



UNIVERSIDAD NACIONAL AUTONOMA
DE MEXICO



FACULTAD DE INGENIERIA

2
2 es.

"DISEÑO E IMPLANTACION DE UNA RED DE
CONTROL BASADA EN MICROPROCESADORES
DE 8 BITS"

T E S I S

QUE PARA OBTENER EL TITULO DE
INGENIERO MECANICO ELECTRICISTA
(AREA ELECTRICA-ELECTRONICA)

P R E S E N T A :
ANTONIO AKE FARFAN

DIRECTOR DE TESIS: ING. MOISES RUEDA GUTIERREZ,

MEXICO, D. F.

1998



Universidad Nacional
Autónoma de México

Dirección General de Bibliotecas de la UNAM

Biblioteca Central



UNAM – Dirección General de Bibliotecas
Tesis Digitales
Restricciones de uso

DERECHOS RESERVADOS ©
PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL

Todo el material contenido en esta tesis esta protegido por la Ley Federal del Derecho de Autor (LFDA) de los Estados Unidos Mexicanos (México).

El uso de imágenes, fragmentos de videos, y demás material que sea objeto de protección de los derechos de autor, será exclusivamente para fines educativos e informativos y deberá citar la fuente donde la obtuvo mencionando el autor o autores. Cualquier uso distinto como el lucro, reproducción, edición o modificación, será perseguido y sancionado por el respectivo titular de los Derechos de Autor.

Agradecimientos

A mi madre por el apoyo y cariño en todos estos años.
A mi madre por darme la mentalidad de trabajo que poseo.

A mi padre por preocuparse tanto por mí, a pesar de mostrarme grosero muchas veces.
A mi padre por enseñarme el amor al "verde".

A mis hermanas por su cariño, comprensión, regaños y consejos.

A la UNAM y a sus maestros por todas sus enseñanzas y principalmente por :
"Aprender a aprender".

Al Programa de Tecnología en Computación (PTC) por enseñarme tantas cosas y
principalmente por "Aprender a enseñar" y "Aprender enseñando".

A todas las personas que alguna vez he dicho gracias, de nueva cuenta : gracias

A mis amigos de la FI :

A la "chela" :Siempre adorable, tanto buenos momentos, y por supuesto, por siempre amigos

Al chavo Mauricio F. por ser tan buen amigo y muy chistoso.

A Martha : gracias por organizar siempre todo, por tu amistad, y ayudarme en este trabajo

Al CHAVO *ojets*: a pesar de ser tan diferentes entre nosotros, demostramos llevarnos bien.

A Polo : buen compañero, muy buen amigo.

A Beto : Que decir del *buen beto*

A Lucho : por escucharme y darme su punto de vista.

A Julian y Memo : por trabajar tan bien en equipo : un uP, un tren, un cajero... .

A Oscar M. : con el que he compartido muchas cosas el basquet y..... casi 9 años de amistad.

A araujo, rafa, pepoyo, yadira, daniel, claudia, glendy, alfonso, angel, roxana, mariza, erick, santiago, lorenzo, toño, raymundo, morquecho, y a todos los compañeros que en este momento no me acuerdo de su nombre, pero que alguna vez me apoyaron de manera desinteresada.

A mis amigos del PTC :

A todos los chicos, con los cuales alguna vez experimente dando clase.

A los becarios, que a pesar de los pocos recursos, y a pesar de los pesares, son de los mejores estudiantes en la Facultad de Ingeniería

A mena, rodrigo, chesst, leo, rody, carlos, mendieta, knigh, falcon, firebird, david, gork, yoatzia, skid, navajo, , ram, sampras, yumbe, salvador, greedo, sigfried, cesarg, steeler,

phoenix, vanhua, nieto, mune, fernando, renorio, mario, gus, patty, kyo, alma, rex, moifeo, sergio, mercurio y a todos los becarios que en este momento no me acuerdo, pero que constituyen o han constituido al PTC

INDÍCE GENERAL

INTRODUCCIÓN	1
CAPÍTULO 1 Generalidades	5
1.1 Importancia de los dispositivos de 8 bits	5
1.2 Aplicaciones de los microcontroladores de 8 bits	6
1.3 Transmisión de datos	7
1.4 Transmisión serial asíncrona	7
1.5 Transmisión serial síncrona	8
1.6 Redes, Protocolos y Modelo OSI	9
1.7 Componentes de una red de microcomputadoras	10
1.8 Medios de comunicación	11
1.8.1 Cable de Par Trenzado	13
1.8.1.1 Componentes de una red de par trenzado	13
1.8.2 Cables de par trenzado blindado	15
1.8.3 Cable coaxial de Banda Base	15
1.8.3.1 Componentes de una red de cable coaxial de banda base.	16
1.8.4 Cable coaxial de banda ancha	17
1.8.5 Cable de fibra óptica	17
1.8.5.1 Componentes de una red de cable de fibra óptica.	18
1.8.6 Redes Inalámbricas	20
1.8.6.1 Infrarrojos	20
1.8.6.2 Radio	20
1.9 Comparación entre los diferentes medios de transmisión	21
1.10 Topologías	22
1.10.1 Topología en estrella	22
1.10.2 Topología en anillo	23
1.10.3 Topología en bus	25
1.11 Estándares de comunicación	26
1.11.1 Importancia de los estándares en las comunicaciones de datos	27
1.11.2 Estándares y organizaciones de normabzación	27
1.12 Características de las redes de control distribuido	28
1.13 Diferencias entre las redes de control y las redes LAN de computadoras	29
1.14 Criterios de diseño	30
CAPÍTULO 2 Diseño de la red	31
2.1 Diseño de capa física	31
2.1.1 Modems telefónicos	33
2.1.2 EIA-232	35
2.1.3 EIA 422	38
2.1.4 EIA-485	40

ÍNDICE GENERAL

2.2 Diseño de capa enlace de datos	43
2.2.1 Selección de protocolo de acceso al medio	44
2.2.1.1 Polling (Llamada Selectiva)	44
2.2.1.2 Token Bus	45
2.2.1.3 Conteo descendente binario	47
2.2.1.4 CSMA/CD Carrier Sense Media Access / Collision Detect (Protocolo de acceso múltiple con detección de portadora y detección de colisiones)	48
2.2.2 Delimitación de mensajes	50
2.2.3 Detección y Corrección de Errores	52
2.2.3.1 Comprobación de Paridad	53
2.2.3.2 "Checksum" aritmético	54
2.2.3.3 Comprobación de redundancia cíclica (CRC : Cyclic Redundacy Check)	54
2.2.4 Control de errores	57
2.2.5 Control de flujo	60
2.3 Diseño de Capa de aplicación	62
CAPÍTULO 3 Desarrollo de la Red	63
3.1 Selección del microcontrolador	63
3.1.1.1 Generalidades del MC68HC11	63
3.1.2 Unidad de control de procesamiento (CPU) del M68HC11	65
3.1.3 Modelo de programación	67
3.2 Sistema de Desarrollo	67
3.3 Capa física	71
3.3.1 Sistema de Comunicaciones Serial del HC11	71
3.3.2 Interfaz de la PC con la interfaz serial EIA-485	72
3.4 Capa enlace de datos	74
3.4.1 Implementación de control de acceso al medio	74
3.4.1.1 Realización del detector de IDLE	74
3.4.1.2 Realización del detector de Colisiones	77
3.4.2 Detección de errores	81
3.4.2.1 Métodos para calcular CRC	82
3.4.2.1.1 Método por Hardware	82
3.4.2.1.2 Método por Software	83
3.4.2.1.2.1 Algoritmo Bitwise	83
3.4.2.1.2.2 Algoritmo manejado por tabla	85
3.4.3 Estructura de los mensajes	86
3.4.3.1 Mensajes de control	86
3.4.3.1.1 Mensaje Solicitud de Conexión (S.C.)	87
3.4.3.1.2 Mensaje Cerrar Conexión (C.C.)	87
3.4.3.1.3 Mensaje de confirmación positiva (ACK)	88
3.4.3.1.4 Mensaje de confirmación negativa (NAK)	88
3.4.3.1.5 Mensaje Solicitud de estado de un mensaje de datos (REP)	89
3.4.3.2 Mensaje de datos	89
3.4.4 Operación del Protocolo	90
3.4.4.1 Establecimiento de una conexión	90
3.4.4.2 Transferencia de Mensajes de datos	92
3.4.4.3 Desconexión	93
3.4.4.4 Temporizador de respuesta	93
3.4.4.5 Mensajes válidos	94

3.4.4.6 Contador de Time-out's	94
3.4.5 Control de Flujo	94
3.4.6 Interfaz de Usuario	94
3.5 Programación del Protocolo	95
CAPÍTULO 4 Pruebas y Resultados	103
4.1 Pruebas de CSMA/CD	104
4.2 Pruebas de establecimiento de conexión	105
4.3 Pruebas de Transmisión de Mensajes	106
4.4 Pruebas de Liberación de Conexión	107
4.5 Cálculo del temporizador de respuesta	107
4.6 Cantidad de memoria utilizada	108
4.7 Factores que limitan la velocidad máxima de transmisión	110
4.8 Elementos de programación	112
CONCLUSIONES	115
APÉNDICES	119
Apéndice A EIA-485	119
Medio de interconexión	119
Opciones de terminación	119
Parámetros diferenciales de 485	119
Protección en contienda	122
Apéndice B Circuito de cada estación HC11	125
Apéndice C Descripción de los comandos del Protocolo	126
Descriptores	126
Variable STN_CONNECTED	127
Comandos	127
STARTCON	127
TRANSMITE_DATOS	128
STATUSMSG_DATOS	130
SENDAGAIN_MSG	131
CLOSECON	131
RESET_PROTOCOL	132
STATUSMSGRCV	132
FREEBUFFRCV	133
INICIAVARS	134
Manejo de buffers de recepción	134
Manejo de buffers de transmisión	136

ÍNDICE GENERAL

Apéndice D Listado parcial de programas	139
Apéndice E Fotografía del sistema utilizado	145
Apéndice F Hojas de especificación	146
BIBLIOGRAFÍA	163

INTRODUCCIÓN

En la década de los '80s la mayoría de los sistemas de control fueron implementados por medio de un microprocesador. A medida que los requerimientos de las aplicaciones fueron aumentando, los sistemas se volvieron cada vez más complejos, y resolver el problema con un solo microprocesador se volvía complicado. El problema de controlar el acceso a una oficina se puede realizar por medio de un control centralizado pero el de todo el edificio requiere pensar en otra solución. Una de ellas es la creación de las redes de control distribuido, dividiendo de esta manera un sistema complejo en pequeños sistemas, que utilizan un microprocesador por nodo, que se comunican entre sí.

Las redes de control distribuido presentan muchas ventajas sobre un diseño estándar utilizando un microprocesador central para llevar a cabo el control :

- Sistema flexible. Las unidades de control pueden ser fácilmente añadidas o removidas.
- Tolerancia a fallas. Si una unidad falla, no necesariamente fallará todo el sistema.
- Facilidad de desarrollo. Cada unidad de control puede ser desarrollada y probada individualmente
- Interconexión y cableado simplificado. El proceso de control ocurre muy cerca de la acción. Un controlador puede ser colocado en el lugar donde se encuentra el sensor o el actuador o bien puede ser integrado como controlador del sistema sensor/actuador

El elemento principal de un sistema distribuido es la red de comunicaciones, la cual tiene como objetivo transferir información, de manera confiable y eficiente, a través de los nodos de red.

El uso de las redes de control en sistemas comerciales e industriales se ha incrementado de manera considerable en los últimos años, y algunos de los campos en los que se han encontrado mayor aplicación son .

- Monitoreo médico

INTRODUCCIÓN

- Sistemas de seguridad.
- Sistemas de detección de fuego.
- Control de Procesos.
- Control HVAC (Heating Ventilation and Air Conditioning).
- Transacción en puntos de venta (POS) (Point-of-Sales) .
- Casas y edificios inteligentes.
- Automatización en fábricas.
- Sistemas de control en automóviles.

Actualmente existen tres vertientes en lo que se refiere a las redes de control distribuido :

1. Protocolos específicos a cada campo de aplicación. Por ejemplo : CAN (Controller Area Network, utilizado en la industria automotriz), BACNET (creado por la ASHRAE : American Society of Heating, Refrigeration and Air-Conditioning Engineers, para la automatización de edificios).
2. Protocolos generales. Por ejemplo : LONWorks (utilizado en varias aplicaciones).
3. Protocolo específico a la aplicación.

En el primer y segundo caso, varios fabricantes pueden desarrollar productos para el protocolo, lo que genera una competencia en el mercado, provocando una reducción en los precios. La más importante de las ventajas, en el segundo caso, es que representa una solución totalmente definida, que puede ser implantada en cualquier aplicación. Pero es únicamente la tercera la que brinda la posibilidad de crear una solución de acuerdo a los requerimientos de la aplicación. Además, los protocolos específicos a cada campo de aplicación presentan la desventaja de encontrarse en su etapa de desarrollo, el cual ha sido lento, esto debido a la necesidad de hacer un análisis de todos los dispositivos requeridos por esa área y agruparlos en unos cuantos solamente. Los protocolos generales pueden resultar caros o lentos para ciertas aplicaciones, esto es porque su carácter general lo vuelve difícil de implantar en algunos casos. La desventaja de un protocolo específico a la aplicación es su tiempo de desarrollo, debido a que el diseñador requiere implantar todas las funciones de comunicación.

Algunas de las decisiones que se deben de tomar al momento de desarrollar la red de un sistema son :

- ¿Qué tipo de transreceptores se deben de utilizar ?
- ¿Cómo compartir el canal de comunicaciones ?
- ¿Cómo detectar errores ?
- ¿Qué se debe hacer cuando se detecta un error ?
- ¿Cómo se debe organizar el software ?

El crear estas redes de control puede consumir mucho tiempo y resultar muy complejo, por lo que el presente trabajo trata de dar a conocer una metodología de diseño aplicable a cualquier sistema digital que opera con microprocesadores y una red de control, simplificando así el diseño y desarrollo de las aplicaciones que utilizan redes de control. Para ello se ha dividido en 4 capítulos. En el primer capítulo se exponen las generalidades de las redes de comunicación, haciendo énfasis en los medios de comunicación, las topologías de red, las similitudes y diferencias entre las redes de computadoras y las de redes control, además, se mencionan los criterios de diseño del protocolo de comunicaciones. En el segundo capítulo se presenta el diseño del protocolo, analizando cada una de las características de las capas física y enlace de datos del modelo de referencia OSI (Open Systems Interconnection). En el tercer capítulo se describe, en detalle, la implantación de cada una de las características del protocolo de comunicación en un microcontrolador en específico. En el capítulo cuarto, se describen las pruebas realizadas para comprobar el funcionamiento del protocolo y finalmente se presentan las conclusiones, bibliografía y apéndices.

CAPÍTULO 1

1. Generalidades

En el presente capítulo se mencionarán algunos conceptos básicos de transmisión de datos y redes. Además se explicará la importancia de los dispositivos de 8 bits en la industria de los microprocesadores.

1.1 Importancia de los dispositivos de 8 bits

Aunque hay una fuerte tendencia a alejarse de los microprocesadores de propósito general de 8 bits, para pasar a los dispositivos de 16 y 32 bits, los microcontroladores de 8 bits de un sólo circuito integrado, generalmente inspirados en la arquitectura de sus predecesores de propósito general continúan usándose en aplicaciones sencillas y de bajo costo: equipo doméstico, controladores sencillos, procesadores de periféricos, etc..

La gama actual de aplicaciones de los microprocesadores es muy grande. En uno de los extremos, los microprocesadores que cuestan más de 200 dólares se usan simplemente como computadoras de propósito general. Sin embargo, la mayor parte de la producción de microprocesadores no se usa en computadoras de propósito general, sino en aplicaciones dedicadas, en las que el microprocesador actúa como un microcontrolador con un programa fijo en memoria. Los avances en los niveles de integración han logrado incluir no sólo la UCP (Unidad de Control de Procesamiento), sino también los otros componentes de una computadora, como la memoria, en un solo circuito integrado con lo cual el "circuito computadora" puede ser de tan sólo dos dólares en el caso de aplicaciones dedicadas con mercados suficientemente grandes (100 000 dispositivos). En la Tabla 1.1 y Tabla 1.2 se muestra la magnitud del mercado de los microprocesadores de 4, 8 y 16 bits.

	'90	'91	'92	'93	'94	'95	'96	'97	'98	'99	'2000
4 bits	1,393	1,597	1,596	1,698	1,761	1,826	1,849	1,881	1,856	1,816	1,757
8 bits	2,077	2,615	2,862	3,703	4,689	5,634	6,553	7,529	8,423	9,219	9,715
16 bits	192	303	340	484	810	1,170	1,628	2,191	2,969	3,678	4,405

Tabla 1.1 Venta en millones de dólares de microcontroladores alrededor del mundo.

Autor

Robin Getz de National Semiconductor

	'90	'91	'92	'93	'94	'95	'96	'97	'98	'99	'2000
4 bits	778	906	979	1,036	1,063	1,110	1,100	1,096	1,064	1,025	970
8 bits	588	753	843	1,073	1,449	1,803	2,123	2,374	2,556	2,681	2,700
16 bits	22	38	45	59	106	157	227	313	419	501	585

Tabla 1.2 Ventas de microcontroladores alrededor del mundo (en millones de piezas).

Autor

Robin Getz de National Semiconductor

Se puede observar que los dispositivos de 8 bits cuentan con más de la mitad del mercado, y eventualmente seguirá aumentando. Esta es la razón por la cual los fabricantes de microprocesadores de microprocesadores están continuamente mejorando y sacando nuevos modelos microcontroladores de 8 bits.

1.2 Aplicaciones de los microcontroladores de 8 bits

Como se mencionó anteriormente, el campo de aplicación de los microcontroladores de 8 bits es muy amplio, citar todas sus aplicaciones sería imposible por lo cual sólo se mencionarán algunas de las aplicaciones típicas de los dispositivos de 8 bits :

- Sistemas de seguridad
- Terminales en Puntos-de-venta
- Sistemas médicos,
- Instrumentos inteligentes
- Sistemas de reconocimiento de patrones
- Control industrial
- Robótica

- Una aplicación especial en la cual los microcontroladores son muy apropiados, es el monitoreo de información. Un microcontrolador puede monitorear parámetros ambientales tales como : temperatura, humedad, lluvia, etc.. Su pequeño tamaño, bajo consumo de energía y flexibilidad hace de éstos dispositivos ideales para registrar y monitorear un sistema "sin atención".

1.3 Transmisión de datos

El término "transmisión de datos" se refiere al proceso de transporte de datos de un punto a otro. El término dato se refiere a caracteres alfabéticos, numéricos o de propósito especial, los cuales cuando son agrupados, constituyen alguna palabra ó mensaje. La transmisión de los datos se puede efectuar de muchas formas, pero en este trabajo sólo estamos interesados en las que se utilizan en las redes de comunicaciones.

1.4 Transmisión serial asíncrona

El método de transmisión asíncrona envía los datos transmitiéndolos caracter por caracter (donde cada caracter tiene una longitud de 5 o 8 bits). Cuando no se transmite ningún caracter, la línea entre el emisor y el receptor estará en estado de "reposo" (IDLE). El principio de cada caracter se indica mediante un bit de comienzo. A continuación se transmite el caracter comenzado por el bit menos significativo. El último elemento es el bit de parada. Se debe especificar la longitud mínima de este último elemento y normalmente corresponde a 1, 1.5 ó 2 veces la duración de un bit convencional.

La sincronización se realiza disponiendo en el emisor y receptor de relojes, normalmente ajustados a la misma frecuencia, la resincronización del receptor se realiza a la recepción del bit de inicio.

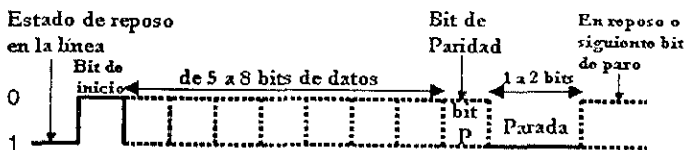


Figura 1-1 Formato de un carácter asíncrono

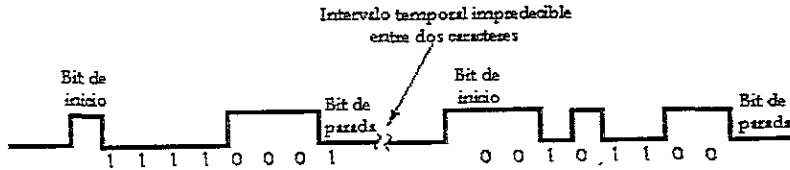


Figura 1-2 Cadena de caracteres de 8 bits asíncronos

En la transmisión asíncrona el mensaje debe ser lo suficientemente corto para evitar que los efectos de pérdida de sincronía debido a cualquier variación de las condiciones de transmisión o a la misma variación de tolerancia de los componentes electrónicos, afecte a dicho mensaje.

1.5 Transmisión serial síncrona

En la transmisión síncrona, se transmite un bloque de bits como una cadena estacionaria sin utilizar códigos de comienzo o parada. El bloque puede tener una longitud de muchos bits. Para prevenir la desincronización entre el emisor y el receptor, se deben sincronizar sus relojes de alguna manera. Una posibilidad puede ser proporcionar la señal de reloj a través de una línea independiente; uno de los extremos utilizará esta señal a modo de reloj. Esta técnica funciona bien a distancias cortas, sin embargo a distancias superiores, los pulsos de reloj son susceptibles de las mismas dificultades y defectos que las propias señales de datos, por lo que pueden aparecer errores de sincronización en la propia señal de datos. Para la señalización digital esto se puede llevar a cabo mediante la codificación Manchester o Manchester Diferencial. Para señales analógicas se han desarrollado, a su vez, diversas técnicas; por ejemplo, la frecuencia de la portadora por sí misma se puede utilizar para sincronizar el receptor usando la fase.

En la transmisión síncrona se requiere además un nivel de sincronización adicional para que el receptor pueda determinar donde está el comienzo y el final de cada bloque de datos. Para llevar esto a cabo, cada bloque comienza con un patrón de bits de preámbulo y generalmente termina con un patrón de bits al final.

Delimitador 8-bit	Campos de control	Campos de datos	Campos de control	Delimitador 8-bit
----------------------	----------------------	--------------------	----------------------	----------------------

Figura 1-3 Formato de una trama síncrona

En la Figura 1-3 se muestra, en términos generales, un formato típico para un trama de una transmisión síncrona. Normalmente, la trama comienza con un preámbulo de 8 bits denominado indicador. Este mismo se utiliza al final del mensaje. El receptor buscará la aparición del indicador que informará sobre el comienzo de la trama. El indicador estará seguido por algunos campos de control, el campo de datos (de longitud variable para la mayoría de los protocolos), más campos de control y, al final, se repite el indicador.

1.6 Redes, Protocolos y Modelo OSI

Una red de computadoras es un sistema de comunicaciones que permite a un número de dispositivos independientes comunicarse entre sí. Para lograr esta comunicación es necesaria la definición y utilización de un protocolo. El cual se define como el conjunto de reglas que gobiernan el intercambio de datos entre dos dispositivos. Los puntos que se definen en un protocolo son :

- La sintaxis : incluye aspectos tales como el formato de los datos y los niveles de señal.
- La semántica : incluye información de control para la coordinación y el manejo de errores.

Una arquitectura de protocolos divide las tareas involucradas en la comunicación en un conjunto de módulos que realizan todas las funciones. El modelo de referencia OSI (The reference Model of Open Systems Interconnection) divide las funciones y los servicios de red en 7 capas. Muchas de las redes comerciales no implementan todas las capas del modelo OSI, pero es más sencillo entender cualquier estructura de red comparándola con el modelo OSI.

En la Figura 1-4 se muestra el modelo OSI y se definen brevemente las funciones que realiza cada capa.

Aplicación
Proporciona el acceso al entorno OSI para los usuarios y también proporciona servicios de información distribuida
Presentación
Proporciona a los procesos de aplicación independencia respecto a las diferencias en la representación de los datos (sintaxis)
Sesión
Proporciona el control de la comunicación entre las aplicaciones ; establece, gestiona y cierra las conexiones (sesiones) entre las aplicaciones
Transporte
Proporciona seguridad, transferencia transparente de datos entre los puntos finales ; proporciona además procedimientos de recuperación de errores y control de flujo origen-destino
Red
Proporciona independencia a los niveles superiores respecto a las técnicas de conmutación y de transmisión utilizadas para conectar los sistemas ; es responsable del establecimiento, mantenimiento y cierre de las conexiones.
Enlace de datos
Proporciona un servicio de transferencia de datos seguro a través del enlace físico ; envía bloques de datos (frames) llevando a cabo la sincronización, el control de errores y de flujo necesarios
Física
Se encarga de la transmisión de cadenas de bit no estructuradas sobre el medio físico ; está relacionada con las características mecánicas, eléctricas, funcionales y de procedimiento para acceder al medio físico.

Figura 1-4 Modelo OSI

El modelo OSI especifica las funciones de cada capa en términos abstractos, dejando los detalles de la implementación a los usuarios.

1.7 Componentes de una red de microcomputadoras

Una red de microcomputadoras consta de los siguientes componentes (Figura 1-5) :

- Microcomputadoras (Estaciones) : son las que proporcionan a la red toda la potencia de proceso y las que crean los mensajes.
- Interfaces : conectan los dispositivos a la red y hacen posible la comunicación con otros dispositivos. Una interfaz puede ser una tarjeta de red, un modem, o un puerto de comunicaciones de una microcomputadora.

- Topología : forma física de interconexión entre los dispositivos de la red. Es la forma de poner orden a la conexión indiscriminada de dispositivos.
- Medio de transmisión : proporciona el enlace físico que lleva la información de un punto a otro de la red. A este enlace se le denomina también “canal”, “línea”, o “circuito”.
- Protocolo : son las reglas y convenciones que controlan el intercambio de información.

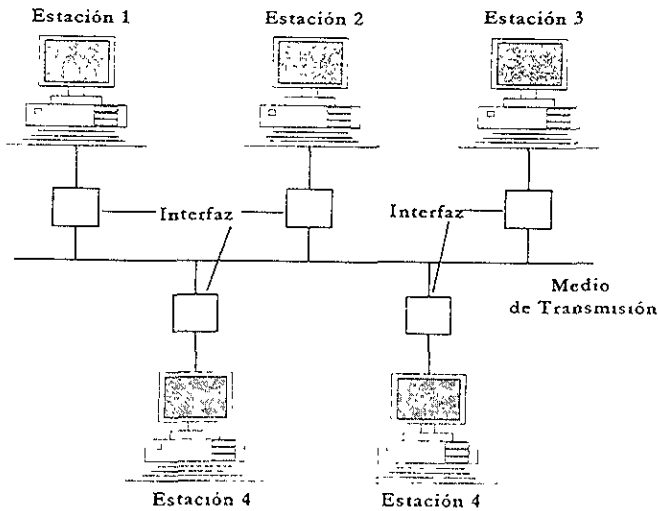


Figura 1-5 Red de microcomputadoras

1.8 Medios de comunicación

Cualquier medio, físico o no (comunicaciones inalámbricas), que pueda transportar información en forma de señales electromagnéticas, se puede utilizar en redes locales como medio de transmisión.

Las líneas de transmisión son la espina dorsal de la red, por ellas se transmite la información entre los distintos nodos. Para efectuar la transmisión de la información se utilizan varias técnicas, pero las más comunes son : banda base y banda ancha (*baseband* y *broadband*)

a) Banda Base

La transmisión en banda base consiste en el envío de una señal en sus frecuencias originales, es decir, desmodulada. Los diseñadores de redes locales han adoptado ampliamente técnicas de transmisión de banda base porque no es necesario el uso de modems y porque la señal se puede transmitir a alta velocidad. Banda base no es muy adecuada para transmisiones a largas distancias ni para instalaciones sometidas a un alto nivel de ruido e interferencia.

El empleo de esta técnica permite utilizar dispositivos de interfaz y repetidores que resultan muy económicos.

La técnica de banda base es especialmente adecuada en la transmisión a corta distancia.

Un canal que trabaje en modo de banda base utiliza todo el ancho de banda, por lo que, en un determinado momento, sólo puede transmitir una señal.

Los tipos de cable de banda base son : el cable de par trenzado y el cable coaxial de banda base.

b) Banda Ancha

Esta técnica consiste básicamente en modular la información sobre ondas portadoras analógicas. Varias portadoras pueden compartir la capacidad del medio de transmisión, mediante técnicas de multiplexaje por división de frecuencia. Los fabricantes de dispositivos de televisión por cable (CATV) han utilizado esta técnica desde hace mucho tiempo.

Cuando se utiliza el sistema de banda ancha para transmitir, se necesita utilizar modems para modular la información. Los modems utilizados en las redes de banda ancha son

dispositivos muy complejos, pues realizan funciones de modulación / demodulación y de transmisión/recepción. Los medios de transmisión de banda ancha son : el cable coaxial de banda ancha y el cable de fibra óptica.

1.8.1 Cable de Par Trenzado

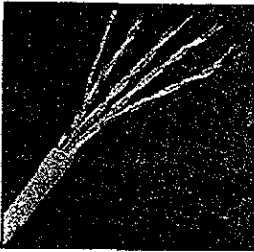


Figura 1-6 Cable coaxial sin blindaje

El cable de par trenzado es el cable que se utiliza normalmente en las instalaciones telefónicas, sea para conectar teléfonos, telex u otros dispositivos.

Como su nombre implica, este cable está compuesto por un par de hilos trenzados entre sí. El grosor de los hilos varía, al igual que el número de vueltas (o trenzados) por pulgada. El trenzado mantiene estables las propiedades eléctricas a lo largo de toda la longitud del cable y reduce las interferencias

creadas por los hilos adyacentes en los cables compuestos por varios pares. Este tipo de cable suele estar compuesto por hilos de cobre. Por otro lado, es importante conocer la impedancia característica de una línea con cierta precisión para diseñar una terminación adecuada para ésta, de modo que se evite la reflexión de las señales transmitidas, lo que puede dar lugar a errores en la transmisión.



Figura 1-7 Par trenzado

1.8.1.1 Componentes de una red de par trenzado

Una red de par trenzado consta del cable principal y de los siguiente dispositivos (Figura 1-7):

- Transreceptores: Unidades interfaz de red que proporcionan la inteligencia necesaria para leer direcciones de los mensajes y otras funciones de comunicación orientadas a la red

- Derivadores de cable: Conectan el transreceptor al cable principal
- Repetidores : Amplían la potencia de la señal a medida que los mensajes pasan de una sección del cable a otra.

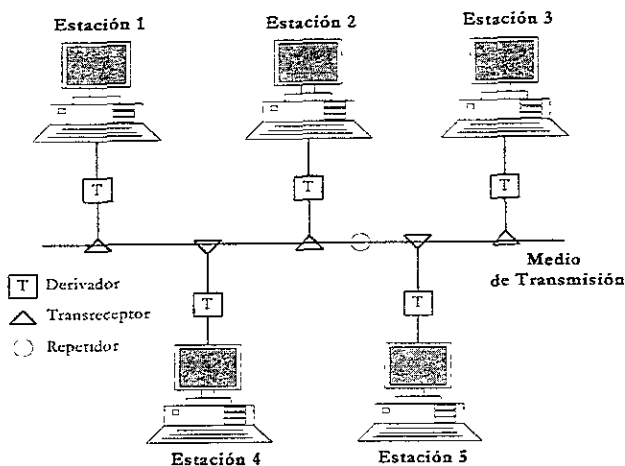


Figura 1-7 Componentes de una red de cable de par trenzado.

Ventajas del cable de par trenzado

- El cable y la instalación son baratos.
- Fácil de instalar
- Permite ser configurado en diferentes topologías bus, estrella y en anillo

Desventajas

- El bajo nivel de protección lo hace vulnerable a interferencias eléctricas, lo que puede producir errores en la recepción. Para obtener un mayor protección se utiliza el cable de par trenzado blindado.
- La mayoría de las implementaciones limitan el número de estaciones conectadas a la línea y limitan las distancias a un sólo edificio.

1.8.2 Cables de par trenzado blindado

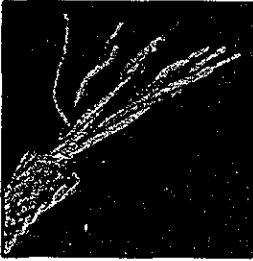


Figura 1-8 Cable par trenzado blindado

El cable de par trenzado blindado es muy similar al anterior, la diferencia está en que el grosor de los hilos es mayor y que están forrados de una hoja metálica muy fina que actúa como blindaje. Este blindaje reduce el efecto de las interferencias causados por fuentes externas como cables de alta tensión o tubos fluorescentes.

Las ventajas de cable de par trenzado blindado son :

- Menor sensibilidad al ruido que los cables de par trenzado sin blindaje
- Soporta velocidades de transmisión más elevadas
- Es relativamente más fácil de trabajar que el cable coaxial.

Sus desventajas incluyen :

- Cable cuya instalación es difícil y de alto costo

1.8.3 Cable coaxial de Banda Base

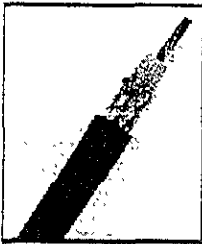


Figura 1-9 Cable coaxial

En el cable coaxial de banda base, el hilo conductor central está rodeado de una malla muy fina de hilos de cobre. El espacio que queda entre el hilo y la malla está aislado para soportar los 2 conductores y para mantener las propiedades eléctricas. Todo el cable está cubierto por un aislamiento de protección para reducir emisiones eléctricas.

El cable coaxial de banda base tiene un canal que transporta una sola señal digital a una velocidad de transmisión muy alta. Los bits se ponen directamente en el cable sin modulación alguna.

Aunque la distancia máxima recomendada de una red local de banda base es de aproximadamente 3 km, no es conveniente realizar instalaciones de más de 500 m

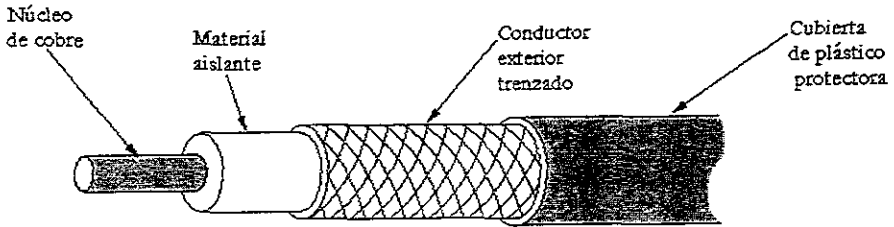


Figura 1-10 Componentes del cable coaxial

1.8.3.1 Componentes de una red de cable coaxial de banda base.

En muchos aspectos el cable coaxial de banda base es similar al par trenzado. Para convertir un cable en una red (Figura 1-11) son necesarios los siguientes dispositivos :

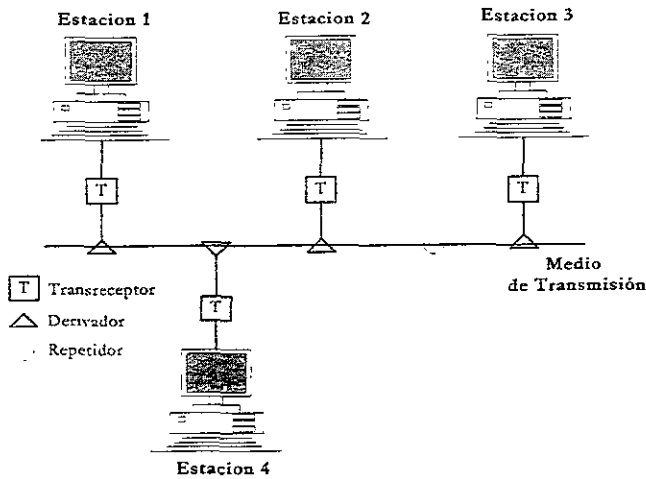


Figura 1-11 Componentes de una red de cable coaxial

- **Transreceptores :** Unidades interfaz de red que proporcionan la inteligencia necesaria para leer direcciones de los mensajes y otras funciones de comunicación orientadas a la red.

- **Derivadores de cable :** Conectan el transreceptor al cable principal
- **Repetidores :** Amplifican la señal a medida que los mensajes pasan de una sección de cable a otra.

La mayoría de las redes de cable coaxial de banda base limitan la distancia entre estaciones y el número de nodos.

El cable coaxial de banda base ofrece mayor resistencia a las interferencias y un mejor rendimiento que el cable de pares trenzados, a un costo mayor.

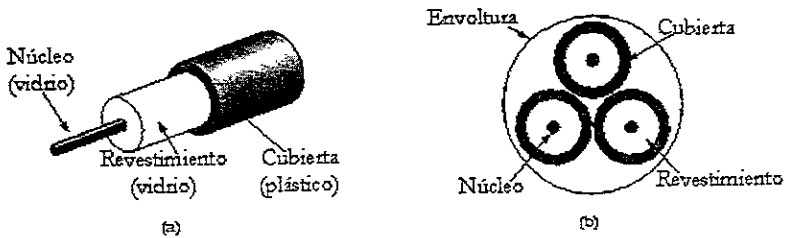
1.8.4 Cable coaxial de banda ancha

El cable coaxial de banda ancha es adecuado para aplicaciones de alta frecuencia, de banda muy ancha y alta velocidad. Actualmente es la opción más práctica para redes que cubren zonas no muy extensas y que han de ser capaces de transmitir señales digitales, voz y video.

El costo del sistema del cable coaxial de banda ancha provoca que sea poco práctico para redes pequeñas.

1.8.5 Cable de fibra óptica

El cable de fibra óptica es un medio de transmisión que cada vez se está utilizando más en redes que exigen velocidades de transmisión muy altas y fiables. Las señales luminosas se transmiten a través de un cable (guía de ondas) compuesto por un grupo de cristales o de fibras plásticas. Cada filamento tiene un núcleo central de plástico o cristal con un alto índice de refracción, rodeado de una capa de un material similar con un índice de refracción ligeramente menor. El revestimiento aísla las fibras y evita que se produzcan interferencias entre filamentos adyacentes, y al mismo tiempo proporcionan protección al núcleo. Todo el conjunto está normalmente protegido por otras capas que no tienen ninguna otra función que la de proteger los filamentos. (Figura 1.12)



(a) Vista de lado de una fibra individual
 (b) Vista de extremo de una envoltura con 3 fibras

Figura 1-12 Fibra óptica

Un pulso de luz puede usarse para indicar un bit 1, la ausencia de un pulso indica una señal 0.

1.8.5.1 Componentes de una red de cable de fibra óptica.

Además del cable, una red de cable de fibra óptica (Figura 1-13) necesita los siguientes elementos :

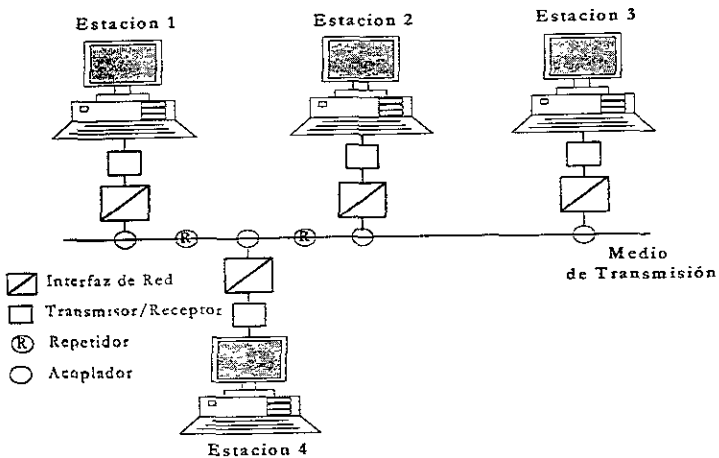


Figura 1-13 Componentes de una red de fibra óptica

- **Transreceptores** : Realizan funciones de transmisión y recepción y constan de :
 - 1) Un transmisor. El cual está formado por una fuente de alimentación y un foco de luz. La luz se emite por medio de un láser o a través de un Diodo emisor de luz LED *Light Emiting Diode*.
 - 2) Un receptor. Llamado también detector, detecta las señales de luz y las convierte en señales eléctricas.
- **Repetidores** : Estos son necesarios cuando se utiliza el cable para redes muy grandes. Los sistemas de fibra óptica usan una menor cantidad de repetidores que ningún otro medio de transmisión.
- **Acopladores y conectores** : Se usan para unir los extremos de dos cables.

En una red que disponga de este medio de transmisión, una estación envía datos al transmisor en forma de señales eléctricas donde se convierten en señales luminosas. Estas señales son recogidas por la guía de onda y enviadas al receptor. En el receptor, las señales luminosas se vuelven a convertir en señales eléctricas y se envían a la estación destino.

Los sistemas de fibra óptica, al contrario que los de cable coaxial, generalmente no necesitan repetidores. Un sistema de fibra óptica puede tener hasta cuatro mil metros de cable, no siendo necesario utilizar repetidores cuando la longitud de la red supera los 1,500 metros.

Las señales que transmite la fibra óptica pueden ser digitales o analógicas. Las señales digitales están formadas por pulsos "on/off"; mientras que las señales analógicas utilizan intensidades luminosas variables. Un cable de fibra óptica sólo pueden transmitir en una dirección, para que la comunicación se efectúe en ambos sentidos se necesitan dos cables.

La fibra óptica sirve perfectamente en sistemas que necesitan efectuar transmisiones de datos y video a una velocidad alta, a distancias mucho mayores que lo que permiten

otros tipos de medios de transmisión. Es particularmente conveniente en instalaciones en las que se espera aumente rápidamente la demanda de comunicaciones, y donde el espacio y las interferencias causan problemas.

La fibra óptica no resulta interesante en instalaciones pequeñas donde el costo es un factor importante. En la actualidad las redes de fibra óptica son muy caras, al igual que la instalación y el equipo, además resulta muy compleja, debido a las conexiones extremadamente precisas.

1.8.6 Redes Inalámbricas

En algunos entornos el tendido de cables puede resultar muy difícil o los equipos pueden estar cambiando continuamente de lugar de trabajo. Una solución a éstos y otros problemas son las redes inalámbricas. Las estaciones de una red inalámbrica disponen de tarjetas especiales para transmisión y recepción.

Se pueden distinguir 2 tipos de tecnologías de comunicación inalámbricas de redes :

1.8.6.1 Infrarrojos

Las redes de infrarrojos utilizan fuentes de luz infrarrojas para comunicar dos puntos de la red pero no pueden atravesar estructuras opacas, lo que implica la necesidad de tener una visión libre y sin obstáculos para poder establecer una comunicación. Siendo esto uno de los inconvenientes más importantes de este tipo de redes.

Las principales características de este medio de comunicación son :

- Propagación en línea recta.
- No se necesita ningún tipo de permisos ni licencias para su instalación.
- Permite anchos de banda elevados de hasta 20 Mbps.
- Inmune a las interferencias de frecuencias de radio.

1.8.6.2 Radio

Las redes por medio de radio utilizan frecuencias de radio de banda muy estrecha como portadora para comunicar puntos de la red. La mayoría de los productos de redes

inalámbricas por radio utilizan la banda de 902 a 928 Mhz, una banda que también utilizan los teléfonos móviles, teléfonos inalámbricos y muchos otros dispositivos utilizados en el entorno domestico.

Aunque las ondas de radio pueden penetrar estructuras y pueden funcionar en entornos muy cerrados, por desgracia las redes por radio están expuestas a todo tipo de interferencias, provenientes de motores eléctricos, microondas, etc.

En la actualidad los productos sólo soportan redes Ethernet y Token Ring y requieren tener asignada una determinada frecuencia. Uno de los inconvenientes de este tipo de red son los problemas de saturación de frecuencias y seguridad de la información, punto que algunos fabricantes han resuelto utilizando dispositivos para cambiar continuamente la frecuencia de transmisión y evitar que la información pueda ser recibida por elementos no autorizados.

1.9 Comparación entre los diferentes medios de transmisión

En la Tabla 13 se muestra un cuadro comparativo de los diferentes medios de transmisión. Las características que se comparan son; longitud, velocidad típica de transmisión, costos relativos por nodo y por último los costos relacionados con la instalación.

Características	Par Trenzado	Radio	Coaxial	Infrarrojo	Fibra óptica
Longitud (metros)	1-1000	50-10,000	10-10,000	0.5-30	10-10,000
Velocidad de transmisión (kbps)	0.3-2000	1.2-9.6	300-10,000	0.05-20	1-100,000
Costo por nodo(dólares)	\$10-\$30	\$50-\$100	\$30-\$50	\$20-\$75	\$75-\$200
Costo de instalación	Bajo	---	Medio	-	Medio alto

Tabla 13 Cuadro comparativo de los medios de transmisión. Autor: Reza S. Raji de Echelon Corp.

1.10 Topologías

Se denomina topología a la forma geométrica de colocar las estaciones y los cables que las conectan.

Las topologías de red más comunes son

- Topologías en estrella
- Topologías en anillo
- Topologías enbus

1.10.1 Topología en estrella

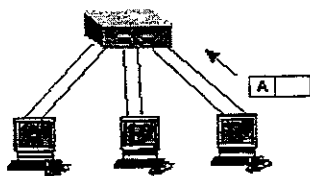


Figura 1-14 Topología en estrella

En una configuración de red en estrella cada estación de trabajo está conectada a un nodo central por medio de un canal punto a punto dedicado. Todos los mensajes son enviados al centro de conmutación, el controlador, para su reenvío a otros nodos. El uso de este controlador central para llevar a cabo todas las transferencias de información simplifica la estructura

de los nodos, pero a expensas de crear una estación de transferencia más compleja.

El tamaño y capacidad de la red está directamente relacionado con la potencia de la estación central.

Ventajas de la topología en estrella

- Es ideal en configuraciones en las que hay que conectar muchas estaciones a una misma estación.
- Las estaciones pueden tener velocidades de transmisión diferentes.
- Las estaciones pueden utilizar distintos medios de transmisión.
- Es fácil detectar y localizar averías.

Inconvenientes

- El fallo de un nodo de la red no repercute en el comportamiento global de la red y solo afectará el tráfico relacionado con ese nodo. Cuando afecta al *hub* central, toda la red se viene abajo.
- Elevado costo debido a la complejidad de la tecnología que se necesita en el nodo central.
- La instalación de los cables resultan bastante cara.
- La actividad que debe soportar el nodo central hace que normalmente las velocidades de transmisión sean inferiores a las que se consiguen en las topologías de bus y anillo.

1.10.2 Topología en anillo

La red en anillo forma un círculo de conexiones punto a punto con estaciones contiguas. Los mensajes van de una estación a otra hasta llegar a la estación adecuada.

Para poder recibir mensajes, cada estación ha de ser capaz de reconocer su propia dirección; sin embargo, no es necesario desviar los mensajes puesto que éstos van automáticamente a la siguiente estación de la red. Los mensajes viajan por el anillo de nodo en nodo, en una única dirección, de un nodo fuente a un nodo destino. No obstante, toda la información pasa por todos los módulos de comunicación, de las estaciones. Las topologías en anillo proporcionan un entorno en el cual varios mensajes pueden circular simultáneamente dentro de la red.

Para enviar un mensaje de un nodo a otro en el anillo, el nodo que envía, pone el mensaje en el medio de comunicación. El mensaje viaja alrededor del anillo hasta que alcanza el nodo destino o es devuelto al remitente que lo envió.

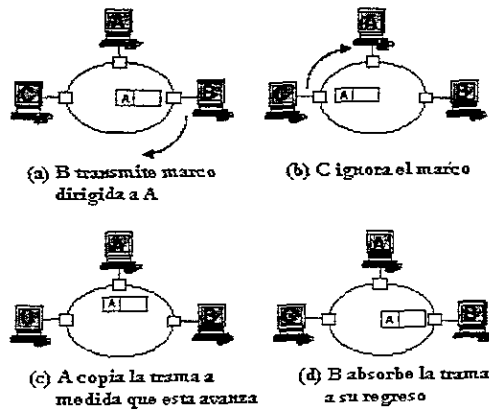


Figura 1-15 Topología en anillo

Ventajas

- La red no depende de un nodo central
- Es fácil localizar los nodos y enlaces que originan errores.
- Permite incrementar o disminuir el número de estaciones sin gran dificultad.
- El tiempo de acceso es moderado incluso en situaciones de mucho tráfico.
- La velocidad de la red es buena ya que no hay contienda por el medio físico. Sólo se está limitado por el más lento de los nodos, el expeditor, receptor o la velocidad de conexión.
- Permite utilizar diferentes medios de comunicación.

Inconvenientes

- Una estructura de anillo en su más pura configuración es altamente susceptible al fallo de un nodo. Un fallo en cualquier parte del anillo de comunicación deja bloqueada a la red en su totalidad. Si el fallo se produce en una de las estaciones del anillo, la repercusión en el resto de la red será diferente dependiendo de si se avería o no el

módulo de retransmisión. En el caso de que la estación quede fuera de funcionamiento pero el módulo de retransmisión siga operando con normalidad, la avería sólo afectaría a la estación en cuestión. Pero si lo que falla es el modulo de comunicaciones, el anillo quedaría cortado y la red bloqueada.

- Es difícil incorporar nuevos dispositivos sin interrumpir la actividad de la red.

1.10.3 Topología en bus

En una configuración en bus, todas las estaciones están conectadas a un único canal de comunicaciones por medio de unidades interfaz.

El bus, como en el caso de anillo, prescinde de la necesidad de un nodo central de las primeras redes de topología estrella. El bus es muy conveniente para las redes debido a su bajo costo, pero está limitado en cuanto a la distancia.

En las redes con estructura en bus, a diferencia de las de anillo, cada nodo no actúa como rependedor de los mensajes, sino que simplemente ha de reconocer su propia dirección para captar aquellos mensajes que viajan por el bus, y van dirigidos a él. Cuando una estación deposita un mensaje en la red, esta información es difundida a través del bus y todas las estaciones están capacitadas para recibirla.

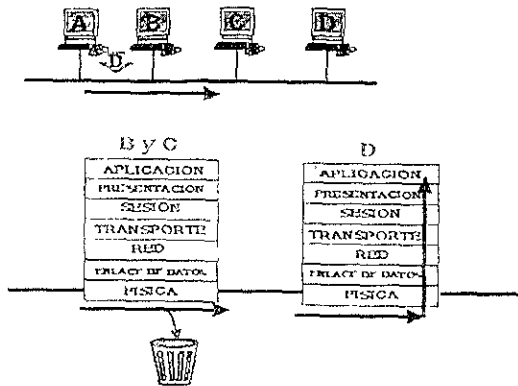


Figura 1-16 Topología en Bus

Ventajas

- Las redes con topología en bus son sencillas de instalar y se adaptan con facilidad a las características de terreno o local. Presentan una gran flexibilidad en lo referente a aumentar o reducir el número de estaciones de la red. Ello unido a su facilidad de uso, hace que esta topología haya sido elegida por numerosos fabricantes.
- Son de bajo costo.
- Es particularmente adecuada para tráfico muy alto.

Inconvenientes

- El fallo en una estación aislada sólo repercutirá en los mensajes a ella vinculados, siendo su efecto nulo en el resto de la red. Una ruptura en el bus, en cambio, deja la red dividida en 2 segmentos inutilizables totalmente.
- La red en sí es fácil de intervenir con el equipo adecuado, sin perturbar el funcionamiento normal de la misma.
- El sistema no reparte equitativamente los recursos.
- A veces los mensajes interfieren entre sí.

1.11 Estándares de comunicación

Es posible tener diferentes protocolos de comunicación para cada enlace de comunicaciones, para cada compañía o para cada tipo de computadora. Estas posibles situaciones han ocurrido en el pasado, y en una reducida extensión, continúan en el presente. Fabricantes de computadoras y de equipo de comunicaciones, así como los usuarios han reconocido que existe una ventaja al manejar un sólo estándar, pues permite la intercomunicación de una mayor cantidad de dispositivos. Los usuarios y fabricantes se han juntado en varios grupos para producir especificaciones detalladas de protocolos que sean utilizados más allá de los alcances de los confines de una sola compañía.

1.11.1 Importancia de los estándares en las comunicaciones de datos.

Los estándares son importantes en muchas industrias, no únicamente en la de comunicaciones. Por ejemplo, la industria eléctrica tiene estándares para conectores y seguridad eléctrica. La industria de los ferrocarriles tiene estándares para la distancia entre rieles. La industria del video tiene estándares para el formato de información que es almacenado en *video-cassette*.

En las comunicaciones de datos, la incompatibilidad entre diferentes protocolos provoca la incomunicación entre las estaciones. Una razón secundaria para la estandarización que no únicamente aplica a las comunicaciones de datos sino a la industria de la computación en su conjunto es el crecimiento de las economías de escala. Mientras mayor sea la estandarización de un protocolo, mayor será su mercado. Como un ejemplo podemos mencionar a las interfaces Ethernet de las redes locales. Cuando este tipo de interfaz LAN fue introducido al mercado tenían un precio de 5,100 dólares. Los diferentes métodos de producción para estas interfaces dio como resultado un mayor volumen de ventas de las interfaces Ethernet, actualmente estas tarjetas cuestan alrededor de 8 dólares.

1.11.2 Estándares y organizaciones de normalización

Los estándares son de 2 tipos. el primero son los estándares "reales" realizados por comités nacionales e internacionales llamados estándares *de jure*, y los segundos son los protocolos producidos por fabricantes, los cuales son adoptados paulatinamente por varios fabricantes, convirtiéndose en un estándar *de facto*. Como ejemplo de este último, tenemos al lenguaje producido por el fabricante de modems: Hayes. Algunas veces el estándar *de facto* se convierte en un estándar *de jure* como en el caso de las redes LAN token-ring, originalmente un estándar de IBM, actualmente un estándar internacional aceptado por la organización de estándares internacionales (ISO).

Existe un gran número de cuerpos de normalización que van desde organizaciones nacionales tales como ANSI (American National Standard Institute), a las organizaciones

internaciones tales como la International Telecommunications Union (ITU) (anteriormente CCITT) y la ISO (International Standards Organization).

Una observación general es que mientras mayor sea el número de gente y organizaciones que apruebe los estándares, mayor será el tiempo en producir los estándares. De esta manera algunos estándares específicos a un fabricante o estándares nacionales son utilizados por varios años antes de que se conviertan en un estándar internacional. Algunos fabricantes introducen nuevos protocolos esperando obtener alguna ventaja competitiva por su mayor desempeño sobre los estándares y también esperando que su protocolo se convierta primero en un estándar *de facto* y después en un estándar *de jure*.

En los últimos años los cuerpos de estándares internacionales han reconocido que el ciclo de revisiones de 4 años es muy lento para varias áreas de tecnología que se mueven rápidamente por lo cual han introducido ciclos de aprobación menos lentos en muchas áreas. Aun así existen áreas importantes en donde los estándares *de facto* son más comunes que los estándares *de jure*. La mayor de estas áreas es el INTERNET por sí mismo. Los protocolos de INTERNET son aprobados por la Internet Engineering Task Force (IETF), la cual, aunque tiene alguna representación internacional, no es una organización de normalización internacional sino un grupo estadounidense muy grande de INTERNET. Por supuesto esto tiene una ventaja, permitir cambios más rápidos de los protocolos en respuesta al cambio en las necesidades.

1.12 Características de las redes de control distribuido

Las redes de control distribuido tienen una gran similitud con las redes de área local. Algunas de las características que comparten las 2 redes son :

- Un medio de comunicaciones común a través del cual todos los dispositivos pueden compartir información, independiente del lugar físico donde se encuentre el dispositivo. En la mayoría de los casos, las redes están contenidas dentro de una reducida área física, que puede ser un edificio, una empresa, una universidad, etc.

- Una distancia entre estaciones relativamente corta, entre unos metros y varios kilómetros (2 km ó 3 km) aunque la distancia puede ser mayor utilizando dispositivos de transmisión especiales (repetidores).
- Todos los dispositivos pueden comunicarse con el resto, y algunos de ellos pueden funcionar independientemente.
- Un sistema fiable, con un índice de errores muy bajo. Las redes locales disponen normalmente de su propio sistema de detección y corrección de errores de transmisión.
- Flexibilidad y facilidad de uso.
- Las redes tienen topologías simples como anillo, estrella y bus
- Las redes no tienen generalmente un controlador central que gobierna el acceso al medio. La omisión de un controlador centralizado significa que no existe un componente de la red que es crítico en su operación.

1.13 Diferencias entre las redes de control y las redes LAN de computadoras

Algunas de las diferencias entre las redes de control y las redes locales de computadoras son:

- Las redes LAN implementan la mayoría de las capas del modelo OSI, las redes de control están basadas en microprocesadores de menor poder de procesamiento, por lo cual generalmente únicamente se implementan 3 capas del modelo OSI: capa física, capa enlace de datos y capa de aplicación, con las cuales se realizan las principales funciones de comunicación:
 - Medio físico de interconexión
 - Detección y control de errores
 - Control de flujo
 - Establecimiento y liberación de conexión

- Mecanismo para reconocer la recepción de unidades de datos.
 - Mecanismos para manejar mensajes perdidos y dañados
 - Métodos para indicar la posición de una unidad de datos en una secuencia de unidades de datos (números de secuencia)
 - Métodos de direccionamiento en las unidades de datos.
- En las redes de control distribuido existe una mayor cantidad de servidores que de clientes. Los servicios ofrecidos por un nodo son del tipo "Lectura del Sensor de temperatura".
 - En las redes de control se suelen personalizar los protocolos para aplicaciones específicas, a diferencia de la redes de computadoras en donde la tendencia es utilizar protocolos abiertos, tales como TCP/IP.

1.14 Criterios de diseño

Con base en lo expuesto en los dos últimos apartados, esta sección establece los criterios de diseño del protocolo de comunicaciones para dispositivos de 8 bits.

- Sistema descentralizado.
- Bajo costo.
- Multimaestro.
- Facilidad de implementación.
- Menor utilización de los recursos internos del microcontrolador.
- Confiabilidad, en lo que se refiere a la transmisión/recepción de datos.
- Minimización del tiempo de procesamiento asociado con el protocolo.

CAPÍTULO 2

2. Diseño de la red

Las redes de control distribuido son más simples que las redes de computadoras como TCP/IP, Netware, NT. Por lo cual en vez de implementar las 7 capas del modelo OSI generalmente se implementan únicamente 3: capa física, enlace de datos y aplicación, con las cuales se cubre la mayoría de las funciones de comunicación. En este capítulo se explicará el diseño de cada una de estas capas.

2.1 Diseño de capa física

Para conectar varios dispositivos en red existen muchas opciones. Varias áreas industriales han desarrollado sus propios estándares de red cubriendo en algunos casos la mayoría de las capas del modelo OSI. El uso de estas tecnologías es complejo y caro a la vez. Sin embargo, actualmente, la mayoría de los microcontroladores incluyen un UART y sólo requieren el uso de una interfaz que permita conectar varios dispositivos a una gran distancia. En el caso de los microcontroladores que no cuenten con un UART se puede añadir un circuito integrado a un relativo bajo costo, o si el espacio del circuito integrado es limitado, se puede implementar el UART en software.

Sin embargo, comparado con otras tecnologías de red, los protocolos de red basados en UART tienen sus desventajas. Los dispositivos operan a una menor velocidad, con velocidades típicas de menos de 200 kbps. Además los UART's proveen de menos capacidades de red que los controladores CAN ó Ethernet. Para implementar la mayoría de las funciones de la red es necesario programarlas en software, en CAN ó Ethernet, los chips controladores de éstos estándares frecuentemente evitan la necesidad de software adicional.

En nuestro caso, tenemos como criterio de diseño crear una red de bajo costo, por lo cual la red basada en UART es la mejor opción. Además si se requiere agregar a la PC

como un nodo de red ó como un monitor de ésta, la interfaz resulta muy sencilla, debido a que la mayoría de las PC's cuentan con un puerto serial basado en UART.

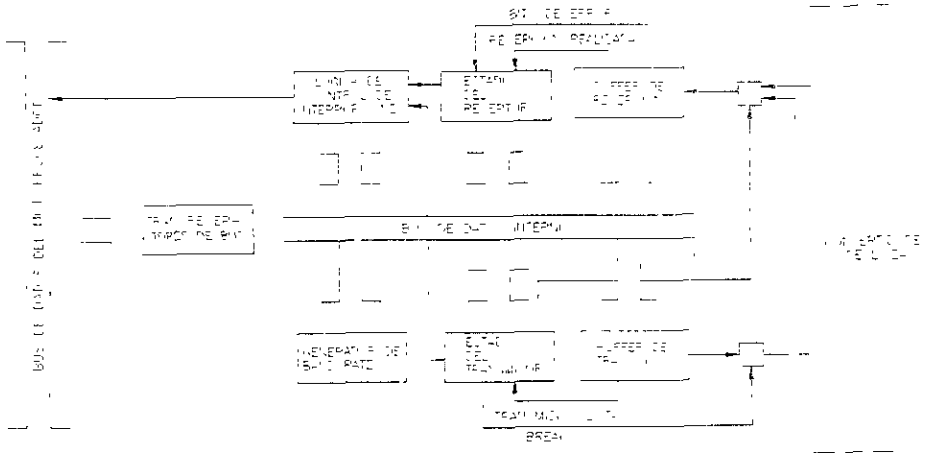


Figura 2-1 Diagrama de bloques interno de un UART

Los UARTS generan una señal digital serial que consiste de 2 valores: MARK (usualmente un estado alto TTL) y un SPACE (un estado bajo TTL). La capa física debe especificar de que manera son representados estos valores en la red. La interfaz entre un UART y la red puede ser tan simple como un buffer de colector abierto. Sin embargo, un transceptor especial o un modem, que generan un señal apropiada para el medio de comunicación, es más común. La Tabla 2.1 presenta una lista parcial de los estándares de señalización y metodologías frecuentemente utilizados con protocolos UART.

Estándares de Señalización.	Descripción
Bell 103, Bell 202,	Usado por modems telefónicos : V.32, V.34, etc.
RS-232C (V.28)	Para conexiones de baja velocidad, conexiones punto a punto de distancia corta.
RS-422 (V.11)	Usa transmisión balanceada para obtener mayores velocidades y distancias mayores.
RS-485	Similar a RS-422 ; permite manejar hasta 32 maestros conectados en un bus de comunicación ; comúnmente utilizado en aplicaciones de adquisición de datos y control distribuido.

Tabla 2.1 Metodologías de Señalización para protocolos basados en UART

2.1.1 Modems telefónicos

Un modem es un dispositivo electrónico que envía una cadena de 1s y 0s sobre la línea telefónica como ondas senoidales moduladas , esto es, una cierta característica de la señal senoidal es modulada (cambiada) de acuerdo a el nivel lógico transmitido. Un modem en el otro extremo de la línea recibe la señal modulada y la demodula, esto es, la convierte a señales digitales. Las frecuencias de las señales senoidales moduladas permanecen dentro del ancho de banda de las líneas telefónicas (3KHz).

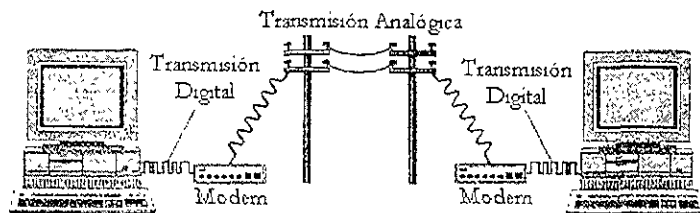


Figura 2-2 Conexión entre microcomputadoras via modem

El módem realiza 5 funciones básicas

- 1 Mediante la señal RS-232 de transmisión de datos (TD), el modem recibe los datos digitales señales provenientes del DIT (ver sección 2.1.2).

2. Modula los datos en una portadora analógica y los transmite sobre un medio de comunicación analógico.
3. Demodula la portadora analógica, que recibe del medio de comunicaciones, y los convierte en datos digitales.
4. Transfiere los datos seriales al DTE, mediante una señal RS-232 de recepción de datos (RD).
5. En sistemas síncronos, el DCE suministra una señal de reloj al DTE.

Para que un enlace de comunicaciones funcione apropiadamente, los modems en los extremos del enlace deben ser compatibles entre sí. Deben usar el mismo tipo de modulación y el mismo protocolo, y deben operar a la misma velocidad. Todos los modems construidos con el mismo estándar idealmente deben ser capaces de comunicarse uno con el otro.

Existen literalmente docenas de estándares de modems, y cada año se introducen más. En la Tabla 2.2 se mencionan los estándares de modems más utilizados comercialmente.

Protocolo	bps(Bits/Segundo)	Kbps(Kilobits/segundo)
V.21, Bell 103	300	0.3
V.22, Bell 212A	1200	1.2
V.22bis	2400	2.4
V.23	1200	1.2
V.32	9600	9.6
V.32bis	14400	14.4
V.34	28800	28.8
V.34+	33600	33.6
V.90	56000	56

Tabla 2.2 Estándares de comunicación de modems

2.1.2 EIA-232.

En 1962, la EIA (Electronic Industry Association) creó el estándar RS-232 para fijar las bases de la comunicación entre el equipo "terminal de datos", DTE (Data Terminal Equipment), y el equipo "comunicación de datos", DCE (Data Communication Equipment). Los dispositivos considerados como DTE son las terminales y computadoras. Un ejemplo de DCE es un modem.

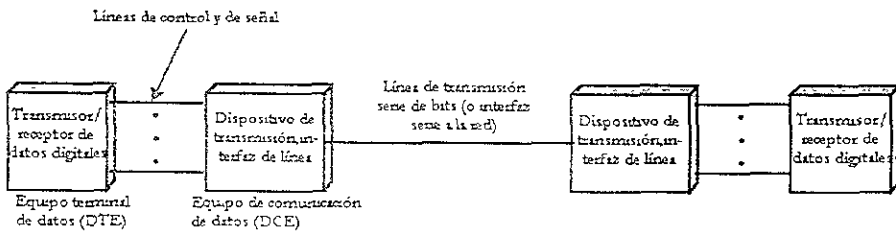


Figura 2-3 Interfaz genérica al medio de comunicación

Para facilitar las cosas, tanto a los usuarios, como a los fabricantes de equipos para el procesamiento de datos, se han desarrollado normas que especifican la naturaleza de la interfaz entre el DTE y el DCE. Una de ellas es el estándar EIA-232 creado por la Electronic Industry Association (EIA) en 1963. Se dice que EIA-232 es un estándar completo debido a que contiene las siguientes especificaciones:

- **Eléctrica**—Aquí se define la señalización entre el DTE y el DCE.

Algunos parámetros que se definen aquí son:

1. Un "Space" (lógico) se representa entre +3 y -25 V.
2. Una "Mark" (lógico) se representa entre -3 y 25 V.
3. Un region entre -3 y 8 no está definido.
4. Un voltaje de circuito abierto nunca debe de ser de 20 V.
5. El elemento de datos de 10 de serie es de 5 bit. La interfaz debe ser capaz de transferir estos datos.

- **Mecánica** : Especifica el tipo de conectores que utilizan los dispositivos involucrados en la comunicación. En el caso de EIA-232 se describe un conector de 25 "pines" de $47.04 \pm .13$ mm de ancho (de centro de tornillo a centro de tornillo), con todas las demás dimensiones especificadas con igual precisión. La hilera superior tiene los pines numerados de 1 a 13 (de izquierda a derecha); la hilera de abajo tiene los pines numerados de 14 a 25 (también de izquierda a derecha).
- **Funcional** : Esta especificación dice qué circuitos se conectan a cada uno de los 25 pines, y su significado. En la Tabla 2.3 se muestran los 9 pines que casi siempre están implementados en la PC.

ABREVIACIÓN	NOMBRE COMPLETO	DTE-DCE
TD	Transmit Data	→
RD	Receive Data	←
RTS	Request To Send	→
CTS	Clear to Send	←
DSR	Data Set Ready	←
DTR	Data Terminal Ready	→
RI	Ring Indicator	←
CD	Carrier Detect	←

Tabla 2.3 Asignación de las terminales para EIA-232

- **Procedimientos** : La especificación de procedimientos es el protocolo, esto es, la secuencia permitida de sucesos. El protocolo se basa en pares de acción-reacción. Por ejemplo, cuando la terminal activa RTS (Request to Send), el modem contesta con CTS (Clear to Send), si está en condiciones de aceptar datos. También existen pares de acción-reacción similares para otros circuitos.

En general, las características más importantes del estándar EIA-232 son :

- Interfaz balanceada
- Interfaz punto a punto
- Rango de salida del transmisor muy grande (± 25 Volts)
- Interfaz totalmente definida
- Máxima velocidad de transmisión de 20 kbps.

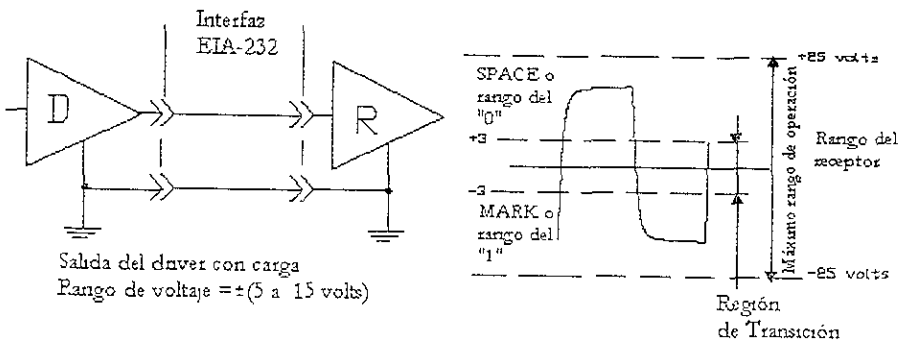


Figura 2-4 Interfaz EIA-232

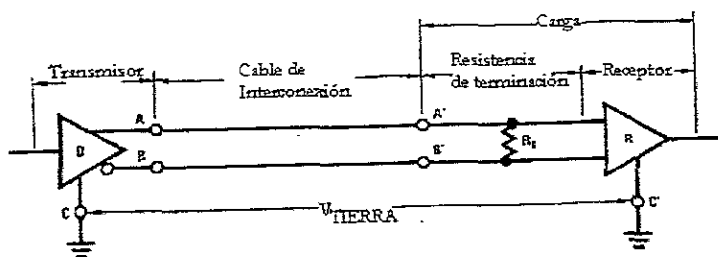
Por otro lado, el EIA-232 ha estado en circulación desde hace varios años. Gradualmente, la limitación de la velocidad de transmisión de datos a no más de 20 kbps, la longitud máxima del cable de 15 metros y su falta de inmunidad al ruido se han vuelto cada vez más intolerables, por lo cual la EIA creó un nuevo estándar. El estándar nuevo, llamado EIA-449, en realidad es tres estándares en uno. Las interfaces mecánica, eléctrica y de procedimiento se especifican en EIA-449, pero la interfaz eléctrica se especifica en 2 estándares diferentes. El primero de ellos, EIA-423, es parecido al EIA-232 en que todos sus circuitos comparten una tierra común. El estándar EIA-423 especifica un rango de voltaje ± 5 volts contra los ± 25 volts de EIA 232. EIA 423 fue intentado para usarse en las señales de control y estado del estándar EIA-449. El segundo estándar eléctrico, EIA-422 usa en contraste la transmisión balanceada, en la que cada uno de los circuitos

principales requiere 2 hilos, sin tierra común. Esto permite usar EIA-422 a velocidades de hasta 2 Mbps por cables de 60 metros.

En nuestro caso la elección de EIA-232 quedó descartada debido a que es un estándar punto-a-punto, que no permite la conexión de varios maestros en una red.

2.1.3 EIA-422

En 1978, la EIA adoptó el estándar EIA-422 para resolver los problemas de distancia y ruido asociados con EIA-232. A diferencia de éste último, EIA-422 es un estándar balanceado.



- R_t = Resistencia de terminación
- V_{TIERRA} = Diferencia del potencial de tierra
- A, B = Interfaz del transmisor
- A' B' = Interfaz de la carga
- C = Tierra del circuito transmisor
- C' = Tierra del circuito de carga

Figura 2-5 Interfaz balanceada

En una interfaz no balanceada se utiliza un voltaje de línea que se compara con una señal de referencia (tierra), para determinar un 1 o un 0 lógico. La transmisión balanceada requiere 2 conductores por señal, con estados lógicos definidos por la diferencia de potencial entre ellos. La tabla de verdad es similar a esta :

$$V_A - V_B < -0.2V = 1$$

$$V_A - V_B > +0.2V = 0$$

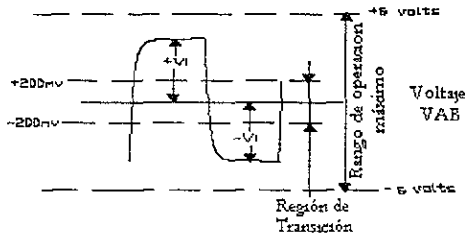


Figura 2-6 Entrada diferencial en el receptor EIA-422

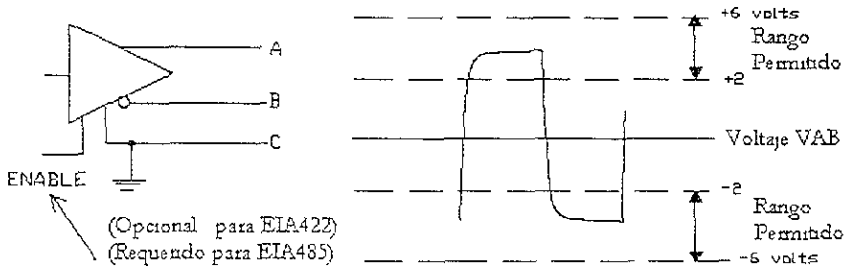


Figura 2-7 Salida del voltaje diferencial de transmisor

Las interfaces balanceadas disponen de una inmunidad frente al ruido mucho mayor que las interfaces desbalanceadas, como EIA-232 y el EIA-423. Debido a que se maneja un voltaje diferencial, la interfaz no es afectada por las diferencias de voltaje de tierra entre el receptor y transmisor.

Las interfaces de alta velocidad, por lo general son equilibrados, debido a que cuando se emplean voltajes bajos, el tiempo necesario para variar el voltaje es menor. La sensibilidad del receptor será mayor cuando se emplean interfaces balanceadas, dado que los niveles de ruido son más bajos, por lo que es posible transmitir señales a mayor distancia cuando se emplean interfaces balanceadas que cuando se emplean interfaces no balanceadas.

Las características más comunes de EIA-422 son las siguientes:

- Interfaz balanceada

- Rango de voltaje en modo común ($\pm 7V$)
- Maestro-Esclavo (Operación de hasta 10 receptores)

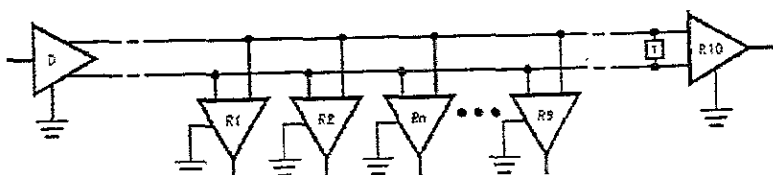


Figura 2-8 Conexión maestro-esclavo EIA-422

- Velocidad máxima de 10 Mbps a 12 metros.
- Longitud máxima de 1200 metros a 100 kbps.

En el caso de nuestra aplicación, EIA-422 tiene como desventaja que no puede manejar múltiples maestros (drivers) en el bus.

2.1.4 -EIA-485

EIA-485 es un estándar que al igual que EIA-422 sólo define la especificación eléctrica es una interfaz balanceada que tiene muchas semejanzas con EIA-422. A diferencia de éste que únicamente permite conectar un maestro con hasta 10 esclavos, EIA-485 permite conectar hasta 32 cargas transreceptoras en el mismo bus. La mayoría de los chips controladores EIA-485 son compatibles con versiones previas de EIA-422, pero el proceso inverso no es permitido. La EIA-485 incluye limitaciones de corriente en caso de contienda, y los rangos de voltaje en modo común se extienden de +12 V a -7 V.

La Figura 2-9 muestra una red multipunto de 2 alambres típica de EIA-485. Podemos observar que el bus es terminado en ambos extremos de la línea, pero no en los puntos intermedios. La línea de señal de tierra es recomendada en un sistema EIA-485 para mantener el voltaje en modo dentro de un rango de -7 a +12 volts.

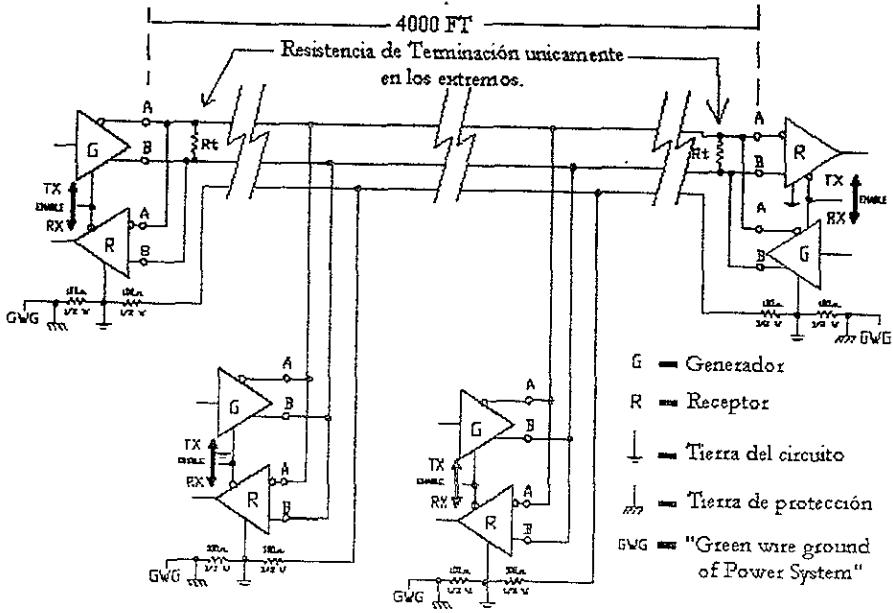


Figura 2-9 Red multipunto EIA-485 de 2 alambres

Una red EIA-485 puede también ser conectada en un modo de 4 alambres (Figura 2-10). En una red de 4 alambres es necesario que un nodo sea un nodo maestro y el resto esclavos. La red es conectada de tal manera que el nodo maestro comunica a los nodos esclavos.

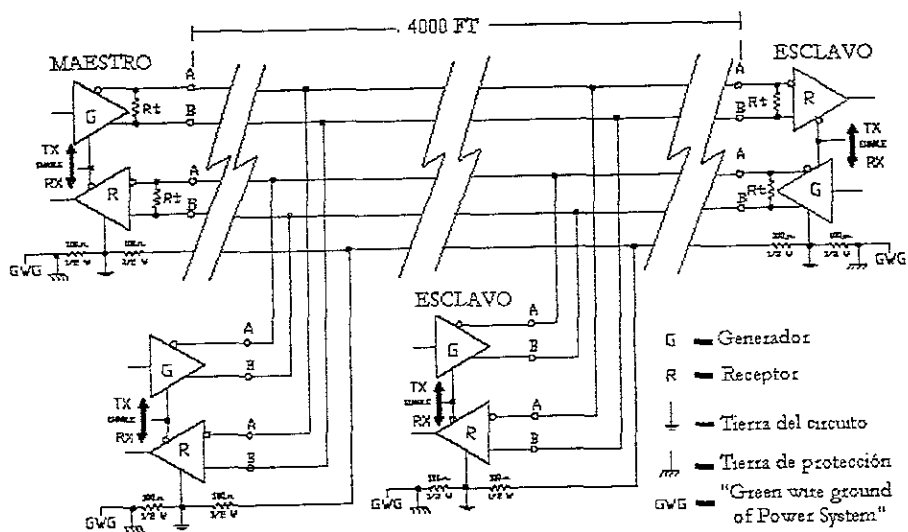


Figura 2-10 Red multipunto EIA-485 de 4 alambres

A continuación se mencionan las principales características de EIA-485 :

- Interfaz balanceada
- Operación multimaestro.
- Operación con una sola fuente de +5V
- Rango de voltajes de modo común (+12 a -7V)
- Permite el uso de 32 cargas transreceptoras.

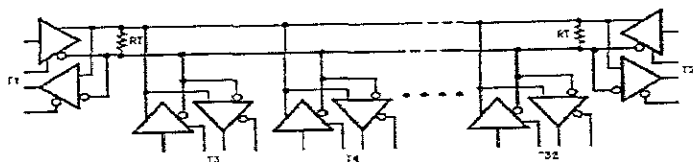


Figura 2-11 Conexión multimaestro EIA-485

- Velocidad máxima de 10 Mbps a 12 metros.
- Longitud máxima de 1200 metros a 100 kbps.

En el Apéndice A se encuentran algunas de las más importantes características del estándar EIA-485 , así como también las gráficas de velocidad contra longitud de EIA-232, EIA-422, EIA-423, EIA-485.

Para nuestro diseño, el EIA-485 es la interfaz indicada debido principalmente a que permite conectar varios nodos maestro en el mismo bus (configuración half-duplex de 2 alambres). Un circuito integrado que cumple con el estándar EIA-485 es el MAX485.

2.2 Diseño de capa enlace de datos

Habiendo diseñado la capa física cuya características principales son :

- Estándar EIA-485
- Comunicación half-duplex por medio de 2 alambres.

Toca el turno del diseño de la capa de enlace de datos. Este diseño es largo y complicado debido a que se incluyen las funciones más importantes del protocolo. La capa de enlace de datos transmite los mensajes, usualmente empaquetados, entre los nodos de red, que produce la capa superior. En protocolos complejos, esta capa superior es la capa de red, en nuestro caso es la capa de aplicación. La especificación de la capa de enlace de datos debe incluir :

- MAC Control de acceso al medio (Media Access Control). De qué manera es compartido el medio de comunicación
- Delimitación de mensajes.
- La estructura de los mensaje de red (paquetes).
- Detección de errores de transmisión
- Manejo de errores

- Control de flujo.

2.2.1 Selección de protocolo de acceso al medio

Uno de los factores que cualquier red debe definir es el protocolo de acceso al medio. En este tipo de protocolo, el asunto clave es la manera de determinar quién puede usar el canal cuando hay competencia por él. A continuación se analizan los principales protocolos MAC que se utilizan en las redes de control distribuido :

2.2.1.1 Polling (Llamada Selectiva)

Polling es uno de los protocolos más populares de los sistemas de control debido a que es el protocolo más sencillo de definir y de implementar.

El método polling (llamar a una estación para que transmita o se disponga a recibir un mensaje), (véase Figura 2-12) requiere un control centralizado de todas las estaciones de la red.

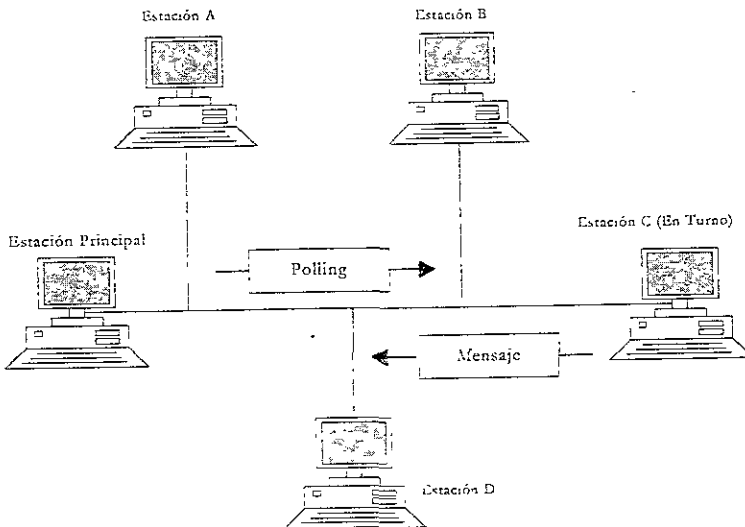


Figura 2-12 Polling

Una red que utilice el método polling tiene dos tipos de estaciones : la estación principal, y las estaciones secundarias conectadas a ella. Cada estación secundaria dispone de un

área de almacenamiento temporal (o buffer). Cuando una estación secundaria desea transmitir un mensaje, lo envía a este buffer, donde permanece hasta que la estación central pide que le sea transmitido. La estación central llama a las estaciones secundarias de una en una para determinar si hay alguna que tenga un mensaje para transmitir. Si la respuesta es afirmativa, se autoriza a la estación secundaria para que transmita inmediatamente, o se le asigna un determinado tiempo para que lleve a cabo la transmisión. Si la estación no tiene mensajes para transmitir contesta mediante un pequeño mensaje de control. En algunas redes, cuando un nodo no tiene datos que transmitir, éste pasa el control a la siguiente estación secundaria, ahorrando así algo de tiempo y haciendo que el rendimiento sea más alto.

La mayoría del software del protocolo es almacenado en el maestro y el trabajo de comunicación de los nodos esclavos tiende a ser menor. Este protocolo es ideal para sistemas de adquisición de datos centralizados en donde la comunicación punto a punto no es requerida. Sin embargo, en varios sistemas de control distribuido, la falla en el nodo maestro (o el costo de instalación de hardware redundante) no es aceptable. Adicionalmente, el protocolo de encuesta consume considerable ancho de banda sin importar la carga de red. Estos protocolos han sido estandarizados por instituciones militares (MIL-STD-1553B) para comunicaciones de subsistemas de aviones. La principal desventaja de este tipo de protocolo es que tiene un control centralizado, en donde si el sistema maestro sufre un daño o no existe comunicación con este nodo toda la red se viene abajo.

2.2.1.2 Token Bus

Es un método de acceso al medio en donde únicamente un nodo tiene permiso para transmitir en un determinado tiempo. Este permiso es comúnmente referido como estafeta ("token"). El nodo que posee la estafeta puede transmitir un paquete a otro nodo. La estafeta es pasada de nodo a nodo como si las estaciones estuvieran conectadas en forma de anillo. Un nodo que no necesite transmitir pasa la estafeta a el siguiente nodo. Algunos ejemplos de redes de microcontroladores que utilizan este tipo de red

son : ARCNET (Attached Resource Computer Network) y MAP (Manufacturing Automating Protocol).

El rendimiento de este protocolo en condiciones de carga baja es malo, mientras que en condiciones de tráfico normal y alto, el rendimiento se puede calificar de excelente. Los tiempos de espera de este tipo de redes suelen ser bastante constantes y se pueden calcular fácilmente bajo determinado volumen de tráfico.

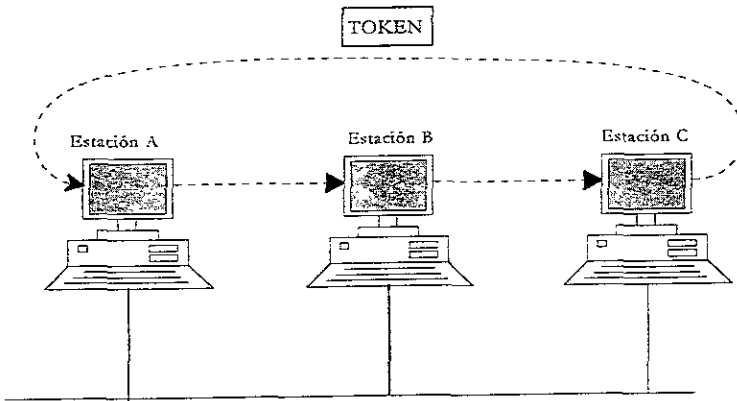


Figura 2-13 Token-Bus

La principal desventaja de este protocolo es su dificultad de implantación, por ejemplo algunas de las situaciones que se tienen que tomar en cuenta en su realización son [Tanenbaum] :

- a) Periódicamente cada poseedor de la estafeta envía un marco de SOLICIT_SUCESSOR para solicitar peticiones de estaciones que deseen unirse al anillo. Se necesita un temporizador de estaciones para saber cuando se envía este marco.
- b) ¿Qué sucede si 2 o más estaciones solicitan entrar al mismo tiempo? En la implantación se resuelve con un protocolo de conteo descendente binario.

- c) Salida del Anillo : Una estación x , con su sucesor S y su antecesor P , deja el anillo enviando a P un marco SET_SUCESSOR
- d) El inicio del anillo es un caso especial de adición de estaciones nuevas. Se enciende la primer máquina, periódicamente, emite solicitudes de conexión de estaciones nuevas. Si dos máquinas se encienden simultáneamente, se utiliza el protocolo de conteo descendente.
- e) Pérdida de estafeta o desactivación de sucesor. Tras pasar la estafeta, una estación escucha si su sucesor transmite un marco o pasa la estafeta, si no lo hace, se le pasa la ficha por segunda vez. Si también esto falla, la estación transmite un marco WHO_FOLLOWS especificando la dirección de su sucesor. Cuando el sucesor de la estación fallida ve un marco nombrando a su antecesor, responde con un marco SET_SUCESSOR a la estación a que le falla el sucesor, nombrándose como nuevo sucesor. Si también estaba abajo esa máquina, la estación envía un marco SOLICIT_SUCESSOR_2 para ver si queda alguna estación viva.
- f) El poseedor de la estafeta se cac y se lleva la estafeta con él. Cada estación tiene un temporizador que se restablece cada vez que aparece un marco en la red. Cuando este alcanza un valor límite, la estación envía un CLAIM_TOKEN, por medio de conteo descendente se decide a el dueño del token
- g) Estafetas Múltiples. Si una estación poseedora de la ficha nota una transmisión desde otra estación, se deshace de su ficha

La implantación del HDLC de token bus llamada 802.4 mantiene 10 temporizadores y más de 24 variables de estado internas, simplemente para MAC.

2.2.1.3 Conteo descendente binario

En este tipo de protocolo también conocido como protocolo de dominio de bit, todos los nodos esperan que el canal se encuentre HDLC antes de transmitir un mensaje. Los nodos compitiendo (transmitiendo simultáneamente) resuelven la contienda mediante la transmisión de una señal basada en su valor de identificación única de nodo. El medio de

transmisión debe de tener la característica que un valor (normalmente el "1") sobrescriba un valor contrario (un "0"). Durante este tipo de transmisión un nodo se sale de la competencia si detecta una señal contraria en lugar de la suya. Figura 2-14

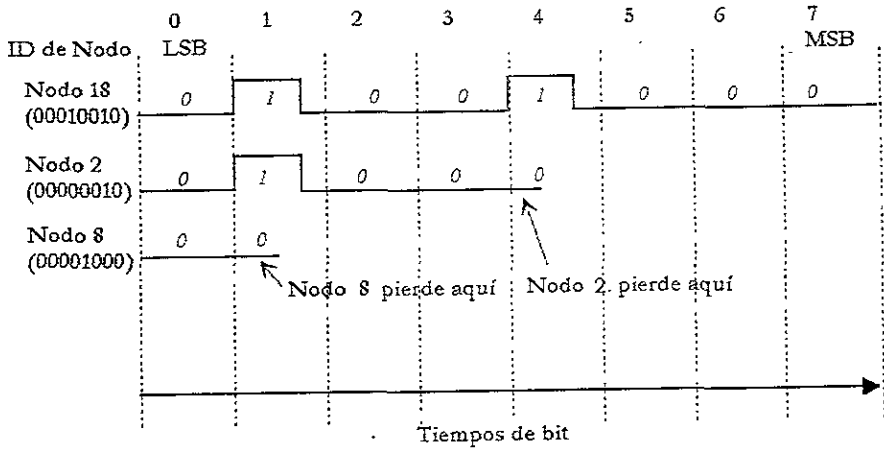


Figura 2-14 Conteo descendente binario

Este protocolo tiene buena eficiencia pues el arbitraje se realiza mediante el ID de los mensajes. Usando este protocolo, BOSCH desarrolló la especificación de CAN (Controller Area Network) para aplicaciones automotrices ; PHILIPS desarrolló el bus I2C y el SAE creó estándar SAE J-1850.

La principal desventaja de este protocolo es que necesita el uso de controladores especiales, además de una interfaz relativamente compleja con el microcontrolador , lo cual implica un mayor costo.

2.2.1.4 CSMA/CD Carrier Sense Media Access / Collision Detect (Protocolo de acceso múltiple con detección de portadora y detección de colisiones)

En este tipo de protocolos, un nodo de red espera que el medio de comunicaciones se encuentre IDLE antes de poder transmitir (como en conteo descendente binario). Si varias estaciones transmiten casi simultáneamente, los mensajes chocan, como se observan en la Figura 2-15. Los nodos deben detectar esta colisión, y resolverla

esperando un periodo de tiempo aleatorio o predefinido antes de volver a transmitir. El protocolo Ethernet usado en las LAN's de computadoras está basado en este protocolo.

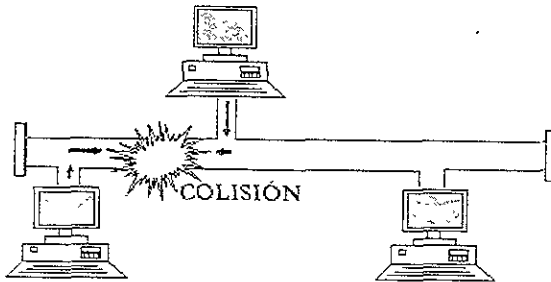


Figura 2-15 Acceso múltiple con detección de portadora/ Detección de colisiones (CSMA/CD)

La ventaja principal de este protocolo es que en un principio puede soportar un número limitado de nodos que no requieren reconfigurar completamente la red para incluir nuevas estaciones y definir una nueva secuencia de circulación del token. De esta manera, CSMA/CD permite la inclusión y exclusión de nodos de red sin requerir la inicialización o reconfiguración de la red. Para condiciones de tráfico bajo y medio, su desempeño se considera muy bueno. Pero para cargas muy altas, la red tiende a ser muy inestable y los tiempos de respuesta aumentan considerablemente.

En la tabla 2.4 se muestra un cuadro comparativo entre las 4 propuestas de solución antes mencionadas. Los factores de evaluación que se presentan son complejidad de realización, sistema descentralizado y costo.

MAC	VENTAJAS	DESVENTAJAS
Polling	<ul style="list-style-type: none"> - Es sencillo de realizar - No requiere hardware adicional - La carga de procesamiento del protocolo es menor en los esclavos que en los maestros 	-Es un esquema Maestro-Esclavo, centralizado, en donde la falla del maestro provoca la caída de la red.
Token Bus	<ul style="list-style-type: none"> - Su tiempo de espera es determinístico - Su rendimiento es alto en condiciones de carga elevada. - Sistema descentralizado 	- Su implementación es compleja, requiriendo una gran cantidad de recursos del procesador.
Conteo descendente binario	<ul style="list-style-type: none"> - Su rendimiento es elevado. - Sistema descentralizado 	-Es relativamente caro, pues requiere el uso de CI especiales.
CSMA/CD	<ul style="list-style-type: none"> - Menor complejidad de realización que Token Bus - Sistema descentralizado - Su costo es relativamente bajo. 	<ul style="list-style-type: none"> - Su tiempo de espera no es determinístico. - Su rendimiento es bajo cuando la carga es elevada

Tabla 2.4 Ventajas y Desventajas de los protocolos de acceso al medio

Analizando los requerimientos principales (facilidad de realización, sistema multimaestro y bajo costo), CSMA/CD fue elegido como el protocolo de acceso al medio.

2.2.2 Delimitación de mensajes

En las redes de comunicaciones es frecuente que el transmisor fragmente un bloque de datos en pequeños bloques, transmitiendo los datos usando varias tramas de tamaño menor. Esto es así por las siguientes razones [STALLINGS]

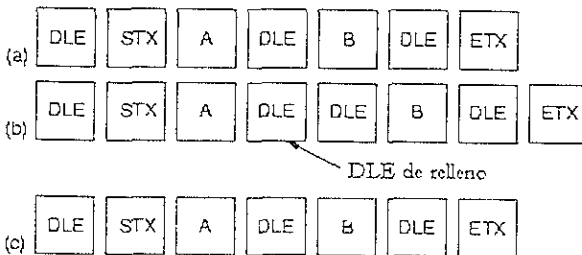
1. El tamaño de la memoria temporal del receptor es limitado
2. Cuanto más larga sea la transmisión, es más probable que haya errores, necesitando en ese caso la retransmisión de la trama completa. Si se usan tramas más pequeñas, los

errores se detectarán antes, y en ese caso se necesitará retransmitir una cantidad de datos menor.

3. En un medio compartido, tal como una LAN, es frecuente no permitir que una estación ocupe el medio durante un periodo largo, evitando así grandes retardos en las otras estaciones que intenten transmitir

En la delimitación de mensajes, o enmarcado, se define de qué manera se identifica el inicio y el final de cada paquete. A continuación se mencionan 2 de los métodos más comunes para enmarcar mensajes :

1. Uso de caracteres especiales : Se utiliza un símbolo para indicar el inicio de un paquete (STX), y, en la mayoría de los casos, otro símbolo indica el fin del paquete (ETX). Para evitar confusión en el receptor, estos símbolos delimitadores no deben de aparecer en el interior del mensaje. Debido a que esta situación muchas veces es imposible de evitar se introduce un tercer caracter especial, DLE, el cual se coloca antes de los caracteres delimitadores, STX y ETX. La secuencia de dos caracteres DLE STX representa el inicio de un paquete , DLE ETX representa el final. Dentro del paquete, todos los caracteres son transmitidos sin modificación, excepto el caracter DLE, el cual es transmitido como una secuencia de dos caracteres DLE DLE.



(a) Datos enviados por la capa superior (b) Datos después del relleno de caracteres por la capa de enlace de datos (c) Datos pasados a la capa superior del lado receptor

Figura 2-16 Enmarcado por caracteres de inicio y fin con relleno de caracteres

Uno de los protocolos más utilizado y a su vez uno de los más antiguos de este tipo de protocolos es el Binary Synchronous Communications (BSC) de IBM.

2. Conteo de caracteres : Este tipo de protocolos usa caracteres especiales junto con un encabezado de tamaño fijo, el cual contiene un campo de número de caracteres. Un ejemplo de este tipo de protocolos es el Digital Data Communications Message Protocol (DDCMP) de Digital Equipment Corporation Figura 2-17.

S Y N	S Y N	Clase	Cuenta (14 bits)	Banderas (2 bits)	Respuesta (8 bits)	Secuencia (8 bits)	Dirección (8 bits)	C R C 1	D A T O S	C R C 2

Figura 2-17 Formato de un mensaje DDCMP

Actualmente la gran mayoría de protocolos de red : TCP/IP, Netware de Novell y NT, utilizan el conteo de caracteres como método de delimitación de mensajes. Con lo que respecta a la realización, el primer método es más complicado que el segundo, pues requiere añadir y buscar caracteres DLE en el transmisor y receptor, respectivamente.

2.2.3 Detección y Corrección de Errores

Los errores de transmisión son inevitables en un sistema de comunicaciones, por lo cual se debe implementar un método de detección de errores.

Los diseñadores de redes han desarrollado dos estrategias básicas para manejar los errores. Una es incluir suficiente información redundante en cada bloque de datos transmitido para que el receptor pueda deducir lo que debió ser el carácter transmitido. La otra estrategia es incluir sólo suficiente redundancia para que el receptor sepa que ha ocurrido un error (pero no qué error) y entonces solicite una retransmisión. La *primer* estrategia usa códigos de corrección de errores ; la segunda usa códigos de detección de errores.

El método de detección de errores es generalmente preferido por varias razones. Una de ellas es que el número de bits adicionales necesitados para implementar un esquema de

detección de errores es mucho menor que el número de bits necesitados para corregir el mismo error. Además la realización de los métodos de corrección de errores es más compleja que los métodos de detección de errores.

Los códigos de corrección de errores son principalmente utilizados en conexiones en donde la retransmisión es imposible ó no práctico. Como un ejemplo de este tipo de sistemas tenemos a los equipos espaciales, en donde la retransmisión es frecuentemente imposible.

Otras de las ventajas de los códigos de detección de errores son :

1. Son sencillos de implementar
2. Sus tiempos de procesamiento son cortos
3. Detectan una gran cantidad de errores.

A continuación mencionaremos los métodos de detección de errores más comunes en las comunicación de datos :

2.2.3.1 Comprobación de Paridad

El esquema más sencillo para detectar errores consiste en añadir un bit de paridad al final del bloque de datos. Un ejemplo típico es la transmisión ASCII, en la que se añade un bit de paridad por cada caracter ASCII de 7 bits. El valor de este bit se determina de tal forma que el caracter resultante tenga un número impar de unos (paridad impar) o un número par (paridad par). Así por ejemplo, si el transmisor está transmitiendo una G en ASCII (1110001) y se utiliza paridad impar, se añadirá un 1 y se transmitirá 11100011. El receptor examina el caracter recibido y, si el número total de unos es impar, supondrá que no ha habido errores. Si un bit (o cualquier número impar de bits) se invierte erróneamente durante la transmisión (por ejemplo, 11000011), entonces el receptor detectará un error. Sin embargo, si dos (o cualquier número par) de bits se invierten, aparecerá un error no detectado.

2.2.3.2 “Checksum” aritmético

En el esquema de suma de comprobación (“checksum”), cada byte de un bloque de datos es tratado como un número de 8 bits independiente, en donde la suma de todos estos números producen el “checksum”. El total puede ser más grande que el tamaño máximo del checksum. Si este es el caso entonces un número de bits deben de ser seleccionados del resultado final, típicamente los n bits menos significativos del “checksum”.

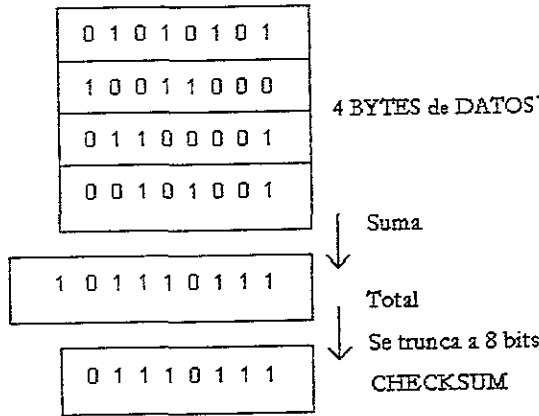


Figura 2-18 Cálculo de checksum

La desventaja de este método es que diferentes mensajes pueden producir el mismo “checksum”. Si un bloque es dañado, se puede dar el caso, que el bloque dañado y el original tengan el mismo “checksum”, con lo cual no se detectaría el error.

2.2.3.3 Comprobación de redundancia cíclica (CRC : Cyclic Redundacy Check)

El CRC es muy poderoso y a su vez es relativamente sencillo de implementar. La técnica del CRC es usada para proteger bloques de datos llamados marcos. Usando esta técnica, el transmisor anexa un secuencia extra de n-bits a cada marco llamada “suma de comprobación” (FCS : Frame Check Sequence). La suma de comprobación almacena información redundante acerca de los datos en el marco que ayuda al receptor a detectar

errores en el marco. El CRC es una de las técnicas más usadas para la detección de errores en la comunicación de datos. Esta técnica ganó su popularidad debido a que combina 3 factores :

- Detecta una gran cantidad de errores.
- Poco tiempo de procesamiento asociado.
- Fácil de implementar.

El algoritmo para calcular la suma de comprobación es el siguiente :

1. Sea r el grado de $G(x)$. Anexe r bits cero al final del marco, para que ahora contenga $m+r$ bits y corresponda al polinomio $x^r M(x)$
2. Divida la cadena de bits correspondiente a $x^r M(x)$ entre la correspondiente a $G(x)$ usando la división módulo 2
3. Reste el residuo (que siempre es de r o menos bits) a la cadena de bits correspondiente a $x^r M(x)$ usando resta módulo 2. El resultado $T(x)$ es el marco con suma de comprobación a transmitirse

$G(x)$: Polinomio generador

$M(x)$: Polinomio que representa el mensaje

Cuatro polinomios se han vuelto estándares internacionales :

$$\text{CRC-12: } X^{12} + X^{11} + X^3 + X^2 + X + 1$$

$$\text{CRC-16: } X^{16} + X^{15} + X^3 + 1$$

$$\text{CRC-CCITT: } X^{16} + X^{12} + X^5 + 1$$

$$\text{CRC-32: } X^{32} + X^{26} + X^{23} + X^{22} + X^{16} + X^{13} + X^{11} + X^{10} + X^8 + X^7 + X^5 + X^4 + X^2 + X + 1$$

Los 4 contiene $x+1$ como factor primo. CRC-12 se usa cuando la longitud de un carácter es de 6 bits. Los otros dos se utilizan con caracteres de 8 bits. Un CRC de 16 bits, como el CRC-16 o el CRC-CCITT, detecta todos los errores sencillos y dobles, todos los

errores con un número impar de bits, todos los errores en ráfaga de longitud 16 o menos, 99.997% de las ráfagas de 17 bits y 99.998 de las ráfagas de 18 bits o mayores.

CRC-16 y CRC-CCITT son ampliamente usados en los E.U.A y Europa respectivamente y dan una adecuada protección a la mayoría de las aplicaciones. Aplicaciones que necesiten protección extra pueden hacer uso del CRC-32. El CRC-32 es utilizado en el estándar de redes locales IEEE-802.

A continuación se presenta un cuadro comparativo de los 3 métodos de errores presentados anteriormente. Actualmente la gran mayoría de protocolos de red usan la técnica de CRC para detectar errores, además existen diferentes algoritmos que permite la computación veloz del CRC.

MÉTODO	VENTAJAS	DESVENTAJAS
Paridad	1.Sencillo de realizar. 2.Muy rápido.	1.Unicamente detecta errores en los que cambia un número non de bits.
Checksum	1.Sencillo de realizar.	1.No detecta todos los errores. 2.Su tiempo de procesamiento es más lento comparado con el método de paridad.
CRC	1.Detecta una mayor cantidad de errores que los dos métodos anteriores. 2.Su realización en hardware es muy rápida.	1.Su realización en "software" es más complicada que en los otros dos métodos.

Tabla 2.5 Ventajas y desventajas de los métodos de detección de errores

2.2.4 Control de errores.

En general cuando se transmite un paquete se pueden presentar dos tipos de errores :

1. Tramas perdidas : que tiene lugar cuando una trama enviada no llega al otro extremo. Por ejemplo, una ráfaga de ruido puede dañar a una trama de tal manera que el receptor no detecte dicha trama.
2. Tramas dañadas : ocurre cuando llega una trama, pero con algunos bits erróneos (modificados durante la transmisión)

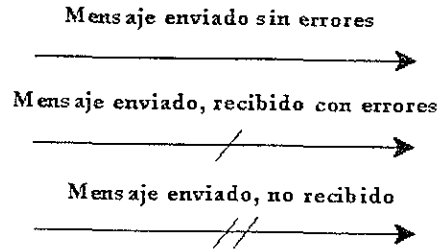


Figura 2-19 Simbología de los diagramas de comunicación entre nodos

Cuando un nodo receptor detecta un error, debe de tomar alguna acción correctiva. Usualmente esto significa la retransmisión del mensaje por parte del nodo transmisor. La estación transmisora debe tener realimentación sobre lo que está ocurriendo en el otro lado de la línea. En algunos protocolos, la estación receptora envía un reconocimiento positivo si no se detectó un error y un reconocimiento negativo si detecta. En otros protocolos, el receptor únicamente reconoce los paquetes sin errores pero los dañados no son reconocidos. Si la estación transmisora no recibe un reconocimiento dentro de un tiempo específico (TIME-OUT), entonces se retransmite el paquete. Después de un número específico de retransmisiones, la estación transmisora deja de intentar transmitir el mensaje. A cada una de las técnicas de control de errores se le asigna un nivel de confiabilidad. Tabla 2.6

NIVEL DE CONFIABILIDAD	DESCRIPCIÓN
1	Garantiza la entrega de exactamente una copia de cada mensaje en el orden transmitido
2	Garantiza la entrega de cuando menos una copia de cada mensaje
3	La entrega no es garantizada

Tabla 2.6 Niveles de confiabilidad

En el caso de la técnica presentada, de retransmisión con reconocimientos positivos y negativos, se ofrece un nivel 2 de confiabilidad. Como se muestra en la Figura 2-20 el problema que tiene esta técnica es la presencia de mensajes duplicados.

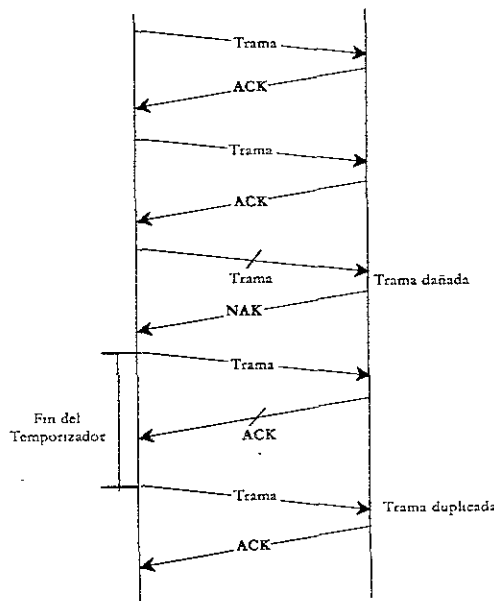


Figura 2-20 Diagrama de comunicación con nivel de confiabilidad 2

Alcanzar el nivel 1 de confiabilidad involucra combinar las técnicas de control de errores de realimentación con un mecanismo de números de secuencia que trabaja como las páginas de un libro. Los números de secuencia pueden manejar paquetes duplicados,

perdidos y fuera de orden. Para usar números de secuencia es necesario establecer una conexión antes de poder transferir datos. Esta conexión es necesaria principalmente para iniciar los números de secuencia. Cuando una capa de la arquitectura de red proporciona este tipo de servicio se dice que ofrece un *servicio orientado a la conexión*.

En el caso de transmisiones half-duplex, el nivel 1 de confiabilidad se logra utilizando: confirmaciones positivas y negativas, números de secuencia (0 y 1), establecimiento de conexión y liberación de conexión. El procedimiento comúnmente utilizado es el siguiente:

Dos nodos de red establecen una conexión virtual, la cual inicia los números de secuencia a 0. La estación origen transmite una única trama, trama 0, y ésta debe esperar la recepción de una confirmación positiva ó negativa (ACK ó NAK). No se podrá enviar ninguna otra trama hasta que la respuesta de la estación destino llegue al emisor. Si en la transmisión no ocurrieron errores entonces el receptor envía una confirmación positiva, ACK0; el receptor incrementa su número de secuencia y el transmisor recibe el ACK0, el cual inicia la transmisión de la siguiente trama, trama 1.

Pueden ocurrir dos tipos de error. El primero, consiste en que la trama que llega al destino puede estar dañada. El receptor detectará esto mediante la utilización de las técnicas de detección de errores mencionadas anteriormente y enviará una confirmación negativa junto con el número de secuencia del mensaje que está esperando. Al recibir este mensaje el lado transmisor reenvía la trama.

El segundo tipo de error puede originarse si la confirmación se deteriora ó el mensaje de datos se pierde. Considérese la siguiente situación: la estación A envía una trama con un número de secuencia 0. La trama se recibe correctamente por la estación B, la cual responde con una confirmación positiva, ACK0. Esta respuesta se deteriora en el camino y se modifica de tal forma que no es identificable por A como tal, en este caso se producirá una expiración del temporizador y se reenvía la trama. La trama duplicada llega a la estación B; debido a que el lado receptor espera el mensaje con el número de

secuencia 1, se da cuenta que el mensaje es duplicado y no lo guarda en memoria ; la estación receptora responde con ACK0.

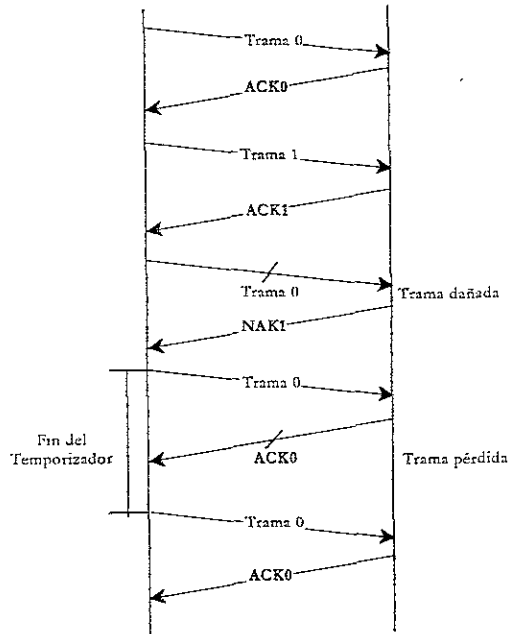


Figura 2-21 Diagrama de comunicación con nivel de confiabilidad 3

2.2.5 Control de flujo

El control de flujo es una técnica utilizada para asegurar que la entidad de transmisión no sobreescriba a la entidad receptora con una excesiva cantidad de datos. La entidad receptora reserva típicamente una zona de memoria temporal para el almacenamiento de los datos. Cuando se reciben, el receptor debe realizar algún tipo de procesamiento antes de pasar los datos al software de los niveles superiores. Si no hubiera procedimientos para el control de flujo, la memoria temporal del receptor se podría llenar y eventualmente desbordarse mientras se procesan datos anteriores.

El control de flujo a veces es implementado en la capa de vínculo de datos, en otras ocasiones se le deja la responsabilidad a capas superiores.

Una de las técnicas de control de flujo más utilizadas en los protocolos full-duplex es el uso de las ventanas corredizas. Con este tipo de control de flujo, existe un límite en el número de tramas que el transmisor puede enviar antes de tener que esperar un reconocimiento. Este valor es conocido como el "tamaño de ventana". El valor de tamaño de ventana que mantienen las estaciones transmisora y receptora previene que la estación transmisora envíe más tramas de las que la estación receptora pueda aceptar en un momento dado. Con algunos protocolos, las estaciones intercambian tramas de control en la iniciación del enlace para establecer el valor del tamaño de la ventana. Cuando una estación envía el número de tramas especificadas por el tamaño de ventana sin recibir un reconocimiento, se detiene hasta recibir ese mensaje. La estación receptora puede utilizar los reconocimientos y marcos de control para controlar la velocidad a la cual se reciben las tramas. De esta manera el equipo receptor puede asegurar que no va a recibir más datos de los que puede manejar.

Existe otra aproximación que es ampliamente utilizada llamada "paquete de paro". La idea básica contempla que los nodos que detectan congestión envían un "paquete de paro" al nodo transmisor, cada vez que se entra en la región de congestiónamiento. El nodo transmisor reduce ó elimina este tipo de tráfico deteniendo su flujo de paquetes de datos. Esta aproximación es usada en DNA, BSC y otras redes. El uso de tramas "Receive Not Ready" en HDLC ó las tramas "WACK" en BISYNC sirven para el mismo propósito. Un ejemplo típico de este tipo de mecanismos lo encontramos en los protocolos XON/XOFF que se utilizan para controlar el flujo de datos de una comunicación asíncrona entre dos PC. Dos caracteres ASCII son designados para controlar el flujo de información: XON y XOFF. XOFF significa "parar transmisión" XON significa iniciar, continuar transmisión. Si el receptor, por alguna razón no puede aceptar más información, se envía un carácter XOFF a el transmisor. Cuando es capaz de aceptar de nueva cuenta datos, se envía un XON a el transmisor. Como se puede ver este mecanismo de control de flujo es muy sencillo de implementar.

2.3 Diseño de Capa de aplicación

Mientras que la capa de enlace de datos realiza una tarea no muy diferente a la del servicio de mensajería (cuya única función es la transmisión y recepción del mensaje, sin tomar en cuenta su contenido), la capa de aplicación es responsable del contenido de los mensajes específicos de la aplicación. Algunos de los protocolos de Internet que funcionan dentro de la capa de aplicación incluyen a SMTP (Simple Message Transfer Protocol), FTP (File Transfer Protocol), y TELNET. Los protocolos de aplicación de sistemas de control distribuido generalmente tienen usos diferentes. Por ejemplo, las aplicaciones de control de procesos probablemente usarán una red que realice las funciones de lectura de sensores, encender y/o apagar bombas u operar válvulas. En general, los desarrolladores de los sistemas de control distribuido personalizan los protocolos para aplicaciones específicas.

Podemos pensar en la capa de aplicación en términos de los servicios que provee. Los proveedores de servicios son llamados "servidores"; los dispositivos o procesos que usan los servicios de red son llamados "clientes". A diferencia de las redes de computadoras, en las cuales existen un mayor número de clientes que de servidores, en las redes de control usualmente existen más servidores que clientes. Los servicios pueden ser tan simples como leer un sensor binario. Un servicio más complejo podría permitir combinar los parámetros de control del equipo de automatización. Los clientes típicos son estaciones de trabajo del operador y controladores autónomos que monitorean o controlan múltiples dispositivos. Un nodo de red puede proveer y usar servicios de red.

Se pueden tener 2 aproximaciones en el diseño de la capa de aplicación. Una aproximación involucra definir mensajes concisos que un significado único (por ejemplo, "lee el sensor #2"). La otra aproximación usa memoria compartida, en la cual cada nodo de red tiene un bloque de memoria; en donde otros nodos pueden leer y escribir a esta memoria usando comandos de red.

CAPÍTULO 3

3. Desarrollo de la red

En este capítulo se tratará la construcción de la red de dispositivos de 8 bits ; en primer lugar se establecerá el sistema de desarrollo utilizado, para después explicar el desarrollo de las capas del modelo OSI involucradas en el protocolo de red: capa física y enlace de datos. Siendo el de esta última extenso por las características antes mencionadas (control de acceso al medio (CSMA/CD), detección de errores, servicio orientado a la conexión y control de flujo).

3.1 Selección del microcontrolador.

En la actualidad existen muchos tipos de microcontroladores, con diferentes fabricantes, características y capacidades. Por mencionar algunos de los fabricantes más conocidos tenemos a: Motorola, Microchip Technology, INTEL, National Semiconductor, Texas Instruments, AMD, Atmel, Harris, Dallas Semiconductor, Siemens, etc.

Uno de los microcontroladores de 8 bits más conocidos en el mundo de los microcontroladores es el HC11, el cual, contiene la mayoría de los recursos necesarios para el desarrollo de una amplia gama de aplicaciones. Además posee tecnología HCMOS, es de bajo consumo de energía, fácil adquisición en el mercado nacional a un costo razonable , disponibilidad de soporte técnico. Por lo cual, aunado con la disposición de un sistema de desarrollo, se seleccionó al MC68HC11 como microcontrolador de cada nodo de red

3.1.1.1 Generalidades del MC68HC11

El MCU 68HC11 es uno de los MCU's más poderosos en 8 bits, fabricado por Motorola con tecnología de HCMOS (High Density Complementary Metal-oxide Semiconductor), tiene como cerebro a un microprocesador 6800 mejorado, los códigos de operación son los mismos, por lo que un programa hecho para un 6800 corre sin ningún cambio en un

HC11. Sin embargo tiene más registros e instrucciones, puede operar a mayores frecuencias y su registro de código de condición (CCR) se ha completado a 8 bits, además cuenta con modo de operación : StandBY, en el cual el microprocesador consume muy poca energía.

La mayoría de las versiones cuentan con memorias : RAM, ROM y EEPROM, algunas versiones tienen EPROM ; los tamaños de las memorias dependen de la versión que se trate.

El HC11 tiene cinco puertos paralelos, dos puertos serie, uno síncrono y otro asíncrono.

Cuenta con un temporizador muy poderoso, que puede hacer operaciones de entrada y salida, y un acumulador de pulsos o tiempo.

Tiene un convertidor analógico-digital, con resolución de 8 bits que opera a buena velocidad y no requiere elemento de muestreo o retención.

El microcontrolador que se utilizó en el sistema de desarrolló fue el MC68HC11A1, a continuación se nombran sus principales características :

- Bus de 2 MHz, HCMOS, permite operaciones a frecuencias tan bajas como DC
- 512 bytes de EEPROM
- 256 bytes de RAM (Guardados durante standby). Relocalizables a cualquier límite de 4K
- Sistema de temporizador mejorado de 16 bits :
 - Preescalador programable de 4 etapas
 - Tres canales de captura de entrada IC
 - Canales de comparación de salida OC
- Acumulador de pulsos de 8 bits.
- Convertidor analógico-digital de 8 bits

- Circuito de interrupción de tiempo real
- Sistema de watchdog (Computer Operating Properly) COP.
- Set de instrucciones mejorado M6800/M6801
- División entera y fraccionaria de 16 x 16 bits.
- Instrucciones a manipulación a nivel de bits
- Modos de WAIT y STOP para reducir el consumo de energía.

En la Figura 3-1 se muestra el diagrama de bloques del MC68HC11A8. Este diagrama muestra los principales subsistemas y como se relacionan con los pines del MCU.

3.1.2 Unidad de control de procesamiento (CPU) del M68HC11

El CPU M68HC11 puede ejecutar todas las instrucciones del M6800 y 6801 y más de 90 nuevos códigos de operaciones. El MPU 68HC11 tiene 108 tipos diferentes de instrucción.

La arquitectura del 68HC11 es del tipo Von Neumann. Un sistema típico Von Neumann tiene 3 componentes principales: la unidad de control de procesamiento (CPU), memoria y dispositivos de E/S (Figura 3-2).

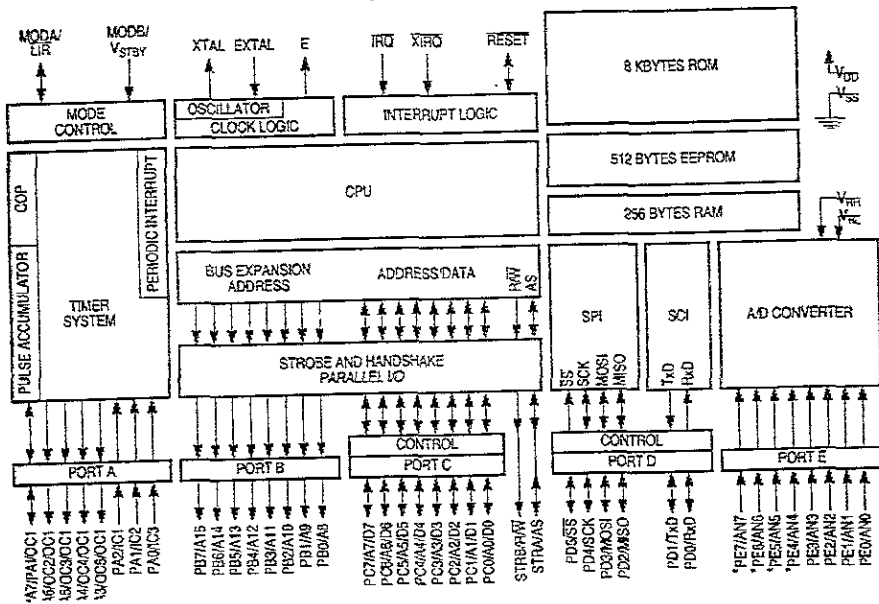
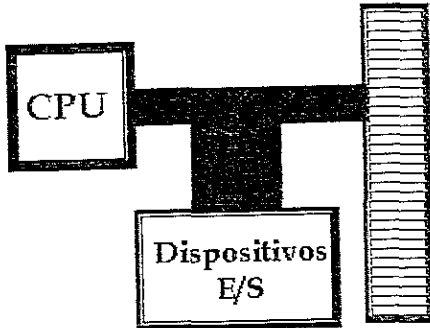


Figura 3-1 Diagrama de bloques del MC68HCA8

Memoria

En las máquinas Von Neumann, como el HC11, los datos y las instrucciones del CPU residen en memoria, hasta que son requeridos por el CPU.



El 68HC11 al igual que la mayoría de los MPU (Microprocessor Unit) de 8 bits tiene un bus de direcciones de 16 bits, o sea, que puede direccionar hasta $64K=65536$ localidades.

Figura 3-2 Máquina Von Neumann

La arquitectura del CPU M68HC11 trata a

los periféricos, dispositivos E/S, como localidades del mapa de memoria de 64Kbytes. De esta manera no existen instrucciones especiales para manejar los dispositivos de E/S.

3.1.3 Modelo de programación.

La Figura 3-3 muestra el modelo de programación del CPU M68HC11 .

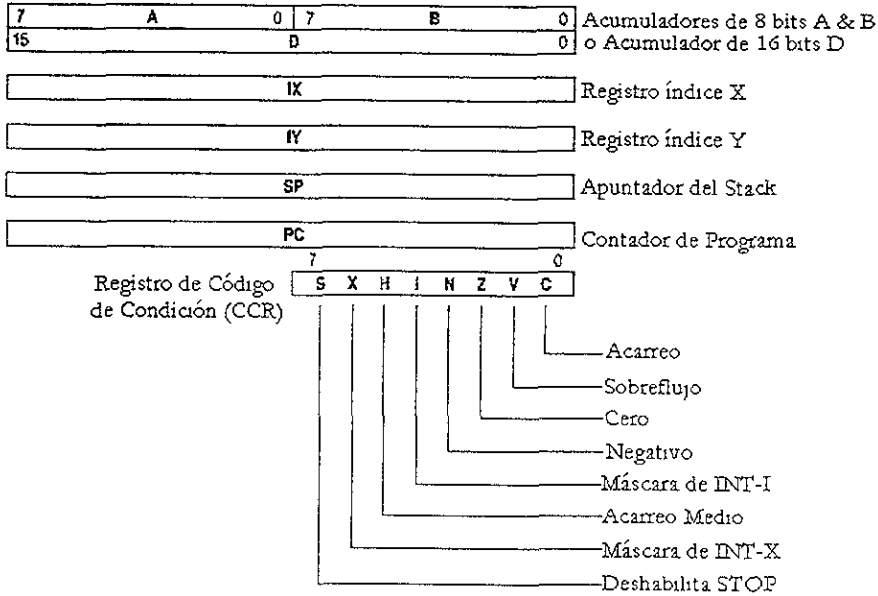


Figura 3-3 Modelo de Programación del HC11

3.2 Sistema de Desarrollo

El sistema de desarrollo del protocolo de red se basa en una PC y 2 estaciones controlados por un 68HC11. El diagrama simplificado de la red es el siguiente :

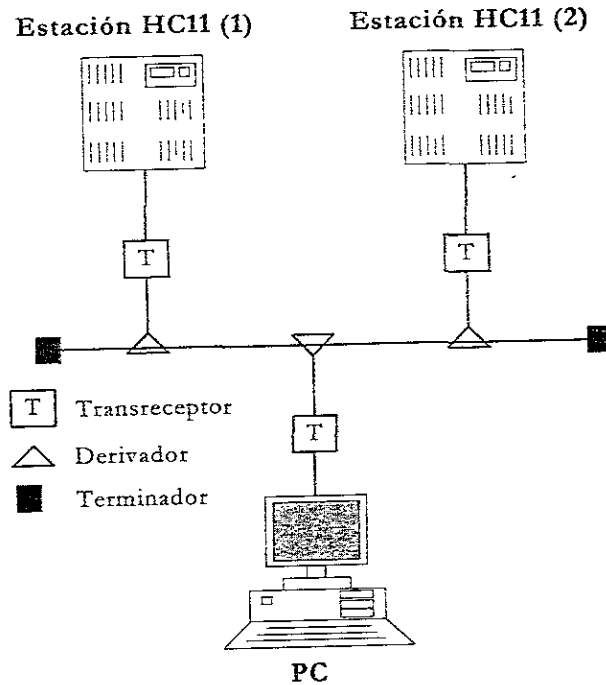


Figura 3-4 Red del sistema de desarrollo

Las estaciones HC11 se basaron en un sistema de desarrollo de propósito general. Las características del sistema de desarrollo son las siguientes :

- El HC11 funciona en modo *expandido*.
- El mapa de memoria de 64K esta dividido en bloques de 8K.
- Los dispositivos de memoria son :
 - 8K de memoria RAM
 - 8K de memoria EPROM
 - 8K de memoria EEPROM

- Los dispositivos de entrada y salida son :
 - Un teclado de 24 funciones
 - Un puerto serial adicional basado en el UART 8251
 - Un display de cristal líquido (LCD) de 2 columnas por 16 caracteres.

Como “sistema operativo” se maneja un programa “monitor”, que se almacena en la memoria EPROM, cuyas funciones principales son :

- Cargar programas, en formato S19, desde la PC hacia el sistema de desarrollo.
- Manejar los dispositivos de entrada y salida. Decodifica datos del teclado, despliega mensaje en pantalla, envía y recibe datos por medio del puerto serial adicional.
- Desplegar en representación ASCII, binario, hexadecimal el contenido de las localidades del mapa de memoria.
- Ejecutar programas.
- Manejar códigos de operación ilegales

En la Figura 3-5 se muestra la microcomputadora basada en HC11 que se utilizó como estación de red.

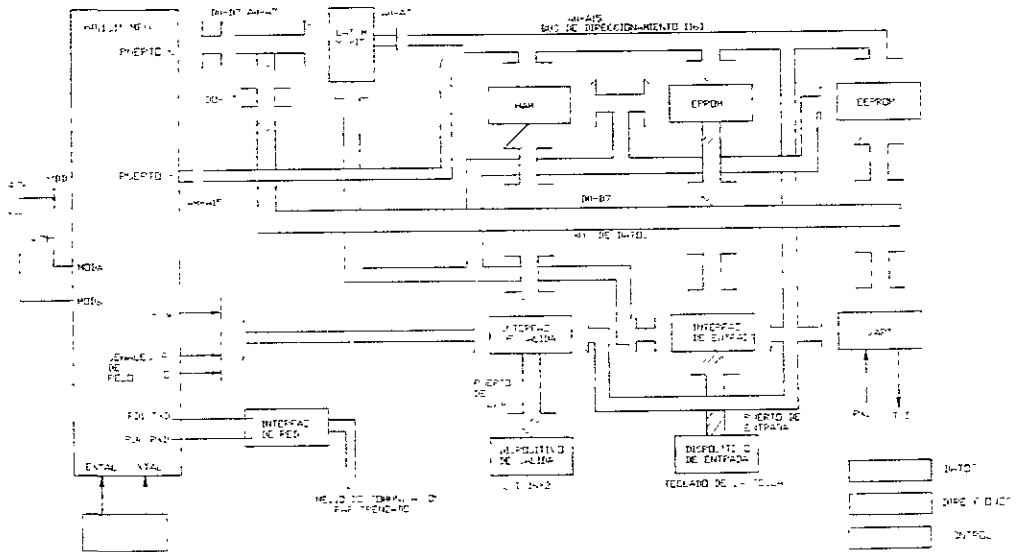


Figura 3-5 Sistema de desarrollo de cada nodo HC11

El mapa de memoria de cada estación es :

RAM interna (256 bytes)	← 0000
No utilizada	← 0100
Registros Internos (64 bytes)	← 1000
No utilizada	← 1040
RAM externa (8KB)	← 2000
EEPROM externa (8KB)	← 4000
TECLADO	← 6000
DISPLAY	← 8000
EPROM (Programa Monitor)	← A000
Puerto Serial (UART) adicional	← C000
No utilizada	← E000
	← FFFF

3.3 Capa física

Como se explico en el capítulo anterior la capa física de nuestro diseño utiliza una interfaz serial EIA-485 ; a continuación se explica la interconexión entre la estación y el transceptor, así como las principales características del sistema serial del HC11.

La conexión entre el MAX485 y el HC11 es sencilla (Figura 3-6), se utilizan los pines TXD y RXD del microcontrolador para la transmisión/recepción de datos ; un pin de salida (PA4) controla la transmisión del MAX485.

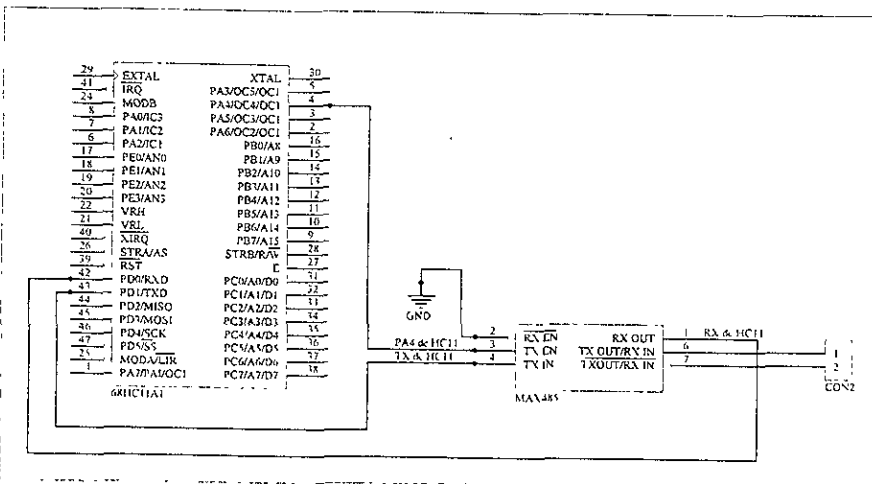


Figura 3-6 Interfaz capa física

3.3.1 Sistema de Comunicaciones Serial del HC11

Las principales características del sistema SCI (Serial Communication Interface) del HC11:

- Funcionamiento Full-duplex
- Sistema asíncrono del tipo UART
- Formato NRZ
- Doble buffer de transmisión y recepción

- Permite el manejo de formatos de palabra de 9 bits
- El receptor tiene la capacidad de ingresar temporalmente en un modo "standby" que le permite ignorar mensajes, enviados a un receptor diferente.
- El receptor tiene la lógica para despertarse automáticamente en el momento que se cumple una de las siguientes condiciones :a) noveno bit activado, b) un idle detectado. Esta característica de "wake-up" reduce el tiempo procesamiento del CPU cuando se trabajan con redes multipunto.

Como se puede ver el HC11 cuenta con propiedades que facilitan la realización de una red de comunicaciones. El poder dormir y despertar al HC11 selectivamente permite que los nodos que no están involucrados en la comunicación realicen otras tareas. Existen 2 métodos para despertar al HC11: el primero llamado "idle wake-up" despierta al microcontrolador cada vez que el estado del canal pasa de un estado ocupado a un estado ocioso. El estado ocioso es definido como la recepción de 10 bits en alto, cuando se maneja un formato de datos de 8 bits. El segundo método, "address mark wake-up", despierta al microcontrolador cada vez que se recibe un dato en el que el bit más significativo se encuentra activo.

En nuestro caso el uso de "address mark wake-up" es el método que se acopla a nuestras necesidades. Al utilizar este método es necesario cambiar el formato de 8 bits a 9 bits, para que el bit más significativo sea utilizado como marca de dirección y los otros 8 como datos. El primer carácter de cada mensaje contiene la dirección del receptor (noveno bit activado). Todos los receptores en el sistema evalúan este carácter para determinar si el mensaje va dirigido hacia uno de ellos. Tan pronto como un receptor determina que el mensaje no va dirigido a él, el receptor se pone a dormir.

3.3.2 Interfaz de la PC con la interfaz serial EIA-485

El sistema serial de la PC tiene algunas diferencias con respecto al sistema serial del HC11. Por ejemplo en nuestra red hacemos uso de un formato de palabras de 9 bits; sin embargo, la familia de UARTS de la PC (los más comunes: el 8251 y el 16550) no

soportan comunicaciones de 9 bits. Afortunadamente, se puede hacer un “truco” con el cual, la PC pueda soportar ese tipo de comunicaciones. La opción de paridad “mark”(HPSTICK) y “space”(LPSTICK) puede añadir el noveno bit a las comunicaciones de 8 bits del UART. A través de esta opción se puede transmitir datos con 9 bits y detectar los que tengan el noveno bit activado. La inicialización del UART debe ser : 8 bits de datos, 1 bit de inicio, 1 bit de paro, y paridad “space”.

En general, la mayoría de las características del sistema serial del HC11 se pueden implementar en la PC, a costa de un mayor tiempo de procesamiento. Por ejemplo, la implementación de “wake-up por noveno bit” requiere recibir todos los caracteres y detectar un error de paridad que indica la recepción de un dato con el noveno bit activado.

Para poder conectar la PC a la red fue necesario adaptar la interfaz EIA-485 al puerto serial de la PC, el circuito utilizado se muestra en la Figura 3-7.

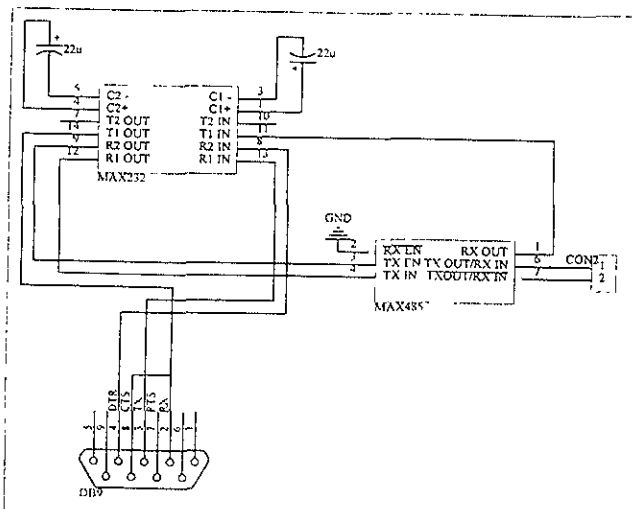


Figura 3-7 Interfaz PC-EIA485

Se utilizó el MAX232 para convertir los niveles lógicos de EIA-232 a TTL y viceversa. La señal de salida del estándar EIA-232 RTS *Request To Send* de la PC se utilizó como salida de propósito general.

3.4 Capa enlace de datos

3.4.1 Implementación de control de acceso al medio

En esta sección se explica el desarrollo del protocolo de acceso al medio CSMA/CD. Primero se explica la realización del detector de IDLE, que sirve como detector de portadora y después la realización del detector de colisiones. Por último se presenta el algoritmo de funcionamiento de CSMA/CD.

3.4.1.1 Realización del detector de IDLE

La característica de IDLE es muy importante en CSMA/CD debido a que le indica a la estación, el momento en el que el canal se encuentra libre.

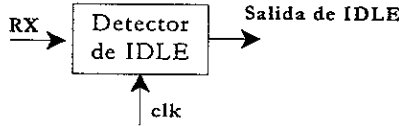
En nuestro diseño, la primer opción del detector de IDLE que se analizó utilizaba los recursos internos del HC11. Parecía que el sistema SCI (Serial Communications Interface) podía cubrir las necesidades de IDLE mediante del esquema de "IDLE wake-up". Se hicieron los programas de prueba correspondientes y se concluyó que esta opción no era posible debido a que "IDLE wake-up" y "address mark wake-up" no pueden trabajar al mismo tiempo.

La segunda opción que se analizó, también haciendo uso de los recursos internos del microcontrolador, consideraba conectar RXD a un pin de entrada y medir un pulso en alto de amplitud x tiempos de bit mediante un temporizador "input capture".

La desventaja de este método es la carga de procesamiento que implica utilizarlo, pues en el peor de los casos se puede producir una interrupción cada tiempo de bit. Por lo tanto se pensó en una tercera opción: la realización por hardware del detector de IDLE.

Se consideró el uso de una GAL16V8, debido a su facilidad de uso, tamaño pequeño y fácil disponibilidad.

El esquema entrada/ salida del detector de IDLE es:



El tabla de estados del circuito secuencial utilizado es el siguiente :

Entrada		Estado Presente						Estado Siguiente						Salida	
RX	H	F	E	D	C	B	A	Q _H	Q _F	Q _E	Q _D	Q _C	Q _B	Q _A	OUT
0	*	*	*	*	*	*	*	0	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0
1	0	0	0	0	0	0	1	0	0	0	0	0	1	0	0
1	0	0	0	0	0	1	0	0	0	0	0	0	1	1	0

1	1	1	1	1	1	1	0	1	1	1	1	1	1	1	0
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Las ecuaciones obtenidas, en formato CUPL, son las siguientes :

$$\begin{aligned}
 A_d &= (A \& RX) \# (H \& F \& E \& D \& C \& B \& A \& RX), \\
 B_d &= (B \& A \& RX) \# (B \& !A \& RX) \# (H \& F \& E \& D \& C \& B \& !A \& RX), \\
 C_d &= (C \& B \& A \& RX) \# (C \& !B \& RX) \# (C \& !A \& RX) \# (H \& F \& E \& D \& C \& B \& A \& RX), \\
 D_d &= (!D \& C \& B \& A \& RX) \# (D \& !C \& RX) \# (D \& !B \& RX) \# (D \& !A \& RX) \# (H \& F \& E \& D \& C \& B \& A \& RX), \\
 E_d &= (E \& D \& C \& B \& A \& RX) \# (E \& !D \& RX) \# (E \& !C \& RX) \# (E \& !B \& RX) \# (E \& !A \& RX) \# (H \& F \& E \& D \& C \& B \& A \& RX), \\
 F_d &= (!F \& E \& D \& C \& B \& A \& RX) \# (!F \& !E \& RX) \# (F \& !D \& RX) \# (F \& !C \& RX) \# (F \& !B \& RX) \# (!F \& !A \& RX) \# (H \& F \& E \& D \& C \& B \& A \& RX), \\
 H_d &= (!H \& F \& E \& D \& C \& B \& A \& RX) \# (!H \& !F \& RX) \# (H \& !E \& RX) \# (H \& !D \& RX) \# (H \& !C \& RX) \# (!H \& !B \& RX) \# (H \& !A \& RX) \# (!H \& !F \& E \& D \& C \& B \& A \& RX), \\
 ISAL &= H \& F \& E \& D \& C \& B \& A \& RX,
 \end{aligned}$$

En donde los operadores son :

OPERADOR	SIMBOLO
and	&
or	#
not	!

3.4.1.2 Realización del detector de Colisiones

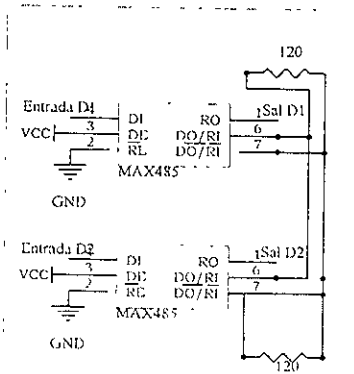
El objetivo de ese dispositivo al detectar colisiones es garantizar que únicamente un nodo de la red utilice el bus de comunicaciones.

Se investigaron los diferentes métodos que se utilizan para detectar colisiones, los más comunes son :

- Por hardware adicional : Observando la potencia o el ancho de pulso de la señal recibida y comparándola con la señal transmitida.
- Por software : Se envía información, al mismo tiempo que se escucha ; se compara lo transmitido con lo recibido.

La desventaja del primer método es su complejidad y costo de realización, por lo cual se optó por el segundo método.

Se realizaron algunas pruebas con el MAX485 para observar su comportamiento en situaciones de contienda, Figura 3-9 y Figura 3-10.



Entradas		Salidas	
D1	D2	D1	D2
0	0	0	0
0	1	0	0
1	0	1	1
1	1	1	1

Tabla 3.1 Resultados del circuito 2 drivers en contienda

Figura 3-9 Contienda entre 2 MAX485

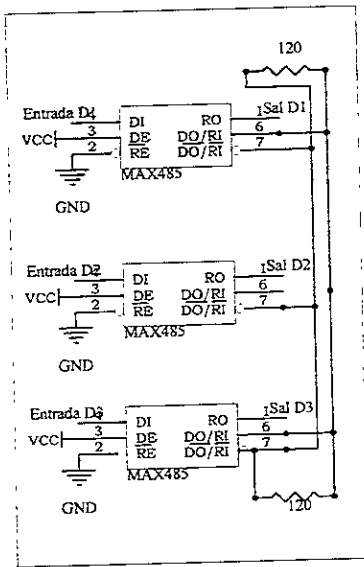


Figura 3-10 Contienda entre 3 MAX485

Entradas			Salidas		
D1	D2	D3	D1	D2	D3
0	0	0	0	0	0
0	0	1	0	0	0
0	1	0	0	0	0
0	1	1	1	1	1
1	0	0	0	0	0
1	0	1	1	1	1
1	1	0	1	1	0
1	1	1	1	1	1

Tabla 3.2 Resultados del circuito 3 drivers en contienda

Observando los resultados de las Tabla 3.1 y Tabla 3.2, se puede concluir lo siguiente :

- Cuando 2 drivers están activos al mismo tiempo y transmiten el mismo estado, el estado del bus será el mismo que manejen los drivers ; pero si transmiten estados diferentes, uno de los 2 drivers ganará el estado del bus.
- Cuando 3 drivers están encendidos al mismo tiempo, el estado del bus será el que maneje la mayoría.

Considerando lo anterior, el esquema propuesto para detectar colisiones es el siguiente :

Se compara el dato recibido con el transmitido, si son diferentes quiere decir que existió una colisión, si son iguales, puede empezar a transmitir.

Manejando 3 nodos pueden existir 3 casos de colisión :

Caso 1: Si 2 estaciones transmiten al mismo tiempo, una estación detecta colisión ; la otra gana el control del canal En la Figura 3-11 se muestra el ejemplo cuando 2 estaciones con direcciones 1 y 2, transmiten al mismo tiempo ; la estación 2 al enviar el bit 0, recibe el bit contrario, por lo cual detecta colisión.

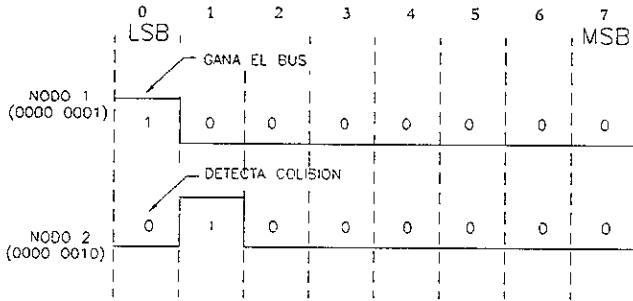


Figura 3-11 Nodo 1 y Nodo 2 en contienda

Caso 2: Si existen 3 estaciones con las direcciones 1,2,3, y transmiten al mismo tiempo, aplicando el esquema anterior, las estaciones 1 y 2 detectan colisión, la estación 3 gana el bus. Ver Figura 3-12

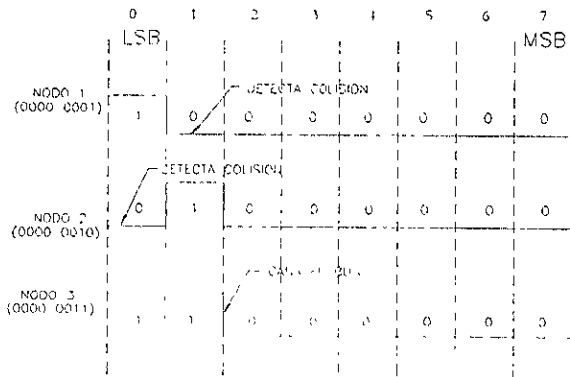


Figura 3-12 Nodo 1, Nodo 2 y Nodo 3 en contienda

CASO 3 : Si cambiamos las direcciones de las estaciones por 1,2,4 respectivamente, y transmiten al mismo tiempo, las 3 estaciones detectan colisión. Ver Figura 3-13.

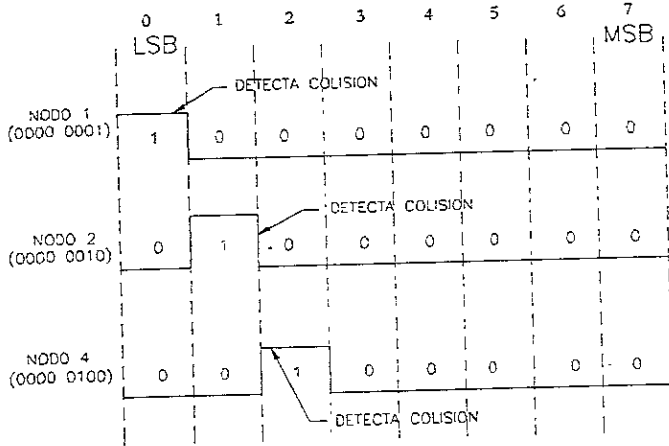


Figura 3-13 Nodo 1, Nodo 2, Nodo 4 en contienda

El esquema cumple con los requerimientos de detección de colisiones ; únicamente una estación maneja al canal.

El procedimiento general del CSMA/CD para empezar a transmitir se muestra en la Figura 3-14.

Uno de los puntos más interesantes en la implementación de CSMA/CD es el cálculo del tiempo de espera. Existen algunas implementaciones de CSMA/CD que generan tiempos aleatorios y otras que utilizan tiempos fijos. En nuestro diseño se utilizó esta última debido a su simplicidad de programación. Por lo tanto,

$$\text{Tiempo de espera} = \frac{\text{Tiempo de IDLE}}{(\# \text{ de estaciones} + 1)}$$

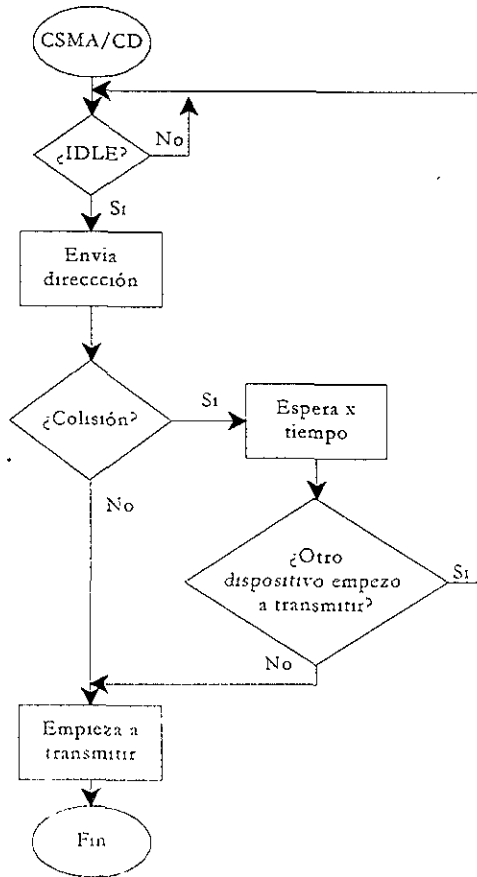


Figura 3-14 Algoritmo CSMA/CD

El temporizador que se utilizó para generar el tiempo de espera fue OC2, éste mismo se utilizó para generar el tiempo del temporizador de respuesta, que se explicará posteriormente.

3.4.2 Detección de errores

Como se mencionó en la parte de diseño, el método de detección de errores que se eligió es el CRC.

3.4.2.1 Métodos para calcular CRC

Existen 3 métodos para calcular el CRC de un bloque de datos, de los cuales se analizaron las características de cada uno de ellos :

3.4.2.1.1 Método por Hardware

La implantación por hardware es la realización más rápida que existe y se utiliza comúnmente en sistemas de comunicaciones que usan un microprocesador independiente que se encarga de las tareas de la capa física. Ethernet hace uso de este procedimiento para calcular la suma de comprobación de un CRC de 32 bits.

En el caso de nuestro diseño esta opción fue descartada pues se buscaba un método de detección de errores que utilizara la menor cantidad de recursos y su tiempo de procesamiento fuera corto. Además el colocar un circuito especial para calcular el CRC implicaba ocupar más espacio en el PCB (Print Circuit Board), una interfaz adicional con el HC11 y el costo mismo del dispositivo.

El procedimiento que se utiliza para calcular el CRC mediante hardware utiliza un circuito divisor formado por compuertas "or-exclusiva" y un registro de desplazamiento. El circuito se realiza de la siguiente manera :

1. El registro contendrá n bits, igual a la longitud de la FCS.
2. Hay n compuertas "or-exclusiva".
3. La presencia o ausencia de puerta corresponderá con la presencia o ausencia del término correspondiente en el polinomio divisor $P(X)$

En la Figura 3-15 se muestra la arquitectura de este circuito para un polinomio $P(X) = X^5 + X^4 + X^2 + 1$

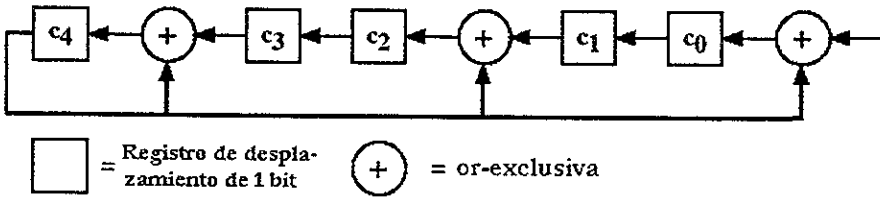


Figura 3-15 Implementación de un registro de desplazamiento

3.4.2.1.2 Método por Software

La manera más simple de implementar el algoritmo CRC en software es tomar como base la implementación en hardware y escribir el código apropiado, reemplazando el registro de corrimiento con una variable y las compuertas XOR con el operador `xor`. Este método, llamado *bitwise*, es la implementación más sencilla y consume muy poca memoria. Su desempeño, sin embargo es pobre. Cuando el tiempo de procesamiento del CRC es de importancia, diferentes algoritmos deben de ser usados. Estos algoritmos, basados en tablas almacenadas en memoria, procesan un byte a la vez, a diferencia de la implementación bitwise que procesa sólo uno. Estos algoritmos tienen una desventaja: tienen que mantener tablas en memoria, por lo cual necesitan más espacio en memoria que el primer algoritmo.

3.4.2.1.2.1 Algoritmo Bitwise

A continuación se presenta el diagrama de flujo del algoritmo *bitwise* del cálculo del CRC

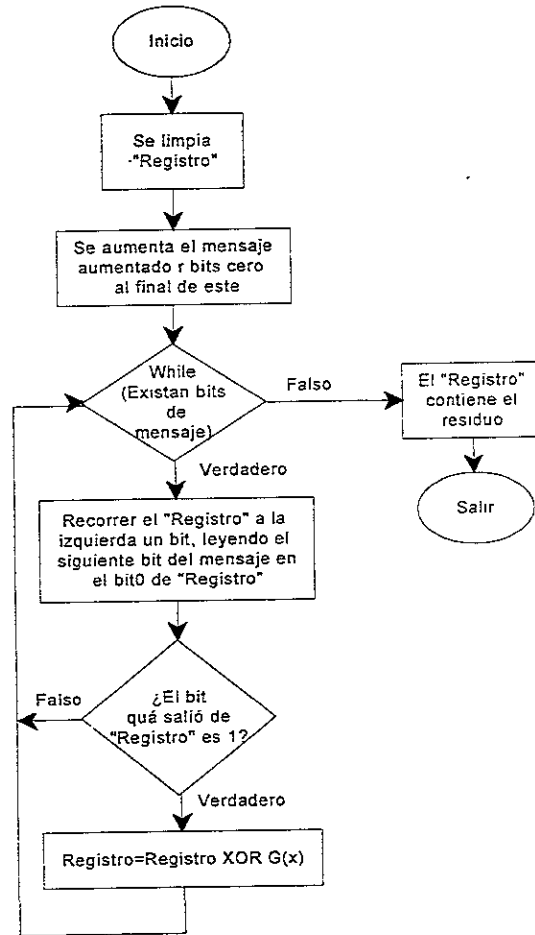


Figura 3-16 Algoritmo CRC-Bitwise

En donde,

r =Grado del polinomio generador.
 Registro tiene una amplitud de r bits y
 $G(x)$ es el polinomio generador.

3.4.2.1.2.2 Algoritmo manejado por tabla

El algoritmo manejado por tabla trabaja con unidades de 8 bits, además la mayoría de las operaciones son precomputadas en una tabla. El diagrama de flujo es :

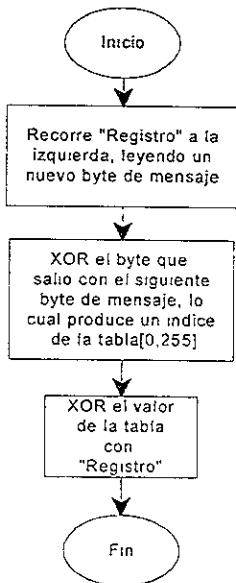


Figura 3-17 Algoritmo CRC manejado por tabla

Método	Por hardware	Algoritmo "bitwise"	Algoritmo manejado por tabla
Ventajas	Es el más rápido	Ocupa poca memoria	Tiene mayor desempeño en velocidad que el método "bitwise"
Desventajas	Mayor costo	Es lento comparado con el método de la tabla	Ocupa mayor espacio en memoria

Tabla 3-3 Ventajas y Desventajas de los diferentes métodos para calcular el CRC

Analizando cada método, el método elegido fue el algoritmo manejado por tabla debido a su alto desempeño y la cantidad de recursos que ocupa en el sistema. El CRC utilizado fue el CRC-16 $X^{16} + X^{15} + X^2 + 1$.

3.4.3 Estructura de los mensajes

Los caracteres en un paquete forman varios grupos, llamados campos. Como en un registro de una base de datos, cada campo contiene un tipo particular de información. Los campos típicamente encontrados en un paquete de la capa de enlace de datos son :

- Direcciones de red de los nodos emisor y receptor
- Bits de control definiendo el tipo de paquete y números de secuencia
- Datos generados por capas superiores
- Campo de suma de verificación para la detección de errores.

Para generar el formato de marco de nuestro sistema se revisaron algunas tramas de protocolos existentes :

SOH	COUNT		FLAGS	RESP	NUM	ADDR	CRC1	DATA	CRC2
ENQ	TYPE	SUB-TYPE	FLAGS	RCVR	SNDR	ADDR	CRC 3		

Figura 3-18 Marcos de control y datos de DDCMP

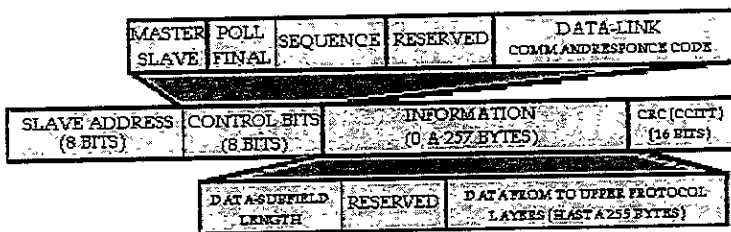


Figura 3-19 Marco utilizado en una red maestro-esclavo (Compañía Cimetrics)

Analizando estas tramas y tomando en cuenta las características de nuestro protocolo : orientado a la conexión, confirmaciones positivas y negativas, se diseñaron los campos de los mensajes.

3.4.3.1 Mensajes de control

El protocolo de nuestro diseño maneja dos tipos de tramas : Control y Datos ; a continuación se presenta cada una de ellas :

8 bits	8 bits	b7=1	b6b5b4	b2b1b0	16 bits
Dirección Destino	Dirección Origen	Tipo de mensaje	Tipo de Marco	Subtipo	CRC3

en donde :

Dirección Destino (1 byte) : Indica la dirección del nodo destino, el 9º bit es activado.

Dirección Origen (1 bit) : Identifica al nodo transmisor.

Tipo de mensaje(1 bit) : Indica si el marco es de control ó de datos.

bit 7 = 1 Indica marco de control.

Tipo de marco (3 bits) : Identifica los tipos de marcos de control

Subtipo(3bits) : Provee información adicional para algunos mensajes de control

CRC3(2 bytes) : La suma de comprobación del mensaje de control

Existen 5 tipos de mensajes de control :

3.4.3.1.1 Mensaje Solicitud de Conexión (S.C.)

El mensaje SC es usado para solicitar el establecimiento de una conexión. La forma del mensaje SC es .

Dirección Destino	Dirección Origen	Mensaje de control	Mensaje SC	Subtipo de SC	CRC3
-------------------	------------------	--------------------	------------	---------------	------

Dirección Destino (1 byte)

Dirección Origen (1 bytes)

Mensaje de control (1 bit) , bit 7=1

Mensaje SC (3 bits), bit6=0 , bit5=1 ;bit4=0

Subtipo de SC(3 bits) : No existe subtipo de solicitud de conexión.

bit2=0 ;bit1=0 ;bit0=0

CRC3(2 Bytes) . Suma de comprobación

3.4.3.1.2 Mensaje Cerrar Conexión (C.C)

El mensaje CC es usado para terminar con la conexión

Dirección Destino	Dirección Origen	Mensaje de control	Mensaje CC	Subtipo de CC	CRC3
-------------------	------------------	--------------------	------------	---------------	------

Dirección Destino (1 byte)

Dirección Origen (1 bytes)

Mensaje de control (1 bit) , bit 7=1

Mensaje CC (3 bits), bit6=0 ; bit5=1 ;bit4=1

Subtipo de CC(3 bits) : No existe subtipo de cerrar conexión. bit2=0 ;bit1=0 ;bit0=0

CRC3 (2 Bytes) : Suma de comprobación

3.4.3.1.3 Mensaje de confirmación positiva (ACK)

El mensaje es usado para confirmar la correcta recepción de un mensaje de datos, CC, SC. Además se utiliza para indicarle al nodo emisor que no existe conexión con ese dispositivo, esto por medio del subtipo.

Dirección Destino	Dirección Origen	Mensaje de control	Mensaje ACK	Subtipo de ACK	CRC3
-------------------	------------------	--------------------	-------------	----------------	------

Dirección Destino (1 byte)

Dirección Origen (1 bytes)

Mensaje de control (1 bit) , bit 7=1

Mensaje ACK (3 bits), bit6=0 ; bit5=0 ;bit4=0

Subtipo de ACK(3 bits) :

b2b1b0	Subtipos de ACK
0 0 0	ACK0
0 0 1	ACK1
0 1 0	Solicitud de Conexión Aceptada
0 1 1	ACK de Cerrar conexión
1 0 0	No existe conexión

CRC3 (2 Bytes) : Suma de comprobación

3.4.3.1.4 Mensaje de confirmación negativa (NAK)

El mensaje es usado para indicar una condición de error por parte del nodo receptor. Además se utiliza para rechazar una solicitud de conexión En el campo de subtipo se incluye la razón del error.

Dirección Destino	Dirección Origen	Mensaje de control	Mensaje NAK	Subtipo de NAK	CRC3
-------------------	------------------	--------------------	-------------	----------------	------

Dirección Destino (1 byte)

Dirección Origen (1 bytes)

Mensaje de control (1 bit) , bit 7=1
 Mensaje NAK (3 bits), bit6=0 ; bit5=1 ;bit4=0
 Subtipo de NAK(3 bits) :

b2b1b0	Subtipos de ACK
0 0 0	Error en el CRC de datos
0 0 1	Buffer no disponible
0 1 0	Solicitud de Conexión rechazada
0 1 1	Respuesta al REP

CRC3 (2 Bytes) : Suma de comprobación

3.4.3.1.5 Mensaje Solicitud de estado de un mensaje de datos (REP)

El mensaje REP se utiliza para preguntar el estado de un mensaje de datos. Usualmente se envía cuando el nodo emisor sufre un TIME-OUT de respuesta y quiere saber si el mensaje llegó correctamente.

Dirección Destino	Dirección Origen	Mensaje de control	Mensaje REP	Subtipo de REP	CRC3
-------------------	------------------	--------------------	-------------	----------------	------

Dirección Destino (1 byte)
 Dirección Origen (1 bytes)
 Mensaje de control (1 bit) , bit 7=1
 Mensaje REP (3 bits), bit6=1 , bit5=0 ;bit4=0
 Subtipo de REP(3 bits) · Indica el número de secuencia

Código	Subtipos de REP
0 0 0	REP0
0 0 1	REP1

CRC3 (2 Bytes) · Suma de comprobación

3.4.3.2 Mensaje de datos

8 bits	8 bits	b7=0	b1	b0	8 bits	16 bits		16 bits
Dirección Destino	Dirección Origen	Tipo de mensaje	FRAG	Número de Secuencia	Longitud	CRC1	Datos	CRC2

Dirección Destino (1 byte) : Indica la dirección del nodo destino, el 9º bit es activado.

Dirección Origen (1 bit) : Identifica al nodo transmisor.

Tipo de Mensaje (1 bit) : Indica si el marco es de control o de datos

bit 7=0 Indica marco de datos

FRAG (1bit) : Información acerca de la fragmentación del mensaje de datos.

bit 4=0 EOT (Último marco del bloque de datos)

bit 4=1 STX (Trama de un bloque de datos)

Número de Secuencia(1 bit) : Campo que contiene el número de secuencia para el mensaje de datos.

Longitud(1 byte) : Indica el número de datos que contiene la trama.

CRC1(2 bytes) : Suma de comprobación del encabezado del mensaje. Es calculado desde

Dirección Destino hasta el campo de Longitud

DATOS (longitud) : En este campo se incorporan los datos recibidos de capas superiores.

CRC2(2 bytes) : Suma de comprobación del campo de datos. Únicamente son tomados en cuenta los bytes del campo de datos.

3.4.4 Operación del Protocolo

Las transferencias de información en el protocolo tienen tres fases distintas :

1. Conexión

2. Intercambio de marcos de datos

3. Desconexión

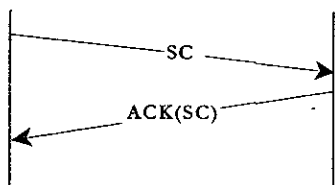
A continuación se verá el intercambio de mensajes que involucra cada una de ellas.

3.4.4.1 Establecimiento de una conexión.

En analogía con TCP de TCP/IP, nuestro protocolo es orientado a la conexión. Esto quiere decir que antes de que se envíe información entre 2 puntos debe establecerse una conexión.

En TCP un nodo de red puede manejar múltiples conexiones a través del uso de puertos. En nuestro caso , por facilidad de realización, a cada nodo de red se le permite tener una sola conexión. Esto no significa que si en un momento dado 1 y 2 se encuentran conectados y 3 y 4 se quieran conectar, tengan que esperar que 1 y 2 se desconecten. En

un tiempo dado, pueden ocurrir múltiples conexiones siempre y cuando estén involucrados diferentes pares de nodos.



El intercambio de mensajes para establecer una conexión se muestra en la Figura 3-20

Los mensajes perdidos o dañados son detectados por el temporizador de respuesta, enviando de nueva cuenta el mensaje SC. Figura 3-21 y Figura 3-22.

Figura 3-20 Solicitud de conexión

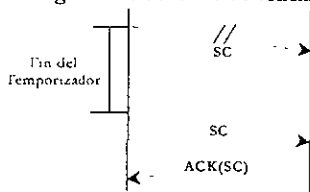


Figura 3-21 Solicitud de conexión con trama perdida

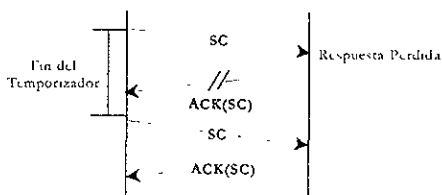


Figura 3-22 Solicitud de conexión con respuesta perdida

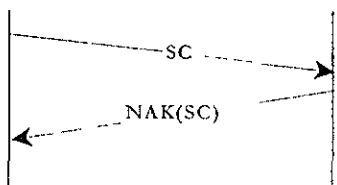


Figura 3-23 Solicitud de conexión rechazada

Existe conexión

Otro mensaje involucrado con SC es NAK (Solicitud de conexión rechazada) el cual es enviado cuando se bloquean las entradas de conexiones o existe una conexión.

Figura 3-23

3.4.4.2 Transferencia de Mensajes de datos

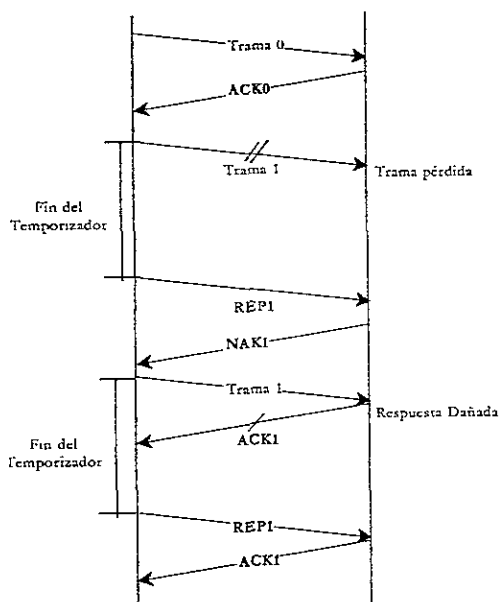


Figura 3-24 Transferencia de datos

En la sección de control de errores se habló del manejo de los números de secuencia para manejar un nivel de confiabilidad 1. En nuestro protocolo el manejo de los números de secuencia se realiza de la misma manera, pero el manejo de errores es diferente. Cuando un nodo emisor sufre la terminación de su temporizador de respuesta, no es claro si el mensaje fue recibido incorrectamente ó si la confirmación sufrió un error. En este caso, en vez de reenviar un mensaje posiblemente largo, se envía un mensaje de control REP junto con el número de

mensaje previamente enviado. Si el mensaje con ese número fue recibido correctamente, la respuesta al REP es un ACK, de otra manera es una NAK. La recepción de un NAK provoca la retransmisión; el temporizador de respuesta es reactivado después de la transmisión del REP. Si el temporizador expira de nueva cuenta, el proceso es repetido. Después de 3 time-out's, el software transmisor notifica al usuario de la capa de enlace de datos.

3.4.4.3 Desconexión

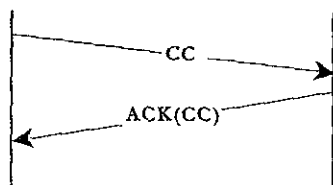


Figura 3-25 Cerrar conexión

El intercambio de mensajes para liberar una conexión se muestra en la Figura 3-25.

Los mensajes perdidos o dañados son detectados por el temporizador de respuesta, enviando de nuevo el mensaje CC. Figura 3-26 y Figura 3-27

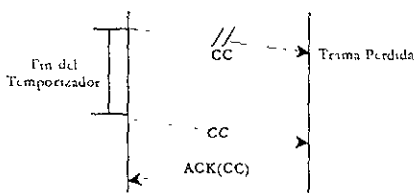


Figura 3-26 Cerrar conexión con trama perdida

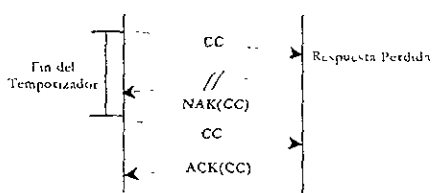


Figura 3-27 Cerrar conexión con respuesta perdida

3.4.4.4 Temporizador de respuesta

Cada vez que se termina de transmitir el último carácter de los siguientes mensajes se activa el temporizador:

- a) Mensaje SC
- b) Mensaje de datos
- c) Mensaje RHP
- d) Mensaje CC

Cada vez que llega uno de los siguientes mensajes se desactiva el temporizador:

- a) Mensaje ACK
- b) Mensaje NAK

3.4.4.5 Mensajes válidos

Únicamente los mensajes válidos son procesados por el protocolo. Un mensaje no válido tiene las siguientes características :

- Error en el CRC1 de un mensaje de datos.
- Error en el CRC3 de un mensaje control.
- Los campos TIPO DE MARCO y SUBTIPO tienen un valor no definido.

3.4.4.6 Contador de Time-out's

Antes de avisarle al usuario que ocurrió un time-out, se reenvía el mensaje tres veces más, si el contador de time-out llega a esta valor entonces se le avisa al usuario.

3.4.5 Control de Flujo

Por facilidad de realización, el control de flujo utilizado en nuestro protocolo trabaja de la siguiente manera :

Cuando el receptor no tiene espacio en la memoria temporal (buffer) para almacenar el mensaje, envía una confirmación negativa con un subtipo de no hay buffer disponible.

El nodo emisor le avisa al usuario de la capa de enlace de datos de esta condición, y es el usuario quien decide reenviar el mensaje o cancelar la transmisión.

3.4.6 Interfaz de Usuario

La interfaz entre el protocolo de enlace de datos y el usuario de ésta misma, consiste de un número de comandos y respuestas. Los comandos (subrutinas) que incluye el protocolo son los siguientes :

Comando	Descripción
STARTCON	Inicio de conexión
CLOSECON	Liberar conexión
TRANSMITE_DATOS	Envía Mensaje de datos
STATUSMSGDAT	Revisa el estado del mensaje de datos transmitido
SENDAGAIN_MSG	Vuelve a enviar el mensaje de datos
RESET_PROTOCOL	Restaura el protocolo
STATUSMSGRCV	Revisa si ha llegado algún mensaje
FREEBUFFRCV	Libera buffer de recepción
INICIAVARS	Inicia los valores de algunas variables utilizadas por el protocolo.

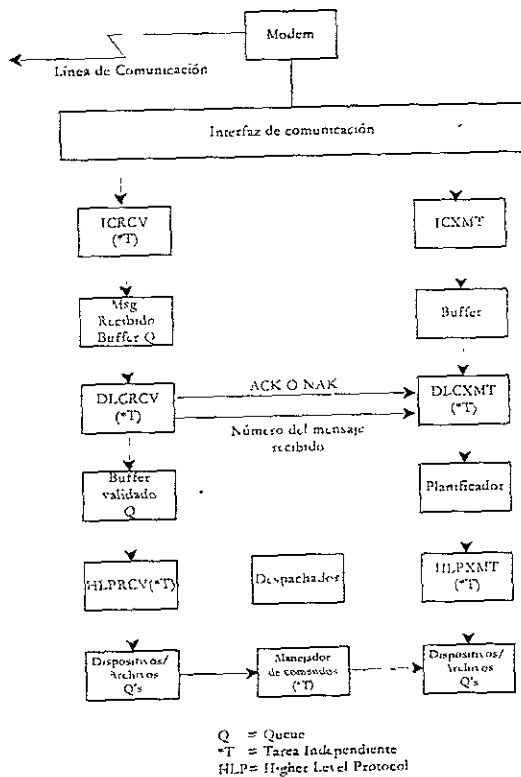
Tabla 3.4 Comandos del protocolo enlace de datos

3.5 Programación del Protocolo

Uno de los puntos más importantes al programar un protocolo es la manera de organizar el programa. Existe mucha información acerca de los protocolos de comunicaciones de datos, estándares de interfaz, redes de área local, redes conmutadas pero existe muy poca información acerca de las metodologías e implementación de software de comunicaciones.

Dos autores que han investigado dichas metodologías: [LANE] y [KRATZER] proponen un manejo multitareas del software de comunicaciones. A continuación se explica, de forma breve, la del primero.

En esta aproximación se sugiere dividir el protocolo en tareas independientes (Figura 3-28). El manejo de estas tareas involucra muchas veces, técnicas usadas en la implementación de sistemas operativos. Uno de los métodos más populares para implementar sistemas operativos llamado modelo de bloque de control de tareas ("Task Control Block") se suele utilizar en la programación de software de comunicaciones. En esta técnica, se asocia cada tarea con un contexto (por ejemplo, su contador de programa, el contenido de sus registros), un número de identificación, un "status" y una prioridad si es aplicable. Estas variables son almacenadas en una estructura llamada bloque de control de tareas (TCB). Las tareas son unidas entre sí por medio de listas ligadas. La Figura 3-29 muestra un bloque de control de tareas típico.



ICRCV :Controlador para la interfaz receptora.

DLCRCV :Realiza las funciones de la capa de enlace de datos en la parte de recepción

HLPXMT :Realiza las funciones de capas superiores del protocolo en la parte de recepción

Manejador de comandos : Controla el software de comunicaciones por medio de comandos.

HLPXMT :Realiza las funciones de capas superiores del protocolo en la parte de transmisión

DLCRCV :Realiza las funciones de la capa de enlace de datos en la parte de transmisión

ICRCV :Controlador para la interfaz transmisora

Despachador : Controla la ejecución de cada tarea.

Figura 3-28 Software de comunicaciones con un esquema multitareas y un manejo de colas

Contador de programa
Estado de la tarea
ID de la tarea
Contenido del registro 0
.
.
.
Contenido del registro n
Otro contexto
Puntero al siguiente TCB

Figura 3-29 Bloque de control de tareas

Tres funciones básicas de un sistema multitareas son[LAPLANTE]: planificación de tareas, despachador de tareas y comunicación entre tareas. El planificador determina cual es la siguiente tarea que correrá en un sistema multitareas, mientras que el despachador realiza la administración necesaria para iniciar la tarea.

El planificador de tareas que se suele utilizar en la programación de software comunicaciones es el de "round-robin", debido a su simplicidad. Cada tarea revisa las banderas apropiadas para ver si tiene que hacer algún trabajo. Si es así, lo realiza y regresa el control al despachador. Si no, regresa inmediatamente dejando el control del CPU a la siguiente tarea. En este sistema se puede presentar el problema que una tarea muy grande tome el control del CPU retardando la ejecución de las siguientes tareas. Para resolver este problema se utilizan sistemas con división de tiempo, en donde a cada tarea se le asigna una cantidad de tiempo fijo, en la cual puede ejecutarse. Si una tarea no termina de ejecutarse en su división de tiempo, entonces el despachador debe de guardar el contexto de la tarea, la cual, es puesta al final de la lista de tareas. El contexto de la siguiente tarea en la lista es restaurado y continua su ejecución.

En los sistemas operativos las interrupciones suelen activar tareas. En algunos, por medio de banderas se comunican con una tarea. Esta tarea espera su oportunidad para poder realizar la acción apropiada. La eficiencia de estos sistemas puede incrementarse permitiendo que algunas tareas puedan adelantarse en la lista de ejecución ("preemptive systems"). En el momento que se presenta una interrupción se guarda el contexto de la tarea que se esté ejecutando, se planifica la tarea respectiva, se ejecuta y se reanuda con la ejecución de la tarea anterior.

La metodología multitareas tiene como principal ventaja su división en módulos independientes, que permite cambiar o modificar algunos módulos sin la necesidad de

cambiar los demás. Por otro lado son más fáciles de comprender por la gente de mantenimiento del software. Desgraciadamente el manejo de un sistema multitareas involucra desventajas claves en nuestro diseño: La principal es el tiempo de procesamiento involucrado al adelantar, restaurar y comunicar tareas.

Por lo tanto se pensó en otra estructura, en la cual el peso del software recae sobre las Rutinas de Servicio de Interrupción (ISR Interrupt Service Routine) de comunicación y recepción. Las interrupciones provocan una respuesta inmediata y directa sin la necesidad de utilizar varias tareas. Sin embargo, los problemas más comunes al manejar ISR son los siguientes :

- Mantenimiento
- Depuración
- Modularidad

Para resolver estos problemas convertimos a la ISR en una máquina de estados manejada por interrupciones. De esta manera la ISR es dividida en estados; cada vez que se produce una interrupción, la ISR toma el valor de la variable "estado", con este valor (que sirve de índice) entra en una tabla de estados y salta a la dirección encontrada en la tabla.

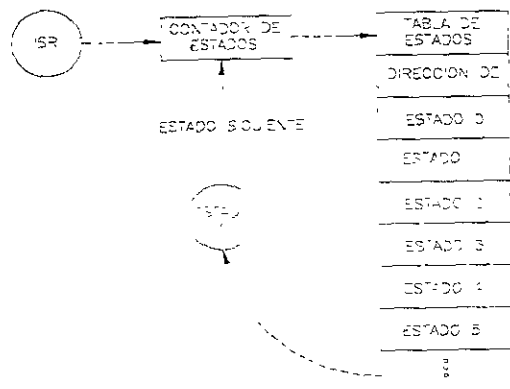


Figura 3-30 Estructura de la ISR del Sistema de Comunicaciones

Otra de las ventajas del uso de las máquinas de estados manejados por interrupciones es su facilidad de implementación. En el protocolo se manejaron dos máquinas de estados : la de transmisión y la recepción, mostradas en las Figura 3-32, Figura 3-33, Figura 3-34.

La organización final del software del protocolo, tomando en cuenta la máquina de estados manejados por interrupciones y los comandos hacia el usuario, es la siguiente .

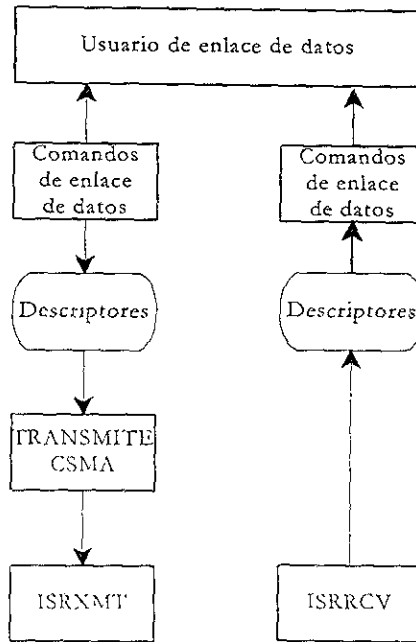


Figura 3-31 Organización final del software

Los descriptores son estructuras que permiten intercambiar información entre diferentes bloques del protocolo, principalmente entre los comandos y las ISR de comunicación.

La rutina "Transmite_CSMA" se encarga de las tareas del elemento control de acceso al medio, de la capa enlace de datos.

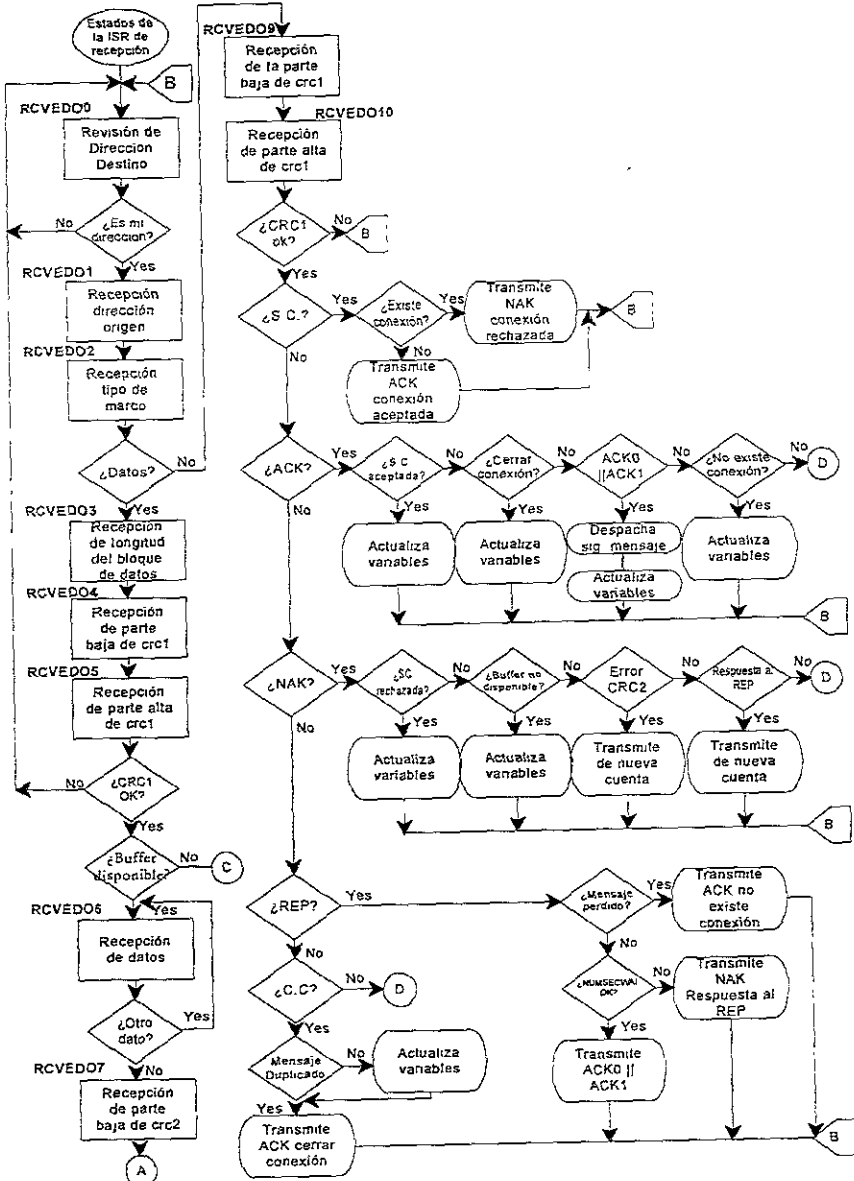


Figura 3-32 ISR de Recepción parte 1

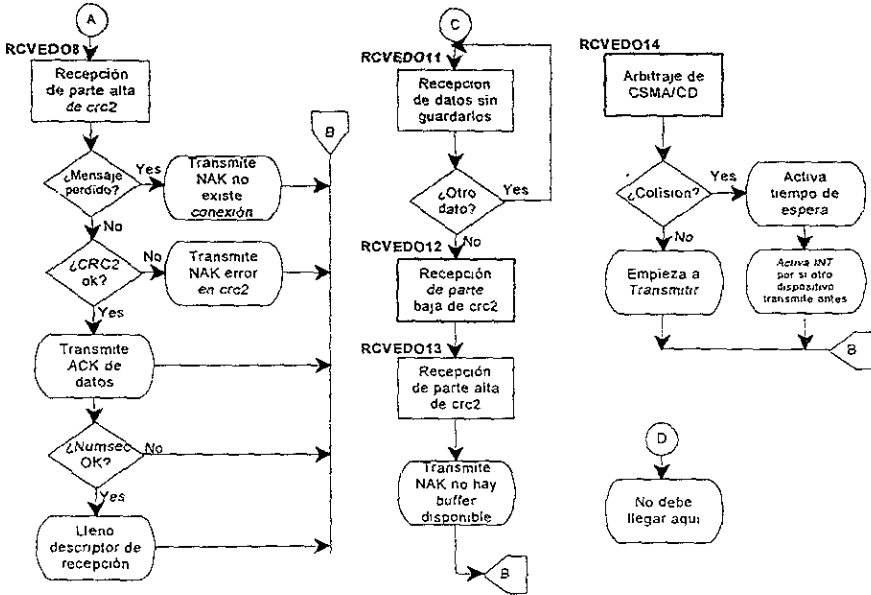


Figura 3-33 ISR de Recepción parte 2

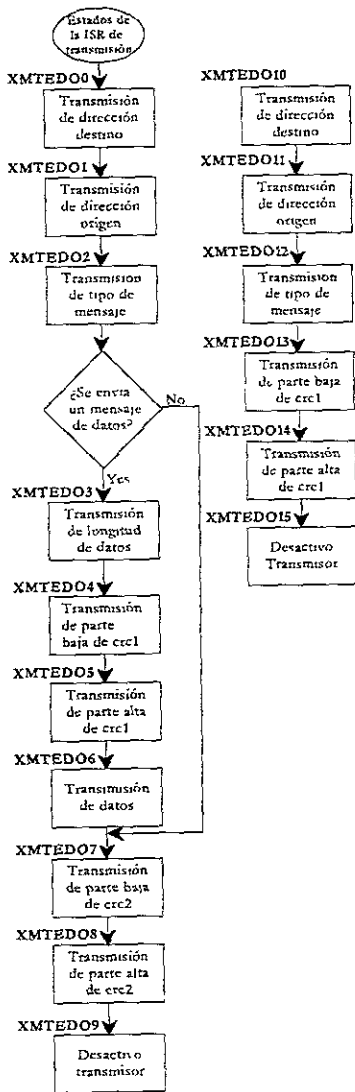


Figura 3-34 ISR de Transmisión

PRUEBAS Y RESULTADOS

4. Pruebas y Resultados

En este capítulo se describirán las diferentes pruebas realizadas para revisar el funcionamiento del protocolo de comunicaciones.

Como se mencionó en el capítulo 3, la red de comunicaciones que se utilizó para probar el protocolo incluye dos estaciones HC11 (Figura 4-1y Figura 4-2) y una PC, la cual tuvo 2 funciones principales en el proceso de pruebas:

- 1) Depuración del algoritmo CSMA/CD de los nodos HC11
- 2) Como monitor de red para revisar el intercambio de información.

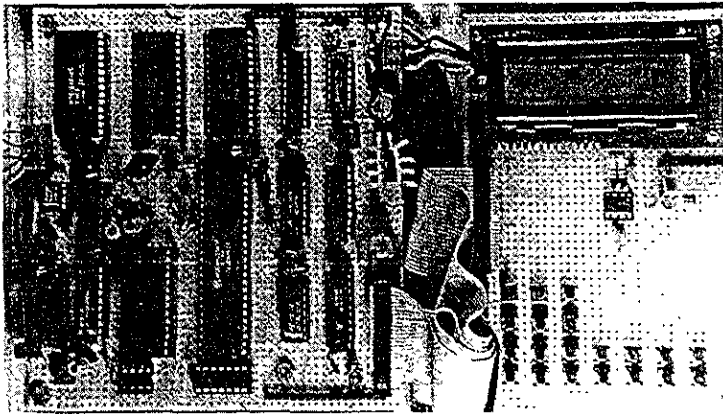


Figura 4-1 Estación "1" HC11.

El proceso de pruebas estuvo dividido en 4 etapas :

- 1) Pruebas del protocolo de control de acceso al medio (MAC)
- 2) Pruebas de establecimiento de conexión

- 3) Pruebas de transmisión de mensajes
- 4) Pruebas de liberación de conexión

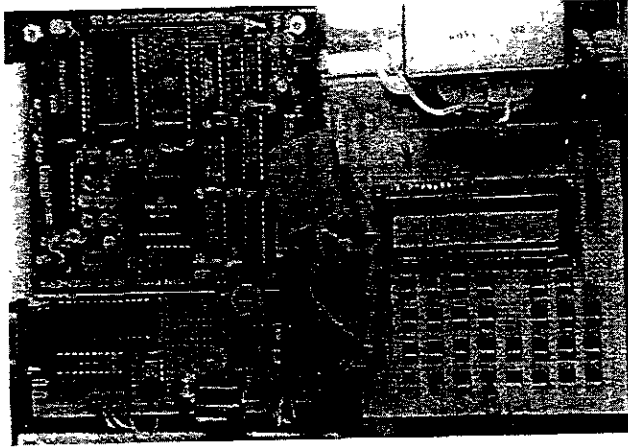


Figura 4-2 Estación "2" HC11

4.1 Pruebas de CSMA/CD

Para poder revisar el funcionamiento del protocolo de acceso al medio (CSMA/CD), se desarrolló un programa de aplicación en lenguaje C en la PC. El procedimiento para probar MAC es el siguiente : la computadora en un tiempo dado ocupa el medio de comunicación, y mientras ocurre esta transmisión, los 2 nodos (con direcciones 1 y 2) solicitan transmitir, cada uno detecta que el canal está ocupado y activa su interrupción por IDLE. Al momento que la PC desocupa el medio, los 2 nodos intentan transmitir al mismo tiempo, uno de ellos detecta colisión y el otro empieza a transmitir su mensaje. Si las direcciones de las estaciones se cambian por 1 y 3, los 2 nodos detectarán colisión, un nodo esperará n tiempo y el otro $3n$, de tal forma que el nodo que tenga menor tiempo de espera ganará el canal. Este tiempo de espera (n) tiene un valor fijo, definido por la siguiente fórmula :

$$\text{Tiempo de espera}(n) = \frac{\text{Tiempo de IDLE}}{(\# \text{de estaciones} + 1)}$$

En nuestro desarrollo se utilizó una velocidad de 2400 bps, y 9 tiempos de bit como IDLE, suponiendo que se utilizan 32 estaciones, la separación entre cada tiempo de espera es :

$$\text{Tiempo de espera} = \frac{9}{2400 * (32 + 1)} = 113.6363 \mu\text{s} \approx 227 \text{ ciclos}$$

Los tiempos de espera definidos para cada una de las estaciones son :

$$\text{Nodo1} = 113.6363 \mu\text{s} = 227 \text{ ciclos}$$

$$\text{Nodo2} = 227.2726 \mu\text{s} = 554 \text{ ciclos}$$

Sin embargo, existe un valor de tiempo de espera mínimo controlado por el tiempo de procesamiento del protocolo, las cantidades involucradas en este cálculo se mencionan en la Tabla 4.1. (No son tomados en cuenta los tiempos de propagación y la latencia).

Periodos de tiempo considerados :	# de ciclos
Tiempo desde la activación del TIMER hasta la salida del estado 14 de la ISR de recepción (incluyendo RTI).	25
Entrada a la ISR del TIMER de Espera (Nodo transmisor).	aproximadamente 20
Tiempo desde que se entra a la ISR y se envía el primer dato. (Nodo transmisor)	46
Entrada en la ISR de detección de flanco de bajada (RX)	aproximadamente 20
Tiempo desde que se entra a la ISR y se desactiva el temporizador de respuesta	18
Tiempo mínimo de espera (total)	129 ciclos = 64.5 μs

Tabla 4.1 Cantidades involucradas en el cálculo del tiempo mínimo de espera

4.2 Pruebas de establecimiento de conexión

Para revisar el funcionamiento de la fase de solicitud de conexión, se simularon algunas condiciones anómalas que debe solucionar el protocolo, tales como :

- 1) Inicio de sesión simultáneo entre estaciones
- 2) Solicitud de conexión rechazada, debido a que existe una conexión

3) Solicitud de conexión rechazada, debido a que se está esperando respuesta de una solicitud de conexión.

4) Solicitud de conexión duplicado.

Para revisar el intercambio de mensajes entre las estaciones, se hizo un programa "Monitor de Red", en Lenguaje C en la PC, de tal manera que en la pantalla de la computadora aparecieran los datos transmitidos por cada una de las estaciones. En la Figura 4-3 se muestra la pantalla del programa "Monitor de Red", después de que el nodo2 envía un SC al nodo1.

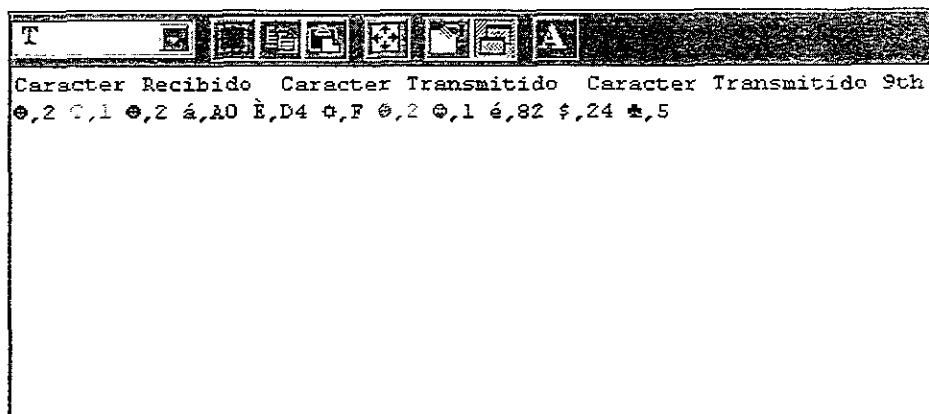


Figura 4-3 Pantalla del programa Monitor de Red, después de que el nodo2 le envía un SC al nodo1

4.3 Pruebas de Transmisión de Mensajes

La fase de transmisión de mensajes es la etapa más larga del intercambio de información entre estaciones. Esta etapa lleva a cabo varias tareas y para revisar su funcionamiento se simuló las siguientes condiciones :

- 1) Transmisión de un mensaje de datos duplicado.
- 2) Mensajes dañados y perdidos, que involucran el envío de un mensaje REP .
- 3) Transmisión de un mensaje de datos, a un nodo con el cual no se tiene conexión.

- 4) Transmisión de un mensaje de datos con el buffer del nodo receptor lleno.
- 5) Transmisión de un mensaje de datos con el CRC de datos erróneo,
- 6) Transmisión de un mensaje mayor a 255 bytes.

4.4 Pruebas de Liberación de Conexión

Para liberar una conexión de manera normal se debe enviar el comando CC al nodo receptor.

De las 3 etapas que conforman la operación del protocolo, ésta es la más sencilla. La única condición anormal asociada con el comando CC es :

- 1) Mensaje Cerrar Conexión duplicado.

4.5 Cálculo del temporizador de respuesta

El temporizador de respuesta se activa después de enviar el último carácter de un mensaje: SC, Datos, REP ó CC. Los mensajes ACK y NAK detienen a este temporizador

El cálculo del temporizador de respuesta es muy importante debido a que, si su valor de conteo inicial es muy pequeño, puede provocar retransmisiones innecesarias, cargando a la red de paquetes inútiles. Si se escoge un valor demasiado largo, el desempeño sufrirá debido al retardo de retransmisión.

El cálculo del temporizador de respuesta involucra varios parámetros :

1. El retardo de propagación del medio de transmisión
2. El tiempo de procesamiento del protocolo
3. El tiempo de transmisión del mensaje de respuesta ACK/NAK
4. La latencia antes de poder servir a las interrupciones

A grandes distancias, el retardo de propagación debe ser tomado en cuenta. Por ejemplo, a una distancia de 1200 metros el retardo de propagación sería de 6µs.

El tiempo de procesamiento del protocolo es difícil de calcular debido a que varía según el mensaje que se envíe. Las cantidades involucradas en su cálculo son las siguientes :

Procesamiento de :	Tiempo involucrado
Recepción del último carácter por parte del nodo receptor	El procesamiento varía para cada mensaje, el tiempo más largo es 0.1155ms
Transmisión del Primer carácter al quinto carácter del mensaje ACK/NAK por parte del nodo receptor	$0.043ms + 0.0685ms + 0.0615ms + 0.0325ms + 0.0395ms = 0.245ms$
Recepción del último carácter del mensaje de control ACK/NAK y detención del temporizador de respuesta por parte del nodo transmisor	El procesamiento varía para cada mensaje, el tiempo más largo es 0.1155ms
Total	0.476ms

El tiempo de transmisión del mensaje de confirmación positiva o negativa es :

$$Tiempo\ de\ Transmision = \frac{11 * 5}{VELOCIDAD}$$

La latencia involucrada al servir las interrupciones en el caso de nuestro software es despreciable, debido a que no existe una aplicación sobre el protocolo de red. La velocidad que se manejó al realizar las pruebas fue de 2400 bps, y el retardo de propagación fue despreciable, por lo que, el valor de tiempo de respuesta que se utilizó fue :

$$Tiempo\ de\ respuesta = \frac{11 * 5}{velocidad} + 0.476ms = 22.91ms + 0.476ms = 23.386ms$$

4.6 Cantidad de memoria utilizada

A continuación se presenta la cantidad de memoria utilizada por el software de comunicaciones :

Programas	Cantidad de memoria utilizada (EEPROM) bytes
ISR de SCI	1586
ISR de IC2	0043
ISR de IC3	0028
ISR de OC2	0148
INICIAVARS	0129
TRANSMITE_DATOS	0152
STATUSMSGDAT	0031
STARTCON	0033
CLOSECON	0040
SENDAGINMSG	0004
RESET_PROTOCOL	0030
STATUSMSG_RCV	0025
FREEBUFFRCV	0017
TRANSMITE_CSMA	0040
CRCGEN	0533
TOTAL	2835

Memoria utilizada en el manejo de buffers de recepción y transmisión y Variables del protocolo		
EEPROM	RAM	Descripción
-----	$x \cdot m$	m buffers de x bytes de recepción
-----	$5 \cdot m$	m descriptores de recepción de 5 bytes
$2 \cdot m$		m apuntadores a descriptores de recepción
	$6 \cdot n$	n descriptores de transmisión de 6 bytes
$2 \cdot n$		n apuntadores a descriptores de transmisión
	27	Variables internas del protocolo

donde:

- x=Tamaño del Buffer de Recepción
- m=Número de buffers de Recepción
- n=Número de buffers de Transmisión

Memoria RAM Total = $xm + 5m + 6n + 27$

Memoria EEPROM Total = $2m + 2n + 2835$

En la etapa de pruebas se manejaron 16 buffers de recepción de 255 bytes (o sea, 4080 bytes) y 16 buffers de transmisión, por lo tanto la memoria total ocupada por el protocolo de comunicaciones es :

RAM =4283 bytes

EEPROM=2899 bytes

4.7 Factores que limitan la velocidad máxima de transmisión

Uno de los factores que limitan la velocidad máxima a la cual puede trabajar un protocolo, es el tiempo de procesamiento por cada caracter recibido. Si el tiempo de procesamiento es muy grande puede provocar un "Overrun" (error de desbordamiento) en el sistema de recepción. En general, se debe de procesar el dato recibido, antes de que llegue el siguiente. El caso extremo se presenta cuando el transmisor envía los caracteres sin algún retardo entre ellos. Por lo tanto conociendo la velocidad de transmisión, se puede calcular el valor del tiempo máximo de procesamiento por caracter. Con un formato de 9 bits, 1 bit de inicio y 1 bit de paro, el tiempo máximo de procesamiento es igual a $\frac{11}{\text{velocidad}}$. En la Figura 4.6 se presenta el cálculo del tiempo máximo de procesamiento para las algunas velocidades de transmisión del HC11 (@ 8Mhz). La máxima velocidad de transmisión que puede manejar el sistema serial del HC11 son 125 Kbps, lo cual da como resultado un tiempo máximo de procesamiento de 0.088ms/caracter. En la Figura 4-5 y Figura 4-4 se muestra el tiempo de procesamiento asociado en la recepción de cada caracter de los mensajes de control y datos. Como se puede observar, el tiempo máximo de procesamiento en un caracter es de 0.0705ms (mensaje de datos, estado 6) , por lo que teóricamente el protocolo puede trabajar a 125 Kbps. Sin embargo, el servicio de otras interrupciones ó la desactivación de las interrupciones por un cierto periodo de tiempo, puede provocar el error de desbordamiento en el sistema serial del microcontrolador. Por lo tanto, al seleccionar la velocidad de transmisión de la red, además del tiempo máximo de procesamiento por

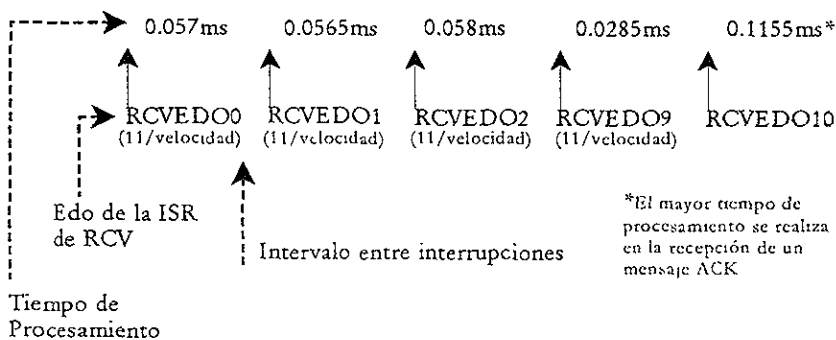


Figura 4-4 Tiempo de procesamiento asociado a la recepción de un mensaje de control

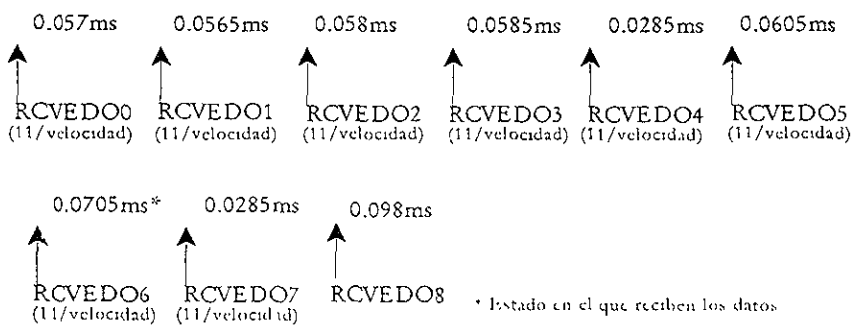


Figura 4-5 Tiempo de procesamiento asociado con la recepción de un mensaje de datos

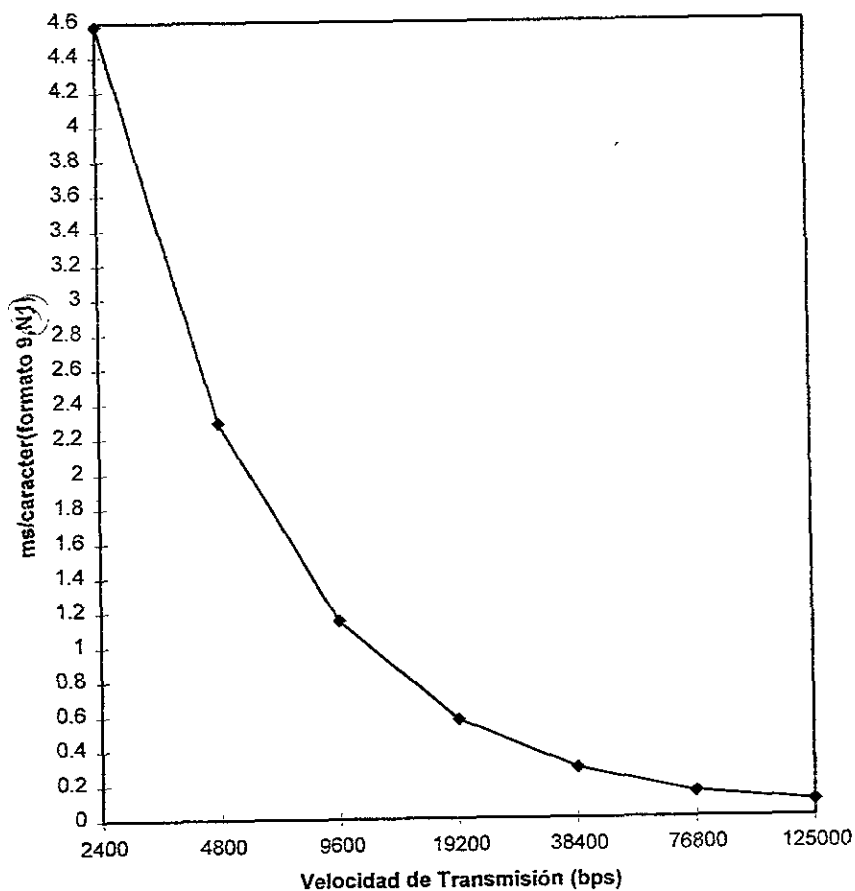


Figura 4-6

4.8 Elementos de programación

A continuación se mencionan el equipo y las herramientas utilizadas para el desarrollo de las estaciones HC11:

- Macroensamblador AVMAC11 para 68HC11.

- Simulador/Depurador AVSIM11 para 68HC11.
- Ensamblador de dominio público AS11
- Programa PCbug11 .
- Programador Universal Super Pro
- Compilador de PLD's y FPGA's CUPL de ATMEL para Windows v. 4.7b
- PC 486DX2 a 50 MHz con sistema operativo MSDOS 6.2 y Windows 95.
- Turbo C++ 3.0

CONCLUSIONES

En el presente trabajo se describió el diseño y la implantación las capas físicas y enlace de datos de un protocolo de comunicaciones para dispositivos de 8 bits. El protocolo incluye la mayoría de las funciones de comunicación de una red, tales como :

- Medio físico de interconexión : EIA-485, multimaestro, serial con un formato de 9 bits.
- Protocolo de acceso al medio : CSMA/CD.
- Servicio orientado a la conexión: 2 números de secuencia con reconocimientos positivos y negativos.
- Detección de errores : CRC-16
- Control de flujo : "Paquete de paro".

Se encontró que existe muy poca información acerca de la implantación de un protocolo de comunicaciones por lo que el presente trabajo puede ayudar a simplificar el diseño de las aplicaciones que hagan uso de una red de control

El protocolo fue diseñado de tal manera que pueda ser modificado para el desarrollo de diferentes aplicaciones, convirtiéndose en una herramienta útil para el desarrollador de aplicaciones de control distribuido. Esto es porque se ha hecho un análisis detallado de cada característica del protocolo de comunicaciones

Aunque este proyecto fue realizado a nivel experimental, es posible determinar su precio aproximado en el mercado mediante un análisis comparativo entre una solución comercial, llamada NBS (Nine Bit Solution) de la compañía Cinnetics Inc. cuyo precio es de 3000 USD*, y nuestra solución, del cual se desprenden las siguientes similitudes .

CONCLUSIONES

	NODO MAESTRO "NBS"	NODO ESCLAVO "NBS"	NUESTRA SOLUCIÓN
Espacio de Programa	16 KB	8KB	3KB
RAM	Depende de BUFSIZE		
Líneas de E/S	<ul style="list-style-type: none"> • TXD(PD1), RXD(PD0), • Un pin de salida para controlar al transmisor EIA-485 	<ul style="list-style-type: none"> • TXD(PD1), RXD(PD0). • PA4 : para controlar a TX EIA-485 • PA0 y PA1 para el funcionamiento de CSMA/CD 	
Interrupciones	Puerto Serial	Puerto Serial	
Temporizadores	Acumulador de pulsos	Ninguno	OC2 (Output Compare)
Herramienta de programación para usar el producto	Compilador de C	Ensamblador	Ensamblador

*Este precio de 3000 USD incluye :

- El software del nodo maestro
- El software del nodo esclavo (hasta 5 licencias)
- El programa Monitor de Red para MSDOS.
- Dos tarjetas RS-485/422 para PC basadas en el UART 82510

La principal diferencia entre el protocolo comercial y nuestra solución es el protocolo de control de acceso al medio. El primero maneja un protocolo centralizado: Polling (cuya principal desventaja se presenta cuando el nodo maestro deja de funcionar, pues ocasiona que toda la red se caiga), mientras que nuestra solución propone un protocolo descentralizado: CSMA/CD (lo cual representa una ventaja, ya que en el momento en que se descompone un nodo, la red continúa funcionando, es decir, no existe un nodo crítico).

Por último, al momento de concluir este trabajo, se encontraron 4 áreas susceptibles a ser depurados y desarrollados para una mayor eficiencia del sistema.

1. *Distancias*: Probar diferentes distancias, para realizar posibles modificaciones en los valores de los temporizadores asociadas con el tiempo de propagación
2. *Velocidad de transmisión* : Probar diferentes velocidades de transmisión, para observar como responde el protocolo con velocidades bajas(300 bps), velocidades medias (56Kbps) y velocidades altas(125kbps).
3. *Número de nodos* . Aumentar números de estaciones que se conectan a la red, por ejemplo: 8, 12, 24, 32, 64, 128, 256.
4. *Demanda del Protocolo* : Probar con diferentes aplicaciones que demanden de forma distinta al protocolo, es decir, la frecuencia con la que envían paquetes al medio, y su implicación con la eficiencia del protocolo.

Por otro lado, la aportación que este trabajo hace a la Facultad de Ingeniería es una posible aplicación en un taller de redes basados en microprocesadores, donde se realicen experimentos de las características más representativas de las redes de comunicaciones, con el objetivo de ver de forma tangible los conceptos del Modelo de Referencia OSI (Open Systems Interconnection), el cual comúnmente es tratado de forma abstracta en su enseñanza

CONCLUSIONES

APÉNDICES

Apéndice A EIA-485

En este apéndice se mencionan algunos de los puntos más importantes del estándar EIA-485. La nota de aplicación 1057 *Ten Ways of Bulletproof RS-485 Interfaces* de National Semiconductor sirvió como referencia.

Medio de interconexión

El estándar especifica únicamente las características de salida del transmisor y las de entrada del receptor, no define el medio de interconexión. Se pueden construir buses EIA-485 usando cables de par trenzado, cable plano y otros tipos de cable. Sin embargo, el cable par trenzado es el más común. Se puede usar un rango de calibres, pero los diseñadores utilizan frecuentemente el calibre 24 AWG. La impedancia característica del cable debe estar entre 100Ω y 120Ω .

Opciones de terminación

La opción de terminación más popular es conectar una resistencia entre el par de conductores en cada extremo. El valor de la resistencia debe equiparar el de la impedancia característica en modo diferencial del cable. Si el bus cuenta con esta terminación, no ocurrirán reflexiones, y la fidelidad de la señal es excelente.

Parámetros diferenciales de 485

Existen cuatro parámetros que son muy importantes para la transmisión de datos diferencial EIA 485: VOD, VOS, VGPD y VCM. La Figura A-1, Figura A-2, Figura A-3 y Figura A-4 ilustran estos parámetros, los cuales no son comunes en el mundo de transmisión no balanceada de las familias lógicas estándar.

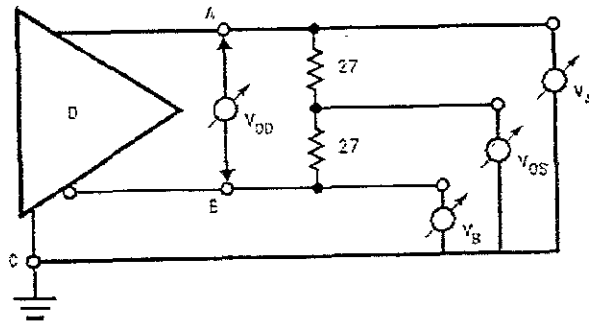


Figura A-1

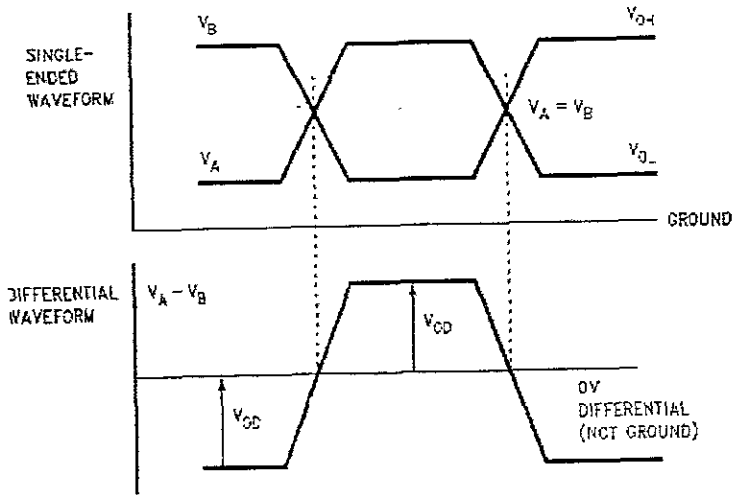


Figura A-2

VOD representa el voltaje de salida diferencial del transmisor a través de la carga de terminación. El estándar EIA-485 se refiere a este parámetro como voltaje de terminación (VT), pero VOD es también utilizado comúnmente. VOD se mide diferencialmente a través de la línea de transmisión (no a referencia a tierra). En cableados muy largos, la resistencia de DC atenúa VOD, pero el receptor requiere únicamente un voltaje de 200 mV

para asumir el estado apropiado. Por lo tanto, la atenuación no es un problema. A la salida del transmisor, VOD tiene un valor mínimo de 1.5V

VOS representa el voltaje de offset del transmisor medido en el punto central de la carga con respecto a la referencia de tierra del transmisor. VOS es también llamado VOC : voltaje de salida en modo común.

VCM representa el voltaje en modo común, por el cual EIA-485 es famoso. El límite es de -7V a 12V (El voltaje en modo común es definido como el promedio algebraico de los 2 voltajes referidos a la tierra local de las entradas del receptor ó a las salidas del transmisor)

El voltaje en modo común representa la suma de 3 voltajes : el primero, es el voltaje de offset en activo del transmisor. El segundo es el ruido acoplado en modo común en ambas líneas de la señal. El tercero es la diferencia de potencial de tierra entre el transmisor y receptor activos en el bus. Matemáticamente .

$$VCM = VOS + VNOISE + VGPD$$

VGPD representa la diferencia de potencial de tierra que puede existir entre nodos en el sistema. EIA-485 permite una variación de 7V en las tierras. Un corrimiento de 7V abajo de la barrera de la fuente de poder negativa produce un límite en modo común de -7V, mientras que 7V arriba de la barrera de la fuente de poder positiva produce el otro límite en modo común de 12V.

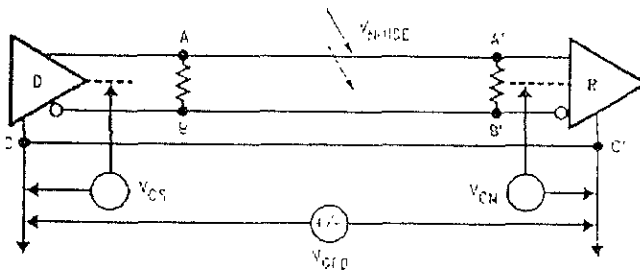


Figura A-3

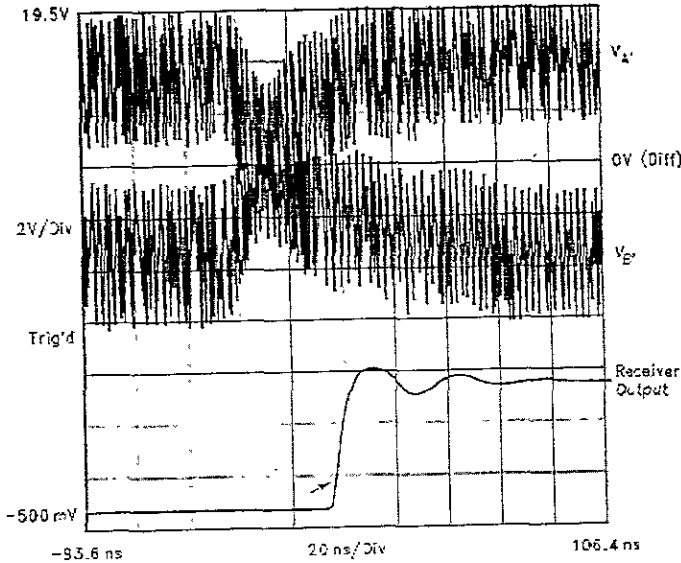


Figura A-4

Protección en contienda

Debido a que EIA-485 permite conectar varios transmisores a el bus, el estándar toca el tópico de la contienda. Cuando 2 o más drivers están en contienda, el estado de la señal en el bus no es garantizado. Si 2 drivers se encuentran activos al mismo tiempo y tratan de manejar el mismo estado, entonces el estado del bus es válido. Sin embargo, si los drivers están en estados contrarios, el estado del bus es indeterminado, debido a que el voltaje diferencial en el bus cae a un valor bajo, dentro el rango de umbral del receptor.

La contienda entre nodos puede dañar los circuito integrados. Si varios transmisores están en un estado y un transmisor maneja el estado contrario, este nodo puede suministrar un corriente tan alta de hasta 250 mA. Esta corriente provoca una excesiva disipación de energía. Una diferencia en los potenciales de tierra entre los nodos agrava esta disipación. En esta situación, la temperatura de la juntura del transmisor puede incrementarse más allá de sus límites. El estándar EIA-485 recomienda el uso de como el circuito de apagado

Parámetros de Entrada	Valores devueltos por el comando
ACCX= Número de bytes ACCY= Apuntador de datos ACCA= Dirección del nodo remoto	CARRY= Estado del comando ACCX= Apuntador al primer descriptor de transmisión.

CARRY	Definición
1	Los buffers de transmisión son insuficientes
0	Operación exitosa

Ejemplo de uso :

```
MSG    DB    "HOLA"
        LDAA  #1
        LCX  #4
        LDY  #MSG
        JSR  TRANSMITE_DATOS
        BCS  ERROR
        STX  STORE_PTRDES
```

RESTRICCIONES :

- No se puede transmitir un mensaje mayor a 64KB
- No se puede utilizar este comando cuando no exista conexión.
- La longitud de un mensaje debe ser mayor ó igual a uno.

NOTAS :El valor que se regresa en el acumulador X es posteriormente utilizado para preguntar acerca del estado de transmisión de los mensajes. Si un mensaje es mayor a 255 bytes, se fragmenta en tramas de 255 bytes

Cada vez que llega un mensaje de datos a un nodo se compara la dirección origen con STN_CONNECTED, si son diferentes se envia un mensaje ACK (No existe conexión)

STATUSMSG_DATOS

El comando STATUSMSG_DATOS devuelve al usuario el "status" de un bloque de datos que se haya transmitido mediante el comando TRANSMITE_DATOS. Los parámetros de entrada y salida se muestran en la siguiente tabla :

Parámetros de Entrada	Valores devueltos por el comando
ACCY= Apuntador devuelto por TRANSMITE_DATOS	ACCA= Estado del mensaje.

Los posibles estados de un mensaje de datos son :

Estado	Definición
0	Aun no transmitido
1	Transmitiéndose
2	Transmitido
3	Buffer no disponible
4	No existe conexión
5	Time-out

Ejemplo de uso :

```
MSG    DB      "HOLA"
        LDAA   #1
        LDX   #4
        LDY   #MSG
        JSR   TRANSMITE DATOS
        STX   STORE_PTRDES
LOOP   LDY   STORE_PTRDES
        JSR   STATUSMSGDAT
        CMFA  #TRANSMITIDO
        BEQ   ENVIAREPMAIN
        JSR   DESPACHADOR
        JMP   LOOP
```

NOTAS : Cuando el estado del mensaje es "Buffer no disponible", "Time-out" ó "No existe conexión" la transmisión del mensaje es suspendida. El usuario puede volver a intentar transmitir el mensaje mediante SENDAGAIN_MSG ó restaurar el protocolo mediante RESET_PROTOCOL. Si un mensaje es fragmentado en tres partes y la segunda

trama sufre un TIME-OUT, al momento de ejecutar SENDAGAIN_MSG, no se reenvía todo el mensaje , sino únicamente desde la trama que sufrió el TIME-OUT.

SENDAGAIN_MSG

El comando SENDAGAIN_MSG es ejecutado después de haber obtenido, mediante STATUSMSG_DAT, los estados “Time-out” ó “Buffer no disponible”. El comando reenvía el mensaje desde la trama que sufrió el TIME-OUT o que recibió Buffer No disponible.

No tiene parámetros de entrada y no regresa algún valor al usuario.

CLOSECON

El comando CLOSECON permite liberar una conexión.

No tiene parámetros de entrada. A través del descriptor cerrar conexión se devuelve el estado del comando.

Valores devueltos por el comando
En [DESC_CC+STATUS]= Estado del comando

DESC_CC Etiqueta que contiene la dirección del primer byte del descriptor Cerrar Conexión

STATUS : Símbolo que contiene el offset del campo STATUS del descriptor DESC_CC

Los estados del comando son .

Estado	Definición
0	Aun no transmitido
1	Transmitiéndose
2	Conexión cerrada
3	Time-Out

Ejemplo de uso

```

      JSR      CLOSECON
REV_STATUS
      LDX      #DESC_CC
      LDAA     STATUS,X
      CMPA     #CONEXION_CERRADA
      BEQ      FIN
      JMP      REV_STATUS

```

RESTRICCIONES : El usuario debe ejecutar este comando cuando se hayan transmitido todos los mensajes, debido a que CLOSECON libera los buffers de transmisión.

NOTAS :

Si el estado del comando es "Time-out" el usuario puede volver a intentar transmitir el mensaje mediante CLOSECON ó restaurar el protocolo mediante RESET_PROTOCOL. Si se cierra una conexión mediante RESET_PROTOCOL, se responderá a éste mediante ACK (No existe conexión).

El nodo receptor del CC informa al usuario, sobre la desconexión, limpiando la variable STN_CONNECTED.

RESET_PROTOCOL

El comando RESET_PROTOCOL los buffers de recepción y de transmisión, además de limpiar la variable STN_CONNECTED.

El comando no tienen parámetros de entrada ni de salida.

NOTAS : El comando puede utilizarse como una alternativa cuando se tiene conexión con un nodo remoto que no responde, a los mensajes de datos o a un CC.

STATUSMSGRCV

El comando STATUSMSGRCV se ejecuta para saber si ha llegado algún mensaje.

El comando devuelve la dirección del descriptor de recepción.

Si no ha llegado algún mensaje se devuelve en ACCX el valor 0.

Un descriptor de recepción tiene la siguiente forma :

Valores devueltos por el comando
 En ACCX se devuelve el apuntador
 al descriptor de recepción.

Ejemplo de uso :

```
DATOS EQU      $2
POINTER EQU    $3
REV_RCV
    JSR        STATUSMSGRCV
    CPX        #0
    BEQ        REV_RCV
    LDAA       DATOS,X      *En accA se almacena la longitud*
    LDX        POINTER,X    *En accX se carga el apuntador de datos
    LDAA       0,X          *En accA se almacena el primer dato
```

NOTAS :En el campo “Tipo de marco” del descriptor de recepción se encuentra la información de fragmentación :

Valor en Tipo de Marco	Significado
0	STX
1	ETX

FREEBUFFRCV

Este comando libera un buffer de recepción. Se debe de utilizar después de que se leyeron los datos de un buffer de recepción.

No tiene parámetros de entrada ni parámetros de salida

Ejemplos de uso

```
REV_RCV
    JSR        STATUSMSGRCV
    CPX        #0
    BEQ        REV_RCV
    LDAA       #1
    JSR        CONTROLD
    LDX        POINTER,X
    JSR        ESCRIBFD
    JSR        FREEBUFFRCV
```

INICIAVARS

Este comando permite iniciar los valores de algunas variables utilizadas por el protocolo. Este comando debe de invocarse al inicio del programa principal.

No tiene parámetros de entrada ni de salida

Ejemplo de uso :

```
LDS    #$3FFF
JSR    INICIAVARS
LDAA  #1
JSR    STARTCON
```

Manejo de buffers de recepción

El manejo de los buffers de recepción es relativamente sencillo. Se utilizan buffers de tamaño fijo de 255 bytes cada uno, 4080 en total. Cada buffer es asociado con un descriptor de recepción (5 bytes cada uno, 80 bytes en total). Se utiliza una cola circular en donde se almacenan los apuntadores de descriptores de recepción(32 bytes en total). La cola circular es manejada por un apuntador de salida y un apuntador de entrada.

El procedimiento para definir los buffers es el siguiente :

1. Se definen en memoria RAM los 16 buffers de 255 bytes y 80 bytes para los descriptores de recepción

```
BUFFPOOLRCV    DS 255*16 ;16 buffers de 255 bytes
BUFFPOOLDES_RCV DS 16*5  ;espacio en donde se almacena físicamente los 16 descriptores
```

2. En memoria EEPROM se crea una cola de apuntadores de descriptores de recepción (32 bytes en total)

```
BUFFDESRCV    ;16 apuntadores a descriptores de recepcion
FDB    0*5+BUFFPOOLDES_RCV
FDB    1*5+BUFFPOOLDES_RCV
..de la misma manera hasta el 15...
FDB    15*5+BUFFPOOLDES_RCV
```

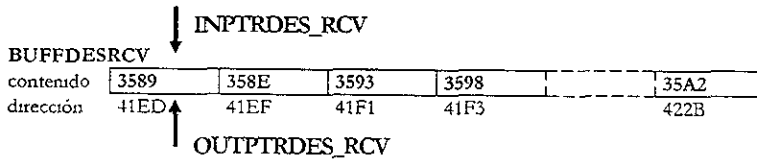
3. Se definen, en RAM, los apuntadores de salida y de entrada de la cola de apuntadores de descriptores de recepción.

OUTPTRDES_RCV DS 2 ;Apuntador de salida del buffer de descriptores de Recepcion
 INPTRDES_RCV DS 2 ;Apuntador de entrada del buffer de descriptores de Recepcion

4. Se inician los valores de los apuntadores de la cola circular. Esta rutina se encuentra en

```

INICIAVARS
LDD #BUFFDESRCV
STD INPTRDES_RCV
STD OUTPTRDES_RCV
    
```



5. Se inician los valores de los apuntadores de la cola circular. Esta rutina se encuentra en

```

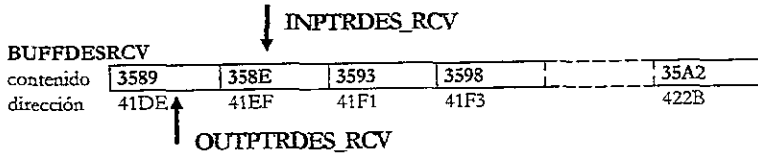
INICIAVARS.
LDX #(0*255+BUFFPOOLRCV)
STX 0*5+3+BUFFPOOLDES_RCV
LDX #(1*255+BUFFPOOLRCV)
STX 1*5+3+BUFFPOOLDES_RCV
...de la misma manera hasta el 15...
LDX #(15*255+BUFFPOOLRCV)
STX 15*5+3+BUFFPOOLDES_RCV
    
```

BUFFPOOLDESRCV

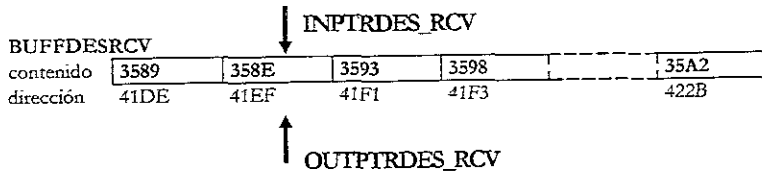
contenido	dd	tm	lo	25	99	dd	tm	lo	26	98
direccion	3589	358A	358B	358C	358D	358E	358F	3590	3591	3592

Símbolo	Descripción
dd	dirección destino
tm	tipo de marco
l	longitud

Cada vez que llega un mensaje de datos, el apuntador de entrada de la cola circular se incrementa y el usuario, mediante STATUSMSGDAT, recibe el apuntador del descriptor de recepción.



Cada vez que se libera un buffer de recepción (FREEBUFRCV), se incrementa el apuntador de salida de la cola circular.



Manejo de buffers de transmisión

Es muy parecido al manejo de buffers de recepción. Se definen 16 descriptores de transmisión de 6 bytes cada uno (96 bytes en total). Se utiliza una cola circular en donde se almacenan los apuntadores de descriptores de transmisión (32 bytes en total). La cola circular es manejada por un apuntador de salida y un apuntador de entrada.

El procedimiento para definir los buffers y la cola circular es el siguiente :

1. Se definen en RAM 16 descriptores de transmisión :

```

BUFFPOOLDES_XMT DS    96 ;Espacio en donde se almacena físicamente los 16 descriptores
    
```

2. Se definen, en memoria EEPROM, la cola de apuntadores de descriptores de transmisión (32 bytes).

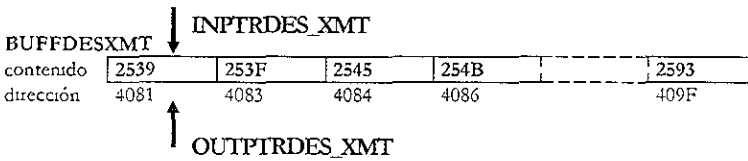
```

*****
*Inicializamos a BUFFDESXMT con los apuntadores a los descriptores:
*No los puedo poner en RAM porque se borrarían cuando apague el uP
*Definición de Buffers de transmisión
*****
BUFFDESXMT
    FDB    0*6+BUFFPOOLDES_XMT
    FDB    1*6+BUFFPOOLDES_XMT
    ...asi hasta el 15....
    
```

FDB 15*6+BUFFPOOLDES_XMT

4. Se inician los valores de los apuntadores de la cola circular. Esta rutina se encuentra en INICIAVARS

```
LDD #BUFFDESXMT ;Se inicializan los apuntadores de salida y
STD INPTRDES_XMT ;y de entrada del buffer de apuntadores
STD OUTPTRDES_XMT ;de descriptores
```

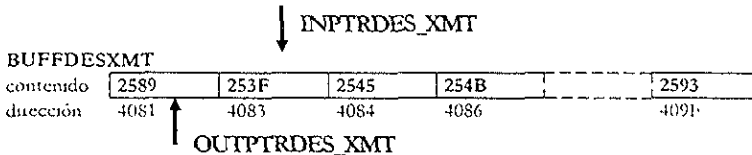


BUFFPOOLDESXMT

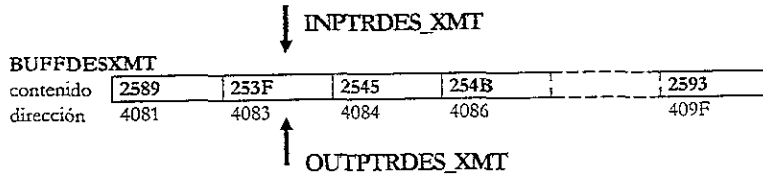
contenido	dd	tm	st	lo	ad	ad	dd	tm	st	lo
dirección	2539	253A	253B	253C	253D	253E	253F	2540	2541	2542

Símbolo	Descripción
dd	dirección destino
tm	tipo de marco
st	estado
l	longitud
ad	apuntador de datos(2 bytes)

La rutina TRANSMITE_DATOS utiliza los descriptores de transmisión para pasar información a la ISR de transmisión. Cada vez que TRANSMITE DATOS llena un descriptor incrementa el apuntador de entrada de la cola circular de los descriptores de transmisión.



La ISR de recepción cada vez que recibe un ACK de datos (ACK/0 ACK/1) libera un descriptor de transmisión :



Apéndice D Listado parcial de programas

Variables

```

seg      regs
_STARTREGS

DEFSEG  VARS,START=$2500,ABSOLUTE
SEG     VARS           ; RAM Starts at $2500
    
```

DEFINICION DE MACROS

```

de_enable %MACRO      ;Macro para habilitar el transmisor RS485 PA3->DE
LDAA     PORTA ;Hay que tener cuidado porque el acc A no se guarda.
ORAA     #$10  ;Se activa el DE del transmisor rs485
STAA     PORTA
%ENDM

de_disable %MACRO     ;Se desactiva DE, se envia 0 a PA4
LDAA     PORTA ;Hay que tener cuidado porque el acc A no se guarda.
ANDAA    #SEF
STAA     PORTA
%ENDM

set9th %MACRO         ;Macro para activar el 9th bit, 9N1, y wake_up por turk_address
LDAA     #$50 ;Hay que tener cuidado porque el acc A no se guarda.
STAA     SCCR1
%ENDM

gothere_xmt %MACRO
STAA     INTXMTSTATE ;Macro para modificar el estado de la ISR de transmision
RTI      ;y saliese de la ISR
%ENDM

forward_rcv %MACRO   ;Macro que salta al edo siguiente de recepcion
INC      INTRCVSTATE
RTI
%ENDM

forward_xmt %MACRO   ;Macro que salta al edo siguiente de transmision
INC      INTXMTSTATE
RTI
%ENDM

a_dormir %MACRO      ;Macro que pone a dormir a la estacion
LDAA     #(RE OR RIE OR RWU OR TD)
STAA     SCCR2
%ENDM

incnumsecxmt %MACRO  ;Macro que incrementa el numero de secuencia
;de transmision
    
```

```

%GENSYM FIN_XMT
LDAA NUMSECXMT
BNE CLRNUMSECXMT
INC NUMSECXMT
JMP FIN_XMT
CLRNUMSECXMT CLR NUMSECXMT
FIN_XMT
%ENDM

```

Programa para calcular el CRC por medio de tabla

La tabla obtenida de CRC-16 :

```

unsigned int tablacrc[] =
{0x0000, 0x8005, 0x800f, 0x000a, 0x801b, 0x001e, 0x0014, 0x8011, 0x8033, 0x0036,
0x003c, 0x8039, 0x0028, 0x802d, 0x0027, 0x8063, 0x0066, 0x006c, 0x8069,
0x0078, 0x807d, 0x8077, 0x0072, 0x0050, 0x8055, 0x805f, 0x005a, 0x804b, 0x004e,
0x0044, 0x8041, 0x80c3, 0x00c6, 0x00cc, 0x80c9, 0x00d8, 0x80dd, 0x80d7, 0x00d2,
0x00f0, 0x80f5, 0x80ff, 0x00fa, 0x80eb, 0x00ee, 0x00e4, 0x80e1, 0x00a0, 0x80a5,
0x80af, 0x00aa, 0x80bb, 0x00be, 0x00b4, 0x80b1, 0x8093, 0x0096, 0x009c, 0x8099,
0x0088, 0x808d, 0x8087, 0x0082, 0x8183, 0x0186, 0x018c, 0x8189, 0x0198, 0x819d,
0x8197, 0x0192, 0x01b0, 0x81b5, 0x81bf, 0x01ba, 0x81ab, 0x01ae, 0x01a4, 0x81a1,
0x01e0, 0x81e5, 0x81ef, 0x01ea, 0x81fb, 0x01fe, 0x01f4, 0x81f1, 0x81d3, 0x01d6,
0x01dc, 0x81d9, 0x01c8, 0x81cd, 0x81c7, 0x01c2, 0x0140, 0x8145, 0x814f, 0x014a,
0x815b, 0x015e, 0x0154, 0x8151, 0x8173, 0x0176, 0x017c, 0x8179, 0x0168, 0x816d,
0x8167, 0x0162, 0x8123, 0x0126, 0x012c, 0x8129, 0x0138, 0x813d, 0x8137, 0x0132,
0x0110, 0x8115, 0x811f, 0x011a, 0x810b, 0x010e, 0x0104, 0x8101, 0x8303, 0x0306,
0x030c, 0x8309, 0x0318, 0x831d, 0x8317, 0x0312, 0x0330, 0x8335, 0x833f, 0x033a,
0x832b, 0x032e, 0x0324, 0x8321, 0x0360, 0x8365, 0x836f, 0x036a, 0x837b, 0x037e,
0x0374, 0x8371, 0x8353, 0x0356, 0x035c, 0x8359, 0x0348, 0x834d, 0x8347, 0x0342,
0x03c0, 0x83c5, 0x83cf, 0x03ca, 0x83db, 0x03de, 0x03d4, 0x83d1, 0x83f3, 0x03f6,
0x03fc, 0x83f9, 0x03e8, 0x83ed, 0x83e7, 0x03e2, 0x83a3, 0x03a6, 0x03ac, 0x83a9,
0x03b8, 0x83bd, 0x83b7, 0x03b2, 0x0390, 0x8395, 0x839f, 0x039a, 0x838b, 0x038e,
0x0384, 0x8381, 0x0280, 0x8285, 0x828f, 0x028a, 0x829b, 0x029e, 0x0294, 0x8291,
0x82b3, 0x02b6, 0x02bc, 0x82b9, 0x02a8, 0x82ad, 0x82a7, 0x02a2, 0x82e3, 0x02e6,
0x02ec, 0x82e9, 0x02f8, 0x82fd, 0x82f7, 0x02f2, 0x02d0, 0x82d5, 0x82df, 0x02da,
0x82cb, 0x02ce, 0x02c4, 0x82c1, 0x8243, 0x0246, 0x024c, 0x8249, 0x0258, 0x825d,
0x8257, 0x0252, 0x0270, 0x8275, 0x827f, 0x027a, 0x826b, 0x026e, 0x0264, 0x8261,
0x0220, 0x8225, 0x822f, 0x022a, 0x823b, 0x023e, 0x0234, 0x8231, 0x8213, 0x0216,
0x021c, 0x8219, 0x0208, 0x820d, 0x8207, 0x0202};

```

El programa en ensamblador del HC11 utilizado para calcular el CRC es :

CRCGEN

```

PSHX
PSHA
EORA CRCFINAL
LDX #TABLECRC
TAB
ABX
ABX
LDD 0,X

```

```
EORA   CRCFINAL+1
STD    CRCFINAL
PULA
PULK
RTS
```

ISR del SCI

```
*****
*      SCIINT
*      Esta es la rutina de servicio de interrupcion
*      Se salta al estado correspondiente
*      Vector FFD6
*****
        defseg sciint, START=$5A00, absolute
        seg sciint
INTRCVTBL EQU *
RCVED00 EQU    LOW (*-INTRCVTBL)/2
          FDB    REDO0
RCVED01 EQU    LOW (*-INTRCVTBL)/2
          FDB    REDO1

*****
*      ¿Es mi direccion?
*****
REDO0 LDA    SCDR
        CMPA  #STN_ADDR    ;¿Es mi direccion?
        BEQ   ..SALTAEDO1
        LDA  #RCVED00      ;y nos quedamos en el mismo estado
        gothere_rcv
..SALTAEDO1
        JSR   CRCGEN        ;Si concuerdan las direccion, saltamos
        forward_rcv        ;a recibir la direccion origen
```

ISR DE IC2

```
*****
*IC2INT (Interrupcion por IDLE) Vector en FFDC
*esta es la rutina de servicio de interrupcion de IC2
*La entrada IC2 genera una interrupcion cuando hay un flanco
*de bajada, indicando que se encontro un IDLE
*****
        defseg idleint, START=$5600, ABSOLUTE
        seg idleint
        LDX  # STARTREGS
        BCLR XTCTL2, X, (EDG2B OR EDG2A) ;Desactivo INT por IDLE
        BCLR XTFLG1, X, (LOW NOT IC2F)  ;Se limpia bandera IC2I
        de_enable ;Activo DE del 485
        LDA  SCSR ;Debemos de borrar RDRF, dubico
        LDA  SCDR ;a que vamos a despertar al receptor
        LDA  # (RIE OR RE OR TE) ;Despertamos al receptor, RWU=0
        STAA SCDR2
        LDA  #STN_ADDR ;Envio mi direccion para detectar colisiones
        STAA SCDR
        LDA  #RCVED014 ;Saltamos al edo 14 de rcv para ver si existe una colision
        STAA INTRCVSTAT
        R11
```

ISR DE IC3

```

*****
* IC3INT (Interrupcion por flanco de bajada de RX)
* Esta es la rutina de servicio de interrupcion de IC3
* La entrada IC3 genera una interrupcion cuando hay un flanco
* de bajada, indicando que otro dispositivo con un tiempo de
* espera menor empezo a transmitir
* Vector en FFEA
*****
defseg rxflanco,START=$5700,ABSOLUTE
seg rxflanco
LDX #_STARTREGS
BCLR XTMSK1,X,OC2I ;Se desactiva INT por tiempo de espera
BCLR XTCTL2,X,EDG3B ;Se desactiva INT por IC3
BCLR XTFLG1,X,(LOW NOT IC3F) ;Se limpia bandera IC3f
BSET XTCTL2,X,(EDG2B) ;activo INT por IDLE
RTI

```

ISR DE OC2

```

*****
* OC2INT (Interrupcion por Timer de Espera y Timer de Respuesta )
* Esta es la rutina de servicio de interrupcion de OC2.
* Si TIPODETIMER=0, entonces se ejecuta la ISR DE TIEMPO DE ESPERA
* si es igual a 1, entoces se ejecuta la ISR de TIMER DE RESPUESTA
* Vector en FFE6
*****
defseg timerseg,START=$5800,ABSOLUTE
seg timerseg
LDAA TIPODETIMER ;¿Qué tipo de TIMER hay que procesar?
BNE RESPTIMER ;Salta a RESPTIMER
set9th ;Activa 9th bit
de enable ;Activa el driver 485
LDX #_STARTREGS
BCLR XTCTL2,X,EDG3B ;Se desactiva la INT de RX flanco de bajada
LDX INTXMTPTR ;Empieza XMIEDO0:
LDAA 0,X ;Se lee la Direccion Destino
STAA SCDR ;Se envia
JSR CROGEN ;Se genera el CRC
LDAA #TRANSMITIENDOSE ;Se cambia el status del mensaje
STAA STATUS,X
LDAA #XMIEDO1 ;Se salta al edo1 de la ISR de XMT
STAA INTXMISTATE
LDAA #(TE OR TCIE OR RE OR RIE OR RWU) ;Se activan las interrupciones por TC
STAA SCCR2 ;Y se desactiva al sistema receptor
LDX #_STARTREGS
BCLR XTMSK1,X,OC2I ;Se desactiva la INT por tiempo de espera
BCLR XTFLG1,X,(LOW NOT OC2F) ;Se limpia OC2F
RTI
RESPTIMER ;Si el timer es de respuesta entonces
LDX #_STARTREGS

```



```

BCLR    XTMSK1,X,OC2I ;Desactivamos la INT por TIMER
BCLR    XTFLG1,X,LOW NOT OC2F ;Limpio la bandera del timer
LDX     INTXMTPTR
LDAA    1,X
BPL     SENDRESP      ;Si el mensaje es de dato(N=0) se envia un REP
LDAA    TIME_OUTCOUNT ;de lo contrario se revisa el num. de time-out's
CMPA    #3
BNE     SENDAGAIN    ;si no ha llegado a 3 se vuelve a enviar el mensaje
CLR     TIME_OUTCOUNT ;de lo contrario
LDAA    #TIMEOUT      ;se le avisa al usuario por medio del
STAA    STATUS,X      ;descriptor que existe un TIMEOUT
LDAA    1,X
ANDA    #TIPODECONTROL
CMPA    #REPMMSG
BEQ     RESTAURA_INTXMTPTR ;¿El mensaje que se envio fue REP?
CMPA    #SCMSG        ,si no lo es entonces preguntamos
BNE     SALIREP       ;¿El mensaje que se envio fue SC?
CLR     BLOCKCONN     ;si lo es activamos la entrada de conexiones
SALIREP RTI           ;si no fue simplemente nos salimos
SENDRESP
STX     INTXMTPTR.BAK ;Si el mensaje es de datos, se
LDX     #DESC REP     ;envia un REP
LDAA    STN_CONNECTED
STAA    0,X
LDAA    #(CONTROL+REPMMSG)
ORAA    NUMSECMT
STAA    1,X
STX     INTXMTPTR
JSR     TRANSMITE_CSMA
RTI

SENDAGAIN ;En el caso que todavia la cuenta
INC     TIME_OUTCOUNT ;de TIMEOUT's no haya llegado a 3
JSR     TRANSMITE_CSMA ;se vuelve a enviar el mensaje
RTI

RESTAURA_INTXMTPTR ;Si el mensaje de control REP llega
LDX     INTXMTPTR.BAK ;a 3 Timeouts se recupero el apuntador
STX     INTXMTPTR     ;al descriptor de datos y se
LDAA    #TIMEOUT      ;actualiza su STATUS
STAA    STATUS,X
RTI

```

Programas en la PC del Monitor de Red

serialp.c

```

#include <stdio.h>
#include <stdlib.h>
#include <conio.h>
#include <dos.h>
#include "serial.h"

PORT *port;
void main()
{

```

```

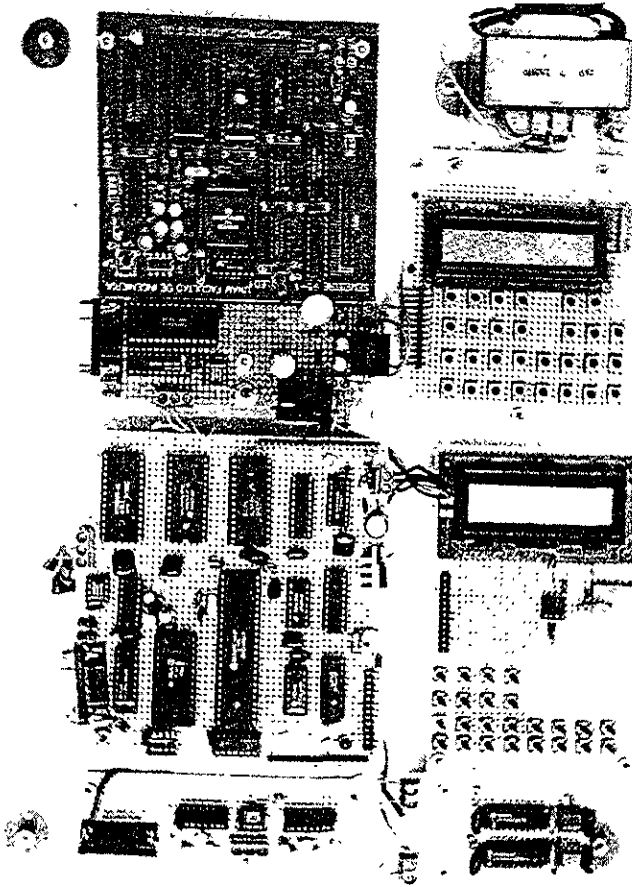
int c,c9th;
clrscr();
textcolor(RED);
cprintf("Caracter Recibido ");
textcolor(BLUE);
cprintf("Caracter Transmitido ");
textcolor(GREEN);
cprintf("Caracter Transmitido 9th");
gotoxy(1,2);
port_inicio(COM1, COM1_INT);
if (port==NULL) {
    printf("No se pudo abrir el puerto");
    exit(-1);
}
port_set(2400,LPSTICK,8,1);
activa_int();
/*El programa esta en este loop hasta que el usuario ingresa ESC. En el
loop se lee un caracter del teclado y se envia a el puerto. Despues lee
un caracter del puerto, y lo imprime en pantalla*/

while(1)
{
    c=port_getc();
    if (c!=100) {
        textcolor(RED);
        cprintf("%c,%X ",(unsigned char)c,(unsigned char)c);
    }
    c9th=port_getc9th();
    if(c9th!=100){
        textcolor(LIGHTCYAN);
        cprintf("%c,%X ",(unsigned char)c9th,(unsigned char)c9th);
    }

    if(kbhit()) {
        c=getch();
        if (c==27)
            break;
        if (c!=0) {
            textcolor(BLUE);
            cprintf("%c",c);
            port_putc((unsigned char)c);}
    }
}
/* Salir cuando sea presionado ESC (ASC=27) */
port_close();
}

```

Apéndice E Fotografía del sistema utilizado



Apéndice F Hojas de especificación

19-1167 REV. D 1/99

MAXIM

Low-Power, Slew-Rate-Limited RS-485/RS-422 Transceivers

General Description

The MAX481, MAX483, MAX485, MAX487-MAX491, and MAX1487 are low-power transceivers for RS-485 and RS-422 communication. Each part contains one driver and one receiver. The MAX483, MAX487, MAX488 and MAX489 feature reduced slew-rate drivers that minimize EMI and reduce reflections caused by improperly terminated cables, thus allowing error free data transmission up to 250kbps. The driver slew rates of the MAX481, MAX485, MAX490, MAX491 and MAX1487 are not limited, allowing them to transmit up to 2.5Mbps.

These transceivers draw between 120µA and 500µA of supply current when unloaded or fully loaded with disabled drivers. Additionally, the MAX481, MAX483, and MAX487 have a low-current shutdown mode in which they consume only 0.1µA. All parts operate from a single 5V supply.

Drivers are short circuit current limited and are protected against excessive power dissipation by thermal shutdown circuitry that places the driver outputs into a high-impedance state. The receiver input has a fail-safe feature that guarantees a logic-high output if the input is open circuit.

The MAX487 and MAX1487 feature quarter-unit-load receiver input impedance, allowing up to 128 MAX487/MAX1487 transceivers on the bus. Full-duplex communications are obtained using the MAX488-MAX491, while the MAX481, MAX483, MAX485, MAX487, and MAX1487 are designed for half-duplex applications.

Applications

- Low-Power RS-485 Transceivers
- Low-Power RS-422 Transceivers
- Level Translators
- Transceivers for EMI-Sensitive Applications
- Industrial-Control Local Area Networks

Features

- ◆ In µMAX Package: Smallest 8-Pin SO
- ◆ Slew-Rate Limited for Error-Free Data Transmission (MAX483/487/488/489)
- ◆ 0.1µA Low-Current Shutdown Mode (MAX481/483/487)
- ◆ Low Quiescent Current:
120µA (MAX483/487/488/489)
230µA (MAX1487)
300µA (MAX481/485/490/491)
- ◆ -7V to +12V Common-Mode Input Voltage Range
- ◆ Three-State Outputs
- ◆ 30ns Propagation Delays, 5ns Skew (MAX481/485/490/491/1487)
- ◆ Full-Duplex and Half-Duplex Versions Available
- ◆ Operate from a Single 5V Supply
- ◆ Allows up to 128 Transceivers on the Bus (MAX487/MAX1487)
- ◆ Current-Limiting and Thermal Shutdown for Driver Overload Protection

Ordering Information

PART	TEMP. RANGE	PIN-PACKAGE
MAX481CPA	0°C to +70°C	8 Plastic DIP
MAX481CSA	0°C to +70°C	8 SO
MAX481CUA	0°C to +70°C	8 µMAX
MAX481CD	0°C to +70°C	Dice*

Ordering information continued at end of data sheet.
* Contact factory for dice specifications.

Selection Table

PART NUMBER	HALF/FULL DUPLEX	DATA RATE (Mbps)	SLEW-RATE LIMITED	LOW-POWER SHUTDOWN	RECEIVER/DRIVER ENABLE	QUIESCENT CURRENT (µA)	NUMBER OF TRANSMITTERS ON BUS	PIN COUNT
MAX481	Half	2.5	No	Yes	Yes	300	32	8
MAX483	Half	0.25	Yes	Yes	Yes	120	32	8
MAX485	Half	2.5	No	No	Yes	300	32	8
MAX487	Half	0.25	Yes	Yes	Yes	120	128	8
MAX488	Full	0.25	Yes	No	No	120	32	8
MAX489	Full	0.25	Yes	No	Yes	120	32	14
MAX490	Full	2.5	No	No	No	300	32	6
MAX491	Full	2.5	No	No	Yes	300	32	14
MAX1487	Half	2.5	No	No	Yes	230	128	8

MAX481/MAX483/MAX485/MAX487-MAX491/MAX1487

Low-Power, Slew-Rate-Limited RS-485/RS-422 Transceivers

MAX481/MAX483/MAX485/MAX487-MAX491/MAX1487

ABSOLUTE MAXIMUM RATINGS

Supply Voltage (V _{CC})	12V	14-Pin SO (derate 8.33mW/°C above +70°C)	867mW
Control Input Voltage (RE, DE)	-0.5V to (V _{CC} + 0.5V)	8-Pin µMAX (derate 4.1mW/°C above +70°C)	830mW
Driver Input Voltage (DI)	-0.5V to (V _{CC} + 0.5V)	8-Pin CERDIP (derate 8.00mW/°C above +70°C)	640mW
Driver Output Voltage (A, B)	-8V to +12.5V	14-Pin CERDIP (derate 9.09mW/°C above +70°C)	727mW
Receiver Input Voltage (A, B)	8V to +12.5V	Operating Temperature Ranges	
Receiver Output Voltage (RO)	0.5V to (V _{CC} + 0.5V)	MAX4_ C_ /MAX1487C_ A	0°C to +70°C
Continuous Power Dissipation (T _A = +70°C)		MAX4_ E_ /MAX1487E_ A	40°C to +85°C
8-Pin Plastic DIP (derate 9.09mW/°C above +70°C)	727mW	MAX4_ MJ_ /MAX1487MJA	-55°C to +125°C
14-Pin Plastic DIP (derate 10.00mW/°C above +70°C)	800mW	Storage Temperature Range	
8-Pin SO (derate 5.88mW/°C above +70°C)	471mW	-65°C to +160°C	
		Lead Temperature (soldering 10sec)	
		+300°C	

Stresses beyond those listed under "Absolute Maximum Ratings" may cause permanent damage to the device. These are stress ratings only; normal operation of the device at these or any other conditions beyond those indicated in the operational sections of the specification is not implied. Exposure to absolute maximum rating conditions for extended periods may affect device reliability.

DC ELECTRICAL CHARACTERISTICS

V_{CC} = 5V ±5% T_A = T_{MIN} to T_{MAX} unless otherwise noted (Notes 1, 2)

PARAMETER	SYMBOL	CONDITIONS	MIN	TYP	MAX	UNITS
Differential Driver Output (no load)	V _{OD1}				5	V
Differential Driver Output (with load)	V _{OD2}	R = 50Ω (RS-422)		2		V
		R = 27Ω (RS-485), Figure 4	1.5		5	V
Change in Magnitude of Driver Differential Output Voltage for Complementary Output States	ΔV _{OD}	R = 27Ω or 50Ω, Figure 4			0.2	V
Driver Common Mode Output Voltage	V _{OC}	R = 27Ω or 50Ω, Figure 4			3	V
Change in Magnitude of Driver Common Mode Output Voltage for Complementary Output States	ΔV _{OD}	R = 27Ω or 50Ω, Figure 4			0.2	V
Input High Voltage	V _{IH}	DE, DI, RE	2.0			V
Input Low Voltage	V _{IL}	DE, DI, RE			0.9	V
Input Current	I _{IN}	DE, DI, RE			-2	µA
Input Current (A, B)	I _{IN2}	DE = 0V V _{CC} = 0V or 5.25V, all devices except MAX487/MAX1487	V _{IN} = 12V		1.0	mA
			V _{IN} = 7V		0.8	mA
		MAX487/MAX1487	V _{IN} = 12V		0.25	mA
		DI, 0V V _{CC} = 0V or 5.25V	V _{IN} = 7V		0.2	mA
Receiver Differential Threshold Voltage	V _{IH1}	7V ≤ V _{CM} ≤ 12V	0.2		0.2	V
Receiver Input Hysteresis	ΔV _{IH}	V _{CM} = 0V		20		mV
Receiver Output High Voltage	V _{OH}	I _O = 4mA V _{ID} = 200mV	3.5			V
Receiver Output Low Voltage	V _{OL}	I _O = 4mA V _{ID} = 200mV			0.4	V
Three-State (high impedance) Output Current at Receiver	I _{OZR}	0.4V ≤ V _O ≤ 2.4V			±1	µA
Receiver Input Resistance	R _{IN}	7V ≤ V _{CM} ≤ 12V, all devices except MAX487/MAX1487		12		kΩ
		7V ≤ V _{CM} ≤ 12V, MAX487/MAX1487		48		kΩ

Low-Power, Slew-Rate-Limited RS-485/RS-422 Transceivers

DC ELECTRICAL CHARACTERISTICS (continued)

(V_{CC} = 5V ± 5%, T_A = T_{MIN} to T_{MAX}, unless otherwise noted.) (Notes 1-2)

PARAMETER	SYMBOL	CONDITIONS	MIN	TYP	MAX	UNITS
No Load Supply Current (Note 3)	I _{CC}	MAX488/MAX489 DE, DI, RE = 0V or V _{CC}		120	250	μA
		MAX490/MAX491 DE, DI, RE = 0V or V _{CC}		300	500	
		MAX481/MAX485 RE = 0V or V _{CC}	DE = V _{CC}	500	900	
			DE = 0V	300	500	
		MAX1487, RE = 0V or V _{CC}	DE = V _{CC}	300	500	
			DE = 0V	230	400	
		MAX483/MAX487, RE = 0V or V _{CC}	DE = 5V	MAX483	350	
MAX487	250			400		
	DE = 0V		120	250		
Supply Current in Shutdown	I _{SHDN}	MAX481/483/487 DE = 0V RE = V _{CC}		0.1	10	μA
Driver Short-Circuit Current V _O = High	I _{OSD1}	-7V ≤ V _O ≤ 12V (Note 4)	35		250	mA
Driver Short-Circuit Current V _O = Low	I _{OSD2}	-7V ≤ V _O ≤ 12V (Note 4)	35		250	mA
Receiver Short-Circuit Current	I _{OSR}	0V ≤ V _O ≤ V _{CC}	7		95	mA

SWITCHING CHARACTERISTICS—MAX481/MAX485, MAX490/MAX491, MAX1487

(V_{CC} = 5V ± 5% T_A = T_{MIN} to T_{MAX}, unless otherwise noted.) (Notes 1-2)

PARAMETER	SYMBOL	CONDITIONS	MIN	TYP	MAX	UNITS	
Driver Input to Output	t _{PLH}	Figures 6 and 8, R _{DIFF} = 54Ω C _{L1} = C _{L2} = 100pF	10	30	60	ns	
	t _{PHL}		10	30	60		
Driver Output Skew to Output	t _{SKEW}	Figures 6 and 8, R _{OUT} = 54Ω, C _{L1} = C _{L2} = 100pF		5	10	ns	
Driver Rise or Fall Time	tr, tf	Figures 6 and 8, R _{DIFF} = 54Ω, C _{L1} = C _{L2} = 100pF	MAX481, MAX485, MAX1487	3	15	40	ns
			MAX490C/E, MAX491C/E	5	15	25	
			MAX490M, MAX491M	3	15	40	
Driver Enable to Output High	t _{ZH}	Figures 7 and 9, C _L = 100pF, S ₂ closed		40	70	ns	
Driver Enable to Output Low	t _{ZL}	Figures 7 and 9, C _L = 100pF, S ₁ closed		40	70	ns	
Driver Disable Time from Low	t _{LZ}	Figures 7 and 9, C _L = 15pF, S ₁ closed		40	70	ns	
Driver Disable Time from High	t _{HZ}	Figures 7 and 9, C _L = 15pF, S ₂ closed		40	70	ns	
Receiver Input to Output	t _{PLH} , t _{PHL}	Figures 6 and 10, R _{DIFF} = 54Ω, C _{L1} = C _{L2} = 100pF	MAX481, MAX485, MAX1487	20	90	200	ns
			MAX490C/E, MAX491C/E	20	90	150	
			MAX490M, MAX491M	20	90	200	
t _{PLH} - t _{PHL} Differential Receiver Skew	t _{SKD}	Figures 6 and 10, R _{DIFF} = 54Ω, C _{L1} = C _{L2} = 100pF		13		ns	
Receiver Enable to Output Low	t _{ZL}	Figures 5 and 11, C _{RL} = 15pF, S ₁ closed		20	50	ns	
Receiver Enable to Output High	t _{ZH}	Figures 5 and 11, C _{RL} = 15pF, S ₂ closed		20	50	ns	
Receiver Disable Time from Low	t _{LZ}	Figures 5 and 11, C _{RL} = 15pF, S ₁ closed		20	50	ns	
Receiver Disable Time from High	t _{HZ}	Figures 5 and 11, C _{RL} = 15pF, S ₂ closed		20	50	ns	
Maximum Data Rate	f _{MAX}		2.5			Mbps	
Time to Shutdown	t _{SHDN}	MAX481 (Note 5)	50	200	600	ns	

MAX481/MAX483/MAX485/MAX487-MAX491/MAX1487

Low-Power, Slew-Rate-Limited RS-485/RS-422 Transceivers

MAX481/MAX483/MAX485/MAX487-MAX491/MAX1487

SWITCHING CHARACTERISTICS—MAX481/MAX485, MAX490/MAX491, MAX1487 (continued)

(V_{CC} = 5V ±5% T_A = T_{MIN} to T_{MAX}, unless otherwise noted) (Notes 1 2)

PARAMETER	SYMBOL	CONDITIONS	MIN	TYP	MAX	UNITS
Driver Enable from Shutdown to Output High (MAX481)	t _{ZH(SHDN)}	Figures 7 and 9 C _L = 100pF S2 closed		40	100	ns
Driver Enable from Shutdown to Output Low (MAX481)	t _{ZL(SHDN)}	Figures 7 and 9 C _L = 100pF S1 closed		40	100	ns
Receiver Enable from Shutdown to Output High (MAX481)	t _{ZH(SHDN)}	Figures 5 and 11, C _L = 15pF, S2 closed, A - B = 2V		300	1000	ns
Receiver Enable from Shutdown to Output Low (MAX481)	t _{ZL(SHDN)}	Figures 5 and 11, C _L = 15pF, S1 closed, A - B = 2V		300	1000	ns

SWITCHING CHARACTERISTICS—MAX483, MAX487/MAX488/MAX489

(V_{CC} = 5V ±5% T_A = T_{MIN} to T_{MAX}, unless otherwise noted) (Notes 1 2)

PARAMETER	SYMBOL	CONDITIONS	MIN	TYP	MAX	UNITS
Driver Input to Output	t _{PLH} t _{PHL}	Figures 6 and 8, R _{DI1} = 54Ω C _{L1} = C _{L2} = 100pF	250	800	2000	ns
Driver Output Skew to Output	t _{SKLV}	Figures 6 and 8, R _{DI1} = 54Ω C _{L1} = C _{L2} = 100pF		100	800	ns
Driver Rise or Fall Time	t _r t _f	Figures 6 and 8, R _{DI1} = 54Ω C _{L1} = C _{L2} = 100pF	250		2000	ns
Driver Enable to Output High	t _{ZH}	Figures 7 and 9 C _L = 100pF S2 closed	250		2000	ns
Driver Enable to Output Low	t _{ZL}	Figures 7 and 9, C _L = 100pF S1 closed	250		2000	ns
Driver Disable Time from Low	t _{LL}	Figures 7 and 9, C _L = 15pF S1 closed	300		3000	ns
Driver Disable Time from High	t _{HL}	Figures 7 and 9 C _L = 15pF S2 closed	300		3000	ns
Receiver Input to Output	t _{PLH} t _{PHL}	Figures 6 and 10 R _{DI1} = 54Ω C _{L1} = C _{L2} = 100pF	250		2000	ns
t _{PHL} - t _{PLH} Differential Receiver Skew	t _{RKD}	Figures 6 and 10 R _{DI1} = 54Ω C _{L1} = C _{L2} = 100pF		100		ns
Receiver Enable to Output Low	t _{ZL}	Figures 5 and 11, C _{RI} = 15pF S1 closed	20	50		ns
Receiver Enable to Output High	t _{ZH}	Figures 5 and 11 C _{RI} = 15pF S2 closed	20	50		ns
Receiver Disable Time from Low	t _{LL}	Figures 5 and 11, C _{RI} = 15pF S1 closed	20	50		ns
Receiver Disable Time from High	t _{HL}	Figures 5 and 11 C _{RI} = 15pF S2 closed	20	50		ns
Maximum Data Rate	f _{MAX}	t _{PHL} t _{PLH} < 50% of data period	250			kbp/s
Time to Shutdown	t _{SHDN}	MAX483/MAX487 (Note 5)	50	200	600	ns
Driver Enable from Shutdown to Output High	t _{ZH(SHDN)}	MAX483/MAX487 Figures 7 and 9 C _L = 100pF, S2 closed			2000	ns
Driver Enable from Shutdown to Output Low	t _{ZL(SHDN)}	MAX483/MAX487 Figures 7 and 9 C _L = 100pF, S1 closed			2000	ns
Receiver Enable from Shutdown to Output High	t _{ZH(SHDN)}	MAX483/MAX487 Figures 5 and 11 C _L = 15pF S2 closed			2500	ns
Receiver Enable from Shutdown to Output Low	t _{ZL(SHDN)}	MAX483/MAX487 Figures 5 and 11 C _L = 15pF S1 closed			2500	ns

Low-Power, Slew-Rate-Limited RS-485/RS-422 Transceivers

Pin Description

PIN					NAME	FUNCTION
MAX481/MAX483/ MAX485/MAX487/ MAX1487		MAX488/ MAX490		MAX489/ MAX491		
DIP/SO	μMAX	DIP/SO	μMAX	DIP/SO		
1	3	2	4	2	RO	Receiver Output. If A > B by 200mV, RO will be high. If A < B by 200mV, RO will be low.
2	4	—	—	3	RE	Receiver Output Enable. RO is enabled when RE is low. RO is high impedance when RE is high.
3	5	—	—	4	DE	Driver Output Enable. The driver outputs, Y and Z, are enabled by bringing DE high. They are high impedance when DE is low. If the driver outputs are enabled, the parts function as line drivers. While they are high impedance, they function as line receivers if RE is low.
4	6	3	5	5	DI	Driver Input. A low on DI forces output Y low and output Z high. Similarly, a high on DI forces output Y high and output Z low.
5	7	4	6	6, 7	GND	Ground
—	—	5	7	9	Y	Noninverting Driver Output
—	—	6	8	10	Z	Inverting Driver Output
6	8	—	—	—	A	Noninverting Receiver Input and Noninverting Driver Output
—	—	8	2	12	A	Noninverting Receiver Input
7	1	—	—	—	B	Inverting Receiver Input and Inverting Driver Output
—	—	7	1	11	B	Inverting Receiver Input
8	2	1	3	14	VCC	Positive Supply. 4.75V ≤ VCC ≤ 5.25V
—	—	—	—	1, 8, 13	N.C.	No Connect—not internally connected

MAX481/MAX483/MAX485/MAX487-MAX491/MAX1487

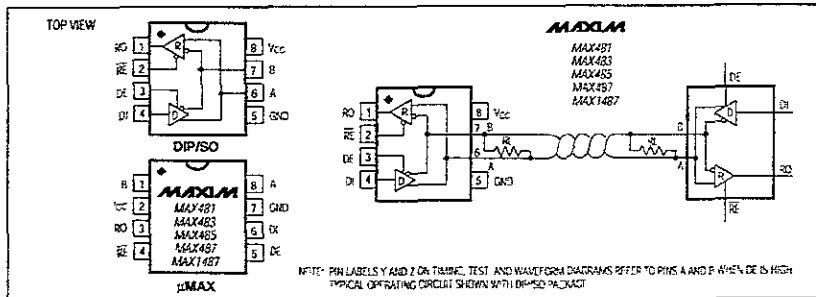


Figure 1. MAX481/MAX483/MAX485/MAX487/MAX1487 Pin Configuration and Typical Operating Circuit

Low-Power, Slew-Rate-Limited RS-485/RS-422 Transceivers

MAX481/MAX483/MAX485/MAX487-MAX491/MAX1487

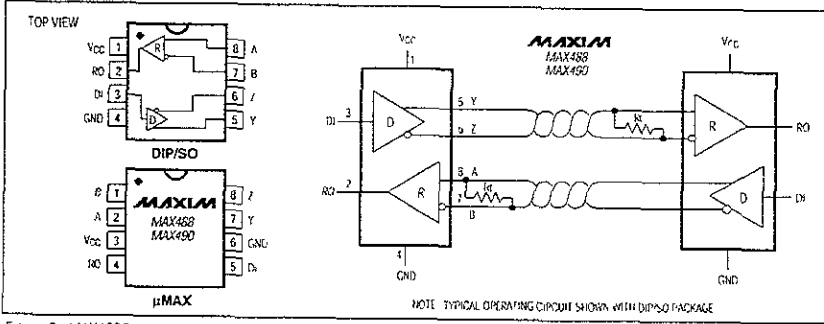


Figure 2 MAX488/MAX490 Pin Configuration and Typical Operating Circuit

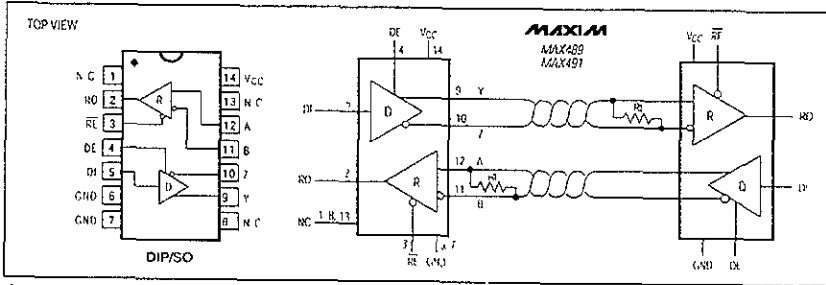


Figure 3 MAX489/MAX491 Pin Configuration and Typical Operating Circuit

Applications Information

The MAX481/MAX483/MAX485/MAX487-MAX491 and MAX1487 are low power transceivers for RS-485 and RS-422 communications. The MAX481, MAX485, MAX490, MAX491, and MAX1487 can transmit and receive at data rates up to 2.5Mbps while the MAX483, MAX487, MAX488, and MAX489 are specified for data rates up to 250kbps. The MAX488-MAX491 are full duplex transceivers while the MAX481, MAX483, MAX485, MAX487, and MAX1487 are half duplex. In addition, Driver Enable (DE) and Receiver Enable (RE) pins are included on the MAX481, MAX483, MAX485, MAX487, MAX489, MAX491, and MAX1487. When disabled, the driver and receiver outputs are high impedance.

MAX487/MAX1487: 128 Transceivers on the Bus

The 48kΩ, 1/2-unit load receiver input impedance of the MAX487 and MAX1487 allows up to 128 transceivers on a bus, compared to the 1-unit load (12kΩ input impedance) of standard RS-485 drivers (32 transceivers maximum). Any combination of MAX487/MAX1487 and other RS-485 transceivers with a total of 12 unit loads or less can be put on the bus. The MAX481/MAX483/MAX485 and MAX488-MAX491 have standard 12kΩ Receiver Input Impedance.

Low-Power, Slew-Rate-Limited RS-485/RS-422 Transceivers

Test Circuits

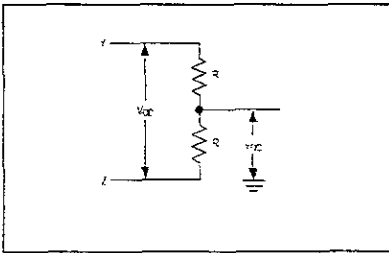


Figure 4 Driver DC Test Load

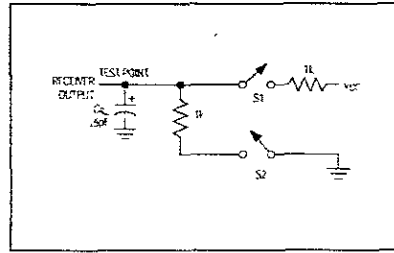


Figure 5 Receiver Timing Test Load

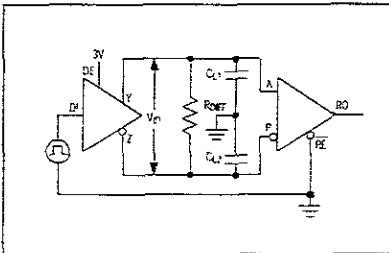


Figure 6 Driver/Receiver Timing Test Circuit

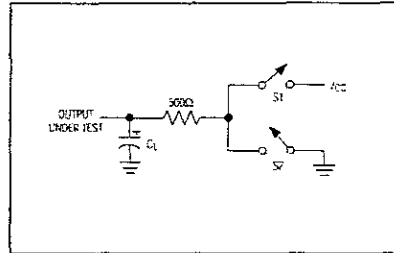


Figure 7 Driver Timing Test Load

MAX483/MAX487/MAX488/MAX489: Reduced EMI and Reflections

The MAX483 and MAX487-MAX489 are slew-rate limited, minimizing EMI and reducing reflections caused by improperly terminated cables. Figure 12 shows the driver output waveform and its Fourier analysis of a 150kHz signal transmitted by a MAX481, MAX485, MAX490, MAX491 or MAX1487. High-frequency har-

monics with large amplitudes are evident. Figure 13 shows the same information displayed for a MAX483, MAX487, MAX488, or MAX489 transmitting under the same conditions. Figure 13's high-frequency harmonics have much lower amplitudes, and the potential for EMI is significantly reduced.

MAX481/MAX483/MAX485/MAX487-MAX491/MAX1487

Low-Power, Slew-Rate-Limited RS-485/RS-422 Transceivers

MAX481/MAX483/MAX485/MAX487-MAX491/MAX1487

Switching Waveforms

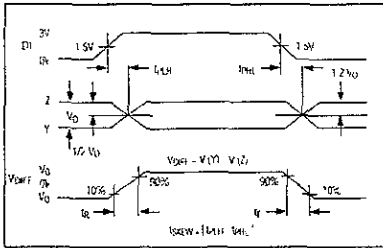


Figure 8 Driver Propagation Delays

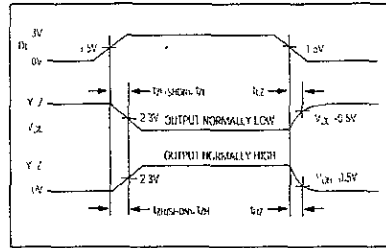


Figure 9 Driver Enable and Disable Times (except MAX488 and MAX490)

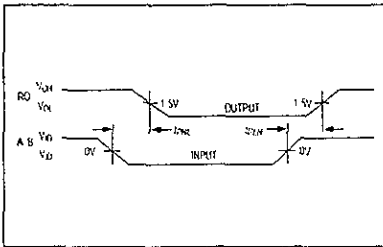


Figure 10 Receiver Propagation Delays

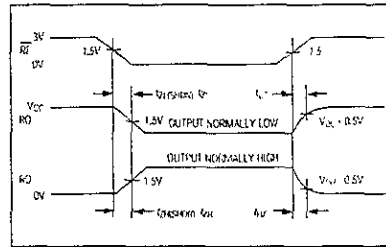


Figure 11 Receiver Enable and Disable Times (except MAX488 and MAX490)

Function Tables (MAX481/MAX483/MAX485/MAX487/MAX1487)

Table 1. Transmitting

INPUTS			OUTPUTS	
YB	DE	DI	Z	Y
X	1	1	0	1
X	1	0	1	0
0	0	X	High Z	High Z
1	0	X	High Z*	High Z*

X = Don't care
High Z = High impedance
* Shutdown mode for MAX481/MAX483/MAX485/MAX487

Table 2. Receiving

INPUTS			OUTPUT
RE	DE	A-B	RO
0	0	$\geq +0.7V$	1
0	0	$\leq -0.7V$	0
0	0	Input open	1
1	0	X	High Z*

X = Don't care
High Z = High impedance
* Shutdown mode for MAX481/MAX483/MAX485

Low-Power, Slew-Rate-Limited RS-485/RS-422 Transceivers

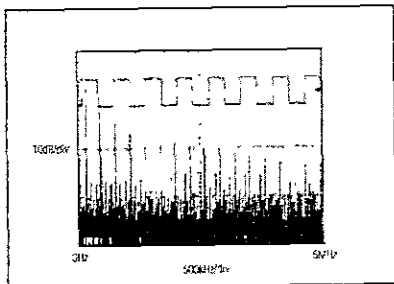


Figure 12 Driver Output Waveform and FFT Plot of MAX481/MAX485/MAX490/MAX491/MAX1487 Transmitting a 150kHz Signal

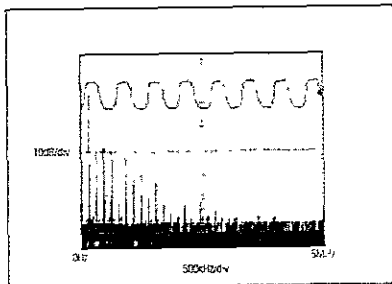


Figure 13 Driver Output Waveform and FFT Plot of MAX483/MAX487-MAX489 Transmitting a 150kHz Signal

Low-Power Shutdown Mode (MAX481/MAX493/MAX487)

A low-power shutdown mode is initiated by bringing both RE high and DE low. The devices will not shut down unless both the driver and receiver are disabled. In shutdown, the devices typically draw only 0.1µA of supply current.

RE and DE may be driven simultaneously; the parts are guaranteed not to enter shutdown if RE is high and DE is low for less than 50ns. If the inputs are in this state for at least 600ns, the parts are guaranteed to enter shutdown.

For the MAX481, MAX483, and MAX487, the tZH and tZL enable times assume the part was not in the low-power shutdown state (the MAX485/MAX488-MAX491 and MAX1487 can not be shut down). The tZH(SHDN) and tZL(SHDN) enable times assume the parts were shut down (see Electrical Characteristics).

It takes the drivers and receivers longer to become enabled from the low-power shutdown state (tZH(SHDN), tZL(SHDN)) than from the operating mode (tZH, tZL). (The parts are in operating mode if the RE, DE inputs equal a logical 0 1 or 1, 1 or 0, 0.)

Driver Output Protection

Excessive output current and power dissipation caused by faults or by bus contention are prevented by two mechanisms. A foldback current limit on the output stage provides immediate protection against short circuits over the whole common-mode voltage range (see Typical Operating Characteristics). In addition, a thermal shutdown circuit forces the driver outputs into a high-impedance state if the die temperature rises excessively.

Propagation Delay

Many digital encoding schemes depend on the difference between the driver and receiver propagation delay times. Typical propagation delays are shown in Figures 15-18 using Figure 14's test circuit.

The difference in receiver delay times $|t_{PLH} - t_{PHL}|$ is typically under 13ns for the MAX481, MAX485, MAX490, MAX491, and MAX1487 and is typically less than 100ns for the MAX483 and MAX487-MAX489.

The driver skew times are typically 5ns (10ns max) for the MAX481, MAX485, MAX490, MAX491, and MAX1487 and are typically 100ns (800ns max) for the MAX483 and MAX487-MAX489.

MAX481/MAX483/MAX485/MAX487-MAX491/MAX1487

Low-Power, Slew-Rate-Limited RS-485/RS-422 Transceivers

MAX481/MAX483/MAX485/MAX487-MAX491/MAX1487

Switching Waveforms

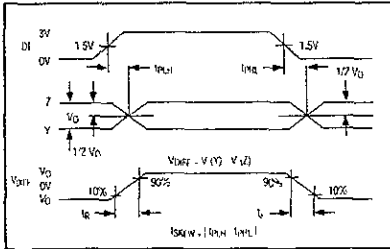


Figure 8 Driver Propagation Delays

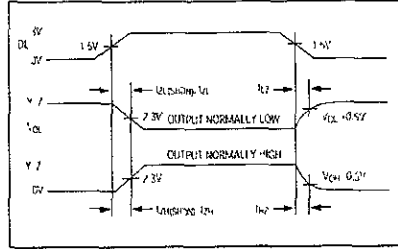


Figure 9 Driver Enable and Disable Times (except MAX488 and MAX490)

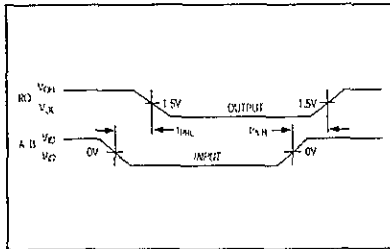


Figure 10 Receiver Propagation Delays

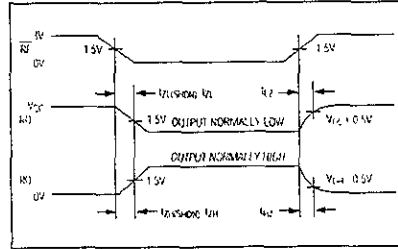


Figure 11 Receiver Enable and Disable Times (except MAX488 and MAX490)

Function Tables (MAX481/MAX483/MAX485/MAX487/MAX1487)

Table 1. Transmitting

INPUTS			OUTPUTS	
RE	DE	DI	Z	Y
X	1	1	0	1
X	1	0	1	0
0	0	X	High Z	High Z
1	0	X	High Z*	High Z*

X = Don't care
High Z = High impedance
* 5V input mode for MAX481/MAX483/MAX487

Table 2. Receiving

INPUTS			OUTPUT
RE	DE	A, B	RO
0	0	$\geq 0.7V$	1
1	0	$\leq 0.7V$	0
0	0	Input open	1
1	0	X	High Z*

X = Don't care
High Z = High impedance
* 5V input mode for MAX481/MAX483/MAX487

Low-Power, Slew-Rate-Limited RS-485/RS-422 Transceivers

Line Length vs. Data Rate

The RS-485/RS-422 standard covers line lengths up to 4000 feet. For line lengths greater than 4000 feet, see Figure 23.

Figures 19 and 20 show the system differential voltage for the parts driving 4000 feet of 26AWG twisted-pair wire at 110kHz into 120Ω loads.

Typical Applications

The MAX481, MAX483, MAX485, MAX487, MAX491, and MAX1487 transceivers are designed for bidirectional data communications on multipoint bus transmission lines.

Figures 21 and 22 show typical network applications circuits. These parts can also be used as line repeaters, with cable lengths longer than 4000 feet as shown in Figure 23.

To minimize reflections, the line should be terminated at both ends in its characteristic impedance, and stub lengths off the main line should be kept as short as possible. The slew-rate-limited MAX483 and MAX487-MAX489 are more tolerant of imperfect termination.

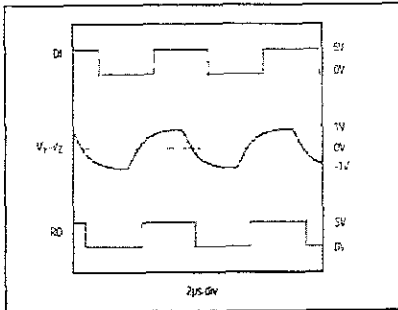


Figure 19. MAX481/MAX485/MAX491/MAX487 System Differential Voltage at 110kHz Driving 4000ft of Cable

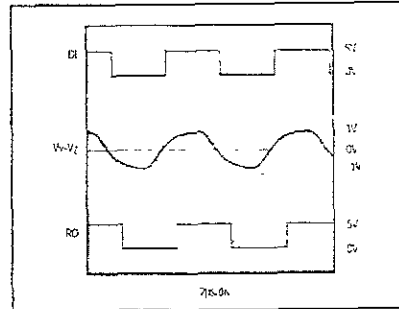


Figure 20. MAX483/MAX487-MAX489 System Differential Voltage at 110kHz Driving 4000ft of Cable

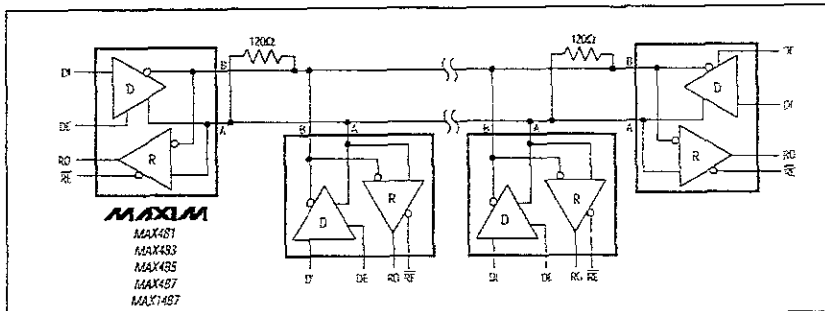


Figure 21. MAX481/MAX483/MAX485/MAX487/MAX1487 Typical Half-Duplex RS-485 Network

MAX481/MAX483/MAX485/MAX487-MAX491/MAX1487

Low-Power, Slew-Rate-Limited RS-485/RS-422 Transceivers

MAX481/MAX483/MAX485/MAX487-MAX491/MAX1487

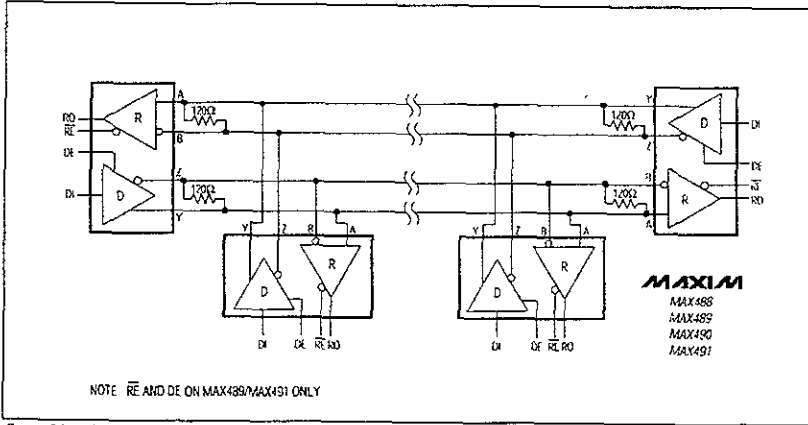


Figure 22 MAX488-MAX491 Full-Duplex RS-485 Network

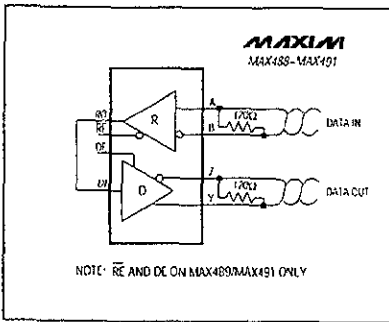


Figure 23 Line Repeater for MAX488-MAX491

Isolated RS-485
For isolated RS-485 applications see the MAX253 and MAX1480 data sheets

A ELECTRICAL CHARACTERISTICS

Table A-1 Maximum Rating

Rating	Symbol	Value	Unit
Supply Voltage	V_{DD}	-0.3 to +7.0	V
Input Voltage	V_{in}	-0.3 to +7.0	V
Operating Temperature Range MC68HC11A8 MC68HC11A8C MC68HC11A8V MC68HC11A8M MC68L11A8	T_A	T_L to T_H 0 to 70 -40 to 85 -40 to 105 -40 to 125 -20 to 70	°C
Storage Temperature Range	T_{stg}	-55 to 150	°C
Current Drain per Pin* Excluding V_{DD} , V_{SS} , V_{RH} , and V_{RL}	I_D	25	mA

*One pin at a time, observing maximum power dissipation limits.

Internal circuitry protects the inputs against damage caused by high static voltages or electric fields; however, normal precautions are necessary to avoid application of any voltage higher than maximum-rated voltages to this high-impedance circuit. Extended operation at the maximum ratings can adversely affect device reliability. Tying unused inputs to an appropriate logic voltage level (either GND or V_{DD}) enhances reliability of operation.

Table A-2 Thermal Characteristics

Characteristic	Symbol	Value	Unit
Average Junction Temperature	T_J	$T_A + (P_D \times \theta_{JA})$	°C
Ambient Temperature	T_A	User-determined	°C
Package Thermal Resistance (Junction-to-Ambient) 52-Pin Plastic Quad Pack (PLCC) 48-Pin Plastic Dual In-Line Package (DIP)	θ_{JA}	50 40	°C/W
Total Power Dissipation	P_D	$P_{INT} + P_{I/O}$ $K - (T_J + 273^\circ\text{C})$ (Note 1)	W
Device Internal Power Dissipation	P_{INT}	$I_{DD} \times V_{DD}$	W
I/O Pin Power Dissipation	$P_{I/O}$ (Note 2)	User-determined	W
A Constant	K	$P_D \times (T_A + 273^\circ\text{C}) + \theta_{JA} \times P_{D2}$ (Note 3)	W · °C

NOTES:

1. This is an approximate value, neglecting $P_{I/O}$.
2. For most applications $P_{I/O} \ll P_{INT}$ and can be neglected.
3. K is a constant pertaining to the device. Solve for K with a known T_A and a measured P_D (at equilibrium). Use this value of K to solve for P_D and T_J iteratively for any value of T_A .

Table A-3 DC Electrical Characteristics

V_{DD} = 5.0 Vdc ± 10%, V_{SS} = 0 Vdc, T_A = T_L to T_H, unless otherwise noted

Characteristics	Symbol	Min	Max	Unit
Output Voltage (Note 1) All Outputs except XTAL	V _{OL}	—	0.1	V
I _{Load} = ± 10.0 μA All Outputs Except XTAL, RESET, and MODA	V _{OH}	V _{DD} - 0.1	—	V
Output High Voltage (Note 1) I _{Load} = - 0.8 mA, V _{DD} = 4.5 V All Outputs Except XTAL, RESET, and MODA	V _{OH}	V _{DD} - 0.8	—	V
Output Low Voltage I _{Load} = 1.6 mA All Outputs Except XTAL	V _{OL}	—	0.4	V
Input High Voltage All Inputs Except RESET RESET	V _{IH}	0.7 x V _{DD} 0.8 x V _{DD}	V _{DD} + 0.3 V _{DD} + 0.3	V
Input Low Voltage All Inputs	V _{IL}	V _{SS} - 0.3	0.2 x V _{DD}	V
I/O Ports, Three-State Leakage V _{in} = V _{IH} or V _{IL} PA7, PC0-PC7, PD0-PD5, AS/STRA, MODA/LTR, RESET	I _{OZ}	—	±10	μA
Input Leakage Current (Note 2) V _{in} = V _{DD} or V _{SS} V _{in} = V _{DD} or V _{SS} PA0-PA2, IRQ, XIRQ, MODB/V _{STBY}	I _{in}	—	±1 ±10	μA
RAM Standby Voltage Power down	V _{SB}	4.0	V _{DD}	V
RAM Standby Current Power down	I _{SB}	—	10	μA
Total Supply Current (Note 3)				
RUN:				
Single-Chip Mode	I _{DD}			mA
	dc - 2 MHz	—	15	
	3 MHz	—	27	
Expanded Multiplexed Mode	I _{DD}			mA
	dc - 2 MHz	—	27	
	3 MHz	—	35	
WAIT:				
All Peripheral Functions Shut Down	W _{IDD}			mA
Single-Chip Mode				
	dc - 2 MHz	—	6	
	3 MHz	—	15	
Expanded Multiplexed Mode				
	dc - 2 MHz	—	10	
	3 MHz	—	20	
STOP:				
No Clocks, Single-Chip Mode	S _{IDD}			μA
	dc - 2 MHz	—	50	
	3 MHz	—	150	
Input Capacitance PA0-PA2, PE0-PE7, IRQ, XIRQ, EXTAL PA7, PC0-PC7, PD0-PD5, AS/STRA, MODA/LTR, RESET	C _{in}	—	8 12	pF
Power Dissipation	P _D			mW
	2 MHz	—	85	
	Expanded Multiplexed Mode	—	150	
	Single-Chip Mode	3 MHz	150	
	Expanded Multiplexed Mode	—	195	

NOTES:

- 1 V_{OH} specification for RESET and MODA is not applicable because they are open-drain pins. V_{OH} specification not applicable to ports C and D in wired-OR mode.
- 2 Refer to A/D specification for leakage current for port E
- 3 EXTAL is driven with a square wave, and
t_{SV} = 500 ns for 2 MHz rating;
t_{SV} = 333 ns for 3 MHz rating.
V_{IL} ≤ 0.2 V
V_{IH} ≥ V_{DD} - 0.2 V
No dc loads.

Table A-3a DC Electrical Characteristics (MC68L11A8)

$V_{DD} = 3.0 \text{ Vdc to } 5.5 \text{ Vdc}$, $V_{SS} = 0 \text{ Vdc}$, $T_A = T_L \text{ to } T_H$, unless otherwise noted

Characteristics	Symbol	Min	Max	Unit
Output Voltage (Note 1)	V_{OL}	—	0.1	V
All Outputs except XTAL	V_{OH}	$V_{DD} - 0.1$	—	V
All Outputs Except XTAL, RESET, and MODA				
$I_{Load} = \pm 10.0 \mu\text{A}$				
Output High Voltage (Note 1)	V_{OH}	$V_{DD} - 0.8$	3	V
All Outputs Except XTAL, RESET, and MODA				
$I_{Load} = -0.8 \text{ mA}$, $V_{DD} = 4.5 \text{ V}$				
Output Low Voltage	V_{OL}	—	0.4	V
All Outputs Except XTAL				
$I_{Load} = 1.6 \text{ mA}$				
Input High Voltage	V_{IH}	$0.7 \times V_{DD}$ $0.8 \times V_{DD}$	$V_{DD} + 0.3$ $V_{DD} + 0.3$	V
All Inputs Except RESET				
RESET				
Input Low Voltage	V_{IL}	$V_{SS} - 0.3$	$0.2 \times V_{DD}$	V
All Inputs				
I/O Ports, Three-State Leakage	I_{OZ}	—	± 10	μA
PA7, PC0-PC7, PD0-PD5, $V_{in} = V_{IH}$ or V_{IL} AS/STRA, MODA/LIR, RESET				
Input Leakage Current (Note 2)	I_{in}	—	± 1 ± 10	μA
PA0-PA2, IRQ, XIRQ $V_{in} = V_{DD}$ or V_{SS} $V_{in} = V_{DD}$ or V_{SS} MODB/VSTBY				
RAM Standby Voltage	Power down	V_{SS}	2.0	V_{DD}
RAM Standby Current	Power down	I_{SB}	—	10
Total Supply Current (Note 3)				
RUN:	I_{DD}			mA
Single-Chip Mode	dc - 1 MHz	—	4	
	2 MHz	—	8	
Expanded Multiplexed Mode	dc - 1 MHz	—	7	
	2 MHz	—	14	
WAIT:	W_{IDD}			mA
All Peripheral Functions Shut Down				
Single-Chip Mode	dc - 1 MHz	—	3	
	2 MHz	—	6	
Expanded Multiplexed Mode	dc - 1 MHz	—	2.5	
	2 MHz	—	5	
STOP:	S_{IDD}			μA
No Clocks, Single-Chip Mode	dc - 1 MHz	—	25	
	2 MHz	—	25	
Input Capacitance	C_{in}	—	8 12	pF
PA0-PA2, PE0-PE7, IRQ, XIRQ, EXTAL PA7, PC0-PC7, PD0-PD5, AS/STRA, MODA/LIR, RESET				
Power Dissipation	P_D			mW
Single-Chip Mode	1 MHz	—	12	
Expanded Multiplexed Mode		—	21	
Single-Chip Mode	2 MHz	—	24	
Expanded Multiplexed Mode		—	42	

NOTES:

- V_{OH} specification for RESET and MODA is not applicable because they are open-drain pins. V_{OH} specification not applicable to ports C and D in wired-OR mode.
- Refer to A/D specification for leakage current for port E
- EXTAL is driven with a square wave, and
 $t_{cyc} = 1000 \text{ ns}$ for 1 MHz rating;
 $t_{cyc} = 500 \text{ ns}$ for 2 MHz rating.
 $V_{IL} \leq 0.2 \text{ V}$
 $V_{IH} \geq V_{DD} - 0.2 \text{ V}$
 No dc loads.

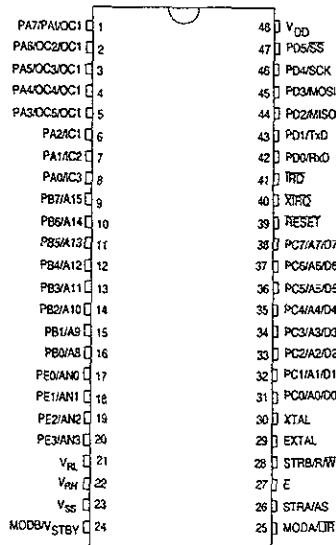
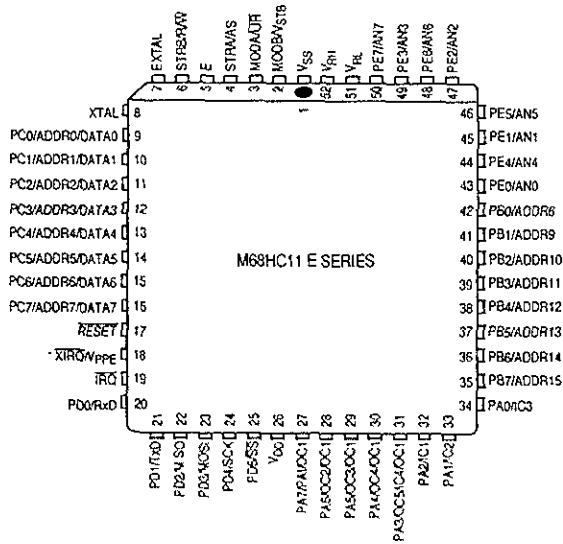


Figure B-2 48-Pin DIP



térmico automático, para prevenir daño en el dispositivo. La mayoría de los dispositivos EIA-485 usan esta técnica. El circuito de apagado, desactiva las salidas del transmisor cuando la temperatura de la junta excede 150 °C y automáticamente reactiva las salidas cuando la junta se enfría.

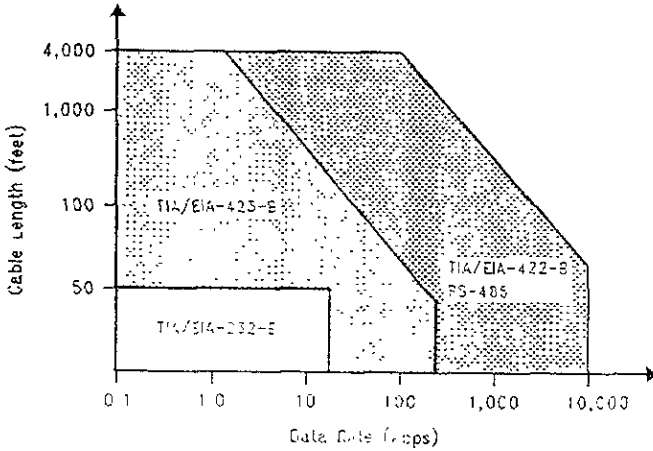


Figura A-5 Gráfica velocidad - longitud de los estándares EIA-485, EIA-422, EIA-423, EIA-232

TABLA A-1 Principales especificaciones eléctricas de EIA-485				
Parameter	Conditions	Min	Max	Units
Driver Output Voltage Loaded	$R_L = 110 \Omega$ RS-422	0		V
Driver Output Voltage Loaded	$P_L = 51 \text{ pF}$ - 115 pF $C_L = 50 \text{ pF}$	1.5		V
Driver Output Short-Circuit Current			± 250	mA
Driver Common Mode Voltage			3	V
Difference in Common Mode Offset			0.2	V
Receiver Sensitivity			± 2.0	mV
Receiver Common-Mode Voltage Range			± 15	V
Receiver Input Impedance		20		k Ω

Apéndice B Descripción de los comandos del Protocolo

Descriptores

Para pasar información de los comandos del protocolo a las ISR de transmisión y de recepción se utilizan descriptores. Un descriptor es una estructura de datos compuesta de varios campos. Estos campos le indican a la ISR cuál es la dirección destino, cuánto mide un mensaje, en dónde se encuentra los datos, etc. En el protocolo se manejan 3 tipos de descriptores : transmisión, recepción de datos y de control.

Descriptor de transmisión de datos	
Nombre del campo	# de bytes
Dirección Destino	1
Tipo de Marco	1
STATUS del mensaje	1
Longitud	1
Apuntador de datos	2

Descriptor de recepción de datos	
Nombre del campo	# de bytes
Dirección Origen	1
Tipo de Marco	1
Longitud	1
Apuntador de datos	2

Descriptor de control	
Nombre	# de bytes
Dirección Destino	1
Tipo de Marco	1
STATUS del mensaje	1

En el protocolo se manejan :

- 5 descriptores de control : 1)SC, 2)CC, 3)ACK, 4)NAK, 5)REP.
- 1 descriptor de Transmisión de datos
- 1 descriptor de Recepción de datos

El usuario del protocolo utiliza los descriptors SC y CC para leer el estado de los comandos Solicitud de Conexión y Cerrar conexión, respectivamente. Además utiliza el descriptor de recepción de mensajes para saber cuánto mide el mensaje que llegó, en qué localidad se encuentra, si el mensaje está fragmentado y la dirección del nodo transmisor.

Variable STN_CONNECTED

La función de la variable STN_CONNECTED es indicar la dirección de la estación con la cual existe una conexión. Si el valor de la variable es cero significa que no existe conexión alguna.

Comandos

STARTCON

Este comando solicita establecer una conexión con un nodo remoto de la red. El parámetro que recibe es la dirección del nodo remoto. A través del descriptor solicitud de conexión se devuelve el estado del comando.

Parámetros de Entrada	Valores devueltos por el comando
ACCA= Dirección del nodo remoto	En [DESC_SC+STATUS]= Estado del comando

DESC_SC: Etiqueta que contiene la dirección del primer byte del descriptor solicitud de conexión.

STATUS: Símbolo que contiene el offset del campo STATUS del descriptor DESC_SC.

Los estados del comando son:

Estado	Definición
0	Aun no transmitido
1	Transmitiéndose
2	Conexión rechazada
3	Otra estación se conecta antes con nosotros
4	Conexión aceptada
5	Time Out

Ejemplo de uso :

```

        LDAA    #1
        JSR    STARTCON
REV_STATUS
        LDX    #DESC_SC
        LDAA   STATUS,X
        CMPA  #CONEXION_ACEPTADA
        BEQ   ENVIADATOS
        JMP   REV_STATUS
ENVIADATOS

```

RESTRICCIONES :

- El comando SC debe de ejecutarse cuando no exista conexión.
- Cuando se establece una conexión, el dispositivo receptor de la SC, no puede empezar a transmitir, hasta que el dispositivo emisor envíe el primer mensaje de datos.

NOTAS :

Una estación aceptará establecer una conexión cuando STN_CONNECTED=0 y BLOCKCONN=0.

Al momento de recibir una SC y aceptarla, los números de secuencia de recepción y transmisión son iniciados a 0.

Si el estado del comando es "Time-out" el usuario necesita ejecutar STARTCON para volver a intentar la solicitud de conexión ; no se puede ejecutar SENDMSG_AGAIN.

TRANSMITE_DATOS

El comando TRANSMITE_DATOS permite transmitir un bloque de datos. Los parámetros son pasados a través de los registros internos del HC11 (ACCX, ACCY y ACCA). Las posibles respuestas son indicadas por medio del CARRY y del ACCX.

BIBLIOGRAFÍA

Microprocesadores y manuales

68hc11 microcontroller FAQ

<http://www.lib.ox.ac.uk/internet/news/faq/archive/microcontroller-faq.68hc11.html>

AVOCET Systems. *AVSIM11 68HC11 Simulator/Debugger User's Manual*, Revisión 1.3, 1988

AVOCET Systems. *AVMAC11 68HC11 Cross-Assembler User's Manual*, Revisión 2.0, 1987

Anderson, John A. *Foundations of computer technology*. Chapman & Hall London, 5ª edición, Inglaterra 1994

ATMEL. *Configurable Logic Data Book*.

Downton, A.C. *Computadoras y microprocesadores*. Addison-Wesley Iberoamericana, 3ª edición, Estados Unidos 1993

MOTOROLA. *M68HC11 Reference Manual*, Revisión 3, 1991

MOTOROLA. *MC68HC11/18 Programming Reference Guide*, Revisión 1.

MOTOROLA. *MC68HC11 E-Series Programming Reference Guide*, Revisión 1

MOTOROLA. *Pcbug11 : User's Manual*, 2ª edición, 1992

MOTOROLA. *FAST and LS TTL Data*. 5ª edición, 1992

HITACHI. *How to use Hitachi's Built-in Controller Driver LCD-II (HD44780) Dot Matrix LCD Module*.

Pcatman, John B. y otros *Design with Microcontrollers*, McGraw-Hill, Estados Unidos, 1988

Ramirez, Edward V. and Weiss Melvyn. *Microprocessing Fundamentals : Hardware & Software*, 1981

Sayers, Ian L. y otros. *Principios de Microprocesadores*, CECSA, México, 1991

Tocci, Ronald J. y otros *Microprocessors and Microcomputers : hardware and software*, 4ª edición, Prentice Hall, Estados Unidos, 1997

Wray, William C. y Greenfield, Joseph D. *Using microprocessors and microcomputers : the Motorola family*. 3ª edición, Prentice Hall, Estados Unidos, 1997

Comunicaciones

Aké, F. Antonio. *Interfaz Paralela y Serial de la PC*, UNAM-PIC, México, 1998

- Aké, F. Antonio. *Internet : Comandos básicos y Servicios*, UNAM-PTC, México, 1996
- Black, Uyless. *Data Link Protocols*, Prentice Hall, Estados Unidos
- Butler, Jim. *UARTs make possible low-cost networks of embedded systems*. EDN Magazine , 30 de marzo de 1995
<http://www.ednmag.com/reg/1995/033095/07df3.cfm>
- Cebrián, Antonio R. y Borraz Eduardo F. *Guía práctica de comunicaciones y redes locales*. Ediciones G. Gili. España, 1993
- DEC. *Digital Data Communications Message Protocol: DDCMP Funcional Specification Phase IV version 4.1*, 1998
- Kratzer, Ann. *A general model for implementation of communications protocols*. Dartmouth College, Estados Unidos
- Félix, José R. *Introducción a las redes locales*, 7ª edición, Ediciones Anaya Multimedia, España, 1997
- García, Jesús T. *Redes para proceso distribuido*, RA-MA Editorial, España
- Held, Gilbert. *The complete modem reference*. John Wiley. 3ª edición., Estados Unidos, 1997
- Held, Gilbert. *Understanding data communications : from fundamentals to networking*. John Wiley, 2ª edición, Estados Unidos, 1996
- Lane, Malcolm G. *Data communications software design*, Editorial Tom Walker, 1985
- Mcnamara, John E. *Technical Aspects of Data Communications*. Digital Equipment Corporation, 1988
- McCracken, David. *Software Partitioning for Multitasking Communication*. Dr. Dobb's Journal, Septiembre 1991
- Nelson, Mark. *Servicing COM Port Interrupts*, *Tech Specialist*, Mayo 1990
<http://dogma.net/markn/articles/comports/comports.htm>
- Pimentel, Juan R. *Communication Networks for Manufacturing*, Prentice Hall, Estados Unidos
- Purser, Michael. *Redes de Telecomunicaciones y Ordenadores*, Ediciones Díaz de Santos, España 1987
- Purser, Michael. *Comunicación de datos para programadores*, Addison-Wesley-Iberoamericana, México, 1989
- Quinn, Jack. *Digital Data Communications*, Prentice Hall Career and Technology, Estados Unidos, 1995
- Reiss, Leszek. *Introduction to local area networks with microcomputer*, Prentice Hall, Estados Unidos
- Raji, Reza S. *Smart Networks for control*, IEEE Spectrum, Estados Unidos Junio 1994.

BIBLIOGRAFÍA

- Sinnerna, William y McGovern Tom. *Digital Analog, and Data communication*, Prentice Hall 2ª Edición, 1986
- Spragins, John D. *Telecommunications Protocols and Design*, Prentice Hall, Estados Unidos
- Stallings, William. *Comunicaciones y redes de computadores*, 5ª edición, Prentice Hall, España, 1997
- Stevens, Richard W. Y Wright Gary R. *TCP/IP Illustrated, Volume 1: The protocols*, Addison-Wesley, Estados Unidos, 1996
- Tanenbaum, Andrew S. *Redes de computadoras*, 3ª edición, Prentice Hall México, 1997
- Uspender, Bhargav P. y Koopman, Philip J. *Communication Protocols for Embedded Systems*. Embedded systems Programming, Noviembre de 1994. <http://www.cs.cmu.edu/~koopman/protsrvy/protsrvy.html>
- Williams, Ross N. *A painless guide to CRC error detection algorithms*, ftp://ftp.adelaide.edu.au/pub/rocksoft/crc_v3.txt, 1993
- Sistemas en tiempo real**
- Allworth, S. *Introduction to real time software design*, Macmillan Press, Reino Unido, 1981
- Auslander, David M. y Tham Cheng H. *Real-Time Software Program Examples in C*, Prentice Hall, Estados Unidos
- Burns, Alan. *Real time systems and their programming languages*, Addison-Wesley, Inglaterra, 1990
- Cooling, J. E., *Software design for real time systems*. Chapman and Hall, Londres Inglaterra, 1990
- Laplante, Philip. *Real-Time Systems Design and Analysis: An Engineer's Handbook*, IEEE Computer Society Press, Estados Unidos
- Stallings, William. *Sistemas Operativos*. LIMUSA, México 1995.
- EIA-232, EIA-422, EIA-423, EIA-485**
RS-232, RS-422 and V.35 interfaces.
<http://www.sangoma.com/signal.htm>
- National Semiconductor *AN-216 Summary of Electrical Characteristics of some well known digital interface standards*
- National Semiconductors *AN-214 Transmission line drivers and receivers for RS-422 and RS-423*
- National Semiconductor *AN-1031 TL1/EIA-422-B Overview.*
- National Semiconductor *AN-979 The Practical Limits of RS-485*
- National Semiconductor *AN-759 Comparing EIA-485 and EIA-422 Line Drivers and Receivers in Multipoint Applications*

National Semiconductor *AN-1057 Ten Ways to Bulletproof RS-485 interfaces.*

B&B Electronics Manufacturing Company.
RS-422 and RS-485 Application Note.

ICS DataCom. *Serial Data Communications Backgroundler (For RS-232, RS-422 and RS-485 devices)*

DY 4 Systems. *Application of RS-423 in RS-422 and RS-232 Systems*

EIA Standard RS-232-C. Agosto 1969

EIA Standard RS-422. Abril 1975

EIA Standard RS-423. Abril 1975

MAXIM New Releases Data Book :
Volume IV. 1997