



SISTEMAS DE COMUNICACIÓN

Capítulo Sexto

INTRODUCCIÓN

Actualmente existe una clara tendencia a sustituir los tradicionales sistemas de control centralizados por sistemas distribuidos de control. Ya que es una naturaleza de los sistemas de control, distribuir sensores a lo largo de todo el proceso. Así que por ello la tendencia es controlar los procesos en pequeños sistemas situados en cerca del proceso. Así ya no es necesario poner un complicado, caro y poco fiable sistema de cableado. Y así se utiliza un controlador maestro que coordine a los otros controladores de los procesos en particular, a través de una red industrial o una pequeña red de comunicaciones digitales. Existen muchos métodos para comunicar estos controladores. Algunos de ellos son los que vamos a ver aquí

HISTORIA

La Historia del RS-232

El RS-232 fue creado originalmente con el propósito de estandarizar las interconexiones de terminales y la computadora servidor utilizando el la línea telefónica para ello. El dispositivo encargado de hacer la conexión a través de la línea telefónica es el modem. Los Modemes son los encargados de traducir los datos enviados por el ordenador que los manda en forma digital y pasarlos a una forma analógica para que puedan ser transmitidos por la red telefónica y una vez que llegaron al modem destino, regresarlos a su formato digital. Entonces, el estándar RS-232 fue creado para establecer dicha comunicación.

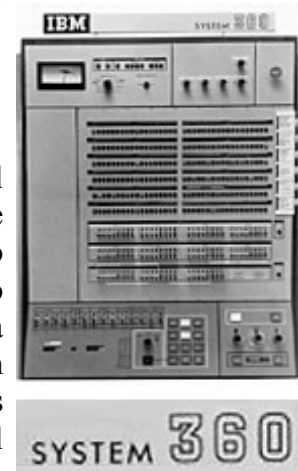


Figura 6.2 - El IBM 360, una supercomputadora de 1964



Figura 6.1 - Logotipo de la EIA

En los años 60's casi todas las conexiones de acceso remoto existentes, entre computadoras era vía telefónica. El acceso a los Mainframes de la época era casi exclusivamente por vía telefónica. Pero el problema que existía en aquella época, era que cada compañía que fabricaba equipo de comunicación usaba una configuración propia para hacer la interfase entre el DTE (Data Terminal Equipment) y el DCE (Data Communication Equipment).

Cada fabricante tenía su configuración, cada fabricante tenía su propio estándar, cada fabricante utilizaba conectores, cables, voltajes, etc. diferentes. Por tanto cada marca era incompatible con la otra. Para poder comunicar dos dispositivos de diferentes marcas, era necesario elaborar circuitos especiales que ajustaran los voltajes apropiados para cada fabricante, así como de hacer conectores especiales que conviertan una interfaz propia de la marca en la de la contraparte. Por eso es que, aún hoy día, cuando se va a instalar o diseñar una red, se tiene la precaución de casarse con una marca en especial. Y todo los dispositivos, software y demás que tenga que ver con la red, tenga que ser de esa marca en específico.

En 1969 EIA (Electronic Industries Alliance) junto con los Laboratorios Bell establecieron una serie de estándares recomendados (Recommended Standard) para comunicar y hacer la interfaz entre los equipos DTE y DCE. Teniendo como principios de esos estándares el simplificar la difícil tarea de hacer una interconexión entre equipos de diferentes marcas o fabricantes. Estos estándares definían las características eléctricas, mecánica y funcionales de la conexión. Las características eléctricas definían parámetros como voltajes, impedancias, longitudes, etc. Los parámetros mecánicos es el arreglo de los pines, la forma del conector. Y los parámetros funcionales definían la función de cada una de las señales a utilizar.

El primer estándar fue el RS-232-C (Recommended Standard number 232, revision C de la Electronic Industry Association) y a la par la CCITT (Comite Consultatif Internatinal de Telegraphie et Telephonie), creó un estándar muy parecido llamado V.24 (descripción funcional) y el V.28 (Especificaciones eléctricas), éste último tuvo más auge en Europa, pero el que en realidad tuvo éxito fue el RS-232-C, ya que la gran mayoría de los fabricantes de equipo a nivel mundial adoptaron este estándar

RS-232

RS-232

El RS-232 es una de los estándares de comunicación digital, más popular y versátil de la historia. Ha sido el caballo de batalla en cuanto a interfases se refiere. Ha estado presente en casi toda las PC's así como micro controladores, periféricos, Main frames, etc. El RS-232 en un inicio fue pensado, y generalmente aplicado, para conectar una computadora con un modem. No pasó mucho tiempo, para que otros dispositivos como: impresoras, ratones, teclados, sistemas de adquisición de datos, instrumentos de medición, PLC's, etc. adoptaran este estándar para comunicarse con la PC o entre ellos mismos. Ya que el RS-232 es tan popular, podemos afirmar que puedes conectar dos computadoras cualquiera a través del RS-232, de una forma muy rápida y económica.

El estándar de comunicación RS-232 ha sido popular desde hace ya algunas décadas, y sigue siendo popular, debido a que es muy simple de programar, y el hardware necesario no es muy costoso. De hecho podríamos afirmar que es uno de los más baratos del mercado. Otra gran ventaja es que, actualmente, existen muchos dispositivos que ya vienen diseñados para poderse comunicar vía serial utilizando RS-232, y por tanto es más barato y fácil comunicar un dispositivo utilizando el mismo estándar que su contraparte. El desarrollo de las comunicaciones digitales ha seguido su curso, y han aparecido nuevos estándares que mejoran al RS-232, incrementan su velocidad o su distancia máxima recomendada.

Pero la gran mayoría permanece compatible con su predecesor RS-232. Debido a que los puertos para RS-232 son tan comunes, muchos dispositivos que utilizan otros estándares, recomiendan usar adaptadores que se conectan al puerto serial RS-232 y convierten esa información al otro estándar.

El RS-232 es un estándar para comunicación serial asíncrona entre dos dispositivos. La distancia máxima recomendada puede variar de 50 a 100 pies. El RS-232 se dice que es un estándar que usa líneas de transmisión desbalanceadas, esto quiere decir que sólo utilizan una línea de tierra común. Mientras que un sistema balanceado utilizaría una línea de tierra por cada señal que se esté utilizando.

VENTAJAS Ventajas del RS-232:

DEL RS-232

- T Es ubicuo, prácticamente cada PC tiene al menos un puerto serial RS-232. Y aunque las computadoras nuevas que soportan enlaces vía USB. No pueden dejar a un lado la interfaz RS-232, ya que hay cosas que no puede hacer el USB. Muchos usuarios de computadoras, se ven forzados a comprar puertos RS-232 debido a su gran popularidad entre los periféricos
- T Es muy fácil convertir voltajes TTL de los microprocesadores a voltajes adecuados para transmisión en RS-232. Basta con unos capacitores y el integrado Max232
- T Los enlaces pueden ser de entre 50 y 100 pies de longitud máxima recomendada. Los enlaces basados en USB sólo pueden alcanzar 16 pies, si hacemos un enlace por el puerto paralelo, podremos alcanzar distancias máximas de entre 10 y 15 pies. Pero si utilizamos RS-232 junto con un modem, podemos transmitir la información por la línea telefónica y abarcar distancias intercontinentales.
- T El número mínimo de líneas de transmisión para tener una comunicación de dos vías es de tres: Tx, Rx, y tierra.

DESVENTAJAS Desventajas del RS-232

- DEL RS-232**
- X Si el dispositivo remoto requiere la información en forma paralela, deberá de añadir hardware para transformar la información de vía serial a vía paralela. Esto se hace fácilmente con un circuito integrado UART.
 - X Los puertos seriales son tan útiles, que a veces es difícil encontrar uno libre
 - X El RS-232 está diseñado para tener comunicaciones punto a punto, es decir, solamente dos dispositivos puedan estar comunicándose al mismo tiempo
 - X La velocidad máxima es de 20,000 bits por segundo.

EL ESTÁNDAR El puerto serie RS-232C, presente en todos los ordenadores actuales, es la forma más comúnmente usada para realizar transmisiones de datos entre ordenadores. El RS-232C 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 serie 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.

Número pin		Nombre	Descripción	Dirección
DB25	DB9			
1		Earth	Protección a Tierra	
2	3	TxD	Transmit Data	S
3	2	RxD	Receive Data	E
4	7	RTS	Request To Send	S
5	8	CTS	Clear to Send	E
6	6	DSR	Data Set Ready	E
7	5	SG	Signal Ground	
8	1	CD/DCD	Data Carrier Detect	E
15	-	TxC	Transmit Clock	S
17	-	RxC	Receive Clock	E
20	4	DTR	Data Terminal Ready	S
22	9	RI	Ring Indicator	E
24	-	RTxC	Transmit / Receive	S

Protección a tierra:	Protección a tierra: Esta línea va conectada al neutro de la línea de alimentación del adaptador serial. No debería de ser utilizada como tierra de la señal, sino que debe de ir conectada al blindaje del cableado, si existe. Al ser implementada en ambos lados de la comunicación asegura que no hay grandes corrientes fluyendo en la tierra de la señal debido a un defecto de aislamiento o algún otro defecto en cualquier lado del cableado. Por otra parte, cuando dos dispositivos están separados por grandes distancias no debe de utilizarse debido a la diferencia de potencial que puede existir entre los equipos haciendo posible que pueda llevar una corriente importante como bucle a tierra. En cualquier caso, la falla de esta línea puede causar interferencia eléctrica.
Señal de Tierra	Señal de Tierra: Esta es la tierra lógica que es utilizada como punto de referencia por todas las señales recibidas y transmitidas. Esta señal es indispensable y debe de estar presente para todas las comunicaciones.
Transmisión de datos	Transmisión de datos (TX): Esta línea es utilizada para transmitir datos desde el DTE al DCE. Es mantenida en estado de 1 lógico cuando nada se transmite. La terminal comenzará a transmitir cuando un 1 lógico esté presente en las siguientes líneas: Autorización de envío, Terminal de datos lista, Datos listos para enviar, Detección de portadora.
Recepción de datos	Recepción de datos (Rx): Este circuito es utilizado para recibir datos desde el DCE al DTE. La terminal comenzará a transmitir cuando un 1 lógico esté presente en las siguientes líneas: Autorización de envío, Terminal de datos lista, Datos listos para enviar, Detección de portadora. El estándar especifica que los niveles de salida son -5 a -15 volts para el 1 lógico y +5 a +15 volts para el 0 lógico, mientras que los niveles de entrada son -3 a -15 volts para un 1 lógico y +3 a +15 volts para un 0 lógico. Esto asegura que los bits puedan ser leídos correctamente aún con grandes distancias entre la DTE y la DCE, especificados como 16.5 metros o 50 pies, aún cuando estas señales soportan mayores distancias dependiendo de la calidad del cableado y el blindaje.
Solicitud de envío	Solicitud de envío (RTS): En esta línea el DTE envía una señal cuando está listo para recibir datos del DCE. El DCE revisa esta línea para conocer el estado del DTE y saber si puede enviar datos.
Autorización de envío	Autorización de envío (CTS): En este circuito el DCE envía una señal cuando está listo para recibir datos del DTE.
Datos listos para enviar	Datos listos para enviar (DSR): Cuando este circuito está en 1 lógico indica al DTE que el DCE está listo para enviar datos.
Terminal de datos lista	Terminal de datos lista (DTR): Cuando ésta línea está en estado de 1 lógico se puede comenzar a enviar y recibir datos. Cuando esta línea está en nivel de 0 lógico, el DCE terminará la comunicación
Detección de Portadora de Datos	Detección de Portadora de Datos (DCD): En esta línea el DCE indica al DTE que ha establecido una línea portadora (una conexión) con un dispositivo remoto.

RS-485 Lo que la mayoría de la gente llama RS-485, es la interfase descrita en el documento TIA/EIA- 485 o el ISO/IEC 8482.1993. El RS-485 es la mejora del RS-422 que a su vez es la mejora del RS-232. Por tanto ellos tres son muy parecidos. Pero tienen sus diferencias, por ejemplo: Las señales del RS232 se manejan por voltaje con respecto a una tierra común. Esto es muy útil para distancias cortas y bajas velocidades. Pero cuando las distancias son mayores las interferencia y ruido obtenido se hace considerable, por eso es que RS422 y RS485 utiliza líneas de par trenzado. Y cada señal tiene su propia línea de retorno, con lo cual se estamos hablando de una transmisión de datos en línea balanceada o una transmisión de datos por diferencia de voltaje. Con esto podemos alcanzar distancias de hasta 1,200 metros y velocidades de 2.5 Mbps. Otra diferencia, es que RS485 está diseñado para enlaces multipunto, mientras que RS232 está diseñado para enlaces punto a punto (aunque no quiere decir que no se pueda hacer enlaces multipunto). Y la mayoría de las aplicaciones de RS385 es conectar dispositivos en red utilizando la arquitectura Maestro / Esclavo, en donde cada esclavo tiene una dirección única y responde solamente con paquetes que tengan esa dirección. Los paquetes son generados por el maestro, quien periódicamente polea a cada uno de los esclavos conectados.

LÍNEAS BALANCEADAS Y LÍNEAS NO BALANCEADAS

LÍNEAS BALANCEADAS Y LÍNEAS NO BALANCEADAS

Entonces la principal razón por la cual el RS485 puede transmitir sobre largas distancias es que utiliza líneas balanceadas. Cada señal tiene un par de cables dedicados única y exclusivamente a esa señal. Y en la que un cable es el que porta la señal y el otro es el negativo o el complemento de la otra. Y el receptor lo que hace es ver si existe una diferencia de voltaje entre ellas.

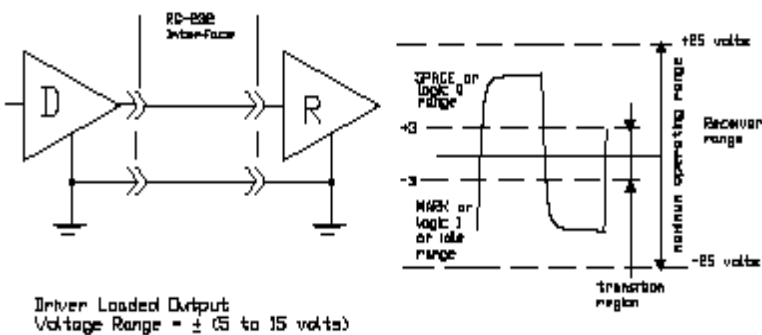


Figura 6.3 - Línea no balanceada, todas las señales se miden con respecto a tierra común.

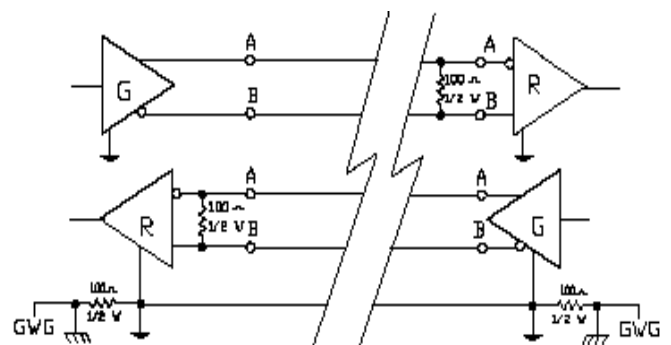


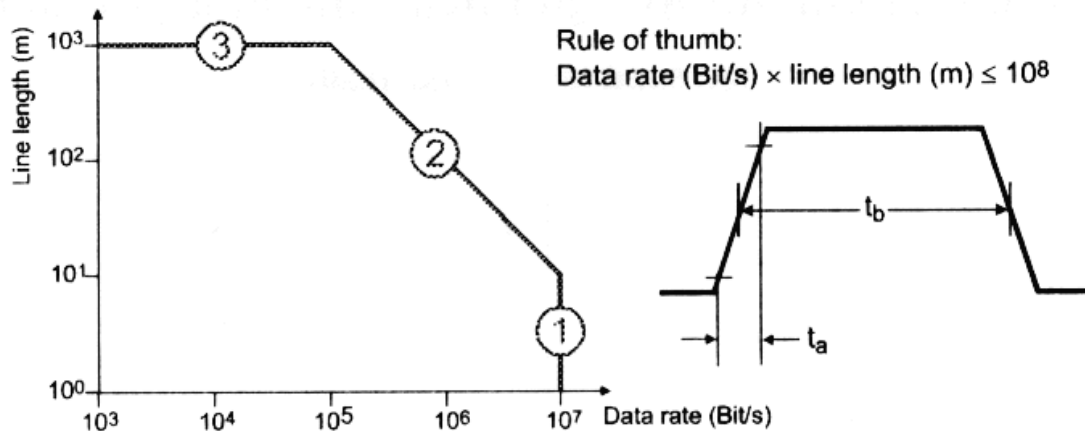
Figura 6.4 - Línea balanceada, todas las señales tienen dos cables que las conducen

La gran ventaja de medir las señales en diferencia, es que cuando se induce ruido, se va a inducir el mismo ruido en los dos cables, y que a la hora de sacar la diferencia se elimina el ruido. Otra gran ventaja de utilizar líneas balanceadas es que son altamente inmunes a las variaciones de voltaje del suministro eléctrico.

LA VELOCIDAD

LA VELOCIDAD Un enlace basado en RS-485 puede ser tan rápido como 10MBps o puede abarcar distancias tan grandes como 4000 pies. Pero lo que no puede hacer es ambas a la vez. Distancias largas requieren de tasas más bajas, ya que los cables largos traen consigo capacitancias grandes. La TIA/EIA recomienda usar la siguiente gráfica como guía para determinar la longitud y la velocidad recomendada.

Maximum Data Rate in a Balanced Interface



- ① Data rate is limited by the rise time of the output signal of the generator (ITU-T V.11: $t_r \leq 0.1 \times t_b$; ISO8482: $t_r \leq 0.3 \times t_b$).
- ② Data rate is limited due to signal distortion caused by losses on the transmission line (e.g. skin effect).
- ③ Data rate is limited due to the attenuation caused by the ohmic resistance of the wires (a 1000 m twisted pair cable with wires of 0.6 mm diameter has a resistance of $\approx 100 \Omega$).

Figura 6.5 - Gráfica recomendada por la TIA/ EIA para determinar la tasa de transferencia en base a la longitud del enlace

PROTOCOLOS

PROTOCOLOS

Se les llama protocolo de red o protocolo de comunicación al conjunto de reglas que controlan la secuencia de mensajes que ocurren durante una comunicación entre entidades que forman una red. En este contexto, las entidades de las cuales se habla son programas de computadora o automatismos de otro tipo, tales y como dispositivos electrónicos capaces de interactuar en una red.

Los protocolos de red establecen aspectos tales como:

- T Las secuencias posibles de mensaje que pueden arribar durante el proceso de la comunicación.
- T La sintaxis de los mensajes intercambiados.
- T Estrategias para corregir los casos de error.
- T Estrategias para asegurar la seguridad (autenticación, encriptación).

Los protocolos que son implementados en sistemas de comunicación que tienen un amplio impacto, suelen convertirse en estándares, debido a que la comunicación e intercambio de información (datos) es un factor fundamental en numerosos sistemas, y para asegurar tal comunicación se vuelve necesario copiar el diseño y funcionamiento a partir del ejemplo pre-existente. Esto ocurre tanto de manera informal como deliberada.

Existen consorcios empresariales, que tienen como propósito precisamente el de proponer recomendaciones de estándares que se deben respetar para asegurar la interoperabilidad de los productos. Según la clasificación OSI, la comunicación de varios dispositivos se puede estudiar dividiéndola en 7 niveles:

Nivel	Nombre	Categoría
7	Aplicación	
6	Presentación	Aplicación
5	Sesión	
4	Transporte	
3	Red	
2	Enlace de Datos	Transporte de Datos
1	Físico	

Algunos de los protocolos existentes son:

MAESTRO / ESCLAVO

PROCOLO MAESTRO / ESCLAVO

El protocolo Maestro esclavo es el más simple de todos. Un nodo se designa como maestro y es el que está a cargo de controlar el tráfico de información en la red. En este protocolo, todos los esclavos permanecerán callados hasta que el maestro le indique lo contrario. Para asegurarse de que todos los nodos tengan la posibilidad de ser escuchados, el maestro hace un poleo, y cada poleo requiere de una respuesta y la que puede ser un simple “enterado” o puede pedir información, mandar un error o un estado. Este protocolo es muy útil. Pero tiene la enorme desventaja de que es vulnerable a tener mucho retraso, debido a que el maestro debe de preguntar a cada esclavo su situación.

TOKEN RING

TOKEN RING

Arquitectura de red desarrollada por IBM con topología lógica en anillo y técnica de acceso de paso de testigo. Cumple el estándar IEEE 802.5. El acceso al medio es determinista por el paso de testigo o token, o también llamado testigo, a diferencia de otras redes de acceso no determinístico (estocástico, como Ethernet). Un token es pasado de computadora en computadora, y cuando una de ellas desea transmitir datos, debe esperar la llegada del token vacío, el cual tomará e introducirá los datos a transmitir, y enviará el token con los datos al destino. Una vez que la computadora destino recibe el token con los datos, lo envía de regreso a la computadora que lo envió con los datos, con el mensaje de que los datos fueron recibidos correctamente, y se libera el token, yendo nuevamente de computadora en computadora hasta que otra máquina desee transmitir, y así se repetirá el proceso. La norma IEEE 802.5 es un estándar definido por el IEEE (Institute of Electrical and Electronics Engineers,) y define una red de área local (LAN) en configuración de anillo (Ring), con método de paso de testigo (Token) como control de acceso al medio. Su velocidad del estándar es de 4 ó 16 Mbps. El primer diseño de una red de Token Ring fue atribuido a E. E. Newhall en el año 1969. IBM publicó por primera vez su topología de Token Ring en marzo de 1982, cuando esta compañía presentó los papeles para el proyecto 802 del IEEE. IBM anunció un producto Token Ring en 1984, y en 1985 éste llegó a ser un estándar de ANSI/IEEE. Es casi idéntica y totalmente compatible con la red del token ring de IBM. De hecho, la especificación de IEEE 802.5 fue modelada después del token ring, y continúa sombreando el desarrollo del mismo. Además, el token ring de la IBM especifica una estrella, con todas las estaciones del extremo unidas a un dispositivo al que se le llama "unidad del acceso multiestación" (MSAU). En contraste, IEEE 802.5 no especifica una topología, aunque virtualmente todo el IEEE 802.5 puesto en práctica se basa en una estrella, y tampoco especifica un tipo de medios, mientras que las redes del token ring de la IBM utilizan el tamaño del campo de información de encaminamiento.

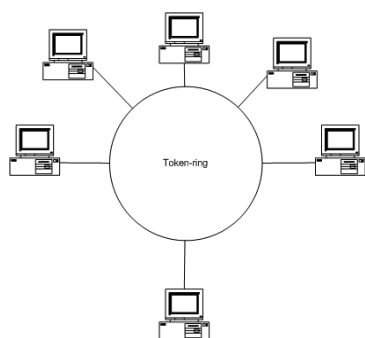


Figura 6.6 - Esquema de una red Token Ring

TOKEN BUS

TOKEN BUS

Token Bus es un protocolo para redes de área local análogo a Token Ring, pero en vez de estar destinado a topologías en anillo está diseñado para topologías en bus. Los nodos están conectados por cable coaxial y se organizan en un anillo virtual. En todo momento hay un testigo (token) que los nodos de la red se lo van pasando, y únicamente el nodo que tiene el testigo tiene permiso para transmitir. Si el nodo no tuviese que enviar ningún dato, el testigo es pasado al siguiente nodo del anillo virtual. Todos los nodos deben saber las direcciones de sus vecinos en el anillo, por lo que es necesario un protocolo que notifique si un nodo se desconecta del anillo, o las incorporaciones al mismo.

Token bus está definido en el estándar IEEE 802.4. El protocolo ARCNET es similar, pero no sigue este estándar. Token Bus se utiliza principalmente en aplicaciones industriales. Fue muy apoyado por GM. Actualmente en desuso.

CSMA/CD

CSMA/CD

CSMA/CD, siglas que corresponden a Carrier Sense Multiple Access with Collision Detection (en español, "Acceso Múltiple con Escucha de Portadora y Detección de Colisiones"), En el método de acceso CSMA/CD, los dispositivos de red que tienen datos para transmitir funcionan en el modo "escuchar antes de transmitir". Esto significa que cuando un nodo desea enviar datos, primero debe determinar si los medios de red están ocupados. CSMA (Carrier Sense Multiple Access, acceso múltiple con escucha de portadora) significa que se utiliza un medio de acceso múltiple y que la estación que desea emitir previamente escucha el canal antes de emitir. Si el canal está ocupado espera un tiempo aleatorio y vuelve a escuchar. Cuando detecta libre el canal puede actuar de dos formas distintas: emitiendo de inmediato o esperando un tiempo aleatorio antes de emitir. Si emite con una probabilidad p , se dice que es un sistema CSMA p -persistente, mientras que si emite de inmediato se dice que es un sistema CSMA 1-persistente. Una vez comenzado a emitir, no para hasta terminar de emitir la trama completa. Esto supone que se puede producir una colisión si dos estaciones intentan transmitir a la vez, de forma que las tramas emitidas por ambas serán incompresibles para las otras estaciones y la transmisión habrá sido infructuosa.

Finalmente CSMA/CD supone una mejora sobre CSMA, pues la estación está a la escucha a la vez que emite, de forma que si detecta que se produce una colisión para inmediatamente la transmisión. La ganancia producida es el tiempo que no se continua utilizando el medio para realizar una transmisión que resultará inútil, y que se podrá utilizar por otra estación para transmitir.

BITBUS

EL PROTOCOLO DE COMUNICACIONES: BITBUS

BITBUS es un bus de control serie maestro / esclavo, específicamente diseñado por INTEL que soporta hasta 250 nodos y distancias que van desde los 30 metros, en modo síncrono, hasta cientos de metros en modo auto sincronizado. Permite transmitir mensajes de control, normalmente de corta longitud, en distancias largas y a alta velocidad. BITBUS proporciona no sólo las normas eléctricas y los protocolos de enlace, sino también protocolos de transferencia de mensajes en entornos multitarea y órdenes para entrada/salida remota y control de tareas de aplicación.

El protocolo BITBUS fue diseñado para proporcionar un mecanismo de interconexión de alta velocidad para sistemas jerárquicos. Esta jerarquía puede representarse en un solo nivel, donde un equipo maestro puede comunicarse con múltiples dispositivos esclavos, o en varios niveles, donde un dispositivo puede comportarse como maestro en un nivel y como esclavo de otro. El objetivo es dotar de un mecanismo de intercambio de mensajes entre las tareas del nodo maestro y las tareas de los diferentes esclavos. Esto se logra mediante un protocolo de paso de mensajes del tipo orden/respuesta. Las tareas del nodo maestro envían órdenes a las tareas de nodos esclavo, que a su vez envían respuestas a las tareas del nodo maestro.

**DISPOSITIVO
MAESTRO**

DISPOSITIVO MAESTRO

Es el encargado de controlar todas las operaciones. Inicia la comunicación enviando una orden al esclavo y a continuación sonde por el polling al esclavo hasta que recibe un mensaje de respuesta. En el modo normal de funcionamiento cada transferencia desde el maestro es reconocida por el esclavo diseccionado. Si la respuesta no está disponible, el esclavo envía un reconocimiento de la orden al nivel de enlace, lo que libera al maestro para realizar otras operaciones, hasta que envíe un mensaje de polling al esclavo del que quedó pendiente la respuesta.

El funcionamiento como maestro puede ser estacionario o puede ir alternándose entre los diferentes nodos. Esta alternancia se consigue con un esquema similar al de paso de testigo. Sin embargo, con dicho esquema se reduce la fiabilidad del sistema.

**DISPOSITIVO
ESCLAVO**

DISPOSITIVO ESCLAVO

Un esclavo sólo responde, no puede iniciar espontáneamente ninguna transferencia, sino únicamente responder a órdenes recibidas desde el dispositivo maestro.

EXTENSIÓN

EXTENSIÓN

Es un dispositivo secundario. El interfaz entre maestro o esclavo con su correspondiente extensión no forma parte de la especificación BITBUS

REPETIDORES

REPETIDORES

Nodo utilizado para regenerar la señal, sólo en modo autosincronizado (2 repetidores como máximo a 375kbits/seg. Y 10 repetidores a 62.5kbits/seg.) Empleados para aumentar la distancia entre nodos y en número de esclavos conectados al bus.

NIVELES DE ESPECIFICACIÓN

NIVELES DE ESPECIFICACIÓN

Si bien en la norma BITBUS nunca se hace referencia al modelo OSI, realmente se ajusta a él, aunque haciendo correr el nivel de aplicación directamente sobre el nivel de enlace, para evitar una sobrecarga de protocolos, tal como se muestra en la figura. Los niveles de especificación son los siguientes:

- < Nivel físico: RS-485
- < Nivel de Enlace: SDCL
- < Protocolo de mensajes
- < Acceso Remoto y Control

NIVEL FÍSICO: RS485

NIVEL FÍSICO: RS485

Los modos de funcionamiento en BITBUS son síncronos. Sin embargo, esto no impide que se puedan implementar los niveles superiores de enlace y aplicación sobre redes asíncronas, obteniendo una alta eficacia. El sincronismo se puede conseguir de dos formas: con un par de reloj aparte o incluyendo el reloj en la codificación de los datos (modo auto sincronizado)

Modo síncrono **Modo síncrono:** Optimiza la velocidad en distancias cortas. En este modo se pueden interconectar hasta 28 nodos en 30 metros y las velocidades de transmisión alcanzadas oscilan entre los 500knits/seg. y los 2.4Mbits/seg. Utiliza dos pares de señales diferentes, uno para los datos y otro para el reloj. Los datos siempre se toman con referencia al reloj, cambiando en el flanco de bajada y se muestrean en el flanco de subida.

Modo auto sincronizado **Modo auto sincronizado:** Permite mayores distancias entre los nodos a costa d velocidades de transmisión inferiores. Existen dos velocidades estándar de trabajo: 375kbit/seg. y 62.5kbits/seg. Soporta longitudes de segmento de hasta 300 metros a 375kbits/seg. Y de 1200 metros a 62.5kbits/seg. Cada segmento soporta hasta 28 nodos. Conectando segmentos mediante repetidores podemos conseguir hasta 250 nodos. En este modo también se emplean dos pares diferenciales de señales, uno para los datos y otro para el control de los repetidores. Los datos se codifican en NRZI (No Retorno a Cero Invertido), con lo que se combinan datos y reloj en el mismo par de control.

NIVEL DE ENLACE:

NIVEL DE ENLACE: SDLC

SDLC El protocolo de enlace de datos empleado en BITBUS es un subconjunto del SDLC (Synchronous Data Link Control) de IBM, aunque no tiene porque ser estrictamente compatible. Este nivel es el responsable del encapsulado de los mensajes en tramas y del control de la transferencia de las mismas a través del enlace de datos. La especificación define los conceptos de estado del sistema, número de secuencia, formato de trama y campos de control específicos.

Estado Del Sistema **Estado Del Sistema:** Se define desde el punto de vista del maestro. El maestro debe tener un pleno conocimiento y control de su propio estado. Este estado no afecta al funcionamiento de los esclavos, que por tanto no precisan de ninguna información sobre el mismo. Sin embargo, el estado de los esclavos si que debe ser conocido por el dispositivo maestro. El dispositivo esclavo puede encontrarse en dos modos diferentes:

- , NDM: Normal Disconnect Mode
- , NRM: Normal Response Mode

El esclavo entra en NDM tras un reset local, o cuando detecta un error de protocolo irreparable. El esclavo está a la espera de un orden especial del maestro para pasar a NRM. En este modo el esclavo no puede intercambiar mensajes con el dispositivo maestro. El esclavo entra en NRM sólo al recibir una trama especial por parte del maestro. Tras recibirla, el esclavo se considera sincronizado (contadores inicializados a cero) y ya puede intercambiar mensajes con el maestro.

Números De Secuencia **Números De Secuencia:** Se utilizan para evitar pérdidas o duplicación de tramas. Se utiliza un campo de tres bits para su codificación. Se utiliza un par de números de secuencia en el maestro y otro en el esclavo para cada enlace maestro-esclavo. Este par es:

- , Nr.- Número de secuencia de la trama que se espera recibir
- , Ns.- Número de secuencia de la última trama enviada

En cada transferencia efectuada en NRM, estos números se incluyen y verifican pudiendo obtener tres posibles resultados:

- < Secuencia correcta
- < Error de secuencia recuperable
- < Error de secuencia irreparable

El último de los casos obliga al maestro a resincronizarse con el esclavo y para ello éste pasa a NDM para luego volver a NRM. En cualquiera de los otros dos casos el esclavo permanece en NRM

FORMATO DE TRAMA

FORMATO DE TRAMA

La trama de mensaje se compone de entre 6 y 255 bytes. Todas las tramas tienen el siguiente formato:

Flag (7Eh)	Dirección Esclavo	Control	Información (Función/Datos)	CRC (16bits)	Flag (7Eh)
---------------	----------------------	---------	--------------------------------	-----------------	---------------

Flag **Flag:** Delimitan el comienzo y el final de la trama. Contienen la secuencia 01111110 que es irrepetible

Dirección **Dirección:** Contiene la dirección del esclavo involucrado en la transferencia. Es de 8 bits pero las direcciones 0 y de la 251 a la 255 están reservadas (ningún esclavo las puede tener asignadas)

- Control **Control:** Utilizado para el intercambio de información de control entre el maestro y el esclavo. Pueden ser de tres tipos:
- T Sincronización: Tramas no numeradas empleadas para sincronizar
 - T Supervisión: Tramas numeradas que permiten intercambiar información de estado
 - T Información: Tramas numeradas que permiten intercambiar mensajes de información
- Información **Información:** Campo destinado el mensaje BITBUS
- CRC **CRC:** Secuencia de Verificación de Trama (SVT), Código estándar CRC-CCITT de 16 bits, con polinomio generador:
- $$x^{16}+x^{12}+x^5+1$$

TIPOS DE TRAMAS

TIPOS DE TRAMAS

Existen fundamentalmente tres tipos de tramas que corresponden con los tres tipos de campos de control reconocidos:

- T No numeradas
- T Supervisión
- T Información

La trama contiene el mismo número de campos en los tres casos, pero lo que distingue el tipo es el campo de control según el cual podemos encontrar:

- Tramas No Numeradas **Tramas No Numeradas:** Empleadas para sincronizar los esclavos con el maestro. Reciben su nombre debido a que no llevan ninguna numeración de secuencia. Soporta dos órdenes no numeradas (SNRM y DISC) y dos respuestas no numeradas (UA y FRMR).

SNRM: Orden enviada por el maestro para sincronizar al esclavo. Si éste se encontraba en NDM pasará a NRM, pudiendo, a partir de ese momento, intercambiar mensajes con el maestro. Si el esclavo se encontraba ya en NRM ésta orden no es válida.

DISC: Orden enviada por el maestro para iniciar el procedimiento de sincronización con el esclavo, y provocar que éste pase o permanezca en NDM. El esclavo contesta al maestro con un reconocimiento no numerado UA, que confirma al maestro que el esclavo se encuentra en NDM. Esta orden la utiliza el maestro cuando necesita resincronizar (después de un reset, errores de secuencia irrecuperables, etc)

UA: Respuesta devuelta por el esclavo para reconocer la recepción de una orden no numerada válida en NDM

FRMR: Respuesta devuelta por el esclavo cuando detecta un campo de control inválido. Se emplea para contestar a cualquier trama no numerada cuando el esclavo se encuentra en NDM, o a cualquier trama de supervisión o de información cuando se encuentra en NRM, cualquier campo de control inválido o ante un error irrecuperable en los números de control de secuencia. Tras recibir esta trama el maestro debe iniciar una resincronización con el esclavo.

Tramas De Supervisión **Tramas De Supervisión:** Empleadas para intercambiar información de estado entre el nodo maestro y el esclavo cuando éste se encuentra en NRM. Empleadas por el maestro para sondear al esclavo (polling) y por el esclavo para reconocer una trama válida recibida del maestro. BitBUS Soporta dos códigos de supervisión:

RR: Empleados por el maestro para encuestar al esclavo y por el esclavo para reconocer la recepción válida de una trama de supervisión o de información previamente enviada por el maestro y para lo que no existe un mensaje de respuesta disponible (si existe un mensaje disponible, se envía una trama de información que llevará implícito el reconocimiento, salvo que la trama enviada por el maestro sea una RNR)

RNR: Indica que no hay buffers disponibles para recibir una trama. Utilizada por el maestro para encuestar al esclavo y obtener información sobre el enlace. El esclavo utiliza este tipo de tramas para indicar al maestro el reconocimiento de la última trama recibida pero advirtiéndole que no ha podido almacenarla (el maestro deberá retransmitirla)

Tanto RR como RNR contienen números de secuencia, el Nr reconocerá la recepción correcta hasta la trama Nr-1. El receptor de una trama de supervisión debe comparar siempre este valor con su Ns. Cuando el Nr recibido sea igual al Ns se suponen reconocidas todas las tramas enviadas. Si la trama enviada previamente era una trama de información, el Nr recibido debe ser igual a Ns+1, en cuyo caso nos ha reconocido la última trama enviada y el Ns puede ser incrementado y el buffer liberado. Si el Nr es igual al Ns supone que no se nos reconoce la última trama enviada y debe ser retransmitida. Cualquier otra relación Nr-Ns supone un error irrecuperable del protocolo que requerirá una resincronización entre el maestro y el esclavo.

Tramas De Información **Tramas De Información:** Utilizadas para la transferencia de mensajes. Son las únicas tramas que disponen de campo de información, que contendrá el mensaje BITBUS, y de número de secuencia de transmisión. Las tramas de información son un subconjunto de las tramas de supervisión. El mecanismo de reconocimiento de tramas vía Ns y Nr es idéntico al descrito anteriormente.

A continuación exponemos una tabla con el resumen de las tramas anteriores:

Función	Tipo de Trama	Código	Descripción
Modo de respuesta normal (SNRM)	Control M > E	93h	Se ordena al esclavo pasar al estado de respuesta normal (NRM)
Desconexión (DISC)	Control M > E	53h	Se ordena al esclavo pasar al estado de desconexión (NDM)
Reconocimiento no numerado (UA)	Control E > M	53h	Respuesta del esclavo indicando que ha recibido una orden válida estando en estado de desconexión.
Trama rechazada (FRMR)	Control E > M	97h	Respuesta del esclavo indicando que ha recibido una orden incorrecta estando en estado normal (NRM)
Consulta o test (polling)	Supervisión M > E	RRR10001 (binario)	RRR=n° de secuencias anteriores recibidas como correctas El maestro indica que el esclavo debe contestar como consulta o test, respondiendo con RR o RNR.
Receptor dispuesto (RR)	Supervisión E > M	RRR10001 (binario)	RRR=n° de secuencias anteriores recibidas como correctas Respuesta del esclavo indicando que la consulta es correcta y que no tiene datos que transmitir.
Receptor no dispuesto (RNR)	Supervisión E > M	RRR10101 (binario)	RRR=n° de secuencias anteriores recibidas como correctas Respuesta del esclavo indicando que la consulta es correcta pero que no puede aceptarla por tener el buffer lleno.
Información (I)	Información M > E E > M	RRR1EEE 0 (binario)	RRR=n° de secuencias anteriores recibidas como correctas EEE=n° de secuencias enviadas Tramas normales de intercambio de información. En cada trama se comprueba los números de secuencia para evitar duplicidad de mensajes y detectar la pérdida de tramas o errores de transmisión.

TRAMA DEL CAMPO
DE INFORMACIÓN

TRAMA DEL CAMPO DE INFORMACIÓN

El campo de información de la trama obedece a un formato que especifica un mínimo de 7 bytes y un máximo de 250 (aunque se acepta como dispositivo estándar aquel que permite entre 7 y 20 bytes de longitud). Los bytes se corresponden con el siguiente formato:

Byte	Descripción
1	Longitud de información
2	MT - SE - DE - TR - 4 bits de reserva
3	Dirección esclavo
4	Tarea fuente - Tarea destino
5	Tareas usuario / Errores
6 ... 250	Otros datos (Mínimo 2 bytes)

Donde:

- i **Longitud de información:** Indica la longitud del campo de información (entre 7 y 250).
- i **Tipo de mensaje (MT):**
 - i 0 = orden
 - i 1 = respuesta
- i **Fuente de la orden (SE):** Indica si la fuente de una orden o el destino de una respuesta es:
 - i 0 = bloque maestro
 - i 1 = bloque de expansión E/S de éste
- i **Destino de la orden (DE):** Indica si el destino de una orden o la fuente de una respuesta es:
 - i 0 = bloque esclavo
 - i 1 = bloque de expansión E/S de éste
- i **Pista (TR):**
 - i 0 = mensaje de maestro a esclavo
 - i 1 = respuesta de esclavo a maestro
- i **4 bits reservados:** Se reservan para futuras ampliaciones, deben ser puestos a 0 al enviar el mensaje.

- i **Dirección esclavo:** El valor debe estar comprendido entre 1 y 250
- i **Codificación de tareas:** Este byte se compone de dos grupos de 4 bits:
 - i *Tarea fuente (4 bits altos):* las tareas 12 a 15 están libres y pueden ser definidas por el usuario. La tarea 0 está reservada a los servicios RAC (ver tabla adjunta) y las tareas 1 a 11 están reservadas por Intel.
 - i *Tarea destino (4 bits altos):* definibles por el usuario (salvo que la tarea fuente sea la 0).

La tarea 0 o RAC (Remote Acces and Control) proporciona una serie de funciones que todo dispositivo BITBUS debe soportar:

Código	Tipo	Tarea
00h	Control	Reset esclavo
01h	Control	Llamar tarea programada en un esclavo
02h	Control	Finalizar tarea
03h	Control	Obtener identificador/puntero de la función.
04h	Control	Habilitar/deshabilitar el resto de tareas (excepto RAC) en un esclavo.
05h	Acceso	Leer 1 byte de E/S
06h	Acceso	Escribir 1 byte de E/S
07h	Acceso	Actualizar líneas de E/S
08h	Acceso	Leer n bytes de la memoria del esclavo
09h	Acceso	Grabar n bytes en la memoria del esclavo
0Ah	Acceso	Operación O (OR) con byte de E/S
0Bh	Acceso	Operación Y (AND) con byte de E/S
0Ch	Acceso	Operación O-exclusiva (XOR) con byte de E/S
0Dh	Acceso	Leer registro de estado
0Eh	Acceso	Escribir registro de estado
0Fh- BFh	Reservado	Funciones reservadas por INTEL
C0h- FFh	Reservado	Funciones definibles por el usuario

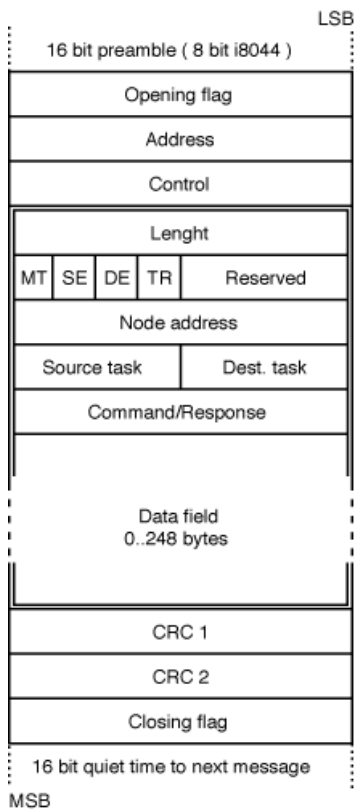


Figura 6.7 - Estructura general de un mensaje en BitBus

- i **Tareas usuario/Errores:** Este byte, en los mensajes de órdenes de maestro a esclavo, contendrá los datos necesarios para las subfunciones definidas por el usuario. En los mensajes de respuesta contendrá los códigos de error.

La siguiente tabla muestra los códigos de error que contempla el estándar y señala aquellos definibles por el usuario:

Código	Error
00h	No hay error
01h - 7Fh	Errores definidos por el usuario
80h	No se encuentra la tarea destino
81h	Imposible iniciar tarea debido a que se está realizando otra
82h	Imposible crear tarea por falta de banco de registros
83h	Solicitud de tarea ya activa
84h	Imposible iniciar tarea por falta de memoria
85h - 90h	Reservado por Intel
91h	Error de protocolo
92h	Reservado por Intel
93h	El nodo de destino no responde
94h	Reservado por Intel
95h	Tareas deshabilitadas por RAC
96h	Orden RAC desconocida
97h - FFh	Reservado por Intel

- i **Datos:** Estos bytes están libres para uso general, pero al menos deben existir los bytes 6 y 7 en cualquier trama que tenga campo de información.

CONCLUSIÓN DE BITBUS

CONCLUSIÓN DE BITBUS Ya que conocemos la estructura interna de BitBus, los códigos, las tramas de datos y para que sirve cada una de ellas, podremos entender mejor el diagrama general de la comunicación en BitBus y que se muestra en la Figura 6.7 de la página anterior. También sabemos que podemos aplicar BitBus en la industria, así como lo muestran los de BEUG (Bitbus European Users Group):

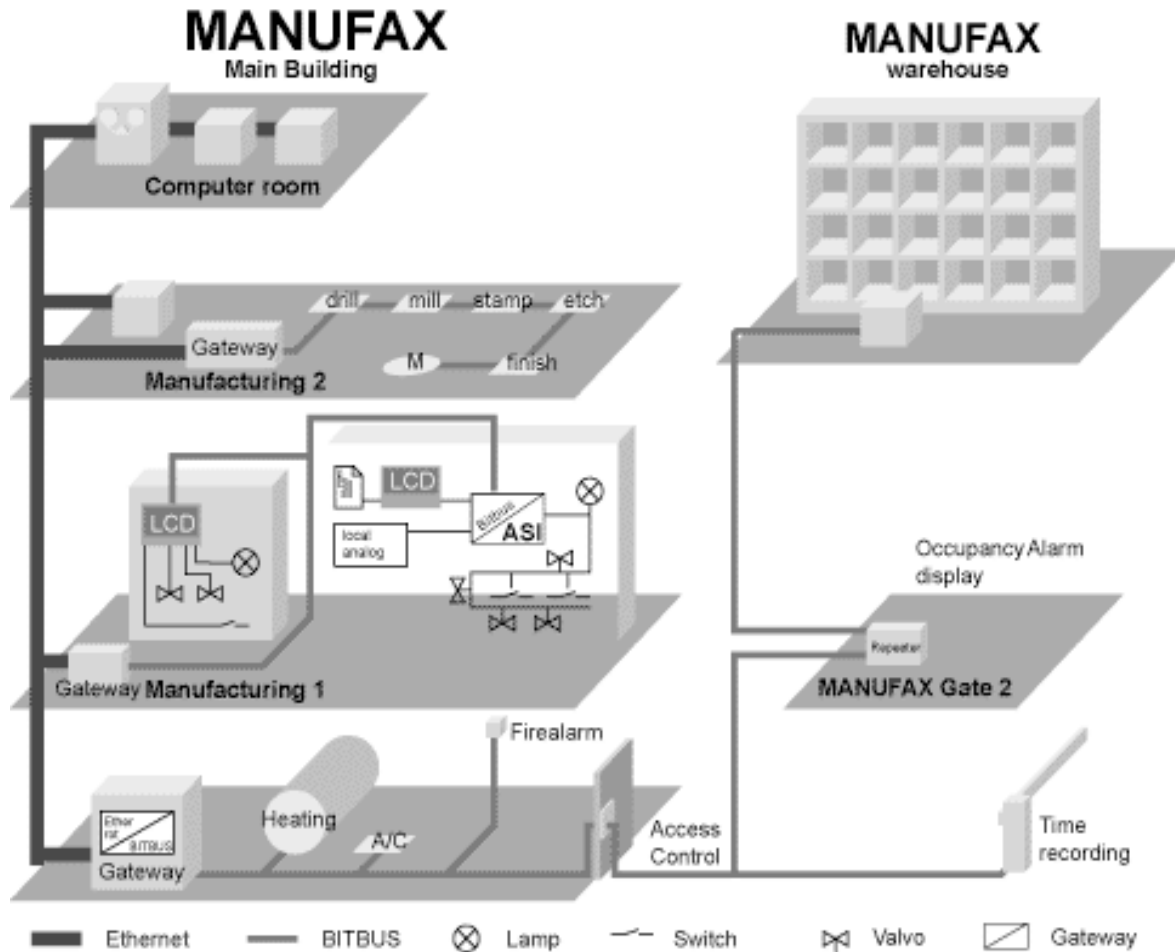


Figura 6.8 - Aplicación de BitBus en una planta de manufactura, se puede observar como integran Ethernet para la red de usuarios y Bitbus para la red industrial. Gráfica tomada de la página de BEUG

BIBLIOGRAFÍA

- [1] Jan Axelson; **SERIAL PORT COMPLETE**; Lakeview Research; ISBN 0-9650819-2-3
- [2] Sara Blanc Clavero, Rafael Ors Carot; **PRÁCTICAS DE REDES INDUSTRIALES**. Universidad Politécnica de Valencia; ISBN 84-9705-019-3
- [3] RS-232 **HISTORIC OVERVIEW**; <http://www.raduniversity.com/networks/1995/rs232/hist.htm>
- [4] IBM; **THE IBM MAINFRAME, BULDING ON DE PAST, DEFINING THE FUTURE**; <http://www-8.ibm.com/servers/eserver/id/zseries/about/timeline/1960>
- [5] **E S T Á N D A R D E COMUNICACIONES RS-232C (O CÓMO FUNCIONA UN PUERTO S E R I E . . .)** ; <http://www.euskalnet.net/shizuka/rs232.htm>
- [6] Héctor Francisco Rentería Toledo, Ignacio Murillo Tornero, Daniel Castell I b a r r a ; **RS - 2 3 2** ; <http://mx.geocities.com/pcmuseo/mecatronica/puertors232.htm#BA>
- [7] Jan Stanek; **RS 485 & RS 422**; <http://www.hw-server.com/docs/rs485.html>
- [8] B&B Electronics Ltd; **RS422/485 APPLICATION NOTE** ; http://www.bb-elec.com/tech_articles/rs422_485_app_note/overview.asp
- [9] Wikipedia, La enciclopedia libre; **PROTOCOLO DE RED**; <http://es.wikipedia.org/wiki/Protocolo>
- [10] Wikipedia, La enciclopedia libre; **TOKEN BUS** ; http://es.wikipedia.org/wiki/Token_Bus
- [11] Wikipedia, La enciclopedia libre; **TOKEN RING** ; http://es.wikipedia.org/wiki/Token_Ring
- [12] Wikipedia, La enciclopedia libre; **802.5**; <http://es.wikipedia.org/wiki/802.5>
- [13] Miguel Angel Montejo Ráez; **BITBUS**; <http://www.automatas.org/redes/bitbus.htm>
- [14] Miguel Angel Montejo Ráez; **BITBUS: ESTRUCTURA FÍSICA Y PROTOCOLO** ; http://pomelo.ivia.es/mecanizacion/www/Manual_Electronica/redes/bitbus2.htm
- [15] BEUG; **BITBUS SPECIFICATION OVERVIEW**; <http://www.bitbus.org>