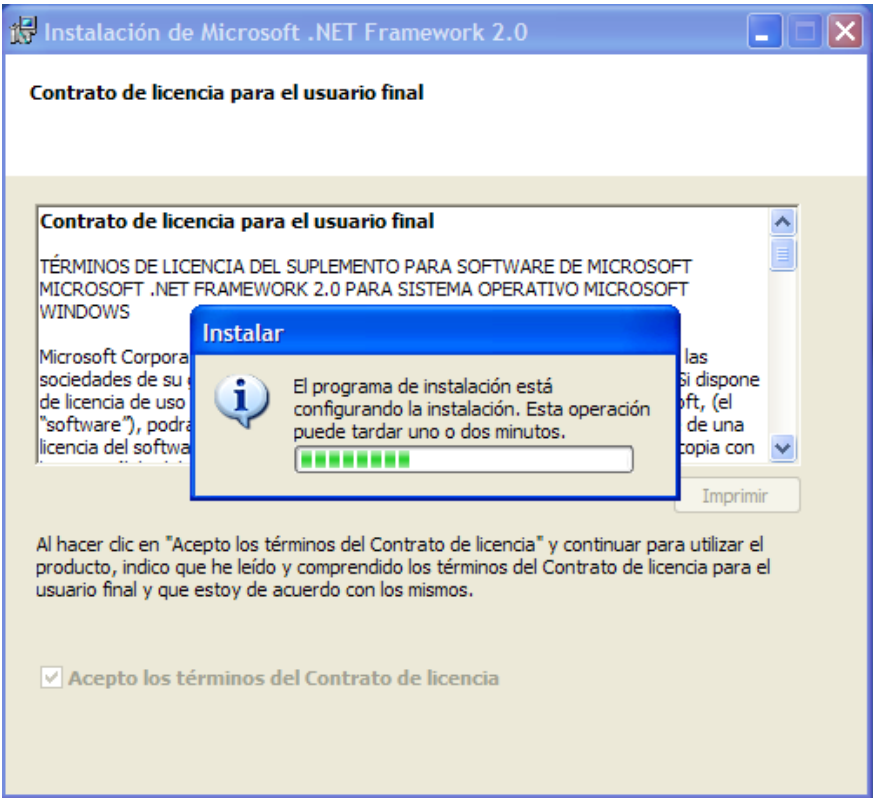
- Para obtener la versión más actual, puede entrar a la web de Microchip, www.microchip.com buscar PICKIT2, y descargar la versión de software y firmware más reciente. El hardware diseñado del PICKIT2 por

APM MICRO, le permite que el equipo se pueda actualizar directamente usando la interfaz del Software, opción: **Tools: “Download Pickit2 OperationSystem”**.

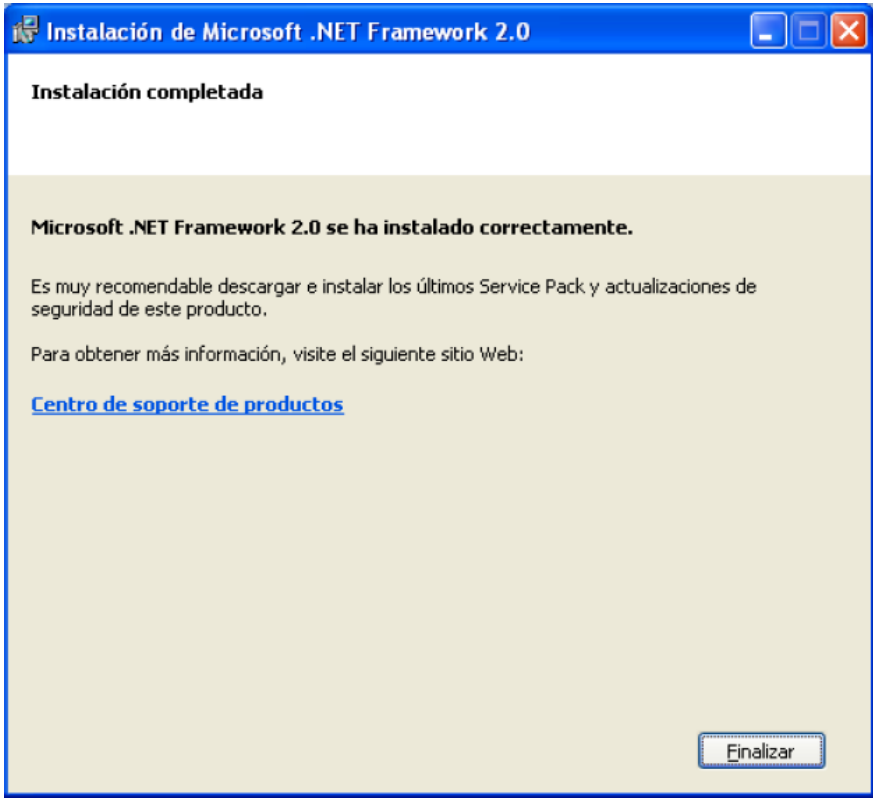
- El software PicKit2, requiere para su correcta operación tener instalada una versión de NetFrameWork 2.0 o superior. Instalador que se encuentra dentro de la carpeta **VPICKIT2_v2.61WETFRAMEWORK2.0**, del instalador de PicKit2 que se descargo desde la página de MICROCHIP.
- Para instalar Netframework 2.0, ingresamos a la carpeta correspondiente. Damos doble click sobre el archivo **dotnetfx**, y seguimos todos los pasos de instalación de netframework 2.0



- Aceptar Términos de contrato, dar click en **Siguiete >**.

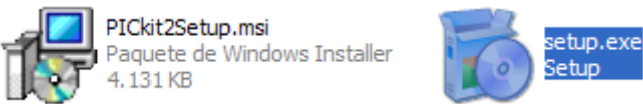


➤ Se instalará los componentes de **Netframework**, hasta terminar.

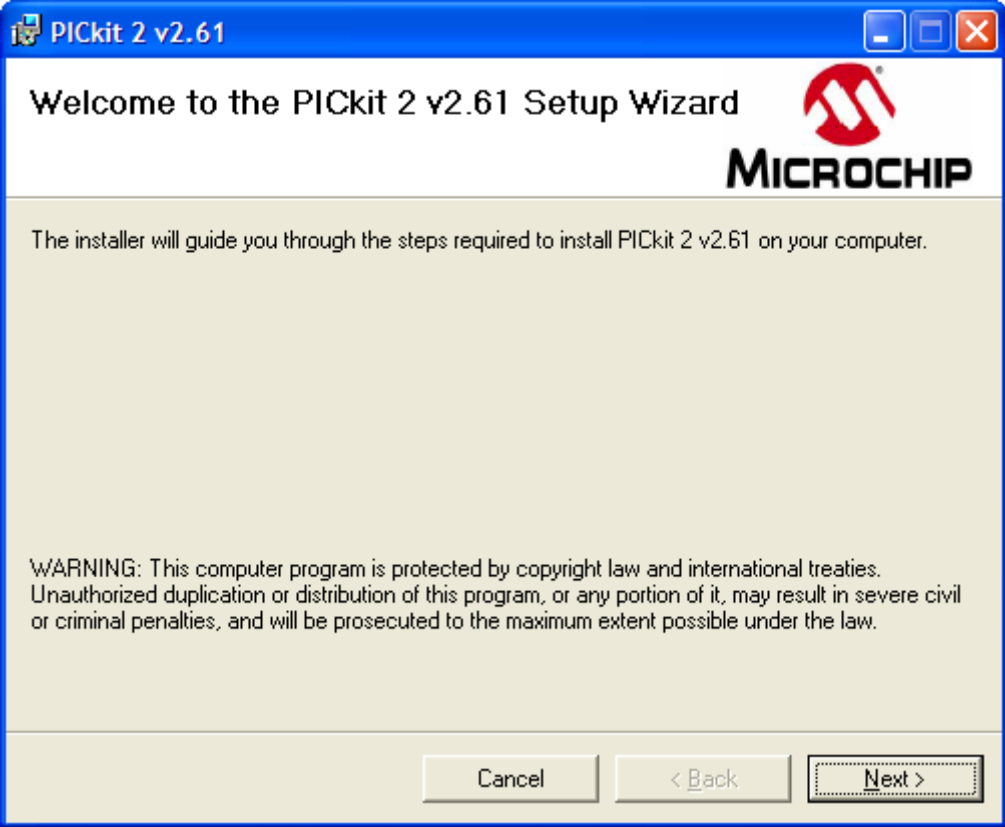


➤ Damos click en **Finalizar**.

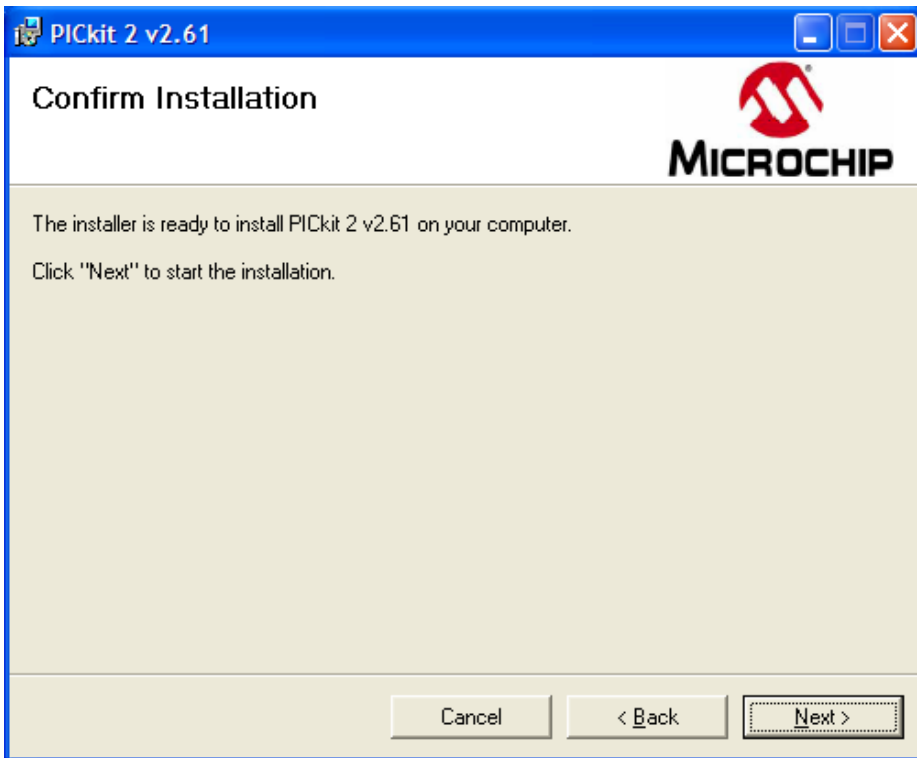
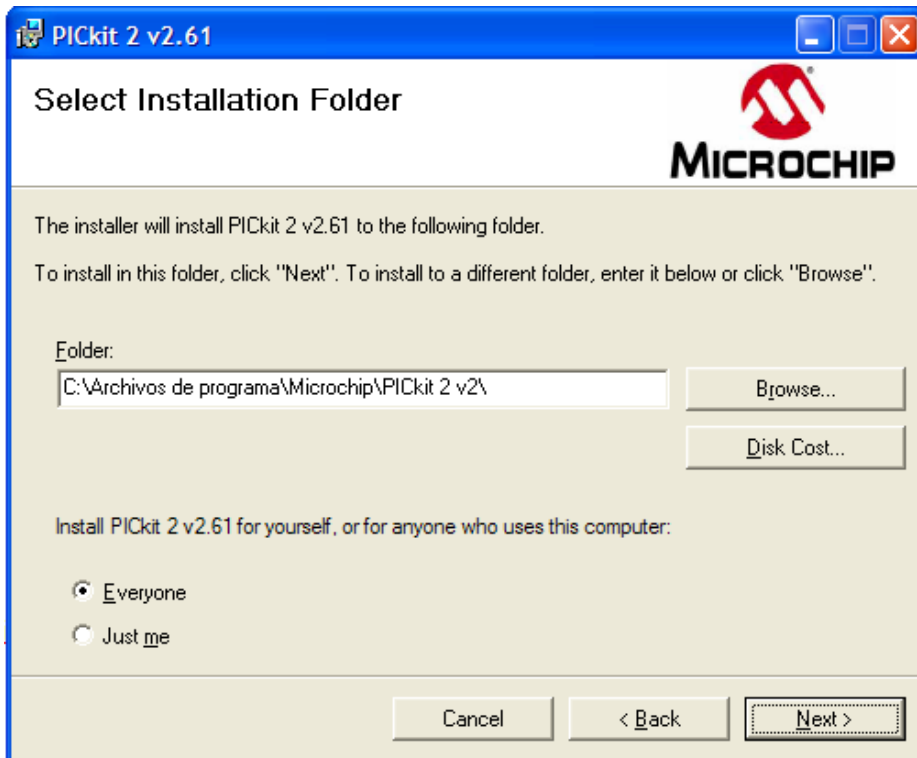
- Una vez terminada la instalación de NetFrameWork 2.0, continuamos instalando el software PICKIT2. Para instalar la aplicación PICKIT2_v2.61, abrimos la carpeta correspondiente y encontraremos los archivos.



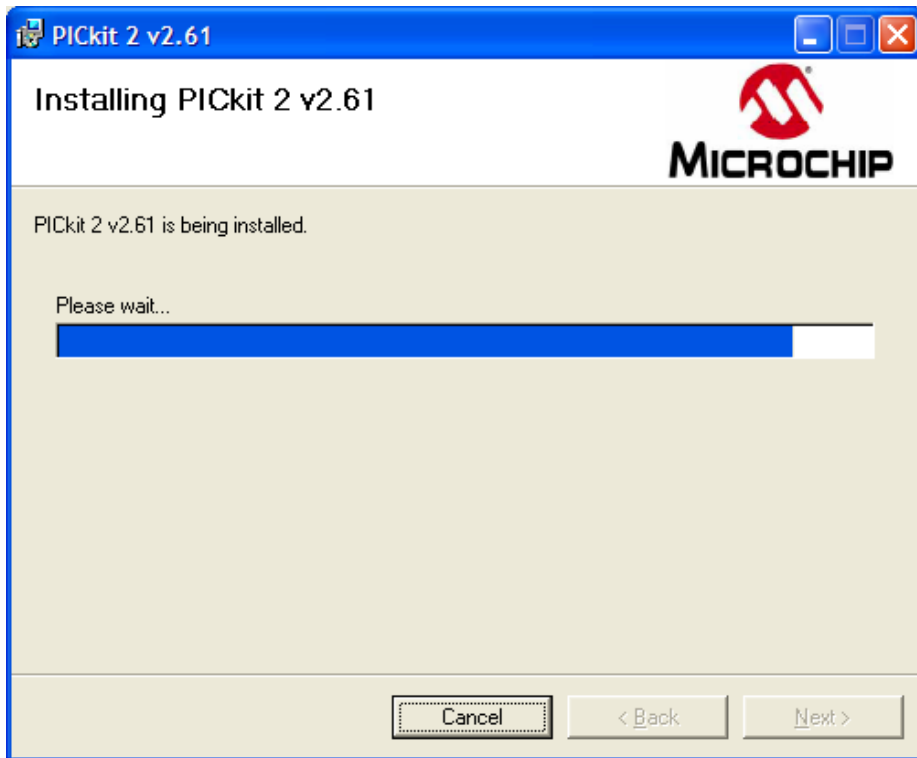
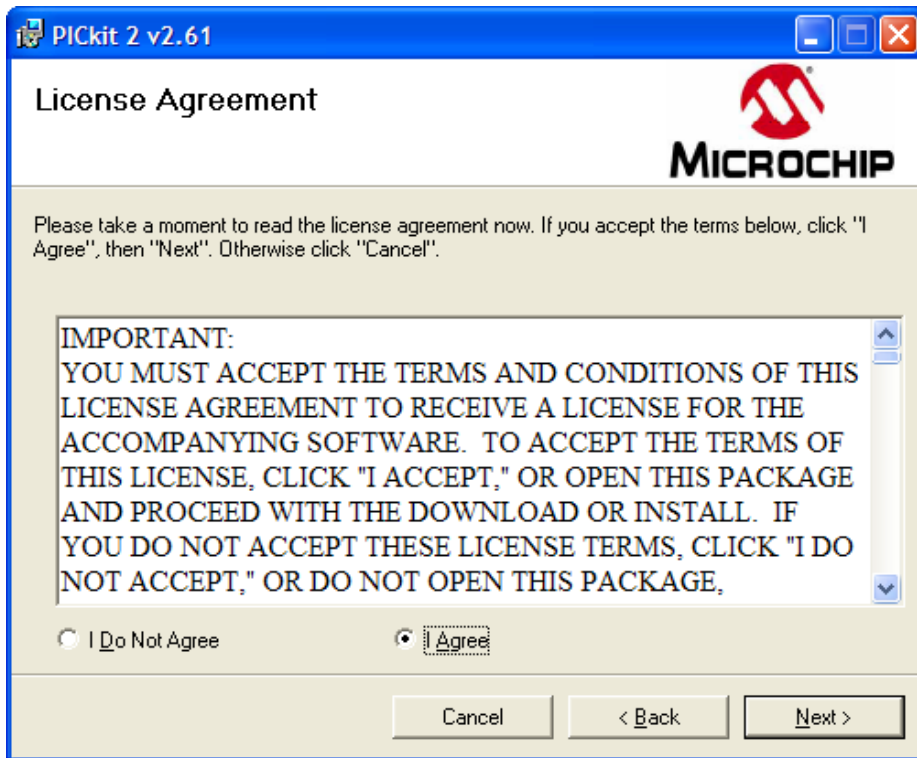
- Damos doble click sobre el icono **SETUP**, e instalamos la aplicación



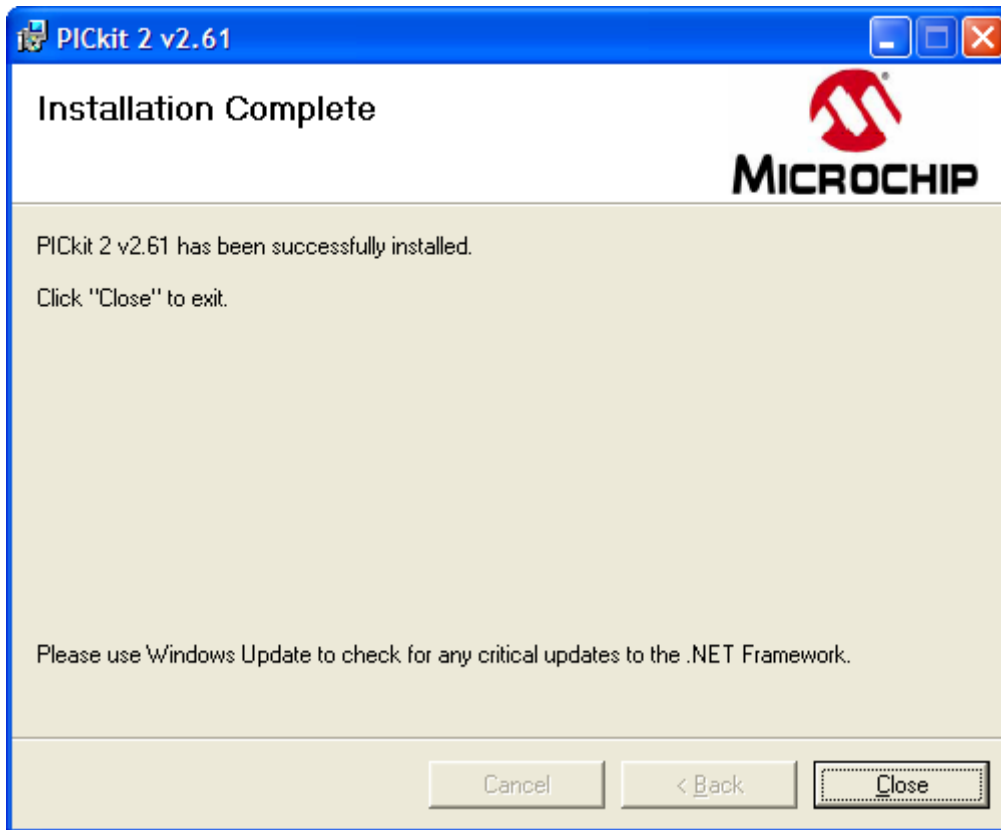
- Dar click en **Next>**. Se recomienda dejar la carpeta de instalación por Default.



- Damos click en **Next>**. Y luego aceptamos la licencia *I agree*, y damos click en **Next>**.



- Una vez terminado tenemos nuestro software listo para trabajar, dar click en **Close**.



En este punto procedemos a conectar el programador PICKIT2 a nuestra computadora, el sistema detectará un ***Nuevo hardware encontrado***, pero no nos pedirá ningún driver ya que el grabador tiene una interfaz HID, por lo que Windows lo instalará automáticamente.

2.6. FUENTE DE ALIMENTACIÓN DEL MÓDULO DE PRÁCTICAS

Cualquier circuito electrónico necesita energía para funcionar, esta energía la podemos obtener desde una pila o batería o a través de la red eléctrica. La tensión que nos suministra la red eléctrica es alterna (AC) y habitualmente excede en mucho el voltaje que necesitamos, por lo que tenemos que insertar un circuito electrónico que nos transforme el voltaje y tipo de corriente de la red (110V AC) al voltaje y tipo de corriente (AC o DC) que

necesitamos en nuestro circuito. Este circuito se denomina fuente de alimentación.

Básicamente existen dos tipos de fuentes de alimentación, las lineales, que utilizan un transformador para disminuir el nivel de tensión en la red eléctrica al nivel necesario en nuestro circuito y las fuentes conmutadas que utilizan circuitos basados en transistores y bobinas trabajando en conmutación para reducir la tensión. Las ventajas de la fuente de alimentación lineal es su sencillez y que generan menos ruido electromagnético, las desventajas son su mayor tamaño y su menor eficiencia (disipan más energía en forma de calor que las fuentes conmutadas).

2.6.1. Estructura Básica

En la siguiente figura podemos ver la estructura básica de una fuente de alimentación lineal:

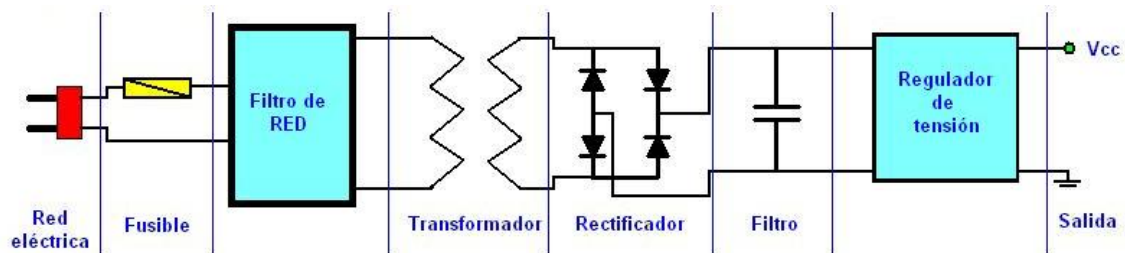


Figura 2.30. Diagrama de Bloques de una fuente de alimentación lineal

Podemos apreciar en el diagrama que una fuente de alimentación esta compuesta por distintos módulos que poseen una función concreta.

2.6.2. Conexión Red Eléctrica

Esta formada por el enchufe, bornes o cualquier dispositivo físico, que nos permite conectar nuestra fuente de alimentación a la red eléctrica. Los

parámetros que hay que tener en cuenta a la hora de elegir el enchufe es que soporte la tensión de la red (110V a60Hz) y la corriente que va a consumir el circuito. Parámetros que serán también de gran importancia al momento de elegir el cable de alimentación.

2.6.3. Fusible

Si nuestra fuente de alimentación tuviera un fallo y se cortocircuitara, producirá una subida muy fuerte en el consumo de corriente, las consecuencias de esta subida son impredecibles, ya que si esta fuera muy elevada podríamos hacer saltar el automático de nuestra vivienda e incluso del edificio y si fuera relativamente pequeña podría subir la temperatura de nuestro circuito hasta el punto de producir un incendio. El fusible es un dispositivo que cuando la corriente que circula por él es superior a su corriente nominal se funde interrumpiendo el suministro de corriente. El parámetro básico que necesitamos calcular para seleccionar nuestro fusible es la corriente nominal.

2.6.4. Filtro de Red

Este dispositivo no es estrictamente necesario ya que su función es la de eliminar las posibles perturbaciones electromagnéticas que puedan llegar a nuestra fuente de alimentación desde la red eléctrica, pero su uso es imprescindible si queremos hacer a nuestro equipo inmune a dichas interferencias. Aunque el filtro de red lo podemos realizar nosotros, lo mejor es adquirir un filtro comercial, ya que estos han sido testados para cumplir con las normas sobre EMIS (Egton Medical Information Systems).

2.6.5. Transformador

El transformador es un dispositivo electrónico que nos permite transformar una tensión alterna de entrada en una tensión alterna de salida de distinto valor. La principal ventaja que tienen los transformadores es su alto rendimiento.

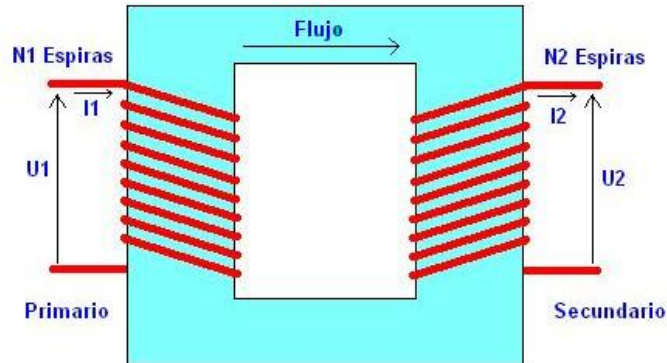


Figura 2.31. Esquema de un transformador

La corriente alterna que circula por el devanado del primario induce un flujo magnético que circula por el núcleo induciendo en el secundario una tensión alterna. El flujo magnético en el devanado 1 y 2, suponiendo que no hay **pérdidas**, lo podemos expresar según las ecuaciones:

$$U1 = -N1 \frac{\partial \phi}{\partial t}; \quad \text{Ecuación (1)}$$

$$U2 = -N2 \frac{\partial \phi}{\partial t}; \quad \text{Ecuación (2)}$$

Como el flujo es igual en los dos devanados, si dividimos la primera ecuación por la segunda tenemos:

$$\frac{U1}{U2} = \frac{N1}{N2} = r; \quad \text{Ecuación (3)}$$

Esta ecuación nos dice que la relación entre la tensión de entrada y de salida viene dada por la relación que existe entre el número de espiras que tengan los devanados. A esta relación r se le denomina relación de transformación en vacío.

Como hemos dicho anteriormente el transformador es un dispositivo con muy pocas pérdidas por lo que podemos decir que la potencia en el primario será igual a la potencia en el secundario (si hubierapérdidas la potencia del primario sería igual a la potencia del secundario más la potencia de las perdidas). Esto nos permite igual las potencial del primario y del secundario según la siguiente ecuación:

$$P_1 = P_2 \rightarrow U_1 \cdot I_1 = U_2 \cdot I_2 \rightarrow \frac{U_1}{U_2} = \frac{I_2}{I_1}; \quad \text{Ecuación (4)}$$

Esta ecuación es muy útil para calcular la intensidad nominal del fusible de alimentación, ya que si tenemos, por ejemplo, un transformador con 110V en el primario y 9V en el secundario y estamos consumiendo 1A en el secundario, podemos calcular la intensidad en el primario de la siguiente manera:

$$\frac{U_1}{U_2} = \frac{I_2}{I_1} \rightarrow I_1 = I_2 \frac{U_2}{U_1} = (1A) \frac{9v}{110v} = 0,082 A$$

Esto significa que en el primario tendríamos que poner un fusible mayor de 82mA para poder soportar esta intensidad en el secundario. En el mercado no hay una variedad infinita de fusibles por lo que habrá que buscar el valor estándar que más se aproxime al valor calculado.

Aunque hemos dicho que el rendimiento del transformador es muy alto, este valor no es 100% y por tanto siempre hay **pérdidas** que aumentan según vamos aumentando la intensidad consumida en el secundario, esto se traduce en una bajada de la tensión en el secundario y un desfase entre la señal de entrada y la de salida. De todas maneras si no sobrepasamos la corriente del transformador estas ecuaciones son perfectamente validas.

Los transformadores suelen llevar dos **bornes** de entrada para el devanado del primario en el que conectaremos los 110V, en el secundario sin embargo podemos encontrar 3 configuraciones básicas:

- **Un devanado secundario:** En este caso solo hay dos **bornes** para el secundario por las que obtenemos la tensión de salida. Un ejemplo sería un transformador reductor de 110V-AC a 12V-AC y de 1A.
- **Un devanado con toma intermedia:** El secundario dispone de 3 **bornes**, en el que la tercera toma esta conectada en medio de la bobina del secundario. Un ejemplo sería un transformador de 110V-AC a 12V medidos desde la toma central, 24V entre los extremos y de 1A.
- **Dos devanados independientes:** El secundario esta dividida en dos bobinas independientes para poder conectarlas de la forma que nosotros queramos, de tal forma que podríamos obtener dos tensiones diferentes, una tensión que será la suma de los dos devanados o un solo devanado con toma intermedia. Si nos basamos en el diagrama de conexiones del transformador encapsulado de la figura 3 y usando como ejemplo un transformador de 110V/12V+12V y

1A, podríamos obtener 12V de cada una de las bobinas si las utilizamos independientemente o en el caso de unir las tomas O y V', podríamos obtener 24V de las tomas V y O'. También podemos utilizar OV' como toma intermedia de un rectificar de media onda.

En el dibujo inferior podemos ver dos fotos de transformadores reales, uno encapsulado y pensado para ser soldado directamente en la placa de CI y otro con terminales soldables preparado para poner en panel.

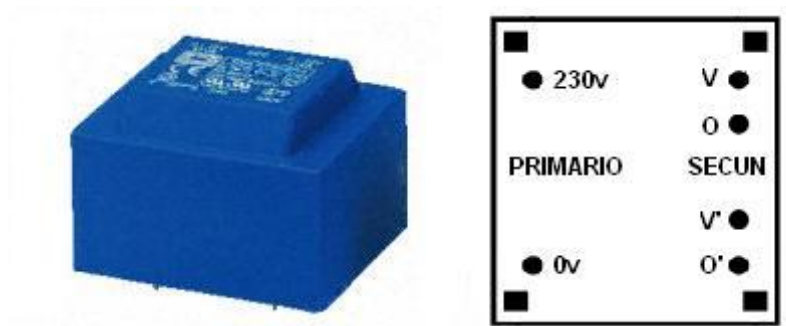


Figura 2.32. Transformador encapsulado

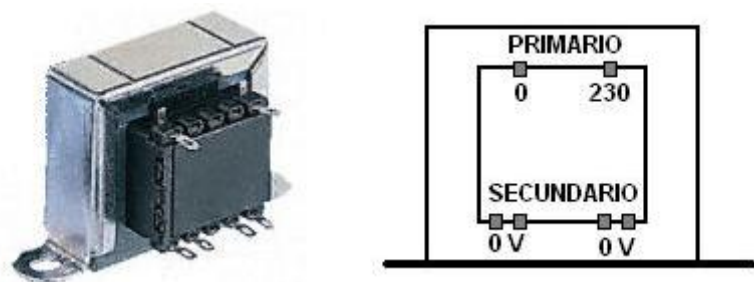


Figura 2.33. Transformador con brida

Un detalle a tener en cuenta es la diferencia que existe entre tensión eficaz y tensión de pico. Cuando utilizamos corriente alterna las tensiones se dan en su valor eficaz, es decir, el valor que la tensión tendría si fuera continua,

pero como esta no lo es, aparece otro parámetro que es la tensión de pico Vpk que podemos ver gráficamente en la *figura 2.34* y que esta relaciona con la tensión eficaz mediante la siguiente ecuación:

$$V_{ef} = \frac{V_{pk}}{\sqrt{2}}; \text{ Ecuación (5)}$$

2.6.6. Rectificador

La mayoría de los circuitos electrónicos utilizan para funcionar corriente continua (DC), mientras que, como hemos comentado anteriormente, la tensión que llega y sale del transformador es alterna (AC). Para poder transformar esta corriente alterna en continua utilizamos un circuito basado en diodos semiconductores al que denominamos *rectificador*. En la *figura 2.34* vemos la forma de la tensión alterna como sale del transformador y como queda después de rectificarla:

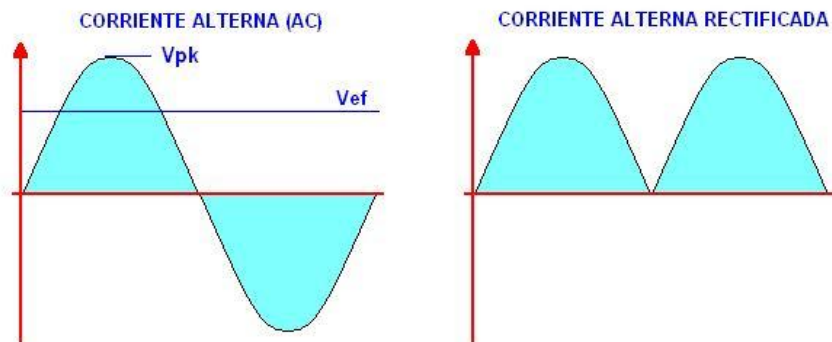


Figura 2.34. Forma de onda rectificada.

Existen diversas configuraciones para realizar esta función, aquí nos vamos a centrar en la rectificación monofásica de onda completa. En la

figura2.35 se puede ver dos configuraciones para un rectificador de onda completa con transformador con y sin toma intermedia.

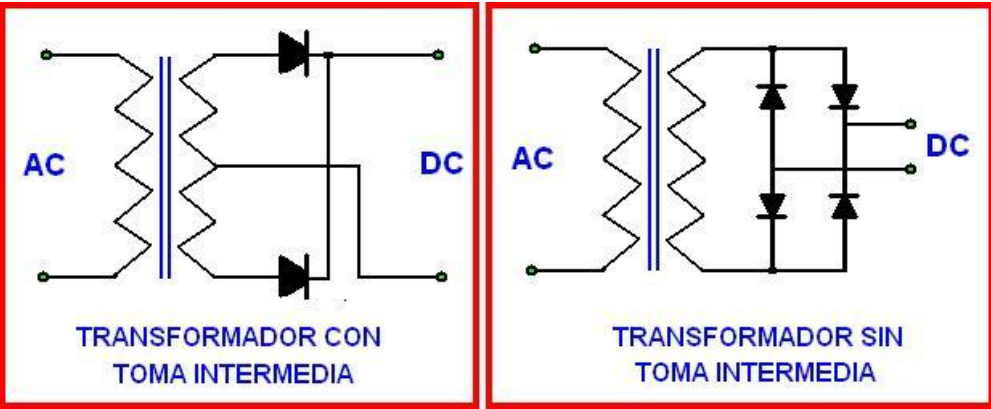


Figura2.35. Rectificación de onda completa

La configuración de los cuatro diodos se denomina puente rectificador y en el mercado existen puentes rectificadores que integran en un mismo encapsulado los 4 diodos. En la figura 2.36 vemos algunos de componentes reales.

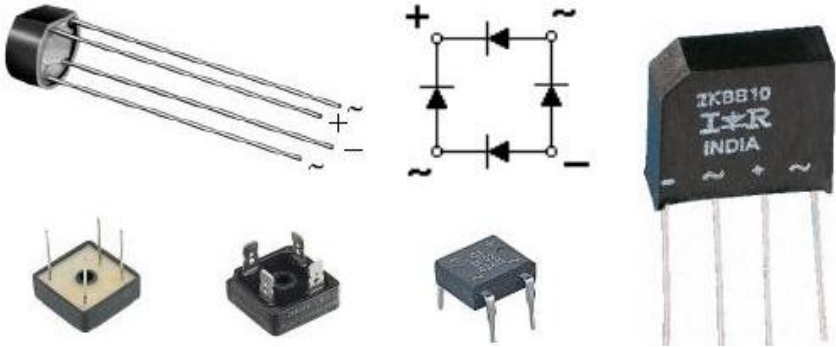


Figura 2.36. Encapsulados de puentes rectificadores

Normalmente estos componentes tienen impresos el nombre de las patillas siendo + y - las salidas en continua y ~ las entradas de alterna. Para

seleccionar el puente rectificador (o diodos) que necesitamos, necesitamos determinar la tensión y la corriente máxima de trabajo, que han de ser suficientes para nuestro circuito. Por ejemplo, si queremos construir una fuente de alimentación de 12V y 1A en el secundario, necesitaremos un puente rectificador (o 4 diodos) que soporten al menos 1 amperio y 12V, siempre intentando dejar un margen de al menos un 30%, lo que quiere decir que necesitaríamos uno de 1,3A y 15,6V (este valor de corriente posiblemente no lo encontremos en el mercado y tendremos que ir a uno de 1,5A, en cuanto a la tensión normalmente utilizaremos de 110V por lo que no habrá problemas).

2.6.7. Filtro

Una vez la señal esta rectificada, obtenemos una forma de onda que no es precisamente continua (ver figura 2.37). Para poner eliminar la ondulación, y dejar la tensión lo más continua posible, filtraremos la señal utilizando uno o más condensadores en paralelo. En la *figura 2.37* se puede apreciar como queda esta señal una vez filtrada.

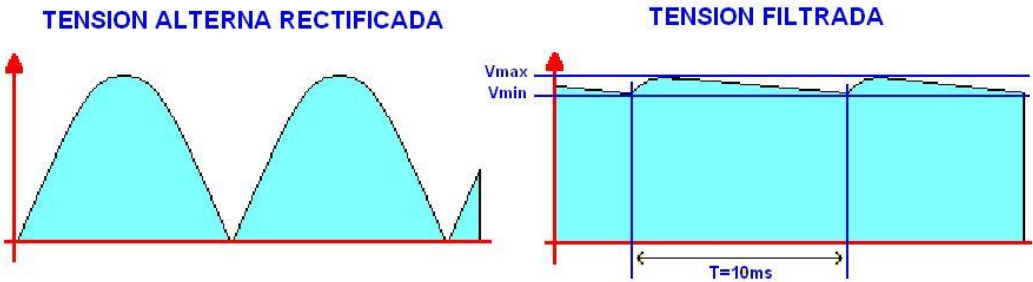


Figura 2.37. Señal filtrada después de la rectificación.

Para calcular el valor del condensador, podemos utilizar una aproximación bastante buena con la siguiente ecuación:

$$C = \frac{Q}{V_{max} - V_{min}} = \frac{I_{max}T}{V_{max} - V_{min}}; \quad \text{Ecuación (6)}$$

En donde:

- Vmax: Es el valor máximo de la tensión de entrada que equivale al valor de pico del secundario del transformador (Vpk).
- Vmin: Tensión mínima que queremos que tenga la tensión de entrada y que determina el rizado de la fuente.
- Imax: Intensidad máxima en el secundario.
- T: Periodo de la señal de la red, para 60Hz y rectificador de onda completa son 8,33 ms. En media onda sería 16,66 ms.
- C: Capacidad del capacitor del filtro en faradios.

2.6.8. Regulador de Tensión

Como vemos en la figura 2.37, la salida filtrada presenta una pequeña ondulación, para eliminar esta ondulación y controlar la tensión para que esta no cambie ante variaciones de corriente en la carga, utilizamos un regulador de tensión. Lo mejor es utilizar un circuito integrado comercial como es el caso de la serie 78XX. Vamos a realizar un caso práctico de una fuente de alimentación con salida 5V a 0,5A utilizando el 78LS05.

El esquema de la fuente de alimentación es el siguiente:

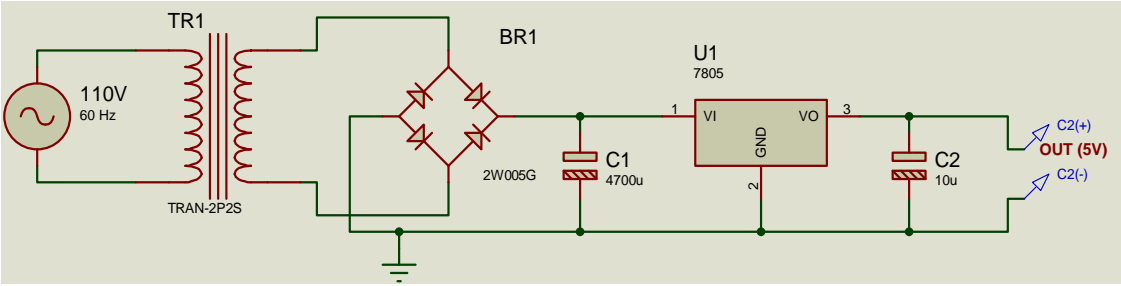


Figura 2.38. Fuente de alimentación de 5V a 1ª

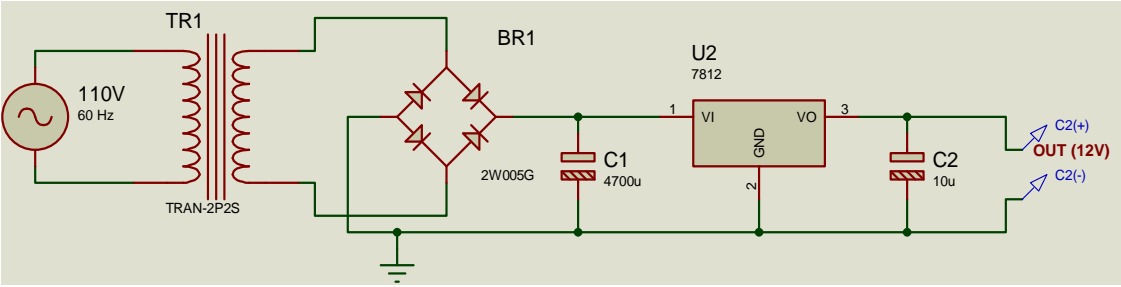


Figura 2.39. Fuente de alimentación de 12V a 1ª

Calculemos cada uno de los componentes:

- Transformador: El CI 78LS05 necesita una tensión mínima de trabajo, es decir, si tenemos un CI 78LS05 cuya tensión de salida son 5V, no podemos hacer funcionar el circuito con una tensión de entrada al regulador de 5V, ya que no habría tensión para activar el circuito, en el Datasheet del CI 78LS05 encontramos un parámetro llamado “dropout voltaje” que nos dice la tensión mínima que ha de caer en el integrado para que este funcione, como este valor es 2,5V, determinaremos que para que el circuito funcione correctamente necesitaremos una tensión en la entrada de al menos $5V+2,5V=7,5V$. Además en el diodo caen 0,7V cuando este esta conduciendo por los que la tensión mínima de entrada ha de ser de al menos 8,2V. Según lo visto vamos a utilizar un transformador de 110V en primario y 9V+9V y 0.5A en secundario.

- Fusible: Utilizando la ecuación (4), calculamos que el fusible ha de ser de 19 mA, como este valor no lo vamos a encontrar utilizaremos uno de 0,1A.
- Condensador: Para calcular el condensador utilizamos la ecuación (6). El valor de los parámetros de la ecuación son:
 - T : Para un rectificador de onda completa vale 16,66 ms.
 - I_{max} : Hemos determinado que la intensidad máxima que va a suministrar la fuente son 0,5A.
 - V_{max} : Mediante la ecuación 2 y sabiendo que la tensión eficaz del secundario es 9V AC, el valor de $V_{max} = 9 \cdot \sqrt{2} = 12,72V$
 - V_{min} : En los cálculos de transformador dijimos que la tensión mínima que necesitamos para que la fuente funcione es de 8,2V, utilizaremos el valor de 9V para dejar un margen de seguridad.

Con todos estos parámetros y aplicando la ecuación 3 calculamos que $C = 0,001344 F = 1344 \mu F$. Utilizaremos el valor comercial común más cercano por arriba que es de 2200 μF .

- Regulador: Este va a ser el CI 78LS05, que da perfectamente los valores pedidos de 5V y 0,5A.
- Otros componentes: El CI 78LS05 necesita un condensador de valor $4700\mu F$ en la entrada y en la salida, añadimos un condensador más (10 μF) para estabilizar la tensión en la salida del circuito. El diodo D3 se utiliza para evitar que el CI 78LS05 se polarice en inversa y pueda sufrir algún daño.

Utilizando el mismo esquema y sustituyendo el CI 78LS05 por otros de la misma serie como puede ser el 78LS12, 78LS15, 78LS24 y recalculando el valor de todos los componentes podremos obtener fuentes de alimentación de distintas tensiones de salida.

Otro elemento importante del diseño es el radiador de calor que puede, o no, ser necesario en nuestro circuito. Para calcular si necesitamos radiador primero necesitamos saber la potencia que va a disipar el regulador de tensión en el peor de los casos (cuando la tensión de entrada es máxima), para ello utilizaremos la fórmula $P=VI$, en donde V será la tensión que cae en nuestro regulador, cuyo valor vendrá determinado por la tensión máxima de entrada V_{max} menos las caídas de tensión que se producen fuera del regulador, es decir, la caída de tensión en el diodo (0,7V) y la caída de tensión en la carga (los 5v de salida del regulador).

$$\text{Así tenemos: } V_{reg} = V_{max} - V_{diodo} - V_{salida}, \quad \text{Ecuación (7)}$$

Y la multiplicaremos por la intensidad máxima, quedando al final la potencia como:

$$P_{max} = V_{reg} * I_{max}, \quad \text{Ecuación (8)}$$

$$P_{max} = (V_{max} - V_{diodo} - V_{salida}) * I_{max} = (7.02) * (0.5) = 3,51W.$$

Si vamos al Datasheet del L7805 y vemos la resistencia térmica del encapsulado TO-220 entre unión y ambiente ($R_{thj-amb}$), esta tiene un valor de 50 °C/W, esto quiere decir que por cada vatio que disipemos en el encapsulado, la temperatura de este subirá 50°C. Con una simple multiplicación calculamos

que si la potencia disipada por el CI 78LS05 es de 3,51W, la temperatura que alcanzara será de:

$$T_{\text{disipada}} = 3,51\text{W} \cdot 50^\circ\text{C/W} = 175,5^\circ\text{C}$$

Que es una temperatura que el integrado no puede soportar (la temperatura máxima de la unión es de 150°C), por lo que tendremos que poner un radiador. Vamos a calcularlo.

Para calcular un circuito térmico, utilizamos la ley de ohm igual que si fuera un circuito eléctrico, pero sustituyendo (ver ecuación 9) la intensidad por la potencia, la tensión por temperatura y la resistencia por resistencia térmica.

$$V = IR \rightarrow T = PR_{th}, \quad \text{Ecuación (9)}$$

El circuito térmico del 78LS05 sería:

CALCULO RADIADOR

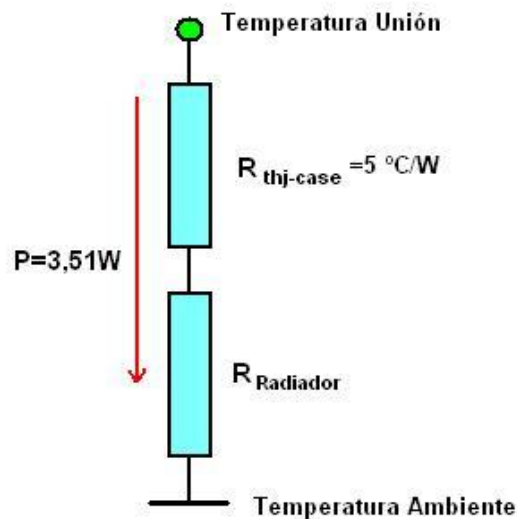


Figura 2.40. Cálculo del Radiador

Si siguiendo el símil entre circuito eléctrico y circuito térmico explicado antes tenemos que la ecuación de este circuito sería:

$$T_{unión} = PR_{thj-case} + PR_{Radiador} + T_{Ambiente}, \quad \text{Ecuación (10)}$$

Veamos el significado y valor de cada uno de los términos:

- $T_{Unión}$: Es la temperatura máxima que queremos que tenga la unión del integrado. En el datasheet del 78LS05 pone que la temperatura máxima de la unión son 150°C, de todas formas utilizaremos una temperatura máxima de 90°C para no tener problemas de calentamiento en otros circuito cercanos y para garantizar una larga vida al integrado.
- $T_{Ambiente}$: La temperatura media que se utiliza para este valor son 25°C, pero esto no es del todo cierto, ya que no es lo mismo la temperatura media en el Polo Norte que en el desierto del Sahara. Se suele utilizar el valor de 35°.
- $R_{thj-case}$: Esta es la resistencia térmica entre la unión y el encapsulado. En el datasheet del 78LS05, podemos ver que para el encapsulado TO-220 vale 5°C/W.
- P : El la potencia que hemos calculado antes y vale 3,51W.
- $R_{Radiador}$: Es el valor a calcular.

Si despejamos $R_{Radiador}$ de la ecuación anterior obtenemos la siguiente ecuación:

$$R_{Radiador} = \frac{T_{Unión} - T_{Ambiente}}{P} - R_{thj-case}, \quad \text{Ecuación (11)}$$

Si sustituimos los valores en esta ecuación obtenemos un valor para R_{Radiador} de $10,6 \text{ }^\circ\text{C/W}$, por lo que buscaremos un radiador con una resistencia térmica de ese valor o inferior.

III. MATERIALES

3.1. MATERIALES DEL MÓDULO

	Cantidad	Elemento
FUENTE DE PODER	1	Transformador con derivación: 5V y 12V (2A)
	2	Capacitores 4700uF
	2	Capacitores 10uF
	4	Diodos 1n4007
	1	Regulador 78LS05
	1	Regulador 78LS12
	1	Placa de baquela virgen 10x18
	2	Fundas de percloruro férrico
MODULO CENTRAL	25	Resistencias 220Ω
	20	Resistencias 4,7kΩ
	1	Resistencia 10Ω
	1	Potenciómetro 10kΩ
	6	Transistores 2n3906
	12	Transistores 2n3904
	3	Leds de alto brillo Rojos
	3	Leds de alto brillo Amarillo
	3	Leds de alto brillo Verde
	1	74LS47
	6	Displays 7 segmentos
	1	LCD
	1	Teclado 4x4
	6	Diodos 1n4007
	6	Relés 12V
	1	Zócalo 16 Pines
	12	Borneras de 2 Pines
	1	Regleta macho 40 Pines
	3	Regletas hembra 40 Pines
	2	Regletas de protoboard

	1	Placa de baquela virgen 21x30
	1	Matriz de leds de 5x7
	2	Fundas de percloruro férrico
COMPUTADOR	1	Mother Intel dp 67de sock 1155 ddr3-1333 Soport 13/15/17 (bulk).
	1	Memoria RAM 4GbAdata 1333 MHZ/ddr3
	1	Disco Duro 1Tb Samsung/Hitachi 7200 rpm
	1	Tarjeta Video 1GbZogisPciExpGforce Ddr3
	1	Tarjeta de Red TrhndnetPci 10/100/1000 GhipRealtek/Teg-Pcitr
	1	Monitor 19 HP Compaq LCD W185q
	1	Super Case Power a TX 6246
	1	Procesador Intel Core 15-2500 a 3.30 GHZ MB Sock 1155
	1	Unidad de DVD WriterSamsumg SH-222AB 22x Sata
	1	Mouse Genius KB06XE Negro USB

IV. PROCESO METODOLÓGICO EMPLEADO

4.1. PROCESO METODOLÓGICO

Este tipo de trabajo está orientado a permitir el contacto y conocimiento del medio organizacional público o privado y de la comunidad. Con el objeto de estudio con el estudiante universitario; mismo que, contribuye al mejoramiento de las condiciones del Laboratorio de Electrónica.

La Práctica en Tecnología aplicada en la presente investigación; comprende el análisis, diseño e implementación de innovación y/o transferencia tecnológica de un módulo de prácticas para microcontroladores y microprocesadores, realizado durante un período de tiempo fijo, que es aplicado únicamente como requisito para la obtención del título de grado en el programa de formación técnica profesional.

4.2. MICROCODE STUDIO

4.2.1. Configuración de Microcode Studio (IDE).

Se debe configurar el editor de texto IDE, para tener el mejor rendimiento posible. Para ello debemos agrupar dentro de la carpeta **C:\mecanique** los dos programas, el **pbp 2.47** y el **IC-prog 106A**, con la finalidad de que la primera vez que ejecute el programa Microcode, pueda encontrar inmediatamente su compilador **pbp 2.47** y su programador **IC-prog 106A**.

Al ejecutar por primera vez **C:\mecanique\MCS\CodeStudio.exe**, el programa buscará el compilador disponible, en este caso el pbp247, y aparecerá una pantalla similar al siguiente gráfico:

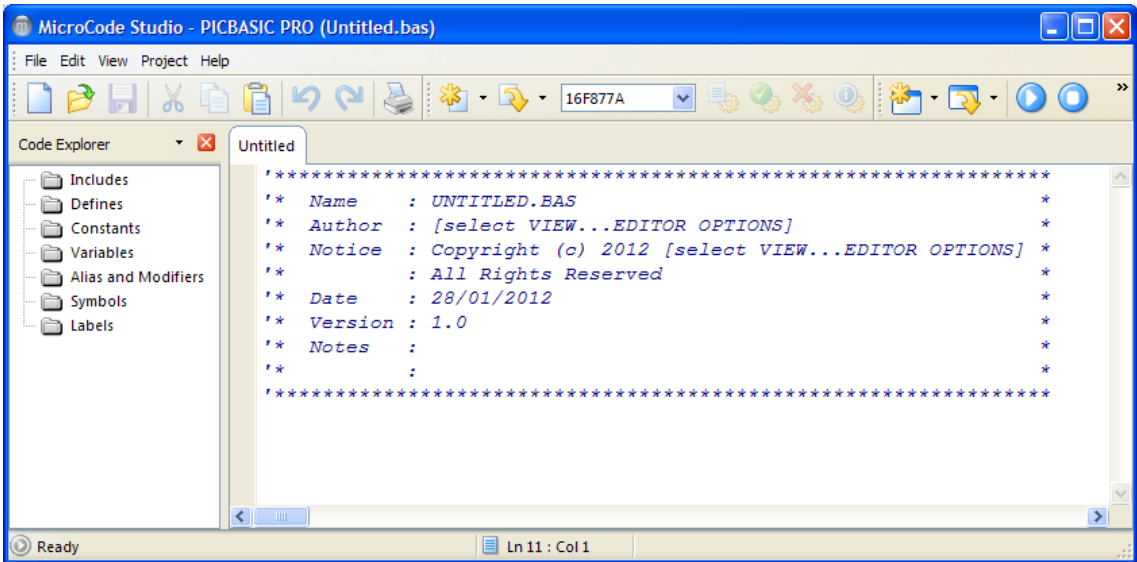


Figura 4.1.Pantalla principal de Microcode Studio

Si en esta pantalla buscamos **View** hacemos clic en **Editor Options**, y posteriormente marcamos **show line numbers in leftgutter**, podremos observar el número de la línea que está programando, resultándonos de gran utilidad al momento de encontrar errores.

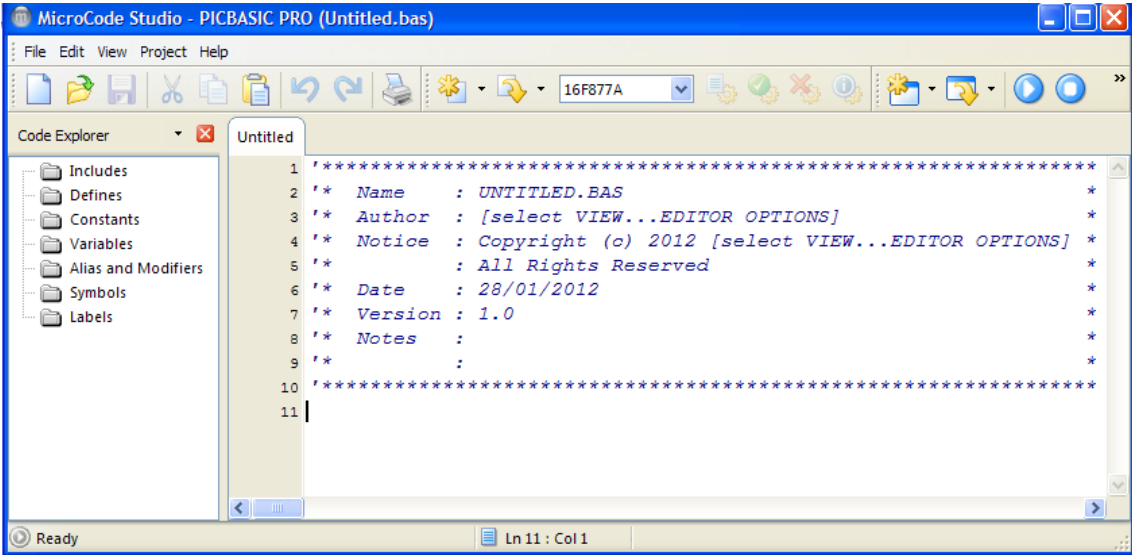


Figura 4.2.Pantalla principal con las líneas de programación

También podemos predefinir la carpeta en la que deseamos guardar los archivos *.pbp, por ejemplo podría ser una ubicación donde previamente

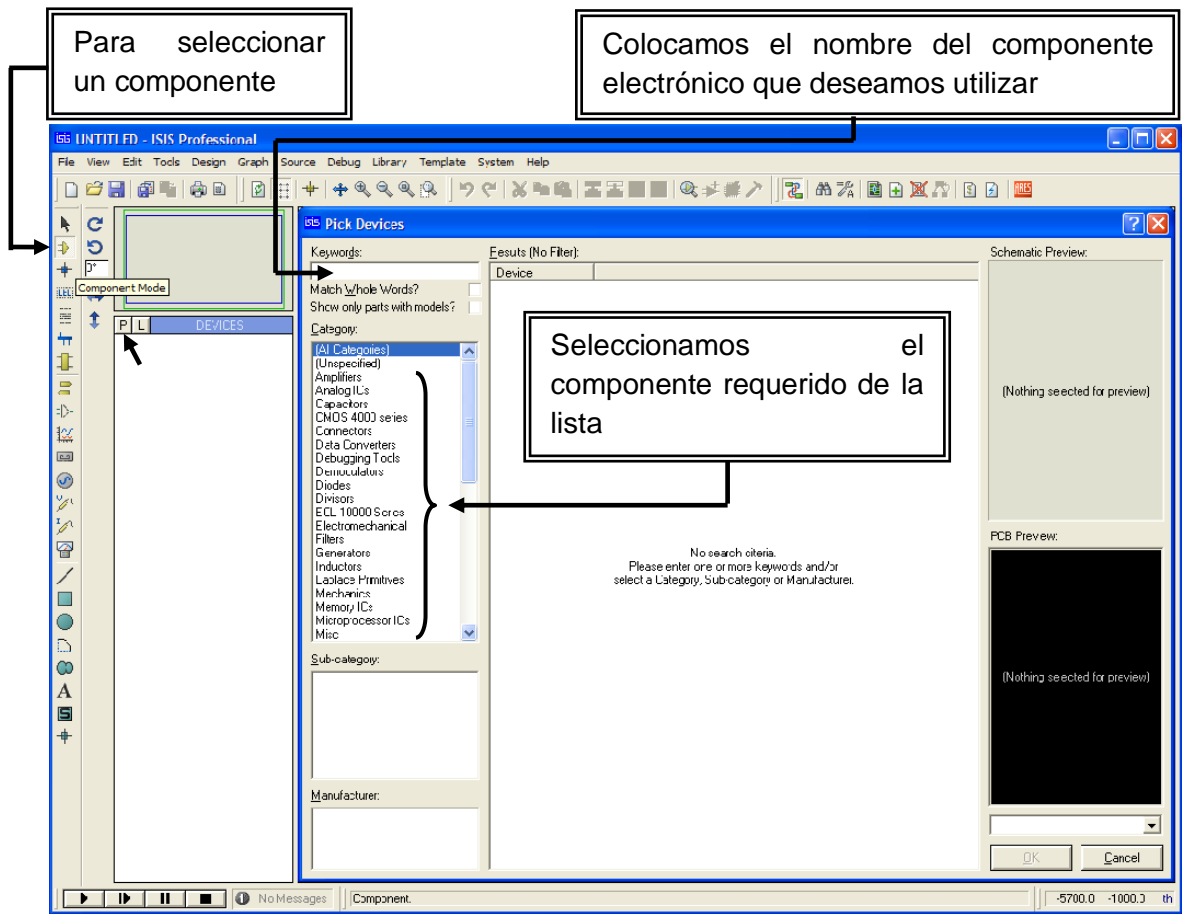



Figura 4.4. Selección de componentes en Isis

- Hacemos clic en  que se encuentra en la barra lateral de la pantalla principal, y en la que permite la selección de componentes de cualquier naturaleza (A/D).
- Ahora debemos hacer clic en **P** para que se desplace la pantalla en la que se encuentran los componentes (A/D).
- En la pantalla de componentes podemos seleccionar; ubicando el código o nombre del componente debajo de *keywords*, o directamente buscándolos en la lista que se desplaza debajo de *keywords*.

- Una vez identificado y seleccionado el componente requerido presionamos OK y lo ubicamos en la pantalla principal de simulación.

4.3.2. Títulos y Etiquetado de Componentes

Para ubicar una etiqueta sobre un componente o en general en ISIS, y colocarla en la pantalla principal de simulación, es necesario:

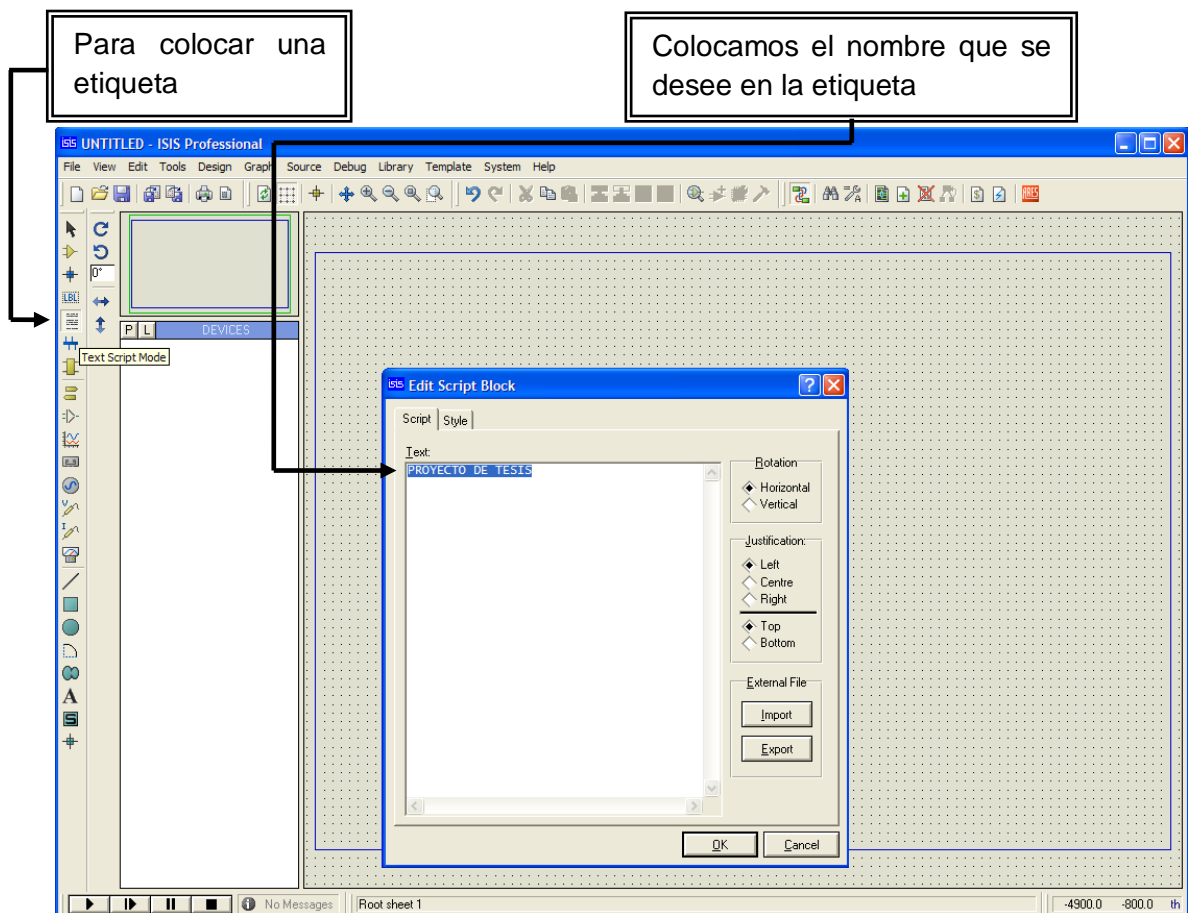



Figura 4.5. Títulos y Etiquetas en ISIS.

- Hacemos clic en  que se encuentra en la barra lateral de la pantalla principal y que es la única que nos permite ubicar cualquier texto en la pantalla de simulación.

- Al desplazarse la pantalla colocamos el TEXTO que se requiere, presionamos OK y lo colocamos en la pantalla principal en donde se necesita.

4.3.3. Selección de Instrumentos de Medida

Al realizar una simulación en ISIS siempre se vuelve necesaria la utilización de instrumentos de medida, para poder determinar con exactitud los posibles problemas que se presenten y corregirlos oportunamente. Ubicamos los instrumentos de medida en la pantalla, así:

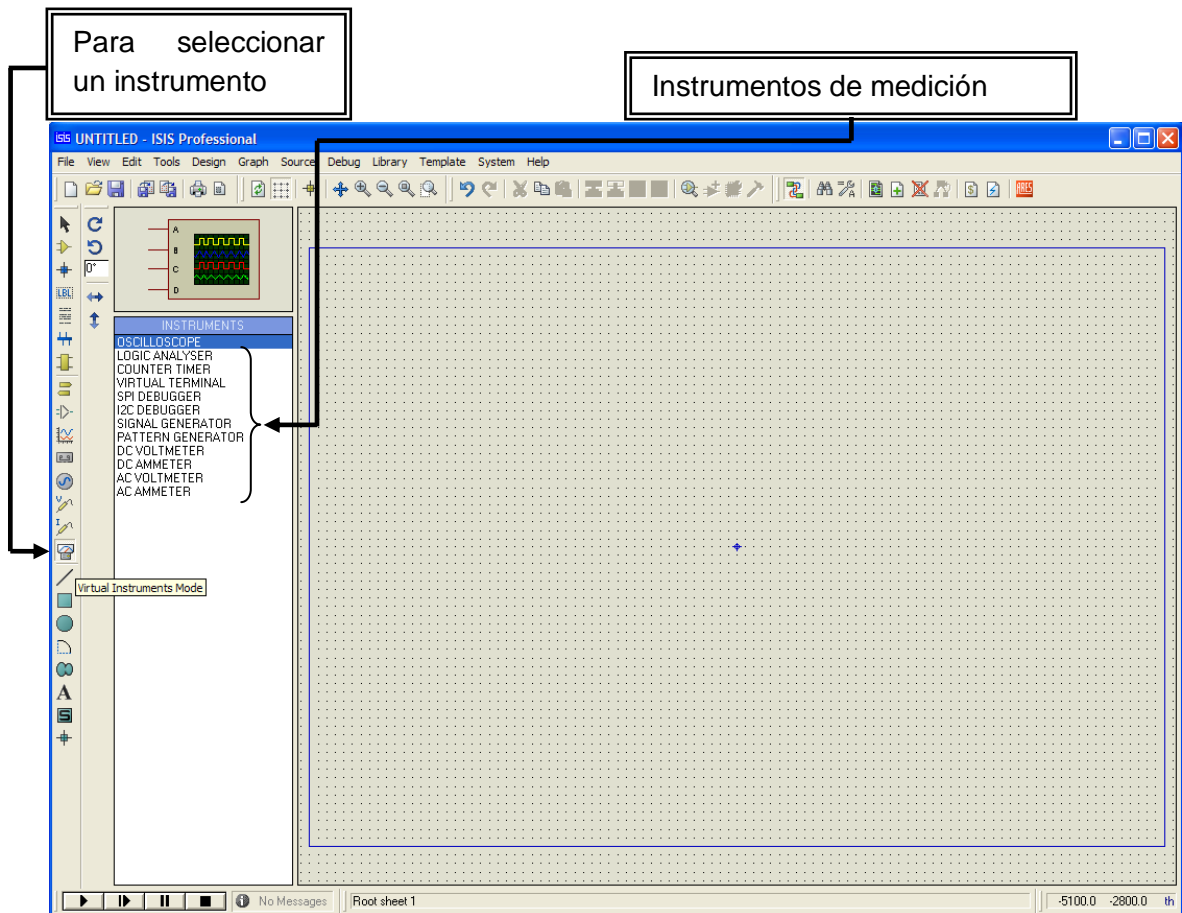



Figura 4.6. Instrumentos de Medida en ISIS.

- Hacemos clic en  que se encuentra en la barra lateral de la pantalla principal y que permite visualizar todos los instrumentos de medida, para ubicarlos en la pantalla de simulación.
- Seleccionamos de la lista desplazada el instrumento requerido, presionamos OK y lo colocamos en la pantalla principal en donde se necesite.

4.3.4. Selección de Fuentes de Alimentación

En toda simulación realizada en ISIS, sobre todo en simulaciones analógicas se vuelve necesaria la utilización de Fuentes de Alimentación (DC/AC), para energizar el circuito. Las podemos ubicar, así:

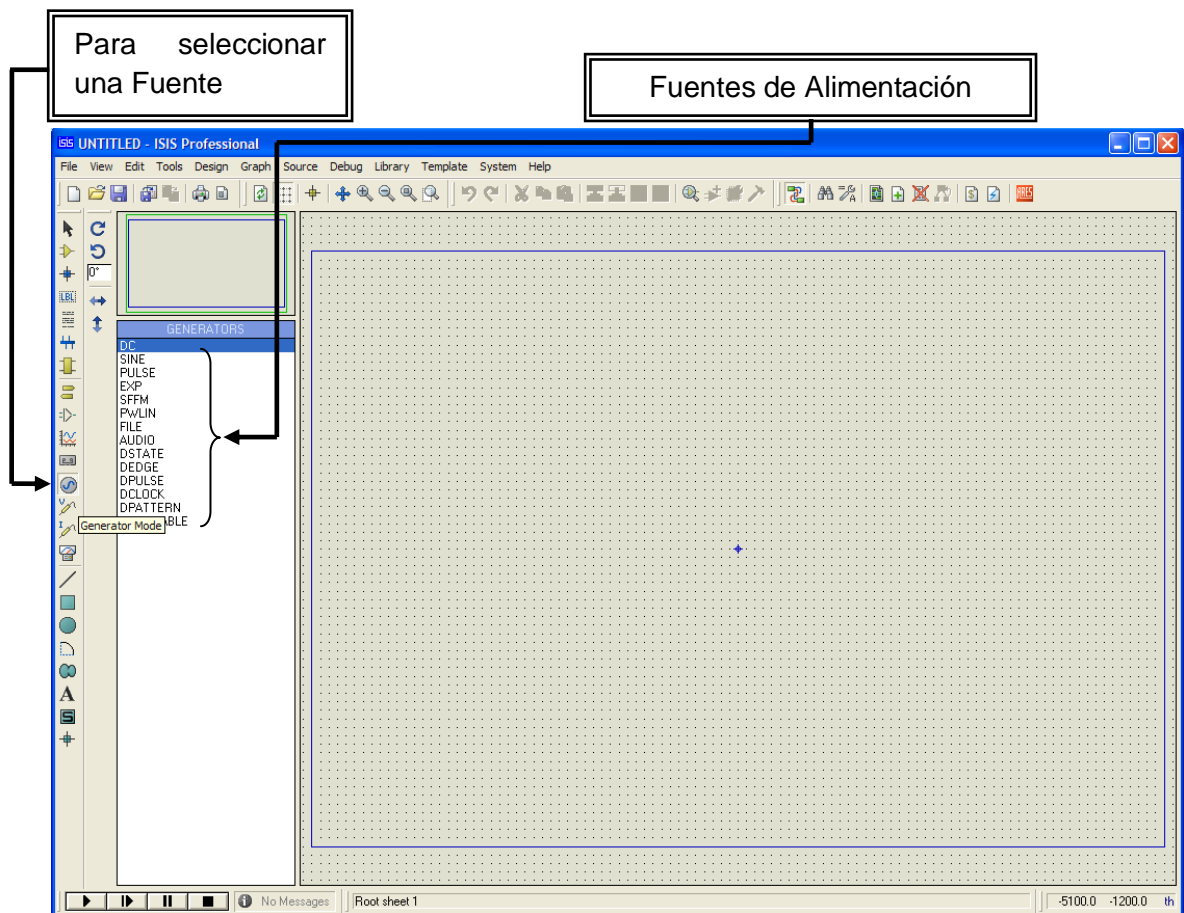
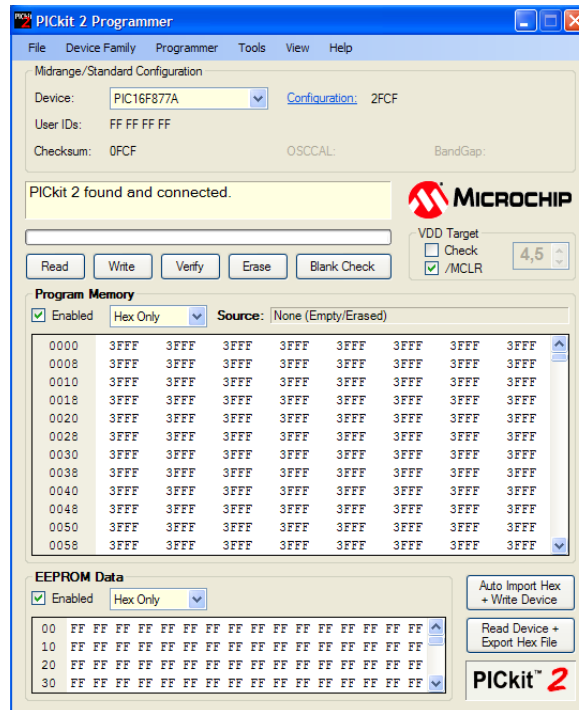
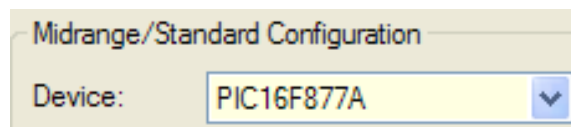


Figura 4.7. Fuentes de Alimentación en ISIS.

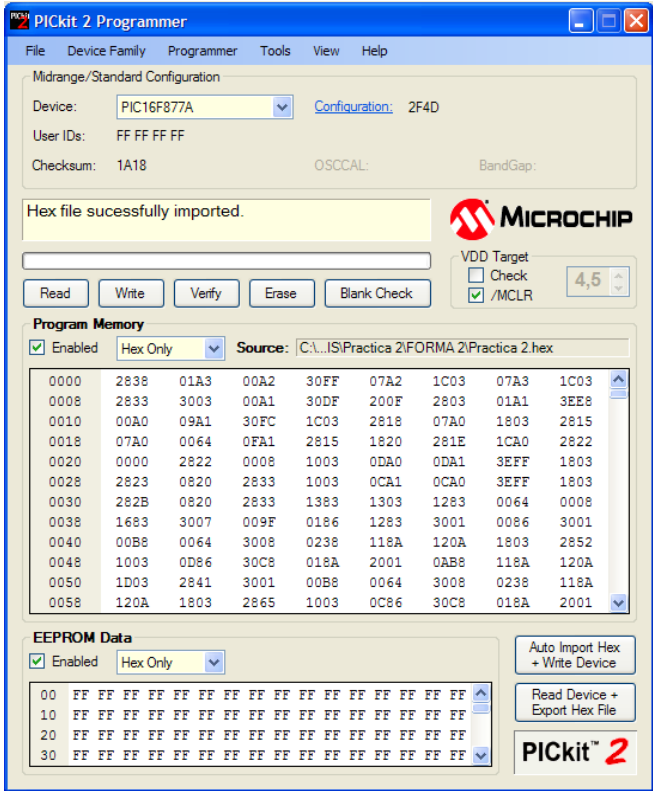
- El quemador de PIC será detectado automáticamente, al igual que el dispositivo a ser programado (siempre y cuando pertenezca a una de las familias que soporten autodetección). Para que la autodetección del dispositivo tenga efecto, el comando *Programmer* → *Manual DeviceSelect* debe estar desactivado.



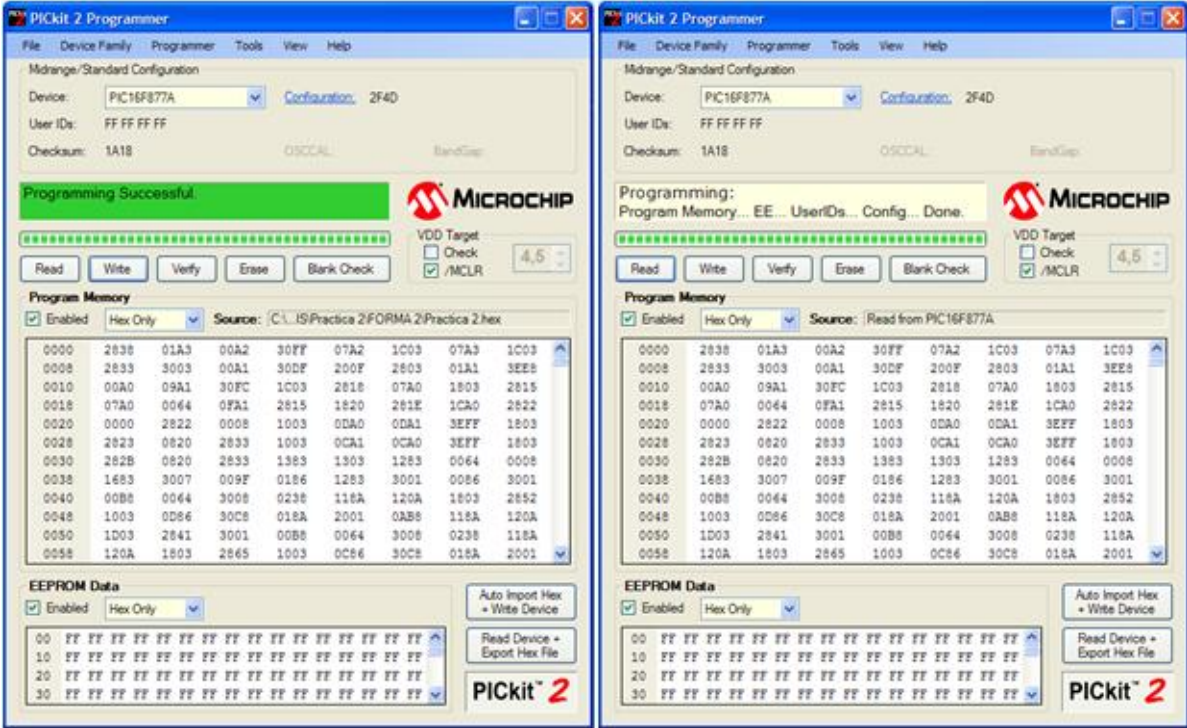
- Si el dispositivo no soporta autodetección, debe ser seleccionado manualmente, para lo cual el comando *Programmer* → *Manual DeviceSelect* debe estar activado. Por ejemplo, para el PIC16F677A se debe seleccionar el comando *DeviceFamily* → *Midrange* → *Standard*, y a continuación seleccionar el PIC de la lista desplegable *Device*:



- Con el comando *File* → *ImportHex* abra el archivo ejecutable (*.hex) que va a ser grabado en el dispositivo. Deberá observar el siguiente mensaje.



➤ Programe el dispositivo haciendo clic en el botón (Write). Espere hasta que aparezca el mensaje de programación exitosa, como se muestra en la figura pudiendo ser en **Windows XP ó 7**:



- Desconecte el cable USB del computador y extraiga el dispositivo programado.
- Si va a programar otro dispositivo, conéctelo al grabador de PIC USB, conecte nuevamente el cable USB al computador y luego seleccione el comando *Tools* → *CheckCommunication*.

Nota: Los dispositivos se encuentran agrupados por familias, así que si no logra encontrarlo en una de las familias vaya al menú *DeviceFamily* y búsquelo en las otras familias.

4.5. MÓDULO DE PRÁCTICAS PARA MICROCONTROLADORES

4.5.1. Diseño PCB de la Fuente de Alimentación

Una vez realizado los circuitos de las fuentes de alimentación, procedemos a realizar el diseño de los ruteados de las pistas en PCB Wizard, a partir de lo cual tenemos:



Figura 4.8. Vista 3D de la Fuente de Poder

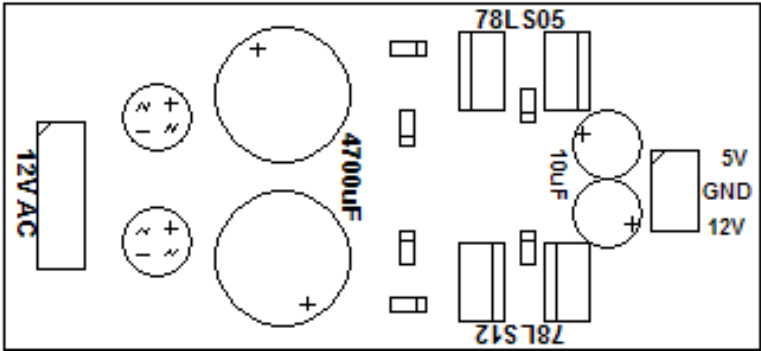


Figura 4.9. Etiquetas de la Fuente de Poder

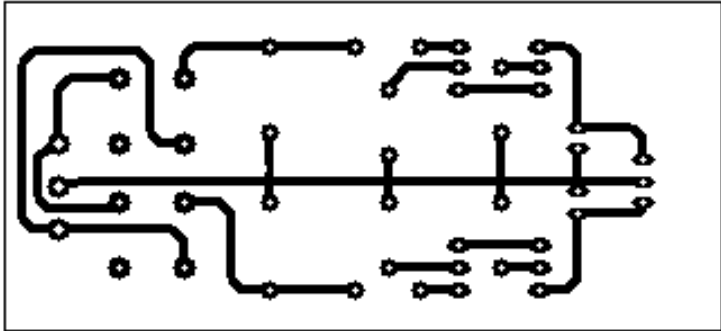


Figura 4.10. Ruteado de la Fuente de Poder

4.5.2. Diseño del Módulo de Prácticas

Una vez culminadas las prácticas planteadas en el presente proyecto de tesis, ante la necesidad de demostrar con elementos reales el funcionamiento de las mismas, se diseñó un módulo capaz de implementar cualquier tipo de prácticas con:

- Leds.
- Displays.
- Matriz de leds de 5x7.
- Relés.
- Teclado matricial 4x4.
- Salidas Analógicas para Audio, etc.

4.1.3. Diseño PCB del Módulo de Prácticas.

Para realizar el circuito impreso del módulo de prácticas, es necesario primero elaborar un dibujo con las pistas conductoras, en donde se soldarán cada uno de los elementos electrónicos del circuito impreso.

Esto lo podemos realizar con la ayuda de cualquiera del software de dibujo como: ARES de Proteus Profesional, Corel Draw, Orcad, Protel, PCB Wizard (en nuestro caso).

En la figura de a continuación se pueden observar los diagramas de las pistas y el screen de los diferentes elementos, que con la ayuda de los métodos de transferencia térmica, fotograbado, o dibujando directamente en la placa, transferimos el dibujo a la placa.

Para reducir o quitar el cobre de la placa, en las partes que no se encuentran protegidas por las líneas de tinta o película protectora, se emplea dos cuartos de cloruro férrico diluidos en 2 tazas de agua (H_2O), se sumerge la placa por algunos minutos hasta que haya salido todo el cobre que no se encuentra protegido.

Después se realizan las perforaciones para la correcta instalación y soldadura de los elementos en el módulo.

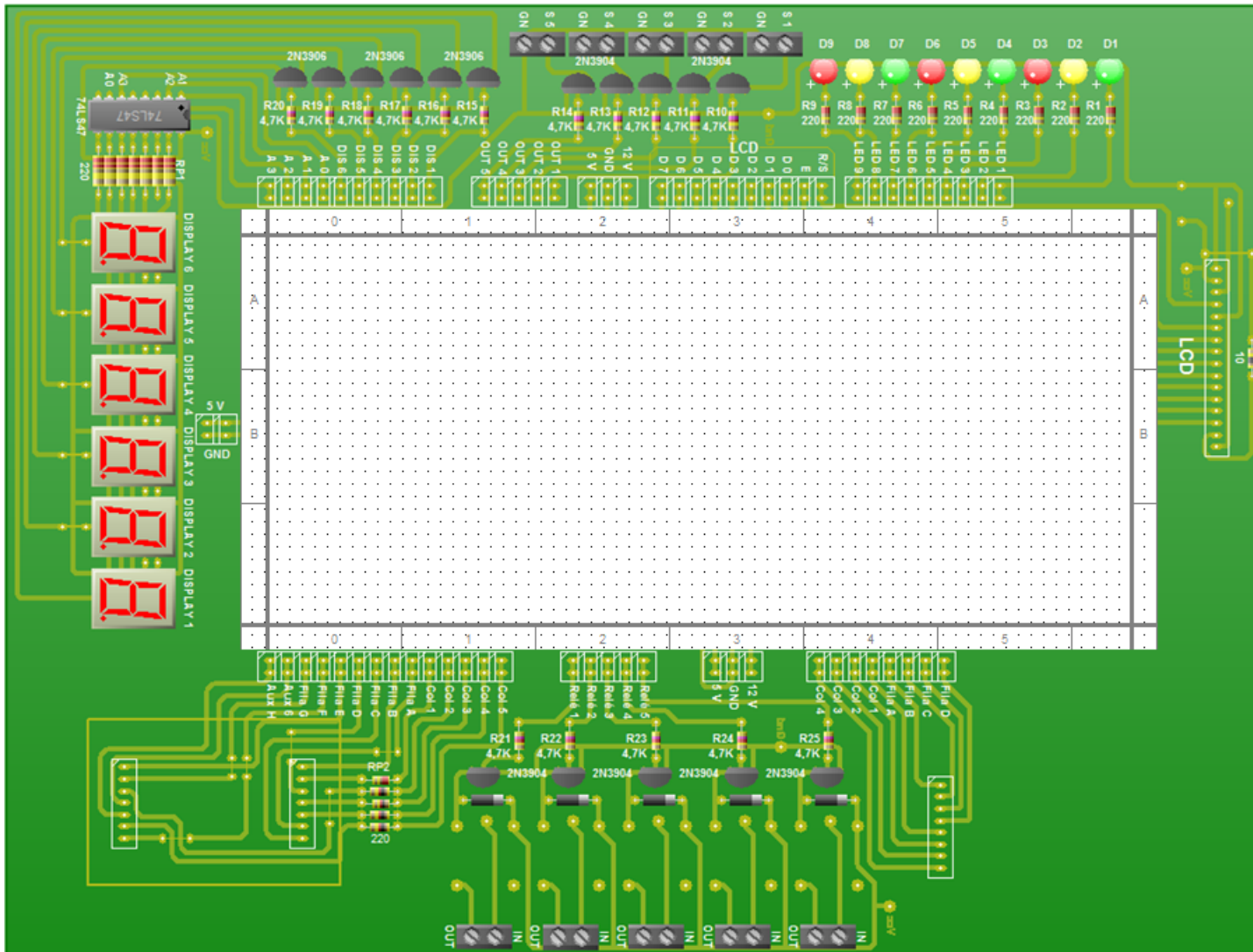


Figura 4.11. Vista 3D del Módulo de Prácticas

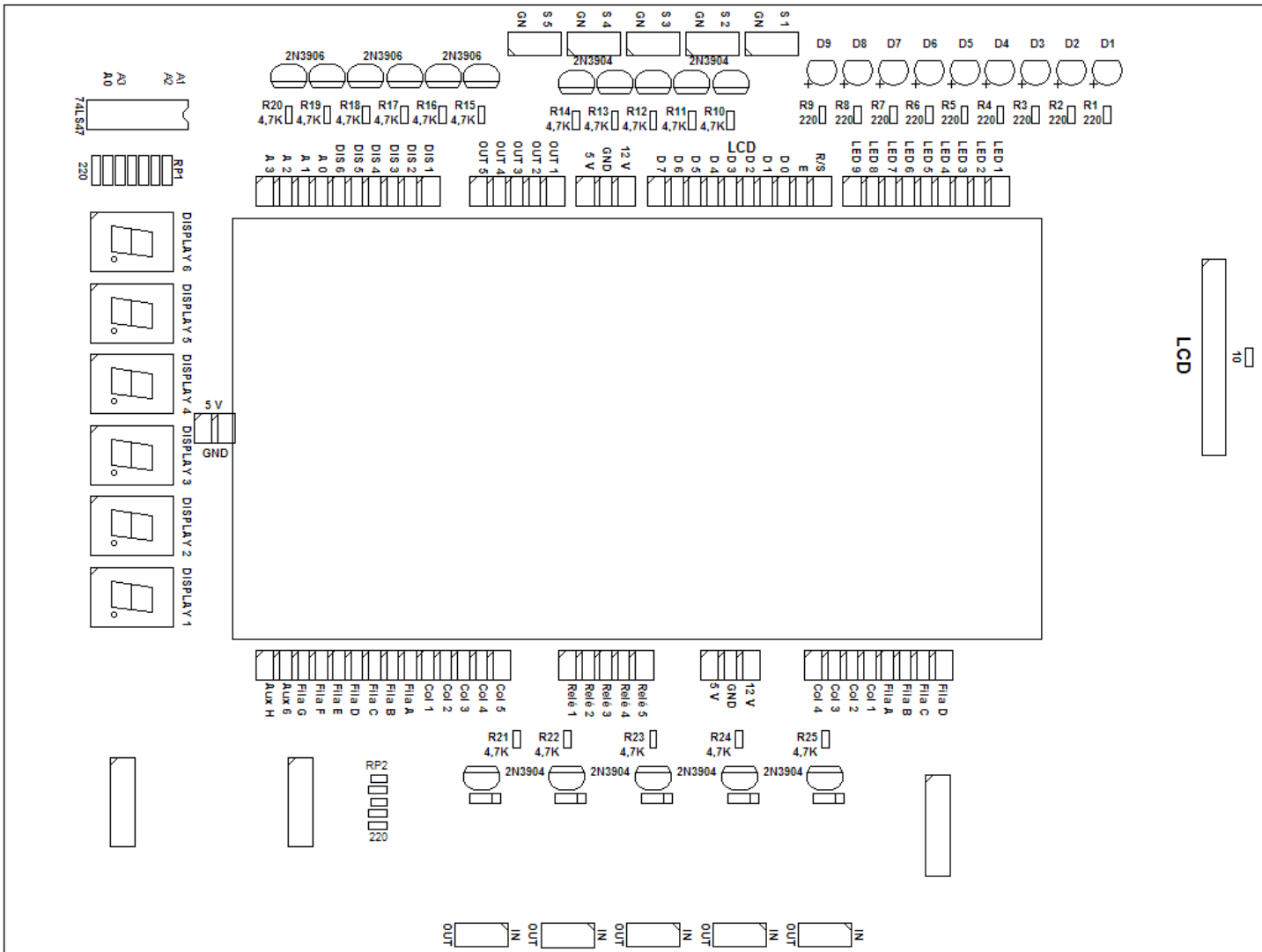


Figura 4.12. Etiquetas del Módulo de Prácticas

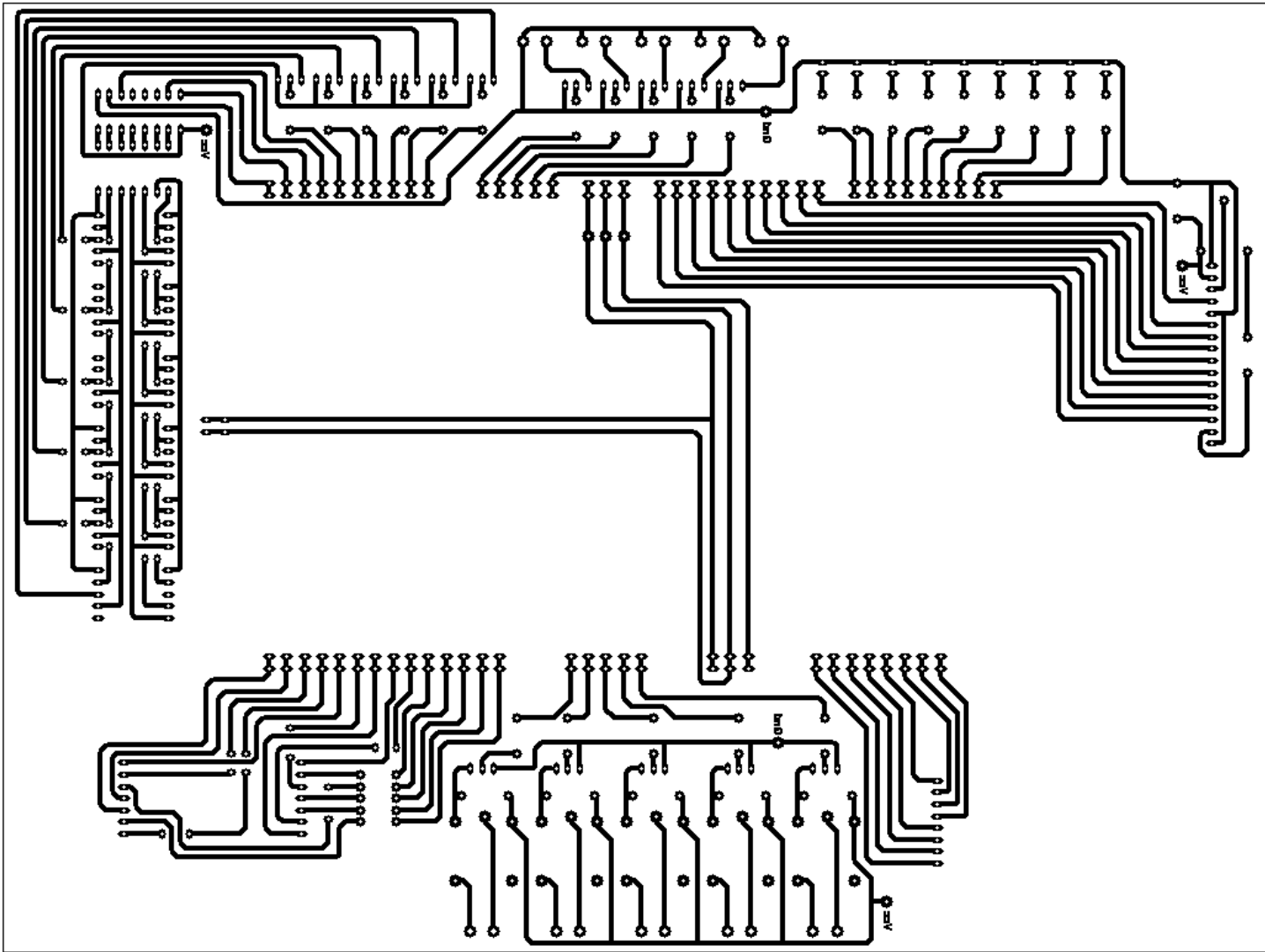


Figura 4.13. Ruteado del Módulo de Prácticas

V. RESULTADOS

5.1. PRÁCTICAS PROPUESTAS PARA EL MÓDULO

En el presente proceso investigativo, es imperiosa también la necesidad de plantear PRÁCTICAS con el afán de aplicar cada concepto aprendido.

Las prácticas planteadas en el presente capítulo, han sido diseñadas estrictamente bajo el precepto de utilizar cada sección que contiene el módulo diseñado.

Las prácticas contienen y mantienen una estructura de tal forma que cualquier lector del presente trabajo investigativo pueda implementarla de manera correcta y adecuada, sin ningún tipo de errores en el módulo diseñado.

A continuación se encuentran desarrolladas las prácticas, de manera detallada y paso a paso, para que sea de gran utilidad para los estudiantes universitarios que deseen utilizarla como una guía de aprendizaje, en la simulación de circuitos electrónicos con ISIS de Proteus Profesional, y; la implementación de programas en Microcode Studio.

PRACTICA 1

1. TEMA:

Parpadeo de Leds en Intervalos de un Segundo

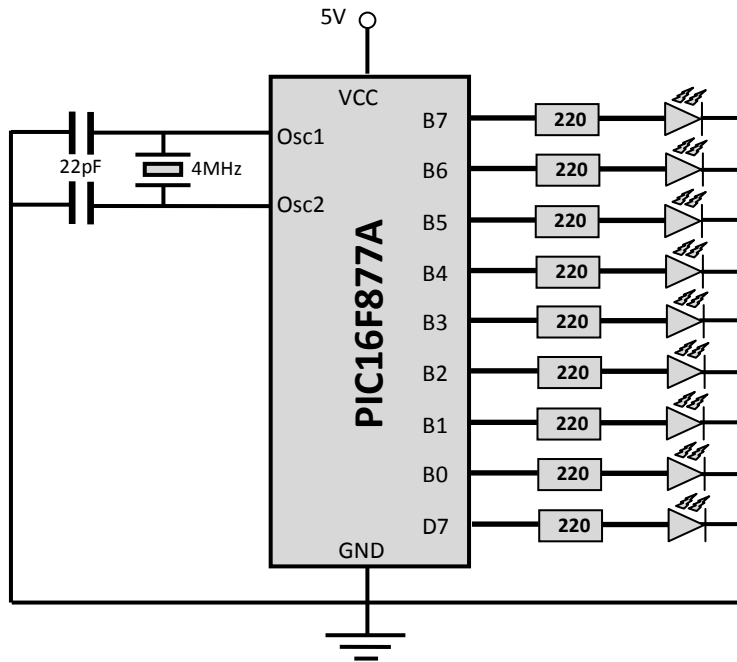
2. OBJETIVOS:

- Diseñar y armar un circuito en el software de simulación ISIS de PROTEUS, que permita simular el parpadeo de nueve leds con intervalos de un segundo.
- Diseñar un algoritmo y un programa en el software de programación MICROCODE, que permita hacer parpadear nueve leds con intervalos de un segundo.
- Analizar el comportamiento del circuito y el algoritmo del programa.
- Comprobar el funcionamiento del circuito.

3. MATERIALES:

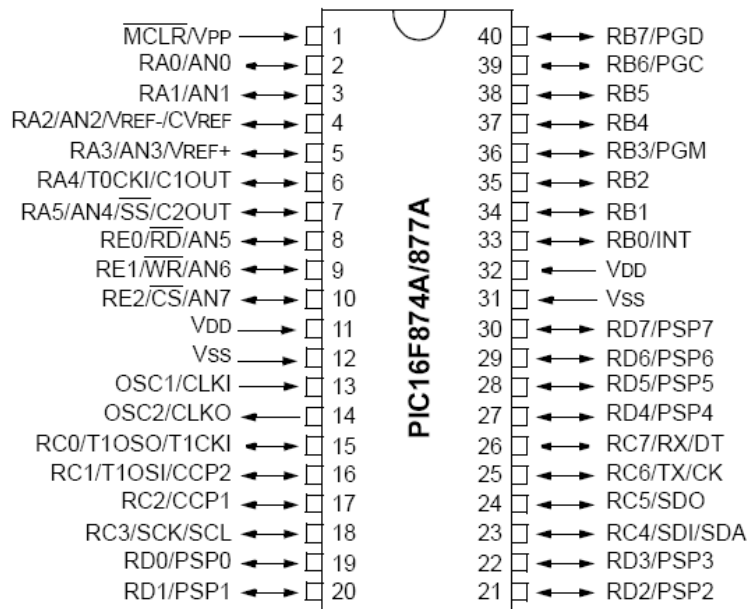
Cantidad	Designación	Características
9	Leds	3 Rojos, 3 Verdes, 3 Amarillos
9	Resistores	220Ω (1/4W)
1	Resistor	4,7kΩ (1/4W)
1	Pulsador NA	
1	Microcontrolador	PIC (16F877A ó 16F628A)
2	Capacitores Cerámicos	22pF ó 33pF
1	Cristal de Cuarzo	4MHz

4. ESQUEMA:



5. SISTEMA CATEGORIAL:

5.1. Datasheet 16F877A.



5.2. Cristal de Cuarzo.



Los cristales de cuarzo son ampliamente utilizados en los equipos que utilizan los radioaficionados. Basta dar una rápida ojeada al esquema de cualquiera de nuestros transceptores para encontrar varios de ellos.

Estos dispositivos están formados por una fina lámina de cuarzo situada entre dos electrodos. Como es sabido, el cuarzo, también llamado cristal de roca, es un mineral compuesto por silicio y oxígeno, (óxido anhidro de silicio, bióxido de silicio o anhídrido silícico, SiO_2) cuyos cristales tienen forma de prisma hexagonal terminado por dos romboedros que parecen una bipirámide hexagonal. El cuarzo es el mineral más difundido en la corteza terrestre, bien en forma de cristales o formando parte otras rocas, como el granito (cuarzo, feldespato y mica). En la figura número uno se puede ver un grupo de cristales de cuarzo tal como aparecen en la naturaleza.

5.3. Que es un Algoritmo.

Un algoritmo es un sistema por el cual se llega a una solución, teniendo en cuenta que debe de ser definido, finito y preciso. Por preciso entendemos que cada paso a seguir tiene un orden; finito implica que tiene un determinado número de pasos, o sea, que tiene un fin; y definido, que si se

sigue el mismo proceso más de un vez llegaremos al mismo resultado.

Estructura Básica:

- Inicio
- Datos de entrada (operaciones básicas)
- Datos de entrada (operaciones básicas)
- *Procesamiento de los datos*
- Datos de salida
- Fin

Implementación

La función principal que posee realizar la traza de un algoritmo es la de comprobar que éste funciona correctamente o para realizar la etapa de depuración en la que se intenta corregir errores, simplificar el algoritmo al máximo e incrementar su eficacia y velocidad.

6. PROCEDIMIENTO:

6.1. Para armar el circuito de simulación utilizamos ISIS de Proteus Profesional y su interfaz.

6.1.1. En el buscador de Proteus seleccionamos el Pic16F877A y lo ubicamos en la pantalla principal.

6.1.2. En el buscador de Proteus seleccionamos el cristal de cuarzo, lo ubicamos en la pantalla principal y lo conectamos al oscilador 1 y 2 del PIC.

6.1.3. En el buscador de Proteus seleccionamos dos capacitores de 22pF ó 33pF, los ubicamos en la pantalla principal y los conectamos al oscilador 1 y 2 del PIC y a GND.

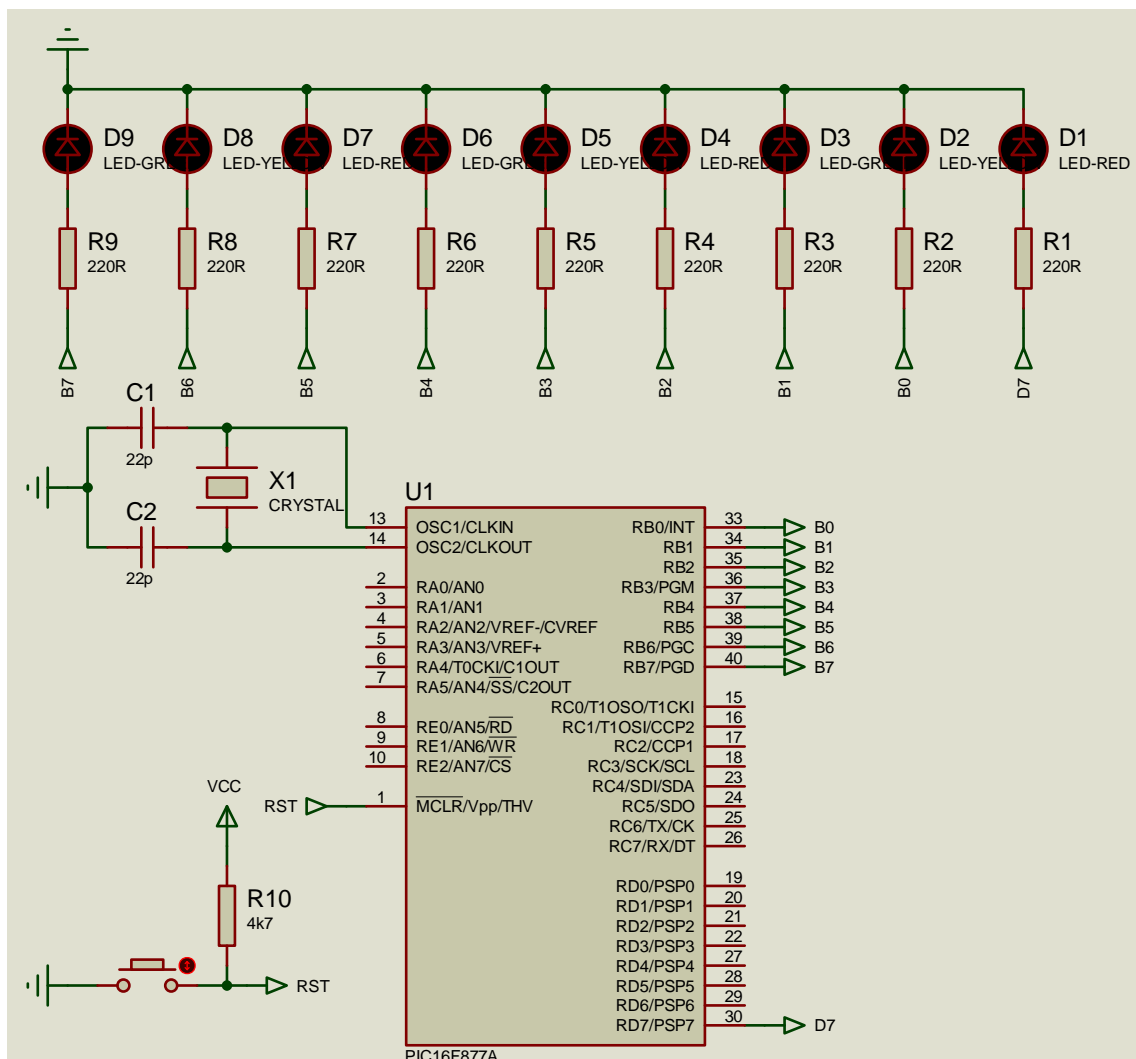
- 6.1.4. En el buscador de Proteus seleccionamos una resistencia de 220Ω , ubicamos en la pantalla principal nueve de las mismas y las conectamos a los pines puerto B y D7.
 - 6.1.5. En el buscador de Proteus seleccionamos un led rojo, amarillo y verde, ubicamos tres de cada uno en la pantalla principal y los conectamos a las resistencias de 220Ω y a GND.
 - 6.1.6. En el buscador de Proteus seleccionamos una resistencia de $4,7k\Omega$, la ubicamos en la pantalla principal y la conectamos al reset del PIC y a Vcc.
 - 6.1.7. En el buscador de Proteus seleccionamos un pulsador, lo ubicamos en la pantalla principal y lo conectamos al reset del PIC y a GND.
- 6.2. Para la programación del PIC utilizamos Microcode Studio.
- 6.2.1. Transformamos a digitales todos los puertos analógicos del PIC.
 - 6.2.2. Declaramos el puerto B como salida.
 - 6.2.3. Le damos el nombre **led** al **puerto D.7** del PIC.
 - 6.2.4. Le asignamos un nombre al programa.
 - 6.2.5. Encendemos los nueve leds conectados al puerto B y D7.
 - 6.2.6. Le damos una pausa al programa de 1000 milisegundos.
 - 6.2.7. Apagamos los nueve leds conectados al puerto B y D7.
 - 6.2.8. Le damos una nueva pausa al programa de 1000 milisegundos.
 - 6.2.9. Repetimos el programa de manera indefinida.
- 6.3. Antes de la ejecución del programa en Proteus es indispensable sincronizar los osciladores del cristal y PIC en 4MHz, seteando dicho valor en propiedades de los dispositivos.

6.4. Cargamos el archivo **.HEX** generado por Microcode en el PIC de Proteus y ejecutamos la simulación, y los nueve leds deben parpadear en intervalos de un segundo.

6.5. Para armar la práctica en el módulo es necesario comprar los materiales descritos en la lista y adicionalmente traer un pelacables y cable UTP para conexiones.

7. DESARROLLO:

El circuito armado completamente y la simulación realizada en Proteus Profesional queda de la siguiente forma:



El algoritmo correspondiente y programa correctamente elaborado en Microcode Studio para hacer parpadear los nueve leds armados en la simulación de Proteus nos queda de la siguiente forma:

```

                                ;PROGRAMA PARA ENCENDER 9 LEDS
ADCON1 = 7                       ; Transforma en digitales los puertos analógicos
Trisb = 0                         ; Declaramos el puerto B como salida
ledVAR portd.7                   ; Le asignamos el nombre de led al puerto D.7

  Practica1:
HIGHled :portb = %11111111      ; Encendemos los 9 leds
PAUSE 1000                       ; Pausamos 1000 milisegundos
LOWled :portb = %00000000       ; Apagamos los 9 leds
PAUSE 1000                       ; Pausamos 1000 milisegundos
GOTO Practica1                   ; Retornar a Practica1

  END                             ; Fin de la Programación

```

8. PREGUNTAS DE CONTROL:

8.1. ¿Por qué es importante consultar el Datasheet del PIC?

El datasheet es el único documento en que los diseñadores de los PIC, detallan los Pines de conexión, además de los niveles de corriente, voltaje y temperatura que éstos soportan.

8.2. ¿Qué importancia tiene el Watiaje de las resistencias al momento de armar el circuito?

De la fórmula $P=I^2R$, se deduce que el Watiaje requerido por el resistor depende del nivel de corriente que el circuito diseñado consuma.

8.3. ¿Por qué Microcode y Proteus pueden interactuar entre sí?

Porque Microcode Studio genera un archivo **.HEX**, que tiene un lenguaje de ensamblador, mismo que puede ser interpretado por cualquier simulador o programador de PIC's.

8.4. **¿Qué consideraciones se debe tener para la selección de un PIC?**

Depende de las necesidades del circuito a diseñar, y del número de entradas y salidas que el mismo requiera.

PRACTICA 2

1. TEMA:

Desplazamiento de Leds Tipo Auto Fantástico.

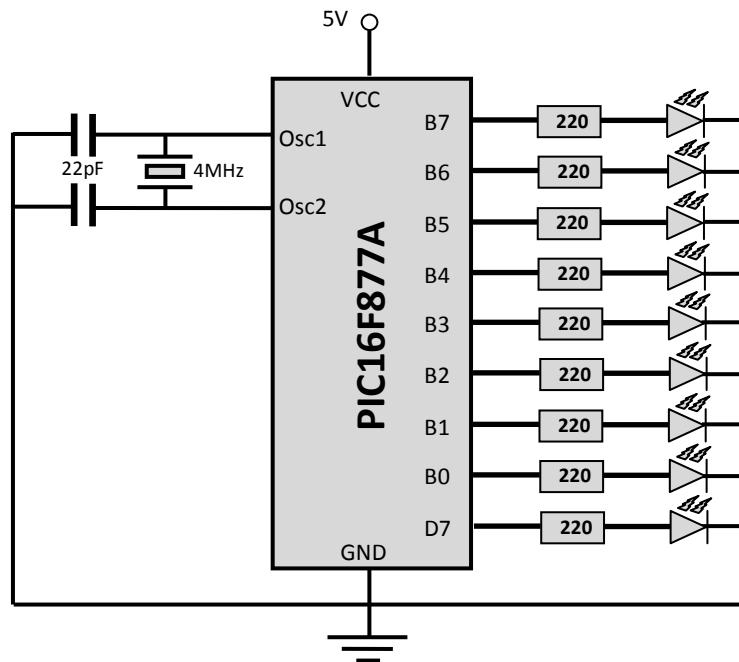
2. OBJETIVOS:

- Diseñar y armar un circuito en el software de simulación PROTEUS, que permita simular el desplazamiento de leds tipo auto fantástico.
- Diseñar un algoritmo y un programa en el software de programación MICROCODE, que permita el desplazamiento de leds tipo auto fantástico.
- Analizar el comportamiento del circuito y el algoritmo del programa.

3. MATERIALES:

Cantidad	Designación	Características
9	Leds	3 Rojos, 3Verdes, 3 Amarillos
9	Resistores	220Ω (1/4W)
1	Resistor	4,7kΩ (1/4W)
1	Pulsador	NA
1	Microcontrolador	PIC (16F877A ó 16F628A)
2	Capacitores Cerámico	22pF ó 33pF
1	Cristal de Cuarzo	4MHz

4. ESQUEMA:



5. SISTEMA CATEGORIAL:

5.1. Descripción de Puertos del Microcontrolador Pic 16877A

Puerto A:

- Puertos de E/S de 6 pines
- RA0/ANO – PIN 2
- RA1/AN1 – PIN 3
- RA2/AN2 y Vref(-) – PIN 4
- RA3/AN3 y Vref(+) – PIN 5
- RA4(salida de colector abierto) y TOCKI(Entrada de reloj del módulo Timer0) – PIN 6
- RA5/AN4 y SS(Selección esclavo para el puerto serie síncrono) – PIN

Puerto B:

- Puertos de E/S de 8 pines

- Resistencias pull-up programables
- RB0(Interrupción interna) – PIN 33
- RB4-7(Interrupción por cambio de flanco)
- RB4, RB7 y RB3(programación y debugger in circuit)

Puerto C:

- Puertos de E/S de 8 pines
- RC0/T1OSO(Timer1 salida oscilador) y T1CKI(Entrada de reloj del modulo Timer1) – PIN 15
- RC1-RC2(PWM/COMP/CAPT)
- RC1/T1OSI(Estrada osc timer1) – PIN 16
- RC3-4/IIC
- RC3-5/SPI
- RC6-7/USART

Puerto D:

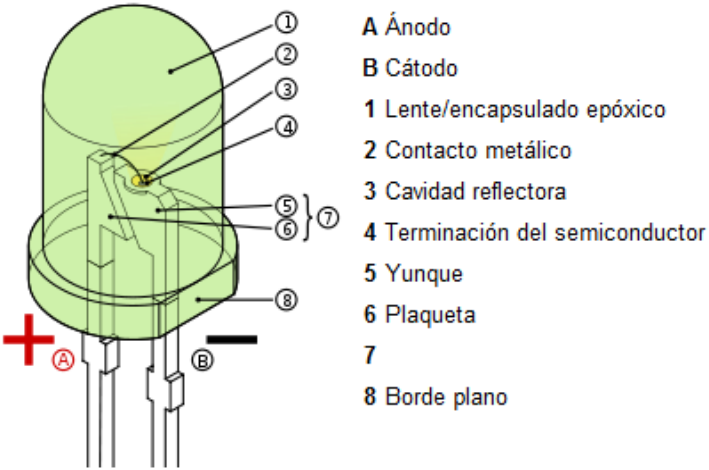
- Puertos de E/S de 8 pines
- Bus de datos en PPS(Puerto paralelo esclavo)

Puerto E:

- Puertos de E/S de 3 pines
- RE0/AN5 y Read de PPS – PIN 8
- RE1/AN6 y Write de PPS – PIN 9
- RE2/AN7 y CS de PPS – PIN 10

5.2. DIODOS LED

Un **led** (de la sigla inglesa *LED: Light-Emitting Diode*: ‘diodo emisor de luz’, también ‘diodo luminoso’) es un diodo semiconductor que emite luz. Se usan como indicadores en muchos dispositivos, y cada vez con mucha más frecuencia, en iluminación.



Cuando un led se encuentra en polarización directa, los electrones pueden recombinarse con los huecos en el dispositivo, liberando energía en forma de fotones. Este efecto es llamado electroluminiscencia y el color de la luz (correspondiente a la energía del fotón) se determina a partir de la banda de energía del semiconductor. Por lo general, el área de un led es muy pequeña (menor a 1 mm²), y se pueden usar componentes ópticos integrados para formar su patrón de radiación.



6. PROCEDIMIENTO:

6.1. Para armar el circuito de simulación utilizamos ISIS de Proteus

Profesional y su interfaz.

6.1.1. En el buscador de Proteus seleccionamos el Pic16F877A y lo ubicamos en la pantalla principal.

6.1.2. En el buscador de Proteus seleccionamos el cristal de cuarzo, lo ubicamos en la pantalla principal y lo conectamos al oscilador 1 y 2 del PIC.

6.1.3. En el buscador de Proteus seleccionamos dos capacitores de 22pF ó 33pF, los ubicamos en la pantalla principal y los conectamos al oscilador 1 y 2 del PIC y a GND.

6.1.4. En el buscador de Proteus seleccionamos una resistencia de 220 Ω , ubicamos en la pantalla principal nueve de las mismas y las conectamos a los pines puerto B y D7.

6.1.5. En el buscador de Proteus seleccionamos un led rojo, amarillo y verde, ubicamos tres de cada uno en la pantalla principal y los conectamos a las resistencias de 220 Ω y a GND.

6.1.6. En el buscador de Proteus seleccionamos una resistencia de 4,7k Ω , la ubicamos en la pantalla principal y la conectamos al reset del PIC y a Vcc.

6.1.7. En el buscador de Proteus seleccionamos un pulsador, lo ubicamos en la pantalla principal y lo conectamos al reset del PIC y a GND.

6.2. Para la programación del PIC utilizamos Microcode Studio.

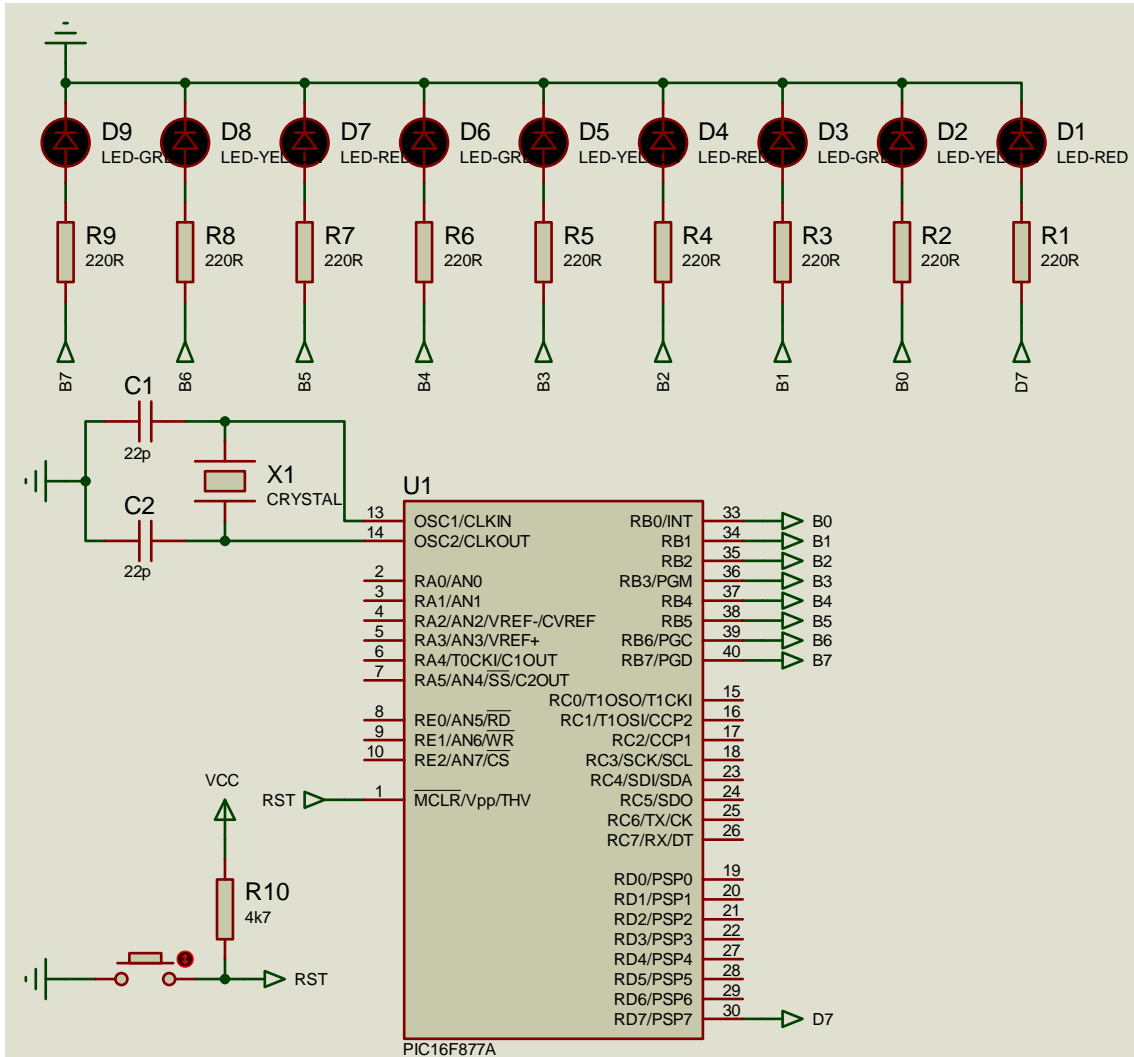
6.2.1. Transformamosa digitales todos los puertos analógicos del PIC.

- 6.2.2. Declaramos el puerto B como salida.
 - 6.2.3. Le damos el nombre *led* al **puerto D.7** del PIC.
 - 6.2.4. Le asignamos un nombre al programa.
 - 6.2.5. Encendemos uno a uno los nueve leds conectados al puerto B y D7.
 - 6.2.6. Le damos una pausa al programa de 200 milisegundos.
 - 6.2.7. Apagamos uno a uno los nueve leds conectados al puerto B y D7.
 - 6.2.8. Le damos una nueva pausa al programa de 200 milisegundos.
 - 6.2.9. Repetimos el programa de manera indefinida.
- 6.3. Antes de la ejecución del programa en Proteus es indispensable sincronizar los osciladores del cristal y PIC en 4MHz, seteando dicho valor en propiedades de los dispositivos.
- 6.4. Cargamos el archivo **.HEX** generado por Microcode en el PIC de Proteus y ejecutamos la simulación, y los nueve leds deben parpadear en intervalos de un segundo.
- 6.5. Para armar la práctica en el módulo es necesario comprar los materiales descritos en la lista y adicionalmente traer un pelacables y cable UTP para conexiones.

7. DESARROLLO:

El circuito armado completamente y la simulación realizada en Proteus

Profesional queda de la siguiente forma:



El algoritmo correspondiente y programa correctamente elaborado en Microcode Studio para hacer parpadear los nueve leds armados en la simulación de Proteus nos queda de la siguiente forma:

; PROGRAMA PARA LUCES DE AUTO FANTÁSTICO (Forma 1)

```

ADCON1 = 7                ; transforma en digitales los puertos analógicos
Trisb = 0                 ; declaramos el puerto B como salida
ledVAR portd.7           ; Le asignamos el nombre de led al puerto D.7
  Practica2:
HIGHled :portb = %00000000; encendemos el puerto D7
PAUSE 200                 ; esperar 200 milisegundos
portb = %00000001        ; encendemos el puerto B0
  PAUSE 200               ; esperar 200 milisegundos
portb = %00000011        ; encendemos el puerto B1
  PAUSE 200               ; esperar 200 milisegundos
portb = %00000111        ; encendemos el puerto B2
  PAUSE 200               ; esperar 200 milisegundos
portb = %00001111        ; encendemos el puerto B3
  PAUSE 200               ; esperar 200 milisegundos
portb = %00011111        ; encendemos el puerto B4
  PAUSE 200               ; esperar 200 milisegundos
portb = %00111111        ; encendemos el puerto B5
  PAUSE 200               ; esperar 200 milisegundos
portb = %01111111        ; encendemos el puerto B6
  PAUSE 200               ; esperar 200 milisegundos
portb = %11111111        ; encendemos el puerto B7
  PAUSE 200               ; esperar 200 milisegundos
portb = %01111111        ; apagamos el puerto B7
  PAUSE 200               ; esperar 200 milisegundos
portb = %00111111        ; apagamos el puerto B6
  PAUSE 200               ; esperar 200 milisegundos
portb = %00011111        ; apagamos el puerto B5
  PAUSE 200               ; esperar 200 milisegundos
portb = %00001111        ; apagamos el puerto B4
  PAUSE 200               ; esperar 200 milisegundos
portb = %00000111        ; apagamos el puerto B3
  PAUSE 200               ; esperar 200 milisegundos
portb = %00000011        ; apagamos el puerto B2
  PAUSE 200               ; esperar 200 milisegundos
portb = %00000001        ; apagamos el puerto B1
  PAUSE 200               ; esperar 200 milisegundos
portb = %00000000        ; apagamos el puerto B0
  PAUSE 200               ; esperar 200 milisegundos
LOW led                   ; apagamos el puerto D7
PAUSE 200                 ; esperar 200 milisegundos
GOTO Practica2           ; Retornar a Practica1
END                       ; Fin de la Programación

```

```

; PROGRAMA PARA LUCES DE AUTO FANTÁSTICO (Forma 2)
ADCON1 = 7 ; transforma en digitales los puertos analógicos
x VAR BYTE ; creamos la variable x y le asignamos tamaño de 255
LEDS VAR PORTB ; todo el puerto B se llamará LEDS
Trisb = 0 ; hacemos salidas a todo el puerto B
LEDS = 1 ; Cargamos el puerto B con 1 (%00000001)
Practica2:
FOR x = 1 TO 7 ; repetir 7 veces
    LEDS = LEDS << 1 ; desplazar uno a uno a la izquierda
PAUSE 200 ; esperar 200 milisegundos
NEXT ; repetir hasta que x sea = a 7
FOR x = 1 TO 7 ; repetir 7 veces
    LEDS = LEDS >> 1 ; desplazar uno a uno a la derecha
PAUSE 200 ; esperar 200 milisegundos
NEXT ; repetir hasta que x sea = a 7
GOTO Practica2 ; ir a Practica2
END ; fin de la programación

```

8. PREGUNTAS DE CONTROL:

8.1. ¿En qué se diferencia un PIC de otro?

La diferencia radica en las prestaciones que ofrecen, al estar constituidos por cantidades distintas de puertos y pines de conexión, que pueden ser configurados como entradas o salidas.

8.2. ¿Cuál es la diferencia entre un PIC y un Microprocesador?

Un microcontrolador es un solo circuito integrado, que contiene todos los elementos electrónicos que se utilizan para hacer funcionar un sistema basado en Microprocesadores.

8.3. ¿En qué consiste un algoritmo de desplazamiento?

Un algoritmo de desplazamiento, es aquel que a través de un proceso repetitivo **FOR**, aumenta el valor de una variable definida como contador. Y para cada conteo el ciclo realiza la acción de desplazar de manera automática datos de izquierda a derecha en un puerto previamente definido.

PRACTICA 3

1. TEMA:

Semáforo de Tres Intersecciones.

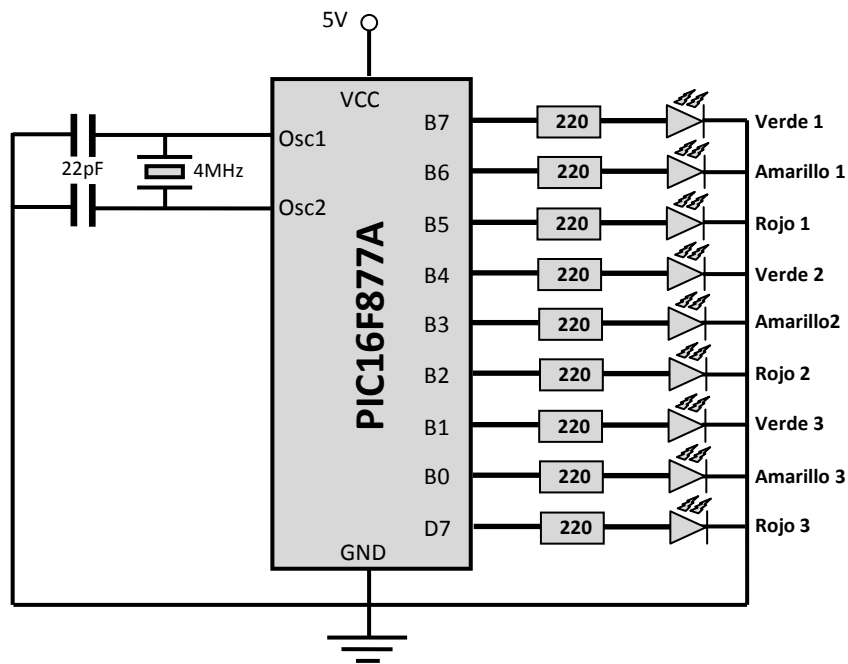
2. OBJETIVOS:

- Diseñar y armar en el software de simulación ISIS de PROTEUS, un circuito que permita realizar la simulación de un semáforo de tres intersecciones.
- Diseñar un algoritmo y un programa en el software de programación MICROCODE, que permita hacer un semáforo de tres intersecciones.
- Analizar el comportamiento del circuito y el algoritmo del programa.

3. MATERIALES:

Cantidad	Designación	Características
9	Leds	3 Rojos, 3Verdes, 3 Amarillos
9	Resistores	220Ω (1/4W)
1	Resistor	4,7kΩ (1/4W)
1	Pulsador	NA
1	Microcontrolador	PIC (16F877A ó 16F628A)
2	Capacitores Cerámicos	22pF ó 33pF
1	Cristal de Cuarzo	4MHz

4. ESQUEMA:



5. SISTEMA CATEGORIAL:

5.1. SEMÁFORO

Los **semáforos** son dispositivos de señales que se sitúan en intersecciones viales, pasos de peatones y otros lugares para regular el tráfico de el tránsito depeatones.



Elección de colores

El motivo por el que se eligieron el color rojo y el verde para la regulación del tráfico es que se heredaron del mundo del ferrocarril que a su vez las

había heredado del marítimo. Desde siglos atrás, los barcos utilizaban un código de colores para señalar el derecho de paso (código de colores que se sigue usando hoy en día y, ahora también, en las alas de los aviones): rojo a babor y verde a estribor. De este modo, si dos barcos se acercan el uno al otro perpendicularmente; uno de ellos ve la luz roja en el babor del otro, que se le acerca por la derecha, y el barco que viene por la derecha ve la luz verde en el estribor del otro barco. El timonel que veía la luz roja sabía que debía ceder el paso al otro barco, y el que veía la luz verde sabía que podía continuar sin problemas.

Componentes

El semáforo está formado por los siguientes componentes:

- **Cabeza:** Es la armadura que contiene las partes visibles del semáforo. Cada cabeza contiene un número determinado de caras orientadas en diferentes direcciones.
- **Soportes:** Los soportes son las estructuras que se utilizan para sujetar la cabeza de los semáforos de forma que les permitan algunos ajustes angulares, verticales y horizontales.
- **Cara:** Son las distintas luces de las cuales están formados los semáforos. En cada cara puede haber desde dos luces hasta más de tres, siendo la de tres luces las caras más usuales.
- **Lente:** Es la parte de la unidad óptica que por refracción dirige la luz proveniente de la lámpara y de su reflector en la dirección deseada. Este elemento desaparece en los nuevos semáforos de LEDs.

- **Visera:** Es un elemento que se coloca encima o alrededor de cada una de las unidades ópticas, para evitar que, a determinadas horas, los rayos del sol incidan sobre éstas y den la impresión de estar iluminadas, así como también para impedir que la señal emitida por el semáforo sea vista desde otros lugares distintos hacia el cual está enfocado. Como el caso de las lentes, esta parte está desapareciendo ya que los nuevos semáforos de LEDs iluminan de mejor forma que los antiguos.
- **Placa de contraste:** Elemento utilizado para incrementar la visibilidad del semáforo y evitar que otras fuentes lumínicas confundan al conductor.

6. PROCEDIMIENTO:

6.1. Para armar el circuito de simulación utilizamos ISIS de Proteus

Profesional y su interfaz.

6.1.1. En el buscador de Proteus seleccionamos el Pic16F877A y lo ubicamos en la pantalla principal.

6.1.2. En el buscador de Proteus seleccionamos el cristal de cuarzo, lo ubicamos en la pantalla principal y lo conectamos al oscilador 1 y 2 del PIC.

6.1.3. En el buscador de Proteus seleccionamos dos capacitores de 22pF ó 33pF, los ubicamos en la pantalla principal y los conectamos al oscilador 1 y 2 del PIC y a GND.

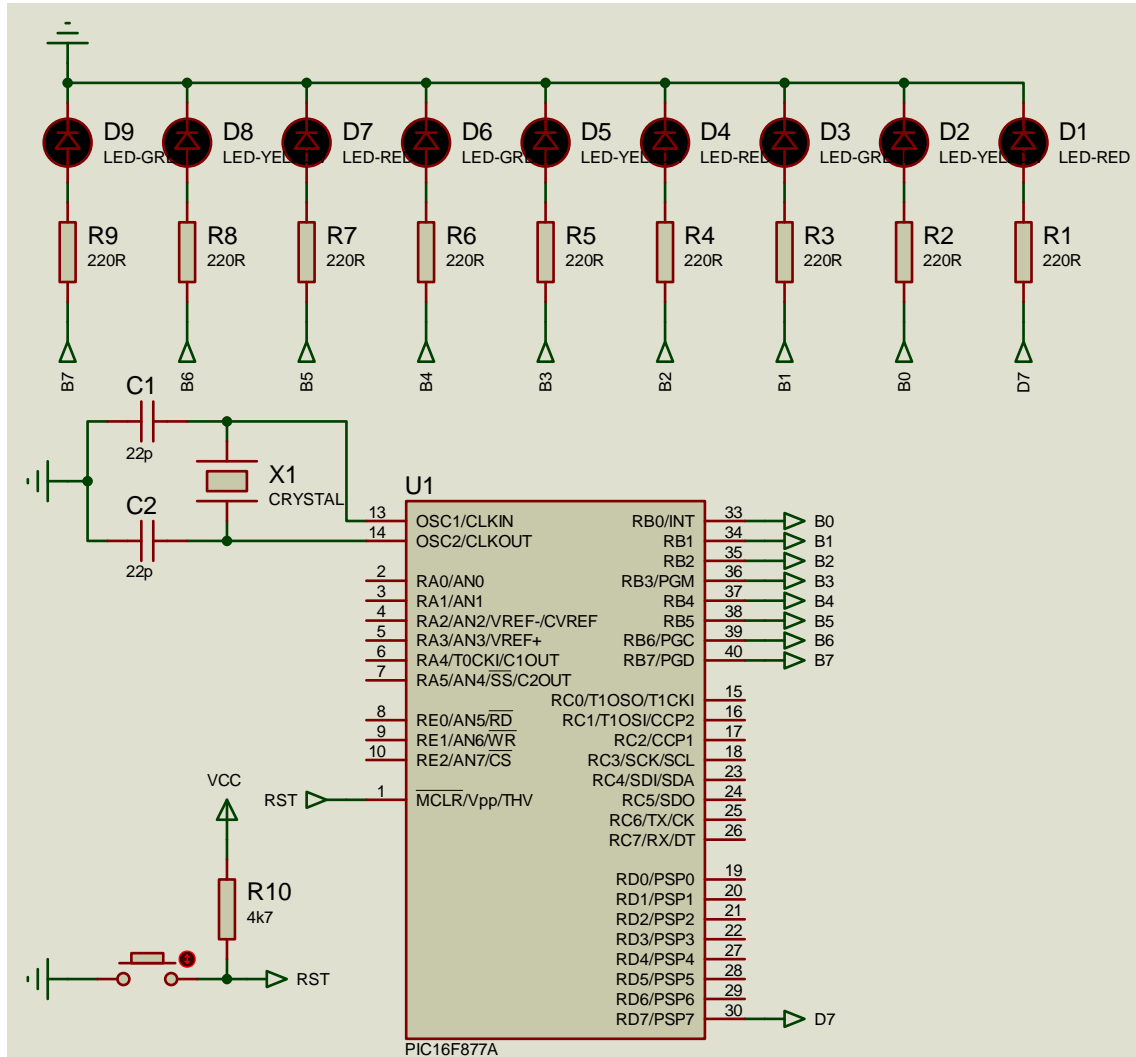
- 6.1.4. En el buscador de Proteus seleccionamos una resistencia de 220Ω , ubicamos en la pantalla principal nueve de las mismas y las conectamos a los pines puerto B y D7.
 - 6.1.5. En el buscador de Proteus seleccionamos un led rojo, amarillo y verde, ubicamos tres de cada uno en la pantalla principal y los conectamos a las resistencias de 220Ω y a GND.
 - 6.1.6. En el buscador de Proteus seleccionamos una resistencia de $4,7k\Omega$, la ubicamos en la pantalla principal y la conectamos al reset del PIC y a Vcc.
 - 6.1.7. En el buscador de Proteus seleccionamos un pulsador, lo ubicamos en la pantalla principal y lo conectamos al reset del PIC y a GND.
- 6.2. Para la programación del PIC utilizamos Microcode Studio.
- 6.2.1. Transformamos digitales todos los puertos analógicos del PIC.
 - 6.2.2. Declaramos el puerto B como salida.
 - 6.2.3. Le damos el nombre **rojo3** al **puerto D.7** del PIC.
 - 6.2.4. Le asignamos un nombre al programa.
 - 6.2.5. Encendemos el Verde1, el Rojo2 y Rojo3
 - 6.2.6. Le damos una pausa al programa de 10000 milisegundos.
 - 6.2.7. Apagamos el Verde1 y Encendemos el Amarillo1.
 - 6.2.8. Le damos una nueva pausa al programa de 3000 milisegundos.
 - 6.2.9. Apagamos el Amarillo1 y Encendemos el Rojo1 y el Verde2
 - 6.2.10. Le damos una nueva pausa al programa de 10000 milisegundos.
 - 6.2.11. Apagamos el Verde2 y Encendemos el Amarillo2.

- 6.2.12. Le damos una nueva pausa al programa de 3000 milisegundos.
 - 6.2.13. Apagamos el Amarillo2 y Encendemos el Rojo2 y el Verde3
 - 6.2.14. Le damos una nueva pausa al programa de 10000 milisegundos.
 - 6.2.15. Apagamos el Verde3 y Encendemos el Amarillo3.
 - 6.2.16. Le damos una nueva pausa al programa de 3000 milisegundos.
 - 6.2.17. Apagamos el Amarillo3.
 - 6.2.18. Repetimos el programa de manera indefinida.
- 6.3. Antes de la ejecución del programa en Proteus es indispensable sincronizar los osciladores del cristal y PIC en 4MHz, seteando dicho valor en propiedades de los dispositivos.
- 6.4. Cargamos el archivo **.HEX** generado por Microcode en el PIC de Proteus y ejecutamos la simulación, y los nueve leds deben parpadear en intervalos de un segundo.
- 6.5. Para armar la práctica en el módulo es necesario comprar los materiales descritos en la lista y adicionalmente traer un pelacables y cable UTP para conexiones.

7. DESARROLLO:

El circuito armado completamente y la simulación realizada en Proteus

Profesional queda de la siguiente forma:



El algoritmo correspondiente y programa correctamente elaborado en Microcode Studio para hacer parpadear los nueve leds armados en la simulación de Proteus nos queda de la siguiente forma:

```

; SEMÁFORO DE TRES INTERSECCIONES
ADCON1 = 7 ; transforma en digitales los puertos analógicos
Trisb = 0 ; declaramos el puerto B como salida
rojo3 VAR portd.7 ; Le asignamos el nombre de rojo3 al puerto D.7
Practica3:
HIGHrojo3: portb = %10000100 ; Encendemos el puerto B0, B5, y D7
PAUSE 10000 ; esperar 10000 milisegundos
portb = %01000100 ; Apagamos el puerto B0 y Encendemos el B1
PAUSE 3000 ; esperar 3000 milisegundos
portb = %00110000 ; Apagamos el puerto B1 y Encendemos el B2 y B3
PAUSE 10000 ; esperar 10000 milisegundos
portb = %00101000 ; Apagamos el puerto B3 y Encendemos el B4
PAUSE 3000 ; esperar 3000 milisegundos
LOWrojo3: portb = %00100110 ; Apagamos el puerto B4, D7, y Encendemos el B5 y B6
PAUSE 10000 ; esperar 10000 milisegundos
portb = %00100101 ; Apagamos el puerto B6 y Encendemos el B7
PAUSE 3000 ; esperar 3000 milisegundos
GOTO Practica3 ; Ir a Practica3
END ; Fin de la programación

```

8. PREGUNTAS DE CONTROL:

8.1. ¿En qué consiste un algoritmo de semaforización?

Un algoritmo de semaforización, es aquel que permite hacer un juego de cambio de luces en determinados instantes de tiempo, con el afán de regular el tránsito en una ciudad.

8.2. ¿En la práctica, en qué se puede emplear el algoritmo?

El algoritmo en la vida real, se utiliza para la fabricación de semáforos a gran escala.

8.3. ¿Qué es un semáforo?

Los semáforos son dispositivos de señales que se sitúan en intersecciones viales, pasos de peatones y otros lugares para regular el tráfico de el tránsito de peatones.

8.4. ¿Se puede implementar el algoritmo con Focos de 110V-AC?

¿Cómo?

El algoritmo en la vida real, se utiliza para activar relés conectados a focos de 110V-AC, que dan la cantidad de lúmenes necesarios para que los conductores, se puedan guiar y transitar de acuerdo a su funcionamiento.

PRACTICA 4

1. TEMA:

Parpadeo Alternado de Leds con Pulsadores

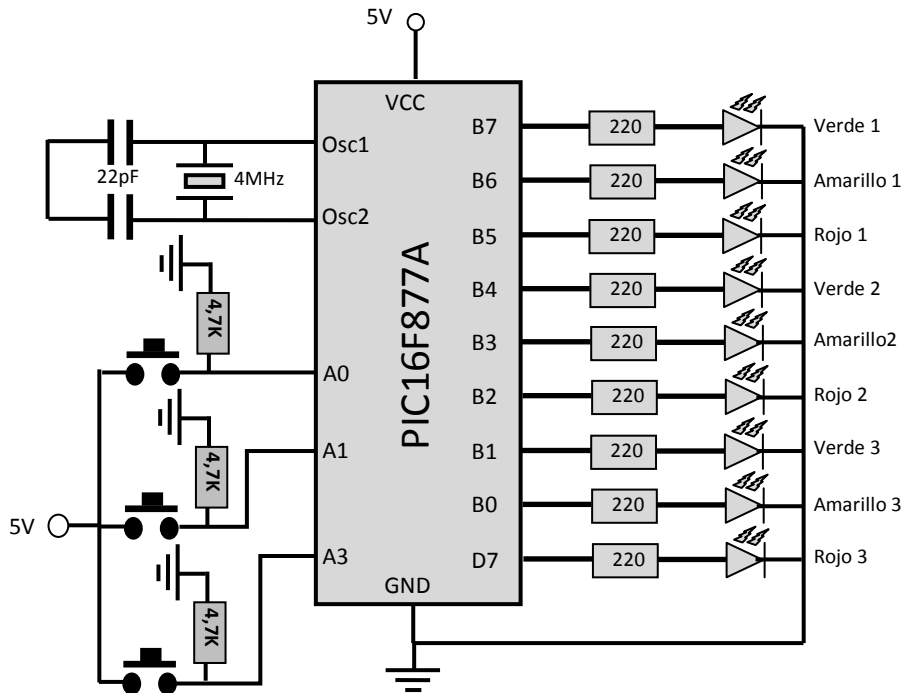
2. OBJETIVOS:

- Diseñar y armar simular en el software de simulación ISIS de PROTEUS, un circuito que permita simular el parpadeo de leds de manera alternada con pulsadores y con tiempos diferentes.
- Diseñar un algoritmo y un programa en el software de programación MICROCODE, que permita hacer parpadear leds de manera alternada con pulsadores y con tiempos diferentes.
- Analizar el comportamiento del circuito y el algoritmo del programa.

3. MATERIALES:

Cantidad	Designación	Características
9	Leds	3 Rojos, 3Verdes, 3 Amarillos
9	Resistores	220 Ω (1/4W)
4	Resistor	4,7k Ω (1/4W)
4	Pulsador	
1	Microcontrolador	PIC (16F877A ó 16F628A)
2	Capacitores Cerámicos	22pF ó 33pF
1	Cristal de Cuarzo	4MHz

4. ESQUEMA:



5. SISTEMA CATEGORIAL:

5.1. Secuencias IF.

La estructura de control IF permite decidir entre dos opciones resultantes de la evaluación de una sentencia. Si la evaluación es positiva hace una cosa, también podemos especificar acciones para realizar en caso de que la evaluación sea negativa. Veamos cómo funciona en VBscript.

```
IF (expresion) then
    Sentencias
    ...
END IF
```

Vemos que en prime lugar tenemos la sentencia IF, luego una expresión, que puede o no ir entre paréntesis, y mas tarde la palabra THEN. Vemos

que luego hay un salto de línea antes de colocar las sentencias asociadas a la evaluación positiva de la sentencia.

Después de poner las sentencias asociadas a la evaluación positiva colocamos un END IF, para acabar la estructura del IF

Enunciado ELSE

Opcionalmente se puede colocar una serie de sentencias asociadas a la evaluación negativa de la expresión. Estas sentencias se deben colocar después del orden ELSE y antes del ENDIF.

```

IF (expresion) then
    Sentencias
    ....
ELSE
    Sentencias
    ....
END IF

```

Enunciado ELSEIF

En visual Basic Script existe la posibilidad de utilizar un enunciado especial en el lugar donde utilizamos un ELSE. Sirve para encadenar sentencias IF de modo que un resultado negativo de un IF se pueda evaluar otra expresión que tendría a su vez otros enunciados THEN y probablemente ELSE u otro ELSEIF se vería en un ejemplo con más facilidad.

```

IF (expresion1) then
    Sentencias1
    ....
ELSEIF (expresion2) then
    Sentencias2
    ....
ELSEIF (expresion3) then
    Sentencias3
    ....
ELSE
    Sentencias4
    ....
END IF

```

Se evalúa la primera expresión, en caso positivo se ejecutan las sentencias1, en caso negativo se evalúa la expresión2. Si la expresión2 es positiva se ejecutan las sentencias2 en caso negativo, la expresión3 con el siguiente ELSEIF. Todo acaba un ELSE en este ejemplo, pero el ELSE final no es obligatorio.

5.2. Pulsadores, modos de configuración.

Un pulsador es simplemente un interruptor o switch cuya función es permitir o interrumpir el paso de la corriente eléctrica, a diferencia de un switch común, un pulsador solo realiza su trabajo mientras lo tengas presionado.



Existen pulsadores **NC** y **NA**, es decir **normalmente abiertos** y **normalmente cerrados**, cuando pulsas uno de tipo NC, se abre mientras lo presiones es decir no permite el paso de la corriente, y en un NA, cuando lo presionas permites el paso, es decir lo contrario, generalmente se usa para enviar pulsos o para activar algo. Como ejemplo el teclado de tu PC cada tecla es un pulsador, cuando presionas uno permite el paso de una señal eléctrica que le dice a la CPU que estas presionando esa tecla.

6. PROCEDIMIENTO:

6.1. Para armar el circuito de simulación utilizamos ISIS de Proteus

Profesional y su interfaz.

6.1.1. En el buscador de Proteus seleccionamos el Pic16F877A y lo ubicamos en la pantalla principal.

6.1.2. En el buscador de Proteus seleccionamos el cristal de cuarzo, lo ubicamos en la pantalla principal y lo conectamos al oscilador 1 y 2 del PIC.

6.1.3. En el buscador de Proteus seleccionamos dos capacitores de 22pF ó 33pF, los ubicamos en la pantalla principal y los conectamos al oscilador 1 y 2 del PIC y a GND.

6.1.4. En el buscador de Proteus seleccionamos una resistencia de 220Ω , ubicamos en la pantalla principal nueve de las mismas y las conectamos a los pines puerto B y D7.

6.1.5. En el buscador de Proteus seleccionamos un led rojo, amarillo y verde, ubicamos tres de cada uno en la pantalla principal y los conectamos a las resistencias de 220Ω y a GND.

6.1.6. En el buscador de Proteus seleccionamos una resistencia de $4,7k\Omega$, la ubicamos en la pantalla principal y la conectamos al reset del PIC y a Vcc.

6.1.7. En el buscador de Proteus seleccionamos un pulsador, lo ubicamos en la pantalla principal y lo conectamos al reset del PIC y a GND.

6.2. Para la programación del PIC utilizamos Microcode Studio.

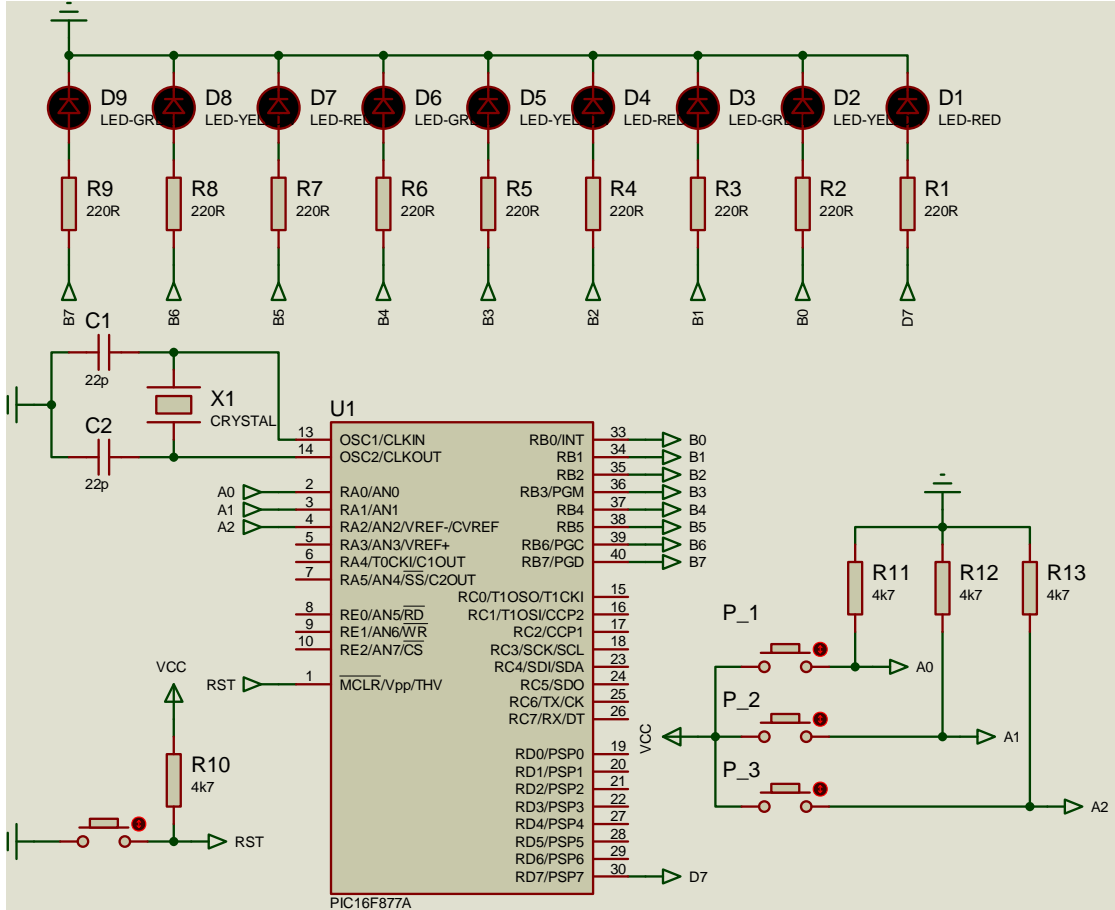
6.2.1. Transformamosa digitales todos los puertos analógicos del PIC.

- 6.2.2. Declaramos el puerto B como salida.
 - 6.2.3. Le damos el nombre *led* al **puerto D.7** del PIC.
 - 6.2.4. Le asignamos un nombre al programa.
 - 6.2.5. Creamos una secuencia **IF**, para que cuando esté en estado **ALTO** el pulsador_1, se enciendan secuencialmente únicamente los 3 leds rojos.
 - 6.2.6. Le damos una pausa al programa de 500 milisegundos.
 - 6.2.7. Creamos una secuencia **IF**, para que cuando esté en estado **ALTO** el pulsador_2, se enciendan secuencialmente únicamente los 3 leds amarillos.
 - 6.2.8. Le damos una pausa al programa de 1500 milisegundos.
 - 6.2.9. Creamos una secuencia **IF**, para que cuando esté en estado **ALTO** el pulsador_1, se enciendan secuencialmente únicamente los 3 leds verdes.
 - 6.2.10. Le damos una pausa al programa de 2500 milisegundos.
 - 6.2.11. Repetimos el programa de manera indefinida.
- 6.3. Antes de la ejecución del programa en Proteus es indispensable sincronizar los osciladores del cristal y PIC en 4MHz, seteando dicho valor en propiedades de los dispositivos.
- 6.4. Cargamos el archivo **.HEX** generado por Microcode en el PIC de Proteus y ejecutamos la simulación, y los nueve leds deben parpadear en intervalos de un segundo.
- 6.5. Para armar la práctica en el módulo es necesario comprar los materiales descritos en la lista y adicionalmente traer un pelacables y cable UTP para conexiones.

7. DESARROLLO:

El circuito armado completamente y la simulación realizada en Proteus

Profesional queda de la siguiente forma:



El algoritmo correspondiente y programa correctamente elaborado en Microcode Studio para el parpadeo alternado de leds con pulsadores armados en la simulación de Proteus nos queda de la siguiente forma:

; PARPADEO ALTERNADO DE LEDS CON PULSADORES

```

ADCON1 = 7 ; transforma en digitales los puertos analógicos
Trisb = 0 ; Declaramos el puerto B como salida
led VAR portd.7 ; Le asignamos el nombre de led al puerto D.7
Practica4:
    portb = %00000000 : LOW led ; apagamos todo el puerto B y D.7
    IF porta.0 = 1 THEN Rojo ; ir a rojo si el puerto E0 es 1
    IF porta.1 = 1 THEN Amarillo ; ir a amarillo si el puerto E0 es 1
    IF porta.2 = 1 THEN Verde ; ir a verde si el puerto E0 es 1
    GOTO Practica4
Rojo:
    HIGH portb.2 ; Encendemos el puerto B2
    PAUSE 500 ; esperar 500 milisegundos
    LOW portb.2 ; Apagamos el puerto B2
    PAUSE 500 ; esperar 500 milisegundos
    HIGH portb.5 ; Encendemos el puerto B5
    PAUSE 500 ; esperar 500 milisegundos
    LOW portb.5 ; Apagamos el puerto B5
    PAUSE 500 ; esperar 500 milisegundos
    HIGH led ; Encendemos el puerto D7
    PAUSE 500 ; esperar 500 milisegundos
    LOW led ; Apagamos el puerto D7
    PAUSE 500 ; esperar 500 milisegundos
    GOTO Practica4
Amarillo:
    HIGH portb.1 ; Encendemos el puerto B1
    PAUSE 1500 ; esperar 1500 milisegundos
    LOW portb.1 ; Apagamos el puerto B1
    PAUSE 1500 ; esperar 1500 milisegundos
    HIGH portb.4 ; Encendemos el puerto B4
    PAUSE 1500 ; esperar 1500 milisegundos
    LOW portb.4 ; Apagamos el puerto B4
    PAUSE 1500 ; esperar 1500 milisegundos
    HIGH portb.7 ; Encendemos el puerto B7
    PAUSE 1500 ; esperar 1500 milisegundos
    LOW portb.7 ; Apagamos el puerto B7
    PAUSE 1500 ; esperar 1500 milisegundos
    GOTO Practica4
Verde:
    HIGH portb.0 ; Encendemos el puerto B0
    PAUSE 2500 ; esperar 2500 milisegundos
    LOW portb.0 ; Apagamos el puerto B0
    PAUSE 2500 ; esperar 2500 milisegundos

```



```

HIGH portb.3           ; Encendemos el puerto B3
    PAUSE 2500         ; esperar 2500 milisegundos
LOW portb.3           ; Apagamos el puerto B3
    PAUSE 2500         ; esperar 2500 milisegundos
HIGH portb.6          ; Encendemos el puerto B6
    PAUSE 2500         ; esperar 2500 milisegundos
LOW portb.6           ; Apagamos el puerto B6
    PAUSE 2500         ; esperar 2500 milisegundos
    GOTO Practica4
END

```

8. PREGUNTAS DE CONTROL:

8.1. ¿Por qué no utilizar un Microprocesador para ésta práctica?

Porque es necesario, una gran cantidad de memoria al momento de guardar los datos para cada caso, y la ALU interna de un PIC permite que los datos se ejecuten con menor cantidad de errores.

8.2. ¿Cuál es la mejor forma de configurar un pulsador?

La mejor forma de configurar un pulsador, es con el uso de un 74LS14 (TRIGGER SMITH), que permite eliminar los rebotes de los pulsos generados, por la variación del voltaje.

8.3. ¿Cuántos casos de análisis se pueden considerar en una secuencia IF?

Las secuencias IF, permiten ejecutar varios casos internos bajo una condición específica.

PRACTICA 5

1. TEMA:

Encender un Foco de 110V-AC con un Pulsador.

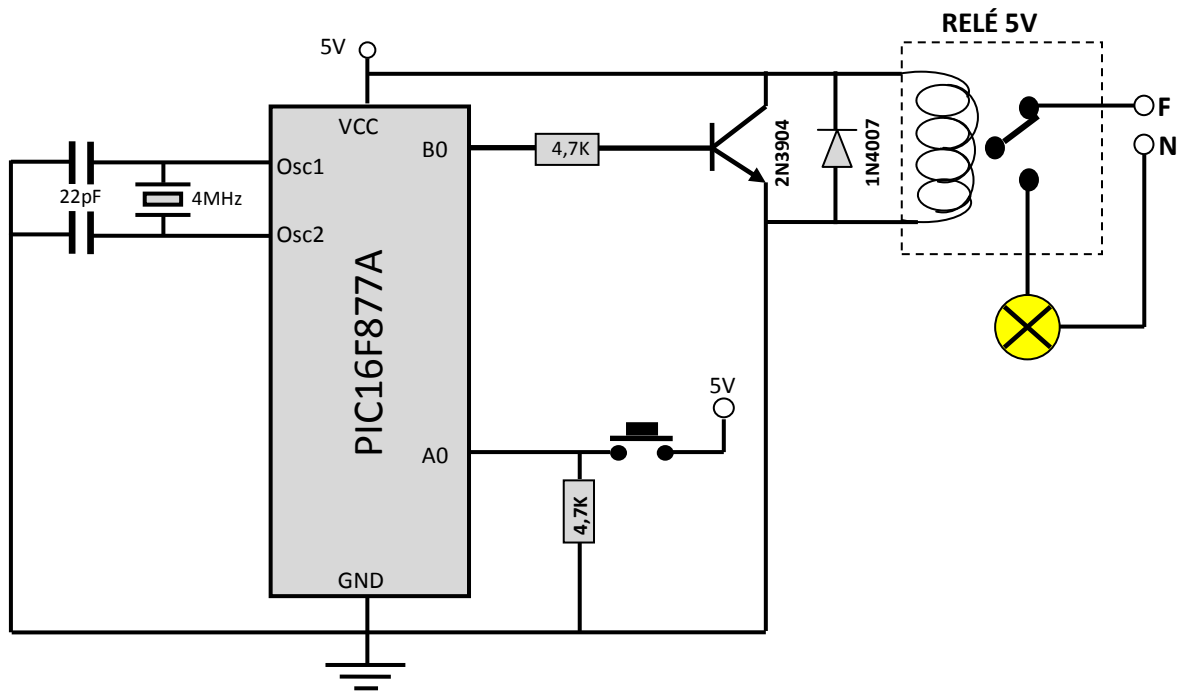
2. OBJETIVOS:

- Diseñar y armar en el software de simulación ISIS de PROTEUS, un circuito que permita simular el encendido de un foco de 110v-ac con un pulsador.
- Diseñar un algoritmo y un programa en el software de programación MICROCODE, que permita encender un foco de 110v-ac con un pulsador.
- Analizar el comportamiento del circuito y el algoritmo del programa.

3. MATERIALES:

Cantidad	Designación	Características
1	Boquilla	110V-AC
1	Foco	110V-AC (30W)
1	Relé	5V-DC
1	Diodo	1N4007
1	Transistor	2N3904
2	Resistores	4,7k Ω (1/4W)
2	Pulsadores	NA
2	Capacitores Cerámicos	22pF ó 33pF
1	Cristal de Cuarzo	4MHz
1	Microcontrolador	PIC (16F877A ó 16F628A)

4. ESQUEMA:



5. SISTEMA CATEGORIAL:

5.1. Tensión límite de seguridad

Según las condiciones del entorno, particular mente en presencia de humedad, la tensión limite de seguridad esta definida como la tensión por debajo de la cual no existe riesgo para las personas.

Para la corriente alterna se tiene que las tensiones limites de seguridad son:

- 50 voltios para locales secos
- 24 voltios para locales humedados
- 12 voltios para áreas mojadas (baños, piscinas, exteriores)
- 60 V en corriente continúa.

TIPOS DE CONTACTOS

La electrocución de una persona puede ocurrir por dos tipos de contacto:

- CONTACTO DIRECTO
- CONTACTO INDIRECTO

Contacto directo

Este tipo de situación ocurre cuando una persona toca directamente partes activas o entra en contacto con elementos energizados, y puede sufrir un choque eléctrico.

Es particularmente peligroso cuando se tiene contacto con tensiones superiores a las tensiones límites de seguridad; es decir se tienen en cuenta las condiciones del sitio en el cual puede ocurrir dicho contacto.

Teniendo en cuenta que la energía eléctrica es de uso generalizado, las personas están en contacto permanente con conductores eléctricos, electrodomésticos, equipos eléctricos, motores eléctricos...

Contacto Indirecto:

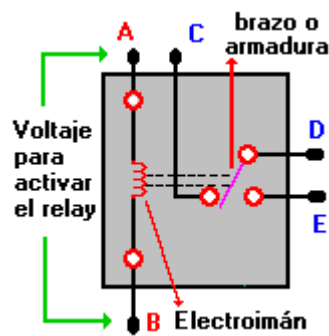
Cuando se produce un contacto con una masa puesta accidental mente en tensión, el umbral de peligro esta determinado por la tensión límite de seguridad.

Para que no exista peligro cuando la tensión de la red sea superior a la tensión límite de seguridad, La tensión de contacto debe ser inferior a la tensión límite de seguridad.

El contacto indirecto sucede cuando la persona toca una estructura metálica, o una carcasa de un motor la cual en condiciones normales esta desenergizadas. Una falla común en un sistema eléctrico es la pérdida de aislamiento provocando fugas de corriente.

5.2. Funcionamiento del Relé:

Si el electroimán está activo jala el brazo (armadura) y conecta los puntos C y D. Si el electroimán se desactiva, conecta los puntos D y E. De esta manera se puede conectar algo, cuando el electroimán está activo, y otra cosa conectada, cuando está inactivo. Es importante saber cual es la resistencia del bobinado del electroimán (lo que está entre los terminales A y B) que activa el **relé** y con cuanto voltaje este se activa. Este voltaje y esta resistencia nos informan que magnitud debe de tener la señal que activará el **relé** y cuanta corriente se debe suministrar a éste. La corriente se obtiene con ayuda de la Ley de Ohm: $I = V / R$.



6. PROCEDIMIENTO:

6.1. Para armar el circuito de simulación utilizamos ISIS de Proteus

Profesional y su interfaz.

6.1.1. En el buscador de Proteus seleccionamos el Pic16F877A y lo ubicamos en la pantalla principal.

6.1.2. En el buscador de Proteus seleccionamos el cristal de cuarzo, lo ubicamos en la pantalla principal y lo conectamos al oscilador 1 y 2 del PIC.

6.1.3. En el buscador de Proteus seleccionamos dos capacitores de 22pF ó 33pF, los ubicamos en la pantalla principal y los conectamos al oscilador 1 y 2 del PIC y a GND.

6.1.4. En el buscador de Proteus seleccionamos un pulsador, ubicamos dos en la pantalla principal, uno lo conectamos al reset del PIC y a GND; y al otro lo conectamos al puerto A0 y a VCC.

6.1.5. En el buscador de Proteus seleccionamos un transistor 2N3904, lo ubicamos en la pantalla principal y la *BASE* la conectamos a una resistencia de 4,7KΩ, el *EMISOR* lo conectamos a VCC y el *COLECTOR* a GND.

6.1.6. En el buscador de Proteus seleccionamos una resistencia de 4,7kΩ, ubicamos tres de las mismas en la pantalla principal, una la conectamos al reset del PIC y a VCC, la segunda al pulsador y a GND, la tercera al puerto B0.

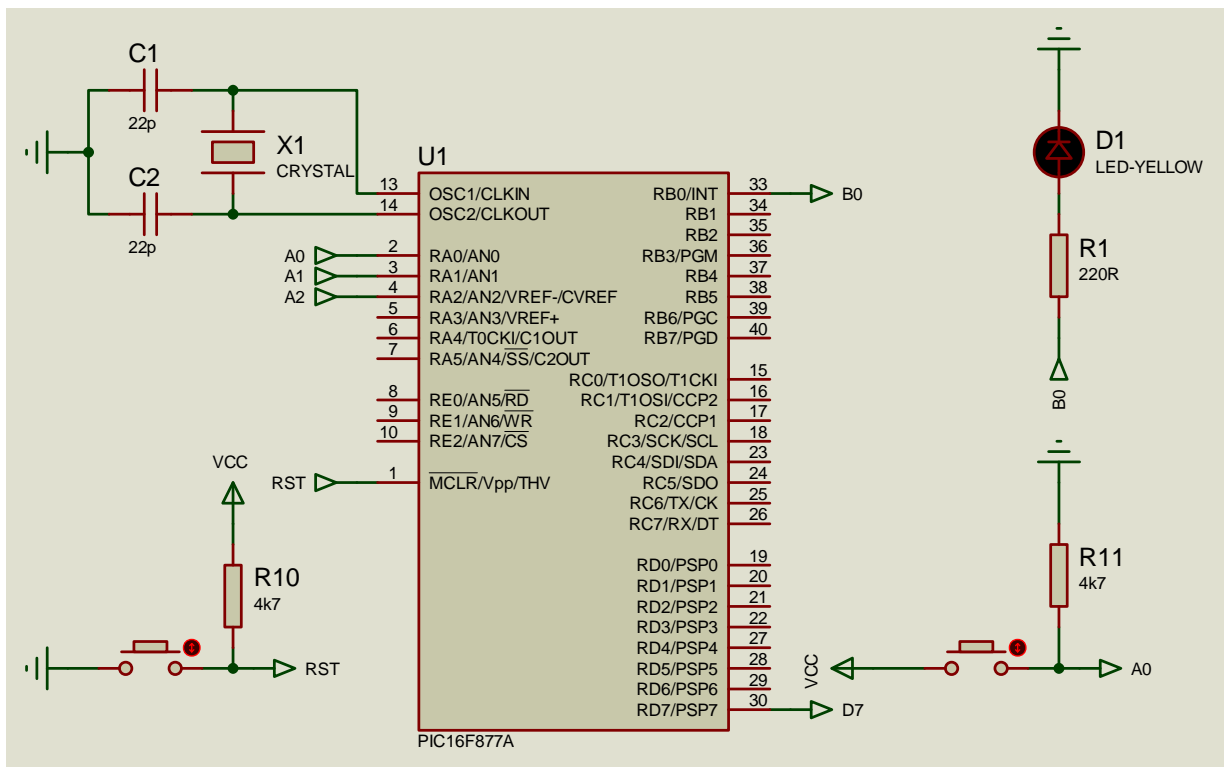
6.1.7. En el buscador de Proteus seleccionamos un diodo 1N4007, lo ubicamos en la pantalla principal y lo conectamos entre el *EMISOR* y *COLECTOR* del transistor.

- 6.1.8. En el buscador de Proteus seleccionamos un relé de 5V, lo ubicamos en la pantalla principal lo conectamos entre el *EMISOR* y *COLECTOR* del transistor la parte de la bobina y el switch a 110V-AC.
- 6.1.9. En el buscador de Proteus seleccionamos una lámpara de 110V-AC, la ubicamos en la pantalla principal y la conectamos al switch del relé y a GND.
- 6.2. Para la programación del PIC utilizamos Microcode Studio.
- 6.2.1. Transformamosa digitales todos los puertos analógicos del PIC.
- 6.2.2. Declaramos el puerto B como salida.
- 6.2.3. Le asignamos un nombre al programa.
- 6.2.4. Creamos una secuencia **IF**, para que cuando esté en estado **ALTO**el pulsador_1 conectado al puerto A0, se encienda la lámpara de 110V-AC conectada con el relé al puerto B0.
- 6.2.5. Le damos una pausa al programa de 1000 milisegundos.
- 6.2.6. **Caso contrario** apagamos la lámpara de 110V-AC conectada con el relé al puerto B0.
- 6.2.7. Le damos una nueva pausa al programa de 1000 milisegundos.
- 6.2.8. Repetimos el programa de manera indefinida.
- 6.3. Antes de la ejecución del programa en Proteus es indispensable sincronizar los osciladores del cristal y PIC en 4MHz, seteando dicho valor en propiedades de los dispositivos.
- 6.4. Cargamos el archivo **.HEX** generado por Microcode en el PIC de Proteus y ejecutamos la simulación, y los nueve leds deben parpadear en intervalos de un segundo.

6.5. Para armar la práctica en el módulo es necesario comprar los materiales descritos en la lista y adicionalmente traer un pelacables y cable UTP para conexiones.

7. DESARROLLO:

El circuito armado completamente y la simulación realizada en Proteus Profesional queda de la siguiente forma:



El algoritmo correspondiente y programa correctamente elaborado en Microcode Studio para encender un foco de 110v-AC con un pulsador. Armados en la simulación de Proteus nos queda de la siguiente forma:


```

; PARPADEO DE UN FOCO DE 110V CON UN PULSADOR
ADCON1 = 7 ; transforma en digitales los puertos analógicos
Trisb = 0 ; Declaramos el puerto B como salida
led VAR portd.7 ; Le asignamos el nombre de led al puerto D.7
Practica5:
portb = %00000000 : LOWled ; apagamos todo el puerto B y D.7
  IF porta.0 = 1 THEN ; ir a rojo si el puerto A0 es 1
    HIGH portb.0 ; Encendemos el puerto B0
    PAUSE 500 ; esperar 500 milisegundos
  ELSE
    LOW portb.0 ; Apagamos el puerto B0
    PAUSE 500 ; esperar 500 milisegundos
  ENDIF
GOTO Practica5
END

```

8. PREGUNTAS DE CONTROL:

8.1. ¿Qué consideraciones de debe tener al momento de manipular voltajes AC?

La fase debe ser manipulada con sumo cuidado, debido a que ella es la que contiene el voltaje que puede generar una fuerte corriente al momento de hacer tierra.

8.2. ¿Cómo se conecta un foco de 110V-AC?

La fase es la que se debe controlar, para evitar cualquier percance, mientras que el neutro es el que va directamente a la boquilla del foco.

8.3. ¿Cómo funciona un relé?

Es un switcheador magnético, que se activa cuando en los bornes de la bobina se genera la cantidad de corriente y voltajes establecidos.

8.4. ¿Qué tipo de voltajes se puede conectar a los pines de conmutación del relé?

El voltaje es independiente y puede ser AC ó DC, según las condiciones lo requieran.

8.5. ¿Qué niveles de voltaje se pueden conectar a los pines de conmutación del relé?

Los niveles pueden ser 240V-AC y 28V-DC, pero estos valores pueden fluctuar dependiendo del fabricante.

8.6. ¿Qué niveles de voltaje maneja un PIC?

Los niveles son entre 0V-DC y 5V-DC, que permiten generar un 1 lógico o un 0 lógico según se requiera.

8.7. ¿Se puede conectar 110V-AC a un PIC? ¿Por qué?

No, porque el PIC no está diseñado para soportar voltaje alterno y menos en esos niveles.

8.8. ¿Cómo se puede conectar 110V-AC a un PIC?

A través del uso de un relé, que recibirá voltaje DC y permitirá el paso del voltaje AC.

PRACTICA 6

1. TEMA:

Decodificación, Arreglo y Manejo de Displays.

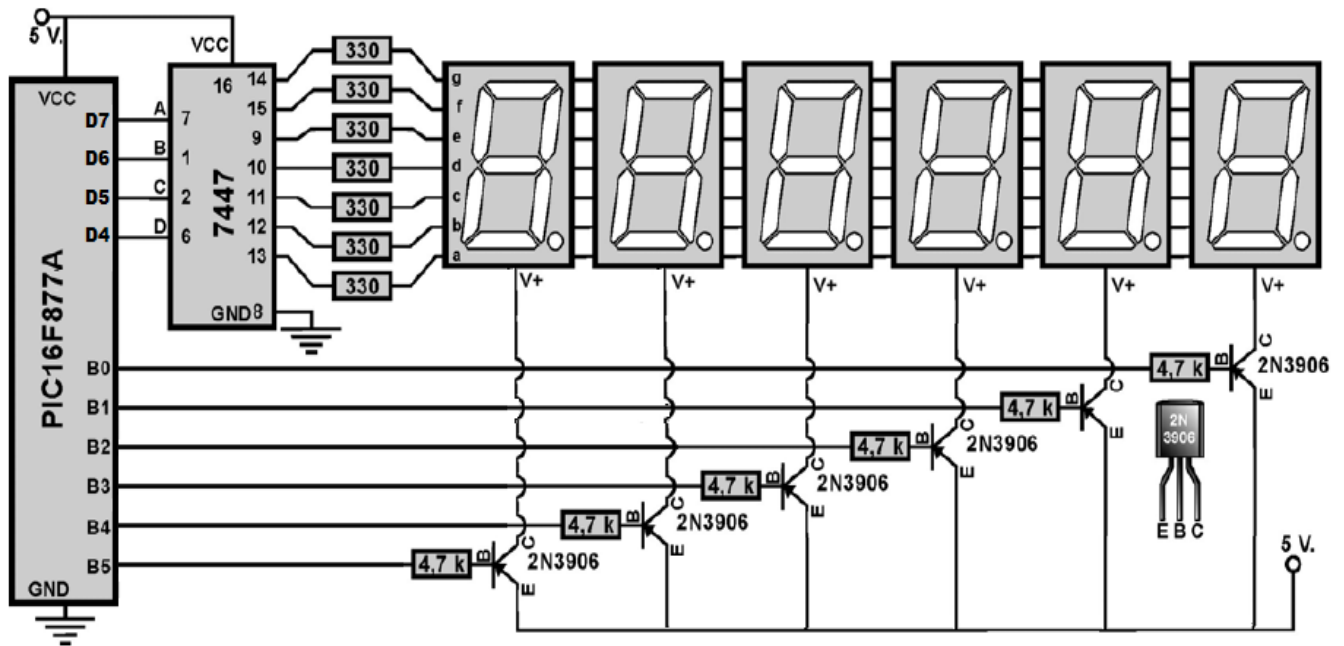
2. OBJETIVOS:

- Diseñar y armar en el software de simulación ISIS de PROTEUS, un circuito que permita simular la decodificación en un arreglo de Displays la frase HOLA PEPE.
- Diseñar un algoritmo y un programa en el software de programación MICROCODE, que permita decodificar en un arreglo de Displays la frase HOLA PEPE.
- Analizar el comportamiento del circuito y el algoritmo del programa.

3. MATERIALES:

Cantidad	Designación	Características
6	Displays	Ánodo común
6	Transistores	2N3906
7	Resistores	4.7KΩ
7	Resistores	220Ω
1	Decodificador	74LS47
1	Pulsadores	NA
2	Capacitores Cerámicos	22pF ó 33pF
1	Cristal de Cuarzo	4MHz
1	Microcontrolador	PIC (16F877A ó 16F628A)

4. ESQUEMA:



5. SISTEMA CATEGORIAL:

5.1. TRANSISTORES

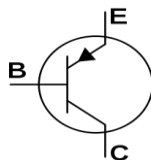
El **transistor de unión bipolar** (del inglés *Bipolar Junction Transistor*, o sus siglas *BJT*) es un dispositivo electrónico de estado sólido consistente en dos uniones PN muy cercanas entre sí, que permite controlar el paso de la corriente a través de sus terminales. La denominación de bipolar se debe a que la conducción tiene lugar gracias al desplazamiento de portadores de dos polaridades (huecos positivos y electrones negativos), y son de gran utilidad en gran número de aplicaciones; pero tienen ciertos inconvenientes, entre ellos su impedancia de entrada bastante baja.

Un transistor de unión bipolar está formado por dos Uniones PN en un solo cristal semiconductor, separados por una región muy estrecha. De esta manera quedan formadas tres regiones:

- **Emisor**, que se diferencia de las otras dos por estar fuertemente dopada, comportándose como un metal. Su nombre se debe a que esta terminal funciona como *emisor* de portadores de carga.
- **Base**, la intermedia, muy estrecha, que separa el emisor del colector.
- **Colector**, de extensión mucho mayor.

5.2. TRANSISTOR PNP

El transistor PNP con las letras "P" y "N" refiriéndose a las cargas mayoritarias dentro de las diferentes regiones del transistor. Pocos transistores usados hoy en día son PNP, debido a que el NPN brinda mucho mejor desempeño en la mayoría de las circunstancias.

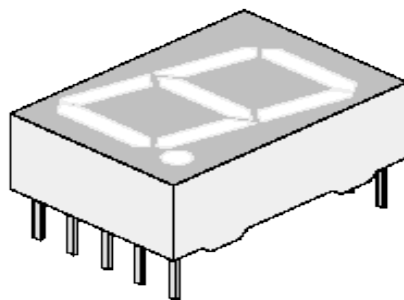


Los transistores PNP consisten en una capa de material semiconductor dopado N entre dos capas de material dopado P. Los transistores PNP son comúnmente operados con el colector a masa y el emisor conectado al terminal positivo de la fuente de alimentación a través de una carga eléctrica externa. Una pequeña corriente circulando desde la base

permite que una corriente mucho mayor circule desde el emisor hacia el colector.

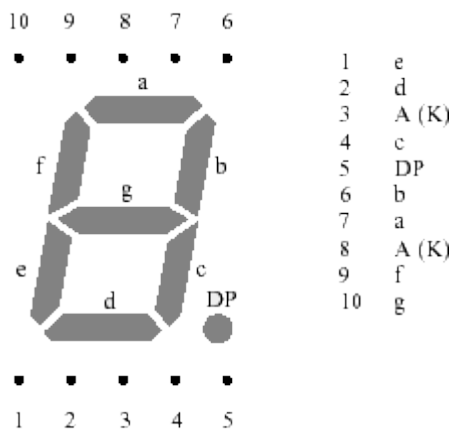
La flecha en el transistor PNP está en el terminal del emisor y apunta en la dirección en la que la corriente convencional circula cuando el dispositivo está en funcionamiento activo.

5.3. Displays de 7 Segmentos



Muchos equipos electrónicos proporcionan información al usuario mediante la utilización de señales luminosas, como la emisora sintonizada en un equipo de radio o la lectura de tensión en un voltímetro digital.

Para representar las cifras numéricas se agrupan siete diodos en de segmentos. Estos diodos tienen conectados entre si todos los ánodos.



Un Display de este tipo está compuesto por siete u ocho leds de diferentes formas especiales y dispuestas sobre una base de manera que puedan representarse todos los símbolos numéricos y algunas letras. Los primeros siete segmentos son los encargados de formar el símbolo y con el octavo podemos encender y apagar el punto decimal.

6. PROCEDIMIENTO:

6.1. Para armar el circuito de simulación utilizamos ISIS de Proteus

Profesional y su interfaz.

6.1.1. En el buscador de Proteus seleccionamos el Pic16F877A y lo ubicamos en la pantalla principal.

6.1.2. En el buscador de ISIS de Proteus seleccionamos el cristal de cuarzo, lo ubicamos en la pantalla principal y lo conectamos al oscilador 1 y 2 del PIC.

6.1.3. En el buscador de ISIS de Proteus seleccionamos dos capacitores de 22pF ó 33pF, los ubicamos en la pantalla principal y los conectamos al oscilador 1 y 2 del PIC y a GND.

6.1.4. En el buscador de Proteus seleccionamos un pulsador, ubicamos dos en la pantalla principal, uno lo conectamos al reset del PIC y a GND; y al otro lo conectamos al puerto A0 y a VCC.

6.1.5. En el buscador de ISIS de Proteus seleccionamos una resistencia de 4,7kΩ, ubicamos siete de las mismas en la pantalla principal, una la conectamos al reset del PIC y a VCC, las demás a la *BASE* de los transistores y al puerto B, desde el B0 al B5.

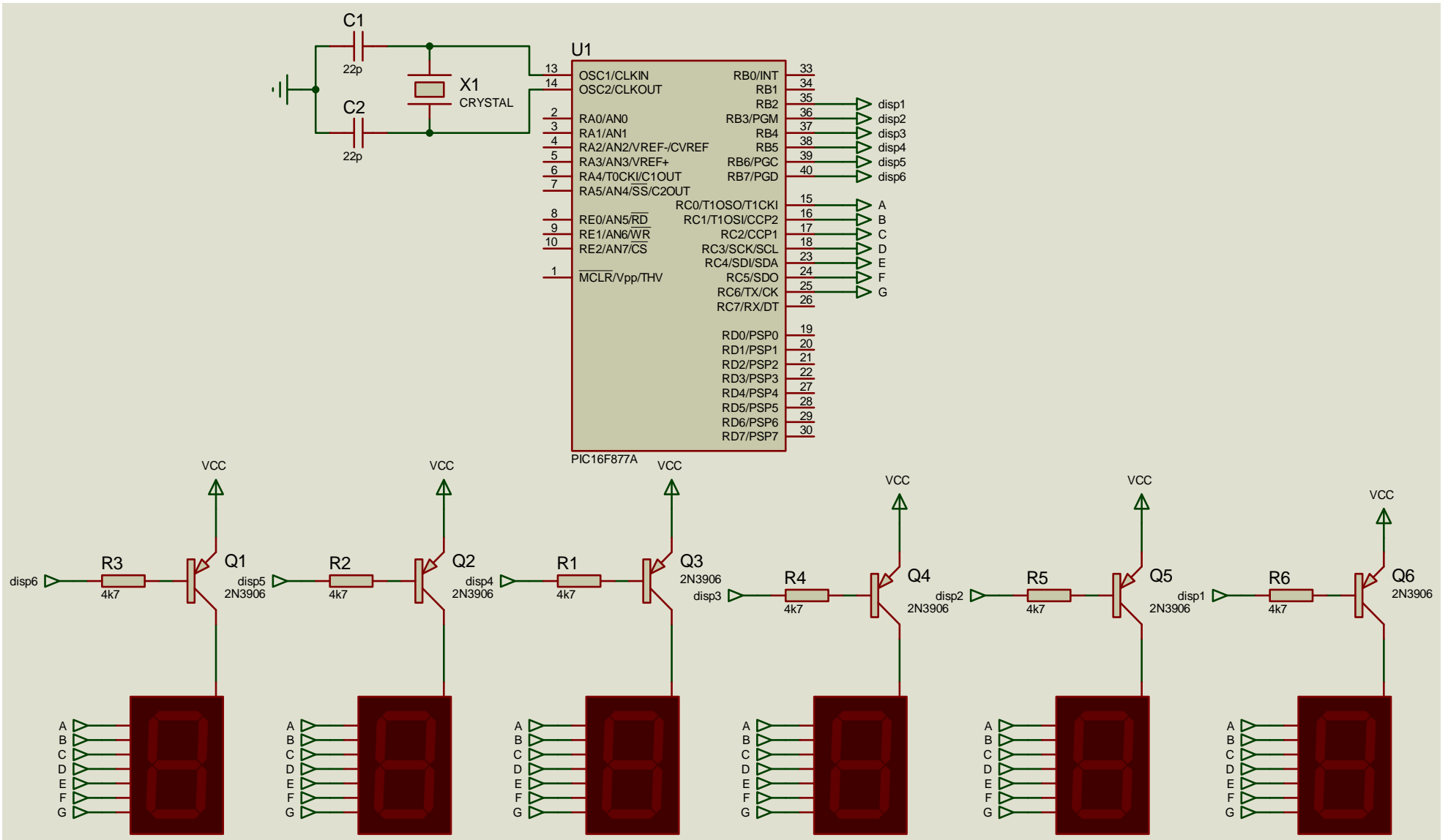
- 6.1.6. En el buscador de ISIS de Proteus seleccionamos un transistor 2N3904, y ubicamos seis en la pantalla principal y la *BASE* la conectamos a las seis resistencias de 4,7K Ω , el *EMISOR* lo conectamos a VCC y el *COLECTOR* a VCC de los seis Displays.
- 6.1.7. En el buscador de ISIS de Proteus seleccionamos una resistencia de 220 Ω , ubicamos siete de las mismas en la pantalla principal y las conectamos a los siete segmentos (A, B, C, D, E, F, G) del 74LS47 y de los Displays.
- 6.1.8. En ISIS de Proteus conectamos los puertos (D7, D6, D5, D4) del PIC a los pines (A0, A1, A2, A3).
- 6.2. Para la programación del PIC utilizamos Microcode Studio.
 - 6.2.1. Transformamosa digitales todos los puertos analógicos del PIC.
 - 6.2.2. Declaramos los puertos B y D como salida.
 - 6.2.3. Le asignamos un nombre al programa.
 - 6.2.4. Creamos una secuencia **FOR**, en la que haremos el barrido de los Displays para que se enciendan y en los mismos aparezcan las letras deseadas.
 - 6.2.5. Activamos el DISPLAY0 y en éste decodificaremos la letra H.
 - 6.2.6. Le damos una pausa al programa de 200 milisegundos.
 - 6.2.7. Activamos el DISPLAY1 y en éste decodificaremos la letra O.
 - 6.2.8. Le damos una pausa al programa de 200 milisegundos.
 - 6.2.9. Activamos el DISPLAY2 y en éste decodificaremos la letra L.
 - 6.2.10. Le damos una pausa al programa de 200 milisegundos.
 - 6.2.11. Activamos el DISPLAY3 y en éste decodificaremos la letra A.

- 6.2.12. Le damos una pausa al programa de 200 milisegundos.
 - 6.2.13. Activamos el DISPLAY4 y en éste decodificaremos la letra
P.
 - 6.2.14. Le damos una pausa al programa de 200 milisegundos.
 - 6.2.15. Activamos el DISPLAY5 y en éste decodificaremos la letra
P.
 - 6.2.16. Le damos una pausa al programa de 200 milisegundos.
 - 6.2.17. Repetimos el programa de manera indefinida.
- 6.3. Antes de la ejecución del programa en Proteus es indispensable sincronizar los osciladores del cristal y PIC en 4MHz, seteando dicho valor en propiedades de los dispositivos.
- 6.4. Cargamos el archivo **.HEX** generado por Microcode en el PIC de Proteus y ejecutamos la simulación, y los nueve leds deben parpadear en intervalos de un segundo.
- 6.5. Para armar la práctica en el módulo es necesario comprar los materiales descritos en la lista y adicionalmente traer un pelacables y cable UTP para conexiones.

7. DESARROLLO:

El circuito armado completamente y la simulación realizada en Proteus

Profesional queda de la siguiente forma:



El algoritmo correspondiente y programa correctamente elaborado en Microcode Studio para la decodificación, arreglo y manejo de Displays armados en la simulación de Proteus nos queda de la siguiente forma:

```

|                                     ; DECODIFICACION DE LA PALABRA HOLAPP
| cmcon=7                             ; convierte todo el puerto A en digital
| trisb=0                              ; convierte todos los pines del puerto B en salidas
| trisc=0                              ; convierte todos los pines del puerto A en salidas
| xVAR BYTE                            ; crea la variable x con un tamaño de 255
| Practica6:
| FOR x=1 TO 20                        ; repeticiones de este segmento
| portc=%00001001 :portb=%01111111    ; pC%1110 y pB%01111111 forma la letra H
| PAUSE 500
| portc=%01000000 :portb=%10111111    ; pC%1101 y pB%10111111 forma la letra O
| PAUSE 500
| portc=%01000111 :portb=%11011111    ; pC%1011 y pB%11011111 forma la letra L
| PAUSE 500
| portc=%00001000 :portb=%11101111    ; pC%0111 y pB%11101111 forma la letra A
| PAUSE 500
| portc=%10001100 :portb=%11110111    ; pC%0111 y pB%11110111 forma la letra P
| PAUSE 500
| portc=%10001100 :portb=%11111011    ; pC%0111 y pB%11111011 forma la letra P
| PAUSE 500
| NEXT
| GOTO Practica6
| END

```

8. PREGUNTAS DE CONTROL:

8.1. ¿Cómo funciona el decodificador?

Recibe un conjunto de bits y los transforma a otro conjunto de mayor o menor tamaño según convenga.

8.2. ¿Qué tipo de código decodifica el 74LS47?

Recibe un código BCD de 4 bits y lo transforma a uno de 7 bits activos a bajo que permiten hacer funcionar los 7 segmentos de un Display.

8.3. ¿Cómo funciona un Display de 7 segmentos?

Depende de su fabricante, puede ser Ánodo Común en cuyo caso se alimenta con Vcc y los segmentos se encienden cuando reciben 0V-DC.

Si el Display es Cátodo Común se alimenta con GND y se encienden los segmentos cuando reciben voltajes entre 2.4V-DC y 5V-DC.

8.4. ¿Cómo se hace un barrido de Displays?

Con la utilización de un algoritmo de desplazamiento, y el uso de las secuencias FOR.

PRACTICA 7

1. TEMA:

Manejo de Elementos Externos de Audio (Parlante de 8Ω).

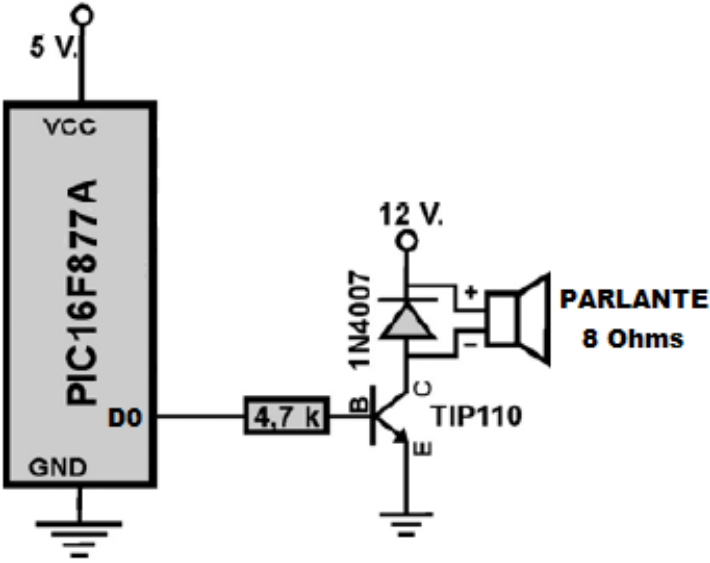
2. OBJETIVOS:

- Diseñar y armar en el software de simulación ISIS de PROTEUS, un circuito que permita simular un tono para reproducirlo a través de un parlante de 8Ω.
- Diseñar un algoritmo y un programa en el software de programación MICROCODE, que permita crear un tono y reproducirlo a través de un parlante de 8Ω.
- Analizar el comportamiento del circuito y el algoritmo del programa.

3. MATERIALES:

Cantidad	Designación	Características
1	Transistor	2N3904
2	Resistores	4.7kΩ
1	Parlante de audio	8Ω
1	Pulsador	NA
2	Capacitores Cerámicos	22pF ó 33pF
1	Cristal de Cuarzo	4MHz
1	Microcontrolador	PIC (16F877A ó 16F628A)

4. ESQUEMA:



5. SISTEMA CATEGORIAL:

5.1. PARLANTE

Un **altavoz** (también conocido como **parlante** es un transductor electroacústico utilizado para la reproducción de sonido. Uno o varios altavoces pueden formar una pantalla acústica.



La transducción sigue un doble procedimiento: eléctrico-mecánico-acústico. En la primera etapa convierte las ondas eléctricas en energía mecánica, y en la segunda convierte la energía mecánica en ondas de frecuencia acústica. Es por tanto la puerta por donde sale el sonido al

exterior desde los aparatos que posibilitaron su amplificación, su transmisión por medios telefónicos o radioeléctricos, o su tratamiento.

El sonido se transmite mediante ondas sonoras, en este caso, a través del aire. El oído capta estas ondas y las transforma en impulsos nerviosos que llegan al cerebro. Si se dispone de una grabación de voz, de música en soporte magnético o digital, o si se recibe estas señales por radio, se dispondrá a la salida del aparato de señales eléctricas que deben ser convertidas en sonidos; para ello se utiliza el altavoz

Las principales características de un altavoz son:

- Respuesta en frecuencia.
- Impedancia.
- Potencia admitida.
- Sensibilidad.
- Rendimiento.
- Distorsión.
- Directividad.

Potencia

Hace referencia a la potencia eléctrica que admite el altavoz (no a la potencia acústica). Es la cantidad de energía (en vatios) que se puede introducir en el altavoz antes de que éste distorsione en exceso o de que pueda sufrir daños. Dentro de la potencia se diferencia entre potencia nominal y potencia admisible.

Impedancia

La impedancia, conceptualmente, es la oposición que presenta cualquier elemento o dispositivo al paso de una corriente alterna (sinusoidal), en este caso la fuente de audio es una mezcla de varias frecuencias con lo cual la impedancia no tendrá el mismo valor en todo el rango de frecuencias. La impedancia se expresa en Ohmios.

Como en los altavoces la impedancia varía en función de la frecuencia, cada modelo de altavoz en sus especificaciones técnicas tendrá una curva con esta relación impedancia-frecuencia distinta. La impedancia de los altavoces viene especificada para una frecuencia concreta que sirva de referencia, generalmente 1 KHz, a menos que el fabricante indique otro valor.

6. PROCEDIMIENTO:

6.1. Para armar el circuito de simulación utilizamos Proteus Profesional y su interfaz.

6.1.1. En el buscador de Proteus seleccionamos el Pic16F877A y lo ubicamos en la pantalla principal.

6.1.2. En el buscador de Proteus seleccionamos el cristal de cuarzo, lo ubicamos en la pantalla principal y lo conectamos al oscilador 1 y 2 del PIC.

6.1.3. En el buscador de Proteus seleccionamos dos capacitores de 22pF ó 33pF, los ubicamos en la pantalla principal y los conectamos al oscilador 1 y 2 del PIC y a GND.

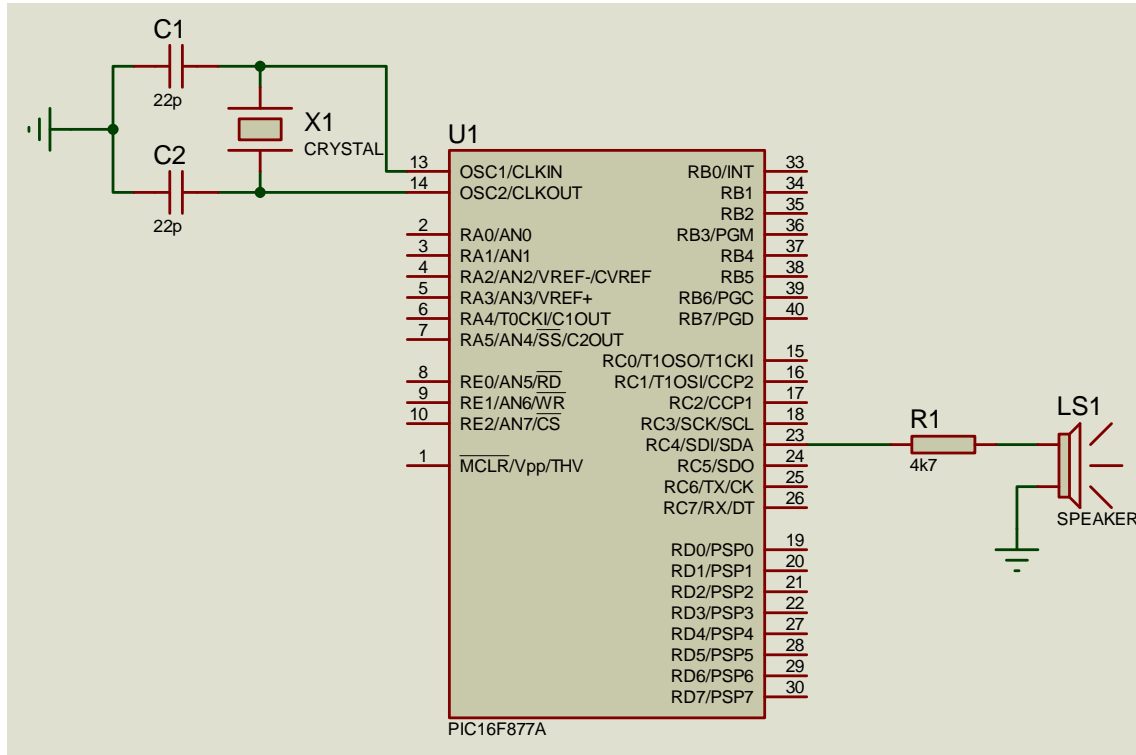
- 6.1.4. En el buscador de Proteus seleccionamos un pulsador, lo ubicamos en la pantalla principal y lo conectamos al reset del PIC y a GND.
 - 6.1.5. En el buscador de Proteus seleccionamos un diodo 1N4007, lo ubicamos en la pantalla principal y conectamos *CÁTODO* a VCC.
 - 6.1.6. En el buscador de Proteus seleccionamos un parlante de 8Ω, lo ubicamos en la pantalla principal y conectamos el positivo al *CÁTODO* y el negativo al *ÁNODO* del diodo.
 - 6.1.7. En el buscador de Proteus seleccionamos una resistencia de 4,7kΩ, ubicamos dos de las mismas en la pantalla principal, una la conectamos al reset del PIC y a VCC, la segunda al puerto D0 del PIC.
 - 6.1.8. En el buscador de Proteus seleccionamos un transistor TIP-110, lo ubicamos en la pantalla principal y la *BASE* la conectamos a la segunda resistencia de 4,7KΩ, el *EMISOR* lo conectamos a GND y el *COLECTOR* al *ÁNODO* del diodo.
- 6.2. Para la programación del PIC utilizamos Microcode Studio.
- 6.2.1. Transformamos digitales todos los puertos analógicos del PIC.
 - 6.2.2. Declaramos el puerto D como salida.
 - 6.2.3. Le asignamos un nombre al programa.
 - 6.2.4. Creamos una secuencia **FOR**, para hacer un proceso repetitivo definido para crear un sonido.
 - 6.2.5. En la secuencia **FOR** utilizamos el comando SOUND para crear un sonido por el puerto D0.

- 6.2.6. Fuera de la secuencia **FOR** damos una pausa al programa de 2500 milisegundos antes de volver a reproducir el tono.
- 6.2.7. Repetimos el programa de manera indefinida.
- 6.3. Antes de la ejecución del programa en Proteus es indispensable sincronizar los osciladores del cristal y PIC en 20MHz, seteando dicho valor en propiedades de los dispositivos.
- 6.4. Cargamos el archivo **.HEX** generado por Microcode en el PIC de Proteus y ejecutamos la simulación, y los nueve leds deben parpadear en intervalos de un segundo.
- 6.5. Para armar la práctica en el módulo es necesario comprar los materiales descritos en la lista y adicionalmente traer un pelacables y cable UTP para conexiones.

7. DESARROLLO:

El circuito armado completamente y la simulación realizada en Proteus

Profesional queda de la siguiente forma:



El algoritmo correspondiente y programa correctamente elaborado en Microcode Studio para el manejo de elementos externos de audio (parlante de 8Ω.)armados en la simulación de Proteus nos queda de la siguiente forma:

```
      ; MANEJO DE ELEMENTOS EXTERNOS DE AUDIO (Parlante de 8 OHMS).
xVAR BYTE          ; variable x de 255
Practica7:
FOR X= 1 TO 15    ; repetir de 1 a 15 veces
SOUND PortC.4,[125,4,123,5] ; sacar tonos por el puerto D.0
NEXT             ; siguiente repetición
PAUSE 2500      ; esperar 2,5 segundos antes de volver a timbrar
GOTO Practica7
END
```

8. PREGUNTAS DE CONTROL:

8.1. ¿Cómo se genera un sonido en Microcode?

Con la manipulación de todo el rango de frecuencias que se pueden generar a partir de la división de la frecuencia principal de funcionamiento del circuito que es de 4 MHz.

8.2. ¿Qué tipo de transistor es el TIP-110?

El TIP-110 es un transistor de potencia, que permite amplificar una señal con excelentes niveles de ganancia.

8.3. ¿Por qué no se utiliza un parlante de un ohmiaje distinto al de 8Ω ?

Porque la impedancia de acople de todo circuito es 8Ω , y si este valor cambia se debe reconsiderar todo el diseño del circuito, con la impedancia deseada para la salida.

PRACTICA 8

1. TEMA:

DECODIFICACIÓN Y MANEJO DE UNA LCD.

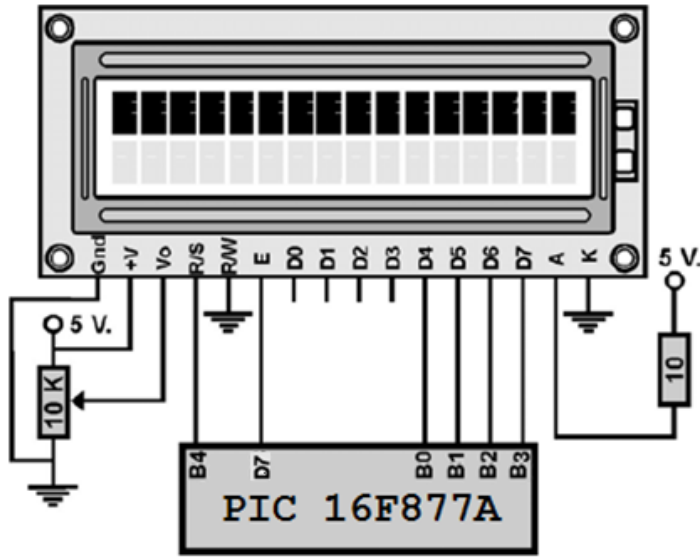
2. OBJETIVOS:

- Diseñar y armar en el software de simulación ISIS de PROTEUS, un circuito que permita simular la decodificación en una LCD el tema de tesis y el nombre de su expositor.
- Diseñar un algoritmo y un programa en el software de programación MICROCODE, que permita decodificar en una LCD el tema de tesis y su expositor.
- Analizar el comportamiento del circuito y el algoritmo del programa.

3. MATERIALES:

Cantidad	Designación	Características
1	LCD	16 Bits
1	Resistor	10Ω
1	Resistor	4.7kΩ
1	Potenciómetro	10kΩ
1	Pulsadores	
2	Capacitores Cerámicos	22pF ó 33pF
1	Cristal de Cuarzo	4MHz
1	Microcontrolador	PIC (16F877A ó 16F628A)

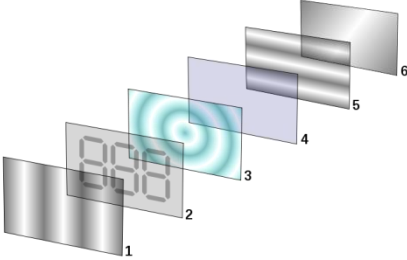
4. ESQUEMA:



5. SISTEMA CATEGORIAL:

5.1. LCD

Una **pantalla de cristal líquido** o **LCD** (sigla del inglés *liquid crystal display*) es una pantalla delgada y plana formada por un número de píxeles en color o monocromos colocados delante de una fuente de luz o reflectora. A menudo se utiliza en dispositivos electrónicos de pilas, ya que utiliza cantidades muy pequeñas de energía eléctrica.



PANTALLA DE CRISTAL LÍQUIDO (LCD)

- Film de filtro vertical para polarizar la luz que entra.
- Sustrato de vidrio con electrodos de Óxido de Indio ITO. Las formas de los electrodos determinan las formas negras que aparecen cuando la pantalla se enciende y apaga. Los cantos verticales de la superficie son suaves.
- Cristales líquidos "Twisted Nematic" (TN).
- Sustrato de vidrio con film electrodo común (ITO) con los cantos horizontales para alinearse con el filtro horizontal.
- Film de filtro horizontal para bloquear/permitir el paso de luz.
- Superficie reflectante para enviar/devolver la luz al espectador. En un LCD retroiluminado, esta capa es reemplazada por una fuente luminosa.

FUNCIONAMIENTO LCD

Cuando se aplica un voltaje a través de los electrodos, una fuerza de giro orienta las moléculas de cristal líquido paralelas al campo eléctrico, que distorsiona la estructura helicoidal (esto se puede resistir gracias a las fuerzas elásticas desde que las moléculas están limitadas a las superficies). Esto reduce la rotación de la polarización de la luz incidente, y el dispositivo aparece gris. Si la tensión aplicada es lo suficientemente grande, las moléculas de cristal líquido en el centro de la capa son casi completamente desenrolladas y la polarización de la luz incidente no es rotada ya que pasa a través de la capa de cristal líquido. Esta luz será principalmente polarizada perpendicular al segundo filtro, y por eso será bloqueada y el pixel aparecerá negro. Por el control de la

tensión aplicada a través de la capa de cristal líquido en cada píxel, la luz se puede permitir pasar a través de distintas cantidades, constituyéndose los diferentes tonos de gris.

6. PROCEDIMIENTO:

6.1. Para armar el circuito de simulación utilizamos ISIS de Proteus

Profesional y su interfaz.

6.1.1. En el buscador de Proteus seleccionamos el Pic16F877A y lo ubicamos en la pantalla principal.

6.1.2. En el buscador de Proteus seleccionamos el cristal de cuarzo, lo ubicamos en la pantalla principal y lo conectamos al oscilador 1 y 2 del PIC.

6.1.3. En el buscador de Proteus seleccionamos dos capacitores de 22pF ó 33pF, los ubicamos en la pantalla principal y los conectamos al oscilador 1 y 2 del PIC y a GND.

6.1.4. En el buscador de Proteus seleccionamos un pulsador, la ubicamos en la pantalla principal y la conectamos al reset del PIC y a GND.

6.1.5. En el buscador de Proteus seleccionamos una resistencia de 4,7k Ω , la ubicamos en la pantalla principal y la conectamos al reset del PIC y a VCC.

6.1.6. En el buscador de Proteus seleccionamos una LCD de 16 bits, la ubicamos en la pantalla principal y la conectamos los pines del 11 al 14, al puerto del B0 al B3 del PIC; el pin 4 de la LCD al puerto B4 del PIC; el pin 6 de la LCD al puerto D7 del PIC; y los pines 5 y 16 de la LCD a GND.

6.1.7. En el buscador de Proteus seleccionamos una resistencia de 10Ω , la ubicamos en la pantalla principal y la conectamos entre el ÁNODO de la LCD y VCC.

6.1.8. En el buscador de Proteus seleccionamos un potenciómetro de $10k\Omega$, lo ubicamos en la pantalla principal y la conectamos entre GND y VCC de la LCD y la fuente principal y la toma central al pin 3 de la LCD.

6.2. Para la programación del PIC utilizamos Microcode Studio.

6.2.1. Transformamosa digitales todos los puertos analógicos del PIC.

6.2.2. Declaramos el puerto D como salida.

6.2.3. Le damos un retardo de 200 milisegundos para esperar que funcione el LCD.

6.2.4. Creamos cuatro variables (x, y, abc, xyz) de 255 bits.

6.2.5. Le asignamos un nombre al programa.

6.2.6. Configuramos el desplazamiento hacia la izquierda de la LCD.

6.2.7. Limpiamos la LCD.

6.2.8. Ubicamos el cursor de la LCD en la casilla 17.

6.2.9. Creamos una secuencia **FOR**, para poder sacar el nombre del TESISISTA en la LCD.

6.2.10. Dentro de la secuencia guardamos carácter por carácter el texto en la variable abc.

6.2.11. Ahora mostramos en la LCD el contenido de abc.

6.2.12. Le damos una pausa al programa de 400 milisegundos.

6.2.13. Creamos una secuencia **FOR**, para poder sacar el nombre del proyecto de tesis en la LCD.

6.2.14. Dentro de la secuencia guardamos carácter por carácter el texto en la variable xyz.

6.2.15. Ahora mostramos en la LCD el contenido de xyz.

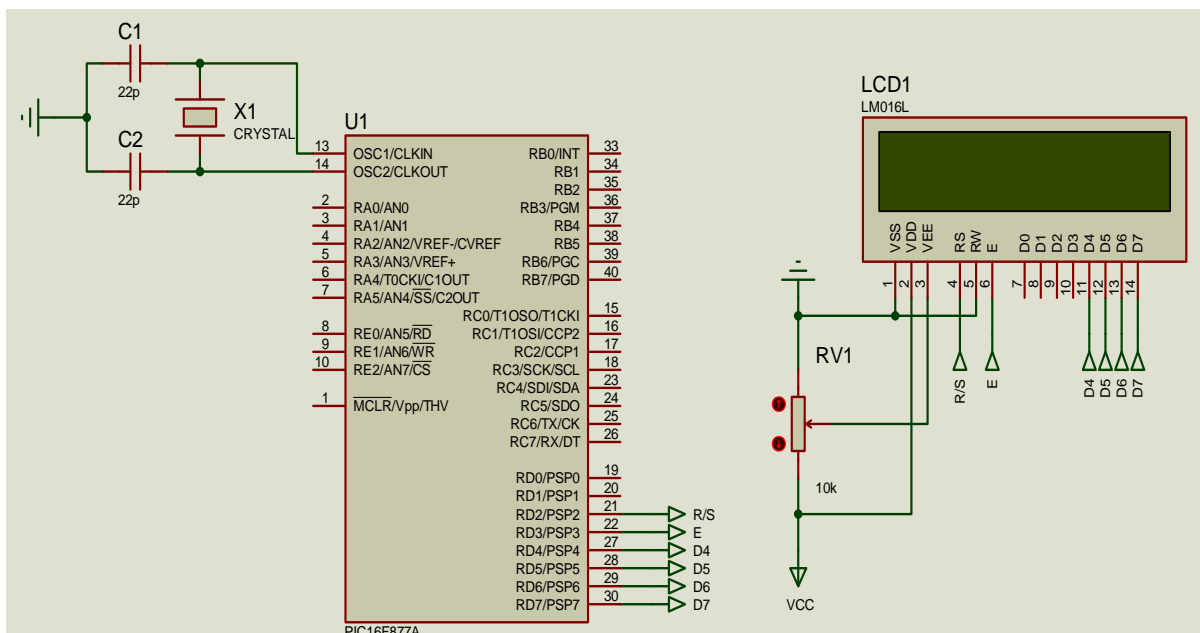
6.3. Antes de la ejecución del programa en Proteus es indispensable sincronizar los osciladores del cristal y PIC en 4MHz, seteando dicho valor en propiedades de los dispositivos.

6.4. Cargamos el archivo **.HEX** generado por Microcode en el PIC de Proteus y ejecutamos la simulación, y los nueve leds deben parpadear en intervalos de un segundo.

6.5. Para armar la práctica en el módulo es necesario comprar los materiales descritos en la lista y adicionalmente traer un pelacables y cable UTP para conexiones.

7. DESARROLLO:

El circuito armado completamente y la simulación realizada en Proteus Profesional queda de la siguiente forma:



El algoritmo correspondiente y programa correctamente elaborado en Microcode decodificación y manejo de una LCD, armados en la simulación de Proteus nos queda de la siguiente forma:

```

; DECODIFICACIÓN Y MANEJO DE UNA LCD
DEFINE LCD_DREG PORTD ; utiliza el puerto D como salida
DEFINE LCD_DBIT 4 ; empieza utilizando el puerto D4 hasta el D7
DEFINE LCD_RSREG PORTD ; utiliza el puerto D como salida
DEFINE LCD_RSBIT 2 ; utiliza el puerto D2 para R/S
DEFINE LCD_EREG PORTD ; utiliza el puerto D como salida
DEFINE LCD_EBIT 3 ; utiliza el puerto D3 para E (enable)
PAUSE 200 ; retardo para esperar que funcione el LCD
x VAR BYTE ; crear la variable x de 255
y VAR BYTE ; crear la variable y de 255
abc VAR BYTE ; crear la variable abc de 255
xyz VAR BYTE ; crear la variable xyz de 255
LCDOUT $FE,$7 ; configura para desplazamiento izquierdo
LCDOUT $FE,1 ; limpiar pantalla
Practica8:
LCDOUT $FE,$90 ; ubica el cursor en la casilla 17
FOR x = 0 TO 23 ; repetir 24 veces
LOOKUP x,["Tesista: Pedro Martinez"],abc ; tomar caracter por caracter y guardar en abc
LCDOUT, abc ; sacar en LCD el contenido de abc
PAUSE 400 ; esperar 400 mls
NEXT ;siguiente repetición
FOR x = 0 TO 23 ; repetir 24 veces
LOOKUP y,["PRACTICAS DE MICROCONTROLADORES"],xyz; tomar carácter x carácter y guardar en abc
LCDOUT, xyz ; sacar en LCD el contenido de abc
PAUSE 400 ; esperar 400 mls
NEXT ;siguiente repetición
GOTO Practica8
END

```

8. PREGUNTAS DE CONTROL:

8.1. ¿Para qué sirve una LCD?

La LCD es un instrumento de visualización, que permite mostrar letras de la A-Z y números 0-9 y ciertos símbolos.

8.2. ¿Cuántos bits son necesarios para escribir en una LCD?

De los 8 bits de entrada que posee la LCD, son únicamente necesarios 4 bits para poder visualizar frases, cantidades numéricas, etc.

8.3. ¿De qué depende el contraste de una LCD?

El nivel de voltaje que ingresa en el pin de contraste de la LCD, del divisor de voltaje es el que aumenta o disminuye el brillo en la misma.

PRACTICA 9

1. TEMA:

Decodificación y Manejo de una Matriz de Leds de 5x7.

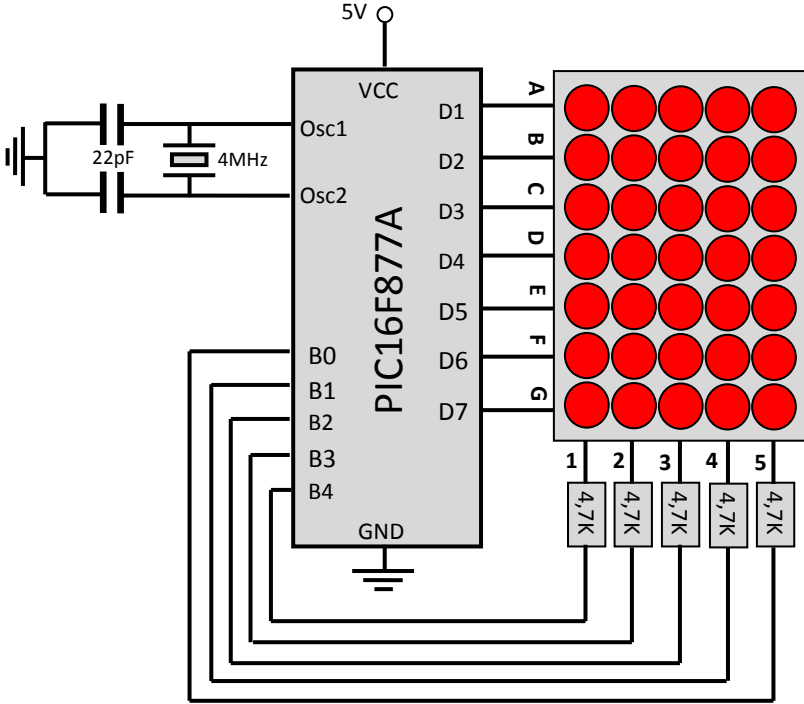
2. OBJETIVOS:

- Diseñar y armar en el software de simulación ISIS de PROTEUS, un circuito que permita simular la decodificación de un muñeco en movimiento en una matriz de led.
- Diseñar un algoritmo y un programa en el software de programación MICROCODE, que permita decodificar un muñeco en movimiento en una matriz de led.
- Analizar el comportamiento del circuito y el algoritmo del programa.

3. MATERIALES:

Cantidad	Designación	Características
1	Matriz de leds	5x7
1	Resistor	4.7k Ω
5	Resistores	220 Ω
1	Pulsadores	NA
2	Capacitores Cerámicos	22pF ó 33pF
1	Cristal de Cuarzo	4MHz
1	Microcontrolador	PIC (16F877A ó 16F628A)

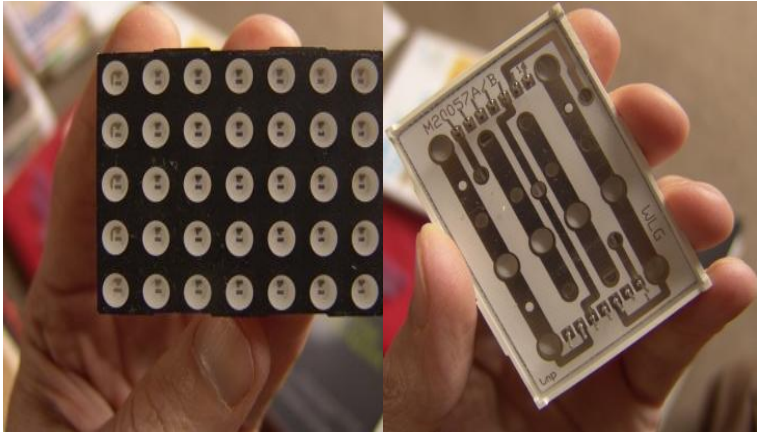
4. ESQUEMA:



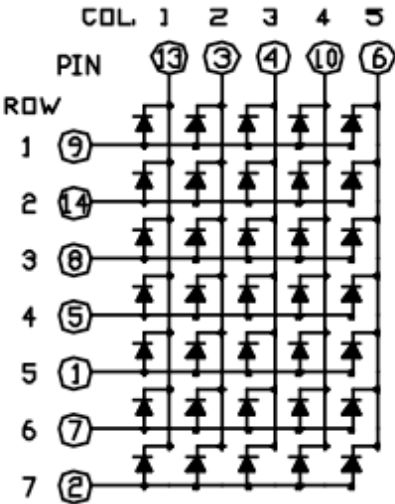
5. SISTEMA CATEGORIAL:

5.1. Matriz de Leds

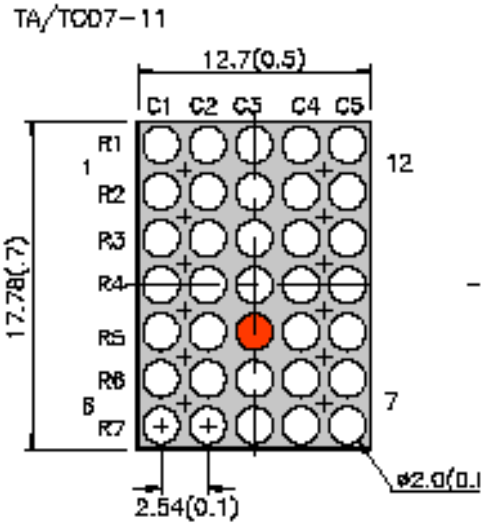
Una matriz de Leds consiste en un arreglo de Leds que pueden ser encendidos y apagados individualmente desde un microcontrolador. Pueden pensar en ella como una pantalla de pocos pixeles en los cuales pueden presentar gráficos y textos, tanto estáticos como en movimiento.



En la siguiente figura se presenta una matriz de 5x7 Leds, donde las columnas corresponden a los cátodos (deben ir conectados a tierra) y las filas corresponden a los ánodos (deben ir conectados a voltaje).



Para encender un LED específico debe ubicarse la intersección entre la columna y la línea correspondiente. Por ejemplo, para encender el LED que se muestra en la figura siguiente, la columna 3 (C3) debe ser conectada a tierra (a través de una resistencia de 220 ohm) mientras que la fila 5 (R5) es conectada a voltaje (5v).



5.2. Definición de matriz

Es una tabla bidimensional de números consistente en cantidades abstractas que pueden sumarse y multiplicarse.

Una matriz es un cuadrado o tabla de números ordenados. Se llama matriz de dimensión $m \times n$ a un conjunto de números reales dispuestos en m filas y n columnas de la siguiente forma

$$\begin{pmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1} & a_{m2} & \dots & a_{mn} \end{pmatrix}$$

6. PROCEDIMIENTO:

6.1. Para armar el circuito de simulación utilizamos ISIS de Proteus

Profesional y su interfaz.

6.1.1. En el buscador de Proteus seleccionamos el Pic16F877A y lo ubicamos en la pantalla principal.

6.1.2. En el buscador de Proteus seleccionamos el cristal de cuarzo, lo ubicamos en la pantalla principal y lo conectamos al oscilador 1 y 2 del PIC.

6.1.3. En el buscador de Proteus seleccionamos dos capacitores de 22pF ó 33pF, los ubicamos en la pantalla principal y los conectamos al oscilador 1 y 2 del PIC y a GND.

6.1.4. En el buscador de Proteus seleccionamos un pulsador, lo ubicamos en la pantalla principal y lo conectamos al reset del PIC y a GND.

6.1.5. En el buscador de Proteus seleccionamos una resistencia de 4,7kΩ, ubicamos seis de las mismas en la pantalla principal, una la conectamos al reset del PIC y a VCC, y las demás a los puertos B0, B1, B2, B3, B4.

6.1.6. En el buscador de Proteus seleccionamos una matriz 5x7 de leds, la ubicamos en la pantalla principal, las filas las conectamos al puerto D y las columnas a las resistencias de 4,7kΩ de los puertos B0, B1, B2, B3, B4.

6.2. Para la programación del PIC utilizamos Microcode Studio.

6.2.1. Transformamosa digitales todos los puertos analógicos del PIC.

6.2.2. Declaramos el puerto B como salida.

- 6.2.3. Le asignamos un nombre al programa.
- 6.2.4. Creamos una secuencia **FOR**, para dibujar en los leds un muñeco.
- 6.2.5. En la secuencia **FOR**, a través de operaciones de fila y columna, encendemos los leds necesarios para formar la figura en una posición inicial.
- 6.2.6. Le damos una pausa al programa de 4 milisegundos.
- 6.2.7. En una segunda secuencia **FOR**, a través de operaciones de fila y columna, encendemos los leds necesarios para formar una nueva figura en una nueva posición distinta a la inicial, de tal forma que parezca que el muñeco se mueve.
- 6.2.8. Le damos una pausa al programa de 4 milisegundos.
- 6.2.9. En una tercera secuencia **FOR**, a través de operaciones de fila y columna, encendemos los leds necesarios para formar una nueva figura en una nueva posición distinta a la segunda, de tal forma que parezca que el muñeco se mueve.
- 6.2.10. Le damos una pausa al programa de 4 milisegundos.
- 6.2.11. En una cuarta secuencia **FOR**, a través de operaciones de fila y columna, encendemos los leds necesarios para formar una nueva figura en una nueva posición distinta a la tercera, de tal forma que parezca que el muñeco se mueve.
- 6.2.12. Repetimos el programa de manera indefinida, para darle un movimiento indefinido.

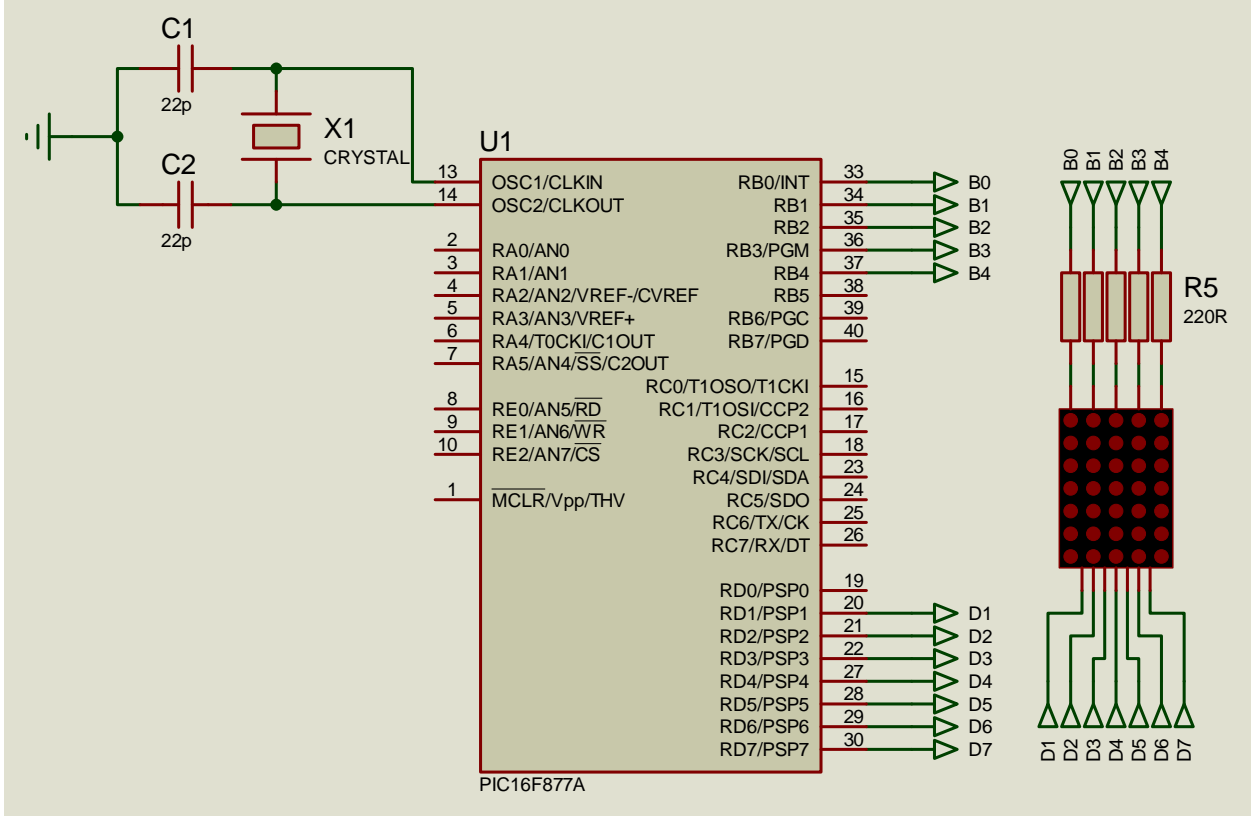
6.3. Antes de la ejecución del programa en Proteus es indispensable sincronizar los osciladores del cristal y PIC en 4MHz, seteando dicho valor en propiedades de los dispositivos.

6.4. Cargamos el archivo **.HEX** generado por Microcode en el PIC de Proteus y ejecutamos la simulación, y los nueve leds deben parpadear en intervalos de un segundo.

6.5. Para armar la práctica en el módulo es necesario comprar los materiales descritos en la lista y adicionalmente traer un pelacables y cable UTP para conexiones.

7. DESARROLLO:

El circuito armado completamente y la simulación realizada en Proteus Profesional queda de la siguiente forma:



El algoritmo correspondiente y programa correctamente elaborado en Microcode Studio para decodificación y manejo de una matriz de leds de 5x7, armados en la simulación de Proteus nos queda de la siguiente forma:

```

                                ;DECODIFICACIÓN Y MANEJO DE UNA MATRIZ DE LEDS DE 5X7
cmcon = 7                        ;convierte el puerto A en digital
trisa=0                          ;hace salida todo el puerto A
trisd=0                          ;convierte en salidas el puerto B
xVAR BYTE                        ;crea variable x de 255
Practica9:
  FOR x = 1 TO 10                ;repetir esta secuencia 10 veces
    portb=%00000001 :portd=%11111101 :PAUSE 4      ;esperar 4 mls, total 20 mls cada
    portb=%00000010 :portd=%00011011 :PAUSE 4      ;escena
    portb=%00000100 :portd=%11100000 :PAUSE 4
    portb=%00001000 :portd=%00011011 :PAUSE 4
    portb=%00010000 :portd=%11111101 :PAUSE 4
  NEXT
  FOR x = 1 TO 10                ;repetir esta secuencia 10 veces
    portb=%00000001 :portd=%11111011 :PAUSE 4      ;esperar 4 mls, total 20 mls cada
    portb=%00000010 :portd=%00011011 :PAUSE 4      ;escena
    portb=%00000100 :portd=%11100000 :PAUSE 4
    portb=%00001000 :portd=%00011011 :PAUSE 4
    portb=%00010000 :portd=%11111011 :PAUSE 4
  NEXT
  GOTO Practica9                ;ir a Practica9
END

```

8. PREGUNTAS DE CONTROL:

8.1. ¿Qué es una matriz?

Es un arreglo de filas y columnas.

8.2. ¿Cómo funciona una matriz 5x7 de leds?

Con niveles de voltaje entre 2.4V-DC y 5V-DC en las 5 columnas y GND en las 7 filas.

8.3. ¿Qué se puede hacer con una matriz de leds?

Las aplicaciones son varias entre las que puedo mencionar, el diseño de rótulos numéricos, de letras o figuras específicas.

8.4. ¿Cuánta corriente consume cada led del arreglo?

Una corriente aproximada de 30mA.

PRACTICA 10

1. TEMA:

Decodificación y Manejode un Teclado 4x4 con Relés de 110V.

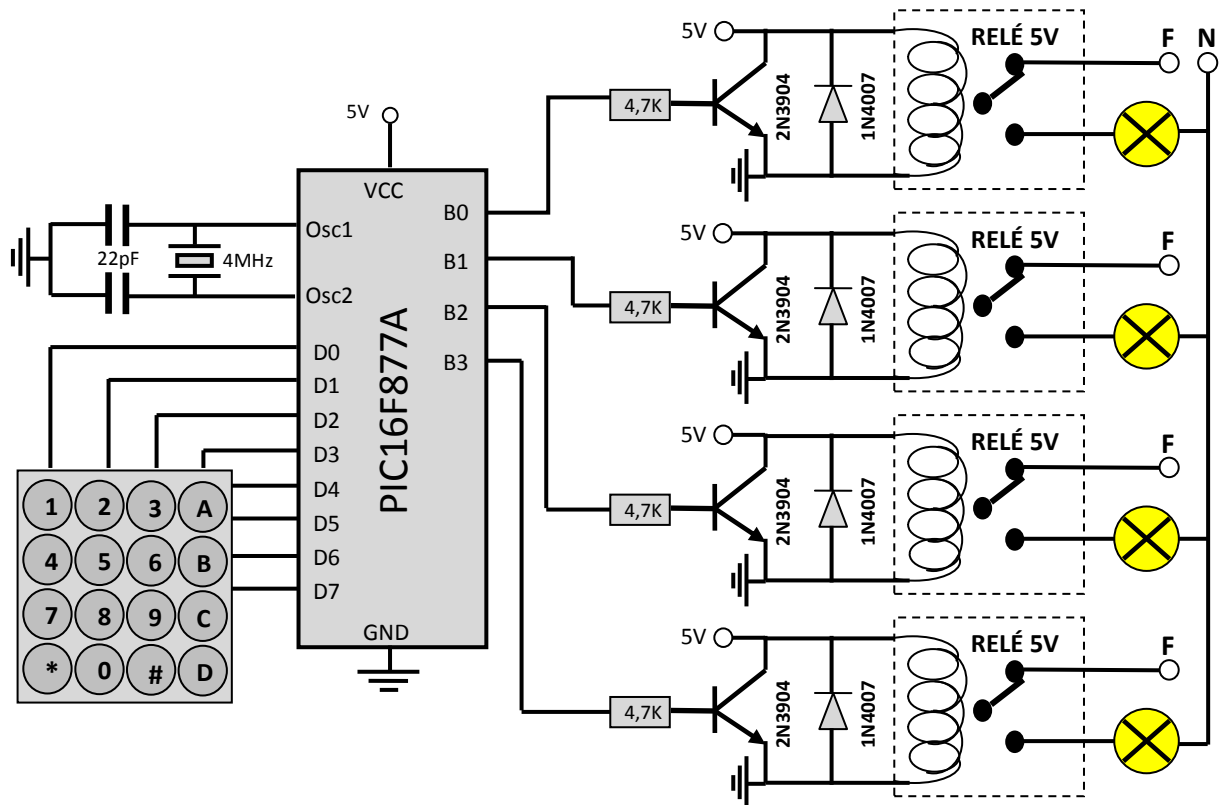
2. OBJETIVOS:

- Diseñar y arma en el software de simulación ISIS de PROTEUS, un circuito que permita la simulación de la decodificación de un teclado y la generación de un código BCD que se visualice en focos de 110V.
- Diseñar un algoritmo y un programa en el software de programación MICROCODE, que permita decodificar un teclado y generar un código BCD que se visualice en focos de 110V.
- Analizar el comportamiento del circuito y el algoritmo del programa.

3. MATERIALES:

Cantidad	Designación	Características
4	Boquilla	110V-AC
4	Foco	110V-AC (30W)
4	Relé	5V-DC
4	Diodo	1N4007
4	Transistor	2N3904
13	Resistores	4,7kΩ (1/4W)
1	Pulsadores	NA
2	Capacitores Cerámicos	22pF ó 33pF
1	Cristal de Cuarzo	4MHz
1	Microcontrolador	PIC (16F877A ó 16F628A)

4. ESQUEMA:



5. SISTEMA CATEGORIAL:

5.1. Teclado Matricial 4X4

Un **teclado** es un periférico que consiste en un sistema de teclas, como las de una máquina de escribir, que te permite introducir datos a un ordenador o dispositivo digital.

Dispositivo de entrada de datos que consta de 16 teclas o pulsadores, dispuestos e interconectados en filas y columnas. Dispone de un conector SIL (Single In Line) macho de 8 pines que se corresponden con las 4 filas y las cuatro columnas de las que dispone.

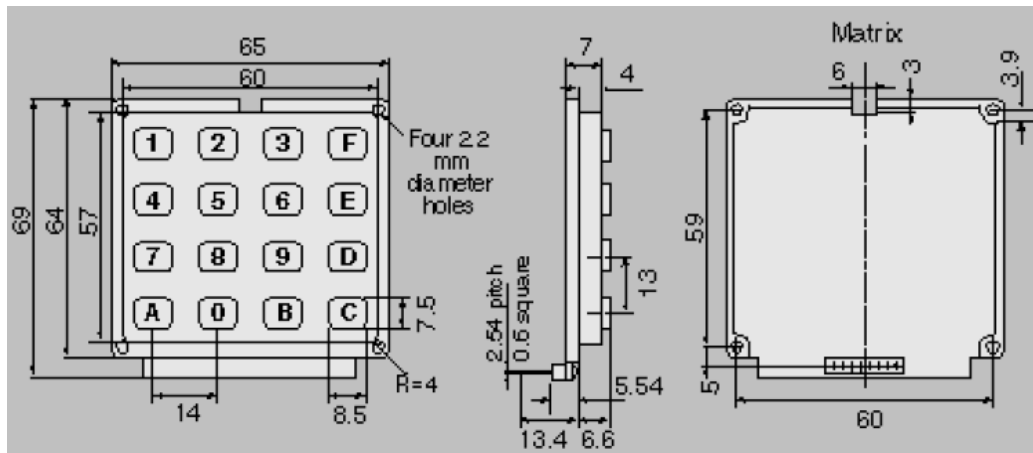


Figura 2.16. Teclado matricial 4x4

Como se ve en la figura las filas del teclado se conectan a las líneas RB0, RB1, RB2, RB3 configuradas como salidas mientras que las columnas se conectan a las líneas RB4, RB5, RB6, RB7 configuradas como entradas.

Para detectar si hay una tecla pulsada se aplica el siguiente procedimiento: Se hace un barrido por cada una de las filas, inicialmente se aplica a la primera fila un nivel bajo y a las otras tres un nivel alto si en este instante se pulsa alguna tecla de esta fila entonces el nivel bajo aparecerá en la columna en la que se encontraba la tecla, seguidamente se pasa a verificar la segunda fila enviando un nivel bajo a esta y a las otras un nivel alto y así sucesivamente. Verificando los niveles de los terminales de las columnas se podrá deducir si hay alguna tecla pulsada en la fila donde se esta aplicando el nivel bajo, considerando que si no

hay alguna tecla pulsada en las entradas del PIC RB4, RB5, RB6, RB7 se encontrara un nivel alto.



6. PROCEDIMIENTO:

6.1. Para armar el circuito de simulación utilizamos ISIS de Proteus

Profesional y su interfaz.

6.1.1. En el buscador de Proteus seleccionamos el Pic16F877A y lo ubicamos en la pantalla principal.

6.1.2. En el buscador de Proteus seleccionamos el cristal de cuarzo, lo ubicamos en la pantalla principal y lo conectamos al oscilador 1 y 2 del PIC.

6.1.3. En el buscador de Proteus seleccionamos dos capacitores de 22pF ó 33pF, los ubicamos en la pantalla principal y los conectamos al oscilador 1 y 2 del PIC y a GND.

6.1.4. En el buscador de Proteus seleccionamos un pulsador, lo ubicamos en la pantalla principal y lo conectamos al reset del PIC y a GND.

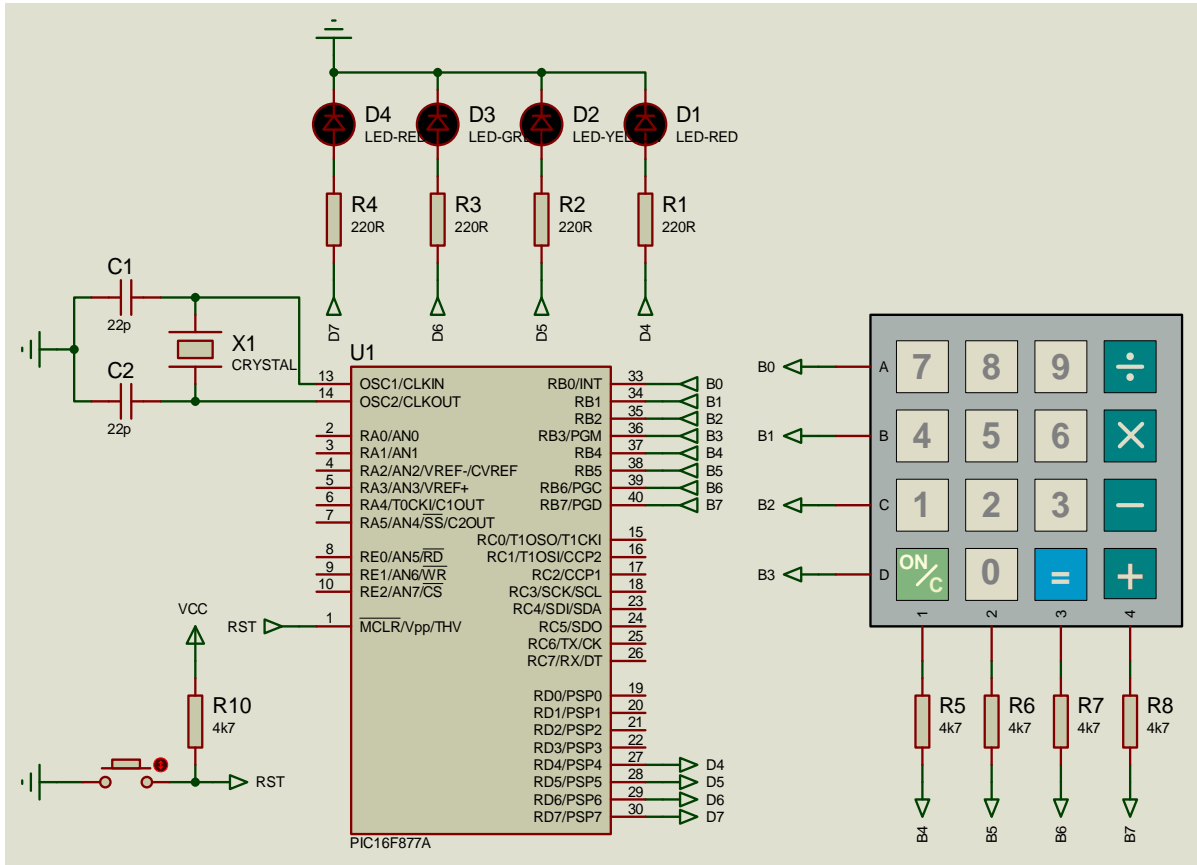
- 6.1.5. En el buscador de Proteus seleccionamos un transistor 2N3904, ubicamos cuatro de los mismos en la pantalla principal y la *BASE* de todos la conectamos a las resistencias de 4,7K Ω de los puertos B0, B1, B2, B3; el *EMISOR* lo conectamos a VCC y el *COLECTOR* a GND.
- 6.1.6. En el buscador de Proteus seleccionamos una resistencia de 4,7k Ω , ubicamos cinco de las mismas en la pantalla principal, una la conectamos al reset del PIC y a VCC, las demás a los puertos B0, B1, B2, B3.
- 6.1.7. En el buscador de Proteus seleccionamos un diodo 1N4007, ubicamos cuatro de los mismos en la pantalla principal y los conectamos entre el *EMISOR* y *COLECTOR* de los transistores.
- 6.1.8. En el buscador de Proteus seleccionamos un relé de 5V, ubicamos cuatro de los mismos en la pantalla principal y los conectamos entre el *EMISOR* y *COLECTOR* del transistor la parte de la bobina y el switch a 110V-AC.
- 6.1.9. En el buscador de Proteus seleccionamos una lámpara de 110V-AC, ubicamos cuatro de las mismas en la pantalla principal y las conectamos al switch del relé y a GND.
- 6.1.10. En el buscador de Proteus seleccionamos un teclado 4x4, lo ubicamos en la pantalla principal y lo conectamos a los pines del puerto D.
- 6.2. Para la programación del PIC utilizamos Microcode Studio.
- 6.2.1. Transformamosa digitales todos los puertos analógicos del PIC.
- 6.2.2. Declaramos el puerto B como salida.

- 6.2.3. Le asignamos un nombre al programa.
- 6.2.4. Creamos varias secuencias **IF**, para poder decodificar cada número del teclado, a través de operaciones de fila y columna, y le asignamos un valor numérico.
- 6.2.5. Una vez decodificado el teclado, el resultado (activo a bajo) lo sacamos por el puerto B para poder conmutar los relés y así encender los focos de 110V-AC, y visualizar en código BCD el valor.
- 6.2.6. Le damos una pausa al programa de 10 milisegundos.
- 6.2.7. Repetimos el programa de manera indefinida.
- 6.3. Antes de la ejecución del programa en Proteus es indispensable sincronizar los osciladores del cristal y PIC en 4MHz, seteando dicho valor en propiedades de los dispositivos.
- 6.4. Cargamos el archivo **.HEX** generado por Microcode en el PIC de Proteus y ejecutamos la simulación, y los nueve leds deben parpadear en intervalos de un segundo.
- 6.5. Para armar la práctica en el módulo es necesario comprar los materiales descritos en la lista y adicionalmente traer un pelacables y cable UTP para conexiones.

7. DESARROLLO:

El circuito armado completamente y la simulación realizada en Proteus

Profesional queda de la siguiente forma:



El algoritmo correspondiente y programa correctamente elaborado en Microcode Studio para la decodificación y manejo de un teclado 4x4 con relés de 110V armados en la simulación de Proteus nos queda de la siguiente forma:

;DECODIFICACION Y MANEJO DE UN TECLADO 4x4 CON RELES DE 110V

```

TRISD = 0           ; todo el puerto A es configurado como salidas
A VAR PORTB.0      ; nombres para los pines de las filas
B VAR PORTB.1
C VAR PORTB.2
D VAR PORTB.3
UNO VAR PORTB.4    ; nombres para los pines de las columnas
DOS VAR PORTB.5
TRES VAR PORTB.6
CUATRO VAR PORTB.7

Practica10:
    PORTD = %00000000
LOW A           ; hacer bajo la 1era fila
IF UNO = 0 THEN PORTD = %01110000 ; si la 1ra tecla es presionada sacar 7
IF DOS = 0 THEN PORTD = %10000000 ; si la 2da tecla es presionada sacar 8
IF TRES = 0 THEN PORTD = %10010000 ; si la 3ra tecla es presionada sacar 9
IF CUATRO = 0 THEN PORTD = %10100000 ; si la 4ta tecla es presionada sacar 10
HIGH A          ; poner en alto la 1era fila
LOW B           ; hacer bajo la 2da fila
IF UNO = 0 THEN PORTD = %01000000 ; si la 1ra tecla es presionada sacar 4
IF DOS = 0 THEN PORTD = %01010000 ; si la 2da tecla es presionada sacar 5
IF TRES = 0 THEN PORTD = %01100000 ; si la 3ra tecla es presionada sacar 6
IF CUATRO = 0 THEN PORTD = %10110000 ; si la 4ta tecla es presionada sacar 11
HIGH B          ; poner en alto la 2da fila
LOW C           ; hacer bajo la 3ra fila
IF UNO = 0 THEN PORTD = %00010000 ; si la 1ra tecla es presionada sacar 1
IF DOS = 0 THEN PORTD = %00100000 ; si la 2da tecla es presionada sacar 2
IF TRES = 0 THEN PORTD = %00110000 ; si la 3ra tecla es presionada sacar 3
IF CUATRO = 0 THEN PORTD = %11000000 ; si la 4ta tecla es presionada sacar 12
HIGH C          ; poner en alto la 3ra fila
LOW D           ; hacer bajo la 4ta fila
IF UNO = 0 THEN PORTD = %11110000 ; si la 1ra tecla es presionada sacar 15
IF DOS = 0 THEN PORTD = %00000000 ; si la 2da tecla es presionada sacar 0
IF TRES = 0 THEN PORTD = %11100000 ; si la 3ra tecla es presionada sacar 14
IF CUATRO = 0 THEN PORTD = %11010000 ; si la 4ta tecla es presionada sacar 13
HIGH D          ; poner en alto la 4ta fila
PAUSE 10        ; pausa de 10 milisegundos
GOTO Practica10 ; continuar con el barrido de teclas
END

```

8. PREGUNTAS DE CONTROL:

8.1. ¿Qué es un teclado?

Es un arreglo de pulsadores que permiten a los usuarios ingresar códigos de bits que son relacionados con letras o números.

8.2. ¿Cómo funciona un teclado?

A través de las relaciones de filas y columnas, se ingresa Vcc a las filas y GND a las columnas con el afán de configurar los pulsadores que lo conforman.

8.3. ¿Cómo se configura un teclado 4x4?

Se coloca una resistencia entre Vcc y el pin de las filas para generar los niveles de corriente que permiten identificar al PIC un 0 lógico o un 1 lógico.

VI. CONCLUSIONES

6.1. CONCLUSIONES:

- Dentro de un contexto importantísimo como es el campo social y estudiantil, este proyecto se convierte en una de las mejores alternativas para el aprendizaje de la correcta programación, simulación e implementación de proyectos prácticos con microcontroladores o microprocesadores.
- La construcción de un módulo para prácticas de microcontroladores resultó eficaz, versátil y moderno; el mismo que permite utilizar un microcontrolador como el PIC 16F877A, para potencializar al máximo las cualidades del mismo en el desarrollo de las prácticas; además que permite la utilización de cualquier otro microcontrolador o microprocesador.
- La utilización del microcontrolador PIC 16F877A es el elemento fundamental en la construcción y funcionamiento del módulo en este proyecto, por su versatilidad, confiabilidad y eficiencia ya que reduce errores, fallas y tiempos de respuesta.
- Las prácticas planteadas en el presente proyecto contienen y mantienen una estructura de tal forma, que cualquier estudiante o lector del presente trabajo investigativo pueda implementarlas de manera correcta y adecuada, sin ningún tipo de errores en el módulo diseñado.

VII. RECOMENDACIONES

7.1. RECOMENDACIONES

- Para la correcta realización de las distintas prácticas planteadas es necesario seguir el procedimiento descrito paso a paso en cada práctica.
- El módulo está diseñado para no dañarse en caso de cortocircuitos; por lo que, si alguno se aconteciere lo único necesario es el cambio del respectivo fusible.
- El uso del PIC 16F877A, se debe a su robustez y resistencia ante cortos, además de gran cantidad de pines de conexión, que permite hacer funcionar varios circuitos al mismo tiempo.
- En el módulo antes conectar la corriente y hacer funcionar los distintos circuitos que posee es necesario que se verifique las conexiones realizadas en el protoboard, para evitar cortocircuitos y que los distintos elementos del módulo se quemen.

VIII. BIBLIOGRAFÍA

8.1. BIBLIOGRAFÍA:

LIBROS:

- [1] ARTIGAS, J; BARRAGÁN, L; ORRITE, C. (1999). **Aplicaciones y problemas de electrónica digital**. España: Textos docentes Zaragoza. p.341.
- [2] BARNETT RICHARD, COX SARAH y O´CULL LARRY. (2003). **Embedded C Proqraming and the Atmel AVR**. Learning, Inc. Thomson.
- [3] GAJSKI, Daniel D. (1997). **Principios de diseño digital**. Madrid: Prentice Hall Iberia. S/f. p.488. "Principles of digital design". Traducido por: Alberto Prieto Espinosa.
- [4] LLORIS, Antonio. PRIETO, Alberto. (1996). **Diseño lógico**. Madrid: McGraw Hill. S/f. p.403.
- [5] REYES CARLOS A. (2008). **Microcontroladores PIC Programación en Basic**. RISPERGRAF. Tercera Edición.
- [6] TOCCI, RONALD J. y WIDMER, NEAL S. (2003). **SISTEMAS DIGITALES: Principios y Aplicaciones**. PEARSON EDUCACIÓN. Mexico. Octava Edición.

SITIOS WEB DE INTERÉS.

- [7] "Productos con y para PIC´S", www.electronicaestudio.com
- [8] "Descarga manual de pbp, teoría del PIC", www.frino.com.ar
- [9] "Descarga de programa lc-prog106A.zip", www.IC-prog.com

- [10] “Descarga de programas: Microcode y Pbp demo”,
www.mecanique.co.uk
- [11] “Compilador PicBasic Pro, Ejemplos en pbp”, www.melabs.com
- [12] “Productos e información de la familia de PIC´S”,
www.microchip.com
- [13] “Entrenadores, grabadores y libros de PIC´S”,
www.mikroelektronika.co.yu
- [14] “Todo tipo de temas relacionado al PIC”, www.monografias.com
- [15] “Diagramas de proyectos con PIC´S”, www.pablin.com.ar
- [16] “Historia del PIC”, www.Redeya.com
- [17] “Ejemplos en pbp, información de dispositivos”, www.rentron.com
- [18] “Descarga manual en español de pbp”, www.todopic.com.ar
- [19] “Información sobre LCD 2x16, motores PAP, etc.”,
www.x-robotics.com

IX. ANEXOS

ANEXO A

CÓDIGOS DE ACCESO RÁPIDO PARA COMPONENTES DE ISIS

COMPONENTE	CÓDIGO	EJEMPLO
RESISTOR	MINRES	MINRES220R, para resistor de 220Ω MINRES15K, para resistor de 15kΩ
POTENCIÓMETRO	POT-LIN	POT-LIN, en las propiedades se debe setear el valor de resistencia necesaria.
CAPACITOR CERÁMICO	CERAMIC	CERAMIC22P, para capacitor de 22pF CERAMIC22N, para capacitor de 22nF
CAPACITOR ELECTROLÍTICO	GENELECT	GENELECT220U10V, capacitor genérico electrolítico de 220uF a 10 voltios. GENELECT22U16V, capacitor genérico electrolítico de 22uF a 16 voltios
MICROCONTROLADOR	PIC	PIC16F877A, para el PIC 16F877A PIC16F628A, para el PIC 16F628A
CRISTAL	CRYSTAL	CRYSTAL, en las propiedades se debe setear la frecuencia necesaria 4MHz,2MHz
LED	LED	LED-YELLOW, para led color amarillo LED-BLUE, para led color azul
DISPLAY	7SEG	7SEG-COM-AN-RED, display 7 segmentos rojo.
RELÉ	RELAY	RELAY, relé animado con un switch RELAY2P, relé animado con doble switch
TECLADO MATRICIAL	KEYPAD	KEYPAD-PHONE, teclado para teléfono KEYPAD-SMALLCALC, teclado calculador
LCD	LM016L	LM016L, LCD de 16x32 bits
MATRIZ DE LED'S	MATRIX	MATRIX-5x7-BLUE, matriz de 5x7 azul MATRIX-5x7-RED, matriz de 5x7 rojo
PULSADOR	BUTTON	BUTTON, pulsador on/off
TRANSISTOR		2N2222, para transistor bipolar 2N2222 2N3904, para transistor bipolar 2N3904 2N3906, para transistor bipolar 2N3906

ANEXO B

DECLARACIONES DE MICROCODE

DECLARACIONES	APLICACIÓN
@	Inserta una línea de código ensamblador
ADCIN	Lee el conversor analógico
ASM...ENDASM	Insertar una sección de código ensamblador
BRANCH	GOTO computado (equivale a ON..GOTO)
BRANCHL	BRANCH fuera de página (BRANCH Largo)
BUTTON	Anti-rebote y auto-repetición de entrada en el pin especificado
CALL	Llamada a subrutina de ensamblador
CLEAR	Hace cero todas las variables
CLEARWDT	Hace cero el contador del WatchdogTimer
COUNT	Cuenta el número de pulsos en un pin
DATA	Define el contenido inicial en un chip EEPROM
DEBUG	Señal asincrónica de salida en un pin fijo y baud
DEBUGIN	Señal asincrónica de entrada en un pin fijo y baud
DISABLE	Deshabilita el procesamiento de ON INTERRUPT, ON DEBUG
DISABLE DEBUG	Deshabilita el procesamiento de ON DEBUG
DISABLE INTERRUPT	Deshabilita el procesamiento de ON INTERRUPT
DTMFOUT	Produce tonos telefónicos en un pin
EEPROM	Define el contenido inicial en un chip EEPROM
ENABLE	Habilita el procesamiento de ON INTERRUPT, ON DEBUG
ENABLE DEBUG	Habilita el procesamiento de ON DEBUG
ENABLE INTERRUPT	Habilita el procesamiento de ON INTERRUPT
END	Detiene la ejecución e ingresa en modo de baja potencia
FOR...NEXT	Ejecuta declaraciones en forma repetitiva
FREQOUT	Produce hasta 2 frecuencias en un pin
GOSUB	Llama a una subrutina BASIC en la línea especificada
GOTO	Continúa la ejecución en la línea especificada
HIGH	Saca un 1 lógico (5 V.) por un pin
HPWM	Salida de hardware con ancho de pulsos modulados
HSERIN	Entrada serial asincrónica (hardware)
HSEROUT	Salida serial asincrónica (hardware)
I2CREAD	Lee bytes de dispositivos I2C
I2CWRITE	Graba bytes de dispositivos I2C

DECLARACIONES	APLICACIÓN
IF..THEN..ELSE..ENDIF	Ejecuta declaraciones en forma condicional
INPUT	Convierte un pin en entrada
LCDIN	Lee caracteres desde una RAM de un LCD
LCDOUT	Muestra caracteres en un LCD
LET	Asigna el resultado de una expresión a una variable
LOOKDOWN	Busca un valor en una tabla de constantes
LOOKDOWN2	Busca un valor en una tabla de constantes o variables
LOOKUP	Obtiene un valor constante de una tabla
LOOKUP2	Obtiene un valor constante o variable de una tabla
LOW	Hace 0 lógico (0 V.) un pin específico
NAP	Apaga el procesador por un corto período de tiempo
ON DEBUG	Ejecuta un Debug en BASIC
ON INTERRUPT	Ejecuta una subrutina BASIC en un interrupt
OUTPUT	Convierte un pin en salida
OWIN	Entrada de dispositivos one-wire
OWOUT	Salida a dispositivos one-wire
PAUSE	Demora con resolución de 1 milisegundo (mS.)
PAUSEUS	Demora con resolución de 1 microsegundo (uS.)
PEEK	Lee un byte del registro
POKE	Graba un byte en el registro
POKE	Graba un byte en el registro
POT	Lee el potenciómetro en el pin especificado
PULSIN	Mide el ancho de pulso en un pin
PULSOUT	Genera pulso hacia un pin
PWM	Salida modulada en ancho de pulso por un pin especificado
RANDOM	Genera número pseudo-aleatorio
RCTIME	Mide el ancho de pulso en un pin
READ	Lee byte de un chip EEPROM
READCODE	Lee palabra desde un código de memoria
RESUME	Continúa la ejecución después de una interrupción
RETURN	Continúa en la declaración que sigue al último GOSUB
REVERSE	Convierte un pin de salida en entrada, o uno de entrada en salida
SELECT CASE	Compara una variable con diferentes valores
SERIN	Entrada serial asincrónica (tipo BASIC Stamp 1)
SERIN2	Entrada serial asincrónica (tipo BASIC Stamp 2)

DECLARACIONES	APLICACIÓN
SEROUT	Salida serial asincrónica (tipo BS1)
SEROUT2	Salida serial asincrónica (tipo BS2)
SHIFTIN	Entrada serial sincrónica
SHIFTOUT	Salida serial sincrónica
SLEEP	Apaga el procesador por un período de tiempo
SOUND	Genera un tono o ruido blanco en un pin
STOP	Detiene la ejecución del programa
SWAP	Intercambia los valores de dos variables
TOGGLE	Hace salida a un pin y cambia su estado
USBIN	Entrada de USB
USBINIT	Inicializar USB
USBOUT	Salida de USB
WHILE...WEND	Ejecuta declaraciones mientras la condición sea cierta
WRITE	Graba bytes en un chip EEPROM
WRITECODE	Escribe palabra en código de memoria
XIN	Entrada X - 10
XOUT	Salida X - 10

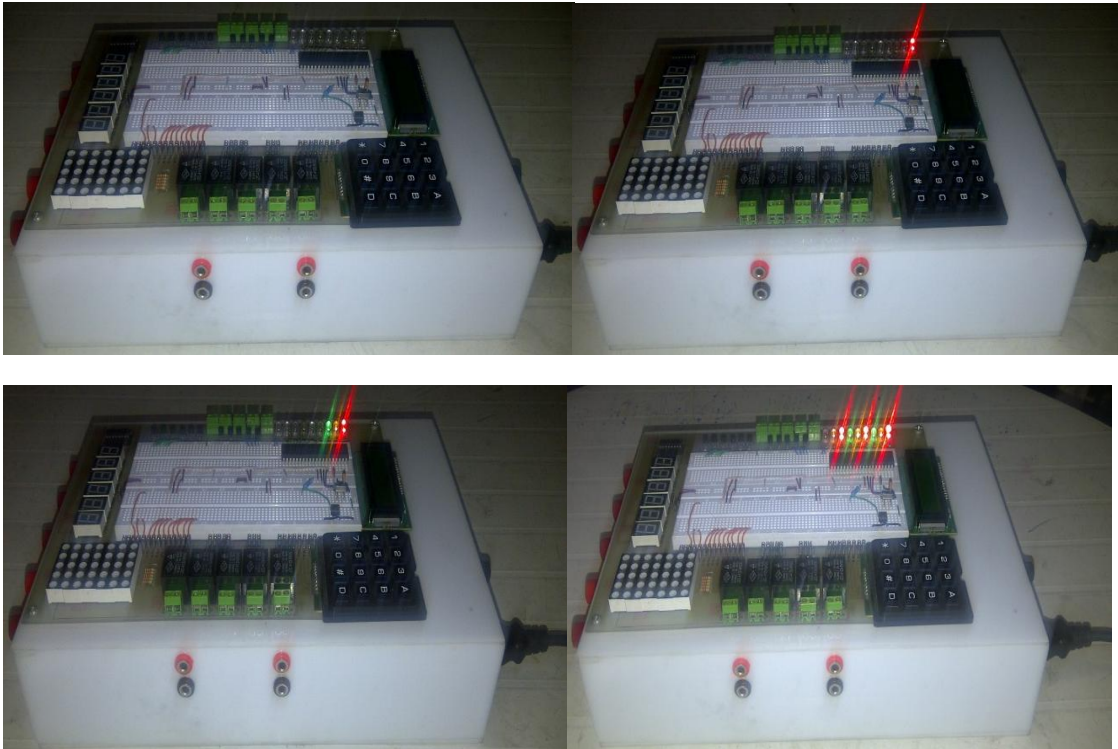
ANEXO C

FOTOS DE LAS PRÁCTICAS IMPLEMENTADAS

PRÁCTICA 1:



PRÁCTICA 2:



PRÁCTICA 3:



PRÁCTICA 4:

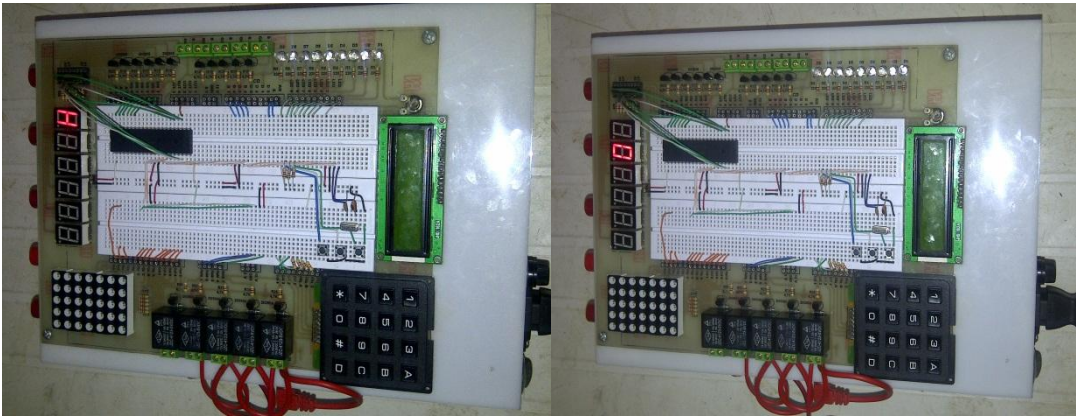


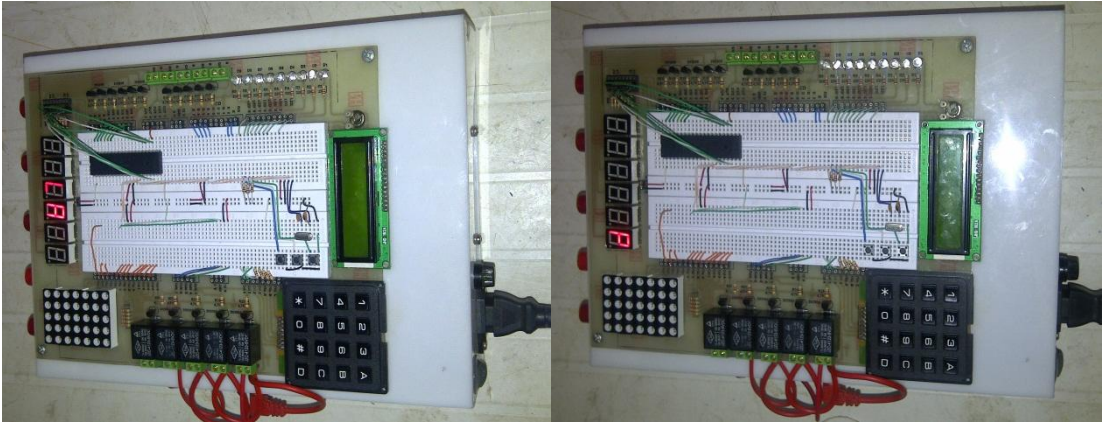


PRÁCTICA 5:

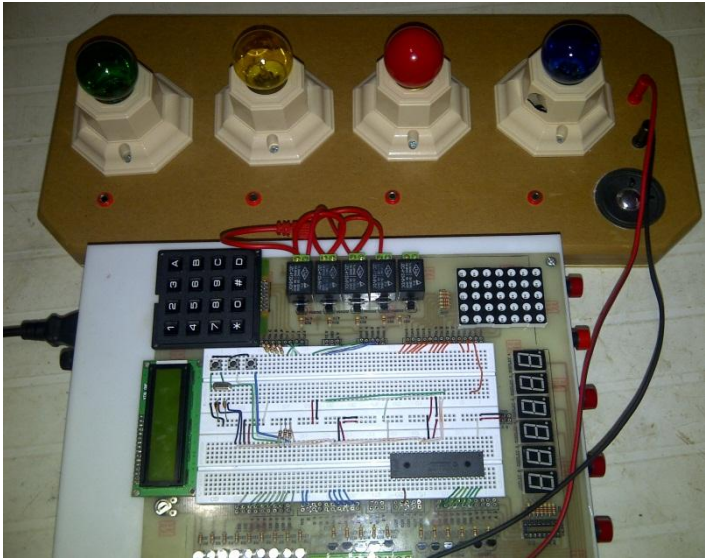


PRÁCTICA 6:

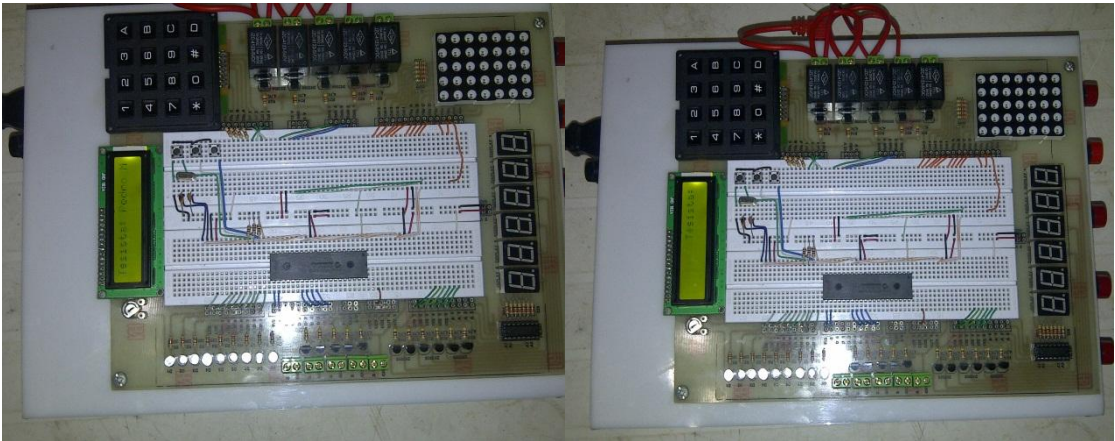




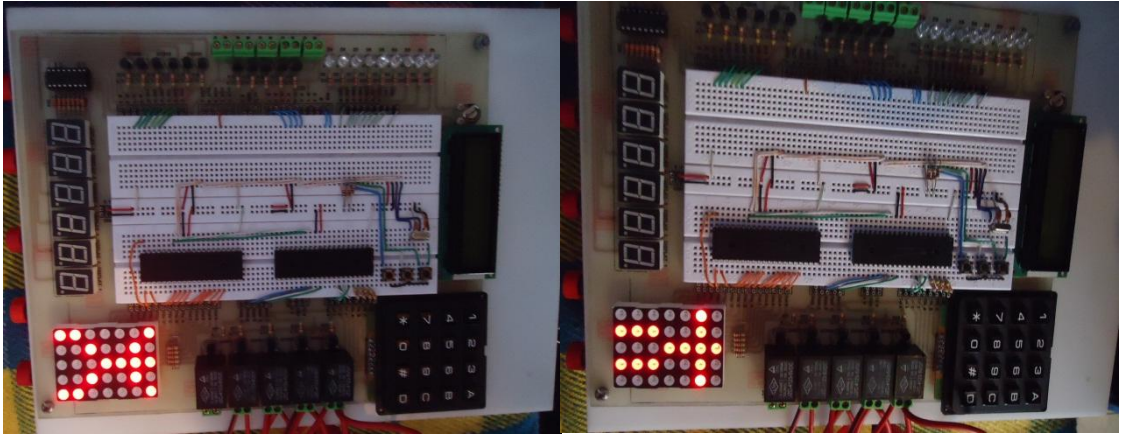
PRÁCTICA 7:



PRÁCTICA 8:



PRÁCTICA 9:



PRÁCTICA 10:

