LA COMUNICACIÓN SERIE

Índice de contenidos

Introducción

Cuando hablamos con alguien, en primer lugar llamamos su atención y entonces se transmite el mensaje, una palabra cada vez. Cuando terminamos, realizamos una pausa para indicar que hemos concluído. Lo mismo se cumple con la lectura o la escritura, se comienza una oración con la letra mayúscula, y lee o escribe una palabra cada vez, con intervalos de cierto período. Estas formas de comunicación humanas son serie, no paralelas.

Los sistemas microprogramables basados en CPU internamente están diseñados para la transferencia de datos en buses o líneas de 8 bits o múltiplos de 8. Así el bus de datos está optimizado para el tratamiento de datos en paralelo lo cual es mucho más rápido que el tratamiento serie.

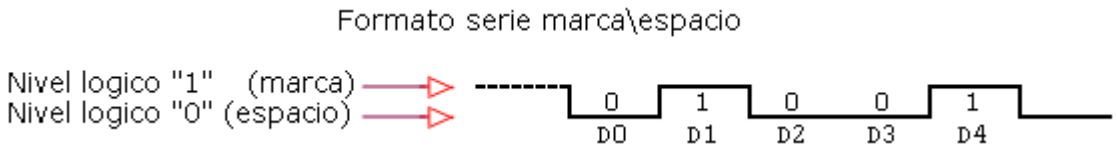
Si la velocidad de transferencia de datos en paralelo es mucho más rápida, ¿porqué se utiliza la transmisión de datos serie?. Algunas respuestan se dan a continuación:

1. Para realizar la comunicación de datos en paralelo se requiere gran cantidad de hilos conductores, pues debe ser establecido un hilo para cada bit de datos, además de las señales de control. Esto encarece notablemente la comunicación en función de la distancia. La comunicación serie requiere 2, 3 ó 4 hilos.
2. Una entrada salida/serie puede ser transmitida a través de pares de cobre, cable coaxial, fibra óptica, vía rádio o vía satélite, lo que proporciona comunicación con equipos remotos (redes locales) o muy remotos (Internet a través de las redes telefónicas y de datos).
3. La comunicación paralelo no posee el alto grado de estandarización que ha alcanzado la comunicación serie, lo que permite la intercominicación entre equipos, por ejemplo mediante RS232, USB o FireWire.

Transmisión modulada en amplitud

Dentro de las múltiples posibilidades existentes nos centraremos en la comunicación serie a través de la interpretación de dos niveles lógicos de tensión o corriente denominado *formato marca/espacio*.

El nivel lógico "1" representa un estado de tensión o corriente denominado marca, el nivel lógico "0" representa un estado de tensión o corriente denominado espacio.



Estándares con formato marca/espacio

Existen varios estandares que usan el formato marca/espacio, de los que nos interesan:

- TTL
- Lazo de corriente de 20mA
- RS232

Niveles para cada estándar.

	Nivel Lógico "1" (Marca)	Nivel Lógico "0" (Espacio)
TTL	5V	0
Lazo 20 mA	20 mA	0 mA
RS 232C	-3V a -15V	+3V a +15V

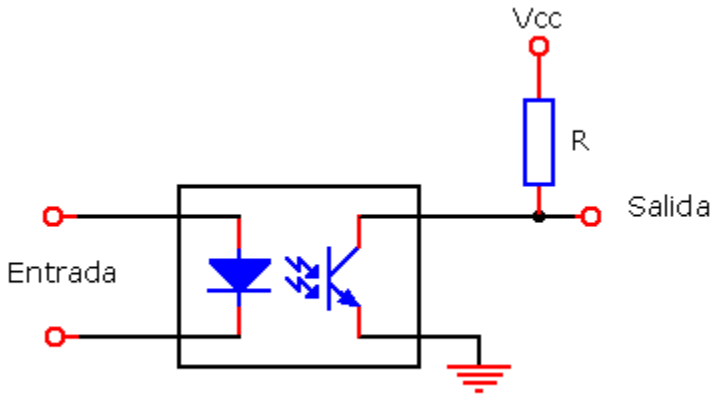
Enlace TTL

Enlace TTL, no recomendable para distancias mayores de 5 m.

Lazo de corriente 20mA

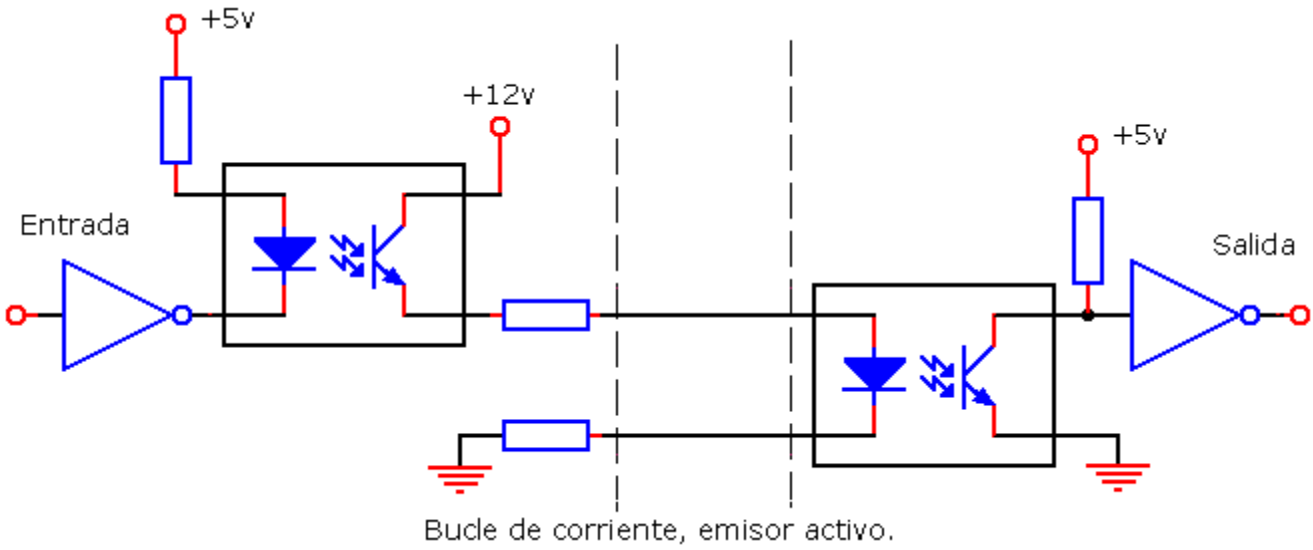
El lazo de corriente de 20 mA es usado para transmitir datos hasta 1609m (1 milla). Para este tipo de interfaz la señal del sistema de datos debe ser convertida a 20mA para ser aceptada por el periférico.

Una forma barata de convertir TTL a 20 mA es usando optoacopladores.

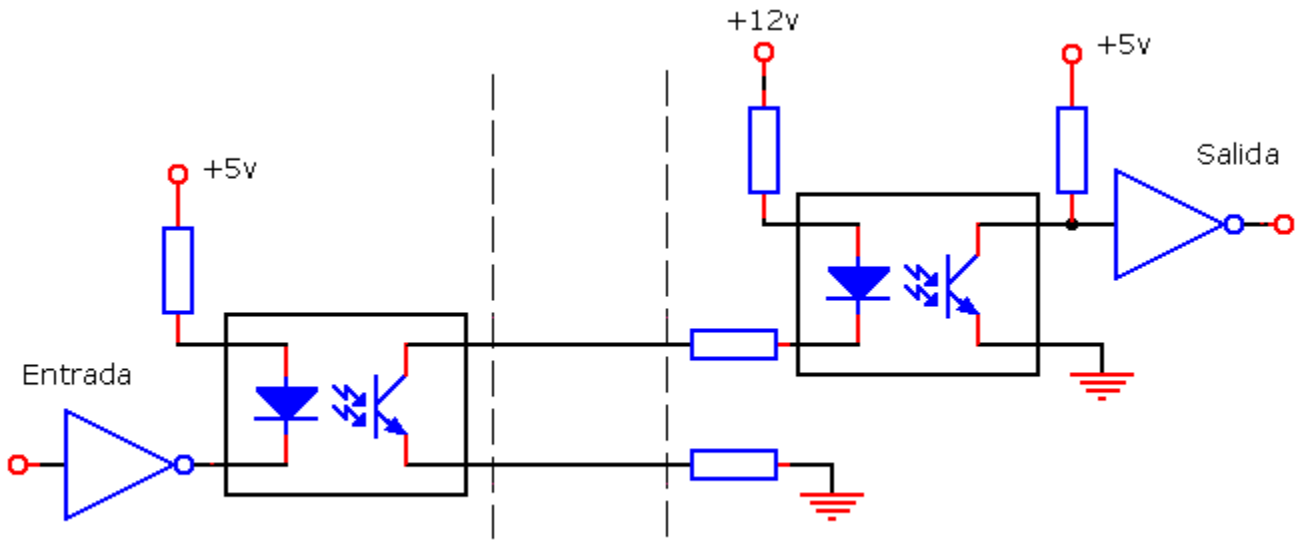


Circuito típico con optoacoplador

Un optoacoplador consiste en un led y un fototransistor los cuales, unidos, trabajan como un réle. Cuando la señal de entrada polariza en directo al led, la luz emitida por el diodo provoca que el transistor conduzca. Como no existe conexión electrica entre el diodo y el transistor, se obtiene un aislamiento eléctrico entre el transmisor y el receptor, que es otra ventaja de este tipo de interfaz.



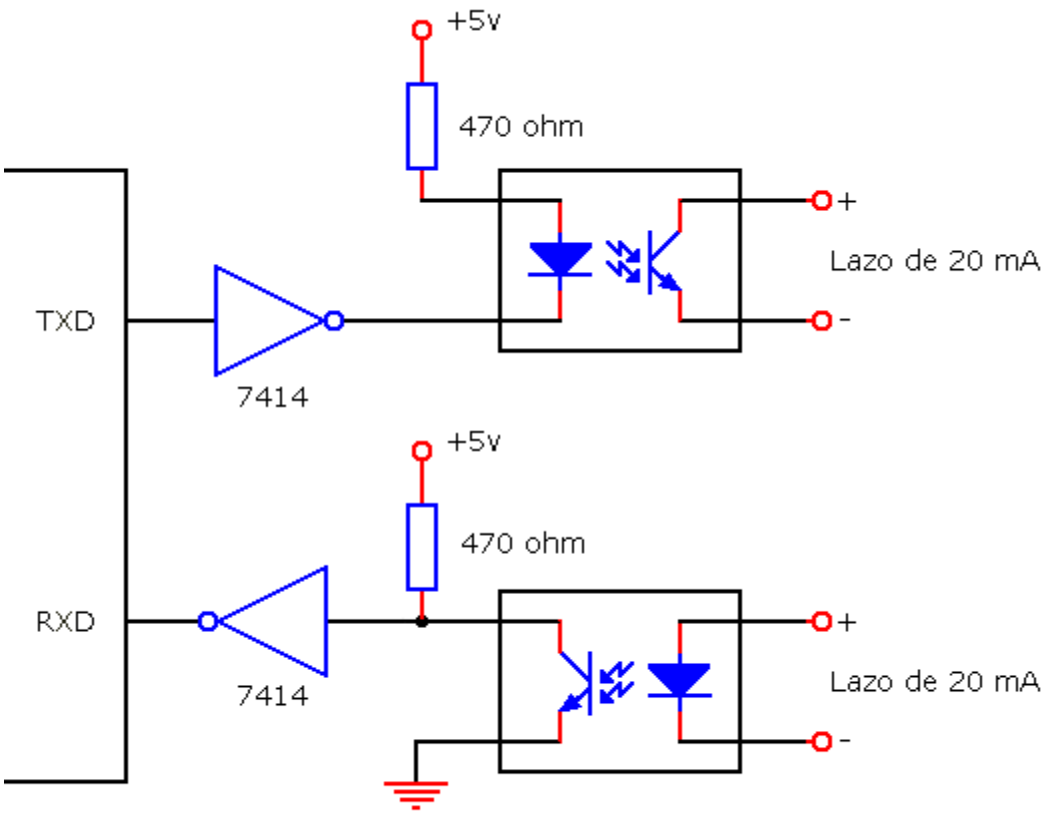
Bucle de corriente, emisor activo.



Bucle de corriente, receptor activo

Interfaz TTL/20mA

En la siguiente figura se muestra como obtener la interfaz TTL/20mA a través de un optoacoplador con una linea transmisora y otra receptora.



Conversión TTL/lazo de corriente de 20 mA

La línea transmisora TXD provee los niveles TTL (0/5V). Un nivel lógico "1" (5V) en TXD será invertido a 0V por el 7414 (un inversor trigger schmitt usado para proveer mejor inmunidad al ruído en el circuíto); este nivel, aplicado al cátodo del led provoca que conduzca, emitiendo luz infrarroja a la base del fototransistor. Ello provoca que el fototransistor pueda conducir. Si un "0" lógico aparece en TXD se aplicarán 5 voltios al cátodo del led y no conducirá, manteniendo el fototransistor en corte y comportándose como un circuito abierto (se abre el lazo de corriente). Es importante notar que el fototransistor no suministra los 20mA , este se comporta solo como un interruptor que cerrado permite que la corriente fluya y abierto impide el paso de corriente.

Cuando 20 mA, o un "1" lógico, es aplicado al optoacoplador de la parte inferior de la figura, el fototransistor conduce y aplica un nivel "0" a la entrada inversora, el cual colocará un "1" lógico en la entrada de la línea receptora RXD. Si no fluye corriente en el lazo, el fototransistor estará al corte y +5V se aplicarán a la entrada inversora por la resistencia de 470 ohm a positivo (en pull-up). El inversor colocará en "0" lógico en RXD, la entrada a la línea receptora.

RS232

Es una de las normas más populares empleadas en la comunicación serie (su inserción en el PC incremento su popularidad). Fue desarrollada en la década de los 60 para gobernar la interconexión de terminales y MODEM. Está patrocinada por la EIA (Asociación de Industrias Eléctricas).

Dado su interés se verá con mas profundidad mas adelante en la norma RS232

Consideraciones en la comunicación serie

Cuando se transmite información a través de una línea serie es necesario utilizar un sistema de codificación que permita resolver los siguientes problemas :

1. Sincronización de bits: El receptor necesita saber donde comienza y donde termina cada bit en la señal recibida para efectuar el muestreo de la misma en el centro del intervalo de cada símbolo (bit para señales binarias).
2. Sincronización del carácter: La información serie se transmite por definición bit a bit, pero la misma tiene sentido en palabras o bytes.
3. Sincronización del mensaje: Es necesario conocer el inicio y fin de una cadena de caracteres por parte del receptor para, por ejemplo, detectar algún error en la comunicación de un mensaje.

Velocidad de transmisión

La velocidad de transmisión de datos es expresada en bits por segundo o baudios. El baudio es un concepto más general que bit por segundo. El primero queda definido como el número de estados de la señal por segundo, si sólo existe dos estados (que pueden ser representados por un bit, que identifica dos unidades de información) entonces baudio es equivalente a bit por segundo. Baudio y bit por segundo se diferencian cuando es necesario más de un bit para representar más de dos estados de la señal.

La velocidad de transmisión queda limitada por el ancho de banda, potencia de señal y ruido en el conductor de señal. La velocidad de transmisión queda básicamente establecida por el reloj. Su misión es examinar o muestrear continuamente la línea para detectar la presencia o ausencia de los niveles de señal ya predefinidos. El reloj sincroniza además todos los componentes internos.

La base de reloj

Cuando se establece la comunicación es necesario implementar una base de tiempo que controle la velocidad. En un microcontrolador, se utilizaría la base de tiempos del reloj del sistema, si bien, en términos genéricos se utilizaría uno de los siguientes métodos:

a. Mediante la división de la base de reloj del sistema. por ejemplo mesiante un contador temporizador programable.
b. A través de un oscilador TTL. Para cambiar frecuencia hay que cambiar el cristal.
c. Generador de razón de baudios. Existen diferentes dispositivos especializados que generan diferentes frecuencias de reloj.

Líneas o canales de comunicación

Se pueden establecer canales para la comunicación de acuerdo a tres técnicas, siempre tamando al microprocesador o microcontrolador como referencia (transmisor) y al periférico como destino (receptor):

a. Simplex
b. Semi duplex (Half duplex)
c. Totalmente duplex (Full duplex)

Simplex: En ella la comunicación serie usa una dirección y una línea de comunicación. Siempre existirá un transmisor y un receptor, no ambos.

La ventaja de este sistema consiste en que es necesario sólo un enlace a dos hilos.

La desventaja radica en que el extremo receptor no tiene ninguna forma de avisar al extremo transmisor sobre su estado y sobre la calidad de la información que se recibe. Esta es la razón por la cual, generalmente, no se utiliza.

Semi duplex: La comunicación serie se establece a través de una sóla línea, pero en ambos sentidos. En un momento el transmisor enviará información y en otro recibirá, por lo que no se puede transferir información en ambos sentidos de forma simultánea .

Este modo permite la transmisión desde el extremo receptor de la información, sobre el estado de dicho receptor y sobre la calidad de la información recibida por lo que permite así la realización de procedimientos de detección y corrección de errores.

Full duplex: Se utilizan dos líneas (una transmisora y otra receptora) y se transfiere información en ambos sentidos. La ventaja de este método es que se puede transmitir y recibir información de manera simultánea.

La mayoría de los dispositivos especializados para la comunicación pueden transferir información tanto en full duplex como en half duplex (el modo simplex es un caso especial dentro de half duplex).

Modos de transmisión

Existen dos modos básicos para realizar la transmisión de datos y son:

- Modo asíncrono.
- Modo síncrono.

Las transmisiones asíncronas son aquellas en que los bits que constituyen el código de un caracter se emiten con la ayuda de impulsos suplementarios que permiten mantener en sincronismo los dos extremos.

En las transmisiones síncronas los caracteres se transmiten consecutivamente, no existiendo ni bit de inicio ni bit de parada entre los caracteres, estando dividida la corriente de caracteres en bloques, enviándose una secuencia de sincronización al inicio de cada bloque.

La transmisión asíncrona

Cuando se opera en modo asíncrono no existe una línea de reloj común que establezca la duración de un bit y el caracter puede ser enviado en cualquier momento. Esto conlleva que cada dispositivo tiene su propio reloj y que previamente se ha acordado que ambos dispositivos transmitirán datos a la misma velocidad.

No obstante, en un sistema digital, un reloj es normalmente utilizado para sincronizar la transferencia de datos entre las diferentes partes del sistema. El reloj definirá el inicio y fin de cada unidad de información así como la velocidad de transmisión. Si no existe reloj común, algún modo debe ser utilizado para sincronizar el mensaje.

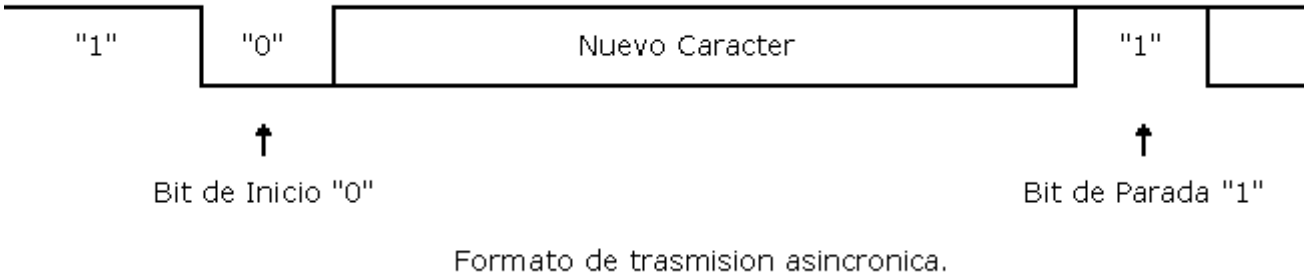
En realidad, la frecuencia con que el reloj muestrea la línea de comunicación es mucho mayor que la cadencia con que llegan los datos. Por ejemplo, si los datos están llegando a una cadencia de 2400 bps, el reloj examinará la línea unas 19200 veces por segundo, es decir, ocho veces la cadencia binaria. La gran rapidez con que el reloj muestrea la línea, permite al dispositivo receptor detectar una transmisión de 1 a 0 o de 0 a 1 muy rápidamente, y mantener así la mejor sincronización entre los dispositivos emisor y receptor.

El tiempo por bit en una línea en que se transfiere la información a 2400 bps es de unos 416 microsegundos (1 seg/2400). Una frecuencia de muestreo de 2400 veces por segundo nos permitirá muestrear el principio o el final del bit. En ambos casos detectaremos el bit, sin embargo, no es extraño que la señal cambie ligeramente, y permanezca la línea con una duración un poco más larga o más corta de lo normal. Por todo ello, una frecuencia de muestreo lenta no sería capaz de detectar el cambio de estado de la señal a su debido tiempo, y esto daría lugar a que la estación terminal no recibiera los bits correctamente.

Bit de inicio y bit de parada

En la transmisión asíncrona un carácter a transmitir es encuadrado con un indicador de inicio y fin de carácter, de la misma forma que se separa una palabra con una letra mayúscula y un espacio en una oración. La forma estándar de encuadrar un carácter es a través de un bit de inicio y un bit de parada.

Durante el intervalo de tiempo en que no son transferidos caracteres, el canal debe poseer un "1" lógico. Al bit de parada se le asigna también un "1". Al bit de inicio del carácter a transmitir se le asigna un "0". Por todo lo anterior, un cambio de nivel de "1" a "0" lógico le indicará al receptor que un nuevo carácter será transmitido.



Formato de trasmision asincronica.

Reglas de transmisión asíncrona

La transmisión asíncrona que vamos a ver es la definida por la norma RS232, en la que profundizaremos más adelante y que se basa en las siguientes reglas:

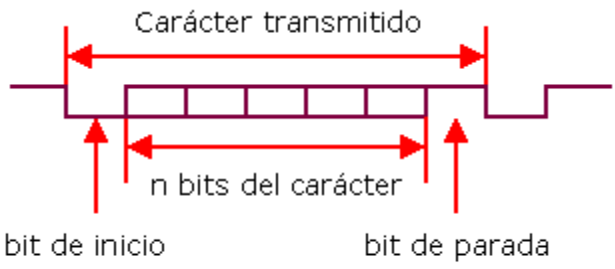
a. Cuando no se envían datos por la línea, ésta se mantiene en estado alto (1).
b. Cuando se desea transmitir un carácter, se envía primero un bit de inicio que pone la línea a estado bajo (0) durante el tiempo de un bit.
c. Durante la transmisión, si la línea está a nivel bajo, se envía un 0 y si está a nivel alto se envía un 1.
d. A continuación se envían todos los bits del mensaje a transmitir con los intervalos que marca el reloj de transmisión. Por convenio se transmiten entre 5 y 8 bits.
e. Se envía primero el bit menos significativo, siendo el más significativo el último en enviarse.
f. A continuación del último bit del mensaje se envía el bit (o los bits) del final que hace que la línea se ponga a 1 por lo menos durante el tiempo mínimo de un bit. Estos bits pueden ser un bit de paridad para detectar errores y el bit o bits de stop, que indican el fin de la transmisión de un carácter.

Los datos codificados por esta regla, pueden ser recibidos siguiendo los pasos siguientes:

a. Esperar la transición 1 a 0 en la señal recibida.
b. Activar el reloj con una frecuencia igual a la del transmisor.
c. Muestrear la señal recibida al ritmo de ese reloj para formar el mensaje.
d. Leer un bit más de la línea y comprobar si es 1 para confirmar que no ha habido error en la sincronización.

Velocidad de transmisión

En la transmisión asíncrona por cada caracter se envía al menos 1 bit de inicio y 1 bit de parada así como opcionalmente 1 bit de paridad. Esta es la razón de que los baudios no se correspondan con el número de bits de datos que son transmitidos.



Formato básico de transmisión asincrónica

Ejemplo:

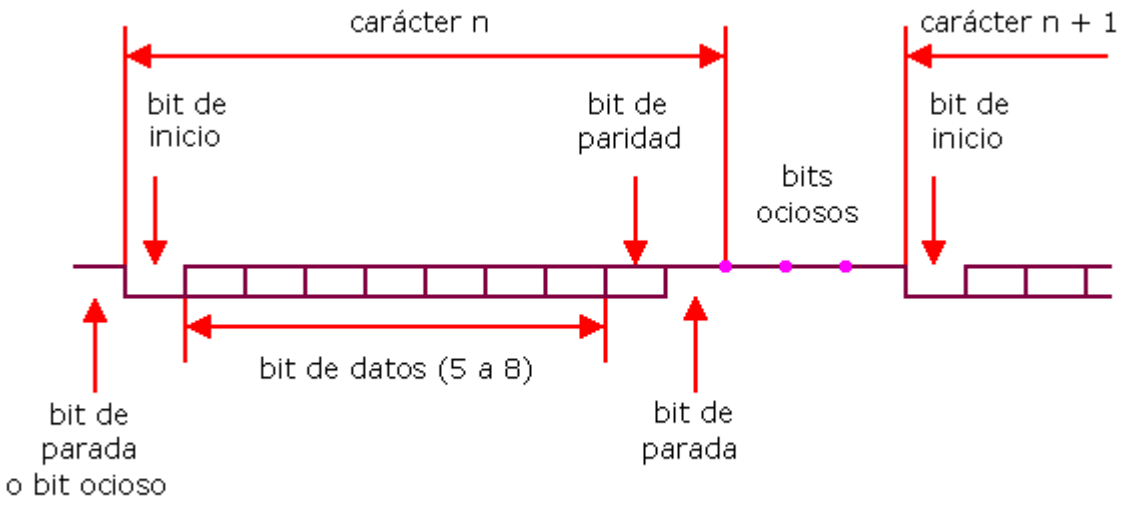
Determinar cuántos bits de datos y caracteres son transmitidos de manera asíncrona en 1 segundo si se transmite a una velocidad de 2400 baudios con 1 bit de inicio, 2 bits de parada, 1 bit de paridad y 6 bits de datos por caracter:

Para transmitir un caracter se necesitará:

1 bit inicio + 6 bits datos + 1 bit paridad + 2 bits parada =10 bits.

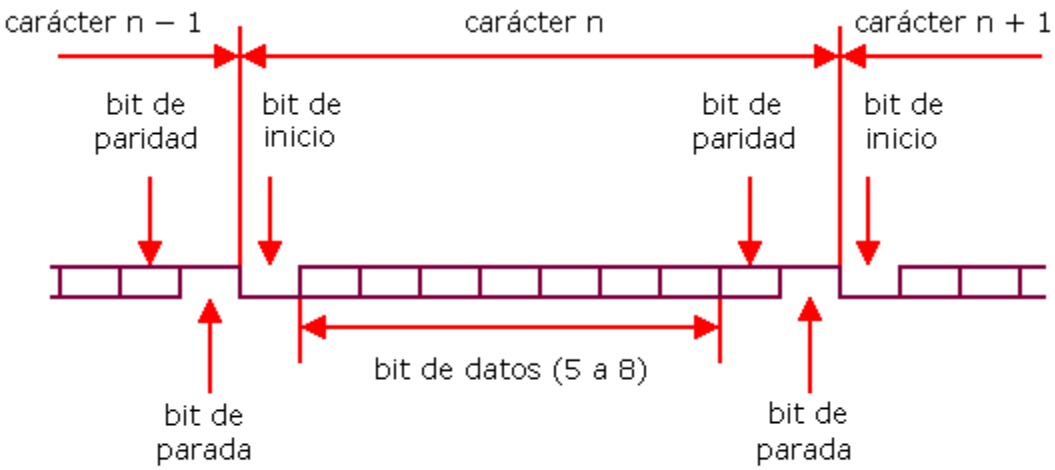
Como la velocidad de transmisión es 2400 baudios y cada caracter consume 10 bits, se transmitirán 240 caracteres por segundo (2400/10). Como cada caracter posee 6 bits de datos serán transmitidos 240 * 6 = 1440 bits de datos por segundo.

La característica fundamental del formato de transmisión asíncrono es su capacidad de manejar datos en tiempo real, con un intervalo de longitud arbitraria entre caracteres sucesivos. Al final de cada caracter, la línea va a 1 en el bit de parada y permanece en ese estado durante un número arbitrario de bits ociosos. El inicio del nuevo caracter estará definido por la transición a 0 del bit de inicio.



Transmisión asincrónica con velocidad menor que la máxima posible

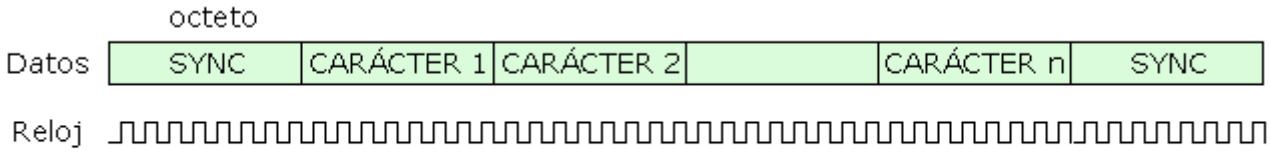
En la siguiente figura se muestra la mayor velocidad asíncrona posible con el bit de paridad.



Transmisión asincrónica con la velocidad máxima posible

La transmisión síncrona

Es un método más eficiente de comunicación en cuanto a velocidad de transmisión. Ello viene dado porque no existe ningún tipo de información adicional entre los caracteres a ser transmitidos.



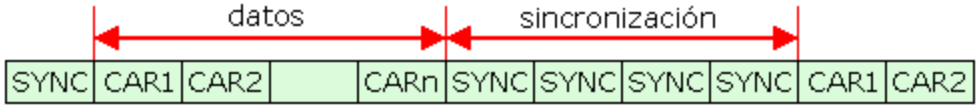
Transmision sincrónica

Cuando se transmite de manera síncrona lo primero que se envía es un octeto de sincronismo ("sync"). El octeto de sincronismo realiza la misma función que el bit de inicio en la transmisión asíncrona, indicando al receptor que va ha ser enviado un mensaje. Este carater, además, utiliza la señal local de reloj para determinar cuándo y con qué frecuencia será muestreada la señal, es decir, permite sincronizar los relojes de los dispositivos transmisor y receptor. La mayoría de los dispositivos de comunicación llevan a cabo una resincronización contra posibles desviaciones del reloj, cada uno o dos segundos, insertando para ello caracteres del tipo "sync" periódicamente dentro del mensaje.

Los carácteres de sincronismo deben diferenciarse de los datos del usuario para permitir al receptor detectar los caracteres "sync". Por ejemplo, el código ASCII utiliza el octeto 10010110.

Existen ocasiones en que son definidos dos caracteres de sincronismo, ello puede ser necesario si, por cualquier motivo el caracter "sync" original se desvirtuara, el siguiente permitirá la reinicialización del receptor. En segundo lugar, puede ocurrir que el equipo receptor necesite un tiempo adicional para adaptarse a la señal entrante.

Cuando se transmite de forma síncrona, es necesario mantener el sincronismo entre el trasnmisor y el receptor cuando no se envían caracteres, para ello son insertados caracteres de sincronismo de manera automática por el dispositivo que realiza la comunicación.



Inserción automática de caracteres de sincronismo

El receptor/transmisor síncrono debe indicar además cuándo el sincronismo ha sido logrado por parte del receptor.

Detectar errores en la comunicación

Cuando se escriben o se envían datos, pueden producirse errores, entre otras cosas, por ruidos inducidos en las líneas de transmisión de datos. Es por tanto necesario comprobar la integridad de los datos transmitidos mediante algún método que permita determina sis se ha producido un error.

En un caso típico, si al transmitirse un mensaje se determina que se ha producido un error, el receptor solicita de nuevo el mensaje al emisor.

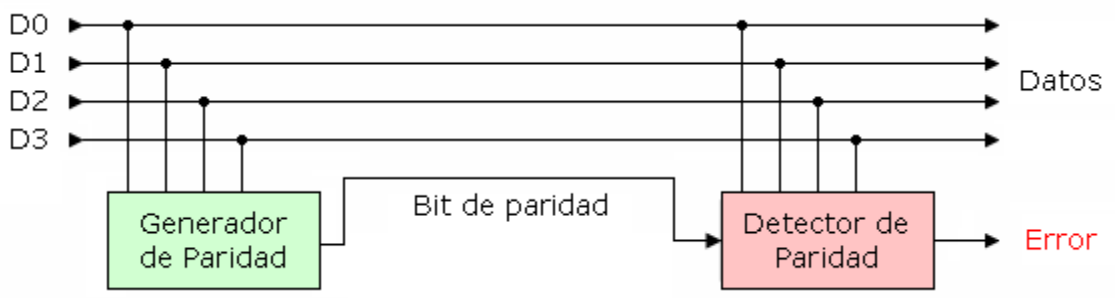
Se pueden detectar errores de acuerdo a la forma de transmisión:

1. Transmisión asíncrona:
    a. Paridad.
    b. Sobre escritura.
    c. Error de encuadre (framing).
2. Transmisión síncrona:
    a. Paridad.
    b. Sobre escritura.

Generadores y detectores de paridad

Como un error en una transmisión serie solamente suele afectar a un bit, uno de los métodos más comunes para detectar errores es el control de la paridad.

El control de paridad consiste en añadir un bit, denominado de paridad, a los datos que se envían o escriben.



La paridad puede ser par o impar.

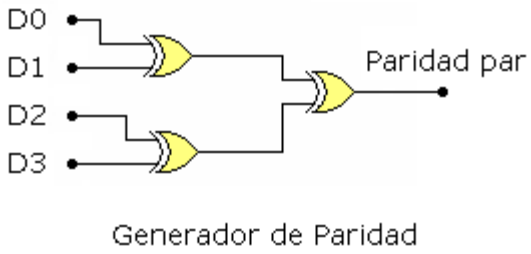
Paridad par

El bit de paridad será cero, cuando el número de bit "unos" que contienen los datos a transmitir sea un número par, y el bit de paridad será uno cuando los datos que se mandan contienen un número impar de unos.

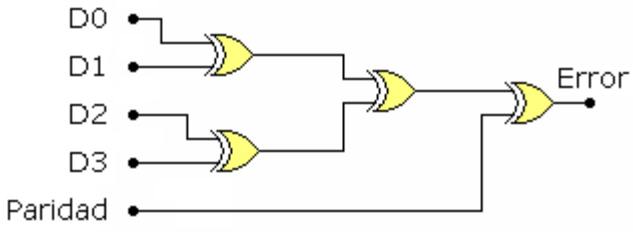
Dato	Paridad
0000 0001	1
0101 0001	1
0101 0101	0
0000 0000	0

La suma de los bits que son unos, contando datos y bit de paridad dará siempre como resultado un número par de unos.

En las siguientes figuras se muestra como se puede realizar un generador de paridad y un detector de paridad con puertas lógicas or-exclusivas (EXOR).



Generador de Paridad



Detector de Paridad

Paridad impar

Dato	Paridad

0000 0001	0
0101 0001	0
0101 0101	1
0000 0000	1

En el sistema de paridad impar, el número de unos (datos + paridad) siempre debe ser impar.

Ejemplo:

Se quieren transmitir los datos C3H y 43H con paridad impar.

- C3H = 1100 0011
- 43H = 0100 0011

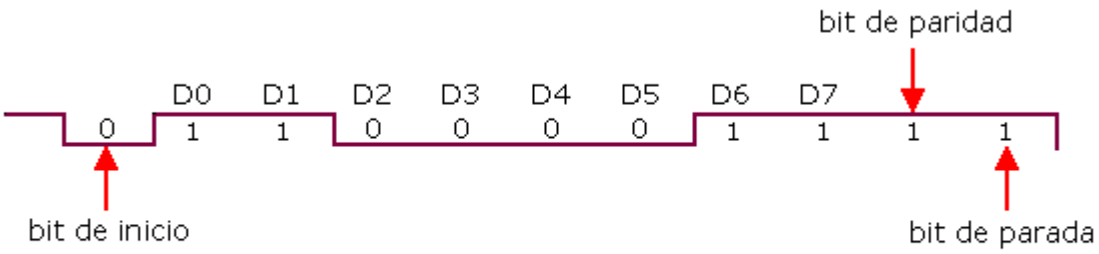
C3H tiene un número par de unos, por lo que el bit de paridad a insertar debe ser 1 para que se cumpla que el número de unos (datos + paridad) siempre debe ser impar:

D0	D1	D2	D3	D4	D5	D6	D7	BIT DE PARIDAD	
1	1	0	0	0	0	1	1	1	5 "unos"

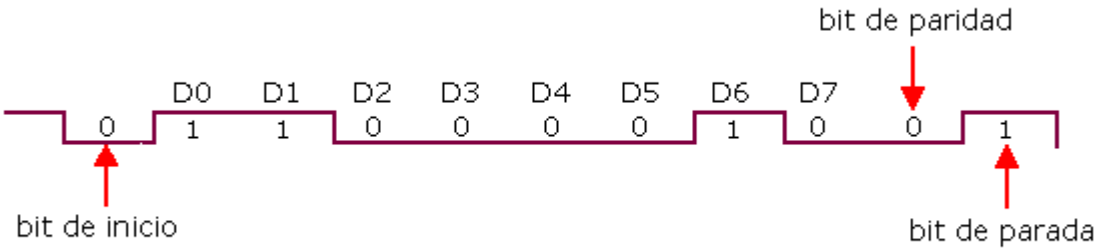
43H tiene un número impar de unos, por lo que el bit de paridad a insertar debe ser 0 para que se cumpla que el número de unos (datos + paridad) siempre debe ser impar:

D0	D1	D2	D3	D4	D5	D6	D7	BIT DE PARIDAD	
1	1	0	0	0	0	1	0	0	3 "unos"

La secuencia de transmisión se muestra en la figura siguiente.



Transmisión del carácter C3H, 1100 0011B.



Transmisión del carácter 43H, 0100 0011B.

Supongamos que se comete un error en la recepción de 43H en la posición más significativa (D7). Esto significa que se ha recibido el caracter C3H (bit MSB complementado), el receptor discrimina este error al recibir un número par de unos (bit D0, D1, D6, D7 y paridad):

D0	D1	D2	D3	D4	D5	D6	D7	BIT DE PARIDAD	
1	1	0	0	0	0	1	"1"	0	4 "unos"

Como fue definida paridad impar, se detecta el error pues deberia haber un número impar de unos y hay cuatro. Como respuesta a la detección el sistema podría solicitar la transmisión de este caracter nuevamente.

Por último, y considerándo lo anterior, indicar que el método de detección de errores mediante paridad sólo es válido cuando falla un bit, si por ejemplo fallan dos, no se detectará el error.

Método checksum

Puede existir el caso en que, por ejemplo, se alteren dos bits en un caracter transmitido y si se ha implementado la comprobación de paridad, el error no será detectado.

Existen otros métodos de detección de errores como son la comprobación de redundancia cíclica (CRC) y la comprobación de suma (checksum). Por su simplicidad, será abordado el método checksum.

El método checksum puede ser utilizado tanto en la transmisión síncrona como en la asíncrona. Se basa en la transmisión, al final del mensaje, de un byte (o bytes) cuyo valor sea el complemento

a dos de la suma de todos los carácteres que han sido transmitidos en el mensaje. El receptor implementará una rutina que suma todos los bytes de datos recibidos y al resultado se le sumará el último byte (que posee la información en complemento a dos de la suma de los caracteres transmitidos) y si la recepción del mensaje ha sido correcta, el resultado debe ser cero.

Ejemplo:

Indicar el último caracter a transmitir cuando se implementa el método de checksum. Los datos a transmitir serán 40H, 35H y 0EH.

Se realiza la suma:

40H + 35H + 0EH = 83H

Ahora se determina el complemento a dos del resultado:

83H = 1000 0011
0111 1100
+ 1

0111 1101 = 7DH

El último carácter a transmitir será 7DH.

Conversión serie/paralelo.

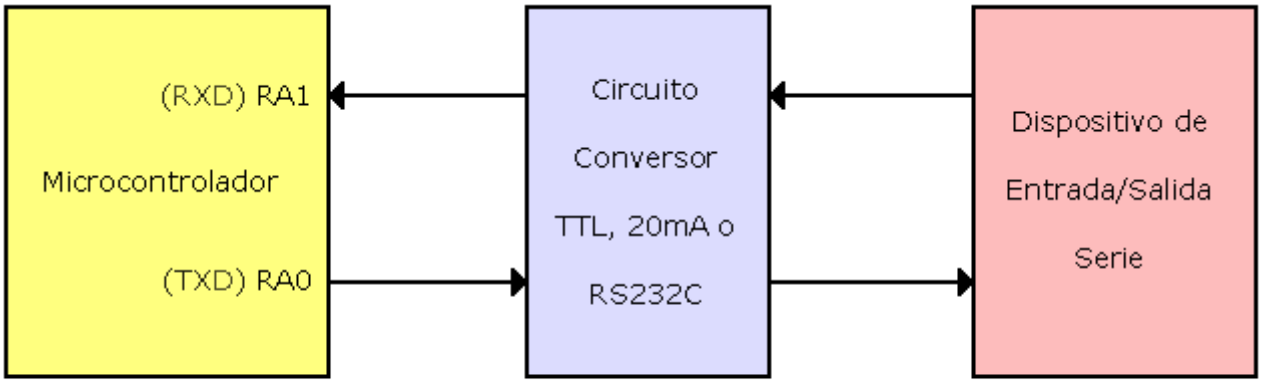
Como un sistema microprogramable basado en CPU es un dispositivo que inherentemente maneja los datos de forma paralela, debe realizarse una conversión para obtener el formato de datos serie que requiere los dispositivos periféricos que pueda tener conectados a través de la comunicación serie.

Existen dos formas en se pueden implementar la conversión serie/paralelo. Esta puede ser conversión por software y conversión por hardware.

Conversión por software

Para este método es necesario utilizar un terminal de salida de un puerto conectado al microprocesador o una patilla del microcontrolador. De la misma manera para la recepción se necesitará un terminal de entrada.

A modo de ejemplo en la siguiente figura se muestra la utilización de un microcontrolador PIC donde RA0 se utilizará como salida (transmisor) y RA1 como entrada (receptor). El modo de transmisión será asíncrono.



Circuito de interfaz para el metodo de conversion serie/paralelo por software.

Como ejemplo se implementará una rutina de conversión para la comunicación asíncrona con 1 bit de inicio y 2 bits de parada con una velocidad de 300 bauds.

La transmisión de datos

Debe realizarse una rutina que permita la transmisión de datos por la patilla RA0 del microcontrolador.

La rutina de transmisión puede ser llamada por interrupción, cada vez que un dato serie quiera ser transmitido o por que se establezca en una o varias partes del flujo del programa que ejecuta el microcontrolador.

Se asumirá que serán transmitidos 11 bits. Un bit de inicio, 8 bits de datos y 2 bits de parada. Se asumirá que los 8 bits a transmitir estarán en el registro de trabajo o acumulador y que la frecuencia de transmisión será de 300 baudios.

La rutina debe en primer lugar colocar la línea a cero (que debe previamente estar en "1") durante el tiempo de un bit para establecer el bit de inicio, posteriormente el acumulador será enviado a puerto RA0, lo que transmitirá el bit menos significativo de datos a través de RA0.

Después de un tiempo equivalente a la duración de 1 bit, un contador debe ser establecido para controlar el número de veces que el acumulador debe ser rotado a la derecha para transmitir cada bit, el acumulador es entonces rotado a la derecha y se envia a RA0 su contenido para transmitir el segundo bit de dato. El bit a ser transmitido estará siempre en el bit menos significativo del acumulador. El procedimiento de rotación y demora será repetido hasta que los 8 bits de datos sean transmitidos, finalmente la línea (RA0) será colocada a 1 lógico durante el tiempo de dos bits.

La recepción de datos

Una rutina de recepción por software puede ser llamada cada un milisegundo para detectar la presencia de un bit de inicio, o por interrupción. La rutina lee la entrada RA1 y monitoriza su estado. Si el valor en RA1 es "1", se debe retornar al programa principal, si es "0", es indicador de la

presencia de un bit de inicio y se deberá comenzar el ensamblaje del carácter que está transmitiendo el periférico remoto.

Cuando el bit de inicio es detectado, se espera el intervalo de tiempo equivalente a la duración de 1/2 bit para comprobar nuevamente la entrada RA1, esto se realiza para asegurar el muestreo del bit de inicio en el centro de su intervalo y evitar iniciar la recepción cuando el "0" haya sido provocado por ruidos en la línea. Para obtener el tiempo de 1/2 bit se tiene en consideración la frecuencia de transmisión. La duración de un simbolo a transmitir a 300 bauds será 1/300 de segundo, o lo que es lo mismo 3,33 mseg; por ello la duración de 1/2 bit será de 1,667 mseg.

Después de haberse asegurado la presencia del bit de inicio, se espera el tiempo de duración de 1 bit para encuestar la primera unidad de información del dato a recibir en la mitad de su intervalo. Posteriormente, un contador es establecido con el número de bits de datos que serán recibidos, el cual decrementará con cada bit recibido.

El bit leído en PA1 pasa al acumulador y se rota a la izguierda a través del acarreo, posteriormente se realiza una rotación a la derecha con el acarreo al registro donde debe almacenarse el resultado de la recepción. Esta operación de doble rotación permite que un bit sea recibido sin que el bit previo sea destruído. El acarreo actúa como almacenador intermedio entre el acumulador y el registro.

Después que el primer bit de dato es recibido, el contador será decrementado después de una demora de un bit, que permitirá comprobar el próximo bit de dato a recibir en la mitad del intervalo. Este proceso continúa hasta que los 8 bits de datos queden almacenados en el registro. Cuando el proceso se ha completado, el primer bit recibido estará almacenado en el bit menos significativo del registro y el último dato en el bit más significativo del registro.

Después de que los bits de datos han sido recibidos, el programa debe chequear los bits de parada. Para ello el valor presente en RA1 será rotado dos veces a la izquierda con acarreo, si en algún momento el acarreo no vale "1", habrá existido un error de encuadre (framing) y deberá ser abortada la recepción. Si los dos bits están presentes, el dato presente en el registro debe ser salvado por el microcontrolador y retornar al programa principal.

A rutina se le puede añadir una sección que compruebe un bit de paridad. Después de aceptar los 8 datos, se puede tomar un noveno (con información de paridad) y comprobar su estado de manera tal que si es detectada una paridad incorrecta, sea generado un mensaje de error y el dato se ignore.

La ventaja de la conversión por software es la simplicidad en el hardware. La desventaja radica en que existe una pérdida en la eficiencia del trabajo del sistema por el tiempo que se debe tomar en la ejecución de estas rutinas. También la velocidad de transmisión puede quedar comprometida. Por todo ello, en la mayoría de los sistemas la conversión serie/paralelo es implementada en términos de hardware.

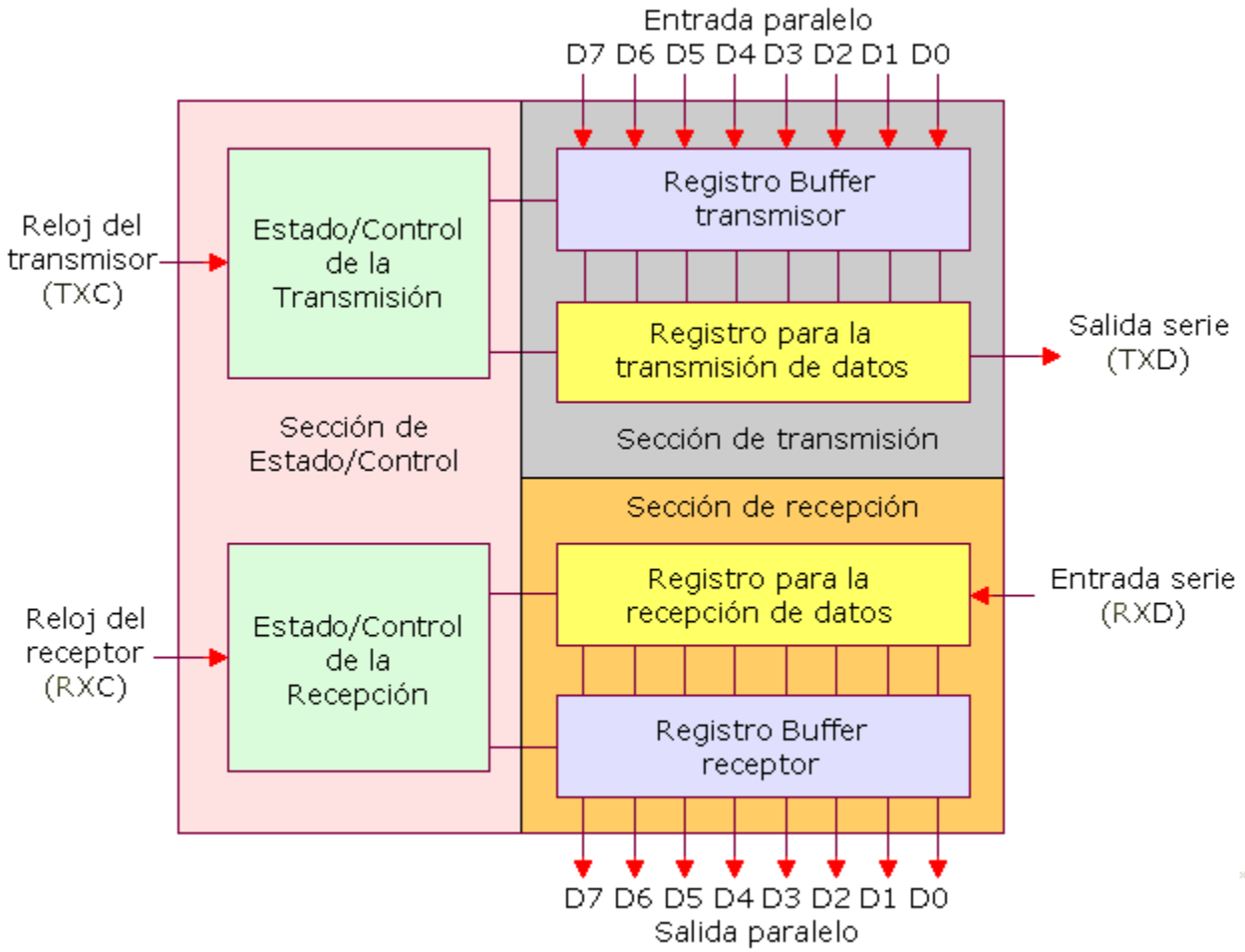
Conversión por hardware

Como puede ser apreciado en la conversión por software, el algorítmose se basa simplemente en operaciones de rotación o desplazamiento de registros. Los elementos que permiten la conversión por hardware se basan en registros de desplazamiento.

Existen tres tipos de dispositivos que permiten la conversión serie/paralelo:

a. El receptor/transmisor asíncrono universal (UART): usado en la transmisión serie asíncrona.
b. El receptor/transmisor síncrono universal (USRT): usado en la transmisión serie síncrona, usado en la transmisión a alta velocidad.
c. El receptor/transmisor síncrono/asíncrono universal (USART): permite la comunicación serie de forma tanto asíncrona como síncrona.

La estructura interna de estos dispositivos puede ser funcionalmente dividida en tres secciones: sección de transmisión, sección de recepción y sección de estado/control. Esta estructura se muestra en la siguiente figura.



Arquitectura interna de una UART

Los datos paralelos a ser convertidos por la sección de transmisión entran al registro buffer y son transferidos al registro para la transmisión de datos. Los datos son desplazados a la salida a través

de la línea de salida serie a una velocidad determinada por el reloj y la sección de control. Los bits de inicio, parada y bit de paridad serán añadidos automáticamente por la UART.

Los datos a ser convertidos en paralelo entrarán a través de la línea de entrada serie al registro para la recepción de datos. Los datos serie serán desplazados en este registro a una razón determinada por el reloj del receptor y la sección de control. Los bits de inicio y de parada serán discriminados de los datos asíncronos recibidos y la palabra recibida será transferida al registro buffer de salida paralelo.

La sección de estado/control no sólo controla la razón de transmisión/recepción, ésta genera interrupción, comprueba paridad, determina el número de bits de parada, comprueba el error de encuadre, etc.

Existen microcontroladores que incluyen internamente la circuiteria equivalente a una UART, lo que proporciona una mayor eficacia pues el programa se libera de la necesidad de inplementar la conversión serie/paralelo y la detección de errores.

También existen UART en circuitos integrados para ser enlazados a un microprocesador. Este dispositivo implementa la conversión asíncrona paralelo/serie para convertir el formato de palabra que maneja el microprocesador y el formato de datos usado en la transmisión serie.

A la UART se le deben especificar varios parámetros:

a. Bits de datos por caracter (usualmente de 5 a 8).
b. Bits de parada (1, 1,5 y 2).
c. Bit de paridad, para utilizar su capacidad de detección de error.
d. Velocidad de transmisión.

La UART comprueba de manera automática cuando debe ocurrir la sincronización del bit de parada. Si en este momento es detectado un 0, un bit de estado (error de encuadre o framming error) se activará. El sistema podrá leer el estado de este bit después de la lectura de cada caracter y determinar que hacer.

Otro tipo de error que comprueba el UART es el error de paridad, si es incluído en el protocolo de comunicación. Cuando se inserta la generación/comprobación de paridad la UART inserta (después de los bits de datos) un bit adicional, el cuál se obtiene realizándo una operación lógica OR exclusiva con los bits de datos que han sido transmitidos dará como resultado un 1 (paridad impar) o en 0 (paridad par). Por ejemplo, si es establecida la paridad impar el transmisor insertará un bit de paridad de forma tal que el número de unos de los bits de datos más el bit de paridad sea una cantidad siempre impar.

El error de sobreescritura (overwrite)

La UART presenta una estructura interna con doble buffer, ello le permite tener un caracter almacenado en el registro almacenador (buffer) de la recepción mientras que el registro de desplazamiento serie paralelo continúa ensamblando un nuevo caracter.

Cuando el registro almacenador posee un dato, se le indica al sistema que controla la UART, por ejemplo un microprocesador, que debe leer el caracter que ha ensamblado. Esto el microprocesador debe realizarlo antes de que el registro serie/paralelo ensamble un nuevo caracter (por ejemplo el tiempo que la UART en ensamblar un dato es de aproximadamente 1 mseg a 9600 baudios). Si el microprocesador no lee el dato antes de que se ensamble el próximo caracter, se perderá la información del caracter previo, pues el nuevo que ha sido enlazado ocupará su lugar en el registro buffer receptor. Cuando esto ocurre se produce un error de sobreescritura (overwrite), convenientemente indicado por el UART a través de un bit de la palabra de estado.

La sincronización de la recepción

Para realizar la sincronización del dato recibido se debe comprobar el bit en la mitad del intervalo del tiempo que dura para evitar la lectura de falsas transiciones producto del ruido en la línea. Para la sincronización se utiliza un reloj externo de período TC que cumple la relación:

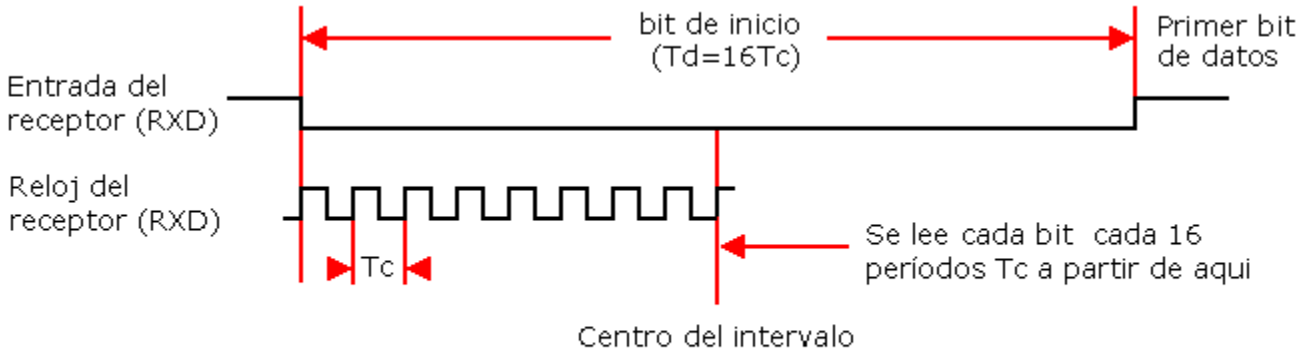
$Td = K * Tc$, donde K, generalmente, toma el valor 16.

Td es el tiempo de duración de cualquier bit de datos transmitidos, bit de paridad, bit de parada o bit de inicio.

Para lograr la sincronización entre el transmisor y el receptor tanto Tc como K deben ser el mismo para ambos, ello permitirá que el bit de datos se compruebe en el momento preciso sin necesidad de conectar una línea adicional de reloj para lograr el sincronismo.

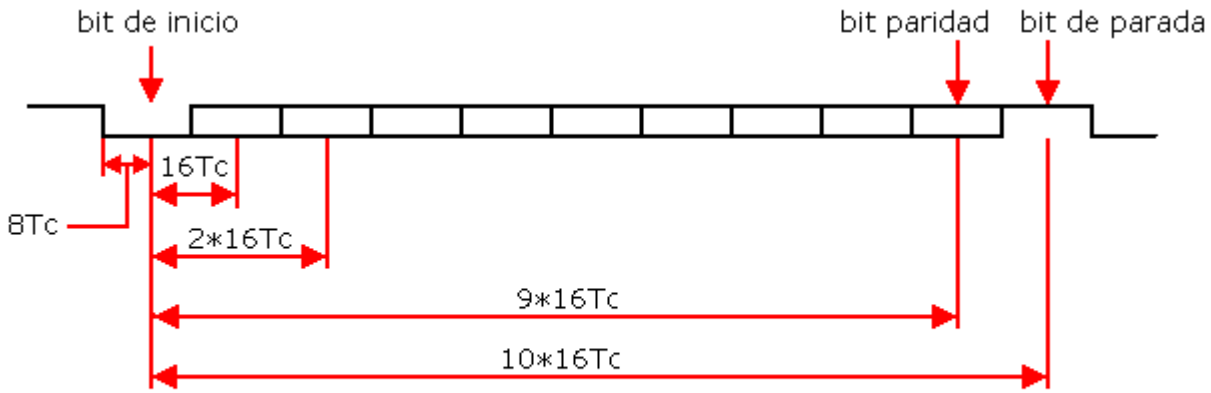
A continuación vamos a ver como se sincroniza el dato en una transmisión asíncrona.

En la figura siguiente se observa como después de detectado el bit de inicio y transcurridas 8 transiciones de reloj, ha transcurrido un tiempo igual a la mitad del bit de información que establece el inicio de la recepción de un nuevo caracter. A partir de ese tiempo se leerán los datos cada 16 pulsos de reloj.



Deteccion del bit de inicio.

Esto permite comprobar la información en la mitad del intervalo de cada bit de información.



Comprobación de la informacion en la mitad del intervalo de cada bit

La norma RS232

Como antes se adelantó, la norma RS232 es una de las más populares que se utilizan en la comunicación serie, y es la que se utiliza en los PC's, si bien hoy día está ampliamente superada por la transmisión serei a través de USB, de manera que está remitiendo su uso (por ejemplo, ya no se implementa en ordenadores portátiles). Se desarrolló en la década de los 60 para gobernar la interconexión de terminales y MODEM.

La norma RS232 resuelve tres aspectos en la comunicación que se establece entre el DTE, Equipo Terminal de Datos, por ejemplo un PC y el DCE, Equipo para la comunicacion de datos, por ejemplo un ratón:

1. Características eléctricas de la señal: Se establece que la longitud máxima entre el DTE y el DCE no debe ser superior a los 15 metros y la velocidad máxima de transmisión es de 20.000 bps. Los niveles lógicos no son compatibles TTL, considerando:
    a. 1 lógico entre -3V y -15V
    b. 0 lógico entre +3V y +15V
2. Características mecánicas de los conectores: Se utiliza un conector 25 patillas, DB 25, o de 9 patillas, DB 9, donde el conector macho identifica al DTE y el conector hembra al DCE.
3. Descripción funcional de las señales usadas: Las señales están básicamente divididas en dos grupos:
    a. Señales primarias, que son normalmente utilizadas para la transferencias de datos
    b. Señales secundarias, utilizadas para el control de la información que será transferida.

La norma RS232 está definida tanto para la transmisión síncrona como para la asíncrona, pero cuando se utiliza esta última, sólo un conjunto de terminales (de los 25), es utilizado.

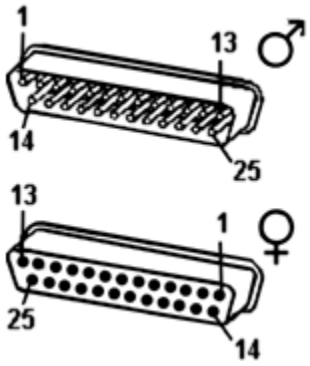
Velocidad

La velocidad está estandarizada según la norma RS 232C en baudios:

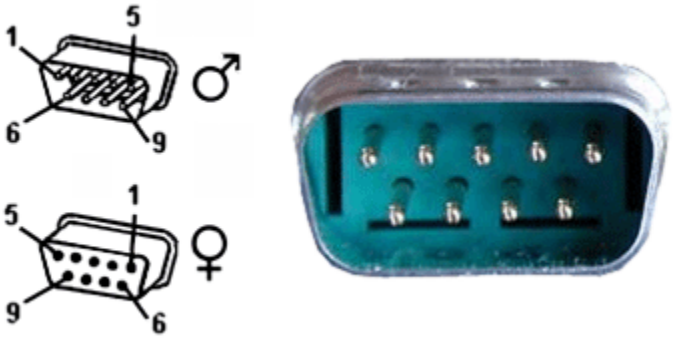
a. 75
b. 110
c. 150
d. 300
e. 600
f. 1200
g. 2400
h. 4800
i. 9600
j. 19200

Conectores

DB25 patillas macho y hembra



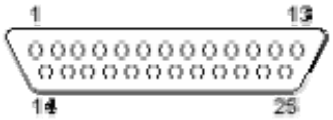
DB9 patillas4 macho y hembra



Ambos conectores son totalmente compatibles entre sí y existen adaptadores para pasar de un conector a otro

Descripción de terminales en RS232

Para ilustrar mejor el significado de cada terminal, consideremos a modo de ejemplo que el DTE podría ser un PC y el DCE un ratón. Se considerará el terminal DB25. Más adelante, en RS232 en el PC se volverá al tema de los conectores.



- TXD (Transmit Data, transmisión de datos, salida, pat. 2): Señales de datos que se transmiten del DTE al DCE. En principio, los datos no se pueden transmitir si alguno de los terminales RTS, CTS, DSR ó DTR está desactivado.
- RXD (Receive Data, recepción de datos, entrada, pat. 3): Señales de datos transmitidos desde el DCE al DTE.
- DTR (Data Terminal Ready, terminal de datos preparado, salida, pat. 20): Señal del DTE que indica que está conectado, generalmente en "0" indica que el DTE está listo para transmitir o recibir.
- DSR (Data Set Ready, dispositivo preparado, entrada, pat. 6): Señal del DCE que indica que el dispositivo está en modo de transmisión de datos.
- RTS (Request To Send, petición de envío, salida, pat. 4): Señal del DTE al DCE, notifica al DCE que el DTE dispone de datos para enviar. Se emplea en líneas semiduplex para controlar la dirección de transmisión. Una transición de 1 a 0 avisa al DCE que tome las medidas necesarias para prepararse para la transmisión.
- CTS (Clear To Send, preparado para transmitir, entrada, pat. 5): Señal del DCE al DTE indicando que puede transmitirle datos.
- CD (Carrier Detect, detección de portadora, entrada, pat. 8): Señal del DCE que ha detectado la señal portadora enviado por un modem remoto o que la línea telefónica está abierta.
- RI (Ring Indicator, timbre o indicador de llamada entrante, entrada, pat. 22): Señal del DCE indicando que está recibiendo una llamada por un canal conmutado.
- SG (GND) (System Ground ó Signal Ground, masa de señal, pat. 7): Masa común para todos las líneas.
- FG (GND) (Shield ó Protective Ground, tierra de protección, pat. 1): El conductor esta eléctricamente conectado al equipo.

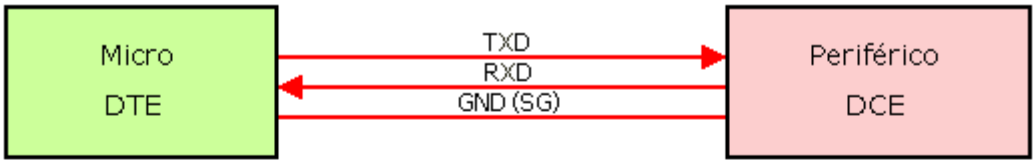
Una secuencia normal, a través de la RS232, es la siguiente:

1. Ambos dispositivos son alimentados, indicando encendido (si ha sido establecido en el equipo). El DTE activa el terminal DTR y el DCE activa el terminal DSR. Una interfase RS232 bien diseñada no comunicará hasta que estos dos terminales estén activos. El DTE esperará la activación del terminal DSR y el DTE la activación del terminal DTR. Aunque DTR y DSR algunas veces pueden ser utilizados para el control del flujo, estos terminales solo indican que los dispositivos están conectados.
2. El DTE pregunta al DCE si este está listo. El DTE activa la línea RTS. El DCE si está listo, responde activando la línea CTS. Puestos de acuerdo ambos equipos, se puede entrar a comunicar.

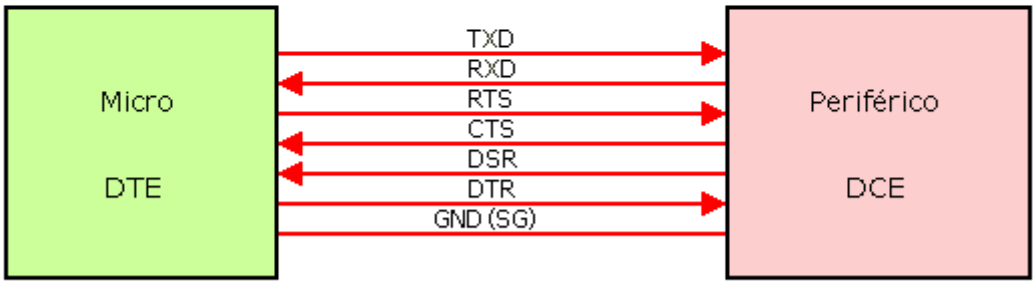
3. Los datos son transferidos en ambos sentidos. El DTE envía información al DCE a través del terminal TXD. El DCE envía información al DTE a través del terminal RXD.

Interfaz TTL-RS232

Para una comunicación full duplex desde la UART de un microprocesador o microcontrolador deben conectarse un mínimo número de señales, concretamente TXD y RXD así como la masa (GND, SG o Signal Ground). Sin embargo una interfaz típica RS232 requiere al menos 7 señales.



Requerimientos de señal para mínima conexion full duplex

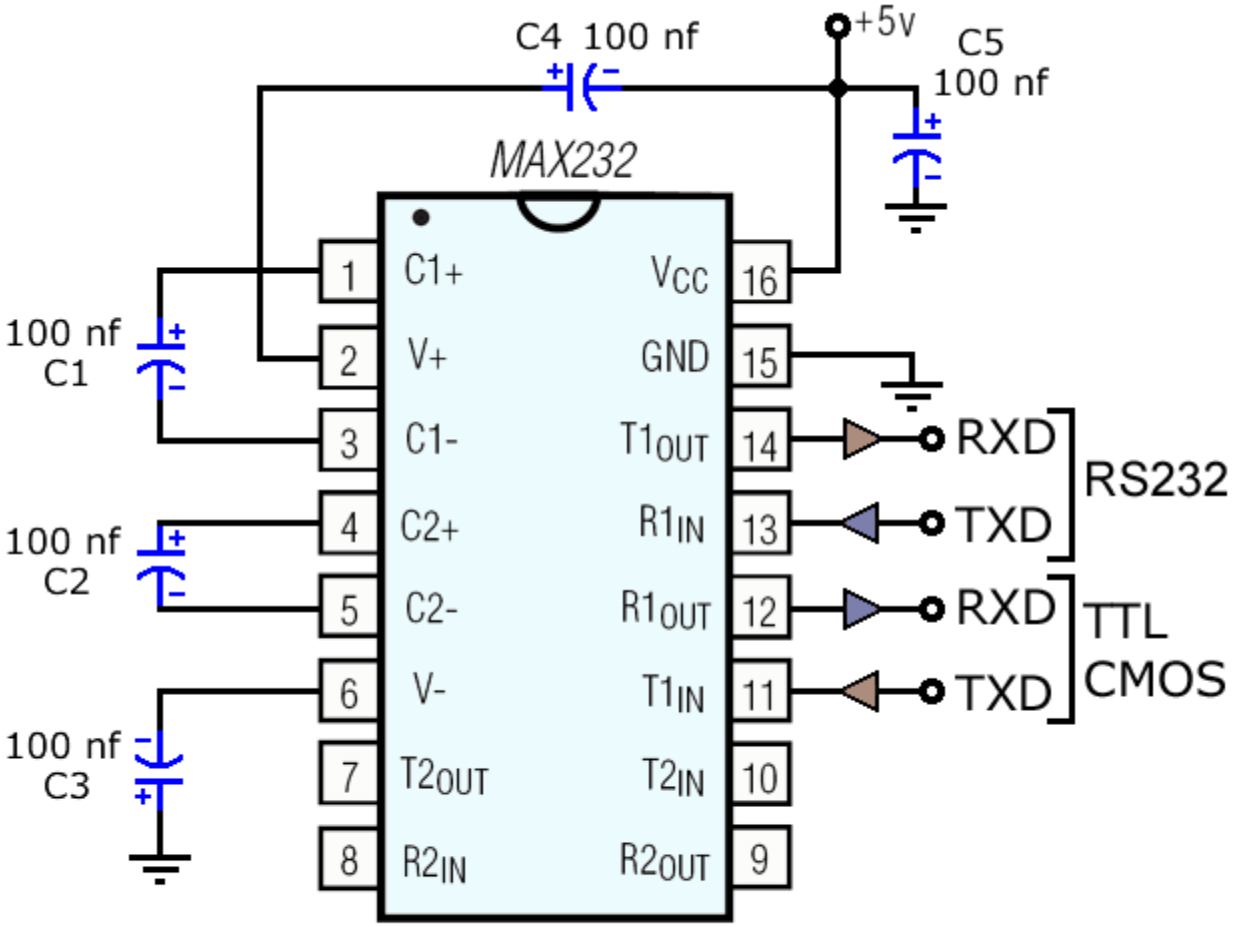


Requerimientos de señal típica para comunicacion full duplex

Las líneas adicionales se utilizan para la puesta de acuerdo entre el DTE (por ejemplo un PC) y el DCE (por ejemplo un ratón).

El terminal para transmitir datos (TXD) es utilizado para transferir datos del DTE al DCE, por lo que debe ser conectado a la línea receptora serie del periférico. De manera idéntica la línea receptora de datos (RXD) debe ser conectada a la línea transmisora del periférico.

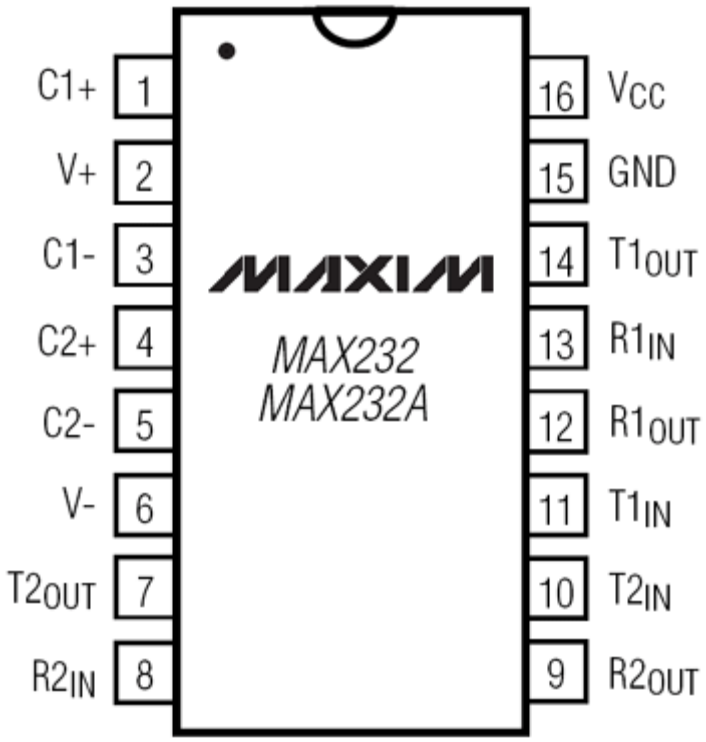
Para convertir TTL a RS232 se pueden usar circuítos típicos de transistores y diodos discretos o los circuitos integrados MC1488 y MC1489, sin embargo, existe un circuíto integrado muy popular que permiten esta conversión. El MAX232 es un conversor de nivel TTL/RS232. Sólo es necesario este circuito integrado y 4 condensadores. La interfaz mínima con el MAX232 entre un dispositivo con salida serie TTL o CMOS y el conector RS232 se muestra en la siguiente figura.



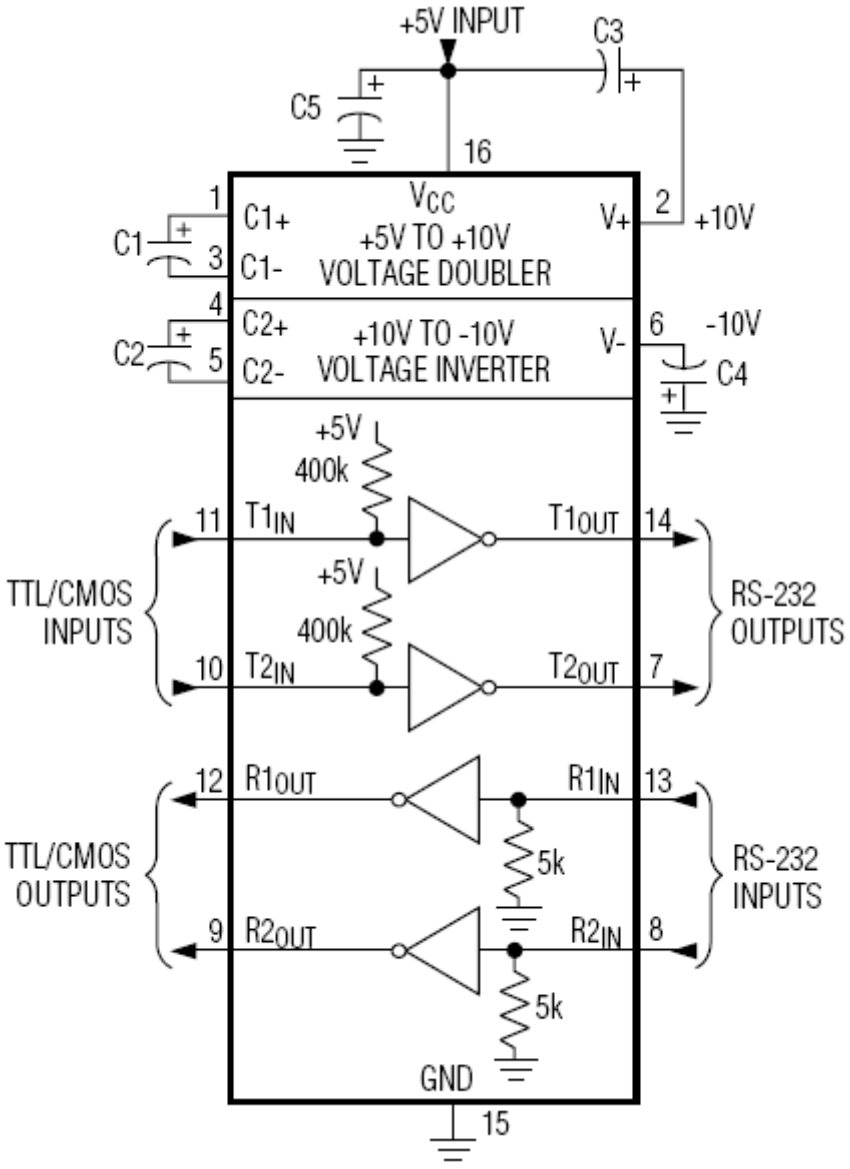
Conexión mínima RS232/TTL con un MAX232. Con los condensadores de 100 nF se puede llegar hasta los 64 Kbps, si se colocan de 1 µF se puede llegar hasta 120 Kbps

El MAX232

Descripción: El MAX232 dispone internamente de 4 conversores de niveles TTL al estandar RS232 y viceversa, para comunicación serie como los usados en los ordenadores, el COM1 y el COM2. Puede encontrar mas información en MAX232.PDF.



Funcionamiento: El circuito integrado lleva internamente 2 conversores de nivel de TTL a RS232 y otros 2 de RS232 a TTL con lo que en total podremos manejar 4 señales del puerto serie del PC, por lo general las mas usadas son; TXD, RXD, RTS, CTS, estas dos últimas son las usadas para el protocolo handshaking pero no es imprescindible su uso. Para que el MAX232 funcione correctamente debemos poner unos condensadores externos, todo esto lo podemos ver en la siguiente figura en la que solo se han cableado las lineas TXD y RXD que son las mas usualmente usadas para casi cualquier aplicacion.



En el MAX232 todos los condensadores deben ser de 1 microfaradio para llegar hasta 120 Kbps o de 100 nanofaradios para llegar hasta 64 Kbps. Para el MAX232A los condensadores han de ser de 100 nanofaradios y se consiguen hasta 200 Kbps.

Usos: Este integrado es usado para comunicar un microcontrolador o sistema digital con un PC o sistema basado en el estándar RS232.

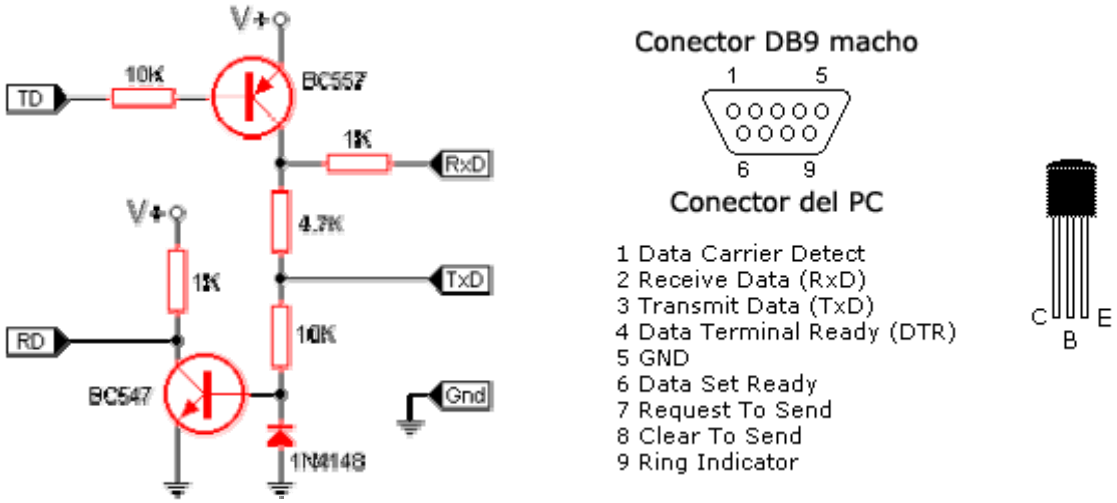
Características a +5v, condensadores de 100 nF:

- Vcc: de 4,5v a 5,5v.
- Consumo: 4 mA (15 mA con carga a la salida de 3 Kohm).
- Entradas compatibles TTL y CMOS.
- Tensión de entrada máxima RS232: +/- 30v.
- Tensión de Salida RS232: +/- 15v.

- Tensión de salida típica de +/-8v con carga nominal de 5 Kohm en RS232.
- Resistencia entrada RS232: 5 Kohm (a masa).
- Resistencia entrada TTL/CMOS: 400 Kohm (a positivo).
- Las entradas se pueden dejar al aire.
  - Entrada TTL al aire, se considera un "0" al invertirse en la salida.
  - Entrada RS232 al aire, se considera un "1" al invertirse en la salida.
- Salidas cortocircuitables contínuamente:
  - Salida RS232: +/- 22 mA.
  - Salida TTL/CMOS: a masa -10 mA, a positivo +30 mA.
- Data Rate: 200 Kbps (mín 116 Kbps).

Interfaz TTL-RS232 sin MAX232

Hemos visto que para conseguir un interfaz TTL-RS232, utilizar el MAX232 es lo más sencillo pues además del integrado sólo se necesitan cinco condensadores. No obstante, si se va a utilizar con un PC, se puede realizar un interfaz mediante componentes discretos, 5 resistencias, 2 transistores y 1 diodo. El circuito aprovecha la propia corriente del puerto COM del PC para generar las señales del RS232.



Los terminales marcados como TxD, RxD y Gnd corresponden al conector RS232 del PC mientras que los terminales marcados como RD y TD van directamente a sistema con las señales TTL.

Este tipo de interfaz puede verse en ratones o elementos de control de puntero del PC. Los puntos de alimentación son de +5V.

RS232 en el PC

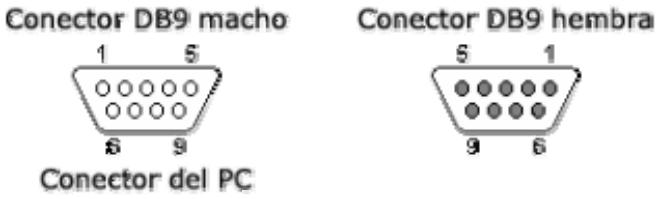
El puerto serie de un ordenador trabaja en modo asíncronico. En puerto serie recibe y envía información fuera del ordenador mediante un determinado software de comunicación o un driver del puerto serie. La información se envía al puerto carácter a carácter. Cuando se ha recibido un

carácter, el puerto serie envía una señal por medio de una interrupción indicando que el carácter está listo. Cuando el ordenador ve la señal, los servicios del puerto serie leen el carácter.

Existen dos tipos de interfaces RS232 puesto que la norma fue diseñada para dos tipos de equipos, el DTE (Equipo Terminal de Datos) y el DCE (Equipo de Comunicación de Datos). Existen entonces dos tipos de interfaz RS232, la DTE (conector macho) y la DCE (conector hembra):

- Interfaz DTE (macho) en el PC.
- Interfaz DCE (hembra) en los modem, ratones y otros dispositivos.

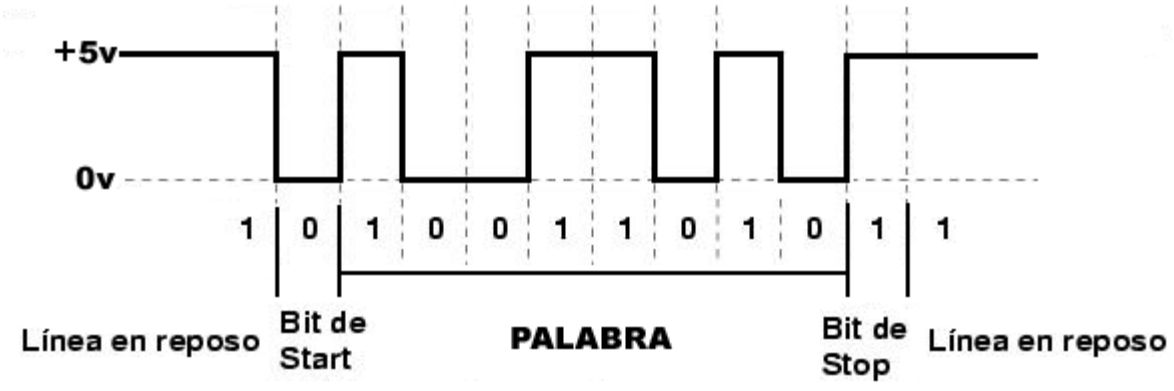
Por tanto en un PC se utilizan conectores DB9 macho, de 9 patillas, por los que se conectan los dispositivos al puerto serie. Los conectores hembra que se enchufan tienen una colocación de patillas diferente, de manera que se conectan la patilla 1 del macho con la patilla 1 del hembra, la patilla 2 con el 2, etc...



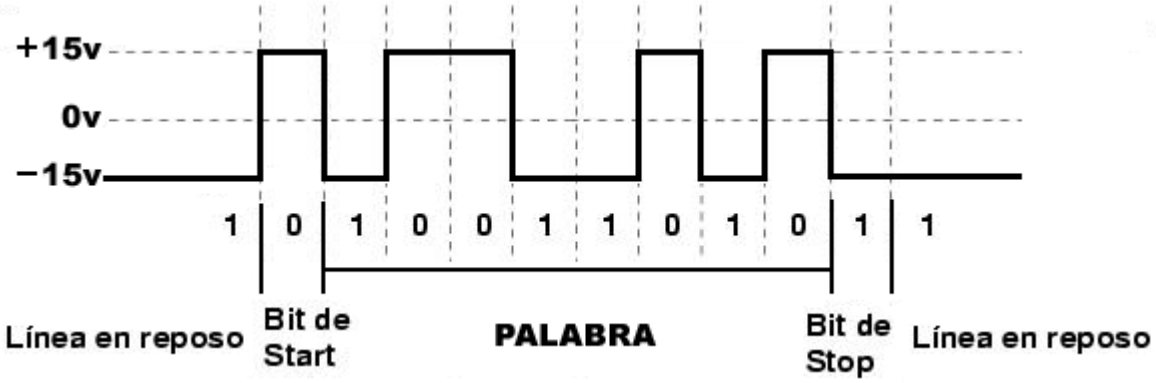
RS232 no admite comunicaciones a más de 15 metros y 20 Kbps (se puede utilizar mayor distancia y velocidad, pero no es el estándar). La comunicación es efectuada con 25 terminales diferentes, cada uno con su función. RS232 está definida tanto para la comunicación síncrona como asíncrona, pero cuando se utiliza esta última sólo se utiliza un conjunto de los 25 terminales.

Normalmente, las comunicaciones serie en el PC tienen los siguientes parámetros: 9.600 baudios, 1 bit de Start, 8 bits de Datos, 1 bit de Stop y sin paridad.

En la figura siguiente se puede ver un ejemplo de la transmisión en TTL del dato binario 01011001. La línea en reposo está a nivel lógico alto (+5 voltios).



En la figura siguiente se puede ver un ejemplo de la transmisión en RS232 del dato binario 01011001. La línea en reposo está a nivel lógico alto (-15 voltios).



Direcciones e IRQ de los puertos serie

El puerto serie utiliza direcciones I/O y una interrupción para llamar la atención del procesador. Además el software de control debe conocer la dirección.

La mayoría de los puertos series utilizan direcciones estandar predefinidas. Éstas están descritas normalmente en base hexadecimal. Las direcciones I/O e IRQ pueden seleccionarse en la BIOS o bajo Windows.

Las señales son:

Puerto	Dir. I/O	IRQ
COM1	3F8-3FF	4
COM2	2F8-2FF	3
COM3	3E8-3EF	4
COM4	2E8-2EF	3

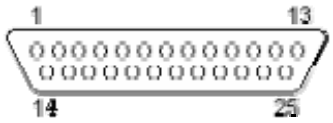
Las direcciones e IRQ usadas por los puertos serie fueron definidas al diseñar el PC, sin embargo, las del COM3 y COM4 no se han definido oficialmente, aunque están aceptadas por convenios.

El IBM-PC utilizaba la UART 8250, siendo la 16550A una de las últimas que se utilizan.

Conector Serie DB25

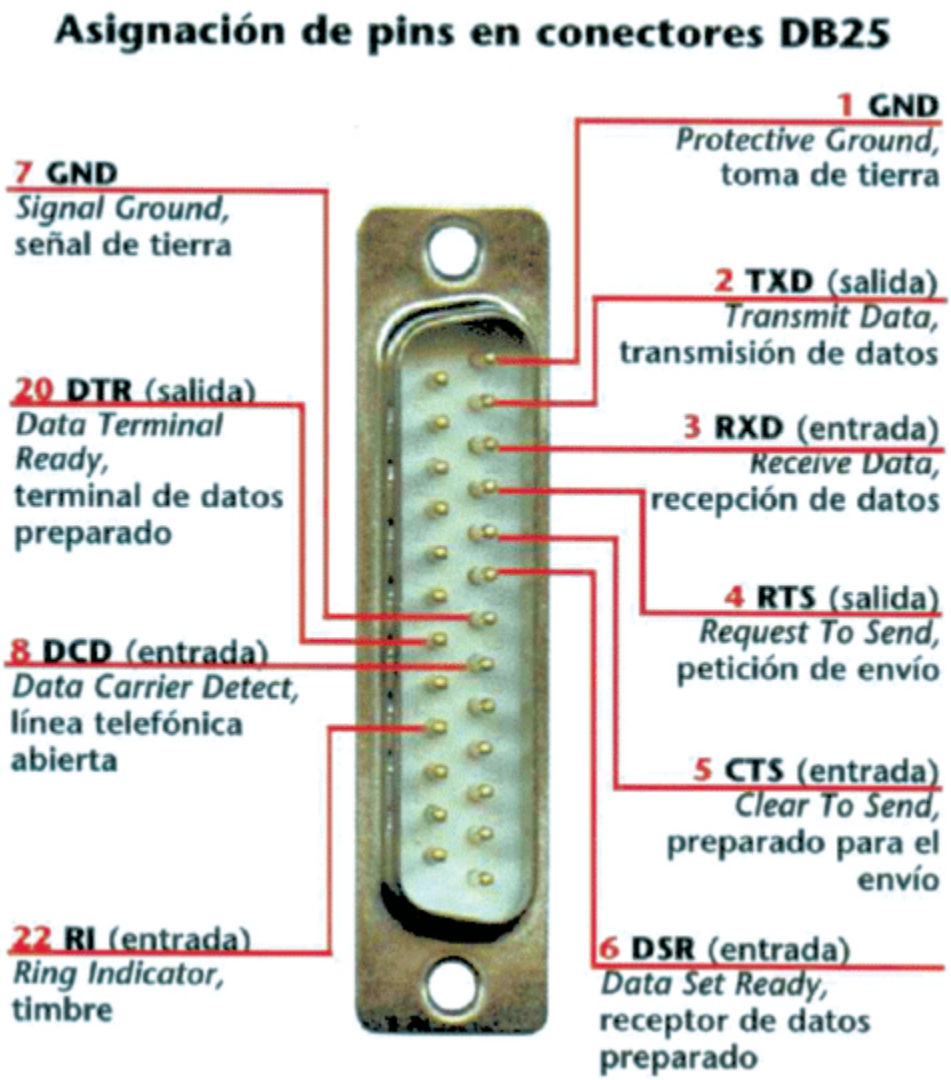
Pat.	Nombre	Dir	Descripción	Visto del lado PC (DB25 Macho)

1	FG (GND)	-	Shield Ground, tierra de protección
2	TXD	→	Transmit Data, transmisión de datos
3	RXD	←	Receive Data, recepción de datos
4	RTS	→	Request to Send, petición de envío
5	CTS	←	Clear to Send, preparado para transmitir
6	DSR	←	Data Set Ready, dispositivo preparado
7	GND	-	System Ground ó Signal Ground, tierra de señal
8	CD	←	Carrier Detect, detección de portadora
9 al 19	n/c	-	
20	DTR	→	Data Terminal Ready, terminal de datos preparado
21	n/c	-	
22	RI	←	Ring Indicator, indicador de llamada entrante
23 al	n/c	-	



La dirección (Dir) es DTE (PC) relativa a DCE (Dispositivo).

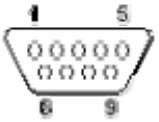
- DTE (PC) ← DCE (Dispositivo), entrada en el DTE (PC).
- DTE (PC) → DCE (Dispositivo), salida en el DTE (PC).



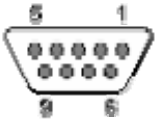
Conector Serie DB9

Pat.	Nombre	RS232	V.24	Dir	Descripción
1	CD	CF	109	←	Carrier Detect, detección de portadora
2	RXD	BB	104	←	Receive Data, recepción de datos
3	TXD	BA	103	→	Transmit Data, transmisión de datos
4	DTR	CD	108.2	→	Data Terminal Ready, terminal de datos preparado

Conector DB9 macho
Conector del PC



Conector DB9 hembra

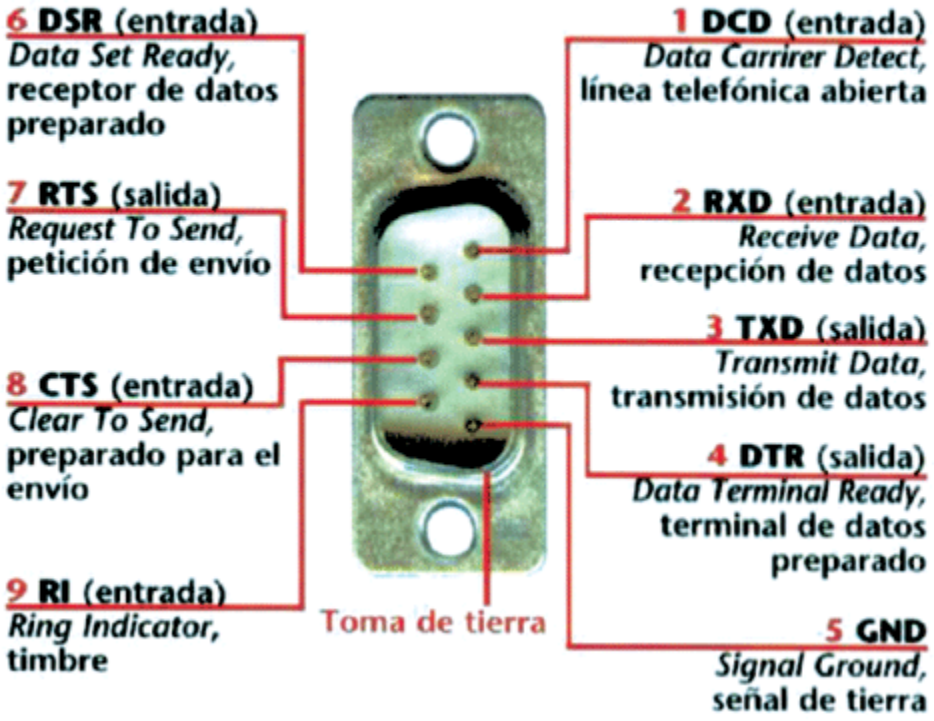


5	GND	AB	102	—	System Ground ó Signal Ground, tierra de señal
6	DSR	CC	107	←	Data Set Ready, dispositivo preparado
7	RTS	CA	105	→	Request to Send, petición de envío
8	CTS	CB	106	←	Clear to Send, preparado para transmitir
9	RI	CE	125	←	Ring Indicator, indicador de llamada entrante

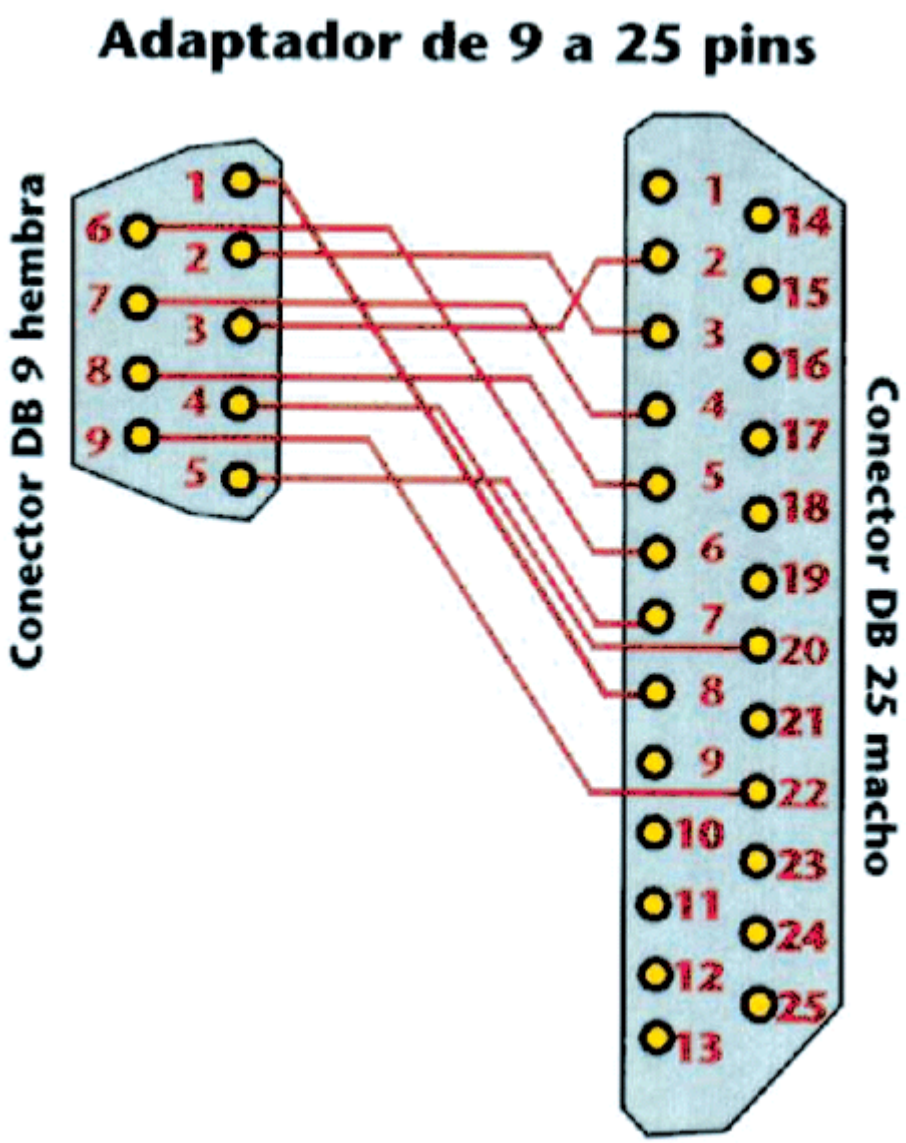
La dirección (Dir) es DTE (PC) relativa a DCE (Dispositivo).

- DTE (PC) ← DCE (Dispositivo), entrada en el DTE (PC).
- DTE (PC) → DCE (Dispositivo), salida en el DTE (PC).

Asignación de pins en conectores DB9



Adaptador de 9 a 25 patillas



Existen dispositivos compactos capaces de adaptar un conector a otro

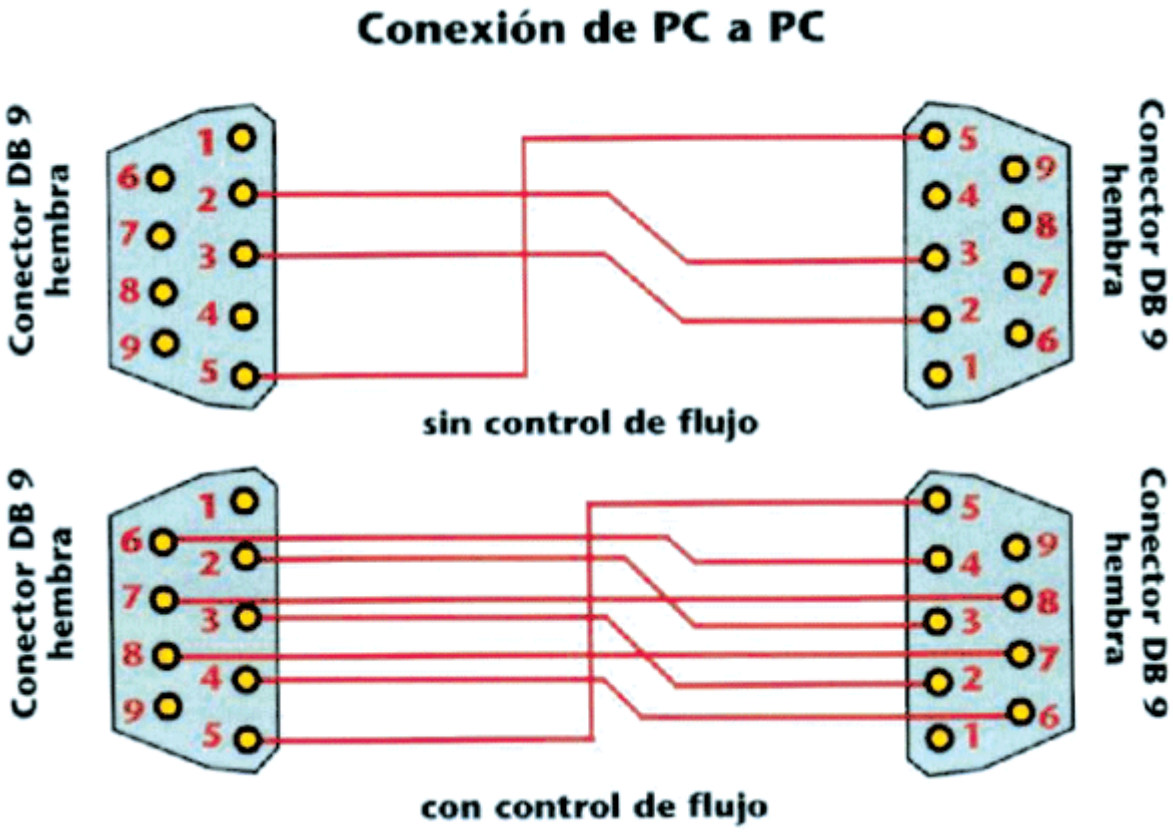
Tipos de conexiones con DB9

Dos PC´s no se puede conectar de manera directa entre sí, pues son dos DTE, pero no obstante se puede hacer de acuerdo a la forma de conectar el cable.

Para conectar dos DTE hay que tener en cuenta que ambos transmiten por la línea 2 y reciben por la línea 3, por ello, basta cruzar RXD (2) y TXD (3). También debe conectarse la línea de tierra de señal. Esta conexión es válida cuando el software que controla la comunicación no utiliza los terminales de control.

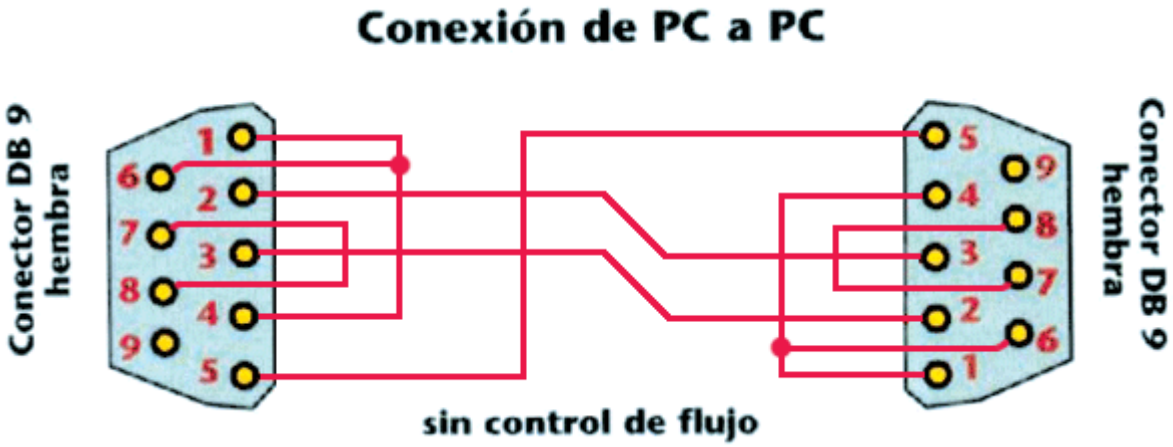
Si es necesario utilizar los terminales "en línea" (DSR y DTR) se debe considerar que ambos DTE activarán el terminal DTR (4) y esperarán por la activación del terminal DSR (6). Como ninguno activará el terminal DSR, estarán esperando siempre. Este problema se puede solucionar mediante el intercambio de las señales de control, basta cruzar los terminales DSR (6) y DTR (4)

Con respecto a los terminales RTS (7) y CTS (8) sucede algo similar a DSR y RTS, por ello se pueden cruzar los terminales 7 y 8.

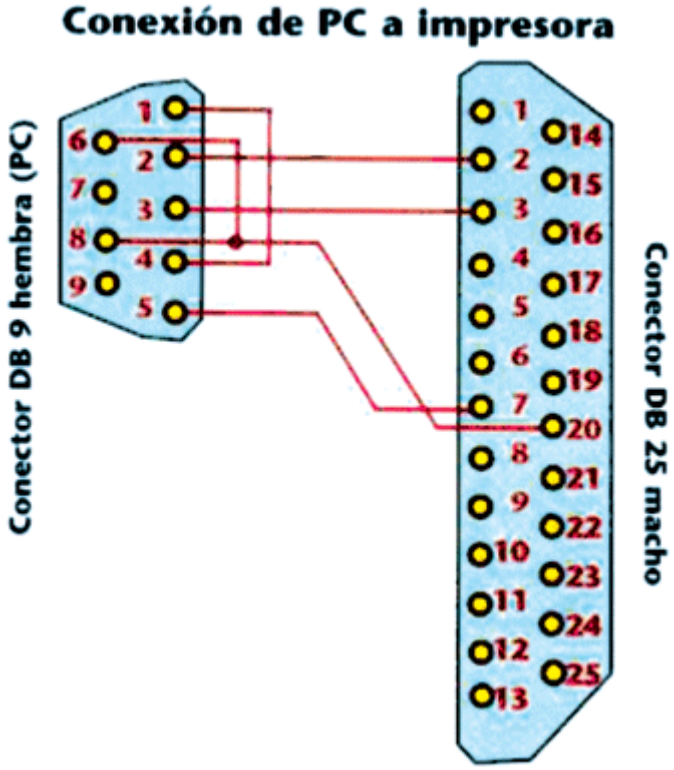


Otra forma de conexión, en este caso sin control de flujo, se haría considerando que como cada DTE espera la activación del terminal DSR al mismo tiempo que activa el DTR, se unan en cada DTE, para que cada DTE se de a sí mismo la posibilidad de transmisión. Lo mismo se haría con RTS y CTS. También se conectará el terminal CD a DTR. Algunos programas no trabajan si este terminal

no está activo. De manera que como CD es entrada en ambos DTE, se debe mantener activo conectándolo a DTR.



Conexión del PC a una impresora serie



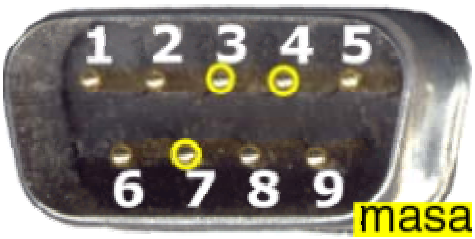
Configuración de los puertos

- Bit por segundo:
  - Define la velocidad máxima, en bits por segundo (bps), a la que se transmiten los datos a través del puerto. Normalmente, se establece a la velocidad máxima admitida por el equipo o dispositivo con el que se está comunicando.
- Bits de datos:
  - Cambia el número de bits de datos a utilizar para cada carácter transmitido y recibido. El equipo o dispositivo con el que comunica debe tener la misma configuración que aquí. La mayor parte de los caracteres se transmiten con siete u ocho bits de datos.
- Paridad:
  - Cambia el tipo de comprobación de errores a utilizar para el puerto seleccionado. El equipo o dispositivo con el que se comunica debe tener la misma configuración que aquí. Se debe elegir una de las siguientes:
    - Ninguna: significa que no se agregará ningún bit de paridad a los bits de datos enviados desde este puerto. Esto deshabilitará la comprobación de errores.
    - Par: significa que el bit de paridad se establece a 1 si se necesita para que el número de unos (1) de los bits de datos sea par. Esto habilitará la comprobación de errores.
    - Impar: significa que se agrega un bit de paridad si se necesita para que el número de unos (1) de los bits de datos sea impar. Esto habilitará la comprobación de errores.
    - Marca: significa que se agrega un bit de paridad, pero siempre está establecido a 0.

        - Espacio: significa que se agrega un bit de paridad, pero siempre está establecido a 1.
- Bit de parada:
    - Cambia el tiempo entre cada carácter que se transmite (cuando el tiempo se mide en bits por segundo).
- Control de flujo:
    - Cambia la forma en que se controla el flujo de datos.
        - Ninguno
        - Xon/Xoff, llamado en ocasiones protocolo de enlace software, es el método de software estándar para controlar el flujo de datos entre dos módems.
        - Control de flujo Hardware, llamado en ocasiones protocolo de enlace hardware, es el método estándar de controlar el flujo de datos entre un equipo y un dispositivo serie.

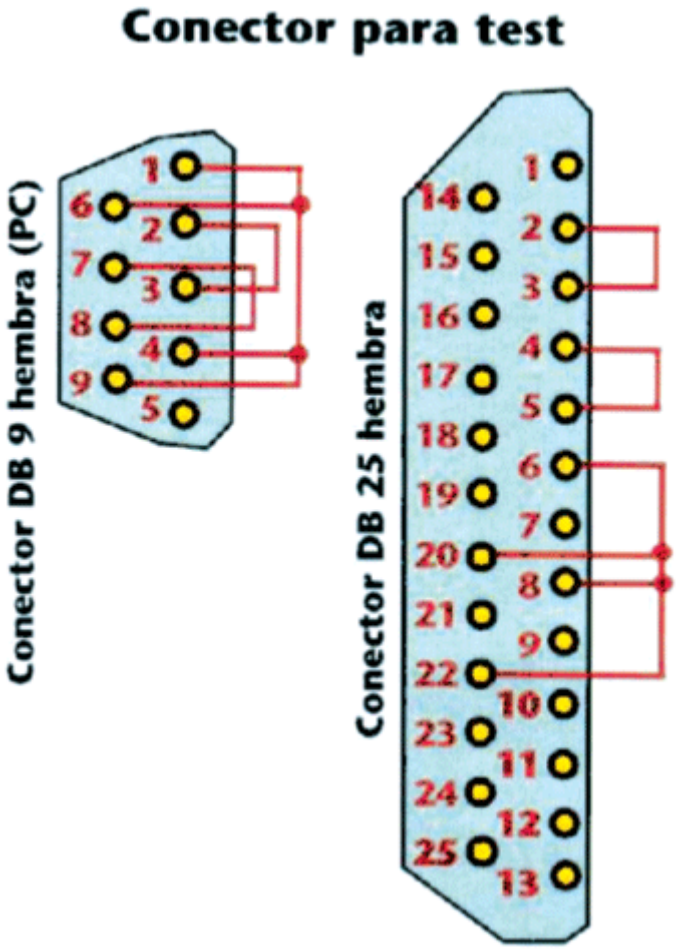
Comprobación de los puertos serie

Con un voltímetro en tensión continua colocar una sonda en la patilla 3 y la otra sonda a masa. El valor que tiene que dar es de unos 11V, hacer lo mismo con la patilla 4 y la patilla 7. Si no da 11V algo va mal en el ordenador.



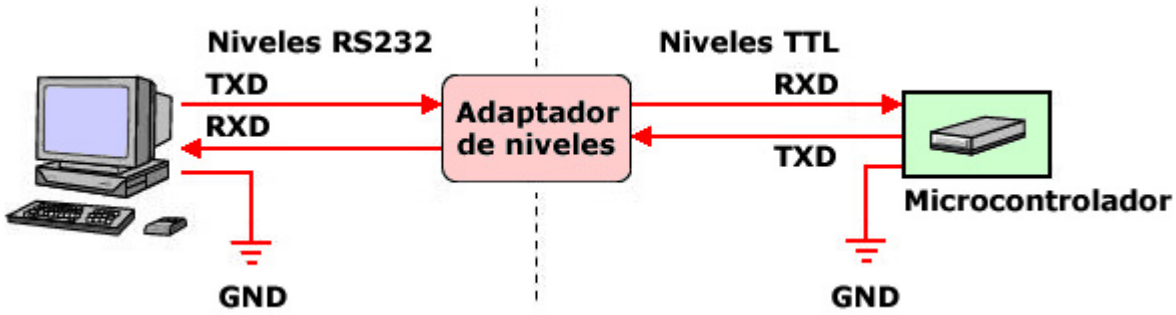
Si colocamos la sonda negra del polímetro en la masa del conector, la tensión de las patillas 3, 4 y 7 será negativa (-11,54v). En el resto de patillas la tensión es aproximadamente de 0 (0,16v)

También podemos utilizar un programa de diagnóstico como CheckIt para lo cual hemos de realizar las siguientes conexiones.

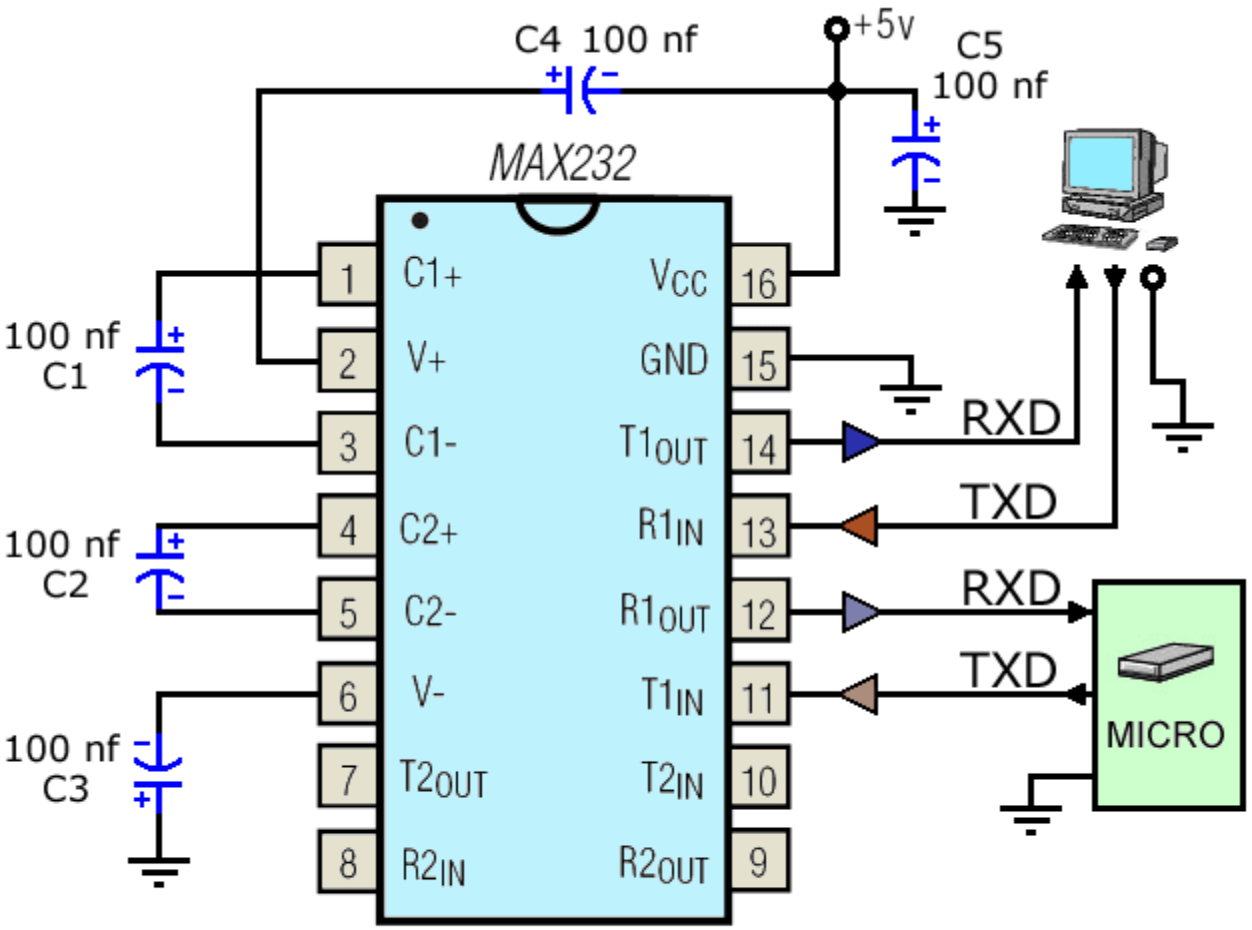


Conexión de un microcontrolador al puerto serie del PC

Para conectar el PC a un microcontrolador por el puerto serie se utilizan las señales TXD, RXD y GND. El PC utiliza la norma RS232, por lo que los niveles de tensión de las patillas entán comprendidos entre +15 y -15 voltios. Los microcontroladores normalmente trabajan con niveles TTL (0-5v). Es necesario por tanto intercalar un circuito que adapte los niveles:



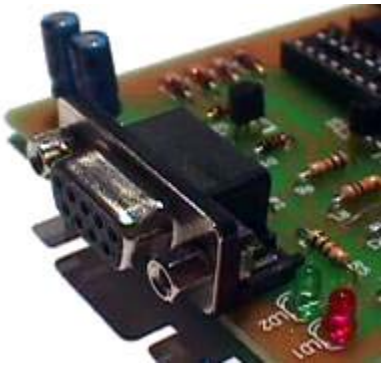
Uno de estos circuitos, que se utiliza mucho, es el MAX232.



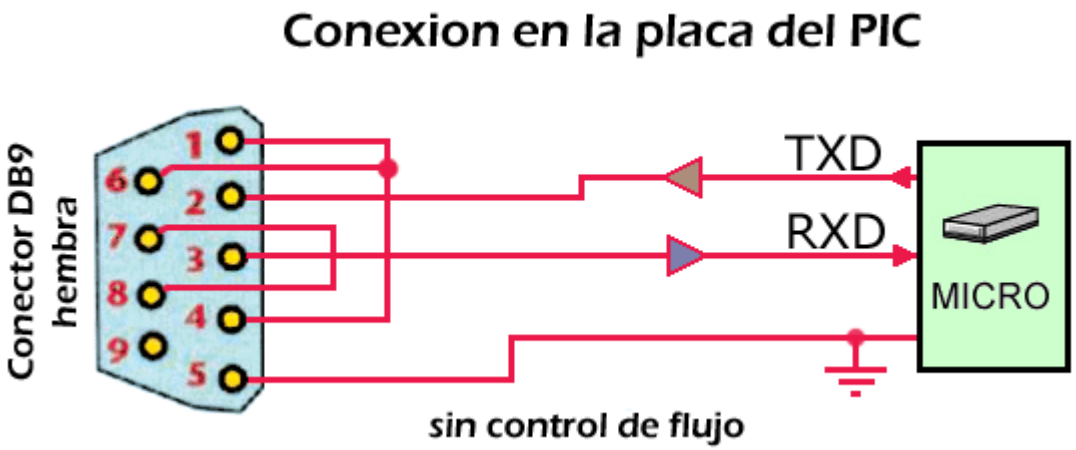
Conexión de un microcontrolador con un PC mediante MAX232. Con los condensadores de 100 nF se puede llegar hasta los 64 Kbps, si se colocan condensadores de 1 µF se puede llegar hasta 120 Kbps

Cable de conexión

Para realizar la conexión entre el PC y un microcontrolador circuito podemos usar diferentes alternativas. Una manera es utilizar un cable serie macho-hembra no cruzado, y en el circuito un conector hembra DB9 para circuito impreso.



En la placa de circuito impreso donde se encuentra el PIC y donde se colocará el conector DB9 hembra sería conveniente realizar la interconexión entre patillas que se describe en la siguiente figura.



Las conesxiones que presenta la figura garantizan que cualquier programa de comunicación acepte la transmisión del PIC, si bien se realizará sin control de flujo. La salida DTR (patilla 4, Terminal de Datos Preparado) entrega señal a la entrada DCD (patilla 1, Detección de Portadora) y a la entrada DSR (patilla 6, Dispositivo Preparado). Por otro lado la salida RTS (patilla 7, Petición de Envío), entraga señal a la entrada CTS (patilla 8, Preparado para el Envío).

Esta configuración no es necesaria ni para Hyperterminal de Windows ni para TerminalTOB.

USB

Los ordenadores personales actuales aún conservan prácticamente todos los puertos heredados desde que se diseñó el primer PC de IBM. Por razones de compatibilidad aún seguiremos viendo este tipo de puertos, pero poco a poco irán apareciendo nuevas máquinas en las que no contaremos con los típicos conectores serie, paralelo, teclado, etc... y en su lugar sólo encontraremos puertos USB o Fireware.

Conectores como el de la salida paralelo (o Centronics), la salida serie (RS232) o el conector del teclado han sufrido muy pocas variaciones.

Si bien es cierto que estos conectores todavía hoy cumplen su función correctamente en casos como la conexión de un teclado o un ratón, se han quedado ya desfasados cuando tratamos de conectar dispositivos más rápidos como por ejemplo una cámara de video digital.

USB (Bus Serie Universal) nace como un estandar de entrada/salida de velocidad media-alta que permite conectar dispositivos que hasta ahora requerían de una tarjeta especial para sacarles todo el rendimiento, lo que ocasionaba un encarecimiento del producto además de ser productos propietarios ya que obligaban a adquirir una tarjeta para cada dispositivo.

Pero además, USB nos proporciona un único conector para solventar casi todos los problemas de comunicación con el exterior, pudiendose formar una auténtica red de periféricos de hasta 127 elementos.

Mediante un par de conectores USB que ya hoy en día son estandar en todas las placas base, y en el espacio que hoy ocupa un sólo conector serie de 9 pines nos va a permitir conectar todos los dispositivos que tengamos, desde el teclado al modem, pasando por ratones, impresoras, altavoces, monitores, scaners, camaras digitales, de video, plotters, etc... sin necesidad de que nuestro PC disponga de un conector dedicado para cada uno de estos elementos, permitiendo ahorrar espacio y dinero.

Al igual que las tarjeta ISA desaparecieron, todos los conectores anteriormente citados también desaparecerán de nuestro ordenador, eliminando además la necesidad de contar en la placa base o en una tarjeta de expansión los correspondientes controladores para dispositivos serie, paralelo, ratón PS/2, joystick, etc...

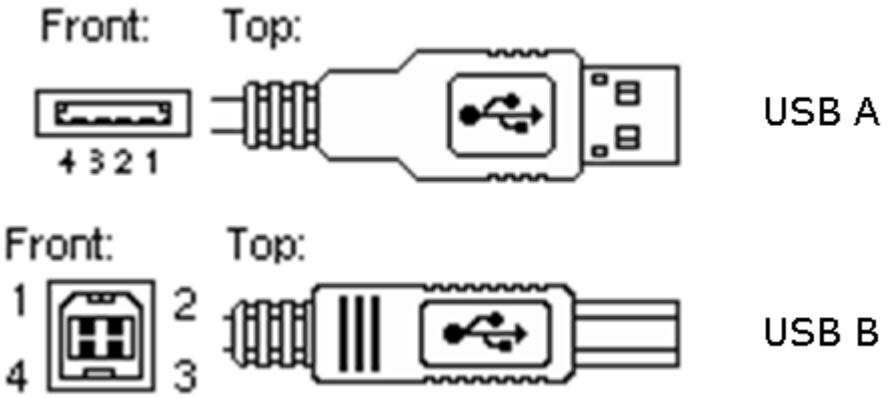
USB es PnP (Plug and Play) y permite la conexión "en caliente", es decir, que se pueden conectar y desconectar los periféricos sin necesidad de reiniciar el ordenador.

Características de USB

- Cable de 4 hilos.
  - 2 de alimentación para dispositivos max 0,5A.
  - 2 de transmisión diferencial.
- Permite suministrar energía electrica a dispositivos que no tengan un alto consumo y que no estén a más de 5 metros, lo que elimina la necesidad de conectar dichos periféricos a la red eléctrica, con sus correspondientes fuentes de alimentación, como ocurre por ejemplo con los modems externos.
- Segmentos de cable de 5m max.
- Dos tipos de conectores.
- Hasta 127 dispositivos.
- Conexión/desconexión en caliente.
- Auténtico Plug & Play.
- Muchos dispositivos pueden funcionar en PC y MAC.
- Transmisión de datos entre PC´s.

- Si trabajamos bajo Windows necesitaremos como mínimo la versión OSR 2.1 del Windows 95 para que reconozca los dispositivos.
- Soportado por Win98 mediante drivers.
- Nativo en Windows XP.
- Requiere una sóla IRQ para todos los dispositivos.
- Topología en estrella, lo que implica la necesidad de dispositivos tipo "hub" que centralicen las conexiones, aunque en algunos dispositivos como teclados y monitores ya se implementa esta característica, lo que permite tener un sólo conector al PC, y desde estos dispositivos sacar conexiones adicionales.
- Por ejemplo en los teclados USB se suele implementar una conexión adicional para el ratón, o incluso otras para joystick, etc.. y en los monitores varias salidas para el modem, los altavoces...
- HUB USB externo de dos salidas. Posibilidad de encadenar varios HUB
- HUB interno para una bahía de 3,5" de 4 salidas. Posibilidad de encadenar varios HUB
- Velocidad USB
    - Velocidad baja 1,5Mb/s (192KB/s)
    - Velocidad alta 12Mb/s (1,5MB/s)
- Velocidad USB2
    - 480 Mbps (34 MBps)
    - Compatible con USB 1.1. Utiliza los cables USB existentes
- Periféricos:
    - Ratón, módem, joystick, teclado, altavoces, escáner, impresoras, digitalizadoras de vídeo etc.
    - Están apareciendo placas base con puertos USB para dispositivos internos
    - Adaptadores PCMCIA - USB para portatiles
    - Conversor USB a puerto serie
    - Con USB2
        - Disco duros externos, grabadoras, videocamaras

Conectores



Los conectores tipo "A" se utilizan en el PC y los tipos "B" suelen utilizarse en los dispositivos USB (también existe otro conector mas pequeño).

Pat.	Nombre	Descripción
1	VBUS	+5 VDC
2	D-	Data -
3	D+	Data +
4	GND	Ground