

ESCUELA POLITÉCNICA NACIONAL

ESCUELA DE FORMACIÓN DE TECNÓLOGOS

DISEÑO Y CONSTRUCCIÓN DE UN ODÓMETRO DIGITAL.

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

TORRES TORO VICTOR RICARDO
victor_riccardot@hotmail.com

DIRECTOR: ING. PABLO LÓPEZ
pablo.lopez@epn.edu.ec

Quito, Noviembre del 2008

DECLARACIÓN

Yo, Torres Toro Víctor Ricardo, declaro bajo juramento que el trabajo aquí descrito es de mi autoría; que no ha sido previamente presentado para ningún grado o calificación profesional; y, que he consultado las referencias bibliográficas que se incluyen en este documento.

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

Torres Toro Victor Ricardo

CERTIFICACIÓN

Certifico que el presente trabajo fue desarrollado por Victor Ricardo Torres Toro, bajo mi supervisión.

Ing. Pablo López
DIRECTOR DEL PROYECTO

DEDICATORIA

El presente trabajo esta dedicado en primer lugar a Dios, por permitirme habitar este planeta, con todos mis sentidos en perfecto estado, luego a mi madre, por haberme traído al mundo a pesar de las diversas circunstancias que existían en su entorno, además por todo su apoyo para realizarme como profesional.

A mi Papá José que desde el cielo se que siempre me estuvo dando la fuerza necesaria para no decaer en el camino y por todo su ejemplo a seguir, a mi Mamá Bacha por todos los años compartidos como un hijo más, a mis tíos y tías, y en especial a Dayana y Elihann, porque es por ellos que seguiré luchando por alcanzar mis metas, por ser la razón y esencia de mi existir.

AGRADECIMIENTOS

A mi Dios, por ser la fuerza espiritual que guía mi camino, a mis dos mamás Maryuri y Bacha por su apoyo incondicional en mi vida, a mis tíos Ani y Laura por su entrega desinteresada en los años de universidad, a mis hermanos Gela, Homero, Cristina y Diana por todos los momentos compartidos que nos hacen sentir el lazo de hermandad que siempre perdurara, a mis primos Karina, Pablo y Denisse.

Un agradecimiento especial al Ing. Pablo López, por ser el profesional responsable de dirigir el presente trabajo, además por el apoyo frontal desde el nacimiento de la idea que se cristalizó en la realización del proyecto que se ha concluido.

A Dayana por ser mi fuente de vida y la madre de mi hijo, por quien me faltaría vida para agradecer todo lo que ha hecho por Elihann y en especial por mi por hacerme creer que todo lo que se pide con devoción se logra y es posible alcanzar.

A Ricardo Calderón por formar parte de mi vida y darme la fuerza para regresar a estudiar y poder culminar un proceso que pensé estaba perdido.

RESUMEN

Cada vez se hace más familiar el mundo de la electrónica, pues en la mayoría de áreas es muy frecuente ver el incremento de dispositivos electrónicos, nuestro país también avanza de una manera acelerada al mundo fantástico de la electrónica.

En los últimos años, la utilización de dispositivos digitales, han ido desplazando a los dispositivos mecánicos, pues por poseer sensores electrónicos reemplazan a partes mecánicas y visualizan con más precisión la medida realizada optimizando tiempo y recursos humanos.

Es así que en recorridos en los cuales hay que medir distancia es mucho más sencillo tomar lecturas directas y guardar datos, con mucha más facilidad en dispositivos digitales que permiten el almacenamiento de los mismos, cosa que no lo permitían los antiguos medidores mecánicos que debían ser encerados antes de realizar una medida.

Es de esta forma que surge la idea de implementar un dispositivo capaz de transformar datos mecánicos a datos digitales, al cual se lo ha denominado Odómetro Digital, que está basado principalmente en opto acopladores y un microcontrolador que se encarga de todo el proceso de conversión.

La ejecución del presente trabajo inicia con la investigación de los medidores mecánicos de distancias, así como definiciones y teoría de los dispositivos electrónicos que formaran parte integral del presente proyecto, y por último la obtención de resultados, pruebas, conclusiones y recomendaciones para mejoras del trabajo en tiempos futuros, por compañeros a los cuales les interesaría esta investigación.

INDICE

Contenido	Pagina
CAPITULO I	1
Marco Teórico Conceptual.	1
1.1.- Medidores de Distancias. (Odómetro)	1
1.1.1.- Historia del Odómetro.	1
1.1.2.- Tipos de Odómetros.	4
1.1.2.1.- Odómetro Mecánico.	4
1.1.2.2.- Odómetro Digital.	5
1.2.- Microcontroladores	8
1.2.1.- Diferencia entre Microprocesador y Microcontrolador.....	9
1.2.2.- Aplicaciones de los Microcontroladores.	10
1.2.3.- El mercado de los Microcontroladores.	11
1.2.4.- ¿Qué Microcontrolador Emplear?.	12
1.2.5.- Recursos comunes a todos los Microcontroladores.	13
1.2.5.1.- Arquitectura básica.	13
1.2.5.2.- El procesador o UCP.	14
1.2.5.3.- Memoria.	15
1.2.5.4.- Puertos de entrada y salida.	18
1.2.5.5.- Reloj principal.	18
1.2.6.- Recursos especiales.	19
1.2.6.1.- Temporizadores o “Timers”.	19
1.2.6.2.- Perro guardián o “Watchdog”.	20
1.2.6.3.- Protección ante fallo de alimentación o “Brownout”.	20
1.2.6.4.- Estado de reposo o de bajo consumo.	20
1.2.6.5.- Conversor A/D (CAD).	21
1.2.6.6.- Conversos D/A (CDA).	21
1.2.6.7.- Comparador analógico.	21

1.2.6.8.- Modulador de anchura de impulso o PWM.	21
1.2.6.9.- Puertos de E/S digitales.	22
1.2.6.10.- Puertos de comunicación.	22
1.2.7.- Herramientas para el desarrollo de aplicaciones.	22
1.2.7.1.- Desarrollo del software.	23
1.3.- Opto Acoplador.	25
1.3.1.- Tipos.	26
1.3.2.- Encapsulados.	26
1.4.- Display de Cristal Líquido (LCD).	28
1.4.1.- Display de Cristal Liquido de 2 líneas de 16 caracteres.	31
1.4.1.1.- Características del Display.	31
1.4.1.1.1.- Aspecto físico.	31
1.4.1.1.2.- Alimentación.	32
1.4.1.1.3.- Los caracteres del LCD.	32
1.4.1.1.4.- La memoria del LCD.	33
1.4.1.1.4.1.- DD RAM. (Display Data Ram).	33
1.4.1.1.4.2.- La CG RAM. (Carácter Generador Ram).....	36
1.4.1.2.- Interfaz Hardware.	38
1.4.1.2.1.- Asignación Pines.	38
1.4.1.2.2.- El interfaz del display con el mundo exterior.	39
1.4.1.2.3.- El bus de datos.	39
1.4.1.2.4.- El bus de control.	40
1.4.1.2.5.- El control del contraste.	41
1.4.1.2.6.- Temporización.	42
1.5.- Teclado.	45
1.5.1.- Técnicas de barrido del teclado.....	46
1.5.2.- Teclado matricial 4x4.	47

1.6.- Dispositivo Mecánico.	49
1.6.1.- Funcionamiento.	49
 CAPITULO II	 51
 Construcción del Odómetro Digital.	 51
2.1.- Esquema del Odómetro Digital.	51
2.1.1.- Etapa mecánica.	51
2.1.2.- Etapa de transformación de datos.	51
2.1.3.- Etapa electrónica.	51
2.1.4.- Etapa de visualización.	52
2.2.- Partes que conforman el Odómetro Digital.	53
2.2.1.- Partes de la etapa mecánica.	53
2.2.2.- Partes de la etapa de transformación de datos.	53
2.2.3.- Partes de la etapa electrónica.	54
2.2.4.- Partes de la etapa de visualización.	55
2.3.- Elaboración del programa.	56
2.4.- Construcción del Odómetro Digital.	73
2.4.1.- Construcción mecánica.	73
2.4.2.- Montaje electrónico.	74
2.4.3.- Odómetro digital terminado.	81
 CAPITULO III	 82
 Pruebas y Resultados.	 82
3.1.- Pruebas de funcionamiento del Odómetro Digital.	82
3.1.1.- Calibración de opto acopladores.	83
3.2.- Obtención de resultados.	83
3.3.- Análisis de resultados obtenidos.	84
 CONCLUSIONES.	 86
 RECOMENDACIONES.	 87

BIBLIOGRAFIA. 88

ANEXOS. 89

CAPITULO I

Marco Teórico Conceptual.

1.1.- Medidores de Distancias (Odómetro).

Se llama así al aparato que se utiliza para medir la distancia recorrida entre dos puntos, o el camino recorrido entre dos puntos.

Los odómetros llevan mucho tiempo de existir en su forma mecánica y recientemente, los dispositivos digitales han empezado a reemplazarlos.

En este trabajo, exploraremos que hacen los odómetros mecánicos para llevar medición exacta de la distancia y luego hablaremos sobre los odómetros digitales.

El aparato que utilizan los automóviles para registrar exactamente la distancia que recorren de un lugar a otro es probablemente lo primero que pensamos cuando escuchamos la palabra odómetro, aunque fue inventado mucho antes que el automóvil y es utilizado en muchos otros vehículos, incluso bicicletas.

Su nombre se deriva de dos vocablos griegos, “odos” que significa camino y “metro” que quiere decir medir, es decir “que mide el camino”.

1.1.1.- Historia del odómetro

De acuerdo a la enciclopedia Británica, alrededor del año 15 A.C. el arquitecto e ingeniero romano Vitruvius montó una rueda de circunferencia conocida en un pequeño marco y lo configuró de tal manera que cuando era halado este dejaba caer una pequeña piedra en un contenedor por cada revolución de la rueda, siendo este uno de los primeros odómetros conocidos.

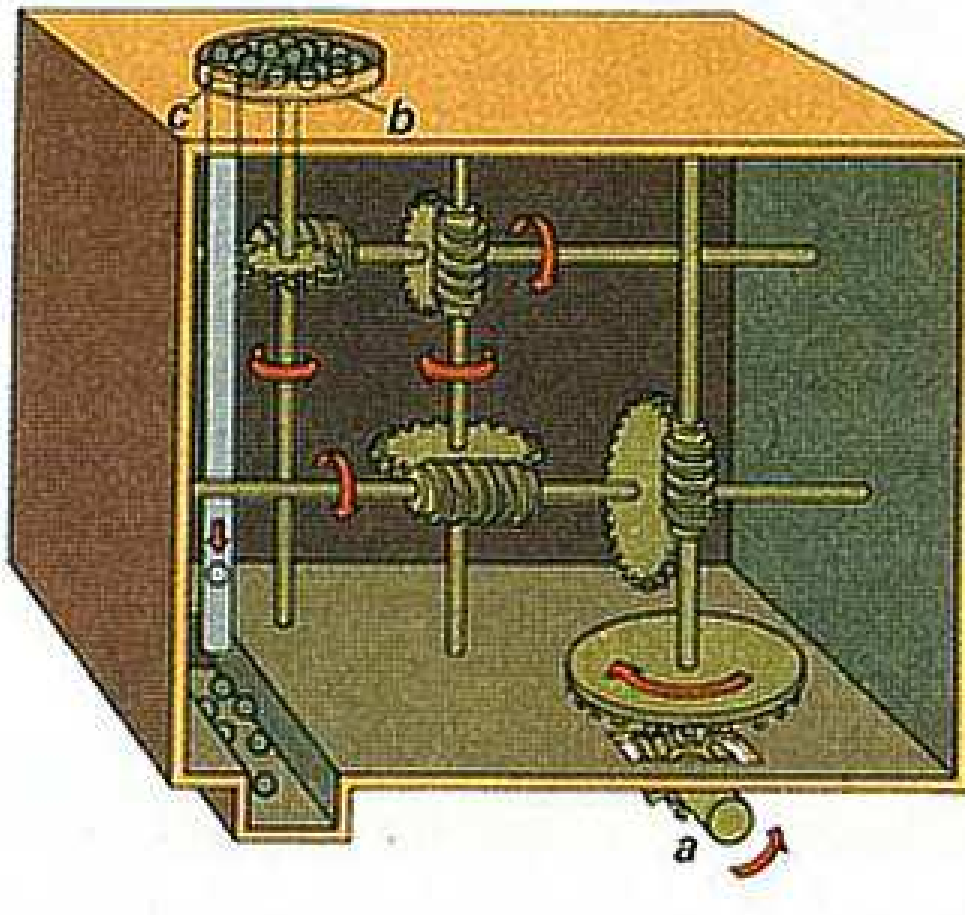


Figura 1.1.- Odómetro de Marcus Vitruvius

Thomas Savery, un ingeniero militar inglés e inventor patentó en 1698 el primer odómetro para barcos.

William Clayton, un pionero mormón inventó un aparato al que llamó "roadometer" para llevar un conteo de las revoluciones de las ruedas de las carretas que usaban los pioneros, el había determinado que 360 revoluciones de una rueda de carreta equivalían a una milla.

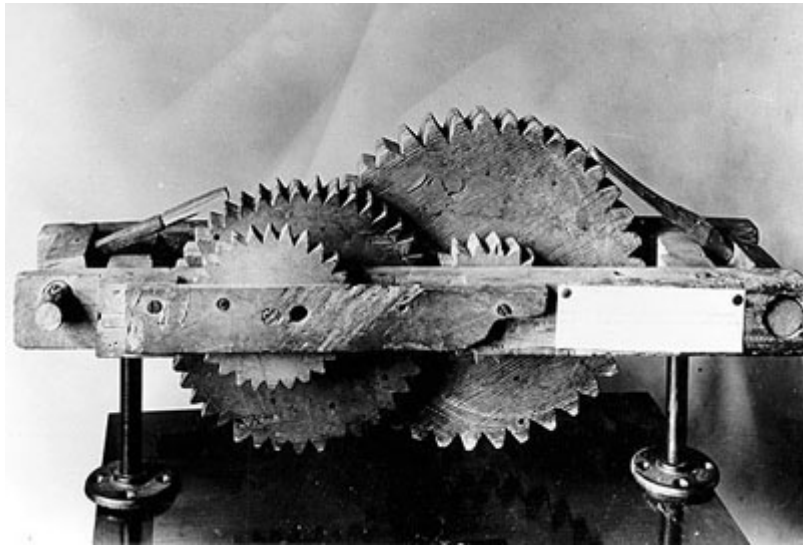


Figura 1.2.- Roadometer

Benjamín Franklin en 1775 cuando trabajaba para la oficina de correos, se le asignó la tarea de optimizar las rutas de reparto de los carteros, entonces salió con su carruaje a medir las distancias entre los diferentes pueblos a los que había que repartir correo y se dio cuenta de cuan importante era tener un instrumento que pudiera medir las distancias.

Diseñó entonces un aparato que media las revoluciones del eje de las ruedas de su carruaje y lo ajustó para que sonara una campana cada 20 revoluciones, curiosamente esto es el equivalente actual a 5.02 metros.

Otras fuentes le atribuyen la invención del odómetro a Samuel McKeen, en 1854.



Figura 1.3.- Odómetro antiguo

1.1.2.- Tipos de Odómetros

Para el presente trabajo se los ha dividido de la siguiente manera:

1.1.2.1.- Odómetro Mecánico.-

Los odómetros mecánicos están formados por una serie de engranajes que se encuentran conectados entre sí y logran una reducción de hasta 1690:1 por medio de engranajes en espiral o “de gusano” y engranajes normales.



Figura 1.4.- Engranaje de gusano

Una rueda que al girar, está conectada en un extremo con un engranaje de gusano que impulsa a otro engranaje, este a su vez mueve otros engranajes de gusano el cual se conecta con otro engranaje normal y así sucesivamente hasta que el ultimo engranaje de gusano mueve directamente el último número en la rueda de números que nosotros podemos ver en la parte frontal que se visualiza, esta rueda es en realidad otro engranaje que está conectada con las otras ruedas de números, que representan las unidades, decenas, centenas, etc. de la distancia recorrida.

Para que todo el mecanismo funcione correctamente, el sistema debe ser cuidadosamente calibrado al momento de su fabricación.

Un dato curioso es que como puedes ver, el sistema no es más que una sencilla unión de engranajes, de manera que si mueves las ruedas del aparato en sentido contrario, puedes hacer que el odómetro regrese su medida también, de hecho, cuando el aparato marcha en reversa, el odómetro efectivamente reduce el conteo de millas o kilómetros recorridos.

1.1.2.2.- Odómetro Digital.-

Si alguna vez has instalado un odómetro para bicicletas, sabes que no hay ningún cable dando vueltas que mueva engranajes, en cambio, se instala en uno de los rayos de la rueda un pequeño imán, y en el poste del tenedor por donde pasa el rayo, se instala un pequeño sensor, cada vez que la rueda gira una vez completa, el sensor envía un impulso a la computadora de navegación de la bicicleta, es necesario programar en ella el tamaño de llantas que tiene la bicicleta y con esta información, la computadora puede calcular la distancia recorrida, agregando la circunferencia de la rueda cada vez que el sensor recibe un impulso.



Figura 1.5.- Odómetro digital

Los automóviles que utilizan odómetros digitales, emplean un sistema similar para registrar la distancia, una rueda metálica dentada se encuentra instalada a la salida de la transmisión y un sensor magnético recibe un impulso cada vez

que pasa uno de los dientes metálicos de la rueda, con esto es posible determinar la distancia, una vez que se ha calibrado el dispositivo.

Algunos automóviles utilizan una rueda perforada y un sensor óptico semejante al que utilizan los mouse de computadoras.



Figura 1.6.- Nótese las ruedas ópticas en este modelo de Mouse antiguo

Un dato interesante es cómo el odómetro digital envía la información a la unidad de control del auto, esta es enviada en un bus de datos común para muchos otros tipos de información que pueden viajar en dicho bus, algunos de estos son:

Información del sistema de climatización.

Información del tablero.

Control de los vidrios eléctricos.

Radio

ABS

Información de la unidad de control de las bolsas de aire.

Módulo de control (se encarga de las luces interiores y otros).

Módulo de control de transmisión.

Muchos vehículos utilizan un protocolo estándar llamado SAE J1850 para permitir que muchos de los diferentes módulos electrónicos se comuniquen entre sí.

Estos paquetes de información son enviados a la computadora central compuestos por un encabezado que consiste en un código que identifica al paquete como información de distancia y el cuerpo del paquete que contiene la distancia recorrida.

Esta información es almacenada en la computadora central para llevar un registro, de manera que no sea posible disminuir la cuenta del odómetro.

Otra computadora instalada en el tablero se encarga de buscar estos paquetes de información de distancia y desplegar la lectura total en el tablero, cada vez que recibe un paquete de distancia solamente suma la distancia que trae el paquete con la que esta desplegando en ese momento y así la lectura es incrementada.

1.2.- Microcontroladores

Los microcontroladores están conquistando el mundo. Están presentes en nuestro trabajo, en nuestra casa y en nuestra vida, en general. Se pueden encontrar controlando el funcionamiento de los ratones y teclados de los computadores, en los teléfonos, en los hornos microondas y los televisores de nuestro hogar. Pero la invasión acaba de comenzar y el nacimiento del siglo XXI será testigo de la conquista masiva de estos diminutos computadores, que gobernarán la mayor parte de los aparatos que fabricaremos y usamos los humanos.

Recibe el nombre de controlador el dispositivo que se emplea para el gobierno de uno o varios procesos. Por ejemplo, el controlador que regula el funcionamiento de un horno dispone de un sensor que mide constantemente su temperatura interna y, cuando traspasa los límites prefijados, genera las señales adecuadas que accionan los actuadores que intentan llevar el valor de la temperatura dentro del rango estipulado.

Aunque el concepto de controlador ha permanecido invariable a través del tiempo, su implementación física ha variado frecuentemente. Hace tres décadas, los controladores se construían exclusivamente con componentes de lógica discreta, posteriormente se emplearon los microprocesadores, que son chips sobre una tarjeta de circuito impreso. En la actualidad, todos los elementos del controlador se han podido incluir en un chip, el cual recibe el nombre de microcontrolador. Realmente consiste en un sencillo pero completo computador contenido en el corazón (chip) de un circuito integrado.

Un microcontrolador es un circuito integrado de alta escala de integración que incorpora la mayor parte de los elementos que configuran un controlador.

Un microcontrolador dispone normalmente de los siguientes componentes:

Procesador o UCP (Unidad Central de Proceso).

Memoria RAM para Contener los datos.

Memoria para el programa tipo ROM/PROM/EPROM.

Líneas de E/S para comunicarse con el exterior.

Diversos módulos para el control de periféricos (temporizadores, Puertas Serie y Paralelo, CAD: Conversores Analógico/Digital, CDA: Conversores Digital/Analógico, etc.).

Generador de impulsos de reloj que sincronizan el funcionamiento de todo el sistema.

1.2.1.- Diferencia entre microprocesador y microcontrolador.-

El microprocesador es un circuito integrado que contiene la Unidad Central de Proceso (UCP), también llamada procesador, de un computador. La UCP está formada por la Unidad de Control, que interpreta las instrucciones, y el Camino de Datos, que las ejecuta.

Las patitas de un microprocesador sacan al exterior las líneas de sus buses de direcciones, datos y control, para permitir conectarle con la Memoria y los Módulos de E/S y configurar un computador implementado por varios circuitos integrados. Se dice que un microprocesador es un sistema abierto porque su configuración es variable de acuerdo con la aplicación a la que se destine.

(Figura 1.7)

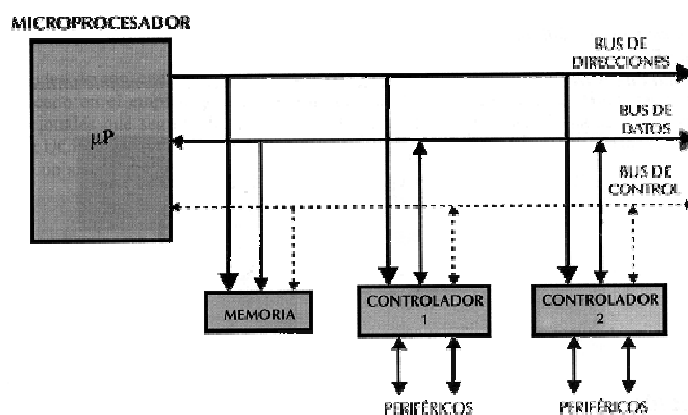


Figura 1.7.- Estructura de un sistema abierto basado en un microprocesador.

Es muy habitual confundir los términos de microcontrolador y microprocesador, cayendo así en un error de cierta magnitud. Un microcontrolador es, un sistema completo, con unas prestaciones limitadas que no pueden modificarse y que puede llevar a cabo las tareas para las que ha sido programado de forma

autónoma. Un microprocesador, en cambio, es simplemente un componente que conforma el microcontrolador, que lleva a cabo ciertas tareas y que, en conjunto con otros componentes, forman un microcontrolador.

Si sólo se dispusiese de un modelo de microcontrolador, éste debería tener muy potenciados todos sus recursos para poderse adaptar a las exigencias de las diferentes aplicaciones. Esta potenciación supondría en muchos casos un despilfarro. En la práctica cada fabricante de microcontroladores oferta un elevado número de modelos diferentes, desde los más sencillos hasta los más poderosos. Es posible seleccionar la capacidad de las memorias, el número de líneas de E/S, la cantidad y potencia de los elementos auxiliares, la velocidad de funcionamiento, etc. Por todo ello, un aspecto muy destacado del diseño es la selección del microcontrolador a utilizar.

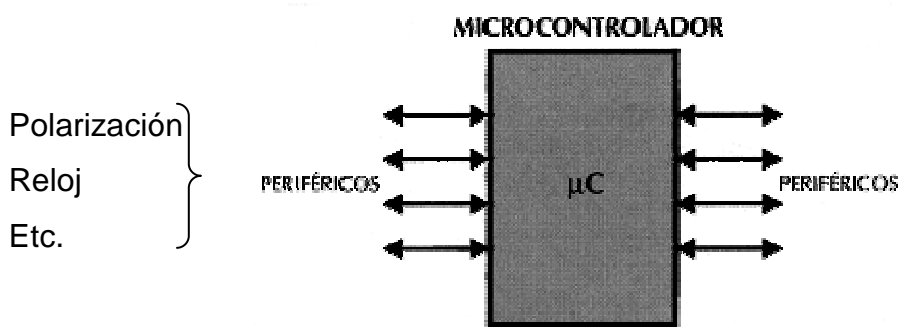


Figura 1.8.- El microcontrolador es un sistema cerrado.

1.2.2.- Aplicaciones de los microcontroladores.- Cada vez existen más productos que incorporan un microcontrolador con el fin de aumentar sustancialmente sus prestaciones, reducir su tamaño y costo, mejorar su fiabilidad y disminuir el consumo.

Algunos fabricantes de microcontroladores superan el millón de unidades de un modelo determinado producidas en una semana. Este dato puede dar una idea de la masiva utilización de estos componentes.

Los microcontroladores están siendo empleados en multitud de sistemas presentes en nuestra vida diaria, como pueden ser juguetes, horno microondas, frigoríficos, televisores, computadoras, impresoras, módems, el sistema de

arranque de nuestro coche, etc. Y otras aplicaciones con las que seguramente no estaremos tan familiarizados como instrumentación electrónica, control de sistemas en una nave espacial, etc. Una aplicación típica podría emplear varios microcontroladores para controlar pequeñas partes del sistema. Estos pequeños controladores podrían comunicarse entre ellos y con un procesador central, probablemente más potente, para compartir la información y coordinar sus acciones, como, de hecho, ocurre ya habitualmente en cualquier PC.

1.2.3.- El mercado de los microcontroladores.- Aunque en el mercado de la microinformática la mayor atención la acaparan los desarrollos de los microprocesadores, lo cierto es que se venden cientos de microcontroladores por cada uno de aquéllos.

Existe una gran diversidad de microcontroladores. Quizá la clasificación más importante sea entre microcontroladores de 4, 8, 16 ó 32 bits. Aunque las prestaciones de los microcontroladores de 16 y 32 bits son superiores a los de 4 y 8 bits, la realidad es que los microcontroladores de 8 bits dominan el mercado y los de 4 bits se resisten a desaparecer. La razón de esta tendencia es que los microcontroladores de 4 y 8 bits son apropiados para la gran mayoría de las aplicaciones, lo que hace absurdo emplear micros más potentes y consecuentemente más caros. Uno de los sectores que más tira del mercado del microcontrolador es el mercado automovilístico. De hecho, algunas de las familias de microcontroladores actuales se desarrollaron pensando en este sector, siendo modificadas posteriormente para adaptarse a sistemas más genéricos. El mercado del automóvil es además uno de los más exigentes: los componentes electrónicos deben operar bajo condiciones extremas de vibraciones, choques, ruido, etc. y seguir siendo fiables. El fallo de cualquier componente en un automóvil puede ser el origen de un accidente.

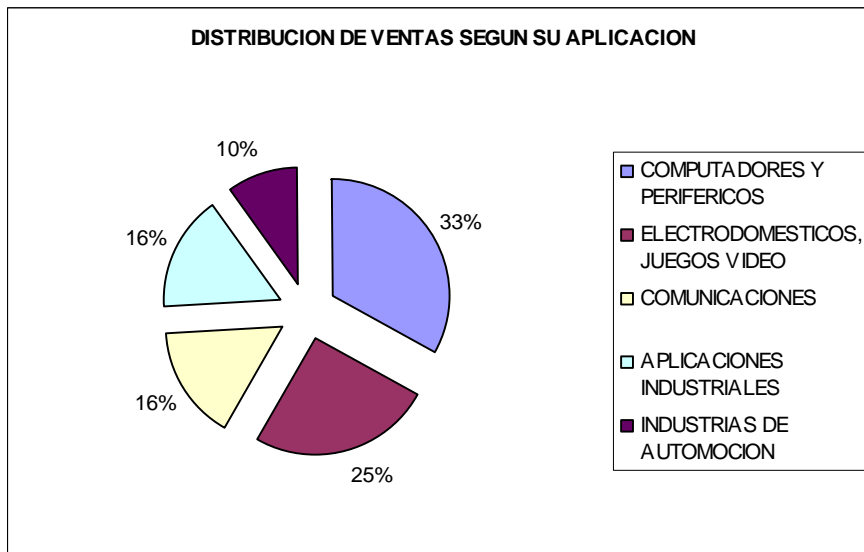


Figura 1.9.- Distribución de ventas según su aplicación.

En cuanto a las técnicas de fabricación, cabe decir que prácticamente la totalidad de los microcontroladores actuales se fabrican con tecnología CMOS 4 (Complementary Metal Oxide Semiconductor). Esta tecnología supera a las técnicas anteriores por su bajo consumo de potencia y alta inmunidad al ruido.

También los modernos microcontroladores de 32 bits van afianzando sus posiciones en el mercado, siendo las áreas de más interés el procesamiento de imágenes, las comunicaciones, las aplicaciones militares, los procesos industriales y el control de los dispositivos de almacenamiento masivo de datos.

1.2.4.- ¿Qué microcontrolador emplear?

A la hora de escoger el microcontrolador a emplear en un diseño concreto hay que tener en cuenta multitud de factores, como la documentación y herramientas de desarrollo disponibles y su precio, la cantidad de fabricantes que lo producen y por supuesto las características del microcontrolador (tipo de memoria de programa, número de temporizadores, interrupciones, etc.)

Como es lógico, los fabricantes de microcontroladores compiten duramente para vender sus productos. Y no les va demasiado mal ya que sin hacer

demasiado ruido venden 10 veces más microcontroladores que microprocesadores.

Para que nos hagamos una idea, para el fabricante que usa el microcontrolador en su producto una diferencia de precio en el microcontrolador de algunos centavos es importante (el consumidor deberá pagar además el costo del empaquetado, el de los otros componentes, el diseño del hardware y el desarrollo del software). Si el fabricante desea reducir costos debe tener en cuenta las herramientas de apoyo con que va a contar: emuladores, simuladores, ensambladores, compiladores, etc. Es habitual que muchos de ellos siempre se decidan por microcontroladores pertenecientes a una única familia.

1.2.5.- Recursos comunes a todos los microcontroladores.- Al estar todos los microcontroladores integrados en un chip, su estructura fundamental y sus características básicas son muy parecidas. Todos deben disponer de los bloques esenciales Procesador, memoria de datos y de instrucciones, líneas de E/S, oscilador de reloj y módulos controladores de periféricos. Sin embargo, cada fabricante intenta enfatizar los recursos más idóneos para las aplicaciones a las que se destinan preferentemente.

En este apartado se hace un recorrido de todos los recursos que se hallan en todos los microcontroladores describiendo las diversas alternativas y opciones que pueden encontrarse según el modelo seleccionado.

1.2.5.1.- Arquitectura básica

Aunque inicialmente todos los microcontroladores adoptaron la arquitectura clásica de Von Neumann, en el momento presente se impone la arquitectura Harvard. La arquitectura de Von Neumann se caracteriza por disponer de una sola memoria principal donde se almacenan datos e instrucciones de forma indistinta. A dicha memoria se accede a través de un sistema de buses único (direcciones, datos y control).

La arquitectura Harvard dispone de dos memorias independientes una, que contiene sólo instrucciones y otra, sólo datos. Ambas disponen de sus respectivos sistemas de buses de acceso y es posible realizar operaciones de acceso (lectura o escritura) simultáneamente en ambas memorias.

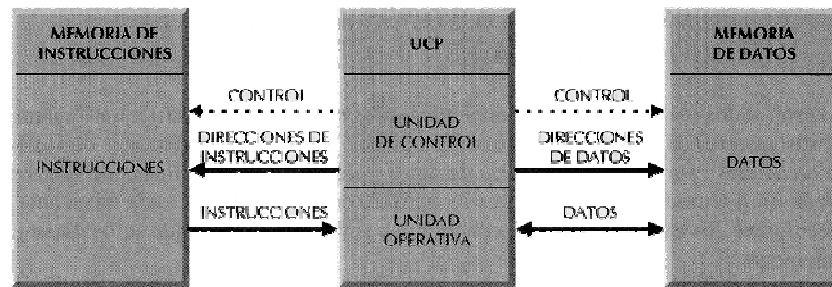


Figura 1.10.- La arquitectura Harvard.

1.2.5.2.- El procesador o UCP.-

Es el elemento más importante del microcontrolador y determina sus principales características, tanto a nivel hardware como software.

Se encarga de direccionar la memoria de instrucciones, recibir el código de la instrucción en curso, su decodificación y la ejecución de la operación que implica la instrucción, así como la búsqueda de los operandos y el almacenamiento del resultado.

Existen tres orientaciones en cuanto a la arquitectura y funcionalidad de los procesadores actuales.

CISC: Un gran número de procesadores usados en los microcontroladores están basados en la filosofía CISC (Computadores de Juego de Instrucciones Complejo). Disponen de más de 80 instrucciones máquina en su repertorio, algunas de las cuales son muy sofisticadas y potentes, requiriendo muchos ciclos para su ejecución.

Una ventaja de los procesadores CISC es que ofrecen al programador instrucciones complejas que actúan como macros.

RISC: Tanto la industria de los computadores comerciales como la de los microcontroladores están decidiéndose hacia la filosofía RISC (Computadores de Juego de Instrucciones Reducido). En estos procesadores el repertorio de instrucciones máquina es muy reducido y las instrucciones son simples y, generalmente, se ejecutan en un ciclo.

La sencillez y rapidez de las instrucciones permiten optimizar el hardware y el software del procesador.

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

1.2.5.3.- Memoria.-

En los microcontroladores la memoria de instrucciones y datos está integrada en el propio chip. Una parte debe ser no volátil, tipo ROM, y se destina a contener el programa de instrucciones que gobierna la aplicación. Otra parte de memoria será tipo RAM, volátil, y se destina a guardar las variables y los datos.

Hay dos peculiaridades que diferencian a los microcontroladores de los computadores personales:

No existen sistemas de almacenamiento masivo como disco duro o disquetes. Como el microcontrolador sólo se destina a una tarea en la memoria ROM, sólo hay que almacenar un único programa de trabajo.

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

Los usuarios de computadores personales están habituados a manejar Megabytes de memoria, pero, los diseñadores con microcontroladores trabajan con capacidades de ROM comprendidas entre 512 bytes y 8 k bytes y de RAM comprendidas entre 20 y 512 bytes.

Según el tipo de memoria ROM que dispongan los microcontroladores, la aplicación y utilización de los mismos es diferente. Se describen las cinco versiones de memoria no volátil que se pueden encontrar en los microcontroladores del mercado.

1º. ROM con máscara.- Es una memoria no volátil de sólo lectura cuyo contenido se graba durante la fabricación del chip. El elevado coste del diseño de la máscara sólo hace aconsejable el empleo de los microcontroladores con este tipo de memoria cuando se precisan cantidades superiores a varios miles de unidades.

2ª. OTP.- El microcontrolador contiene una memoria no volátil de sólo lectura "programable una sola vez" por el usuario. OTP (One Time Programmable). Es el usuario quien puede escribir el programa en el chip mediante un sencillo grabador controlado por un programa desde un PC. La versión OTP es recomendable cuando es muy corto el ciclo de diseño del producto, o bien, en la construcción de prototipos y series muy pequeñas.

Tanto en este tipo de memoria como en la EPROM, se suele usar la encriptación mediante fusibles para proteger el código contenido.

3ª EPROM.- Los microcontroladores que disponen de memoria EPROM (Erasable Programmable Read Only Memory) pueden borrarse y grabarse muchas veces. La grabación se realiza, como en el caso de los OTP, con un grabador gobernado desde un PC. Si, posteriormente, se desea borrar el contenido, disponen de una ventana de cristal en su superficie por la que se somete a la EPROM a rayos ultravioleta durante varios minutos. Las cápsulas son de material cerámico y son más caros que los microcontroladores con memoria OTP que están hechos con material plástico.

4ª EEPROM.- Se trata de memorias de sólo lectura, programables y borrables eléctricamente EEPROM (Electrical Erasable Programmable Read Only Memory). Tanto la programación como el borrado, se realizan eléctricamente desde el propio grabador y bajo el control programado de un PC. Es muy cómoda y rápida la operación de grabado y la de borrado. No disponen de ventana de cristal en la superficie.

Los microcontroladores dotados de memoria EEPROM una vez instalados en el circuito, pueden grabarse y borrarse cuantas veces se quiera sin ser retirados de dicho circuito. Para ello se usan "grabadores en circuito" que confieren una gran flexibilidad y rapidez a la hora de realizar modificaciones en el programa de trabajo.

El número de veces que puede grabarse y borrarse una memoria EEPROM es finito, por lo que no es recomendable una reprogramación continúa. Son muy idóneos para la enseñanza y la Ingeniería de diseño.

Se va extendiendo en los fabricantes la tendencia de incluir una pequeña zona de memoria EEPROM en los circuitos programables para guardar y modificar cómodamente una serie de parámetros que adecuan el dispositivo a las condiciones del entorno.

Este tipo de memoria es relativamente lenta.

5ª FLASH.- Se trata de una memoria no volátil, de bajo consumo, que se puede escribir y borrar. Funciona como una ROM y una RAM pero consume menos y es más pequeña.

A diferencia de la ROM, la memoria FLASH es programable en el circuito. Es más rápida y de mayor densidad que la EEPROM.

La alternativa FLASH está recomendada frente a la EEPROM cuando se precisa gran cantidad de memoria de programa no volátil. Es más veloz y tolera más ciclos de escritura/borrado.

Las memorias EEPROM y FLASH son muy útiles al permitir que los microcontroladores que las incorporan puedan ser reprogramados "en circuito", es decir, sin tener que sacar el circuito integrado de la tarjeta. Así, un dispositivo con este tipo de memoria incorporado al control del motor de un automóvil permite que pueda modificarse el programa durante la rutina de mantenimiento periódico, compensando los desgastes y otros factores tales como la compresión, la instalación de nuevas piezas, etc. La reprogramación del microcontrolador puede convertirse en una labor rutinaria dentro de la puesta a punto.

1.2.5.4.- Puertos de Entrada y Salida.-

La principal utilidad de las patitas que posee la cápsula que contiene un microcontrolador es soportar las líneas de E/S que comunican al computador interno con los periféricos exteriores.

Según los controladores de periféricos que posea cada modelo de microcontrolador, las líneas de E/S se destinan a proporcionar el soporte a las señales de entrada, salida y control.

1.2.5.5.- Reloj principal.-

Todos los microcontroladores disponen de un circuito oscilador que genera una onda cuadrada de alta frecuencia, que configura los impulsos de reloj usados en la sincronización de todas las operaciones del sistema.

Generalmente, el circuito de reloj está incorporado en el microcontrolador y sólo se necesitan unos pocos componentes exteriores para seleccionar y estabilizar la frecuencia de trabajo. Dichos componentes suelen consistir en un cristal de cuarzo junto a elementos pasivos o bien un resonador cerámico o una red R-C.

Aumentar la frecuencia de reloj supone disminuir el tiempo en que se ejecutan las instrucciones pero lleva aparejado un incremento del consumo de energía.

1.2.6.- Recursos Especiales.- Cada fabricante oferta numerosas versiones de una arquitectura básica de microcontrolador. En algunas amplía las capacidades de las memorias, en otras incorpora nuevos recursos, en otras reduce las prestaciones al mínimo para aplicaciones muy simples, etc. La labor del diseñador es encontrar el modelo mínimo que satisfaga todos los requerimientos de su aplicación. De esta forma, minimizará el costo, el hardware y el software. Los principales recursos específicos que incorporan los microcontroladores son:

- Temporizadores o "Timers".
- Perro guardián o "Watchdog".
- Protección ante fallo de alimentación o "Brownout".
- Estado de reposo o de bajo consumo.
- Conversor A/D.
- Conversor D/A.
- Comparador analógico.
- Modulador de anchura de impulsos o PWM.
- Puertos de E/S digitales.
- Puertos de comunicación.

1.2.6.1.- Temporizadores o "Timers".-

Se emplean para controlar periodos de tiempo (temporizadores) y para llevar la cuenta de acontecimientos que suceden en el exterior (contadores).

Para la medida de tiempos se carga un registro con el valor adecuado y a continuación dicho valor se va incrementando o decrementando al ritmo de los impulsos de reloj o algún múltiplo hasta que se desborde y llegue a 0, momento en el que se produce un aviso.

Cuando se desean contar acontecimientos que se materializan por cambios de nivel o flancos en alguna de las patitas del microcontrolador, el mencionado registro se va incrementando o decrementando al ritmo de dichos impulsos.

1.2.6.2.- Perro guardián o "Watchdog".-

Cuando el computador personal se bloquea por un fallo del software u otra causa, se pulsa el botón del reset y se reinicializa el sistema. Pero un microcontrolador funciona sin el control de un supervisor y de forma continuada las 24 horas del día. El Perro guardián consiste en un temporizador que, cuando se desborda y pasa por 0, provoca un reset automáticamente en el sistema.

Se debe diseñar el programa de trabajo que controla la tarea de forma que refresque o inicialice al Perro guardián antes de que provoque el reset. Si falla el programa o se bloquea, no se refrescará al Perro guardián y, al completar su temporización, "ladrará y ladrará" hasta provocar el reset.

1.2.6.3.- Protección ante fallo de alimentación o "Brownout".-

Se trata de un circuito que resetea al microcontrolador cuando el voltaje de alimentación (VDD) es inferior a un voltaje mínimo ("brownout"). Mientras el voltaje de alimentación sea inferior al de brownout el dispositivo se mantiene reseteado, comenzando a funcionar normalmente cuando sobrepasa dicho valor.

1.2.6.4.- Estado de reposo ó de bajo consumo.-

Son abundantes las situaciones reales de trabajo en que el microcontrolador debe esperar, sin hacer nada, a que se produzca algún acontecimiento externo que le ponga de nuevo en funcionamiento. Para ahorrar energía, (factor clave en los aparatos portátiles), los microcontroladores disponen de una instrucción especial (SLEEP en los PIC), que les pasa al estado de reposo o de bajo consumo, en el cual los requerimientos de potencia son mínimos. En dicho estado se detiene el reloj principal y se "congelan" sus circuitos asociados, quedando sumido en un profundo "sueño" el microcontrolador. Al activarse una interrupción ocasionada por el acontecimiento esperado, el microcontrolador se despierta y reanuda su trabajo.

1.2.6.5.- Conversor A/D (CAD).-

Los microcontroladores que incorporan un Conversor A/D (Analógico/Digital) pueden procesar señales analógicas, tan abundantes en las aplicaciones.

Suelen disponer de un multiplexor que permite aplicar a la entrada del CAD diversas señales analógicas desde las patitas del circuito integrado.

1.2.6.6.- Conversor D/A (CDA).-

Transforma los datos digitales obtenidos del procesamiento del computador en su correspondiente señal analógica que saca al exterior por una de las patitas de la cápsula. Existen muchos efectores que trabajan con señales analógicas.

1.2.6.7.- Comparador analógico.-

Algunos modelos de microcontroladores disponen internamente de un Amplificador Operacional que actúa como comparador entre una señal fija de referencia y otra variable que se aplica por una de las patitas de la cápsula. La salida del comparador proporciona un nivel lógico 1 ó 0 según una señal sea mayor o menor que la otra.

También hay modelos de microcontroladores con un módulo de tensión de referencia que proporciona diversas tensiones de referencia que se pueden aplicar en los comparadores.

1.2.6.8.- Modulador de anchura de impulsos o PWM.-

Son circuitos que proporcionan en su salida impulsos de anchura variable, que se ofrecen al exterior a través de las patitas del encapsulado.

1.2.6.9.- Puertos de E/S digitales.-

Todos los microcontroladores destinan algunas de sus patitas a soportar líneas de E/S digitales. Por lo general, estas líneas se agrupan de ocho en ocho formando Puertos.

Las líneas digitales de los Puertos pueden configurarse como Entrada o como Salida cargando un 1 ó un 0 en el bit correspondiente de un registro destinado a su configuración.

1.2.6.10.- Puertos de comunicación.-

Con objeto de dotar al microcontrolador de la posibilidad de comunicarse con otros dispositivos externos, otros buses de microprocesadores, buses de sistemas, buses de redes y poder adaptarlos con otros elementos bajo otras normas y protocolos. Algunos modelos disponen de recursos que permiten directamente esta tarea, entre los que destacan:

UART, adaptador de comunicación serie asíncrona.

USART, adaptador de comunicación serie síncrona y asíncrona.

Puerta paralela esclava para poder conectarse con los buses de otros microprocesadores.

USB (Universal Serial Bus), que es un moderno bus serie para los PC.

Bus I²C, que es un interfaz serie de dos hilos desarrollado por Philips.

CAN (Controller Area Network), para permitir la adaptación con redes de conexionado multiplexado desarrollado conjuntamente por Bosch e Intel para el cableado de dispositivos en automóviles. En EE.UU. se usa el J1850.

1.2.7.- Herramientas para el desarrollo de aplicaciones.- Uno de los factores que más importancia tiene a la hora de seleccionar un microcontrolador entre todos los demás es el soporte tanto software como hardware de que dispone.

Un buen conjunto de herramientas de desarrollo puede ser decisivo en la elección, ya que pueden suponer una ayuda inestimable en el desarrollo del proyecto.

Las principales herramientas de ayuda al desarrollo de sistemas basados en microcontroladores son:

1.2.7.1.- Desarrollo del software:

Ensamblador.- La programación en lenguaje ensamblador puede resultar un tanto ardua para el principiante, pero permite desarrollar programas muy eficientes, ya que otorga al programador el dominio absoluto del sistema. Los fabricantes suelen proporcionar el programa ensamblador de forma gratuita y en cualquier caso siempre se puede encontrar una versión gratuita para los microcontroladores más populares.

Compilador.- La programación en un lenguaje de alto nivel (como el C ó el Basic) permite disminuir el tiempo de desarrollo de un producto. No obstante, si no se programa con cuidado, el código resultante puede ser mucho más ineficiente que el programado en ensamblador. Las versiones más potentes suelen ser muy caras, aunque para los microcontroladores más populares pueden encontrarse versiones demo limitadas e incluso compiladores gratuitos.

Depuración: debido a que los microcontroladores van a controlar dispositivos físicos, los desarrolladores necesitan herramientas que les permitan comprobar el buen funcionamiento del microcontrolador cuando es conectado al resto de circuitos.

Simulador.- Son capaces de ejecutar en un PC programas realizados para el microcontrolador. Los simuladores permiten tener un control absoluto sobre la ejecución de un programa, siendo ideales para la depuración de los mismos. Su gran inconveniente es que es difícil simular la entrada y salida de datos del microcontrolador. Tampoco cuentan con los posibles ruidos en las entradas, pero, al menos, permiten el paso físico de la implementación de un modo más

seguro y menos costoso, puesto que ahorraremos en grabaciones de chips para la prueba in-situ.

Placas de evaluación.- Se trata de pequeños sistemas con un microcontrolador ya montado y que suelen conectarse a un PC desde el que se cargan los programas que se ejecutan en el microcontrolador. Las placas suelen incluir visualizadores LCD, teclados, LEDs, fácil acceso a los pines de E/S, etc. El sistema operativo de la placa recibe el nombre de programa monitor. El programa monitor de algunas placas de evaluación, aparte de permitir cargar programas y datos en la memoria del microcontrolador, puede permitir en cualquier momento realizar ejecución paso a paso, monitorizar el estado del microcontrolador o modificar los valores almacenados los registros o en la memoria.

Emuladores en circuito.- Se trata de un instrumento que se coloca entre el PC anfitrión y el zócalo de la tarjeta de circuito impreso donde se alojará el microcontrolador definitivo. El programa es ejecutado desde el PC, pero para la tarjeta de aplicación es como si lo hiciese el mismo microcontrolador que luego irá en el zócalo. Presenta en pantalla toda la información tal y como luego sucederá cuando se coloque la cápsula.

1.3.- Opto Acoplador

Un opto acoplador, también llamado *opto aislador* o aislador acoplado ópticamente, es un dispositivo de emisión y recepción de luz que funciona como un interruptor excitado mediante la luz. La mencionada luz es emitida por un diodo emisor de luz que satura un componente opto electrónico, normalmente en forma de fototransistor. De este modo se combinan en un solo dispositivo semiconductor, un foto emisor y un fotorreceptor cuya conexión entre ambos es óptica. Estos elementos se encuentran dentro de un encapsulado que por lo general es del tipo DIP. Se suelen utilizar como medio de protección para dispositivos muy sensibles.

La figura 1.11 muestra un opto acoplador 4N35 formado por un LED y un fototransistor. La tensión de la fuente de la izquierda y la resistencia en serie establecen una corriente en el LED emisor cuando se cierra el interruptor S1. Si dicha corriente proporciona un nivel de luz adecuado, al incidir sobre el fototransistor lo saturará, generando una corriente en R2. De este modo la tensión de salida será igual a cero con S1 cerrado y a V2 con S1 abierto.

Si la tensión de entrada varía, la cantidad de luz también lo hará, lo que significa que la tensión de salida cambia de acuerdo con la tensión de entrada. De este modo el dispositivo puede acoplar una señal de entrada con el circuito de salida.

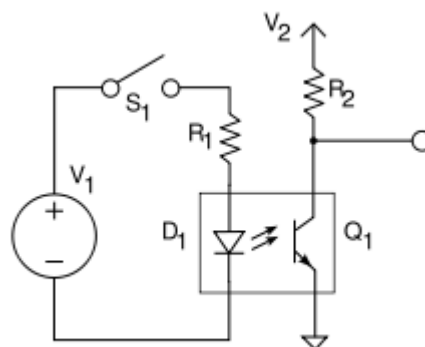


Figura 1.11.- El opto acoplador combina un LED y un fotodiodo.

La ventaja fundamental de un opto acoplador es el aislamiento eléctrico entre los circuitos de entrada y salida. Mediante el opto acoplador, el único contacto

que entre ambos circuitos es un haz de luz. Esto se traduce en una resistencia de aislamiento entre los dos circuitos del orden de miles de MΩ. Estos aislamientos son útiles en aplicaciones de alta tensión en las que los potenciales de los dos circuitos pueden diferir en varios miles de voltios.

1.3.1.- Tipos.- En general, los diferentes tipos de opto acopladores se distinguen por su diferente etapa de salida. Entre los principales cabe destacar el fototransistor, ya mencionado, el *fototriac* y el fototriac de paso por cero. En este último, su etapa de salida es un *triac* de cruce por cero, que posee un circuito interno que conmuta al triac sólo en los cruce por cero de la fuente.

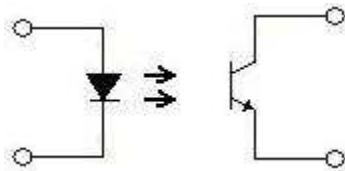


Figura 1.12.- Etapa de salida a fototransistor.

1.3.2.- Encapsulados.- El encapsulado varia en función del tipo de opto acoplador y de su aplicación, así como del número de unidades que se encuentren en su interior. En el caso de opto acopladores sencillos la cápsula, de tipo DIL, suele tener 6 pines, siendo estos los más utilizados. Los dobles, también de tipo DIL tienen 8 pines; algunos pueden tener hasta cuatro unidades en cápsulas DIL de 16 pines. Normalmente, los pines del elemento emisor están a un lado de la cápsula y los del sensor en el lado opuesto.

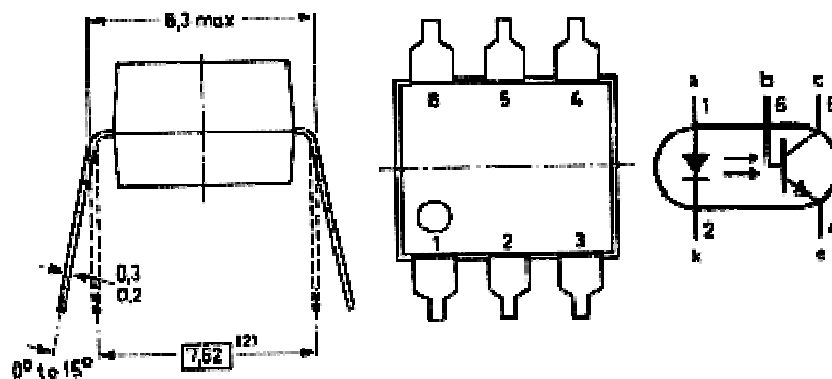


Figura 1.13.- El opto acoplador encapsulado.



Figura 1.14.- Encapsulado Tipo DIP / Encapsulado Tipo DIL

Existen unos encapsulados diferentes en los que, físicamente se puede interrumpir el haz luminoso (usados para control de posición, nº de revoluciones, cerraduras...).



Figura 1.15.- Encapsulados de Ranura

De esta forma el encapsulado presenta una ranura entre el emisor y el receptor. Se les denomina de cápsula ranurada o fotocélulas de herradura.

1.4.- Display de Cristal Líquido (LCD)

Los LCD's difieren de otros tipos de displays en que no generan luz sino que trabajan con la reflexión de la luz. El principio de funcionamiento es sencillo. Estos cristales líquidos están formados por unas moléculas alargadas con forma de puro, que se llaman moléculas nemáticas y se alinean con una estructura simétrica. En este estado el material es transparente. Un campo eléctrico provoca que las moléculas se desalineen de manera que se vuelven opacas a la luz. De esta manera, aplicando o no aplicando un campo eléctrico (es decir, polarizando o no polarizando), podemos jugar con oscuridad o transparencia respectivamente. Si aplicamos el campo localmente en geometrías iguales al display de 7 segmentos, conseguiremos un display análogo al de los LED's pero con cristal líquido.

En la construcción de un LCD se depositan electrodos transparentes en la cara interior de los cristales, tal y como aparece en la Figura 14). Estos electrodos tienen la geometría deseada, por ejemplo, el display de 7 segmentos. El espesor del cristal líquido es muy pequeño, del orden de 0.01mm.

Si no se polarizan los terminales, al incidir la luz sobre el cristal frontal, pasa a través del cristal líquido y es reflejada por el espejo incidiendo en el ojo que está mirando. El resultado: todo se ve de color claro.

Si polarizamos un electrodo, por ejemplo, el electrodo a, el cristal líquido pegado al electrodo se vuelve opaco, negro, oscuro. La luz ya no es reflejada.

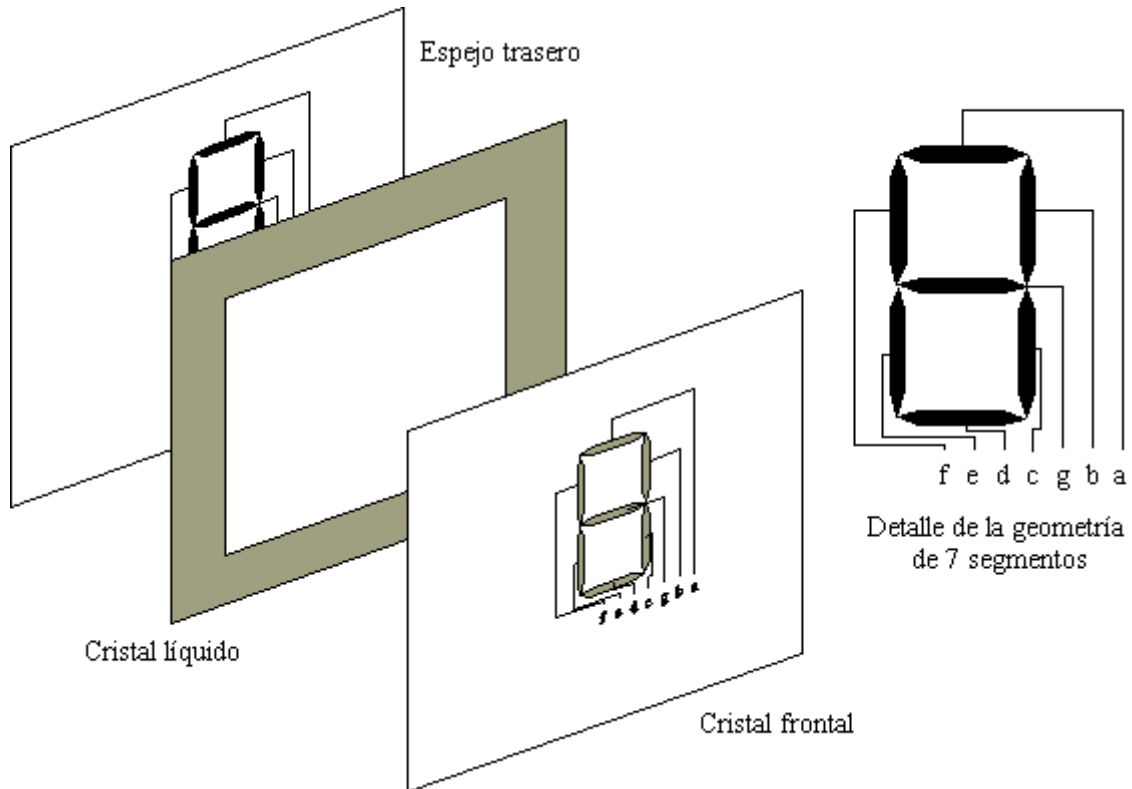


Figura 1.16.- Esquema constructivo de un LCD

Existen diferentes tipos de "displays", según la utilización para la que están diseñados. El más elemental es el "display" de 7 segmentos, diseñado para representar números que van del 0 al 9 como ya lo vimos anteriormente. Con este tipo de indicador luminoso también se pueden representar algunas letras, pero éstas son unas veces mayúsculas (como la A) y otras, minúsculas (como la b o la d). Aun así, es imposible generar algunos caracteres como la Y o la X, entre otras.

Para solucionar estos problemas se han diseñado otros tipos distintos de "displays", llamados alfanuméricos. Existen dos tipos básicos:

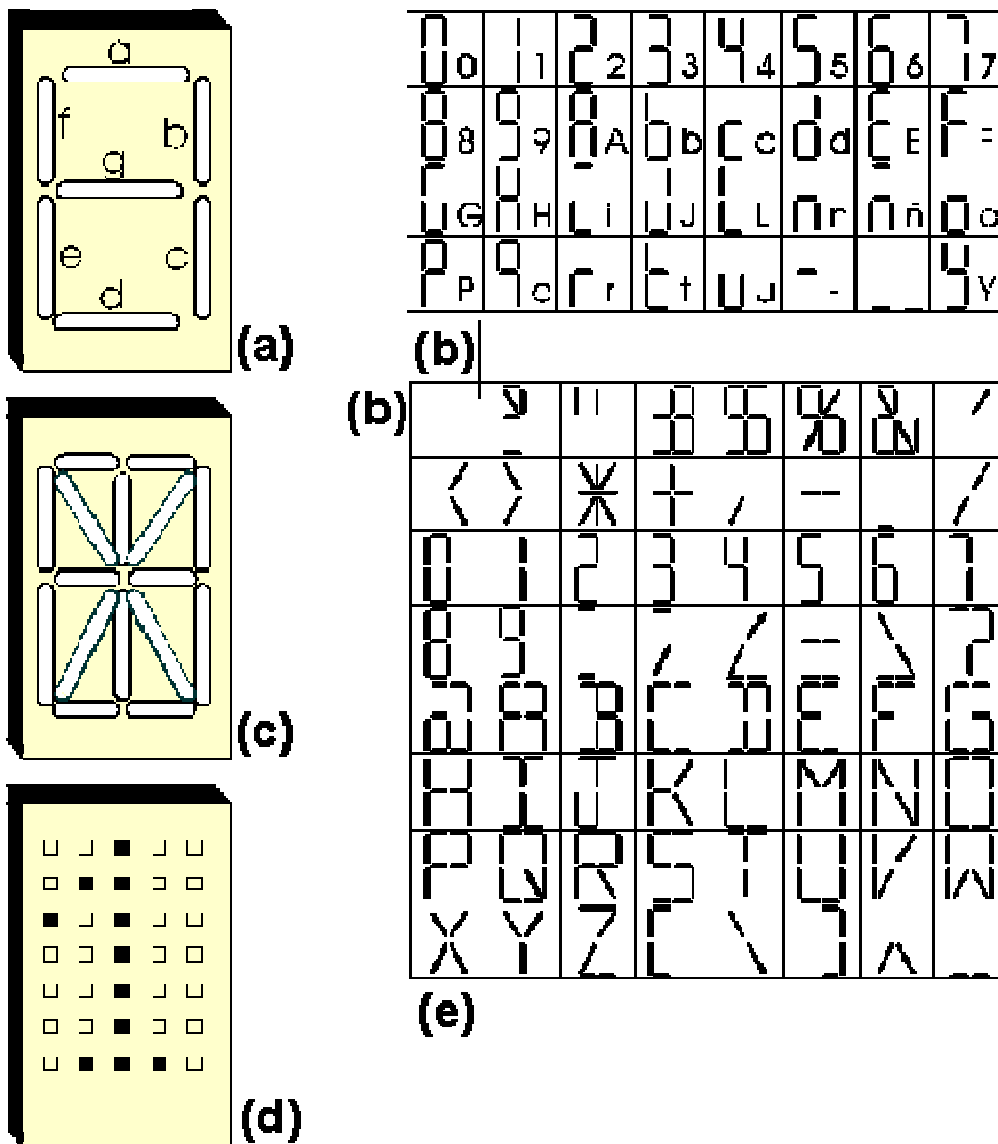


Figura 1.17.- Indicadores luminosos numéricos y alfanuméricos.

- a) Aspecto físico de un "display" numérico de 7 segmentos.
- b) Tabla de caracteres y números que se pueden representar en el "display" de 7 segmentos.
- c) Aspecto físico de un "display" alfanumérico de 16 segmentos.
- d) Idem. de un "display" alfanumérico de 35 puntos.
- e) Tabla de caracteres y números que pueden ser representados en el "display" de 16 segmentos.

El de 16 segmentos, que muestra el esquema de la figura 1.17-c, y el de 35 puntos, en la figura 1.17-d.

El primero es muy parecido al de siete segmentos, pero usa dieciséis segmentos para poder representar un mayor número de caracteres y, en este caso, también signos especiales como los indicados en la figura 1.17-e. El segundo, como se puede observar en la figura 1.17-d, es, en realidad, una matriz de puntos ordenados en siete filas de cinco puntos por cada fila. Como es lógico pensar, la capacidad de realizar símbolos o caracteres es ahora superior a la del "display" de 16 segmentos.

1.4.1.-Display de Cristal Líquido de 2 Líneas de 16 Caracteres.- En esta parte se trata sobre los detalles de funcionamiento de un LCD de 2 líneas de 16 caracteres. Todos los displays de este tipo disponibles en el mercado son compatibles entre sí. A continuación se tratará sobre las características principales del display: caracteres, memoria interna, alimentación, aspecto físico. Esta sección debe ser leída por cualquiera que vaya a manejar un LCD.

También se incluirá los detalles relacionados con el Hardware: asignación de pines, bus de datos, de control, control del contraste, cronogramas. Finalmente se encontrarán explicados los comandos de control del LCD y la secuencia de inicialización.

1.4.1.1.- Características del display.- a continuación se describirá algunos aspectos importantes del display de cristal líquido, que será usado en el presente trabajo.

1.4.1.1.1.- Aspecto físico.- El LCD tiene un aspecto físico como el mostrado en la Figura 1.18. Está constituido por un circuito impreso en el que están integrados los controladores del display y los pines para la conexión del display. Sobre el circuito impreso se encuentra el LCD en sí, rodeado por una estructura metálica que lo protege. En total se pueden visualizar 2 líneas de 16 caracteres cada una, es decir, $2 \times 16 = 32$ caracteres, como se muestra en la Figura 1.19.

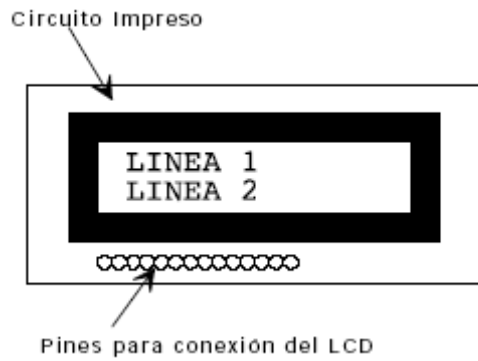


Figura 1.18.- Aspecto Físico del LCD

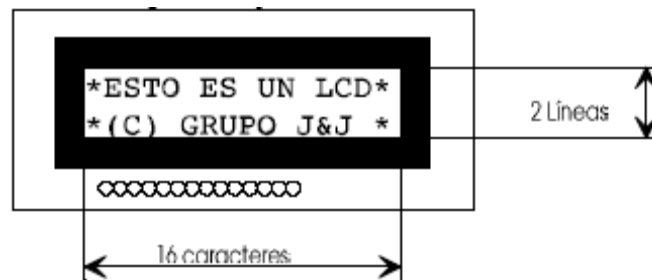


Figura 1.19.- Capacidad de Visualización de caracteres.

A pesar de que el display sólo puede visualizar 16 caracteres por línea, puede almacenar en total 40 por línea. Es el usuario el que especifica qué 16 caracteres son los que se van a visualizar.

1.4.1.1.2.- Alimentación.- La tensión nominal de alimentación es de 5V, con un consumo menor de 5mA.

1.4.1.1.3.- Los caracteres del LCD.- El LCD dispone de una matriz de 5x8 puntos para representar cada carácter. En total se pueden representar 256 caracteres diferentes. 240 caracteres están grabados dentro del LCD y representan las letras mayúsculas, minúsculas, signos de puntuación, números, etc... Existen 8 caracteres que pueden ser definidos por el usuario.

En la Figura 1.20 se muestra gráficamente cómo es la matriz de representación de los caracteres. Se ha dibujado el carácter A y un carácter definido por el usuario.

Figura 1.20.- Matriz de representación de caracteres, representación del carácter A y de un carácter definido por el usuario

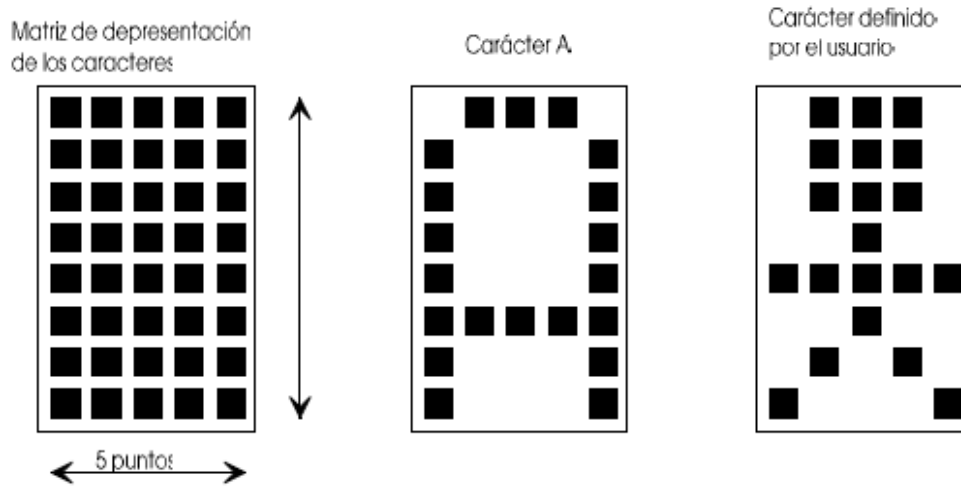


Tabla 1.1: Código asociado a cada carácter imprimible por el display.

Código	Car.	Código	Car.	Código	Car.	Código	Car.	Código	Car.	Código	Car.
\$20	espacio	\$30	0	\$40		\$50	P	\$60	'	\$70	p
\$21	!	\$31	1	\$41	A	\$51	Q	\$61	a	\$71	q
\$22	"	\$32	2	\$42	B	\$52	R	\$62	b	\$72	r
\$23	#	\$33	3	\$43	C	\$53	S	\$63	c	\$73	s
\$24	\$	\$34	4	\$44	D	\$54	T	\$64	d	\$74	t
\$25	%	\$35	5	\$45	E	\$55	U	\$65	e	\$75	u
\$26	&	\$36	6	\$46	F	\$56	V	\$66	f	\$76	v
\$27	'	\$37	7	\$47	G	\$57	W	\$67	g	\$77	w
\$28	(\$38	8	\$48	H	\$58	X	\$68	h	\$78	x
\$29)	\$39	9	\$49	I	\$59	Y	\$69	i	\$79	y
\$2A	*	\$3A	:	\$4A	J	\$5A	Z	\$6A	j	\$7A	z
\$2B	+	\$3B	;	\$4B	K	\$5B	[\$6B	k	\$7B	{
\$2C	,	\$3C	<	\$4C	L	\$5C		\$6C	l	\$7C	
\$2D	-	\$3D	=	\$4D	M	\$5D]	\$6D	m	\$7D	}
\$2E	.	\$3E	>	\$4E	N	\$5E	^	\$6E	n	\$7E	→
\$2F	/	\$3F	?	\$4F	O	\$5F	_	\$6F	o	\$7F	←

En la tabla 1.1 se muestran los caracteres más importantes que es capaz de imprimir el display. Todos los códigos están en hexadecimal. No se han representado los caracteres correspondientes a los códigos desde el \$80 hasta el \$FF, que corresponden a símbolos extraños. Los códigos comprendidos entre el 0 y el 7 están reservados para que el usuario los defina.

1.4.1.1.4.- La memoria del LCD.- El LCD dispone de dos tipos de memorias independientes: la **DD RAM** y la **CG RAM**

1.4.1.1.4.1.- DD RAM (Display Data Ram).- En esta memoria se almacenan los caracteres que están siendo visualizados o que se encuentran en

posiciones no visibles. El display almacena en esta memoria dos líneas de 40 caracteres pero sólo se visualizan 2 líneas de 16 caracteres. Por ello la **DD RAM** tiene un tamaño de $2 \times 40 = 80$ bytes.

Debido a esta peculiar disposición de la **DD RAM** se puede pensar en el display como un *display virtual* constituido por dos líneas de 40 caracteres cada una (Fig. 1.21). La posición situada más a la izquierda de cada línea es la **posición 1** y la situada más a la derecha es la **posición 40**. Para localizar los elementos dentro del *display virtual* se va a utilizar un par de coordenadas (x,y) donde x representa la posición horizontal (comprendida entre 1-40) y y representa la línea (1-2). El display real es una *ventana* en la que se visualizan dos líneas de 16 caracteres. Es lo que el usuario está viendo. En el ejemplo de la Figura 1.21 se encuentra almacenado en la línea 1 del display virtual el mensaje: "ESTO ES UNA PRUEBA DE UN MENSAJE".

Sin embargo, en este ejemplo, el usuario sólo verá en el display el mensaje "PRUEBA DE UN MEN" que tiene exactamente 16 caracteres de longitud. Más adelante se verá cómo es posible 'mover' el display real para que se desplace a lo largo del display virtual. Tal y como se encuentra configurado el display real en la Figura 1.21, la posición (14,1) se corresponde con la letra P, la posición (15,1) con la letra R,, y la posición (29,1) con la letra N. Cuando se inicializa el LCD, el display real se sitúa en el extremo más izquierdo del display virtual, que se corresponde con los valores de x comprendidos entre 1 y 16. En la figura 1.22 se muestra la situación del display real respecto al virtual al inicializar el LCD.

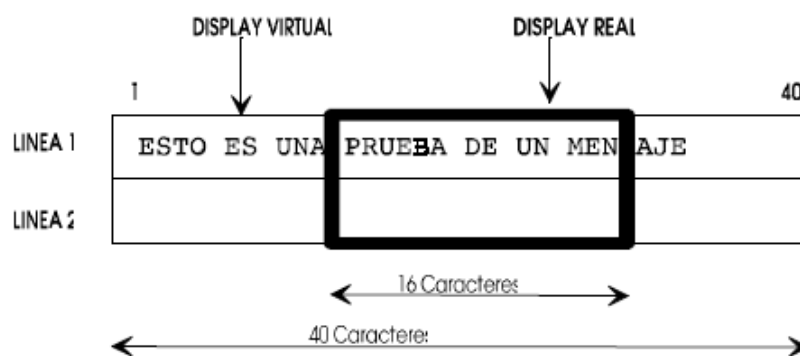


Figura 1.21.- Display Virtual y Display Real.

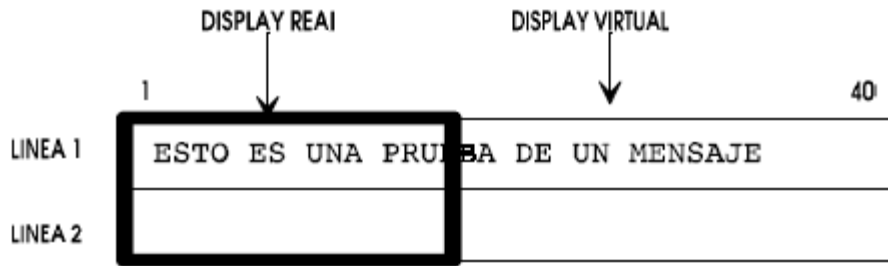


Figura 1.22.- Posición del display real respecto al virtual cuando se inicializa el LCD

En el ejemplo de la Figura 1.22, en la posición (2,1) se encuentra la letra E y en la posición (16,1) la letra U.

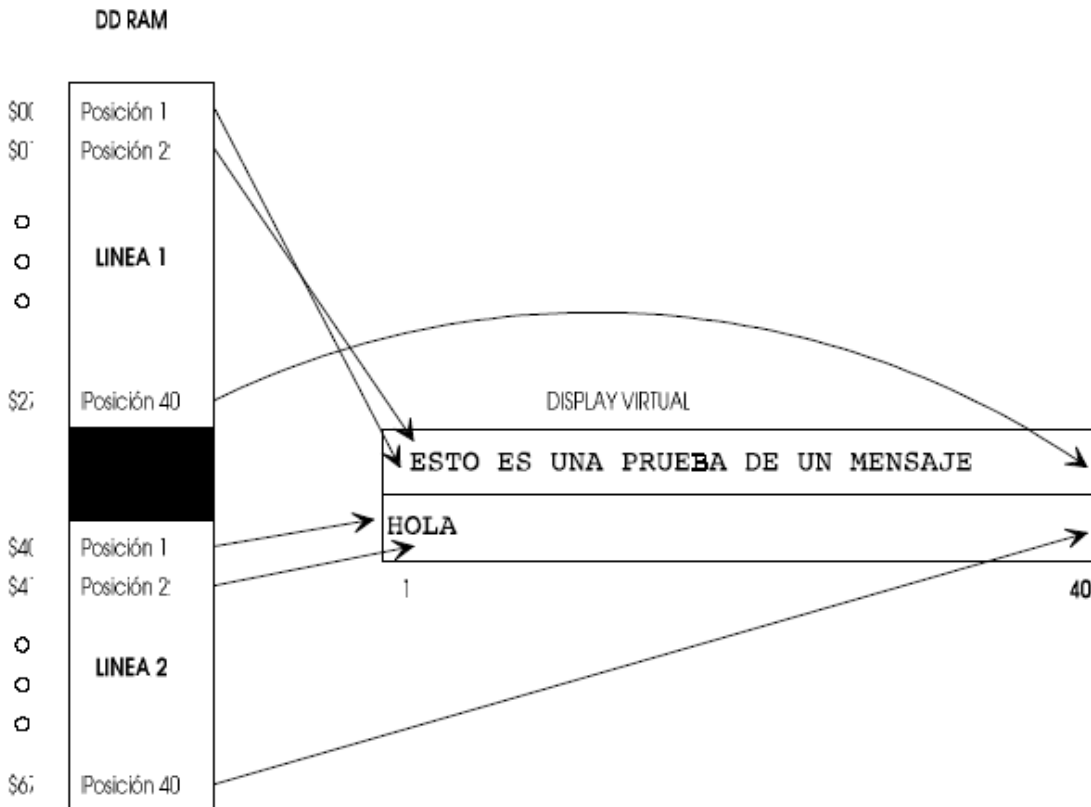


Figura 1.23.- Mapa de memoria de la DD RAM y sus posiciones en el display virtual.

El mapa de memoria de la DD RAM se muestra en la Figura 1.23. Está constituido por dos bloques de 40 bytes. El primer bloque se corresponde con los 40 caracteres de la línea 1 del display virtual. El segundo bloque con la segunda línea. En la figura se han representado las direcciones en hexadecimal.

Así, las direcciones \$00-\$27 están asociadas a las posiciones (1,1)-(40,1) del display virtual y las direcciones \$40-\$67 a las posiciones (1,2)-(40,2). En el ejemplo, en la dirección \$40 de la DD RAM se encuentra almacenado el carácter H, que se corresponde con la posición (1,2) del display virtual. En la dirección \$02 se encuentra el carácter S, posición (3,1) del display virtual. Nótese que los bloques de memoria asociados a la línea 1 y 2 no son contiguos.

Las operaciones de escritura en el display, en realidad son operaciones de escritura en la memoria DD RAM. Según en la posición de esta memoria en la que se escriba el carácter, aparecerá en una posición u otra en el display real. Los caracteres enviados al display pueden ser visibles si se encuentran en posiciones que caen dentro del display real o pueden ser no visibles. En la Figura 1.22, las posiciones (1,1)-(16,1) y (1,2)-(16,2) son visibles. Todos los caracteres enviados a esas posiciones serán visibles. Si se envía un carácter a cualquiera de las otras posiciones no será visible.

1.4.1.1.4.2.- LA CG RAM (Character Generator RAM).- La CG RAM es la memoria que contiene los caracteres definibles por el usuario. Está formada por 64 posiciones, con direcciones \$00-\$3F. Cada posición es de 5 bits.

La memoria está dividida en 8 bloques, correspondiendo cada bloque a un carácter definible por el usuario. Por ello el usuario puede definir como máximo 8 caracteres, cuyos códigos van del 0 al 7.

En la Figura 1.24 se ha representado la CG RAM. Todas las direcciones están en hexadecimal. Como se representó en la figura 3, cada carácter está constituido por una matriz de 5 columnas x 8 filas. Para definir un carácter y asignarle por ejemplo el código 0 habrá que almacenar en las posiciones \$00-\$07 los valores binarios de las 8 filas del carácter del usuario. Un bit con valor 1 representa un punto encendido. Un bit con valor 0 es un punto apagado.

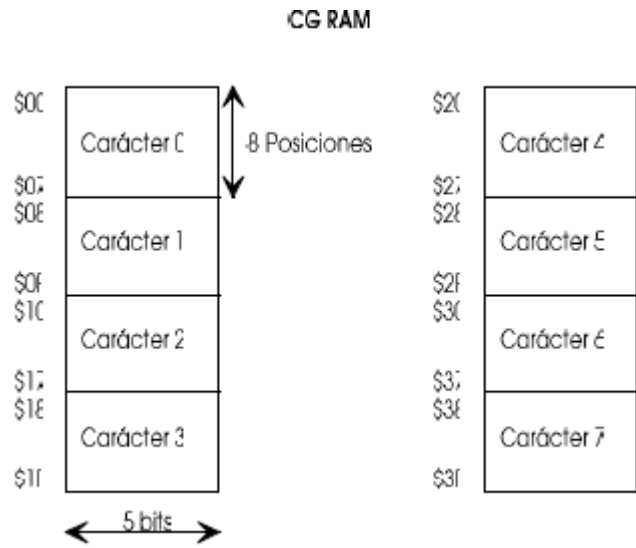


Figura 1.24.- Mapa de memoria de la CG RAM

En la Figura 1.25 se ha dibujado un carácter que se quiere definir. A la Derecha del dibujo se encuentran los valores en binario y en hexadecimal que hay que almacenar en las posiciones de la CG RAM. Si se quiere que este carácter tenga asignado el código 0 habrá que almacenar el valor \$0E en la posición \$00, \$01 y \$02, el valor \$04 en la \$03.... y el valor \$11 en la posición \$07, como se muestra en la Figura 24. Una vez definido el nuevo carácter, cada vez que se envíe su código correspondiente al display se visualizará.

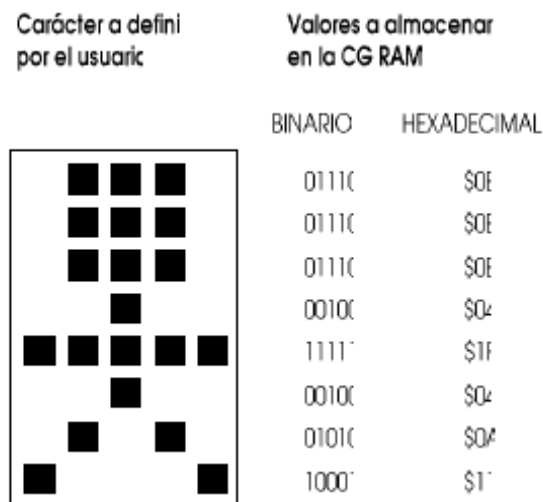


Figura 1.25.- Carácter definido por el usuario y los valores a almacenar en la CG RAM

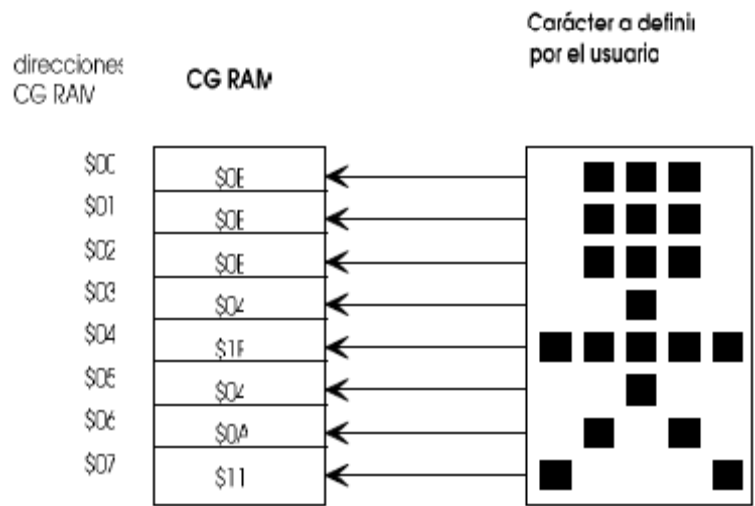


Figura 1.26.- Valores a almacenar en la CG RAM para definir el carácter 0

1.4.1.2.- Interfaz hardware

1.4.1.2.1.- Asignación de pines

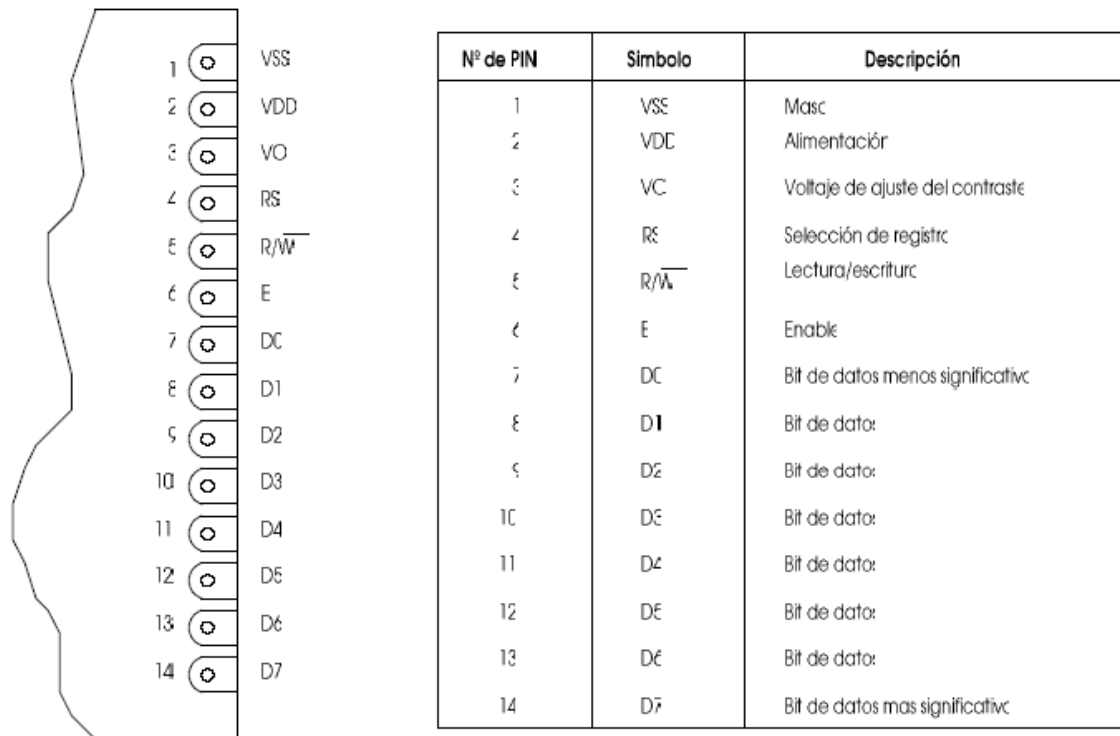


Figura 1.27.- Asignación de pines del LCD.

1.4.1.2.2.- El interfaz del display con el mundo exterior.- En la Figura 1.28 aparecen las señales necesarias para el funcionamiento y control del display.

Los datos se transmiten por un bus de datos de 8 bits de anchura (El display ofrece la posibilidad de trabajar con este bus multiplexado en dos grupos de 4 bits, pero esto se verá más adelante).

Para el control del display son necesarios 3 bits: una señal de enable (E), una para indicar lectura/escritura (R/W) y otra para seleccionar uno de los dos registros internos (RS). Por ello, en el caso peor, el sistema de control del display necesitará utilizar $8+3=11$ bits. En el caso de utilizarse la tarjeta CT6811 será necesario utilizar 2 puertos, por ejemplo el PUERTO C para los datos y el PUERTO B para el control.

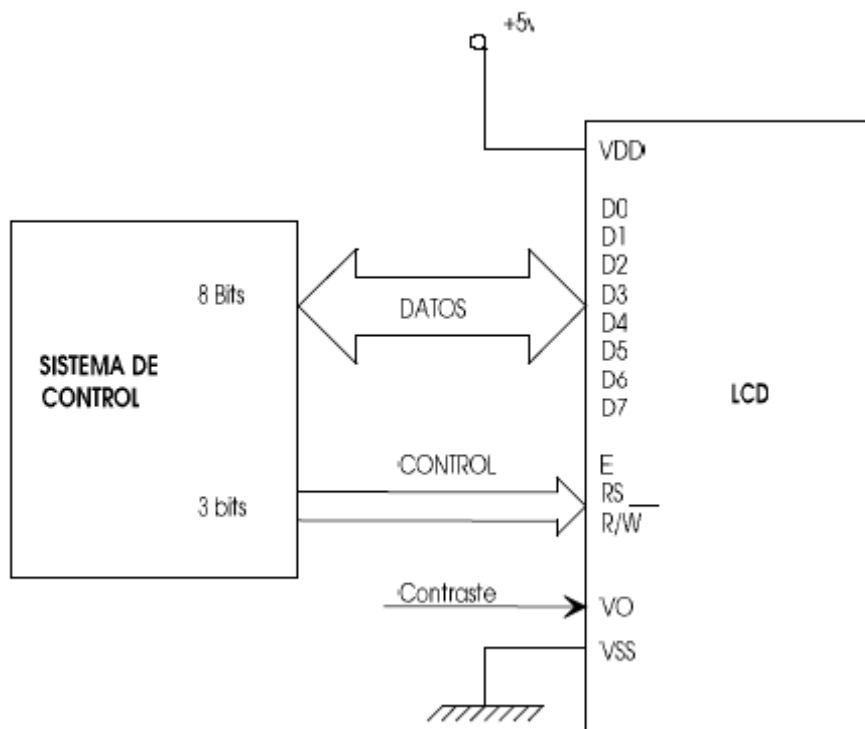


Figura 1.28.- Interfaz del LCD con un sistema de control

1.4.1.2.3.- El bus de datos.- El bus de datos del display se puede configurar para funcionar de dos formas diferentes. Bien como un bus de 8 bits o bien como un bus multiplexado de 4 bits. El utilizar el bus multiplexado de 4 bits es

una opción muy útil para ahorrar bits en el sistema de control. En vez de utilizar 11 bits en total, se utilizan 7.

De esta forma se ahorran bits pero se gana en complejidad del controlador, que tiene que multiplexar y demultiplexar los datos. Al utilizar un bus de 8 bits hacemos que el controlador sea más sencillo pero se 'gastan' muchos mas bits.

En la Figura 1.29 aparecen representados los dos tipos de buses.

Cuando se utiliza un bus de 4 bits sólo se utilizan los pines D4-D7 del display dejándose D0-D3 'al aire'. La transferencia de la información se realiza de la siguiente manera: primero los 4 bits más significativos y luego los 4 menos significativos.

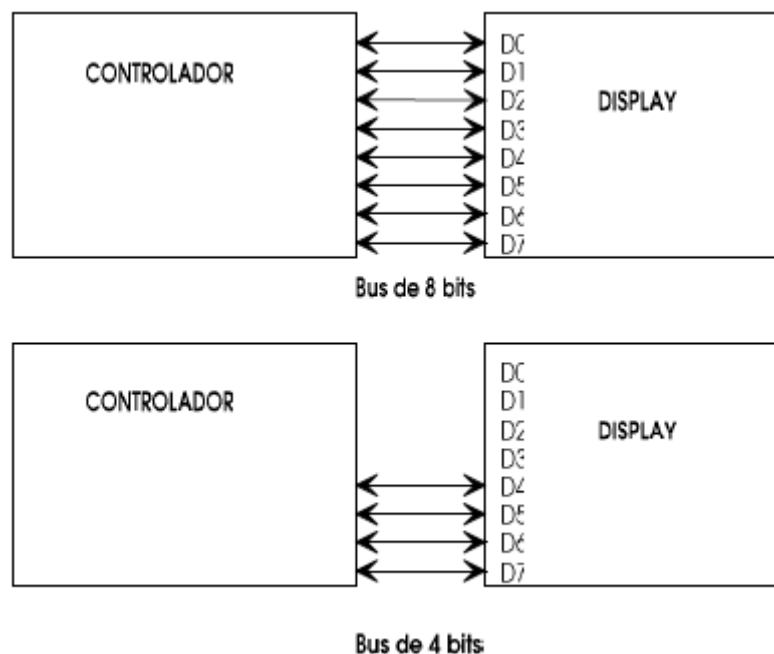


Figura 1.29.- Conexión del LCD utilizando un bus de 8 bits y de 4 bits

1.4.1.2.4.- El bus de control.- El bus de control está formado por 3 señales: RS, R/W y E. La señal E es la señal de validación de los datos. Cuando no se utiliza el display esta señal debe permanecer a 0. Sólo en las transferencias de información (lecturas o escrituras) es cuando se pone a nivel 1 para validar los datos, pasando después de un tiempo a nivel 0. En la siguiente sección se explican detalladamente las temporizaciones.

La señal R/W permite seleccionar si la operación que se va a realizar sobre el display es una lectura o una escritura. Cuando R/W=1 se realizan lecturas y cuando R/W=0 escrituras. Lo normal siempre es realizar escrituras, no obstante, el display ofrece la posibilidad de poder leer los contenidos de la memoria CG RAM y DD RAM así como leer el estado interno del display (ocupado o disponible) y el contador de direcciones.

Con RS (Register Select) se selecciona el registro interno del display sobre el que se va a leer/escribir. El LCD dispone de dos registros internos: Registro de control y registro de datos como se observa en la Figura 1.30. Ambos registros son de lectura y escritura. RS=0 selecciona el registro de control. RS=1 el registro de datos.

	REGISTRO DE CONTROL	REGISTRO DE DATOS
LECTURA	Lectura del flag de ocupado (D7) y del contador de direcciones (D0-D6)	Leer contenido de la memoria CG RAM o DD RAM
ESCRITURA	Ejecución de un comando interno: borrar display, desplazar el display, mover cursor...	Escribir en la DD RAM o CG RAM

Figura 1.30.- Registros de control y datos.

1.4.1.2.5.- El control del contraste.- Para controlar el contraste hay que introducir por el Pin Vo una tensión entre 5 y 0 voltios. La tensión típica es de 0.6 voltios. Normalmente se coloca un potenciómetro para poder ajustar en cada momento el contraste más adecuado. En la Figura 1.31 se muestra un esquema típico de control del contraste.

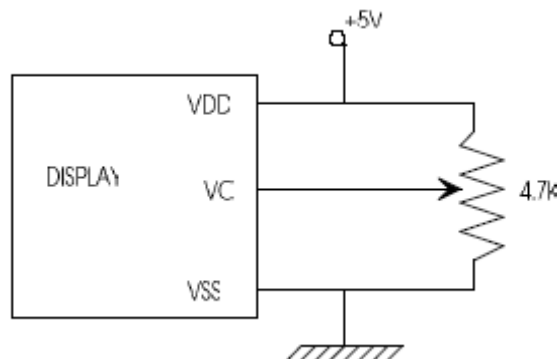


Figura 1.31.- Control del contraste en el LCD

1.4.1.2.6.- Temporización.- En la Figura 1.32 se han representado los cronogramas correspondientes a una operación de escritura y otra de lectura. Al controlar el display los tiempos empleados deben ser siempre mayores que los mínimos indicados en la figura y menores que los máximos. A pesar de la aparente complejidad del cronograma, las operaciones de lectura y escritura son muy sencillas. En la figura 1.32 se ha supuesto que las transmisiones de los datos se realizaban a 8 bits.

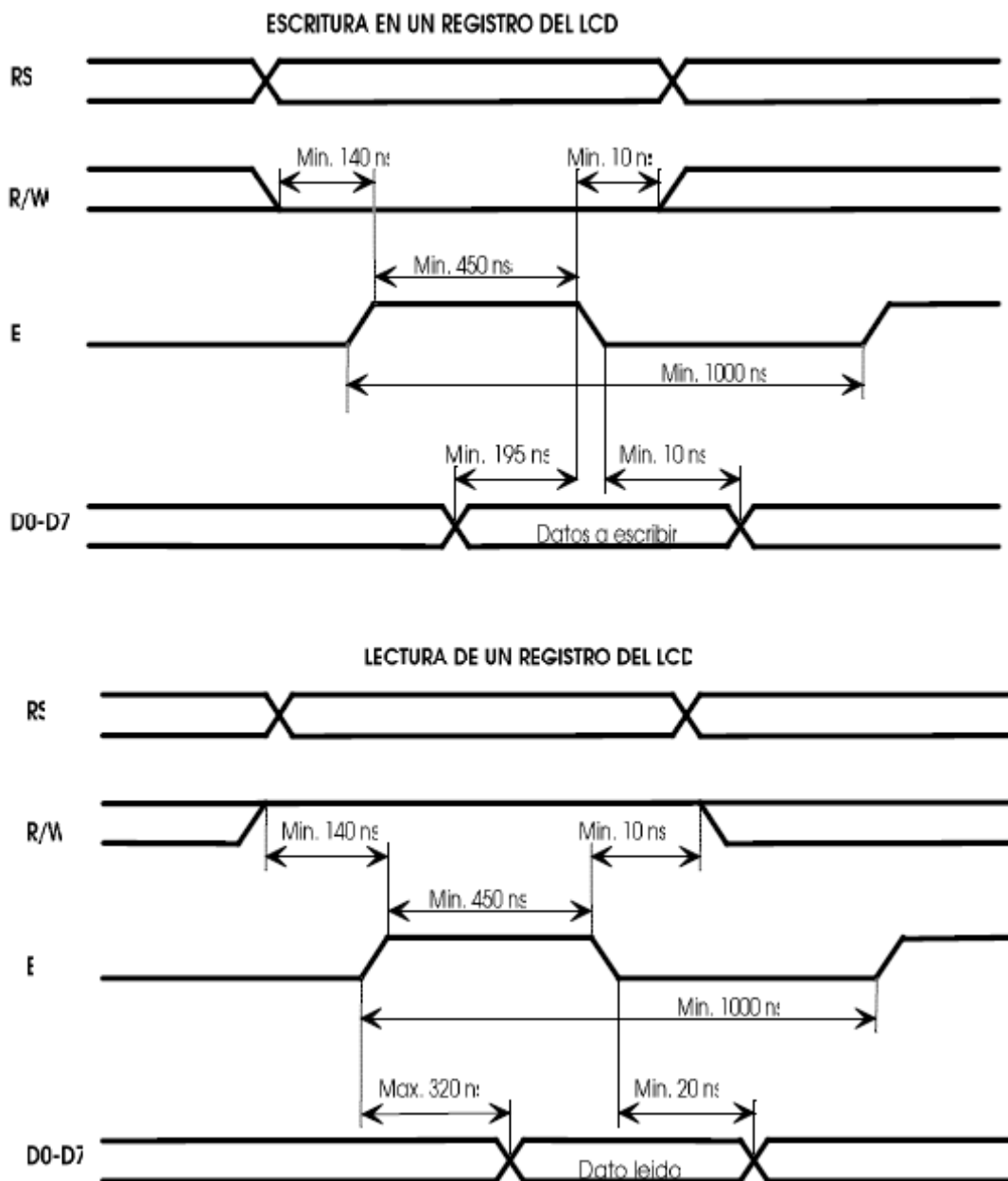


Figura 1.32.- Cronogramas del LCD

En la Figura 1.33 se muestran los cronogramas correspondientes a las operaciones más habituales: escritura de un carácter en el LCD. Se presenta el cronograma cuando se usa un bus de 8 bits y cuando se usa un bus multiplexado de 4 bits. Se ha supuesto que el carácter enviado es el \$41 (La letra 'A').

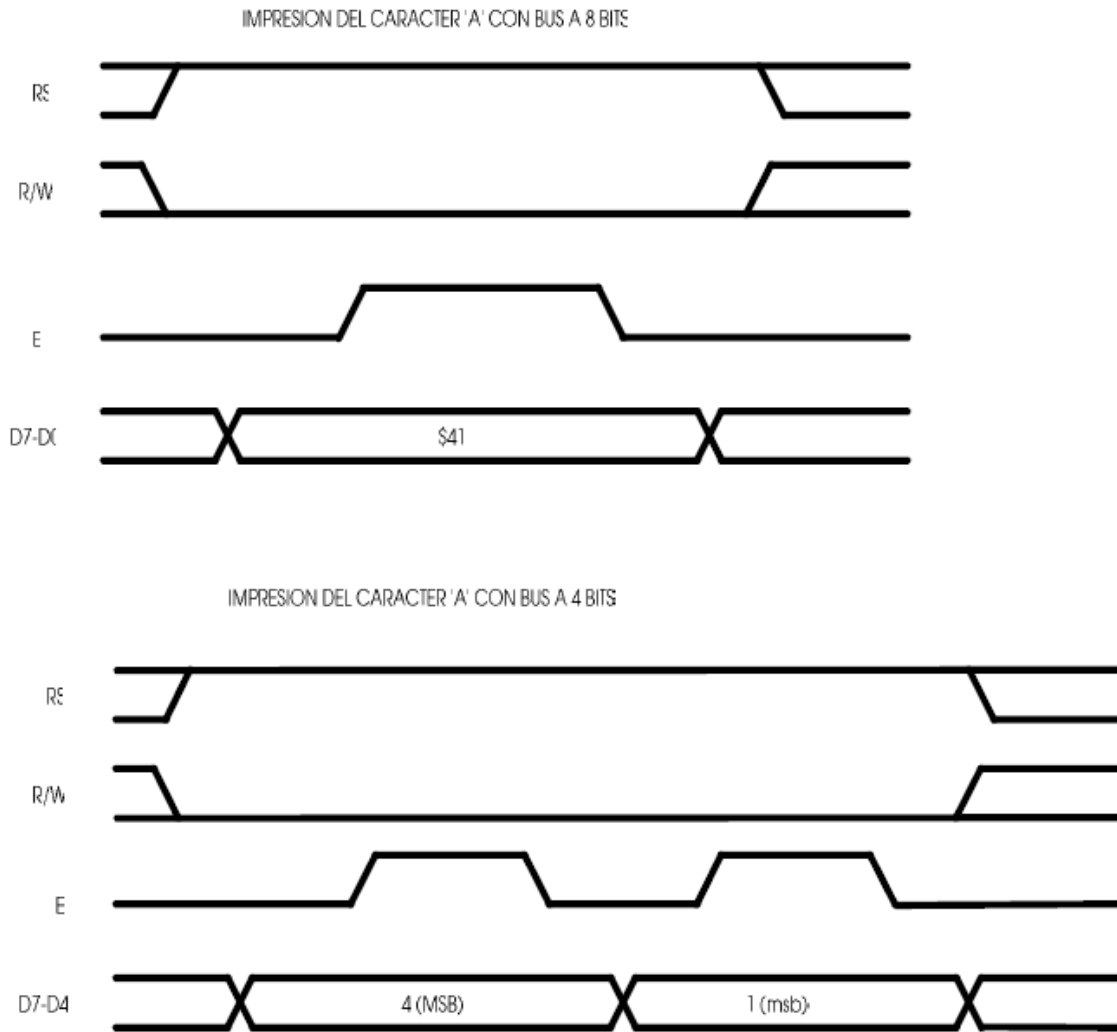


Figura 1.33.- Cronogramas correspondientes a la operación de escribir el dato \$41 (Carácter 'A') en el display utilizando un bus de datos de 8 y 4 bits.

Para el caso de 4 bits, primero se envían los 4 bits MAS SIGNIFICATIVOS y después los 4 bits menos significativos. Los cronogramas de arriba se pueden expresar 'secuencialmente', describiendo las operaciones que tendría que realizar un microcontrolador para escribir el carácter A en el LCD.

- **Operaciones a realizar para el caso de 8 bits:**

1. La señal E se encuentra siempre a 0 antes de realizar cualquier operación.
2. Poner RS=1 y R/W=0.
3. Situar el dato a imprimir en el bus de datos del LCD (En este ejemplo se enviaría \$41).
4. E=1.
5. E=0.
6. El carácter ha sido imprimido en el LCD.

- **Operaciones a realizar para el caso de 4 bits:**

1. Poner RS=1 y R/W=0.
2. Situar el valor 4 en el bus de datos del LCD (4 bits más significativos).
3. E=1.
4. E=0.
5. Situar el valor 1 en el bus de datos del LCD (4 bits menos significativos).
6. E=1.
7. E=0.
8. El carácter ha sido imprimido en el LCD.

1.5.- Teclado

Un teclado es un conjunto de conmutadores o interruptores. Ha menudo tienen imperfecciones, y en el momento del contacto se producen principalmente dos tipos de problemas:

- Ruido eléctrico
- “Rebotes” (*contact bounce*).

Un interruptor se conecta, habitualmente como se indica en la figura 1.34, la resistencia sirve para disponer de una tensión “5 V.” cuando el interruptor se encuentra abierto, mientras que la tensión de salida es “0” cuando el interruptor está cerrado. Ya que normalmente se piensa que la variable asociada con el interruptor es cierta cuando éste se encuentra cerrado, a señal V está en lógica negativa.

Esta elección se debe a la naturaleza de la lógica TTL, que requiere una baja resistencia de conexión a tierra para conseguir una señal “baja”. Por este motivo se conecta la resistencia a “5 V.” y el interruptor a tierra. Esta configuración no es necesaria pero se suele utilizar en tecnología MOS, buscando la compatibilidad con diseños TTL.

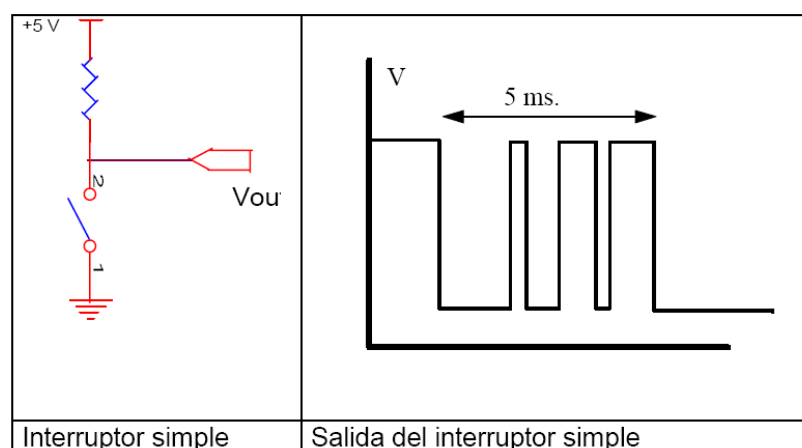


Figura 1.34.- Diagrama de interruptor simple.

Como en ocasiones anteriores, el ruido eléctrico y los rebotes se pueden solucionar mediante hardware (análogo o digital) o mediante software.

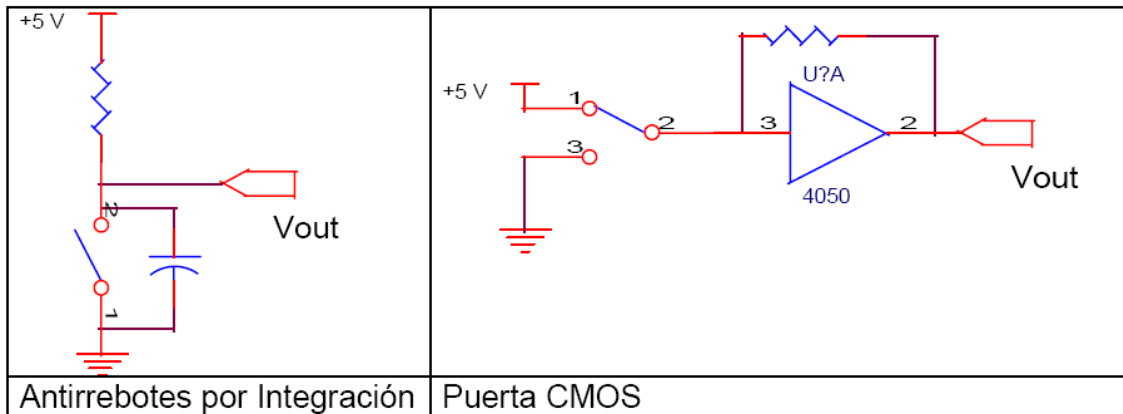


Figura 1.35.- Diagrama de circuitos antirrebotes.

En los anteriores ejemplos se observa el comportamiento de una tecla elemental, y en la que no podemos asegurar el valor de la tensión hasta después de pasados 5 ms de la conmutación. (Valores típicos para un teclado.)

Como dato de referencia un operador humano no puede pulsar y dejar una tecla en menos de 20 mseg.

1.5.1.- Técnicas de barrido del teclado.- Los teclados se suelen ordenar en estructuras matriciales. Existen dos grandes grupos:

- El teclado lineal (selector lineal), que utiliza un decodificador y un dispositivo seleccionador. Es útil hasta unas 10 teclas.

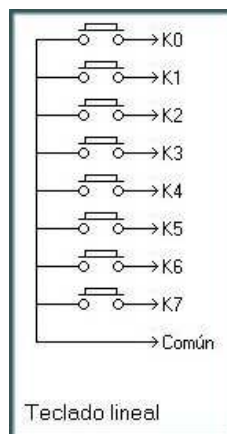
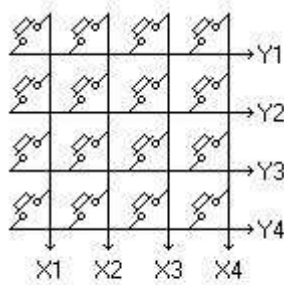


Figura 1.36.- Teclado lineal.

- El teclado matricial (selector por coincidencia), que necesita de dos decodificadores y dos dispositivos seleccionadores.



Teclado Matricial

Figura 1.37.- Modelo Teclado matricial.

1.5.2.- Teclado Matricial 4x4.- Dispositivo de entrada de datos que consta de 16 teclas o pulsadores, dispuestos e interconectados en filas y columnas. Dispone de un conector SIL (Single In Line) macho de 8 pines que se corresponden con las cuatro filas y las cuatro columnas de las que dispone.



Figura 1.38.- Teclado Matricial de 4x4

En la Figura 1.38 vemos el esquema de conexionado interno del teclado matricial y sus correspondientes pines de salida numerados de izquierda a derecha mirando el teclado tal y como se ve en la foto anterior. Cuando se presiona un pulsador se conecta una fila con una columna, teniendo en cuenta este hecho es muy fácil averiguar que tecla fue pulsada. También podemos ver el conexionado típico con el puerto B del μ C PIC.

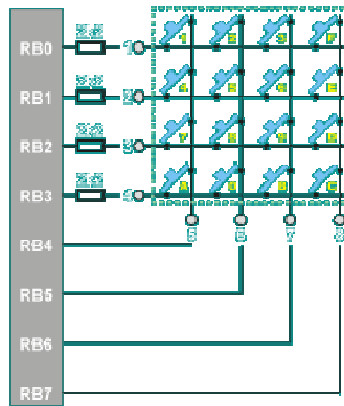


Figura 1.38- Conexión Típico con el puerto B del PIC

Las resistencias de 2k2 son necesarias para poder compartir el puerto del PIC independientemente del teclado y por ejemplo poder conectar un LCD o una barra de leds al mismo tiempo. Durante la fase de lectura del teclado la mitad de la puerta B es configurada como entrada y la otra mitad como lectura y durante la escritura en el LCD u otro sistema, la puerta B es configurada como salidas. Entonces se podrían cortocircuitar accidentalmente las salidas de los puertos provocando su destrucción, si pulsásemos alguna tecla es ese momento con lo que al poner estas resistencias evitamos este hecho y así si se produjera el cortocircuito tan solo circularía una pequeña corriente y el puerto del μC no correría ningún riesgo.

1.6.- Dispositivo Mecánico

Para el presente proyecto era necesario acoplar la parte electrónica con lo mecánico, por lo que se procedió a dar forma física al odómetro, basándose en medidores de distancia comerciales, los cuales constan de una rueda en su parte inferior, la misma que es sujeta por un trinche en cuya parte superior tiene un mango con el que se traslada de un lugar a otro dicho aparato.

Lo interesante de este trabajo fue el dispositivo mecánico que permita sensar la distancia recorrida por lo que se construyó una pieza circular plástica, que debería ir acoplada mecánicamente con la rueda del odómetro para que gire de la misma manera que la rueda principal pero que sea de menor diámetro, y a su vez perforada "n" veces en su parte externa como se describe en el siguiente grafico.

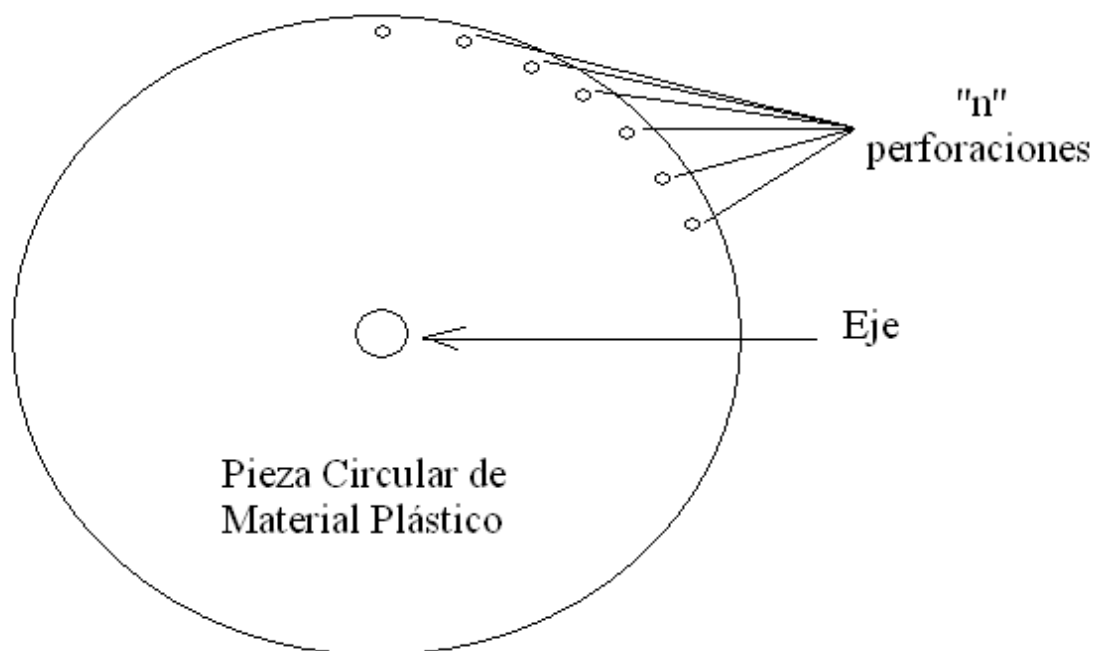


Figura 1.39.- Pieza circular de "n" perforaciones

1.6.1.- Funcionamiento.- la idea principal para el correcto funcionamiento de este odómetro digital es realizar la toma de datos cada vez que pase por uno de las "n" perforaciones realizadas a la pieza circular plástica, es decir que los sensores que serán colocados en una posición estratégica para la medición

tendrán que sensor el paso de luz de cada uno de las perforaciones realizadas y transformar el numero de pasos en medida de longitud, así lograremos un mecanismo mediante el cual se realice la transformación de la distancia recorrida en un dato digital que podremos observar en el display que formara parte de este proyecto.

Cabe indicar que a su vez esta pieza circular plástica deberá ser acoplada al eje principal de la rueda que será la encargada de realizar el recorrido dispuesto para efectuar las mediciones.

CAPITULO II

Construcción del Odómetro Digital.

2.1.- Esquema del Odómetro Digital.- para un mejor desarrollo esquemático del Odómetro Digital lo vamos a dividir por etapas las cuales van a ser descritas a continuación:

- Etapa mecánica.
- Etapa de transformación de datos.
- Etapa electrónica
- Etapa de visualización.

2.1.1.- Etapa mecánica.- se trata con este nombre a todos los componentes mecánicos del proyecto por medio de los cuales se realizaran las lecturas físicas, es decir el recorrido de las distancias a medir, como también todos los accesorios en los cuales irán sujetos las partes electrónicas del trabajo en ejecución.

2.1.2.- Etapa de transformación de datos.- se considera dentro de este bloque a los componentes mediante los cuales es factible la conversión de la lectura realizada por el recorrido de la rueda principal del Odómetro a un dato digital.

Dentro de este bloque se puede considerar como principal a los Opto Acopladores distribuidos específicamente para el recorrido hacia delante y hacia atrás, como el que se encargara del conteo de los metros recorridos.

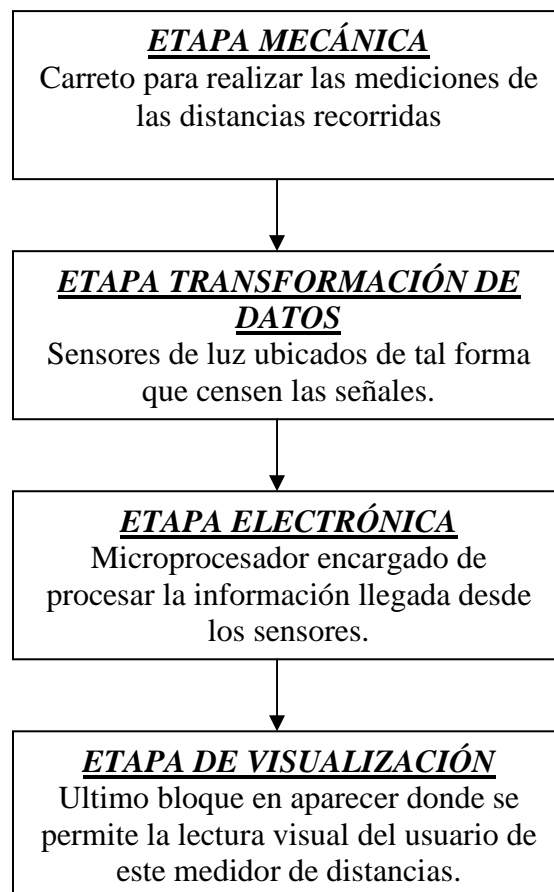
2.1.3.- Etapa Electrónica.- es la más compleja del proyecto y podría considerarse como el cerebro del Odómetro Digital, visto que es aquí que se realizan las operaciones necesarias para la conversión de los datos digitales enviados por los sensores de la etapa anterior, para ser procesados.

La parte fundamental de este bloque es el microcontrolador que será el encargado del procesamiento de información, el almacenaje de información y operaciones matemáticas que servirán al usuario del medidor de distancias.

2.1.4.- Etapa de Visualización.- podría ser considerada como una parte mas de la etapa anterior pues esta ubicada físicamente en el mismo dispositivo de la parte electrónica, y se trata también de una pieza electrónica, pero se ha tomado por separado puesto que es la ultima en aparecer y mostrar los datos obtenidos en todo el proceso a la vista del usuario.

Consta esencialmente de un Display de Cristal Líquido de 2 líneas y 16 caracteres.

ESQUEMA DEL ODÓMETRO DIGITAL.



2.2.- Partes que conforman el Odómetro Digital.- cada una de las partes que forman este proyecto serán descritas por la etapa a las cuales pertenecen, así empezaremos por:

2.2.1.- Partes de la Etapa Mecánica.- como se le ha indicado anteriormente esta conformada por materiales mecánicos, que permitirán el montaje de la parte electrónica del trabajo en ejecución, y son las siguientes:

- Una rueda de caucho, con su eje de plástico.
- Un trinche de bicicleta.
- Un bastón de aluminio.
- Pernos, remaches, etc.



Figura 2.1.- Elementos de la parte mecánica.

2.2.2.- Partes de la Etapa de Transformación de Datos.- esencialmente se trata de dos materiales los cuales son:

- Rueda perforada de plástico.
- Opto acopladores del tipo ranurado.



Figura 2.2.- Elementos que conforman la etapa de Transformación de datos.

2.2.3.- Partes de la Etapa Electrónica.- son varios materiales electrónicos y se los enlista de la siguiente forma:

- Capacitores electrolíticos de 10 uF.
- Capacitores tipo pastilla de 22 pF y 0.1 uF.
- Diodos emisores de luz (LED).
- Diodos del tipo 1N4007.
- Circuito Integrado 7805 (Regulador de Voltaje Positivo).
- Potenciómetro de 20 K ohmios.
- Transistores del tipo 2N3906 (PNP).
- Resistencias de 470 ohmios.
- Resistencia de 100 K ohmios.
- Resistencias de 1 K ohmios.
- Resistencias de 3.9 K ohmios.
- Resistencias de 4.7 K ohmios.
- Resistencias de 330 K ohmios.
- Resistencia de 10 ohmios.
- Cristal de 20 Mhz.
- Microcontrolador PIC16F876A
- Jumpers, portapines, postes, etc.

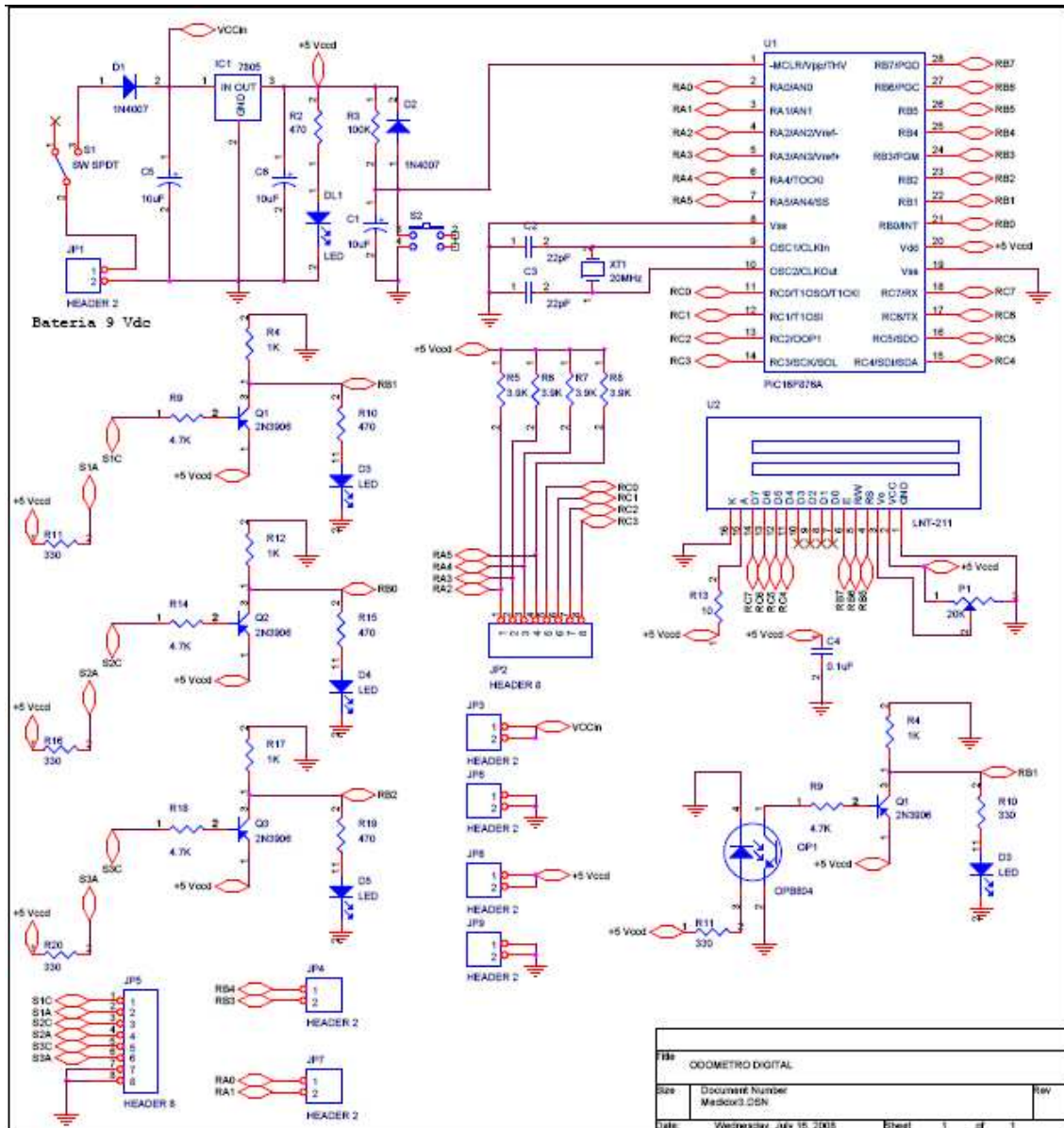


Figura 2.3.- Circuito general de los elementos electrónicos.

2.2.4.- Partes de la Etapa de Visualización.- como se lo ha manifestado en la descripción de las etapas, esta es la ultima en aparecer y es la que esta en contacto con el usuario del Odómetro y sus elementos son:

- Pantalla LCD de 2 filas de 16 caracteres.
- Teclado matricial 4x4.



Figura 2.4.- Elementos periféricos de la etapa de visualización.

2.3.- Elaboración del Programa.- se ha programado el microcontrolador PIC16F876A, para que realice la transformación de la medición a datos que se puedan observar en la pantalla del LCD, así como también el almacenaje de los datos leídos, y operaciones tales como: calculo de área, volumen y suma de datos.

Por tal motivo se describe el detalle de la programación del PIC en Mikrobasic, a continuación:

Para empezar a usar el programador en Mikrobasic, tenemos que ingresar un nuevo proyecto a crea así nos desplegara una pantalla en la cual se solicitara:

- El nombre del proyecto a crear, en este caso lo hemos puesto "Medidor2".
- Luego en donde se lo va a Guardar; D:\Documentsos\TESIS POLI\medidor\mikrobasic\, este será el sitio donde se archivara.
- Luego solicita una breve descripción, que se la puede poner o no.
- A continuación se escoge el dispositivo a usarse, en este caso en particular el P16F876A, de varios que existen para programar con este software.

- Se escoge también el reloj (oscilador), con el que va a trabajar el micro en el proyecto real, para mi trabajo se escogió uno de 20 MHz. debido a que se necesita uno con alta velocidad, para efectuar las mediciones a la velocidad que corre el mecanismo.
- A continuación viene una ventana en el que se configura los flags, dispositivos de grabación, etc.

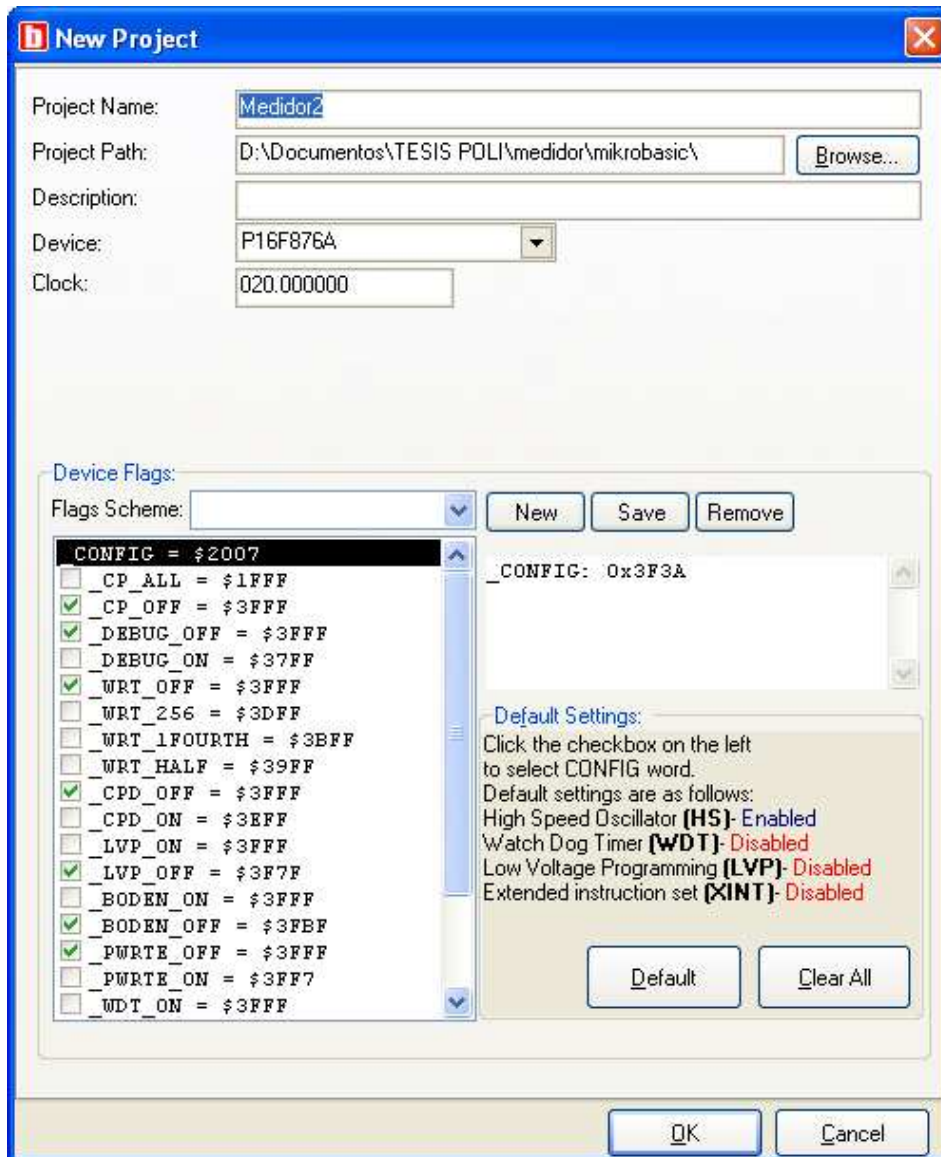


Figura 2.5.- Ventana del nuevo proyecto en MikroBasic.

Luego de realizado este primer paso aparece una pantalla nueva con la primera línea con la cual empezara la programación, en la misma aparecerá lo siguiente:

program Medidor2

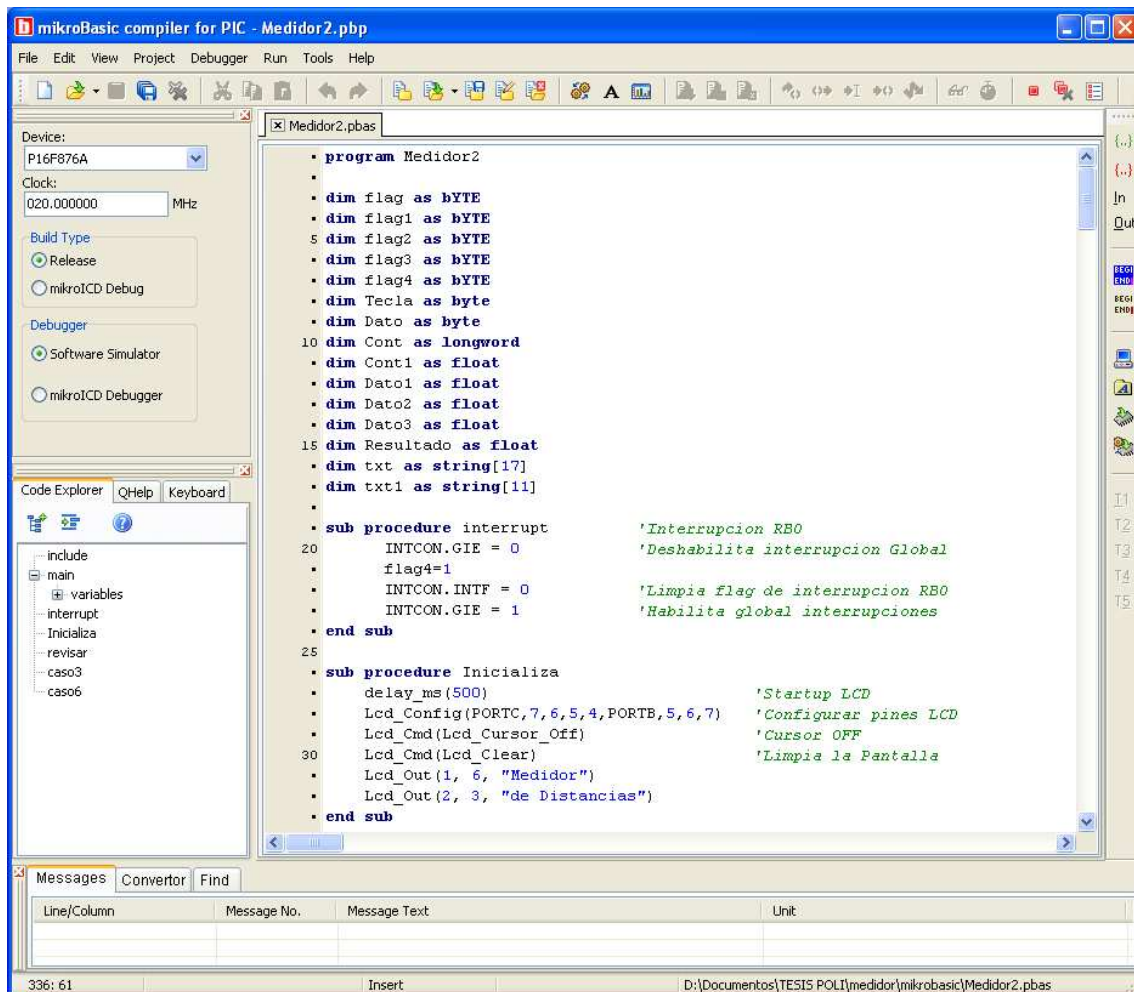


Figura 2.6.- Ventana principal de MikroBasic.

Luego se tiene que empezar a poner símbolos, constantes, variables, subrutinas.

A continuación la palabra "MAIN:" todo lo que se requiera para la programación y por ultimo la palabra "END." Con lo que se finaliza la programación; en el anexo 1 del presente trabajo se encontrara la programación realizada para el odómetro digital con la cual se realiza el conteo, el guardado de datos en la memoria del PIC, y las operaciones que se han descrito anteriormente.

En esta sección se describirá cada uno de las subrutinas, y partes de la programación, como comandos utilizados en el mikrobasic.

En la línea 1 va colocado el nombre del programa.

1: program Medidor2 *“Nombre del programa”*

Desde la línea 3 hasta la línea 17 se especifica las variables a utilizarse y de que tipo van hacer.

3:dim flag as bYTE	<i>“Definición variable”</i>
4:dim flag1 as bYTE	<i>“Definición variable”</i>
5:dim flag2 as bYTE	<i>“Definición variable”</i>
6:dim flag3 as bYTE	<i>“Definición variable”</i>
7:dim flag4 as bYTE	<i>“Definición variable”</i>
8:dim Tecla as byte	<i>“Definición variable”</i>
9:dim Dato as byte	<i>“Definición variable”</i>
10:dim Cont as longword	<i>“Definición variable”</i>
11:dim Cont1 as float	<i>“Definición variable”</i>
12:dim Dato1 as float	<i>“Definición variable”</i>
13:dim Dato2 as float	<i>“Definición variable”</i>
14:dim Dato3 as float	<i>“Definición variable”</i>
15:dim Resultado as float	<i>“Definición variable”</i>
16:dim txt as string[17]	<i>“Definición variable”</i>
17:dim txt1 as string[11]	<i>“Definición variable”</i>

Desde la línea 19 hasta la línea 24 esta colocada la subrutina que controlara la línea RB0 que comandara la secuencia de conteo del medidor.

19:sub procedure interrupt	'Interruccion RB0
20: INTCON.GIE = 0	'Deshabilita interrupcion Global
21: flag4=1	
22: INTCON.INTF = 0	'Limpia flag de interrupcion RB0
23: INTCON.GIE = 1	'Habilita global interrupciones
24:end sub	

Desde la línea 26 hasta la línea 33 se ha realizado una subrutina para inicialización del LCD, asignando las filas y columnas que se han establecido para el funcionamiento del mismo, apagando el cursor, limpiado de pantalla y asignando desde donde queremos que se visualice el texto "Medidor" en la primera línea del LCD y de igual forma el texto "de Distancias" en la segunda línea del LCD, para que quede centrado en el mismo.

26:sub procedure Inicializa

```
27: delay_ms(500)                'Startup LCD  
28: Lcd_Config(PORTC,7,6,5,4,PORTB,5,6,7) 'Configurar pines LCD  
29: Lcd_Cmd(Lcd_Cursor_Off)      'Cursor OFF  
30: Lcd_Cmd(Lcd_Clear)          'Limpia la Pantalla  
31: Lcd_Out(1, 6, "Medidor")  
32: Lcd_Out(2, 3, "de Distancias")  
33:end sub
```

Desde la línea 35 hasta la línea 45, esta realizada una subrutina para que controle el avance o el retroceso de la medición, es decir existe dos posibilidades, si uno de los opto acopladores que se encuentran en el aparato envía la señal antes que el otro este registrará para que el microcontrolador tome la decisión de avanzar en la cuenta o restar la cuenta, así solo uno de los dos dispositivos funcionaran a la vez originando el proceso indicado anteriormente.

35:sub procedure revisar

```
36: if (TestBit(PORTB,2) and (flag4=1)) then  
37:     FLAG4=0  
38:     cont=cont-1  
39: end if  
40: if (TestBit(PORTB,1) and (flag4=1)) then  
41:     FLAG4=0  
42:     cont=cont+1  
43: end if  
44: end if  
45:end sub
```

Desde la línea 47 hasta la línea 60, se ha realizado la subrutina para efectuar la operación del área de un rectángulo, así como se ha limitado el ingreso de los datos hasta 10.000 metros, puesto que la multiplicación de los mismos no alcanzaría a representar el PIC por su capacidad, y en ese caso aparecerá en la pantalla el texto "Overflow", caso contrario desplegara en el LCD el texto "Medición: AREA" en la primera línea y en la segunda línea el dato de la operación seguida por el texto "mts2".

```
47:sub procedure caso3  
48:  if((Dato1 >= 10000.0) or (Dato2>=10000.0))then  
49:    Lcd_Cmd(Lcd_Clear)  
50:    Lcd_Out(1, 1, "Overflow")  
51:    Lcd_Out(2, 1, "Presione A Medir")  
52:  else  
53:    Resultado=Dato1*Dato2  
54:    Lcd_Cmd(Lcd_Clear)  
55:    Lcd_Out(1, 1, "Medicion: AREA")  
56:    FloatToStr(Resultado, txt)  
57:    Lcd_Out(2, 1, txt)  
58:    Lcd_Out(2, 13, "mts2")  
59:  end if  
60:end sub
```

Desde la línea 62 hasta la línea 75, se ha realizado la subrutina para efectuar la operación del volumen de un cubo, así como se ha limitado el ingreso de los datos hasta 1.000 metros, puesto que la multiplicación de los datos en este caso tres no alcanzaría a representar el PIC por su capacidad, y por consiguiente aparecerá en la pantalla el texto "Overflow", caso contrario desplegara en el LCD el texto "Medición: VOLUMEN" en la primera línea y en la segunda línea el dato de la operación seguida por el texto "mts3".

```
62:sub procedure caso6  
63:  if((Dato1 >= 1000.0) or (Dato2>=1000.0) or (Dato3>=1000.0 ))then  
64:    Lcd_Cmd(Lcd_Clear)
```

```

65:   Lcd_Out(1, 1, "Overflow")
66:   Lcd_Out(2, 1, "Presione A Medir")
67:  else
68:   Resultado=Dato1*Dato2*Dato3
69:   Lcd_Cmd(Lcd_Clear)
70:   Lcd_Out(1, 1, "Medicion:VOLUMEN")
71:   FloatToStr(Resultado, txt)
72:   Lcd_Out(2, 1, txt)
73:   Lcd_Out(2, 13, "mts3")
74:  end if
75: end sub

```

En las siguientes líneas de la 77 a la 81, se realiza la definición de las entradas al micro en los puertos A, B, y C, tanto para el teclado que se le ha considerado para un bus de 4 bits, opto acopladores y LCD (tanto Datos y Control), como también se habilita los pórtilos A como datos digitales.

```

77: main:
78: TRISA = %00111100      'Columnas del TECLADO
79: TRISB = %00000111    'Entradas de los Optoacopladores y Control LCD
80: TRISC = %00000000    'Lcd Datos y filas del Teclado
81: ADCON1 = %00000111  'Pines del PORTA como digitales

```

De la línea 83 a la 86, son parámetros para inicializar el conteo de datos medidos por el opto acoplador de medición.

```

83: INTCON.GIE = 0      'Deshabilita interrupcion Global
84: INTCON.INTF = 0    'Limpia flag de interrupcion RB0
85: INTCON.INTE = 1    'Habilita interrupcion RB0
86: OPTION_REG.INTEDG=0  'Interrupcion en flanco de Bajada

```

Desde la línea 87 hasta la línea 97, se inicializa el programa para que no salga nada al momento de encender el Odómetro.


```
87:'Inicializa Variables
88:Dato1=0.0
89:Dato2=0.0
90:Dato3=0.0
91:Resultado=0.0
92:flag1=1
93:flag2=1
94:flag4=0
95:Dato=0
96:Cont=0
97:txt="          "
```

En la línea 98 y 99 se inicializa los casos realizados en pequeños programas cada uno para las diferentes opciones del programa.

```
98:Inicializa
99:INICIO:
```

De la línea 100 hasta la 103, esta realizando un lazo infinito hasta verificar que tecla es la que el usuario pulsa para ver que caso es el que debería correr.

```
100:   Tecla=255
101:   GOSUB Barrido
102:   GOSUB Rebote
103:   select case Tecla
```

Desde la línea 104 hasta la línea 108, se realiza un lazo en el cual se ingresa un dato uno medido a la memoria del PIC, a través de la tecla N°1, para luego ser visto en el LCD.

```
104:   case 1
105:       Dato1=Resultado
106:       Lcd_Cmd(Lcd_Clear)
107:       Lcd_Out(1, 1, "Grabado DATO1")
```

108: Lcd_Out(2, 1, "Presione A Medir")

Desde la línea 109 hasta la línea 113, se realiza un lazo en el cual se ingresa un dato dos medido a la memoria del PIC, a través de la tecla N°4, para luego ser visto en el LCD.

109: case 4

110: Dato2=Resultado

111: Lcd_Cmd(Lcd_Clear)

112: Lcd_Out(1, 1, "Grabado DATO2")

113: Lcd_Out(2, 1, "Presione A Medir")

Desde la línea 114 hasta la línea 118, se realiza un lazo en el cual se ingresa un dato tres medido a la memoria del PIC, a través de la tecla N°7, para luego ser visto en el LCD.

114: case 7

115: Dato3=Resultado

116: Lcd_Cmd(Lcd_Clear)

117: Lcd_Out(1, 1, "Grabado DATO3")

118: Lcd_Out(2, 1, "Presione A Medir")

En las líneas 119 y 120, se presenta el lazo que envía directamente a la subrutina caso 3, al momento que el usuario presiona la tecla 3.

119: case 3

120: caso3

En las líneas 121 y 122, se presenta el lazo que envía directamente a la subrutina caso 6, al momento que el usuario presiona la tecla 6.

121: case 6

122: caso6

Desde la línea 123 hasta la línea 128, se ha realizado un lazo en el cual se limpia el LCD, para que aparezca el dato guardado en la memoria del PIC cuando se presiono la tecla N¹, al momento de pres ionar la tecla N².

```
123:    case 2
124:        Lcd_Cmd(Lcd_Clear)
125:        Lcd_Out(1, 1, "Dato 1")
126:        FloatToStr(Dato1, txt)
127:        Lcd_Out(2, 1, txt)
128:        Lcd_Out(2, 13, "mts.")
```

Desde la línea 129 hasta la línea 134, se ha realizado un lazo en el cual se limpia el LCD, para que aparezca el dato guardado en la memoria del PIC cuando se presiono la tecla N⁴, al momento de pres ionar la tecla N⁵.

```
129:    case 5
130:        Lcd_Cmd(Lcd_Clear)
131:        Lcd_Out(1, 1, "Dato 2")
132:        FloatToStr(Dato2, txt)
133:        Lcd_Out(2, 1, txt)
134:        Lcd_Out(2, 13, "mts.")
```

Desde la línea 135 hasta la línea 140, se ha realizado un lazo en el cual se limpia el LCD, para que aparezca el dato guardado en la memoria del PIC cuando se presiono la tecla N⁷, al momento de pres ionar la tecla N⁸.

```
135:    case 8
136:        Lcd_Cmd(Lcd_Clear)
137:        Lcd_Out(1, 1, "Dato 3")
138:        FloatToStr(Dato3, txt)
139:        Lcd_Out(2, 1, txt)
140:        Lcd_Out(2, 13, "mts.")
```

De la línea 141 a la línea 147, se ha realizado un lazo que cuando se presione la tecla N° 9, limpie la pantalla del LCD, envíe el texto "Medicion:LINEAL", y presente la suma de los datos uno y dos almacenados en el PIC.

```
141:   case 9
142:     Resultado=Dato1+Dato2
143:     Lcd_Cmd(Lcd_Clear)
144:     Lcd_Out(1, 1, "Medicion:LINEAL")
145:     FloatToStr(Resultado, txt)
146:     Lcd_Out(2, 1, txt)
147:     Lcd_Out(2, 13, "mts.")
```

De la línea 148 a la línea 154, se ha realizado un lazo que cuando se presione la tecla N° #, limpie la pantalla del LCD, envíe el texto "Medicion:LINEAL", y presente la suma de los datos uno, dos y tres almacenados en el PIC.

```
148:   case 15
149:     Resultado=Dato1+Dato2+Dato3
150:     Lcd_Cmd(Lcd_Clear)
151:     Lcd_Out(1, 1, "Medicion:LINEAL")
152:     FloatToStr(Resultado, txt)
153:     Lcd_Out(2, 1, txt)
154:     Lcd_Out(2, 13, "mts.")
```

Desde la línea 155 hasta 159 se ha programado un lazo mediante el cual al presionar la tecla N°0, se realiza el reseteo de todos los datos almacenados en memoria del PIC.

```
155:   case 0
156:     Dato1=0.0
157:     Dato2=0.0
158:     Dato3=0.0
159:     Resultado=0.0
```

De la línea 160 hasta la línea 205, realiza todo lo que es medición, es decir esta chequeando si se avanza o si se retrocede, mediante un flanco de bajada, para que al momento de presionar la tecla N° 10 (A) , nos visualice en el display la palabra "MIDIENDO.....", y se quedara con ese texto hasta que el usuario realice la medición deseada en el camino, mientras tanto, estará revisando si se suma o se resta un dato para al final que se vuelva a presionar la tecla N° 10, aparezca en el display el texto "Medición: " y en la segunda línea la distancia en metros recorrida esto se da gracias a la operación que realiza el PIC en la línea 195, que se le ha colocado que realice la división de $Cont1=(cont*1.11)/44.5$, donde cont es el número de perforaciones medidas por el PIC; 1,11 es la longitud externa de la rueda principal del odómetro, y 44.5 es la constante de calibración con la cual se realiza una mejor medida.

Además existen dos subrutinas Barrido1 y Rebote1, que se las estudiara posteriormente.

Por ultimo en la línea 205 envía nuevamente a la línea 99 para que se vuelva a repetir todo el proceso.

```
160:   case 10
161:     flag1=1
162:     flag2=1
163:     flag3=1
164:     while flag2=1
165:       if flag3=1 then
166:         flag3=0
167:         Tecla=10
168:       else
169:         Tecla=255
170:       end if
171:     revisar
172:     GOSUB Barrido1
173:     GOSUB Rebote1
174:     SELECT CASE Tecla
```

```

175:      case 10
176:          if flag1=1 then
177:              flag1=0
178:          else
179:              flag1=1
180:          end if
181:          IF FLAG1=0 THEN
182:              Cont=0  !*****
183:              Lcd_Cmd(Lcd_Clear)
184:              Lcd_Out(1, 1, "MIDIENDO.....")
185:              flag4=0
186:              INTCON.GIE = 1    'Habilita interrupcion Global
187:          ELSE
188:              INTCON.GIE = 0    'Deshabilita interrupcion Global
189:              Lcd_Cmd(Lcd_Clear)
190:              'ByteToStr(dato,txt2)
191:              LongintToStr(cont, txt1)
192:              Lcd_Out(1,6,txt1)
193:              'Lcd_Out(1,12,txt2)
194:              Lcd_Out(1, 1, "Medicion: ")
195:              Cont1=(cont*1.11)/44.5
196:              Resultado=cont1
197:              FloatToStr(cont1, txt)
198:              Lcd_Out(2, 1, txt)
199:              Lcd_Out(2, 13, "mts.")
200:              flag2=0
201:          END IF
202:      end select
203:  wend
204: end select
205:goto INICIO

```

Desde la línea 206 hasta la línea 216, se ha efectuado un lazo mediante el cual se queda censando solo la fila A del teclado para ganar tiempo de respuesta en el programa pues solo censará esa fila cuando se presione la tecla N°10.

206:BARRIDO1:

```
207:   ClearBit(PORTC,0)   'sensar la fila A  
208:   flag = TestBit(PORTA,5)  
209:   IF flag = 0 THEN  
210:       Tecla=10  
211:'   else  
212:'       Tecla=255  
213:   return  
214:   END IF 'tecla pulsada retorne con variable cargada con 10  
215:   SetBit(PORTC,0)  
216:return
```

De la línea 217 a línea 223, se ha realizado una subrutina para evitar rebotes por la presión de las teclas al momento de digitarlas.

217:Rebote1:

```
218:   Suelta1:           'Subrutina de antirrebote de teclas  
219:   flag = TestBit(PORTA,5)  
220:   IF flag = 0 THEN   'si la tecla sigue pulsada ir espacio  
221:       goto Suelta1   'si la tecla sigue pulsada ir espacio  
222:   end if  
223:RETURN
```

Desde la línea 225 hasta la línea 314, se ha efectuado varios lazos mediante los cuales se censaran cada una de las filas a la que pertenezca la tecla presionada en ese instante en el teclado para ganar tiempo de respuesta en el programa pues solo censará esa fila escogida, es decir si pertenece a la fila A, se quedara solo en esa fila, si pertenece a la fila B, solo esa fila, si la tecla pertenece a la fila C, solo esa fila; si es de la fila C, censará esa única fila y por ultimo si la tecla esta en la fila D, solo censará esa fila.

225: BARRIDO:

```
226:   ClearBit(PORTC,0)      'senyar la fila A
227:   flag = TestBit(PORTA,2)
228:   IF flag = 0 THEN
229:     Tecla=1
230:   return
231:   END IF 'tecla pulsada retorne con variable cargada con 1
232:   flag = TestBit(PORTA,3)
233:   IF flag = 0 THEN
234:     Tecla=2
235:   return
236:   END IF 'tecla pulsada retorne con variable cargada con 2
237:   flag = TestBit(PORTA,4)
238:   IF flag = 0 THEN
239:     Tecla=3
240:   return
241:   END IF 'tecla pulsada retorne con variable cargada con 3
242:   flag = TestBit(PORTA,5)
243:   IF flag = 0 THEN
244:     Tecla=10
245:   return
246:   END IF 'tecla pulsada retorne con variable cargada con 10
247:   SetBit(PORTC,0)
248:   ClearBit(PORTC,1)      'senyar la fila B
249:   flag = TestBit(PORTA,2)
250:   IF flag = 0 THEN
251:     Tecla=4
252:   return
253:   END IF 'tecla pulsada retorne con variable cargada con 4
254:   flag = TestBit(PORTA,3)
255:   IF flag = 0 THEN
256:     Tecla=5
257:   return
258:   END IF 'tecla pulsada retorne con variable cargada con 5
```



```

259:   flag = TestBit(PORTA,4)
260:   IF flag = 0 THEN
261:     Tecla=6
262:   return
263:   END IF 'tecla pulsada retorne con variable cargada con 6
264:   flag = TestBit(PORTA,5)
265:   IF flag = 0 THEN
266:     Tecla=11
267:   return
268:   END IF 'tecla pulsada retorne con variable cargada con 11
269:   SetBit(PORTC,1)
270:   ClearBit(PORTC,2)      'senyar la fila C
271:   flag = TestBit(PORTA,2)
272:   IF flag = 0 THEN
273:     Tecla=7
274:   return
275:   END IF 'tecla pulsada retorne con variable cargada con 7
276:   flag = TestBit(PORTA,3)
277:   IF flag = 0 THEN
278:     Tecla=8
279:   return
280:   END IF 'tecla pulsada retorne con variable cargada con 8
281:   flag = TestBit(PORTA,4)
282:   IF flag = 0 THEN
283:     Tecla=9
284:   return
285:   END IF 'tecla pulsada retorne con variable cargada con 9
286:   flag = TestBit(PORTA,5)
287:   IF flag = 0 THEN
288:     Tecla=12
289:   return
290:   END IF 'tecla pulsada retorne con variable cargada con 12
291:   SetBit(PORTC,2)
292:   ClearBit(PORTC,3)      'senyar la fila D

```

```

293:    flag = TestBit(PORTA,2)
294:    IF flag = 0 THEN
295:        Tecla=14
296:    return
297:    END IF 'tecla pulsada retorne con variable cargada con 14
298:    flag = TestBit(PORTA,3)
299:    IF flag = 0 THEN
300:        Tecla=0
301:    return
302:    END IF 'tecla pulsada retorne con variable cargada con 0
303:    flag = TestBit(PORTA,4)
304:    IF flag = 0 THEN
305:        Tecla=15
306:    return
307:    END IF 'tecla pulsada retorne con variable cargada con 15
308:    flag = TestBit(PORTA,5)
309:    IF flag = 0 THEN
310:        Tecla=13
311:    return
312:    END IF 'tecla pulsada retorne con variable cargada con 13
313:    SetBit(PORTC,3)
314: return

```

Desde la línea 316 hasta la línea 334, se realiza un grupo de subrutinas para evitar los rebotes originados al pulsar las diferentes teclas que comandaran el programa externamente mediante el teclado usado.

```

316: Rebote:
317:    Suelta:                'Subrutina de antirrebote de teclas
318:    flag = TestBit(PORTA,2)
319:    IF flag = 0 THEN
320:        goto Suelta        'si la tecla sigue pulsada ir espacio
321:    end if
322:    flag = TestBit(PORTA,3)

```

```

323:   IF flag = 0 THEN           'si la tecla sigue pulsada ir espacio
324:     goto Suelta              'si la tecla sigue pulsada ir espacio
325:   end if
326:   flag = TestBit(PORTA,4)
327:   IF flag = 0 THEN           'si la tecla sigue pulsada ir espacio
328:     goto Suelta              'si la tecla sigue pulsada ir espacio
329:   end if
330:   flag = TestBit(PORTA,5)
331:   IF flag = 0 THEN           'si la tecla sigue pulsada ir espacio
332:     goto Suelta              'si la tecla sigue pulsada ir espacio
333:   end if
334: RETURN

```

La línea 335 es la última en aparecer y es la que nos indica que el programa a finalizado, sin esta no es factible hacer correr la programación.

335:end.

2.4.- Construcción del Odómetro Digital.- se ha efectuado varias partes para el montaje total del odómetro digital, una fue la construcción mecánica del aparato por medio del cual se va a realizar la toma de la medida física, y luego el montaje electrónico.

2.4.1.- Construcción Mecánica.- el montaje de esta parte se lo ha efectuado de la siguiente manera:

a.- En un trinche de bicicleta, se procedió a montar una rueda principal de caucho, cuyo perímetro es 1.11 metros, además se ha efectuado la adaptación de una rueda perforada en el filo de la misma 44 huecos, los cuales serán los encargados de enviar el recorrido realizado por la rueda de caucho, cabe indicar que las perforaciones se han realizado equidistantes, por lo que esta pieza ha sido construida con suma precisión.

Según se muestra en la figura veremos la primera parte realizada con todas las especificaciones indicadas.

b.- Luego de montado esta parte, se ha instalado un bastón de aluminio para que su manejo sea más accesible por el usuario de este aparato de medición, el mismo que ha sido fijado a la parte final del trinche de bicicleta por medio de remaches.

2.4.2.- Montaje electrónico.- esta es la parte principal del presente trabajo por lo que será explicado cada una de sus partes detenidamente, para una mejor comprensión, así empezaremos por:

a.- Montaje de los opto acopladores.- se han colocado en la periferia de la rueda plástica perforada en una forma estratégica para que solo dos de los tres opto acopladores coincidan al mismo tiempo con los huecos realizados en la rueda, y el otro deberá calzar justo en una parte donde se obstruya el paso de luz con la finalidad de que se envíe una sola señal en el caso de avance o retroceso del recorrido medido.

A continuación veremos el circuito electrónico que regirá la conexión de los opto acopladores:

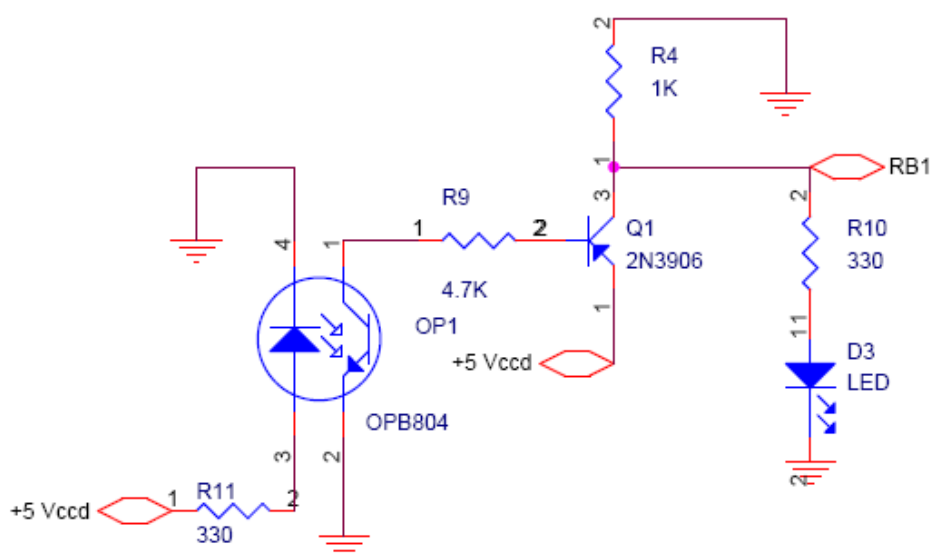


Figura 2.7.- Circuito de conexión para cada uno de los 3 opto acopladores.

El opto acoplador usado para el proyecto es el OPB804, cuyas características técnicas se la se encontrara en la sección de los Anexos de este trabajo.

Como un dato adicional se puede decir que la señal de salida del opto acoplador es muy débil por lo que antes de procesarla es necesario amplificarlo y esa función la realiza el transistor 2N3906, también se ha colocado un diodo emisor de luz (LED) para saber visualmente cual de los opto acopladores esta funcionando.

Este mismo circuito se ha implementado para los tres opto acopladores utilizados en el proyecto, es decir el que cuenta el numero de perforaciones, y los que verifican el sentido de avance y retroceso.

b.- Regulador de voltaje.- los dispositivos usados para este proyecto necesitan una alimentación de 5 V. y como se necesita una fuente de voltaje que en este caso es una batería de 9 V. se hizo necesario implementar un regular de voltaje, y para esto se ha usado el C.I. 7805, que estabiliza el voltaje a 5 voltios, adicional a esto se ha colocado dos diodos 1N4007, para protección del circuito el uno y el otro para un circuito de reseteo general, también se ha usado capacitores electrolíticos para que la carga y descarga sea moderada y no instantánea.

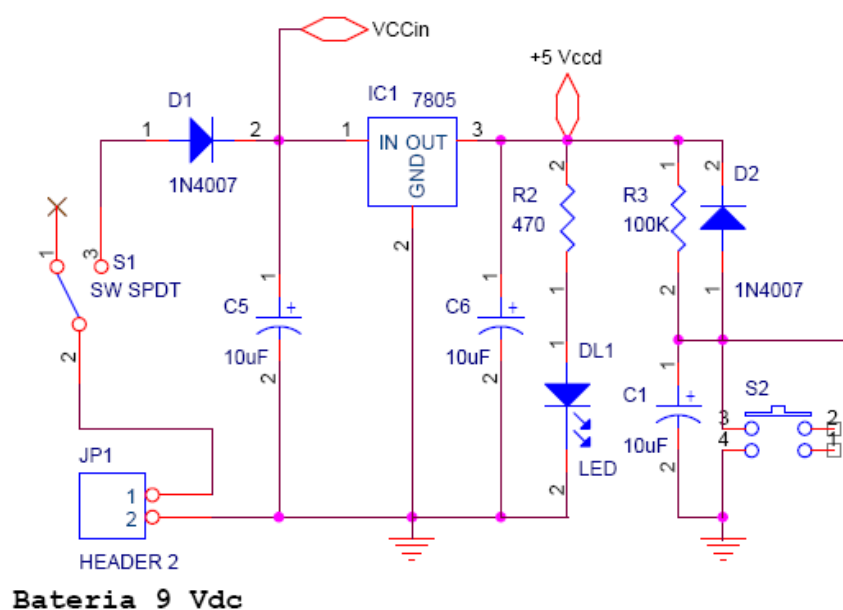


Figura 2.8.- Circuito del regulador de voltaje y circuito de reseteo general.

Se puede también acentuar que la salida del control de reseteo general va a ir directamente al Pin N°1 del PIC, es decir al Master Clear.

c.- Conexión del Teclado Matricial 4 x 4 .- como se ha indicado anteriormente, el teclado usado para este proyecto es un teclado matricial 4 x 4, cuya conexión es muy sencilla, como se muestra a continuación.

Como va a ir externamente se ha colocado un header o zócalo para mediante un bus de datos llevarlo a donde se lo pueda colocar.

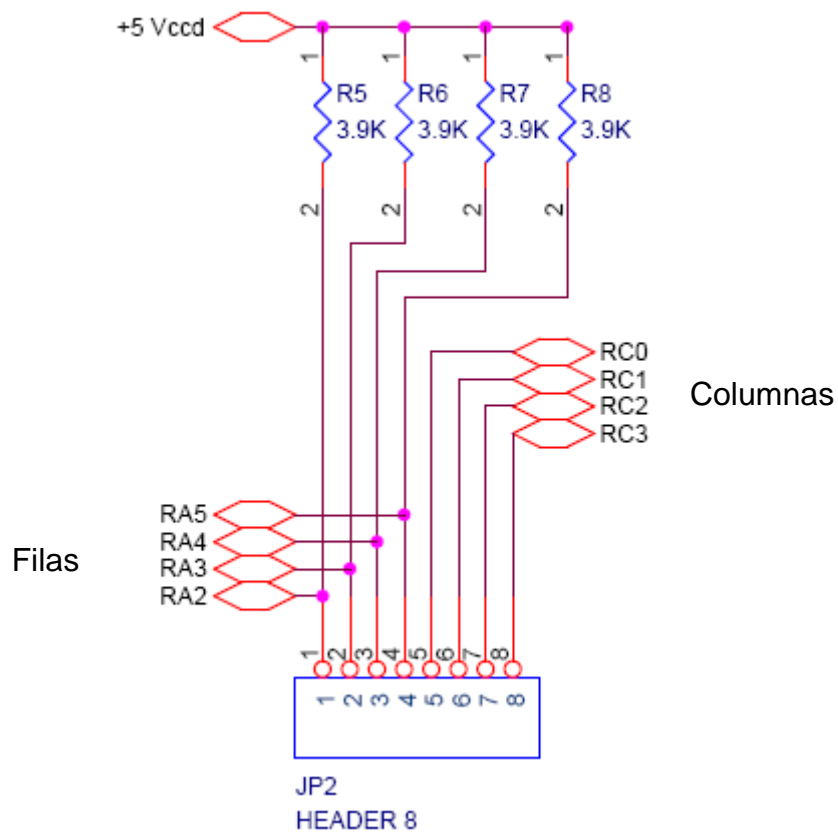


Figura 2.9.- Circuito de conexión para el teclado matricial de 4x4.

d.- Conexión de la pantalla de 2 filas y 16 caracteres (LCD).- el display utilizado para el proyecto es el LNT-211, y la conexión es bastante sencilla pues se tiene que polarizar el display en los pines 16 (tierra) y 15 (Vcc), luego se realiza las conexiones de control que son los pines 6 (Enable), 5 (R/W), y 4 (RS), también se realiza un pequeño circuito de control de brillo que es conectado a los pines 3 (Vo), 2 (Vcc) y 1 (Gnd).

Para nuestro trabajo se ha utilizado un de datos de 4 bits que serán conectados a los pines 14 (D7), 13 (D6), 12 (D5) y 11 (D4).

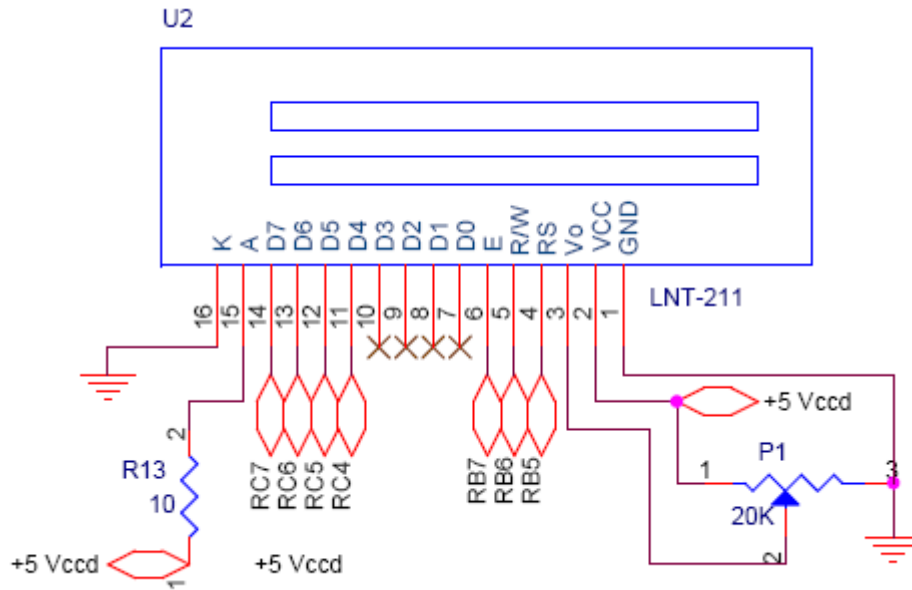


Figura 2.10.- Circuito de conexión para el LCD con un bus de datos de 4 bits.

e.- Conexión del Microcontrolador PIC16F876A.- se han tomado en cuenta las tres entradas A, B y C para ser usadas en el presente trabajo, por tal razón la conexión del Pic se resume en energizar al Pic con un Vdd de +5 V y un Vss igual a Gnd como se muestra en la figura, el Pin N°1 ira conectado al circuito de reseteo general al PIC, analizado anteriormente, colocaremos un oscilador mediante un cristal en nuestro caso de 20 MHz mediante su respectivo circuito expuesto en la figura, por ultimo el resto serán las entradas al PIC.

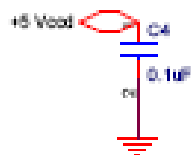


Figura 2.11.- Circuito de aislamiento de ruido al microcontrolador.

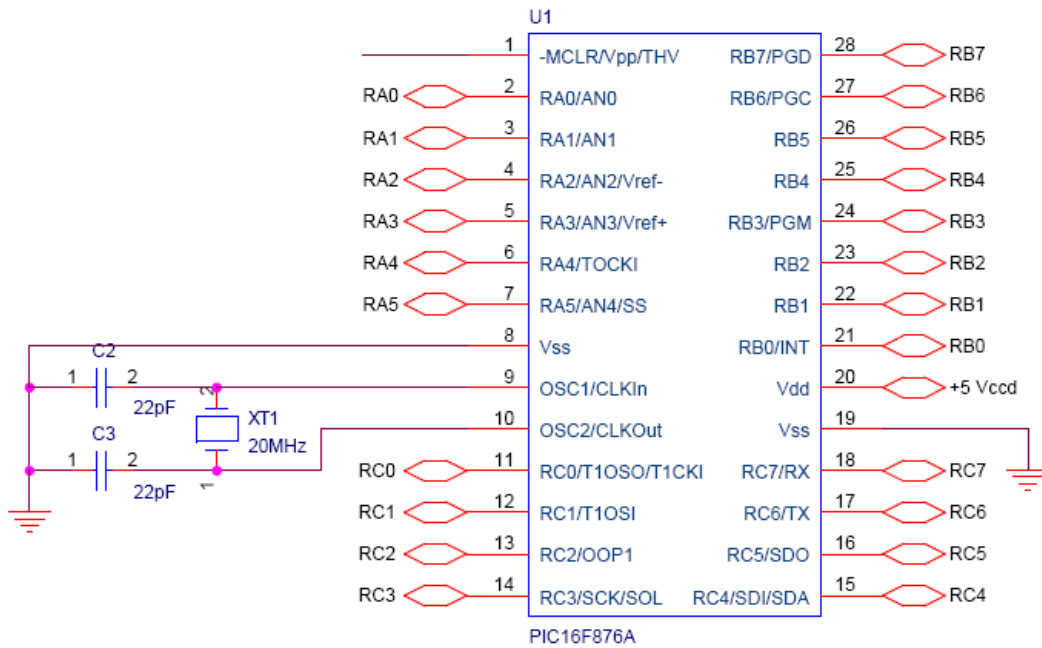


Figura 2.12.- Circuito de conexiones al microcontrolador.

f.- Placa utilizada para el montaje de los dispositivos electrónicos.- como todo los circuitos revisados anteriormente, tienen que ir interconectados entre si en una placa de baquelita, a continuación se mostrara la placa utilizada de diversas vistas para saber donde van los dispositivos de la lista de materiales indicada anteriormente.

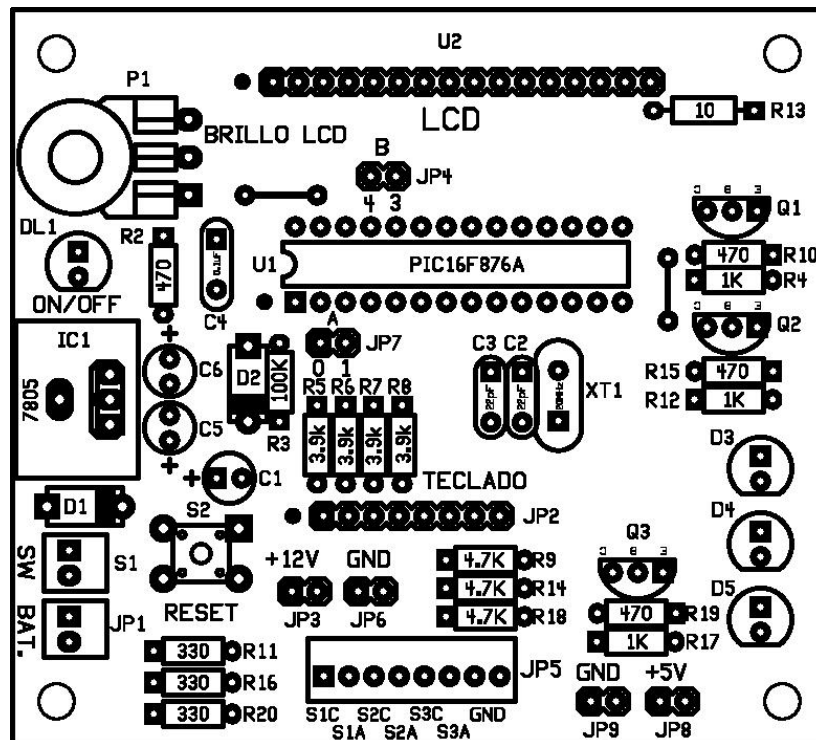


Figura 2.13.- Placa vista desde la parte superior sin elementos soldados.

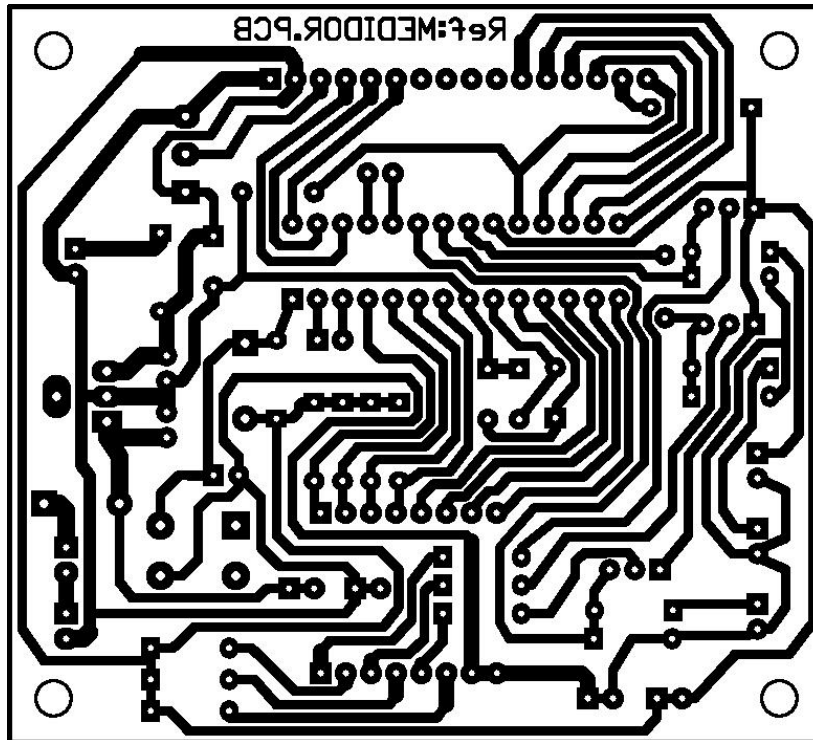


Figura 2.14.- Placa vista desde la parte inferior, es decir únicamente las pistas de cobre.

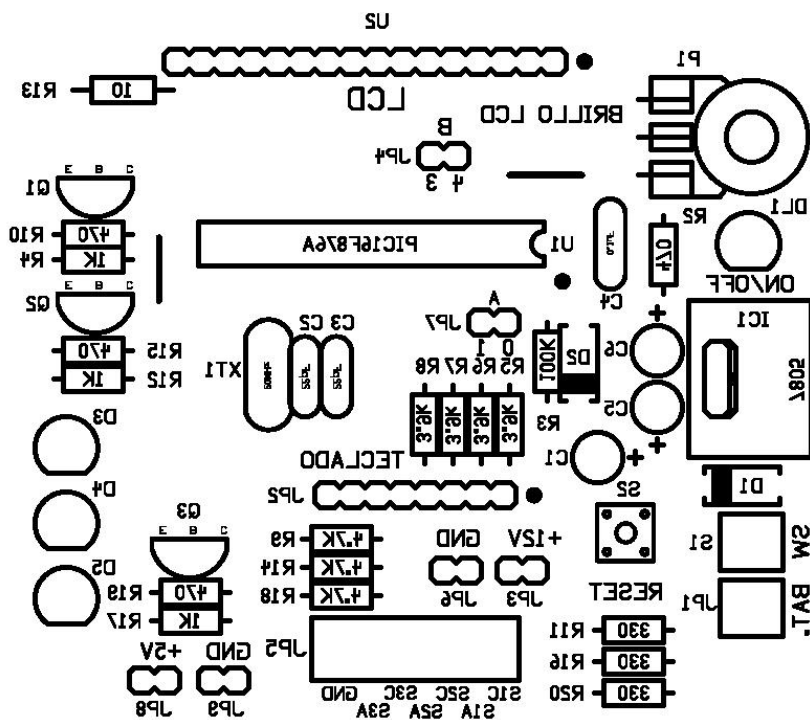


Figura 2.15.- Placa vista desde la parte inferior con los elementos colocados (así es como se debería ver, esto se lo hizo para demostración físicamente es imposible ver esta figura.)

Por ultimo se muestra la placa terminada.



Figura 2.16.- Placa terminada con elementos electrónicos.

g.- Montaje de partes electrónicas en una cajita de plástico.- en la figura observamos el montaje final de los dispositivos y la placa electrónica dentro de una cajita de plástico.



Figura 2.17.- Modelo terminado montado en caja plástica.

2.4.3.- Odómetro Digital terminado.- ahora se muestra el medidor de distancias como quedo finalmente.



Figura 2.18.- Odómetro Digital Terminado.

CAPITULO III

Pruebas y Resultados.

3.1.- Pruebas de funcionamiento del Odómetro Digital.- como todo aparato de medición, se tiene que realizar varias pruebas para calibración del equipo, una vez que esta realizado la programación del microcontrolador y verificado el funcionamiento correcto en software, se realizo las siguientes pruebas de calibración montando el equipo electrónico en la parte mecánica del odómetro digital.

En la Figura 3.1 se visualiza el programa (Icisis) mediante el cual se probó el funcionamiento de la programación, con los principales dispositivos del Odómetro Digital, antes de montar todos los elementos en la plaqueta de baquelita.

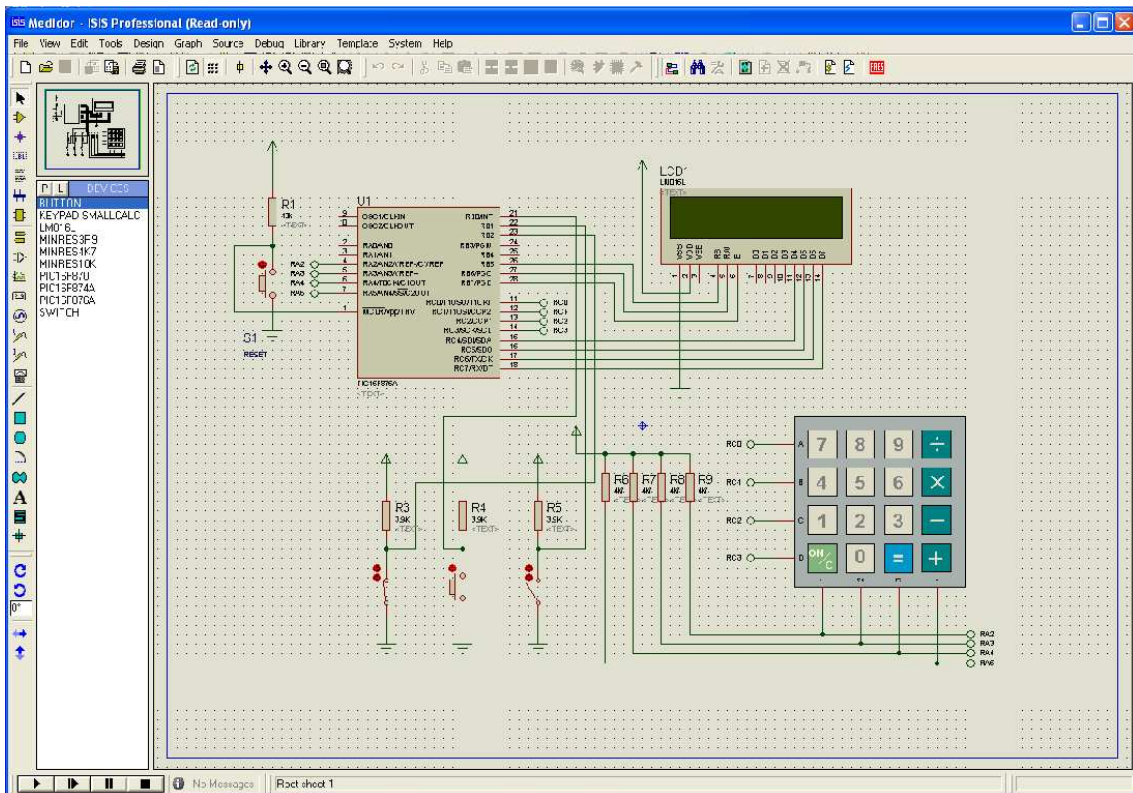


Figura 3.1.- Programa Isis de simulación del proyecto.

3.1.1.- Calibración de Opto Acopladores.- como se ha descrito en paginas anteriores, el mecanismo mediante el cual se transforma la medida realizada en datos que podemos visualizar en la pantalla del LCD, no son mas que tres opto acopladores ubicados estratégicamente y en una posición en la cual el primer opto acoplador en sensor la perforación en la rueda plástica perforada, se encarga de mandar una señal de avance al microcontrolador, el segundo opto acoplador en sensor el agujero es el que se encargara de mandar una señal de la distancia recorrida al microcontrolador, y por ultimo el opto acoplador siguiente en sensor la perforación en cuestión, se encarga de sensor el retroceso, de esta manera, el primero y el ultimo opto acoplador deben estar físicamente colocados en una posición en la cual no sensen al mismo tiempo la perforación que gira con el eje principal de la rueda.



Figura 3.2.- Opto acopladores en posición de la calibración manual.

3.2.- Obtención de resultados.- cuando se probó el circuito en el software (programa Ares), se producía los valores que fueron programados con anticipación, es decir todos los pulsos dados por medio del pulsador que simulaba el paso del haz de luz a través de las perforaciones realizadas en la rueda plástica eran medidos, sin embargo era necesario saber que pasaba cuando todo fuera montado en el lugar donde van a ir fijos en el medidor de distancias, obteniéndose por resultado que no se median todas las perforaciones realizadas, y por tanto teníamos una mala medida.

Las distancias recorridas variaban constantemente cuando se realizaba la toma de una misma medida, generando incertidumbre, pues si se recorría un mismo trayecto la distancia debía ser aproximada.

TABLA DE VALORES OBTENIDOS EN LA REALIDAD						
Item	Valor medido manualmente	Unid.	Valor medido con el Odómetro	Unid.	Error calculado	Unid.
1	2,00	mts.	1,95	mts.	2,50	%
2	2,50	mts.	2,40	mts.	4,00	%
3	3,00	mts.	2,98	mts.	0,67	%
4	3,50	mts.	3,55	mts.	-1,43	%
5	4,00	mts.	3,94	mts.	1,50	%
6	4,50	mts.	4,49	mts.	0,22	%
7	5,00	mts.	4,98	mts.	0,40	%
SE REALIZO UNA CORRECCION Y LOS VALORES SE ESTABILIZARON						
9	2,00	mts.	1,99	mts.	0,50	%
10	2,50	mts.	2,51	mts.	-0,40	%
11	3,00	mts.	2,99	mts.	0,33	%
12	3,50	mts.	3,51	mts.	-0,29	%
13	4,00	mts.	3,99	mts.	0,25	%
14	4,50	mts.	4,49	mts.	0,22	%
15	5,00	mts.	5,01	mts.	-0,20	%

Tabla 3.1.- Valores reales obtenidos por el Odómetro.

Con estos resultados obtenidos en la realidad se tenía que tomar ajustes en la práctica, por lo que a continuación se describirá el análisis y los respectivos ajustes realizados.

3.3.- Análisis de resultados.- debido a la toma de solo algunas de las perforaciones de la rueda plástica usada, no se tenía una buena aproximación de la distancia recorrida, esto se debía a que los opto acopladores no estaban en el lugar adecuado, además las perforaciones de la rueda tenía residuos del material plástico, por lo que se tuvo que sacar estos residuos y verificar de uno en uno los huecos que estén limpios de cualquier impedimento que obstruya el paso de la luz.

Además es importante que no exista roce entre los opto acopladores y la rueda plástica perforada, estos deben estar separados una distancia prudente.

Con esto se disminuyo a cero las perdidas debidas a la no captación de las perforaciones realizadas, pues ayudado por una señal realizada en la rueda,

esta media todas las perforaciones, y la velocidad que se utilizaba no dependía pues podríamos realizarla rápida o lenta, siempre se obtenía la misma cantidad de perforaciones censadas por los opto acopladores.

Para una mejor medida, se tomo como una constante probada después de varios análisis realizados el número 44.5 que se implemento en el programa para el cálculo de número de perforaciones medidas a metros de la distancia recorrida por el odómetro digital.

Se puede decir que entre mayor es la distancia medida el error de la medición disminuye, además el valor no es exacto debido al la manipulación por parte del usuario quien toma solo una referencia, a pesar de haber colocado una marca en la rueda para detenerse en el lugar preciso, al momento de la transformación a digital existe una pequeña variación en la medida.

CONCLUSIONES

- Toda medida realizada por un aparato mecánico se la puede digitalizar mediante un dispositivo electrónico adecuado, adaptado a este y valiéndose siempre de un circuito electrónico.
- La velocidad con la cual se toma la lectura de las perforaciones censadas depende siempre de la velocidad con la que trabaje el PIC, en otras palabras dependerá mucho del cristal de oscilación que se coloque en el microcontrolador.
- La señal de luz transformada a pulso por el opto acoplador es muy baja por lo que es imprescindible, realizar un circuito de amplificación de señal para que ingrese esta señal al PIC.
- Siempre es necesario basarse en las características técnicas de los elementos usados, pues existen recomendaciones que se tienen que tomar en cuenta al momento de implementar el circuito electrónico para la conexión del mismo.
- A pesar que los componentes electrónicos del proyecto son de 5 voltios, es mejor utilizar una batería de 9 voltios, pues es muy fácil de encontrarla en el mercado y por otra parte tiene una corriente adecuada para el circuito y su tamaño no es exagerado.
- Antes de escoger un microcontrolador, es mejor saber cual es el uso que lo vamos a dar, y su alcance de memoria interna para la programación, de lo contrario estaríamos usando un PIC inadecuado, para este caso específico el PIC16F876A, sirvió para realizar las operaciones de suma y multiplicación y guardada de datos medidos.
- El operador del Odómetro tiene la movilidad suficiente para la toma de lecturas efectuadas.

RECOMENDACIONES

- Es mejor usar un trinche de aluminio pues el peso es inferior al de tubo de hierro, por tal razón se hace más manejable el uso del aparato de medición.
- Es recomendable usar una rueda de plástico que tenga por perímetro un metro de longitud, pues los cálculos para las perforaciones serán mucho más sencillas, si se realizan “n” huecos estos serán directamente proporcionales a la medida de longitud.
- Se debe tomar en cuenta siempre la observación física de las piezas mecánicas, pues en el proyecto por no observar que las perforaciones realizadas en la rueda plástica estén limpias de cualquier residuo, se tuvo problemas con la calibración del odómetro.
- Siempre es necesario tener un punto de referencia para la calibración de cualquier aparato de medición para que este sea ajustado lo más preciso posible.
- Se podría en un futuro montar la parte electrónica en un dispositivo lo suficientemente estable para prescindir del direccionamiento por parte del operador del odómetro.

BIBLIOGRAFÍA:

- Ángulo, José M. 2002. PIC Diseño práctico de aplicaciones. Segunda edición.
- Ángulo, José M. 2002. PIC Diseño práctico de aplicaciones. Tercera edición.
- <http://electronred.iespana.es/optoacopla.htm>
- <http://www.comunidad.witsuk.com/node/115>
- <http://www.microbotica.es/web/downl/docs/display.pdf>
- <http://www.mikroelektronika.con.yu/spanish/product/tools/easypic2.htm>
- <http://www.x-robotics.com/rutinas.htm#Teclado%20Matricial%204x4>
- <http://www.elalejandre.net/Arquitecture.ArquitectureP.htm>
- <http://www.mikroelektronika.con.yu/english/product/books/PICbook/Uvod.htm>
- <http://www.microchip.com/downloads/en/DeviceDoc/51331a.pdf>

ANEXOS

ANEXO N° 1

ODÓMETRO DIGITAL



MANUAL DEL USUARIO

Descripción de las Teclas del Controlador Digital del Odómetro .

Control de Encendido ON/OFF _____



Control de Luminosidad del LCD _____



Led de encendido _____



Led's de control _____



Tecla 1 (Guarda Dato 1) _____



Tecla 4 (Guarda Dato 2) _____



Tecla 7 (Guarda Dato 3) _____



Tecla 2 (Muestra Dato 1) _____



Tecla 5 (Muestra Dato 2) _____



Tecla 8 (Muestra Dato 3) _____



Tecla 0 (Resetea Controlador) _____



Tecla 3 (Calcula AREA entre Dato 1 y 2) _____



Tecla 6 (Calcula VOLUMEN entre Dato 1, 2 y 3) _____



Tecla 9 (Calcula SUMA entre Dato 1 y 2) _____



Tecla # (Calcula SUMA entre Dato 1, 2 y 3) _____



Tecla A (Puesta en marcha y parada del contador.) _____



Display de Cristal Líquido (LCD) _____



Visualizar el Dato Guardado en la memoria del Odómetro.- al igual que para el almacenamiento de la información, se ha dispuesto de las teclas 2, 5 y 8 para traer al LCD el valor guardado en la memoria del Odómetro, solo presionando cualquiera de estas teclas se podrá visualizar lo siguiente: “Dato X” (X puede ser 1, 2 ó 3), y a continuación el valor medido en metros.

Calculo del Área.- para el calculo de el área de un terreno x medido, será factible hacerlo siempre y cuando este sea rectangular, así se tomara las medidas de dos de sus lados, almacenando un dato a la vez en la tecla 1 y el otro en la tecla 4, para de esta forma al presionar la tecla 3, se visualizara en el LCD “Medicion : AREA” y en la segunda línea se desplegará el valor en metros cuadrados.

Calculo del Volumen.- para el calculo de el volumen, será procederá de manera similar a la parte anterior, con la diferencia que esta vez también tenemos que realizar el ingreso de un tercer dato, así al presionar la tecla 6, se visualizara en el LCD “Medicion: VOLUMEN” y en la segunda línea se desplegará el valor en metros cúbicos.

Reseteo de valores almacenados.- el odómetro tiene la tecla 0, para efectuar el reseteo total de las mediciones efectuadas para dejar sin información al mismo.

Al pulsar esta tecla se dejara listo para realizar mediciones desde cero sin nada de información almacenada, es decir esta listo para empezar.

ANEXO N° 2

ANEXO 2: Resumen del PIC16F876/877

AN2.1 INTRODUCCION

El presente anexo está destinado a hacer un breve resumen de las características generales del dispositivo PIC16F876, el que viene de serie con el laboratorio PIC'School, así como del PIC16F877, idéntico al anterior pero con más líneas de E/S. Para mas información se recomienda acudir a la página www.microchip.com donde podemos encontrar los data sheets completos con todos los detalles técnicos de todos los modelos de PIC, notas de aplicación, actualizaciones de las diferentes herramientas software, etc.

AN2.2 CARACTERISTICAS GENERALES

A continuación se detallan las características más relevantes que definen la CPU y periféricos que incluye el dispositivo PIC16F876/F877

- CPU con arquitectura RISC (juego de instrucciones reducido)
- Juego de 35 instrucciones de un único word de 14 bits cada una de ellas.
- Todas las instrucciones se ejecutan en un único ciclo de instrucción excepto los saltos que necesitan dos. Un ciclo de instrucción necesita a su vez de cuatro ciclos de reloj. Trabajando a 4MHz una instrucción se ejecuta en 1µS.
- Memoria FLASH de programa de 8K word de 14 bits, 368 bytes de memoria RAM para datos variables y 256 bytes de memoria EEPROM para datos no volátiles.
- Capacidad de interrupción que puede ser provocada por 14 fuentes o eventos distintos.
- Stack autónomo y transparente de 8 niveles
- Modos de direccionamiento directo, indirecto y relativo.
- Circuito Power-On Reset (POR) integrado. Automáticamente genera la señal de arranque cuando la tensión de alimentación alcanza un determinado valor.
- Temporizadores Power-up Timer (PWRT) y Oscillator Start-up Timer (OST) que retrasan el arranque, en espera de la total estabilización tanto de la tensión de alimentación como de la frecuencia de trabajo del oscilador.
- Temporizador WDT o "Perro guardián" capaz de provocar un arranque automático en determinadas ocasiones.
- Modo de trabajo en standby que reduce el consumo así como velocidades del oscilador seleccionables.
- Tensión de alimentación de 2 a 5.5V con un bajo consumo: < 2 mA a 5V y 4MHz; 20µA a 3V y 32KHz; < 1µA en el modo de espera o stanby.
- Corriente de salida de hasta 25 mA por pin
- Temporizador/contador de 8 bits con preescaler (TMR0)
- Temporizador/contador de 16 bits con preescaler (TMR1)
- Temporizador/contador de 8 bits con registro de periodos, prescaler y postcaler (TMR2)
- Dos módulos CCP independientes para la captura, comparación y modulación de anchura de pulsos (PWM).
- Convertidor analógico/digital con resolución de 10 bits y hasta 8 canales de entrada.
- Puerta serie síncrona (SSP) con modalidad de bus SPI (modo master) y bus I²C (Master/Slave).
- Receptor/Transmisor universal síncrono/asíncrono (USART).
- Puerto paralelo esclavo (PSP) de 8 bits para transferencias en paralelo con señales de control del tipo read (RD), write (WR) y selección (CS). Sólo está disponible en el modelo PIC16F877.

AN2.3 DESCRIPCION DE PINES

La figura AN2-1 presenta la distribución de patillas para el modelo PIC16F876 de 28 pines y el PIC16F877 de 40. A continuación se presenta una tabla que resume brevemente la finalidad de cada una de ellas.

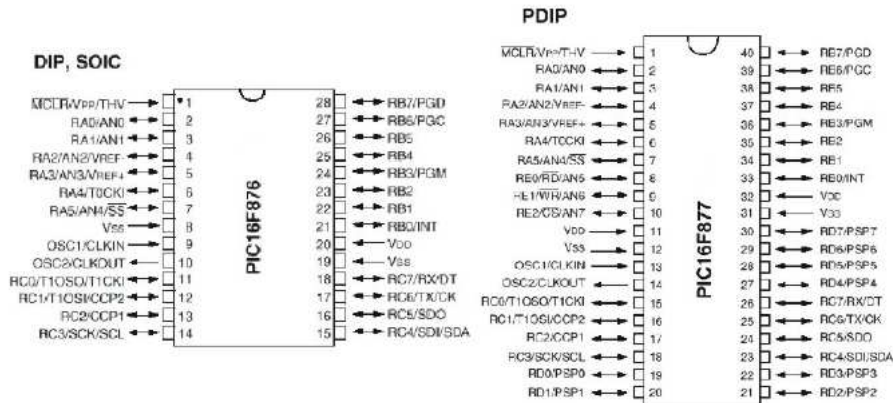


Figura AN2-1. Diagrama de conexiones del PIC16F876/877

NOMBRE	TIPO	DESCRIPCION
OSC1/CLKIN	I	Entrada para el cristal del oscilador/entrada de oscilador externo.
OSC2/CLKOUT	O	Salida del cristal del oscilador. En el modo RC por este pin sale 1/4 de la frecuencia de OSC1 y denota la ejecución de un ciclo de instrucción
MCLR/Vpp/THV	I/P	Entrada de reset activa por nivel bajo. Por esta patilla también se introduce la tensión Vpp de grabación y la tensión de test THV
		PUERTA A: 6 líneas
RA0/AN0	I/O	Entrada/salida digital RA0 o entrada analógica del canal 0 AN0
RA1/AN1	I/O	Entrada/salida digital RA1 o entrada analógica del canal 1 AN1
RA2/AN2/VREF-	I/O	Entrada/salida digital RA2, entrada analógica AN2 o entrada - de tensión de referencia VREF-
RA3/AN3/VREF+	I/O	Entrada/salida digital RA3, entrada analógica AN3 o entrada + de tensión de referencia VREF+
RA4/T0CKI	I/O	Entrada/salida digital en colector abierto RA4. Entrada de reloj para el TMR0 en modo contador
RA5/SS/AN4	I/O	Entrada/salida digital RA5 o entrada analógica AN4. También actúa como señal SS de selección de esclavo para el módulo SSP.

NOMBRE	TIPO	DESCRIPCION
PUERTA B: 8 líneas		
RB0/INT	I/O	Entrada/salida digital RB0 o entrada de interrupción externa INT
RB1	I/O	Entrada/salida digital
RB2	I/O	Entrada/salida digital RB2
RB3/PGM	I/O	Entrada/salida digital RB3. Entrada en baja tensión de programación PGM
RB4	I/O	Entrada/salida digital RB4. Produce interrupción por cambio de estado
RB5	I/O	Entrada/salida digital RB5. Produce interrupción por cambio de estado
RB6	I/O	Entrada/salida digital RB6. Produce interrupción por cambio de estado
RB7	I/O	Entrada/salida digital RB7. Produce interrupción por cambio de estado
PUERTA C: 8 líneas		
RC0/T1OSO/T1CKI	I/O	Entrada/salida digital RC0, salida T1OSO de oscilador del TMR1 o entrada T1CKI de reloj para el TMR1
RC1/T1OSI/CCP2	I/O	Entrada/salida digital RC1, entrada de oscilador del TMR1 o E/S del módulo CCP2
RC2/CCP1	I/O	Entrada/salida digital RC2 o E/S del módulo CCP1
RC3/SCK/SCL	I/O	Entrada/salida digital RC3 o E/S de reloj para los modos SPI (SCK) e I ² C (SCL) de la puerta serie síncrona (SSP)
RC4/SDI/SDA	I/O	Entrada/salida digital RC4, entrada SDI de datos para el bus SPI o E/S SDA de datos para el bus I ² C.
RC5/SDO	I/O	Entrada/salida digital RC5 o salida SDO de datos para el bus SPI
RC6/TX/CK	I/O	Entrada/salida digital RC6, transmisor Tx del USART en modo asíncrono o reloj del USART en modo síncrono
RC7/RX/DT	I/O	Entrada/salida digital RC7, recepción Rx del USART en modo asíncrono o línea de datos del USAR en modo síncrono

ANEXO N° 3

```

program Medidor2

dim flag as bYTE
dim flag1 as bYTE
dim flag2 as bYTE
dim flag3 as bYTE
dim flag4 as bYTE
dim Tecla as byte
dim Dato as byte
dim Cont as longword
dim Cont1 as float
dim Dato1 as float
dim Dato2 as float
dim Dato3 as float
dim Resultado as float
dim txt as string[17]
dim txt1 as string[11]

sub procedure interrupt      'Interrupcion RB0
    INTCON.GIE = 0          'Deshabilita interrupcion Global
    flag4=1
    INTCON.INTF = 0        'Limpia flag de interrupcion RB0
    INTCON.GIE = 1        'Habilita global interrupciones
end sub

sub procedure Inicializa
    delay_ms(500)          'Startup LCD
    Lcd_Config(PORTC,7,6,5,4,PORTB,5,6,7) 'Configurar pines LCD
    Lcd_Cmd(Lcd_Cursor_Off) 'Cursor OFF
    Lcd_Cmd(Lcd_Clear)      'Limpia la Pantalla
    Lcd_Out(1, 6, "Medidor")
    Lcd_Out(2, 3, "de Distancias")
end sub

sub procedure revisar
    if (TestBit(PORTB,2) and (flag4=1)) then
        FLAG4=0
        cont=cont-1
    end if
    if (TestBit(PORTB,1) and (flag4=1)) then
        FLAG4=0
        cont=cont+1
    end if
' end if
end sub

sub procedure caso3
    if((Dato1 >= 10000.0) or (Dato2>=10000.0))then
        Lcd_Cmd(Lcd_Clear)
        Lcd_Out(1, 1, "Overflow")
    end if
end sub

```

```

    Lcd_Out(2, 1, "Presione A Medir")
else
    Resultado=Dato1*Dato2
    Lcd_Cmd(Lcd_Clear)
    Lcd_Out(1, 1, "Medicion: AREA")
    FloatToStr(Resultado, txt)
    Lcd_Out(2, 1, txt)
    Lcd_Out(2, 13, "mts2")
end if
end sub

sub procedure caso6
if((Dato1 >= 1000.0) or (Dato2>=1000.0) or (Dato3>=1000.0 ))then
    Lcd_Cmd(Lcd_Clear)
    Lcd_Out(1, 1, "Overflow")
    Lcd_Out(2, 1, "Presione A Medir")
else
    Resultado=Dato1*Dato2*Dato3
    Lcd_Cmd(Lcd_Clear)
    Lcd_Out(1, 1, "Medicion:VOLUMEN")
    FloatToStr(Resultado, txt)
    Lcd_Out(2, 1, txt)
    Lcd_Out(2, 13, "mts3")
end if
end sub

main:
TRISA = %00111100      'Columnas del TECLADO
TRISB = %00000111      'Entradas de los Optoacopladores y Control LCD
TRISC = %00000000      'Lcd Datos y filas del Teclado
ADCON1 = %00000111      'Pines del PORTA como digitales

INTCON.GIE = 0          'Deshabilita interrupcion Global
INTCON.INTF = 0          'Limpia flag de interrupcion RB0
INTCON.INTE = 1          'Habilita interrupcion RB0
OPTION_REG.INTEDG=0      'Interrupcion en flanco de Bajada
'Inicializa Variables
Dato1=0.0
Dato2=0.0
Dato3=0.0
Resultado=0.0
flag1=1
flag2=1
flag4=0
Dato=0
Cont=0
txt=""
'Inicializa
INICIO:
    Tecla=255

```

```

GOSUB Barrido
GOSUB Rebote
select case Tecla
case 1
    Dato1=Resultado
    Lcd_Cmd(Lcd_Clear)
    Lcd_Out(1, 1, "Grabado DATO1")
    Lcd_Out(2, 1, "Presione A Medir")
case 4
    Dato2=Resultado
    Lcd_Cmd(Lcd_Clear)
    Lcd_Out(1, 1, "Grabado DATO2")
    Lcd_Out(2, 1, "Presione A Medir")
case 7
    Dato3=Resultado
    Lcd_Cmd(Lcd_Clear)
    Lcd_Out(1, 1, "Grabado DATO3")
    Lcd_Out(2, 1, "Presione A Medir")
case 3
    caso3
case 6
    caso6
case 2
    Lcd_Cmd(Lcd_Clear)
    Lcd_Out(1, 1, "Dato 1")
    FloatToStr(Dato1, txt)
    Lcd_Out(2, 1, txt)
    Lcd_Out(2, 13, "mts.")
case 5
    Lcd_Cmd(Lcd_Clear)
    Lcd_Out(1, 1, "Dato 2")
    FloatToStr(Dato2, txt)
    Lcd_Out(2, 1, txt)
    Lcd_Out(2, 13, "mts.")
case 8
    Lcd_Cmd(Lcd_Clear)
    Lcd_Out(1, 1, "Dato 3")
    FloatToStr(Dato3, txt)
    Lcd_Out(2, 1, txt)
    Lcd_Out(2, 13, "mts.")
case 9
    Resultado=Dato1+Dato2
    Lcd_Cmd(Lcd_Clear)
    Lcd_Out(1, 1, "Medicion:LINEAL")
    FloatToStr(Resultado, txt)
    Lcd_Out(2, 1, txt)
    Lcd_Out(2, 13, "mts.")
case 15
    Resultado=Dato1+Dato2+Dato3
    Lcd_Cmd(Lcd_Clear)

```



```

Lcd_Out(1, 1, "Medicion:LINEAL")
FloatToStr(Resultado, txt)
Lcd_Out(2, 1, txt)
Lcd_Out(2, 13, "mts.")
case 0
  Dato1=0.0
  Dato2=0.0
  Dato3=0.0
  Resultado=0.0
case 10
  flag1=1
  flag2=1
  flag3=1
  while flag2=1
    if flag3=1 then
      flag3=0
      Tecla=10
    else
      Tecla=255
    end if
    revisar
    GOSUB Barrido1
    GOSUB Rebote1
    SELECT CASE Tecla
    case 10
      if flag1=1 then
        flag1=0
      else
        flag1=1
      end if
      IF FLAG1=0 THEN
        Cont=0 '*****
        Lcd_Cmd(Lcd_Clear)
        Lcd_Out(1, 1, "MIDIENDO.....")
        flag4=0
        INTCON.GIE = 1 'Habilita interrupcion Global
      ELSE
        INTCON.GIE = 0 'Deshabilita interrupcion Global
        Lcd_Cmd(Lcd_Clear)
        'ByteToStr(dato,txt2)
        LongintToStr(cont, txt1)
        Lcd_Out(1,6,txt1)
        'Lcd_Out(1,12,txt2)
        Lcd_Out(1, 1, "Medicion: ")
        Cont1=(cont*1.11)/44.5
        Resultado=cont1
        FloatToStr(cont1, txt)
        Lcd_Out(2, 1, txt)
        Lcd_Out(2, 13, "mts.")
        flag2=0

```

```

        END IF
    end select
wend
end select
goto INICIO
BARRIDO1:
    ClearBit(PORTC,0)    'sensar la fila A
    flag = TestBit(PORTA,5)
    IF flag = 0 THEN
        Tecla=10
    '
    ' else
    '     Tecla=255
    return
    END IF 'tecla pulsada retorne con variable cargada con 10
    SetBit(PORTC,0)
return
Rebote1:
    Suelta1:            'Subrutina de antirrebote de teclas
    flag = TestBit(PORTA,5)
    IF flag = 0 THEN    'si la tecla sigue pulsada ir espacio
        goto Suelta1    'si la tecla sigue pulsada ir espacio
    end if
RETURN

BARRIDO:
    ClearBit(PORTC,0)    'sensar la fila A
    flag = TestBit(PORTA,2)
    IF flag = 0 THEN
        Tecla=1
    return
    END IF 'tecla pulsada retorne con variable cargada con 1
    flag = TestBit(PORTA,3)
    IF flag = 0 THEN
        Tecla=2
    return
    END IF 'tecla pulsada retorne con variable cargada con 2
    flag = TestBit(PORTA,4)
    IF flag = 0 THEN
        Tecla=3
    return
    END IF 'tecla pulsada retorne con variable cargada con 3
    flag = TestBit(PORTA,5)
    IF flag = 0 THEN
        Tecla=10
    return
    END IF 'tecla pulsada retorne con variable cargada con 10
    SetBit(PORTC,0)
    ClearBit(PORTC,1)    'sensar la fila B
    flag = TestBit(PORTA,2)
    IF flag = 0 THEN

```

```

    Tecla=4
return
END IF 'tecla pulsada retorne con variable cargada con 4
flag = TestBit(PORTA,3)
IF flag = 0 THEN
    Tecla=5
return
END IF 'tecla pulsada retorne con variable cargada con 5
flag = TestBit(PORTA,4)
IF flag = 0 THEN
    Tecla=6
return
END IF 'tecla pulsada retorne con variable cargada con 6
flag = TestBit(PORTA,5)
IF flag = 0 THEN
    Tecla=11
return
END IF 'tecla pulsada retorne con variable cargada con 11
SetBit(PORTC,1)
ClearBit(PORTC,2)      'sensar la fila C
flag = TestBit(PORTA,2)
IF flag = 0 THEN
    Tecla=7
return
END IF 'tecla pulsada retorne con variable cargada con 7
flag = TestBit(PORTA,3)
IF flag = 0 THEN
    Tecla=8
return
END IF 'tecla pulsada retorne con variable cargada con 8
flag = TestBit(PORTA,4)
IF flag = 0 THEN
    Tecla=9
return
END IF 'tecla pulsada retorne con variable cargada con 9
flag = TestBit(PORTA,5)
IF flag = 0 THEN
    Tecla=12
return
END IF 'tecla pulsada retorne con variable cargada con 12
SetBit(PORTC,2)
ClearBit(PORTC,3)      'sensar la fila D
flag = TestBit(PORTA,2)
IF flag = 0 THEN
    Tecla=14
return
END IF 'tecla pulsada retorne con variable cargada con 14
flag = TestBit(PORTA,3)
IF flag = 0 THEN
    Tecla=0

```

```

return
END IF 'tecla pulsada retorne con variable cargada con 0
flag = TestBit(PORTA,4)
IF flag = 0 THEN
    Tecla=15
return
END IF 'tecla pulsada retorne con variable cargada con 15
flag = TestBit(PORTA,5)
IF flag = 0 THEN
    Tecla=13
return
END IF 'tecla pulsada retorne con variable cargada con 13
SetBit(PORTC,3)
return

```

Rebote:

```

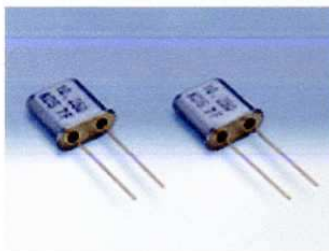
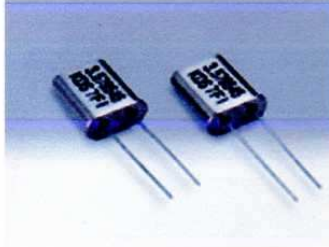
Suelta:                'Subrutina de antirrebote de teclas
flag = TestBit(PORTA,2)
IF flag = 0 THEN
    goto Suelta        'si la tecla sigue pulsada ir espacio
end if
flag = TestBit(PORTA,3)
IF flag = 0 THEN      'si la tecla sigue pulsada ir espacio
    goto Suelta        'si la tecla sigue pulsada ir espacio
end if
flag = TestBit(PORTA,4)
IF flag = 0 THEN      'si la tecla sigue pulsada ir espacio
    goto Suelta        'si la tecla sigue pulsada ir espacio
end if
flag = TestBit(PORTA,5)
IF flag = 0 THEN      'si la tecla sigue pulsada ir espacio
    goto Suelta        'si la tecla sigue pulsada ir espacio
end if
RETURN
end.

```

ANEXO N° 4

Crystal Resonators / MHz Band Crystal Resonators

UM-1, UM-4, UM-5, HC-49/T, HC-49/U, HC-50/T, HC-50/U



The UM series of resonators offers excellent frequency stability and are ideal for a wide range of mobile radio communications. The designs offer excellent shock resistance and reliability, and despite the small size there is no trade off with the performance. In addition, the HC series is designed for use in microprocessors and other standard clocks, offering excellent frequency stability.



■ Features

- Resonator with a high frequency stability ideal for use in mobile radio communications.
- High precision and high reliability
- Taped and reeled allowing for automatic surface mounting. (HC-49/T, HC-49/U)

■ Series Resistance

Frequency	Type	Overtone Order	UM-4 Ω max.	UM-5 Ω max.	UM-1 Ω max.	HC-49/T Ω max.	HC-50/T Ω max.	HC-49/U Ω max.	HC-50/U Ω max.
2.4~ 3.0MHz	F	F	-	-	-	-	-	350	-
3.0~ 3.5MHz	F	F	-	-	-	-	-	150	-
3.5~ 4.0MHz	F	F	-	-	-	150	150	90	90
4.0~ 7.0MHz	F	F	-	-	-	90	90	60	60
7.0~ 10MHz	F	F	-	-	-	45	45	35	35
10~ 15MHz	F	F	50	50	50	45	45	35	35
15~ 20MHz	F	F	50	50	50	25	25	25	25
20~ 25MHz	F/3	F/3	50/-	50/-	50/-	25/60	25/60	25/60	25/60
25~ 30MHz	F/3	F/3	50/-	50/-	50/-	25/40	25/40	25/40	25/40
30~ 75MHz	3	3	70	70	70	40	40	40	40
75~100MHz	3/5	3/5	70/-	70/80	70/80	-/60	-/60	-/60	-/60
100~125MHz	5	5	80	80	80	60	60	60	60
125~150MHz	5	5	100	100	100	-	-	-	-
150~200MHz	7	7	-	120	120	-	-	-	-

Consult our sales representative for other specifications.

F: fundamental 3: 3rd overtone 5: 5th overtone 7: 7th overtone

Refer to page 39 for standard specification.

■ Frequency Range

2.4~200MHz

■ Overtone Order

Fundamental, 3rd overtone, 5th overtone, 7th overtone

■ Drive Level

10μW, 50μW, 100μW, 500μW

■ Load Capacitance

Series, 12pF, 16pF, 20pF, 32pF (fundamental)
8pF, 10pF, 12pF, 16pF (3rd, 5th, 7th overtone)

■ Frequency Tolerance (at 25°C)

±5×10⁻⁶, ±10×10⁻⁶, ±15×10⁻⁶, ±20×10⁻⁶, ±30×10⁻⁶

■ Frequency Characteristics over Temperature (Ref. to 25°C)

±5×10⁻⁶, ±10×10⁻⁶, ±20×10⁻⁶, ±30×10⁻⁶, ±50×10⁻⁶

■ Operating Temperature Range

-10°C~+60°C

■ Storage Temperature Range

-30°C~+80°C

■ Aging Characteristics

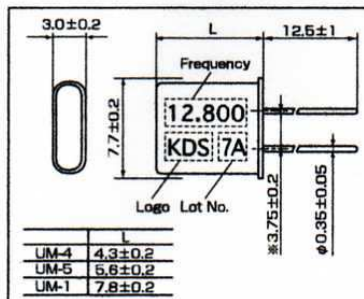
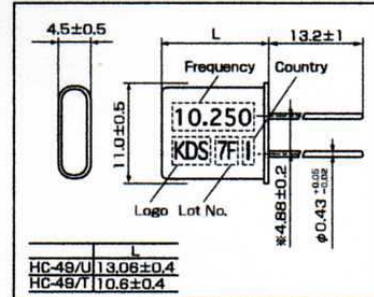
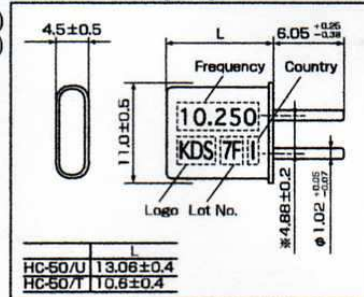
UM Series:

±1×10⁻⁶, ±2×10⁻⁶, ±3×10⁻⁶/year (max.)

HC Series:

±5×10⁻⁶/year (max.)

■ Dimensions [mm]



※Measurement between the root of the leads.

ANEXO N° 5

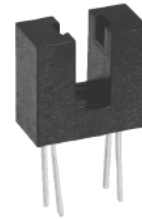
Slotted Optical Switch



OPB804

Features

- Non-contact switching
- Housing opaque material
- Printed circuit board mount
- 0.155" (3.94 mm) Width Gap
- 0.330" (8.38 mm) Depth Slot



Description

The OPB804 contains an IRLED and phototransistor paired in a plastic housing .

The housing is an opaque grade of injection molded plastic which minimizes the assembly's sensitivity to visible and near-infrared radiation. The wide open aperture makes it versatile for general applications.

The output switches when a opaque object to IR (700nm to 1100nm) is inserted into the gap, between the emitter and sensor, and interrupts the light beam.

Applications

- Non-contact object sensing
- Assembly line automation
- Machine automation
- Equipment Security
- Machine safety

Ordering Information

OPB804 Slotted Switch

Customization:

Contact OPTEK for value added and next level assembly services on this sensor.

Contact OPTEK for special electrical screening on this sensor.

Absolute Maximum Ratings:

(T_A = 25° C unless otherwise noted)

Storage Temperature Range	-40°C to +100°C
Operating Temperature Range	-40°C to +85°C
Lead Soldering Temperature ⁽¹⁾	240°C for 5 Seconds

Input IRLED

Forward DC Current	50 mA
Peak Forward Current (1 μs pulse, 300pps)	1.0A
Reverse DC Voltage	2.0V
Power Dissipation	75 mW

Output Phototransistor

Collector-Emitter Voltage	30V
Emitter-Collector Voltage	5V
Collector DC Current	30 mA
Power Dissipation	100 mW

Maximum Rating Notes:

1. With soldering iron 1/16 inch (1.6mm) from the case. Duration can be extended to 10 seconds max. when flow soldering. RMA flux is recommended.
2. All parameters measured using pulse technique.

March 2003
Issue 2.0

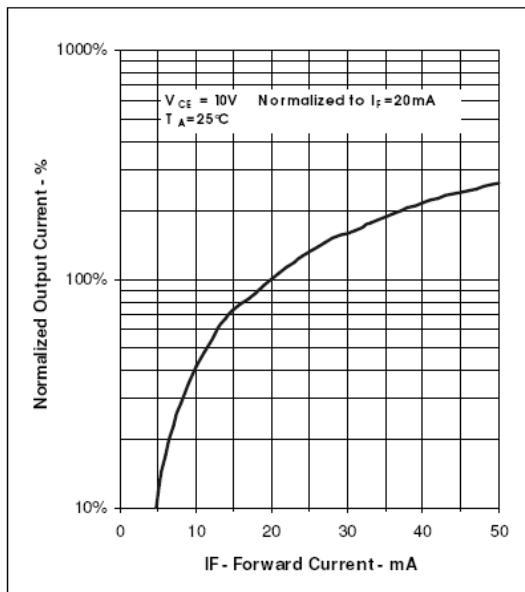
OPB804 Technical Data



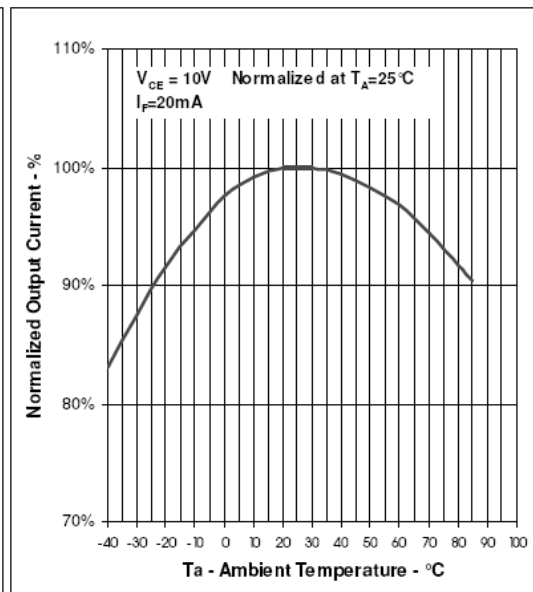
Electrical Characteristics (T_A = 25°C unless otherwise noted)

SYMBOL	PARAMETER	MIN	TYP	MAX	UNITS	TEST CONDITION
Input Diode (See OP140 for additional information, reference only)						
V _F	Forward Voltage	-	1.25	1.70	V	I _F = 20mA
I _R	Reverse Current	-	-	100	μA	V _R = 2.0V
Output Phototransistor (See OP550 for additional information, reference only)						
V _{(BR)CEO}	Collector-Emitter Breakdown Voltage	30	-	-	V	I _C = 1.0 mA, E _E = 0mw/cm ²
V _{(BR)ECO}	Emitter-Collector Breakdown Voltage	5.0	-	-	V	I _E = 100μA, E _E = 0mw/cm ²
I _{CEO}	Collector-Emitter Dark Current	-	-	100	nA	V _{CE} = 10V, I _F = 0, E _E = 0mw/cm ²
Coupled						
V _{CE(SAT)}	Collector-Emitter Saturation Voltage:	-	-	0.40	V	I _C = 250μA, I _F = 20mA
I _{C(ON)}	On-State Collector Current:	0.50	5.00	-	mA	V _{CE} = 10.0V, I _F = 20mA

Normalized Output Current vs. Forward Current



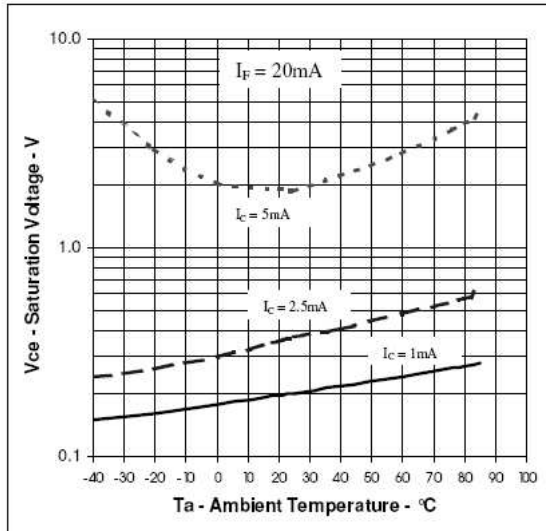
Normalized Output Current vs. Ambient Temperature



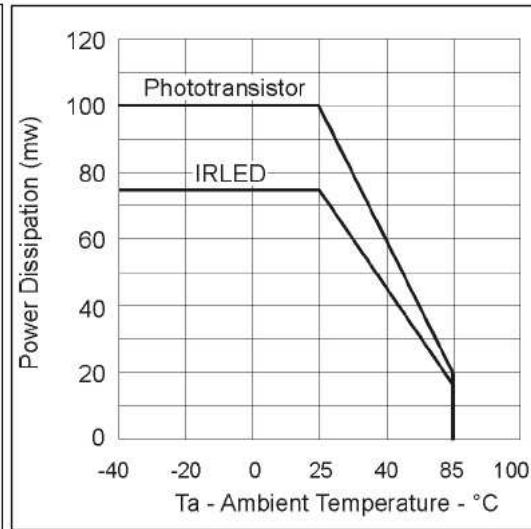
OPB804 Technical Data



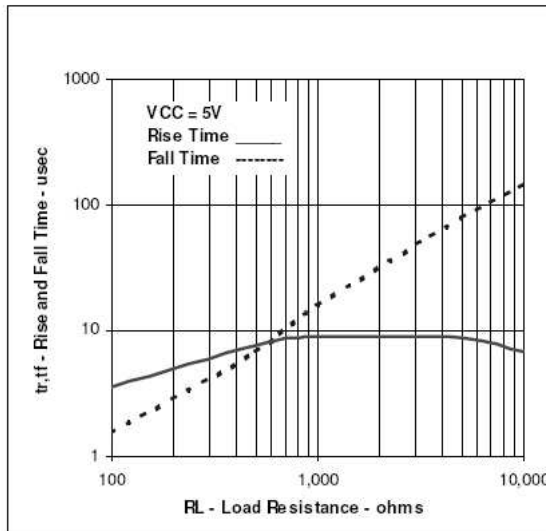
Saturation Voltage $V_{ce(sat)}$ vs. Ambient Temperature



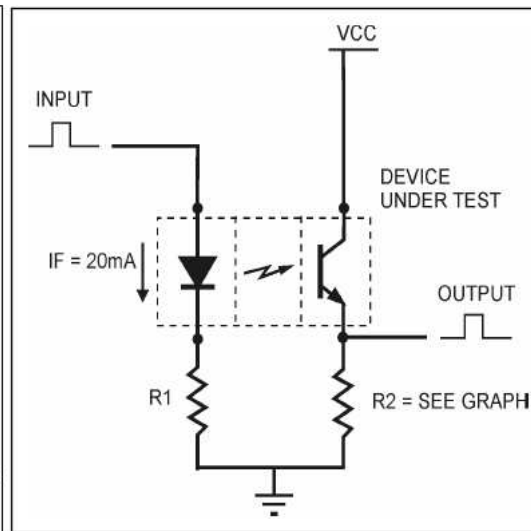
Power Dissipation Derating Curves vs. Ambient Temperature



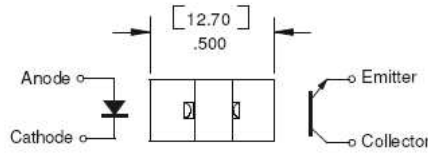
Typical Rise and Fall Time vs. Load Resistance



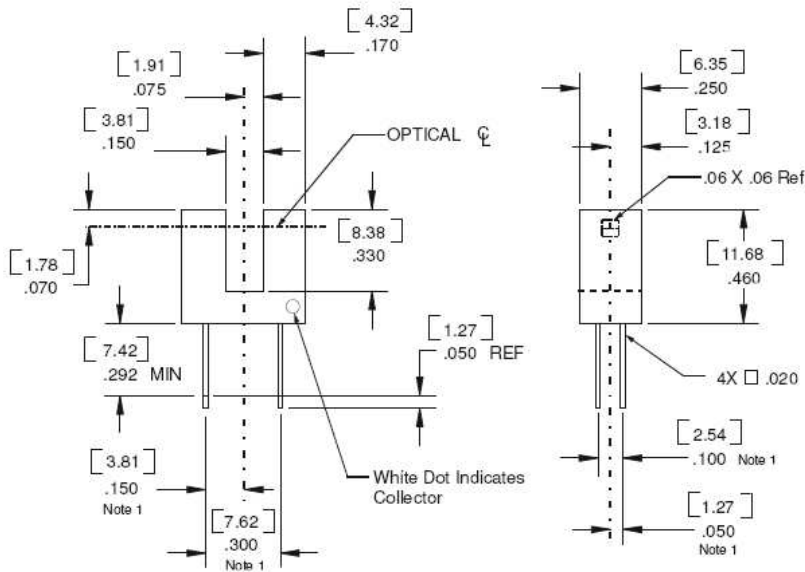
Test Schematic



OPB804 Technical Data

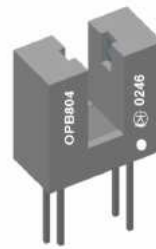
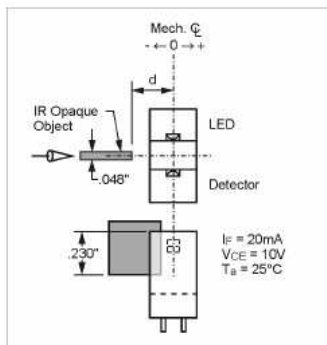
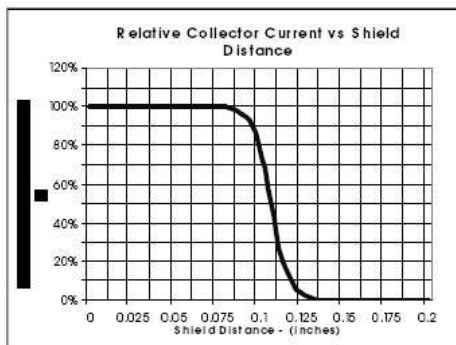


- Notes:**
1. Dimension controlled at housing surface.
 2. Dimensions are inches [mm].
 3. Tolerances $\pm .010"$ [0.25]



PRECAUTIONS:
Methanol and isopropanol alcohols are recommended as cleaning reagents. Spray or wipe do not immerse. Exposure of the plastic body to chlorinated hydrocarbons and ketones such as thread lock and instant adhesives will degrade the plastic body. Highly activated, water soluble fluxes may also attack housings in some situations. It is recommended a sample be tested.

Mechanical Object Position vs. Relative Collector Current



Optek reserves the right to make changes at any time in order to improve design and to supply the best product possible.

March 2003
Issue 2.0

Optek Technology, Inc.

1215 West Crosby Road

Carrollton, Texas 75006

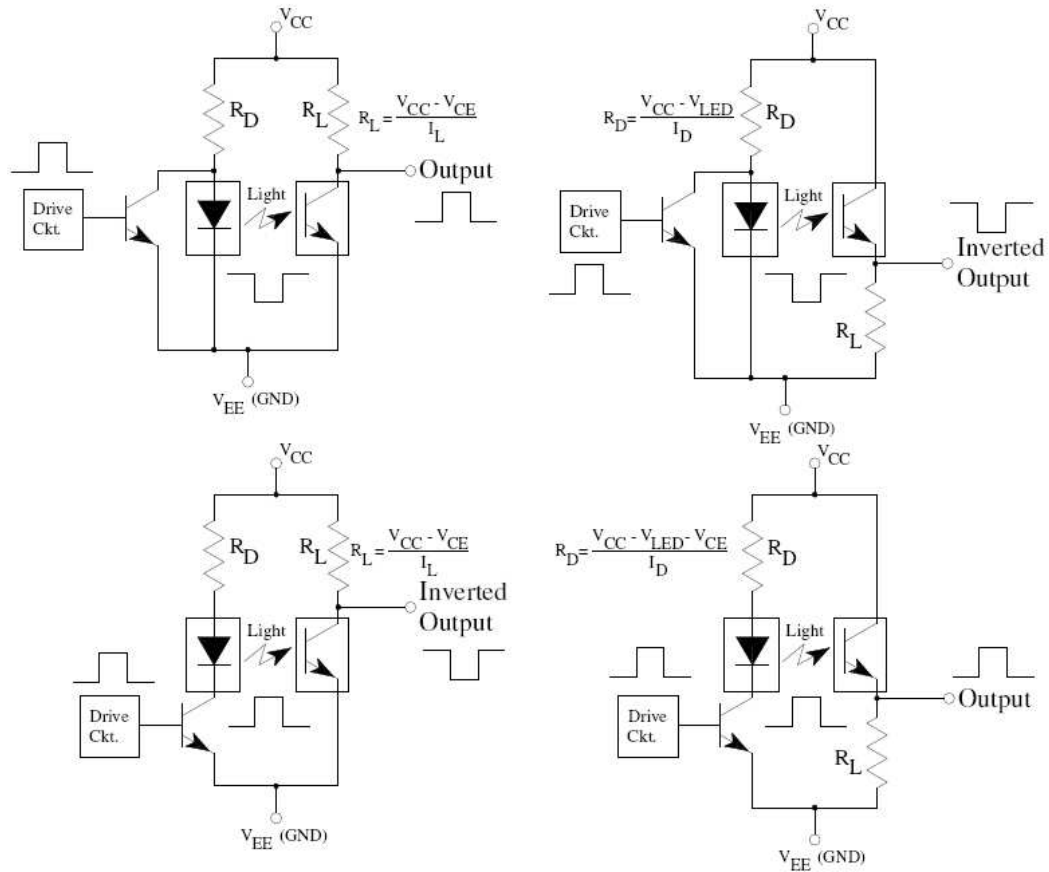
(972) 323-2200

www.optekinc.com

OPB804 Technical Data



Driver Circuits for LED & Phototransistor



ANEXO N° 6

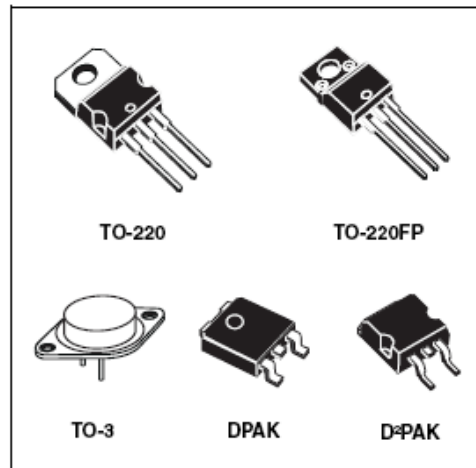
Positive voltage regulators

Features

- Output current to 1.5 A
- Output voltages of 5; 6; 8; 8.5; 9; 12; 15; 18; 24 V
- Thermal overload protection
- Short circuit protection
- Output transition SOA protection

Description

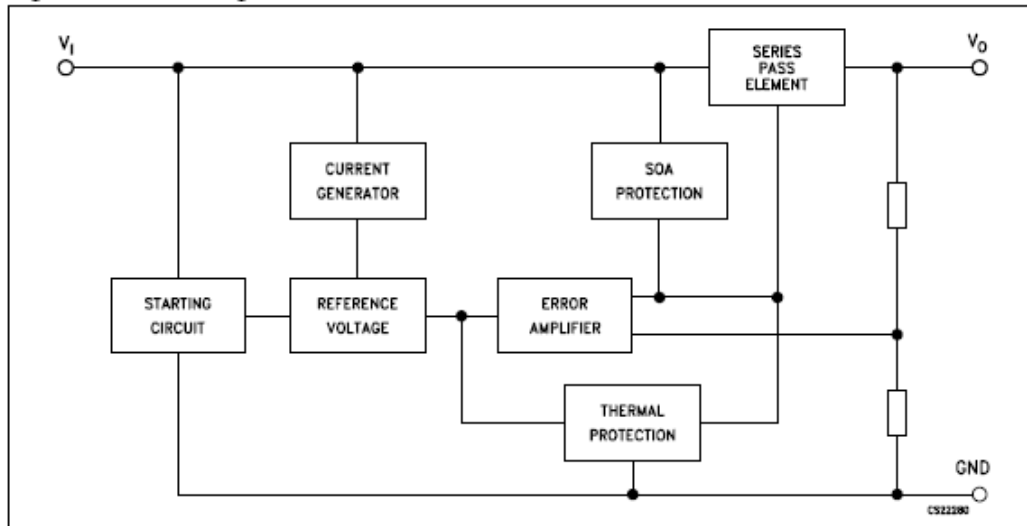
The L78xx series of three-terminal positive regulators is available in TO-220, TO-220FP, TO-3, D²PAK and DPAK packages and several fixed output voltages, making it useful in a wide range of applications. These regulators can provide local on-card regulation, eliminating the distribution problems associated with single point regulation. Each type employs internal current limiting, thermal shut-down and safe area protection, making it essentially indestructible. If adequate heat sinking is provided, they can deliver over 1 A output current. Although designed primarily as fixed voltage regulators, these devices can be used with external components to obtain adjustable voltage and currents.


Table 1. Device summary

Part numbers	
L7805	L7809C
L7805C	L7812C
L7806C	L7815C
L7808C	L7818C
L7885C	L7824C

1 Diagram

Figure 1. Block diagram



2 Pin configuration

Figure 2. Pin connections (top view)

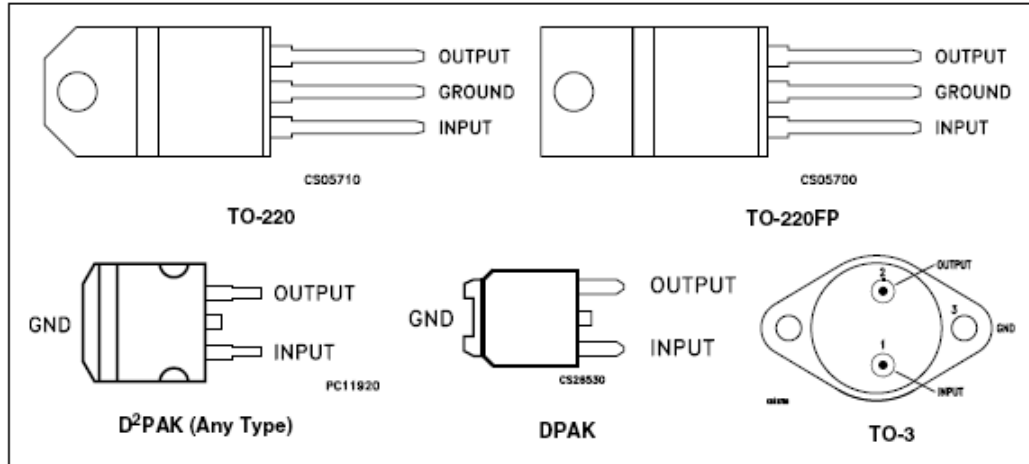
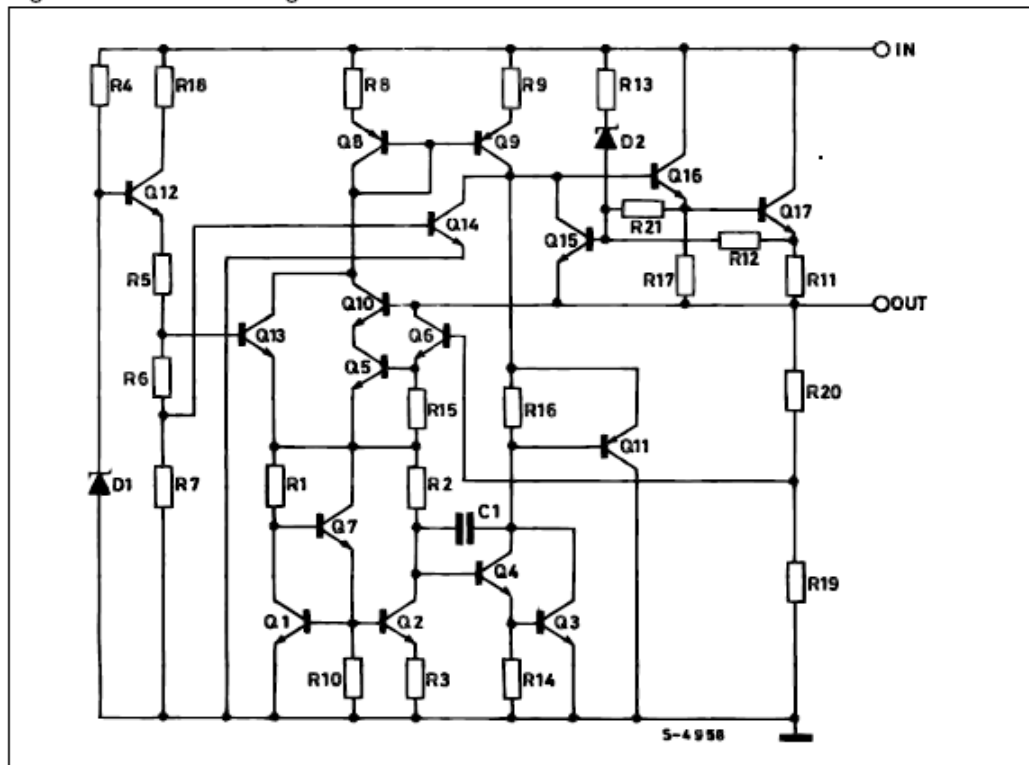


Figure 3. Schematic diagram



3 Maximum ratings

Table 2. Absolute maximum ratings

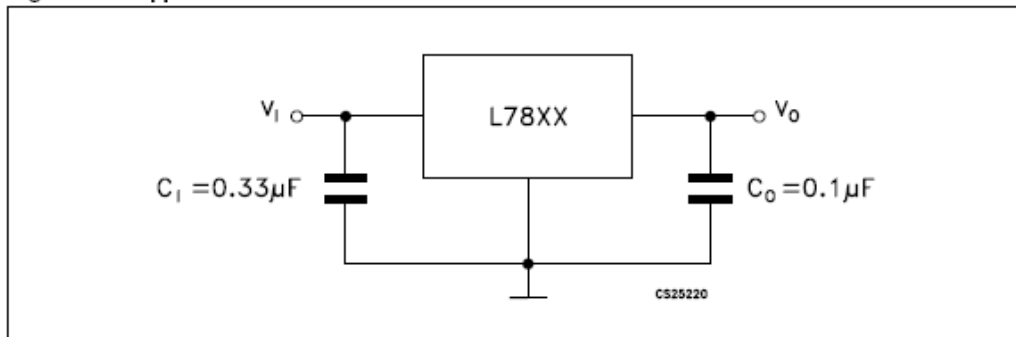
Symbol	Parameter		Value	Unit
V_I	DC input voltage	for $V_O = 5$ to 18 V	35	V
		for $V_O = 20, 24$ V	40	
I_O	Output current		Internally limited	
P_D	Power dissipation		Internally limited	
T_{STG}	Storage temperature range		-65 to 150	°C
T_{OP}	Operating junction temperature range	for L7800	-55 to 150	°C
		for L7800C	0 to 150	

Note: Absolute maximum ratings are those values beyond which damage to the device may occur. Functional operation under these condition is not implied.

Table 3. Thermal data

Symbol	Parameter	D ² PAK	DPAK	TO-220	TO-220FP	TO-3	Unit
R_{thJC}	Thermal resistance junction-case	3	8	5	5	4	°C/W
R_{thJA}	Thermal resistance junction-ambient	62.5	100	50	60	35	°C/W

Figure 4. Application circuits



4 Test circuits

Figure 5. DC parameter

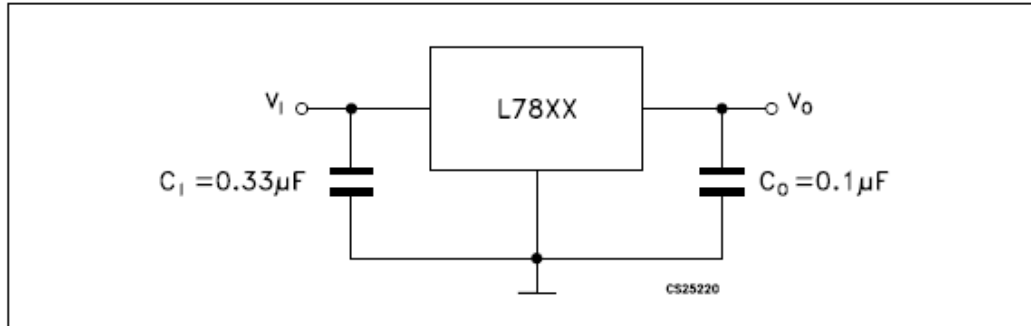


Figure 6. Load regulation

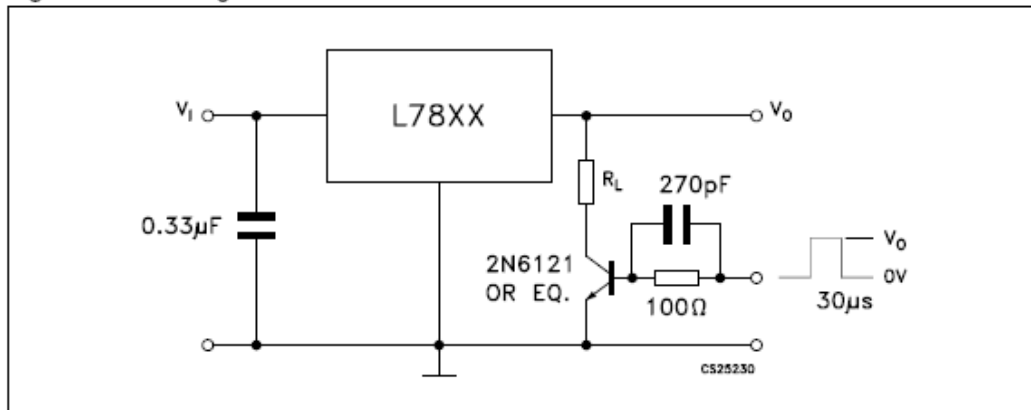
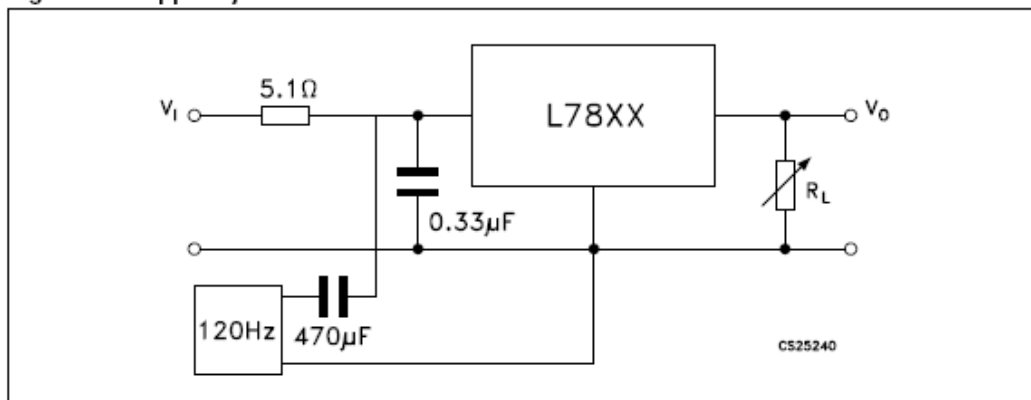


Figure 7. Ripple rejection



5 Electrical characteristics

Table 4. Electrical characteristics of L7805 (refer to the test circuits, $T_J = -55$ to 150 °C, $V_I = 10$ V, $I_O = 500$ mA, $C_I = 0.33$ μ F, $C_O = 0.1$ μ F unless otherwise specified)

Symbol	Parameter	Test conditions	Min.	Typ.	Max.	Unit
V_O	Output voltage	$T_J = 25^\circ\text{C}$	4.8	5	5.2	V
V_O	Output voltage	$I_O = 5$ mA to 1 A, $P_O \leq 15$ W $V_I = 8$ to 20 V	4.65	5	5.35	V
$\Delta V_O^{(1)}$	Line regulation	$V_I = 7$ to 25 V, $T_J = 25^\circ\text{C}$		3	50	mV
		$V_I = 8$ to 12 V, $T_J = 25^\circ\text{C}$		1	25	
$\Delta V_O^{(1)}$	Load regulation	$I_O = 5$ mA to 1.5 A, $T_J = 25^\circ\text{C}$			100	mV
		$I_O = 250$ to 750 mA, $T_J = 25^\circ\text{C}$			25	
I_d	Quiescent current	$T_J = 25^\circ\text{C}$			6	mA
ΔI_d	Quiescent current change	$I_O = 5$ mA to 1 A			0.5	mA
		$V_I = 8$ to 25 V			0.8	
$\Delta V_O/\Delta T$	Output voltage drift	$I_O = 5$ mA		0.6		mV/°C
eN	Output noise voltage	$B = 10$ Hz to 100 kHz, $T_J = 25^\circ\text{C}$			40	$\mu\text{V}/V_O$
SVR	Supply voltage rejection	$V_I = 8$ to 18 V, $f = 120$ Hz	68			dB
V_d	Dropout voltage	$I_O = 1$ A, $T_J = 25^\circ\text{C}$		2	2.5	V
R_O	Output resistance	$f = 1$ kHz		17		m Ω
I_{sc}	Short circuit current	$V_I = 35$ V, $T_J = 25^\circ\text{C}$		0.75	1.2	A
I_{scp}	Short circuit peak current	$T_J = 25^\circ\text{C}$	1.3	2.2	3.3	A

1. Load and line regulation are specified at constant junction temperature. Changes in V_O due to heating effects must be taken into account separately. Pulse testing with low duty cycle is used.

ANEXO N° 7

GENERAL SPECIFICATION

ITEM	DESCRIPTION			
Product No				
LCD Type	<input type="checkbox"/> STN Gray	<input type="checkbox"/> STN Yellow Green	<input checked="" type="checkbox"/> STN Blue	
	<input type="checkbox"/> TN Negative		<input type="checkbox"/> TN Positive	
	<input type="checkbox"/> FSTN Negative White & Black		<input type="checkbox"/> FSTN Positive Black & White	
Rear Polarizer	<input type="checkbox"/> Reflective	<input type="checkbox"/> Transflective	<input checked="" type="checkbox"/> Transmissive	
Backlight Type	<input type="checkbox"/> NO B/L	<input checked="" type="checkbox"/> LED	<input type="checkbox"/> CCFL	<input type="checkbox"/> EL
Backlight Color	<input type="checkbox"/> Yellow Green	<input type="checkbox"/> Green	<input type="checkbox"/> Amber	<input checked="" type="checkbox"/> White
				<input type="checkbox"/> Blue Green
View Direction	<input checked="" type="checkbox"/> 6 O'clock		<input type="checkbox"/> 12 O'clock	
Temperature Range	<input checked="" type="checkbox"/> Normal		<input type="checkbox"/> Wide	
Frame	<input checked="" type="checkbox"/> Black		<input type="checkbox"/> Silver	

TO BE VERY CAREFUL !

The LCD driver ICs are made by CMOS process, which are very easy to be damaged by static charge, make sure the user is grounded when handling the LCM.

! MUCHO CUIDADO !

El driver del LCD es fabricado con tecnología CMOS, los cuales pueden dañarse muy fácilmente por descargas estáticas, asegúrese de estar conectado a tierra cuando maneje el LCM

ABSOLUTE MAXIMUM RATING

(1) Electrical Absolute Ratings

Item	Symbol	Min.	Max.	Unit	Note
Power Supply for Logic	$V_{DD}-V_{SS}$	-0.3	7.0	Volt	
Power Supply for LCD	$V_{DD}-V_O$	-0.3	12.0	Volt	
Input Voltage	V_I	-0.3	V_{DD}	Volt	
LED Power Dissipation	P_{AD}	-	0.17	W	
LED Forward current	I_{AF}	-	40	mA	
LED Reverse Voltage	V_R	-	5	V	

(2) Environmental Absolute Maximum Ratings

Item	Normal Temperature				Wide Temperature			
	Operating		Storage		Operating		Storage	
	Min.	Max.	Min.	Max.	Min.	Max.	Min.	Max.
Ambient Temperature	0	+50°C	-20°C	+70°C	-20°C	+70°C	-30°C	+80°C
Humidity(without condensation)	Note 2,4		Note 3,5		Note 4,5		Note 4,6	

Note 2 $T_a \leq 50^\circ\text{C}$: 80% RH max

$T_a > 50^\circ\text{C}$: Absolute humidity must be lower than the humidity of 85%RH at 50°C

Note 3 T_a at -20°C will be <48hrs at 70°C will be <120hrs when humidity is higher than 70%.

Note 4 Background color changes slightly depending on ambient temperature. This phenomenon is reversible.

Note 5 $T_a \leq 70^\circ\text{C}$: 75RH max

$T_a > 70^\circ\text{C}$: absolute humidity must be lower than the humidity of 75%RH at 70°C

Note 6 T_a at -30°C will be <48hrs, at 80°C will be <120hrs when humidity is higher than 70%.

ORDEN DE ENCLAVAMIENTO

Escuela de Formación de Tecnólogos

FORNATO ENVA ORDEN DE ENCLAVAMIENTO

ELECTRICAL CHARACTERISTICS

Item	Symbol	Condition	Min.	Typ	Max.	Unit	note
Power Supply for Logic	$V_{DD}-V_{SS}$	-	4.5	5.0	5.5	Volt	
Input Voltage	V_{IL}	L level	0	-	0.6	Volt	
	V_{IH}	H level	2.2	-	V_{DD}	Volt	
LCM Recommend LCD Module Driving Voltage	$V_{DD}-V_O$	$T_a=0C$	-	-	-	Volt	
		$T_a=25C$	4.2	4.5	4.8		
		$T_a=50C$	-	-	-		
Power Supply Current for LCM	I_{DD}	$V_{DD}=5.0V$ $V_{DD}-V_O=4.5V$	-	2.0	3.0	mA	
LED For Ward Doltage	V_F	$I_f=20\text{ mA}$	-	3.4	3.6	Volt	
LED For Ward Current	I_F	-	-	20	-	mA	
LED Reverse Current	I_R	$V_R=5V$	-	-	0.2	mA	

OPTICAL CHARACTERISTICS

Item	Symbol	Condition	Min.	Typ	Max.	Unit	note
Viewing angle range	$\Phi f(12\text{ o'clock})$	When $Cr \geq 1.4$	-	10	-	Degree	9,10
	$\Phi b(6\text{ o'clock})$		-	30	-		
	$\Phi l(9\text{ o'clock})$		-	30	-		
	$\Phi r(3\text{ o'clock})$		-	30	-		
Rise Time	T_r	$V_{DD}-V_O=4.5V$ $T_a=25C$	-	200	-	mS	
Fall Time	T_f		-	250	-		
Frame frequency	F_{rm}		-	64	-		
Contrast	Cr		-	3.0	-		7
the Brightness Of Backlight	L	$I_f=20\text{ mA}$	120	190	-	cd/m ²	
Peak Emission Wavelength	λP		-	$X=0.51$ $Y=0.34$	-	nm	

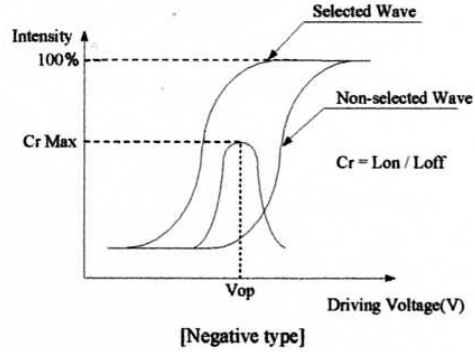
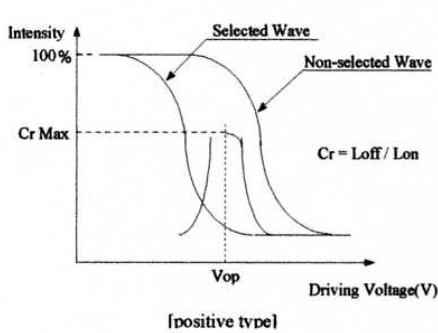
MECHANICAL SPECIFICATION

ITEM	DESCRIPTION
Product No.	SC1602A
Module Size	80.0(W)×36.0(H)×8.7(12.7) max(D)
Dot Size	0.56(W)mm×0.66(H)mm
Dot Pitch	0.60(W)mm×0.70(H)mm
Display Format	16 characters (W)×2 lines (H)
Duty Ratio	1/16 Duty
Controller	KS0066 or Equivalent

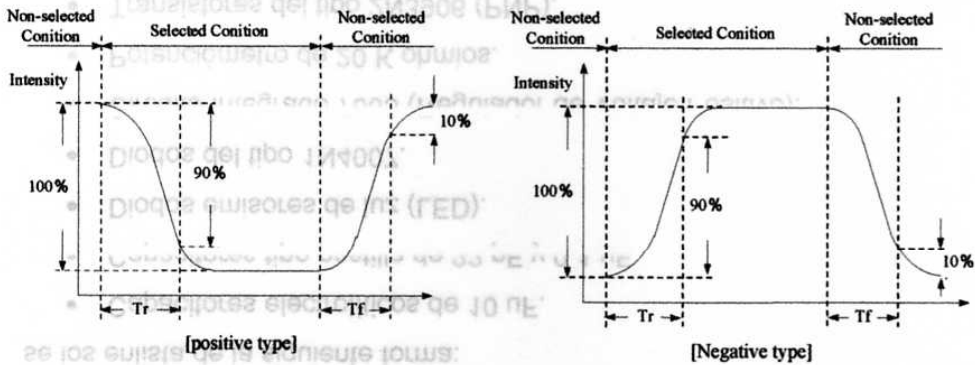
INTERFACE PIN ASSIGNMENT

Pin No.	Pin Out	Level	Description
1	VSS	0V	Power Supply Ground
2	VDD	5V	Power Supply Voltage
3	Vo	---	Contrast Adj
4	RS	H/L	Register Select
5	R/W	H/L	Read / Write
6	E	H,H→L	Enable Signal
7	DB0	H/L	Data Bit 0
8	DB1	H/L	Data Bit 1
9	DB2	H/L	Data Bit 2
10	DB3	H/L	Data Bit 3
11	DB4	H/L	Data Bit 4
12	DB5	H/L	Data Bit 5
13	DB6	H/L	Data Bit 6
14	DB7	H/L	Data Bit 7
15	A	3.5V	LED Power Supply (+)
16	K	0V	LED Power Supply (-)

[Note 7] Definition of Operation Voltage (Vop)



[Note 8] Definition of Response Time (Tr, Tf)



Conditions:

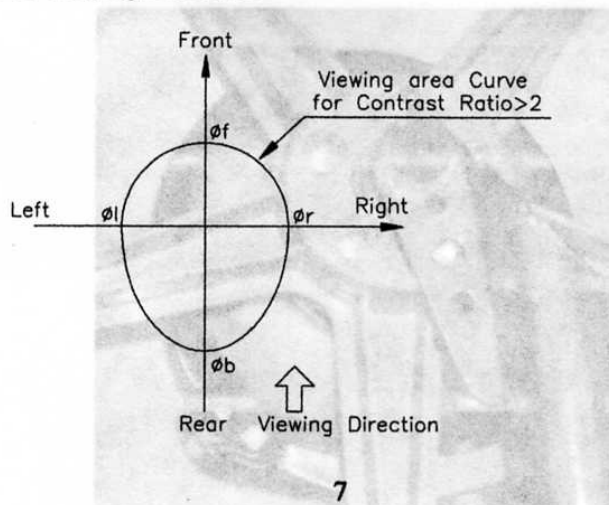
Operating Voltage : Vop

Viewing Angle(θ , ϕ) : 0° , 0°

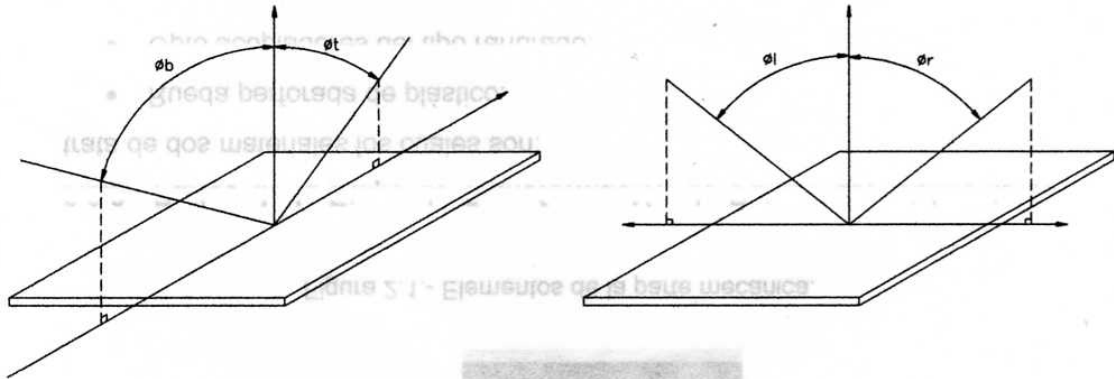
Frame Frequency : 64 Hz

Driving Wave form : 1/N duty, 1/a bias

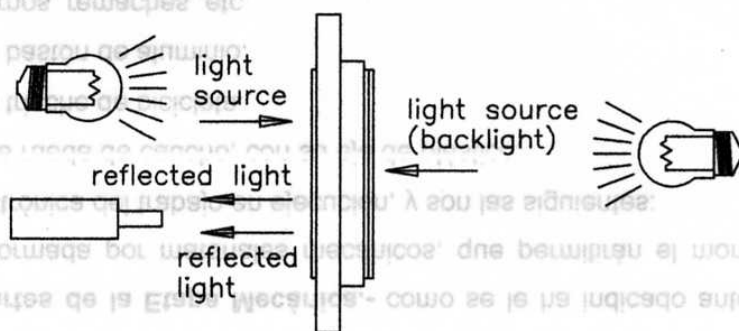
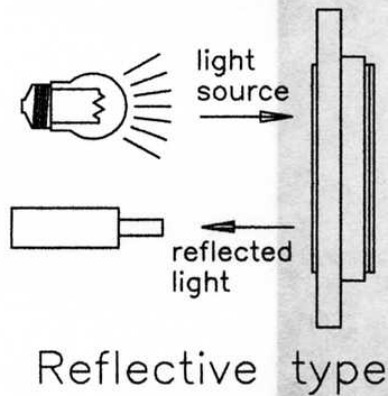
[Note 9] Definition of Viewing Direction



[Note 10] Definition of viewing angle

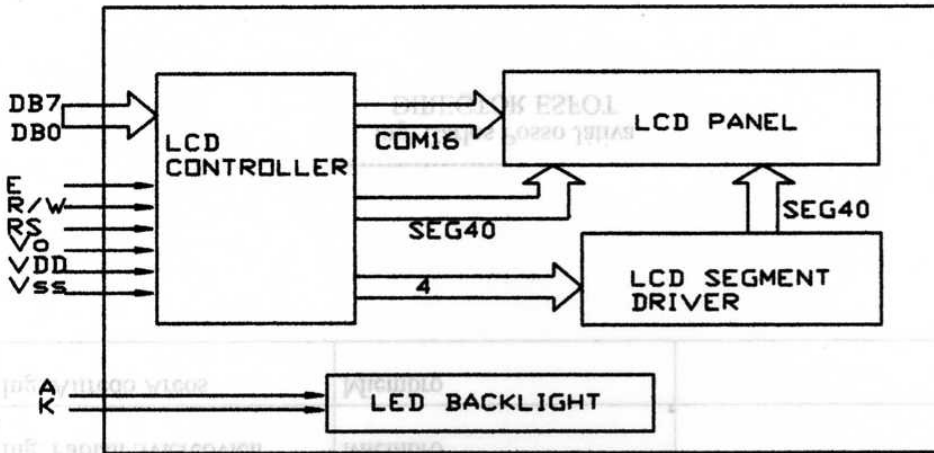


[Note 11] Description of Measuring Equipment



Transflective type

BLOCK DIAGRAM



POWER SUPPLY

