

METRAJE CINEMATOGRAFICO

Estas tablas muestran los metros de película que corresponden a determinadas duraciones en minutos y a un ritmo de 24 fotogramas por segundo. La equivalencia para el formato de 35 mm. son 52,6 fotogramas por metro y para el formato de 16 mm., 132 fotogramas por metro.

mn	35 mm.	16 mm.	mn	35 mm.	16 mm.	mn	35 mm.	16 mm.	mn	35 mm.	16 mm.
1	27,56	10,97	16	437,76	175,55	31	848,16	340,13	46	1258,56	504,71
2	54,72	21,94	17	465,12	186,52	32	875,52	351,10	47	1285,92	515,68
3	82,08	32,92	18	492,48	197,50	33	902,86	362,07	48	1313,28	526,66
4	109,44	43,90	19	519,84	208,47	34	930,24	373,05	49	1340,64	537,63
5	156,60	54,06	20	547,20	219,44	35	957,60	384,02	50	1368,00	548,60
6	164,16	65,84	21	574,56	230,41	36	984,96	395,00	51	1395,36	559,57
7	193,52	76,01	22	601,92	241,38	37	1012,32	405,97	52	1422,72	570,54
8	218,83	87,76	23	629,28	252,36	38	1039,60	416,94	53	1450,08	581,51
9	246,24	98,75	24	656,64	263,33	39	1067,04	427,91	54	1477,44	592,49
10	273,60	109,72	25	684,00	274,30	40	1094,40	438,38	55	1504,80	603,46
11	300,96	110,69	26	711,36	285,27	41	1121,76	449,85	56	1532,16	614,43
12	328,32	131,66	27	738,72	296,24	42	1149,12	460,82	57	1559,52	625,40
13	355,68	142,64	28	766,00	307,22	43	1176,48	471,79	58	1586,88	636,38
14	383,04	153,61	29	793,44	316,19	44	1203,84	482,77	59	1614,24	647,35
15	410,40	164,58	30	820,80	329,16	45	1231,20	493,74	60	1641,60	658,32

NEURON

Mezcla de hard y firmware (8K ROM) en un solo chip. Protocolo implantación más que protocolo especificación. Neuron cuenta con varios watchdog timers, autodiagnósticos, búsqueda de corrupción en la memoria, librerías run time, 3 tipos de memoria, 35 tipos de controladores y 48 bits para fijar direcciones de nodos exclusivos. No emplea checksum ni paridad, sino el completo 16 bit error polynomial y adicionalmente forward error correction, capaz de detectar y corregir errores de simple bit sin retransmisión. (para escasos anchos de banda con mucho ruido y atenuación). Lon Talk emplea una técnica llamada CSMA para predecir y evitar problemas de colisiones y prioridades. Neuron se programa en Neuron C e introduce eventos y su orden de ejecución, 37 adicionales data types, 35 I/O objetos y 2 timers objetos y mensajes integrales (variables de red).

Disponible en varios modelos que se diferencian en la configuración de memoria; al menos 0.5K de EEPROM, 1K de RAM, ROM y posibilidad de memoria externa; todo en único voltaje. Puerto transceptor para la red, incluyendo manchester phase encoded modulation hasta 1,25 M en single-ended y diferencial con histeresis y filtro programable para impedancia dinámica, 20mA señal.

Hard y Soft para construir el chip con librería de 35 opciones, incluyendo 11 canales I/O sin multiplexar, controladores de triacs, pulsos y moduladores. Quadrature inputs, puerto serie y paralelo de alta velocidad, lector de barras y magnético, doble convertidor A/D, etc.

Diagnósticos de entrada y salida, reset, reloj (normalmente 624 K y hasta 10M). Alimentación a 5 V. con posibilidad de aletargamiento a 15ma.

Lon talk reside típicamente en PROM externa (firmware). El transceptor puede incluir transformador / acoplador

El coste de un equipo de desarrollo para nodos puede costar sobre 4.000 y sobre 25.000 para poder montar y mantener una red.

Lon Works dispone de transceptores adaptados para variedad de voltajes y formatos: transformador de acoplamiento twisted par (1,25 Mb), cualquier topología twisted par con o sin enlace de fuerza, spread spectrum power line (FCC & CENELEC compliant), power line de banda estrecha, radio (2,4 GHz spread spectrum, 900 MHz, 450, 49, etc.), cable coaxial (con posibilidad de video multiplexado), fibra óptica, infrarrojos, etc.

Todo con adaptación plena adaptación a velocidades, ruteo, segregación de tráfico, mezcla de medios, etc.

Además de Adaptadores para PC via RS-232: Ethernet, T1, X.25, Bitbus, Profibus, CAN, Modnet, SINEC, Grayhill, Opto 22 (digital), OptoMux, Modbus, ISA bus, STD 32 bus, PC/104 bus, VME bus, EXM bus, etc.

La conexión con Windows es perfecta como soporte para confección de programas de red o como herramienta para la construcción programación de neuron y mantenimiento. Son posibles API y Microsoft Dynamic Data Exchange. Gracias a Lon Builder muchos trabajos ya pueden estar hechos o poder remodelarlos a otra necesidad, abajando costes y ahorrando tiempo.

C-BUS (CEBUS o Consumer Electronic Bus)

La idea surge en 1984 por miembros de la EIA (Electronics Industry Association). En 1992 se presenta como standard de arquitectura abierta. Puede correr en power line wires, low voltaje twisted pairs, coaxial, IR, RF y fibra óptica. Obliga a direccionar cada device por hard y en fábrica (4 billones de posibilidades). Permite definir un lenguaje orientado a objetos que incluye comandos como volumen, rebobinado, pausa, salto, temperatura, on, off, dim, bright, all light on, all units off, etc. El hard para las comunicaciones, el lenguaje y el protocolo están disponibles en un chip producido por Intellon Corporation (Florida) para uso e implementación en máquinas de cualquier fabricante. Intellon otorga a cada cliente una etiqueta privada y OEM oridictis ysubg tge CBUS standard y un quit de desarrollo.

Comercialmente está disponible a través del grupo creado por la EIA para ello, la CIC (C-bus industry Council) donde hay muchos representantes internacionales (por ejemplo: Sony, Compaq, Ibm, Microsoft, Bell, Panasonic, etc.). La CIC ha desarrollado un laboratorio de pruebas para la certificación de productos. Sólo se paga cuando se consigue la certificación y se otorga la licencia y el derecho de mostrar el logotipo en el producto.

El espectro extendido incluye comenzar la modulación a una frecuencia y alterarla durante su ciclo. El power CBUS standar comienza cada estallido a 100Khz. e incrementa linealmente hasta 400 Khz. durante una duración de 100µsg. Ambos estallidos (referidos a un estado superior) y la ausencia de estallido (referido a un estado inferior) crea dígitos similares, así que no es necesario ninguna pausa entre ellos.

El dígito 1 es creado por un estado inferior o superior dura 100µsg. El dígito 0 es creado por un estado inferior o superior dura 200µsg.

Por tanto la velocidad de transmisión es variable, dependiendo de la cantidad de caracteres 0 y 1; aunque de promedio son 7,500 bits por segundo.

Las transmisiones CBus están basadas en paquetes de datos que también varían en longitud, dependiendo de cuántos datos están incluidos. Algunos paquetes pueden tener cientos de bits de longitud. El tamaño mínimo de paquete es 64 bits, el cual tiene una velocidad media de 7500 bits por segundo y puede tomar 1/117 parte de un segundo para ser transmitido y recibido.

X MIDI (XM o eXtended MIDI)

El MIDI que habitualmente conocemos, para un nivel de equipamiento medio / alto no tiene competencia, pero para altas prestaciones hay que buscar una solución a sus limitaciones. En opinión del desarrollador de X MIDI, el MIDI habitual resulta lento por el software y hardware MIDI, sin grandes pretensiones, que montan las máquinas; tratando el MIDI como algo simple, sin buscarle una explotación máxima a nivel de prestaciones. X MIDI pretende elevar el rendimiento básico del MIDI para aplicaciones de precisión.

La idea parte de Eric Likac-Kuruc que ha trabajado en Steinberg, Sirzweil, Oberheim, E-mu y Lexicon; y la patente es de la firma Digital Design & Development (DDD) (empresa e inventor son de Bélgica). Desde el año 1992 se promociona comercialmente su desarrollo y son más de 100 las empresas que lo implantan; entre otras, Akai, Allen & Heath, Bohm, C-lab, Emagic, Fpstex, General music, Kurzweil, Lexicon, Mackie Designs, MIDIMAN, Mark of the Unicorn, Oberheim, Steinberg, Penny & Giles, etc.

Su implantación es completamente transparente a los equipos y cableado habituales, permitiendo la convivencia de máquinas de los 2 tipos en un mismo sistema; no se aprovecharán las ventajas, pero la compatibilidad permanece. No requiere ningún interface, las conexiones son las mismas. No usa otros mensajes, ni más largos, ni aumenta el ancho de banda. Sólo modifica la electrónica en los puertos de los aparatos y el software de control. Se basa en elementos triestado (con valores 0, +1, -1).

XM está patentado, pero puede ser usado libre y gratuitamente como protocolo; lo que sí hay que comprar (a nivel de fabricantes de aparatos) es el chip (desarrollado por DDD) que sustituye las UART y optoacopladores habituales y que se vende a los fabricantes por un módico precio (incluso quizás más barato que la suma de los componentes habituales). Hay varios modelos y soft emulador para PC. Consiste en un hard-chip / soft a 32 bits y se basa en un procesador RISC dedicado que incluye en un VLSI un conjunto de instrucciones XMIDI que codifican y decodifican las palabras XM ternarias en mensajes MIDI. Incluye un diodo para proteger la polaridad inversa, el flujo de la señal es por lazo de corriente y ocupa igual ancho de banda que MIDI; sin embargo, las UART habituales no pueden leer voltajes triestado, por eso XM precisa un chip. También cuenta con auto setup, donde los modelos y capacidades de los aparatos pueden ser reconocidos de forma automática. esto alivia de programación, cableado

- La iluminación, que es un campo secundario en MIDI, tiene valor propio en XM, que cuenta con mensajes específicos para canales,, intensidad, efectos, focus, elevation, rotation, etc. Hasta 324 devices en comunicación bidireccional con un solo cable.
- El habitual MIDI byte (binario) ahora es llamado XM Word (ternario). XM puede expresar hasta 13122 valores ($3^n * 2^{(m-n)} * 2^p$) con un solo byte, como la transmisión de cada bit dura el mismo tiempo, esto se traduce en un incremento de velocidad sin aumentar el ancho de banda, ya que para un cambio de programa más Bank select solo necesitamos un byte. para mandar 2000 notas de polifonía. En volcados de muestras, con solo 2 bytes podemos mandar una resolución de 24 bit. Ni técnicos ni programadores conocen bien que se transmite por la línea telefónica, la gran sorpresa es que XM se transmite perfectamente con su sistema triestado... aunque el análisis de la señal en un osciloscopio resulte extraño.
- 324 canales MIDI sobre cada puerto en comunicación bidireccional.
- 753.145.430.616 presets direccionables usando bancos.
- 1,5 trillones de controladores.

- 510 valores de resolución en los mensajes (velocidad, etc.) (en mesas de mezcla 0,125 db; en contra de los 0,5 de midi) (más frecuencias tonales)
- 4374 valores de resolución en valores no lineales (cambio de programa, controladores, etc.).
- 2611 instrucciones básicas.
- La velocidad de transmisión es dinamica, adecuandose de forma automatica a las circunstancias.
- Para el programador, la implementación de mensajes MMC y MSC es más sencilla en XM, pues el chip hace la conversión.

Como rectora de la norma MIDI, la MMA opina sobre XMIDI que, éste, no será sustituto ni ampliación del actual MIDI, como estandar. Aunque conoce y consiente la fabricación de aparatos XMIDI, no la apoya por varios motivos fundamentales:

- La actual tecnología MIDI es gratuita y basada en el consenso y la colaboración. No es admisible un hardware personalizado e impuesto.
- El interface XMIDI dificulta la confección del software y puede causar confusión, dada la cantidad de fabricantes y la difusión del protocolo.
- El aumento de velocidad de transmisión no resulta aconsejable ya que podría dar lugar a problemas por excso de densidad de datos.
- Para la mayoría de los actuales usuarios, el actual MIDI es suficiente y sencillo; además el futuro actual es el alto volumen de expansión del protocolo en el medio informatico de consumo y por tanto, debe ser sencillo y barato de implementar.

OSI

Sobre el año 1980, la Organización Internacional de Standares (ISO), el Comité Consultivo Internacional Telefónico y Telagráfico (CCITT) y otras organizaciones internacionales, comenzaron el diseño de un Modelo de Referencia para la Interconexión de Sistemas abiertos (OSI), un marco que pretende ordenar y estandarizar todos los aspectos relacionados con la comunicación de datos entre maquinas. El modelo OSI no describe ningun protocolo, sólo describe el comportamiento de un sistema de comunicación, pero no su construcción interna. El OSI se desarrolla paralelamente en el tiempo a Internet.

OSI establece 7 niveles para abordar una comunicación, estableciendo para cada nivel cometidos específicos. Cada nivel hace uso del trabajo que ha hecho su nivel inmediatamente inferior y a su vez servira de apoyo para el inmediatamente superior. Cada nivel asume que el trabajo hecho por sus niveles inferiores es el correcto, Esta estratificación de funciones sirve como mecanismo de abstracción y concentra a cada nivel en su tarea concreta. La elección de 7 niveles es arbitraria; algún nivel realiza cometidos muy simples y otros podrian desdoblarse en varios.

Los niveles del OSI:

- 7 Nivel de Aplicación (A): Ofrece a l usuario final todos los servicios de la red y aplicaciones para su acceso y uso sencillas y agradables de manejar.
- 6 Nivel de Presentación (P): Ofrece la posibilidad de intercambiar estructuras complejas de datos por la red conservando su significado
- 5 Nivel de Sesión (S): Aporta mecanismos de control y sincronización al transporte de los datos.
- 4 Nivel de Transporte (T): Transfiere los datos, libres de errores, entre maquinas; independientemente de la tecnología empleada en la red subyacente.
- 3 Nivel de Red (N):
- 2 Nivel de Enlace de Datos (DI):
- 1 Nivel Físico (Ph):

Podemos agrupar los niveles anteriores de la siguiente forma:

- Servicios de Usuario (niveles 5, 6 y 7): recogen las funciones orientadas al usuario.
- Nivel de Transporte (nivel 4): Orece a los servicios de usuario una visión homogenea de lared, independientemente de la tecnologia empleada; lo que facilita el desarrollo de aplicaciones, etc.
- Sub red de Comunicaciones (niveles 1, 2 y 3): Tecnologia de la red; su misión es transportar octetos sin errores y fiablemente.

OSI tambien establece como será el entendimiento entre niveles adyacentes, estableciendo el uso de Servicios. Cada nivel (ejerciend de Usuario), puede solicitar Servicios a su nivel inmediatamente inferior (su Proveedor); igualmente cada Proveedor informa a su Usuario acerca de las ocurrencias de su interes. Se establecen 4 categorias de Servicios, cada una con sus Primitivas:

- Servicios Confirmados
- Servicios No Confirmados
- Servicios Confirmados por el Proveedor
- Servicios iniciados por el Proveedor

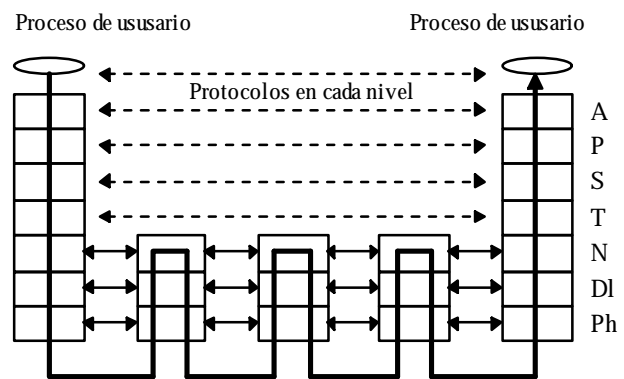
Protocolos. Un protocolo de nivel n describe la forma en que 2 entidades pares de ese nivel se intercambian información (en forma de n(PDUs)) con el objeto de ofrecer a sus usuarios (entidades de nivel n+1) el servicio de nivel n. La definición de un protocolo de nivel n, incluye:

- Como se utilizan los servicios de nivel n-1 para realizar el intercambio de (1) PDUs.
- Qué PDUs se intercambian, qué misión tienen y cómo están codificadas.
- Cómo es el dialogo, las reglas de comportambineto de las entidades.

La última parte suele describirse en forma de una máquina de estados (FSM): el automata que describe el protocolo parte de un Estado Inicial; la ocurrencia de un Suceso en un determinado Estado, provoca la ejecución de una Acción y el cambio a otro Estado.

ISO y CCITT han desaroollado lenguajes formales para la especificación de protocolos: Lotos, Estelle y SDL. La especificación completa de un protocolo es muy compleja.

Comunicación extremo a extremo. La comunicación entre 2 procesos ubicados en maquinas remotas supone la interacción de multiples pares de entidades, no solo en los extremos comunicantes sino que muy a menudo, intervienndn elemntns intermedios, nodos, que hacen posible la comunicación remota o anidadda. El proposito de los nodos es permitir la interconexión de sistemas finales; enb ellos no se ejecutan procesos de usuario por lo que no necesitan entidades de todos los niveles. Además los usuarios finales no deben percatarse de su existencia. En el dibujo, 2 máquinas se comunican a través de 3 nodos.



Parámetros, SDUs y PDUs: Asociados a cada primitiva tenemos una serie de parametros que aportan información complementaria para la operación n solicitada.

- PDI (Protocol Control Information). Información de control relacionada con el protocolo e invisible para los usuarios.
- SDUs (Service Data Units). Unidad de servicio, es cada bloque de datos que un usuario entrega a su proveedor (con la intención que sea entregado al usuario final).
- PDUs (Protocol Data Unit). Es cada una de las unidades de información que se intercambian como parte del protocolo. Normalmente, una PDU contiene una SDU + PCI. Si una SDU es muy

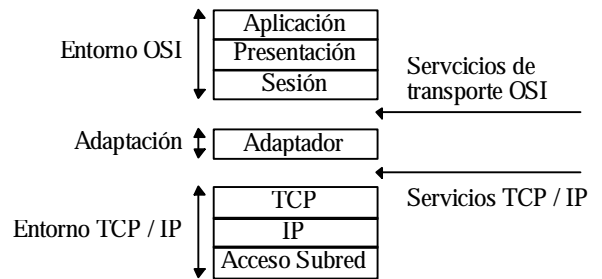
grande, se fragmenta y viaja en varias PDU, posteriormente se reensambla. Este proceso es parte del protocolo.

- Puesto que en una comunicación intervienen todos los niveles, cada uno de ellos incorporará su propia información de control; así, conforme la información desciende de nivel, la adición de PCIs acaba formando Tramas (PDUs del nivel de enlace de datos) de la siguiente forma:

Establecer una arquitectura por niveles plantea el problema de utilizar muchos recursos para mantener el protocolo.

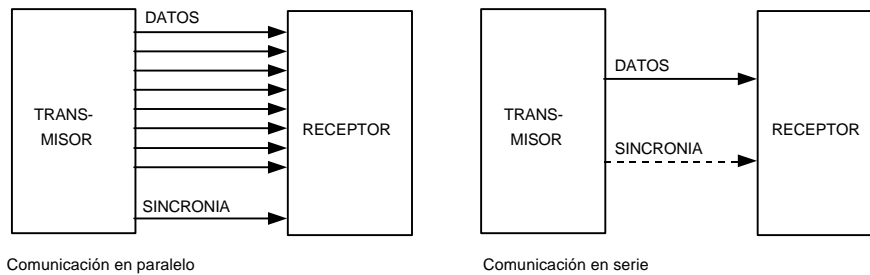


Redes mixtas. Una ventaja de las arquitecturas por niveles es que algunos de ellos realizan funciones similares en redes distintas y permite establecer fácilmente redes mixtas. Por ejemplo, TCP/IP ha demostrado su fiabilidad y flexibilidad trabajando a niveles de red y OSI se ha planteado como meta estratégica de futuro en muchas organizaciones, debido a la riqueza de sus niveles superiores. Así pues, y dada su similitud, podemos establecer una red mixta sin más que añadir un adaptador de interface y confeccionar el protocolo adecuado.



PROTOSCOLOS DE COMUNICACION: Cuando queremos conectar una máquina con otra hemos de sacar el Bus de datos al exterior y esto no lo podemos hacer directamente por 2 motivos:

- Seguridad: Si una anomalía ocurriera en el exterior, toda la máquina se vería afectada y debemos tomar alguna medida que, en parte, proteja el bus interno del externo.
- Entendimiento. Las 2 máquinas no tienen por que ser iguales, ni siquiera utilizar el mismo microprocesador. Aunque los datos para trabajar sean los mismos, el modo de trabajo y los relojes internos no y entonces han de ponerse de acuerdo en como se va a desarrollar la transmisión.



Un protocolo de comunicación es un convenio universal que dicta unas normas hardware y software concretas (características eléctricas, mecánicas y funcionales de las señales), para que la transmisión entre aparatos sea segura y entendible por cualquier máquina que cumpla el protocolo. Básicamente, hay 2 modos de establecer una comunicación: paralelo y serie.

Por otra parte, atendiendo al sentido de transmisión, pueden establecerse 3 modos de comunicación:

- Simplex: Transmite información en un solo sentido. Es el caso del MIDI.
- Semiduplex (half-duplex): Transmisión en ambos sentidos, pero no simultáneamente.
- Duplex (full-duplex): Transmisión simultánea en ambos sentidos.

PROTOSCOLO DE COMUNICACIONES PARALELO: Consiste en sacar todo el Bus de Datos completo al exterior por medio de una unidad adaptadora y de protección (PIA, u otro elemento) y también alguna señal de control y sincronía adicional. La transmisión se efectúa palabra a palabra, en toda su longitud.

Su ventaja es ser una transmisión muy rápida (de una sola atacada envía toda la palabra completa), pero necesita muchos cables (uno para cada bit de la palabra y alguno extra para control) y es propensa a interferencias y diafonía (permite cables de 3 metros de longitud como máximo).

PROTOSCOLO DE COMUNICACIONES SERIE: El Bus de Datos sale al exterior, pero no todos sus bits simultáneamente, sino uno detrás de otro, en serie en lugar de en paralelo. Como el Bus de Datos de los sistemas microprocesados es paralelo, se necesita un dispositivo (UART, ACIA, SSDA, etc.) que realice un proceso de Serialización de los bits de cada palabra antes de que salgan al exterior. A su vez, el mismo dispositivo, realizará en el receptor un proceso inverso, de Paralelización, para obtener nuevamente la misma palabra, con sus bits en paralelo.

Pero si mandamos varias palabras, una detrás de otra, nos vamos a encontrar, en la transmisión, con una cadena de bits muy larga e ininterrumpida, planteándose los problemas de como averigua el receptor con que bit acaba una palabra y con cual empieza la siguiente (Sincronización de palabra) y donde acaba un bit y empieza el siguiente (Sincronización de bit).

Sincronización de mensaje es el conjunto de normas, parte del protocolo, establecidas para superar estas dificultades de la transmisión serie para que el conjunto de palabras se entienda correctamente.

Para superar los problemas de sincronización de bit y de palabra tenemos 3 soluciones, motivando, cada una de ellas, un tipo de comunicación distinto:

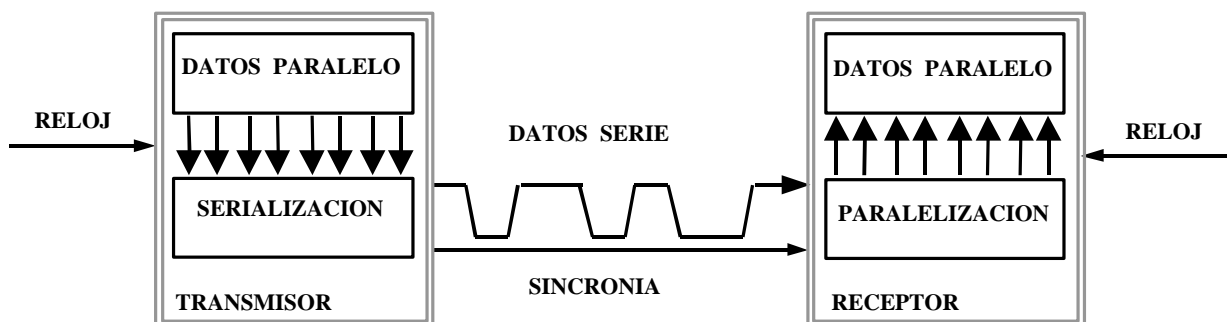
- Enviar por una línea distinta a la de datos, la señal del reloj que hace la serialización para que indique los centros de cada bit: comunicación Sincronía.
- Lograr que los relojes de transmisión y recepción se mantengan continuamente síncronos por algún procedimiento, generando el receptor su propio reloj sincronizado con el de transmisión: Asíncrono.
- Enviar junto a cada bit y por la misma línea, información extra que permita extraer la señal de reloj: Auto Reloj.

Por otra parte cada protocolo serie, en particular, marcará unas determinadas especificaciones eléctricas a la transmisión partiendo, básicamente, de uno de los siguientes modelos:

- Señales TTL: Conexión directa entre puertas TTL. (máximo 5 metros de longitud).
- Bucle de corriente: Los niveles se codifican por la ausencia o presencia de una corriente unidireccional de 20 mA. en la línea. (10 bps: 300 m / 1200 bps: 30 m.). Es el caso del MIDI.
- RS-232-C: Tensión simétrica con valores comprendidos entre ± 3 V. y ± 25 V. (habitualmente ± 12 V.). Los flancos de subida de la señal se limitan a 30 V/ μ sg. (Velocidad máxima: 20 Kbps y 15 m.).
- RS-422: Utiliza emisores y receptores balanceados (diferenciales) y tensiones de hasta 6 V. El receptor tiene un umbral de disparo de 200 mV. (100 Kbps: 1200m. / 10 Mbps: 12m.).
- RS-423: Utiliza receptor diferencial y emisor no diferencial. Así permite su interconexión con emisores o receptores de las clases anteriores. (3 Kbps: 1200 m / 300 Kbps: 12 m.).

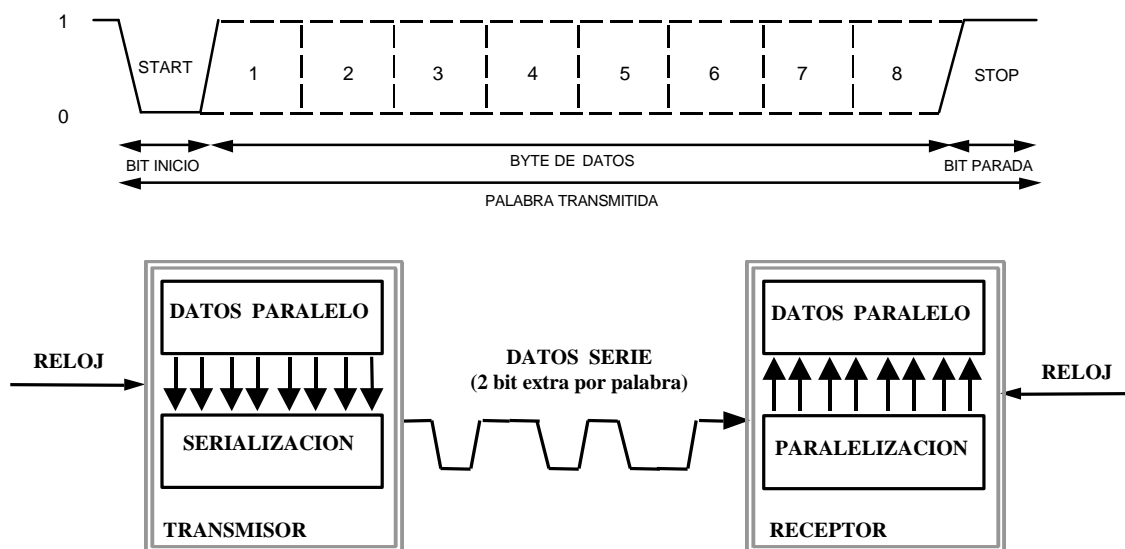
Las ventajas de cualquier transmisión serie son la de requerir pocos cables para establecer la comunicación (en principio solo 2) y también que es menos propensa a interferencias que la paralelo (permitiendo mayor longitud). Su inconveniente es la lentitud, ya que se envían los bits de uno en uno, en lugar de hacerlo en bloque.

PROTOCOLO DE COMUNICACIONES SERIE SINCRONO: Además de los cables para transmitir los bits en serie, utiliza uno extra para conectar el reloj de Serialización del transmisor con el de Paralelización del receptor. De esta manera ambas máquinas trabajan en sincronía perfecta a la hora de identificar los comienzos y finales de bit y, por conteo de bits, los comienzos y finales de cada palabra transmitida. Además, se añaden en cada bloque de datos a transmitir 2 bits extra, SYN, al comienzo de la transmisión de cada uno.



PROTOCOLO DE COMUNICACIONES SERIE ASINCRONO (es el usado por el MIDI): Cuando el transmisor serializa la palabra binaria, añade un bit extra al comienzo de la misma (Bit de Start, con valor binario 0) y otro al final (Bit de Stop, con valor binario 1). Proceso detallado de la transmisión:

- En ausencia de transmisión la línea debe encontrarse en nivel alto de tensión (valor 1).
- El receptor aguarda y detecta la transición de la línea de 1 a 0 (causada por la llegada del bit de Start).
- El receptor pone a cero su reloj y lo activa, a la frecuencia dictada por el protocolo y va leyendo los bits de la palabra recibida (en MIDI un bit cada 32 μ sg.).
- Por conteo determina cada uno de los bits, conformando la palabra y después lee uno más, comprobando que su valor sea 1 (bit de Stop).
- Tras la recepción de esta palabra, volverá a aguardar la llegada del siguiente bit de Start, poniendo a cero nuevamente su reloj y evitando, así, errores acumulativos del reloj en la lectura de palabras consecutivas. (Las fluctuaciones del reloj en el conteo de los bits de una palabra no son suficientes para causar error en la interpretación... si dicho reloj es mínimamente preciso).



El bit de Stop puede enviarse 2 veces (según protocolos), para mayor seguridad de que el final de una palabra ha llegado, pero ésta es lastrada con un bit más, perdiendo velocidad la transmisión. La ventaja de este sistema es que necesita una cantidad mínima de cables y la desventaja es una pérdida de velocidad, pues lastramos cada palabra enviada con varios bits extra.

CONTROL DE PARIDAD: Método para mejorar la fiabilidad de datos transmitidos. Consiste en determinar si la suma de los valores de los bits transmitidos resulta par o impar. El resultado de la operación se añade en forma de bit o palabra a los datos transmitidos. El receptor vuelve a realizar la cuenta con los datos llegados y comprueba la coincidencia del resultado, certificando que no hay error. Si el número de errores es par, se autocompensan y el sistema resulta engañado, sin detectar error.

Este sistema es similar al empleado en los números de cuenta de los bancos, donde hay algún dígito que es comprobador de los demás. Ocurre lo mismo con la letra del carnet de identidad, que se calcula automáticamente partiendo de los números del carnet.

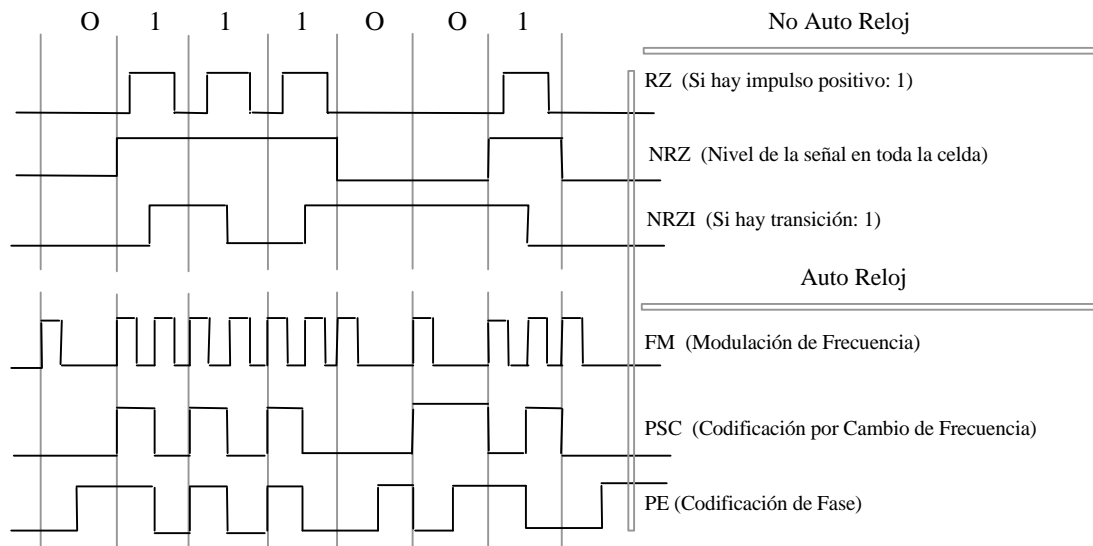
Hay dos formas de calcular la paridad, para adaptarse mejor al tipo de información a transmitir:

- Paridad Transversal: El valor del bit de paridad resulta de la cuenta con los bits de cada palabra y dicho bit se añade al final de la misma.
- Paridad Longitudinal: Se parte de un bloque de palabras a transmitir y se hallan tantos bits de paridad como bits tenga cada palabra. El resultado es una palabra de paridad, donde su primer bit será el control de paridad de todos los primeros bits del conjunto de palabras, y así sucesivamente. La palabra de paridad se envía a continuación del bloque de palabras de datos.

Se establecen, también, dos tipos de paridad, según la cuenta que se haga con los bits:

- Paridad Par: Resultado de efectuar la función lógica denominada O-exclusive.
- Paridad Impar: Utiliza el resultado complementario (inverso) de la operación anterior.

SELF-CLOCK (AUTO RELOJ): Llamado así porque integra la información de reloj en la propia transmisión. Cuando la información se lee posteriormente recupera el reloj de los propios datos. Si es necesario grabar una información serie en soporte magnético para su posterior lectura, desde luego nos quedamos sin el reloj del transmisor y sin la posibilidad de recuperarlo de la grabación (pues la velocidad de lectura puede tener oscilaciones con respecto a la de grabación), haciéndose necesario utilizar un método de auto reloj. A la tarea de descifrar el contenido de un bit, se añade la de extraer una señal de reloj que permita discernir un bit de otro.



Los métodos de comunicación serie Asíncrono y Síncrono, que no integran el reloj en sus datos, son denominados como No Auto Reloj. Colocan los bits en su hueco correspondiente según el método RZ, NRZ, NRZI, etc.

El método Auto Reloj, para transmitir la misma información necesita el doble de frecuencia (aumentando su ancho de banda), pues debe indicar además donde comienza el espacio que ocupará cada bit (determina los pulsos del reloj).