



**UNIVERSIDAD CATÓLICA
DE SANTIAGO DE GUAYAQUIL**

FACULTAD DE EDUCACIÓN TÉCNICA PARA EL DESARROLLO
CARRERA DE INGENIERÍA EN TELECOMUNICACIONES

TÍTULO:

**EVALUACIÓN DEL SISTEMA DE ENTRENAMIENTO PIC18F4550
MEDIANTE APLICACIONES PRÁCTICAS PARA LA ASIGNATURA DE
MICROCONTROLADORES**

AUTORES:

David Miguel Albán Quezada

Gabriel Alejandro Peña Díaz

Previa la obtención del Título

INGENIERO EN TELECOMUNICACIONES

TUTOR:

MSc. Luzmila Ruilova Aguirre

Guayaquil, Ecuador

2015



**UNIVERSIDAD CATÓLICA
DE SANTIAGO DE GUAYAQUIL**

FACULTAD DE EDUCACIÓN TÉCNICA PARA EL DESARROLLO
CARRERA DE INGENIERÍA EN TELECOMUNICACIONES

CERTIFICACIÓN

Certificamos que el presente trabajo fue realizado en su totalidad por los
Sres. **David Miguel Albán Quezada** y **Gabriel Alejandro Peña Díaz** como
requerimiento parcial para la obtención del título de INGENIERO EN
TELECOMUNICACIONES.

TUTOR

MSc. Luzmila Ruilova Aguirre

DIRECTOR DE CARRERA

MSc. Miguel A. Heras Sánchez.

Guayaquil, a los 15 del mes de Febrero del año 2015



**UNIVERSIDAD CATÓLICA
DE SANTIAGO DE GUAYAQUIL**

FACULTAD DE EDUCACIÓN TÉCNICA PARA EL DESARROLLO
CARRERA DE INGENIERÍA EN TELECOMUNICACIONES

DECLARACIÓN DE RESPONSABILIDAD

Nosotros, **David Miguel Albán Quezada** y **Gabriel Alejandro Peña Díaz**

DECLARAMOS QUE:

El trabajo de titulación “EVALUACIÓN DEL SISTEMA DE ENTRENAMIENTO PIC18F4550 MEDIANTE APLICACIONES PRÁCTICAS PARA LA ASIGNATURA DE MICROCONTROLADORES” previa a la obtención del Título de Ingeniero en Telecomunicaciones, ha sido desarrollado respetando derechos intelectuales de terceros conforme las citas que constan al pie de las páginas correspondientes, cuyas fuentes se incorporan en la bibliografía. Consecuentemente este trabajo es de nuestra autoría.

En virtud de esta declaración, me responsabilizo del contenido, veracidad y alcance científico del Trabajo de Titulación referido.

Guayaquil, a los 15 del mes de Febrero del año 2015

LOS AUTORES

DAVID MIGUEL ALBÁN QUEZADA

GABRIEL ALEJANDRO PEÑA DÍAZ



**UNIVERSIDAD CATÓLICA
DE SANTIAGO DE GUAYAQUIL**

FACULTAD DE EDUCACIÓN TÉCNICA PARA EL DESARROLLO
CARRERA DE INGENIERÍA EN TELECOMUNICACIONES

AUTORIZACIÓN

Nosotros, **David Miguel Albán Quezada** y **Gabriel Alejandro Peña Díaz**

Autorizamos a la Universidad Católica de Santiago de Guayaquil, la publicación, en la biblioteca de la institución del Trabajo de Titulación: “EVALUACIÓN DEL SISTEMA DE ENTRENAMIENTO PIC18F4550 MEDIANTE APLICACIONES PRÁCTICAS PARA LA ASIGNATURA DE MICROCONTROLADORES”, cuyo contenido, ideas y criterios es de nuestra exclusiva responsabilidad y autoría.

Guayaquil, a los 15 del mes de Febrero del año 2015

LOS AUTORES

DAVID MIGUEL ALBÁN QUEZADA

GABRIEL ALEJANDRO PEÑA DÍAZ

DEDICATORIA

Quisiera dedicar mi proyecto de titulación primeramente a nuestro Padre Celestial por haberme regalado la vida y permitirme estar aquí en la última etapa de pregrado la cual es solo el inicio de un largo porvenir, quiero continuar dedicando este trabajo de titulación con todo mi amor y cariño a la persona que me enseñó a vivir, la que forjó mis valores, principios, moral cristiana, la que dedicó toda su vida a la mía por cumplir mis sueños, necesidades y anhelos, la que siempre estuvo a mi lado en los momentos más difíciles, en los triunfos y en las enfermedades, siempre velando por mi bienestar, el día de hoy cumplo una meta más que fue prometida hace ya casi ocho años y sé que si estuvieras físicamente hoy aquí conmigo serías la persona más feliz del mundo, esto es para ti mamá.

Agradezco a mi papá el Sr. Gabriel Peña Cedeño por su esfuerzo, por la comprensión que me dio como su hijo mayor, por su atención, por su paciencia y por haberme enseñado valores tan importantes como la puntualidad, la firmeza y la disciplina quienes me acompañan siempre en cada paso que doy.

Quiero continuar agradeciendo a mi abuelita la cual tengo la dicha de tenerla con vida a sus 90 años, la madre de mi mamá, la Sra. Mercedes Romero Ochoa quien con sus anécdotas y enseñanzas, siempre quedaran en mí las lecciones de vida, quiero agradecer a mis tíos, el Sr. Jorge Díaz Romero y la Sra. Grace Ortega González los cuales se encargaron a lo largo de mi formación que nunca me falte un bocado de comida a quienes siempre recurrí cuando necesité un consejo de padre o de madre, quiero decirles hoy que me considero un hijo más de su familia porque tal sacrificio como el suyo nadie lo da y esto hace que esté eternamente agradecido.

Por último pero no menos importante agradecer a dos personas que estuvieron a mi lado a pesar de la distancia que nos separa, quienes con sus enseñanzas forjaron en mí lo que soy y se preocuparon cada día porque no me falte nada, ellos saben los momentos más duros que me toco pasar y a

quienes considero pilares importantes en mi formación desde niño, con todo mi cariño al Sr. Julio Díaz Romero y a mi maestra Janine Castro Cárdenas.

GABRIEL ALEJANDRO PEÑA DÍAZ

Quiero agradecer primeramente a Dios por un día más de vida y a mis padres quienes me dieron vida, educación, apoyo y consejos ellos son el Sr. Arturo Albán Campos y mi mamá Liliberg Quezada Arguello.

DAVID MIGUEL ALBÁN QUEZADA

AGRADECIMIENTO

Queremos agradecer especialmente al Ing. Manuel Romero Paz por sus conocimientos brindados a lo largo de la carrera, por su ayuda, su tiempo y por sus críticas constructivas que forjaron en nosotros la excelencia profesional, así mismo agradecemos a la Ing. Luzmila Ruilova nuestra Tutora por sus sabios consejos, al Ing. Fernando Palacios Meléndez, al Ing. Carlos Zambrano Montes, al Ing. Armando Heras Sánchez y al Ing. Carlos Romero Rosero.

EL AUTOR

DAVID MIGUEL ALBÁN QUEZADA
GABRIEL ALEJANDRO PEÑA DÍAZ

Índice General

Índice de Figuras	X
Índice de Tablas.....	XII
Resumen	XIII
CAPÍTULO 1: GENERALIDADES DEL TRABAJO DE TITULACIÓN	14
1.1. Antecedentes.	14
1.2. Justificación del Problema.....	15
1.3. Definición del Problema.....	16
1.4. Objetivos del Problema de Investigación.....	16
1.4.1. Objetivo General.....	16
1.4.2. Objetivos Específicos.	17
1.5. Hipótesis.....	17
1.6. Metodología de Investigación.....	17
CAPÍTULO 2: Estado del Arte de los Sistemas Embebidos PICs.....	19
2.1. Introducción de sistemas microprocesados y microcontroladores.	19
2.2. Sistemas Microcontroladores.	19
2.2.1. Memorias RAM.	25
2.2.2. Memorias ROM.....	27
2.2.3. Memorias PROM.	28
2.2.4. Memorias EPROM.....	28
2.2.5. Memorias EEPROM.	29
2.2.6. Memorias Flash EEPROM.....	30
2.3. Características de los microcontroladores PIC.....	30
2.3.1. Voltaje de alimentación.....	30
2.3.2. Señal de Reloj.	32

2.3.3.	Temporizadores Timers.....	33
2.3.3.1.	Perro guardián (watchdog).	35
2.3.4.	Convertidores Analógico/Digital.....	35
2.4.	Arquitecturas de los microcontroladores.	36
2.4.1.	RISC.....	38
2.4.2.	CISC.....	38
2.5.	Otros Sistemas de Entrenamiento de Microcontroladores.	38
2.5.1.	Entrenador de Microcontroladores PICAXE	38
2.5.2.	Módulo de entrenamiento PIC EDUTRONICA JJTSA.....	44
CAPÍTULO 3: DESARROLLO DE APLICACIONES PRÁCTICAS DEL		
	PIC18F4550.	48
3.1.	Aplicación práctica 1: Secuencia de encendido de LEDs.....	48
3.2.	Aplicación práctica 2: Seguridad con Clave.	50
3.3.	Aplicación práctica 2: Alarma con temporizador y varios periféricos.....	55
3.4.	Comunicación serial con el PIC18F4550.....	61
CAPÍTULO 4: CONCLUSIONES Y RECOMENDACIONES.....		
4.1.	Conclusiones.....	65
4.2.	Recomendaciones.....	66
REFERENCIAS BIBLIOGRÁFICAS.....		
		68

Índice de Figuras

Capítulo 2

Figura 2. 1: Diagrama de bloques de un sistema controlador de temperatura de un horno basados en un microcontrolador.....	22
Figura 2. 2: Diagrama del sistema controlador de temperatura con teclado y LCD.....	23
Figura 2. 3: Un controlador de temperatura más sofisticado	24
Figura 2. 4: Arquitectura de un microcontrolador PIC.....	25
Figura 2. 5: Diagrama de bloques de memoria RAM en un PIC.....	26
Figura 2. 6: Mapa de la memoria RAM.....	27
Figura 2. 7: Proceso de grabar el programa a una memoria ROM.....	28
Figura 2. 8: Alimentación mediante regulador de voltaje LM7805.....	31
Figura 2. 9: Conexión del cristal de cuarzo al PIC 16F887.....	32
Figura 2. 10: Resonador cerámico.....	33
Figura 2. 11: Diagrama de bloques de temporización.....	34
Figura 2. 12: Conexión de watchdog con osciladores R.....	35
Figura 2. 13: Conexión de watchdog con osciladores R.....	36
Figura 2. 14: Arquitectura Von Neumann.....	37
Figura 2. 15: Arquitectura Harvard.....	37
Figura 2. 16: Diagrama por bloques del entrenador PICAXE.....	39
Figura 2. 16: Módulo de entrenamiento PICAXE.....	40
Figura 2. 18: Circuito esquemático de la conexión de los 2 displays.....	41
Figura 2. 19: Circuito esquemático de la conexión de 4 LEDs.....	41
Figura 2. 20: Circuito esquemático de la salida con relé.....	42
Figura 2. 21: Circuito esquemático de pulsadores.....	42

Figura 2. 22: Circuito esquemático del Dip-switch. 43

Figura 2. 23: Circuito esquemático del Teclado Matricial..... 43

Figura 2. 24: Circuito esquemático del Buzzer. 44

Figura 2. 25: Circuito esquemático de los dispositivos de salida. 44

Figura 2. 26: Tarjeta de la etapa de programación de PIC. 45

Figura 2. 27: Tarjeta de la etapa de evaluación de programas. 46

Capítulo 3

Figura 3. 1: Encendido de los primero cuatro LEDs del puerto D 49

Figura 3. 2: Encendido de los últimos cuatro LEDs del puerto D..... 50

Figura 3. 3: Conexión del teclado y LCD en los puertos B y D del PIC
18F4550..... 54

Figura 3. 4: Primer caso, cuando la clave digitada es correcta en
comparación al guardado en el código de programación. 54

Figura 3. 5: Segundo caso, cuando la clave digitada no coincide con los
dígitos que están en el código de programación..... 55

Figura 3. 6: Diseño de la simulación del circuito con temporizador. 60

Figura 3. 7: LCD muestra al usuario el tiempo no alarmar el sistema..... 60

Figura 3. 8: LCD recibe señal del sensor que detecta un cambio de estado.
..... 61

Figura 3. 9: Diseño de la simulación del circuito mediante comunicación
serial. 64

Figura 3. 10: Resultado obtenido mediante comunicación serial..... 64

Índice de Tablas**Capítulo 2**

Tabla 2. 1: Modos de osciladores de cuarzo.	32
Tabla 2. 2: Modos de un resonador cerámico.....	33

Resumen

El trabajo de titulación desarrollado permitió evaluar el sistema o módulo de entrenamiento PIC18F4550 mediante cuatro aplicaciones prácticas que demostraron la utilidad de este módulo diseñado por la empresa ecuatoriana All Power Microcontroller (APM). La validación del módulo PIC18F4550 fue un éxito mediante el diseño de aplicaciones reales simuladas en ISIS PROTEUS y de la programación en Compiler C CSS.

CAPÍTULO 1: GENERALIDADES DEL TRABAJO DE TITULACIÓN

1.1. Antecedentes.

A través de las reformas realizadas en la Ley Orgánica de Educación Superior (LOES) las Universidades Públicas y Privadas del Ecuador, especialmente las que ofertan Carreras de Ingeniería en Telecomunicaciones, Electrónica y afines, están reestructurando y equipando sus laboratorios con la finalidad de que sus carreras sean Acreditadas.

En la actualidad el Laboratorio de Electrónica de la Facultad de Educación Técnica para el Desarrollo se dictan clases de Microcontroladores y Diseño Electrónico Digital en la Carrera de Ingeniería en Telecomunicaciones. El laboratorio tiene disponible 18 computadoras de escritorio, en ellas se tiene instalado PROTEUS versión de evaluación en donde se desarrollan las simulaciones de microcontroladores PIC.

También está disponible el sistema de entrenamiento EasyPic5 con 9 tarjetas, éstas son utilizadas en clases aunque solo es compatible con los lenguajes de programación de alto nivel MikroBasic y MikroC pero estos no disponen de algunas librerías que son necesarias para determinados códigos de programación. El software *Compilador C CSS* se utiliza para el presente trabajo de titulación porque dispone de todas las librerías y su

presentación es amigable para los estudiantes de 6to Ciclo y para principiantes que deseen aprender a programar.

Y finalmente, se tienen unos módulos de entrenamiento desarrollados por Ideas y Tecnología de profesionales ecuatorianos, estos módulos también son utilizados en clases, pero manejan solamente el PIC16F887.

1.2. Justificación del Problema.

De acuerdo a lo explicado en los antecedentes del presente trabajo de titulación, es necesaria la búsqueda de otros sistemas de entrenamiento de microcontroladores como es el caso del PIC18F4550. Este microcontrolador en la actualidad son empleados en la mayoría de Instituciones de Educación Superior.

El PIC18F4550 permite integrar más dispositivos de entrada/salida, serían de gran ayuda para desarrollar y proponer nuevos trabajos de titulación, así como proyectos de investigación experimental y porque no en la formación de Maestros en Telecomunicaciones.

Los microcontroladores pueden ser aplicados al procesamiento de señales biomédica, por ejemplo el Instituto Politécnico Nacional de México se desarrolló un Sistema de Adquisición y Registro de Señales Electrocardiográficas a través del PIC18F4550. En la Revista Colombiana de Tecnologías de Avanzada se presentó el artículo "Sistema de

Posicionamiento Solar utilizando la Plataforma de Labview y el Microcontrolador PIC18F4550.

En España hay muchas tesis de grado, especialmente de Maestría y Doctorado que utilizan el microcontrolador PIC18F4550 para sus investigaciones. Por ejemplo en la Universidad Politécnica de Cataluña se encontró una tesis de maestría denominada Sistema de adquisición y registro de los parámetros de interés en la conducción de vehículo, en el cual emplearon el PIC18F4550.

1.3. Definición del Problema.

Necesidad de evaluar nuevos sistemas de entrenamiento de microcontroladores como el PIC18F4550 mediante el desarrollo de aplicaciones prácticas que sirvan como herramienta didáctica de la asignatura de Microcontroladores que se dicta en la Carrera de Ingeniería en Telecomunicaciones y también en la Carrera de Ingeniería Electrónica en Control y Automatismo

1.4. Objetivos del Problema de Investigación.

1.4.1. Objetivo General.

Evaluar el sistema de entrenamiento PIC18F4550 mediante aplicaciones prácticas para la asignatura de Microcontroladores.

1.4.2. Objetivos Específicos.

- Describir el estado del arte o fundamentos teóricos de los Sistemas Embebidos PICs.
- Diseñar aplicaciones prácticas a través de simulaciones en la plataforma ISIS PROTEUS.
- Realizar la programación en lenguaje de alto nivel C a través del Compilador C CSS.
- Mostrar los resultados obtenidos a través de las simulaciones diseñadas en PROTEUS y en la tarjeta de entrenamiento PIC18F4550.

1.5. Hipótesis.

La evaluación del sistema de entrenamiento PIC18F4550 mediante aplicaciones prácticas para la asignatura de Microcontroladores permitirá mejorar el proceso de aprendizaje de los estudiantes de sexto ciclo de Ingeniería en Telecomunicaciones y serán de gran ayuda para desarrollar robots para participaciones en concursos de robótica que se desarrollan a nivel nacional y del CER 2015 ha desarrollarse en la Universidad Católica de Santiago de Guayaquil.

1.6. Metodología de Investigación.

La metodología de investigación utilizada para el trabajo de titulación, es de carácter Exploratorio y Explicativo con paradigma Empírico-Analítico cuyo enfoque es cuantitativo. El diseño de investigación es experimental,

porque se desarrollaron prácticas experimentales en laboratorio, tanto a nivel de simulación como de implementación sobre la tarjeta de entrenamiento PIC18F4550.

CAPÍTULO 2: Estado del Arte de los Sistemas Embebidos PICs

2.1. Introducción de sistemas microprocesados y microcontroladores.

El término microprocesador se utiliza para describir un sistema que incluye al mínimo un microprocesador, memoria de programa, memoria de datos, y dispositivos de entrada/salida (*I/O*). Algunos sistemas microprocesados incluyen componentes adicionales tales como temporizadores, contadores y convertidores analógico/digital. Por lo tanto, un sistema microprocesador puede ser una computadora grande que tiene varios discos duros, lectores de cd's e impresoras integrados a un controlador conocido como chip.

En esta sección consideramos solamente el tipo de microprocesador que consta de un solo chip de silicio. Existen otros sistemas microinformáticos denominados microcontroladores, y son utilizados en muchos artículos para el hogar tales como hornos microondas, unidades de control remoto de TV, cocinas, equipos de alta fidelidad, reproductor de CD, ordenadores personales y frigoríficos. Muchos microcontroladores de diferentes fabricantes están disponibles en el mercado.

2.2. Sistemas Microcontroladores.

Un microcontrolador es un circuito controlador reprogramable según las necesidades para las que se desea desarrollar un proyecto. Micro indica que el dispositivo es pequeño, y controlador porque se utiliza en aplicaciones de

control. Otro término para el microcontrolador es un controlador incrustado, debido a que la mayoría de los microcontroladores están incorporados en los dispositivos que lo controlan.

Un microprocesador difiere de un microcontrolador en un número de maneras. La distinción principal es que un microprocesador requiere varios otros componentes para su funcionamiento, tales como memoria de programa y memoria de datos, dispositivos de entrada-salida, y un circuito de reloj externo. Un microcontrolador, por otro lado, tiene todas las fichas de apoyo incorporados dentro de un solo chip.

Todos los microcontroladores operan en un conjunto de instrucciones (programa de usuario) almacenado en su memoria. Un microcontrolador obtiene las instrucciones de su memoria un programa por uno, decodificando estas instrucciones, y luego llevando a cabo operaciones necesarias.

Los microcontroladores tradicionalmente se han programado utilizando el lenguaje ensamblador o de bajo nivel. Aunque el lenguaje ensamblador es rápido, tiene varias desventajas. Un programa en ensamblado consiste en un conjunto de mnemotecnia, lo que hace que el aprendizaje y el mantenimiento de un programa escrito en lenguaje ensamblador sea difícil. También, los microcontroladores fabricados por diferentes empresas tienen diferentes lenguajes ensambladores, por lo que el usuario debe aprender un nuevo lenguaje para cada microcontrolador.

Los microcontroladores, también se pueden programar utilizando un lenguaje de alto nivel, tales como BASIC, PASCAL, o C. Los lenguajes de alto nivel son mucho más fáciles de aprender que el de lenguaje ensamblador. También facilitan el desarrollo de programas grandes y complejos.

El presente trabajo de titulación consiste en desarrollar aplicaciones prácticas para la asignatura de Microcontroladores mediante el PIC18F4550 y programados en lenguaje C conocido como MCSC.

En teoría, un solo chip es suficiente para tener un sistema microcontrolador en funcionamiento. En aplicaciones prácticas, sin embargo, los componentes adicionales pueden ser requeridos por lo que el microcontrolador puede interactuar con su entorno. Con la llegada de la familia de microcontroladores PIC el tiempo de desarrollo de un proyecto electrónico se ha reducido a pocas horas o minutos.

Básicamente, un microcontrolador ejecuta un programa de usuario que se carga en la memoria de programa. Bajo el control de este programa, se reciben datos desde dispositivos externos (entradas), y luego enviado a los dispositivos externos (salidas). Por ejemplo, en un sistema de control de temperatura de un horno se basa en un microcontrolador que permite la adquisición de datos de temperatura mediante un sensor de temperatura y

posterior se activa un ventilador para mantener la temperatura en el valor requerido.

La figura 2.1 muestra un diagrama de bloques de un sistema de control de temperatura de un horno simple. El sistema mostrado en la figura 2.1 es muy simple. Un sistema más sofisticado puede incluir un teclado para ajustar la temperatura y una pantalla LCD para que aparezca. La figura 2.2 muestra un diagrama de bloques de este sistema de control de temperatura más sofisticado.

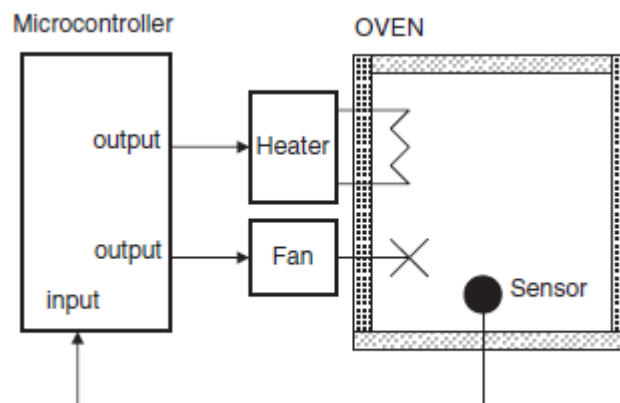


Figura 2. 1: Diagrama de bloques de un sistema controlador de temperatura de un horno basados en un microcontrolador.

Fuente: Barra Z., O., & Barra Z., F. (2011).

Podemos hacer el diseño aún más sofisticado (véase la figura 2.2) mediante la adición de una alarma que se activa si la temperatura está fuera del rango deseado. Además, las lecturas de temperatura pueden ser enviadas a un PC cada segundo para ser archivado y posterior procesamiento (véase la figura 2.3). Por ejemplo, un gráfico de la temperatura diaria se puede obtener en la PC.

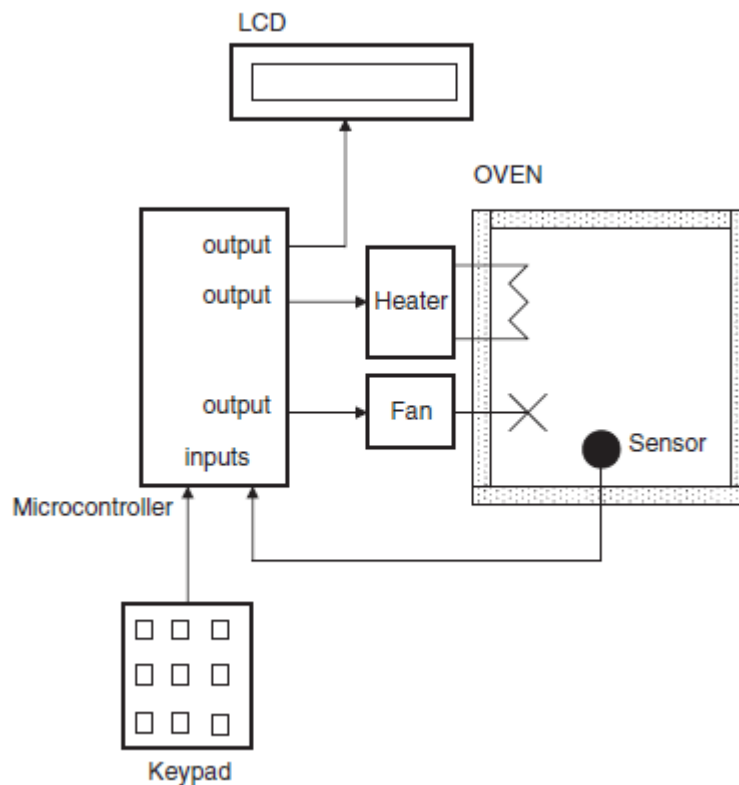


Figura 2. 2: Diagrama del sistema controlador de temperatura con teclado y LCD.
Fuente: Barra Z., O., & Barra Z., F. (2011).

Un microcontrolador es una herramienta muy potente que permite a los diseñadores crear la manipulación sofisticada de datos de los dispositivos de I/O controlados desde el programa. Los microcontroladores son clasificados por el número de bits que procesan.

Los microcontroladores de 8 bits son los más populares y se utilizan en la mayoría de las aplicaciones basadas en microcontroladores. Mientras que los microcontroladores de 16 y 32 bits son mucho más poderosos, especialmente los PIC 18F4550 pero generalmente son más caros y dependen del tipo de aplicación, si es pequeño y mediano (proyectos sencillos) no son necesarias utilizar microcontroladores de 16 y 32 bits.

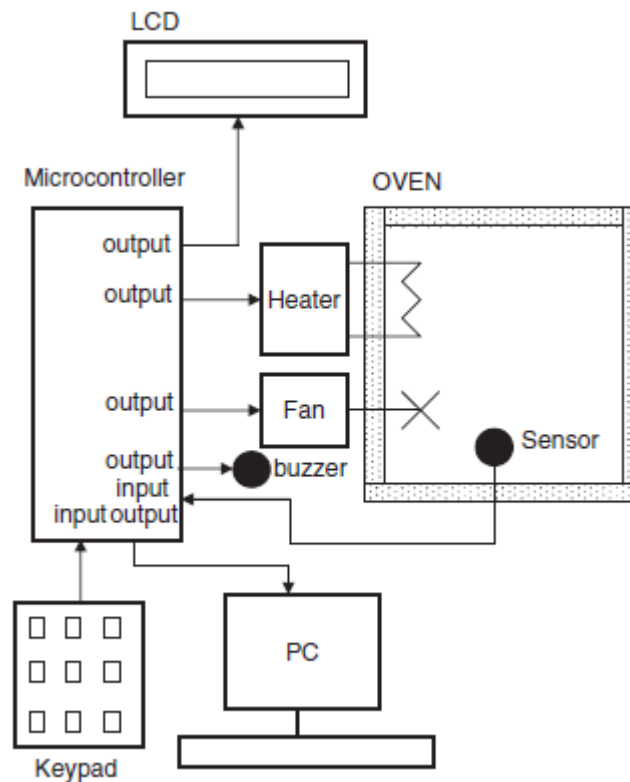


Figura 2. 3: Un controlador de temperatura más sofisticado
Fuente: Barra Z., O., & Barra Z., F. (2011).

La figura 2.4 muestra la arquitectura de los microcontroladores PIC, en la que consta de un microprocesador, memoria RAM, memoria ROM (EEPROM), etc. La memoria, es una parte importante de un sistema de microcontrolador, se puede clasificar en dos tipos: memoria de programa y memoria de datos. La memoria de programa almacena el programa escrito por el programador y por lo general es no volátil (es decir, los datos no se pierden después de que la alimentación se desconecta).

La memoria de datos almacena los datos temporalmente utilizados en un programa y es generalmente volátil (es decir, los datos se pierden

después que la alimentación se desconecta). Básicamente, existen seis tipos de memorias, que se resumen de la siguiente manera:

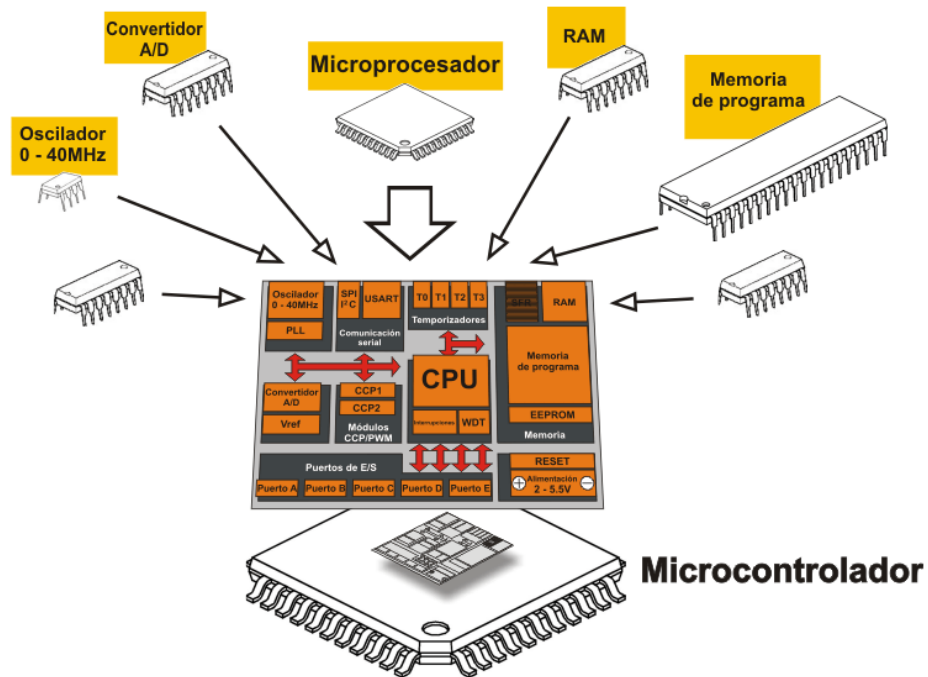


Figura 2. 4: Arquitectura de un microcontrolador PIC.

Fuente: Mikroelektronika

2.2.1. Memorias RAM.

La memoria RAM es conocida como memoria de acceso aleatorio, es decir, que es una memoria de propósito general que almacena los datos de usuario en un programa, tal como se muestra en la figura 2.5. La memoria RAM es volátil en el sentido de que no puede retener los datos en ausencia de alimentación (es decir, los datos se pierden después que la alimentación se desconecta).

La mayoría de los microcontroladores tienen una cierta cantidad de memoria RAM interna, 256 bytes siendo una cantidad común, aunque

algunos microcontroladores tienen más, otros menos. El microcontrolador PIC18F452, por ejemplo, tiene una memoria RAM de 1536 bytes.

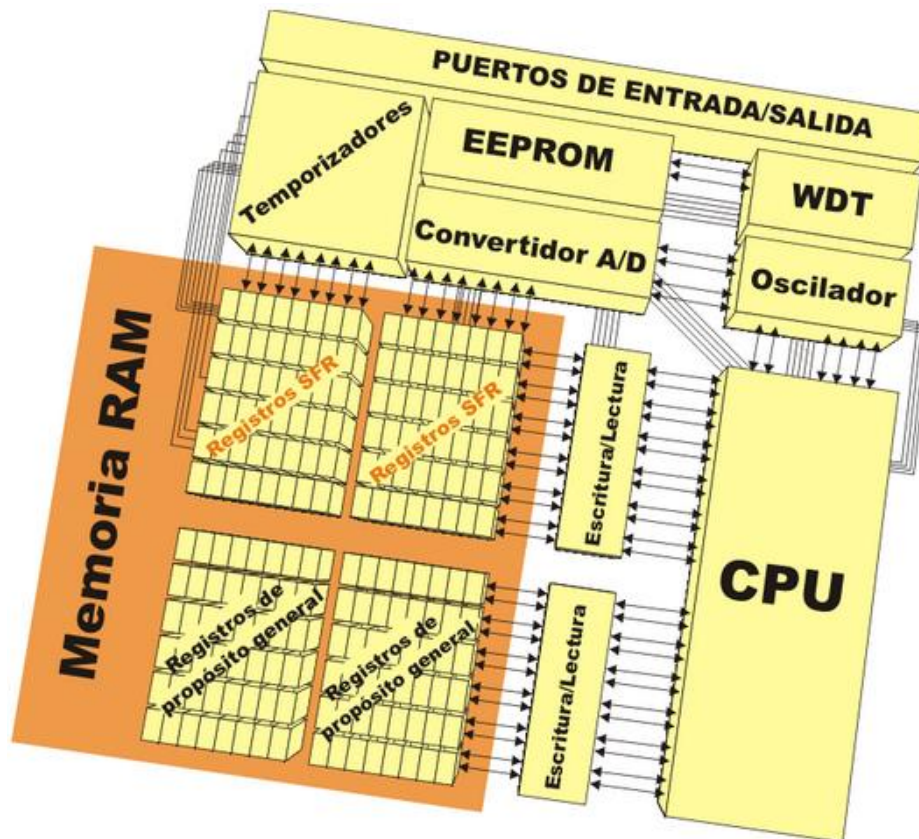


Figura 2. 5: Diagrama de bloques de memoria RAM en un PIC.
Fuente: Mikroelektronika

En general una memoria RAM se puede ampliar mediante la adición de chips de memorias externas. La figura 2.6 muestra el mapa de la memoria RAM de datos para un microcontrolador específico, como por ejemplo el PIC 18F4550.

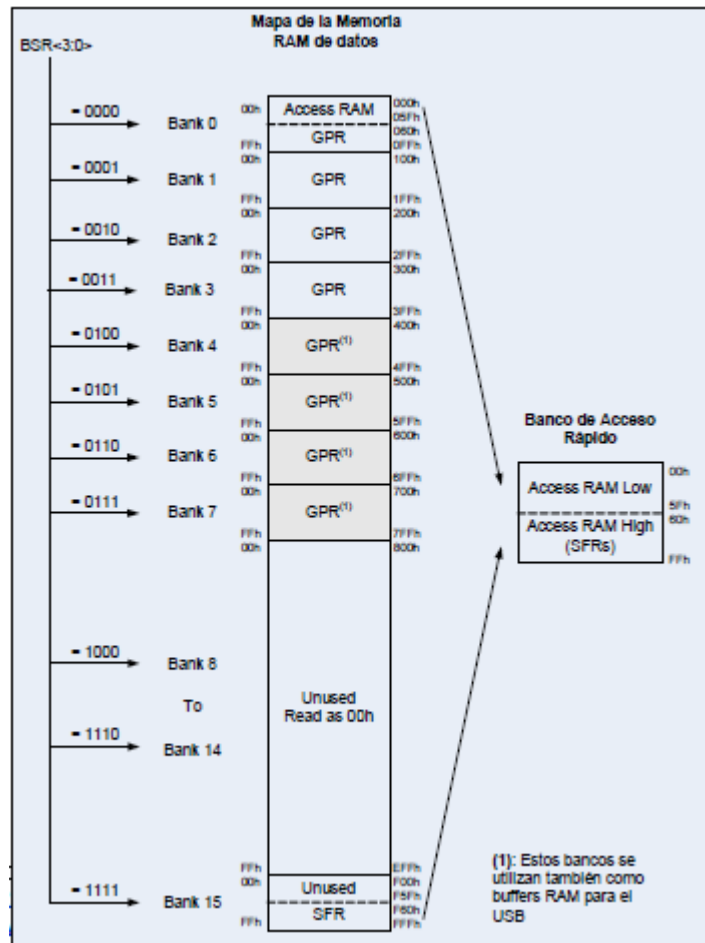


Figura 2. 6: Mapa de la memoria RAM.

Fuente: Pérez M., Y., (2012).

2.2.2. Memorias ROM.

La memoria ROM es conocida como memoria de sólo lectura, por lo general tiene un programa o datos de usuario fijo. La memoria ROM es no volátil. Si se desconecta y se conecta la alimentación de la ROM, los datos originales seguirán ahí. La memoria ROM está programada durante el proceso de fabricación (véase la figura 2.7), y el usuario no puede cambiar su contenido. La memoria ROM sólo es útil si desarrollamos un programa y desea crear varios miles de copias de la misma. Las memorias ROM se clasifican en:



Figura 2. 7: Proceso de grabar el programa a una memoria ROM.
Fuente: Mikroelektronika

2.2.3. Memorias PROM.

La memoria PROM, es una memoria de sólo lectura programable, es un tipo de memoria ROM que puede ser programado a menudo por el usuario final. Una vez que una PROM se ha programado, su contenido no se puede cambiar. Las PROM se utilizan generalmente en aplicaciones de baja producción que requieren de pocos datos de esos recuerdos.

2.2.4. Memorias EPROM.

La memoria EPROM, es una memoria de sólo lectura programable y borrrable, es similar a la ROM, pero EPROM se puede programar utilizando un dispositivo de programación adecuado. Una memoria EPROM tiene una pequeña ventana de cristal transparente en la parte superior del chip donde los datos pueden ser borrados bajo una fuerte luz ultravioleta.

Una vez que la memoria se programa, la ventana puede ser cubierta con cinta oscura para evitar el borrado accidental de los datos. Una memoria EPROM debe ser borrada antes de que pueda ser reprogramado. Muchas versiones de desarrollo de microcontroladores se fabrican con las memorias EPROM en el programa de usuario se almacenan.

Estos registros se borran y se reprograman hasta que el usuario este satisfecho con el programa. Algunas versiones de EPROM, conocidas como OTP (programable una vez), se pueden programar utilizando un dispositivo programador adecuado, pero no se pueden borrar. Las memorias OTP cuestan mucho menos que las EPROM. Las memorias OTP son útiles después que un proyecto se ha desarrollado por completo, es decir, que se deben adicionar memorias.

2.2.5. Memorias EEPROM.

Las memorias EEPROM, son memorias de sólo lecturas eléctricamente programables y borrables, es una memoria no volátil que puede ser borrada y reprogramada utilizando un dispositivo de programación adecuado. Las EEPROMs son utilizadas para guardar información de configuración.

Algunos microcontroladores de la gama alta tiene incorporado memorias EEPROM. Por ejemplo, el PIC 18F452 contiene una memoria EEPROM de 256 bytes donde cada byte se puede programar y borrar directamente por el software de aplicaciones. Las memorias EEPROM son

generalmente lentas. Una memoria EEPROM es mucho más costosa que una memoria EPROM.

2.2.6. Memorias Flash EEPROM.

Las memorias Flash EEPROM, es una versión mejorada de la memoria EEPROM, se ha convertido en una memoria popular y muy utilizada en aplicaciones de microcontroladores. Las memorias Flash EEPROM son no volátiles y por generalmente son muy rápidas. Los datos se pueden borrar y luego ser reprogramado mediante un dispositivo de programación adecuado. Algunos microcontroladores tienen sólo 1K de memoria Flash EEPROM mientras que otros tienen 32K o más. El microcontrolador PIC18F452 tiene 32K bytes de memoria flash.

2.3. Características de los microcontroladores PIC.

Los microcontroladores de fabricantes diferentes no tienen una misma arquitectura. Algunos pueden adaptarse a una aplicación particular, mientras que otros pueden ser totalmente inadecuados para la misma aplicación. Las características de hardware comunes a la mayoría de los microcontroladores se describen en esta sección.

2.3.1. Voltaje de alimentación.

La mayoría de los microcontroladores funcionan con un voltaje de corriente directa de +5V. Algunos microcontroladores pueden funcionar con una fuente de voltaje más baja de +2.7V, y otros llegan hasta +6V sin ningún

problema para que funciones correctamente el PIC. Por lo general, los fabricantes de microcontroladores elaboran hojas de datos por cada MCU, en la que tiene información relevante, por ejemplo, el voltaje límite permitido de alimentación.

Los microcontroladores *PIC16F887* (véase la figura 2.8) pueden funcionar con una fuente de alimentación entre $+2V$ y $+5.5V$.

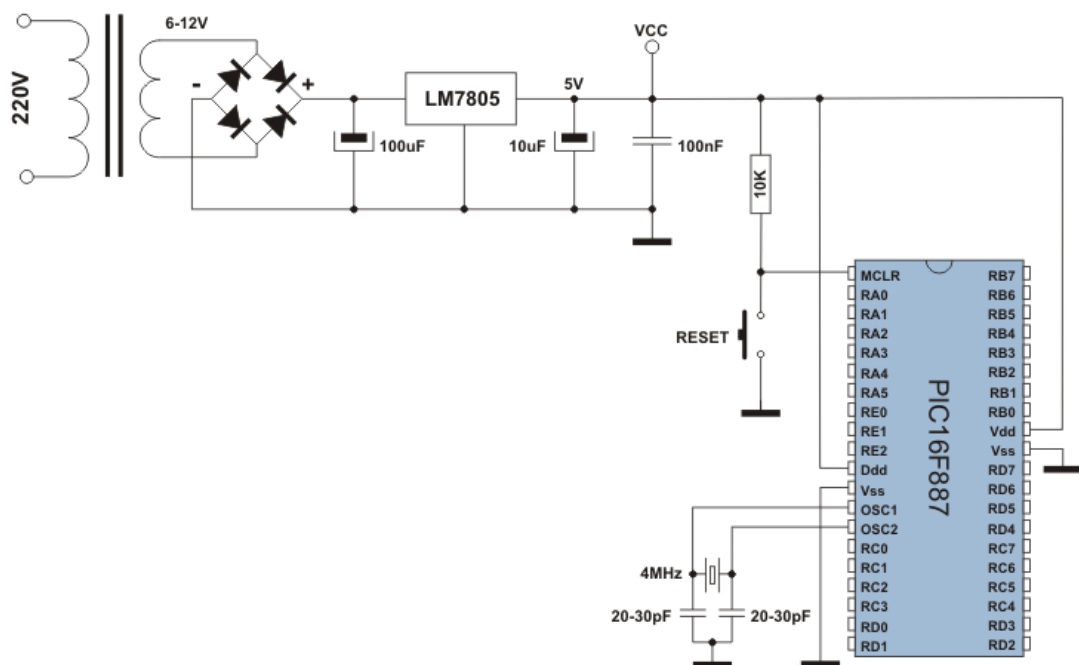


Figura 2. 8: Alimentación mediante regulador de voltaje LM7805.

Fuente: Mikroelektronika

Por lo general, se utiliza un circuito regulador de voltaje para obtener un voltaje de alimentación requerida cuando el dispositivo se alimenta desde una batería o banco de pilas. Por ejemplo, se requiere un regulador (véase la figura 2.8 el LM7805) de 5V si el microcontrolador está usando una batería de 9V.

2.3.2. Señal de Reloj.

Todos los microcontroladores requieren de una señal de reloj, también conocido como oscilador. Este oscilador, permite que el MCU funcione, por lo general estos dispositivos de temporización externos son conectados al microcontrolador. En la mayoría de los casos, estos dispositivos de temporización externos pueden ser cristales de cuarzo más dos pequeños capacitores, tal como se muestra en la figura 2.9.

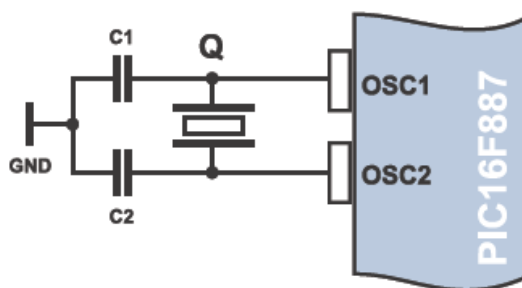


Figura 2. 9: Conexión del cristal de cuarzo al PIC 16F887.
Fuente: Mikroelektronika

Los tipos de osciladores de cuarzo son: LP, XT y HS. La tabla 2.1 muestra los tipos de osciladores de cuarzo y los valores correspondientes de los capacitores C1 y C2.

Tabla 2. 1: Modos de osciladores de cuarzo.

Modo	Frecuencia	C1, C2
LP	32 KHz	33pF
	200 KHz	15pF
XT	200 KHz	47-68 pF
	1 MHz	15 pF
	4 MHz	15 pF
HS	4 MHz	15 pF
	8 MHz	15-33 pF
	20 MHz	15-33 pF

Fuente: Mikroelektronika

En algunos casos son resonadores cerámicos, también conocidos como CR tal como se muestra en la figura 2.10. La tabla 2.2 muestra los modos para un resonador cerámico.

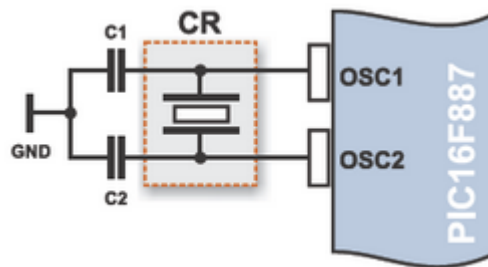


Figura 2. 10: Resonador cerámico.
Fuente: Mikroelektronika

Tabla 2. 2: Modos de un resonador cerámico.

Modo	Frecuencia	C1, C2
XT	455 KHz	68-100 pF
	2 MHz	15-68 pF
	4 MHz	15-68 pF
HS	8 MHz	10-68 pF
	16 MHz	10-22 pF

Fuente: Mikroelektronika

Otro tipo es el oscilador RC (Resistivo-Capacitivo) tal como se muestra en la figura 2.11. Algunos microcontroladores tienen incorporados los circuitos de temporización, es decir, que no requieren componentes de temporización externos.

2.3.3. Temporizadores Timers.

Los temporizadores son parte importante de cualquier microcontrolador. Un temporizador es básicamente un contador que es accionado a prueba de un pulso de reloj externo u oscilador interno del

microcontrolador. Un contador de tiempo puede ser de 8 bits o 16 bits. Los datos pueden ser cargados en un temporizador bajo control de programa, y el temporizador puede ser detenido o iniciado por el control del programa. La figura 2.11 muestra el diagrama de bloques de la estructura de temporizadores.

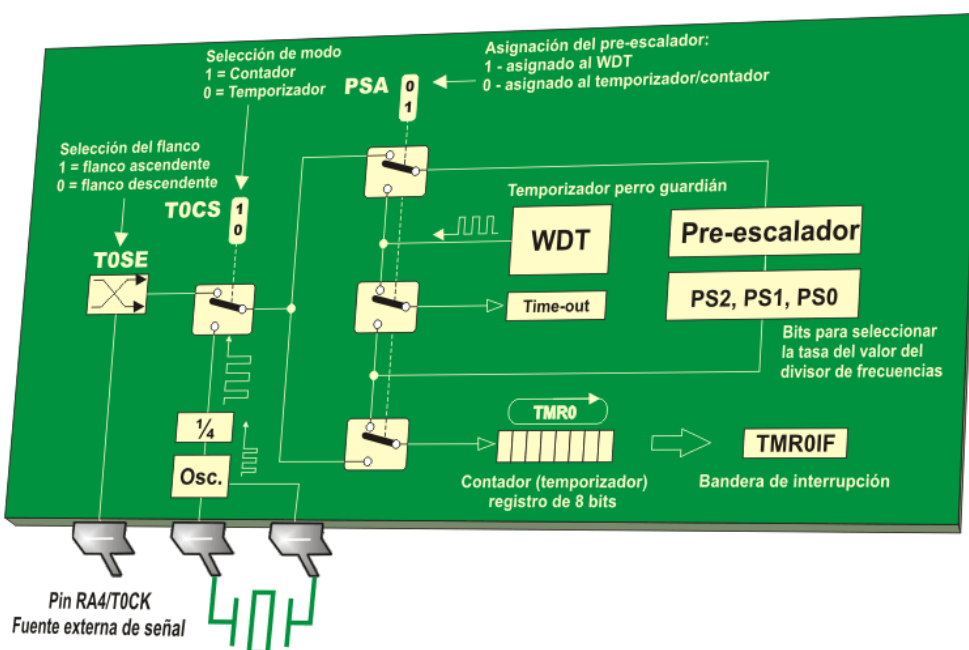


Figura 2. 11: Diagrama de bloques de temporización.
Fuente: Mikroelektronika

La mayoría de los temporizadores pueden configurarse para generar una interrupción cuando alcanzan un cierto conteo (normalmente cuando se desbordan). Un programa puede utilizar una interrupción para llevar a cabo operaciones relacionadas con la temporización precisa dentro del microcontrolador. Los microcontroladores de la familia PIC18F tienen al menos tres temporizadores. Por ejemplo, el microcontrolador PIC18F452 tiene tres temporizadores incorporados.

2.3.3.1. Perro guardián (watchdog).

La mayoría de los microcontroladores tienen al menos una función de vigilancia (watchdog). El módulo de control es básicamente un temporizador que se actualiza del programa de usuario. Cada vez que el programa no puede refrescar el módulo de control, se produce un reset. La figura 2.12 muestra la configuración del watchdog para ser conectada al oscilador RC.

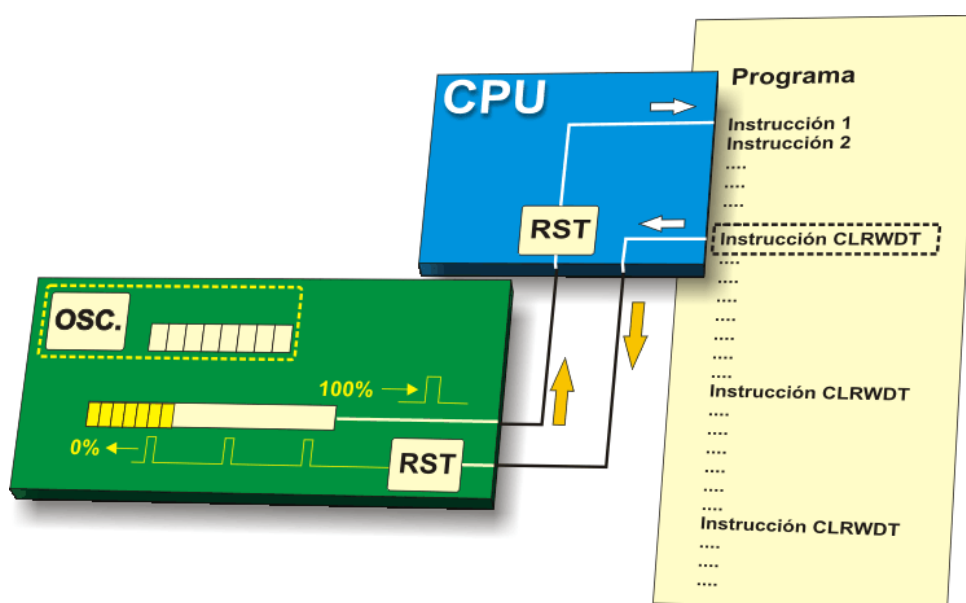


Figura 2. 12: Conexión de watchdog con osciladores R.
Fuente: Mikroelektronika

2.3.4. Convertidores Analógico/Digital.

Los microcontroladores utilizan datos digitales para representar todo. Incluso la música, vídeos e imágenes, todas son representadas como datos digitales, que es una serie de datos lógicos de '0' y '1'. Sin embargo, nuestros datos del mundo real no son digitales sino más bien analógicos. Se dice con razón: "Vivimos en un mundo analógico, pero los datos son procesados en forma digital".

Datos del mundo real como la luz, la temperatura, la presión, el calor, la altura, la distancia, la velocidad, la fuerza, etc. todo son datos analógicos. Para utilizar estos datos tenemos que adquirirlos con sensores o transductores específicos y luego convertirlos a formato digital para su uso dentro de mundo digital de un microcontrolador o microprocesador. Muchos microcontroladores requieren un chip ADC externo para implementar esto, sin embargo esta característica ha sido muy bien integrada en los microcontroladores PIC tal como se muestra en la figura 2.13.

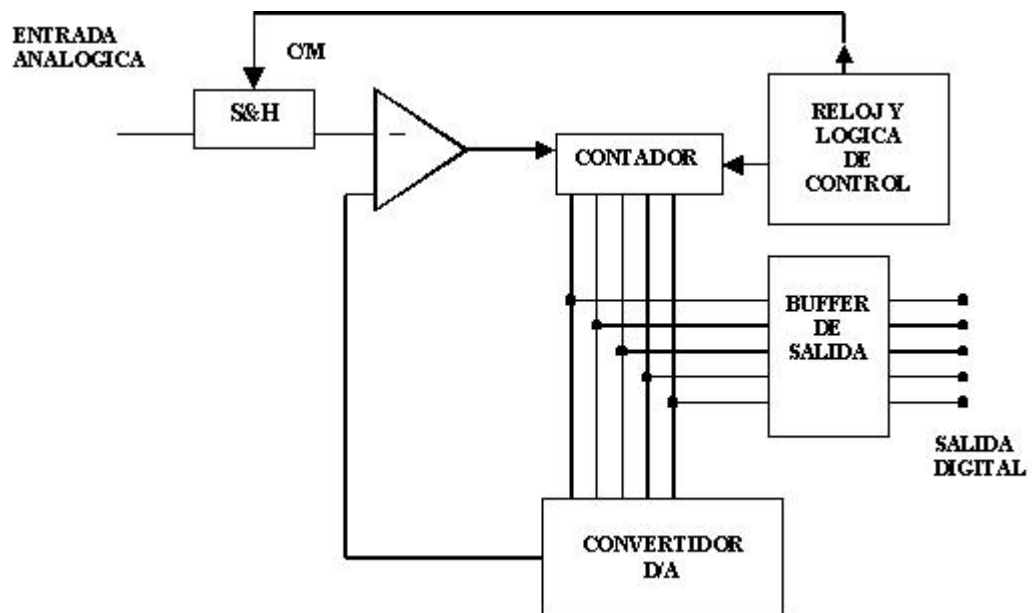


Figura 2. 13: Conexión de watchdog con osciladores R.
Fuente: Jiménez V., W., & Ochoa C., D., (2010)

2.4. Arquitecturas de los microcontroladores.

Hay dos tipos de arquitecturas que son convencionales en microcontroladores. La arquitectura Von Neumann (véase la figura 2.14), utilizada por un gran porcentaje de los microcontroladores, pone todo el

espacio de memoria en el mismo bus; la instrucción y los datos también utilizan el mismo bus.

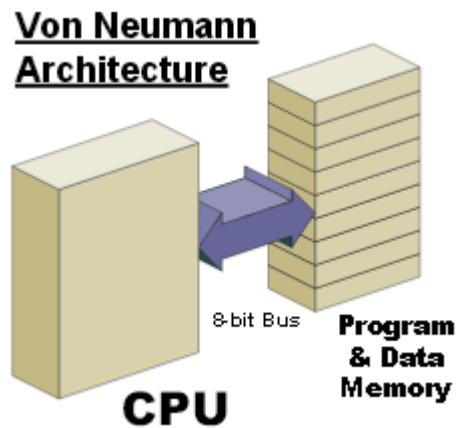


Figura 2. 14: Arquitectura Von Neumann.
Fuente: Pazmiño C., C., & Romero V., D., (2013).

La figura 2.15 muestra la arquitectura de Harvard (usado por los microcontroladores PIC), el código y los datos se encuentran en buses separados, lo que les permite ser traída al mismo tiempo, lo que resulta en un mejor desempeño.

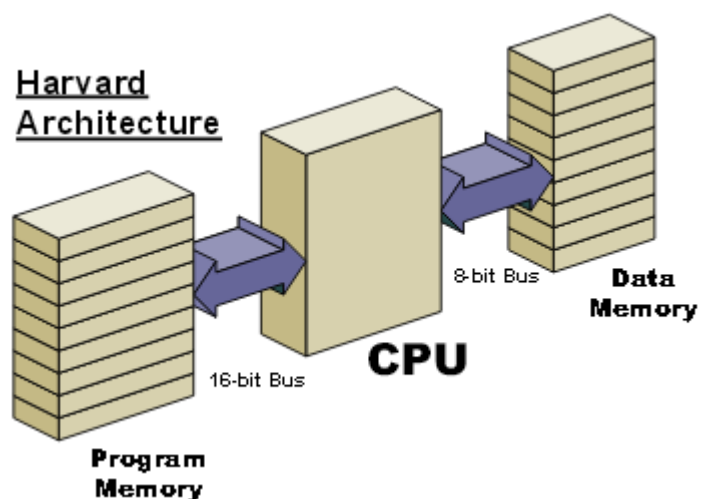


Figura 2. 15: Arquitectura Harvard.
Fuente: Pazmiño C., C., & Romero V., D., (2013).

2.4.1. RISC.

RISC (computadora de conjunto de instrucciones reducido) y CISC (ordenador instrucción compleja) se refieren al conjunto de instrucciones de un microcontrolador. En un microcontrolador RISC de 8 bits, los datos tienen 8 bits de ancho, pero las palabras de instrucción son más de 8 bits de ancho (generalmente 12, 14, o 16 bits) y las instrucciones ocupan una palabra en la memoria de programa. Así, las instrucciones se recuperan y son ejecutadas en un ciclo, lo que mejora el rendimiento.

2.4.2. CISC.

En un microcontrolador CISC, tanto los datos e instrucciones son 8 bits de ancho. Microcontroladores CISC por lo general tienen más de doscientos instrucciones. Los datos y el código están en el mismo autobús y no se pueden recuperar de forma simultánea.

2.5. Otros Sistemas de Entrenamiento de Microcontroladores.

Esta sección se describirá brevemente otros tipos de sistemas de entrenamiento de microcontroladores, que son: PICAXE y PIC EDUTRONICA JJTSA.

2.5.1. Entrenador de Microcontroladores PICAXE

PICAXE incluye un conjunto de módulos (circuitos) que son independientes entre sí. Estos módulos vienen etiquetados los diferentes elementos que lo componen, también dispone de terminales de conexión. Es

decir, que podemos utilizar un cable telefónico para enviar señales (datos) de control de un módulo a otro. La figura 2.16 muestra el diagrama de bloques del módulo de entrenamiento PICAXE.

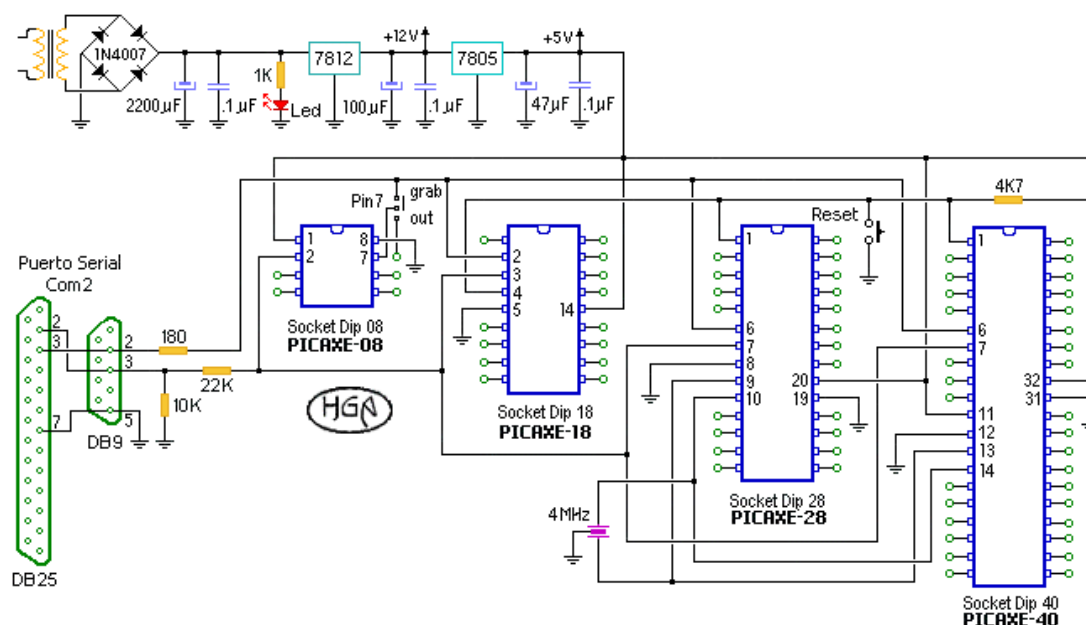


Figura 2. 16: Diagrama por bloques del entrenador PICAXE.
Fuente: PICAXE.

En la figura 2.16 se muestra dos señales del puerto serial (DB25), que son 2 y 7, que son las que permiten la conexión del circuito con el computador fijo o móvil. Para Curtidor, A., Herrera, C., & Ariza, D., (2011). El puerto serial de entrada (pin 2) permite la transmisión de datos desde el computador (software o compilador) hasta el microcontrolador deseado. En cambio, el puerto serial de salida (pin 3) envía información del microcontrolador hasta el computador donde este leerá los datos desde el programador. Finalmente, viene integrado el puerto (pin 7) para conexión a tierra.

código de programa viejo de la memoria. La memoria sólo permite el almacenamiento de un programa a la vez.

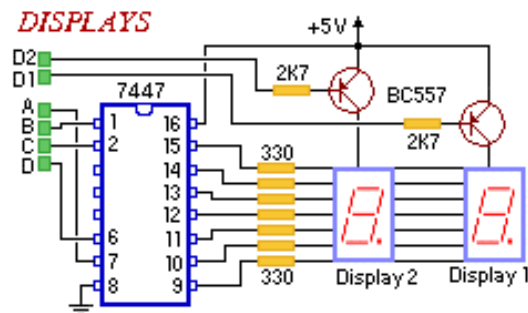


Figura 2. 18: Circuito esquemático de la conexión de los 2 displays.
Fuente: PICAXE.

La figura 2.18 muestra dos displays (D1 y D2) de ánodo-común, que están conectados al mismo bus de datos mientras que los ánodos son manejados de forma independiente. Para visualizar la información en los display utilizamos el decodificador SN74LS47 que permite la conversión de datos previamente enviados por el microcontrolador (A, B, C, D).

La figura 2.19 muestra el esquemático del conjunto de LEDs, es decir, que nos muestra 4 LEDs (que son L1, L2, L3, L4). Para encenderlos se necesita un nivel lógico alto en su respectivo acceso.

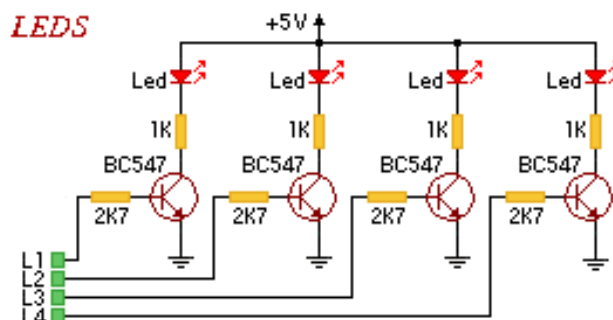


Figura 2. 19: Circuito esquemático de la conexión de 4 LEDs.
Fuente: PICAXE.

En la figura 2.20 se muestra la SALIDA CON RELE, donde se utiliza un relé para manejar cargas de potencia desde el microcontrolador. Para su activación, se requiere de un nivel lógico alto en el pin R.

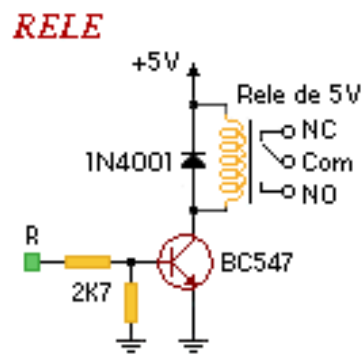


Figura 2. 20: Circuito esquemático de la salida con relé.
Fuente: PICAXE.

En la figura 2.21 se muestra la conexión de PULSADORES, el cual consta de dos interruptores (pulsadores) normalmente abiertos (P1, P2), conectados a tierra con resistencias de 10K. En los pines de conexión se tiene normalmente un nivel lógico (0V) bajo, pero cuando se oprimen los botones se obtiene un nivel lógico (5V) alto.

PULSADORES

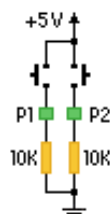


Figura 2. 21: Circuito esquemático de pulsadores.
Fuente: PICAXE.

En la figura 2.22 se muestra la conexión del DIPSWITCH, el mismo tiene cuatro interruptores (S1, S2, S3, S4), los cuales se conectan a tierra

con resistencias de $10K\Omega$. Los pines de conexión del Dip switch se tiene un nivel lógico bajo si el interruptor está abierto, pero cuando el interruptor se cierra, se obtiene un nivel lógico alto.

DIPSWITCH

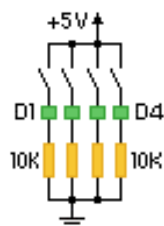


Figura 2. 22: Circuito esquemático del Dip-switch.
Fuente: PICAXE.

En la figura 2.23 se ilustra la conexión del TECLADO MATRICIAL, este tiene un teclado con una matriz de 3×4 , con caracteres de 0 a 9, *, y #. Las columnas (véase los pines C1, C2, C3) del teclado se conectan resistencias de $2,7K\Omega$ para fijar un nivel lógico alto cuando no se está oprimiendo ninguna tecla.

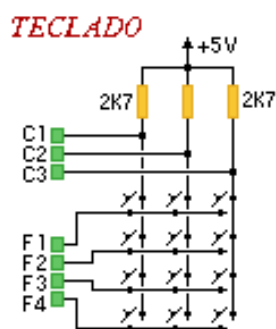


Figura 2. 23: Circuito esquemático del Teclado Matricial.
Fuente: PICAXE.

La figura 2.24 muestra el circuito esquemático de la conexión BUZZER, el mismo crea en forma automática sonidos a diferentes frecuencias utilizando el comando sound.

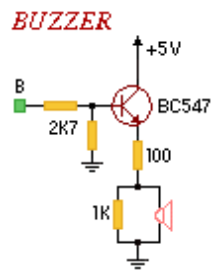


Figura 2. 24: Circuito esquemático del Buzzer.
Fuente: PICAXE.

En el circuito de la figura 2.10 se ilustra el circuito para los dispositivos de salida, que provee de siete salidas (+5 Vcc o +12 Vcc) con óptima capacidad de corriente. Para amplificar la corriente de salida del microcontrolador se utiliza el IC ULN2003 como driver de los transistores de salida. Para que los BJTs de salida estén habilitados (ENABLE) los pines (s1 hasta S7) de conexión deben tener un nivel lógico alto. En caso de que los BJTs de salida conmuten a tierra la carga debe colocarse entre las salidas (1 hasta 7), y la alimentación ya sea +5 Vcc, o +12 Vcc.

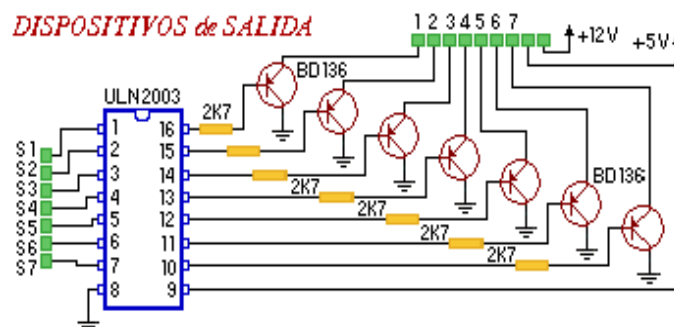


Figura 2. 25: Circuito esquemático de los dispositivos de salida.
Fuente: PICAXE.

2.5.2. Módulo de entrenamiento PIC EDUTRONICA JJTSA.

El módulo de entrenamiento de esta sección se compone de dos tarjetas: Quemador (Programador) y el sistema de evaluación de programas.

Mediante este sistema se pueden elaborar y probar una serie de programas que se aproximan a la realidad de control que alguien quiera realizar. Se ahorra, tiempo y esfuerzo al evitar conexiones en un “protoboard”, el cual puede generar demasiados problemas como son falsos contactos e incertidumbre en el funcionamiento al no saber dónde está la falla.

La figura 2.26 muestra el programador que permite grabar diversos modelos de microcontrolador PIC de gama media, mediante el uso de dos bases una de 40 pines y otra de 18 pines. Para mayor flexibilidad, se puede insertar una base tipo ZIF (Zero In Force) sobre la base de 40 pines para programar la mayoría de los modelos. Cuenta además con un conector que puede usarse para programar externamente circuitos en línea.

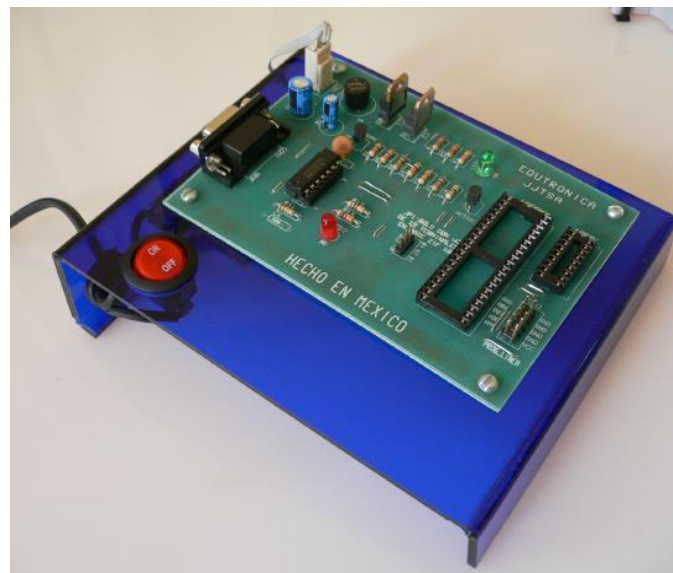


Figura 2. 26: Tarjeta de la etapa de programación de PIC.
Fuente: Edutrónica.

La figura 2.27 muestra la tarjeta que permite realizar la evaluación de programas, tiene una gran flexibilidad en el manejo de señales de entradas

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



Figura 2. 27: Tarjeta de la etapa de evaluación de programas.
Fuente: Edutrónica.

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

- 1) Display de Cristal Líquido de 2x20 caracteres (LCD).
- 2) Doble Display de 7 segmentos.
- 3) Puertos con salida de LED's.
- 4) Un puerto con Interruptor Táctil.
- 5) Interruptor táctil para interrupción externa (RB0/INT).
- 6) Buzzer Piezoeléctrico.
- 7) Sensor analógico de luz (Fotorresistencia).
- 8) Sensor de Temperatura (LM35DZ).

- 9) Dos resistencias variables para voltaje variable (0-5 VCD).
- 10) Sensor infrarrojo para decodificación de señales infrarrojas.
(Control Remoto).
- 11) Interruptor de Reset.
- 12) Conector “header” macho del Puerto B para cargas externas al sistema.
- 13) Conector “header” macho para programación en línea del PIC, sin necesidad de estar quitando el PIC para programarlo.

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

CAPÍTULO 3: DESARROLLO DE APLICACIONES PRÁCTICAS DEL PIC18F4550.

3.1. Aplicación práctica 1: Secuencia de encendido de LEDs.

La figura 3.1 muestra el circuito que permite realizar una secuencia de encendido de LEDs, el PIC 18F4550 se encarga de ejecutar tal función. A continuación se explicará brevemente, la programación del circuito de la secuencia de encendido de LEDs. Primero declaramos al reloj que estará trabajando a una frecuencia de 20 MHz con un cristal externo (Fuses XT). En el programa principal declaramos que todos los pines del puerto D inicialicen con un cero lógico en cada puerto, es decir, que no se encenderá ningún LED que esté conectado en esos pines.

```

1  #include <18f4550.h>
2  #use delay (clock=20000000)
3  #fuses xt
4  #byte portd=0xf83
5
6  void main()
7  {
8      set_tris_d(0x00);
9      while(true)
10     {
11         portd=0xf0;
12         delay_ms(500);
13         portd=0x0f;
14         delay_ms(500);
15     }
16 }
17

```

La función “While” especifica un loop infinito y damos como un “1” lógico a las primeras salidas del puerto D, es decir, a los cuatro primeros pines, D0, D1, D2, D3, con el fin de encender los LEDs conectados, y los

últimos cuatro pines en cero lógico para tener que apagarlos. En este momento esperamos 500 milisegundos para cambiar totalmente las salidas, ya que los primeros 4 pines pasan de un “1” lógico a un “0” lógico, apagando los primeros 4 LEDs y dando paso a encender la segunda mitad del total de los pines del puerto D, y finalmente esperamos 500 milisegundos para que se repita la secuencia.

Este es un ejemplo básico de cómo se puede automatizar ordenes gracias al PIC18F4550, y no solo eso ya que podríamos automatizar cualquier cosa que deseemos, por ejemplo apertura de puertas, de relés, etc. La figuras 3.1 muestra el funcionamiento de los primeros 4 LEDs (D0, D1, D2 y D3) encendidos, mientras que la figura 3.2 muestra el funcionamiento del resto de LEDs (D5, D6, D7 y D8)

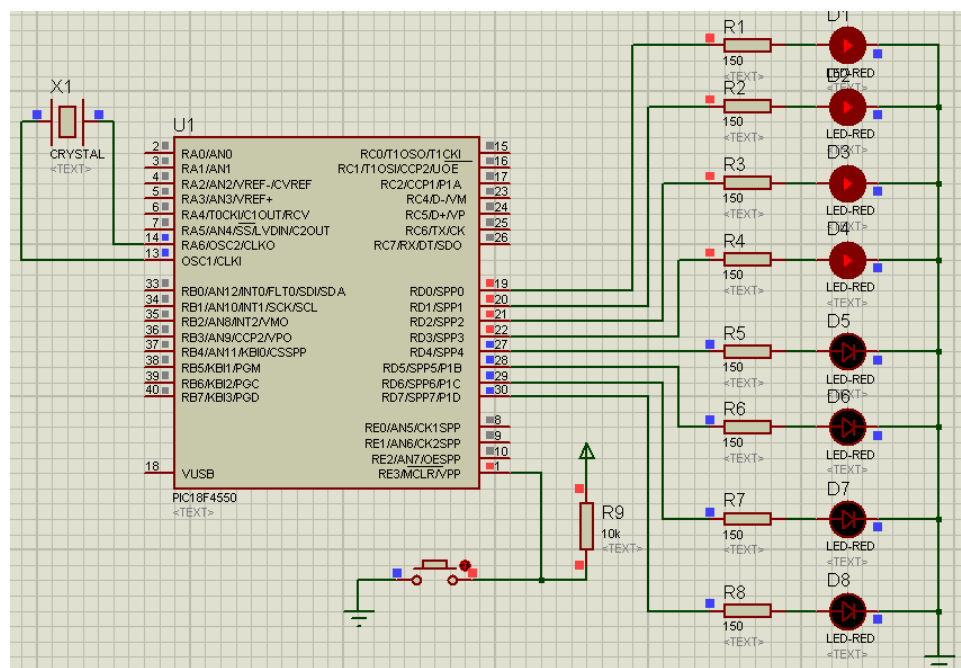


Figura 3. 1: Encendido de los primero cuatro LEDs del puerto D
Elaborado por: El Autor.

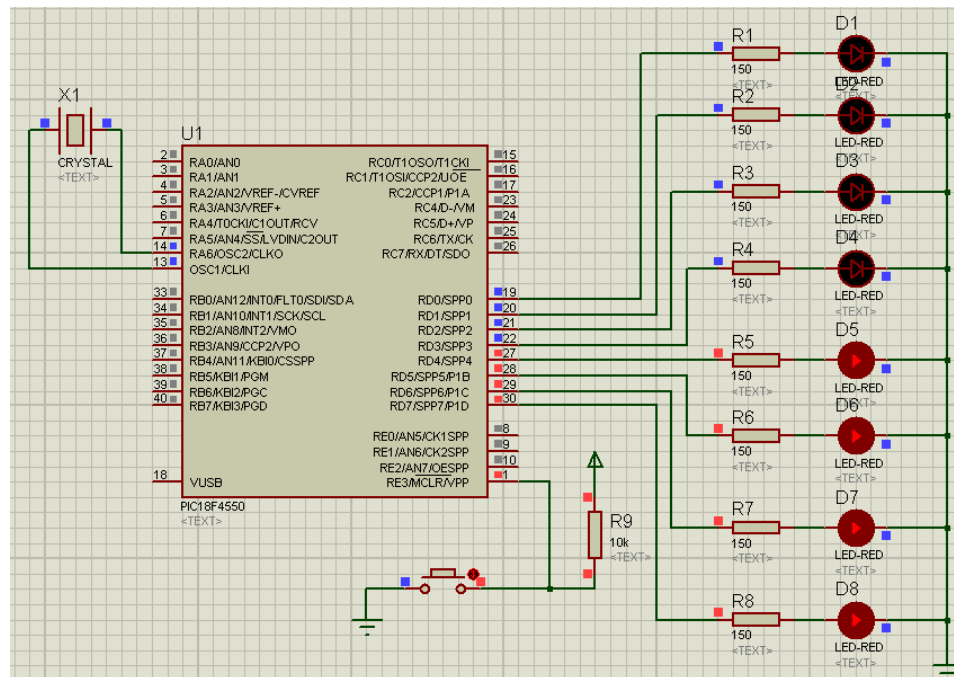


Figura 3. 2: Encendido de los últimos cuatro LEDs del puerto D
Elaborado por: El Autor.

3.2. Aplicación práctica 2: Seguridad con Clave.

Mediante esta aplicación práctica nos damos cuenta de la gran utilidad que tienen los Microcontroladores a nivel de seguridad. El diseño consiste en conectar un teclado numérico junto al microcontrolador PIC18F4550, así como también se conectan a él, un LCD 20x4 y dos LEDs que permite indicarnos la salida lógica de los pines. La figura

A continuación se explica el programa desarrollado en C. Consiste en solicitar al usuario una contraseña previamente designada en el código fuente, luego si dichos números ingresados son incorrectos el programa automáticamente lo detecta y enciende el LED que está conectado al pin C7, dando un valor de "1" lógico.

Al ingresar los números correctamente el programa lo detecta y enciende automáticamente el LED que tenemos conectado al pin C6, para demostrar el valor 1 lógico de esta salida. Ahora ciertamente tenemos en la simulación conectada un LED, pero esto claramente podría ser cambiado para conectar a un circuito que active cualquier dispositivo del que se desee interactuar. El microcontrolador PIC 18F4550 nos da la facilidad de utilizarlo según nuestra necesidad. El programa en C para esta aplicación se describe de la siguiente manera:

```

1  #include <18f4550.h>
2  #use delay (clock=4000000)
3  #fuses xt
4  #byte portc=0xf82
5  #byte portd=0xf83
6  #include <lcd.c>
7  #include <KBD.c>
8
9  int digito1,digito2,digito3,digito4;
10
11 void ingreso_clave(void)
12 {
13     int k=0;
14     while(k==0)
15         k=kbd_getc();
16     k=k-48;
17     digito1=k;
18     lcd_gotoxy(5,2);
19     lcd_putc("**");
20
21     k=0;
22     while(k==0)
23         k=kbd_getc();
24     k=k-48;
25     digito2=k;
26     lcd_gotoxy(6,2);
27     lcd_putc("**");
28
29     k=0;
30     while(k==0)
31         k=kbd_getc();
32     k=k-48;

```

```

33     digito3=k;
34     lcd_gotoxy(7,2);
35     lcd_putc("*");
36
37     k=0;
38     while(k==0)
39     k=kbd_getc();
40     k=k-48;
41     digito4=k;
42     lcd_gotoxy(8,2);
43     lcd_putc("*");
44 }
45
46 void chequeo_clave(void)
47 {
48     if( digito1==1 & digito2==2 & digito3==3 & digito4==4)
49     {
50         lcd_putc("\fCLAVE CORRECTA\n");
51         output_high(PIN_C6);
52         delay_ms(1000);
53         output_low(PIN_C6);
54     } else
55     {
56         lcd_putc("\fCLAVE INCORRECTA\n");
57         output_high(PIN_C7);
58         delay_ms(2000);
59         output_low(PIN_C7);
60     }
61 }
62
63

```

En la cabecera tenemos el “include” que llama la librería del PIC a utilizar, los fuses, la velocidad de reloj, el driver del LCD y por último el driver del teclado numérico “KBD.c”. También declaramos variables globales de tipo entero para los dígitos que el usuario ingresa.

Realizamos una función que permita ingresar una clave a través del teclado, la clave es un número digitado por el teclado, estos son recibidos en el PIC 18F4550 como código ASCII y por ello le restamos 48 a todos los números para obtener el valor real y se asigna a cada valor una variable que declaramos al principio del programa.

Realizamos otra función, que permita la comparación de los valores de las variables explicadas anteriormente con los valores que nosotros deseamos que pertenezca nuestra clave, si es correcto le asignamos un valor lógico “1” a un pin y si es incorrecto asignamos un valor lógico al otro pin, logrando en la simulación el encendido del LED correcto e incorrecto.

Posteriormente, en el programa principal después de dar el saludo correspondiente solicitamos al usuario que ingrese la clave dentro de un lazo “while” que está configurado como un lazo infinito.

```
64 void main()
65 {
66
67     set_tris_d(0x00);
68     set_tris_c(0x0f);
69     lcd_init();
70     kbd_init();
71
72     lcd_putc("hola buenos dias");
73     delay_ms(1000);
74     lcd_putc("\f");
75 while(true)
76 {
77     lcd_putc("\fDIGITE LA CLAVE\n");
78     ingreso_clave();
79     chequeo_clave();
80 }
81 }
```

La figura 3.3 muestra el diseño del circuito mediante acceso por clave en ISIS Proteus.

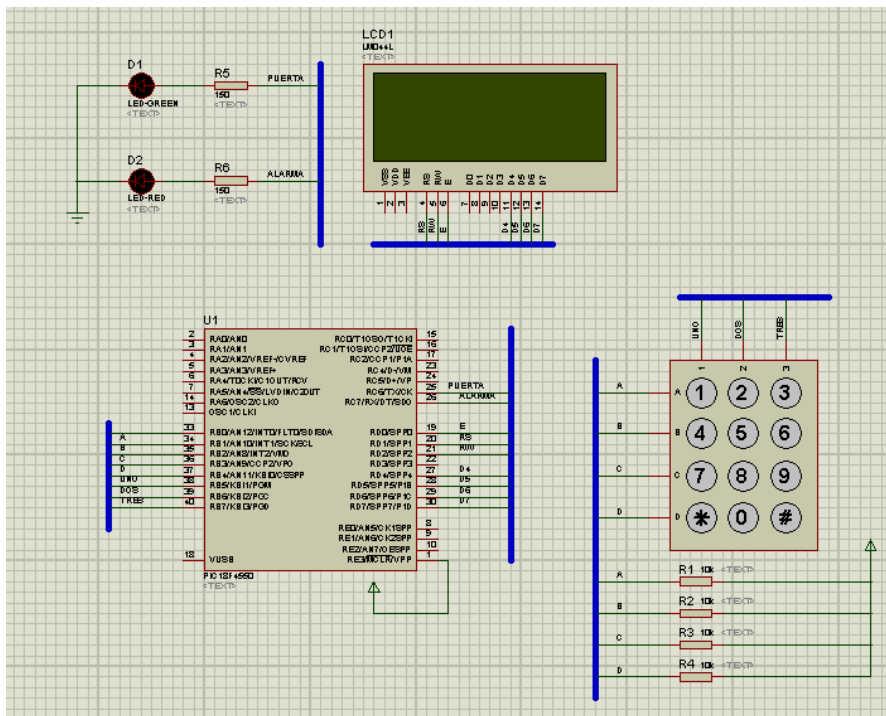


Figura 3. 3: Conexión del teclado y LCD en los puertos B y D del PIC 18F4550.
Elaborado por: El Autor.

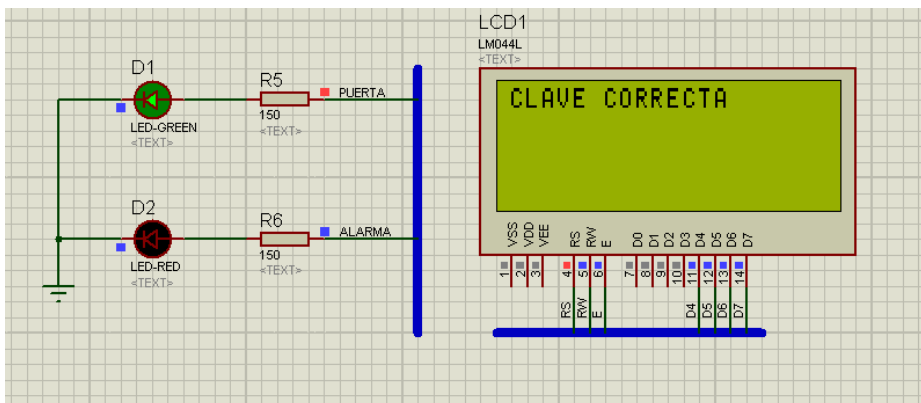


Figura 3. 4: Primer caso, cuando la clave digitada es correcta en comparación al guardado en el código de programación.
Elaborado por: El Autor.

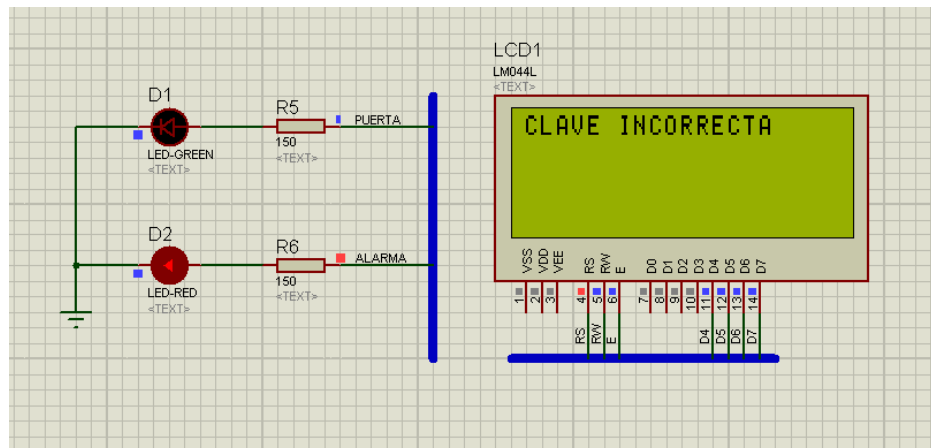


Figura 3. 5: Segundo caso, cuando la clave digitada no coincide con los dígitos que están en el código de programación.

Elaborado por: El Autor.

3.3. Aplicación práctica 2: Alarma con temporizador y varios periféricos

Esta aplicación del microcontrolador está orientada a la seguridad de una casa. Inicialmente, al ser ejecutado el PIC 18F4550 solicita al usuario un tiempo para activar la alarma, debido a que el usuario necesita un tiempo mínimo para poder salir de la casa sin activar los sensores que están debidamente colocados en sitios que permitan acceder al domicilio.

Aquí es donde tenemos conectado 4 sensores, dos de ellos normalmente abiertos y dos cerrados. Estos sensores al ser alterados envían un "0" lógico a las entradas del PIC 18F4550 previamente programado.

La programación del código es un poco extensa pero muy sencillo al momento de entender su algoritmo:

```

1  #include <18f4550.h> //micropic
2  #device adc=8
3  #fuses XT          //Cristal EXTERNO
4  #use delay(clock=4000000)
5  #include <lcd.c>
6
7  float a,dato;
8  int dat, n, m, time;
9  char c,p;
10
11 void intruso()
12 {
13     while(input(PIN_B1))
14     {
15         lcd_putc("\fINTRUSO \nINTRUSO");
16
17         output_high(PIN_B6);
18         delay_ms(800);
19         output_low(PIN_B6);
20         output_high(PIN_B7);
21         delay_ms(800);
22         output_low(PIN_B7);
23         delay_ms(800);
24         lcd_putc("\f");
25     }
26 }
27
28
29 void settime()
30 {
31

```

```

32     printf(lcd_putc, "\f");
33     printf(lcd_putc, "CALIBRAR TIEMPO \nEN SEGUNDOS ");
34     delay_ms(750);
35     printf(lcd_putc, "\f");
36
37     while(input(PIN_B1)) //IGUALAMOS EL TIEMPO Y AL PRESIONAR EL PIN SALIMOS DE AQUI
38     {
39         output_high(PIN_C7);
40         lcd_init();
41         set_adc_channel(0);
42         delay_us(100);
43         a = read_adc();
44         dato=(a * 24)/256;
45         dat=(int)dato;
46         printf(lcd_putc, "TIME = %d ", dat);
47         delay_ms(300);
48         printf(lcd_putc, "\f");
49     }
50
51
52     printf(lcd_putc, "TIEMPO CALIBRADO");
53     delay_ms(750);
54     printf(lcd_putc, "\f");
55
56 }
57
58
59 void main()
60 {
61     setup_adc_ports(AN0);
62     setup_adc(ADC_CLOCK_INTERNAL);

```



```

63
64
65     lcd_init();
66     devolucion: ;
67     printf(lcd_putc, "\f",);
68     printf(lcd_putc, "INICIANDO\n");
69     delay_ms(1000);
70     printf(lcd_putc, "\f",);
71     printf(lcd_putc, "ALARMA \nINTELIGENTE");
72     delay_ms(1000);
73     printf(lcd_putc, "\f",);
74     printf(lcd_putc, "PROGRAMA DE\nTITULACION");
75     // delay_ms(2000);
76     output_high(PIN_B6);
77     delay_ms(800);
78     output_low(PIN_B6);
79     output_high(PIN_B7);
80     delay_ms(800);
81     output_low(PIN_B7);
82     output_high(PIN_C7);
83     delay_ms(800);
84     output_low(PIN_C7);
85     printf(lcd_putc, "CALIBRE TIEMPO\nDE SALIDA");
86     settime(); //llamar tiempo para salir en segundos
87     printf(lcd_putc, "PRESIONE BOTON A \nPARA ARMAR");
88     delay_ms(800);
89     while(input(PIN_B1));
90     output_low(PIN_C7);
91     printf(lcd_putc, "\f");
92
93
94     for(n=0; n<=dat; n++)

```

```

95         {
96
97             m=dat-n;
98             printf(lcd_putc, "TIEMPO SALIR\n    %d SEGUNDOS "m);
99             output_high(PIN_C7);
100            delay_ms(500);
101            output_low(PIN_C7);
102            delay_ms(500);
103            printf(lcd_putc, "\f");
104
105
106        }
107
108
109     printf(lcd_putc, "\f",);
110     printf(lcd_putc, "ARMADO.....");
111
112
113     while(input(PIN_B1))
114     {
115
116         if(input(PIN_B2)){
117             delay_ms(10);
118             if(input(PIN_B3)){
119                 delay_ms(10);
120                 if(input(PIN_B4)){
121                     delay_ms(10);
122                     if(input(PIN_B5)){
123                         delay_ms(10);
124                     }
125                 }
126                 else{
127                     delay ms(10000);

```

```

127         intruso();
128         goto devolucion;}
129     }
130     else{
131         delay_ms(10000);
132         intruso();
133         goto devolucion;
134     }
135 }
136 else{
137     delay_ms(10000);
138     intruso();
139     goto devolucion;
140 }
141 }
142 else{
143     delay_ms(10000);
144     intruso();
145     goto devolucion;
146 }
147 }
148 }
149 goto devolucion;
150 }

```

Explicación del código:

Al inicio llamamos las librerías pertenecientes al pic18f4550, luego describimos que el convertidor ADC sea de 8bits.

Las siguientes líneas estamos llamando a los fuses y la velocidad del cristal de cuarzo externo.

Declaramos dos funciones, una para alertar de los intrusos activando dos LED en forma secuencial y enviando un mensaje al LCD que diga la palabra “INTRUSO”.

Y otra función es para solicitar el tiempo en segundos la cual el usuario tendrá disponible para salir de la oficina o casa, y claro tener todos los accesos cerrados para que cada sensor este en la posición.

Al inicio del programa principal activamos los puertos analógicos que comprenden desde el pin A0 hasta el pin A7 y otros pines que se encuentran en los puertos B y E, pero en nuestro caso de nuestra pequeña alarma utilizaremos el puerto AN0, ya que solo utilizamos un potenciómetro para variar el tiempo que necesitaremos para la salida.

Luego el programa efectúa una serie de saludos e indicaciones para ser impresa en la pantalla LCD y esta pueda ser vista por el usuario, el programa solicita que presione un botón para iniciar la función de "settime" la cual es la que solicita el tiempo.

La función "settime" solicita al usuario iniciar el conteo al presionar el botón de aceptación, en este caso botón A, y procede a realizar el contador.

Al terminar el conteo el programa ingresa a un while el cual es interrumpido siempre y cuando el valor sea de cero lógico de un pin específico.

Dentro el while el programa se encarga de realizar una revisión continua de los pines donde se encuentra conectado los sensores, en el caso que encuentre una alteración en su estado este dará 10 segundos para poder ser reiniciado o desactivado el micro con el otro pulsador, caso contrario ejecutara la función "intruso".

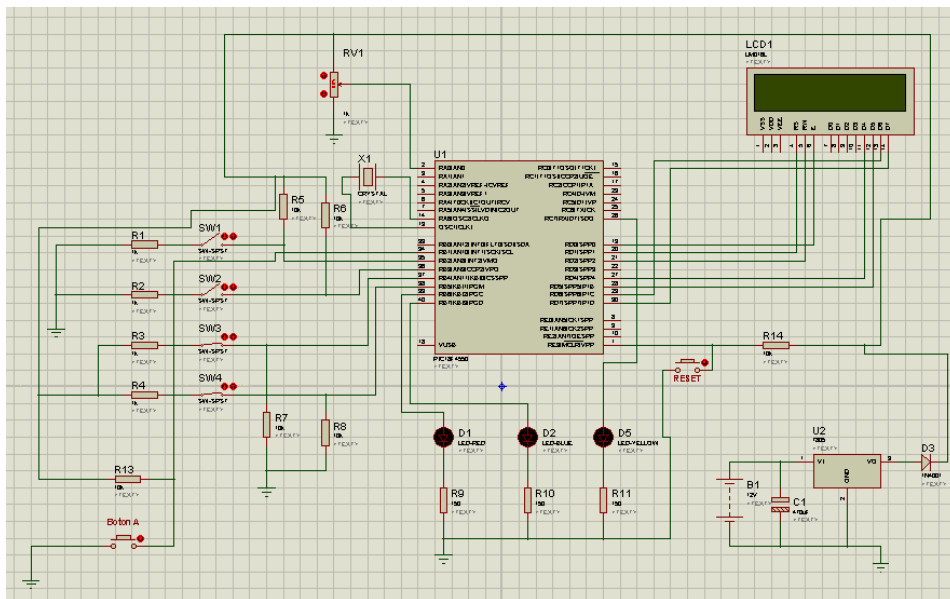


Figura 3. 6: Diseño de la simulación del circuito con temporizador.
Elaborado por: El Autor.

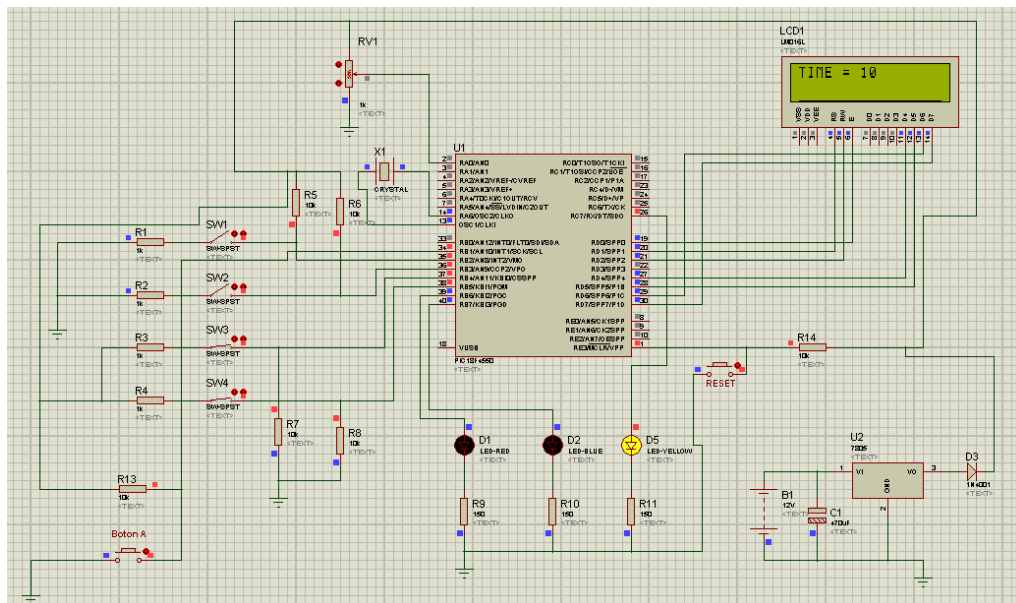


Figura 3. 7: LCD muestra al usuario el tiempo no alarmar el sistema.
Elaborado por: El Autor.

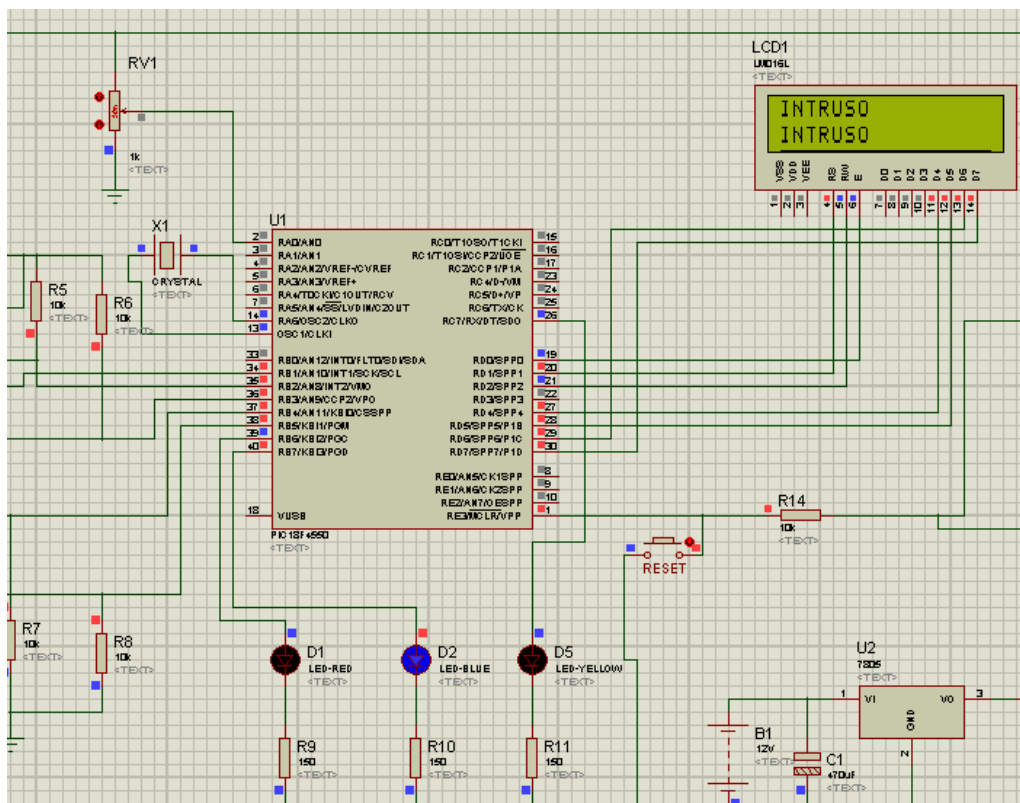


Figura 3. 8: LCD recibe señal del sensor que detecta un cambio de estado.
Elaborado por: El Autor.

3.4. Comunicación serial con el PIC18F4550.

Este es un sencillo programa que nos ayuda a comprender el uso que se le puede dar un microcontrolador PIC18F4550 para las comunicaciones, estas nos permitirán utilizarlo como ejemplo en la comunicación de un computador y un micro para el control y automatización de cualquier caso que se requiera.

En este caso tenemos como ejemplo en la simulación un microcontrolador que por medio de un Hiperterminal podemos observar una secuencia de números del 0 al 9 que el PIC nos envía por medio del protocolo RS232.

En la cabecera de nuestro código llamamos siempre las librerías del micro 18f4550, consecuentemente especificamos la velocidad del cristal o reloj interno dependiendo de los fuses, luego especificamos el protocolo de comunicación y los pines que esta se realizara.

Declaramos una variable global llamada contador de tipo entero, después continuamos y escribimos nuestro código de interrupción de comunicación, dentro de esta función obtenemos los datos por medio del pin de recepción y convertimos el valor restando 48 ya que en se trabaja con el código ASCII para las comunicaciones.

En el mismo LCD imprimimos los valores que se reciben. En el programa principal se encarga de enviar los datos, una secuencia del 0 al 9, por medio del pin de transmisión al Hiperterminal donde podemos observarlo, al momento que nosotros en esta pantalla presionamos una tecla numérica sea cual sea el programa automáticamente es interrumpido para ejecutar la función que se explicó anteriormente.

Este programa y simulación es un claro ejemplo de las comunicaciones, siendo estas en algún momento necesarias para el control y automatización. Este PIC 18F4550, está especialmente hecho para las comunicaciones ya que cuenta con un puerto nativo de USB la cual es posible una comunicación más fácil entre el micro y un ordenador.

En el ordenador se pueden realizar programas o aplicaciones para el control de domótica por ejemplo, en el pc le ordenamos a un PIC encender una luz y esta orden es recibida por el PIC que este ejecutando un programa y la cual ya tiene instrucciones prediseñadas para realizar ciertas funciones con determinados mandos.

Código del programa:

```

1  #include <18f4550.h> //micropic
2  #fuses XT           //Cristal EXTERNO
3  #use delay(clock=4000000)
4  #use rs232(baud=9600, xmit=pin_c6,rcv=pin_c7)
5  #include <lcd.c>
6  int contador;
7
8  #int_RDA
9  void RDA_isr()
10 {
11
12     contador=getc();
13     contador=contador - 48;
14     lcd_gotoxy(1,2);
15     printf(lcd_putc, "Recibiendo=%1d", contador);
16
17 }
18
19 void main(void)
20 {
21     int valor;
22     lcd_init();
23     enable_interrupts(INT_RDA);
24     enable_interrupts(GLOBAL);
25
26     while(true)
27     {
28         for( valor=0; valor<=9; valor++){
29             putc(valor);
30             lcd_gotoxy(1,1);
31             printf(lcd_putc, "Enviando=%1d", valor);
32             delay_ms(1000);
33         }
34     }
35 }

```

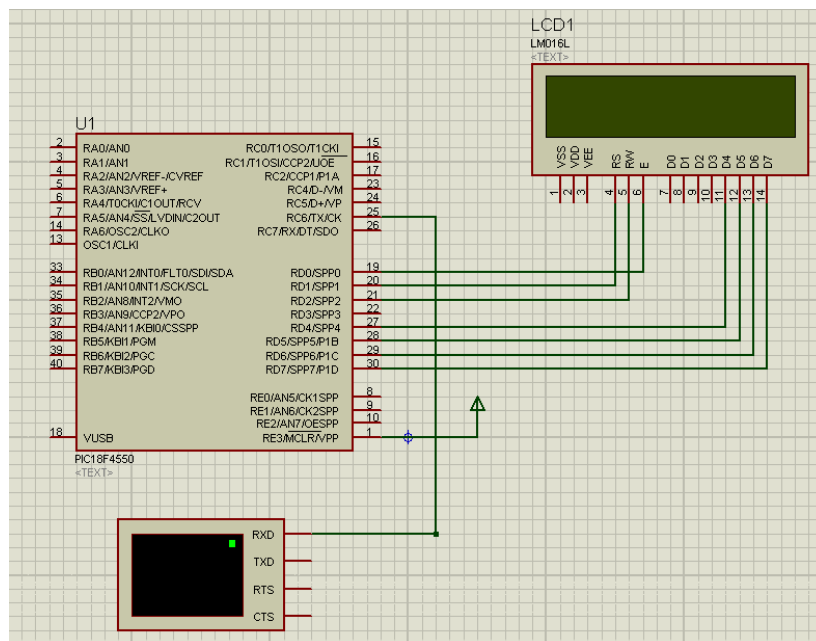


Figura 3. 9: Diseño de la simulación del circuito mediante comunicación serial.
Elaborado por: El Autor.

El diagrama muestra un LCD conectado en el puerto D y una Hiperterminal para recibir la información que el PIC genera.

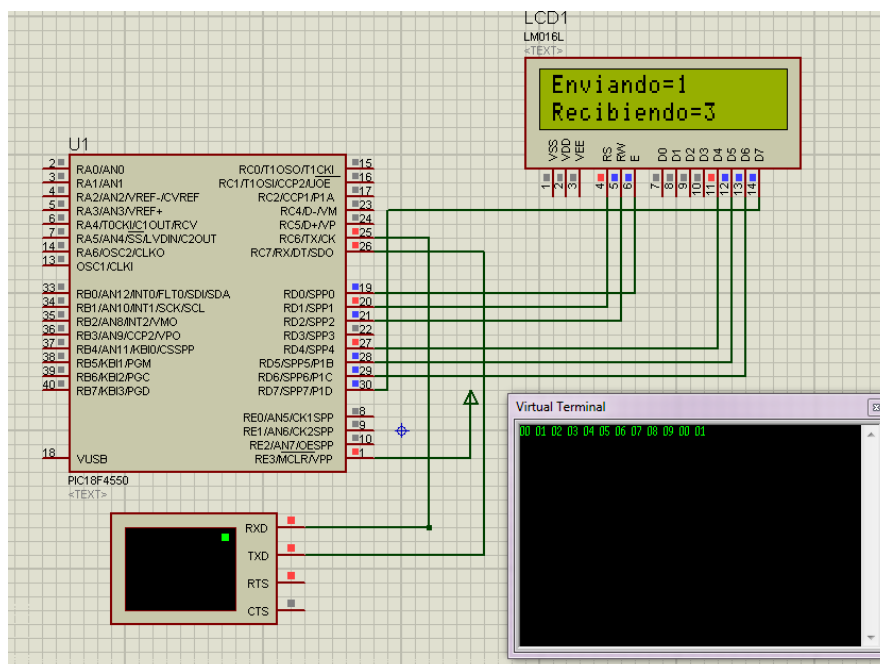


Figura 3. 10: Resultado obtenido mediante comunicación serial.
Elaborado por: El Autor.

CAPÍTULO 4: CONCLUSIONES Y RECOMENDACIONES.

4.1. Conclusiones.

- Definición clara de las habilidades de los estudiantes para formar futuros profesionales comprometidos con la sociedad y el conocimiento, en su recorrido y estudio de la asignatura de Microcontroladores, tanto en práctica como en la teoría, a partir de los objetivos que persigue la asignatura.
- El programa Compiler C CSS resulta amigable para programadores que han tenido experiencia en programación C.
- Se conocen cada uno de los recursos con los que debe contar los diferentes módulos de entrenamiento de microcontroladores, para manejarse de acuerdo al conjunto de acciones a emprender en el desarrollo de las prácticas por los estudiantes en el laboratorio de electrónica.
- Los microcontroladores PIC18F4550 nos permite controlar de una manera fácil los diferentes tipos de displays y arreglos de leds gracias a la considerable cantidad de pines bidireccionales que este contiene.
- El módulo de entrenamiento para microcontroladores PIC18F4550 se implementa como una plataforma o entrenador abierto, dinámico y

que evoluciona en forma constante, para así integrar diversos dispositivos electrónicos para aplicaciones en domótica, robótica, procesos industriales, entre otras.

- El sistema de entrenamiento PIC18F4550 será utilizado en la materia de microcontroladores y materias afines que aparecen la malla curricular de la Carrera de Ingeniería en Telecomunicaciones.

4.2. Recomendaciones.

- Tener a disposición toda la ayuda bibliográfica y de software necesario acerca de la programación en Compiler C CSS para que los estudiantes puedan incrementar sus conocimientos en programación.
- A partir del módulo de entrenamiento PIC18F4550 diseñado por la compañía ecuatoriana APM (*All Power Microcontroller*), incentivar a los alumnos a diseñar y construir nuevos sistemas modulares de entrenamiento o maquetas que permitan trabajar en forma similar al entrenador PIC18F4550.
- Que la Facultad de Educación Técnica para el Desarrollo promueva cada año concursos internos de robótica y de proyectos que utilicen microcontroladores PIC u otros similares, para así apreciar sus diseños y creatividad en beneficio de la colectividad.

- Por último, sería necesario el despliegue de esta herramienta en la Maestría en Telecomunicaciones y puedan desarrollar proyectos de investigación avanzados.

REFERENCIAS BIBLIOGRÁFICAS

Barra Z., O., & Barra Z., F. (2011). *Microcontroladores PIC con programación PBP*. Editorial Ra-Ma, S. A.

Beleño, K., Pardo, A., & Torres, I. (2011). *Sistema de Posicionamiento solar utilizando la Plataforma de Labview y el microcontrolador PIC18F4550*. Revista Colombiana de Tecnologías de Avanzada, 2(18).

Curtidor, A., Herrera, C., Ariza, D., (2011). *Diseño y construcción de módulos entrenadores para programación de microcontroladores*. Trabajo de grado de Ingeniero en Telecomunicaciones, Universidad de San Buenaventura, Sede Bogotá.

Betancourt G., R., (2011). *Diseño y Desarrollo de un Sistema de Control y Monitoreo para tanques de maduración de gelatina con Microcontroladores PIC*. Repositorio digital de las tesis de Ingeniería en Computación de la Universidad Nacional Autónoma de México.

Fonseca, J., Cazares, G., Montes, J., Del Ángel, H., & Aguilar, C., (2011). *Módulo de entrenamiento basado en microcontroladores PIC: diseño e implementación*. Revista de Sociedad, Cultura y Desarrollo Sustentable, Ra Ximhai, 7(3).

García B., E. (2009). *Compilador C CSS y Simulador PROTEUS para Microcontroladores PIC*. Editorial Marcombo - Ediciones Técnicas, Barcelona, España.

Galeano, G. (2009). Programación de sistemas embebidos en C. Volumen 1. Editorial Alfaomega.

Jiménez V., W., & Ochoa C., D., (2010). *Diseño y Estudio de Factibilidad para la Implementación de un Laboratorio de Procesamiento de Señales y Simulación para la Universidad Politécnica Salesiana Sede Cuenca*. Repositorio digital de las tesis de Ingeniería Electrónica de la UPS, Cuenca.

Melchor H., N., (2009). *Tarjeta de Desarrollo para Microcontroladores PIC*. Repositorio digital de las tesis de Ingeniería Electrónica del Instituto Politécnico Nacional, México.

Moreno R., P., & Ávalos Y., J., (2009). *Simulación de circuitos mediante la utilización de microcontroladores PIC para el Laboratorio de Mecatrónica*. Repositorio digital de la Tesis de Ingeniero de Mantenimiento de la Escuela Superior Politécnica de Chimborazo.

Palacios M., E., Remiro D., F., & López P., L. J. (2009). *Microcontrolador PIC16F84: Desarrollo de Proyectos*. Editorial Alfaomega, México.

Pazmiño C., C., & Romero V., D., (2013). *Diseño e Implementación de un prototipo para monitoreo y control remoto mediante GPSR de tablero de medidores comerciales de la Empresa Eléctrica Riobamba S. A.* Repositorio digital de la Universidad Politécnica del Ejército, Latacunga.

Pérez M., Y., (2012). Implementación de Comunicación USB con Microcontrolador PIC18F4550 y LabVIEW. URL www.youblisher.com/pdf/481469.

Valdés P., F. E., & Pallas A., R., (2007). *Microcontroladores: Fundamentos y Aplicaciones con PIC*. Editorial Marcombo S.A., España.
