



ESCUELA DE INGENIERÍA EN SISTEMAS

Tema:

“DISEÑO Y CONSTRUCCIÓN DE UN ENTRENADOR BASADO EN MICROCONTROLADORES PIC, PARA EL LABORATORIO DE ELECTROLOGÍA DE LA ESCUELA DE INGENIERÍA EN SISTEMAS DE LA PUCESA”

**Disertación de Grado previo la obtención del título de
Ingeniero de Sistemas y Computación**

Línea de investigación:

Inteligencia Artificial, Robótica, Domótica y Sistemas Expertos.

AUTOR:

PAÚL FERNANDO TOAPANTA BARONA

DIRECTOR:

ING. DARIO JAVIER ROBAYO JÁCOME

Ambato – Ecuador

Mayo 2015

PONTIFICIA UNIVERSIDAD CATÓLICA DEL ECUADOR

SEDE AMBATO

HOJA DE APROBACIÓN

Tema:

“DISEÑO Y CONSTRUCCIÓN DE UN ENTRENADOR BASADO EN
MICROCONTROLADORES PIC, PARA EL LABORATORIO DE
ELECTROLOGÍA DE LA ESCUELA DE INGENIERÍA EN SISTEMAS DE LA
PUCESA”

Línea de investigación:

Inteligencia Artificial, Robótica, Domótica y Sistemas Expertos.

Autor:

PAÚL FERNANDO TOAPANTA BARONA

Darío Javier Robayo Jácome, Ing. Mg. **f.** _____
CALIFICADOR

Verónica Maribel Pailiacho Mena, Ing. Mg. **f.** _____
CALIFICADORA

Pablo Ernesto Montalvo Jaramillo, Ing. Mg. **f.** _____
CALIFICADOR

Galo Mauricio López Sevilla, Ing. Mg **f.** _____
**DIRECTOR DE LA ESCUELA DE
INGENIERÍA EN SISTEMAS**

Hugo Rogelio Altamirano Villarroel, Dr. **f.** _____
SECRETARIO GENERAL PUCESA

Ambato – Ecuador

Mayo 2015

DECLARACIÓN DE AUTENTICIDAD Y RESPONSABILIDAD

Yo, Paúl Fernando Toapanta Barona, portador de la cédula de ciudadanía No. 180353302-3, declaro que los resultados obtenidos en la investigación que presento como informe final, previo la obtención del título de Ingeniero en Sistemas son absolutamente originales, auténticos y personales.

En tal virtud, declaro que el contenido, las conclusiones y los efectos legales y académicos que se desprenden del trabajo propuesto de investigación y luego de la redacción de este documento son y serán de mi sola y exclusiva responsabilidad legal y académica.

Paúl Fernando Toapanta Barona
C.I. 180353302-3

AGRADECIMIENTO

A Dios por las bendiciones recibidas y las oportunidades para poder seguir adelante.

A mi madre, Graciela, que me ha brindado su apoyo incondicional en las buenas y en las malas para poder culminar con esta gran etapa de mi vida.

A mis hermanos, que me apoyaron e incentivaron para poder superarme.

A mi esposa e hijos, que son la razón de mi esfuerzo, pasado, presente y futuro.

DEDICATORIA

Dedico este trabajo al esfuerzo y sacrificio de mi madre

Paúl.

RESUMEN

La tecnología avanza a pasos muy grandes, por lo que se ve la necesidad de crear nuevas herramientas que faciliten el aprendizaje diario, es por esta razón que el presente trabajo tiene como objetivo principal desarrollar un entrenador en donde los estudiantes puedan programar, borrar, y ejecutar Microcontroladores PIC, a su vez hacer pruebas antes de montar proyectos en el laboratorio de electrología de la Escuela de Ingeniería en Sistemas de la PUCESA. Para el desarrollo del trabajo se utilizó el método concurrente, pues este permitió recopilar datos y características importantes para el diseño de la placa, además se utilizó como soporte a ello la información bibliográfica, con lo que se pudo conocer bases teóricas sobre el tema, para luego ponerlas en práctica, dando como resultado final la construcción de un Entrenador Basado en Microcontroladores PIC (Peripheral Interface Controller), para lo cual se utilizó el lenguaje de programación, el cual permite desarrollar nuevas ideas y crear nueva tecnología, finalmente el desarrollo de este entrenador que servirá de soporte para muchos estudiantes, pues les permitirá contar con una herramienta gratuita para comprobar el eficiente funcionamiento de los proyectos que se encuentren realizando.

Palabras clave: entrenador, microcontroladores PIC, lenguaje de programación.

ABSTRACT

Technology is developing by leap and bounds, that is why it is necessary to create new tools that help daily learning. As a result this research is aiming to develop a training board where students can program, erase and run PIC microcontrollers, and at the same time to test before deploying projects in the electrology lab of the School of Systems Engineering at PUCESA. The current method was used to develop this project because it allowed collecting data and important features to design the board; in addition, literature was used as a support which enable to learn theoretical bases on the subject and then put them in practice, obtaining as a result the design of a training board based on PIC microcontrollers, for which language programming was applied, allowing to establish new ideas and create new technology,.Finally developing a training board will help many students because it will allow them to have a free tool to check the efficient operation of the projects that they are performing.

Keywords: traing board, PIC microcontrollers, language programing.

TABLA DE CONTENIDOS

PRELIMINARES.....	i
DECLARACIÓN DE AUTENTICIDAD Y RESPONSABILIDAD	iii
AGRADECIMIENTO	iv
DEDICATORIA	v
RESUMEN.....	vi
ABSTRACT	vii
TABLA DE CONTENIDOS.....	viii
ÍNDICE DE GRÁFICOS.....	xi
INTRODUCCIÓN.....	1
CAPÍTULO I.....	2
FUNDAMENTOS TEÓRICOS	2
1.1 Antecedentes	2
1.2 Problema	3
1.2.1 Descripción del problema	3
1.2.2 Significado del Problema	3
1.2.3 Preguntas Básicas.....	4
1.3 Justificación	4
1.4 Objetivos	5
1.4.1 Objetivo General	5
1.4.2 Objetivos Específicos.....	5
1.5 Meta.....	6
1.6 Fundamentos Teóricos.....	6
1.6.1 Introducción a los microcontroladores.....	6
1.6.2 Breve reseña histórica	6
1.6.3 Controlador	7
1.6.4 Microcontroladores	9

1.6.5 Diferencias entre Microprocesadores y Microcontroladores	11
1.6.6 Organización de una computadora.....	14
1.6.8 Familias populares de Microcontroladores	24
1.6.9 Lenguajes de programación	25
1.6.9.1 Lenguaje Máquina/ensamblador	26
1.6.9.2 Compiladores	26
1.6.9.3 Simuladores.....	27
1.6.10 Herramientas de desarrollo	27
1.6.11 Elementos de desarrollo.....	27
CAPÍTULO II.....	40
METODOLOGÍA	40
2.1 Metodología de Investigación.....	40
2.1.1 Método General.....	40
2.1.2 Método Específico	40
2.2 Metodología de desarrollo	40
CAPÍTULO III.....	45
RESULTADOS.....	45
3.1 Diseño y Construcción del Entrenador	46
3.1.1 Fase 1: Ideación.....	46
3.1.2 Desarrollo conceptual y básico	47
3.1.3 Desarrollo avanzado.....	48
3.1.3.1 Fase 1: Esquemático Virtual	48
3.1.3.2 Fase 2 Análisis y simulación.....	62
3.1.3.3 Fase 3 Diseño Final Y Generación de PCB	63
3.1.4 Lanzamiento.....	66
3.1.4.1 Fase 1: Elección de Elementos:.....	66
3.1.4.2 Fase 2 Preparación de Baquelita:	69
3.1.4.3 Fase 3 Soldadura:	73
3.1.4.4 Fase 4 Pruebas:.....	74
CAPÍTULO IV	76

ANÁLISIS Y VALIDACIÓN DE LOS RESULTADOS	76
4.1 Procedimientos para la utilización del entrenador	76
4.2 Guía del usuario del entrenador PIC	76
4.3 Programación de un ejercicio completo	88
4.4 Validación de Resultados.....	103
4.4.1 Análisis General	109
CONCLUSIONES Y RECOMENDACIONES.....	110
Conclusiones	110
Recomendaciones	112
BIBLIOGRAFÍA.....	113
ANEXOS	115
Anexo 1: PIC16F877	115
Anexo 2: Encuesta	136

TABLA DE GRÁFICOS

Ilustraciones

Ilustración 1.1: Controlador	9
Ilustración 1.2: Aplicaciones de los microcontroladores	10
Ilustración 1.3: Áreas y Porcentajes de utilización de los Microcontroladores	10
Ilustración 1.4: Arquitectura de una computadora.....	15
Ilustración 1.5: Arquitectura de un Microcontrolador	16
Ilustración 1.6: Vista Simplificada CPU	18
Ilustración 1.7: Arquitectura Harvard	20
Ilustración 1.8: Arquitectura Von Neuman	21
Ilustración 1.9: Módulo Temporizador	22
Ilustración 1.10: Serial Port.....	23
Ilustración 1.11: Conversión analógica a digital.....	24
Ilustración 1.12: Tecnología SMD y THT	28
Ilustración 1.13: PIC	29
Ilustración 1.14: Diodos Led.....	29
Ilustración 1.15: Pulsador.....	30
Ilustración 1.16: LCD 16X12.....	31
Ilustración 1.17: Dip Switch 4	31
Ilustración 1.18: Relé	32
Ilustración 1.19: Conector DB9 Hembra	32
Ilustración 1.20: Display Ánodo Común	33
Ilustración 1.21: Teclado 4x4.....	33
Ilustración 1.22: Zócalo	34
Ilustración 1.23: Conector tipo Jack.....	34
Ilustración 1.24: Cristal 4MHZ	35
Ilustración 1.25: Capacitor Cerámico 22 pf	35
Ilustración 1.26: Resistencia 330 ohmios	36
Ilustración 1.27: Ejemplo Resistencia 330 ohmios	36
Ilustración 1.28: Max 232	37
Ilustración 1.29: Capacitor 0.1uf.....	37

Ilustración 1.30: Transistor 2n3904	38
Ilustración 1.31: Bluetooth HC05	38
Ilustración 1.32: Programador.....	39
Ilustración 2.33: Fases para el Diseño del Circuito.....	42
Ilustración 2.34: Fases para la Construcción del entrenador.....	43
Ilustración 3.35: Selección de File/New/Project.....	49
Ilustración 3.36: Selección de File/New/Schematic.....	49
Ilustración 3.37: Selección para agregar Librerías.....	50
Ilustración 3.38: Documento Resistencia.....	50
Ilustración 3.39: Botón Place wire	51
Ilustración 3.40: Cuadro de diálogo para editar propiedades de un elemento	51
Ilustración 3.41: Esquema Diodos Led	52
Ilustración 3.42: Esquema Diodos LCD	53
Ilustración 3.43: Esquema Teclado 4x4	53
Ilustración 3.44: Esquema Fuente de Poder	54
Ilustración 3.45: Esquema de RS232	54
Ilustración 3.46: Esquema Bluetooth	55
Ilustración 3.47: Esquema Relé de Potencia	55
Ilustración 3.48: Esquema Programador	56
Ilustración 3.49: Esquema Pulsadores.....	56
Ilustración 3.50: Esquema Display 7 segmentos.....	57
Ilustración 3.51: Dip Switch	57
Ilustración 3.52: Potenciómetro	58
Ilustración 3.53: Puerto 18 pines.....	59
Ilustración 3.54: Puerto 28 pines.....	59
Ilustración 3.55: Puerto 40 pines.....	60
Ilustración 3.56: Alimentación y masa.....	60
Ilustración 3.57: Capacitores de 22pf.....	61
Ilustración 3.58: Botón Reset.....	61
Ilustración 3.59: Resistencias de 20k y 10k	61
Ilustración 3.60: Prototipo Capa Superior.....	62
Ilustración 3.61: Prototipo Capa Trasera	63
Ilustración 3.62: Simulación Capa Superior	63

Ilustración 3.63: Simulación Capa Trasera	64
Ilustración 3.64: Plano Capa 1	64
Ilustración 3.65: Plano Capa 2	65
Ilustración 3.66: Plano Capa 3	65
Ilustración 3.67: Transferencia de diseño	70
Ilustración 3.68: Lavado de la Placa	71
Ilustración 3.69: Perforación de la placa.....	71
Ilustración 3.70: Resultado final preparación de la Baquelita Capa superior	72
Ilustración 3.71: Resultado final preparación de la Baquelita Capa posterior	72
Ilustración 3.72: Colocación de elementos en la placa	73
Ilustración 3.73: Soldadura con cautín.....	74
Ilustración 3.74: Fase de pruebas	74
Ilustración 3.75: Resultado Final de del diseño y construcción del Entrenador	75
Ilustración 4.76: Icono CodeStudio.....	77
Ilustración 4.77: Ingreso al Software	77
Ilustración 4.78: Selección del microcontrolador	78
Ilustración 4.79: Interruptores DipSwitch.....	78
Ilustración 4.80: Opción Compile	81
Ilustración 4.81: Archivo Leds.Hex	82
Ilustración 4.82: Colocación del PIC en el entrenador	82
Ilustración 4.83: Orden de conexión	83
Ilustración 4.84: Software PICKIT2	83
Ilustración 4.85: Mensaje del Software cuando no reconoce el dispositivo	84
Ilustración 4.86: Mensaje del Software cuando reconoce el dispositivo	85
Ilustración 4.87: Selección File/Import Hex	86
Ilustración 4.88: Selección de archivo Leds.Hex.....	86
Ilustración 4.89: Botón Grabar.....	86
Ilustración 4.90: Secuencia de luces módulo led	87
Ilustración 4.91: Estados del Potenciómetro	87
Ilustración 4.92: Cumplimiento de expectativas	103
Ilustración 4.93: Comprobación del funcionamiento de proyectos	104
Ilustración 4.94: Utilización del entrenador simple y entendible	105
Ilustración 4.95: Adaptación de las funciones del entrenador a las necesidades	106

Ilustración 4.96: Ayuda para evitar pérdida de tiempo y dinero	107
Ilustración 4.97: Calificación del entrenador universal	108

Tablas

Tabla 1.1: Tabla Resumen de las Familias de Microcontroladores	25
Tabla 3.2: Elección de Elementos	66
Tabla 4.3: Procedimientos para utilizar el entrenador	76
Tabla 4.4: Cumplimiento de expectativas	103
Tabla 4.5: Comprobación del funcionamiento de proyectos	104
Tabla 4.6: Utilización del entrenador simple y entendible	105
Tabla 4.7: Adaptación de la funciones del entrenador a las necesidades.....	106
Tabla 4.8: Ayuda para evitar pérdida de tiempo y dinero	107
Tabla 4.9: Calificación del entrenador universal	108

INTRODUCCIÓN

El presente trabajo de investigación, se lo realizó con el objetivo de diseñar y construir un entrenador basado en microcontroladores PIC para el Laboratorio de Electrología de la Escuela de Ingeniería en Sistemas. La estructura del trabajo está conformada por cuatro capítulos a saber:

Capítulo I. Fundamentación Teórica, donde se explica la necesidad de construir el entrenador PIC, la definición, la justificación, además los objetivos divididos en un objetivo general y cuatro específicos, finalmente se fundamenta teóricamente temas importantes que intervienen en la construcción del entrenador basado en microcontroladores PIC.

Capítulo II. Metodología, aquí se detalla los métodos de investigación general y específico que se utilizará en la investigación, además se explica las fases de la metodología con la que se realizará el diseño y la construcción del entrenador basado en microcontroladores PIC.

Capítulo III. Resultados, en esta sección se desarrolla y documenta las fases del diseño y construcción de un entrenador basado en microcontroladores PIC, para el laboratorio de electrología de la Escuela de Ingeniería en Sistemas de la PUCESA, en el periodo académico 2014.

Capítulo IV. Análisis y validación de los resultados, en este punto se evidencia la utilidad práctica y la validación de resultados del Entrenador, basado en microcontroladores PIC, además las conclusiones y recomendaciones obtenidas en el desarrollo del trabajo de investigación se expondrán en este capítulo.

CAPÍTULO I

FUNDAMENTOS TEÓRICOS

1.1 Antecedentes

La Escuela de Ingeniería en Sistemas, tiene como misión formar profesionales con ética social y científicamente comprometidos en responder a nuevas realidades tecnológicas, con un alto espíritu propositivo e investigador, actualmente la Escuela se encuentra impulsando varios proyectos con el objeto de alcanzar los fines que se propuso, es así que pretende ofrecer al estudiante la utilización de metodologías y herramientas acordes con la gestión de las nuevas tecnologías de la información y comunicación, para que los estudiantes cuenten con herramientas para realizar los trabajos que se refiere al desarrollo de proyectos.

Siendo un entrenador basado en microcontroladores Programmable Interrupt Controller (PIC), una herramienta importante, con la cual aún no cuenta el laboratorio de electrología, este contribuiría a que los estudiantes elaboren proyectos individuales con la seguridad de que la programación del PIC funcione correctamente e implementen en sus proyectos, sin pérdida de tiempo y dinero, pues no tendría que volverlos a desarrollar, al no contar con una herramienta de entrenamiento como ésta.

La inutilización de esta herramienta no se ha dado principalmente porque su precio no es muy accesible para la mayoría de alumnos, además la inexistencia de esta herramienta en el laboratorio ha dificultado que los estudiantes practiquen la

programación de microcontroladores y realicen las pruebas que sean necesarias para verificar su buen funcionamiento, para posteriormente montarlos en los proyectos con la certeza de que están funcionando eficientemente.

Otro motivo es que, estas tarjetas no se las encuentran a disposición, o no se ajustan a las necesidades, lo que deja muchas interrogantes y no permiten al estudiante desarrollar los proyectos completamente.

Por todo lo anteriormente descrito, la idea de este trabajo es diseñar y construir una tarjeta entrenadora basado en microcontroladores PIC, con soporte para varios modelos, de tal manera que facilite el aprendizaje de los alumnos en el laboratorio de electrología de la Escuela de Ingeniería en Sistemas de la PUCESA y así se pueda contribuir con el cumplimiento efectivo de la misión y los objetivos planteada por la institución.

1.2 Problema

1.2.1 Descripción del problema

La inexistencia de un entrenador basado en microcontroladores PIC, en el laboratorio de electrología de la Escuela de Ingeniería en Sistemas de la PUCESA.

1.2.2 Significado del Problema

Esta investigación, proyecta como base la carencia de un entrenador de microcontroladores PIC, para el correcto aprendizaje, prácticas y desarrollo de proyectos de los estudiantes que utilizan el laboratorio de electrología de la Escuela

de Ingeniería en Sistemas de la PUCESA.

1.2.3 Preguntas Básicas

1.2.3.1 ¿Por qué se origina?

Porque no existe en el laboratorio de electrología un entrenador basado en microcontroladores PIC, que permita a los estudiantes identificar las falencias en el desarrollo de sus proyectos.

1.3 Justificación

El entrenador de microcontroladores PIC planteado, mejorará el desarrollo del aprendizaje del estudiante que utilice el laboratorio de electrología de la PUCESA, pudiendo hacer prácticas en éste antes de montarlo en su proyecto final. Esto ayudará a tener mejores posibilidades de éxito en su desarrollo de trabajos, sabiendo con exactitud que microcontrolador PIC utilizar y conocer las ventajas que cada uno de éstos le ofrecen, evitando el desperdicio de tiempo, dinero, frustrarse al desarrollar placas PCB para alguna rutina de software y que por algún motivo no funcione.

La falta de un dispositivo en donde realizar las pruebas de los proyectos de los estudiantes con microcontroladores PIC queda resuelto, generando beneficios tanto para el desarrollo de nueva tecnología por parte de los estudiantes, como para la colectividad académica de la universidad, mejorando el método de aprendizaje.

Todo esto se lo puede hacer gracias al software, con el cual se conseguirá controlar el hardware, teniendo como base la lógica de programación, que aprueba manipular las

acciones a realizar con información que se tenga.

En la actualidad existe un avance tecnológico muy grande, que permite incluso interactuar con máquinas, en muchos campos tales como: automotriz, agrícola, médico, espacial, investigación, etc. Son capaces de realizar trabajos tan perfectos como en el caso automotriz con brazos robóticos coordinados que en cuestión de minutos ya tienen un trabajo terminado, llegando a lugares difíciles para el ser humano y optimizando todos los recursos.

1.4 Objetivos

1.4.1 Objetivo General

Diseñar y construir un entrenador basado en microcontroladores PIC, para el Laboratorio de electrología de la Escuela de Ingeniería en Sistemas de la PUCESA.

1.4.2 Objetivos Específicos

- Desarrollar un entrenador PIC, que sirva para los alumnos como herramienta de entrenamiento útil en aplicaciones relacionadas con robótica y electrónica.
- Ayudar a desarrollar el correcto aprendizaje de los alumnos.
- Integrar varias herramientas en un solo dispositivo para realizar pruebas con microcontroladores.
- Realizar una placa que permita al estudiante determinar los componentes necesarios para el diseño de su propio proyecto.

1.5 Meta

Desarrollar un entrenador PIC, para elevar la confianza y disminuir la pérdida de tiempo y dinero de los estudiantes al momento de realizar sus proyectos.

1.6 Fundamentos Teóricos

1.6.1 Introducción a los microcontroladores.

Tomando como referencia a (Angulo, 2010), se puede decir que los microcontroladores en los últimos tiempos se han convertido en parte de la vida moderna en general de niños, adolescentes y adultos, pues están presentes en el trabajo, en la casa y se los puede encontrar controlando el funcionamiento de los computadores, teléfonos celulares, hornos microondas y televisores. Es por todo esto que se puede evidenciar el inicio de la conquista masiva de estos diminutos computadores, que con el pasar del tiempo gobernarán en la fabricación de la mayor parte de los aparatos que usan los seres humanos.

En conclusión los microcontroladores, están inmersos en todas las actividades diarias que se realizan, pues estos son circuitos programados para ejecutar órdenes, se los encuentran en los aparatos electrónicos que se utilizan en la vida cotidiana ya sea en el trabajo o en el hogar.

1.6.2 Breve reseña histórica

Según lo expresado por (Romero, 2011) :

En el año 1969, un grupo de ingenieros de origen japonés, representantes de la empresa BUSICOM, viajaron a Estados Unidos con la idea de contactarse con Marcian Hoff, representante de INTEL, pues querían usar para sus proyectos pocos circuitos integrados como los que se usaban en las calculadoras; la propuesta le hicieron a esta empresa porque tenía experiencia trabajando con una computadora (PC) PDP8, es así como se inició la fabricación de estos circuitos, pero a Marcian Hoff, se le ocurrió pensar en una solución diferente en lugar a la construcción sugerida, ésta consistía en que la función del circuito integrado se determinaría por un programa almacenado en él, es decir que la configuración sería más simple, pero que requeriría mucho más memoria de lo que requería el proyecto que propusieron los ingenieros japoneses, después de un tiempo la idea de Marcian ganó, y el primer microprocesador nació.

Para transformar la idea en un producto real, General Instrument logra utilizar micro código simple y almacenado en la ROM para realizar tareas, era uno de los primeros pic, llamado pic1650, denominados también microcomputadoras de un solo chip que más tarde se lo llamaran microcontroladores.

Los fabricantes de circuitos comenzaron la difusión de la noticia, era un chip integrado de memoria y dispositivos de entrada y salida para el control de más circuitos.

1.6.3 Controlador

Tomando como referencia lo expresado por (Valdés, 2011) se puede decir que un controlador es:

Un dispositivo que permite dirigir uno o varios procesos, es un elemento del software usado por un sistema que permite interactuar con el hardware.

Anteriormente los controladores se construían únicamente con componentes de lógica discreta, después se emplearon los microprocesadores, que son chips de memoria y E/S sobre una tarjeta de circuito impreso, los cuales prácticamente son una computadora en una sola tarjeta (SBC, Single Board Computer).

Cabe indicar que, aunque un controlador prácticamente posee todos los elementos necesarios para ser considerado como una computadora, no lo es porque su uso característico consiste en el desempeño de funciones de “control” interactuando con el “mundo real” para monitorear situaciones por medio de sensores y como respuesta a ello, encender o apagar dispositivos por medio de actuadores.

En la actualidad, todos los elementos del controlador se han podido incluir en un circuito integrado, el cual recibe el nombre de microcontrolador y por lo general están enfocados a sistemas de propósito específico, como por ejemplo: máquinas de escribir, videojuegos, cajas registradoras, equipos de sonido, videocámaras, hornos microondas, instrumentos musicales, fotocopiadoras, etc.

En conclusión un controlador es un manual de instrucciones que permite conocer cómo se puede manejar y controlar un dispositivo en general, es decir es un elemento indispensable considerado como software sin el cual el hardware no puede funcionar.



Ilustración 1.1: Controlador

Fuente: Microcontroladores: Fundamentos y Aplicaciones con PIC

1.6.4 Microcontroladores

Según (López, 2011) se puede definir a un microcontrolador como:

Un equipo presente en un solo circuito integrado con tecnología VLSI (Very Large Scale Integration - integración en muy alta escala), que contiene una Unidad Central de Procesos (CPU), y se dedica a realizar una tarea y ejecutar una aplicación específica.

Contiene periféricos de memoria de entrada y salida programables, los microcontroladores están diseñados principalmente para el desarrollo de aplicaciones y se utilizan mucho en los dispositivos electrónicos controlados automáticamente, tales como: teléfonos móviles, cámaras, hornos de microondas, lavadoras, etc.



Ilustración 1.2: Aplicaciones de los microcontroladores

Fuente: Microcontroladores PIC: Sistema integrado para el autoaprendizaje

Según el fabricante de los microcontroladores MICROCHIP, las compras de estos circuitos las realizan empresas que se dedican a varias actividades entre las que se destacan las siguientes:

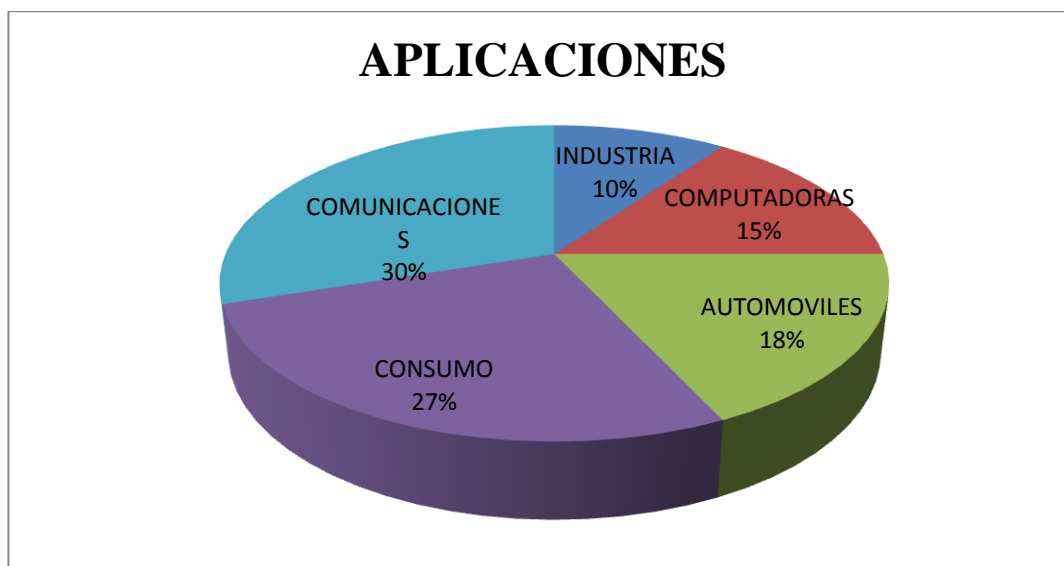


Ilustración 1.3: Áreas y Porcentajes de utilización de los Microcontroladores

Fuente: MICROCHIP

En conclusión los microcontroladores son utilizados para la fabricación de aparatos electrónicos, pues este circuito permite grabar instrucciones para que posteriormente sean ejecutadas en el uso de estos aparatos.

1.6.5 Diferencias entre Microprocesadores y Microcontroladores

Los microprocesadores y microcontroladores, son dispositivos con varias similitudes pero también tienen diferencias importantes entre ellos y estas son:

1.6.5.1 Hardware

Tomando como referencia a (Santamaría, 2011), se puede decir que existen las siguientes diferencias:

Microprocesadores

- Es un circuito integrado que tiene sólo la CPU dentro de ellos, es decir, solo los poderes de procesamiento como de Intel Pentium 1,2,3,4, Core 2 Duo, i3, i5 etc.
- No tienen memoria RAM, ROM y otros periféricos en el chip, por lo que un diseñador del sistema tiene que añadirlos externamente para que sean funcionales

Microcontroladores

- Tiene la CPU y además una cantidad fija de memoria RAM, ROM y otros periféricos todo integrado en un solo chip, se denomina como un mini ordenador o un ordenador en un solo chip.
- Tiene varias limitantes, entre las que se destacan las siguientes:
 - ✓ Tamaño de los datos
 - ✓ Velocidad de procesamiento
 - ✓ Espacio de direccionamiento de memoria

En conclusión en cuanto al hardware un microcontrolador es mucho mejor en relación a un microprocesador, pues este tiene mayor almacenamiento de memoria RAM, ROM, periféricos integrados en un solo chip y mayor velocidad de procesamiento.

1.6.5.2 Aplicaciones

Tomando en consideración lo expresado por (Santamaría, 2011), se pueden citar las siguientes diferencias:

Microprocesadores

- Son la base para la construcción de sistemas de propósito general, sus tareas son inespecíficas como el desarrollo de software, juegos, páginas web, edición de fotos, creación de documentos, etc.
- La relación entre la entrada y la salida no está definida.
- Necesitan gran cantidad de recursos, como los puertos de RAM, ROM, E / S, etc.
- La velocidad de reloj del microprocesador es bastante alta en comparación con el microcontrolador; considerando que los microcontroladores operan desde unos pocos MHz a 30 a 50 MHz, el microprocesador de hoy opera por encima de 1 GHz a medida que realizan tareas complejas.

Microcontroladores

- Están diseñados para realizar tareas específicas, es decir puede efectuar aplicaciones donde se define la relación de entrada y salida, por ejemplo:

teclado, mouse, lavadora, cámara digital, pendrive, control remoto, microondas, automóviles, bicicletas, teléfonos, móviles, relojes, etc.

- Necesita pocos recursos como RAM, ROM, puertos I / O etc., y por lo tanto se pueden incrustar en un solo chip, lo que a su vez reduce el tamaño y el costo.

En conclusión en cuanto a las aplicaciones un microcontrolador es mucho mejor en relación a un microprocesador, pues sus tareas son específicas y se necesita menos recursos para realizarlas lo que minimiza los costos.

1.6.5.3 Costo

Tomando como base lo expresado por (Santamaría, 2011) se puede decir lo siguiente:

La comparación de microcontroladores y microprocesadores en términos de costo no se justifica, pues sin lugar a dudas un microcontrolador es mucho más barato que un microprocesador.

Sin embargo un microcontrolador no se puede utilizar en lugar de un microprocesador y su uso no se aconseja en lugar de un microcontrolador, ya que hace que una aplicación sea bastante costosa.

El microprocesador no se puede utilizar de forma independiente sino que se necesitan otros periféricos como RAM, ROM, tampón, puertos I / O, etc., por lo tanto, un sistema diseñado alrededor de un microprocesador es bastante costoso.

En conclusión un microcontrolador es mucho mejor en relación a un microprocesador, pues el costo de su aplicación es menor porque este tiene incluidos periféricos de entrada y salida, por lo que no se necesita realizar gastos adicionales.

1.6.6 Organización de una computadora

Para (Ramirez, 2013), las secciones que componen un sistema microprogramable basado en CPU, son los siguientes:

- Reloj
- Memoria
- Buses del sistema
- Unidad de entrada y salida
- CPU, el mismo que está compuesto por: unidad aritmética- lógica (ALU); Acumuladores y Registros; y Unidad de Control (UC)

En conclusión un sistema microprogramable está dividido en cinco circuitos, dentro de los cuales el CPU está dividido en tres unidades.

Para mejor comprensión a continuación se muestra en un gráfico de todos los elementos descritos anteriormente:

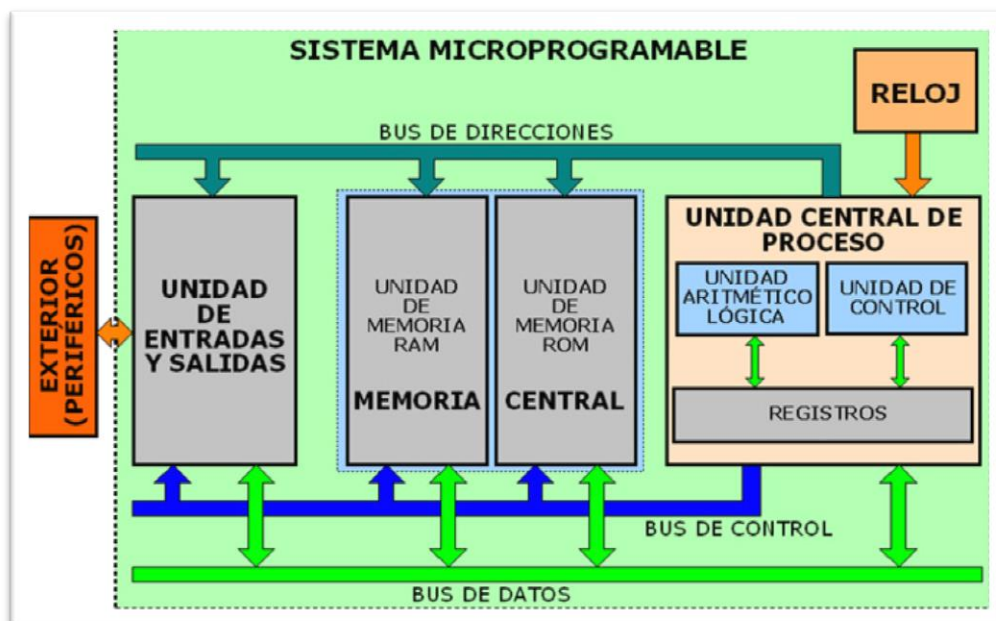


Ilustración 1.4: Arquitectura de una computadora
Fuente: Introducción a los Microcontroladores Equipo y Sistemas

1.6.7 Organización de un Microcontrolador

Tomando como referencia a (Orduña, 2012) se puede decir que:

La estructura del Microcontrolador PIC, se fundamenta en arquitectura Harvard, la cual consiste en registros con memoria y buses separados para las instrucciones y los datos

Entre las características importantes de esta arquitectura se pueden mencionar:

- Se emplean dos espacios de memoria: uno para datos y otro para programas
- Se utilizan dos buses distintos: uno para el tráfico entre la CPU y los datos, y otro para la comunicación entre la memoria de programa y la CPU.

3. Ejecutarla.

Cabe indicar que, cada instrucción cuenta con un código de operación llamado opcode y uno o más operandos, sobre los que se realizará la operación. Estas instrucciones por lo general incluyen:

- Transferencia de datos.
- Bifurcaciones (condicionales o incondicionales)
- Aritméticas: suma, resta, producto, división, etc.
- Lógicas: AND, OR, NOT, etc.

En conclusión la unidad central de procesos es aquella que administra y pone en marcha las actividades en el sistema, su tarea consiste en atrapar la información, decodificarla y ejecutarla, a través de un código de operación denominado “opcode”.

1.6.7.2 Tipos de CPUs.

Complex Instruction Set Computers (CISC).- Para (Rossano, 2012) es un procesador donde cada instrucción puede realizar varias operaciones de bajo nivel, tales como acceso a la memoria, operaciones aritméticas o cálculos de dirección. El término se originó en contraste con la reducción del conjunto de instrucciones del ordenador.

En conclusión el CISC, realiza instrucciones reducidas o de bajo nivel, es decir realiza operaciones simples y que no tienen ninguna complicación.

Reduced Instruction Set Computers (RISC).- Es un microprocesador que está diseñado para realizar un número menor de tipos de instrucciones de ordenador, para

que pueda funcionar a una velocidad más alta (realizar millones de instrucciones por segundo, o MIPS). Puesto que cada tipo de instrucción que una computadora debe realizar requiere transistores y circuitos adicionales, una lista más grande o un conjunto de instrucciones de ordenador tiende a hacer que el microprocesador más complicado y más lento en funcionamiento. (Palacios, 2013)

En conclusión el RISC, es capaz de realizar muchas instrucciones por segundo, por lo que posee mayor número de transistores y circuitos.

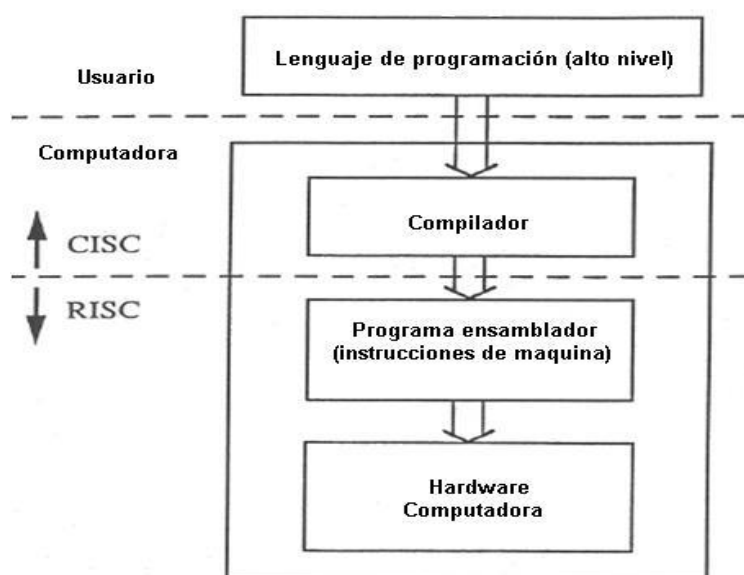


Ilustración 1.6: Vista Simplificada CPU
Fuente: Microcontrolador PIC

1.6.7.3 Tareas de la CPU

Según (Stefan, 2010) las tareas que realiza la CPU son las siguientes:

- 1. Obtención de la instrucción:** La instrucción se descarga de la dirección de memoria que se almacena en el contador de programa (PC), y se almacena en el registro de instrucción (IR). Al final de la operación de búsqueda, los

puntos de PC pasan a la siguiente instrucción, que será leído en el siguiente ciclo.

2. Decodificar la instrucción: El decodificador interpreta la instrucción, durante este ciclo de la instrucción dentro de la IR (registro de instrucciones) se decodifica.

3. Leer la dirección efectiva: En caso de una instrucción de la memoria (directa o indirecta), la fase de ejecución será en el siguiente pulso de reloj; si la instrucción tiene una dirección indirecta, la dirección efectiva se lee de la memoria principal y luego se coloca en los registros de datos (Pulso Reloj: T3). Si la instrucción es directa, no se hace nada en este pulso de reloj.

4. Ejecución: La unidad de control de la CPU, pasa la información decodificada como una secuencia de señales de control, a las unidades funcionales pertinentes de la CPU, para realizar las acciones requeridas por la instrucción, tales como: valores de lectura de los registros, pasarlos a la ALU (Arithmetic Logic Unit) para realizar funciones matemáticas o lógicas en ellos, y por escrito el resultado de nuevo a un registro. Si la ALU está involucrado, envía una señal de condición de nuevo a la CU, el resultado generado por la operación se almacena en la memoria principal o es enviada a un dispositivo de salida.

En conclusión las tareas que realiza la CPU son cuatro: obtiene la instrucción, la decodifica es decir la interpreta, lee la dirección y la ejecuta es decir pasa la

información decodificada como una secuencia de señales de control, a las unidades funcionales pertinentes de la CPU.

1.6.7.4 Sistema de Memoria

Según (Angulo, 2010), existen dos modelos o Arquitecturas de acuerdo con la organización de la memoria:

Arquitectura Harvard.

Esta arquitectura, utiliza memoria separadas para las instrucciones y datos, tiene sus propios buses de dirección, control y datos; la memoria de programas es solo de lectura y la de datos se puede leer y escribir. Esta se muestra a continuación:

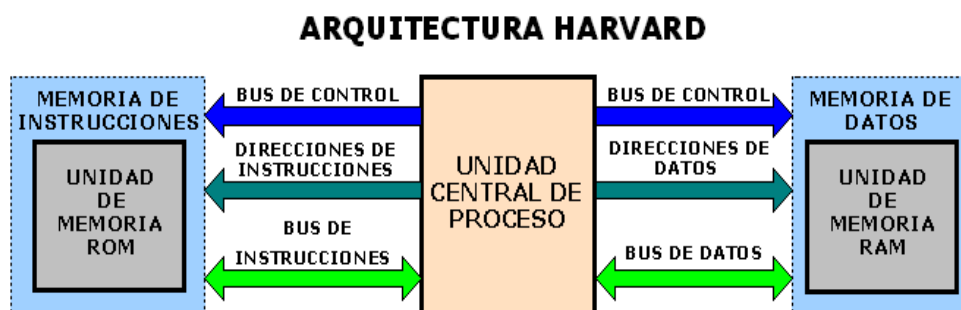


Ilustración 1.7: Arquitectura Harvard
Fuente: Microcontroladores “PIC”: diseño práctico de aplicaciones

Arquitectura Von Neuman.

Esta arquitectura conecta la CPU con la memoria; tiene un conjunto de direcciones y buses de datos para leer datos desde memoria y escribir datos en

la misma, y otro conjunto de direcciones y buses de datos para ir a buscar instrucciones. A continuación se muestra esta:

ARQUITECTURA VON NEUMANN

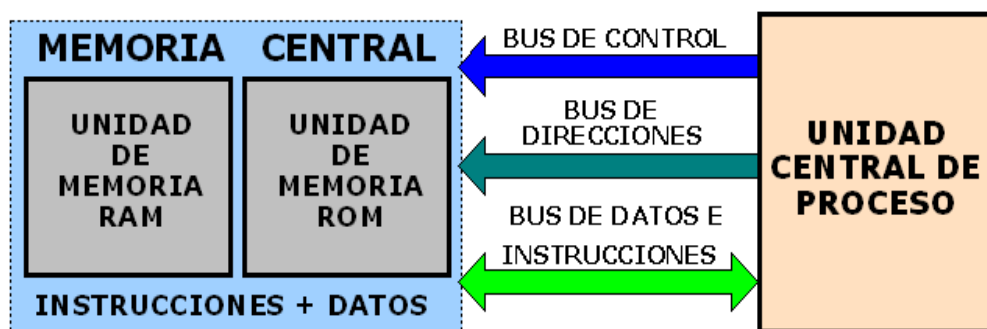


Ilustración 1.8: Arquitectura Von Neuman
Fuente: Microcontroladores “PIC”: diseño práctico de aplicaciones

En conclusión la arquitectura Harvard utiliza memorias y buses separadas para datos y programa, la memoria para datos puede leer y escribir y la de programa únicamente puede leer. La arquitectura Von Neuman tiene un conjunto de buses y direcciones para leer y escribir y otro conjuntos de buses y direcciones para ir a buscar instrucciones.

1.6.7.5 Periféricos incorporados a un Microcontrolador

Para (López, 2011) la mayoría de los microcontroladores contienen una serie de módulos de hardware, entre los dispositivos típicos se pueden mencionar los siguientes:

Módulo temporizador

Un temporizador de hardware, permite al microcontrolador contar el número de impulsos durante una unidad de tiempo para determinar la velocidad de rotación; la mayoría de los microcontroladores tienen al menos un módulo temporizador, normalmente con un número de entradas y salidas.

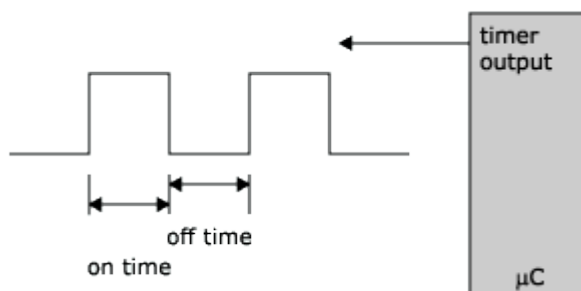


Ilustración 1.9: Módulo Temporizador
Fuente: Microcontroladores PIC: Sistema integrado para el autoaprendizaje

En conclusión se puede decir que módulo temporizador permite determinar la velocidad de rotación a través del conteo de los impulsos en una unidad de tiempo

Serial Port

Una interfaz serie es un puerto que permite que los datos de un microcontrolador sean transmitidos a otro microcontrolador, PC o un sistema remoto a través de un par de hilos. Esto puede ser una forma muy conveniente de la transmisión de datos entre dos dispositivos. Una desventaja puede ser la velocidad lenta en la transferencia de datos.

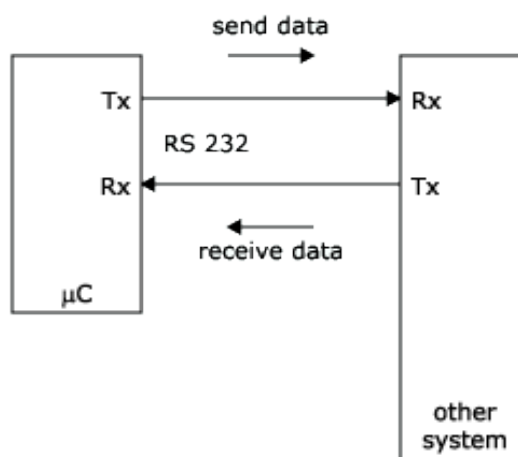


Ilustración 1.10: Serial Port

Fuente: Microcontroladores PIC: Sistema integrado para el autoaprendizaje

En conclusión un serial port es un dispositivo que permite transmitir los datos de un microcontrolador a otro, y tiene como principal desventaja el tiempo que tarda en realizar esta operación.

Conversión analógica a digital

En el mundo real las señales son a menudo análogo en la naturaleza, por ejemplo, se podría necesitar monitorizar señales (log) analógicas de un medidor de tensión, o de un transductor de conversión de temperatura en una señal analógica eléctrica (transductor de temperatura).

Desde microcontroladores a veces se requieren para este tipo de aplicaciones, algunas están equipadas con un convertidor analógico al digital (ADC).

En la mayoría de los casos, el ADC es del tipo de aproximación sucesiva, y hay un número de canales analógicos (entre 4 y 8 es más común); los canales de entrada analógica son generalmente compartidas con uno o más de los puertos de E / S digital, es decir que son multiplexados con los puertos

digitales y la decisión para seleccionar si los pasadores de chips se van a utilizar para las entradas de canal analógico o puertos digitales se realiza en el software (durante la configuración inicial del programa de rearme)

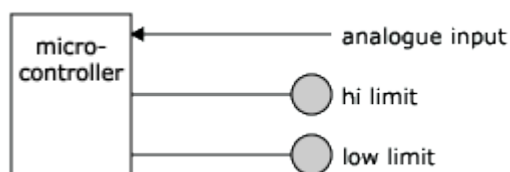


Ilustración 1.11: Conversión analógica a digital

Fuente: Microcontroladores PIC: Sistema integrado para el autoaprendizaje

En conclusión la conversión consiste en la transcripción de los datos análogos a digitales con la finalidad de facilitar su procesamiento, codificación y comprensión.

1.6.8 Familias populares de Microcontroladores

Según (Orduña, 2012) los microcontroladores se clasifican en tres grupos:

Gama Baja: Microcontroladores con instrucciones de 12 bits

Estos disponen de un repertorio de 33 instrucciones, la memoria de programa tiene una capacidad de 2.048 palabras y está organizada en páginas de 512 palabras cada una, la memoria de datos está formada por registros de 8 bits, organizada en bancos de hasta 32 registros.

Gama Media: Microcontroladores con instrucciones de 14 bits

Estos tienen un repertorio de 35 instrucciones, la memoria de programa tiene una capacidad de 8.192 palabras y está organizada en páginas de 2.048 palabras cada una, la memoria de datos está formada por registros de 8 bits, organizada en bancos de hasta 120 registros, con un máximo de cuatro bancos

Gama Alta: Microcontroladores con instrucciones de 16 bits

Estos se distinguen por sus instrucciones de 16 bits, posee mayor profundidad en la pila, un sistema de interrupciones más elaborado, posea los dispositivos integrados para las interrupciones internas y varias entradas para las interrupciones externas, tienen una arquitectura abierta, lo que permite la aplicación de memorias de programas y de datos

A continuación se muestra una tabla resumen de la familia de Microcontroladores:

Tabla 1.1: Tabla Resumen de las Familias de Microcontroladores

Familia Microcontroladores	Gama			Rasgo Definitivo
	Baja	Media	Alta	
PIC10	x			6 terminales
PIC12X5	x			8 terminales
PIC12 (excepto PIC12X5)		x		8 terminales
PIC16X5	x			-
PIC16 (excepto PIC16X5)		x		-
PIC17			x	-
PIC18			x	Gama alta mejorada

Fuente: Arquitectura y Programación de Microprocesadores

En conclusión existen tres familias populares de microcontroladores: Gama Alta, Media y Baja, clasificadas según el número de bits, capacidad de instrucciones y memoria.

1.6.9 Lenguajes de programación

Según (Perez, Menendez, Fernandez, & López, 2010) se puede decir que el lenguaje de programación es:

Un grupo de instrucciones que permiten actuar sobre los bits de combinaciones binarias.

Existen algunos lenguajes de programación de bajo nivel como ensamblador o de alto nivel como lenguaje C que es uno de lenguajes más comunes hoy en día en muchas tareas.

1.6.9.1 Lenguaje Máquina/ensamblador

Para (León, 2011) EL lenguaje ensamblador es un lenguaje de bajo nivel que se usa para programar microprocesadores, microcontroladores, y circuito integrados.

En conclusión el lenguaje máquina/ ensamblador expresa información de manera entendible, pues trabaja con caracteres alfanuméricos los cuales son traducidos por un microcontrolador para ponerlos en marcha.

1.6.9.2 Compiladores

Para (García, 2012) el compilador transforma o convierte el lenguaje en que está hecho el programa a otro que pueda entender la máquina, en el mismo entorno de trabajo que está hecho, por ejemplo el PC y puede ser transferido al PIC mediante cualquier programador.

En conclusión el compilador toma un texto escrito en lenguaje de alto nivel y lo traduce a un lenguaje comprensible.

1.6.9.3 Simuladores

Para (León, 2011) los simuladores permiten ver como el programa se podrá ejecutar en la realidad sin salir del PC, permitiendo ver paso a paso la ejecución y hacer correcciones del contenido del mismo.

En conclusión el lenguaje simulador permite una adecuada interacción entre la edición, depuración y ejecución.

1.6.10 Herramientas de desarrollo

La herramienta de software que se utilizara para el desarrollo del trabajo de investigación es Altium Designer que es un sistema completo para el diseño electrónico en todas sus fases y para todas las disciplinas, ya sean esquemas, simulación, diseño de circuitos impresos, edición de circuitos impresos, o desarrollo de código para microprocesadores, con potentes visores y mensajes. Se utilizó este programa porque es el más utilizado por los estudiantes de universidades de reconocimiento mundial pues es práctico y completo al momento de realizar este tipo de trabajos.

1.6.11 Elementos de desarrollo.

Como elementos principales de la placa entrenadora que se va a construir se tiene que destacar que se va a utilizar dos tipos: SMD y THT para (Benchimol, 2011) SMD (Surface-Mount-Device) o dispositivos de montaje superficial, son componentes más pequeños que los normales y se sueldan superficialmente, una de las ventajas principales son su tamaño, ya que ahorran mucho espacio a comparación

de los THT (Trough-Hole-Tecnology) o tecnología de hueco pasante, que su principal ventaja es que pueden manejar altas potencial, y la desventaja es su tamaño grande.

Tecnología SMD



Tecnología THT



Ilustración 1.12: Tecnología SMD y THT
Fuente: Arquitectura y Programación de Microprocesadores

1.6.11.1 PIC

Según (Reyes, 2011)

En el interior de PIC (Peripheral interface Controller) existe la arquitectura de un computador, esto quiere decir que posee, memoria, circuitos de entrada salida y sobre todo el CPU.

Para poder utilizarlo se debe programarlo, para realizar tareas como simplemente prender luces, hasta tenerlo ejecutando varias tareas dentro de un robot.

Los PIC, que se van a utilizar son de la familia **MICROCHIP Technology INC.** Ya que existen varias más, en especial el 16f877 ya que es uno de los más completos.

Una de las principales ventajas es el bajo consumo, bajo costo y acceso a estos.

(Ver Anexo 1: PIC16F877)

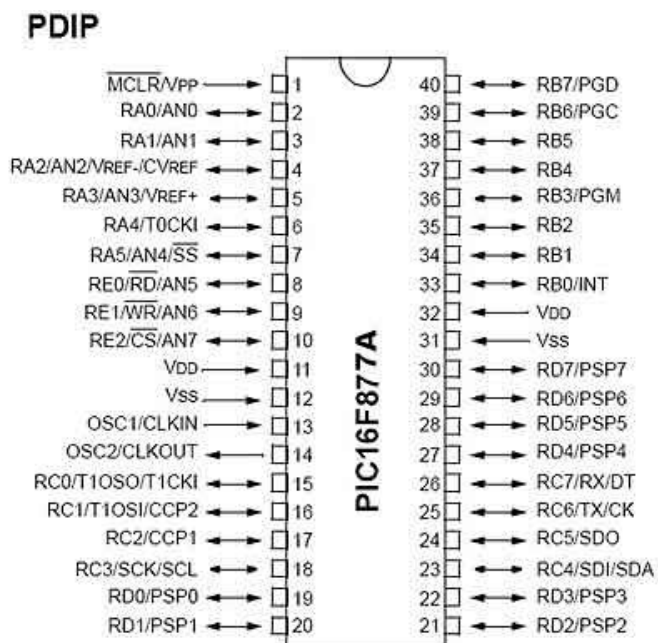


Ilustración 1.13: PIC

Fuente: Microcontroladores PIC programación en Basic

1.6.11.2 Diodos Led (smd)

Para (Reyes, 2011), este dispositivo sirve para emitir luz con poco consumo de energía, es de un tamaño pequeño y de gran utilidad en infinitos campos.

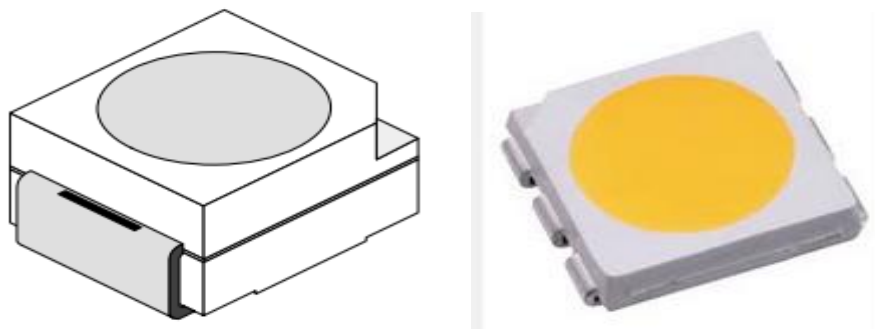


Ilustración 1.14: Diodos Led

Fuente: Microcontroladores PIC programación en Basic

1.6.11.3 Pulsador (hueco pasante)

Para (Benchimol, 2011), son dispositivos que al pulsar o soltar un botón permiten o no el paso de corriente.

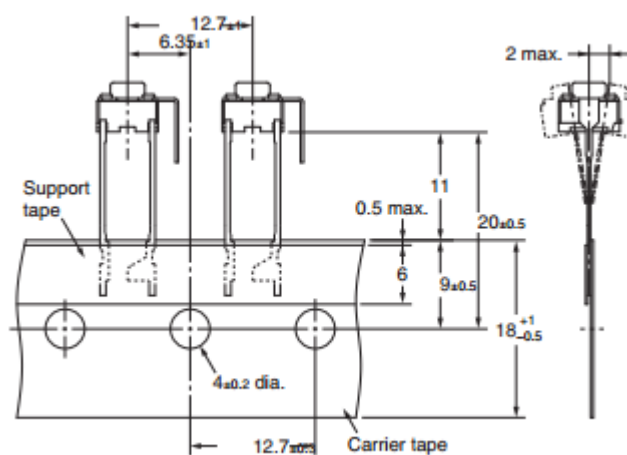


Ilustración 1.15: Pulsador
Fuente: Electrónica Práctica

1.6.11.4 LCD 16x2 (hueco pasante)

Para (Reyes, 2011) los Visualizadores de cristal líquido (LCD), son pantallas electrónicas que permiten ver caracteres sean letras o números. 16x2 significa que puede mostrar 16 caracteres por 2 líneas, es un elemento económico y programable.

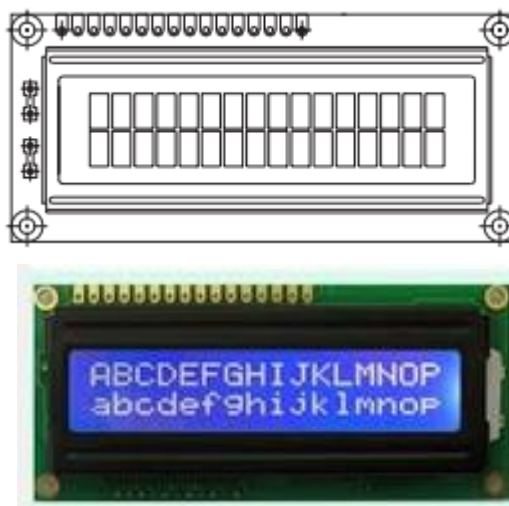


Ilustración 1.16: LCD 16X2
Fuente: Controladores PIC programación en Basic

1.6.11.5 Dip Switch 4 (hueco pasante)

Según (Reyes, 2011) este dispositivo funciona como un interruptor eléctrico en conjunto, que sirve para permitir o negar el paso de corriente según la posición.

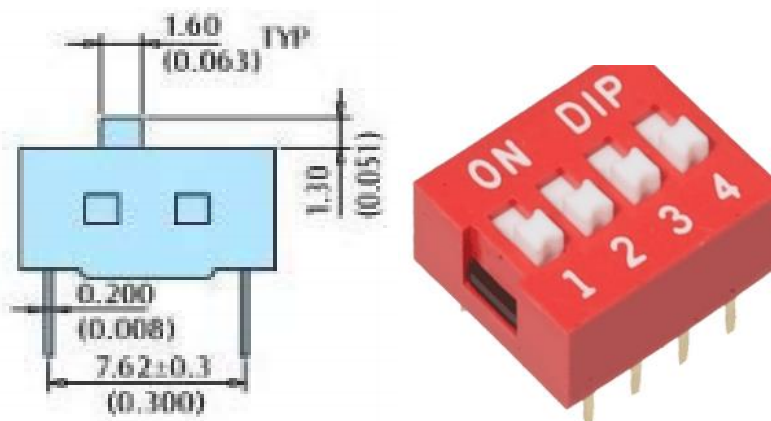


Ilustración 1.17: Dip Switch 4
Fuente: Microcontroladores PIC programación en Basic

1.6.11.6 Relé (Hueco pasante)

Para (Reyes, 2011) este dispositivo funciona como un interruptor electromecánico, que por motivo de una bobina y un electroimán se acciona permitiendo conectar circuitos eléctricos independientes de más voltaje, por ejemplo un foco externo.

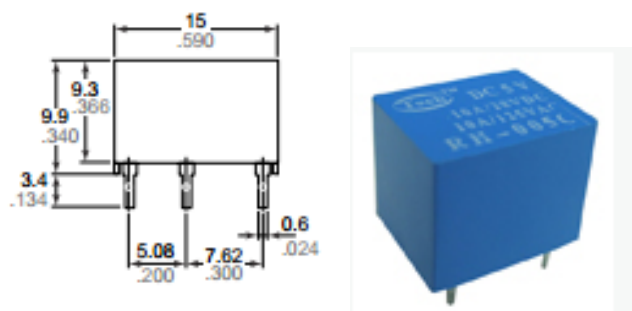


Ilustración 1.18: Relé

Fuente: Microcontroladores PIC programación en Basic

1.6.11.7 Conector DB9 Hembra (hueco pasante)

Tomando como referencia a (Reyes, 2011) se puede decir que este dispositivo permite conexiones en serie.

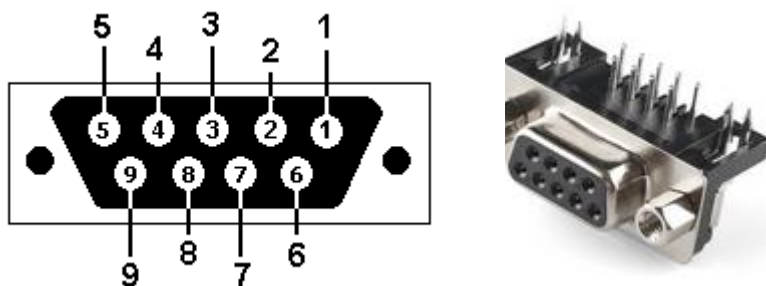


Ilustración 1.19: Conector DB9 Hembra

Fuente: Microcontroladores PIC programación en Basic

1.6.11.8 Display Ánodo Común (hueco Pasante)

Para (Benchimol, 2011), este dispositivo es un visualizador de 7 segmentos, quiere decir que se puede mostrar números al encender sus segmentos de manera que el usuario requiera.

Los displays de tipo ánodo común, quieren decir que existe una conexión interna en común de los leds de cada segmento que se unen por la parte positiva.

También existen los de cátodo común, que quiere decir que la conexión interna en común de los leds de cada segmento, es por la parte negativa.

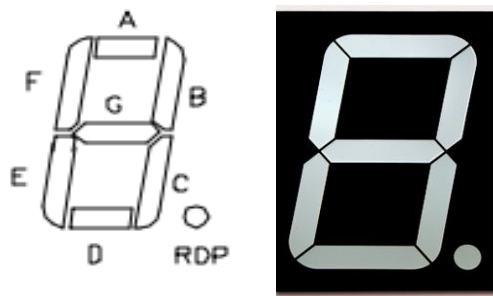


Ilustración 1.20: Display Ánodo Común
Fuente: Electrónica Práctica

1.6.11.9 Teclado 4x4 (Hueco pasante)

Tomando como referencia a (Reyes, 2011), este dispositivo es de entrada manual, de gran utilidad con el microcontrolador PIC, su funcionamiento es como pulsadores en forma de matriz.

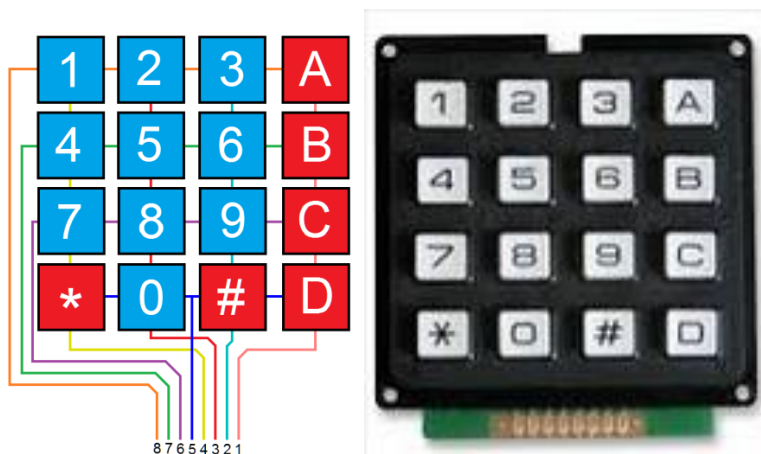


Ilustración 1.21: Teclado 4x4
Fuente: Microcontroladores PIC programación en Basic

1.6.11.10 Zócalo (hueco pasante)

Según (Benchimol, 2011), es un soporte para el PIC, en donde se va a hacer las conexiones con el resto del circuito. El tamaño de pines depende del tipo de PIC que se utilizara, en este caso de 40, 18 y 16 pines.



Ilustración 1.22: Zócalo
Fuente: Electrónica Práctica

1.6.11.11 Conector tipo Jack (hueco pasante)

Según (Reyes, 2011) este dispositivo es un conector, que permitirá la alimentación de energía eléctrica a la placa, desde una fuente externa



Ilustración 1.23: Conector tipo Jack
Fuente: Microcontroladores PIC programación en Basic

1.6.11.12 Cristal 4MHZ (hueco pasante)

Para (Sánchez, 2011) este dispositivo actúa como un resonador sintonizado a cierta frecuencia según dicho elemento, sirve como oscilador que produce una señal repetitiva, el proyecto QUE SE ESTA DESARROLLANDO utilizara cristal de 4MHZ y 10MHZ.



Ilustración 1.24: Cristal 4MHZ
Fuente: Construcción de Microprocesadores

1.6.11.13 Capacitor Cerámico 22pf (hueco pasante)

Para (Sánchez, 2011), este dispositivo es utilizado gran parte en circuitos de alta frecuencia, sirve para mantener o almacenar una carga eléctrica de acuerdo al tamaño de su superficie.



Ilustración 1.25: Capacitor Cerámico 22 pf
Fuente: Construcción de Microprocesadores

1.6.11.14 Resistencia 330 ohmnios (smd)

Tomando como referencia a (Sánchez, 2011), este dispositivo ejerce oposición a la corriente que pasa a través de él. Se necesitará de 330, 4.7 k y 1k ohmnios. Anexo datasheet smd resistor.



Ilustración 1.26: Resistencia 330 ohmios
Fuente: Construcción de Microprocesadores

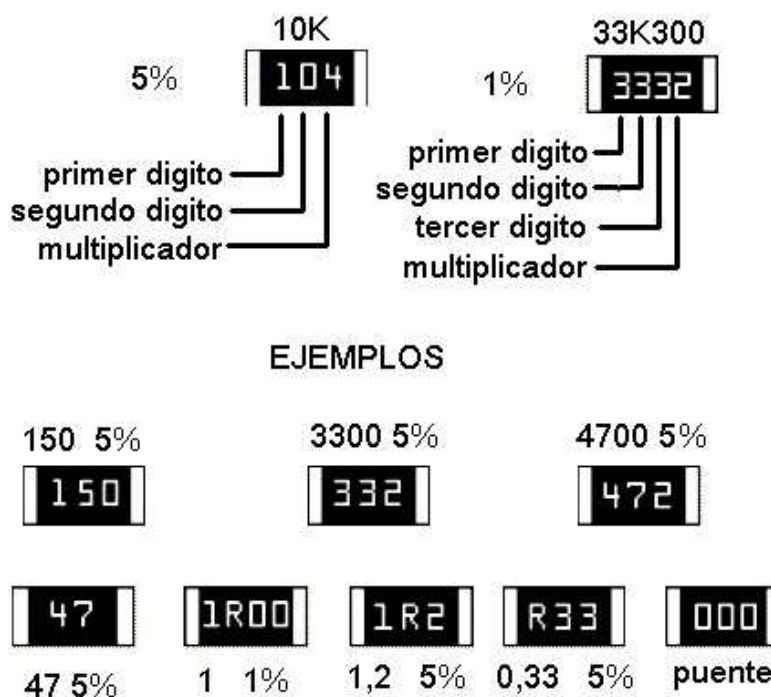


Ilustración 1.27: Ejemplo Resistencia 330 ohmios
Fuente: Construcción de Microprocesadores

1.6.11.15 Max232 (Hueco pasante)

Según (Sánchez, 2011), este circuito integrado es de maxim y convierte la señal de un SR-232 a señales que entienden los circuitos lógicos, sirve como interface para emisión y recepción de señales.

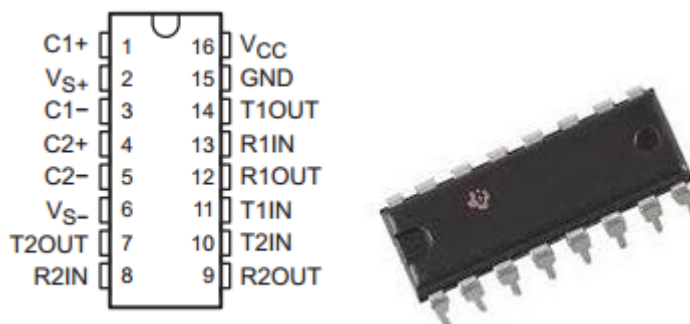


Ilustración 1.28: Max 232

Fuente: Construcción de Microprocesadores

1.6.11.16 Capacitor 0.1uf (smd 0805)

Para (Sánchez, 2011), este dispositivo es utilizado gran parte en circuitos de alta frecuencia, sirve para mantener o almacenar una carga eléctrica de acuerdo al tamaño de su superficie, este capacitor es de montaje superficial.

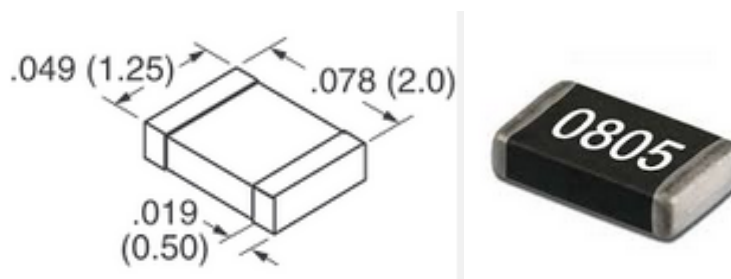


Ilustración 1.29: Capacitor 0.1uf

Fuente: Construcción de Microprocesadores

1.6.11.16 Transistor 2n3904 (huevo pasante)

Según (Sánchez, 2011), este dispositivo es semiconductor que sirve para transformar la señal en 0 o 1 y cierra o abre un circuito a amplifica una señal. Se utilizará el 2n3904 y 2n3906.

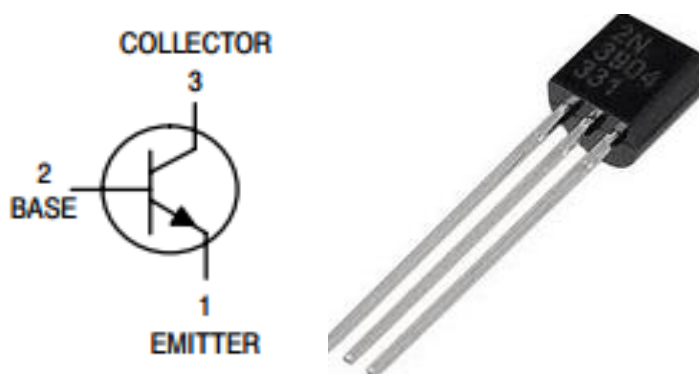


Ilustración 1.30: Transistor 2n3904
Fuente: Construcción de Microprocesadores

1.6.11.17 Bluetooth HC05

Para (Reyes, 2011), este dispositivo está diseñado para trabajar de manera inalámbrica con transmisión serial de datos.

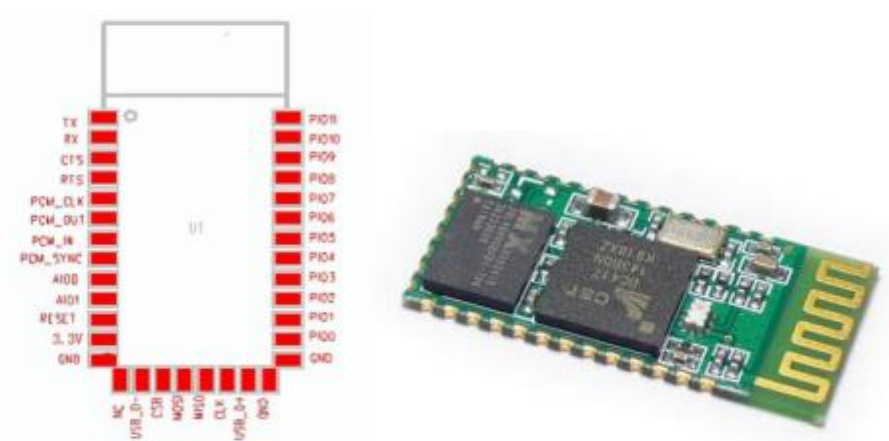


Ilustración 1.31: Bluetooth HC05
Fuente: Microcontroladores PIC programación en Basic

1.6.11.18 Programador PIC Kit 2

Según (Reyes, 2011), el programador para PIC de microchip, permite grabar información.



Ilustración 1.32: Programador

Fuente: Microcontroladores PIC programación en Basic

CAPÍTULO II

METODOLOGÍA

2.1 Metodología de Investigación

2.1.1 Método General

El método general que se utilizará para el desarrollo del presente trabajo será el descriptivo, pues a través de este se identificarán los elementos que intervienen en una placa basada en microcontroladores PIC, sus características, los mecanismos de funcionamiento, además se podrán identificar cada una de las piezas y dispositivos a usar, cuál será su funcionamiento y en qué lugar se deben adaptar.

2.1.2 Método Específico

Se usará la investigación bibliográfica, porque esta permite obtener información acertada, específica y confiable sobre los diferentes tópicos a tratarse en este proyecto tanto en software como en hardware. Además es necesario tener a disposición bibliografía de expertos en la rama, quienes generalmente comparten sus estudios e investigaciones en medios escritos.

2.2 Metodología de desarrollo

La metodología por la cual este proyecto se desarrollará, será la propuesta por el autor, tomando como base la Ingeniería Concurrente, que según el autor (Riba & Molina, 2006) sirve como referencia para empezar a desarrollar proyectos o productos.

- Ideación
- Desarrollo conceptual y básico
- Desarrollo avanzado
- Lanzamiento

Ideación

Esta fase determina los requerimientos del entrenador y las funciones que debe cumplir.

Desarrollo conceptual y básico

En esta etapa se desarrolla las alternativas de solución del producto dando como inicio al Diseño conceptual, el cual permite hacer un bosquejo y materialización del mismo, en el cual se pretende tener un orden u organización de los módulos a construir.

Para esto se escogerá un software adecuado a nuestro diseño, se analizará los módulos que va a tener el entrenador y se hará una lista de los componentes a utilizar.

Desarrollo avanzado

Esta fase involucra las actividades que son la base para la fabricación del producto.

A continuación las actividades que determinan como llegar a cumplir esta fase:

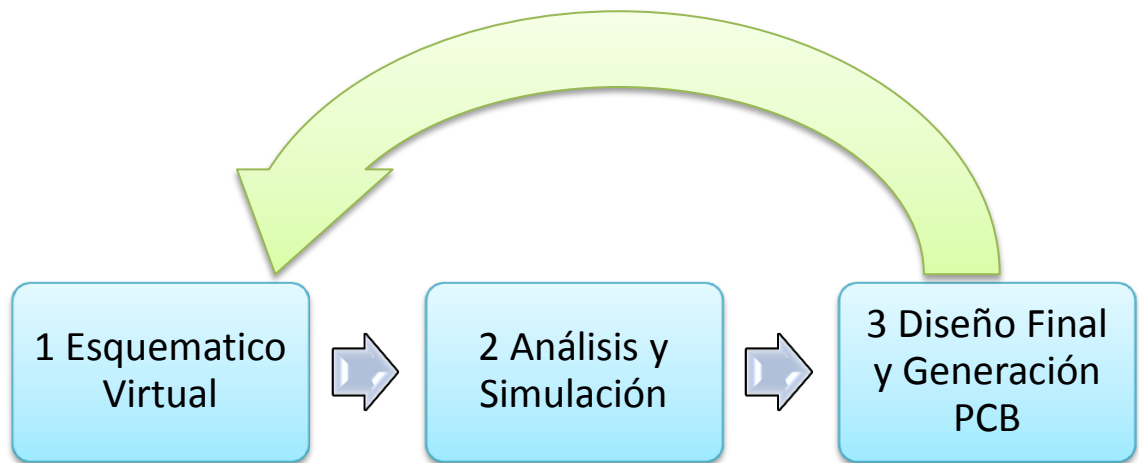


Ilustración 2.33: Fases para el Diseño del Circuito
Elaborado por: Paúl Toapanta

Fase 1 Esquemático Virtual:

En esta fase se escogerán los diagramas específicos de cada módulo que se ha elegido que tendrá el entrenador, se los implementara virtualmente en el software eligiendo los componentes correspondientes, los mismos esquemas se mostraran gráficamente para hacer uso de estos como guía gráfica.

Fase 2 Análisis y Simulación:

Esta fase es muy importante, ya que se podrá determinar los errores realizados con la ayuda de un simulador, este nos provee de información en tiempo real como si se estuviera puesto en marcha, analizando cuales son los errores más comunes cometidos, y dando la oportunidad de cambiar los componentes o el diseño en algún lugar del mismo.

Fase 3 Diseño Final Y Generación de Printed Circuit Board (PCB):

Aquí se aprobará la fase anterior dando como resultado el diseño final y se procederá a la creación de planos para la fabricación.

Lanzamiento

En esta fase se demuestra el proceso de producción del producto final así.

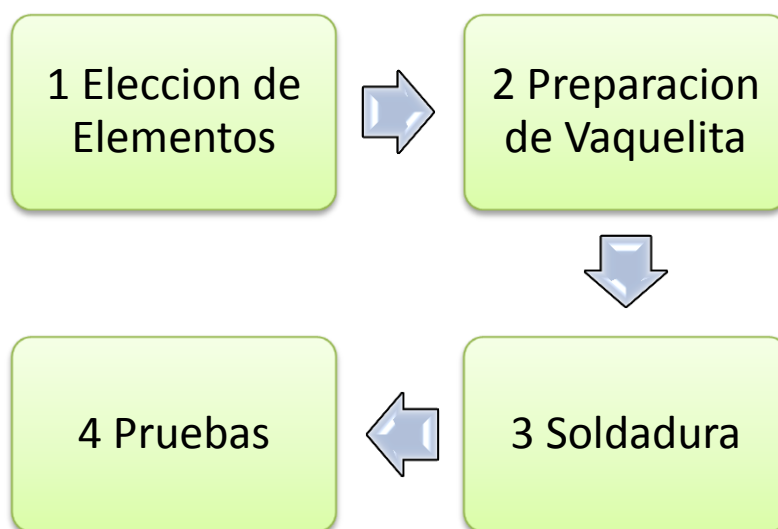


Ilustración 2.34: Fases para la Construcción del entrenador
Elaborado por: Paúl Toapanta

Fase 1 Elección de Elementos:

En esta fase se busca los elementos necesarios para la creación o fabricación de la placa electrónica ya que si alguno de los componentes no está disponible en el mercado, se tendrá que tomar correctivos en nuestro diseño, mas no por funcionalidad sino por disponibilidad de elementos.

Fase 2 Preparación de Baquelita:

Existen diferentes técnicas para la creación circuitos impresos de la baquelita, algunos de ellos dependen mucho de la complejidad del proyecto. Como los más conocidos tenemos:

- Impresión por tinta indeleble.
- Impresión por serigrafía
- Impresión térmica transferible

Todos estos tienen el mismo principio que es cubrir las pistas para que no se desgaste del cobre por acción del ácido.

Fase 3 Soldadura:

En esta fase se deberá colocar los elementos y dispositivos en lugar correspondiente ya predefinidos por el diseño, y mantenerlos ahí con soldadura de estaño.

Fase 4 Pruebas:

Aquí se deberá hacer pruebas correspondientes en cada pista para comprobar la continuidad de cada una y que todo esté bien soldado y en su lugar, para empezar a utilizarla.

CAPÍTULO III

RESULTADOS

El entrenador universal basado en microcontroladores PIC para el laboratorio de electrología de la Escuela de Ingeniería en Sistemas de la PUCESA, tiene la finalidad de ser una herramienta de apoyo académico que soporte múltiples aplicaciones entorno a la programación de microcontroladores, con los cuales los alumnos puedan desarrollar sus proyectos de robótica o electrónica.

Se fundamenta en los siguientes aspectos:

- **Factibilidad Técnica- Operativa:** existen los recursos económicos, materiales y tecnológicos para el desarrollo de la propuesta, los mismos que serán financiados por el autor.
- **Factibilidad Institucional:** se visualiza en la voluntad de las personas que están a cargo del Laboratorio de Electrología, para hacerse cargo del entrenador universal diseñado y construida por el autor del trabajo y darle el uso correspondiente.
- **Factibilidad Social:** El diseño y construcción del entrenador PIC, no afecta el desarrollo normal de las actividades realizadas por los estudiantes en el laboratorio de electrología, más bien pretende ser una ayuda en el tratamiento de los proyectos de los estudiantes.

Para el diseño y construcción del entrenador basado en microcontroladores PIC se ha determinado realizar las etapas que se muestran a continuación.

3.1 Diseño y Construcción del Entrenador

3.1.1 Fase 1: Ideación

Los datos obtenidos para determinar la necesidad por parte de estudiantes del laboratorio de Electrología de la escuela de Ingeniería en Sistemas de la Pucesa , son obtenidas por medio de una entrevista con el docente encargado de este laboratorio, el cual supo indicar que los estudiantes experimentan problemas al no tener acceso a los siguientes elementos:

- Programador universal PIC, para grabar la programación en el microcontrolador.
- Luces led, para probar puertos de salida.
- Display de 7 segmentos, para probar puertos de salida.
- Pantalla LCD, para mostrar información.
- Teclado 4x4, para ingresar datos.
- Comunicación Serial, intercambio de información.
- Comunicación Inalámbrica Bluetooth, intercambio de información.
- Relé, para encender elementos externos.
- Pulsadores, para ingreso de información.
- Potenciómetros, para regular la intensidad de voltaje.

Todos estos elementos están entre los más importantes y necesarios con este tipo de microcontroladores, los cuales servirían de apoyo al practicar la programación y pruebas sobre elementos electrónicos, y así poder realizar sus proyectos.

3.1.2 Desarrollo conceptual y básico

Uno de los aspectos más importantes a considerar en el diseño de una placa entrenadora para microcontroladores PIC de Microchip, son los tipos de elementos que se controlará con este dispositivo, teniendo como objetivo realizar los siguientes módulos de entrada y salida:

- Pantalla LCD
- Teclado matricial 4x4
- Fuente de Poder
- Comunicación Serial RS232
- Comunicación Inalambrica Bluetooth
- Rele de potencia
- Programador y lector PIC
- Pulsadores
- Diodos Led
- Display de 7 segmentos
- Interruptores
- Potensiómetros

Otro aspecto importante es cómo se va a realizar las conexiones entre el microcontrolador y los módulos, se tiene la opción de cablear desde cada pin del

microcontrolador a cada pin de los elementos al hacer prácticas, pero esta opción no es muy favorable ya que se debe consultar constantemente las fichas técnicas de los elementos para poder hacer las conexiones correctamente y los cables se pueden desprender de sus terminales con facilidad y eso dará lugar a confusión, la mejor opción es hacer conexiones internas desde el microcontrolador a los módulos y habilitarlos o deshabilitarlos por medio de interruptores. Para este proceso el software que se va a utilizar es Altium Designer, el cual contiene licencia de paga, pero ofrece una demostración gratis por 15 días con todas sus funciones y es así como se lo utilizó.

3.1.3 Desarrollo avanzado

Esta fase involucra las actividades que son la base para la fabricación del producto.

3.1.3.1 Fase 1: Esquemático Virtual

En esta fase se escogerán los diagramas específicos de cada módulo que se ha elegido que tendrá el entrenador, se los implementará virtualmente en el software eligiendo los componentes correspondientes, los mismos esquemas se mostrarán gráficamente para hacer uso de estos como guía gráfica.

En esta fase se indicará el desarrollo del circuito utilizando el software Altium Designer, en el cual crearemos el espacio de trabajo

1. Se abre el programa, posteriormente se selecciona la opción: File/New/Project y PCB Project así:

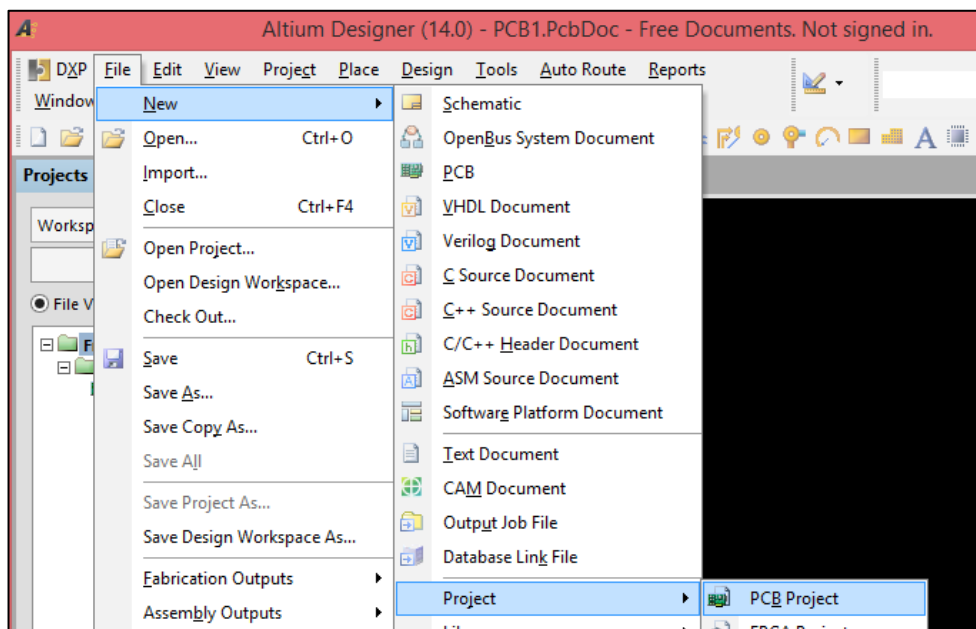


Ilustración 3.35: Selección de File/New/Project

Fuente: Altium Designer

2. Se crea un documento esquemático File/New/Schematic

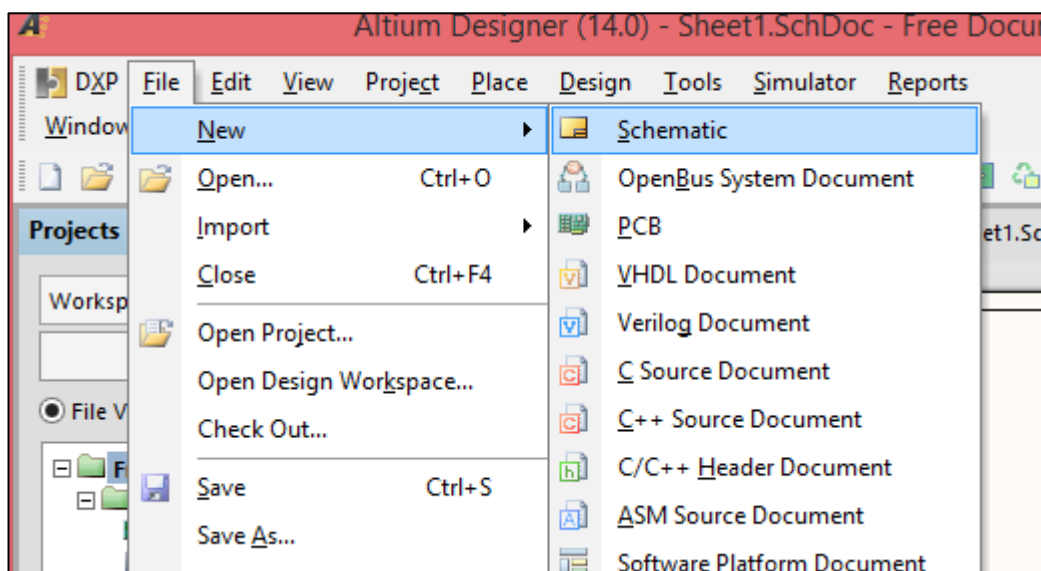


Ilustración 3.36: Selección de File/New/Schematic

Fuente: Altium Designer

3. Ahora se debe agregar librerías, dentro de View/workspace Panels/System/Libraries

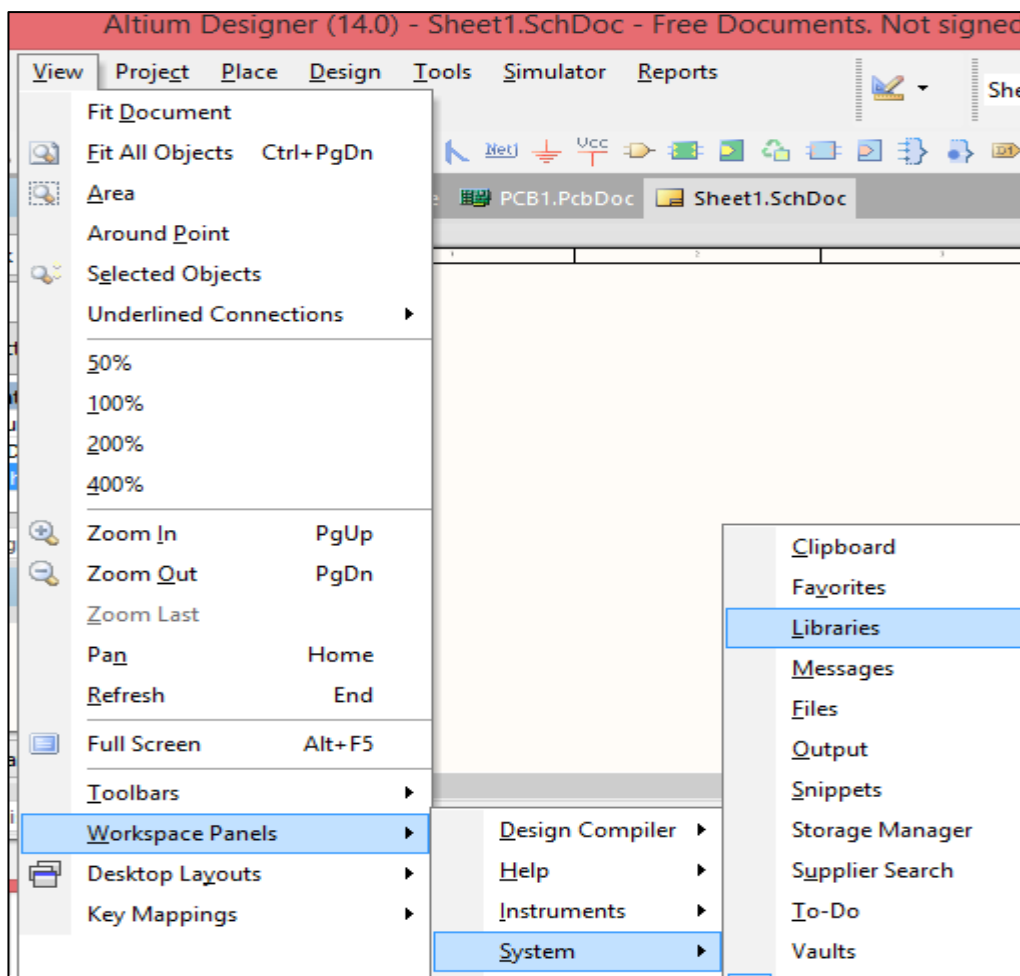


Ilustración 3.37: Selección para agregar Librerías
Fuente: Altium Designer

Aquí se debe buscar las librerías dentro de los directorios del programa, estos contienen los componentes de los cuales se seleccionará los necesarios, una vez elegido el componente se lo debe arrastrar al documento, en este caso una resistencia.

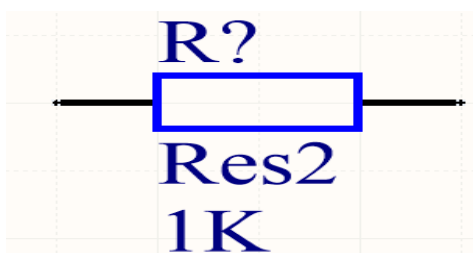


Ilustración 3.38: Documento Resistencia
Fuente: Altium Designer

El mismo procedimiento se lo realiza para todos los componentes de modulo.

Una vez seleccionados todos los componentes se los debe unir con el botón place wire

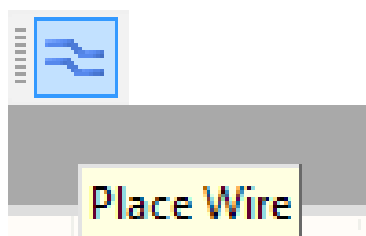


Ilustración 3.39: Botón Place wire
Fuente: Altium Designer

Para editar las propiedades de un elemento, se lo debe seleccionar y posteriormente dar doble click, aquí se abrirá una ventana en la cual se podrá editar los parámetros como: valores, tipo, comentarios etc.

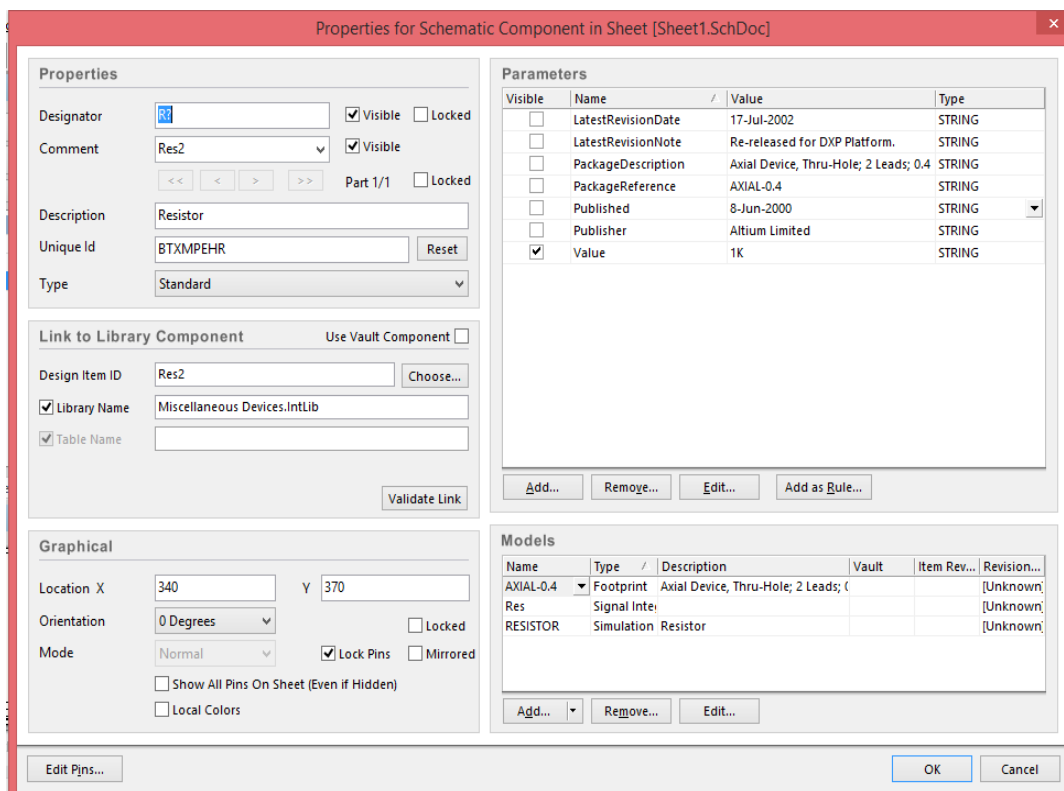


Ilustración 3.40: Cuadro de diálogo para editar propiedades de un elemento
Fuente: Altium Designer

3.1.3.1.1 Esquema de Módulos

El esquema se lo mostrará por módulos, porque al hacerlo de manera general existirá confusión de quien lo interprete, debido a que al existir muchas líneas no se entenderá adecuadamente, por tal razón se describe de manera individual indicando a que parte irá conectada tanto en el microcontrolador como en el periférico.

Diodos led

Es un periférico de salida que permite encender luces según las señales enviadas.

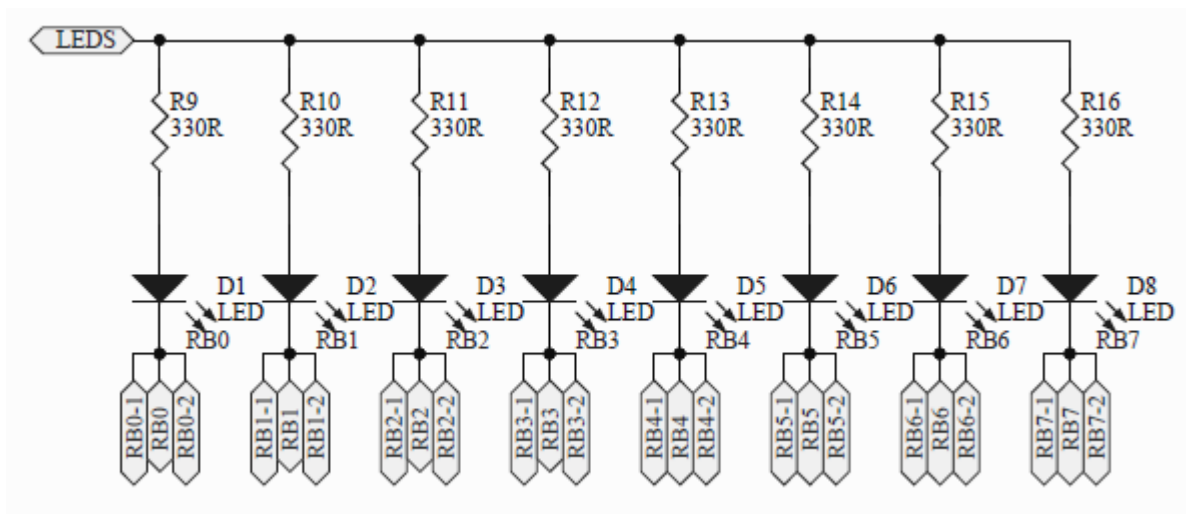


Ilustración 3.41: Esquema Diodos Led

Fuente: Altium Designer

LCD

Es un periférico de salida que permite visualizar números y letras según se programe en el microcontrolador.

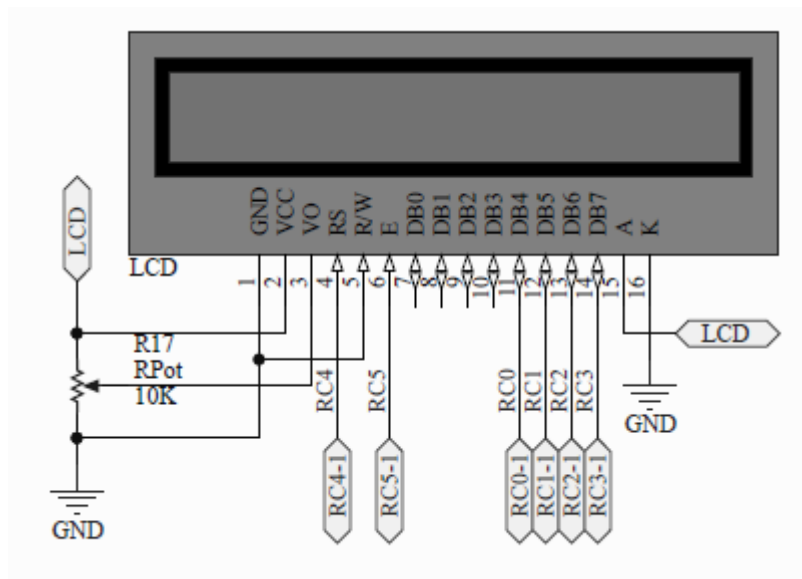


Ilustración 3.42: Esquema Diodos LCD

Fuente: Altium Designer

Teclado 4x4

Es un periférico de entrada que permite el ingreso de información sea en letras o números hacia el PIC.

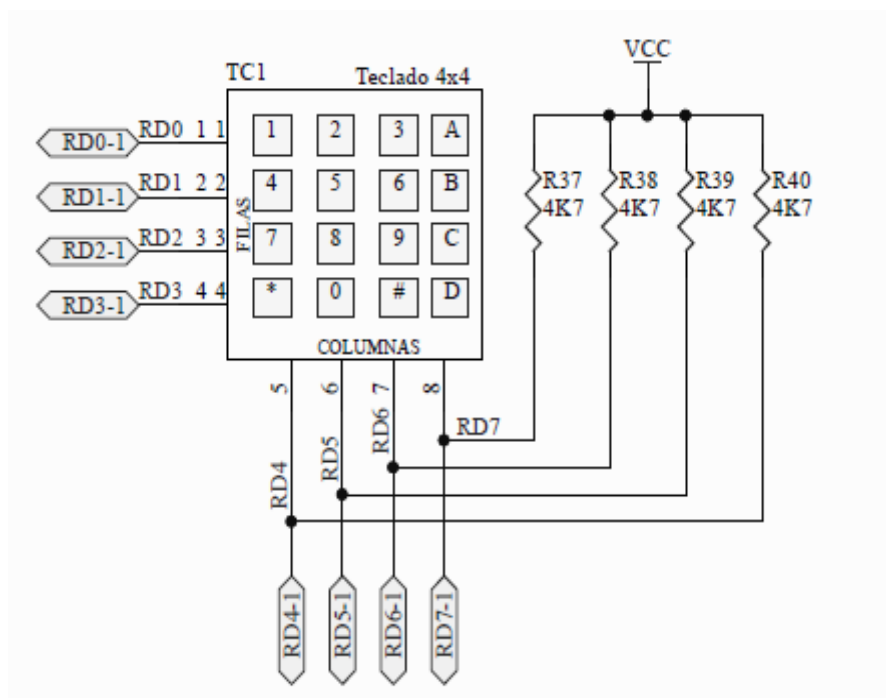


Ilustración 3.43: Esquema Teclado 4x4

Fuente: Altium Designer

Fuente de Poder

Se compone de 8 capacitores de 0.1uF, una resistencia de 330OR, una fuente de poder y un diodo led que muestra el estado del voltaje

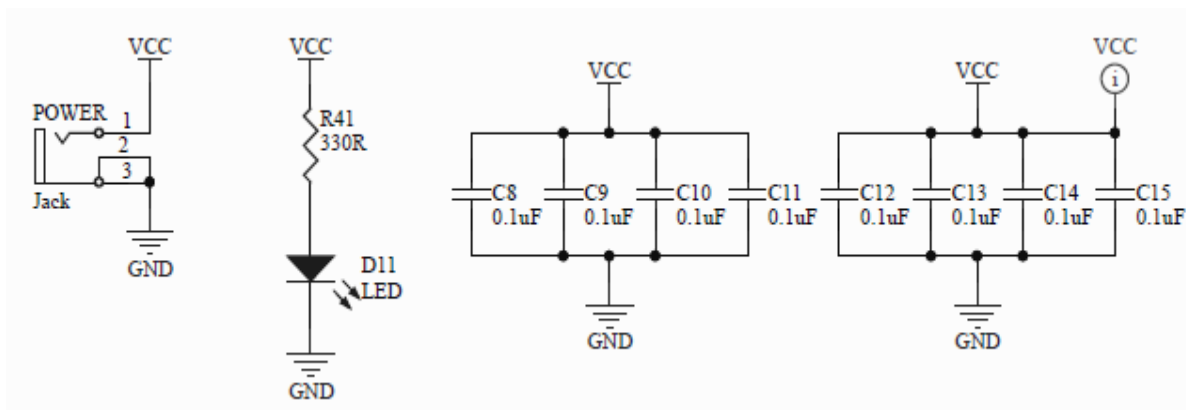


Ilustración 3.44: Esquema Fuente de Poder
Fuente: Altium Designer

Puerto Serial

Es un periférico de entrada y salida que permite envío y recepción de información desde el PIC hacia al computador u otro dispositivo externo y viceversa por medio de un cable.

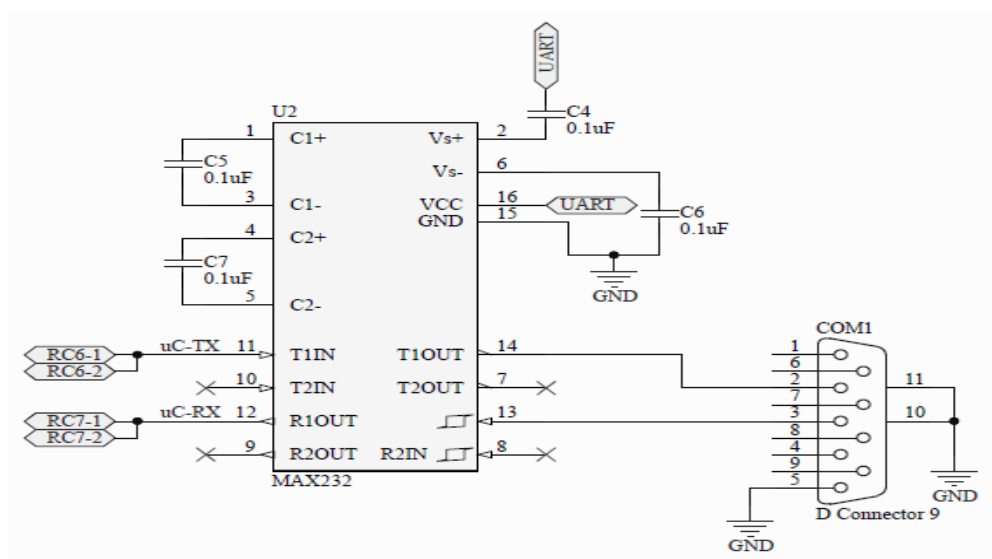


Ilustración 3.45: Esquema de RS232
Fuente: Altium Designer

Bluetooth

Es un periférico de entrada y salida que permite envío y recepción de información desde el PIC hacia al computador u otro dispositivo externo y viceversa, de manera inalámbrica. Utiliza el mismo puerto serial.

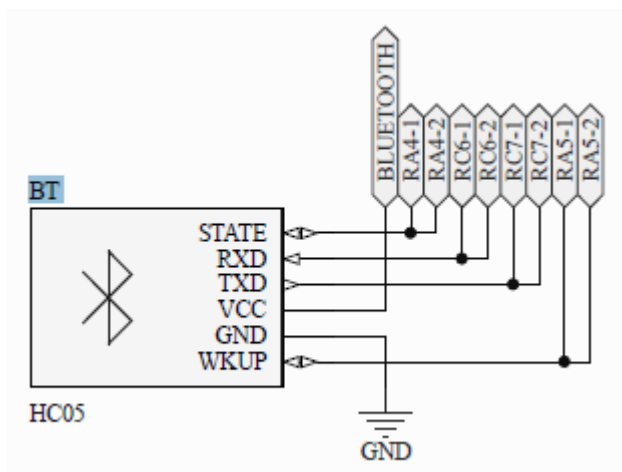


Ilustración 3.46: Esquema Bluetooth
Fuente: Altium Designer

Rele de potencia

Es un periférico de salida que permite encender dispositivos externos con un voltaje superior, ejemplo: lámpara de 110 v.

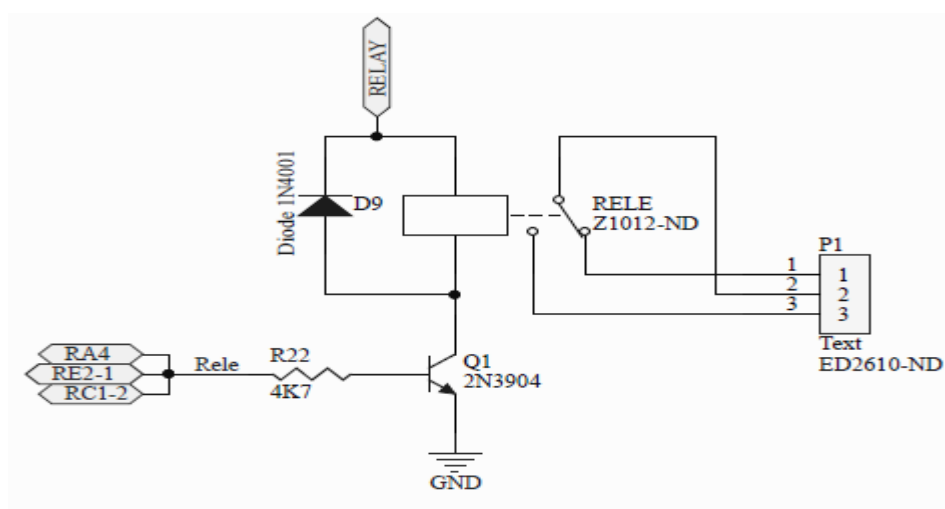


Ilustración 3.47: Esquema Rele de Potencia
Fuente: Altium Designer

Programador

Es un periférico de programación externa, denominado In Circuit Serial Programming (ICSP) que posibilita la programación del microcontrolador sin sacarlo del entrenador.

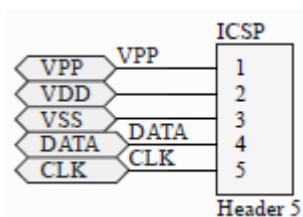


Ilustración 3.48: Esquema Programador
Fuente: Altium Designer

Pulsadores

Es un periférico de entrada y sirve para enviar una señal o señales al microcontrolador.

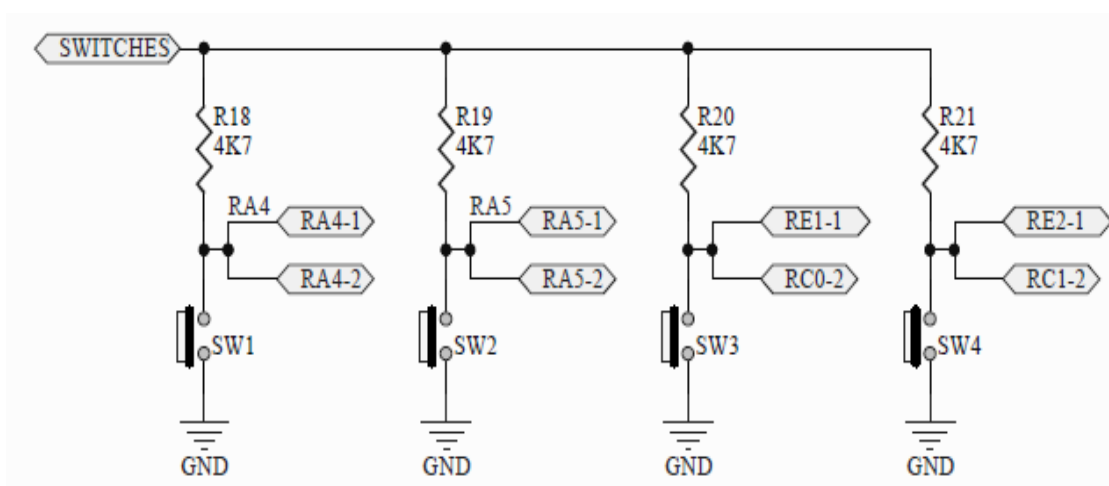


Ilustración 3.49: Esquema Pulsadores
Fuente: Altium Designer

Display 7 Segmentos

Es un periférico de salida, sirve para mostrar información al usuario.

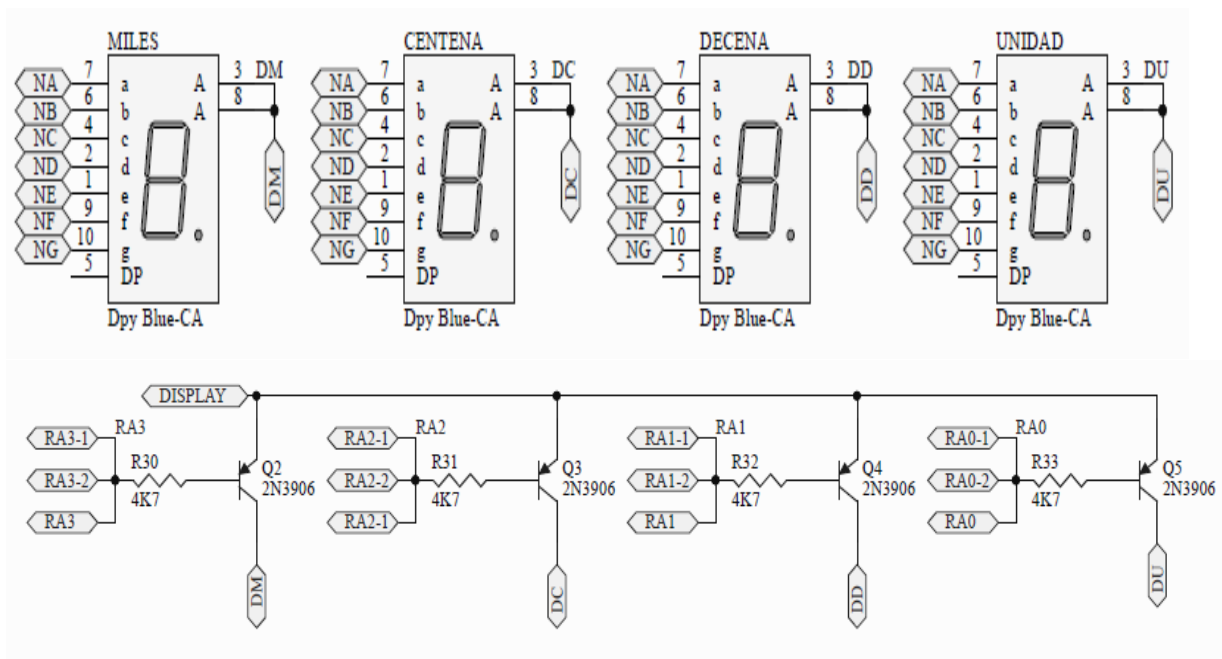


Ilustración 3.50: Esquema Display 7 segmentos
Fuente: Altium Designer

Dip Switch

Es un habilitador de los módulos a utilizar, puede estar en estado on y off, además está habilitando el estado análogo /digital del potenciómetro

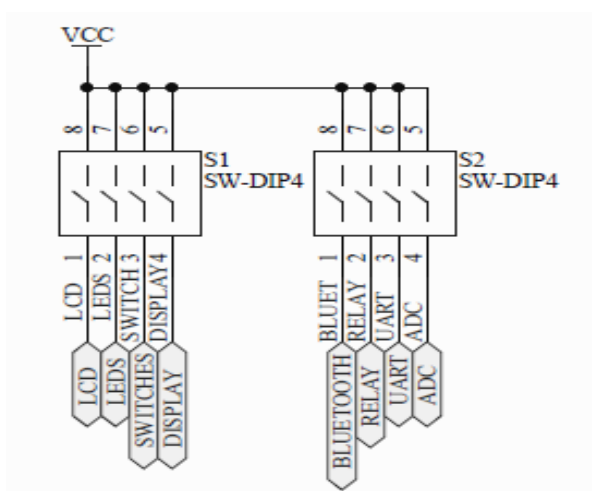


Ilustración 3.51: Dip Switch
Fuente: Altium Designer

Potenciómetro

Es un periférico de salida, es una resistencia variable, sirve para variar la intensidad de corriente.

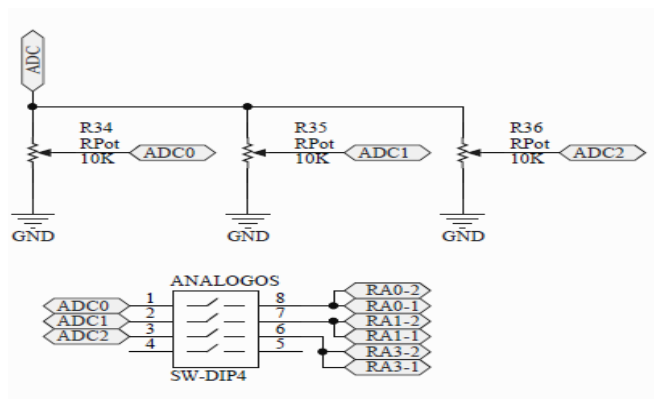


Ilustración 3.52: Potenciómetro

Fuente: Altium Designer

Microcontrolador

- **Pines**

Aquí se muestran a donde va conectado cada módulo anteriormente descrito según los puertos, ejemplo: Pulsadores (SW1) RA4-1 se conecta al PIN 3 que se observa en el gráfico.

Además cabe indicar que el código que no tiene ningún guion numeral pertenece al PIC de 18 pines; el que contiene terminación -2 pertenece al de 28 pines y el que contiene terminación en -1 pertenece al de 40 pines. Para mejor entendimiento estos se muestra a continuación:

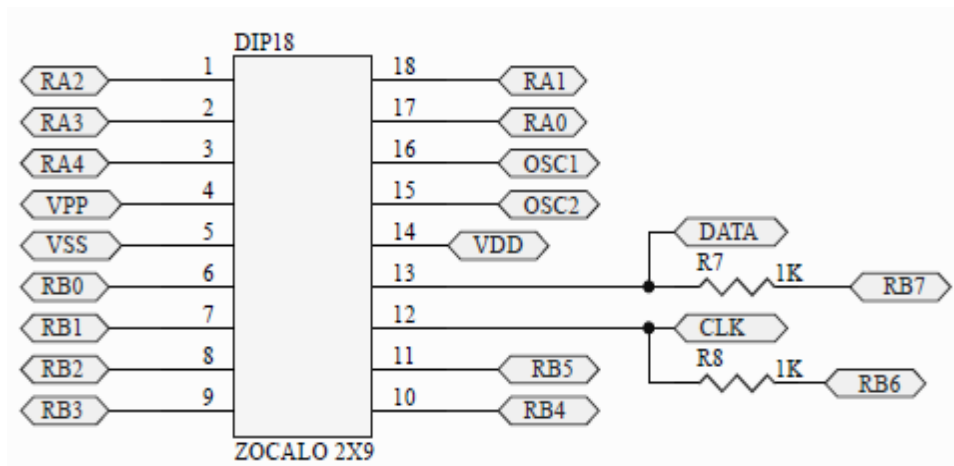


Ilustración 3.53: Puerto 18 pines
Fuente: Altium Designer

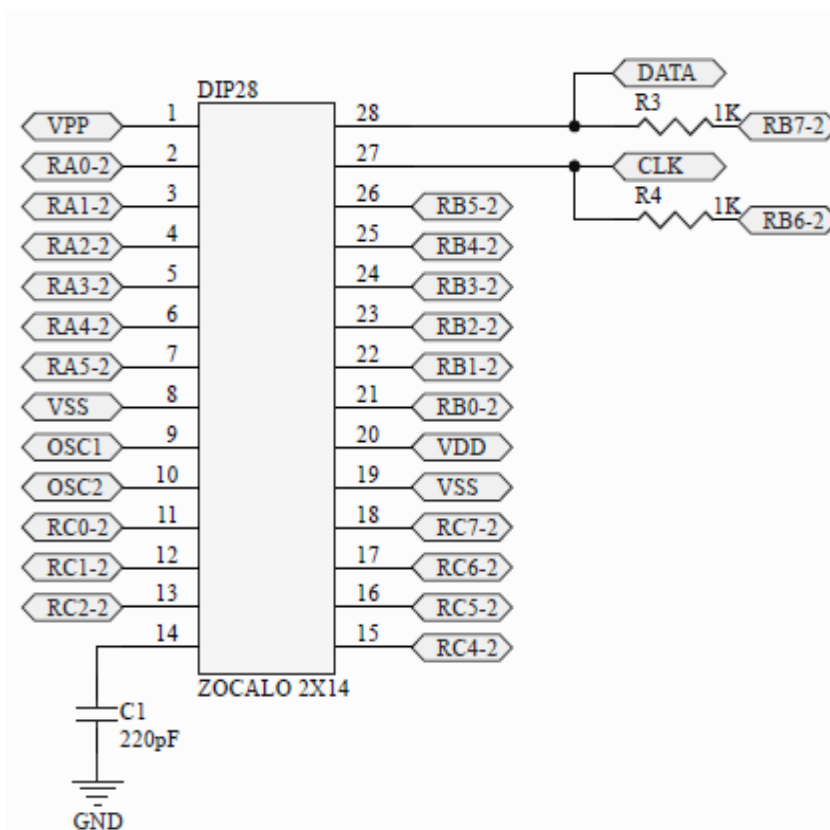


Ilustración 3.54: Puerto 28 pines
Fuente: Altium Designer

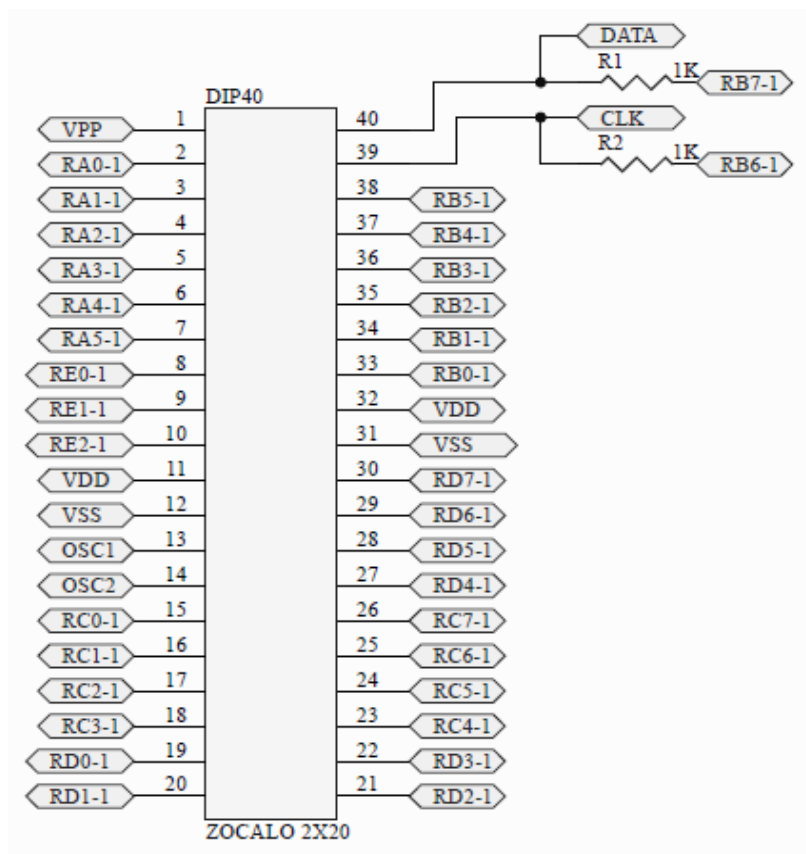


Ilustración 3.55: Puerto 40 pines

Fuente: Altium Designer

- **Conecciones compartidas**

Son conecciones y elementos propios del microcontrolador para poder funcionar correctamente, estos son:

Alimentacion y Masa

Recibe fuente de energía externa positivo y negativo

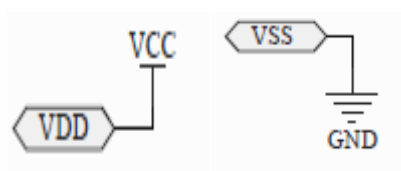


Ilustración 3.56: Alimentación y masa

Fuente: Altium Designer

Capacitores de 22pf

Controla la frecuencia en la que trabaja el microcontrolador.

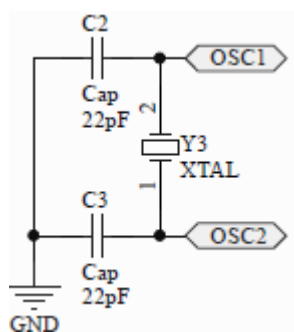


Ilustración 3.57: Capacitores de 22pf
Fuente: Altium Designer

Botón Reset (Sirve para reiniciar el programa del PIC)

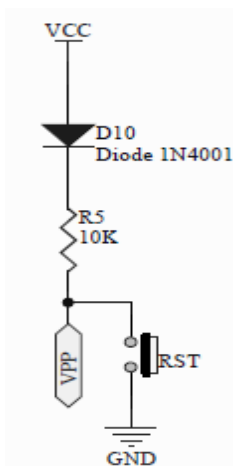


Ilustración 3.58: Botón Reset
Fuente: Altium Designer

Resistencias

Sirven para impedir o limitar el paso de corriente según cada elemento.

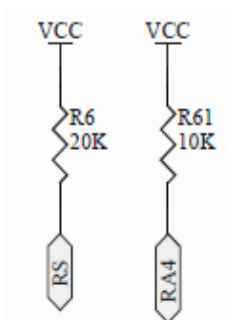


Ilustración 3.59: Resistencias de 20k y 10k
Fuente: Altium Designer

Capa Trasera

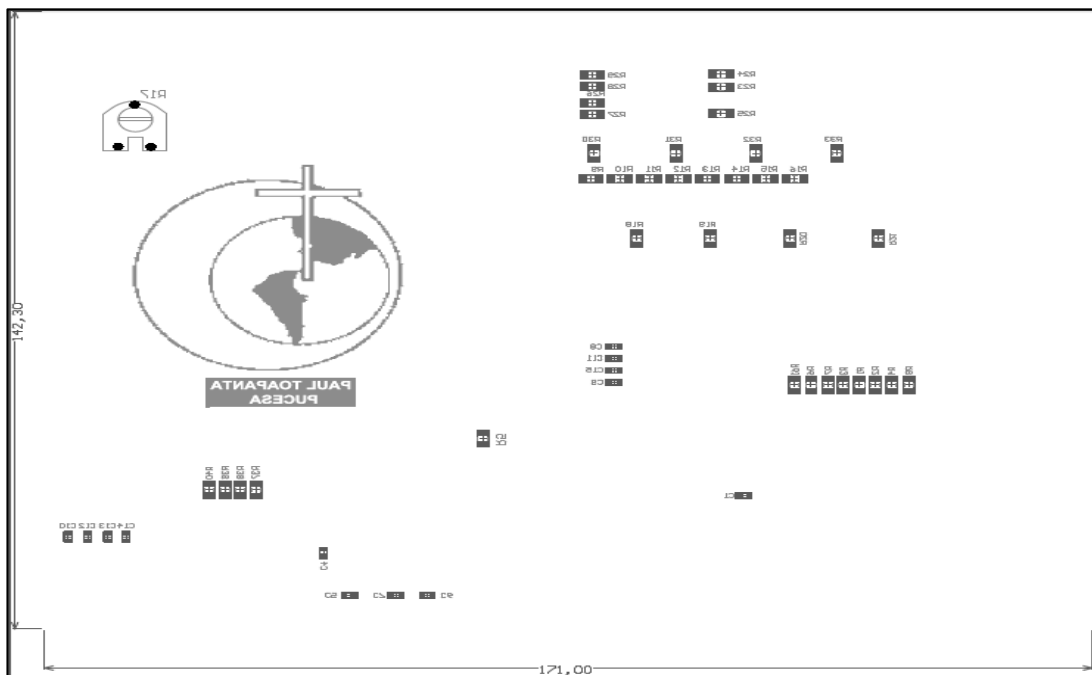


Ilustración 3.61: Prototipo Capa Trasera
Fuente: Altium Designer

3.1.3.3 Fase 3 Diseño Final Y Generación de PCB

Aquí se muestra el resultado de la simulación y como quedaría el entrenador final según la simulación así:

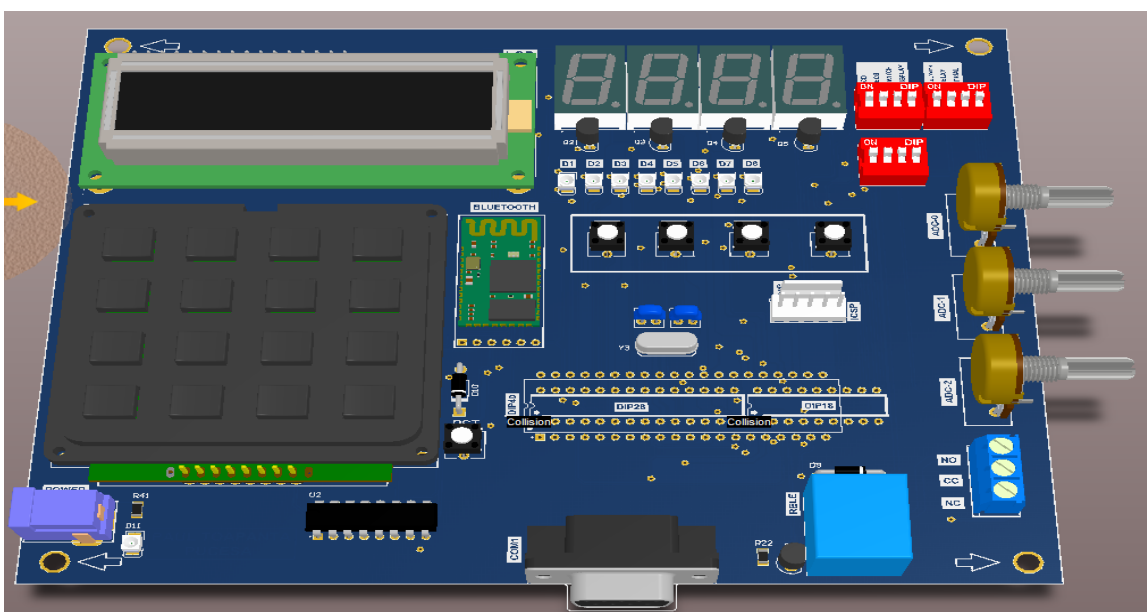


Ilustración 3.62: Simulación Capa Superior
Fuente: Altium Designer

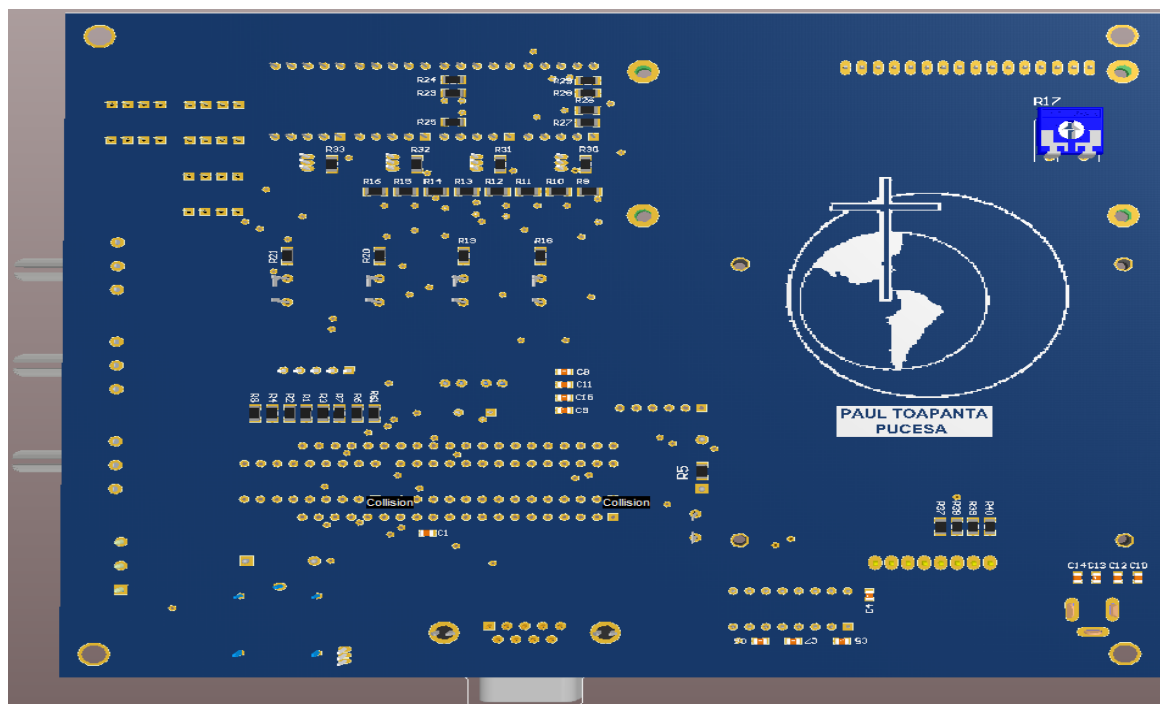


Ilustración 3.63: Simulación Capa Trasera
Fuente: Altium Designer

Generación de la PCB

Con el resultado final según la simulación termina la etapa de diseño y se procede a la fabricación. Para esto se necesita de los planos que nos proporciona el software así:

Capa 1:

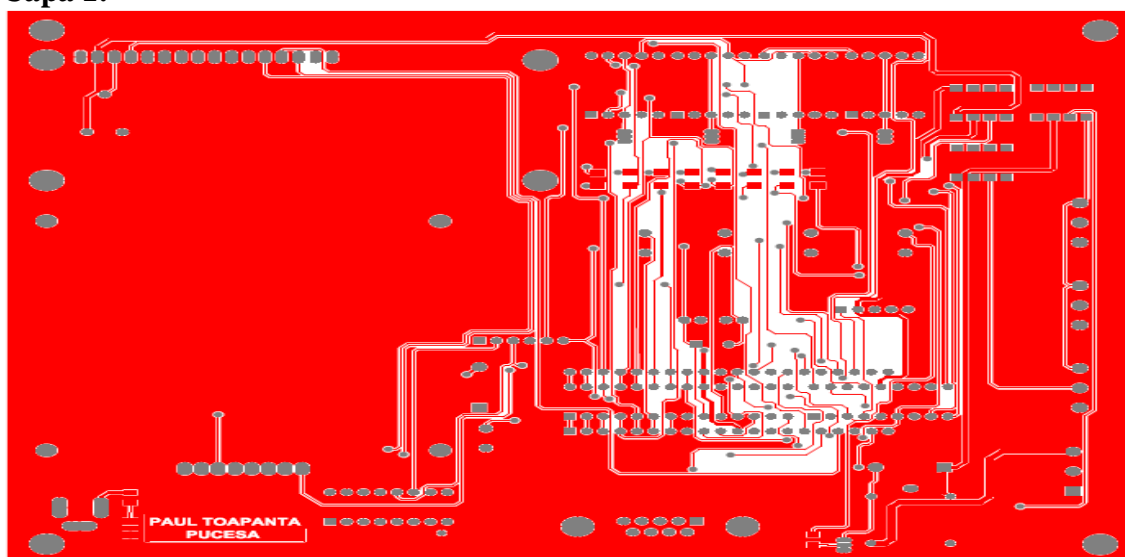


Ilustración 3.64: Plano Capa 1
Fuente: Altium Designer

Capa 2:

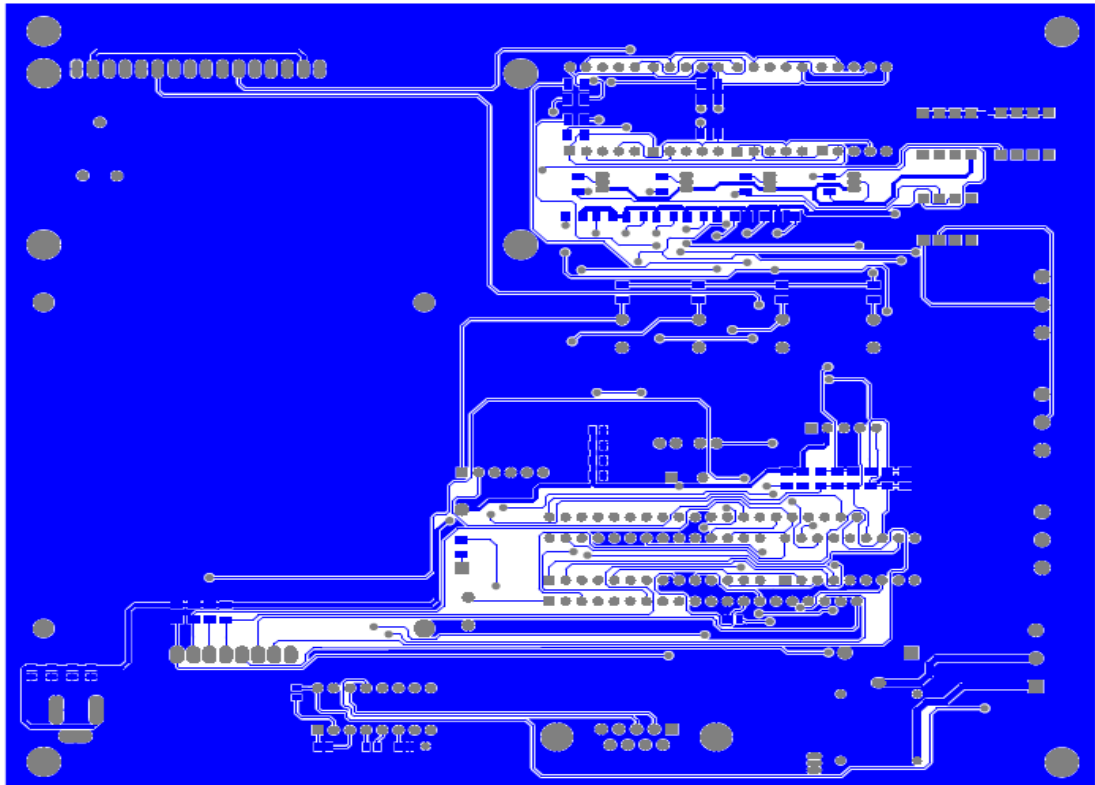


Ilustración 3.65: Plano Capa 2
Fuente: Altium Designer

Capa 3: Nomenclatura

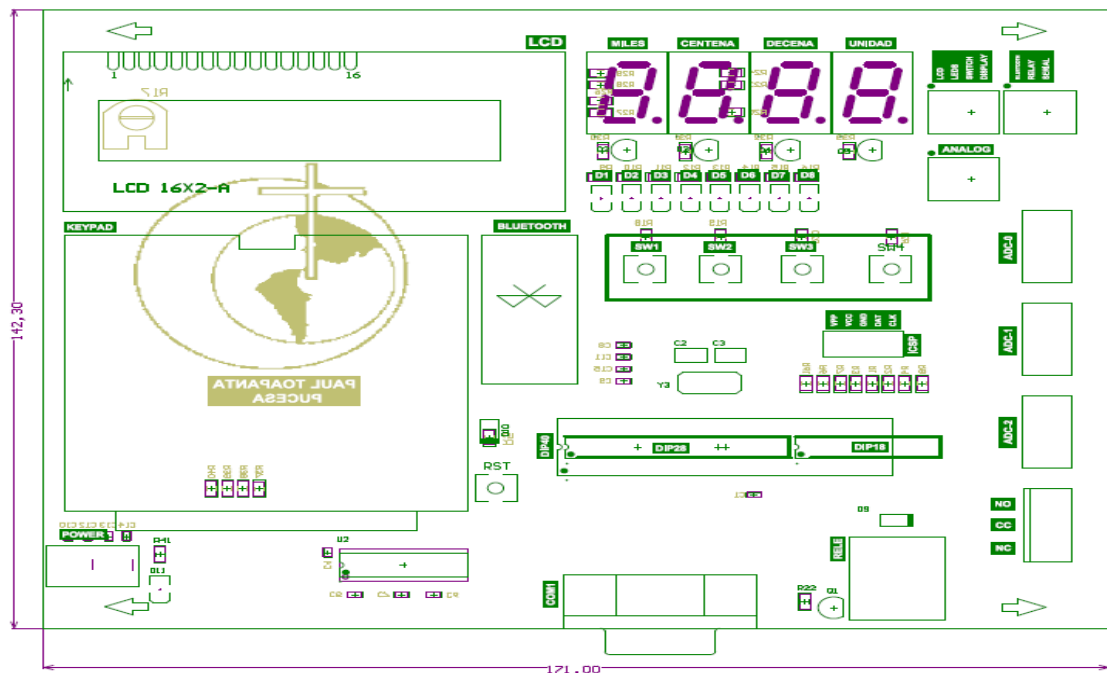


Ilustración 3.66: Plano Capa 3
Fuente: Altium Designer









3.1.4 Lanzamiento











3.1.4.1 Fase 1: Elección de Elementos:




Los elementos que se van a utilizar para la construcción del entrenador son:

Tabla 3.2: Elección de Elementos

Detalle	Cantidad	Encapsulado	Referencia
Diodos LED Rojo	15	SMD (máximo 4mm de largo)	
Pulsador 2 pines	6	Hueco pasante	
LCD 16x2 con Backlight	1	Hueco pasante	
Dip switch 4 contactos	4	Hueco pasante	
Rele 5V	1	Hueco pasante	
DB9 Hembra	1	Hueco pasante	
Display Ánodo Común	4	Hueco pasante	
Teclado 4x4	1	Hueco pasante	

Zócalo 40 pines (20 pines por lado)	2	Hueco pasante	Para PIC18F877 
Zócalo 28 pines (14 pines por lado)	2	Hueco pasante	Para PIC18F2550 
Zócalo 18 pines (9 pines por lado)	2	Hueco pasante	Para PIC16f628 
Zócalo 16 pines (8 pines por lado)	2	Hueco pasante	Para el MAX232 
Conector tipo JACK	1	Hueco pasante	
Cristal 4MHz	1	Hueco pasante	
Cristal 10MHz	1	Hueco pasante	
Capacito cerámico 22pF	4	Hueco pasante	

Resistencia 330 Ohmios	20	Montaje Superficial SMD 1206	
Resistencia 4.7K Ohmios	20	Montaje Superficial SMD 1206	
Resistencia 1K Ohmios	10	Montaje Superficial SMD 1206	
Max232	2	Hueco pasante	
Capacitor 0.1uF	10	Montaje Superficial SMD 0805	
2n3904	2	Hueco pasante	
2n3906	8	Hueco pasante	
Diodos 1n4001	4	Hueco pasante	
Bluetooth HC05	1	Hueco pasante	
Baquelita cobre (143x171mm)	1	Cobre a los dos lados	

Bornera 3 pines	1	Hueco pasante	
Cargador de 5v y 0.5 ^a	1	De Celular	
Cloruro férrico ½ libra	1		

Elaborado por: Paúl Toapanta

3.1.4.2 Fase 2 Preparación de Baquelita:

Todos los métodos de fabricación tienen el mismo principio, que es el de transferir el diseño PCB en forma de espejo sobre la placa o baquelita, lo que varía es el método para hacerlo, existen métodos manuales y automatizados y estos son:

Paso 1: Transferir el diseño

Transferir el diseño PCB en espejo a la placa por el método más conveniente, en este caso se lo puede hacer por tóner, el cual se imprime en una hoja de papel transfer o papel fotográfico y se transfiere a la placa por medio de una plancha por 15 minutos.



Ilustración 3.67: Transferencia de diseño
Fuente: Transferencia del diseño, Paúl Toapanta

Paso 2: Lavado de la placa

La persona que ejecuta este paso debe tomar precauciones como: usar gafas protectoras, mascarilla y guantes y hacerlo en un lugar ventilado.

Una vez considerado lo anterior se debe seguir el siguiente proceso: Disolver el contenido de cloruro férrico sobre un recipiente de preferencia plástico para que no afecte el ácido a componentes metálicos, colocar agua tibia e introducir la placa con el diseño.

Esperar entre 15 a 20 minutos hasta ver disuelto todo el cobre que no está cubierto por el diseño.

Después de ese tiempo lavar muy bien la placa con agua y jabón para quitar todos los residuos desprendidos.



Ilustración 3.68: Lavado de la Placa

Fuente: Lavado de la placa, Paúl Toapanta

Paso 3: Perforación

Una vez limpia la placa se procede a perforarla con un taladro de mano en el lugar donde se van a introducir los elementos.

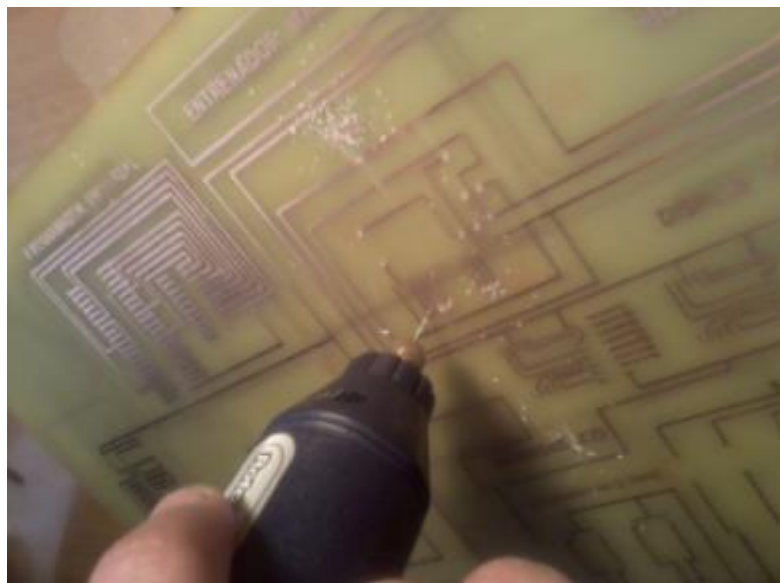


Ilustración 3.69: Perforación de la placa

Fuente: Perforación de la placa, Paúl Toapanta

Paso 4: Opcional.

Existen métodos que se utilizan como SILK SCREEN que es una capa protectora para la placa, una vez terminado la transferencia del diseño a la misma.

Resultado final

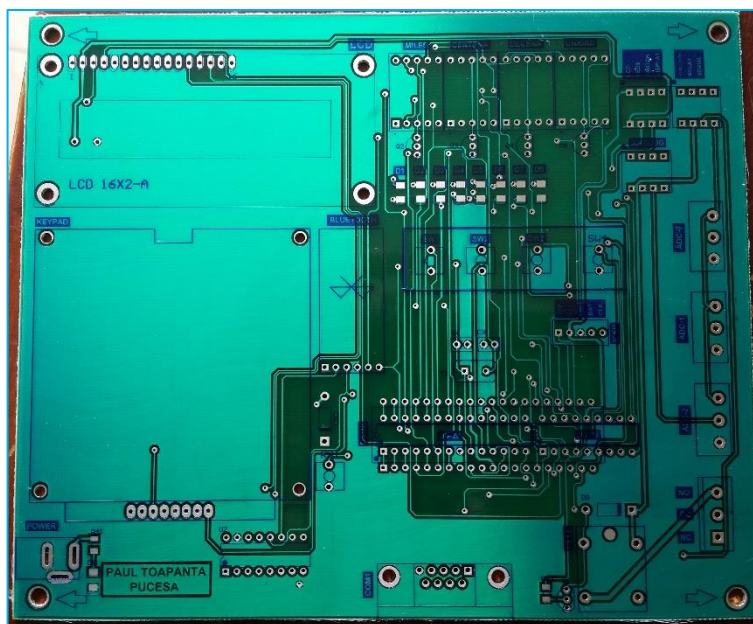


Ilustración 3.70: Resultado final preparación de la Baquelita Capa superior
Fuente: Entrenador para Microcontroladores PIC, Paúl Toapanta

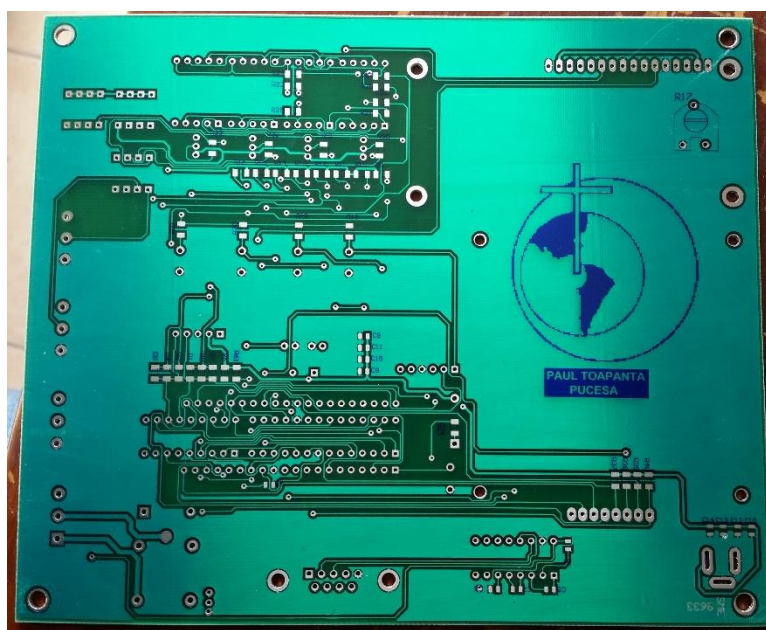


Ilustración 3.71: Resultado final preparación de la Baquelita Capa posterior
Fuente: Entrenador para Microcontroladores PIC, Paúl Toapanta

3.1.4.3 Fase 3 Soldadura:

La soldadura con estaño establece la unión conductiva entre el elemento y las pistas de la placa, la mayoría de las veces causa muchos problemas este procedimiento ya que no se lo realiza de la forma correcta, y la solución para esto es usar pasta soldadora y estaño de buena calidad. La pasta permite la distribución correcta del estaño sobre el elemento y la pista de cobre. Al Principio va a ser difícil, pero con la práctica se va mejorando la técnica. Los pasos que se deben seguir son los siguientes:

Paso 1

Colocar los elementos en su sitio ya definido, para esto hay que cortar los pines de tal manera que quede bien empotrado en la placa.

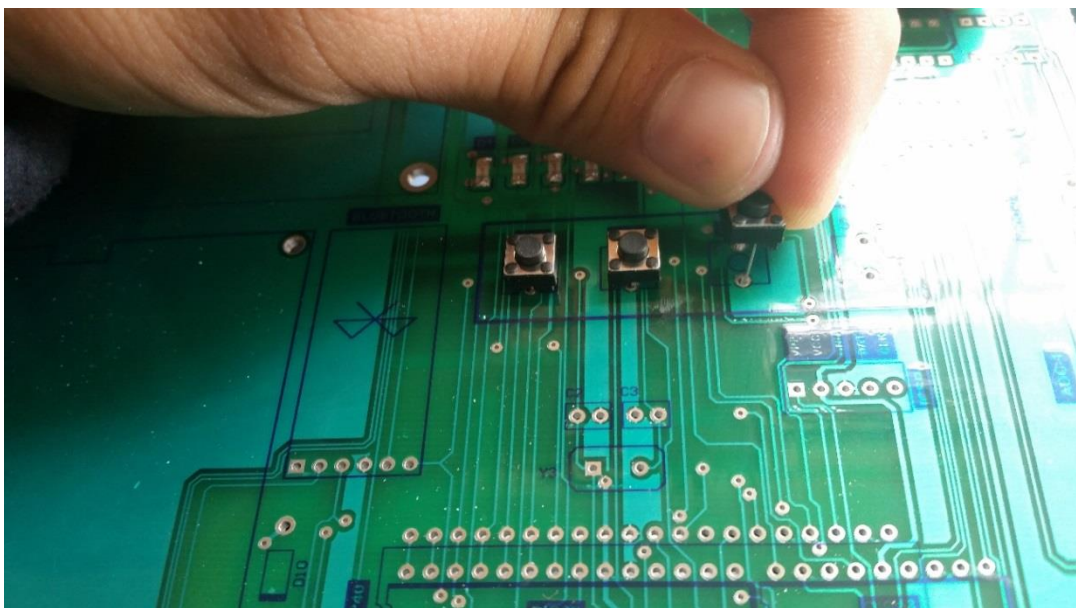


Ilustración 3.72: Colocación de elementos en la placa
Fuente: Entrenador para Microcontroladores PIC, Paúl Toapanta

Paso 2

Finalmente se debe soldar la pieza con un caudín, poner pasta y estaño.

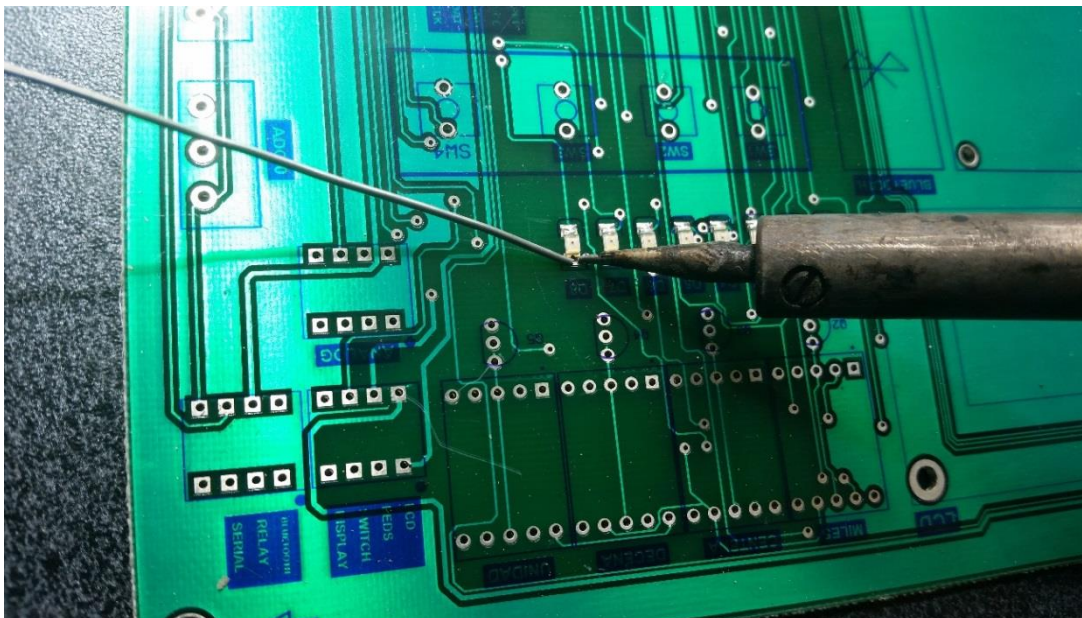


Ilustración 3.73: Soldadura con caudín

Fuente: Entrenador para Microcontroladores PIC, Paúl Toapanta

3.1.4.4 Fase 4 Pruebas:

En esta fase se debe verificar cada elemento que este bien soldado así:

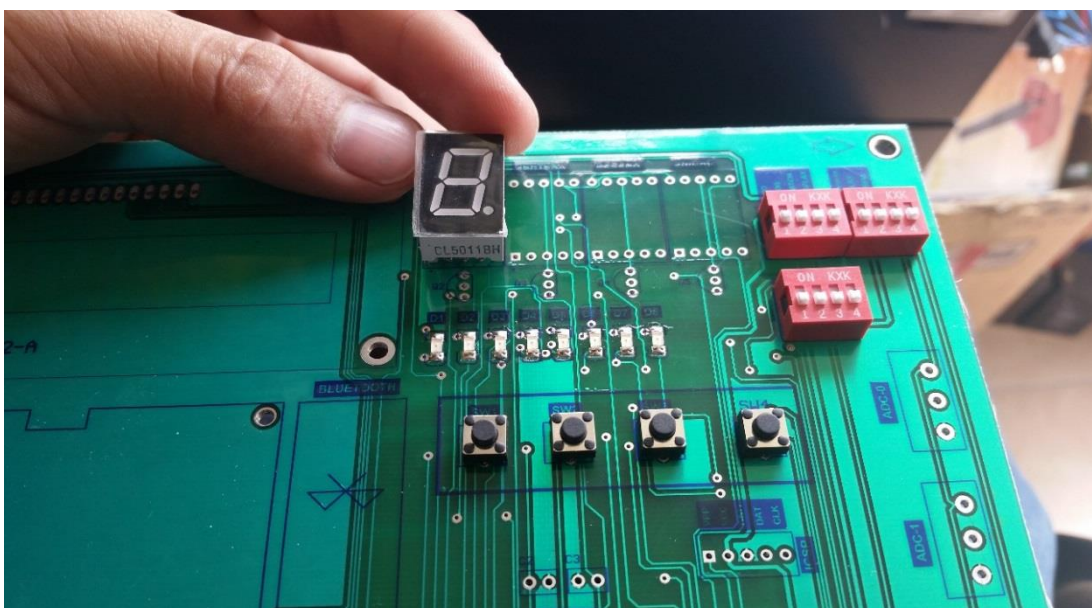


Ilustración 3.74: Fase de pruebas

Fuente: Entrenador para Microcontroladores PIC, Paúl Toapanta

Resultado final

Una vez que se han cumplido con todas las fases determinadas para el diseño y la construcción del entrenador basado en microcontroladores PIC, se obtuvo el siguiente resultado:

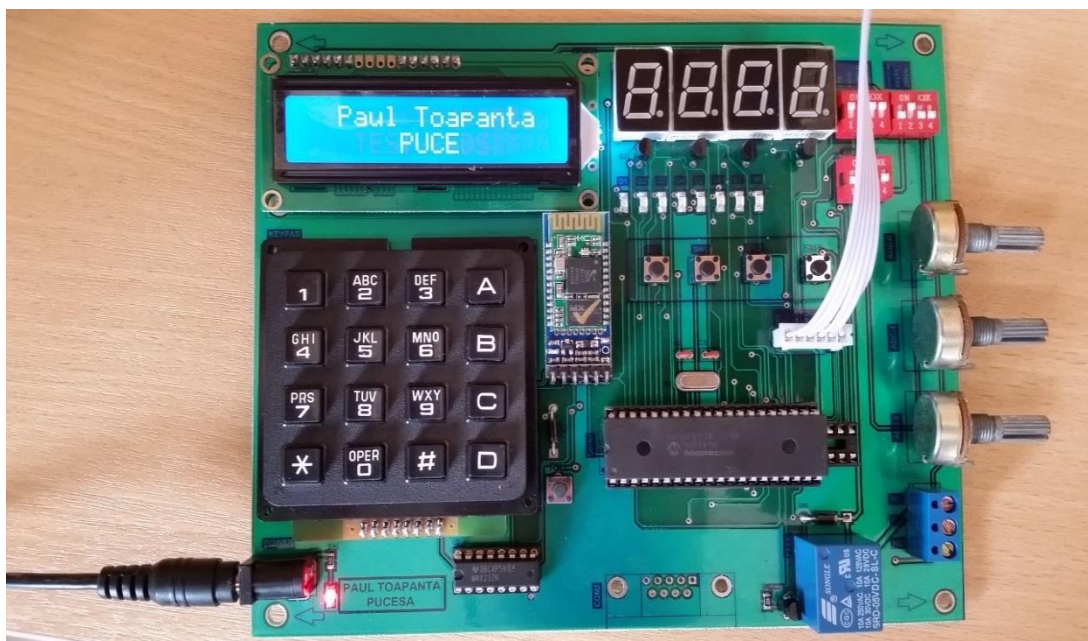


Ilustración 3.75: Resultado Final de del diseño y construcción del Entrenador
Fuente: Entrenador para Microcontroladores PIC, Paúl Toapanta

CAPITULO IV

ANÁLISIS Y VALIDACIÓN DE LOS RESULTADOS

4.1 Procedimientos para la utilización del entrenador

Los procedimientos que se deben seguir para utilizar el entrenador son los siguientes:

Tabla 4.3: Procedimientos para utilizar el entrenador

Evento	Proceso
Pedido de Herramienta	En el laboratorio de Electrología el estudiante deberá solicitar al Docente encargado el entrenador universal PIC.
Constatar el estado del entrenador	El Docente encargado y el alumno deberán constatar el estado en que se entrega el entrenador, sus partes o accesorios.
Programación	El estudiante para poder programar el PIC de su preferencia deberá tener un conocimiento básico del esquema del mismo, con el cual determinara que módulos están disponibles para el PIC de su elección.
Pruebas	Una vez finalizada la programación el estudiante podrá grabar el archivo hexadecimal, resultado de la compilación de la programación, al PIC mediante el módulo de grabar, que se encuentra en el programador.
Resultados	Aquí podrá determinar si el resultado es satisfactorio o deberá realizar correcciones en la programación, y/o repetir los procesos desde la programación, compilación y grabación.

Elaborado por: Paúl Toapanta

4.2 Guía del usuario del entrenador PIC

Para poder programar un microcontrolador PIC se deberá utilizar el software Microcode Studio el cual se lo puede descargar de internet y es de licencia gratuita, o

cualquier software programador de PIC. El estudiante para utilizar esta herramienta debe realizar los siguientes pasos:

Ingresa en el software, para ello se deberá dar doble click sobre el icono CodeStudio.



Ilustración 4.76: Icono CodeStudio
Fuente: Escritorio de Windows

Aparecerá la siguiente ventana:

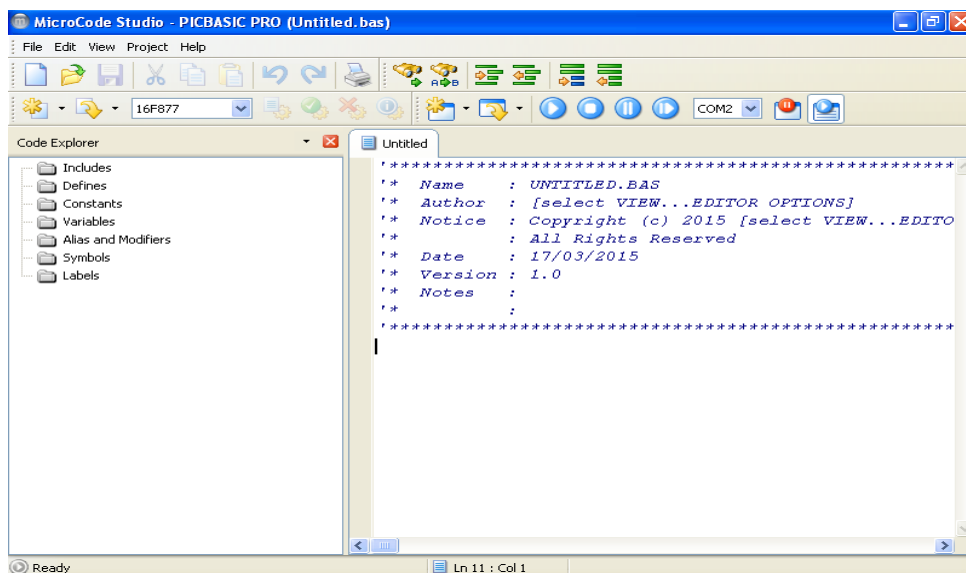


Ilustración 4.77: Ingreso al Software
Fuente: MicroCode Studio

Una vez dentro del programa se debe determinar el microcontrolador que se va a utilizar

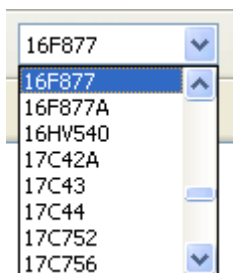


Ilustración 4.78: Selección del microcontrolador
Fuente: MicroCode Studio

Además, se debe saber el módulo del entrenador que se va a utilizar para las pruebas, puede ser Leds, Serial, Rele, Bluetooth, Pulsadores, Teclado, Display, LCD, Potenciómetro, o todos los anteriores.

Para esto el entrenador dispone de unos Interruptores llamados DipSwitch los cuales habilitan o deshabilitan los módulos con los que se desee entrenar así:

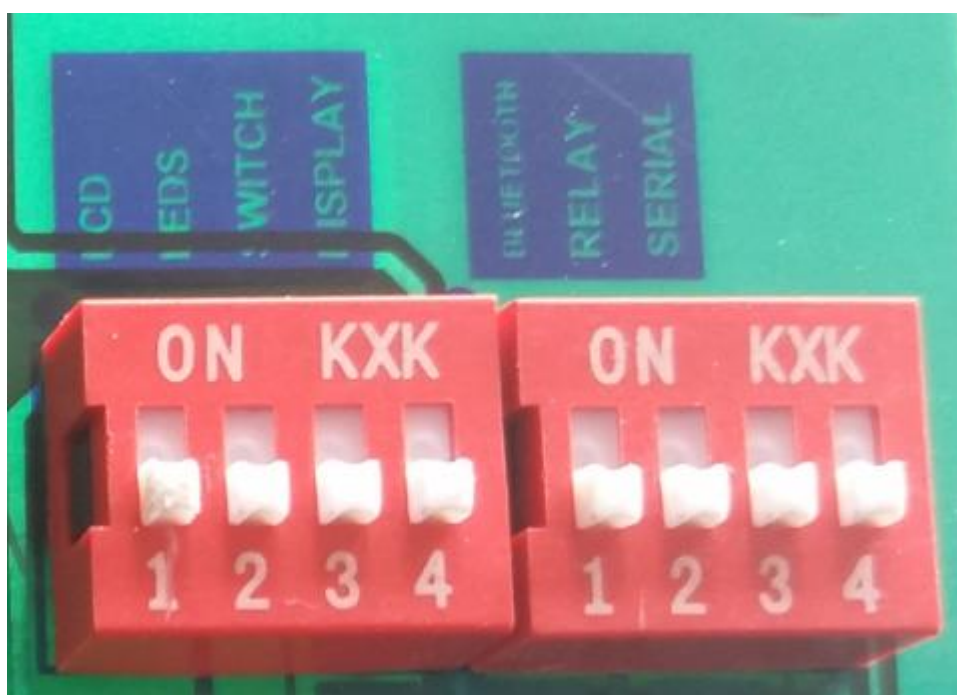


Ilustración 4.79: Interruptores DipSwitch
Fuente: Entrenador para Microcontroladores PIC, Paúl Toapanta

Cada interruptor está identificado con su respectivo nombre, en el sentido izquierda-derecha se encuentra distribuido así:

1=LCD

2=LEDS

3=SWITCH

4=DISPLAY

1=BLUETOOTH

2=RELAY

3=SERIAL

4= NO ASIGNADO (El ultimo es libre, o no tiene asignado ningún modulo).

Después de determinar el modulo a usar se procede a programar el PIC, en este caso se utilizará el modulo “LEDS”.

Con el siguiente código se hará una secuencia de luces. Los 8 leds se encenderán de manera secuencial, de izquierda a derecha y se apagara de la misma manera.

Programacion:

TEST LEDS

unidad = 1

Decena = 1

Centena = 1

Miles = 1

X = 255

for y = 0 to 255

portb = x

```
    pause 50
    x = x - 1
    if x = 0 then
        x = 255
    endif
next
for y = 1 to 6
    portb = %01111111
    pause 100
    portb = %10111111
    pause 100
    portb = %11011111
    pause 100
    portb = %11101111
    pause 100
    portb = %11110111
    pause 100
    portb = %11111011
    pause 100
    portb = %11111101
    pause 100
    portb = %11111110
    pause 100
    portb = %11111101
    pause 100
    portb = %11111011
```

```

pause 100

portb = %11110111

pause 100

portb = %11101111

pause 100

portb = %11011111

pause 100

portb = %10111111

pause 100

next

'FIN TEST LEDS

```

Una vez hecha la programación se deberá guardar el archivo, en este caso se lo llamara Leds.pbp y luego compilar presionando la Tecla F9 en el software MicroCode.

Es muy importante que en el resultado de la compilación no existan errores.

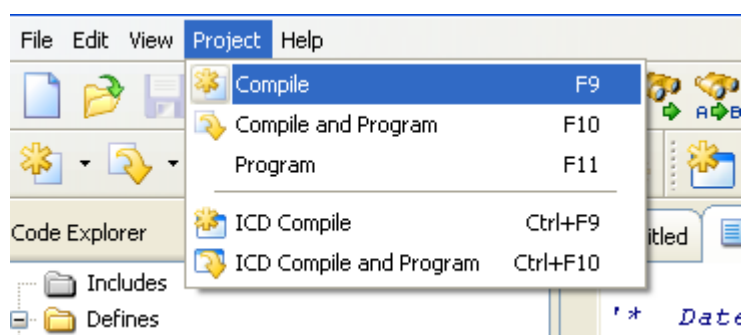


Ilustración 4.80: Opción Compile

Fuente: MicroCode Studio

Este proceso permitirá crear 3 archivos más, con el mismo nombre, pero diferente extensión, por ejemplo:

LEDS.MAC

LEDS.HEX

LEDS.ASM

El que interesa es LEDS.HEX, ya que contiene la programación en hexadecimal, que se deberá grabar en el microcontrolador PIC.



Ilustración 4.81: Archivo Leds.Hex
Fuente: MicroCode Studio

Una vez hecho esto se debe grabar este archivo en el microcontrolador PIC, por medio del software y el grabador, para lo cual se debe hacer de la siguiente forma.

- Colocar el PIC en el entrenador así:

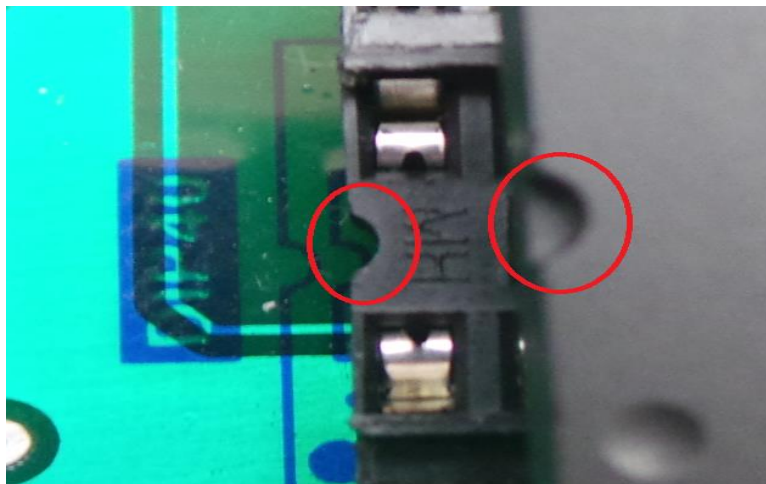


Ilustración 4.82: Colocación del PIC en el entrenador
Fuente: Entrenador para Microcontroladores PIC, Paúl Toapanta

Hay que considerar el lado de la muesca al momento de insertar el zócalo correspondiente.

- Conectar el Accesorio PICKIT 2 al entrenador

Es muy importante seguir el orden de conexión que muestra la figura (VPP con VPP), ya que se podrá causar daños permanentes en el PIC o en la placa entrenadora, como referencia se tomara el pin VPP del entrenador y se lo unirá con el otro extremo del cable en la misma posición VPP de PICKIT2.



Ilustración 4.83: Orden de conexión

Fuente: Entrenador para Microcontroladores PIC, Paúl Toapanta

- Conectar el cable USB del Accesorio al computador.
- Abrir el software PICKIT 2 que es de licencia libre y se lo puede descargar desde el internet.



Ilustración 4.84: Software PICKIT2

Fuente: Escritorio de Windows

Si el software no reconoce el dispositivo aparecerá el siguiente mensaje en el software:

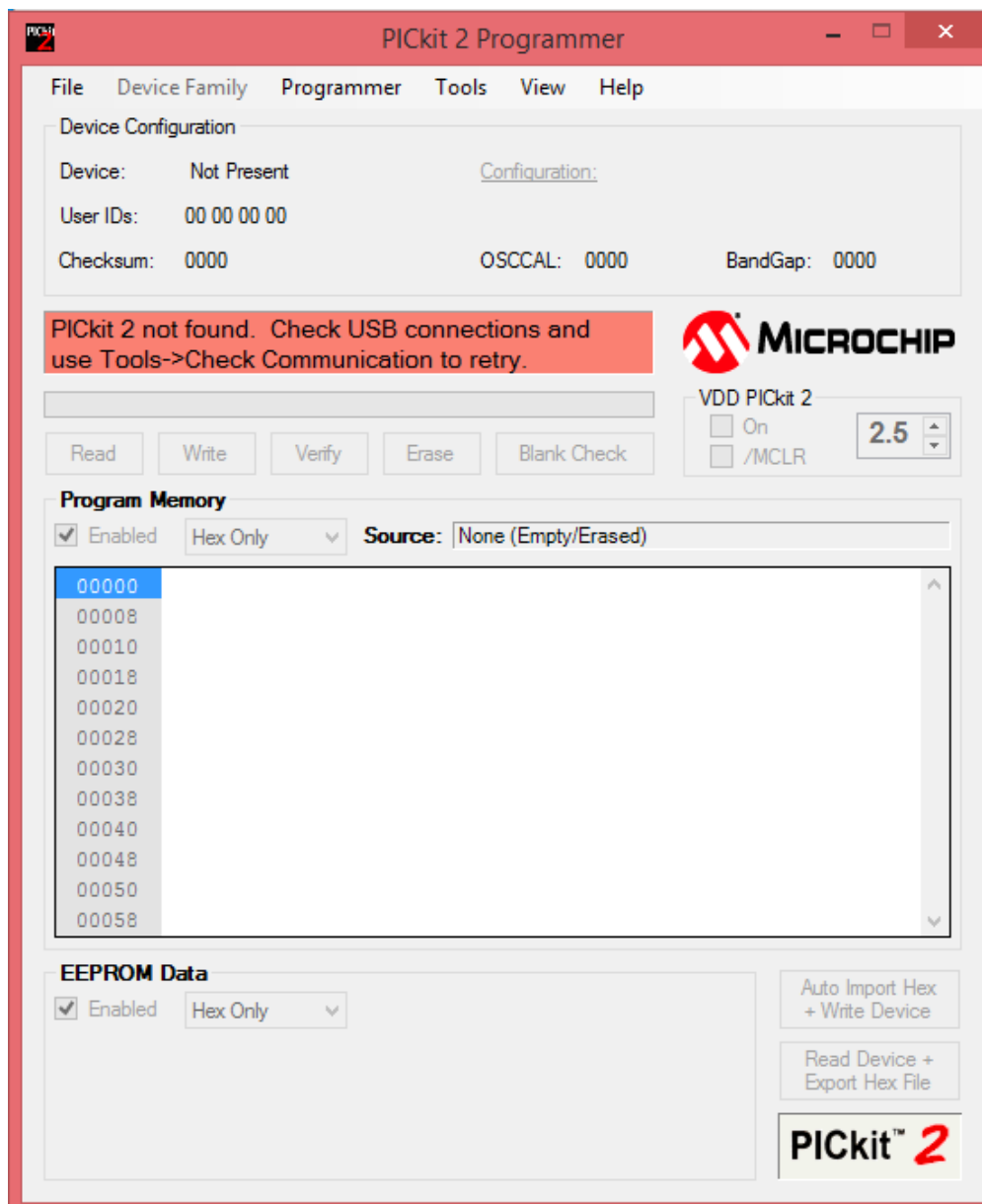


Ilustración 4.85: Mensaje del Software cuando no reconoce el dispositivo
Fuente: Software Pickit2

Caso contrario, reconocerá el modelo de PIC y desplegará la información que este contiene así:

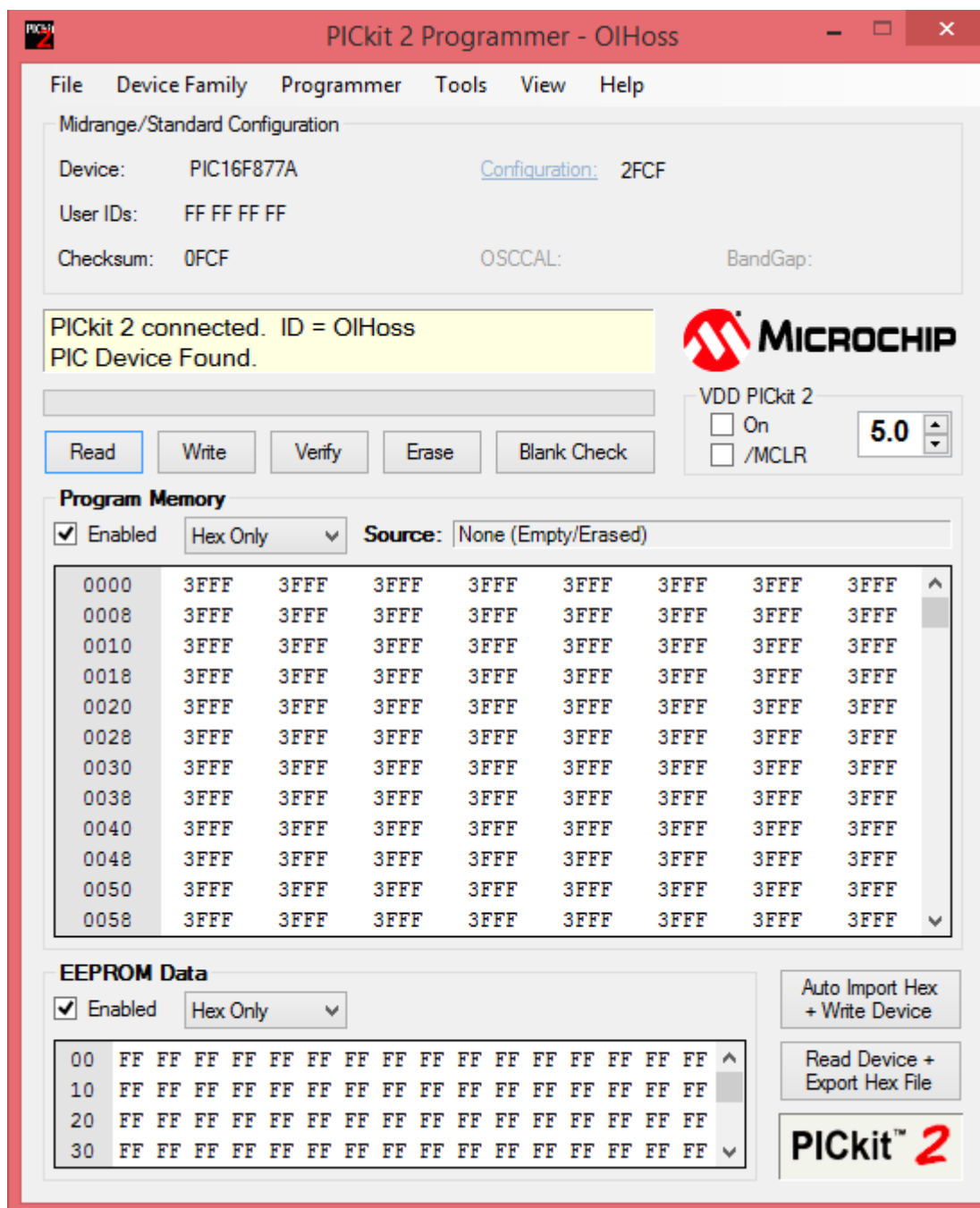


Ilustración 4.86: Mensaje del Software cuando reconoce el dispositivo
Fuente: Software Pickit2

- Una vez que el software ha reconocido el dispositivo de debe Seleccionar File/Import Hex así:

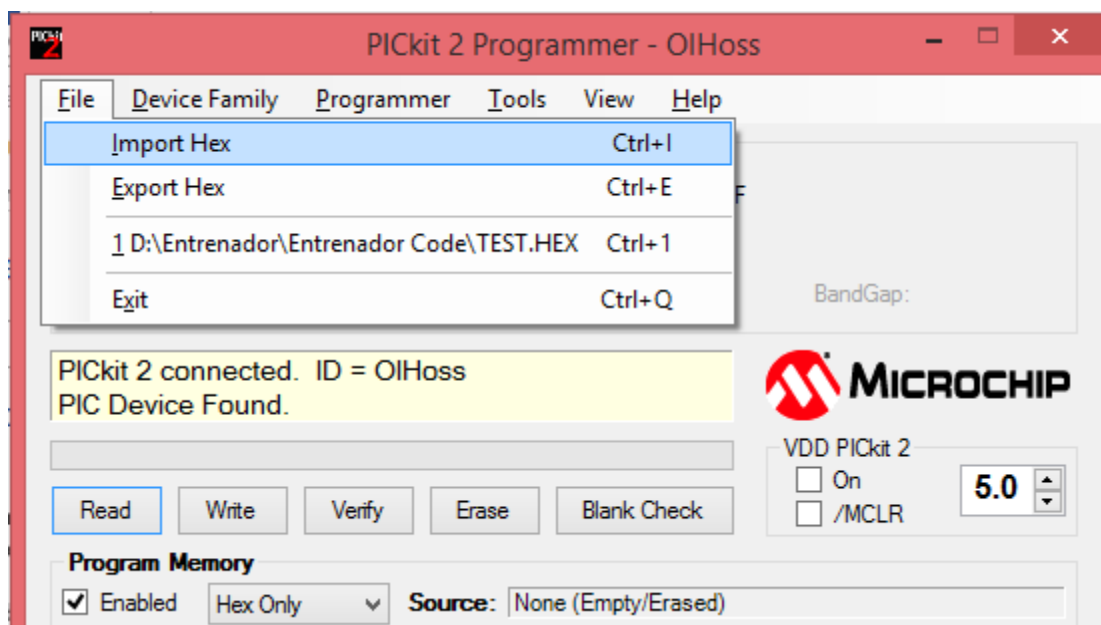


Ilustración 4.87: Selección File/Import Hex
Fuente: Software Pickit2

Aquí se debe seleccionar el archivo que se creó anteriormente LEDS.HEX



Ilustración 4.88: Selección de archivo Leds.Hex
Fuente: MicroCode Studio

Y presionar el botón GRABAR.



Ilustración 4.89: Botón Grabar
Fuente: Software Pickit2

Listo el PIC quedará grabado con la nueva información.

Y como resultado tendremos la secuencia de luces en el módulo led según la programación así.



Ilustración 4.90: Secuencia de luces módulo led
Fuente: Entrenador para Microcontroladores PIC, Paúl Toapanta

EL accesorio Grabador se lo puede retirar o no del entrenador por comodidad.

Una cosa muy importante a considerar, es que en el módulo Potenciómetro tiene dos estados, uno Analógico y otro Digital. Siendo Analógico en On y Digital en Off, siempre se debe trabajar en OFF cuando no se utilice este módulo.

Hay que tener muy en cuenta con que estado señal se va a trabajar porque si se utiliza el modo incorrecto puede producir un corto circuito y dañar el PIC.

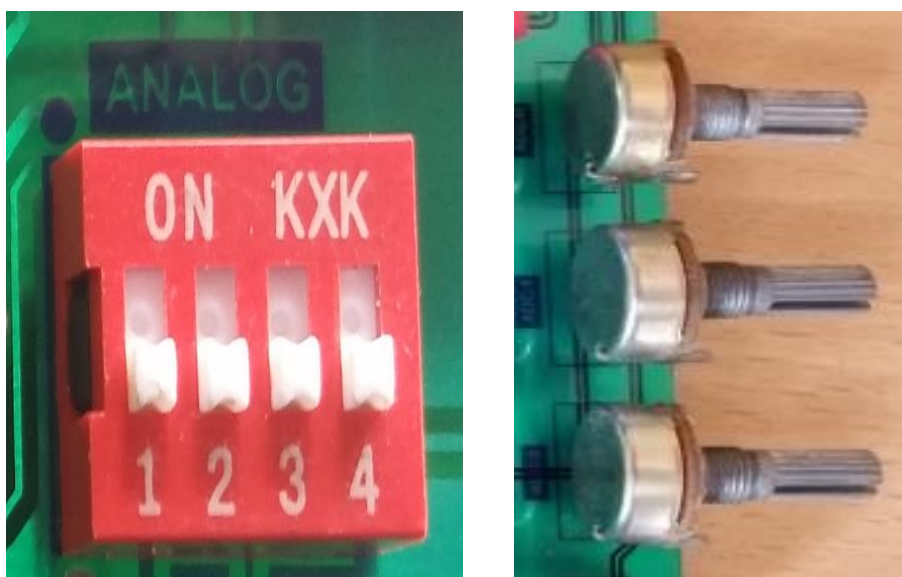


Ilustración 4.91: Estados del Potenciómetro
Fuente: Entrenador para Microcontroladores PIC, Paúl Toapanta

4.3 Programación de un ejercicio completo

```

*****
'* Name   : UNTITLED.BAS                *
'* Author : [select VIEW...EDITOR OPTIONS] *
'* Notice : Copyright (c) 2015 [select VIEW...EDITOR OPTIONS] *
'*       : All Rights Reserved          *
'* Date   : 11/03/2015                  *
'* Version : 1.0                        *
'* Notes  :                             *
'*       :                               *
*****

INCLUDE "modedefs.bas"

@ DEVICE HS_OSC

define OSC 10

trisb = 0 'Todo puerto B como salida

adcon1 = 7 'Convierte en digitales los pines del puerto A y E

'TRISB.0 = 0 'Salida (Fila 1)

trisa = 0

trisb = 0 'Todo puerto B como salida

Unidad var porta.0

Decena var porta.1

Centena var porta.2

Miles var porta.3

'DEFINE LCD_BITS 4

DEFINE LCD_DREG PORTC

DEFINE LCD_DBIT 0

DEFINE LCD_RSREG PORTC

DEFINE LCD_RSBIT 4

```

```
DEFINE LCD_EREG  PORTC
DEFINE LCD_EBIT  5
X VAR BYTE
x = 255
y var byte
abc var byte
inicio:
portb = %11111111 'APAGAR LEDS
LCDOUT $FE,1
for x = 0 to 16
lookup x,[" Paul Toapanta "],abc
lcdout abc
pause 400
next
pause 2000
LCDOUT $FE,1, " Paul Toapanta "
LCDOUT $FE,$C0,"  PUCE  "
pause 3000
LCDOUT $FE,1, " Paul Toapanta "
LCDOUT $FE,$C0,"  TEST LEDS  "
pause 1000
'TEST LEDS
    unidad = 1
    DEcena = 1
    Centena = 1
    Miles = 1
X = 255
for y = 0 to 255
    portb = x
```

```
    pause 50
    x = x - 1
    if x = 0 then
        x = 255
    endif
next
for y = 1 to 6
    portb = %01111111
    pause 100
    portb = %10111111
    pause 100
    portb = %11011111
    pause 100
    portb = %11101111
    pause 100
    portb = %11110111
    pause 100
    portb = %11111011
    pause 100
    portb = %11111101
    pause 100
    portb = %11111110
    pause 100
    portb = %11111101
    pause 100
    portb = %11111011
    pause 100
    portb = %11110111
    pause 100
```

```
portb = %11101111
pause 100
portb = %11011111
pause 100
portb = %10111111
pause 100
next
'FIN TEST LEDS
' TEST DISPLAY
LCDOUT $FE,1, " Paul Toapanta "
LCDOUT $FE,$C0," TEST DISPLAY "
pause 1000
FOR Y = 0 TO 2
    gosub test_display
NEXT
' FIN TEST DISPLAY
' TEST RELAY
    unidad = 1
    DEcena = 1
    Centena = 1
    Miles = 1
portb = 255 'APAGAR LEDS
LCDOUT $FE,1, " Paul Toapanta "
LCDOUT $FE,$C0," TEST RELAY "
pause 1000
trise.2 = 0
RELAY VAR porte.2
RELAY = 1
pause 1000
```

```
relay = 0
pause 1000
RELAY = 1
pause 1000
relay = 0
pause 1000
RELAY = 1
pause 1000
relay = 0
pause 1000
'FIN TEST RELAY
'TEST PULSADORES
LCDOUT $FE,1, " Paul Toapanta "
LCDOUT $FE,$C0," TEST PULSADORES"
PAUSE 3000
trise = %110
trisa.4 = 1
trisa.5 = 1
sw1 var porta.4
sw2 var porta.5
sw3 var porte.1
sw4 var porte.2
LCDOUT $FE,1, " Paul Toapanta "
LCDOUT $FE,$C0," PULSE UN BOTON "
PAUSE 3000
for x = 1 to 255
PAUSE 50
  if sw1 = 0 then
    LCDOUT $FE,1, " BOTON PULSADO "
```

```
LCDOUT $FE,$C0," SW1 "
```

```
pause 500
```

```
LCDOUT $FE,1, " Paul Toapanta "
```

```
LCDOUT $FE,$C0," PULSE UN BOTON "
```

```
endif
```

```
if sw2 = 0 then
```

```
    LCDOUT $FE,1, " BOTON PULSADO "
```

```
    LCDOUT $FE,$C0," SW2 "
```

```
    pause 500
```

```
    LCDOUT $FE,1, " Paul Toapanta "
```

```
    LCDOUT $FE,$C0," PULSE UN BOTON "
```

```
endif
```

```
if sw3 = 0 then
```

```
    LCDOUT $FE,1, " BOTON PULSADO "
```

```
    LCDOUT $FE,$C0," SW3 "
```

```
    pause 500
```

```
    LCDOUT $FE,1, " Paul Toapanta "
```

```
    LCDOUT $FE,$C0," PULSE UN BOTON "
```

```
endif
```

```
if sw4 = 0 then
```

```
    LCDOUT $FE,1, " BOTON PULSADO "
```

```
    LCDOUT $FE,$C0," SW4 "
```

```
    pause 500
```

```
    LCDOUT $FE,1, " Paul Toapanta "
```

```
    LCDOUT $FE,$C0," PULSE UN BOTON "
```

```
endif
```

```
next
```

```
LCDOUT $FE,1, " Paul Toapanta "
```

```
LCDOUT $FE,$C0," TEST TECLADO "
```

```
PAUSE 3000
LCDOUT $FE,1, " Paul Toapanta "
LCDOUT $FE,$C0," PRESS TECLADO "
PAUSE 1000
A VAR PORTD.0
B VAR PORTD.1
C VAR PORTD.2
D VAR PORTD.3

UNO VAR PORTD.4
DOS VAR PORTD.5
TRES VAR PORTD.6
CUATRO VAR PORTD.7
Z VAR BYTE
FOR Z = 1 TO 255
  low a
  high b
  high c
  high d
  if uno = 0 then
    LCDOUT $FE,1, "TECLA:"
    LCDOUT $FE,$C0,"1"
    PAUSE 1000
  endif
  if DOS = 0 then
    LCDOUT $FE,1, "TECLA:"
    LCDOUT $FE,$C0,"2 / A / B / C"
    PAUSE 1000
  endif
```

```
if TRES = 0 then
    LCDOUT $FE,1, "TECLA:"
    LCDOUT $FE,$C0,"3 / D / E / F"
    PAUSE 1000
endif

if CUATRO = 0 then
    LCDOUT $FE,1, "TECLA:"
    LCDOUT $FE,$C0,"HEX A"
    PAUSE 1000
endif

PAUSE 10

low B
high A
high c
high d

if uno = 0 then
    LCDOUT $FE,1, "TECLA:"
    LCDOUT $FE,$C0,"4 / G / H / I"
    PAUSE 1000
endif

if DOS = 0 then
    LCDOUT $FE,1, "TECLA:"
    LCDOUT $FE,$C0,"5 / J / K / L"
    PAUSE 1000
endif

if TRES = 0 then
    LCDOUT $FE,1, "TECLA:"
    LCDOUT $FE,$C0,"6 / M / N / O"
    PAUSE 1000
```

```
endif
if CUATRO = 0 then
    LCDOUT $FE,1, "TECLA:"
    LCDOUT $FE,$C0,"HEX B"
    PAUSE 1000
endif
PAUSE 10
low C
high A
high B
high d
if uno = 0 then
    LCDOUT $FE,1, "TECLA:"
    LCDOUT $FE,$C0,"7 / P / R / S"
    PAUSE 1000
endif
if DOS = 0 then
    LCDOUT $FE,1, "TECLA:"
    LCDOUT $FE,$C0,"8 / T / U / V"
    PAUSE 1000
endif
if TRES = 0 then
    LCDOUT $FE,1, "TECLA:"
    LCDOUT $FE,$C0,"9 / W / X / Y"
    PAUSE 1000
endif
if CUATRO = 0 then
    LCDOUT $FE,1, "TECLA:"
    LCDOUT $FE,$C0,"HEX C"
```

```
        PAUSE 1000
    endif
    PAUSE 10
    low D
    high A
    high B
    high C
    if uno = 0 then
        LCDOUT $FE,1, "TECLA:"
        LCDOUT $FE,$C0,"*"
        PAUSE 1000
    endif
    if DOS = 0 then
        LCDOUT $FE,1, "TECLA:"
        LCDOUT $FE,$C0,"0 / P / E / R"
        PAUSE 1000
    endif
    if TRES = 0 then
        LCDOUT $FE,1, "TECLA:"
        LCDOUT $FE,$C0,"#"
        PAUSE 1000
    endif
    if CUATRO = 0 then
        LCDOUT $FE,1, "TECLA:"
        LCDOUT $FE,$C0,"HEX D"
        PAUSE 1000
    endif
    PAUSE 10
next
```

```
LCDOUT $FE,1, " Paul Toapanta "  
LCDOUT $FE,$C0," TEST SERIAL "  
PAUSE 2000  
tris  
for y = 1 to 10  
    serout portc.6,T9600,["Test COM1: ",#Y]  
    serout portc.6,T9600,[%1101]  
    pause 500  
next  
goto inicio  
return  
test_display:  
for x = 1 to 200  
    'P  
    portb = %0001100  
    unidad = 0  
    DEcena = 1  
    Centena = 1  
    Miles = 1  
    pause 5  
next  
for x = 1 to 100  
    'P  
    portb = %0001100  
    unidad = 1  
    DEcena = 0  
    Centena = 1  
    Miles = 1  
    pause 5
```

```
'U
portb = %1000001
unidad = 0
DEcena = 1
Centena = 1
Miles = 1
pause 5
next
for x = 1 to 66
  'P
  portb = %0001100
  unidad = 1
  DEcena = 1
  Centena = 0
  Miles = 1
  pause 5
  'U
  portb = %1000001
  unidad = 1
  DEcena = 0
  Centena = 1
  Miles = 1
  pause 5
  'C
  portb = %1000110
  unidad = 0
  DEcena = 1
  Centena = 1
  Miles = 1
```

```
    pause 5
next
'ESTATICO
for x = 1 to 50
    'P
    portb = %0001100
    unidad = 1
    DEcena = 1
    Centena = 1
    Miles = 0
    pause 5
    'U
    portb = %1000001
    unidad = 1
    DEcena = 1
    Centena = 0
    Miles = 1
    pause 5
    'C
    portb = %1000110
    unidad = 1
    DEcena = 0
    Centena = 1
    Miles = 1
    pause 5
    'E
    portb = %0000110
    unidad = 0
    DEcena = 1
```

```
Centena = 1
Miles = 1
pause 5
next
for x = 1 to 66
  'U
  portb = %1000001
  unidad = 1
  DEcena = 1
  Centena = 1
  Miles = 0
  pause 5
  'C
  portb = %1000110
  unidad = 1
  DEcena = 1
  Centena = 0
  Miles = 1
  pause 5
  'E
  portb = %0000110
  unidad = 1
  DEcena = 0
  Centena = 1
  Miles = 1
  pause 5
next
for x = 1 to 100
  'C
```

```
portb = %1000110
unidad = 1
DEcena = 1
Centena = 1
Miles = 0
pause 5
'E
portb = %0000110
unidad = 1
DEcena = 1
Centena = 0
Miles = 1
pause 5
next
for x = 1 to 200
'E
portb = %0000110
unidad = 1
DEcena = 1
Centena = 1
Miles = 0
pause 5
next
unidad = 1
DEcena = 1
Centena = 1
Miles = 1
PAUSE 2000
return
```

4.4 Validación de Resultados

Una vez que se ha obtenido como resultado final el Entrenador basado en microcontroladores PIC, se realizó una encuesta de satisfacción (Ver anexo 2) a los beneficiarios directos, quienes son los estudiantes de segundo nivel de la escuela de Sistemas de la PUCESA, en el laboratorio de electrología de un total de 8 alumnos. A continuación se muestran los resultados:

1. ¿El entrenador universal ha cumplido con sus expectativas?

Tabla 4.4: Cumplimiento de expectativas

Alternativas	Frecuencia	Porcentaje
Totalmente	7	87,5
En su mayor parte	1	12,5
No ha cumplido	0	0,0
Total	8	100,0

Fuente: Encuesta
Elaborado por: Paúl Toapanta

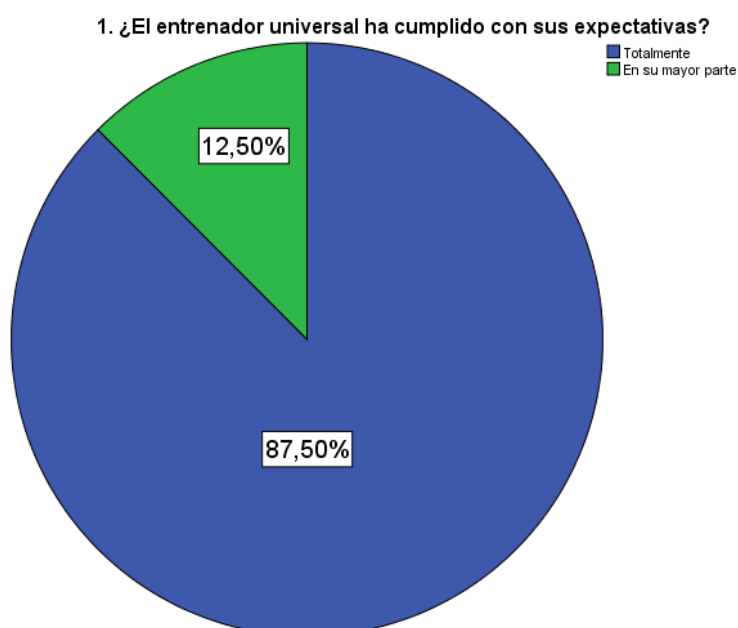


Ilustración 4.92: Cumplimiento de expectativas

Fuente: Encuesta
Elaborado por: Paúl Toapanta

Análisis

La mayoría de los encuestados manifiestan que el entrenador universal PIC ha cumplido con sus expectativas, lo que quiere decir que el entrenador está de acuerdo a los intereses de los estudiantes.

2. ¿Piensa Ud. que el entrenador universal le ayudaría a comprobar el buen funcionamiento de sus proyectos?

Tabla 4.5: Comprobación del funcionamiento de proyectos

Alternativas	Frecuencia	Porcentaje
Totalmente	7	87,5
En su mayor parte	1	12,5
No ayudaría	0	0,0
Total	8	100,0

Fuente: Encuesta
Elaborado por: Paúl Toapanta

2. ¿Piensa Ud. que el entrenador universal le ayudaría a comprobar el buen funcionamiento de sus proyectos?

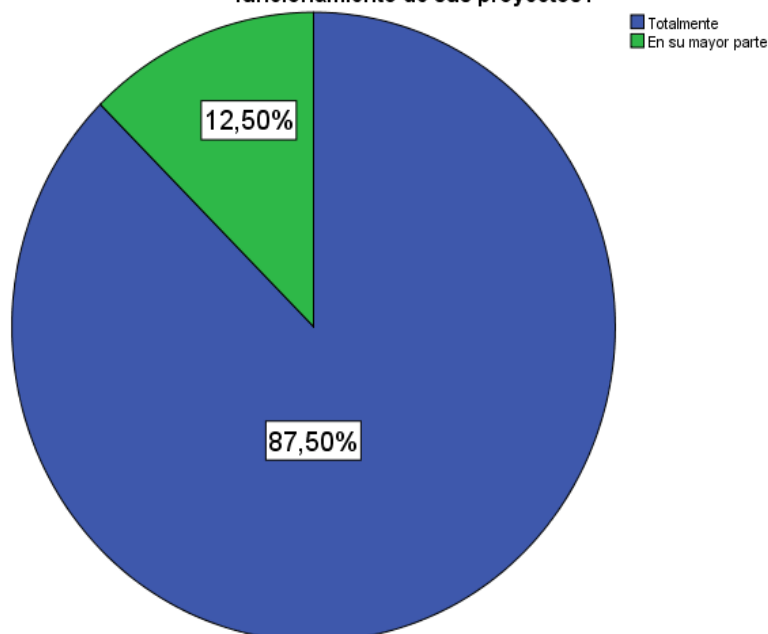


Ilustración 4.93: Comprobación del funcionamiento de proyectos

Fuente: Encuesta
Elaborado por: Paúl Toapanta

Análisis

La mayoría de los encuestados consideran que el entrenador le ayudaría a comprobar el buen funcionamiento de sus proyectos, lo que quiere decir que para los estudiantes es útil el trabajo realizado.

3. ¿Considera Ud. que la utilización del entrenador universal es simple y entendible?

Tabla 4.6: Utilización del entrenador simple y entendible

Alternativas	Frecuencia	Porcentaje
Totalmente	8	100,0
En su mayor parte	0	0,0
No es simple ni entendible	0	0,0

Fuente: Encuesta
Elaborado por: Paúl Toapanta

3. ¿Considera Ud. que la utilización del entrenador universal es simple y entendible?

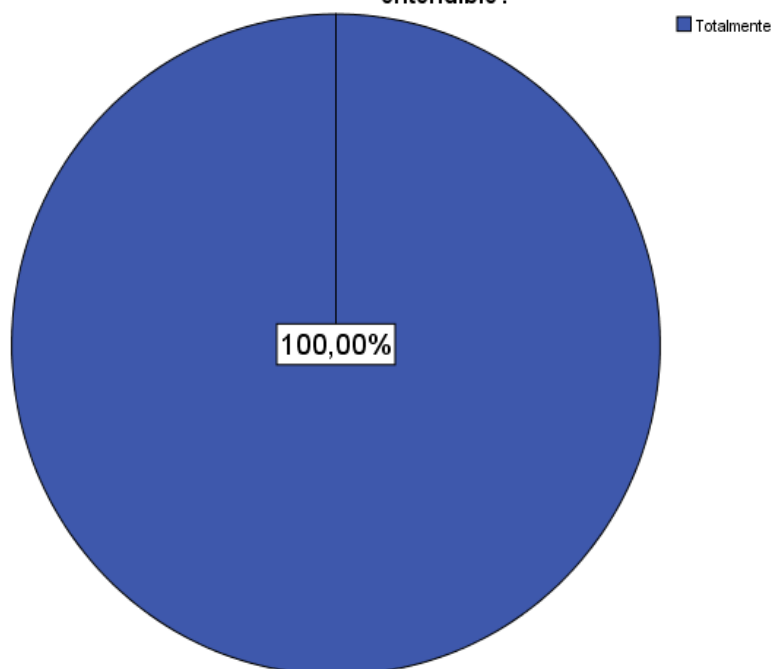


Ilustración 4.94: Utilización del entrenador simple y entendible

Fuente: Encuesta
Elaborado por: Paúl Toapanta

Análisis

La totalidad de los encuestados consideran que la utilización del entrenador es simple y entendible, lo que quiere decir que podrán utilizarlo sin ningún problema en el desarrollo de sus proyectos.

4. ¿Las funciones que proporciona el entrenador universal, se adaptan a las necesidades existentes en el Laboratorio de Electrología?

Tabla 4.7: Adaptación de la funciones del entrenador a las necesidades

Alternativas	Frecuencia	Porcentaje
Totalmente	6	75,0
En su mayor parte	2	25,0
No se adaptan	0	0,0
Total	8	100,0

Fuente: Encuesta
Elaborado por: Paúl Toapanta

4. ¿Las funciones que proporciona el entrenador universal, se adapta a las necesidades existentes en el Laboratorio de Electrología?

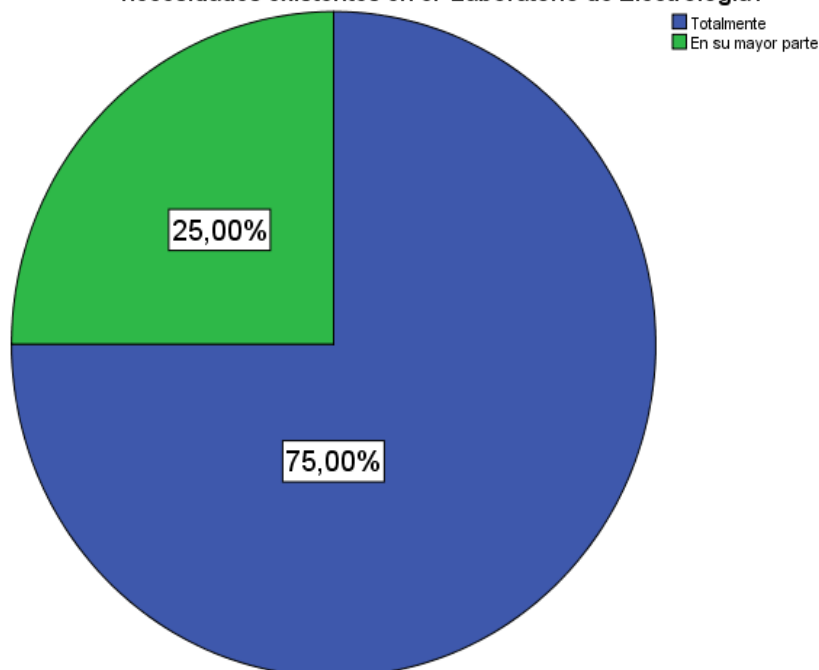


Ilustración 4.95: Adaptación de las funciones del entrenador a las necesidades

Fuente: Encuesta
Elaborado por: Paúl Toapanta

Análisis

La mayoría de los encuestados manifiestan que las funciones que proporciona el entrenador universal se adaptan a las necesidades existentes en el laboratorio de electrología, lo que quiere decir que este contribuirá a suplir las necesidades existentes y ayudará a mejorar el desempeño de los estudiantes.

5. Cree Ud. que el uso del entrenador universal para el desarrollo de sus proyectos le ayudaría a evitar la pérdida de tiempo y dinero?

Tabla 4.8: Ayuda para evitar pérdida de tiempo y dinero

Alternativas	Frecuencia	Porcentaje
Totalmente	6	75,0
En su mayor parte	2	25,0
No ayudaría	0	0,0
Total	8	100,0

Fuente: Encuesta
Elaborado por: Paúl Toapanta

5. Cree Ud. que el uso del entrenador universal para el desarrollo de sus proyectos le ayudaría a evitar la pérdida de tiempo y dinero?

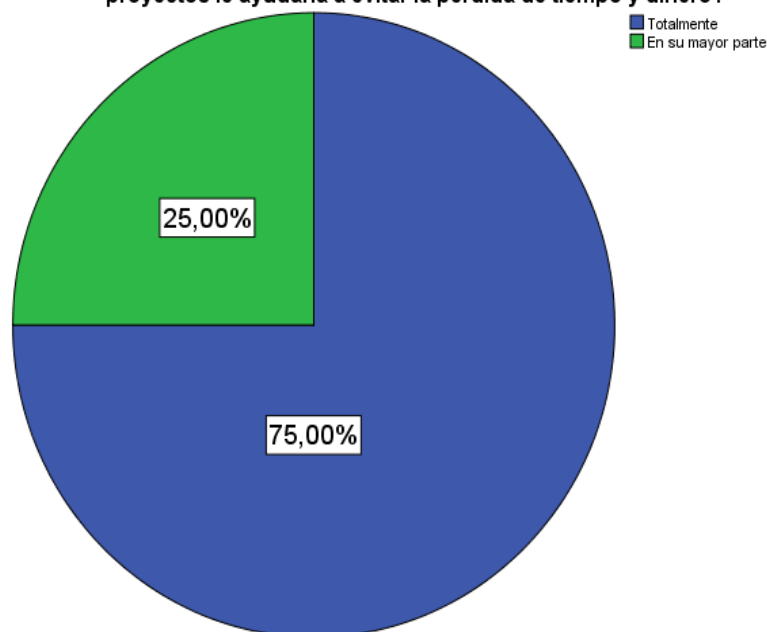


Ilustración 4.96: Ayuda para evitar pérdida de tiempo y dinero

Fuente: Encuesta
Elaborado por: Paúl Toapanta

Análisis

La mayoría de los encuestados creen que el uso del entrenador universal ayudaría a evitar la pérdida de tiempo y dinero, lo que quiere decir que este trabajo contribuirá a disminuir la pérdida de recursos.

6. ¿Cuál es la calificación que le asignaría al entrenador universal?

Tabla 4.9: Calificación del entrenador universal

Alternativas	Frecuencia	Porcentaje
Excelente	7	87,5
Bueno	1	12,5
Regular	0	0,0
Malo	0	0,0
Total	8	100,0

Fuente: Encuesta
Elaborado por: Paúl Toapanta

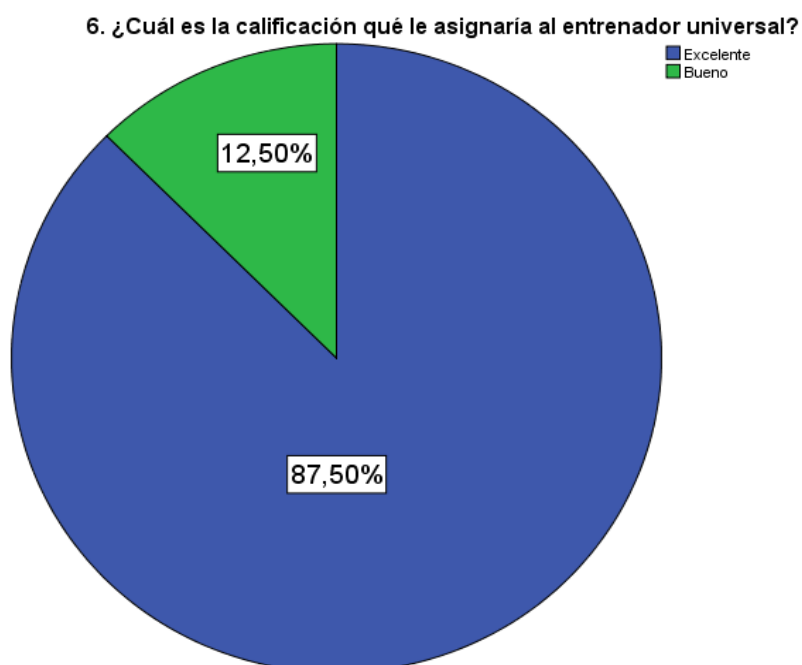


Ilustración 4.97: Calificación del entrenador universal

Fuente: Encuesta
Elaborado por: Paúl Toapanta

Análisis

La mayoría de los encuestados le asignaron la calificación de excelente al entrenador universal, lo que quiere decir que cumple con las expectativas y cubre las necesidades de los estudiantes del laboratorio de electrología.

4.4.1 Análisis General

Una vez analizados los resultados de la encuesta, se puede observar que éstos son altamente positivos ya que los estudiantes que utilizan el laboratorio de electrología, consideran que el entrenador universal es de fácil uso. Esta situación les permitirá realizar sus proyectos evitando posibles pérdidas de tiempo y dinero, ya que cubre varias de las necesidades que actualmente se presentan en el laboratorio.

CONCLUSIONES Y RECOMENDACIONES

Conclusiones

Para el diseño del circuito de la placa entrenadora fue necesario recurrir a un software de componentes electrónicos, por lo cual se utilizó para el desarrollo del trabajo el programa Altium Designer, ya que este permitió diseñar los esquemas electrónicos mismos que constan de periféricos de entrada y salida que son importantes para el manejo de dispositivos de forma autónoma e interactuar con el mismo, además la utilización de este programa permite hacer un análisis del funcionamiento de cada componente para posteriormente corregir errores y obtener el resultado final.

La placa acepta tres tipos de microcontroladores de 40, 28 y 18 pines, los cuales pueden manejar los periféricos según su capacidad, para lo cual se integró periféricos de entrada y salida como: display de 16x2, diodos led, pulsadores, teclado 4x4, bluetooth, puerto serial, potenciómetros y relé. La ventaja de este entrenador es que se puede utilizar módulos de una manera autónoma o en forma asistida con otros periféricos, utilizando la rama de ingeniería en sistemas se tiene la posibilidad de utilizar lenguajes de programación al gusto o la necesidad de quien lo realiza, teniendo como único requisito conocer los componentes que se van a utilizar para los proyectos.

Para la construcción del entrenador se determinó en primer lugar los elementos electrónicos necesarios como: la baquelita, transistores, cristales, max232 para

comunicación serial, resistencias, entre otros, y herramientas como: taladro, brocas, cautín, estaño; con todos estos elementos se pudo realizar la fase de soldadura y posteriormente realizar las pruebas, dando como resultado el Entrenador Universal PIC, mismo que ofrece una alternativa didáctica para el aula de electrología de la Escuela de Ingeniería en Sistemas de la PUCESA.

Finalmente, los estudiantes del Laboratorio de Electrología quienes son los beneficiarios directos del trabajo realizado, piensan que el entrenador universal les ayudaría a comprobar el buen funcionamiento de sus proyectos, por lo que evitaría la pérdida de tiempo y dinero, además consideran que es de fácil uso y que sus funciones se adaptan a las necesidades existentes en el laboratorio, por todo ello cumple con sus expectativas y le asignaron la calificación de excelente.

Recomendaciones

Algo muy importante que se debe conocer es que no se puede colocar más de un PIC en el entrenador ya que se puede tener reacciones desfavorables para la placa.

Se recomienda analizar y comprender un poco el funcionamiento de elementos que se vayan a conectar adicionalmente al entrenador, sea el caso de protoboards, PICs de otro modelo o tipo que no sea de MICROCHIP, porque pics de otra marca pueden tener arquitecturas diferentes y pueden dar resultados no deseados.

Al conectar al computador debe utilizar el cable Serial to USB adjunto como accesorio, y se puede utilizar cualquier software de su agrado para hacer pruebas con el modulo Serial y verificar que el entrenador se encuentre en la posición correcta para utilizarlo.

No se debe sobrepasar los niveles de corriente de entrada, para esto se adjunta su respectivo adaptador de corriente el cual es de salida de 5.2V y 625mA.

En el módulo del potenciómetro, no se puede utilizar también el módulo display, porque comparten pistas y puede ocasionar cortocircuito.

BIBLIOGRAFÍA

- Angulo, J. (2010). *Microcontroladores "PIC": diseño práctico de aplicaciones* (Segunda ed.). México: McGraw-Hill.
- Benchimol, D. (2011). *Electrónica Práctica*. Argentina: Fox Andina.
- García, E. (2012). *Compilador C CCS y simulador Proteus para microcontroladores PIC*. Barcelona: Alfaomega Grupo Editor.
- León, N. (2011). *Lenguaje de Programación para Microcontroladores*. España: McGrawHill.
- López, E. (2011). *Microcontroladores PIC: Sistema integrado para el autoaprendizaje*. Barcelona: Ediciones Técnicas.
- Orduña, J. (2012). *Arquitectura y Programación de Microprocesadores*. España: LLORÉNS.
- Palacios, E. (2013). *Microcontrolador PIC* (Tercera ed.). México: RA-MA S.A., Editorial y Publicaciones.
- Perez, E., Menendez, L., Fernandez, L., & López, E. (2010). *MICROCONTROLADORES PIC. Sistema integrado para el autoaprendizaje*. Barcelona: Marcobo.
- Ramirez, E. (2013). *Introducción a los Microcontroladores Equipo y Sistemas*. México: Limusa.
- Reyes, C. (2011). *Microcontroladores PIC programación en Basic*. Quito: Rispergraf.
- Riba, C., & Molina, A. (2006). *Ingeniería Concurrente: Una metodología integradora*. Barcelona: RETDIC.
- Romero, S. (2011). *El mundo de los microprocesadores*. España: MARCOMBO, SA.

Rossano, V. (2012). *Electrónica y Microcontroladores PIC*. México: USERS.

Sánchez, F. (2011). *Contrucción de Microprocesadores*. España: Limusa.

Santamaría, E. (2011). *Microprocesador y Microcontrolador*. Madrid: Biblos
Industria Gráfica, SL.

Stefan, W. (2010). *Microcontroladores PIC Prácticas de Programación*. Madrid:
Marcombo.

Valdés, F. (2011). *Microcontroladores: Fundamentos y Aplicaciones con PIC*.
España: Carles Parcerisas Civit.

ANEXOS

Anexo 1: PIC16F877

El Microcontrolador PIC16F877

Resumen de hoja de datos

El PIC16F877 es un microcontrolador con memoria de programa tipo FLASH, lo que representa gran facilidad en el desarrollo de prototipos y en su aprendizaje ya que no se requiere borrarlo con luz ultravioleta como las versiones EPROM, sino que permite reprogramarlo nuevamente sin ser borrado con anterioridad.

El PIC16F877 es un microcontrolador de Microchip Technology fabricado en tecnología CMOS, su consumo de potencia es muy bajo y además es completamente estático, esto quiere decir que el reloj puede detenerse y los datos de la memoria no se pierden.

El encapsulado más común para este microcontrolador es el DIP (Dual In-line Pin) de 40 pines, propio para usarlo en experimentación. La referencia completa es PIC16F877-04 para el dispositivo que utiliza cristal oscilador de hasta 4 MHz, PIC16F877-20 para el dispositivo que utiliza cristal oscilador de hasta 20 MHz o PIC16F877A-I para el dispositivo tipo industrial que puede trabajar hasta a 20 MHz. Sin embargo, hay otros tipos de encapsulado que se pueden utilizar según el diseño y la aplicación que se quiere realizar. Por ejemplo, el encapsulado tipo surface mount (montaje superficial) tiene un reducido tamaño y bajo costo, que lo hace propio para producciones en serie o para utilizarlo en lugares de espacio muy reducido.

Configuración de pines

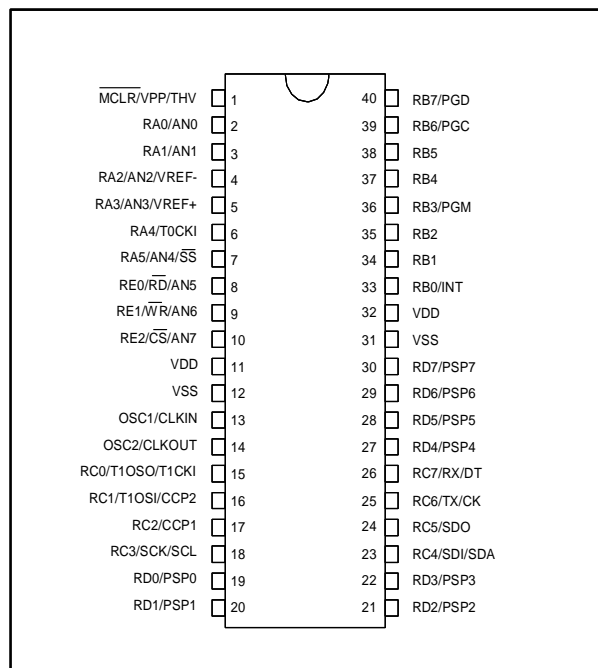


Figura 3.1. Distribución de pines del PIC16F877.

Los pines de entrada/salida de este microcontrolador están organizados en cinco puertos, el puerto A con 6 líneas, el puerto B con 8 líneas, el puerto C con 8 líneas, el puerto D con 8 líneas y el puerto E con 3 líneas. Cada pin de esos puertos se puede configurar como entrada o como salida independiente programando un par de registros diseñados para tal fin. En ese registro un bit en "0" configura el pin del puerto correspondiente como salida y un bit en "1" lo configura como entrada. Dichos pines del microcontrolador también pueden cumplir otras funciones especiales, siempre y cuando se configuren para ello, según se verá más adelante.

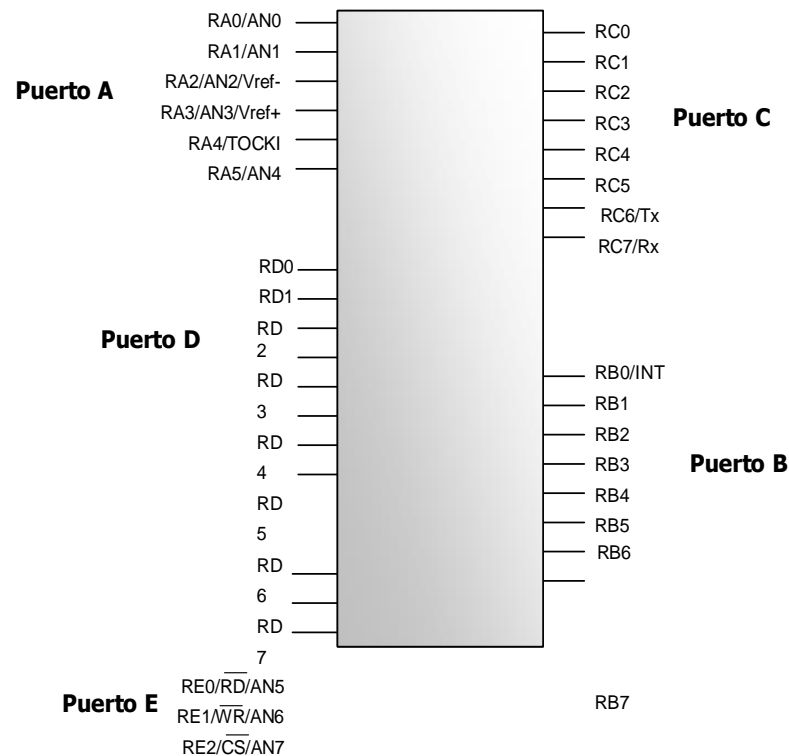


Figura 3.2. Distribución de los puertos del PIC16F877

Los pines del puerto A y del puerto E pueden trabajar como entradas para el convertidor Análogo a Digital interno, es decir, allí se podría conectar una señal proveniente de un sensor o de un circuito analógico para que el microcontrolador la convierta en su equivalente digital y pueda realizar algún proceso de control o de instrumentación digital. El pin RB0/INT se puede configurar por software para que funcione como interrupción externa, para configurarlo se utilizan unos bits de los registros que controlan las interrupciones.

El pin RA4/TOCKI del puerto A puede ser configurado como un pin de entrada/salida o como entrada del temporizador/contador. Cuando este pin se programa como entrada digital,

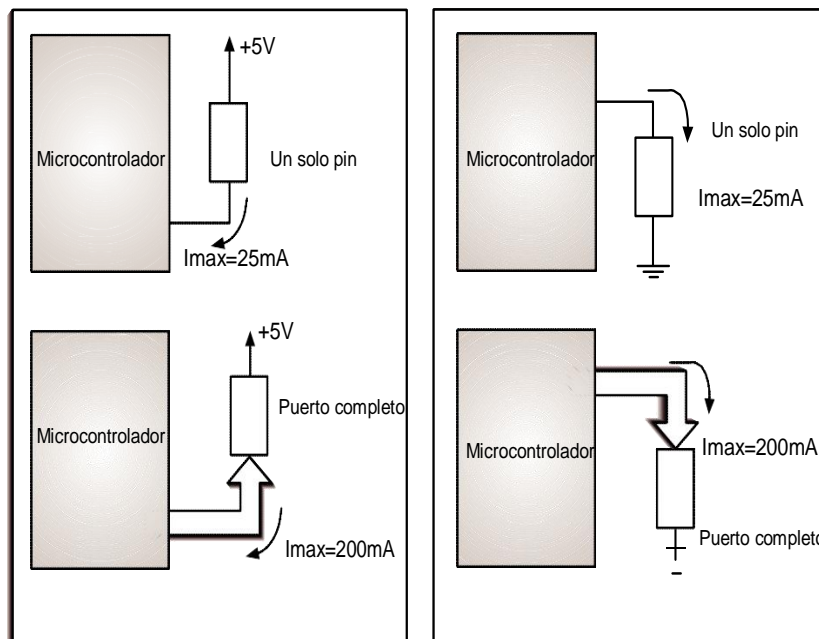
funciona como un disparador de Schmitt (Schmitt trigger), puede reconocer señales un poco distorsionadas y llevarlas a niveles lógicos (cero y cinco voltios). Cuando se usa como salida digital se comporta como colector abierto (open collector), por lo tanto, se debe poner una resistencia de pull-up (resistencia externa conectada a un nivel de cinco voltios). Como salida, la lógica es inversa: un "0" escrito al pin del puerto entrega en el pin un "1" lógico. Además, como salida no puede manejar cargas como fuente, sólo en el modo sumidero.

El puerto E puede controlar la conexión en modo microprocesador con otros dispositivos utilizando las líneas RD (read), WR (write) y CS (chip select). En este modo el puerto D funciona como un bus de datos de 8 bits (pines PSP).

La máxima capacidad de corriente de cada uno de los pines de los puertos en modo sumidero (sink) o en modo fuente (source) es de 25 mA . La máxima capacidad de corriente total de los puertos es:

	PUERTO A	PUERTO B	PUERTO C	PUERTO D
Modo sumidero	150 mA	200 mA	200 mA	200 mA
Modo fuente	150 mA	200 mA	200mA	200mA

El consumo de corriente del microcontrolador para su funcionamiento depende del voltaje de operación, la frecuencia y de las cargas que tengan sus pines. Para un oscilador de 4 MHz el consumo es de aproximadamente 2 mA; aunque este se puede reducir a 40 microamperios cuando se está en el modo sleep (en este modo el micro se detiene y disminuye el consumo de potencia). Se sale de ese estado cuando se produce alguna condición especial que veremos más adelante.



Descripción de los pines del microcontrolador

Nombre pin	Pin	Descripción
RA0/AN0	2	E/S Digital o Entrada análoga 0.
RA1/AN1	3	E/S Digital o Entrada análoga 1.
RA2/AN2 V_{ref}^-	4	E/S Digital o Entrada análoga 2.
RA3/AN3 V_{ref}^+	5	E/S Digital o Entrada análoga 3.
RA4/TOCKI	6	Bit 4 del puerto A (E/S bidireccional). También se usa como entrada de reloj al temporizador/contador TMR0. Salida de colector abierto.
RA5/SS/AN4	7	E/S Digital o Entrada análoga 4. También lo usa el puerto serial síncrono.
RB0/INT	33	Bit 0 del puerto B (E/S bidireccional). Buffer E/S: TTL/ST. También se usa como entrada de interrupción externa (INT).
RB1	34	Bit 1 del puerto B (E/S bidireccional). Buffer E/S: TTL
RB2	35	Bit 2 del puerto B (E/S bidireccional). Buffer E/S: TTL
RB3/PGM	36	Bit 3 del puerto B (E/S bidireccional). Buffer E/S: TTL (Programación en bajo voltaje)
RB4	37	Bit 4 del puerto B (E/S bidireccional). Buffer E/S: TTL. Interrupción por cambio del pin.
RB5	38	Bit 5 del puerto B (E/S bidireccional). Buffer E/S: TTL. Interrupción por cambio del pin.
RB6/PGC	39	Bit 6 del puerto B (E/S bidireccional). Buffer E/S: TTL/ST. Interrupción por cambio del pin. Entrada de reloj para programación serial.
RB7/PGD	40	Bit 7 del puerto B (E/S bidireccional). Buffer E/S: TTL/ST. Interrupción por cambio del pin. Entrada de datos para programación serial.
RC0/T1OSO/T1CKI	15	E/S Digital. Salida del oscilador Timer 1 o entrada de reloj Timer 1.
RC1/T1OSI/CCP2	16	E/S Digital. Entrada del oscilador Timer 1. Entrada Captura 2; Salida Compara 2; Salida PWM 2
RC2/CCP1	17	E/S Digital. Entrada Captura 1; Salida Compara 1; Salida PWM 1
RC3/SCK/SCL	18	E/S Digital. Línea de reloj serial asíncrono en el modo SPI y el modo I ² C
RC4/SDI/SDA	23	E/S Digital. Línea de datos en el modo SPI o en el modo I ² C
RC5/SDO	24	E/S Digital.
RC6/TX/CK	25	E/S Digital. Transmisión asíncrona (USART) o reloj síncrono (SSP).
RC7/RX/DT	26	E/S Digital. Recepción asíncrona (USART) o línea de datos (SSP).
V_{DD}	11,32	Voltaje de alimentación DC (+)
V_{SS}	12,31	Referencia de voltaje (GND).
MCLR	1	Entrada de RESET al microcontrolador. Voltaje de entrada durante la programación. En nivel bajo resetea el microcontrolador.
OSC1/CLKIN	13	Entrada oscilador cristal oscilador / Entrada fuente de reloj externa.
OSC2/CLKOUT	14	Salida oscilador cristal. Oscilador RC: Salida con un ¼ frecuencia OSC1
RD0/PSP0	19	E/S Digital. Puede ser puerto paralelo en bus de 8 bits.
RD1/PSP1	20	E/S Digital. Puede ser puerto paralelo en bus de 8 bits.
RD2/PSP2	21	E/S Digital. Puede ser puerto paralelo en bus de 8 bits.
RD3/PSP3	22	E/S Digital. Puede ser puerto paralelo en bus de 8 bits.
RD4/PSP4	27	E/S Digital. Puede ser puerto paralelo en bus de 8 bits.
RD5/PSP5	28	E/S Digital. Puede ser puerto paralelo en bus de 8 bits.
RD6/PSP6	29	E/S Digital. Puede ser puerto paralelo en bus de 8 bits.
RD7/PSP7	30	E/S Digital. Puede ser puerto paralelo en bus de 8 bits.
RE0/RD/AN5	8	E/S Digital. Puede ser pin de lectura (read) en modo microprocesador.
RE1/WR/AN6	9	E/S Digital. Puede ser pin de escritura (write) en modo microprocesador.
RE2/CS/AN7	10	E/S Digital. Puede ser pin de selección de chip (chip select) en modo microprocesador.

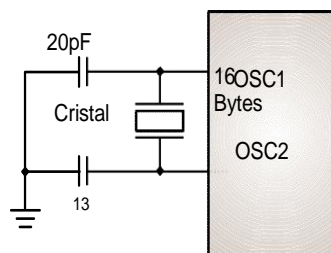
El oscilador externo

Todo microcontrolador requiere un circuito externo que le indique la velocidad a la que debe trabajar. Este circuito, que se conoce como oscilador o reloj, es muy simple pero de vital importancia para el buen funcionamiento del sistema. El PIC16F877 puede utilizar cuatro tipos de oscilador diferentes. Estos tipos son:

- **RC.** Oscilador con resistencia y condensador.
- **XT.** Cristal (por ejemplo de 1 a 4 MHz).
- **HS.** Cristal de alta frecuencia (por ejemplo 10 a 20 MHz).
- **LP.** Cristal para baja frecuencia y bajo consumo de potencia.

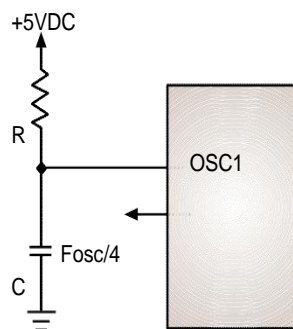
En el momento de programar o "quemar" el microcontrolador se debe especificar que tipo de oscilador se usa. Esto se hace a través de unos fusibles llamados "fusibles de configuración".

El tipo de oscilador que se sugiere para las prácticas es el XT con un cristal de 4 MHz, porque garantiza precisión y es muy comercial. Internamente esta frecuencia es dividida por cuatro, lo que hace que la frecuencia efectiva de trabajo sea de 1 MHz en este caso, por lo que cada instrucción se ejecuta en un microsegundo. El cristal debe ir acompañado de dos condensadores y se conecta como se muestra en la figura 3.4.



Conexión de un oscilador XT.

Si no se requiere mucha precisión en el oscilador y se quiere economizar dinero, se puede utilizar una resistencia y un condensador, como se muestra en la figura 3.5.



Conexión de un oscilador RC.

Reset

En los microcontroladores se requiere un pin de reset para reiniciar el funcionamiento del sistema cuando sea necesario, ya sea por una falla que se presente o porque así se haya diseñado el sistema. El pin de reset en los PIC es llamado MCLR (master clear). Existen varias formas de resetear o reiniciar el sistema:

- Al encendido (Power On Reset)
- Pulso en el pin MCLR durante operación normal
- Pulso en el pin MCLR durante el modo de bajo consumo (modo sleep)
- El rebase del conteo del circuito de vigilancia (watchdog) durante operación normal
- El rebase del conteo del circuito de vigilancia (watchdog) durante el modo de bajo consumo (sleep)

El reset al encendido se consigue gracias a dos temporizadores. El primero de ellos es el OST (Oscillator Start-Up Timer: Temporizador de encendido del oscilador), orientado a mantener el microcontrolador en reset hasta que el oscilador del cristal es estable. El segundo es el PWRT (Power-Up Timer: Temporizador de encendido), que provee un retardo fijo de 72 ms (nominal) en el encendido únicamente, diseñado para mantener el dispositivo en reset mientras la fuente se estabiliza. Para utilizar estos temporizadores, sólo basta con conectar el pin MCLR a la fuente de alimentación, evitándose utilizar las tradicionales redes de resistencias externas en el pin de reset.

El reset por MCLR se consigue llevando momentáneamente este pin a un estado lógico bajo, mientras que el watchdog WDT produce el reset cuando su temporizador rebasa la cuenta, o sea que pasa de 0FFh a 00h. Cuando se quiere tener control sobre el reset del sistema se puede conectar un botón como se muestra en la figura 3.6.

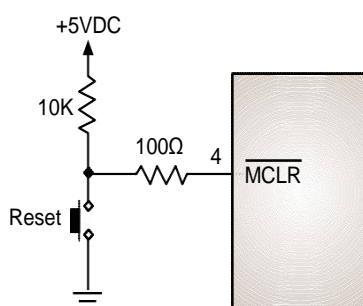
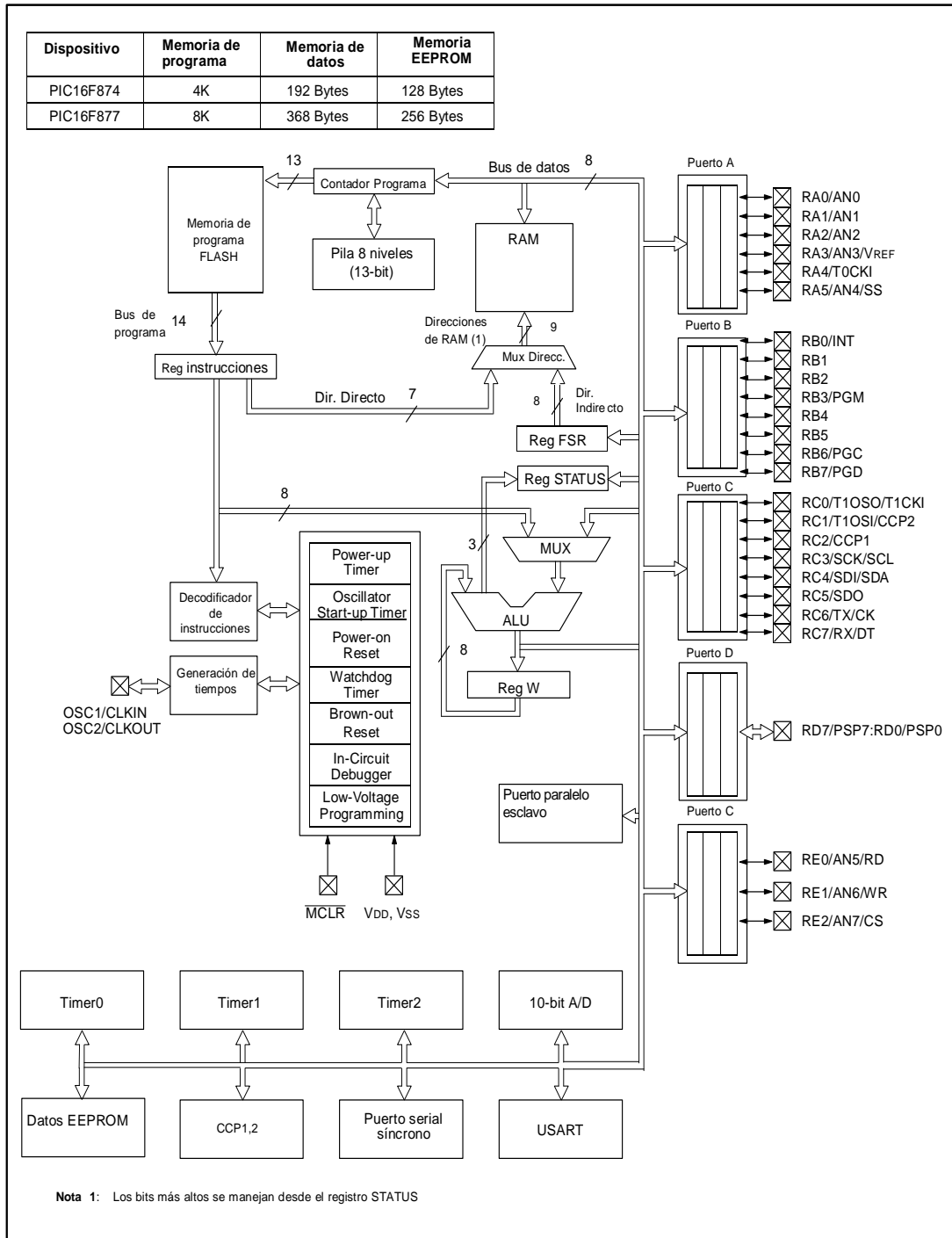


Figura 3.6 Conexión del botón de reset.

Arquitectura interna del microcontrolador

Este término se refiere a los bloques funcionales internos que conforman el microcontrolador y la forma en que están conectados, por ejemplo la memoria FLASH (de programa), la memoria RAM (de datos), los puertos, la lógica de control que permite que todo el conjunto funcione, etc.



Arquitectura del PIC16F877

La figura 3.7 muestra la arquitectura general del PIC16F877, en ella se pueden apreciar los diferentes bloques que lo componen y la forma en que se conectan. Se muestra la conexión de los puertos, las memorias de datos y de programa, los bloques especiales como el watchdog, los temporizadores de arranque, el oscilador, etc.

Todos los elementos se conectan entre sí por medio de buses. Un bus es un conjunto de líneas que transportan información entre dos o más módulos. Vale la pena destacar que el PIC16F877 tiene un bloque especial de memoria de datos de 256 bytes del tipo EEPROM, además de los dos bloques de memoria principales que son el de programa y el de datos o registros.

El PIC16F877 se basa en la arquitectura Harvard, en la cual el programa y los datos se pueden trabajar con buses y memorias separadas, lo que posibilita que las instrucciones y los datos posean longitudes diferentes. Esta misma estructura es la que permite la superposición de los ciclos de búsqueda y ejecución de las instrucciones, lo cual se ve reflejado en una mayor velocidad del microcontrolador.

Memoria de programa (FLASH)

Es una memoria de 8K de longitud con datos de 14 bits en cada posición. Como es del tipo FLASH se puede programar y borrar eléctricamente, lo que facilita el desarrollo de los programas y la experimentación. En ella se graba o almacena el programa o códigos que el microcontrolador debe ejecutar. En la figura 3.8 se muestra el mapa de la memoria de programa.

La memoria de programa está dividida en cuatro bancos o páginas de 2K cada uno. El primero va de la posición de memoria 0000h a la 07FFh, el segundo va de la 0800h a la 0FFFh, el tercero de la 1000h a la 17FFh y el cuarto de la 1800h a la 1FFFh.

Vector de reset. Cuando ocurre un reset al microcontrolador, el contador de programa se pone en ceros (0000H). Por esta razón, en la primera dirección del programa se debe escribir todo lo relacionado con la iniciación del mismo.

Vector de interrupción. Cuando el microcontrolador recibe una señal de interrupción, el contador de programa apunta a la dirección 04H de la memoria de programa, por eso, allí se debe escribir toda la programación necesaria para atender dicha interrupción.

Pila (Stack)

Estos registros no forman parte de ningún banco de memoria y no permiten el acceso por parte del usuario. Se usan para guardar el valor del contador de programa cuando se hace un llamado a una subrutina o cuando se atiende una interrupción; luego, cuando el micro regresa a seguir ejecutando su tarea normal, el contador de programa recupera su valor leyéndolo nuevamente desde la pila. El PIC16F877 tiene una pila de 8 niveles, esto significa que se pueden anidar 8

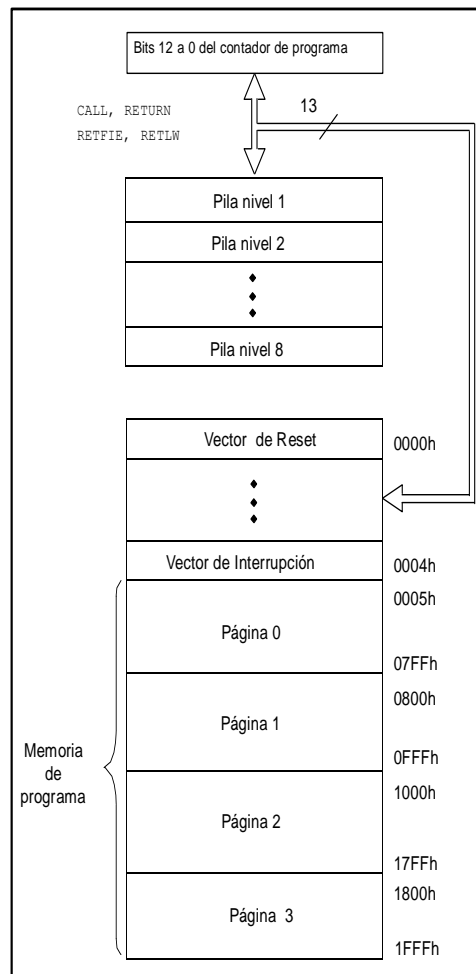


Figura 3.8. Memoria de programa del PIC16F877.

Memoria de datos (RAM)

El PIC16F877 posee cuatro bancos de memoria RAM, cada banco posee 128 bytes. De estos 128 los primeros 32 (hasta el 1Fh) son registros que cumplen un propósito especial en el control del microcontrolador y en su configuración. Los 96 siguientes son registros de uso general que se pueden usar para guardar los datos temporales de la tarea que se está ejecutando, figura 3.9.

Todas las posiciones o registros de memoria se pueden acceder directa o indirectamente (esta última forma a través del registro selector FSR). Para seleccionar que página o banco de memoria se trabaja en un momento determinado se utilizan los bits RP0 y RP1 del registro STATUS.

Resumen de algunos de los registros de configuración

BANCO 0:

- **TMR0:** Registro del temporizador/contador de 8 bits.
- **PCL:** Byte menos significativo del contador de programa (PC).
- **STATUS:** Contiene banderas (bits) que indican el estado del procesador después de una operación aritmética/lógica.
- **FSR:** Registro de direccionamiento indirecto.
- **PORTA, PORTB, PORTC, PORTD, PORTE:** Registro de puertos de E/S de datos. Conectan con los pines físicos del micro.
- **PCLATH:** Byte alto (más significativo) del contador de programa (PC).
- **INTCON:** Registro de control de las interrupciones.
- **ADRESH:** Parte alta del resultado de la conversión A/D.
- **ADCON0:** Controla la operación del módulo de conversión A/D

BANCO 1:

- **OPTION:** Registro de control de frecuencia del TMR0.
- **TRISA, TRISB, TRISC, TRISD. TRISE:** Registros de configuración de la operación de los pines de los puertos.
- **ADRESL:** Parte baja del resultado de la conversión A/D.
- **ADCON1:** Controla la configuración de los pines de entrada análoga.

BANCO 2:

- **TMR0:** Registro del temporizador/contador de 8 bits.
- **PCL:** Byte menos significativo del contador de programa (PC).
- **FSR:** Registro de direccionamiento indirecto.
- **EEDATA:** Registro de datos de la memoria EEPROM.
- **EEADR:** Registro de dirección de la memoria EEPROM.
- **PCLATH:** Byte alto (más significativo) del contador de programa (PC).
- **INTCON:** Registro de control de las interrupciones.

BANCO 3:

- **OPTION:** Registro de control de frecuencia del TMR0.
- **EECON1:** Control de lectura/escritura de la memoria EEPROM de datos.

EECON2: No es un registro físico.

INDF	00h	INDF	80h	INDF	100h	INDF	180h
TMR0	01h	OPTION_REG	81h	TMR0	101h	OPTION_REG	181h
PCL	02h	PCL	82h	PCL	102h	PCL	182h
STATUS	03h	STATUS	83h	STATUS	103h	STATUS	183h
FSR	04h	FSR TRISA	84h	FSR	104h	FSR	184h
PORTA	05h	TRISA	85h		105h		185h
PORTB	06h	TRISB	86h	PORTB	106h	TRISB	186h
PORTC	07h	TRISC	87h		107h		187h
PORTD	08h	TRISD	88h		108h		188h
PORTE	09h	TRISE	89h		109h		189h
PCLATH	0Ah	PCLATH	8Ah	PCLATH	10Ah	PCLATH	18Ah
INTCON	0Bh	INTCON	8Bh	INTCON	10Bh	INTCON	18Bh
PIR1	0Ch	PIE1	8Ch	EEDATA	10Ch	EECON1	18Ch
PIR2	0Dh	PIE2	8Dh	EEADR	10Dh	EECON2	18Dh
TMR1L	0Eh	PCON	8Eh	EEDATH	10Eh	Reservado	18Eh
TMR1H	0Fh		8Fh	EEADRH	10Fh	Reservado	18Fh
T1CON	10h		90h		110h		190h
TMR2	11h	SSPCON2	91h	Registros de Propósito General 16 Bytes		Registros de Propósito General 16 Bytes	
T2CON	12h	PR2	92h				
SSPBUF	13h	SSPADD	93h				
SSPCON	14h	SSPSTAT	94h				
CCPR1L	15h		95h				
CCPR1H	16h		96h				
CCP1CON	17h		97h				
RCSTA	18h	TXSTA	98h				
TXREG	19h	SPBRG	99h				
RCREG	1Ah		9Ah				
CCPR2L	1Bh		9Bh				
CCPR2H	1Ch		9Ch				
CCP2CON	1Dh		9Dh				
ADRESH	1Eh	ADRESL	9Eh				
ADCON0	1Fh	ADCON1	9Fh				
Registros de Propósito General 96 Bytes	20h	Registros de Propósito General 80 Bytes	A0h				
				120h	1A0h		
			0EFh 0F0h	16Fh 170h	1EFh 1F0h		
Banco 0	7Fh	Banco 1	FFh	Banco 2	17Fh	Banco 3	1FFh

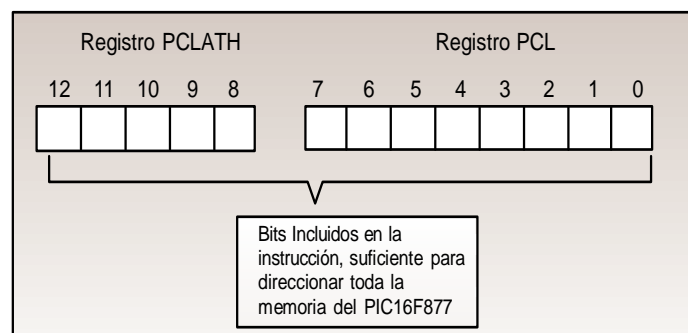
Organización de la memoria RAM del PIC16F877.

Función de algunos registros especiales

00h o INDF: Registro para direccionamiento indirecto de datos. Este no es un registro disponible físicamente; utiliza el contenido del registro FSR y los bits RP0 y RP1 del registro STATUS para seleccionar indirectamente la memoria de datos, la instrucción que lo acompañe determinará que se debe realizar con el registro señalado.

01h o TMR0. Temporizador/contador de 8 bits. Este es un contador que se puede incrementar con una señal externa aplicada al pin RA4/TOCKI o de acuerdo a una señal interna proveniente del reloj de instrucciones del microcontrolador. La tasa de incremento del registro se puede determinar por medio de un preescalador o divisor de frecuencia, localizado en el registro OPTION. Como una mejora con respecto a referencias anteriores, se le ha agregado la generación de interrupción cuando se rebasa la cuenta (el paso de 0FFh a 00h).

02h o PCL: Contador de programa. Se utiliza para direccionar las palabras de 14 bits del programa que se encuentra almacenado en la memoria ROM; este contador de programas es de 13 bits de ancho, figura 3.10. Sobre el byte bajo, se puede escribir o leer directamente, mientras que sobre el byte alto, no. El byte alto se maneja mediante el registro PCLATH (0Ah). Ante una condición de reset el microcontrolador inicia el contador de programa con todos sus bits en "cero". Durante la ejecución normal del programa, y dado que todas las instrucciones ocupan sólo una posición de memoria, el contador se incrementa en uno con cada instrucción, a menos que se trate de alguna instrucción de salto.



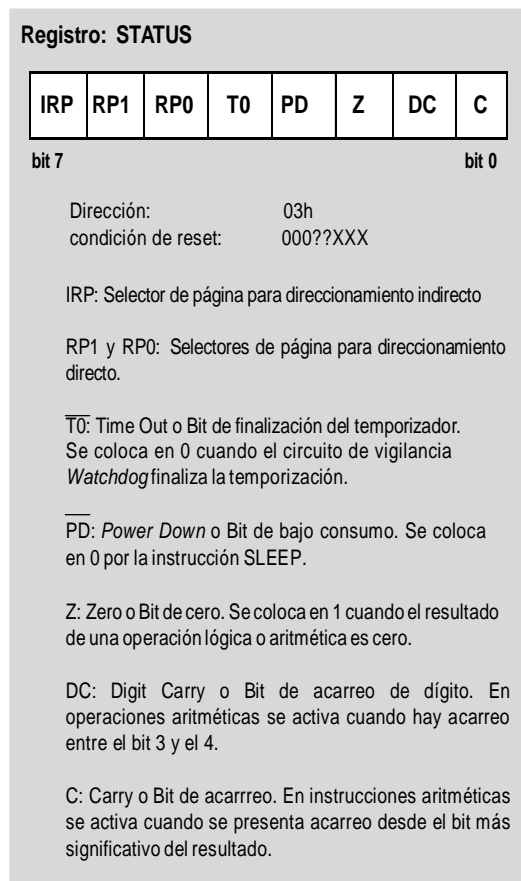
Registros que conforman el Contador de programa.

En una instrucción CALL o GOTO la dirección de memoria a donde se quiere ir, es decir, los bits 12 a 0 del contador de programa se cargan desde el código de operación de la instrucción.

En algunas instrucciones donde la parte baja del contador de programa, es decir los ocho bits bajos del registro PCL, es el destino, los otros cinco bits se cargan directamente desde el

PCLATH (bits 4 a 0), por ejemplo en el caso de la instrucción ADDWF. Esto se debe tener en cuenta cuando se desea hacer lectura de tablas usando el comando: ADDWF PC,1 , en este caso se debe tener en cuenta que la tabla debe estar comprendida dentro de un solo bloque de 256 bytes (0-255, 256-511, etc.).

03h o STATUS: Registro de estados. Contiene el estado aritmético de la ALU, la causa del reset y los bits de preselección de página para la memoria de datos. La figura 3.11 muestra los bits correspondientes a este registro. Los bits 5 y 6 (RP0 y RP1) son los bits de selección de página para el direccionamiento directo de la memoria de datos; solamente RP0 se usa en los PIC16F84. RP1 se puede utilizar como un bit de propósito general de lectura/escritura. Los bits T0 y PD no se pueden modificar por un proceso de escritura; ellos muestran la condición por la cual se ocasionó el último reset.



Bits del Registro de Estados.

04h o FSR: Registro selector de registros. En asocio con el registro INDO, se utiliza para seleccionar indirectamente los otros registros disponibles. Si en el programa no se utilizan llamadas indirectas, este registro se puede utilizar como un registro de propósito general.

Para entender mejor el funcionamiento de este registro veamos un programa simple que borra el contenido de la memoria RAM, empleando direccionamiento indirecto.

```

        MOVLW 20h      ;inicializa el puntero en la posición de memoria RAM
        MOVWF FSR      ;que se va a borrar
NEXT   CLRF   INDO     ;borra el registro indexado (es decir el que está
                        ;siendo direccionado por el FSR)
        INCF   FSR,1   ;incrementa el puntero
        BTFSS FSR,6    ;pregunta por el bit 6 para ver si ya acabó de borrar
        GOTO  NEXT     ;sigue borrando los registros que faltan
        continúa .....

```

05h o PORTA: Puerto de Entrada/Salida de 6 bits. Este puerto, al igual que todos sus similares en los PIC, puede leerse o escribirse como si se tratara de un registro cualquiera. El registro que controla el sentido (entrada o salida) de los pines de este puerto está localizado en la página 1, en la posición 85h y se llama TRISA. El puerto A también puede ser configurado para que trabaje como entradas análogas para el convertidor Análogo a Digital interno del microcontrolador.

06h o PORTB: Puerto de entrada/salida de 8 bits. Al igual que en todos los PIC, este puede leerse o escribirse como si se tratara de un registro cualquiera; algunos de sus pines tienen funciones alternas en la generación de interrupciones. El registro de control para la configuración de la función de sus pines se localiza en la página 1, en la dirección 86h y se llama TRISB. Puede ser configurado también para cumplir otras funciones.

07h o PORTC: Puerto de entrada/salida de 8 bits. Al igual que en todos los PIC, este puede leerse o escribirse como si se tratara de un registro cualquiera; algunos de sus pines tienen funciones alternas. El registro de control para la configuración de la función de sus pines se localiza en la página 1, en la dirección 87h y se llama TRISC. Puede ser configurado también para cumplir otras funciones.

08h o PORTD: Puerto de entrada/salida de 8 bits. Al igual que en todos los PIC, este puede leerse o escribirse como si se tratara de un registro cualquiera; algunos de sus pines tienen funciones alternas cuando se utiliza el micro en modo microprocesador. El registro de control para la configuración de la función de sus pines se localiza en la página 1, en la dirección 88h y se llama TRISD. Puede ser configurado también para cumplir otras funciones.

09h o PORTE: Puerto de Entrada/Salida de 3 bits. Este puerto, al igual que todos sus similares en los PIC, puede leerse o escribirse como si se tratara de un

registro cualquiera. El registro que controla el sentido (entrada o salida) de los pines de este puerto está localizado en la página 1, en la posición 89h y se llama TRISE. El puerto E también puede ser configurado para que trabaje como entradas análogas para el convertidor Análogo a Digital interno del microcontrolador o para que maneje las señales de control en el modo microprocesador.

85h o TRISA: Registro de configuración del puerto A. Como ya se mencionó, es el registro de control para el puerto A. Un "cero" en el bit correspondiente al pin lo configura como salida, mientras que un "uno" lo hace como entrada.

86h o TRISB: Registro de configuración del puerto B. Orientado hacia el control del puerto B. Son válidas las mismas consideraciones del registro TRISA.

87h o TRISC: Registro de configuración del puerto C. Orientado hacia el control del puerto C. Son válidas las mismas consideraciones del registro TRISA.

88h o TRISD: Registro de configuración del puerto D. Orientado hacia el control del puerto D. Son válidas las mismas consideraciones del registro TRISA.

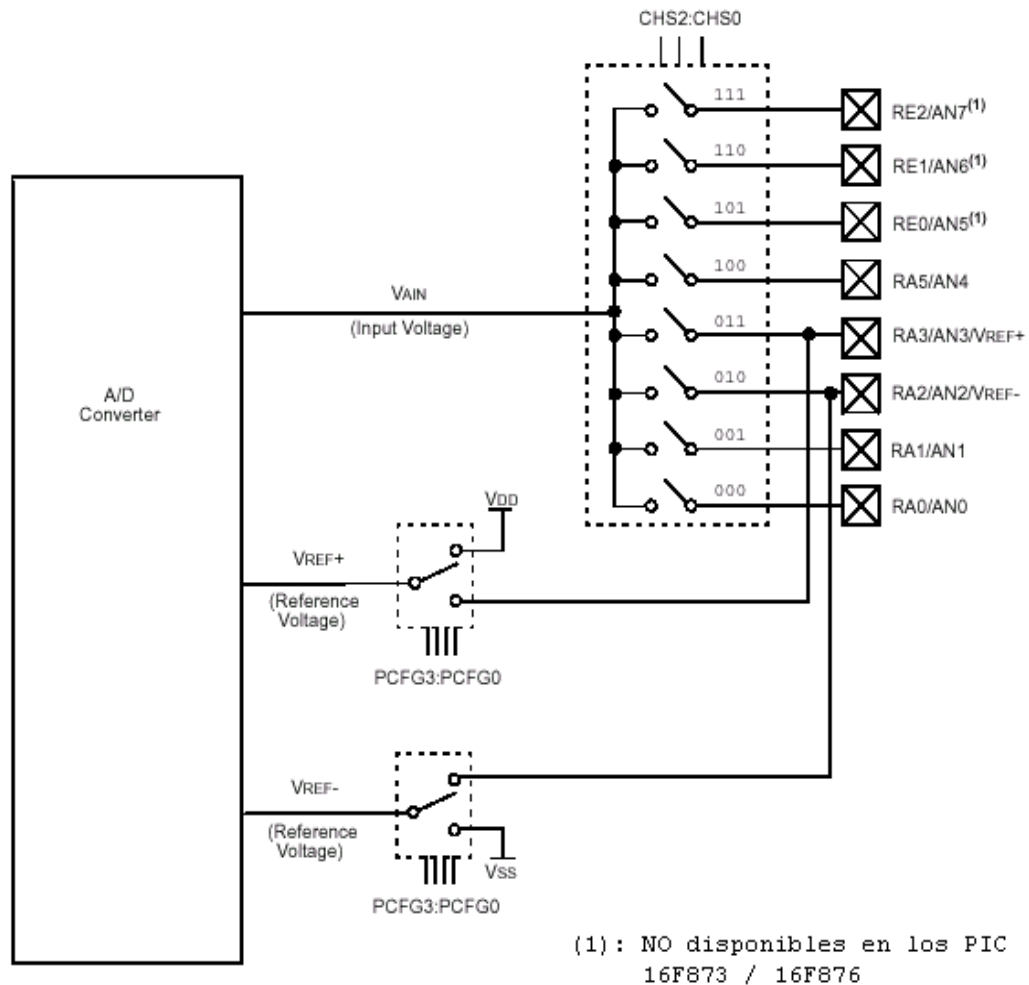
89h o TRISE: Registro de configuración del puerto E. Orientado hacia el control del puerto E. Son válidas las mismas consideraciones del registro TRISA.

020h a 7Fh: Registros de propósito general. Estas 96 posiciones están implementadas en la memoria RAM estática, la cual conforma el área de trabajo del usuario. Pueden ser utilizadas para almacenar cualquier dato de 8 bits.

Registro de trabajo W. Este es el registro de trabajo principal, se comporta de manera similar al acumulador en los microprocesadores. Este registro participa en la mayoría de las instrucciones. Está directamente relacionado con la Unidad Aritmética y Lógica ALU.

Módulo del convertidor Análogo a Digital

Este módulo permite la conversión de una señal de entrada análoga a su correspondiente valor numérico de 10 bits. El módulo tiene ocho entradas análogas, las cuales son multiplexadas dentro de un circuito de muestreo y retención. La salida del multiplexor es la entrada al convertidor, el cual genera el resultado por medio de aproximaciones sucesivas, figura 3.12.



Módulo del Convertidor A/D

La referencia análoga de voltaje es seleccionada por software permitiendo utilizar la fuente de alimentación del PIC (VDD) o un nivel de voltaje externo aplicado al pin 5 (RA3/AN3/ VREF +).

El módulo tiene los siguientes registros asociados:

- ADCON0 : Controla la operación del módulo A/D.
- ADCON1 : Configura las funciones de los pines del puerto análogo.

- ADRESL : Contiene la parte BAJA del resultado de la conversión A/D.
- ADRESH : Contiene la parte ALTA del resultado de la conversión A/D.

Registros de Control del Módulo Convertidor Análogo/Digital

Registro ADCON0

Este es un registro que permite seleccionar cual de las entradas análogas va a ser leída y permite dar la orden de iniciar el proceso de conversión, sus ocho bits son los siguientes:

ADCS1	ADCS0	CHS2	CHS1	CHS0	GO/DONE	-	ADON
bit7							bit 0

- Bit 0 (ADON): Bit de activación del módulo. ADON = 1, Módulo A/D operando. ADON = 0, Módulo A/D desactivado.
- Bit 2 (GO/DONE): Estado de conversión: GO = 1, Empieza conversión. GO = 0, conversión finalizada.
Si ADON = 0, Este bit es cero.
- Bits 3, 4 y 5 (CHS0, CHS1, CHS2): Selección del canal a convertir (canal 0 - 7).
- Bits 6 y 7(ADCS0, ADCS1): Selección del reloj de conversión.

ADCS1	ADCS0	FRECUENCIA DE CONVERSIÓN
0	0	$F_{osc} / 2$
0	1	$F_{osc} / 8$
1	0	$F_{osc} / 32$
1	1	FRC

Frecuencias de conversión para el módulo A/D

Registro ADCON1.

Este es un registro que permite seleccionar como se ubican los diez bits resultado de la conversión A/D y permite seleccionar cuales de los pines del puerto A trabajarán como entradas análogas y cuales como entradas digitales. Adicionalmente, permite seleccionar los voltajes de referencia del convertidor.

ADFM	-	-	-	-	PCFG2	PCFG1	PCFG0
bit7							bit 0

El bit 7 (ADFM) selecciona el formato del resultado de la conversión:

Si ADFM = 1, el resultado se justifica a la derecha: Los 6 bits más significativos de ADRESH son cero.

Si ADFM = 0, el resultado se justifica a la izquierda: Los 6 bits menos significativos de ADRESL son cero.

Con los tres bits (PCFG0, PCFG1, PCFG2) se configuran los pines del puerto A como de entradas análogas o entrada/salida digital, así como la referencia de voltaje que utilizará el convertidor. Figura 3.14.

PCFG3: PCFG0	AN7 ⁽¹⁾ RE2	AN6 ⁽¹⁾ RE1	AN5 ⁽¹⁾ RE0	AN4 RA5	AN3 RA3	AN2 RA2	AN1 RA1	AN0 RA0	VREF+	VREF-	CHAN / REFS
0000	A	A	A	A	A	A	A	A	VDD	VSS	8/0
0001	A	A	A	A	VREF+	A	A	A	RA3	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	RA3	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	RA3	VSS	2/1
011x	D	D	D	D	D	D	D	D	VDD	VSS	0/0
1000	A	A	A	A	VREF+	VREF-	A	A	RA3	RA2	6/2
1001	D	D	A	A	A	A	A	A	VDD	VSS	6/0
1010	D	D	A	A	VREF+	A	A	A	RA3	VSS	5/1
1011	D	D	A	A	VREF+	VREF-	A	A	RA3	RA2	4/2
1100	D	D	D	A	VREF+	VREF-	A	A	RA3	RA2	3/2
1101	D	D	D	D	VREF+	VREF-	A	A	RA3	RA2	2/2
1110	D	D	D	D	D	D	D	A	VDD	VSS	1/0
1111	D	D	D	D	VREF+	VREF-	D	A	RA3	RA2	1/2

A = Entrada Análoga

D = Entrada/Salida Digital

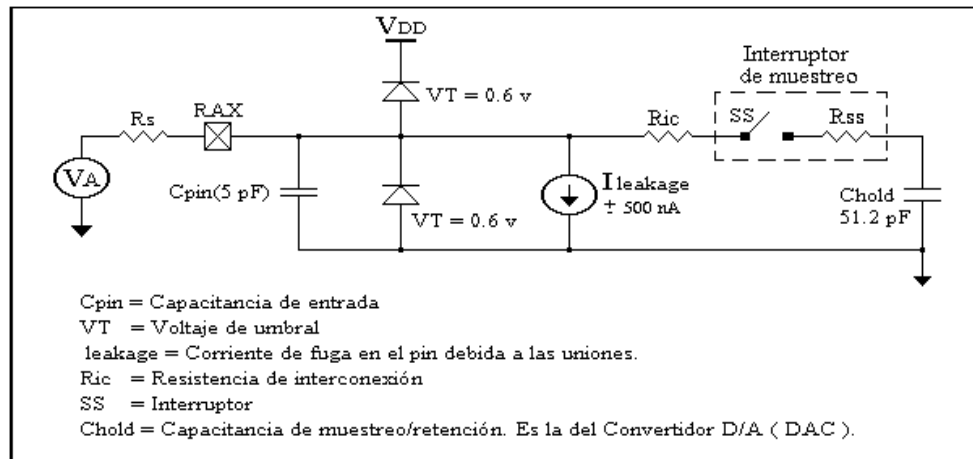
Nota 1: Estos canales no están disponibles en los dispositivos de 28 pines

Selección de los canales análogos a utilizar.

Cuando se completa la conversión A/D, el resultado se carga en los registros ADRESH y ADRESL (en el formato configurado por el bit ADFM).

El bit GO/DONE (ADCON0<2>) se pone en cero y el bit bandera de la interrupción A/D (ADIF) se pone en uno.

Después de que el módulo ha sido configurado, al canal seleccionado se debe hacer un muestreo antes de empezar la conversión. El tiempo requerido para el muestreo es definido como Tad.



Modelo circuital de la Entrada Análoga.

Requerimientos para el Muestreo:

Para que el convertidor A/D tenga precisión, se debe permitir que el condensador de retención se cargue con todo el nivel de voltaje del canal de entrada. En la figura

3.15. se muestra el modelo de entrada análoga. La máxima impedancia recomendada para las fuentes análogas es de 10 Kohm. Después que se selecciona el canal de entrada análoga (o es cambiado) se debe esperar un tiempo de muestreo antes de que la conversión se inicie.

Selección de la frecuencia de conversión:

La conversión A/D requiere $10 T_{ad}$. La fuente del reloj de conversión es seleccionada por software. Las cuatro opciones posibles para T_{ad} son:

- 2
 T_{osc}
- 8
 T_{osc}
- 32
 T_{osc}
- Oscilador interno
 RC

Para conversiones correctas, el reloj de conversión (T_{ad}) debe ser seleccionado para tener un tiempo mínimo T_{ad} de 1.6 useg.

FORMA DE REALIZAR LA CONVERSIÓN A/D

:

- Configurar el módulo

- Asegurar el tiempo de muestreo requerido [Tad].
 - Empezar la conversión.
 - Esperar que se realice la conversión, así :
 - Haciendo un muestreo al bit GO/DONE hasta que éste es cero.
 - Esperar la interrupción del convertidor.
 - Leer los registros ADRESH y ADRESL. ADIF se debe resetear si se usa interrupción.
 - Para una siguiente conversión vaya al paso primero o segundo según se requiera.
- Antes de empezar otro muestreo se debe esperar como mínimo 2 Tad.

Resumen de Características principales del PIC16F877:

- Memoria de programa : FLASH, 8 K de instrucciones de 14 bits c/u.
- Memoria de datos : 368 bytes RAM, 256 bytes EEPROM.
- Pila (Stack) : 8 niveles (14 bits).
- Fuentes de interrupción : 13
- Instrucciones : 35
- Encapsulado : DIP de 40 pines.
- Frecuencia oscilador : 20 MHz (máxima)
- Temporizadores/Contadores: 1 de 8 bits (Timer 0); 1 de 16 bits (Timer 1); 1 de 8 bits (Timer 2) con pre y post escalador. Un perro guardián (WDT)
- Líneas de E/S : 6 del puerto A, 8 del puerto B, 8 del puerto C, 8 del puerto D y 3 del puerto E, además de 8 entradas análogas.
- Dos módulos de Captura, Comparación y PWM:
 - Captura: 16 bits. Resolución máx. = 12.5 nseg.
 - Comparación: 16 bits. Resolución máx. = 200 nseg.
 - PWM: Resolución máx. = 10 bits.
- Convertidor Análogo/Digital de 10 bits multicanal (8 canales de entrada).
- Puerto serial síncrono (SSP) con bus SPI (modo maestro) y bus I²C (maestro/esclavo).
- USART (Universal Synchronous Asynchronous Receiver Transmitter) con dirección de detección de 9 bits.
- Corriente máxima absorbida/suministrada (sink/source) por línea (pin): 25 mA
- Oscilador : Soporta 4 configuraciones diferentes: XT, RC, HS, LP.
- Tecnología de Fabricación: CMOS
- Voltaje de alimentación: 3.0 a 5.5 V DC
- Puede operar en modo microprocesador

Anexo 2: Encuesta

Objetivo: Conocer el nivel de satisfacción que tienen los estudiantes del Laboratorio de Electrología con respecto a la utilización de la Placa basada en Microcontroladores PIC

Instrucciones: Después de haber utilizado el Entrenador Universal, marque con una x la respuesta que a su juicio considere la correcta, evite realizar tachones o borrones.

1. ¿El entrenador universal ha cumplido con sus expectativas?

- Totalmente
- En su mayor parte
- No ha cumplido

2. ¿Piensa Ud. que el entrenador universal le ayudaría a comprobar el buen funcionamiento de sus proyectos?

- Totalmente
- En su mayor parte
- No ayudaría

3. ¿Considera Ud. que la utilización del entrenador universal es simple y entendible?

- Totalmente
- En su mayor parte
- No es simple ni entendible

4. ¿Las funciones que proporciona el entrenador universal, se adapta a las necesidades existentes en el Laboratorio de Electrología?

- Totalmente
- En su mayor parte
- No se adapta



5. Cree Ud. que el uso del entrenador universal para el desarrollo de sus proyectos le ayudaría a evitar la pérdida de tiempo y dinero?

- Totalmente
- En su mayor parte
- No ayudaría

6. ¿Cuál es la calificación que le asignaría al entrenador universal?

- Excelente
- Bueno
- Regular
- Malo

Gracias por su colaboración.