

**UNIVERSIDAD CATÓLICA
DE SANTIAGO DE GUAYAQUIL**

FACULTAD DE EDUCACIÓN TÉCNICA PARA EL DESARROLLO

**TESIS DE GRADO PREVIO A LA OBTENCION DEL TITULO DE
INGENIERO EN TELECOMUNICACIONES CON MENCIÓN EN
GESTION EMPRESARIAL**

TEMA:

**“DISEÑO E IMPLEMENTACION DE PLATAFORMA BASADA EN
MICROCONTROLADORES PIC PARA FACILITAR EL ESTUDIO PRACTICO Y
LA ELABORACION DE PROYECTOS EN LA CARRERA DE INGENIERIA EN
TELECOMUNICACIONES”**

ELABORADO POR:

ALEX MENDEZ LINO

DIRECTOR:

MSC. EDUARDO MENDOZA

GUAYAQUIL ABRIL, 2014

CERTIFICACIÓN

Certifico que el presente trabajo fue realizado en su totalidad por el **Sr. ALEX MENDEZ LINO** como requerimiento parcial para la obtención del título de INGENIERO EN TELECOMUNICACIONES.

DIRECTOR

Ing. Eduardo Mendoza, Msg

REVISADO POR:

Ing. Washington Medina
Revisor Metodológico

Ing. Jimmy Alvarado
Revisor de Contenido

DIRECTOR DE LA CARRERA

Ing. Armando Heras Sánchez

Guayaquil, 2014



UNIVERSIDAD CATÓLICA
DE SANTIAGO DE GUAYAQUIL

INGENIERÍA EN TELECOMUNICACIONES

DECLARACIÓN DE RESPONSABILIDAD

ALEX MENDEZ LINO

DECLARO QUE:

El proyecto de tesis denominado **“DISEÑO E IMPLEMENTACION DE PLATAFORMA BASADA EN MICROCONTROLADORES PIC PARA FACILITAR EL ESTUDIO PRACTICO Y LA ELABORACION DE PROYECTOS EN LA CARRERA DE INGENIERIA EN TELECOMUNICACIONES ”** ha sido desarrollado con base a una investigación exhaustiva, respetando derechos intelectuales de terceros conforme las citas que constan al pie de las páginas correspondientes, cuyas fuentes se incorporan en la bibliografía.

Consecuentemente este trabajo es de mi autoría.

En virtud de esta declaración, me responsabilizo del contenido, veracidad y alcance científico del proyecto de grado en mención.

Guayaquil, 2014



UNIVERSIDAD CATÓLICA
DE SANTIAGO DE GUAYAQUIL

INGENIERÍA EN TELECOMUNICACIONES

AUTORIZACIÓN

Yo, ALEX MENDEZ LINO

Autorizo a la Universidad Católica de Santiago de Guayaquil, la publicación, en la biblioteca de la institución del proyecto titulado: **“DISEÑO E IMPLEMENTACIÓN DE PLATAFORMA BASADA EN MICROCONTROLADORES PIC PARA FACILITAR EL ESTUDIO PRÁCTICO Y LA ELABORACIÓN DE PROYECTOS EN LA CARRERA DE INGENIERÍA EN TELECOMUNICACIONES”**, cuyo contenido, ideas y criterios son de nuestra exclusiva responsabilidad y autoría.

Guayaquil Abril, 2014

ALEX MENDEZ LINO

AGRADECIMIENTO

Al finalizar un trabajo tan arduo y lleno de dificultades como el desarrollo de una tesis es inevitable que te asalte un muy humano egocentrismo que te lleva a concentrar la mayor parte del mérito en el aporte que has hecho. Sin embargo, el análisis objetivo muestra inmediatamente que la magnitud de este aporte hubiese sido imposible sin la participación de personas que han facilitado las cosas para que este trabajo llegue a un feliz término. Por ello, es para mí un verdadero placer utilizar este espacio para ser justo y consecuente con ellas, expresando mis agradecimientos.

Debo agradecer de manera especial y sincera al Profesor Eduardo Mendoza por aceptarme para realizar esta tesis bajo su dirección. Su apoyo y confianza en mi trabajo y su capacidad para guiar mis ideas ha sido un aporte invaluable, no solamente en el desarrollo de esta tesis, sino también en mi formación como investigador. Las ideas propias, siempre enmarcadas en su orientación y rigurosidad, han sido la clave del buen trabajo que hemos realizado juntos, el cual no se puede concebir sin su siempre oportuna participación. Le agradezco también el haberme facilitado siempre los medios suficientes para llevar a cabo todas las actividades propuestas durante el desarrollo de esta tesis.

Muchas gracias a mi familia que siempre me brindó su apoyo y a DIOS que me ha permitido seguir con salud para seguir creciendo en este largo camino de la vida y dar este orgullo a mi familia.

DEDICATORIA

Al creador de todas las cosas, el que me ha dado fortaleza para continuar cuando a punto de caer he estado; por ello, con toda la humildad que de mi corazón puede emanar, dedico primeramente mi trabajo a Dios.

De igual forma, dedico esta tesis a mi madre que ha sabido formarme con buenos sentimientos, hábitos y valores, lo cual me ha ayudado a salir adelante en los momentos más difíciles.

A mi hermana que siempre ha estado junto a mí y brindándome su apoyo, muchas veces poniéndose en el papel de padre.

A mi familia en general, porque me han brindado su apoyo incondicional y por compartir conmigo buenos y malos momentos.

CONTENIDO

RESUMEN	
ABSTRACT	
GLOSARIO TÉCNICO.....	XII
INTRODUCCION.....	1
CAPÍTULO 1: GENERALIDADES	
1.1 OBJETIVOS.....	2
1.1.1 OBJETIVO GENERAL.....	2
1.1.2 OBJETIVOS ESPECÍFICOS.....	2
1.2 PLANTEAMIENTO DEL PROBLEMA.....	3
1.3 JUSTIFICACIÓN.....	3
1.4 HIPÓTESIS.....	3
CAPÍTULO 2: MARCO TEÓRICO	
2.1 EL MICROCONTROLADOR.....	4
2.2 CARACTERÍSTICAS DEL MICROCONTROLADOR.....	5
2.2.1 ARQUITECTURA BÁSICA.....	5
2.2.2 PROCESADOR CPU.....	6
2.2.3 CISC.....	7
2.2.4 RISC.....	7
2.2.5 SISC.....	7
2.2.6 MEMORIA.....	7
2.2.6.1 ROM CON MASCARA.....	8
2.2.6.2 EPROM.....	8
2.2.6.3 EEPROM.....	9
2.3 EL MICROCONTROLADOR PIC 18F4520.....	9
2.3.1 CARACTERÍSTICAS DEL PIC 18F4520.....	10
2.3.1.1 RESISTENCIA DE LA MEMORIA.....	11
2.3.1.2 AUTO PROGRAMACIÓN.....	11
2.3.1.3 INSTRUCCIÓN DE SET EXTENDIDO.....	12
2.3.1.4 MÓDULO CCP.....	12
2.3.1.5 USART DIRECCIONABLE.....	12
2.3.1.6 CONVERTIDOR A/D DE 10 BIT.....	12
2.3.1. WATCHDOG (WDT).....	13

2.4 BLOQUES QUE COMPONEN UNA PLATAFORMA HARDWARE PARA PIC.....	13
2.4.1 FUENTE DE ALIMENTACIÓN.....	13
2.4.2 PERIFÉRICOS DE SALIDA.....	13
2.4.3 PERIFÉRICOS DE ENTRADA.....	14
2.4.4 COMUNICACIONES.....	14
2.5 COMPONENTES DE UNA PLATAFORMA HARDWARE PARA MICROCONTROLADOR.....	14
2.5.1 ENCAPSULADOS THT.....	16
2.5.2 ENCAPSULADOS SMD/SMT.....	17
2.5.3 PISTAS DE UN CIRCUITO IMPRESO.....	18
2.5.3.1 PADS.....	19
2.5.3.2 CAMINOS DE COBRE.....	20
2.5.3.3 PERFORACIONES METALIZADAS THRU-HOLE.....	21
2.5.3.4 PERFORACIONES METALIZADAS CIEGAS.....	22
2.5.3.5 PERFORACIONES METALIZADAS OCULTAS.....	23
2.5.4 APILAMIENTOS DE CABECERAS.....	24
2.5.5 CONTENCIÓN DE PATILLAS.....	25
2.5.6 SOFTWARE.....	26
2.5.7 REQUERIMIENTO DE ENERGÍA.....	27
2.6 COMPILADOR PIC-C CCS.....	28
2.7 PROGRAMACIÓN EN CCS.....	31
2.7.1 ESTRUCTURA DE UN PROGRAMA.....	32
2.7.1.1 DIRECTIVA DE PROCESADO.....	32
2.7.1.2 PROGRAMAS O FUNCIONES.....	32
2.7.1.3 INSTRUCCIONES.....	32
2.7.1.4 COMENTARIOS.....	32
2.7.2 TIPOS DE DATOS.....	32
2.7.3 CONSTANTES.....	33
2.7.4 VARIABLES.....	33
2.8 SOFTWARE DE DISEÑO EAGLE.....	35
2.8.1 INTERFAZ DE USUARIO EAGLE.....	36
CAPÍTULO 3: DISEÑO DE PLATAFORMA HARDWARE PARA PIC18F4520.	
3.1 INTRODUCCIÓN.....	37
3.2 RESET Y OSCILADOR EXTERNO.....	40

3.3 COMUNICACIÓN ICD2.....	40
3.4 ESPECIFICACIONES.....	45
3.5 RESET (MCLR).....	47
CAPÍTULO 4: SOFTWARE Y PROGRAMACIÓN DE LA PLATAFORMA HARDWARE.	
4.1 Comunicación entre el PIC18f4520 y la memoria EEPROM mediante el protocolo SPI.....	48
4.2 Configuración de Puertos A, B, C Entradas/Salidas.....	48
4.3 Entradas Analógicas.....	48
4.4 Led en Puerto B (PORTB).....	49
4.5 Programación.....	50
CAPÍTULO 5: COSTOS.....	51
CONCLUSIONES.....	52
RECOMENDACIONES.....	53
BIBLIOGRAFÍA.....	54
ANEXOS.....	56

FIGURAS

Figura 2.1 Microcontrolador PIC 18F4520.....	1
Figura 2.2: Representación de Arquitectura Harvard.....	2
Figura 2.3: Esquemas de pines del PIC18F4520.....	3
Figura 2.4: Ejemplos de componentes Thru-Hole.....	4
Figura 2.5: Componentes Thru-Hole en una plataforma base.....	5
Figura 2.6: Ejemplos de componentes SMD/SMT.....	6
Figura 2.7: Componentes SMD/SMT en una plataforma base.....	7
Figura 2.8: Caminos de circuito impreso en una plataforma base.....	8
Figura 2.9: Apilamientos de cabeceras de una plataforma base.....	9
Figura 2.10: Patillas que contienen sus respectivas pistas de conexión.....	10
Figura 2.11: Requerimiento de energía externo de una tarjeta escudo.....	11
Figura 2.12: Expansión de la máscara de soldado.....	12
Figura 2.13: Expansión de la máscara de soldado.....	13

Figura 2.14: Starckups de 2, 4 y 6 capas.....	14
Figura 2.15: Circuito impreso con 4 capas.....	15

TABLAS

Tabla 2.1: Características del PIC18F4550.....	2
Tabla 2.2: Orden de la pila de capas.....	18
Tabla 2.3: Detalles de las secciones de colores en un circuito impreso.....	24
Tabla 2.4: Tipos de Datos.....	32
Tabla 2.5: Tipos de Constantes en CCS C.....	33
Tabla 2.6: Definición de Constantes con un sufijo en CCS C.....	34
Tabla 2.7: Caracteres Especiales en CCS C.....	35

GLOSARIO TÉCNICO.

A/D	Analógico/Digital.
ADC	Analog Digital Converter / Convertidor Analógico Digital.
ADCON0	Registro 0 de control del convertidor analógico/digital.
ADCON1	Registro 1 de control del convertidor analógico/digital.
ADFM	Bit selector de formato de resultado del convertidor analógico/digital.
ADRESH	Byte alto del registro de resultado del convertidor analógico/digital.
ADRESL	Byte bajo del registro de resultado del convertidor analógico/digital.
AC	Alternating Current / Corriente Alterna.
Bms	Bit menos significativo.
CAN	Controller Area Network / Red de Area de Controlador.
DC	Direct Current / Corriente Directa.
CMOS	Complementary Metal Oxide / Semiconductor de Oxido de Metal Complementario.
COP8	Control Oriented Processor / Procesador Orientado a Control.
CPU	Central Processing Unit / Unidad Central de Procesamiento.
D/A	Digital/Analógico.
DAC	Digital Analog Converter / Convertidor Digital Analógico.
DIP	Dual Inline Package / Encapsulado Doble en línea.
DSC	Digital Signal Controller / Controlador Digital de Señales.
DSP	Digital Signal Processor / Procesador Digital de Señales.
E/S	Entradas y Salidas.
EEPROM	Electrically Erasable Programmable Read-Only Memory / Memoria de solo lectura programable eléctricamente borrrable.
EIA	Electronic Industries Association / Asociación de Industrias de Electronica.
FFT	Fast Fourier Transform / Transformada Rápida de Fourier.

FUSES	Fusibles.
HSPLL	Oscilador del microcontrolador.
ICD	In-Circuit Debugger / Depurador en Circuito.
ICE	In-Circuit Emulator / Emulador en Circuito.
IDE	Integrated Development Environment / Ambiente Integrado de Desarrollo.
IEEE	Institute for Electrical and Electronics Engineers / Instituto para Ingenieros Electricos y Electronicos.
IIC	Inter-Integrated Circuit / Inter-conexión de Circuitos Integrados.
ISP	In System Programming / Programacion en Sistema.
KHz	Kilo-Hertz.
GLCD	Graphic Liquid Crystal Display / Pantalla Grafica de Cristal Líquido.
LED	Light Emitting Diode / Diodo Emisor de Luz.
LPT	Line Print Terminal / Terminal de Línea de Impresora.
mA	Mili-Ampere.
MHz.	Mega-Hertz.
MIDI	Musical Instrument Digital Interface / Interfaz Digital de Instrumento Musical.
MIMO	Multiple Input - Multiple Output / Múltiples Entradas - Múltiples Salidas.
MIPS	Million of Instructions per Second / Millones de Instrucciones por Segundo.
MOSFET	Metal Oxide Semiconductor Field Effect Transistor / Transistor de efecto de campo de Semiconductor de Oxido de Metal.
nF	Nano-Faradio.
pF	Pico-Faradio.
PIC	Controlador de Interfaz Periférico.
PWM	Pulse Width Modulation / Modulación por Ancho de Pulso.
RAM	Random Access Memory / Memoria de Acceso Aleatorio.
RC	Resistencia Capacitor.

RISC	Reduced Instruction Set Computer / Computadora con Conjunto Reducido de Instrucciones.
ROM	Read-Only Memory / Memoria de Solo Lectura.
RS232	Protocolo de comunicación serial.
SCI	Serial Communication Interface / Interfaz de Comunicación Serial.
SCL	Serial Clock / Reloj Serial.
SD	Secure Digital / Seguro Digital.
SDA	Serial Data / Dato Serial.
SISO	Single Input - Single Output / Una Entrada – Una Salida.
Spbrg	Registro Generador de Baudios.
TOSC	Tiempo de Oscilación.
TTL	Transistor-Transistor Logic / Lógica Transistor-Transistor.
μC	Microcontrolador.
μF	Micro-Faradio.
μs	Micro-segundo.
USART	Universal Synchronous – Asynchronous Receiver Transmitter / Transmisor-Receptor Universal Síncrono-Asíncrono.
USB	Universal Serial Bus / Bus Serie Universal.
VDC	Voltage Direct Current / Voltaje de Corriente Directa.

RESUMEN

El presente anteproyecto de tesis tiene como finalidad realizar el diseño e implementación de una plataforma de hardware para facilitar el estudio teórico y práctico de microcontroladores de la carrera de Ingeniería en Telecomunicaciones de la Facultad Técnica para el Desarrollo de la Universidad Católica de Santiago de Guayaquil, teniendo como objetivo su aplicación en un hardware que permita realizar prácticas programables, ayudando de esta manera el aprendizaje de Estudiantes y Docentes.

Esto se lo realizara integrando la parte conceptual y el área práctica que corresponde a la materia y así poder combinar un sistema físico con un programa, obteniendo un mejor aprendizaje y de una manera más fácil poder aplicar los conocimientos adquiridos en el área profesional que se requiera.

El capítulo 1, trata sobre los aspectos generales y fundamentales del proyecto de tesis.

El capítulo 2, trata sobre el marco teórico de los microcontroladores y la plataforma hardware base que será implementada.

El capítulo 3, trata sobre el diseño y la implementación de la plataforma hardware con microcontrolador pic18f4520.

El capítulo 4, trata sobre la programación de la plataforma hardware.

El capítulo 5, trata sobre los costos de la implementación.

ABSTRACT

This draft thesis aims to make the design and implementation of a hardware platform with mountable cards to provide theoretical and practical study of the microcontrolador Career Telecommunications Engineering Faculty technical Development of the Catholic University of Santiago de Guayaquil, taking aim at a hardware implementation that allows for programmable practices, thus helping the learning of students and teachers.

This is what conduct integrating the conceptual part and practice area corresponding to the subject and be able to combine a physical system with a program, getting a better learning and an easier way to apply the knowledge acquired in the professional field is required .

Chapter 1 deals with general and fundamental aspects of the thesis project.

Chapter 2 discusses the theoretical framework based microcontrollers and hardware platform that will be implemented.

Chapter 3 discusses the design and implementation of the hardware platform with PIC18F4520 microcontroller.

Chapter 4 deals with the programming of the hardware platform.

Chapter 5 discusses the implementation costs.

INTRODUCCIÓN

El proyecto de tesis, consiste en diseñar e implementar una plataforma de hardware con microcontrolador PIC18F4520 que ayude a la realización práctica de las simulaciones que los estudiantes efectúan durante el transcurso de las clases con el software llamado PROTEUS, que es una aplicación capaz de realizar simulaciones de modelado del sistema virtual y circuitos, que ayudara a simular y predecir cómo funcionará un circuito, así como su integridad en el diseño a nivel de circuito y la placa con su código de programación correspondiente.

Las aplicaciones que se pueden crear a partir de la plataforma base son muy sencillas y de gran variedad tecnológica. El microcontrolador PIC18F4520 tiene una infraestructura muy fácil de entender para el aprendizaje.

La evolución de plataformas con microcontroladores ha dado un giro muy radical, con aplicaciones de sistemas embebidos resultando por la tecnología que usan muy prácticas para el aprendizaje de los estudiantes.

Por lo tanto, la implementación de la plataforma hardware para facilitar el estudio de aprendizaje sobre microcontroladores, y disminuir costos y tiempo es muy necesaria para los estudiantes de la Facultad Técnica de la UCSG.

CAPITULO 1.- GENERALIDADES

1.1 OBJETIVOS.

1.1.1 OBJETIVO GENERAL.

- Demostrar las funciones y aplicaciones de la plataforma hardware basada en microcontrolador a través de la elaboración de una tarjeta principal, que permita la enseñanza de los microcontroladores.

1.1.2 OBJETIVOS ESPECIFICOS.

- Desarrollar una herramienta que aporte a estudiantes un hardware probado, confiable, que se pueda utilizar cuando se quiera realizar prácticas.
- Elaborar un conjunto de librerías y prácticas para el aprendizaje de microcontroladores.

1.2 PLANTEAMIENTO DEL PROBLEMA:

En la Facultad Técnica no se posee ningún tipo de hardware sobre tarjetas de desarrollo basadas en microcontroladores que permitan al estudiante o docente adaptar periféricos a dicha tarjeta para programar de manera directa sobre ellas, después de haber practicado y programado en los simuladores usados. Lo cual hace que sea necesaria la implementación de una tarjeta de desarrollo con microcontrolador durante el estudio de dicha materia.

1.3 JUSTIFICACION:

El proyecto de diseñar e implementar una plataforma hardware con PIC18F4520, es porque durante el periodo de estudiante no hubo lo necesario para hacer las prácticas de manera física e implementando materiales, sino solo el uso a base a simuladores, por eso se necesita también la programación directa a un tipo de hardware que permita enriquecer los conocimientos del estudiante a un nivel más cercano a la realidad de la carrera.

Por eso que el diseño y la implementación de la plataforma de desarrollo para la programación de microcontroladores PIC son indispensables para un óptimo conocimiento y una gran ayuda para la elaboración de proyectos.

1.4 HIPÓTESIS:

La implementación de la plataforma hardware puede ser un recurso muy fiable y económico para el aprendizaje con microcontroladores de los estudiantes, a través de un método práctico y real.

CAPÍTULO 2 - MARCO TEÓRICO

2.1 MICROCONTROLADOR.-

S. Romero (2010) "Un microcontrolador es una computadora de un solo chip. Micro se refiere a que el dispositivo es pequeño y controlador, es decir que es empleado en sistemas de control".

Un microprocesador difiere de un Microcontrolador en muchos aspectos. La principal es que un microprocesador requiere severos componentes externos para su operación, como memoria de programa y memoria de datos, dispositivos de Entrada/Salida, y un circuito de reloj externo. Un microcontrolador tiene todos los chips de soporte incorporados dentro del único chip. Todos los microcontroladores operan en conjunto de instrucciones (o el programa de usuario) almacenada en la memoria.

Según (Breijo 2009) "Un microcontrolador obtiene las instrucciones de su memoria programa una por una decodificada, estas instrucciones y luego lleva a cabo las operaciones requeridas."

Los microcontroladores han sido tradicionalmente programados usando el lenguaje de ensamblador del dispositivo de destino. A pesar de que el lenguaje ensamblador es rápido, tiene severas desventajas. Un programa ensamblador consiste en nemónicos (dato simbólico que identifica un comando generalmente numérico (binario, octal, hexadecimal) de una forma más sencilla que su numeración original) lo cual hace que su aprendizaje y mantenimiento de programa escrito sea difícil.

También, los fabricantes de Microcontroladores de diferentes firmas tienen diferentes lenguajes de ensamblador, así el usuario debe de aprender un nuevo lenguaje de programación para cada nuevo microcontrolador que se use.



Figura 2.1: Microcontrolador PIC 18F4520.
Fuente: Hoja de datos MICROCHIP.

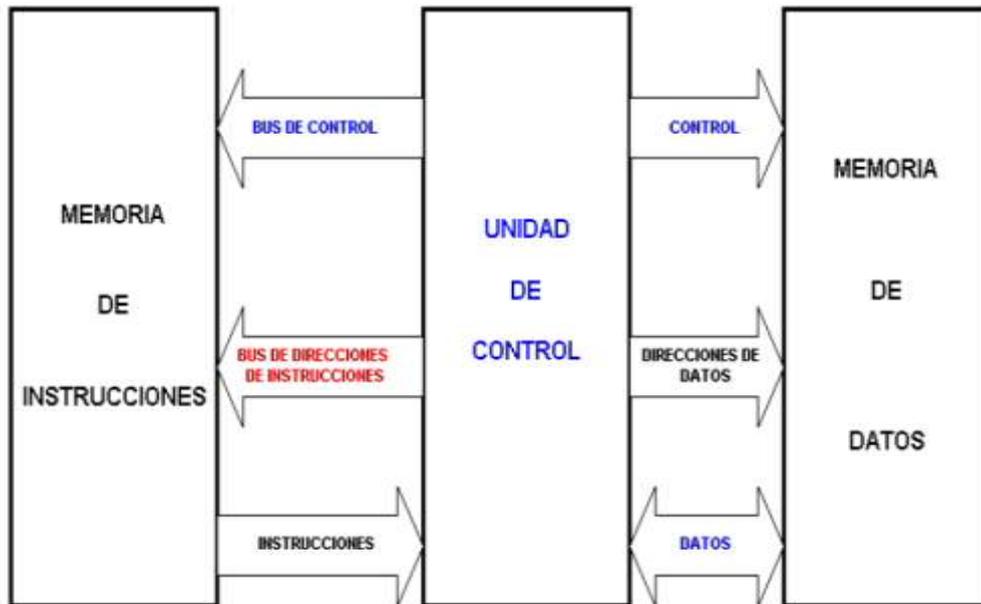
2.2 CARACTERISTICAS DE UN MICROCONTROLADOR.-

Los microcontroladores de distintos fabricantes tienen distintas arquitecturas y capacidades. Algunos se pueden enfocar en aplicaciones particulares mientras que otros pueden ser totalmente inservibles para la misma aplicación.

2.2.1 ARQUITECTURA BASICA.-

O. Barra (2011) "La arquitectura de los microcontroladores es independiente: una, que contiene solo instrucciones y otra, solo datos". Ambas disponen de sus respectivos sistemas de buses de acceso y es posible realizar operaciones de acceso (lectura o escritura) simultáneamente en ambas memorias, con lo cual la velocidad del sistema aumenta.

Figura 2.2: Representación de Arquitectura Harvard.



Fuente: Libro Microcontroladores PIC- H.VALLEJO.
Elaborado: Autor.

2.2.2 PROCESADOR CPU.-

Es la parte central e importante del microcontrolador y según E. Palacios (2005) "determina sus principales características a nivel de la parte física (hardware) y de la programación (software)".

Se encarga de direccionar la memoria de instrucciones, recibir el código lenguaje máquina de la instrucción en curso, su decodificación y la ejecución de la operación que implica la instrucción, así como la búsqueda de los operadores y el almacenamiento de resultado. Existen tres orientaciones en cuanto a la arquitectura y funcionalidad de los procesadores actuales:

2.2.3 CISC.-

M. Bates (2006) "Computadores de Juego de Instrucciones Complejo; disponen de más de 80 instrucciones maquina en su repertorio, algunas de las cuales son muy sofisticadas y potentes", requiriendo muchos ciclos para su ejecución.

2.2.4 RISC.-

En estos microcontroladores el número de instrucciones maquina es muy reducido y las instrucciones son simples, y generalmente, se ejecutan en un ciclo. La sencillez y rapidez de las instrucciones permiten optimizar el hardware y el software del procesador.

2.2.5 SISC.-

J. Angulo (2010) "En los microcontroladores destinados a aplicaciones muy específicas, el juego de instrucciones, además de ser reducido, es específico", o sea, las instrucciones se adaptan a las necesidades de la aplicación prevista. Esta filosofía se ha bautizado con el nombre de SISC (Computadores de Juego de Instrucciones Especifico).

2.2.6 MEMORIA.-

H. Vallejo (1998) "En los microcontroladores disponen de dos memorias integradas en el propio chip. La primera que es la memoria de programa que debe ser no volátil, tipo ROM, y se destina a contener el programa de instrucciones que gobierna la aplicación".

La segunda memoria que es de tipo RAM, volátil, que se la emplea para guardar las variables y los datos.

La RAM en estos dispositivos es de poca capacidad pues solo debe contener las variables y los cambios de información que se produzcan en el transcurso del programa. Por otra parte, como solo existe un programa activo, no se requiere guardar una copia del mismo en la RAM pues se ejecuta directamente desde la ROM.

Existen memorias no volátiles que pueden poseer los microcontroladores puesto que la aplicación y utilización del mismo es diferente. A continuación se describen las cinco versiones de memoria no volátil.

2.2.6.1 ROM CON MASCARA.-

M. Bates (2006) "Es una memoria no volátil de solo lectura cuyo contenido se graba durante la fabricación del chip". El elevado costo del diseño de la máscara solo hace aconsejable el empleo de los microcontroladores con este tipo de memoria cuando se precisan cantidades superiores a varios miles de unidades.

2.2.6.2 EPROM.-

H. Vallejo (2008) "Los microcontroladores que disponen de memoria EPROM, pueden borrarse y grabarse varias veces". La grabación se realiza, como en el caso de los OTP, con un grabador gobernado desde un PC. Si, posteriormente, se desea borrar el contenido. Se emplea luz ultravioleta y se la expone a una ventana de cristal en su superficie donde somete a la EPROM durante varios minutos. Las capsulas son de material cerámico y son más caros que los microcontroladores con memoria OTP que están hecho con material plástico.

2.2.6.3 EEPROM.-

F. Barra (2011) "Se trata de memorias de solo lectura, escritura, programables y borrables eléctricamente EEPROM". Tanto la

programación como el borrado, se realizan eléctricamente desde el propio grabador y bajo control programado de un PC. Es muy cómo y rápida la operación de grabado y la de borrado.

2.3 MICROCONTROLADOR PIC18F4520.-

T. Wilmshurst (2007) "Los microcontroladores PIC existen en gamas de 8bit, 16bit y 32bit. Dentro de la gama más simple de 8bit se encuentra el microcontrolador PIC18F4520", el cual pertenece a la familia PIC18 MCU. Sus características de memoria de programa, memoria RAM, número de Entradas/Salidas, número de canales analógicos y tipos de puertos de comunicación, han hecho de este PICZ uno de los más utilizados para diversas aplicaciones.

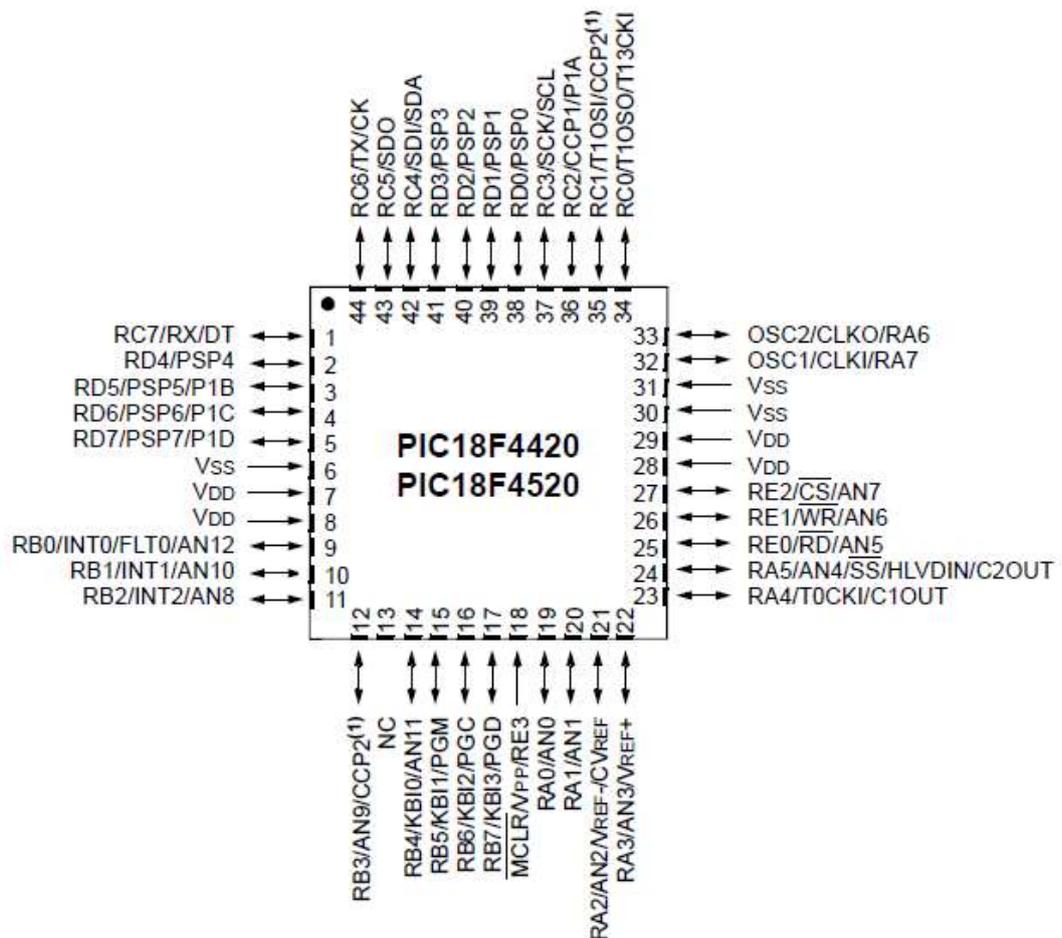


Figura 2.3: Esquemas de pines del PIC18F4520.

Fuente: Hoja de datos de MICROCHIP.

2.3.1 CARACTERÍSTICAS DEL PIC18F4520.-

A continuación se detallara las principales características del PIC18F4520.

Tabla 2.1: Características del PIC18F4550.

	PIC18F4520
Frecuencia de operación	DC - 40MHZ
Memoria de programa (BYTES)	32768
Memoria del programa (INSTRUCCIONES)	16384
Memoria de datos (BYTES)	1536
Memoria EEPROM de datos (BYTES)	256
Fuentes de interrupción	20
Puertos de E/S	Ports A, B, C, D, E
Temporizadores	4
Módulos de captura (PWM)	1
Comunicaciones serie	MSSP, Enhanced USART
Comunicaciones paralelas	Si
Módulo A/D 10-BIT	13 Canales de entrada
Restablecimiento	POR, BOR, Reset Instruction, Stack Full, Stack Underflow (PWRT, OST), MCLR (opcional), WDT
Detector programable de alta/baja tensión	Si
Detector de restablecimiento de programación	Si
Conjunto de instrucciones	75 y 83 con set extendido
Paquete	40 pin / 44 pin QFN y TQFP

Fuente: www.microchip.com

En la tabla se presentan las características principales del a familia PIC18. En particular, se pueden observar en la última columna de la derecha las características del PIC18F4520. Microchip (2009) "Este microcontrolador cuenta con 5 puertos E/S, 4 temporizadores, 20 fuentes de interrupción, comunicación serial, modulo USB, 13 canales de entradas analógicas y 2 módulos PWM".

- Arquitectura RISC avanzada Harvard: 18 bit con 8 bit de datos.
- Set de 77 instrucciones.
- Desde 18 a 80 patillas de conexión.
- Hasta 64K bytes de programa (Hasta 2M bytes en ROMless).
- Multiplicador Hardware 8x8.
- Hasta 3968 bytes de RAM Y 1K bytes de EEPROM.
- Frecuencia máxima de reloj 40Mhz. Hasta 10 MIPS.
- Pila de 32 niveles.
- Múltiples fuentes de interrupción.
- Periféricos de comunicación avanzados (CAN y USB).

2.3.1.1 RESISTENCIA DE LA MEMORIA.-

J. Angulo (2010) "Las celdas de memoria temporal (memoria flash) tanto para la memoria de programa y los datos EEPROM permiten ciclos de escritura hasta 100.000 para la memoria de programa y 1.000.000 para la memoria EEPROM".

2.3.1.2 AUTO-PROGRAMACION.-

T., Wilmshurst (2007) "Estos dispositivos pueden escribir en sus propios espacios de memoria del programa en virtud del control de software". Mediante el uso de una rutina de gestor de arranque situado en el bloque de arranque, protegida en la parte superior de la memoria de programa, se hace posible crear una aplicación que permita actualizarse a sí mismo.

2.3.2.3 INSTRUCCIÓN DE SET EXTENDIDO.-

J. González (2008) "La familia PIC18F4520 introduce una extensión opcional para el conjunto de instrucciones PIC18, que añade 8 nuevas instrucciones y un modo de direccionamiento indexado". Esta extensión, habilitada como opción en la configuración del dispositivo, ha sido

diseñado específicamente para optimizar el código de aplicación reentrante desarrollado originalmente en lenguajes de alto nivel, tales como el C.

2.3.2.4 MÓDULO CCP.-

J. Angulo (2010) "En el modo PWM, este módulo ofrece 1, 2 o 4 salidas moduladas para el control de medio puente y los controladores de puente completo". Otras características incluyen auto-apagado, para desactivar salidas PWM de las condiciones de selección de interrupción o de otro tipo y el pre-arranque automático, para reactivar las salidas una vez que la condición ha despejado.

2.3.2.5 USART DIRECCIONABLE.-

F. Barra (2011) "Este módulo de comunicación serie es capaz de funcionar con el estándar RS-232 y proporciona soporte para el protocolo de bus LIN". Además, incluye la detección automática de velocidad y una velocidad de transmisión generador de 16 bits para una mejor resolución. Cuando el microcontrolador utiliza el bloque oscilador interno, el USART proporciona un funcionamiento estable para aplicaciones sin necesidad de utilizar un cristal externo (o la necesidad de añadir potencia de acompañamiento).

2.3.2.6 CONVERTIDOR A/D DE 10-BIT.-

M. Bates (2006) "Este módulo incorpora el tiempo de adquisición programable, lo que permite un canal que se seleccionará y una conversión para ser iniciado sin esperar un período de muestreo y, por tanto, reducir el código de inicio".

2.3.2.7 WATCHDOG (WDT).-

E. Palacios (2005) "Esta versión mejorada incluye una de 16 bits, lo que permite un rango de tiempo de espera prolongado que es estable a través de la tensión de funcionamiento y la temperatura". Su función es contar cada cierto número de pulsos de reloj en un determinado tiempo, esperando algún evento generado por el programa, si no llega, el watchdog se activa y hace que todo empiece de nuevo y si llega el evento, entonces no hace nada.

2.4 BLOQUES QUE COMPONEN UNA PLATAFORMA HARDWARE PARA PIC.-

2.4.1 FUENTE DE ALIMENTACION DE TENSION REGULADA.-

Permite obtener diferentes tensiones para que la plataforma pueda disponer de una o varias líneas de alimentación. Se facilita diferentes valores de tensión, concretamente los más utilizados; 5V para circuitos integrados, y 12V para relés de salida, o múltiples tensiones con el mismo valor (+5V/+5V), separando así la alimentación de los circuitos integrados de la plataforma, de la utilizada en periféricos que pueden generar picos de corriente por altos consumos, como puede ser la visualización dinámica con displays de 7 segmentos. De esta forma, se consigue una mayor estabilidad en la alimentación de los microcontroladores que integran la plataforma.

2.4.2 PERIFÉRICOS DE SALIDA.-

Sirven para permitir al usuario de la plataforma visualizar acciones producidas por el μC . Sin estos dispositivos, el sistema del μC no podría mostrar los resultados de sus operaciones, en dispositivos como por ejemplo displays de 7 segmentos, visualizador LCD, conjunto de diodos LED, etc.

2.4.3 PERIFÉRICOS DE ENTRADA.-

Sirven para permitir al usuario y entorno interactuar con el μC , pudiendo introducir datos digitales o señales, en el caso de las entradas analógicas. Como por ejemplo un conjunto de pulsadores e interruptores, teclado matricial, entradas analógicas con juego de sensores y potenciómetros, generador de onda rectangular, etc.

2.4.4 COMUNICACIONES.-

Posibilitan al μC comunicarse con otros dispositivos, ya sea un ordenador (comunicación RS232 o USB) o con otros periféricos (comunicación I2C). Además, se engloban dentro de este apartado sistemas como el ICSP que permite al μC ser programado mientras se encuentra montado sobre el propio circuito.

2.5 COMPONENTES PARA UNA PLATAFORMA PARA MICROCONTROLADORES.-

2.5.1 ENCAPSULADOS DE TECNOLOGÍA DE AJUGEROS PASANTES (THT).-

Son todos aquellos componentes que poseen patillas para ser instalados en perforaciones metalizadas (llamadas Thru-Hole). Este tipo de componentes se sueldan por la capa opuesta de la tarjeta escudo. Generalmente son montados por un solo lado de la tarjeta y soldados del otro lado.



Figura 2.4: Ejemplos de componentes Thru-Hole.
Fuente: www.mcelectronics.com.ar

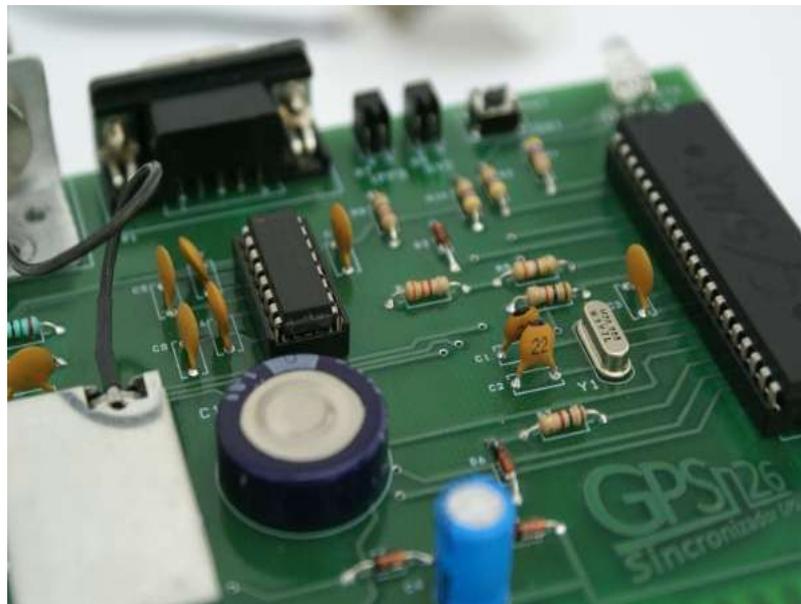


Figura 2.5: Componentes Thru-Hole en una plataforma base.
Fuente: www.traffitec.cl

VENTAJAS.-

- Tienen uniones mecánicas fuertes.
- Son económicos y asequibles.
- Son componentes que tienen mayores resistencias térmicas.

DESVENTAJAS.-

- Limitan el área de encaminamiento disponible para las trazas de señal entre los componentes.
- La perforación adicional requerida hace que las tarjetas electrónicas sean más caras de producir.

2.5.2 ENCAPSULADOS DE MONTAJE SUPERFICIAL SMD/SMT).-

Son todos aquellos componentes que se montan en forma superficial, es decir sin necesidad de hacer agujeros en la placa base de la tarjeta. Tienen la ventaja de que pueden montarse por ambos lados de la tarjeta escudo, además de ser más pequeños que los Thru-Hole, lo que permite hacer circuitos más pequeños y más densos. Se los utiliza frecuentemente para diseños en alta frecuencia debido a su tamaño reducido.



Figura 2.6: Ejemplos de componentes SMD/SMT.
Fuente: www.u.jimdo.com



Figura 2.7: Componentes SMD/SMT en una plataforma base.

Fuente: Alex Méndez

VENTAJAS.-

- Son componentes sumamente pequeños.
- Proporcionan menor costo y menor tiempo de producción para cada tarjeta escudo.
- Mejor comportamiento mecánico y más resistencia a vibraciones y golpes.
- Son más fáciles de ensamblar en la tarjeta.
- Generan menos agujeros en la placa base de la tarjeta.
- Ahorran espacio y longitud de pistas de cobre.
- Son componentes que están preparados para las últimas tecnologías.
- Minimizan la inductancia y la capacitancia parásita de cables conductores, que pueden menoscabar la función del circuito.

DESVENTAJAS.-

- El reducido tamaño, implica que la superficie de disipación también es menor, y normalmente la resistencia térmica entre el interior del componente y el exterior es más grande.
- Al ser sus conexiones más pequeñas, requieren de técnicas especializadas para su ensamblaje en la tarjeta.
- La reparación de este tipo de componentes es más difícil y costosa.
- No están diseñados para manejar altas potencias o voltajes, por lo que en ciertas ocasiones se combinan con los componentes Thru-Hole.
- Requieren un mayor control de la temperatura al ser ensamblados en la placa de la tarjeta escudo.

2.5.3 PISTAS DE UN CIRCUITO IMPRESO.-

Son líneas continuas y laminadas de cobre que están constituidas por un material conductor, para poder conectar eléctricamente a los componentes de una tarjeta escudo y sostenerlos mecánicamente.

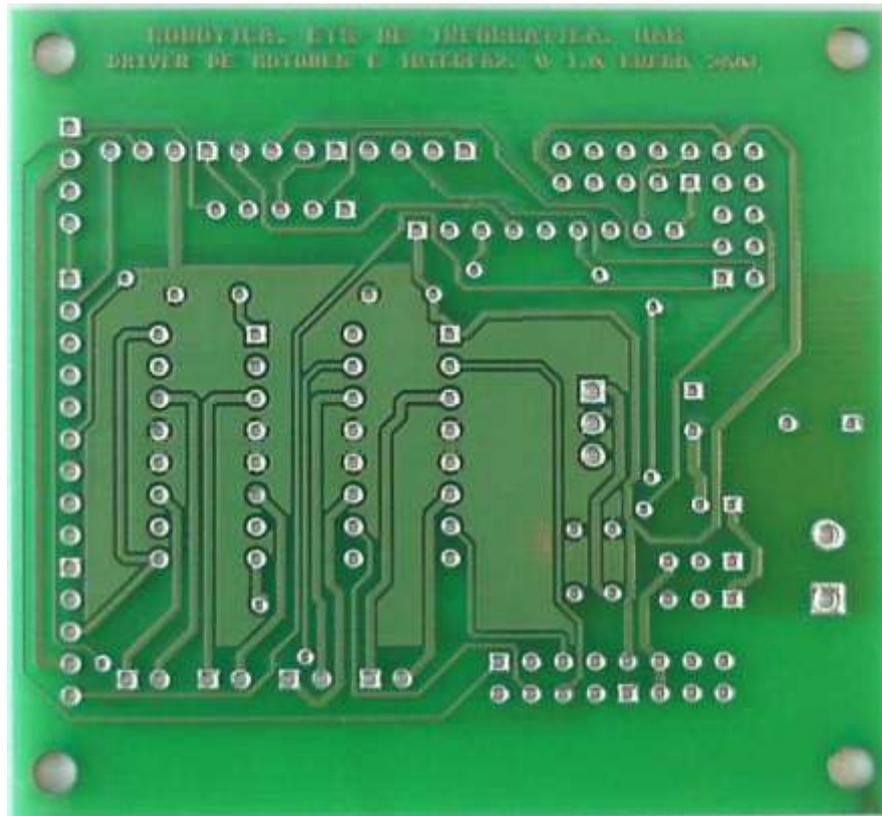


Figura 2.8: Caminos de circuito impreso en una plataforma base.
Fuente: www.pcb.electrosoft.cl

2.5.3.1 PADS.-

F. Barra (2011) "Un pad es una superficie de cobre en un circuito impreso o PCB que permite soldar o fijar la componente a la placa base". Existen dos tipos de pads; los thru-hole y los smd (montaje de superficie).

Los pads thru-hole están pensados para introducir el pin de la componente para luego soldarla por el lado opuesto al cual se introdujo. Este tipo de pads es muy similar a una via thru-hole.

Los pads smd están pensados para montaje superficial, es decir, soldar la componente por el mismo lado de la placa en donde se emplazó.

A continuación se muestran en la figura 4, cuatro componentes. La componente IC1 y R1 tienen 8 y 2 pads SMD respectivamente, mientras que ambas componentes Q1 y PW tienen 3 pads thru-hole.

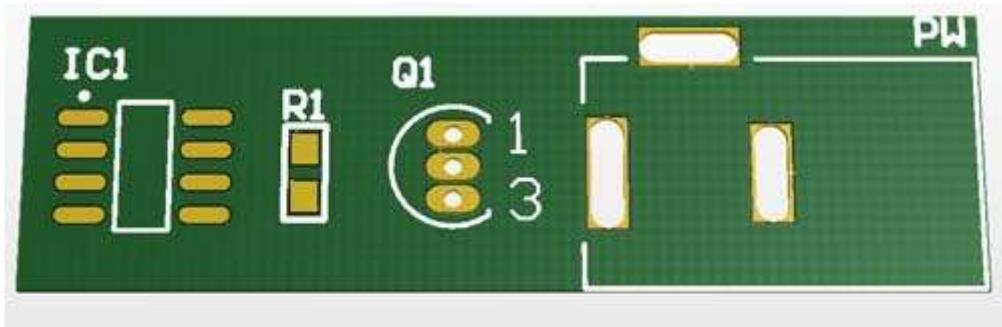


Figura 2.9: Pads SMD y THT.
Fuente: www.pcb.electrosoft.cl

2.5.3.2 Caminos de cobres (pistas o tracks).-

Según M. Batess (2006) "Un track es un camino conductor de cobre que sirve para conectar un pad (donde descansa el pin o terminal de un componente) a otro". Los tracks pueden ser de distinto ancho dependiendo de las corrientes que fluyen a través de ellos.

Cabe destacar, que en altas frecuencias es necesario calcular el ancho del track de forma que exista una adaptación de impedancias durante todo su recorrido (más de este tema en una futura publicación).

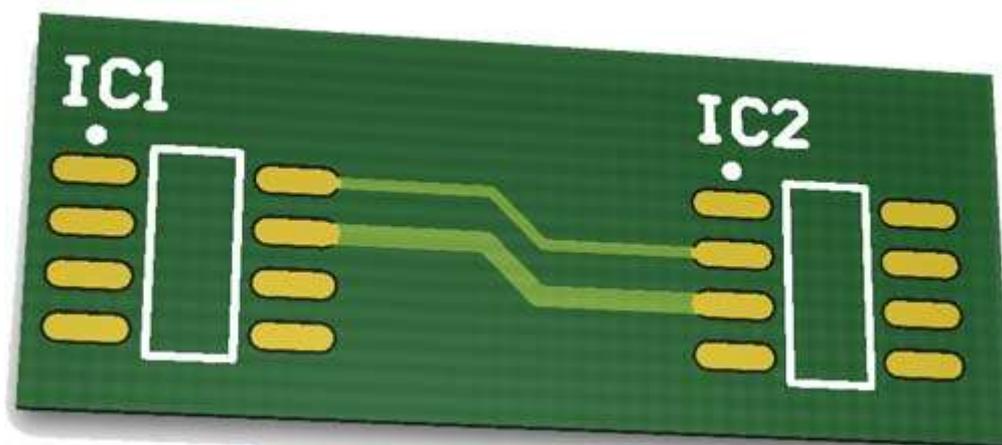


Figura 2.10: Tracks que interconectan 2 circuitos impresos.
Fuente: www.pcb.electrosoft.cl

2.5.3.3 Perforaciones Metalizadas a través de orificio (Thru-hole Vias o Full Stack Vias).-

Cuando se debe realizar una conexión de un componente que se encuentra en la capa superior de la PCB con otro de la capa inferior, se utiliza una vía. E. Palacios (2005) "Una vía es una perforación metalizada (en inglés, plated via) que permite que la conducción eléctrica no se interrumpa cuando se pasa de una superficie a otra". En la figura 6 puede apreciarse como salen 2 tracks desde los pads de un chip que se encuentra en la capa superior de la PCB, que luego de pasar por 2 vías, se conectan a los pads del chip que se encuentra en la capa inferior.

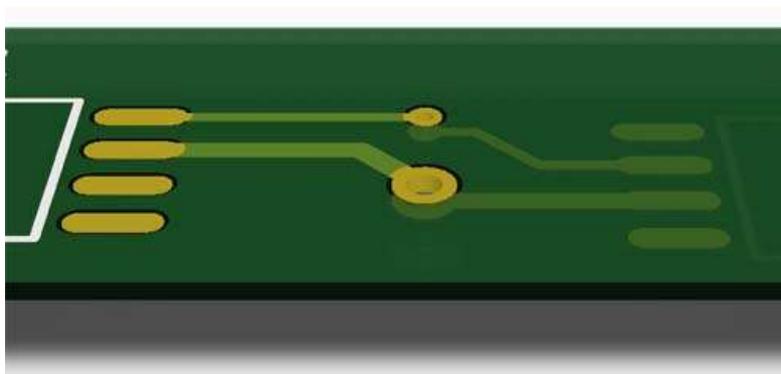


Figura 2.11: Dos chips en caras opuestas se conectan, atravesando la placa con vías.
Fuente: www.pcb.electrosoft.cl

En la tabla se muestra una vista más detallada de una sección de circuito impreso. Los colores significan:

Tabla 2.2: Secciones de circuito impreso.

Color	Descripción
Verde	máscaras de soldado de las capas superior (top) e inferior (bottom)
Rojo	corresponde a la capa superior (top)
Violeta	corresponde a la segunda layer que en este caso es un plano de energía (i.e. Vcc o Gnd)
Amarillo	corresponde a la tercera layer que en este caso es un plano de energía (i.e. Vcc o Gnd)
Azul	corresponde a la capa inferior (bottom)

Fuente: www.pcb.electrosoft.cl

Este circuito impreso PCB es de 4 capas, y se puede apreciar como un track de la capa superior (top) atraviesa la placa donde sale un track en la capa inferior (bottom).

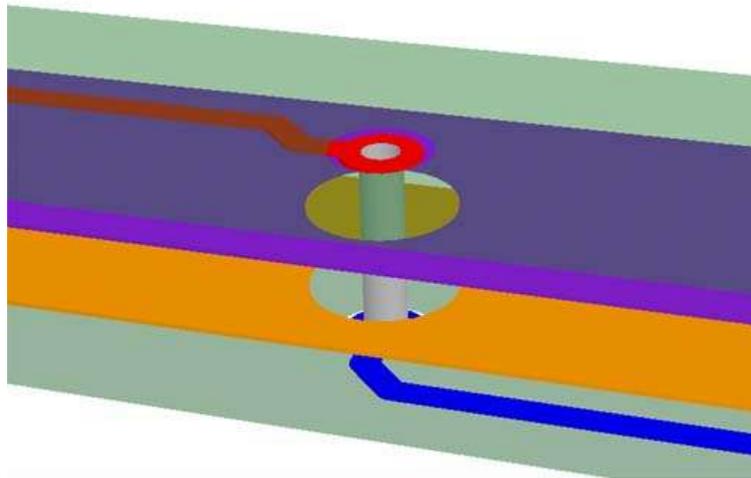


Figura 2.12: Track de la capa superior atravesando la PCB y saliendo en la capa inferior Perforaciones Metalizadas ciegas (Blind Vias).
Fuente: www.pcb.electrosoft.cl

2.5.3.4 Perforaciones Metalizadas ciegas (Blind Vias).-

En diseños complejos de alta densidad es necesario utilizar más de 2 layers como hemos mostrado en la figura 7. Generalmente en los diseños de sistemas multicapas donde hay muchos integrados, se utilizan planos de energía (de Vcc's o de Gnd) para evitar tener que routear muchos tracks de alimentación. Dicho en otras palabras es más fácil y seguro conectarse a la alimentación directamente bajo del chip (capa contigua a la superior) que con tracks largos hacia el PDS (Power Delivery System).

También hay veces que se debe routear un track de señal desde una capa externa a una interna con el mínimo largo posible debido a problemas que se generan cuando se trabaja en frecuencias altas que afectan la integridad de las señales (más de esto en otra publicación futura).

Para esos tipos de conexionado se utilizan vias ciegas, las cuales permiten conectar una capa externa a una capa interna. La via ciega comienza en una capa externa y termina en una capa interna, es por eso que se llama ciega. Para darse cuenta si una via es ciega, puede ponerse el circuito impreso contra una fuente de luz y ver si la luz pasa a través de ella. En el caso de que la luz no traspase la PCB estamos hablando de una via ciega, en caso contrario, de una via thru-hole.

Es muy útil utilizar este tipo de vias en diseño de circuitos impresos cuando no se posee de mucho espacio para emplazar componentes, por lo que se debe poblar la placa por componentes en ambas caras. Si las vias fuesen del tipo thru-hole utilizaría espacio extra el hecho de que una via atravesara la placa completa.

En la figura podemos apreciar 3 vias en este circuito impreso. Si vemos de izquierda a derecha, la primera via que veremos es una thru-hole o fullstack. La segunda es una via que nace en la capa superior y termina en la segunda capa, por lo que decimos que es una 1-2 blind via. Por último la tercera es una via que nace en la capa inferior y termina en la tercera, por lo tanto es una 3-4 blind via.

Es importante destacar que las blind vias se suelen manufacturar en layers consecutivas, en otras palabras L1 L2, L3 L4, Ln-1 Ln.

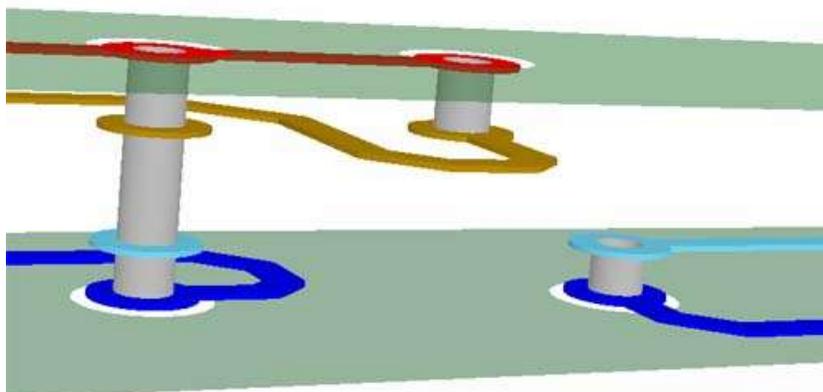


Figura 2.13: Comparación de una Thru-hole via con una Blind via.
Fuente: www.pcb.electrosoft.cl

La desventaja que tiene el utilizar este tipo de vias es su alto costo en comparación a la alternativa de vias thruhole.

2.5.3.5 Perforaciones Metalizadas enterradas u ocultas (Buried Vias).-

Estas vias son similares a las ciegas, la diferencia es que no nacen ni terminan en ninguna capa externa o de superficie. Si vemos de izquierda a derecha, la primera es una thru-hole o fullstack y la segunda es una 1-2 blind, al igual que en el ejemplo anterior. La tercera en este caso es una 2-3 buried via que nace en la segunda capa y termina en la tercera.

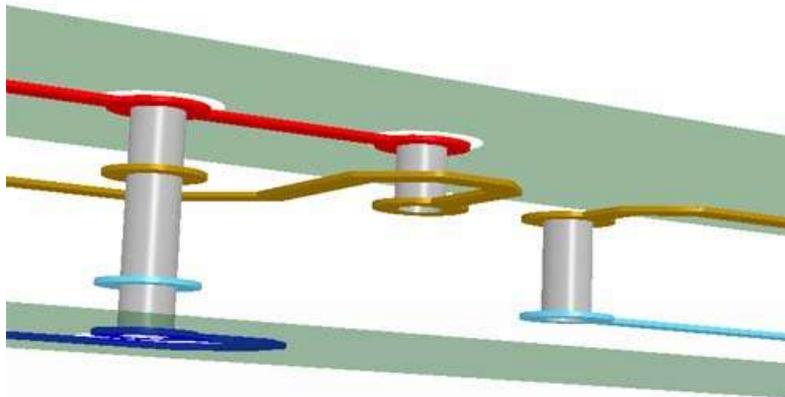


Figura 2.14: Buried via.
Fuente: www.pcb.electrosoft.cl

Es importante destacar que las buried vias se suelen manufacturar en layers consecutivas (i.e. L1 L2)

La desventaja que tiene el utilizar este tipo de vias es su alto costo en comparación a la alternativa de vias thruhole.

2.5.4 APILAMIENTOS DE CABECERAS.-

Todas las tarjetas deben de tener apilamientos de cabeceras. Muchos escudos están equipados con encabezados regulares separatistas que sobresalen por debajo de la tarjeta, pero no proporcionan un lugar en la parte superior donde otro escudo puede ser conectado.

Por eso esta característica se debe utilizar siempre en la estructura del diseño de una tarjeta escudo, como se muestra a continuación.



Figura 2.15: Apilamientos de cabeceras de una plataforma base.
Fuente: www.pcb.electrosoft.cl

2.5.5 CONTENCIÓN DE PATILLAS EN LA PLATAFORMA HARDWARE.-

Es comprobar la asignación de patillas para los módulos y las tarjetas escudos que se desean utilizar, y asegurar de que estas tarjetas no van a estar conectadas por las mismas patillas y causar interrupción en la plataforma hardware base. Puede ser muy difícil determinar a través de la esquemática del fabricante qué patillas se pueden utilizar.

Solo se pueden compartir las patillas que contengan una tensión o voltaje de referencia; ya que las tarjetas escudos no pueden compartir las mismas patillas de datos, porque no tendrían comunicación con el microcontrolador.

Si las tarjetas escudos utilizan comunicaciones SPI todos pueden coexistir en el mismo bus SPI, con la excepción de la SS (Slave Select)

de la línea (a veces llamado CS o Chip Select), que tendrá que ser configurado de forma diferente para cada tarjeta.

El MISO, SCLK, MOSI, y todo puede ser compartido por múltiples escudos. Si se dispone de varias pantallas que utilizan SPI y también se usa el mismo pin de SS, se debe modificar una de las tarjetas.

Un enfoque típico es cortar el camino que va a la SS en una de las tarjetas y utilizar un cable de puente para conectarlo a una patilla de alternativa, luego, modificar el código software para que la tarjeta pueda utilizar esa conexión en comunicación con la otra tarjeta.



Figura 2.16: Patillas que contienen sus respectivas pistas de conexión.
Fuente: www.microsystem.com

2.5.6 SOFTWARE.-

Una plataforma hardware debe tener una buena cantidad de sobrecarga de software, a veces se puede tornar un poco complicado el combinar tarjetas escudos, si se utiliza un montón de memorias FLASH o SRAM, ya

que se genera conflictos de interrupciones o requisitos de tiempo ajustados.

Para determinar el valor nominal se debe saber el funcionamiento interno de todas las bibliotecas que participan muy íntimamente.

2.5.7 ENERGÍA.-

Las plataformas hardware con microcontroladores requieren una cantidad de energía mínima o máxima de alimentación, que se puede disponer desde la misma plataforma base para su correcto funcionamiento, y en caso de no obtenerla desde la plataforma se la debe alimentar externamente a través de conectores como el USB o de fuentes de voltaje.

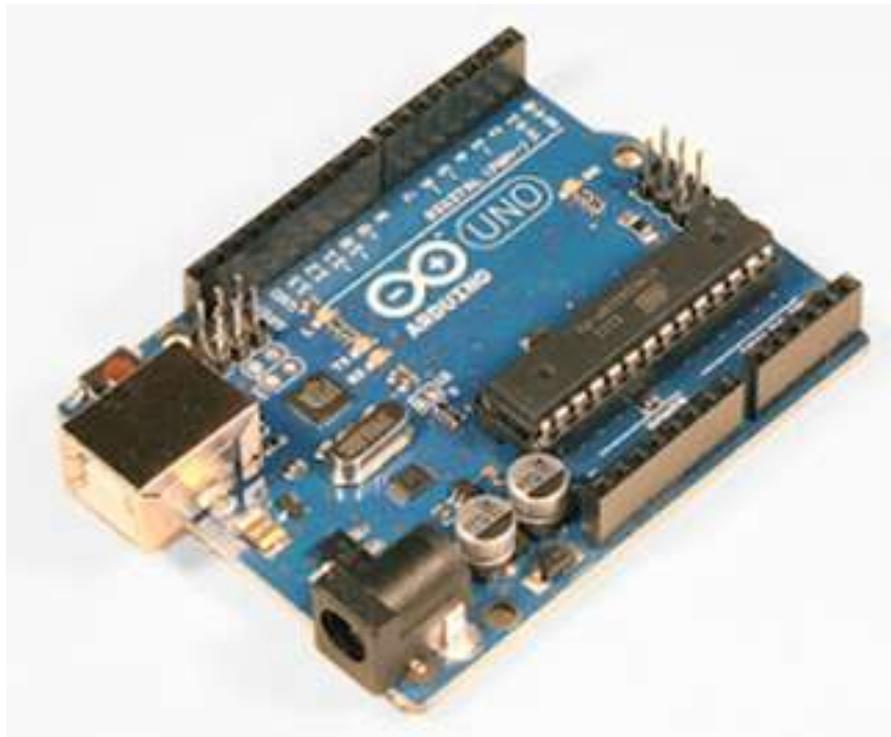


Figura 2.17: Requerimiento de energía.
Fuente: www.microsystem.com

2.6 COMPILADOR PIC-C CCS.-

G. Breijo (2008) "El compilador PIC-C CCS, denominado PCWH, utiliza un lenguaje de códigos muy eficiente, fácil y manejable, pero sobre todo de alto nivel, basado en C ANSI" que contiene las funciones y librerías necesarias para el diseño de cualquier aplicación basada en microcontroladores PIC: matemáticas, control de protocolos serial, I2C, etc. Además, suministra los controladores (drivers) para diversos dispositivos como LCD, convertidores AD, relojes en tiempo real, EEPROM serie, etc.

La versión PCWH admite todos los dispositivos PIC de las gamas: baja 12XXX, media 16XXX y alta 18XXX, con núcleos de 12, 14, 16 bits respectivamente e incluso núcleos hasta con 24 bits como son la familia dsPIC30F2010. Incluye IDE (Entorno Integrado de Desarrollo) en Windows lo cual facilita la edición y depuración de los programas, líneas de comando PCB,PCM, PCH y PCD, un terminal de comunicación serial perfecto; así como ayuda a la configuración de los dispositivos. Cabe recalcar que las instrucciones y el manual de ayuda en el programa se encuentran totalmente en inglés.

Este compilador soporta dispositivos de todas las familias de los microcontroladores PIC y en su última versión (4.1) se incluyen librerías para poder controlar más de 300 modelos de microcontroladores PIC.

El compilador incluye funciones para acceder al hardware de los procesadores PIC, tal como ***read_adc ()*** para leer el valor de un convertor A/D. La entrada y salida de la comunicación se maneja describiendo las características de los puertos en un PRAGMA. Funciones tales como ***input ()*** y ***output_high ()*** mantienen apropiadamente los registros tri-Estado. Las variables, incluyendo estructuras pueden ser

directamente mapeadas a la memoria tal como los puertos de entrada y salida para representar mejor la estructura del Hardware en C.

La velocidad del reloj del microcontrolador se puede especificar en un PRAGMA para permitir que las funciones incorporadas retrasen un número dado de microsegundos o milisegundos. Las funciones de E/S serie permiten que funciones estándar como **get ()** y **printf ()** sean usadas para RS-232.

El transceptor serie del Hardware se usa en las partes que aplican cuando es posible. Para otros casos el compilador genera un transceptor serie de software. Los operadores estándar de C y las funciones estándar incorporadas se optimizan para producir código muy eficiente para funciones de bits y de E/S.

Pueden implementarse funciones in-line o separadas, que permiten optimizar según mejoras en la ROM o en la velocidad. Los parámetros de las funciones se pasan en registros reusables. Las funciones in-line con parámetros de referencia se implementan eficientemente sin sobrecarga de memoria.

Durante el proceso de enlazado se analiza la estructura del programa. Las funciones que se llaman unas a otras con frecuencia se agrupan juntas en el mismo segmento de página. La herramienta transparente al usuario maneja funciones a través de las páginas automáticamente. Las funciones se pueden implementar in-line o separadas. La RAM se reserva eficientemente para determinar cuántas ubicaciones pueden ser reutilizadas. Las cadenas constantes y tablas se almacenan en la ROM del dispositivo.

El compilador produce principalmente tres tipos de archivos:

- Archivos con extensión **.hex** que permitirá grabar el programa ejecutable en el PIC por medio del uso de un programador.

- El archivo **.asm** contendrá un listado en assembler del programa compilado con la información del mapeo de memoria. Estos archivos son muy útiles para el *debugging* de los programas y para determinar la cantidad de pasos de programas (ciclos de ejecución) contiene la aplicación.
- Los archivos con extensiones **.pre** contienen la información procesada del programa, **#defines**, **#includes**, etc. La cual es expandida y guardada en el archivo.
- La salida en **.hex** y los archivos de depuración son seleccionables y compatibles con emuladores y programadores populares incluyendo MPLAB IDE para depuración a nivel de fuente. PCW incluye un poderoso IDE bajo Windows. El compilador requiere Windows 95, 98, ME, NT4, 2000, XP, VISTA, W7 o Linux para su normal funcionamiento.

PRINCIPALES VENTAJAS.-

- Está basado en el ANSI C.
- Soporte completo de las familias de Microcontroladores PIC.
- Salida Assembly.
- Biblioteca de funciones pre-compiladas y directivas de las que disponen.
- Funciones listas para usarse, ahorra mucho trabajo al programador.
- Industria estándar INTEL Hex 8 bit Mergerd format (INHX8M).

- Soporta interrupciones internas y externas.
- Tipos de datos 8, 6, 16 bit int, char, long, punteros, unsigned, etc.
- Inserción de código assembly asm ().
- Operadores aritméticos, incluyendo multiplicación, división, modulo y otros.
- Las variables y funciones no utilizadas son borradas automáticamente.
- Reutilización de RAM.
- Instrucciones simples.
- Se tiene un aprovechamiento eficiente de los recursos del PIC.

2.7 PROGRAMACION EN PIC-C CCS.-

La programación en lenguaje C, para PIC, es bastante similar en su estructura y sintaxis a la programación tradicional para ordenadores, una de las diferencias está en las librerías creadas específicamente para los microcontroladores PIC como por ejemplo para el LCD, teclado matricial, bus I2C, etc.

La función ***main ()*** es imprescindible en todo programa escrito en lenguaje C, pues es la función principal, desde aquí se puede hacer el llamado a otras funciones. Para crear un programa en lenguaje C, hay que seguir los pasos siguientes:

1. Especificaciones del programa.
2. Hacer organigrama.

3. Escribir el código fuente.
4. Compilar + enlazar.

5. Depurar errores, si es que los hay.

2.7.1 ESTRUCTURA DE UN PROGRAMA EN CCS.-

G. Breijo (2008) "Para escribir un programa en C con el CCS se deben tener en cuenta una serie de elementos básicos de su estructura", que se detallan a continuación:

2.7.1.1 DIRECTIVAS DE PROCESADO.-

Controlan la conversión del programa a código máquina por parte del compilador.

2.7.1.2 PROGRAMAS O FUNCIONES.-

Conjunto de instrucciones. Puede haber uno o varios; en cualquier caso siempre debe haber uno definido como principal mediante la inclusión de la llamada *main ()*.

2.7.1.3 INSTRUCCIONES.-

Indican cómo se debe comportar el PIC en todo momento.

2.7.1.4 COMENTARIOS.-

Permiten describir lo que significa cada línea del programa.

2.7.2 TIPOS DE DATOS.-

CCS C acepta los tipos de datos que se detallan en la siguiente tabla:

Tabla 2.3: Tipos de Datos.

TIPO	TAMAÑO	RANGO	DESCRIPCION
Int1 Short	1 bit	0 a 1	Entero de 1 bit
Int Int8	8 bit	0 a 255	Entero
Int16 Long	16 bit	0 a 65.535	Entero de 16 bit
Int 32	32 bit	0 a 4.294.967.295	Entero de 32 bit
Float	32 bit	$\pm 1.175 \times 10^{-38}$ a $\pm 3.402 \times 10^{+38}$	Coma flotante
Char	8 bit	0 a 255	Carácter
Void	-	-	Sin valor
Signed Int8	8 bit	-128 +127	Entero con signo
Signed Int16	16 bit	-32768 a 32767	Entero largo con signo
Signed Int 32	32 bit	-2^{31} a $(2^{31} + 1)$	Entero 32 bit con signo

Fuente: Libro Compilador CCS C.

2.7.3 LAS CONSTANTES.-

Las constantes se pueden especificar en decimal, octal, hexadecimal o en binario, como se detalla en la siguiente tabla:

Tabla 2.4: Tipos de Constantes en CCS C.

123	Decimal
0123	Octal (0)
0x123	Hexadecimal (0x)
0b010010	Binario (0b)
'/x'	Carácter
'/010'	Carácter octal
'/xA5'	Carácter hexadecimal

Fuente: Manual CCS C

Además, se pueden definir constantes con un sufijo:

Tabla 2.5: Definición de Constantes con un sufijo en CCS C.

Int8	127U
Long	80UL
Signed INT16	80L
Float	3.14F
Char	Con comillas simples 'C'

Fuente: Manual CCS C.

También se definen caracteres especiales, algunos como:

Tabla 2.6: Caracteres Especiales en CCS C.

/n	Cambio de línea.
/r	Retorno de carro.
/T	Tabulación.
/b	Backspace.

Fuente: Manual CCS C.

2.7.4 VARIABLES.-

G. Breijo (2008) Las variables se utilizan para nombrar posiciones de memoria RAM; se deben declarar obligatoriamente, antes de ser utilizadas; para eso se debe indicar el nombre y el tipo de dato que se utilizará. Se definen de la siguiente forma:

TIPO NOMBRE_VARIABLE [=VALOR INICIAL]

TIPO.- Hace referencia a cualquiera de los tipos de datos visto.

NOMBRE_VARIABLE.- Puede ser cualquiera y el valor inicial es opcional; como se muestra en el siguiente ejemplo:

Float temp_limit=500.0

Las variables definidas en un programa pueden ser de tipo LOCAL o GLOBAL. Las variables locales sólo se utilizan en la función donde se encuentran declaradas; las variables globales se pueden utilizar en todas las funciones del programa. Ambas deben declararse antes de ser utilizadas y las globales deben declararse antes de cualquier función y fuera de ellas. Las variables globales son puestas a cero cuando se inicia la función principal *main ()*.

Las variables pueden ser definidas con:

- **AUTO.-** Es usada por defecto, no hace falta que se indique; donde la variable existe mientras la función esta activa. Estas variables no se inicializan a cero. Su valor se pierde cuando se sale de la función.
- **STATIC.-** Una variable local se activa como global, se inicializa a cero y mantiene su valor al entrar y salir de la función.
- **EXTERN.-** Permite el uso de variables en compilaciones múltiples.

2.8 EAGLE: SOFTWARE PARA DISEÑOS DE PCB'S.-

Todas las tarjetas escudos, incluida la plataforma hardware que contiene al microcontrolador están diseñadas en un software de dibujo asistido llamado Eagle PCB. Usando este programa se puede crear circuitos impresos partiendo de un esquemático previamente diseñado, utilizando herramientas de fácil uso.

2.8.1 INTERFAZ DE USUARIO DE EAGLE.

M. Guadilla (2000) "Eagle se ha diseñado de tal modo que cualquier acción se realiza mediante un comando". Normalmente el usuario selecciona los comandos de la barra de herramientas que se encuentra a la izquierda de la ventana de trabajo, haciendo clic sobre los iconos. Pero existe otra opción, que es introducir los comandos mediante texto en la barra superior de la ventana de trabajo.

CAPÍTULO 3

DISEÑO DE PLATAFORMA HARDWARE CON PIC18F4520

3.1 INTRODUCCIÓN.-

La plataforma entrenadora implementada en el presente trabajo es un conjunto didáctico en donde se puede trabajar con los μC PIC. Concretamente, y como hemos mencionado previamente, está diseñada para trabajar con un PIC18F4520.

En la placa se incluyen una gran variedad de periféricos para probar sobre ésta, los circuitos con μC , tanto a nivel software como hardware. Todos estos periféricos se encuentran directamente conectados con las diferentes patillas del μC , por lo que el usuario no deberá hacer ningún tipo de conexión adicional, lo que hace aún más fiable su funcionamiento y permitirá centrarse únicamente en los errores de software.



Figura 3.1: Plataforma Hardware con PIC18F4520.

Fuente: Autor

Cabe remarcar que, en la implementación de la presente propuesta, se ha escogido una plataforma que proporcione facilidad al momento de trabajar en la parte práctica.

Para realizar el diseño de la plataforma base que contiene al PIC18F4520, primero se busca el respectivo Data sheet y de acuerdo a la información proporcionado por el fabricante se hizo las conexiones a las patillas del PIC.

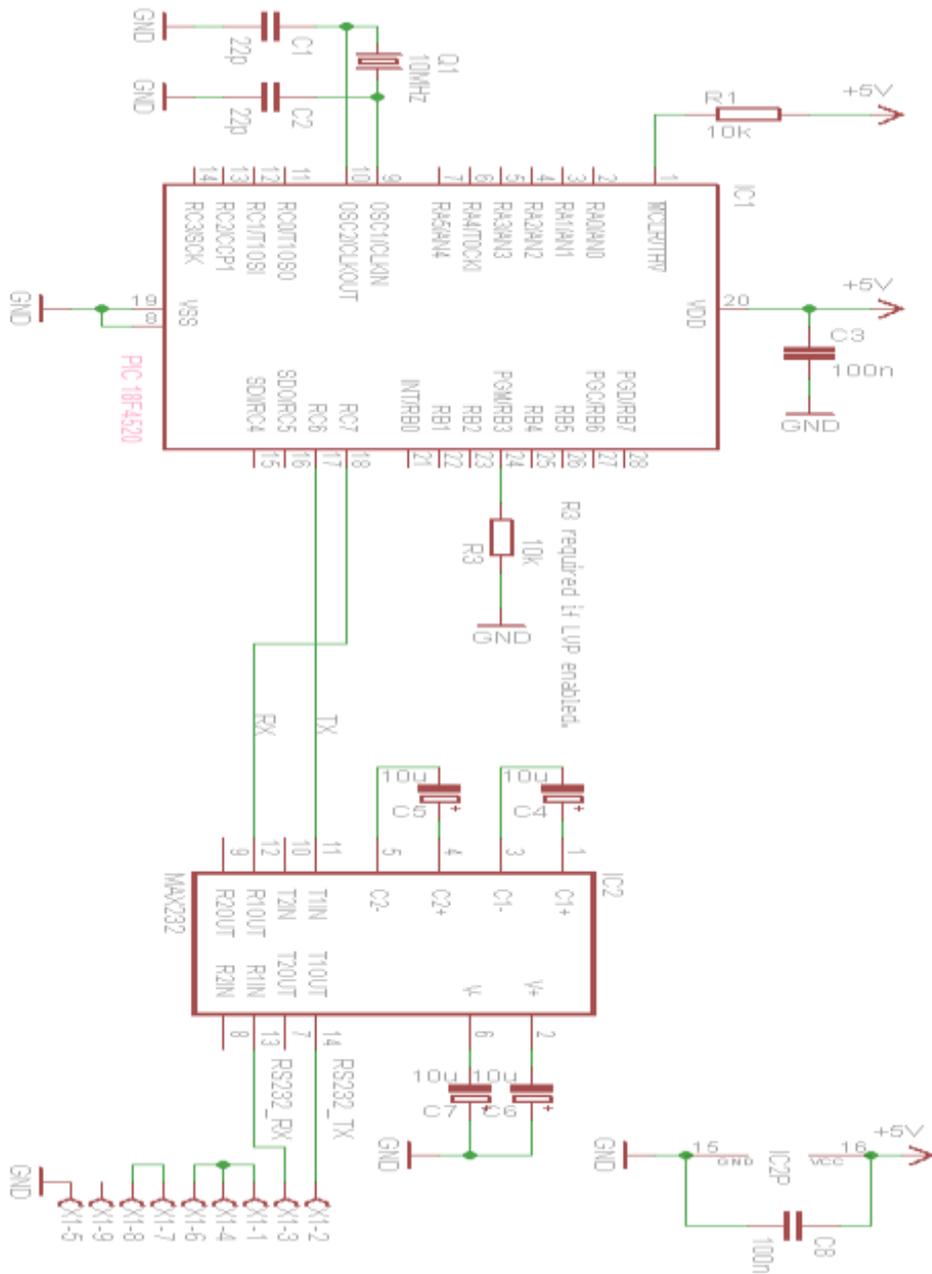
Este PIC es el que vamos a usar para que sea el cerebro de nuestro entrenador. Este PIC es de la gama alta por lo que nos dará grandes prestaciones y un sinfín de posibilidades para poder trabajar con total comodidad.

Características generales:

- Memoria Flash de 32K bytes
- Memoria SRAM de 2048 bytes
- Memoria EEPROM de 256 bytes
- I/O: 35
- Conversores A/D: 13
- Módulos ECCP: 1
- Soporta SPI
- Soporta I2C
- Soporta USB
- Temporizadores de 8bits: 1
- Temporizadores de 16bits: 3

Se traza una línea en cada patilla del PIC, y las nombramos con etiquetas cada una (RA0, RA6). Luego se procede hacer lo mismo, pero en este caso con bus unidos a los puertos.

Figura 3.2: Descripción de las patillas del PIC18F4520.



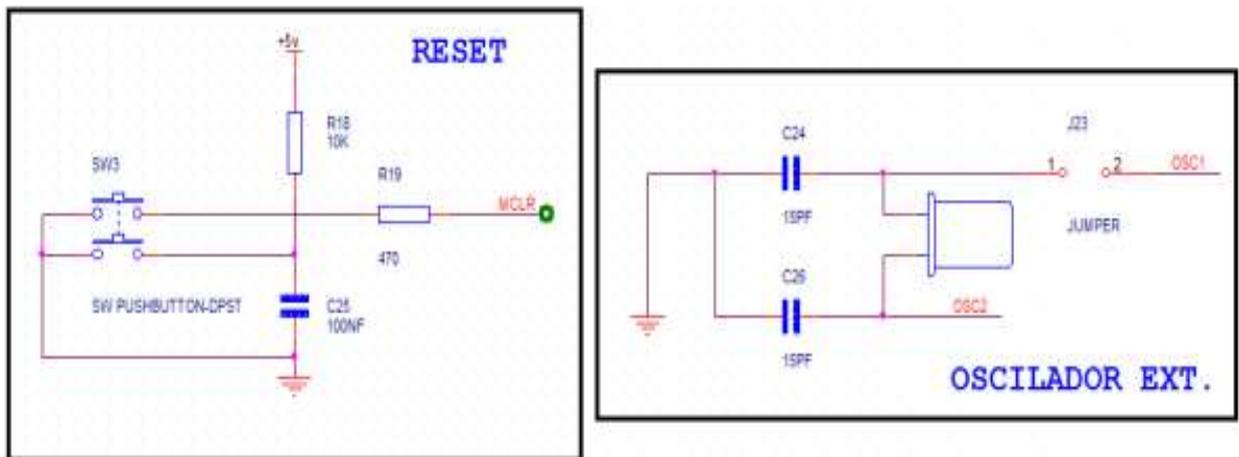
Fuente: Autor
Elaborado: Autor

Una vez obtenido el cableado del PIC se sigue diseñando pero en distintas hojas las partes que va a tener la plataforma base. Cada una será independiente pero al final se irán uniendo.

3.2 RESET Y OSCILADOR EXTERNO.

Una de esas partes es el diseño del **Reset** y el **Oscilador Externo**. El reset está compuesto por un pulsador que su función será resetear el PIC manualmente. En cuanto al oscilador, todo microcontrolador necesita que le indiquen a qué velocidad trabajar es como el motor por lo tanto, este pequeño circuito no debe faltar.

Figura 3.3: Diseño del Reset y Oscilador externo.

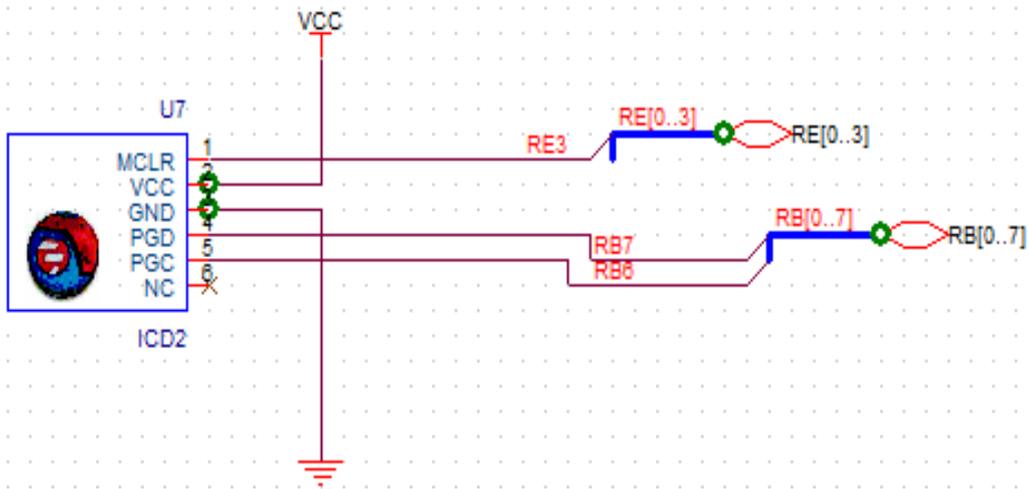


Fuente: Alex Méndez
Elaborado: Alex Méndez

3.3 COMUNICACIÓN ICD2.

Otra parte de la plataforma base es la comunicación **ICD2** una comunicación propia de Microchip la cual permite comunicarse con el PIC usando el MPLAB.

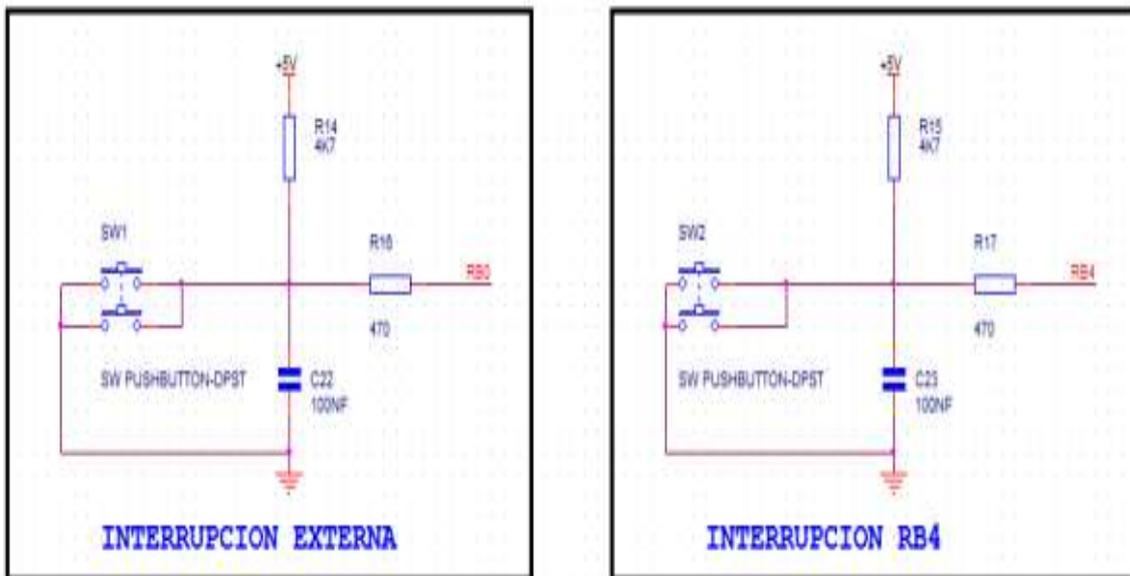
Figura 3.4: Diseño de Comunicación ICD2.



Fuente: Alex Méndez
Elaborado: Alex Méndez

El siguiente bloque del esquema es la **Interrupción externa** con un dos pulsadores uno a Rb4 y Rb0 poniendo los dos a cero.

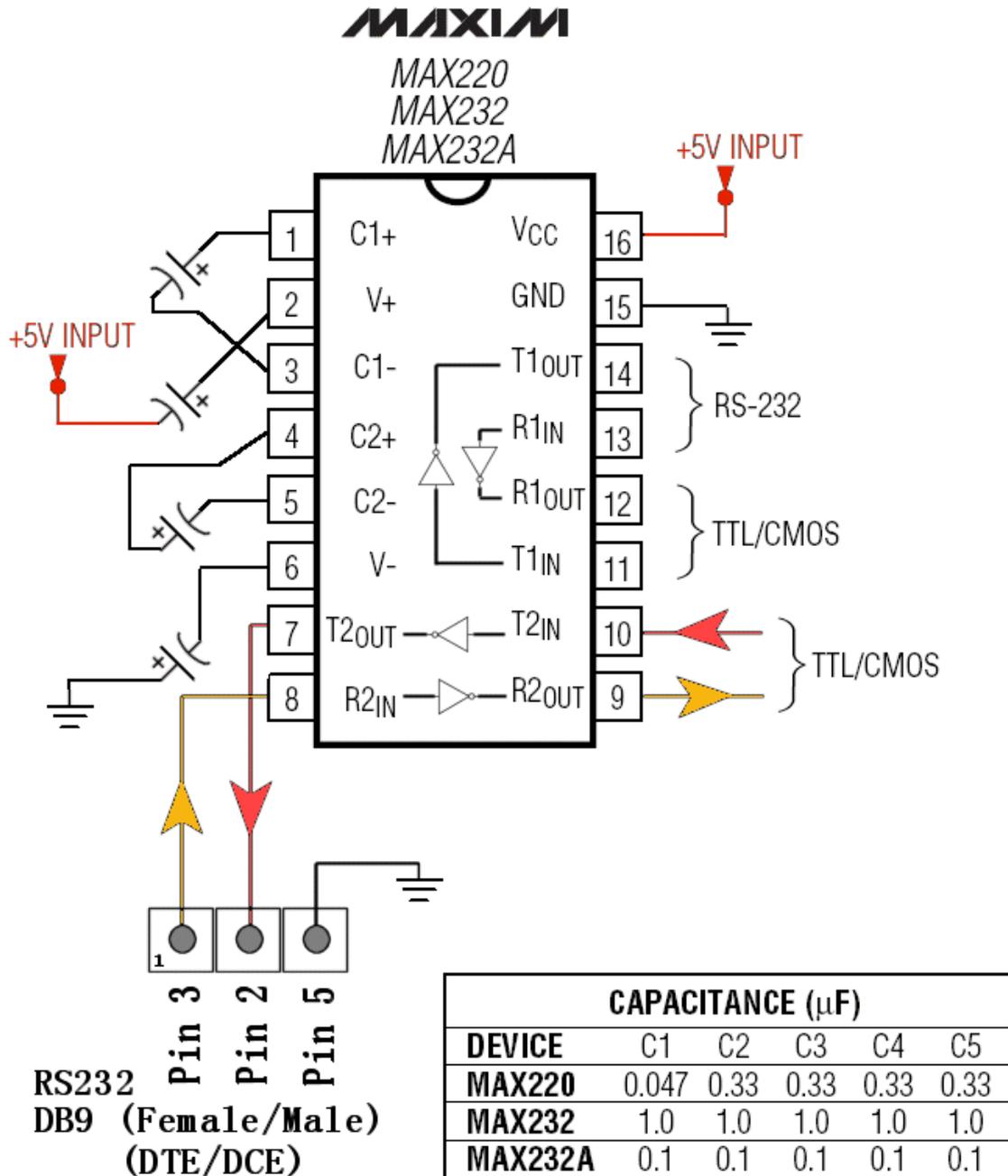
Figura 16: Diseño de la Interrupción Externa.



Fuente: Alex Méndez
Elaborado: Alex Méndez

El siguiente bloque contiene una parte de comunicación, que permite comunicar al PIC al ordenador mediante un RS232.

Figura 3.5: Esquema de Comunicación RS232.



Fuente: www.siongboon.com/projects/2006-03-06_serial_communication/
Elaborado: Alex Méndez

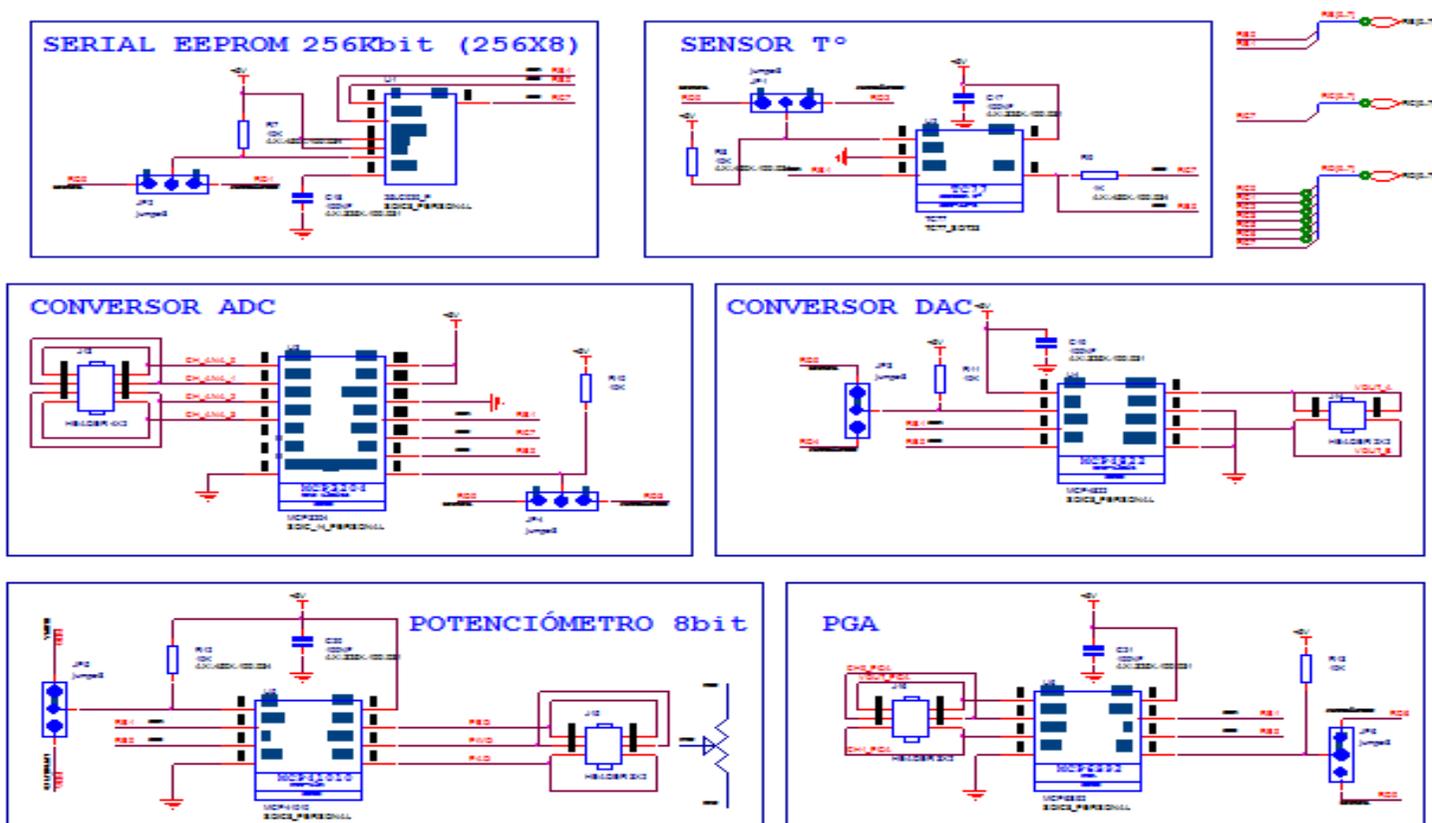
El siguiente bloque es el de expansión, que nos permitirá conectar más PIC a la placa.

El último bloque de comunicación son los SPI (serial de interfaz de periféricos), que es un estándar de enlace de datos sincronizados por un reloj. En pocas palabras el SPI nos permite la comunicación entre los circuitos integrados.

En este caso los que vamos a utilizar no están en la librería de Orcad, por lo tanto toca diseñar tal y como se hizo con el pic. Para ello necesitamos la ayuda de los fabricantes (Data Sheet) de los siguientes integrados:

- MCP3204
- TC77
- MCP4822
- MCP42010
- MCP6292
- MCP23S08

Figura 3.6: Diseño de los bloques que contendrá la plataforma base.

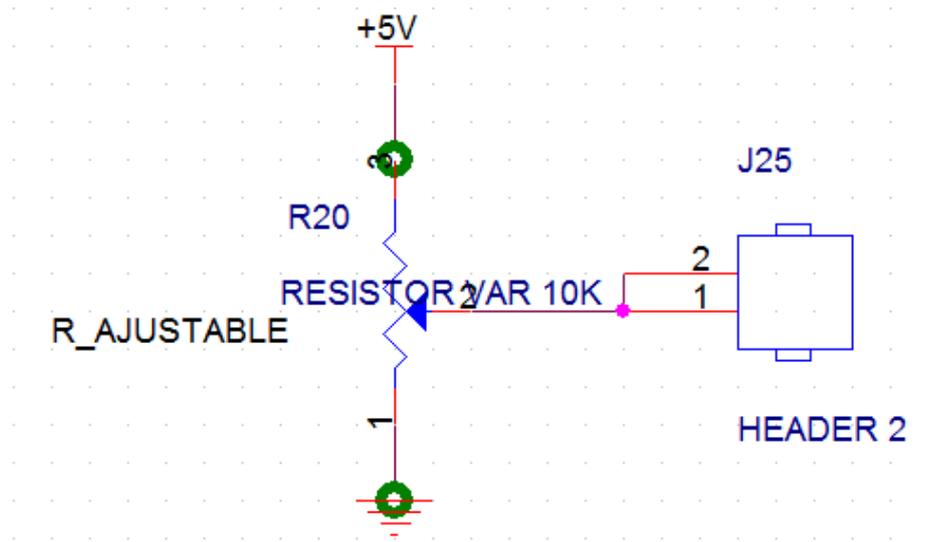


Fuente: Alex Méndez

Elaborado: Alex Méndez

Por último se agrega un potenciómetro analógico.

Figura 3.8: Diseño de potenciómetro analógico.



Fuente: Alex Méndez
Elaborado: Alex Méndez

3.4 ESPECIFICACIONES:

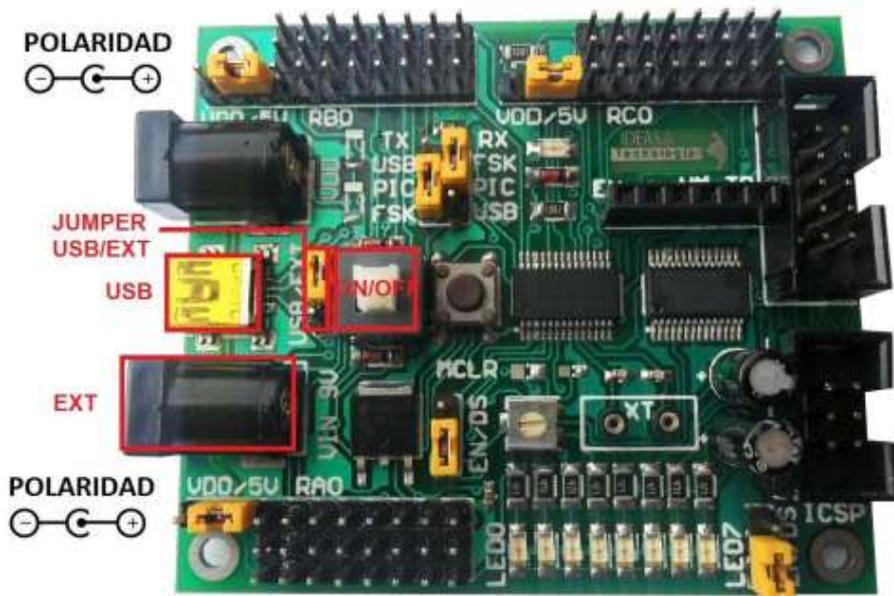


Figura 3.9: Especificaciones de la plataforma hardware.

Fuente: Autor.

- Comunicación serial asíncrona UART.
- Comunicación serial síncrona SPI e I2C.
- Comunicación ONE WIRE y USART.
- Comunicación inalámbrica RX y TX con módulos.
- FSK y ASK.
- 1 Potenciómetro integrado.
- 10 entradas analógicas.
- 24 entradas y salidas digitales.
- Tiene 8 leds que indican las salidas digitales.
- Controla 4 servomotores.
- Controla 2 motores DC (Dirección y Velocidad).
- Programación ICSP in circuit.
- Reset manual.
- 1 Switch de ON/OFF.
- 1 Led indicador de power

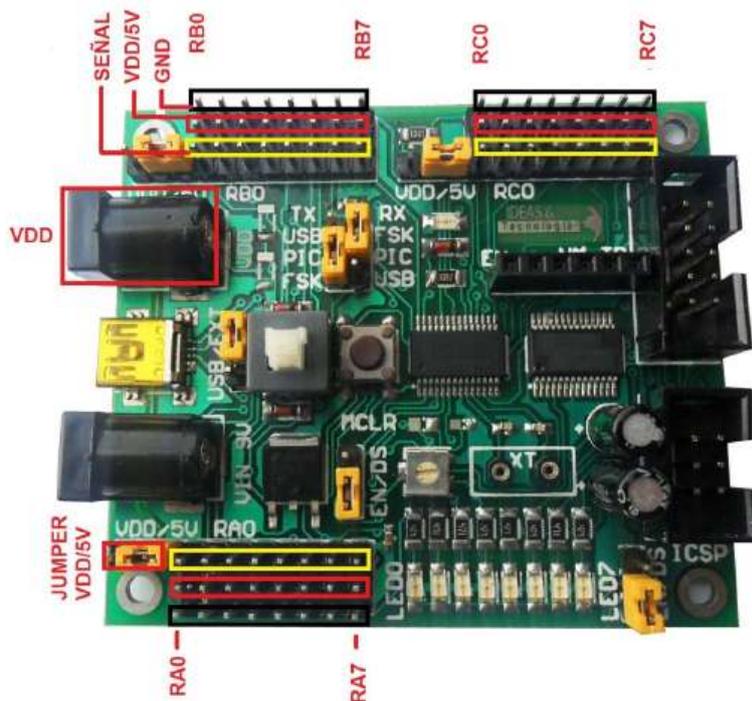


Figura 3.10: Especificaciones de puertos de la tarjeta hardware.
Fuente: Autor.

3.5 RESET (MCLR).

Este botón posee un resistor pull up y está conectado al PIN MCLR. Para utilizar este botón es necesario que se lo habilite mediante software.

CAPITULO 4

SOFTWARE Y PROGRAMACIÓN DE LA PLATAFORMA HARDWARE

4.1 Comunicación entre el PIC18f4520 y la memoria EEPROM 25LC020A mediante el protocolo SPI:

En la programación del software vamos a leer y escribir datos en una memoria EEprom 25LC020A que trabaja mediante el bus SPI y los vamos a programar atreves del PIC18f4520.

Para acceder al dispositivo es necesario hacerlo mediante el bus SPI. Las señales requeridas son de una entrada de reloj (SCK), además de los datos de entrada (SI) y los datos de salida (SO) y para acceder al dispositivo se controla mediante el pin de selección (CS).

4.2 Configuración de Puertos A, B, C Entradas/Salidas:

Cada puerto contiene 8 pines correspondientes a los 8 bits, a cada bit se denomina señal entrada/salida (I/O) acompañada de las patillas correspondientes a las conexiones de **+Vcc** y **Gnd**, donde Vcc puede ser seleccionable es decir utilizar el voltaje interno del módulo entrenamiento o externo mediante el Jack VDD.

- PORTA (0,1,2,3,4,5,6,7).
- PORTB (0,1,2,3,4,5,6,7).
- PORTC (0,1,2,3,4,5,6,7)22,4+3,95+0,8.

4.3 Entradas Analógicas.

- ✓ AN0=RA0, AN1=RA1, AN2=RA2, AN3=RA3, AN4=RA5.
- ✓ AN8=RB2, AN9=RB3, AN10=RB1
- ✓ AN11=RB4, AN12=RB0, AN13=RB5.

4.4 Led en Puerto B (PORTB):

Para poder utilizar los led del puerto B (PORTB), debemos reiniciar las patillas de este puerto como salidas, además debemos de habilitar el jumper LED (**EN/DS**). De esta manera quedan habilitados todos los indicadores led del módulo de entrenamiento.

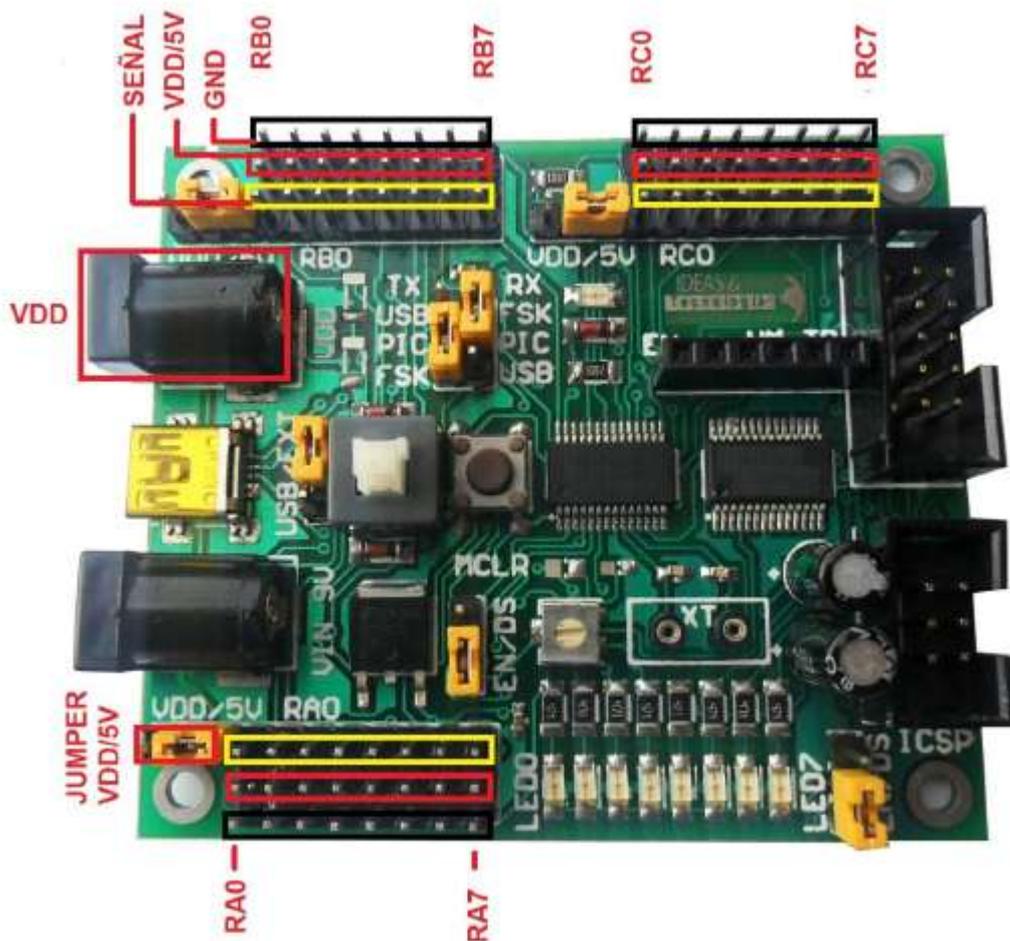


Figura 4.1: Especificaciones de puertos para la programación de la tarjeta hardware.
Fuente: Autor.

4.5 Programación:

Lo primero es escribir el código fuente del programa en lenguaje C para PIC's. Vamos a comenzar escribiendo, que en mi caso los valores que voy a escribir serán 4 datos "0, 1, 2, 3" esto quiere decir que desde el valor 0x00 hasta 0x03 en hexadecimal. Esto también se va a ir mostrando en el LCD según vaya escribiendo en la memoria y de la misma manera cuando termine de escribir pasaremos a leerlos y mostrarlos.

Para la asignación del encabezado de códigos para la programación de las tarjetas utilizamos los siguientes pasos y comandos:

1. Usamos el protocolo de comunicación serial RS232 a través del siguiente comando

`#use rs232`

2. Asignamos al protocolo la velocidad de comunicación en baudios (baud) en que vamos a transmitir los datos.

`Baud = 9600`

3. Agregamos el bit de paridad que nos permite comprobar los errores.

`Parity = N`

4. Asignamos la patilla (PIN) de la tarjeta que tiene comunicación con el puerto del microcontrolador para la salida de datos.

`Xmit = PIN_C6`

5. Asignamos a que patilla (PIN) del microcontrolador por donde va a recibir los datos.

`Rcv = PIN_C7`

6. Por último el tamaño de la cantidad de datos que recibe el microcontrolador que es de 8 bits.

`Bits = 8`

El código completo de la programación de la tarjeta se lo encontrará en la sección de ANEXOS.

CAPÍTULO 5 – COSTOS

[1] Plataforma hardware con PIC18f4520.

COSTO	
PIC18f4520	4
Componentes Electrónicos	9,56
Circuito Impreso y Diseño PCB	22
SUBTOTAL	35,56
IVA 12%	4,2672
TOTAL	39,8272

CONCLUSIONES:

- ✓ El microcontrolador es el dispositivo que más utilizamos en la actualidad, además del uso diario ellos reduce el costo de funcionalidad de muchos dispositivos electrónicos.
- ✓ Esta plataforma permite desarrollar aplicaciones y controlar su funcionamiento; es muy ideal para el aprendizaje de los estudiantes.
- ✓ La programación y las librerías que se desarrollaron fueron lo más sencillas para su entendimiento en el aprendizaje de esta plataforma hardware.
- ✓ El presente proyecto de tesis mostró el diseño y la realización práctica de una plataforma docente para el aprendizaje de μ C PIC de la firma Microchip.
- ✓ La plataforma que se desarrollo es del tipo modular, ya que permite agregar diversidad de módulos para realizar prácticas de aprendizaje.
- ✓ La plataforma hardware se diseñó con elementos electrónicos básicos para su fácil implementación.
- ✓ En el diseño de la plataforma docente se resolvió la adición o incorporación de periféricos más allá de los incluidos inicialmente mediante módulos de expansión conectados de los puertos del μ C a través de los conectores previstos en la tarjeta.
- ✓ Se desarrolló unos pequeños manuales para el uso del software que se necesita y también un par de prácticas para la enseñanza.

RECOMENDACIONES

- Usar componentes SMT para la implementación de cualquier plataforma didáctica, ya que en el inicio del proyecto se utilizó componentes THT, pero fueron sustituidos por los antes mencionados ya que dificultaban la manipulación de los módulos y conexiones.
- Revisar y verificar que las conexiones realizadas se encuentran bien efectuadas, con el fin de evitar cortocircuitos que puedan dañar a la plataforma u otro dispositivo, porque durante el desarrollo de la plataforma y la comprobación de módulos se quemaron algunos componentes que luego fueron sustituidos.
- Respetar los códigos de programación que fueron creados en lenguaje C y exclusivamente para el compilador CCS PIC, mas no para utilizarlos en MICRO C, porque utilizan lenguajes y comandos con mayor dificultad para programar.
- Para futuros proyectos con plataformas, tratar que sea construidos con los componentes más básicos posibles, para evitar que el tiempo de construcción se prolongue debido a la cantidad y complejidad de elementos que fueron elegidos en este proyecto.

BIBLIOGRAFÍA

- ANGULO José María, Susana Romero, Ignacio Angulo. (2010) MICROCONTROLADORES PIC / PIC16F87X – PIC18FXXX. (2da. Ed.) Madrid, España. McGraw-Hill.
- BARRA Franklin. (2011). MICROCONTROLADORES PIC CON PROGRAMACION Pic Basic Pro. (8va. Ed.). Barcelona, España. McGraw-Hill.
- BATES Martin. (2006). INTERFACING PIC MICROCONTROLLERS, EMBEDDED DESIGN BY INTERACTIVE SIMULATION. (1era. Ed.) Gran Bretaña. Newnes.
- GARCÍA Breijo Eduardo. (2008). COMPILADOR C CCS Y SIMULADOR PROTEUS PARA MICROCONTROLADORES PIC. (1ra. Ed.) México DF, México. Algaomega.
- PALACIOS Enrique. (2005) Microcontrolador PIC18F4550 Desarrollo de proyectos (1ea Ed.). Madrid-España. Editorial RA-MA
- MICROCHIP Technology Inc. (1999) MPASM ASSEMBLER USERS GUIDE (3ra Ed.). Estados Unidos.
- VALLEJO Horacio. (1998) Microcontroladores PIC – Aprenda una profesión (1ra Ed.). Madrid-España. Editorial Quearks S.R.L
- WILMSHURTS Tim (2007). DESIGNING EMBEDDED SYSTEMS WITH PIC MICROCONTROLLERS, PRINCIPLES AND APPLICATIONS. (1ra. Ed.) Gran Bretaña. Newnes.
- Andrés Curtidor, Camilo Herrera. (2012). DISEÑO Y CONSTRUCCION DE MODULOS ENTRENADORES PARA MICROCONTROLADORES. (1ra. Ed.) Madrid, España. Editorial Académica Española.

- F. Valdés, R. Pallas. (2007). MICROCONTROLADORES: FUNDAMENTOS Y APLICACIONES CON PIC. (1ra. Ed.) Barcelona, España. Marcombo S.A.
- G. Tojeiro Calaza. (2009). SIMULACION DE CIRCUITOS ELECTRÓNICOS Y MICROCONTROLADORES A TRAVÉS DE EJEMPLOS. (1ra. Ed.) Barcelona, España. Editorial Marcombo.
- Alda Mazarredo (2007). INGENIERÍA DE MICROSISTEMAS PROGRAMADOS. (Vol. 1) Bilbao, España.
- Introducción a los microcontroladores, José Adolfo González V., McGraw Hill 2008

BIBLIOGRAFÍA ELECTRÓNICA:

- www.dontronics.com
- www.ccsinfo.com
- www.dynamoelectronics.com
- www.arv.umh.es
- www.impronic.com

ANEXOS

MANUAL PARA CREAR PROYECTOS EN PIC C COMPILER

Se presentan a continuación los pasos para crear un proyecto en PIC C CCS.

OBJETIVOS.-

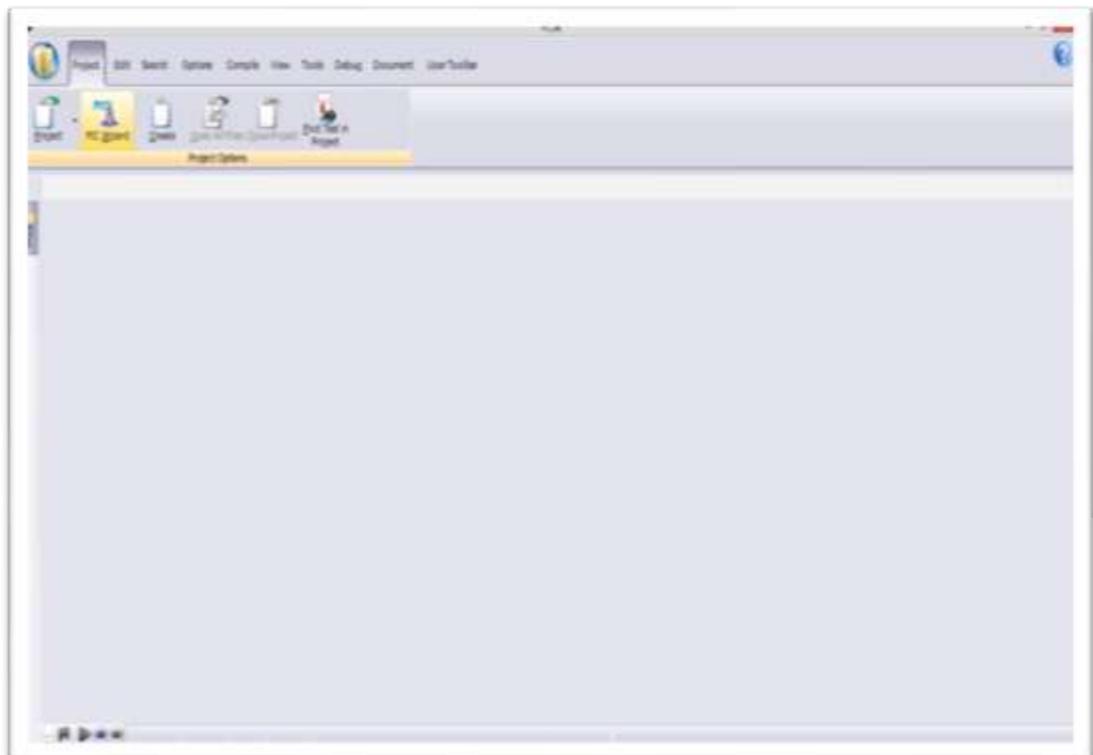
- Crear un proyecto para el desarrollo de código embebido orientado a microcontroladores, para el desarrollo de prácticas y analizar su comportamiento.

MODO DE USO.-

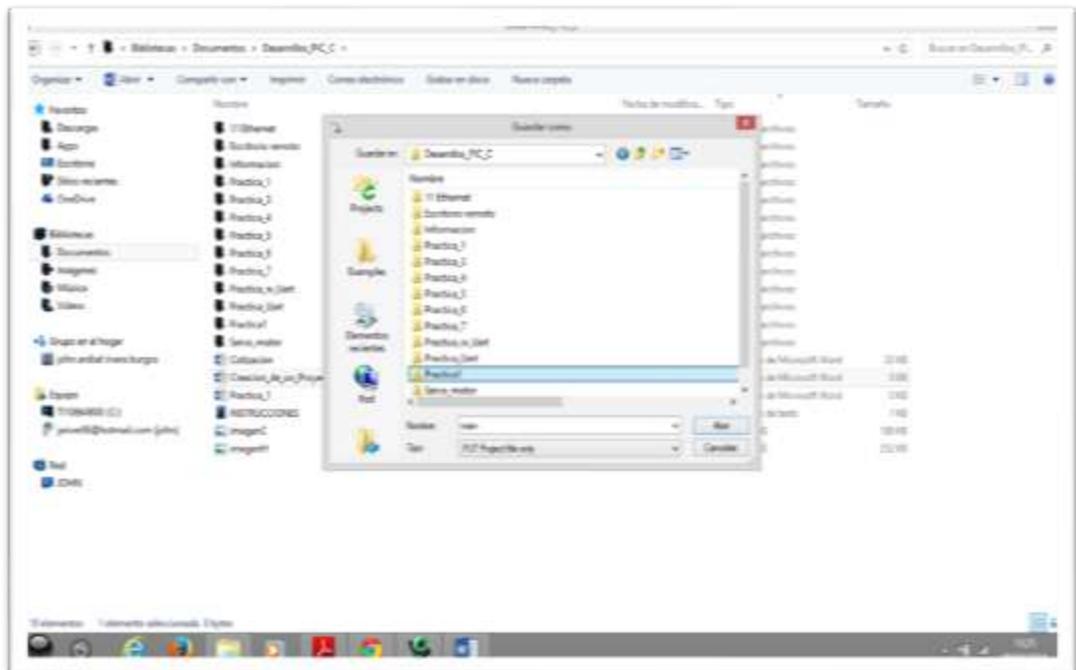
1. Ejecutar el ícono de PIC C CCS.



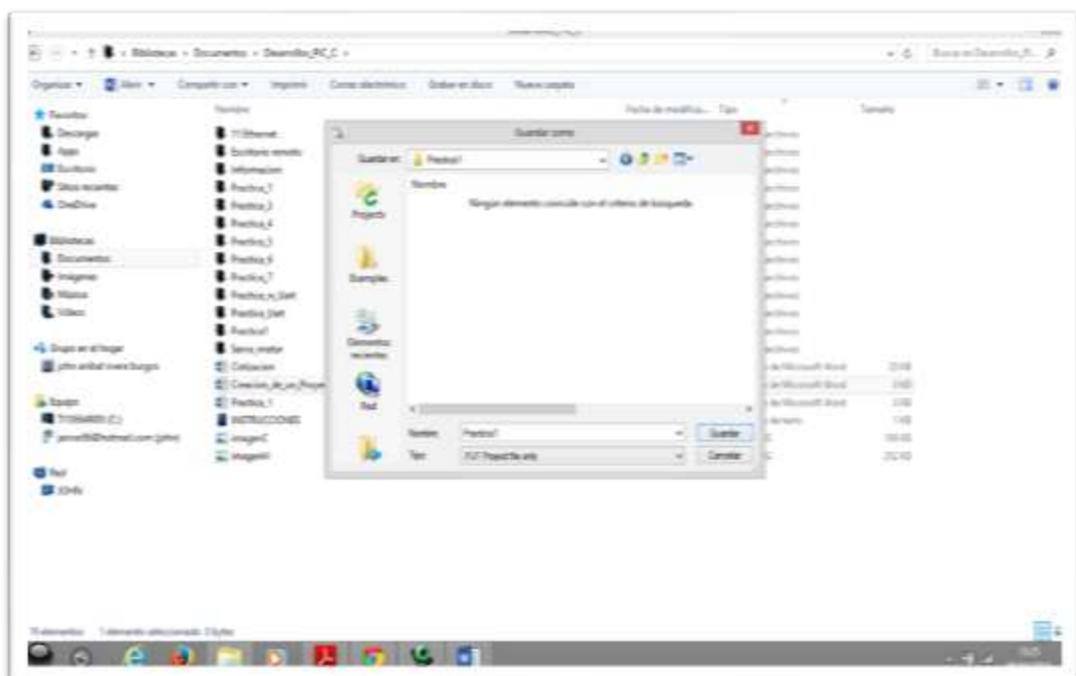
2. Seleccionar **Project**.
3. Seleccionar **Pic Wizard**.



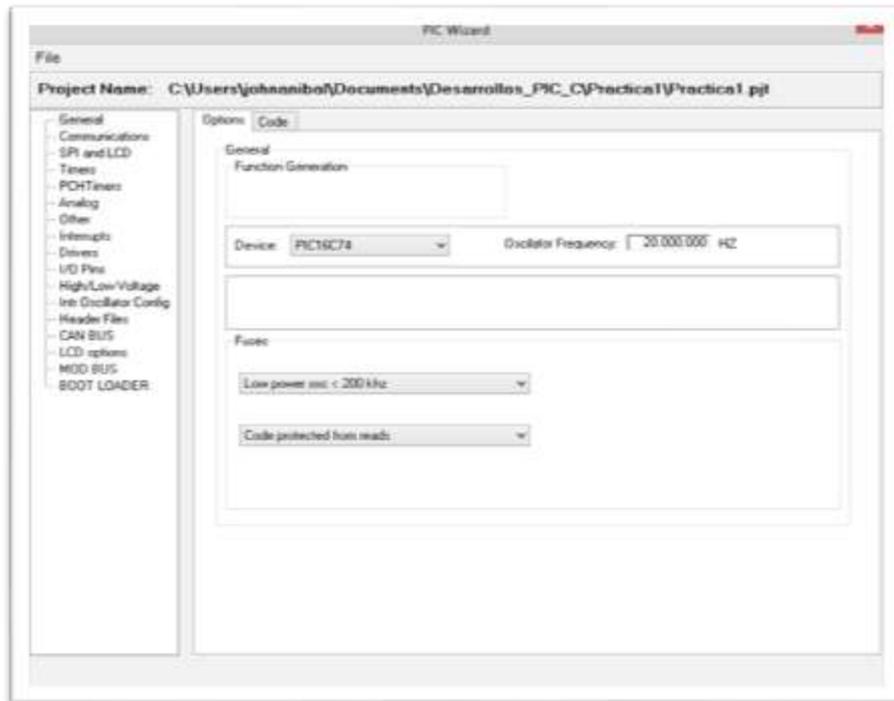
4. Crear una carpeta donde se van a guardar los códigos.



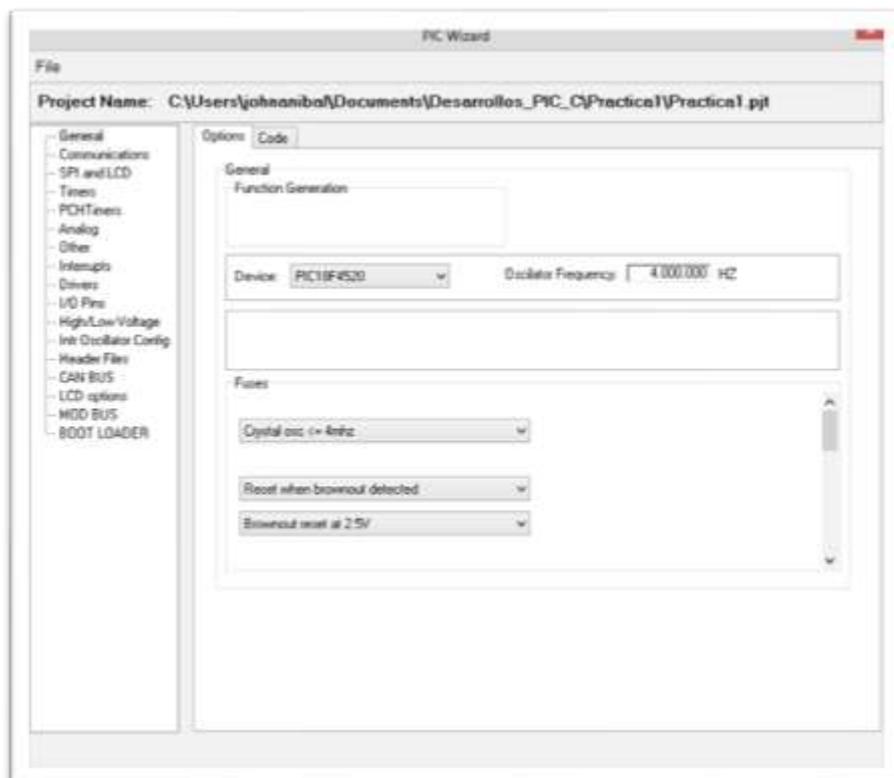
5. Escribir el nombre que se desea para el proyecto.



6. Ventana de configuración inicial.

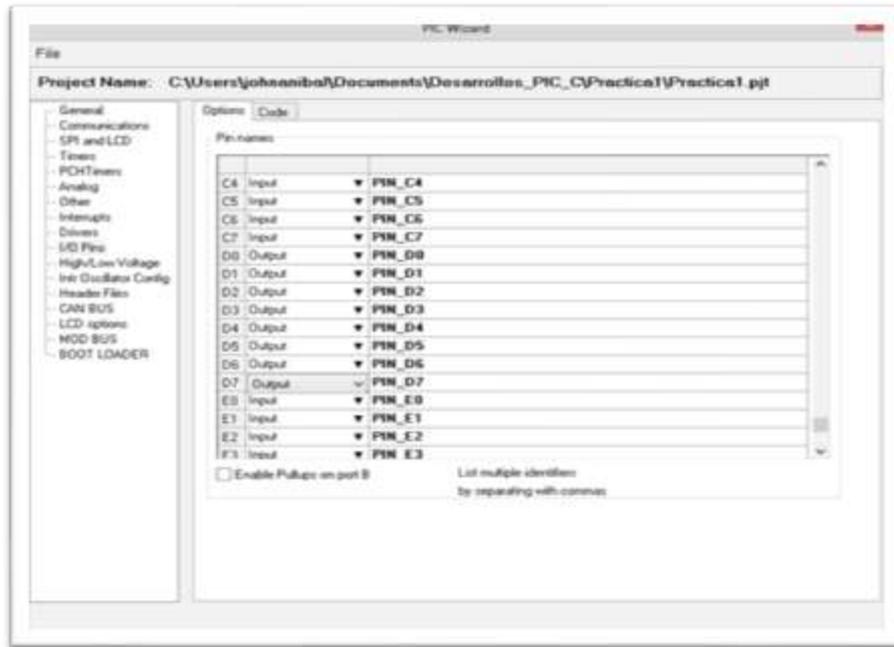


7. Configurar el PIC a utilizar y la frecuencia de trabajo.



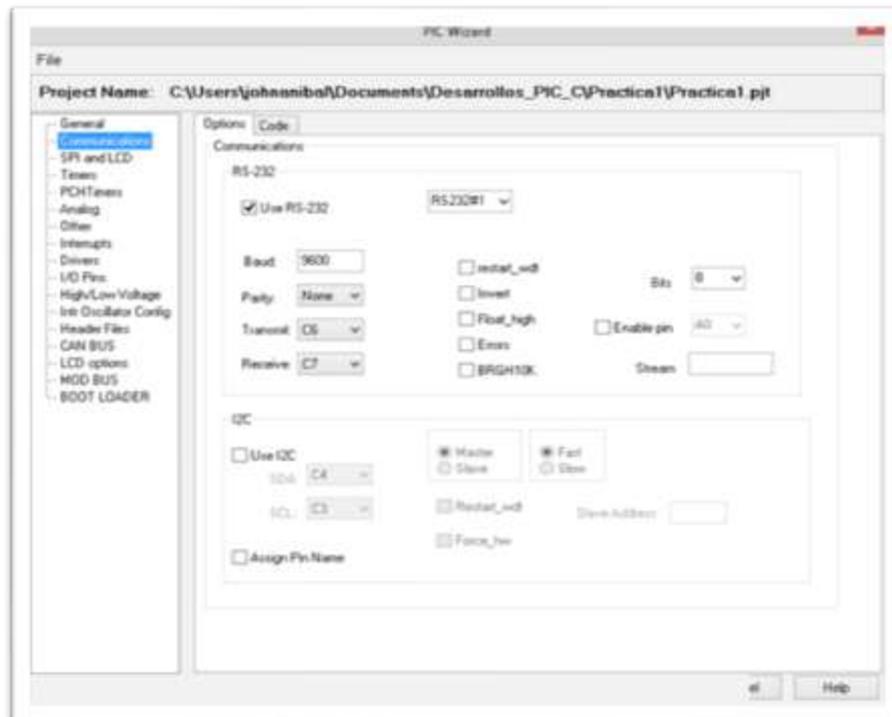
8. Seleccionar en el menú de la izquierda la opción **I/O Pins**.

9. Configurar como salida los pines desde **D0** hasta **D7** para el encendido de led's.



10. Seleccionar en el menú de la izquierda la opción **Communications**.

11. Configurar el PIC para comunicación **Serial UART**.

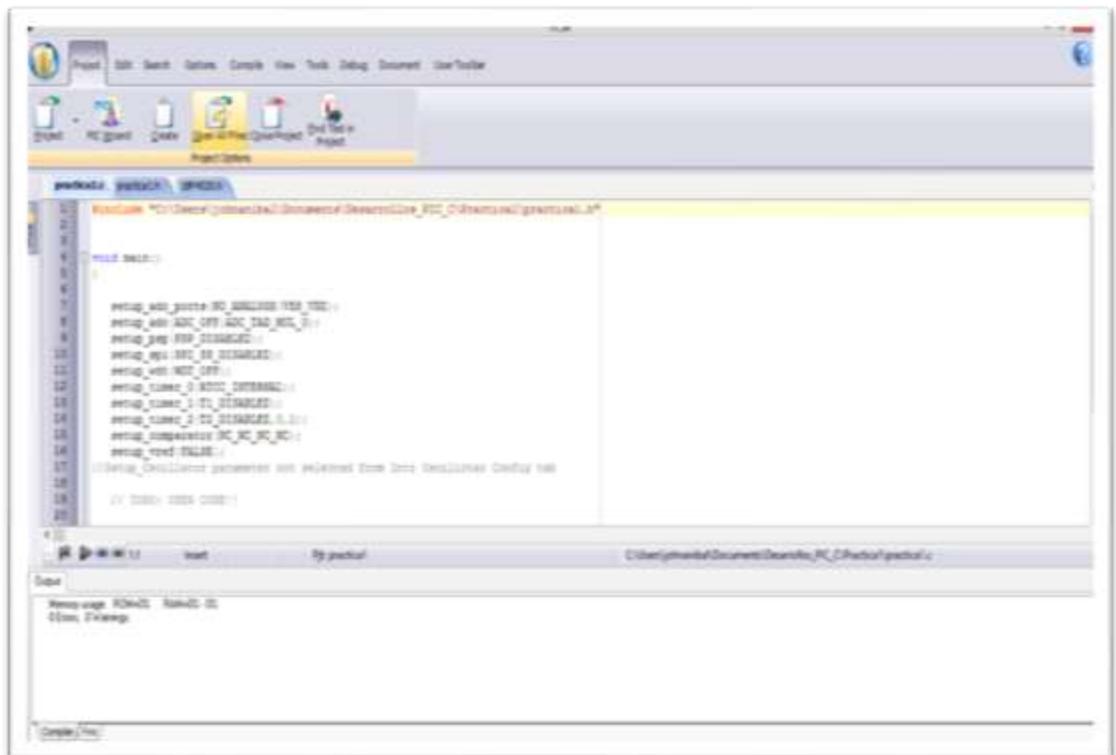


12. Seleccionar en el menú de la izquierda la opción **Analog**.

13. Configurar la conversión **Analógica/Digital A0**.



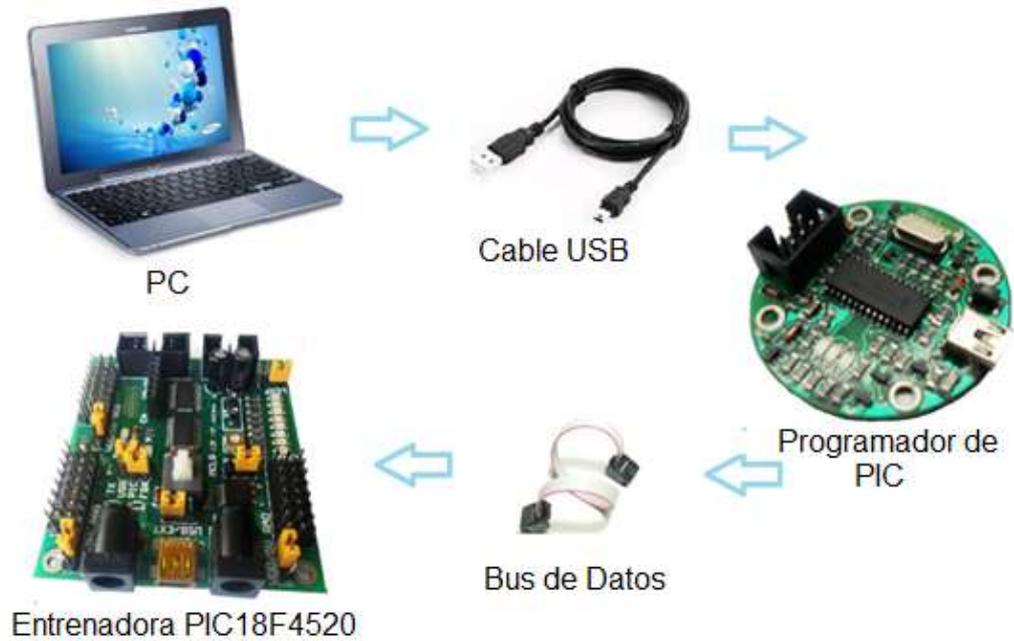
16. Ventana lista para el inicio de desarrollo de código.



MANUAL PARA EL USO DE SOFTWARE PICKIT2

Se presentan a continuación los pasos para poder usar el PICKIT2.

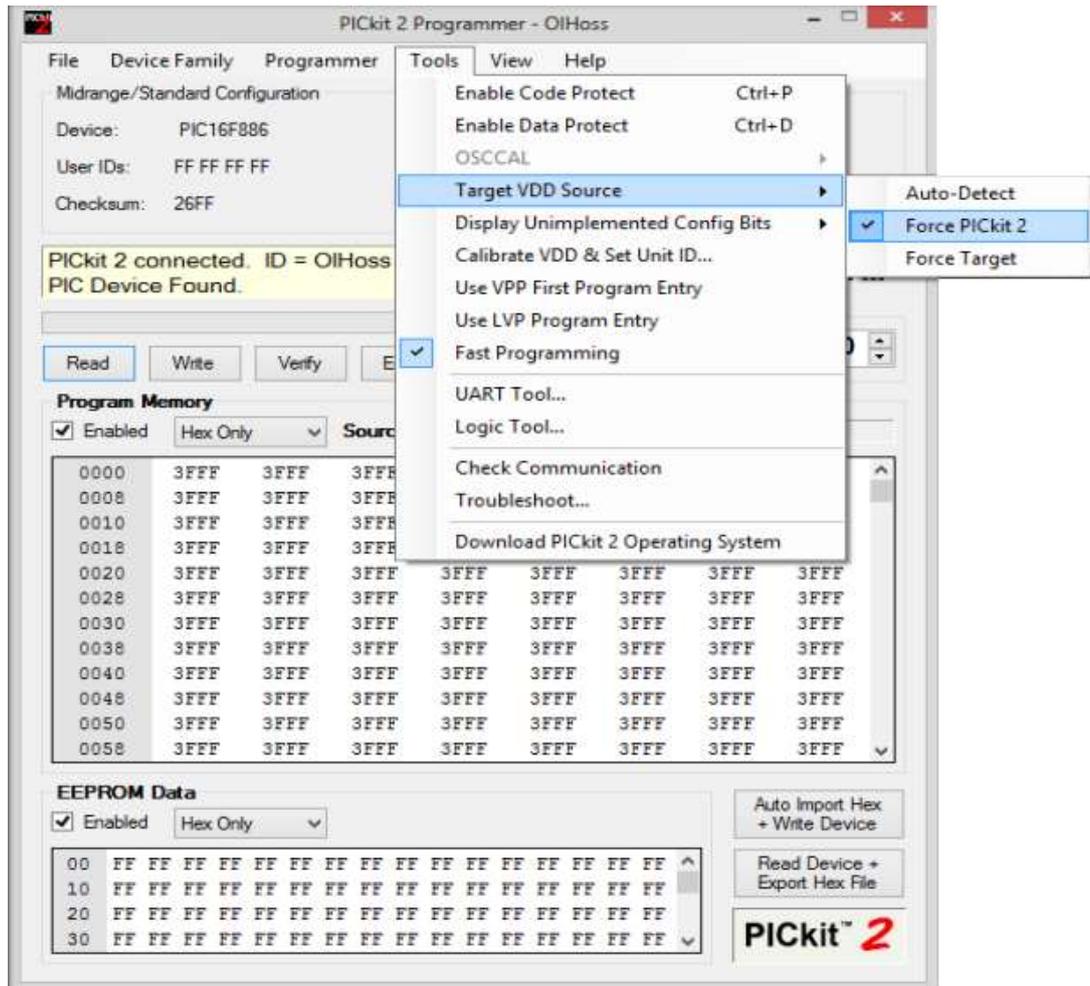
1. Conectar el hardware, como se detalla en el siguiente gráfico.



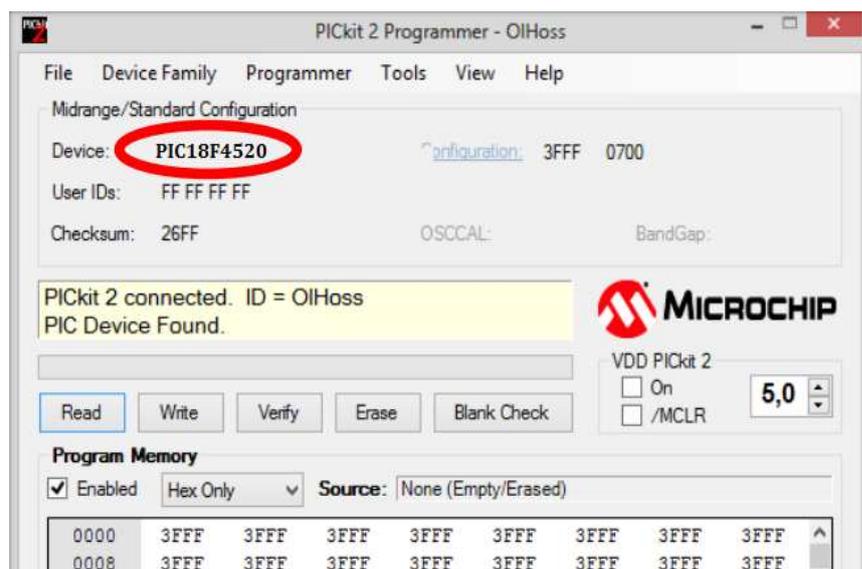
2. Ejecutar el ícono de Pickit2.



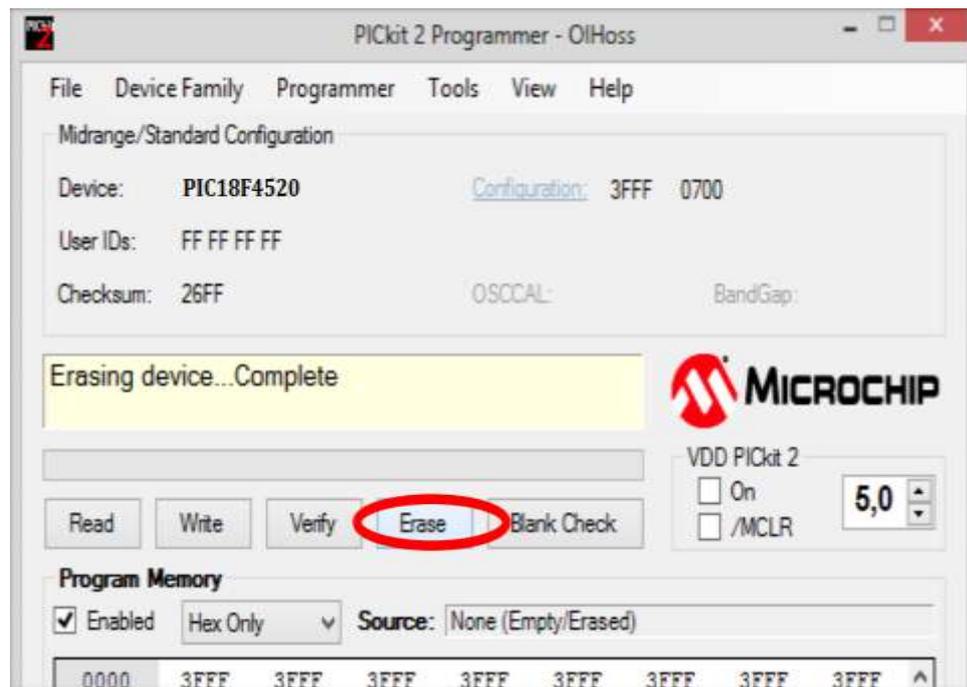
3. Seleccionar en el menú principal la opción **Tools**.
4. Seleccionar **Target VDD Source**.
5. Seleccionar **Force PICKit2**.



6. Observar que el dispositivo haya sido detectado, en este caso es el PIC18F4520.

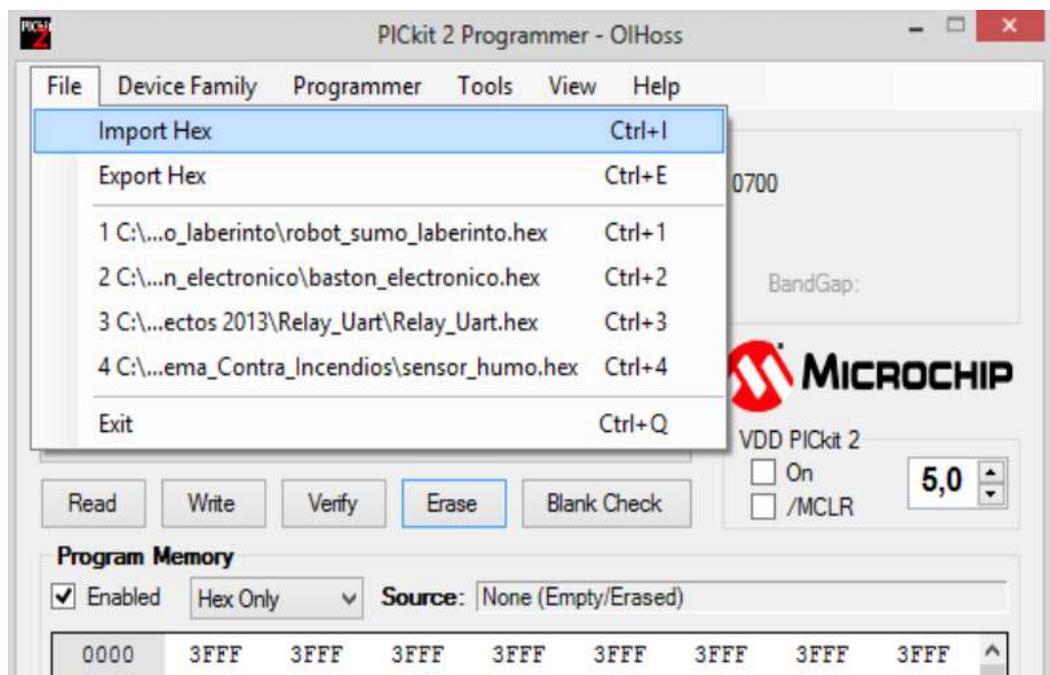


7. Seleccionar la opción **Erase**.

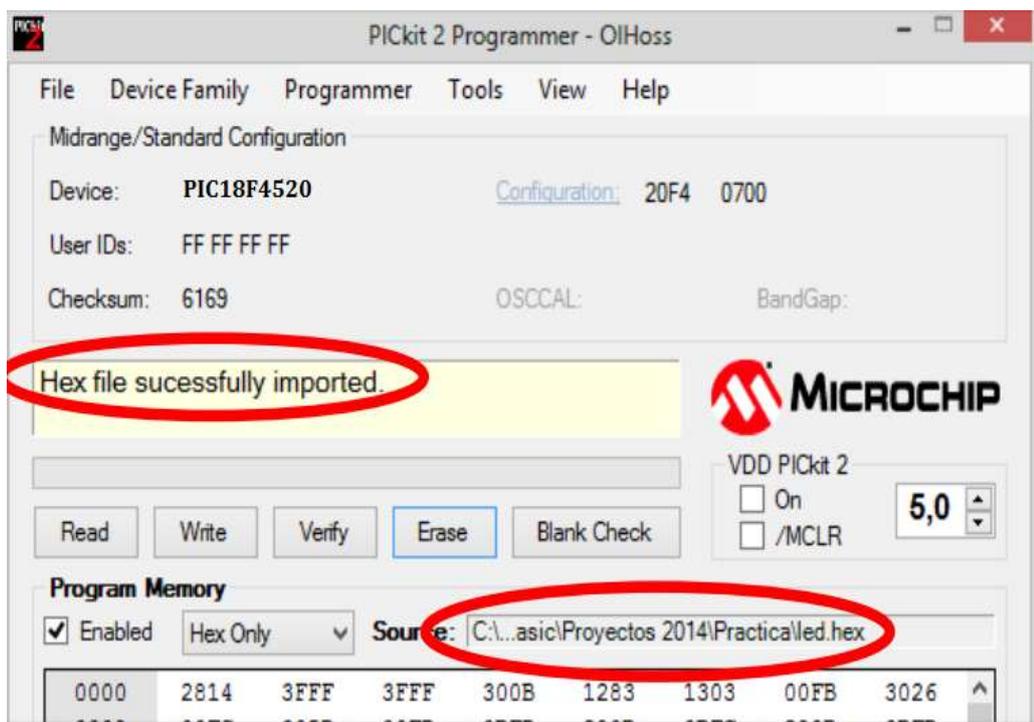
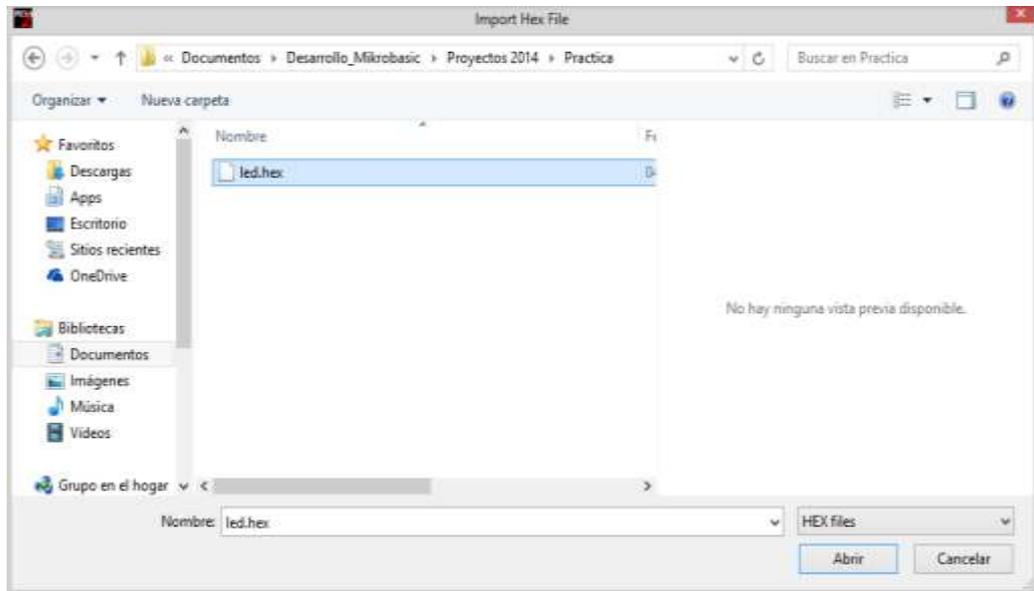


8. Seleccionar en el menú la opción **File**.

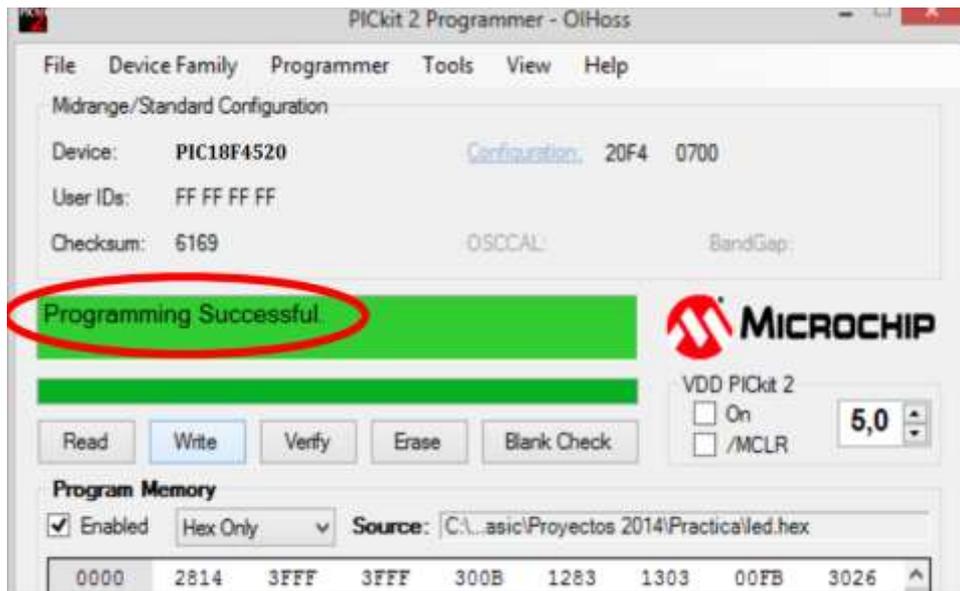
9. Seleccionar la opción **Import Hex**.



10. Buscar el archivo .HEX en la carpeta donde se creó el proyecto cuando se utilizó el programa PIC C CCS, para que se pueda realizar la importación del archivo.



11. Seleccionar la opción **Write** para que el código de programa sea escrito en el microcontrolador.



MANUAL PARA USO DE SOFTWARE ACCESSPORT

AccessPort es software que permite comunicación con dispositivos por medio del protocolo serial UART.

OBJETIVOS.-

- Manipular la ventana de AccessPort para la comunicación Serial UART
- Identificar la ventana de envío de datos.
- Identificar la ventana de recepción de datos.
- Identificar el menú de configuración de Puerto COM
- Identificar el menú de configuración BAUDRATE.

DIAGRAMA ESQUEMÁTICO.-



MODO DE USO.-

1. Conectar la tarjeta de control al Computador por el puerto USB.



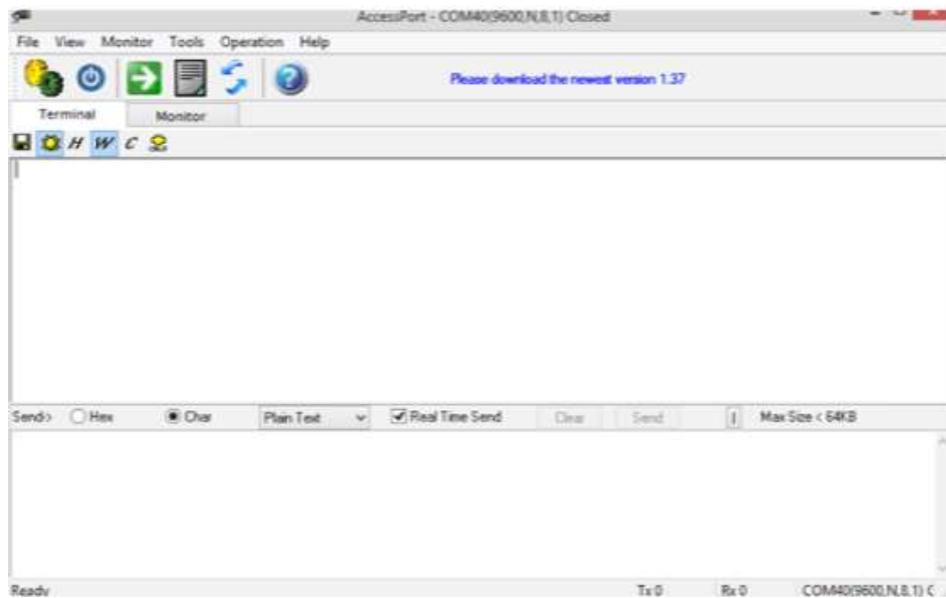
2. Ingresar al Administrador de Dispositivos y verificar el Puerto COM asignado.



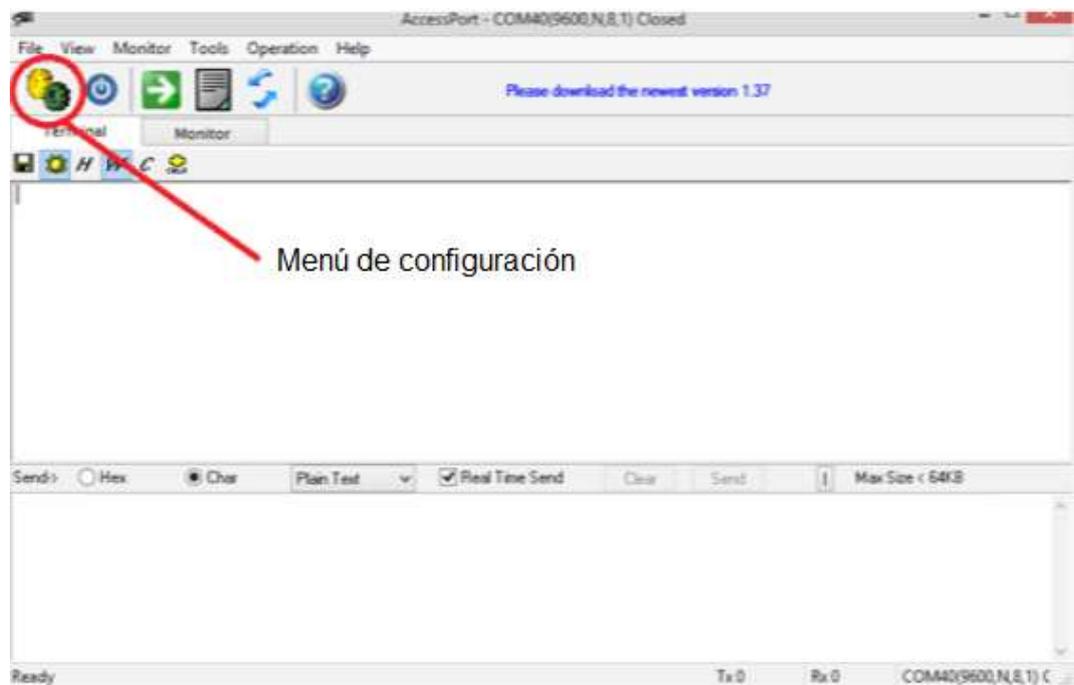
3. Ejecutar el ícono de AccessPort.



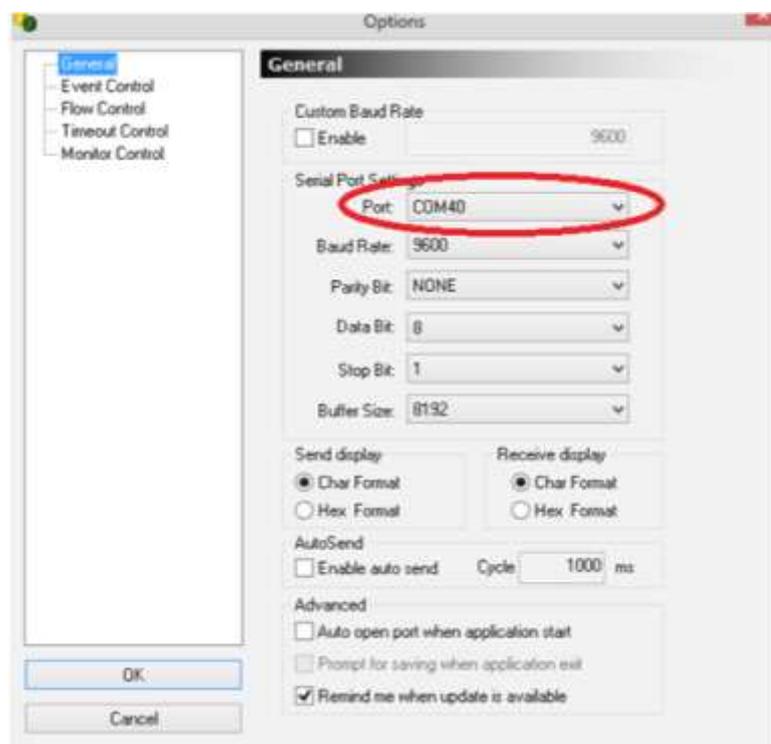
4. Ventana principal de configuración.



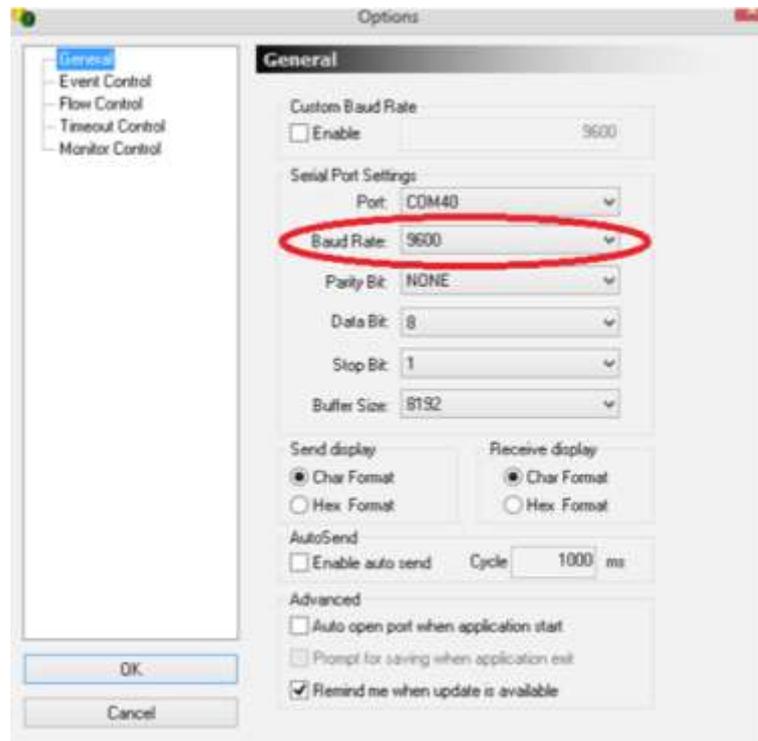
5. Ingresar al menú de configuración.



6. Configurar el Puerto COM asignado en el Administrador de Dispositivos



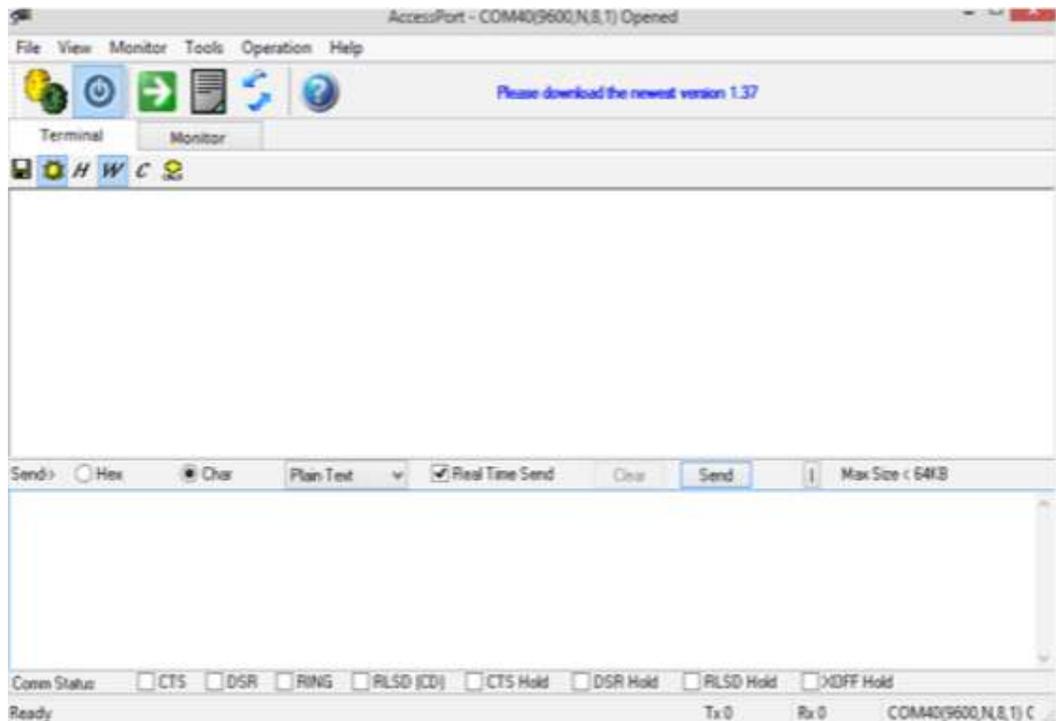
7. Configurar la tasa de transmisión de BITS según lo programado "9600".



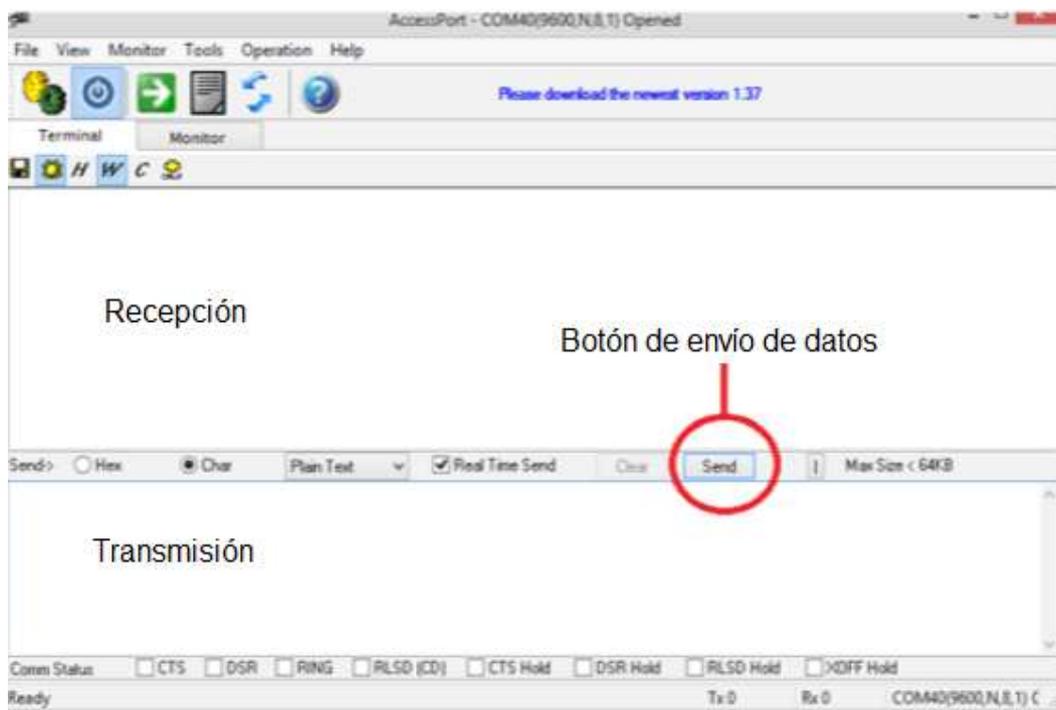
8. Seleccionar la opción **OK**.



9. Cuando el enlace se realiza con éxito, es indicado por **BOTON Azul**.



NOTA: No olvidar las Partes importantes de la ventana principal.



PRÁCTICA # 1

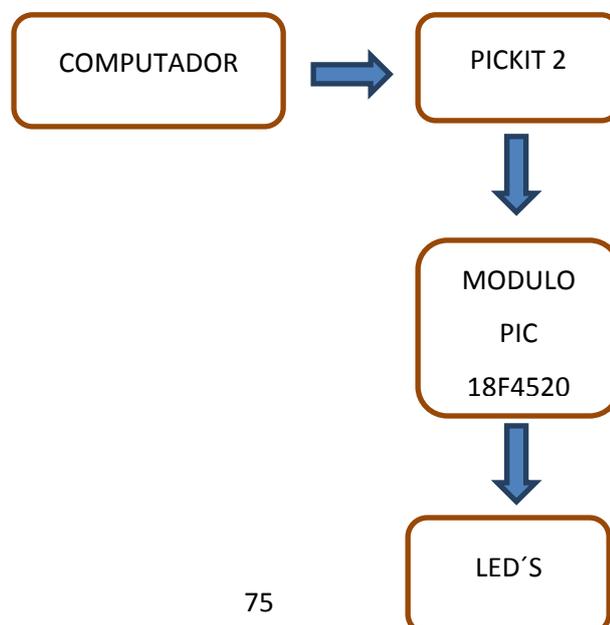
ENCENDIDO DE LED'S

Esta práctica se dedica a dar una introducción para el encendido de Led's de la Tarjeta de Entrenamiento con PIC 18F4520. Una vez desarrollado el código de ejemplo, el alumno debe realizar cambios en el software de control, para verificar lo aprendido.

OBJETIVOS.-

- Crear un proyecto con el software de control PIC C COMPILER.
- Crear un código para el control del Módulo de Entrenamiento con PIC 18F4520.
- Programar la tarjeta de control con el software PICKIT 2.
- Programar la tarjeta con el Software de control inicial de la Practica 1.
- Realizar pruebas de laboratorio, mediante cambios en el algoritmo de control para consolidar los conocimientos adquiridos.
- Programar el Módulo con PIC 18f4520 para el encendido de Led's

DIAGRAMA ESQUEMÁTICO.-



CÓDIGO DE PROGRAMACIÓN.-

```
#include
#use delay (clock=4000000)
#use rs232 (baud=9600, parity=N, xmit=PIN_C6, rcv=PIN_C7, bits=8)
void main ()
{
    setup_adc_ports (NO_ANALOGS|VSS_VDD);
    setup_adc (ADC_OFF|ADC_TAD_MUL_0);
    setup_psp (PSP_DISABLED);
    setup_spi (SPI_SS_DISABLED);
    setup_wdt (WDT_OFF);
    setup_timer_0 (RTCC_INTERNAL);
    setup_timer_1 (T1_DISABLED);
    setup_timer_2 (T2_DISABLED,0,1);
    //setup_timer_3 (T3_DISABLED|T3_DIV_BY_1);
    setup_comparator (NC_NC_NC_NC);
    setup_vref (FALSE);
    //Setup_Oscillator parameter not selected from Intr Oscillotar Config
    tab

    // TODO: USER CODE!!

    SET_TRIS_D (0B00000000);
    SET_TRIS_E (0B00001000);

    while (1)
    {

        OUTPUT_D (0B11111110);
        Delay_ms (100);
        OUTPUT_D (0B11111101);
        Delay_ms (100);
        OUTPUT_D (0B11111011);
```

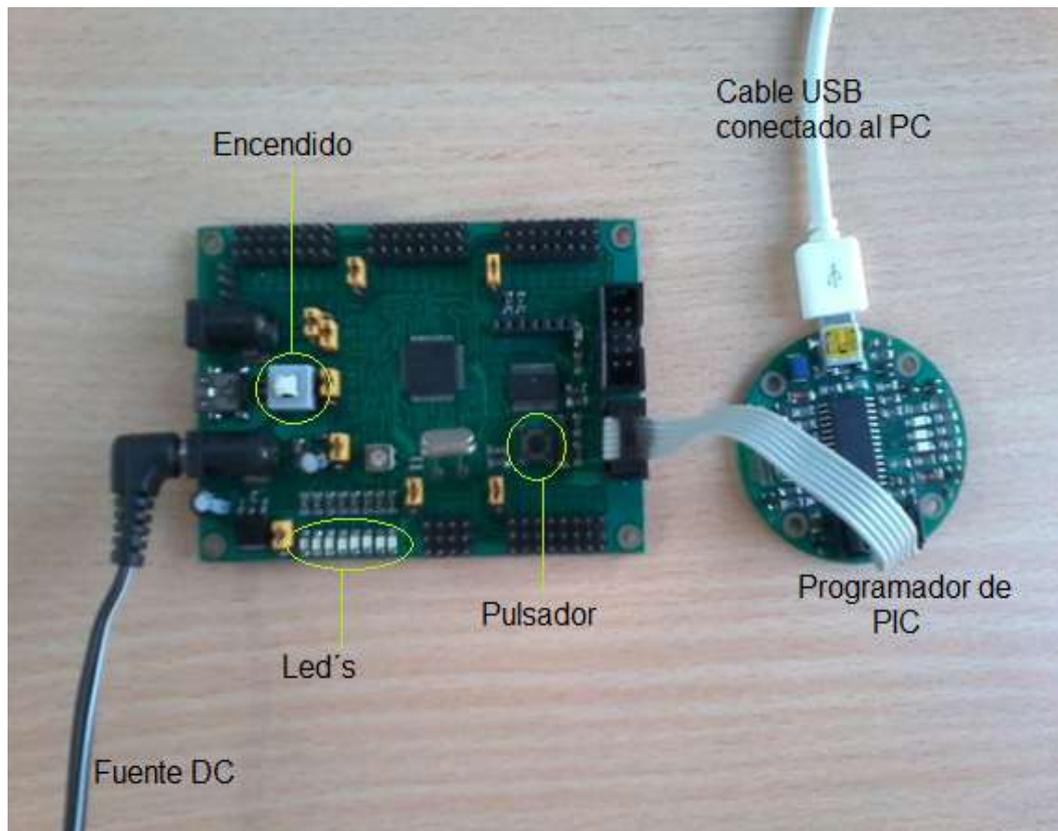
```
    Delay_ms (100);
    OUTPUT_D (0B11110111);
    Delay_ms (100);
    OUTPUT_D (0B11101111);
    Delay_ms (100);
    OUTPUT_D (0B11011111);
    Delay_ms (100);
    OUTPUT_D (0B10111111);
    Delay_ms (100);
    OUTPUT_D (0B01111111);
    Delay_ms (100);
    OUTPUT_D (0B11111111);
    Delay_ms (100);
    OUTPUT_D (0B01111111);
    Delay_ms (100);
    OUTPUT_D (0B10111111);
    Delay_ms (100);
    OUTPUT_D (0B11011111);
    Delay_ms (100);
    OUTPUT_D (0B11101111);
    Delay_ms (100);
    OUTPUT_D (0B11110111);
    Delay_ms (100);
    OUTPUT_D (0B11110111);
    Delay_ms (100);
    OUTPUT_D (0B11111011);
    Delay_ms (100);
    OUTPUT_D (0B11111101);
    Delay_ms (100);
    OUTPUT_D (0B11111110);
    Delay_ms (100);
    OUTPUT_D (0B11111111);
    Delay_ms (100);
}
}
```

SOFTWARE DE CONTROL ARCHIVO .C.-

```

1 #include "C:\Users\johanna\Documents\Desarrollo_PIC_C\Practica_1\escrondo_leds.h"
2 #asm delay @lock=4000000
3 #asm rs232 @baud=9600 parity=N,smis=PIN_06,rcv=PIN_07,bits=8
4
5
6
7 void main()
8
9
10 setup_adc_ports(NO_ANALOGS,VSS_VDD);
11 setup_adc(ADC_OFF/ADC_TAD_MUX_0);
12 setup_psp(PSW_DISABLED);
13 setup_spl(SPL_30_DISABLED);
14 setup_wdt(WDT_OFF);
15 setup_timer_0(T0CC_INTERNAL);
16 setup_timer_1(T1_DISABLED);
17 setup_timer_2(T2_DISABLED,0,1);
18 //setup_timer_3(T3_DISABLED,0,1);
19 setup_comparator(NC_NC_NC_NC);
20 setup_vref(FALSE);
21 //Setup_Oscillator parameter not selected from Intel Oscillator Config tab
22
23 // TODO: USER CODE!
24
25 SET_TRIS_D(0b00000000);
26 SET_TRIS_E(0b00001000);
27
28 while(1)
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
840
841
842
843
844
845
846
847
848
849
850
851
852
853
854
855
856
857
858
859
860
861
862
863
864
865
866
867
868
869
870
871
872
873
874
875
876
877
878
879
880
881
882
883
884
885
886
887
888
889
890
891
892
893
894
895
896
897
898
899
900
901
902
903
904
905
906
907
908
909
910
911
912
913
914
915
916
917
918
919
920
921
922
923
924
925
926
927
928
929
930
931
932
933
934
935
936
937
938
939
940
941
942
943
944
945
946
947
948
949
950
951
952
953
954
955
956
957
958
959
960
961
962
963
964
965
966
967
968
969
970
971
972
973
974
975
976
977
978
979
980
981
982
983
984
985
986
987
988
989
990
991
992
993
994
995
996
997
998
999
1000
1001
1002
1003
1004
1005
1006
1007
1008
1009
1010
1011
1012
1013
1014
1015
1016
1017
1018
1019
1020
1021
1022
1023
1024
1025
1026
1027
1028
1029
1030
1031
1032
1033
1034
1035
1036
1037
1038
1039
1040
1041
1042
1043
1044
1045
1046
1047
1048
1049
1050
1051
1052
1053
1054
1055
1056
1057
1058
1059
1060
1061
1062
1063
1064
1065
1066
1067
1068
1069
1070
1071
1072
1073
1074
1075
1076
1077
1078
1079
1080
1081
1082
1083
1084
1085
1086
1087
1088
1089
1090
1091
1092
1093
1094
1095
1096
1097
1098
1099
1100
1101
1102
1103
1104
1105
1106
1107
1108
1109
1110
1111
1112
1113
1114
1115
1116
1117
1118
1119
1120
1121
1122
1123
1124
1125
1126
1127
1128
1129
1130
1131
1132
1133
1134
1135
1136
1137
1138
1139
1140
1141
1142
1143
1144
1145
1146
1147
1148
1149
1150
1151
1152
1153
1154
1155
1156
1157
1158
1159
1160
1161
1162
1163
1164
1165
1166
1167
1168
1169
1170
1171
1172
1173
1174
1175
1176
1177
1178
1179
1180
1181
1182
1183
1184
1185
1186
1187
1188
1189
1190
1191
1192
1193
1194
1195
1196
1197
1198
1199
1200
1201
1202
1203
1204
1205
1206
1207
1208
1209
1210
1211
1212
1213
1214
1215
1216
1217
1218
1219
1220
1221
1222
1223
1224
1225
1226
1227
1228
1229
1230
1231
1232
1233
1234
1235
1236
1237
1238
1239
1240
1241
1242
1243
1244
1245
1246
1247
1248
1249
1250
1251
1252
1253
1254
1255
1256
1257
1258
1259
1260
1261
1262
1263
1264
1265
1266
1267
1268
1269
1270
1271
1272
1273
1274
1275
1276
1277
1278
1279
1280
1281
1282
1283
1284
1285
1286
1287
1288
1289
1290
1291
1292
1293
1294
1295
1296
1297
1298
1299
1300
1301
1302
1303
1304
1305
1306
1307
1308
1309
1310
1311
1312
1313
1314
1315
1316
1317
1318
1319
1320
1321
1322
1323
1324
1325
1326
1327
1328
1329
1330
1331
1332
1333
1334
1335
1336
1337
1338
1339
1340
1341
1342
1343
1344
1345
1346
1347
1348
1349
1350
1351
1352
1353
1354
1355
1356
1357
1358
1359
1360
1361
1362
1363
1364
1365
1366
1367
1368
1369
1370
1371
1372
1373
1374
1375
1376
1377
1378
1379
1380
1381
1382
1383
1384
1385
1386
1387
1388
1389
1390
1391
1392
1393
1394
1395
1396
1397
1398
1399
1400
1401
1402
1403
1404
1405
1406
1407
1408
1409
1410
1411
1412
1413
1414
1415
1416
1417
1418
1419
1420
1421
1422
1423
1424
1425
1426
1427
1428
1429
1430
1431
1432
1433
1434
1435
1436
1437
1438
1439
1440
1441
1442
1443
1444
1445
1446
1447
1448
1449
1450
1451
1452
1453
1454
1455
1456
1457
1458
1459
1460
1461
1462
1463
1464
1465
1466
1467
1468
1469
1470
1471
1472
1473
1474
1475
1476
1477
1478
1479
1480
1481
1482
1483
1484
1485
1486
1487
1488
1489
1490
1491
1492
1493
1494
1495
1496
1497
1498
1499
1500
1501
1502
1503
1504
1505
1506
1507
1508
1509
1510
1511
1512
1513
1514
1515
1516
1517
1518
1519
1520
1521
1522
1523
1524
1525
1526
1527
1528
1529
1530
1531
1532
1533
1534
1535
1536
1537
1538
1539
1540
1541
1542
1543
1544
1545
1546
1547
1548
1549
1550
1551
1552
1553
1554
1555
1556
1557
1558
1559
1560
1561
1562
1563
1564
1565
1566
1567
1568
1569
1570
1571
1572
1573
1574
1575
1576
1577
1578
1579
1580
1581
1582
1583
1584
1585
1586
1587
1588
1589
1590
1591
1592
1593
1594
1595
1596
1597
1598
1599
1600
1601
1602
1603
1604
1605
1606
1607
1608
1609
1610
1611
1612
1613
1614
1615
1616
1617
1618
1619
1620
1621
1622
1623
1624
1625
1626
1627
1628
1629
1630
1631
1632
1633
1634
1635
1636
1637
1638
1639
1640
1641
1642
1643
1644
1645
1646
1647
1648
1649
1650
1651
1652
1653
1654
1655
1656
1657
1658
1659
1660
1661
1662
1663
1664
1665
1666
1667
1668
1669
1670
1671
1672
1673
1674
1675
1676
1677
1678
1679
1680
1681
1682
1683
1684
1685
1686
1687
1688
1689
1690
1691
1692
1693
1694
1695
1696
1697
1698
1699
1700
1701
1702
1703
1704
1705
1706
1707
1708
1709
1710
1711
1712
1713
1714
1715
1716
1717
1718
1719
1720
1721
1722
1723
1724
1725
1726
1727
1728
1729
1730
1731
1732
1733
1734
1735
1736
1737
1738
1739
1740
1741
1742
1743
1744
1745
1746
1747
1748
1749
1750
1751
1752
1753
1754
1755
1756
1757
1758
1759
1760
1761
1762
1763
1764
1765
1766
1767
1768
1769
1770
1771
1772
1773
1774
1775
1776
1777
1778
1779
1780
1781
1782
1783
1784
1785
1786
1787
1788
1789
1790
1791
1792
1793
1794
1795
1796
1797
1798
1799
1800
1801
1802
1803
1804
1805
1806
1807
1808
1809
1810
1811
1812
1813
1814
1815
1816
1817
1818
1819
1820
1821
1822
1823
1824
1825
1826
1827
1828
1829
1830
1831
1832
1833
1834
1835
1836
1837
1838
1839
1840
1841
1842
1843
1844
1845
1846
1847
1848
1849
1850
1851
1852
1853
1854
1855
1856
1857
1858
1859
1860
1861
1862
1863
1864
1865
1866
1867
1868
1869
1870
1871
1872
1873
1874
1875
1876
1877
1878
1879
1880
1881
1882
1883
1884
1885
1886
1887
1888
1889
1890
1891
1892
1893
1894
1895
1896
1897
1898
1899
1900
1901
1902
1903
1904
1905
1906
1907
1908
1909
1910
1911
1912
1913
1914
1915
1916
1917
1918
1919
1920
1921
1922
1923
1924
1925
1926
1927
1928
1929
1930
1931
1932
1933
1934
1935
1936
1937
1938
1939
1940
1941
1942
1943
1944
1945
1946
1947
1948
1949
1950
1951
1952
1953
1954
1955
1956
1957
1958
1959
1960
1961
1962
1963
1964
1965
1966
1967
1968
1969
1970
1971
1972
1973
1974
1975
1976
1977
1978
1979
1980
1981
1982
1983
1984
1985
1986
1987
1988
1989
1990
1991
1992
1993
1994
1995
1996
1997
1998
1999
2000
2001
2002
2003
2004
2005
2006
2007
2008
2009
2010
2011
2012
2013
2014
2015
2016
2017
2018
2019
2020
2021
2022
2023
2024
2025
2026
2027
2028
2029
2030
2031
2032
2033
2034
2035
2036
2037
2038
2039
2040
2041
2042
2043
2044
2045
2046
2047
2048
2049
2050
2051
2052
2053
2054
2055
2056
2057
2058
2059
2060
2061
2062
2063
2064
2065
2066
2067
2068
2069
2070
2071
2072
2073
2074
2075
2076
2077
2078
2079
2080
2081
2082
2083
2084
2085
2086
2087
2088
2089
2090
2091
2092
2093
2094
2095
2096
2097
2098
2099
2100
2101
2102
2103
2104
2105
2106
2107
2108
2109
2110
2111
2112
2113
2114
2115
2116
2117
2118
2119
2120
2121
2122
2123
2124
2125
2126
2127
2128
2129
2130
2131
2132
2133
2134
2135
2136
2137
2138
2139
2140
2141
2142
2143
2144
2145
2146
2147
2148
2149
2150
2151
2152
2153
2154
2155
2156
2157
2158
2159
2160
2161
2162
2163
2164
2165
2166
2167
2168
2169
2170
2171
2172
2173
2174
2175
2176
2177
2178
2179
2180
2181
2182
2183
2184
2185
2186
2187
2188
2189
2190
2191
2192
2193
2194
2195
2196
2197
2198
2199
2200
2201
2202
2203
2204
2205
2206
2207
2208
2209
2210
2211
2212
2213
2214
2215
2216
2217
2218
2219
2220
2221
2222
2223
2224
2225
2226
2227
2228
2229
2230
2231
2232
2233
2234
2235
2236
2237
2238
2239
2240
2241
2242
2243
2244
2245
2246
2247
2248
2249
2250
2251
2252
2253
2254
2255
2256
2257
2258
2259
2260
2261
2262
2263
2264
2265
2266
2267
2268
2269
2270
2271
2272
2273
2274
2275
2276
2277
2278
2279
2280
2281
2282
2283
2284
2285
2286
2287
2288
2289
2290
2291
2292
2293
2294
2295
2296
2297
2298
2299
2300
2301
2302
2303
2304
2305
2306
2307
2308
2309
2310
2311
2312
2313
2314
2315
2316
2317
2318
2319
2320
2321
2322
2323
2324
2325
2326
2327
2328
2329
2330
2331
2332
2333
2334
2335
2336
2337
2338
2339
2340
2341
2342
2343
2344
2345
2346
2347
2348
2349
2350
2351
2352
2353
2354
2355
2356
2357
2358
2359
2360
2361
2362
2363
2364
2365
2366
2367
2368
2369
2370
2371
2372
2373
2374
2375
2376
2377
2378
2379
2380
2381
2382
2383
2384
2385
2386
2387
2388
2389
2390
2391
2392
2393
2394
2395
2396
2397
2398
2399
2400
2401
2402
2403
2404
2405
2406
2407
2408
2409
2410
2411
2412
2413
2414
2415
2416
2417
2418
2419
2420
2421
2422
2423
2424
2425
2426
2427
2428
2429
2430
2431
2432
2433
2434
2435
2436
2437
2438
2439
2440
2441
2442
2443
2444
2445
2446
2447
2448
2449
2450
2451
2452
2453
2454
2455
2456
2457
2458
2459
2460
2461
2462
2463
2464
2465
2466
2467
2468
2469
2470
2471
2472
2473
2474
2475
2476
2477
2478
2479
2480
2481
2482
2483
2484
2485
2486
2487
2488
2489
2490
2491
2492
2493
2494
2495
2496
2497
2498
2499
2500
2501
2502
2503
2504
2505
2506
2507
2508
2509
2510
2511
2512
2513
2514
2515
2516
2517
2518
2519
2520
2521
2522
2523
2524
2525
2526
2527
2528
2529
2530
2531
2532
2533
2534
2535
2536
2537
2538
2539
2540
2541
2542
2543
2544
2545
2546
2547
2548
2549
2550
2551
2552
2553
2554
2555
2556
2557
2558
2559
2560
2561
2562
2563
2564
2565
2566
2567
2568
2569
2570
2571
2572
2573
2574
2575
2576
2577
2578
2579
2580
2581
2582
2583
2584
2585
2586
2587
2588
2589
2590
2591
2592
2593
2594
2595
2596
2597
2598
2599
2600
2601
2602
2603
2604
2605
2606
2607
2608
2609
2610
2611
2612
2613
2614
2615
2616
2617
2618
2619
2620
2621
2622
2623
2624
26
```

IMPLEMENTACIÓN.-



CONEXIONES DE PUERTOS.-

PINES DE CONEXIÓN	LUCES LED
PUERTO A	NC
PUERTO B	NC
PUERTO C	NC
PUERTO D	ENCENDICO
PUERTO E	NC
PIN_E3	PULSADOR

NOTA.- Las siglas NC significan No Conectado.

PRÁCTICA # 2

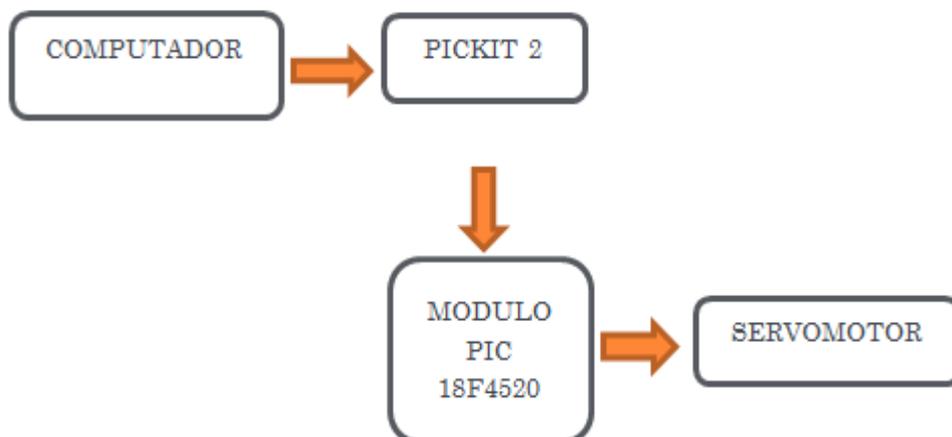
SERVOMOTOR

Esta práctica se dedica a dar una introducción para controlar el giro de un Servomotor utilizado en robótica, con la Tarjeta de Entrenamiento con PIC 18F4520. Una vez desarrollado el código de ejemplo, el alumno debe realizar cambios en el software de control, para verificar lo aprendido.

OBJETIVOS.-

- Crear un proyecto con el software de control PIC C COMPILER.
- Crear un código para el control del Módulo de Entrenamiento con PIC 18F4520.
- Programar la tarjeta de control con el software PICKIT 2.
- Programar la tarjeta con el Software de control inicial de la Practica 7.
- Realizar pruebas de laboratorio, mediante cambios en el algoritmo de control para consolidar los conocimientos adquiridos.
- Programar el Módulo con PIC 18f4520 para hacer girar el servomotor en 3 posiciones diferentes.

DIAGRAMA ESQUEMÁTICO.-



CÓDIGO DE PROGRAMACIÓN.-

```
#include
#use delay (clock=4000000, RESTART_WDT)
#use rs232 (baud=9600, parity=N, xmit=PIN_C6,rcv=PIN_C7,bits=8)

void main()
{
    setup_adc_ports (NO_ANALOGS|VSS_VDD);
    setup_adc (ADC_OFF|ADC_TAD_MUL_0);
    setup_psp (PSP_DISABLED);
    setup_spi (SPI_SS_DISABLED);
    setup_wdt (WDT_OFF);
    setup_timer_0 (RTCC_INTERNAL);
    setup_timer_1 (T1_DISABLED);
    setup_timer_2 (T2_DISABLED,0,1);
    setup_comparator (NC_NC_NC_NC);
    setup_vref (FALSE);
        //Setup_Oscillator parameter not selected from Intr Oscillotar
        Config tab
    // TODO: USER CODE!!
    SET_TRIS_E (0B00001000);
    SET_TRIS_D (0B00000000);
while (1)
{
    //IZQUIERDA
    /**
    OUTPUT_D (0XFF);
    delay_us (800);
    OUTPUT_D(0X00);
    delay_us (1200);
    /**/
    //CENTRO
    /*
```

```

        OUTPUT_D (0XFF);
        delay_us (1500);
        OUTPUT_D (0X00);
        delay_us (500);
    */
    //DERECHA
    /*
        OUTPUT_D (0XFF);
        delay_us (2400);
        OUTPUT_D (0X00);
        delay_us (300);
    */
}
}

```

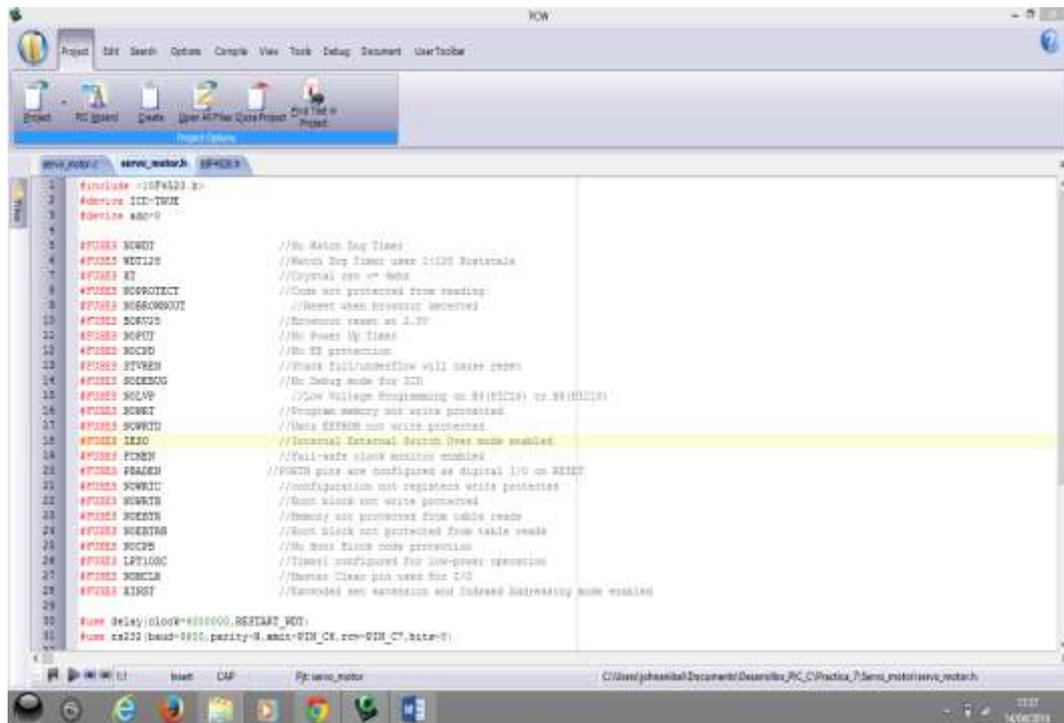
SOFTWARE DE CONTROL ARCHIVO .C.-

```

1  #include "C:\Users\johannibal\Documents\Desarrolla_FIC_C\practica_7\servo_motor\servo_motor.h"
2  #asm delay (clock=4000000, RESTART_M0);
3  #asm rs232 (baud=9600, parity=0, smir=PIN_C6, rsv=PIN_C7, bits=8);
4
5  void main()
6  {
7
8      setup_adc_ports (NO_ANALOGS | VCC_VDD);
9      setup_adc (ADC_OFF | ADC_TAD | ADC_0);
10     setup_adc_disable ();
11     setup_spi (SPI_SS_DISABLE);
12     setup_uart (MDT_OFF);
13     setup_timer_0 (KTCC_INTERNAL);
14     setup_timer_1 (T1_DISABLE);
15     setup_timer_2 (T2_DISABLE, 0, 1);
16     setup_comparator (NC_NC_NC);
17     setup_vref (FALSE);
18     //Setup_Oscillator parameters not selected from this Oscillator Config tab
19
20     // TODO: USER CODE //
21     SET_TRIS_E (0x00001000);
22     SET_TRIS_D (0x00001000);
23
24     while (1)
25     {
26
27         //INVIERNA
28         //***
29         OUTPUT_D (0XFF);
30         delay_us (1500);
31         OUTPUT_D (0X00);
32     }
33 }

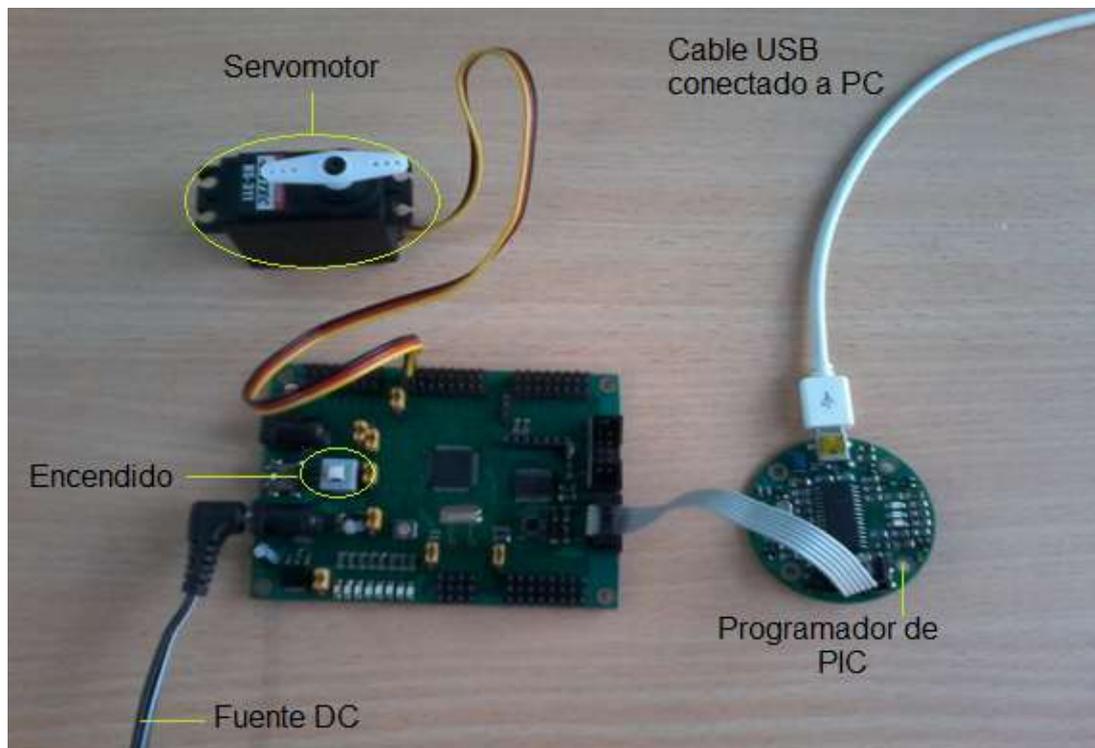
```

SOFTWARE DE CONTROL ARCHIVO .H.-



```
1 #include <16F628.H>
2 #define ICE_TRIM
3 #define abc=0
4
5 #FUSES NOWDT           //No Watch Dog Timer
6 #FUSES WDTPRS         //Watch Dog Timer uses 1:128 Postscale
7 #FUSES XT              //Crystal use 4M Hz
8 #FUSES NOPROTECT      //Data not protected from reading
9 #FUSES NOBROWNOUT     //Reset when brownout detected
10 #FUSES NOCPD          //No Code Protection
11 #FUSES NOCPOL         //No Power Up Timer
12 #FUSES NOCPOL         //No Code Protection
13 #FUSES STVREN         //Brown Out/underflow will reset device
14 #FUSES BROWNOUT       //No Debug mode for SW
15 #FUSES NOCLKDIV       //Low Voltage Programming on 8M(HLVD) or 8M(HLVDL)
16 #FUSES NOCLKDIV       //Brown Out/underflow will reset device
17 #FUSES NOCLKDIV       //No External Reset Over mode enabled
18 #FUSES FDS           //Full-time clock monitor enabled
19 #FUSES PPS           //PPS pins are configured as digital I/O on RESET
20 #FUSES NOWRT          //configuration not registers write protected
21 #FUSES NOWRT          //No Blank non write protected
22 #FUSES NOWRT          //Memory not protected from table reads
23 #FUSES NOWRT          //No Blank non protected from table reads
24 #FUSES NOWRT          //No Blank non protected from table reads
25 #FUSES NOWRT          //No Blank non protected from table reads
26 #FUSES LPT1000       //Times configured for low-power operation
27 #FUSES NOBURN        //Burner Clear pin used for I/O
28 #FUSES BURN          //Extended use watchdog and Address latching mode enabled
29
30 #asm delay(1000) #asm RESTART_DELAY
31 #asm xax20 (base=000, parity=0, smit=0, ck=0, vdr=0, vdr=0, vdr=0)
```

IMPLEMENTACIÓN.-



CONEXIONES DE PUERTOS.-

PUERTOS DE CONEXIÓN	SERVOMOTOR
PUERTO A	NC
PUERTO B	NC
PUERTO C	NC
PUERTO D	RD0 a RD7
PUERTO E	NC

NOTA.- Las siglas NC significan No conectado.