

Universidad Nacional de San Luis
Facultad de Ciencias Físico Matemáticas y Naturales
Ingeniería Electrónica con orientación en Sistemas Digitales
Técnico Universitario en Microprocesadores
Profesorado en Tecnología Electrónica

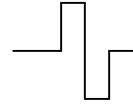
LABORATORIO DE INTERFACES

PRÁCTICO Nº 4

Comunicación Serie

Índice:

1. Objetivos.
2. Material de Referencia
3. Listado de Materiales.
4. Listado de Instrumental.
5. Desarrollo de la Práctica.
6. Comunicación PC – Microcontrolador con Visual Basic.
7. Anexo I - Cables para comunicación serie.
8. Anexo II - Circuitos Integrados Utilizados.



TRABAJO PRÁCTICO N° 4

Comunicación Serie

1. Objetivos

- § Estudiar las características de las normas RS-232, RS422 y RS-485.
- § Realizar la comunicación serie entre dos PC y entre una PC y un Microcontrolador.

2. Material de Referencia

- § Apuntes de la Cátedra.
- § "Microcontroller Projects in C for the 8051" Capítulo 6 - Dogan Ibrahim
- § "Introducción a los Microcontroladores" Capítulo 12 - Gonzáles Vázquez

3. Listado de Materiales

- | | |
|--|------------------|
| 1 Microcontrolador AT89C2051 o AT89C51 | 1 Cap 10uf x 25V |
| 1 Cristal 12MHz | 2 R 4K7 |
| 1 Cristal 11.059.200MHz | 1 R 8K2 |

4. Listado de Instrumental

- | | |
|-----------------------|------------------------------------|
| 1 Tester digital | 1 Fuente de alimentación simétrica |
| 1 Entrenador LAB - MC | 1 Programador Universal Chip Max |

5. Desarrollo de la Práctica

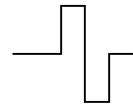
Introducción

Una de las ventajas más importante de la comunicación serie es que utiliza menos líneas de datos cuando efectuamos la comparación con la comunicación paralelo.

La comunicación serie entre dos dispositivos puede realizarse con apenas tres líneas, una línea usada para transmisión (TX), otra para recepción (RX) y una línea común (GND). En cuanto al control de flujo, este puede llevarse a cabo a nivel software utilizando protocolos como XON-XOFF o a nivel hardware mediante líneas adicionales de control. El Control de Flujo por Software (Xon-Xoff) usa dos caracteres Xon y Xoff. Xon es normalmente indicado por el ASCII 17 y Xoff por el carácter ASCII 19.

La transmisión puede ser de tipo Half Duplex o Full Duplex. La velocidad de transmisión posee varios valores estandarizados.

En la transmisión serie, los bits de un carácter se transmiten por una línea única, uno detrás de otro y cada dispositivo tiene su propio reloj. Previamente se ha acordado que ambos dispositivos transmitirán datos a la misma velocidad. En la figura 1 vemos el formato de transmisión serie asíncrono donde podemos ver un bit de comienzo (START), 8 bits de datos (carácter a ser transmitido) y un bit de parada (STOP). Los bits correspondientes al carácter son transmitidos comenzando con los LSB.



Primero se envía un bit de start, luego los bits de datos (LSB primero) y finalmente los bits de stop. El número de bits de datos y de bits de stop es uno de los parámetros configurables, así como el criterio de paridad par o impar para detección de errores. Normalmente, las comunicaciones serie tienen los siguientes parámetros: 1 bit de Start, 8 bits de Datos, 1 bit de Stop y sin paridad.

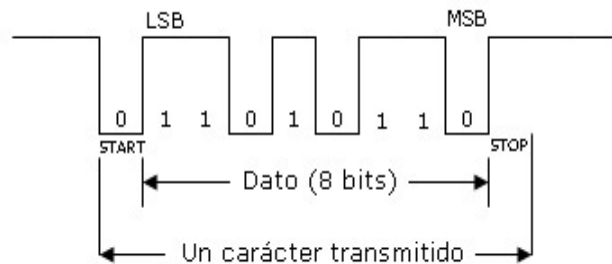


Fig. 1. Transmisión serie del dato 01101011.

El Estándar EIA RS-232C

Características Eléctricas

La norma RS-232C fue desarrollada en 1960 como un estándar para la conexión entre equipos terminales de datos y Modems.

Los niveles lógicos que maneja este estándar no son compatibles con los niveles TTL. Cuando necesitemos conectar dispositivos de lógica TTL a un puerto serie, debemos adaptar las señales RS-232C a TTL y viceversa.

Un receptor RS-232C interpreta una tensión por debajo de $-3V$ como un 1 lógico, y una tensión por encima a $+3V$ como un 0 lógico. Un transmisor RS-232C debe entregar a su salida una tensión inferior $-5V$ para un 1 lógico y una tensión superior a $+5V$ para un 0 lógico. Los niveles de tensión típicos de RS-232C son para un 0 ó 1 lógico de $\pm 3V$ a $\pm 25V$. Estas características permiten alcanzar distancias mayores para los cables y alta inmunidad al ruido.

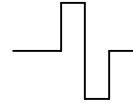
Designación de Señales

El estándar RS-232C fue desarrollado en sus comienzos para la interconexión entre un Modem, denominado "Data Communication Equipment" o DCE, y un Terminal denominado "Data Terminal equipment" o DTE. La tabla 1 muestra la asignación de pines y los nombres de las señales más utilizadas en un conector RS-232C de 25 Pines.

Pin DB-9	Pin DB-25	Nombre	Sentido
3	2	TX	DTE à DCE
2	3	RX	DCE à DTE
7	4	RTS	DTE à DCE
8	5	CTS	DCE à DTE
6	6	DSR	DCE à DTE
5	7	GND	N.A.
1	8	CD	DCE à DTE
4	20	DTR	DTE à DCE
9	22	RI	DCE à DTE

Tabla 1. Descripción de señales.

La descripción de estas señales se puede ver en la tabla 2.



Señal	Descripción
TX (Transmitted Data)	Línea de datos desde el DTE al DCE.
RX (Received Data)	Línea de datos desde el DCE al DTE.
DTR (Data Terminal Ready)	La envía el DTE cuando se halla encendido y funcionando correctamente.
DSR(Data Set Ready)	La envía el DCE cuando se halla encendido y trabajando correctamente.
RTS (Request To Send)	La envía el DTE cuando desea enviar información.
DCD (Data Carrier Detect)	La envía el DCE cuando recibe una portadora del enlace de comunicación (Por ejemplo el tono de una línea telefónica).
CTS (Clear To Send)	La envía el DCE cuando tiene control sobre el enlace.
GND (Ground)	Señal de referencia (tierra).

Tabla 2. Descripción de señales.

Protocolo de Comunicación Serial asincrónico estándar

La norma RS-232C define el protocolo para la comunicación serie. Este protocolo describe reglas para estandarizar la comunicación. El mismo establece un bit de comienzo (START) activo en '0' y un bit de parada (STOP) activo en '1'.

Las velocidades estandarizadas incluyen: 110, 150, 300, 600, 1200, 2400, 4800, 9600 y 19200 bit/seg. Al momento de configurar el puerto serie debemos especificar:

- § Bits por carácter (puede ser 5 a 8 bits por carácter).
- § Número bits de stop (1 o 2).
- § Bit de paridad (se especifica como paridad par, impar o sin paridad).
- § Velocidad de transmisión (se elige una de las antes mencionadas).

Control de Flujo

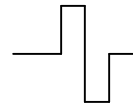
Si la velocidad de transferencia del DTE es mayor que la velocidad a la que el DCE procesa los datos, se hace necesario un control de flujos ya que el buffer del DCE se llenará y perderá los datos recibidos.

El control de flujo se puede realizar software o por hardware. El control de flujo por software, a veces denominado Xon/Xoff utiliza dos caracteres ASCII. Xon, por lo general se indica con el carácter ASCII 17, y Xoff por el carácter ASCII 19.

El DCE posee un buffer de memoria chico, envía el carácter Xoff cuando este se llena, indicándole con esto al DTE que pare el envío de datos. Una vez que el DCE recupera espacio en el buffer, envía el carácter Xon para indicarle al DTE que envíe más datos.

Este tipo de control de flujo posee la ventaja de que no requiere líneas adicionales ya que los caracteres Xon y Xoff se envían a través de las líneas TX-RX. La desventaja es que se disminuye la velocidad de transferencias de datos por el envío de los caracteres Xon y Xoff, los cuales no forman parte del mensaje original.

El control de flujo por hardware, conocido como RTS/CTS, usa dos líneas del cable serial: RTS y CTS. Cuando el DTE quiere enviar datos, activa la línea RTS. Si el DCE tiene espacio para los datos, lo indica activando la línea CTS y así el DTE inicia la transferencia de información. Si el DCE se queda sin espacio en el buffer, entonces desactiva la línea CTS, indicándole al DTE que no envíe más datos.



Como ejemplo podemos mencionar el programa Hyperterminal de Windows, el cual posee control de flujo por Hardware, por software Xon/Xoff y None, este último se usa cuando no se necesita control de flujo.

5.1 La comunicación serie en la PC

El puerto serie de la PC, conocido también como puerto RS-232, es muy útil, ya que permite la comunicación, no solo con otras PC, sino también con otros dispositivos como el mouse, impresoras y, por supuesto microcontroladores. La tabla 3 muestra las direcciones del puerto serie en la PC:

NOMBRE	DIRECCION	IRQ
COM 1	3F8	4
COM 2	2F8	3
COM 3	3E8	4
COM 4	2E8	3

Tabla 3. Dirección del puerto serie.

5.1.1 Características eléctricas de las señales

La Norma RS-232 fue definida para conectar una computadora a un MODEM. Además de transmitirse los datos de una forma serie asíncrona son necesarias una serie de señales adicionales, que se definen en la norma. Las tensiones empleadas están comprendidas entre +15 / -15 voltios.

En RS232, un valor lógico "1", también denominado "marca" se transmite como una tensión negativa y un valor lógico "0", también llamado "espacio" se transmite como un valor de tensión positiva. Esta diferencia de tensión ayuda a que las señales tengan una *mayor inmunidad al ruido*.

Señales de datos	
"0" lógico	+3V a +15V
"1" lógico	- 3V a -15V

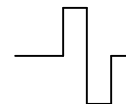
Señales de control	
"Off"	-3V a -15V
"On"	+3V a +15V

Tabla 4. Señales lógicas en RS-232.

5.1.2 Conectores

En la versión "C" de la norma sólo se especificaba un conector de 25 contactos, hembra para el DCE y macho para el DTE. Más allá de eso, no se establecían otros requerimientos para la interfaz mecánica. Esto se completó en la versión "E", donde se especifica el uso de los conocidos conectores DB-25. Un conector de 25 contactos es necesario para acomodar todas las señales de la RS-232, pero en la mayoría de las aplicaciones es suficiente con un subconjunto de ellas, que puede acomodarse en un conector más pequeño, como el DB-9.

El conector tipo DB-9, es muy usado en las PC pero no responde a las especificaciones de la norma EIA/TIA 232. Debido a su popularidad, sin embargo, se diseñó otra norma que si lo emplea, con las mismas especificaciones eléctricas y lógicas que la 232, la norma TIA/EIA 574.



DB-9	DB-25	Nombre	Función
2	3	RD	Receive Data
3	2	TD	Transmit Data
7	4	RTS	Request to send
8	5	CTS	Clear to send
6	6	DSR	Data ready state
5	7	SG	Tierra (Ground)
1	8	DCD	Data Carrier detect
4	20	DTR	Data Terminal Ready
9	22	RI	Ring Indicator

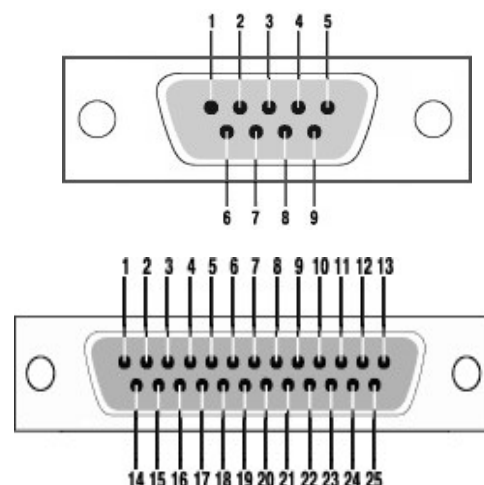


Tabla 5. Identificación de señales.

Fig. 2. Conector DB-9 M y DB-25 M

5.1.4 Observe en la parte posterior del gabinete de la PC, ¿Cuántos conectores de Puerto Serie posee su PC?

.....
.....
.....

5.2 Comunicación entre PCs usando RS-232.

Realizaremos la comunicación serie entre dos computadoras. El carácter introducido por el teclado de una de una de las PC será transmitido por el puerto serie, recibido por la otra PC y mostrado en la pantalla de esta última.

5.2.1 Realice la conexión física de su PC con la de otro grupo mediante el cable de comunicación serie MODEM Null SinHandshake.

5.2.2 Abra la aplicación TestSerie.

5.2.3 En la aplicación TestSerie configure los siguientes parámetros:

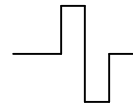
Puerto = COM1.
Velocidad = 1200bps
Bits de datos = 8.
Paridad = ninguna
Bits de Parada = 1.

Luego de configurar estos parámetros debe activar el puerto con el botón Abrir Puerto.

5.2.4 Escriba un texto en el cuadro "TX-Enviar Mensaje" y luego haga Clic en Enviar.

5.2.5 Pruebe modificar distintas velocidades de comunicación cambiando este parámetro en el casillero correspondiente a velocidad.

5.2.6 Antes de cerrar la aplicación asegúrese de Cerrar el Puerto.



5.3 Comunicación serie en Microcontroladores.

Para conectar un microcontrolador a una PC u otro dispositivo serie a veces alcanza con sólo tres señales (GND, TX y RX) o cinco si se emplea algún tipo de control de flujo por hardware, agregándose DTR y DSR o CTS y RTS.

La mayoría de los microcontroladores incluyen un periférico que resuelve la transmisión de datos en formato serie. Tal módulo recibe el nombre de UART, USART o SCI. En micros de bajo costo ese periférico suele estar ausente. En estos casos, la transmisión y recepción de datos serie se puede implementar por software, utilizando retardos precisos para generar las señales, ya sea con el auxilio de un timer o por software.

Comunicación Serie en el 8051

El puerto serie del 8051 puede operar en modo full-duplex, es decir, puede recibir y transmitir simultáneamente. Posee un buffer, el SBUF ubicado en la posmen 99h, que permite acceder a los registros receptor y transmisor del puerto serie. Escribiendo en el SBUF carga el byte a transmitir y leyendo del SBUF se accede al byte recibido.

Los modos en los que puede operar el puerto serie son:

Modo 0: Los datos en serie entran y salen a través de la línea RXD. TXD se usa como señal de reloj. La palabra a transmitir o recibir es de 8 bits. La velocidad se expresa en baudios y esta dada por la relación 1/12 de la frecuencia de reloj del microcontrolador.

Modo1: Es una comunicación acorde con los estándares. Usa 10 bits que se transmiten por TXD o se reciben por RXD. Los bits usados tienen la siguiente función:

1 bit START (0) indica el comienzo - 8 bits de datos (LSB primero) - 1 bit de STOP (1) indica final. La velocidad de transmisión puede ajustarse por el usuario dentro de un amplio rango.

Modo 2: transmite a través de TXD y recibe or RXD igual al Modo, la diferencia es que usa 11 bits: 1 bit START (0) - 8 bits de datos - 1 bit de dato (9° bit) programable - 1 bit de STOP (1).

El noveno bit es la imagen de TB8 del registro SCON y puede usarse para control de paridad. La velocidad de transmisión es programable bien a 1/32 o 1/64 de la frecuencia del oscilador.

Modo 3: idem a modo 2 excepto que la velocidad se puede ajustar en un amplio rango.

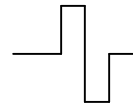
En los 4 modos la transmisión comienza escribiendo en SBUF. La recepción se inicia en MODO 0 por la condición RI=0 y REN=1 y en los modos 1, 2 y3, cuando se recibe el bit START y REN=1.

El registro de control del puerto serie (SCON)

B7				B0			
SM0	SM1	SM2	REN	TB8	RB8	TI	RI

RI: Se activa por hardware al finalizar la recepción del 8° bit en el Modo0 o hacia la mitad del intervalo de tiempo del bit de stop en los otros modos. Debe desactivarse por software.

TI: Se activa por hardware al final de la transmisión del 8° bit en el Modo0 o al comienzo del bit de stop en los otros modos. Debe desactivarse por software.



RB8: En los modos 2 y 3 es el 9° bit que se recibe. En Modo 1, si SM2 = 0, RB8 es el bit de stop. En modo 0 no se usa.

TB8: Corresponde al 9° bit de datos a transmitir de los modos 2 y 3. Es programable por el usuario. Habitualmente es el bit de paridad.

REN: Si REN = 1 (por software) permite la recepción de datos. Si REN=0 no la permite.

SM2: En Modo 2 y 3, si SM2=1 entonces RI no se activará si el 9° bit de datos (RB8) es igual a cero. En Modo 1, si SM2=1 entonces no se activará si el bit de stop no se ha recibido. En modo0, si SM2 debe estar a cero.

SM0-SM1: Especifica el modo de trabajo. 00 = Modo 0, 01= Modo1, 10= Modo2 y 11= Modo 3.

Velocidad de comunicación

Nos interesa el Modo 1 de comunicación ya que es el que se adapta a la comunicación con una PC. La velocidad en Baudios esta dada por:

$$\text{BAUD RATE} = \frac{2^{\text{SMOD}} \times \text{Frecuencia de Oscilación}}{32 \times 12 \times (256 - \text{TH1})}$$

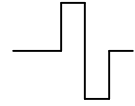
Podemos obtener velocidades de comunicación muy bajas con el Timer 1, permitiendo la interrupción y configurando el Timer para 16 bits (TMOD = 10H) y usando la interrupción del Timer para efectuar la recarga sobre el registro de conteo.

La tabla siguiente muestra algunas velocidades de comunicación comunes y como se debe configurar el timer.

BAUDIOS	FOSC	SMOD	Timer 1		
			C / T	MODO	Valor de Recarga
M0: Max 1MHz	12MHz	X	X	X	X
M2: Max 375K	12MHz	1	X	X	X
M1,3: 62.5K	12MHz	1	0	2	FFh
19200	11.059200MHz	1	0	2	FDh
9600	11.059200MHz	0	0	2	FDh
4800	11.059200MHz	0	0	2	FAh
2400	11.059200MHz	0	0	2	F4h
1200	11.059200MHz	0	0	2	E8h
137.5K	11.986MHz	0	0	2	1Dh
110K	6MHz	0	0	2	72h
110K	12MHz	0	0	1	FEEDh

Ejemplo: se desea obtener una velocidad de comunicación de 9600 baudios con el Timer 1 en modo "autorrecarga" como temporizador. Se usará el Modo 2. Si SMOD = 0, y TH1 = FDh = 253, aplicando la formula:

$$\text{Baudios} = \frac{1}{32} \times \frac{11059 \times 10^3}{(256 - 253) \times 12} = 9599,83 \approx 9600$$



Por lo general interesa obtener el valor del registro TH1 en función de la velocidad en baudios y la frecuencia del oscilador.

$$TH1 = 256 - (F_{cia \text{ Oscilador}} / XXX) / \text{Velocidad en Baudios}$$

Si SMOD = 0 entonces XXX = 384 - Si SMOD = 1 entonces XXX = 192

$$TH1 = 256 - \frac{11.095200 \times 10^3 / 384}{9600} = 256 - 3 = 253 = FDh$$

5.3.1 Utilizando el Keil uVision2, compile los programas TX51.C y RX51.C (uno para cada microcontrolador), luego realice la conexión física de ambos microcontroladores pero conectando el pin TX (P3.1) de uno al pin RX (P3.0) del otro. Un grupo realizara el Transmisor y otro el receptor.

Los programas fueron realizados para una comunicación a 300 bps. El programa de transmisión lee el Puerto 1 cada cierto tiempo y envía el dato. El programa de recepción recibe los datos y los saca por el puerto 1.

Programa de Recepción Serie

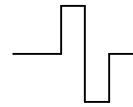
```
#include <AT89x51.h>
void Timer_Init (void);
unsigned char Data_RX;

// Rutina de atención de la interrupción de recepción
void Recibe (void) interrupt 4 {
    Data_RX = SBUF;
    P1 = ~(Data_RX); // P1 Negado para verlo en los Monitores Lógicos
    RI=0;           // Limpio RI
}

void Serial_Init(){
    TMOD=0x20;      // Timer 1 en Modo 2 -8 bit auto recarga
    TH1=TL1=0x98;   // Baudios = 300
    TR1 = 1;        // Habilito Timer 1
    SCON=0x50;      // Seteo REN, recepción en modo 1 y reset RI
}

void main(void){
    Serial_Init();
    IE=0x90;        // Habilito interrupción Serie
    PCON=0x00;
    Data_RX=0x00;

    for(;;);
}
```



Programa de Transmisión Serie

```
#include <AT89x51.h>

unsigned int i;

void Serial_Init(){
    TMOD = 0x20;           // Timer 1 en Modo 2 -8 bit auto recarga
    TH1 = TL1 = 0x98;      // Baudios = 300
    TR1 = 1;               // Habilito Timer 1
    SCON = 0x50;           // Seteo REN, recepción en modo 1 y reset RI
}

void main(void){
    Serial_Init();
    PCON=0x00;

    while(1){
        SBUF = P1;         // Dato a Transmitir
        while (!TI);       // Esperar a que acabe
        for (i=0;i<20000;i++); // No agobiar al compañero
        TI = 0;
    }
}
```

5.3.2 Conexión de un microcontrolador al puerto serie del PC

Para conectar el PC a un microcontrolador por el puerto serie se utilizan las señales TX, RX y GND. La PC utiliza niveles RS-232, por lo que los niveles de tensión de los pines están comprendidos entre +15 y -15 voltios. Los microcontroladores trabajan con niveles TTL (0-5V). Es necesario por tanto adaptar los niveles, un C.I. que permite esto es el MAX232.

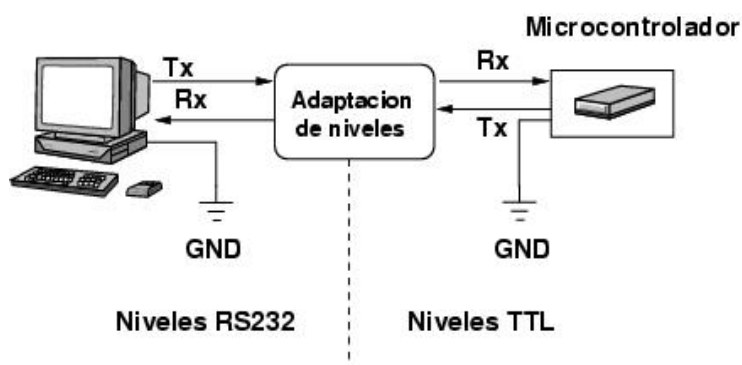


Fig. 3. Conexión de un Microcontrolador a la PC.

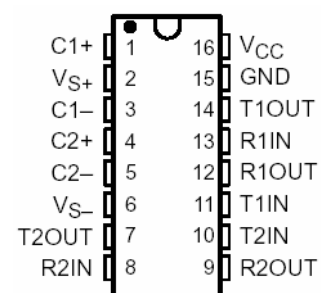
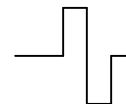


Fig. 4. MAX 232.



5.3.3 El circuito integrado MAX 232

Este dispositivo permite adaptar niveles RS-232 a TTL y viceversa, permitiendo conectar una PC con un microcontrolador. Sólo es necesario este chip y 5 condensadores electrolíticos de 10 μ F.

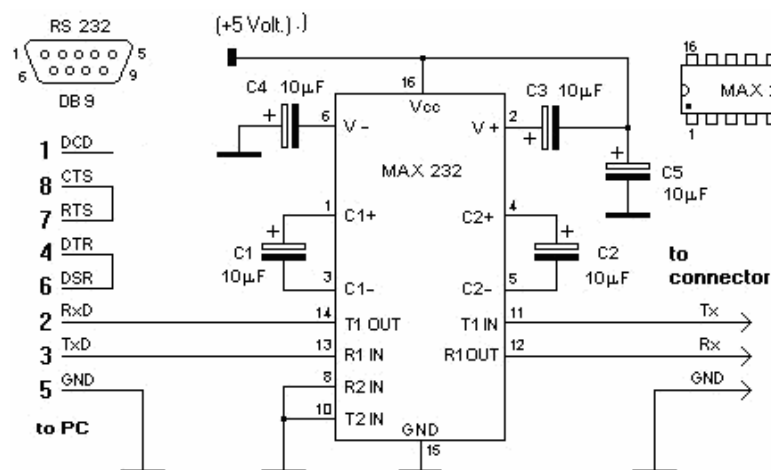


Fig. 5. Uso del C.I MAX 232.

El entrenador LAB-MC incluye un módulo muy completo para comunicación serie RS-232 y RS-485. El esquema del circuito es el que se ve a continuación:

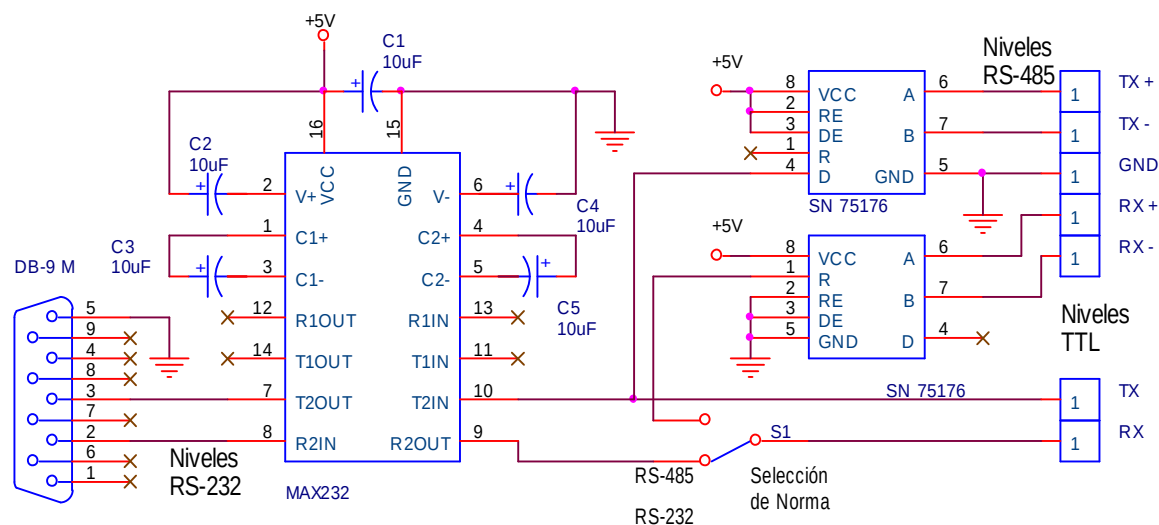
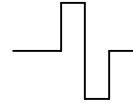


Fig. 6. Módulo de comunicación serie del entrenador LAB – MC.

5.3.4 Ahora se verificarán los niveles de tensión RS-232 que se obtienen del MAX232. Coloque la llave de selección de Norma del panel posterior del LAB-MC en la posición RS-232. Excite el terminal TX sobre el panel superior del LAB-MC con una llave lógica en la posición '0'. Mida ahora con el tester el nivel de tensión en el pin 3 del conector DB-9 (Parte posterior del entrenador).

.....



5.3.5 Excite el terminal TX sobre el panel superior del LAB-MC con una llave lógica en la posición '1'. ¿Cual es el nivel de tensión en el pin 3 del conector DB-9?

.....

5.3.6 Utilizando una fuente variable, excite el pin RX del conector DB-9M con -9V. ¿Cual es el nivel de tensión en el terminal RX del panel superior del entrenador?

.....

5.3.7 Excite el pin RX del conector DB-9M con +9V. ¿Cual es el nivel de tensión en el pin RX del panel superior del entrenador?

.....

5.3.8 Comunicar Microcontroladores usando RS-232.

Conecte ahora los microcontroladores como en el punto 5.2.1 pero agregando la interface RS-232. Utilice el cable MODEM NULL sin handshake para la conexión física.

5.3.9 Puede ver si los datos están siendo enviados y recibidos mirando los LEDs en los paneles de comunicación serie, estos indican la presencia de datos pues están conectados a un circuito monoestable.

5.4 Interface RS-485

El estándar RS-485 fue creado para permitir la conexión de varios dispositivos a través de una línea común.

La transmisión se hace en modo diferencial: el cable (usualmente par trenzado) lleva la señal y su inverso. El receptor detecta la diferencia entre ambas. Como la mayor parte del ruido se acopla a los dos cables por igual, en el receptor se cancela. Esto le otorga mayor inmunidad al ruido que otras interfaces, como la RS232.

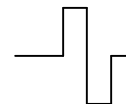
El cable puede tener una longitud de hasta 1200 metros (4000 pies) y la velocidad de transmisión puede llegar hasta 10 Megabits/s. Sin embargo, estos límites no pueden alcanzarse simultáneamente: a mayor longitud del cable, menor velocidad.

La transmisión de los datos se hace generalmente de manera asíncrona, como en RS-232, con un bit de start, bits de datos, un bit de paridad opcional y un bit de stop. Aunque el bus RS-485 es bidireccional (pueden fluir datos en ambos sentidos) la transmisión es half-dúplex, es decir, en un sentido por vez y no simultáneamente.

RS485 no define ningún protocolo para controlar el flujo de datos en el bus. Lo habitual es que un nodo, denominado "MASTER", interrogue al resto, denominados "SLAVES".

El master envía a través del bus un requerimiento que incluye un número o código que identifica a un slave (ID), que al recibirlo, contesta con la información solicitada. También es posible una configuración multi-master, en la que la comunicación pueda ser iniciada por mas de un nodo en la red. Sin embargo deben tomarse recaudos para evitar colisiones cuando dos master intentan transmitir simultáneamente. En resumen, las características sobresalientes de la RS-485 son:

- § Alta inmunidad al ruido.
- § Largo máximo de los conductores: 1200 metros (depende de la velocidad)
- § Alta velocidad (hasta 10 Mega bits/s)
- § Capacidad de conectar hasta 32 nodos
- § Capacidad de configuración en modo Multi-Master



5.4.1 El circuito integrado SN 75176

Estudiaremos este integrado por ser el que utilizaremos en las prácticas para adaptar niveles TTL a RS-485. Este chip se encuentra ubicado en la placa de comunicación serie del entrenador LAB-MC.

Observe en la figura 7 la configuración interna. Posee dos trancéptores de bus diferenciales, diseñados para comunicación bidireccional de datos en líneas de transmisión de bus multipunto. El control se efectúa por medio de dos líneas de habilitación DE y /RE, como se puede ver en la tabla.

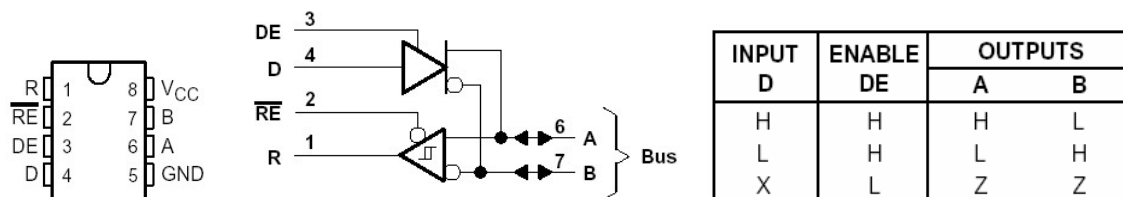


Figura 7. Transceiver SN75176.

Tabla 7. Señales de control.

5.4.2 ¿Según este esquema si se quiere enviar un dato (transmitir) en que estado deben estar DE y /RE?

DE : /RE:

5.4.3 ¿Si lo que se desea es recibir datos en que estado deben estar DE y /RE?

DE : /RE:

5.4.4 Excite la entrada TX del panel superior del LAB-MC y mida las tensiones diferenciales a la salida de este transceptor. Mida estas tensiones en la bornera que se encuentra en la parte posterior del equipo. Los bornes se denominan TX + y TX -. Tenga presente que debe colocar la llave selectora de Norma en la posición RS-485.

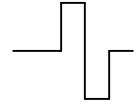
NIVEL	TX +	TX -
TX = '0'		
TX = '1'		

5.4.5 Excite ahora las entradas RX + y RX - del panel posterior (bornera) con tensiones diferenciales de aproximadamente + 4 y - 4 V. Mida la tensión presente en la línea RX del panel superior.

NIVEL	Tensión RX
RX + = + 4V RX - = - 4V	
RX + = - 4V RX - = + 4V	

5.4.6 ¿Son correctos estos valores? ¿Corresponden con niveles lógicos TTL?

.....

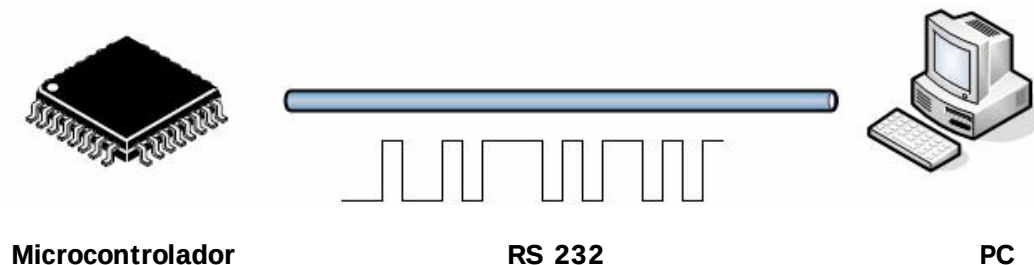


6. Comunicación PC – Microcontrolador con Visual Basic.

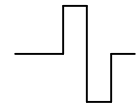
Se realizará la comunicación serie entre un microcontrolador AT89C2051 y la PC realizando una aplicación en Visual Basic.

6.1 Siguiendo los pasos en el tutorial “Comunicación serie entre Microcontroladores y PC”, realice una aplicación que permita leer el puerto serie de la PC y mostrar este dato en la pantalla de la misma. El dato corresponde a un valor de temperatura digitalizado que envía el microcontrolador.

6.2 Grabe en el microcontrolador el programa TX51 que utilizó anteriormente y pruebe la conexión.



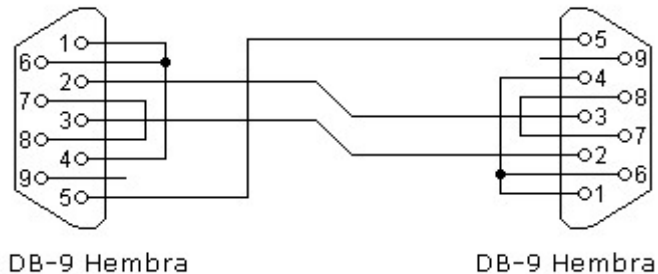
Para aprobar la práctica debe presentar los el informes de laboratorio y el punto 6 funcionando ante el docente a cargo de la práctica.



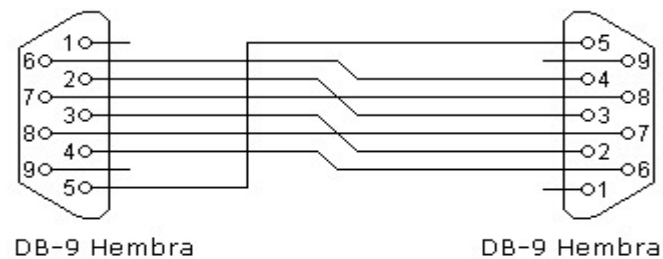
7. Anexo I. Cables para comunicación serie.

Cable MODEM NULL sin handshake con DB-9

Un Modem Null se usa para conectar dos DTE. Es común en redes de juegos o para transferir archivos entre PC's.



Cable MODEM NULL con handshake Usando conectores DB-9

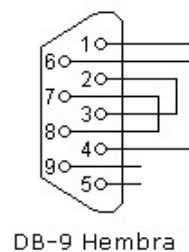


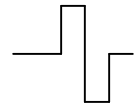
Cable RS-232. Utiliza conectores de tipo DB-9



Loopback para testeo

Este plug loopback es útil cuando se escriben programas de comunicación serie. Posee las líneas RX y TX unidas, por lo tanto cualquier cosa que se transmite por el puerto serie se recibe inmediatamente por el mismo puerto.





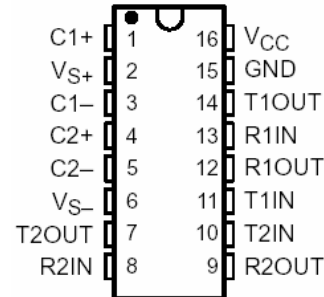
8. Anexo III. Circuitos Integrados Utilizados.

MAX232: Dual RS232 Transmitter and Receiver (5V).

EL dispositivo MAX232 es un driver/receptor doble que incluye un generador de voltaje capacitivo para proveer los niveles de tensión EIA-232 desde una fuente simple de 5V.

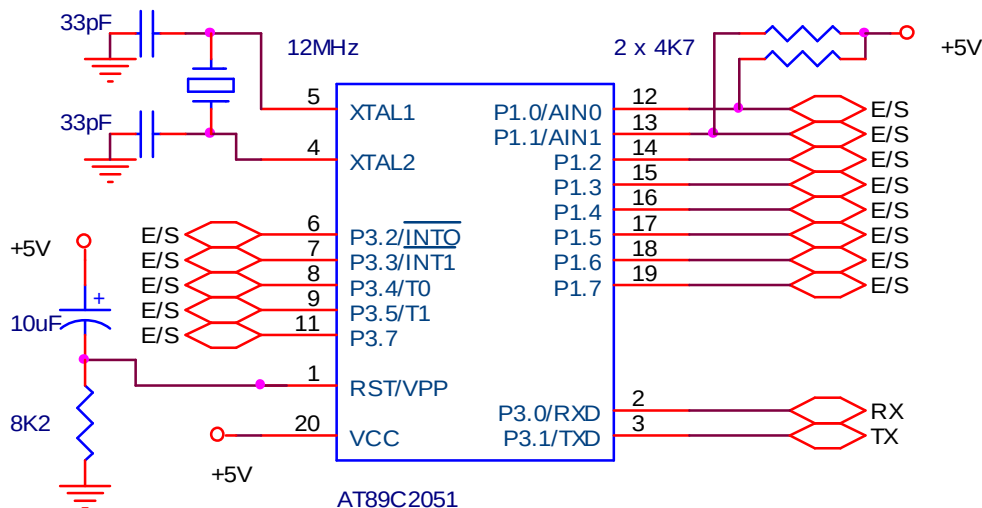
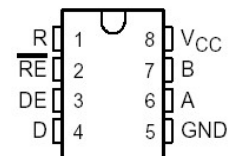
Cada receptor convierte niveles de entrada EIA-232 a niveles TTL/CMOS de 5V.

Cada driver convierte niveles de entrada TTL/CMOS a niveles EIA-232.



SN75176: Differential Bus Transceiver.

EL SN75176 es un circuito integrado transceptor de bus diferencial diseñado para comunicación de datos en forma bidireccional con líneas de transmisión de bus multipunto. Esta diseñado para líneas de transmisión balanceadas para los estándares EIA/TIA-422-B y RS-485.



Esquema de conexiones del circuito con microcontrolador AT89C2051 utilizado para esta práctica.