

ESCUELA POLITÉCNICA NACIONAL

ESCUELA DE FORMACIÓN DE TECNÓLOGOS

CONSTRUCCIÓN DE UN MÓDULO DIDÁCTICO PARA EL CONTROL CON MICROCONTROLADORES DE UN GLCD, MOTOR PAP-DC, SENSOR Tº Y TECLADO HEXADECIMAL, DESDE UN COMPUTADOR PERSONAL UTILIZANDO EL PUERTO USB PARA EL LABORATORIO DE MICROPROCESADORES Y REDES DE LA INFORMACIÓN.

PROYECTO PREVIO A LA OBTENCIÓN DEL TÍTULO DE TECNÓLOGO EN ELECTRÓNICA Y TELECOMUNICACIONES.

RAMÍREZ GUAÑA ALEX RODRIGO.

silver70_5@hotmail.com

DIRECTOR: ING. ALCÍVAR EDUARDO COSTALES G.

eduardo.costales@epn.edu.ec

QUITO, Junio 2010.

DECLARACIÓN

Yo Alex Rodrigo Ramírez Guaña, declaro bajo juramento que el trabajo aquí descrito es de mi autoría; que no ha sido previamente presentada para ningún grado o calificación profesional; y, que he consultado las referencias bibliográficas que se incluyen en este documento.

A través de la presente declaración cedo mis derechos de propiedad intelectual correspondientes a este trabajo, a la Escuela Politécnica Nacional, según lo establecido por la Ley de Propiedad Intelectual, por su Reglamento y por la normatividad institucional vigente.

Alex Rodrigo Ramírez Guaña

CERTIFICACIÓN

Certifico que el presente trabajo fue desarrollado por Alex Rodrigo Ramírez Guña, bajo mi supervisión.

**ING. ALCÍVAR COSTALES.
DIRECTOR DE PROYECTO**

AGRADECIMIENTO

Empezare dando las gracias a Dios, por permitirme nacer, y crecer junto a seres tan valiosos en mi cuidado, y dirección como mis padres. El presente trabajo va dirigido con una expresión de gratitud para mis distinguidos maestros que con nobleza, firmeza y entusiasmo vertieron sus conocimientos en nuestras almas, formando a su imagen, el modelo ideal de profesional que vieron en nosotros.

Y a mi querida Escuela de Tecnología porque en sus aulas recibí los más gratos recuerdos y momentos que nunca olvidare.

Atentamente, Alex Rodrigo.

DEDICATORIA

Al mejor regalo, que la vida puede darnos,
Al regalo, que ves cada mañana al despertar,
Al regalo, que ilumina tú camino,
Al regalo, que te ayuda en tus primeros pasos,
Al regalo, que te asiste si caes,
Al regalo, que se entristece si sufres,
Al regalo, que alegras por que sonríes,
Al regalo, que cela por que te quiere,
Al regalo, que protege si en peligro te ve,
Al regalo, que reprende porque ¡te ama!,
Al regalo, que espera que vuelvas pronto,
Al regalo, que le es suficiente el amor y el cariño,
El regalo, que triunfa si unido está,
El regalo, que es tan especial, porque estas tú,
Al regalo, que lo es todo para mi.

¡Mi Familia!

Para ti mi amada Ma. Raquel y mi primer amor Alan Rodrigo.

Atentamente: Papá.

CONTENIDO

DECLARACIÓN	i
CERTIFICACIÓN	ii
AGRADECIMIENTO	iii
DEDICATORIA	iv
CONTENIDO	v
RESUMEN	ix
PRESENTACIÓN	x
CAPITULO 1.	1
1.1 MICROCONTROLADORES.	1
1.1.1 CARACTERÍSTICAS GENERALES.	1
1.1.2 ARQUITECTURA DEL MICROCONTROLADOR.	1
1.1.3 ARQUITECTURA HARVARD Y SUS VENTAJAS.	2
1.1.4 PROCESADOR.	3
1.2 MICROCONTROLADORES PIC.	4
1.2.1 FABRICACIÓN DE MICROCONTROLADORES PIC.	5
1.3.1 CARACTERÍSTICAS MICROCONTROLADORES PIC18.	7
1.4 MICROCONTROLADOR USB PIC18F4550.	8
1.4.1 CARACTERÍSTICAS INTERNAS.	8
1.4.2 ENCAPSULADO DE 40PDIP.	9
1.4.3 ORGANIZACIÓN DE LA MEMORIA.	10
1.4.3.1 Memoria de Programa.	10
1.4.3.2 Memoria RAM de datos.	11
1.4.3.3 Registros de Función Especial (SFR).	12
1.4.3.4 Memoria EEPROM de datos.	13
1.4.3.5 Memoria de Configuración.	13
1.4.4 UNIDADES FUNCIONALES.	15
1.4.4.1 Puertos de entrada/salida (Anexo 1).	16
1.5.1 ANTECEDENTES HISTÓRICOS.	17
1.5.2 EXTENSIONES E INTERFACES RELATIVAS AL USB.	20

1.5.3	CARACTERÍSTICAS GENERALES DEL USB.	21
1.6	ESTÁNDAR USB 2.0	22
1.6.1	TOPOLOGÍA DEL BUS USB.	24
1.6.2	CAPA FÍSICA.	26
1.6.2.1	El host USB.	27
1.6.2.2	El Controlador del Host.	27
1.6.2.3	Dispositivos USB.	29
1.6.2.4	Hubs.	30
1.6.2.5	Funciones.	33
1.6.3	CAPA LÓGICA.	34
1.6.3.1	El flujo de datos del bus USB.	35
1.6.3.2	Puntos finales (endpoints) y direcciones de dispositivo.	35
1.6.3.3	Tuberías.	36
1.6.3.4	Tipos de transferencias.	36
1.7	SOFTWARE DE COMPUTADOR (COMPILADORES).	37
1.7.1	PROTON IDE (Integrated Developmet Environment).	37
1.7.1.1	Editor de Código.	39
1.7.1.2	Creando Aplicaciones.	39
1.7.1.3	Compilando.	39
1.7.1.4	Estadísticas.	40
CAPITULO 2.		41
2.1	CONSTRUCCIÓN DEL MÓDULO DIDÁCTICO.	41
2.1.1	DIAGRAMA DE BLOQUES MÓDULO DIDÁCTICO.	41
2.1.2	DISTRIBUCIÓN DE PÓRTICOS.	42
2.1.2.1	Pórtico A.	42
2.1.2.2	Pórtico B.	43
2.1.2.3	Pórtico C.	43
2.1.2.4	Pórtico D.	43
2.1.2.5	Pórtico E.	44
2.1.3	FUENTE DE ALIMENTACIÓN.	44
2.1.3.1	Características de Fuente de Alimentación.	45

2.1.3.2	Circuitos detectores de voltaje externo.....	45
2.1.4	POLARIZACIÓN DEL MICROCONTROLADOR.	46
2.1.4.1	Circuito externo de Reset.....	47
2.1.4.2	Frecuencia de Trabajo.	48
2.1.5	MÓDULO GLCD (GRAPHIC LIQUID CRYSTAL DISPLAY).	49
2.1.5.1	Introducción.....	49
2.1.5.2	Características Eléctricas.....	50
2.1.5.3	Bus de Datos y Control de GLCD.....	51
2.1.5.5	Polarización del Módulo GLCD.....	54
2.1.6	MÓDULO SENSOR DE TEMPERATURA.	56
2.1.6.1	Introducción.....	56
2.1.6.2	Diagrama de Bloques del Módulo Sensor de Temperatura.....	58
2.1.6.3	Funcionamiento.	58
2.1.7	MÓDULO TECLADO HEXADECIMAL 4 X 4.	61
2.1.7.1	Introducción.....	61
2.1.7.2	Diagrama circuital del Módulo Teclado Hexadecimal.....	64
2.1.7.3	Funcionamiento.	64
2.1.8	MÓDULO DE CONTROL DE MOTORES PASO A PASO.	65
2.1.8.1	Introducción.....	65
2.1.8.2	Principio de funcionamiento de motores Paso a paso.....	65
2.1.8.3	Diagrama circuital.....	69
2.1.9	MÓDULO – CONTROL DE MOTORES DC.	71
2.1.9.1	Introducción.....	71
2.1.9.2	Driver L293B.	71
2.1.9.3	Aplicación.....	74
2.1.9.4	Diagrama circuital del Módulo de Control de Motores DC.....	76
2.1.10	MÓDULO DE COMUNICACIONES USB.	77
2.1.10.1	Introducción.....	77
2.1.10.2	Diagrama circuital del Interfaz de Comunicaciones USB.	79
2.1.10.3	Cables y conectores USB.....	80
2.1.10.4	Software controlador de las comunicaciones USB.....	82
CAPITULO 3.		85

3.1	PRUEBAS DE DISEÑO.....	85
3.1.1	MÓDULO DIDÁCTICO ENSAMBLADO EN PROTOBOARD.	85
3.1.2	MÓDULO GLCD (GRAPHIC LIQUID CRYSTAL DISPLAY).	86
3.1.3	MÓDULO SENSOR DE TEMPERATURA.....	87
3.1.4	MÓDULO DE TECLADO HEXADECIMAL.	88
3.1.5	MÓDULO DE CONTROL DE MOTORES PASO A PASO.....	89
3.1.6	MÓDULO DE CONTROL DE MOTORES DC.....	90
3.2	CONSTRUCCIÓN DE TARJETA ELECTRÓNICA.....	91
3.2.1	SOFTWARE UTILIZADO.	91
3.2.2	DIAGRAMA CIRCUITAL ELABORADO EN ORCAD.	92
3.2.3	MÓDULO DIDÁCTICO – DIAGRAMA CIRCUITAL.....	93
3.2.4	MÓDULO DIDÁCTICO – CIRCUITO IMPRESO.	94
3.2.5	MÓDULO DIDÁCTICO – SCREEN DE TARJETA.....	95
3.2.6	TARJETA ELECTRÓNICA ENSAMBLADA.	96
3.2.7	PRUEBA DE COMUNICACIÓN USB.....	97
3.2.7.1	Generación de la interfaz USB en el PC.....	97
3.2.7.2	Módulo Encendido con conexión USB.	103
3.2.7.3	Control del Módulo Didáctico desde el computador vía USB.....	105
CAPITULO 4.....		109
4.1	CONCLUSIONES.....	109
4.2	RECOMENDACIONES.....	110
4.3	REFERENCIAS BIBLIOGRÁFICAS.....	111

RESUMEN

Lo que se busca con el presente proyecto es construir un Módulo Didáctico para el control de dispositivos eléctricos, electrónicos a través de un microcontrolador, que se comunica vía puerto USB con el computador, y desde cual el programador puede maniobrar los dispositivos conectados a la tarjeta electrónica.

Capítulo 1. Se ahonda en el tema sobre las características generales de los microcontroladores específicamente del microcontrolador PIC18F4550, que nos permitirá establecer una comunicación USB con el computador y controlar dispositivos conectados a él. Se cita información sobre interfaz de transmisión de datos USB 2.0 así como el software que asiste al programador para el desarrollo de aplicaciones de control.

Capítulo 2. Trata sobre el diseño del Módulo Didáctico en el que se cita y describe a cada dispositivo electrónico que forma parte de cada diagrama circuital respectivo, así como el funcionamiento general de cada módulo que compone a la tarjeta electrónica.

Capítulo 3. Este capítulo presenta visualmente las pruebas realizadas desde su ensamblaje en el módulo de prueba (protoboard), hasta las pruebas finales con la tarjeta electrónica ya ensamblada, mostrando los resultados en cada módulo que compone la tarjeta electrónica así como los software's de computador que asistieron en la construcción de la tarjeta electrónica como también para el control de la interfaz de comunicación USB entre el Módulo Didáctico y el Computador.

Capítulo 4. Se establecen las conclusiones y recomendaciones de la construcción del Módulo Didáctico, así como los aspectos que el usuario del Módulo Didáctico debe tener presente antes de utilizar el dispositivo implementado.

PRESENTACIÓN

Lo que se busca con la construcción del Módulo Didáctico es tener un dispositivo siempre al alcance de los estudiantes en el Laboratorio de Microprocesadores y Redes de la Información, que nos permita palpar la verdadera potencialidad práctica del uso de microcontroladores en el manejo y automatización de procesos a través de un ordenador (PC), así como generar fuentes de información que sigan alimentando los conocimientos y habilidades del futuro profesional, es el fin de éste trabajo.

CAPITULO 1.

1.1 MICROCONTROLADORES.

1.1.1 CARACTERÍSTICAS GENERALES.

Un microcontrolador es un computador completo (microprocesador + E/S + memoria + otros periféricos), aunque de limitadas prestaciones, que esta contenido con el chip de un circuito integrado programable y se destina a gobernar una sola tarea con el programa que reside en su memoria, sus líneas de entrada/salida soportan la conexión de sensores y actuadores a controlar desde el dispositivo.

1.1.2 ARQUITECTURA DEL MICROCONTROLADOR.

La arquitectura tradicional de computadoras y microprocesadores se basa en el esquema propuesto por John Von Neumann, en el cual la unidad central de proceso, ó CPU, está conectada a una memoria única que contiene las instrucciones del programa y los datos (figura 1.1).

El tamaño de la unidad de datos o instrucciones esta fijado por el ancho del bus de la memoria. Es decir que un microprocesador de 8 bits, que tiene además un bus de 8 bits que lo conecta con la memoria, deberá manejar datos e instrucciones de una o más unidades de 8 bits (1 bytes) de longitud. Cuando deba acceder a una instrucción o dato de más de un byte de longitud, deberá realizar más de un acceso a la memoria. Por otro lado este bus único limita la velocidad de operación del microprocesador, ya que no se puede buscar de memoria una nueva instrucción, antes de que finalicen las transferencias de datos que pudieran resultar de la instrucción anterior. Es decir que las dos principales limitaciones de esta arquitectura tradicional son:

- La longitud de las instrucciones esta limitada por la unidad de longitud de los datos, por lo tanto el microprocesador debe hacer varios accesos a memoria para buscar instrucciones complejas.

- La velocidad de operación (ancho de banda de operación) esta limitada por el efecto de cuello de botella que significa un bus único para datos e instrucciones que impide superponer ambos tiempos de acceso.

La arquitectura Von Neumann permite el diseño de programas con código auto-modificables, práctica bastante usada en las antiguas computadoras que solo tenían acumulador y pocos modos de direccionamiento, pero innecesaria, en las computadoras modernas.

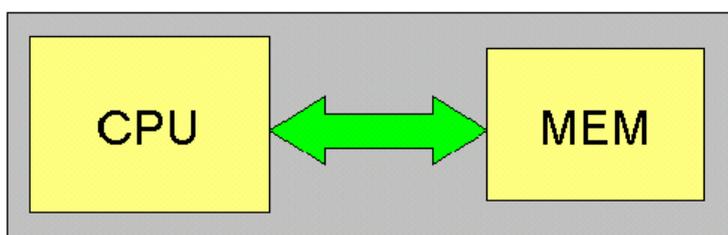


Figura 1.1 Arquitectura Von Newmann.

1.1.3 ARQUITECTURA HARVARD Y SUS VENTAJAS.

La arquitectura conocida como Harvard, consiste simplemente en un esquema en el que el CPU esta conectado a dos memorias por intermedio de dos buses separados. Una de las memorias contiene solamente las instrucciones del programa, y es llamada Memoria de Programa. La otra memoria solo almacena los datos y es llamada Memoria de Datos (figura 1.2). Ambos buses son totalmente independientes y pueden ser de distintos anchos. Para un procesador de Set de Instrucciones Reducido, o RISC (Reduced Instrucción Set Computer), el set de instrucciones y el bus de la memoria de programa pueden diseñarse de manera tal que todas las instrucciones tengan una sola posición de memoria de programa de longitud. Además, como los buses son independientes, el CPU puede estar accediendo a los datos para completar la ejecución de una instrucción, y al mismo tiempo estar leyendo la próxima instrucción a ejecutar. Se puede observar claramente que las principales ventajas de esta arquitectura son:

- El tamaño de las instrucciones no esta relacionado con el de los datos, y por lo

tanto puede ser optimizado para que cualquier instrucción ocupe una sola posición de memoria de programa, logrando así mayor velocidad y menor longitud de programa.

- El tiempo de acceso a las instrucciones puede superponerse con el de los datos, logrando una mayor velocidad de operación. Una pequeña desventaja de los procesadores con arquitectura Harvard, es que deben poseer instrucciones especiales para acceder a tablas de valores constantes que pueda ser necesario incluir en los programas, ya que estas tablas se encontrarán físicamente en la memoria de programa (por ejemplo en la EPROM de un microprocesador).

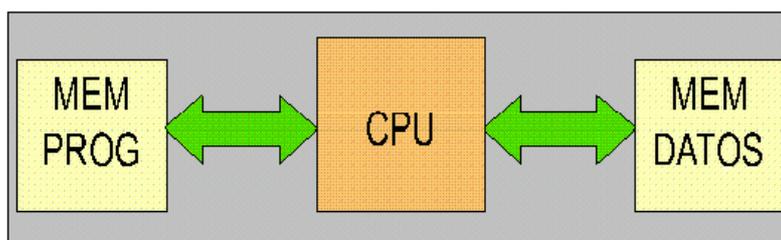


Figura 1.2 Arquitectura Harvard.

Los microcontroladores PIC18FXXXX poseen arquitectura Harvard, con una memoria de datos de 2048 Bytes, 256 bytes de EEPROM, una memoria de programa de 16.384 instrucciones.

1.1.4 PROCESADOR.

Es el elemento más importante del microcontrolador y determina sus principales características, tanto a nivel hardware como software. Se encarga de direccionar la memoria de instrucciones, recibir la instrucción en su curso, su descodificación y la ejecución de la operación que implica dicha instrucción, así como la búsqueda de los operandos y el almacenamiento del resultado. Existen tres orientaciones en cuanto a la arquitectura u funcionalidad de los procesadores actuales.

- **CISC.** Un gran número de procesadores usados en los microcontroladores

están basados en la filosofía CISC (Computadores de Juego de Instrucciones Complejo). Disponen de más de 80 instrucciones máquina en su repertorio algunas de las cuales son muy sofisticadas y potentes, requiriendo muchos ciclos para una ejecución.

- **RISC:** Tanto la industria de los computadores comerciales como la de los microcontroladores están decantándose hacia la filosofía RISC (Computadores de Juego de Instrucciones Reducido). En estos procesadores el repertorio de instrucciones máquina es muy reducido y las instrucciones son simples y, generalmente, se ejecutan en un ciclo de máquina. La sencillez y rapidez de las instrucciones permiten optimizar el hardware y el software del procesador.
- **SISC:** (Computadores de Juego de Instrucciones Específico): En los microcontroladores destinados a aplicaciones muy concretas, el juego de instrucciones, además de ser muy reducido, es “específico”, es decir, las instrucciones se adaptan a las necesidades de la aplicación prevista.

1.2 MICROCONTROLADORES PIC.

Microchip ingresa al mercado con sus microcontroladores PIC, con grandes ventajas frente a sus competidores con algunas de las siguientes características con las que logró acaparar gran parte del mercado.

- Gran variedad de familias que permiten adaptar el microcontrolador a las necesidades de cada aplicación.
- Herramientas de desarrollo comunes.
- Gran variedad de unidades embebidas (temporizadores, USART, I2C, SPI, unidades de comparación/captura/PWM, convertidores A/D, USB, receptores/transmisores de RF, Ethernet, etc.).
- Precios competitivos.
- Buen soporte (descripción de pines, libros, notas de aplicación, seminarios, mucha información disponible en el Internet).

1.2.1 FABRICACIÓN DE MICROCONTROLADORES PIC.

PIC10: Microcontroladores de 8 bits, de bajo costo, de seis pines y bajas prestaciones.

PIC12: Microcontroladores de 8 bits, de bajo costo, de ocho pines y bajas prestaciones.

PIC16: Microcontroladores USB de 8 bits de uso tradicional para el desarrollo de dispositivos de ventana en lugar de flash, por ser un dispositivo de baja velocidad, poco apoyo a granel isocrónica y transferencias, además el sea borrable mediante rayos ultravioleta, en lugar de flash son desventajas que minimizan su vialidad.

Ejemplos:

- **PIC16C745.**

- 8Kbyte de Memoria de Programa.
- Velocidad baja de 1Mbps con dispositivos USB con 6 puntos finales (endpoints).
- ADC de 8 bits, 5 canales.
- (USART / SCI) Universal Síncrono/Asíncrono, Receptor/Transmisor.
- UV Cerdip de 28 pines, re-programable.

- **PIC16C765.**

- 8Kbyte de Memoria de Programa.
- Baja velocidad de 1Mbps de dispositivos USB con 6 puntos finales.
- ADC de 8 bits, 8 canales.
- USART / SCI (Universal Síncrono/Asíncrono, Receptor/Transmisor).
- UV Cerdip de 40 pines - borrable.

PIC18: Microcontroladores USB de 8 bits, con gran variedad de número de pines y prestaciones medias/altas, microcontroladores que trataremos a continuación.

PIC24: Microcontroladores de 16 bits y altas prestaciones.

1.3 FAMILIA DE MICROCONTROLADORES USB PIC18.

Microchip retoma el mercado con los microcontroladores USB PIC's basados en su popular serie de dispositivos PIC16CXXX con características fundamentales como las siguientes:

- Arquitectura RISC avanzada Harvard: 16 bits con 8 bits de datos.
- 77 instrucciones con 83 extensiones.
- Encapsulados desde 18 hasta 80 pines.
- Hasta 64Kbytes de memoria de programa (hasta 2 Mbytes en ROMless).
- Multiplicador Hardware 8x8.
- Hasta 3968 bytes de RAM y 1Kbyte en EEPROM.
- Frecuencia máxima de reloj 40MHz, hasta 10 MIPS.
- Pila de 32 niveles, múltiples fuentes de interrupción.
- Periféricos de comunicación avanzada (CAN y USB), entre otros.

Dentro de la familia PIC18FXXXX aparece el primer microcontrolador PIC Flash con interfaz USB 2.0 "Full Speed" y 48 Mhz de frecuencia de oscilación para un rango de transferencia de datos de 12 Mbps, éstas y otras características combinadas, con una gran variedad de periféricos y un avanzado control de alimentación con tecnología nanoWatt, proporcionan la solución completa para los diseñadores que trabajan con USB en aplicaciones médicas, industriales, etc.

La mayoría de microcontroladores USB están optimizados para aplicaciones del mercado de consumo y del PC más que para diseños industriales. La nueva familia de microcontroladores USB PIC de Microchip ofrecen los beneficios del USB "Full Speed" para un amplio rango de aplicaciones que pueden operar en entornos duros, industriales y conectados a ordenadores personales.

El objetivo de las aplicaciones para estos productos incluye el mercado del automóvil, aplicaciones industriales, médicas así como aplicaciones de consumo avanzadas y otros dispositivos conectados al PC o alimentados por baterías. Las características de la interfaz USB 2.0 "Full Speed" incluyen un transmisor/receptor en el dispositivo y un puerto paralelo para transferir datos directamente hacia periféricos externos con una mínima carga para el CPU.

Otras características para la familia son sus 24 o 32 Kbytes de memoria flash mejorada auto-programables, soportando actualizaciones vía el puerto USB. Con la tecnología flash de Microchip, PEEC celdas PMOS eléctricamente borrables, los dispositivos proporcionan un elevado número de ciclos de escritura/borrado 100.000 y más de 40 años de retención de datos.

Este microcontrolador también incluye otras características como son: 2 Kbytes de RAM, de los cuales 1 Kbyte puede ser dedicado al buffer USB, 256 bytes de memoria de datos EEPROM, módulo EUSART que soporta interfaz serie RS232, RS485 y LIN, puerto SSP de comunicaciones serie I2C™ y SPI™, convertidor A/D de 10 bits de resolución con más de 12 canales, 2 comparadores analógicos, módulo Captura/Comparación/PWM con 16 bits de resolución, 4 temporizadores (3 de 16 bits y uno de 8 bits), circuito detector de bajo voltaje, reset por caída de tensión programable y mejoras en la depuración en circuito con tres puntos de ruptura. Una de las consideraciones claves para los diseñadores de aplicaciones USB es la disponibilidad de soporte firmware.

Microchip ofrece un amplio conjunto de librerías para las aplicaciones con las "clases" más comunes, incluyendo el interfaz humano (HID), dispositivo de comunicación (CDC) y que se pueden personalizar para el cliente.

Los cuatro microcontroladores USB PIC y sus encapsulados son:

- PIC18F2550.
- PIC18F2455 en SOIC y SDIP de 28 pines y el
- PIC18F4550.
- PIC18F4455 en encapsulados DIP de 40 pines, TQFP y QFN de 44 pines.

1.3.1 CARACTERÍSTICAS MICROCONTROLADORES PIC18.

Device	Program Memory		Data Memory		I/O	10-Bit A/D (ch)	CCP/ECCP (PWM)	SPP	MSSP		EUSART	Comparators	Timers 8/16-Bit
	Flash (bytes)	# Single-Word Instructions	SRAM (bytes)	EEPROM (bytes)					SPI	Master I ² C™			
PIC18F2455	24K	12288	2048	256	24	10	2/0	No	Y	Y	1	2	1/3
PIC18F2550	32K	16384	2048	256	24	10	2/0	No	Y	Y	1	2	1/3
PIC18F4455	24K	12288	2048	256	35	13	1/1	Yes	Y	Y	1	2	1/3
PIC18F4550	32K	16384	2048	256	35	13	1/1	Yes	Y	Y	1	2	1/3

Tabla 1.1 Familia de microcontroladores USB PIC18.

CARACTERÍSTICAS	PIC18F2455	PIC18F2450	PIC18F4455	PIC18F4450
Frecuencia de Operación	Hasta 48MHz	Hasta 48MHz	Hasta 48MHz	Hasta 48MHz
Memoria de Programa (bytes)	24.576	32.768	24.576	32.768
Memoria RAM de Datos (bytes)	2.048	2.048	2.048	2.048
Memoria EEPROM Datos (bytes)	256	256	256	256
Interrupciones	19	19	20	20
Líneas de E/S	24	24	35	35
Temporizadores	4	4	4	4
Módulos de Comparación/Captura/PWM (CCP)	2	2	1	1
Módulos de Comparación/Captura/PWM mejorado (ECCP)	0	0	1	1
Canales de Comunicación Serie	MSSP,EUSART	MSSP,EUSART	MSSP,EUSART	MSSP,EUSART
Canal USB	1	1	1	1
Puerto Paralelo de Transmisión de Datos (SPP)	0	0	1	1
Canales de Conversión A/D de 10 bits	10 Canales	10 Canales	13 Canales	13 Canales
Comparadores analógicos	2	2	2	2
Juego de instrucciones	75 (83 ext.)	75 (83 ext.)	75 (83 ext.)	75 (83 ext.)
Encapsulados	PDIP 28 pines SOIC 28 pines	PDIP 28 pines SOIC 28 pines	PDIP 40 pines QFN 40 pines TQFP 40 pines	PDIP 40 pines QFN 40 pines TQFP 40 pines

Tabla 1.2 Características de microcontroladores USB PIC18.

1.4 MICROCONTROLADOR USB PIC18F4550.

1.4.1 CARACTERÍSTICAS INTERNAS.

PARÁMETRO	CARACTERÍSTICA
Tipo de Memoria de Programa	Flash
Memoria de Programa (KB)	32
CPU Velocidad (MIPS)	12
RAM Bytes	2,048
EEPROM para datos (bytes)	256
Periféricos de Comunicación Digital	1-A/E/USART, 1-MSSP(SPI/I2C)
Periféricos de Captura/Comparación/PWM.	1 CCP, 1 ECCP
Temporizadores	1 x 8-bits, 3 x 16-bits
Convertidores ADC	13 canales de 10-bits
Comparadores	2
USB (canal, velocidad , compatibilidad)	1, Full Speed, USB 2.0
Rango de temperatura (°C)	-40 a 85
Voltaje de Operación (V)	2 a 5.5 Vdc
Encapsulado	40 pines

Tabla 1.3 Características del microcontrolador USB PIC18F4550.

CARACTERÍSTICAS

- Interfaz de alta velocidad USB 2.0 (12Mbit/s).
- 1Kbyte Dual Port RAM + 1K byte GP RAM.
- Transmisor de alta velocidad.
- 16 Endpoints (IN/OUT).
- Streaming Port.
- Resistencias internas Pull Up para D+ / D-.
- Funcionamiento a 48 MHz. (12 MIPS).
- Pin a pin compatible con PIC16C7X5.

Consumo (Anexo 1) :

- Menor a 2 mA, valor para 5V y 4 Mhz.
- 20 μ A para 3V y 32 MHz.
- Menor a 1 μ A en standby.

Tabla 1.4 Características de funcionamiento del PIC18F4550.

1.4.2 ENCAPSULADO DE 40PDIP.

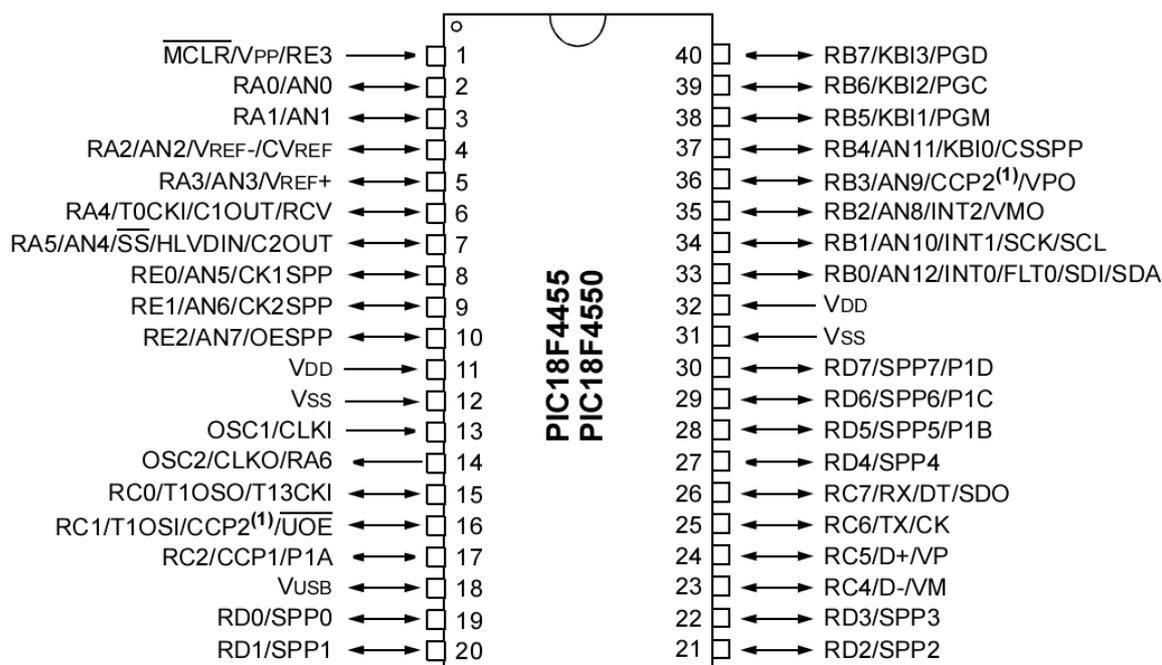


Figura 1.3 Encapsulado de 40 pines - PIC18F4550.

1.4.3 ORGANIZACIÓN DE LA MEMORIA.

El microcontrolador PIC18F4550 dispone de las siguientes memorias:

1.4.3.1 Memoria de Programa.

El PIC18F4550 dispone de una memoria flash interna de 32.768 bytes de memoria de programa (0000H-7FFFH). Las instrucciones ocupan 2 bytes (excepto CALL, MOVF, GOTO y LSFR que ocupan 4). Por lo tanto la memoria de programa puede almacenar hasta 16.384 instrucciones. La operación de lectura en posición de memoria de 7FFFH es igual a "0" como resultado (equivalente a la instrucción NOP). En resumen la memoria de programa almacena instrucciones y constantes/datos. Puede ser escrita/leída mediante un programador externo o durante la ejecución del programa mediante unos punteros

Direcciones especiales de la Memoria de programa.

- Vectorización del Reset es 0000H.
- Vectorización de las interrupciones de alta prioridad es la 0008H.
- Vectorización de las interrupciones de baja prioridad es la 0018H.

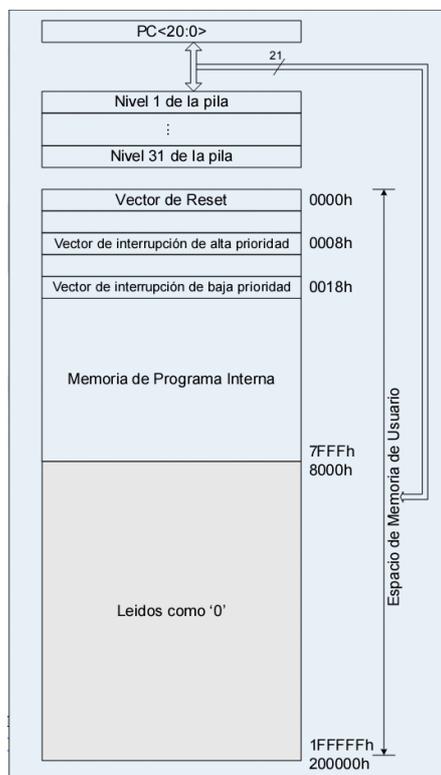


Figura 1.4 Memoria de Programa.

Los datos de la memoria de programa (Flash) pueden ser leídos, borrados y escritos durante la ejecución del programa. La operación que normalmente se utiliza en tiempo de ejecución es la de lectura de tablas o datos almacenados en memoria de programa.

1.4.3.2 Memoria RAM de datos.

El microcontrolador PIC18F4550 dispone de una memoria RAM de datos de 2048 bytes (8 bancos de 256 bytes). Además dispone de 160 bytes dedicados a los registros de función especial (SFR's) situados en la parte alta del banco 15. Para acceder a un byte de la memoria RAM de datos primero debe seleccionarse el banco al que pertenece el byte mediante el registro de selección del banco (BSR) y continuación direccionar el byte dentro del banco. Además existe una modalidad de acceso rápido a las 96 posiciones de la parte baja del banco 0 y los 160 Bytes del SFR's (banco de acceso rápido). Los bancos 4, 5, 6 y 7 se utilizan también para el puerto USB.

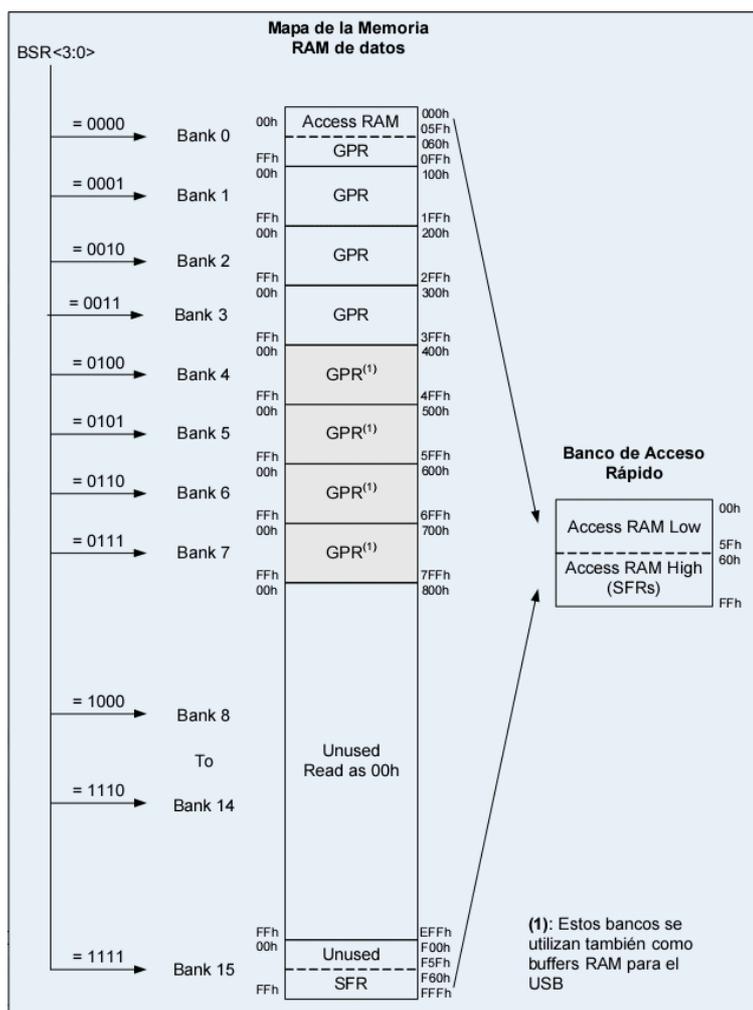


Figura 1.5 Memoria de Datos.

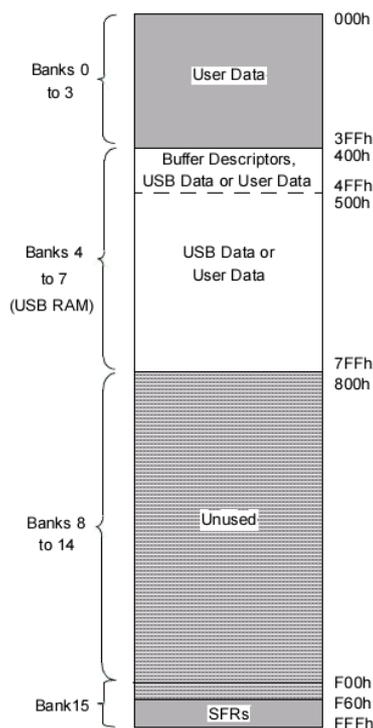


Figura 1.6 Memoria RAM USB en la Memoria RAM de Datos.

En resumen:

- Es una memoria SRAM de 2048 bytes en la que están incluidos los registros de función especial.
- Almacena datos de forma temporal durante la ejecución del programa.
- Puede ser escrita /leída en tiempo de ejecución mediante diversas instrucciones.

1.4.3.3 Registros de Función Especial (SFR).

La memoria RAM de datos se compone de Registros de Función General (GPR's) y de Registro de Función Especial (SFR's). Los SFR's son los registros mediante los cuales se pueden monitorear/controlar el funcionamiento de CPU y de las unidades funcionales del microcontrolador. Se distinguen dos conjuntos de SFR's: Los asociados con el núcleo del microcontrolador como:

- CPU: WREG, STATUS, BSR, etc.
- Interrupciones: INTCON, PIE1, PIR1, IPR1, etc.
- Reset: RCON.

SFR's asociados con las unidades funcionales:

- Temporizadores.
- Convertidores.
- Canales de comunicación.
- Puertos de entrada y salida, etc.

1.4.3.4 Memoria EEPROM de datos.

El PIC18F4550 dispone de una memoria EEPROM de datos de 256 bytes. Al ser una memoria no volátil los datos almacenados en ella se mantienen aún en ausencia de tensión de alimentación. El acceso a esta memoria se realiza mediante los SFR's: EECON1, EECON2, EEDATA, EEADR. Esta memoria permite hasta 1.000.000 de ciclos de borrado/escritura. Se puede leer/escribir de forma individual en cada una de las 256 posiciones de la memoria. Cuando se realiza una operación de escritura la circuitería interna del microcontrolador se encarga de borrar previamente la posición en la que se desea escribir. La duración de un ciclo completo de borrado/escritura de un byte en la memoria EEPROM suele ser de unos 4 ms.

En resumen:

- Es una memoria no volátil de 256 bytes.
- Almacena datos que se deben conservar aun en ausencia de tensión de alimentación.
- Puede ser escrita/leída en tiempo de ejecución a través de registros.

1.4.3.5 Memoria de Configuración.

Memoria en la que se incluyen los bits de configuración (12 bytes de la memoria flash) y los registros de identificación (2 bytes de memoria de solo lectura).

Se trata de un bloque de memoria situado a partir de la posición 30000H de memoria de programa (más allá de la zona de memoria de programa de usuario).

En esta memoria de configuración se incluyen:

- **Bits de configuración:** contenidos en 12 bytes de memoria flash permiten la configuración de algunas opciones del microcontrolador como:
 - Opciones del oscilador.
 - Opciones de reset.
 - Opciones de watchdog.
 - Opciones de la circuitería de depuración y programación.
 - Opciones de protección contra circuitería de memoria de programa y memoria EEPROM de datos.

Estos bits se configuran generalmente durante la programación del microcontrolador, aunque también pueden ser leídos y modificados durante la ejecución del programa.

- **Registros de identificación:** se trata de dos registros situados en las direcciones 3FFFEH y 3FFFFH que contienen información del modelo y revisión del dispositivo. Son registros de solo lectura y no pueden ser modificados por el usuario.

El microcontrolador PIC18F4550 dispone de buses diferentes para el acceso a la memoria de programa y memoria de datos:

Bus de Memoria de Programa:

- 21 líneas de dirección.
- 16/8 líneas de datos (16 líneas de instrucciones/8 líneas para datos).

Bus de la memoria de Datos:

- 12 líneas de dirección.
- 8 líneas de datos.

Esto permite acceder simultáneamente a la memoria de programa y a la memoria de datos. Es decir se puede ejecutar una instrucción mientras se lee de la memoria de programa la siguiente instrucción (proceso pipeline – figura 1.7).



Figura 1.7 Proceso Pipeline.

Por lo tanto la ejecución completa de una instrucción (lectura de la instrucción + ejecución) se hace en un ciclo de instrucción (4TOSC). Excepto las instrucciones que modifican el contenido del PC requieren dos ciclos de instrucción.

1.4.4 UNIDADES FUNCIONALES.

El microcontrolador PIC18F4550 dispone de una serie de Unidades Funcionales que le permiten:

- Realizar tareas específicas avanzadas (conversión A/D, transmisión/recepción de datos, generación de señales digitales con temporizaciones programables, etc.).
- Optimizar el rendimiento del microcontrolador, ya que estas unidades trabajan en paralelo a la CPU permitiendo que ésta se centre en otras operaciones como procesamiento de datos, cálculos, movimiento de datos, etc.

Las Unidades Funcionales más importantes del microcontrolador PIC18F4550 son:

- Puertos de E/S
- Temporizador 0
- Temporizador 1
- Temporizador 2
- Temporizador 3
- Convertidor A/D
- Unidad de comparación/captura/PWM mejorada (ECCP)
- Canal de comunicación serie EUSART.
- Canal de comunicación serie MSSP.

- Canal de comunicación serie USB.
- Módulo analógico de comparación.
- Canal de transmisión de datos en paralelo SPP.

1.4.4.1 Puertos de entrada/salida (Anexo 1).

El microcontrolador 18F4550 dispone de puertos que incluyen un total de 35 líneas digitales de E/S.

PUERTO	LÍNEAS DE ENTRADA Y SALIDA
Pórtico A	7 líneas de entrada y salida
Pórtico B	8 líneas de entrada y salida. Todas las líneas del puerto B disponen de Resistencias de Pull-up
Pórtico C	6 líneas de entrada y salida + 2 líneas de entrada. Puerto en el que hallan las líneas que permiten la comunicación USB
Pórtico D	8 líneas de entrada y salida. Todas las líneas del puerto D disponen de Resistencias de Pull-up
Pórtico E	3 líneas de entrada y salida + 1 línea de entrada.

Tabla 1.5 Puertos de E/S del PIC18F4550.

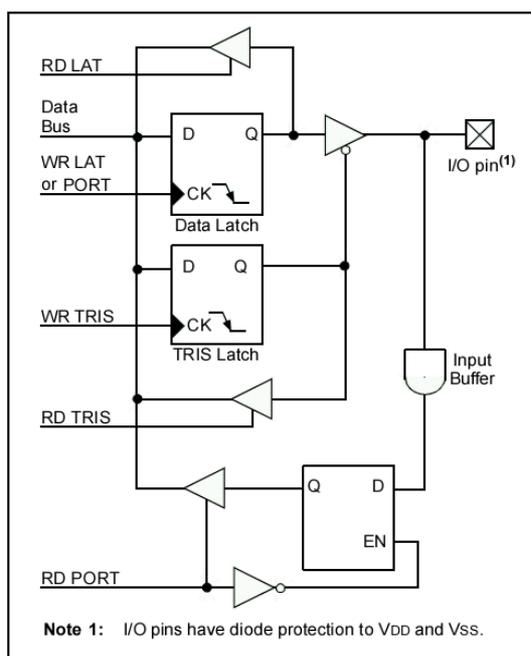


Figura 1.8 Diagrama de Bloques genérico de un pin de E/S.

1.5 PUERTO USB (UNIVERSAL SERIAL BUS).

El USB o Universal Serial Bus es una interfaz para la transmisión serie de datos y distribución de energía desarrollado por empresas líderes del sector de las telecomunicaciones y de los ordenadores que ha sido introducida en el mercado de los PC's, periféricos para mejorar las lentas interfaces serie RS-232 y paralelo. Provee una mayor velocidad de transferencia (de hasta 100 veces más rápido) comparado con el puerto paralelo de 25 pines y el Serial DB-9, DB-25, RS-232 que son los puertos que se encuentran en la mayoría de los computadores. Tenía en un principio como objetivo el conectar periféricos relativamente lentos (ratones, impresoras, cámaras digitales, unidades zip, etc.) de una forma realmente sencilla, rápida y basada en comunicaciones serie, aunque por sus características también podía conectarse hasta discos duros.

La interfaz USB de 4 hilos distribuye 5Vdc para la alimentación y puede transmitir datos a una velocidad de hasta 480 Mbps en su versión 2.0. Es un bus serie que hace posible la conexión de hasta 127 periféricos a una única puerta de un PC, con detección y configuración automáticas, siendo esto posible con el PC conectado a la red y sin tener que instalar software adicional, y reiniciar el ordenador (plug and play), algo que con los puertos convencionales serie y paralelo no sucedía. Tampoco hay que preocuparse por conflictos de IRQs o instalar tarjetas de adaptador para cada periférico. Estos periféricos pueden ser: ratones, teclados, impresoras, escáneres, grabadoras, discos duros, módems, cámaras digitales, etc.

1.5.1 ANTECEDENTES HISTÓRICOS.

Después de más de **12 años** desde su versión inicial, y casi **8 años** de su última versión, hay que comenzar a prepararse para la nueva versión de **USB**, de la que destaca su impresionante tasa de transmisión.

La especificación **USB 1.0** (denominada de baja velocidad) es la primera que se estableció en 1996, y debido a su baja velocidad (0,192MB/s) sólo se utiliza para dispositivos de interfaz humana como ratones, teclados, trackballs, etc.

Poco más tarde, en 1998, se lanzaría una nueva revisión (**USB 1.1**) que mejora la velocidad (hasta 1,5MB/s). Aunque es una mejora, se va volviendo cada vez más insuficiente para transferir información de varios megas de peso a medida que pasa el tiempo. Es entonces, cuando surge **USB 2.0**, un interfaz de alta velocidad (60MB/s) que consigue satisfacer las necesidades de transferencia y comienza a ser comercializado para discos duros externos, pendrives, etc.

Sin embargo, está programada la salida del nuevo **USB 3.0** para los próximos años, una nueva especificación en la que los fabricantes ya están realizando pruebas y se tiene como objetivo alcanzar 10 veces la velocidad de su predecesor 600MB/s. Una velocidad que nos mantendrá perplejos (sólo durante unos pocos años más) hasta la salida de un futuro **USB 4.0**.

En el siguiente gráfico (figura 1.9) podemos ver un ejemplo de lo que tardarían las transferencias de un fichero MP3 (aproximadamente 5Mb) y un fichero AVI (aproximadamente 700Mb) mediante las diferentes especificaciones USB 1.0, USB 1.1, USB 2.0 y USB 3.0.

USB Especificación	MB/s Tasa transferencia	MP3 5MB aprox.	AVI 700MB aprox.
1.0	0,192MB/s	 27seg	 1h
1.1	1,5MB/s	 4seg	 8min
2.0	60MB/s	 <1seg	 12seg
3.0	600MB/s	 <1seg	 1,5seg

Figura 1.9 Tasas de transferencia USB.

Hay que tener en cuenta la limitación de velocidad de los discos duros, puesto que si tenemos un disco duro lento, todas las operaciones que hagamos en la que intervenga, aunque tengamos un interfaz **USB 3.0** se verán limitadas por un cuello de botella de la tasa de transferencia del disco duro.

A continuación un listado de las **tasas de transferencia** de los distintos discos duros:

TECNOLOGÍA	DENOMINACIÓN	TASA DE TRANSFERENCIA
IDE UDMA 0	-	16,7Mb/s
IDE UDMA 1	-	25Mb/s
IDE UDMA 2	Ultra-ATA/33	33,3Mb/s
IDE UDMA 3	-	44,4Mb/s
IDE UDMA 4	Ultra-ATA/66	66,7Mb/s
IDE UDMA 5	Ultra-ATA/100	100Mb/s
IDE UDMA 6	Ultra-ATA/133	133Mb/s
Serial ATA	SATA1	150Mb/s
Serial ATA	SATA2	300Mb/s

Tabla 1.6 Tasa de Transferencia en discos duros.

Respecto a los cambios más técnicos, deberemos saber que en **USB 3.0** se mantendrá la retrocompatibilidad con su predecesor **USB 2.0**, salvo los conectores **mini** (usados en cámaras digitales y algunos móviles) que si cambiarán de formato.

El **consumo de energía** de los dispositivos USB 3.0 crecerá exponencialmente, pasando a poder requerir hasta un 80% más que su predecesor. Esto implicará un mayor gasto de energía (cuidado con los portátiles), pero también una mayor velocidad de carga y energía (por ejemplo, se podrán cargar móviles en poco tiempo).

También deberemos saber que se ha modificado el número de canales (de 4 a 9), implementando por separado el envío y recepción, permitiendo realizar las dos operaciones a la vez.

1.5.2 EXTENSIONES E INTERFACES RELATIVAS AL USB.

Actualmente, existen muchas y diferentes extensiones e interfaces relativas al **USB**, como por ejemplo, se confunde **USB 3.0** con **WUSB**. Así pues, vamos a proceder a explicarlos uno a uno.



USB 1.0 / 1.1: Cuando la interfaz USB nació, supuso una revolución por el tema de ser un dispositivo de conexión **en caliente** (conectar y listo, sin necesidad de reiniciar) y proporcionar la posibilidad de conectar hasta un número **máximo de 127 dispositivos**. Además de esto, que para aquel entonces, no era poco, USB 1.0/1.1 disponía de dos velocidades diferentes para transferir los datos; la primera de ellas diseñada para periféricos "directos" (teclados, ratones, trackballs, etc.) a una velocidad de **1.5Mbps**, y la segunda diseñada para otros periféricos (webcams, cámaras digitales, etc.) a la velocidad de **12Mbps**.



USB 2.0: Una notable mejora sobre el interfaz de USB 1.1 con respecto a la velocidad de transferencia de datos, que la aumenta aproximadamente cuarenta veces más, hasta llegar a la cifra de **480Mbps**. Esta interfaz nació expresamente para la transferencia de datos en grandes cantidades (discos duros, pendrives, etc.) de una forma más rápida y cómoda.



Wireless USB¹ (WUSB): Existe el difundido error de confundir esta extensión del protocolo USB con el próximo USB 3.0 que, en realidad, no ha salido aún a la luz. **Wireless USB** es una extensión del USB que combina el actual USB 2.0 con las capacidades de una transferencia inalámbrica. Sus características son de una velocidad de **480Mbps** a menos de 3 metros de distancia, y **100Mbps** a menos de 10 metros de distancia. Este sistema mejora bastante en comodidad al estándar 2.0 por el hecho de la tecnología inalámbrica (imaginemos la comodidad de pasar las fotos de una cámara digital al PC sin tener que realizar ninguna conexión de cables) pero también habrá que tener en cuenta las repercusiones de seguridad.

¹ No hay que confundir el **Certified Wireless USB** con el **Wireless USB** de Cypress Semiconductor. Mientras el primero es el estándar que busca ser aprobado, el segundo es un protocolo diseñado para periféricos "directos" que funcionan a una velocidad de **1Mbps** a 10m de distancia y **62,5Kbps** a 50m.



USB On-The-Go: Se trata de una mejora del interfaz USB, permitiendo elegir el estado de cada puerto USB (actuar como **dispositivo de datos** o como **servidor de datos**). Así pues, es posible transferir todos los datos desde un puerto USB actuando como servidor hasta otro actuando como dispositivo sin necesidad de usar un ordenador. Además es posible cambiar en cualquier momento el perfil de servidor a dispositivo o viceversa.

1.5.3 CARACTERÍSTICAS GENERALES DEL USB.

La especificación del USB proporciona una serie de características que pueden ser distribuidas en categorías. Estas características son comunes para todas las versiones (desde la 1.0 hasta la 2.0).

- Fácil uso para los usuarios.
- Modelo simple para el cableado y los conectores.
- Detalles eléctricos aislados del usuario (terminaciones del bus).
- Periféricos auto-identificativos, acoplados y reconfigurados dinámicamente (Hot Swappable).
- Flexibilidad y amplio rango de tamaños de paquetes, permitiendo variedad de opciones de buffering de dispositivos.
- Gran variedad de tasas de datos de dispositivos acomodando el tamaño de buffer para los paquetes y las latencias.
- Se garantiza un ancho de banda y bajas latencias apropiadas para telefonía, audio, etc.
- Cantidad de trabajo isócrono que puede usar el ancho de banda completo del bus.
- Control de flujo para el manejo del buffer construido en el protocolo.
- Amplia gama de aplicaciones y cargas de trabajo.
- Adecuando ancho de banda desde unos pocos Kbps hasta varios Mbps.
- Soporta tanto el tipo de transferencia isócrono como el asíncrono sobre el mismo conjunto de cables.

- Conexiones múltiples (hasta 127 dispositivos), soportando operaciones concurrentes de varios dispositivos.
- Soporta la transferencia de múltiples datos y flujos de mensajes entre el Host y los dispositivos.
- Robustez, manejo de errores y mecanismos de recuperación ante fallos implementados en el protocolo.
- Inserción dinámica de dispositivos.
- Soporte para la identificación de dispositivos defectuosos.
- Sub-canal de bajo costo a 1.5 Mbps.
- Adecuado para el desarrollo de periféricos USB como dispositivos de interfaz humana (HID) de bajo costo en conectores y cables.

En este proyecto se utilizan las especificaciones del estándar USB 2.0.

1.6 ESTÁNDAR USB 2.0

Esta versión que salió al mercado a mitad del 2000, tras la unión al consorcio de otras 3 compañías (Hewlett Packard, Philips y Lucent), con lo que volvían a ser siete las integrantes del USB-IF.

Como principal característica nos encontramos con un aumento de velocidad hasta 480 Mbps (casi 40 veces la velocidad anterior) con una diferencia de costo casi mínimo. De este aumento de velocidad le viene el nombre de Hi-Speed, que es fácil de confundir con Full-Speed. Full-Speed era el término que se le daba a las versiones USB 1.X que alcanzaban 12 Mbps, por lo que se debe tener en cuenta a la hora de hablar sobre ello. Este incremento de velocidad ha permitido sustituir en muchos casos el interfaz SCSI por el USB.

En total USB 2.0 soporta tres tipos distintos de velocidad:

- Hi-Speed 480 Mbps.
- Full-Speed 12 Mbps.
- Low-Speed 1.5 Mbps.

El alcance de uso para la versión USB 2.0 se ve incrementada notablemente respecto a sus predecesoras.

RENDIMIENTO	APLICACIONES	ATRIBUTOS
VELOCIDAD BAJA		
Dispositivos interactivos	Teclado y ratón	Bajo costo
10-100 Kb/s	Periféricos de entretenimiento (juegos y realidad virtual)	(Des) Conexión en caliente
	Configuración del monitor	Fácil uso
		Múltiples periféricos
VELOCIDAD MEDIA		
Teléfono	ISDN/RDSI	Bajo costo
Audio	PBX	(Des) conexión en caliente
Video comprimido	POTS	Fácil uso
500Kb/s – 10Mb/s	Audio	Múltiples dispositivos
		Garantizado el ancho de banda
		Garantizada la latencia
VELOCIDAD ALTA		
Video	Video	Gran ancho de banda
Unidades de almacenamiento	Discos duros y grabadoras	Latencia garantizada
Video comprimido		Fácil uso
25-500 Mb/s		

Tabla 1.7 Aplicaciones del Estándar USB 2.0

Básicamente, USB 2.0 incluye todo lo que ofrece USB 1.1 y añade el modo de alta velocidad. USB 2.0 también usa el mismo tipo de cables y conectores para conectar los dispositivos de alta velocidad, sin embargo los hubs USB clásicos

ralentizarán los dispositivos USB 2.0. Otro requisito es, que es necesario un controlador de host para USB 2.0 si queremos tener disponibles la conexión de alta velocidad con un dispositivo de este tipo.

Los hubs USB 2.0 tienen ahora mucho más trabajo que hacer que en el USB 1.1 ya que necesitan manejar todo el tráfico de tres tipo de dispositivos con velocidades distintas. Conectando un dispositivo USB 1.1 en uno USB 2.0 funcionaría bien, pero no lo haría si lo hiciéramos al revés, además de ralentizarse a 12 Mbps, posiblemente, el sistema operativo avisaría de su mal uso.

La forma de distinguir si un ordenador tiene USB de alta velocidad es mediante los logotipos registrados por el grupo promotor del USB (el USB-IF), como Hi-Speed. Dicho logotipo sólo lo pueden usar las empresas desarrolladas de hardware que hayan pasados los tests del USB-IF.

Cabe destacar que el USB 2.0 es simplemente una extensión y nunca llegara a reemplazar completamente al 1.1 ya que hay productos como teclados genéricos, ratones, joystick o altavoces que no requieren la gran velocidad que ofrece la tecnología USB 2.0. Sólo dispositivos de alta velocidad como Webcams o sistemas de alta capacidad necesitarán el máximo de velocidad, aunque los ordenadores los fabriquen sólo con puertos USB 2.0.

1.6.1 TOPOLOGÍA DEL BUS USB.

El Bus Serial Universal conecta los dispositivos USB con el host USB. La interconexión física USB es una topología de estrellas apiladas donde un hub es el centro de cada estrella. Cada segmento de cable es una conexión punto-a-punto entre el host y los hubs o función, ó un hub conectado a otro hub o función. El número máximo de dispositivos que puede conectar USB es de 127, pero debido a las constantes de tiempo permitidas para los tiempos de propagación del hub y el cable, el número máximo de capas permitido es de siete (incluida la capa raíz) con un máximo de longitud de cable entre el hub y el dispositivo de 5m. Cabe destacar que en siete capas, sólo se soportan cinco hubs que no sean raíz en una ruta de comunicación entre el host y cualquier dispositivo (figura 1.10). Un dispositivo compuesto ocupa dos capas, por eso, no puede ser activado si está acoplado en la última capa de nivel siete. Sólo las funciones pueden ponerse en este nivel.

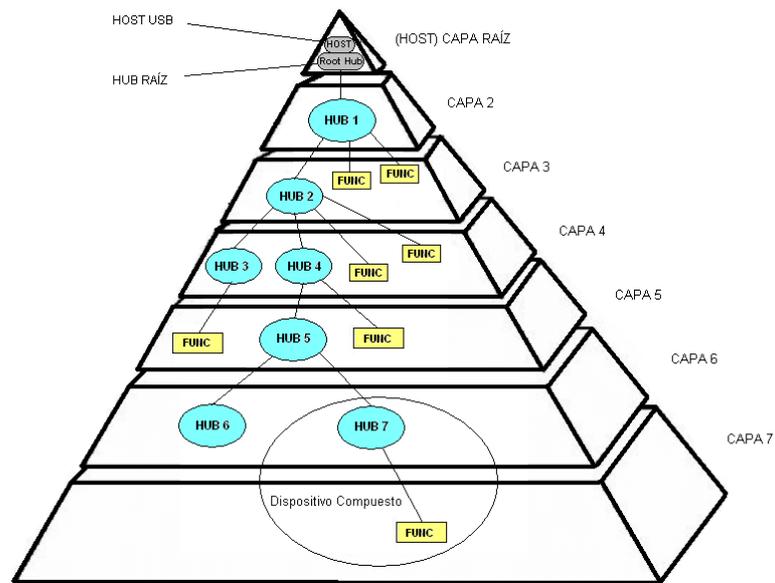


Figura 1.10 Estructura de capas del bus USB.

La topología del bus USB se puede dividir en tres partes:

- **La capa física:** Como están conectados los elementos físicamente.
- **La capa lógica:** Los roles y las responsabilidades de los elementos USB.
- **La relación software del cliente - función:** Como se ven mutuamente el software del cliente y los interfaces de las funciones relacionadas.

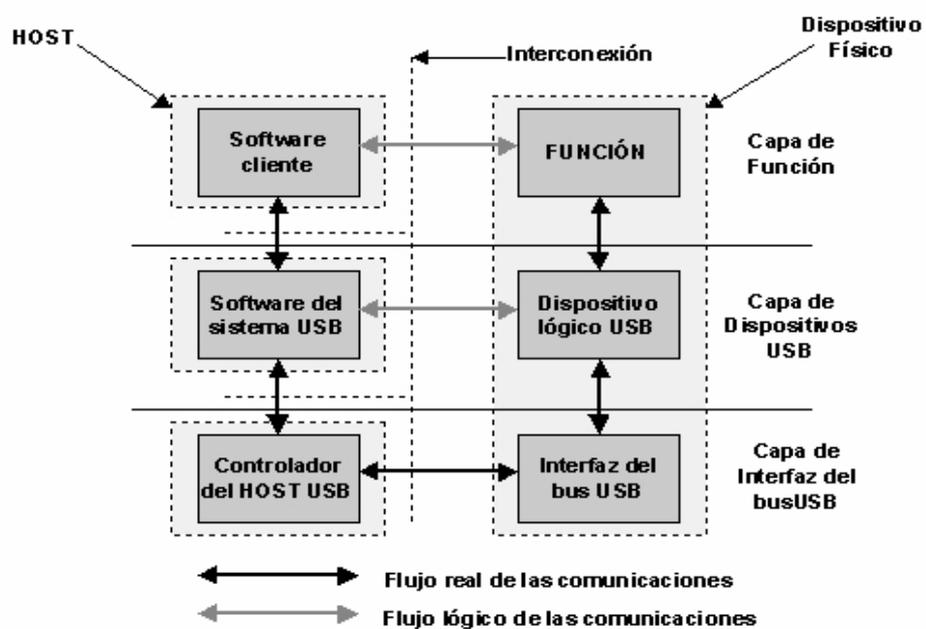


Figura 1.11 Flujo de las Comunicaciones.

1.6.2 CAPA FÍSICA.

La arquitectura física del USB se centra en las piezas de plástico y de metal con las que el usuario debe tratar para construir un entorno USB. Cada entorno físico USB esta compuesto por cinco tipos de componentes:

- El host.
- El controlador del host.
- Los enlaces (cables y conectores USB).
- Los dispositivos USB.
- Los hubs.

Los dispositivos están conectados físicamente al host a través de una topología en estrella. Los puntos de acople están provistos de una clase de dispositivos USB llamados hubs, los cuales tienen puntos de acople adicionales llamados puertos. Estos hubs se conectan a otros dispositivos a través de enlaces (cables de cuatro hilos). El host proporciona uno o más puntos de acople a través del hub raíz. Para prevenir los acoples circulares, se impone una estructura ordenada por capas en la topología de estrella y como resultado se obtiene una configuración al estilo de un árbol como se ve en la figura 1.12. Todas las comunicaciones físicas son iniciadas por el host. Esto quiere decir que cada milisegundo, o en cada ventana de tiempo que el bus lo permita, el host preguntará por nuevos dispositivos en el bus USB. Además el host inicia todas las transacciones físicas y soporta todas las transferencias de datos sobre la capa física.

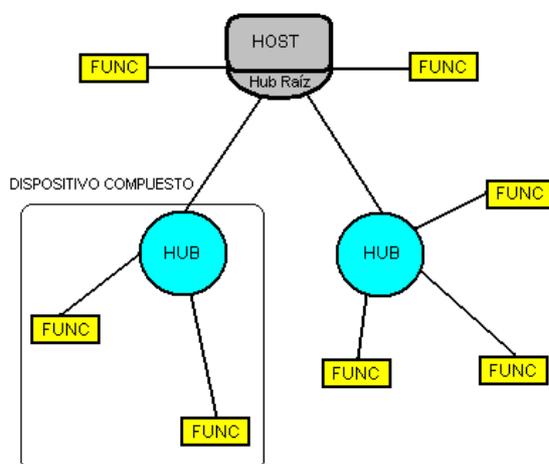


Figura 1.12 Topología física del bus USB.

1.6.2.1 El host USB.

El host es el sistema de computación completo, incluyendo el software y el hardware, sobre el cual se sostiene el USB. El host tiene la habilidad de procesar y gestionar los cambios de configuración que puedan ocurrir en el bus durante su funcionamiento. El host gestiona el sistema y los recursos del bus como el uso de la memoria del sistema, la asignación del ancho de banda del bus y la alimentación del bus. El host también ayuda al usuario con la configuración automática de los dispositivos conectados y reaccionando cuando son desconectados.

Un host puede soportar uno o más buses USB. El host gestiona cada bus independientemente de los demás. Los recursos específicos del bus como el ancho de banda asignado son únicos a cada bus. Cada bus está conectado al host a través de un controlador del host.

Sólo hay un host en cualquier sistema USB. Desde el interfaz USB hasta el sistema de host del ordenador es lo que se le llama controlador de host y puede estar implementado como combinación de hardware, firmware ó software. Integrado dentro del sistema de host hay un **hub raíz** que provee de un mayor número de puntos de acople al sistema.

Siempre que es posible, el software del USB usa el interfaz existente del sistema de host para gestionar las interacciones superiores. Por ejemplo, si un sistema de host usa la Gestión de Energía Avanzada (APM), el software del USB conecta al APM para interceptar, suspender las notificaciones.

1.6.2.2 El Controlador del Host.

El controlador de host está formado por el hardware y el software que permite a los dispositivos USB ser conectados al host. Este controlador es el agente iniciador del bus, es decir es el que comienza las transferencias en el bus. El controlador de bus es el maestro en un bus USB. Otros buses como PCI, permiten la presencia de múltiples maestros donde cada uno arbitra sus accesos al bus. En la arquitectura USB sólo hay un controlador de host por cada bus USB y por eso no hay arbitración para el acceso al bus.

Como las transferencias de datos de los dispositivos pueden ser basadas en datos o en la disponibilidad espacial del dispositivo, la mayoría de los controladores de

host están implementados como dispositivos maestros de bus PCI. Esto permite al controlador de host iniciar una transferencia de datos en el bus del sistema cuando le sea necesario, sin requerir la intervención del host de la CPU para cada transferencia. El controlador se comporta como un bus maestro PCI multicanal programable para dar soporte a las necesidades de transferencia de datos de múltiples dispositivos conectados al bus USB.

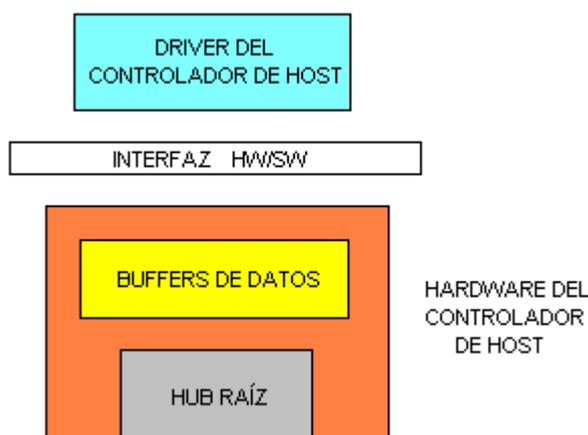


Figura 1.13 Controlador de bus USB.

La figura 1.13 muestra una vista conceptual del controlador de bus USB. La parte software consiste en el driver del controlador de host (HCD). Este software interactúa con el hardware del controlador de host a través del interfaz hardware/software.

La parte hardware del controlador de host consiste en un hub raíz que proporciona los puertos USB y los buffers de datos (colas) donde son almacenadas cuando son movidas a/desde memoria. El Host USB interactúa con los dispositivos USB a través del controlador.

Las funciones básicas del controlador de host son:

- Detectar la inserción o desconexión de dispositivos USB.
- Gestionar el flujo de control entre el host y los dispositivos.
- Gestionar el flujo de datos entre el host y los dispositivos.
- Coleccionar estadísticas de actividad y estado.
- Proveer una cantidad limitada de energía a los dispositivos conectados.

Hay dos implementaciones estandarizadas de la parte hardware de los controladores de host USB. Ambas proporcionan la misma funcionalidad y rendimiento para la interconexión. Estas implementaciones son:

- El Universal Host Controller Interface (UHCI) definido por Intel.
- Open Host Controller Interface (OpenHCI o OHCI) definido por Microsoft.

UHCI.- Esta definido como la parte software que tiene una gran responsabilidad para mantener el hardware en funcionamiento. Esto permite a esta implementación ser relativamente simple y realizarse con un bajo número de puertas.

OHCI.- Esta definida como la parte hardware que tiene más responsabilidad por el mantenimiento del flujo de datos, para que la parte software tenga menos trabajo que hacer. Esta otra implementación tiende a ser más compleja y tiene una cantidad mayor de puertas que la UHCI.

1.6.2.3 Dispositivos USB.

Un dispositivo es una colección de funcionalidad que lleva a cabo algún propósito de utilidad. Por ejemplo, un dispositivo podría ser un ratón, un teclado, una cámara, etc. Puede haber múltiples dispositivos simultáneamente en el mismo bus. Cada dispositivo lleva consigo información que puede ser útil para identificar sus características.

La información que describe al dispositivo se encuentra asociada con el canal de control. Esta información se divide en tres categorías:

- **Estándar:** Esta es la información cuya definición es común a todos los dispositivos USB e incluye elementos como la identificación del fabricante, la clase, la gestión de energía, etc.
- **Clase:** La definición de esta información varía dependiendo del aparato. Es una clasificación de los dispositivos en cuanto a sus prestaciones.
- **USB Vendor:** El fabricante del periférico puede poner aquí cualquier información deseada.

El software del host es capaz de determinar el tipo de dispositivo conectado haciendo uso de esta información y de un direccionamiento individual. Todos los dispositivos USB son accedidos por una dirección USB que es asignada dinámicamente cuando se conecta, asignándole también un número. Cada aparato soporta además uno o más canales a través de los cuales el host puede comunicarse con el dispositivo. Una vez ha sido reconocido e identificado el dispositivo, el software del host puede hacer que los driver's del dispositivo apropiados obtengan el control del nuevo dispositivo conectado. Cuando desconectamos el dispositivo, la dirección puede ser reutilizada para el próximo dispositivo conectado.

En cuanto a los tipos de dispositivos nos encontramos con dos clases:

- **Hubs**, proporcionan los puntos de acople adicionales al USB (figura 1.14).
- **Funciones**, que le dan al sistema la funcionalidad (HIDs, impresoras, unidades de almacenamiento, etc.)

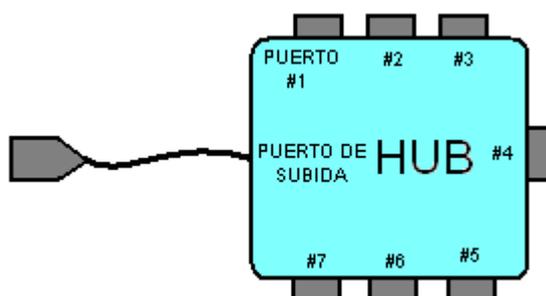


Figura 1.14 HUB típico.

1.6.2.4 Hubs.

Los Hubs son un elemento clave en la arquitectura Plug and Play del USB. Un bus tradicional puede ser dividido en segmentos de bus individuales conectados por puentes. Cada segmento tiene un número limitado de puntos de conexión debido a la limitada energía y la carga del bus. En la arquitectura USB, el número de puntos de conexión de dispositivos se expande añadiendo dispositivos únicos llamados **Hubs** o concentradores.

Estos dispositivos expanden la arquitectura del USB de dos formas:

- Incrementando los puntos de conexión a través de puertos adicionales.
- Proporcionando energía a los dispositivos al expandir el bus.

Los Hubs sirven para simplificar la conectividad USB desde la perspectiva del usuario y proporcionar robustez y complejidad a un precio relativamente bajo. Según esta arquitectura, se pueden expandir el bus acoplando hubs adicionales, interconectando dichos hubs mediante enlaces. Cada hub convierte un punto simple de conexión en múltiples puntos, donde estos puntos de conexión se les llaman puertos. Un hub requiere tener un puerto de subida y de 1ⁿ de bajada.

El **puerto de subida** (upstream port) de un hub conecta el hub al host. Es el que eléctricamente está más cerca del controlador de host. Está numerado como el puerto 0.

Los **puertos de bajada** permiten la conexión a otro hub o a una función. Están numerados del 1 a la n. Son los que más lejos están del controlador de host. Los hubs pueden detectar conexiones y desconexiones en cada puerto de bajada y activar la distribución de energía para cada dispositivo de bajada. Cada puerto puede ser individualmente activado para cada dispositivo High - Full - o Low-speed. Los hubs tienen dos formas de obtención de la energía: A través del puerto de subida (bus-powered) o de una fuente externa (self-powered).

1.6.2.4.1 Arquitectura Interna del Hub.

Un hub USB 1.x consiste en:

- **El repetidor HUB**, que es el responsable de gestionar la conectividad entre el puerto de subida y el de bajada, los cuales están operando a la misma velocidad.
- **El controlador del hub** hace posible el acceso del hub al host y viceversa.

Para la versión USB 2.0 hay un tercer elemento:

- **Traductor de transacciones** que proporciona los mecanismos que dan soporte a los dispositivos Full-Low-speed tras el hub mientras se transmiten todos los datos entre el host y el hub en el modo Hi-speed.

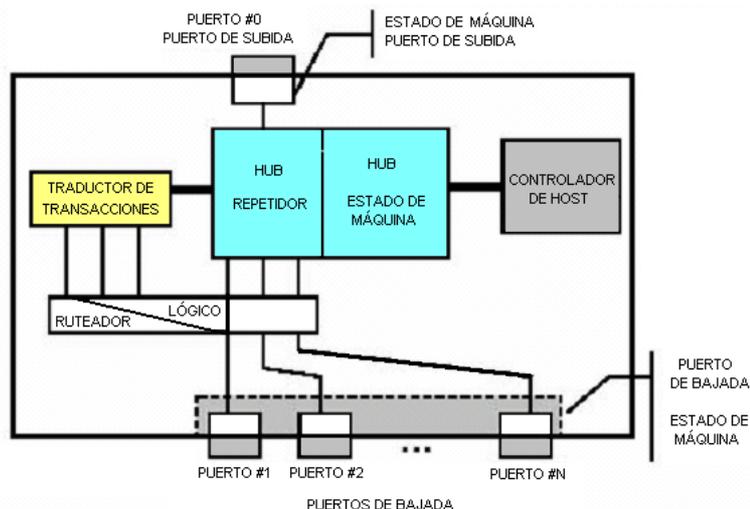


Figura 1.15 Arquitectura interna USB 2.0

Los hubs tienen diferente comportamiento en cuanto a conectividad dependiendo si están propagando el tráfico de paquetes, reanudando la señalización o están en estado **idle**.

Conectividad de Señalización de Paquetes.

El repetidor del hub contiene un puerto que debe estar siempre conectado en la dirección de subida, que es la conexión al host y uno o más puertos de bajada (la de los dispositivos). La conectividad de subida es la conectividad hacia el host y la de bajada hacia el dispositivo. Un hub también tiene un estado **idle** durante el cual el hub no está conectado. En este estado, todos los puertos del hub están en modo recepción esperando empezar el siguiente paquete.

Si un puerto de bajada está activo (en un estado donde puede propagar la señal a través del hub) y el hub detecta el comienzo de un paquete en ese puerto, se establece la conexión en dirección al puerto de subida del hub, pero no a cualquier otro puerto de bajada. Esto significa que cuando un dispositivo o un hub transmiten un paquete de subida, solo los hubs en línea entre el dispositivo que transmite y el host verán el paquete.

En la dirección de bajada, los hubs operan en modo broadcast. Cuando un hub detecta el comienzo de un paquete en su puerto de subida, establece la conexión a todos los puertos activos de bajada. Si un puerto no está activo no se propaga la señal de bajada por él.

1.6.2.5 Funciones.

Una función es un dispositivo USB que es capaz de transmitir y recibir datos o información de control sobre el bus. Típicamente se implementa como un periférico separado con un cable que se conecta en un puerto del hub. Sin embargo hay una gran flexibilidad a la hora de construir dispositivos. Una función simple pueden dar una funcionalidad simple (un micrófono, unos altavoces, etc.) o puede estar compuesto en distintos tipos de funcionalidad, como unos altavoces con un panel LCD. Este tipo de dispositivos se les llama función múltiples o composite device (dispositivo compuesto).

Otra forma de construir productos con múltiples funciones es creando un compound device (que significa también dispositivo compuesto). Este es el término usado cuando un hub está acoplado junto a múltiples dispositivos USB dentro de un mismo paquete. El usuario verá una sola unidad en el extremo del cable, pero internamente tiene un hub y varios dispositivos. Este tipo de "paquetes" tienen una dirección de bus para cada uno de los componentes, en contraposición a los composite devices que tienen una única dirección (figura 1.16).

Un ejemplo de un dispositivo compuesto sería un teclado USB que tuviera una conexión adicional para ratón ó nuestro módulo de entrenamiento. A pesar de que el teclado es un periférico, en este caso se le puede acoplar un ratón y por supuesto se necesitaría de un hub interno en el teclado para que esto pueda funcionar.

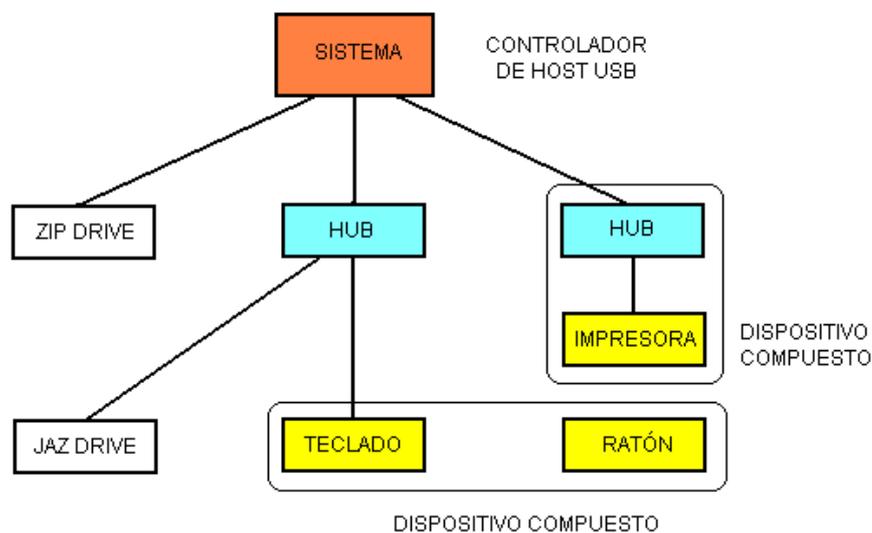


Figura 1.16 Composición típica de un dispositivo compuesto.

Cada función contiene la información sobre la configuración que describe su capacidad y requisitos en cuestión de recursos. Antes de que una función pueda ser usado debe ser configurado por el host. Algunos ejemplos de funciones pueden ser los siguientes:

- Una interfaz humana (HID) como ratón, teclado, tablas digitalizadoras o controladores de juegos, tarjetas de entrenamiento USB.
- Dispositivos de imágenes: cámaras, escáneres o impresoras.
- Dispositivos de almacenamiento: CD-ROMs, DVDs, disqueteras, etc.

1.6.3 CAPA LÓGICA.

La arquitectura lógica describe como unir el hardware del dispositivo USB a un driver del dispositivo en el host para que tenga el comportamiento que el usuario final desea. La vista lógica de esta conexión es la mostrada en el esquema siguiente (figura 1.17). En el podemos ver como el host proporciona conexión al dispositivo, donde esta conexión es a través de un simple enlace USB.

La mayoría de los demás buses como PCI, ISA, etc., proporcionan múltiples conexiones a dispositivos y los driver's lo manipulan mediante algunas combinaciones de estas conexiones (I/O y direcciones de memoria, interrupciones y canales DMA). Físicamente el USB tiene sólo un cable simple de bus que es compartido por todos los dispositivos del bus. Sin embargo, desde el punto de vista lógico cada dispositivo tiene su propia conexión punto a punto al host.

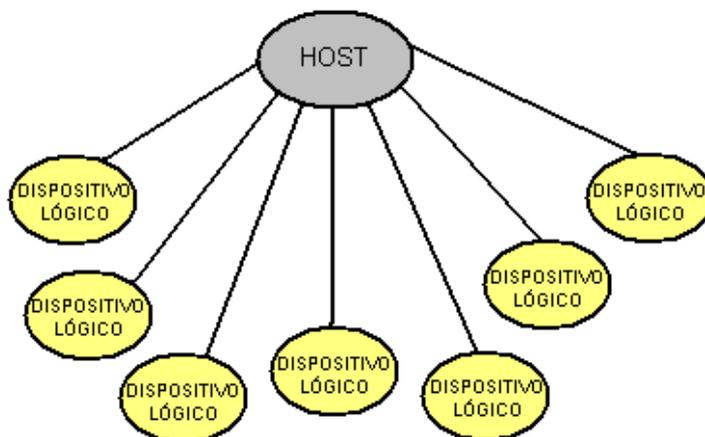


Figura 1.17 Capa lógica del USB.

Si lo miramos desde el punto de vista lógico, los hubs son también dispositivos pero no se muestran en la figura para la simplificación del esquema. Aunque la mayoría de las actividades de los host o de los dispositivos lógicos usan esta perspectiva lógica, el host mantiene el conocimiento de la topología física para dar soporte al proceso de desconexión de los hubs. Cuando se quita un hub, todos los dispositivos conectados a él son quitados también de la vista lógica de la topología.

1.6.3.1 El flujo de datos del bus USB.

Un dispositivo USB desde un punto de vista lógico hay que entenderlo como una serie de puntos finales o endpoints, a su vez los endpoints se agrupan en conjuntos que dan lugar a interfaces, las cuales permiten controlar la función del dispositivo. Como ya se ha visto la comunicación entre el host y un dispositivo físico USB se puede dividir en tres niveles o capas. En el nivel mas bajo el controlador de host USB se comunica con la interfaz del bus utilizando el cable USB, mientras que en un nivel superior el software USB del sistema se comunica con el dispositivo lógico utilizando la tubería de control por defecto ("*Default Control Pipe*"). En lo que al nivel de función se refiere, el software cliente establece la comunicación con las interfaces de la función a través de tuberías asociadas a endpoints.

1.6.3.2 Puntos finales (endpoints) y direcciones de dispositivo.

Cada dispositivo USB está compuesto por una colección de puntos finales o endpoints independientes, y una dirección única asignada por el sistema en tiempo de conexión de forma dinámica. A su vez cada endpoint dispone de un identificador único dentro del dispositivo al que pertenece, a este identificador se le conoce como número de endpoint y viene asignado de fábrica. Cada endpoint tiene una determinada orientación de flujo de datos. La combinación de dirección, número de endpoint y orientación, permite referenciar cada endpoint de forma inequívoca. Cada endpoint es por si solo una conexión simple que soporta un flujo de datos en una única dirección, bien de entrada o bien de salida.

1.6.3.3 Tuberías.

Una tubería USB es una asociación entre uno o dos endpoints en un dispositivo, y el software en el host. Las tuberías permiten mover datos entre software en el host, a través de un buffer, y un endpoint en un dispositivo. Hay dos tipos de tuberías:

- **Stream:** Los datos se mueven a través de la tubería sin una estructura definida.
- **Mensaje:** Los datos se mueven a través de la tubería utilizando una estructura USB.

Además una tubería se caracteriza por:

- Demanda de acceso al bus y uso del ancho de banda.
- Un tipo de transferencia.
- Las características asociadas a los endpoints.

1.6.3.4 Tipos de transferencias.

La interpretación de los datos que se transmitan a través de las tuberías, independientemente de que se haga siguiendo o no una estructura USB definida, corre a cargo del dispositivo y del software cliente. No obstante, USB proporciona cuatro tipos de transferencia de datos sobre las tuberías para optimizar la utilización del bus en función del tipo de servicio que ofrece la función. Estos cuatro tipos son:

- Transferencias de control.
- Transferencias isócronas.
- Transferencias de interrupción.
- Transferencias de bultos ("Bulk").

1.7 SOFTWARE DE COMPUTADOR (COMPILADORES).

Existen diversos software de computador que nos permiten la comunicación entre el computador personal (PC) con microcontroladores externos vía USB para el control y automatización de procesos aquí mostramos los más importantes por flexibilidad de programación.

Los Compiladores de Mecanique son una herramienta poderosa de desarrollo para Microcontroladores PIC. Fueron diseñados para asistir al programador en el desarrollo de aplicaciones de control con un mejor rendimiento de los sistemas a mando. En el mundo el Microcontrolador PIC más utilizado es el de 8 bits, utilizado en gran cantidad de aplicaciones, junto con un lenguaje de programación (Lenguaje C o Visual Basic) apropiado, son la principal opción a tomar para desarrollar sistemas automatizados. Mecanique proporciona compiladores IDE (Integrated Developmet Environment) - Entorno Integrado de Desarrollo, muy avanzados, con ANSI el recopilador dócil, un gran número de librerías, documentación, y ejemplos suficientes listos a probar.

1.7.1 PROTON IDE (Integrated Developmet Environment).



Figura 1.18 Programa – Proton IDE.

Protón IDE es profesional con un Ambiente de Desarrollo Integrado visual poderoso (IDE), diseñado específicamente para el Protón Plus compilador. Protón IDE se diseña para acelerar el producto en desarrollo por el usuario en un ambiente cómodo, amistoso proporcionando flexibilidad y mando.

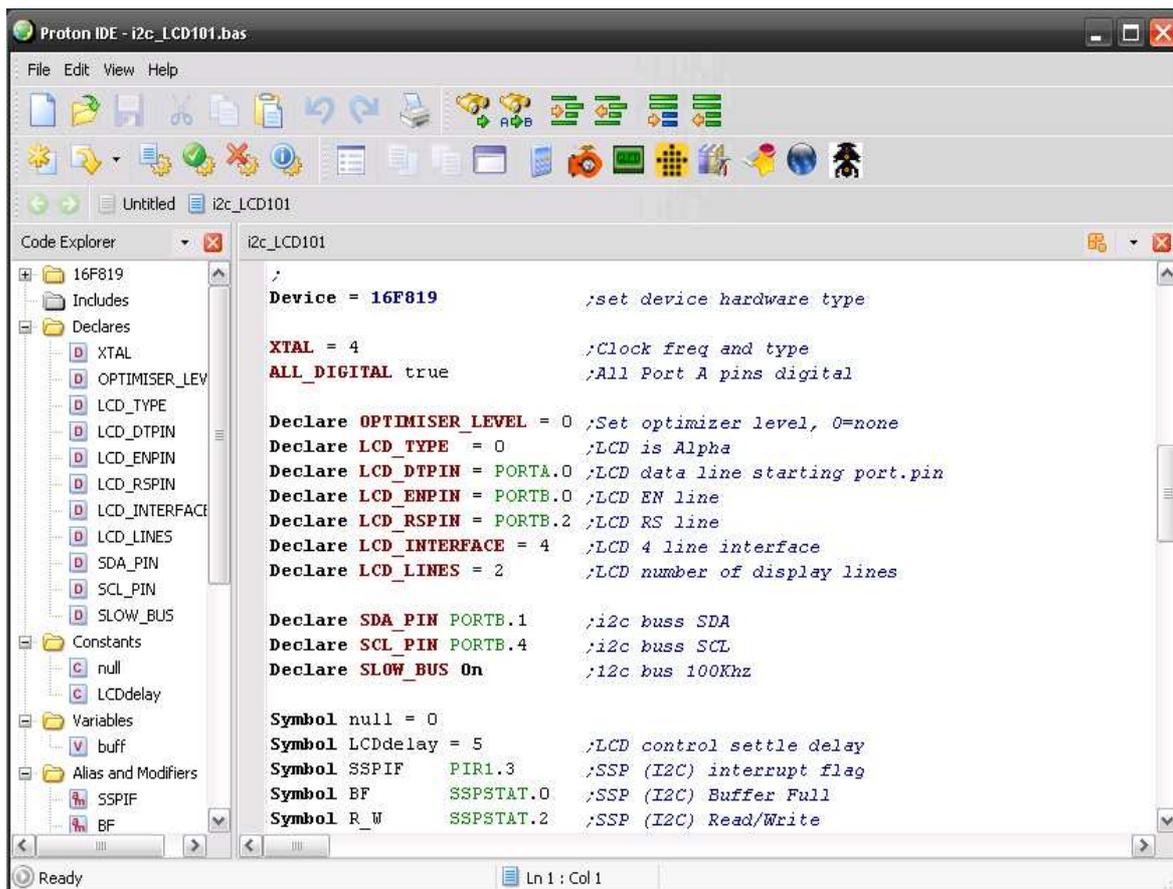


Figura 1.19 Ventana principal - PROTON IDE.

Los compiladores como **Proton IDE**, **mikroBasic** o **mikroC**, etc, de Mecanique le permiten desarrollar y resolver rápidamente aplicaciones complejas:

- Escribiendo el código fuente en el Editor del Código.
- Usando las librerías, para acelerar el desarrollo de la tarea: adquisición de los datos, la memoria, display's, las conversiones, las comunicaciones...etc.
- Generando comentarios y el código HEX estándar compatible con todos los programadores.
- Inspeccionando el flujo del programa, depurando y ejecutando la lógica.
- Detallando informes y gráficos en estadísticas.
- Generando un interfaz de comunicación USB.

1.7.1.1 Editor de Código.

El Editor del Código es el editor del texto avanzado formado para satisfacer las necesidades del profesional, es similar como trabajar con cualquier editor de texto, incluso el familiar copy (ctrl.+c) + paste (ctrl.+v) y deshacer acciones (ctrl.+z), son aplicables en estos compiladores. Los rasgos avanzados del Editor incluyen:

- Sintaxis ajustable
- Asistente de código.
- Ayudante de parámetros
- Plantillas del código.
- Auto corrección.
- Marcadores y Goto line.

1.7.1.2 Creando Aplicaciones.

Crear aplicaciones en Protón IDE es fácil e intuitivo. El asistente le permite generar un proyecto nuevo en simplemente pocos pasos: nombre del proyecto, elegir microcontrolador, ubicación, frecuencia de reloj.

Estos compiladores le permiten almacenar sus proyectos en cualquiera de las tantas unidades como usted encuentre apropiado. Usted puede compartir sus librerías mikroCompiled (files.mcl).

1.7.1.3 Compilando.

Cuando usted ha creado el proyecto y ha editado el código fuente, usted querrá compilarlo. Una manera rápida es presionando el icono engranaje.

La barra de progreso aparecerá informando sobre el estado de la compilación. Si existen errores, se notificará en la ventana de error. Si no hay ningún error, el compilador generará unos archivos en la carpeta en la que se almaceno el proyecto.

A continuación se redactan algunos de los más importantes.

- **Archivo (.hex):** este archivo se usa para programar MCU.
- **Archivo (.mcl):** (Binary mikro Compiled Library), muestra la distribución binaria de la aplicación que además puede ser incluida en otros proyectos.
- **Archivo (.lst):** archivo de lista, apreciación del reparto global de la memoria MCU, direcciones de las instrucciones, registros, rutinas, etc.
- **Archivo (.asm):** ensamblador en lenguaje humano con los nombres, simbólicos, extraído del archivo (.lst).

1.7.1.4 Estadísticas.

Después de compilar exitosamente, el compilador nos permite observar gráficos estadísticos en los que nos muestran el porcentaje de uso de la memoria, ventana de procedimientos, situaciones, detalles, ventana de la RAM y ROM.

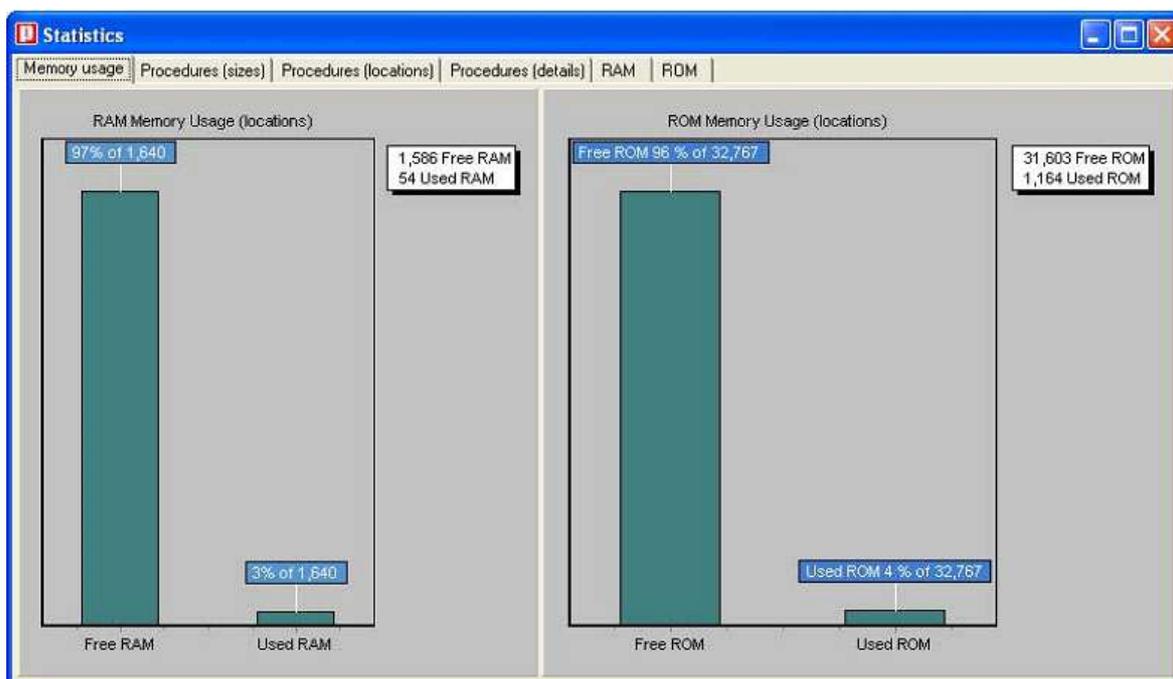


Figura 1. 20 Estadísticas del uso de la memoria.

CAPITULO 2.

2.1 CONSTRUCCIÓN DEL MÓDULO DIDÁCTICO.

Dentro de los diferentes microcontroladores USB que hemos citado anteriormente, él que mejor se ajusta a nuestras necesidades y demandas es el microcontrolador USB PIC18F4550.

El Módulo Didáctico está diseñado para trabajar únicamente con el microcontrolador PIC 18F4550 de 40 PDIP, ya que en total hace uso de 33 líneas (pines) del microcontrolador de tal manera que cada módulo que lo compone está estratégicamente asignado a un pórtilo para su gestión.

2.1.1 DIAGRAMA DE BLOQUES MÓDULO DIDÁCTICO.

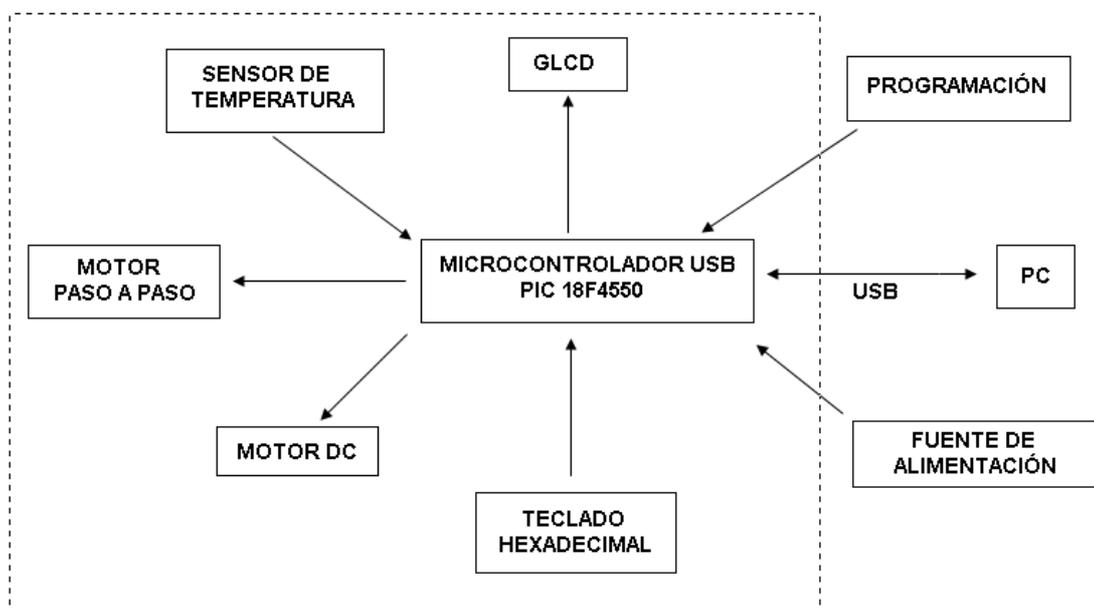


Figura 2.1 Diagrama de Bloques del Módulo Didáctico.

Como podemos observar en el diagrama de bloques (figura 2.1), el Módulo está conformado principalmente por el Microcontrolador USB PIC18F4550 con el que se conectan todos los dispositivos. Todos los circuitos, buses (datos y control), conexiones están completamente implementados excepto la fuente de

alimentación que será externa, cuyos conectores (jack hembra) quedarán listos para su conexión en la placa electrónica.

La tarjeta de entrenamiento consta de los siguientes módulos:

Periféricos de Salida.

- Módulo GLCD (Graphic Liquid Crystal Display).
- Módulo Motores Paso a Paso.
- Módulo Motores DC.

Periféricos de Entrada (Señales Análogas).

- Módulo Sensor de Temperatura.
- Teclado Hexadecimal.

Transmisión y Recepción de Datos.

- Comunicación USB.

2.1.2 DISTRIBUCIÓN DE PÓRTICOS.

El microcontrolador USB PIC18F4550, posee cinco pórticos bidireccionales (A, B, C, D y E) que incluyen un total de 35 líneas digitales configurables como entradas o salidas de datos, como lo determine el usuario (Diagrama Circuitual – Pág. 93).

Todas las líneas digitales de entrada y salida disponen de al menos una función alternativa asociada a una circuitería específica del microcontrolador. Cuando una línea trabaja en el modo alternativo no puede ser utilizada como línea digital de E/S estándar.

La tarjeta de desarrollo se ha construido de la siguiente manera:

2.1.2.1 Pórtico A.

Pórtico conformado por siete líneas bidireccionales (7 pines), de los cuales seis son maniobrables por el usuario, conocido como pórtico de entradas análogas,

por tal motivo en él se encuentra conectado el **Módulo Sensor de Temperatura** haciendo uso de un sólo canal análogo (RA0/AN0 – pin 2) de los trece canales análogos con los que dispone el microcontrolador. Las restantes cinco líneas del pórtico A son de utilidad para el manejo del **Teclado Hexadecimal 4X4**, que simplemente utiliza 5 líneas de las ocho que normalmente se requieren.

La línea restante (RA6 - pin 14) del Pórtico A no es maniobrable por el usuario ya que esta línea permite configurar la frecuencia de oscilación externa con la que trabajará el microcontrolador.

2.1.2.2 Pórtico B.

Pórtico que permite al usuario ejecutar interrupciones externas, en el cual se conectan las ocho líneas del bus de datos del GLCD (Graphic Liquid Crystal Display). El GLCD necesita seis líneas adicionales, que conforman el bus de control del **Módulo GLCD**, las cuales están tomadas del Pórtico D, 6 líneas de la siguiente manera (RD7, RD6, RD5, RD4, RD3, RD2).

2.1.2.3 Pórtico C.

Conocido como el Pórtico de las comunicaciones. Conformado por cinco líneas de entrada/salida (RC0, RC1, RC2, RC6 y RC7) y dos líneas de solo entrada (RC4, RC5). Se caracteriza por que en él se ubican los tres pines RC3 (V_{USB}), RC4 (D-), RC5 (D+) que permiten configurar la **Comunicación USB** con el PC. Mientras RC6 y RC7 en conjunto con dos líneas del pórtico D (RD0, DR1) controlan el **Módulo de Motor Paso a Paso**. Las tres líneas restantes del pórtico C (RC0, RC1, RC2) maniobran al **Módulo de Control de Motores DC** en conjunto con los tres únicos pines maniobrables del Pórtico E (RE0, RE1, RE2).

2.1.2.4 Pórtico D.

Dispone de ocho líneas de entrada y salida, seis de ellas (RD7, RD6, RD5, RD4, RD3, RD2) conforman el bus de control del GLCD, mientras que las restantes RD0 y DR1 forman parte del control del **Módulo de Motor Paso a Paso**.

2.1.2.5 Pórtico E.

Dispone de tres líneas de entrada/salida (RE0, RE1, RE2) y una línea de solo entrada (RE3). La línea RE3 (pin 1) por defecto tiene la función de **Reset externo** del microcontrolador. Las líneas maniobrables del pórtico E (RE0, RE1, RE2) operan al **Módulo de Motores DC** en conjunto con las líneas (RC0, RC1, RC2) del Pórtico C.

A continuación describimos cada diagrama lógico implementado:

2.1.3 FUENTE DE ALIMENTACIÓN.

El Módulo Didáctico posee una fuente de alimentación de corriente continua externa, con el fin de satisfacer la necesidad de una corriente eléctrica considerable para el normal funcionamiento de los módulos inmersos en la tarjeta electrónica, se ha elegido hacer uso de las bondades de una fuente Switchchig,

La fuente de alimentación externa (fuente Switchchig) proporciona al módulo un voltaje DC de 5V con una corriente de hasta 20A, para la polarización del microcontrolador y desarrollo de la lógica del módulo, a través de un conector jack hembra que se encuentra fijo en la tarjeta electrónica.

El Módulo tiene la opción de conectar otra fuente externa (conector jack hembra) para pruebas que requieran de una fuente de poder mayor, en el caso de los Módulos de Motor Paso a Paso y Motores DC.

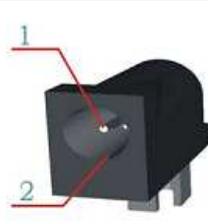
PIN	SEÑAL	GRÁFICO
1	V _{CC}	 <p>Conector hembra</p>
2	GND (tierra)	

Tabla 2.1 Conector para fuente externa.

Cada módulo mencionado anteriormente dispone de conectores (jack's, borneras, terminales, Switch, etc.) además de dispositivos de visualización (LED's) para verificar la presencia de voltaje, encendido y apagado de cada módulo.

2.1.3.1 Características de Fuente de Alimentación.

FUENTE SWITHCHIG.

Marca: RoHS

Modelo: 200XA.

Voltajes AC de Entrada: 115/10A.
230/5A.

Voltajes DC de Salida: + 5V / 20A , + 12V / 16A.
+ 3.3V / 14A , - 12V / 0.8A.
- 5 V / 0.5A , + 5 VSB / 2A.

Potencia Máxima: 550 W.

2.1.3.2 Circuitos detectores de voltaje externo.

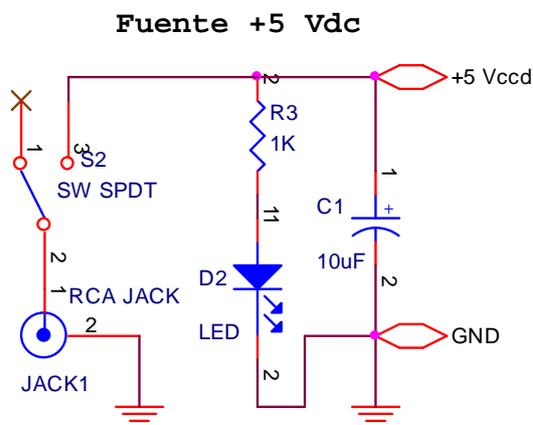


Figura 2.2 Circuito detector de 5Vdc de Fuente externa.

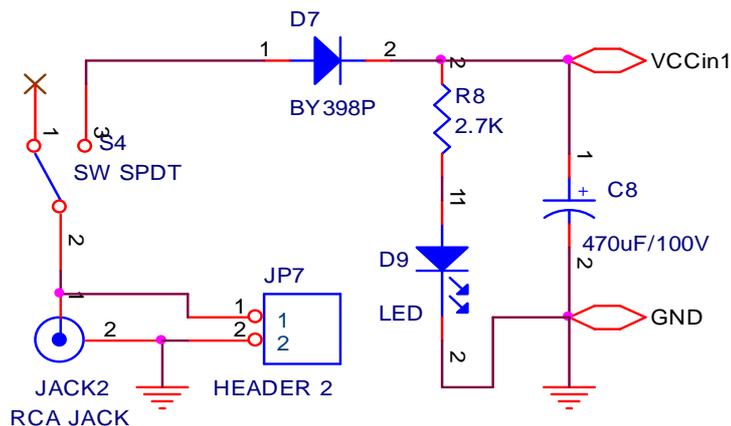


Figura 2.3 Circuito detector de Voltaje DC externo - Módulo de Motores DC.

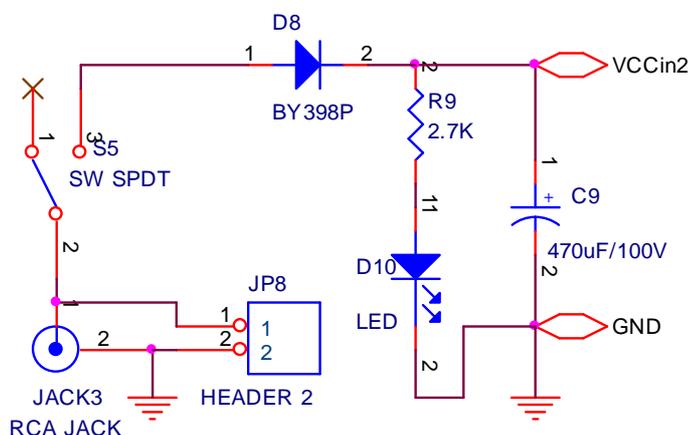


Figura 2.4 Circuito detector de Voltaje DC externo - Módulo de Motores Paso a Paso.

2.1.4 POLARIZACIÓN DEL MICROCONTROLADOR.

La fuente de alimentación externa provee de +5Vdc, con una corriente suficiente para el desarrollo de la lógica del microcontrolador, la tarjeta electrónica se polariza a través de los pines 11 y 32 (V_{dd}) del PIC, mientras que la tierra ó GND de la fuente de alimentación es común para toda la tarjeta electrónica por consiguiente, está conectado en los pines 12 y 31 (V_{ss}) del PIC (figura 2.5).

Como habíamos mencionado anteriormente la línea RE3 (pin 1) del pòrtico E del microcontrolador por defecto tiene la función de **Reset externo (MCLR)**.

2.1.4.1 Circuito externo de Reset.

Se ha dotado de un circuito para la opción **Master Clear Reset (MCLR)** del microcontrolador para casos en que se desee resetear manualmente el microcontrolador. El pin -MCLR es una entrada baja activa que permite un método externo de reseteo del dispositivo. El reseteo del microcontrolador se produce al ubicar un nivel bajo (0L – 0V) en el pin1 de reset.

Este dispositivo tiene un filtro de ruido que permite detectar e ignorar pequeños pulsos en el pin -MCLR del microcontrolador. Hay que tener en cuenta que puede ocurrir un reseteo automático del microcontrolador si el nivel de voltaje de polarización desciende al referencial (2.0 a 5.5 Vdc). El circuito de Reset externo es importante cuando el voltaje de polarización es inestable. El diodo que se ha ubicado en el circuito RESET ayuda a descargar rápidamente al capacitor, cuando el nivel de voltaje V_{dd} (5Vcd) desciende.

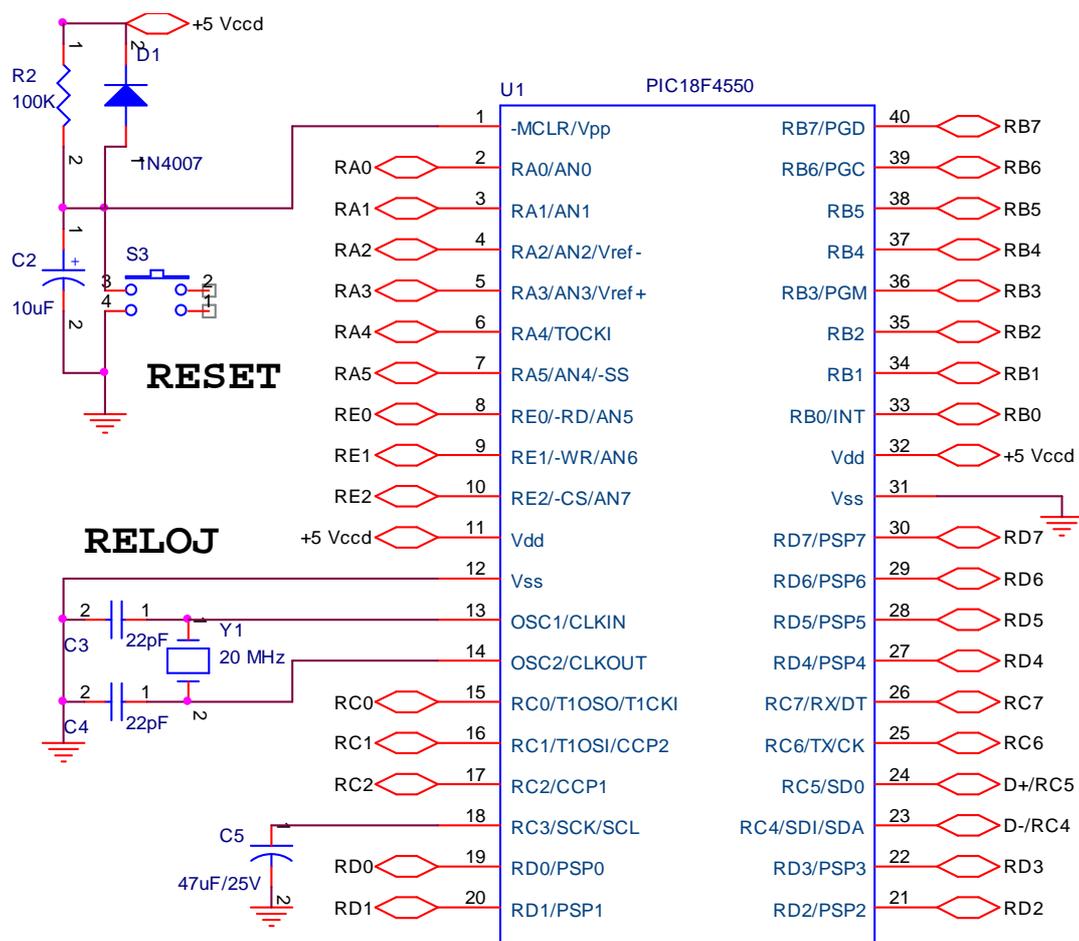


Figura 2.5 Polarización del Microcontrolador - Circuito de Reloj - Circuito Reset.

2.1.4.2 Frecuencia de Trabajo.

Los microcontroladores ejecutan cada instrucción en un determinado tiempo, generalmente lo hacen en un ciclo de máquina ($1CM = 4T$ para los microcontroladores), el cual es controlado por un cristal oscilador externo que para nuestro caso es de 20MHz, haciendo que cada instrucción se ejecute en un tiempo no mayor a los 0.2 us. La adición del módulo USB, en los microcontroladores 18FXXXX hace necesario una fuente del reloj estable, y necesario proporcionar una fuente del reloj separada que es dócil para trabajar en baja y alta velocidad USB (low and full speed).

El cristal oscilador y los condensadores cerámicos son conectados en los pines 13 (OSC1) y 14 (OSC2) como se muestra en la figura 2.5, para estabilizar la oscilación, de este modo el microcontrolador está configurado para trabajar en los modos de HS, HSPLL, XT, XTPLL que permiten trabajar dentro de las especificaciones USB 2.0.

- Modo XT Cristal/Condensador.
- Modo XTPLL Cristal/Condensador con PLL habilitado.
- Modo HS High-Speed Cristal/Condensador.
- Modo HSPLL High-Speed Cristal/Condensador con PLL habilitado.

Tipo de Oscilación	Frecuencia del Cristal	Valor típico del capacitor	
		C1	C2
XT	4 MHz	27pF	27pF
HS	4 MHz	27pF	27pF
	8 MHz	22pF	22pF
	20 MHz	15pF	15pF

Tabla 2.2 Configuraciones del Oscilación (HS, HSPLL, XT, XTPLL)

La serie de Microcontroladores USB de Microchip, incluye un circuito PLL (Phase Locked Loop) “bucle de fase cerrada” provisto específicamente para aplicaciones USB con una velocidad baja de oscilación.

2.1.5 MÓDULO GLCD (GRAPHIC LIQUID CRYSTAL DISPLAY).

Un display de cristal líquido está formado por una matriz de pequeños cristales que pueden ser manipulados de forma que presenten datos, letras, números, íconos, e incluso gráficos.

2.1.5.1 Introducción.

Las siguientes páginas intentan dar una breve descripción del funcionamiento del módulo GLCD (128X64 píxeles – controlador Samsung KS108/KS107- color azul) del Módulo Didáctico de Entrenamiento.

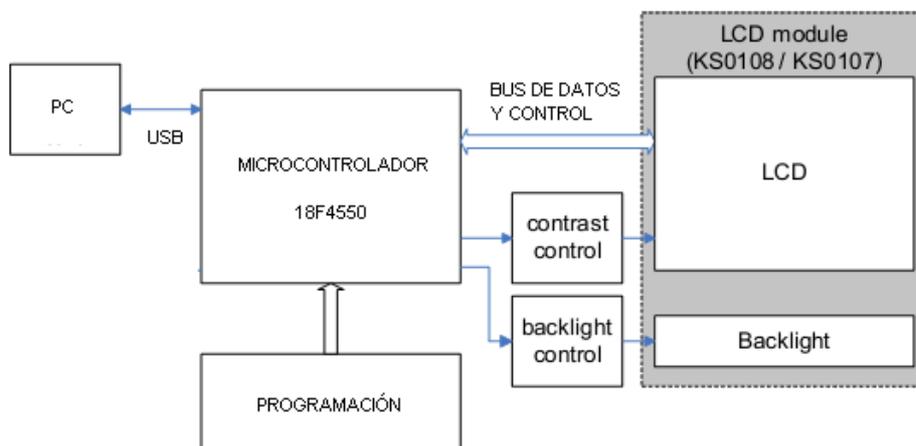


Figura 2.6 Diagrama de bloques del Módulo GLCD.

Se trata de un módulo microcontrolado capaz de representar datos, letras, números, íconos, gráficos (bitmaps), etc. A través de 8 líneas de datos (bus de datos) se envían códigos, comandos o instrucciones que se desean visualizar y que permiten realizar diferentes efectos de visualización en el LCD Gráfico. Igualmente mediante estas líneas el módulo devuelve información de su estado interno. Con seis líneas más (bus de control) se controla el flujo de información entre el GLCD y el microcontrolador que lo gestiona.

El GLCD posee veinte terminales de los cuales catorce (bus de datos + bus de control) son para el manejo del GLCD y los restantes son destinadas a su polarización, para el control de la luz de fondo (backlight) y variación del contraste del display. Ver Anexo 2.

A continuación se presenta la descripción de pines y señales empleadas por el GLCD (Display de Cristal Liquido Gráfico) determinado por el fabricante.

# PIN	SIMBOLO	NIVEL	FUNCIÓN
1	V_{SS}	0V	Tierra.
2	V_{DD}	+5V	Fuente de voltaje para la lógica.
3	V_O	-7.5V	Fuente de voltaje (-) para operar el LCD.
4	RS ó D / \bar{I}	H / L	H: ingresa dato / L: ingresa instrucción.
5	R / \hat{W}	H / L	H: Señal de lectura / L: Señal de escritura.
6	E	H, H→L	Ingresa señal de habilitación.
7-14	DB0-DB7	H / L	Bus de datos (8 bits).
15	CS1	H / L	Selector del controlador #1 – columnas 1-64. Selector del controlador #2 – columnas 65-128.
16	CS2	H / L	Selector de columnas 65-128.
17	RSTB	H / L	Señal de reset.
18	V_{EE}	-15V	Voltaje de salida (-) para el LCD.
19	BLA	V_{DD}	Fuente + para el LED (A) de la luz de fondo.
20	BLK	V_{SS}	Fuente – para el LED (K) de la luz de fondo.

Tabla 2.3 Descripción de pines GLCD.

2.1.5.2 Características Eléctricas.

Características DC (T=25°C, Vdc=5V, ±10%)

PARÁMETRO	SÍMBOLO	CONDICIÓN	MÍN	NORMAL	MAX	UNIDAD	
Voltaje de operación	V_{DD}		4.5	5.0	5.5	V	
Voltaje de entrada	Alto	V_{INH}	Terminales: E, R/ \hat{W} , D/I, DB0-DB7	2.0	--	V_{DD}	V
	Bajo	V_{INL}		0	--	0.8	
Voltaje de salida	Alto	V_{OH}	Terminales: DB0-DB7	2.4	--	--	V
	Bajo	V_{OL}		--	--	0.4	
Corriente de operación	I_{dd}	Display	--	--	2750	uA	

Tabla 2.4 Características eléctricas del GLCD.

2.1.5.3 Bus de Datos y Control de GLCD.

Como lo mencionamos anteriormente en la designación de los pórtricos, el manejo del GLCD está regido por el pórtrico B, que constituye el bus de datos, mientras que el bus de control está compuesto por seis líneas (RD7, RD6, RD5, RD4, RD3, RD2) del pórtrico D.

RBO-RB7: Líneas del pórtrico B que están conectadas a la líneas de datos DBO-DB7 del GLCD. A través del pórtrico B, se envían códigos de control al módulo GLCD ó se recibe por parte de éste, el estado interno del mismo. EL Pórtrico B habrá que programarlo como salida cuando se vayan a enviar códigos de control y como entrada cuando se desee conocer el estado interno del módulo GLCD (Display de Cristal Liquido Gráfico).

A continuación se describe cada una de las seis líneas del pórtrico D que conforman el **bus de control** del GLCD para mayor comprensión.

RD7: Línea del pórtrico D que se conecta con el pin de control RS ó D/Ī del GLCD. Al ubicar un nivel lógico bajo ó “0L” en la línea de entrada (RS ó D/Ī), se selecciona el registro de control del módulo GLCD para ingreso de una instrucción, mientras que ubicando un nivel lógico alto ó “1L” en este, se selecciona el registro de datos para su ingreso. **La línea (RD7) del microcontrolador PIC debe programarse como salida de datos.**

RD6: Línea del pórtrico D que se conecta con el pin de control R/W del GLCD. Al ubicar un nivel lógico bajo ó “0L” en este pin de control, el controlador del GLCD, permite la escritura de la información presente en el pórtrico B del microcontrolador que deberá previamente programarse como salida de datos.

Ubicando un nivel lógico alto o “1L” se lee el estado interno del módulo GLCD. Dicho estado se recibe a través del pórtrico B que deberá estar programado como entrada de datos al microcontrolador. **La línea RD6 del microcontrolador PIC debe programarse como salida.**

RD5: Línea del pórtilco D que entrega la señal de habilitación (Enable Signal) al pin de control E del GLCD. Cuando por el pin E del GLCD ingresa un pulso, mientras transcurre el flanco positivo el módulo queda habilitado, haciendo posible la lectura y transferencia de la información (datos) presente en el pórtilco B, a las líneas DBO-DB7 del GLCD; cuando transcurre el flanco negativo del pulso el dato es capturado y escrito, mientras que las líneas DBO-DB7 permanecen en alta impedancia. **La línea RD5 del microcontrolador PIC debe programarse como pin de salida.**

RD4: Línea del pórtilco D que conforma el bus de control para el manejo del GLCD, se conecta con el pin de control CS1 y es la línea por la cual se transmite la señal de selección del chip controlador del segmento #1 que rige a las columnas 1 a la 64 ó selección del chip controlador del segmento #2 que gestiona a las columnas 65 a la 128 del GLCD. Con un nivel lógico bajo o alto nos permite seleccionar el chip controlador del segmento #1 ó #2 para la visualización de la información. **La línea RD4 del pórtilco D del microcontrolador debe programarse como un pin de salida.**

RD3: Pin del pórtilco D que conforma el bus de control para el manejo del GLCD, se conecta con pin el CS2 en el GLCD y es la línea por la cual se transmite la señal (1L o 0L) de selección del chip controlador que gestiona de la columna 65 a la columna 128. **Ésta línea RD3 del pórtilco D del microcontrolador debe programarse como salida de datos.**

RD2: Línea del pórtilco D que se conecta con el pin de control RSTB del módulo GLCD y por donde se recibe la señal de reset. Cuando se transmite un nivel lógico bajo ó "0L" desde el microcontrolador a este pin, se ejecuta el reseteo del chip controlador de segmento #1 y #2 del módulo GLCD, con un nivel lógico alto ó "1L", el GLCD trabaja normalmente. **RD2 debe programarse como salida.**

Hay que tener presente que el manejo de un GLCD se realiza mediante una programación previa del microcontrolador PIC con un determinado software de computador, con la ayuda de subrutinas, instrucciones creadas por el programador o por el fabricante del GLCD.

2.1.5.4 Juego de instrucciones para el manejo básico del GLCD.²

A continuación se presenta una serie de comandos e instrucciones que permiten configurar diferentes opciones básicas de trabajo del Display de Cristal Líquido Gráfico y conseguir con ello distintos efectos de visualización.

El juego de instrucciones consiste en diferentes códigos que se envían desde el microcontrolador por el bus de datos y control (pórtico B y D) al GLCD para su mando.

Instrucción	R/W	D/I	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0	Función
Display On/Off	L	L	L	L	H	H	H	H	H	L/H	Controls the display on/off L: OFF, H: ON
Set Address	L	L	L	H	Y address (0~63)					Sets the Y address in the Y address counter.	
Set Page (X Address)	L	L	L	H	H	H	H	Page (0~7)		Sets the X address at the X address register.	
Display start Line	L	L	H	H	Display start line (0~63)					Indicates the display data RAM display at the top of the screen.	
Status read	H	L	B U S Y	L	O N / O F F	R E S E T	L	L	L	L	Read the status: Busy H: working L: ready ADC H: clockwise output L: counter clockwise ON/OFF: L: display Off H: display On RESET: L: reset H: normal
Write Display Data	L	H	Write Data					Write data (DB0...DB7) into display data RAM. After writing instruction, Y address is increased by 1 automatically.			
Read Display Data	H	H	Read Data					Read data (DB0~DB7) from display data RAM to the data bus			

Tabla 2.5 Códigos básicos de funcionamiento del GLCD.

² TS-12864A-2 LCD Specification Ver 1.0 Total 6 pages.

2.1.5.5 Polarización del Módulo GLCD.

La gran mayoría de displays suelen trabajar a 5 Vdc. Aunque existen módulos que trabajan a 3,3 Vdc pero la gama de productos es más limitada.

La tensión de trabajo del módulo GLCD es 5Vdc conectada al terminal V_{DD} (pin 2) del GLCD, pero existe otra tensión que dependiendo del display hay que generarla independientemente del LCD para polarizar el cristal liquido. En LCD's alfanuméricos esta tensión suele ser de -5 voltios. En los LCD Gráficos suele ser una tensión de -15 voltios, aunque en este caso, existen modelos en el que el propio display lo genera.

Con el voltaje (V_{EE} , -15Vdc) podemos modificar el **contraste** del display mediante el terminal variable del potenciómetro conectado al pin V_O del GLCD como se indica en el diagrama lógico del Módulo GLCD (figura 2.7). También hay la posibilidad de hacerlo por software aunque esto depende del controlador que lleve integrado el GLCD.

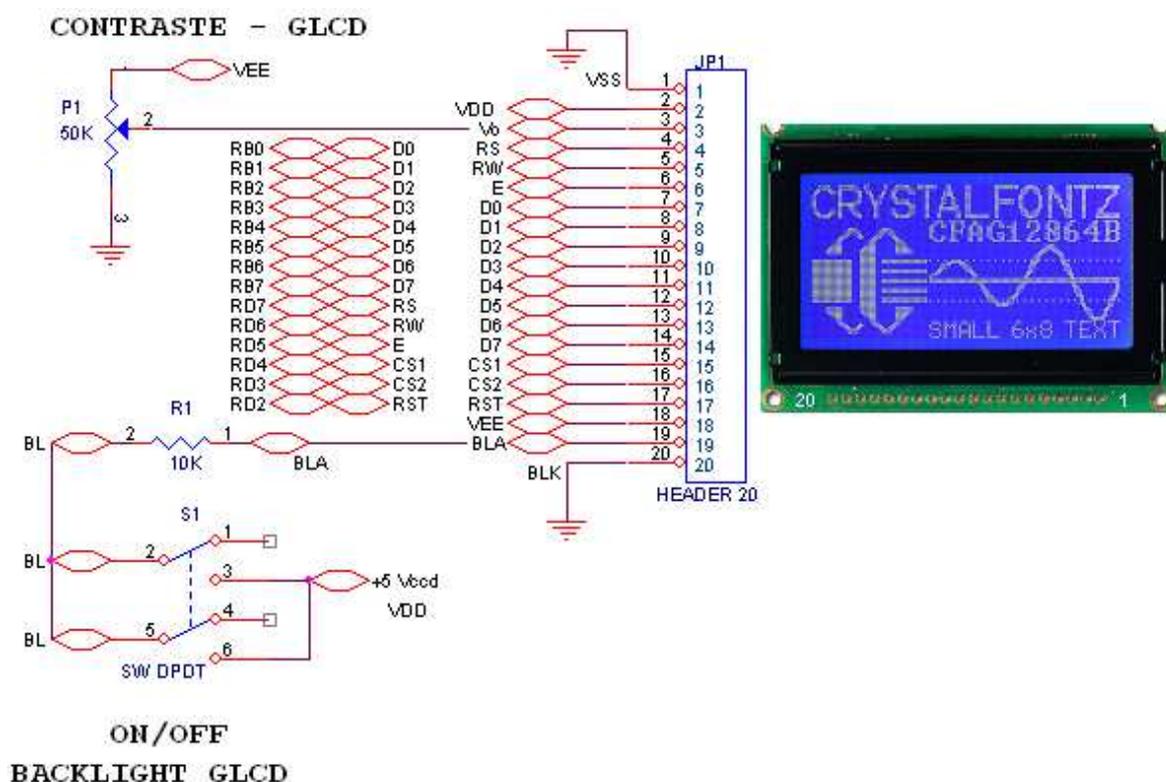


Figura 2.7 Diagrama circuital – Módulo GLCD.

Cabe destacar que el módulo GLCD posee un switch pulsador NC en serie con una resistencia que limita la circulación de corriente desde la fuente de alimentación externa (+5Vdc) hacia el ánodo del backlight del GLCD (pin 19 – figura 2.7). La razón de su ubicación es completamente didáctica³.

Es importante mencionar que el microcontrolador además de operar al GLCD está dispuesto para maniobrar un LCD alfanumérico 2X16, aprovechando la similitud de conexiones, así como el espacio disponible añadiendo una pequeña circuitería adicional.

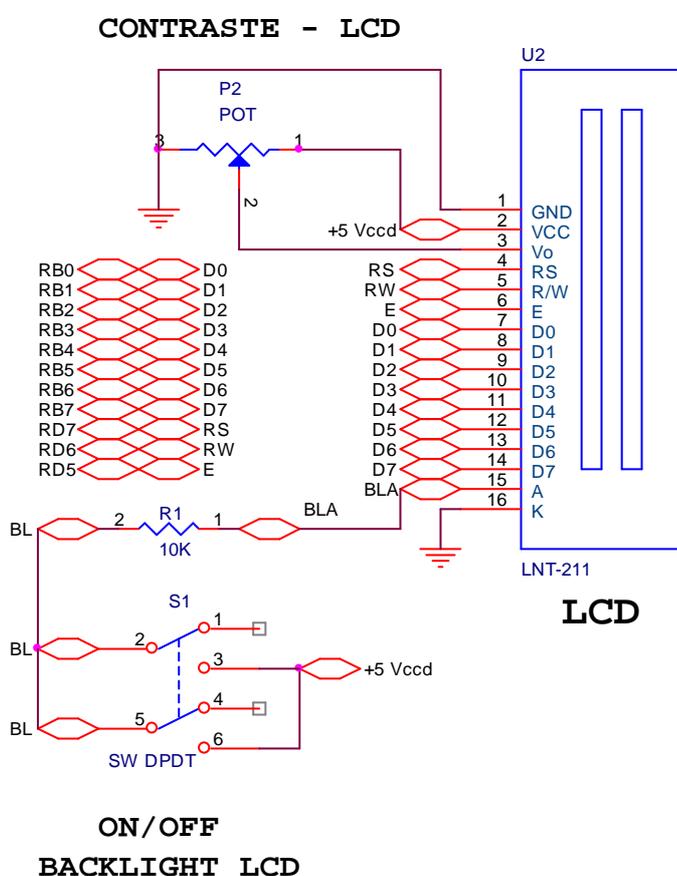


Figura 2.8 Diagrama circuital – Módulo LCD (Opcional).

³ Con este elemento podemos evitar apreciar la información basura que se genera y se almacena en registros, luego de ejecutar las instrucciones y al terminar un programa. Debido a que no todos los módulos estarán activos al mismo tiempo, este dispositivo me permite desactivar la luz de fondo (backlight) del GLCD e impedir observar dicha información que se almacena en pórticos que no se están utilizando en la práctica.

Tanto si el display es alfanumérico como gráfico, y dependiendo del diseño efectuado pueden contener drivers o, drivers y controladores.

Drivers: son circuitos integrados que lo que hacen es multiplexar la señal para poder activar cada píxel. Cuanto mayor es el display más píxels contiene, por tanto, mayor número de drivers serán necesarios.

Controladores: son circuitos integrados que ayudan a controlar al display de una forma más sencilla. Técnicamente es un interfaz entre los drivers que lleva el LCD ó GLCD y el hardware que se utiliza para el control (por ejemplo, un microcontrolador). Convierte unas señales para que el display pueda activar los píxeles de forma más fácil. Tienen una serie de comandos que suelen ser bastante Standard (speedy, blinking, clear, etc.) que simplifican las acciones que puede realizar el usuario con el LCD o GLCD. Además, incluso hay controladores que llevan incorporado un SET de caracteres de un determinado idioma que puede ser utilizado por el usuario.

2.1.6 MÓDULO SENSOR DE TEMPERATURA.

Ubicado en el pórtico de entradas análogas (Pórtico A), específicamente en el canal análogo RA0/AN0 (pin 2) que nos permite comunicarnos con el módulo interno de conversión análogo a digital que posee el microcontrolador PIC18F4550.

2.1.6.1 Introducción.

Existen muchos elementos para medición de temperatura en el mercado como: **Termopares**, **PT100** (Detector de película de platino), **PRC100** (sensor de resistencia de hilo arrollado de platino), el **CI LM35** (Sensor de Temperatura), etc. Para la implementación del módulo no se requiere que el sensor tenga un amplio rango de temperaturas, ni tampoco demasiada precisión. Por tal motivo se considera que la opción más indicada es la del circuito integrado LM35.

LM35: Es un circuito integrado sensor de temperatura con tres terminales, que proporciona una salida lineal de tensión de 10mV por grado centígrado (°C). Es un elemento económico y fácil de usar. El rango de temperaturas va desde -55 °C a 150 °C además tiene una precisión calibrada de aproximadamente $\pm 1^\circ\text{C}$.

Ejemplo:

- 1500mV = 150 °C.
- 250mV = 25 °C.
- -550mV = -55 °C.

Con el LM35 encapsulado de frente y hacia arriba tenemos que de izquierda a derecha los pines son: $+V_S$ (5Vdc) – V_{OUT} – GND.

TO-92 PLÁSTICO



Figura 2.9 Encapsulado LM35.

CARACTERÍSTICA	VALOR
Factor de escala lineal	+10mV/°C
Precisión a 25 °C	0.5 °C
Voltaje de Alimentación	4 a 30 Vdc
No linealidad	$\pm 1/4$ °C
Impedancia de salida baja	0.1Ω para una carga de 1mA

Tabla 2.6 Características eléctricas LM35 – TO-92.

El CI LM35 funciona en el rango de alimentación comprendido entre 4 a 30 Vdc, para nuestro caso son solo necesarios +5Vdc, debido a su baja corriente de alimentación (60uA), se produce un efecto de autocalentamiento reducido, menos de 0.1 °C en situación de aire estacionario.

La baja impedancia de salida, su salida lineal y su precisa calibración inherente hace posible una fácil instalación en un circuito de control.

2.1.6.2 Diagrama de Bloques del Módulo Sensor de Temperatura.

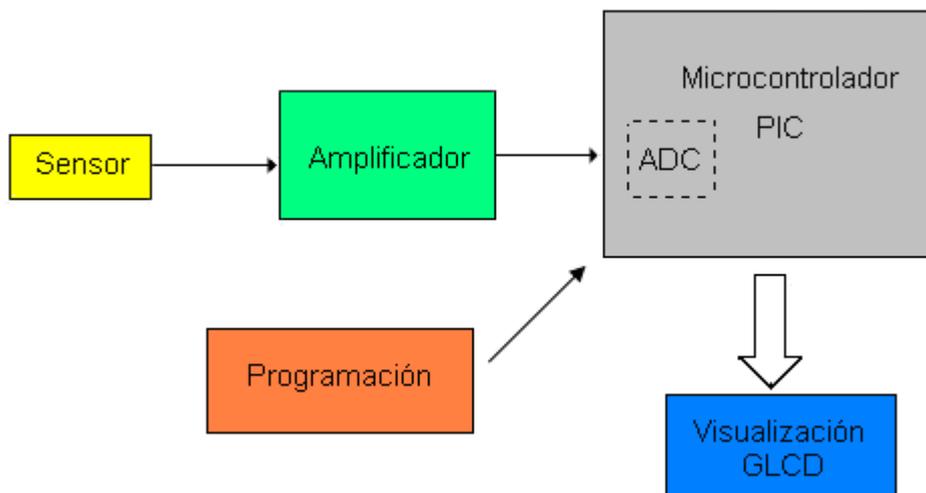


Figura 2.10 Diagrama de Bloques - Módulo Sensor de Temperatura.

Como se puede observar está formado básicamente por tres bloques:

- Sensor de temperatura (Medición – LM35)
- Amplificación (Acondicionamiento de Señal I.C. LM358P).
- Conversión Analógico/Digital (Módulo Conversor Interno A/D de 8 ó 10 bits – Microcontrolador PIC18F4550).

2.1.6.3 Funcionamiento.

El sensor LM35 detecta la temperatura presente en el ambiente y entre su salida (V_{out}) y el nodo de referencia (**GND**) presenta una diferencia de voltaje análogo por cada grado centígrado que varíe, habrá un cambio de 10mV en su salida. Estos niveles de voltaje se amplificarán para que cada 10mV de cambio en el voltaje del sensor, represente un nivel en el convertidor A/D interno del microcontrolador.

Etapa de amplificación.

Para la etapa de amplificación utilizaremos el Circuito Integrado LM358P, que está compuesto internamente por dos amplificadores operacionales que trabajan con +5Vdc para su polarización y funcionan correctamente para estas aplicaciones. A

continuación se muestra la distribución de terminales del circuito integrado LM358P.

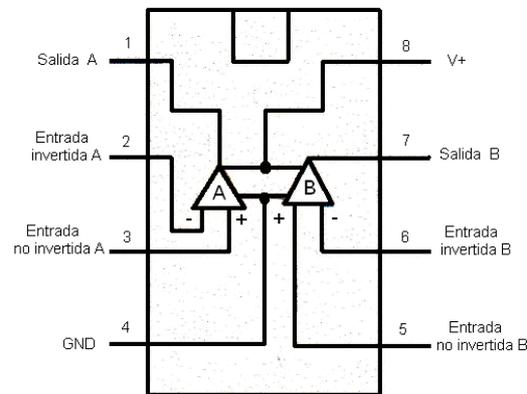


Figura 2.11 Amplificador Operacional LM358P ~ ECG-928M.⁴

El diagrama circuital que se usa para la etapa de amplificación es el siguiente:

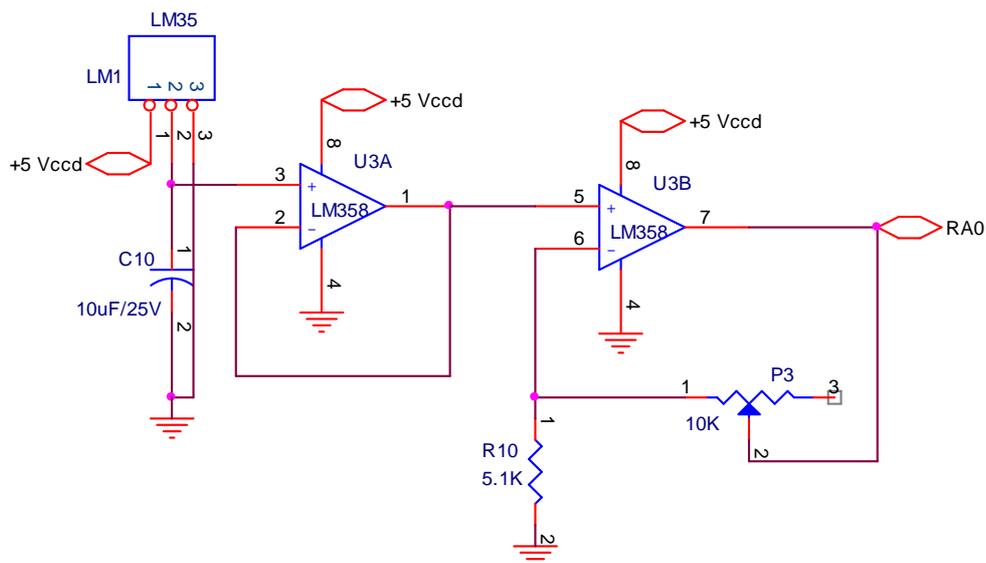


Figura 2.12 Diagrama circuital – Sensor de Temperatura.

La señal de salida del sensor de temperatura, (pin 2 – LM35) se introduce en la etapa de acondicionamiento de la señal, (pin 3 – LM358P) que es la entrada no invertida del amplificador operacional U3A en configuración seguidor de tensión, la salida de esta etapa seguidora (pin 1 – U3A) se introduce en la entrada no invertida (pin 5 – U3B) del otro amplificador operacional U3B del IC LM358P. Esta etapa amplifica el voltaje del sensor de temperatura al doble, las resistencias

⁴ ECG Semiconductors Master Replacement Guide. Page No. 1-176. Fig No. L97

utilizadas en esta configuración son de baja tolerancia para respetar al máximo la señal análoga que entrega del sensor, de tal manera que cada variación de 10mV en la entrada corresponda a la salida del amplificador una señal de 20mV. Estos niveles de voltaje se condicionan para que cada 10mV de variación en el voltaje del sensor represente un nivel en el convertidor A/D del microcontrolador e igual a 1°C. Por último, la señal de salida se introduce a uno de los canales análogos del microcontrolador, concretamente, a la entrada análoga **AN0/RA0** para la conversión.

Conversión Análoga a Digital.

En este apartado se debe programar el módulo ADC del microcontrolador para que realice la conversión de análogo a digital de 8 ó 10 bits, del voltaje presente a la salida del sensor de temperatura (salida del circuito amplificador) y procesarlo para obtener en una posición de la memoria, el valor de la temperatura ambiente.

El convertidor A/D interno del microcontrolador detecta el voltaje análogo presente a la salida del amplificador y lo convierte en niveles digitales, teniendo un rango de 0 a 255 posibles combinaciones en los convertidores de 8 bits y de 0 a 1023 posibles combinaciones en convertidores de 10 bits de resolución. Cuando se produzca una variación de voltaje en el amplificador (la temperatura aumente o disminuya), cambia la combinación a la salida del convertidor A/D. Es decir, que de acuerdo al valor del voltaje analógico que tenga a la entrada el ADC da su equivalente en código digital.

El microcontrolador PIC gestiona la entrega del código digital de salida del ADC interno (8 ó 10 bits), para el siguiente paso realizar las comparaciones de voltaje y enviar la información obtenida en el sensor a dispositivos de visualización como el módulo GLCD.

En conclusión el hombre configura al microcontrolador por medio de un lenguaje de programación para que éste controle dispositivos, realice determinados procesos o acciones (conversión, comparación, transferencia, etc.) es decir darles el trato adecuado a los datos análogos obtenidos de los dispositivos de entrada como sensores, teclados, micrófonos, etc.

2.1.7 MÓDULO TECLADO HEXADECIMAL 4 X 4.

Conectado en el pÓrtico A, utiliza cinco líneas (RA1, RA2, RA3, RA4, RA5) en lugar de las ocho líneas que comÚnmente usamos para el manejo de un teclado matricial 4X4, la razÓn del por que usamos solamente cinco líneas para el control del teclado es por la limitaciÓn de pines (PIC de 40 PDIP) que tenemos.

En estas condiciones el camino ha seguir es utilizar un decodificador de teclado, el cual se adapta perfectamente a nuestra necesidad, convirtiéndose ademÁs en una ventaja potencial de nuestro mÓdulo, en lo que a uso de teclado se refiere como la eliminaciÓn de rebotes y ruido que se genera al pulsar una tecla, asÍ como disminuir el tiempo en que tarda el microcontrolador en reconocer un dato en el pÓrtico al pulsar una tecla, etc.

2.1.7.1 IntroducciÓn.

Teclado Hexadecimal.

Dispositivo de entrada de datos que consta de 16 teclas o pulsadores, dispuestos e interconectados en filas y columnas. Dispone de un conector SIL (Single In Line) macho de 8 pines que corresponden a las cuatro filas y las cuatro columnas de las que dispone. La matriz del teclado es la que se muestra a continuaciÓn:

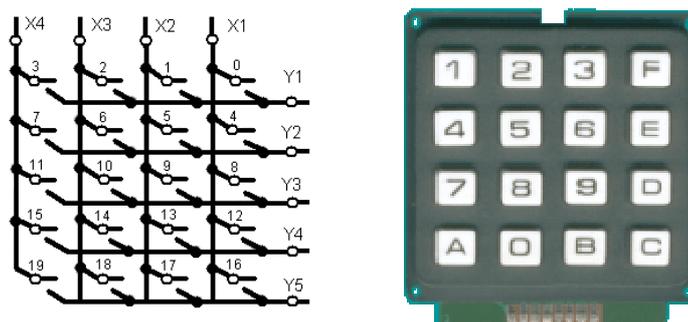


Figura 2.13 Estructura del Teclado matricial 4X4.

Decodificador de Teclado.

El decodificador de teclado, es el I.C. MM74C922 (figura 2.14) es muy eficiente debido a las características y ventajas que tiene, una de las ventajas que tiene

este circuito es muy utilizada en el sistema, esta es la de tener un registro de la última tecla presionada, es decir, si presionamos la tecla numero cinco, el dato de salida se queda guardado y activo en la salida aunque dejemos de presionar el número. Esta ventaja es necesaria debido a que en microcontroladores se trabaja con nibbles, por lo tanto necesitamos que el primer dato se quede guardado y activo para poder enviarlo a la memoria de datos y luego así mandar el siguiente dato a la memoria pero en la misma dirección en que se quedo guardado el nibble anterior.

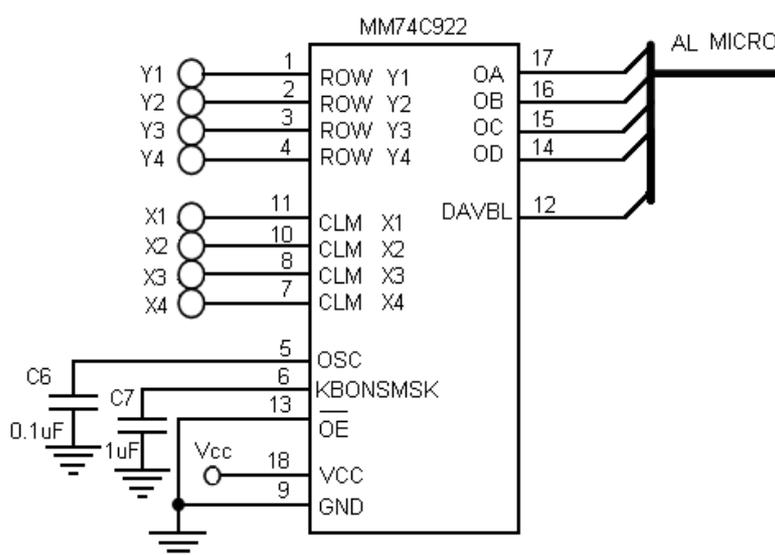


Figura 2.14 Decodificador de Teclado.

Para seleccionar el valor del capacitor que se encargara de eliminar los rebotes es necesario apoyarse en la tabla 2.7 que propone el fabricante.

CAPACITOR (uF)	TIEMPO (s)
0.1	0.001
1	0.01
10	0.1
100	1

Tabla 2.7 Valor del capacitor (Eliminador de rebotes).

El ruido que provoca el cerrar los interruptores no pasa del orden de varios milisegundos, pero también se toma en cuenta, el ruido que produce la fuente al encenderse, aunque también esté en el orden de varios milisegundos, siguiendo la tabla 2.7, un capacitor de $0.1\mu\text{F}$ solo retiene en un rango de 1 ms, ese capacitor no es muy ideal para el circuito, pero el siguiente capacitor de $1\mu\text{F}$ retiene la señal hasta los 10 ms este tiempo ya es apropiado para el circuito, eliminará el ruido o rebotes provocado por los botones del teclado y por el ruido de la fuente, entonces el capacitor puede ser de $C = 1\mu\text{F}$ (1 micro Faradio).

Además se necesita verificar el valor del capacitor de la velocidad de barrido de las filas y columnas del teclado, es decir es el que se encargara de determinar la frecuencia de veces que rastreará un cambio en las terminales que vienen desde el teclado, este capacitor se determina por una tabla de opciones que propone el fabricante y es la siguiente:

CAPACITOR (μF)	FRECUENCIA (Hz)
0.01	5000
0.1	850
1	85
10	-

Tabla 2.8 Capacitor para barrido de Teclado matricial 4x4 (fabricante).

En el sistema no se estarán presionando las teclas muy rápidamente por lo que la velocidad de barrido no se necesita muy alta, si colocáramos una frecuencia alta el sistema podría desestabilizarse debido a la alta frecuencia, pero también es necesario que no sea una frecuencia por debajo de los 100 Hz, el sistema podría carecer de rapidez, según la tabla tenemos que 85 Hz no son ideales y tenemos que 5000 Hz ya entra en una alta frecuencia, entonces se opta por la frecuencia de 850 Hz y el capacitor propuesto es de: $C = 0.1\mu\text{F}$ (0.1 micro Faradios).

En consideración con los aspectos de análisis, el Diagrama circuital del Módulo Teclado Hexadecimal es el siguiente:

2.1.7.2 Diagrama circuital del Módulo Teclado Hexadecimal.

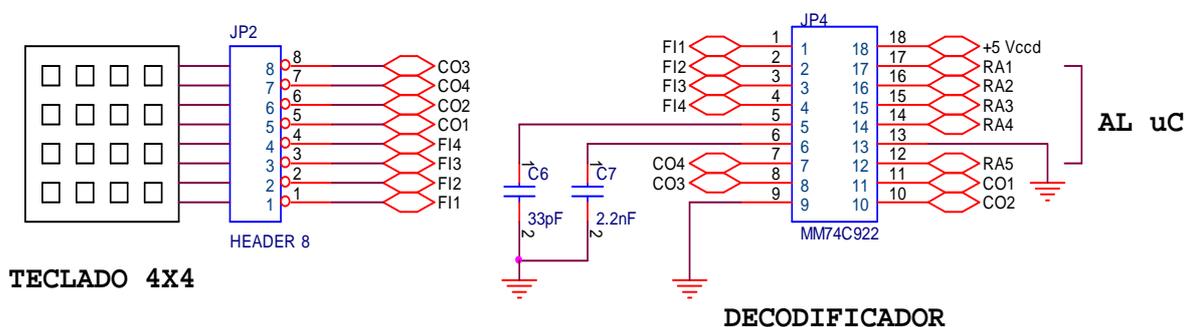


Figura 2.15 Diagrama circuital – Módulo Teclado Hexadecimal.

2.1.7.3 Funcionamiento.

El decodificador de teclado (MM74C922) detecta la tecla que se presione en el teclado y ubicará un código en el puerto A del microcontrolador mediante las líneas de salida del decodificador (OA, OB, OC, OD, DAVBL).

TECLA	SALIDA
0	0000
1	0001
2	0010
3	0011
4	0100
5	0101
6	0110
7	0111
8	1000
9	1001
A	1010
B	1011
C	1100
D	1101
*	1110
#	1111

Tabla 2.9 Código de salida del decodificador MM74C922.

Al presionar una tecla el decodificador detecta cada que se presiona ésta e indica mediante una interrupción que fue presionada una tecla, ésta es el pin 12 del I.C. MM74C922 (**Data Available**) la cual se conecta a una de las cinco líneas del pórtico A que va a controlar el teclado, para que el microcontrolador identifique cuando es presionada una tecla. Las otras cuatro líneas de salida del decodificador de teclado presentaran cada vez que se presione una tecla el código binario correspondiente a la tecla pulsada como se muestra en la tabla 2.9.

2.1.8 MÓDULO DE CONTROL DE MOTORES PASO A PASO.

2.1.8.1 Introducción.

Los motores paso a paso son ideales para la construcción de mecanismos en donde se requieren movimientos muy precisos. La característica principal de estos motores es el hecho de poder moverlos un paso a la vez por cada pulso que se le aplique. Este paso puede variar desde 90° hasta pequeños movimientos de tan solo 1.8° , es decir, que se necesitarán 4 pasos en el primer caso (90°) y 200 para el segundo caso (1.8°), para completar un giro completo de 360° .

Estos motores poseen la habilidad de poder quedar enclavados en una posición o bien totalmente libres. Si una o más de sus bobinas están energizadas, el motor estará enclavado en la posición correspondiente y por el contrario quedará completamente libre si no circula corriente por ninguna de sus bobinas.

En este capítulo trataremos solamente los motores Paso a Paso del tipo de imán permanente.

2.1.8.2 Principio de funcionamiento de motores Paso a paso.

Básicamente estos motores están constituidos normalmente por un rotor sobre el que van aplicados distintos imanes permanentes y por un cierto número de bobinas excitadoras bobinadas en su estator. Las bobinas son parte del estator y el rotor es un imán permanente. Toda la conmutación (o excitación de las bobinas) deber ser externamente manejada por un controlador.

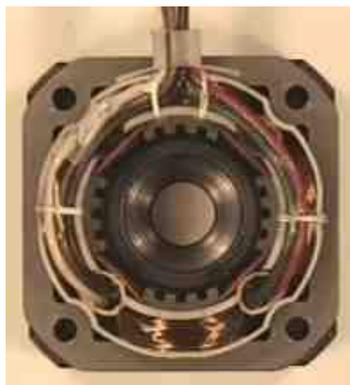


Figura 2.16 Imagen de un estator de 4 bobinas.

Existen dos tipos de motores paso a paso de imán permanente:

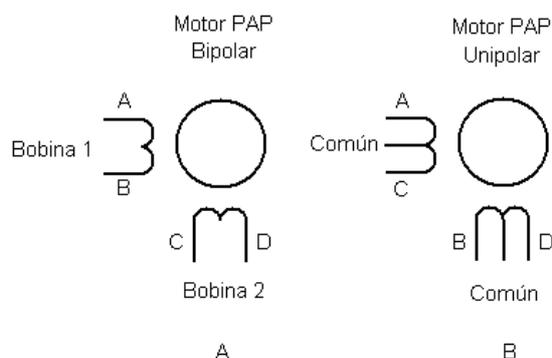


Figura 2.17 Tipos de Motores Paso a Paso.

Bipolar: Estos tienen generalmente cuatro cables de salida (figura 2.17A). Necesitan ciertos trucos para ser controlados, debido a que requieren del cambio de dirección del flujo de corriente a través de las bobinas en la secuencia apropiada para realizar un movimiento.

Unipolar: Estos motores suelen tener seis ó cinco cables de salida, dependiendo de su conexión interna (figura 2.17B). Este tipo se caracteriza por ser más simple de controlar.

Secuencias para manejar motores paso a paso Unipolares

Existen tres secuencias posibles para este tipo de motores, las cuales se detallan a continuación. Todas las secuencias comienzan nuevamente por el paso uno

una vez alcanzado el paso final (cuatro u ocho). Para revertir el sentido de giro, simplemente se deben ejecutar las secuencias en modo inverso.

Secuencia Normal: Esta es la secuencia más usada y la que generalmente recomienda el fabricante. Con esta secuencia el motor avanza un paso por vez y debido a que siempre hay al menos dos bobinas activadas, se obtiene un alto torque de paso y de retención.

Secuencia del tipo wave drive: En esta secuencia se activa solo una bobina a la vez. En algunos motores esto brinda un funcionamiento más suave. La contrapartida es que al estar solo una bobina activada, el torque de paso y retención es menor.

Secuencia del tipo medio paso: En esta secuencia se activan las bobinas de tal forma de brindar un movimiento igual a la mitad del paso real. Para ello se activan primero dos bobinas y luego solo una y así sucesivamente. Como vemos en la tabla la secuencia completa consta de ocho movimientos en lugar de cuatro ya que así se fusionan los dos tipos de secuencia anterior.

PASO	BOBINA A	BOBINA B	BOBINA C	BOBINA D	RECREACIÓN
1	ON 1	OFF 0	OFF 0	OFF 0	
2	ON 1	ON 1	OFF 0	OFF 0	
3	OFF 0	ON 1	OFF 0	OFF 0	
4	OFF 0	ON 1	ON 1	OFF 0	

5	OFF 0	OFF 0	ON 1	OFF 0	
6	OFF 0	OFF 0	ON 1	ON 1	
7	OFF 0	OFF 0	OFF 0	ON 1	
8	ON 1	OFF 0	OFF 0	ON 1	

Tabla 2.10 Secuencia Tipo medio paso.

Debido a que los motores paso a paso son dispositivos mecánicos y como tal deben vencer ciertas inercias, el tiempo de duración y la frecuencia de los pulsos son aspectos muy importantes. Si la frecuencia de pulsos es muy elevada, el motor puede reaccionar en alguna de las siguientes formas:

- Puede que no realice ningún movimiento en absoluto.
- Puede comenzar a vibrar pero sin llegar a girar.
- Puede girar erráticamente.
- Puede llegar a girar en sentido opuesto.

Para obtener un arranque suave y preciso, es recomendable comenzar con una frecuencia de pulso baja y gradualmente ir aumentándola hasta la velocidad deseada sin superar la máxima tolerada. El giro en reversa debería también ser realizado previamente bajando la velocidad de giro y luego cambiar el sentido de rotación.

2.1.8.3 Diagrama circuital.

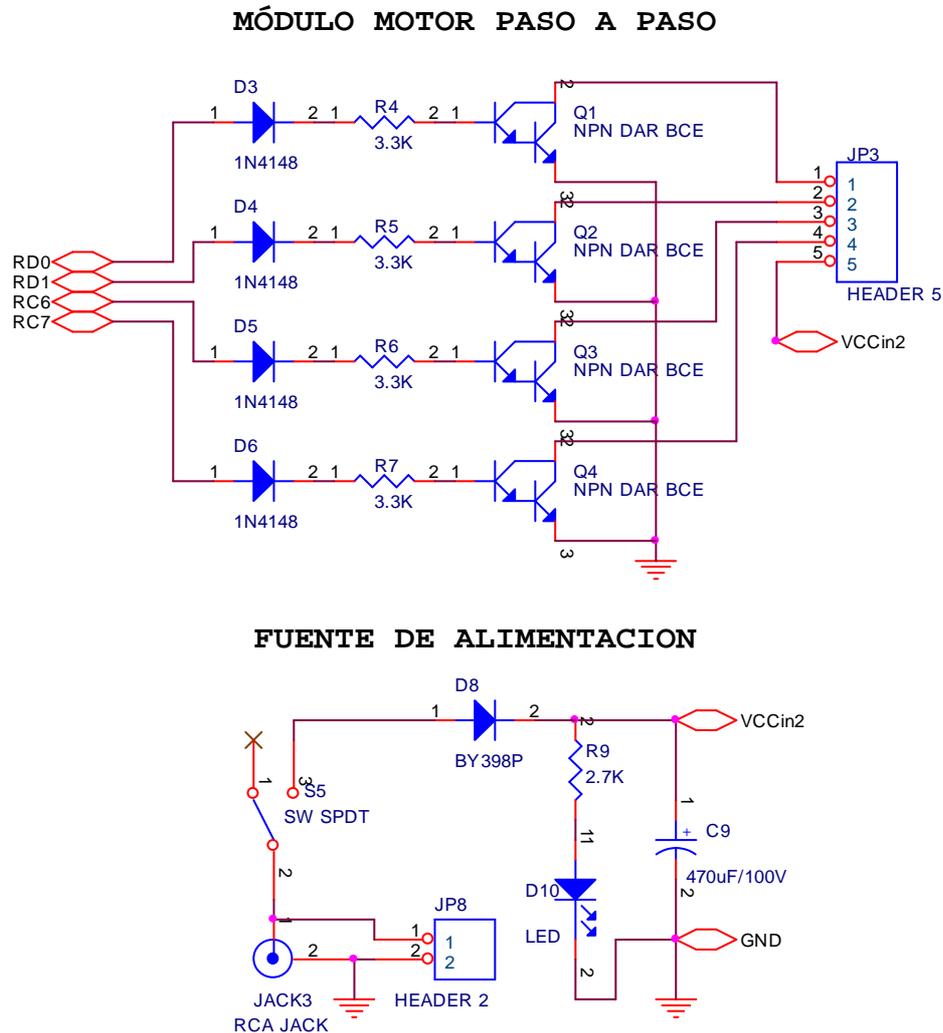


Figura 2.18 Diagrama circuital – Módulo Control de Motor Paso a Paso Unipolar.

El Módulo de control es exclusivo para motores PAP unipolares. Utiliza las líneas RD0, RD1 del pòrtico D y dos líneas RC6, RC7 del pòrtico C del microcontrolador, en las cuales se ubica la secuencia de control correspondiente para el manejo del motor haciendo que gire en sentido horario, antihorario o reproduzca los giros que el programador desee realizar.

Debido a que el microcontrolador PIC18F4550 responde a especificaciones de la familia TTL (Transistor-Transistor-Logic) es capaz de suministrar 25 mA a 5 V como máximo en cada una de sus líneas, el nivel de voltaje y corriente son relativamente pequeños para el control de motores paso a paso, es por tal motivo que se ha implementado un circuito de fuerza para superar esta limitación, en

donde se destacan los cuatro transistores de potencia tipo Darlington (TIP 122) que nos permiten manejar cargas de hasta 500mA (5A) como podemos observar en la tabla 2.11 de datos del dispositivo. Para proteger al microcontrolador de posibles picos de corriente inversa que se generan al momento de arrancar el motor, se han añadido cuatro diodos (1N4148) en serie con la resistencia de base (3.3K Ω) de cada transistor de potencia (figura 2.18).

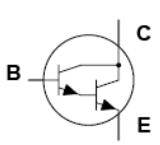
TIP122	CARACTERÍSTICAS BÁSICAS				Distribución de terminales	
	Símbolo	Mínimo	Típico	Máximo		
Transistor dárlington NPN de mediana potencia (65W) diseñado para funcionar como interruptor	I_c			5A		
	V_{ce0}			100v		
	V_{be}		2.5v			
	β	20		200		
	$V_{ce(sat)}$	2v		4v		

Tabla 2.11 Características - Transistor NPN Darlington (TIP122).

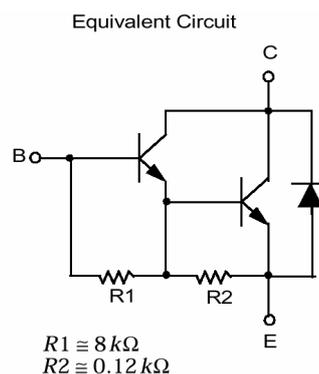


Figura 2.19 Circuito Equivalente al Transistor NPN Darlington (TIP122).

Como se puede observar, el transistor Darlington tiene una ganancia mucho mayor que un transistor normal, pues se multiplican las ganancias de los dos transistores. Si las ganancias de ambos transistores fuesen, por ejemplo, igual a 100, el conjunto tendría un ganancia de 10000. Los transistores Darlington se emplean en circuitos donde se necesita controlar grandes cargas con corrientes muy pequeñas. Por último, queda decir que la caída de tensión entre la base **B** y el emisor V_{BE} en este tipo de transistores es el doble que en los ordinarios (aproximadamente unos 1.4 v), a la vez el colector de cada transistor TIP122 se conecta a una bornera, para su conexión con cada cable de salida del motor PAP en uso.

Como se mencionó en el apartado que tiene que ver con la fuente de alimentación de la tarjeta electrónica y polarización del microcontrolador, el módulo de Motor Paso a Paso posee fuente de voltaje DC externa que también será suministrada por la fuente que alimenta a la tarjeta electrónica (Fuente Switchig). Para esto se ha dotado de un circuito de encendido/apagado (ON/OFF) que protege y a la vez permite detectar la presencia del voltaje DC desde la fuente externa, se ha dotado además de un conector (bornera) adicional que nos permite la alternativa de trabajar con una fuente externa para los casos en que se desee controlar motores paso a paso de mayor consumo.

2.1.9 MÓDULO – CONTROL DE MOTORES DC.

2.1.9.1 Introducción.

Una de las mejores opciones para controlar motores de corriente continua es hacer uso del **Driver L293B** (Doble puente “H” independiente), al cual no hay que conectarle más que el motor, polarizarlo e introducir las señales de control que vienen del microcontrolador.

El chip L293B es un conductor de 4 canales capaz de proporcionar una corriente de salida de hasta 1A por canal. Cada canal es controlado por señales de entrada compatibles TTL y cada pareja de canales dispone de una señal de habilitación que desconecta las salidas de los mismos. Dispone de una patilla para la alimentación de las cargas que se están controlando, de forma que dicha alimentación es totalmente independiente de la lógica de control.

Con este circuito integrado se puede controlar el sentido de giro de hasta dos motores simultáneamente.

2.1.9.2 Driver L293B.

Características.

- Intensidad por canal = 1A
- Intensidad pico por canal = 2 A
- Protección contra sobre-temperaturas.

- Sin protección interna contra sobre-tensiones producidas por cargas inductivas.
- Alimentación independiente de cargas y lógica de control.
- Alta inmunidad al ruido.
- Tensión de alimentación de las cargas ≤ 36 Vdc.
- Tensión de alimentación del chip = de 4.5 a 36 Vdc.
- El control de los canales (driver's) es compatible con señales TTL es decir con 5 voltios (estamos hablando de señales lógicas).
- Cada uno de los 4 canales (driver's) puede ser activado de forma independiente (por su terminal de entrada), o habilitado de dos en dos con un solo terminal (enable).

PIN	NOMBRE	DESCRIPCIÓN
1	EN1	Habilitación de los canales 1 y 2
2	IN1	Entrada del canal 1
3	OUT1	Salida del canal 1
4	GND	Tierra de alimentación
5	GND	Tierra de alimentación
6	OUT2	Salida del canal 2
7	IN2	Entrada del canal 2
8	V _S	Alimentación de las cargas
9	EN2	Habilitación de los canales 3 y 4
10	IN3	Entrada del canal 3
11	OUT3	Salida del canal 3
12	GND	Tierra de alimentación
13	GND	Tierra de alimentación
14	OUT4	Salida del canal 4
15	IN4	Entrada del canal 4
16	V _{SS}	Tensión de alimentación de la lógica

Tabla 2.12 Descripción de pines del Driver L293B.

En la figura 2.20 se muestra el encapsulado, su distribución de pines y descripción:

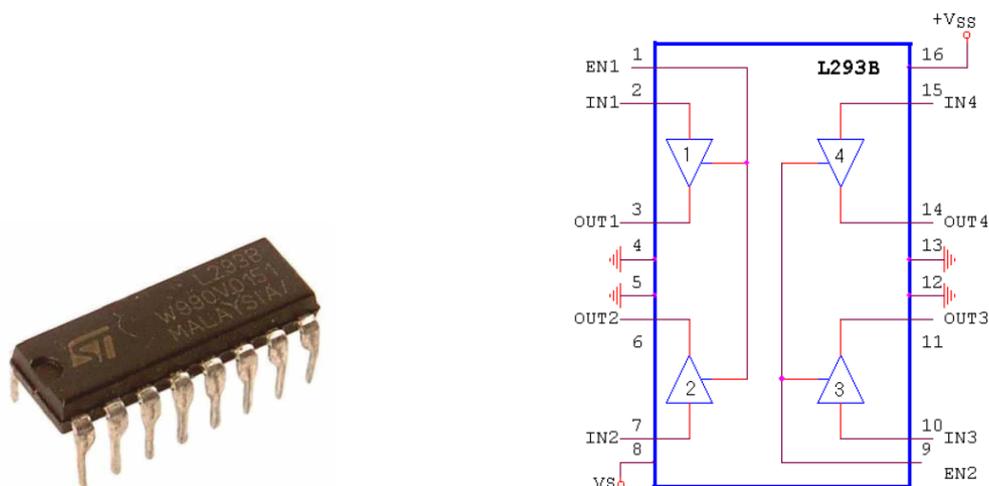


Figura 2.20 Encapsulado C.I. DRIVER L293B.

En la figura 2.20, se muestra el diagrama de bloques del L293B. La señal de control **EN1** activa la pareja de canales formada por los drivers 1 y 2. La señal **EN2** activa la pareja de drivers 3 y 4. Las salidas **OUT_N** se asocian con las correspondientes **IN_N**. Las señales de salida son amplificadas respecto a las de entrada tanto en tensión como en corriente (máximo 1A por canal).

La tabla de funcionamiento para cada uno de los drivers es la siguiente:

H = Nivel Alto, 1L.
L = Nivel Bajo, 0L.
Z= Alta Impedancia.

EN₁₋₂	IN_N	OUT_N
H	H	H
H	L	L
L	H	Z
L	L	Z

Tabla 2.13 Funcionamiento Driver L293B.

En la tabla de verdad la entrada **EN₁₋₂** habilita dos de los canales de este integrado, cuando se encuentra un nivel alto (**H**) en esta entrada, de tal modo que la salida **OUT_N** tomará el valor de la entrada **IN_N**. Mientras que **OUT_N** quedará en alta impedancia (**Z**) si el terminal **EN₁₋₂** se encuentra en un nivel bajo (**L**), es decir que en este caso ya no tiene importancia el valor de **IN_N** y por lo tanto **OUT_N** queda totalmente libre.

Por último, aclarar que V_S y V_{SS} son los pines de alimentación, V_S para los 4 driver's que según el fabricante puede tomar valores desde $V_{SS} = 5V_{dc}$ hasta $36V_{dc}$, y V_{SS} es el terminal de alimentación TTL, es decir $5V_{dc}$.

SÍMBOLO	PARÁMETRO	V. TÍPICO	UND
V_S	Tensión de alimentación para las cargas	24	Vdc
V_{SS}	Tensión de alimentación de la lógica	5	Vdc
V_{IN}	Tensión de entrada	7	Vdc
V_{EN}	Tensión de habilitación	7	Vdc
I_{OUT}	Intensidad pico de salida	2	A
P_{TOT}	Potencia total de disipación	5	W
T°	Temperatura	-40 a +150	$^{\circ}C$

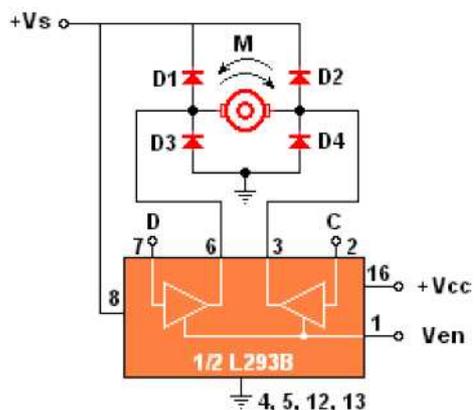
Tabla 2.14 Rangos típicos de funcionamiento del Driver L293B.

2.1.9.3 Aplicación.

En este apartado se muestra la configuración de conexión de motores al L293B.

Control del giro de un motor en los dos sentidos

Para el control de giro de un motor DC necesitamos dos de los cuatro drivers que el chip posee (EN1 o EN2), el circuito de la figura permite controlar el doble sentido de giro del motor. Cuando la entrada **C** está a nivel bajo y la **D** a nivel alto, el motor gira hacia la izquierda. Cambiando la entrada **C** a nivel alto y la **D** a nivel bajo, se cambia el sentido de giro del motor hacia la derecha.



Ven	C	D	Función del motor
H	H	L	Giro a Derecha
H	L	H	Giro a Izquierda
H	C=D		Detención
L	X	X	Motor Libre

Figura 2.21 Control de un motor en ambos sentidos.

Para proteger el circuito contra posibles picos de corriente inversa al arrancar el motor, se ha dispuesto unos diodos (1N4007) de tal manera como se muestra en la figura 2.21. La tabla anterior muestra claramente como controlar un motor, en un sentido o en otro, detenerlo ó dejarlo libre es decir que al ubicar cualquier señal en los terminales de control C y D no tendrá efecto alguno sobre el motor, este quedará liberado totalmente.

Con este circuito integrado se puede controlar el sentido de giro de hasta dos motores simultáneamente para ello se requiere de los 4 driver's del integrado, la misma forma de conexión como se observa en la figura 2.21 para el otro motor DC que se desee controlar simultáneamente. Uno de los aspectos importantes que se ha tomado en consideración es el control de la velocidad de los Motores DC lo cual lo podemos manejar a través de una unidad funcional que posee el microcontrolador en su interior, el Módulo ECCP en modo PWM que nos permite generar una señal PWM de ciclo de trabajo y frecuencia programable.

Control de Velocidad de motores por PWM.

Se puede regular la velocidad de un motor de corriente continua, por el método, denominado PWM (modulación por ancho de pulso) que consiste en alimentar al motor por impulsos eléctricos de modo que cuando estos impulsos son anchos, el motor interpreta estar alimentado con una tensión elevada. Si por el contrario, los impulsos son estrechos, el motor girará lentamente, sin que se produzca una reducción de voltaje considerable del par (bobina).

Modulación de Ancho de Pulso

Conocido como PWM, básicamente consiste en enviar a los terminales de control un tren de pulsos, los cuales varían en mayor o menor medida el tiempo en que se mantiene el nivel alto de esos pulsos, manteniendo la frecuencia constante.

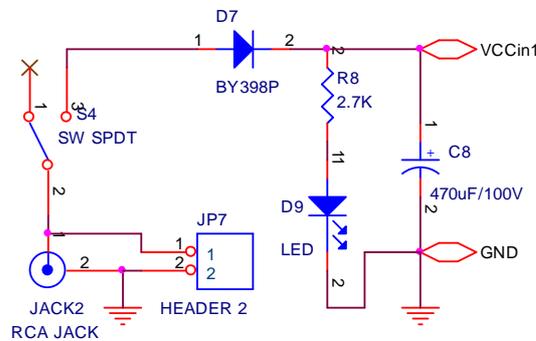


Figura 2.22 Tren de pulsos PWM.

Con esta forma de control, la cantidad de corriente sigue siendo la misma, la tensión no varía y en consecuencia el torque del motor se mantiene, que es justamente lo que estamos buscando.

El rendimiento de este sistema es excelente ya que nosotros podemos generar señales de modulación de ancho de pulso configurando a nuestro microcontrolador PIC (18F4550) en modo PWM que es uno de los muchos modos en que puede trabajar nuestro microcontrolador.

2.1.9.4 Diagrama circuital del Módulo de Control de Motores DC.



FUENTE DE ALIMENTACIÓN

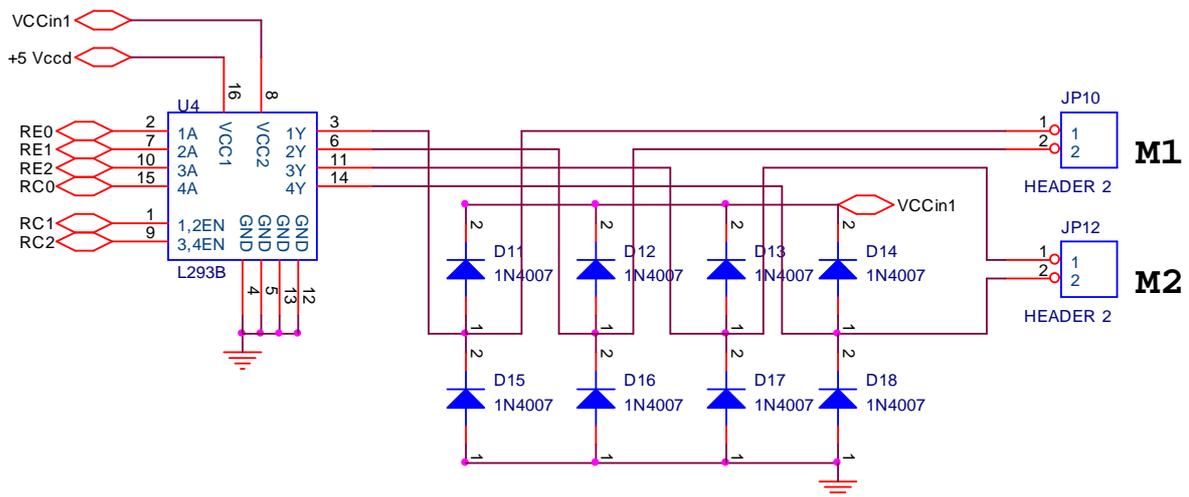


Figura 2.23 Diagrama circuital – Modulo Motor DC

Conectado en las tres líneas bidireccionales del pòrtico C (RC0, RC1, RC2) en conjunto con las únicas tres líneas maniobrables del pòrtico E (RE0, RE1, RE2) configurando el bus de seis líneas para el control de motores DC. Por el bajo nivel de las señales de salida en las líneas de los pòrticos (25 mA c/u a 5Vdc) del

microcontrolador, necesitamos utilizar un interfaz para controlar estos motores, para ello hacemos uso de el Driver L293B con el que conseguimos trabajar con una corriente por canal de 1A, suficiente para aspectos metodológicos sin mencionar que además contamos con una fuente de alimentación externa para trabajar con que este módulo. El voltaje DC de la fuente externa se conduce al módulo para el control de los motores DC a través de un conector (jack hembra) y un circuito de ON/OFF que permite visualizar la presencia del voltaje por medio de un diodo emisor de luz (LED) (figura 2.23). Se ha dotado además de borneras adicionales que nos permiten conectar los terminales de una fuente externa para casos en que se desee controlar motores DC de mayor consumo.

Para el caso en que nosotros deseemos controlar la velocidad del motor utilizaremos los pines 16 y 17 (CCP2-CCP1 respectivamente) del microcontrolador PIC18F4550 previo a una determinada configuración del microcontrolador por el usuario (programador), para generar señales de modulación de ancho de pulso PWM (Pulse Weight Modulation) que se transmitirán a los pines habilitadores EN1 y EN2 (pines 1-9 respectivamente) del **L293B** para variar la velocidad del motor DC en conjunto con las señales que se ubicaran en las entradas de los driver's para el control de giro del motor. Las cuatro salidas del Driver L293B se conectan a borneras en las que se conectan los cables de salida del Motor DC a controlar.

2.1.10 MÓDULO DE COMUNICACIONES USB.

El microcontrolador PIC18F4550 está dotado de periféricos de comunicación avanzados como CAN, USB, RF, Ethernet, etc.

2.1.10.1 Introducción.

El dispositivo PIC18F4550 abarca full- low- speed, compatible con USB 2.0 y con el equipo de interfaz serial (SIE) lo que permite al microcontrolador una comunicación rápida entre el dispositivo USB y el PC. El SIE (Serial Interface Engine) puede unirse directamente al USB, utilizando el transmisor interno, en conjunto con el regulador interno de 3.3V que es una fuente de poder disponible

del transmisor interno para aplicaciones que requieran de Voltaje (V_{USB}), y que es el método utilizado en nuestro módulo para las comunicaciones con el computador personal (PC). Además el SIE puede conectarse con el USB por medio de un transmisor externo como se muestra en la figura 2.24. Algunos rasgos especiales del hardware han sido incluidos para mejorar su rendimiento como:

- Un puerto dual para dispositivos con memoria de datos (USB RAM), además viene diseñado para compartir el acceso directo a la memoria entre el microcontrolador y el SIE.
- También se proporcionan descriptores que permiten a los usuarios programar endpoints del espacio de memoria USB RAM.
- Operando con el pòrtico paralelo se garantiza el traslado permanente de volúmenes grandes de datos, así como de datos isócronos hacia los buffers de salida de la memoria.

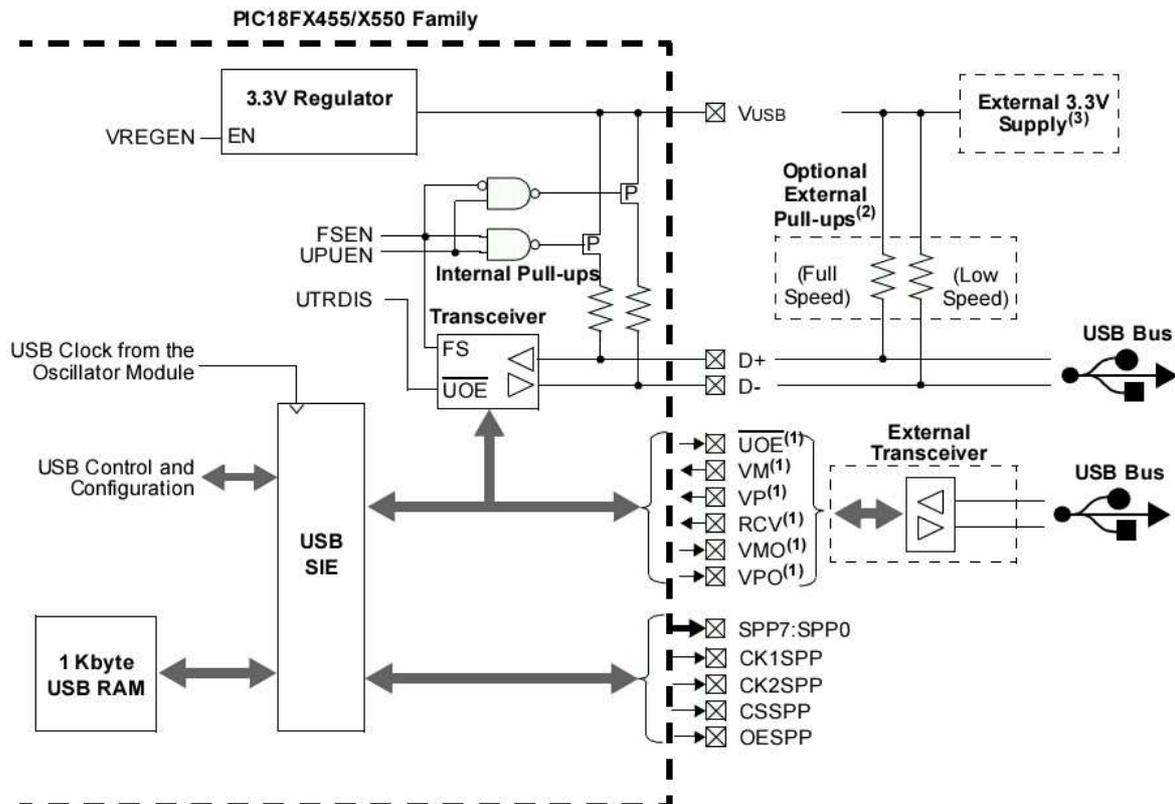


Figura 2.24 Opción de configuración del Periférico USB.

2.1.10.2 Diagrama circuital del Interfaz de Comunicaciones USB.

El interfaz USB es bien conocido en la actualidad. Hay tres velocidades de transmisión en las que puede trabajar el estándar USB 2.0 (Ver capítulo I).

- Alta velocidad: 480 Mbps.
- Velocidad completa: 12 Mbps.
- Baja velocidad: 1.5 Mbps.

El módulo de comunicaciones USB requiere la operación del microcontrolador a 48MHz con un cristal externo de 20MHz el cual está conectado en paralelo a los pines 13 (OSC1) y 14 (OSC2) de nuestro microcontrolador como se muestra en la figura 2.25.

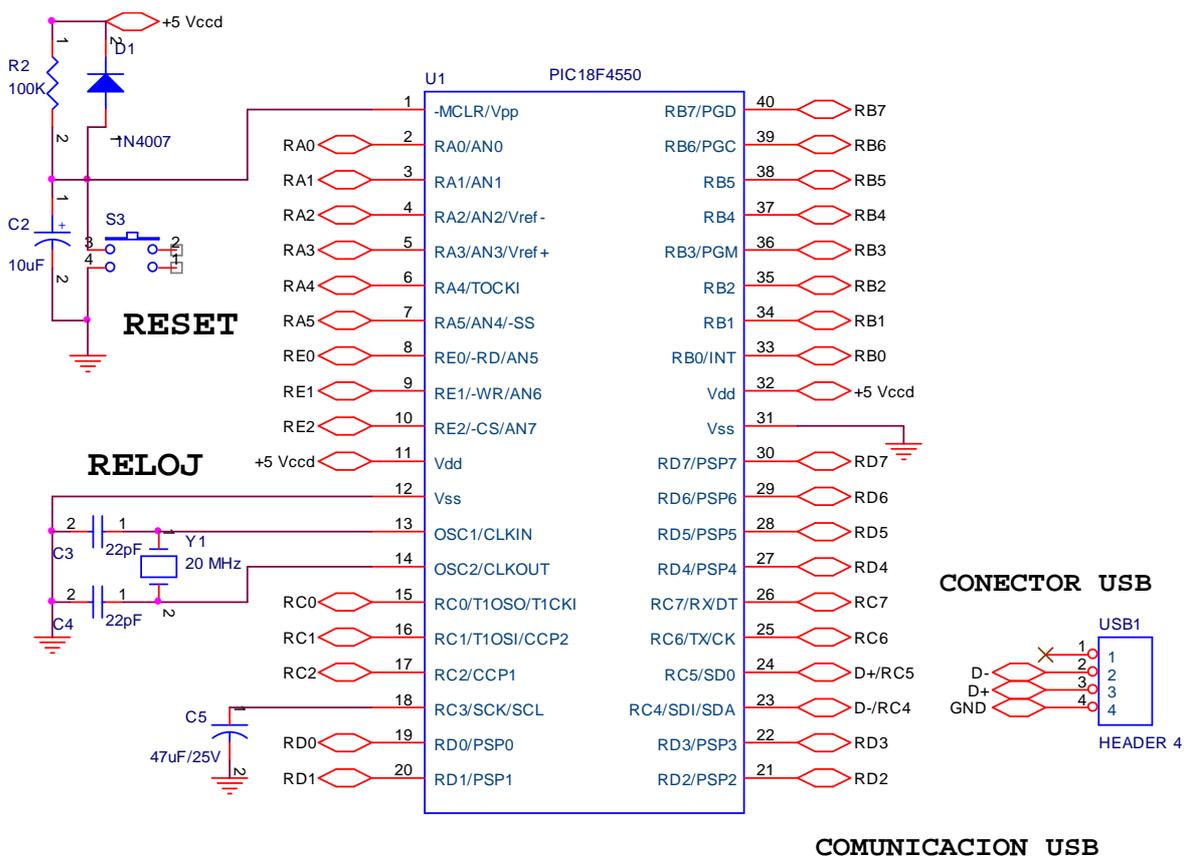


Figura 2.25 Diagrama circuital del Interfaz USB.

Conexión del Microcontrolador con el interfaz USB.

El microcontrolador PIC18F4550 se conecta con el interfaz USB solamente a través de cuatro cables.

PIN μ C	CONEXIÓN	USB SOCKET-PIN
	V_{CC} (opcional)	1
23	D-	2
24	D+	3
12-31	V_{SS} – Tierra	4

Tabla 2.15 Conexión del microcontrolador al USB Socket.



Figura 2.26 USB Socket.

Como se puede apreciar en la tabla 2.15 de conexión del microcontrolador PIC con el conector de bajada USB (terminal tipo B), no se toma en cuenta el voltaje V_{dc} de polarización del microcontrolador como fuente de voltaje para el módulo USB, que como se indica es opcional, ya que el utilizar ó no utilizar esta fuente, no repercute en el normal funcionamiento de las comunicaciones USB entre nuestra tarjeta y el PC. Más bien obtenemos independencia de fuente en la tarjeta electrónica como en el computador personal, protegiendo así al puerto USB del computador de posibles sobrecargas o cortocircuitos de los que no esta exenta nuestra tarjeta de entrenamiento.

2.1.10.3 Cables y conectores USB.

El cable del bus USB es de cuatro hilos, y comprende líneas de señal (datos) y alimentación, con lo que las funciones pueden utilizar un único cable.

Existen dos tipos de cable: apantallado y sin apantallar. En el primer caso el par de hilos de señal es trenzado; los de tierra y alimentación son rectos, y la cubierta de protección (pantalla) solo puede conectarse a tierra en el dispositivo anfitrión (computador personal).

En el cable sin apantallar todos los hilos son rectos. Las conexiones a 15 Mbps y superiores exigen cable apantallado. En el que utilizan diámetros estándar para los hilos de alimentación del bus. Para cada sección se autoriza una longitud máxima del segmento.

En la tabla de la izquierda se muestran estas distancias, a la derecha se muestra la disposición de pines y colores de identificación⁵.

AWG	mm Ø	long. Máx.
28	0.321	0.81 m
26	0.405	1.31 m
24	0.511	2.08 m
22	0.644	3.33 m
20	0.812	5.00 m

Tabla 2.16 Diámetro del cable USB

Pin	Nombre	Descripción	Color
1	VBUS	5 Vdc	rojo
2	D-	Dato -	azul
3	D+	Dato +	amarillo
4	GND	Tierra	verde

Tabla 2.17 Colores de identificación USB

Se usan dos tipos de conectores “A” y “B”. Ambos son polarizados (solo pueden insertarse en una posición) y utilizan sistemas de presión para sujetarse. Los de tipo **A** utilizan la hembra en el sistema anfitrión (PC), y suelen usarse en dispositivos en los que la conexión es permanente (por ejemplo, ratones y teclados). Los de tipo **B** utilizan la hembra en el dispositivo USB (Módulo Didáctico), y se utilizan en sistemas móviles (por ejemplo, cámaras fotográficas o altavoces). En general podemos afirmar que la hembra de los conectores **A** están en el lado del host (PC) o de los concentradores (hubs), mientras las de tipo **B** están del lado de los periféricos. Las conexiones USB soportan una distancia máxima de 5 metros, aunque con dispositivos amplificadores se puede superar esta distancia.



Figura 2.27 Tipos de conectores USB.

⁵ Dado que los cables y conectores transmiten tensión a los periféricos, es muy importantes, sobre todo en las conexiones internas (a placa base mediante pines) seguir fielmente las indicaciones de conexión suministradas por fabricante de la placa base, ya que un USB mal conectado puede causar graves averías, tanto en el periférico conectado como en la propia placa base.

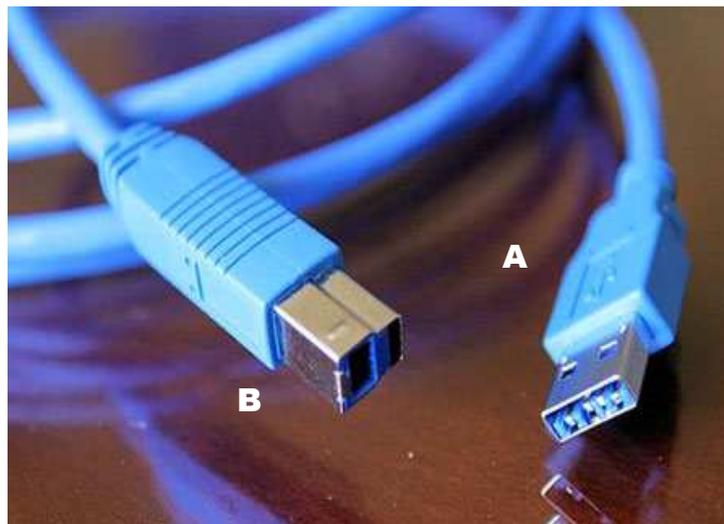


Figura 2.28 Terminales USB Estándar.

2.1.10.4 Software controlador de las comunicaciones USB.

Para gestionar las comunicaciones del bus USB hemos utilizado el firmware que nos proporciona el propio fabricante del microcontrolador (Microchip), integrándolo en nuestro software **“PROTON IDE – MECANIQUE”**. Esta librería está implementada siguiendo la especificación USB 2.0. Para gestionar las comunicaciones USB, lo que hemos tenido que hacer es incluir una llamada a la función principal del Driver USB cada vez que el bus genera una interrupción, para darle servicio a ésta. Lo que nosotros vamos a utilizar son las funciones que nos proporciona el Driver para enviar datos o leer los datos recibidos. El firmware es configurable a cualquier tipo de transferencia USB, ya sea de tipo bulk, por interrupción o síncrona. Nosotros vamos a configurar el sistema para que funcione con transferencias del tipo interrupción, ya que los datos que nos llegan desde el PC no los recibimos de forma continuada, sino que nos llegan en forma de ráfagas.

Formato del protocolo de comunicación.

Nuestro sistema va a tener que comunicarse con un computador, así que es necesario que este sepa cuando estamos listos para recibir datos y cuando no. Para ello, vamos a implementar tanto en la aplicación que envía los datos a través del USB, como en nuestro periférico, una serie de comandos.

Esto nos va a permitir atender diferentes peticiones, que son:

- Leer versión del software del periférico.
- Testear la memoria RAM. Se escribe y se lee un valor en una posición aleatoria de la RAM.
- Comienzo de recepción de datos.
- Prepara al dispositivo para recibir datos.
- Fin de envío de datos.

Proceso de comunicaciones USB.

Vamos a explicar cual es el proceso de transferencia de datos, envío, recepción cada vez que se recibe una petición por parte del Host (PC).

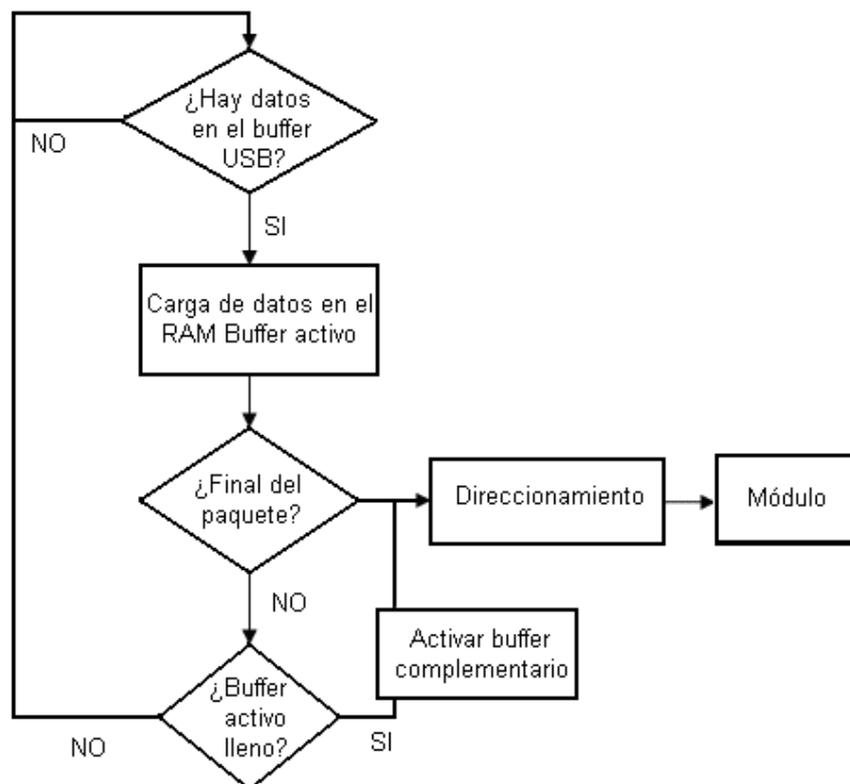


Figura 2.29 Diagrama de Flujo – Petición del PC.

Este es el diagrama de flujo de la gestión de los datos en nuestro sistema. Lo que hacemos es cada vez que nos llega una interrupción por parte del bus USB,

comprobamos si este ha colocado datos en el buffer USB (allí donde el Driver del bus deposita los datos cuando acaba la transacción).

Si hay datos en el buffer, estos los copiamos en uno de nuestros 2 buffers externos ubicados en la memoria RAM externa. Cuando recibimos el comando de fin de recepción de datos comienza el direccionamiento de los paquetes (información) y el envío de los datos a cada pÓrtico del microcontrolador para el control del mÓdulo respectivo.

Si los datos recibidos consiguen llenar el primer buffer, lo que hacemos es activar el segundo buffer para ir guardando allí los datos mientras se van almacenando los datos del primer buffer y se van enviando, para así no hacer esperar al PC en el envÍo de datos.

Una vez enviados los datos del primer buffer, este podrÍa ir llenÁndose de nuevo con datos nuevos, y mientras este se va llenando, los datos del segundo buffer se continúan enviando. Y así sucesivamente hasta que el Host (PC) comunique que ha terminado la transferencia de datos.

CAPITULO 3.

3.1 PRUEBAS DE DISEÑO.

Una vez elaborado el diagrama lógico en papel nos disponemos a adquirir los elementos, terminales, dispositivos electrónicos que conforman el módulo de entrenamiento (Anexo 3). Posteriormente ensamblamos el circuito en el protoboard para realizar las pruebas de diseño.

3.1.1 MÓDULO DIDÁCTICO ENSAMBLADO EN PROTOBOARD.

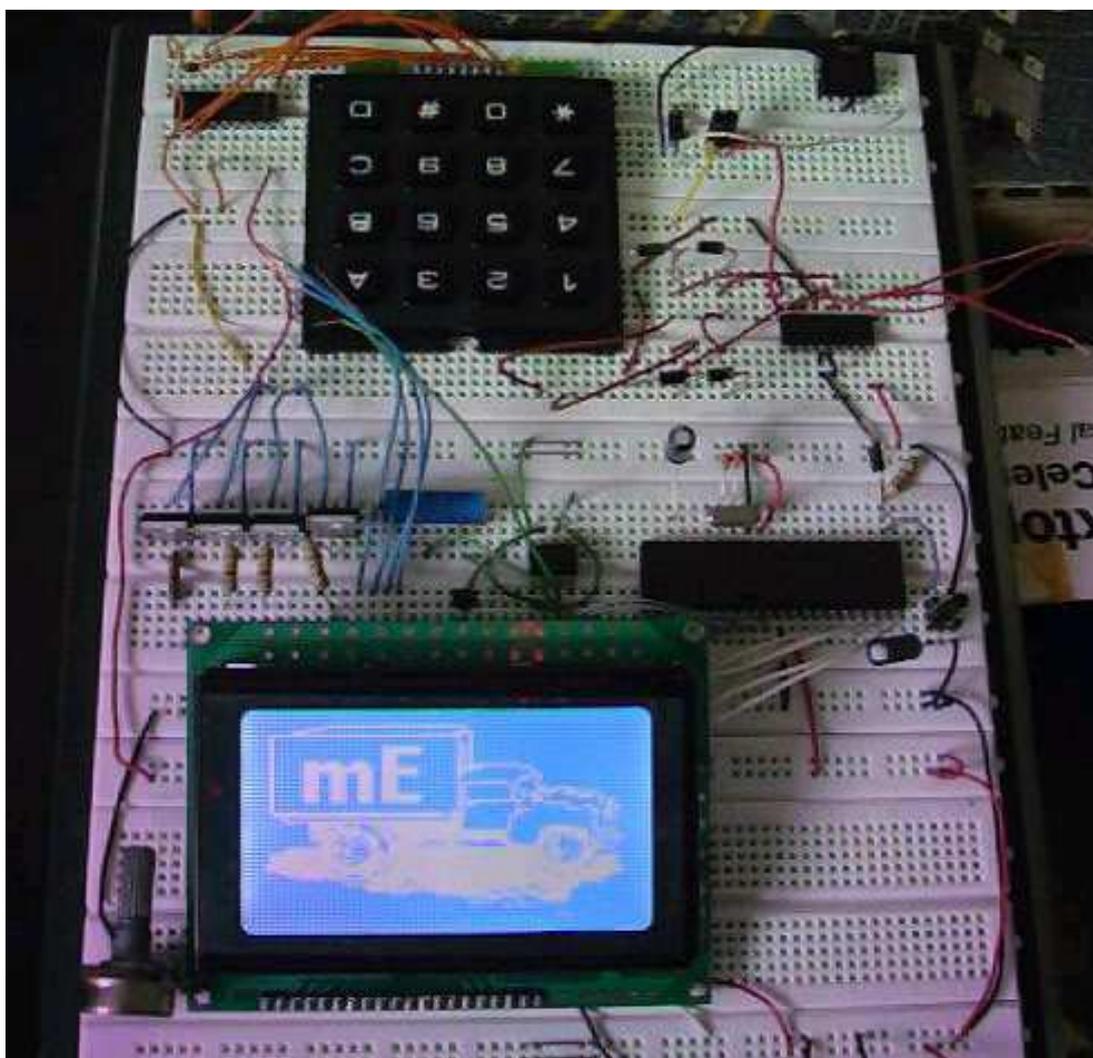


Figura 3.1 Módulo Didáctico ensamblado en el Protoboard.

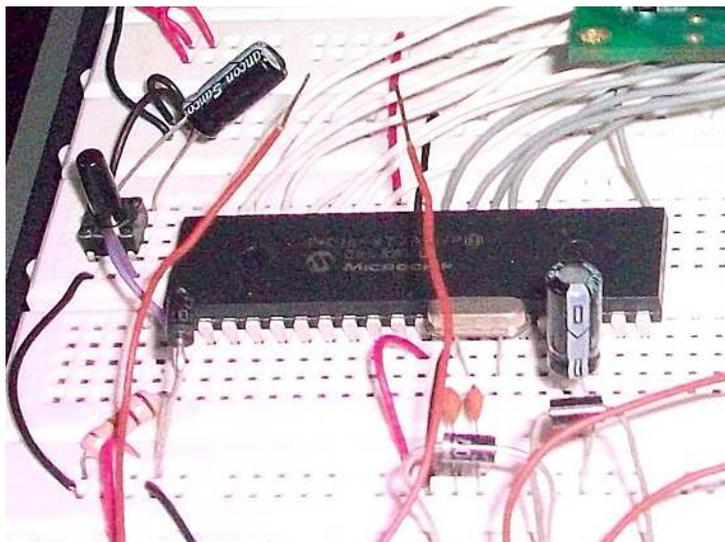


Figura 3.2 Circuito de Reset – Circuito de Reloj

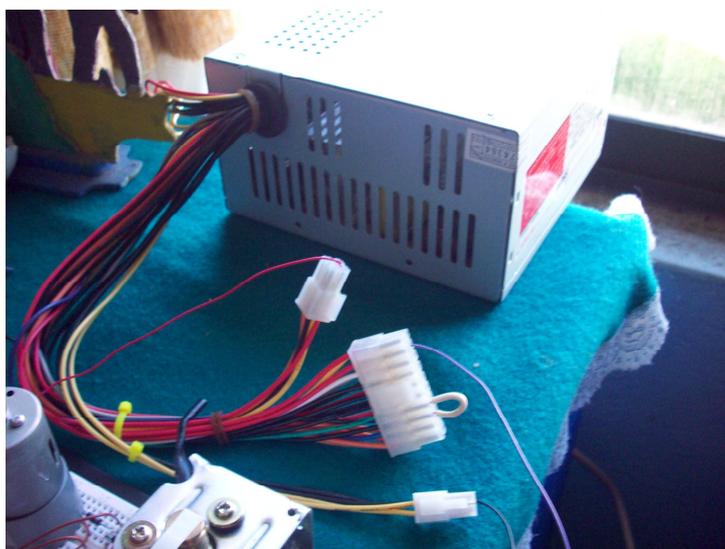


Figura 3.3 Fuente de Alimentación DC externa (Fuente Switching)

3.1.2 MÓDULO GLCD (GRAPHIC LIQUID CRYSTAL DISPLAY).

Se grabó en el microcontrolador un miniprograma (serie de comandos e instrucciones) desarrollado por el fabricante del software de programación (MickroElectronika) para el manejo del GLCD (Graphic Liquid Crystal Display) para mostrar las potencialidades del uso y control de este periférico de visualización.

El juego de instrucciones y comandos consiste en diferentes códigos que se envían al GLCD a través del bus de datos y control conectado al microcontrolador

en sus pórtricos (B – D) para obtener diferentes efectos de visualización como el de la fotografía.

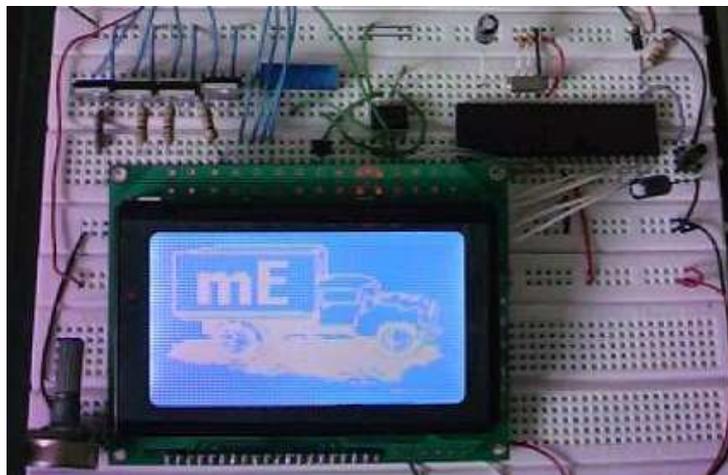


Figura 3.4 Visualización de un Bitmap en el GLCD.

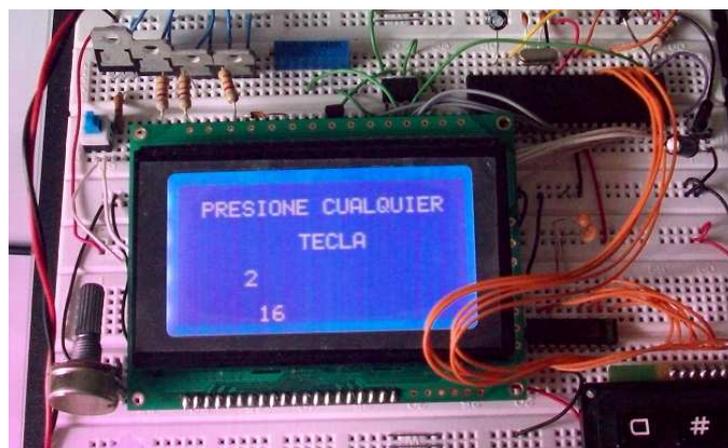


Figura 3.5 Visualización de caracteres numéricos y alfanuméricos.

3.1.3 MÓDULO SENSOR DE TEMPERATURA.

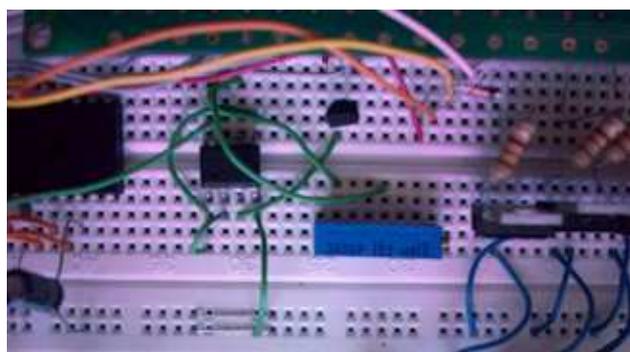


Figura 3.6 Conexión del Módulo Sensor de Temperatura.



Figura 3.7 Conversión de señales análogas (T°) a Digitales.

3.1.4 MÓDULO DE TECLADO HEXADECIMAL.

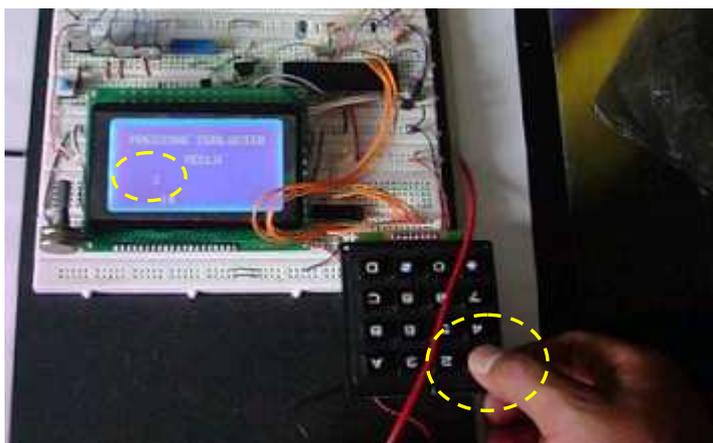


Figura 3.8 Conversión de señal análoga a digital.



Figura 3.9 Presentación de la tecla presionada (#1)

3.1.5 MÓDULO DE CONTROL DE MOTORES PASO A PASO.

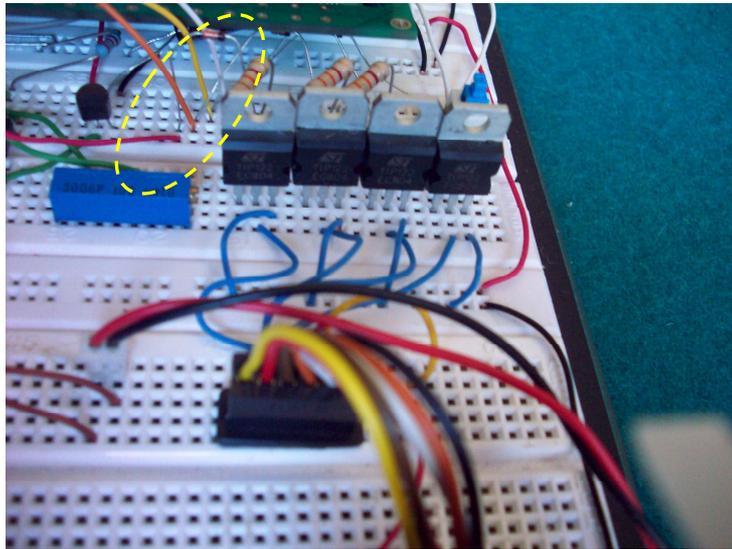


Figura 3.10 Conexión – Módulo Motores Paso a Paso.

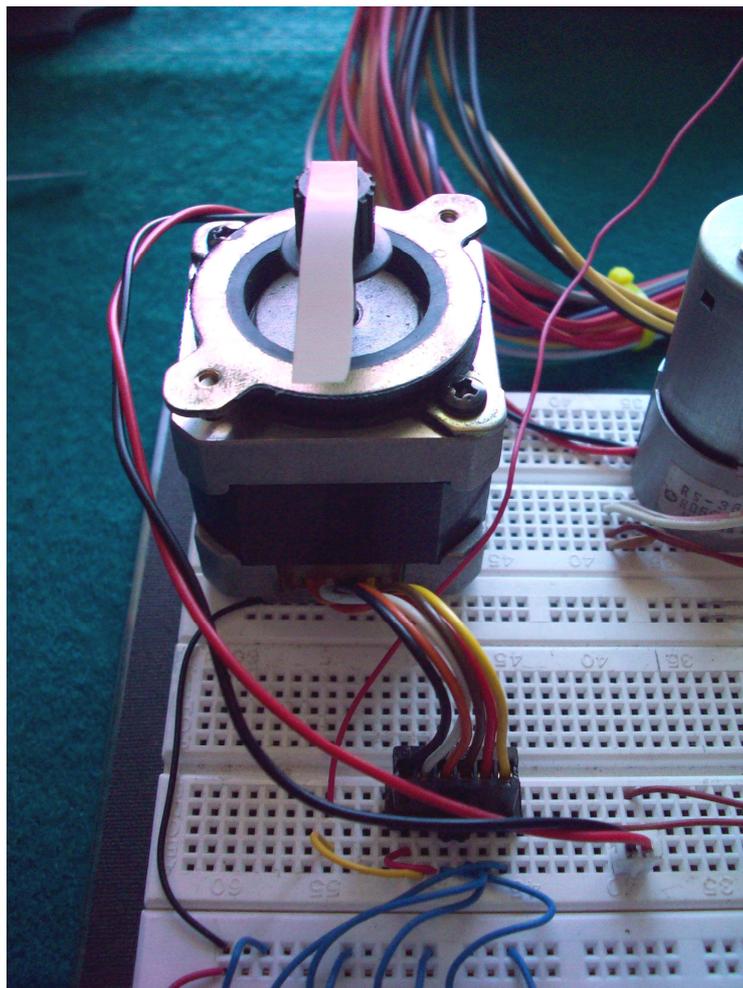


Figura 3.11 Conexión del Motor PAP.

3.1.6 MÓDULO DE CONTROL DE MOTORES DC.

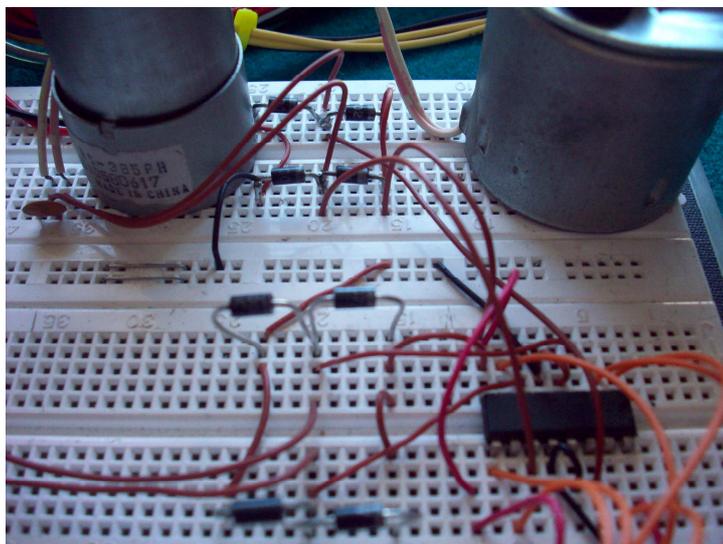


Figura 3.12 Configuración del L293B para el control de dos Motores DC.



Figura 3.13 Motores DC en marcha.

Una vez probados cada uno de los módulos, y verificado su normal funcionamiento, el siguiente paso es establecer el **interfaz de Comunicación USB** entre el Computador Personal (PC) y nuestra tarjeta de entrenamiento (Microcontrolador PIC18F4550), ha sabiendas que ésta actividad, es una potencialidad del microcontrolador y que la desarrolla sin mayor inconveniente de hardware, enviamos al diseño y creación de la tarjeta electrónica de nuestro Módulo.

3.2 CONSTRUCCIÓN DE TARJETA ELECTRÓNICA.

3.2.1 SOFTWARE UTILIZADO.



Figura 3.14 ORCAD CAPTURE FOR WINDOWS.

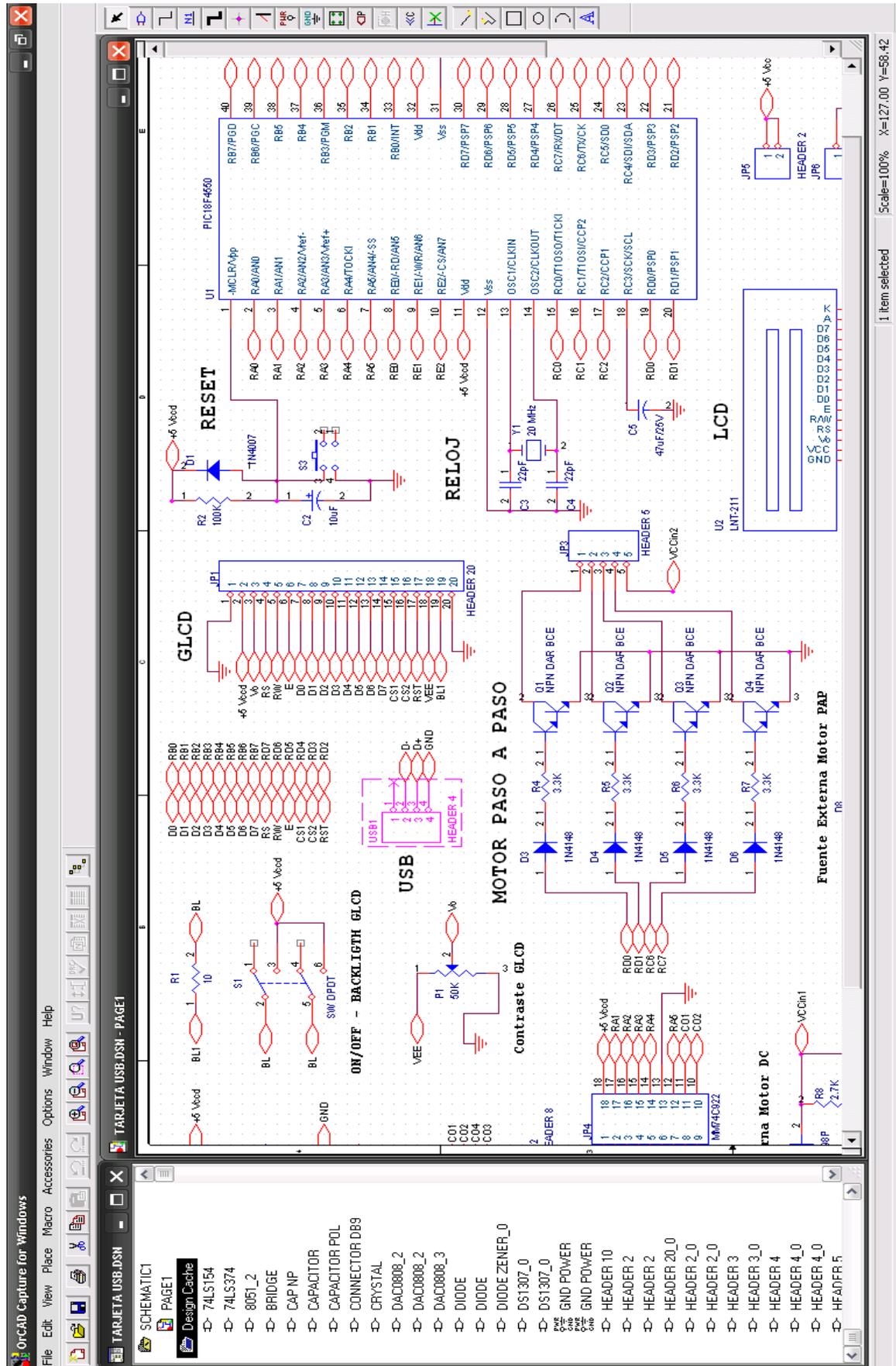
Nos permite generar páginas y diseños esquemáticos. Todos los diagramas y partes de esquemático están almacenados dentro de una librería y pueden fácilmente ser copiados, movidos y editados por el usuario hacia el diagrama esquemático creado, además nos presenta una gran cantidad de librerías con gran cantidad de dispositivos, una de las grandes ventajas radica en que nos permite crear librerías, dispositivos, nombrarlos, modificarlos, etc.



Figura 3.15 ORCAD LAYOUT FOR WINDOWS.

Nos permite diseñar y realizar el ruteo de las pistas del circuito impreso de manera automática ó manual, partiendo del Diagrama circuital creado en ORCAD CAPTURE FOR WINDOWS que crea el archivo.DSN que es el acceso directo a nuestro proyecto y a la vez varios archivos ejecutables como archivo.DBK, archivo.BOM, archivo.NET y el archivo.PCB que es el acceso directo a nuestro diagrama del circuito impreso.

3.2.2 DIAGRAMA CIRCUITAL ELABORADO EN ORCAD.



3.2.4 MÓDULO DIDÁCTICO – CIRCUITO IMPRESO.

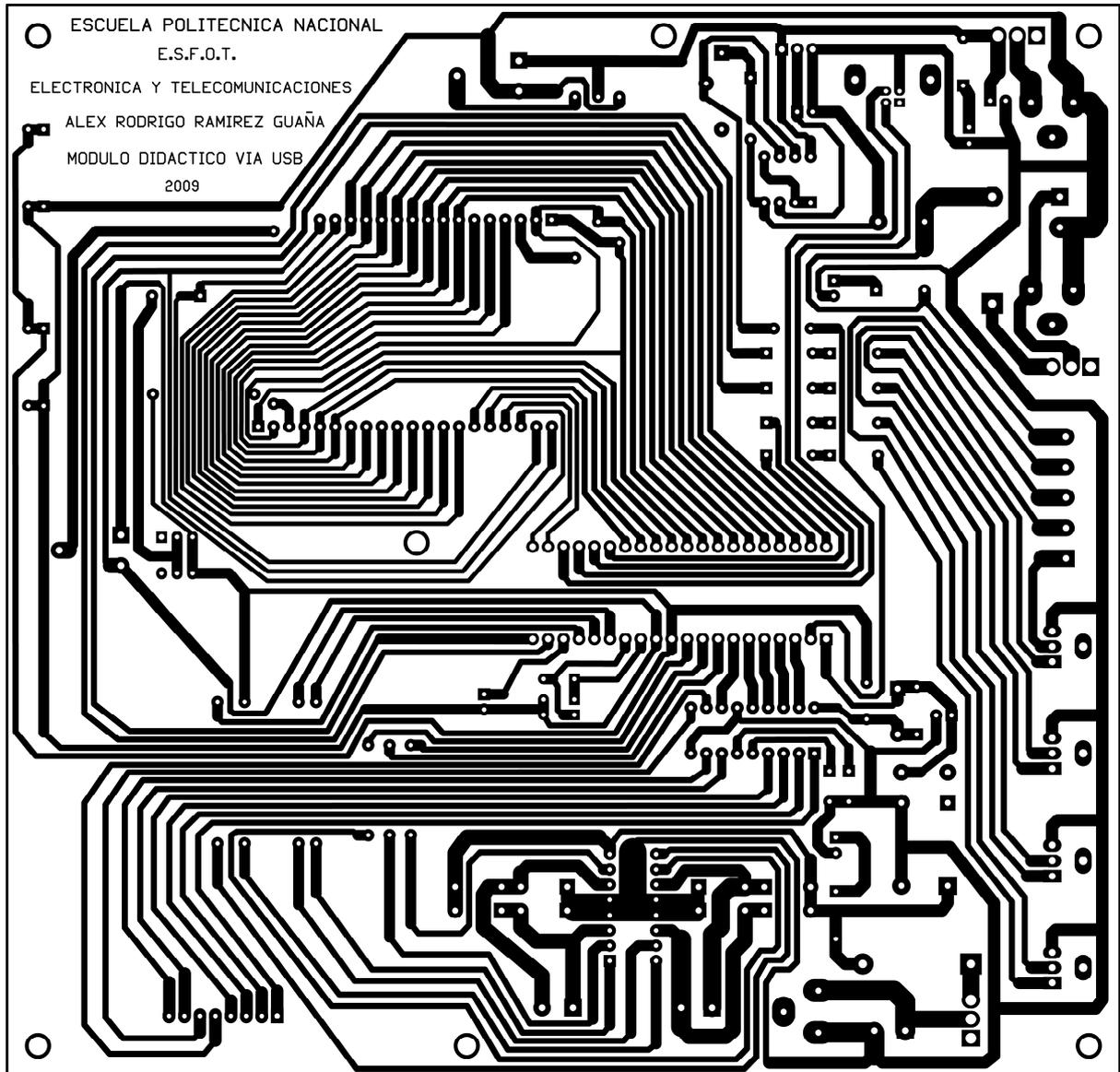


Figura 3.16 Circuito Impreso final.

3.2.5 MÓDULO DIDÁCTICO – SCREEN DE TARJETA.

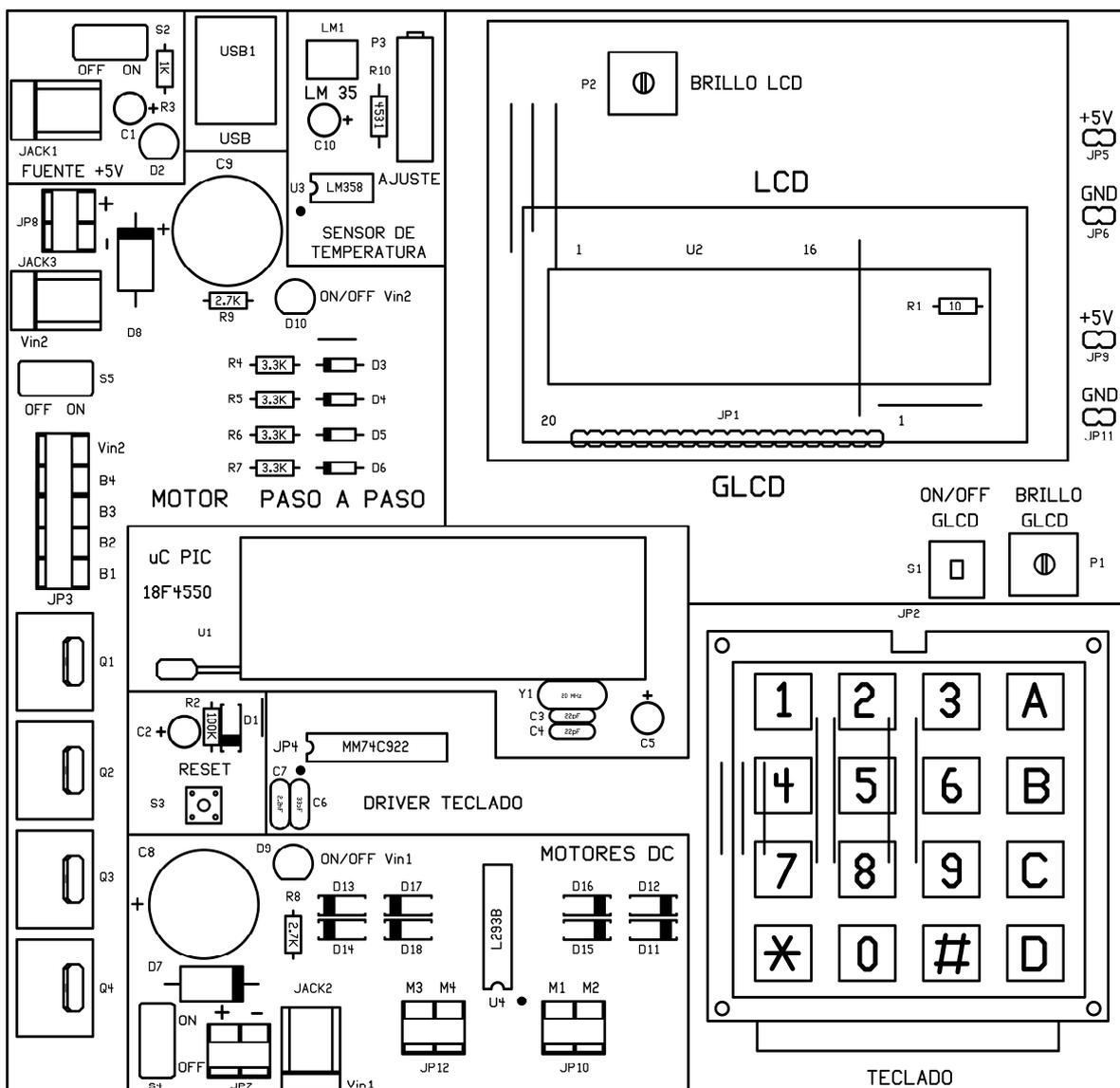


Figura 3.17 Screen de la Tarjeta ensamblada.

3.2.6 TARJETA ELECTRÓNICA ENSAMBLADA.

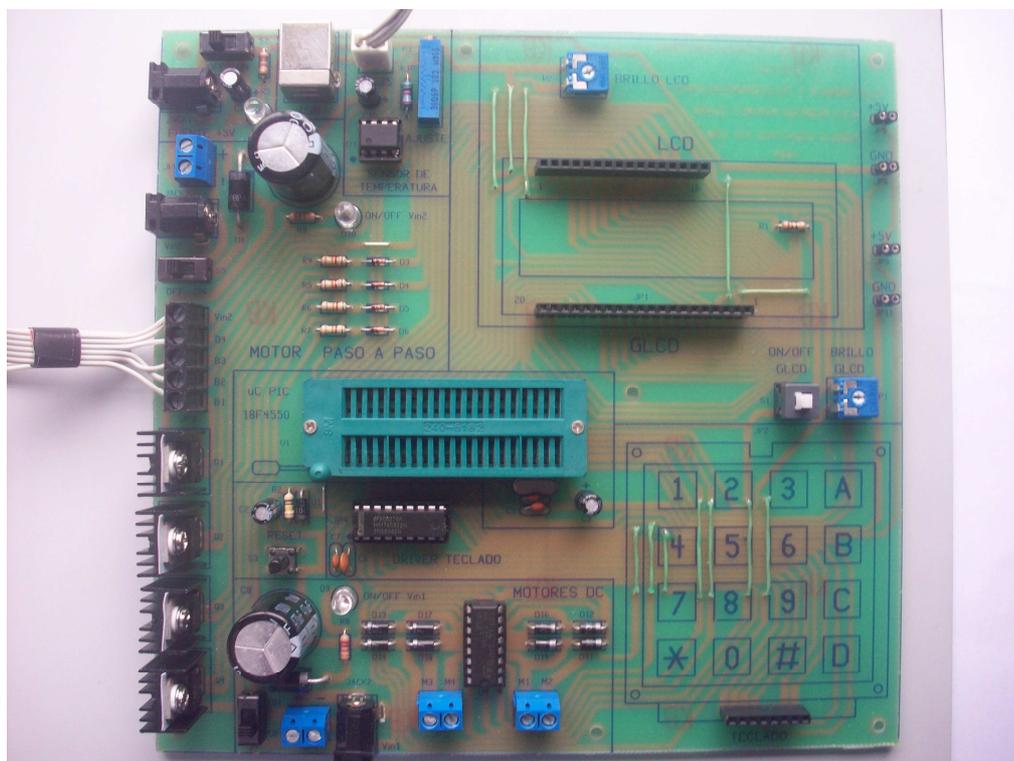


Figura 3.18 Módulo Didáctico ensamblado totalmente.

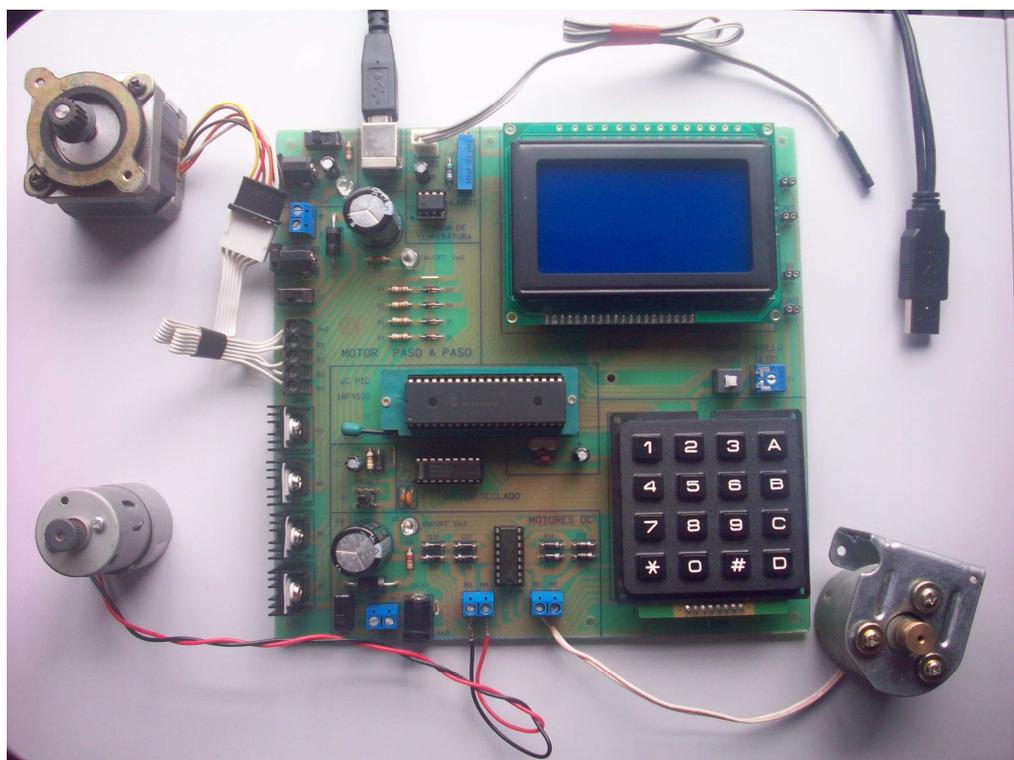


Figura 3.19 Módulo Didáctico con dispositivos a controlar.

3.2.7 PRUEBA DE COMUNICACIÓN USB.

Para el desarrollo de esta etapa se deben tener ciertos requerimientos previos:

- El módulo didáctico no debe estar sujeto a potenciales electrostáticos altos.
- Cable USB (conector A-B Estándar), largo desde 80cm a 5m.
- Módulo Didáctico compatible con software de programación IDE.
- Software Compilador (Proton Development Suite – MECANIQUE).
- Compilador de Aplicación (Visual Basic, Visual C, etc.).
- Fuente de alimentación DC externa (+5Vcc, +9Vcc, +12Vcc).

3.2.7.1 Generación de la interfaz USB en el PC.

SOFTWARE DE PC – PROTON IDE COMPILER.

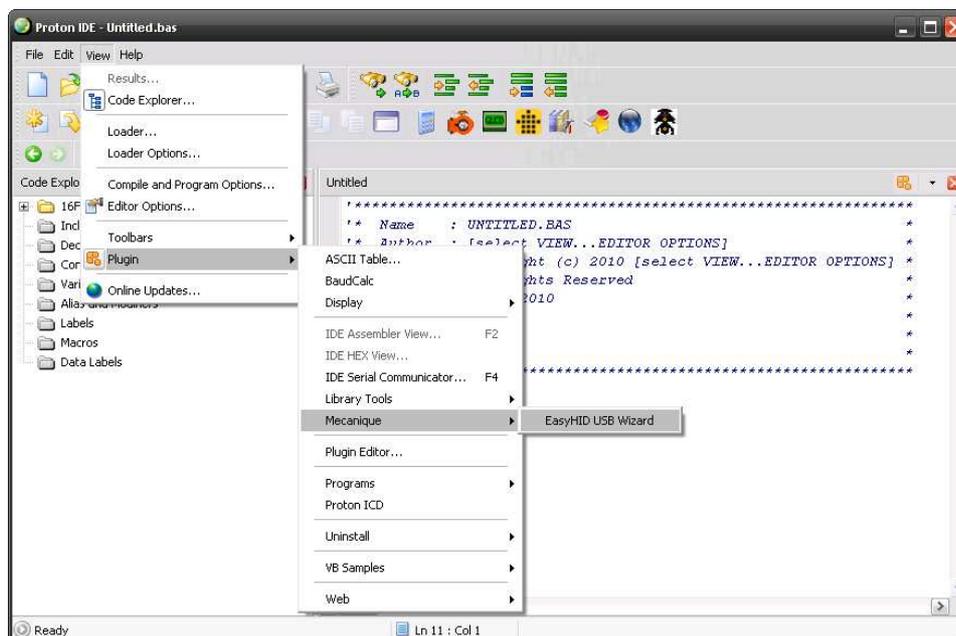


Figura 3.20 Interfaz de comunicación USB – EasyHID USB Wizard.

Se refiere a la creación del programa interfaz de comunicación USB esto lo conseguimos con una de las herramientas de las que dispone este software PROTON IDE el EASYHID USB Wizard (figura 3.20). EasyHID (Human Interface Device) es diseñado para proveer una solución simple a los problemas

normalmente asociados con la implementación de comunicaciones USB entre un microcontrolador PIC y un computador personal (PC). Una limitación del HID es su tasa de transmisión de datos que se limita a 64KB por segundo, sin embargo esta tasa de transmisión es mucho mayor que la tasa de comunicaciones RS-232 (12Kbps).

EasyHID WIZARD crea automáticamente dos plantillas de programa para compilar. El primero programa de uso exclusivo para el computador personal (Host Software), el segundo programa es grabado en el microcontrolador PIC (18F4550). Posteriormente se ingresa el nombre de la compañía, producto, serie (ID), y alguna configuración básica del dispositivo, para luego generar los dos programas y su posterior compilación.

El nombre y la localización del proyecto son usados por el EasyHID para ubicar el código generado, y los dos programas tanto para el PC como para el dispositivo USB que también son llamados. A continuación se designa el software compilador (PROTON IDE), el microcontrolador a utilizar PIC18F4550 y el uso interrupciones para el servicio de USB (figura 3.20) que consiste en revisar periódicamente si existe dispositivo USB conectado al computador, además se elige el lenguaje de programación para la aplicación a desarrollar (Visual Basic).

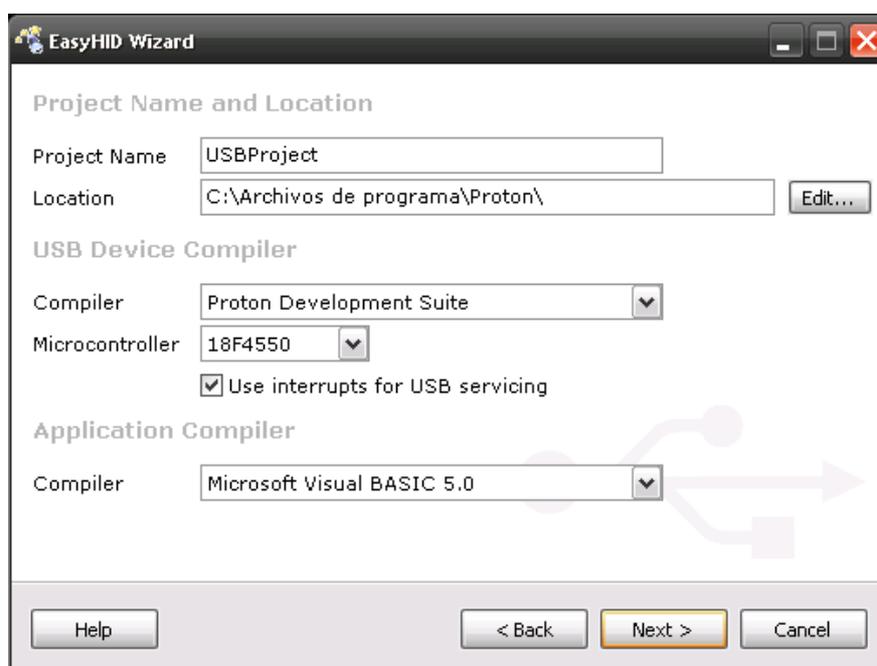


Figura 3.21 Ventana del EasyHID USB Wizard.

Ejemplo:

Si la localización es c:\Archivos de programa\Proton\ y el nombre esta dado por default, cuando el EasyHID Wizard genere los archivos se guardará en las siguientes carpetas:

C:\Archivos de programa\Proton\USBProject\PDS

Los archivos:

- USBPROJECT.ASM
- USBProject.bas (Visual Basic Module)
- USBPROJECT.COD
- USBPROJECT.ERR (Registro de Errores)
- USBPROJECT.HEX
- USBPROJECT.LST
- USBProject.mci
- USBPROJECT.PBE
- USBPROJECT.PBP
- USBProjectDES.inc

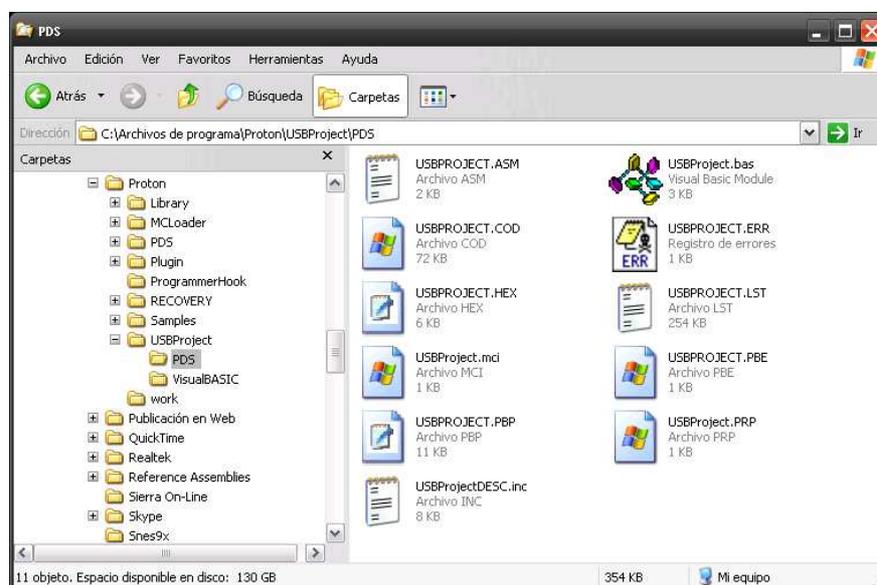


Figura 3.22 Archivos creados por el EasyHID USB Wizard.

Y para el interfaz de comunicación USB se guardará en la siguiente carpeta:

C:\Archivos de programa\Proton\USBProject\VisualBASIC

Los archivos:

- FormMain.frm (Visual Basic Form File)
- mcHIDInterface.bas(Visual Basic Module)
- USBProject.vbp (Visual Basic Project)
- USBProject.vbw (Visual Basic Project Workspace)



Figura 3.23 Archivos creados por el EasyHID USB Wizard.

EasyHID es de uso simple, para generar el código que permite al usuario establecer rápidamente la comunicación bidireccional entre el microcontrolador PIC y el PC. EasyHID ha sido diseñado para el trabajo con el estándar USB 2.0 y demás aplicaciones con Windows. El EasyHID crea y usa el archivo DLL (mcHID.dll) que se envía al host para la aplicación final.

EDITANDO Y COMPILANDO EL PROGRAMA USB CREADO.

Como lo mencionamos anteriormente cada pin del microcontrolador (40PDIP) esta estratégicamente asignado, definido y determinado para desarrollar una actividad específica dentro del funcionamiento de cada módulo de nuestra tarjeta electrónica; estos y otros aspectos debe preveer y tener presente el programador al momento de editar el programa (definir variables) con cien por ciento de correspondencia entre el software y el hardware implementado.

Una vez creado el archivo USBProject.bas, en PROTON IDE el siguiente paso es editar y generar el lenguaje hombre-máquina para comunicarse y entenderse. Para ello hacemos uso del Set de Instrucciones que nos proporciona el software de programación PROTON IDE (Anexo 4), que presenta gran flexibilidad y

similitud en programación que Visual Basic, software's que nos permitirán entablar la interfaz de comunicación USB entre el dispositivo de interfaz humana (Módulo Didáctico) y el computador personal.

NOTA. En conclusión son lenguajes de programación que se crean, para que el hombre manipule y controle dispositivos eléctricos, electrónicos ó mecánicos mediante la manipulación de las señales eléctricas. Una vez editado el programa el siguiente paso es compilar y verificar errores.

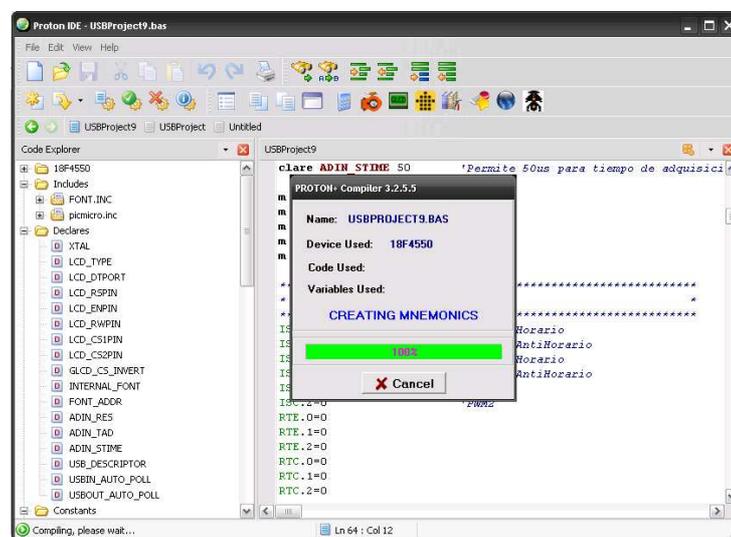


Figura 3.24 Estado de la compilación del proyecto.

VENTANA DE COMUNICACIÓN USB.

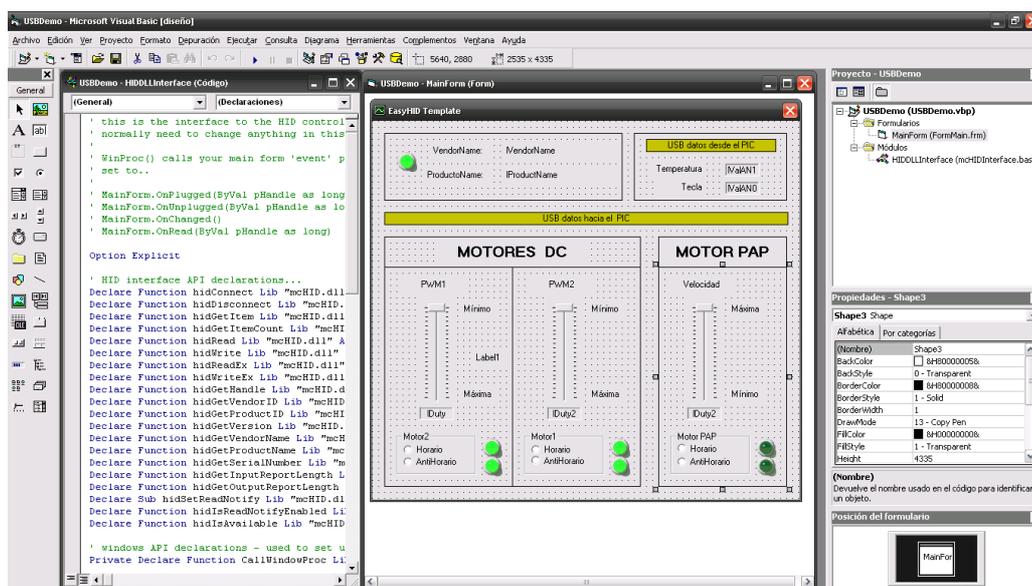


Figura 3.25 Archivo ejecutable de Visual Basic.

Como se mencionó en apartados anteriores el microcontrolador es un computador completo (microprocesador + E/S + memoria + otros periféricos) de limitadas prestaciones contenido en un circuito integrado programable y se destina a gobernar una sola tarea con el programa que reside en su memoria, sus líneas de entrada/salida soportan las conexiones de sensores y dispositivos actuadores a controlar.

El programa se lo graba en el microcontrolador para ello, existe hardware y software diseñados específicamente a desarrollar esta tarea. En la memoria disponible del microcontrolador PIC se almacena el archivo.hex que se generó al compilar el programa.

Como se menciona anteriormente en el software compilador se generan dos plantillas de programa uno de uso para el host del PC (USBProject.vbp, ó archivo de aplicación.exe que es generado por el programador) y el segundo programa para que resida en la memoria del microcontrolador PIC (USBProject.bas-archivo.hex), y de esta manera establecer la comunicación bidireccional entre cualquier computador personal (PC) y nuestro dispositivo USB (Módulo Didáctico). A continuación se ubica en la tarjeta electrónica el microcontrolador con el programa ya grabado en su memoria.

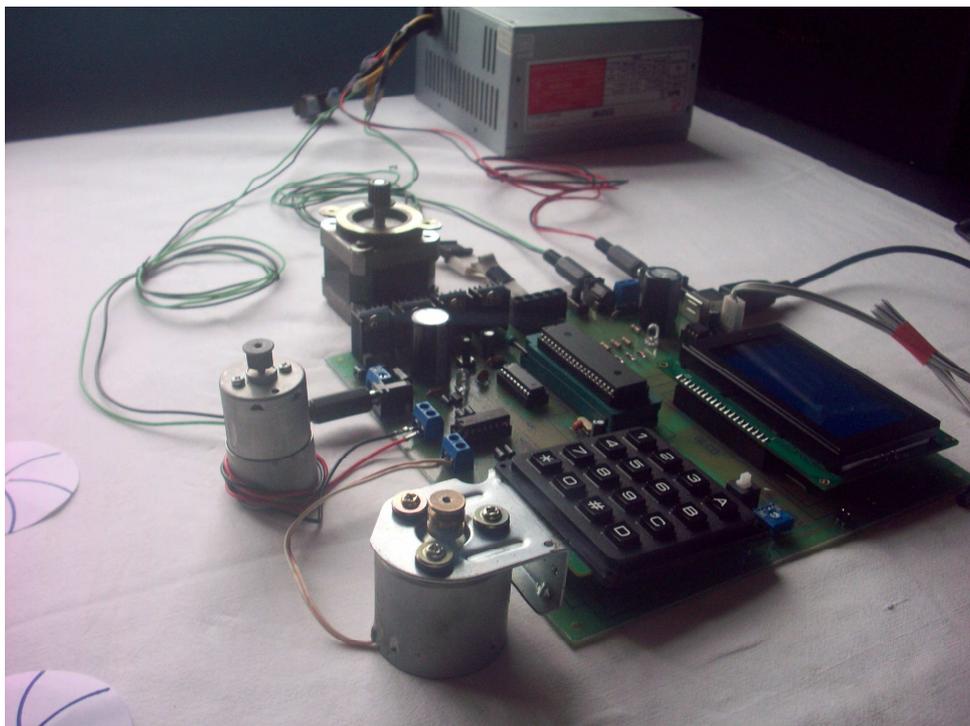


Figura 3.26 Módulo Didáctico listo para entrenamiento.

3.2.7.2 Módulo Encendido con conexión USB.

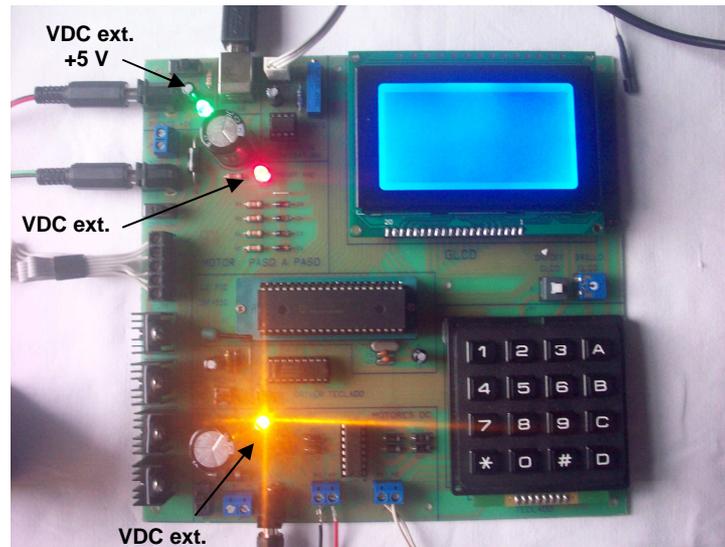


Figura 3.27 Presencia de voltaje DC externo en la tarjeta.

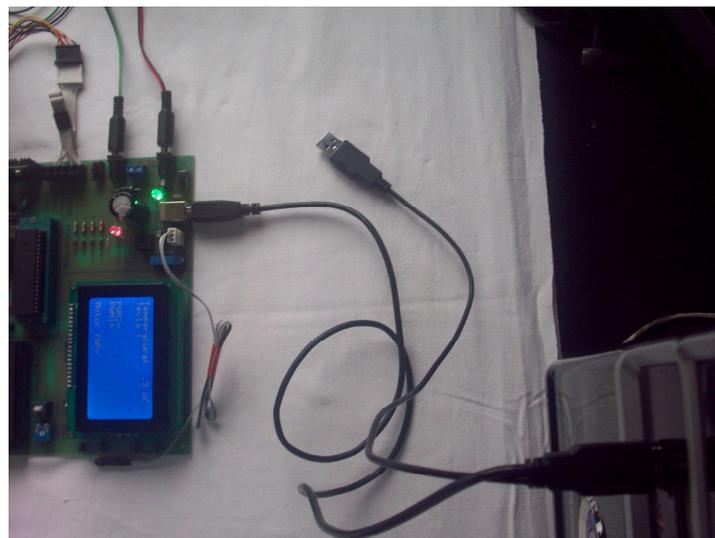


Figura 3.28 Conexión USB del Módulo Didáctico con el PC.

Al conectar el Módulo Didáctico al puerto USB del computador aparece un dispositivo de interfaz humana (HID) es decir nuestro Módulo Didáctico lo reconoce e instala el software de identificación automáticamente para poder controlarlo.

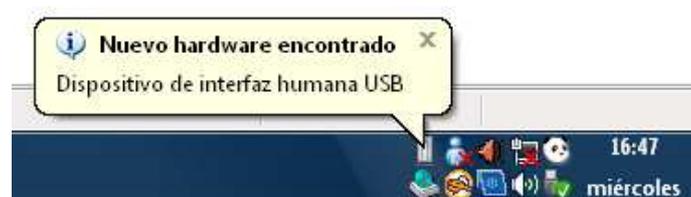


Figura 3.29 Nuevo Hardware encontrado por el PC.

A continuación ubicamos nuestro programa “**USBProjetc.vbp**” que se guardo en la carpeta: C:\Archivos de programa\Proton\USBProject\VisualBASIC, lo abrimos y hacemos compilar ó directamente damos doble clic en el archivo.exe de aplicación, como se muestra a continuación.

Al ejecutar este archivo.exe en el computador, el dispositivo es reconocido e inmediatamente inicia la comunicación bidireccional entre el dispositivo USB (Módulo Didáctico) y el host del computador personal (PC). Nuestra tarjeta electrónica posee un Módulo Sensor de Temperatura el cual mide y presenta la temperatura ambiente inmediatamente iniciada la comunicación USB mientras espera instrucción desde el computador personal (PC) para realizar el control de los dispositivos conectados como el GLCD, Motores PAP, Motores DC, teclado hexadecimal.



Figura 3.30 Ventana de Aplicación (Archivo.exe-Visual Basic).

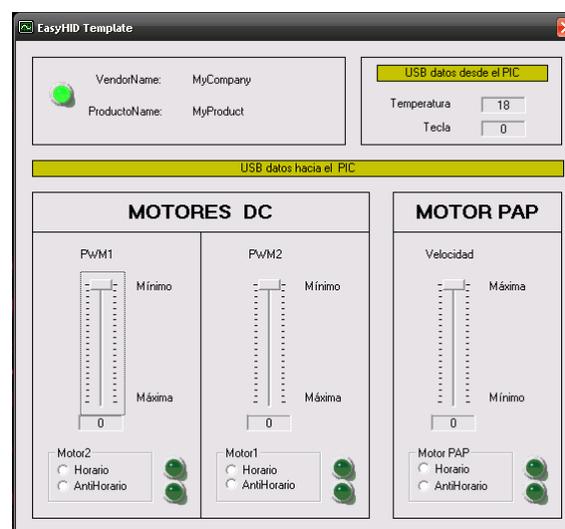


Figura 3.31 Ventana de aplicación con el módulo conectado al PC.

3.2.7.3 Control del Módulo Didáctico desde el computador vía USB.

El Display de Cristal Líquido Gráfico (GLDC), al ser un dispositivo de visualización en él podemos presentar caracteres numéricos, alfanuméricos e incluso bitmaps. En este caso el display presenta caracteres que nos darán información del valor de temperatura ambiente que mide el CI LM35, el valor de la tecla presionada en el Módulo, así como la velocidad y sentido de giro de los Motores DC y Paso a Paso. Todos los datos informativos que presenta el GLCD, también se visualizan en la ventana de aplicación (archivo.exe – Visual Basic) en el PC para comprobar la comunicación USB bidireccional del computador personal con el microcontrolador PIC de nuestra tarjeta electrónica.

MODULO GLCD (GRAPHIC LIQUID CRYSTAL DISPLAY).

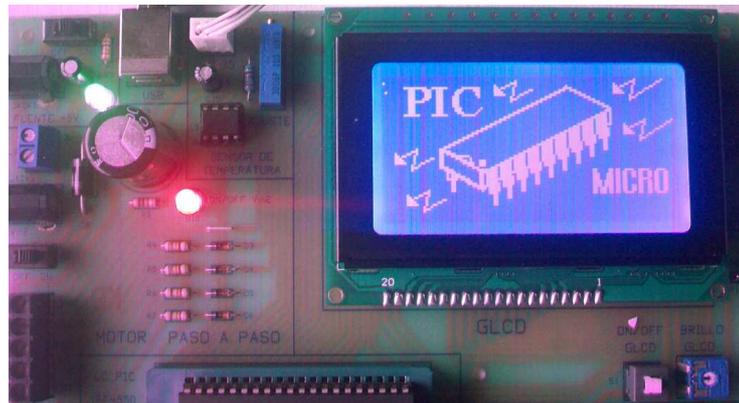


Figura 3.32 Bitmap presentado en el GLCD al encender el Módulo.

MÓDULO SENSOR DE TEMPERATURA.

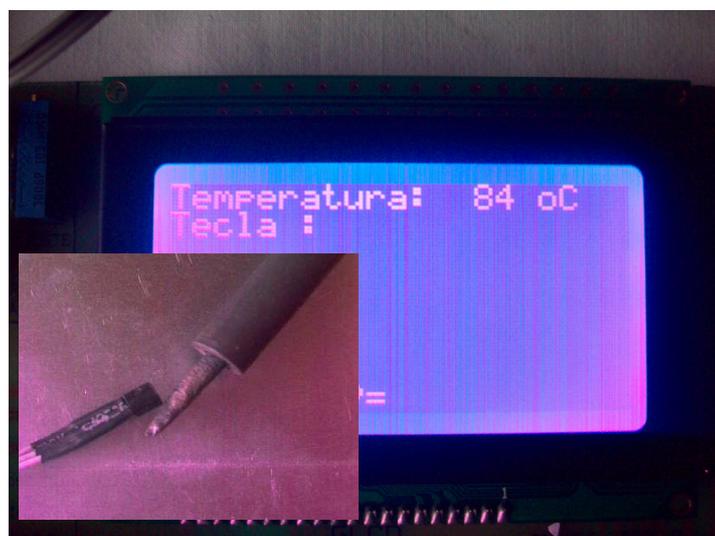


Figura 3.33 Temperatura medida y enviada al GLCD.

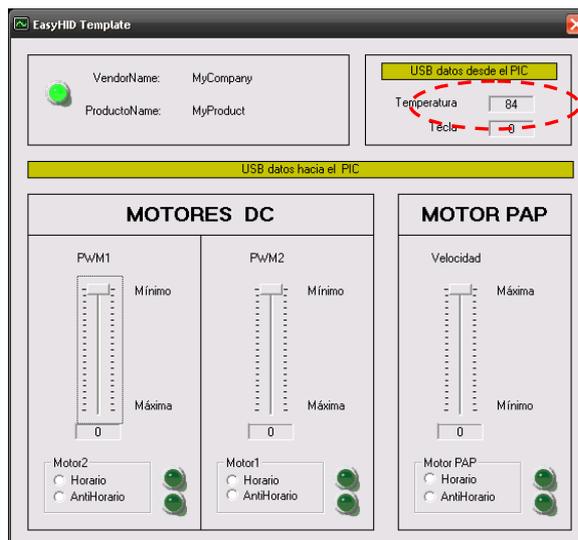


Figura 3.34 Ventana PC - Valor de temperatura enviado al PC.

MÓDULO TECLADO HEXADECIMAL.



Figura 3.35 Tecla "A" presionada GLCD.

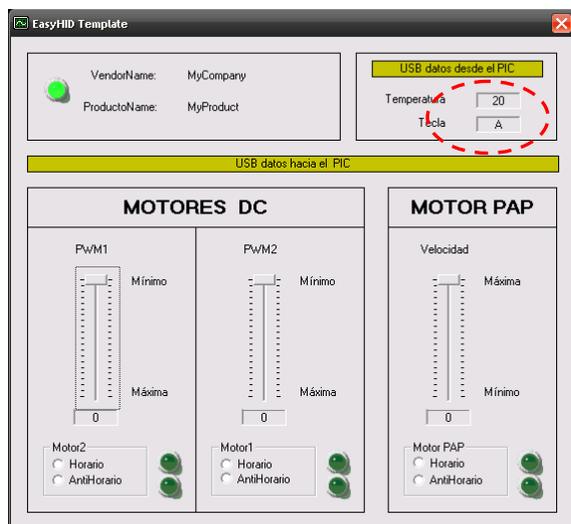


Figura 3.36 Ventana PC – Tecla "A" identificada por el PC.

MÓDULO MOTORES DC



Figura 3.37 Motores DC en marcha.

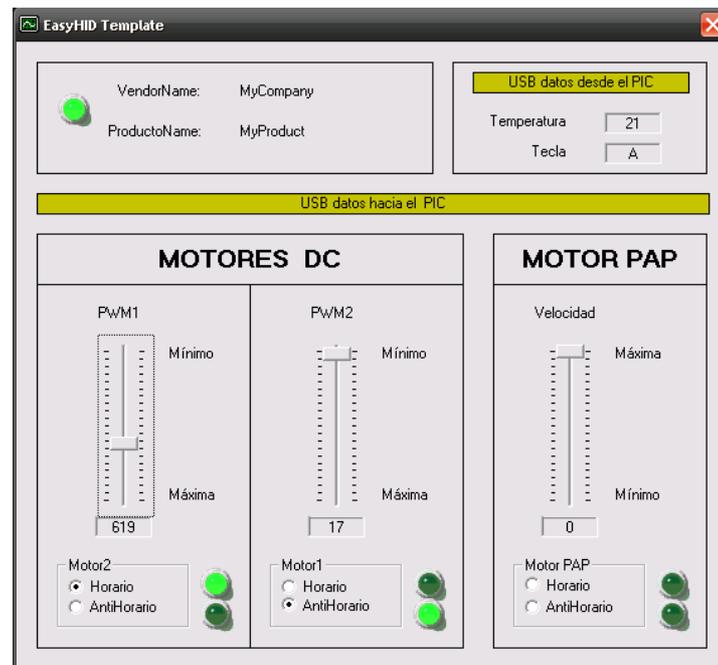


Figura 3.38 Ventana PC – Motores DC en marcha.

MÓDULO MOTOR PASO A PASO

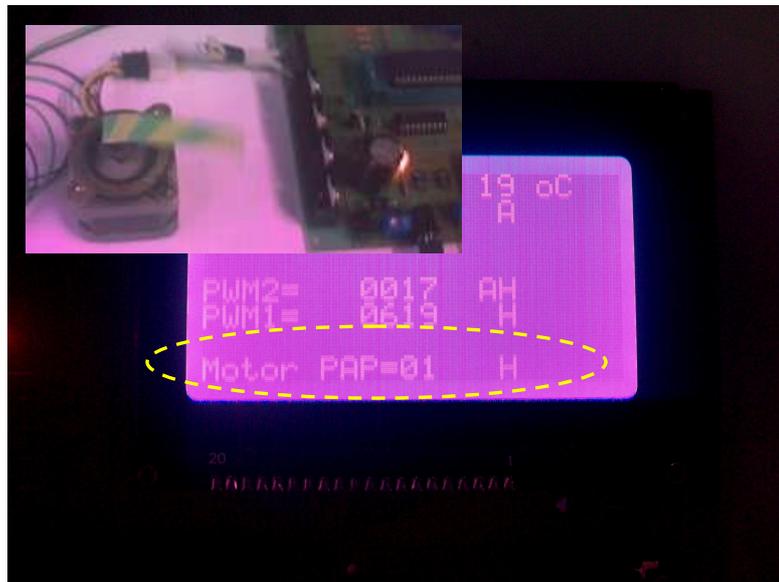


Figura 3.39 Motor Paso a Paso en marcha GLCD – sentido horario.

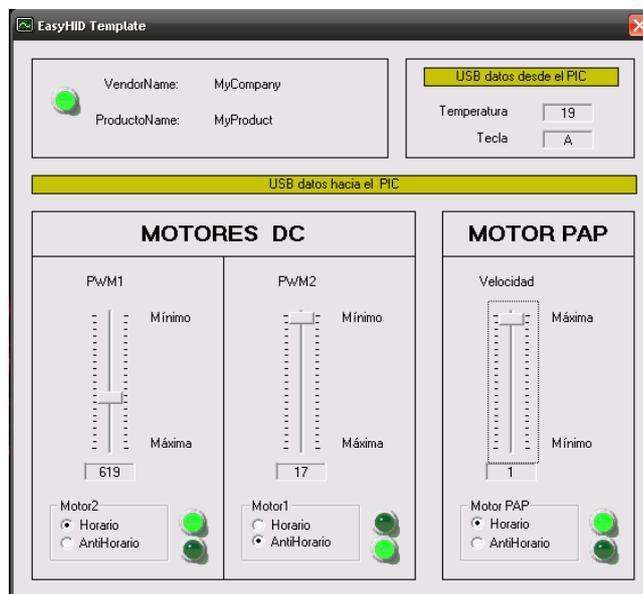


Figura 3.40 Ventana PC – Motor Paso a Paso en marcha.

CAPITULO 4.

4.1 CONCLUSIONES.

- A través del proyecto desarrollado, se ha creado un módulo didáctico microcontrolado capaz de comunicarse vía puerto USB con el computador y desde el cual el usuario (programador) puede manipular dispositivos eléctricos, electrónicos conectados a la tarjeta electrónica implementada previo a una determinada programación del microcontrolador.
- Se obtuvo el microcontrolador USB que cumple satisfactoriamente nuestras exigencias de funcionamiento y de comunicación, pero nos presenta una limitación en la capacidad de memoria que posee, para el desarrollo de proyectos sofisticados (extensos).
- Una de las características de la comunicación USB, entre un dispositivo de interfaz humana (teclado, ratón, parlantes, etc.) y el computador consiste en que el primero necesita regularmente una fuente de voltaje para su funcionamiento la misma que de costumbre es proporcionada por el computador, a través de los cables y conectores USB. Después de haber desarrollado el proyecto y determinar que la fuente del PC, en nuestro caso es opcional hemos omitido el uso de la línea $+V_{CC}$ del puerto USB con el fin de evitar daños ó averías, que se producen en el periférico o en la placa base del computador por mala manipulación o mala conexión del periférico (módulo didáctico).
- Sabiendo que una potencialidad del módulo didáctico es el LCD Gráfico como dispositivo de visualización, el control de este dispositivo se convierte en un reto para el programador, dejando como único limitante para el desarrollo de proyectos el conocimiento e ingenio del usuario.

4.2 RECOMENDACIONES.

- Es importante que al generar el programa que se almacena en el computador y el que se graba en el microcontrolador para establecer la comunicación USB tenga el cien por ciento de correspondencia en software y hardware ya que si no coinciden se obtendrán resultados erróneos a los deseados y el sistema operativo del computador indicará el mal uso o funcionamiento del dispositivo USB (Módulo Didáctico).
- Teniendo en conocimiento que la capacidad de la memoria interna de los microcontroladores es relativamente pequeña, la solución a esta limitante, puede ser, implementar una memoria EEPROM externa de gran capacidad y bajo consumo.
- A partir de la creación del Módulo Didáctico (dispositivo USB de interfaz humana), se podría diseñar una red de computadores, a través de la cual, el dispositivo implementado puede ser compartido y controlado por el usuario desde la comodidad del escritorio, lo que significaría tener el módulo disponible para cada estudiante o para cada grupo de trabajo en el Laboratorio.
- Cada vez que se construya o implemente una tarjeta de entrenamiento (módulo didáctico) el estudiante debe emitir el respectivo Manual del Usuario (Anexo 6) que servirá de ayuda y fuente de información para el desarrollo de las prácticas o pruebas en el Laboratorio.

4.3 REFERENCIAS BIBLIOGRÁFICAS.

Referencia en textos.

1. GONZÁLEZ VÁZQUEZ. McGraw-Hill. Introducción a los microcontroladores.
2. BERNARD Odant. Paraninfo. Microcontroladores 8051 y 8052.
3. MARTÍNEZ PÉREZ Javier, BARRÓN RUÍZ Mariano. McGraw-Hill. Prácticas con microcontroladores de 8 bits.
4. PEATMAN. McGraw-Hill. Design with Microcontrollers.
5. J. MA. Angulo. Paraninfo. Microcontroladores PIC.
6. BOYLESTAD Nashelsky. Prentice Hall, México, 449, (1997). Fundamentos de Electrónica.
7. RODRÍGUEZ Irlenys Tersek. Edición 179. Página 5. Funcionamiento y Control a Través de un PC.
8. ANCHIÑA AGUAS Marcelo, CAJAMARCA Luis Fernando, Diseño y construcción de una tarjeta de desarrollo para el funcionamiento del PIC18F2550. Tesis de grado. 2007.

Referencia en la Web.

9. MICROCONTROLADOR PIC18F4550.
<http://www.microchip.com>
10. PUERTO USB
<http://www.beyondlogic.org/usbnutshell/usb7.htm>
11. SOFTWARE DE COMPUTADOR - PROTÓN IDE
www.mikroe.com
www.mecanique.co.uk
12. GLCD (GRAPHIC LIQUID CRYSTAL DISPLAY).
www.microcontroladores.com.
PDF TS-12864A-2 LCD Module Specification Ver1.0 Total 6 pages.

13. SENSOR DE TEMPERATURA.

LM35 National Semiconductors Corporation Guide – 1995

LM358P ECG Semiconductors Master Replacement Guide.

14. TECLADO HEXADECIMAL

<http://atc.urg.es/docencia/udigital/index.html> 30-08-2002.

15. CONTROL DE MOTORES PASO A PASO.

<http://www.tecnologiaseso.es>. charliebrawn2001@yahoo.es

16. CONTROL DE MOTOR DE CORRIENTE CONTÍNUA

L293B SGS-THOMSON MICROELECTRONIC

17. CONSTRUCCION DE LA TARJETA ELETRÓNICA.

PDF Manual Orcad Capture / Layout for Windows. intl@orcad.com.