



UNIVERSIDAD NACIONAL DE LOJA

INDUSTRIAS Y LOS RECURSOS NATURALES NO RENOVABLES

CARRERA DE TECNOLOGÍA EN ELECTRICIDAD Y CONTROL INDUSTRIAL

TEMA:

*“LABORATORIO PARA ENTRENAMIENTO DE
MICROCONTROLADORES”*

INFORME TÉCNICO PREVIA A LA
OBTENCIÓN DEL TÍTULO DE
TECNÓLOGO EN ELECTRICIDAD Y
CONTROL INDUSTRIAL

AUTOR:

MÁXIMO XAVIER SALINAS POMA

DIRECTOR:

ING. RODOLFO PABEL MERINO VIVANCO

LOJA- ECUADOR

2014

CERTIFICACIÓN

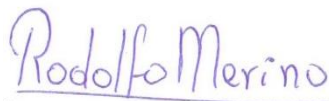
Ing. Rodolfo Pabel Merino Vivanco

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 “**LABORATORIO PARA ENTRENAMIENTO DE MICROCONTROLADORES**” desarrollado por el señor Máximo Xavier Salinas Poma, previo a optar el título 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, Enero del 2014



Ing. Rodolfo Pabel Merino Vivanco

DIRECTOR DEL INFORME TÉCNICO.

AUTORÍA

Yo **Máximo Xavier Salinas Poma** declaro ser autor del presente trabajo de tesis y eximo expresamente a la Universidad Nacional de Loja y a sus representantes jurídicos de posibles reclamos o acciones legales, por el contenido de la misma.

Adicionalmente acepto y autorizo a la Universidad Nacional de Loja, la publicación de mi tesis en el Repositorio Institucional-Biblioteca Virtual.

Autor: Máximo Xavier Salinas Poma



Firma:

Cédula: 1104884869

Fecha: 17-01-2014

CARTA DE AUTORIZACIÓN DE TESIS POR PARTE DEL AUTOR, PARA LA CONSULTA, RPRODUCCIÓN PARCIOAL O TOTAL, Y PUBLICACIÓN ELECTRÓNICA DEL TEXTO COMPLETO.

Yo **Máximo Xavier Salinas Poma** declaro ser autor de la tesis titulada: **“LABORATORIO PARA ENTRENAMIENTO DE MICROCONTROLADORES”**, como requisito para optar al grado de: **Tecnólogo en Electricidad y Control Industrial**; autorizo al Sistema Bibliotecario de la Universidad Nacional de Loja para que con fines académicos, muestre al mundo la producción intelectual de la Universidad, a través de la visibilidad de su contenido de la siguiente manera en el Repositorio Digital Institucional.

Los usuarios pueden consultar el contenido de este trabajo en el RDI, en las redes de información del país y del exterior, con las cuales tenga convenio la Universidad.

La Universidad Nacional de Loja, no se responsabiliza por el plagio o copia de la tesis que realice un tercero.

Para constancia de esta autorización, en la ciudad de Loja, a los 17 días del mes de enero del dos mil catorce, firma el autor.



Firma:.....

Autor: Máximo Xavier Salinas Poma

Cédula: 1104884869

Dirección: Esteban Godoy **Correo Electrónico:** maxjavi2@hotmail.com

Teléfono: 2545285 **Celular:** 0989090675

DATOS COMPLEMENTARIOS

Director de Tesis: Ing. Rodolfo Pabel Merino Vivanco

Tribunal de Grado: Ing. Ramiro Marcelo Borrero Espinosa

Ing. Norman Augusto Jiménez León Mg. Sc.

Ing. Edwin Bladimir Paccha Herrera, Mg. Sc.

DEDICATORÍA

Este trabajo se lo dedico a Dios por iluminarme siempre en mi vida y que me ha dado fortaleza para culminar mi tesis, a mis padres, hermanos por su amor, sus consejos y su apoyo incondicional en mis estudios, siendo para todos ellos dedicado la presente.

Máximo X. Salinas P.

AGRADECIMIENTO

Agradezco a la Universidad Nacional de Loja, a los docentes de la carrera de Tecnología Eléctrica y Control Industrial, por las enseñanzas impartidas en mi vida universitaria.

Así como también de manera especial al Ing. Rodolfo Pabel Merino Vivanco, por el apoyo y la guía continúa brindada durante la ejecución y desarrollo del presente trabajo investigativo.

A mis padres, hermanos y demás familiares que siempre estuvieron apoyándome, dándome palabras de aliento y confianza, y a todas las personas que colaboraron de una u otra manera para la culminación de este trabajo.

Máximo X. Salinas P.

ÍNDICE GENERAL

PORTADA.....	I
CERTIFICACIÓN.....	II
AUTORÍA	III
CARTA DE AUTORIZACIÓN.....	IV
DEDICATORIA	V
AGRADECIMIENTO	VI
ÍNDICE GENERAL	VII
RESUMEN	X
SUMMARY	XI
I. INTRODUCCIÓN.....	2
II. DESCRIPCIÓN TÉCNICA.....	4
2.1. CONTROLADORES Y MICROCONTROLADORES	5
2.1.1. Controlador.....	5
2.1.2. Microcontrolador	6
2.1.3. Ventajas de un Microcontrolador frente a un Microprocesador	8
2.1.4. Tipos de Arquitecturas de Microcontroladores	9
2.1.5. Arquitectura Von Neumann.....	9
2.1.6. Arquitectura Harvard.....	10
2.1.7. Características Relevantes de los PIC	12
2.1.8. Arquitectura del PIC16F628A	13
2.1.9. La memoria de programa.....	15
2.1.10. La memoria de datos	17
2.1.11. Características Generales	19
2.1.12. Diagrama de pines y funciones.....	20
2.2. DISPOSITIVOS ELECTRÓNICOS	21

2.2.1. Capacitores	21
2.2.2. Resistores.....	24
2.2.3. Relé, Relay - Relevador.....	27
2.2.4. Pulsador	28
2.2.5. Displays de 7 Segmentos	30
2.2.6. Módulo LCD.....	32
2.2.7. Matriz de Leds	34
2.3. SOFTWARE MICROCODE STUDIO	36
2.3.1. Características Microcode Studio	36
2.3.2. PIC que Soporta Microcode Studio	37
2.3.3. Configuraciones de Hardware.....	38
2.3.4. Configuración de Microcode Studio (IDE)	39
2.4. FUENTE DE ALIMENTACIÓN DEL MÓDULO DE PRÁCTICAS	42
2.4.1. Estructura Básica.....	43
2.4.2. Conexión Red Eléctrica.	43
2.4.3. Fusible... ..	44
2.4.4. Filtro de Red	44
2.4.5. Transformador	45
2.4.6. Rectificador.....	49
2.4.7. Filtro	52
2.4.8. Regulador de Tensión.....	53
III. MATERIALES	59
3.1. MATERIALES DEL MÓDULO.....	60
IV. PROCESO METODOLÓGICO	62
4.1. MÓDULO DE PRÁCTICAS PARA MICROCONTROLADORES	63
4.1.1. Diseño PCB de la Fuente de Alimentación	63
4.1.2. Diseño del Módulo de Prácticas.....	64

4.1.3. Diseño PCB del Módulo de Prácticas	65
V. RESULTADOS	68
5.1. PRÁCTICAS PROPUESTAS PARA EL MODULO	69
PRÁCTICA 1	70
PRÁCTICA 2	73
PRÁCTICA 3	78
PRÁCTICA 4	84
PRÁCTICA 5	87
PRÁCTICA 6	93
PRÁCTICA 7	99
PRÁCTICA 8	105
PRÁCTICA 9	110
VI. CONCLUSIONES Y RECOMENDACIONES	115
6.1. CONCLUSIONES.....	116
6.2. RECOMENDACIONES.....	117
VII. BIBLIOGRAFÍA	118
8.1. BIBLIOGRAFÍA.....	119
VIII. ANEXOS	121
ANEXO A	122
ANEXO B	125
ANTEPROYECTO.....	129

RESUMEN

El presente trabajo de investigación muestra el diseño e implementación de un laboratorio para entrenamiento de microcontroladores, que está formado por distintos componentes electrónicos con el fin de proporcionar diferentes funciones dependiendo de la programación que se realice en el microcontrolador.

Para una formación profesional en este campo, se debe contar con un equipo que nos permita preparar adecuadamente a los estudiantes, es por eso que se creyó conveniente la implementación de un laboratorio donde consten diferentes elementos electrónicos, para realización de prácticas, de esta forma se logrará mejorar sus conocimientos tales como:

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 16F628A, u otro de la misma familia.

SUMMARY

The following Research Project has to deal with the design and development of a Laboratory for Microcontrollers Training, which is formed by some electronic components, in order to provide different functions, according to the programming made by the microcontroller.

For professional training in this field, equipment should be provided that enables us to adequately prepare students. That is why I believe that it would be a good idea to create a laboratory that provides hands-on training in various elements of electronics, thus improving their skills in:

Designing programming algorithms.

Programming in MicroCode Studio.

Using the interface between PROTEUS ISIS and MicroCode Studio.

Reading datasheets.

Thus, students would have the ability to perform any type of project proposed that uses the PIC 16F628A microcontroller, or a related device.

*“LABORATORIO PARA ENTRENAMIENTO DE
MICROCONTROLADORES”*

1. INTRODUCCIÓN

La industria electrónica evoluciona constantemente en la aplicación de nuevas tecnologías para conseguir los múltiples procesos que en ella se realizan, incorporando automatismos cada vez más sofisticados. Creando la necesidad de diseñar sistemas con mejores características, de menor tamaño, bajos requerimientos de energía, mejor realización, teniendo un especial énfasis sobre todo en la facilidad del sistema diseñado.

Uno de los sistemas utilizados actualmente es el microcontrolador¹ que se lo encuentra en aplicaciones tales como: control de alarmas, tableros de control en la industria automotriz, en la instrumentación médica, en los teclados de computadora, sistemas portátiles de almacenamiento de datos, en equipos de laboratorio, entre otros.

Un microcontrolador es todo un sistema mínimo dentro de un solo dispositivo lo cual ofrece enormes beneficios tecnológicos. En estos circuitos se puede programar en un solo "chip", para desarrollar e implementar varias aplicaciones y funciones con menos componentes que la circueteria tradicional de compuertas logicas, optimizando mucho espacio físico, 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

¹ Microcontrolador: son computadores digitales integrados en un chip que cuentan con un microprocesador o unidad de procesamiento central (CPU), una memoria para almacenar el programa, una memoria para almacenar datos y puertos de entrada salida.

MICROCONTROLADORES PIC: LA SOLUCION ESTA EN UN CHIP, THOMSON PARANINFO, 2001

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.

A medida que va pasando el tiempo, se ha visto la necesidad de conocer cada vez más, sobre alguna de las tantas aplicaciones de la electrónica, debido al uso en sus diferentes tipos de servicios que esta presta.

Para una formación profesional en este campo, se debe contar con un equipo que nos permita preparar adecuadamente a los estudiantes, es por eso que se creyó conveniente la implementacion de un laboratorio para la Universidad Nacional de Loja, en la Carrera de Electrónica, donde consten diferentes elementos electrónicos, para realización de prácticas, de esta forma se logrará mejorar sus conocimientos tales como:

Diseñar algoritmos de programación.

Programar en MicroCode Studio.

Utilización de la interfaz entre ISIS de PROTEUS y MicroCode Studio.

Lectura de Datasheets.

Permitiendoles de esta manera estar en la capacidad de realizar cualquier tipo de proyecto que se propongan con la utilización del microcontrolador PIC 16F628A, u otro de la misma familia.

DESCRIPCIÓN TÉCNICA

El desarrollo de la electrónica, se ha constituido en uno de los más grandes sucesos de la época moderna. El Microcontrolador, uno de los dispositivos electrónicos programables más sofisticados, es y será fundamental para los grandes adelantos tecnológicos, en casi todos los campos del saber.

Teniendo en cuenta, que el conocimiento es dinámico, y avanza paralelamente con el desarrollo de la ciencia y la tecnología, hoy día, es supremamente importante, no limitarse en el desarrollo, implementación y aplicación de una tecnología, sino contar con los elementos y herramientas necesarias para adaptarse al aprendizaje y manejo de los recursos computacionales y técnicos que están disponibles.

2.1 CONTROLADORES Y MICROCONTROLADORES²

2.1.1 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

²“CONTROLADORES Y MICROCONTROLADORES” (en línea) Autor: Luis E. Sánchez Toledo (Universidad Católica del Uruguay), 2005, <http://www.monografias.com/trabajos34/microcontroladores-genericos/microcontroladores-genericos.shtml>

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

2.1.2 Microcontrolador

Es un dispositivo electrónico capaz de llevar a cabo procesos lógicos, ya que en su interior posee toda la arquitectura de un computador, esto es CPU, memorias RAM, EEPROM, y circuitos de entrada y salida. Un microcontrolador de fábrica, no realiza tarea alguna, este debe ser programado para que realice desde un simple parpadeo de un led hasta un sofisticado control de un robot. Un microcontrolador es capaz de realizar la tarea de muchos circuitos lógicos como compuertas AND, OR, NOT, NAND, conversores A/D, D/A, temporizadores, decodificadores, etc., simplificando todo el diseño a una placa de reducido tamaño y pocos elementos. 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 serían mucho más pequeños y simplificados.

La diferencia entre Microprocesador y Microcontrolador es la siguiente:

Con un microprocesador se puede observar en la **Figura 1**, 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.

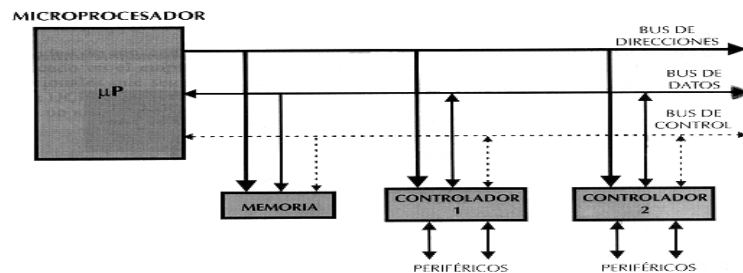


Figura 1. Estructura de un sistema abierto basado en un microprocesador

Fuente: "CONTROLADORES Y MICROCONTROLADORES" (en línea) Autor: Luis E. Sánchez Toledo (Universidad Católica del Uruguay), 2005, <http://www.monografias.com/trabajos34/microcontroladores-genericos/microcontroladores-genericos.shtml>

En cambio 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.

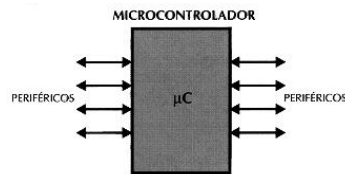


Figura 2. El Microcontrolador es un sistema cerrado

Fuente: "CONTROLADORES Y MICROCONTROLADORES" (en línea) Autor: Luis E. Sánchez Toledo (Universidad Católica del Uruguay), 2005, <http://www.monografias.com/trabajos34/microcontroladores-genericos/microcontroladores-genericos.shtml>

2.1.3 Ventajas de un Microcontrolador frente a un Microprocesador.³

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

³ "Ventajas de un Microcontrolador frente a un Microprocesador" (en línea) Autor: José David Verbel Tous, 2005, <http://www.monografias.com/trabajos27/microcontroladores/microcontroladores.shtml>

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 Tipos de Arquitecturas de Microcontroladores

2.1.5 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.

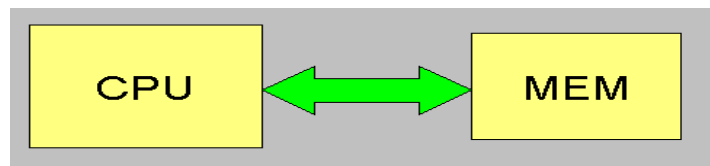


Figura 3. *Arquitectura Von Neumann*

Fuente: "Ventajas de un Microcontrolador frente a un Microprocesador" (en línea) Autor: José David Verbel Tous, 2005, <http://www.monografias.com/trabajos27/microcontroladores/microcontroladores.shtml>

2.1.6 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 encontraran físicamente en la memoria de programa (por ejemplo en la EPROM de un microprocesador).

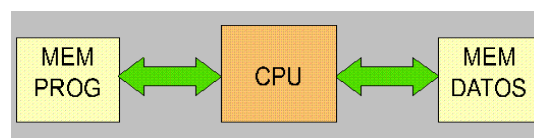


Figura 4. *Arquitectura Harvard*

2.1.7 Características Relevantes de los PIC ⁴

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.

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

PIC 16F628A

PIC 16F627A

PIC 16F648A

PIC 16F877A

⁴“Microcontroladores PIC diseño práctico de aplicaciones” (en línea) Ed. Mac Graw Gil. Autores: José Ma Angulo Usategui / Ignacio Angulo Martínez , <http://perso.wanadoo.es/pictob/micropic.htm>

Estos microcontroladores soportan hasta 100.000 ciclos de escritura en su memoria FLASH, y 1'000.000 ciclos en su memoria Eeprom, este está reemplazando rápidamente al popular PIC16F84A, pues presenta grandes ventajas como son:

	PIC16F84A	PIC16F627A	PIC16F628A	PIC16F648A
Memoria de programa Flash	1024 x 14	1024 x 14	2048 x 14	4096 x 14
Memoria datos RAM	68 x 8	224 x 8	128 x 8	256 x 8
Memoria datos EEPROM	64 x 8	128 x 8	128 x 8	256 x 8
Pines de entrada salida	13	16	16	16
Comparadores de voltaje	0	2	2	2
Interrupciones	4	10	10	10
Timers 8 16 bits	1	3	3	3
Modulos PWM CCP	No	Si	Si	Si
Comunicación serial USART	No	Si	Si	Si

Tabla 1. Tabla de comparación entre el PIC16F84A y los PIC16F6XX.

Fuente: “Microcontroladores PIC diseño práctico de aplicaciones” (en línea) Ed. Mac Graw Gil. Autores: José Ma Angulo Usategui / Ignacio Angulo Martínez, <http://perso.wanadoo.es/pictob/micropic.htm>

2.1.8 Arquitectura del PIC16F628A ⁵

El PIC16F628A utiliza un procesador con arquitectura Harvard, consiguiendo mayor rendimiento en el procesamiento de las instrucciones, esta arquitectura a diferencia de la Von Neumann, utiliza dos bloques de memoria independientes, una contiene instrucciones y la otra solo datos, cada una con su respectivo sistema de buses de acceso, 8 líneas para los datos y 14 líneas para las instrucciones, con lo que es posible realizar operaciones de acceso lectura o escritura simultáneamente en las 2 memorias, a esto se conoce como paralelismo.

⁵ Microcontroladores PIC Programación En BASIC, Carlos A. Reyes (TERCERA EDICIÓN), Pág. 18.

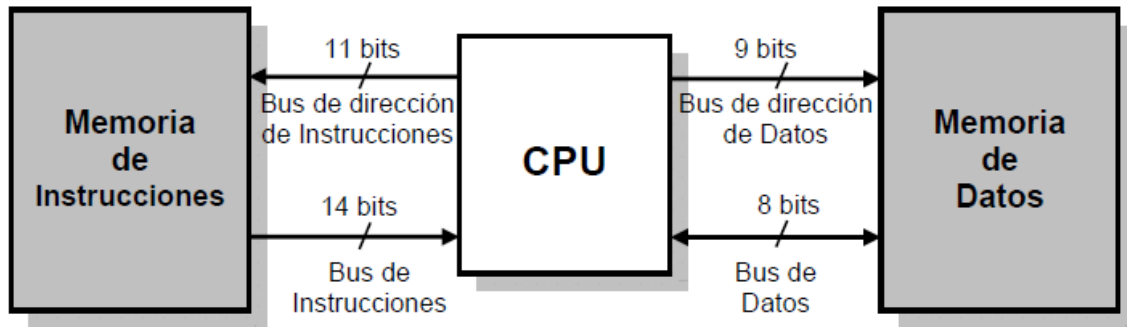


Figura 5. La arquitectura Harvard maneja la memoria de datos y la memoria de instrucciones por separado y con diferentes capacidades.

Fuente: Microcontroladores PIC Programación En BASIC, Carlos A. Reyes (TERCERA EDICIÓN), Pág.18

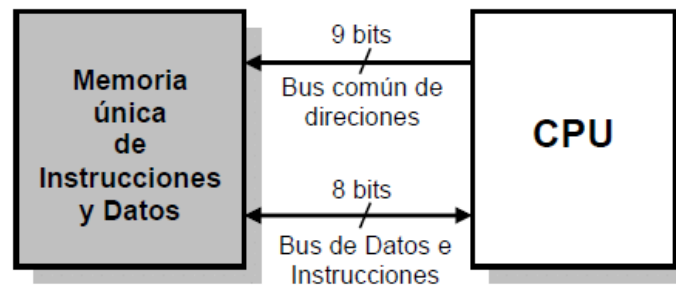


Figura 6. En la arquitectura Von Neumann se conecta el CPU con una memoria única en donde se almacenan datos e instrucciones en forma indistinta, compartiendo el mismo bus.

Fuente: Microcontroladores PIC Programación En BASIC, Carlos A. Reyes (TERCERA EDICIÓN), Pág.18

El CPU del microcontrolador 16F628A emplea una avanzada arquitectura RISC (computador con juego de instrucciones reducido) con un set de 35 instrucciones poderosas pertenecientes a la gama media de la familia de los microcontroladores PIC, la mayoría de instrucciones se ejecutan en un ciclo de instrucción a excepción de los saltos que requieren de 2 ciclos, dentro de su procesador existe una pila de 8 niveles que permite el anidamiento de subrutinas, esto quiere decir que puede retomar 8 lugares diferentes de línea de programa e ir regresando a cada uno en el orden inverso al que fueron anidados

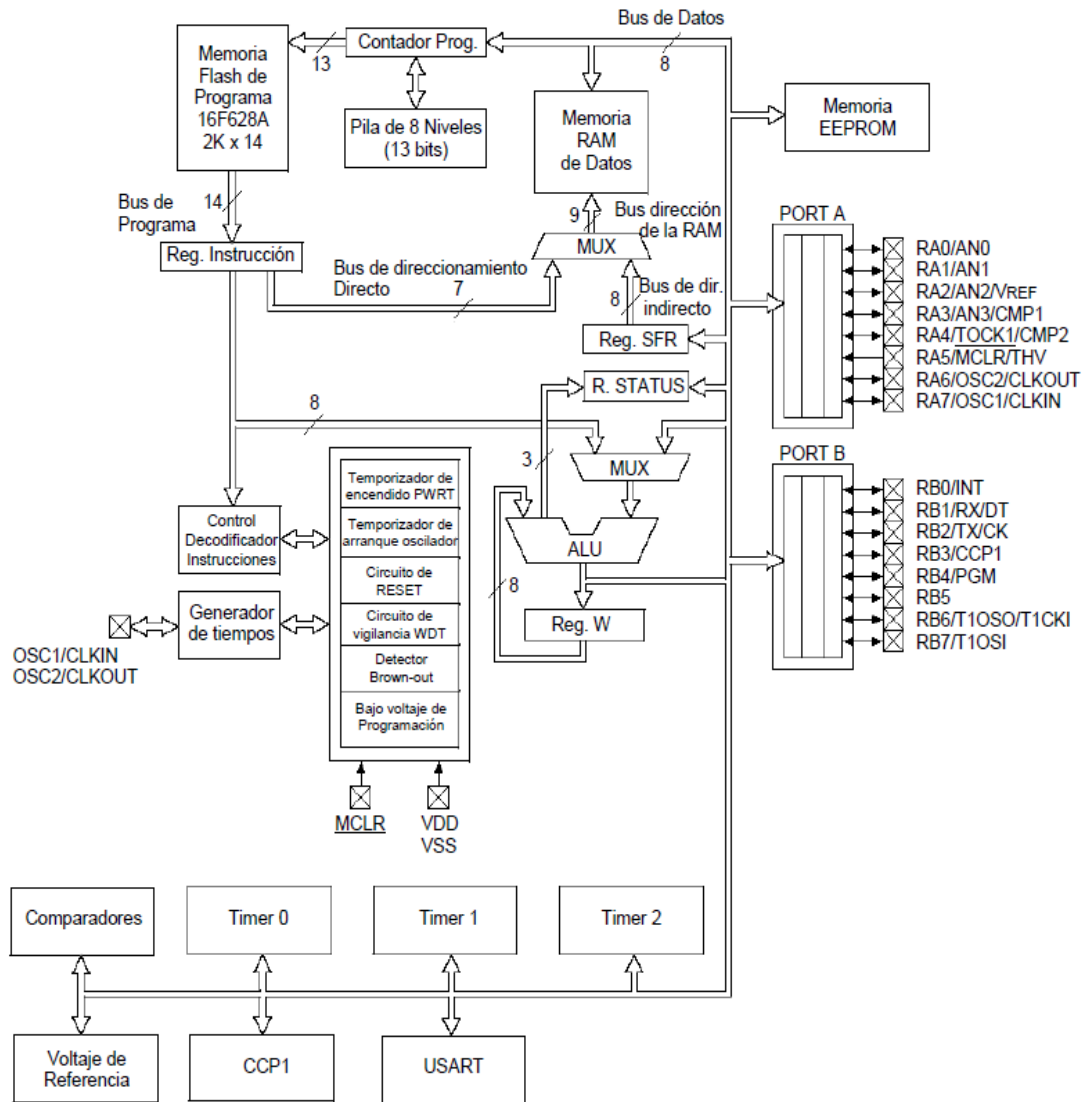


Figura 7. Diagrama de los bloques funcionales del PIC, su conexión interna es mediante buses, se aprecia la conexión de las 3 memorias Flash, Ram y Eeprom.

Fuente: Microcontroladores PIC Programación En BASIC, Carlos A. Reyes (TERCERA EDICIÓN), Pág.19

2.1.9 La memoria de programa

Conocido también como memoria de instrucciones, aquí se escribe las órdenes para que el CPU las ejecute. En el caso del microcontrolador PIC16F628A tiene memoria de programa no volátil tipo FLASH, en comparación a su antecesor la memoria EEPROM, este se caracteriza por

ser más rápido en el proceso de escritura/borrado eléctrico, además dispone de mayor capacidad de almacenamiento, esta característica hace que sea ideal para prácticas de laboratorio en donde la grabación y el borrado son frecuentes (recuerde que soporta 100.00 ciclos de escritura/borrado).

El bus de direcciones de la memoria de programa es de 13bits, por lo que el contador de programa (PC) puede direccionar 8192 posiciones de 14bits cada una (desde la 0000h hasta 1FFFh), de las cuales solo las primeras 2048 líneas tienen implementadas (desde 0000h hasta la 07FFh), es decir que el PC solo utiliza los 11 primeros bits de direcciones los demás bits los ignora.

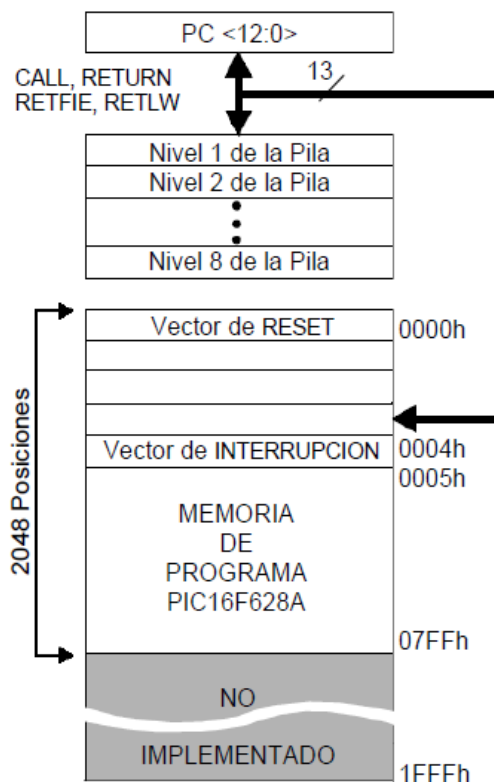


Figura 8. Mapa de la memoria de programa de las 8196 posiciones, solo tiene implementado las primeras 2048 posiciones, la dirección 0000h está reservada para el vector de reset y las 0004h está reservada para el vector de interrupción.

2.1.10 La memoria de datos ⁶

El PIC16F628A, tiene dos tipos de memorias de datos, la RAM estática o SRAM o memoria de acceso casual que es un tipo de memoria volátil, es decir sus datos permanecen en la memoria mientras esta alimentación en el dispositivo y es de vital importancia porque ahí residen dos tipos de datos, los registrados de propósito general, en donde se almacenan las variables y los registros especiales, que son los encargados de llevar al contador de programa, el conteo del Temporizador, el estado de los puertos, la configuración de las interrupciones, etc.

El otro tipo de memoria es una memoria auxiliar no volátil llamada EEPROM, con capacidad de 128 posiciones de 8bits cada una. Esta memoria puede ser accedida por el usuario mediante programación, es muy útil para almacenar datos que el usuario necesita que se conserven aun sin alimentación, tal es el caso de la clave de una alarma, esta puede ser modificada, pero no debe perderse por un corte de energía.

Como este microcontrolador es fabricado con tecnología CMOS, su consumo de potencia es muy bajo (2mA a 4Mhz) y además es completamente estático, lo que significa que si el reloj se detiene los datos de la memoria RAM no se pierden, esto mientras el microprocesador siga alimentado. La memoria de datos RAM, tiene 512 líneas de 8 bits cada una de estas será particionada por 4 bancos; el banco 0, banco 1, banco 2 y

⁶ Microcontroladores PIC Programación En BASIC, Carlos A. Reyes (TERCERA EDICIÓN), Pág. 20

2.1.11 Característica Generales ⁷

Características más relevantes del PIC16F628A:

- Velocidad de operación hasta 20MHZ con oscilador externo.
- Oscilador interno RC (resistencia condensador) de 4MHZ calibrado de fábrica al $\pm 1\%$.
- Admite 8 configuraciones de oscilador.
- 8 niveles de pila.
- Procesador con arquitectura HARVARD.
- Conjunto reducido de instrucciones RISC (35) gama media.
- Instrucciones de un ciclo excepto los saltos (200nSpor instrucción a 20 MHZ).
- Resistencias PULL-UP programables en el puerto B.
- Pin RA5 MCLR programable como reset externo o pin de entrada.
- Rango de operación desde 3V hasta 5.5V.
- 15 pines de I/O y 1 solo de entrada (RA5).
- Temporizador perro guardián WDT independiente del oscilador.
- Programable con bajo voltaje LPV (5V).
- Programación serial en Circuito ICSP por 2 pines: RB6 reloj y RB7 datos.
- Código de protección programable por sectores.
- Memoria de programa FLASH 2048K. de 100.000 ciclos escritura/borrado.

⁷ Microcontroladores PIC Programación En BASIC, Carlos A. Reyes (TERCERA EDICIÓN), Pág.22

- Memoria de datos EEPROM de 1.000.000 ciclos escritura/borrado de 100 años retención.
- 2 circuitos comparadores análogos con entradas multiplexadas.
- Timers, Timer 0 a 8 bits, Timer 1 a 16 bits y Timer 2 a 8 bits.
- Módulo CCP, Captura compara 16 bits y PWM, modulación de ancho de pulso 10 bits.
- 10 fuentes de interrupción.
- Módulo de comunicación serial USART/SCI.
- Capacidad de corriente para encender leds directamente (25 mA I/O)) por cada pin.

2.1.12 Diagrama de pines y Funciones

Excluyendo los 2 pines de alimentación, todos los 16 pines restantes pueden ser configurados como entradas o salidas, algunos de ellos tienen funciones especiales.

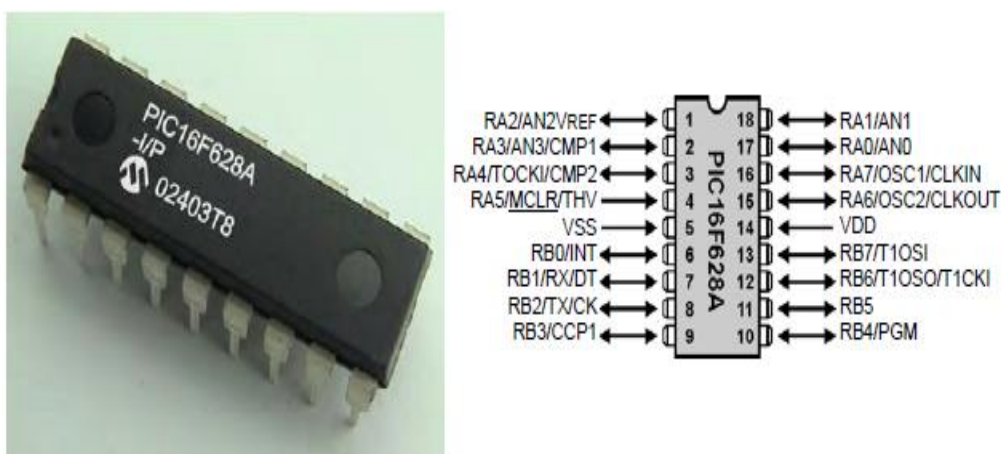


Figura 10. PIC16F628A y su diagrama de pines

Fuente: Microcontroladores PIC Programación En BASIC, Carlos A. Reyes (TERCERA EDICIÓN), Pág.22

PIN	NOMBRE	DESCRIPCION
17	RA0/AN0	Pin bidireccional I/O, entrada comparador análogo.
18	RA1/AN1	Pin bidireccional I/O, entrada comparador análogo.
1	RA2/AN2/VREF	Pin bidireccional I/O, entrada comp. Análogo y Voltaje de referencia
2	RA3/AN3/CMP1	Pin I/O, entrada comp. Análogo y salida del comparador análogo 1.
3	RA4/T0CKI/CMP2	Pin I/O, entrada reloj TIMER0 y salida del comparador análogo 2.
4	RA5/MCLR/VPP	Pin de entrada, en modo MCLR activa RESET externo.
15	RA6/OSC2/CLKIN	Pin I/O, entrada oscilador externo, salida de ¼ de la frecuencia OSC 1.
16	RA7/OSC1/CLKIN	Pin I/O, entrada oscilador externo, entrada del reloj externo.
6	RB0/INT	Pin I/O, resistencia Pull-Up programable, entrada de interrupción ext.
7	RB1/RX/DT	Pin I/O, resist. Pull-Up, entrada dato RS232, I/O dato serial asíncrono.
8	RB2/TX/CK	Pin I/O, resist. Pull-Up, salida dato RS232, I/O señal de reloj asíncrono.
9	RB3/CCP1	Pin I/O, resist. Pull-Up, módulo CCP/PWM entrada o salida.
10	RB4/PGM	Pin I/O, resist. Pull-Up, entrada del voltaje bajo de programación.
11	RB5	Pin I/O, resistencia Pull-Up programable.
12	RB6/T1OSO/T1CKI	Pin I/O, resist. Pull-Up, salida oscilador TIMER1, entrada reloj de ICSP
13	RB7/T1OSI	Pin I/O, resist. Pull-Up, entrada oscilador TIMER 1, I/O datos de ICSP.

Tabla 2. Tabla de pines con sus funciones especiales

NOTA: sus 2 puertos el A y el B entregan un total de 200mA cada uno, es decir 25mA cada pin. En modo sumidero puede soportar cada uno de sus puertos 200mA es decir 25mA cada pin.

Fuente: Microcontroladores PIC Programación En BASIC, Carlos A. Reyes (TERCERA EDICION), Pág.23

2.2. DISPOSITIVOS ELECTRÓNICOS.

2.2.1. Capacitores⁸

Los 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.

⁸ "Capacitores" (en línea) Autor: B. Montesinos (Universidad de La Frontera, de Temuco, Chile), <http://www.inele.ufro.cl/bmonteci/semic/apuntes/capacitores/capacitores.htm>

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 se va 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.

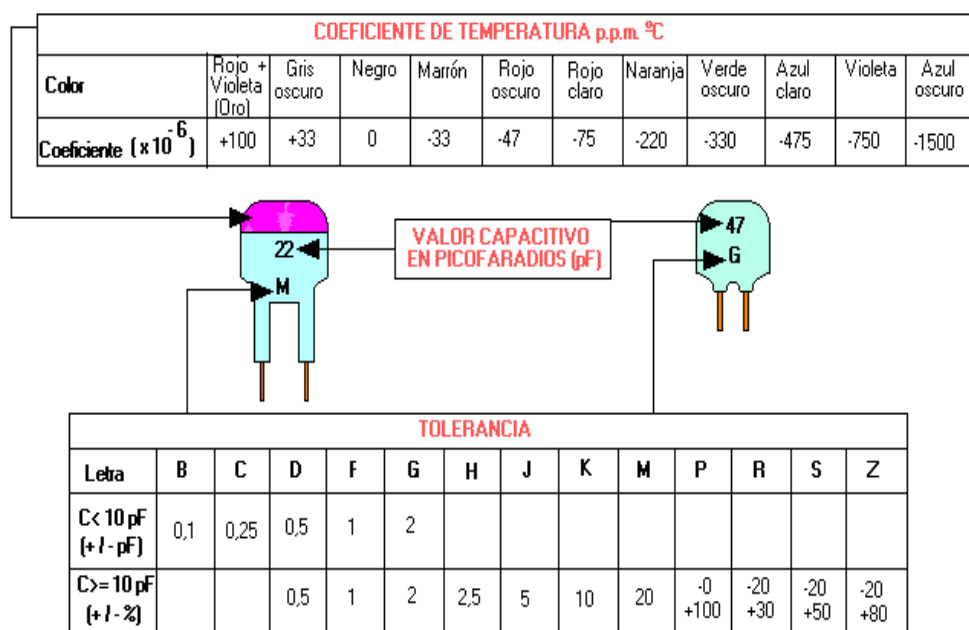


Figura 11. Capacitores cerámicos Grupo 1

Fuente: "Capacitores" (en línea) Autor: B. Montesinos (Universidad de La Frontera, de Temuco, Chile), <http://www.inele.ufro.cl/bmonteci/semic/apuntes/capacitores/capacitores.htm>

Código de colores

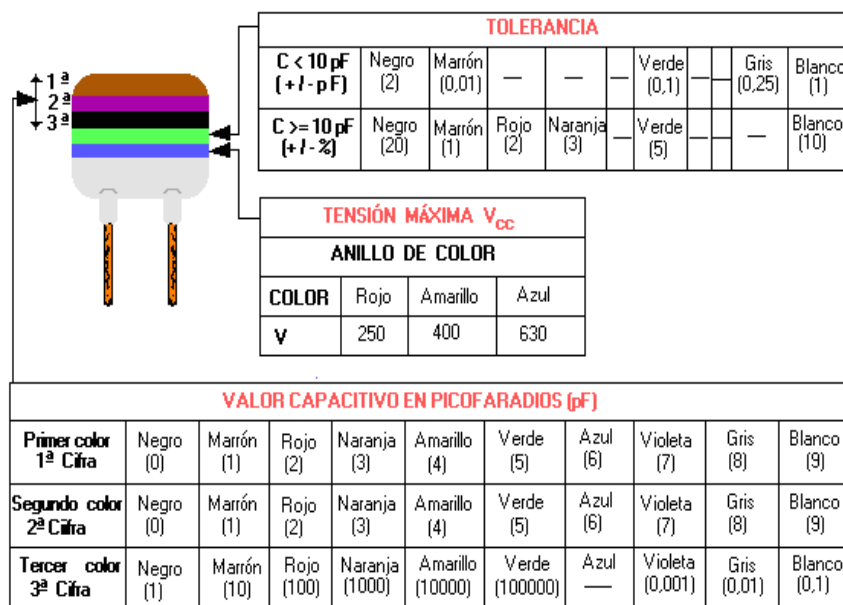


Figura 12. Capacitores cerámicos Grupo 2

Fuente: "Capacitores" (en línea) Autor: B. Montesinos (Universidad de La Frontera, de Temuco, Chile), <http://www.inele.ufro.cl/bmonteci/semic/apuntes/capacitores/capacitores.htm>

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:

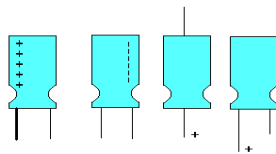


Figura 13. Capacitores electrolíticos

Fuente: "Capacitores" (en línea) Autor: B. Montesinos (Universidad de La Frontera, de Temuco, Chile), <http://www.inele.ufro.cl/bmonteci/semic/apuntes/capacitores/capacitores.htm>

2.2.2. Resistores

2.2.3. 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 fijas: su valor de resistencia es constante y está predeterminado por el fabricante.



Figura 14. Resistor fijo

Fuente: "Resistores" (en línea) Autor: B. Molina (Argentina)
<http://www.profesormolina.com.ar/electronica/componentes/resistores/resist.htm>

Resistores no lineales: su valor de resistencia varia 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 estos 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.).

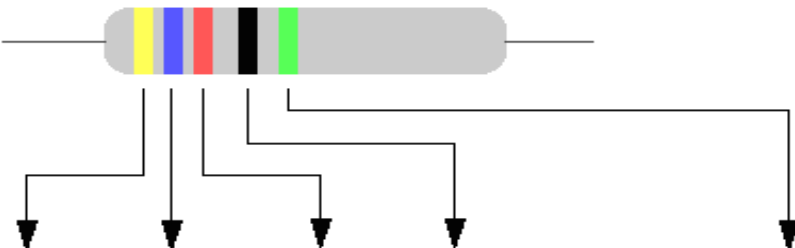
Identificación de resistores⁹

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.

⁹ "Resistores" (en línea) Autor: B. Molina (Argentina)
<http://www.profesormolina.com.ar/electronica/componentes/resistores/resist.htm>

Código de colores

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



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%

Figura 15. Código de colores para tres o cuatro bandas

Fuente: "Resistores" (en línea) Autor: B. Molina (Argentina)

<http://www.profesormolina.com.ar/electronica/componentes/resistores/resist.htm>

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. ¹⁰

2.2.4. Relé, Relay – Relevador ¹¹

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 atrae 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

¹⁰ "Resistores" (en línea) Autor: B. Molina (Argentina)
<http://www.profesormolina.com.ar/electronica/componentes/resistores/resist.htm>

¹¹ "Relé, Relay-Relevador" (en línea) Unicrom, 2002-2012,
http://www.unicrom.com/Tut_relay.asp

cosa conectada, cuando está inactivo. Es importante saber cuál 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 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: $I = V / R$.

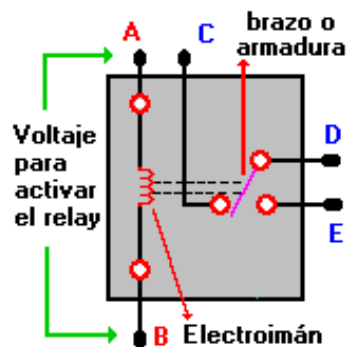


Figura 16. Estructura interna del relé

Fuente: "Relé, Relay-Relevador" (en línea) Unicrom, 2002-2012,
http://www.unicrom.com/Tut_relay.asp

2.2.5. Pulsador ¹²

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

¹² "Pulsador" (en línea) WIKIPEDIA, 2013, [http://es.wikipedia.org/wiki/Bot%C3%B3n_\(dispositivo\)](http://es.wikipedia.org/wiki/Bot%C3%B3n_(dispositivo))

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:

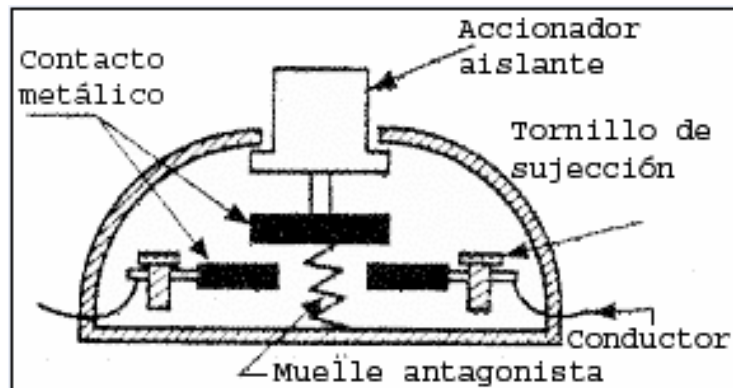


Figura 17. Estructura interna de un pulsador

Fuente: "Pulsador" (en línea) WIKIPEDIA, 2013,
[http://es.wikipedia.org/wiki/Bot%C3%B3n_\(dispositivo\)](http://es.wikipedia.org/wiki/Bot%C3%B3n_(dispositivo))



Figura 18. Pulsadores en el mercado

Fuente: "Pulsador" (en línea) WIKIPEDIA, 2013,
[http://es.wikipedia.org/wiki/Bot%C3%B3n_\(dispositivo\)](http://es.wikipedia.org/wiki/Bot%C3%B3n_(dispositivo))

2.2.6. Displays de 7 Segmentos ¹³

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 sí 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.

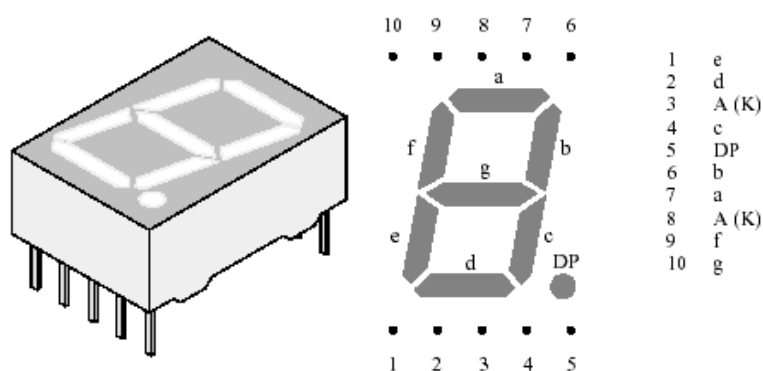


Figura 19. Denominación de los segmentos de Display

Fuente: "Displays de 7 Segmentos" (en línea) SCRIBD, 2013
<http://es.scribd.com/doc/5811293/DISPLAY-7-SEGMENTOS>

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

¹³"Displays de 7 Segmentos" (en línea) SCRIBD, 2013 <http://es.scribd.com/doc/5811293/DISPLAY-7-SEGMENTOS>

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).

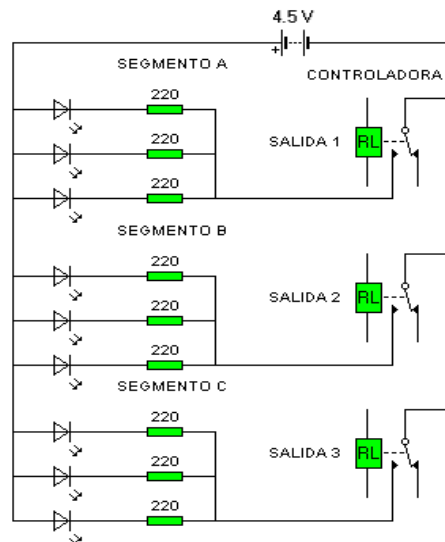


Figura 20. Denominación de los segmentos de Display

Fuente: "Displays de 7 Segmentos" (en línea) SCRIBD, 2013
<http://es.scribd.com/doc/5811293/DISPLAY-7-SEGMENTOS>

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.

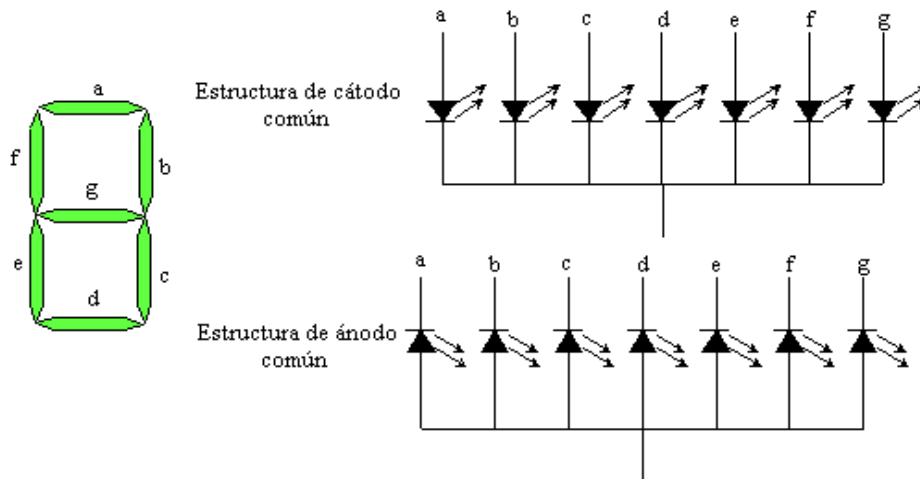


Figura 21. Estructura de un Display ánodo y cátodo común

Fuente: "Displays de 7 Segmentos" (en línea) SCRIBD, 2013
<http://es.scribd.com/doc/5811293/DISPLAY-7-SEGMENTOS>

2.2.7. Módulo LCD ¹⁴

Las pantallas de cristal líquido LCD o display LCD para mensajes (*Liquid Cristal Display*) 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.

¹⁴"Módulo LCD" (en línea) autor: Luis Rosero, SCRIBD 2010,
<http://es.scribd.com/doc/44252680/LCD-16X2>

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 la tensión desde 0.2V hasta los 4.5V mínimos necesario sea entre 0.1 ms y 10ms. Igualmente el tiempo de desconexión debe ser como mínimo de 1 ms antes de volver a conectar.

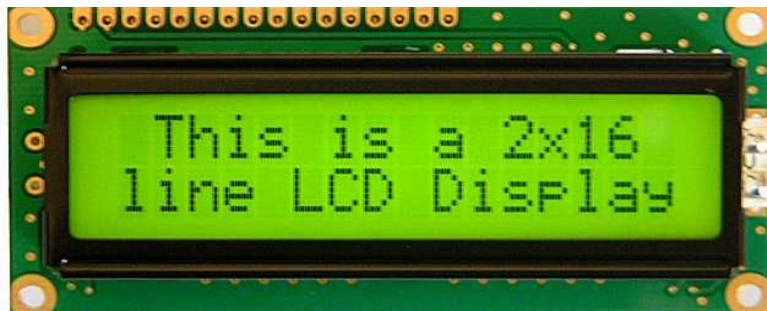


Figura 22. *LCD 2x16: está compuesto por 2 líneas de 16 caracteres*

Fuente: "LCD 2x16" (en línea) electronica4u, 2012,
<http://electronica4u.blogspot.com/2012/03/how-16x2-alphanumeric-lcd-works.html>

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.

2.2.8. 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.¹⁵

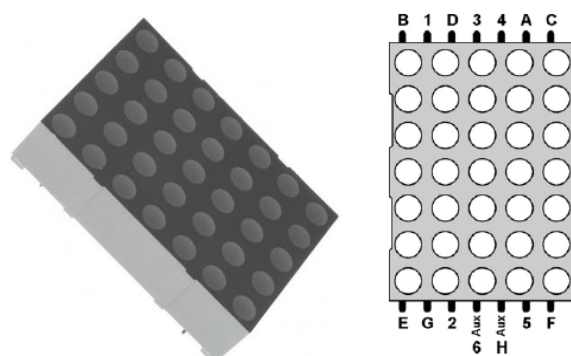


Figura 23. Forma real y pines de conexión de una matriz leds de 5x7

Fuente: "Matriz de Leds" (en línea) Autor: A. Tamayo, 2009, <http://galaxi0.wordpress.com/about/salidas-y-entradas-digitales/matriz-de-leds/>

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).

¹⁵ "Matriz de Leds" (en línea) Autor: A. Tamayo, 2009, <http://galaxi0.wordpress.com/about/salidas-y-entradas-digitales/matriz-de-leds/>

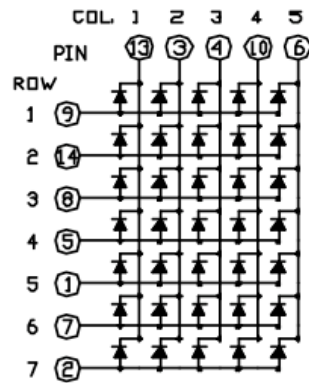


Figura 24. Estructura interna de conexión de una matriz leds de 5x7

Fuente: “Matriz de Leds” (en línea) Autor: A. Tamayo, 2009,
<http://galaxi0.wordpress.com/about/salidas-y-entradas-digitales/matriz-de-leds/>

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).¹⁶

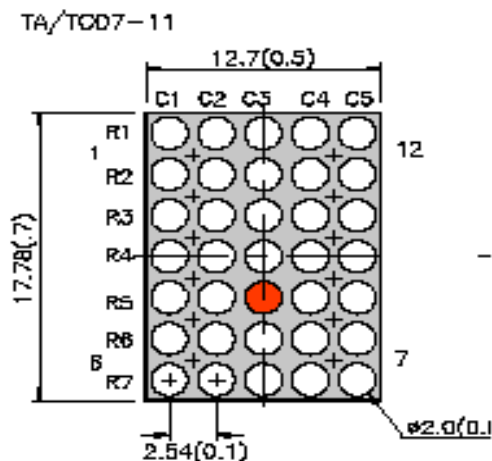


Figura 25. Como encender un led de una matriz de 5x7

Fuente: “Matriz de Leds” (en línea) Autor: A. Tamayo, 2009,
<http://galaxi0.wordpress.com/about/salidas-y-entradas-digitales/matriz-de-leds/>

¹⁶ “Matriz de Leds” (en línea) Autor: A. Tamayo, 2009, <http://galaxi0.wordpress.com/about/salidas-y-entradas-digitales/matriz-de-leds/>

2.3. SOFTWARE MICROCODE STUDIO.¹⁷

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 copilamos 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 Easy HID Wisard, 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.

¹⁷ Software Microcode Studio (en línea) autor: unrobotica 2010
<http://www.unrobotica.com/manuales/basic.pdf>

- Identifica y corrige la compilación y el ensamblado de errores.
- Ver la salida de serie del microcontrolador.

Palabra clave basada en el contexto de ayuda sensible al soporte para MP SAM.

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 llevará 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 Studio¹⁸

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

¹⁸ Pic que soporta Microcode Studio (en línea) Autor: MECANIQUE, <http://www.mecanique.co.uk/code-studio/>

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.¹⁹

Microcode Studio 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.

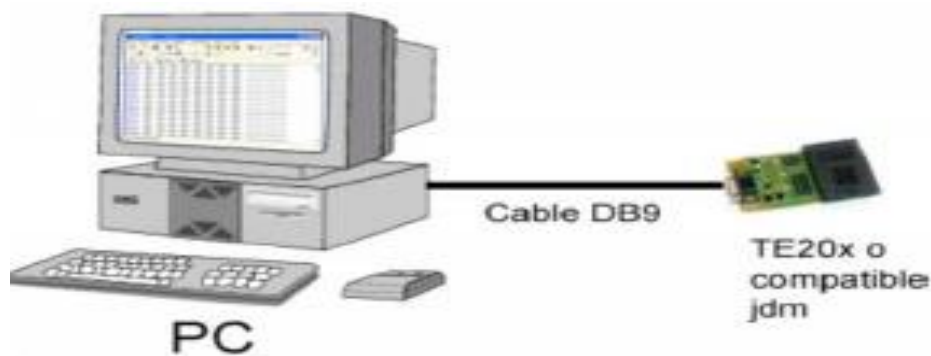


Figura 26. Comunicación entre el PIC y el PC

Fuente: Microcontroladores PIC Programación En BASIC, Carlos A. Reyes (TERCERA EDICION)

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

¹⁹ Microcontroladores PIC Programación En BASIC, Carlos A. Reyes (TERCERA EDICION), Pág. 25

En la actualidad las velocidades soportadas incluyen 4, 8, 10, 12, 16, 20, 24, 25, 32, 33 y 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.²⁰

2.3.4. Configuración de Microcode Studio (IDE).²¹

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:

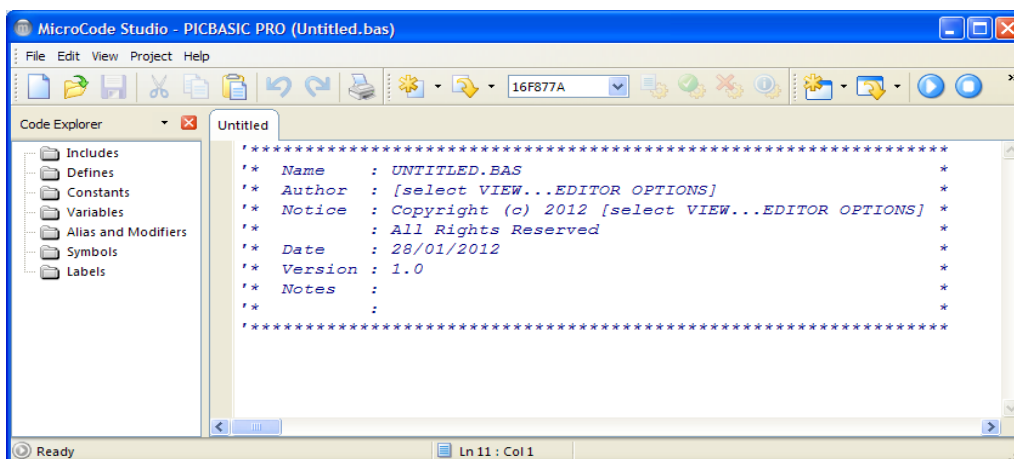


Figura 27. Pantalla principal de Microcode Studio

Fuente: Microcontroladores PIC Programación En BASIC, Carlos A. Reyes (TERCERA EDICION), Pág.25

²⁰ Microcontroladores PIC Programación En BASIC, Carlos A. Reyes (TERCERA EDICION), Pág. 25

²¹ Microcontroladores PIC Programación En BASIC, Carlos A. Reyes (TERCERA EDICION), Pág. 25

Si en esta pantalla buscamos **View** y 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.

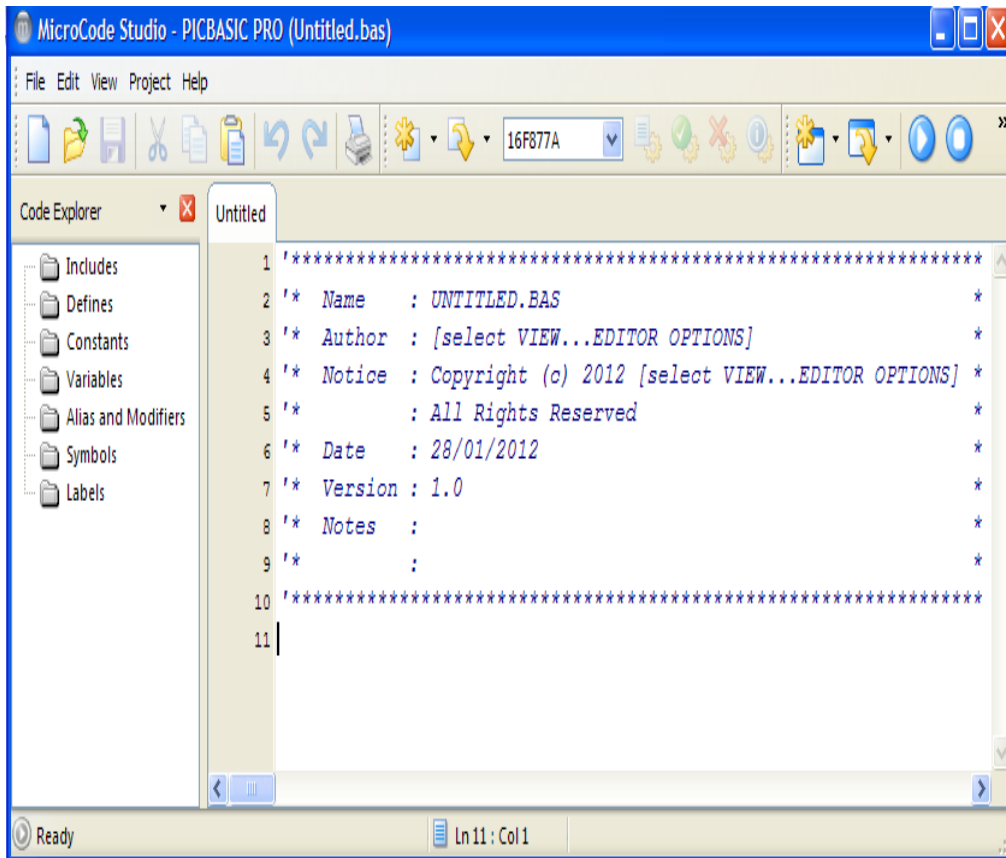


Figura 28. Pantalla principal con las líneas de programación

Fuente: Microcontroladores PIC Programación En BASIC, Carlos A. Reyes (TERCERA EDICION),

También podemos predefinir la carpeta en la que deseamos guardar los archivos *.pbp, por ejemplo podría ser una ubicación donde previamente hayamos creado una carpeta como: **C:\mis**

documentos\Prácticas\Practica1, si no deseamos crear ningún vínculo, sólo presionamos la tecla OK.

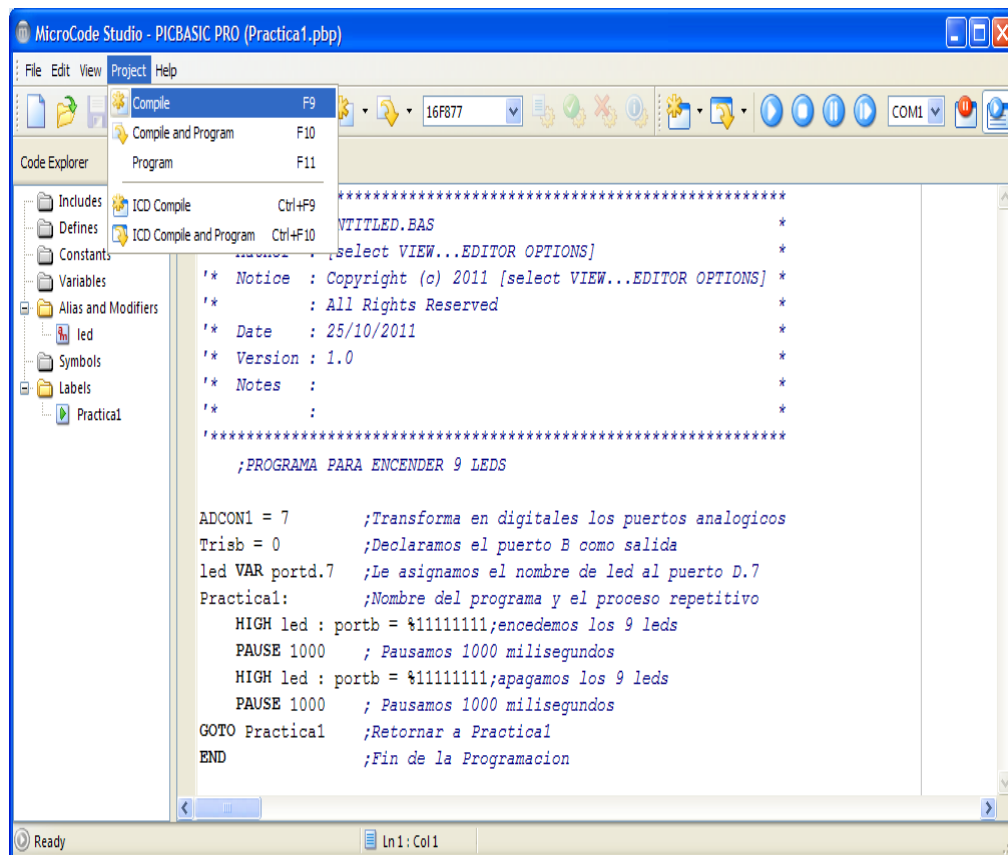


Figura 29. *Compilación de un programa*

Fuente: Microcontroladores PIC Programación En BASIC, Carlos A. Reyes (TERCERA EDICION), Pág.26

Una vez terminado un programa, lo podemos compilar para verificar que no tenga ningún error. Para ello buscamos **Project** y hacemos clic en **Compile** o en su defecto lo podemos compilar de manera directa desde el teclado presionando **F9**.

Si el programa se encuentra realizado en forma correcta nos aparecerá en pantalla **Ready**; y si tiene un error de cualquier naturaleza nos aparecerá **Compile Error**, permitiéndonos corregirlo para procurar el correcto y éxito del mismo.

2.4. FUENTE DE ALIMENTACIÓN DEL MÓDULO DE PRÁCTICAS²²

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).

²²"Fuente de alimentación" (en línea) Autor: MCB, Alicante (España) 2004-2008, http://www.mcbtec.com/fuentes_lineales.htm

2.4.1. Estructura Básica

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

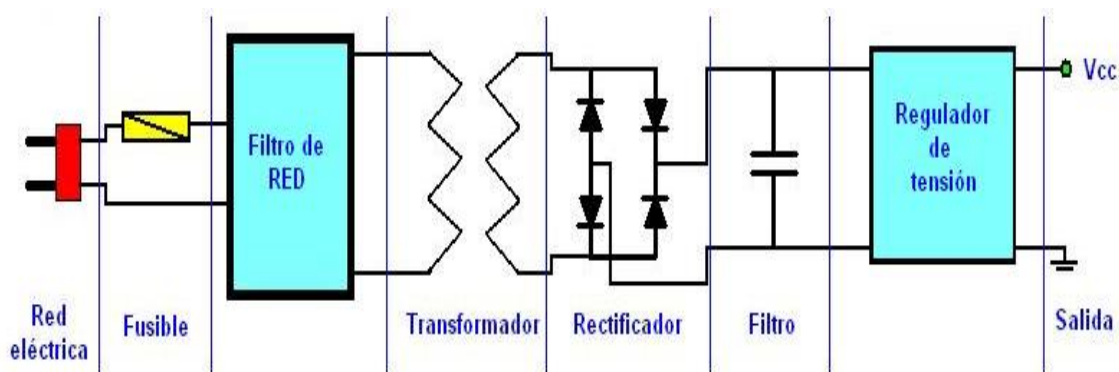


Figura 30. *Diagrama de Bloques de una fuente de alimentación lineal*

Fuente: “Fuente de alimentación” (en línea) Autor: MCB, Alicante (España) 2004-2008, http://www.mcbtec.com/fuentes_lineales.htm

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

2.4.2. Conexión Red Eléctrica ²³

Está 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

²³ “Fuente de alimentación” (en línea) Autor: MCB, Alicante (España) 2004-2008, http://www.mcbtec.com/fuentes_lineales.htm

consumir el circuito. Parámetros que serán también de gran importancia al momento de elegir el cable de alimentación.

2.4.3. Fusible²⁴

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.

2.4.4. Filtro de Red²⁵

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

²⁴ “Fuente de alimentación” (en línea) Autor: MCB, Alicante (España) 2004-2008, http://www.mcbtec.com/fuentes_lineales.htm

²⁵ “Fuente de alimentación” (en línea) Autor: MCB, Alicante (España) 2004-2008, http://www.mcbtec.com/fuentes_lineales.htm

imprescindibles 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.

2.4.5. Transformador²⁶

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.

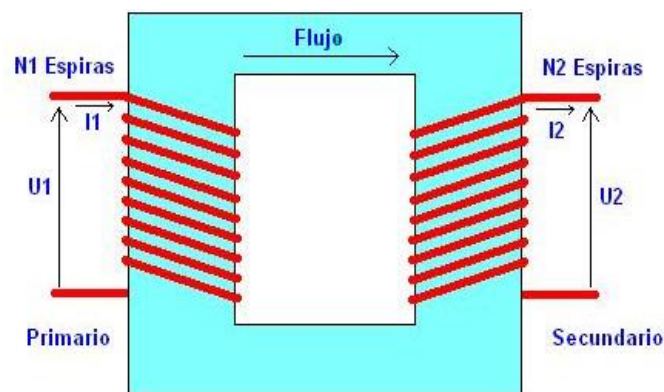


Figura 31. *Esquema de un transformador*

Fuente: "Fuente de alimentación" (en línea) Autor: MCB, Alicante (España) 2004-2008, http://www.mcbtec.com/fuentes_lineales.htm

²⁶"Fuente de alimentación" (en línea) Autor: MCB, Alicante (España) 2004-2008, http://www.mcbtec.com/fuentes_lineales.htm

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:

$$U_1 = -N_1 \frac{d\Phi}{dt} \quad U_2 = -N_2 \frac{d\Phi}{dt}$$

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

$$\frac{U_1}{U_2} = \frac{N_1}{N_2} = r$$

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 hubiera pérdidas la potencia del primario sería igual a la potencia del secundario más la potencia de las pérdidas). Esto nos permite igual las potencial del primario y del secundario según la siguiente ecuación:

$$P_1 = P_2 \Rightarrow U_1 \cdot I_1 = U_2 \cdot I_2 \Rightarrow \frac{U_1}{U_2} = \frac{I_2}{I_1}$$

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 válidas.

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 de 110V/12V y 1A.

Un devanado con toma intermedia: El secundario dispone de 3 bornes, en el que la tercera toma está conectada en medio de la bobina del secundario. Un ejemplo sería un transformador de 110V/12V+12V y 1A.

Dos devanados independientes: El secundario está 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 utilizarlas 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.

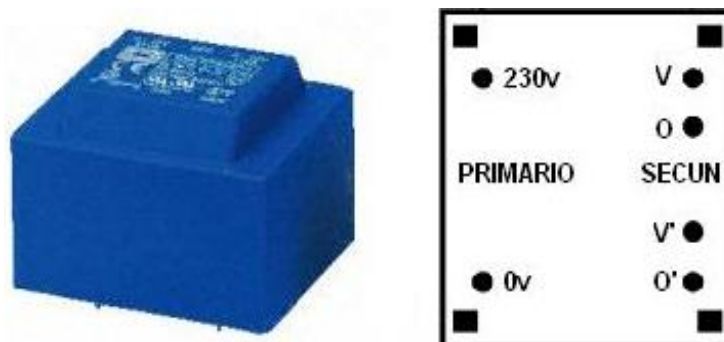


Figura 32. Transformador encapsulado

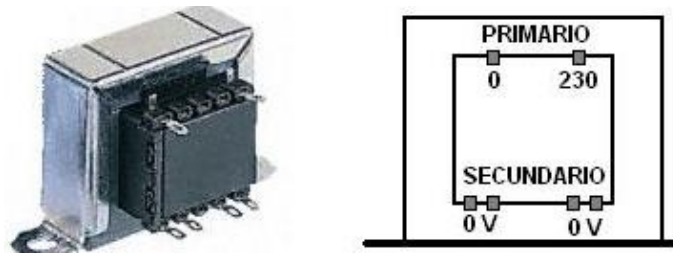


Figura 33. Transformador con brida

Fuente: “Fuente de alimentación” (en línea) Autor: MCB, Alicante (España) 2004-2008,
http://www.mcbtec.com/fuentes_lineales.htm

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 V_{pk} que podemos ver gráficamente en la figura 4 y que esta relaciona con la tensión eficaz mediante la siguiente ecuación:

$$V_{ef} = \frac{V_{pk}}{\sqrt{2}}$$

2.4.6. Rectificador

La mayoría de los circuitos electrónicos utilizan para funcionar corriente continua (DC), mientras que, como he 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 34**. Vemos la forma de la tensión alterna como sale del transformador y como queda después de rectificarla:

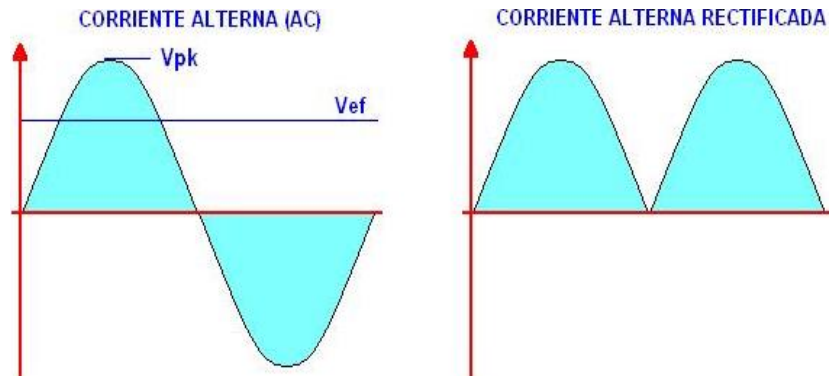


Figura 34. *Forma de onda rectificada.*

Fuente: “Fuente de alimentación” (en línea) Autor: MCB, Alicante (España) 2004-2008, http://www.mcbtec.com/fuentes_lineales.htm

Existen diversas configuraciones para realizar esta función, aquí nos vamos a centrar en la rectificación monofásica de onda completa. En la siguiente **Figura 35**. Se puede ver dos configuraciones para un rectificador de onda completa con transformador con y sin toma intermedia.

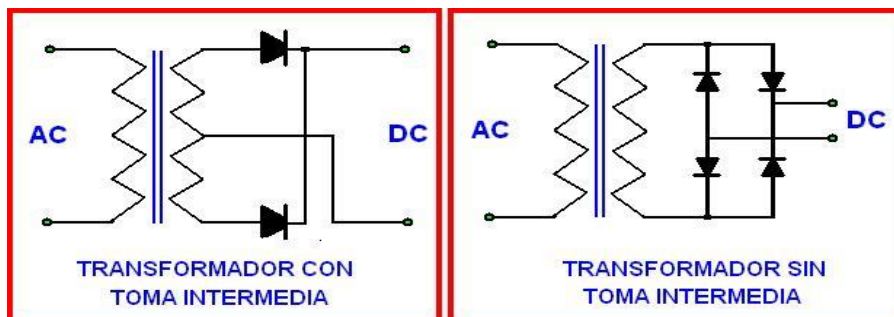


Figura 35. *Rectificación de onda completa*

Fuente: “Fuente de alimentación” (en línea) Autor: MCB, Alicante (España) 2004-2008, http://www.mcbtec.com/fuentes_lineales.htm

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.

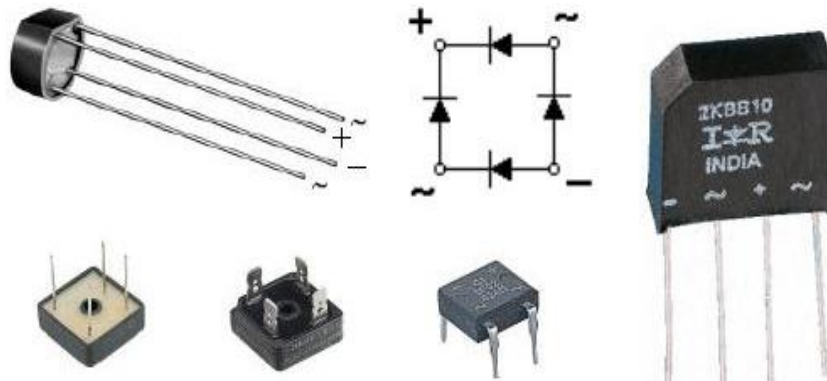


Figura 36. Encapsulados de puentes rectificadores

Fuente: “Fuente de alimentación” (en línea) Autor: MCB, Alicante (España) 2004-2008,
http://www.mcbtec.com/fuentes_lineales.htm

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).

2.4.7. Filtro

Una vez la señal esta rectificada, obtenemos una forma de onda que no es precisamente continua (ver **Figura 37.**). Para poder eliminar la ondulación, y dejar la tensión lo más continua posible, filtraremos la señal utilizando uno o más condensadores en paralelo. En la **Figura 37.** Se puede apreciar cómo queda esta señal una vez filtrada.

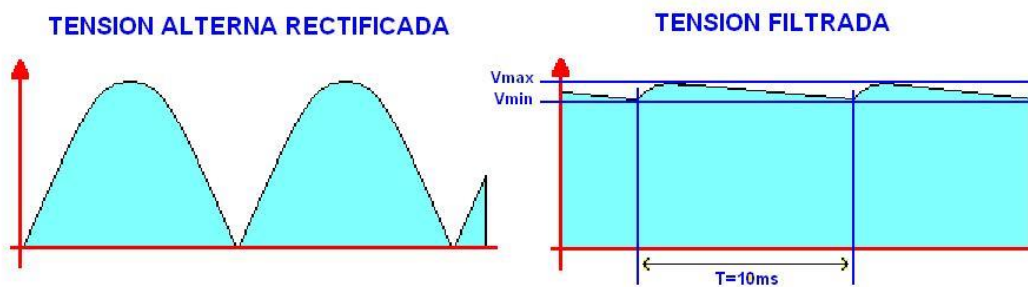


Figura 37. Señal filtrada después de la rectificación.

Fuente: "Fuente de alimentación" (en línea) Autor: MCB, Alicante (España) 2004-2008, http://www.mcbtec.com/fuentes_lineales.htm

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

$$C = \frac{Q}{V_{max} - V_{min}} = \frac{I_{max}T}{V_{max} - V_{min}}$$

En donde:

V_{max} : Es el valor máximo de la tensión de entrada que equivale al valor de pico del secundario del transformador (V_{pk}).

V_{min} : Tensión mínima que queremos que tenga la tensión de entrada y que determina el rizado de la fuente.

I_{max} : Intensidad máxima en el secundario.

T : Periodo de la señal de la red, para 60Hz y rectificador de onda completa son 10 ms. En media onda sería 20 ms.

C : Capacidad del condensador de filtro en faradios.

2.4.8. Regulador de Tensión

Como vemos en la de figura anterior 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 0,5A utilizando el L7805.

El esquema de la fuente de alimentación es el siguiente:

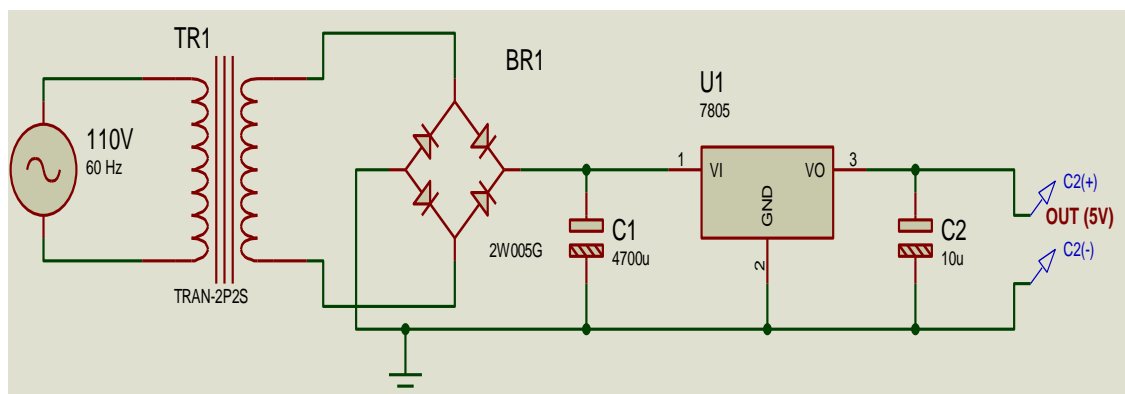


Figura 38. Fuente de alimentación de 5V a 1A

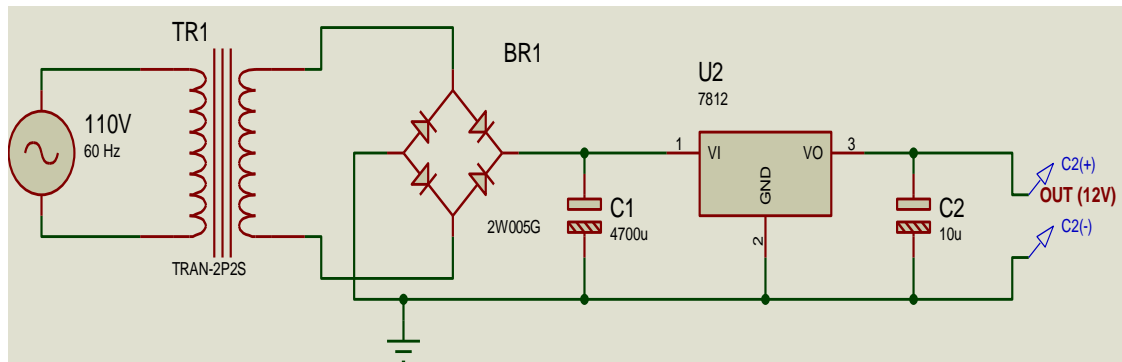


Figura 39. Fuente de alimentación de 12V a 1A

Calculemos cada uno de los componentes²⁷:

Transformador: El 7805 necesita una tensión mínima de trabajo, es decir, si tenemos un 7805 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 L7805 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 está 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 120V en primario y 9V+9V y 0.5A en secundario.

²⁷ “Fuente de alimentación” (en línea) Autor: MCB, Alicante (España) 2004-2008, http://www.mcbtec.com/fuentes_lineales.htm

Fusible: Utilizando la ecuación 1, 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

El valor de los parámetros de la ecuación es:

T: Para un rectificador de onda completa vale 10 ms.

I_{max}: Hemos determinado que la intensidad máxima que va a suministrar la fuente es 0,5A.

V_{max}: Mediante la ecuación 2 y sabiendo que la tensión eficaz del secundario es 9V AC, el valor de $V_{max} = 9 \times \sqrt{2} = 12.72V$

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 $C = 0,001344 \text{ F} = 1344 \text{ mF}$. Utilizaremos el valor comercial común más cercano por arriba que es de 2200 μF .

Regulador: Este va a ser el L7805, que da perfectamente los valores pedidos de 5V y 0,5A.

Otros componentes: El L7805 necesita un condensador de pequeño valor 100nF en la entrada y en la salida y añadimos un condensador más grande (470 μF) para estabilizar la tensión en la salida del circuito. El diodo D3 se utiliza para evitar que el L7805 se polarice en inversa y pueda sufrir algún daño.

Utilizando el mismo esquema y sustituyendo el L7805 por otros de la misma serie como puede ser el 7812, 7815, 7824 y recalculando el valor de todos los componentes podremos obtener fuentes de alimentación de distintas tensiones de salida.

Diseño del Radiador de Calor.²⁸

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), así tenemos $V_{reg} = V_{max} - V_{diodo} - V_{salida}$ y la multiplicaremos por la intensidad máxima, quedando al final la potencia como $P_{max} = (V_{max} - V_{diodo} - V_{salida}) * I_{max} = 7.02 * 0.5 = 3,51W$.

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 dispone en el encapsulado, la temperatura de este subirá 50°C. Con una simple

²⁸ "Fuente de alimentación" (en línea) Autor: MCB, Alicante (España) 2004-2008, http://www.mcbtec.com/fuentes_lineales.htm

multiplicación calculamos que si la potencia disipada por el L7805 es de 3,51W, la temperatura que alcanzara será de $3,51W \cdot 50^{\circ}C/W = 175,5^{\circ}C$ que es una temperatura que el integrado no puede soportar (la temperatura máxima de la unión es de $150^{\circ}C$), 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 la intensidad por la potencia, la tensión por temperatura y la resistencia por resistencia térmica tal como lo muestra la formulan siguiente.

$$V = IR \quad T = PR_{th}$$

El circuito térmico del 7805 sería:

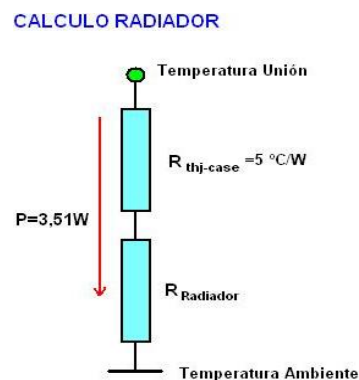


Figura 40. *Cálculo del Radiador*

Fuente: “Fuente de alimentación” (en línea) Autor: MCB, Alicante (España) 2004-2008, http://www.mcbtec.com/fuentes_lineales.htm

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

$$T_{Unión} = PR_{thj-case} + PR_{radiador} + T_{Ambiente}$$

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 L7805 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. Yo suelo utilizar el valor de 35°.

$R_{thj-case}$: Esta es la resistencia térmica entre la unión y el encapsulado. En el datasheet del L7805, 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:

$$R_{Radiador} = \frac{T_{Unión} - T_{Ambiente}}{P} - R_{thj-case}$$

Si sustituimos los valores en esta ecuación obtenemos un valor para $R_{Radiador}$ de 10,6 °C/W, por lo que buscaremos un radiador con una resistencia térmica de ese valor o inferior.²⁹

²⁹ "Fuente de alimentación" (en línea) Autor: MCB, Alicante (España) 2004-2008, http://www.mcbtec.com/fuentes_lineales.htm

MATERIALES

3.1. MATERIALES DEL MÓDULO

FUENTE DE PODER	
Cantidad	Elemento
1	Transformador con derivación:12V y 24V(1A)
2	Capacitores 4700uF
2	Capacitores 10uF
4	Diodos 1n4007
1	Regulador 78LS05
1	Regulador 78LS12
1	Placa de baquela virgen
2	Fundas de percloruro férrico

MÓDULO CENTRAL	
Cantidad	Elemento
23	Resistores 330Ω
24	Resistores 4,7kΩ
1	Resistores 100Ω
2	Capacitor 100nf
1	Capacitor 1000uf
1	Capacitor 10uf
5	TIP 120
1	17805
1	Potenciómetro 10kΩ
4	Transistores 2n3906
9	Transistores 2n3904
9	Leds Rojos
1	PIC16F628A
4	Displays 7 segmentos
1	LCD 16x2
1	Teclado hexadecimal 4x4
9	Diodos 1n4007
4	Relay 12V

8	Zócalo 18 Pines
1	Zócalo 14 Pines
6	Borneras de 3 Pines
1	Borneras de 2 Pines
1	Regleta macho
1	Regleta hembra
1	Placa de baquela virgen 20.5x31
1	Matriz de leds de 5x7
2	Fundas de percloruro férrico

PROCESO METODOLÓGICO

MÓDULO DE PRÁCTICAS PARA MICROCONTROLADORES

Este trabajo está orientado a permitir el contacto y conocimiento del medio organizacional público o privado y de la comunidad objeto de estudio con el estudiante.

Contribuye al mejoramiento de las condiciones de la organización en la cual se desarrolla el proyecto.

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, 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.

4.1.1. 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:

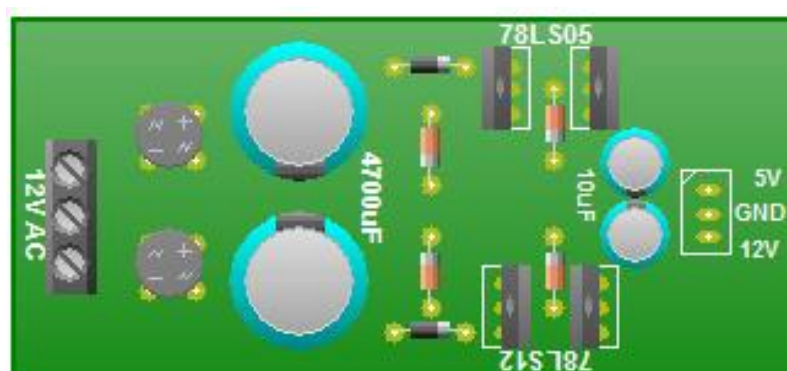


Figura 41. Vista 3D de la Fuente de Poder

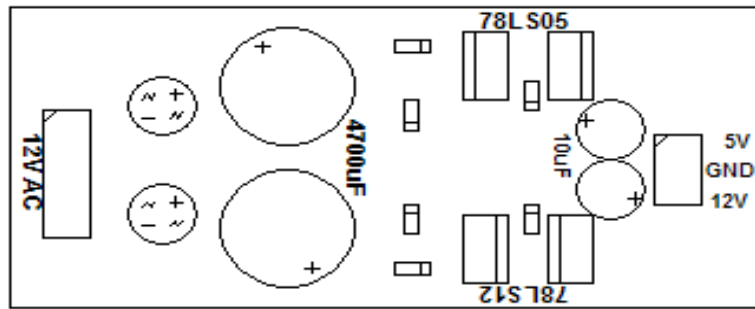


Figura 42. Etiquetas de la Fuente de Poder

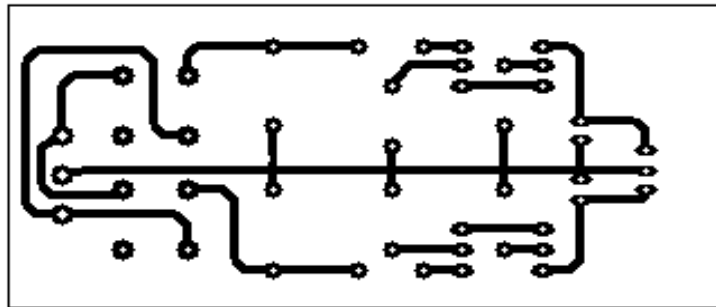


Figura 43. Ruteado de la Fuente de Poder

4.1.2. Diseño del Módulo de Prácticas

Una vez culminado el módulo de prácticas del presente proyecto de tesis, el cual es capaz de implementar cualquier tipo de prácticas con:

Leds.

Displays.

Matriz de leds de 5x7.

Relés.

Teclado matricial 4x4.

Motores, 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 cualquier software de dibujo como: Corel Draw, Orcad, Protel, PCB Wizard, Proteus ARES de Profesional (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 una solución de cloruro férrico diluida en agua, 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.

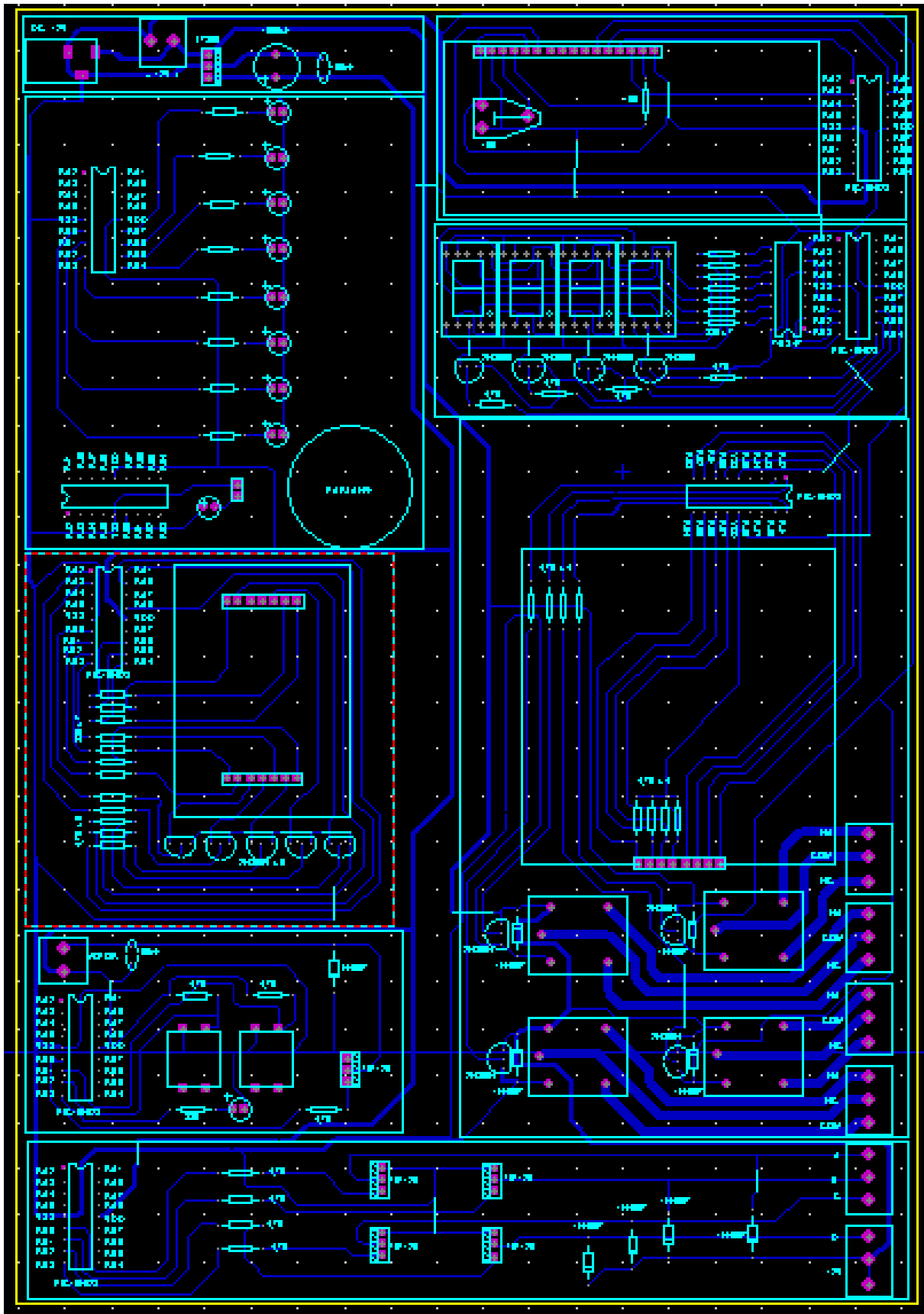


Figura 44. Esquema del Módulo de Prácticas

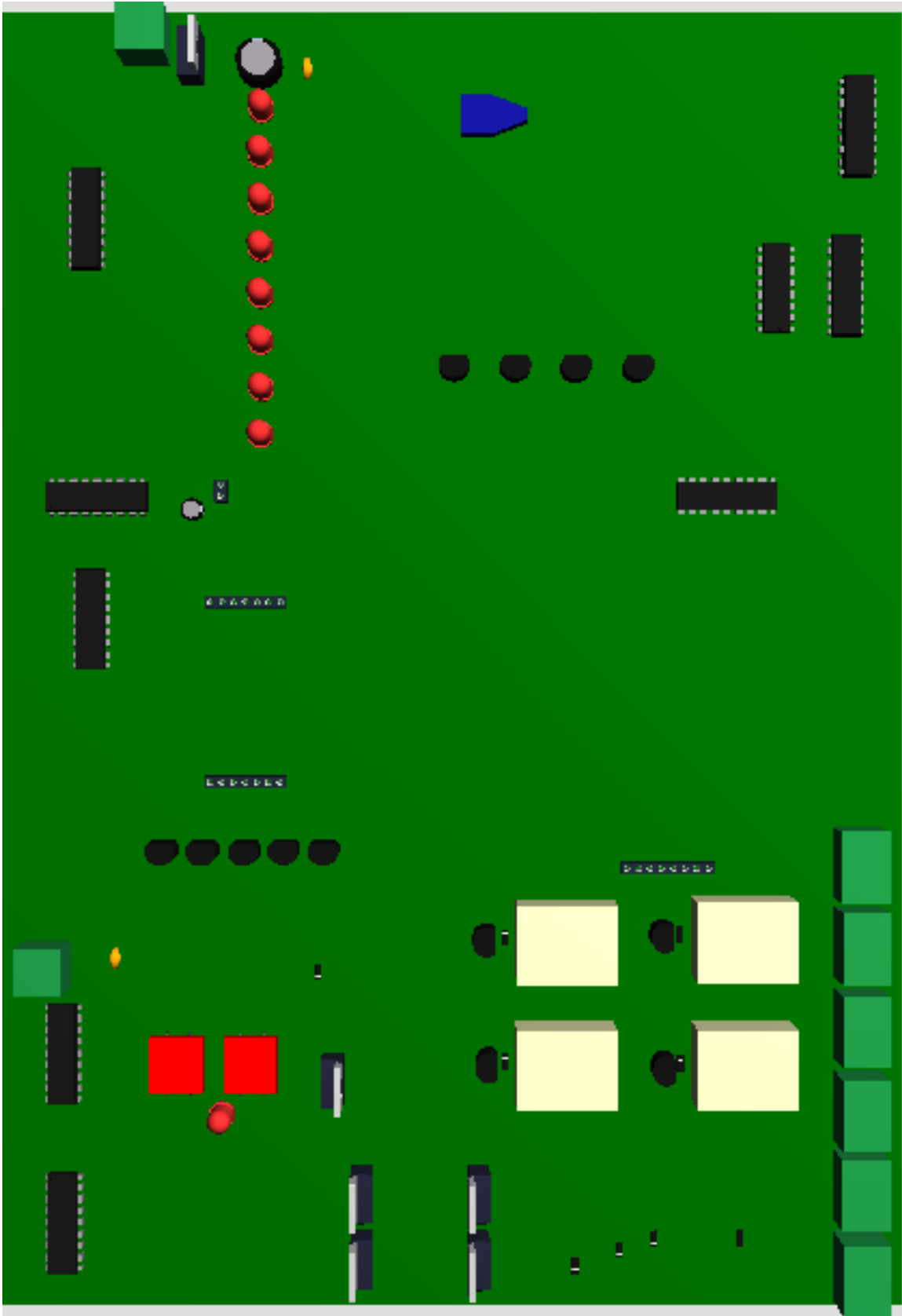


Figura 45. Vista 3D del Módulo de Prácticas

RESULTADOS

PRÁCTICAS PROPUESTAS PARA EL MÓDULO

Con la culminación de la investigación del proyecto de tesis propuesto, 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.

PRÁCTICA 1

TEMA: PARPADEO DE LEDS EN INTERVALOS DE UN SEGUNDO

OBJETIVOS:

Diseñar un algoritmo y un programa en el software de programación MICROCODE, que permita hacer parpadear ocho leds con intervalos de un segundo.

Analizar el comportamiento del circuito y el algoritmo del programa.

Comprobar el funcionamiento del circuito.

MATERIALES:

Cantidad	Designación	Características
8	Leds	3 Rojos, 3 Verdes, 2 Amarillos
8	Resistencias	330 Ω (1/2W)
1	Microcontrolador	PIC 16F628A

ESQUEMA:

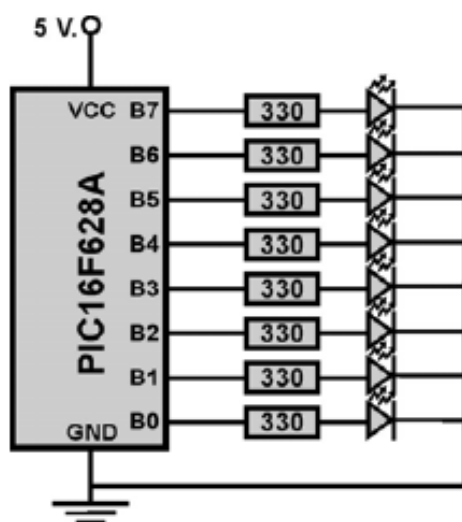


Figura 46.

SISTEMA CATEGORIAL:

Resistencias.

Código de colores para la identificación de resistencias.

Microcontroladores (PIC).

Datasheet PIC16F628AA.

Que es un algoritmo.

PROCEDIMIENTO:

Para realizar esta práctica en el módulo es necesario realizar la programación adecuada en MicroCode Studio y luego compilarla para proceder a grabar el archivo .HEX, para que cumpla dicha función el PIC, adicionalmente traer un pelacables y cable UTP para conexiones (si es necesario).

DESARROLLO:

El algoritmo correspondiente y programa correctamente elaborado en Microcode Studio para hacer parpadear los 8 leds armados en la placa que se realizó en Proteus nos queda de la siguiente forma:

```

;Programa para encender en secuencia las luces de una discoteca

Trisb=%00000000           ;convierte todos los pines del puerto B en salidas

practica1:                ;nombre de la subrutina
  Portb=%00000001         ;enciende el puerto B.0 , los demás permanecen apagados
  GOSUB pause             ; ir a subrutina tiempo y volver cuando diga RETURN
  Portb=%00000010        ;enciende el puerto B.1 , los demás les apaga
  GOSUB pause             ; ir a subrutina tiempo y volver cuando diga RETURN
  Portb=%00000100        ;enciende el puerto B.2 , los demás les apaga
  GOSUB pause             ; ir a subrutina tiempo y volver cuando diga RETURN
  Portb=%00001000        ;enciende el puerto B.3 , los demás les apaga
  GOSUB pause             ; ir a subrutina tiempo y volver cuando diga RETURN
  Portb=%00010000        ;enciende el puerto B.4 , los demás les apaga
  GOSUB pause             ; ir a subrutina tiempo y volver cuando diga RETURN
  Portb=%00100000        ;enciende el puerto B.5 , los demás les apaga
  GOSUB pause             ; ir a subrutina tiempo y volver cuando diga RETURN
  Portb=%01000000        ;enciende el puerto B.6 , los demás les apaga
  GOSUB pause             ; ir a subrutina tiempo y volver cuando diga RETURN
  Portb=%10000000        ;enciende el puerto B.7 , los demás les apaga
  GOSUB pause             ; ir a subrutina tiempo y volver cuando diga RETURN
  GOTO practica1         ; ir al inicio del programa

tiempo:                   ;esta es la subrutina tiempo
PAUSE 200                 ;retardo de 200 milisegundos, aquí podemos cambiarlo
RETURN                   ;volver al GOSUB que le envió

```

PREGUNTAS DE CONTROL:

- ¿Por qué es importante consultar el Datasheet del PIC?
- ¿Qué importancia tiene el Watiaje de las resistencias al momento de armar el circuito?
- ¿Qué consideraciones se debe tener para la selección de un PIC?
- ¿En qué se diferencia un PIC de otro?
- ¿Cuál es la diferencia entre un PIC y un Microprocesador?
- ¿Cuál es la ventaja principal de un PIC y un Microprocesador?
- ¿En la práctica, en qué se puede emplear el algoritmo?

PRÁCTICA 2

TEMA: SEMÁFORO DE DOS INTERSECCIONES CON SEMÁFORO PEATONAL.

OBJETIVOS:

1. Diseñar un algoritmo y un programa en el software de programación MICROCODE, que permita hacer un semáforo de dos intersecciones con cruce peatonal.
2. Analizar el comportamiento del circuito y el algoritmo del programa.

MATERIALES:

Cantidad	Designación	Características
8	Leds	3 Rojos, 3Verdes, 2 Amarillos
8	Resistencias	220 Ω (1/2W)
1	Microcontrolador	PIC 16F628A

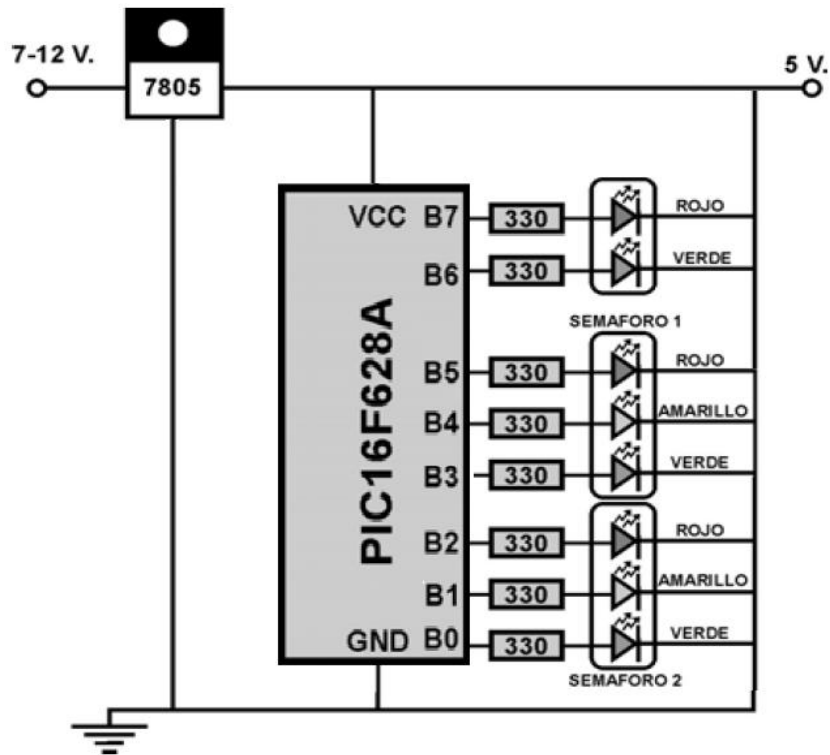
ESQUEMA:

Figura 47.

SISTEMA CATEGORIAL:

1. Resistencias.
2. Código de colores para la identificación de resistencias.
3. Microcontroladores (PIC).
4. Datasheet 16F628A.
5. Algoritmos de semaforización.

PROCEDIMIENTO:

Para la programación del PIC utilizamos Microcode Studio.

1. Declaramos el puerto B como salida.
2. Le asignamos un nombre al programa **semáforo**.
3. Encendemos el Verde1, el Rojo2 y verde3 Le damos una pausa al programa de 9000 milisegundos.
4. Apagamos el Verde1 y Encendemos el Amarillo1.
5. Le damos una nueva pausa al programa de 3000 milisegundos.
6. Apagamos el Amarillo1, el Verde 3 y Encendemos el Rojo1 y Rojo 3 y el Verde2
7. Le damos una nueva pausa al programa de 9000 milisegundos.
8. Apagamos el Verde2 y Encendemos el Amarillo2.
9. Le damos una nueva pausa al programa de 3000 milisegundos.
10. Apagamos el Amarillo2 y Encendemos el Rojo2 y el Verde3
11. Le damos una nueva pausa al programa de 9000 milisegundos.
12. Repetimos el programa de manera indefinida.

Cargamos el archivo **.HEX** generado por Microcode en el PIC 16F628A y alimentamos el modulo, y los ocho leds deben parpadear en intervalos de nueve segundo.

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 (si es necesario).

DESARROLLO:

El algoritmo correspondiente y programa correctamente elaborado en Microcode Studio para hacer parpadear los ocho leds armados en el módulo nos queda de la siguiente forma:

```

; SEMÁFORO DE DOS INTERSECCIONES CON SEMÁFORO PEATONAL
|
| trisb=0 ;indica que todos los pines del puerto B son de salida
| semaforo: ;nombre de la línea semáforo
| portb=%10100001 ;encender rojo del 1er semáforo y verde del 2do semáforo
| PAUSE 9000 ;esperar 9 segundos
| portb=%10100010 ;cambiar en el 2do semáforo de verde a amarillo
| PAUSE 2000 ;esperar 3 segundos
| portb=%01001100 ;cambiar a verde en el 1er semáforo y rojo el 2do semáforo
| PAUSE 9000 ;esperar 9 segundos
| portb=%01010100 ;cambiar en el 1er semáforo de verde a amarillo
| PAUSE 2000 ;esperar 3 segundos
| GOTO semaforo ;continuar con el ciclo para siempre
| END ; fin de la programación

```

PREGUNTAS DE CONTROL:

6. ¿Por qué es importante consultar el Datasheet del PIC?
7. ¿Qué importancia tiene el Watiaje de las resistencias al momento de armar el circuito?

8. ¿Qué consideraciones se debe tener para la selección de un PIC?
9. ¿En qué se diferencia un PIC de otro?
10. ¿Cuál es la diferencia entre un PIC y un Microprocesador?
11. ¿Cuál es la ventaja principal de un PIC y un Microprocesador?
12. ¿Por qué no utilizar un Microprocesador para ésta práctica?
13. ¿En qué consiste un algoritmo de semaforización?
14. ¿En la práctica, en qué se puede emplear el algoritmo?
15. ¿Se puede implementar el algoritmo con Focos de 110V-AC?
¿Cómo?

PRÁCTICA 3

TEMA: DECODIFICACIÓN Y MANEJO DE DISPLAYS.

OBJETIVOS:

1. Diseñar un algoritmo y un programa en el software de programación MICROCODE, que permita decodificar en un arreglo de Displays el número 5678.
2. Analizar el comportamiento del circuito y el algoritmo del programa.

MATERIALES:

Cantidad	Designación	Características
4	Displays	Ánodo común
4	Transistores	2N3906
4	Resistencia	4.7K Ω
7	Resistencias	330 Ω
1	Decodificador	74LS47
1	Microcontrolador	PIC 16F628A

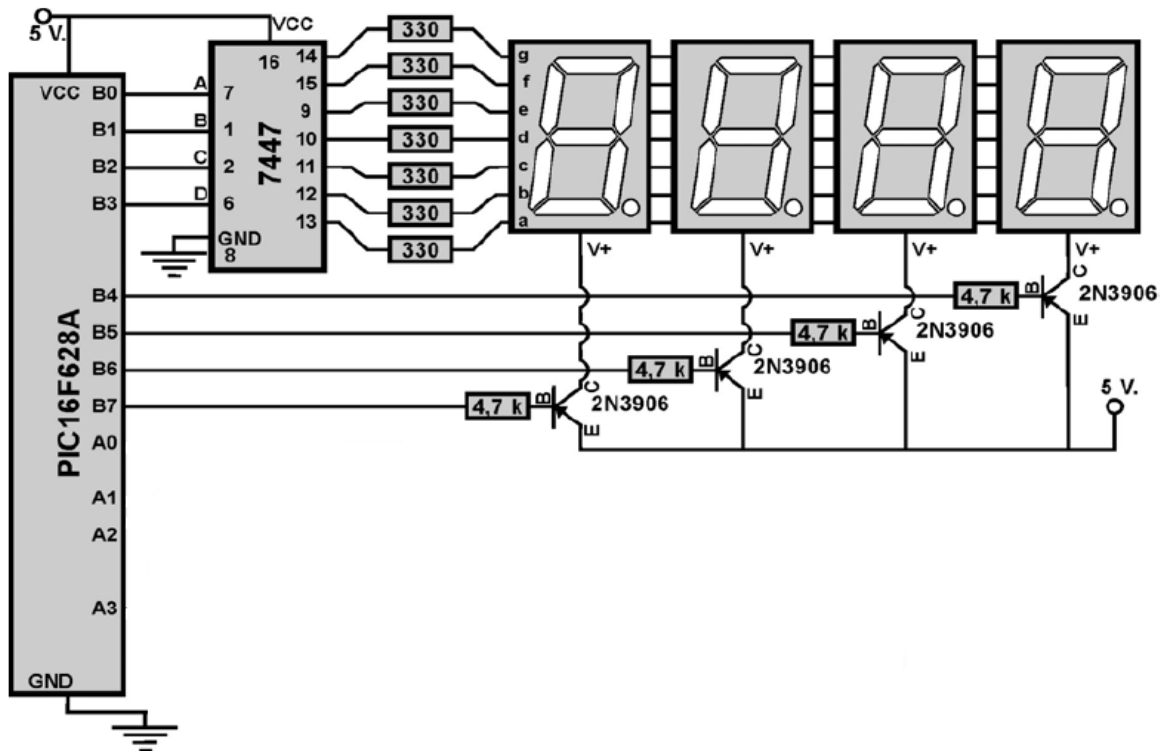
ESQUEMA:

Figura 48.

SISTEMA CATEGORIAL:

1. Código numérico y de colores para la identificación de capacitores.
2. Resistencias.
3. Código de colores para la identificación de resistencias.
4. Microcontroladores (PIC).
5. Datasheet 16F628A.
6. Que es un algoritmo.
7. Secuencias FOR.

8. Transistores PNP.
9. Datasheet 3906.
10. Displays 7 segmentos.
11. Datasheet Display Ánodo Común.
12. Datasheet 74LS47.

PROCEDIMIENTO:

Para la programación del PIC utilizamos Microcode Studio.

1. Transformamos a digitales todos los puertos analógicos del PIC.
2. Declaramos los puertos A y B como salida.
3. 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.
4. Activamos el DISPLAY0 y en éste decodificaremos la letra H.
5. Le damos una pausa al programa de 200 milisegundos.
6. Activamos el DISPLAY1 y en éste decodificaremos la letra O.
7. Le damos una pausa al programa de 200 milisegundos.
8. Activamos el DISPLAY2 y en éste decodificaremos la letra L.
9. Le damos una pausa al programa de 200 milisegundos.

10. Activamos el DISPLAY3 y en éste decodificaremos la letra A.
11. Le damos una pausa al programa de 200 milisegundos.
12. Activamos el DISPLAY4 y en éste decodificaremos la letra P.
13. Le damos una pausa al programa de 200 milisegundos.
14. Activamos el DISPLAY1 y en éste decodificaremos la letra P.
15. Le damos una pausa al programa de 200 milisegundos.
16. Repetimos el programa de manera indefinida.

Cargamos el archivo **.HEX** generado por Microcode en el PIC 16F628A y alimentamos el modulo.

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.

DESARROLLO:

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

```

; PROGRAMA PARA PRESENTAR EL NÚMERO 5678 FIJOS
trisb=0                ; convierte en salida todo el puerto B

display:
  portb=224+8          ;224 %11100000,activa el transistor de las unidades
  PAUSE 5
  portb=208+7          ;208 %11010000,activa el transistor de las decenas
  PAUSE 5
  portb=176+6          ;176 %10110000,activa el transistor de las centenas
  PAUSE 5
  portb=112+5          ;112 %01110000,activa el transistor de los miles
  PAUSE 5
GOTO display
END

```

PREGUNTAS DE CONTROL:

1. ¿Por qué es importante consultar el Datasheet del PIC?
2. ¿Qué importancia tiene el Watiaje de las resistencias al momento de armar el circuito?
3. ¿Qué consideraciones se debe tener para la selección de un PIC?
4. ¿En qué se diferencia un PIC de otro?
5. ¿Cuál es la diferencia entre un PIC y un Microprocesador?
6. ¿Cuál es la ventaja principal de un PIC y un Microprocesador?
7. ¿Cómo funciona el decodificador?
8. ¿Qué tipo de código decodifica el 74LS47?
9. ¿Cómo funciona un display de 7 segmentos?
10. ¿Cómo se hace un barrido de displays?

11. ¿Cuál es la corriente de activación del transistor?
12. ¿En la práctica, en qué se puede emplear el algoritmo?
13. ¿Por qué no utilizar un Microprocesador para ésta práctica?

PRÁCTICA 4

TEMA: MANEJO DE ELEMENTOS EXTERNOS DE AUDIO (Parlante de 8Ω).

OBJETIVOS:

1. 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.
2. Analizar el comportamiento del circuito y el algoritmo del programa.

MATERIALES:

Cantidad	Designación	Características
1	Parlante de audio	8Ω
1	Capacitor	10uF
1	Microcontrolador	PIC 16F628A

ESQUEMA:

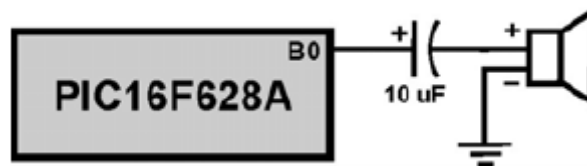


Figura 49.

SISTEMA CATEGORIAL:

1. Capacitores.

2. Código numérico y de colores para la identificación de capacitores.
3. Resistencias.
4. Código de colores para la identificación de resistencias.
5. Microcontroladores (PIC).
6. Datasheet 16F628A.
7. Que es un algoritmo.
8. Parlantes y niveles de resistencia.
9. Generación de Sonidos en Microcode.
10. Tabla de frecuencias para la generación de sonidos.

PROCEDIMIENTO:

Para la programación del PIC utilizamos Microcode Studio.

1. Le asignamos un nombre al programa.
2. Creamos una secuencia con la declaración SOUND para crear un sonido por el puerto B0.
3. Repetimos el programa de manera indefinida.

Cargamos el archivo **.HEX** generado por Microcode en el PIC y montamos en el módulo para la prueba.

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.

DESARROLLO:

```

                                ;programa para generar un timbre de celular
x VAR BYTE                        ;variable x de 255
prog:
FOR x = 1 TO 15                    ;repetir de 1 a 15 veces
    SOUND portB.0, [125, 4, 123,5] ;sacar tonos por el puerto b.0
    NEXT                            ;siguiente repetición
    PAUSE 2500                       ;esperar 2,5 segundos antes de volver a timbrar
GOTO prog

```

PREGUNTAS DE CONTROL:

11. ¿Por qué es importante consultar el Datasheet del PIC?
12. ¿Qué importancia tiene el Watiaje de las resistencias al momento de armar el circuito?
13. ¿Qué consideraciones se debe tener para la selección de un PIC?
14. ¿En qué se diferencia un PIC de otro?
15. ¿Cuál es la diferencia entre un PIC y un Microprocesador?
16. ¿Cuál es la ventaja principal de un PIC y un Microprocesador?
17. ¿Cómo se genera un sonido en Microcode?
18. ¿Qué frecuencias generan sonidos audibles?
19. ¿Por qué no se utiliza un parlante de un ohmiaje distinto al de 8Ω ?
20. ¿En la práctica, en qué se puede emplear el algoritmo?

PRÁCTICA 5

TEMA: DECODIFICACIÓN Y MANEJO DE UNA LCD.

OBJETIVOS:

1. 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.
2. Analizar el comportamiento del circuito y el algoritmo del programa.

MATERIALES:

Cantidad	Designación	Características
1	LCD	16 Bits
1	Resistencia	10 Ω
1	Potenciómetro	10k Ω
1	Microcontrolador	PIC 16F628A

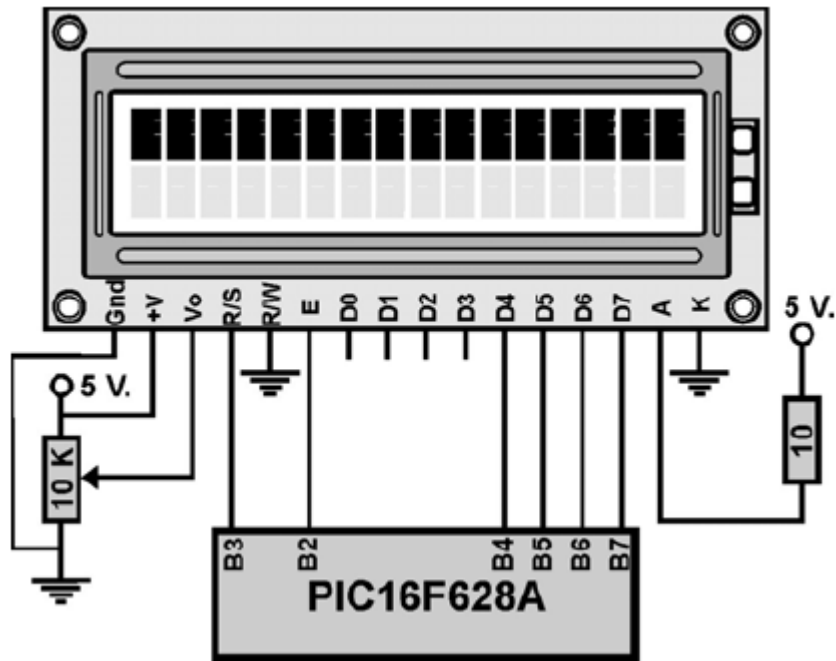
ESQUEMA:

Figura 50.

SISTEMA CATEGORIAL:

1. Resistencias.
2. Código de colores para la identificación de resistencias.
3. Microcontroladores (PIC).
4. Datasheet 16F628A.
5. Que es un algoritmo.
6. Declaración LCDOUT.
7. Módulo LCD.
8. Datasheet LCD 16 bits.

PROCEDIMIENTO:

Para la programación del PIC utilizamos Microcode Studio.

1. Transformamos a digitales todos los puertos analógicos del PIC.
2. Declaramos el puerto B como salida.
3. Le damos un retardo de 200 milisegundos para esperar que funcione el LCD.
4. Creamos cuatro variables (x, y, abc, xyz) de 255 bits.
5. Le asignamos un nombre al programa.
6. Configuramos el desplazamiento hacia la izquierda de la LCD.
7. Limpiamos la LCD.
8. Ubicamos el cursor de la LCD en la casilla 17.
9. Creamos una secuencia **FOR**, para poder sacar el nombre del TESISISTA en la LCD.
10. Dentro de la secuencia guardamos carácter por carácter el texto en la variable abc.
11. Ahora mostramos en la LCD el contenido de abc.
12. Le damos una pausa al programa de 400 milisegundos.
13. Creamos una secuencia **FOR**, para poder sacar el nombre del proyecto de tesis en la LCD.

14. Dentro de la secuencia guardamos carácter por carácter el texto en la variable xyz.
15. Ahora mostramos en la LCD el contenido de xyz.
16. Le damos una pausa al programa de 400 milisegundos.
17. Repetimos el programa de manera indefinida.

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.

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.

DESARROLLO:

El algoritmo correspondiente y programa correctamente elaborado en Microcode Studio nos queda de la siguiente forma:

	;programa para sacar texto en movimiento
PAUSE 200	;retardo para esperar que funcione el LCD
x VAR BYTE	;crear la variable x de 255
abc VAR BYTE	;crear la variable abc de 255
y VAR BYTE	;crear la variable abc de 255
texto VAR BYTE	;crear la variable abc de 255
LCDOUT \$FE,1	;limpiar pantalla
LCDOUT \$FE,2	;limpiar pantalla
LCDOUT \$FE,\$80," BIENVENIDOS "	;limpiar pantalla y sacar el texto Hola
LCDOUT \$FE,\$C0," TESIS UNL "	;pasar al comienzo de la segunda línea
PAUSE 4000	;esperar 4seg.
LCDOUT \$FE,\$7	;desplazamiento izquierdo
LCDOUT \$FE,1	;limpiar pantalla
ini:	
LCDOUT \$FE,\$90	;ubica el cursor en la casilla 17
FOR x = 0 TO 15	;repetir 15 veces
LOOKUP x,["*Maximo-Salinas*"],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
PAUSE 3000	;esperar 3seg.
LCDOUT \$fe,7	;desplazamiento izquierdo
LCDOUT \$FE,1	;limpia LCD
LCDOUT \$FE,\$90	;ubica el cursor en la casilla 17
FOR y = 0 TO 19	;repetir 19 veces
LOOKUP y,["PRACTICAS PIC16F628A"],texto	;tomar caracter por caracter y guardar en texto
LCDOUT,texto	;sacar en LCD el contenido de abc
PAUSE 400	;esperar 400 mls
NEXT	;siguiente repetición
PAUSE 500	;esperar 500 mls
LCDOUT \$FE,1	;limpiar pantalla
PAUSE 1000	;esperar 1seg.
GOTO ini	
END	

PREGUNTAS DE CONTROL:

1. ¿Por qué es importante consultar el Datasheet del PIC?
2. ¿Qué importancia tiene el Watiage de las resistencias al momento de armar el circuito?

3. ¿Qué consideraciones se debe tener para la selección de un PIC?
4. ¿En qué se diferencia un PIC de otro?
5. ¿Cuál es la diferencia entre un PIC y un Microprocesador?
6. ¿Cuál es la ventaja principal de un PIC y un Microprocesador?
7. ¿En la práctica, en qué se puede emplear el algoritmo?
8. ¿Por qué no utilizar un Microprocesador para ésta práctica?
9. Que es un módulo LCD 2x16

PRÁCTICA 6

TEMA: DECODIFICACIÓN Y MANEJO DE UNA MATRIZ DE LEDS DE 5X7.

OBJETIVOS:

1. Diseñar un algoritmo y un programa en el software de programación MICROCODE, que permita decodificar un muñeco en movimiento en una matriz de led.
2. Analizar el comportamiento del circuito y el algoritmo del programa.

MATERIALES:

Cantidad	Designación	Características
1	Matriz de leds	5x7
5	Resistencia	4.7k Ω
7	Resistencias	330 Ω
5	Transistores	42n3904
1	Microcontrolador	PIC 16F628A

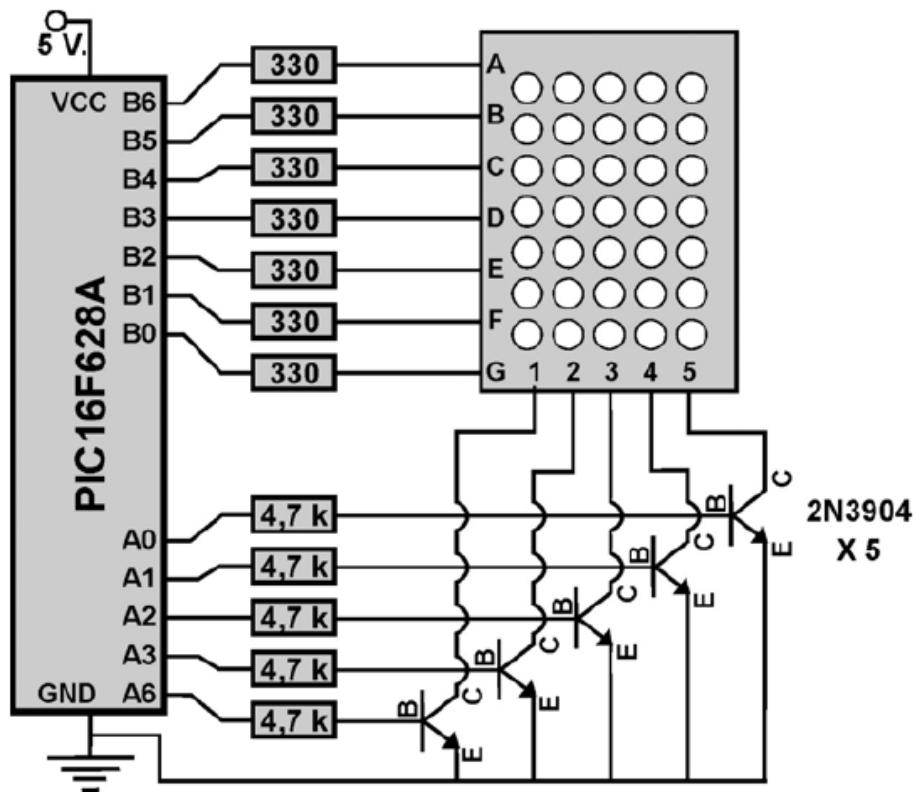
ESQUEMA:

Figura 51.

SISTEMA CATEGORIAL:

1. Resistencias.
2. Código de colores para la identificación de resistencias.
3. Microcontroladores (PIC).
4. Datasheet 16F628A.
5. Que es un algoritmo.
6. Leds.
7. Matriz de Leds.
8. Datasheet matriz 5x7.

PROCEDIMIENTO:

Para la programación del PIC utilizamos Microcode Studio.

1. Transformamos a digitales todos los puertos analógicos del PIC.
2. Declaramos el puerto B como salida.
3. Le asignamos un nombre al programa.
4. Creamos una secuencia **FOR**, para dibujar en los leds un muñeco.
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. Le damos una pausa al programa de 4 milisegundos.
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.
8. Le damos una pausa al programa de 4 milisegundos.
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.

10. Le damos una pausa al programa de 4 milisegundos.
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.
12. Repetimos el programa de manera indefinida, para darle un movimiento indefinido.

Cargamos el archivo **.HEX** generado por Microcode en el PIC y montamos en el módulo para su respectiva prueba.

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.

DESARROLLO:

El algoritmo correspondiente y programa correctamente elaborado en Microcode Studio 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
x VAR BYTE                               ;crea variable x de 255

Practica6:
FOR x = 1 TO 10                          ;repetir esta secuencia 10 veces
portb=%0000001 :portd=%0010000 :PAUSE 4 ;esperar 4 mls, total 20 mls cada
portb=%0000010 :portd=%0100111 :PAUSE 4 ;escena
portb=%0000100 :portd=%1111000 :PAUSE 4
portb=%0001000 :portd=%0100111 :PAUSE 4
portb=%1000000 :portd=%0010000 :PAUSE 4
NEXT
FOR x = 1 TO 10                          ;repetir esta secuencia 10 veces
portb=%0000001 :portd=%0100000 :PAUSE 4
portb=%0000010 :portd=%0100111 :PAUSE 4
portb=%0000100 :portd=%1111000 :PAUSE 4
portb=%0001000 :portd=%0100111 :PAUSE 4
portb=%1000000 :portd=%0100000 :PAUSE 4
NEXT
FOR x = 1 TO 10                          ;repetir esta secuencia 10 veces
portb=%0000001 :portd=%1000000 :PAUSE 4
portb=%0000010 :portd=%0100111 :PAUSE 4
portb=%0000100 :portd=%1111000 :PAUSE 4
portb=%0001000 :portd=%0100111 :PAUSE 4
portb=%1000000 :portd=%1000000 :PAUSE 4
NEXT
FOR x = 1 TO 10                          ;repetir esta secuencia 10 veces
portb=%0000001 :portd=%0100000 :PAUSE 4
portb=%0000010 :portd=%0100111 :PAUSE 4
portb=%0000100 :portd=%1111000 :PAUSE 4
portb=%0001000 :portd=%0100111 :PAUSE 4
portb=%1000000 :portd=%0100000 :PAUSE 4
NEXT
GOTO Practica6                          ;ir a Practica9
END

```

PREGUNTAS DE CONTROL:

1. ¿Por qué es importante consultar el Datasheet del PIC?
2. ¿Qué importancia tiene el Watiaje de las resistencias al momento de armar el circuito?
3. ¿Cuán importante es la utilización de un cristal de cuarzo en el circuito?
4. ¿Qué consideraciones se debe tener para la selección de un PIC?
5. ¿En qué se diferencia un PIC de otro?
6. ¿Cuál es la diferencia entre un PIC y un Microprocesador?
7. ¿Cuál es la ventaja principal de un PIC y un Microprocesador?
8. ¿Qué es una matriz?
9. ¿Cómo funciona una matriz 5x7 de leds?
10. ¿Qué se puede hacer con una matriz de leds?
11. ¿Cuánta corriente consume cada led del arreglo?
12. ¿En la práctica, en qué se puede emplear el algoritmo?
13. ¿Por qué no utilizar un Microprocesador para ésta práctica?

PRÁCTICA 7

TEMA: DECODIFICACIÓN Y MANEJO DE UN TECLADO 4x4 CON RELÉS DE 110V.

OBJETIVOS:

1. 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.
2. Analizar el comportamiento del circuito y el algoritmo del programa.

MATERIALES:

Cantidad	Designación	Características
4	Boquilla	110V-AC
4	Foco	110V-AC (30W)
4	Relé	5V-DC
4	Diodo	1N4007
4	Transistor	2N3904
8	Resistencias	4,7k Ω (1/2W)
1	Microcontrolador	PIC 16F628A

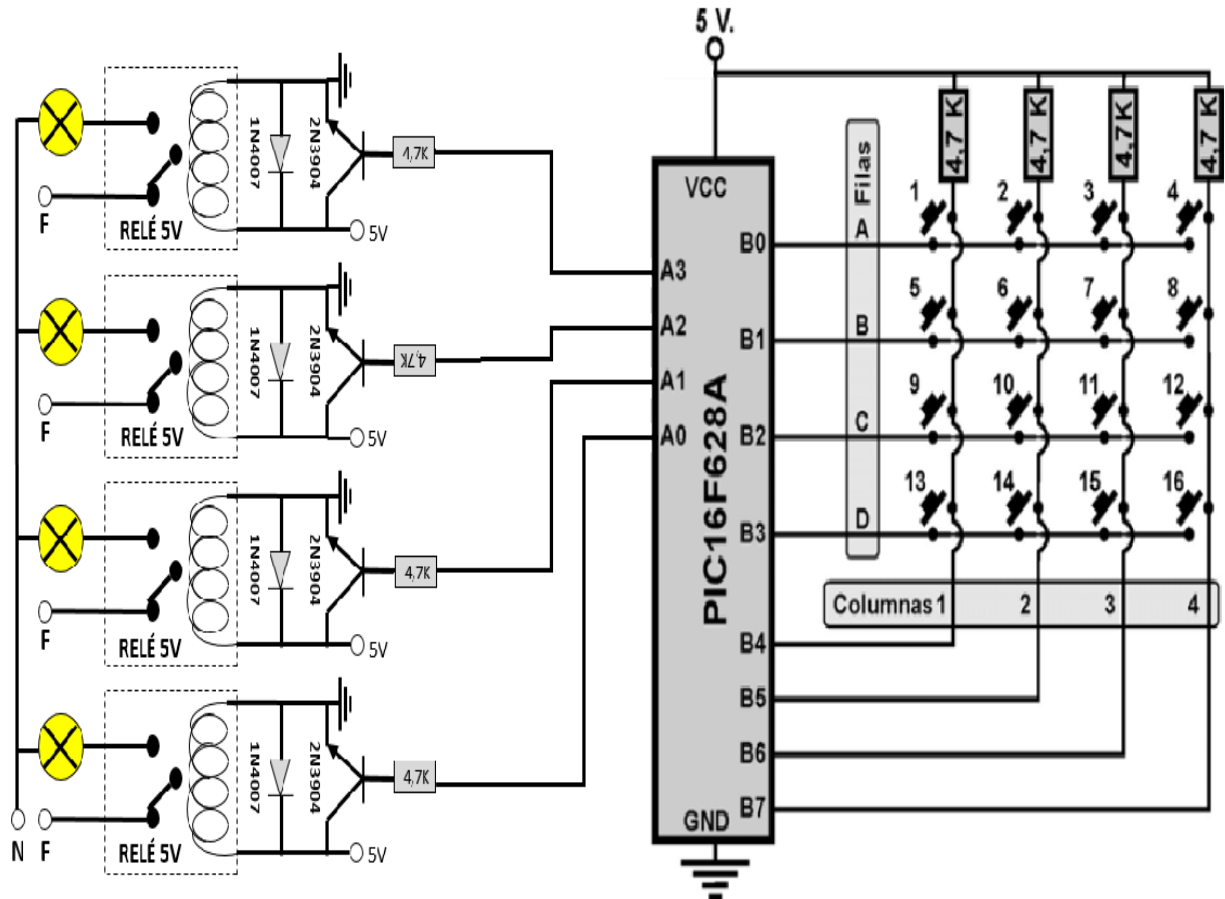
ESQUEMA:

Figura 52.

SISTEMA CATEGORIAL:

1. Resistencias.
2. Código de colores para la identificación de resistencias.
3. Microcontroladores (PIC).
4. Datasheet 16F628A.
5. Que es un algoritmo.
6. Pulsadores, modos de configuración

7. Diodos.
8. Transistores.
9. Datasheet 3904.
10. Relés, modos de conexión y voltajes de funcionamiento.
11. Datasheet relé 5V.
12. Corriente Alterna (AC).
13. Focos 110V-AC.
14. Teclados.
15. Datasheet teclado 4x4.

PROCEDIMIENTO:

Para la programación del PIC utilizamos Microcode Studio.

1. Transformamos a digitales todos los puertos analógicos del PIC.
2. Declaramos el puerto A como salida.
3. Le asignamos un nombre al programa.
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.
5. Una vez decodificado el teclado, el resultado (activo a bajo) lo sacamos por el puerto A para poder conmutar los relés y

así encender los focos de 110V-AC, y visualizar en código BCD el valor.

6. Le damos una pausa al programa de 10 milisegundos.
7. Repetimos el programa de manera indefinida.

Cargamos el archivo **.HEX** generado por Microcode en el PIC 16F628A y alimentamos el modulo, y los cuatro Relays deberán accionarse con el teclado

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.

DESARROLLO:

El algoritmo correspondiente y programa correctamente elaborado en Microcode nos queda de la siguiente forma:

;DECODIFICACIÓN Y MANEJO DE UN TECLADO 4x4 CON RELÉS DE 110V

```

cmcon=7                ;cambiar a modo digital todo el puerto A
TRISA = 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

Practica7:
LOW A                 ;hacer bajo la 1era fila
IF UNO = 0 THEN PORTA = 1 ;si la 1ra tecla es presionada sacar 1
IF DOS = 0 THEN PORTA = 2 ;si la 2da tecla es presionada sacar 2
IF TRES = 0 THEN PORTA = 3 ;si la 3ra tecla es presionada sacar 3
IF CUATRO = 0 THEN PORTA = 10 ;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 PORTA = 4 ;si la 1ra tecla es presionada sacar 4
IF DOS = 0 THEN PORTA = 5 ;y así sucesivamente
IF TRES = 0 THEN PORTA = 6
IF CUATRO = 0 THEN PORTA = 11
HIGH B                ;poner en alto la 2da fila
LOW C                 ;hacer bajo la 3ra fila
IF UNO = 0 THEN PORTA = 7
IF DOS = 0 THEN PORTA = 8
IF TRES = 0 THEN PORTA = 9
IF CUATRO = 0 THEN PORTA = 12
HIGH C                ; poner en alto la 3ra fila
LOW D                 ; hacer bajo la 4ta fila
IF UNO = 0 THEN PORTA = 14
IF DOS = 0 THEN PORTA = 0
IF TRES = 0 THEN PORTA = 15
IF CUATRO = 0 THEN PORTA = 13
HIGH D                ; poner en alto la 4ta fila
PAUSE 10              ; pausa de 10 milisegundos
GOTO Practica7        ; continuar con el barrido de teclas

END

```

PREGUNTAS DE CONTROL:

1. ¿Por qué es importante consultar el Datasheet del PIC?
2. ¿Qué importancia tiene el Watiaje de las resistencias al momento de armar el circuito?
3. ¿Qué consideraciones se debe tener para la selección de un PIC?
4. ¿En qué se diferencia un PIC de otro?
5. ¿Cuál es la diferencia entre un PIC y un Microprocesador?
6. ¿Cuál es la ventaja principal de un PIC y un Microprocesador?
7. ¿Qué consideraciones de debe tener al momento de manipular voltajes AC?
8. ¿Cómo se conecta un foco de 110V-AC?
9. ¿Cómo funciona un relé?
10. ¿Qué niveles de voltaje se pueden conectar a los pines de conmutación del relé?
11. ¿Qué niveles de voltaje maneja un PIC?
12. ¿Se puede conectar 110V-AC a un PIC? ¿Por qué?
13. ¿Cómo se puede conectar 110V-AC a un PIC?
14. ¿Qué es un teclado?
15. ¿Cómo funciona un teclado?
16. ¿Cómo se configura un teclado 4x4?
17. ¿En la práctica, en qué se puede emplear el algoritmo?
18. ¿Por qué no utilizar un Microprocesador para ésta práctica?

PRÁCTICA 8

TEMA: MANEJO DEL PWM COMO VARIADOR DE VELOCIDAD DE UN MOTOR DC.

OBJETIVOS:

1. Diseñar un algoritmo y un programa en el software de programación MICROCODE, que permita variar la velocidad de un motor DC
2. Analizar el comportamiento del circuito y el algoritmo del programa.

MATERIALES:

Cantidad	Designación	Características
1	LED	Rojo de 5mm.
1	Transistor	TIP 120
2	Pulsadores	Normalmente abierto
1	Capacitor cerámico	0,1uF (104)
1	Diodo	1N4007
4	Transistor	2N3904
1	Resistencia	330Ω (1/2W)
3	Resistencias	4,7kΩ (1/2W)
1	Microcontrolador	PIC 16F628A

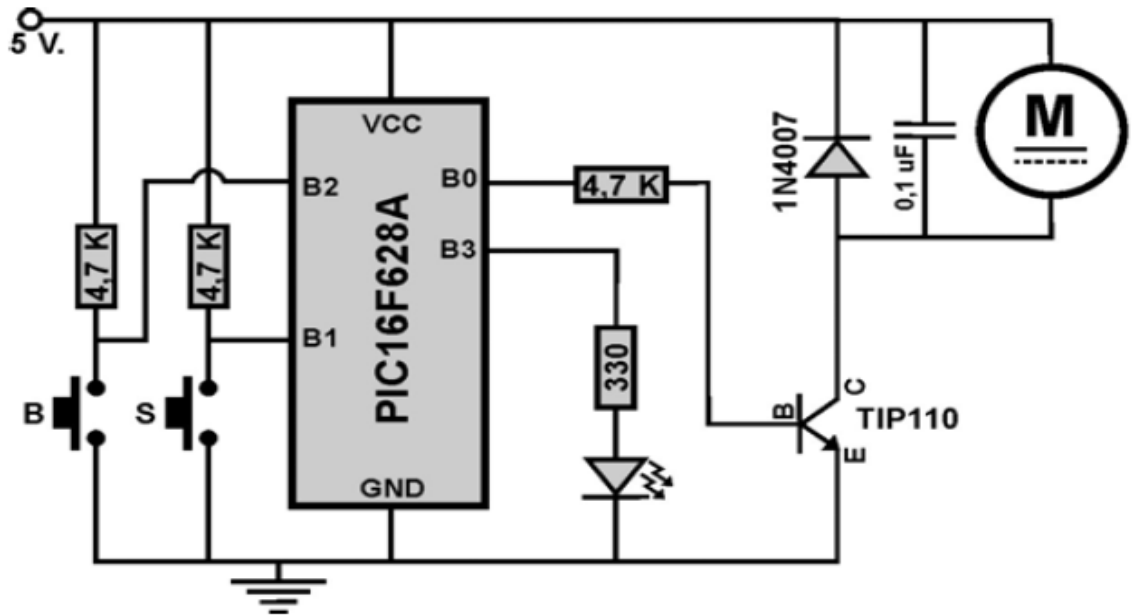
ESQUEMA:

Figura 53.

SISTEMA CATEGORIAL:

1. Resistencias.
2. Código de colores para la identificación de resistencias.
3. Microcontroladores (PIC).
4. Datasheet 16F628A.
5. Que es un algoritmo.
6. Pulsadores, modos de configuración
7. Diodos.
8. Transistores TIP 120.
9. Motores (DC).
10. LED

PROCEDIMIENTO:

Para la programación del PIC utilizamos Microcode Studio.

1. Colocamos el nombre a todos los puertos B del PIC.
2. Le asignamos la frecuencia al programa.
3. Programamos un puerto para el LED, este nos indicara que ya arranco la programación.
4. Le asignamos el PWM al programa para subir y bajar la velocidad con los pulsadores S y B.
5. Repetimos el programa de manera indefinida.

Cargamos el archivo **.HEX** generado por Microcode en el PIC 16F628A y alimentamos el modulo.

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.

DESARROLLO:

El algoritmo correspondiente y programa correctamente elaborado en Microcode nos queda de la siguiente forma:

;MANEJO DEL PWM COMO VARIADOR DE VELOCIDAD DE UN MOTOR DC.

```

botsubir VAR portb.1           ;nombre para el pin B1
botbajar VAR portb.2          ;nombre para el pin B2
LED VAR portb.3              ;nombre para el pin B3
FREC VAR BYTE                ;variable FREC tamaño 255
bandera VAR BIT              ;variable bandera de 1 bit
FREC = 125                    ;valor inicial para variable FREC

HIGH LED                      ;encender el led para saber que ya arrancó
PAUSE 500                     ;por medio segundo
LOW LED                       ;apagar el led

PROG:
PWM PORTB.0,FREC,30           ;sacar PWM 30 ciclos de 125 señal útil
LOW LED ;apagar el led
IF botsubir =0 THEN SUBIR     ;si se pulsa el botón S ir a subir
IF botbajar =0 THEN BAJAR    ;si se pulsa el botón B ir a bajar
bandera =0                       ;bandera cargado con cero
GOTO PROG

SUBIR:
IF FREC > 249 THEN aviso     ;si supera a 249 ir a aviso
IF bandera = 1 THEN prog     ;si la bandera esta en 1 salir
HIGH LED                     ;encender el led
bandera=1                         ;cargar la bandera con uno
FREC = FREC +25              ;sumar 25 a la variable FREC
GOTO PROG                   ;ir a prog

BAJAR:
IF FREC < 26 THEN aviso     ;si baja de 26 ir a aviso
IF bandera = 1 THEN prog     ;si la bandera esta en 1 salir
HIGH LED                     ;encender el led
bandera=1                         ;cargar la bandera con uno
FREC = FREC -25              ;restar 25 a la variable FREC
GOTO PROG                   ;ir a prog

aviso:
HIGH LED                     ;encender el led
GOTO PROG                   ;ir a prog

END

```

PREGUNTAS DE CONTROL:

1. ¿Por qué es importante consultar el Datasheet del PIC?
2. ¿Qué importancia tiene el Watiaje de las resistencias al momento de armar el circuito?
3. ¿Qué consideraciones se debe tener para la selección de un PIC?
4. ¿En qué se diferencia un PIC de otro?
5. ¿Cuál es la diferencia entre un PIC y un Microprocesador?
6. ¿Cuál es la ventaja principal de un PIC y un Microprocesador?
7. ¿Qué niveles de voltaje maneja un PIC?
8. ¿Qué es un motor DC?
9. ¿Cómo funciona un pulsador normalmente abierto?
10. ¿En la práctica, en qué se puede emplear el algoritmo?
11. ¿Por qué no utilizar un Microprocesador para ésta práctica?

PRÁCTICA 9

TEMA: MANEJO DE UN MOTOR PASO A PASO EN SECUENCIA WAVE DRIVE

OBJETIVOS:

1. Diseñar un algoritmo y un programa en el software de programación MICROCODE, que permita variar la velocidad de un motor DC
2. Analizar el comportamiento del circuito y el algoritmo del programa.

MATERIALES:

Cantidad	Designación	Características
4	Transistor	TIP 120
4	Diodo	1N4007
4	Resistencias	4,7k Ω (1/2W)
1	Microcontrolador	PIC 16F628A

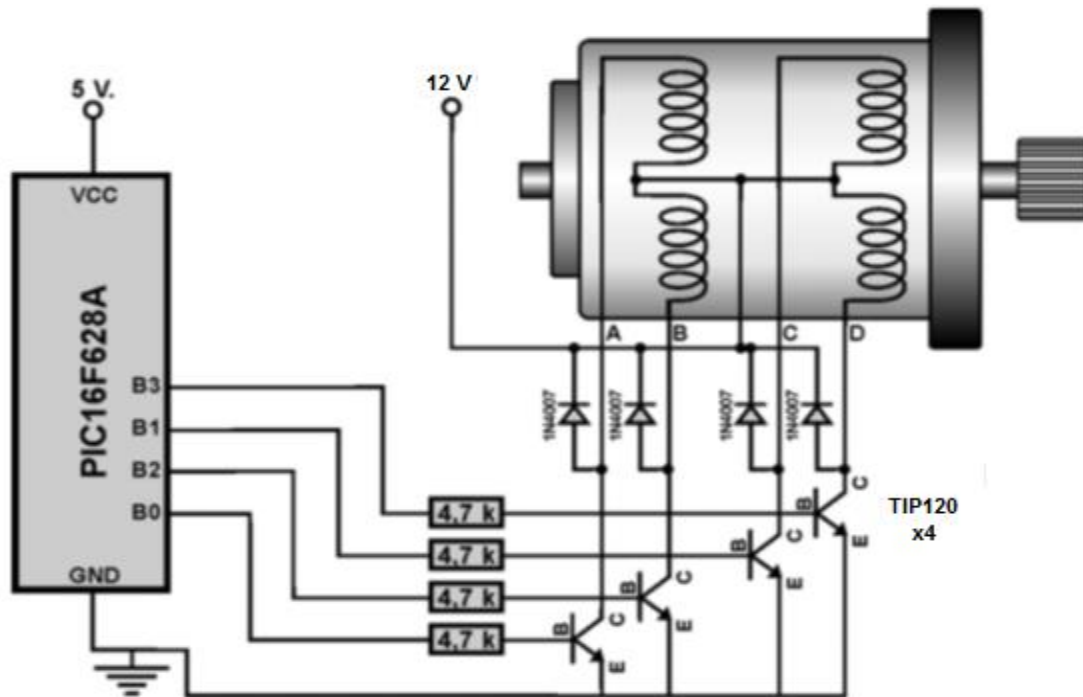
ESQUEMA:

Figura 54.

SISTEMA CATEGORIAL:

1. Resistencias.
2. Código de colores para la identificación de resistencias.
3. Microcontroladores (PIC).
4. Datasheet 16F628A.
5. Que es un algoritmo.
6. Diodos.
7. Transistores.
8. Datasheet 3904.
9. Motores paso a paso
10. Que significa WAVE DRIVE.
11. Datasheet Motores paso a paso.
12. Datasheet Transistores TIP 120.

PROCEDIMIENTO:

Para la programación del PIC utilizamos Microcode Studio.

1. Declaramos el puerto B como salidas.
2. Colocamos la variable x de 255
3. Le asignamos un nombre al programa.
4. Creamos valores con la secuencia **FOR** y **TO** que estos nos dará las veces que se repitiera el giro anti horario.
5. Utilizando la función **GOSUB** declaramos el tiempo que se va alimentar cada bobina (A, B, C, D) para realizar el giro.
6. Le damos una pausa al programa de 1 segundo.
7. Repetimos el programa de manera que el motor gire en 12 secuencias en sentido horario y de forma indefinida repita el proceso después de cada 5 segundos.

Cargamos el archivo **.HEX** generado por Microcode en el PIC 16F628A y alimentamos el modulo, lo cual el motor empezará a girar en un sentido y luego así otro.

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.

DESARROLLO:

El algoritmo correspondiente y programa correctamente elaborado en Microcode nos queda de la siguiente forma:

```

; MANEJO DE UN MOTOR PASO A PASO EN SECUENCIA WAVE DRIVE

trisb=0 ;hacer salidas el puerto b
x VAR BYTE ; variable x de 255
antihorario:
FOR x = 1 TO 12 ;12 veces repetir secuencia de giro antihorario
portb=%0001 ;energiza bobina A
GOSUB timer ;espera 5 mls
portb=%0010 ;energiza bobina C
GOSUB timer ;espera 5 mls
portb=%0100 ;energiza bobina B
GOSUB timer ;espera 5 mls
portb=%1000 ;energiza bobina D
GOSUB timer
NEXT
PAUSE 1000 ;espera 1 s

FOR x = 1 TO 12 ;12 secuencias para giro en sentido horario
portb=%1000 ;energiza bobina D
GOSUB timer ;espera 5 mls
portb=%0100 ;energiza bobina B
GOSUB timer ;espera 5 mls
portb=%0010 ;energiza bobina C
GOSUB timer ;espera 5 mls
portb=%0001 ;energiza bobina A
GOSUB timer ;espera 5 mls
NEXT
PAUSE 1000 ;espera 1 s
GOTO antihorario

timer:
PAUSE 5 ;pausa de 5 milisegundos
RETURN
END

```

PREGUNTAS DE CONTROL:

1. ¿Por qué es importante consultar el Datasheet del PIC?
2. ¿Qué importancia tiene el Watiaje de las resistencias al momento de armar el circuito?
3. ¿Qué consideraciones se debe tener para la selección de un PIC?
4. ¿En qué se diferencia un PIC de otro?
5. ¿Cuál es la diferencia entre un PIC y un Microprocesador?
6. ¿Cuál es la ventaja principal de un PIC y un Microprocesador?
7. ¿Cómo se conecta Motor paso a paso?
8. ¿Cómo funciona Motor paso a paso?
9. ¿Qué función cumple el TIP120?
10. ¿Qué función cumplen los Diodos?
11. ¿Por qué no utilizar un Microprocesador para ésta práctica?

CONCLUSIONES Y RECOMENDACIONES

CONCLUSIONES

- Con la realización de este proyecto se ha logrado obtener un módulo de prácticas para microcontroladores muy eficaz, que puede realizar distintas funciones de acuerdo a las necesidades de los estudiantes en el estudio de microcontroladores ya que cada práctica está diseñada para estudiar las distintas utilidades y funciones de los PIC's.
- Al utilizar un microcontrolador como el PIC 16F628A se ha podido potencializar al máximo las cualidades del módulo y de las prácticas planteadas, que brinda además la posibilidad de la utilización de cualquier otro microcontrolador o microprocesador.
- Dentro de un contexto académico, este proyecto se convierte en alternativas para el aprendizaje de la correcta programación e implementación de proyectos prácticos con microcontroladores.
- La utilización del microcontrolador PIC 16F628A 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.
- El resultado de las prácticas es 100% funcional ya que todas responden bien para lo que fueron diseñadas en el presente proyecto 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.

6.2. RECOMENDACIONES

- 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.
- En el módulo antes conectar la corriente y hacer funcionar los distintos circuitos que posee es necesario que se verifique las conexiones para evitar cortocircuitos y que los distintos elementos del módulo se quemen.

BIBLIOGRAFÍA

8.1. BIBLIOGRAFÍA:

LIBROS.

- BARNETT RICHARD, COX SARAH y O´CULL LARRY. (2003). **Embedded C Programing and the Atmel AVR**. Learning, Inc. Thomson.
- 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.
- REYES CARLOS A. (2008). **Microcontroladores PIC Programación en Basic**. RISPGRAF. Tercera Edición.
- TOCCI, RONALD J. y WIDMER, NEAL S. (2003). **SISTEMAS DIGITALES: Principios y Aplicaciones**. PEARSON EDUCACIÓN. México. Octava Edición.

SITIOS WEB DE INTERÉS.

- “Descarga manual de pbp, TEORÍA DEL PIC”, (en línea), Mar de Plata Argentina, 1999-2012 www.frino.com.ar
- “Descarga de programa lc-prog106A.zip”, (en línea) IC-Prog Prototype Programmer (Reino Unido), 06 sep 2008, www.IC-prog.com
- “Compilador PicBasic Pro, Ejemplos en pbp”, (en línea) micro Engineering Labs, Inc., (USA) 2011, www.melabs.com
- “Descarga de programas: Microcode y Pbp demo”, (en línea) MECANIQUE (Reino Unido), 2012, <http://www.mecanique.co.uk/>

- “Productos e información de la familia de PIC´S”, (en línea) MICROCHIP (Chandler, Arizona, USA), 1998-2012, www.microchip.com
- Microcontroladores genéricos, Autor Luis E. Sánchez Toledo, <http://www.monografias.com/trabajos34/microcontroladoresgenericos/microcontroladores-genericos.shtml>
- “Ejemplos en pbp, información de dispositivos”, (en línea), Reynolds Electronics, 1999-2011, www.rentron.com
- “Descarga manual en español de pbp”, (en línea), Pic Basic Pro (Argentina) 03 Jun 2001, www.todopic.com.ar
- “Información sobre LCD 2x16, motores PAP, etc.”, X-ROBOTICS, Barcelona (España), 2001-2008 www.x-robotics.com

ANEXOS

ANEXO A

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 B

FUNCIONAMIENTO DEL MODULO ENTRENADOR DE MICROCONTROLADORES



Fig. 55 Prueba de encendido del módulo.

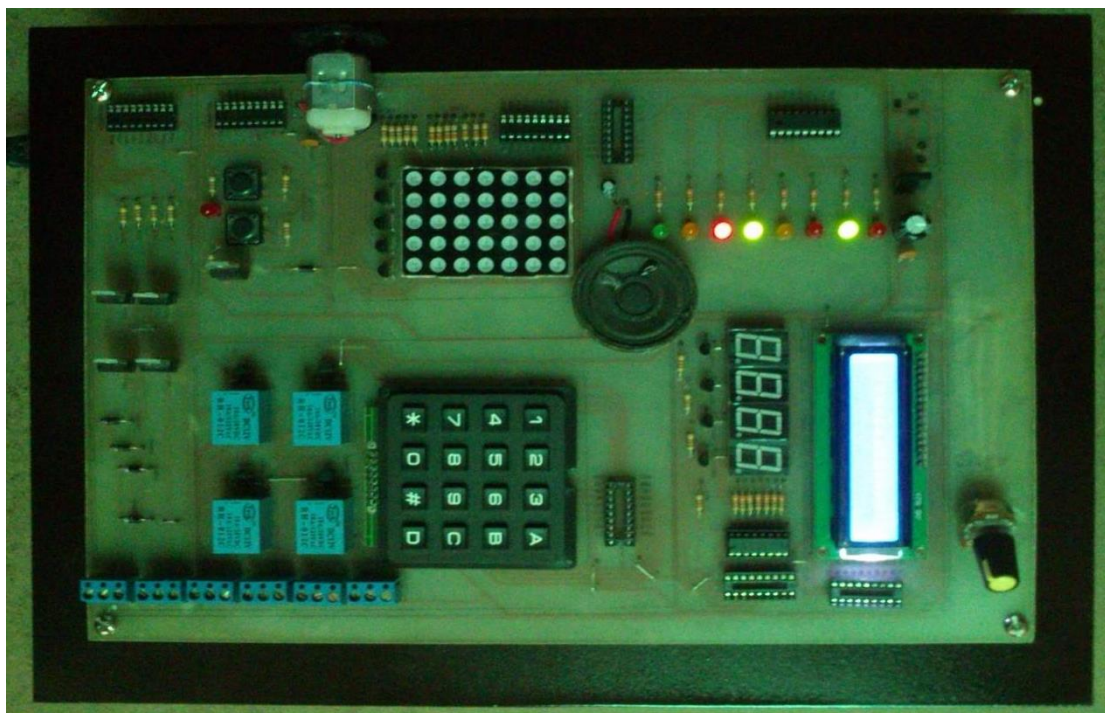


Fig. 56 Prueba de LED's y parlante.



Fig. 57 Prueba de Displays 7 segmentos.

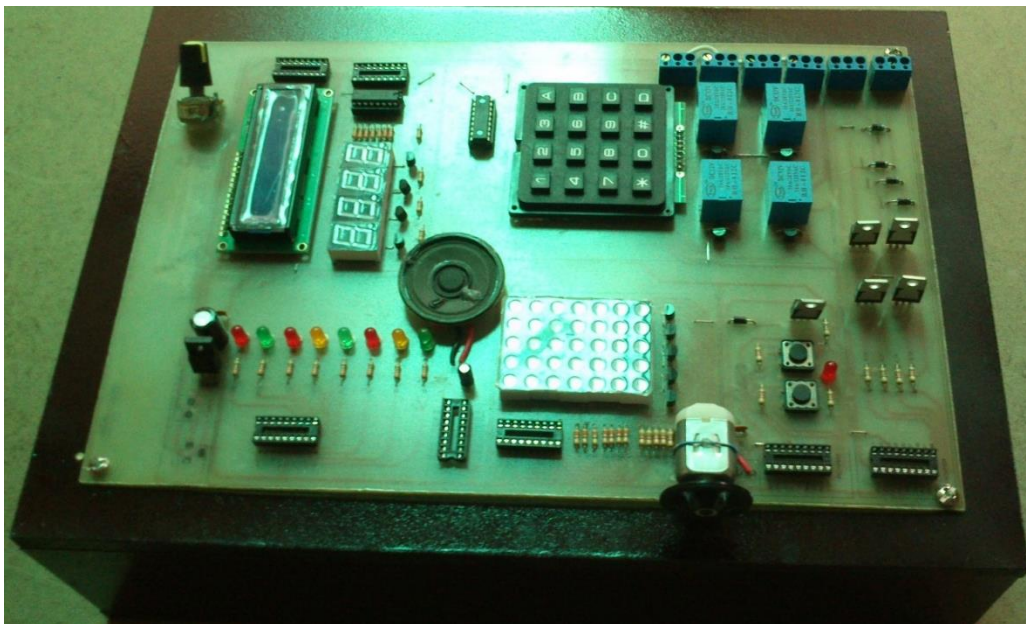


Fig. 58 Prueba de Teclado y motor DC.



Fig. 59 Prueba de LCD.

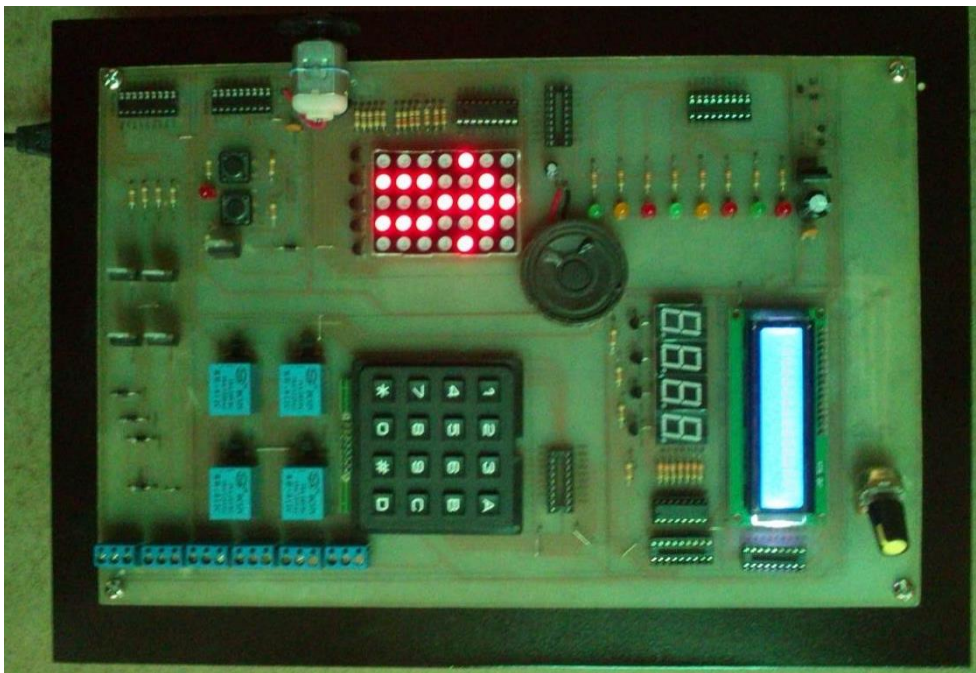


Fig. 60 Prueba de Matriz de LED's.



Fig. 61 Interior del Módulo.



Fig. 62 Interior del Módulo y Accesorio.

ANTEPROYECTO