

Universidad Politécnica de Cartagena



**Escuela Técnica Superior de Ingeniería de
Telecomunicación**

FUNDAMENTOS DE TELEMÁTICA

**Práctica 1: Comunicaciones serie asíncronas. Nivel
físico y de enlace.**

Profesores:

Elena Pérez Andrés.
Felipe García Sánchez.
Antonio Javier García Sánchez.

ÍNDICE

1. Objetivos de la práctica.....	3
2. Elementos que intervienen en la práctica.....	3
3. Nivel físico.....	4
3.1 El interfaz RS-232.....	4
4. Control de flujo.....	8
5. Null-Modem (supresor de modem).....	8
6. Software de comunicaciones.....	10
6.1 X-MODEM.....	10
7. Desarrollo de la Práctica.....	14
7.1 Inicialización y configuración.....	14
7.2 Estudio del nivel físico.....	14
7.2.1 Conexionado.....	14
7.2.2 Forma de onda.....	14
7.2.3 Entramado y sincronización.....	15
7.3 Estudio del control de flujo.....	15
7.3.1 Control de flujo software.....	15
7.3.2 Control de flujo hardware.....	15
7.4 Control de dispositivos usando el puerto serie.....	16
7.4.1 Determinación del cable a usar.....	16
7.4.2 Configuración del dispositivo.....	16
7.4.3 Comunicación y control del dispositivo.....	16
7.5 Estudio del nivel de enlace.....	16
7.5.1 Estudio del protocolo X-MODEM.....	17
Anexo 1: Terminal de Windows.....	18

1. Objetivos de la práctica

- Analizar los distintos aspectos telemáticos que intervienen en una comunicación punto-punto entre dos equipos terminales de datos (ETD) a través del interfaz RS-232.
 - Comprender las comunicaciones asíncronas. Entramado.
 - Conocer las distintas líneas del interfaz RS-232.
 - Conocer el código de línea.
 - Mostrar la necesidad del control de flujo.
 - Conocer el equipo de terminal de datos VT-100.
 - Analizar los distintos mecanismos de control de flujo.
 - Comprender los mecanismos de control de errores.
 - Manejar un programa de comunicaciones del entorno PC. Transmisión de ficheros.

2. Elementos que intervienen en la práctica

- 2 PCs que actúan como DTE's con entorno Windows interconectados mediante los respectivos puertos serie.
- Un pod de RS-232, intercalado en paralelo en la línea que une los dos DTE's.
- Un PC conectado al pod RS-232 que actúa como analizador de puerto serie.
- 1 osciloscopio digital.
- Programa de gestión del puerto serie (**Terminal** de Windows).
- Varios ficheros de pruebas.

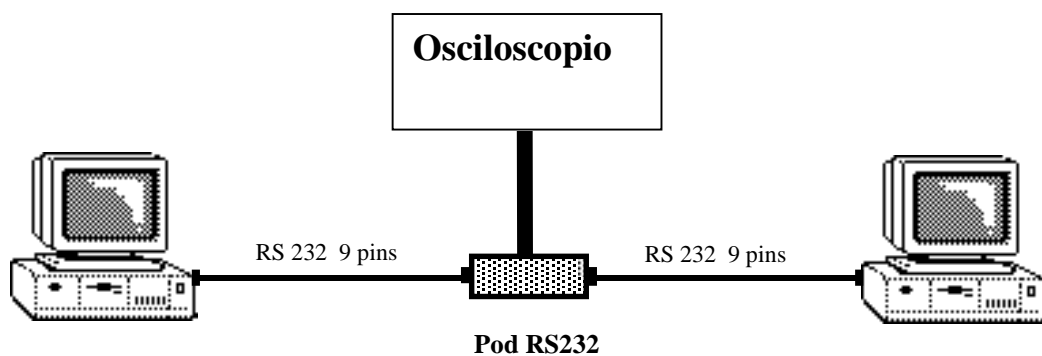


Figura 1.1. Configuración del puesto de trabajo.

3. Nivel físico

El nivel físico es el nivel más bajo de la arquitectura de protocolos ISO/OSI. Tiene por misión la transmisión de un flujo de bits entre dos entidades sin interpretar su contenido, siendo esta misión responsabilidad de los niveles superiores.

Los protocolos de nivel físico describen cuatro tipos de características: eléctricas, mecánicas, de procedimiento y funcionales.

Las características eléctricas definen niveles lógicos y de tensión, tiempos, compatibilidad eléctrica entre interfaces y velocidades de transmisión. Las mecánicas, el dimensionado del interfaz, el tipo de conector y el número de contactos.

Las especificaciones funcionales definen el modo de operación de los circuitos clasificándolos en circuitos de datos, control, temporización y masas. Finalmente, las especificaciones de procedimiento indican las secuencias de control y datos para establecer, mantener y liberar una comunicación.

3.1 El interfaz RS-232

Uno de los interfaces de nivel físico más utilizados es el RS232 que fue estandarizado por EIA en 1962. El interfaz consta de 25 contactos formando circuitos que trabajan en modo no balanceado, manejando todos ellos una masa común. La transmisión de información se realiza en serie tanto en modo asíncrono como síncrono.

Las especificaciones eléctricas definen con un margen de tensión entre -5 y -15 voltios el nivel lógico **1** ó **marca** para los circuitos de datos, y **OFF** para las líneas de control. El rango de tensión entre +5 y +15 voltios define el nivel lógico **0** ó **espacio** para los circuitos de datos y **ON** para los circuitos de control. El tiempo de tránsito entre estados tiene que ser menor que 1 mseg. En estado de reposo (**Idle**) la línea permanece en el estado lógico **1**.

Las especificaciones mecánicas se muestran en la figura 1.2 y definen el conector de 25 patillas DB-25.

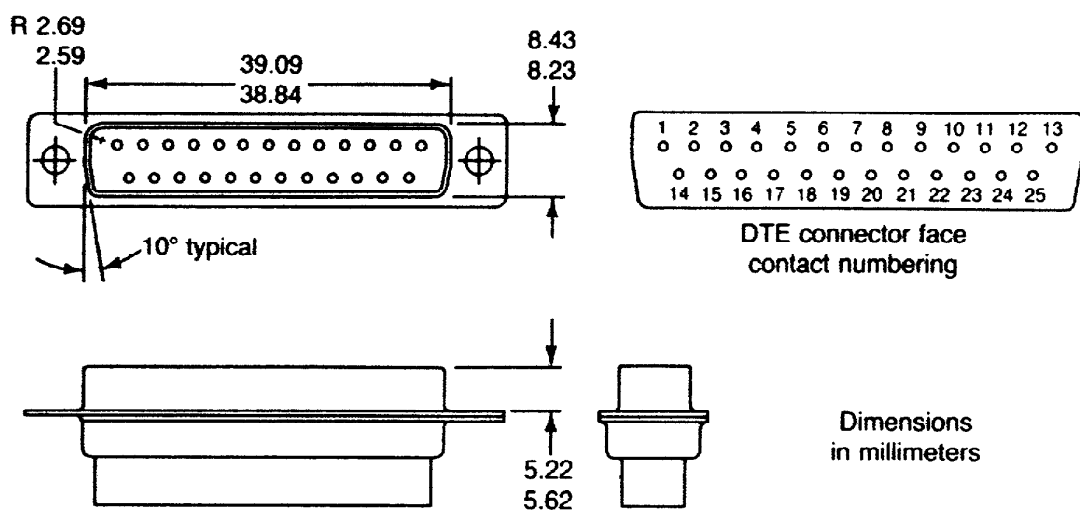


Figura 1.2 Especificaciones mecánicas del interfaz RS-232

Con estas especificaciones se puede establecer una comunicación entre dispositivos a una distancia máxima de 15 metros a 20 Kbps.

En 1987 EIA creó el estándar RS-232-D que concuerda con las normas V-24/V-28 del CCITT. Este interfaz proviene de RS-232-C añadiendo bucles de comprobación locales y remotos, nuevos circuitos de comprobación y definición del tipo de cable.

Número del circuito de enlace	Pin	Nombre de la señal	Del/hacia el ETCD	Tipo señal	DESCRIPCIÓN
102	7	signal ground	-	Tierra	Tierra de señalización o retorno común
103	2	TXD	hacia	Datos	Transmisión de datos de terminal local hacia el módem remoto
104	3	RXD	del	Datos	Recepción de los datos demodulados desde el terminal remoto
105	4	RTS	hacia	Control	Petición de transmisión
106	5	CTS	del	Control	Preparado para transmitir
107	6	DSR	del	Control	Aparato de datos preparado. Indica que el módem está preparado para recibir o transmitir datos
108.1	20	DTR	hacia	Control	Conectar el módem a la línea
108.2	20	DTR	hacia	Control	Terminal de datos preparado. El terminal está listo para transmitir y recibir datos
109	8	DCD	del	Control	Detector de señales de línea recibidas por el canal de datos. El módem genera una señal hacia el terminal de datos cuando detecta un nivel de portadora aceptable.
110	13	Signal Quality	del	Control	Detector de la calidad de la señal de datos. Cuando la señal es ON la calidad de la señal es aceptable. OFF baja calidad
111	23	Data Signal Rate	hacia	Control	Selector de velocidad binaria del equipo terminal de datos.
112	23		del	Control	Selector de velocidad binaria del módem.
113	24	DAA	hacia	Tempori.	Temporización para los elementos de señal en la transmisión. Señal de temporización generada por un elemento externo al módem.
114	15	DB	del	Tempori.	Temporización para los elementos de señal en la transmisión. Señal de temporización generada por el módem para sincronizar los datos a transmitir TXD.
116	14	Select Standby	hacia	Control	Conmutación de seguridad en modo directo. OFF línea punto-punto. ON línea RTC.
117	16	Standby Indicator	del	Control	OFF línea punto-punto. ON línea RTC.
118			hacia	Datos	Transmisión de datos por el canal de retorno
119			del	Datos	Recepción de datos por el canal de retorno

Figura 1.3 Descripción de los circuitos RS-232 DB 25 y sus funcionalidades.

Número del circuito de enlace	Pin	Nombre de la señal	Del/hacia el ETCD	Tipo señal	DESCRIPCIÓN
102	5	signal ground	-	Tierra	Tierra de señalización o retorno común
103	3	TXD	hacia	Datos	Transmisión de datos de terminal local hacia el módem remoto
104	2	RXD	del	Datos	Recepción de los datos demodulados desde el terminal remoto
105	7	RTS	hacia	Control	Petición de transmisión
106	8	CTS	del	Control	Preparado para transmitir
107	6	DSR	del	Control	Aparato de datos preparado. Indica que el módem está preparado para recibir o transmitir datos
108.1	4	DTR	hacia	Control	Conectar el módem a la línea
108.2	4	DTR	hacia	Control	Terminal de datos preparado. El terminal está listo para transmitir y recibir datos
109	1	DCD	del	Control	Detector de señales de línea recibidas por el canal de datos. El módem genera una señal hacia el terminal de datos cuando detecta un nivel de portadora aceptable.
	9	RI	del	Control	Detector de señal de llamada. Indica se esta produciendo una llamada telefónica al módem

Figura 1.4 Descripción de los circuitos RS-232 DB 9 y sus funcionalidades.

En la figura 1.3 y 1.4 se muestran las especificaciones funcionales del RS-232-D para los conectores DB 25 y DB 9 respectivamente. Los circuitos se pueden agrupar en circuitos de datos que mantienen una comunicación *full duplex* (simultánea en ambos sentidos) y catorce circuitos de control. Préstese especial atención a los circuitos comprendidos entre el 102 y el 109.

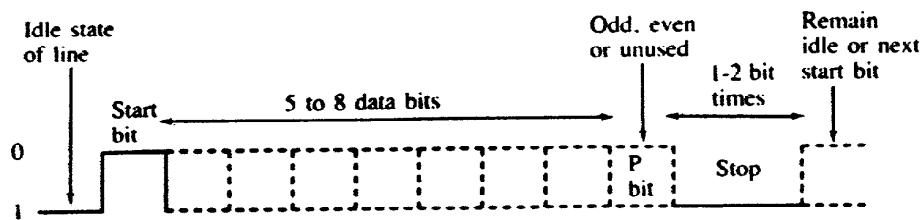
Mediante este interfaz se puede transmitir información tanto en modo asíncrono como síncrono. La transmisión asíncrona transmite caracteres de forma individual. El carácter de información junto con unos bits de control forman una trama cuya longitud oscila entre siete y diez bits. Esta trama permite al receptor sincronizarse, y saber cuando se inicia y finaliza la transmisión de la información. En este tipo de comunicaciones digitales el calificativo de asíncrono es debido a que el tiempo entre dos tramas contiguas es arbitrario.

La velocidad de transmisión fija el tiempo de bit y el inicio de una trama se señala mediante un bit de arranque "*start*" de nivel lógico **0**. Después siguen de cinco a ocho bits de datos formando el carácter, un bit de paridad (opcional), y de uno a dos bits de parada "*stop*", cuyo estado lógico es **1**. Su

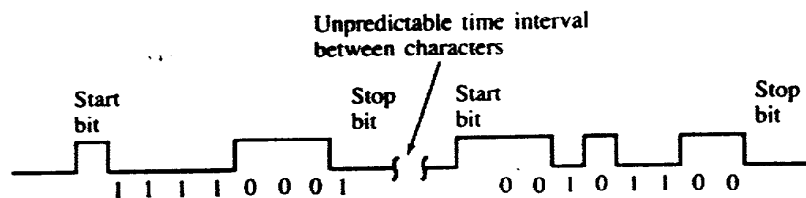
función es la de delimitar el fin de una trama con el principio de la siguiente, permitiendo la correcta extracción del campo de datos por parte del receptor.

El contenido del bit de paridad depende de la paridad que se elija, par o impar. En la paridad par el número de 1's de la trama contando el propio bit de paridad debe ser par. Análogamente sucede con la paridad impar. También es posible configurar de forma fija el bit de paridad a *marca* o *espacio*, o simplemente eliminarlo.

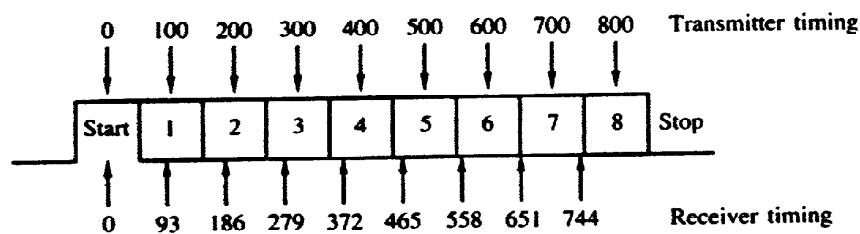
Las transmisiones asíncronas sobre RS-232 más simples sólo necesitan tres circuitos TXD, RXD y masa para transferir información, las funciones de control las realizan los niveles superiores.



(a) Character Format



(b) 8-bit Asynchronous Character Stream



(c) Effect of Timing Error

Figura 1.5 Transmisión asíncrona.

En las transmisiones síncronas la información se transmite en flujos continuos, ya sea de caracteres o de bits, sincronizándose el receptor con unas líneas específicas de temporización - o reloj - generadas por el emisor. En RS-232 los circuitos de temporización corresponden a las patillas 15, 17 y 24 del conector DB-25.

En los equipos de datos las tareas relacionadas con la transmisión asíncrona normalmente son gestionadas por las UART (Universal Asynchronous Receiver/Transmitter) independizando al procesador central y al programa de la problemática de la comunicación. La UART se puede ver como un bloque funcional que recibe/entrega información del/al canal via una comunicación asíncrona RS-232.

Las UARTs se realizan físicamente en hardware y llevan a cabo las siguientes funciones:

- Sincronización de la comunicación, relojes e interrupciones.

- Conversión de los datos serie recibidos por el interfaz RS-232 a formato paralelo manejables por la CPU del terminal o dispositivo o viceversa.
- Entramado y desentramado de la información. Construcción de la trama, bits de arranque, datos, paridad y parada.
- Control de la velocidad de la comunicación.
- Control de errores: paridad, sobrescritura (*overrun*), etc.

4. Control de flujo

Durante la transferencia de información entre dispositivos, puede ocurrir que la velocidad de procesamiento de uno de ellos sea inferior a la del otro, con lo que se podría perder parte de dicha información si no se utilizase algún mecanismo de arbitraje. Este mecanismo es precisamente el control de flujo.

Un ejemplo muy claro de la necesidad de estas técnicas es la comunicación entre un ordenador personal y una impresora a través del puerto serie. Evidentemente, la impresora no es capaz de procesar (imprimir) la información a la misma velocidad que le es enviada por el ordenador, debido a sus limitaciones mecánicas. Así, deberá tener algún modo de indicarle al ordenador cuando no puede procesar más información, y cuando vuelve a poder hacerlo. Para ello, existen dos modalidades:

- **Hardware:** El control de flujo se realiza mediante las líneas **RTS** y **CTS** del RS-232. Mediante RTS el emisor indica que quiere transmitir, y mediante CTS el receptor le da permiso o no (CTS activada/desactivada) para hacerlo. También se pueden encontrar algunos equipos que, de forma análoga, usan las líneas DTR y DSR para realizar el control de flujo.
- **Software:** En este caso tan solo es necesario utilizar las líneas de datos para llevar a cabo el control de flujo. Cuando el receptor no puede procesar más información, lo indica al emisor enviando un carácter predeterminado (**Xoff**) por la línea de datos. En el momento en que puede volver a procesar datos, envía otro carácter (**Xon**) con lo que el emisor vuelve a transmitir. Esta modalidad no está asociada al nivel físico, ya que se está interpretando la información que viaja por la línea.

5. Null-Modem (supresor de modem)

Generalmente el interfaz RS-232 interconecta un equipo terminal de datos ETD (**Data Terminal Equipment**) con un equipo terminal de circuito de datos ETCD (**Data Communication Equipment**) o módem. En entornos informáticos es usual conectar punto a punto via RS-232 dos terminales asíncronos de datos (DTE), siendo necesario modificar algunos circuitos de datos y de control del interfaz. Al cable que realiza tal tipo de conexión se denomina *null-modem*.

En un *null-modem* (Figura 1.6) se cruzan los circuitos de datos de emisión TXD y recepción RXD de ambos DTE's. A nivel de control en cada DTE se conecta el circuito RTS con CTS y DCD, y el circuito DTR con el DSR. Nótese que existen distintas configuraciones para el cableado obteniéndose diferentes funcionalidades. Durante el desarrollo de esta práctica se observarán estas conexiones en el pod de RS-232.

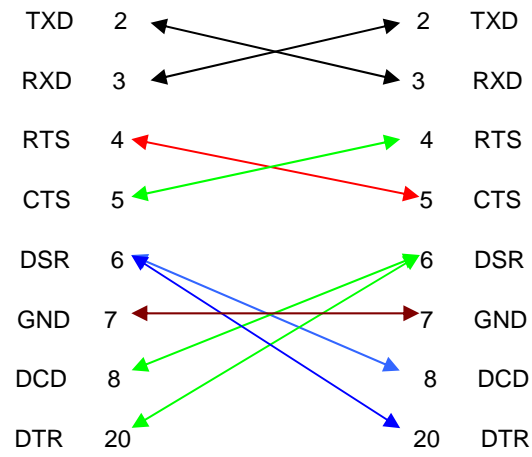


Figura 1.6. Null-Modem.

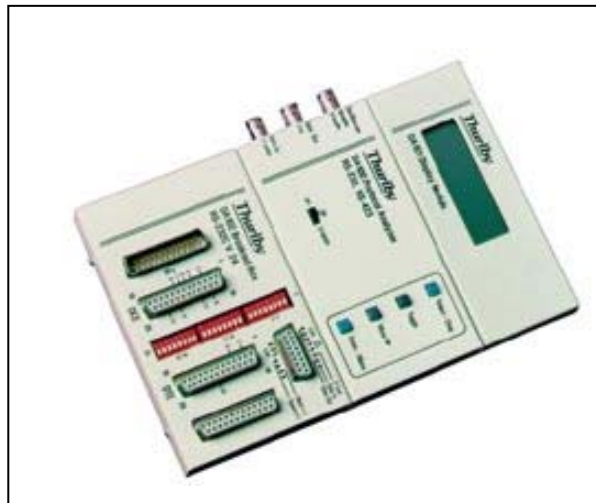


Figura 1.7. Pod RS-232.

6. Software de comunicaciones

En el entorno de los ordenadores personales existen numerosas aplicaciones denominadas *software de comunicaciones* que tienen por finalidad crear un entorno de comunicaciones, de fácil manejo para el usuario, que permita interconectar dos DTE's ya sea mediante un enlace digital punto a punto, o a través de una conexión analógica vía módems. También poseen comandos específicos para gobernar el módem o *emular* distintos terminales estándares, VT-100 y TTY. Además, proporcionan **protocolos de transferencia de ficheros**, los cuales construyen sobre el nivel físico un nivel de enlace de datos, ofreciendo servicios de entramado de la información, control de la comunicación (establecimiento, transmisión de la información y liberación de la conexión), control de flujo y control de errores.

Durante el desarrollo de la práctica utilizaremos el software de comunicaciones **Terminal** del entorno Windows. Para la transmisión de ficheros, este programa permite utilizar distintos protocolos a elegir por el usuario como **X-Modem** o **Kermit**.

6.1 X-MODEM

X-Modem es un protocolo ARQ (*automatic repeat request*, solicitud de repetición automática) de parada y espera (Stop&Wait) con un campo de datos de longitud fija (128 bytes). La trama del protocolo consta de cinco campos. Empieza con una cabecera de inicio de trama cuyo contenido es SOH (Start Of Header) de valor 01H. El segundo campo indica el número de secuencia de la trama y su valor se inicializa a 1. El contenido del tercer campo es el complemento a 1 del número de secuencia de la trama.

Sincronismo SOH	Control secuenciamiento	Complementario Control Secuenciamiento	Datos	Checksum/CRC
1 byte	1 byte	1 byte	128 bytes	1/2 bytes

Figura 1.8. Trama X-Modem.

El último campo de la trama tiene como misión el control de errores. En X-Modem, durante el establecimiento de la conexión, los terminales negocian la técnica a utilizar entre las dos posibilidades siguientes:

- **Checksum.** Se obtiene sumando el contenido del campo de datos y expresándolo en módulo 256. Su longitud será por tanto de 1 byte.
- **CRC.** Consta de 2 bytes y se calcula a partir de un polinomio de grado 16 normalizado por el CCITT ($x^{16} + x^{12} + x^5 + 1$).

En X-Modem una transferencia de información consta de tres fases: establecimiento, transferencia de datos y liberación. Para poder llevarlas a cabo, se hace necesaria la utilización de ciertos caracteres de control, incluidos dentro del código ASCII: **SOH, ACK, NAK, CAN** y **EOT**.

El **establecimiento** de la comunicación corre a cargo del receptor, que es el responsable de gestionar la comunicación manteniendo un flujo de tramas. Inicialmente, el receptor genera continuamente NAK's, permaneciendo a la espera de recibir la primera trama. En el momento en que el emisor reciba correctamente un NAK, enviará su primera trama. La fase de establecimiento finaliza si el receptor decodifica correctamente el SOH de esta primera trama.

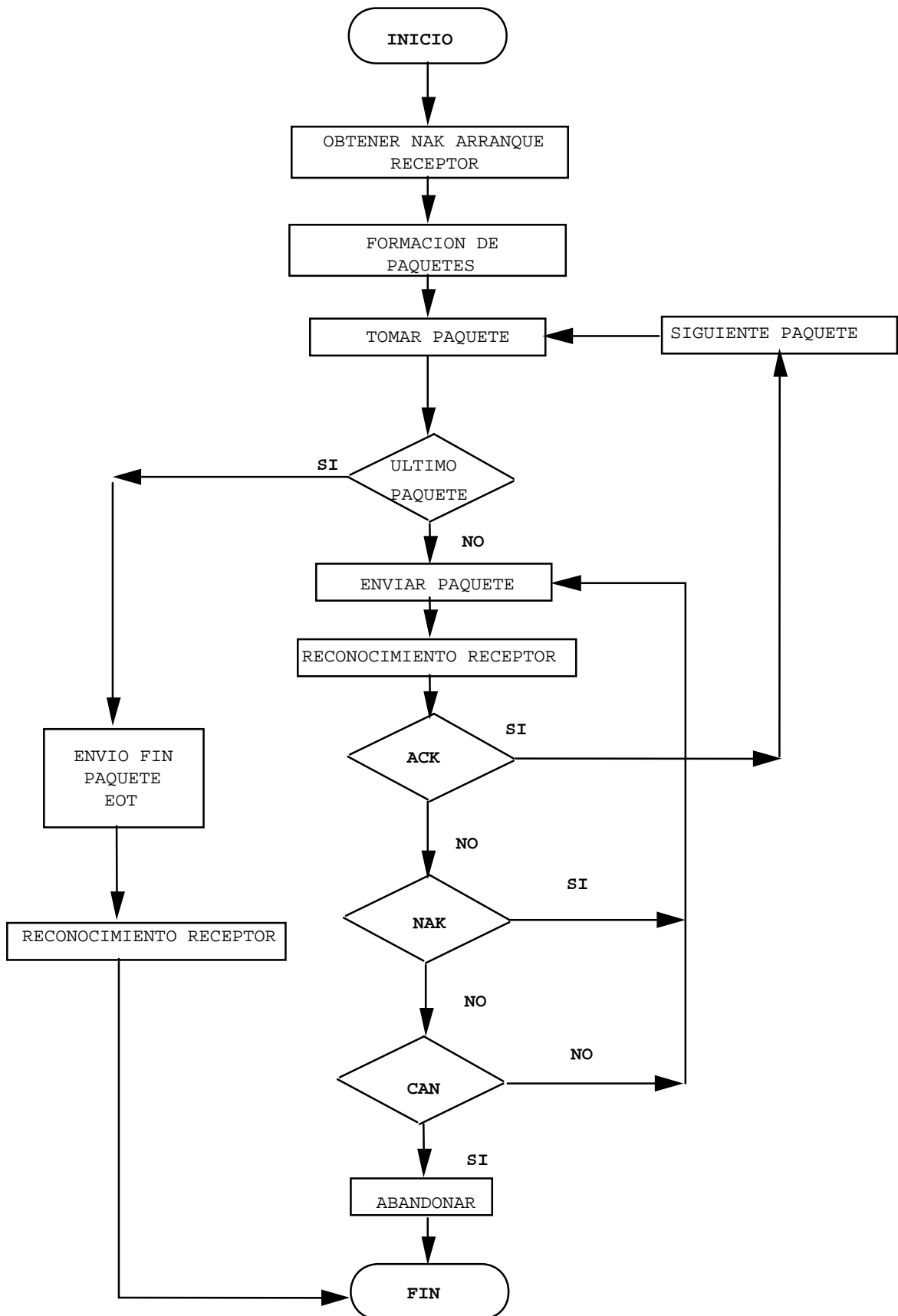
Durante la fase de **transferencia** el emisor espera siempre un reconocimiento positivo (ACK) o negativo (NAK) de cada una de las tramas que va enviando (de ahí el tipo de protocolo: parada y espera). Este reconocimiento ha de ser enviado por el receptor, una vez comprobados los campos de secuenciamiento y de control de errores. La fase de transferencia se aborta bruscamente si el emisor genera un paquete CAN (Cancel).

La **liberación** de la conexión se realiza ordenadamente cuando el emisor genera un EOT (End Of Transmission, 04H), recibiendo un ACK por parte del receptor.

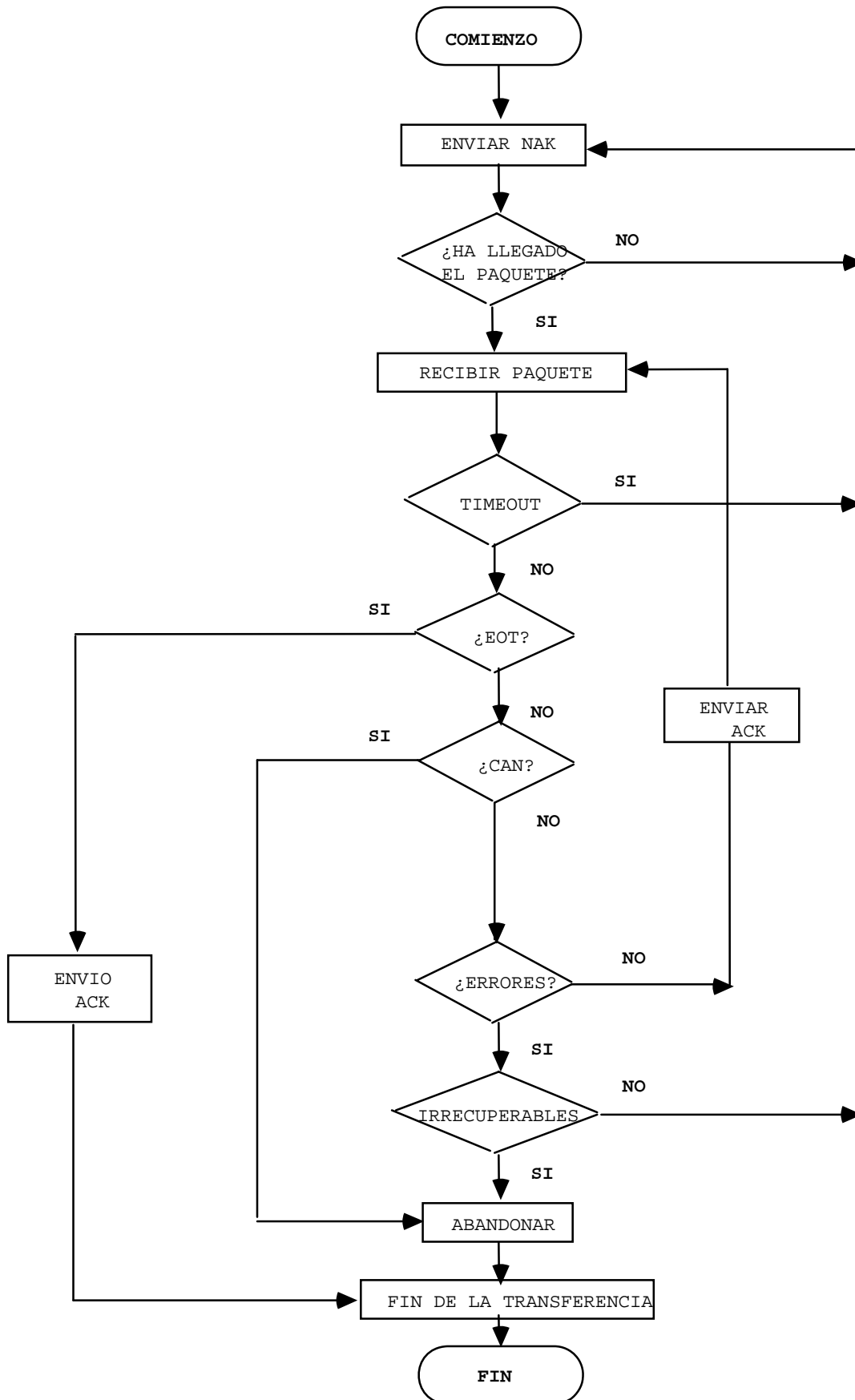
Nuevas versiones del protocolo permiten enviar varios ficheros.

Se debe observar que durante la transferencia de un fichero mediante X-Modem, el control de flujo es intrínseco al protocolo, ya que el receptor no permite al emisor que transmita más información hasta que no haya procesado la última trama. En realidad, el mecanismo de parada y espera es ya de por sí un método de control de flujo. De todas formas, algunas implementaciones del protocolo permiten combinar este control de flujo con el realizado mediante la técnica hardware (RTS/CTS).

X-MODEM : EMISION



X-MODEM: RECEPCION



7. Desarrollo de la práctica 1.

Nota: los ficheros que se utilicen durante las sesiones de prácticas, se encuentran en el directorio c:\. En las ocasiones en que se deba salvar algún fichero en el disco duro del ordenador, se utilizará el directorio c:\.

7.1 Inicialización y configuración.

En este primer apartado nos limitaremos a establecer una comunicación entre ambos ETDs (en la práctica ordenadores de tipo PC) con unos parámetros determinados, que nos ayudará a comenzar a familiarizarnos con el puesto de trabajo. Para ello configuraremos cada ETD con los siguientes parámetros:

- Emulación de terminal VT-100.
- Eco activado (opción half-duplex).
- Nueva línea asociada al retorno de carro (sólo para transferencias ASCII).
- Velocidad de transmisión: 2400 bits por segundo.
- Trama con 8 bits de datos, sin paridad y 1 bit de stop.

7.2 Estudio del nivel físico.

7.2.1 Conexionado.

Una vez finalizada la configuración, se realizarán distintas conexiones entre los puertos serie de ambos ETDs formando un enlace punto a punto via RS-232. Para ello, utilícese el POD RS-232 .

Recordando el concepto de Null Módem realizar 2 configuraciones distintas de conexión en el POD usando en la configuración del terminal la opción de ‘control de flujo hardware’.

Realizar 1 configuración de conexión en el POD usando en la configuración del terminal la opción de ‘control de flujo ninguno’.

Establézcase un diálogo entre los ETDs, para cada una de las conexiones propuestas observando como viaja la información en ambos sentidos con la ayuda del osciloscopio y del programa terminal.

7.2.2 Forma de onda.

En este apartado estudiaremos las características eléctricas de la comunicación, con la ayuda del osciloscopio. Para ello, debemos conectar uno de sus canales a alguna de las líneas de transmisión de datos del RS-232. Una vez realizada dicha conexión, se deberá configurar el osciloscopio para poder capturar los caracteres entramados en función de la velocidad elegida.

Durante la sesión de laboratorio, se capturarán las formas de onda mediante el osciloscopio correspondientes a la transmisión del carácter “a” con las configuraciones mostradas a continuación, prestando especial atención a la codificación y tiempos de bit y trama. En todas las transmisiones la configuración de los terminales receptor y transmisor serán iguales y se usarán las siguientes configuraciones.

- 2400 bps, 7 bits de datos, paridad impar y 1 bit de stop (7O1)
- 1200, 8N1.
- 9600 bps, 7N2.

Analizar los resultados obtenidos con el osciloscopio y compararlos con el código ASCII correspondiente a letra "a" (Ver tabla final adjunta).

7.2.3 Entramado y sincronización.

En este apartado, con la ayuda del osciloscopio, estudiaremos el formato de trama empleado por el interfaz RS-232. Para ello, debemos conectar uno de sus canales a alguna de las líneas de transmisión de datos del RS-232. Se realizará una conexión sin control de flujo y con el mínimo número de conexiones en el POD. Una vez realizada dicha conexión, se deberá configurar el osciloscopio para poder capturar los caracteres entramados en función de la velocidad elegida.

Se capturarán las formas de onda correspondientes a la transmisión de caracteres elegidos por el alumno con las configuraciones mostradas a continuación, prestando especial atención a la codificación y tiempos de bit y trama. En todas las transmisiones la configuración del terminal receptor será fija e igual a la primera de las que se muestran abajo. Para las configuraciones del transmisor serán usadas las siguientes:

- 2400 bps, 7 bits de datos, paridad impar(I) y 1 bit de stop (7I1)
- 1200, 8N1 (N = ninguna).
- 9600 bps, 7N2.

Obsérvese para cada uno de los casos anteriores la interpretación de los caracteres por parte del ETD receptor. Coméntense los resultados obtenidos con la ayuda de la tabla ASCII y justificar la interpretación de los caracteres recibidos por parte del terminal receptor.

7.3 Estudio del control de flujo.

Como ya hemos comentado, el control de flujo se hace necesario para adaptar las velocidades de procesamiento de datos de distintos elementos en una comunicación. En el caso que nos ocupa, tenemos dos ETDs con la misma velocidad de procesamiento por lo que, en general, dicho control de flujo será innecesario. Sin embargo, podemos simularlo de forma que nos quedará muy clara la forma en que se realizaría si fuese necesario.

7.3.1 Control de flujo software.

En primer lugar se conectarán los dos ETDs realizando las conexiones necesarias en el POD RS-232 para que se pueda realizar control de flujo hardware.

Seleccionaremos en ambos ETDs la modalidad de control de flujo 'software'. Los caracteres Xon y Xoff pueden simularse tecleando las combinaciones *CNTRL+ q* y *CNTRL + s*. Llévase a cabo en uno de los terminales y obsérvese el efecto que produce en el otro. Con la ayuda del osciloscopio, compruébese la codificación de los caracteres Xon y Xoff en línea, y dedúzcase que ocurre con la información tecleada en un ETD cuando el otro ETD le ha inhabilitado la transmisión mediante un Xoff.

Realícese una transferencia en modo texto y obsérvese el efecto de los caracteres Xon y Xoff al teclearlos desde el terminal receptor. Obsérvese que si el control de flujo seleccionado es software, el estado de las líneas RTS y CTS es indiferente, con lo que incluso podemos desconectarlas del POD.

7.3.2 Control de flujo hardware.

En primer lugar se conectarán los dos ETDs realizando las conexiones necesarias en el POD RS-232 para que se pueda realizar control de flujo hardware.

Seleccionaremos en ambos ETDs la modalidad de control de flujo 'hardware'. Procédase como en el caso anterior, observando qué ocurre ahora con los caracteres Xon/Xoff y qué pasa con la información tecleada desde un ETD y con la que se transmite al enviar un fichero.

Intentar la comunicación entre terminales y comentar los resultados cuando se desconectan algunas de las líneas de control. Es importante fijarse en lo que ocurre en cada uno de los terminales, pues el comportamiento puede ser diferente.

7.4 Control de dispositivos usando el puerto serie.

En este apartado usaremos los conocimientos adquiridos para realizar un caso práctico real en el que es necesario conocer el puerto serie. Se trata de controlar un dispositivo desde el puerto serie de un PC. En este caso será el osciloscopio digital que hemos usado en las prácticas. Este osciloscopio dispone un puerto serie para la comunicación con otros dispositivos y mediante en envío de comandos a través de este puerto serie podemos manejar el osciloscopio desde un PC. Para comunicarnos con el osciloscopio usaremos el programa terminal.

7.4.1 Determinación del cable a usar.

Lo primero que debemos hacer para hacer la conexión entre PC y dispositivo, en este caso osciloscopio, es determinar si el dispositivo funciona como ETD o como ETCD (como PC o como módem). Para ello se consultará la pagina 2.3 del manual de comunicaciones del osciloscopio. Una vez echo esto se realizará la conexión entre PC y dispositivo, usando el POD RS-232 con las conexiones necesarias, en el caso de que fuese necesaria una conexión null-módem.

7.4.2 Configuración del dispositivo.

El siguiente paso es configurar el dispositivo y el PC con los mismos parámetros de comunicación. Para ello, primero se consultará la pagina 2.6 del manual de comunicaciones del osciloscopio, se seleccionará una de las configuraciones posibles y por ultimo pondrá esa misma configuración en el programa terminal del PC.

7.4.3 Comunicación y control del dispositivo.

En el capítulo 4 del manual de comunicaciones del osciloscopio hay un listado y explicación con ejemplos de cada uno de los comandos que podemos usar para controlar el osciloscopio. Usando esta información se deberá enviar los comandos oportunos par controlar como mínimo:

- Activación de canales.
- Control de la base de tiempos.
- Control de voltios/división de cada canal
- Posición vertical de cada canal.
- Modo DC, AC y GND de cada canal.

Nota: El formato de los mensajes enviados se encuentra explicado con detalle en el capítulo 3 del manual de comunicaciones del osciloscopio.

7.5 Estudio del nivel de enlace.

Hasta ahora, nos hemos limitado a la transmisión de información sin preocuparnos de lo que ocurre en el caso de que se produzcan errores durante la comunicación. En este apartado nos dedicaremos a la transmisión fiable de información, utilizando los mecanismos de nivel de enlace que proporciona el X-Modem: entramado, control de flujo y de errores. Para la monitorización de las comunicaciones de este

apartado de las prácticas se usará el programa analizador que se encontrará en el PC monitorizador de cada puesto de trabajo.

7.5.1 Estudio del protocolo X-MODEM.

En primer lugar configuraremos ambos terminales a 9600 bps, 8N1 con el protocolo de transferencia de ficheros X-Modem. A continuación, se realizará la transferencia del fichero “**Fichero2.txt**” activando primero el emisor y luego el receptor. Analízese el protocolo:

- Verificar el entramado y tamaño del paquete
- Mídanse los tiempos transcurridos en cada una de las fases de la comunicación (establecimiento, mantenimiento y liberación) y el tiempo total de la transmisión completa.
- Comentar el mecanismo de control de flujo y errores.
- Realícense dibujos de los momentos más significativos de la transmisión.

A continuación se transmitirá de nuevo el mismo fichero, generando errores mediante la desconexión manual de las líneas de datos durante la transferencia, para ello se pueden abrir los interruptores del POD. Obsérvese la robustez del protocolo.

Realízese los siguientes cálculos:

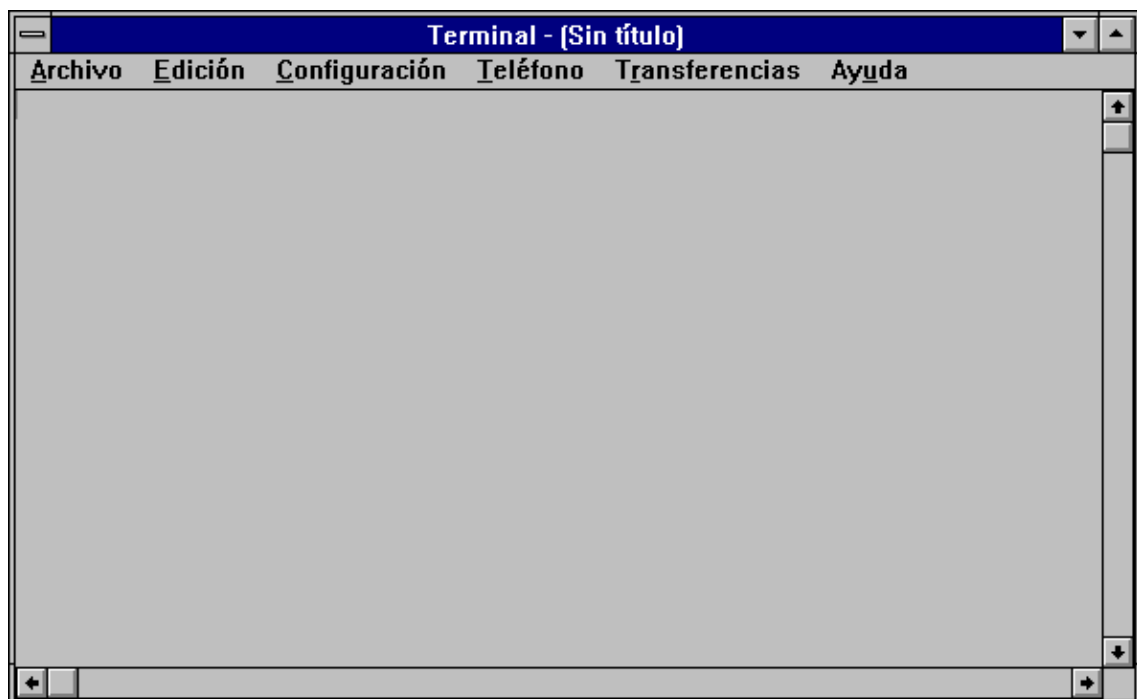
- Longitud del fichero transmitido mediante el Administrador de Archivos de Windows.
- Calcular de forma teórica la duración de una transmisión de tipo texto. Con los resultados obtenidos, calcúlese la relación de tiempos de transmisión del fichero anterior y de caracteres transmitidos (texto/X-Modem).
- Calcúlese también el rendimiento teórico del protocolo X-Modem (carga útil / tamaño de la trama). Coméntense las diferencias obtenidas en los cálculos anteriores.

Anexo 1: Terminal de Windows.

1.1 Introducción.

El programa **TERMINAL**, incluido en *Windows*, se encuentra en la carpeta *Accesorios* del Administrador de Programas. Ha sido copiado a la carpeta de *Inicio*, para que se ejecute automáticamente al entrar en *Windows*.

Permite intercambiar información y transferir ficheros entre distintos ordenadores o dispositivos que dispongan de conexión serie RS-232.



Al ejecutarse, aparece la ventana anterior, en la que se observa la línea con las opciones del menú y la zona del terminal en la que se pueden escribir caracteres desde el teclado que serán enviados via puerto serie al equipo remoto, y también en esta zona aparecerán los caracteres que recibamos del terminal distante.

1.2 Configuración.

1.2.1 Configuración del puerto de comunicaciones.

Para poder utilizar correctamente el programa terminal, se han de configurar previamente los parámetros de comunicación serie, así como el puerto que se usará. Todos los ordenadores del laboratorio disponen de dos puertos series.

Deben conocerse a priori los parámetros de transmisión que utiliza el dispositivo que está conectado a nuestro ordenador, en caso contrario no será posible la comunicación; éstos son: velocidad de transmisión, bits de datos, bits de parada, tipo de paridad y control de flujo. En el caso de que se conecte un módem, es muy probable que éste se adapte automáticamente a cualquier velocidad que fijemos.

Con los datos anteriores claros, procedemos a configurar el programa, accediendo en el menú de **Configuración** a la opción **Comunicaciones**, con lo que se presenta la siguiente ventana:



Las opción de verificación de paridad realiza la comprobación de errores en caso de que se produzcan debido a que la paridad de los bits recibidos no corresponde con la que se indicó. La opción de detección de portadora consiste en escuchar la línea CD (*Carrier Detect*) del interface serie; y no permitir enviar nada por el puerto serie hasta que esté activa. Es útil con módems para evitar que se pierdan caracteres cuando se intenta transmitir algo y el módem todavía no se ha conectado con el otro extremo (módem remoto).

1.2.2 Configuración del terminal.

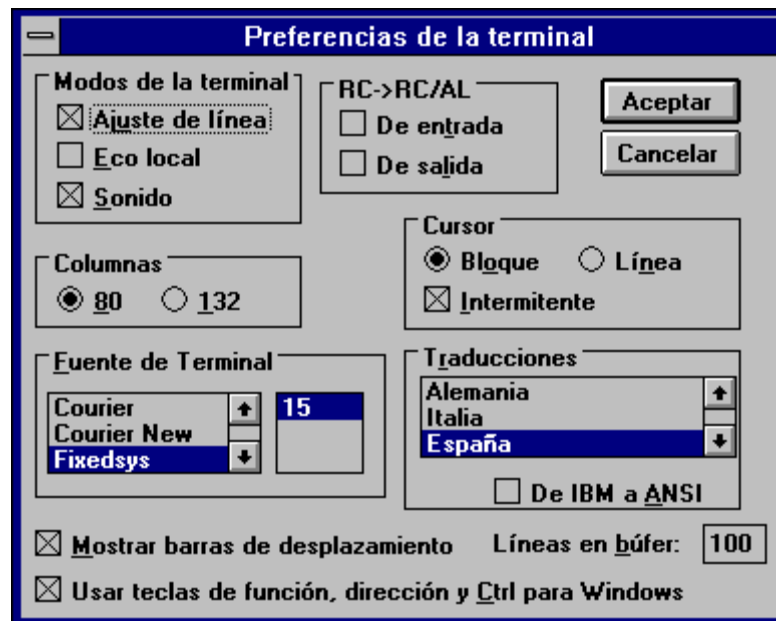
La opción **Preferencias del terminal**, del menú de **Configuración**, permite fijar los parámetros de visualización:

Los parámetros más interesantes de esta ventana son:

- **Eco local:** visualiza en la ventana del terminal todos los caracteres que se pulsan desde el teclado, para que podamos ver lo que escribimos. Nota: existen dispositivos, como los módems que también pueden realizar ecos, es decir si se les envía un carácter, además de transmitirlo al otro

extremo, también lo reenvían de nuevo hacia el terminal, de tal forma que si tenemos la opción anterior activada, veremos cada carácter doble.

- **RC->RC/AL:** Convertir el código ASCII de retorno de carro (13) en los códigos ASCII de retorno de carro (13) + avance de línea (10). Algunos dispositivos, como ordenadores con sistema operativo Unix, interpretan el código de retorno de carro (13) como una función compleja de retorno de carro y avance a la línea siguiente, mientras que en otros, como puede ser MSDOS o Windows, se requieren dos códigos diferentes, uno para cada función. Si se observa el efecto al pulsar la tecla Return, vuelve el cursor a la izquierda, pero no se baja una línea, sino que se sobrescribe siempre sobre la misma línea, conviene entonces, activar estas opciones adecuadamente.



1.2.3 Otros parámetros

El resto de parámetros del menú de **Configuración**, permiten afinar la configuración del terminal en detalle. De forma resumida, las opciones del menú de configuración son:

1. **Número de teléfono:** permite especificar un número telefónico, el cual se marcará automáticamente cuando se acceda a la opción de **Marcar** del menú **Teléfono**.
2. **Emulación del terminal:** permitir que el terminal se comporte como un terminal TTY, VT-52 o VT-100. Éstos se caracterizan en que interpretan ciertos códigos de control que permiten moverse por la pantalla o cambiar los atributos de los caracteres (colores, subrayado, etc.).

3. **Preferencias del terminal:** explicado anteriormente.

4. **Teclas de función:** permite asignar a las teclas de función (F1 a F10) secuencias de caracteres, de tal forma que al pulsar una tecla F, se reproducen automáticamente los caracteres asignados. Es útil para definir las secuencias de caracteres más comunes que utilizemos.

5. **Transferencias de texto:** especifica la forma de transferir archivos de texto, el método de control de flujo utilizado, el ajuste de las líneas de texto y si el texto se transferirá carácter a carácter cada vez o toda una línea cada vez.

6. **Transferencias binarias:** especifica el protocolo de transmisión de archivos binarios (Xmodem con CRC o Kermit).

7. **Comunicaciones:** explicado anteriormente.

8. **Comandos del módem:** permite configurar con los comandos del módem las funciones básicas de éste como la secuencia de códigos para marcar un número telefónico o colgar la línea. Con estos parámetros se puede automatizar la conexión/desconexión del módem, utilizando el menú **Teléfono**.

1.3 Descripción de los menús

1.3.1 Menú Archivo:

El programa terminal puede ser utilizado con diferentes dispositivos, y cada uno de ellos requerirá una configuración específica. Como puede ser muy tedioso tener que configurar cada vez el terminal, el programa permite grabar las configuraciones en ficheros, de tal forma que sólo abriendo uno de estos ficheros, se configuran automáticamente todos los parámetros asociados.

- **Nuevo:** permite crear un nuevo fichero de configuración. Inicialmente se utilizan los parámetros predeterminados.
- **Abrir:** abre un fichero de configuración y ajusta todos los parámetros.
- **Guardar, Guardar como:** permiten grabar la configuración actual del terminal en un fichero.
- **Especificar impresora:** permite indicar una impresora de Windows en la que se podrá imprimir todo lo que aparece en la ventana del terminal.

1.3.2 Edición:

Incluye las funciones básicas de copiar, cortar y pegar.

1.3.3 Configuración:

Ya se ha comentado en el punto 2.

1.3.4 Teléfono:

Permite automatizar las tareas de conexión y desconexión si se utiliza un módem.

1.3.5 Transferencias:

Permite realizar todas las opciones de transferencias de ficheros (texto o binarios), tanto enviar como recibir. La diferencia entre un fichero de texto y uno binario radica en que en los ficheros de textos pueden ser interpretados, incluso modificados mientras se transmiten (por ejemplo la conversión del código de retorno de carro en retorno de carro y avance de línea o viceversa), mientras que los ficheros binarios representan una secuencia de 1 y 0 que no deben ser modificados de ninguna manera, ya que representan algún tipo de información estructurada como imágenes, programas ejecutables, etc.

- **Enviar archivo de texto:** permite enviar un archivo de texto. Aparece una ventana en la que pregunta por un nombre de fichero y si se desea realizar algún tipo de mapeo (CR -> CR/LF, etc.)
- **Recibir archivo de texto:** permite especificar el nombre de un fichero donde se recibirá un fichero de texto.
- **Ver archivo de texto:** muestra por la ventana del terminal un fichero.

- **Enviar archivo binario:** permite indicar el fichero que transmitiremos con un protocolo Xmodem o Kermit, según hallamos configurado.
- **Recibir archivo binario:** permite indicar el fichero donde guardaremos los datos que recibiremos desde el puerto serie. **Nota:** el terminal que transmite el fichero debe tener configurado el mismo protocolo (Xmodem o Kermit), en caso contrario se producirán múltiples errores.

Anexo 2: Tabla de caracteres ASCII.

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	00 0000 0000 NUL	01 0001 0001 SOH	02 0010 0010 STX	03 0011 0011 ETX	04 0100 0100 EOT	05 0101 0101 ENQ	06 0110 0110 ACK	07 0111 0111 BEL	08 1000 1000 BS	09 1001 1001 HT	10 1010 1010 LF	11 1011 1011 VT	12 1100 1100 FF	13 1101 1101 CR	14 1110 1110 SO	15 1111 1111 SI
1	16 0001 0000 DLE	17 0001 0001 DC1	18 0001 0010 DC2	19 0001 0011 DC3	20 0000 0100 DC4	21 0001 0101 NAK	22 0001 0110 SYN	23 0001 0111 ETB	24 0001 1000 CAN	25 0001 1001 EM	26 0001 1010 SUB	27 0001 1011 ESC	28 0001 1100 FS	29 0001 1101 GS	30 0001 1110 RS	31 0001 1111 US
2	32 0010 0000 SP	33 0010 0001 !	34 0010 0010 "	35 0010 0011 #	36 0010 0100 \$	37 0010 0101 %	38 0010 0110 &	39 0010 0111 '	40 0010 1000 (41 0010 1001)	42 0010 1010 *	43 0010 1011 +	44 0010 1100 ,	45 0010 1101 -	46 0010 1110 .	47 0010 1111 /
3	48 0101 0000 0	49 0101 0001 1	50 0101 0010 2	51 0101 0011 3	52 0101 0100 4	53 0101 0101 5	54 0101 0110 6	55 0101 0111 7	56 0101 1000 8	57 0101 1001 9	58 0101 1010 :	59 0101 1011 ;	60 0101 1100 <	61 0101 1101 =	62 0101 1110 >	63 0101 1111 ?
4	64 0100 0000 @	65 0100 0001 A	66 0100 0010 B	67 0100 0011 C	68 0100 0100 D	69 0100 0101 E	70 0100 0110 F	71 0100 0111 G	72 0100 1000 H	73 0100 1001 I	74 0100 1010 J	75 0100 1011 K	76 0100 1100 L	77 0100 1101 M	78 0100 1110 N	79 0100 1111 O
5	80 0101 0000 P	81 0101 0001 Q	82 0101 0010 R	83 0101 0011 S	84 0101 0100 T	85 0101 0101 U	86 0101 0110 V	87 0101 0111 W	88 0101 1000 X	89 0101 1001 Y	90 0101 1010 Z	91 0101 1011 [92 0101 1100 \	93 0101 1101]	94 0101 1110 ^	95 0101 1111 _
6	96 0110 0000 ,	97 0110 0001 a	98 0110 0010 b	99 0110 0011 c	100 0110 0100 d	101 0110 0101 e	102 0110 0110 f	103 0110 0111 g	104 0110 1000 h	105 0110 1001 i	106 0110 1010 j	107 0110 1011 k	108 0110 1100 l	109 0110 1101 m	110 0110 1110 n	111 0110 1111 o
7	112 0111 0000 p	113 0111 0001 q	114 0111 0010 r	115 0111 0011 s	116 0111 0100 t	117 0111 0101 u	118 0111 0110 v	119 0111 0111 w	120 0111 1000 x	121 0111 1001 y	122 0111 1010 z	123 0111 1011 {	124 0111 1100 \	125 0111 1101 }	126 0111 1110 ~	127 0111 1111 DEL