

ESCUELA POLITÉCNICA NACIONAL

ESCUELA DE FORMACIÓN TECNOLÓGICA

**CONSTRUCCIÓN DE UN MÓDULO PARA VERIFICAR
LOS PROGRAMAS A IMPLEMENTAR EN
MICROCONTROLADORES AT89C51, QUE INCLUYE
COMUNICACIÓN SERIAL CON PROTOCOLO RS – 232**

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

**OBANDO PANCHI DIEGO SEBASTIÁN
TUPIZA SILVA DANNY JAVIER**

DIRECTOR: ING. ALCÍVAR COSTALES

Quito, Febrero 2006

DECLARACIÓN

Nosotros, Diego Sebastián Obando Panchi y Danny Javier Tupiza Silva, declaramos bajo juramento que el trabajo aquí descrito es de nuestra auditoría; que no ha sido previamente presentada para ningún grado o calificación profesional; y, que hemos consultado las referencias bibliográficas que se incluyen en este documento.

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

Diego Sebastián Obando Panchi

Danny Javier Tupiza Silva

CERTIFICACIÓN

Certifico que el presente trabajo fue desarrollado por Diego Sebastián Obando Panchi y Danny Javier Tupiza Silva, bajo mi supervisión.

Ing. Alcívar Costales
DIRECTOR DEL PROYECTO

AGRADECIMIENTO

A Dios por darme la sabiduría de culminar este proyecto, y una meta más de mi vida mis estudios de Tecnólogo.

A mi familia que con su infinito amor, bondad, afecto, cariño, paciencia, y apoyo incondicional me lo han brindado siempre.

A mis profesores, compañeros y amigos con los que compartí gratos y malos momentos.

A la Poli cuyas aulas son testigos de algunos de mis sueños que hoy lo veo realizado.

Diego Sebastián Obando Panchi
Autor

DEDICATORIA

A Dios por guiarme por un buen camino para culminar una meta más de mi vida.

A mi papá Miguel, mi mamá Mariana y mi hermano Santiago que con su infinito amor, sus ejemplos de humildad, responsabilidad, sinceridad, bondad, sencillez, honestidad, honradez me han brindado y sabido guiar siempre en todo momento.

A mi madrina Alita, me dio su apoyo incondicional.

A todos, que Dios les bendiga.

Diego Sebastián Obando Panchi

Autor

AGRADECIMIENTO

*A todas las personas que directa o indirectamente me han dado sus conocimientos en el transcurso de mi carrera, profesores y compañeros.
Y aquellos quienes estuvieron conmigo en las buenas y en las malas en mi vida como estudiante, mis amigos.*

Yo, Danny Javier Tupiza Silva

DEDICATORIA

A todas las personas que han estado conmigo en el transcurso de todo mi vida estudiantil sobre todo a mi familia: Padres, Abuela, y Hermana, que en base de su sacrificio me han podido brindar el apoyo necesario para la culminación de este proyecto y carrera, y también a todos los que estudian, enseñan o admiran la prodigiosa ciencia de la electrónica y las telecomunicaciones.

Yo, Danny Javier Tupiza Silva

CONTENIDO

**CONSTRUCCIÓN DE UN MÓDULO PARA VERIFICAR LOS
PROGRAMAS A IMPLEMENTAR EN
MICROCONTROLADORES AT89C51, QUE INCLUYE
COMUNICACIÓN SERIAL CON PROTOCOLO RS – 232**

RESUMEN	12
INTRODUCCIÓN	13

CAPITULO 1: ANÁLISIS DEL MÓDULO EXISTENTE Y A CONSTRUIR

1.1.- ANÁLISIS DEL MÓDULO EXISTENTE EN EL LABORATORIO PARA MANEJO DE MICROCONTROLADORES AT89C51	14
1.2.- ANÁLISIS Y PROPUESTA DEL NUEVO MÓDULO	16

CAPITULO 2: CONSTRUCCIÓN DEL NUEVO MÓDULO

2.1.- CONSTRUCCIÓN Y ENSAMBLAJE DEL NUEVO MÓDULO PARA VERIFICAR LOS PROGRAMAS A IMPLEMENTAR EN MICROCONTROLADORES AT89C51 QUE INCLUYE PROTOCOLO RS - 232	18
2.1.1- PRINCIPIO DE FUNCIONAMIENTO DE LOS ELEMENTOS A UTILIZAR.	18

2.1.1.1- Microcontrolador AT89C51	18
<i>2.1.1.1.1- Características generales</i>	19
<i>2.1.1.1.2.- Distribución y descripción de pines</i>	20
<i>2.1.1.1.3.- Descripción de los espacios de memoria</i>	24
<i>2.1.1.1.4.- Localidades de los registros de funciones especiales</i>	27
<i>2.1.1.1.5.- Pórtico serial</i>	35
<i>2.1.1.1.6.- Modos de direccionamiento</i>	39
2.1.1.2.- Diodo led	44
2.1.1.3.- Matriz de led`s	46
2.1.1.4.- Display`s de siete segmentos	46
2.1.1.5.- LCD	47
<i>2.1.1.5.1.- Principio de funcionamiento</i>	47
<i>2.1.1.5.2.- Descripción de pines</i>	49
<i>2.1.1.5.3.- Juego de instrucciones</i>	49
2.1.1.6.- Motores paso a paso	52
<i>2.1.1.6.1- Principio de funcionamiento</i>	53
<i>2.1.1.6.2- Tipos de motores paso a paso</i>	55
<i>2.1.1.6.3.- Secuencias para manejar motores paso a paso bipolares</i>	56
<i>2.1.1.6.4.- Secuencias para manejar motores paso a paso unipolares</i>	57
2.1.1.7.- Teclado matricial hexadecimal.	60
<i>2.1.1.7.1.- Introducción</i>	60
<i>2.1.1.7.2.- Codificador de teclado 74C922</i>	61
2.1.2.- TRANSMISIÓN DE DATOS	62
2.1.2.1. - Introducción	62
2.1.2.2. - Conversor TTL - RS232	65
2.1.3- CONSTRUCCIÓN Y CONEXIONES DE CADA TARJETA	66
2.1.3.1.- Tarjeta principal	66
2.1.3.2.- Tarjeta motor paso a paso	73
2.1.3.3.- Tarjeta matriz de led`s	75

2.1.3.4.- Tarjeta LCD	77
2.1.3.5.- Tarjeta display de siete segmentos	79
2.1.4.- GRÁFICAS DE LAS TARJETAS	82
2.1.4.1.- Tarjeta principal	82
2.1.4.2.- Tarjeta motor paso a paso	84
2.1.4.3.- Tarjeta matriz de led's	86
2.1.4.4.- Tarjeta LCD	88
2.1.4.5.- Tarjeta display de siete segmentos	90
2.1.4.6.- Tarjeta MAX 232	92
2.1.4.7.- Diagrama del módulo	94
2.1.5.- GRÁFICA DEL CHASIS	94
2.2.- DESARROLLO DEL SOFTWARE EN VISUAL BASIC PARA CONTROL DE FUNCIONAMIENTO DEL MOTOR PASO A PASO POR MEDIO DE PROTOCOLO RS – 232	95
2.3.- DESARROLLO DE PROGRAMAS PEQUEÑOS PARA PRUEBAS EN EL MÓDULO	98
2.3.1.- PROGRAMA PARA CONTROL DEL MOTOR PASO A PASO	98
2.3.2.- PROGRAMA PARA CONTROL DE LA MATRIZ DE LED'S	100
2.3.3.- PROGRAMA PARA CONTROL DEL LCD	103
2.3.4.-PROGRAMA PARA CONTROL DEL DISPLAY DE SIETE SEGMENTOS	105
2.3.5.- PROGRAMA PARA CONTROL DEL TECLADO	107
2.3.6.- PROGRAMA PARA LA TRANSMISIÓN SERIAL	113

CAPITULO 3: PRUEBAS Y CALIBRACIONES

3.1.- PRUEBA Y CALIBRACIONES DEL NUEVO MÓDULO PARA MANEJO DE MICROCONTROLADORES AT89C51.	115
3.1.1.- PRUEBA Y CALIBRACIÓN DE LA TARJETA MADRE	115
3.1.2.- PRUEBA Y CALIBRACIÓN DE LA TARJETA MOTOR PASO A PASO	115
3.1.3.- PRUEBA Y CALIBRACIÓN DE LA TARJETA MATRIZ DE LED'S	116
3.1.4.- PRUEBA Y CALIBRACIÓN DE LA TARJETA LCD	116
3.1.5.- PRUEBA Y CALIBRACIÓN DE LA TARJETA DISPLAY DE 7 SEGMENTOS	117
CONCLUSIONES	118
RECOMENDACIONES	119
BIBLIOGRAFIA	120
ANEXOS	122
ANEXO 1: EL MICROCONTROLADOR AT89C51	123
ANEXO 2: 74C922	128
ANEXO 3: 74LS11	131
ANEXO 4: L293D	134
ANEXO 5: DIAGRAMA DEL MÓDULO	137
ANEXO 6: MANUAL DE USUARIO	139

RESUMEN

En el laboratorio de Microprocesadores de la ESFOT se requiere de módulos didácticos que permitan la verificación en tiempo real de la programación del microcontrolador AT89C51 elaborada por los estudiantes que están aprendiendo a programar. Por esta razón se ha planteado la posibilidad de diseñar y crear un Módulo Didáctico en el cual se tenga disponible el hardware necesario para probar la programación del microcontrolador AT89C51.

El hardware del Módulo Didáctico consta de cinco circuitos: el primero permite controlar un motor paso a paso, el segundo permite leer la programación en una matriz de led's, el tercero permite leer la programación en un LCD, el cuarto permite leer la programación en dos display's dobles y el quinto permite la comunicación serial mediante el protocolo RS-232 entre la PC y el módulo

Con este Módulo Didáctico se espera colaborar y ayudar al estudiante y al profesor en el proceso de aprendizaje de la programación del microcontrolador AT89C51.

INTRODUCCIÓN

El Módulo Didáctico diseñado y construido tiene la finalidad de ayudar al estudiante del Laboratorio de Microprocesadores que está aprendiendo a programar el microcontrolador AT89C51. Con este Módulo Didáctico el estudiante puede observar en tiempo real en los circuitos que se incluyen en este módulo el funcionamiento del programa creado por el mismo.

Además el estudiante no tendrá que preocuparse en realizar el hardware de su práctica de laboratorio, puesto que el Módulo Didáctico ya lo tiene hecho, y él solamente se enfocará en el software, es decir la programación del microcontrolador, ahorrándose tiempo y dinero.

CAPITULO 1

ANÁLISIS DEL MÓDULO EXISTENTE Y A CONSTRUIR

1.1.- ANÁLISIS DEL MÓDULO EXISTENTE EN EL LABORATORIO

El módulo existente en el laboratorio para manejo de microcontroladores AT89C51, consta de cuatro dispositivos de salida que son: una matriz de 7X5 led's, dos display's dobles de 7 segmentos, un LCD de 2 x16 y un motor paso a paso, y cada uno de los dispositivos es activado por 4 switchs independientemente y un dispositivo de entrada como es el teclado hexadecimal, los cuales son manejados por el microcontrolador.

Este lector del Microcontrolador AT89C51 ha sido utilizado durante algunos años, lo cual ha deteriorado sus condiciones dando lugar a, los siguientes daños:

Zócalo para colocación del Microcontrolador.- Palanca de sujeción dañada.

Switchs de reset para el Microcontrolador.- Rotos en la parte superior.

Motor paso a paso.- Sin funcionamiento.

Las malas condiciones del módulo obliga a los estudiantes a armar su propio hardware en un protoboard, perdiendo demasiado tiempo, que puede ser destinado a otras actividades.

Además este módulo carece de una conexión directa al computador, lo cual impide la realización de prácticas en conexión con una PC.

En el siguiente diagrama de bloques de la figura 1.1 se muestra como está constituido el módulo antiguo.

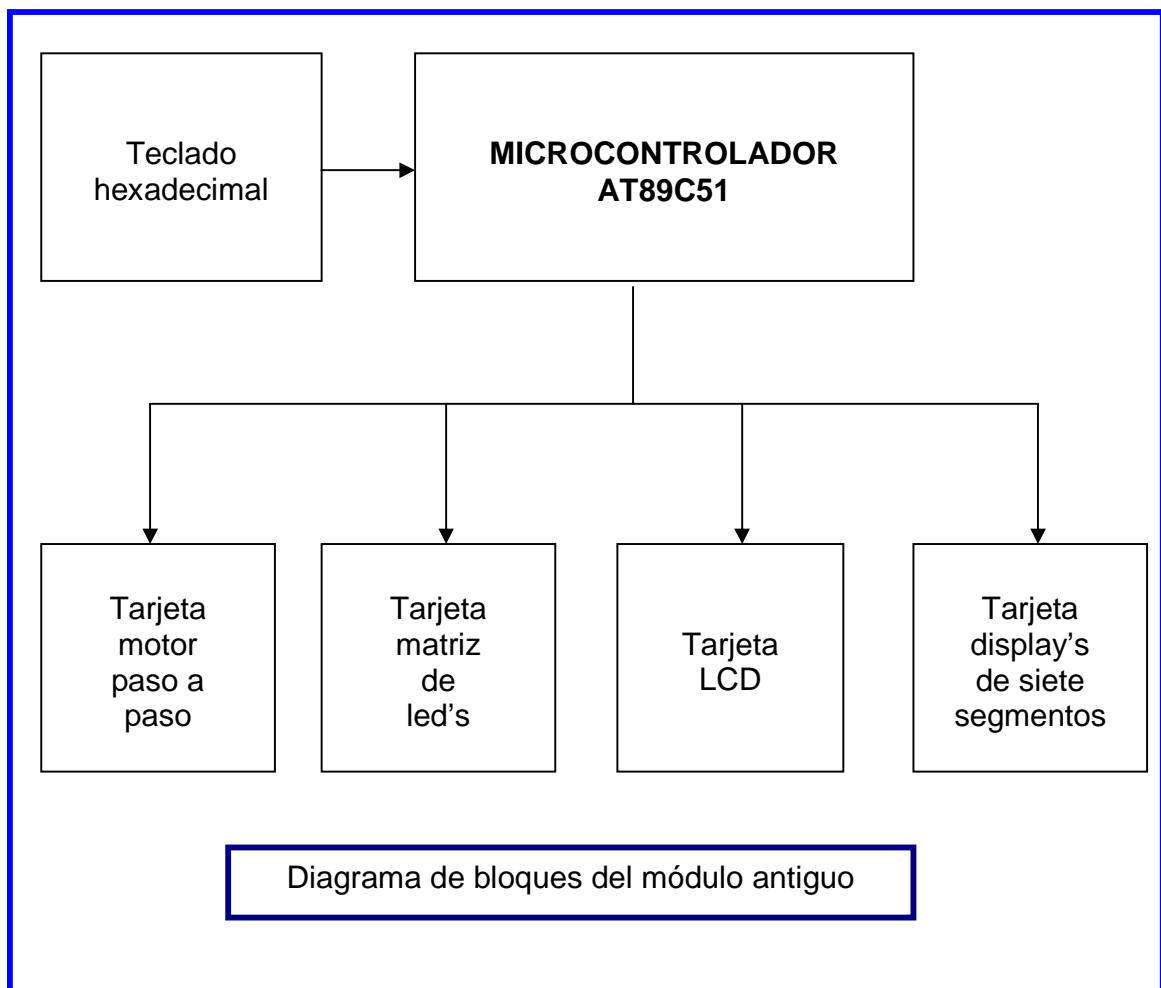


Figura 1.1

1.2.- ANÁLISIS Y PROPUESTA DEL NUEVO MÓDULO

El nuevo módulo para manejo de microcontroladores AT89C51, constará de dispositivos de entrada y salida; para la entrada se utilizará un teclado hexadecimal y la comunicación serial, para la salida se utilizará: una matriz de 7x5 led's, dos display's dobles de 7 segmentos, un LCD de 2x16 y un motor paso a paso bipolar, cada dispositivo controlado por un switch.

En este módulo se implementará la posibilidad de conexión al computador mediante comunicación serial usando el protocolo RS – 232, mediante el cual podremos controlar cualquiera de los cuatro dispositivos de salida desde la PC ya que el módulo anterior no dispone de esta comunicación serial.

En el objetivo de este proyecto, se propuso como ejemplo controlar desde la PC el motor paso a paso mediante un programa realizado en visual Basic.

Además se realizarán las pruebas de programación, con programas pequeños de cada uno de los dispositivos de salida.

Al término del proyecto se espera contribuir para que las practicas del Laboratorio de Microprocesadores, en el estudio del Microcontrolador AT89C51, se realicen de una manera eficiente y rápida.

En el siguiente diagrama de bloques de la figura 1.2 se muestra cómo está constituido el nuevo módulo.

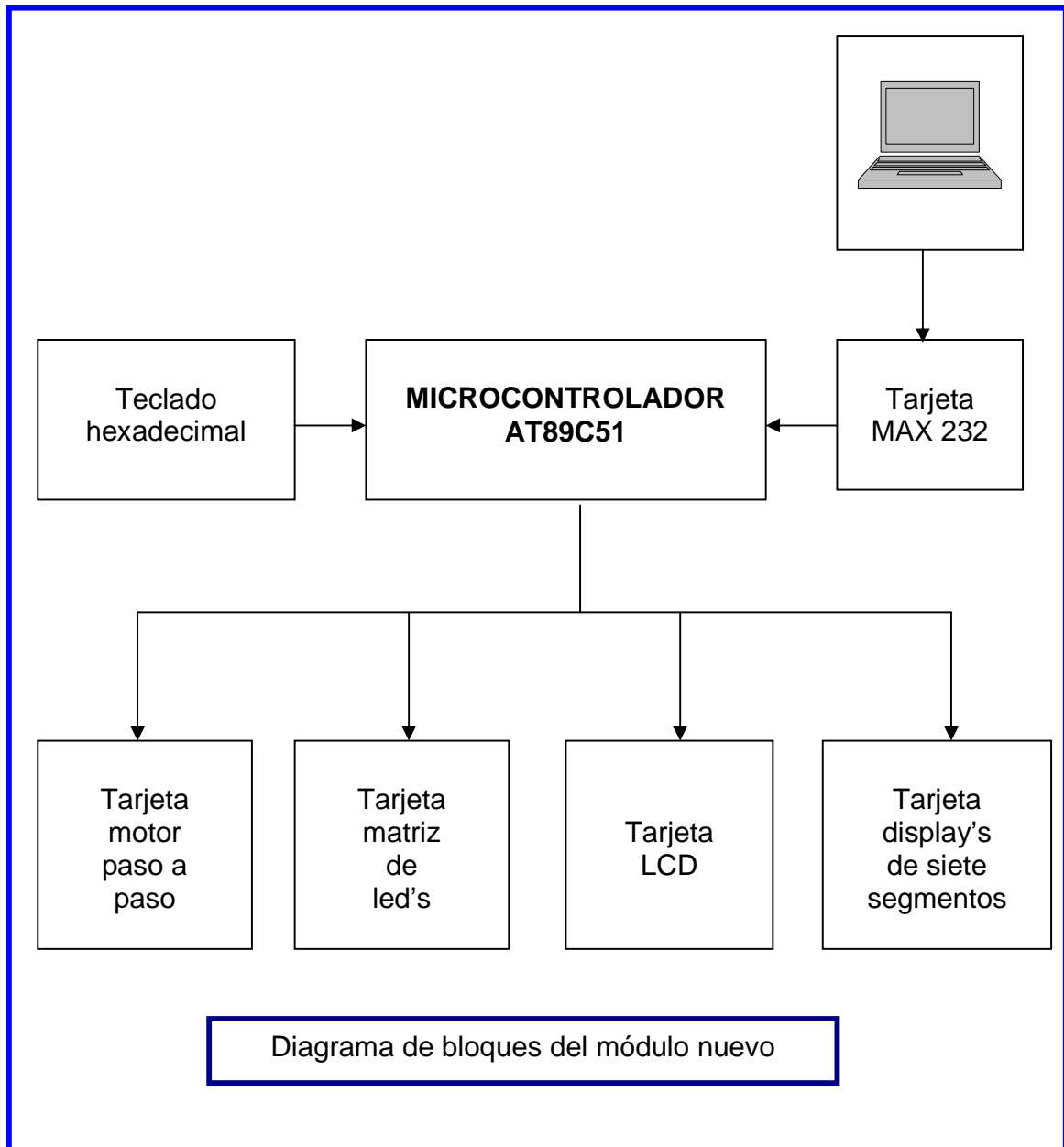


Figura 1.2

CAPITULO 2

CONSTRUCCIÓN DEL NUEVO MÓDULO

2.1.- CONSTRUCCIÓN Y ENSAMBLAJE DEL NUEVO MÓDULO PARA VERIFICAR LOS PROGRAMAS A IMPLEMENTAR EN MICROCONTROLADORES AT89C51 QUE INCLUYE PROTOCOLO RS - 232

2.1.1- PRINCIPIO DE FUNCIONAMIENTO DE LOS ELEMENTOS A UTILIZAR

2.1.1.1- Microcontrolador AT89C51

Microcontrolador.- Un microcontrolador es un circuito integrado que contiene toda la estructura (arquitectura) de un microcomputador, o sea CPU, RAM, ROM y circuitos de entrada y salida. Los resultados de tipo práctico, que pueden lograrse a partir de estos elementos, son sorprendentes.

Este microcontrolador viene con una nomenclatura con números y letras así por ejemplo: El AT89C51 que es el que usaremos en este proyecto, se detalla en la tabla 2.1.

Nomenclatura	AT	89	C	51
Significado	Fabricado por: ATMEL	Puede ser programado	Tecnología usada (CMOS)	Número del microcontrolador

Tabla 2.1 Nomenclatura del microcontrolador AT89C51

2.1.1.1.1- Características generales

El AT89C51 es un microcontrolador de baja potencia y alto rendimiento, usa tecnología CMOS de 8 bits y 4 Kbytes de memoria Flash programable y borrable (EEPROM) que permite a la memoria ser reprogramada por un programador convencional de memorias no volátiles.

Es fabricado usando la tecnología de alta densidad no volátil de ATMEL y es un circuito integrado VLSI (Integrado de muy grande escala).

Las características más importantes son las siguientes:

- 4 Kbytes de memoria Flash reprogramable.
- Soporta 1000 ciclos de escritura/borrado.
- Funcionamiento totalmente estático: 0 Hz a 24 MHz.
- 128 x 8-bit de RAM interior.
- 32 líneas de I/O programables.
- Dos timers de 16 bits.
- Seis fuentes de interrupción.
- Canal serial programable (full duplex).
- 5 estructuras de interrupción con dos niveles de prioridad.
- 1 circuito de reloj.
- 64 Kbytes de espacio para programa.
- 64 Kbytes de espacio para datos.

2.1.1.1.2- Distribución y descripción de pines

La figura 2.1 indica la distribución de los 40 pines del AT89C51.

P1.0	1	40	VCC
P1.1	2	39	P0.0 (AD0)
P1.2	3	38	P0.1 (AD1)
P1.3	4	37	P0.2 (AD2)
P1.4	5	36	P0.3 (AD3)
P1.5	6	35	P0.4 (AD4)
P1.6	7	34	P0.5 (AD5)
P1.7	8	33	P0.6 (AD6)
RST	9	32	P0.7 (AD7)
(RXD) P3.0	10	31	\overline{EA} / VPP
(TXD) P3.1	11	30	ALE / \overline{PROG}
($\overline{INT0}$) P3.2	12	29	\overline{PSEN}
($\overline{INT1}$) P3.3	13	28	P2.7 (A15)
(T0) P3.4	14	27	P2.6 (A14)
(T1) P3.5	15	26	P2.5 (A13)
(WR) P3.6	16	25	P2.4 (A12)
(RD) P3.7	17	24	P2.3 (A11)
XTAL1	18	23	P2.2 (A10)
XTAL2	19	22	P2.1 (A9)
GND	20	21	P2.0 (A8)

Figura 2.1 Distribución de pines del microcontrolador AT89C51

Vcc (pin 40) Suministro de voltaje.

GND (pin 20) Tierra.

Pórtico 0.- Comprende desde el pin 32 (P0.7) hasta el pin 39 (P0.0). Es un pórtico bidireccional de entrada y salida de datos (I/O) de 8 bits, y para su funcionamiento requiere de resistencias externas de 10 K Ω a la salida de todos sus pórtricos para su protección interna como se muestra en la figura 2.2.

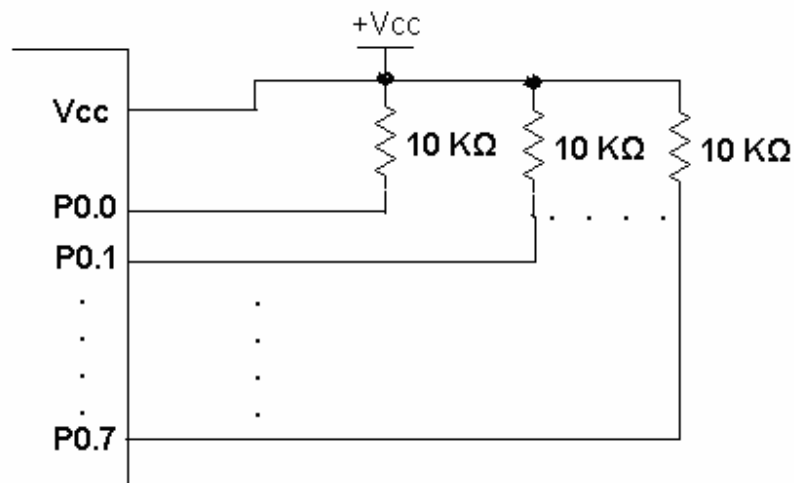


Figura 2.2 Conexión de las resistencias externas a las salidas del pòrtico 0

Pòrtico 1.- Comprende desde el pin 1 (P1.0) hasta el pin 8 (P1.7). Es un pòrtico de entrada y salida de datos (I/O) de 8 bits. Cuando se escribe 1's en el pòrtico 1, el pòrtico puede ser utilizado como entrada.

Pòrtico 2.- Comprende desde el pin 21 (P2.0) hasta el pin 28 (P2.7). Es un pòrtico de entrada y salida de datos (I/O) de 8 bits.

Pòrtico 3.- Comprende desde el pin 10 (P3.0) hasta el pin 17 (P3.7). Es un pòrtico de entrada y salida de datos (I/O) de 8 bits

El pòrtico 3 se utiliza ademàs para producir se~nales de control de dispositivos externos como se muestra en la tabla 2.2.

SIMBOLO	PINES	TIPO	PÓRTICO	NOMBRE Y FUNCION
RxD	10	E	P3.0	Pórtico serie de entrada.
TxD	11	S	P3.1	Pórtico serie de salida.
INT0	12	E	P3.2	Interrupción externa.
INT1	13	E	P3.3	Interrupción externa.
T0	14	E	P3.4	Entrada externa timer0.
T1	15	E	P3.5	Entrada externa timer1.
WR	16	S	P3.6	Habilitador de escritura para memoria externa de datos.
RD	17	S	P3.7	Habilitador de lectura para memoria externa de datos.

Tabla 2.2 Funciones especiales del p3rtico 3

En el microcontrolador AT89C51 también existen pines de control como son:

RST.- Corresponde al pin 9 y es la entrada del reset. Se resetea el dispositivo con 1L durante dos ciclos de máquina mientras el oscilador está funcionando.

ALE/PROG.- Corresponde al pin 30 ALE (Addres Latch Enable) pulso que emite el microcontrolador para enclavar el byte bajo del bus de direcciones en el acceso a la memoria externa. PROG pin de entrada de pulsos de programación de la memoria EPROM se activa con 0L.

PSEN.- (Program Store Enable) corresponde al pin 29, es la señal habilitadora de lectura para memoria de programas externos.

EA/VPP.- (External Access Enable) corresponde al pin 31, permite acceder a la memoria interna o externa. EA debería ir conectado a GND para permitir extraer el programa externo, y a Vcc para la ejecución del programa interno.

XTAL 1.- Corresponde al pin 19 y es la entrada al amplificador del oscilador y la entrada al reloj interior del circuito que opera.

XTAL 2.- Corresponde al pin 18 y es la salida del amplificador oscilador inversor.

Localidades asignadas a las interrupciones.- En la tabla 2.3 se muestra las localidades que han sido asignadas por el fabricante, para dar servicio a las rutinas de interrupción.

FUENTE DE INTERRUPCIÓN	VECTOR DE DIRECCIONES
IE0 (Interrupción 0 externa)	0003H
TF0 (Interrupción del timer 0)	000BH
IE1 (Interrupción 1 externa)	0013H
TF1 (Interrupción del timer 1)	001BH
R1 y T1 (Interrupción serial)	0023H
TF2 y EXF2 (Sólo para el 8052)	002BH

Tabla 2.3 Localidades de las interrupciones

Una interrupción puede ser causada de manera externa o interna, es decir puede ser producida por un dispositivo periférico o por programación respectivamente. La interrupción con mayor alto orden es el RESET. Cuando el RESET ocurre el programa comienza a partir de la dirección 0000H.

Cuando una interrupción es producida, el Contador del Programa (PC) almacena su contenido temporalmente dentro de la RAM como una pila de datos, y se carga con la dirección de la localidad donde se encuentra la rutina de servicio de la interrupción correspondiente. Una vez posicionado en esa localidad deberá comenzar la ejecución de la rutina de servicio, hasta que encuentre la instrucción RETI, que le

permitirá al PC recuperar nuevamente su valor original almacenado en la pila de datos de la RAM, y continuar con el programa anterior a la interrupción.

Por ejemplo a la interrupción 0, se le asigna la localidad 0003H, si la interrupción no se utiliza, esta localidad puede utilizarse para propósitos generales del programa, si la interrupción ha sido permitida, (estableciendo el bit correspondiente dentro del registro de control IE), en el momento que exista una activación de la interrupción (estado bajo en la línea INTO) el PC se cargará con 0003 y saltará a esa localidad para comenzar a ejecutar la rutina de servicio.

Estas localidades de memoria de los servicios de interrupción están separadas en intervalos de 8 bytes, entre sí. Cuando un servicio de interrupción es corto, éste puede estar contenido en los 8 bytes. En el caso de que fuese largo se puede ejecutar un salto a otra localidad de memoria para continuar con la secuencia de interrupción. El término del servicio de interrupción deberá realizarse mediante la ejecución de la instrucción RETI.

2.1.1.1.3.- Descripción de los espacios de memoria

La memoria del sistema del AT89C51 se clasifica en tres partes fundamentales:

La primera, llamada memoria de programa, en donde se encuentran todas las instrucciones que van a ser ejecutadas por el AT89C51, es decir, el programa de trabajo.

El segundo espacio de memoria denominado, memoria de datos es accesado mediante la activación de las señales RD y WR, durante la lectura o escritura de datos respectivamente.

El tercer espacio de memoria es denominado como memoria RAM interna, el cual se subdivide en 128 bytes de memoria bajos y en 128 bytes de memoria altos, como se observa en la figura 2.3.

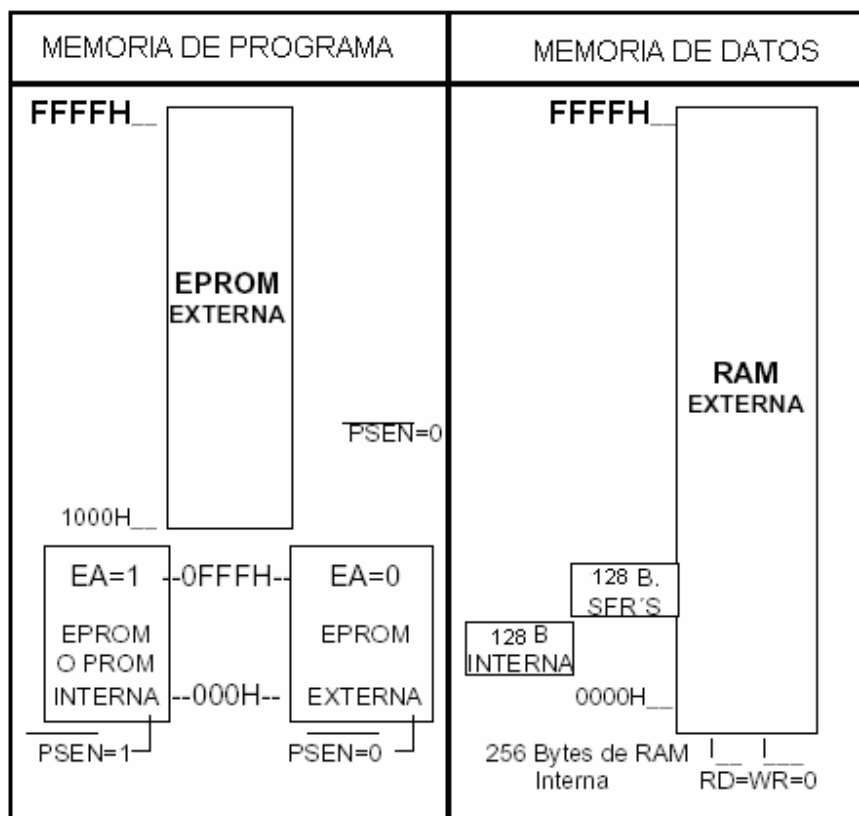


Figura 2.3 Espacios de memoria

Memoria de programa interna y externa.- Cuando se utiliza memoria interna ROM esta puede ser accesada mediante la conexión de la línea $EA = 1$ (Vcc). Si la memoria interna es de 4 Kbytes y $EA = 1$, el CPU seleccionará internamente el ROM, desde **0000H** hasta **0FFFH** y de manera externa automáticamente a partir de **1000H** hasta **FFFFH**.

Por el contrario, si la línea $EA = 0$, el CPU seleccionará de forma externa el ROM, desde la dirección **0000H** hasta **FFFFH**.

Memoria de datos.- El espacio de memoria RAM interno está dividido en dos espacios, el primer bloque es referido como la parte baja de 128 bytes, el segundo la parte alta de 128 bytes llamado espacio SFR (Registros de Funciones Especiales). Las direcciones de la Memoria Interna de Datos siempre son de un byte (de 00H a FFH).

Como se puede apreciar en la figura 2.4 los 128 bytes más bajos son divididos en 4 bloques de 8 registros cada uno, que contienen los valores de los registros R0 A R7, los bloques pueden ser seleccionados mediante la escritura en los bits 3 y 4 del registro PSW. La utilización de registros permite un uso más eficiente del espacio de códigos debido a que sus direccionamientos son de 8 bits únicamente. También contienen un espacio de 128 bytes en la parte alta de la memoria que son direccionados directamente, en este espacio se localizan los registros de funciones especiales (SFR). Estos registros especiales, tienen sus localidades bien establecidas, y son utilizados por el microcontrolador para realizar las distintas operaciones internas que ejecuta el microcontrolador, así como también para el control y acceso de los diferentes pórtricos de entrada y salida.

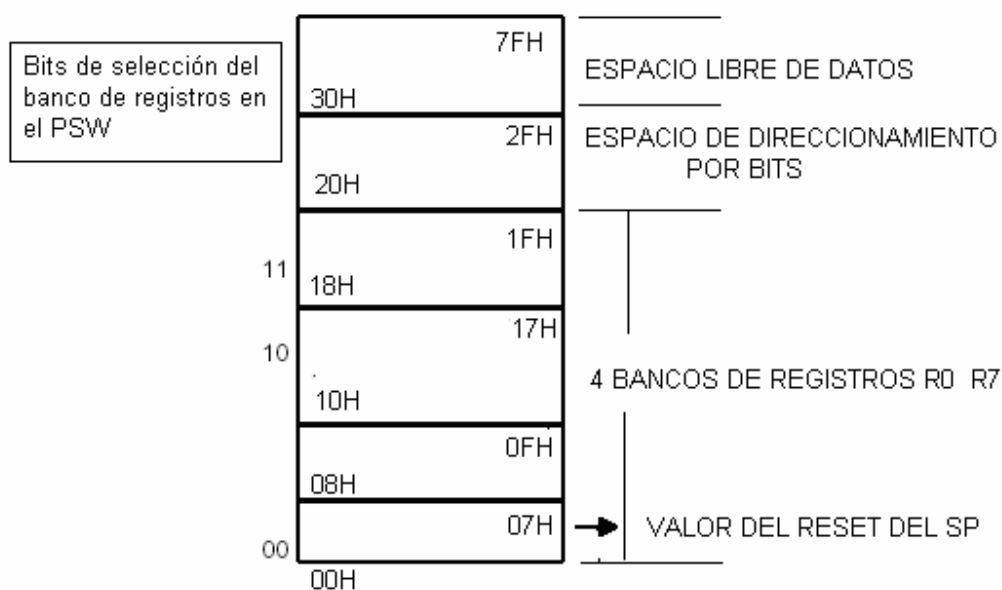


Figura 2.4 Memoria de datos

2.1.1.1.4.- Localidades de los registros de funciones especiales

Los registros de función especial (*SFR*) son localidades de memoria interna que controlan a los periféricos disponibles en el microcontrolador, como los pórtricos, interrupciones, así como otras características del procesador. Las direcciones de los SFR se indican en la tabla 2.4.

SÍMBOLO	NOMBRE	DIRECCIÓN
ACC	Acumulador	E0H
B	Registro B	0F0H
SP	Puntero de Pila (apuntador de apilamiento)	81H
DPTR	Puntero de Datos (apuntador de datos)16bits	
DPL	Parte Baja Puntero de Datos	82H
DPH	Parte Alta Puntero de Datos	83H
P0	Pórtico 0	80H
P1	Pórtico 1	90H
P2	Pórtico 2	0A0H
P3	Pórtico 3	0B0H
IP	Control de Prioridad de Interrupción.	0B8H
IE	Control de Validación de Interrupción	0A8H
TMOD	Modo de control Timer/Contador	89H
TCON	Control del Timer/Contador	88H
TH0	Byte alto del Timer/Contador 0	8CH
TL0	Byte bajo del Timer/Contador 0	8AH
TH1	Byte alto del Timer/Contador 1	8DH
TL1	Byte bajo del Timer/Contador 1	8BH
SCON	Control serie	98H
SBUF	Buffer de datos serie	99H
PCON	Control de Potencia	87H

Tabla 2.4 Registros SFR

Mapa de memoria de los registros de funciones especiales (SFR).- Va desde la dirección 80H a FFH como se indica en la tabla 2.5.

	0	1	2	3	4	5	6	7	
F8									FF
F0	B								F7
E8									EF
E0	ACC								E7
D8									DF
D0	PSW								D7
C8									CF
C0									C7
B8	IP								BF
B0	P3								B7
A8	IE								AF
A0	P2								A7
98	SCON	SBUF							9F
90	P1								97
88	TCON	TMOD	TL0	TL1	TH0	TH1			8F
80	P0	SP	DPL	DPH				PCON	87

Tabla 2.5 Mapa de memorias de los registros de funciones especiales

Registro de palabra del estado del programa (PSW).- El registro PSW (Program status word) de palabra del estado del programa contiene algunos bits que reflejan el estado del CPU en ese instante. El registro PSW reside en el espacio SFR.

Las funciones del registro PSW se muestra en la tabla 2.6.

CY	AC	F0	RS1	RS0	OV	-	P
CY	PSW.7	Bandera del carry.					
AC	PSW.6	Bandera del carry auxiliar (operaciones en BCD).					
F0	PSW.5	Bandera 0 para usos generales.					
RS1	PSW.4	Bit 1 selector del banco de registros.					
RS0	PSW.3	Bit 0 selector del banco de registros.					
OV	PSW.2	Bandera del overflow.					
-	PSW.1	Bandera sin definir.					
P	PSW.0	Bandera de paridad. Se establece/limpia por hardware, indica si el número de 1's en el acumulador es par o impar.					

Tabla 2.6 Registro PSW

Este registro PSW contiene:

El bit de carry.

El bit de carry Auxiliar (para operaciones BCD).

Los dos bits de selección del banco de registros.

La bandera de overflow.

El bit de paridad

Dos banderas sin definir.

El bit de Paridad refleja el número de 1's, en el acumulador:

P=1, si el Acumulador contiene un número impar de 1's

P=0, si el Acumulador contiene un número par de 1's, es decir el número de 1's, en el acumulador más P es siempre par.

Registro de control de potencia (Consumo de energía).- En la tabla 2.5 se observa la distribución del registro PCON, que sirve para controlar el consumo de energía, el cual es utilizado sólo por los dispositivos fabricados con la tecnología CHMOS que permite disminuir dicho consumo de energía, en estados de espera. En la tabla 2.7 se muestra la bandera PCON.7 (SMOD), que sirve para dividir la frecuencia de transmisión o de recepción por el pórtico serie, proporcionada ya sea, por la fase 2 de los estados, (1/2 de la frecuencia del oscilador en la transmisión serie en modo 2), o bien, por el timer 1 en los modos 1 y 3. El funcionamiento de los timers en los modos 1, 2 y 3 se los puede ver más adelante.

DIRECCIÓN: 87H

SMOD	-	-	-	GF1	GF0	PD	IDL
-------------	---	---	---	------------	------------	-----------	------------

SMOD	Dobla el "BAUD RATE" para el pórtico serie cuando se utiliza el timer para generar el BAUD RATE.
GF1	Propósitos generales.
GF0	Propósitos generales.
PD	Bajo consumo de energía 80C51BH CHMOS.
IDL	Bajo consumo de energía.

Tabla 2.7 Registro PCON

Registro habilitador de interrupciones (IE).- La desactivación general de las interrupciones es efectuada mediante la escritura de un 0 lógico, en la bandera EA (IE.7).

Con la bandera EA=1, el AT89C51 está en condiciones de aceptar interrupciones, aunque la verdadera aceptación es realizada cuando se escribe un 1 lógico, en la bandera de la interrupción correspondiente del registro de interrupciones, IE. Los bits que conforman este registro se indican en la tabla 2.8.

DIRECCIÓN: A8H

EA	-	ET2	ES	ET1	EX1	ET0	EX0
EA	IE.7	Desactiva todas las INTERRUPTACIONES EA=0.					
ET2	IE.5	Activa la interrupción causada por el timer2 (ET2=1).					
ES	IE.4	Activa la interrupción causada por el p \acute{o} rtico serial.					
ET1	IE.3	Activa la interrupción de sobreflujo causada por el timer 1.					
EX1	IE.2	Activa la interrupción causada externamente en INT1.					
ET0	IE.1	Activa la interrupción de sobreflujo causada por el timer 0.					
EX0	IE.0	Activa la interrupción causada externamente en INT 0.					

Tabla 2.8 Registro habilitador de interrupciones

Registro de prioridad.- El AT89C51 tiene dos planos de prioridad para trabajar las interrupciones, llamadas alta y baja, respectivamente. En la inicialización, todas las interrupciones trabajan en el plano de baja prioridad. Para pasar del plano de baja prioridad al de alta, es necesario escribir un 1 lógico en las banderas correspondientes a las interrupciones que se desean aumentar de prioridad, ubicadas dentro del registro IP como se muestra en la tabla 2.9.

DIRECCIÓN: B8H

-	-	PT2	PS	PT1	PX1	PT0	PX0
PT2	IP.5	Timer 2 PT2=1 mayor prioridad.					
PS	IP.4	Define el nivel de prioridad de la interrupción del p \acute{o} rtico serial.					
PT1	IP.3	Define el nivel de prioridad de la interrupción del Timer 1.					
PX1	IP.2	Define el nivel de prioridad de la interrupción externa 1.					
PT0	IP.1	Define el nivel de prioridad de la interrupción del Timer 0.					
PX0	IP.0	Define el nivel de prioridad de la interrupción externa 0.					

Tabla2.9 Registro de prioridad

Timer / Contador.- El AT89C51 tiene 2 timer/contadores de 16 bits cada uno, llamados Timer 0 y el Timer 1 respectivamente. Ambos pueden ser configurados para operar como temporizadores (timers) o como contadores (counters).

Cuando se trabaja como contador, el registro interno del contador, es incrementado cada vez que existe una transición negativa (de 1L a 0L) por la línea de entrada correspondiente a T0 ó T1. En cambio, cuando funciona como temporizador "Timer", el registro es incrementado cada 12 periodos de oscilación es decir su frecuencia de conteo es 1/12 de la frecuencia del oscilador.

En el momento que los bits del registro del contador pasan de 1L a todos 0L, se activa la línea de interrupción interna correspondiente a TF0 o TF1, generándose (si ha sido permitida) una interrupción.

Registro de control del pórtilo timer/contador (TCON).- El registro de control del Timer/Contador de la tabla 2.10 es direccionable por Bit, para activar o desactivar cada una de sus banderas.

DIRECCIÓN: 88H

TF1	TR1	TF0	TR0	IE1	IT1	IE0	IT0
-----	-----	-----	-----	-----	-----	-----	-----

TF1	TCON.7	Bandera de sobreflujo (overflow) del registro del Timer 1. Activada por hardware cuando el registro que guarda la cuenta del Timer/Contador 1, incrementa su contenido pasando todos sus bits de 1L a 0L.
TR1	TCON.6	Bit de control de activación del timer 1. Habilitado/Deshabilitado por software para colocar el Timer/Contador en Encendido/Apagado.
TF0	TCON.5	Bandera de sobreflujo (overflow) del registro del Timer 0. Activada por hardware cuando el registro que guarda la cuenta del timer/contador 0, incrementa su contenido pasando todos sus bits de 1L a 0L.

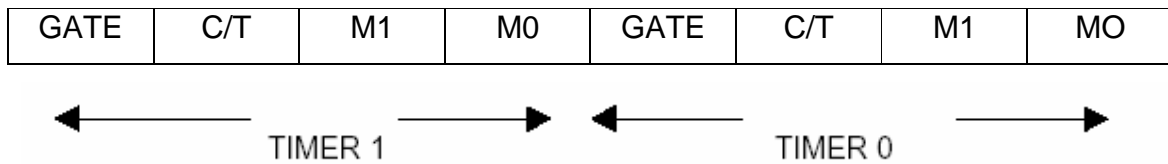
TR0	TCON.4	Bit de control de activación del timer 0. Habilitado/Deshabilitado por software para colocar el Timer/Contador en Encendido/Apagado.
IE1	TCON.3	Bandera de transición de la interrupción externa 1. Activada por hardware cuando una transición (de 1L a 0L) en la línea de interrupción externa 1, es detectada. Limpiada por hardware cuando la interrupción es procesada. (Solamente se acciona si se programó la aceptación de la interrupción por transiente, IT1=1).
IT1	TCON.2	Bit de control de la interrupción 1. Activado/Limpiado por software para especificar el tipo de interrupción, por nivel bajo (IT1= 0L) o por transiente negativo (IT1=1L).
IE0	TCON.1	Bandera de transición de la interrupción externa 0. Activada por hardware cuando una transición (de 1L a 0L) en la línea de interrupción externa 0, es detectada. Limpiada por hardware cuando la interrupción es procesada. (Solamente se acciona si se programó la aceptación de la interrupción por transiente, IT0=1).
IT0	TCON.0	Bit de control de la interrupción 0. Activado/Limpiado por software para especificar el tipo de interrupción, por nivel bajo (IT0= 0L) o por transiente negativo (IT0=1L).

Tabla 2.10 Registro TCON

Registro de modo de control del timer/contador.- Este registro permite especificar si se van a trabajar como Temporizadores (Timers) o Contadores (Counters), los pórtricos denominados Timer 0 y Timer 1, que se muestran en la tabla 2.11.

Existen 4 modos de trabajo para estos pórtricos, los cuales son definidos por la escritura en los bits M1 y M0 de TMOD como se indica en la tabla 2.12.

DIRECCIÓN: 89H



GATE	Cuando TRx (en TCON) está activada y GATE=1, TIMER/COUNTERx correrá solamente si la línea INTx está en posición alta (control por hardware). Cuando GATE=0, TIMER/COUNTERx correrá solamente si TRx=1 (control por software).
C/T	Selector de Timer o de Contador. Es limpiado por la operación del Timer (entrada del reloj del sistema interno). Es activada por la operación del Contador (entrada de la línea Tx)
M1	Bit selector del modo.
M0	Bit selector del modo.

Tabla 2.11 Registro TMOD

M1	M0	MODO	ESPECIFICACIÓN
0	0	0	Timer/contador de 13 bits
0	1	1	Timer/contador de 16 bits
1	0	2	Timer/contador de 8 bits recargables
1	1	3	Timer 0, TL0 Timer/contador de 8 bits, controlado por los bits de control del Timer 0. TH0 Timer de 8 bits controlado por los bits de Control del timer 1. (El Timer 1 no se utiliza)

Tabla 2.12 Los cuatro modos de trabajo de M1 y M0

Modo 0 del timer/contador.- En este modo, cualquiera de los dos timers, 0 ó 1, trabajan como un contador de 8 bits.

El registro del Timer 1 está configurado como un registro de 13 bits, que consisten de los 8 bits de TH1 y los 5 bits menos significativos de TL1. Los 3 bits más significativos de TL1 no se utilizan en este modo.

Modo 1 del timer/contador.- Este modo es utilizado por cualquiera de los dos Timers, se caracteriza principalmente por ser un Timer/contador de 16 bits cuyos valores se encuentran cargados en los registros TH y TL de cada uno de los Timers.

Modo 2 del timer/contador.- Este modo puede ser utilizado tanto por el timer 0 como por el 1, tienen un registro de conteo de 8 bits (TLx).

El registro TL1 es cargado automáticamente con el contenido de TH1, cuando se produce el rebalse en TL1, el cual además establece la bandera de TF1.

Modo 3 del timer/contador.- El Timer 1, en el modo 3 mantiene su cuenta, es decir, tiene el mismo efecto que cuando se establece la bandera TR1=0.

El Timer 0, en éste modo, establece TL0 y TH0 como dos contadores separados. TL0 utiliza los bits de control (C/T, GATE, TR0, INT0) del Timer 0. TH0 es bloqueado como temporizador "Timer", el cual emplea las señales de control del Timer 1, TR1 y TF1.

2.1.1.1.5.- Pórtico Serial

El AT89C51 incorpora un pórtico serie que permite la comunicación en 4 modos, desde sincrónica unidireccional (half-duplex) a asíncrona bidireccional (full-duplex) en sistemas punto a punto o punto a multipunto. Es decir, que dota al dispositivo de gran flexibilidad en sus capacidades de comunicación. El receptor contiene un almacén "Buffer", que le permite comenzar a recibir un segundo dato sin necesidad de que el

primero haya sido completamente leído del registro Buffer. Sin embargo si el primer byte permanece sin ser leído hasta el final de la recepción del segundo dato, éste se perderá.

El dato de la recepción y de la transmisión se encuentra en el registro SBUF del SFR (espacio de funciones especiales).

- SBUF: (99H) 2 registros con la misma dirección para contener el dato recibido o a transmitir.
- SCON: (98H) control de la transmisión.

Las direcciones de los registros SBUF Y SCON se muestran en la tabla 2.5.

Registro de control del p rtico serial (SCON).- El p rtico serie puede ser operado en 4 modos diferentes que son especificados mediante la escritura en los bits SM0 y SM1 del registro de Control del P rtico Serie. En la tabla 2.13 se observa los bits que pertenecen al registro SCON.

SM0	SM1	SM2	REN	TB8	RB8	TI	RI
-----	-----	-----	-----	-----	-----	----	----

SM0	SCON.7	Especifica el modo de control del p�rtico serie.
SM1	SCON.6	Especifica el modo de control del p�rtico serie.
SM2	SCON.5	Habilita la comunicaci�n del tipo "multiprocesador" utilizado en los modos 2 y 3. En estos modos, si SM2 = 1, RI no es activado si el noveno dato recibido (RB8) es 0. En modo 1, RI no es activado si no se recibe un bit de stop. En el modo 0, SM2 ser� 0.
REN	SCON.4	Establece la recepci�n serie, cuando REN = 0 se desactiva la recepci�n (por software).

TB8	SCON.3	Almacena el noveno bit que será transmitido en los modos 2 y 3.
RB8	SCON.2	Es el noveno bit que fue recibido en los modos 2 y 3. En el modo 1, si SM = 0, RB8 es el bit de stop recibido. En el modo 0 RB8 no es usado.
TI	SCON.1	Bandera de interrupción de la transmisión. Activada por hardware al final del octavo bit en el modo 0, o al principio del bit de stop en los otros modos. Debe ser limpiado por software.
RI	SCON.0	Bandera de interrupción de la recepción. Activada por hardware al final del octavo bit en el modo 0, o al medio tiempo de transmitido el bit de stop en los otros modos. Debe ser limpiado por software.

Tabla 2.13 Registro SCON

Mediante los bits SM0 y SM1 podemos seleccionar uno de entre los 4 modos de operación que son seleccionados por software utilizando los bits: 6 = SM1 y el bit 7 = SM0 del registro SCON como se muestra en la tabla 2.13. Estos modos y sus características principales son:

SM0	SM1	MODO
0	0	0
0	1	1
1	0	2
1	1	3

Descripción del funcionamiento de cada modo.

Modo 0:

- Los datos serie entran y salen a través del pin Rx.
- Los pulsos para el desplazamiento salen por el pin Tx.
- El tamaño de palabra transmitida es de 8 bits, el primer bit que sale es el LSB.
- Velocidad de transmisión fija, 1/12 de la frecuencia del reloj.

Modo 1:

- Los datos transmitidos salen por el pin Tx.
- Los datos recibidos se recogen por el pin Rx.
- La transmisión se inicia con 1 bit de start (0), seguido de 8 bits de datos (primero en salir el LSB) y finaliza con 1 bit de stop (1).
- Velocidad variable según la programación del timer 1 o timer 0.

Modo 2:

- 11 bits son transmitidos, datos a transmitir salen por el pin Tx.
- Los datos a recibir se recogen por el pin Rx.
- La transmisión se inicia con 1 bit de start (0), seguido de 9 bits de datos (primero en salir el LSB, 9º bits el contenido en TB8 de SCON) y finaliza con 1 bit de stop (1)
- Velocidad fija a 1/32 o 1/64 de la frecuencia del reloj

Modo 3:

- 11 bits son transmisión (Tx) o recibidos (Rx), en la misma forma que el modo 2, sólo que aquí la frecuencia de transmisión/recepción (Baud Rate) es variable. Se utiliza el Timer 1 para generar el Baud Rate.

2.1.1.1.6.- Modos de direccionamiento

Los modos de direccionamiento que trabajan son los siguientes:

Direccionamiento directo.- En este direccionamiento el operando es especificado por una dirección de 8 bits en la instrucción. Solamente los datos de la RAM interna y los del campo del SFR pueden ser directamente direccionados.

Ejemplo:

ADD A, 7FH; El Acumulador es sumado al dato que se encuentra en la dirección 7FH de la RAM interna y el resultado será almacenado en el Acumulador.

MOV A, 2EH; El Acumulador es cargado con el dato que se encuentra en la dirección 2EH de la memoria RAM interna.

MOV 3DH, 4EH; La localidad con dirección 3DH es cargado con el dato que se encuentra en la dirección 4EH.

Direccionamiento indirecto.- En este direccionamiento se utiliza un registro en el cual se encuentra la dirección del operando. Toda la memoria RAM interna y externa puede ser direccionada indirectamente. Los registros de direcciones de 8 bits, pueden ser los registros R0 y R1 del banco de registros.

El registro de dirección de 16 bits puede ser solamente el registro DPTR.

Ejemplo:

ADD A, @R0; El Acumulador es sumado con el contenido de la dirección que está apuntando R0.

MOV A, @R0; El Acumulador es cargado con el dato que se encuentra en la dirección apuntada por R0.

MOVX A, @DPTR; El Acumulador es cargado con el dato que se encuentra en la dirección apuntada por el DPTR.

Direccionamiento inmediato.- El valor de una constante sigue al código de operación en el programa.

Ejemplo:

MOV A, #64H; El acumulador es cargado con el dato 64H inmediatamente.

ADD A, #120; El acumulador es sumado al número decimal 120 y el resultado se almacena en el acumulador.

MOV DPTR, #1245H; El DPTR es cargado con el dato 1245H en forma inmediata.

Direccionamiento indexado.- Solamente la memoria del programa puede ser accesada mediante este modo de direccionamiento y sólo en lecturas. Este modo de direccionamiento es utilizado en las lecturas de tablas de la memoria del programa o datos que se encuentran como constantes.

Un registro de 16 bits (el DPTR o el PC), apunta la base de la tabla y mediante el Acumulador se establece el número de la entrada de la tabla. La dirección de la entrada de la tabla en la memoria del programa está formada por la suma del Acumulador y el Apuntador de Base (DPTR o PC).

Otro tipo de Direccionamiento indexado, es usando la instrucción "Salto de casillero". En este caso la dirección del destino el salto es calculada como la suma del apuntador de base más el Acumulador.

Ejemplo:

MOVC A, @A+DPTR; Mueve una constante que se encuentra en la memoria del programa. El Acumulador es cargado con el dato que se encuentra apuntado por la dirección formada por la suma del Acumulador A y el Apuntador de Datos.

MOVC A, @A+PC; El Acumulador es cargado con dato que se encuentra en la dirección formada por la suma del mismo Acumulador A y el Contador del Programa (PC).

Direccionamiento por registro.- Los 8 registros pueden ser accedidos mediante ciertas instrucciones que simplifican sus códigos de operación y en la mayoría de los casos son más rápidas.

Existen 4 bancos de registros, cada banco contiene los 8 registros (R0 a R7). Estos bancos pueden ser accedidos mediante los bits 3 y 4 del PSW.

Ejemplo:

ADD A, R7; El acumulador es cargado con el resultado de suma del Acumulador y el contenido del registro R7.

DEC R0; Decrementa el registro R0.

Transferencia de datos.

Ram interna.- Para poder mover datos de la memoria interna RAM o SFR, existen 8 instrucciones principales que se observan en la tabla 2.14.

MOV A, <fuente>	A = <fuente>
MOV <destino>,A	<destino = A
MOV <destino>,<fuente>	; <destino> = <fuente>
MOV DPTR,#dato 16 bits	DPTR = constante de 16 bits inmediata.
PUSH <fuente>	INC SP à <@SP> ß (fuente)
POP <destino>	(destino) ß <@SP> ; DEC SP
XCH A, <byte>	ACC y <byte> intercambia sus datos
XCHD A, @Ri	ACC y @Ri intercambian nibble bajo

Tabla 2.14 Transferencia de datos a la RAM interna

La instrucción **MOV <dest>, <fuente>**; permite transferir datos de la memoria interna RAM y del SFR sin pasar a través del Acumulador, mientras que las instrucciones **MOV A, <fuente>** y **MOV <dest>,A** utilizan al Acumulador para el movimiento de datos dentro de la memoria interna RAM.

La instrucción **PUSH**, primeramente incrementa el Stack Pointer (SP), y después guarda el dato dentro de la localidad de memoria apuntada por el Stack.

La instrucción **POP**, primero toma el dato de la memoria y después decrementa el SP.

La instrucción **XCH A, <dirección>** permite al Acumulador y al dato apuntado por la dirección de intercambiarse entre si. La instrucción **XCHD A, @R0** intercambia los 4 bits menos significativos (low nibble), de Acumulador con los 4 bits menos significativos del dato apuntado por el registro R0.

Ram externa.- Las instrucciones para la transferencia de datos en la memoria RAM externa, son 4 básicamente y se observan en la tabla 2.15:

MOVX A, @Ri	$A \leftarrow \langle @Ri \rangle$
MOVX @Ri,A	$\langle @Ri \rangle \leftarrow A$
MOVX A,@DPTR	$A \leftarrow \langle @DPTR \rangle$
MOVX @DPTR,A	$\langle @DPTR \rangle \leftarrow A$

Tabla 2.15 Transferencia de datos a la RAM externa

Las habilitaciones de lectura y escritura en RAM externa son activadas solamente durante la activación de una instrucción **MOVX**.

Instrucciones booleanas.- El AT89C51 contiene un completo procesador booleano (por bits), el cual permite ejecutar instrucciones de movimiento, limpieza, establecimiento, complementación de un solo bit, y operaciones de AND y OR entre bits como se observa en la tabla 2.16.

ANL C,bit	$C \leftarrow C \text{ .AND. bit}$
ANL C,/bit	$C \leftarrow C \text{ .AND. .NOT. bit}$
ORL C,bit	$C \leftarrow C \text{ .OR. bit}$
ORL C,/bit	$C \leftarrow C \text{ .OR. .NOT. bit}$
MOV C,bit	$C \leftarrow \text{bit}$
MOV bit,C	$\text{bit} \leftarrow C$
CLR C	$C \leftarrow 0$
CLR bit	$\text{bit} \leftarrow 0$
SETB C	$C \leftarrow 1$
SETB bit	$\text{bit} \leftarrow 1$
CPL C	$C \leftarrow \text{.NOT. } C$
CPL bit	$\text{bit} \leftarrow \text{.NOT. bit}$

Tabla 2.16 Instrucciones booleanas

Instrucciones De Salto.- Tenemos dos instrucciones de salto y estas son:

- **Salto s condicionados.-** El juego de instrucciones con saltos que están condicionados a la activación o desactivación de algunas de las banderas del PSW son las que se muestran en la tabla 2.17.

JZ rel	Salta si A = 0
JNZ rel	Salta si A ≠ 0
DJNZ <byte>, rel	Decrementa byte directo y salta si no es igual a 0
CJNE A,<byte>,rel	Salta si A ≠ <byte>
CJNE <byte>,#dato,rel	Salta si <byte> ≠ #dato

Tabla 2.17 Saltos condicionados

- **Salto s incondicionados.-** Las instrucciones que permiten hacer los saltos incondicionados se muestran en la tabla 2.18.

JMP dirección	Salta a la dirección indicada.
JMP @A + DPTR	Salta a la dirección A + DPTR
CALL dirección	Llama a la subrutina “dirección”.
RET	Regreso de la subrutina.
RETI	Regreso de la interrupción.
NOP	Sin operación.

Tabla 2.18 Saltos incondicionados

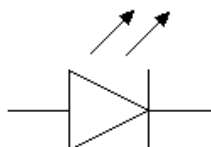
2.1.1.2.- Diodo led

Un diodo led (diodo emisor de luz) es un dispositivo semiconductor que emite luz visible cuando se polariza en forma directa.

El color depende del material semiconductor como el galio, arsénico, fósforo empleado en la construcción del diodo pudiendo variar desde el ultravioleta, pasando

por el espectro de luz visible, hasta el infrarrojo, recibiendo éstos últimos la denominación de diodos IRED (Diodo emisor infrarrojo).

Su símbolo es el siguiente:



El dispositivo semiconductor está comúnmente encapsulado en una cubierta de plástico de mayor resistencia que las de cristal que usualmente se emplean en las bombillas. Aunque el plástico puede estar coloreado, es sólo por razones estéticas, ya que ello no influye en el color de la luz emitida.

En la figura 2.5 se puede observar los componentes del led:

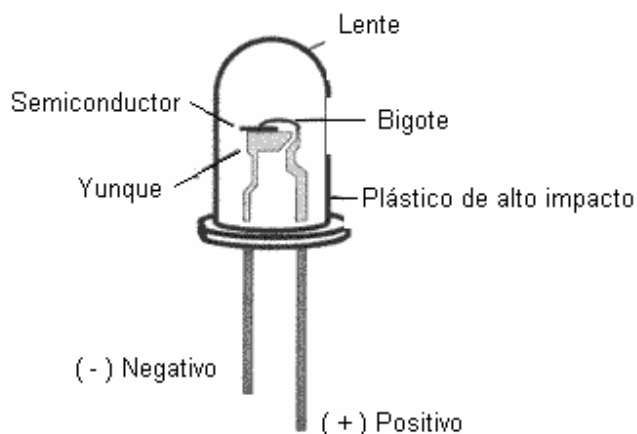


Figura 2.5 Componentes del diodo led

Voltaje de funcionamiento:

Rojo: 1.6V

Naranja: 1.7V

Verde: 2.4V

Amarillo: 2.4V

Azul: 2.5V

Intensidad de funcionamiento:

Mínima: 10mA

Media: 20mA

Máxima: 30mA

2.1.1.3.- Matriz de led's

Una matriz de led's es un conjunto de led's con un número específico de columnas por filas, de acuerdo a la necesidad que se presente. En la figura 2.6 se puede observar una matriz de 5 x 7 puntos.

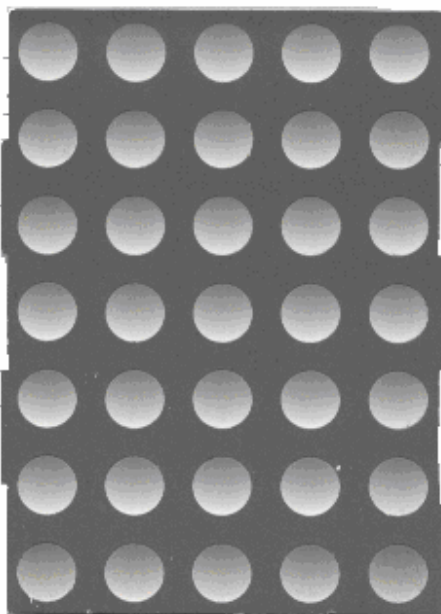


Figura 2.6 Matriz de led's de 5 x 7 puntos

2.1.1.4.- Display's de siete segmentos

El display está formado por un conjunto de 7 led's conectados en un punto común en su salida. Cuando la salida es común en los ánodos, el display es llamado de ánodo común como se observa en la figura 2.7 (a) y por el contrario, si la salida es común en los cátodos, llamamos al display de cátodo común como se observa en la figura 2.7 (b). En el display de cátodo común, una señal alta encenderá el segmento excitado por la señal.

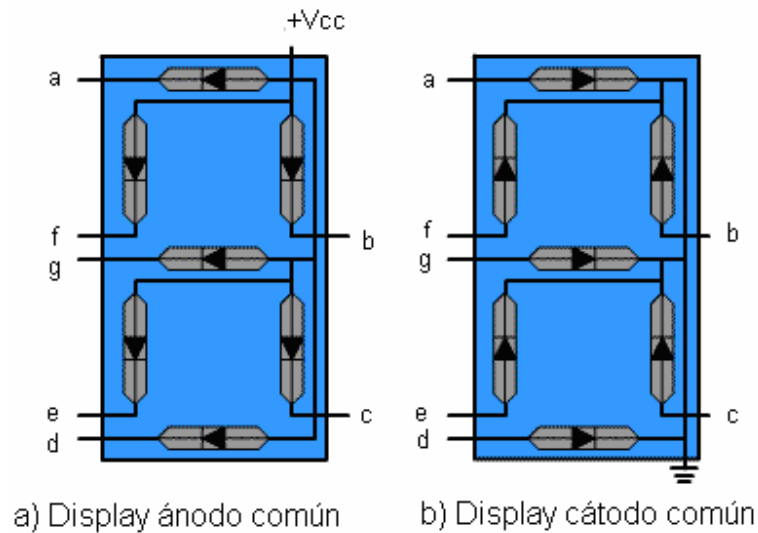


Figura 2.7 Tipos de display

2.1.1.5.- LCD

2.1.1.5.1.- Principio de funcionamiento

Es una pantalla de cristal líquido que visualiza unos ciertos caracteres. Para poder hacer funcionar un LCD, debe estar conectado a un circuito impreso en el que estén 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, como se observa en la figura 2.8.

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.

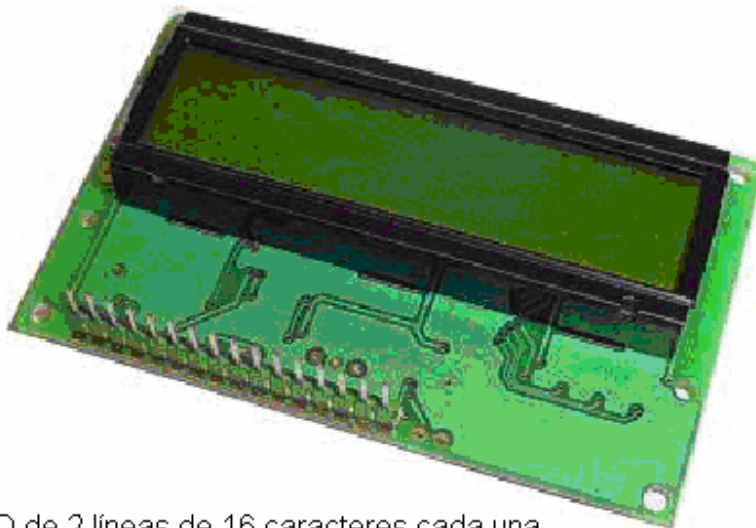


Figura 2.8 LCD de 2 líneas de 16 caracteres cada una

El LCD dispone de una matriz de 5 x 9 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, como se observa en la figura 2.9.

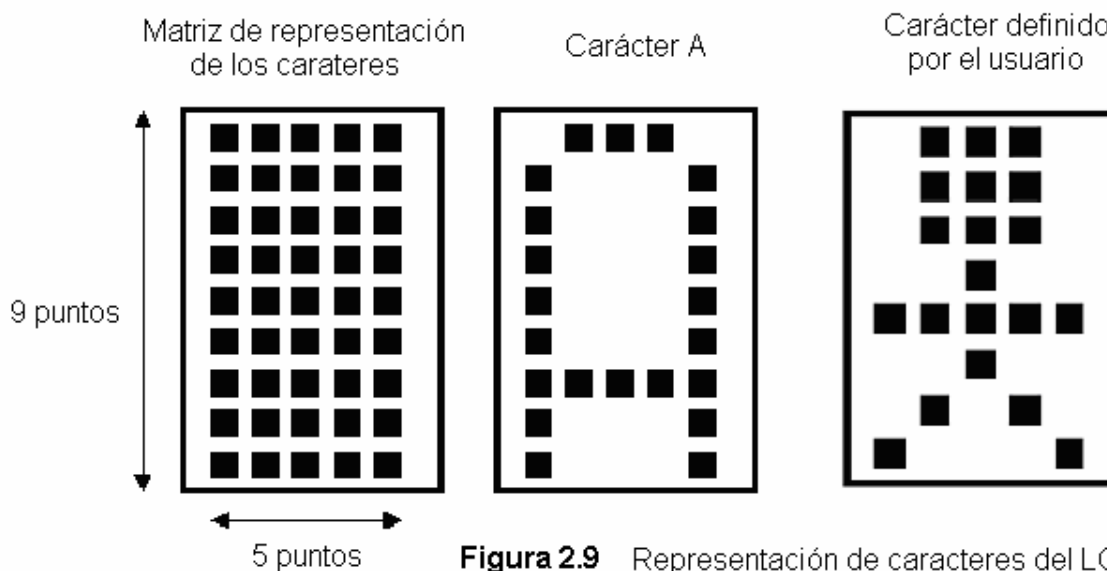
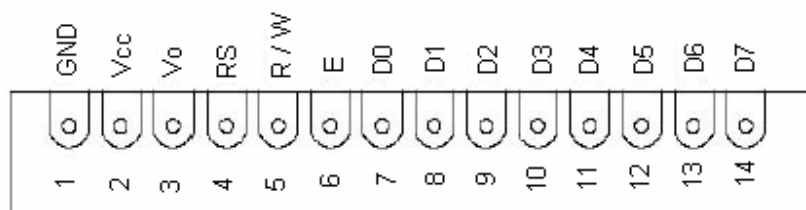


Figura 2.9 Representación de caracteres del LCD

2.1.1.5.2.- Descripción de pines

Normalmente un LCD tiene 14 pines. Si tiene 15 o 16 pines, entonces los pines restantes son usados para producir iluminación posterior. La descripción de pines se observa en la tabla 2.19.



1	GND	Tierra de alimentación GND
2	Vcc	Alimentación de +5V
3	Vo	Contraste del cristal líquido. (Aquí se coloca un potenciómetro de 10 K Ω).
4	RS	Selección del registro de control / registro de datos: RS=0 Selección registro de control RS=1 Selección registro de datos
5	R/W	Señal de lectura / escritura: R/W=0 Escritura (Read) R/W=1 Lectura (Write)
6	E	Habilitación del módulo: Se da un pulso
7-14	D0-D7	Bus de datos bidireccional.

Tabla 2.19 Descripción de pines del LCD

2.1.1.5.3.- Juego de instrucciones

Clear display.- Borra el LCD y coloca el cursor en la primera posición. Pone el bit I/D a 1 por defecto. Su tiempo de ejecución: 1.64ms.

RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
0	0	0	0	0	0	0	0	0	1

Home.- Coloca el cursor en la posición de inicio y hace que el display comience a desplazarse desde la posición original. El contenido de la memoria RAM de datos de visualización (DD RAM) permanece invariable. La dirección de la DD RAM es puesta a 0. Su tiempo de ejecución: 1.64ms.

RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
0	0	0	0	0	0	0	0	1	X

Setear modo de entrada.- Establece la dirección de movimiento del cursor y especifica si la visualización se va desplazando a la siguiente posición de la pantalla o no. Estas operaciones se ejecutan durante la lectura o escritura de la DD RAM o CG RAM. Para visualizar normalmente poner el bit RS=0. Su tiempo de ejecución: 40 μ S.

RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
0	0	0	0	0	0	0	1	I/D	S

Display On/Off control.- Activa o desactiva poniendo en ON/OFF tanto al display (D) como al cursor (C) y se establece si este último debe o no parpadear (B). Su tiempo de ejecución: 40 μ s

RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
0	0	0	0	0	0	1	D	C	B

Cursor o display shift.- Mueve el cursor y desplaza el display sin cambiar el contenido de la memoria de datos de visualización DDRAM. Su tiempo de ejecución: 40µs.

RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
0	0	0	0	0	1	S/C	R/L	X	X

Función set.- Fija el tamaño del bus de datos (DL), número de líneas del display (N) y tipo de carácter (F). Su tiempo de ejecución: 40µs.

RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
0	0	0	0	1	DL	N	F	X	X

Abreviaturas empleadas en los códigos anteriores se observa en la tabla 2.20.

RS	1 – Desplaza la visualización cada vez que se escribe un dato 0 – Modo normal
I/D	1 – Incremento del cursor 0 – Decremento del cursor
S/C	1 – Desplaza el display 0 – Mueve el cursor
R/L	1 – Desplazamiento a la derecha 0 – Desplazamiento a la izquierda
BF	1 – Módulo ocupado 0 – Módulo disponible
DL	1 – Bus de datos de 8 bits 0 – Bus de datos de 4 bits
N	1 – LCD de dos líneas

	0 – LCD de una línea
F	1 – Carácter de 5 x 10 puntos 0 – Carácter de 5 x 7 puntos
B	1 – Parpadeo del cursor ON 0 – Parpadeo del cursor OFF
C	1 – Cursor ON 0 – Cursor OFF
D	1 – Display ON 0 – Display OFF
X	Indeterminado

Tabla 2.20 Abreviaturas de los códigos del juego de instrucciones.

2.1.1.6.- Motor paso a paso

Un motor paso a paso, como todo motor, es en esencia un convertidor electromecánico, que transforma energía eléctrica en mecánica.

Mientras que un motor convencional gira libremente al aplicarle una tensión, el motor paso a paso gira un determinado ángulo de forma incremental (transforma impulsos eléctricos en movimientos de giro controlados), lo que le permite realizar desplazamientos angulares fijos muy precisos (pueden variar desde $1,80^\circ$ hasta unos 90°).

Los motores, tanto de corriente continua como de corriente alterna, son muy efectivos en muchas labores cotidianas desde la tracción de grandes trenes hasta el funcionamiento de lavarropas. Pero debido a problemas tales como la inercia mecánica o su dificultad para controlar su velocidad, se desarrollaron otro tipo de motores cuya característica principal es la precisión de giro.

Este tipo de motores son ideales cuando lo que queremos es posicionamiento con un elevado grado de exactitud y/o una muy buena regulación de la velocidad.

Están constituidos esencialmente por dos partes:

- **Estator:** parte fija construida a base de cavidades en las que van depositadas las bobinas.
- **Rotor:** parte móvil construida mediante un imán permanente.

Este conjunto va montado sobre un eje soportado por dos cojinetes que le permiten girar libremente, como se observa en la figura 2.10.



Figura 2.10 Motor paso a paso

2.1.1.6.1- Principio de funcionamiento

Los motores eléctricos, basan su funcionamiento en las fuerzas ejercidas por campos magnéticos y creadas al hacer circular una corriente eléctrica a través de una o varias bobinas. Si dicha bobina, generalmente circular y denominada estator, se mantiene en una posición mecánica fija y en su interior, bajo la influencia del campo electromagnético, se coloca otra bobina o imán permanente, llamada rotor, recorrida por una corriente y capaz de girar sobre su eje.

Al excitar el estator, se crearan los polos norte-sur, provocando la variación del campo magnético formado.

La respuesta del rotor será seguir el movimiento de dicho campo (tenderá a buscar la posición de equilibrio magnético), es decir, orientará sus polos norte-sur hacia los polos sur-norte del estator, respectivamente.

Cuando el rotor alcanza esta posición de equilibrio, el estator cambia la orientación de sus polos y se tratará de buscar la nueva posición de equilibrio. Manteniendo dicha situación de manera continuada, se conseguirá un movimiento giratorio y continuo del rotor, produciéndose de este modo el giro del eje del motor, y a la vez la transformación de una energía eléctrica en mecánica en forma de movimiento circular como se observa en la figura 2.11.

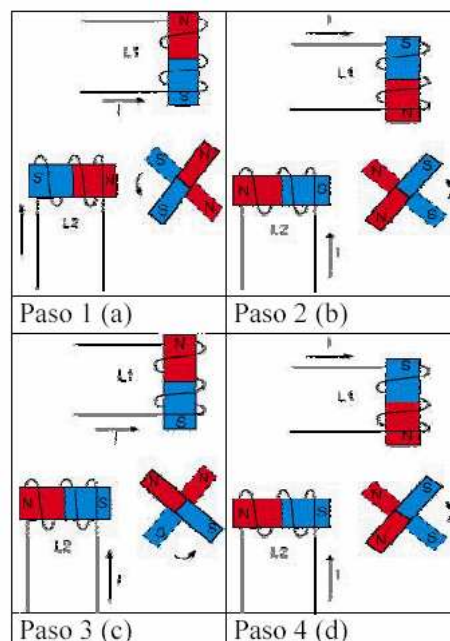


Figura 2.11 Principio de funcionamiento de un motor paso a paso

Al número de grados que gira el rotor, cuando se efectúa un cambio de polaridad en las bobinas del estator, se le denomina ángulo de paso.

Existe la posibilidad de conseguir una rotación de medio paso con el control electrónico apropiado, aunque el giro se hará con menor precisión.

Los motores son fabricados para trabajar en un rango de frecuencias determinado por el fabricante, rebasado dicho rango, provocaremos la pérdida de sincronización.

Los motores paso a paso, se controlan por el cambio de dirección del flujo de corriente a través de las bobinas que lo forman.

Además estos motores poseen la habilidad de poder quedar enclavados en una posición (si una o más de sus bobinas está energizada) o bien totalmente libres (si no circula corriente por ninguna de sus bobinas).

2.1.1.6.2- Tipos de motores paso a paso

En los motores paso a paso, podemos distinguir 3 tipos desde el punto de vista de su construcción los cuáles son:

Motor paso a paso de reluctancia variable.- En los de reluctancia variable, su rotor está fabricado por un cilindro de hierro dentado y el estator está formado por bobinas. Este tipo de motor trabaja a mayor velocidad que los de imán permanente.

Motor paso a paso de imán permanente.- En los motores tipo imán permanente, su rotor es un imán que posee una ranura en toda su longitud y el estator está formado por una serie de bobinas enrolladas alrededor de un núcleo o polo.

Motor paso a paso híbrido.- Es una combinación de los anteriores, logrando un alto rendimiento a una buena velocidad. Según la construcción de las bobinas del estator, tenemos dos tipos: los unipolares y bipolares.

- **Motor paso a paso unipolar.**- Se llaman así porque la corriente que circula por los diferentes bobinados siempre circula en el mismo sentido.

Tienen las bobinas un arrollamiento único, estos se componen de 4 bobinas y 6 cables externos, dos para cada bobina, y otro para cada par de éstas, aunque también se pueden ver con 5 cables, compartiendo el de alimentación para los 2 pares de bobinas (Figura 2.12).

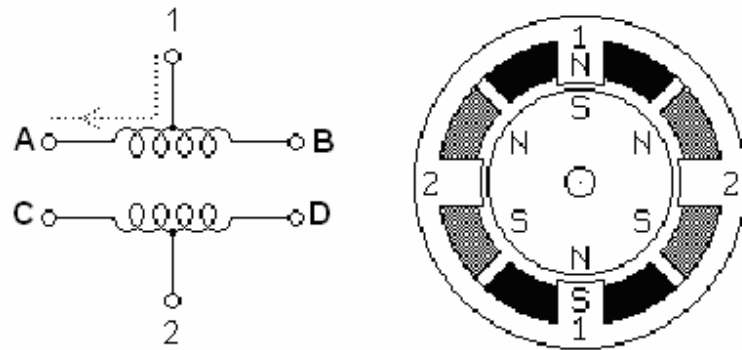


Figura 2.12 Motor paso a paso unipolar

- **Motor paso a paso bipolar.-** La corriente que circula por los bobinados cambia de sentido en función de la tensión que se aplica por lo que un mismo bobinado puede tener en uno de sus extremos distinta polaridad (bipolar). Tienen las bobinas compuestas por dos arrollamientos cada una, y 4 cables de conexión como se muestra en la figura 2.13.

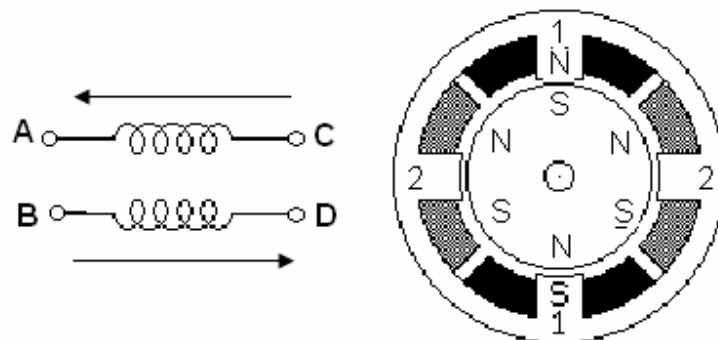


Figura 2.13 Motor paso a paso bipolar

2.1.1.6.3.- Secuencias para manejar motores paso a paso bipolares

Este tipo de motores necesitan la inversión de la corriente que circula en sus bobinas en una secuencia determinada. Cada inversión de la polaridad provoca el movimiento del eje en un paso, cuyo sentido de giro está determinado por la secuencia seguida.

En la tabla 2.21 se muestra la secuencia necesaria para controlar el giro en sentido horario de un motor paso a paso del tipo bipolar.

PASO	TERMINALES			
	A	B	C	D
1	+V	-V	+V	-V
2	+V	-V	-V	+V
3	-V	+V	-V	+V
4	-V	+V	+V	-V

Tabla 2.21 Secuencia de giro en sentido horario de un motor paso a paso bipolar

Para cambiar el giro a sentido antihorario se debe invertir las polaridades de cada terminal (A, B, C, D) como se muestra en la tabla 2.22.

PASO	TERMINALES			
	A	B	C	D
1	-V	+V	-V	+V
2	-V	+V	+V	-V
3	+V	-V	+V	-V
4	+V	-V	-V	+V

Tabla 2.22 Secuencia de giro en sentido antihorario de un motor paso a paso bipolar

2.1.1.6.4. - Secuencias para manejar motores paso a paso unipolares

Existen tres secuencias posibles para este tipo de motores, las cuales se detallan a continuación. Todas las secuencias comienzan nuevamente por el paso 1 una vez alcanzado el paso final (4 u 8). Para revertir el sentido de giro, simplemente se deben ejecutar las secuencias en modo inverso.

- **Secuencia Normal.-** Esta es la secuencia más usada y la que generalmente recomienda el fabricante. Con esta secuencia el motor avanza un paso por vez y debido a que siempre hay al menos dos bobinas activadas, se obtiene un alto torque de paso y de retención como se muestra en la tabla 2.23.

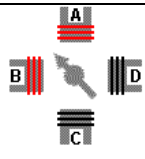
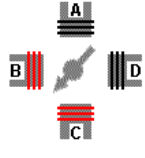
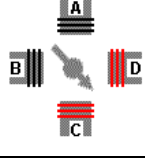
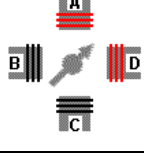
PASO	A	B	C	D	
1	1	1	0	0	
2	0	1	1	0	
3	0	0	1	1	
4	1	0	0	1	

Tabla 2.23 Secuencia normal

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

En la tabla 2.24 se muestra la secuencia del tipo wave drive.

PASO	A	B	C	D	
1	1	0	0	0	
2	0	1	0	0	
3	0	0	1	0	
4	1	0	0	1	

Tabla 2.24 Secuencia del tipo wave drive

- **Secuencia del tipo medio paso.-** En esta secuencia se activan las bobinas de tal forma de brindar un movimiento igual a la mitad del paso real. Para ello se activan primero 2 bobinas y luego solo 1 y así sucesivamente. En la tabla 2.25 se muestra la secuencia que completa consta de 8 movimientos en lugar de 4.

PASO	A	B	C	D	
1	1	0	0	0	
2	1	1	0	0	
3	0	1	0	0	

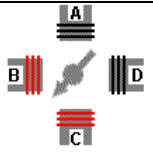
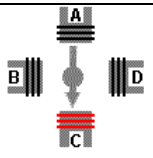
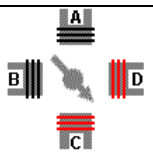
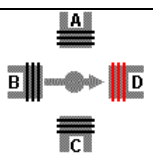
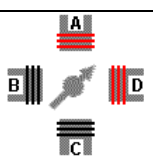
4	0	1	1	0	
5	0	0	1	0	
6	0	0	1	1	
7	0	0	0	1	
8	1	0	0	1	

Tabla 2.25 Secuencia del tipo medio paso

2.1.1.7.- Teclado matricial hexadecimal

2.1.1.7.1.- Introducción

El tamaño de un teclado matricial esta definido por el número de teclas y como éstas se encuentran organizadas.

Un teclado de este tipo consta de 16 teclas (matriz 4 X 4). Por cada fila y cada columna de la matriz hay un "cable" que pasa por detrás de las teclas, las cuales están colocadas en las intersecciones entre filas y columnas como se muestra en la figura 2.14a.

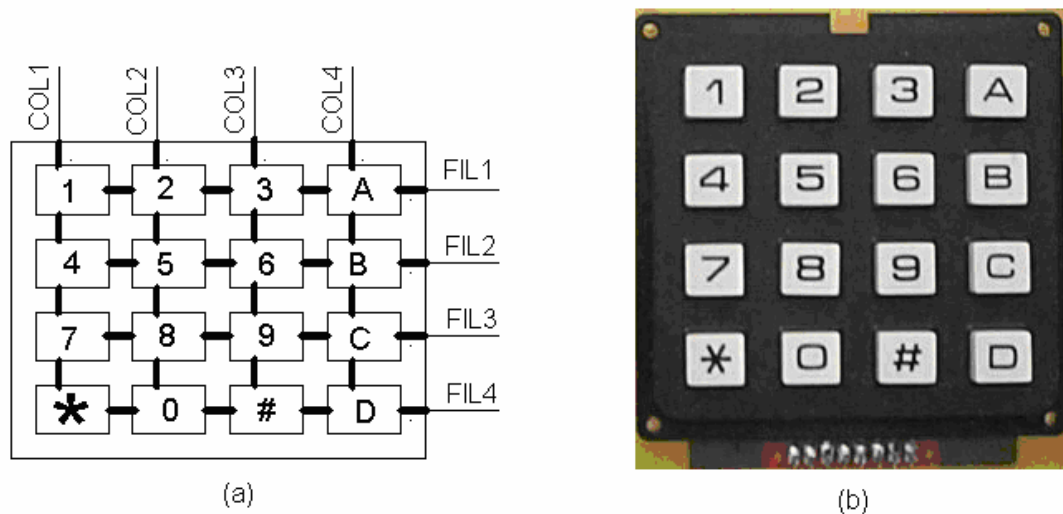


Figura 2.14 Teclado matricial hexadecimal

Típicamente el primer número de la descripción, como por ejemplo 6 x 4, le definirá el número de columnas y el segundo el número de filas. La mayoría de los teclados se leen por una técnica de exploración consistente en ir leyendo consecutivamente las filas o las columnas de éste. Hay circuitos especializados en esta tarea, pero es fácil hacer que un microcontrolador lea estos teclados matriciales

La disposición en matriz de los teclados responde a la necesidad de leer gran cantidad de conmutadores con pocas líneas de entrada, piénsese en que si se necesitase una línea por cada tecla del teclado de un PC, serían necesarias más de 100 líneas.

2.1.1.7.2.- Codificador de teclado 74C922

Este integrado es un decodificador de teclado, este chip tiene como entradas las líneas de filas y columnas del teclado y como salidas cuatro líneas correspondientes a todas las teclas posibles, éste se encuentra escaneando el teclado, esperando que una tecla se presione. Cuando un botón del teclado es presionado el pin 12 del integrado se pone en 1L. Entonces se envía a los pines 14, 15, 16 y 17 un número

binario de 4 bits correspondiente al botón presionado. Normalmente se utiliza mucho el pin 13, ya que éste nos da un indicador de cuando almacenar los datos en los pines 14, 15, 16 y 17. El chip utiliza dos capacitores (pines 5 y 6), el capacitor del pin 5 es para escanear y el del pin 6 es para antirebote. En la figura 2.15 se muestra la distribución de pines del C.I. 74C922.

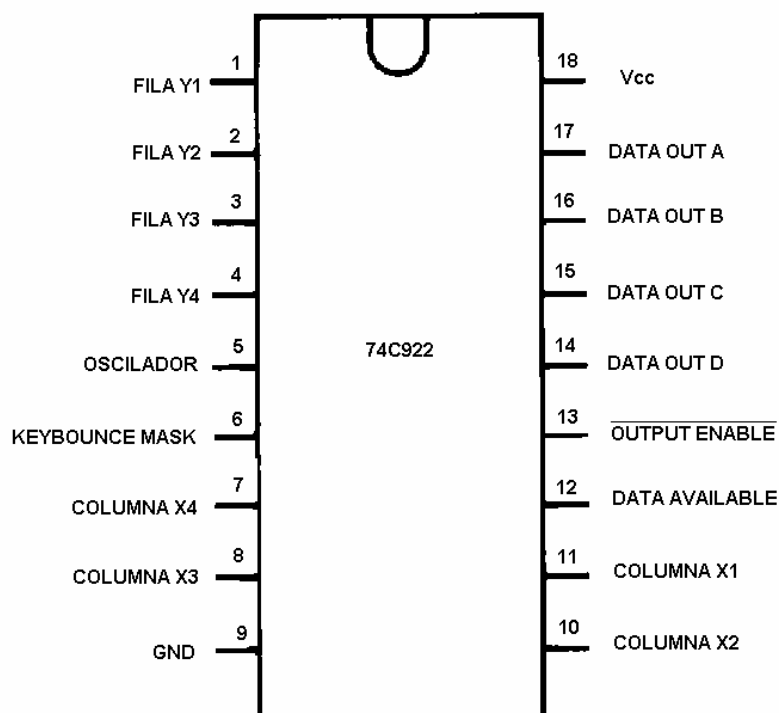


Figura 2.15 Distribución de pines del C.I. 74C922

2.1.2.- TRANSMISIÓN DE DATOS

2.1.2.1. - Introducción

Para el protocolo de transmisión de datos, sólo se tienen en cuenta dos estados de la línea, 0L y 1L, también llamados low y high. Cuando no se transmite ningún carácter, la línea está high. Si su estado pasa a low, se sabe que se transmiten datos.

Por convenio se transmiten entre 5 y 8 bits, pero la BIOS sólo soporta una anchura de datos de 7 u 8 bits. Durante la transmisión, si la línea está low, se envía un 0L y si

está high indica un bit en 1L. Se envía primero el bit menos significativo, siendo el más significativo el último en enviarse.

A continuación se envía un bit de paridad para detectar errores. Finalmente se mandan los bits de stop, que indican el fin de la transmisión de un carácter. El protocolo permite utilizar 1, 1.5 y 2 bits de stop.

El número de bits que componen cada palabra depende del código a emplear, así, si se quieren transmitir datos ASCII normales, serán de 7 bits, pero si se quiere enviar el juego completo de caracteres, cada uno debería estar formada por 8 bits.

Dependiendo de esto, si se emplea paridad y el número de bits de stop, se mandará un determinado número de caracteres dependiendo de la velocidad en baudios (bits por segundo) por el pòrtico serie.

En la figura 2.16 se puede ver con detalle la transmisión del carácter ASCII "A", que corresponde al número 65 del código ASCII y su representación binaria es 01000001.

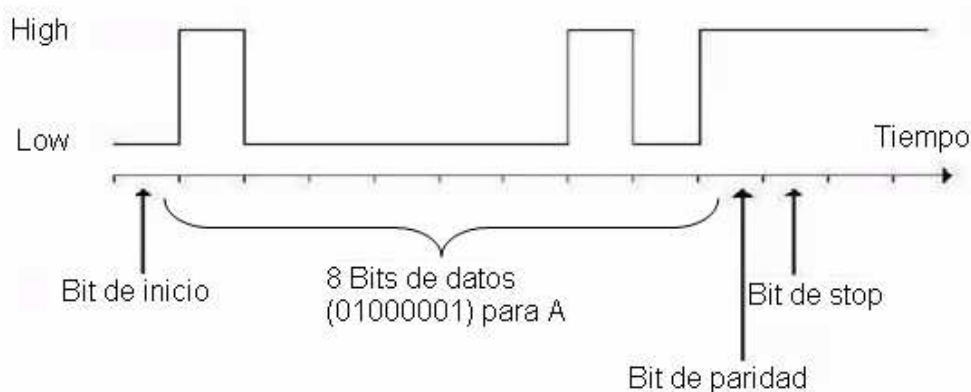


Figura 2.16 Transmisión del carácter ASCII A

El puerto serie es un dispositivo muy extendido, con conector grande o pequeño, todos los equipos PC lo incorporan actualmente. Debido a que el estándar del pòrtico serie se mantiene desde hace muchos años, la institución de normalización

americana (EIA) ha escrito la norma **RS-232-C** que regula el protocolo de la transmisión de datos, el cableado, las señales eléctricas y los conectores en los que debe basarse una conexión RS-232.

La comunicación realizada con el pÓrtico serie es una comunicación asíncrona. Para la sincronización de una comunicación se precisa siempre de una línea adicional a través de la cual el emisor y el receptor intercambian la señal del pulso. Pero en la transmisión serie a través de un cable de dos líneas esto no es posible ya que ambas están ocupadas por los datos y la masa. Por este motivo se intercalan antes y después de los datos informaciones de estado según el protocolo RS-232.

Esta información es determinada por el emisor y receptor al estructurar la conexión mediante la correspondiente programación de sus pÓrticos serie. Esta información puede ser la siguiente:

- **Bit de paridad.-** con este bit se pueden descubrir errores en la transmisión. Se puede dar paridad par o impar. En la paridad par, por ejemplo, la palabra de datos a transmitir se completa con el bit de paridad de manera que el número de bits 1 enviados es par.
- **Bit de parada.-** indica la finalización de la transmisión de una palabra de datos. El protocolo de transmisión de datos permite 1, 1.5 y 2 bits de parada.
- **Bit de inicio.-** cuando el receptor detecta el bit de inicio sabe que la transmisión ha comenzado y es a partir de entonces que debe leer, la transmisión ha comenzado, y es a partir de entonces que debe leer las señales de la línea a distancias concretas de tiempo, en función de la velocidad determinada.

El RS-232 consiste en un conector tipo DB-9 de 9 pines como se muestra en la figura 2.17. Las señales con las que trabaja este pÓrtico serie son digitales, así los niveles de salida son de -5 a -15 voltios para el 1 lógico y +5 a +15 voltios para el 0 lógico, mientras que los niveles de entrada son -3 a -15 voltios para un 1 lógico y +3 a +15 voltios para un 0 lógico.

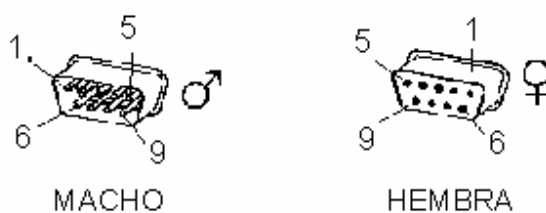


Figura 2.17 Conector DB 9

Cada pin puede ser de entrada o de salida, teniendo una función específica cada uno de ellos. Las señales TXD, DTR y RTS son de salida, mientras que RXD, DSR, CTS y DCD son de entrada como se muestra en la tabla 2.26. La masa de referencia para todas las señales es SG (Tierra de Señal).

Numero de Pin	Señal	Descripción	E/S
3	TxD	Transmisión de datos	S
2	RxD	Recepción de datos	E
7	RTS	Solicitud de envío	S
8	CTS	Autorización de envío	E
6	DSR	Datos listos para enviar	E
5	SG	Tierra (GND)	-
1	CD/DCD	Detección de portadora	E
4	DTR	Terminal de datos listo	S
9	RI	Detector de llamada	E

Tabla 2.26 Descripción de cada pin del conector DB 9

2.1.2.2. - Conversor TTL - RS232

El MAX 232 es un transmisor/receptor que incluye un generador de voltaje capacitivo que suministra niveles de voltaje del estándar EIA – 232 desde solo 5V, de esta manera es un interfaz de conversión que cambia los niveles lógicos TTL a los estándares RS – 232 cuando se realiza una transmisión hacia la PC desde el circuito

en comunicación, y cambia los niveles RS – 232 a TTL cuando tiene una recepción desde la PC. Además este circuito integrado lleva internamente 2 conversores de nivel de TTL a RS – 232 y otros 2 de RS – 232 a TTL con lo que en total podremos manejar 4 señales del pòrtico serie del PC

Distribución de pines.- El MAX 232 tiene un encapsulado tipo DIP de 16 pines como se muestra en la figura 2.18.

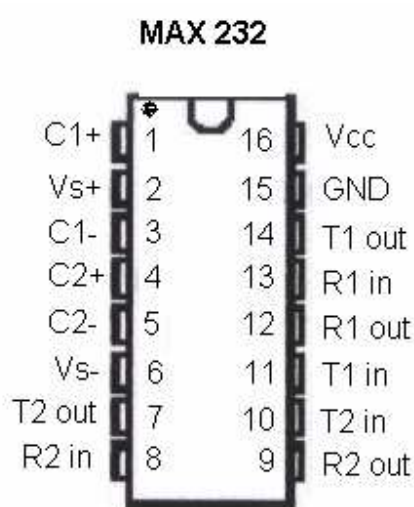


Figura 2.18 Distribución de pines del C.I. MAX 232

2.1.3- CONSTRUCCIÓN Y CONEXIONES DE CADA TARJETA

2.1.3.1.- Tarjeta principal

La tarjeta principal, figura 2.33, consta de una entrada de alimentación de 5v de corriente continua. Tiene entrada de datos desde el teclado hexadecimal que lee el microprocesador, un control que sirve para seleccionar uno de los dispositivos de salida como son el motor paso a paso, matriz de led's, LCD, display de 7 segmentos, y además posee conexión al circuito integrado MAX 232 para la comunicación serial desde el computador.

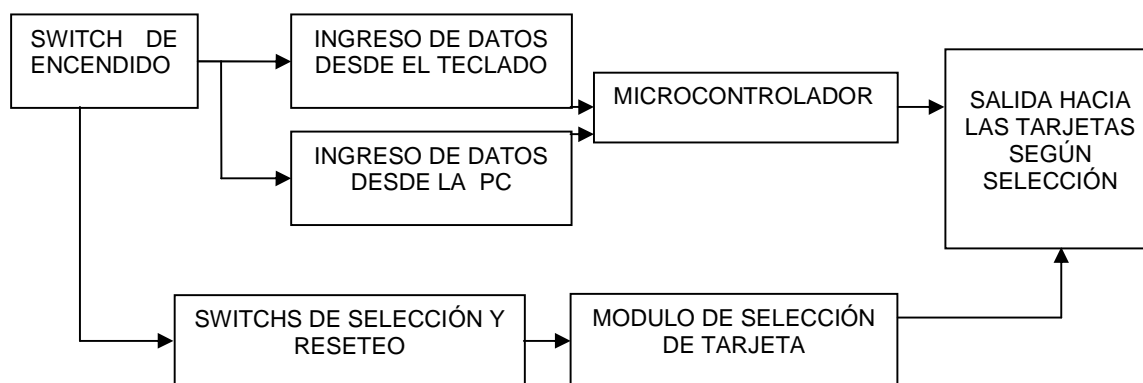


Figura 2.19 Diagrama de bloques de la tarjeta principal

Switch de encendido.- Sirve para el encendido de la tarjeta principal y demás tarjetas del módulo entero. Para esto se cuenta con un switch que se encuentra en las líneas de energía, que al mismo tiempo enciende un led amarillo que indica que el módulo está encendido y listo para ser utilizado.

Teclado.- De la tarjeta principal sale un bus de datos de 8 líneas que se conecta con un teclado hexadecimal para poder ingresar los datos al microcontrolador.

Estas 8 líneas están conectadas hacia los 8 pines de entrada que posee el integrado codificador de teclado 74C922, 4 líneas son para las columnas (X1, X2, X3, X4) y las otras 4 líneas son para las filas (Y1, Y2, Y3, Y4). Este integrado ya está polarizado debido a que se encuentra en la tarjeta principal y conectada a dos condensadores en los pines (KBM, OSC) para proporcionar el tiempo de "debounce", y para darnos la frecuencia a la cual trabajará el circuito integrado. También se conectan los pines (OE) directamente hacia GND y (DA) a una de las interrupciones del microcontrolador.

Las salidas (A, B, C, D) del C.I. 74C922 codificador de teclado, van por caminos de conexión independientes hacia los cuatro pines del pòrtico uno del microcontrolador (P1.0, P1.1, P1.2, P1.3) respectivamente, donde A va al pòrtico P1.0 que es el bit

menos significativo, B a P1.1, C a P1.2, D a P1.3. El resto de los pines del p rtico 1 quedan sueltos.

Para poder utilizar el teclado con cualquiera de los dispositivos de salida se utiliza una compuerta NAND de 8 entradas (CI 7430). Los 4 switches de control (C1,C2,C3, C4) que controlan el encendido de los dispositivos de salida del m dulo, van conectados independientes a cuatro entradas de la compuerta NAND y las 4 entradas restantes van conectadas a 1L, mientras la salida estar  conectada al pin 12 del codificador 74C922 que representa al DATA AVAILABLE.

De esta manera al encender el m dulo se tiene a las entradas de la compuerta NAND 1L y por ende a la salida 0L, pero cuando se presiona cualquiera de los switches se tiene a la salida de la compuerta NAND 1L, de esta manera se activa el codificador de teclado.

Microcontrolador.- Todas las conexiones hacia el microcontrolador se hacen por medio de un z calo ZIF donde el microcontrolador va sobrepuesto para facilitar su movilidad. Este integrado ya est  polarizado de igual manera que el CI 74C922 debido a que se encuentra en la tarjeta principal. La entrada de reset (RST) est  conectado a Vcc por medio de un condensador y a GND a trav s de una resistencia, para el control del reseteo se coloca un switch que est  conectado desde Vcc, como se muestra en la figura 2.20.

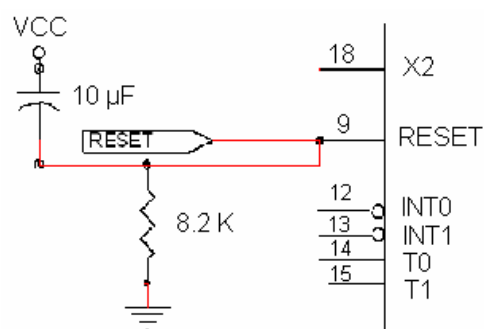


Figura 2.20 Conexi n del reset

Para que el reseteo funcione correctamente se ubica en nivel alto (1L) durante dos ciclos de m quina mientras el oscilador est  en funcionamiento.

A las entradas (XTAL1, XTAL2) se conecta el cristal oscilador que dará la frecuencia de trabajo, éstos se conectan a dos condensadores como muestra la figura 2.21.

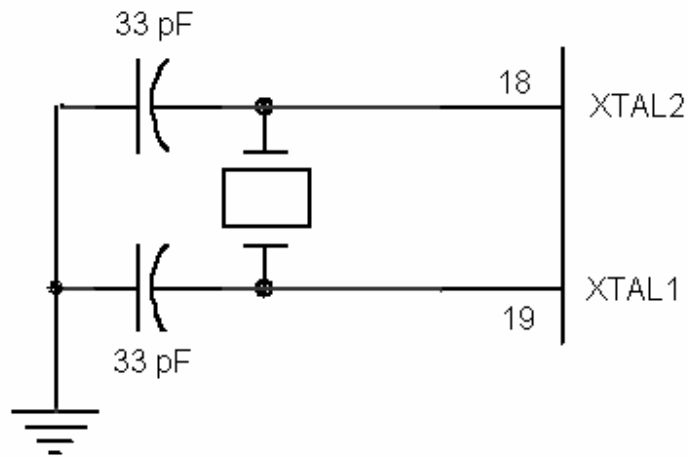


Figura 2.21 Conexión del cristal

El valor de los condensadores a utilizarse es de 33 pf con un rango de tolerancia del $\pm 10\%$.

Los pórtricos P0 y P2 se utilizan para la comunicación hacia los dispositivos de salida del módulo teniendo como bits menos significativos (LSB) al P0.0 y P2.0 respectivamente.

Los pines que se utilizan para cada salida son los siguientes:

- Para el motor paso a paso, cuatro pines del pórtrico P0 (P0.0, P0.1 P0.2, P0.3).
- Para la matriz de led's, siete pines del pórtrico P0 (P0.0, P0.1 P0.2, P0.3, P0.4, P0.5, P0.6) y cinco pines del pórtrico P2 (P2.0, P2.1 P2.2, P2.3, P2.4).
- Para los display's de 7 segmentos, ocho pines el pórtrico P0 (P0.0, P0.1 P0.2, P0.3, P0.4, P0.5, P0.6, P0.7) y cuatro del pórtrico P2 (P2.0, P2.1 P2.2, P2.3).

- Para el LCD, ocho pines del p rtico P0 (P0.0, P0.1, P0.2, P0.3, P0.4, P0.5, P0.6, P0.7) y dos del p rtico P3 (RD) y (WR) que van conectados a sus correspondientes en el LCD (RS y E) respectivamente.
- Para la tarjeta transmisi n y recepci n serial, los pines 10 (Rx) y 11 (Tx) que corresponden al P3.0 y P3.1 respectivamente.

El p rtico P0 necesita de resistencias de protecci n para evitar quemar sus salidas. Las entradas ALE/PROG y PSEN sirven para manejo de la memoria externa por lo que no se utilizan.

Selecci n de salidas.- La selecci n para cada uno de los dispositivos de salida viene desde un circuito a base que est  dise ado con los siguientes elementos: los circuitos integrados C.I. 7474 y C.I. 7411, resistencias y de cuatro switches.

La alimentaci n de voltaje (Vcc) va conectada a un extremo de las resistencias (R1, R2, R3, R4) como se observa en la figura 2.22, mientras el otro extremo de la resistencia va conectado a un pin de los switches de control (SW1, SW2, SW3 y SW4), y el pin restante de los switches se conecta a GND como se observa en la figura 2.22. Los switches de control presentan un estado normalmente abierto mientras no sean pulsados o activados.

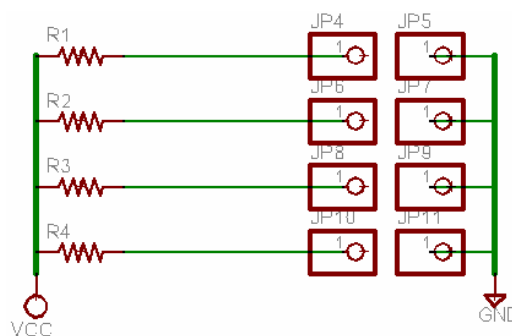


Figura 2.22 Conexi n de los switches de selecci n de salidas

Los switches de control están distribuidos de la siguiente manera, para controlar a cada tarjeta del módulo, como se muestra en la tabla 2.27.

Switch de control	Tarjeta que controla
C1	Motor paso a paso
C2	Matriz de led's
C3	LCD
C4	Display de 7 segmentos

Tabla 2.27 Switchs de control

De la unión entre la resistencia y switch salen las líneas que se dirigen hacia la entrada de los CLK de los FLIP-FLOPS tipo D. Ejemplo:

La unión Resistencia – Switch (1) se conecta al FLIP-FLOP (1), al pin de la entrada señal de reloj (CLK), de la misma manera se conectarán las demás uniones Resistencia – Switch hacia los FLIP-FLOPS respectivos como se muestra en la figura 2.23.

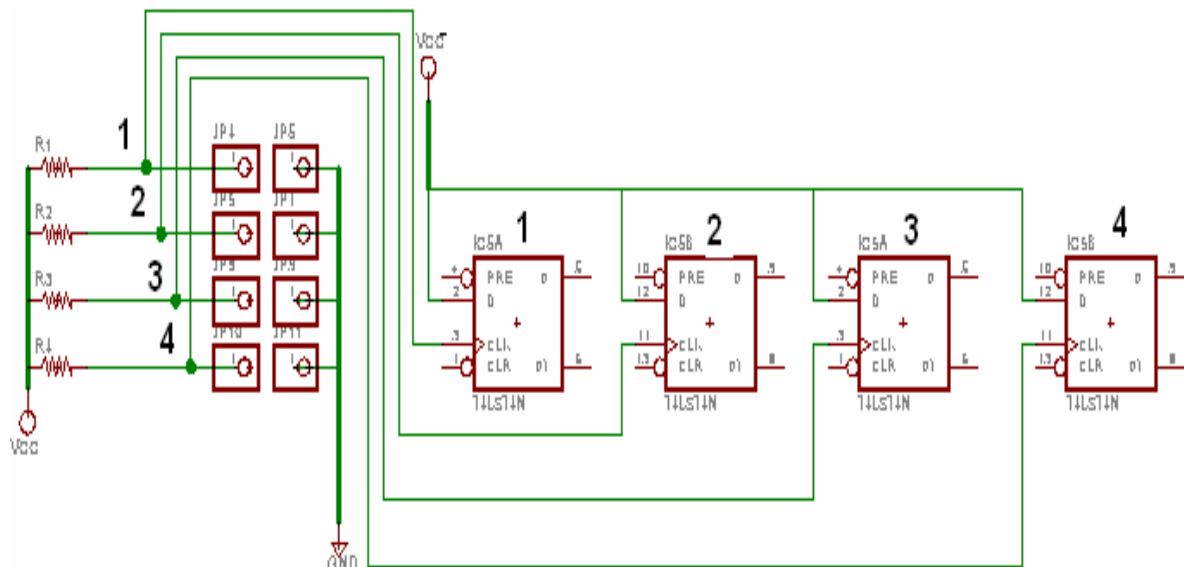


Figura 2.23 Conexión del reloj (CLK)

A la entrada D de los FLIP-FLOPS llega la línea de Vcc por lo que se tiene como dato un (1L). Se trabaja con transición negativa ya que se tiene 1L en las entradas del reloj.

Se utiliza la salida Q' de cada FLIP-FLOP debido a que en la entrada tipo D se tiene 1L y se necesita trabajar con 0L a la salida para activar a la compuerta del buffer de los C.I. 74244.

Para controlar una salida a la vez se lo hace mediante cuatro compuertas AND de 3 entradas, las cuales controlan los clear de los FLIP-FLOPS de la siguiente manera: a las entradas de la primera compuerta AND se conectan los switches 2, 3 y 4, para la segunda compuerta AND se conectan los switches 1, 3 y 4, a la tercera compuerta corresponden los switches 1, 2 y 4 y para la cuarta compuerta, los switches 1, 2 y 3, como se observa en la figura 2.24.

Con esta secuencia al momento de encender el módulo todas las entradas de los switches se encuentran en 1L por lo que a la salida de cualquiera de estas compuertas AND (7411) se tiene 1L, y al momento de poner cualquier switch de control a 0L, las compuertas AND respectivas donde ésta se encuentre conectada pasa a 0L sus salidas activando así los clears de los FLIP-FLOPS. Por ejemplo: Al activar el switch 1 las compuertas 2, 3 y 4 presentan a la salida 0L y por lo tanto activan al clear de los FLIP-FLOPS 2, 3 y 4, logrando que permanezca en funcionamiento solamente el FLIP-FLOP 1.

El mismo procedimiento se realiza en cada uno de los FLIP-FLOPS restantes 2, 3, y 4 que poseen el mismo funcionamiento activando y desactivando los FLIP-FLOPS que son necesarios.

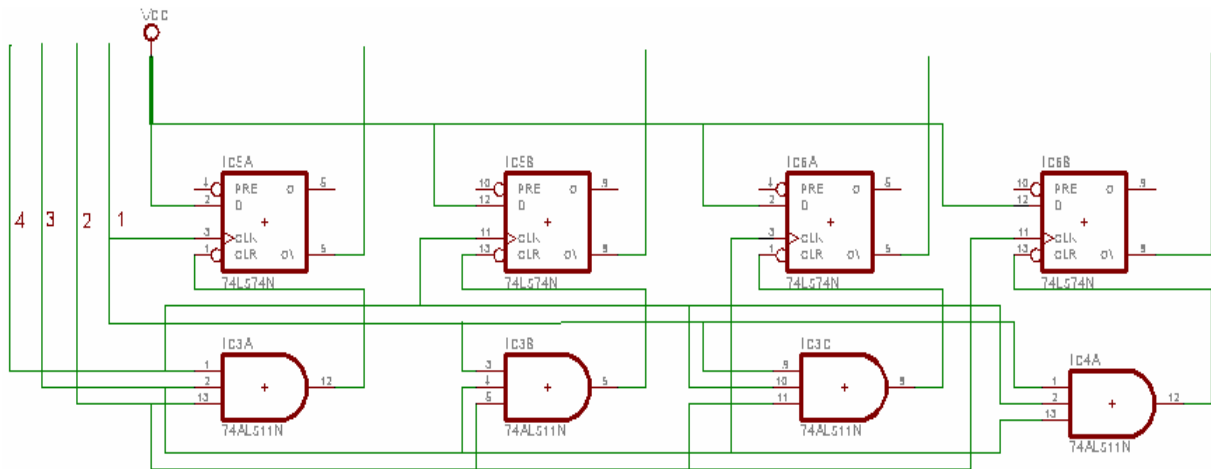


Figura 2.24 Control de la selección de salidas

Desde la salida Q' de los FLIP-FLOPS sale una línea hacia un zócalo que sirve para conectar las tarjetas de salida, donde se comanda el encendido de cada una de estas tarjetas por los switches (C1, C2, C3, C4).

Transmisión Serial.- Para la transmisión serial se utiliza el pin 10 (RX) y el pin 11 (TX) de salida del microcontrolador AT89C51.

Estos van conectados a la tarjeta MAX 232 que convierte la señal TTL a señal binaria.

2.1.3.2.- Tarjeta motor paso a paso

Esta tarjeta figura 2.35, consta de entradas de datos que provienen de las salidas del microcontrolador. Estos datos llegan al integrado 74LS244 que son los buffer que permiten el paso de los datos que ingresan al motor paso a paso pasando por un circuito integrado limitador de corriente L293D como se muestra en la figura 2.25.

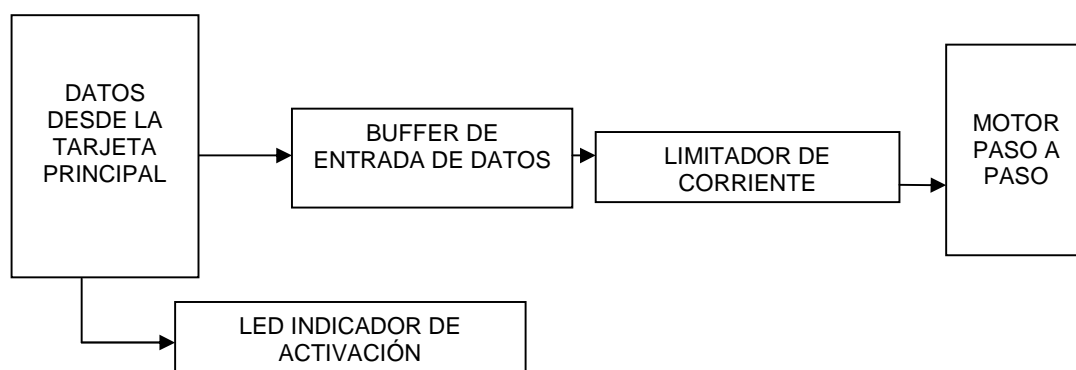


Figura 2.25 Diagrama de bloques de la tarjeta motor paso a paso

Conexiones.- En esta tarjeta se encuentra un zócalo donde llegan las pistas desde el microcontrolador ubicado en la tarjeta principal.

Los datos llegan por los dos lados de la placa; y Vcc y GND se usan para polarización de toda la placa; del pórtico P0, llegan cuatro pines (P0.0, P0.1, P0.2, P0.3), una línea C1 para la habilitación de la compuerta del buffer CI. 74LS244.

Los datos del pórtico P0 van a las entradas de los buffers que se activan por un 0L que recibe desde C1 en los pines 1G', 2G' del CI. 74244 permitiendo así pasar los datos de P0 a las salidas de cada uno de los buffers que están conectadas a las entradas de cada limitador de corriente, las salidas de éstos van a través de diodos a las entradas del motor paso a paso.

Para conocer el estado de la tarjeta se utilizo 1 Led de dos colores, cuando se encienda el led rojo indica el estado inactivo de la tarjeta y cuando se enciende el led verde indica que la tarjeta se encuentra en funcionamiento. Esta conexión se realiza con el control C1 la cual está conectada hacia el ánodo del led rojo mientras el cátodo va por medio de una resistencia hacia GND. Para el led verde se conecta C1 por medio de un inversor CI. 7404 al ánodo del led y el cátodo va por medio de una resistencia hacia GND, como se muestra en la figura 2.26.

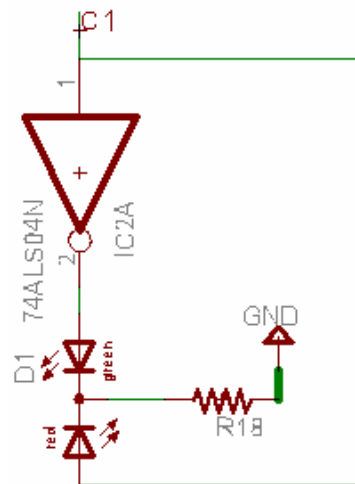


Figura 2.26 Circuito On/Off de la tarjeta motor paso a paso

Cuando el módulo no está en funcionamiento se recibe 1L del control C1 prendiendo el led rojo, pero cuando se recibe 0L del control C1 éste pasa por el inversor y prende el led verde.

2.1.3.3.- Tarjeta matriz de led's

La tarjeta matriz de led's figura 2.37, consta de entradas de datos que provienen de las salidas del microcontrolador. Estos datos llegan a los integrados 74LS244 que son los buffer que permiten el paso de los datos de columnas y filas, es activado desde el control de selección de salidas Q' de los FLIP FLOP ubicados en la tarjeta madre.

Por medio de transistores PNP y NPN pasan los datos a la matriz de Led's. Como se muestra en la figura 2.27.

Esta, como el resto de tarjetas tiene un LED indicador.

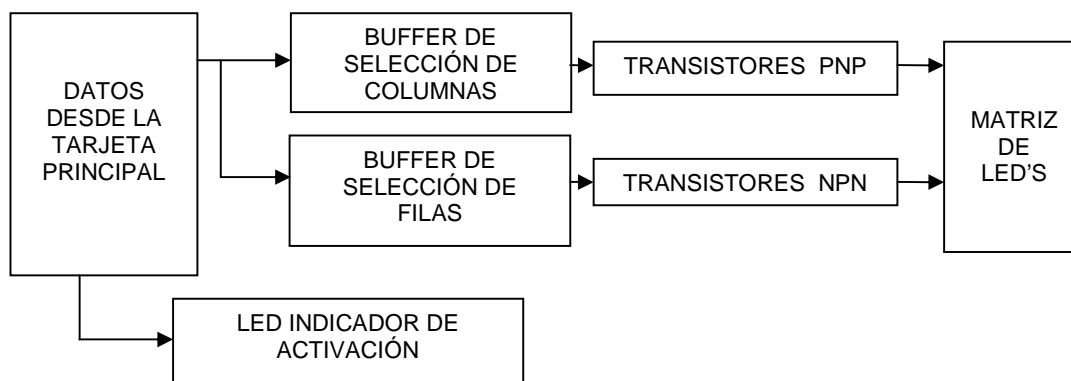


Figura 2.27 Diagrama de bloques de la tarjeta matriz de led's

Conexiones.- En esta tarjeta se encuentra un zócalo donde llegan las pistas desde el microcontrolador ubicado en la tarjeta principal.

Los datos llegan por los dos lados de la placa; Vcc y GND se usan para polarización de toda la placa, del pórtico P0, siete pines (P0.0, P0.1, P0.2, P0.3, P0.4, P0.5, P0.6) portadores de la información y del pórtico P2, 5 pines (P2.0, P2.1, P2.2, P2.3, P2.4) para el encendido de las columnas, una línea (C2) para la habilitación de los buffers del CI 74LS244.

Los datos de los pórticos P0 y P2 van a las entradas de los buffers que se activan por un 0L que recibe desde C2 en los pines 1G', 2G', 3G', 4G' del CI. 74244 permitiendo así pasar los datos de P0 y P2 a las salidas de cada uno de los buffers que se conectan a las columnas y filas de la matriz de LED's respectivamente a través de resistencias. Los datos a la salida de los buffer pasan a través de las resistencias y se colocan sobre la base de los transistores tipo PNP y desde ese momento comienza a circular corriente desde emisor hacia el colector para las columnas poniendo Vcc en el ánodo del led de la matriz, y para las filas se colocará la señal en la base del transistor tipo NPN el cual hará que se conecte el cátodo con el punto de GND, encendiéndose el LED seleccionado y evitando problemas de corriente.

Para conocer el estado de la tarjeta se utiliza 1 Led de dos colores.

Cuando se enciende el led rojo indica el estado inactivo de la tarjeta y cuando se enciende el led verde indica que ésta entra en funcionamiento.

Esta conexión se realiza con el control C2 que va conectado hacia el ánodo del led rojo mientras el cátodo va por medio de una resistencia hacia GND.

Para el led verde se conecta C2 por medio de un inversor CI. 7404 al ánodo del led y el cátodo va por medio de una resistencia hacia GND, como se muestra en la figura 2.28.

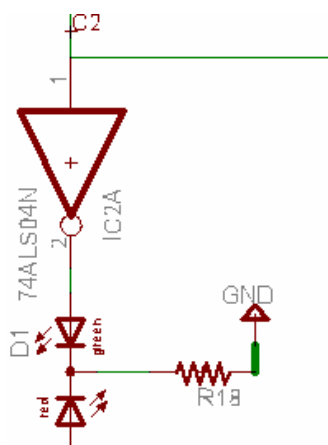


Figura 2.28 Circuito On/Off de la tarjeta matriz de led's

Cuando el módulo no está en funcionamiento se recibe 1L del control C2 prendiendo el led rojo, pero cuando se recibe 0L del control C2 éste pasa por el inversor y prende el led verde.

2.1.3.4.- Tarjeta LCD

La tarjeta del módulo LCD figura 2.39, consta de entradas de datos que provienen de las salidas del microcontrolador a través de los buffer, y control de contraste de la intensidad de luminosidad del LCD como se muestra en la figura 2.29.

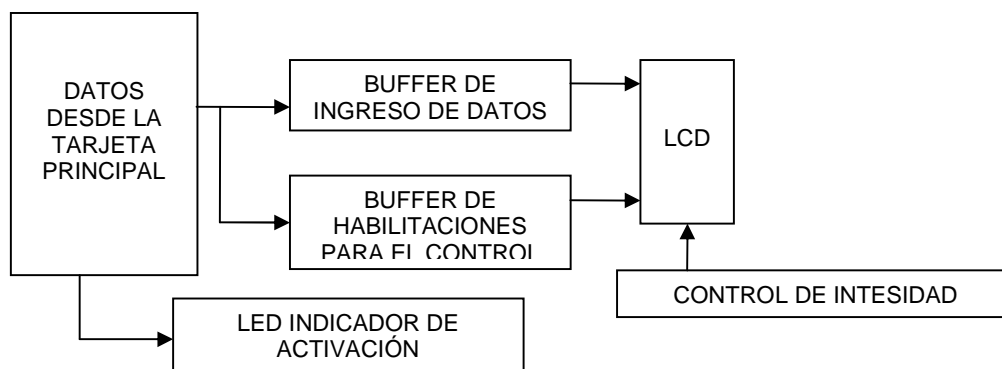


Figura 2.29 Diagrama de bloques de la tarjeta LCD

Conexiones.- En esta tarjeta se encuentra un zócalo a donde llegan las pistas desde el microcontrolador ubicado en la tarjeta principal.

Los datos llegan por los dos lados de la placa; Vcc y GND, se usan para polarización de toda la placa, del pórtico P0 ocho pines (P0.0, P0.1, P0.2, P0.3, P0.4, P0.5, P0.6, P0.7) portadores de la información, RW y RD para manipular las habilitaciones de lectura y escritura en el LCD directamente del microcontrolador, una línea (C3) para la habilitación de los buffers del CI 74LS244.

Los datos del P0 van a las entradas de los buffers que se activan por un 0L que recibe desde C3 en los pines 1G', 2G', 3G', permitiendo así pasar los datos a las salidas de cada uno de los buffers.

De la salida de las buffers van los 8 pines del pórtico P0 hacia los datos de entrada en el módulo LCD desde DB0 a DB7 desde el pórtico P0.0 a P0.7 respectivamente.

El resto de entradas para manipulación del LCD son RS y E, pasando por los buffers respectivos.

De igual forma que para las tarjetas anteriores para conocer el estado de la tarjeta se utilizan 2 Led's de diferentes colores, sabiéndose que cuando se enciende el led rojo la tarjeta no puede ser utilizada y cuando se enciende el led verde ésta entra en funcionamiento.

Esta conexión se realiza con el control C3 que va conectado hacia el ánodo del Led rojo mientras el cátodo va por medio de una resistencia hacia GND.

Para el led verde se conecta C3 al ánodo del led por medio de un inversor CI7404 y el cátodo va por medio de una resistencia hacia GND, como se muestra en la figura 2.30.

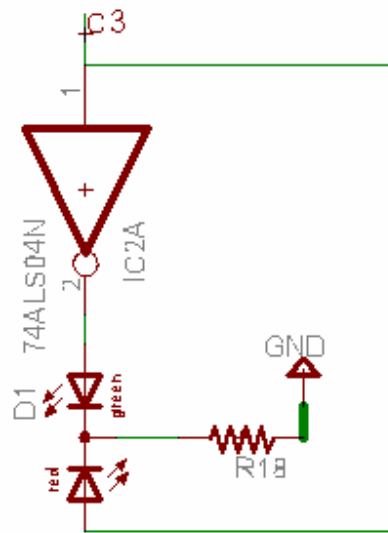


Figura 2.30 Circuito On/Off de la tarjeta LCD

Cuando el módulo no está en funcionamiento se recibe 1L del control C3 prendiendo el led rojo indicando el estado de inactividad de la placa, pero cuando se reciba 0L del control C3 éste pasa por el inversor y prende el led verde indicando el estado activo de la placa.

2.1.3.5.- Tarjeta display de siete segmentos

Esta tarjeta para el arreglo de display's figura 2.41 consta de entradas de datos que provienen de las salidas del microcontrolador. Estos datos llegan a las entradas de los buffers que controlan los 8 segmentos del display incluyendo el punto, y la habilitación de uno de los 4 display's como se muestra en la figura 2.31.

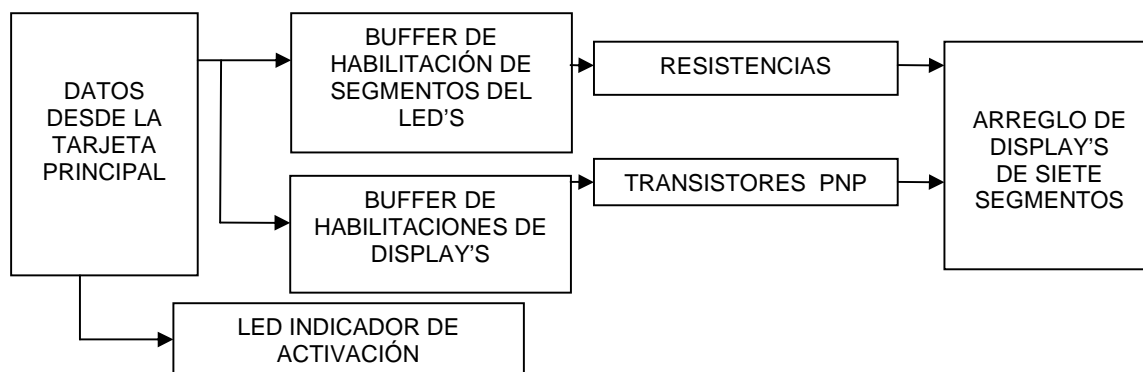


Figura 2.31 Diagrama de bloques de la tarjeta display de siete segmentos

Conexiones.- En esta tarjeta se encuentra un zócalo donde llegan las pistas desde el microcontrolador ubicado en la tarjeta principal.

Los datos llegan por los dos lados de la placa; Vcc y GND se usan para polarización de toda la placa, del pórtico P0, llegan ocho pines (P0.0, P0.1, P0.2, P0.3, P0.4, P0.5, P0.6, P0.7) portadores de la información, y del pórtico P2, llegan cuatro pines (P2.0, P2.1, P2.2, P2.3) que controlan el encendido de los display's, y también una línea (C4) para la habilitación de los buffers del CI 74LS244.

Los datos de los pórticos P0 y P2 van a las entradas de los buffers que se activan por un 0L que recibe desde C4 en los pines 1G', 2G', 3G' del CI. 74244, permitiendo así pasar los datos de P0 y P2 a las salidas de cada uno de los buffers.

Los datos de los ocho pines del pórtico P0 controlan los ocho segmentos del display (a, b, c, d, e, f, g) incluyendo el punto, teniendo un orden de P0.0 para el segmento "a" y sí sucesivamente hasta llegar al segmento punto al cual corresponde P0.7.

Los datos del pórtico P2 controlan el encendido de los display's pasando por resistencias que están conectadas a la base de los transistores PNP que en el lado del emisor van conectadas a Vcc y del lado del colector van conectadas a las entradas de ánodo común que se encuentran en los display's.

Para conocer el estado de la tarjeta se utiliza 1 Led de dos colores, cuando se enciende el led rojo indica el estado inactivo de la tarjeta mientras que cuando se enciende el led verde indica el estado activo de la misma.

Esta conexión se realiza con el control C4 que va conectado hacia el ánodo del Led rojo mientras el cátodo va por medio de una resistencia hacia GND.

Al ánodo del led verde se conecta C4 por medio de un inversor CI. 7404, y al cátodo va conectado por medio de una resistencia hacia GND, como se muestra en la figura 2.32.

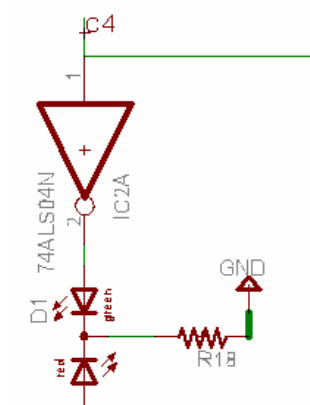


Figura 2.32 Circuito On/Off de la tarjeta display de siete segmentos

2.1.4.- GRÁFICAS DE LAS TARJETAS

2.1.4.1.- Tarjeta principal

Diagrama esquemático

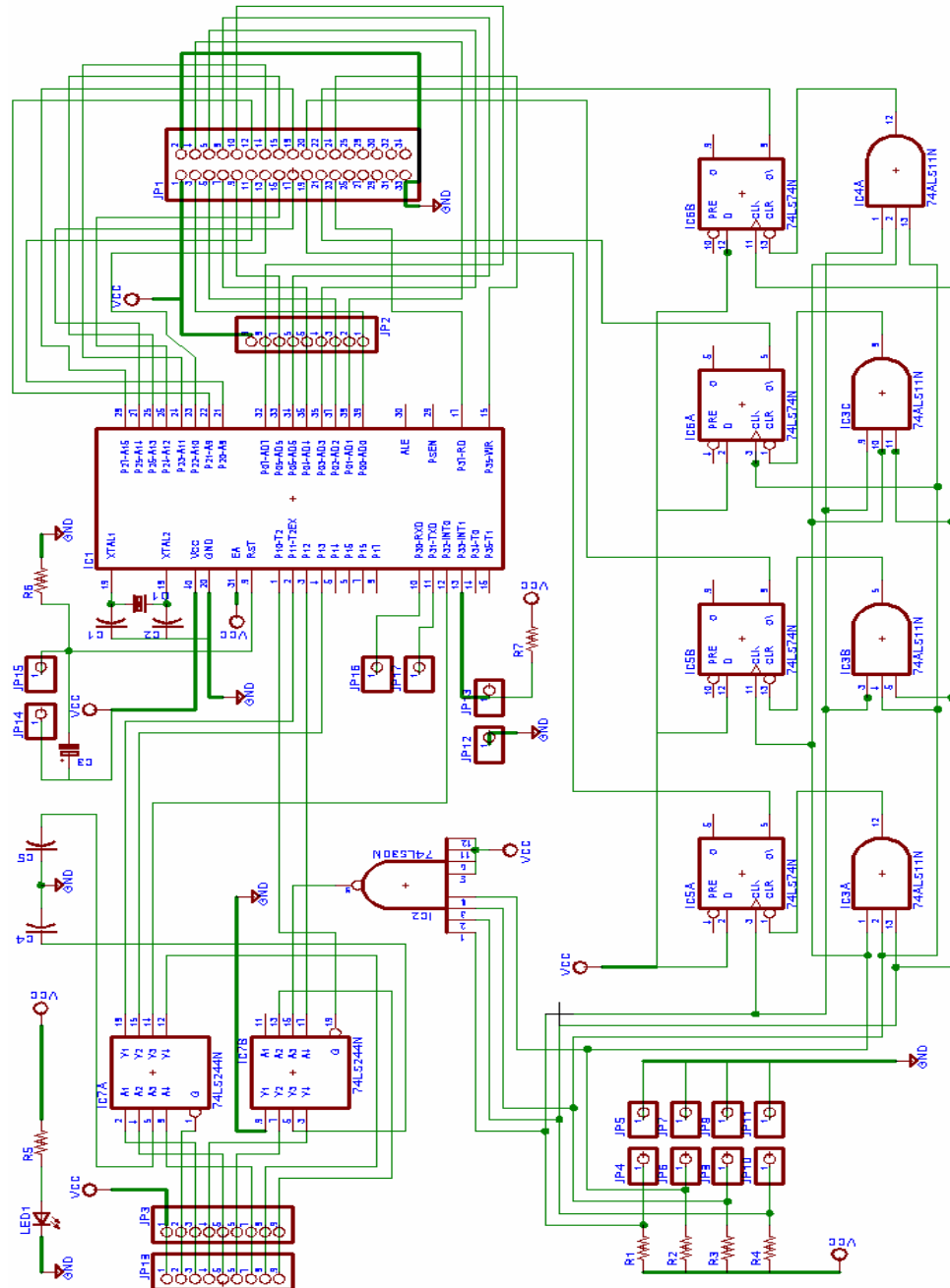


Figura 2.33 Diagrama de la tarjeta principal

Diagrama board

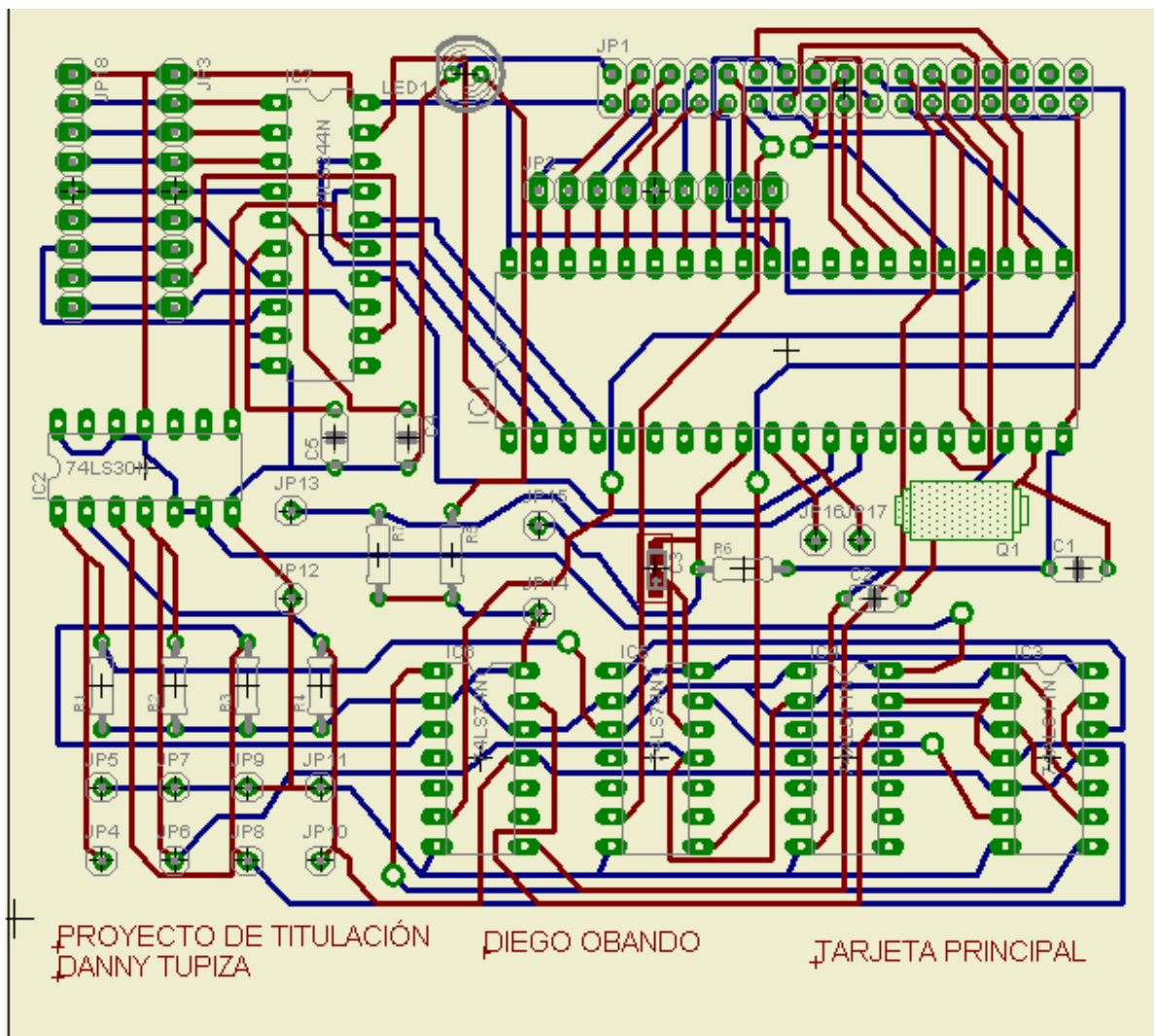


Figura 2.34 Diagrama board de la tarjeta principal

2.1.4.2.- Tarjeta motor paso a paso

Diagrama esquemático

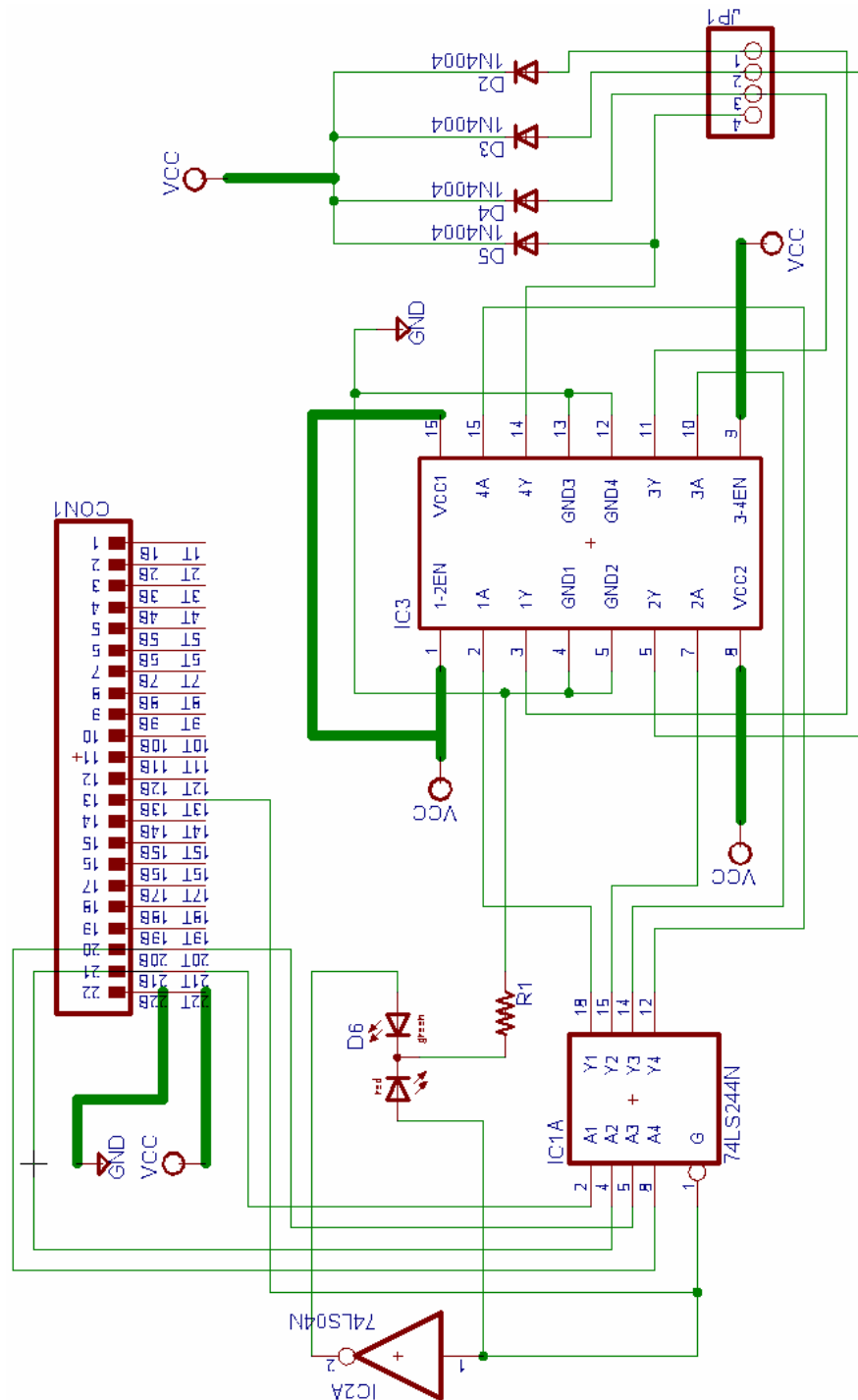


Figura 2.35 Diagrama esquemático de la tarjeta motor paso a paso

Diagrama board

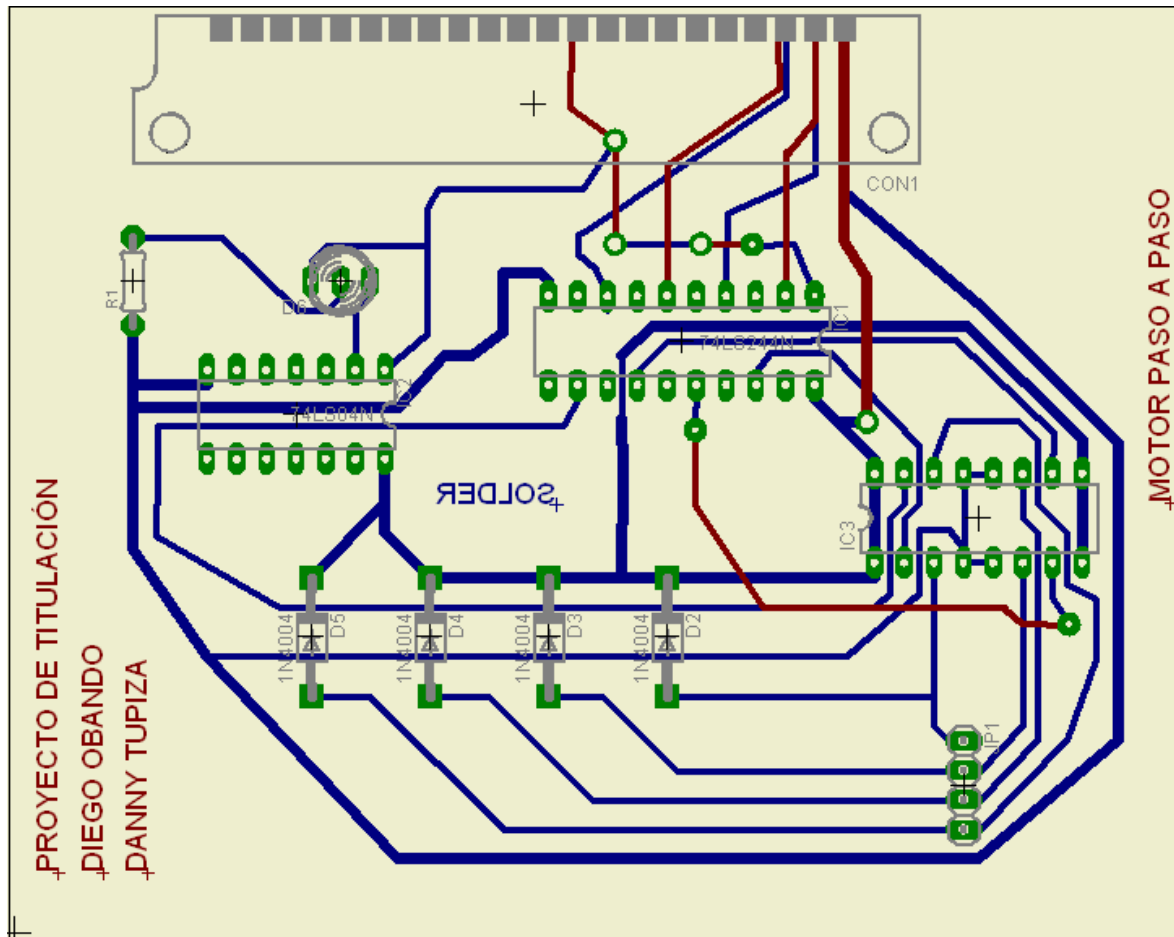


Figura 2.36 Diagrama board de la tarjeta motor paso a paso

2.1.4.3.- Tarjeta matriz de led's

Diagrama esquemático

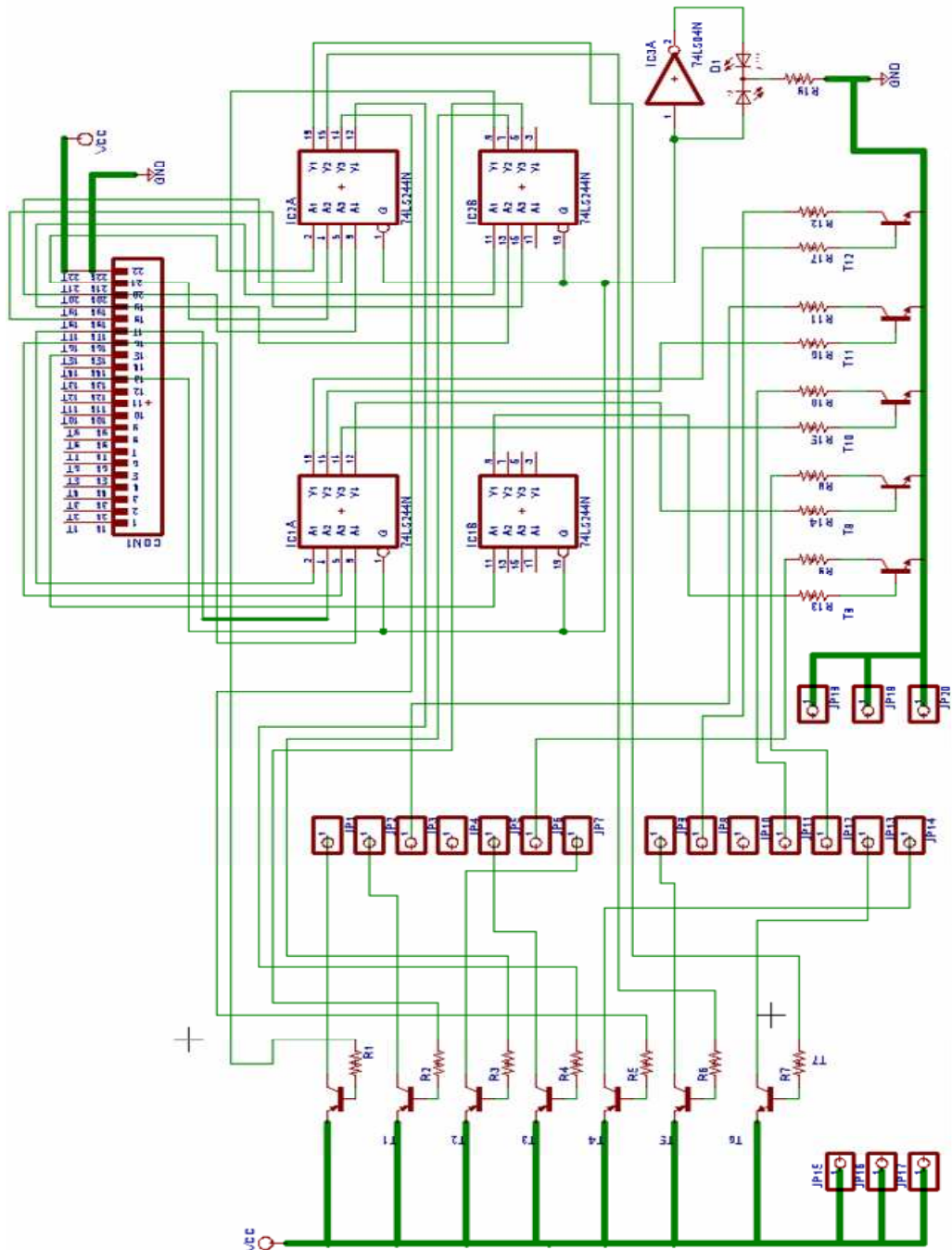


Figura 2.37 Diagrama esquemático de la tarjeta matriz de led's

Diagrama board

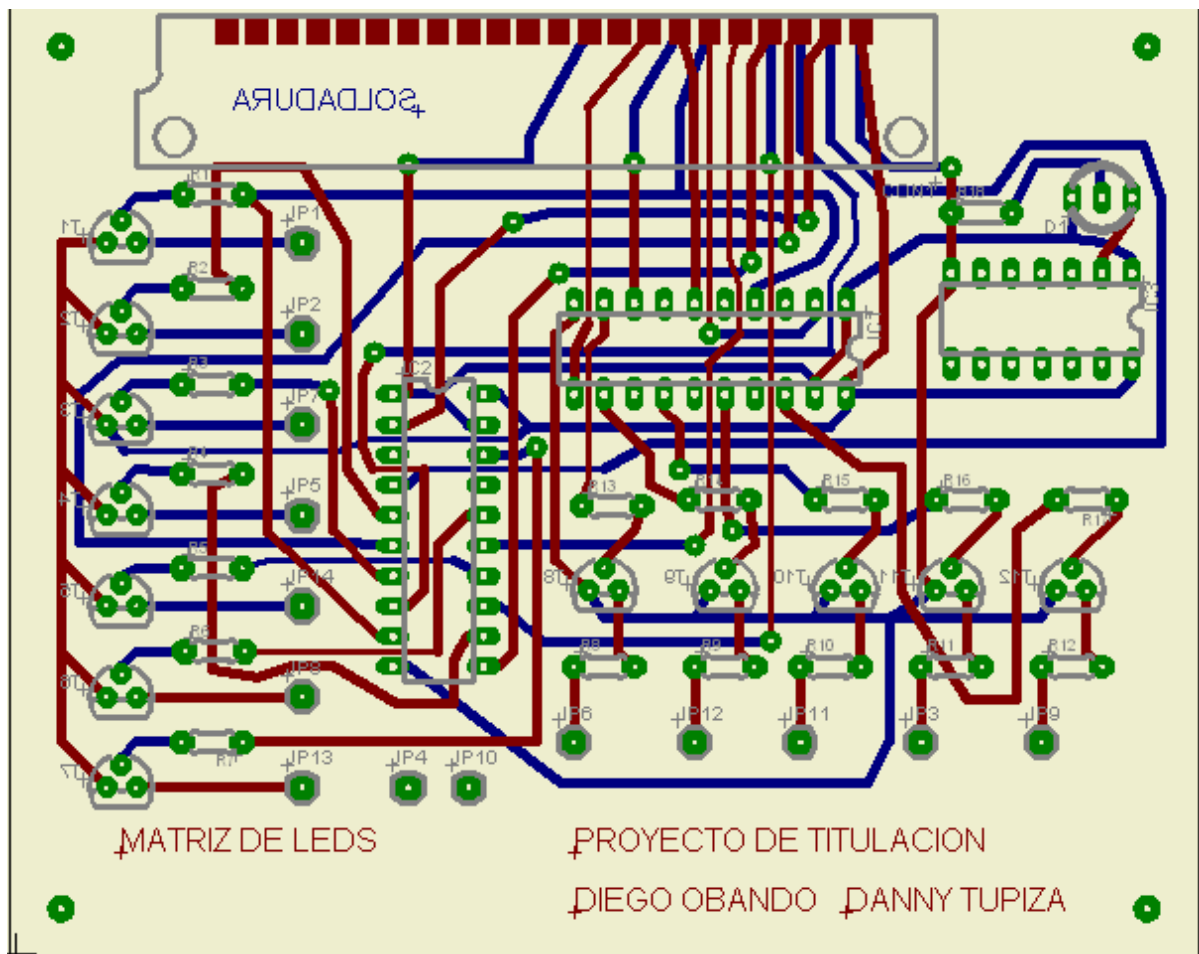


Figura 2.38 Diagrama board de la tarjeta matriz de led's

2.1.4.4.- Tarjeta LCD

Diagrama esquemático

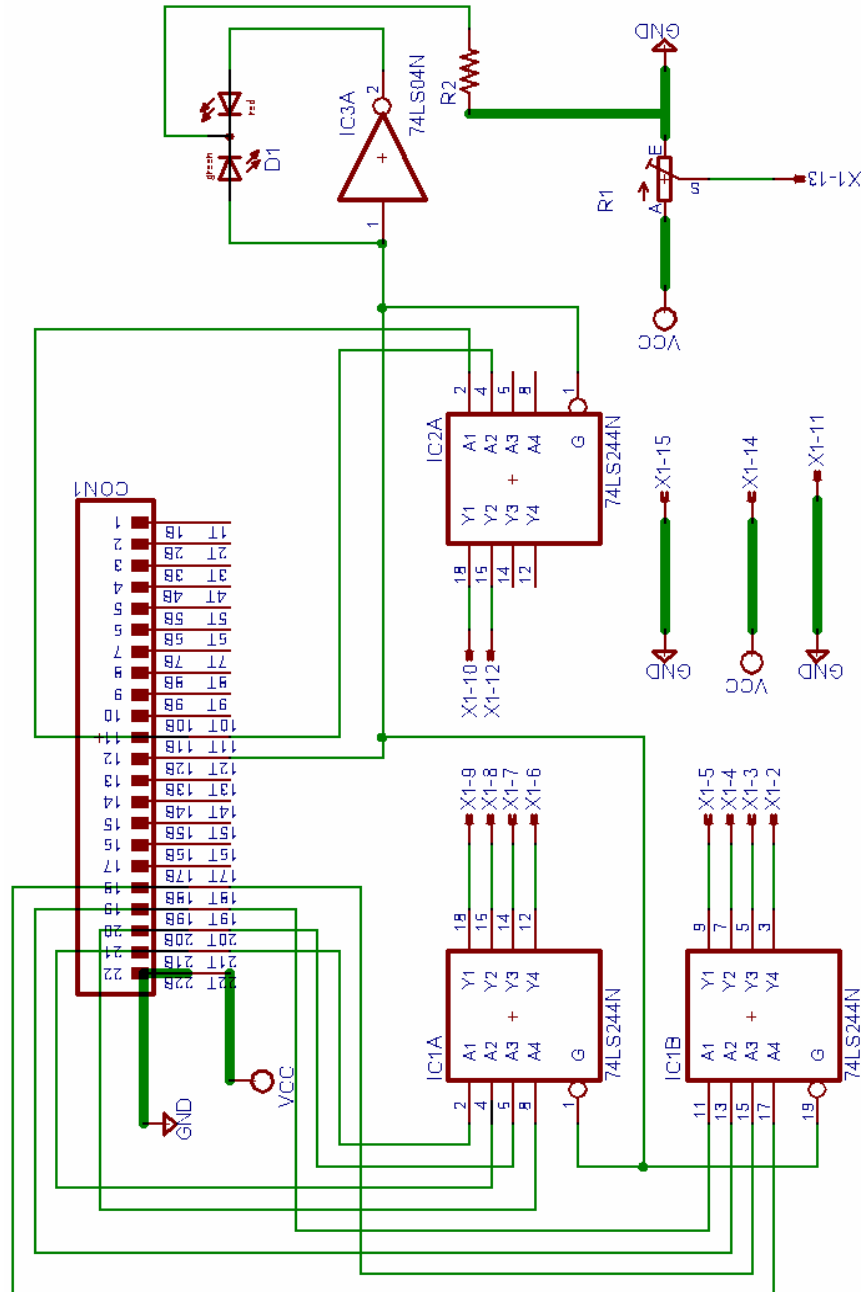


Figura 2.39 Diagrama esquemático de la tarjeta LCD

Diagrama board

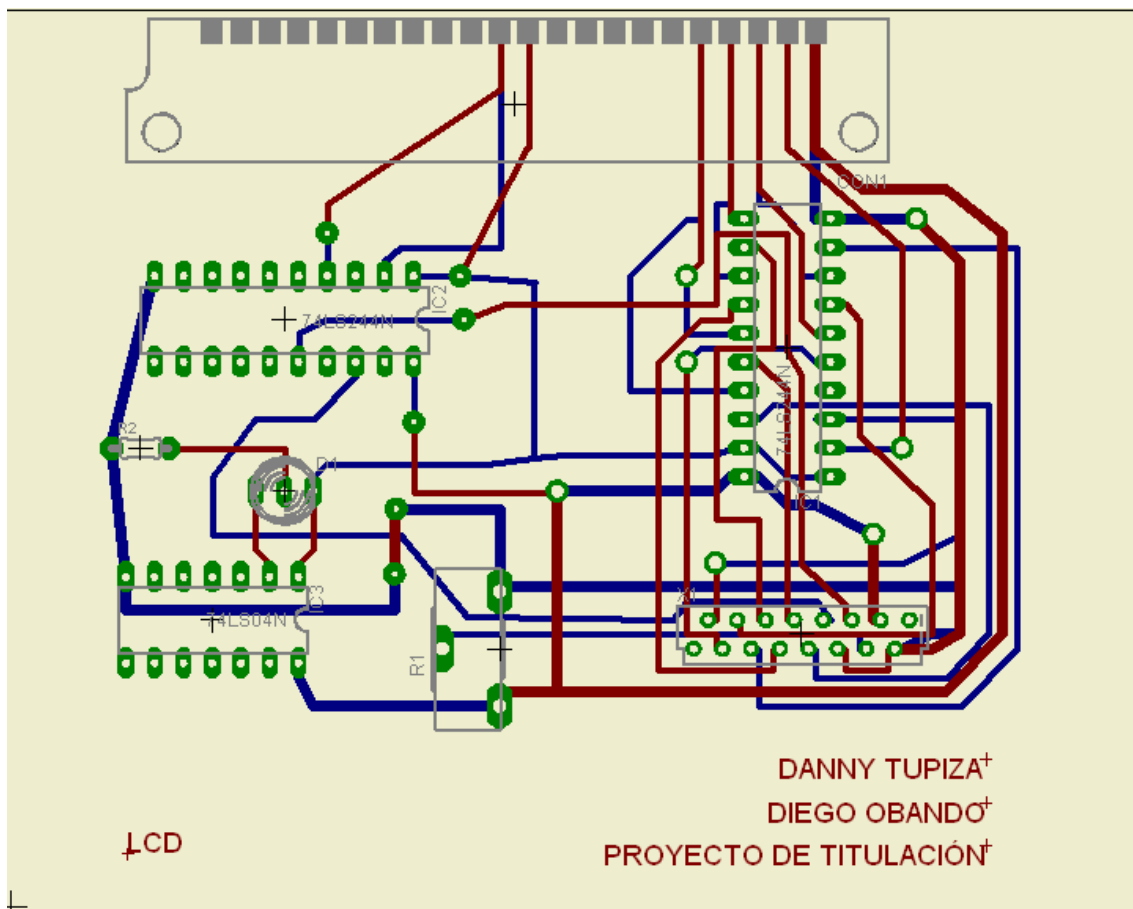


Figura 2.40 Diagrama board de la tarjeta LCD

2.1.4.5.- Tarjeta display de siete segmentos

Diagrama esquemático

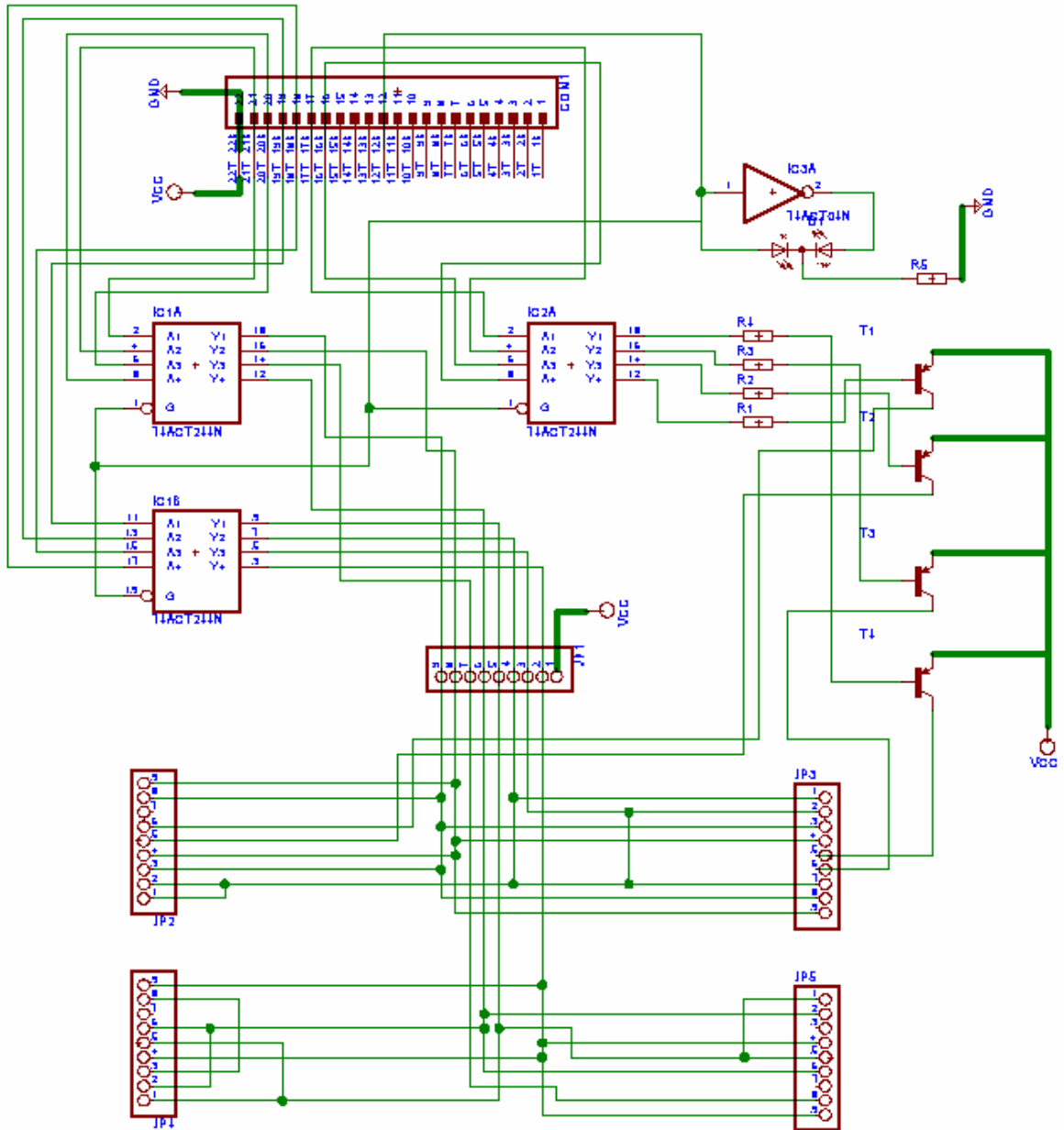


Figura 2.41 Diagrama esquemático de la tarjeta display de siete segmentos

Diagrama board

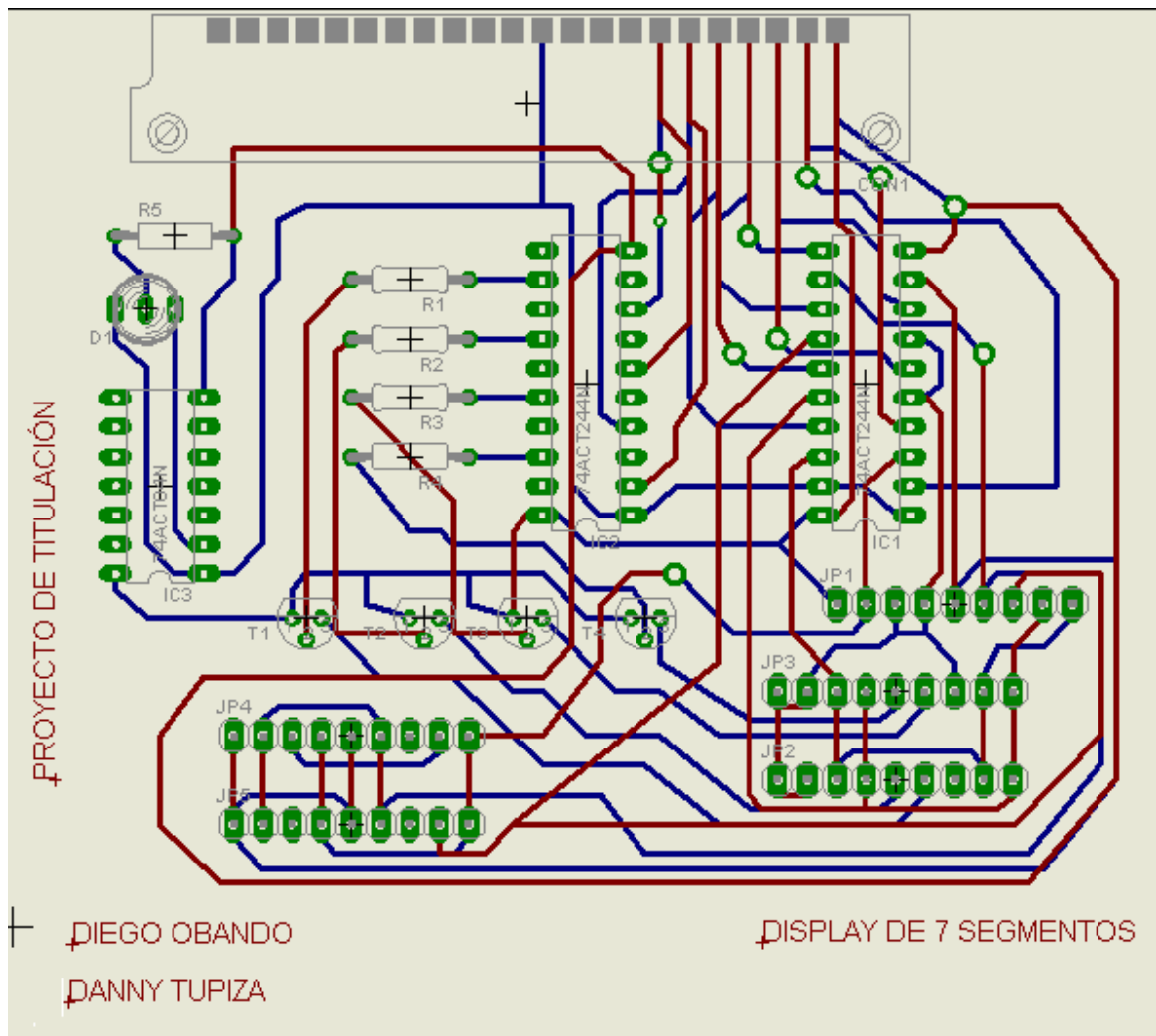


Figura 2.42 Diagrama board de la tarjeta display de siete segmentos

2.1.4.6.- Tarjeta MAX 232

Diagrama esquemático

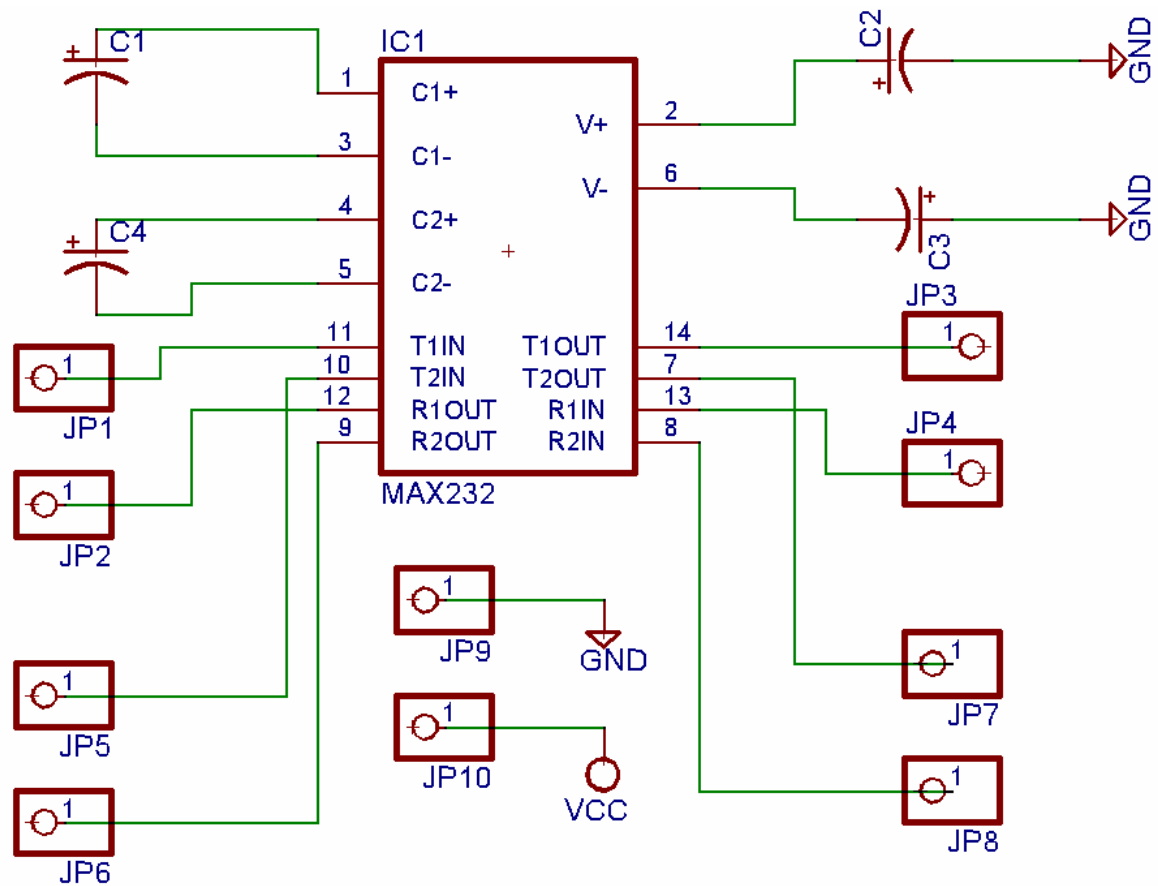


Figura 2.43 Diagrama esquemático de la tarjeta MAX 232

Diagrama board

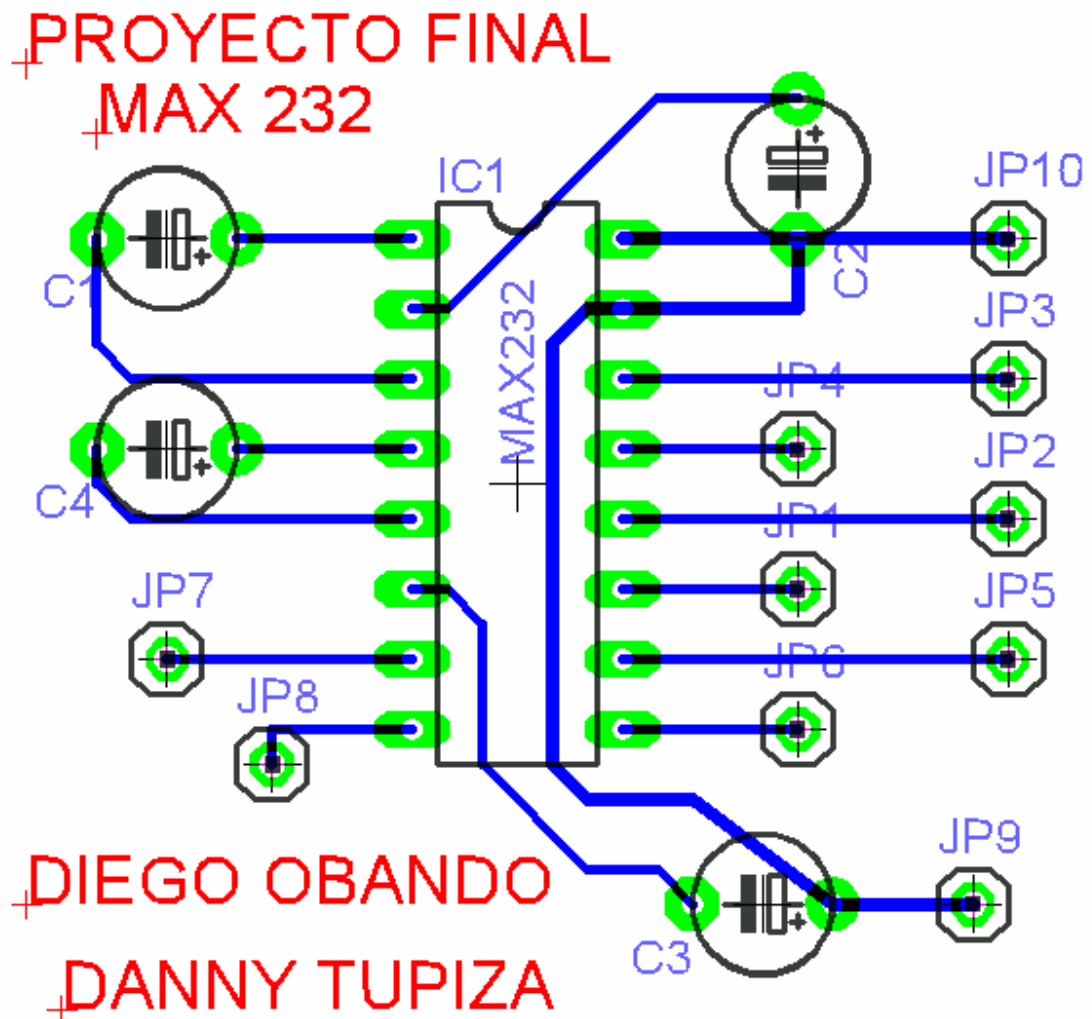
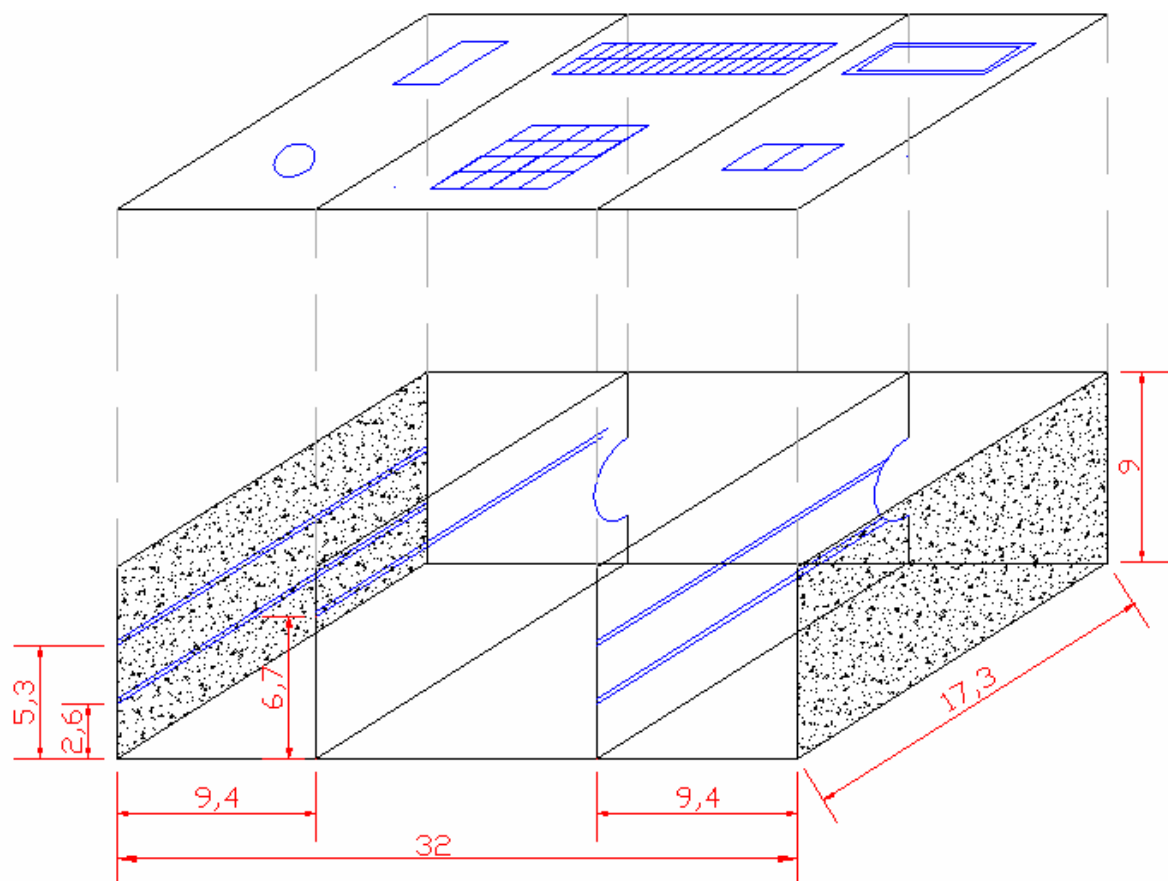


Figura 2.44 Diagrama board de la tarjeta MAX 232

2.1.4.7.- Diagrama del módulo

Se lo puede ver en Anexo 5.

2.1.5.- GRÁFICA DEL CHASIS



CHASIS

Figura 2.45

2.2.- DESARROLLO DEL SOFTWARE EN VISUAL BASIC PARA CONTROL DE FUNCIONAMIENTO DEL MOTOR PASO A PASO POR MEDIO DE PROTOCOLO RS – 232

Para la transmisión de datos a través del puerto serial al módulo, se realizó el siguiente programa en visual Basic:

```
Private Sub Command1_Click()  
MSComm1.Output = "I"  
End Sub
```

```
Private Sub Command2_Click()  
MSComm1.Output = "D"  
End Sub
```

```
Private Sub Command3_Click()  
Unload Me  
End Sub
```

```
Private Sub Form_Load()  
MSComm1.PortOpen = True  
End Sub
```

```
Private Sub Picture1_Click()  
Unload Me  
End Sub
```

```
Private Sub Picture2_Click()  
MSComm1.Output = "D"  
End Sub
```

```
Private Sub Picture3_Click()  
MSComm1.Output = "I"  
End Sub
```

En la figura 2.46 se muestra la presentación al iniciar el programa para la transmisión de datos desde el computador.



Figura 2. 46 Presentación del programa

Al dar un clic en el botón "Girar a la izquierda" el motor gira a la izquierda y al dar un clic en el botón "Girar a la derecha el" motor gira a la derecha como se muestra en la figura 2.47.



Figura 2.47 Programa para la comunicación serial

2.3.- DESARROLLO DE PROGRAMAS PEQUEÑOS PARA PRUEBAS EN EL MÓDULO

2.3.1.- PROGRAMA PARA CONTROL DEL MOTOR PASO A PASO

```
ORG      00H
LJMP     INICIO
ORG      30H
INICIO:
MOV      P0,#00H

MOV      P0,#05H
LCALL   TIEMPO
MOV      P0,#09H
LCALL   TIEMPO
MOV      P0,#0AH
LCALL   TIEMPO
MOV      P0,#06H
LCALL   TIEMPO

MOV      P0,#05H
LCALL   TIEMPO
MOV      P0,#09H
LCALL   TIEMPO
MOV      P0,#0AH
LCALL   TIEMPO
MOV      P0,#06H
LCALL   TIEMPO

MOV      P0,#05H
```

```
LCALL TIEMPO
MOV     P0,#09H
LCALL TIEMPO
MOV     P0,#0AH
LCALL TIEMPO
MOV     P0,#06H
LCALL TIEMPO
```

'Cambio de giro'

```
MOV     P0,#00H
```

```
MOV     P0,#06H
LCALL TIEMPO
MOV     P0,#0AH
LCALL TIEMPO
MOV     P0,#09H
LCALL TIEMPO
MOV     P0,#05H
LCALL TIEMPO
```

```
MOV     P0,#06H
LCALL TIEMPO
MOV     P0,#0AH
LCALL TIEMPO
MOV     P0,#09H
LCALL TIEMPO
MOV     P0,#05H
LCALL TIEMPO
```

```
MOV     P0,#06H
LCALL TIEMPO
```

```

MOV      P0,#0AH
LCALL   TIEMPO
MOV      P0,#09H
LCALL   TIEMPO
MOV      P0,#05H
LCALL   TIEMPO
SJMP    $

```

```

;          "SUBROUTINA DE TIEMPO"

```

```

TIEMPO:  MOV   R1,#98
          MOV   R2,#168
REPI:    DJNZ  R2,REPI
          DJNZ  R1,REPI
          RET

```

2.3.2.- PROGRAMA PARA CONTROL DE LA MATRIZ DE LED'S.

```

          ORG   00H
          LJMP  INICIO
          ORG   30H

INICIO:   MOV   R1,#0FAH

SIGA:    MOV   P0,#00H
          SETB  P2.0
          LCALL TIEMPO
          CLR   P2.0

          MOV   P0,#66H
          SETB  P2.1

```

```
LCALL    TIEMPO
CLR      P2.1

MOV      P0,#66H
SETB    P2.2
LCALL    TIEMPO
CLR      P2.2

MOV      P0,#66H
SETB    P2.3
LCALL    TIEMPO
CLR      P2.3

MOV      P0,#66H
SETB    P2.4
LCALL    TIEMPO
CLR      P2.4

DJNZ     R1, SIGA
MOV      R2,#64H

SIGA1:   MOV      P0,#00H
SETB    P2.0
LCALL    TIEMPO
CLR      P2.0

MOV      P0,#76H
SETB    P2.1
LCALL    TIEMPO
CLR      P2.1
```

```
MOV      P0,#76H
SETB     P2.2
LCALL    TIEMPO
CLR      P2.2

MOV      P0,#76H
SETB     P2.3
LCALL    TIEMPO
CLR      P2.3

MOV      P0,#79H
SETB     P2.4
LCALL    TIEMPO
CLR      P2.4

DJNZ     R2, SIGA1
MOV      R3,#64H

SIGA2:   MOV      P0,#00H
SETB     P2.0
LCALL    TIEMPO
CLR      P2.0

MOV      P0,#7EH
SETB     P2.1
LCALL    TIEMPO
CLR      P2.1

MOV      P0,#7EH
SETB     P2.2
LCALL    TIEMPO
```

```

CLR      P2.2

MOV      P0,#7EH
SETB     P2.3
LCALL    TIEMPO
CLR      P2.3

MOV      P0,#00H
SETB     P2.4
LCALL    TIEMPO
CLR      P2.4

DJNZ     R3, SIGA2
SJMP     $

```

```

;          "SUBROUTINA DE TIEMPO"

```

```

TIEMPO:  MOV      R4,#8
          MOV      R5,#208
REPI:    DJNZ     R5,REPI
          DJNZ     R4,REPI
          RET

```

2.3.3.- PROGRAMA PARA CONTROL DEL LCD.

```
$crystal = 12000000
```

```
Config Lcd = 16 * 2
```

```
Home U
```

```
Lcd "PROYECTO DE "
```

Home L

Lcd " Titulacion "

Wait 3

Home U

Lcd "DIEGO OBANDO "

Home L

Lcd "DANNY TUPIZA "

Wait 3

Home U

Lcd " E. P. N "

Home L

Lcd " ESFOT "

Wait 2

Do

Display On

Wait 1

Display Off

Wait 1

Loop

2.3.4.- PROGRAMA PARA CONTROL DEL DISPLAY DE SIETE SEGMENTOS

```

                ORG      00H
                LJMP     INICIO
                ORG      30H

INICIO:        MOV      P2,#0FFH

                MOV      R1,#0FAH
SIGA:          MOV      P0,#8CH      ;'P'
                CLR      P2.0
                LCALL    TIEMPO
                SETB     P2.0
                MOV      P0,#0AFH    ;'r'
                CLR      P2.1
                LCALL    TIEMPO
                SETB     P2.1
                MOV      P0,#0A3H    ;'o'
                CLR      P2.2
                LCALL    TIEMPO
                SETB     P2.2
                MOV      P0,#99H     ;'Y'
                CLR      P2.3
                LCALL    TIEMPO
                SETB     P2.3
                DJNZ     R1,SIGA

                MOV      R1,#0FFH
SIGA1:        MOV      P0,#87H      ;'t'
                CLR      P2.0
                LCALL    TIEMPO

```

```

        SETB    P2.0
        MOV     P0,#0F9H      ; 'I'
        CLR     P2.1
        LCALL   TIEMPO
        SETB    P2.1
        MOV     P0,#87H      ; 't'
        CLR     P2.2
        LCALL   TIEMPO
        SETB    P2.2
        MOV     P0,#0C1H     ; 'U'
        CLR     P2.3
        LCALL   TIEMPO
        SETB    P2.3
        DJNZ    R1,SIGA1

        MOV     R1,#0FFH
SIGA2:  MOV     P0,#86H      ; 'E'
        CLR     P2.0
        LCALL   TIEMPO
        SETB    P2.0
        MOV     P0,#8CH      ; 'P'
        CLR     P2.1
        LCALL   TIEMPO
        SETB    P2.1
        MOV     P0,#0ABH     ; 'n'
        CLR     P2.2
        LCALL   TIEMPO
        SETB    P2.2
        DJNZ    R1,SIGA2

SJMP    $

```

; "SUBROUTINA DE TIEMPO"

```

TIEMPO:      MOV      R4,#8
              MOV      R5,#208
REPI:        DJNZ     R5,REPI
              DJNZ     R4,REPI
              RET

```

2.3.5.- PROGRAMA PARA CONTROL DEL TECLADO

```

              ORG      0000H
              LJMP     INICIO
              ORG      0030H

EXTI0:       LJMP     CAMBIO
              ORG      0030H

INICIO:      MOV      SP,#2FH
              MOV      TCON,#01H
              MOV      IE,#81H
              SJMP     $

CAMBIO:      MOV      A,#00H
              MOV      A,P1

              CJNE     A,#0F0H, SALTE1
              MOV      R1,#0FAH

SIGA:        MOV      P0,#0F9H      ;1'
              CLR      P2.0
              LCALL   TIEMPO
              SETB    P2.0

```

```

        DJNZ     R1,SIGA
        LCALL    TIEMPO
        LCALL    TIEMPO

SALTE1:  CJNE    A,#0F1H,SALTE2
        MOV     R1,#0FFH
        SIGA1:  MOV     P0,#0A4H      ; '2'
        CLR     P2.1
        LCALL    TIEMPO
        SETB    P2.1
        DJNZ    R1,SIGA1

SALTE2:  CJNE    A,#0F2H,SALTE3

        MOV     R1,#0FFH
        SIGA2:  MOV     P0,#0B0H      ; '3'
        CLR     P2.2
        LCALL    TIEMPO
        SETB    P2.2
        DJNZ    R1,SIGA2

SALTE3:  CJNE    A,#0F3H,SALTE4
        MOV     R1,#0FFH
        SIGA3:  MOV     P0,#088H      ; 'A'
        CLR     P2.3
        LCALL    TIEMPO
        SETB    P2.3
        DJNZ    R1,SIGA3

SALTE4:  CJNE    A,#0F4H,SALTE5

```

```

        MOV        R1,#0FFH
SIGA4:  MOV        P0,#099H        ; '4'
        CLR        P2.0
        LCALL     TIEMPO
        SETB       P2.0
        DJNZ      R1,SIGA4

SALTE5:  CJNE     A,#0F5H,SALTE6
        MOV        R1,#0FFH
SIGA5:  MOV        P0,#092H        ; '5'
        CLR        P2.1
        LCALL     TIEMPO
        SETB       P2.1
        DJNZ      R1,SIGA5

SALTE6:  CJNE     A,#0F6H,SALTE7
        MOV        R1,#0FFH
SIGA6:  MOV        P0,#082H        ; '6'
        CLR        P2.2
        LCALL     TIEMPO
        SETB       P2.2
        DJNZ      R1,SIGA6

SALTE7:  CJNE     A,#0F7H,SALTE8
        MOV        R1,#0FFH
SIGA7:  MOV        P0,#083H        ; 'B'
        CLR        P2.3
        LCALL     TIEMPO
        SETB       P2.3
        DJNZ      R1,SIGA7
```

```
SALTE8:      CJNE      A,#0F8H,SALTE9
              MOV       R1,#0FFH
              SIGA8:    MOV       P0,#0B8H      ; '7'
              CLR       P2.0
              LCALL     TIEMPO
              SETB      P2.0
              DJNZ      R1,SIGA8

SALTE9:      CJNE      A,#0F9H,SALTE10
              MOV       R1,#0FFH
              SIGA9:    MOV       P0,#080H      ; '8'
              CLR       P2.1
              LCALL     TIEMPO
              SETB      P2.1
              DJNZ      R1,SIGA9

SALTE10:     CJNE      A,#0FAH,SALTE11
              MOV       R1,#0FFH
              SIGA10:   MOV       P0,#098H      ; '9'
              CLR       P2.2
              LCALL     TIEMPO
              SETB      P2.2
              DJNZ      R1,SIGA10

SALTE11:     CJNE      A,#0FBH,SALTE12
              MOV       R1,#0FFH
              SIGA11:   MOV       P0,#0C6H      ; 'C'
              CLR       P2.3
              LCALL     TIEMPO
              SETB      P2.3
              DJNZ      R1,SIGA11
```

```
SALTE12:    CJNE    A,#0FCH,SALTE13
            MOV     R1,#0FFH
            SIGA12: MOV     P0,#0D2H      ; '*'
            CLR     P2.0
            LCALL   TIEMPO
            SETB    P2.0
            DJNZ    R1,SIGA12

SALTE13:    CJNE    A,#0FDH,SALTE14
            MOV     R1,#0FFH
            SIGA13: MOV     P0,#0C0H      ; '0'
            CLR     P2.1
            LCALL   TIEMPO
            SETB    P2.1
            DJNZ    R1,SIGA13

SALTE14:    CJNE    A,#0FEH,SALTE15
            MOV     R1,#0FFH
            SIGA14: MOV     P0,#089H      ; '#'
            CLR     P2.2
            LCALL   TIEMPO
            SETB    P2.2
            DJNZ    R1,SIGA14

SALTE15:    CJNE    A,#0FFH,SALTE16
            MOV     R1,#0FFH
            SIGA15: MOV     P0,#0A1H      ; 'D'
            CLR     P2.3
            LCALL   TIEMPO
            SETB    P2.3
            DJNZ    R1,SIGA15
```

```

                LCALL    TIEMPO

SALTE16:       MOV      R1,#0FAH

SIGA17:        MOV      P0,#86H      ;E'
                CLR      P2.0
                LCALL    TIEMPO
                SETB     P2.0
                MOV      P0,#0ABH    ;'n'
                CLR      P2.1
                LCALL    TIEMPO
                SETB     P2.1
                MOV      P0,#0A1H    ;'d'
                CLR      P2.2
                LCALL    TIEMPO
                SETB     P2.2
                DJNZ     R1,SIGA17

LJMP  INICIO

                ;"SUBROUTINA DE TIEMPO"

TIEMPO:        MOV      R4,#8
                MOV      R5,#208

REPI:          DJNZ     R5,REPI
                DJNZ     R4,REPI
                RET

END

```


2.3.6.- PROGRAMA PARA LA TRANSMISIÓN SERIAL

```
Dim Serial_in As Byte
```

```
Dim Giro As Byte
```

```
Do
```

```
Serial_in = Inkey()
```

```
If Serial_in = "D" Then Giro = 1
```

```
If Serial_in = "I" Then Giro = 2
```

```
If Serial_in = "P" Then Giro = 3
```

```
    If Giro = 1 Then
```

```
        P0 = 5
```

```
        Waitms 200
```

```
        P0 = 9
```

```
        Waitms 200
```

```
        P0 = 10
```

```
        Waitms 200
```

```
        P0 = 6
```

```
        Waitms 200
```

```
    End If
```

```
    If Giro = 2 Then
```

```
        P0 = 6
```

```
        Waitms 200
```

```
        P0 = 10
```

```
        Waitms 200
```

```
P0 = 9  
Waitms 200  
P0 = 5  
Waitms 200  
End If
```

```
If Giro = 3 Then  
    P0 = 0  
End If  
Loop
```

CAPITULO 3

PRUEBAS Y CALIBRACIONES

3.1 PRUEBA Y CALIBRACIONES DEL NUEVO MÓDULO PARA MANEJO DE MICROCONTROLADORES AT89C51.

3.1.1 PRUEBA Y CALIBRACIÓN DE LA TARJETA MADRE

Primero se verificó que no existan presentes cortocircuitos en esta tarjeta, al igual que entre dicha tarjeta con el bus de datos y con las demás tarjetas.

Lo que se verificó en esta tarjeta fue el funcionamiento de los cuatro Flip-Flops que activan a las cuatro tarjetas del módulo independientemente. Además se verificó que cada uno de los cuatro switches de control, de cada tarjeta del módulo, controla a una sola a la vez, y esto se comprobó cuando el led de dos colores pasa a color verde en la tarjeta activada.

La calibración que se realizó en esta tarjeta, tuvo que ver con sueldas frías que se encontraron en la misma.

3.1.2 PRUEBA Y CALIBRACIÓN DE LA TARJETA MOTOR PASO A PASO

Primero se verificó que no existan cortocircuitos en esta tarjeta, en el bus de datos y con las demás tarjetas.

En esta tarjeta se verificó la programación desarrollada en el AT89C51, que consistía en hacer girar el motor paso a paso en sentido horario.

También se verificó las pruebas desde la PC a través del programa diseñado para controlar al motor paso a paso que se encuentra en el módulo.

El trabajo que se realizó en esta tarjeta y en el motor paso a paso bipolar, tuvo que ver con la identificación de las bobinas A, B, C, D respectivamente.

3.1.3 PRUEBA Y CALIBRACIÓN DE LA TARJETA MATRIZ DE LED'S

Primero se verificó que no estén presentes cortocircuitos en esta tarjeta, en el bus de datos y con las demás tarjetas.

En esta tarjeta se verificó la programación desarrollada en el AT89C51, dicho programa consistía en observar en la matriz de led's la sigla: "EPN".

3.1.4 PRUEBA Y CALIBRACIÓN DE LA TARJETA LCD

Primero se verificó que no estén presentes cortocircuitos en esta tarjeta, en el bus de datos y con las demás tarjetas.

En esta tarjeta se verificó la programación desarrollada en el AT89C51, que consistía en observar las siglas:

"PROYECTO DE "
" Titulacion "

"DIEGO OBANDO "
"DANNY TUPIZA "

" E. P. N "

" ESFOT "

El trabajo que se realizó en esta tarjeta, tuvo que ver con la identificación de soldas frías que se encontraron en la placa y conectar al pin 4 del C.I. 74244 (#1) con el pin 6 (P0.4) del bus de datos.

3.1.5 PRUEBA Y CALIBRACIÓN DE LA TARJETA DISPLAY 7 SEGMENTOS

Primero se verificó que no estén presentes cortocircuitos en esta tarjeta, en el bus de datos y con las demás tarjetas.

En esta tarjeta se verificó la programación realizada en el AT89C51, dicho programa consistía en observar en los dos display's dobles las siglas:

“Proy “

“Titu”.

“EPN”

CONCLUSIONES

Después de la elaboración de este proyecto se concluye que:

- Se construyó un módulo didáctico diseñado para probar la programación del microcontrolador AT89C51 aplicando cuatro dispositivos generales que son: motor paso a paso, matriz de led's, LCD y display's; y usando un circuito que permite la comunicación serial mediante el protocolo RS-232 entre la PC y el microcontrolador.
- Este módulo didáctico tiene la finalidad de ser usado en el Laboratorio de Microprocesadores de la ESFOT para la enseñanza práctica de la programación del microcontrolador AT89C51
- Debido al carácter didáctico que tiene este módulo, se puede mejorar la enseñanza de la programación del microcontrolador AT89C51, por ejemplo el programador/ra no solo tiene disponible un simple grupo de led's para manejar la programación del microcontrolador, si no que tiene la posibilidad de ser controlado desde una computadora.
- Con este Módulo Didáctico, el programador del AT89C51 tiene la facilidad de poner en práctica el programa del motor paso a paso programado por él, porque ya tiene listo todo el hardware para la comunicación serial, y por lo tanto no tiene que preocuparse en armar un circuito para verificar la programación realizada en el microcontrolador con el riesgo de que su circuito presente errores y fallas de diseño y ensamblaje.

RECOMENDACIONES

- Antes de usar el módulo didáctico es indispensable que se lea el Manual del Usuario.
- Con los circuitos integrados es muy importante tomar en cuenta las recomendaciones del fabricante, puesto que muchas veces, ya viene diseñado en los Data Sheets, el circuito que nos puede ser útil.
- Para evitar sueldas frías, en la soldadura de componentes electrónicos influye la calidad del cautín, la suelda y que las pistas de la baquelita estén limpias.
- Para evitar que se introduzcan ruidos en el circuito oscilador que usa el microcontrolador AT89C51, este oscilador tiene que ubicarse lo más cerca del microcontrolador.
- Se recomienda no solo crear módulos para el laboratorio de microprocesadores, si no también para el resto de las áreas académicas de la carrera de Electrónica y Telecomunicaciones, para ayudar a mejorar el aprendizaje de los estudiantes.

BIBLIOGRAFÍA

DIRECCIONES ELECTRÓNICAS

DIODO LED.

<http://es.wikipedia.org/w/index.php>

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

http://es.wikipedia.org/wiki/Diodo_LED

MOTORES PASO A PASO

<http://www.micropik.com/motores.htm>

<http://alek.pucp.edu.pe/cursos/pregrado/iee215/pasos/>

<http://www.todorobot.com.ar/informacion/tutorial%20stepper/stepper-tutorial.htm>

COMUNICACIÓN SERIAL.

http://www.geocities.com/alva_cesar/rs232/hp_pic.html

http://www.camiresearch.com/Data_Com_Basics/New_Cable_pdfs/1.pdf

<http://ceres.ugr.es/~alumnos/redrs232/fisica.htm>

<http://www.euskalnet.net/shizuka/rs232.htm>

<http://www.kmitl.ac.th/~kswichit/MAX232/MAX232.htm>

LCD

<http://www.x-robotics.com/rutinas.htm#LCD>

<http://www.terra.es/personal/fremiro/Archivos/Lcd.pdf>

<http://www.beyondlogic.org/lcd/lcd.pdf>

MICROCONTROLADOR AT89C51

http://www.atmel.com/dyn/resources/prod_documents/doc0265.pdf

<http://www.iguanalabs.com/8951pin.htm>

<http://www.olimex.cl/tutorial1.pdf?osCsid=29c885451c66bd348f72d2a1dbe656c7>

http://www.dea.icaei.upco.es/daniel/asignaturas/SisEleDig1IAEI/UG_prince_2.1.pdf

http://www.datsi.fi.upm.es/docencia/Micro_C/atmel/doc0497.pdf

<http://www.amcomputersystems.com/AM//robots/archivos/robmoii/detodos/89C51.pdf>

<http://integrados.8m.com/89c51.pdf>

TECLADO HEXADECIMAL

<http://perso.wanadoo.es/chyryes/circuitos/teclado.htm>

http://www.uv.es/rosado/dcse/Interface_teclado_hexa.PDF

LIBROS:

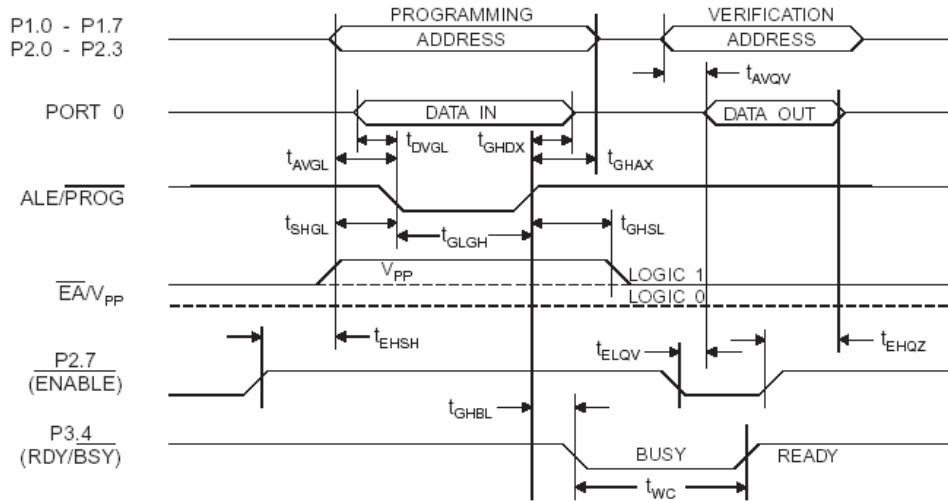
Instituto Tecnológico Superior Central Técnico, (1999) Folleto Electrónica Industrial Quito.

TOCCI RONALD, J (2001) "Sistemas Digitales Principios y Aplicaciones"

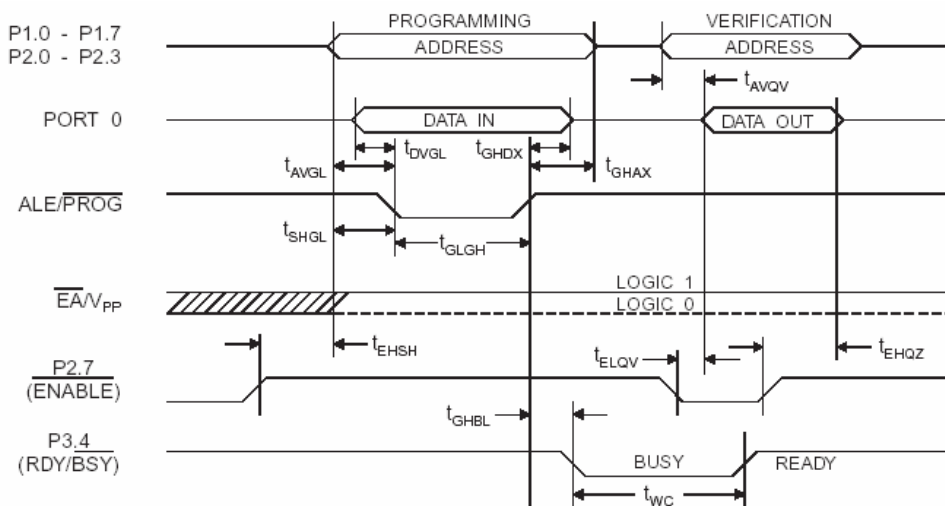
ANEXOS

ANEXO 1
EI MICROCONTROLADOR AT89C51

Programación y verificación de formas de ondas en modo Alto Voltaje (Vpp = 12V)



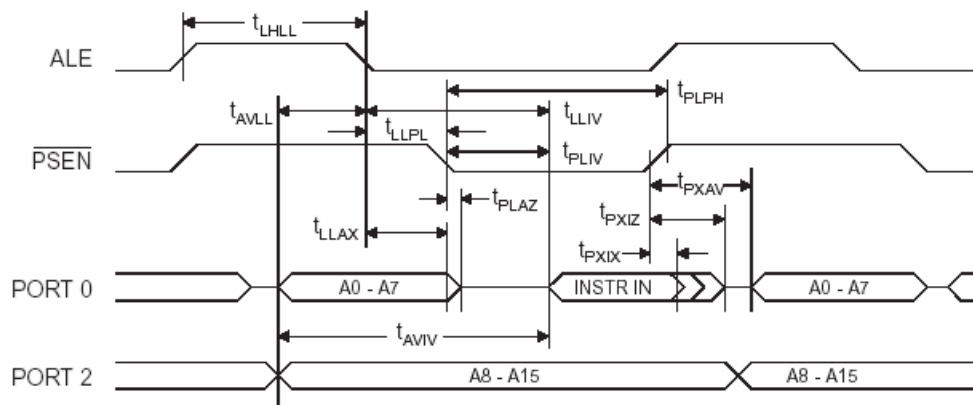
Programación y verificación de formas de ondas en modo Bajo Voltaje (Vpp = 5V)



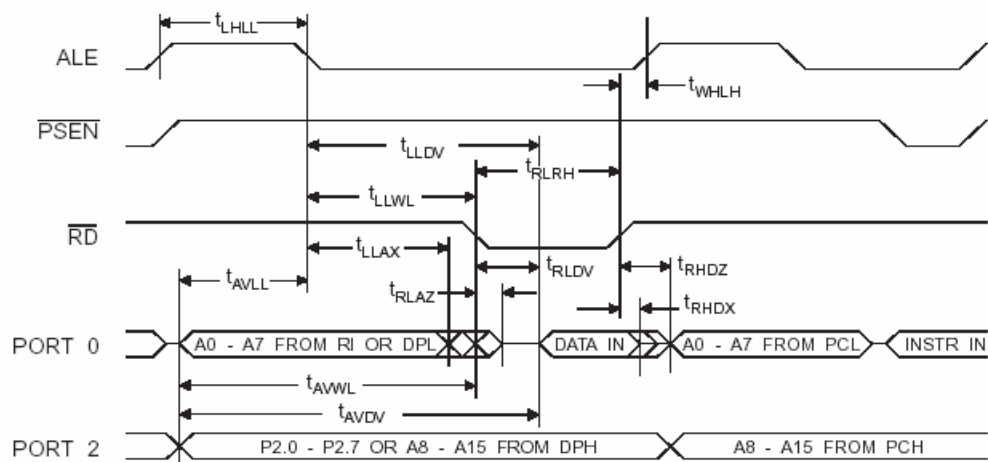
Características de programa externo y datos de memoria.

Symbol	Parameter	12 MHz Oscillator		16 to 24 MHz Oscillator		Units
		Min	Max	Min	Max	
$1/t_{CLCL}$	Oscillator Frequency			0	24	MHz
t_{LHLL}	ALE Pulse Width	127		$2t_{CLCL}-40$		ns
t_{AVLL}	Address Valid to ALE Low	43		$t_{CLCL}-13$		ns
t_{LLAX}	Address Hold after ALE Low	48		$t_{CLCL}-20$		ns
t_{LLIV}	ALE Low to Valid Instruction In		233		$4t_{CLCL}-65$	ns
t_{LLPL}	ALE Low to \overline{PSEN} Low	43		$t_{CLCL}-13$		ns
t_{PLPH}	\overline{PSEN} Pulse Width	205		$3t_{CLCL}-20$		ns
t_{PLIV}	\overline{PSEN} Low to Valid Instruction In		145		$3t_{CLCL}-45$	ns
t_{PXIX}	Input Instruction Hold after \overline{PSEN}	0		0		ns
t_{PXIZ}	Input Instruction Float after \overline{PSEN}		59		$t_{CLCL}-10$	ns
t_{PXAV}	\overline{PSEN} to Address Valid	75		$t_{CLCL}-8$		ns
t_{AVIV}	Address to Valid Instruction In		312		$5t_{CLCL}-55$	ns
t_{PLAZ}	\overline{PSEN} Low to Address Float		10		10	ns
t_{RLRH}	\overline{RD} Pulse Width	400		$6t_{CLCL}-100$		ns
t_{WLWH}	\overline{WR} Pulse Width	400		$6t_{CLCL}-100$		ns
t_{RLDV}	\overline{RD} Low to Valid Data In		252		$5t_{CLCL}-90$	ns
t_{RHDX}	Data Hold after \overline{RD}	0		0		ns
t_{RHDZ}	Data Float after \overline{RD}		97		$2t_{CLCL}-28$	ns
t_{LLDV}	ALE Low to Valid Data In		517		$8t_{CLCL}-150$	ns
t_{AVDV}	Address to Valid Data In		585		$9t_{CLCL}-165$	ns
t_{LLWL}	ALE Low to \overline{RD} or \overline{WR} Low	200	300	$3t_{CLCL}-50$	$3t_{CLCL}+50$	ns
t_{AVWL}	Address to \overline{RD} or \overline{WR} Low	203		$4t_{CLCL}-75$		ns
t_{QVWX}	Data Valid to \overline{WR} Transition	23		$t_{CLCL}-20$		ns
t_{QVWH}	Data Valid to \overline{WR} High	433		$7t_{CLCL}-120$		ns
t_{WHQX}	Data Hold after \overline{WR}	33		$t_{CLCL}-20$		ns
t_{RLAZ}	\overline{RD} Low to Address Float		0		0	ns
t_{WHLH}	\overline{RD} or \overline{WR} High to ALE High	43	123	$t_{CLCL}-20$	$t_{CLCL}+25$	ns

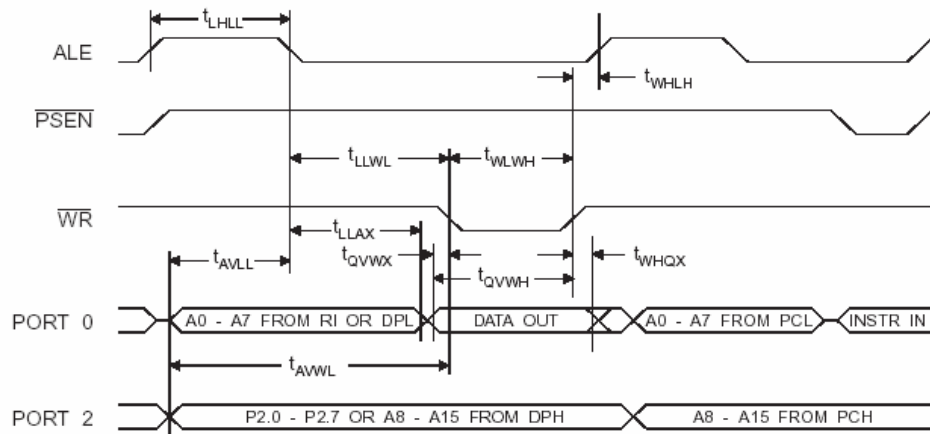
Ciclo de lectura de memoria de programa externo.



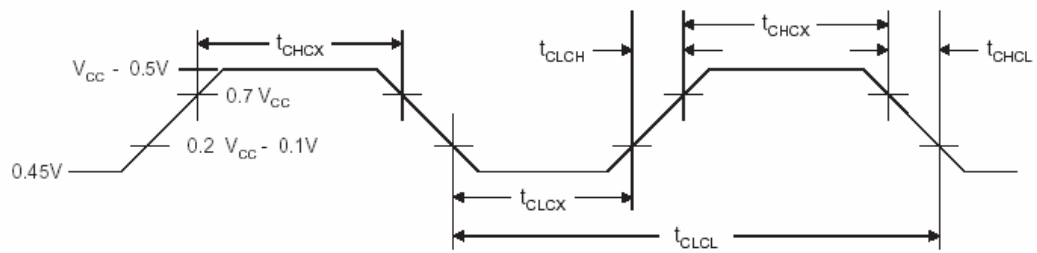
Ciclo de lectura de memoria de dato externo.



Dato de memoria externo del ciclo de escritura



Formas de onda del manejo del reloj externo



ANEXO 2
74C922

TABLAS DE VERDAD

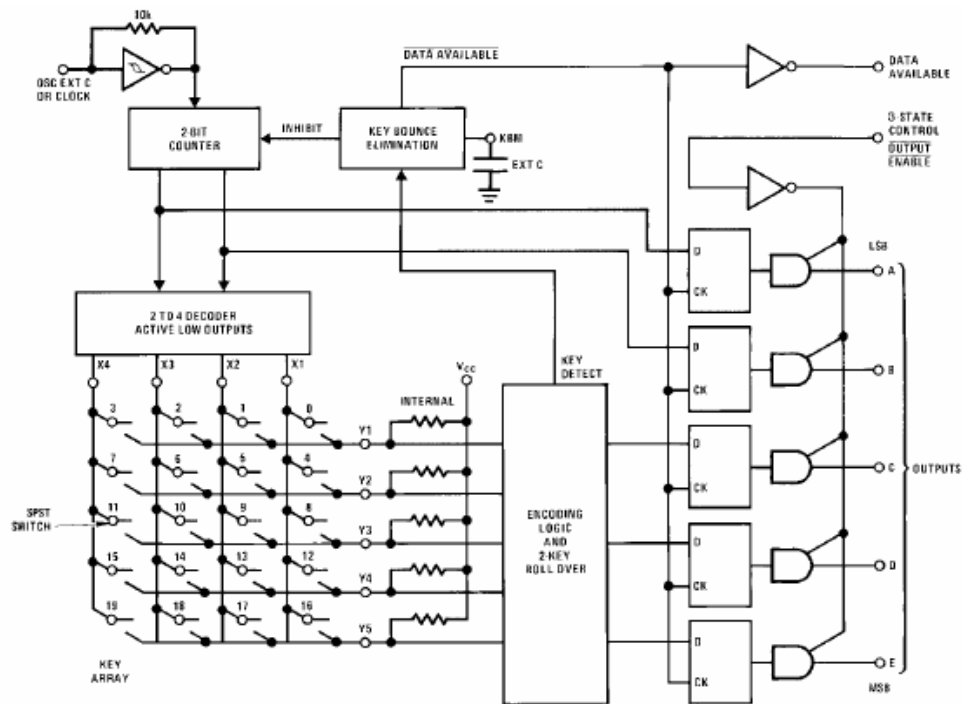
(Pins 0 through 11)

Switch Position	0	1	2	3	4	5	6	7	8	9	10	11
	Y1, X1	Y1, X2	Y1, X3	Y1, X4	Y2, X1	Y2, X2	Y2, X3	Y2, X4	Y3, X1	Y3, X2	Y3, X3	Y3, X4
D												
A A	0	1	0	1	0	1	0	1	0	1	0	1
T B	0	0	1	1	0	0	1	1	0	0	1	1
A C	0	0	0	0	1	1	1	1	0	0	0	0
O D	0	0	0	0	0	0	0	0	1	1	1	1
U E (Note 1)	0	0	0	0	0	0	0	0	0	0	0	0
T												

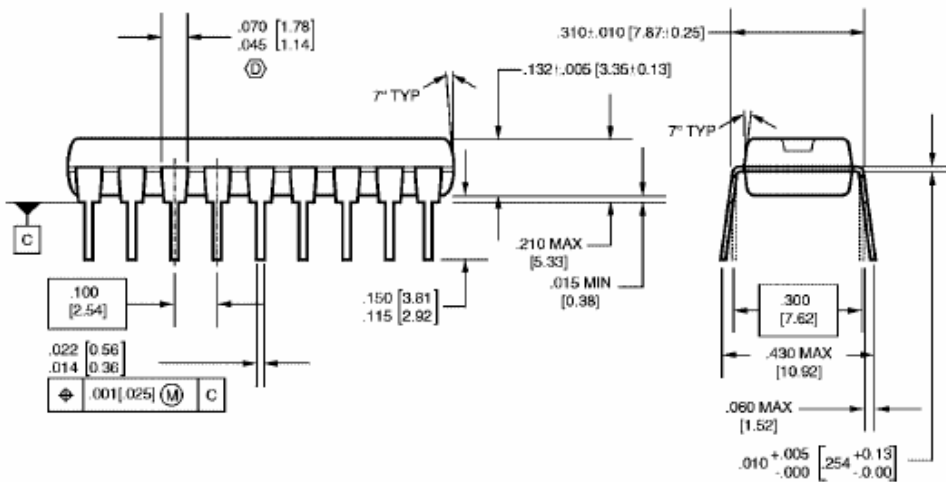
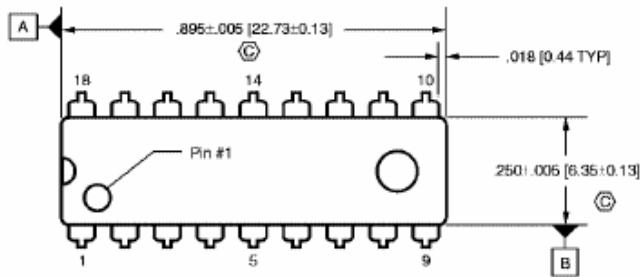
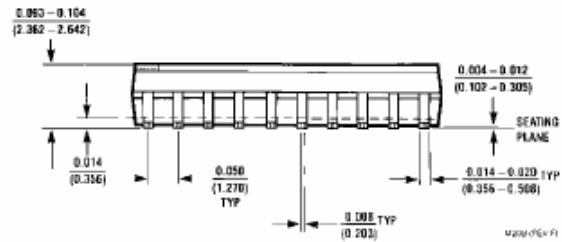
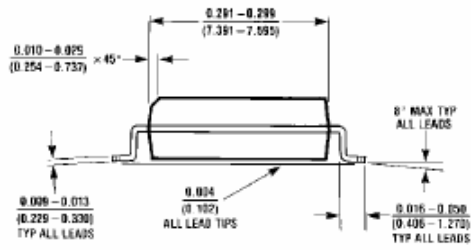
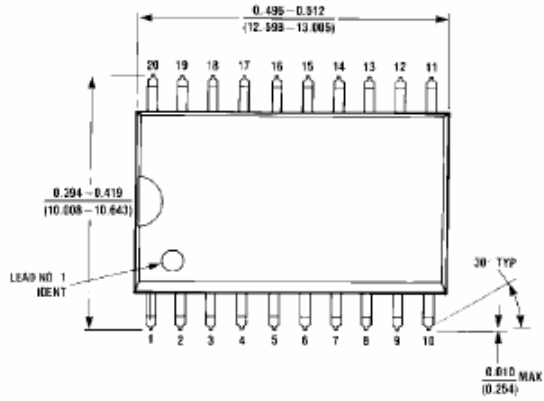
(Pins 12 through 19)

Switch Position	12	13	14	15	16	17	18	19
	Y4, X1	Y4, X2	Y4, X3	Y4, X4	Y5 (Note 1), X1	Y5 (Note 1), X2	Y5 (Note 1), X3	Y5 (Note 1), X4
D								
A A	0	1	0	1	0	1	0	1
T B	0	0	1	1	0	0	1	1
A C	1	1	1	1	0	0	0	0
O D	1	1	1	1	0	0	0	0
U E (Note 1)	0	0	0	0	1	1	1	1
T								

DIAGRAMA DE BLOQUES



DIMENSIONES



ANEXO 3
74LS11

DISTRIBUCIÓN DE PINES

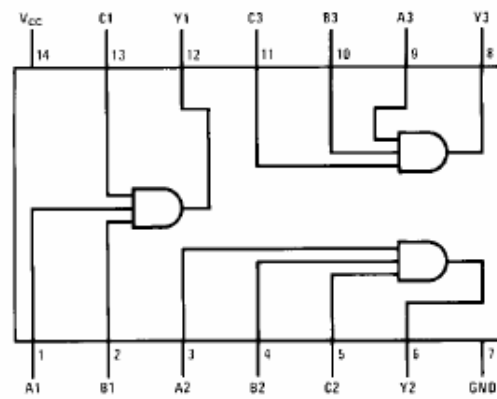


TABLA DE FUNCIÓN

$$Y = ABC$$

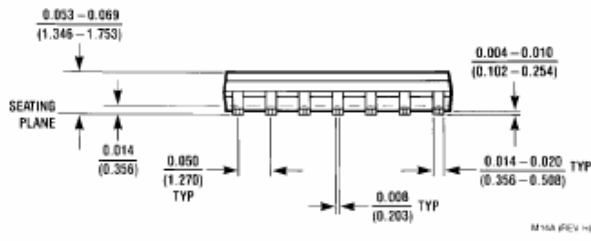
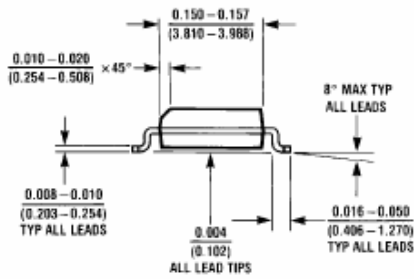
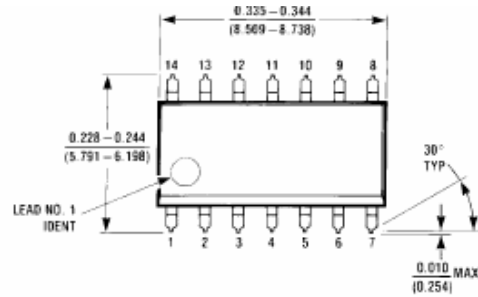
Inputs			Output
A	B	C	Y
X	X	L	L
X	L	X	L
L	X	X	L
H	H	H	H

H = HIGH Logic Level

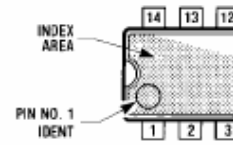
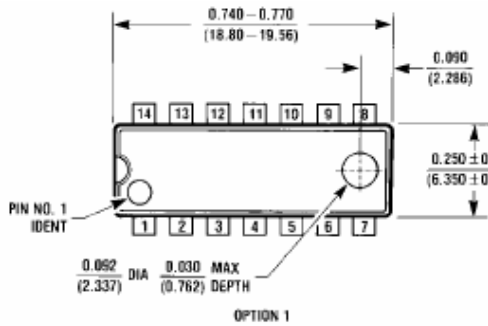
L = LOW Logic Level

X = Either LOW or HIGH Logic Level

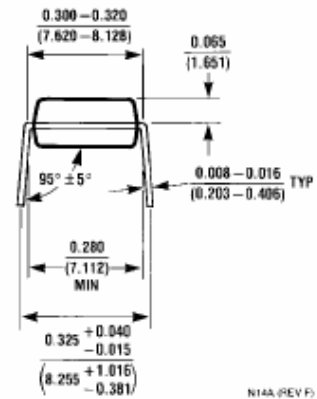
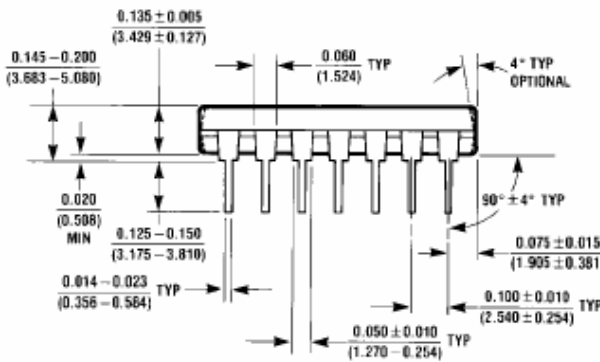
DIMENSIONES



14-Lead Small Outline Integrated Circuit (SOIC), JEDEC MS-120, 0.150 Narrow Package Number M14A



OPTION 2



ANEXO 4
L293D

DISTRIBUCIÓN DE PINES

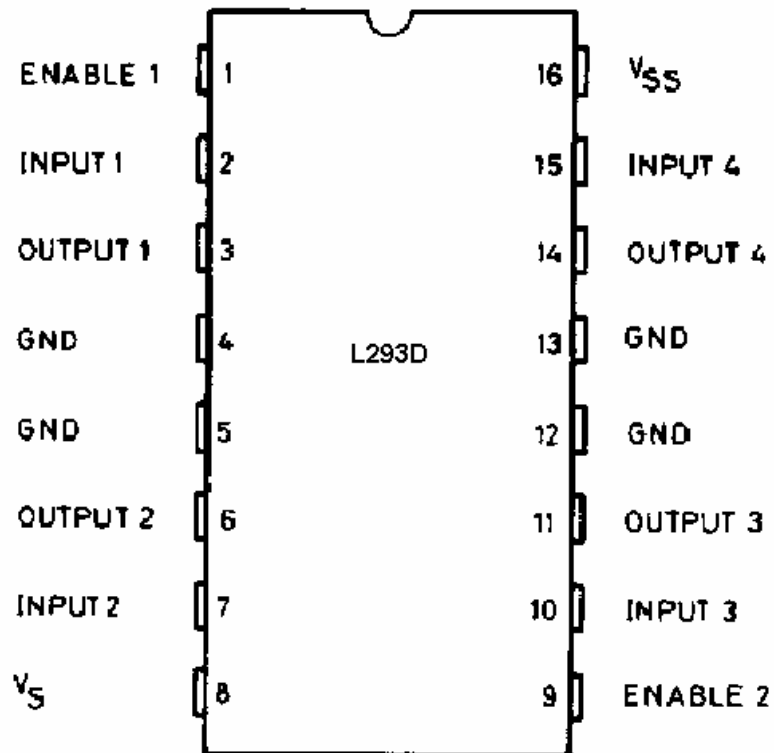
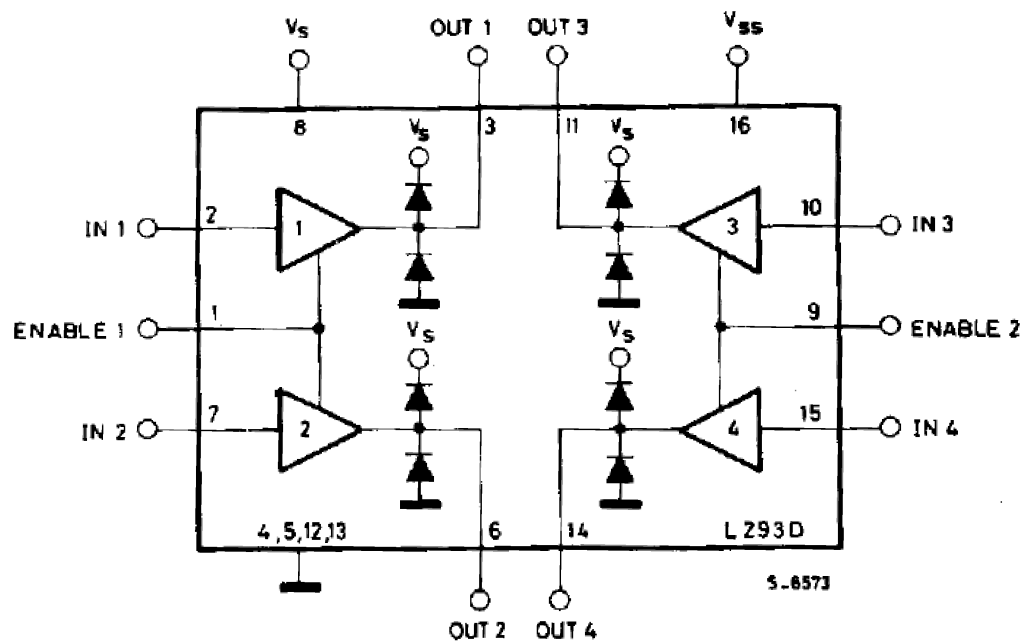
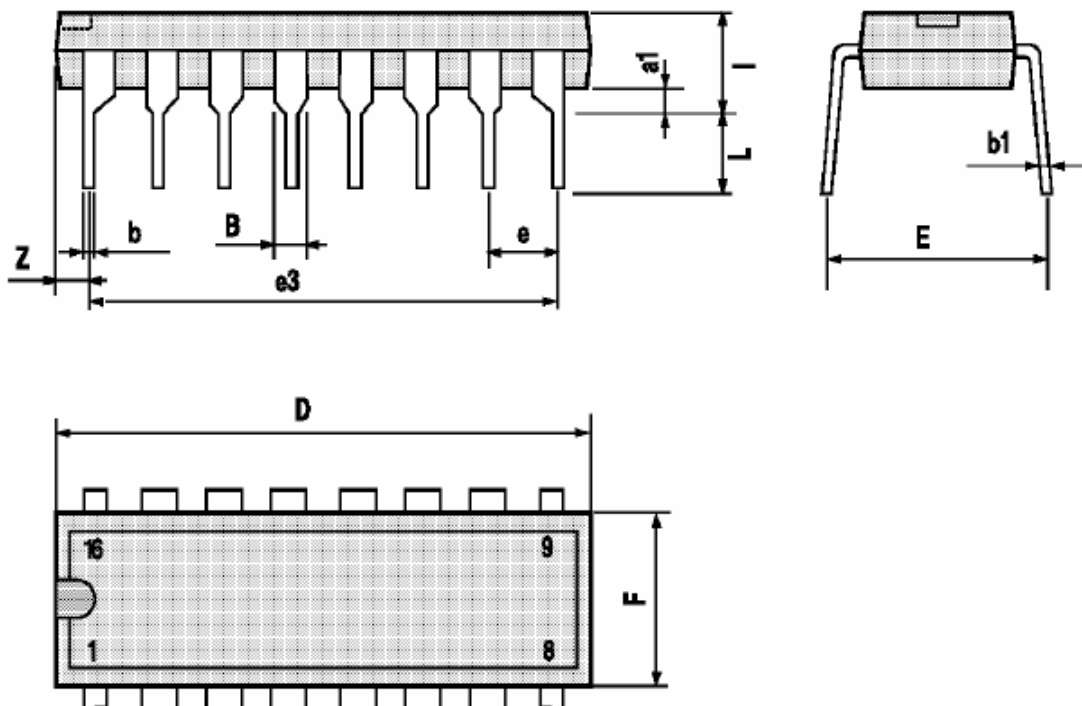


DIAGRAMA DE BLOQUES



DIMENSIONES

DIM.	mm			inch		
	MIN.	TYP.	MAX.	MIN.	TYP.	MAX.
a1	0.51			0.020		
B	0.85		1.40	0.033		0.055
b		0.50			0.020	
b1	0.38		0.50	0.015		0.020
D			20.0			0.787
E		8.80			0.346	
e		2.54			0.100	
e3		17.78			0.700	
F			7.10			0.280
I			5.10			0.201
L		3.30			0.130	
Z			1.27			0.050



ANEXO 5
DIAGRAMA DEL MÓDULO

ANEXO 6
MANUAL DE USUARIO

MANUAL DE USUARIO DEL MÓDULO DIDÁCTICO

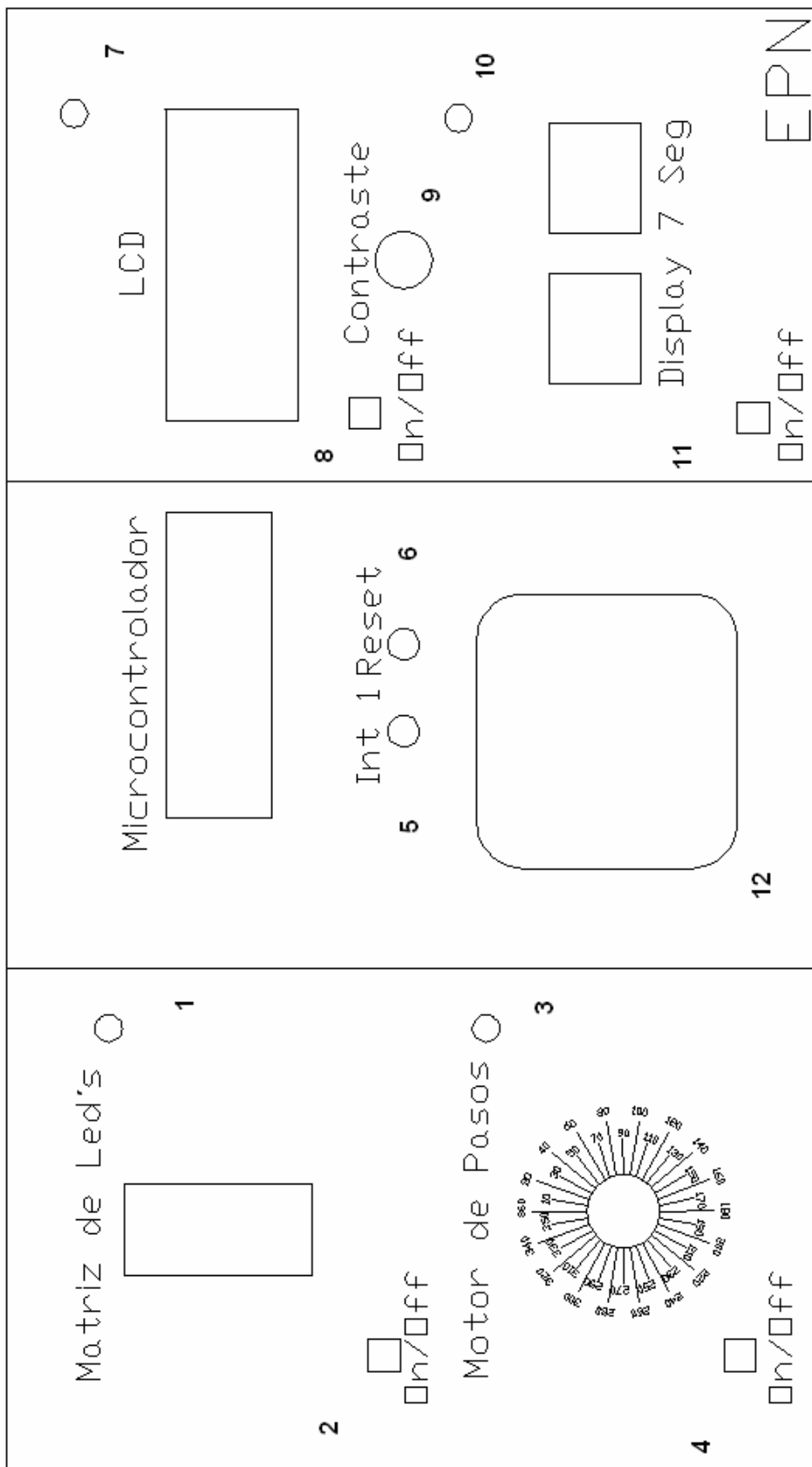
INSTRUCCIONES

Este Módulo Didáctico está diseñado como entrenador para verificar y probar la programación del microcontrolador AT89C51 para uso exclusivo en el Laboratorio de Microprocesadores de la Escuela de Formación Tecnológica.

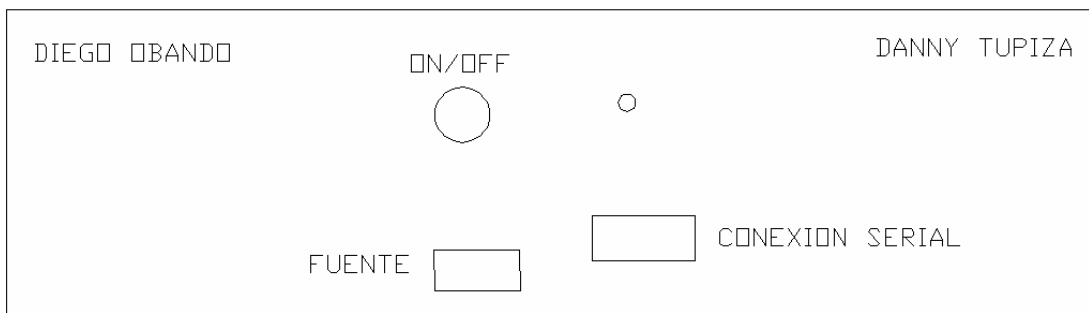
Conozca el Módulo Didáctico.

1. **Led Rojo / Verde**, indica el estado desactivado o activado respectivamente de la tarjeta Matriz de Led's.
2. **Switch** permite activar la tarjeta Matriz de Led's.
3. **Led Rojo / Verde**, indica el estado desactivado o activado respectivamente de la tarjeta Motor de Pasos.
4. **Switch** permite activar la tarjeta Motor de Pasos.
5. **Interrupción** externa 1 del Microcontrolador.
6. **Reset** del Microcontrolador.
7. **Led Rojo / Verde**, indica el estado desactivado o activado respectivamente de la tarjeta LCD.
8. **Switch** permite activar la tarjeta LCD.
9. **Potenciómetro** para control de contraste del LCD.
10. **Led Rojo / Verde**, indica el estado desactivado o activado respectivamente de la tarjeta Display's de 7 segmentos.
11. **Switch** permite activar la tarjeta Display's de 7 segmentos.
12. **Teclado** permite ingresar datos.
13. **Switch** de encendido o apagado del Módulo Didáctico.
14. **Led Amarillo**, indica el estado del Módulo Didáctico
15. **Conexión** de la fuente de poder externa.
16. **DB9** para comunicación serial desde el computador.

Vista Superior.



Vista Frontal.



Partes del Módulo Didáctico.

El Módulo está conformado de las siguientes Partes.

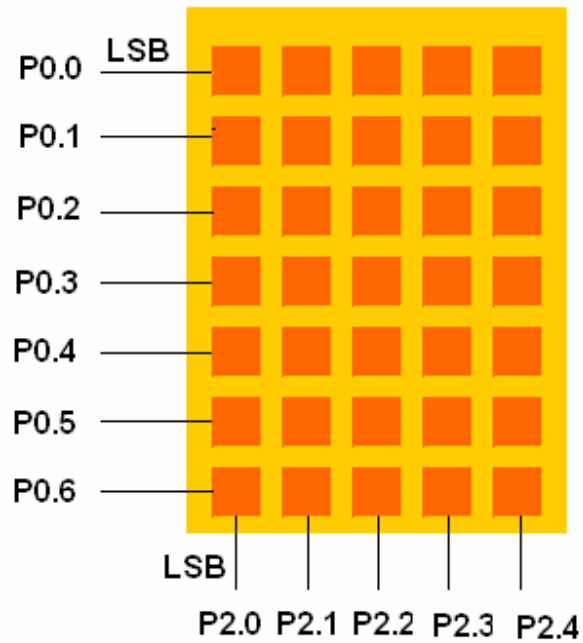
1. Tarjeta Matriz de Led's con su respectiva Matriz.
2. Tarjeta Motor de Pasos, con su respectivo Motor Paso a Paso.
3. Zócalo ZIF para la colocación del Microcontrolador AT89C51.
4. Teclado hexadecimal.
5. Tarjeta LCD, con su respectivo LCD.
6. Tarjeta Display's de 7 segmentos, con dos juegos de display's de 2 cada uno.
7. Comunicación serial, con su respectivo cable.

Conexión

- El Módulo se alimenta con una fuente externa de +5V, GND existentes en el laboratorio.
- Este Módulo está diseñado exclusivamente para probar Microcontroladores AT89C51.

Asignación de pines a de cada tarjeta.

Matriz de Led's.



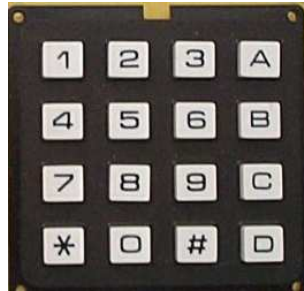
Desde el pórtilo P0.0 al P0.6 se controlan las 7 filas columnas de la matriz, mientras desde el pórtilo P2.0 al P2.4 se controlan las 5 columnas.

Motor de pasos.



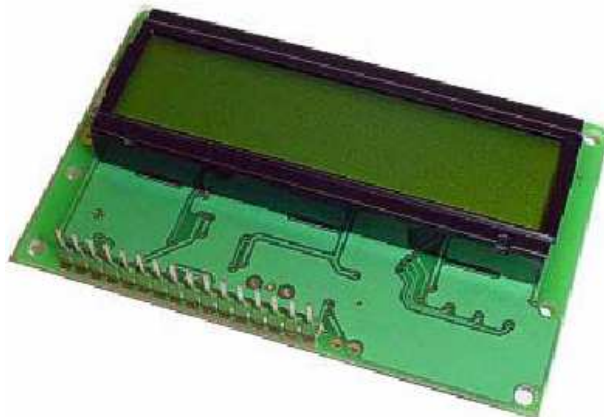
El motor de Pasos es controlado por los p rticos P0 (P0.0 LSB, P0.1, P0.2, P0.3), que llegan desde el Microcontrolador.

Teclado.



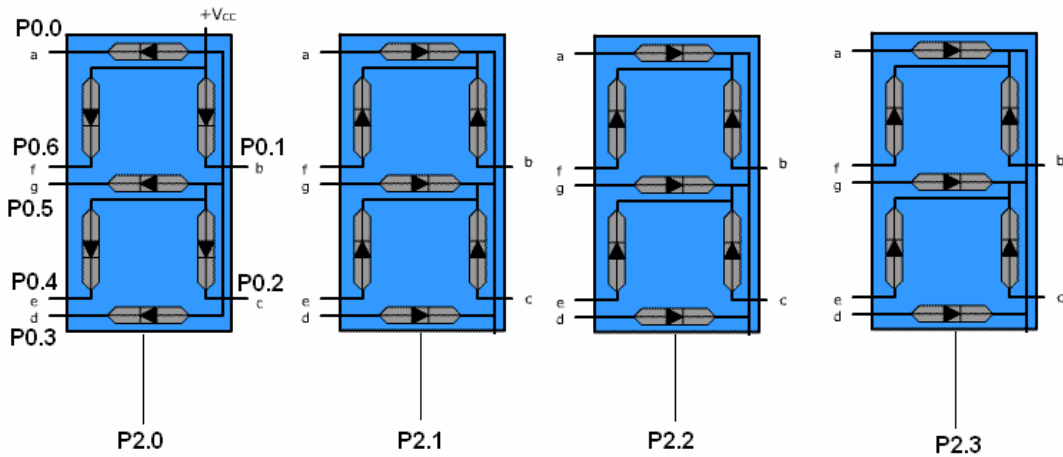
Los datos que son enviados desde el teclado tienen como destino el p rtico P1 (P1.0 LSB, P1.1, P1.2, P1.3) teniendo en cuenta, que al presionar el n mero 1 tendremos el n mero en hexadecimal 0H, hasta llegar al numero FH que resulta al presionar la letra D.

LCD.



Los datos que llegan al LCD provienen desde el p rtico P0 (P0.0 LSB, P0.1, P0.2, P0.3 P0.4, P0.5, P0.6, P0.7) El p rtico P3.6 env a datos para el Enable del LCD y el P3.7 env a datos para controlar el registro de selecci n del LCD.

Display's de 7 segmentos.



Los display's utilizados son de Ánodo común por lo que se encienden con ceros, que son distribuidos de la siguiente manera.

Los datos que controlan los segmentos provienen del Pórtico P0 (P0.0 LSB, P0.1, P0.2, P0.3, P0.4, P0.5, P0.6, P0.7), sabiendo que P0.0 corresponde al segmento "a" del display.

El control para encendido de cada display se realiza por medio del pórtico P2 (P2.0 LSB, P2.1, P2.2, P2.3), los cuales se encenderán al poner cero lógico en cada uno de ellos.

Uso del Módulo Didáctico.

Al encender el módulo todos los led's de indicación de activación de cada tarjeta se encontrarán en color rojo, y no se podrá observar nada.

Para activar una de las tarjetas se debe aplastar el Switch correspondiente con lo cual encenderá un color verde en led respectivo, y podremos observar la información grabada en el microcontrolador AT89C51.