

INTRODUCCIÓN

En la actualidad se han desarrollado técnicas efectivas para el monitoreo y mantenimiento de la maquinaria de forma rápida y confiable que pone todo en manos de la instrumentación para evitar grandes costos y poder hacer un mantenimiento predictivo.

Los sistemas confiables para monitoreo y diagnóstico de maquinaria, provistos de señales de entrada de mediciones correctas, son muy apreciadas por su valor. Para medir su velocidad por ejemplo, se cuentan con diversos equipos entre ellos se encuentra el tacómetro.

El tacómetro es un dispositivo que mide las revoluciones por minuto (RPM) del rotor de un motor o de una turbina. Son utilizados para llevar un registro de las velocidades del elemento que tengamos en estudio, que nos permita saber si está trabajando en forma adecuada. Con este tipo de instrumentos evitaremos que se detenga la maquinaria, pudiendo hacer un mantenimiento en el momento adecuado. También se puede emplear para conocer distancias recorridas por ruedas, engranes o bandas.

El presente trabajo esta basado fundamentalmente en la implementación de un dispositivo en el cual se pueda observar el control de un servomotor y la medición de su velocidad con un tacómetro digital que sirva como ayuda didáctica para los estudiantes de la ESFOT que estén cursando en la actualidad la materia de Microprocesadores.

Todas las aplicaciones ya mencionadas y otras más, son posibles gracias al hecho de que los sistemas de control automático se diseñan para conservar uno o más parámetros de un proceso dado dentro de límites específicos con una pequeña intervención humana.

CAPITULO I: FUNDAMENTOS TEÓRICOS Y CONCEPTUALES

1.1. ATMEGA8L

Pues según ATMEL, sólo es un nombre que se sacaron de la manga. Algunos dicen que significa Advanced Virtual RISC. Otros dicen que lleva las iniciales de los inventores de los AVR: Alf Egil Bogen and Vegard Wollan... AlfVegardRisk. Reduced Instruction Set Computer es lo de RISC.

El dispositivo ATmega8L es un microcontrolador de baja potencia CMOS de 8 bits basado en arquitectura AVR RISC. Debido a que ejecuta poderosas instrucciones en un simple ciclo de reloj, el Atmega8L tiene un desempeño aproximado de 1 MIPS por Mhz, permitiendo un diseño del sistema para optimizar el consumo de energía contra la velocidad de procesamiento. El núcleo del microcontrolador AVR combina un set de instrucciones con 32 registros de propósito general. Todos los 32 registros están conectados directamente con la Unidad de Aritmética y Lógica permitiendo que dos registros independientes sean accedidos con una simple instrucción en un simple ciclo de reloj.

1.1.1. CARACTERÍSTICAS DEL ATMEGA8L:

- Alto-rendimiento, Bajo-power AVR® 8-bit Microcontroller.
- Avanzada Arquitectura RISK
 - 130 Poderosas Instrucciones – Más ejecución de un solo ciclo de reloj. 32 x 8 Registros Activos de propósito General.
 - Funcionamiento totalmente estático. Desde 16 MIPS hasta 16 MHz.
- Programa y Memoria de datos no volátil.
 - 8K Bytes memoria flash auto programable resiste : 10,000 ciclos escritura / borrado
 - Sistema interno programable por un chip interno programable, lectura exacta mientras realiza la operación de escritura.
 - 512 Bytes EEPROM resiste: 100,000 ciclos de escritura / borrado.
- Características Periféricas.
 - Dos Temporizadores / Contador de 8 bits con Prescaler separado, modo de comparación.
 - Un Temporizador / Contador de 16-bits con Prescaler separado, modo de comparación y modo de captación.
 - Contador con tiempo real y con Oscilador separador.
 - Tres canales PWM.
 - 8 canales ADC en TQFP y un paquete MLF de seis canales de 10-bit exactos, dos canales de 8-bit exactos. 6 canales ADC en PDIP exactos.
 - Cuatro canales de 10-bit exactos. Dos canales de 8-bit exactos.
 - Byte de orientación. Dos líneas seriales de Interfase.
 - USART Serial Programable. Interface Serial SPI Master/Slave.

- Temporizador del Perro Guardián Programable con el oscilador on-chip separado. On-chip Comparador Analógico.
- Características Especiales del Microcontrolador
 - Power-on Reset and Programmable Brown-out Detection.
 - Calibración Interna del Oscilador RC. Interrupciones generadas internas y externas.
 - Cinco modos sleep : detenido , ADC Reducción de Ruido, Power-save, Power-down, y estado de espera.
- I/O y Paquetes.
 - 23 líneas Programables I/O.
 - Voltajes de operación. 2.7 - 5.5V (ATmega8L)
- Grados de velocidad. 0 - 8 MHz (ATmega8L)

1.1.2. Distribución de pines del ATmega8L

Los pines están asignados de la siguiente forma

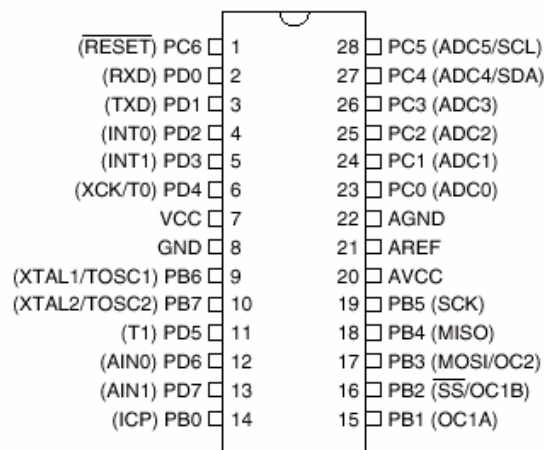


Figura 1.1.

1.1.3. Memorias del Atmega8L

El AVR tiene dos memorias principales:

& Memoria de Datos

& Memoria de espacio para el Programa

Adicionalmente el Atmega8l tiene una Memoria EEPROM para el almacenamiento de los datos. Las tres memorias son lineales y regulares.

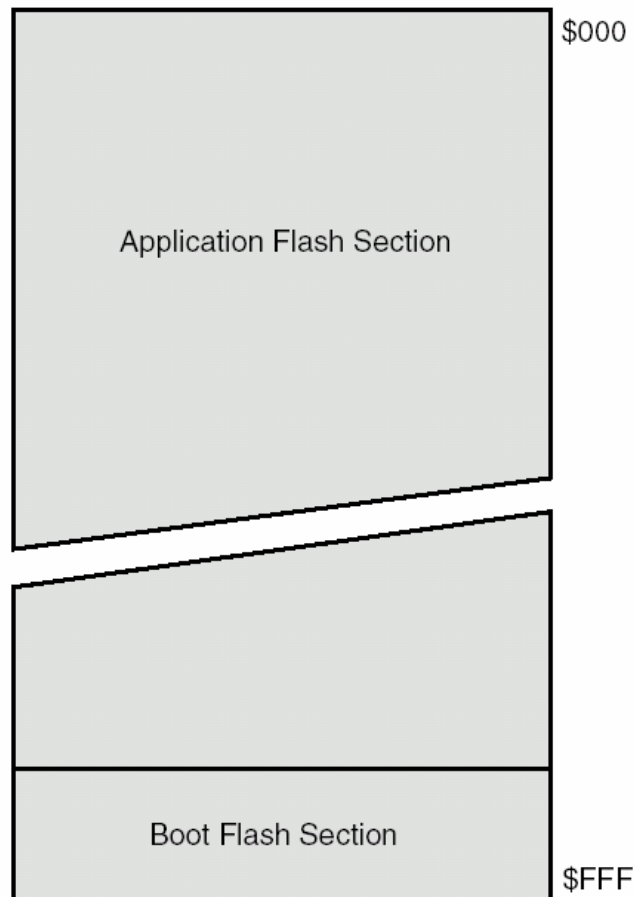


Figura 1.2.

1.1.3.1. Programación de la Memoria

Las instrucciones del AVR son desde 16 hasta 32 bits de ancho, el envío es organizada como 4K x 16 bits.

Para la seguridad del software, el envío del Programa al espacio de la memoria es dividido en dos secciones, sección de Boot Program y sección de Programa de Aplicación.

La operación de la sección Boot Program y la asociación de los bits del Boot Lock sirve para la protección de software para leer mientras escribe.

1.1.3.2. AVR CPU General Purpose Working Registers

	7	0	Addr.	
General Purpose Working Registers	R0		0x00	
	R1		0x01	
	R2		0x02	
	...			
	R13		0x0D	
	R14		0x0E	
	R15		0x0F	
	R16		0x10	
	R17		0x11	
	...			
	R26		0x1A	X-register Low Byte
	R27		0x1B	X-register High Byte
	R28		0x1C	Y-register Low Byte
	R29		0x1D	Y-register High Byte
	R30		0x1E	Z-register Low Byte
	R31		0x1F	Z-register High Byte

Figura 1.3.

1.1.3.3. Mapa de la Memoria de Datos

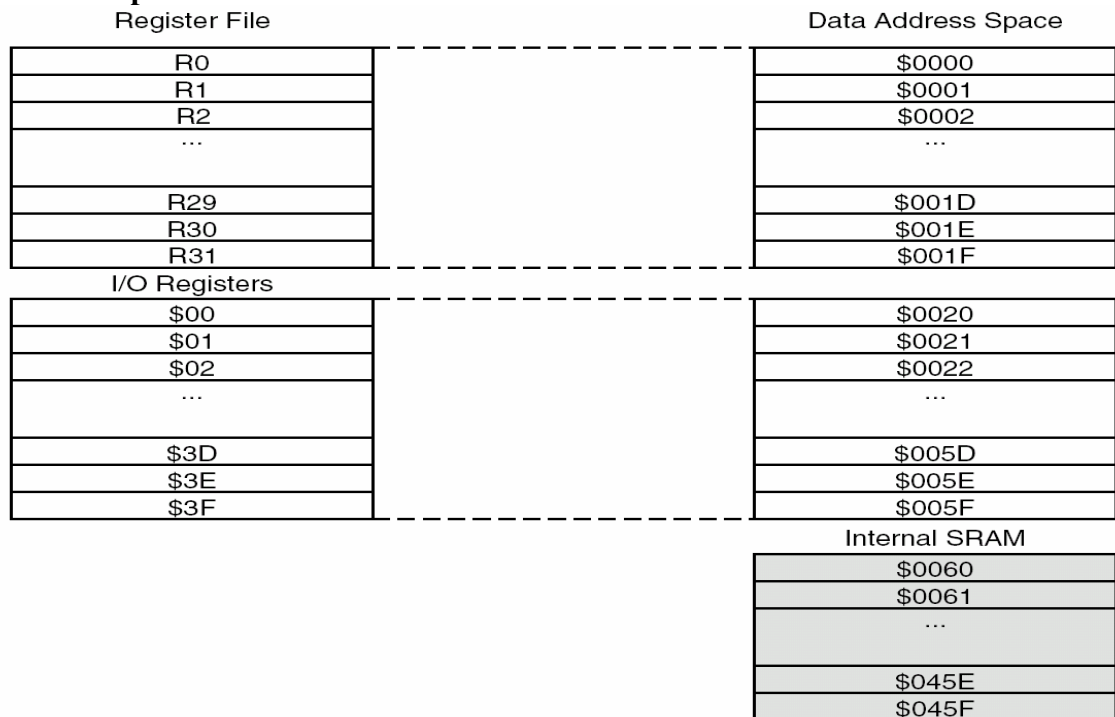


Figura 1.4.

1.2. Motores C.C.

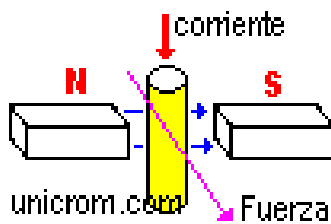
1.2.1. Introducción

Un motor de corriente continua está compuesto de un estator y un rotor. En muchos motores c.c., generalmente los más pequeños, el estator está compuesto de imanes para crear un campo magnético. En motores c.c. más grandes este campo magnético se logra con devanados de excitación de campo.

El rotor es el dispositivo que gira en el centro del motor y está compuesto de arrollados de cable conductores de corriente continua. Esta corriente continua es suministrada al rotor por medio de las "escobillas" generalmente fabricadas de carbón.

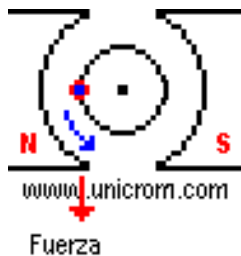
1.2.2. Principio básico de funcionamiento

Cuando un conductor por el que fluye una corriente continua es colocado bajo la influencia de un campo magnético, se induce sobre él (el conductor) una fuerza que es perpendicular tanto a las líneas de campo magnético como al sentido del flujo de la corriente.



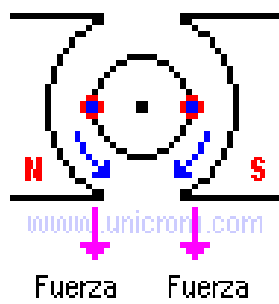
- Campo magnético en azul
- Corriente continua en rojo
- Dirección de la fuerza en violeta
- Imanes: N (norte) y S (sur)

Para que se entienda mejor, ver como se tiene que colocar este conductor con respecto al eje de rotación del rotor para que exista movimiento. En este caso la corriente por el conductor fluye introduciéndose en el gráfico.



- Par motor en azul
- Fuerza en violeta
- Conductor con corriente entrante en el gráfico azul y rojo
- Imanes: N (norte) y S (sur)

Pero en el rotor de un motor cc no hay solamente un conductor sino muchos. Si se incluye otro conductor exactamente al otro lado del rotor y con la corriente fluyendo en el mismo sentido, el motor no girará pues las dos fuerzas ejercidas para el giro del motor se cancelan.



- Par motor en azul
- Fuerza en violeta
- Conductor con corriente entrante en el gráfico azul y rojo
- Imanes: N (norte) y S (sur)

Es por esta razón que las corrientes que circulan por conductores opuestos deben tener sentidos de circulación opuestos. Si se hace lo anterior el motor girará por la suma de la fuerza ejercida en los dos conductores.

Para controlar el sentido del flujo de la corriente en los conductores se usa un conmutador que realiza la inversión del sentido de la corriente cuando el conductor pasa por la línea muerta del campo magnético.

La fuerza con la que el motor gira (el par motor) es proporcional a la corriente que hay por los conductores. A mayor tensión, mayor corriente y mayor par motor.

1.3. MOTOR PASO A PASO

1.3.1. INTRODUCCIÓN

En numerosas ocasiones es necesario convertir la energía eléctrica en energía mecánica, esto se puede lograr, por ejemplo, usando los motores de corriente continua. Pero cuando lo deseado es posicionamiento con un elevado grado de exactitud y/o una muy buena

regulación de la velocidad, se puede contar con una gran solución: utilizar un motor paso a paso.

Los motores paso a paso son ideales para la construcción de mecanismos en donde se requieren movimientos muy precisos.

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

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

1.3.2. PRINCIPIO BÁSICO DE FUNCIONAMIENTO

Básicamente estos motores están constituidos normalmente por un rotor sobre el que van aplicados distintos imanes permanentes y por un cierto número de bobinas excitadoras (bobinadas en su estator).

Las bobinas son parte del estator y el rotor es un imán permanente. Toda la conmutación (o excitación de las bobinas) deber ser externamente manejada por un controlador.

1.3.3. TIPOS DE MOTOR PASO A PASO

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

1.3.3.1. Motores Unipolares

En este tipo de motores, todas las bobinas del estator están conectadas en serie formando cuatro grupos. Esta a su vez, se conectan dos a dos, también en serie, y se montan sobre dos estatores diferentes. Según puede apreciarse en la figura 1.5, del motor paso a paso salen dos grupos de tres cables, uno de los cuales es común a dos bobinados. Los seis terminales que parten del motor, deben ser conectados al circuito de control, el cual, se comporta como cuatro conmutadores electrónicos que, al ser activados o desactivados, producen la alimentación de los cuatro grupos de bobinas con que está formado el estator. Si generamos una secuencia adecuada de funcionamiento de estos interruptores, se pueden producir saltos de un paso en el número y sentido que se desee.

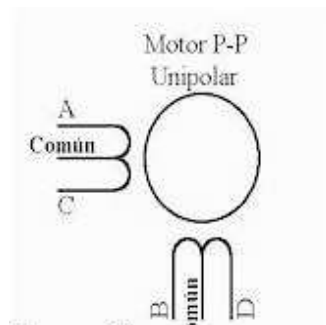


Figura 1.5.

1.3.3.2. Motores Bipolares

En este tipo de motores las bobinas del estator se conectan en serie formando solamente dos grupos, que se montan sobre dos estatores.

Según se observa en el esquema de este motor salen cuatro hilos que se conectan, al circuito de control, que realiza la función de cuatro interruptores electrónicos dobles, que nos permiten variar la polaridad de la alimentación de las bobinas. Con la activación y desactivación adecuada de dichos interruptores dobles, podemos obtener las secuencias adecuadas para que el motor pueda girar en un sentido o en otro.

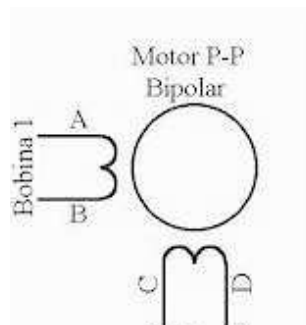


Figura 1.6.

1.4. SERVO MOTORES

1.4.1. INTRODUCCIÓN

El trabajo con servomotores juega un papel muy importante en el área industrial, ya que la tendencia en esta área es de automatizar las Máquinas-Herramientas y en este proyecto lo hemos utilizado para una mejor precisión y observación.

En todo sistema de control, el objetivo es mantener la variable y controlar dentro de ciertos límites permisibles por el sistema, en control de movimiento y aún más específicamente en los servo sistemas la variable típica de control es la posición.

El servo motor realiza el movimiento programado con precisión, controlando su posición y en consecuencia su velocidad y aceleración.

El siguiente es un esquema típico de un sistema de control de movimiento con servo motor sin escobillas:

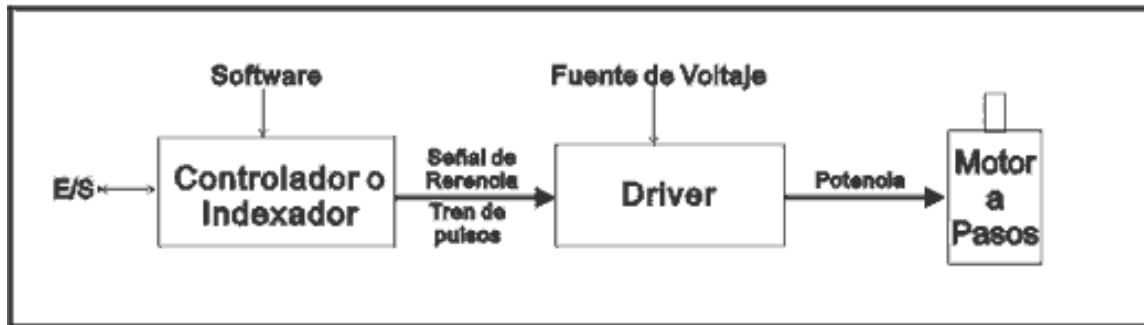


Figura 1.7.

1.4.2. PRINCIPIO DE FUNCIONAMIENTO

Que es un servomotor AC?

Es un motor eléctrico AC que puede realizar posicionamientos muy precisos y repetibles, ya que tiene retroalimentación permanente de posición (encoder). Es un sistema de lazo de control cerrado muy utilizado para generar posicionamientos específicos y controlados.

En otras palabras un servomotor (Servo) es un dispositivo electromecánico en el cual una entrada eléctrica determina la posición de un motor con la ayuda de un codificador óptico (*encoder*).

Este se encuentra adaptado al servomotor, del cual se obtiene una serie de pulsos que indican la posición en que se encuentra el servo, además del sentido de giro.

La importancia de conocer la posición de un servomotor es que en base a esta se puede conocer la velocidad y aceleración del dispositivo que lo utiliza haciendo uso, por ejemplo, de un controlador.



Figura 1.8.

En lo que respecta al campo de aplicación del encoder, lo tiene mayormente, en los sistemas de control de movimiento. El encoder entrega una serie de pulsos en cuadratura a través de dos canales, y dependiendo de la secuencia que estos tengan se obtiene el incremento o decremento de la posición en que se encuentre el servo. Hay un tercer canal, llamado índice, el cual indica el término de una vuelta completa del encoder. En la figura 1.10. es mostrada la serie de pulsos que entrega este dispositivo a través de sus tres canales:

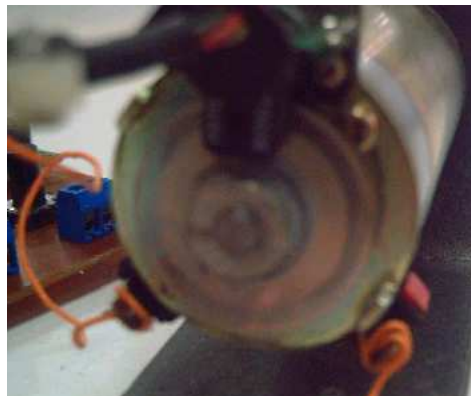


Figura 1.9.

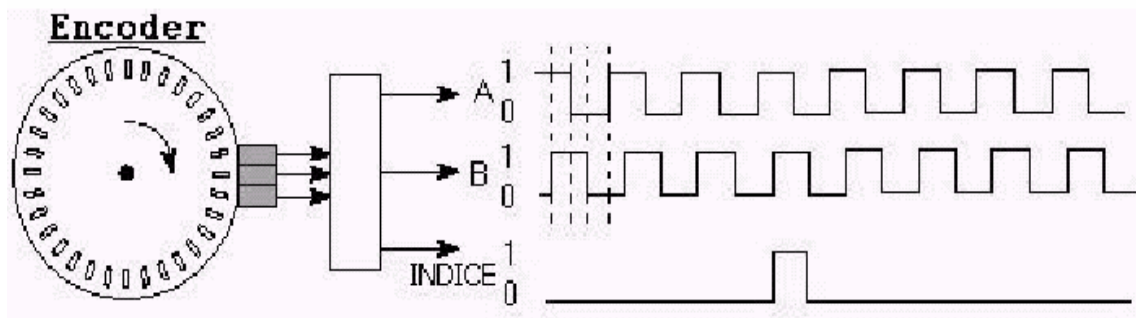


Figura 1.10.

Forma de onda que entrega el encoder

La serie de pulsos entregada por el encoder es relativa, mientras que a la salida del contador de pulsos se obtiene una posición absoluta.

1.4.3. TIPOS DE ENCODER

1.4.3.1. Encoders Incrementales

Los codificadores ópticos o encoders incrementales se utilizan fundamentalmente para el cálculo de la posición angular. Básicamente constan de un disco transparente, el cual tiene una serie de marcas opacas colocadas radialmente y equidistantes entre si; de un elemento emisor de luz (como un diodo LED); y de un elemento fotosensible que actúa como receptor.

El eje cuya posición angular se va a medir va acoplado al disco. (Ver figura1.11.)

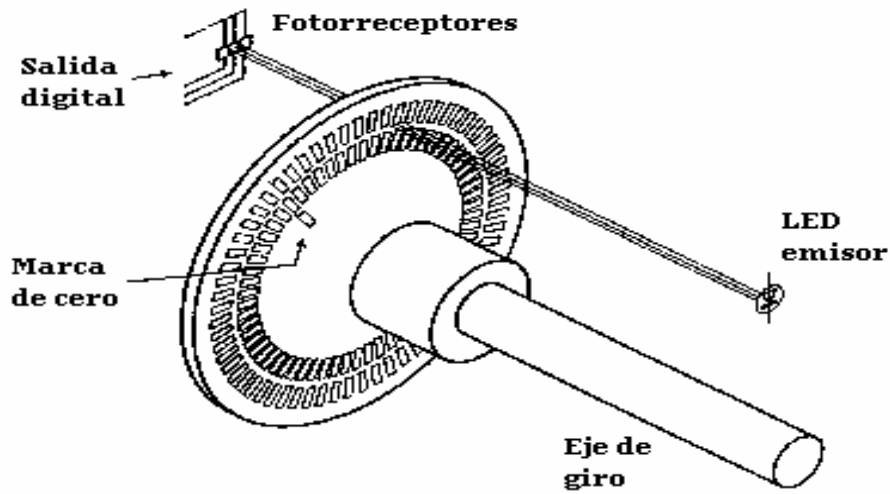


Figura 1.11.

El funcionamiento es el siguiente: cuando el sistema comienza a funcionar el emisor de luz empieza a emitir; a medida que el eje vaya girando, se producirán una serie de pulsos de luz en el receptor, correspondientes a la luz que atraviesa los huecos entre las marcas. Llevando una cuenta de esos pulsos es posible conocer la posición del eje.

Sobre este esquema básico es habitual encontrar algunas mejoras. Por ejemplo, se suele introducir otra franja de marcas por debajo, desplazada de la anterior, para poder controlar el sentido del giro; además suele ser necesario el empleo de una marca de referencia que nos ayudará a saber si hemos completado una vuelta.

Realmente los encoders incrementales miden la velocidad de giro, pero podemos extrapolar la posición angular. Como es lógico, la resolución de este tipo de sensores depende directamente del número de marcas que podamos poner físicamente en el disco.

1.4.3.2. Encoder absoluto

La función de este tipo de dispositivos es similar a la de los anteriores, medir la posición angular. Sin embargo en este caso lo que se va a medir no es el incremento de esa posición, sino la posición exacta. La disposición es parecida a la de los encoders incrementales. También se dispone de una fuente de luz, de un disco graduado y de un fotorreceptor. La diferencia estriba en la graduación o codificación del disco. En este caso el disco se divide en un número fijo de sectores (potencia de 2) y se codifica cada uno con un código cíclico (normalmente un código de Gray); este código queda representado en el disco por zonas transparentes y opacas dispuestas radialmente, como se puede apreciar en la figura 1.12. No es necesaria ninguna mejora para detectar el sentido del giro, ya que la codificación de los distintos sectores angulares es absoluta. La resolución de estos sensores es fija y viene dada por el número de anillos que posea el disco, o lo que es lo mismo, el número de bits del código utilizado. Normalmente se usan códigos de 8 a 19 bits. Tanto los encoders absolutos como los incrementales pueden presentar problemas debido a la gran precisión que es necesaria en el proceso de fabricación. Además son dispositivos especialmente sensibles a golpes y vibraciones.

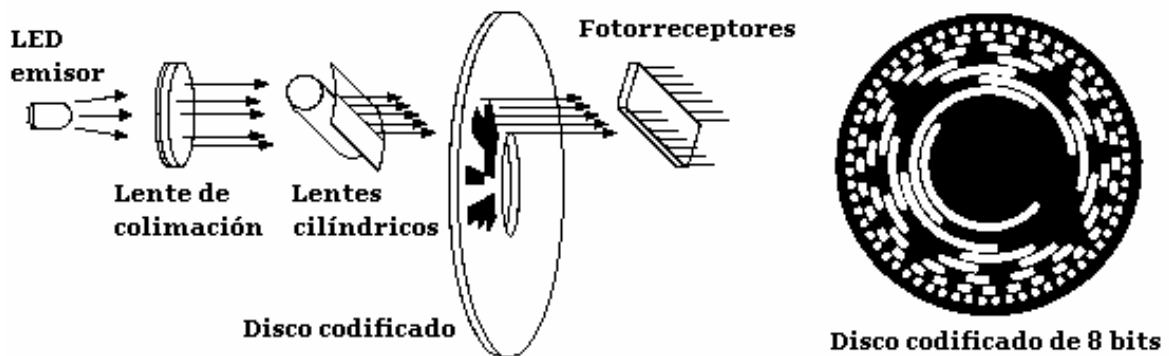


Figura 1.12.

1.5. COMPONENTES ELECTRÓNICOS UTILIZADOS

1.5.1. RESISTENCIAS

Desde el punto de vista de la resistividad, podemos encontrar materiales conductores (no presentan ninguna oposición al paso de la corriente eléctrica), aislantes (no permiten el flujo de corriente), y resistivos (que presentan cierta resistencia).

Dentro de este último grupo se sitúan las resistencias. Es por esto que se fabrican un tipo de componentes llamados resistores cuyo único objeto es proporcionar en un pequeño tamaño una determinada resistencia, especificada por el fabricante.



Figura 1.13.

Las resistencias son componentes eléctricos pasivos en los que la tensión instantánea aplicada es proporcional a la intensidad de corriente que circula por ellos. Su unidad de medida es el ohmio (Ω).

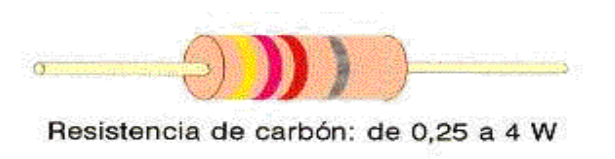
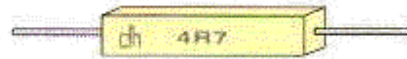


Figura 1.14.

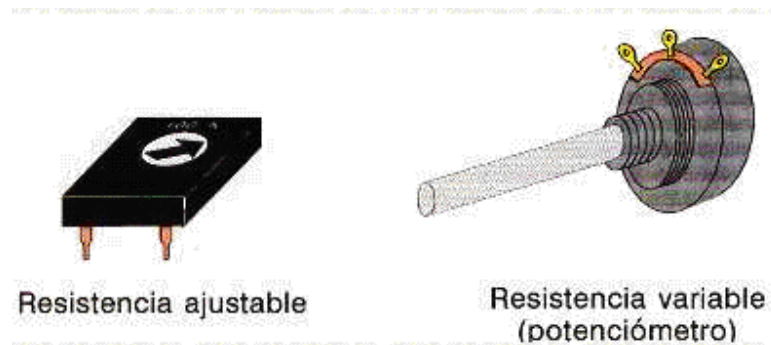


Figura 1.15.



Resistencia bobinada cementada: de 2 a 15 W

Figura 1.16.



Resistencia ajustable

Resistencia variable
(potenciómetro)

Figura 1.17.

1.5.2. CONDENSADOR

El condensador es uno de los componentes mas utilizados en los circuitos eléctricos. Un condensador es un componente pasivo que presenta la cualidad de almacenar energía eléctrica. Esta formado por dos láminas de material conductor (metal) que se encuentran separados por un material dieléctrico (material aislante). En un condensador simple, cualquiera sea su aspecto exterior, dispondrá de dos terminales, los cuales a su vez están conectados a las dos laminas conductoras.



1.5.2.1. Condensador no polarizado Condensador variable

Básicamente, un capacitor, en su expresión más simple, está formado por dos placas metálicas (conductoras de la electricidad) enfrentadas y separadas entre sí por una mínima distancia, y un dieléctrico, que se define como el material no conductor de la electricidad (aire, mica, papel, aceite, cerámica, etc.) que se encuentra entre dichas placas. La magnitud del valor de capacidad de un capacitor es directamente proporcional al área de sus placas e inversamente proporcional a la distancia que las separa. Es decir, cuanto mayor sea el área de las placas, mayor será el valor de capacidad.

1.5.2.2. Capacitores Electrolíticos

Dentro de la gran variedad de tecnologías de fabricación de capacitores, los electrolíticos son los de mayor capacidad, debido a que se recurre a reducir la separación entre las placas, a aumentar el área enfrentada de las mismas y a la utilización de un dieléctrico de elevada constante dieléctrica.

Los capacitores electrolíticos deben su nombre a que el material dieléctrico que contienen es un ácido llamado electrolito y que se aplica en estado líquido. Para que pueda ser conectado en un circuito electrónico, el capacitor llevará sus terminales de conexión remachados o soldados con soldadura de punto.

1.5.2.3. Los capacitores cerámicos

Están hechos en numerosos estilos:

La fabricación de los capacitores comienza con cilindros u hojas de cerámica que está hecha de una pantalla de seda o pintada con una pasta de metal precioso consistiendo en platino, paladio o plata para formar los electrodos. Las hojas de electrodo están apiladas o

procesadas como una capa simple y está subsecuentemente cortada en cuadrados, discos, rectángulos u otros diseños.

En un diseño monolítico las hojas están apiladas y compactadas, conteniendo electrodos en cada corte.

Los terminales de la sección cerámica con electrodos expuestos a una pasta de metal precioso generalmente plata, o plata – platino fundidos por encima de los 750 °C para formar terminales de metal soldable.

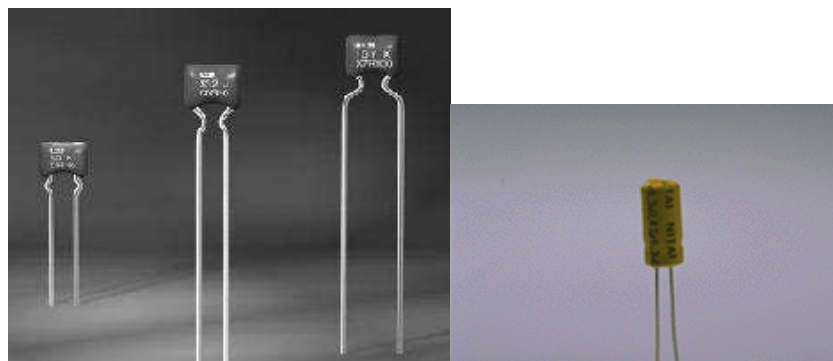
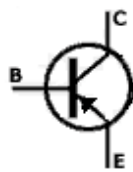


Figura 1.18.

1.5.3. TRANSISTORES

Los transistores se componen de semiconductores. Se trata de materiales, como el silicio o el germanio, dopados (es decir, se les han incrustado pequeñas cantidades de materias extrañas), de manera que se produce un exceso o una carencia de electrones libres. En el primer caso, se dice que el semiconductor es del tipo n, y en el segundo, que es del tipo p. Combinando materiales del tipo n y del tipo p se puede producir un diodo. Cuando éste se

conecta a una batería de manera tal que el material tipo p es positivo y el material tipo n es negativo, los electrones son repelidos desde el terminal negativo de la batería y pasan, sin ningún obstáculo, a la región p, que carece de electrones. Con la batería invertida, los electrones que llegan al material p pueden pasar sólo con muchas dificultades hacia el material n, que ya está lleno de electrones libres, en cuyo caso la corriente es prácticamente cero.



Transistor PNP



Transistor NPN

Las capas semiconductoras con sus conexiones son el emisor E, el colector C y la base B. Los transistores se emplean como interruptores o bien como un relé, y también como amplificadores. El transistor en su función como interruptor cumple las siguientes exigencias: Sin necesidad de contactos conectan con pequeña corriente de mando una gran corriente de trabajo, ya que no existen las partes móviles mecánicamente, trabaja sin desgaste y sin ruido. Los procesos de conexión se efectúan sin retardo, en el campo de

micros segundos (us). Además, no puede producirse salto de chispa. Ver el siguiente grafico.

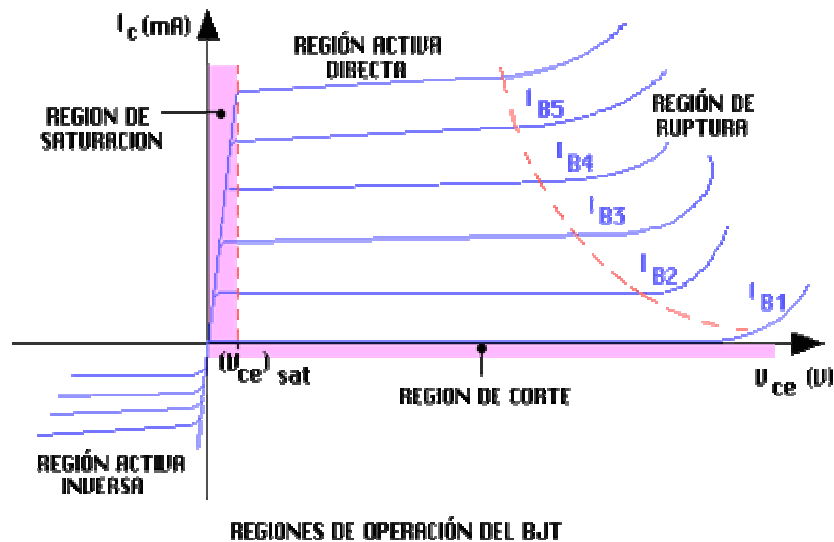


Figura 1.19.

1.5.3.1. Transistor de efecto campo.

Existen unos dispositivos que eliminan este inconveniente en particular y que pertenece a la familia de dispositivos en los que existe un solo tipo de portador de cargas, y por tanto, son unipolares. Se llama un transistor de efecto campo (FET) típico está formado por una barrita de material p ó n, llamada canal, rodeada en parte de su longitud por un collar del otro tipo de material que forma con el canal una unión p-n.

En los extremos del canal se hacen sendas conexiones óhmicas llamadas respectivamente sumidero (d-drain) y fuente (s-source), más una conexión llamada puerta (g-gate) en el collar.

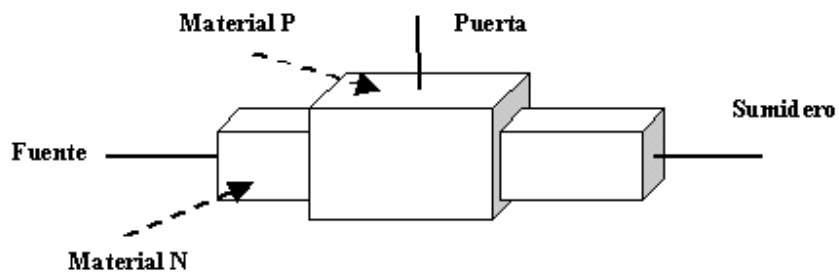


Figura 1.20.

La figura 1.21. muestra el croquis de un FET con canal N

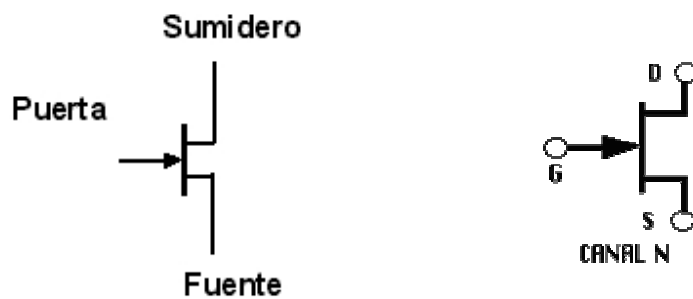


Figura 1.21.

Símbolos gráficos para un FET de canal P

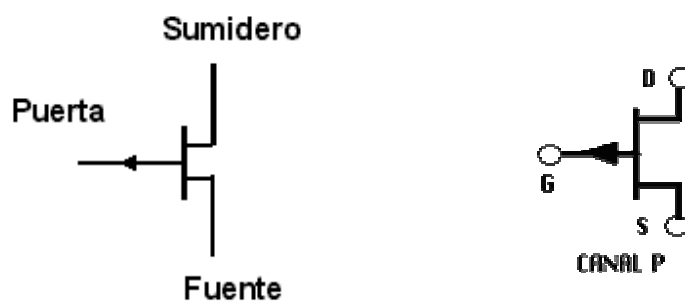


Figura 1.22.

Símbolos gráficos para un FET de canal N

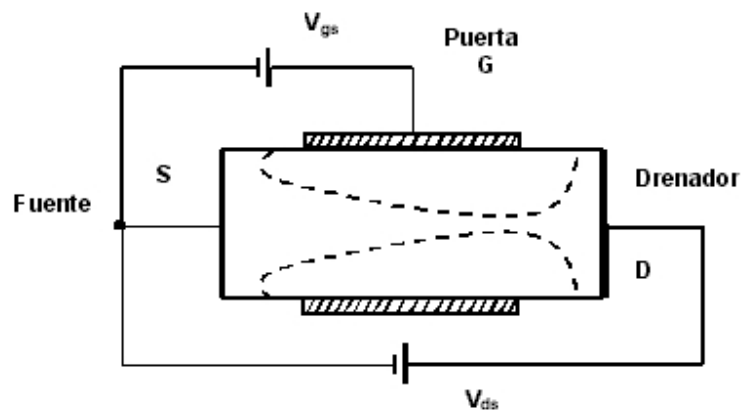


Figura 1.23.

1.5.3.2. Darlington

1. Un ejemplo de Darlington

El TIP120 es un ejemplo de par Darlington, tiene un encapsulado del tipo TO220.

La ganancia de corriente según las especificaciones del fabricante es de 1000, y la máxima corriente que puede circular por el colector es de 5 A.

Además de los dos transistores propios del par Darlington, este dispositivo, lleva un diodo adicional y un par de resistencias con fines de protección.

El transistor Darlington es un tipo especial de transistor que tiene una alta ganancia de corriente. Está compuesto internamente por dos transistores de la forma que muestra la figura 1.24.

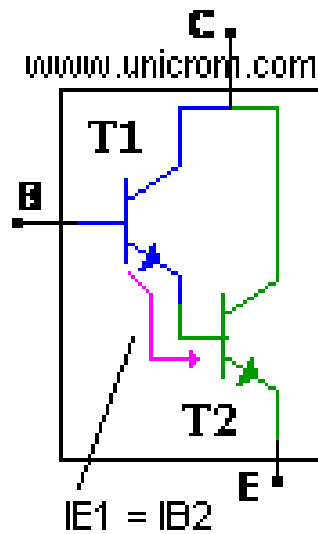


Figura 1.24.
*El transistor Darlington con la identificación de las patillas
 y su estructura interna*

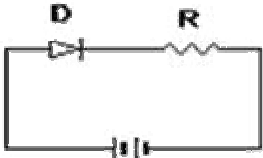
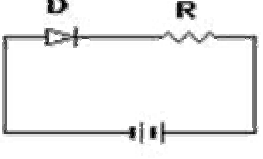
El transistor T1 entrega la corriente que sale por su emisor a la base del transistor T2.

La ecuación de ganancia de un transistor típico es: $IE = \beta \times IB$ (Corriente de colector es igual a beta por la corriente de base).

Muy importante: la caída de tensión entre la base y el emisor del transistor Darlington es 1.4 voltios que resulta de la suma de las caídas de tensión de base a emisor del primer transistor B1 a E1 (0.7 voltios) y base a emisor del segundo transistor B2 y E2 (0.7 voltios).

1.5.4. EL DIODO

Como todos los componentes electrónicos, los diodos poseen propiedades que les diferencia de los demás semiconductores.

POLARIZACIÓN	CIRCUITO	CARACTERÍSTICAS
<p><i>DIRECTA</i></p> <p><i>el ánodo se conecta al positivo de la batería y el cátodo al negativo.</i></p>		<p>El diodo conduce con una caída de tensión de 0,6 a 0,7V. El valor de la resistencia interna será muy bajo. Se comporta como un interruptor cerrado</p>
<p><i>INVERSA</i></p> <p><i>el ánodo se conecta al negativo y el cátodo al positivo de la batería</i></p>		<p>El diodo no conduce y toda la tensión de la pila cae sobre él.</p> <p>Puede existir una corriente de fuga del orden de uA.</p> <p>El valor de la resistencia interna sería muy alto. Se comporta como un interruptor abierto.</p>

Cuadro 1.1.

1.5.4.1. Tipos de diodos

1.5.4.1.1. Diodos Metal-Semiconductor

Los más antiguos son los de germanio con punta de tungsteno o de oro. Su aplicación más importante se encuentra en HF, VHF y UHF. También se utilizan como detectores en los receptores de modulación de frecuencia. Por el tipo de unión que tiene posee una capacidad

muy baja, así como una resistencia interna en conducción que produce una tensión máxima de 0,2 a 0,3v.

El encapsulado de estos diodos es en forma de cilindro, de plástico o de vidrio. De configuración axial. Sobre el cuerpo se marca el cátodo, mediante un anillo serigrafiado.

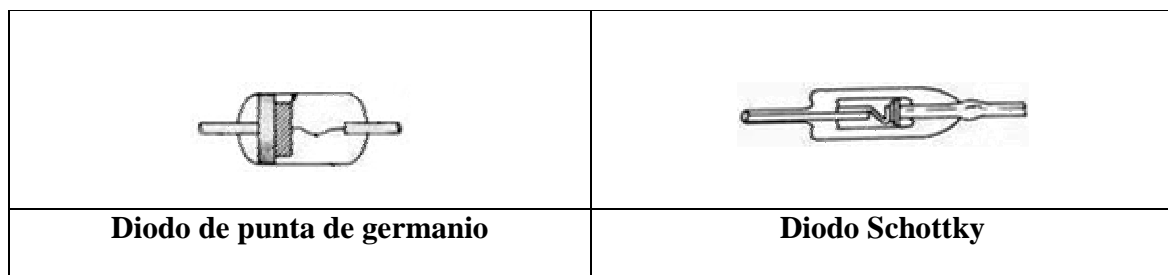


Figura 1.25.

1.5.4.1.2. Diodos Led

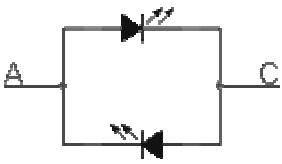
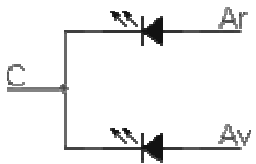

Es un diodo que presenta un comportamiento parecido al de un diodo rectificador sin embargo, su tensión de umbral, se encuentra entre 1,3 y 4v dependiendo del color del diodo.

& LED bicolor.- Están formados por dos diodos conectados en paralelo e inverso. Se suele utilizar en la detección de polaridad.

& LED tricolor.- Formado por dos diodos LED (verde y rojo) montado con el cátodo común. El terminal más corto es el ánodo rojo, el del centro, es el cátodo común y el tercero es el ánodo verde.

1.5.4.1.3. Display

Es una combinación de diodos LED que permiten visualizar letras y números. Se denominan comúnmente displays de 7 segmentos. Se fabrican en dos configuraciones: ánodo común y cátodo común.

Estructura de un LED bicolor	Estructura de un LED tricolor	Display
		
Display de cátodo común	Display de ánodo común	Disposición de los pines en un display

Cuadro 1.2.

1.6. EL LCD

1.6.1. INTRODUCCION

Es una pantalla de cristal líquido que visualiza unos ciertos caracteres. En total se pueden visualizar 2 líneas de 16 caracteres cada una, es decir, $2 \times 16 = 32$ 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.

Tiene un consumo de energía de menos de 5mA y son ideales para dispositivos que requieran una visualización pequeña o media.



Figura 1.26.

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.

1.6.2. TIPOS DE LCD

En la actualidad existe una gran variedad de versiones, clasificadas en dos grupos. El primer grupo está referido a los módulos LCD de caracteres (solamente se podrán presentar caracteres y símbolos especiales en las líneas predefinidas en el módulo LCD) y el segundo grupo está referido a los módulos LCD matriciales (Se podrán presentar caracteres, símbolos especiales y gráficos). Los módulos LCD varían su tamaño físico dependiendo de la marca.

1.6.3. MEMORIAS DEL LCD















Un dispositivo LCD dispone de dos tipos de memorias ambas independientes.

Estas memorias se denominan DD RAM y CG RAM.

La memoria DD RAM es la memoria encargada de almacenar los caracteres de la pantalla que se estén visualizando en ese momento, o bien, que estén en una posición no visible. El display tiene una capacidad de 2 líneas horizontales por 40 caracteres cada una, de los cuales solo serán visibles 2 líneas de 16 caracteres cada una. La DDRAM tiene un tamaño de $2 \times 40 = 80$ bytes. El orden de conteo de los caracteres sería empezando de izquierda a derecha, de tal modo que el carácter 1 sería el primero de la izquierda y el 40 sería el situado más a la derecha. Para localizar los caracteres se utilizan dos coordenadas (X,Y) siendo Y el valor vertical comprendido entre 1 y 2 (valor de línea) y X el valor horizontal (1-40) que indicara el carácter.

1.6.4. ASIGNACIÓN DE PINES

A continuación podemos notar la asignación de pines en una pantalla LCD

1		VSS		
2		VDD		
3		VC		
4		RS		
5		R/W		
6		E		
7		D0		
8		D1		
9		D2		
10		D3		
11		D4		
12		D5		
13		D6		
14		D7		

Nº de PIN	Simbolo	Descripción
1	VSS	Max
2	VDD	Alimentación
3	VC	Voltaje de ajuste del contraste
4	RS	Selección de registro
5	R/W	Lectura/escritura
6	E	Enable
7	D0	Bit de datos menos significativo
8	D1	Bit de datos
9	D2	Bit de datos
10	D3	Bit de datos
11	D4	Bit de datos
12	D5	Bit de datos
13	D6	Bit de datos
14	D7	Bit de datos mas significativo

Figura 1.27.

- & Los pines 1 y 2, son los utilizados para la alimentación del módulo LCD. La tensión utilizada es de 5 voltios.
- & El pin 3 se utiliza para ajustar el contraste de la pantalla LCD. Por medio de un potenciómetro regularémos la intensidad de los caracteres, a mayor tensión mayor intensidad.
- & El pin 4 se utiliza para indicar al bus de datos si la información que le llega es una instrucción o por el contrario es un carácter. Si $RS=0$ indicara que en el bus de datos hay presente una instrucción, y si $RD=1$, indicará que tiene un carácter alfanumérico.
- & El pin 5 es el de escritura o lectura. Si esta a 0 el módulo escribe en pantalla el dato que haya en el bus de datos, y si esta a 1 leeremos lo que hay en el bus de datos.
- & El pin 6 es el indicado de hacer que el módulo LCD funcione, o por el contrario no acepte órdenes de funcionamiento. Cuando $E=0$ no se podrá utilizar el display y cuando $E=1$ se podrán transferir datos y realizar las demás operaciones.
- & Los pines del 7 al 10 no van conectados.
- & Los pines del 11 al 14 son los del bus de datos que para este proyecto hemos tomado el pin 4 (RS) y el pin 6 (E) como datos.
- & El pin 15 es ánodo y lo conectamos a Vcc.
- & El pin 16 es cátodo y lo conectamos a GND

CAPITULO II: IMPLEMENTACIÓN DEL DISPOSITIVO

2.1. DIAGRAMA DE BLOQUES DEL SISTEMA:

El diagrama muestra que la señal enviada por el Servomotor es dirigida al contador del microcontrolador Atmega8L el cual se encarga de enviar otra señal al LCD en RPM (velocidad del motor)

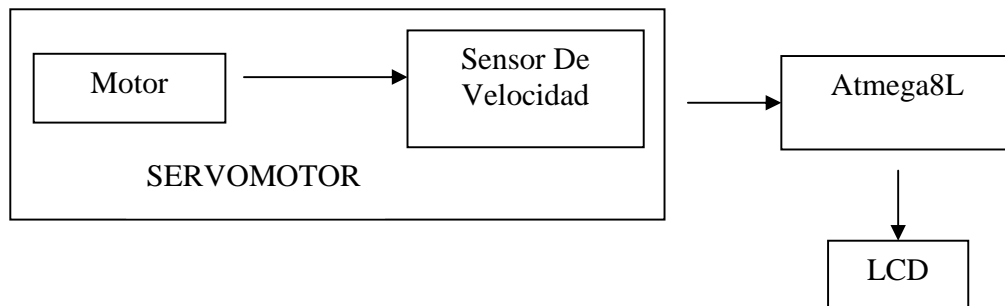


Figura 2.1.

2.2. DISEÑO Y CONSTRUCCIÓN DEL DISPOSITIVO

El dispositivo a implementar consiste en un Tacómetro Digital que nos servirá como material didáctico basado en el microcontrolador Atmega8L para los estudiantes de la ESFOT.

Este proyecto beneficiará al Laboratorio de Microprocesadores de la ESFOT de la EPN en la optimización del tiempo en la comprensión del funcionamiento de los diferentes Microcontroladores que existen en el mercado.

Los estudiantes que estén cursando la materia de Control con Microprocesadores tendrán material didáctico para el estudio de los Microcontroladores.

El Tacómetro Digital está diseñado para medir la velocidad de un encoder, en el que podremos observar la variación de la velocidad (RPM) del mismo y con la ayuda de dos pulsadores que incrementa y decrementa la rapidez de acuerdo a nuestra necesidad de visualización.

El Tacómetro Digital consta de las siguientes programaciones divididas en partes correspondientes como son:

- Programación del Microcontrolador
- Programación del LCD

2.3. PARTES DE LA ESTRUCTURA DEL TACOMETRO DIGITAL

El dispositivo consta de un Microcontrolador Atmega8L, una fuente reguladora de voltaje para el motor, un circuito programador, un encoder, circuitería adicional como son transistores, resistencias, capacitores, etc., el LCD y un código de programa que se incluye en los anexos.

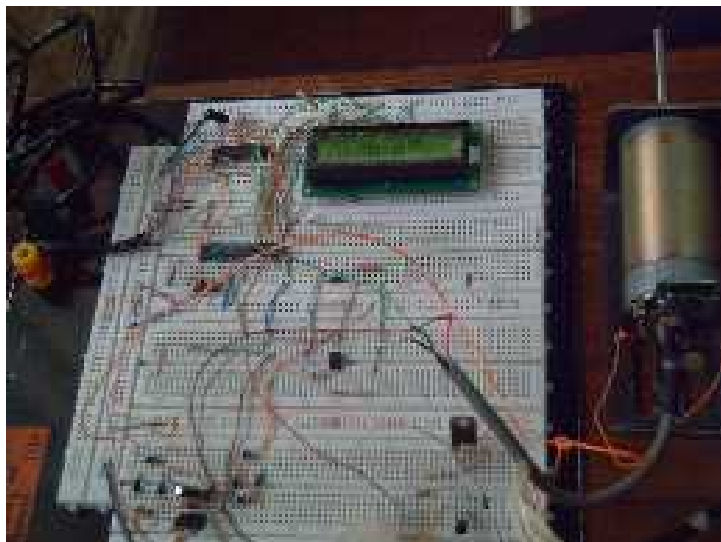


Figura 2.2.

2.3.1. PWM

2.3.1.1. Introducción

Para variar la velocidad del motor, hemos empleado una técnica conocida como "Modulación en Anchura de Pulsos" (o PWM, del inglés "Pulse Width Modulation"). Con esta técnica conseguimos que la velocidad del motor varíe en función de la duración del ciclo de trabajo de una señal que se envía al "encoder".

Desafortunadamente, un determinado ciclo de trabajo de la señal PWM no se corresponde con una determinada velocidad, pues el rendimiento del motor para una determinada señal PWM depende de otros factores (peso de la carga, pendiente del terreno,...). De hecho, dos motores iguales no trabajan a la misma velocidad para la misma señal PWM debido a diferencias en la construcción de los mismos. Por tanto, no podemos garantizar una determinada velocidad a partir únicamente de la señal PWM.

Para solventar este problema se ha implementado un algoritmo de control (relación de números de pulsos del encoder en una vuelta) en lazo cerrado que a partir de la velocidad del encoder varía el ciclo de trabajo de la señal PWM para intentar aproximar la velocidad medida en instantes sucesivos a la velocidad deseada.

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



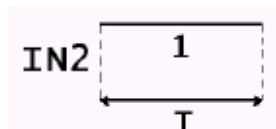
Con esta forma de control la cantidad de corriente sigue siendo la misma, la tensión no varía y en consecuencia el torque del motor se mantiene, que es justamente lo que estábamos buscando.

La dirección y la velocidad van a venir dadas por el ciclo de trabajo de la señal PWM.

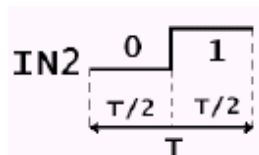
De esta manera la corriente del motor cambia de sentido en cada uno de los ciclos de la señal de control, entonces se establece la relación PWM-dirección /velocidad

de la siguiente manera:

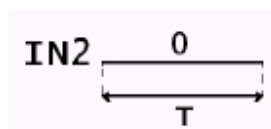
& 0%) Adelante/ atrás máxima velocidad, la corriente solo circula en un único sentido.



& 50%) Parado, la corriente circula en un sentido tanto tiempo como en el otro.



& 100%) Atrás /adelante máxima velocidad, similar al 0%.



Los valores intermedios entre los dados nos permiten controlar la velocidad del motor de manera sencilla.

Utilizando el simulador en tiempo real Proteus (ISIS) hemos variado y observado las siguientes formas de onda de la señal entregada por el PWM del Tacómetro.

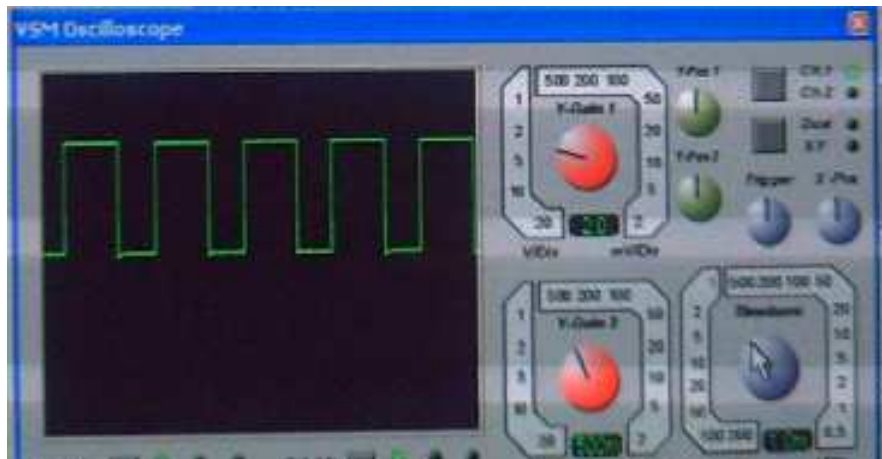


Figura 2.3.

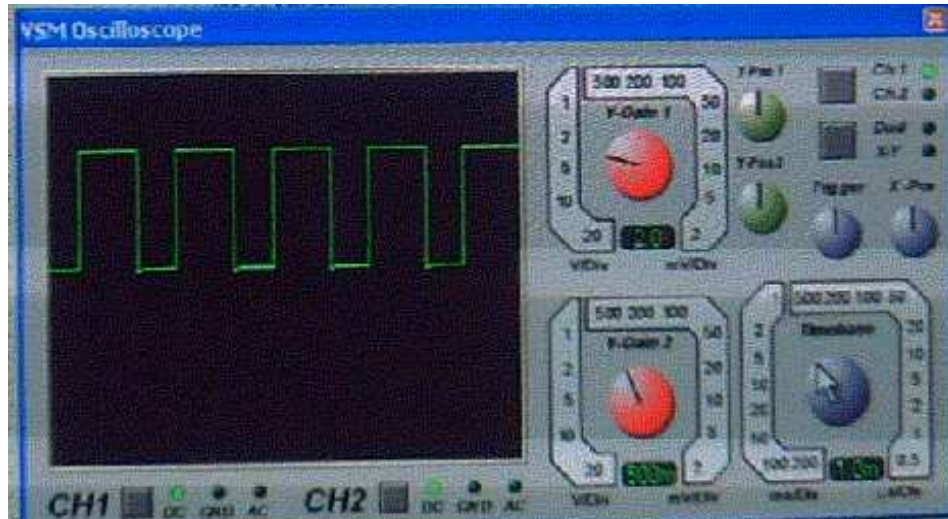


Figura 2.4.

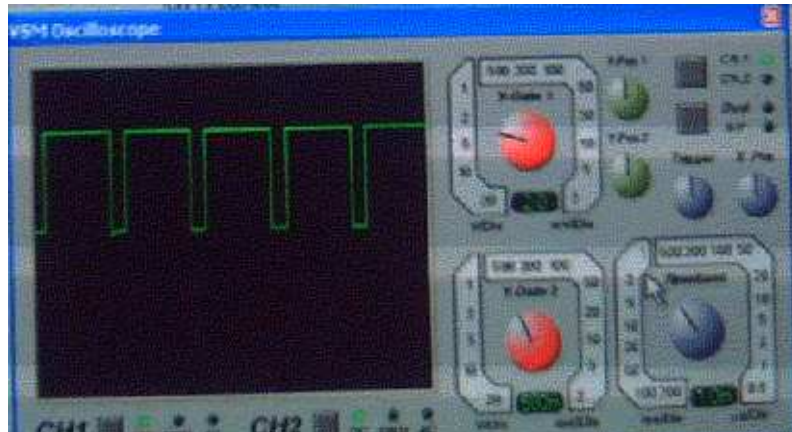


Figura 2.5.

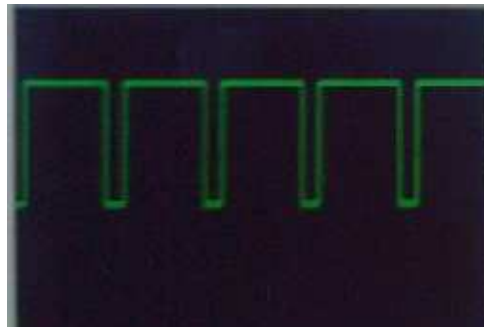


Figura 2.6.

Se observa que las señales son diferentes en cuanto al ancho de pulso ya que vamos variando la señal desde el microcontrolador utilizando unos pulsadores y por tanto la velocidad también varía.

El PWM está ligado directamente al T1 (Timer 1).

Salidas PWM :

& OC1A
 & OC1B

} Son independientes ligados al T1

2.3.2. PROGRAMACIÓN DEL MICROCONTROLADOR

2.3.2.1. Programador del Atmega8L

Utilizaremos el puerto paralelo para conectar al programador que emplearemos. La figura 2.7. muestra cómo montamos el programador.

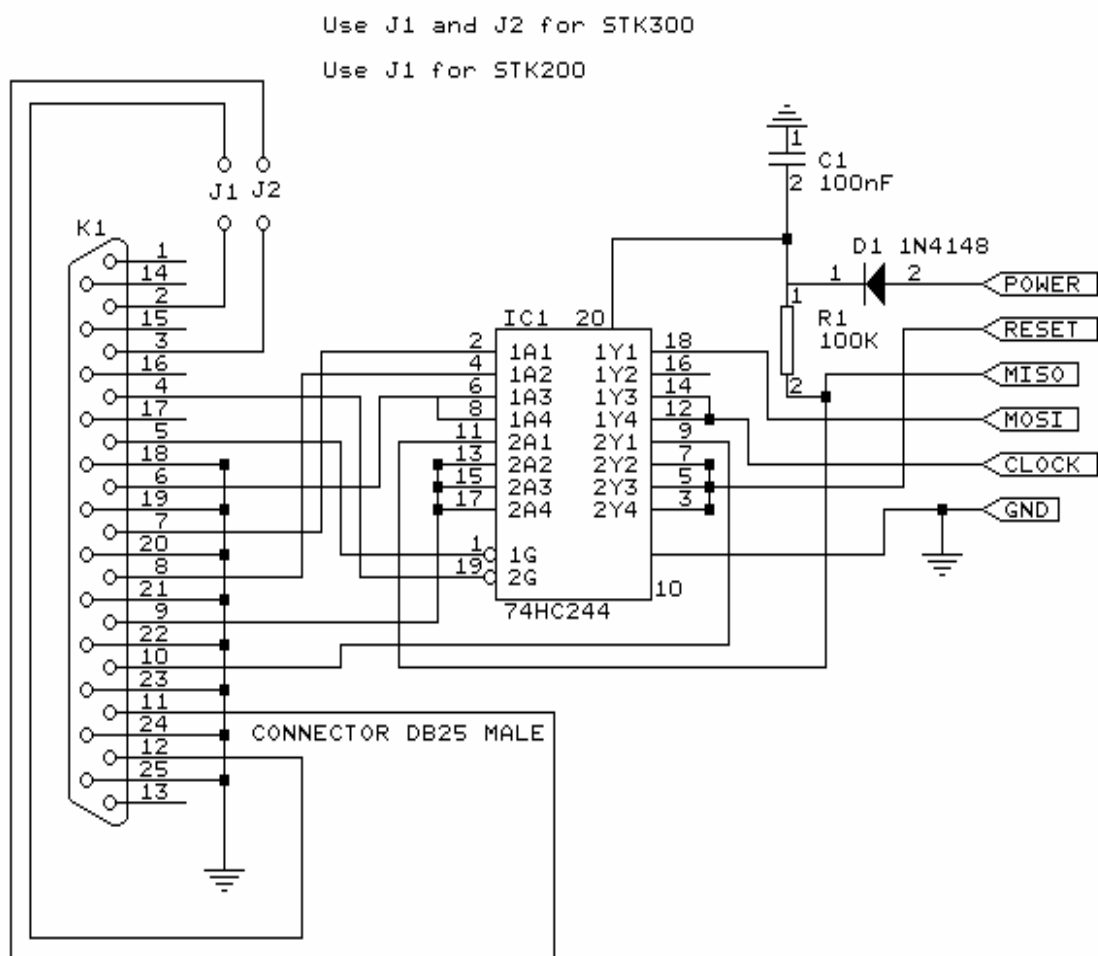


Figura 2.7.

En la figura 2.7. muestra que el 74HC244 está conectado directamente al conector DB25 que es una parte adicional del circuito que utilizamos en el momento de programar.

El buffer 74HC244 tiene dos intenciones que son :

- & Protege el computador en caso de un sobre voltaje este de desconecta.
- & Hace que el microcontrolador trabaje en SISTEM esto quiere decir que mientras está trabajando puedo programarlo sin tener la necesidad de desconectar el circuito.

2.3.2.2. Características del Circuito Programador

- & Se tiene que tomar en cuenta que no es necesario tener tres voltajes para programar y leer la memoria flash del AVR.
- & No necesitamos otro microcontrolador para programar el AVR.
- & No necesitamos 10 líneas de I/O para decidir como los vamos a utilizar.
- & No necesitamos quitar al AVR del circuito, antes de programarlo para hacerlo mas sencillo.
- & El cable no ha de ser más largo de 70cm.
- & El circuito de la figura 2.7. nos permite leer y escribir la información de la memoria flash y crearla en la EEPRON.

2.3.2.3. Líneas de Código

Esta interfaz trabaja consecutivamente y necesita tres líneas del código:

- SCK: Una señal de reloj que cambia los bits al escribirlos en la memoria en un registro de cambio interno, entonces los bits de salida cambian para ser leídos por otro registro de cambio interno.
- MOSI: La señal de datos que envía los bits de escritura en el AVR.
- MISO: La señal de datos que recibe los bits de lectura del AVR

Estos tres pines de señal son internamente conectados al programador solo si se cambia el RESET el pin pasa a ser cero.

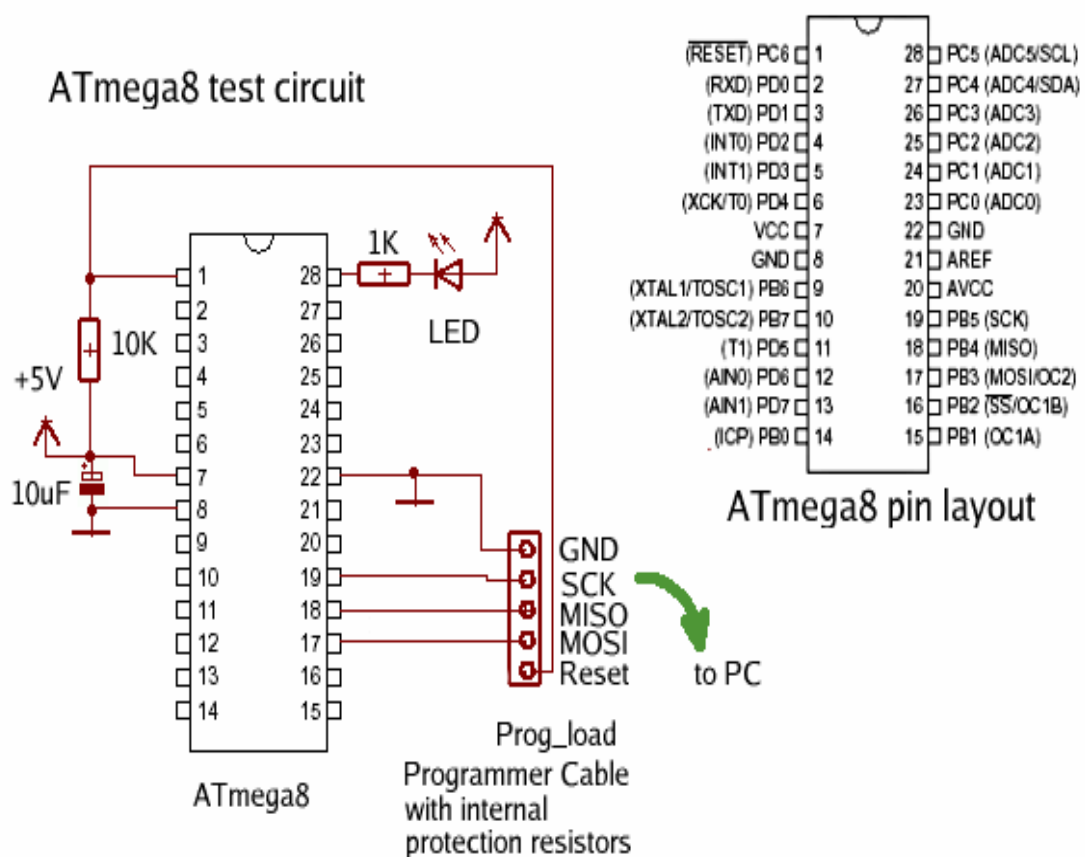


Figura 2.8.

Observamos que no es necesario un cristal. Actualmente el ATmega8 tiene incorporado un oscilador. Se puede usar este oscilador cuando no se necesite un alta precisión de reloj. El tipo de oscilador usado se puede modificar con el programa. Por defecto (selección de fábrica) esta altivo el oscilador interno de 8MHz.

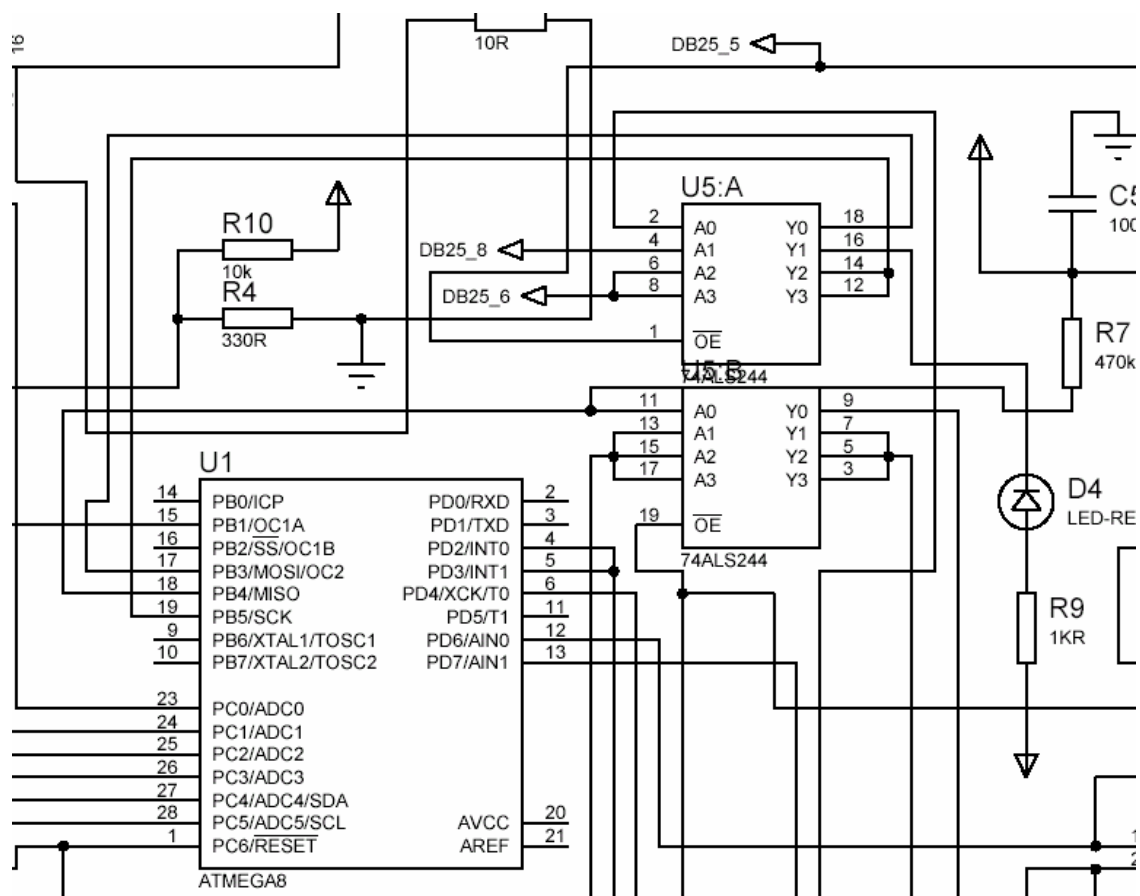


Figura 2.9.

Se puede observar que las tres líneas de código como son MOSI, MISO, SCK están conectadas directamente al microcontrolador Atmega8, viniendo estas del buffer 74ALS244. Se observan dos buffer pero en realidad es uno solo ya que el otro es el complemento acompañado de resistencias y condensadores que permiten su buen funcionamiento.

La necesidad de implementar el circuito programador en el proyecto es que éste trabaja en alta impedancia y no tenemos que apagar el circuito para cambiar algo del programa. Si se puede trabajar sin el programador pero sería necesario apagar y poner en el mismo pin algunas funciones.

2.3.2.4. Funcionamiento del Circuito Programador

Este es un programador del puerto de la impresora (paralelo) muy fiable. Si el puerto paralelo está desconectado de la interfaz y el conector izquierdo, los rendimientos de los circuitos integrados 244 cambiarán, mientras el microcontrolador produce un reset durante su funcionamiento.

EL resistor entre el pin 1 y 20 del integrado, y otro entre el pin 19 y 20, eliminará este problema. Entonces tendremos ALTO-Z en los rendimientos del integrado cuando el cable está desconectado (cuando esta conectado y no esta programando), para que se pueda usar el MOSI y otros pines de entrada y salida.

Este es el único hardware adicional que se utilizará para realizar la programación.



Figura 2.10.



Figura 2.11.

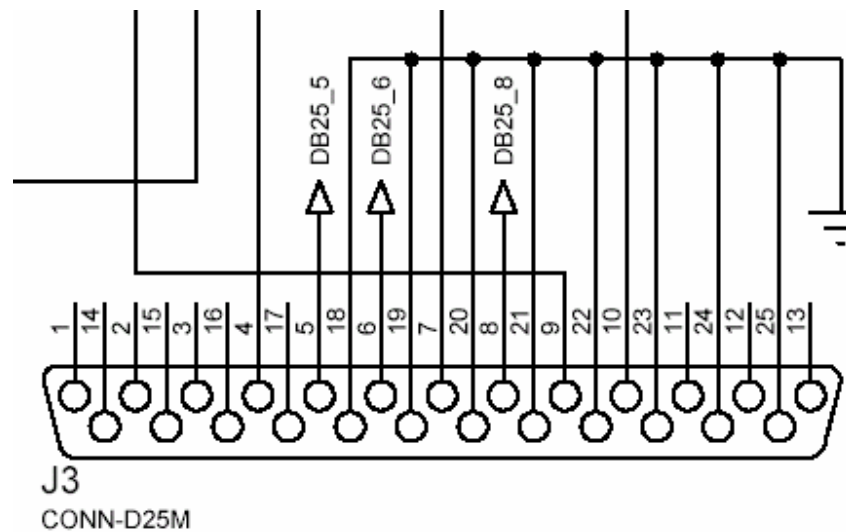


Figura 2.12.

2.3.3. PROGRAMACIÓN DEL LCD

Para programar el LCD 16 x 2 (2 líneas por 16 caracteres) explicaremos como están asignados los pines y como se configura en el compilador Bascom AVR

Este tipo de LCD consta de 14 pines de los cuales 8 son de entrada y salida de datos, 3 sirven para manejar las comunicaciones en el bus de datos, 2 son para la polarización y 1 sirve para manejar el contraste.

Pin	Simb.	Descripción
1	GND	Tierra de alimentación GND
2	Vcc	Alimentación de +5Vcc
3	Vo	Ajuste del contraste del cristal líquido
4	RS	Selección del registro control / datos RS=0L entrada de una instrucción; RS=1L entrada de un dato
5	R/W	Lectura / escritura en LCD R/W=0L escritura; R/W=1L lectura
6	E	Habilitación E=0L módulo desconectado; E=1L módulo conectado
7	D0	Bit menos significativo (bus de datos bidireccional)
14	D7	Bit mas significativo (bus de datos bidireccional)
15	A	Alimentación del backlight (+5Vcc)
16	K	Tierra GND del backlight

Cuadro 2.1.

Los pines del 8 al 13 del LCD 16 x 2 van sueltos.

2.3.3.1. Circuito de conexión LCD

A continuación en la figura 2.13. se muestra como esta conectado el LCD al microcontrolador Atmega8L.

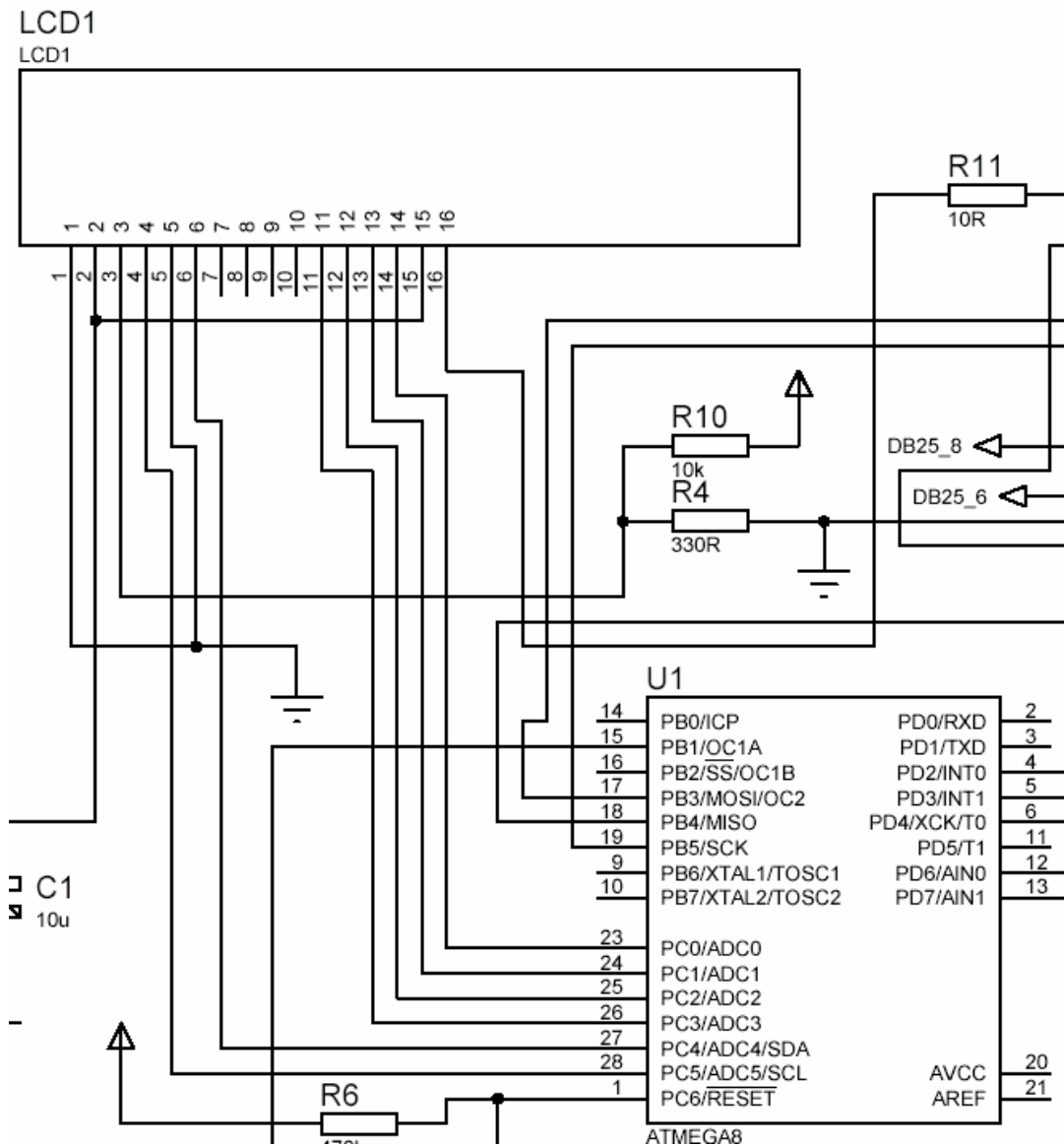


Figura 2.13.

Se observa que utilizamos el Pórtico C para la conexión del LCD y el Pórtico B para el circuito grabador del Microcontrolador Atmega8.

El circuito esquemático fue diseñado en el simulador Proteus 6 Demonstration (ISIS 6 Demo) ya que simula en tiempo real, lo encuentro más sencillo en su utilización y a la vez se puede realizar las pistas en el ISIS 6 Demo al igual que se pueden observar la ubicación de los elementos.

En la figura 2.14. se muestra como se configura el LCD 16X2 en el compilador BASCOM-AVR.

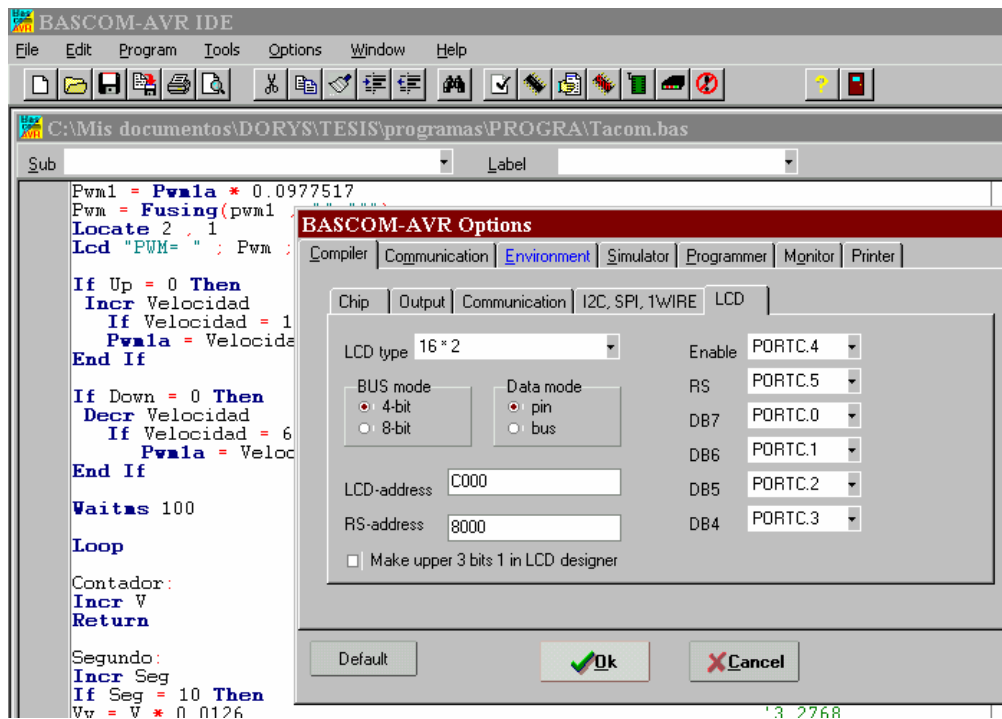


Figura 2.14.

Hemos utilizado el Portb para la asignación de pines del LCD 16 x 2 así:

ENABLE = PORC.4

RS = PORTC.5

DB7 = PORTC.0

DB6 = PORTC.1

DB5 = PORTC.2

DB4 = PORTC.3

2.3.4. CORTE Y SATURACIÓN

2.3.4.1. Introducción

El transistor es un amplificador de corriente, esto quiere decir que si le introducimos una cantidad de corriente por una de sus patillas (base), él entregará por otra (emisor) , una cantidad mayor a ésta, en un factor que se llama amplificación. Este factor se llama β (beta) y es un dato propio de cada transistor. Entonces:

- I_c (corriente que pasa por la patilla colector) es igual a β (factor de amplificación) por I_b (corriente que pasa por la patilla base).

$$- I_c = \beta * I_b$$

- I_e (corriente que pasa por la patilla emisor) es del mismo valor que I_c , sólo que, la corriente en un caso entra al transistor y en el otro caso sale de el, o viceversa.

Según la fórmula anterior las corrientes no dependen del voltaje que alimenta el circuito (V_{cc}), pero en la realidad si lo hace y la corriente I_b cambia ligeramente cuando se cambia V_{cc} . Ver figura 2.15.

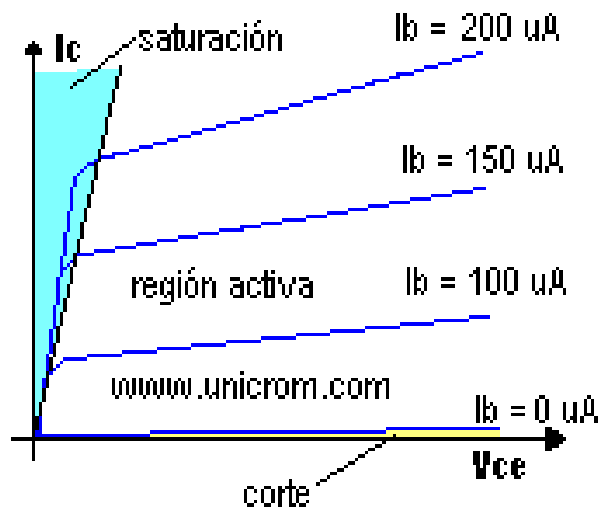


Figura 2.15

En el segundo gráfico las corrientes de base (I_b) son ejemplos para poder entender que a más corriente la curva es más alta.

En el proyecto se detalla el circuito esquemático del transistor con sus respectivas corrientes.

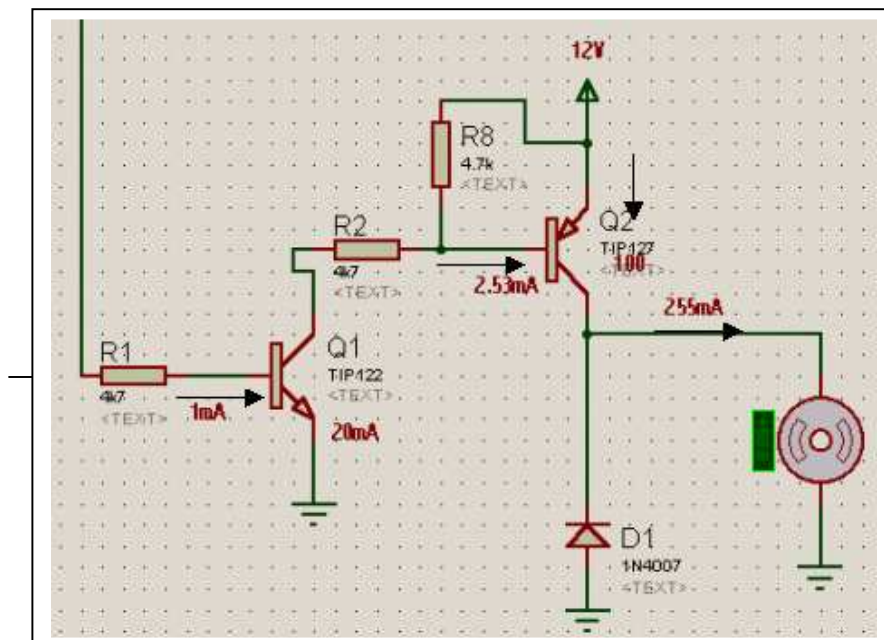


Figura 2.16.

2.3.4.2. Regiones operativas del transistor.

& **Región de corte:** Un transistor está en corte cuando:

corriente de colector = corriente de emisor = 0, ($I_c = I_e = 0$)

En este caso el voltaje entre el colector y el emisor del transistor es el voltaje de alimentación del circuito. (como no hay corriente circulando, no hay caída de voltaje, *ver Ley de Ohm*). Este caso normalmente se presenta cuando la corriente de base = 0 ($I_b = 0$)

& **Región de saturación:** Un transistor está saturado cuando:

corriente de colector = corriente de emisor = corriente máxima, ($I_c = I_e = I_{\text{máxima}}$)

En este caso la magnitud de la corriente depende del voltaje de alimentación del circuito y de las resistencias conectadas en el colector o el emisor o en ambos, *ver ley de Ohm*. Este caso normalmente se presenta cuando la corriente de base es lo suficientemente grande como para inducir una corriente de colector β veces más grande. (recordar que $I_c = \beta * I_b$).

& **Región activa:** Cuando un transistor no está ni en su región de saturación ni en la región de corte entonces está en una región intermedia, la región activa. En esta región la corriente de colector (I_c) depende principalmente de la corriente de base (I_b), de β (ganancia de corriente de un amplificador, es un dato del fabricante) y de las resistencias que hayan conectadas en el colector y emisor). Esta región es la más importante si lo que se desea es utilizar el transistor como un amplificador.

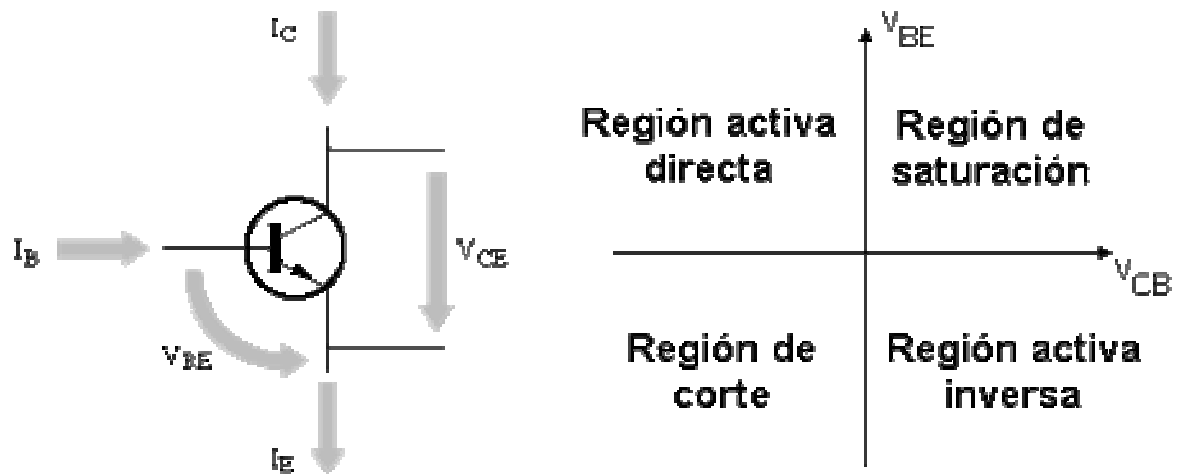


Figura 2.17.

2.3.4.3. Diferencia entre transistor bipolar y fet

La diferencia entre un transistor bipolar y un transistor unipolar o FET es el modo de actuación sobre el terminal de control. En el transistor bipolar hay que inyectar una corriente de base para regular la corriente de colector, mientras que en el FET el control se hace mediante la aplicación de una tensión entre puerta y fuente. Esta diferencia viene determinada por la estructura interna de ambos dispositivos, que son substancialmente distintas.

Es una característica común, sin embargo, el hecho de que la potencia que consume el terminal de control (base o puerta) es siempre más pequeña que la potencia manejada en los otros dos terminales.

En resumen, destacamos tres cosas fundamentales:

- & En un transistor bipolar I_B controla la magnitud de I_C .
- & En un FET, la tensión V_{GS} controla la corriente I_D .
- & En ambos casos, con una potencia pequeña puede controlarse otra bastante mayor.

2.3.5. IMPLEMENTACION DEL SOFTWARE

2.3.5.1. Compilador Basic

El compilador Basic es el más indicado para programar micros de la familia AVR de Atmel ya que con esta herramienta reducirá los tiempos de desarrollo de forma drástica y la programación de microcontroladores adquiere otra dimensión.

BASCOM-AVR© es un COMPILADOR BASIC en Windows para la familia de microcontroladores AVR.

Está diseñado para trabajar en W95/W98/NT/XP.

2.3.5.2. Características

2.3.5.2.1. BASIC estructurado con etiquetas.

- & Programación estructurada con IF-THEN-ELSE-END IF, DO-LOOP, WHILE-WEND, SELECT- CASE.
- & Código máquina compilado, mucho más rápido que los interpretados.
- & Nombres de variables y etiquetas largos, hasta 32 caracteres de longitud.
- & Variables Bit, Byte, Integer, Word, Long, Single y String.
- & Los programas compilados trabajan con todos los microcontroladores AVR que tienen memoria RAM interna.
- & Mezcla Assembler y Basic en el mismo fuente. (Control total del chip)
- & Puede generar librerías.
- & Comandos especiales para displays-LCD, chips I2C y chips 1WIRE , teclado PC , teclado de matriz, recepción RC5, UARTsoftware y SPI.

- & Variables locales, funciones de usuario, soporte de librería.
- & Emulador de terminal integrado con opción de descarga.
- & Simulador integrado para test.
- & Programador SPI integrado (nota de aplicación AVR910.ASM).
- & Programador STK200 y STK300 integrados. Soporta un programador de ejemplo de bajo costo. (Puede construirse en 10 minutos)
- & Editor con resalte de las instrucciones.
- & Contexto de ayuda sensitivo..
- & Trabaja perfectamente con los productos de Digital Micro Devices basados en AVR.
- & La versión DEMO compila 2KB de código. Adecuada para el AT2313.
- & Ayuda muy completa y documentada con ejemplos gráficos y demos. (Inglés)

2.3.5.2.2. Decisión Y Estructuras

IF, THEN, ELSE, ELSEIF, END IF, DO, LOOP, WHILE, WEND, UNTIL, EXIT DO, EXIT WHILE, FOR, NEXT, TO, DOWNT0, STEP, EXIT FOR, ON .. GOTO/GOSUB, SELECT, CASE.

2.3.5.2.3. Entrada Y Salida

PRINT, INPUT, INKEY, PRINT, INPUTHEX, LCD, UPPERLINE, LOWERLINE, DISPLAY ON/OFF, CURSOR ON/OFF/BLINK/NOBLINK, HOME, LOCATE, SHIFTLCD LEFT/RIGHT, SHIFTCURSOR LEFT/RIGHT, CLS, DEFLCDCHAR, WAITKEY, INPUTBIN, PRINTBIN, OPEN, CLOSE, DEBOUNCE, SHIFTTIN, SHIFTTOUT, GETATKBD

2.3.5.2.4. Funciones Numéricas

AND, OR, XOR, INC, DEC, MOD, NOT, ABS, BCD, LOG, EXP, SQR, SIN, COS, TAN, EXP.

2.3.5.2.5. I2C

I2CSTART, I2CSTOP, I2CWBYTE, I2CRBYTE, I2CSEND and I2CRECEIVE.

2.3.5.2.6. 1WIRE

1WRITE, 1WREAD, 1WRESET, 1WIRECOUNT, 1WSEARCHFIRST, 1WSEARCHNEXT.

2.3.5.2.7. SPI

SPIINIT, SPIIN, SPIOUT, SPIMOVE.

2.3.5.2.8. Interrupciones

ON INT0/INT1/TIMER0/TIMER1/SERIAL, RETURN, ENABLE, DISABLE, COUNTER_x, CAPTURE_x, INTERRUPTS, CONFIG, START, LOAD.

2.3.5.2.9. Manipulacion Bit

SET, RESET, ROTATE, SHIFT, BITWAIT, TOGGLE.

2.3.5.2.10. Variables

DIM, BIT, BYTE, INTEGER, WORD, LONG, SINGLE, STRING, DEFBIT, DEFBYTE, DEFINT, DEFWORD.

2.3.5.2.11. Varias

REM, ', SWAP, END, STOP, CONST, DELAY, WAIT, WAITMS, GOTO, GOSUB, POWERDOWN, IDLE, DECLARE, CALL, SUB, END SUB, MAKEDEC, MAKEBCD, INP, OUT, ALIAS, DIM, ERASE, DATA, READ, RESTORE, INCR, DECR, PEEK, POKE, CPEEK, FUNCTION, READMAGCARD, SPC.

2.3.5.2.12. Directivas del compilador

\$INCLUDE, \$BAUD and \$CRYSTAL, \$SERIALINPUT, \$SERIALOUTPUT, \$RAMSIZE, \$RAMSTART, \$DEFAULT XRAM, \$ASM-\$END ASM, \$LCD, \$EXTERNAL, \$LIB.

2.3.5.2.13. Manipulación de cadenas

STRING, SPACE, LEFT, RIGHT, MID, VAL, HEXVAL, LEN, STR, HEX, LTRIM, RTRIM, TRIM, LCASE, UCASE, FORMAT, FUSING, INSTR.

2.3.5.3. Como Hacer Un Programa

Para hacer un programa sólo se debe seguir 4 pasos:

- & Escribir el programa en BASIC.
- & Compilar a código máquina binario (ejecución rápida).
- & Testee el resultado con el simulador integrado
- & Programamos el chip con uno de los programadores integrados.

2.3.5.4. Programa

El programa puede escribirse en un editor MDI de código en color cómodo.

Además de las características habituales el editor soporta Undo, Redo, Bookmarks y bloques.

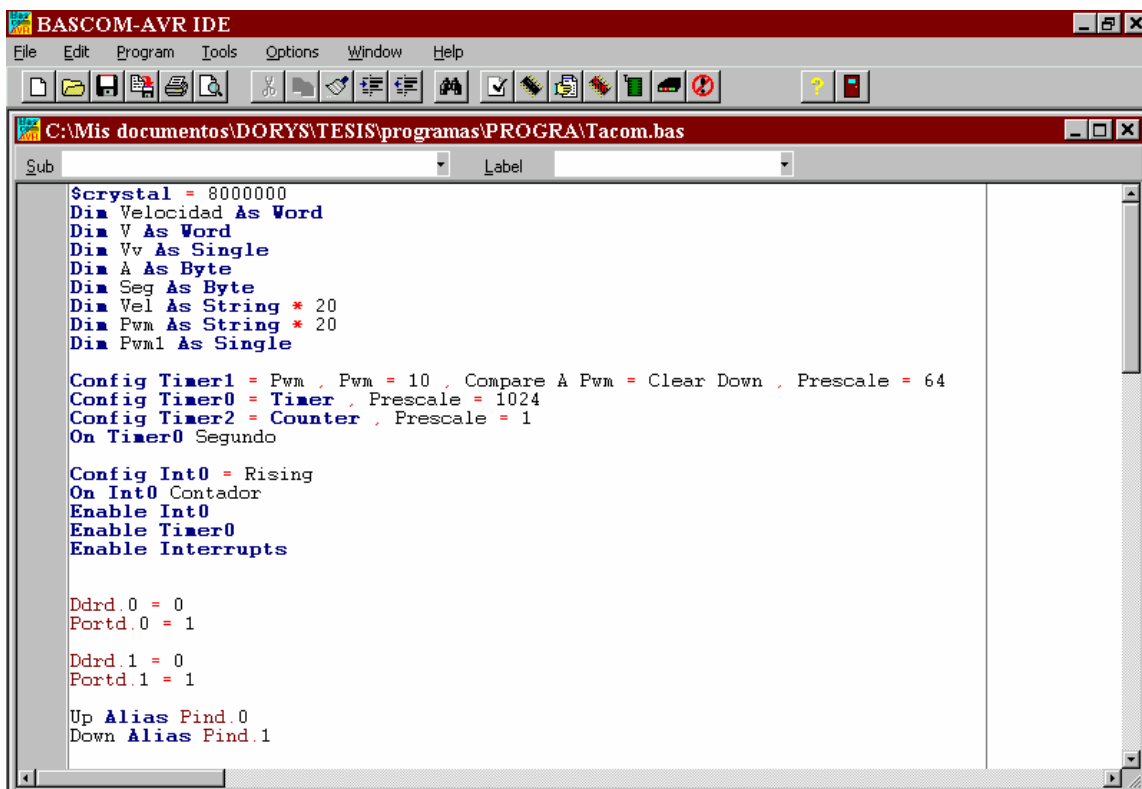
Nuestro programa utiliza tres Timers, cada uno con una diferente función:

- & **T0 = Timer Presescale** . El prescale es un divisor de frecuencia para obtener la frecuencia deseada tomando en cuenta el cristal interno que es de 8MHz cuyos parámetros son 1, 8, 64, 256 o 1024.

& **T1 = PWM** (regula la Velocidad del motor según nuestra manipulación y el PWM está ligado directamente al T1)

& **T2 = Contador** (Cuenta los pulsos enviados por el encoder) por tanto trabaja con pulsos externos ya que podemos usarlo de dos maneras:

- Como Timer es interno
- Como Contador es externo



```

Sub
Label

$crystal = 8000000
Dim Velocidad As Word
Dim V As Word
Dim Vv As Single
Dim A As Byte
Dim Seg As Byte
Dim Vel As String * 20
Dim Pwm As String * 20
Dim Pwm1 As Single

Config Timer1 = Pwm , Pwm = 10 , Compare A Pwm = Clear Down , Prescale = 64
Config Timer0 = Timer , Prescale = 1024
Config Timer2 = Counter , Prescale = 1
On Timer0 Segundo

Config Int0 = Rising
On Int0 Contador
Enable Int0
Enable Timer0
Enable Interrupts

Ddrd.0 = 0
Portd.0 = 1

Ddrd.1 = 0
Portd.1 = 1

Up Alias Pind.0
Down Alias Pind.1

```

Figura 2.18.

Con el simulador puedes testar el programa antes de grabar el uP.

Se puede marcar variables, ejecutar el programa paso a paso ó hasta una línea específica, también se puedes modificar registros y variables.

Para ver el valor de una variable, basta colocar el puntero del ratón encima.

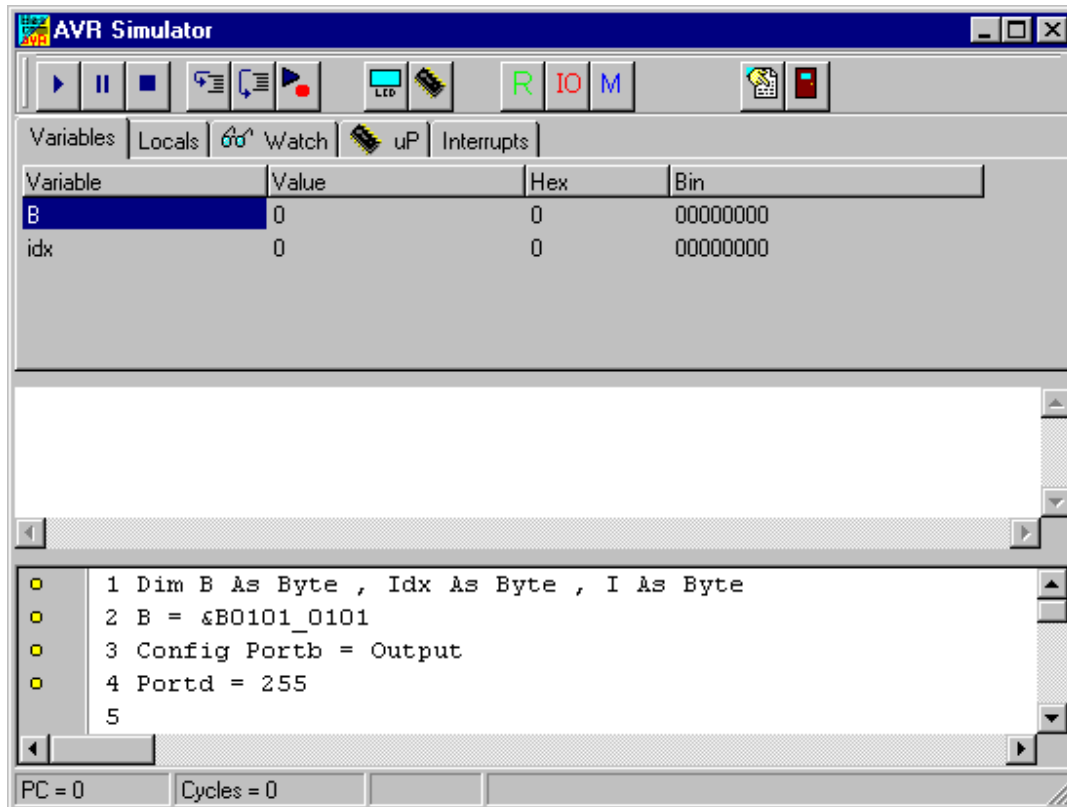


Figura 2.19

Múltiples opciones para visualizar los datos del uP con el simulador.

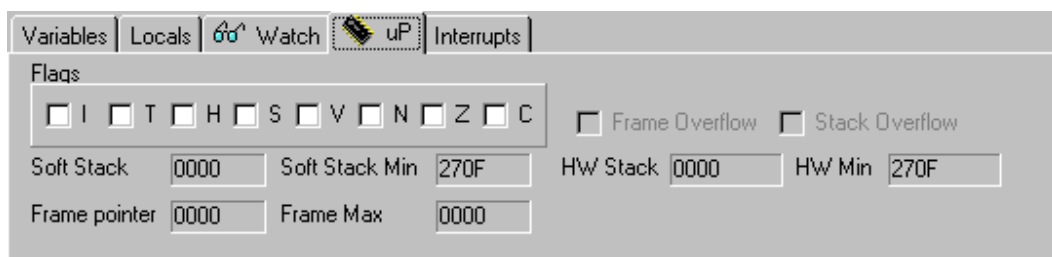


Figura 2.20.

Una característica interesante es el emulador de hardware del display LCD y los ports.

El emulador del LCD puede emular los caracteres diseñados a medida.

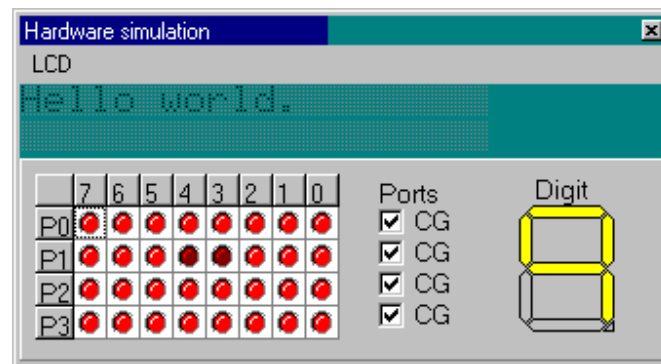


Figura 2.21.

Cuando se esté seguro del funcionamiento del programa es el momento de programar el chip usando cualquiera de los drivers de programador soportados.

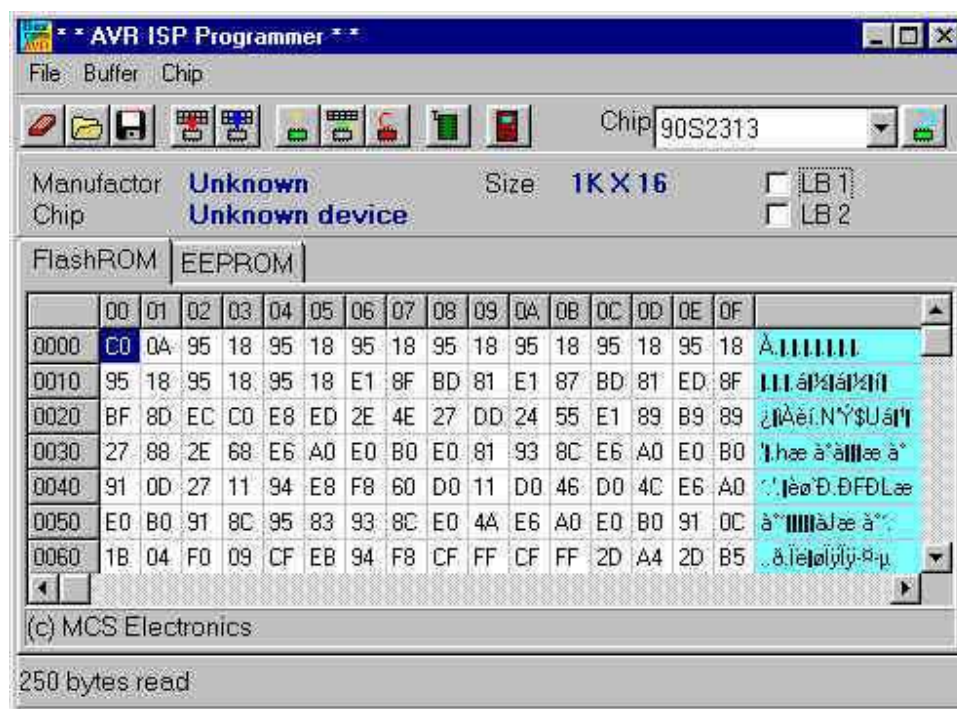


Figura 2.22.

2.3.6. DISTRIBUCIÓN DE LOS ELEMENTOS

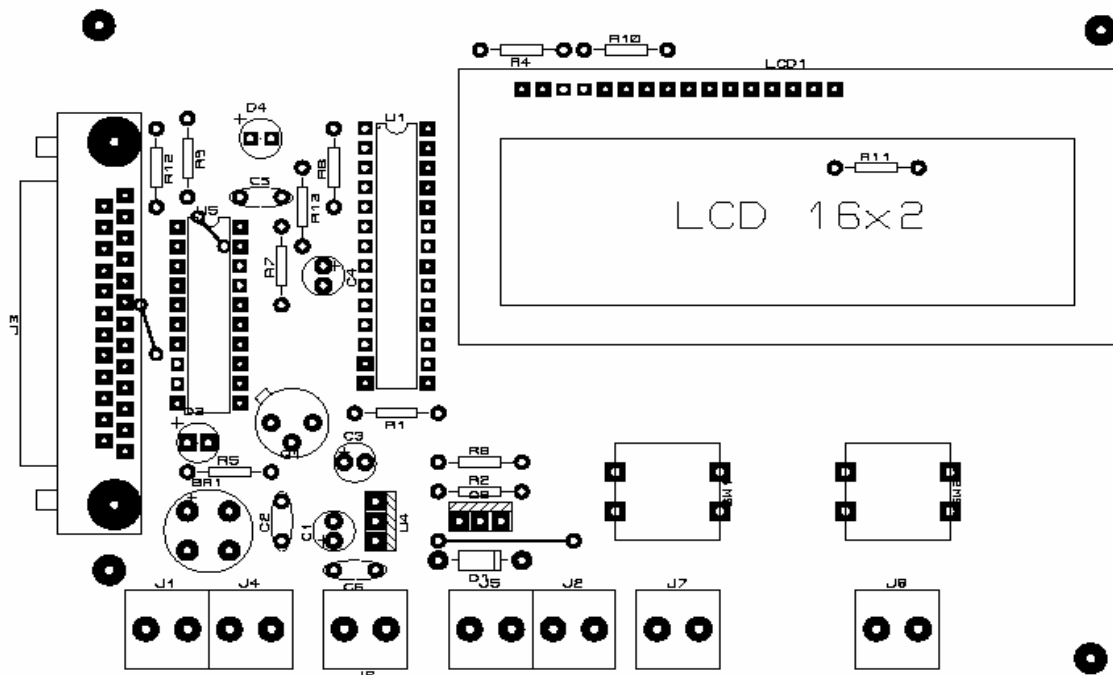


Figura 2.23.

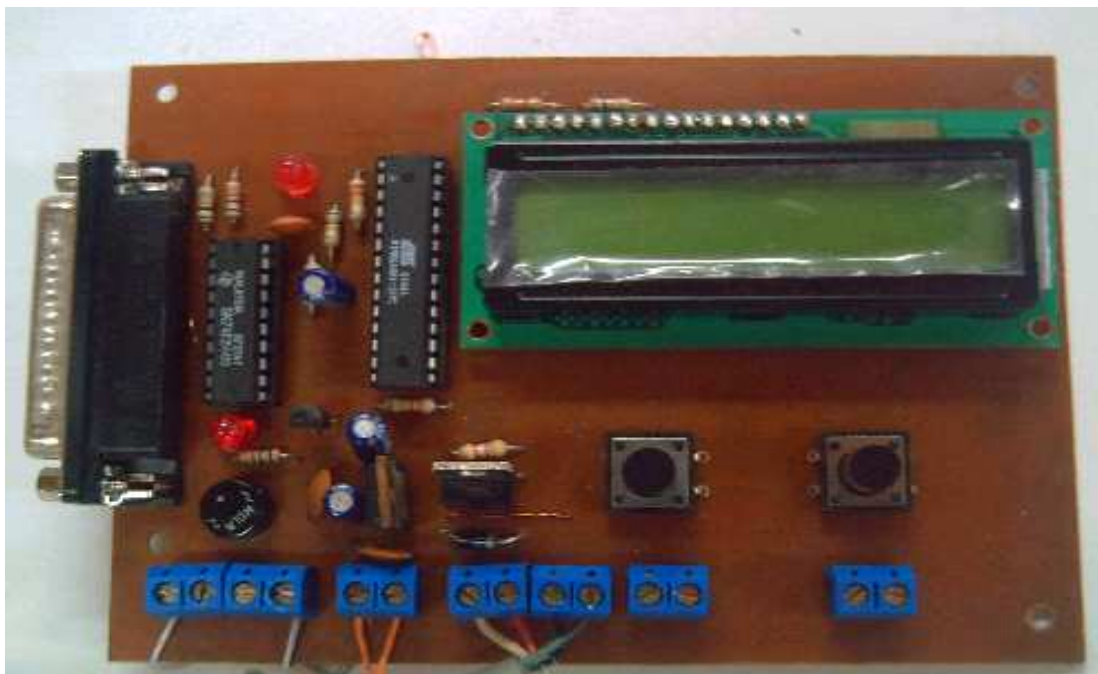


Figura 2.24.

2.3.7. CIRCUITO IMPRESO

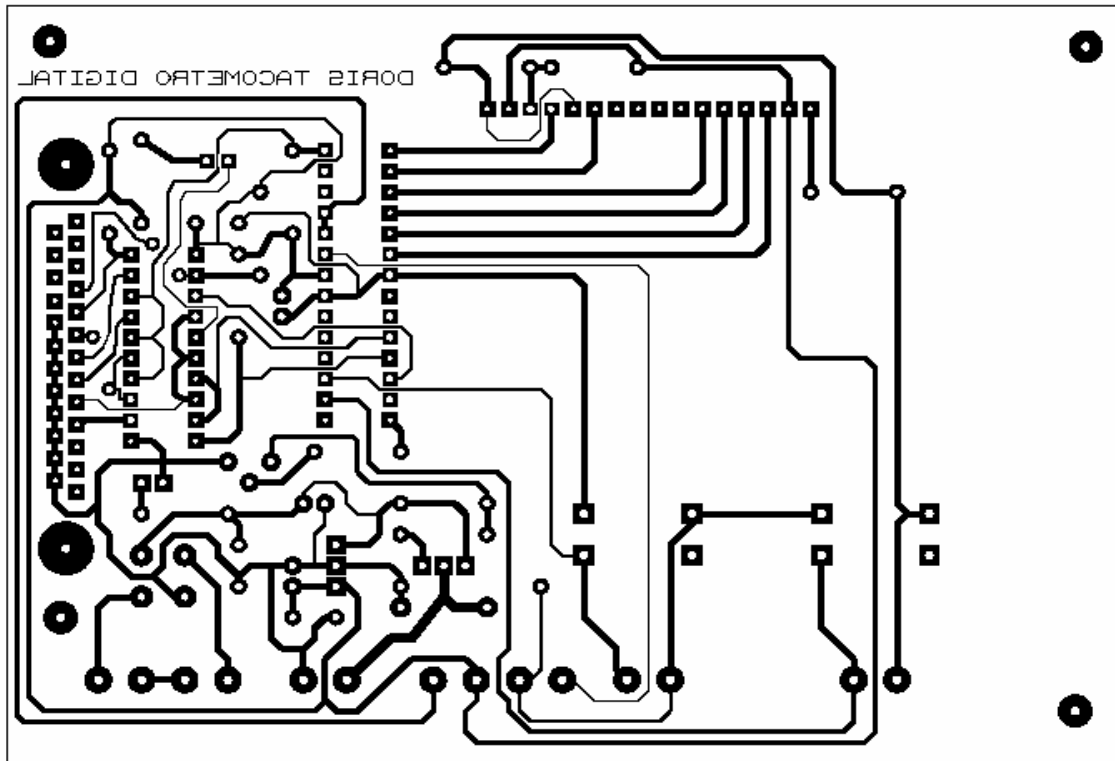


Figura 2.25.

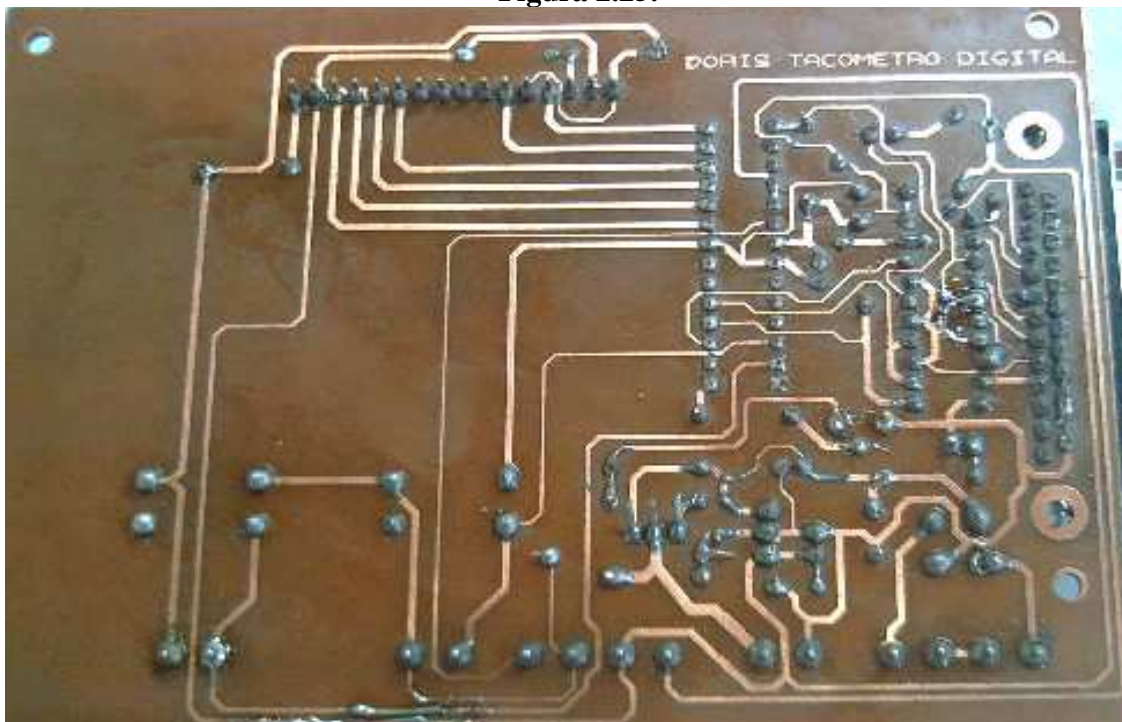


Figura 2.26.

2.3.8. LISTA DE MATERIALES UTILIZADOS

12 Resistors

<u>Quantity:</u>	<u>References</u>	<u>Value</u>
3	R1, R6, R7	470k
2	R2, R8	4.7k
1	R4	330R
2	R5, R9	1kR
3	R10, R12, R13	10k
1	R11	10R

6 Capacitors

<u>Quantity:</u>	<u>References</u>	<u>Value</u>
3	C1, C3, C4	10u
3	C2, C5, C6	100n

3 Integrated Circuits

<u>Quantity:</u>	<u>References</u>	<u>Value</u>
1	U1	ATMEGA8
1	U4	7805
1	U5	74ALS244

2 Transistors

<u>Quantity:</u>	<u>References</u>	<u>Value</u>
1	Q1	2N3904
1	Q2	TIP127

3 Diodes

<u>Quantity:</u>	<u>References</u>	<u>Value</u>
1	D1	1N4007
2	D2, D4	LED-RED

12 Miscellaneous

<u>Quantity:</u>	<u>References</u>	<u>Value</u>
------------------	-------------------	--------------

1	BR1	W04
1	J1	vcc+
6	J2, J4-J8	TBLOCK-I2
1	J3	CONN-D25M
1	LCD1	LCD1
2	SW1, SW2	SW4

Cuadro 2.2

CAPITULO III: ANÁLISIS DE RESULTADOS

3.1. COMPORTAMIENTO DEL TACOMETRO DIGITAL EN LA ETAPA EXPERIMENTAL

Al utilizar un motor de CC los resultados no fueron óptimos ya que al momento de enviar la señal al micro este enviaba señales con picos y quemaba el LCD que es muy sensible es por eso que pensamos en la posibilidad de optar acoplar el motor utilizando una fuente adicional para su funcionamiento, luego hicimos las pruebas con un encoder y observamos que las señales que enviaba al contador eran más precisas y decidimos utilizarlo ya que la diferencia fue notable.

El funcionamiento del Tacómetro Digital resultó satisfactoriamente con la utilización del encoder.

El indicador como es el LCD fue probado de la siguiente manera:

El LCD, como se requería, nos brinda la ayuda de visualizar algunos parámetros tales como: porcentaje en la variación de velocidad del encoder desde 0% (mínima) hasta 100% máxima y la velocidad del encoder en RPM (Revolución por minuto).

Para comprobar que el tacómetro Digital medía en tiempos reales se realizó una relación de frecuencias utilizando un multímetro que mida frecuencias.

La siguiente tabla muestra los valores experimentales que indica el LCD del Tacómetro los cuales se tomarán para la realización de su respectiva curva.

VOLTAJE MOTOR	RPM(TACOMETRO)	PWM(TACOMETRO)
0	0	0
2.77	500	8.02
5.34	1000	12.32
7.92	1500	26.1
9.51	1850	100

Cuadro 3.1.

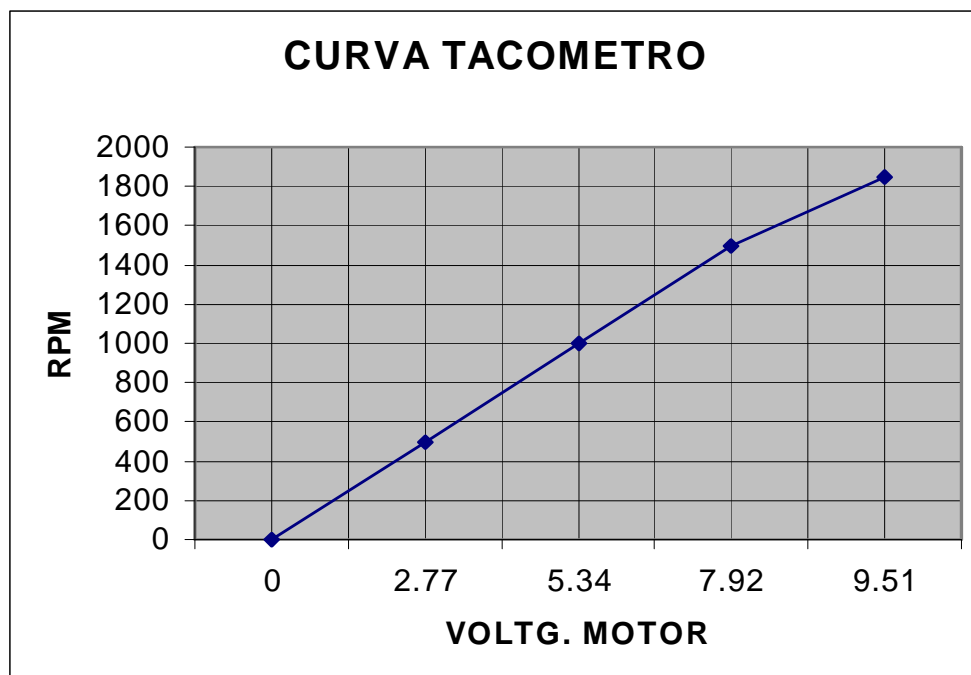


Figura 3.1.

En la tabla siguiente se pueden observar los valores medidos con el multímetro y su respectiva curva. Para obtener los datos de la columna RPM (MULT) se hicieron los siguientes cálculos:

Como la frecuencia está dada por Hz (1/s) se transformó a:

$$\frac{1}{\text{minuto}} * \# \text{ pulsos} = \text{RPM}$$

Ejemplo: 2.16 Hz entonces:

$$216 * \left(\frac{1}{\text{seg}} \right) * \frac{60 \text{ seg}}{1 \text{ minuto}} * 250 \text{ pulsos} = 518.4 \text{ RPM}$$

VOLTAJE MOTOR	FRECUENCIA(KHz) (MULT)	RPM (MULT)
0	0	0
2.77	2.16	518.4
5.34	4	960
7.92	6.4	1536
9.51	7.37	1768.8

Cuadro 3.2.

Podemos observar que tomamos como referencia el Voltaje del motor para poder comparar las curvas.

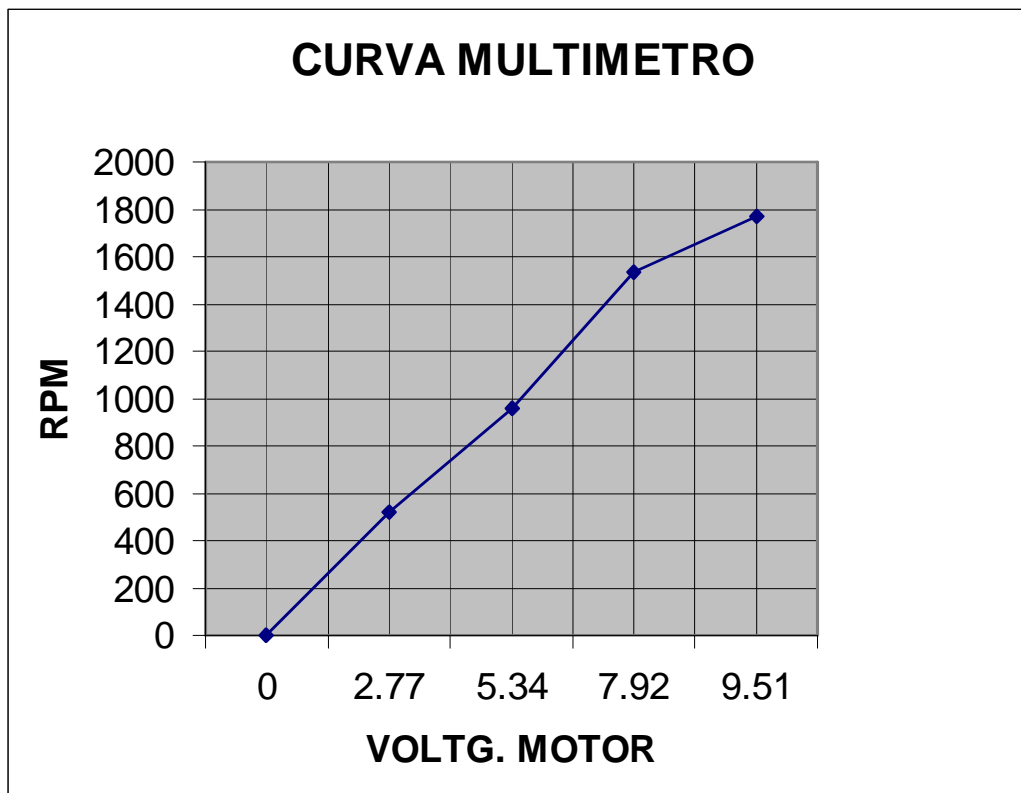


Figura 3.2.

El objetivo de tomar valores tanto del tacómetro como del multímetro es para comparar las curvas correspondientes y observar el margen de error del tacómetro, también se comprueba que el Tacómetro Digital mide en tiempos reales y que el error es mínimo.

A continuación se muestra la curva del Tacómetro y la curva del Multímetro

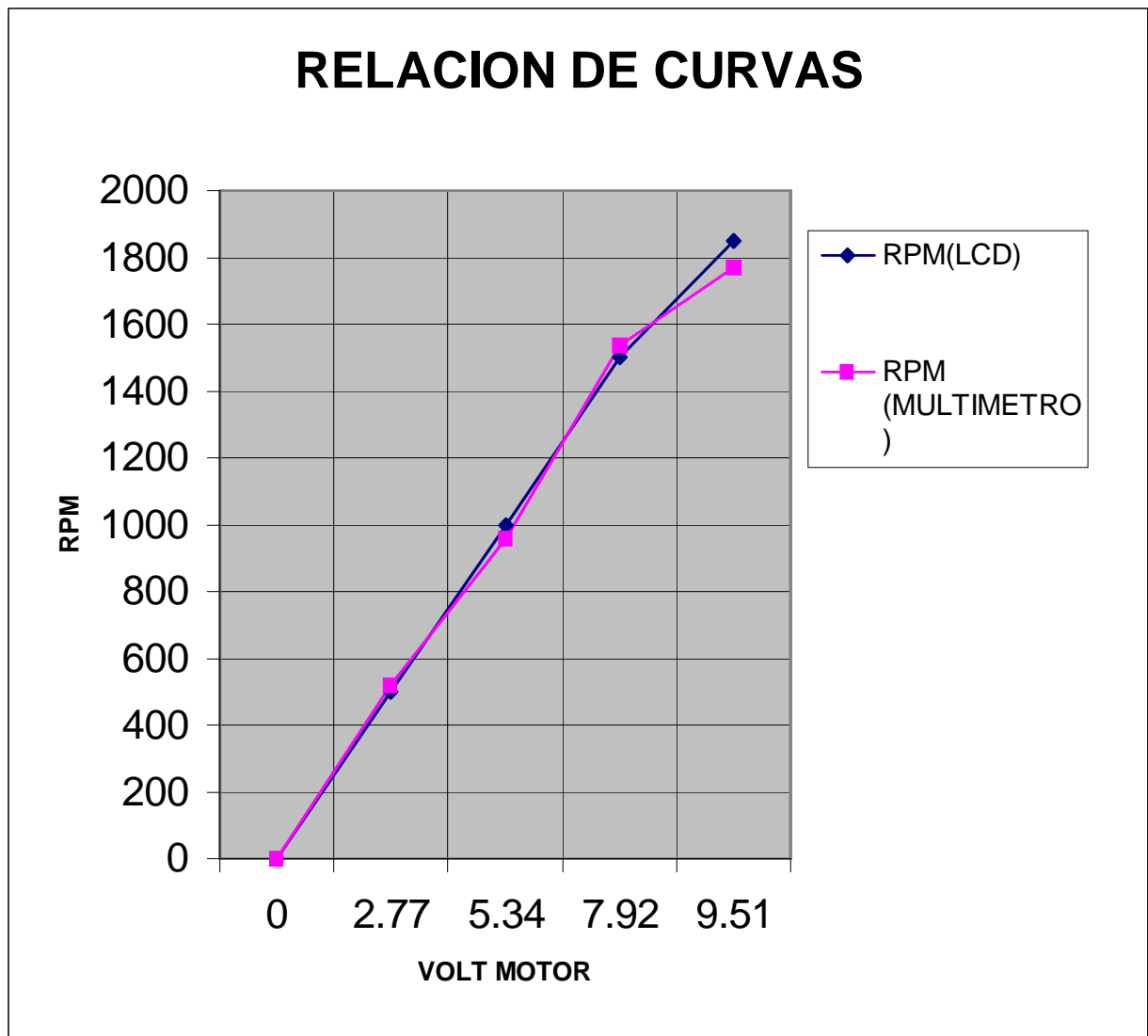


Figura 3.3.

3.2 CALCULO DEL ERROR

Error Absoluto

El error absoluto es la diferencia entre el valor leído (multímetro) y el valor convencionalmente verdadero correspondiente (tacómetro).

Error absoluto = valor leído - valor convencionalmente verdadero

Error Relativo

El error relativo es la razón del error absoluto y el valor convencionalmente verdadero

Error relativo = Error absoluto / valor convencionalmente verdadero

Como el error absoluto es igual a la lectura menos el valor convencionalmente verdadero, entonces:

Error relativo = (valor leído - valor real) / valor real

Con frecuencia, el error relativo se expresa como un porcentaje de error, multiplicándolo por cien:

Porcentaje de error = Error relativo * 100%

RPM (TACÓMETRO)	RPM (MULTIMETRO)	ERROR ABSOLUTO	ERROR RELATIVO	PORCENT ERROR(%)
0.00	0.00	0.00	0.00	0.00
500.00	518.40	18.40	0.04	3.68
1,000.00	960.00	-40.00	-0.04	4.00
1,500.00	1,536.00	36.00	0.02	2.40
1,850.00	1,768.80	-81.20	-0.04	4.39

Cuadro 3.3.

3.3. FUNCIONAMIENTO DEL DISPOSITIVO

Se realizó una relación entre frecuencias utilizando un multímetro con frecuencímetro para comparar y comprobar que el Tacómetro Digital precisa la velocidad del encoder en tiempo real.

El funcionamiento del dispositivo fue lo esperado con lo cual se hubo cumplido mi propósito en la realización del proyecto ya que el error que presenta el tacómetro es de 3.61% que se puede decir es mínimo y este es producido por el ruido del encoder.

3.3. CONCLUSIONES Y RECOMENDACIONES

3.3.1. CONCLUSIONES

Siendo el objetivo de este Proyecto de Titulación, el diseño y construcción de un Tacómetro Digital como material didáctico para los estudiantes de la ESFOT de la EPN y demostrar su buen funcionamiento, podemos concluir lo siguiente:

1. El sistema cumple con las especificaciones técnicas indicadas en el diseño, esto es: obtener una frecuencia adecuada para que el tacómetro pueda manejar valores reales de tiempo (timer0), variación de la velocidad del encoder (timer1 = PWM), recepción de los pulsos enviados por el encoder (Contador = Timer2) para luego observar la velocidad del servomotor en el LCD.
2. El dispositivo está diseñado para controlar un servomotor , sin embargo el circuito y el programa pueden ser modificados para que maneje otro motor ya sea en CC o en AC utilizando un opto acoplador.
3. El proyecto cumple con las exigencias de seguridad, ya que ningún estudiante podrá lastimarse con la utilización del Tacómetro Digital puesto que se va a medir la velocidad de un motor pequeño.
4. Es de fácil uso porque tenemos dos pulsadores con los cuales incrementamos o decrementamos la velocidad sin tener contacto con el circuito interno ni con el motor.
5. En las curvas se observa un error en cuanto a lo medido con lo experimental ya que el encoder al momento de variar la velocidad ya sea incrementando o decrementando tiene que salir de la inercia, es decir debe estabilizarse para la lectura en LCD.

3.3.2. RECOMENDACIONES

1. La utilización del compilador Basic es mas rápido y sencillo porque reduce los tiempos de desarrollo de forma drástica y la programación de microcontroladores adquiere otra dimensión.
2. El empleo de un microcontrolador en cualquier proyecto reduce el tamaño de complejidad del mismo, no se requieren muchos componentes para hacerlo funcionar y es muy preciso en los tiempos de operación para el cual esté programado
3. Es necesario utilizar un opto acoplador para aislar la etapa de control y la de potencia para evitar daños a los circuitos integrados.
4. Para que se establezca la lectura de la velocidad en el LCD es recomendable filtrar el encoder para disminuir el error.
5. Utilizar una fuente adicional si se quiere medir otro encoder ya que este Tacómetro Digital está diseñado precisamente para el servomotor indicado en el proyecto.

BIBLIOGRAFÍA

- [1] BULLIET, L. J.(1967). *Servomechanisms*, Addison Wesley.

- [2] PARDO, F. Boluda J. (1999). *VHDL lenguaje para síntesis y modelado de circuitos*. Alfaomega.

- [3] COOPER, W.D. y MELLFRIICK, A.D. ,(1990). *Instrumentación electrónica moderna y técnicas de medición*. Prentice.

- [4] CONSIDINE, Douglas M.,(1993). *Process Industrial Instruments Controls Hand Book*, Mc. Graw-Hill.

- [5] GROOVER M and WEIS M.,(1999). *Industrial Robotics Technology, Programming and Applications*.

- [6] BISHOP R. and DORF R.,(1998). *Modern Control Systems*, Addison Wesley.

- [7] WILLIAMS, Arthur. (1994). *Manual de Circuitos integrados: selección, diseño y aplicaciones*. 1ra edición. MacGraw-Hill. México.

DIRECCIONES ELECTRÓNICAS

<http://mx.geocities.com/migsantiagov/avr/index.htm#iniciopag>

<http://www.tldp.org/linuxfocus/Castellano/November2004/article352.shtml>

<http://www.dmd.es/bascom-a.htm>

<http://www.urany.net/teoria.htm>

<http://www.controldemovimiento.com/index.ewp?section=obra>

<http://www.dccia.ua.es/dccia/inf/ asignaturas/ROB/optativos/Sensores/internos.html>

<http://www.monografias.com/trabajos7/inba/inba2.shtml>

<http://www.monografias.com/trabajos7/case/case.shtml>

<http://www.comunidadelectronicos.com/articulos/electroliticos.htm>

http://www.solomantenimiento.com/m_transistores.htm

<http://www.monografias.com/trabajos11/trans/trans.shtml>

http://www.unicrom.com/tut_darlington.asp

CONTENIDO

<i>INTRODUCCIÓN</i>	<i>1</i>
<i>CAPITULO I: FUNDAMENTOS TEÓRICOS Y CONCEPTUALES</i>	<i>2</i>
1.1. ATMEGA8L	2
1.1.1. CARACTERISTICAS DEL ATMEGA8L:	3
1.1.2. Distribución de pines del ATmega8L	4
1.1.3. Memorias del Atmega8L	5
1.1.3.1. Programación de la Memoria	6
1.1.3.2. AVR CPU General Purpose Working Registers	6
1.1.3.3. Mapa de la Memoria de Datos	7
1.2. Motores C.C.	7
1.2.1. Introducción	7
1.2.2. Principio básico de funcionamiento	8
1.3. MOTOR PASO A PASO	9
1.3.1. INTRODUCCIÓN	9
1.3.2. PRINCIPIO BÁSICO DE FUNCIONAMIENTO	10
1.3.3. TIPOS DE MOTOR PASO A PASO	11
1.4. SERVO MOTORES	12
1.4.1. INTRODUCCIÓN	12
1.4.2. PRINCIPIO DE FUNCIONAMIENTO	13
1.4.3. TIPOS DE ENCODER	15
1.4.3.1. Encoders Incrementales	15
1.4.3.2. Encoder absoluto	17
1.5. COMPONENTES ELECTRÓNICOS UTILIZADOS	18
1.5.1. RESISTENCIAS	18
1.5.2. CONDENSADOR	19
1.5.2.1. Condensador no polarizado Condensador variable	20
1.5.2.2. Capacitores Electrolíticos	20
1.5.2.3. Los capacitores cerámicos	20
1.5.3. TRANSISTORES	21
1.5.3.1. Transistor de efecto campo.	23

	75
1.5.3.2. Darlington _____	25
1.5.4. EL DIODO _____	26
1.5.4.1. Tipos de diodos _____	27
1.5.4.1.1. Diodos Metal-Semiconductor _____	27
1.5.4.1.2. Diodos Led _____	28
1.5.4.1.3. Display _____	29
1.6. EL LCD _____	29
1.6.1. INTRODUCCION _____	29
1.6.2. TIPOS DE LCD _____	30
1.6.3. MEMORIAS DEL LCD _____	30
1.6.4. ASIGNACIÓN DE PINES _____	31
<i>CAPITULO II: IMPLEMENTACIÓN DEL DISPOSITIVO</i> _____	33
2.1. DIAGRAMA DE BLOQUES DEL SISTEMA: _____	33
2.2. DISEÑO Y CONSTRUCCIÓN DEL DISPOSITIVO _____	33
2.3. PARTES DE LA ESTRUCTURA DEL TACOMETRO DIGITAL	
_____	34
2.3.1. PWM _____	35
2.3.1.1. Introducción _____	35
2.3.2. PROGRAMACIÓN DEL MICROCONTROLADOR _____	39
2.3.2.1. Programador del Atmega8L _____	39
2.3.2.2. Características del Circuito Programador _____	40
2.3.2.3. Líneas de Código _____	40
2.3.2.4. Funcionamiento del Circuito Programador _____	43
2.3.3. PROGRAMACIÓN DEL LCD _____	44
2.3.3.1. Circuito de conexión LCD _____	46
2.3.4. CORTE Y SATURACIÓN _____	48
2.3.4.1. Introducción _____	48
2.3.4.2. Regiones operativas del transistor. _____	50
2.3.4.3. Diferencia entre transistor bipolar y fet _____	51
2.3.5. IMPLEMENTACION DEL SOFTWARE _____	52
2.3.5.1. Compilador Basic _____	52
2.3.5.2. Características _____	52
2.3.5.2.1. BASIC estructurado con etiquetas. _____	52
2.3.5.2.2. Decisión Y Estructuras _____	53
2.3.5.2.3. Entrada Y Salida _____	53
2.3.5.2.4. Funciones Numéricas _____	54

	76
2.3.5.2.5. I2C _____	54
2.3.5.2.6. 1WIRE _____	54
2.3.5.2.7. SPI _____	54
2.3.5.2.8. Interrupciones _____	54
2.3.5.2.9. Manipulacion Bit _____	54
2.3.5.2.10. Variables _____	54
2.3.5.2.11. Varias _____	54
2.3.5.2.12. Directivas del compilador _____	55
2.3.5.2.13. Manipulación de cadenas _____	55
2.3.5.3. Como Hacer Un Programa _____	55
2.3.5.4. Programa _____	55
2.3.6. DISTRIBUCIÓN DE LOS ELEMENTOS _____	59
2.3.7. CIRCUITO IMPRESO _____	60
2.3.8. LISTA DE MATERIALES UTILIZADOS _____	61
<i>CAPITULO III: ANÁLISIS DE RESULTADOS</i> _____	62
3.1. COMPORTAMIENTO DEL TACOMETRO DIGITAL EN LA ETAPA EXPERIMENTAL _____	63
3.2 CALCULO DEL ERROR _____	67
3.3. FUNCIONAMIENTO DEL DISPOSITIVO _____	69
3.3. CONCLUSIONES Y RECOMENDACIONES _____	70
3.3.1. CONCLUSIONES _____	70
3.3.2. RECOMENDACIONES _____	71
<i>BIBLIOGRAFÍA</i> _____	72
<i>DIRECCIONES ELECTRÓNICAS</i> _____	73
ANEXO 1: Programa y subrutinas en lenguaje Basic	
ANEXO 2: Circuito esquemático del Tacómetro Digital	
ANEXO 3: Ubicación de los elementos en la placa del Tacómetro Digital	

