

UNIVERSIDAD MAYOR DE SAN ANDRES
FACULTAD DE TECNOLOGIA
CARRERA DE ELECTRONICA Y TELECOMUNICACIONES



BANDA TRANSPORTADORA DE LADRILLOS PARA LA
EMPRESA DE CERAMICA AVAROA

Trabajo de Aplicación – Examen de Grado presentado para obtener el Grado de Licenciatura

POR: IVAN ALVARO JUAREZ QUISBERT

LA PAZ – BOLIVIA

Noviembre, 2022

DEDICATORIA

Dedico el presente trabajo a mi madre Irene Quisbert Torrez por haberme apoyado incondicionalmente en cada etapa de mi vida gracias a su apoyo pude superar metas y obstáculos agradezco su amor incondicional y por haberme dado la educación que no pudo tener ella, gracias por todo mama.

AGRADECIMIENTOS

Agradecer a Dios por estar en esta vida y agradecer a todos los docentes de la carrera de Electrónica y Telecomunicaciones (UMSA) por haberme instruido sus conocimientos a través de todos estos años, que su dedicación en la enseñanza siempre perdure para construir una Bolivia con desarrollos tecnológicos para nuestras futuras generaciones.

RESUMEN

El presente trabajo de aplicación, propone implementar un sistema de banda transportadora de ladrillos para la empresa de cerámica “Avaroa” ubicado en la ciudad industrial de Viacha de la provincia Ingavi del Departamento de La Paz, tiene como objetivo, transportar el producto ya concluido y secuencialmente realizar el conteo de los mismos, para su respectivo transporte y ser distribuido en el mercado interno de acuerdo a las demandas de este producto.

La implementación del proyecto de banda transportadora de ladrillos, requiere de dispositivos tales como: un microcontrolador, motores de corriente continua, sensores ópticos laser - LDR, visualizador LCD, teclado matricial, leds indicadores del proceso, parlante o altavoz, pulsadores, circuitos de potencia (amplificadores de corriente, transistores NPN y PNP), con todos estos elementos, se mostrará el desarrollo del proceso del montaje y funcionalidad del prototipo de banda transportadora plasmados en una maqueta.

De inicio, describimos cada uno de estos elementos electrónicos a utilizarse en el marco teórico, posterior a ello, el desarrollo del proyecto se realizó por etapas, por otro lado, la implementación de una maqueta me permitió observar todo el sistema de control. Para desarrollar la banda transportadora utilizaremos el PIC16F877A que es un chip de 40 pines el cual realizará todo el proceso que se haya programado para dicho proceso, debidamente este PIC será programado para realizar todo el proceso que queremos demostrar.

Para la programación, se utilizara el lenguaje de programación C el cual con ayuda del compilador MikroC PRO for Pic y Proteus, se realizará previamente una simulación integrando todos los dispositivos haciendo las debidas conexiones de manera digital y evaluando cada suceso que se presente para que posteriormente sea grabado nuestro código con ayuda de un grabador de PICs, es este caso, utilizaremos el PICKit 2 el cual

nos ayudara para el grabado del código para posteriormente implementarlo en nuestra maqueta y poder observar el proceso que queremos.

La banda transportadora de ladrillos, iniciará el proceso primeramente inicializando la pantalla LCD con el objetivo de introducir un valor numérico con ayuda del teclado matricial, el cual será la cantidad de ladrillos a ser contados, inicializando el número de conteo nos mostrará el dato en pantalla LCD, para que posteriormente inicialice la banda transportadora el cual llevara los ladrillos el cual pasara por el sensor óptico laser – LDR para que este pueda ser contado y mostrado por pantalla.

La banda transportadora puede ser regulado la velocidad del motor, aumentando y disminuyendo su ciclo de trabajo, una vez contado el ladrillo pasara a un almacenaje temporal el cual mantendrá los ladrillos hasta que la cuenta finalice, este proceso se realizara hasta culminar todo el conteo que se introdujo por el teclado realizando una comparativa internamente el cual cuando se haya terminado todo el conteo sonara una alarma de finalización de conteo y ahí una segunda banda transportadora llevara todos los ladrillos almacenados para que estas pueda ser recogidas por un medio de transporte para su posterior comercialización y distribución en los puestos de venta del producto.

INDICE

DEDICATORIA	i
AGRADECIMIENTOS	ii
.....	ii
RESUMEN.....	iii
CAPITULO I.....	1
1.1 Introducción.....	1
1.2 Antecedentes.....	1
1.3 Planteamiento del problema	4
1.4 Objetivos del proyecto.....	4
1.4.1 Objetivo General.....	4
1.4.2 Objetivos Específicos.....	4
1.5 Justificación.....	4
1.6 Delimitación	5
CAPITULO II	6
MARCO TEORICO.....	6
2.1 El Microcontrolador	6
2.1.1 Microcontrolador	6
2.1.2 Elementos de un Microcontrolador	6
2.1.3 Elementos internos en un microcontrolador	7
2.1.4 Diferencia entre un microcontrolador y un microprocesador	8
2.1.5 Microcontrolador incluye un Microprocesador	9
2.1.6 Microcontrolador y sus Periféricos.....	10

2.1.7 Microcontrolador y sus memorias	12
2.1.8 Microcontrolador y sus Registros	13
2.1.9 Memoria externa	13
2.1.10 Como se programa a un microcontrolador	14
2.2 El PIC 16f877a	14
2.2.1 Introducción	14
2.2.2 Descripción del PIC16F877A	15
2.2.3 Características del PIC16F877A	15
2.2.4 Diagrama en Bloques del PIC16F877A	17
2.2.5 Descripción de los pines del PIC16F877A	17
2.2.6 Organización de la memoria	18
2.2.7 Memoria de Programa Flash ROM 8Kx14	19
2.2.8 Memoria de Pila 8x13 SRAM 8x13	19
2.2.9 Memoria de Datos EEPROM 256 Bytes	19
2.2.10 Memoria de Datos SRAM 368 Bytes	20
2.2.11 Memoria de Configuración Flash ROM 1x14 bit	20
2.2.12 Puertos Entrada y Salida	20
2.3 Display Liquido Lcd	22
2.3.1 ¿Que es un LCD?	22
2.3.2 ¿Como es su conexión?	22
2.3.3 ¿DDRAM y CGROM?	24
2.3.4 Características y especificaciones técnicas de la pantalla LCD HD44780	25
2.3.5 Características eléctricas	25
2.3.6 Características mecánicas	26

2.3.7 Índices absolutos máximos	26
2.3.8 Conexiones de clavijas de interfaz.....	27
2.3.9 Mostrar código de dirección de caracteres	28
2.4 Teclado Matricial.....	28
2.4.1 Descripción	28
2.4.2 Especificaciones Técnicas	29
2.4.3 Funcionamiento y programación de un Teclado Matricial.....	30
2.5 Motores DC	31
2.5.1 Descripción	31
2.5.2 Funcionamiento	32
2.5.3 Variaciones en el diseño del motor.....	33
2.5.4 Motores de imán permanente.....	33
2.5.5 Motor eléctrico compound.....	34
2.5.6 Motores sin escobillas.....	34
2.5.7 Ventajas	34
2.5.8 Especificaciones Técnicas	35
2.6 Driver L293D	36
2.6.1 Descripción.....	36
2.6.2 Diagrama interno	37
2.7 Transistores.....	38
2.7.1 El Transistor BC558	38
2.7.2 El Transistor Bc548	42
2.7.3 Transistor TIP 41	46
2.8 Cristales de Cuarzo.....	47

2.8.1 Descripción.....	47
2.8.2 Características.....	48
2.8.3 Resonancia.....	48
2.8.4 Construcción.....	49
2.8.5 Circuito eléctrico equivalente.....	50
2.9 Parlante o Altavoz.....	50
2.9.1 Descripción.....	50
2.9.2 Principales características de los altavoces.....	51
2.9.3 Especificaciones y características técnicas.....	53
2.10 Sensor Óptico Laser – LDR (SOL).....	53
2.10.1 Descripción.....	53
2.10.2 Diodo Laser.....	54
2.10.3 LDR.....	55
2.11 Pulsadores.....	58
2.11.1 Introducción.....	58
2.11.2 Pulsadores.....	59
2.11.3 Resistencias de polarización.....	59
2.12 LED.....	60
2.12.1 Qué es un led?.....	60
2.12.2 Funcionamiento de un led.....	61
2.12.3 ¿Cuáles son las partes del diodo led?.....	61
CAPITULO III.....	63
INGENIERIA DEL PROYECTO.....	63
3.1 Herramientas Software.....	63

3.1.1 El compilador MikroC PRO	63
3.1.2 El simulador ISIS de Proteus	64
3.1.3 PICkit2	66
3.3 Diagramas de flujo	69
3.3.1 Diagrama de flujo del programa principal.....	69
3.3.2 Diagrama de flujo de Inicio de motores.....	71
3.3.4 Diagrama de flujo para detener motores.....	71
3.3.5 Diagrama de flujo para la alarma.....	72
3.3.6 Diagrama de flujo para leer datos por teclado matricial.....	74
3.4 Utilización de librerías de MikroPRO C for pic.....	75
3.5 Diseño en Proteus ISIS.....	76
3.6 Desarrollo del código	77
3.7 Diseño de maqueta	89
CAPITULO IV	92
ANALISIS DE COSTOS.....	92
4.1 Costos Fijos	92
CAPITULO V	93
CONCLUSIONES Y RECOMENDACIONES.....	93
5.1 Conclusiones	93
5.2 Recomendaciones	93
REFERENCIAS BIBLIOGRAFICAS.....	94
Bibliografía	94

ÍNDICE DE FIGURAS

Figura 2.1 El microcontrolador	6
Figura 2.2 Esquema general de un Microprocesador.....	9
Figura 2.3 Registros del Microcontrolador	13
Figura 2.4 Microcontrolador PIC16F877A.....	15
Figura 2.5 Diagrama en bloques PIC 16F877A	17
Figura 2.6 Diagrama de Pines PIC16F877A.....	17
Figura 2.7 Descripción de los pines PIC16F877A.....	18
Figura 2.8 Puertos del PIC16F877A	21
Figura 2.9 Display liquido LCD	22
Figura 2.10 Descripción de pines del LCD.....	23
Figura 2.11 La memoria CGRAM	24
Figura 2.12 Teclado Matricial 4x4.....	29
Figura 2.13 Diagrama interno del teclado matricial.....	31
Figura 2.14 Motor DC.....	31
Figura 2.15 Partes del motor DC	32
Figura 2.16 Especificaciones Técnicas del motor DC	35
Figura 2.17 Descripción interna del L293D.....	36
Figura 2.18 Configuración interna del L293D.....	36
Figura 2.19 Pines del L293D	37
Figura 2.20 Descripción de pines del L293D.....	37
Figura 2.21 Vista pictórica del transistor BC558 y de los miembros de su familia con indicaciones generales.....	38
Figura 2.22 Tensiones y corriente máxima de los transistores de la familia BC558.....	39

Figura 2.23 Datos técnicos principales de los transistores BC556, BC557, BC558, BC559 y BC560.....	40
Figura 2.24 Tabla con las ganancias de la serie BC55x en base a la clase (indicada con una letra al final del código).	41
Figura 2.25 Vista pictórica del transistor BC548 y de los miembros de su familia con indicaciones generales.....	42
Figura 2.26 Tensiones y corriente máxima de los transistores BC546, BC547, BC548, BC549 y BC550.....	43
Figura 2.27 Datos técnicos principales de los transistores BC546, BC547, BC548, BC549 y BC550.....	44
Figura 2.28 Tabla con las ganancias de la serie BC54x en base a la clase (indicada con una letra al final del código).	45
Figura 2.29 Configuración del transistor TIP 41	47
Figura 2.30 Cristal Oscilador	47
Figura 2.31 Osciladores de Cristal.....	49
Figura 2.32 Circuito Equivalente del cristal de cuarzo.....	50
Figura 2.33 Parlante o altavoz de 8 Ohmios	51
Figura 2.34 Diagrama de sensor Óptico Laser - LDR	54
Figura 2.35 Diodo Laser	55
Figura 2.36 LDR	55
Figura 2.37 Configuración del LDR	57
Figura 2.38 Símbolo y forma física del pulsador.....	59
Figura 2.39 Configuración de uso de los pulsadores	60
Figura 2.40 Diodos Leds.....	60
Figura 2.41 Partes del diodo Led	62

Figura 3.1 Entorno del mikroC PRO for Pic.....	64
Figura 3.2 Entorno de Proteus ISIS	65
Figura 3.3 Grabador PICKit 2.....	69
Figura 3.4 Diagrama esquemático de la Banda Transportadora	77
Figura 3.5 Maqueta de banda transportadora.....	90
Figura 3.6 Conexiones del protoboard montadas en la maqueta	90
Figura 3.7 Primera y segunda banda transportadora de la maqueta.....	91

INDICE DE TABLAS

Tabla 2.1 Características eléctricas del LCD	25
Tabla 2.2 Características mecánicas del LCD.....	26
Tabla 2.3 Índices máximos del LCD.....	26
Tabla 2.4 Conexiones del LCD.....	27
Tabla 2.5 Código de dirección de caracteres del LCD.....	28
Tabla 4.1 Costos de los componentes y material utilizado	92

CAPITULO I

1.1 Introducción

En la ciudad industrial de Viacha, se encuentra en gran cantidad este material natural llamado “arcilla” a través de este material, se puede elaborar productos en cerámica y una de ellas son los ladrillos en sus diferentes variedades que se utilizan para la construcción de muros y otras estructuras.

Al ser un material abundante en la ciudad de Viacha surgen varias empresas de cerámicas que explotan este material para realizar ladrillos y estas puedan ser comercializadas y distribuidas en nuestro país y el exterior.

Al realizar todo el proceso de producción de ladrillos se tiene que transportar para que estas lleguen a su comercialización, a través de eso se plantea un sistema el cual pueda transportar los ladrillos para que puedan ser embalados y transportados, dando así una idea de automatización para que pueda ser este proceso efectuado de manera que se reduzca el tiempo de transporte de un punto a otro.

1.2 Antecedentes

Los ladrillos son pequeñas piezas cerámicas en forma de paralelepípedo, formadas por tierras arcillosas, moldeadas, comprimidas y sometidas a una cocción. Pueden utilizarse en toda clase de construcciones por ser su forma regular y fácil su manejo.

La arcilla, en su estado natural, está compuesta de uno o, como es el caso general, varios minerales arcillosos. En esencia los minerales de arcilla son silicatos de aluminio, pero también hay presente productos hidratados de la descomposición de las rocas aluminosas y silicatadas, y otras sustancias como fragmentos de rocas, de óxidos hidratados, álcalis y materiales coloidales.

La producción de ladrillos puede llevarse a cabo de tres formas, de acuerdo a la Normas:

1. Artesanal: Ladrillo fabricado con procedimientos predominantemente manuales. El amasado o moldeado es hecho a mano. El ladrillo producido artesanalmente se caracteriza por variaciones de unidad a unidad.
2. Semi-Industrial: Es el ladrillo fabricado con procedimientos manuales, donde el proceso de moldeado se realiza con maquinaria elemental que en ciertos casos extruye, a baja presión, la pasta de arcilla. El ladrillo semi-industrial se caracteriza por presentar una superficie lisa.
3. Industrial: Es el ladrillo fabricado con maquinaria que amasa, moldea y prensa o extruye la pasta de arcilla. El ladrillo producido industrialmente se caracteriza por su uniformidad.

Los pasos básicos para la producción de ladrillos son:

Maduración

Antes de incorporar la arcilla al ciclo de producción hay que someterla a ciertos tratamientos de trituración, homogeneización y reposo al aire libre, para obtener una adecuada consistencia, secado tangente y uniformidad de las características físicas y químicas deseadas.

Pre elaboración

Después de la maduración, que se produce en la zona de acopio, sigue la fase de pre elaboración, que consiste en una serie de operaciones que tienen la finalidad de purificar y refinar la materia prima.

Depósito de materia prima procesada

A la fase de pre elaboración, le sigue el depósito de material en silos especiales en un lugar techado, donde el material se homogeneiza definitivamente tanto en apariencia como en características físico-químicas.

Humidificación

Antes de llegar a la operación de moldeo, se saca la arcilla de los silos y se lleva a un laminador refinador, y posteriormente a un mezclador humedecedor, donde se agrega agua para obtener la humedad precisa.

Moldeado

El moldeado consiste en hacer pasar la mezcla de arcilla por una boquilla al final de la estructura (extrusión). La boquilla es una plancha perforada que tiene la forma del objeto que se quiere producir.

Secado

El secado es una de las fases más delicadas del proceso de producción. De esta etapa depende, en gran parte, el buen resultado y calidad del material, más que nada en lo que respecta a la ausencia de fisuras. El secado tiene la finalidad de eliminar el agua agregada en la fase de moldeado para poder pasar a la fase de cocción.

Cocción

Se realiza en hornos de túnel, que en algunos casos pueden llegar a medir hasta 120 m de longitud, y donde la temperatura de la zona de cocción oscila entre 900 °C y 1000 °C. En el interior del horno la temperatura varía de forma continua y uniforme. El material secado se coloca en carros especiales, en paquetes estándar y es introducido por una de los extremos del túnel, saliendo por el extremo opuesto una vez que está cocido.

Almacenaje

Antes del embalaje se procede a la formación de paquetes sobre pallets, que permitirán después moverlos fácilmente con carretillas de horquilla. El proceso de embalaje consiste en envolver los paquetes con cintas de plástico o de metal, de modo que puedan ser depositados en lugares de almacenamiento hasta que sean llevados a la obra.

1.3 Planteamiento del problema

La empresa de cerámica “Avaroa” es una empresa en la ciudad de Viacha, el cual al ser una empresa semi industrial necesita de procesos automatizados el cual realice algunas tareas para realizar mayor producción, una de esas tareas es la distribución de ladrillos para que puedan ser comercializados y distribuidos, al ver la necesidad de poder transportar este producto para ser cargados a un camión para ser llevados a su destino.

Viendo este problema se optó por diseñar un sistema de banda transportadora que dé solución al transporte del lugar de almacenamiento al camión distribuidor cumpliendo con las cantidades que se necesitan.

1.4 Objetivos del proyecto

1.4.1 Objetivo General

Implementar el prototipo de una banda transportadora de ladrillos, mediante el microcontrolador PIC16F877A, para la empresa de cerámica “Avaroa”, en la ciudad de Viacha.

1.4.2 Objetivos Específicos

- Diseñar una banda transportadora utilizando el PIC16F877A.
- Implementar un sistema de conteo mediante un sensor óptico laser – LDR el cual pueda mostrar la cantidad de ladrillos contados.
- Utilizar lenguaje de programación C a través del MikroC PRO for Pic para la programación del PIC.

1.5 Justificación

Con este proyecto, se quiere dar solución al proceso final de producción de ladrillo que es transportar de un lugar a otro para su envío y distribución, con esto se quiere realizar una mayor producción en el envío del producto final ya que esto consume tiempo y desgaste de energía, los procesos de automatización en una empresa son benéficas y de gran aporte para la industrialización en la ciudad de Viacha y a la vez en nuestro país, necesitamos crear proyectos de esta clase para la industrialización de nuestro productos

y así generar mayores ingresos y comodidades y a la vez estar en competencia con países que tienen un gran avance tecnológico.

1.6 Delimitación

El área considerada para la implementación del presente proyecto está limitada a empresas en donde se requiera el transporte de ladrillos en regiones en donde se pueda extraer la materia prima para el producto.

Además, el proyecto se enfoca en el recojo y transporte del material ya terminado para la distribución de la misma, este proceso no influye en los demás procesos de producción del producto.

CAPITULO II

MARCO TEORICO

2.1 El Microcontrolador

2.1.1 Microcontrolador

El Microcontrolador es un circuito integrado que es el componente principal de una aplicación embebida. Es como una pequeña computadora que incluye sistemas para controlar elementos de entrada/salida. También incluye a un procesador y por supuesto memoria que puede guardar el programa y sus variables (flash y RAM). Funciona como una mini PC. Su función es la de automatizar procesos y procesar información.

El microcontrolador se aplica en toda clase de inventos y productos donde se requiere seguir un proceso automático dependiendo de las condiciones de distintas entradas.

2.1.2 Elementos de un Microcontrolador

Un microcontrolador al menos tendrá:

- Microprocesador.
- Periféricos (unidades de entrada/salida).
- Memoria.

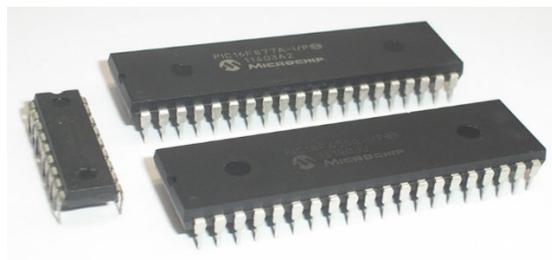


Figura 2.1 El microcontrolador

Fuente: <https://hetpro-store.com/TUTORIALES/microcontrolador/>.

2.1.3 Elementos internos en un microcontrolador

2.1.3.1 Procesador o Microprocesador.

- Un procesador incluye al menos tres elementos, ALU, unidad de control y registros.
 - **ALU.** También conocida como Unidad Aritmética y Lógica. Esta unidad está compuesta por los circuitos electrónicos digitales del tipo combinatorios (compuertas, sumadores, multiplicadores), cuya principal función es el realizar operaciones. Estas operaciones están divididas en tres tipos:
 - **Lógicas.** Como las operaciones básicas de las compuertas lógicas, como la suma lógica (OR), multiplicación lógica (AND), diferencia lógica (XOR) y negación (NOT). Una operación lógica sólo puede tener como entradas y como salidas una respuesta lógica (0 o 1). Esto dependiendo de los niveles de voltajes de una señal digital.
 - **Aritméticas.** Las operaciones aritméticas son la suma, resta, multiplicación y división. Dependiendo del procesador (8, 16, 32 o 64 bits) será la rapidez con la que se pueden hacer dichas operaciones.
 - **Misceláneas.** En estas operaciones caen todas las demás operaciones como la transferencia de bits (<<>>).
 - **Unidad de control.** La unidad de control es el conjunto de sistemas digitales secuenciales (aquellos que tienen memoria) que permiten distribuir la lógica de las señales.
 - **Registros.** Los registros son las memorias principales de los procesadores, ya que funcionan a la misma velocidad que el procesador a diferencia de otras memorias un tanto más lentas (como la RAM, FLASH

o la CACHE). Los registros están contruidos por Flip-Flops. Los Flip-Flops son circuitos digitales secuenciales.

2.1.3.2 Periféricos.

- Los periféricos son los circuitos digitales que nos permiten una interacción con el mundo «exterior» al microcontrolador. Su función es la de poder habilitar o deshabilitar las salidas digitales, leer sensores analógicos, comunicación con terminales digitales o sacar señales analógicas de una conversión digital.
 - **Puertos de entrada/salida** paralelos. Los puertos están relacionados al tamaño del procesador, es decir que un puerto de 8 bits es porque el procesador es de 8 bits. Un procesador de 64 bits, tiene la capacidad de tener un puerto de 64 bits.
 - **Puertos seriales.** Nos permiten transformar la información digital paralela (bytes de información) en tramas que se pueden transferir por una o varias líneas de comunicación. Existen, por ejemplo: puerto serial, i2c, SPI, USB, CAN, etc.
 - **Periféricos analógicos.** Como los que convierten señales analógicas a digitales (ADC) o señales digitales a analógicas (DAC) o comparadores analógicos.

2.1.3.3 Tipos de Memoria.

La memoria está dividida en tres. La memoria para el programa (FLASH), la memoria para los datos o variables del programa (RAM) y la memoria para configuraciones o no volátil (EEPROM).

2.1.4 Diferencia entre un microcontrolador y un microprocesador

Un microprocesador es un sistema programable abierto (configuración variable) con el que puede construirse un computador con las características que se desee, acoplándole los módulos necesarios.

- Requiere de varios circuitos integrados para crear un sistema completo: ROM, RAM, Puertos I/O, Periféricos.
- Tiene alta capacidad de procesado

- Memorias masivas de programas y datos
- De propósito general
- Alto consumo de energía
- Costo elevado

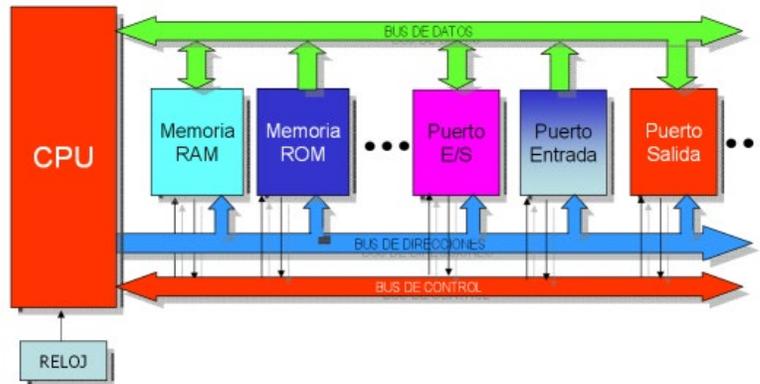


Figura 2.2 Esquema general de un Microprocesador.

Fuente: Guachalla, R. (2015). *Laboratorio de Microprocesadores I*. La Paz.

Un microcontrolador es un sistema programable cerrado que contiene un computador completo y de prestaciones limitadas que no se pueden modificar.

- Incorpora todos los circuitos integrados para crear un sistema completo: ROM, RAM, Puertos I/O, Periféricos.
- Tiene baja capacidad de procesamiento
- Memorias reducidas para programas y datos
- De propósito específico: Control
- Bajo consumo de energía
- Costo reducido

2.1.5 Microcontrolador incluye un Microprocesador

Para el Microcontrolador, un Microprocesador es el conjunto de tres sistemas digitales: ALU, unidad de control y registros. Los microprocesadores fueron los padres de los microcontroladores. Uno de los primeros procesadores comerciales que se enseñaban en las universidades era el Z80.

El procesador Z80 requiere de componentes adicional para poder funcionar en alguna aplicación. Por ejemplo, no tiene RAM o FLASH. Estas dos memorias sirven para guardar el contenido de los datos del programa y al programa en sí mismo respectivamente.

Para ensamblar un circuito mínimo, se tenía que conectar al Z80: memoria externa y controladores para poder tener algún puerto paralelo. No se diga un puerto serial, se tenía que diseñar mediante electrónica reconfigurable (GALs, Compuertas logicas, Flip-Flops, Multiplexores, etc.).

2.1.6 Microcontrolador y sus Periféricos

En el Microcontrolador los Periféricos son módulos electrónicos digitales que se encuentran embebidos o empotrados en el microcontrolador. Permiten controlar distintas funciones de los puertos. Por ejemplo, conexión de puertos de entrada/salida en paralelo. Adicionalmente, los periféricos pueden cumplir con funciones adicionales o secundarias. Dichas funciones se pueden configurar mediante registros de propósito específico.

2.1.6.1 Periféricos de comunicación

- UART. Unidad de Recepción y Transmisión Serial Asíncrona. Consiste de dos cables llamados RX y Tx que sirven para enviar mensajes binarios con otros sistemas digitales.
- I2C. Puerto de comunicación serial síncrono. Permite enviar y recibir datos mensajes con hasta 127 dispositivos conectados al mismo de bus que consiste en dos cables: SDA y SCL. Serial Data y Serial CLock respectivamente.
- SPI. Serial Peripheral Interface. Interfaz Serial Periférica. Es un puerto de comunicación serial síncrona. Su función es la de enviar información binaria mediante 4 cables de conexión.
 - MISO – Master Input Slave Output. Entrada del Maestro y Salida del Esclavo.

- MOSI – Master Output Slave Input. Salida del Maestro Entrada del Esclavo.
- SCK. Serial Clock. Reloj Serial.
- /CS ó /SS. Chip Select ó Slave Select. Selector del Chip o Selector del Esclavo. Permite activar un esclavo.
- Puerto Paralelo. Se controlan por medio de registros. Generalmente se pueden controlar mediante tres registros.
 - Registro de dirección para configurar entrada ó salida.
 - Registro de estado para la entrada. Nos indica el nivel lógico a la entrada del pin de cada puerto, cuando esté está configurado como entrada.
 - Registro de configuración del estado lógico de salida. Permite escribir 1's y 0's a los pines del puerto, cuando esté está configurado como salida.

2.1.6.2 Periféricos de adecuación de señales

- **ADC.** Convertidor de Señales Analógicas a Digitales. Permiten generar una representación binaria para la magnitud de una señal analógica.
- **DAC.** Digital to Analog Converter. Lo opuesto a un DAC. También es ideal cuando se quiere generar audio. Una tarjeta de audio no es más que un DAC de muy buena calidad y muy rápido.

2.1.6.3 Periféricos generales

- **Timer/Contador.** Son circuitos electrónicos digitales y secuenciales. Es decir que funcionan en flancos de una señal de reloj que oscila a una frecuencia determinada. Su función principal es la de contar eventos, ya sea internos (timer) o externos (contador), a una frecuencia que nos permite, por ejemplo, contar el tiempo transcurrido de un proceso.

- Controlador de Interrupciones. Son dispositivos que permiten alterar la secuencia de ejecución del procesador. Esto logra que se pueda interrumpir al procesador para atender el llamado de dispositivos electrónicos de baja velocidad. Por ejemplo, teclados o sensores.
- Controlador de Oscilador. Permite dividir y/o multiplicar a osciladores internos o externos para poder hacer funcionar a la parte secuencial de la electrónica del microcontrolador.

2.1.7 Microcontrolador y sus memorias

La memoria de los microcontroladores puede dividirse en cuatro tipos distintos: para el programa, generalmente es una memoria interna del tipo FLASH. También para las variables de los programas se llama RAM. Los registros de propósito general que utiliza el procesador para guardar los resultados de las operaciones, así como los datos que se traen y llevan a la RAM y finalmente la memoria externa que se utiliza para guardar mediciones y/o datos de calibración.

2.1.7.1 Memoria de programa – FLASH

La memoria de programa es la que se utiliza cuando estamos programando a nuestra aplicación. Cada instrucción del lenguaje ensamblador o de lenguaje C o de un lenguaje de alto nivel, es convertida a instrucciones máquina que requieren de un tamaño particular de bits para ser guardadas. Estos bits o instrucciones se guardan en la FLASH. Si tenemos poca FLASH, entonces sólo podremos hacer programas pequeños.

2.1.7.2 Memoria de datos – RAM

La memoria de datos se utiliza cada vez que agregamos una variable nuestro programa. Como la variable seguramente se utilizará como entrada para un proceso o calculo, está requiere de estar disponible a una velocidad relativamente rápida. Este tipo de información se guarda en la RAM – Random Access Memory ó Memoria de Acceso Aleatorio por sus siglas en ingles.

Si tenemos poca RAM, nuestra aplicación no podrá tener muchas variables.

2.1.8 Microcontrolador y sus Registros

Los registros son las memorias digitales más rápidas. Se construyen con Flip-Flops y generalmente funcionan a una velocidad cercana a la del procesador. En algunos procesadores, también incluyen un tipo de memoria llamada CACHE. Está no puede guardar operaciones y sólo es un puente entre el procesador y la memoria principal. Por ejemplo, se puede ver la estructura del registro 74LS377 el cual puede funcionar como un registro (memoria).

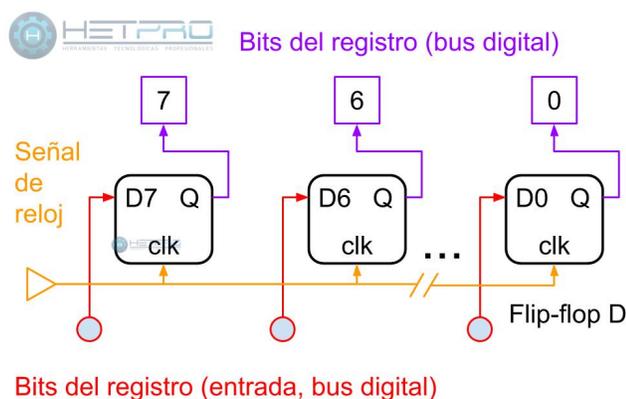


Figura 2.3 Registros del Microcontrolador

Fuente: <https://hetpro-store.com/TUTORIALES/microcontrolador/>

2.1.9 Memoria externa

La memoria externa se suele utilizar cuando queremos guardar parte de la información que estamos recolectando y/o procesando. Se suelen usar también para procesos de calibración. Existen dos distintos tipos de memorias no volátiles que pueden ser usadas externamente para los microcontroladores.

- EEPROM – Memoria Eléctricamente Borrable de Solo Lectura. Ojo, también se pueden programar y escribir.
- FLASH – Como la que tienen internamente los microcontroladores. Pero estas suelen funcionar a una velocidad mucho más lenta a través de protocolos seriales como I2C, Serial o SPI.

2.1.10 Como se programa a un microcontrolador

Un microcontrolador se programa en lenguaje ensamblador. Entonces cada microcontrolador, tiene su propio conjunto de instrucciones. Las instrucciones se dividen de acuerdo de las operaciones que realizan en:

- Aritméticas.
- Lógicas.
- Transferencia de bits.
- Misceláneas.

Por lo general la curva de aprendizaje del lenguaje ensamblador es muy grande. Para el ciclo de desarrollo de las aplicaciones para sistemas embebidos, esto es mucho tiempo. Por lo tanto, se utilizan lenguajes de un nivel superior como basic o C/C++. Además, estos lenguajes se traducen al lenguaje ensamblador mediante compiladores. Por ejemplo, el compilador C18 permite traducir el código C al lenguaje máquina de los microcontroladores PIC18XXX.

Para poder programar a un microcontrolador se requieren tres cosas:

- Compilador. Traduce lenguaje como el C y C++ al lenguaje ensamblador.
- IDE. Integrated Development Environment. Entorno de Desarrollo Integrado. Es donde mediante el compilador, se crea y compilan los programas.
- Programador. Cada fabricante y arquitectura tiene su propio programador. Existen programadores llamados Universales que pueden programar distintas arquitecturas.

2.2 El PIC 16f877a

2.2.1 Introducción

El microcontrolador PIC16F877A es un dispositivo programable que se compone de una computadora digital, una unidad de memoria de datos, una unidad de memoria de programa y puertos de entrada/salida en un circuito integrado, funciona como un

controlador de periféricos en un sistema mínimo. El microcontrolador depende de una alimentación de al menos 5V y 0V en sus entradas de Vdd y Vss respectivamente para su operación, requiere de una señal de reloj que le indique la frecuencia de trabajo, está señal la introducimos a través de un oscilador de cristal de cuarzo, y una alimentación al pin MCLR, que es un pin de reset que activa al microcontrolador. El funcionamiento del microcontrolador está determinado por un programa almacenado en su memoria Flash ROM y puede programarse más de una vez para cambiar su estado y su comportamiento, lo que lo convierte al microcontrolador en una pieza esencial en el rápido desarrollo de aplicaciones electrónicas.



Figura 2.4 Microcontrolador PIC16F877A

Fuente: <https://mvelectronica.com/producto/microcontrolador-pic16f877a>

2.2.2 Descripción del PIC16F877A

Este potente microcontrolador CMOS de 8 bits basado en FLASH, pero fácil de programar empaqueta la potente arquitectura PIC de Microchip en un paquete de 40 y es compatible con los dispositivos PIC16C5X, PIC12CXXX y PIC16C7X. El PIC16F877A cuenta con 256 bytes de memoria de datos EEPROM, Auto programación, ICD, 2 comparadores, 8 canales de convertidor analógico a digital (A/D) de 10 bits, 2 funciones de captura/comparación/PWM, el puerto serie síncrono se puede configurar como interfaz periférica serial de 3 hilos (SPI) o bus de circuito integrado de 2 hilos (I2C) y transmisor de receptor asincrónico universal (USART).

2.2.3 Características del PIC16F877A

- 100.000 ciclos de borrado/escritura Enhanced Flash memoria del programa típica
- 1.000.000 de borrado/ciclo de escritura Datos EEPROM memoria típica

- Retención EEPROM de datos > 40 años
- Auto-reprogramable bajo control de software
- Programación serie en circuito (ICSP) a través de dos pines
- Programación serie de 5V in-circuit de un solo suministro
- Temporizador watchdog (WDT) con su propio RC en chip oscilador para un funcionamiento fiable
- Protección programable del código
- Ahorro de energía Modo de suspensión
- Opciones de oscilador seleccionables
- Depuración en circuito (ICD) a través de dos pines

Otra característica de los PICs es el manejo de los bancos de registros. En línea general, los registros se clasifican como de uso general (GPR) y de uso específico o de funciones especiales (SFR).

- Los registros de uso general pueden ser usados directamente por el usuario, sin existir restricciones. Pueden servir para almacenar resultados que se reciben desde el registro W (acumulador), datos que provienen de las puertas de entradas, etc.
- Los registros de uso específicos no pueden ser usados directamente por el usuario. Estos registros controlan prácticamente todo el funcionamiento del microcontrolador, pues toda la configuración necesaria para funcionamiento del microcontrolador es hecha a través de algún tipo de SFR.

2.2.4 Diagrama en Bloques del PIC16F877A

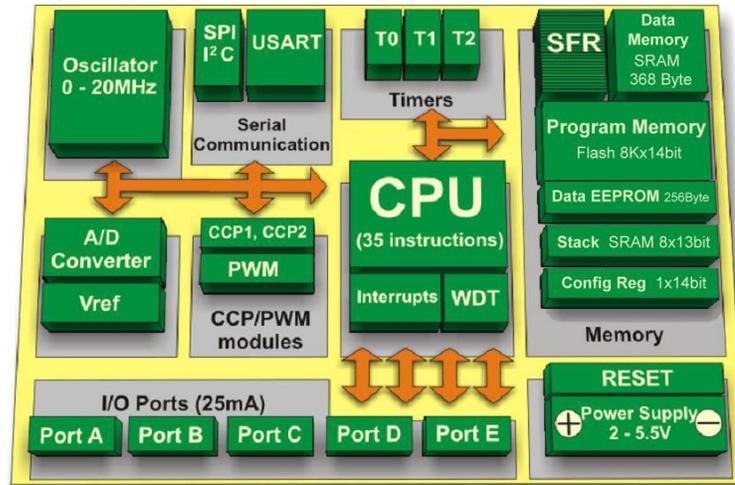


Figura 2.5 Diagrama en bloques PIC 16F877A

Fuente: Guachalla, R. (2015). *Laboratorio de Microprocesadores I.*

2.2.5 Descripción de los pines del PIC16F877A

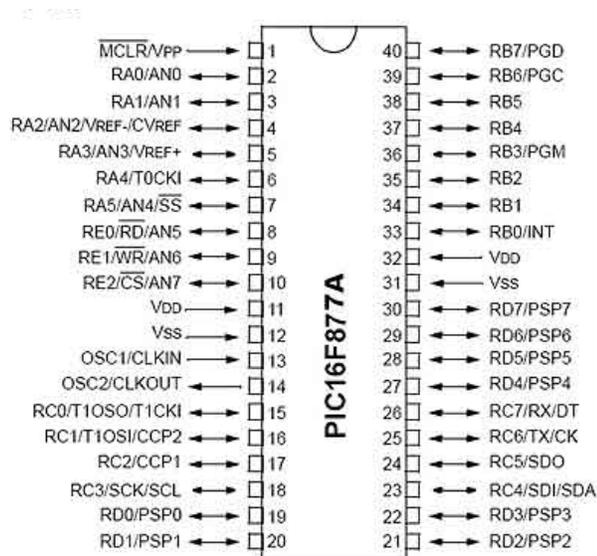


Figura 2.6 Diagrama de Pines PIC16F877A

Fuente: Guachalla, R. (2015). *Laboratorio de Microprocesadores I.*

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

Figura 2.7 Descripción de los pines PIC16F877A

Fuente: Guachalla, R. (2015). *Laboratorio de Microprocesadores I*.

2.2.6 Organización de la memoria

Existen cinco áreas de memoria dentro de un PIC16F877A:

- La memoria de Programa o Código Flash ROM 8Kx14 bit
- La memoria de Pila (Stack) SRAM 8x13 bit
- La memoria de Datos EEPROM 256 Bytes
- La memoria de Datos o Variables SRAM 368 Bytes
- La memoria de Configuración Flash ROM 1x14 bit

2.2.7 Memoria de Programa Flash ROM 8Kx14

Memoria destinada para almacenamiento del código ejecutable en el microcontrolador

- Capacidad de 8 Kilo words de (14 bits) tipo Flash ROM dividida en 4 páginas de 2Kx14
- Contador de programa PC de 13 bit con capacidad de direccionar programas de hasta 8k instrucciones de 14 bits
- La ubicación del vector de RESET (0000h) es el valor del contador de programa cuando la máquina se inicializa
- Vector de interrupción ubicado en 04H

2.2.8 Memoria de Pila 8x13 SRAM 8x13

Memoria destinada para el almacenamiento del PC durante las subrutinas e interrupciones

- Capacidad de 8 words de 13 bits tipo SRAM (8 niveles de almacenamiento)
- Es independiente de la memoria de datos y de programa
- Operación de “Buffer circular”: Si se ingresan 8 datos, se llena el Stack, un noveno dato que ingrese desplaza los contenidos de la pila y elimina el primer dato que se ingresó a la pila.

2.2.9 Memoria de Datos EEPROM 256 Bytes

Memoria destinada para almacenamiento secundario permanente de datos y variables

- Capacidad de 256 bytes de memoria tipo EEPROM
- Se utiliza de manera indirecta a través de los registros EEADR y EEDATA, EEWR, EERD
- El registro EEADR guarda la dirección a leer/escribir, en el registro EEDATA se escriben/leen los datos.

2.2.10 Memoria de Datos SRAM 368 Bytes

Memoria destinada para almacenamiento temporal de datos y variables

- Capacidad de 368 Bytes de memoria tipo SRAM
- Dividida en 4 bancos seleccionados con los bits RP0 y RP1 del registro “Status”
- Cada banco es de 128 bytes
- Las ubicaciones más bajas son reservadas para registros de control/configuración SFR (Registros de Función Especial)
- En las posiciones superiores de cada banco se constituyen los registros de propósito general GPR (General Purpose Registers)

2.2.11 Memoria de Configuración Flash ROM 1x14 bit

Ubicada en la dirección 0x2007, permite configurar el modo de funcionamiento del microcontrolador

2.2.12 Puertos Entrada y Salida

Los pines de entrada/salida de este microcontrolador están organizados en cinco puertos, el puerto A con 6 líneas, el puerto B con 8 líneas, el puerto C con 8 líneas, el puerto D con 8 líneas y el puerto E con 3 líneas.

- Al aplicar un Reset, todos los puertos se inicializan configurados como Entrada
- Los registros TRISA, TRISB, TRISC, TRISD y TRISE configuran la dirección de los puertos.
- Cada pin de esos puertos se puede configurar como entrada o como salida independiente programando los registros TRISx. Al poner un 0 en un bit de TRISx se configura el pin del puerto correspondiente como Salida y al poner un 1 se configura el pin de puerto correspondiente como Entrada.
- Los pines del puerto A y del puerto E pueden trabajar como entradas para el convertidor Análogo a Digital interno, es decir, allí se podría conectar una señal proveniente de un sensor o de un circuito analógico para que el microcontrolador

la convierta en su equivalente digital y pueda realizar algún proceso de control o de instrumentación digital.

- El pin RB0/INT se puede configurar por software para que funcione como interrupción externa, para configurarlo se utilizan unos bits de los registros que controlan las interrupciones.
- El pin RA4/TOCKI del puerto A puede ser configurado como un pin de entrada/salida o como entrada del temporizador/contador. Cuando este pin se programa como entrada digital, funciona como un disparador de Schmitt (Schmitt trigger), puede reconocer señales un poco distorsionadas y llevarlas a niveles lógicos (cero y cinco voltios). Cuando se usa como salida digital se comporta como colector abierto (open collector), por lo tanto, se debe poner una resistencia de pull-up (resistencia externa conectada a un nivel de cinco voltios). Como salida, la lógica es inversa: un "0" escrito al pin del puerto entrega en el pin un "1" lógico.
- El puerto E puede controlar la conexión en modo microprocesador con otros dispositivos utilizando las líneas RD (read), WR (write) y CS (chip select). En este modo el puerto D funciona como un bus de datos de 8 bits (pines PSP).

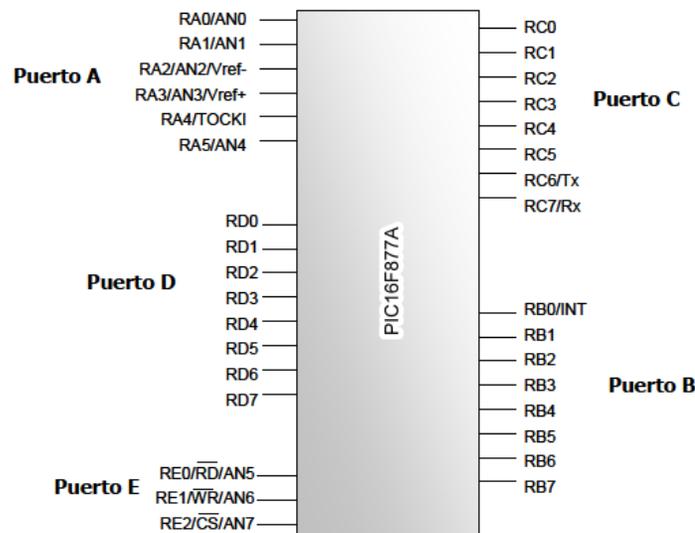


Figura 2.8 Puertos del PIC16F877A

Fuente: Guachalla, R. (2015). *Laboratorio de Microprocesadores I*.

2.3 Display Liquido Lcd

2.3.1 ¿Que es un LCD?

El LCD (Liquid Crystal Display) o pantalla de cristal líquido es un dispositivo empleado para la visualización de contenidos o información de una forma gráfica, mediante caracteres, símbolos o pequeños dibujos dependiendo del modelo. Está gobernado por un microcontrolador el cual dirige todo su funcionamiento.

En este caso vamos a emplear un LCD de 16x2, esto quiere decir que dispone de 2 filas de 16 caracteres cada una. Los píxeles de cada símbolo o carácter, varían en función de cada modelo.

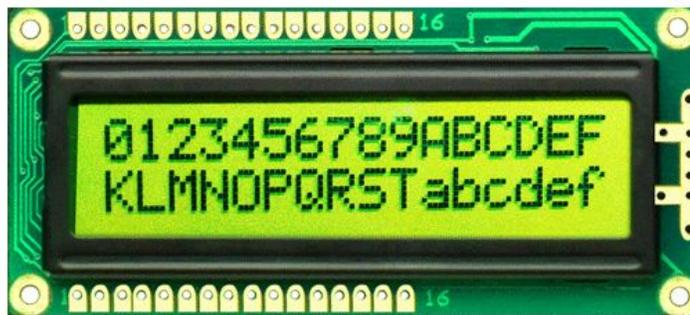


Figura 2.9 Display liquido LCD

Fuente: <https://ingenieriaelectronica.org/display-de-cristal-liquido-lcd-definicion-caracteristicas-y-conexion/>

2.3.2 ¿Como es su conexión?

En la siguiente imagen de Proteus se puede observar la estructura de sus pines.

Lo podemos dividir en los Pines de alimentación, pines de control y los pines del bus de datos bidireccional. Por lo general podemos encontrar además en su estructura los pines de ánodo de led back light y cátodo de led back light.

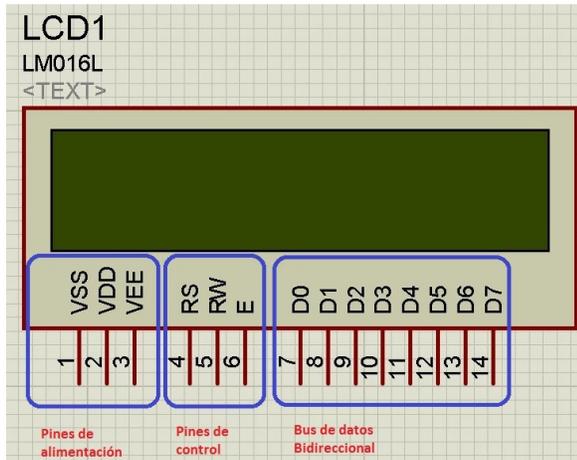


Figura 2.10 Descripción de pines del LCD

Fuente: <https://circuitdigest.com/article/16x2-lcd-display-module-pinout-datasheet>

Pines de alimentación:

Vss: Gnd

Vdd: +5voltios.

Vee: corresponde al pin de contraste, lo regularemos con un potenciómetro de 10K conectado a Vdd.

Pines de control:

RS: Corresponde al pin de selección de registro de control de datos (0) o registro de datos (1). Es decir, el pin RS funciona paralelamente a los pines del bus de datos. Cuando RS es 0 el dato presente en el bus pertenece a un registro de control/instrucción. y cuando RS es 1 el dato presente en el bus de datos pertenece a un registro de datos o un carácter.

RW: Corresponde al pin de Escritura (0) o de Lectura (1). Nos permite escribir un dato en la pantalla o leer un dato desde la pantalla.

E: Corresponde al pin Enable o de habilitación. Si E(0) esto quiere decir que el LCD no está activado para recibir datos, pero si E(1) se encuentra activo y podemos escribir o leer desde el LCD.

Pines de Bus de datos:

El Bus de datos bidireccional comprende desde los pines D0 a D7. Para realizar la comunicación con el LCD podemos hacerlo utilizando los 8 bits del bus de datos (D0 a D7) o empleando los 4 bits más significativos del bus de datos (D4 a D7). En este caso vamos a explicar la comunicación con el bus de 4 bits.

2.3.3 ¿DDRAM y CGROM?

Son las dos zonas de memoria del LCD.

La memoria DDRAM (Data Display Ram): corresponde a una zona de memoria donde se almacenan los caracteres que se van a representar en pantalla. Es decir, es la memoria donde se almacenan los caracteres a mostrar con su correspondiente posición.

La memoria CGROM es una memoria interna donde se almacena una tabla con los caracteres que podemos visualizar en el lcd. En la imagen podemos ver un ejemplo de la tabla con un contenido de 192 caracteres.

		4 bits más altos de la dirección																
		0000	0001	0010	0011	0100	0101	0110	0111	1000	1001	1010	1011	1100	1101	1110	1111	
4 bits más bajos de la dirección	XXXX 0000	CG RAM (1)	00P`P														-ヲΞαρ	
	XXXX 0001	CG RAM (2)	!1AQaα															。アチ4äq
	XXXX 0010	CG RAM (3)	"2BRbr															「イウ×ρθ
	XXXX 0011	CG RAM (4)	#3CScs															」ウチεα*
	XXXX 0100	CG RAM (5)	\$4DTdt															、Iトtμα
	XXXX 0101	CG RAM (6)	%5EUeu															。オチ1εÜ
	XXXX 0110	CG RAM (7)	&6FVfv															ヲカニヨρΣ
	XXXX 0111	CG RAM (8)	?7GWgw															ヲチヌōγπ
	XXXX 1000	CG RAM (9)	(8HXhx															イウチリル
	XXXX 1001	CG RAM (0))9IYiy															ちケリル'γ
	XXXX 1010	CG RAM (1)	*:JZjz															エコんレjチ
	XXXX 1011	CG RAM (2)	+;K[k(オサヒロ*チ
	XXXX 1100	CG RAM (3)	,<L¥l															チシフワチチ
	XXXX 1101	CG RAM (4)	-=M]m)															ユズ'ニチチ
	XXXX 1110	CG RAM (5)	.>N^n^															オチチ'ん
	XXXX 1111	CG RAM (6)	/?O_Lo+															ウツマ°ö

Figura 2.11 La memoria CGRAM

Fuente: <https://circuitdigest.com/article/16x2-lcd-display-module-pinout-datasheet>

2.3.4 Características y especificaciones técnicas de la pantalla LCD HD44780

- El voltaje de funcionamiento es de 4,7 V a 5,3 V.
- El consumo de corriente es de 1 mA sin retroiluminación.
- Módulo de pantalla LCD alfanumérico, lo que significa que puede mostrar alfabetos y números
- Consta de dos filas y cada fila puede imprimir 16 caracteres.
- Cada personaje está construido por un cuadro de 5×8 píxeles
- Puede funcionar tanto en modo de 8 bits como de 4 bits
- También puede mostrar cualquier carácter personalizado generado
- Disponible en retroiluminación verde y azul.

2.3.5 Características eléctricas

Artículo	Símbolo	condición	mínimo	Escribe	máx.	Unidad
Voltaje de suministro para LCD	VDD-VO	Ta=0°C, Ta=25°C, Ta=50°C		4.5		EN
Entrada de alto voltaje	VIH		2.2		VDD	EN
Bajo voltaje de entrada	VOLUNTAD		0.3		0.6	EN
Salida de alto voltaje	VOH		2.4			EN
Salida de bajo voltaje	VOLUMEN				0.4	EN
Corriente de suministro	IDD	VDD=5V		2.0	4.0	mA

Tabla 2.1 Características eléctricas del LCD

Fuente: <https://www.futurlec.com/LED/LCD16X2.shtml>

2.3.6 Características mecánicas

Artículo	Dimensiones	Unidad
Tamaño de pantalla LCD	80x36	milímetro
Área de visualización	64x16	milímetro
Tamaño de punto	0,55x0,66	milímetro
Paso de punto	0,60x0,70	milímetro
Tamaño de caracteres	2,96x5,56	milímetro
Tono de personaje	3,55x5,94	milímetro
Espesor LCD	9.5	milímetro

Tabla 2.2 Características mecánicas del LCD

Fuente: <https://www.futurlec.com/LED/LCD16X2.shtml>

2.3.7 Índices absolutos máximos

Artículo	Símbolo	mínimo	Escribe	máx.	Unidades
Temperatura de funcionamiento.	Parte superior	0		+50	°C
Temperatura de almacenamiento.	Prueba	-20		+70	°C
Voltaje de entrada	Nosotros	vs		Vdd	EN
Tensión de alimentación para lógica	Vdd-Vss	2.7	5.0	5.5	EN
Voltaje de suministro para LCD	Vdd-Vo	3.0		13.0	EN

Tabla 2.3 Índices máximos del LCD

Fuente: <https://www.futurlec.com/LED/LCD16X2.shtml>

2.3.8 Conexiones de clavijas de interfaz

Pin No	Símbolo	Nivel	Función
1	vs		TIERRA(0V)
2	Vdd		Vcc (+5V±5%)
3	vo		Ajuste de contraste
4	RS	A/B	Registrarse Seleccionar
5	L/E	A/B	Leer escribir
6	Y	A/B	Habilitar señal
7	DB0	A/B	Bit de datos 0
8	DB1	A/B	Bit de datos 1
9	DB2	A/B	Bit de datos 2
10	DB3	A/B	Bit de datos 3
11	DB4	A/B	Bit de datos 4
12	DB5	A/B	Bit de datos 5
13	DB6	A/B	Bit de datos 6
14	DB7	A/B	Bit de datos 7
15	CAROLINA DEL NORTE		No conectado
16	CAROLINA DEL NORTE		No conectado

Tabla 2.4 Conexiones del LCD

Fuente: <https://www.futurlec.com/LED/LCD16X2.shtml>

2.3.9 Mostrar código de dirección de caracteres

Posición de visualización	1	2	3	4	5	6	7	8
Dirección de DD RAM - Fila 1	80	81	82	83	84	85	86	87
Dirección de DD RAM - Fila 2	C0	C1	C2	C3	C4	C5	C6	C7

Posición de visualización	9	10	11	12	13	14	15	16
Dirección de DD RAM - Fila 1	88	89	8A	8B	8C	8D	8E	8F
Dirección de DD RAM - Fila 2	C8	C9	QUE	CB	CC	CD	ESTE	FC

Tabla 2.5 Código de dirección de caracteres del LCD

Fuente: <https://www.futurlec.com/LED/LCD16X2.shtml>

2.4 Teclado Matricial

2.4.1 Descripción

El Teclado matricial de botones plásticos formado por 4 filas y 4 columnas para un total de 16 teclas permite agregar una entrada de usuario a tus proyectos. El teclado es de tipo membrana, por lo que entre sus ventajas se encuentra el poco espacio que requiere para ser instalado. Posee una cubierta adhesiva y un cable flexible de conexión. Puede ser conectado a cualquier microcontrolador o tarjetas de desarrollo como Arduino.

El teclado matricial 4x4 está formado por una matriz de pulsadores dispuestos en filas (L1, L2, L3, L4) y columnas (C1, C2, C3, C4), con la intención de reducir el número de pines necesarios para su conexión. Las 16 teclas necesitan sólo 8 pines del microcontrolador en lugar de los 16 pines que se requerirían para la conexión de 16 teclas independientes. Para poder leer que tecla ha sido pulsada se debe de utilizar una técnica de barrido y no solo leer un pin de microcontrolador.

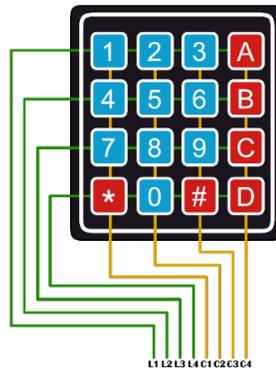


Figura 2.12 Teclado Matricial 4x4

Fuente: <https://naylampmechatronics.com/interfaz-de-usuario/19-teclado-matricial-4x4-tipo-membrana.html>

2.4.2 Especificaciones Técnicas

- 16 botones con organización matricial (4 filas x 4 columnas)
- Teclado tipo membrana
- Mayor resistencia al agua y al polvo
- Auto adhesivo en la parte de atrás
- Tiempo de rebote (Bounce time): ≤ 5 ms
- Máximo voltaje operativo: 24 V DC
- Máxima corriente operativa: 30 mA
- Resistencia de aislamiento: $100 \text{ M}\Omega$ (@ 100 V)
- Voltaje que soporta el dieléctrico: 250 VRMS (@ 60Hz, por 1 min)
- Expectativa de vida: 1.000.000 de operaciones
- Dimensiones teclado: 69*77mm
- Cable de cinta plana de 8.5 cm de largo aprox. (incluido el conector)
- Conector tipo DuPont hembra de una fila y 8 contactos con separación estándar 0.1" (2.54mm)
- Temperatura de operación: 0 a 50 °C

2.4.3 Funcionamiento y programación de un Teclado Matricial

Para programar un Teclado Matricial o un Keypad en un PIC o en cualquier otro microcontrolador, basta con seguir en orden la siguiente secuencia:

1. Inicialmente conectamos el teclado matricial en un puerto configurando los PINES como DIGITALES. Para eso será importante poder identificar cuáles son las columnas y las filas del teclado.
2. Las Filas del Teclado estarán conectadas en PINES DIGITALES configurados como SALIDAS.
3. Las Columnas del Teclado estarán conectadas en PINES DIGITALES configurados como ENTRADAS y con las Resistencias PULLUP activadas (por lo tanto, estas entradas siempre estarán recibiendo un 1 lógico, si ningún botón es presionado).

Si el puerto NO posee resistencias PULLUP éstas deben ser agregadas físicamente (resistencias conectadas a las columnas y a +5v)

4. Configuramos TODAS las SALIDAS (Filas) en 1 lógico o 5v, es decir vamos a dejarlas encendidas.
5. Aplicamos el concepto de la MULTIPLEXACIÓN: Aquí vamos a mandar un 0 lógico por cada fila y vamos a leer todas las columnas, si se detecta que alguna columna recibió el cero lógico, indica que el botón que comparte la fila y la columna fue presionado, en caso contrario vuelvo a colocar la fila en 1 lógico y verifico la siguiente fila.

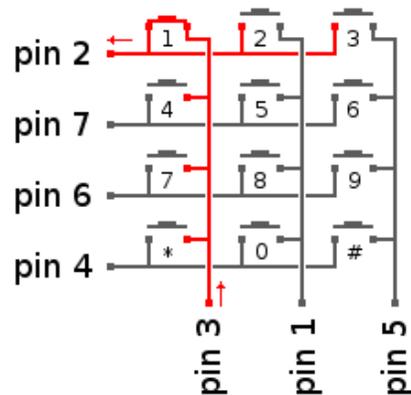


Figura 2.13 Diagrama interno del teclado matricial

Fuente: <https://controlautomaticoeducacion.com/microcontroladores-pic/teclado-matricial/>

2.5 Motores DC

2.5.1 Descripción

Un motor de corriente continua convierte la energía eléctrica en mecánica. Se compone de dos partes: el estator y el rotor. El estator es la parte mecánica del motor donde están los polos del imán. El rotor es la parte móvil del motor con devanado y un núcleo, al que llega la corriente a través de las escobillas. Si queremos cambiar el sentido de giro del rotor, tenemos que cambiar el sentido de la corriente que le proporcionamos al rotor, es decir, basta con invertir la polaridad de la pila o batería.



Figura 2.14 Motor DC

Fuente: <https://naylampmechatronics.com/motores-dc/606-motor-dc-con-caja-reductora-6v-200rpm.html>

El motor CC o motor DC es un motor de corriente continua o directa y su propiedad es la conversión de energía eléctrica a energía mecánica. Es provocado por un movimiento rotatorio, gracias a la acción de un campo magnético.

Normalmente un motor se compone de dos partes: el estátor y el rotor.

- Estátor. Da soporte mecánico y contiene los polos de la máquina, pueden ser un devanado de hilo de cobre sobre un núcleo de hierro, o imanes permanentes. Cubre el rotor.
- Rotor. Con forma cilíndrica (normalmente), también devanado y con núcleo es alimentado con corriente directa a través de delgas, que están en contacto alternante con escobillas fijas (también llamadas carbones).

El inconveniente encontrado es el mantenimiento, debido principalmente al desgaste que sufren las escobillas al entrar en contacto con las delgas.

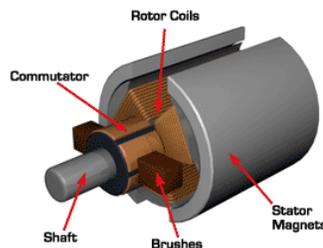


Figura 2.15 Partes del motor DC

Fuente: <https://www.prometec.net/motorcc/>

2.5.2 Funcionamiento

Se basa en la repulsión que ejercen los polos magnéticos de un imán permanente de acuerdo con la Ley de Lorentz, interactúan con los polos magnéticos de un electroimán que se encuentra montado en el rotor.

Cuando la corriente eléctrica circula por la bobina de este electroimán giratorio, el campo electromagnético que se genera interactúa con el campo magnético del imán

permanente o devanado. Si los polos del imán permanente o devanado y los polos del electroimán giratorio coinciden, se produce un rechazo y un torque magnético (par de fuerza que provoca que el rotor rompa la inercia y comience a girar sobre su eje) en sentido a la forma que se encuentre conectado el motor al circuito o la pila.

2.5.3 Variaciones en el diseño del motor

Los motores de corriente continua se construyen con rotores bobinados, y con estatores bobinados o de imanes permanentes. Además, existen muchos tipos de motores especiales, como por ejemplo los motores sin escobillas, los servomotores y los motores paso a paso, que se fabrican utilizando un motor de corriente continua como base.

Motores con estátor bobinado

Si el estátor es bobinado, existen distintas configuraciones posibles para conectar los dos bobinados de la máquina:

- Motor serie o motor de excitación en serie: el devanado de estátor y el devanado de rotor se conectan en serie.
- Motor shunt o de excitación en paralelo: el devanado de estátor y de rotor se conectan en paralelo.
- Motor de excitación compuesta o motor compound: se utiliza una combinación de ambas configuraciones.

2.5.4 Motores de imán permanente

Los motores de imán permanente tienen pocas ventajas de rendimiento frente a los motores síncronos de corriente continua de tipo excitado y han llegado a ser el predominante en las aplicaciones de potencia fraccionaria. Son más pequeños, más ligeros, más eficaces y fiables que otras máquinas eléctricas alimentadas individualmente.

2.5.5 Motor eléctrico compound.

Rotor de una pequeña máquina de corriente directa de 12 V, con imanes permanentes, de dos polos, cinco devanados, cinco delgas y dos escobillas.

Conmutación de una bobina (A) de un devanado en anillo. La bobina A está conectada a las delgas 1 y 2. La escobilla © está situada sobre la línea neutra y tiene igual ancho que las escobillas (en la realidad una escobilla suele ser más ancha y contacta con varias delgas simultáneamente). Se supone que sólo hay dos escobillas por lo que la corriente que circula por una escobilla es la corriente total del inducido I_i . En la Fig. (a) comienza la conmutación de la bobina A y en (c) termina. En la Fig. (b) se muestra un instante intermedio durante la conmutación.

2.5.6 Motores sin escobillas

Los motores de corriente directa sin escobillas están diseñados para conmutar la tensión en sus devanados, sin sufrir desgaste mecánico. Para este efecto utilizan controladores digitales y sensores de posición. Estos motores son frecuentemente utilizados en aplicaciones de baja potencia, por ejemplo, en los ventiladores de computadoras.

2.5.7 Ventajas

Algunas de las ventajas de utilizar este tipo de **motor DC** son:

1. Tiene un par de arranque elevado, es decir, al iniciar su giro poseen una elevada fuerza para mover la carga. Tengamos en cuenta que el torque y la corriente consumida por el motor son directamente proporcionales, pero el torque y la velocidad son inversamente proporcionales, a mayor velocidad, menor torque o fuerza.
2. Es reversible, o sea, que podemos invertir el sentido del giro de su eje con solo invertir la polaridad de su conexión. Aquí es donde cobra importancia una configuración de transistores conocida como Puente H.

3. Podemos controlar su velocidad fácil y suavemente. La velocidad aumenta con la tensión aplicada. Esto es ideal si utilizamos la técnica de **PWM** (*pulse-width modulation*).
4. Puede funcionar como dinamo; si movemos su eje o rotor con una fuerza mecánica, en los terminales de conexión del motor se generará energía eléctrica.

2.5.8 Especificaciones Técnicas

- Voltaje de Operación: 3V – 6V
- Velocidad Angular nominal: 200 RPM
- Reducción: 48:1
- Consumo de corriente sin carga: 100mA (a 5V)
- Consumo de corriente nominal: 140mA (a 5V)
- Consumo de corriente eje detenido: 500mA (a 5V)
- Diámetro eje para rueda: 55 mm
- Peso: 30g

Voltaje de Operación	Parámetros	DC 3V	DC 5V	DC 6V
Parámetros Caja Reductora	Reducción	48:1		
	Velocidad sin carga	125 RPM	200 RPM	230 RPM
	Velocidad con carga	95 RPM	152 RPM	175 RPM
	Torque de salida	0.8kg.cm	1.0kg.cm	1.1kg.cm
	Velocidad del robot sin carga (metros/minuto)	25.9	41.4	47.7
	Corriente	110-130mA	120-140mA	130-150mA
	Diámetro máximo de llanta	6.5cm		
	Dimensiones	70mm x 22mm x 18mm		

Figura 2.16 Especificaciones Técnicas del motor DC

Fuente: <https://naylorlampmechatronics.com/motores-dc/606-motor-dc-con-caja-reductora-6v-200rpm.html>

2.6.2 Diagrama interno

Ahora detallemos la asignación de pines (*pinout*) del L293D.



Figura 2.19 Pines del L293D

Fuente: https://programarfacil.com/electronica/motor-dc/#Que_es_un_motor_electrico

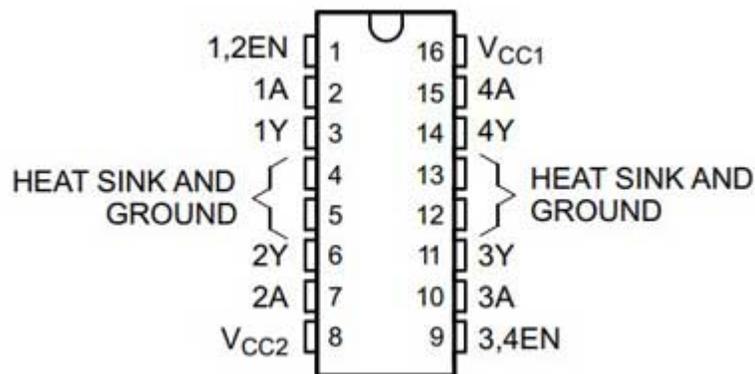


Figura 2.20 Descripción de pines del L293D

Fuente: https://programarfacil.com/electronica/motor-dc/#Que_es_un_motor_electrico

- Lo primero que observamos, es que hay dos Vcc o dos fuentes de alimentación Vcc1 y Vcc2.
- Vcc 1, se refiere a la polarización para la parte lógica TTL (*transistor-transistor logic*) del integrado, esta trabaja a 5V.

- Vcc 2, es para la alimentación de los motores o carga, con un rango de 4.5 a 36 V.
- La corriente máxima soportada es de 600 mA.
- La potencia máxima es de 4 W.
- Los pines 4, 5, 12 y 13 van a GND y allí podemos colocar el disipador de calor.
- Incorpora diodos de protección.
- Tiene protección térmica interna (*Thermal Shutdown*).
- Con el pin 1 podemos habilitar (*enable*) con un alto (HIGH) el motor DC y controlar la velocidad del motor DC conectado a los drivers 1 y 2, mediante señal PWM.
- El pin 9 realiza la misma función, pero con los drivers 3 y 4.
- El driver L293D posee 4 canales independientes, por tanto, podemos utilizarlo para 4 motores DC con un solo sentido de giro, para 2 motores DC con inversión de giro, o para motores paso a paso sin escobillas bipolares.

2.7 Transistores

2.7.1 El Transistor BC558

2.7.1.1 Descripción

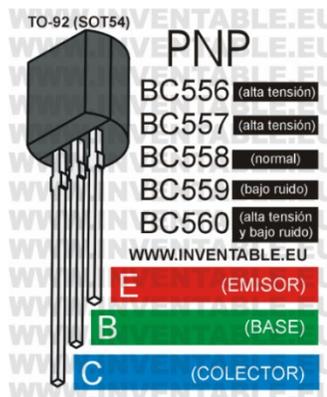


Figura 2.21 Vista pictórica del transistor BC558 y de los miembros de su familia con indicaciones generales.

Fuente: <https://www.inventable.eu/2021/02/13/transistor-bc558/>

El BC558 es un transistor bipolar PNP de alta calidad para aplicaciones generales, desarrollado por la Philips y la Mullard. Este transistor es parte de una familia de transistores con características casi iguales que son los BC556, BC557, BC559 y BC560 y son complementarios de los BC548 y la respectiva familia.

La diferencia entre los distintos modelos es la tensión máxima de trabajo como explicaré en el siguiente párrafo. Además, el BC559 y el BC560 tienen un muy bajo nivel de ruido y son usados con señales débiles de audio en sistemas de alta fidelidad. Para simplificar la explicación usaré el nombre BC55x cuando me refiero a todos los miembros de la familia.

2.7.1.2 Características de la familia BC558

Todos los miembros de la familia BC558 soportan corrientes de salida (corriente entre colector y emisor) hasta 100 mA mientras que las tensiones máximas dependen del modelo, como podemos ver en la figura siguiente.

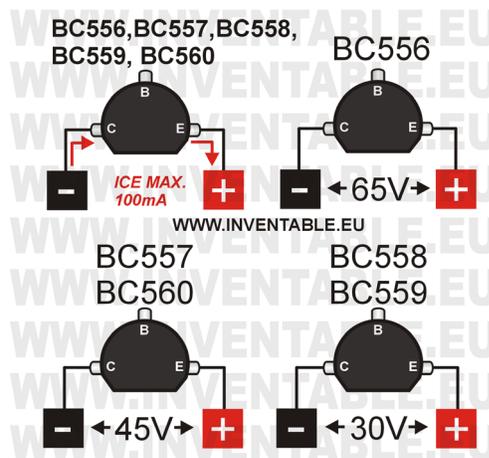


Figura 2.22 Tensiones y corriente máxima de los transistores de la familia BC558.

Fuente: <https://www.inventable.eu/2021/02/13/transistor-bc558/>

Como pueden observar, el rango de tensiones de trabajo según el modelo va desde los 30V hasta los 65V. Si nuestro circuito está alimentado por una tensión de 5V, 12V o 24V, podemos usar indistintamente cualquier modelo de la familia sin inconvenientes. La corriente máxima de 100 mA se refiere a corriente continua porque en el caso de picos de breve duración se puede llegar hasta 200 mA. Es importante aclarar que algunos fabricantes como la Fairchild construyeron modelos de BC558 que soportan corrientes hasta de 500mA no obstante no corresponda a las características estándar de este componente (se puede ver en la hoja del fabricante de la Fairchild). Esto ha creado una cierta confusión en el ambiente técnico sobre las reales capacidades del BC558. Para no correr riesgos, aconsejo de respetar el límite de los 100 mA, especificado en el modelo estándar.

BC556,BC557,BC558, BC559, BC560	
datos técnicos	
V_{CE} máx. <small>tension colector-emisor máxima</small>	65 V (BC556) 45 V (BC557, BC560) 30 V (BC558, BC559)
P_{TOTAL} máx. <small>potencia máxima</small>	500 mW
I_{CE} <small>corriente colector-emisor máxima</small>	100 mA
h_{FE} <small>ganancia en corriente (letra al final del código)</small>	BC55x: 110 - 800 BC55xA: 110 - 220 BC55xB: 220 - 450 BC55xC: 450 - 800
f_T <small>frecuencia de corte</small>	150MHz

Figura 2.23 Datos técnicos principales de los transistores BC556, BC557, BC558, BC559 y BC560.

Fuente: <https://www.inventable.eu/2021/02/13/transistor-bc558/>

Los transistores de la serie BC55x tienen una excelente ganancia, entre 110 y 800. La ganancia es la amplificación de corriente del transistor). Existen dos tipos de ganancia: para señales alternadas y para corriente continua pero como son parecidas, en este breve artículo hablo solo de la ganancia en continua, que se indica con la sigla hFE.

Si encontramos una letra al final del código del transistor, esta sirve para indicar en modo más preciso el rango de ganancia (o amplificación) que el componente puede tener, considerando la tolerancia del mismo. Si no se encuentra dicha letra, la ganancia puede abarcar todo el rango posible (entre 110 y 800).

En la tabla siguiente podemos ver las ganancias de la serie BC55x en base a la letra final en el código.

GANANCIA (h_{FE})

BC556	110 - 800
BC557	
BC558	
BC559	
BC560	
BC556A	110 - 220
BC557A	
BC558A	
BC556B	200 - 450
BC557B	
BC558B	
BC559B	
BC560B	
BC557C	450 - 800
BC558C	
BC559C	
BC560C	

WWW.INVENTABLE.EU

Figura 2.24 Tabla con las ganancias de la serie BC55x en base a la clase (indicada con una letra al final del código).

Fuente: <https://www.inventable.eu/2021/02/13/transistor-bc558/>

2.7.1.3 Respuesta en frecuencia

Toda la serie BC55x posee una buena respuesta en frecuencia, aproximadamente de 150 MHz. La respuesta en frecuencia se indica con la sigla f_T y es la frecuencia máxima en la cual el transistor amplifica por lo menos 1. Esto se debe al hecho que más alta es la frecuencia, más baja será la amplificación del transistor debido a la parte capacitiva del mismo (alrededor de 5 pF en este caso). Naturalmente, usar un transistor con frecuencias donde gana muy poco no es conveniente. Si necesitamos trabajar con frecuencias muy elevadas (muchos MHz) es conveniente usar transistores para RF (radio frecuencia), por ejemplo, los de la serie BF en lugar de esta serie BC.

2.7.2 El Transistor Bc548

2.7.2.1 Descripción

El BC548 es un transistor bipolar NPN de alta calidad para aplicaciones generales, desarrollado por la Philips y la Mullard entre 1963 y 1966. En origen se llamaba BC108 y tenía un encapsulado metálico (TO-18). Después pasó a un extraño encapsulado plástico con el código BC148 hasta llegar al actual BC548 con encapsulado TO-92 (conocido también como SOT-54). Este transistor es parte de una familia de transistores con características casi iguales que son los BC546, BC547, BC549 y BC550 y naturalmente el BC548.

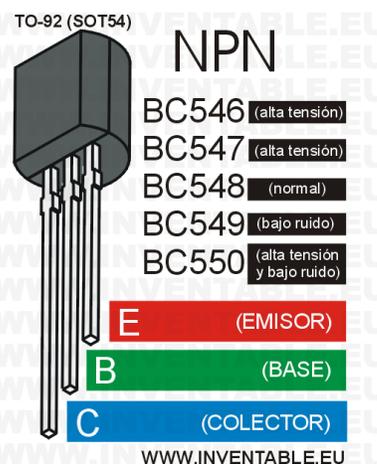


Figura 2.25 Vista pictórica del transistor BC548 y de los miembros de su familia con indicaciones generales.

Fuente: <https://www.inventable.eu/2017/07/09/transistor-bc548/>

La diferencia entre estos transistores es la tensión máxima de trabajo como explicaré en el siguiente párrafo. Además, el BC549 y el BC550 tienen un muy bajo nivel de ruido y son usados con señales débiles de audio en sistemas de alta fidelidad. Para simplificar la explicación usaré el nombre BC54x cuando me refiero a todos los miembros de la familia.

2.7.2.2 Características de la familia BC548

Todos los miembros de la familia BC548 soportan corrientes de salida (corriente entre colector y emisor) hasta 100 mA mientras que las tensiones máximas dependen del modelo, como podemos ver en la figura siguiente.

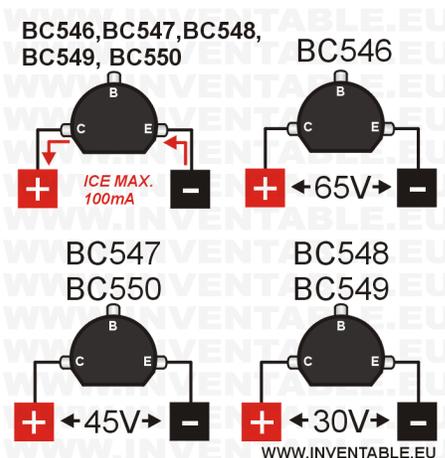


Figura 2.26 Tensiones y corriente máxima de los transistores BC546, BC547, BC548, BC549 y BC550.

Fuente: <https://www.inventable.eu/2017/07/09/transistor-bc548/>

Como pueden observar, el rango de tensiones de trabajo según el modelo va desde los 30V hasta los 65V. Si nuestro circuito está alimentado por una tensión de 5V, 12V o 24V, podemos usar indistintamente cualquier modelo de la familia sin inconvenientes. La corriente máxima de 100 mA se refiere a corriente continua porque en el caso de picos de breve duración se puede llegar hasta 200 mA. Es importante aclarar que algunos fabricantes como la Fairchild construyeron modelos de BC548 que soportan

corrientes hasta de 500mA no obstante no corresponda a las características estándar de este componente (se puede ver en la hoja del fabricante de la Fairchild). Esto ha creado una cierta confusión en el ambiente técnico sobre las reales capacidades del BC548. Para no correr riesgos, aconsejo de respetar el límite de los 100 mA, especificado en el modelo estándar.

BC546, BC547, BC548, BC549, BC550	
datos técnicos	
V_{CE} máx. <small>tension colector-emisor máxima</small>	65 V (BC546) 45 V (BC547, BC550) 30 V (BC548, BC549)
P_{TOTAL} máx. <small>potencia máxima</small>	500 mW
I_{CE} <small>corriente colector-emisor máxima</small>	100 mA
h_{FE} <small>ganancia en corriente (letra al final del código)</small>	BC54x: 110 - 800 BC54xA: 110 - 220 BC54xB: 220 - 450 BC54xC: 450 - 800
f_T <small>frecuencia de corte</small>	150MHz - 300 MHz WWW.INVENTABLE.EU

Figura 2.27 Datos técnicos principales de los transistores BC546, BC547, BC548, BC549 y BC550.

Fuente: <https://www.inventable.eu/2017/07/09/transistor-bc548/>

Los transistores de la serie BC54x tienen una excelente ganancia, entre 110 y 800. La ganancia es la amplificación de corriente del transistor). Existen dos tipos de ganancia: para señales alternadas y para corriente continua pero como son parecidas, en este breve artículo hablo solo de la ganancia en continua, que se indica con la sigla hFE.

Si encontramos una letra al final del código del transistor, esta sirve para indicar en modo más preciso el rango de ganancia (o amplificación) que el componente puede

tener, considerando la tolerancia del mismo. Si no se encuentra dicha letra, la ganancia puede abarcar todo el rango posible (entre 110 y 800).

En la tabla siguiente podemos ver las ganancias de la serie BC54x en base a la letra final en el código.

GANANCIA (h_{FE})	
BC546 BC547 BC548 BC549 BC550	110 - 800
BC546A BC547A BC548A	110 - 220
BC546B BC547B BC548B BC549B BC550B	200 - 450
BC547C BC548C BC549C BC550C	450 - 800

Figura 2.28 Tabla con las ganancias de la serie BC54x en base a la clase (indicada con una letra al final del código).

Fuente: <https://www.inventable.eu/2017/07/09/transistor-bc548/>

2.7.2.3 Respuesta en frecuencia

Toda la serie BC54x posee una buena respuesta en frecuencia, entre 150 MHz y 300 MHz. La respuesta en frecuencia se indica con la sigla f_T y es la frecuencia máxima en la cual el transistor amplifica por lo menos 1. Esto se debe al hecho que más alta es la frecuencia, más baja será la amplificación del transistor debido a la parte capacitiva del mismo (alrededor de 5 pF en este caso). Naturalmente, usar un transistor con frecuencias donde gana muy poco no es conveniente. Si necesitamos trabajar con frecuencias muy elevadas (muchos MHz) es conveniente usar transistores para RF (radio frecuencia), por ejemplo, los de la serie BF en lugar de esta serie BC.

2.7.3 Transistor TIP 41

2.7.3.1 Descripción

Semiconductor bipolar de alta potencia de juntura NPN. Este transistor es capaz de disipar hasta 65W, aun así, puede controlar dispositivos que consuman hasta 6A o que requieran tensiones de hasta 100VDC, siempre y cuando no se exija el dispositivo hasta o sobre su potencia de disipación máxima.

Su juntura es NPN, con un factor de amplificación (hFE) que varía de acuerdo a la carga a utilizar, variando entre 15 y 75.

Dispositivo diseñado para el Procesado de Señales, Administración de Potencia, Dispositivos Portátiles, Electrónica de Consumo e Industrial, puede operar como Amplificador de Audio y Driver

2.7.3.2 Principales características

- Tipo de Encapsulado: TO-220
- Transistor complementario TIP42C
- Transistor equivalente a TIP41, TIP41A, TIP41B, NTE331
- Transistor de Unión Bipolar BJT Tipo NPN
- Voltaje Colector - Emisor (VCE): 100V
- Voltaje Colector - Base (VCB): 100V
- Voltaje Emisor - Base (VEB): 5V
- Corriente Colector (IC): 6A
- Corriente Base (IB): 2A
- Potencia Máxima Disipada (PD): 65W
- Ganancia Máxima (hFE): 75
- Frecuencia de Trabajo (FT): 3 MHz

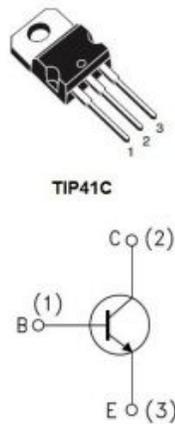


Figura 2.29 Configuración del transistor TIP 41

Fuente: <https://www.zamux.co/transistor-tip41>

2.8 Cristales de Cuarzo

2.8.1 Descripción

Cristal de cuarzo. Son componentes electrónicos utilizados en circuitos osciladores. Estos oscilan a una frecuencia estable. Por lo que tienen una gran utilidad principalmente en la Electrónica Digital y radiofrecuencia.



Figura 2.30 Cristal Oscilador

Fuente: https://www.ecured.cu/Cristal_de_cuarzo

2.8.2 Características

En los cristales de cuarzo se produce un fenómeno llamado piezoelectricidad. Este fenómeno consiste en que la aplicación de una tensión eléctrica produce una deformación del cristal, mientras que la deformación del cristal genera una tensión eléctrica. Esta característica se aprovecha en electrónica para producir tensiones alternas con una gran estabilidad de frecuencia.

La frecuencia de oscilación de un cristal viene determinada por el grueso de la lámina de cuarzo y la dirección en que se dio el corte del cristal original para obtener la lámina.

2.8.3 Resonancia

Cuando se aplica una tensión alterna entre los electrodos, se produce una deformación de la lámina de cuarzo. Esta deformación se traslada de un lado de la lámina hacia el otro lado, donde se refleja hacia el lado contrario. Si esta deformación reflejada coincide en fase con la deformación inicial, las dos se suman incrementándose notablemente. Según se produce la resonancia y la frecuencia de la tensión aplicada se dice que es la fundamental del cristal. Si la frecuencia de la tensión aplicada es, por ejemplo, tres veces mayor, también se produce la coincidencia de fase entre la deformación inicial y la deformación reflejada dándose también la condición de resonancia. Esta condición de resonancia también se da en otras frecuencias múltiplos impares de la frecuencia fundamental. En este caso se dice que el cristal oscila en el tercer, quinto, etc.

Durante las reflexiones de las ondas en los lados de la lámina de cristal, hay un pequeño error de fase, por lo que la frecuencia de un sobre tono no coincidirá exactamente con un múltiplo de la frecuencia fundamental. Es por tanto que no hay que confundir sobre tono con armónico. Sobre tono no es otra cosa que una resonancia mecánica del cristal, mientras que armónico es una señal adicional cuya frecuencia es un múltiplo exacto (también los pares) de la frecuencia fundamental.

2.8.4 Construcción

La construcción de un cristal de cuarzo resulta curiosa: primero se corta una lámina (normalmente de forma circular) de sección aproximada a la frecuencia de resonancia fundamental que se desea obtener. Como es técnicamente imposible conseguir la frecuencia exacta durante el corte, es necesario esmerilarla con máquinas muy precisas. Seguidamente, se pulveriza y hornea en ambas caras unas finas películas metálicas de solución de plata, o también mediante evaporación de oro, plata o aluminio, con objeto de disponer de dos superficies de contacto (los electrodos). Mediante una soldadura de bajo punto de fusión, se conectan a las superficies metálicas dos alambres conductores. El punto donde se sueldan los alambres debe ser elegido cuidadosamente, y tiene que ser forzosamente en un punto llamado “nodal”, es decir, donde el cristal no produzca vibración, pues en caso contrario el alambre amortiguará la resonancia y dejará de vibrar, de la misma forma que una campanilla se ahogaría si la tocamos con la mano cuando está sonando. Finalmente, el cristal es encapsulado en una caja herméticamente cerrada, de vidrio, metálica u otro material adecuado.

Antes de su comercialización, al cristal se le hace vibrar durante unas horas para que envejezca y se estabilice.

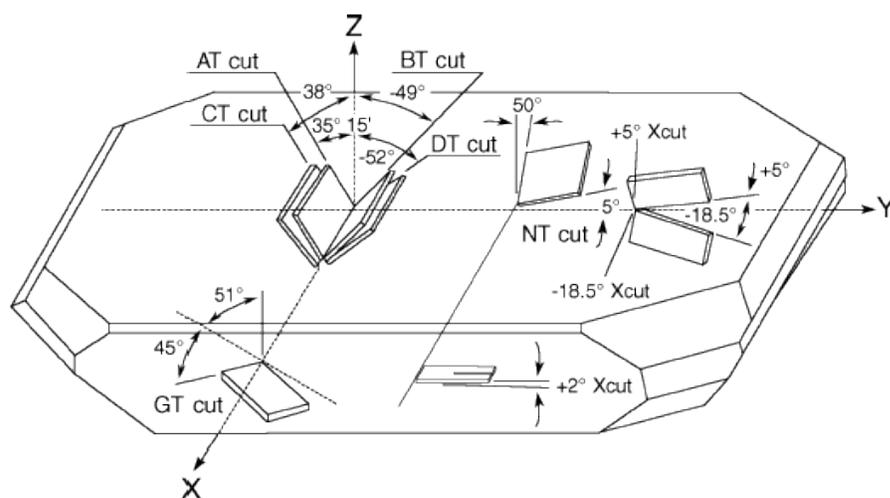


Figura 2.31 Osciladores de Cristal

Fuente: https://www.ecured.cu/Cristal_de_cuarzo

2.8.5 Circuito eléctrico equivalente

Un cristal de cuarzo se puede representar mediante un circuito eléctrico, llamado circuito eléctrico equivalente.

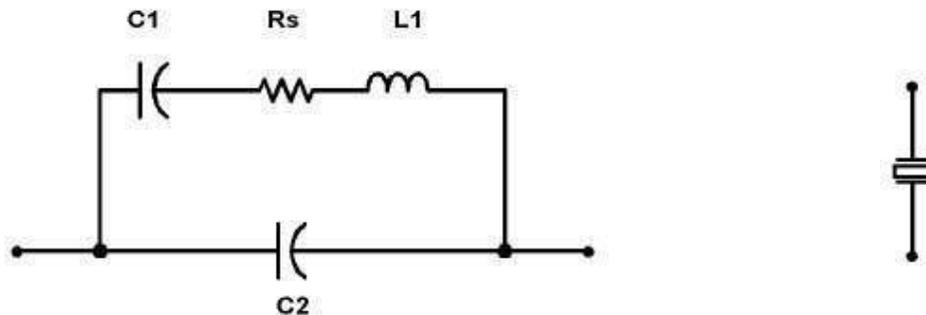


Figura 2.32 Circuito Equivalente del cristal de cuarzo

Fuente: https://www.ecured.cu/Cristal_de_cuarzo

Es un esquema del cristal de cuarzo trabajando a una determinada frecuencia de resonancia. El capacitor C' o capacidad en paralelo, representa en total la capacidad entre los electrodos del cristal más la capacidad de la carcasa y sus terminales. R , C y L conforman la rama principal del cristal y se conocen como componentes o parámetros nomenclatura donde: L representa la masa vibrante del cristal, C representa la elasticidad del cuarzo y R representa las pérdidas que ocurren dentro del cristal.

2.9 Parlante o Altavoz

2.9.1 Descripción

Los parlantes o altavoces tienen como objetivo la reproducción de sonidos a través de un transductor electroacústico, donde las ondas eléctricas se convierten en energía mecánica, que después llegan a ser ondas de frecuencia acústica.

Una vez hecho esto, el sonido se va transmitiendo de forma paulatina a través del aire para que finalmente lleguen a los oídos del receptor. Será en esta parte del cuerpo donde

se capten estas ondas que después serán transformadas en impulsos nerviosos que llegan al cerebro, que este finalmente interpreta como voces, sonido, música u onomatopeyas.



Figura 2.33 Parlante o altavoz de 8 Ohmios

Fuente: <http://www.velasco.com.ec/velasco/producto.php?id=484>

2.9.2 Principales características de los altavoces

2.9.2.1 Respuesta de frecuencia

Este valor tiene relación con la calidad del sonido que llega al oído del ser humano. Se mide en distintos niveles de audiofrecuencia (desde 20 hasta 20.000Hz). En este sentido, a mayor respuesta de frecuencia, mayor será la calidad del sonido final. Debido a que este valor no es completamente plano, la respuesta no será la misma en los sonidos graves o agudos. Por esta razón, se antoja muy importante que exista la menor diferencia de sonido (en decibelios) para evitar así distorsiones.

2.9.2.2 Potencia

Esta característica está relacionada con la cantidad de potencia eléctrica que son capaces de soportar los altavoces. No tiene nada que ver con la potencia acústica y se mide siempre en vatios. Si el altavoz recibe demasiada “potencia”, es posible que sufra daños o el sonido salga distorsionado. Cabe destacar que existen dos tipos de potencia:

Potencia nominal. Es la cantidad de potencia “máxima” que puede soportar el altavoz antes de que este se deteriore. De lo contrario, este puede dañarse de forma definitiva, ya que el periférico no es capaz de disipar el calor producido por la corriente eléctrica.

Potencia admisible. Esta consiste en los picos de señales que pueden llegar a soportar los altavoces en cortos espacios de tiempo, justo antes de que se deteriore el diafragma que hay en el interior de la bobina.

2.9.2.3 Impedancia

La impedancia es un valor que se mide en ohmios y tiene que ver con la “oposición” que tiene el sonido a la hora de pasar por una corriente alterna. En el caso de los altavoces, la impedancia varía según la frecuencia y por tanto también el modelo de altavoz. De una forma u otra, si se quiere conseguir la mayor transferencia de energía entre la fuente de sonido (el amplificador) y el altavoz, las impedancias entre ambos componentes deben ser lo más similar posible.

2.9.2.4 Sensibilidad

La sensibilidad mide nada más y nada menos que la relación entre el nivel eléctrico de la entrada del altavoz y la presión sonora obtenida por el sonido producido. Se mide siempre en DB/W a un metro de distancia y aplicando una potencia de 1W directamente al altavoz.

2.9.2.5 Rendimiento

Aquí se mide el grado de sensibilidad del altavoz. Se trata de un valor que determina la relación entre la potencia acústica radiada y la potencia eléctrica que viene de entrada. Si el rendimiento de los altavoces es bajo, se comprende entre el 1-5%. Es decir, si entran por ejemplo 100W, salen siempre 1-5W.

2.9.2.6 Distorsión

Se puede considerar al altavoz como uno de los sistemas de audio que provocan más distorsión debido a que existen todo tipo de elementos que la provocan, como, por ejemplo, las vibraciones parciales, o el flujo de entrehierro. Gran parte de la distorsión se suele dar en los sonidos graves.

2.9.2.7 Direccionalidad

Este valor determina la dirección del sonido justo desde el momento en el que sale del altavoz. Es decir, la manera que este se distorsiona al llegar al ambiente. A partir de ahí, la dirección del sonido se puede dividir en tres: omnidireccional (el sonido radia a 360°), bidireccional (el sonido viaja en forma de ocho, de forma simétrica) y unidireccional (el sonido va en una dirección muy marcada y por tanto es muy silencioso)

2.9.3 Especificaciones y características técnicas

- Tipo: Altavoz 8 ohm 5W
- Impedancia nominal: $8\Omega \pm 15\%$
- Potencia de entrada nominal: 0.8W
- Potencia máxima de entrada: 5W
- Frecuencia de resonancia: $800 \pm 20\%$ Hz
- Nivel de presión sonora de salida: $93 \pm 3\text{dB}/0.1\text{m}$ 0.1W a 0.8, 1.0, 1.2, 1.5 KHz promedio de 15KHz
- Temperatura de operación: -20°C a $+70^{\circ}\text{C}$
- Diámetro: 20mm
- Longitud del cable: 35mm
- Espesor: 4mm
- Peso: 2g

2.10 Sensor Óptico Laser – LDR (SOL)

2.10.1 Descripción

Permite detección de objetos que cortan el paso de luz que emana desde el diodo laser hacia la resistencia LDR. El pin de Salida puede ser cualquier bit de cualquier puerto del microcontrolador. El pin de Entrada debe ser el bit RB.0 para provocar una interrupción externa.

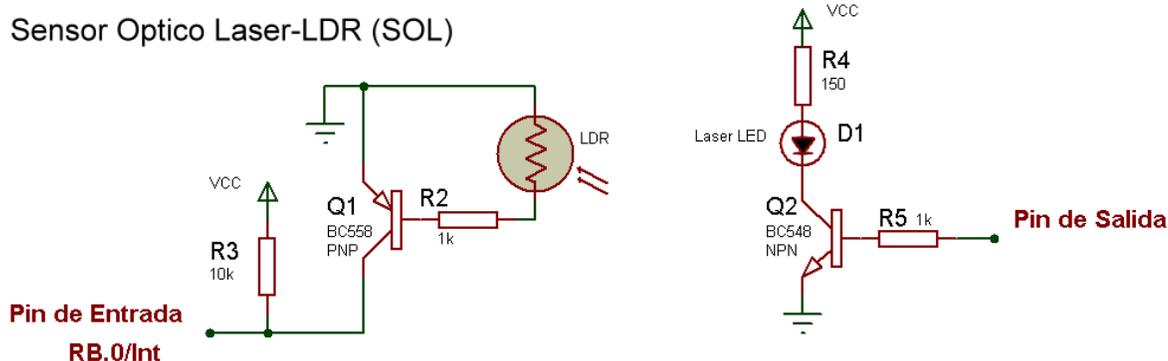


Figura 2.34 Diagrama de sensor Óptico Laser - LDR

Fuente: Guachalla, R. (2015). *Laboratorio de Microprocesadores I*. La Paz.

2.10.2 Diodo Laser

Un láser (del inglés Light Amplification by Stimulated Emission of Radiation - amplificación de luz por emisión estimulada de radiación-) es un dispositivo que utiliza la emisión inducida o estimulada para generar o amplificar la radiación electromagnética en el intervalo de la longitud de onda de la radiación óptica produciendo un haz de luz coherente que viaja a la velocidad de la luz.

La radiación láser, es decir aquella que procede de un dispositivo láser, presenta ciertas propiedades:

es monocromática: radiación compuesta de ondas de la misma longitud de onda.

es coherente: las ondas mantienen la fase relativa al desplazarse. Al resto de radiaciones ópticas se las denomina incoherentes.

es direccional: al alejarse de la fuente, la dispersión de la radiación es muy pequeña.

Estas características hacen que la radiación láser pueda lograr densidades de energía muy elevadas, que casi no disminuyen al aumentar la distancia a la fuente. Por ello el láser se ha convertido en una herramienta muy utilizada en la tecnología, a nivel industrial, médico, de investigación, comunicaciones, etc. y cada vez se utiliza más. Pero

estas características hacen también que tenga unos efectos potenciales de daño a la salud más importantes y su evaluación necesita condiciones particularizadas.



Figura 2.35 Diodo Laser

Fuente: <https://laredelectronica.com/producto/diodo-laser-6mm-3v-mw-650/>

2.10.3 LDR

2.10.3.1 ¿Qué es la fotorresistencia o LDR?

El LDR por sus siglas en inglés (Light Dependent Resistor) o fotorresistor es una resistencia eléctrica la cual varía su valor en función de la cantidad de luz que incide sobre su superficie. Cuanto mayor sea la intensidad de luz que incide en la superficie del LDR o fotorresistor menor será su resistencia y en cuanto menor sea la luz que incida sobre éste mayor será su resistencia.

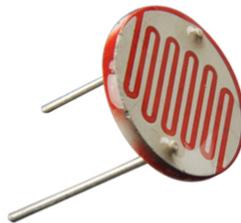


Figura 2.36 LDR

Fuente: <https://www.prometec.net/fotorresistencias-s4a/>

2.10.3.2 ¿Cómo funciona una LDR o fotorresistencia?

Cuando el LDR (fotorresistor) no está expuesto a radiaciones luminosas, los electrones están firmemente unidos en los átomos que lo conforman, pero cuando sobre él inciden radiaciones luminosas, esta energía libera electrones con lo cual el material se hace más conductor, y de esta manera disminuye su resistencia. Las resistencias LDR solamente reducen su resistencia con una radiación luminosa situada dentro de una determinada banda de longitudes de onda. El fotorresistor construido con sulfuro de cadmio es sensible a todas las radiaciones luminosas visibles y las construidas con sulfuro de plomo solamente son sensibles a las radiaciones infrarrojas.

2.10.3.3 Tipos de fotorresistencias

Los tipos de fotorresistencias los podemos clasificar según los materiales fotosensibles o clasificar según su linealidad. Los materiales de cristal semiconductor fotosensible más utilizados para la fabricación de las resistencias LDR son el sulfuro de talio, el sulfuro de cadmio, el sulfuro de plomo, y el seleniuro de cadmio.

- **Sulfato de cadmio:** Las fotorresistencias elaboradas con este químico son extremadamente sensibles a todo tipo de radiaciones luminosas que son visibles en el espectro del ser humano.
- **Sulfato de plomo:** Las fotorresistencias elaboradas con este químico son especialmente sensibles a las radiaciones infrarrojas.

La clasificación más común es mediante lineales y no lineales:

- **Fotorresistencia lineal:** Son más conocidos como fotodiodos, pero en algunas aplicaciones es posible utilizar como fotorresistores debido al comportamiento lineal que presentan y su funcionamiento. (Se polariza de manera inverso)
- **Fotorresistencia no lineal:** Son las más comunes y son aquellas cuyo comportamiento no depende de la polaridad con la que se conecte.

2.10.3.4 Valor óhmico de la LDR

Cuando medimos entre sus extremos nos encontraremos que pueden llegar a medir en la oscuridad valores cercanos al Mega Ohm ($1\text{ M}\Omega$) y expuestas a la luz mediremos valores alrededor de los $100\ \Omega$ (en algunos casos puede descender a valores de $50\ \Omega$).

2.10.3.5 Formas de conectar un fotorresistor

Existen dos formas básicas para la conexión de nuestra LDR, pueden ser utilizadas dependiendo del fin deseado. Si disponemos de un controlador es posible modificar los resultados mediante programación.

1. **Mayor luz, mayor voltaje:** Al conectar la fotorresistencia al nodo positivo de nuestra fuente de voltaje tendremos que, al incidir una mayor cantidad de luz provocará una menor caída de voltaje o diferencial de potencial entre la fuente y el pin de referencia (V_{out}), por lo tanto se tendrá una lectura mayor.
2. **Mayor luz, menor voltaje:** En pocas palabras la fotorresistencia se conecta al nodo de GND y provocará un comportamiento opuesto al punto 1.

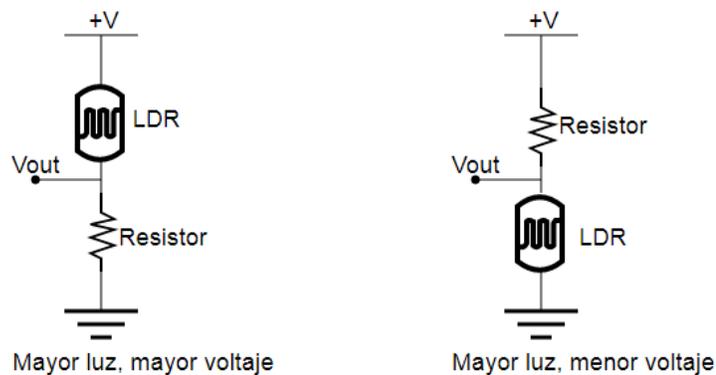


Figura 2.37 Configuración del LDR

Fuente: <https://www.mecatronicalatam.com/es/tutoriales/sensores/sensor-de-luz/ldr/>

2.10.3.6 Características técnicas del fotorresistor

- Los valores típicos varían entre 1 M Ω o más en la oscuridad y 100 Ω con luz brillante.
- Disipación máxima, (50 mW-1W).
- Voltaje máximo (600V).
- Respuesta espectral.
- El tiempo de respuesta típico de un LDR está en el orden de una décima de segundo.

2.11 Pulsadores

2.11.1 Introducción.

Que son los pulsadores. Pulsadores o interruptores, hay toneladas de ellos en su casa. Un interruptor es un dispositivo simple con dos posiciones, EN y AP (Encendido y Apagado). Una clase de interruptor que usted usa cada día es el interruptor de la luz. Cuando conecta, dentro del interruptor, dos cables son unidos, lo que permite fluir a la corriente que enciende la luz o la tostadora se caliente. Cuando lo desconecta, los dos cables son desunidos y corta el flujo de la corriente.

En definitiva, se trata de un mecanismo simple (los hay muy sofisticados), constituido por un par de contactos eléctricos que se unen o separan por medios mecánicos. En electricidad, los falsos contactos que se producen al ser utilizados normalmente, en algunos casos produce una chispa debido a la corriente que atraviesa los contactos, provocando que quemen en parte y ennegreciendo los contactos eléctricos, lo que a la larga acaba deteriorando dichos contactos. La chispa se produce siempre al separar los contactos (desconectar), en ocasiones parece que también salta al conectarlos, eso es debido a los rebotes mecánicos que se producen al cambiar de estado.

Esto que en electricidad se considera normal, en electrónica es un verdadero nido de problemas, debido a dichos falsos contactos. Por su propia naturaleza, al cambiar de

posición un interruptor, los contactos chocan entre sí y esto significa una serie de falsos contactos que se reproducen de un modo sin control, por lo que se generan los temidos rebotes (debounce en inglés), estos rebotes, se producen incluso cuando unimos dos cables desnudos, simulando un interruptor o pulsador.

2.11.2 Pulsadores.

Estos pequeños pulsadores son un 1/4 por cada lado, son bastante económicos y se pueden ‘pinchar’ directamente en una placa de prototipado. Estos dispositivos mecánicos tienen 4 patillas, que pueden hacernos pensar que hay 4 cables que son EN y AP, pero, de hecho, dos de cada lado en realidad dentro están unidos. Por tanto, realmente, este pulsador es solamente un pulsador de 2 cables.



Figura 2.38 Símbolo y forma física del pulsador

Fuente: <https://www.diarioelectronico hoy.com/blog/pulsadores-sin-rebotes>

2.11.3 Resistencias de polarización.

Cuando nos desenvolvemos en el entorno de los microcontroladores, nos encontramos con un término poco común, me refiero a la polarización de una E/S, debemos saber que hay dos tipos de polarización, polarización alta la resistencia (término inglés **Pullup**) va conectada a + (5V) o polarización baja la resistencia (término inglés **Pulldown**) va conectada a masa – (0V). Siguen dos esquemas de estos términos:

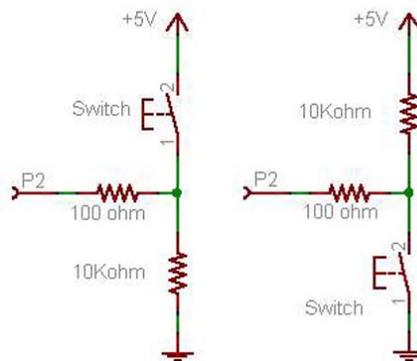


Figura 2.39 Configuración de uso de los pulsadores

Fuente: <https://www.diarioelectronico.com/blog/pulsadores-sin-rebotes>

2.12 LED

2.12.1 Qué es un led?

Un LED (acrónimo del concepto inglés light-emitting diode) es un diodo emisor de luz. En su interior hay un semiconductor que, al ser atravesado por una tensión continua, emite luz, lo que se conoce como electroluminiscencia. Existen distintos tipos de led en función de las tecnologías usadas para su fabricación y montaje sobre circuitos electrónicos.

La tensión de cualquier diodo LED es de 2 voltios y, en el caso que se quiera conectar a otros aparatos con una tensión distinta, se debe crear una conexión de resistencia en serie que permita su correcto funcionamiento.



Figura 2.40 Diodos Leds

Fuente: <https://www.zonamaker.com/electronica/intro-electronica/componentes/el-diodo-led>

2.12.2 Funcionamiento de un led

¿Cómo funciona el LED? Se basa en la activación de una unión PN, consistente en una estructura básica de componentes electrónicos denominados semiconductores (fundamentalmente, diodos y transistores inorgánicos). Todo diodo consta de dos patillas de conexión, una larga y una corta. Mientras la larga se conecta al polo positivo, la corta se conecta al polo negativo. Esto es lo que permite el paso de la corriente.

Los componentes que forman parte de los semiconductores, por otro lado, determinarán los colores que emitirán. Por poner algunos ejemplos, el naranja se produce con fosforo de galio y arsénico, fosforo de aluminio-galio-indio y fosfato de galio; el verde se logra a partir de fosfato de galio y fosforo de aluminio-galio-indio, mientras que el azul se crea a partir de seleniuro de zinc, nitruro de galio-indio y carburo.

2.12.3 ¿Cuáles son las partes del diodo led?

Para entender cómo funciona el LED debemos analizar también los distintos componentes que lo conforman. Entre ellos, encontramos los siguientes.

– **Ánodo y cátodo:** aunque ya hemos hablado de ellos más arriba, debemos recordar que se trata de las dos patillas de conexión. Mientras el ánodo actúa como polo positivo, en cátodo actúa como polo negativo.

– **Copa reflectora:** se encuentra encima del marco (formado a su vez por un yunque y una plaqueta) y, como su nombre sugiere es la encargada de reflejar la luz del color que emita el semiconductor.

– **Semiconductor:** los LED, dependiendo de su tipología, tendrán uno o tres cristales semiconductores, dependiendo de si pueden emitir un solo color o una combinación en RGB.

–**Hilo conductor**: a través del hilo conductor se transmite la corriente hacia el semiconductor que convertirá esta energía en luz.

–**Cápsula**: capa transparente que cubre y protege el resto de los componentes del diodo LED.

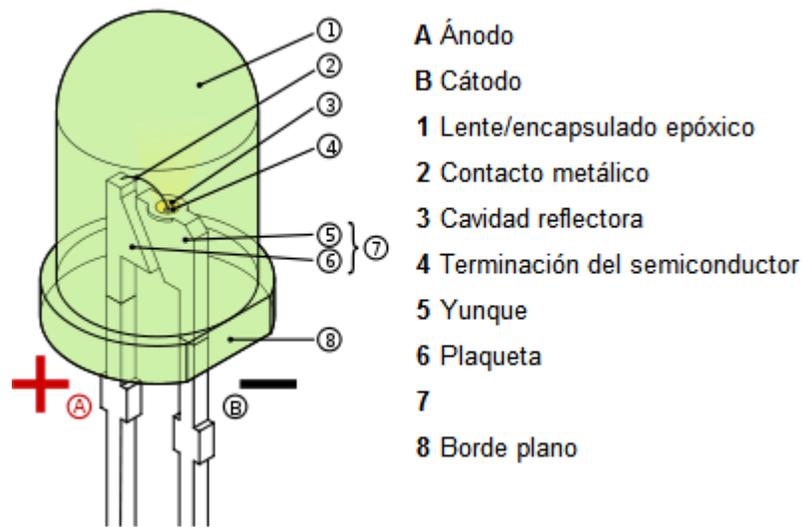


Figura 2.41 Partes del diodo Led

Fuente: <https://www.ledtecnologia.com/que-es-un-led/>

CAPITULO III

INGENIERIA DEL PROYECTO

3.1 Herramientas Software

3.1.1 El compilador MikroC PRO

La programación de microcontroladores se basa en un código de máquina que es conocido como código ensamblador, este código contiene una a una las instrucciones del programa, este código ensamblador o también conocido como código assembler es minucioso, y tedioso de editar. El assembler crea códigos de programa extensos y de difícil comprensión. La creación de compiladores de alto nivel facilitó la edición y creación de programas en todo modo de programación lógica, por supuesto los microcontroladores no fueron la excepción, comercialmente existen varios compiladores de diferentes fabricantes y diferentes lenguajes de alto nivel.

Es posible adquirir compiladores como el PICC, CCS, PIC Basic, entre otros. El estudio de este libro se centra en el compilador MikroC PRO, que es un compilador en lenguaje C para microcontroladores PICMicro de la familia 12F, 16F, y 18F.

MikroC PRO es un paquete de software con una amplia variedad de ayudas y herramientas que facilita la creación de proyectos y aplicativos para los microcontroladores PICMicro.

El estudio de este entorno de desarrollo es posible debido a que el estudiante puede descargar una versión demo o estudiantil, que tiene las mismas características de la versión completa, la única limitación es la dimensión del código de máquina que no puede exceder 2K bytes, sin embargo, es una capacidad suficiente al tratarse de un primer aprendizaje.

En la siguiente figura se puede apreciar la apariencia visual del entorno de desarrollo.

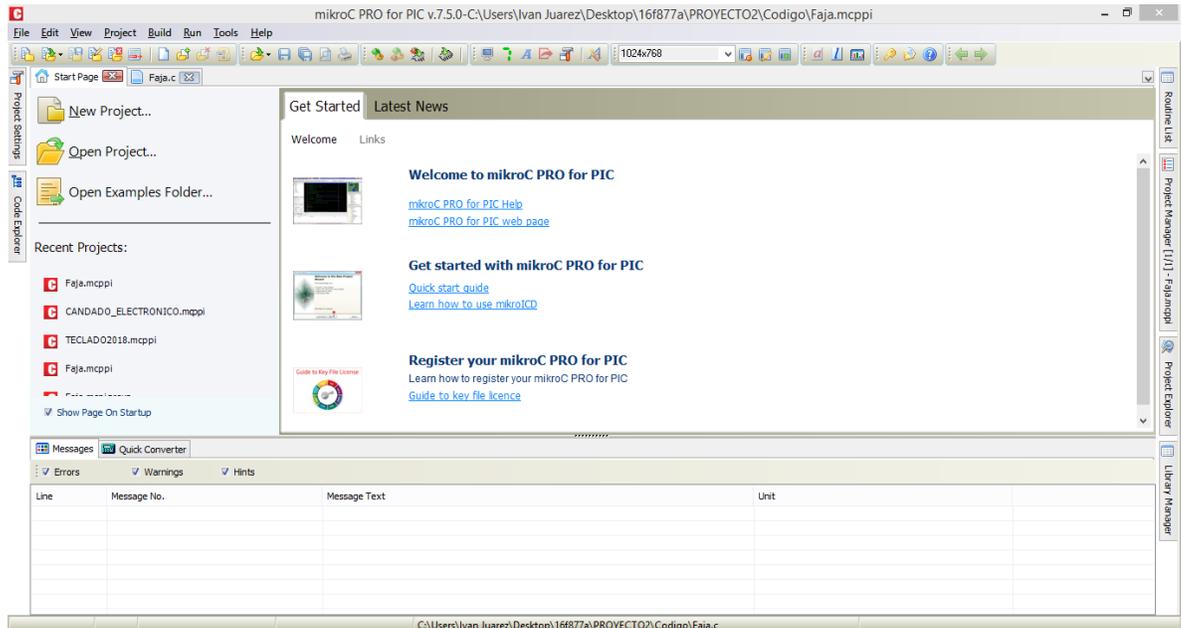


Figura 3.1 Entorno del mikroC PRO for Pic

Fuente: Elaboración Propia

El compilador de alto nivel en lenguaje C utiliza estructuras que facilitan la programación, optimiza las operaciones matemáticas y los procesos, por medio del uso de funciones predefinidas y las no predefinidas que el desarrollador puede crear, así como el uso de un conjunto de variables, de tipo carácter, entero, y punto decimal. El compilador crea automáticamente el código ensamblador y a su vez un código similar consignado en un archivo con extensión *.hex, este archivo es el resultado primordial del compilador dado que con este se programa eléctricamente el microcontrolador o con el mismo se puede realizar una simulación computacional.

3.1.2 El simulador ISIS de Proteus

El simulador ISIS de Proteus es un poderoso paquete de software, desarrollado por la compañía labcenter electronics, que se ha posicionado desde hace más de 10 años, como una de las herramientas más útiles para la simulación de los microcontroladores PICMicro. El ISIS permite la simulación de las familias de los PICMicro más populares

tales como la: 12F, 16F, 18F. Además de los PIC, el ISIS puede simular una gran variedad de dispositivos digitales y analógicos, entre los dispositivos digitales es posible simular displays de siete segmentos, de caracteres, y gráficos. ISIS puede simular sensores de temperatura, humedad, presión, y luminosidad, entre otros. El simulador permite, simular actuadores como: motores dc, servo motores, luces incandescentes entre otros. Es posible simular periféricos de entrada y salida como teclados, y puertos físicos del ordenador como: RS232, y USB. Este simulador cuenta con una amplia variedad de instrumentos de medición como voltímetros, amperímetros, osciloscopios, y analizadores de señal. En conclusión, éstas y otras características hacen del ISIS de Proteus, una herramienta ideal para el diseño y estudio de los PICMicro.

En la siguiente imagen se puede apreciar la apariencia visual del entorno de desarrollo del ISIS:

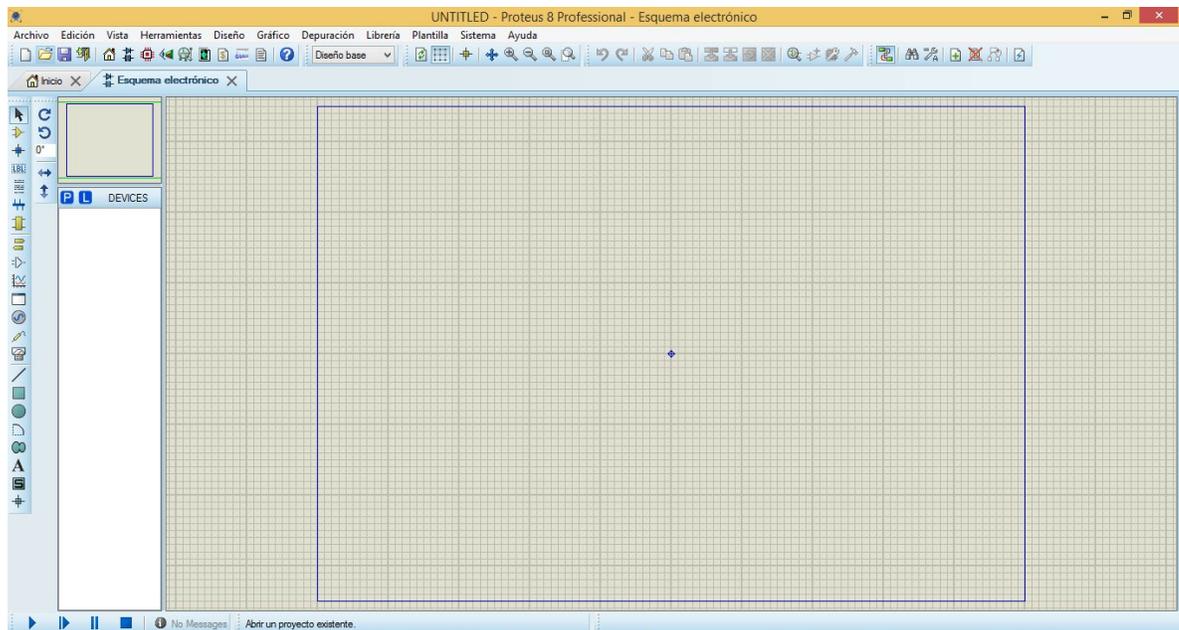


Figura 3.2 Entorno de Proteus ISIS

Fuente: Elaboración Propia

3.1.3 PICkit2

PICkit 2: Programador, Emulador, Debugger, es un programador (también conocido como grabador o quemador) diseñado por **Microchip** para la programación de una gran variedad de dispositivos. La lista de dispositivos soportados por el PICkit 2 es muy amplia e incluye microcontroladores PIC de las familias 12F/16F/18F, DsPIC, memorias EEPROM.

Compatible con la interfaz del entorno de desarrollo integrado (IDE) MPLAB, MPLABX y PICkit 2 GUI. El PICkit 2 se puede conectar a una PC mediante una interfaz USB.

El PICkit 2 permite además la programación **ICSP** (In Circuit Serial Programming) es decir la programación en circuito que nos da la ventaja de no tener que estar moviendo nuestro microcontrolador PIC del programador al circuito de prueba, además de que nos brinda la oportunidad de depuración en tiempo real.

Tiene circuitos para protección de sobre tensión y de corto circuito, permite ejecución en tiempo real y soporta tensiones desde 2.0V. Respetando la norma USB puede dar 100mA al circuito donde está conectado y tiene LEDs de información rápida para el usuario.

Además de estas funcionalidades, hay una que es muy interesante y en algunos entornos bastante útil del PICkit 2 que es la funcionalidad “Programmer-To-Go”. La funcionalidad “Programmer-To-Go” permite de una forma muy sencilla programar cualquier microcontrolador de las familias PIC16, PIC18, PIC24, dsPIC33F y PIC32 de Microchip sin necesitar de un ordenador, lo que es muy útil si estamos en un entorno donde llevar un ordenador no es practico o es del todo imposible. El PICkit2 permite guardar un código de hasta 512KB en su Flash, esto se hace desde el MPLAB y después llevar el PICkit 2 y de forma autónoma sólo alimentando el PICkit 2 desde el USB.

Cuenta además con dos herramientas adicionales integradas en el programador; un analizador de señales digitales y una interface Serial (TTL)-USB por medio del programador las cuales vienen en el paquete de PICkit 2 Programmer Tools.

El zócalo ZIF40 está destinado a la programación de microcontroladores PIC de 8, 18, 28 y 40 pines.

Nota: El Zócalo ZIF es compatible con la mayoría de familias PIC como PIC12F y PIC16F. Para los PICs de la familia PIC18F como el PIC18F2550 o el PIC18F4550 y los dsPIC como el dsPIC30F4013 o dsPIC30F3010, el zócalo ZIF NO es compatible. Estos PICs deben montarse sobre un protoboard o una placa y programarse directamente utilizando los pines de programación ICSP: PGD y PGC, estos pines varían de acuerdo a cada modelo por lo que debemos revisar el datasheet correspondiente.

3.1.3.1 MODOS DE USO

PICkit2 como Programador: Lee y graba microcontroladores utilizando directamente el software original de Microchip: MPLAB o PICKit 2.

PICkit2 como Debugger: Depura tus programas en tiempo real, Corre el programa paso a paso, establece breakpoints, mira las entradas, salidas y registros del micro.

PICkit2 como Analizador Lógico: Utiliza el PICKit 2 como analizador lógico de dos canales.

PICkit2 como UART Tool: Utiliza el PicKit2 como interfaz serie para comunicar con algún microcontrolador.

Funcional con el entorno de desarrollo integrado MPLAB IDE, conexión USB plug & play.

Compatible con sistemas operativos (32bits/64bits): Windows XP, Windows Vista, Windows 7, y Windows 8

3.1.3.2 ESPECIFICACIONES TÉCNICAS

- Interfaz de comunicación: USB
- Compatible con Windows 10, 8, 7, XP.
- Puerto USB tipo Mini USB
- Puerto de programación ICSP
- Interface USB (Full speed 12 Mbits/s)
- Ejecución en tiempo real
- Compatible con el IDE MPLAB
- Monitor de sobre voltaje y sobre corriente incorporado
- Firmware actualizable desde Internet
- Caja plástica totalmente cerrada
- Soporta bajo voltaje hasta 2.0 V (Rango 2.0 V a 6.0 V)
- LEDs de diagnóstico (power, busy, error)
- Pulsador para poner en modo de actualización de firmware y para la función Programmer-to-Go
- Lectura y escritura de las memorias de programa y datos de los microcontroladores
- Borrado del espacio de memoria de programa con verificación
- Congelamiento de los periféricos en los breakpoints
- Programación de hasta 512 kbyte de memoria flash con la función Programmer-to-Go
- Soporta la familia PIC32
- Compatible con Windows XP/Vista/7/8/8.1/10, MAC, Linux



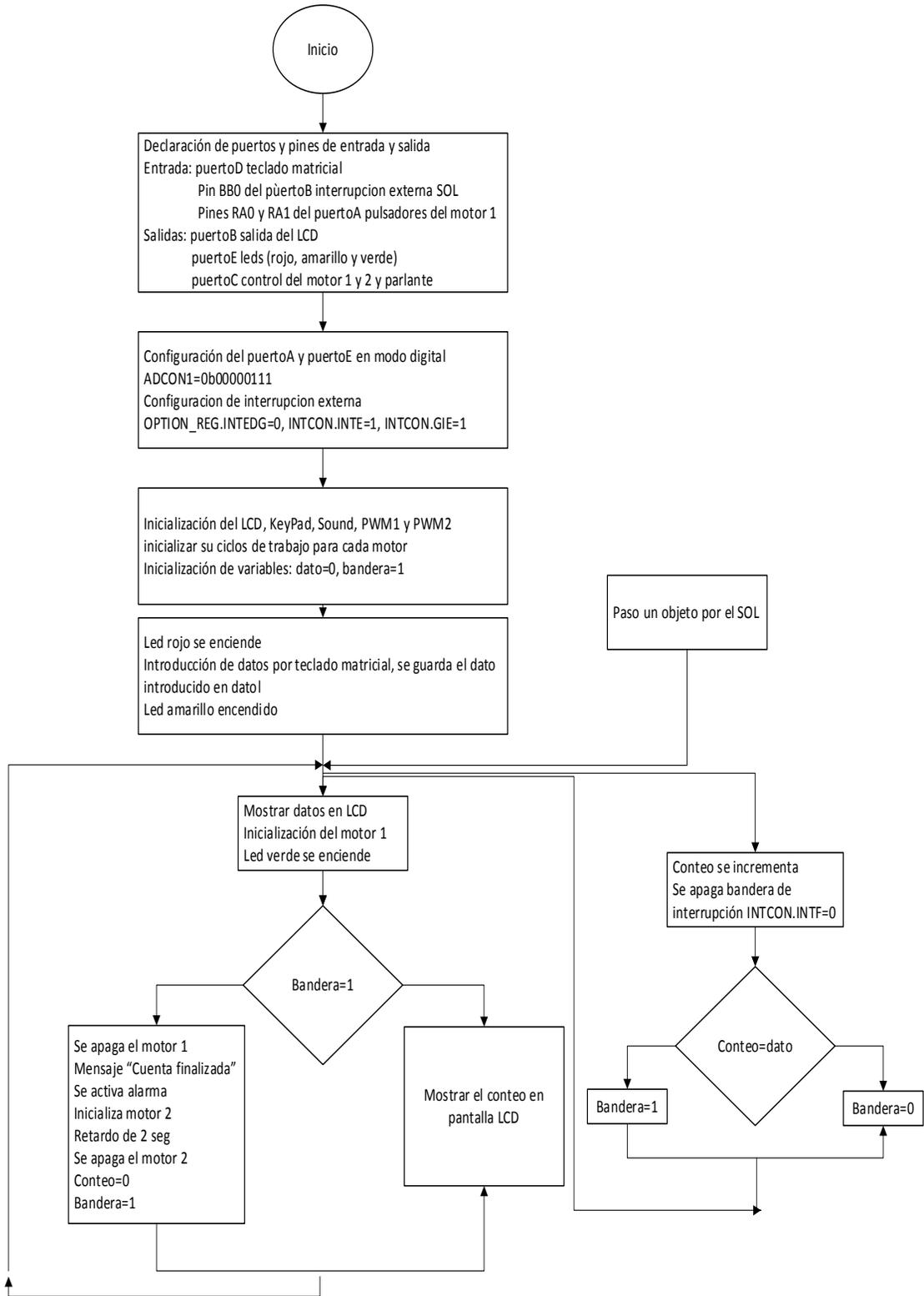
Figura 3.3 Grabador PICKit 2

Fuente: <https://naylampmechatronics.com/programadores/54-pickit-2-clone-programador-pic.html>

3.3 Diagramas de flujo

3.3.1 Diagrama de flujo del programa principal

El siguiente diagrama muestra la secuencia de pasos que se realizará para el programa principal, en esta sección muestra los puertos y pines que trabajaran como entradas o salidas, el tipo de configuración para los puertos que ocuparan los dispositivos de lectura y escritura, configuración para la interrupción externa, comparaciones de datos introducidos y obtenidos, inicialización de dispositivos y acciones para realizar una tarea específica.



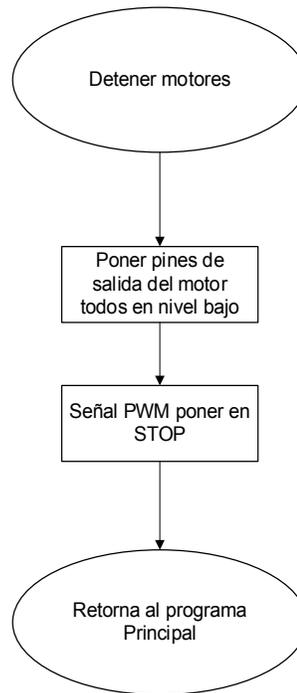
3.3.2 Diagrama de flujo de Inicio de motores

El diagrama de inicio de motores nos muestra la iniciación del motor DC, este es un subproceso en el cual es llamado por el programa principal, como objetivo tiene poner un pin del puerto que está trabajando en estado alto y el otro en estado bajo, dependiendo del motor con el que se está trabajando si es el motor 1 o el motor 2, de eso dependerá que pines o puertos se habilitaran, también esta inicializa la señal PWM con el ciclo de trabajo especificado en el programa principal.



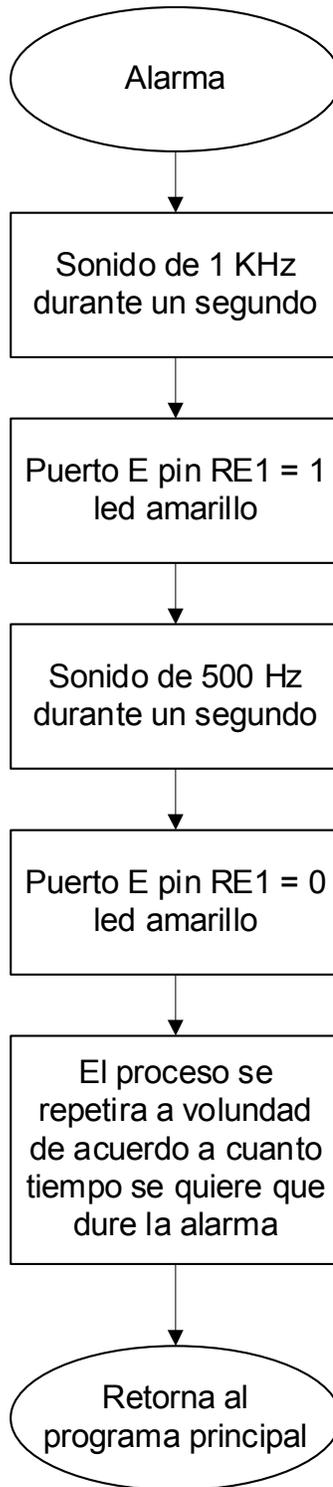
3.3.4 Diagrama de flujo para detener motores

El diagrama muestra la detención del motor DC, este subprograma pone en nivel bajo ambos pines que manda a la salida del motor y además detiene la señal PWM el cual hace la detención total del motor, este subprograma es llamado por el programa principal cuando es requerida y su configuración específica de que pines mandará a nivel bajo dependerá de que motor se esté utilizando si es del motor 1 o del motor 2.



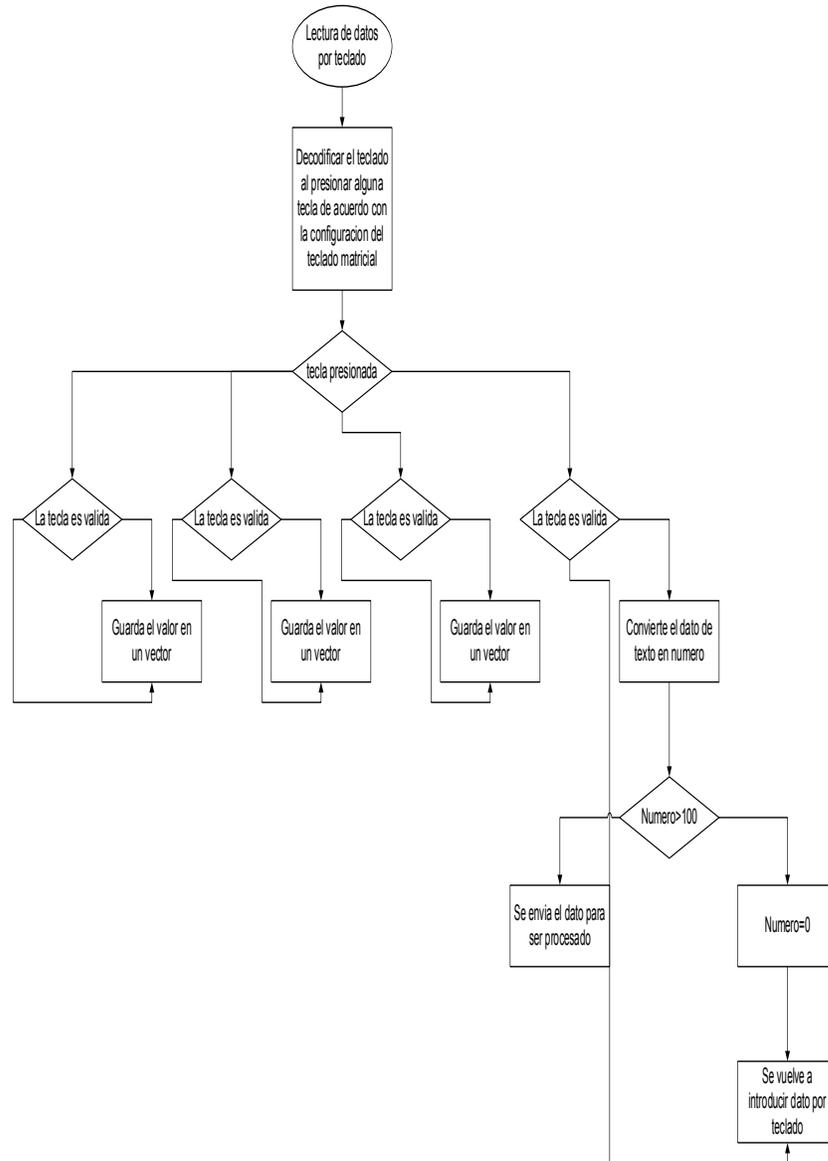
3.3.5 Diagrama de flujo para la alarma

En este diagrama muestra el proceso de alarma aplicado en un subprograma y llamado por el programa principal cuando se lo requiera, el proceso hace que se realice una secuencia de sonidos junto con el parpadeo de un led, esta secuencia de sonidos cambia de frecuencias alternándose secuencialmente, cada frecuencia dura un tiempo determinado y esta será enviado mediante un pin de PIC para que un parlante reciba esta señal y lo haga audible para el usuario.



3.3.6 Diagrama de flujo para leer datos por teclado matricial

El diagrama muestra la lectura de datos que se introducirán por el teclado matricial, este consta de un decodificador de teclado de acuerdo al implementado físicamente, al presionar una tecla esta decodifica que tecla es y si esta es válida se almacena temporalmente hasta introducir los restantes, una vez culminada la introducción de datos se convierte en un numero entero es cual esta es verificada en los rangos de 1 a 100, si el valor no está en ese rango nuevamente se debe introducir los datos, una vez culminada esta tarea el dato es mostrado por pantalla para realizar los siguientes procesos.



3.4 Utilización de librerías de MikroPRO C for pic

Para la realización del proyecto se recurrió a utilizar las librerías que posee el compilador, estas librerías llevan su propia sintaxis en el compilador y posee ejemplos en el cual indica como se puede aplicarlas, además nos indicada el modo y acción que llevan cada una de estas librerías.

Se utilizaron las siguientes librerías:

- Conversions
- C_Math
- C_Stdlib
- C_String
- C_Type
- KeyPad4x4
- Lcd
- PWM
- Sound

3.5 Diseño en Proteus ISIS

Para el diseño del proyecto se utilizo esta herramienta de Proteus ISIS el cual se puede simular y ver el funcionamiento, aquí se implemento la parte de diseño de software y hardware del proyecto el cual me sera de guia para poder implementarlo en la maqueta para su funcionamiento.

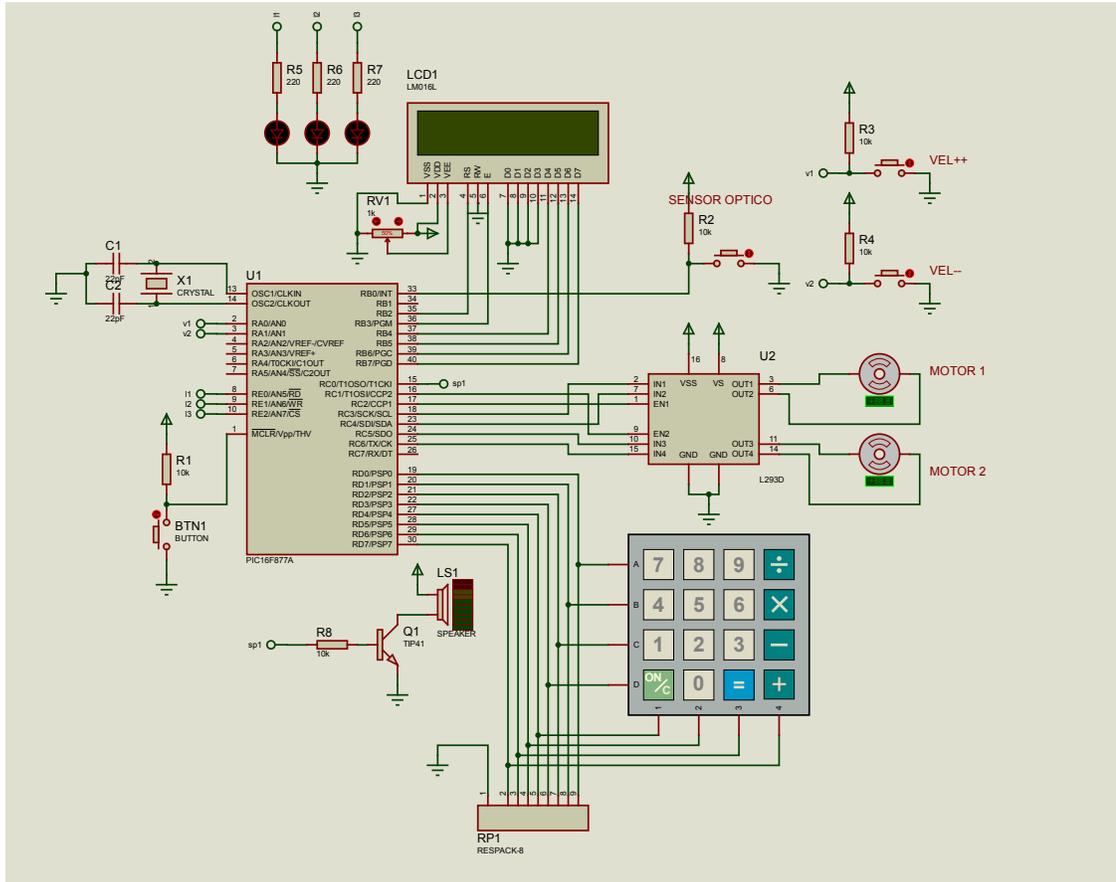


Figura 3.4 Diagrama esquemático de la Banda Transportadora

Fuente: Elaboracion Propia

3.6 Desarrollo del código

Para desarrollar el código se utilizó el lenguaje de programación C, junto al compilador mikroC PRO for pic, el cual el programa realiza las subrutinas, el programa de la interrupción y el programa principal, estos trabajan conjuntamente para desarrollar todo el proceso de la banda transportadora.

```
// Conexiones del LCD
sbit LCD_RS at RB2_bit;
sbit LCD_EN at RB3_bit;
```

```
sbit LCD_D4 at RB4_bit;
sbit LCD_D5 at RB5_bit;
sbit LCD_D6 at RB6_bit;
sbit LCD_D7 at RB7_bit;
```

```
sbit LCD_RS_Direction at TRISB2_bit;
sbit LCD_EN_Direction at TRISB3_bit;
sbit LCD_D4_Direction at TRISB4_bit;
sbit LCD_D5_Direction at TRISB5_bit;
sbit LCD_D6_Direction at TRISB6_bit;
sbit LCD_D7_Direction at TRISB7_bit;
```

```
char keypadPort at PORTD; //Puerto de teclado matricial
```

```
unsigned short dato; //declaracion de variable para datos de entrada
char datoStr[4]; //para convertir dato en texto
unsigned short cont; //declaracion de variable para conteo
char contStr[4]; //para convertir conteo en texto
bit bandera; //declaracion de variable para bandera
unsigned short ciclo1; //declaracion del ciclo de trabajo para motor 1
char ciclo1Str[4]; //convertir el ciclo del motor 1 en texto
unsigned short ciclo2; //declaracion del ciclo de trabajo para motor 2
char ciclo2Str[4]; //convertir el ciclo del motor 2 en texto
```

```
char tecla; //tecla para introducir datos
char texto[10]; //convertir la tecla en texto
```

```
char retardo_cad[10]; //vector para almacenar
```

```
//maquina de estado para capturar varias teclas
```

```
enum teclado_estado {n1,n2,n3,en};
```

```
char estado_teclado=n1;
```

```
//decodificacion para teclado
```

```
char teclado_deco()
```

```
{
```

```
    char tec=0;
```

```
    tec = Keypad_Key_Click();
```

```
    if (tec!=0)
```

```
    {
```

```
        switch (tec)
```

```
        {
```

```
            case 1:tec='1';break;
```

```
            case 2:tec='4';break;
```

```
            case 3:tec='7';break;
```

```
            case 4:tec='*';break;
```

```
            case 5:tec='2';break;
```

```
            case 6:tec='5';break;
```

```
            case 7:tec='8';break;
```

```
            case 8:tec='0';break;
```

```
            case 9:tec='3';break;
```

```
            case 10:tec='6';break;
```

```
            case 11:tec='9';break;
```

```
            case 12:tec='#';break;
```

```
            case 13:tec='A';break;
```

```
            case 14:tec='B';break;
```

```
            case 15:tec='C';break;
```

```
            case 16:tec='D';break;
```

```

    }//Fin del switch
    return (tec);

}

}

//lectura del teclado de 1 a 100
void teclado_estado()
{
    tecla=teclado_deco();

    switch (estado_teclado)
    {
        case n1:

            if ((tecla!=0) && (isdigit(tecla)))
            {
                retardo_cad[0]=tecla;
                estado_teclado=n2;
                Lcd_Chr(1,8,tecla);
                Delay_ms(400); // retardo antirrebote
            }
            break;

        case n2:
            if ((tecla!=0) && (isdigit(tecla)))

```

```

{
    retardo_cad[1]=tecla;
    estado_teclado=n3;
    Lcd_Chr(1,9,tecla);
    Delay_ms(400); // retardo antirrebote
}
break;

```

case n3:

```

if(((tecla!=0) && (isdigit(tecla)))
{
    retardo_cad[2]=tecla;
    estado_teclado=en;
    Lcd_Chr(1,10,tecla);
    Delay_ms(400); // retardo antirrebote
}
break;

```

case en:

```

if(((tecla!=0) && (tecla!='A')) // no se presiono el igual
{
    estado_teclado=n1; // vuelve al estado de captura del primer numero
    Lcd_Out(1,8," "); // borra la pantalla
    Delay_ms(600); // retardo antirrebote
}

```

```

// si si se presiono el enter
else if ((tecla!=0) && (tecla=='A')) //si se presiono el igual
{
    retardo_cad[3]=0; //final de cadena
    dato=atoi(retardo_cad); // lo pasa a numero y calcula el retardo deseado
    if(dato>100)
        dato=0;
    estado_teclado=n1; // vuelve al estado de captura del primer numero.
    Lcd_Out(1,8," "); // borra la pantalla
    Delay_ms(600); // retardo antirrebote
}

break;

}

}

//interrupcion externa puerto RB0 SENSOR OPTICO LASER-LDR
void interrupt() iv 0x0004 ics ICS_AUTO {

    INTCON.INTF=0; //apagar bandera de interrupcion
    cont++; //se incrementa el conteo
    if(cont>=dato) //si el conteo ya culmino bandera se pone en 1
        bandera=0; //para finalizar el conteo
}

//regula la velocidad del motor 1
void motor1(){

```

```

if(!PORTA.F0)          //velocidad ++
{
    Delay_ms(500);

    if(ciclo1>=255)
    {
        ciclo1=255;
    }
    else
        ciclo1=ciclo1+5;
    PWM1_Set_Duty(ciclo1);
}
if(!PORTA.F1)          //velocidad --
{
    Delay_ms(500);

    if(ciclo1<1)
        ciclo1=0;
    else
        ciclo1=ciclo1-5;
    PWM1_Set_Duty(ciclo1);
}
ByteToStr(ciclo1,ciclo1Str);
Lcd_Out(2,8,"Vel: ");
Lcd_Out(2,13,ciclo1Str);
}
//detiene motor 1
void motorStop1(){
    PORTC.F3=0;

```

```

    PORTC.F4=0;
    PWM1_Stop();
}

//inicializa el motor 1
void motorStar1(){
    PORTC.F3=1;
    PORTC.F4=0;
    PWM1_Start();
}

//detiene el motor 2
void motorStop2(){
    PORTC.F5=0;
    PORTC.F6=0;
    PWM2_Stop();
}

//inicializa el motor 2
void motorStar2(){
    PORTC.F5=1;
    PORTC.F6=0;
    PWM2_Start();
}

//alarma de finalizacion de conteo
void tono(){
    PORTE.F1=1;
    Sound_Play(1000, 1000);
}

```

```

PORTE.F1=0;
Sound_Play(500, 1000);
PORTE.F1=1;
Sound_Play(1000, 1000);
PORTE.F1=0;
Sound_Play(500, 1000);
PORTE.F1=1;
Sound_Play(1000, 1000);
PORTE.F1=0;
Sound_Play(500, 1000);
PORTE.F1=1;
}

```

//programa principal

```

void main() {
    TRISB.F0=1;      //pin puerto B, RB0 entrada de la interrupcion con SOL (sensor
optico laser-LDR)
    TRISC=0;        //puerto C configurado como salida para el control de los motores
1 y 2 y parlante
    TRISA.F0=1;     //pin puerto A, RA0 configurado como entrada del pulsador del
motor vel++
    TRISA.F1=1;     //pin puerto A, RA1 configurado como entrada del pulsador del
motor vel--
    TRISE=0;        //puerto E configurado como salida, muestra los leds rojo, amarillo
y verde

    PORTB.F0=0;     //limpiar todos los puertos y pines que se estan utilizando
    PORTC=0;
    PORTA.F0=0;

```

```

PORTA.F1=0;
PORTC=0;
PORTE=0;

ADCON1=0b00001111;           //configuracion del puerto A y puerto E en modo
digital

OPTION_REG.INTEDG=0;         //configuracion de INTEDG por flanco de bajada
INTCON.INTE=1;               //habilitar bit de permiso de interrupcion externa por
RB0/INT
INTCON.GIE=1;                //habilitar bit de permiso global de interrupciones

Lcd_Init();                  //Iniciar LCD
PORTE.F0=1;                  //encender Led Rojo
Lcd_Cmd(_LCD_CURSOR_OFF);   //apagar cursor del LCD
Lcd_Out(1,4,"PROYECTO");     //mostrar mensaje en LCD "PROYECTO PIC
16F877A"
Lcd_Out(2,3,"PIC 16F877A");
Delay_ms(3000);              //retardo de 3 segundos
Keypad_Init();               //inicializar el teclado matricial
Lcd_Cmd(_LCD_CLEAR);         //limpiar pantalla LCD
PORTE.F1=1;                  //encender Led Amarillo
Lcd_Out(1,1,"Dato: ");       //introducir numero para el conteo de objetos
dato=0;
Lcd_Cmd(_LCD_BLINK_CURSOR_ON);
while (dato==0)
{
    teclado_estado();         //lee los datos de teclado
}

```

```

ByteToStr(dato,datoStr);    //convierte el dato a texto
Lcd_Out(1,7,datoStr);      //muestra en LCD en dato introducido
Lcd_Cmd(_LCD_CURSOR_OFF); //apaga el cursor del LCD

cont=0;                    //variable de conteo
bandera=1;                 //bandera para saber si se finalizo el conteo
ciclo1=200;                //inicio de ciclo de trabajo del motor 1
ciclo2=125;                //inicio de ciclo de trabajo del motor 2
PWM1_Init(245);           //iniciar PWM 1 para el motor 1 en 245Hz
PWM1_Set_Duty(ciclo1);    //escribir el ciclo de trabajo para el motor 1
PWM2_Init(245);           //iniciar PWM 1 para el motor 1 en 245Hz
PWM2_Set_Duty(ciclo2);    //escribir el ciclo de trabajo para el motor 2

Sound_Init(&PORTC, 0);    //iniciar sonido para el puertoC pin RC0
bucle:                     //bucle infinito

    PORTE.F1=0;           //se apaga el Led amarillo
    PORTE.F2=1;           //se enciende el Led verde
    motorStar1();         //se incia el motor 1
    motor1();             //se regula la velocidad del mortor 1

    if (bandera)          //mientras la bandera este en 1 realizara el conteo
    {
        ByteToStr(cont,contStr); //convertir el conteo en texto
        Lcd_Out(2,1,contStr);    //mostrar conteo en LCD
    }
    else
    {
        ByteToStr(cont,contStr); //convertir el ultimo conteo en texto

```

```

    Lcd_Out(2,1,contStr); //mostrar el ultimo conteo en LCD
    Lcd_Cmd(_LCD_CLEAR); //Limpiar todo el LCD
    Lcd_Out(1,6,"CUENTA"); //muestra mensaje "CUENTA FINALIZADA"
    Lcd_Out(2,4,"FINALIZADA");

    PORTE.F2=0; //se apaga el Led verde
    motorStop1(); //se detiene el motor 1
    tono(); //alarma de conteo finalizado

    motorStar2(); //inicializa el motor 2
    Delay_ms(2000); //retardo para que el motor 2 recoja los objetos
contados
    motorStop2(); //se detiene el motor 2
    Lcd_Cmd(_LCD_CLEAR); //se hace una limpieza del LCD
    Lcd_Out(1,1,"Dato: "); //se vuelve a mostrar el dato introducido
    Lcd_Out(1,7,datoStr);
    cont=0; //se inicializa de nuevo el conteo
    bandera=1;
}
goto bucle;
}

```

3.7 Diseño de maqueta

Para el diseño de la maqueta se utilizaron materiales como ser melamina delgada, rodamientos reciclados de impresoras, bandas de goma recicladas de equipos de electrónica, madera en tiras, silicona en barra, alambres de conexión y los componentes electrónicos.



Figura 3.5 Maqueta de banda transportadora

Fuente: Elaboración Propia

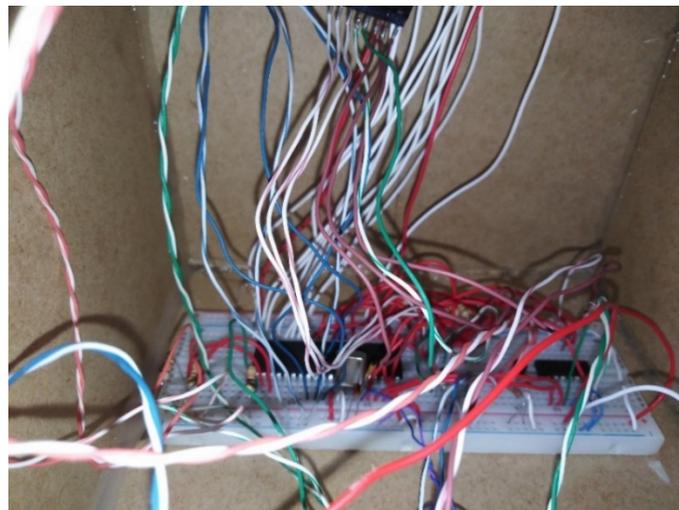


Figura 3.6 Conexiones del protoboard montadas en la maqueta

Fuente: Elaboración Propia



Figura 3.7 Primera y segunda banda transportadora de la maqueta
Fuente: Elaboración Propia

CAPITULO IV

ANALISIS DE COSTOS

4.1 Costos Fijos

En esta sección hablaremos de los costos de los componentes y demás objetos que fueron de ayuda para realizar el prototipo de banda transportadora, se implementara una tabla el cual describa los costos y un total de todo el proyecto elaborado.

Descripción	Cantidad	Costo
Grabador PICKit 2	1	160 Bs
Zócalo de Zif	1	25 Bs
PIC16F877A	1	35 Bs
Motores DC	2	30 Bs
Led Infrarrojo	1	9 Bs
LDR	1	3 Bs
Transistor BC558	1	1 Bs
Transistor BC548	1	1 Bs
Transistor TIP 41	1	3 Bs
Parlante o altavoz	1	15 Bs
Cristal	1	5 Bs
Pulsador	3	3 Bs
Pantalla LCD	1	25 Bs
Leds	3	1 Bs
Capacitores cerámicos	2	1 Bs
Teclado Matricial	1	15 Bs
Resistencias	16	4 Bs
L293D	1	10 Bs
Alambres de conexión	1	10 Bs
Maqueta	1	50 Bs
Total		402 Bs

Tabla 4.1 Costos de los componentes y material utilizado

Fuente: Elaboración Propia

CAPITULO V

CONCLUSIONES Y RECOMENDACIONES

5.1 Conclusiones

- Se llego a implementar el prototipo banda transportadora de ladrillos utilizando el PIC16F877A, el sistema realiza el transporte de ladrillos de un punto a otro cumpliendo así lo esperado regulando a la vez la velocidad de la banda transportadora, se puede introducir el dato para el conteo por medio del teclado matricial y poder visualizar los datos introducidos y el conteo por medio de la pantalla LCD, al final del conteo realiza por una segunda banda el recojo de todos los ladrillos contados.
- Se utilizo el lenguaje de programación C de alto nivel para la programación del PIC con ayuda del compilador de PICs mikroC PRO for pic y realizando simulaciones de funcionamiento con ayuda de Proteus ISIS, así mismo se grabó el código generado con el PICKit 2 y posteriormente implementarlo en una maqueta el cual muestra el funcionamiento de todo el proceso.
- Se desarrollo el sistema de conteo por medio de un sensor óptico laser – LDR el cual por medio de la interrupción externa que nos ofrece el PIC16F877A realiza el incremento del conteo comparando con del dato introducido por teclado para que realice el proceso de conteo y pueda desarrollar toda la parte del proceso del transporte de los ladrillos.

5.2 Recomendaciones

- En la parte de la banda transportadora hay que tener cuidado que esta no esté muy tensionada ya que esto provoca que el motor DC no pueda tener la fuerza de iniciación para que la banda pueda realizar el transporte de los ladrillos, hay que regular esta tensión para que no ocurra este problema.
- Para el sensor óptico laser – LDR hay que seleccionar un LDR el cual no le afecte tanto la luminosidad del ambiente el cual provocara que no realice el conteo correspondiente, de preferencia hay q obstruirlo de la luminosidad del ambiente para evitar este problema

REFERENCIAS BIBLIOGRAFICAS

Bibliografía

- Bodington, C. (2010). *Basic para microcontroladores PIC*. Venezuela: WWW.CONEXIONELECTRONICA.COM.
- Clavijo, J. R. (2011). *Diseño y simulacion de sistemas microcontrolados en lenguaje C*. Colombia: ISBN 978-958-44-8619-6.
- Del Valle, L. (2015). *Programar Facil*. Recuperado el 7 de noviembre de 2022, de <https://programarfacil.com>
- Ecured. (2019). *EcuRed*. Recuperado el 7 de noviembre de 2022, de <https://www.ecured.cu>
- Electrónica, M. (2018). *Mv Electronica*. Recuperado el 7 de noviembre de 2022, de <https://mvelectronica.com/producto/microcontrolador-pic16f877a>
- Futurlec. (2022). *Futurlec*. Recuperado el 7 de noviembre de 2022, de <https://www.futurlec.com>
- Garcia, V. (13 de novimebre de 2010). *Electronica Practica Aplicada*. Recuperado el 7 de noviembre de 2022, de <https://www.diarioelectronicohoy.com/blog/pulsadores-sin-rebotes>
- Guachalla, R. (2015). *Laboratorio de Microprocesadores I*. La Paz.
- Macatronica. (23 de abril de 2021). *MecatronicaLATAM*. Recuperado el 7 de noviembre de 2022, de <https://www.mecatronicalatam.com>
- Marmolejo, R. (2021). *Heptro*. Recuperado el 7 de noviembre de 2022, de <https://hetpro-store.com/TUTORIALES/microcontrolador/>
- Naylamp, M. (2021). *Naylamp Mechatronics SAC*. Recuperado el 7 de noviembre de 2022, de <https://naylampmechatronics.com>
- Palacios, E., Romiro, F., & Lopez, L. (2004). *Microcontrolador PIC16F84*. Mexico: Alfaomega.
- Rapetti, G. (2017). *Inventable.Eu*. Recuperado el 7 de noviembre de 2022, de <https://www.inventable.eu>

Visual, L. (2022). *Visual Led*. Recuperado el 7 de noviembre de 2022, de <https://visualled.com>

Willy, F. (10 de febrero de 2013). *TodoElectrodo*. Recuperado el 7 de noviembre de 2022, de <http://todoelectrodo.blogspot.com/2013/02/lcd-16x2.html>