



UNIVERSIDAD DE CARABOBO
FACULTAD DE INGENIERIA
ESCUELA DE INGENIERIA ELECTRICA
DEPARTAMENTO DE SISTEMAS Y AUTOMATICA



**DISEÑO DE UNA PLATAFORMA DIDACTICA PARA EL
LABORATORIO DE MICROPROCESADORES DE LA
UNIVERSIDAD DE CARABOBO.**

AUTOR:
Van Rooij D. Wijnand A.

TUTOR: Luis Llave.

Valencia, Noviembre de 2011



UNIVERSIDAD DE CARABOBO
FACULTAD DE INGENIERIA
ESCUELA DE INGENIERIA ELECTRICA
DEPARTAMENTO DE SISTEMAS Y AUTOMATICA



**DISEÑO DE UNA PLATAFORMA DIDACTICA PARA EL
LABORATORIO DE MICROPROCESADORES DE LA
UNIVERSIDAD DE CARABOBO.**

**TRABAJO ESPECIAL DE GRADO PRESENTADO ANTE LA
ILUSTRE UNIVERSIDAD DE CARABOBO PARA OPTAR AL
TITULO DE INGENIERO ELECTRICISTA**

AUTOR:
Van Rooij D. Wijnand A.

TUTOR: Luis Llave.

Valencia, Noviembre de 2011



RESUMEN



RESUMEN

El presente trabajo consiste en la elaboración de una plataforma didáctica para el laboratorio de Microprocesadores de la Universidad de Carabobo. Para ello el proyecto se dividió en cinco partes las cuales son:

El Planteamiento del problema, en el cual se plantea la problemática y la posible solución de la misma, tomando en cuenta el objetivo General y seis objetivos específicos que serán llevados a cabo a lo largo del proyecto. Un marco teórico en el cual se hace referencia al dispositivo principal como lo es el Microcontrolador, y los otros elementos tanto de hardware como de software que son usados. Un marco Metodológico en el cual se hace referencia a las fases que se llevaran a cabo para cumplir con los objetivos previamente planteados. Le sigue la elaboración de la plataforma didáctica que se ha planteado, bajo ciertos criterios como: Estudio de las prácticas que se realizan en el Laboratorio, los elementos de hardware que se utilizan a partir de las prácticas, un manual de uso entre otros.

Por último las conclusiones y las recomendaciones que se darán para tener un mejor uso de la plataforma planteada.

Palabras claves: Sistema de desarrollo, Microcontroladores, Programador, Perifericos.

UNIVERSIDAD DE CARABOBO
FACULTAD DE INGENIERIA
ESCUELA DE INGENIERIA ELECTRICA
DEPARTAMENTO DE SISTEMAS Y AUTOMATICA

CERTIFICADO DE APROBACIÓN

Los abajo firmantes miembros del jurado asignado para evaluar el trabajo especial de grado titulado “Diseño de una plataforma didáctica para el laboratorio de Microprocesadores de la Universidad de Carabobo” realizado por el bachiller Van Rooij D. Wijnand A., C.I N° 18.412.764, hacemos constar que hemos revisado y aprobado dicho trabajo.

Ing. Luis Llave
TUTOR

Ing. Wilmer Sanz
JURADO

Ing. Demetrio Rey Lago
JURADO

Valencia, Noviembre de 2011



DEDICATORIA



DEDICATORIA

Este trabajo en primer lugar está dedicado a Dios y los Santos, a mi Familia quienes me han ayudado incondicionalmente en lograr esta meta, a los profesores que día a día nos han preparado y a los compañeros de clase con quienes he pasado parte de mi carrera compartiendo con ellos.



INTRODUCCIÓN



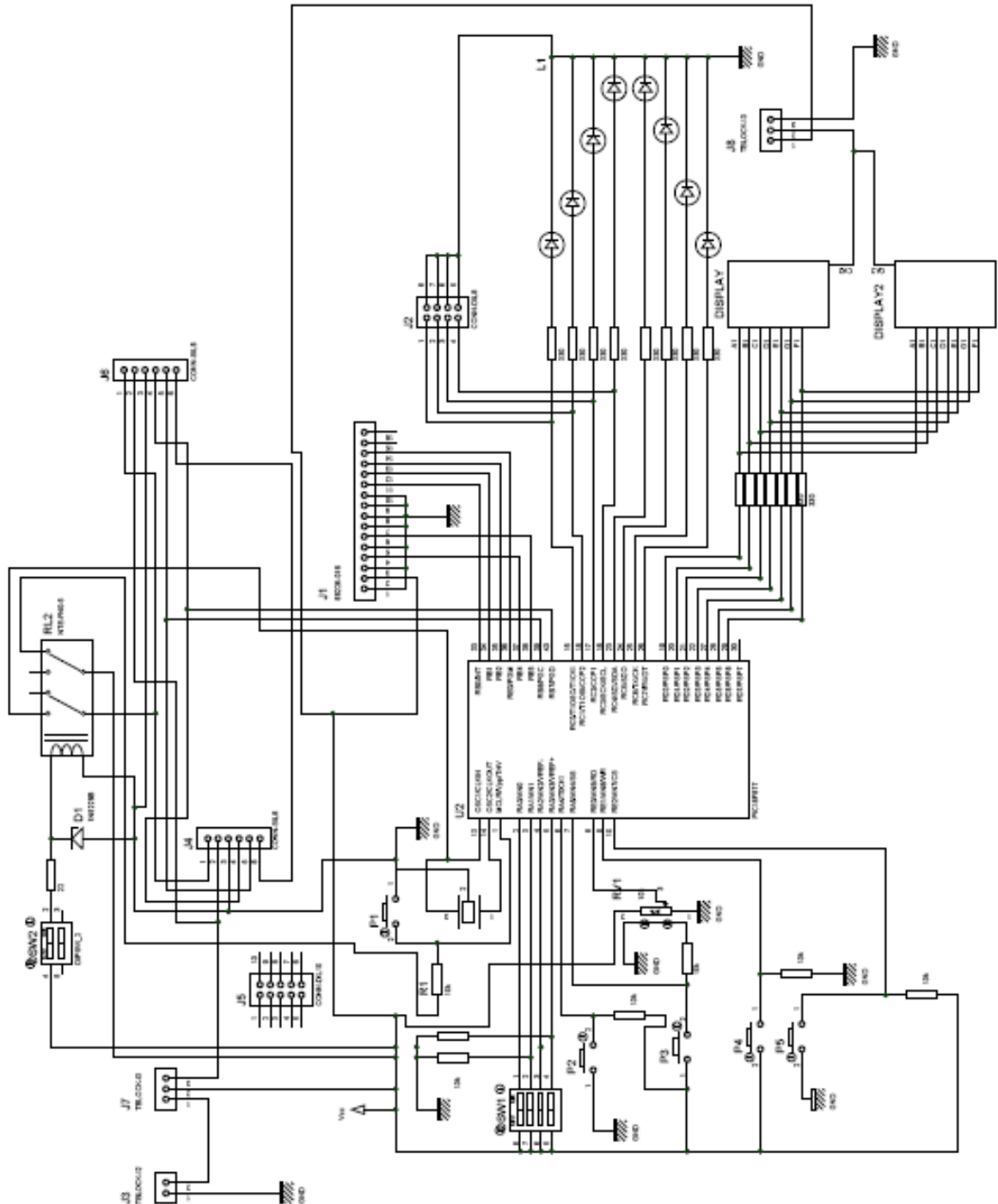
INTRODUCCIÓN

Los microprocesadores y microcontroladores son herramientas útiles que son utilizados en la vida diaria de un Ingeniero Electricista ya que ellos representan una opción considerable en la solución de un problema o para una aplicación en particular, además de ser de fácil uso.

Por esta razón se presenta esta propuesta que está enfocada al uso de una tarjeta que tenga un número de componentes necesarios para llevar a cabo las prácticas que se realizan en el laboratorio de Microprocesadores de la Universidad de Carabobo.

Los siguientes capítulos estarán enfocados en el desarrollo de una plataforma didáctica que será utilizada en el laboratorio de Microprocesadores. Para ellos se necesitarán de las prácticas que se realizan en la asignatura a parte de las características que cada una de ellas posee para así de esta manera elaborar una plataforma que tenga lo necesario para un buen trabajo.

Se utilizará el lenguaje de programación BASIC para realizar las pruebas, este es un lenguaje de alto nivel que ayuda a crear una programación mucho más sencilla sin tener las complejidades que puede tener el actual lenguaje ASSEMBLER. El microcontrolador a utilizar es el PIC 16F877A que posee ciertas características que lo hacen utilizable debido a que es de gama media y posee una gran cantidad de recursos que lo hacen una buena alternativa para elaboración de aplicaciones, que orientadas a la asignatura le abrirán al estudiante una buena opción para solución de problemas de una manera fácil.



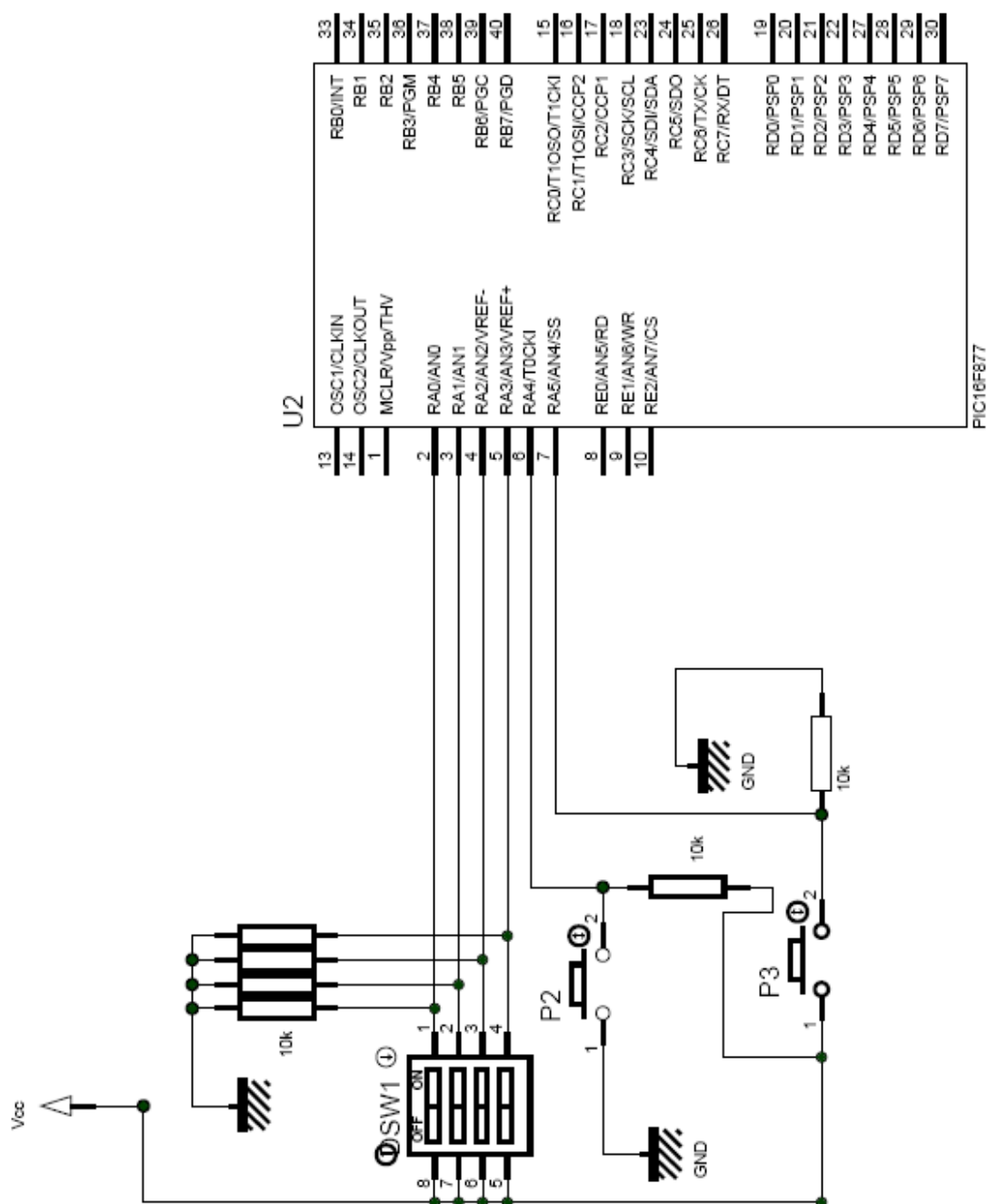


Figura 2: Esquemático de componentes del puerto A.

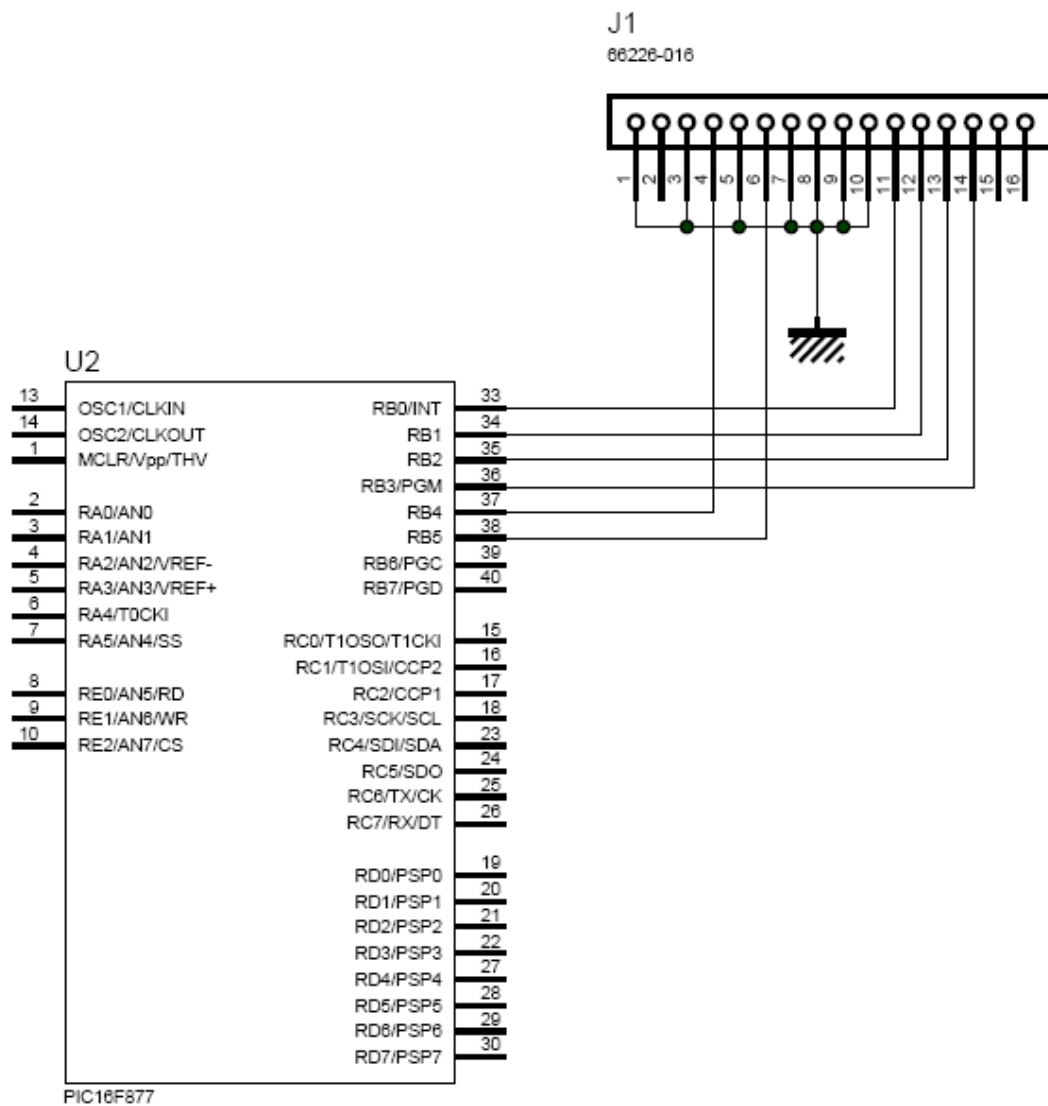


Figura 3: Esquemático de componentes del puesto B

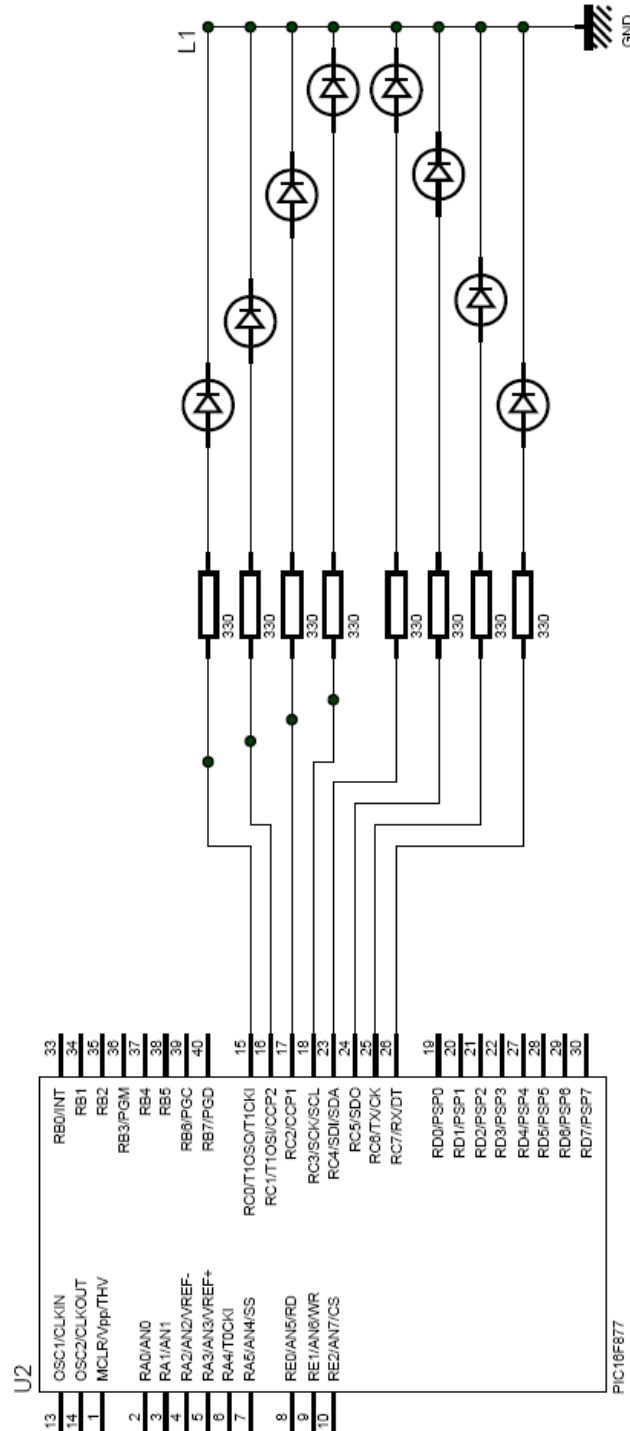


Figura 4: Esquemático de componentes del puesto C

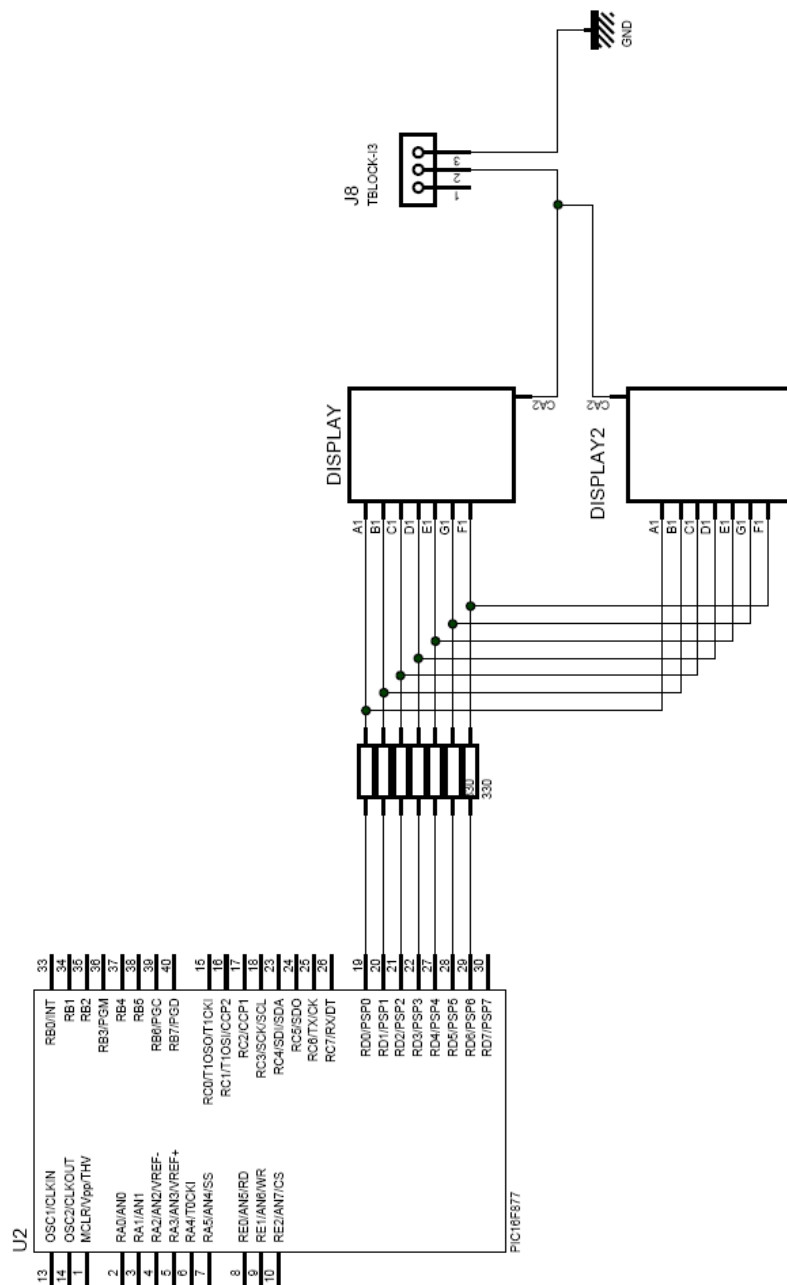


Figura 5: Esquemático de componentes del puesto D

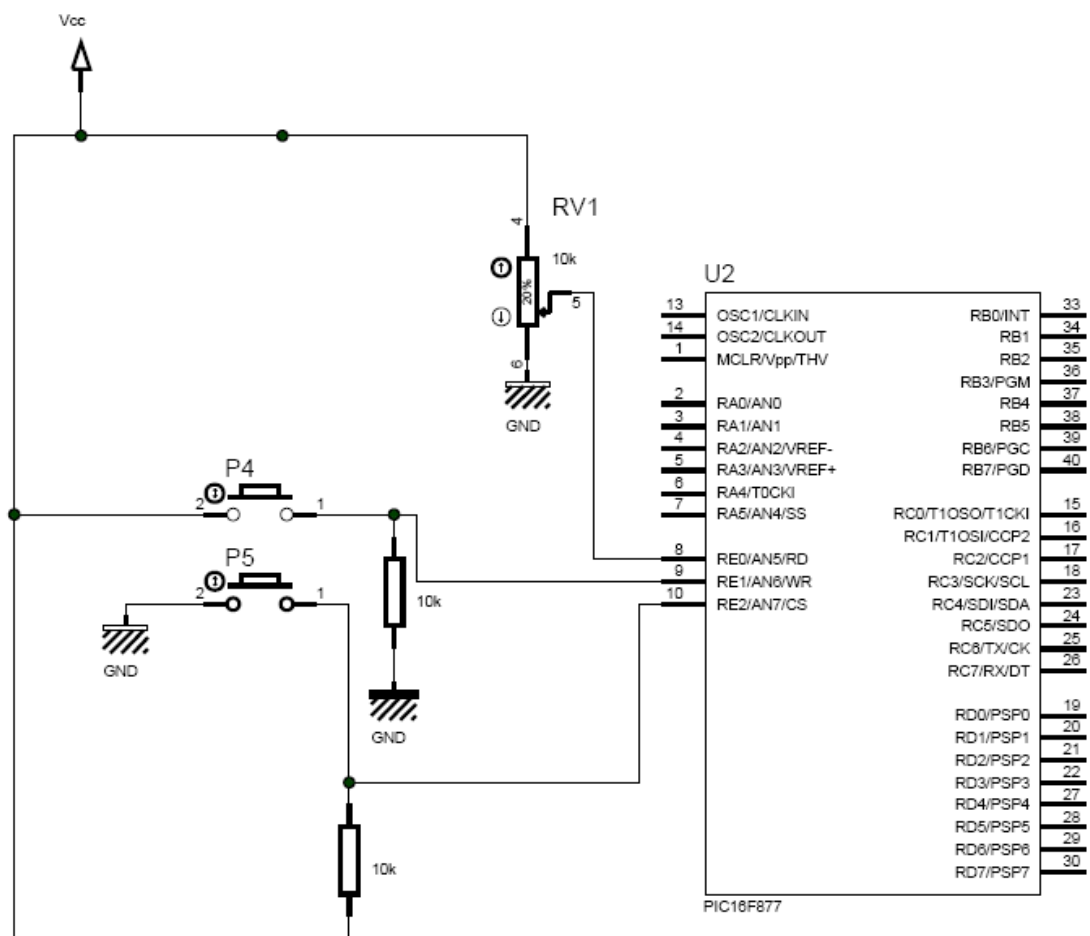


Figura 6: Esquemático de componentes del puesto E

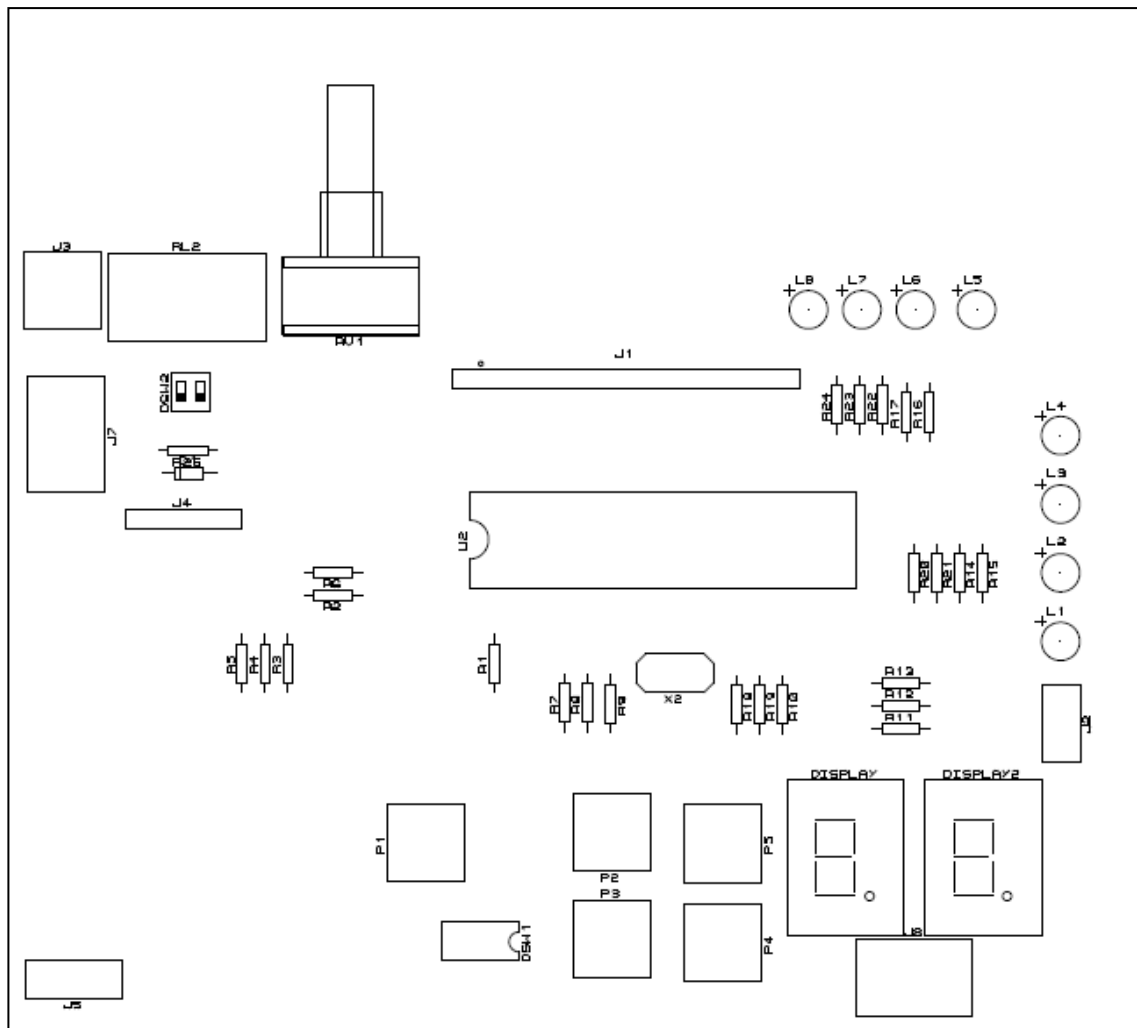


Figura 7: Esquema de componentes del sistema de desarrollo.

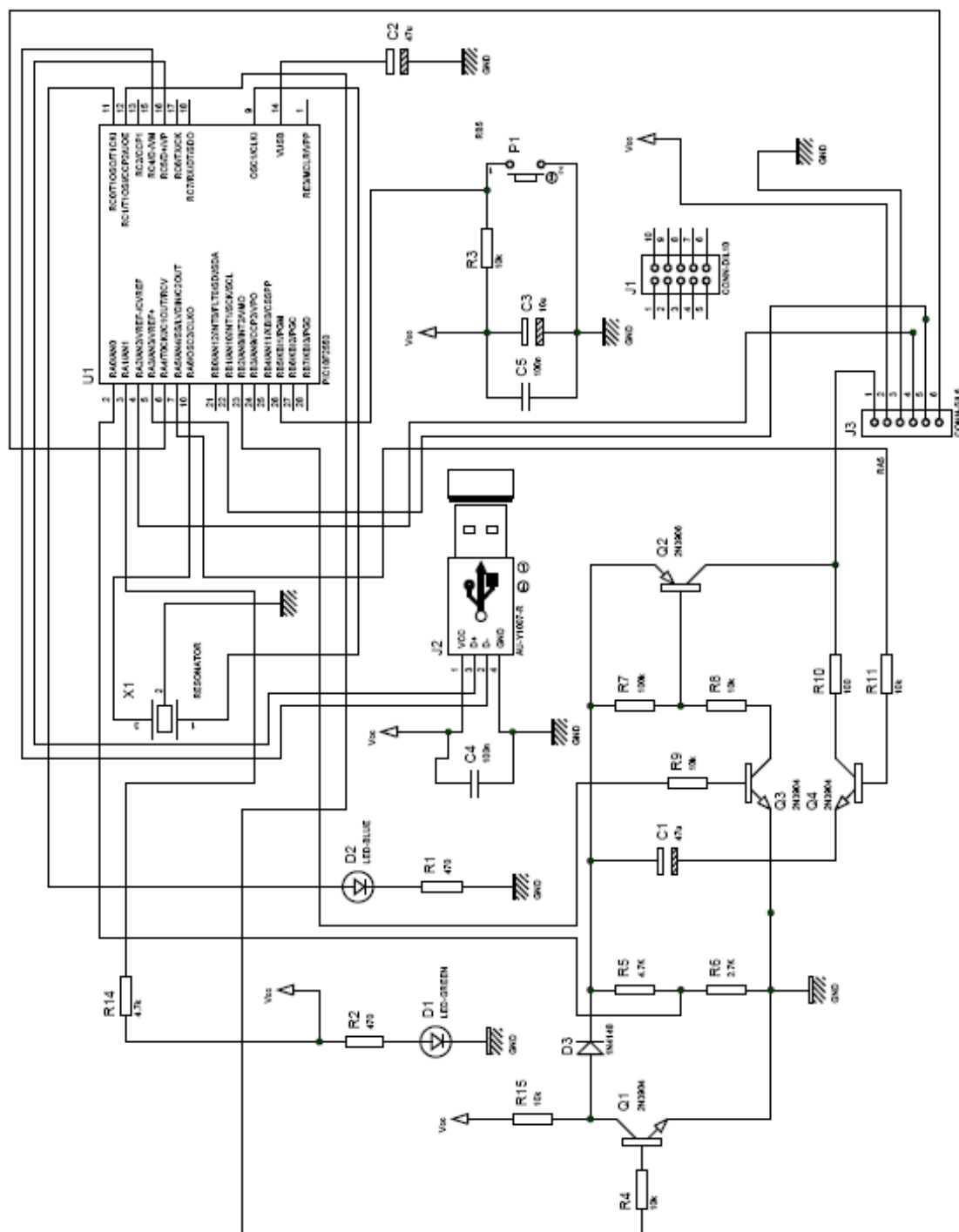


Figura 8: Esquemático de circuito de programación.

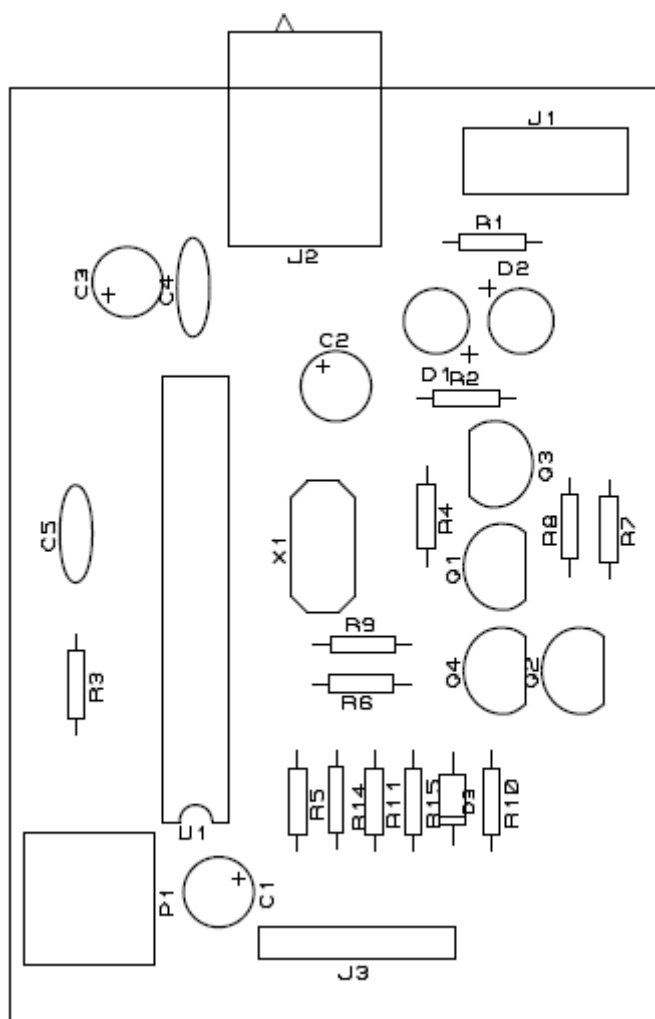


Figura 9: Esquema de componentes del programador.

Materiales	Cantidad	Precio	Total
Transistores 2n3904.	3	1.00	3.00
Transistores 2n3906.	1	1.00	1.00
Display 7 Seg.	2	15.00	30.00
Inductor de 680uH.	1	1.20	1.20
Lcd 16x2.	1	54.00	54.00
Capacitores cerámicos 100nF.	2	0.52	1.04
Capacitores electrolítico 47uF.	2	0.64	1.28
Capacitores electrolítico 10uF.	1	0.64	0.64
Pulsadores.	6	3.00	18.00
Interruptores.	1	7.45	7.45
Resonador de 20 MHz.	2	10.00	20.00
LEDs Varios.	10	2.00	20.00
Microcontrolador PIC 16F877A.	1	80.96	80.96
Microcontrolador PIC 18F2550.	1	79.80	79.80
Potenciómetro de 10K.	1	5.18	5.18
Conector USB tipo B.	1	7.37	7.37
Resistencias varias.	37	0.35	12.95
Relé de doble contacto a 3.3V.	1	15	15
Elaboración del circuito impreso 15 x 15 cm.	1	150	150
Elaboración del circuito impreso 7 x 5 cm.	1	28	28
Diodo 1n4148.	1	0.5	0.5
Diodo Zener 5005A 3.3V.	1	0.5	0.5
Base torneada para PIC16F877A.	1	12.88	12.88
Base torneada para PIC18F2550.	1	8.20	8.20
Interruptor de 3 pines.	2	5.20	10.40
Total Bs.			569.35

Figura 10: Relación de costos del sistema de desarrollo.



INDICE GENERAL

CAPITULO I: EL PROBLEMA	1
1.1 Planteamiento del Problema.	1
1.2 Justificación.	2
1.3 Objetivos y alcancé de la investigación.	3
1.3.1 Objetivos Generales.....	3
1.3.2 Objetivos Específicos.	3
1.4 Alcance y limitaciones.	4
 CAPITULO II: MARCO TEORICO	3
2.1 Antecedentes del Problema.....	5
2.2 Referentes Teóricos.	6
2.2.1 El microprocesador.....	7
2.2.2 El microcontrolador.	10
2.2.3 Componentes de un microcontrolador.....	12
2.2.4 Diferencia entre microprocesador y microcontrolador.....	13
2.2.5 Arquitectura del microcontrolador.	15
2.2.6 El procesador o Unidad Central de Procesamiento.	18
2.2.7 Memorias.	20
2.2.8 Memorias de datos.....	24
2.2.9 Líneas de Entradas y Salidas.	24
2.2.10 Reloj Principal.	25
2.2.11 Recursos Especiales.....	26
2.3 Programación de Microcontroladores.....	35
2.4 Compilador.	36
2.5 Simulador.....	36



INDICE GENERAL



2.6	Circuitos impresos.	36
CAPITULO III: MARCO METODOLOGICO		38
3.1	Tipo y diseño de la investigación.	38
3.2	Fases de la investigación:	38
3.2.1	Fase 1: Análisis de las prácticas.	38
3.2.2	Fase 2: Diseño de la plataforma.	39
3.2.3	Fase 3: Elaboración de la plataforma.	39
3.2.4	Fase 4: Implementación del dispositivo en hardware.....	39
3.2.5	Fase 5: Validación del dispositivo usando las prácticas existentes.....	39
3.2.6	Fase 6: Elaboración de documentación.	40
CAPITULO IV: DESARROLLO DE LA PLATAFORMA DIDÀCTICA:.....		41
4.1	Análisis de las prácticas:.....	41
4.1.1	Práctica N° 1:	42
4.1.1.1	Objetivos:.....	42
4.1.1.1.1	Objetivos Generales:	43
4.1.1.1.2	Objetivos Específicos:.....	43
4.1.1.2	Desarrollo de la práctica:	43
4.1.1.3	LABORATORIO Semana 2	45
4.1.1.4	Fundamento teórico:	45
4.1.1.4.1	Interruptores y pulsadores:	45
4.1.1.4.2	Diodos LED:	48
4.1.2	Práctica N°2:	49
4.1.2.1	Objetivos:.....	49
4.1.2.1.1	Objetivos Generales:	49
4.1.2.1.2	Objetivos Específicos:.....	50



INDICE GENERAL



4.1.2.2	LABORATORIO Semana 1	50
4.1.2.3	LABORATORIO Semana 2	50
4.1.2.4	Fundamento Teórico:	51
4.1.2.4.1	Display 7 Segmentos:	51
4.1.2.4.1.1	El display ánodo común:	51
4.1.2.4.1.2	El display cátodo común:	52
4.1.2.4.2	Rebote:	52
4.1.3	Práctica N° 3:	53
4.1.3.1	Objetivos:	53
4.1.3.1.1	Objetivos Generales:	53
4.1.3.1.2	Objetivos Específicos:	54
4.1.3.2	LABORATORIO.....	54
4.1.4	Práctica N° 4:	56
4.1.4.1	Objetivos:	56
4.1.4.1.1	Objetivos Generales:	56
4.1.4.1.2	Objetivos Específicos:	57
4.1.4.2	LABORATORIO Semana 1:	57
4.1.4.3	PRELABORATORIO Semana 2:	58
4.1.4.4	LABORATORIO Semana 2:	58
4.1.4.5	Fundamento teórico:	59
4.1.4.5.1	Display de cristal líquido (LCD):	59
4.2	Diseño de la plataforma:	60
4.2.1	Elementos utilizados:	60
4.2.2	Programa a utilizar para el diseño del circuito:	61
4.2.3	Diseño del programador a utilizar:	61
4.2.3.1	Programación a bajo voltaje:	62
4.2.3.2	Programación a alto voltaje:	62



INDICE GENERAL



4.2.3.3	Diagrama del programador:	63
4.2.4	Diseño de la plataforma:	64
4.2.4.1	Valores de resistencias a utilizar:	64
4.2.4.2	Ubicación de los componentes a utilizar:	66
4.2.4.3	Esquemático de la plataforma a utilizar:	67
4.3	Diseño del sistema de desarrollo en pcb:	68
4.3.1	Programa a utilizar:	68
4.3.1.1	Forma manual:	68
4.3.1.2	Forma automática:	69
4.3.1.2.1	Método con autorouter:	69
4.3.1.2.2	Método Electra autorouter:	69
4.3.2	Materiales a utilizar:	70
4.3.3	Diagrama en PCB:	71
4.4	Montaje en físico del sistema de desarrollo:	76
4.5	Manual de usuario:	78
4.5.1	Interfaz de programación:	79
4.5.2	Fuente de alimentación:	79
4.5.3	Periféricos de E/S:	80
4.5.4	Como usar el Sistema de Desarrollo:	81
4.5.4.1	Usando el Software Pickit 2.....	84
4.5.4.2	Usando el modo debugger del MPLAB.....	84
4.5.5	Descripción del microcontrolador PIC 16f877:	88
4.5.5.1	Descripción del dispositivo:	89
4.5.5.2	Organización de la memoria del programa:	90
4.5.5.3	Diagrama de conexionado:	91
4.5.5.4	Periféricos de entrada y salidas del pic 16f877:	93
4.5.5.4.1	Temporizador timer 0:	93



INDICE GENERAL



4.5.5.4.2	Temporizador timer 1:.....	95
4.5.5.4.2.1	Operación del Timer1 en modo Temporizador:.....	96
4.5.5.4.2.2	Timer1 en modo Contador Síncrono:.....	96
4.5.5.4.2.3	Timer1 en modo Contador Asíncrono:	97
4.5.5.4.3	Temporizador timer 2:.....	97
4.5.5.4.4	Conversor A/D:	98
4.5.5.4.5	Puerto de comunicación serial síncrona/asíncrona:	103
4.5.5.4.6	Los módulos de ccp (captura, Comparación, PWM):	104
4.5.5.4.6.1	El Módulo CCP1:.....	106
4.5.5.4.6.2	El Módulo CCP2:.....	106
4.5.5.4.6.3	El modulo PWM:	106
4.6	Costo de elaboración del Sistema de Desarrollo:	110
CAPITULO V: CONCLUSIONES.....		111
RECOMENDACIONES.....		113
BIBLIOGRAFIAS		114
ANEXOS.....		115



CAPITULO I: EL PROBLEMA

1.1 Planteamiento del Problema.

Debido a que el laboratorio de microprocesadores cuenta con sólo dos horas académicas en las que cada una representa cincuenta minutos, el estudiante tiene que realizar la práctica en una hora y cuarenta minutos, lo que hace que deba trabajar rápidamente entre la realización y la breve introducción que realiza el profesor al inicio de la clase. Actualmente las prácticas que se realizan en el laboratorio de microprocesadores no requieren de un montaje completo ya que el laboratorio cuenta con un sistema de desarrollo hecho en un protoboard. Las características del sistema que se usa en el laboratorio es tal que en las prácticas se tiene que hacer el montaje de los dispositivos y en algunos casos cablear algunas pistas para poder realizar las pruebas, esto hace que el estudiante tenga que estar manipulando los dispositivos y puedan haber errores en las pruebas.

Una solución para evitar los problemas anteriores, como lo es el retraso de tiempo de entrega de las prácticas, es utilizar un sistema de desarrollo que ya tenga los elementos necesarios para realizar una experiencia práctica con microcontroladores.

Una gran cantidad de sistemas de desarrollo se utilizan para la enseñanza de los microcontroladores. El problema radica en encontrar un sistema que se adapte a las exigencias de la materia Microprocesadores de la escuela de Ingeniería Eléctrica de la Universidad de Carabobo. El sistema además debe considerar la limitación de equipos, mantenimiento y adquisición de los mismos.



En este proyecto se diseñara un sistema de desarrollo que sea útil para la asignatura y cuyos detalles de diseño estarán disponibles en la Escuela de Eléctrica a fin de poder implementar las tarjetas que el laboratorio requiera, evitando así la necesidad de comprar sistemas de desarrollo comerciales que pueden resultar costosos.

1.2 Justificación.

Los microcontroladores hoy en día se pueden conseguir prácticamente en cualquier aparato electrónico, por lo que su estudio es importante; lo que hace que mantengamos un laboratorio con las diferentes prácticas que nos ayudan a entender cómo funcionan y la manera en cómo ellos pueden ser programados para realizar las diferentes tareas o aplicación que se le quiera dar.

Este trabajo será realizado para la materia microprocesadores con el objetivo de realizar las prácticas de una manera más sencilla.

Los beneficios que se obtendrán con el presente trabajo serán los siguientes:

- Se evitarán errores al momento del montaje debido a que los elementos estarán ya soldados a la placa evitando así los falsos contactos que se pueden presentar al realizar las prácticas.
- Es sencillo de manejar ya que después de programar se podrá ejecutar utilizando los puertos seleccionados por el usuario para tal fin.
- Su precio y mantenimiento es económico ya que constituye un diseño propio.



1.3 Objetivos y alcancé de la investigación.

1.3.1 Objetivos Generales.

Diseñar una plataforma didáctica para el laboratorio de microprocesadores de la Universidad de Carabobo.

1.3.2 Objetivos Específicos.

- Analizar las prácticas que se realizan en el laboratorio de microprocesadores para así identificar los elementos necesarios para la elaboración del proyecto.
- Diseñar el modelo del circuito que cumpla con los requerimientos del Laboratorio de Microprocesadores.
- Elaborar el diseño del circuito con todas las conexiones necesarias para su posterior montaje en baquelita.
- Implementar el dispositivo en hardware.
- Validar el dispositivo usando prácticas existentes.
- Elaborar la documentación necesaria para su adecuado funcionamiento y para futuras implementaciones.



1.4 Alcance y limitaciones.

En el presente trabajo se considera desarrollar una herramienta de trabajo que facilitará del estudiante, teniendo en cuenta igualmente todos los instrumentos con los que se cuentan actualmente en uso en el laboratorio de microprocesadores.

La plataforma tendrá las siguientes características:

- Canales de entradas para señales analógicas de 0 a 5V.
- Interfaz de comunicación con la PC vía USB.
- Un manual de funcionamiento del sistema.
- Dispondrá de elementos de hardware tales como Leds, pulsadores, display 7 segmentos, pantalla Lcd.



CAPITULO II: MARCO TEORICO

2.1 Antecedentes del Problema.

Hoy en día muchos de los equipos que conseguimos ya no están basados en circuitos tan complejos como hace algunos años sino que vienen limitados a simples circuiterías definidas por el fabricante esto gracias a el desarrollo de los Microcontroladores, ya que en un solo chip y mediante la programación definida por el fabricante este puede hacer la tarea que se quiera haciendo cálculos complejos, almacenando y procesando información permitiendo así la intercomunicación con el ser humano.

Si bien estos dispositivos se han hecho cada vez más potentes en cuanto a la cantidad de almacenamiento de datos, rapidez y funcionalidad, así mismo se han hecho más pequeños haciendo que cada vez los instrumentos que hacen uso de ellos tengan un menor tamaño.

Los microcontroladores están siendo empleados en multitud de sistemas presentes en nuestra vida diaria, como pueden ser juguetes, horno microondas, frigoríficos, televisores, computadoras, impresoras, módems y otras aplicaciones con las que seguramente no estaremos tan familiarizadas como instrumentación electrónica, control de sistemas, entre otras.

Algunos temas relacionados que pueden servir como base a este proyecto son los siguientes:

- Morillo Gómez, José Gregorio. Mayo 2006. **“Diseño e implementación de prácticas para aplicaciones avanzadas en Diseño Digital usando tarjetas**



- **de desarrollos PICDEM, programador MPLAB ICD2 y lenguaje de programación BASIC**". Trabajo de grado para optar el título de Ingeniero Electricista. Universidad de Carabobo. Valencia, Venezuela.

En la realización de esta investigación se puede apreciar la importancia de la utilización de entornos de desarrollos con microcontroladores en la ejecución de prácticas de laboratorio de Diseño Digital.

- Lemus Henry, Istillarte Carlos. Diciembre 2006. **“Desarrollo de un plan de prácticas para el Laboratorio de la asignatura Microprocesadores, dictada en la escuela de Ingeniería eléctrica de la Universidad de Carabobo utilizando microcontroladores”**. Valencia, Venezuela.

En esta investigación se propone la realización de una serie de prácticas para el Laboratorio de Microprocesadores brindando así la información necesaria para cubrir los objetivos principales de cada práctica.

2.2 Referentes Teóricos.

En un principio cuando no existían los microcontroladores se diseñaban circuitos electrónicos que implicaban en muchos casos gran cantidad de componentes electrónicos y cálculos matemáticos que en ocasiones desembocaban en circuitos con fallos debido a la lógica utilizada.

El desarrollo de la electrónica de estado sólido la revolución comenzó con la invención del transistor de silicio en el año 1947 en manos de la compañía Bell Telephone por los científicos John Bardeen, Walter Brattain y William Shockley siendo el transistor bipolar el que marcaría de esta manera el desarrollo de dispositivos sobre todo en la constitución de los primeros ordenadores digitales que requerían un gran número de



interruptores pequeños y de baja potencia que simplemente el transistor bipolar cumplía con esas características lo que le abrió un buen mercado en su uso.

Posteriormente a esto se fueron añadiendo a esto una gran gama de dispositivos electrónicos como los diodos, transistores de unión, MOSFET entre otros que a su vez generaba el nacimiento de los circuitos integrados y con ello un mundo de posibilidades tecnológicas.

El paso de la electrónica a la microelectrónica supuso un gran paso debido a la transición de varios componentes en solo circuito integrado compuesto solamente por un chip. Con esta técnica de reducción se lograba una mayor complejidad de circuitos, menores dimensiones y mayor eficiencia a un costo de fabricación más reducido.

2.2.1 El microprocesador.

El microprocesador o CPU es un dispositivo electrónico digital, integrado, programable y de actuación secuencial que constituye el cerebro del computador. Funcionalmente, es pues un dispositivo lógico, que asociado a otras unidades, permite el tratamiento de la información almacenada, en forma de "programa de instrucciones", siendo capaz de interpretar estas instrucciones y ejecutarlas controlando a las unidades implicadas en su realización.

Este dispositivo controla la memoria y las operaciones de E/S a través de una serie de conexiones llamadas buses que son los encargados de enviar los datos entre los componentes, estos son formados por cables o pistas dentro de los circuitos impresos. En la figura 2.1 se puede apreciar la configuración de pines del microprocesador INTEL 8088.



- Un bus de direccionamiento, que puede ser de 8, 16 o 32 bits, y que lleva este direccionamiento a la memoria.
- Un bus de datos, que puede ser de los mismos bits anteriormente mencionados, que puede enviar datos a la memoria y recibir datos de la memoria.
- Una línea de lectura (RD) y otra de escritura (WR) para decirle a la memoria si quiere configurar o localizar el direccionamiento.
- Una línea para el reloj que envía pulsos en secuencia al procesador.
- Una línea para resetear el contador del programa a cero y reiniciar la ejecución.

La mayoría de los microprocesadores son de arquitectura del tipo CISC (Computadoras de conjuntos complejos de instrucciones) éstas poseen un conjunto de instrucciones que se caracterizan por ser muy amplio y permitir operaciones complejas entre operando situados en la memoria o registros internos. Este tipo de arquitectura dificulta el paralelismo entre instrucciones, por lo que, en la actualidad, la mayoría de los sistemas CISC de alto rendimiento implementan un sistema que convierte dichas instrucciones complejas en varias instrucciones simples del tipo RISC, llamadas generalmente microinstrucciones. En la figura 2.2 se encuentra el diagrama de bloques del Microprocesador INTEL 8088.

El microprocesador es un dispositivo de arquitectura abierta, ya que posee conexiones externas de todos sus buses, tanto el de datos como el de direcciones y control. Esto permite que se puedan ir añadiendo circuitos externos a medida que se vayan necesitando como pueden ser memorias, dispositivos de E/S, etc. Por lo que el sistema se puede expandir tanto como uno lo necesite para la aplicación a la que se quiera utilizar.

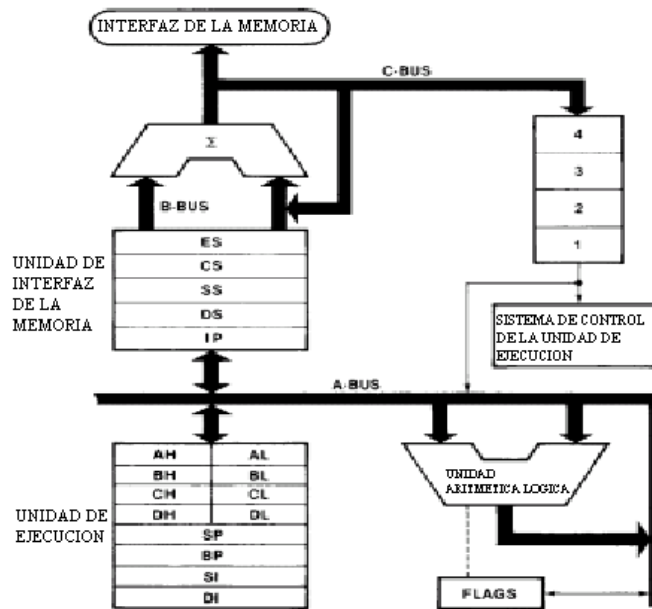


Figura 2.2 Diagrama de bloques del Microprocesador INTEL 8088.

Fuente: Hoja de especificaciones INTEL 8088.

2.2.2 El microcontrolador.

Es un circuito integrado programable que contiene todos los componentes de un computador. Se emplea para controlar el funcionamiento de una tarea determinada y, debido a su reducido tamaño, suele ir incorporado en el propio dispositivo al que gobierna. Esta última característica es la que le confiere la denominación de “controlador incrustado”.

El microcontrolador es un computador dedicado aunque de limitadas prestaciones, que esta contenido en el chip de un circuito integrado, que posee todos los componentes necesarios para realizar una aplicación, esto facilita el trabajo con ellos pero al mismo tiempo dificulta la expansión del sistema al ponerlo a trabajar. Esto implica que dependiendo de la aplicación que se vaya a dar a estos va a haber un microcontrolador de

diferente gama que va ofrecer ciertas características como puede ser desde el manejo de entradas – salidas (los más sencillos) hasta los más poderosos que puede incluir conversores analógicos – digital, modulación de ancho de pulsos, manejos de interrupciones, entre otras. En la figura 2.3 se encuentra el diagrama de pines de un microcontrolador PIC16F84A.

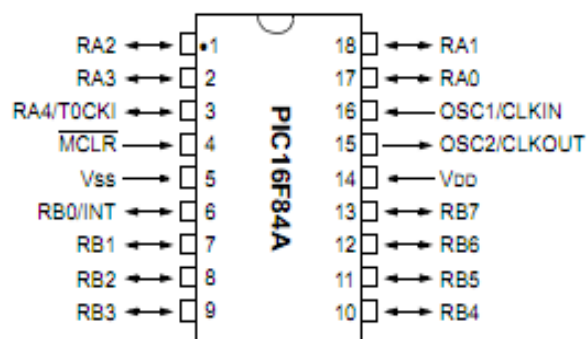


Figura 2.3 Diagrama de pines de un microcontrolador PIC16F84A.

Fuente: Hoja de especificaciones Microchip PIC16F84A.

En la memoria del microcontrolador sólo reside un programa destinado a gobernar una aplicación determinada; sus líneas de entrada/salida soportan el conexionado de los sensores y actuadores del dispositivo a controlar, y todos los recursos complementarios disponibles tienen como única finalidad atender sus requerimientos. Una vez programado y configurado el microcontrolador solamente sirve para gobernar la tarea asignada.

Un microcontrolador es un dispositivo de arquitectura cerrada lo que quiere decir que todos los buses son internos. Todas las partes del procesador están conectadas en su interior y solo salen al exterior las líneas que gobiernan los periféricos 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. En la figura 2.4 Arquitectura cerrada del microcontrolador.

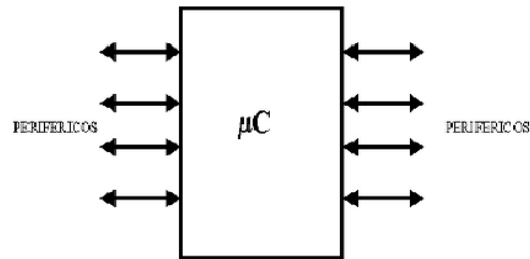


Figura 2.4 Arquitectura cerrada del microcontrolador.

Fuente: Microcontroladores PIC. Ignacio Angulo Martinez.

2.2.3 Componentes de un microcontrolador.

Un microcontrolador dispone normalmente de los siguientes componentes:

- Procesador (Unidad Central de Procesamiento).
- Memoria RAM para contener los datos.
- Memoria para el programa ROM/PROM/EPROM.
- Líneas de E/S para comunicarse con el exterior.
- Diversos módulos para el control de periféricos (temporizadores, puertos seriales, puertos USB, Conversores analógicos – digitales, entre otros).
- Generadores de impulsos de reloj que sincronizan todo el funcionamiento de todo el sistema.
- Recursos auxiliares.

En la figura 2.5 se puede apreciar el diagrama de bloques interno de un microcontrolador PIC16F84A en la cual se notan las partes principales de un microcontrolador.

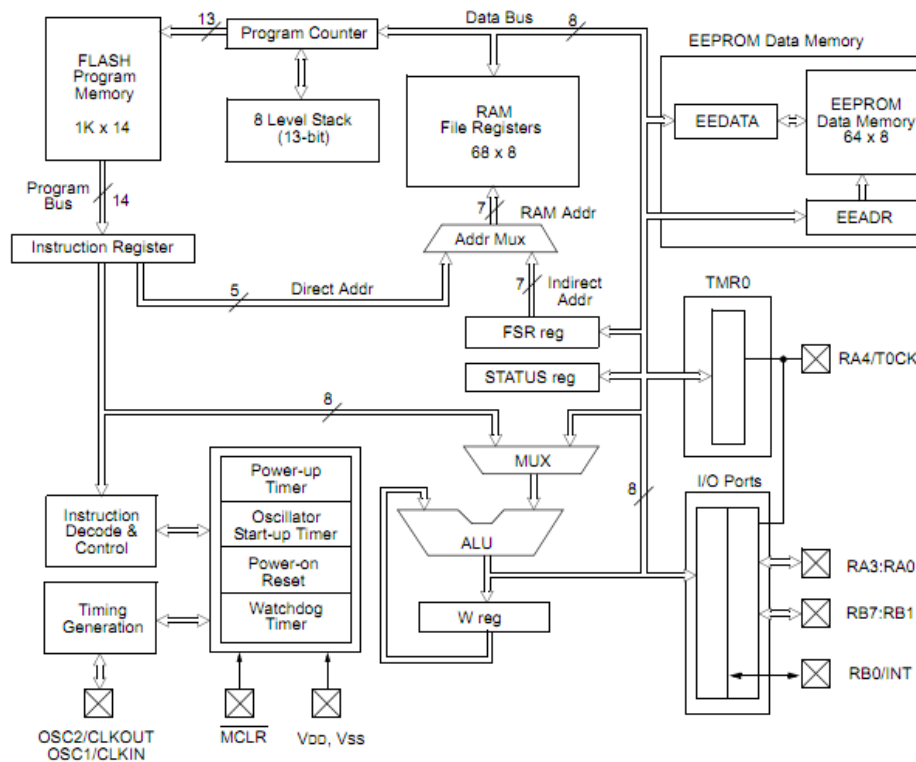


Figura 2.5 Diagrama de bloques interno de un microcontrolador PIC16F84A.

Fuente: Hoja de especificaciones Microchip PIC16F84A.

2.2.4 Diferencia entre microprocesador y microcontrolador.

El microcontrolador tiene su base en el microprocesador que encontramos en una computadora, a diferencia de que el último está diseñado para propósito general, mientras que el microcontrolador se utiliza para tareas más específicas. El microprocesador se define como Unidad Central de Proceso (CPU) en un sólo chip y está diseñado para leer y ejecutar instrucciones almacenadas en memoria. Las principales diferencias entre ambos son las siguientes:

- Los movimientos de los datos en el microprocesador son externos. En el microcontrolador los datos se mueven dentro del chip.

- Los microprocesadores son de propósito general mientras que los microcontroladores tienen un uso específico.
- El microprocesador necesita de componentes adicionales para poder operar. Los microcontroladores pueden operar sin necesidad de agregar dispositivos adicionales.
- El microprocesador posee pines de E/S fijos mientras que en el microcontrolador son programables.
- El microprocesador puede expandir sus capacidades al agregar más dispositivos. El microcontrolador tiene reducida la expansión de sus capacidades.

En la figura 2.6 se aprecia la estructura de un sistema de lazo abierto de un microprocesador.

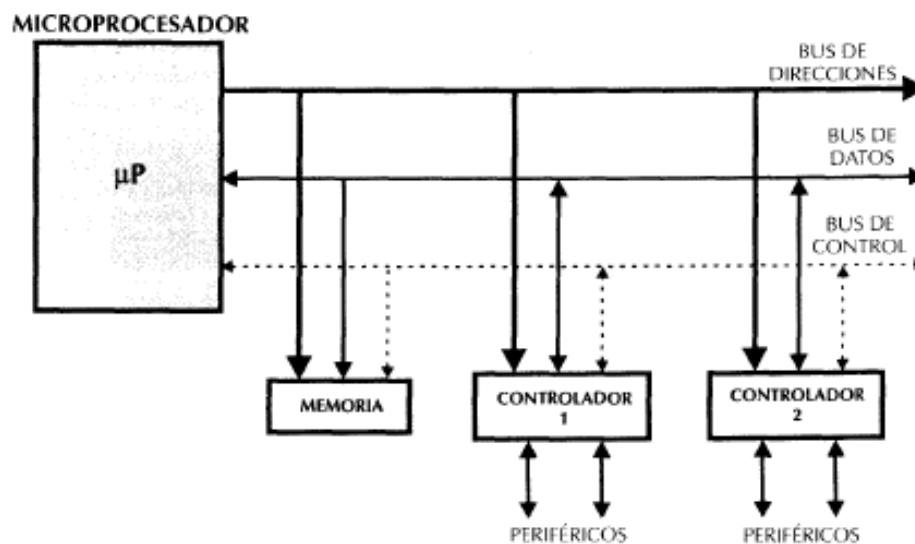


Figura 2.6 Estructura de un sistema de lazo abierto de un microcontrolador. Los buses en el exterior permiten que se configure dependiendo de la aplicación.

Fuente: Microprocesador PIC. Ignacio Angulo Martinez.



2.2.5 Arquitectura del microcontrolador.

Es el diseño conceptual y la estructura operacional fundamental de un sistema de computadora. Es decir, es un modelo y una descripción funcional de los requerimientos y las implementaciones de diseño para varias partes de una computadora, con especial interés en la forma en que la unidad central de procesamiento (UCP) trabaja internamente y accede a las direcciones de memoria.

El ordenador recibe y envía la información a través de los periféricos por medio de los canales. La UCP es la encargada de procesar la información que le llega al ordenador. El intercambio de información se tiene que hacer con los periféricos y la UCP. Todas aquellas unidades de un sistema exceptuando la UCP se denomina periférico, por lo que el ordenador tiene dos partes bien diferenciadas, que son: la UCP (encargada de ejecutar programas y que esta compuesta por la memoria principal, la UAL y la UC) y los periféricos (que pueden ser de entrada, salida, entrada-salida y comunicaciones).

Básicamente existen dos arquitecturas de computadoras, y por supuesto, están presentes en el mundo de los microcontroladores: Von Neumann y Harvard. Ambas se diferencian en la forma de conexión de la memoria al procesador y en los buses que cada una necesita.

2.2.5.1 Arquitectura Von Neumann.

La arquitectura tradicional de computadoras y microprocesadores en la cual la unidad central de proceso (UCP), 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. En la figura 2.7 se representa las unidades funcionales que constituyen la arquitectura Von Neumann.

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. Las principales limitaciones de este tipo de arquitectura son:

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

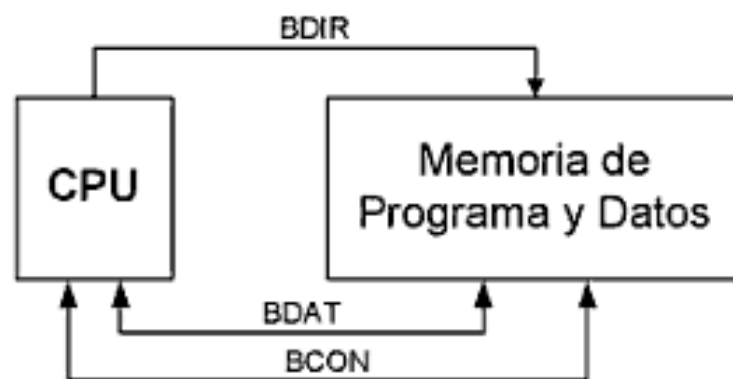


Figura 2.7 Unidad funcional que constituye la arquitectura Von Neumann.

Fuente: Microcontroladores: Fundamentos y Aplicaciones. Ramón Pallás Areny.



2.2.5.2 Arquitectura Harvard

Esta arquitectura 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 solo almacena datos (Memoria de Datos). Ambos buses son totalmente independientes y pueden ser de distintos anchos. Para un procesador de Set de Instrucciones Reducido, 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.

Ventajas de esta arquitectura:

- El tamaño de las instrucciones no esta 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 2.8 se representa la unidad funcional que constituye la arquitectura Harvard.

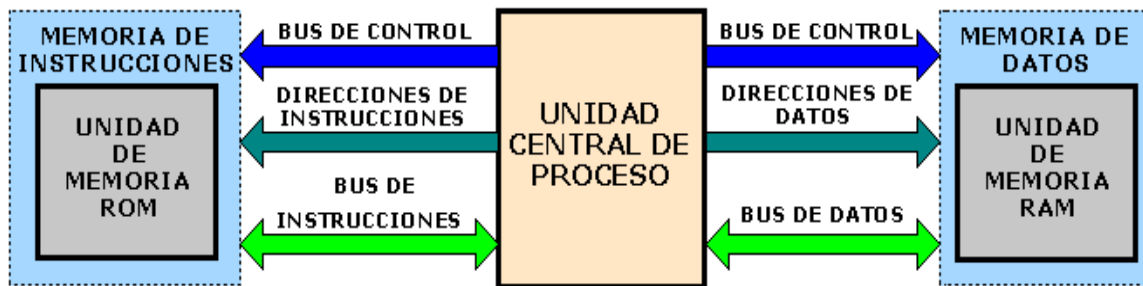


Figura 2.8 Unidad funcional que constituye la arquitectura Harvard.

Fuente: Dispositivos lógicos programable. <http://perso.wanadoo.es/pictob/micropic.htm>

2.2.6 El procesador o Unidad Central de Procesamiento.

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

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

Una de las primeras decisiones a la hora de diseñar un microprocesador es decidir cual será su juego de instrucciones. La decisión es trascendente por dos razones; primero, el juego de instrucciones decide el diseño físico del conjunto; segundo, cualquier operación que deba ejecutarse en el microprocesador deberá poder ser descrita en términos de un lenguaje de estas instrucciones. Frente a esta cuestión caben dos filosofías de diseño.

Existen tres tipos en cuanto al juego de instrucciones que se utilizaran denominadas CISC, RISC Y SISC.



2.2.6.1 Arquitectura CISC.

La arquitectura CISC (Computadora con conjunto de instrucciones complejas) se basa en tener un conjunto grande de instrucciones, incluyendo las complejas. La programación en CISC es más fácil que en otros diseños debido a que hay una instrucción para una tarea simple o compleja. La complejidad del conjunto de instrucciones hace que el sistema de circuitos del UCP y la unidad de control sean muy complicados. Los diseñadores de la arquitectura CISC han ideado una solución para reducir la complejidad.

La programación se realiza en dos niveles. Una instrucción en el lenguaje de la máquina no se ejecuta directamente en la UCP. El UCP solo ejecuta operaciones simples llamadas microoperaciones. El tipo de programación que utiliza las microoperaciones se llama microprogramación. Una instrucción compleja se transforma en un conjunto de estas operaciones simples y entonces son ejecutadas por el UCP. Esto requiere la adición de una memoria especial llamada micromemoria que aloja el conjunto de operaciones para cada instrucción compleja.

2.2.6.2 Arquitectura RISC.

La arquitectura RISC (Computadora de conjunto de instrucciones reducidas) se basa en tener un conjunto pequeño de instrucciones que realicen un número mínimo de operaciones simples. Las instrucciones complejas se simulan usando un subconjunto de instrucciones simples. Esto provoca que la programación en RISC sea más difícil y extensa que en el otro diseño. Por otra parte, es una constatación estadística que la potencia práctica de UCP reside en un conjunto pequeño de su repertorio de instrucciones, con lo cual si se reduce dicho repertorio se sigue conservando la potencia del UCP, y disminuyendo la complejidad del diseño y del costo. El hecho de que la estructura simple



de un procesador RISC hace que el circuito integrado sea mas pequeño aprovechándose con frecuencia para ubicar en el mismo funciones adicionales como:

- Unidades para el procesamiento aritmético de punto flotante.
- Unidad de administración de memoria.
- Funciones de control de memoria cache.
- Implantación de un conjunto de registros múltiples.

El procesador RISC al tener actualmente unos mecanismos rápidos de acceso a memoria, buses de alta velocidad y compiladores especializados en esta arquitectura, hace que los UCP RISC obtengan en general unos mejores rendimientos.

2.2.6.3 Arquitectura SISC.

La arquitectura SISC (Computadores de juego de instrucciones específico) se basa en los microcontroladores destinados a aplicaciones muy concretas, el juego de instrucciones, además de ser reducido, es "específico", es decir, las instrucciones se adaptan a las necesidades de la aplicación prevista.

2.2.7 Memorias.

En los microcontroladores la memoria de instrucciones y datos forman parte del circuito integrado. Una parte debe de ser no volátil, tipo ROM, y se destina a contener el programa de instrucciones que gobierna la aplicación. Otra parte de la memoria debe de ser tipo RAM, volátil, y se destina a guardar las variables y los datos. El microprocesador está diseñado para que en su memoria de programa se almacenen todas las instrucciones del



programa de control. No hay posibilidad de utilizar memorias externas de ampliación. Como el programa a ejecutar siempre es el mismo, debe estar grabado de forma permanente.

Las diferencias de los microcontrolador a las computadoras personales son:

- No existe sistemas de almacenamiento masivo como disco duros.
- Como el microcontrolador solo se destina a una tarea en la memoria ROM, la mayoría de ellos permiten almacenar un único programa de trabajo.

2.2.7.1 ROM de mascara.

En este caso no se graba el programa en memoria sino que el programa se graba en el microcontrolador durante su fabricación, es un proceso similar al de producción de los CD comercial mediante masterización. El programa queda grabado permanentemente por lo que no se puede borrar ni escribir. El costo inicial de producir un circuito de este tipo es alto, porque el diseño y producción de la máscara es un proceso costoso, sin embargo, cuando se necesitan varios miles o incluso cientos de miles de microcontroladores para una aplicación determinada, como por ejemplo, algún electrodoméstico, el costo inicial de producción de la máscara y el de fabricación del circuito se distribuye entre todos los circuitos de la serie y, el costo final de ésta, es bastante menor que el de sus semejantes con otro tipo de memoria.

2.2.7.2 OTP.

Este tipo de memoria, también es conocida como PROM o simplemente ROM. Los microcontroladores con memoria OTP contienen una memoria no volátil de solo lectura



(Programable solo una vez) por el usuario. El usuario es quien puede programar el microcontrolador una sola vez, con algún tipo de programador. Se utilizan en sistemas donde el programa no requiera futuras actualizaciones y para series relativamente pequeñas, donde la variante de máscara sea muy costosa, también para sistemas que requieren serialización de datos, almacenados como constantes en la memoria de programas. Tanto en este tipo de memoria como en las EPROM, se suele usar la encriptación mediante fusibles para así proteger el código contenido.

2.2.7.3 EPROM.

Los microcontroladores que disponen de la memoria EPROM (Memoria de solo lectura borrrable y programable) pueden grabarse y borrarse muchas veces. Los microcontroladores con este tipo de memoria son muy fáciles de identificar porque su encapsulado es de cerámica y llevan encima una ventanita de vidrio desde la cual puede verse la oblea de silicio del microcontrolador.

Se fabrican así porque la memoria EPROM es reprogramable, pero antes debe borrarse, y para ello hay que exponerla a una fuente de luz ultravioleta, el proceso de grabación es similar al empleado para las memorias OTP. Al aparecer tecnologías menos costosas y más flexibles, como las memorias EEPROM y FLASH, este tipo de memoria han caído en desuso, se utilizaban en sistemas que requieren actualizaciones del programa y para los procesos de desarrollo y puesta a punto.

2.2.7.4 EEPROM.

El microcontrolador que utiliza este tipo memoria EEPROM (Memoria de solo lectura borrrable y programable eléctricamente) pueden grabarse y borrarse muchas veces. Este tipo



de memoria fueron el sustituto natural de las memorias EPROM, la diferencia fundamental es que pueden ser borradas eléctricamente, por lo que la ventanilla de cristal de cuarzo y los encapsulados cerámicos no son necesarios.

Al disminuir los costos de los encapsulados, los microcontroladores con este tipo de memoria se hicieron más baratos y cómodos para trabajar que sus equivalentes con memoria EPROM. Otra característica destacable de este tipo de microcontrolador es que fue en ellos donde comenzaron a utilizarse los sistemas de programación en circuito o ICSP que evitan tener que sacar el microcontrolador de la tarjeta que lo aloja para hacer actualizaciones al programa. Este tipo de memoria es idóneo para el aprendizaje.

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

2.2.7.5 FLASH.

Se trata de una memoria no volátil de bajo consumo, que se puede escribir y borrar. Funciona como una ROM y una RAM pero consume menos y es más pequeña. En el campo de las memorias reprogramables para microcontroladores, son el último avance tecnológico en uso a gran escala, y han sustituido a los microcontroladores con memoria

EEPROM, ya que son más rápidas y de mayor densidad. Estas pueden ser programables en el circuito. A las ventajas de las memorias flash se le adicionan su gran densidad respecto a sus predecesoras lo que permite incrementar la cantidad de memoria de programas a un costo muy bajo. Pueden además ser programadas con las mismas tensiones de alimentación



del microcontrolador, el acceso en lectura y la velocidad de programación es superior, disminución de los costos de producción, entre otras. Lo más habitual es encontrar que la memoria de programas y datos está ubicada toda dentro del microcontrolador, de hecho, actualmente son pocos los microcontroladores que permiten conectar memoria de programas en el exterior del encapsulado. Las razones para estas “limitaciones” están dadas porque el objetivo fundamental es obtener la mayor integración posible y conectar memorias externas consume líneas de E/S que son uno de los recursos más preciados de los microcontroladores.

2.2.8 Memorias de datos.

En los microcontroladores la memoria no es abundante, típicamente la memoria de programas no excederá de 16 K-localizaciones de memoria no volátil. Para la memoria RAM está destinada al almacenamiento de información temporal que será utilizada por el procesador para realizar cálculos u otro tipo de operaciones lógicas. En el espacio de direcciones de memoria RAM se ubican además los registros de trabajo del procesador y los de configuración y trabajo de los distintos periféricos del microcontrolador. Es por ello que en la mayoría de los casos, aunque se tenga un espacio de direcciones de un tamaño determinado, la cantidad de memoria RAM de que dispone el programador para almacenar sus datos es menor que la que puede direccionar el procesador.

2.2.9 Líneas de Entradas y Salidas.

Los pines de comunicación de los microcontroladores se agrupan en conjuntos que se llaman puertos, porque dejan entrar y salir la información al procesador. Dichos puertos deben soportar las líneas que precisan los distintos periféricos que hay integrados en la



cápsula. Cuantos más periféricos dispone el modelo, exige más líneas de comunicación y mayor numero de patitas, con más multiplexado de señales.

La mayoría de las patitas de los Puertos son multifunción, es decir, soportan diferentes funciones según sean programadas. Así, por ejemplo, existen pines que a veces funcionan como líneas de E/S digitales y otras como entradas o salidas de señales analógicas para un comparador.

Las líneas de E/S que se adaptan con los periféricos manejan información en paralelo. Hay modelos de microcontroladores que soportan comunicación serial, mientras que otros disponen de conjuntos de líneas que implementan puertas de comunicación para diversos protocolos como puede ser el I2C, OneWire, USB entre otros.

2.2.10 Reloj Principal.

Todos los microcontroladores disponen de un circuito oscilador que genera una onda cuadrada de alta frecuencia, que configura los impulsos de reloj usados en la sincronización de todas las operaciones del sistema. Esta señal del reloj es el motor del sistema y la que hace que el programa y los contadores avancen.

Generalmente, el circuito de reloj está incorporado en el microcontrolador y sólo se necesitan unos pocos componentes exteriores para seleccionar y estabilizar la frecuencia de trabajo. Dichos componentes suelen consistir en un cristal de cuarzo junto a elementos pasivos o bien un resonador cerámico o una red R-C. Aumentar la frecuencia de reloj supone disminuir el tiempo en que se ejecutan las instrucciones pero lleva aparejado un incremento del consumo de energía y de calor generado.



2.2.11 Recursos Especiales.

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

Los principales recursos específicos que incorporan los microcontroladores son:

- Temporizadores o "Timers".
- Perro guardian o Watchdog.
- Protección ante fallo de alimentación o Brownout.
- Estado de reposo a bajo consumo.
- Conversor ADC.
- Conversor DAC.
- Comparador analógico.
- Modulador de ancho de pulsos o PWM.
- Puertas de E/S digitales.
- Puertas de comunicación.

2.2.11.1 Temporizadores o Timers.

Se emplean para controlar periodos de tiempo (temporizadores) y para llevar la cuenta de acontecimientos que suceden en el exterior (contadores). Para la medida de tiempos se carga un registro con el valor adecuado y a continuación dicho valor se va



incrementando o decrementando al ritmo de los impulsos de reloj o algún múltiplo hasta que se desborde y llegue a 0, momento en el que se produce un aviso.

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

2.2.11.2 Perro guardián o Watchdog.

Es un mecanismo de seguridad que provoca un reset del sistema en caso de que éste se haya bloqueado.

Consiste en un temporizador que irá continuamente decrementando un contador, inicialmente con un valor alto. Cuando este contador llegue a cero, se reseteará el sistema.

Así que se debe diseñar el programa de manera que refresque o inicialice el perro guardián antes de que provoque el reset. Si el programa falla o se bloquea, al no poder actualizar el contador del perro guardián a su valor de inicio, éste llegará a decrementarse hasta cero y se reseteará el sistema.

2.2.11.3 Protección ante fallo de alimentación o Brownout.

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



2.2.11.4 Estado de reposo a bajo consumo.

Son abundantes las situaciones reales de trabajo en que el microcontrolador debe esperar, sin hacer nada, a que se produzca algún acontecimiento externo que le ponga de nuevo en funcionamiento. Para ahorrar energía, (factor clave en los aparatos portátiles), los microcontroladores disponen de una instrucción especial (SLEEP en los PIC), que les pasa al estado de reposo o de bajo consumo, en el cual los requerimientos de potencia son mínimos.

En dicho estado se detiene el reloj principal y se “congelan” sus circuitos asociados, quedando sumido en un profundo “sueño” el microcontrolador. Al activarse una interrupción ocasionada por el acontecimiento esperado, el microcontrolador se despierta y reanuda su trabajo.

2.2.11.5 Conversor ADC.

Es un dispositivo electrónico capaz de convertir una entrada analógica de voltaje en un valor binario. La señal analógica, que varía de forma continua en el tiempo, se conecta a la entrada del dispositivo y se somete a un muestreo a una velocidad fija, obteniéndose así una señal digital a la salida del mismo.

Estos conversores poseen dos señales de entrada llamadas V_{ref+} y V_{ref-} y determinan el rango en el cual se convertirá una señal de entrada. El dispositivo establece una relación entre su entrada (señal analógica) y su salida (digital) dependiendo de su resolución. Esta resolución se puede saber, siempre y cuando conozcamos el valor máximo que la entrada de información utiliza y la cantidad máxima de la salida en dígitos binarios.



2.2.11.5.1 Tipos de conversores usuales.

2.2.11.5.1.1 De aproximaciones sucesivas.

Es el empleado más comúnmente, apto para aplicaciones que no necesitan grandes resoluciones ni velocidades. Debido a su bajo coste se suele integrar en la mayoría de microcontroladores permitiendo una solución de bajo coste en un único chip para numerosas aplicaciones de control. El conversor realiza una búsqueda dicotómica del valor presente en la entrada. Su principal carencia es el elevado tiempo de conversión necesario.

2.2.11.5.1.2 Flash

Este conversor destaca por su elevada velocidad de funcionamiento. Está formado por una cadena de divisores de tensión y comparadores, realizando la conversión de manera inmediata en una única operación. Su principal desventaja es el elevado costo.

2.2.11.5.1.3 Sigma – Delta.

Tienen una velocidad máxima de conversión baja pero a cambio poseen una relación señal a ruido muy elevada, la mayor de todos.

2.2.11.6 Conversores DAC.

Es un dispositivo para convertir datos digitales en señales de corriente o de tensión analógica. La mayoría de los DAC utilizan alguna forma de red reostática. Los datos digitales se aplican a los reóstatos en grupos de bits. Las resistencias varían en



proporciones definidas y el flujo de corriente de cada uno está directamente relacionado con el valor binario del bit recibido.

Mediante una suma ponderada de los dígitos de valor 1 se consigue, en forma muy simple, un conversor digital-analógico rápido; la ponderación puede hacerse con una serie de resistencias en progresión geométrica (cada una mitad de la anterior), lo cual obliga a utilizar un amplio rango de resistencias, o bien mediante una red R-2R que efectúa sucesivas divisiones por 2.

2.2.11.7 Comparador Analógico.

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

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

2.2.11.8 Modulador de ancho de pulsos PWM.

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



2.2.11.9 Puertos de entradas y salidas digitales.

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

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

2.2.11.10 Puertos de comunicación.

Son puertos que se utilizan para dotar al microcontrolador de la posibilidad de comunicarse con otros dispositivos externos, otros buses de microprocesadores, buses de sistemas, buses de redes y así poder adaptarlos con otros elementos bajo normas y protocolos.

Algunos modelos disponen de recursos que permiten directamente esta tarea, entre los que destacan:

- UART, adaptador de comunicación serie asíncrona.
- USART, adaptador de comunicación serie síncrona y asíncrona.
- USB, bus serial universal.
- Bus I2C, interfaz serial de dos hilos.
- CAN, permitir la adaptación con redes de conexionado multiplexado desarrollado conjuntamente por Bosch e Intel para el cableado de dispositivos en automóviles. En EE.UU. se usa el J1850.



2.2.11.10.1 UART.

Éste controla los puertos y dispositivos serie. Las funciones principales de chip UART son de manejar las interrupciones de los dispositivos conectados al puerto serie y de convertir los datos en formato paralelo, transmitidos al bus de sistema, a datos en formato serie, para que puedan ser transmitidos a través de los puertos y viceversa.

El UART toma bytes de datos y transmite los bits individuales de forma secuencial. En el destino, un segundo UART reensambla los bits en bytes completos. La transmisión serie de la información digital (bits) a través de un cable único u otros medios es mucho más efectiva en cuanto a costo que la transmisión en paralelo a través de múltiples cables. Se utiliza un UART para convertir la información transmitida entre su forma secuencial y paralela en cada terminal de enlace. Cada UART contiene un registro de desplazamiento que es el método fundamental de conversión entre las forma serie y paralelo.

El UART normalmente no genera directamente o recibe las señales externas entre los diferentes módulos del equipo. Usualmente se usan dispositivos de interfaz separados para convertir las señales de nivel lógico del UART hacia y desde los niveles de señalización externos.

2.2.11.10.2 USART.

Se trata de un periférico para la transmisión de datos en formato serie, utilizando técnicas de transmisión sincrónica o asincrónica, según se configure el periférico. La característica más destacable de este tipo de periféricos es que utiliza solamente dos terminales para el envío y recepción de datos, en cualquiera de los dos modos de trabajo.



En el caso de las comunicaciones sincrónicas uno de los terminales se comporta como reloj (CLK) y el otro como datos (DT). Este tipo de comunicaciones generalmente demanda el uso de más E/S del dispositivo, por ejemplo un microcontrolador, para gestionar el uso del canal de comunicaciones, la ventaja que tiene radica en que no se requiere realizar una configuración previa de los dispositivos conectados al medio de comunicación debido a que la señal de sincronismo viaja por el medio de comunicación.

Para las comunicaciones asincrónicas, se destina un terminal a la transmisión (Tx) y otro a la recepción (Rx), en este caso el sincronismo se hace dentro de cada equipo y la interfaz solo define el uso de un bit de start y otro de stop, para indicar el inicio y fin de transmisión de un byte, es por eso que todos los equipos interconectados deben estar configurados para el mismo bit-rate.

2.2.11.10.3 USB.

Es una interfaz de 4 hilos plug&play entre el ordenador y una gran cantidad de dispositivos. Su función es la de transmitir datos con una determinada velocidad y la de distribuir 5V para su alimentación. Fue desarrollada para mejorar las lentas interfaces serie (RS-232) y paralelo que se usaban antes de esta, y desde entonces ha sido adoptada rápidamente por la industria informática.

Es un bus basado en el paso de un testigo, semejante a otros buses como los de las redes locales en anillo con paso de testigo y las redes FDDI. El controlador USB distribuye testigos por el bus. El dispositivo cuya dirección coincide con la que porta el testigo responde aceptando o enviando datos al controlador. Este también gestiona la distribución de energía a los periféricos que lo requieran.



2.2.11.10.4 Bus I2C.

Es un bus de comunicaciones en serie. La velocidad es de 100Kbits por segundo en el modo estándar, aunque también permite velocidades de 3.4 Mbit/s. Es un bus muy usado en la industria, principalmente para comunicar microcontroladores y sus periféricos en sistemas integrados y generalizando mas para comunicar circuitos integrados entre si que normalmente no residen en un mismo circuito impreso.

La principal característica de I²C es que utiliza dos líneas para transmitir la información: una para los datos y por otra la señal de reloj. También es necesaria una tercera línea, pero esta sólo es la referencia (masa). Como suelen comunicarse circuitos en una misma placa que comparten una misma masa esta tercera línea no suele ser necesaria.

Las líneas se llaman:

- SDA: Datos.
- SCK: Reloj.
- GND: Tierra.

Las dos primeras líneas son drenador abierto, por lo que necesitan resistencias de pull-up.

Los dispositivos conectados al bus I²C tienen una dirección única para cada uno. También pueden ser maestros o esclavos. El dispositivo maestro inicia la transferencia de datos y además genera la señal de reloj, pero no es necesario que el maestro sea siempre el mismo dispositivo, esta característica se la pueden ir pasando los dispositivos que tengan esa capacidad. Esta característica hace que al bus I²C se le denomine bus multimaestro.



2.2.11.10.5 Bus CAN.

Es un protocolo de comunicaciones desarrollado por la firma alemana Robert Bosch GmbH, basado en una topología bus para la transmisión de mensajes en entornos distribuidos. Además ofrece una solución a la gestión de la comunicación entre múltiples CPUs (unidades centrales de proceso).

El protocolo de comunicaciones CAN proporciona los siguientes beneficios:

- Es un protocolo de comunicaciones normalizado, con lo que se simplifica y economiza la tarea de comunicar subsistemas de diferentes fabricantes sobre una red común o bus.
- El procesador anfitrión (host) delega la carga de comunicaciones a un periférico inteligente, por lo tanto el procesador anfitrión dispone de mayor tiempo para ejecutar sus propias tareas.
- Al ser una red multiplexada, reduce considerablemente el cableado y elimina las conexiones punto a punto, excepto en los enganches.

2.3 Programación de Microcontroladores.

La utilización de lenguajes de programación mas cercanos a la maquina (de bajo nivel) representa un considerable ahorro de código en la confección de los programas, lo que es muy importante dada la estricta limitación de la capacidad de la memoria de instrucciones. Los programas bien realizados en lenguaje ensamblador optimizan el tamaño de la memoria que ocupan y su ejecución son muy rápida. Los lenguajes de alto nivel más empleados con microcontroladores son el C y el BASIC, de los que existen varias empresas



comercializan versiones de compiladores e interpretes para diversas familias de microcontroladores.

2.4 Compilador.

Es un programa informático que permite traducir el código fuente de un programa en un lenguaje de alto nivel a otro lenguaje de programación de nivel inferior (típicamente lenguaje de maquina) generando un programa equivalente que la máquina será capaz de interpretar. De esta manera un programador puede diseñar un programa en un lenguaje mucho más cercano a como piensa un ser humano, para luego compilarlo a un programa más manejable para una computadora.

2.5 Simulador.

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

2.6 Circuitos impresos.

Es un medio para sostener mecánicamente y conectar eléctricamente componentes electrónicos, a través de rutas o pistas de material conductor, grabados en hojas de cobre laminadas sobre un sustrato no conductor, comúnmente baquelita o fibra de vidrio.



La composición física de la mayoría de los circuitos impresos están compuestos por entre una a dieciséis capas conductoras, separadas y soportadas por capas de material aislante (sustrato) laminadas (pegadas) entre sí. Las capas pueden conectarse a través de orificios, llamados vías. Los orificios pueden ser electorecubiertos, o se pueden utilizar pequeños remaches. Los circuitos impresos de alta densidad pueden tener vías ciegas, que son visibles en sólo un lado de la tarjeta, o vías enterradas, que no son visibles en el exterior de la tarjeta.

Los tipos de circuitos impresos son:

- Multicapa que es la más habitual en productos comerciales. Suelen tener entre 8 y 10 capas, de las cuales unas deben de estar enterradas en el sustrato.
- 2-sided plated holes que es un diseño muy complicado de bajo coste con taladros metalizados que nos permite hacer pasos de cara.
- Single-sided non-plated holes que es un PCB con agujeros sin metalizar. Se usa en diseños de bajo coste y sencillos.
- 2-sided non-plated holes es un diseño sencillo con taladros sin metalizar. Tiene un sustrato de fibras de vidrio y resina. Hay que soldar por los dos lados para que haya continuidad.



CAPITULO III: MARCO METODOLOGICO

3.1 Tipo y diseño de la investigación.

En el presente capítulo se habla sobre el tipo de investigación, las técnicas de recolección y análisis de datos, finalizando con la metodología a utilizar para el cumplimiento de los objetivos planteados en el Capítulo I.

El presente Trabajo Especial de Grado, titulado “Diseño de una plataforma didáctica para el laboratorio de Microprocesadores de la Universidad de Carabobo”, se enmarca dentro de una investigación Proyectiva del tipo factible, ya que se desarrolló un modelo operativo para solucionar una necesidad instrumental. Esto según los tipos de investigación elaborada por Martha Nelly Córdoba y Carolina Monsalve.

3.2 Fases de la investigación:

Durante el desarrollo del presente proyecto se llevarán a cabo fases de investigación relacionadas con los objetivos planteados:

3.2.1 Fase 1: Análisis de las prácticas.

En esta primera fase se debe realizar una revisión de los modelos de prácticas realizadas en el laboratorio de microprocesadores, esta fase contempló las siguientes actividades:

- Estudiar las prácticas que se realizan en el laboratorio de microprocesadores.
- Identificar los diferentes elementos que se le colocaran a la plataforma en función de las prácticas que se realizaran.



3.2.2 Fase 2: Diseño de la plataforma.

En esta segunda fase, una vez definida las prácticas y los elementos de hardware a utilizar se diseñara el modelo de circuito que cumpla con los requerimientos del laboratorio de microprocesadores tomando en cuenta las siguientes características:

- Verificación de los programas a utilizar.
- Diseño del circuito electrónico del programador.
- Revisión de cada uno de los elementos a utilizar.
- Diseño del circuito electrónico de la plataforma.

3.2.3 Fase 3: Elaboración de la plataforma.

En esta fase se lleva a cabo la elaboración del circuito con todas sus conexiones en baquelita tomando en cuenta lo siguiente:

- Elaboración del circuito electrónico con sus conexiones del programador en baquelita.
- Elaboración del circuito electrónico con sus conexiones de la plataforma en baquelita.

3.2.4 Fase 4: Implementación del dispositivo en hardware.

En esta fase se realizara la implementación del sistema de desarrollo tomando en cuenta todos los elementos que se colocaran en la plataforma.

3.2.5 Fase 5: Validación del dispositivo usando las prácticas existentes.

En esta fase se realizaran las diferentes pruebas para validar el sistema de desarrollo tomando en cuenta lo siguiente:



CAPITULO III



- Revisión de las prácticas a realizar.
- Programación de las prácticas.
- Transferencia del programa a la plataforma.
- Análisis de los resultados obtenidos.

3.2.6 Fase 6: Elaboración de documentación.

En esta fase se lleva a cabo la elaboración de la documentación necesaria utilizando los recursos definidos en las fases anteriores.



CAPITULO IV: DESARROLLO DE LA PLATAFORMA DIDÁCTICA

Para la elaboración del sistema de desarrollo se tomara en cuenta el número de prácticas que se realizan en el Laboratorio de Microprocesadores para así poder hacer una lista de materiales que se utilizaran en el desarrollo de dicho hardware.

4.1 Análisis de las prácticas:

Se realizara una revisión de las prácticas que se realizan en el Laboratorio de Microprocesadores. Estas prácticas están compuestas por los siguientes temas mostrados en la tabla 4.1:

Tabla 4.1 Compendio de prácticas realizadas en el Laboratorio de Microprocesadores.

Práctica	Título	Contenido
Práctica N° 1	Manipulación de entradas y salidas digitales	Conectar periféricos digitales (pulsadores, interruptores, leds) con el microcontrolador.
Práctica N° 2	Entradas y Salidas digitales, manejos de tablas.	Realizar conversiones entre códigos mediante el manejo de tablas. Manejar Displays siete segmentos. Implementar rutinas de retardo.



Práctica N° 3	Manejo y usos de temporizadores.	Configuración de los temporizadores. Utilizar señales de de frecuencia variable. Implementar algoritmos para generación de bases de tiempo.
Práctica N° 4	Adquisición de datos.	Utilizar una LCD de 2x16. Comunicación serial. Manejo de convertidores analógico digitales (ADC). Manejo de convertidores digital analógicos (DAC).

Al tener definido el número de prácticas que se realizaran y los contenidos que se desarrollaran en cada una de ellas se procederá a hacer el desarrollo de cada una de ellas para así identificar cada uno de los componentes de hardware que se utilizaran para la elaboración del sistema de desarrollo.

4.1.1 Práctica N° 1:

MANIPULACIÓN DE ENTRADAS Y SALIDAS DIGITALES

4.1.1.1 Objetivos:

En esta práctica se busca aprender a manipular los puertos de entradas y salidas de un microcontrolador utilizando los periféricos más comunes que se pueden presentar.



4.1.1.1.1 Objetivos Generales:

Diseñar programas en lenguaje ensamblador para manipular entradas y salidas digitales del microcontrolador, que se relacionen con la resolución de un algoritmo.

4.1.1.1.2 Objetivos Específicos:

- Verificar los procedimientos para el desarrollo de aplicaciones con microcontroladores
- Manipular salidas digitales con el microcontrolador.
- Procesar entradas digitales con el microcontrolador.
- Conectar periféricos digitales (pulsadores, interruptores, leds) con el microcontrolador.

4.1.1.2 Desarrollo de la práctica:

Se desea implementar un juego de azar basado en un generador de números aleatorios y la entrada del usuario.

El pulsador controla el LED “L” de la misma manera que la aplicación anterior. Ambos permanecen conectados en los mismos pines de E/S. Cuando se inicia el programa, el LED “L” se enciende para señalar que el usuario debe comenzar a jugar, introduciendo un número binario de 4 bits en los pines RA3, RA2, RA1, RA0, el cual llamaremos ENT.

Mientras el pulsador permanece sin presionar (y el usuario está presumiblemente configurando los switches RA3... RA0), el microcontrolador incrementa constantemente



una cuenta mediante un registro que denominaremos CNT. Cuando el usuario finaliza de ajustar la entrada, procede a presionar el pulsador. En este momento:

- Se detiene la cuenta CNT.
- Se apaga el LED “L”.
- Se verifica si la diferencia absoluta entre ENT y los 4 bits menos significativos de CNT es menor o igual a 6.

Si el resultado referido en es menor o igual a 6, el jugador gana. En este caso se prende el segundo LED “L2”. Si es mayor que 6, entonces el jugador pierde (no se enciende el LED “L2”). En ambos casos, gane o pierda, el valor de CNT se mostrará en los pines de salida RC5... RC2. La salida se mantiene así hasta que se presione el pulsador nuevamente y se encienda el LED “L” (y se pondrán en cero nuevamente los restantes bits del puerto C).

Nota: además del registro CNT, defina cuantos registros adicionales necesite para la aplicación.

Obtenga el diagrama de flujo o pseudocódigo del algoritmo de la aplicación de acuerdo al comportamiento descrito. A partir del DF o pseudocódigo obtenido, codifique el programa correspondiente en lenguaje ensamblador.

El circuito de la aplicación corresponde al del laboratorio. Simule su aplicación en el PROTEUS antes de traerla al laboratorio, para verificar su funcionamiento.



4.1.1.3 LABORATORIO Semana 2

- Muestre el pre-laboratorio a su profesor(a), quien le interrogará sobre el mismo.
- Abra la aplicación desarrollada en el MPLAB y proceda a su ensamblado.
- Efectúe la simulación de la aplicación en PROTEUS y verifique funcionamiento con su profesor(a).
- Programe el sistema de desarrollo PT-40 con la aplicación y verifique su funcionamiento con su profesor(a).
- Obtenga el porcentaje de juegos ganados para unas diez a quince jugadas. ¿Es el porcentaje de juegos ganados obtenido cercano al esperado teóricamente? Razone su respuesta e inclúyala en el informe final de la práctica.

4.1.1.4 Fundamento teórico:

Para el buen funcionamiento de esta práctica es necesario conocer que elementos de hardware se utilizaran para así realizar las pruebas de todas las rutinas que se presentaran en el sistema de desarrollo.

4.1.1.4.1 Interruptores y pulsadores:

Son dispositivos utilizados para activar alguna función. Los pulsadores son de diversa forma y tamaño y se encuentran en todo tipo de dispositivos, aunque principalmente en aparatos eléctricos o electrónicos.

Un pulsador de un dispositivo electrónico, funciona por lo general como un interruptor eléctrico, en su interior tiene un contacto con lo que al pulsarlo se activará la función inversa de la que en ese momento este realizando. Al soltarse el pulsador volverá a la posición inicial o al estado anterior de donde partió. La figura 4.1.1 muestra la simbología de los pulsadores, tanto para N.A. (Normalmente abierto) como para N.C. (Normalmente cerrado).

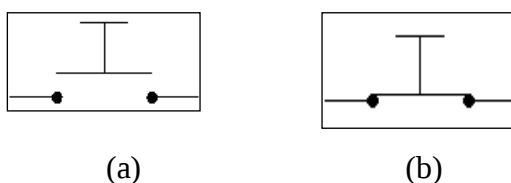


Figura 4.1. Simbología de los pulsadores. (a) N.A. y (b) N.C. Fuente: propia.

Los actuantes de los interruptores pueden ser normalmente abiertos, en cuyo caso al accionarlos se cierra el circuito o normalmente cerrados en cuyo caso al accionarlos se abre el circuito. En la figura 4.1.2 se muestra la simbología de un interruptor.

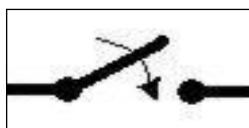


Figura 4.2 Simbología de un interruptor. Fuente: Propia.

El nivel lógico que introduce los interruptores y pulsadores depende de cómo sea la conexión con la fuente de alimentación.

Para los interruptores la forma de conectar se puede ver en la figura 4.3. En la figura (a) se puede ver al interruptor abierto por lo que introduce un nivel bajo en la entrada que al

accionarse pasara a un nivel alto. En la figura (b) se ve al interruptor en estado cerrado por lo que introduce un estado bajo que al abrir el interruptor introduce un nivel alto.

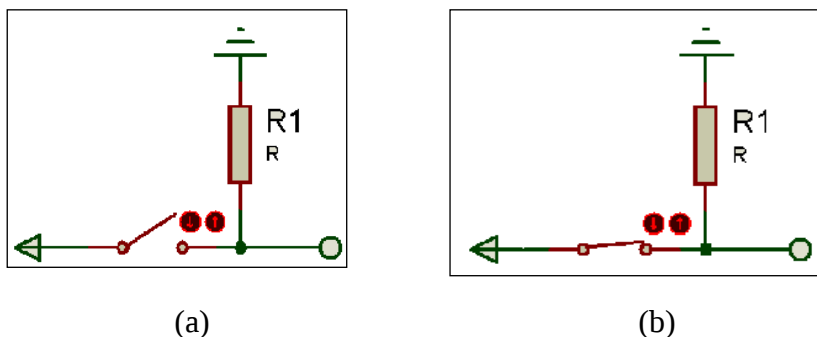


Figura 4.3. Esquema de conexión de los interruptores. En la figura (a) esta el interruptor abierto. En la figura (b) el interruptor se encuentra cerrado.

En el caso de los pulsadores se puede apreciar en la figura 4.4 su modo de conexión. En la figura (a) se ve a un pulsador N.C. en esta introduce un nivel bajo. En la figura (b) se aprecia un pulsador N.A. este introduce un nivel alto.

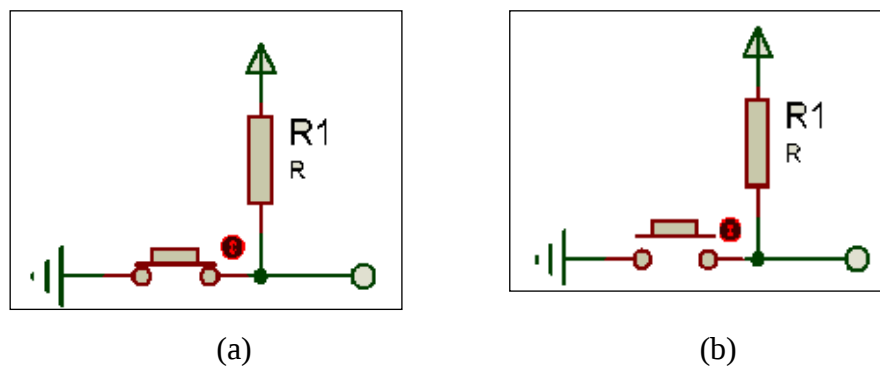


Figura 4.4. Esquema de conexión de un pulsador. En la figura (a) se aprecia una conexión N.C. En la figura (b) se muestra una conexión N.A.

La diferencia entre ambos dispositivos es que el interruptor queda activo mientras que el pulsador vuelve a su estado inicial.

4.1.1.4.2 Diodos LED:

El led es un tipo especial de diodo, que trabaja como un diodo común, pero que al ser atravesado por la corriente eléctrica, emite luz. Existen diodos led de varios colores que dependen del material con el cual fueron construidos. Hay de color rojo, verde, amarillo, ámbar, infrarrojo, entre otros. Eléctricamente el diodo LED se comporta igual que un diodo de silicio o germanio. Si se pasa una corriente a través del diodo semiconductor, se inyectan electrones y huecos en las regiones P y N, respectivamente. En la Figura 4.5 se ilustra la simbología de un diodo led.

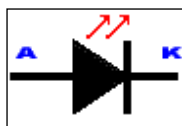


Figura 4.5 Simbología de un diodo led. Fuente: propia.

Dependiendo de la magnitud de la corriente, hay recombinación de los portadores de carga (electrones y huecos). Hay un tipo de recombinaciones que se llaman recombinaciones radiantes (aquí la emisión de luz). La relación entre las recombinaciones radiantes y el total de recombinaciones depende del material semiconductor utilizado.

Dependiendo del material de que está hecho el led, será la emisión de la longitud de onda y por ende el color. Debe de escogerse bien la corriente que atraviesa el led para obtener una buena intensidad luminosa y evitar que este se pueda dañar. El led tiene un voltaje de operación comprendido entre 1.5 V a 2.2 voltios aproximadamente y el valor de corriente que debe circular por él está entre los 10 y 20 miliamperios (mA) en los diodos de color rojo y entre los 20 y 40 miliamperios (mA) para los otros leds.

Los diodos led tienen enormes ventajas sobre las lámparas indicadoras comunes, como su bajo consumo de energía, su mantenimiento casi nulo y con una vida aproximada de 100,000 horas. En la Figura 4.6 se puede apreciar a un diodo led donde se distingue al ánodo del cátodo.

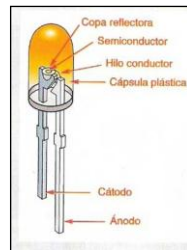


Figura 4.6 Un diodo led. Se puede identificar al ánodo del cátodo. Fuente: Electrónica Unicrom.

4.1.2 Práctica N°2:

ENTRADAS Y SALIDAS DIGITALES. MANEJO DE TABLAS.

4.1.2.1 Objetivos:

En el desarrollo de esta práctica se busca hacer uso de entradas y salidas digitales mediante la utilización de pulsadores y del display de 7 segmentos.

4.1.2.1.1 Objetivos Generales:

Eliminar el rebote inherente a las entradas digitales implementadas mediante pulsadores.



4.1.2.1.2 Objetivos Específicos:

- Realizar conversiones entre códigos mediante el manejo de tablas.
- Manejo de display 7 segmentos.
- Implementar rutinas de retardo.

4.1.2.2 LABORATORIO Semana 1

Durante esta semana debe simular en Proteus y programar el Sistema de desarrollo para mostrar una cuenta ascendente/descendente con el circuito mostrado. La cuenta ascenderá o descenderá unitariamente (de uno en uno) según se decida mediante los pulsadores UP y DOWN.

4.1.2.3 LABORATORIO Semana 2

En el MPLAB simule la rutina de retardo preparada en el Prelaboratorio. Utilice la ventana stopwatch para verificar la duración exacta de su rutina, de acuerdo con el tiempo necesario para eliminar el rebote. Esto no debe llevarle más de 20 o 30 minutos.

En el programa de la semana anterior incluya las instrucciones necesarias para eliminar el rebote de los pulsadores UP y DOWN.

Si no se ha eliminado el problema del rebote, aumente ligeramente el tiempo de retardo y pruebe de nuevo.

4.1.2.4 Fundamento Teórico:

4.1.2.4.1 Display 7 Segmentos:

El display de 7 segmentos, es un componente que se utiliza para la representación de números en muchos dispositivos electrónicos. Este elemento se ensambla o arma de manera que se pueda activar cada segmento (diodo LED) por separado logrando de esta manera combinar los elementos y representar todos los números en el display (del 0 al 9).

El display de 7 segmentos más común es el de color rojo, por su facilidad de visualización. Cada elemento del display tiene asignado una letra que identifica su posición en el arreglo del display. Figura 4.1.7 Esquema de un display de 7 segmentos.

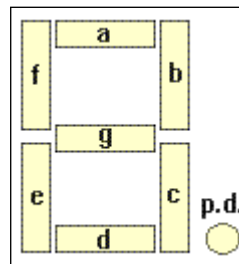


Figura 4.1.7 Esquema de un display de 7 segmentos. Fuente: www.unicrom.com

4.1.2.4.1.1 El display ánodo común:

En el display ánodo común, todos los ánodos de los diodos led están unidos y conectados a la fuente de alimentación. En este caso para activar cualquier elemento hay que poner el cátodo del elemento a tierra de una resistencia para limitar la corriente

que pasa por el elemento. Figura 4.8 Esquema de un display de 7 segmentos ánodo común.

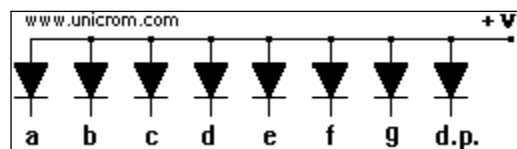


Figura 4.8 Esquema de un display de 7 segmentos ánodo común. Fuente:

www.unicrom.com

4.1.2.4.1.2 El display cátodo común:

El display cátodo común tiene todos los ánodos de los diodos led unidos y conectados a tierra. Para activar un segmento de estos hay que poner el ánodo del segmento a encender a V_{cc} (tensión de la fuente) a través de una resistencia para limitar el paso de la corriente. Figura 4.9 Esquema de un display 7 segmentos cátodo común.

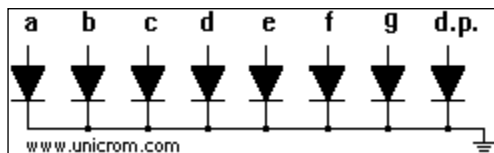


Figura 4.9 Esquema de un display de 7 segmentos cátodo común. Fuente:

www.unicrom.com

4.1.2.4.2 Rebote:

Cuando se abren y se cierran las láminas metálicas de los pulsadores hay unos milisegundos de inestabilidad en los que el circuito se abre y se cierra varias veces. A estas aperturas y cierres del tipo electromecánico que se producen al cambiar de estado se les



llama “Rebote” y puede dar a lugar informaciones erróneas. Ya que el manejo de estos elementos de entradas se hace de manera muy rápido por parte del microcontrolador, estos pueden originar la lectura de un estado incorrecto y por ende, pueden dar origen a una acción no adecuada.

Los rebotes pueden ser eliminados por software generando rutinas de tiempo de espera apropiados. El algoritmo elimina el rebote preguntando al pulsador si fue cambio de estado de esa manera la espera un determinado tiempo y vuelve a preguntar si se mantiene en el mismo estado. Si es así el realiza la acción sino solo fue un ruido y vuelve a su estado inicial.

4.1.3 Práctica N° 3:

MANEJO Y USOS DE TEMPORIZADORES

4.1.3.1 Objetivos:

Con el desarrollo de esta práctica se explica la importancia de las interrupciones aplicadas en sistemas digitales.

4.1.3.1.1 Objetivos Generales:

Desarrollar aplicadas basadas en el uso de los temporizadores de un microcontrolador PIC de gama media.



4.1.3.1.2 Objetivos Específicos:

- Diseño y uso de subrutinas en aplicaciones de mediana complejidad
- Describir las características funcionales, registros asociados y formas de configuración de los temporizadores Timer0, Timer1 y Timer2 del microcontrolador PIC16F877.
- Implementar algoritmos para generación de bases de tiempo en lenguaje ensamblador, en función de una variable de entrada.
- Utilizar señales de de frecuencia variable para aplicaciones que incluyan procesamiento de entradas digitales y activación de salidas digitales.

4.1.3.2 LABORATORIO.

Elabore una aplicación con el PIC16F877 de control de una línea de empaquetamiento de piezas, consistente en una banda transportadora y un dispositivo de empaquetamiento.

La velocidad de la banda es proporcional a la frecuencia de una onda cuadrada generada por el pin de salida B (RC0), de acuerdo a los valores de dos entradas F1 (RA1) y F0 (RA0).

Tabla 4.2 Valores de salidas de acuerdo a los valores de entradas.

F1(RA1)	F0(RA0)	Frecuencia de señal B (RC0)
0	0	20 hz
0	1	30 hz
1	0	40 Hz
1	1	50 hz

Fuente: Practica N°3 del Laboratorio de microprocesadores.

El sistema arranca cuando se presiona el pulsador INICIO/PARADA (RA2). El display 7 segmentos deberá permanecer apagado. Se genera la onda cuadrada de acuerdo a los valores leídos de F1, F0 y se enciende un led de señalización RUN (RC1), que se mantiene encendido mientras la banda transportadora se encuentra en movimiento.

Un sensor S (RA4) detecta el paso de una pieza frente al dispositivo de empaque. Una vez que se detecta la pieza, la banda se detiene ($B=0$), se apaga RUN y se enciende la señal EMPAQUE (RC2) por 0.75 segundos.

Seguidamente, cuando se apaga EMPAQUE, se reinicia el movimiento de la banda (a la misma velocidad) y se vuelve a encender RUN, hasta que otro sensor D (RA5) indica que se ha llegado al final de la banda, entonces se apaga la banda ($B=0$) y la señal de RUN durante 0.5 segundos; a continuación se activan nuevamente dichas señales (B y RUN), para el procesamiento de la siguiente pieza, de acuerdo al mismo procedimiento descrito anteriormente.



El PIC16F877 debe llevar una cuenta de piezas empacadas desde que se inició el proceso. Cuando se presiona el botón INICIO/PARADA con la banda en movimiento, se termina el proceso. El PIC apaga todas las señales de salida, detiene la banda, y muestra la cuenta de piezas empacadas hasta ese momento.

Para reiniciar el sistema y llevar la cuenta a cero, nuevamente se debe pulsar el botón INICIO/PARADA. La señal de frecuencia variable se obtendrá con el Timer1. Los retardos de 0.75 y 0.5 segundos serán por software. El pulsador de INICIO/PARADA presenta rebote, no así los sensores S y D. La rutina antirrebote puede efectuarla con el Timer 2 o por software.

4.1.4 Práctica N° 4:

ADQUISICIÓN DE DATOS

4.1.4.1 Objetivos:

El desarrollo de esta práctica contempla la utilización del convertidor analógico digital (ADC) que se encuentra en el microcontrolador para medir el voltaje de una señal DC y su posterior envío por el puerto serial integrado.

4.1.4.1.1 Objetivos Generales:

Desarrollar una aplicación basada en la adquisición de datos mediante el convertidor Analógico Digital interno de un microcontrolador.



4.1.4.1.2 Objetivos Específicos:

- Utilizar por primera vez un lenguaje de alto nivel para codificar programas de microcontroladores.
- Utilizar una LCD de 2x16 líneas como herramienta de visualización.
- Comprobar la validez de la función de transferencia de los ADC, estudiada en las clases teóricas.
- Comprender la Resolución como parámetro de calidad de los ADC.

4.1.4.2 LABORATORIO Semana 1:

- Copie el archivo P4_1_proton.bas en la ventana de edición de Protón.
- En la sección del listado correspondiente a la Etiqueta o Identificador “Otro”, reemplace las dos líneas siguientes por lo que se le solicita en cada caso.
- Compile y utilice el archivo generado en una simulación con Proteus. Analice.
- Detenga la Simulación.
- Reescriba la variable “N”, de manera que ahora se encienda el LED contrario al que se encendió la primera vez.
- Verifique si es correcta su selección anterior, compilando y simulado nuevamente. Si no obtiene el resultado deseado repita los puntos 5 y 6 hasta que lo logre.
- En listado impreso que Ud. trajo al Laboratorio, añada comentarios explicativos de lo que hace cada línea, donde no exista un comentario previo. Firme dicha hoja y entréguela a su profesor.



4.1.4.3 PRELABORATORIO Semana 2:

- Estudie el comando ADIN de Proton Basic, a través de la ayuda desplegable desde el menú “Help” de Proton.
- Elabore el código, en Protón Basic, de un algoritmo para digitalizar continuamente una señal analógica introducida a través del pín RE0 y visualizar su valor en la LCD. Paralelamente debe mostrarse el Valor de los Registros ADRESH y ADRESL después de cada conversión y además, 2 LED's conectados a los pines RC4 y RC5 se encenderán y apagarán según la entrada analógica varíe del modo indicado en la tabla 4.1.3 siguiente.

Tabla 4.3 Tabla analógica.

V _{in}	D1	D2
Menor a 0.87 V	OFF	OFF
Entre 0.87 V y 2.43 V	OFF	ON
Entre 2.43 V y 3.92 V	ON	ON
Mayor a 3.92	ON	OFF

Fuente: Práctica N°4 Laboratorio de Microprocesadores de la universidad de Carabobo.

4.1.4.4 LABORATORIO Semana 2:

Compile el programa anterior, simúlelo en PROTEUS y pruébelo en el sistema de desarrollo.



4.1.4.5 Fundamento teórico:

4.1.4.5.1 Display de cristal líquido (LCD):

Es uno de los periféricos más utilizados para la presentación de mensajes, variables y casi cualquier información proveniente de un microcontrolador. Gracias a su flexibilidad, buena visibilidad y precio reducido se ha convertido en el estándar de la visualización más utilizada con los microcontroladores.

Las pantallas LCD poseen un microcontrolador interno como el Hitachi 44780 o compatible que les permite:

- Presentar caracteres ASCII o griegos.
- Recibir y mostrar caracteres personalizados.
- Desplazamiento de caracteres (Scroll) tanto a la derecha como a la izquierda.
- Cambiar el aspecto o movimiento del cursor.
- Direcccionar de manera simple la posición de visualización de los caracteres.

Los terminales de conexión de las pantallas LCD de caracteres han sido estandarizados, sigue siendo generalmente compatible pin a pin como lo mostrado en la tabla 4.1.4.

Tabla 4.4 Diagrama de terminales.

PIN	NOMBRE	FUNCIÓN	DESCRIPCIÓN
1	Vss	Energía	Referencia 0 V. GND
2	Vdd	Energía	+5 V DC
3	Vee	Ajuste Contraste	Variable de 0 a 5 V
4	RS	Comando	Selección de Dato/Comando
5	R/ W	Comando	Control de Lectura/Escritura
6	E	Comando	Habilitación
7	D0	E/S	DATO LSB
8	D1	E/S	DATO
9	D2	E/S	DATO
10	D3	E/S	DATO
11	D4	E/S	DATO
12	D5	E/S	DATO
13	D6	E/S	DATO
14	D7	E/S	DATO MSB

4.2 Diseño de la plataforma:

Luego de realizar el marco teórico de cada una de las prácticas se procedió a verificar los componentes que se utilizan en el laboratorio de microprocesadores para así poder verificar como se desarrollara y que elementos tendrá el sistema de desarrollo que se plantea. Los elementos que se plantearan a continuación son los encargados de simular los diferentes procesos que se pueden realizar con los microcontroladores tomando en cuenta que estos se utilizan como dispositivos de entradas y salidas.

4.2.1 Elementos utilizados:

- Interruptores.
- Pulsadores.



- Diodos Leds.
- Display de 7 segmentos.
- Display LCD.
- Potenciómetros.
- Microcontrolador PIC 16F877.

4.2.2 Programa a utilizar para el diseño del circuito:

El programa que se utilizara para realizar el diseño tanto del programador como del sistema de desarrollo es el proteus, debido a que este software dispone de la totalidad de componentes a emplearse en las prácticas del laboratorio de microprocesadores. Principalmente dispone de la serie de microcontroladores PIC, Display 7 segmentos, microswitch, potenciómetros, pulsadores, diodos leds entre otros componentes.

Adicionalmente es una herramienta de simulación poderosa que será usada para la simulación de las diferentes prácticas que se realizaran.

4.2.3 Diseño del programador a utilizar:

El programador a utilizar es un diseño elaborado por la empresa microchip llamado pickit2. Este dispositivo es el encargado de programar al microcontrolador que estará en el sistema de desarrollo a elaborar. El diseño planteado por la empresa microchip es de uso libre por lo que el esquemático que ellos presentan puede ser copiado y rediseñado esto ultimo fue el caso que se presentara a continuación, ya que muchos de los componentes que ellos presentan son de difícil acceso.



Los métodos de programación de la mayoría de los microcontroladores que se encuentran en el mercado son de:

- Programación a bajo voltaje (LVP).
- Programación a alto voltaje (HVP).

4.2.3.1 Programación a bajo voltaje:

En este tipo de programación solo se permite al dispositivo trabajar la tensión de alimentación (V_{dd}). Para la configuración de este modo de programación solo se debe activar el pin específico dependiendo del microcontrolador que es el que indica el control de la entrada y salida de este modo de programación. Mientras que el microcontrolador se encuentra en este estado de programación la entrada en el pin de MCLR también se lleva a tensión de alimentación que va a ser el pin encargado de llevar al microcontrolador a entrar en el modo de programación. Para deshabilitar este modo de programación solo se debe de llevar a un estado bajo el pin específico.

4.2.3.2 Programación a alto voltaje:

Este tipo de programación es el más usado para programar a los microcontroladores ya que solo se lleva la tensión a niveles de entre los 12 V y 14 V por el pin de MCLR. Y se hace la transmisión de datos por dos pines donde uno lleva datos y el otro tiene una señal de reloj.

4.2.3.3 Diagrama del programador:

En la figura 4.10 se presenta el esquemático del programador que será utilizado en el sistema de desarrollo.

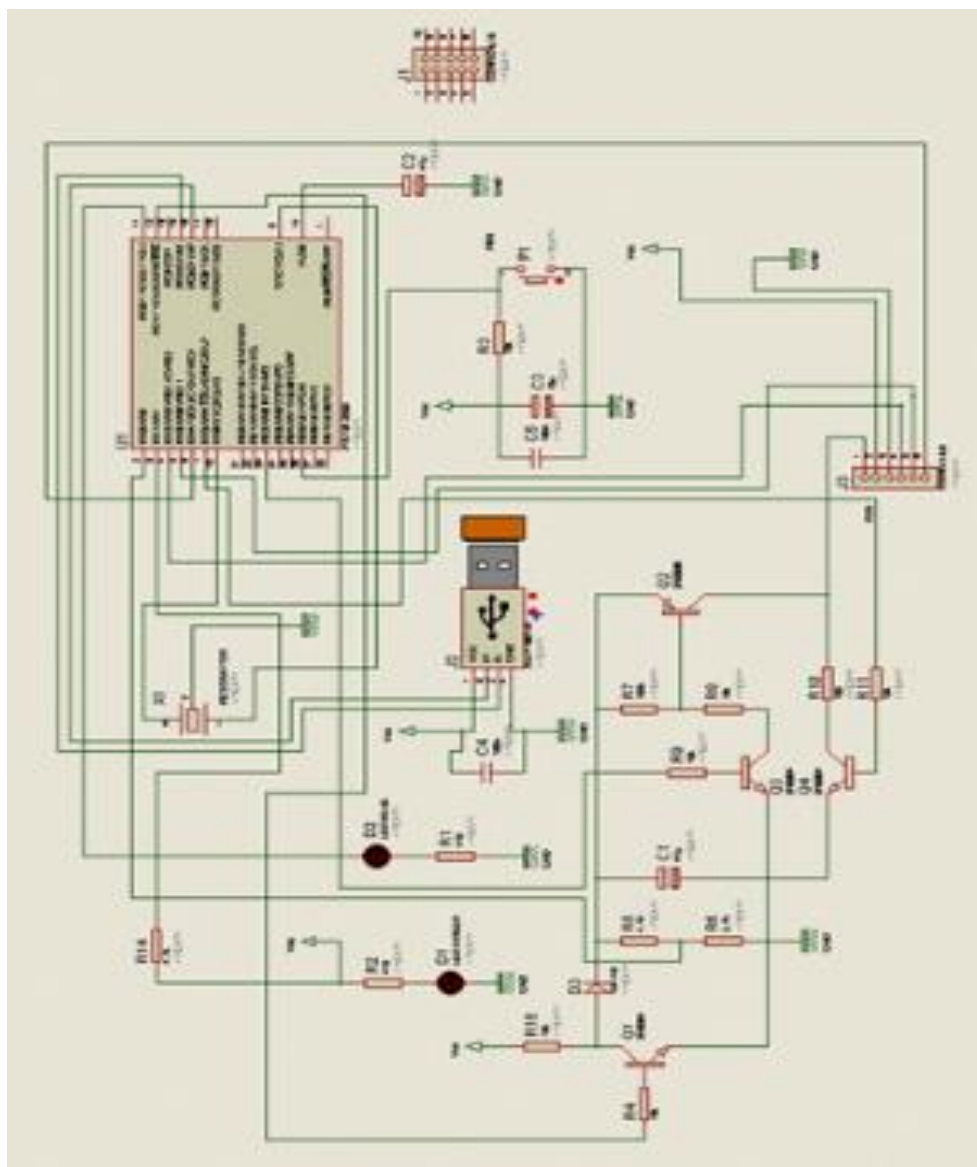


Figura 4.10 Esquemático del programador.

4.2.4 Diseño de la plataforma:

Para el diseño del sistema de desarrollo se tomara en cuenta los componentes que se utilizaran y como van a estar conectados en los diferentes puertos que posee el microcontrolador, para así de esta manera plantear el manual de usuario de la plataforma.

4.2.4.1 Valores de resistencias a utilizar:

La salida de un microcontrolador PIC se encuentra a 5V lo que hace que un dispositivo que este conectado a él se encuentre a esa tensión. Para los diodos leds que utilizamos estos necesitan una tensión de trabajo entre el ánodo y cátodo de 1.7V y una corriente de trabajo de 10mA. Por lo que para el cálculo de la resistencia a utilizar va a estar prevista por la siguiente fórmula:

$$R = \frac{V - V(led)}{I(led)}$$

El valor de la resistencia para colocar en serie con un led es:

$$R = \frac{5 - 1.7}{0.010} = 330\Omega$$

La forma de conectar un diodo led a un microcontrolador se encuentra en la figura 4.11.

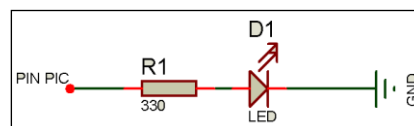


Figura 4.11 Conexionado de un led a la salida de un PIC.

Las resistencias a calcular para conectar un display de 7 segmentos son iguales a las de un led debido a que un display es un juego de leds internamente.

El modo de conectar un display de 7 segmentos se encuentra en la figura 4.12.

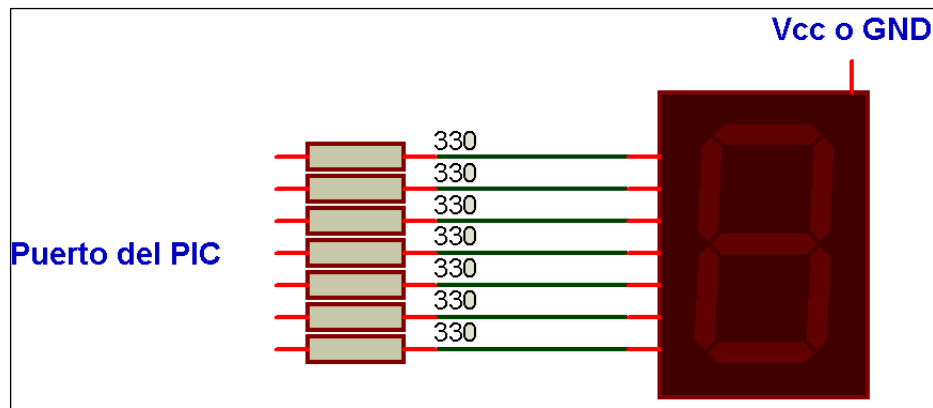


Figura 4.12 Diagrama de conexión de un display 7 segmentos a un PIC.

La entrada de un microcontrolador puede soportar hasta 25mA de corriente a 5V esto para poder identificar entre un 1 lógico y un 0 lógico

Tomando en cuenta lo anterior se debe colocar una resistencia para limitar la corriente y poder hacer detección del nivel lógico. La resistencia a colocar tendrá un valor provisto por la ecuación:

$$R = \frac{V}{I} = \frac{5}{0.025} = 200 \, \Omega$$

Debido a que el valor que nos dio de resistencia es para la máxima corriente que puede soportar el microcontrolador a la entrada se colocan resistencia a la entrada de mayor



resistencia normalmente de valores comprendidos entre 1k y 10k esto para limitar más la corriente. La resistencia que utilizaremos será de 10k.

4.2.4.2 Ubicación de los componentes a utilizar:

Se tomara en cuenta los valores de resistencia y los componentes que se han estudiado a medida que se han presentado en el proyecto.

Los puertos a utilizar y los componentes que cada uno de ellos van a tener estará descrito de la siguiente manera:

El puerto A tendrá desde (RA0 hasta RA3) un microswitch de 4 bits de tamaño, los bits restantes RA4 y RA5 serán utilizados con 1 pulsador cada uno. Se ha tomado esta configuración por decisión propia.

El puerto B será utilizado para colocar la pantalla LCD a través de una interfaz de 6 bits en su conexión.

El puerto C se utilizara para colocar un juegos de diodos leds uno por cada bit del puerto lo que corresponderá a 8 led en el puerto completo. El mismo puerto se utilizara para hacer conexiones hacia aplicaciones externas.

El puerto D será utilizados por dos display de 7 segmentos uno cátodo común y un ánodo común que serán controlados por un interruptor encargado de seleccionar cual de los displays funcionara.

En el puerto E se encontrara un potenciómetro y dos pulsadores debido a que el mismo tiene solo 3 bits para su uso.

4.2.4.3 Esquemático de la plataforma a utilizar:

En la figura 4.13 se presenta el esquemático de la plataforma didáctica.

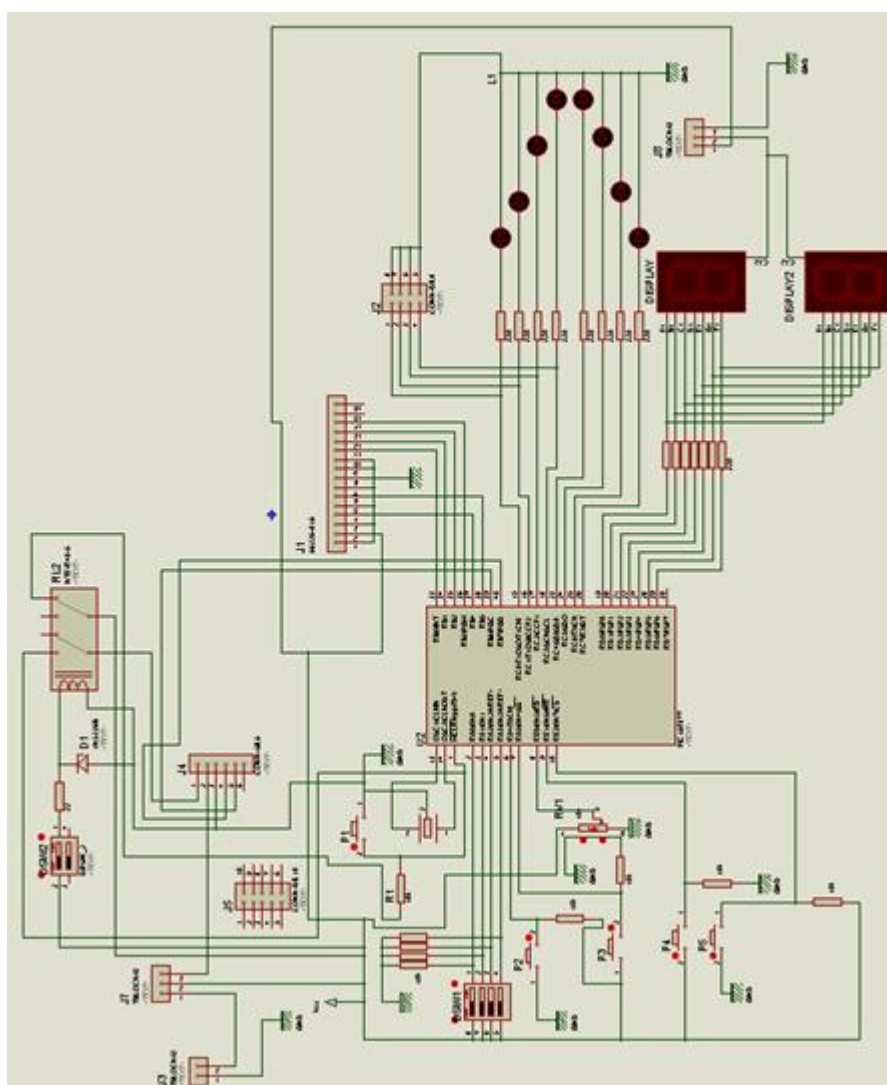


Figura 4.13. Esquemático de la plataforma didáctica. Fuente: Propia.



4.3 Diseño del sistema de desarrollo en pcb:

Después de realizado el esquemático del programador y de la plataforma didáctica, se procederá a hacer el diagrama en pcb del circuito planteado para el sistema de desarrollo. La manera se hacer este proceso va a ser mediante programa que puedan hacer el ruteado de las pistas para su posterior montaje en físico.

4.3.1 Programa a utilizar:

El programa ARES, o Advanced Routing and Editing Software (Software de Edición y Ruteo Avanzado), va a ser el encargado de realizar el enrutado del sistema de desarrollo. Esta es una herramienta de enrutado, ubicación y edición de componentes, que se utiliza para la fabricación de placas de circuito impreso, permitiendo editar generalmente, las capas superficiales (Top Copper), y de soldadura (Bottom Copper). Con Ares además se puede tener una visualización en 3D del PCB que se ha diseñado, al haber terminado de realizar la ubicación de piezas, capas y ruteo, con la herramienta "3D Visualization", en el menú output, la cual se puede demorar, solo haciendo los trazos un periodo de tiempo un poco más largo que el de los componentes, los cuales salen al empezar la visualización en 3D.

El programa cuenta con dos formas de hacer la ubicación y enrutado de las pistas que serán descritas a continuación.

4.3.1.1 Forma manual:

Esta forma es usada por el usuario cuando quiere realizar por si mismo la ubicación de cada componente. En esta se visualizara la cercanía entre cada uno de ellos para así ver



si se cruzan o no. Se debe tener en cuenta al DRC (verificador de las reglas de diseño). En este se realiza las diferentes configuraciones para llevar a cabo el diseño del circuito como lo son las capas en la que se hará el circuito impreso, entre otros.

4.3.1.2 Forma automática:

El propio programa puede trazar las pistas, si se guarda previamente el circuito en ISIS, y haciendo clic en el ícono de ARES, en el programa, el programa compone la Netlist en el cual se transfiere todos los componentes que fueron utilizados en el programa ISIS hacia el ARES.

4.3.1.2.1 Método con autorouter:

En este método se realizan los siguientes pasos:

- Poner solo los componentes en la placa.
- Especificar el área de la placa (Con un rectángulo, tipo “Board Edge”).
- Presionar autorouter.
- Editar las estrategias de ruteo.

4.3.1.2.2 Método Electra autorouter:

Utilizando el módulo Electra (Electra Auto Router), el cual, una vez colocados los componentes trazará automáticamente las pistas realizando varias pasadas para optimizar el resultado.



4.3.2 Materiales a utilizar:

La lista de materiales a utilizar es la siguiente:

- Diodo 1N4148.
- Diodo 5005A.
- Transistores 2N3904.
- Transistores 2N3906.
- Display de 7 segmentos.
- Inductor de 680uH.
- Lcd de 16x2.
- Conector ICSP.
- Capacitores cerámicos de 100nF.
- Condensadores electrolíticos de 47uF.
- Condensadores de 10uF.
- Pulsador.
- Interruptores.
- Resonador de 20MHz.
- Leds rojos.
- Leds verdes.
- Leds azules.
- Microcontrolador PIC 16F877.
- Microcontrolador PIC 18F2550.
- Potenciómetro.
- Conector USB.
- Base para la LCD.

- Resistencia de 10k Ω .
- Resistencia de 4.7k Ω .
- Resistencia de 2.7k Ω .
- Resistencia de 100k Ω .
- Resistencia de 100 Ω .
- Resistencia de 470 Ω .
- Resistencia de 330 Ω .
- Baquelita de 15x15 cm.
- Baquelita de 7x5 cm.
- Relé doble contacto.

4.3.3 Diagrama en PCB:

El siguiente programa presenta el diagrama en pcb elaborado para la realización del sistema de desarrollo en físico.

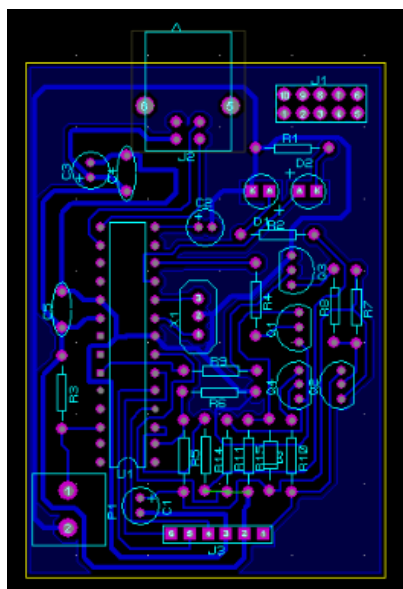


Figura 4.14 Ruteado de las pistas en el circuito impreso. Fuente: Propia.

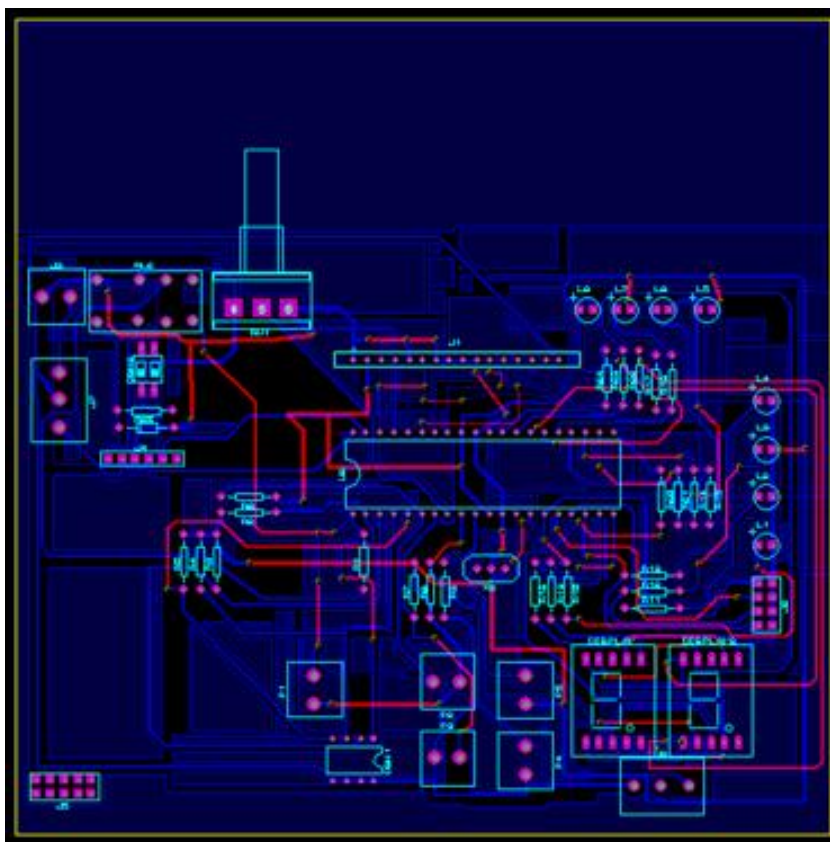


Figura 4.15 Ruteado de ambas capas en el circuito impreso. Fuente: Propia.

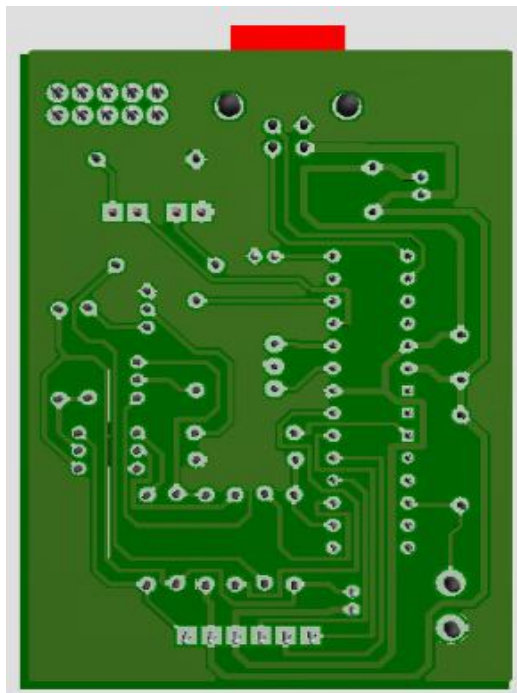


Figura 4.16 Ruteado y de las pista desde abajo. Fuente: Propia.

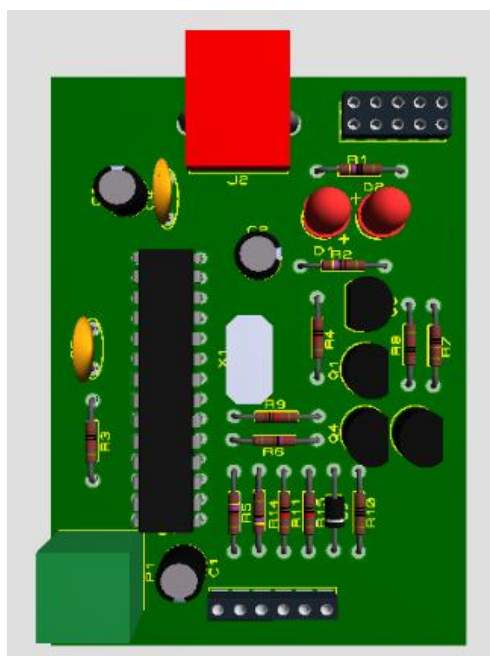


Figura: 4.17 Programador visto desde arriba en 3D. Fuente: Propia.

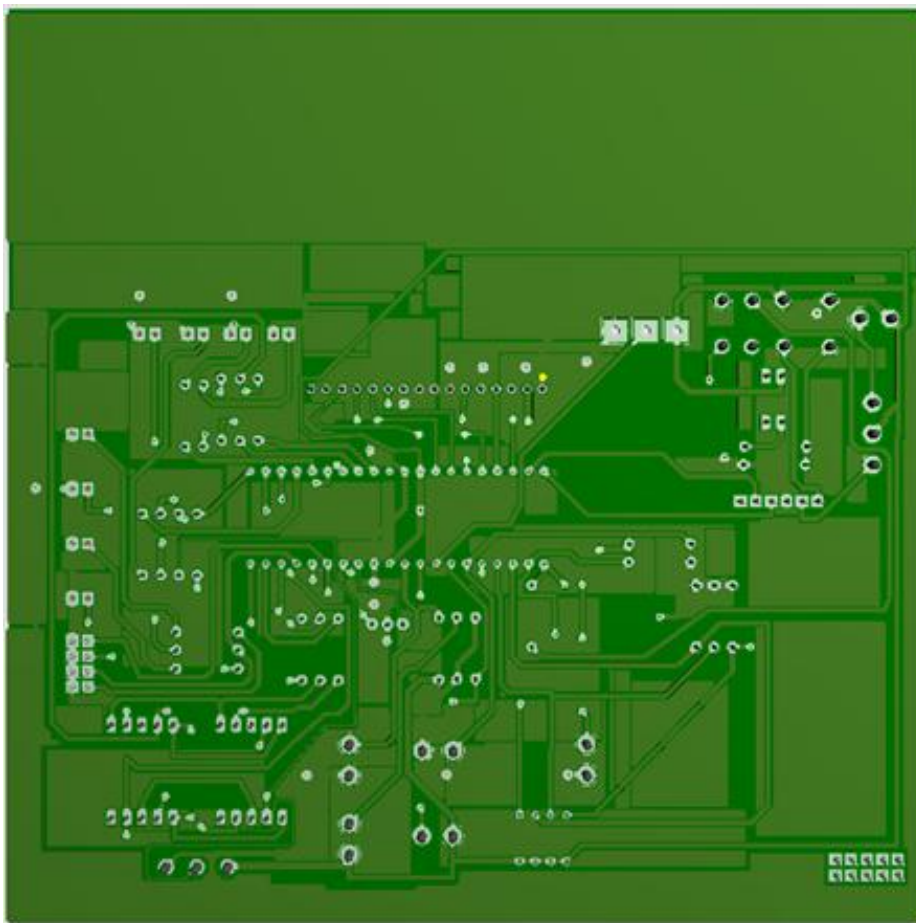


Figura 4.18 Sistema de desarrollo visualizado desde abajo en 3D. Fuente: Propia.

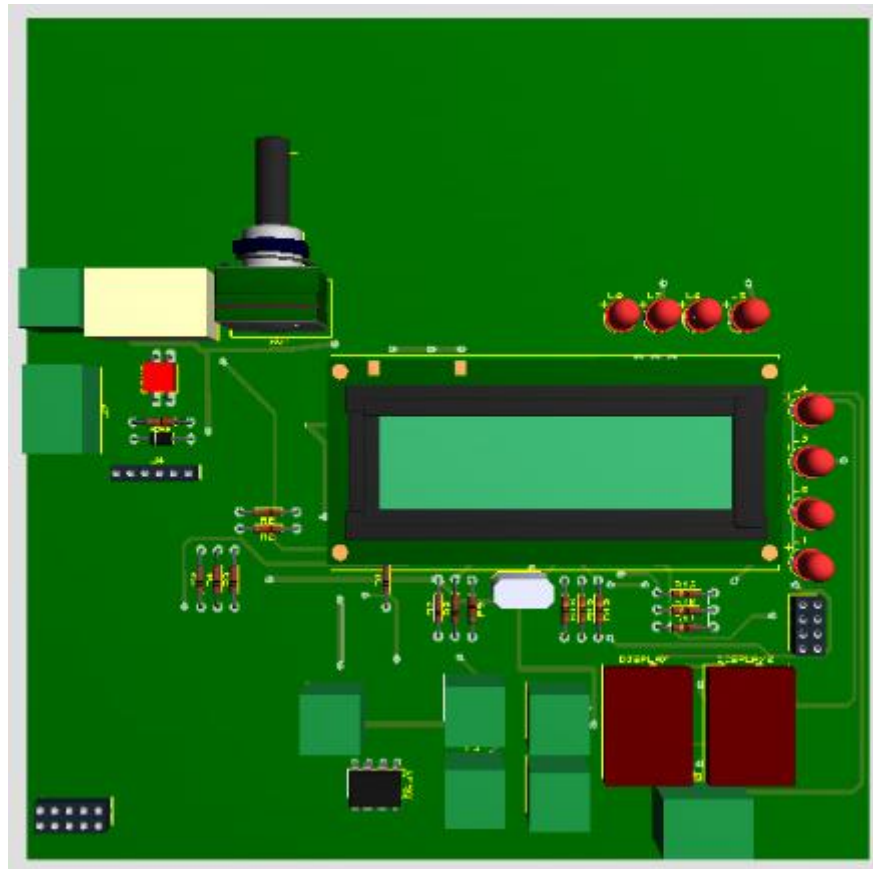


Figura 4.19 Sistema de desarrollo visto desde arriba en 3D. Fuente: Propia.

4.4 Montaje en físico del sistema de desarrollo:

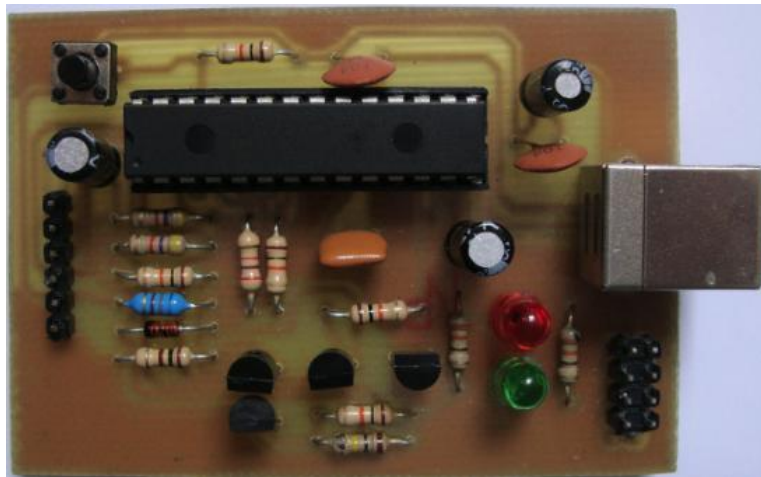


Figura 4.20 Foto del programador.

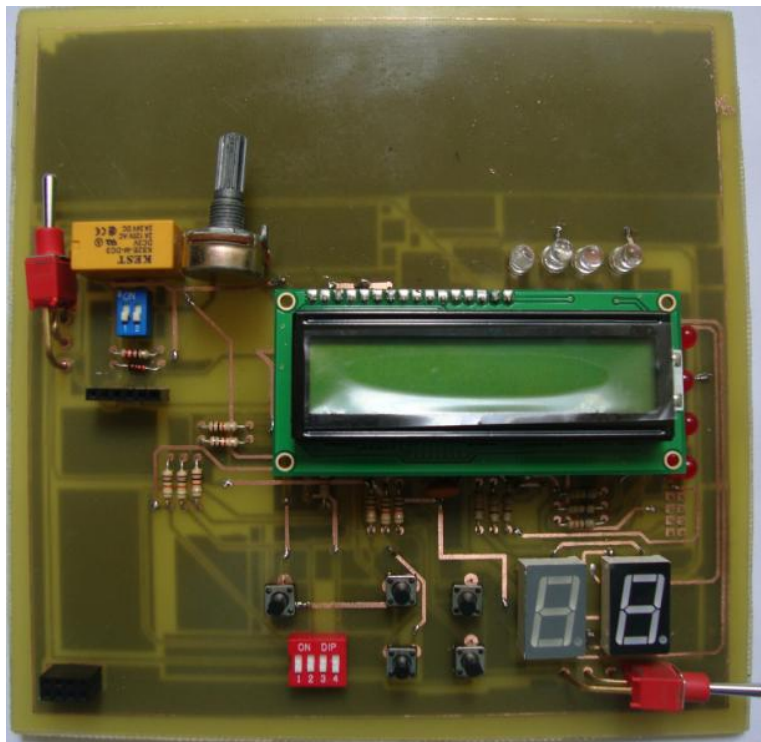


Figura 4.21 Foto del sistema de desarrollo.

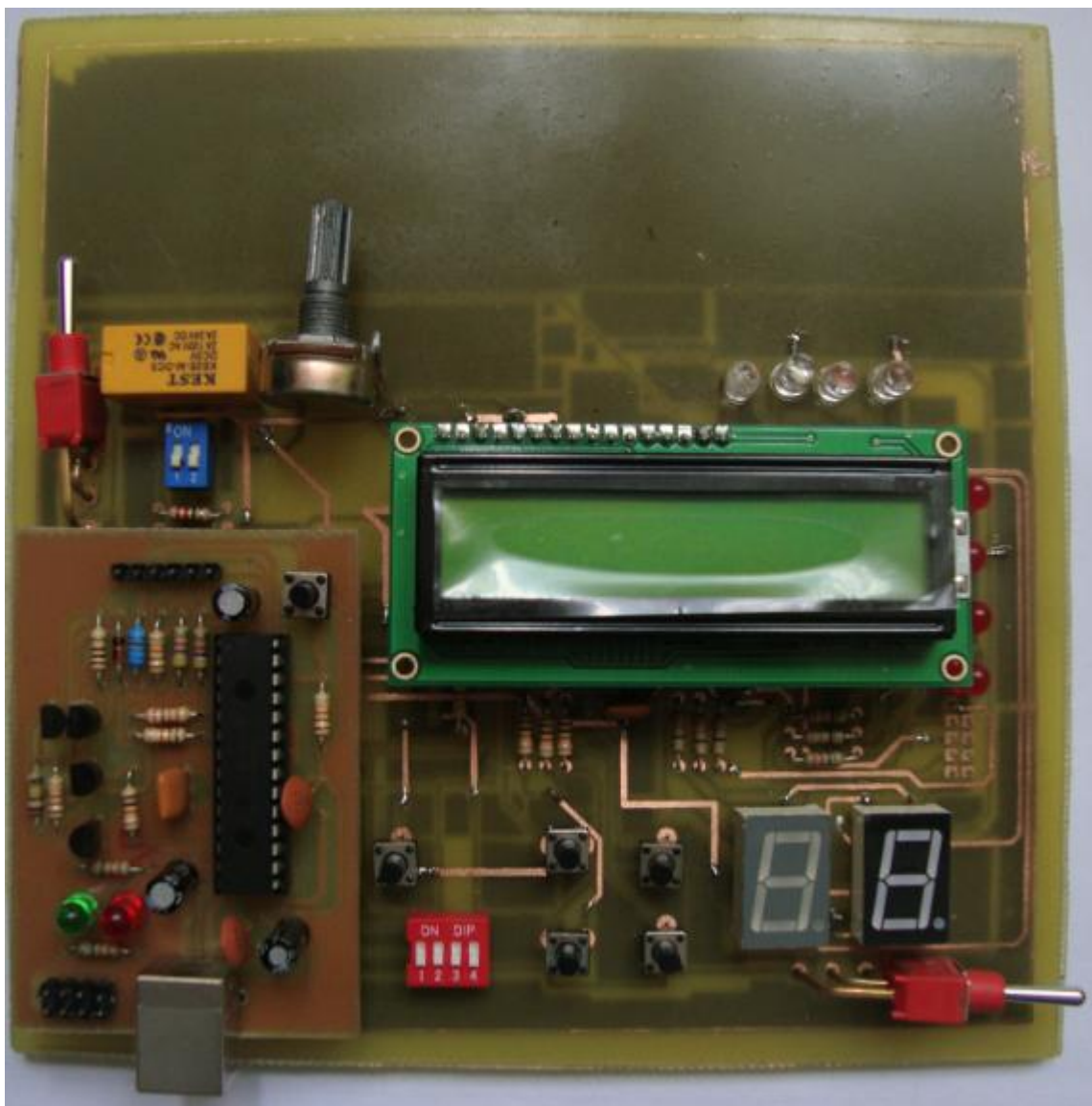


Figura 4.22 Foto final del sistema de desarrollo con programador incluido.



4.5 Manual de usuario:

El presente sistema de desarrollo es una herramienta poderosa, sólida y bien construida, de fácil uso para el usuario, la cual tiene todos los elementos necesario para el uso en el laboratorio de microprocesadores.

Es una unidad compacta de experimentación que permite explorar el mundo de los microcontroladores de una manera fácil y rápida utilizando al máximo el poder de estos dispositivos. Esta plataforma hace uso de ciertos componentes y dispositivos que serán nombrados a continuación:

- Un microcontrolador de clase media como lo es el PIC 16F877A.
- Un microcontrolador de clase media como lo es el PIC 18F2550.
- Leds.
- Display de 7 segmentos.
- Pulsadores e interruptores.
- Potenciómetros.
- Conector USB.
- Pantalla LCD de 2x16.
- Jumper de conexión para aplicaciones externas.

Las características más resaltantes de este sistema de desarrollo son:

- Interfaz de conexión del programador vía USB.
- Interfaz de programación vía ICSP.
- Pulsador de reset para reiniciar al microcontrolador.

- Alimentación del microcontrolador a 5V.
- Led para indicar la alimentación del sistema.
- Un led para indicar la transmisión de datos al momento de programar.
- Un conector para instalar una pantalla lcd.
- Puede programar directamente al microcontrolador en el mismo circuito.
- Zócalo para retirar el microcontrolador y cambiarlo por otro del mismo tipo.

4.5.1 Interfaz de programación:

La interfaz de programación se encarga de elevar los niveles de tensión del puerto usb del computador a los niveles necesarios para programación. Los componentes más resaltantes de esta interfaz de programación son:

- Un circuito que se encarga de elevar la tensión a nivel de programación que es de 13V.
- Un conector usb que se encarga de enviar los datos al microcontrolador PIC 18f2550 que es el encargado de la programación.
- Un diodo led que indica el estado de programación.

4.5.2 Fuente de alimentación:

La fuente de alimentación es proporcionada a través del conector USB que se encuentra conectado directamente al computador. Esto da como resultado que el sistema de desarrollo se encuentre conectado directamente a 5V de tensión. Los niveles de corriente suministrado por medio del computador es de 500mA por lo que el sistema funcionaria bajo niveles estables ya que su consumo estaría por el orden de 200mA aproximadamente.



4.5.3 Periféricos de E/S:

El sistema de desarrollo cuenta con periféricos de suma utilidad y muy utilizados en aplicaciones reales. Está compuesta por los diferentes dispositivos de entrada y salidas que fueron ya nombrados previamente. Los dispositivos están configurados de la siguiente manera:

El puerto A del microcontrolador PIC 16F877 será utilizado por un interruptor de 4 bits ubicados desde el pin RA0 hasta el pin RA3 y dos pulsadores ubicados en los pines RA4 y RA5.

El puerto B del microcontrolador será utilizado por una lcd de 2x16 que será ubicada desde el pin RB2 hasta el pin RB7. Estos son los dispositivos más llamativos que pueden incluir las aplicaciones con microcontroladores. Permite desplegar textos, números, mensajes y hasta caracteres graficas.

El puerto C del microcontrolador será utilizado por un banco de 8 leds que estarán configurados de la siguiente manera:

- 4 leds de luz roja desde el pin RC0 hasta el pin RC3.
- 4 leds de luz AZUL desde el pin RC4 hasta el pin RC7.

El puerto D del microcontrolador será utilizado por un display de 7 segmentos.

El puerto E será utilizado por un potenciómetro ubicado en el pin RE1 y 2 pulsadores ubicados en los pines RE1 y RE2.

Adicionalmente el puerto C tendrá un zócalo de 4 bits para aplicaciones externas.

4.5.4 Como usar el Sistema de Desarrollo:

4.5.4.1 Usando el Software Pickit 2

- Luego de haber terminado el código en el compilador grabar e iniciar la aplicación Pickit 2.
- El programa pickit 2 autodetecta al Pic 16F877A siempre que el mismo esté listo para programar. Para iniciar el modo programación el Switch 1 debe estar en ON como indica la Figura 4.23.



Figura 4.23 Selector de Programación.

- Si el PIC no ha sido detectado hacerlo manualmente manteniendo el switch 1 en ON como indica la figura 4.24.

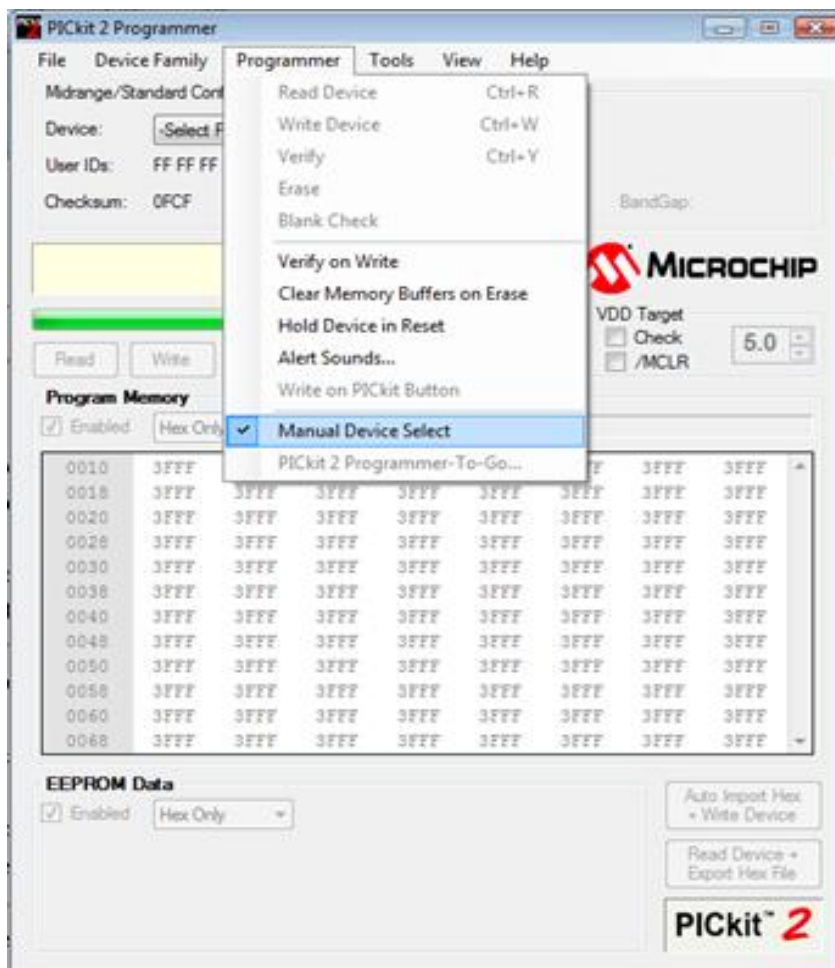


Figura 4.24 Detección del dispositivo de manera manual.

Luego de esto seleccionar el dispositivo con el que quiere trabajar como muestra la figura 4.25.

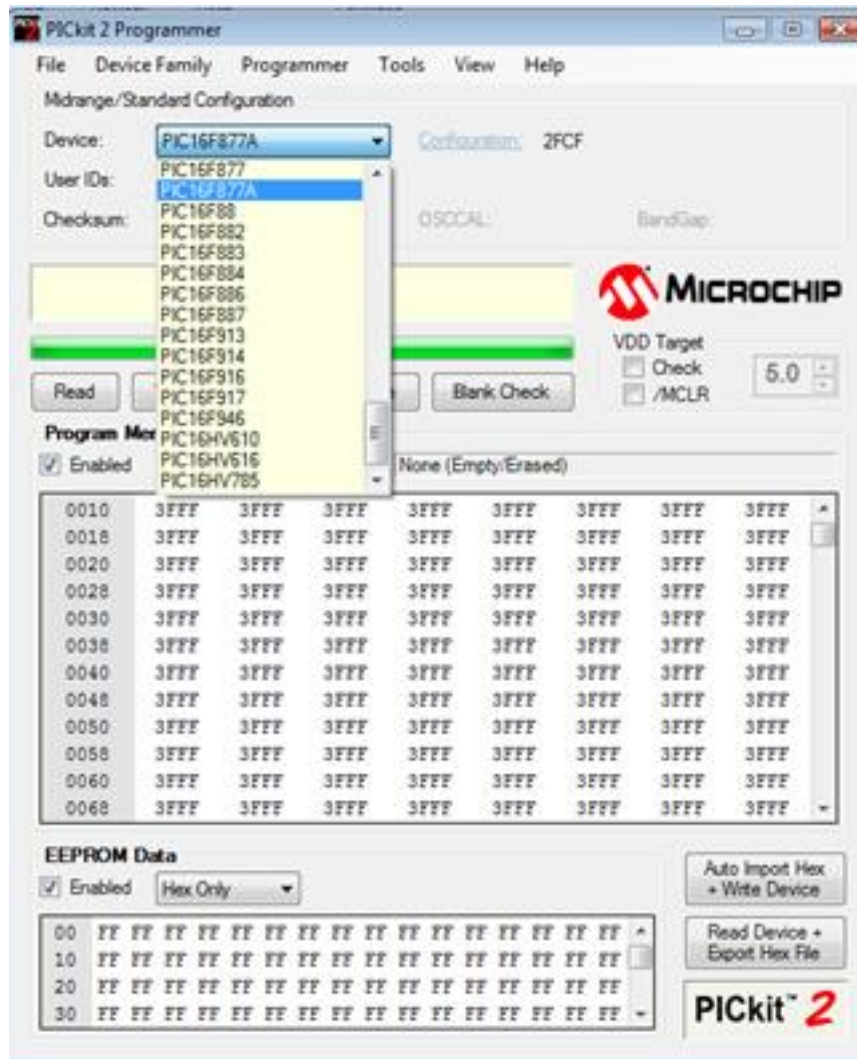


Figura 4.25 Selector de dispositivo de manera manual.

- Después de haber detectado el dispositivo se procede a grabar el PIC. Esto se hace importando el archivo .HEX generado por el programa como se muestra en la figura 4.26. Nota: Si el archivo fue generado por MPLAB se debe añadir la siguiente configuración en los FUSES y exportar el archivo.

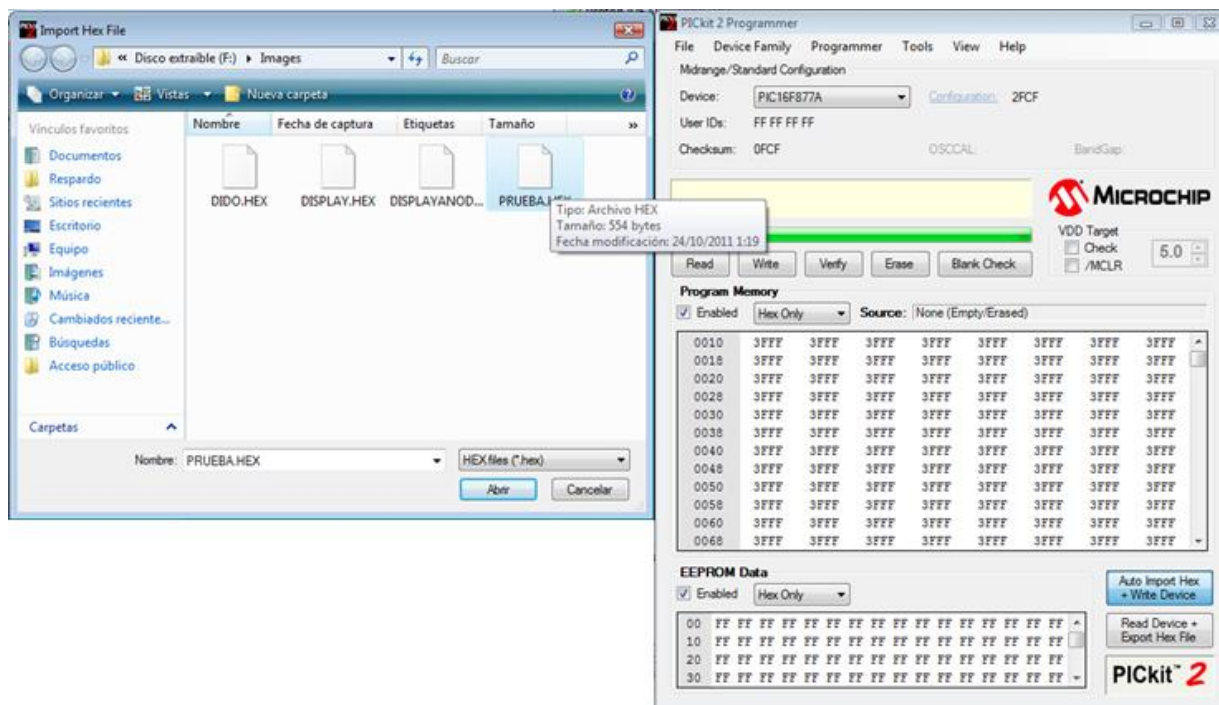


Figura 4.26 Como se importa el archivo .HEX por el software Pickit 2.

- Una vez realizado estos pasos el PIC quedara grabado y listo para usarse.
- Después de programar devolver el switch a la posición off.

4.5.4.2 Usando el modo debugger del MPLAB

- Para el uso de este modo se debe asegurar que el dispositivo seleccionado sea el PIC 16F877A. La forma de hacerlo se puede ver en la figura 4.27.

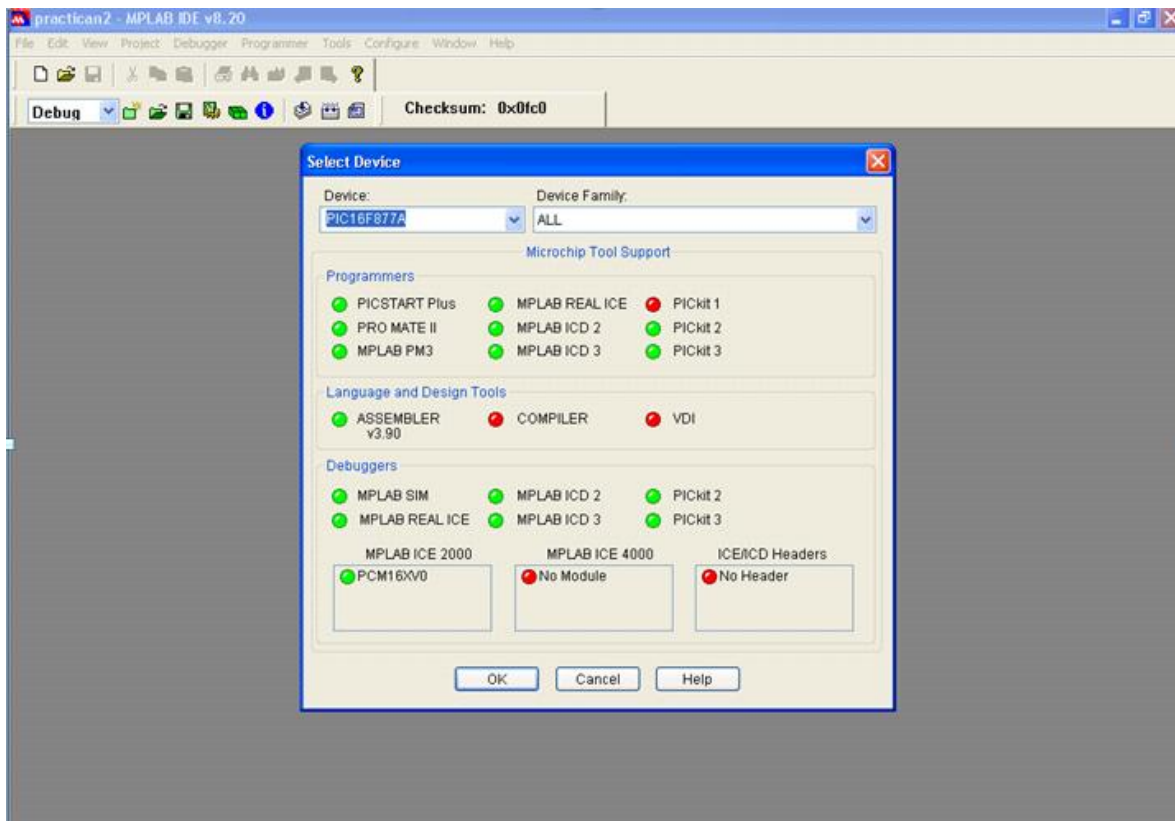


Figura 4.27 Selección del dispositivo a programar.

- Después de haber realizado la selección del dispositivo se tiene que configurar los FUSES. Esto se hace agregando la siguiente línea:
__CONFIG
_WDT_OFF&_PWRTE_ON&_HS_OSC&_LVP_OFF&_CP_OFF
- Escoger el programador Pickit 2 como es mostrado en la figura 4.28.

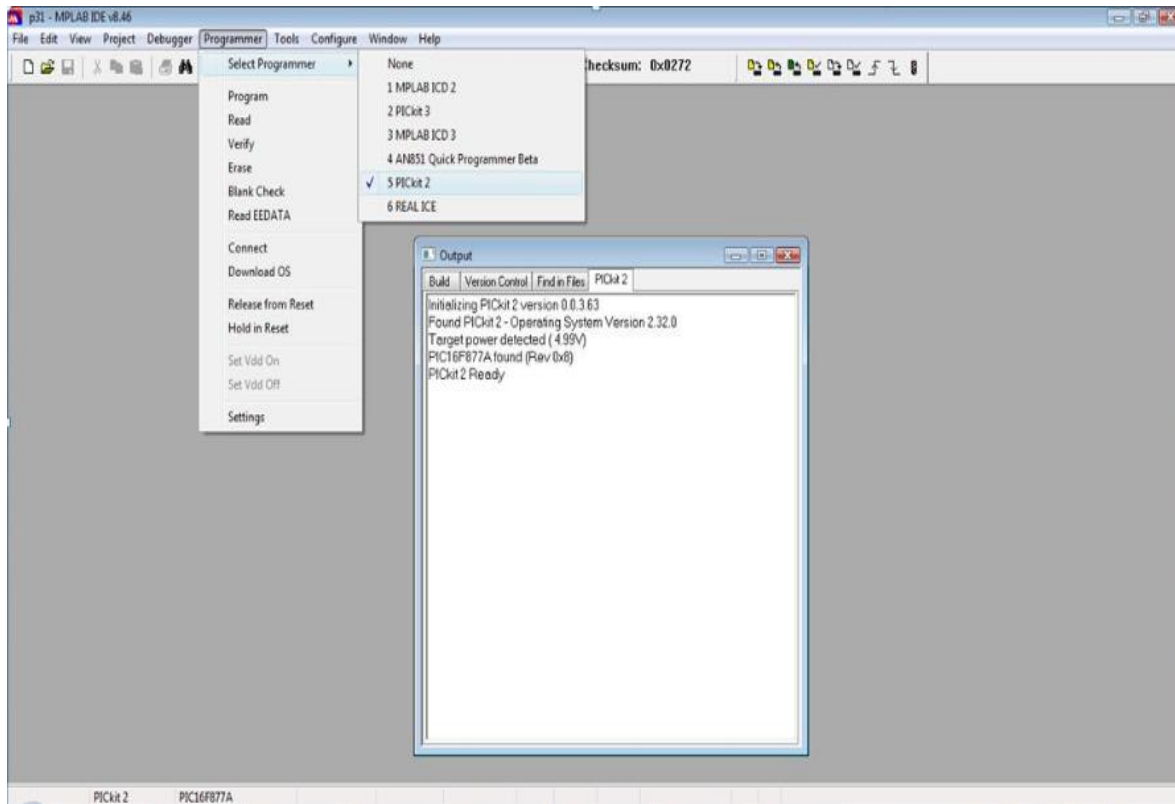


Figura 4.28 Tipo de programador a escoger en nuestro caso Pickit2.

- Luego de esto el pic estará listo para grabarse pulsando el icono mostrado en la figura 4.29.

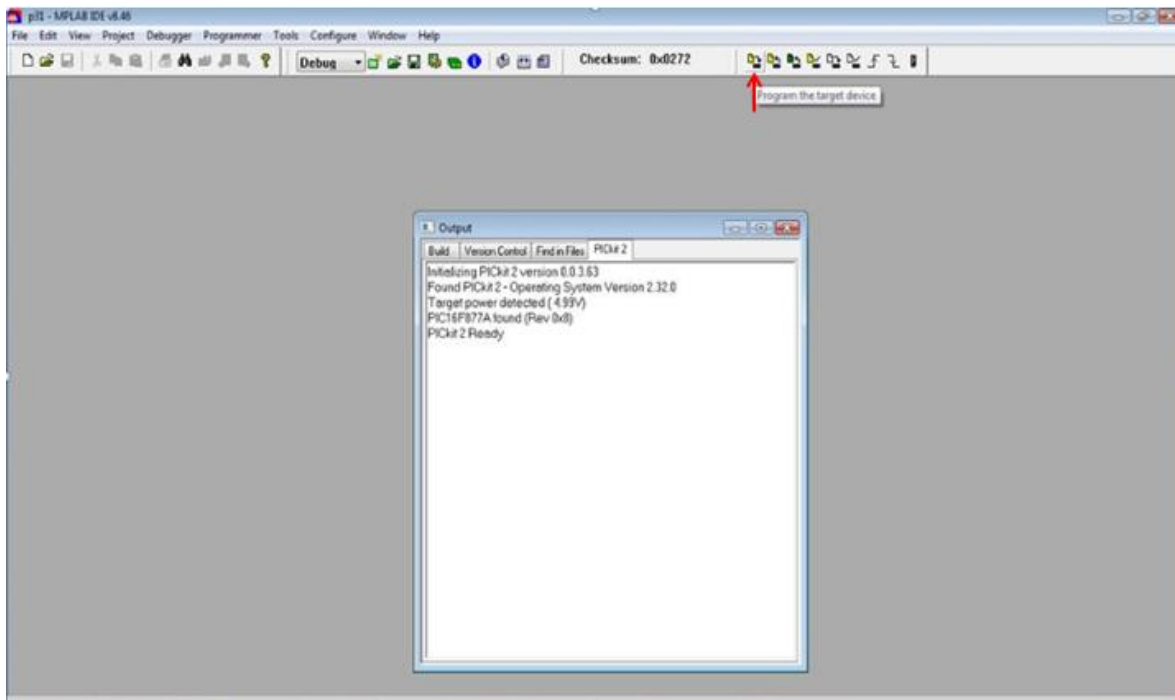


Figura 4.29 Icono para empezar la programación.



4.5.5 Descripción del microcontrolador PIC 16F877:

Este microcontrolador es fabricado por MicroChip familia a la cual se le denomina PIC. El modelo 16F877 posee varias características que hacen a este microcontrolador un dispositivo muy versátil, eficiente y práctico para ser empleado en diferentes tipos de aplicaciones de complejidad media y grande.

Las principales características con que cuenta el 16F877 son:

- Procesador de arquitectura RISC avanzada.
- Juego de 35 instrucciones con 14 bits de longitud. Todas ellas se ejecutan en un ciclo de instrucción menos las de salto que tardan 2.
- Frecuencia de 20 Mhz.
- Hasta 8K palabras de 14 bits para la memoria de código, tipo flash.
- Hasta 368 bytes de memoria de datos RAM.
- Hasta 256 bytes de memoria de datos EEPROM.
- Hasta 14 fuentes de interrupción internas y externas.
- Pila con 8 niveles.
- Modos de direccionamiento directo, indirecto y relativo.
- Perro guardián (WDT).
- Código de protección programable.
- Modo Sleep de bajo consumo.
- Programación serial sobre el circuito (ICSP: In Circuit Serial Programming) a través de dos pines.



4.5.5.1 Descripción del dispositivo:

A continuación se presentara un esquema general de funcionamiento del microcontrolador PIC 16F877. En el diagrama de la figura 4.6.5 podemos identificar la memoria del programa en la parte superior izquierda con 8K posiciones por 14 bits, también esta presenta la memoria de datos (RAM) de 368 posiciones por 8 bits. La memoria EEPROM 256 posiciones x 8 bits. El procesador propiamente dicho esta formado por la ALU (unidad aritmética lógica) el registro de trabajo W. Tenemos los periféricos I/O Port A, B, C, D, E, el TMR0 (temporizador contador de eventos), TMR1 y TMR2 entre otros módulos. También contamos con un registro de instrucción que se carga cada vez que la ALU solicita una nueva instrucción a procesar. En la parte intermedia encontramos algunos bloques como son el Status Reg que es el registro de estado encargado de anotar el estado actual del sistema, cada vez que se ejecuta una instrucción se llevan a cabo cambios dentro del microcontrolador como desborde, acarreo, etc. Cada uno de esos eventos esta asociado a un bit de este registro. Existe un registro de vital importancia que se llama el Program Counter o contador de programa este registro indica la dirección de la instrucción a ejecutar. El registro en cuestión no es necesariamente secuencial, esto es no se incrementa necesariamente de uno en uno ya que puede darse el caso en el que salte dependiendo si hay una instrucción de bifurcación de por medio o puede haber alguna instrucción de llamada a función y/o procedimiento. También observamos el bloque de la pila, la función de la pila es ser un buffer temporal en el que se guarda el contador de programa cada vez que se suscita una llamada a un procedimiento y/o función (incluyendo interrupciones). Por tanto el nivel de anidamiento es de hasta 8 llamadas. También esta presente el FSR reg. Que es el registro que cumple una función similar a la del contador de programa direccionando en este caso la RAM, el FSR es un puntero a una dirección de la RAM. La aparición de multiplexores se debe a que los datos pueden tener diferentes fuentes. Cuando programamos el microcontrolador debemos siempre tener en mente que es lo que el hace.



Cuando lo prendemos asume un valor por defecto, el contador de programa asume la posición cero por tanto el microcontrolador toma la instrucción que se encuentra en esa posición en la memoria de programa y la ejecuta. Al momento de ejecutarla procede a informar si se ha llevado a cabo alguna operación en particular registrando la en el registro de estado (STATUS). Si la instrucción es de salto o bifurcación evaluará las condiciones para saber si continua o no con la siguiente instrucción, en caso que no sea así saltará a otra posición de memoria. En caso el programa haga un llamado a una función guardará en la pila el valor del contador de programa ejecutará la rutina y al momento que termina restituirá el valor correspondiente para seguir con la siguiente instrucción.

4.5.5.2 Organización de la memoria del programa:

La memoria se divide en memoria de datos y programa véase la figura 4.6.6. La de datos a su vez se divide en: SFR (Special Function Register) Registros de propósito especial, son registros que ayudan a configurar el hardware interno así como sirven para escribir o leer valores de los diferentes componentes que constituyen el microcontrolador. Por ejemplo el registro “trisa” que nos permite configurar el modo de trabajo de las líneas del puerto A. GFR (General Function register) Registros de propósito general, son posiciones de memoria que podemos usar para almacenar valores que emplean nuestros programas. Debe señalarse también que la memoria de datos se divide en cuatro bancos (esto para el caso específico del 16F87X). Las posiciones bajas siempre están reservadas para los SFR en tanto que las altas para los GFR. También tenemos una memoria EEPROM, con 256 posiciones, para acceder a la memoria no podemos leer o escribir directamente es decir colocar la dirección y obtener o dejar el valor. Para trabajarla debemos apoyarnos en registros adicionales de tal forma que la usamos indirectamente

4.5.5.3 Diagrama de conexionado:

A continuación se presenta el esquema de conexionado del microcontrolador PIC 16F877.

Tabla 4.5 Diagrama de conexionado del microcontrolador PIC 16F877.

NOMBRE DEL PIN	PIN	TIPO	TIPO DE BUFFER	DESCRIPCIÓN
OSC1/CLKIN	13	I	ST/MOS	Entrada del oscilador de cristal / Entrada de señal de reloj externa
OSC2/CLKOUT	14	O	-	Salida del oscilador de cristal
MCLR/Vpp/THV	1	I/P	ST	Entrada del Master clear (Reset) o entrada de voltaje de programación o modo de control high voltaje test
RA0/AN0	2	I/O	TTL	PORTA es un puerto I/O bidireccional
RA1/AN1				RA0: puede ser salida analógica 0
RA2/AN2/ Vref-				RA1: puede ser salida analógica 1
RA3/AN3/Vref+				RA2: puede ser salida analógica 2 o referencia negativa de voltaje
RA4/T0CKI				RA3: puede ser salida analógica 3 o referencia positiva de voltaje
RA5/SS/AN4	6	I/O	ST	RA4: puede ser entrada de reloj el timer0. RA5: puede ser salida analógica 4 o el esclavo seleccionado por el puerto serial síncrono.
	7	I/O	TTL	
RBO/INT	33	I/O	TTL/ST	PORTB es un puerto I/O bidireccional. Puede ser programado todo como entradas
RB1				RB0 puede ser pin de interrupción externo.
RB2				
RB3/PGM				RB3: puede ser la entada de programación de bajo voltaje
RB4				Pin de interrupción
RB5				Pin de interrupción
RB6/PGC				Pin de interrupción. Reloj de programación serial
RB7/PGD	38	I/O	TTL	



CAPITULO IV: DESARROLLO DE LA PLATAFORMA



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

Fuente: <http://es.scribd.com/doc/27802588/Descripcion-del-PIC-16F877>



4.5.5.4 Periféricos de entrada y salidas del pic 16F877:

4.5.5.4.1 Temporizador timer 0:

El módulo Timer0 puede ser usado para generar periodos de tiempo (Si funciona como temporizador) o puede ser usado para registrar el paso de eventos (si trabaja como contador de eventos). La figura 4.31 muestra el diagrama de bloques del timer0.

Las principales características del modulo timer 0 son:

- Puede ejecutar hasta 256 cuentas (0-255) debido a que el registro TMR0 es de 8 bits.
- El registro TMR0 puede ser leído para saber cual es valor actual de las cuentas o puede ser escrito para colocar un valor inicial.
- Posee un bloque de preescalamiento que permite ampliar el rango de las cuentas.
- Selector interno para definir si trabaja como temporizador o como contador de eventos.
- Genera una señal de interrupción cuando se produce un desborde en el registro TMR0 (cuando pasa de 0xFF a 00). Bit T0IF del registro INTCON.
- Selector para indicar si detecta flanco de subida o bajada cuando opera como contador de eventos. Bit T0SE del registro OPTION_REG (OPTION para fines prácticos lo llamaremos OPTION_REG en adelante).



La parte superior derecha está presente un MUX. El MUX es controlado por el bit TOCS, si es 0 el módulo opera como temporizador; si es 1 como contador de eventos. El bit TOCS pertenece al registro OPTION_REG y es el bit 5.

Cuando el módulo funciona como timer el registro TMR0 se incrementa en cada ciclo de instrucción (siempre y cuando el Preescalamiento este deshabilitado). Cuando el registro TMR0 es escrito el microcontrolador debe esperar 2 ciclos de instrucción para que comenciar la cuenta.

Cuando el módulo trabaja como contador de eventos el registro TMR0 incrementa su valor cada vez que aparece un flanco en el pin RA4/TOCKI. La selección del tipo de flanco (subida o bajada) dependerá de la programación del bit TOSE (registro OPTION_REG bit 4). Si TOSE es 0-lógico trabaja con flanco de subida si es 1-lógico con flanco de bajada.

El preescaler es un módulo compartido por el WATCHDOG y el Timer 0. El preescaler es como un divisor de frecuencia programable. Como se aprecia puede conectarse en serie al modulo timer 0 (dependiendo de los valores de TOCS y PSA). Supongamos que el módulo timer 0 funciona como contador de eventos y el preescaler esta habilitado. El valor es 1:1, eso significa que cada pulso que ingrese incrementará el valor en el registro TMR0 (el valor máximo de cuentas será 256). Si el preescaler vale 1:8 por cada 8 eventos que sucedan solo aumentará una cuenta en el registro TMR0 (el valor máximo de cuentas será 8x256). El valor del preescaler depende de los bits PS2 (bit 2), PS1(bit 1) y PS0 (bit 0) del registro OPTION_REG. El bit PSA (bit 3) del registro INTCON define si el el preescaler funciona con el Watchdog o con el Timer 0.

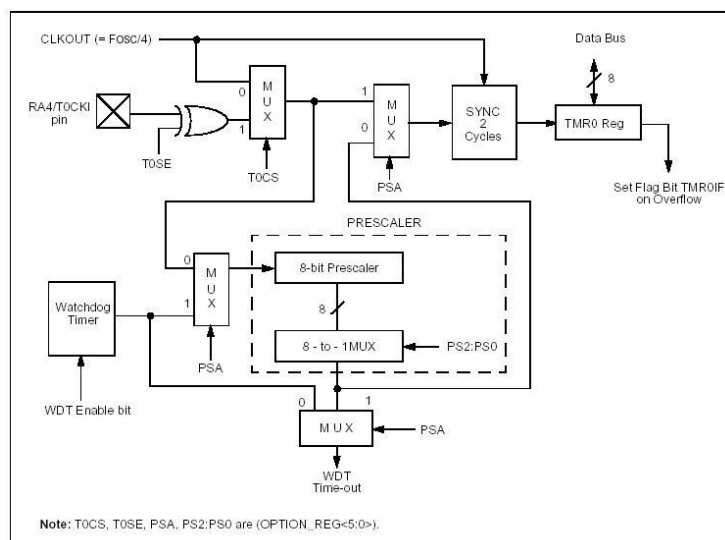


Figura 4.31 Diagrama de bloques del timer0. Fuente:
<http://www.alos.5u.com/pic16f877/Modulo2.html>

4.5.5.4.2 Temporizador timer 1:

El TMR1 es un Temporizador/Contador con un tamaño de 16 bits, lo que requiere el uso de dos registros concatenados de 8 bits: TMR1H : TMR1L, que son los encargados de guardar el valor del conteo en cada momento. Dicho valor evoluciona desde 0000h hasta

FFFFh. Momento en el cual se activa el señalizador TMR1IF y se regresa al valor inicial 0000h.

El valor contenido en TMR1H : TMR1L puede ser leído o escrito y los impulsos de reloj que originan el conteo ascendente pueden provenir del exterior o de la frecuencia de funcionamiento del microcontrolador ($F_{osc}/4$).



El Timer1 tiene tres formas de funcionamiento:

- Temporizador.
- Contador Síncrono.
- Contador Asíncrono.

Como Temporizador el valor concatenado TMR1H : TMR1L se incrementa con cada ciclo de instrucción ($F_{osc}/4$). En el modo contador, el incremento se puede producir con los flancos ascendentes de un reloj externo, cuya entrada se aplica a las líneas RC0 y RC1 de la puerta C, o por impulsos aplicados en la línea RC0.

4.5.5.4.2.1 Operación del Timer1 en modo Temporizador:

Este modo es seleccionado limpiando el bit TMR1CS (T1CON<1>). En este modo la entrada de reloj al timer es $F_{osc}/4$. El bit T1SYNC (T1CON<2>) no tiene efecto desde el reloj interno; es siempre síncrono.

4.5.5.4.2.2 Timer1 en modo Contador Síncrono:

Este modo es seleccionado seteando el bit TMR1CS. En este modo el timer incrementa cada flanco de subida de una entrada de reloj externo en el pin RC1, cuando el bit T1OSCEN es seteado. O cuando una entrada de reloj ingresa por el pin RC0, cuando el bit T1OSCEN es limpiado. Si el T1SYNC es limpiado, entonces el reloj externo es sincronizado con la fase del reloj interno.



4.5.5.4.2.3 Timer1 en modo Contador Asíncrono:

Si el bit T1SYNC (T1CON <2>) es seteado, el reloj externo no es sincronizado. El timer continúa un incremento asíncrono a la fase del reloj interno. El timer continuará funcionando durante el modo SLEEP y podrá generar una interrupción de desborde; el cual podría despertar al procesador del modo SLEEP.

4.5.5.4.3 Temporizador timer 2:

El Timer 2 del PIC 16F877 es de 8 bit, tiene un Prescaler y un Postscaler, puede ser usado como base de tiempo del modulo PWM. El registro de conteo TMR2 es leible y escribible, y es limpiado por cualquier Reset.

El Timer 2 tiene un registro de Periodo PR2 que a diferencia de los anteriores que terminaban en FFh o FFFFh utiliza este registro el cual le indica al Timer 2 cuando reiniciarse utilizando un comparador interno. PR2 tiene un FFh al Reset.

Es importante ver y entender el diagrama de bloques del Timer 2 que se muestra en la figura 4.32.

En el se puede ver que al estar contando los ciclos de reloj $F_{osc}/4$, después del prescaler seleccionable, se va incrementando el registro TMR2, el resultado entra al comparador que continuamente hace su trabajo comparando con PR2 (FFh al inicio). El resultado del comparador pasa por el Postscaler seleccionable el cual se encarga de poner la bandera en un momento determinado.

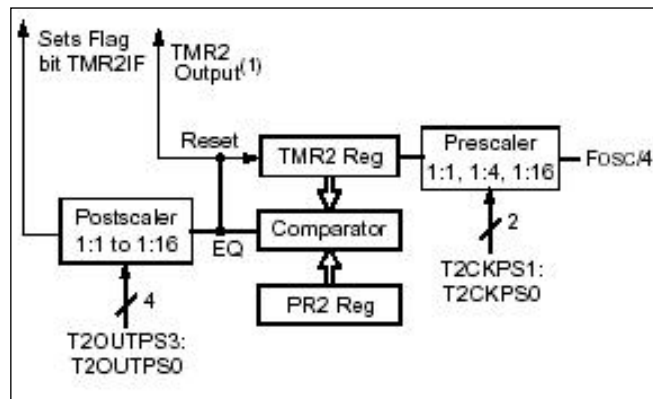


Figura 4.32 Diagrama de bloques del temporizador timer 2. Fuente:

<http://picmicrocontroller877.blogspot.com/2007/10/timer-2-del-pic-16f877a.html>

4.5.5.4.4 Conversor A/D:

El módulo convertidor Análogo Digital (A/D) del PIC 16F877 tiene 8 canales de entrada. La conversión de la señal analógica aplicada (a uno de los canales) se plasma en número binario de 10 dígitos. El módulo A/D posee voltajes de referencia que pueden ser seleccionados para emplear las tensiones VDD, VSS del microcontrolador o puede emplear tensiones aplicadas a los pines RA2 o RA3 (incluso es posible establecer combinaciones de los anteriores valores).

- Registro de resultado de byte alto de la conversión A/D (ADRESH). Banco 0, 0x1E.
- Registro de resultado de byte bajo de la conversión A/D (ADRESL). Banco 1, 0x9E.
- Registro 0 de control del módulo A/D (ADCON0). Banco 0, 0x1F.
- Registro 1 de control del módulo A/D (ADCON1). Banco 1, 0x9F.

El detalle del registro ADCON0 se muestra en la figura 4.33 a continuación:

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	U-0	R/W-0
ADCS1	ADCS0	CHS2	CHS1	CHS0	GO/DONE	—	ADON
bit 7							bit 0

bit 7-6 ADCS1:ADCS0: A/D Conversion Clock Select bits (ADCON0 bits in bold)

ADCON1 <ADCS2>	ADCON0 <ADCS1:ADCS0>	Clock Conversion
0	00	FOSC/2
0	01	FOSC/8
0	10	FOSC/32
0	11	FRC (clock derived from the internal A/D RC oscillator)
1	00	FOSC/4
1	01	FOSC/16
1	10	FOSC/64
1	11	FRC (clock derived from the internal A/D RC oscillator)

bit 5-3 CHS2:CHS0: Analog Channel Select bits

000 = Channel 0 (AN0)
001 = Channel 1 (AN1)
010 = Channel 2 (AN2)
011 = Channel 3 (AN3)
100 = Channel 4 (AN4)
101 = Channel 5 (AN5)
110 = Channel 6 (AN6)
111 = Channel 7 (AN7)

Note: The PIC16F873A/876A devices only implement A/D channels 0 through 4; the unimplemented selections are reserved. Do not select any unimplemented channels with these devices.

bit 2 GO/DONE: A/D Conversion Status bit

When ADON = 1:

1 = A/D conversion in progress (setting this bit starts the A/D conversion which is automatically cleared by hardware when the A/D conversion is complete)
0 = A/D conversion not in progress

bit 1 Unimplemented: Read as '0'

bit 0 ADON: A/D On bit

1 = A/D converter module is powered up
0 = A/D converter module is shut-off and consumes no operating current

Figura 4.33 Registro de control de ADCON1 del PIC 16F877. Fuente: Hoja técnica del PIC 16F877. Microchip.

Y el detalle del registro ADCON1 se muestra en la figura 4.34 a continuación:

R/W-0	R/W-0	U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0
ADFM	ADCS2	—	—	PCFG3	PCFG2	PCFG1	PCFG0
bit 7							bit 0

bit 7 **ADFM**: A/D Result Format Select bit

1 = Right justified. Six (6) Most Significant bits of ADRESH are read as '0'.

0 = Left justified. Six (6) Least Significant bits of ADRESL are read as '0'.

bit 6 **ADCS2**: A/D Conversion Clock Select bit (ADCON1 bits in shaded area and in bold)

ADCON1 <ADCS2>	ADCON0 <ADCS1:ADCS0>	Clock Conversion
0	00	FOSC/2
0	01	FOSC/8
0	10	FOSC/32
0	11	FRC (clock derived from the internal A/D RC oscillator)
1	00	FOSC/4
1	01	FOSC/16
1	10	FOSC/64
1	11	FRC (clock derived from the internal A/D RC oscillator)

bit 5-4 **Unimplemented**: Read as '0'

bit 3-0 **PCFG3:PCFG0**: A/D Port Configuration Control bits

bit 3-0 **PCFG3:PCFG0**: A/D Port Configuration Control bits

PCFG <3:0>	AN7	AN6	AN5	AN4	AN3	AN2	AN1	AN0	VREF+	VREF-	C / R
0000	A	A	A	A	A	A	A	A	VDD	VSS	8 / 0
0001	A	A	A	A	VREF+	A	A	A	AN3	VSS	7 / 1
0010	D	D	D	A	A	A	A	A	VDD	VSS	5 / 0
0011	D	D	D	A	VREF+	A	A	A	AN3	VSS	4 / 1
0100	D	D	D	D	A	D	A	A	VDD	VSS	3 / 0
0101	D	D	D	D	VREF+	D	A	A	AN3	VSS	2 / 1
011x	D	D	D	D	D	D	D	D	—	—	0 / 0
1000	A	A	A	A	VREF+	VREF-	A	A	AN3	AN2	6 / 2
1001	D	D	A	A	A	A	A	A	VDD	VSS	6 / 0
1010	D	D	A	A	VREF+	A	A	A	AN3	VSS	5 / 1
1011	D	D	A	A	VREF+	VREF-	A	A	AN3	AN2	4 / 2
1100	D	D	D	A	VREF+	VREF-	A	A	AN3	AN2	3 / 2
1101	D	D	D	D	VREF+	VREF-	A	A	AN3	AN2	2 / 2
1110	D	D	D	D	D	D	D	A	VDD	VSS	1 / 0
1111	D	D	D	D	VREF+	VREF-	D	A	AN3	AN2	1 / 2

A = Analog input D = Digital I/O

C / R = # of analog input channels / # of A/D voltage references

Figura 4.34 Registro de control del ADCON1 del PIC 16F877. Fuente: Hoja técnica del PIC 16F877. Microchip.



El registro ADCON1 configura las funciones de los pines de entrada al módulo. Como se aprecia se puede configurar los pines del puerto A como entradas analógicas inclusive la línea RA3 puede funcionar como el voltaje de referencia.

Los registros ADRESH:ADRESL contienen el resultado de la conversión (10 bits). Cuando se ha completado una conversión el resultado es almacenado en ADRESH:ADRESL y además el bit GO/-DONE (registro ADCON bit 2) se pone a 0-lógico y el bit ADIF (registro PIR1 bit 7) se pone como 1-lógico.

Después que el módulo A/D ha sido configurado, es necesario esperar un periodo de tiempo para que la señal sea adquirida (antes de las conversiones empiecen). Cada uno de los canales de entrada tiene su correspondiente bit de configuración en los registros TRIS y estos han de ser puestos como entradas.

Una vez que el periodo de adquisición ha terminado la conversión A/D puede empezar. Los siguientes pasos muestran la secuencia de uso:

1. Configure el módulo A/D: (lo cual significa)
 - Configurar los pines de entrada de los canales analógicos a usar. Configure los voltajes de referencia. (en el registro ADCON1).
 - Seleccionar el canal de entrada al módulo A/D (en el registro ADCON0).
 - Seleccionar el clock de conversión A/D (en el registro ADCON0).
 - Poner a funcionar el módulo A/D (en el registro ADCON0).



2. Configure la interrupción del A/D si lo desea:

- ADIF=0 –lógico (bit que indica si se produjo una conversión).
- ADIE=1-lógico (habilitador de interrupción del modulo analógico).
- PEIE=1 lógico (habilitador de interrupción de periféricos).
- GIE=1-lógico (Habilitador general de interrupciones).

3. Espere por el tiempo de adquisición (es el tiempo que se demora en trabajar el bloque de sampling and hold)

4. Comienzo de la conversión

- GO/-DONE=1-lógico (en el registro ADCON0).

5. Esperar a que la conversión se complete. Para saber si la conversión termino podemos:

- Revisar el bit GO/-DONE esperando que sea de nuevo 0 –lógico.
- esperar al flag de ADIF sea 1-lógico (puede emplearse como interrupción).

6. Leer el resultado del A/D en el par de registros (ADRESH:ADRESL). No debemos olvidar colocar el bit ADIF a 0-lógico (si se requiere)

Para la siguiente conversión, regrese al paso 1 o paso 2 dependiendo si solo se usa uno o más canales.



4.5.5.4.5 Puerto de comunicación serial síncrona/asíncrona:

Entre las herramientas que disponen los PIC16F8x se encuentra el USART, llamado SCI (Serial Communications Interface), puede funcionar como un sistema de comunicación bidireccional, adaptándose a multitud de periféricos y dispositivos que transfieren información de forma serial, tales como un ordenador. También puede trabajar en modo unidireccional para soportar periféricos como memorias, conversores, etc.

El USART (Transmisor/Receptor Síncrono/Asíncrono Serie) puede trabajar de dos formas:

- Asíncrono (Bidireccional).
- Síncrono (Maestro).
- Síncrono (Esclavo).

El módulo USART también tiene una capacidad de comunicación multi-procesador con 9 bits de detección de dirección.

A continuación se muestran los registros asociados al USART. Para información adicional consulte la hoja técnica de PIC 16F877.

En la figura 4.35 se presenta el registro de control y transmisión TXSTA del PIC 16F877.

	R/W-0	R/W-0	R/W-0	R/W-0	U-0	R/W-0	R-1	R/W-0
	CSRC	TX9	TXEN	SYNC	—	BRGH	TRMT	TX9D
	bit 7							bit 0
bit 7	CSRC: Clock Source Select bit <u>Asynchronous mode:</u> Don't care <u>Synchronous mode:</u> 1 = Master mode (clock generated internally from BRG) 0 = Slave mode (clock from external source)							
bit 6	TX9: 9-bit Transmit Enable bit 1 = Selects 9-bit transmission 0 = Selects 8-bit transmission							
bit 5	TXEN: Transmit Enable bit 1 = Transmit enabled 0 = Transmit disabled Note: SREN/CREN overrides TXEN in SYNC mode.							
bit 4	SYNC: USART Mode Select bit 1 = Synchronous mode 0 = Asynchronous mode							
bit 3	Unimplemented: Read as '0'							
bit 2	BRGH: High Baud Rate Select bit <u>Asynchronous mode:</u> 1 = High speed 0 = Low speed <u>Synchronous mode:</u> Unused in this mode							
bit 1	TRMT: Transmit Shift Register Status bit 1 = TSR empty 0 = TSR full							
bit 0	TX9D: 9th bit of Transmit Data, can be parity bit							

Figura 4.35 Registro de control y transmisión TXSTA del PIC 16f877. Fuente: Hoja de especificaciones del PIC 16F877. Microchip.

4.5.5.4.6 Los módulos de ccp (captura, Comparación, PWM):

El PIC16F877 posee dos módulos CCP, denominados CCP1 y CCP2. Ambos módulos son prácticamente idénticos con la excepción de la operación del disparo de



evento especial. Cada uno de estos dos módulos poseen un registro de 16 bits, el cual puede operar como:

- Registro de captura de 16 bits.
- Registro de comparación de 16 bits.
- Registro de Ciclo de Trabajo del módulo PWM.

Cada modo de operación requiere como recurso uno de los timers del PIC.

En la tabla 4.6 se muestran los timers usados por cada modo:

Tabla 4.6 Timers usados para los diferentes modos de operación del CCP. Fuente: Propia

Modo de operación del CCP	Recurso utilizado
Captura	Timer 1
Comparación	Timer 1

Fuente: Propia.

A continuación se da un breve resumen de los registros relacionados con cada módulo:



4.5.5.4.6.1 El Módulo CCP1:

El registro principal de este módulo (CCPR1) se compone de dos registros de 8 bits, denominados CCPR1H (16h) (parte más significativa) y CCPR1L (15h) (parte menos significativa). La operación del módulo se controla mediante el registro CCP1CON y el disparo de evento especial, el cual es generado al alcanzarse la igualdad en un registro de comparación reseteará el Timer 1.

4.5.5.4.6.2 El Módulo CCP2:

El registro principal de este módulo (CCPR2) se compone de dos registros de 8 bits, denominados CCPR2H (parte más significativa) y CCPR2L (parte menos significativa).

La operación del módulo se controla mediante el registro CCP2CON y el disparo de evento especial, el cual es generado al alcanzarse la igualdad en un registro de comparación reseteará el Timer 1 e iniciará una conversión analógico/digital (si el módulo convertidor A/D está habilitado).

4.5.5.4.6.3 El modulo PWM:

El Modulo PWM del PIC 16F877 tiene una resolución máxima de 10 bit y su salida se toma del pin RC2, por lo cual debe estar configurado como salida en el registro Tris C.

El Periodo del Modulo PWM es configurado dando un valor al registro PR2. Este periodo puede ser calculado usando la siguiente fórmula:

$$\text{PWM Period} = [(PR2) + 1] \cdot 4 \cdot TOSC \cdot \text{Prescale del Timer 2}$$



Como ya es sabida la frecuencia está determinada por $1/\text{Periodo PWM}$. Cuando el valor del Timer 2 alcanza el valor de PR2, los siguientes eventos ocurren en el siguiente ciclo de tiempo:

- TMR2 es limpiado.
- El pin CCP1 es puesto a 1.
- El periodo de trabajo del PWM es cambiado de CCPR1H a CCPIR1L.

El ciclo de trabajo del PWM es determinado escribiendo al registro CCPR1L y CCP1CON<5:4> (10 bits de resolución).

El ciclo de trabajo se obtiene usando la siguiente fórmula:

$$\text{Tiempo de Trabajo PWM} = (\text{CCPR1L:CCP1CON<5:4>}) \cdot \text{TOSC} \cdot (\text{TMR2 Prescale Value})$$

Para modificar el ciclo de trabajo, en cualquier momento se puede escribir a CCPR1L y CCP1CON<5:4>, pero estos valores no serán tomados en consideración hasta que el Timer alcance al PR2 y reinicie su operación tomando en cuenta los nuevos valores.

$$\text{Tiempo de trabajo} = \% \text{ Ciclo de trabajo} / F_{\text{pwm}}$$

Para calcular la máxima resolución del PWM a determinar frecuencia usamos la siguiente fórmula:

$$\text{Resolution} = \frac{\log\left(\frac{F_{\text{OSC}}}{F_{\text{PWM}}}\right)}{\log(2)} \text{ bits}$$

CONFIGURACION DEL PWM:

Pasos para la configuración del Modulo PWM del PIC 16F877:

- Configurar el Periodo dando un valor al registro PR2.
- Configurar el Ciclo de Trabajo escribiendo en: CCPR1L:CCP1CON<5:4>.
- Limpiar el Tris C, 2 para asignar la salida del modulo (CCP1).
- Asignar el valor del Prescale del Timer 2 (T2CON).
- Configurar el modulo CCP1 para operación pwm.

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on: POR, BOR	Value on all other RESETS
0Bh, 8Bh, 10Bh, 18Bh	INTCON	GIE	PEIE	T0IE	INTE	RBIE	T0IF	INTF	RBIF	0000 000x	0000 000u
0Ch	PIR1	PSPIF ⁽¹⁾	ADIF	RCIF	TXIF	SSPIF	CCP1IF	TMR2IF	TMR1IF	0000 0000	0000 0000
0Dh	PIR2	—	—	—	—	—	—	—	CCP2IF	---- --0	---- --0
8Ch	PIE1	PSPIE ⁽¹⁾	ADIE	RCIE	TXIE	SSPIE	CCP1IE	TMR2IE	TMR1IE	0000 0000	0000 0000
8Dh	PIE2	—	—	—	—	—	—	—	CCP2IE	---- --0	---- --0
87h	TRISC	PORTC Data Direction Register								1111 1111	1111 1111
11h	TMR2	Timer2 Module's Register								0000 0000	0000 0000
92h	PR2	Timer2 Module's Period Register								1111 1111	1111 1111
12h	T2CON	—	TOUTPS3	TOUTPS2	TOUTPS1	TOUTPS0	TMR2ON	T2CKPS1	T2CKPS0	-000 0000	-000 0000
15h	CCPR1L	Capture/Compare/PWM Register1 (LSB)								xxxx xxxx	uuuu uuuu
16h	CCPR1H	Capture/Compare/PWM Register1 (MSB)								xxxx xxxx	uuuu uuuu
17h	CCP1CON	—	—	CCP1X	CCP1Y	CCP1M3	CCP1M2	CCP1M1	CCP1M0	--00 0000	--00 0000
1Bh	CCPR2L	Capture/Compare/PWM Register2 (LSB)								xxxx xxxx	uuuu uuuu
1Ch	CCPR2H	Capture/Compare/PWM Register2 (MSB)								xxxx xxxx	uuuu uuuu
1Dh	CCP2CON	—	—	CCP2X	CCP2Y	CCP2M3	CCP2M2	CCP2M1	CCP2M0	--00 0000	--00 0000

Figura 4.36 Registros asociados al modulo CCP operando en modo PWM. Fuente: Hoja de especificaciones PIC 16F877.Microchip.

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on: POR, BOR	Value on all other RESETS
0Bh, 8Bh, 10Bh, 18Bh	INTCON	GIE	PEIE	T0IE	INTE	RBIE	T0IF	INTF	RBIF	0000 000x	0000 000u
0Ch	PIR1	PSPIF ⁽¹⁾	ADIF	RCIF	TXIF	SSPIF	CCP1IF	TMR2IF	TMR1IF	0000 0000	0000 0000
0Dh	PIR2	—	—	—	—	—	—	—	CCP2IF	---- --0	---- --0
8Ch	PIE1	PSPIE ⁽¹⁾	ADIE	RCIE	TXIE	SSPIE	CCP1IE	TMR2IE	TMR1IE	0000 0000	0000 0000
8Dh	PIE2	—	—	—	—	—	—	—	CCP2IE	---- --0	---- --0
87h	TRISC	PORTC Data Direction Register								1111 1111	1111 1111
0Eh	TMR1L	Holding Register for the Least Significant Byte of the 16-bit TMR1 Register								xxxx xxxx	uuuu uuuu
0Fh	TMR1H	Holding Register for the Most Significant Byte of the 16-bit TMR1 Register								xxxx xxxx	uuuu uuuu
10h	T1CON	—	—	T1CKPS1	T1CKPS0	T1OSCEN	T1SYNC	TMR1CS	TMR1ON	--00 0000	--uu uuuu
15h	CCPR1L	Capture/Compare/PWM Register1 (LSB)								xxxx xxxx	uuuu uuuu
16h	CCPR1H	Capture/Compare/PWM Register1 (MSB)								xxxx xxxx	uuuu uuuu
17h	CCP1CON	—	—	CCP1X	CCP1Y	CCP1M3	CCP1M2	CCP1M1	CCP1M0	--00 0000	--00 0000
18h	CCPR2L	Capture/Compare/PWM Register2 (LSB)								xxxx xxxx	uuuu uuuu
1Ch	CCPR2H	Capture/Compare/PWM Register2 (MSB)								xxxx xxxx	uuuu uuuu
1Dh	CCP2CON	—	—	CCP2X	CCP2Y	CCP2M3	CCP2M2	CCP2M1	CCP2M0	--00 0000	--00 0000

Figura 4.37 Registros asociados al modulo CCP operando en modo captura y comparación.

Fuente: Hoja de especificaciones PIC 16f877. Microchip.

PWM Frequency	1.22 kHz	4.88 kHz	19.53 kHz	78.12kHz	156.3 kHz	208.3 kHz
Timer Prescaler (1, 4, 16)	16	4	1	1	1	1
PR2 Value	0xFFh	0xFFh	0xFFh	0x3Fh	0x1Fh	0x17h
Maximum Resolution (bits)	10	10	10	8	7	5.5

Figura 4.38 Ejemplo de frecuencia y resolución de PWM a 20MHz. Fuente Hoja de especificaciones PIC 16F877. Microchip.



4.6 Costo de elaboración del Sistema de Desarrollo:

La tabla 4.7 presenta la relación de costo para la elaboración del Sistema de desarrollo propuesto en este proyecto.

Tabla 4.7 Relación de costos.

Materiales	Cantidad	Precio	Total
Transistores 2n3904.	3	1.00	3.00
Transistores 2n3906.	1	1.00	1.00
Display 7 Seg.	2	15.00	30.00
Inductor de 680uH.	1	1.20	1.20
Lcd 16x2.	1	54.00	54.00
Capacitores cerámicos 100nF.	2	0.52	1.04
Capacitores electrolítico 47uF.	2	0.64	1.28
Capacitores electrolítico 10uF.	1	0.64	0.64
Pulsadores.	6	3.00	18.00
Interruptores.	1	7.45	7.45
Resonador de 20 MHz.	2	10.00	20.00
LEDs Varios.	10	2.00	20.00
Microcontrolador PIC 16F877A.	1	80.96	80.96
Microcontrolador PIC 18F2550.	1	79.80	79.80
Potenciómetro de 10K.	1	5.18	5.18
Conector USB tipo B.	1	7.37	7.37
Resistencias varias.	37	0.35	12.95
Relé de doble contacto a 3.3V.	1	15	15
Elaboración del circuito impreso 15 x 15 cm.	1	150	150
Elaboración del circuito impreso 7 x 5 cm.	1	28	28
Diodo 1n4148.	1	0.5	0.5
Diodo Zener 5005A 3.3V.	1	0.5	0.5
Base torneada para PIC16F877A.	1	12.88	12.88
Base torneada para PIC18F2550.	1	8.20	8.20
Interruptor de 3 pines.	2	5.20	10.40

Total Bs.

569.35



CAPITULO V: CONCLUSIONES

El presente Trabajo de Grado se elaboró tomando en cuenta como objetivo principal el desarrollar una plataforma didáctica para el laboratorio de la asignatura Microprocesadores, seleccionando el contenido teórico, de acuerdo al programa actual de la asignatura. De esta manera se evalúan las herramientas necesarias para la elaboración de las prácticas que van a ser desarrolladas por el estudiante.

Como resultado de este estudio, se logró un compendio de 4 prácticas realizadas a lo largo del curso que comprenden el contenido programático de la materia Microprocesadores. A partir de las mismas se elaboró un diseño tal que, se adapta a las necesidades requeridas por el estudiante, manejando cierta cantidad de elementos de hardware que son necesarios para la realización de las prácticas.

El manejo desde un punto de vista didáctico, tanto para el profesor como para el estudiante es de fácil uso, ya que permite el fácil aprendizaje debido a la documentación que se incluyó en el presente trabajo. Esta plataforma se elaboró de tal manera que la persona que lo use se adapte rápidamente con solo ver el manual ya que en este se muestran las especificaciones de las misma.

El desarrollo de este Proyecto es de gran importancia, ya que viene enfocado a dejar al estudiante solo a la tarea de programación y no al montaje de prácticas que es donde el estudiante puede pasar la mayor parte del tiempo debido a que esta trae muchas consecuencias ya que no siempre funciona a parte de poder dañar los dispositivos usados.

Este dispositivo es de fácil elaboración y de bajo costo con respecto a productos similares que se pueden conseguir en el mercado y que son utilizados en materias de



CAPITULO V



semestres superiores al de Microprocesadores, las cuales se buscan utilizar debido a que acortan el tiempo en el que un estudiante pueda realizar una tarea.

En caso de que se quiera llevar la propuesta presentada e incluir dicho dispositivo a la asignatura Microprocesadores, se ayudaría al estudiante, ya que este se dedicaría solo a la programación y desarrollo de aplicaciones en el laboratorio, que pueden traer ventajas al momento de aplicarlas.



RECOMENDACIONES

Dependiendo de la facilidad del estudiante de adaptarse ante la presente propuesta se presenta una serie de recomendaciones que pueden ser útiles:

- Se puede utilizar otro tipo de lenguaje de programación como lo puede ser el PROTON u otro lenguaje de programación de alto nivel que pueda ayudar al estudiante a elaborar practicas con mayor facilidad y así de esta manera también enseñar al estudiante a una programación que puede tener mayores beneficios.
- Se puede agregar a la Plataforma fuente de alimentación externa que pueda facilitar el uso de otros dispositivos de mayor consume para las aplicaciones externas.
- Se pueden agregar dispositivos que pueden aumentar el grado de complejidad de las prácticas pero así de esta manera también enseñar al estudiante a utilizarlo ya que en semestres superiores son útiles.
- Realización de nuevas prácticas como por ejemplo control de motores, uso de sensores, entre otros.



BIBLIOGRAFIAS

- **Carlos A. Reyes.** “Microcontroladores: Programación en Basic”. 2da Edición. Rispergraf. 2006.
- **Ignacio Angulo Martinez.** “Microcontroladores PIC. Diseño práctico de aplicaciones”. 2da Edición. Mc Graw Hill. 1999.
- **Angulo Jose M.** “Diseño práctico con microcontroladores: Para todos”. 1ra Edición. Thomson. 2003.
- **Angulo Jose M.** “Microprocesadores PIC, diseño práctico de aplicaciones: Primera parte”. 3ra Edición. Mc Graw Hill. 2003.
- **Angulo Jose M.** “Microprocesadores PIC, diseño práctico de aplicaciones: Segunda parte”. 3ra Edición. Mc Graw Hill. 2003.
- Hoja de Especificaciones técnicas del Microcontrolador PIC 16F877A. Microchip Technology inc.
- Hoja de Especificaciones técnicas del Microcontrolador PIC 18F2550. Microchip Technology inc.
- Manual de usuarios del Pickit 2.