

## **CAPÍTULO 3**

### **LA COMUNICACIÓN SERIAL**

#### **3.1 Introducción.**

La historia de las comunicaciones se remonta a 1810 cuando un señor alemán de apellido Von Soemmering, utilizó 26 cables (1 por cada letra del alfabeto) pegados a la parte inferior de un acuario. Cuando se pasaba corriente por los cables, se producían burbujas en el agua. De esta forma, se podían enviar mensajes codificados por medio de burbujas. Los militares vieron este ingenio y se inició la carrera para desarrollar los sistemas de comunicaciones, [16].

Luego, el señor Morse inventó su dispositivo que enviaba sonidos cortos y largos, que representaban los caracteres. En el código Morse, los sonidos cortos se llaman ESPACIOS y los sonidos largos se denominan MARCAS. Por lo tanto, esto es un código binario, porque usa solo 2 valores posibles: Espacio y Marca. En términos de computadoras se diría que trabaja con los números 0 y 1, dependiendo de cada bit.

Cuando se habla de código Morse, hablamos de comunicación SERIAL. Se llama serial, porque los bits se reciben uno detrás de otro o “en serie”. La comunicación RS-232 también es de tipo serial, ya que los bits vienen uno detrás de otro.

Se utilizan dos computadoras para realizar la interfase, una que simule el medidor de nivel y la otra la interfase con la que trabajará el usuario, es decir, una PC mandará los mismos datos que transmitiría el dispositivo medidor de nivel y la otra PC los recibiría mostrándolos al usuario, y por lo tanto enviaría una respuesta.

Una de las maneras para realizar la comunicación entre dos PC's, es utilizando un cable serial, conectado a los puertos seriales de ambas computadoras. La comunicación serial utiliza el protocolo RS-232 y es el más común de los métodos de comunicaciones.

En un extremo del cable de comunicaciones se encuentra un conector llamado DB9 hembra y del otro extremo un conector DB9 macho. Se llama así porque tiene la forma de una D. Y tiene el número 9 porque tiene 9 patitas o 9 entradas (Figura 3.1).

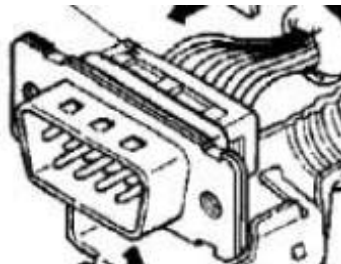


Figura 3.1 Conector DB9.

La otra versión de conector que se usa para comunicaciones seriales RS232 es el conector DB25.

### 3.2 Principio de Funcionamiento.

El puerto serial es un dispositivo muy extendido y ya sean uno o dos puertos, con conector grande o pequeño, todos los equipos PC lo incorporan actualmente. Debido a que el estándar del puerto serial se mantiene desde hace muchos años, la institución de normalización americana (EIA) ha escrito la norma **RS-232-C** que regula el protocolo de la transmisión de datos, el cableado, las señales eléctricas y los conectores en los que debe basarse una conexión RS-232.

La comunicación realizada con el puerto serial es una comunicación asíncrona. Para la sincronización de una comunicación se precisa siempre de un bit adicional a través del cual el emisor y el receptor intercambian la señal del pulso. Pero en la transmisión serial a través de un cable de dos líneas esto no es posible ya que ambas están ocupadas por los datos y la tierra. Por este motivo se intercalan antes y después de los datos de información de estado según el protocolo RS-232. Esta información es determinada por el emisor y el receptor al estructurar la conexión mediante la correspondiente programación de sus puertos seriales. Esta información puede ser la siguiente:

- i. **Bit de inicio.-** cuando el receptor detecta el bit de inicio sabe que la transmisión ha comenzado y es a partir de entonces que debe leer la transmisión y entonces debe leer las señales de la línea a distancias concretas de tiempo, en función de la velocidad determinada.

- ii. **Bit de parada.-** indica la finalización de la transmisión de una palabra de datos. El protocolo de transmisión de datos permite 1, 1.5 y 2 bits de parada.
- iii. **Bit de paridad.-** con este bit se pueden descubrir errores en la transmisión. Se puede dar paridad par o impar. En la paridad par, por ejemplo, la palabra de datos a transmitir se completa con el bit de paridad de manera que el número de bits 1 enviados es par.

### 3.3 El protocolo RS-232.

Consiste en un conector tipo DB-25 de 25 pines, aunque es normal encontrar la versión de 9 pines DB-9, más barato e incluso más extendido para cierto tipo de periféricos (como el ratón serie del PC). En cualquier caso, los PCs no suelen emplear más de 9 pines en el conector DB-25. Las señales con las que trabaja este puerto serial son digitales, de +12V (0 lógico) y -12V (1 lógico), para la entrada y salida de datos, y a la inversa en las señales de control. El estado de reposo en la entrada y salida de datos es -12V. Dependiendo de la velocidad de transmisión empleada, es posible tener cables de hasta 15 metros. Cada pin puede ser de entrada o de salida, teniendo una función específica cada uno de ellos. Las más importantes se observan en las tablas 3.1 y 3.2.

Las señales TXD, DTR y RTS son de salida, mientras que RXD, DSR, CTS son de entrada. La tierra de referencia para todas las señales es SG (Tierra de Señal). Finalmente, existen otras señales como RI (Timbre Telefónico).

#	Pin	E/S	Función	Conector DB 9
1			Tierra de Chasis	
2	RXD	E	Recibir Datos	
3	TXD	S	Transmitir Datos	
4	DTR	S	Terminal de Datos Listo	
5	SG		Tierra de señal	
6	DSR	E	Equipo de Datos Listo	
7	RTS	S	Solicitud de Envío	
8	CTS	E	Libre para Envío	
9	RI	S	Timbre Telefónico	

Tabla 3.1 Pines más importantes del conector DB9.

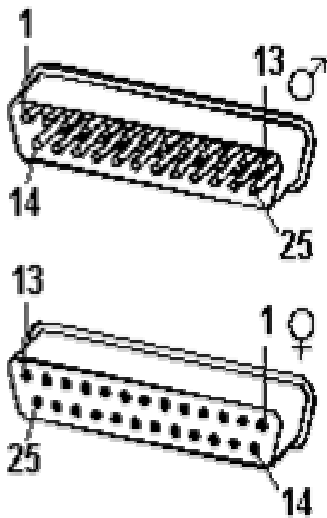
#	Pin	E/S	Función	Conector DB 25
1			Tierra de Chasis	
2	TXD	S	Transmitir Datos	
3	RXD	E	Recibir Datos	
4	RTS	S	Solicitud de Envío	
5	CTS	E	Libre para Envío	
6	DSR	E	Equipo de Datos Listo	
7	SG		Tierra de señal	
8	CD/DCD	E	Detector de Portadora	
15	TxC	S	Transmitir Reloj	
17	RxC	E	Recibir reloj	
20	DTR	S	Terminal de Datos Listo	
22	RI	S	Timbre Telefónico	
24	RTxC	S/E	Transmitir/Recibir Reloj	

Tabla 3.2 Pines más importantes del conector DB25.

### 3.4 Uso de las Señales RS232.

**Tierra de Chasis.-** Se conecta internamente al chasis del dispositivo.

**Recibe Datos (RXD).-** Por esta pata entran los datos del dispositivo externo.

**Transmite Datos (TXD).-** Por esta pata salen los datos para ir al dispositivo externo.

**Terminal de Datos Listo (DTR).-** Esta pata realiza el control maestro del dispositivo externo. Cuando este pin esta en 1, el dispositivo externo no transmite ni recibe datos.

**Tierra de señal.-** Se ha dicho que los datos se envían como voltajes + o -. Pues bien, esta pata es la referencia de señal para esos voltajes.

**Conjunto de Datos Listo (DSR).-** Por lo general, los dispositivos externos tienen esta patita con un valor permanente de 0.

**Solicita Permiso para Enviar Datos (RTS).-** Esta parte del hardware se usa para “estrecharse la mano” entre los dispositivos que se están comunicando. Si el dispositivo externo está de acuerdo, pone un 0 en la patita que se llama Pista Libre para Enviar Datos (CTS).

**Pista Libre para Enviar Datos (CTS).-** Ésta es la otra mitad del hardware usado para “estrecharse la mano”. El dispositivo externo pone esta pata en 0 cuando está lista para recibir datos.

**Timbre Telefónico (RI).-** Esta patita se usa sólo cuando un PLC (Programable Logical Controller) esta conectado a un módem.

### 3.4.1 Diagrama de Conexión.

Para hacer posible la comunicación entre dos equipos PC se han interconectado las terminales descritas anteriormente. La conexión ha sido realizada de la siguiente manera (Figura 3.2):

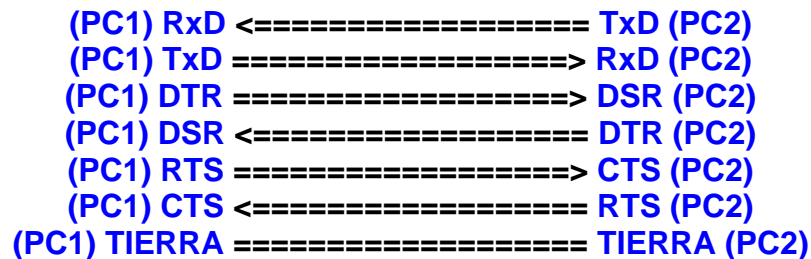


Figura 3.2 Diagrama a Flujos de una Comunicación serial.

### 3.5 Velocidad y Control del Puerto Serial.

El ordenador controla el puerto serial mediante un circuito integrado específico, llamado UART (Transmisor-Receptor-Asíncrono Universal). Normalmente se utilizan los siguientes modelos de este chip: 8250 (bastante antiguo, con fallos, solo llega a 9600 baudios), 16450 (versión corregida del 8250, llega hasta 115200 baudios) y 16550A (con buffers de E/S). A partir de la gama Pentium, la circuitería UART de las placas base son todas de alta velocidad, es decir UART 16550A. De hecho, la mayoría de los módems conectables a puerto serial necesitan dicho tipo de UART, incluso algunos juegos para jugar en red a través del puerto serial necesitan de este tipo de puerto serial. Por eso hay veces que un 486 no se comunica con la suficiente velocidad con un PC Pentium. Los portátiles suelen llevar otros chips: 82510 (con buffer especial, emula al 16450) o el 8251 (no es compatible).

Para controlar al puerto serial, el CPU emplea direcciones de puertos de E/S y líneas de interrupción (IRQ). En el AT-286 se eligieron las direcciones 3F8h (o 0x3f8) e IRQ 4 para el COM1, y 2F8h e IRQ 3 para el COM2. El estándar del PC llega hasta aquí, por lo que al añadir posteriormente otros puertos seriales, se eligieron las direcciones 3E8 y 2E8 para COM3-COM4, pero las IRQ no están especificadas. Cada usuario debe elegir las de acuerdo a las que tenga libres o el uso que vaya a hacer de los puertos seriales (por ejemplo, no importa compartir una misma IRQ en dos puertos siempre que no se usen conjuntamente, ya que en caso contrario puede haber problemas). Es por ello que últimamente, con el auge de las comunicaciones, los fabricantes de PCs incluyen un puerto especial PS/2 para el ratón, dejando así libre un puerto serial.

Mediante los puertos de E/S se pueden intercambiar datos, mientras que las IRQ producen una interrupción para indicar al CPU que ha ocurrido un evento (por ejemplo, que ha llegado un dato, o que ha cambiado el estado de algunas señales de entrada). El CPU debe responder a estas interrupciones lo mas rápido posible, para que de tiempo a recoger el dato antes de que el siguiente lo sobrescriba. Sin embargo, las UART 16550A incluyen unos buffers de tipo FIFO, dos de 16 bytes (para recepción y transmisión), donde se pueden guardar varios datos antes de que el CPU los recoja. Esto también disminuye el número de interrupciones por segundo generadas por el puerto serial.



El RS-232 puede transmitir los datos en grupos de 5, 6, 7 u 8 bits, a unas velocidades determinadas (normalmente, 9600 bits por segundo o más). Después de la transmisión de los datos, le sigue un bit opcional de paridad (indica si el número de bits transmitidos es par o impar, para detectar fallos), y después 1 o 2 bits de Stop. Normalmente, el protocolo utilizado es 8N1 (que significa, 8 bits de datos, sin paridad y con 1 bit de Stop).

Una vez que ha comenzado la transmisión de un dato, los bits tienen que llegar uno detrás de otro a una velocidad constante y en determinados instantes de tiempo, por eso se dice que el RS-232 es asíncrono. Los pines que portan los datos son RXD y TXD. Las demás se encargan de otros trabajos: DTR indica que el ordenador esta encendido, DSR que el aparato conectado a dicho puerto está encendido, RTS que el ordenador puede recibir datos (porque no está ocupado), CTS que el aparato conectado puede recibir datos, y DCD detecta que existe una comunicación, presencia de datos.

Tanto el aparato a conectar como el ordenador (o el programa terminal) tienen que usar el mismo protocolo serial para comunicarse entre sí. Puesto que el estándar RS-232 no permite indicar en que modo se está trabajando, es el usuario quien tiene que decidirlo y configurar ambas partes. Como ya se ha visto, los parámetros que hay que configurar son: protocolo serial (8N1), velocidad del puerto serial, y protocolo de control de flujo. Este último puede ser por hardware (el que ya se ha visto, el handshaking RTS/CTS) o bien por software (XON/XOFF,

el cual no es muy recomendable ya que no se pueden realizar transferencias binarias). La velocidad del puerto serial no tiene por que ser la misma que la de transmisión de los datos, de hecho debe ser superior. Por ejemplo, para transmisiones de 1200 baudios es recomendable usar 9600, y para 9600 baudios se pueden usar 38400 (o 19200).

### **3.6 Convertidor USB a Serial.**

El material disponible para realizar la comunicación serial no tuvo ningún problema de forma parcial, los cables se consiguieron con una gran facilidad, pero sólo una de las PC's contaba con el puerto serial, ya que la tecnología sigue su rumbo y los dispositivos para el envío y recepción de datos han evolucionado, como por ejemplo, al puerto USB que maneja protocolos diferentes, así como su dispositivo de entrada.

Por lo tanto, a falta de un puerto serial en una computadora, volteamos a buscar nuevas alternativas para formar la comunicación, se encontró un convertidor USB a serial como se muestra en la Figura 3.3, el cual hacía un buen manejo de los protocolos para crear la compatibilidad.

El convertidor transforma el dispositivo de conexión USB al conector serial macho DB9, proporcionando dos puertos seriales a la computadora, pero sólo funciona uno a la vez, este dispositivo es muy usado en las nuevas laptop's que

no tienen un puerto serial, pero se quiere conectar un joystick, un scanner, una cámara digital, etc.



Figura 3.3 Dispositivo convertidor USB-Serial.