

ESCUELA POLITÉCNICA NACIONAL

ESCUELA DE INGENIERÍA

ESTUDIO DEL PROTOCOLO CAN (CONTROLLER AREA NETWORK) Y SU APLICACIÓN EN REDES DE CONTROL

**PROYECTO PREVIO A LA OBTENCIÓN DEL TÍTULO DE INGENIERO EN
ELECTRÓNICA Y TELECOMUNICACIONES**

MARIO LUIS DEFAZ ANDRANGO

DIRECTOR: ING. ERWIN BARRIGA

QUITO, Junio 2007

DECLARACIÓN

Yo, Mario Luis Defaz Andrango declaro bajo juramento que el trabajo aquí descrito es de mi autoría; que no ha sido previamente presentada para ningún grado o calificación profesional; y, que he consultado las referencias bibliográficas que se incluyen en este documento.

A través de la presente declaración cedo mis derechos de propiedad intelectual correspondientes a este trabajo, a la Escuela Politécnica Nacional, según lo establecido por la Ley de Propiedad Intelectual, por su reglamento y por la normatividad institucional vigente.

Mario Luis Defaz Andrango

CERTIFICACIÓN

Certifico que el presente trabajo fue desarrollado por Mario Luis Defaz Andrango, bajo mi supervisión.

Ing. Erwin Barriga
DIRECTOR DEL PROYECTO

AGRADECIMIENTO

A mis padres y a mis hermanos, por su comprensión y apoyo incondicional, sin los cuales alcanzar este objetivo hubiese sido una tarea mucho más complicada. Al Ing. Erwin Barriga por su acertada colaboración, y a todas las personas que de alguna manera contribuyeron a la elaboración de este trabajo.

DEDICATORIA

A mis padres, Luis y Mercedes, quienes con amor, sabiduría y sacrificio supieron brindarme todas las herramientas necesarias para mi formación humana e intelectual. No existen palabras para demostrarles mi gratitud. De todo corazón MUCHAS GRACIAS.

ÍNDICE

PRESENTACIÓN	1
RESUMEN	3
1. CAPÍTULO 1: Introducción	5
1.1. Visión general del Protocolo CAN.....	5
1.2. Redes de comunicación de datos.....	6
1.3. Dominio de campo y sistemas de bus de campo.....	6
1.4. Jerarquías de Protocolos.....	8
1.5. El Modelo de referencia OSI y los sistemas de comunicación industrial.....	10
1.6. Modelo de comunicación Cliente-Servidor y Productor-Consumidor.....	14
1.7. Protocolos orientados al nodo.....	16
1.8. Protocolos orientados al mensaje.....	16
1.9. Control de acceso al medio.....	17
1.9.1. Maestro-esclavo.....	19
1.9.2. Token delegado.....	20
1.9.3. Token passing.....	21
1.9.4. Acceso múltiple por división de tiempo (TDMA).....	22
1.9.5. Acceso múltiple por división de portadora (CSMA).....	22
1.9.5.1. CSMA/CD (CSMA con detección de colisiones).....	23
1.9.5.2. CSMA/CA (CSMA con evasión de colisiones).....	24
1.10. Detección y control de errores.....	25
1.10.1. Control pasivo de errores.....	26
1.10.2. Señalización de errores.....	27
1.11. Topologías de red.....	28
1.11.1. Topología en estrella.....	29
1.11.2. Topología en bus.....	30
1.11.3. Topología en árbol.....	30
1.11.4. Topología en anillo.....	31
1.12. Principales características de CAN.....	31
2. CAPÍTULO 2: Protocolo CAN	36
2.1. Capa Enlace.....	36
2.1.1. Principio de Arbitraje del Bus.....	36
2.1.2. Formatos de Trama.....	40
2.1.2.1. Trama de datos.....	41
2.1.2.2. Trama remota.....	46
2.1.2.3. Trama de error.....	47

2.1.2.4.	Trama de sobrecarga.	52
2.1.2.5.	Espacio ínter-tramas.	53
2.1.2.6.	Validación de tramas.	55
2.1.2.7.	Codificación del flujo de bits.	56
2.1.3.	Detección y manejo de errores.	60
2.1.3.1.	Mecanismos de detección.	60
2.1.3.2.	Capacidad de detección de errores.	60
2.1.3.3.	Consistencia de datos.	62
2.1.3.4.	Tiempo de recuperación de errores.	62
2.1.4.	Confinamiento de fallos.	63
2.1.5.	Formato extendido de trama.	69
2.1.6.	Comunicación Time Triggered.	71
2.2.	Capa Física.	72
2.2.1.	Señalización física.	72
2.2.1.1.	Representación de bit.	72
2.2.1.2.	Temporización y sincronización de bit.	74
2.2.1.3.	Dimensionamiento de los segmentos de temporización de bit.	80
2.2.1.4.	Tolerancia de oscilador.	85
2.2.1.5.	Relación entre la velocidad de datos y longitud del bus.	87
2.2.2.	Medios de transmisión.	91
2.2.2.1.	Medio de transmisión eléctrico.	91
2.2.2.2.	Medios de transmisión ópticos.	98
2.2.3.	Topología de red.	100
2.2.4.	Acceso al bus.	106
2.2.5.	Estándares de capa física.	109
2.2.5.1.	Estándar ISO 11898-2.	110
2.2.5.2.	Estándar CiA DS-102.	113
2.2.5.3.	Tolerancia a fallos de la interfaz del bus de acuerdo a ISO 11898-3.	114
2.2.5.4.	Estándar SAE J2411.	117
2.2.5.5.	Estándar ISO 11992.	119
2.2.6.	Controlador del protocolo CAN.	120
2.2.6.1.	Funciones de un controlador CAN.	121
3.	CAPÍTULO 3: PROTOCOLOS DE CAPAS SUPERIORES.	123
3.1.	CAL-Capa Aplicación CAN.	124
3.1.1.	Arquitectura del Protocolo.	125
3.1.2.	Especificación de mensajes CAN (CMS).	128
3.1.3.	Asignación de identificadores de mensajes.	130
3.1.4.	Administración de red.	131
3.1.5.	Administración de capas.	131
3.2.	CANopen.	132
3.2.1.	Modelo de comunicación CANOpen.	133
3.2.1.1.	Objeto de Proceso de Datos (PDO).	134
3.2.1.2.	Objeto de Servicio de Datos (SDO).	137
3.2.2.	Administración de red (NMT).	137

3.2.2.1.	Control de nodo.	139
3.2.2.2.	Monitoreo de nodo.	140
3.2.3.	Servicios de sistema.	141
3.2.3.1.	Operación Sincrónica.	141
3.2.3.2.	Mensajes de Emergencia.	141
3.2.3.3.	Objeto de etiqueta de tiempo.	142
3.2.4.	Diccionario de objetos.	142
3.2.5.	Perfil de comunicación.	142
3.2.6.	Perfiles de dispositivo.	143
3.2.7.	Asignación de identificadores.	143
3.2.8.	Campo de aplicación.	144
3.3.	DeviceNet.	146
3.3.1.	Modelo de objetos.	148
3.3.1.1.	Objetos de comunicación.	150
3.3.1.2.	Objetos de sistema.	150
3.3.1.3.	Objetos de aplicación específica.	151
3.3.2.	Identificador CAN.	151
3.3.3.	Modelo de comunicación.	152
3.3.3.1.	Establecimiento de la conexión.	153
3.3.3.2.	Mensajes explícitos.	153
3.3.3.3.	Mensajes I/O.	154
3.3.3.4.	Protocolo de fragmentación.	156
3.3.4.	Administración de red.	156
3.3.4.1.	Chequeo de MAC ID.	156
3.3.4.2.	Mensaje Heartbeat.	156
3.3.4.3.	Mensaje Shutdown.	156
3.3.5.	Perfiles de Dispositivos.	156
3.3.6.	Campo de aplicación.	157
3.4.	SAE J1939.	158
3.4.1.	Estructura del mensaje.	160
3.4.1.1.	Prioridad.	161
3.4.1.2.	Número de Grupo de Parámetro (PGN).	161
3.4.1.3.	Campo de dirección origen.	163
3.4.1.4.	Campo de datos.	163
3.4.2.	Tipos de mensajes.	163
3.4.2.1.	Comandos.	163
3.4.2.2.	Solicitudes.	163
3.4.2.3.	Mensajes de respuesta.	164
3.4.2.4.	mensajes de confirmación.	164
3.4.3.	Funciones de grupo.	164
3.4.3.1.	Mensajes propietarios.	164
3.4.3.2.	Transmisión fragmentada.	165
3.4.4.	Administración de red.	166
3.4.4.1.	Clases de configuración de dispositivos.	167
3.4.4.2.	Configuración dinámica de dirección de nodo.	168
3.4.5.	Capa Aplicación.	169
3.4.6.	Campo de Aplicación.	170
3.5.	Time-Triggered CAN.	172
3.5.1.	Principio de operación Time Triggered CAN.	173

3.5.1.1.	Ventanas de tiempo, ciclos base y matriz.	173
3.5.1.2.	Mensajes de referencia.	175
3.5.2.	Esquema de transmisión definida (Scheduling).	176
3.5.3.	Tolerancia a fallos y determinación del maestro de tiempo activo.	177
3.5.4.	Sincronización local y global.	178
3.5.5.	Campo de aplicación.	179
4.	CAPÍTULO 4: APLICACIONES.	180
4.1.	Aplicaciones y características de sistemas CAN.	180
4.1.1.	CAN en automatización de la Producción.	180
4.1.2.	CAN en sistemas móviles.	181
4.1.3.	CAN en automatización de edificios.	183
4.1.4.	CAN en sistemas embebidos.	184
4.1.5.	Ejemplos de aplicación de sistemas CAN.	185
4.1.5.1.	Campo automotriz.	185
4.1.5.2.	Electrónica marítima.	187
4.1.5.3.	Electrónica aeroespacial.	187
4.1.5.4.	Sistemas de automatización.	187
4.1.5.5.	Sistemas embebidos.	188
4.2.	Aspectos de diseño e implementación de sistemas CAN.	189
4.2.1.	Alternativas de implementación.	189
4.2.1.1.	Implementación basada en servicios de capa 2.	190
4.2.1.2.	Implementación basada en protocolos estandarizados de capa aplicación.	191
4.2.1.3.	Implementación basada en un perfil estandarizado.	191
4.2.2.	Diseño de redes de control distribuido basadas CAN.	194
4.2.2.1.	Mapeo de mensajes de aplicación específica en grupos de mensajes.	195
4.2.2.2.	Asignación de grupos de mensajes a mensajes CAN.	196
4.2.2.3.	Determinación de la velocidad de transmisión del sistema.	198
4.2.3.	Comportamiento de tiempo real.	198
4.2.4.	Implementación de nodos CAN.	200
4.2.4.1.	Tipos de nodo.	200
4.2.4.2.	Arquitectura del controlador CAN.	203
4.2.5.	Ejemplo de implementación de una red de control distribuido I/O.	206
5.	CAPÍTULO 5: CONCLUSIONES Y RECOMENDACIONES.	215

BIBLIOGRAFÍA

ANEXOS

PRESENTACIÓN

El desarrollo del control distribuido en la industria va paralelo al de las comunicaciones. Cada vez es más necesario disponer de dispositivos inteligentes para realizar tareas de control o supervisión remota, tanto de procesos de fabricación, como de funciones de almacenamiento o distribución. Esta necesidad a su vez requiere de sistemas robustos de comunicación que cumplan satisfactoriamente con los requerimientos de dichas necesidades. El Protocolo de comunicación CAN (Controller Area Network) constituye una fuerte herramienta al respecto.

El Protocolo CAN fue establecido por BOSCH como una red de trabajo para control industrial. CAN constituye un sistema electrónico sofisticado especialmente conveniente para tareas de networking de dispositivos “inteligentes” así como de sensores y actuadores dentro de un sistema o subsistema.

Gracias a su alto nivel de confiabilidad, eficiencia en costo y rapidez en la transmisión de datos, las compañías automotrices vieron en CAN un protocolo robusto capaz de implementar un sistema de comunicación óptimo a bordo de sus vehículos. Los principales fabricantes de autos (FORD, GM, MAZDA y SAAB) empezaron a implementar CAN en el año 2003 y continuarán ampliándolo a todos sus vehículos, para una aplicación total hasta el año 2008, año a partir del cual será obligatorio para todos los autos. Actualmente, sus ventas anuales exceden los 400 millones de unidades de nodos, convirtiéndose así en la red vehicular más extensamente usada.

CAN también se ha difundido en otras áreas de aplicación, por ejemplo control de plantas industriales, aplicaciones domésticas, control de ascensores, control de sistemas de navegación, sistemas embebidos, etc.

Grupos y asociaciones de investigación como CiA y Bosch continuamente organizan eventos para dar a conocer nuevos productos y desarrollos basados en este protocolo.

A partir de CAN se han desarrollado protocolos de capa superior: CAL, CANOpen, DeviceNet, SAE J1939 y TTCAN. Los cuales mejoran aún más las características entregadas por CAN al permitir interoperabilidad e ínter cambiabilidad de los dispositivos utilizados en las redes de comunicación de sus respectivas áreas de aplicación.

En definitiva el protocolo CAN constituye un poderoso sistema de comunicación, que gracias a todas sus ventajas, desde sus inicios hasta hoy en día se ha desplegado en un número interminable de campos de aplicación, además constituye la base para el desarrollo de sistemas más sofisticados.

El presente proyecto de titulación tiene por objeto brindar un estudio detallado del Protocolo CAN, que permita entender su estructura, funcionamiento y aplicaciones. Adicionalmente se abordarán aspectos de diseño e implementación de redes de control CAN. También se llevará a cabo un breve estudio de la estructura, tipos de mensajes y aplicaciones de los protocolos de capa o nivel superior, que permita comprender su funcionamiento. Desarrollando así un trabajo que brinde un entendimiento satisfactorio del Protocolo CAN en la realización de tareas de control y que sirva de fuente de consulta para llevar a cabo posteriores estudios complementarios o implementaciones basadas en lo aquí expuesto.

RESUMEN GENERAL

Para cumplir con los objetivos planteados, el presente proyecto de titulación está compuesto por 5 capítulos. A continuación se presenta un resumen de cada uno de ellos.

El Capítulo 1 introduce los principales términos y conceptos relacionados a la comunicación de datos, y cómo son empleados en el desarrollo del protocolo de comunicaciones CAN. A continuación se expone cómo está estructurado CAN tomando como punto de partida al modelo de referencia OSI. También se explican aquellas características generales que definen a los sistemas de comunicación, tales como: tipos de intercambio de datos, métodos de control de acceso al medio, técnicas de detección y de control de errores, además de las topologías físicas de red mayormente empleadas. A partir de esta introducción finalmente se presenta cuáles de estos conceptos son aplicados al protocolo CAN, definiendo sus principales características como protocolo de comunicación.

La primera parte del capítulo 2 aborda temas referentes a la capa enlace CAN tales como: principio de acceso al medio, formatos de tramas, métodos de detección y confinamiento de errores. La segunda parte se encarga del estudio de las características de la capa física CAN, que tiene que ver con temas tales como: representación, temporización y sincronización del bit, así como la relación entre la temporización de bit, posible longitud de línea y tolerancia del oscilador. Esta sección también describe las alternativas más difundidas para el bus como medio físico y sus posibles topologías de red, además de sus principales estándares de capa física. Finalmente la tercera parte describe las principales características del controlador CAN.

El Capítulo 3 provee una introducción a los más importantes protocolos de capa superior basados en CAN: CAL, CANOpen, DeviceNet y SAE J1939. Citando las características más importantes de su estructura: tipos de mensajes, campo de

aplicación, etc. Debido a su importancia en sistemas que requieren de un comportamiento en tiempo real, adicionalmente se presenta un breve estudio del protocolo de capa superior TTCAN.

El capítulo 4 está dividido en dos partes, en la primera se mencionan los campos de aplicación de las redes CAN y sus características, también se citan ejemplos de aplicaciones específicas de estas redes en los campos mencionados. La segunda parte tiene que ver con aspectos de diseño e implementación de redes basadas en CAN. Se ilustra un proceso sistemático de diseño de redes de control CAN, además se revisan los diferentes tipos de nodos y arquitecturas de implementación de controladores CAN. Finalmente se presenta un ejemplo didáctico de diseño de éstas redes.

Finalmente, en el Capítulo 5 se exponen las conclusiones y recomendaciones más importantes del estudio realizado.

CAPITULO 1

INTRODUCCIÓN

1.1 VISIÓN GENERAL DEL PROTOCOLO CAN

En Febrero de 1986 en la Conferencia SAE¹ en Detroit, fue oficialmente presentado el protocolo CAN “Controller Area Network” (Área de Red Controlada) por BOSCH². Originalmente desarrollado para aplicaciones en automotores, éste sistema de comunicación basado en un bus serial ha empezado a ser usado exitosamente en otros campos de aplicación. CAN predomina en sistemas de control embebidos para máquinas, dispositivos médicos, aplicaciones domésticas, etc. Solo en el año 2005 fueron vendidos más de 300 millones de chips CAN, lo que refleja su nivel de penetración en la industria de la automatización.

En 1991 fue publicada la especificación CAN de BOSCH (version 2.0), seguida por el estándar oficial CAN (ISO 11898) en 1993. Ambas especificaciones contenían vacíos y temas pasados por alto, por lo que los cuerpos de estandarización de la ISO actualizaron el estándar en el año 2003, y están disponibles como ISO 11898-1 [ISO99-1] (para la capa enlace), ISO 11898-2 [ISO99-2] (para la capa física de alta velocidad), e ISO 11898-3 [ISO99-3] (para bajos requerimientos de velocidad). Además se cuenta con la especificación de TTCAN³ [ISO 11898-4] como extensión del Protocolo CAN.

Actualmente las cuatro diferentes partes del documento ISO 11898 [ISO11898-1, ISO11898-2, ISO11898-3 e ISO11898-4] están bajo revisión. El presente proyecto de titulación presentará las últimas modificaciones disponibles del protocolo CAN.

¹ Sociedad de Ingeniería Automotriz, encargada de la diseminación de técnicas y conocimientos aplicados a la tecnología de la movilidad en sus variadas formas: terrestre, marítima, aeroespacial.

² Grupo relacionado al desarrollo de tecnología industrial y manufacturación de automotores.

³ Protocolo que especifica un ambiente determinístico en la transferencia de mensajes CAN por medio de un esquema predefinido de tiempo.

1.2 REDES DE COMUNICACIÓN DE DATOS

El término redes de computadoras se utiliza para designar un conjunto de computadoras autónomas interconectadas para el intercambio de información digital, en forma de ceros y unos, a través de alguna forma de medio de transmisión. Las redes usan procesamiento distribuido, es decir que una tarea está dividida entre múltiples computadoras. En lugar de usar una única máquina grande responsable de todos los aspectos de un proceso, cada computadora individual maneja un subconjunto de ellos. Entre algunas de las ventajas del procesamiento distribuido se tiene: seguridad, bases de datos distribuidas, resolución más rápida de problemas, proceso cooperativo, entre otras.

El primer paso en el campo de las comunicaciones de datos es distribuir el acceso de sus colaboradores, las redes con topología en estrella permitían esta característica, donde un gran número de usuarios se conectaba a una CPU por medio de una conexión punto a punto. La evolución de las topologías de red permitió el desarrollo de nuevos protocolos reduciendo los tiempos necesarios de la comunicación misma, mejorando de esta manera la integridad de los datos transmitidos. El paso entonces fue a una estructura tipo árbol, donde los usuarios están conectados a un cable único, en el que la información a transmitir es multiplexada en el tiempo. Posteriormente la aparición del sistema de bus y su implementación hasta el nivel de actuadores y sensores. El próximo desafío es la sincronización para eventos en tiempo real.

1.3 DOMINIO DE CAMPO Y SISTEMAS DE BUS DE CAMPO

El término “Dominio de Campo”, se refiere a la parte de un sistema de automatización del cual representa la base de su estructura jerárquica (piso de planta¹). Los términos bus de campo y dispositivos de campo constituyen, entonces, elementos del Dominio de Campo.

¹ El Anexo B ilustra esta estructura jerárquica.

Un bus de campo es un sistema de comunicación serial¹ para el intercambio de datos. Típicamente son redes digitales, bi-direccionales, multipunto, montadas sobre un bus serie, que conectan dispositivos de campo.

Los dispositivos de campo, por su parte, hacen mención a todos los equipos y dispositivos que interactúan directamente con un proceso a ser monitoreado o controlado. De tal forma que estos no solamente son utilizados como sensores, medidores o controladores de un proceso técnico (PLCs, PCs, etc), sino también como influenciadores directos sobre estos procesos (regulación, operación, control, visualización, etc.).

Los sistemas Bus de Campo simplifican enormemente la instalación y operación de máquinas y equipamientos industriales utilizados en procesos de producción.

El objetivo de un bus de campo es sustituir las conexiones punto a punto entre los elementos o dispositivos de campo y el equipo de control.

Cada dispositivo de campo incorpora cierta capacidad de proceso, que lo convierte en un dispositivo inteligente, manteniendo siempre un costo bajo.

El uso de sistemas de comunicación serial de datos en lugar de los sistemas clásicos de cableado garantiza mucha más alta flexibilidad con respecto a modificaciones y actualizaciones, ofrece considerable potencial para reducir costos de instalación y planificación en muchas áreas de automatización industrial. En un estudio relacionado al tema [Rath97], el potencial ahorro de una solución basada en un bus de campo fue estimada ser aproximadamente del 40% comparada a una solución basada en sistemas cableados convencionales.

El uso de sistemas de bus seriales también ha prevalecido en la electrónica interna del vehículo para trabajar en redes de unidades de control así como para conexiones sencillas entre el cuerpo electrónico y la tecnología de cableado. El protocolo de Red de Área Controlada CAN (Controller Area Network) especificado

¹ Concepto de comunicación en el que se envían y reciben bytes de información, un bit a la vez.

por la compañía BOSCH y estandarizado internacionalmente por la ISO ha asumido un rol dominante al respecto. Sin embargo, éste sistema de comunicación de datos ha ganado importancia mas allá del uso en vehículos motorizados, ya que también es aplicado para el trabajo en redes de sistemas inteligentes distribuidos de-centralizados en una multitud de aplicaciones industriales, y actualmente representa el concepto de comunicación casi exclusivamente utilizado para los así llamados “sistemas embebidos”¹.

1.4 JERARQUÍAS DE PROTOCOLOS

Para reducir la complejidad de su diseño, la mayoría de las redes están organizadas como una pila de capas o niveles, cada una construida a partir de la que está debajo de ella. El número de capas, así como el nombre, contenido y función de cada una de ellas difieren de red a red. El propósito de cada capa es ofrecer ciertos servicios a las capas superiores, a las cuales no se les muestran los detalles reales de implementación de los servicios ofrecidos.

La capa n de una máquina mantiene una conversación con la capa n de otra máquina, las reglas y convenciones utilizadas en esta conversación se conocen de manera colectiva como protocolo de capa n. Básicamente, un protocolo es un acuerdo entre las partes en comunicación sobre cómo se debe llevar a cabo la comunicación.

La lista de protocolos utilizados por un sistema, un protocolo por capa, se conoce como pila de protocolos. Las entidades que abarcan las capas correspondientes en diferentes máquinas se llaman peers (pares). Y podrían ser procesos, dispositivos de hardware o incluso seres humanos. Es decir los pares son los que se comunican a través de los llamados protocolos “peer to peer” (Figura 1.1).

Entre cada par de capas adyacentes está una interfaz. Esta define que operaciones y servicios pone la capa más baja a disposición de la capa superior inmediata.

¹ Sistemas autocontenidos diseñados para cumplir una o más funciones específicas. Principalmente usados en sistemas y dispositivos de control, por ejemplo: actuadores o sensores.

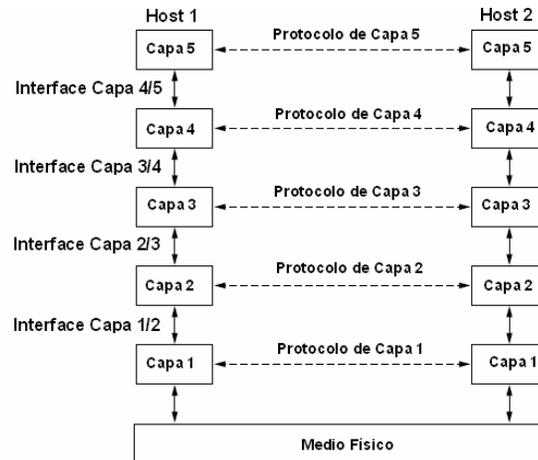


Figura 1.1: Capas, protocolos e interfaces

Un conjunto de capas y protocolos se conoce como arquitectura de red. La especificación de una arquitectura debe contener información suficiente para permitir que un diseñador escriba el programa o construya el hardware para cada capa de modo que se cumpla correctamente con el protocolo apropiado.

Una capa n produce un mensaje, la PDU (unidad de protocolo de datos) de capa n , lo transmite a su capa inmediatamente inferior (capa $n-1$), la cual recibe el mensaje como SDU (unidad de servicio de datos) de capa $n-1$, le agrega información de control y lo transforma en PDU de capa $n-1$, posteriormente lo transmite a su inferior (capa $n-2$), y a sí sucesivamente hasta llegar al medio físico.

Por ejemplo (Figura 1.2), un proceso de aplicación que se ejecuta en la capa 5 produce un mensaje, M , y lo pasa a la capa 4 para su transmisión. La capa 4, pone un encabezado al frente del mensaje para identificarlo y pasa el resultado a la capa 3. El encabezado incluye información de control, como números de secuencia, para que la capa 4 de la máquina de destino entregue los mensajes en el orden correcto si las capas inferiores no mantienen la secuencia. En muchas redes no hay límites para el tamaño de los mensajes en el protocolo de capa 4, pero casi siempre hay un límite impuesto por el protocolo de capa 3. Por lo tanto la capa 3 desintegra en unidades más pequeñas, llamados paquetes, a los

mensajes que llegan, y a cada paquete le coloca un encabezado. En este ejemplo, M se divide en dos partes, M1 y M2.

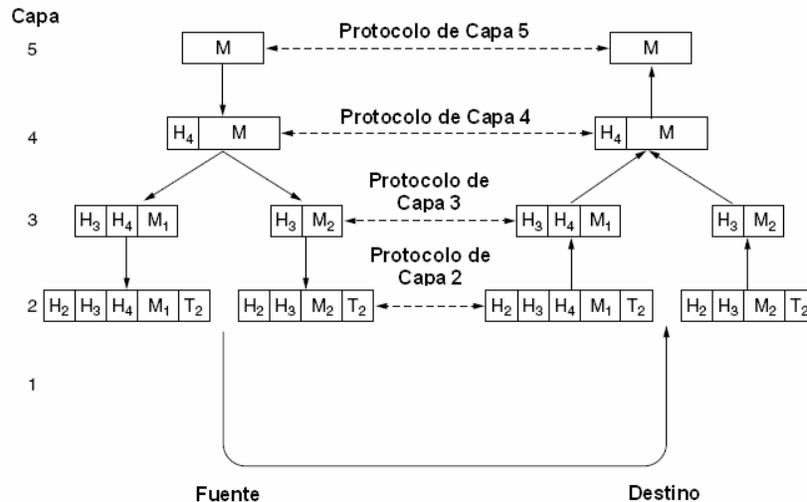


Figura 1.2: Ejemplo de flujo de información

La capa 3 decide cuál de las líneas, que salen, utilizar y pasa los paquetes a la capa 2. Ésta no sólo agrega un encabezado a cada pieza, sino también una cola, y pasa la unidad resultante a la capa 1 para su transmisión física. En la máquina receptora el mensaje pasa hacia arriba de capa en capa, perdiendo los encabezados conforme avanza. Ninguno de los encabezados de capas inferiores a n llega a una capa n.

1.5 EL MODELO DE REFERENCIA OSI Y LOS SISTEMAS DE COMUNICACIÓN INDUSTRIAL

El modelo de referencia se llama OSI (Interconexión de Sistemas Abiertos) porque tiene que ver con la conexión de sistemas abiertos.

El objetivo de OSI es permitir la comunicación entre sistemas distintos sin que sea necesario cambiar la lógica del hardware o el software subyacente. El modelo OSI, no es un protocolo; es un modelo para comprender y diseñar una arquitectura de red flexible, robusta e Inter-operable.

El modelo OSI tiene siete capas. Los principios aplicados para llegar a dichas capas son:

- Una capa se debe crear donde se necesite una abstracción diferente.
- Cada capa debe realizar una función bien definida.
- La función de cada capa se debe elegir con la intención de definir protocolos estandarizados internacionalmente.
- Los límites de las capas se deben elegir a fin de minimizar el flujo de información a través de las interfaces.
- La cantidad de capas debe ser lo suficientemente grande para no tener que agrupar funciones distintas en la misma capa y lo bastante pequeña para que la arquitectura no se vuelva inmanejable.

La funcionalidad de todas las capas más bajas es oculta y completamente transparente para las capas más altas.

El modelo fue diseñado con la intención de proveer una estructura general para cualquier tipo de sistema complejo de comunicación, debido a esto no todos los sistemas pueden usar todas las siete capas del modelo.

Por ello, las capas no usadas son llamadas “capas nulas”. Generalmente, funciones como el establecimiento de la conexión sobre varias redes parciales no son relevantes para la automatización industrial o sistemas de comunicación de automotores.

Esa es la razón por la que, para comunicaciones de datos en el área industrial y de automotores generalmente solo sean necesarias tres capas (Figura 1.3).

Estas son:

- Capa Física
- Capa Enlace

- Capa Aplicación

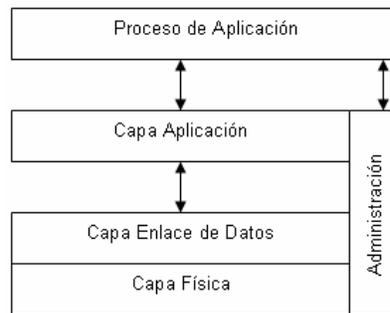


Figura 1.3: Modelo de comunicación de datos de tres capas

Con lo que la funcionalidad es limitada sólo a la que es requerida por los sistemas de bus de campo, obteniendo así un incremento en la eficiencia de la pila de protocolos y reduciendo el tiempo de latencia causado por el software de comunicación involucrado.

A continuación se resume la funcionalidad de las tres capas mencionadas:

- **Capa Física (Capa 1, Capa de Transmisión de Bits)**

Especifica parámetros funcionales y de procedimiento de la conexión física entre los nodos de una red. La Capa Física se encarga de codificar los datos pasados a su correspondiente en el nodo receptor, como un flujo de bits que es transmitido en forma de señal física sobre un medio de transmisión. A su vez el nodo receptor de la señal tiene que decodificarla al formato de datos esperado en su capa enlace. Ésta capa también se encarga de funciones de sincronización y temporización de bit así como la conexión física al medio. La capa Física describe la conexión al medio de transmisión, pero no especifica la forma del medio de transmisión.

- **Capa Enlace (Capa 2)**

Define la transmisión de datos entre los nodos de una red. Su principal función es la de construir tramas de datos, que no solo contienen los datos a ser

transmitidos sino también información de control y códigos de detección de errores. Otra de sus funciones es controlar el acceso del nodo al medio de transmisión (“Control de Acceso al Medio” (MAC)¹). Este mecanismo soluciona el problema de dos o más nodos que quieren acceder al medio de transmisión al mismo tiempo.

- **Capa Aplicación (Capa 7)**

Es la capa a la que el usuario de un determinado programa normalmente accede para comunicarse con la red. Provee aplicaciones básicas de sistemas de comunicaciones que pueden ser usadas en forma general (como intercambio de archivos, uso de terminales de nodos remotos, intercambio y administración de mensajes con usuarios de otros sistemas), también brinda funciones básicas de comunicación comunes a muchas aplicaciones (como establecimiento y terminación de conexiones).

- **Administración y Control de Capas**

Adicionalmente a las funciones relacionadas a comunicación descritas por varias de las capas del modelo OSI, puede haber una necesidad de aplicación de un conjunto de parámetros operacionales de una capa más baja o la de recibir información de una capa más baja de una pila de protocolos, para lograr esto se debe implementar alguna forma de sistema de administración de capas, que permita que la aplicación acceda a la funcionalidad de las diferentes capas. Estas funciones son simbolizadas con el término “Administración de Capas”. Este tipo de funcionalidad es mayormente usada al inicio de una aplicación, por ejemplo para configurar el modo de operación, establecer la velocidad de transmisión u otros parámetros de operación. La aplicación puede también ser informada cuando errores significantes son detectados en una capa específica.

1.6 MODELO DE COMUNICACIÓN CLIENTE-SERVIDOR Y PRODUCTOR-CONSUMIDOR

¹ Los diferentes métodos de control de acceso serán revisados en este capítulo más adelante.

El modelo “Cliente-Servidor” siempre describe una relación de comunicación uno a uno entre un cliente y un servidor (aún si existieran varios posibles clientes), el modelo “Productor-Consumidor” describe una relación de comunicación uno a varios entre un productor y uno o varios consumidores. Cualquiera de los más modernos conceptos de comunicación de datos aplicado en automatización industrial soporta ambos modelos.

La Figura 1.4 muestra el proceso de comunicación de un servicio confirmado de acuerdo al modelo Cliente-Servidor, en donde un proceso N (solicitante del servicio, cliente) pide un servicio específico a través de una “primitiva¹ de solicitud de servicio” enviada al Sistema de Comunicación. La transmisión de la solicitud hecha por el sistema de comunicación es indicada al proceso M (proveedor del servicio, servidor) por medio de una “indicación de servicio”. Después de ejecutar el servicio solicitado, el servidor reporta esta ejecución junto con el resultado de la misma al sistema de comunicación por medio de una “primitiva de respuesta de servicio”. Finalmente el sistema de comunicación informa al cliente del éxito de la ejecución del servicio solicitado con una “primitiva de confirmación de servicio”.

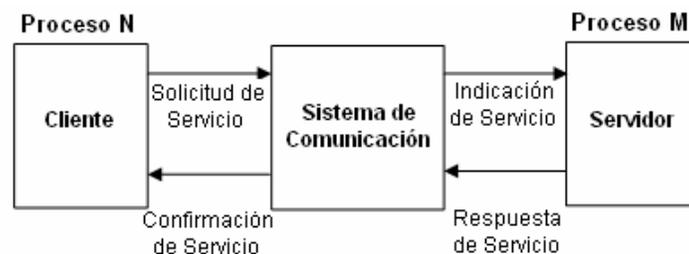


Figura 1.4: Interacción de un servicio confirmado de acuerdo al modelo Cliente-Servidor

En caso de servicios sin confirmación, no se emplean los mensajes de respuesta ni de confirmación de servicio.

En general, los procesos también pueden ser relacionados a capas de comunicación. En este caso una solicitud de servicio se refiere a un servicio provisto por la próxima capa inferior de comunicación. La Figura 1.5 ilustra la

¹ Comandos de software.

secuencia de un servicio confirmado en el modelo Cliente-Servidor. Las líneas verticales simbolizan las capas del stack de comunicación de los respectivos nodos involucrados.

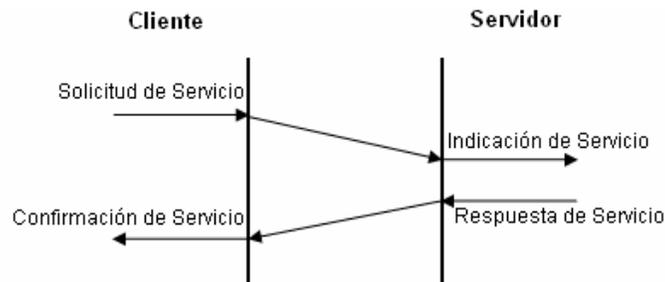


Figura 1.5: Secuencia de un servicio confirmado de acuerdo al Modelo Cliente-Servidor.

Por su parte servicios en los que los datos son provistos en forma de mensajes multicast o broadcast a otros procesos son más apropiadamente descritos por el modelo Productor-Consumidor ilustrado en la Figura 1.6. Los servicios provistos por el Productor son ofrecidos a otros procesos a través de una indicación de servicio y pueden ser aceptados (consumidos) o ignorados por éstos procesos.

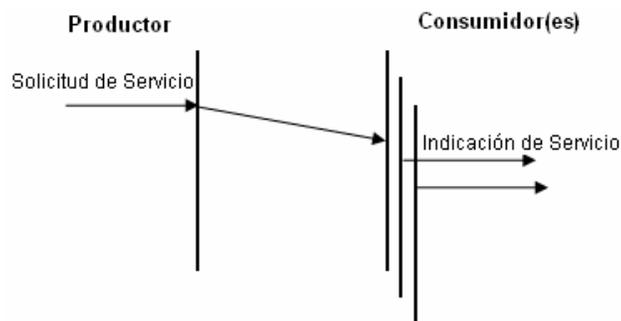


Figura 1.6: Secuencia de una solicitud de servicio de acuerdo al modelo Productor-Consumidor

Debido a que el productor normalmente no sabe quienes son los consumidores, no usa una confirmación de servicio. Este modelo describe la transmisión de mensajes CAN¹.

¹ CAN solamente soporta el modelo Productor-Consumidor, sin embargo protocolos de capa superior basados en CAN (CAL, CANopen o DeviceNet) adicionalmente soportan el modelo Cliente-Servidor (Ver capítulo 3).

1.7 PROTOCOLOS ORIENTADOS AL NODO

En este tipo de protocolos, el intercambio de datos entre dos o más nodos está basado en el direccionamiento de nodos. Generalmente la trama transmitida sobre el medio de transmisión contiene la dirección destino y a veces la de la fuente de los nodos envueltos en la transmisión (Figura 1.7), por lo que una trama puede ser enviada a un nodo específico o a un grupo de ellos (multicast). Se reservan direcciones especiales para tramas que son enviadas a un grupo de nodos o a todos los nodos (broadcast). Este tipo de transmisión requiere de una confirmación por parte del nodo receptor de haber recibido la trama y de que ésta es correcta. La comunicación orientada al nodo es la base para la comunicación de datos confirmados orientados a la conexión¹.

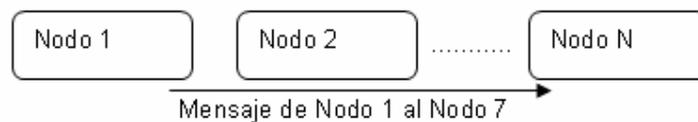


Figura 1.7: Comunicación de datos orientados al nodo

1.8 PROTOCOLOS ORIENTADOS AL MENSAJE

En este protocolo el intercambio de datos está basado en identificadores de mensajes o tramas. Un mensaje transmitido por un nodo se identifica a sí mismo con un único identificador específico (identificador de mensaje). El nodo destino no está definido en el mensaje. Es decisión de los nodos de aceptar o no un mensaje transmitido (Figura 1.8). Es posible que un mensaje sea aceptado por uno (unicast) o varios nodos. Así el principio de la transmisión orientada al mensaje representa un principio básicamente diferente al de la transmisión orientada al nodo.

Este es un método que cada vez está siendo más aplicado en sistemas de comunicación modernos, frecuentemente en combinación con transmisión orientada al nodo. Debido a que el transmisor del mensaje no sabe a ciencia

¹ Un servicio orientado a la conexión consta de un proceso de establecimiento de la conexión, transferencia de la información y liberación de la conexión.

cierta cual será el receptor del mismo, no es posible realizar un intercambio de mensajes de confirmación con el receptor. Habitualmente se usa el principio de “señalización de error” en vez del uso de ACKs¹.

Casi todos los sistemas de comunicación en los automóviles están basados en protocolos orientados al mensaje.

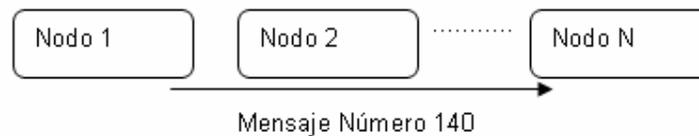


Figura 1.8: Comunicación de datos orientados al nodo

1.9 CONTROL DE ACCESO AL MEDIO

Un Control de acceso al medio, “MAC”, es usado para determinar cuál de los transmisores accede al medio de transmisión. Su objetivo es evitar que dos o más nodos transmitan al mismo tiempo interfiriendo sus señales entre sí.

El método de control de acceso en el cual está basado un sistema de bus de campo es determinante en cuanto a su rendimiento, latencia y comportamiento en tiempo real. Además de que puede ser decisivo a la hora de elegir el protocolo de comunicación.

En general los métodos de acceso al bus pueden ser clasificados como métodos con acceso determinístico y métodos con acceso aleatorio (Ver Figura 1.9).

Con un método de acceso al bus determinístico, el derecho a transmitir está claramente definido antes de acceder al bus, lo cual garantiza que sólo un nodo tendrá acceso al bus y no existirán colisiones. Pueden ser empleados métodos controlados en forma “centralizada” o “no centralizada”, a riesgo de que si la entidad central falla, todo el sistema fallará también. La realización de métodos de acceso no centralizados es más compleja sin embargo éstos son más flexibles y generalmente se mantienen operables aún cuando un nodo falla o se desactiva.

¹ Mensajes usados como acuses de recibo.

Con los métodos de acceso aleatorio al bus, los nodos pueden acceder a éste tan pronto como se encuentre desocupado, sin embargo después de detectar esto más de un nodo puede querer acceder al bus al mismo tiempo, por esta situación éste método es llamado “Acceso Múltiple con Detección de Portadora” (CSMA). Los diferentes métodos de acceso aleatorio al bus están clasificados como métodos con colisiones y métodos libres de colisiones, dependiendo de si ocurren o no colisiones en la transmisión de los mensajes.

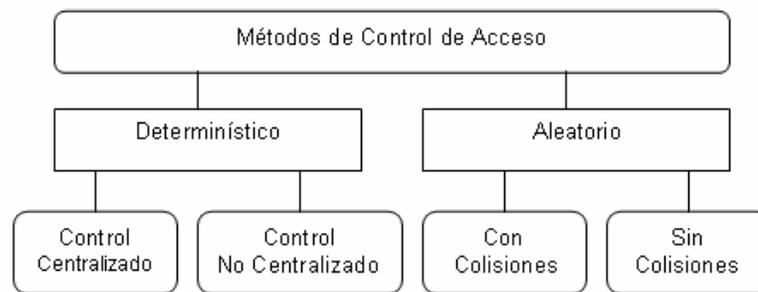


Figura 1.9: Clasificación de los métodos de acceso al bus

En sistemas con acceso aleatorio pero completamente libres de colisiones, como el sistema CAN, un nodo detecta un acceso simultáneo previa una etapa llamada “Fase de Arbitraje” la cual está basada en la evaluación de la prioridad de todos los mensajes que compiten por el acceso al medio, concediéndole acceso al mensaje de mayor prioridad, de tal manera que el peor tiempo de latencia puede ser conocido, lo que asegura un ambiente determinístico para mensajes con alta prioridad de transmisión.

Los siguientes métodos de acceso al bus son los principalmente usados:

- Maestro-Eslavo
- Token Delegado
- Token Passing
- Acceso Múltiple por división de Tiempo (TDMA)
- Acceso Múltiple por Detección de portadora (CSMA)

- CSMA con Detección de Colisiones
- CSMA con Prevención de Colisiones

1.9.1 MAESTRO – ESCLAVO

En este esquema, un nodo es designado como “maestro”, y es el que asume el control de acceso al bus en términos de un intercambio de datos orientado al nodo, “poleando”¹ cíclicamente al resto de nodos (“esclavos”). Un maestro puede transmitir datos a un nodo específico con una “trama de polling”² y recibir datos en una trama de respuesta del nodo en mención (Figura 1.10).



Figura 1.10: Principio del proceso de acceso Maestro-Escavo

La ventaja principal de éste método es su simplicidad, así como un tiempo de latencia máximo determinado después del cual el bus está disponible para un nodo.

Una desventaja de éste método es que un nodo esclavo debe ser siempre “poleado” ya sea que requiera o no acceso al bus (lo que determina el tiempo máximo de latencia). Esto se traduce en una carga innecesaria al bus, provocando a su vez la necesidad de altos requerimientos, particularmente en sistemas con una tasa de generación de datos aleatoria. De otra manera el principio cíclico del poleo consecutivo utilizado en el método maestro-esclavo es el que mejor se ajusta en procesos cíclicos, como por ejemplo en la actualización cíclica del procesamiento de imágenes de una unidad de control PLC.

Otra desventaja del método maestro-esclavo es que solo son posibles las comunicaciones “uno a varios” (Figura 1.11). Esto significa que los datos pueden

¹ El nodo maestro envía un mensaje a sus esclavos, invitándoles a transmitir.

² Mensaje que el maestro envía a sus esclavos para efectuar el polling o poleo.

ser intercambiados entre esclavos únicamente a través del nodo maestro, y si éste falla todo el sistema también fallará.

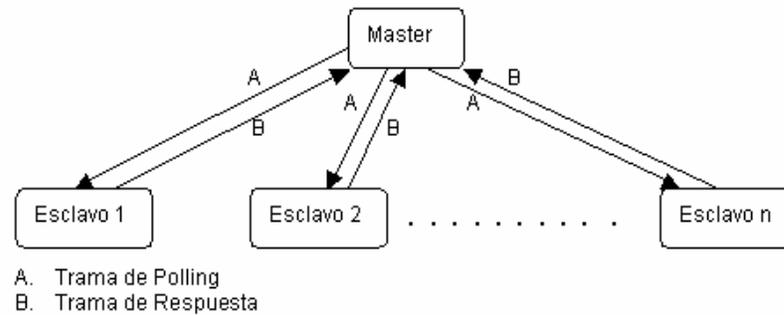


Figura 1.11: Estructura de la comunicación “uno a varios”

1.9.2 TOKEN DELEGADO

De acuerdo a éste sistema una entidad central (master o árbitro del bus) delega el derecho de acceso al medio transmitiendo un “token”¹ que es idéntico al identificador del mensaje. De esta manera el nodo poseedor del token puede enviar mensajes que pueden ser aceptados por otros nodos interesados. De acuerdo a un esquema predefinido de tiempo el token cedido por el árbitro del bus es devuelto por el nodo, y el derecho de acceso al medio regresa al nodo árbitro del mismo, para que se repita el proceso. El resultado es un sistema orientado al mensaje con un ambiente determinístico.

La Figura 1.12 muestra la secuencia de mensajes en el bus de acuerdo al principio del token delegado.

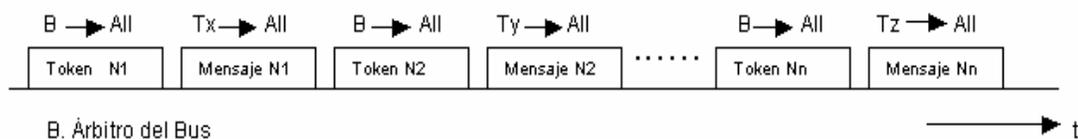


Figura 1.12: Secuencia básica de mensajes en un bus con acceso por token delegado

¹ Secuencia específica de bits que lo identifica como tal.

El método de acceso por token delegado corresponde a un sistema de distribución de mensajes controlado centralmente. Cada nodo poseedor del token es capaz de entregar mensajes a todos los otros nodos, lo que hace posible una estructura “varios a varios” y un intercambio entre cualesquiera de los nodos de un sistema distribuido (Ver Figura 1.13).

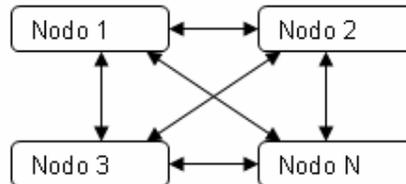


Figura 1.13: Estructura de comunicación Varios a Varios

Una desventaja de éste sistema, es que el sistema entero falla cuando el árbitro del bus falla, por ello el “token delegado” debe ser usado necesariamente en conexiones con un diseño redundante de árbitro del bus.

1.9.3 TOKEN PASSING

Este es un método no centralizado, basado en la rotación de un mensaje específico (token). Como lo indica la Figura 1.14, el derecho de acceso al bus es pasado de estación a estación por medio del token. Tan pronto como una estación ha recibido el mensaje de token, ésta puede usar el bus para intercambiar mensajes con otras estaciones por un periodo de tiempo limitado (“tiempo de espera del token”). Después del lapso de tiempo que dura la posesión del token, éste debe ser pasado a la estación siguiente. Este método de acceso al bus es relativamente complejo en un sistema que soporta integración y exclusión dinámica de estaciones. Además se requiere de medidas relativamente complejas para el establecimiento de la lógica de anillo, monitoreo del proceso de rotación del token o re-inicialización de la lógica de anillo después de una posible pérdida del token. Este principio permite el intercambio de datos dentro de una estructura de comunicación “varios a varios” y es particularmente usada para transferencia de datos entre estaciones con la misma prioridad. El tiempo de posesión del token y el número de estaciones determinan el tiempo de latencia de

un sistema “token passing”. La Figura 1.14 ilustra el principio de operación de éste método.

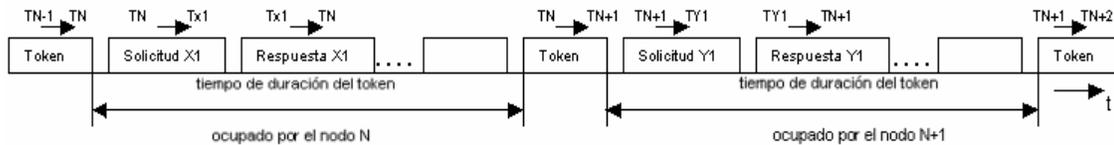


Figura 1.14: Principio del acceso al bus por medio de token passing

1.9.4 ACCESO MULTIPLE POR DIVISIÓN DE TIEMPO (TDMA)

Con este método, se reservan “slots” o intervalos determinados de tiempo para que cada nodo tenga un acceso independiente al bus. Al contrario del método de “token” delegado, los nodos no son “poleados” por una entidad central, sino que son asignados al bus basados en un punto predefinido en el tiempo. Para lo cual se requiere de un alto sincronismo del tiempo de referencia local en cada nodo.

TDMA es aplicado de diferentes maneras cuando es requerido un ambiente determinístico exacto de tiempo. El tiempo de latencia es determinado por el número de nodos, número de mensajes y la velocidad de transmisión.

1.9.5 ACCESO MULTIPLE POR DETECCIÓN DE PORTADORA (CSMA)

Cada uno de los nodos puede acceder al bus tan pronto como éste se encuentre desocupado. Las estaciones realizan un sondeo en busca de una señal portadora que indique que se está llevando a cabo una transmisión, si el medio está libre se transmite inmediatamente. Comparado a procesos cíclicos, éste presenta el promedio más bajo de carga al bus así como un periodo muy corto de latencia. Sin embargo es posible que dos o más estaciones intenten usar el bus simultáneamente después de que éste se desocupe, lo que ocasionaría colisiones. Por esa razón se han realizado algunas variantes a éste método, como son CSMA/CD y CSMA/CA, explicadas a continuación.

1.9.5.1 CSMA/CD (CSMA con detección de colisiones)

Este método de acceso tiene por objeto proveer acceso múltiple a los nodos de una red con topología de bus, y está basado en el principio de sensar o sondear el medio de comunicación antes y durante la transmisión de un paquete de información, asumiendo que el retraso en la propagación es muy corto comparado con el tiempo de transmisión del paquete de datos.

La idea básica del CSMA es que cuando una terminal necesita transmitir, primero revisa al medio de transmisión para ver si no hay una transmisión en progreso y evitar así una colisión. Si el medio está libre, espera un corto periodo de tiempo después del cual inicia la transmisión.

Esta modificación, aún cuando parece muy simple y obvia, logra reducir el número promedio de colisiones, pero no las evita, ya que dos o más terminales pueden iniciar una transmisión al mismo tiempo y producir una colisión.

La utilización máxima usando CSMA depende de la relación del tiempo de propagación con el tiempo de transmisión. En redes locales esta relación se conoce como "a". Los valores pequeños de la relación "a" favorecen una utilización mayor de la red, mientras que los valores grandes provocan una utilización menor.

Las principales características de CSMA/CD son:

- Detecta una colisión durante la transmisión, suspendiendo la transmisión del paquete inmediatamente y generando una señal breve que indica que hubo colisión (señal de jamming).
- Después de indicar la colisión, los nodos involucrados en la misma esperan un tiempo aleatorio, luego del cual harán un nuevo intento por llevar a cabo la transmisión. La intención es que los nodos esperen tiempos diferentes, evitando así una nueva colisión (Figura 1.15).

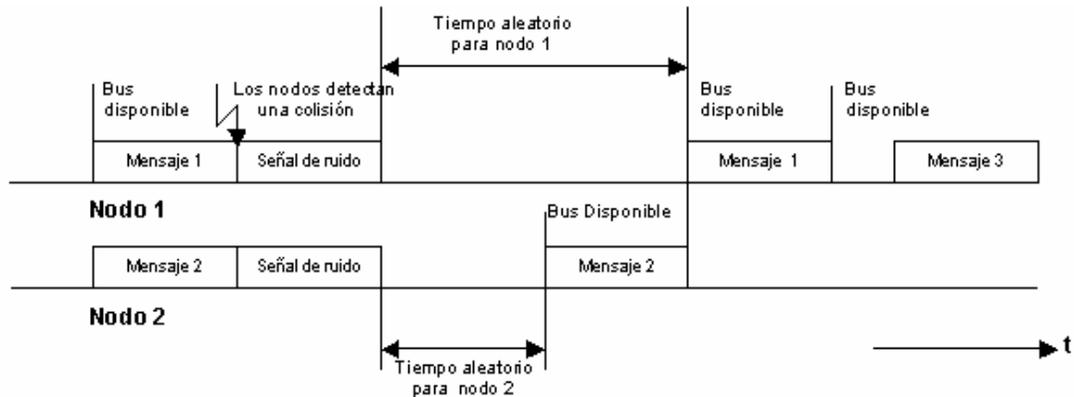


Figura 1.15: Principio de acceso al bus por CSMA/CD

El tiempo máximo que se necesita para detectar una colisión es dos veces el máximo tiempo de propagación.

1.9.5.2 CSMA/CA (CSMA con evasión de colisiones)

Es una variación del método de acceso CSMA/CD con el adicional de que aquí no existen colisiones. Aunque es posible que inicialmente varios nodos empiecen a transmitir un mensaje al mismo tiempo, se asegura que el nodo que desea transmitir los mensajes de mayor prioridad acceda al bus. Durante una fase de acceso múltiple, los nodos envían simultáneamente sus identificadores de mensaje (correspondientes a la prioridad del mensaje) al bus, en donde se identifica al nodo con la máxima prioridad, el cual ganará el acceso al bus. Las estaciones que no consigan el acceso volverán a intentarlo después de que el medio este libre de nuevo. Se ajusta para eventos de transferencia orientada al mensaje. Además, si se asegura que un mensaje de alta prioridad puede ser transmitido en un intervalo mínimo de tiempo definido, a pesar del acceso aleatorio, se podría entonces predecir un tiempo máximo de latencia de los mensajes de alta prioridad y consecuentemente brindar un ambiente determinístico para ellos.

Prácticamente todos los sistemas de bus seriales usados en vehículos motorizados están basados en este principio. Más adelante se describirán más particularidades de este método.

1.10 DETECCIÓN Y CONTROL DE ERRORES

En todo Sistema de Transmisión, independientemente de cómo haya sido diseñado, habrá ruido. El ruido alterará la señal transmitida, lo que dará lugar a errores que modificarán uno o varios bits de la trama. Estas alteraciones de la señal no se pueden evitar por completo aunque se empleen costosos métodos. Para minimizar sus efectos básicamente se requiere de mecanismos de chequeo de error que permiten al nodo receptor detectar si un mensaje recibido es correcto o incorrecto.

Todos los métodos comúnmente empleados para este propósito están basados en el envío de información adicional que permiten al receptor detectar errores en la transmisión con una probabilidad bastante alta (Figura 1.16).

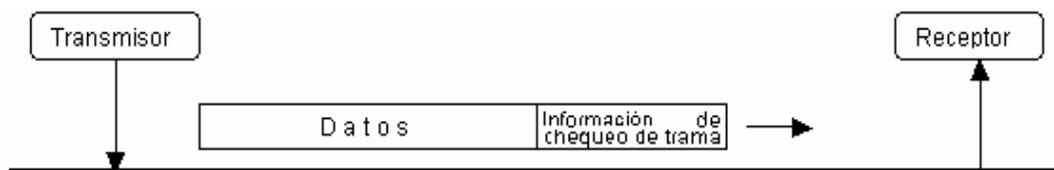


Figura 1.16: Principio básico de detección de errores en la transmisión de datos

También es necesario que el receptor de un mensaje erróneo pueda informar esta situación a su transmisor con el objeto de que éste re-transmita el mensaje.

La información adicional añadida para detección de errores comúnmente se denomina “Frame Check Sequence” (Secuencia de Chequeo de Trama) y se genera en el transmisor de acuerdo a una regla específica, la cual es igualmente aplicada en el receptor para recalcular este valor, que será comparado con el recibido. De existir diferencia entre éstos, se concluye que existen errores en los datos recibidos. Desafortunadamente este método no garantiza una transmisión correcta ya que determinadas secuencias de errores pueden no ser detectadas. Por ejemplo con métodos de chequeo de paridad, no es posible detectar un número par de errores.

El mecanismo de detección de errores más importante para “comunicación serial de datos” es el denominado CRC “Cyclic Redundancy Check” (Chequeo de Redundancia Cíclica). Este mecanismo considera a los bits que van a ser transmitidos en una trama de datos como coeficientes binarios de un polinomio de grado apropiadamente alto. Este polinomio es dividido por un denominado “Polinomio Generador” de grado específico conocido por el receptor. El residuo de ésta división es transmitido como FCS junto con la trama de datos. El receptor lleva a cabo la misma operación y compara el resultado con el FCS recibido. Si ambos resultados coinciden quiere decir que la trama recibida no contiene errores o viceversa.

Otros métodos de chequeo de errores, como el de adicionar a la trama de datos otra trama que forma un valor de “check sum” son mucho menos efectivos.

Generalmente el tamaño del FCS adicionalmente transmitido está en relación a cuan grande es la capacidad de detección de errores que posee un mecanismo de chequeo de errores. Desafortunadamente una secuencia de chequeo más larga reduce la eficiencia de un protocolo, puesto que la secuencia de chequeo adicionalmente transmitida representa información redundante.

El método aplicado para control de errores depende de si se trata de un protocolo orientado al nodo o al mensaje. Para protocolos orientados al nodo principalmente se usa la técnica de “Control Pasivo de Errores”, mientras que para los protocolos orientados al mensaje se aplica el principio de “Señalización de Error”.

1.10.1 CONTROL PASIVO DE ERRORES

Después de la transmisión de un mensaje, el transmisor espera una respuesta del receptor o ACK por un periodo máximo determinado de tiempo. Si el receptor detecta un error en la transmisión de los datos, ignora el mensaje recibido y se abstiene de responder al transmisor con un ACK. Si después del periodo máximo de confirmación el transmisor no recibe respuesta, éste automáticamente inicia la retransmisión del mensaje afectado (Figura 1.17). Puede darse el caso de que el

ACK del receptor llegue alterado, produciéndose una retransmisión, por ello es necesario de medidas adicionales en el receptor para detectar éste tipo de retransmisiones.

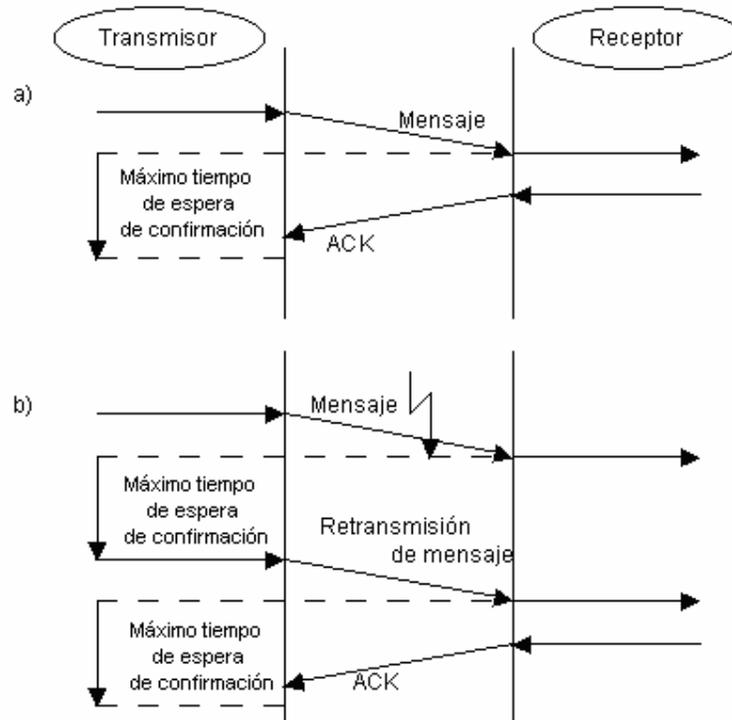


Figura 1.17: Principio del Control Pasivo de Errores

1.10.2 SEÑALIZACIÓN DE ERRORES

Una desventaja del método anterior es que el transmisor tiene que esperar hasta que el periodo máximo de confirmación haya expirado para iniciar una retransmisión. Además, la transferencia de mensajes de acuse de recibo, como se ha descrito, no puede ser aplicada tan fácilmente a mensajes direccionados a un grupo de nodos, pues esto requeriría de capacidad adicional para recolectar ACKs provenientes de cualquiera de estos nodos.

En protocolos orientados al mensaje normalmente los receptores son desconocidos, lo que hace que el principio de confirmación de mensajes no sea aplicable. Estos protocolos emplean el principio de señalización directa. Aquí el receptor indica al transmisor que ha recibido un mensaje erróneo transmitiendo, a

su vez, una señal específica corta de error, así el transmisor puede retransmitir el mensaje inmediatamente después de recibida esta señal (Figura 1.18), con lo que son posibles muy cortos tiempos de recuperación de errores. Una ventaja adicional de este método es que asegura coherencia a una red extensa de datos puesto que todos los nodos reciben la señal de error al mismo tiempo. Este principio es aplicado en el protocolo CAN.

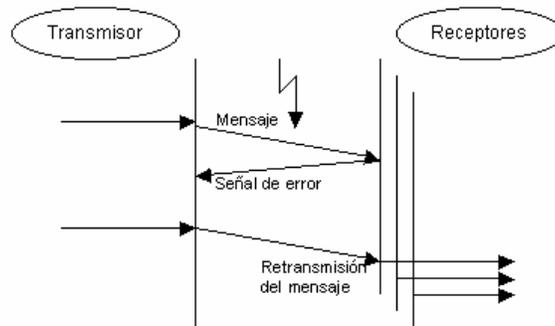


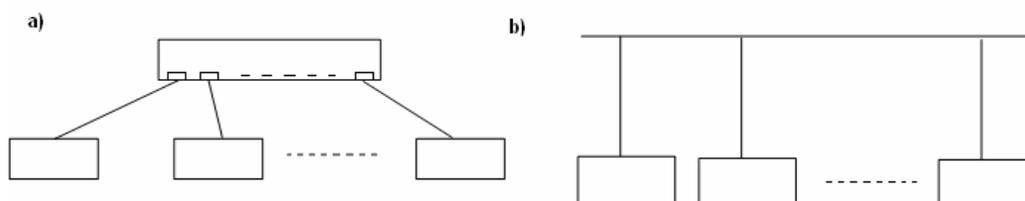
Figura 1.18: Principio del método de señalización de Errores

1.11 TOPOLOGIAS DE RED

Una topología de red describe la estructura física de la conexión de los nodos de una red de comunicación. La cual determina el costo requerido para la implementación física de una red, sus límites de aplicación y parámetros físicos.

Las topologías más importantes para buses de campo son:

- Estrella
- Bus
- Árbol
- Anillo



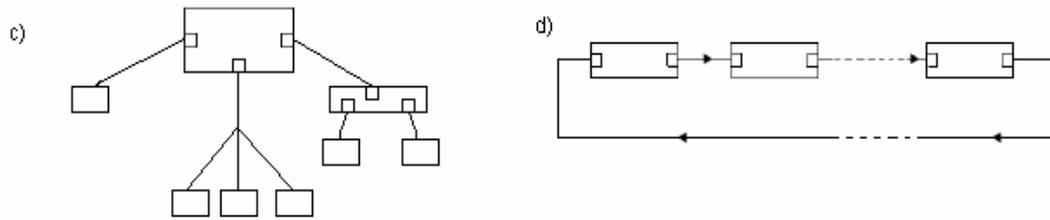


Figura 1.19: Principales Topologías de Red

1.11.1 TOPOLOGÍA EN ESTRELLA

Todos los nodos son conectados a un nodo central a través de conexiones punto a punto (Figura 1.19a).

Sus ventajas son:

- Cada nodo tiene su propia conexión al nodo central.
- Fácil instalación y re-configuración.
- Cada dispositivo necesita solamente un enlace y un puerto de entrada/salida para conectarse a cualquier número de dispositivos.
- Brinda robustez, si falla un enlace solamente este enlace se verá afectado.

Entre sus desventajas se tienen:

- Cada nodo debe estar enlazado al nodo central. Por esta razón, en la estrella se requiere más cable que en otras topologías de red.
- El nodo central requiere de N interfaces para conectar N nodos.
- Los nodos pueden comunicarse entre sí solamente a través del nodo central.
- Si el nodo central falla se pierde toda comunicación.

1.11.2 TOPOLOGÍA EN BUS

Denominado también como “red de difusión” ya que los datos transmitidos por un nodo están disponibles para todos los nodos. Su principal característica es la conexión de todos los nodos a un medio de transmisión en común (Figura 1.19b).

Sus principales ventajas son:

- Conexión simple de los nodos.
- La falla o desactivación de un nodo no afecta al resto de nodos.
- Son posibles estructuras de comunicación de lógica arbitraria.
- Permite una extensión simple a nodos adicionales sin interrumpir la operación de la red.

Sus desventajas más relevantes son:

- Si no se aplican repetidores que regeneren la señal, la longitud del bus será limitada así como la cantidad de nodos que se conectaran al mismo.
- Se requiere de identificación de los nodos.
- Un fallo o rotura del cable del bus interrumpe todas las transmisiones en una parte de la red.

A pesar de ello la topología de bus es de lejos la técnica más frecuentemente aplicada por buses de campo.

1.11.3 TOPOLOGÍA EN ÁRBOL

La topología en árbol es una variante de la estrella. Al igual que en la estrella, los nodos del árbol están conectados a un concentrador central que controla el tráfico de la red. Sin embargo, no todos los dispositivos se conectan directamente al concentrador central. La mayoría de los dispositivos se conectan a un concentrador secundario que, a su vez, se conecta al concentrador central (Figura 1.19c).

Las ventajas y desventajas de la topología en árbol son generalmente las mismas que las de una estrella. Sin embargo la inclusión de concentradores secundarios añade dos ventajas más:

- Permite que se conecten más dispositivos a un único concentrador central, y puede por lo tanto incrementar la distancia que puede viajar la señal entre dos dispositivos.
- Permite a la red aislar y priorizar las comunicaciones de distintas computadoras, por ejemplo las computadoras conectadas a un concentrador secundario pueden tener más prioridad que las computadoras conectadas a otro concentrador secundario.

1.11.4 TOPOLOGÍA EN ANILLO

En esta topología cada dispositivo tiene una línea de conexión dedicada punto a punto solamente con los dos dispositivos que están a sus lados. La señal pasa a lo largo del anillo en una dirección, o de dispositivo a dispositivo, hasta que alcanza su destino (Figura 1.19d).

Un anillo es relativamente fácil de instalar y reconfigurar, para añadir o quitar dispositivos, solamente hay que mover dos conexiones. Las únicas restricciones están relacionadas con aspectos del medio físico y el tráfico (máxima longitud del anillo y número de dispositivos).

El tráfico unidireccional puede ser una desventaja. En anillos sencillos, una rotura del anillo puede inhabilitar toda la red. Sin embargo esta debilidad puede ser superada utilizando un anillo dual o un conmutador capaz de puentear la rotura.

1.12 PRINCIPALES CARACTERÍSTICAS DE CAN

A continuación se citan de forma general las principales características del protocolo CAN, en el siguiente capítulo se las estudiará con más profundidad.

Topología de Bus, velocidad de transmisión, máximo número de nodos

CAN está basado en una topología lineal (o bus). Por medio de repetidores o routers es posible implementar topologías tipo árbol y estructuras jerárquicas. El número de nodos por red no está limitado por el protocolo, pero depende de la

capacidad de los chips empleados. Además, los repetidores permiten incrementar el número de nodos de una red. La máxima extensión posible de una red con una velocidad de transmisión específica es limitada por el tiempo de propagación de la señal a lo largo del bus. Además del uso de un par de cables, también es posible emplear soluciones basadas en fibra óptica de vidrio o plástico.

Protocolo orientado al mensaje

El intercambio de mensajes en el protocolo CAN no está basado en el direccionamiento del receptor del mismo, sino en la identificación de un mensaje transmitido a través de su identificador de mensaje. Basado en este identificador, todos los nodos chequean si un mensaje es relevante para sí mismo. Por lo tanto los mensajes pueden ser aceptados por uno, varios, o todos los nodos, o en su defecto por ninguno de ellos. Los nodos pueden ser librados de mensajes que no interesan por medio de un filtro de aceptación de mensajes integrado en los controladores del protocolo.

Priorización de mensajes

El identificador de un mensaje adicionalmente determina su prioridad con respecto al acceso al bus, de tal manera que los mensajes de mayor importancia tengan preferencia en acceder al medio, y se les garantice tiempos muy cortos de latencia, sin considerar la carga momentánea que el bus pueda tener. Esta característica asegura que en situaciones excepcionales los mensajes importantes sean transmitidos primero, asegurando el funcionamiento de un sistema durante fases de capacidad de transporte limitada.

Este método de manejo de situaciones de sobrecarga es difícilmente posible en otros tipos de sistemas de comunicación de datos.

Capacidad Multi-Maestro

El acceso al bus no es controlado por una unidad central de control (maestro del bus). Sino que cada nodo puede iniciar la transmisión de un mensaje tan pronto como el bus esté desocupado. En caso de un acceso simultáneo de varios

nodos, el nodo que desea transmitir un mensaje con la más alta prioridad recibe el derecho de utilizar el bus. De esta manera cada nodo puede comunicarse directamente con cada uno de los otros. Ya que la transmisión de un mensaje puede ser iniciada por la fuente misma del mensaje, el bus es ocupado solamente cuando un mensaje nuevo tiene que ser enviado (transmisión controlada por eventos). Esto resulta en un promedio significativamente más bajo de carga del bus comparado a un sistema de bus con acceso determinístico (como en el principio “Maestro-Esclavo”).

Acceso aleatorio libre de pérdidas

Debido a que en el protocolo CAN es empleado un acceso aleatorio al bus, es posible que varios nodos deseen ocuparlo simultáneamente. En otros esquemas de acceso aleatorio al bus (como CSMA/CD) los mensajes involucrados en una colisión son destruidos. La solución de un conflicto de acceso al bus requiere un nuevo método para acceder a éste basado en una estrategia de backoff. Como anteriormente se mencionó el protocolo CAN aplica un método de acceso el cual garantiza que el mensaje con la más alta prioridad entre todos los mensajes que están compitiendo simultáneamente por el acceso, sea transmitido sin destrucción.

Tamaño reducido de la trama

El tamaño máximo de los datos de un mensaje CAN está limitado a 8 bytes. Esta longitud de datos es suficiente para los requerimientos de comunicación de datos en vehículos y niveles de entrada/salida en sistemas embebidos y automatización. En estos campos de aplicación, la transmisión de tramas de datos más largas no es tan importante como el soporte de altas velocidades de transmisión para mensajes cortos. La transmisión de tramas de datos más largas por medio de mensajes CAN es requerida solo en casos excepcionales (como en configuración de parámetros del nodo, o carga de un programa).

En este caso, protocolos de capas superiores basados en CAN, tales como CANopen o Devicenet proveen los servicios adecuados para una transmisión fragmentada, limitando el tamaño del mensaje a pocos bytes, por otro lado, esta fragmentación garantiza el tiempo de latencia mas corto posible de acceso al bus

por parte de los mensajes de más alta prioridad. Sin embargo, el tamaño más corto posible del mensaje es particularmente importante cuando la transmisión tiene lugar en un ambiente bastante distorsionado porque la probabilidad de ocurrencia de daño que pueda sufrir una trama se incrementa proporcionalmente con su tamaño. Así, el tamaño corto de los mensajes del protocolo CAN permite transmisiones de datos aún en un ambiente electromagnético extremadamente dificultoso y con una alta tasa de distorsión.

Alta integridad de datos, muy corto tiempo de recuperación de error

CAN permite detección de mensajes corruptos con muy alta probabilidad gracias a que cuenta con varios mecanismos complementarios de detección de errores. La detección de errores consecuentemente permite una retransmisión automática de un mensaje transmitido o recibido en forma incorrecta. A diferencia de los protocolos orientados al nodo, tanto la detección y señalización de errores, como su correspondiente corrección toman lugar en un periodo muy corto de tiempo.

Coherencia de datos en la red de datos

Un aspecto fundamental de la integridad de datos de un sistema distribuido es su coherencia en toda la extensión de la red. Frecuentemente las operaciones de varios nodos en un sistema de control de procesos necesitan ser sincronizados. Esto es posible solo cuando los mensajes de datos o de sincronización de todos los nodos relevantes son simultáneamente y correctamente recibidos por el resto de nodos. Por lo que mensajes dañados localmente deben ser dados a conocer a todos los nodos como “no válidos”. CAN garantiza esto a través de un mecanismo de señalización de errores definido en este protocolo.

Detección y desactivación de nodos defectuosos

El protocolo CAN monitorea funciones específicas en la comunicación de nodos de tal manera que un nodo defectuoso no pueda afectar continuamente la transmisión de datos. Cuando en un nodo se excede la tasa promedio de error predefinida, se limita el acceso del nodo afectado a la red, o se lo desconecta de la misma.

Cortos tiempos de latencia para mensajes de alta prioridad

El esquema de acceso empleado en CAN, el corto tamaño del mensaje así como el principio de mensajes priorizados, permiten a mensajes de alta prioridad acceder al bus con un tiempo de latencia muy corto. El tiempo de latencia máximo para el mensaje de prioridad más alta de una red CAN con mensajes de 8 bytes es de 130 tiempos de bit. Tomando medidas adicionales se pueden asegurar tiempos de latencia definidos para mensajes de alta prioridad.

Estandarización Internacional

Los estándares internacionales ISO-WS 11898 1era, 2da y 3era parte especifican a CAN como un protocolo de 2 capas para comunicación interna de vehículos y sus aplicaciones de alta y baja velocidad respectivamente [ISO99-1, -2, -3], en ISO-WS 11898 4ta parte [ISO 00-4] se especifica la Comunicación Time-Triggered¹, como extensión del protocolo CAN. En [ISO98-1] se especifica un plan de pruebas de conformidad para cualquier implementación CAN, de acuerdo a ISO-WS 11898.

Estándares de protocolos superiores de aplicación industrial basados en CAN fueron especificados por CiA² como un estándar eficiente de 7 capas CAL (CAN Application Layer) [CIA-200] y por el estándar CANopen [CIA 30x-40x] basado en CAL. El proceso de estandarización y desarrollo del estándar DeviceNet [DNSP98] son conducidos por la OVDA (Open DeviceNet Vendors Association). CANopen así como DeviceNet son aplicados como estándares europeos (EN 50 325-4 y EN 50 325-2 respectivamente). La agrupación SAE recomienda el uso de CAN en el campo de los vehículos motorizados comerciales y define un protocolo específico de 7 capas en conjunto con un perfil apropiado de aplicación.

¹ El mismo que será revisado en el capítulo 3.

² "CAN-in-Automation". Organización internacional que apoya y desarrolla al protocolo CAN y a protocolos de capa superior basados en el mismo.

CAPITULO 2

PROTOCOLO CAN

2.1 CAPA ENLACE

2.1.1 PRINCIPIO DE ARBITRAJE DEL BUS

En redes CAN, los nodos acceden al bus de acuerdo a un esquema no centralizado basado en contención. Es posible que varios nodos empiecen la transmisión de una trama simultáneamente ya que las solicitudes de transmisión de una trama, realizadas por procesos de aplicación del nodo, generalmente ocurren asincrónicamente. Sin embargo no es permitido que un nodo inicie una transmisión mientras el bus no esté libre. Un nodo cualquiera considera que el bus está libre si el así llamado “campo de interrupción” de sus tramas no ha sido interrumpido por un bit dominante. Esto es después de una secuencia consecutiva de al menos 11 bits de nivel recesivo al final de la trama. Una trama pendiente de ser transmitida durante la transmisión de otra será iniciada a partir del primer bit después del campo de interrupción.

La metodología utilizada es CSMA/CD+AMP (Carrier Sense Multiple Access, with Collision Detection and Arbitration on Message Priority) Acceso múltiple con Escucha de Portadora, Detección de Colisión y Arbitraje basado en la Prioridad del Mensaje. Las siguientes secciones describen en detalle este tipo de acceso al medio.

Cuando varios nodos empiezan simultáneamente una transmisión, el conflicto en el bus es resuelto por un “Proceso de Arbitraje” no destructivo basado en contención sobre el “Campo de Arbitraje” de la trama CAN. El campo de arbitraje está compuesto por el “Identificador de Trama” y por el “Bit de Solicitud Remota de Transmisión” RTR (Remote Transmission Request) el cual es usado para diferenciar entre una “Trama de Datos” y una “Trama de Solicitud de Datos”. En el formato básico de la trama su identificador de trama contiene 11 bits, en el

formato extendido 29 bits. El bit más significativo del identificador es transmitido primero (el del extremo izquierdo).

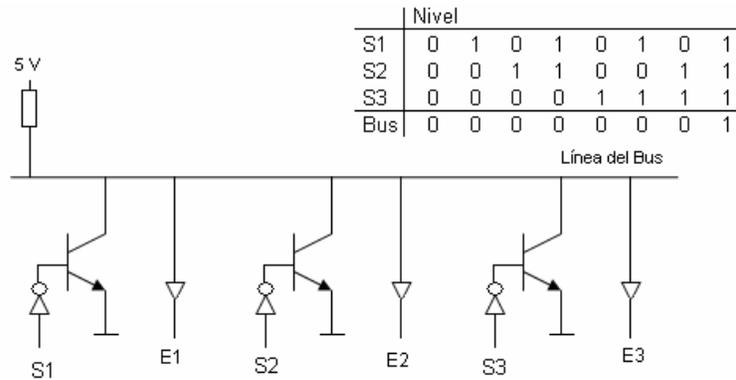


Figura 2.1: Realización del nivel dominante/recesivo basado en una Configuración AND. En el ejemplo, el nivel 0 (0V) es dominante, y el nivel 1 (5V) el recesivo

El arbitraje no destructivo está basado en la posibilidad de dos niveles físicos en el bus, un dominante y un recesivo. Estos niveles pueden ser representados fácilmente, por ejemplo por un circuito transmisor en colector abierto (configuración AND) como lo muestra la Figura 2.1. Para conseguir una lógica 1 en el bus es necesario que todos los nodos transmitan un 1, mientras que para tener una lógica 0 es suficiente que un solo nodo transmita un 0. Por tanto un nivel 0 es llamado “dominante”, y un nivel 1 “recesivo”. Estados de nivel recesivo/dominante también pueden ser representados en medios de transmisión ópticos. En este caso un nivel recesivo es representado por un estado “oscuro” (sin luz), un nivel dominante por un estado “brillo” (los métodos actualmente empleados para señalización eléctrica de los niveles de los bits en CAN son descritos en sección 2.2.5)

El bus estará en nivel recesivo mientras se encuentre desocupado. Un nodo señala el inicio de la transmisión de una trama transmitiendo un bit dominante de

“Inicio de Trama” SOF (Start Of Frame). Durante la fase de arbitraje¹ cada nodo transmisor monitorea el nivel del bus y lo compara con el nivel transmitido (Figura 2.2).

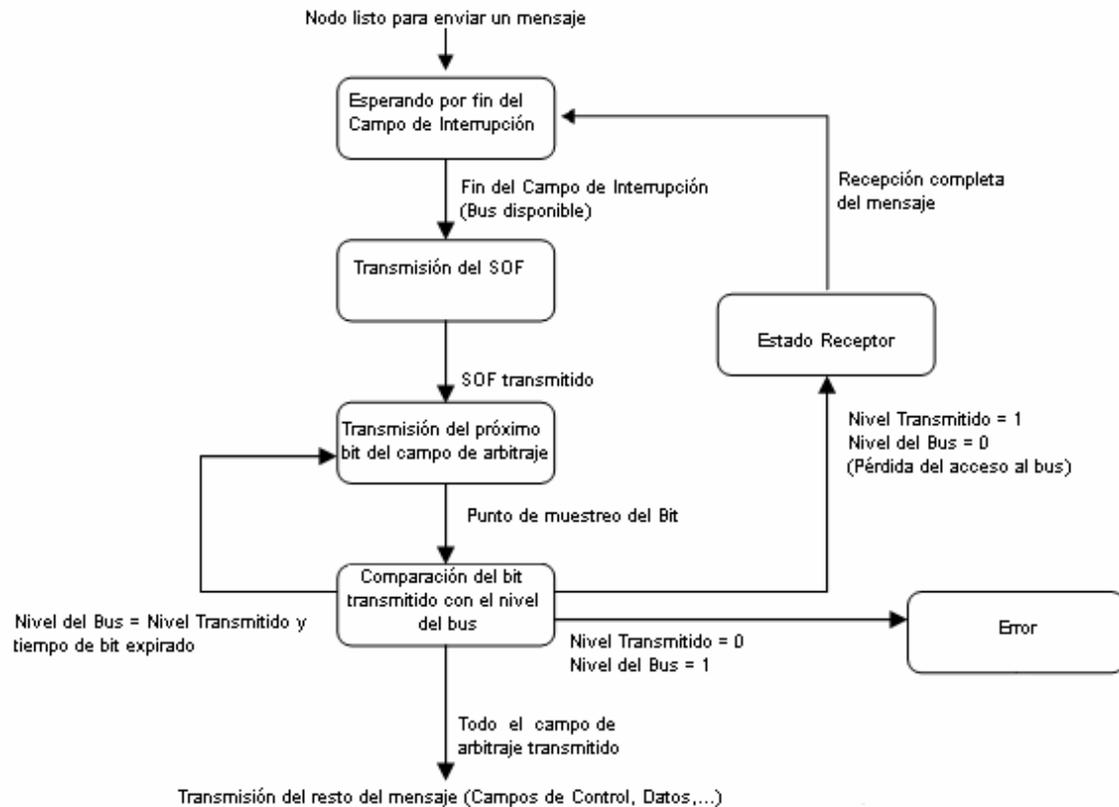


Figura 2.2: Diagrama del proceso de arbitraje del bus

Cada nodo que ha transmitido un bit recesivo y monitoreado un dominante detiene su transmisión inmediatamente y pasa a ser receptor de la trama transmitida por otro nodo. La Figura 2.3 ilustra la secuencia de un proceso de arbitraje entre tres nodos.

¹ Etapa en la que un nodo interesado en ganar el derecho transmitir envía su campo de arbitraje sobre el bus, por medio del cual compite por éste.

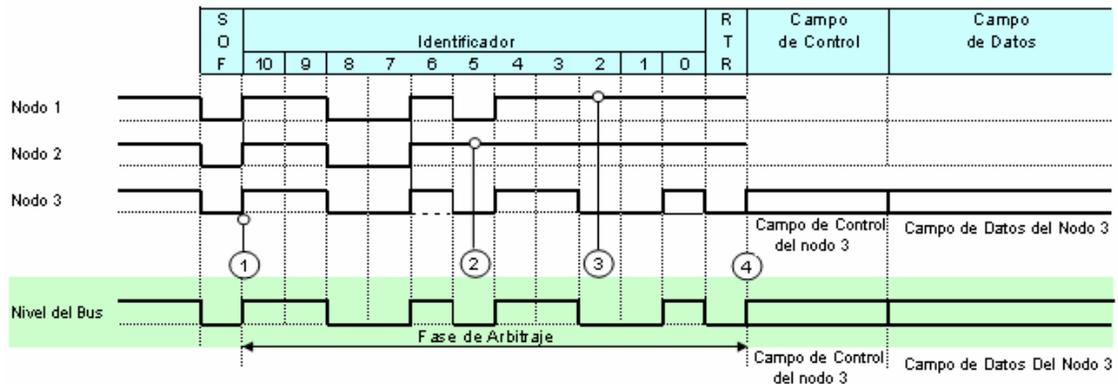


Figura 2.3: Ejemplo de un proceso de arbitraje de acuerdo al Protocolo CAN. Los nodos 1, 2 y 3 inician el proceso al mismo tiempo (1). El nodo 2 pierde acceso al bus al tiempo (2), el nodo 1 lo propio al tiempo (3). Ambos nodos detienen su transmisión pero continúan recibiendo; solo el nodo 3 continúa con el proceso y recibe acceso al bus al final de la fase de arbitraje. Por tanto solo éste nodo puede transmitir su trama sobre el bus.

De esta manera, el arbitraje como es usado en el protocolo CAN garantiza que solo un nodo del bus se mantendrá en el mismo cuando varios quieran acceder simultáneamente. La trama transmitida por este nodo no es destruida¹, es decir que no se desperdicia ancho de banda. Por ende el proceso de arbitraje del protocolo CAN no es destructivo.

Si la transmisión de una trama específica de datos es iniciada simultáneamente con una solicitud de la misma trama hecha por un receptor a través de una "Trama de Solicitud Remota de Transmisión", el conflicto de arbitraje no puede ser resuelto solamente por el identificador de trama (ya que son los mismos). En este caso, el bit RTR que está a continuación del identificador resuelve el conflicto de acceso al bus. En caso de que la trama solicitada por un nodo ya ha sido transmitida con la trama de datos, la correspondiente solicitud remota no será requerida más. Por esta razón, el bit RTR en una trama de datos es transmitido en nivel dominante (0) y en una trama de solicitud remota como recesivo (1). Es posible que la solicitud de la misma trama pueda ser enviada por varios nodos

¹ Se podría pensar que los campos de arbitraje de otros nodos interesados en transmitir, al mismo tiempo, podrían dañar la trama del nodo "vencedor". Esto no sucede ya que un bit recesivo es sobre-escrito en el bus por otro dominante, de mayor prioridad, que se presentó al mismo tiempo durante este proceso.

simultáneamente, por eso las tramas de solicitud deben tener el mismo código de longitud de datos en todo el sistema, de otra manera podría ocurrir una colisión irresoluble.

En definitiva, el principio de arbitraje está basado en una comparación del nivel de bit transmitido por el nodo con el nivel monitoreado en el bus realizado por cualquiera de los nodos transmisores. Durante la transmisión del campo arbitraje, el transmisor comprueba en cada bit si todavía está autorizado para transmitir o si está transmitiendo otra estación de mayor prioridad, los nodos transmisores con mensajes de menor prioridad se convierten inmediatamente en receptores y posteriormente repetirán su intento de emisión.

2.1.2 FORMATOS DE TRAMA

El protocolo CAN distingue cuatro tipos de tramas:

1.- Trama de Datos, que es transmitida a uno o varios receptores bajo la dirección de la fuente (transmisor).

2.- Trama Remota, a través de la cual los nodos receptores del bus pueden solicitar a un nodo transmisor la transmisión de una trama de datos del mismo identificador de la remota.

3.- Trama de Error, que es usada para señalar un error detectado por un nodo del bus (transmisor o receptor) y destruir la trama defectuosa.

4.- Trama de Sobrecarga, es usada para proveer un retardo extra entre dos tramas de datos o de solicitud remota, también sirve para señalar una condición específica de error.

Tramas de Datos y Tramas Remotas son separadas de sus tramas precedentes por un intervalo de tiempo de nivel recesivo en el bus de al menos 3 bits (Campo de Interrupción).

De acuerdo a la especificación CAN [ISO99-1] se especifican dos formatos diferentes pero compatibles de tramas: El llamado “Formato Básico” con un identificador de 11 bits y el “Formato Extendido” con un identificador de 29 bits (Ver sección 2.1.5).

2.1.2.1 Trama de Datos

Una trama de datos (Figura 2.4) consiste de los siguientes campos de bit: Inicio de trama (SOF), Campo de arbitraje, Campo de Control, Campo de Datos, Campo CRC, Campo de Acuse de Recibo (ACK) y Campo de Fin de Trama (EOF).

Bit de Inicio de Trama (SOF)

Este bit marca el inicio de una trama de datos o de una trama remota y es representado por un bit dominante. A un nodo del bus se le permite empezar el arbitraje del bus cuando éste se encuentra libre o después de que se ha desocupado. El fin de una actual transmisión de trama es indicado por una sucesión mínima de 11 bits recesivos¹.

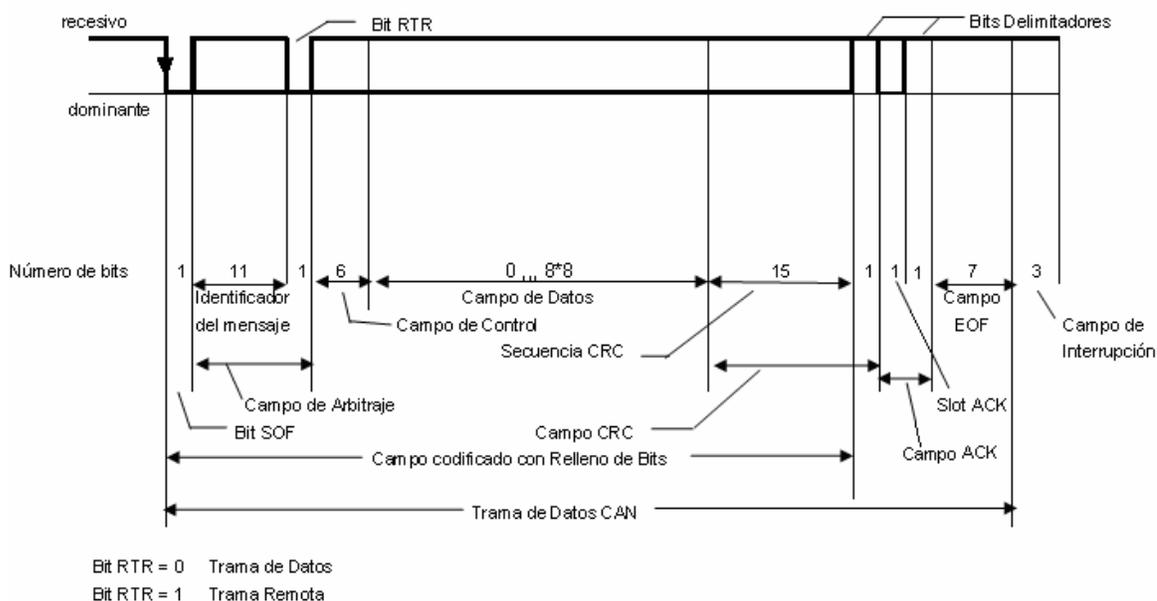


Figura 2.4: Formato básico de una Trama de Datos y de una Trama Remota

¹ Bit Delimitador ACK (1) + Campo EOF (7) + Campo de Interrupción (3).

Todos los nodos deben sincronizarse con la transición¹ causada por el bit inicial del nodo que empezó primero el arbitraje del bus (Ver Sección 2.2.1.2).

Campo de Arbitraje

El campo de arbitraje (Figura 2.5) consiste del campo identificador y del bit de Solicitud Remota de Transmisión RTR (Remote Transmission Request). Una trama es identificada y su prioridad definida por el identificador de trama. En el formato básico de trama la longitud de su identificador es de 11 bits. Esto significa que 2048 (2^{11}) tramas² diferentes son distinguibles en el formato básico del sistema CAN. Estos bits son transmitidos en orden desde el bit 10 al bit 0. El bit más significativo es el bit 10.



Figura 2.5: Formato del Campo de Arbitraje

Ambas tramas, de datos y remota, son identificadas por el bit RTR. Como anteriormente se explicó, en caso de un proceso de arbitraje simultáneo entre una trama de datos y otra de solicitud de la misma trama (Trama RTR) es razonable priorizar a la trama de datos, es por esto que el Bit RTR en una trama de datos es transmitido dominantemente y en una trama RTR recesivamente.

Campo de Control

El Campo de Control de acuerdo a la especificación CAN [ISO99-1]³ consiste de 6 bits, incluyendo el Código de Longitud de Datos DLC (Data Length Code). El primer bit, Bit Identificador de Extensión IDE (Identifier Extension), distingue entre tramas de formato básico y extendido. Para tramas de formato básico

¹ Instante de tiempo en el que se detecta un cambio de nivel de la señal de recesivo a dominante o viceversa.

² En el formato extendido, mencionado más adelante, el identificador consta de 29 bits con lo que 512 millones (2^{29}) de mensajes son posibles.

³ La más completa y actualizada junto con ISO99-2 e ISO99-3.

(identificador de 11 bits) este bit es transmitido dominantemente. Por tanto, colisiones entre una trama de formato básico y otra de formato extendido, ambas con el mismo identificador básico¹, son resueltas de tal manera que la trama de formato básico prevalece sobre la extendida. El segundo bit (r0) es reservado para futuras extensiones del protocolo CAN y es transmitido dominantemente hasta que su función sea definida. Los receptores aceptan ambos niveles, dominante y recesivo, como bit reservado².

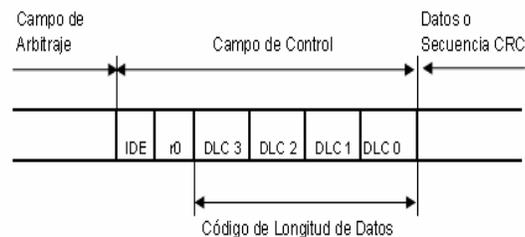


Figura 2.6: Formato del Campo de Control (Formato Básico)

Con los últimos cuatro bits del campo de control (Figura 2.6), se indica el número de bytes transmitidos a continuación en el campo de datos. Un valor de 0 corresponde a 0 bytes de datos, un valor de 8 corresponde a 8 bytes de datos. De acuerdo al modelo de referencia CAN, valores más grandes de 8 son posibles y pueden ser usados para propósitos especiales. En el presente estudio se considera 8 bytes, caso mayormente empleado, como máximo.

Campo de Datos

Este campo contiene los datos a ser transferidos dentro de una trama CAN y puede contener de 0 a 8 bytes. El bit más significativo es transmitido primero.

Campo CRC

El campo CRC (Figura 2.7) consiste de una secuencia de chequeo de 15 bits y un bit delimitador transmitido recesivamente. La secuencia CRC es utilizada para

¹ Primeros 11 bits del identificador de un identificador de trama extendida.

² Este bit puede cambiar de nivel, de tal manera que se mantenga la regla de relleno de bit (ver la sección 2.1.2.7).

que un receptor pueda saber si la trama recibida fue afectada o no por perturbaciones del medio. La secuencia de chequeo de trama es derivada de un Código de Redundancia Cíclica que funciona eficientemente en tramas con menos de 127 bits. Los coeficientes del polinomio a ser dividido (en el cálculo de la secuencia CRC) están dados por los campos de trama sin relleno¹ consistentes de: SOF, campo arbitraje, campo control y campo de datos (Figura 2.4) extendidos por coeficientes de valor 0.

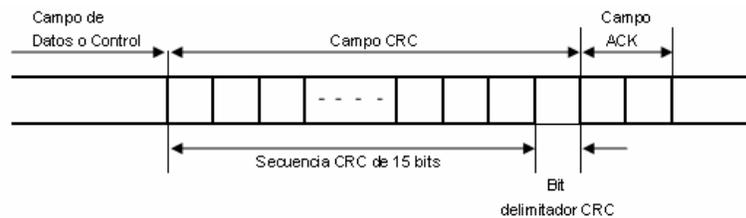


Figura 2.7: Formato del Campo CRC

Campo de Acuse de Recibo (ACK)

El campo de 2 bits de Acuse de Recibo (ACK) (Figura 2.8) consiste del así llamado “Slot² de Acuse de Recibo” seguido por el bit “Delimitador de Acuse de Recibo”. En el campo de ACK, el transmisor envía dos bits recesivos.

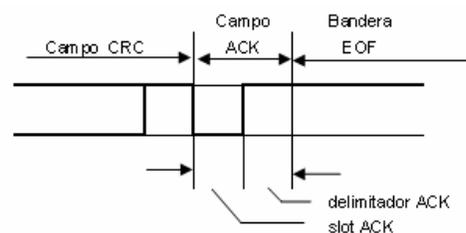


Figura 2.8: Formato del Campo de Acuse de Recibo o ACK

El protocolo CAN está basado en el principio de “estación-neutral, acuse de recepción positiva” [DaUn92]. En el que el transmisor de una trama espera por el ACK de recepción correcta de al menos un nodo receptor. Por tanto cualquier

¹ Se refiere a la “Regla de Relleno” aplicada en el protocolo CAN, ver sección 2.2.7

² Intervalo definido de tiempo, en estos casos generalmente comparable a 1 tiempo de bit.

receptor que ha recibido correctamente una trama lo reporta al transmisor de la misma sobre-escribiendo el bit recesivo recibido del transmisor por un bit dominante durante el slot ACK. Con un bit dominante durante el slot de ACK el transmisor se da por enterado que al menos uno de los nodos ha tenido recepción correcta de la trama transmitida. Esta información es usada solamente por el proceso de confinamiento de errores (ver sección 2.1.4).

Puesto que con el mecanismo de ACK solamente se indica la correcta recepción de una trama transmitida, este mecanismo no es suficiente para garantizar coherencia de datos en todo el sistema. Por tanto, cualquier error detectado por cualquier nodo es señalado enviando una “trama de error” (ver sección 2.1.2.3).

Un nodo de un bus que detecta un error de transmisión por medio de chequeo CRC subsecuentemente señala esto solamente después del campo ACK. Así, es posible que un transmisor que inicialmente recibió un ACK por una trama transmitida supuestamente libre de errores luego reciba una trama de error, transmitida por un nodo perturbado localmente, con la que se le pide que re-transmita dicha trama.

Un bit delimitador ACK recesivo siempre va después del slot ACK. Esto es requerido para distinguir un ACK positivo (bit dominante durante el slot ACK) de un posible inicio simultáneo de una trama de error, como se explicó en el párrafo anterior.

Bandera de Fin de trama (EOF)

Cada trama de datos y remota es delimitada por una secuencia de 7 bits recesivos. Juntos con el igualmente recesivo delimitador ACK resulta en una secuencia total de 8 bits recesivos al final de la trama remota o de datos.

Es considerado que en el protocolo CAN un error de transmisión es señalado a todos los otros nodos del bus antes del fin de la trama afectada, esto permite una señalización simple del error en ésta.

Puesto que un nodo totalmente de-sincronizado interpretaría un bit dominante de ACK como un bit SOF y no sería capaz de detectar un error de relleno sino hasta el quinto bit del campo EOF, es requerido al menos un bit adicional. Por tanto para detectar éste error un sexto bit debería ser necesario en el campo EOF. Sin embargo para propósitos de validación de tramas (ver sección 2.1.2.6), un error detectado en el último bit del campo EOF todavía debe poder ser señalado, por ello se requiere un séptimo bit que cumpla esta función. Esto resulta en un tamaño de 7 bits para el campo EOF, y junto con el bit delimitador ACK recesivo una secuencia de 8 bits recesivos. Por tanto una secuencia de 8 bits recesivos indica que una trama ha sido transmitida completamente libre de errores

2.1.2.2 Trama Remota

Con esta trama cada nodo puede solicitar al nodo responsable el inicio de la transmisión de una trama específica. La trama solicitada es especificada por el identificador transmitido con la trama remota. Mientras que una trama específica puede ser solicitada por todos los nodos receptores, solamente un nodo debe ser responsable de transmitir la trama solicitada.

El formato de una trama remota corresponde a aquella trama de datos ilustrada en la Figura 2.4 con la diferencia de que el bit RTR es transmitido recesivamente en una trama remota y dominantemente en una trama de datos. Así, una trama remota que compite por el acceso al bus con una trama de datos del mismo identificador pierde en éste proceso. Además, el campo de datos de una trama remota esta vacío. El código de longitud de datos en la trama remota debe corresponder a aquel de la trama de datos solicitada.

El principio del ciclo de solicitud de datos es ilustrado en la Figura 2.9. El nodo A solicita la transmisión de la trama con el identificador "Identificador_X" en una trama remota. El Nodo B es responsable de transmitir esta trama y proveerla en una trama de datos al nodo A así como a todos los otros nodos interesados en ella.

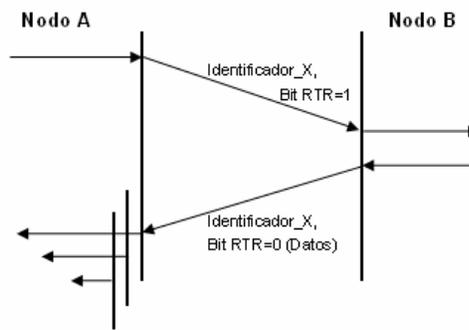


Figura 2.9: Principio del Ciclo de Solicitud de Datos

Dependiendo de la capacidad del nodo B, la trama de datos es automáticamente transmitida. Este caso se ilustra en la figura 2.9.

2.1.2.3 Trama de Error

La detección de cualquier error durante la transmisión o recepción de una trama de datos o remota es señalado por una trama de error que intencionalmente viola la regla de relleno de bits y causa que el transmisor de la trama repita la transmisión de la misma. La detección de un error durante la transmisión o recepción de una trama de sobrecarga o de error también causa la transmisión de una nueva trama de error.

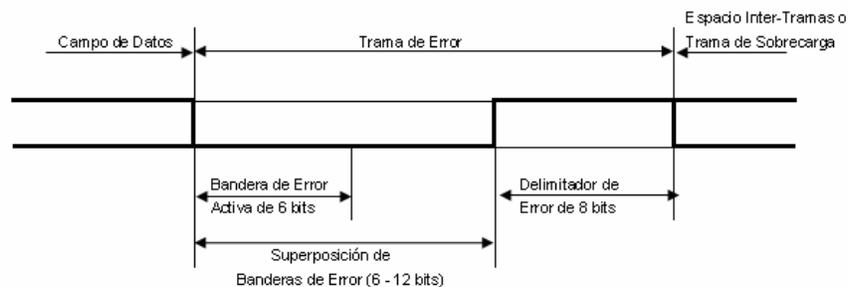


Figura 2.10: Formato de una Trama de Error Activa por un error detectado en el Campo de Datos de una trama

Una trama de error consiste de dos campos de bits. El primer campo está dado por la superposición de banderas de error transmitidas por uno o varios nodos. El

segundo campo, una secuencia de 8 bits recesivos, indica el fin de esta trama (Delimitador de Error) análoga a la trama remota y de datos. La Figura 2.10 muestra el formato de una trama de error activa. Esta forma de trama es transmitida por un nodo que posee completos derechos de señalar un error detectado (“Nodo Activo de Error”¹).

Bandera de Error

Un nodo empieza la transmisión de una bandera de error inmediatamente después de detectar una condición de error², esto es en el próximo intervalo de bit a continuación de la detección; a no ser que se trate de un error CRC, en cuyo caso la transmisión de una bandera de error es iniciada en el bit siguiente al delimitador ACK para no perturbar la función de acuse de recibo, a menos que una bandera de error para otra condición de error ya haya sido iniciada.

Un nodo defectuoso que erróneamente detecta un error puede bloquear el bus ya que la transmisión de una bandera de error activa destruye la trama bajo transmisión. Por consiguiente, dos estados de nodos son diferenciados con respecto al derecho de un nodo en señalar errores. El estado “Activo” es para los nodos con tasas de detección de errores por debajo de un cierto límite y a los que se les permite señalar errores detectados transmitiendo una “Bandera de Error Activa” consistente de 6 bits dominantes. Mientras que aquellos nodos con un promedio de tasa de detección de error que excede un límite fijo son sospechosos de tener un problema local y por ende solo tienen limitados derechos al señalar errores, éstos caen en el estado de “Pasivo”. Un Nodo Pasivo señala la detección de una condición de error transmitiendo una secuencia de 6 bits recesivos (Bandera de Error Pasiva)³.

Un nodo en estado Activo que detecta una condición de error señala esto transmitiendo una bandera de error activa (“señalización primaria de error”). Ésta viola la regla de relleno de bits aplicada a todos los campos desde SOF hasta el

¹ Un nodo está en estado activo cuando su tasa promedio de detección de error está por debajo de un límite específico (Ver Sección 2.1.4).

² Error de bit, error de relleno, error de ACK, error de forma – ver sección 2.1.3.1

³ Algunos o todos estos bits recesivos podrían ser sobre-escritos por bits dominantes de otros nodos.

delimitador CRC o destruye la forma fija del campo ACK o del EOF. Como consecuencia, todos los otros nodos también detectan una condición de error e inician la transmisión de una bandera de error (“señalización secundaria de error”) por su parte. Así la secuencia de bits dominantes que en realidad pueden ser monitoreados en el bus resulta en una superposición de diferentes banderas de error transmitidas por diferentes nodos. Dependiendo del punto de tiempo en el que otros nodos detectan la condición de error esto resulta en una secuencia de entre 6 y 12 bits dominantes.

Si un nodo transmisor Pasivo detecta un error, éste trata de señalarlo transmitiendo una secuencia de 6 bits recesivos (“Bandera de Error Pasiva”). Cuando la bandera de error pasiva empieza dentro de un campo de trama que está codificado con relleno de bit, los receptores reconocen un error de relleno de bit¹. Los bits de la Bandera de Error Pasiva antes que señalar errores sirven para demorar al nodo Pasivo 6 tiempos de bit antes de transmitir de nuevo (periodo en el cual otro nodo puede ganar el acceso).

Delimitador de Error

Una trama de error es finalizada por una secuencia de 8 bits recesivos, la cual se forma de la siguiente manera: Después de transmitir una bandera de error, el nodo transmite bits recesivos y monitorea el nivel del bus hasta reconocer un bit recesivo. Luego inicia la transmisión de 7 bits recesivos más (en total transmite una secuencia de 8 bits recesivos consecutivos). Con este mecanismo un nodo es capaz de determinar si fue el primero en transmitir una bandera de error y así haber detectado primero la condición de error. Este mecanismo es la base para el confinamiento de nodos erróneos (ver sección 2.1.4).

¹ Los receptores no reconocen un error de relleno cuando la bandera de error pasiva es iniciada durante el campo de arbitraje o a menos de 6 bits antes del fin de la secuencia CRC.

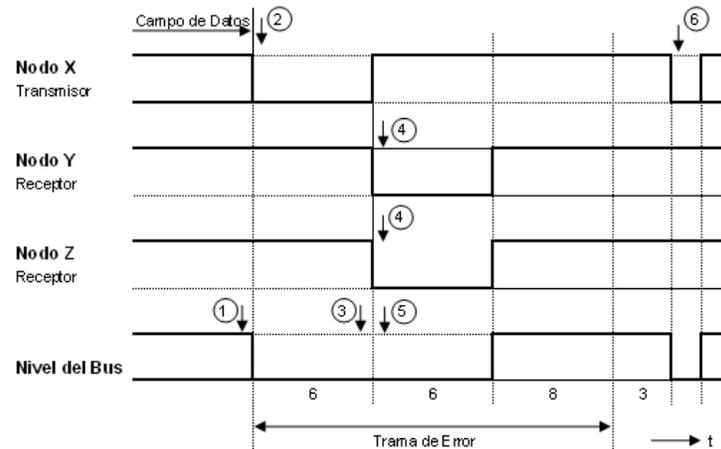


Figura 2.11: Formación de una Trama de Error cuando un error local de Transmisión ha sido detectado. El Nodo X (transmisor) detecta un error de bit en el campo de datos el el punto de tiempo (1). Este señala el error al inicio del próximo bit transmitiendo una Bandera de Error Activa (2). Después de recibir el sexto bit de la Bandera de Error (3) los otros nodos del bus (nodos Y, Z) detectan por su parte un error de relleno y también empiezan a transmitir una Bandera de Error (4) con el próximo bit. Instante en el que el nodo X intentaba transmitir el primer bit recesivo de su delimitador de error, el cual no tiene efecto inmediato debido a las siguientes señalizaciones secundarias de error de los nodos Y y Z. Esto resulta en una Bandera de Error con una longitud total de 12 bits. Después de la transmisión de sus propias Banderas de Error, los nodos Y y Z empiezan a transmitir un nivel recesivo en el bus (de sus respectivos delimitadores de error). Esto le indica al nodo X (5) que fue el primero en señalar el error. El primer bit recesivo detectado en el bus por el nodo X junto con los siete recesivos siguientes forman el delimitador de error de un total de 8 bits. Después de los tres bits del campo de Interrupción, el transmisor puede iniciar la re-transmisión de su trama de datos en un nuevo proceso de arbitraje (6). Posiblemente compitiendo con transmisores de otras tramas tramas.

La Figura 2.11 muestra la formación de una trama de error en caso que un transmisor activo de error haya detectado un bit erróneo. Como ejemplo adicional, la Figura 2.12 ilustra la formación de una trama de error en caso que un nodo activo de error haya detectado un error local por chequeo CRC.

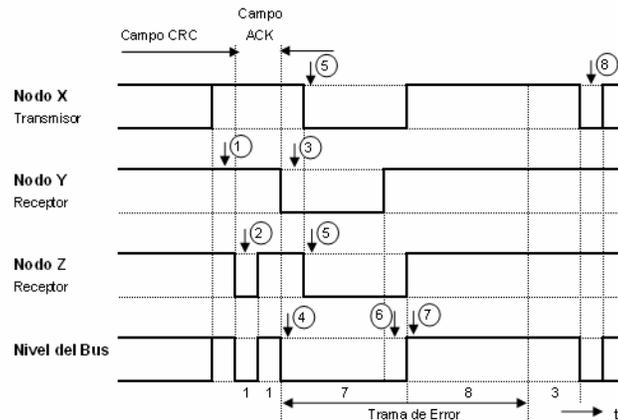


Figura 2.12: Formación de una trama de error cuando un error local de recepción ha sido detectado. El nodo Y detecta un error por chequeo de CRC (1). El nodo Z ha recibido la trama correctamente confirmando esto en el slot ACK (2). Ya que el mecanismo de acuse de recibo (ACK) debe ser preservado, el nodo Y no señala el error detectado sino hasta después del campo ACK (3). Los nodos X y Z detectan un “error de forma” en el punto de tiempo (4) ya que ellos estaban esperando el delimitador final recesivo. Entonces ellos señalan el error detectado en el próximo bit (5). Así, una Bandera de Error Activa es formada en el bus con una longitud total de 7 bits dominantes. El nodo Y detecta en el punto (6) la superposición de banderas de error y sabe así que fue el primero en señalar el error. Después de completar el octavo bit del delimitador así como los tres bits del Campo de Interrupción, el nodo transmisor puede iniciar la re-transmisión de la trama con un nuevo proceso de arbitraje.

En el ejemplo 1 (Figura 2.11) el tiempo entre la detección de un error y el inicio de una retransmisión (“Tiempo de Recuperación de Error”) es de 23 bits, en el ejemplo 2 (Figura 2.12) este tiempo es 20 bits. Comparado a éste, los tiempos de recuperación de error provisto por otros protocolos de bus de campo son mucho más grandes. En estos protocolos, una trama afectada es retransmitida solamente después de que haya pasado un periodo predefinido de tiempo sin recibir ACK positivo (principio del control pasivo de errores). Una importante ventaja adicional del método de señalización de errores del protocolo CAN es que

no ocurren “Cuellos de Botella de Prioridad”¹ cuando una trama afectada tiene que ser retransmitida. Esto es porque la retransmisión de una trama requiere un nuevo proceso de arbitraje. Así el acceso preferencial de tramas de mayor prioridad activado en este proceso es preservado.

2.1.2.4 Trama de Sobrecarga

Esta trama es proporcionada para solicitar un retraso de la próxima trama de datos o remota por el receptor de un nodo (“Trama de Sobrecarga Solicitada”) o para señalar ciertas condiciones de error (“Trama de Sobrecarga Reactiva”) relacionadas al campo de interrupción.

Las tramas de sobrecarga reactiva son transmitidas después de la detección de las siguientes condiciones de error:

- Detección de un bit dominante durante los primeros dos bits del campo de interrupción (ver sección 2.1.2.5). Un bit dominante detectado en el tercer bit del campo de interrupción es interpretado como un SOF.
- Detección de un bit dominante en el último bit del campo EOF por un receptor, o detección de un bit dominante por un receptor o transmisor en el último bit de un delimitador de trama de error o de sobrecarga.

Una trama de sobrecarga puede ser considerada como una forma especial de trama de error y es similar a ésta pues está compuesta por una bandera de sobrecarga y de un delimitador de sobrecarga. En contraste a la transmisión de una trama de error, la generación de una trama de sobrecarga es limitada a condiciones muy específicas relacionadas al campo de interrupción. Una importante distinción adicional es que la transmisión de una trama de sobrecarga no causa la re-transmisión de una trama previa, como una trama de error lo hace.

La bandera de sobrecarga consiste de una secuencia de 6 bits dominantes consecutivos y destruye la forma fija del campo de interrupción. Como

¹ Congestión con datos de prioridad.

consecuencia, todos los otros nodos también detectan una condición de sobrecarga y por su parte transmiten una bandera de sobrecarga. Después de transmitir una bandera de sobrecarga, cada nodo monitorea el bus hasta detectar un bit recesivo, a continuación de lo cual cada nodo transmite 7 bits recesivos adicionales para completar los 8 bits del delimitador de sobrecarga.

Para retrasar tramas adicionales una trama de sobrecarga se debe iniciar en el primer bit de un campo de interrupción esperado. Como máximo se permite generar sucesivamente 2 tramas de sobrecarga para retrasar la próxima trama. Tramas de sobrecarga reactivas empiezan en el siguiente intervalo de bit.

2.1.2.5 Espacio Inter-Tramas

Tramas remotas y de datos son separadas de todas las tramas precedentes (tramas de datos, remotas, de error, de sobrecarga) por un “Espacio Inter-Tramas”. En contraste, las tramas de error y de sobrecarga son transmitidas sucesivamente, sin un espacio inter-tramas entre ellas. El espacio inter-tramas consiste de un campo de bits recesivos que representa una distancia mínima de tres bits entre tramas, “Campo de Interrupción”, seguido por un campo de longitud aleatoria de nivel recesivo (bus desocupado) (Figura 2.13) hasta que una nueva transmisión de trama sea iniciada. El espacio inter-tramas adicionalmente contiene un tiempo de impedimento de transmisión de 8 bits (“Transmisión Suspendida”) para nodos “pasivos de error” que fueron los transmisores de la trama precedente (Figura 2.14).

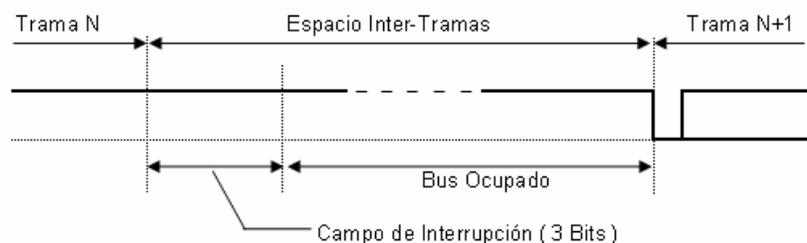


Figura 2.13: Formato de un espacio Inter-Tramas para nodos de estado Activo que fueron receptores de la trama previa

Durante el campo de interrupción es posible señalar solamente una condición de sobrecarga, para lo cual se transmite una bandera de sobrecarga. La transmisión de una trama remota o de datos no debe ser iniciada durante este tiempo.

Con los 3 bits del campo de interrupción, los 7 bits del delimitador de trama y el delimitador ACK recesivo de una trama remota o de datos, la secuencia de bits recesivos entre 2 tramas tiene una longitud mínima total de 11 bits. Por tanto, después de esta secuencia el bus está disponible de nuevo. Luego de lo cual la detección de un bit dominante es interpretada como el bit SOF de una nueva trama.

Si un nodo pendiente de transmisión detecta un bit dominante en el último bit del campo de interrupción, lo interpreta como el SOF de otro nodo. Participa en el proceso de arbitraje durante el próximo intervalo de tiempo de bit transmitiendo el primer bit del identificador de su trama, esto es sin transmitir previamente el bit SOF. Con respecto al acceso repetido al bus se aplica una regla especial para nodos pasivos de error. El derecho de transmisión de nodos en este estado es reducido en relación a todos los otros nodos activos de error por medio de un “Campo de Transmisión Suspendida” de 8 bits a continuación del campo de interrupción. Si mientras un nodo en estado pasivo espera que éste campo expire, una transmisión causada por un nodo de estado activo empieza, el primero vendrá a ser un receptor del segundo. La Figura 2.14 muestra el espacio Inter-Tramas para un nodo pasivo de error que fue el transmisor de la trama previa.

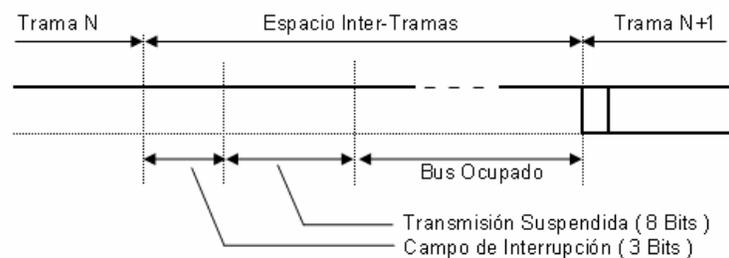


Figura 2.14: Formato de un Espacio Inter-tramas para un nodo de estado Pasivo que fue el transmisor de la trama previa

Ya que el bus solamente es considerado disponible para un nodo transmisor pasivo de error, después de completar el tiempo de transmisión suspendida, tramas de prioridad mas baja de nodos activos de error pendientes de transmitirse pueden acceder al bus, antes de que el nodo pasivo de error pueda hacerlo de nuevo. Esto asegura que un nodo posiblemente defectuoso no pueda dificultar a otros nodos la transmisión de sus tramas.

2.1.2.6 Validación de Tramas

El punto de tiempo en el que una trama es tomada para ser validada es diferente para el transmisor y los receptores de la trama.

Para un transmisor una trama es válida cuando ningún error fue señalado hasta el último bit de la bandera EOF.

Los receptores de una trama aceptan una trama como válida cuando ellos no han detectado una señal de error, es decir el inicio de una bandera de error, de otros nodos hasta el final del sexto bit de la bandera EOF. Si, debido a un error local¹, el sexto bit de la bandera EOF es dominante, el nodo receptor afectado desecha la trama. Para eliminar la ahora existente inconsistencia de datos (otros nodos pueden haber aceptado la misma trama como válida), el campo EOF de una trama CAN posee un séptimo intervalo de tiempo de bit. Durante este intervalo un receptor con un error local en el sexto bit del campo EOF puede señalarlo generando una trama de error y así causar que el transmisor retransmita la trama [DaUn92]. Si esto ocurre, el transmisor es forzado a retransmitir la misma trama. Esto significa que los receptores que ya han recibido la primera trama correctamente recibirán de nuevo la misma trama.

El séptimo bit del campo EOF no es de importancia para un receptor, esto significa que un bit dominante en el último bit de EOF no es considerado como un error de forma². Similar a un bit dominante como el último bit de un delimitador de error o sobrecarga, un receptor que detecta un bit dominante como último bit

¹ Errores locales se deben a perturbaciones internas del nodo, y errores globales a perturbaciones que se introducen en el bus y aparecen en todos los nodos.

² Ver sección 2.1.3.1

del campo EOF genera una trama de sobrecarga. Esto no causa retransmisión de la trama recibida, pero corrige un nodo posiblemente de-sincronizado.

Si ambos, receptor y transmisor, detectan un bit dominante en el último bit de EOF debido a un error global, la trama es válida para el receptor, pero no para el transmisor. Entonces el transmisor repite la trama y el receptor recibe la trama de nuevo¹.

2.1.2.7 Codificación del Flujo de Bits

En el protocolo CAN los bits de una trama son físicamente representados de acuerdo al código “No Retorno a Cero” NRZ (No Return to Zero). Esto significa que durante todo un intervalo de tiempo de bit el nivel generado de éste es dominante o recesivo.

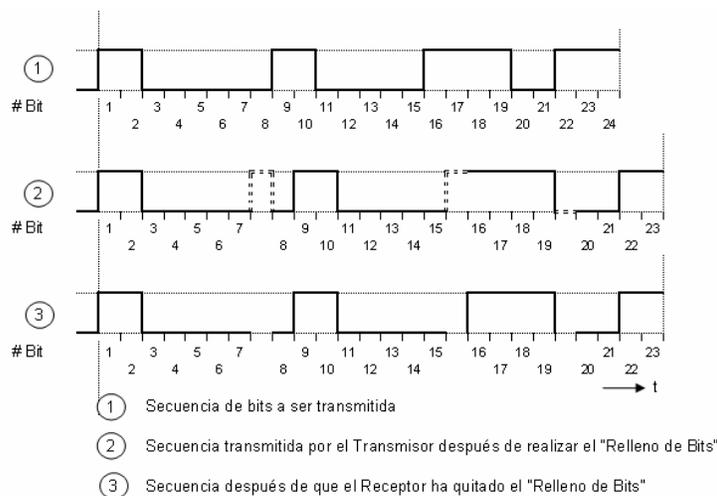


Figura 2.15: Principio del “Relleno de Bits”. En el ejemplo, el transmisor siempre inserta un bit adicional (bit de relleno) después de 5 bits consecutivos del mismo valor. El bit de relleno es complementario a la secuencia precedente de bits. Para recuperar el arreglo de bits original un receptor siempre removerá el bit siguiente a una secuencia de cinco bits de igual valor dentro de los segmentos de trama codificados con “Relleno de Bit”

¹ Esencial para detección de mensajes duplicados. Esta función es provista por protocolos superiores basados en CAN.

Por tanto en caso de largas secuencias de bits de igual nivel no están disponibles transiciones de bits para re-sincronizar los receptores. Esta es una desventaja de la codificación NRZ, la cual puede ser eliminada cuando el transmisor inserta un bit adicional de nivel complementario (“Bit de Relleno”) dentro del arreglo de bits después de una larga secuencia de bits del mismo nivel. Entonces el flujo de bits provee un número suficiente de transiciones para re-sincronizar los nodos receptores. Por supuesto, que el receptor remueve los bits insertados antes de enviar los bits a procesos posteriores. Este proceso es conocido como “Relleno de Bits” (Figura 2.15).

En las tramas remotas y de datos, los segmentos de trama: SOF, campos de arbitraje, de control, de datos, y de secuencia CRC son codificados con el método de relleno de bits. Cuando un transmisor detecta 5 (“distancia de relleno”) bits consecutivos incluyendo bits de relleno de idéntico valor en el flujo de bits a ser transmitido (como el caso de la figura 2.2.7-1), automáticamente inserta un bit complementario¹ en el conjunto de bits transmitidos. El resto de campos de bits de una trama remota o de datos (delimitador CRC, campos ACK y EOF) son de forma fija y no utilizan relleno de bits. También tramas de sobrecarga y de error son de forma fija y no se codifican con el método de relleno de bits.

La bandera de error de 6 bits de igual valor resulta a partir de una distancia de relleno de 5 bits. Ya que esta secuencia de bits no puede ocurrir en segmentos regulares de tramas con relleno con bits, una secuencia de 6 bits de igual polaridad puede ser interpretada como una violación de código intencional usada para indicar una condición de error. De tal manera que el relleno de bits no solo soporta re-sincronización de receptores proveyendo señales de transición adicionales sino que también permite señalización de errores por medio de una violación de código.

¹ Bit de nivel opuesto a los 5 bits consecutivos precedentes.

2.1.3 DETECCIÓN Y MANEJO DE ERRORES

2.1.3.1 Mecanismos de Detección

Se tienen demandas especialmente altas respecto a la seguridad de transmisión de datos, como por ejemplo en aplicaciones del protocolo CAN en vehículos. Para satisfacer estas demandas, el protocolo CAN provee varios mecanismos para detección de errores. Estos son:

- Chequeo de Bit
- Chequeo de Trama
- Chequeo de Redundancia Cíclica
- Chequeo de ACK
- Chequeo de Regla de Relleno

Chequeo de Bit

Cada nodo transmisor monitorea si el nivel del bit enviado al bus difiere del nivel que en verdad ha llegado a éste. Si el valor de bit que ha sido transmitido es diferente del valor de bit que es monitoreado, un “error de bit” es detectado. El monitoreo de bit representa así un mecanismo muy efectivo para detección de todos los errores globales y locales al transmisor.

La sobre-escritura de un nivel de bit transmitido recesivamente por un nivel dominante durante la fase de arbitraje así como durante el slot ACK no es interpretado como un error de bit. La sobre-escritura de una bandera de error pasiva (nivel recesivo) tampoco es interpretada como un error de bit por un nodo transmisor¹.

Chequeo de Trama

Los formatos posibles de trama en el protocolo CAN contienen campos de bits de forma fija específica (por ejemplo los bits delimitadores recesivos) cuya estructura es verificada por todos los nodos. Un “error de forma” es detectado cuando un campo de forma fija contiene uno o más bits no permitidos.

¹ En este caso se asume que otro nodo ha transmitido una bandera de error activa.

Un receptor que monitorea un bit dominante en el último bit de EOF o cualquier otro nodo que monitorea un bit dominante en el último bit de un delimitador de error o de sobrecarga no interpreta esto como un error de forma (ver sección 2.1.2.6).

Chequeo de Redundancia Cíclica

Para permitir a un nodo receptor verificar la integridad de los datos recibidos, el protocolo CAN aplica el principio de chequeo de redundancia. Este método generalmente aplicado en sistemas de transmisión de datos garantiza una probabilidad muy alta de detección de errores y una “Probabilidad de Error Residual”¹ muy baja. En especial, el chequeo de redundancia cíclica proporciona la detección de errores globales.

En chequeo de redundancia cíclica, la información del encabezado de trama y los datos a ser transmitidos son representados como un “Polinomio de Trama” el cual será dividido por un “Generador Polinomial” definido. El residuo de esta división binaria es la secuencia CRC (Código de Redundancia Cíclica) de 15 bits transmitida sobre el bus como parte de la trama. En el nodo receptor el polinomio de trama recibido es también dividido para el mismo generador polinomial. Si la trama fue transmitida libre de errores, el residuo será nulo.

Los coeficientes del polinomio de trama están formados por los valores de bit (0/1) del conjunto de bits sin relleno, el cual consiste de: bit SOF, campos de arbitraje, control y datos, extendidos por 15 coeficientes menos significantes de valor 0. La secuencia de chequeo de 15 bits usada en el protocolo CAN está basada en el Código BCH (Bose Chaudhuri-Hocquenghem), uno de los códigos cíclicos que mejores resultados presenta en tramas de longitudes menores a 127 bits. El generador polinomial de grado 15 aplicado está dado por:

$$X^{15} + X^{14} + X^{10} + X^8 + X^7 + X^4 + X^3 + 1$$

¹ Probabilidad de que un mensaje recibido contenga errores no detectados.

Chequeo de ACK

Un transmisor espera que en el slot ACK, al menos, un receptor confirme la correcta recepción de una trama transmitida. La falta de confirmación (es decir el slot ACK con nivel recesivo en vez de dominante) es interpretada por el transmisor de la trama como un error de ACK.

Chequeo de regla de relleno

Cada nodo encuentra una violación de la regla de relleno de bit (“Error de Relleno”) tan pronto como detecta un sexto bit consecutivo de igual nivel en un campo de trama codificado por el método de relleno de bit. Consecuentemente, cada nodo que detecta una bandera de error interpreta esto como una violación de la regla de relleno de bit y por tanto genera una trama de error automáticamente.

2.1.3.2 Capacidad de Detección de Errores

La capacidad de detectar errores de transmisión depende en gran medida de los mecanismos de detección de errores aplicados y varía extensamente entre los diferentes protocolos.

Las condiciones de interferencia electromagnética bajo las cuales es operado un sistema de transmisión de datos así como la habilidad del sistema para detectar errores de transmisión son decisivas para la evaluación de la integridad de datos de un sistema. Una medida estadística para la integridad de datos transferidos a través de sistemas de transmisión es la “Probabilidad de Error Residual”, esta medida especifica la probabilidad de tramas dañadas no detectables.

Errores de transmisión pueden ocurrir cuando por ejemplo la señal eléctrica de transmisión es alterada debido a interferencia electromagnética. La susceptibilidad de error de un sistema de transmisión de datos puede estar caracterizada por parámetros estadísticos tales como “Tasa de Error de Trama” y “Probabilidad de Error de Bit”. La tasa de error está dada por la relación de tramas afectadas respecto al número total de tramas transmitidas durante un tiempo de monitoreo, describiendo la probabilidad de que una trama sea alterada.

La probabilidad de error de bit por su parte especifica la probabilidad de que un bit de una trama transmitida haya sido dañado.

Basado en la combinación de diferentes mecanismos de detección de error descritos en la sección 2.1.3.1, la capa enlace CAN garantiza un grado muy alto de detección. Todos los errores globales son detectados por el proceso de monitoreo de bit realizado por el transmisor de una trama. Esto se refiere a todas las perturbaciones que se introducen en el bus por interferencia electromagnética y que aparecen en todos los nodos. Adicionalmente, el transmisor también detecta todas las perturbaciones locales.

Aquellos errores que solamente ocurren localmente en receptores individuales son detectados por la secuencia de chequeo CRC de 15 bits en los receptores, la misma que garantiza una "Distancia de Hamming" de 6^1 . En protocolos convencionales de bus de campo la distancia de Hamming es usualmente 4 o menos.

Un análisis muy detallado de la capacidad de detección de error del protocolo CAN (formato básico de trama) es reportado en [UnMa90]. En este análisis se muestra cómo errores indetectables afectan la capacidad de detección de errores. Aquellos errores indetectables causados por daño de relleno de bits existen por ejemplo:

- Cuando un primer error de bit causa la pérdida de un bit de relleno, y un segundo error de bit causa la generación de un bit de relleno, así que se mantiene la longitud de la trama.
- Cuando la diferencia entre la secuencia CRC transmitida y recibida es un múltiplo del generador polinomial.
- Cuando los errores no aparecen globalmente.

Como resultado del análisis reportado en [UnMa90] el valor máximo de probabilidad de error residual está dado por:

¹Hasta 5 errores distribuidos arbitrariamente en una trama serán detectados.

$$\text{Probabilidad de Error Residual} < 4,7 * 10^{-11} * \text{Tasa de error de trama}$$

Con esta regla el número de errores de transmisión indetectables durante el tiempo operativo de una red CAN puede ser estimada por una tasa de error de trama dada (determinada por el ambiente perturbador de una red, promedio de carga de bus, velocidad de transmisión y longitud de trama).

Cuando el formato extendido de trama es usado, se obtiene un valor menos favorable para la probabilidad de error residual debido a la longitud más grande de la trama.

2.1.3.3 Consistencia de datos

Dentro de una red CAN está garantizado que una trama sea simultáneamente aceptada por todos los nodos o hasta por ninguno. Si un nodo detecta una condición de error, señala esto a todos los nodos de la red transmitiendo una bandera de error destruyendo así la trama transmitida. Con este mecanismo se garantiza una consistencia de datos en toda la red, la cual representa una característica especialmente importante del protocolo CAN.

2.1.3.4 Tiempo de recuperación de Errores

Un nodo señala un error transmitiendo una bandera de error. Cuando un error de bit, de relleno, o de ACK es detectado, se inicia la transmisión de una bandera de error por el nodo detector del error en el próximo intervalo de bit. Cuando un error CRC es detectado, se inicia la transmisión de una bandera de error con un retardo de 2 intervalos de tiempo de bit después del slot ACK para conservar el mecanismo de acuse de recibo. Con este principio de señalización de error, el protocolo CAN brinda el tiempo más corto posible para re-transmitir la trama afectada comparado a otros protocolos de bus de campo, en los que la re-

transmisión de una trama afectada generalmente no es iniciada hasta que un tiempo de respuesta de un ACK específico haya terminado.

Los tiempos de recuperación de errores¹ de las situaciones de error mostradas en las figuras 2.11 y 2.12 son 23 bits y 20 bits respectivamente. El tiempo máximo de recuperación después de una trama dañada es de 31 tiempos de bit. Este valor ocurre cuando un nodo pasivo de error detecta un error, y es debido a los 8 bits adicionales del campo de “transmisión suspendida” para nodos de estado pasivo.

2.1.4 CONFINAMIENTO DE FALLOS

Un sistema de bus serial está relacionado básicamente con el problema de que un nodo defectuoso del bus podría bloquear el sistema entero. Particularmente el principio de señalización de error aplicado en el protocolo CAN implica este riesgo ya que un nodo defectuoso podría generar continuamente banderas de error. Para eliminar este riesgo, se provee un mecanismo el cual puede detectar un nodo defectuoso y hasta desconectarlo del bus. Para este propósito, el protocolo CAN especifica un “contador de error de transmisión” y un “contador de error de recepción” para cada nodo. Cuyas cuentas son incrementadas en un determinado valor con cada transmisión o recepción de trama errónea detectada, y decrementada en 1 con cada transmisión o recepción exitosa de una trama, de tal forma que en caso de un error el incremento de la cuenta de error sea más grande que el decremento en caso de un éxito. El valor de incremento depende de si un nodo fue el primero en detectar un error. Esto puede conducir a un incremento notable de las cuentas sobre un cierto periodo de tiempo, sin embargo éstas son reducidas nuevamente durante un periodo de tiempo más largo con una tasa de error más baja. Los valores de cuentas de error son así una medida para la tasa de error de transmisión.

Si un nodo detecta un error local, la bandera de error enviada por éste provoca una subsiguiente señalización de error por parte de los otros nodos de una red.

¹ Intervalo de tiempo (en número de tiempos de bit) entre la ocurrencia de un error y el inicio de la re-transmisión de la trama afectada.

Los nodos reconocen si fueron los primeros en señalar un error al finalizar su bandera de error. Para propósitos de confinamiento de fallos, las cuentas de error de recepción o de transmisión del nodo que primero señaló un error durante la recepción y transmisión respectivamente son incrementados en una cantidad significativamente más alta (en un valor de 9 y 8 respectivamente) que las cuentas de error de los nodos que señalaron el error secundariamente. Un nodo que siempre es el primero en detectar un error por consecutivas ocasiones sobre un largo periodo de tiempo puede estar teniendo problemas consigo mismo, es decir estar defectuoso. Cuando se exceden límites específicos de cuentas de error, se toman medidas para restringir un efecto presumiblemente defectuoso del nodo. Estas medidas van desde prohibir la transmisión de banderas de error activas cuando una cuenta de error específica es alcanzada, hasta impedir a un nodo formar alguna interacción con el bus cuando un nivel aún más alto de cuenta de error es alcanzado.

Con respecto al confinamiento de error, un nodo puede estar en uno de los siguientes tres estados de error, dependiendo de los valores de sus contadores.

- Error-Activo (Estado Activo)
- Error-Pasivo (Estado Pasivo)
- Bus-Off (Estado Suspendido)

Un nodo en estado "Activo" en una red toma parte en la comunicación del bus y envía una bandera de error activa cuando detecta un error (ver sección 2.1.2.3). Esto destruye la trama que estaba transmitiendo, viola la regla de relleno de bit y previene a otros nodos de aceptar la trama errónea.

Un nodo en estado "Pasivo" ya ha acumulado una cuenta de error de transmisión o de recepción relativamente alta y ha monitoreado así una tasa de error significativamente elevada sobre un periodo más largo de tiempo. Un nodo en este estado es todavía totalmente capaz de comunicarse, sin embargo los errores detectados ya no serán señalados por una bandera de error activa. En cambio ahora, un error es señalado por una bandera de error pasiva consistente de 6 bits

recesivos que no perturba la comunicación restante del bus. Además, un nodo en estado pasivo debe esperar por 8 tiempos de bit (“campo de transmisión suspendida”, ver sección 2.1.2.5) antes de intentar re-transmitir la trama que fue detectada como errónea. Ambas medidas aseguran que un nodo que ha estado detectando una tasa de error significativamente más alta sobre un periodo de tiempo mas largo ya no pueda interferir con la comunicación restante del bus.

A un nodo en estado “Suspendido” no se le permite tener ningún tipo de influencia sobre el bus. Este nodo no transmitirá ninguna trama, ni siquiera aquellas de sobrecarga, error ni tampoco ACKs. Dependiendo de la implementación tal nodo podría recibir tramas.

La Figura 2.16 muestra el diagrama de estados de error de un nodo CAN. Después de un Reset un nodo está en el estado Activo. Si uno de las dos cuentas de error excede el valor 127, el nodo entra al estado Pasivo. Un nodo vuelve a ser Activo de nuevo cuando ambas cuentas de error caen a un nivel por debajo de 128. Un nodo es desconectado del bus y entra al estado Suspendido cuando la cuenta de error de transmisión excede 255. De este estado, un nodo puede retornar al estado Activo al re-establecer sus cuentas de error a cero después de re-configurar y re-establecer el nodo y luego que ha detectado 128 secuencias de 11 bits recesivos consecutivos.

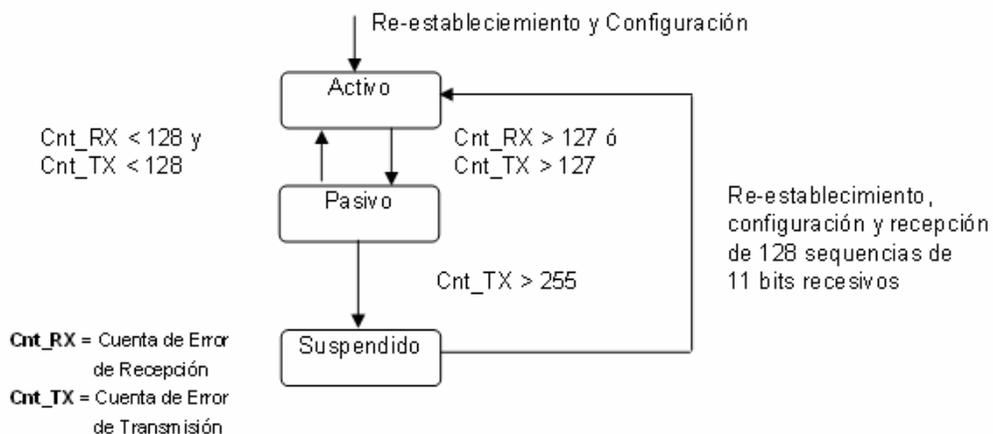


Figura 2.16: Diagrama de Estados de Error de un nodo CAN

Esta medida asegura que un nodo posiblemente erróneo no pueda perturbar la comunicación de nuevo inmediatamente después de haber sido re-establecido. Con lo que más de 128 tramas adicionales pueden ser transmitidas sin ser perturbadas aún en condiciones muy altas de carga del bus.

Modificaciones a cuentas de error de transmisión y recepción implementadas en cada chip CAN para tareas de confinamiento de fallos son dirigidas de acuerdo a reglas definidas. Más de una regla puede aplicarse para la transferencia de una trama dada.

La idea básica del conjunto de reglas definidas para confinamiento de fallos está basada en asignar mayor peso a los errores que han sido detectados primero por un nodo antes que el resto así como de considerar casos particulares de errores. Por otro lado, puede haber un incremento temporal de la tasa de error en el bus, causada por razones externas (por ejemplo una alta interferencia electromagnética momentánea), por lo que este conjunto de reglas también debe permitir la reducción de los contadores con una reducida tasa de error. Esto es logrado decreciendo la correspondiente cuenta de error para cada transmisión o recepción exitosa de trama.

De acuerdo al conjunto de reglas para confinamiento de fallos la cuenta de error de transmisión de un nodo es incrementado por ejemplo en un valor de 8 cuando el nodo detecta un error durante la transmisión de una trama, pero es decrecido solo en 1 cuando el nodo transmitió una trama exitosamente. Esto significa que una red CAN se mantiene funcional cuando en promedio se tiene una (1) trama afectada como máximo por cada 9 tramas transmitidas.

El manejo del error ocurre en el orden siguiente:

1. Se detecta el error.
2. Una trama de error será transmitido.
3. El mensaje será desechado por cada nodo de la red.
4. El contador de errores de cada nodo del bus es incrementado.
5. El mensaje afectado serán re-transmitido.

De acuerdo a [ISO99-1] para confinamiento de fallos deben aplicarse las siguientes reglas en cuentas de transmisión y recepción:

1.- Cuando un receptor detecta un error, su cuenta de error de recepción será incrementada en 1, excepto si el error detectado fue un error de bit durante el envío de una bandera de error activa o una de sobrecarga. En este caso se aplica la regla 5.

2.- Cuando un receptor es el primero en detectar un error después de transmitir una bandera de error, su cuenta de error de recepción es adicionalmente incrementada en 8. Esto sucede cuando un nodo detecta un bit dominante debido a una bandera de error secundaria sucesiva después de enviar la suya.

3.- Cuando un transmisor envía una bandera de error, su cuenta de error de transmisión es incrementado en 8. Esto sucede por ejemplo cuando un transmisor detecta un error de bit o no recibe un ACK.

En caso de la siguiente excepción el contador de transmisión se mantiene inalterado.

Si el transmisor envía una bandera de error porque ha ocurrido un error de relleno durante la fase de arbitraje, considerando que el relleno de bit debería haber sido recesivo, y ha sido enviada como recesivo, pero es monitoreado como dominante (sobre-escrito por otro nodo interesado en transmitir).

4.- Si un transmisor detecta un error de bit mientras envía una bandera de error activa o una de sobrecarga, la cuenta de error de transmisión es incrementada en 8.

5.- Si un receptor detecta un error de bit mientras envía una bandera de error activa o una de sobrecarga, la cuenta de error de recepción es incrementada en 8.

6.- Cualquier nodo tolera hasta 7 bits dominantes consecutivos después de enviar una bandera de error activa, bandera de error pasiva o bandera de sobrecarga¹. Después de detectar el 14avo bit dominante consecutivo (en caso de una bandera de error activa o una bandera de sobrecarga) o después de detectar el 8avo bit dominante consecutivo a continuación de una bandera de error pasiva, cada transmisor y receptor incrementan su respectivo contador de error en 8. Esta consideración penaliza a nodos que continuamente están detectando errores posiblemente debido a problemas consigo mismos.

7.- Después de la transmisión exitosa de una trama (es decir recepción de un bit dominante de ACK y ninguna bandera de error hasta que EOF haya finalizado) la cuenta de error de transmisión es decrementada en 1 a menos que ya fuese cero.

8.- Después de la recepción exitosa de una trama (es decir sin error hasta el slot ACK y envío exitoso del bit ACK) la cuenta de error de recepción es decrementada en 1, si estuvo entre 1 y 127; si la cuenta de error de recepción fue cero, continúa siendo cero; si era mayor que 127, la cuenta de error se establecerá entre 119 y 127.

Una cuenta de error de recepción generalmente es incrementada en 9 por cada error de recepción detectado primero (regla 1 y 2) y en 1 por cada error de recepción detectado simultáneamente con otros nodos (regla 1), y es decrementado en 1 por cada trama recibida exitosamente (regla 8). Cuando la cuenta de error sobrepasa 96 una tasa de error incrementada es indicada al controlador de host, sin embargo el nodo sigue siendo de estado Activo.

Cuando el error sobrepasa el valor 127, el nodo viene a ser de estado Pasivo. Así un nodo indicaría una tasa de error incrementada después de la 11ava² recepción errónea consecutiva y vendría a ser Pasivo después de la 15ava³ recepción consecutiva errónea. Un nodo no puede llegar a ser de estado Suspendido por

¹ En realidad se deberían esperar 6 bits dominantes consecutivos, sin embargo para propósitos de confinamiento de fallos se ha extendido a 7.

² 11ava detección errónea * 9 = 99.

³ 15ava detección errónea * 9 = 135.

medio de errores de recepción. Cuando el contador de recepción sobrepasa el valor 127, en la próxima recepción exitosa de una trama el contador de error de recepción será establecido en un valor por debajo del límite del estado Pasivo de acuerdo a la regla número 8.

La cuenta de error de transmisión es incrementada en un valor de 8 por cada error de transmisión detectado (regla 3) y es decrementado en 1 (regla 7) por cada trama transmitida exitosamente. Si el contador sobrepasa el valor de 96 una indicación de “taza de error incrementada” es establecida; si el contador sobrepasa 127 el estado del nodo cambia a Pasivo, si el contador es más grande que 255 el nodo es establecido en el estado Suspendido. En caso de que cada intento de un nodo por transmitir sea erróneo, el nodo será establecido en estado Suspendido después de 32 intentos de transmisión.

2.1.5 FORMATO EXTENDIDO DE TRAMA

Un método simple para la realización de un sistema de comunicación de datos inter-operable basado en CAN es la definición de mensajes estandarizados para todos los datos a ser transmitidos en un campo específico de aplicación. Esta propuesta es usada por ejemplo para la implementación de redes CAN en buses y camiones por SAE. Por lo que, pronto se concluyó que un identificador de mensaje significativamente más grande que el identificador de 11 bits provisto con el formato de trama básico es requerido debido al alto número de tramas a ser soportadas. Por esta razón en la especificación CAN [ISO99-1] un formato adicional, extendido, con un identificador de 29 bits fue definido. Basado en el identificador de 29 bits, hasta 512 millones de tramas son distinguibles. El tiempo de latencia mas largo relacionado a esto así como la reducción de la efectividad del protocolo e integridad de datos son tolerados para este tipo de aplicación.

De acuerdo a la especificación CAN [ISO99-1], los formatos de trama básico y extendido son compatibles, es decir tramas de ambos formatos pueden co-existir en la misma red y nodo. Los dos formatos son distinguidos por el Bit Identificador de Extensión IDE, en el campo de control (ver sección 2.1.2.1). En el formato de trama extendido, el identificador de 29 bits comprende dos secciones, el

Identificador Básico de 11 bits (ID Básico) así como un Identificador Extendido de 18 bits (ID Extendido) (Figura 2.17). En el formato extendido el ID Básico es seguido por el Bit “Substituto de Solicitud Remota” SRR (Substitute Remote Request Bit) y el bit IDE como parte del campo de arbitraje. El bit RTR del formato básico es sustituido por el bit SRR transmitido recesivamente en el formato extendido. Esto significa que una trama de datos (indicado por el bit dominante RTR) transmitida en formato básico prevalece sobre la transmisión de una trama de datos con un ID Básico idéntico transmitido en formato extendido.

El bit IDE es transmitido en el formato básico predominantemente como parte del campo de control y, recesivamente en el formato extendido como parte del campo de arbitraje. Así que también la transmisión de una trama remota en formato básico prevalece sobre la transmisión de una trama remota con idéntico ID Básico en el formato extendido¹.

El campo de control de 6 bits es ligeramente diferente para formatos básico y extendido. El campo de control del formato básico contiene el bit IDE, que es transmitido siempre predominantemente, el bit reservado r0 (transmitido predominantemente) así como el código de longitud de datos de 4 bits. La trama de formato extendido contiene 2 bits reservados r0 y r1 además del código de longitud de datos. Ambos bits reservados son transmitidos predominantemente. Los receptores aceptan bits reservados en todas las combinaciones. La Figura 2.17 muestra el formato básico y extendido de una trama.

¹ Si una trama de solicitud remota en formato básico accede simultáneamente al bus con otra de solicitud remota con el mismo ID Básico pero en formato extendido, la primera gana el arbitraje debido a que su bit IDE se transmite predominantemente.

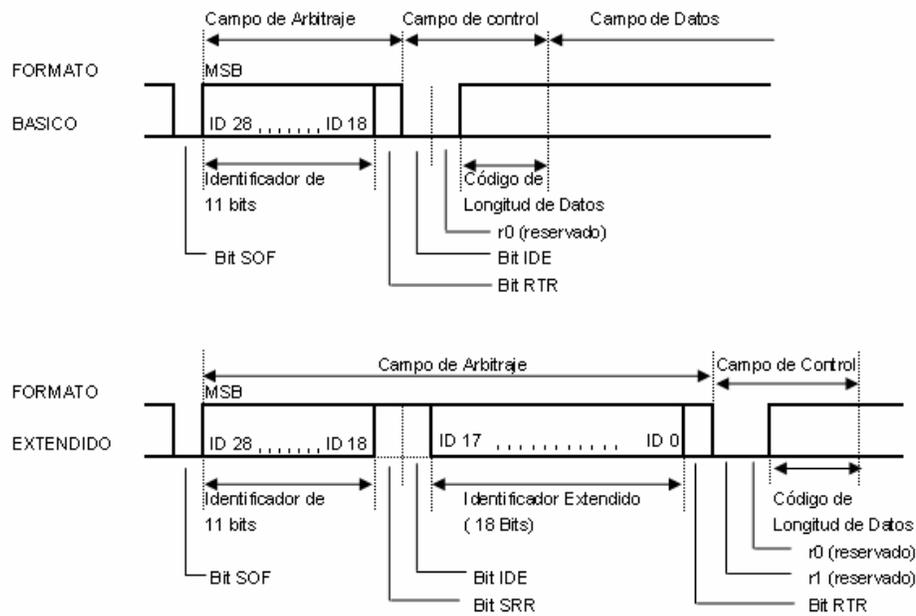


Figura 2.17: Especificación CAN [ISO99-1], Campos de Arbitraje y Control del formato básico (arriba) y del formato extendido (abajo).

2.1.6 COMUNICACIÓN TIME TRIGGERED

Bajo el método de arbitraje aplicado en CAN (CSMA/CD+AMP) el cual esta basado en prioridades, aún si un mensaje posee una alta prioridad, éste puede experimentar cierto retardo, los cuales resultan considerables en sistemas críticos de tiempo. Ya que el Protocolo CAN provee los pre-requisitos necesarios para la opción de una comunicación de este tipo, es creado entonces el Protocolo TTCAN (Time Triggered CAN), como anexo del Protocolo CAN [ISO 00-4], el mismo que garantiza comunicación determinística en el bus CAN.

TTCAN está basado en la sincronización de todos los nodos en la red por un mensaje específico de sincronización y una funcionalidad adicional de capa sesión, especificada en [ISO00-4]. Esta funcionalidad es revisada en el Capítulo 3, correspondiente a Protocolos de Capa Superior basados en el Protocolo CAN. Basado en la recepción del mensaje de sincronización un mensaje puede ser

transmitido en un slot de tiempo específico, sin tener que competir por el bus con otros mensajes, proveyendo así tiempos fácilmente predecibles de latencia.

Para sincronizar las actividades de los nodos es necesario un punto de referencia común. De acuerdo a [ISO99-1] este puede ser el bit SOF o el punto de muestreo del último bit del campo EOF de cualquier mensaje.

2.2 CAPA FÍSICA

Para su estudio, la funcionalidad de la capa física puede ser dividida en:

- Señalización Física (Representación, Temporización y Sincronización de Bit)
- Unidad de Acceso al Medio (Transceiver)
- Medio Físico

El estándar ISO 11898-1 [ISO99-1], define la señalización física válida para cualquier aplicación CAN, en ISO 11898-2 [ISO99-2] se especifica la unidad de acceso de alta velocidad (1 Mbit/s) al medio para vehículos rodantes y en ISO 11898-3 [ISO99-3] para baja velocidad.

Estándares adicionales especifican la unidad de acceso al medio o partes de ésta para automatización industrial o aplicaciones específicas.

2.2.1 SEÑALIZACIÓN FÍSICA

2.2.1.1 Representación de Bit

Varios de los métodos empleados para codificación de bit tales como “No Retorno a Cero (NRZ)”, “Manchester” o por “Ancho de Pulso” difieren con respecto al número de slots de tiempo requeridos para representar 1 bit (Figura 2.18).

Mientras que en codificación NRZ el nivel de señal se mantiene constante durante todo el tiempo de bit, requiriendo así solo 1 slot de tiempo para representar 1 bit, la codificación Manchester por su parte y debido al cambio de nivel de señal dentro de 1 tiempo de bit, requiere de 2 slots de tiempo para representar 1 bit. La ventaja de la codificación Manchester es que siempre tendrá disponible una transición de la señal por cada bit para su sincronización además de una velocidad más elevada que NRZ.

Debido a que en NRZ el nivel de señal puede mantenerse constante durante un largo periodo de tiempo, dependiendo de los datos transmitidos, deben tomarse medidas adicionales para asegurar que el intervalo de tiempo máximo permisible entre dos transiciones de señal no sea excedido. Esto es posible, por ejemplo aplicando el método de “relleno de bits” (ver sección 2.1.2.7). Con éste método un bit de nivel complementario es insertado en el flujo de bits después de un número específico de bits de igual nivel.

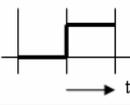
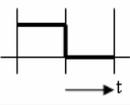
Valor Bit	“0”	“1”
Código		
NRZ		
Manchester		

Figura 2.18: Representación de bits de acuerdo a codificación NRZ y Manchester

Una señal codificada NRZ con relleno de bit y distancia de relleno de 5 aplicada en el protocolo CAN garantiza la capacidad de transporte más alta posible con suficiente capacidad de sincronización. Este tipo de codificación también provee características muy favorables de radiación de la señal¹.

¹ La potencia radiada por una línea del bus es aproximadamente proporcional a la frecuencia de la señal transmitida (número de señales de transición). Información detallada al respecto puede ser encontrada en [DaiUn92].

La codificación NRZ con relleno de bits aplicada en el protocolo CAN comprende a los segmentos de trama SOF, arbitraje, control, campo de datos y secuencia CRC de una trama de datos y de solicitud remota (Ver Figura 2.4). Tan pronto como el transmisor detecta una secuencia de 5 bits de igual valor inserta un bit de valor complementario dentro del arreglo de bits transmitido en realidad. Los segmentos restantes de una trama de datos o remota (delimitador CRC, campo ACK así como EOF) tienen forma fija (nivel recesivo) y son transmitidos sin relleno de bit; al igual que tramas de error y de sobrecarga.

2.2.1.2 Temporización y Sincronización de Bit

Otra de las diferencias del protocolo CAN con otros protocolos de bus de campo es que usa “transmisión sincrónica” de datos en vez de “transmisión asincrónica” (orientada al carácter). Con lo que la capacidad de transmisión es usada más eficientemente¹, aunque de otra manera se requiere de un método mucho más sofisticado de sincronización de bit. Mientras la sincronización de bit en un protocolo orientado al carácter es realizada fácilmente en la recepción del bit inicial disponible con cada carácter, en un protocolo de transmisión sincrónica existe solo un bit inicial disponible al inicio de la trama. Esto no basta para mantener al “muestreo de bit” en el receptor suficientemente sincronizado con el transmisor. Para permitir al receptor realizar correctamente la tarea de muestreo, se requiere de re-sincronización continua en el receptor. Re-sincronización significa detección del periodo de reloj de la señal recibida y es posible con cada transición válida de señal dentro del flujo de bits. El tiempo de diferencia máximo posible entre los osciladores de transmisión y recepción durante el periodo máximo de tiempo entre dos transiciones de señal debe por lo tanto ser compensado por suficiente espacio de tiempo (“segmentos de fase”) antes y después del punto nominal de muestreo dentro de un intervalo de bit.

Debido al tipo de arbitraje utilizado en el protocolo CAN y al bit dominante ACK (como confirmación de correcta recepción) la propagación de la señal desde un transmisor hacia un receptor y su regreso al primero de nuevo debe ser

¹ Ya que el flujo de información es transmitido uniformemente, no así en transmisión asincrónica donde por cada byte se envían bits adicionales para sincronización.

completado dentro de un tiempo de bit. Por lo tanto, en adición al tiempo reservado para re-sincronización, se requiere de un segmento adicional de tiempo (“segmento de retardo de propagación”) para compensar la propagación de la señal sobre la línea de bus así como los retardos internos de la señal en los nodos transmisor y receptor.

La Figura 2.19 muestra la partición resultante del tiempo de bit nominal en cuatro segmentos no-sobrepuestos de tiempo:

- Segmento de Sincronización
- Segmento de Retardo de Propagación
- Segmento 1 de Fase
- Segmento 2 de Fase

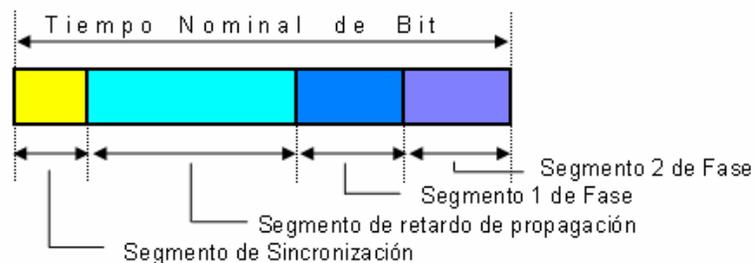


Figura 2.19: Partición del tiempo de bit en segmentos de tiempo de acuerdo a [ISO99-1]

La longitud de los segmentos de tiempo en un intervalo de bit son especificados como múltiplos de una unidad básica de tiempo tq (time quantum) derivado del periodo del oscilador. Una unidad básica de tiempo tq representa el tiempo de resolución del mecanismo de sincronización y es tomado en cuenta por el segmento de sincronización.

El segmento de sincronización es aquella parte de un tiempo de bit donde es esperado que ocurran las transiciones de señal, el cual sin embargo considera que una transición de señal puede ocurrir justo después de que una unidad

básica de tiempo ha pasado (es decir, fuera del segmento de sincronización) y por tanto perderse; la distancia entre una transición que ocurre fuera de este segmento y el segmento mismo es llamada “error de fase e ” de aquella transición.

El segmento de retardo de propagación provee el tiempo necesario para el establecimiento del retardo máximo de propagación de la señal dentro de la red. Este segmento debe ser dos veces tan largo como la suma del retardo de propagación máximo de la señal entre dos nodos a lo largo de la línea del bus y los tiempos de retardo interno de los nodos transmisor y receptor.

Dos tipos de sincronización son distinguidas: “sincronización dura” al inicio de la trama y “re-sincronización” dentro de una trama. Después de una sincronización dura, el tiempo de bit es re-iniciado sin tomar en cuenta el segmento de sincronización. De esta manera la sincronización dura obliga a que la transición causante de la sincronización dura se quede dentro del segmento de sincronización del tiempo de bit re-iniciado. La Re-sincronización por su parte causa un acortamiento o alargamiento del tiempo de bit de tal manera que la posición del punto de muestreo es cambiada con respecto a la transición detectada.

Por medio de los segmentos de fase antes y después del punto nominal de muestreo se reserva tiempo libre para un posible cambio de posición del punto de muestreo actual durante la re-sincronización. La sincronización solamente ocurre en transiciones de nivel de bit recesivo a dominante. Las transiciones son detectadas muestreando el nivel actual del bus en cada t_q y comparándolas con el nivel del bus del muestreo anterior. Una transición es de sincronización si es detectada dentro del segmento de sincronización, de otra manera la distancia entre la transición y el extremo más cercano del segmento de sincronización es el error de fase de transición (e). Si la transición ocurre antes del segmento de sincronización, el error de fase es negativo, de lo contrario positivo.

Para un error de fase positivo el segmento 1 de fase es alargado. La medida máxima permitida de acortamiento o alargamiento de los segmentos de fase para

re-sincronización (Ancho de Salto de Sincronización, SJW “Resynchronization Jump Width”) es limitada y puede ser programada¹ entre 1 y el mínimo entre 4 y el valor del Segmento 1 de Fase.

Si la magnitud del error de fase es menor que SJW, el segmento 1 es alargado en la magnitud del error de fase, caso contrario es alargado en SJW. Para un error de fase negativo, el segmento 2 de fase es acortado.

Cuando la magnitud del error de fase de la transición es menor o igual al valor programado de SJW, los resultados de sincronización dura y re-sincronización son los mismos. Si la magnitud del error de fase es más larga que SJW, la re-sincronización no puede compensar completamente el error de fase, así un error de fase-SJW permanece.

Los diferentes segmentos de un tiempo de bit pueden ser programados dentro de los siguientes límites:

- Segmento de Sincronización: 1 tq
- Segmento de Propagación: 1..... 8 o más tq
- Segmento 1 de Fase: 1..... 8 o más tq
- Segmento 2 de Fase: Máximo valor entre segmento 1 de fase y el llamado “tiempo de procesamiento de información”
- Baud Rate Prescaler²: 1....32
- SJW: 1.....4 pero no más largo que el mínimo entre 4 y el Segmento 1 de Fase

En caso de una sincronización el Segmento 1 de Fase puede ser más largo y el Segmento 2 más corto que su valor nominal programado.

El “tiempo de procesamiento de información” empieza después del punto de muestreo y es el tiempo requerido para determinación del nivel de bit subsiguiente

¹ La sección 2.2.1.3 ilustra ejemplos prácticos de dimensionamiento de estos segmentos.

² Ver sección de 2.2.1.3

a ser enviado (por ejemplo bit de datos, bit CRC, bit de relleno, bandera de error o desocupado). El tiempo requerido para esta implementación es específico pero no puede ser más largo que $2 t_q$. En caso de una sincronización el segmento 2 de fase puede ser acortado a un valor menor que el tiempo de procesamiento de información, que no afecte la temporización del bus.

El número total de unidades t_q por bit tiene que ser programable al menos de 8 a 25.

La Figura 2.20 muestra el principio de re-sincronización. Si el oscilador del transmisor es más lento que el del receptor (caso a), una señal de transición utilizable para sincronización llega retrasada al receptor. El receptor puede compensar esto cambiando de posición al punto de muestreo. El “retraso” máximo acumulado (error de fase de transición, e) que puede ser compensado es dado en este caso por el tiempo reservado por el segmento 1 de fase. Durante re-sincronización, el punto de muestreo es cambiado de acuerdo a si la magnitud del “ e ” ocurrido es menor o igual a SJW, restaurando así la posición del punto de muestreo relativo a la actual posición de bit.

Se aplica lo mismo cuando el oscilador del transmisor es más rápido que el de un receptor (caso b). En este caso, una transición de señal llega más temprano que lo esperado por el receptor, por tanto el próximo intervalo de tiempo de bit tiene que iniciar más temprano. Esto es posible acortando el segmento 2 de fase. En este caso el Segmento de Sincronización debe ser omitido para tener el mismo efecto como si la transición hubiera ocurrido dentro del segmento de sincronización. Similar al primer ejemplo, la magnitud de este error “tempranero” es menor que SJW, así que es completamente compensado.

Los segmentos de fase son cambiados solo temporalmente; en el próximo bit, los segmentos tienen sus valores nominales programados de nuevo, si no se ha detectado un error de fase de nuevo.

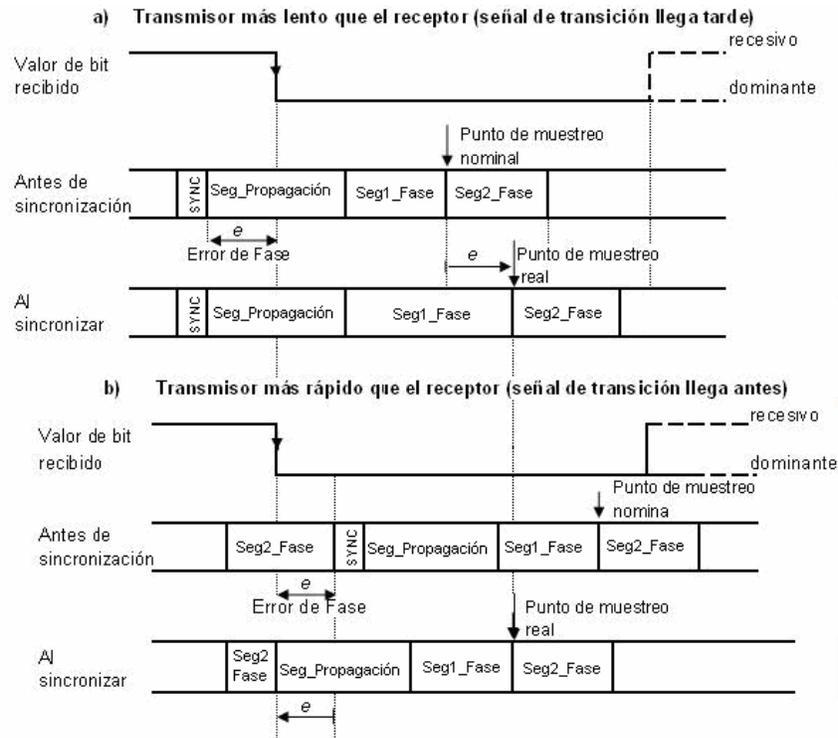


Figura 2.20: Principio de Re-sincronización

Las siguientes reglas se aplican para sincronización de bit [ISO99-1]:

- Solamente una sincronización entre dos puntos de muestreo dentro de un tiempo de bit
- Una señal de transición de nivel recesivo a dominante será usada para sincronización solamente si el nivel de señal detectado en el punto de muestreo anterior difiere del nivel de señal inmediatamente después de la señal de transición. Evitando así que breves pulsos interferentes sean interpretados, erróneamente, como señales de transición.
- Una “sincronización dura” (al inicio de un nuevo tiempo de bit) es realizada cada vez que existe una transición de recesivo a dominante durante el bus desocupado¹

¹ Una sincronización dura después de una señal de transición de recesivo a dominante es también realizada durante el espacio inter-trama con la excepción del primer bit del campo de interrupción.

2.2.1.3 Dimensionamiento de los Segmentos de Temporización de Bit

El dimensionamiento de los segmentos de temporización de bit es determinado por los requerimientos del sistema de comunicación de datos. Si una longitud de bus máxima es requerida a una velocidad de transmisión específica o un tiempo de latencia mínimo (máxima velocidad de transmisión) es requerido a una longitud de bus dada, las reservas de tiempo (segmentos 1 y 2 de fase) disponibles para re-sincronización deben ser establecidos al mínimo. Cuando los segmentos de fase son dimensionados a su valor mínimo, significa sin embargo que solo un error de fase de $|e| = 1$ puede ser corregido durante una re-sincronización. Así entonces, los requerimientos con respecto a sincronización de bit son altos y deben ser realizados solamente usando precisos osciladores de cuarzo¹.

La Figura 2.21 muestra el dimensionamiento de los segmentos de temporización para una máxima velocidad de transmisión y longitud de bus. Este requerimiento principalmente existe para aplicaciones en sistemas de automatización industrial.

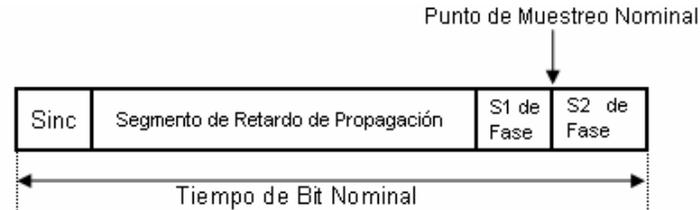


Figura 2.21: Dimensionamiento de los segmentos de temporización de bit para un máximo provecho de la velocidad de transmisión y longitud del bus

Si los requerimientos con respecto a velocidad de transmisión y longitud de bus son bajos, los segmentos de fase para re-sincronización pueden ser incrementados. Así, en relación con el SJW máximo posible, errores de fase de $|e| = 4$ pueden ser corregidos durante una re-sincronización. Por lo tanto, pueden ser empleados osciladores cerámicos eficientes en costo. La Figura 2.22 ilustra el dimensionamiento de los segmentos de temporización de bit tomando en cuenta

¹ El cuarzo habitualmente proporciona una tolerancia menor a 0.1%.

una tolerancia máxima de oscilador como es principalmente usado en la electrónica del confort¹ en automóviles.

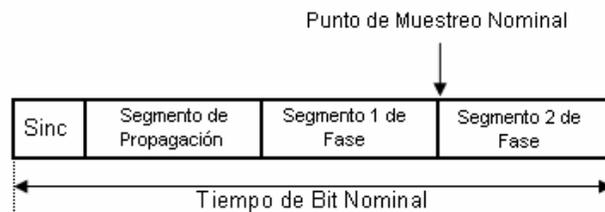


Figura 2.22: Dimensionamiento de los segmentos de temporización de bit para una tolerancia de oscilador máxima

Usualmente, la determinación de los parámetros de temporización de bit inicia con la velocidad de transmisión o el tiempo de bit deseados. El tiempo de bit debe ser un múltiplo entero del periodo de la señal de reloj. El tiempo de bit $t_{Bit} = n \cdot tq$ puede consistir de $n = 4 \dots 25$ tq . Una propuesta para determinar los parámetros de temporización de bit es empezar con el tamaño del segmento de propagación requerido. Por ello la longitud máxima del bus y tiempos máximos de retardo interno tienen que ser considerados.

El tiempo de retardo de ida y vuelta resultante es convertido al número tq correspondiente, redondeándolo a su múltiplo más cercano. Ya que el segmento de sincronización tiene un tamaño de 1 tq , se tiene que $m = t_{Bit} - t_{prop_seg} - tq$ es la cantidad de unidades de tiempo restantes para los dos segmentos de fase. Si este valor de m es un número par, ambos segmentos de fase tendrán la misma longitud, de lo contrario se usará la relación: $t_{phase_seg2} = t_{phase_seg1} + tq$.

La longitud mínima nominal del Segmento 2 de Fase tiene que ser considerada también, ya que este segmento no puede ser más corto que el tiempo de procesamiento de información del controlador CAN que es, dependiendo de la implementación, entre 0 y 2 tq . La longitud de SJW es establecida a su máximo valor, que es el mínimo entre 4 y el Segmento 1 de Fase.

¹ Referente a sistemas de calefacción, climatización, navegación, sonido entre otras.

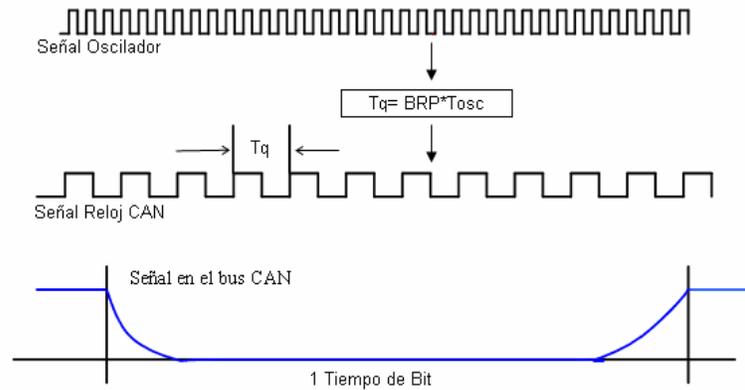


Figura 2.23: Representación gráfica de la señales de oscilador, reloj CAN y de 1 tiempo de bit, para propósitos de sincronización

Para que el controlador, pueda muestrear la señal asíncrona recibida con seguridad y precisión ha de utilizar una frecuencia de muestreo superior a la de la señal transmitida en el bus. La frecuencia básica del controlador es así varias veces superior a la de comunicación en el bus, el número de unidades tq por bit puede hallarse de $1tbit=n*tq$. De tal manera que los segmentos están compuestos de un cierto número programable de ciclos de reloj CAN (f_{CLK_CAN}). Estos ciclos están definidos a partir del oscilador de cuarzo y un valor BRP (Baud Rate Prescaler) que actúa como un divisor de frecuencia indicando cuántos ciclos de señal del oscilador tendrá un ciclo de señal de reloj CAN (Figura 2.23).

El valor de tq (su inverso es la frecuencia del controlador (f_{CLK_CAN})) es obtenido al multiplicar el periodo de la señal del oscilador por un valor de BRP que es normalmente configurable.

$$f_{CLK_CAN} = \frac{f_{Oscilador}}{BRP}$$

$$(f_{CLK_CAN})^{-1} = \left[\frac{f_{Oscilador}}{BRP} \right]^{-1}$$

$$t_q = \frac{BRP}{f_{Oscilador}}$$

$$t_q = BRP * T_{Oscilador}$$

El mínimo valor de BRP es de 1, y su longitud está en función de la frecuencia del oscilador y de la velocidad de transmisión de tal manera que se satisfagan los requerimientos de de los segmentos en cuanto a cantidad de unidades t_q .

La tolerancia necesaria de oscilador es determinada de acuerdo a la sección 2.2.1.4 por las fórmulas (3-1a) y (3-1b) respectivamente.

Como ejemplos del dimensionamiento de parámetros de temporización de bit, a continuación se citan dos ejemplos en los que se los determina para un sistema de alta velocidad de transmisión con su respectiva longitud de bus máxima (Figura 2.21) y para un sistema de baja velocidad (Figura 2.22).

Ejemplo 1: Dimensionamiento de parámetros de temporización de bit para alta velocidad de transmisión con su respectiva longitud de bus y baja tolerancia de oscilador

Frecuencia de Reloj Oscilador= 10 MHz, Valor de BRP=1, Velocidad de Transmisión= 1 Mbit/s

$$V_t = 1 \text{ Mbit} / s$$

$$t_{Bit} = 1 / V_t$$

$$t_{Bit} = 1000 \text{ ns}$$

$$t_q = BRP * T_{Oscilador}$$

$$t_q = BRP * \frac{1}{10 \text{ MHz}} = 100 \text{ ns}$$

$$n = t_{Bit} / t_q$$

⑤ ☞ ☐☐ ①⑤①④④④④④④④ ④④ ①①④④④④ ④④④④④④④ ④④ ①⑤①④④④④④④④
④④ ④④④④④④④ ④④④ ④④④④④④④ ①⑤ ①①④④④④④ ④④ ④④④④

☑④④④④④④ ④④④ ④④④④ ④④④④④④④ ☐☐④④④④

☑④④④④④④ ④④④ ④④④④④④④④④④ ④④④④④④④④④④ ☐☐④④④④

☑④④④④④④ ④④④ ④④④ ④④④④④④④ ④④ ④④④④ ☐☐☐④④④④



$$t_{\text{Synch_Seg}} = 1 \cdot t_q$$

$$t_{\text{Pr_op_Seg}} = 2 \cdot (200 + 80 + 220) = 1000 \text{ ns} = 1 \cdot t_q$$

$$m = n - t_{\text{Pr_op_seg}} - t_{\text{sin c}}$$

$$m = 10 - 1 - 1 = 8 t_q \dots\dots\dots (\text{par})$$

$$t_{\text{Phase_Seg1}} = 4 \cdot t_q = 4 \mu\text{s}$$

$$t_{\text{SJW}} = 4 \cdot t_q = 4 \mu\text{s}$$

$$t_{\text{TSEG1}} = t_{\text{Pr_op_Seg}} + t_{\text{Phase_Seg1}} = 5 \mu\text{s} = 5 \cdot t_q$$

$$t_{\text{Phase_Seg2}} = t_{\text{Phase_Seg1}} = 4 \cdot t_q = 4 \mu\text{s} = t_{\text{TSEG2}}$$

2.2.1.4 Tolerancia de Oscilador

En cada re-sincronización el error de fase (ϵ) puede ser reducido al máximo por el SJW. Un error de fase residual permanece cuando el error de fase no fue completamente compensado durante la re-sincronización. Los errores de fase residual pueden acumularse, sin embargo nunca deben exceder el SJW programado dentro de 10 tiempos de bit¹, ya que ésta representa la mayor distancia posible que puede existir entre dos transiciones de recesivo a dominante de acuerdo al principio de relleno de bit aplicado en este protocolo. Con ésta limitación, la máxima tolerancia admisible de la frecuencia de oscilador (“drift”, abreviada como df) entre dos nodos puede ser hallada dependiendo del SJW (3-1a):

$$df \leq \frac{t_{\text{SJW}}}{20 * t_{\text{BIT}}} \quad (3-1a)$$

con t_{BIT} Tiempo Nominal del Bit
 t_{SJW} Salto Máximo con tiempo de punto nominal de muestreo por re-sincronización (Re-Sincronización Jump Width)

¹ Este es el caso cuando, de acuerdo al relleno de bits, después de una sucesión de 5 bits dominantes, 5 recesivos le siguen. Este caso se ilustra en la Figura 2.15

El *drift* máximo admisible del oscilador sin tomar en cuenta SJW es también limitado por el escenario del peor caso para re-sincronización cuando un bit de relleno es destruido por un error global (Figura 2.24). En este caso resulta un periodo de 12 bits dominantes.

Para aumentar correctamente los contadores de error de los nodos, el nivel recesivo que sigue a la bandera de error activa todavía tiene que ser recibida correctamente con el punto de muestreo del bit 13avo. Por tanto una relación adicional para la tolerancia admisible máxima de oscilador df puede ser hallada de esta (3-1b):

$$df \leq \frac{\min\{t_{Phase_Seg1}, t_{Phase_Seg2}\}}{2 * (13 * t_{Bit} - t_{Phase_Seg2})} \quad (3-1b)$$

con t_{Phase_Seg1} Tiempo del Segmento 1 de Fase
 t_{Phase_Seg2} Tiempo del Segmento 2 de Fase
 t_{Bit} Tiempo nominal de Bit

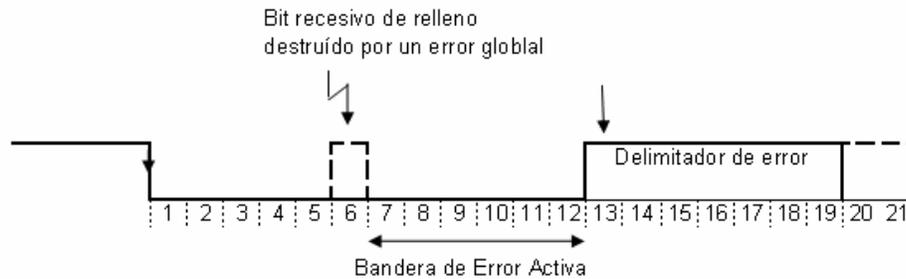


Figura 2.24: Periodo de tiempo más largo posible sin posibilidad de re-sincronización (destrucción de un bit de relleno recesivo por un error global)

Ambas condiciones (3-1a) y (3-1b) deben ser encontradas para el dimensionamiento de los osciladores.

De acuerdo a los ejemplos anteriores las tolerancias requeridas por los osciladores resultarían en lo siguiente:

Ejemplo 1:

$$df \leq \frac{\min\{t_{Phase_Seg1}, t_{Phase_Seg2}\}}{2 * (13 * t_{Bit} - t_{Phase_Seg2})} = \frac{100nS}{2 * (13 * 1000nS - 200nS)} = 0.39\%$$

Ejemplo 2:

$$df \leq \frac{\min\{t_{Phase_Seg1}, t_{Phase_Seg2}\}}{2 * (13 * t_{Bit} - t_{Phase_Seg2})} = \frac{4\mu S}{2 * (13 * 10\mu S - 4\mu S)} = 1.58\%$$

2.2.1.5 Relación entre la Velocidad de Datos y Longitud del Bus

El dimensionamiento del segmento de retardo de propagación define la longitud máxima posible del bus a una velocidad de datos específica o la máxima velocidad de datos posible a una longitud de bus dada. La Figura 2.25 ilustra la peor situación posible entre dos nodos del bus, N y M, lo más separados el uno del otro en una red durante el proceso de arbitraje del bus o el envío del slot ACK.

En este caso, se observa que el nodo N transmite un bit de nivel recesivo en el punto de tiempo t1. Esta señal alcanza el nodo M al punto de tiempo t2 retardado por el tiempo de propagación de la señal Td. El máximo tiempo de retardo está definido por el tiempo de propagación de la señal entre los nodos más alejados. Basado en el tiempo de propagación existente, el nodo M puede iniciar la transmisión de un nuevo nivel de bit al nodo N (en respuesta al bit transmitido por N) y transmitir por ejemplo un bit dominante incluso hasta el tiempo t2. Sin embargo, este nivel no puede ser detectado por el nodo N sino hasta el punto t3. Lo que significa que, hasta ese tiempo el nodo N no sabe si el nivel de señal que transmitió fue retenido o si fue reemplazado por un nivel dominante por parte del nodo M como en el presente caso.

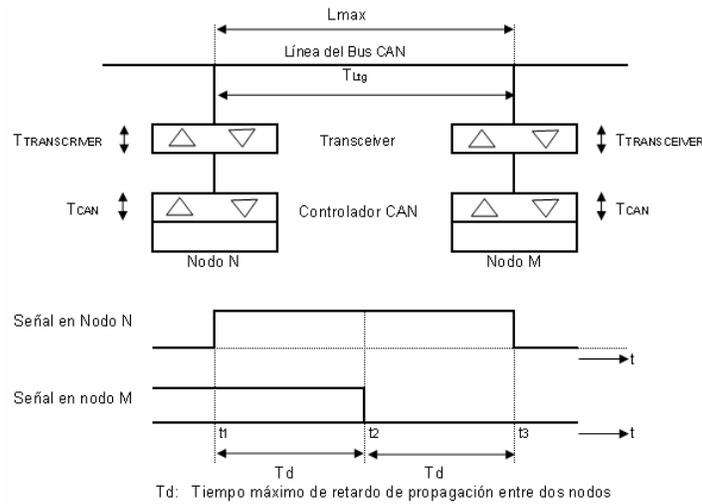


Figura 2.25: Consideración del peor caso para el cálculo de la longitud requerida del segmento de retardo de propagación

Esto explica que, se aplique lo siguiente para dimensionar la longitud del segmento de retardo de propagación:

$$t_{Prop_seg} \geq 2 \cdot t_d \quad (3-2)$$

$$t_d = t_{el} + t_{Lig}$$

$$t_{el} = t_{CAN} + t_{Transceiver}$$

$$t_{Lig} = L_{Lig} \cdot t_p$$

con	t_{Prop_seg}	tiempo de propagación del segmento
	t_d	tiempo de propagación máximo de la señal entre nodos
	t_{el}	tiempo de retardo de la señal eléctrica en el nodo
	t_{Lig}	tiempo de retardo de la señal sobre el bus
	t_{CAN}	tiempo de retardo del CAN protocolo chip
	$t_{TRANSCIVER}$	tiempo de retardo de la unidad transceiver
	t_p	tiempo específico de propagación del medio bus

L_{Ltg} distancia del segmento

La longitud máxima de un bus, L_{max} , de 2 alambres es obtenida con el tiempo específico de propagación t_p sobre un par de cables, el cual es aproximadamente 5 ns/m de acuerdo a la relación (3-3a):

$$L_{max} = \frac{0,5 \cdot t_{Prop_Seg} - t_{el}}{5 \frac{ns}{m}} \quad (3-3a)$$

Con t_{Prop_seg} tiempo de retardo de propagación del segmento
 t_{el} tiempo de retardo de la señal eléctrica en el nodo

De la relación (3-3a) se puede apreciar que la longitud máxima posible del bus es obtenida a través del segmento de propagación de retardo más largo posible (t_{Prop_seg}), y también que los retardos de la señal eléctrica debido a unidades de transceiver, controladores CAN y acopladores optoelectrónicos¹ reduce la longitud máxima posible del bus en 2 metros por cada 10 ns de retardo adicional de señal.

Se puede obtener un estimado de la relación entre la longitud máxima de bus y la velocidad de transmisión asumiendo que existe una cierta porción χ que relaciona el tiempo de bit t_{Bit} con t_{Prop_seg} . Entonces es posible escribir:

$$t_{Prop_seg} = \chi \cdot t_{Bit} = \frac{\chi}{V_{tx}} \quad (3-3b)$$

Con V_{tx} velocidad de transmisión

O insertarlo en la relación (3-3a):

¹ Dispositivo que internamente convierte la señal eléctrica en óptica por medio de un emisor de luz, de esta forma es transmitida y luego re-convertida a la señal eléctrica por un fotodetector.

$$L_{\max} = \frac{\frac{\chi}{2 \cdot V_{tx}} - t_{el}}{5 \frac{ns}{m}} \quad (3-3c)$$

En la tabla 2.1, se aprecia la relación entre velocidad de transmisión y longitud máxima del bus de acuerdo a (3-3c) para varios valores de velocidad bajo las suposiciones: $\chi=0.85$, $t_{el}=300ns$. La Figura 2.26 ilustra una representación gráfica de esta relación.

Velocidad de Transmisión [kBit/s]	Longitud máxima del Bus [m]
500	110
250	280
125	620
100	790
50	1640

Tabla 2.1: Velocidad de transmisión y longitud máxima del bus (de acuerdo a la fórmula 3-3c con $\chi=0.85$, $t_{el}=300nS$)

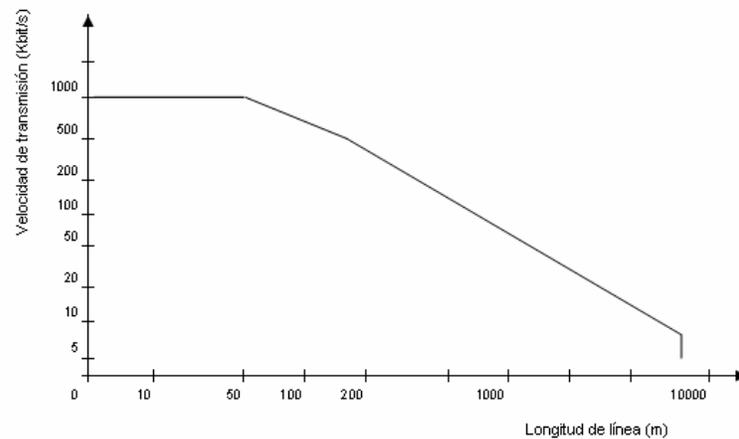


Figura 2.26: Máxima longitud del bus en función de la velocidad de transmisión

Para longitudes de bus mayores a 100 metros, la siguiente relación puede ser establecida como una regla para el máximo provecho de la velocidad de transmisión y longitud del bus:

$$V_{tx_{max}} \left[\frac{MBit}{s} \right] \cdot L_{max} [m] \leq 60 \quad (3-4)$$

Estos resultados demuestran que CAN también puede ser usado para realizar redes extendidas ya que sus magnitudes posibles de velocidad de datos son de orden equivalente a aquellas de otros sistemas de bus de campo.

2.2.2 MEDIOS DE TRANSMISION

Un pre-requisito fundamental para el método de arbitraje de bus en CAN es la habilidad para representar un nivel de señal “dominante” y “recesivo”, es decir que un medio de transmisión no esté en condición recesiva a no ser que todos los nodos hayan aplicado un bit de nivel recesivo. Tan pronto como un nodo ha iniciado la transmisión de un bit de nivel dominante, el bus debe estar en nivel dominante. Este principio puede ser realizado en medios eléctricos y ópticos. Con medios de transmisión ópticos el nivel recesivo es representado por el estado “oscuro” y el nivel dominante por el estado “brillo”.

2.2.2.1 Medio de Transmisión Eléctrico

El medio eléctrico típicamente usado para implementar redes CAN es un par diferencialmente¹ manejado de cables con retorno común (Figura 2.27a). Específicamente para la electrónica del cuerpo del vehículo también un solo cable de bus de línea es implementado (Ver Figura 2.7b). También se han hecho esfuerzos por desarrollar soluciones para transmisión de señales CAN junto con el suministro de energía en las mismas líneas.

¹ Líneas en donde las transiciones de voltajes y de corrientes son iguales y opuestas.

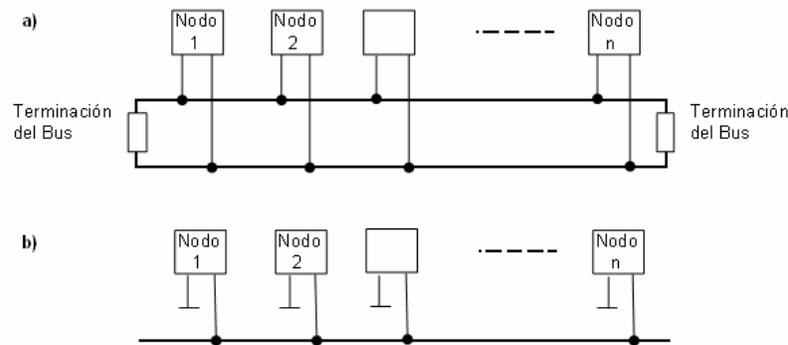


Figura 2.27: Bus CAN de medio eléctrico (a. Bus de dos alambres
b. Bus de un solo alambre)

Bus de dos Alambres

El bus de dos alambres permite transmisión diferencial de la señal siendo así resistente a modos comunes de errores. Una transmisión de señal muy confiable es garantizada a pesar de bajos niveles de señal. Además, la interferencia electromagnéticamente inducida puede ser compensada trenzando el par de cables, incrementándose así la inmunidad a interferencia. Cuando se provee una funcionalidad apropiada de manejo de errores en el bus la comunicación puede continuar bajo reducida inmunidad al ruido aún si una línea se rompe o si está en corto circuito.

Los más importantes estándares adicionales de capa física CAN están basados en un bus de dos alambres.

Bus de un solo Cable

El uso de buses de un solo alambre es particularmente interesante para la electrónica del cuerpo del vehículo. Esta solución asume una tierra común para los nodos. Por lo que un solo cable puede ser usado solamente para aplicaciones particulares. Ya que una sola línea de cable de bus es expuesta a la interferencia cuando no está blindada, es necesario un nivel de señal más alto para mejorar la relación señal a ruido. Esto, a su vez afecta el grado de emisión electromagnética y por tanto de la máxima velocidad de datos posible. El un solo cable del bus está especificado por el estándar SAE J 2411 [SAE00-1].

Transmisión de señal y suministro de energía en las mismas líneas

Varios sistemas de bus de campo tales como AS-I o PROFIBUS-PA transmiten energía además de datos sobre las mismas líneas de bus. La provisión de energía sobre el bus sería deseable también para sistemas CAN, pero la transmisión simultánea de energía y datos es problemática debido al proceso de arbitraje usado en el protocolo CAN. Posibles soluciones fueron investigadas por [RaBei95]. Esta investigación mostró que la transmisión de energía está en relación con costosos circuitos, pérdida de inmunidad a interferencia y longitud del bus. Así estas soluciones no pueden competir con los costos de líneas adicionales de suministro de energía adicionales.

Parámetros más importantes de un medio eléctrico

Los parámetros de un medio eléctrico vienen a ser importantes cuando la longitud del bus es extendida. El tiempo de propagación de señal y la resistencia específica de la línea a través del alambre deben ser tomados en cuenta para el dimensionamiento de redes extendidas. La velocidad de propagación V_p de líneas libres de pérdidas está dada por:

$$V_p = \frac{c}{\sqrt{\mu_r \cdot \epsilon_r}} \quad [m/s]$$

Con c velocidad de la luz ($3 \cdot 10^8 \frac{m}{s}$)

μ_r permeabilidad relativa

ϵ_r dielectricidad relativa

μ_r y ϵ_r son determinadas por la inductividad y capacitancia específicas de la línea respectivamente.

El tiempo de propagación de la señal (t_p) de dos líneas de cable está en el rango de 5 a 7,5 ns/m. La tabla 2.2 muestra los máximos valores posibles de la velocidad de transmisión para líneas, de 1000 m. de longitud, con varios tiempos de propagación.

Tiempo específico de propagación de la señal [nS/m]	Velocidad de transmisión máxima [kBit/s]
5.0	80
5.5	73
6.0	67
6.5	62
7	58

Tabla 2.2: Máxima velocidad de transmisión posible como función del tiempo específico de propagación de la señal en una red CAN de un par de alambres. (Asumiendo: Longitud de línea= 1000m, $\chi = 0.85$, $t_{el} = 300\text{nS}$)

Con líneas de bus más largas, ya no se puede pasar por alto el voltaje que cae sobre las líneas. La sección transversal requerida del alambre está determinada por la caída de voltaje del nivel de la señal entre los dos nodos más lejanos disponibles sobre la línea así como por la resistencia total de entrada de todos los receptores conectados. La máxima caída de voltaje sobre la línea debe ser tal que el nivel de señal pueda ser interpretado confiablemente en cualquier receptor cuando el transmisor más alejado envíe un nivel mínimo de señal. Por simplificación, las resistencias distribuidas en la línea son resumidas como R_x y la resistencias de entrada de todos los $n-1$ nodos como R_{intot} en la Figura 2.28.

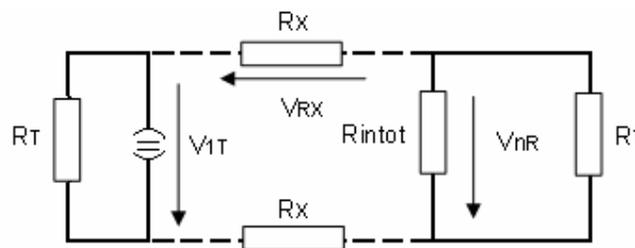


Figura 2.28: Condiciones de voltaje en un bus de dos alambres.

Modelo para provisión de un nivel de voltaje por medio del nodo 1 en un extremo de la línea de un bus y recepción por un nodo n en el otro extremo del bus en una red con n nodos

Con un bus de dos cables, aplicando división de tensión el nivel de voltaje recibido V_{nR} del nodo n más alejado está dado por la relación (3-5a):

$$V_{nR} = V_{1T} \cdot \frac{R_{int\ o\ t} \parallel R_T}{2 \cdot R_x + R_{int\ o\ t} \parallel R_T} \quad [V] \quad (3-5a)$$

Con	V_{nR}	Voltaje visto por el nodo n
	V_{1T}	Voltaje provisto por el nodo 1
	$R_{int\ o\ t}$	Resistencia total de entrada de los receptotes
	$R_{int\ o\ t} = \frac{R_{in}}{n-1}$	(Paralelo de las resistencias R_{in} de los $n-1$ nodos)
	R_{in}	Resistencia de entrada de cada uno de los receptores
	n	Número de nodos
	R_T	Resistencia de terminación
	R_x	Resistencia distribuida en la línea

De (3-5a) se puede despejar la siguiente relación para hallar la resistencia de línea R_x :

$$R_x < \frac{(R_{int\ o\ t} \parallel R_T) \cdot (V_{1T} - V_{nR})}{2 \cdot V_{nR}} \quad [\Omega] \quad (3-5b)$$

$$\text{con } R_x = \rho \cdot \frac{L}{A} \quad [\Omega]$$

ρ resistencia específica de la línea de bus por longitud

El área requerida por la sección transversal de un alambre resulta de (3-5b):

$$A(L) \geq \frac{2 \cdot \rho \cdot L \cdot V_{nR\ min}}{(V_{1T\ min} - V_{nR\ min}) \cdot (R_{int\ o\ t} \parallel R_T)} \quad [m^2] \quad (3-5c)$$

CON	ρ	resistencia específica de la línea de bus por longitud
	L	longitud de la línea de bus
	$V_{1T\min}$	Mínimo voltaje de transmisión
	$V_{nR\min}$	Mínimo voltaje de entrada requerido del receptor más alejado

Cuando un bus es usado de acuerdo a la especificación [ISO99-2] una diferencia en el nivel de voltaje entre las dos líneas esta presente solamente durante el estado dominante (ver sección 2.2.5.1). Por tanto únicamente el nivel dominante es de relevancia para el dimensionamiento del área de la sección transversal del alambre. Si para esto $V_{nR\min}$ (voltaje mínimo a ser recibido por el nodo más alejado) toma en consideración un margen de seguridad del 10% de $V_{1T\min}$ (mínimo voltaje transmitido) en adición a los 0.9V¹ que debe detectar, así como el valor más pequeño permitido de la resistencia terminal ($R_{T\min}$), entonces los siguientes valores se aplican para el cálculo de la sección requerida:

$$V_{1T\min} = 1,5V; \quad V_{nR\min} = 0,9V + 0,15V; \quad R_{T\min} = 100\Omega; \quad \rho = 0,0178 \frac{\Omega m}{mm^2}$$

\downarrow
 $10\% V_{1T\min}$

La Figura 2.29 muestra el área de la sección transversal requerida del alambre requerida dependiendo de la longitud de línea y número de nodos para los niveles de voltaje requeridos por [ISO99-2].

¹ $V_{nR\min}$ representa al voltaje diferencial de entrada V_{diff} y de acuerdo a [ISO99-2] su valor mínimo debe ser 0.9V para que un receptor reconozca un nivel dominante (Ver sección 2.2.5.1)

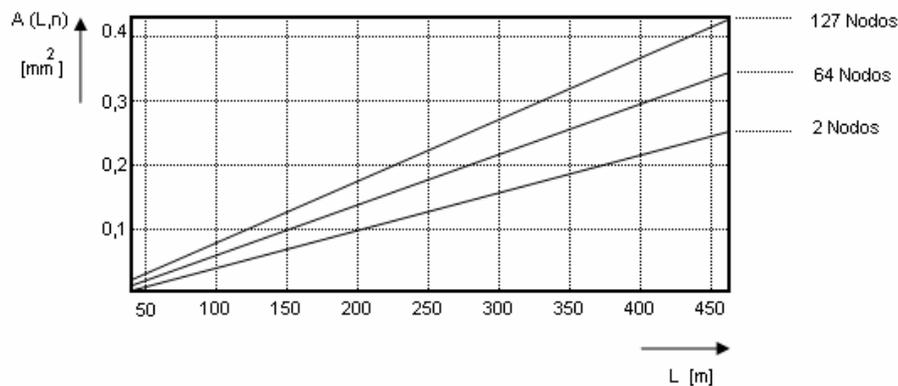


Figura 2.29: Dependencia entre la sección de área requerida del alambre y longitud de la línea (de acuerdo a ISO 11898-2)

El tercer parámetro de una línea eléctrica es la “impedancia de línea”. La impedancia de línea Z de una línea casi libre de pérdidas esta dada por:

$$Z = \sqrt{\frac{L^*}{C^*}}$$

Donde L^* y C^* representan la inductancia y capacitancia específicas de las líneas. Ambos parámetros dependen del diámetro y distancia de las líneas. Para evitar reflexión¹ de la señal en el extremo de una línea es necesario terminarla en ambos extremos con una resistencia $R=Z$. El valor típico de impedancia de una línea de dos cables es de aproximadamente 120 Ohms.

Debido al corto tamaño de la trama y a los tiempos de recuperación de error así como a la alta capacidad de detección de errores, CAN también puede ser aplicado en ambientes de alta interferencia electromagnética. El uso de cable blindado y de par trenzado no es requerido a menos que se espere una interferencia electromagnética extrema. Poner los cables cerca entre sí y libres de frecuencias altas de ruido es ventajoso para emisiones bajas y para evitar la recepción de señales de ruido no deseadas. Si una tierra común no está disponible para los nodos, ésta puede ser provista por ejemplo a través del

¹ Parte de la energía que llega al final de la línea se refleja hacia la fuente, no pudiéndose entregar a sus extremos.

blindaje del alambre blindado. Cuando las líneas son doblemente blindadas también es posible usarlas cerca de líneas de energía.

2.2.2.2 Medios de Transmisión Ópticos

Las ventajas conocidas de los medios ópticos tales como transmisión libre de problemas de inmisión (interferencia electromagnética inducida) y emisión (niveles elevados de potencia electromagnética emitida), están llegando a ser muy importantes para redes CAN debido a la creciente disponibilidad de tecnología de transmisión óptica eficiente en costo, especialmente fibra óptica de plástico. La característica de inmunidad a interferencia eléctrica del medio óptico es atractiva primeramente para aplicaciones en ambientes potencialmente explosivos y perturbados electromagnéticamente. Debido al acople directo en el medio óptico, las líneas de transmisión y recepción deben estar separadas. Además cada línea de recepción debe ser acoplada externamente con cada línea de transmisión para realizar el monitoreo de bit requerido por el protocolo CAN. Esta función puede ser implementada por un acoplador estrella.

El uso de acopladores pasivos¹ estrella es posible con un número pequeño de nodos ya que dividiendo la potencia de transmisión entre las líneas receptoras se limita considerablemente el número de nodos posibles.

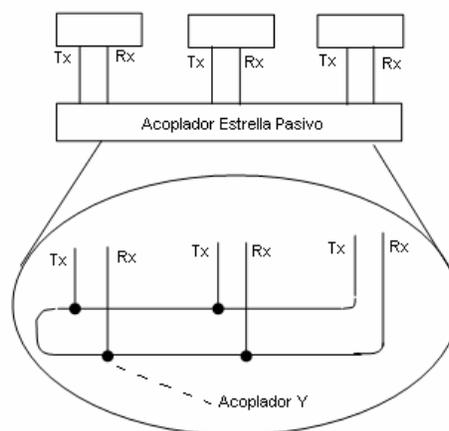


Figura 2.30: Estructura básica de una red CAN basada en un medio óptico con un acoplador estrella pasivo

¹ Dispositivo conmutador que recibe la señal y la re-transmite sin amplificarla.

La figura 2.30 muestra la estructura de una red CAN basada en un medio óptico usando un acoplador pasivo estrella.

Debido al limitado número de nodos posible cuando se usan acopladores estrella pasivos, la extensibilidad de estas redes también es limitada.

La máxima extensión posible de una red CAN con medio óptico es limitada por la potencia de luz disponible, la potencia de atenuación a lo largo de la línea de transmisión y el acoplador pasivo estrella en lugar del tiempo de propagación de la señal como ocurría en los medios eléctricos.

De igual manera que el voltaje disponible y que la caída de éste a lo largo de una línea eléctrica de un bus, la máxima extensión de una red con medio óptico es determinada por la potencia disponible de luz y por las pérdidas causadas por los componentes de la red. En este caso, la potencia de luz que llega al nodo más lejano debe ser lo suficientemente alta para permitir una interpretación certera del nivel de potencia recibido. Tecnología de fibra óptica de plástico de alta calidad con bajos valores de atenuación está disponible en el mercado la cual es una alternativa para remplazar productos de fibra óptica de vidrio que poseen muy bajas pérdidas pero a altos costos económicos. La potencia de atenuación de un acoplador pasivo estrella es principalmente dependiente del número de puntos de acople.

La máxima extensión L_{opt} ¹ de una red óptica con un acoplador pasivo estrella que resulta de la atenuación a lo largo de la red puede ser calculada de acuerdo a (3-6). La máxima extensión depende de: la mínima potencia $P_{TX\ min}$ del transmisor, la máxima atenuación $P_{TL\ max}$ a lo largo de la fibra óptica, la atenuación causada por el acoplador pasivo estrella $P_{STAR\ max}$ así como de la potencia de recepción mínima requerida $P_{RX\ min}$ del receptor.

¹ Máxima distancia entre dos nodos conectados por un acoplador estrella.

$$L_{opt} [m] \leq \frac{P_{TX \min} [dBm] - P_{RX \min} [dBm] - P_{STAR \max} [dB]}{P_{TL \max} [dB / m]} \quad (3-6)$$

El nivel relativo de potencia logarítmico es definido como: $10 \cdot \log(P_{in}/P_{out})$ en decibelios (dB). Con $P_{out} = 1mW$, la relación es especificada en dB_m

Si el acoplador estrella pasivo es sustituido por otro activo¹, la atenuación de éste no debe ser considerada más en la relación (3-6). En este caso la máxima longitud de la red óptica debido a la atenuación es determinada por la potencia óptica disponible para la conexión entre un nodo transmisor y el acoplador estrella activo como receptor, o el acoplador como transmisor y un nodo como receptor.

Muy a menudo, como parte de una red extendida es suficiente hacer uso de una conexión punto a punto de medio óptico. Por ejemplo para puentear secciones potencialmente explosivas de una red. Repetidores y puentes de fibra óptica permiten reemplazar dichas secciones (de medio eléctrico) con medios de transmisión ópticos en las secciones requeridas. Así, es posible una combinación eficiente en costo usando medios de bus eléctrico y óptico. A continuación se provee una visión general de puentes y repetidores así como su posible uso.

2.2.3 TOPOLOGÍA DE RED

Señales eléctricas transmitidas sobre las líneas del bus serán reflejadas en los extremos de una línea eléctrica así como en líneas drop² si no se toman medidas apropiadas. Para la correcta interpretación de los niveles de bus recibidos por un nodo, es necesario que reflexiones sobrepuestas de señal sean suficientemente atenuadas en el momento en que el nivel de bit es muestreado. Estas reflexiones de la señal pueden ser evitadas terminando ambos extremos de las líneas de bus con una resistencia terminal de igual valor a la impedancia de la línea del bus y evitando longitudes de stub³ más largas.

¹ Dispositivo repetidor, que regenera la señal recibida antes de re-transmitirla.

² Segmentos de cable utilizados para acoplar los nodos al medio de transmisión.

³ Se refiere a las líneas drop.

Las reflexiones producidas en las líneas drop pueden ser minimizadas disminuyendo su longitud. Así puede lograrse el más alto provecho posible de velocidad de transmisión y longitud de línea de bus con una topología de línea que sea lo más cercana posible a una estructura de línea única y terminada en ambos extremos como lo indica la figura 2.31.

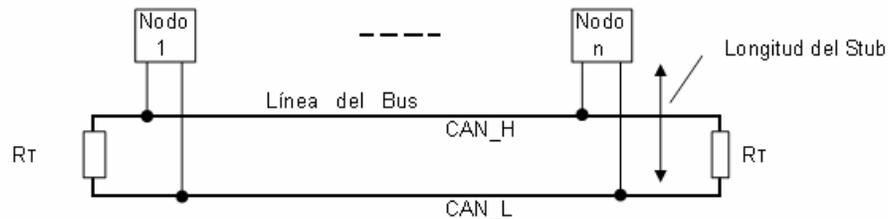


Figura 2.31: Topología del Bus de acuerdo a [ISO99-2]

El estándar CAN de alta velocidad [ISO99-2] especifica una topología como lo muestra la Figura 2.31 con resistencias de valor nominal 120 Ohms en los extremos para una velocidad de 1Mbit/s, una longitud máxima de bus de 40 m y una longitud máxima de stub de 30cm. Por supuesto, con velocidades de transmisión más bajas, pueden obtenerse longitudes de bus así como longitudes de stub más largas.

En aplicaciones industriales, muy a menudo no es posible conectar un nodo con cables drop muy cortos. Sin embargo, con la configuración apropiada de los parámetros de temporización de bit, el punto de muestreo de bit puede posicionarse al extremo de un tiempo de bit y por tanto minimizar el efecto de reflexiones de la señal. Entonces también pueden ser usadas líneas drop más largas. Cabe señalar que es necesario que las reflexiones de señal causadas por el cable drop sean atenuadas dentro del segmento de retardo de propagación. Una regla para estimar la longitud máxima aceptable de las líneas drop L_{dc} está dado por (3-7a):

$$L_{dc} < \frac{t_{Pr op_seg}}{50 \cdot t_p} \quad [m] \quad (3-7a)$$

con t_p retardo específico de propagación de la línea por unidad de longitud
 $t_{Pr op_seg}$ tiempo de retardo de propagación del segmento

De acuerdo a [PHI-4], la máxima longitud posible acumulada de todos los cables drop en la red puede ser estimada con la siguiente relación:

$$\sum_{i=1}^n L_{dc_i} < \frac{t_{Pr op_seg}}{10 \cdot t_p} \quad (3-7b)$$

Por medio de repetidores, puentes o gateways es posible superar las limitaciones de la topología básica de línea de redes CAN y así adaptar la topología de red de acuerdo a las necesidades de una aplicación específica.

Repetidor

Un repetidor transfiere una señal eléctrica desde un segmento físico de bus a otro. La señal es regenerada y transparentemente pasada a otro segmento. Así un repetidor divide un bus en dos segmentos físicamente independientes. Con respecto a la propagación de la señal un repetidor introduce un tiempo de propagación de señal adicional igual a su tiempo de retardo. El tiempo de retardo adicional introducido por un repetidor es aproximadamente de 250 a 350 ns reduciendo así la máxima longitud posible de la línea del bus en alrededor de 50 a 70 m. por repetidor (dependiendo de si un repetidor provee o no aislamiento galvánico). Un repetidor CAN bien diseñado también protegería los segmentos de bus conectados contra una falla eléctrica en el otro segmento correspondiente.

Áreas extendidas o potencialmente explosivas o contaminadas pueden ser puenteadas con un repetidor de fibra óptica CAN.

Modernos sistemas repetidores CAN de fibra óptica (fibra de vidrio) permite puentear distancias de hasta una distancia de 1 Km.

Debido al retardo de propagación adicional que un repetidor introduce, su uso básicamente reduce la máxima extensión posible de la red. Sin embargo, adaptando las condiciones geográficas se puede conseguir la longitud de línea necesaria con la ayuda de repetidores. Como ejemplo, la figura 2.32 muestra la estructura geográfica de una red para conexión de un número de líneas de manufacturación. La topología lineal del bus de la red mostrada requiere una longitud total de bus de 440 m. Así, la máxima velocidad de transmisión es limitada aproximadamente a 140KBits/s. Si la línea “bajada y subida” de una línea de manufacturación es reemplazada por un segmento eléctrico de bus separado que se acopla con el segmento restante por medio de un repetidor como muestra la Figura 2.33, la longitud total de una red será de solo 290 m, la máxima distancia a partir del retardo provocado por los repetidores es de solo 150 m (entre los nodos 6 y 12) y así la máxima velocidad de transmisión con este sistema es aproximadamente 400KBit/s.

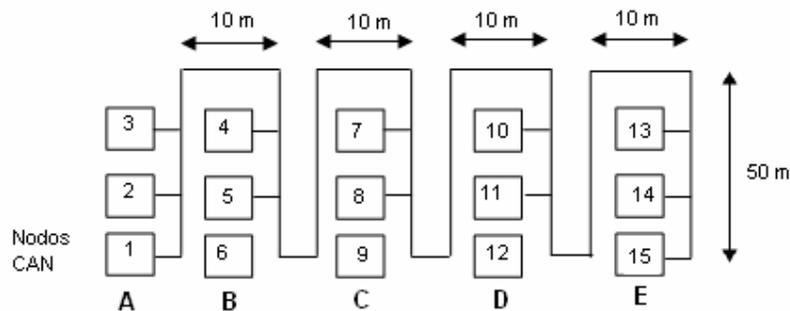


Figura 2.32: Conexión de líneas de manufacturación por medio de una topología lineal de bus

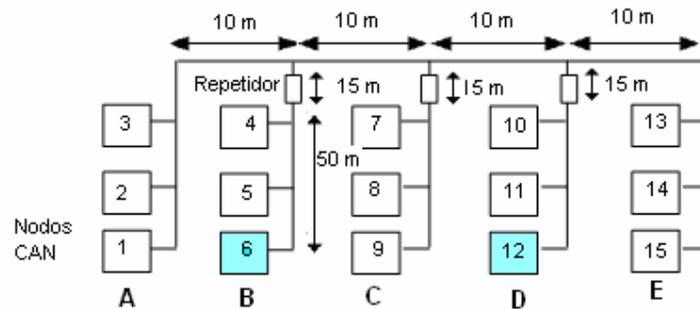


Figura 2.33: Conexión de líneas de manufacturación por medio de una topología optimizada de red

Este ejemplo muestra que por medio del uso de los repetidores es posible una adaptación óptima a las restricciones geográficas. Siendo así utilizados para el diseño optimizado de redes de topología extendida. En el mercado también se tiene disponible una combinación de varios repetidores dentro de una unidad llamada “acopladores estrella”. Este tipo de repetidor es útil para realizar redes con topología estrella. Por supuesto que por medio de los repetidores también se puede incrementar el número limitado de nodos CAN por segmento de red.

Puente

Un puente conecta a redes separadas sobre la capa enlace de datos, lo que permite que al integrar funciones de filtrado, sea posible realizar una estructura organizacional de redes multi-segmentos que envíe solo aquellos mensajes requeridos de un segmento de red a otro. De esta manera también es posible controlar y reducir la carga de diferentes segmentos de un bus.

Los puentes inherentemente proveen una función de almacenamiento y pueden enviar mensajes o partes de éstos en una transmisión independiente “demoradas en tiempo” (principio store and forward¹). Los puentes son diferentes de los repetidores porque los puentes almacenan y envían mensajes completos o partes de estos que no son locales al segmento de red, mientras que los repetidores envían toda la señal eléctrica.

¹ La trama completa es almacenada en el puente antes de enviarla al receptor con el propósito de verificar la su integridad.

Gateway

Un gateway provee la conexión de redes de diferentes protocolos. Una aplicación de gateway realiza la traslación de Unidades de Protocolo de Datos (PDU) entre diferentes sistemas de comunicación (Figura 2.34). El nivel del protocolo en que la conversión toma lugar caracteriza al gateway. Por ejemplo con un gateway de capa 7 toma lugar una conversión de PDUs de capa 7.

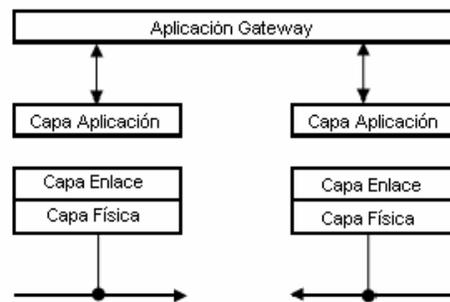


Figura 2.34: Conexión de dos sistemas de comunicación diferentes a través de un gateway de capa 7

En el mercado actualmente se tienen diferentes tipos de gateways CAN. Estos incluyen por ejemplo gateways entre CAN/CANopen/DeviceNet y AS-I, RS232/RS485, Interbus-S, Profibus o Ethernet/TCP-IP.

A través de un gateway redes CAN pueden ser conectadas a cualquier otro tipo de red incluyendo el Internet. Gateways CAN-Internet por ejemplo ofrecen interesantes características con respecto al mantenimiento y diagnóstico remoto de sistemas basados en CAN.

En la Figura 2.35 se muestra un ejemplo de estructura de red compleja usando repetidores, puente y gateways.

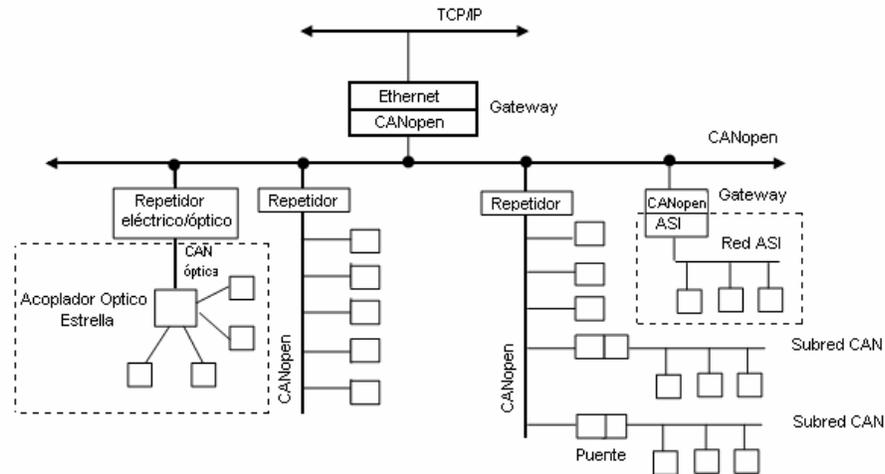


Figura 2.35: Estructura de una red con repetidores, puentes y gateways

2.2.4 ACCESO AL BUS

La conexión de los controladores CAN al bus generalmente es realizada por medio de un transceiver. Esto resulta en una conexión simple, no costosa y estandarizada con lo que se alcanza sin dificultad la interoperabilidad deseada en el nivel físico.

Esta conexión es designada como “Acceso al Medio Bus”. Para la conexión de un controlador CAN a un bus de 2 alambres en el mercado están disponibles una variedad de chips transceivers CAN de acuerdo a los diferentes estándares de capa física. Básicamente, la interfaz al medio eléctrico consiste de un amplificador de transmisión y otro de recepción.

Uno de los principales parámetros provistos por un transceiver CAN es el rango de voltaje de “modo común”. Este parámetro especifica la máxima diferencia de potencial posible entre nodos CAN, en que los niveles de señal pueden todavía ser correctamente interpretados por la circuitería de entrada de un transceiver.

Con bajos requerimientos con respecto a la longitud del bus, se pueden tomar medidas efectivas para reducir la emisión electromagnética para aplicaciones con velocidades de transmisión menores a 250 KBit/s (control slope).

Características adicionales de un transceiver CAN pueden ser la provisión del modo “stand-by”. En este modo, la parte transmisora del transceiver es apagada. El receptor monitorea si ocurre un nivel dominante en el bus. De esta forma un mensaje de “wake up” (o “despertarse”) puede ser reconocido. En este modo de operación muy poca potencia es consumida. Aplicaciones que consumen energía elevadamente pueden así ser apagadas y activadas de nuevo a través de las líneas del bus. Cuando es reconocido un mensaje de “wake up”, el controlador CAN (que puede ser operado normalmente también en modo stand-by) es despertado y de esta manera el nodo CAN se activa completamente.

Los diferentes estándares para interfaz físico del bus (descritos en la sección 2.2.5) pueden ser llevados a cabo usando los transceivers apropiados.

En particular, como transmisor, el transceiver provee las siguientes funciones:

- Provisión de suficiente capacidad de conducción de salida
- Protección del controlador contra sobrecargas
- Reducción de radiación electromagnética

Por su parte como receptor, un transceiver CAN realiza las siguientes funciones:

- Provisión de un nivel de señal recesivo definido
- Protección del comparador de entrada del controlador contra sobre voltajes en las líneas de bus
- Provisión del rango de voltaje de modo común para el comparador de entrada del controlador CAN
- Provisión de suficiente sensibilidad de entrada
- Detección de errores de bus tales como rotura de línea, cortos circuitos a tierra o a la alimentación así como cambio a operación asimétrica¹ de un solo cable

¹ El modo asimétrico es definido en el estándar ISO 11992 y en el estándar de tolerancia a fallos ISO 11898-3.

Una función adicional como el aislamiento galvánico¹ entre un transceiver y el controlador pueden ser implementados con poca circuitería adicional.

Un ejemplo de interfaz de bus óptico se muestra en la figura 2.36.

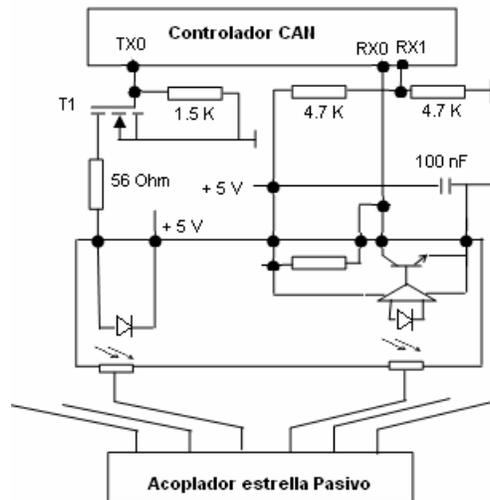


Figura 2.36: Interfaz de bus óptico

De acuerdo al transceiver utilizado se pueden tener tres modos de operación diferentes:

- Modo de alta velocidad que soporta una velocidad máxima de bus y/o la longitud.
- Modo de “control slope” que debería ser considerado si los alambres del bus que serán usados no tiene malla de tierra. En este modo la velocidad de salida puede ser cambiada o disminuida en forma intencional, p.ej reducida emisión electromagnética.

¹ Protección de interfaz que sirve para aislar circuitos intrínsecamente seguros de los circuitos que no son intrínsecamente seguros, según estándares mundiales. El aislamiento galvánico ofrece soluciones económicas para instrumentos y sistemas de control en lugares peligrosos.

- Modo de reserva (stand by) que es de gran interés en aplicaciones que se necesita ahorro de batería, cuando el consumo de energía del sistema necesita ser muy bajo. La reactivación del sistema es realizada por la transmisión de un mensaje.

La Figura 2.37 muestra el diseño básico de una unidad de acceso al medio de un nodo CAN.

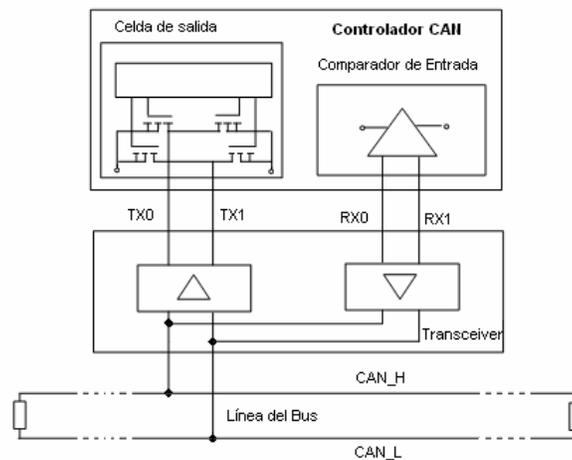


Figura 2.37: Nodo CAN con unidad de acceso al bus

2.2.5 ESTÁNDARES DE CAPA FÍSICA

Diferentes estándares especifican los niveles de señal, topología de red, número de nodos, así como los requerimientos para el medio y conectores. El estándar ISO 11898-2 [ISO99-2], para “comunicación de alta velocidad” en vehículos, es el más importante y también el recomendado por CAN-in-Automation para uso en aplicaciones industriales.

A continuación se presentan los estándares de capa física más importantes así como soluciones para el interfaz físico del bus.

2.2.5.1 Estándar ISO 11898-2

El estándar ISO 11898-2 [ISO99-2] forma la base para una serie de estándares y es el más importante de la capa física para redes CAN en automatización industrial así como en la industria automotriz. Este estándar describe las funciones de la unidad de acceso al medio y también algunas características de la interfaz.

Los datos principales de este estándar son:

- Velocidad de transmisión de hasta 1 MBit/s
- Longitud máxima del bus de 40m a 1 MBit/s¹
- El número total de nodos está limitado por la carga eléctrica (longitud de las líneas, uso de repetidores, procesamiento interno de los nodos) del bus
- Bus de dos alambres
- Seguridad de corto circuito.
- Impedancia característica de línea de 120 Ohms
- Rango de voltaje de modo común desde -2V (CAN_L) a +7V(CAN_H)

ISO 11898-2 especifica un bus de dos alambres terminado en ambos extremos con la impedancia específica de línea del medio. Las siguientes características eléctricas son especificadas por el bus de dos cables:

- | | |
|--|------------|
| • Máxima longitud a 1 MBit/s | 40 m |
| • Máxima longitud de una línea drop a 1 MBit/s | 30 cm |
| • Impedancia característica de línea | 120 Ohms |
| • Resistencia de línea (nominal) por metro | 70 mOhms/m |
| • Retardo de propagación específico nominal | 5 ns/m |

La topología de red debe ser diseñada lo más cercana posible a una estructura de una sola línea para evitar reflexiones de señal. La figura 2.38 muestra la

¹ Distancias mas altas son posibles a velocidades más bajas (Ver sección 2.2.1.5)

topología básica de una red CAN, mientras que la figura 2.39 muestra los niveles nominales del bus de acuerdo a ISO 11898-2.

Un receptor reconoce un bit recesivo si el nivel del voltaje diferencial de entrada V_{diff} ($V_{diff} = V_{CAN_H} - V_{CAN_L}$), esta por debajo de 0.5V, y un nivel dominante si este valor excede 0.9V.

Detección de un Bit Recesivo:

$$CAN_H < CAN_L + 0.5V$$

$$CAN_H - CAN_L < 0.5V$$

$$V_{diff} < 0.5V \quad (\text{Por lo general: } V_{diff} = 0V, \text{ con } CAN_H = CAN_L = 2.5V)$$

Detección de un Bit Dominante:

$$CAN_H > CAN_L + 0.9V$$

$$CAN_H - CAN_L > 0.9V$$

$$V_{diff} > 0.9V \quad (\text{Por lo general: } V_{diff} = 2V, \text{ con } CAN_H = 3.5V \text{ y } CAN_L = 1.5V)$$

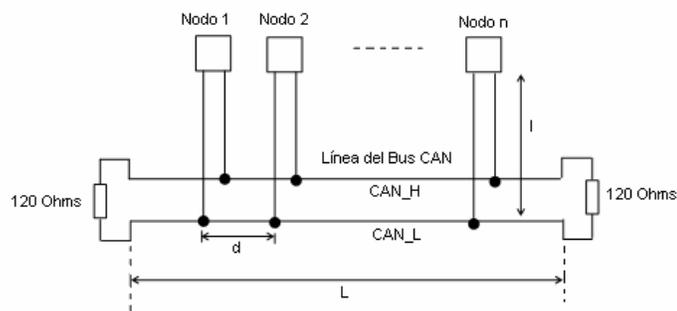


Figura 2.38: Topología de Bus de una red CAN de acuerdo a ISO 11898-2

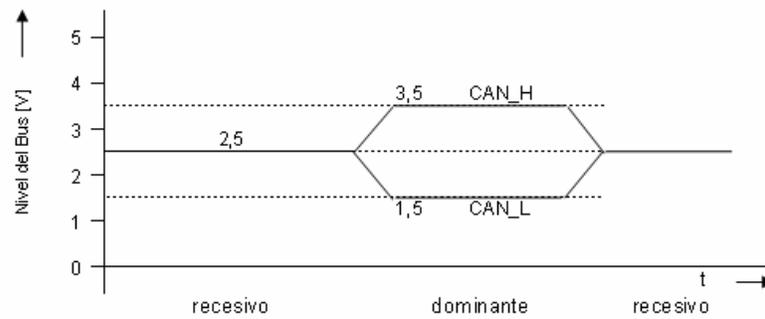


Figura 2.39: Niveles nominales del bus de acuerdo a ISO 11898-2

Unidad de Acceso al Medio Bus de acuerdo a ISO 11898-2

El acceso al medio de acuerdo a ISO 11898-2 es usualmente implementado por un chip transceiver CAN de alta velocidad. A parte del interfaz al medio, estos chips también realizan medidas de protección adicional, particularmente requeridas en vehículos motorizados. Las características más importantes de la unidad transceiver CAN son:

- Compatibilidad completa con ISO 11898-2
- Velocidades de transmisión de hasta 1 MBit/s
- Protección contra sobrecarga de temperatura
- Protección contra corto circuito a tierra o a la fuente de alimentación
- Bajo consumo de corriente en modo stand-by

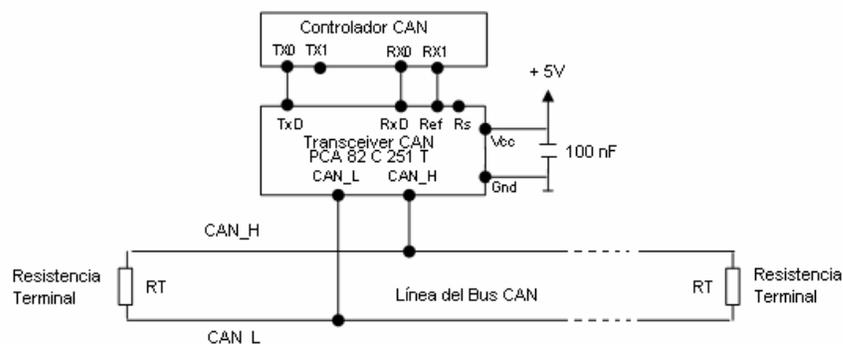


Figura 2.40: Conexión de un controlador CAN con el bus utilizando un Transceiver CAN Philips PCA 82C251

La figura 2.40 ilustra la conexión de un controlador CAN a las líneas del bus usando un transceiver CAN Philips PCA82C251T.

2.2.5.2 Estándar CiA DS-102

La realización de redes abiertas CAN es apoyada por las recomendaciones DS-102 de la asociación CiA (CAN-in-Automation) [CiA-102] para acceso al bus.

Esta propuesta está basada en ISO 11898-1 y -2 y además especifica:

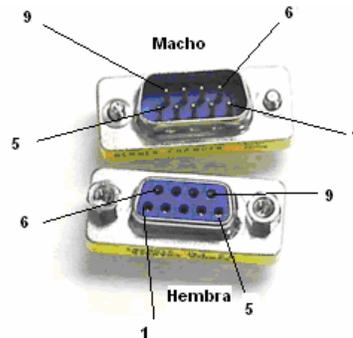
- Velocidades de transmisión estandarizadas de 10 kbit/s a 1 Mbit/s con recomendaciones para el establecimiento de parámetros de temporización de bit (Tabla 2.3).
- Recomendaciones para las líneas de buses
- Conectores y asignación de pines.

DS-102 especifica un par de alambres trenzados y/o blindados terminados en ambos extremos con la impedancia de línea y tierra común. Se especifica el uso de repetidores para incrementar el número de nodos del bus.

Vtx	Longitud máxima de línea	Vtx nominal	Cantidad de unidades tq	Longitud de una unidad de tiempo tq	Punto de muestreo
1 MBit/s	25 m	1 uS	8	125 nS	6 tq
800 kBit/s	50 m	1.25 uS	10	125 nS	8 tq
500 kBit/s	100 m	2 uS	16	125 nS	14 tq
250 kBit/s	250 m	4 uS	16	250 nS	14 tq
125 kBit/s	500 m	8 uS	16	500 nS	14 tq
50 kBit/s	1000 m	20 uS	16	1.25 uS	14 tq
20 kBit/s	2500 m	50 uS	16	3.125 uS	14 tq
10 kBit/s	5000 m	100 uS	16	6.25 uS	14 tq
Frecuencia de Oscilador:		16 MHz +/- 0.1%			
		Muestreo simple			
Modo de Sincronización:		Únicamente transición de recesivo a dominante			
SJW:		1 tq			
Segmento 2 de Fase:		2 tq			

Tabla 2.3: Velocidades de transmisión y parámetros de temporización del bus para longitudes de línea máxima de acuerdo a CIA DS 102

El conector definido es el SUB-D9, usado para aplicaciones industriales cuya asignación de pines se muestra en la figura 2.41



Pin	Señal	Descripción
1	-	Reservado
2	CAN_L	Señal CAN de nivel dominante
3	CAN_GND	Tierra CAN
4	-	Reservado
5	CAN_SHLD	Blindaje Opcional
6	GND	Tierra CAN opcional
7	CAN_H	Señal CAN de nivel recesivo
8	-	Reservado
9	CAN_V+	Fuente de voltaje Vcc opcional externa

Figura 2.41: Conector CAN y asignación de pines de acuerdo a CíA DS-102

2.2.5.3 Tolerancia a Fallos de la Interfaz del Bus de acuerdo a ISO11898-3

Para requerimientos más bajos con respecto a velocidad de transmisión y longitud del bus, la “especificación de baja velocidad tolerante a fallos” está disponible como una forma alternativa de interfaz al bus. Esta especificación, desarrollada por las compañías Philips y Bosch, reemplaza el estándar CAN de baja velocidad ISO-11519 y representa la tercera parte del estándar ISO [ISO99-3]. El uso pensado del interfaz de bus tolerante a fallos es primeramente para redes electrónicas de confort en vehículos motorizados.

La topología bus no está limitada a una estructura lineal con longitudes cortas de líneas drop. Otra ventaja es la posibilidad de transmitir datos asimétricamente sobre, solamente, una línea del bus en caso de falla eléctrica de las líneas del bus.

Los parámetros más importantes de esta especificación son:

- Velocidades de transmisión de hasta 125 kBit/s
- La máxima longitud del bus depende de la velocidad de transmisión y está definida por la carga capacitiva de las líneas del bus.
- Hasta 32 nodos por red
- Transmisión simétrica de la señal si se tiene disponible un par de alambres
- Prueba de corto circuito en un rango de voltaje entre -6 V ... +16 V
- Corriente de salida del transmisor > 1mA
- Rango de voltaje de modo común entre -2 V ... +7 V
- Fuente de alimentación de 5 V

La figura 2.42 muestra los niveles nominales de bus de esta especificación. Los cuales son diferentes para niveles recesivo y dominante; así niveles de bus recesivo y dominante también son detectados comparándolos con un nivel de referencia de por ejemplo 2.5 V.

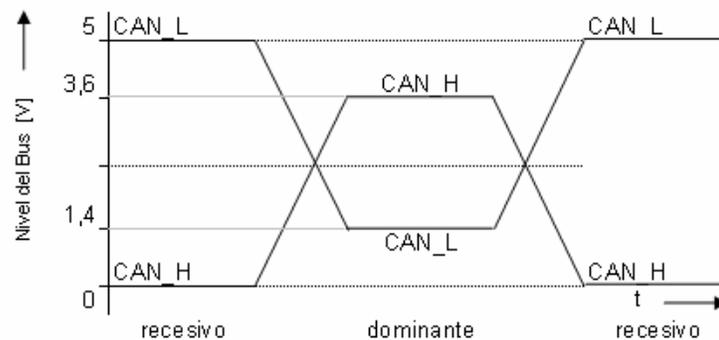


Figura 2.42: Niveles nominales de señal de la especificación de tolerancia a fallos

La especificación de tolerancia a fallos considera la detección y manejo de las siguientes condiciones de error del bus:

- Interrupción de CAN_H
- Interrupción de CAN_L
- Corto circuito de CAN_H con Vcc
- Corto circuito de CAN_L con GND
- Corto circuito de CAN_H con GND
- Corto circuito de CAN_L con Vcc
- Corto circuito de CAN_H con CAN_L

Las diferentes condiciones de error pueden ser detectadas comparando ambas líneas CAN. Desconectando la línea defectuosa y cambiando a una operación asimétrica, la comunicación puede continuar sobre la línea restante del bus. Sin embargo la alta inmunidad a la interferencia eléctrica de la transmisión simétrica se pierde por esta causa.

Conexión al bus de acuerdo al estándar de tolerancia a fallos

Se puede realizar una interfaz de bus que cumple la especificación tolerante a fallos descrita anteriormente por medio de un transceiver CAN TJA 1054 de Philips Semiconductors Company.

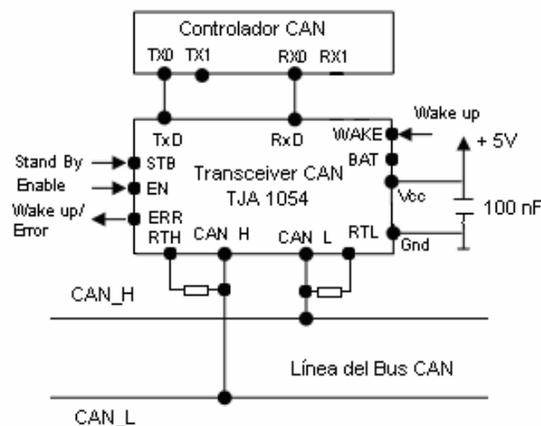


Figura 2.43: Interfaz de Bus con transceiver Philips TJA 1054 de acuerdo al estándar tolerante a fallos

La Figura 2.43 ilustra una conexión de bus realizada con el chip TJA 1054.

El chip transceiver empleado soporta un completo manejo de errores incluyendo detección de errores de bus y cambio automático a transmisión asimétrica de señal. Errores del bus son reportados al host controlador¹. El modo regular de transmisión es automáticamente re-establecido cuando la condición de error desaparece.

2.2.5.4 Estándar SAE J2411

“Single Wire CAN” (Bus de un solo alambre) fue desarrollado originalmente por la Corporación General Motors como una alternativa para el “single-wire UART Bus” de acuerdo a SAE J1850 [SAE01-01]. Este estándar fue la base para el estándar SAE J2411 que es una alternativa adicional para aplicaciones de redes CAN con bajos requerimientos respecto a velocidad de transmisión y longitud de bus. Los parámetros básicos de este estándar son:

- Comunicación a través de una única línea de bus
- Velocidad nominal de transmisión 33,33 kBit/s en modo normal
- Modo de alta velocidad a 83,33 kBit/s para diagnósticos
- Hasta 32 nodos por red
- Modo “stand by”

El principal uso de “Single wire CAN” es para redes electrónicas de confort en vehículos motorizados. Un alambre único no blindado es definido como el medio bus. Debido a la baja velocidad de datos la topología bus no está limitada a una estructura lineal para distancias cortas de líneas drop. La Figura 2.44 muestra los niveles nominales de bus para modo normal de acuerdo a SAE J2411.

¹ Ver sección “controladores del protocolo CAN”.

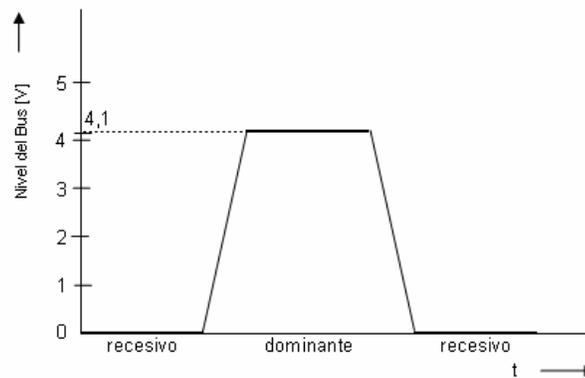


Figura 2.44: Niveles nominales del bus para modo normal de acuerdo a SAE J2411

Para bajar programas o cargar datos de diagnóstico, la red puede cambiar al modo de alta velocidad. La velocidad de transmisión en este modo de operación es 83,33 kBit/s.

El estándar SAE J2411 incluye capacidad “stand by” permitiendo comunicación normal entre varios nodos mientras deja a otros en un estado “suspendido” no perturbado. Esto se logra controlando los niveles lógicos del bus tal que todos los nodos deben despertar cuando reciben un mensaje que es transmitido a un voltaje elevado de $2/3 \cdot V_{ss}$. La comunicación con voltaje en modo normal no perturbará a los nodos en modo stand by.

Acceso al bus de acuerdo al estándar de alambre único CAN SAE J2411

Una conexión al bus que cumpla con el estándar SAE J2411 puede ser realizada con ayuda de una unidad de transceiver CAN TLE 6255 de Infineon Semiconductors Company. La Figura 2.45 ilustra un acceso al bus implementado por medio del TLE 6255. La unidad de transceiver provee modos de operación: normal, alta velocidad y “stand by”.

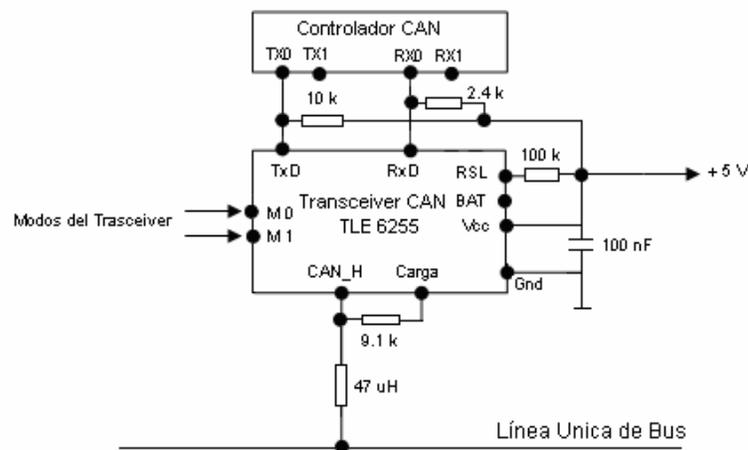


Figura 2.45: Acceso al Bus de acuerdo a SAE J2411

2.2.5.5 Estándar ISO 11992

ISO 11992 describe una propuesta adicional para usar CAN en redes de baja velocidad. Está basado en una conexión punto a punto para uso en vehículos de remolque y sus trailers y es también conocido como el estándar “truck to trailer”.

Los parámetros básicos de este estándar son:

- Conexión punto a punto para vehículos con un trailer
- Conexión en cadena para vehículos con dos trailers
- Velocidad de transmisión de 125 kbit/s
- Longitud de línea máxima de 40 m
- Transmisión simétrica sobre un par de alambres.
- Manejo de errores del bus
- Fuente de voltaje $V_s=12\text{ V}$ o 24 V

Un par trenzado no blindado es definido como el medio del bus. Debido al bus pequeño y a la relativamente baja velocidad de transmisión no se especifica resistencia en los extremos de las líneas del bus. ISO 11992 especifica la máxima capacitancia por longitud, la impedancia de línea así como características

de los conectores. La figura 2.46 muestra los niveles nominales del bus de acuerdo a ISO 11992. Los niveles del bus dependen de la fuente de voltaje V_s de los vehículos. Los niveles de bus para vehículos de 12 V y 24 V son listados en la tabla 2.4.

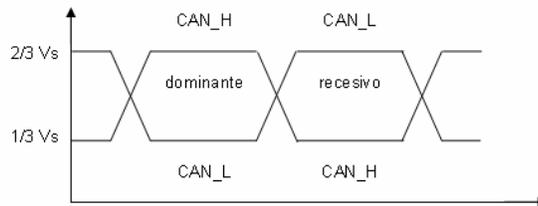


Figura 2.46: Niveles nominales del bus de acuerdo a ISO 11992

	12 V Voltaje del vehículo	24 V Voltaje del vehículo
Nivel Dominante		
CAN_H	6.010.6V	10.6 ... 21.3 V
CAN_L	3.05.3V	5.3 10.6V
Nivel Recesivo		
CAN_H	3.05.3V	5.3 10.6V
CAN_L	6.010.6V	10.6 ... 21.3 V

Tabla 2.4 : Niveles del bus de acuerdo a ISO 11992 en función de la fuente de voltaje del vehículo

2.2.6 CONTROLADOR DEL PROTOCOLO CAN

Todas las funciones necesarias para procesamiento de capa 1 y 2 del protocolo CAN así como funciones adicionales de apoyo son implementadas en los controladores CAN (Controladores del Protocolo). Esta funcionalidad es provista a un “Controlador de Host”¹ a través de una interfaz simple, el controlador de host solamente tiene que pasar al controlador CAN los mensajes a ser transmitidos, o leer mensajes recibidos por éste.

¹ El “Controlador de Host” es entendido como el procesador involucrado en la aplicación. Como se indicará en la sección 4.2.4, es posible que el controlador de host y el controlador CAN constituyan unidades separadas o sean integradas en una sola.

Debido a los requerimientos de tecnología la función del transceiver generalmente no es integrada en el controlador CAN.

2.2.6.1 Funciones de un Controlador CAN

Un controlador CAN realiza todas las tareas relacionadas a la capa 1 y 2 del protocolo CAN, tales como:

- Arbitraje del bus
- Serialización y de-serialización de las tramas a ser enviadas o recibidas
- Cálculo y verificación de la secuencia de redundancia cíclica
- Detección y manejo de errores
- Construcción de mensajes en formato CAN
- Inserción y anulación de bits de relleno
- Generación y chequeo del bit ACK
- Sincronización del flujo de bits recibidos

La realización de todas las funciones relacionadas al protocolo en un hardware controlador separado libera al “controlador de host” de estas tareas críticas de tiempo. Además, los controladores CAN realizan funciones adicionales con respecto al filtrado de mensajes, almacenamiento de mensajes y respuesta automática a solicitudes remotas de mensajes.

Filtrado de mensajes

La transmisión de datos a través de redes CAN está basada en el principio de broadcast. Esto significa que todos los mensajes transmitidos al bus están disponibles para todos los nodos y son recibidos por sus controladores. En muchos casos un cierto nodo solamente estará interesado en pocos mensajes de los que se les ha enviado. Por esta razón es apropiado implementar un mecanismo adicional de filtrado de mensajes en el controlador que asegure que un “controlador de host” sea informado de la recepción de un nuevo mensaje solamente si éste es relevante para el mismo. Esto es llamado “filtro de

aceptación”. Para realizar esta función, el controlador debe ser informado de todos los identificadores de mensajes en los que el nodo está interesado.

Manejo de Mensajes

Con respecto al manejo de mensajes, es posible distinguir entre dos principios de implementación:

- Almacenamiento temporal de todos los mensajes en un buffer receptor y en un buffer transmisor
- Almacenamiento de un número limitado de mensajes en buffer específicos de transmisión o recepción

Los dos principios normalmente están referidos a implementaciones “BasicCAN” o “FullCAN” las cuales serán revisadas en el Capítulo 4.

Provisión de mensajes remotamente solicitados

Debido a los diferentes principios de almacenamiento de mensajes, los principios BasicCAN y FullCAN también difieren con respecto al manejo de mensajes solicitados remotamente, esta característica también será mencionada posteriormente.

CAPITULO 3

PROTOCOLOS DE CAPAS SUPERIORES

De acuerdo con el modelo de referencia de comunicación de datos (Modelo OSI) el protocolo CAN solamente cubre la capa física y la capa enlace de datos. La capa enlace CAN provee los dos siguientes servicios no orientados a conexión:

- Transmisión no confirmada de mensajes CAN, y
- Solicitud Remota no confirmada de mensajes CAN

Con una solicitud remota de transmisión un usuario de capa 2 CAN puede, por un lado, solicitar a su entidad¹ de capa 2 iniciar la transmisión de una solicitud remota de transmisión a través del bus para lo cual tiene que proveer el identificador y la longitud de datos del mensaje solicitado, y por otro lado puede proveer el mensaje solicitado y hacer que la entidad del servicio de capa 2 correspondiente lo transmita. Debido a la naturaleza broadcast de la transmisión utilizada en CAN no es posible la confirmación de mensajes, sin embargo por medio del principio de señalización de errores (sección 2.1.2.3) se garantiza que los datos sean transmitidos y recibidos sin errores.

La simple funcionalidad respecto a transmisión de datos provista por el protocolo CAN es una de las principales razones para su empleo en un gran número de aplicaciones, sin embargo se requieren funciones adicionales para la implementación de sistemas distribuidos² más sofisticados (Por ejemplo sistemas de automatización de una fábrica donde el sistema distribuido controla los robots y máquinas en la línea de montaje), dichas funciones pueden ser: servicios que permitan la transmisión de bloques de datos de más de 8 bytes, un modelo de

¹ Elementos de hardware o software.

² Es un sistema formado por varios componentes, hardware y/o software, localizados en diferentes dispositivos, conectados en red, que se comunican y coordinan a través de mensajes. Diseñados para que muchos usuarios trabajen en forma conjunta.

comunicación cliente-servidor confirmado, una administración de red apropiada, funciones y métodos para la asignación de identificadores o para el establecimiento de parámetros específicos de los nodos. Además debido a que la inter-operabilidad e inter-cambiabilidad de dispositivos de diferentes fabricantes es fundamental en la tecnología de automatización, es necesario proveer una descripción estandarizada de la funcionalidad de los dispositivos y definir perfiles¹ estandarizados de los mismos.

En este sentido el desarrollo de CAL, CAN Application Layer (Capa Aplicación CAN), por CiA constituye un paso importante en esta dirección. CAL no define aplicaciones específicas, por ello estas deben ser definidas por un sistema específico de implementación. Esto significa que los servicios provistos por CAL pueden ser usados eficientemente de acuerdo a los requerimientos específicos de una aplicación. CAL entonces provee una plataforma de comunicación estandarizada para la implementación óptima de aplicaciones distribuidas específicas.

3.1 CAL – CAPA APLICACIÓN CAN

La provisión de una capacidad extendida de comunicación basada en una capa aplicación para sistemas distribuidos CAN fue el punto de inicio para la especificación del estándar CAL (CAN Application Layer) por CAN in Automation [CIA-200]. CAL permite la descripción de aplicaciones distribuidas a través de objetos² de comunicación estandarizados y define los correspondientes protocolos para el intercambio de estos objetos.

Además, tipos de datos estándar y reglas de codificación de bits para su transmisión son definidas, también se especifican servicios y protocolos para administración de red y establecimiento de parámetros específicos al nodo. Como un elemento adicional de servicio, CAL describe un método eficaz para la asignación de identificadores de mensaje a objetos de comunicación.

¹ Funciones estandarizadas que describen el funcionamiento de un dispositivo.

² Datos, funciones o parámetros de un dispositivo que definen su funcionalidad.

Con la definición de CAL, se puso gran énfasis en mantener acceso directo a los servicios de capa 2, es decir que no se introdujo información de control (protocol overhead) adicional para la transmisión de datos de proceso. Las implementaciones basadas en CAL por tanto son tan rápidas como aquellas basadas en dos capas.

3.1.1 ARQUITECTURA DEL PROTOCOLO

La arquitectura de protocolo recomendada por CiA para redes industriales basadas en CAN corresponde a un stack de protocolos consistente de las capas 1, 2 y 7 de acuerdo al modelo de capas ISO/OSI (Figura 3.1). Aquí las capas 1 y 2 representan el estándar ISO-CAN [ISO99-1] [ISO99-2], y la capa 7 el estándar CAL [CIA-200] recomendado por CiA.

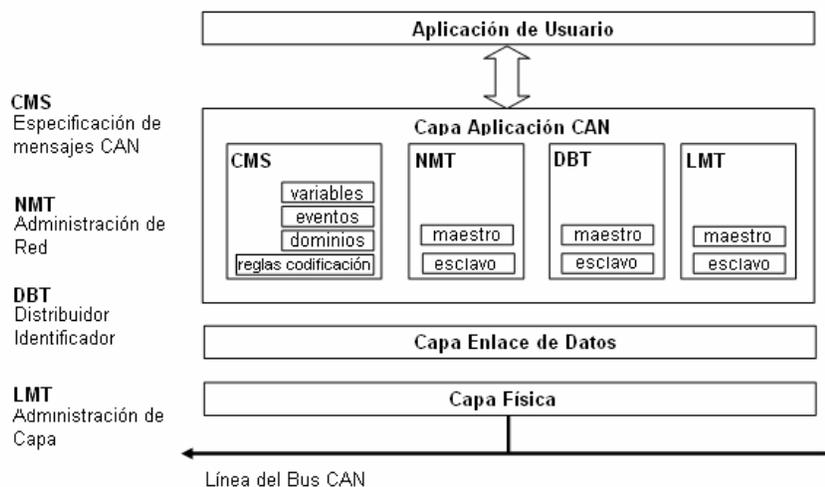


Figura 3.1: Sistema de Comunicación CAN para aplicaciones industriales con CAL de acuerdo a CiA

De acuerdo con la Figura 3.1, CAL provee un proceso de aplicación con los siguientes elementos básicos de servicio:

Especificación de Mensajes CAN (CMS) (CAN Message Specification)

A través de este elemento de servicio, se definen objetos estandarizados de comunicación tales como variables, eventos o dominios que describen la

funcionalidad de la aplicación accesible a través de la red y que se provee al proceso de aplicación en forma de servicios y protocolos correspondientes.

Administración de Red (NMT) (Network Management)

Servicios y protocolos NMT estandarizados posibilitan la configuración, administración y monitoreo de los nodos. De tal forma que un nodo en particular, llamado “Maestro NMT” toma el rol de coordinador mientras que el resto actúa como “Esclavos NMT” con respecto a la administración de la red.

Distribuidor Identificador (DBT) (Identifier Distributor)

Este elemento de servicio provee asignación dinámica de identificadores de mensajes a objetos de comunicación durante la inicialización del sistema.

Capa Administración (LMT)

A través de este elemento de servicio, un cierto nodo (Maestro NMT) puede establecer parámetros específicos de una capa, y transmitirlos a otros nodos (Esclavos NMT) a través de la red.

CAL describe la interacción entre procesos de aplicación de acuerdo al modelo cliente-servidor (ver sección 1.6). Un servicio CAL por tanto es solicitado por medio de una “primitiva solicitante”. A través de una indicación de servicio correspondiente, CAL informa a un proceso de aplicación remoto que un cierto servicio CAL ha sido solicitado. A través del tipo de servicio CAL, está implícitamente definida cual primitiva interactúa con CAL. CAL distingue los siguientes tipos de servicios (Figura 3.2):



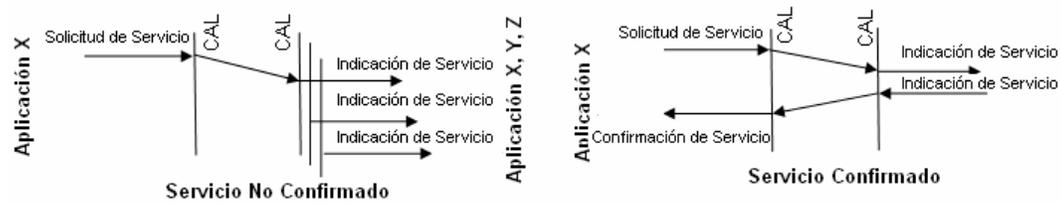


Figura 3.2: Tipos de servicios CAL

Servicios Locales

Por medio de una solicitud el proceso de aplicación pide que un servicio sea llevado a cabo localmente por CAL.

Proveedor de Servicios Iniciados

A través de este tipo de servicio, el proceso de aplicación es informado de un evento detectado por CAL por medio de una primitiva indicadora.

Servicios Sin Confirmación

La solicitud de un servicio sin confirmación por un proceso de aplicación del cliente causa que la CAL del cliente transmita un mensaje de solicitud correspondiente a otros nodos (servidores). La solicitud de servicio es indicada a sus procesos de aplicación a través de una indicación de servicio correspondiente, para que ellos puedan proveer el servicio pedido. Para servicios sin confirmación, puede haber varios proveedores de servicio (servidores).

Servicios Confirmados

Por medio de un servicio confirmado, un proceso de aplicación (cliente) puede pedir a un nodo remoto específico (servidor) proveer cierto servicio. La aplicación del servidor indica el resultado de la respuesta de servicio (positiva o negativa) a través de una primitiva de respuesta. La respuesta del servidor es transmitida por su CAL por medio de ciertos mensajes CAN al nodo solicitante (CAL del cliente) y provista a la aplicación del cliente por medio de una primitiva de confirmación.

3.1.2 ESPECIFICACIÓN DE MENSAJES CAN (CMS)

CMS define objetos de comunicación básicos y mejorados. Los objetos CMS básicos son implementados directamente a través de los servicios de la capa enlace de datos, mientras que los objetos CMS mejorados requieren de la introducción de protocolos y servicios adicionales. Entre los objetos básicos se tienen: variables solo de lectura, variables solo de escritura y eventos no controlados; mientras que entre los mejorados: variables multiplexadas, variables de lectura-escritura, dominios y eventos controlados.

En general CMS define los siguientes objetos de comunicación:

- Variables
- Eventos
- Dominios

Las “Variables CMS” pueden ser del tipo básico o multiplexado. Con multiplexación de variables, varias de éstas pueden enviarse en un mensaje. Esto reduce el número de identificadores de mensajes.

Variables con acceso de tipo “solo lectura” pueden ser usadas por un cliente para leer datos desde un nodo remoto.

Variables con tipo de acceso “solo escritura” pueden ser usadas por un cliente para proveer datos a uno o más servidores o para solicitar a uno o más servidores ejecutar un comando. El cliente no conocerá el resultado de ésta solicitud (servicio no confirmado).

Con variables de tipo de acceso “lectura-escritura” un cliente puede reunir los datos actuales del servidor o solicitar a un servidor ejecutar un comando y ser informado de los resultados de ésta acción (servicio confirmado).

“Eventos” CMS pueden ser usados para modelar un comportamiento asíncrono. La ocurrencia de un evento es detectada por un servidor y puede o no ser

notificada al cliente(s). Un evento puede proporcionar información adicional acerca de qué causó su ocurrencia. Eventos no controlados se refiere a eventos que se notifican a cualquier cliente que está interesado. Eventos controlados en cambio, es cuando un evento puede notificarse a lo sumo a un cliente. El cliente puede controlar si el servidor notifica o no el evento cuando este ocurra.

“Dominios básicos” CMS pueden ser usados para transferir un bloque de longitud arbitraria de datos desde un cliente a un servidor y viceversa. El contenido de un bloque de datos es específico a la aplicación. “Dominios multiplexados” pueden ser usados para transferir múltiples conjuntos de datos desde un cliente a un servidor y viceversa. El cliente puede controlar a través de un “multiplexor” cuál conjunto o juego de datos es transferido. Un dominio es transferido como una secuencia de segmentos. Antes de transferir los segmentos existe una fase de inicialización donde cliente y servidor pueden prepararse para la transferencia. Siempre es el cliente quien toma la iniciativa para iniciar una transmisión. Ambos, cliente o servidor pueden cancelar la transferencia de un dominio.

El dominio CMS multiplexado es usado en CANopen para la transmisión del llamado “Objeto de Servicio de Datos” (SDO) con un multiplexor CANopen específico (Ver sección 3.2.1.2).

CMS también determina cómo la información intercambiada entre procesos de aplicación a través de servicios CMS será codificada dentro de los mensajes CAN requeridos mediante reglas de codificación. Estas reglas, que se aplican a todos los objetos y servicios CMS, describen el orden de transmisión de los bits a través del bus CAN.

CAL solo especifica una capa de comunicación, esto no describe el significado de los mensajes transmitidos. Por tanto los datos transmitidos en los mensajes CAN deben ser especificados separadamente por el fabricante del dispositivo o pueden ser definidos por un perfil de aplicación específica¹.

¹ En caso de CANopen, estas definiciones son provistas en el “perfil estandarizado del dispositivo” el cual es parte del diccionario de objetos de un dispositivo.

CMS tampoco especifica ningún identificador para mensajes CAN de aplicación particular. La asignación de identificadores es considerada como responsabilidad del “integrador de sistema”¹. En los perfiles de aplicación en principio es concebible asignar identificadores fijos para mensajes CAN, sin embargo esto lleva rápidamente a la definición de un número muy grande de identificadores o a que los mismos identificadores sean usados en diferentes perfiles para funciones diferentes. Esto significaría, entre otras cosas, que un dispositivo no pueda ser usado en diferentes perfiles o aplicaciones.

3.1.3 ASIGNACIÓN DE IDENTIFICADORES DE MENSAJE

La asignación de identificadores a mensajes CAN es fundamental para la implementación de redes CAN. Un método por el cual la asignación de identificadores no se realiza estática sino dinámicamente es particularmente ventajoso. Tal método es implementado por el elemento de servicio “distribuidor” (DBT) definido en CAL.

Como los tiempos efectivos de latencia en un sistema distribuido son determinados principalmente por la asignación de identificadores (prioridades) y tiempos de impedimento² en los mensajes CAN requeridos por un sistema, es útil que el “integrador de sistema” pueda fácilmente establecer estos valores. Dispositivos que se comunican entre sí, es decir dispositivos que son cliente o servidor de un cierto objeto CMS deben usar los mismos identificadores para sus mensajes CAN. Sin embargo, solo el integrador del sistema conoce cuáles dispositivos se comunican entre sí y por medio de cuáles identificadores lo hacen. Una razón adicional por la que un nodo debería tener la posibilidad de asignación libre de identificadores es que también debería ser posible la operación de varios dispositivos idénticos en un sistema. Sin embargo, no debe haber más de un transmisor para el mismo mensaje CAN.

¹ Un “integrador de sistema” implementa un sistema de automatización distribuido hecho de dispositivos de diferentes fabricantes, y es responsable por la operabilidad de todo el sistema.

² Tiempo mínimo que se debe esperar entre dos invocaciones consecutivas de un mensaje.

El elemento de servicio distribuidor de CAL (DBT) por tanto especifica servicios así como protocolos correspondientes para la distribución de identificadores de mensajes CAN y tiempos de impedimento.

Para servicios estandarizados CAL, tales como administración de red, distribución de identificador o administración de capas, varios identificadores de mensajes son reservados.

3.1.4 ADMINISTRACIÓN DE RED

El elemento de servicio “administración de red” (NMT) definido bajo CAL provee la funcionalidad necesaria para satisfacer tareas en relación a la configuración, inicialización y monitoreo de los nodos. La funcionalidad NMT está basada en un esquema lógico maestro-esclavo, en la que un nodo del sistema realiza la función de maestro NMT.

NMT ofrece servicios para:

- Control de Módulo
- Control de Errores
- Configuración

A través de servicios de control de módulo el maestro NMT inicializa a los esclavos NMT que deseen tomar parte en la aplicación distribuida permitiéndoles comunicarse entre ellos. Con los servicios de control de errores, el maestro NMT detecta¹ fallas remotas en una red CAN. Por medio de los servicios de configuración un maestro NMT puede cargar y descargar datos de configuración (parámetros, datos o código ejecutable) desde o hacia un nodo.

3.1.5 ADMINISTRACIÓN DE CAPAS

Además de los elementos de servicio CMS, NMT y DBT de la capa aplicación de CAN (CAL) existen servicios adicionales para la administración específica de

¹ Fallas remotas son detectadas por el protocolo “Guardia de Nodo” (Sección 3.2.2.2)

capas. Estos servicios permiten la configuración de la dirección NMT de un esclavo NMT, consistente del nombre del nodo, identificador del nodo y parámetros de temporización de capa física. Varios servicios y protocolos correspondientes son provistos para este propósito.

3.2 CANopen

CANopen es una aplicación estandarizada para sistemas de automatización industrial distribuida y fue definida basada en CAN así como en CAL. CANopen es un estándar de CiA (CAN in Automation) y ha llegado a ser extensamente utilizado desde su lanzamiento. En Europa es considerado el estándar por defecto para la implementación de sistemas industriales basados en CAN.

CANopen está basado en el llamado “perfil de comunicación”, que especifica los mecanismos subyacentes de comunicación y su definición [CIA-304, CIA-302]. La funcionalidad de un tipo de dispositivo en particular está definida en el perfil del dispositivo.

El elemento central de CANopen es la descripción de la funcionalidad, parámetros y datos de un dispositivo por medio de un “Diccionario de objetos” (OD). El diccionario de objetos está dividido en una sección de especificaciones generales acerca del dispositivo tales como identificación del dispositivo, nombre del fabricante y parámetros de comunicación, y en otra sección que describe la funcionalidad, parámetros y datos específicos del dispositivo. Una “entrada” (“objeto”) del diccionario de objetos es identificada por un índice de 16 bits y un sub-índice de 8 bits. Las entradas del diccionario de objetos están implementadas para proveer los “objetos de aplicación” de un dispositivo, tales como por ejemplo entrada y salida de señales, parámetros del dispositivo, funciones del dispositivo o variables estandarizadas de red.

La Figura 3.3 muestra el modelo básico de un dispositivo CANopen con la interfaz de comunicación para conectar el dispositivo a la red, el proceso de aplicación y el diccionario de objetos que actúa como interfaz entre la red y el proceso de aplicación.

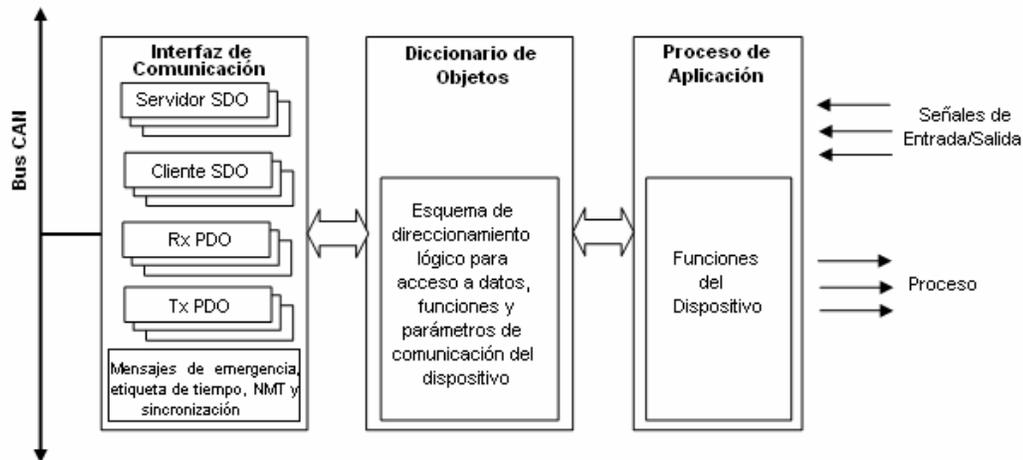


Figura 3.3: Modelo de un dispositivo CANopen

Con redes CANopen es posible conectar hasta 127 nodos. CANopen recomienda el uso de velocidades de transmisión y longitudes del bus de acuerdo al estándar de capa física CiA DS 102 (sección 2.2.5.2).

3.2.1 MODELO DE COMUNICACIÓN CANopen

Cualquier dispositivo CANopen puede ser visto como un dispositivo genérico, el cual en un extremo está conectado al bus CAN y por el otro a la Entrada-Salida de datos de la aplicación específica. La aplicación es el conocimiento clave del fabricante. La interfaz entre la aplicación y CAN está comprendida por un diccionario de objetos. El diccionario de objetos es único para cualquier dispositivo CANopen y representa acceso completo a su aplicación implementada en términos de datos y configuración. Para obtener acceso al diccionario de objetos cada dispositivo CANopen debe contener un stack de protocolos CANopen. Este stack de protocolos consiste de diferentes funciones para diferentes propósitos: Objeto de proceso de datos (PDO), objeto de servicio de datos (SDO), administración de red (NMT), y servicios especiales del sistema.

El modelo de comunicación CANopen provee dos tipos de mecanismos de comunicación:

- Transmisión no confirmada de tramas de datos, de hasta 8 bytes como máximo, para transferir datos de procesos (a través de “PDOs”) sin información de control adicional.
- Transmisión confirmada de datos (a través de “SDOs”) entre dos nodos con acceso directo a las entradas del diccionario de objetos del nodo direccionado. Usada principalmente para configuración del sistema, y también para transferir tramas más largas de datos entre los nodos, en cuyo caso se aplica el protocolo de fragmentación.

3.2.1.1 Objeto de Proceso de Datos (PDO)

La funcionalidad de un dispositivo CANopen es definida por objetos de aplicación¹. Los PDOs constituyen los “vehículos de transporte” para la transmisión de éstos objetos (datos de procesos). Un PDO es transmitido por un solo nodo (productor), pero puede ser recibido por más de uno (consumidores). Los “datos de procesos” transmitidos por un productor PDO pueden contener un máximo de 8 bytes. La transmisión de un PDO no es confirmada y requiere de la asignación de un identificador de mensaje CAN que defina el significado² de los datos transmitidos.

La Figura 3.4 muestra el ejemplo de una posible estructura de comunicación con PDOs.

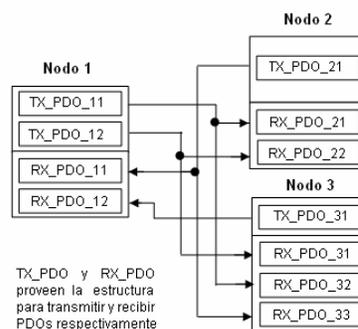


Figura 3.4: Ejemplo de una estructura de comunicación con PDOs

¹ Parámetros, datos y funciones de un dispositivo por ejemplo el valor de una salida o entrada digital, valor de la velocidad de un dispositivo, estado del dispositivo.

² La función de mapeo PDO también contribuye a definir el significado de los datos transmitidos.

Ya que un PDO está implícitamente definido por el identificador de mensaje y por su transmisión limitada a un máximo de 8 bytes, la transmisión de PDOs no requiere de información de control adicional.

[CIA-301] además ha definido un objeto PDO extendido (PDO Multiplexado, MPDO). Con el cual es posible utilizar un solo identificador PDO (un solo PDO pero multiplexado) para transferir una multitud de parámetros a un grupo de nodos. La transferencia de datos con MPDOs puede ser orientada al nodo o al mensaje, para especificar esta situación en el campo de datos debe ser indicada la dirección del nodo destinatario o fuente, respectivamente, además del índice y subíndice de la entrada del diccionario de objetos, es decir que a diferencia de un PDO éste si presenta información de control.

Tipos de Transmisión PDO

Los tipos de transmisión de un PDO son usados para definir el evento que causó su transmisión. Transmisiones PDO a más de ser conducidas por un evento interno (especificado en el perfil del dispositivo), también pueden ser provocadas por un temporizador interno o por solicitudes remotas (hechas por uno o varios consumidores PDO).

En general existen dos tipos de transmisión PDO: “asincrónica” y “sincrónica”. Con transmisión asincrónica un PDO es transmitido al proceso de aplicación tan pronto como se detecta la ocurrencia de un evento específico. Por otro lado con transmisión sincrónica un PDO no es transmitido al proceso de aplicación hasta que un “objeto de sincronización” sea recibido por el productor PDO. Cuando un productor transmite un PDO después de recibir solamente un objeto de sincronización se hace referencia a un “PDO sincrónico acíclicamente transmitido”. De otro modo, si un PDO siempre es transmitido después de un número específico de objetos de sincronización¹ toma lugar una “transmisión cíclica de un PDO sincrónico”.

¹ El objeto de sincronización es transmitido cíclicamente por un “proveedor de sincronización”.

La mayoría de tareas de transmisión dentro de una red utiliza PDOs asincrónicos ya que éstos proveen una transmisión controlada por eventos basada en un evento específico al fabricante o al perfil del dispositivo (por ejemplo al exceder un valor de umbral definido).

La Tabla 3.1 enumera los tipos de transmisión PDO.

Asincrónica	Controlada por Eventos	Se transmite un PDO después de la ocurrencia de un evento específico al fabricante o al perfil del dispositivo. También puede ser llevada a cabo al finalizar un intervalo de tiempo
	Al recibir una Solicitud	Se transmite un PDO después de recibir una solicitud de un consumidor
Sincrónica	Acíclica	Se transmite un PDO después de haber recibido un objeto de sincronización
	Cíclica	Se transmite un PDO después de recibir un cierto número de objetos de sincronización
	Al recibir una Solicitud	Se transmite un PDO después de recibir la solicitud de un consumidor y un objeto de sincronización

Tabla 3.1: Tipos de transmisión PDO

Mapeo PDO

Ya que hasta 8 bytes de datos de aplicación (objetos de aplicación) pueden ser transmitidos en un PDO, se requiere definir la posición de estos dentro del campo de datos de un PDO. Esta especificación es designada como “Mapeo PDO”.

Cada objeto de aplicación está definido por un índice y subíndice de su entrada correspondiente en el diccionario de objetos, entonces el mapeo de estos objetos está dado por una lista¹ de índices y subíndices por medio de los cuales, se asigna el orden de estos objetos dentro del campo de datos del PDO.

De acuerdo al tipo de PDO, éste posee un mapeo específico, el cual puede ser cambiado y por tanto adaptado de acuerdo a los requerimientos de aplicación

¹ Esta lista contiene el orden en que debe mapearse los objetos de aplicación de acuerdo al tipo de PDO.

(“Mapeo Dinámico”), dispositivos más simples no requieren este mecanismo y utilizan mapeo fijo de objetos de aplicación (“Mapeo Estático”).

3.2.1.2 Objeto de Servicio de Datos (SDO)

La aplicación más importante de la transmisión de datos a través de mensajes SDO es la configuración de dispositivos.

Los Objetos de Servicio de Datos (SDOs) proveen transferencia confirmada de juegos de datos, de cualquier longitud, entre nodos. La transferencia de datos de un nodo a otro está definida de acuerdo al modelo cliente-servidor. Un cliente SDO (nodo iniciador) posee acceso directo a las entradas del diccionario de objetos de un servidor SDO por lo que puede cargar y descargar conjuntos de datos de cualquier longitud hacia y desde un servidor. El conjunto de datos a ser transferido es especificado por el índice y subíndice de la entrada del diccionario de objetos, el cual representa la ubicación de los datos. La conexión entre un cliente y un servidor requiere dos identificadores CAN ya que se necesita un identificador por dirección de transferencia (cliente y servidor).

Con el objetivo de que sea posible transferir un conjunto de datos de longitud arbitraria, a pesar de que la capacidad de transferencia de una trama CAN es de solamente de 8 bytes, los protocolos SDO soportan transmisión fragmentada de bloques de datos de más de 8 bytes [CIA-301]. Los datos a ser transmitidos son enviados como una secuencia de segmentos. Previo a la transferencia de los segmentos hay una secuencia de inicialización donde cliente y servidor se preparan para transferir los segmentos. Para bloques de datos de hasta 4 bytes es posible transferir los datos durante la secuencia de inicialización. Esto es llamado “transmisión SDO rápida”.

Una transmisión SDO es confirmada por lo que en ésta siempre participan dos mensajes (del cliente y servidor), de esta manera una transmisión puede ser cancelada por el cliente o servidor que participan en la misma, transmitiendo una trama de cancelación ante el surgimiento de una situación de error.

Un dispositivo CANopen puede ser cliente o productor SDO, sin embargo todos ellos deben proveer al menos un objeto servidor SDO y así ser accesible para otros nodos.

La Figura 3.5 muestra un ejemplo de la estructura de comunicación establecida a través de SDOs.

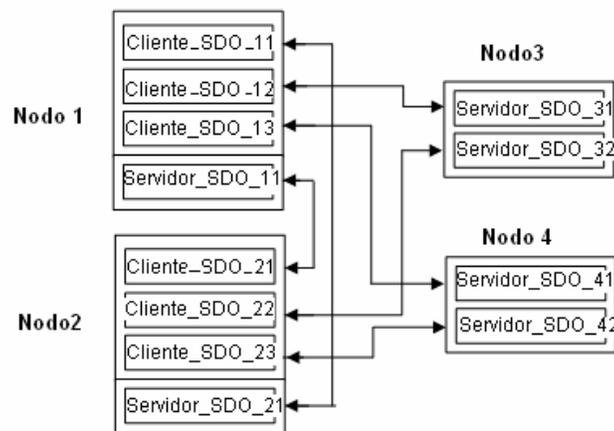


Figura 3.5: Ejemplo de una estructura de comunicación con SDOs

3.2.2 ADMINISTRACIÓN DE RED (NMT)

La función de administración de red especificada por CANopen ofrece dos grupos de servicios.

Control del Nodo.- Estos servicios son usados para controlar la inicialización de los nodos que desean tomar parte en una aplicación distribuida. Permiten control simultáneo del estado de comunicación de un solo nodo o de todos ellos.

Monitoreo del Nodo.- Estos servicios permiten supervisión continua del estado de comunicación de los nodos. Para este propósito está disponible el uso de “mensajes heartbeat”.

3.2.2.1 Control del Nodo

Sistemas de automatización distribuida basados en CANopen consisten de dispositivos operados remotamente cuyas capacidades de comunicación deben ser verificadas cada vez que el sistema es reiniciado. Además debe ser posible controlar el estado de cada nodo de tal manera que éste pueda ser configurado antes de establecer la comunicación o de resetear la aplicación y/o los parámetros de configuración del mismo.

El control del estado de comunicación de un nodo describe su estado con respecto a su capacidad de comunicación. Para este propósito un maestro NMT transmite un mensaje NMT el cual especifica un determinado comando que debe ser ejecutado por el nodo al cual va dirigido. Dependiendo del tipo de comando NMT, el nodo se cambiará a un determinado estado NMT. CANopen especifica los estados: Inicialización, Pre-operacional, Operacional y Detenido. El estado Inicialización y Pre-operacional fueron introducidos adicionalmente a parte del resto que fueron definidos por CAL.

Después de encender el dispositivo, éste entra al estado de Inicialización y automáticamente pasa al estado Pre-operacional. En este estado, el nodo está listo para intercambiar mensajes SDOs, de sincronización, de emergencia y etiquetas de tiempo (Ver Sección 3.2.3). En este estado a través de una herramienta¹ de configuración basada en SDOs es posible establecer parámetros del dispositivo, mapeo de objetos de aplicación a PDOs, asignación de identificadores a PDOs y SDOs, etc.

La transición al estado Operacional crea todos los PDOs de acuerdo a los parámetros descritos en el diccionario de objetos. A los nodos establecidos en este estado se les permite transmitir y recibir PDOs.

En el estado Detenido ninguna comunicación está permitida excepto el intercambio de objetos NMT.

¹ Esta herramienta puede ser provista por un dispositivo separado o estar integrada en una herramienta general que permita la configuración de toda la red así como la programación de nodos inteligentes de red.

3.2.2.2 Monitoreo de Nodo

Los mecanismos utilizados para monitorear la operabilidad de nodos CANopen son poleo cíclico por el maestro NMT (“Guardia de Nodo”) o a través de mensajes “Heartbeat” que cíclicamente el nodo envía en broadcast para informar su habilidad para comunicarse.

Guardia de Nodo

El nodo maestro NMT consulta a cada nodo (“esclavo NMT”) a intervalos regulares de tiempo con una trama especial de solicitud remota de transmisión dirigida a un nodo en particular. El esclavo transmite su estado de comunicación en un mensaje de respuesta. En caso de que el maestro no reciba respuesta dentro de un intervalo de tiempo definido o ésta no sea la esperada por él, éste indica un evento “guardia del nodo” a su aplicación. De igual manera el esclavo indicará un evento “guardia de vida” a su aplicación en caso de que su estado no haya sido consultado por el maestro dentro de un intervalo de tiempo definido.

Protocolo Heartbeat

Con éste método no se requieren consultas periódicas por parte del Master NMT. Un nodo señala su estado de comunicación transmitiendo cíclicamente un mensaje “heartbeat”. El protocolo Heartbeat es para propósitos de control de errores y señala la presencia de un nodo y su estado, además de que indica que el nodo que lo envía está trabajando apropiadamente. El mensaje Heartbeat puede ser recibido por uno o varios de los otros nodos (consumidores heartbeat) para monitorear a determinados nodos (“productores heartbeat”). Si un nodo responsable por monitorear los mensajes heartbeat de otro nodo en particular no los recibe, dentro de un intervalo definido de tiempo, éste señala un evento “heartbeat” a su aplicación.

Mensaje Boot-up

Este es un tipo especial de mensaje heartbeat, por medio del cual un dispositivo indica al master NMT que ha alcanzado estado pre-operacional después de que la

inicialización se ha completado. Sirve para identificar los nodos en estado pre-operacional.

3.2.3 SERVICIOS DE SISTEMA

CANopen también define los más importantes “servicios de sistema” aplicables a sistemas de automatización general, por ejemplo para: operación sincrónica sobre la red, provisión de un sistema global de tiempo y también mensajes estandarizados de error o emergencia.

3.2.3.1 Operación Sincrónica

El principio de sincronización está basado en la generación cíclica de “mensajes de sincronización” de alta prioridad enviados en broadcast por el productor de sincronización. El identificador usado para el mensaje de sincronización está especificado bajo una entrada específica del diccionario de objetos.

Al recibir mensajes de sincronización, las siguientes operaciones pueden ser provocadas en un nodo (consumidor) que soporta PDOs sincrónicos:

- Carga de nuevos datos de entrada (objetos de aplicación), que son mapeados en los respectivos PDOs sincrónicos para su transmisión.
- Transmisión de los datos recibidos durante la sincronización precedente.

3.2.3.2 Mensajes de Emergencia

Los mensajes de emergencia son mensajes de error de alta prioridad y son enviados para señalar estados de error en los dispositivos. Un mensaje de error es transmitido solamente una vez que ocurre o desaparece un error. Cada nodo CANopen dispone de un mensaje específico de emergencia para señalar un cambio en su estado de error. Estos mensajes pueden ser recibidos por varios o por ningún consumidor de emergencia. La señalización de error en los dispositivos es opcional.

3.2.3.3 Objeto de Etiqueta de Tiempo

CANopen define un objeto denominado “etiqueta de tiempo” de prioridad relativamente alta, usado por el productor correspondiente para proveer una referencia de tiempo común a los otros nodos. Los consumidores de mensajes de sincronización determinan la corrección requerida de su tiempo local a partir de la diferencia de tiempo existente entre la recepción de un mensaje de sincronización y la recepción del tiempo de referencia (etiqueta de tiempo) enviado por el productor de sincronización.

3.2.4 DICCIONARIO DE OBJETOS

El diccionario de objetos describe la funcionalidad del dispositivo y representa acceso completo a todos sus objetos (tipos de datos¹, parámetros de comunicación, opciones específicas del fabricante, datos y funciones disponibles para cada clase de dispositivo) por medio de un índice de 16 bits ($2^{16} = 65535$), además de un subíndice de 8 bits ($2^8 = 256$) por cada índice, es decir que es posible distinguir 65535 objetos y hasta 16777216 ($65535 \cdot 256$) entradas en el diccionario de objetos que son suficientes para definir la funcionalidad de un dispositivo.

3.2.5 PERFIL DE COMUNICACIÓN

El “perfil de comunicación” tanto como el “perfil de dispositivo” (sección 3.2.6) constituyen parte del diccionario de objetos. Un perfil de comunicación define toda la información general, parámetros y funciones de un dispositivo CANopen. Es decir, contiene información y funciones relacionadas con:

- Descripción del dispositivo (tipo y nombre del dispositivo, versión del software y del hardware)
- Estado del dispositivo
- Especificación de parámetros para sincronización y monitoreo del nodo
- Definición de parámetros de comunicación y mapeo de PDOs y SDOs.

¹ Estándar, complejo, o tipos de datos específicos al perfil o fabricante.

3.2.6 PERFILES DE DISPOSITIVO

Estos perfiles permiten que el sistema soporte intercambiabilidad de dispositivos de diferentes fabricantes, para lo cual describen las funcionalidades de un dispositivo a través de funciones estandarizadas dentro de dos secciones del diccionario de objetos denominadas “perfil de dispositivo estandarizado” y “perfil de dispositivo específico al fabricante”. Algunos ejemplos de perfiles de dispositivo, son los siguientes:

- Perfiles para dispositivos I/O
- Perfiles para unidades de control de movimiento
- Perfiles para dispositivos programables IEC 61131-3

3.2.7 ASIGNACIÓN DE IDENTIFICADORES

Dependiendo de los requerimientos de una aplicación específica, CANopen permite varios métodos para la asignación de identificadores:

- Asignación Predefinida
- Asignación Por Configuración
- Asignación Dinámica

Estos esquemas permiten la implementación de hasta 128 conexiones peer-to-peer (conexiones SDO), o de otra manera 1 dispositivo maestro con hasta 127 nodos esclavos (conexiones PDO). Estos dispositivos son distinguidos por los 7 bits menos significantes de su identificador de 11 bits, el resto (4 bits) son utilizados como “código de función” y distinguen diferentes funciones de comunicación: emergencia, NMT, sincronización, etiquetas de tiempo y monitoreo. Ver Figura 3.6.

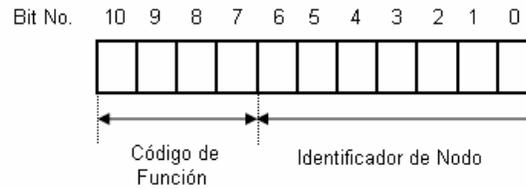


Figura 3.6: Identificador para mensajes CANopen

3.2.8 CAMPO DE APLICACIÓN

CANopen es la capa aplicación estandarizada para redes embebidas¹. Algunos usuarios también han implementado CANopen en automatización de sistemas de control.

Especificaciones CANopen cubren la capa aplicación estandarizada, estructuras para aplicaciones específicas así como perfiles de dispositivos, interfaces, y de aplicaciones.

Los principales campos de aplicación de CANopen incluyen:

CANopen en camiones y buses

Las típicas aplicaciones incluyen camiones grúas y superestructuras para camiones de bomberos.

CANopen en vehículos agrícolas y de construcción

Tales como vehículos de agricultura, máquinas de construcción de carreteras y máquinas de pavimentado. Otras aplicaciones incluyen vehículos de taladrado en túneles así como cosechadoras de papas. La asociación de camiones industriales recomienda CANopen para uso en camiones con elevadores eléctricos.

CANopen en trenes de carga y pasajeros

Varios fabricantes europeos de tranvías, subterráneos y ferrocarriles ligeros han implementado CANopen en estos sistemas de transporte. Las redes CANopen

¹ El término embebido hace referencia al hecho de que un microcomputador está encerrado o instalado dentro de un sistema mayor y su existencia como microcomputador puede no ser aparente.

enlazan sistemas de puertas, iluminación y control del motor. Se está desarrollando un perfil de aplicación estandarizado CANopen para locomotoras a diesel y autos de potencia. En Rusia CANopen es usado como una red embebida en sistemas de conducción de trenes.

CANopen en electrónica marítima

La estructura de CANopen para electrónica marítima es usada para la administración de potencia, control del motor y generador, bombas y válvulas de carga. Para satisfacer condiciones de seguridad, se emplean arquitecturas de red redundantes con capacidad de conmutación automática. CANopen provee un protocolo que facilita la interoperabilidad segura de equipos y sistemas marítimos.

CANopen en automatización de fábricas

Varias compañías europeas han diseñado sistemas de automatización de fábricas basadas en CAN, por ejemplo en líneas de ensamblaje.

CANopen en control industrial de máquinas

Uno de los principales campos de aplicación de CANopen está en el control embebido de máquinas industriales. CANopen es empleado principalmente en medios de impresión, textiles, costura industrial, moldeo a inyección, en robots y en otras numerosas aplicaciones relacionadas. Redes CANopen conectan conjuntos modulares para sensores y controladores de movimiento.

CANopen en elevadores y escaleras mecánicas

Para esto ha sido desarrollado un perfil de aplicación CANopen (CANopen Lift). Elevadores y escaleras mecánicas pueden ser encontrados en edificios de oficinas, casas, tiendas, fábricas, etc.

CANopen en automatización de edificios

Redes CANopen son instaladas en sistemas y subsistemas de automatización de edificios. Existen sistemas de aspersores, sistemas integrados de control de habitaciones y sistemas de control de supermercado conectados con CANopen.

En muchas aplicaciones, sistemas basados en CANopen son alimentados por otra red.

CANopen en equipo y dispositivos médicos

No solamente puede ser empleado en redes embebidas de dispositivos médicos, sino también en cuartos completos enlazando a través de redes CANopen todos los dispositivos tales como mesas de operación, monitores, entre otros.

CANopen en equipo no industrial

CANopen también es implementado en equipos no industriales tales como en laboratorios para experimentos físicos de alta energía. CANopen interviene también en redes embebidas de telescopios. Otras aplicaciones típicas incluyen puertas automáticas y sistemas de refrigeración, etc.

3.3 DEVICENET

DeviceNet fué desarrollado por Rockwell Automation y presentado como un estándar de arquitectura abierta¹ basado en el protocolo CAN. Actualmente ocupa un rol muy importante en EE.UU. y Asia como protocolo eficiente aplicado en la tecnología de automatización. Cada vez más soluciones de sistemas están siendo implementadas con DeviceNet en Europa.

DeviceNet es una de las tres redes abiertas (DeviceNet, ControlNet y Ethernet/IP) que comparten un protocolo en común conocido como el “Protocolo de Control e información” (CIP) (Ver Figura 3.7). La parte de control de CIP es usada para mensajería implícita². Mientras que la parte de información se usa para mensajería explícita³. Los dos tipos de mensajes proveen beneficios óptimos para control industrial.

¹ Permite que la red se comunique y funcione conjuntamente con productos de diferentes vendedores, además de una fácil adición de dispositivos a una instalación ya existente o planeada.

² Control de los datos en tiempo real de entrada-salida (I/O).

³ Soporte de datos de comunicación para información de programación y configuración.

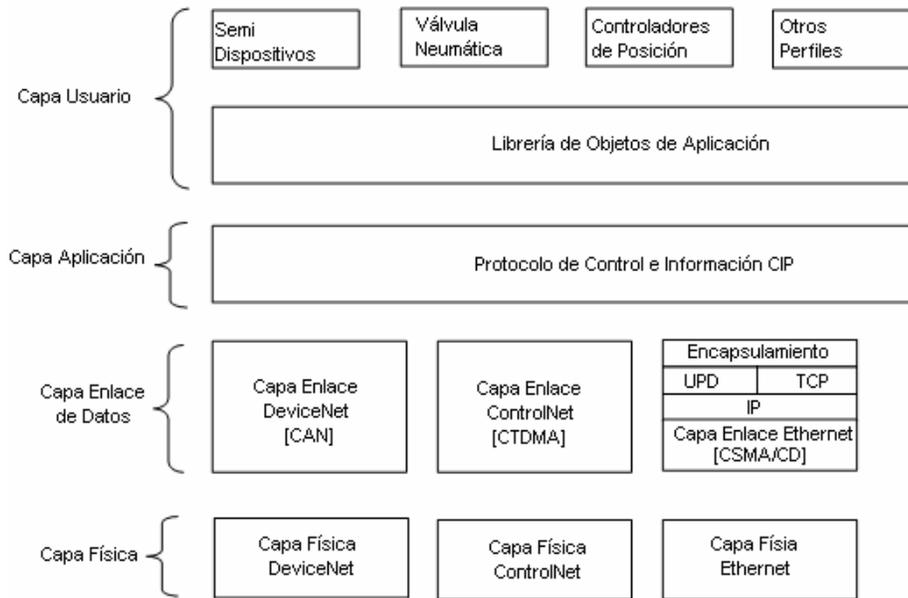


Figura 3.7: Protocolo de Control e Información (CIP)

Además de proveer una definición para la capa 7 (Aplicación) del modelo OSI, la especificación DeviceNet también define partes de capa 1 (transceivers y medios de transmisión). La Tabla 3.2 muestra la clasificación de DeviceNet de acuerdo al Modelo de capas ISO.

Capa ISO 7	Capa Aplicación	Especificación DeviceNet
Capa ISO 2	Capa Enlace de datos	Especificación CAN
Capa ISO 1	Señalización Física	
Capa ISO 1	Transceiver	Especificación DeviceNet
Capa ISO 1	Medio de Transmisión	

Tabla 3.2: DeviceNet en el modelo de capas ISO

El extenso rango de aplicaciones ha dado como resultado la especificación de varias opciones de cables: cable grueso, delgado y plano. Todos tienen cuatro alambres y una tierra (excepto el plano, que no tiene tierra) y han sido diseñados para el ambiente industrial. Una línea completa de conectores está disponible globalmente por múltiples proveedores.

Hasta 64 nodos pueden ser operados en una red DeviceNet. Las líneas del bus son terminadas en ambos extremos con impedancias de 121 Ohms. La energía necesaria para el funcionamiento de los dispositivos puede ser provista a través del mismo bus DeviceNet o a través de fuentes independientes. La Tabla 3.3 muestra la máxima extensión del bus de acuerdo a la velocidad de transmisión y al tipo de cable utilizado.

Velocidad de Transmisión	Longitud, de acuerdo al tipo de cable		
	Cable Grueso	Cable Delgado	Cable Plano
125 KBit/s	500 m.	100 m.	420 m.
250 KBit/s	250 m.	100 m.	200 m.
500 KBit/s	100 m.	100 m.	100 m.

Tabla 3.3: Máxima extensión de la red de acuerdo al tipo de cable

Como otros protocolos, la función principal del protocolo DeviceNet es intercambiar datos entre dispositivos y con las unidades de control asociadas. En DeviceNet, se hace una distinción entre mensajes de procesos de alta prioridad (Mensajes I/O) y mensajes de administración de baja prioridad (Mensajes Explícitos).

La comunicación entre dos o más dispositivos siempre toma lugar de acuerdo con el modelo de comunicación orientado a conexión, ya sea a través de conexión punto a punto o multicast.

3.3.1 MODELO DE OBJETOS

DeviceNet describe todos los datos y funciones de un dispositivo usando un modelo de objetos (CANopen utiliza un Diccionario de Objetos). Un dispositivo puede ser definido como una colección de objetos individuales.

Un objeto representa una descripción abstracta de un componente dentro de un dispositivo. Un objeto es determinado por sus datos o atributos, sus funciones o servicios y por su comportamiento.

Los atributos, representan los datos que un dispositivo hace disponibles a través de DeviceNet (estados de un objeto, número de serie del dispositivo y datos de procesos tales como temperatura, presión, posición, etc.).

El comportamiento de un objeto define cómo éste responde a eventos externos o internos.

La Figura 3.8 muestra los objetos más importantes de un dispositivo DeviceNet.

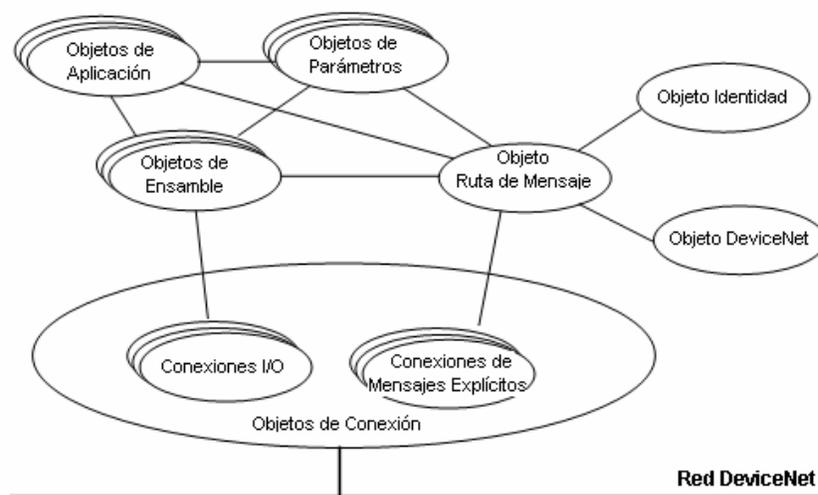


Figura 3.8: Modelo de Objetos DeviceNet

DeviceNet tiene tres tipos generales de objetos: de comunicación, de sistema y de aplicación específica

- Objetos de Comunicación definen y administran los mensajes intercambiados a través de DeviceNet.
- Objetos de Sistema definen datos y funciones generales que son útiles a todos los dispositivos.
- Objetos de Aplicación Específica definen funciones y datos específicos al dispositivo.

3.3.1.1 Objetos de Comunicación

Objeto DeviceNet

Este objeto define la conexión física de un dispositivo a la red DeviceNet. Este objeto también administra la Conexión Maestro-Esclavo, mencionada más adelante.

Objeto de Ruta de Mensaje

Permite el acceso a una funcionalidad en particular de un dispositivo a través de Mensajes Explícitos.

Objeto de Conexión

Entre sus principales características, define:

- longitud de los datos a ser producidos o consumidos
- identificador CAN usado para la conexión
- comportamiento ante errores

Objeto ACK (Confirmación)

Se encarga de la recepción de mensajes de confirmación. Además de notificar al objeto de aplicación si una confirmación de mensaje fue o no recibida.

3.3.1.2 Objetos de Sistema

Objeto de Identidad

Contiene atributos que identifican claramente a un nodo dentro de una red (número de serie, tipo de dispositivo y estado actual del dispositivo).

Objeto de Parámetro

Especifica una interfaz opcional disponible para todos los datos y parámetros de configuración. Provee un amplio número de atributos que permiten al usuario configurar convenientemente el dispositivo.

3.3.1.3 Objetos de Aplicación Específica

Objeto de Aplicación

Pueden estar predefinidos en la librería de objetos DeviceNet o estar definidos por el fabricante del dispositivo. Para aplicaciones típicas de automatización se tienen objetos de aplicación simples, tales como unidades de control analógico y discreto.

Objeto de Ensamble

Con este objeto, el usuario tiene la opción de mapear datos de diferentes objetos de aplicación en uno solo de estos objetos. Particularmente importante en Mensajes I/O para maximizar la eficiencia del intercambio de datos de control en la red.

3.3.2 IDENTIFICADOR CAN

DeviceNet está basado en el estándar del protocolo CAN y por tanto, al igual que CANopen, emplea un identificador de 11 bits. Por lo que pueden hacerse distinciones entre $2^{11} = 2048$ mensajes. Sin embargo una red DeviceNet está limitada a 64 participantes, siendo suficiente 6 bits para identificar un dispositivo, los cuales son conocidos como MAC-ID; los 5 bits restantes se utilizan para identificar el mensaje.

En DeviceNet, el identificador CAN es conocido como Identificador de Conexión (ID Conexión), el cual comprende el Identificador de Grupo de Mensaje, el Identificador de Mensaje dentro de su grupo y el Identificador MAC del dispositivo, que puede ser la dirección origen o destino, su definición depende del Grupo de Mensaje y del Identificador de Mensaje (Ver Figura 3.9).

ID Conexión=Identificador CAN										Usado para
10	9	8	7	6	5	4	3	2	1	
0	ID Mensaje				ID MAC Origen					Grupo 1 de Mensajes
1	0	ID MAC					ID Mensaje			Grupo 2 de Mensajes
1	1	ID Mensaje			ID MAC Origen					Grupo 3 de Mensajes
1	1	1	1	1	ID Mensaje					Grupo 4 de Mensajes

Figura 3.9: Definición de Grupos de Mensajes

Existen 4 grupos de mensajes usados, cuya diferencia entre estos es la prioridad que poseen. Estos son:

Grupo 1 de Mensajes

Es utilizado para intercambiar mensajes de alta prioridad por medio de mensajes I/O.

Grupo 2 de Mensajes

De prioridad menor al anterior grupo, principalmente está dirigido a la Conexión Maestro-Esclavo definida más adelante.

Grupo 3 de Mensajes

Para intercambio de mensajes de baja prioridad.

Grupo 4 de Mensajes

No incluye ninguna dirección de dispositivo, sino solamente el Identificador del Mensaje. Los mensajes de este grupo son solamente usados para administración de red.

3.3.3 MODELO DE COMUNICACIÓN

DeviceNet usa un modelo de comunicación orientado a conexión. Por ello no se pueden acceder a los datos en la red hasta que se establezca una conexión entre objetos.

3.3.3.1 Establecimiento de la Conexión

Previo al establecimiento de la conexión en DeviceNet, se deben establecer objetos de comunicación, los cuales no están inicialmente disponibles una vez que se enciende el dispositivo, sino que primero deben ser creados. El establecimiento de la conexión entre dispositivos DeviceNet está basado en puertos. Un puerto actúa como una “puerta de entrada” al dispositivo y la “llave” que abre esa puerta es el Identificador CAN del puerto seleccionado (ID de Conexión). Una vez encendido un dispositivo DeviceNet, el único puerto por el que éste puede ser direccionado es el Puerto UCMM (Administración de Mensaje sin Conexión) y el Puerto de conexión Maestro-Esclavo. Otras “puertas” del dispositivo solamente podrán ser abiertas una vez que se disponga de la llave apropiada y que otros Objetos de Conexión sean establecidos. De igual manera una conexión I/O o una conexión explícita pueden ser finalizadas a través del puerto UCMM. Cada dispositivo puede establecer una conexión con cada uno de los otros dispositivos de la red, esto depende de cuántas conexiones soporte el mismo. Este representa el procedimiento general empleado en los dispositivos DeviceNet, el cual toma ventaja de las características multicast, peer to peer y productor-consumidor.

Por otro lado adicionalmente se ha definido una conexión maestro-esclavo, para establecer enlaces de forma simple y rápida que requiere menos recursos para administrar las conexiones, ésta técnica representa un subconjunto del método general de establecimiento de la conexión y se limita a relaciones estrictamente maestro-esclavo. Múltiples maestros son permitidos en la red.

3.3.3.2 Mensajes Explícitos

A través de este sistema se pueden transferir datos de configuración de baja prioridad, datos generales de administración o incluso datos de diagnóstico. Este tipo de comunicación utiliza un sistema cliente-servidor (punto a punto), por ello una solicitud del cliente siempre debe ser confirmada con una respuesta del servidor. La Figura 3.10 muestra la relación cliente-servidor con mensajes explícitos.

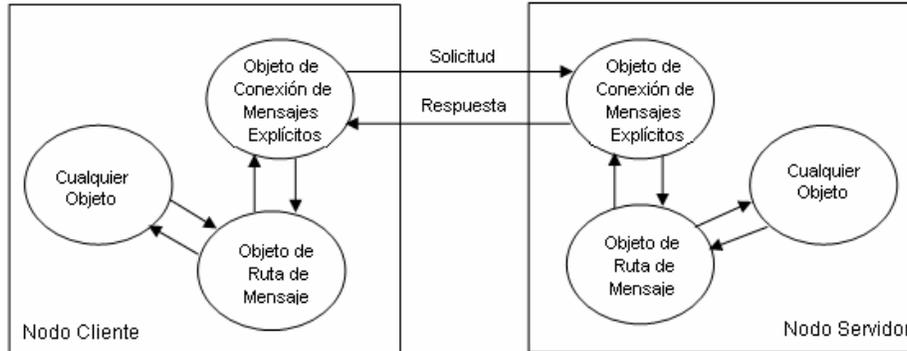


Figura 3.10: Conexión de Mensajes Explícitos

Para transmitir mensajes explícitos previamente deben haber sido establecidos sus respectivos objetos de conexión en el dispositivo. Esto puede ser hecho usando la conexión predefinida Maestro-Esclavo para activar un objeto de conexión estática en el dispositivo o usando el Puerto UCMM (De Administración de Mensaje Sin Conexión) de un dispositivo a través del cual el objeto de conexión de mensajes explícitos puede ser dinámicamente establecido. El único mensaje que puede ser enviado por el puerto UCMM es un pedido de establecimiento de una Conexión explícita de mensajes. Los mensajes explícitos siempre pasan a través de Objetos de Ruta de Mensaje (Ver Figura 3.8).

La característica más importante del formato de un mensaje explícito es que ninguna parte del Identificador CAN es usada para especificar este mensaje. Sino que todo éste se encuentra contenido en el campo de datos CAN. Dos identificadores son requeridos por cada conexión, uno para el mensaje de solicitud y otro para el mensaje de respuesta.

Cuando una conexión es establecida, si la longitud del mensaje es mayor a 8 bytes automáticamente se usará el protocolo de fragmentación.

3.3.3.3 Mensajes I/O

Estos mensajes son usados para intercambiar datos de procesos y aplicaciones de alta prioridad a través de la red DeviceNet. La comunicación a través de estos

mensajes está basada en el modelo productor-consumidor, de tal manera que los datos I/O son siempre transferidos de una aplicación productora a una o más aplicaciones consumidoras. Esto es llevado a cabo usando mensajes I/O a través de Objetos de conexión de estos mensajes (La Figura 3.11 muestra dos aplicaciones consumidoras) previamente establecidos en el dispositivo, y que puede ser hecho de dos maneras:

- Por medio de la conexión Maestro-Esclavo para activar un Objeto de conexión I/O estático ya disponible en el dispositivo.
- Usando un objeto de conexión de mensajes explícitos disponible en el dispositivo para establecimiento dinámico de un Objeto de conexión I/O apropiado.

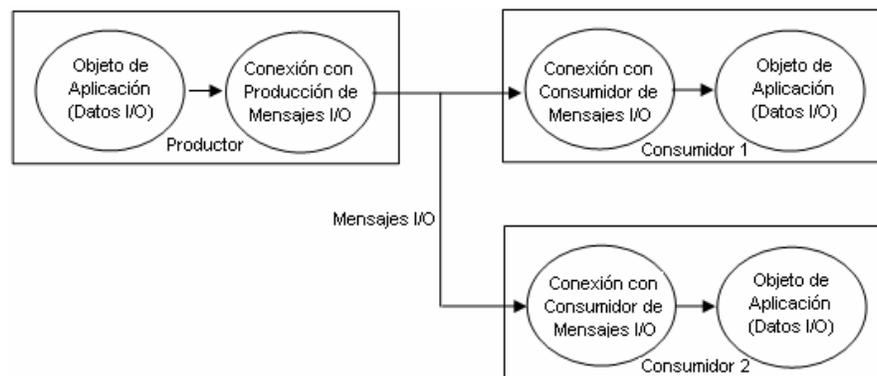


Figura 3.11: Conexión de Mensajes I/O

Mensajes I/O usualmente pasan directamente a los datos del objeto de aplicación designado. El Objeto de Ensamble es el objeto de aplicación mayormente usado en conexiones I/O (Ver Figura 3.8).

La característica más importante del formato de un mensaje I/O es que usa todo el campo de datos de la trama CAN para la transferencia de datos de proceso. En caso de que los 8 bytes disponibles del campo de datos de la trama CAN no sean suficientes, se hará uso del protocolo de fragmentación.

3.3.3.4 Protocolo de Fragmentación

Permite que el protocolo DeviceNet soporte la transferencia de mensajes de cualquier longitud. La transferencia de fragmentos de un mensaje explícito (modelo cliente-servidor) siempre es confirmada, no así en la transmisión de fragmentos de mensajes I/O (modelo productor-consumidor).

3.3.4 ADMINISTRACIÓN DE RED

Describe todos los servicios que no pueden ser asignados a un objeto específico.

3.3.4.1 Chequeo de MAC ID

En muchos casos, la MAC ID puede ser establecida externamente en el dispositivo utilizando switches. Todos los nodos en una red deben tener direcciones diferentes para asegurar perfecta comunicación en un sistema DeviceNet. Este mecanismo revisa que las MAC ID de los dispositivos de una misma red no se repitan.

3.3.4.2 Mensaje Heartbeat

Al igual que en el protocolo CANopen un dispositivo puede usar este mensaje para anunciar su estado y si está o no trabajando apropiadamente en la red.

3.3.4.3 Mensaje Shutdown

Usando este mensaje un dispositivo que está desactivado como resultado de errores internos, puede anunciar la razón para su desactivación antes de cesar completamente sus funciones.

3.3.5 PERFILES DE DISPOSITIVOS

Las dos metas básicas del diseño de productos DeviceNet son la interoperabilidad e intercambiabilidad. Estas metas son alcanzadas por medio del uso de un “perfil de dispositivo” específico en el producto. Muchos perfiles de propósito general ya han sido especificados por ODVA. Sin embargo, el diseñador de un producto puede extender el perfil existente de la especificación

DeviceNet a sus propias necesidades, además de que puede definir su propio perfil si el existente no satisface sus requerimientos.

La especificación DeviceNet define tres características para clasificar el perfil de un dispositivo.

Comportamiento

Todos los dispositivos de un perfil específico deben mostrar el mismo comportamiento, por tanto deben soportar el mismo Modelo de Objetos que incluye todos los objetos y servicios implementados.

Datos I/O

Para ínter cambiabilidad es esencial que todos los dispositivos de un perfil específico produzcan o consuman el mismo formato de datos I/O, especificado en el perfil del dispositivo.

Configuración

Para un perfil específico, todos los dispositivos deben contener el mismo conjunto básico de parámetros de configuración.

3.3.6 CAMPO DE APLICACIÓN

DeviceNet es la capa Aplicación estandarizada para automatización de fábricas. Algunos usuarios han implementado DeviceNet también en sistemas de control de máquinas.

Los principales campos de aplicación de DeviceNet incluyen:

DeviceNet en automatización de fábricas

DeviceNet es la red principal para automatización de fábricas en América del Norte. Se usa para conectar las máquinas, subsistemas tales como bandas transportadoras y otros sistemas de automatización de fábrica. En la industria

automotriz americana, DeviceNet es muy bien aceptado como una red sensor/actuador.

DeviceNet en el control industrial de máquinas

Existe una gran cantidad de redes de DeviceNet instaladas en sistemas de control industrial de máquinas. La organización "Semiconductor Equipment and Materials International (SEMI)" ha seleccionado DeviceNet como una de las opciones de sistemas sensor/actuador.

DeviceNet en la electrónica marítima

DeviceNet se usa, por ejemplo, para controlar grúas y otros equipos montados en naves o buques.

DeviceNet en el control no industrial

En algunos parques de diversión, sistemas basados en DeviceNet controlan tiovivos y otros equipos. Otro ejemplo de la aplicación para DeviceNet en control de máquinas no industriales se pone de manifiesto en el "Ojo de Londres", la rueda (o noria) de Ferris más alta del mundo.

3.4 SAE J1939

Es un protocolo estandarizado para comunicación serial entre unidades de control electrónicas principalmente usado en el sector de vehículos comerciales, a través del cual es posible transmitir valores medidos y datos de control así como datos de configuración. También es posible leer o desechar datos de diagnóstico de componentes individuales y llevar a cabo calibraciones de sistemas individuales de control.

Para hacer esto posible, debe especificarse el tipo de transmisión (cómo se transmite) y también el significado de los mensajes (qué es transmitido).

Debido a la compatibilidad requerida con otros protocolos existentes (J1708/J1587), fue necesario emplear un identificador de 29 bits. Con la disponibilidad del identificador CAN extendido fue posible aplicar el principio de comunicación orientado al nodo.

SAE J1939 es dividido en varios documentos [Stepper97] de acuerdo al modelo de capas OSI, por ello los números de los documentos se refieren en cada caso a la capa asignada en el modelo de capas. De forma análoga, a prácticamente todos los protocolos de bus de campo, en SAE J1939 las capas 5 y 6 no son necesarias y por tanto no son especificadas. La Figura 3.12 muestra la asignación de los documentos SAE J1939 referidos al modelo OSI.



Figura 3.12: SAE J1939 en el modelo de capas ISO/OSI

La capa física describe, entre otras cosas, la interfaz eléctrica con el medio físico. Además realiza una diferenciación entre dos variantes:

- La especificación SAE J1939/11 define una conexión con transmisión diferencial de señal (de acuerdo a ISO 11898) a través de 1 par trenzado blindado como líneas del bus terminadas con resistencias de 120 Ohms en ambos extremos. La velocidad de transmisión es de 250 Kbits/s, el máximo número de nodos es 30 y la longitud máxima del bus es de 40 metros.

- La especificación SAE J1939/12 describe una variante del medio utilizado en el bus consistente de 2 pares de alambres trenzados blindados, siendo posible transmitir un voltaje de 12 V por uno de ellos.

La especificación SAE J1939/21 describe la capa enlace de datos CAN, señalando el uso del “formato extendido” de trama. El “formato básico” solamente es usado para aplicaciones exclusivas del fabricante. Esta especificación esencialmente describe varios servicios de red para solicitud de mensajes, transmisión confirmada y transmisión fragmentada de un bloque de datos.

La especificación SAE J1939/31, en esencia, describe la funcionalidad de un puente¹ para la transmisión de mensajes entre dos segmentos² de red.

La capa aplicación con los documentos SAE J1939/71 y /73 describe los datos en sí. Cada mensaje es distinguido por un número (Número de Parámetro de Grupo).

La función de administración de red es descrita en el documento SAE J1939/81, y puede considerarse como una unidad separada con acceso directo a la capa 1, razón por la que la Figura 3.12 la muestra como un bloque independiente.

3.4.1 ESTRUCTURA DEL MENSAJE

Un mensaje consiste básicamente de un identificador de 29 bits, y hasta 8 bytes de información en el campo de datos. Utilizando el formato extendido es posible proveer una especificación completa de los mensajes a través del identificador CAN.

El identificador CAN es dividido en las siguientes partes (Ver Figura 3.13):

- Prioridad

¹ Un puente es entendido como la conexión de dos segmentos CAN.

² Por ejemplo entre un camión y su trailer.

- Número de Parámetro de Grupo (PGN)
- Dirección origen del mensaje

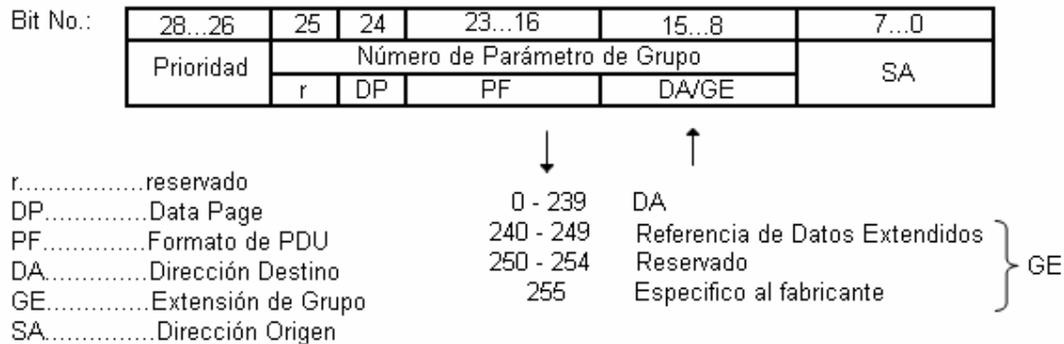


Figura 3.13: Estructura del identificador CAN con SAE J1939

3.4.1.1 Prioridad

Sus bits (3), son los más significantes del identificador y son los que determinan la prioridad del mensaje con respecto al acceso al medio. Estos bits no tienen importancia para el receptor de un mensaje, solamente determinan la prioridad de transmisión de éste.

3.4.1.2 Número de Grupo de Parámetro (PGN)

Consta de los siguientes campos:

- Un bit reservado (r)
- Un bit “Data Page” (DP)
- Formato de PDU¹ (8 bits) (PF)
- Un campo de 8 bits (DA/GE), cuyo uso depende del tipo de formato PDU empleado

Por medio de los campos PF y DP, se describen los datos contenidos dentro de un mensaje CAN. PF especifica 256 valores que representan diferentes

¹ Unidad de Protocolo de Datos (referido a los datos dentro del Campo de Datos del Mensaje CAN).

contenidos del campo de datos y el bit DP posee 2 valores posibles de los cuales solo uno de ellos es usado, el otro es reservado para extensiones futuras.

A través del campo PF, el PGN especifica el tipo de transmisión sirviendo al mismo tiempo como identificador del mensaje en cuestión. Este campo sirve también para diferenciar entre dos tipos de servicios de comunicación:

- Formato 1 PDU: Transmisión orientada al nodo
- Formato 2 PDU: Transmisión orientada al mensaje

El Campo (DA/GE) es usado como Dirección de Destino (DA) o como Extensión de Grupo (GE), dependiendo del Formato de PDU (PF) empleado: Con transmisión orientada al nodo, especificará la “dirección del nodo destinatario” del mensaje (DA), lo cual permite intercambiar datos entre dos nodos en forma de comunicación peer-to-peer. Es posible también, utilizando una dirección de destino específica, direccionar a todos los nodos de una red (mensaje broadcast).

Con transmisión orientada al mensaje, este campo será usado como “extensión de grupo” (GE). No se especifica una dirección destino en particular, y cada nodo receptor reconoce si el mensaje recibido es de su interés examinando el valor de su PF. Siendo posible de esta manera la transmisión multicasting.

Finalmente, un bit (r) es reservado por SAE para usos posteriores, el mismo es establecido en cero.

La Tabla 3.4 indica el rango de valores para los formatos 1 y 2 de PDU.

Formato PDU	Rango de Valores	Significado del campo (DA/GE)
1	0...239	Dirección de destino (DA)
2	240...255	Extensión de grupo (GE)

Tabla 3.4: Rango de valores de mensajes en formato 1 y 2 de PDU

3.4.1.3 Campo Dirección Origen

El campo Dirección de Origen, de 8 bits, especifica al transmisor de los datos. Cada nodo tiene una dirección única de dispositivo (0...253). En realidad son 256 posibles direcciones ($2^8 = 256$), enumeradas de 0 a 255, pero se reservan las dos últimas (254 y 255). Ver tabla 3.5.

3.4.1.4 Campo de Datos

Los bytes de datos asignados a un PGN son transmitidos a través de un campo de datos máximo de 8 bytes de un mensaje CAN. Si los datos de un PGN requieren más de 8 bytes de datos, el bloque de datos es fragmentado, y transmitido en forma de varios mensajes CAN.

3.4.2 TIPOS DE MENSAJES

Por medio del PGN, se distinguen 5 tipos de mensajes. Estos son:

- Comandos
- Solicitudes
- Mensajes de Respuesta o Broadcast
- Mensaje de Confirmación (ACK) o No Confirmación (NACK)
- Funciones de Grupo

3.4.2.1 Comandos

Sirven para llevar a cabo una función en uno o más nodos de destino. Pueden utilizar ambos tipos de formato PDU. Con el Formato 1 de PDU, es posible direccionar a un nodo en particular o a todos ellos (broadcast).

3.4.2.2 Solicitudes

Hace uso de transmisión orientada al nodo, pudiendo direccionar un nodo en particular o a todos ellos a través del campo DA. Para este propósito no se usan tramas RTR sino tramas de datos cuyo campo PF posee el valor 234. Los datos

solicitados son especificados a través de su PGN en el campo de datos de la trama de solicitud.

3.4.2.3 Mensaje de Respuesta

Este tipo de mensajes también utiliza transmisión orientada al nodo y pueden ser enviados a un nodo en particular, o a todos ellos a través de un mensaje de broadcast¹. La respuesta a un mensaje de solicitud puede consistir de más de una sola trama de datos, en cuyo caso se hace uso del protocolo de transmisión fragmentada (Ver Sección 3.4.3.2).

3.4.2.4 Mensajes de Confirmación

Son enviados en respuesta a un comando o a una solicitud de mensaje. Hacen uso de transmisión orientada al nodo en broadcast de la siguiente manera: en su campo PF lleva el valor 232 (correspondiente a transmisión orientada al nodo) y en su campo DA direcciona a todos los nodos de la red. Pueden ser “ACK” ó “NACK”. Un “ACK” (confirmación) es una respuesta positiva y un “NACK” (confirmación negativa) es una respuesta negativa.

3.4.3 FUNCIONES DE GRUPO

Este tipo de mensaje es usado para identificar funciones especiales tales como la transmisión de largos bloques de datos, mensajes propietarios y funciones de administración de red. PGNs específicos son reservados para ser asignados a éstas funciones de grupo.

3.4.3.1 Mensajes Proprietarios

Esta funcionalidad brinda a los fabricantes completa libertad del campo de datos, pudiendo así definir sus propios servicios con mensajes de solicitud y confirmación. Dentro de la función de grupo para mensajes propietarios, se hace una distinción entre dos tipos de comunicación:

¹ Cabe señalar también que de acuerdo al sistema implementado un mensaje broadcast puede ser enviado no solamente en respuesta a un mensaje de solicitud.

- Propietario A: transmisión orientada al nodo a través del Formato 1 PDU (Con 239 en el campo PF).
- Propietario B: transmisión orientada al mensaje a través del Formato 2 PDU (Con 255 en el campo PF).

Con transmisión orientada al mensaje no se requiere indicar la dirección de destino, por lo que los 8 bits del campo GE están disponibles para uso del fabricante, con ello éste cuenta con 256 diferentes mensajes por nodo para transmisión de mensajes propietarios.

3.4.3.2 Transmisión Fragmentada

Bloques de datos con más de 8 bytes de datos deben ser transmitidos en fragmentos.

Ya que con transmisión fragmentada el primer byte de datos de cada mensaje es usado para enumerar los fragmentos o segmentos, solamente pueden ser transmitidos 7 bytes de datos por mensaje.

SAE J1939 soporta dos tipos de transmisión fragmentada:

- Transmisión fragmentada orientada al nodo
- Transmisión fragmentada en broadcast

En ambos casos hacen uso del Formato 1 de PDU.

Protocolo de transmisión fragmentada orientada al nodo

Permite un intercambio de datos entre dos nodos, para lo cual es necesario establecer una conexión virtual llevando a cabo 3 pasos:

- Establecimiento de la conexión.- Etapa en la que se establece el número de bytes de datos a ser transmitidos, el número de segmentos requeridos

y el PGN que especifica la importancia del bloque de datos transmitido, a través del intercambio de mensajes especiales¹.

- Transmisión de datos
- Desconexión.- Después de recibir el último mensaje, el nodo receptor da por finalizada la conexión utilizando un mensaje EOM ACK (Confirmación de Mensaje Final).

Protocolo de Transmisión Fragmentada en Broadcast

Con este tipo de transmisión no se establece ninguna conexión previa entre el transmisor y el receptor por ello el inicio de una transmisión es anunciado solamente por un mensaje denominado “Anuncio de Mensaje Broadcast” (BAM). Este mensaje contiene el número de bytes, número de segmentos y el PGN del bloque de datos a ser enviado. No existe respuesta al mensaje BAM. Como el transmisor no conoce los receptores no puede verificar si todos están listos a recibir el bloque de datos. Por esta razón, la transmisión del primer segmento de datos empieza solamente después de 50 a 200 milisegundos después del mensaje BAM, tiempo suficiente para que los potenciales receptores se preparen para la recepción del bloque de datos anunciado.

3.4.4 ADMINISTRACIÓN DE RED

A diferencia de otros protocolos de capas superiores, la administración de red aplicada en SAE J1939 [J1939/81] sirve para la asignación automática o determinación de direcciones de los nodos, la administración de red no lleva a cabo el monitoreo del nodo y debe ser realizado a través de la aplicación de mensajes cíclicos de evaluación.

La dirección² del nodo representa una precondition básica para el protocolo SAE J1939. Por esta razón, todos ellos revisan su dirección pre-configurada o piden una dirección libre durante su inicialización. Esto ocurre a través del proceso denominado “Solicitud de Dirección”. Así, entonces SAE J1939 no tiene una unidad central que asigne números a los nodos, sino que descentralizadamente controlan esta asignación entre ellos mismos.

¹ Mensajes RTS (Request to Send) y CTS (Clear to Send).

² Dirección origen, y en el caso de transmisión orientada al nodo también la dirección destino.

Sin embargo, como se ha visto, existen dispositivos que tienen direcciones pre-asignadas que normalmente deberían ser usadas por los dispositivos mientras no exista conflicto con otros nodos.

La Tabla 3.5 muestra las clases de direcciones disponibles para los nodos.

Dirección	Uso
0-127	Direcciones asignadas fijas para nodos de red que siempre o casi siempre están presentes
128-247	Direcciones reservadas para asignación específica o determinación dinámica
248-253	Direcciones asignadas fijas para nodos de red que siempre o casi siempre están presentes
254	Dirección Cero. Es usada como dirección origen de un dispositivo cuando no se le ha podido asignar ninguna otra dirección
255	Dirección para mensajes de broadcast

Tabla 3.5: Clases de direcciones de nodos

Para asignar o pedir direcciones de nodo a través de la red, todos los nodos deben ser unívocamente identificables. Esto es conseguido con un campo de 64 bits que representa el nombre¹ del dispositivo y que es transmitido junto con el identificador de 29 bits (en el campo de datos de la trama) de un mensaje llamado “Dirección Solicitada”.

3.4.4.1 Clases de Configuración de los Dispositivos

Dependiendo de su configurabilidad, los nodos SAE J1939 están divididos en 4 categorías:

- **No Configurable:** La dirección del nodo ya está programada para este dispositivo.
- **Configurable por Servicio:** La dirección del nodo puede ser establecida o cambiada por una herramienta de servicio. Para lo cual el dispositivo debe estar en un modo especial de operación.

¹ Especifica características del fabricante, campo de aplicación, tipo de dispositivo, etc.

- **Configurable por Comando:** La dirección del nodo puede ser establecida durante su operación normal por medio de un mensaje llamado “Comando de Dirección”.
- **Auto-configurable:** El propio dispositivo es el que determina su dirección por medio de cálculo interno. A través de un mensaje “Dirección Solicitada” esta dirección es pedida a la red. Si surgen conflictos de dirección, localmente debe determinar una dirección alternativa.

3.4.4.2 Configuración Dinámica de Dirección de Nodo

En el proceso de Configuración Dinámica de Dirección de Nodos, un dispositivo envía en broadcast¹ un mensaje de “Dirección Solicitada” (PF con valor 238) con su actual dirección de nodo en el campo Dirección de Origen de su identificador junto con su nombre de dispositivo de 64 bits en el campo de datos. Si la dirección transmitida ya corresponde a otro, el dispositivo que se queda con ella es aquel que posee mayor prioridad en el campo de nombre de dispositivo². El nodo que pierde la dirección, debe determinar otra y enviarla de nuevo a través de un mensaje de Dirección Solicitada. Si por alguna razón el nodo “perdedor” no puede determinar una nueva dirección debe enviar un mensaje “Dirección Origen No Asignada”. Si un receptor de un mensaje “Dirección Solicitada” posee la misma dirección pero con mayor prioridad, debe enviar este mismo mensaje para indicarle al otro nodo que determine otra dirección para él.

A través del mensaje “Consulta de Dirección de Nodo” (PF de valor 234), un nodo puede consultar a cualquier otro que dirección y nombre de dispositivo posee. Esta función es usada por un nodo que desea establecer su dirección origen y le permite conocer cuáles direcciones ya están siendo usadas.

Para asignar una dirección de nodo específica a un dispositivo perteneciente a la clase “Configurable por Comando” y que no posee dirección de nodo, se le transmite un mensaje denominado “Comando de Dirección” el cual en su campo de datos contiene el nombre del dispositivo (8 bytes) y la dirección asignada a

¹ Dirección destino con valor 255.

² El nodo con el número más bajo como nombre, posee la prioridad más alta para poseer una dirección.

éste (1 byte), haciendo uso del protocolo de fragmentación¹ en broadcast. El receptor del mensaje responderá entonces con un mensaje “Dirección Solicitada” comunicando su nueva dirección a la red.

Usando los mensajes de administración de red mencionados anteriormente, existe la posibilidad de conflicto de identificador en caso de que por ejemplo, dos nodos transmitan un mensaje “Dirección Solicitada” con la misma dirección de nodo al mismo tiempo. Los mensajes tendrán el mismo identificador, pero diferente nombre del dispositivo. Este conflicto es descubierto cuando uno de los controladores CAN reconoce un bit, de su campo de datos, recesivamente transmitido como dominante, destruyendo el mensaje inmediatamente. Esto es luego reconocido por el otro controlador CAN. Esto podría conducir a que el nodo en cuestión alcance el estado de bus-off (suspendido), por ello es necesario emplear mecanismos especiales de retransmisión en el controlador CAN. Por ejemplo con nodos del mismo tipo y comportamiento, un intento de retransmisión terminaría en un nuevo conflicto, por tanto se debe insertar un intervalo de tiempo diferente para cada nodo antes de iniciar la retransmisión.

3.4.5 CAPA APLICACIÓN

La especificación SAE J1939/71 describe una capa de aplicación para vehículos, la cual describe los mensajes de esta capa (parámetros) así como grupos de estos mensajes (grupos de parámetros), con el fin de garantizar interoperabilidad entre sus componentes.

La última actualización de esta especificación incorpora varias características adicionales como un número total de hasta casi 150 mensajes definidos, los cuales brindan funciones de antirrobo, inyección, presión de llantas, turbo-cargador, control de combustible entre otras.

Definición de Parámetros

Asegura interoperabilidad entre nodos individuales, para ello describe:

¹ Pues se estarían transmitiendo 9 bytes.

- Longitud de datos en bits o bytes
- Tipo de mensaje
- “Número de Parámetro de Sospecha”, que sirve de referencia en mensajes de diagnóstico

Definición de Grupos de Parámetros

La definición de un Grupo de Parámetros incluye, además de la descripción de características de transmisión, una lista parámetros usados.

Un Grupo de Parámetros es por tanto descrito por los siguientes parámetros:

- Número de bytes de datos
- Número de “Data Page” (DP)
- Formato PDU “PF”
- Extensión PDU “DA/GE”
- Prioridad
- Número de Grupo de Parámetro “PGN”
- Lista de Parámetros

Definición de Mensajes de Diagnóstico

Además de la definición de parámetros y grupos de parámetros, esta capa también define mensajes adicionales para funciones de diagnóstico, los cuales permiten por ejemplo la lectura y escritura de áreas de almacenamiento y la eliminación de fallos.

3.4.6 CAMPO DE APLICACIÓN

Los protocolos y perfiles basados en SAE J1939 son usados en redes embebidas basadas en CAN implementadas ampliamente en sistemas de transportación y vehículos.

Los principales campos de aplicación de protocolos basados en SAE J1939 son:

SAE J1939 en camiones y buses

El perfil de aplicación SAE J1939-71 es usado en la mayoría de buses y camiones. Estas redes, basadas en CAN, controlan el motor así como unidades de control de cuerpo. En Norteamérica este perfil también es usado para comunicación camión/trailer. En Europa el perfil ISO 11992, también basado en SAE J1939, debe ser implementado por regulación de la administración europea. En particular, ésta se refiere a que los frenos deben estar conectados a través de redes ISO 11992.

SAE J1939 en vehículos agrícolas y de construcción

Vehículos agrícolas y de silvicultura, tales como tractores, segadoras y trilladoras usan dispositivos electrónicos enlazados a través de redes basadas en SAE J1939 (ISO 11873, "Isobus") dentro de aquellos vehículos. Por su parte, los vehículos de construcción (vehículos mineros, máquinas de construcción, camiones muy pesados) emplean redes SAE J1939-71 para el control de sus motores diesel.

SAE J1939 en trenes de pasajeros y de carga

SAE J1939 es implementado en varias locomotoras impulsadas por diesel para control de sus motores.

SAE J1939 en la electrónica marítima

El perfil de aplicación NMEA 2000 será implementado en naves y buques para comunicar sistemas de navegación. Otros perfiles basados en SAE J1939 están siendo desarrollados para su implementación en otras aplicaciones marítimas.

3.5 TIME-TRIGGERED CAN

Las tendencias de investigación, principalmente en el sector automotriz, establecieron que un modelo predefinido controlado por tiempo¹ proveería un esquema óptimo para una transmisión predefinida de mensajes dentro de sistemas CAN que requieren funciones de control en tiempo real². La conclusión fue el desarrollo de una capa sesión para CAN que soporte este modelo. Esta capa está situada sobre la capa enlace del protocolo CAN (Ver Figura 3.14) y provee una funcionalidad que transforma una red CAN controlada por eventos en una red controlada por tiempo, sincronizada globalmente a través de mensajes de referencia. Esta funcionalidad es provista por el Protocolo TTCAN (Time-Triggered CAN) el cual en la actualidad es un estándar ISO 11898-4 [ISO00-4] (actualizado en el 2004).

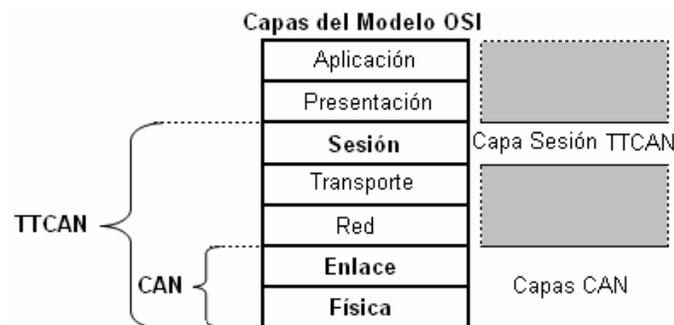


Figura 3.14: Modelo OSI identificando dónde operan TTCAN y CAN.

El Protocolo TTCAN es implementado en dos niveles:

Nivel 1:

- Garantiza la operación controlada por tiempo en una red CAN a través de “Mensajes de Referencia” transmitidos por un “Maestro de Tiempo”.
- Tolerancia a fallas del “Maestro de Tiempo” a través de potenciales Maestros.

¹ De acuerdo al principio “Acceso Múltiple por División de Tiempo” (TDMA).

² Tiempo real es un concepto referente a que el sistema realice cierta actividad en un periodo de tiempo definido, es decir que sea predecible, a la vez también es deseable que el sistema obtenga un alto grado de utilización.

Nivel 2:

- Establece una base de tiempo globalmente sincronizada.
- Corrige desviaciones entre los relojes de los nodos CAN con el Master Time.

3.5.1 PRINCIPIO DE OPERACIÓN TIME-TRIGGERED CAN

3.5.1.1 Ventanas de Tiempo, Ciclos Base y Matriz

La comunicación periódica de TTCAN está basada en “mensajes de referencia” que son transmitidos por un nodo denominado “maestro de tiempo”. Cada periodo que inicia con un mensaje de referencia es llamado un “ciclo base” (Ver Figura 3.15) y es subdividido en varios slots de tiempo (“ventanas de tiempo”). El “ciclo de tiempo” es reiniciado al inicio de cada ciclo base. El punto inicial de cada slot de tiempo es definido por una “marca de tiempo”.

Se distinguen tres tipos de ventanas de tiempo:

- Ventana Exclusiva
- Ventana de Arbitraje, y
- Ventana Libre

“Ventanas Exclusivas” son usadas para la transmisión de mensajes periódicos enviados sin contención¹. Solamente un nodo en particular de la red puede iniciar una transmisión en una ventana de tiempo exclusiva.

“Ventanas de Arbitraje” son usadas para comunicación generada por eventos, por ello existe la posibilidad de que varios mensajes de ésta clase deseen compartir un mismo slot de tiempo, dando lugar a conflictos de acceso al bus, que

¹ Cada mensaje tiene su slot de tiempo reservado exclusivamente para él.

son resueltos por el esquema de arbitraje del protocolo CAN. En caso de pérdida en la fase de arbitraje se deshabilita la retransmisión automática (para conservar el esquema de *scheduling*¹). Una retransmisión es iniciada en la próxima ventana de arbitraje asignada a un mensaje específico generado por algún evento.

“Ventanas Libres” son reservadas para extensiones futuras de la red.

La Figura 3.15 muestra la estructura principal de un ciclo base.

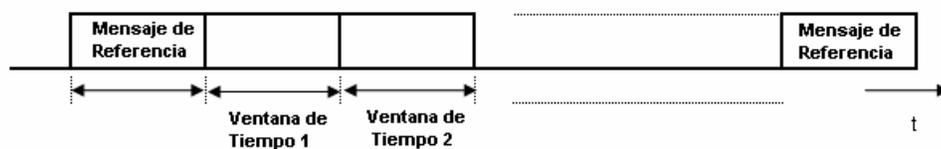


Figura 3.15: Estructura de un ciclo base TTCAN. Una ventana de tiempo puede ser exclusiva, de arbitraje o libre

Ya que un solo ciclo base no provee suficiente flexibilidad para especificar los diferentes periodos de transmisión en un sistema, TTCAN permite emplear varios ciclos base para construir un “ciclo matriz” (ciclo del sistema). Todos los mensajes de todos los nodos en la red son organizados como parte de un ciclo matriz, consistente de un número de ciclos base de acuerdo a la Figura 3.16.

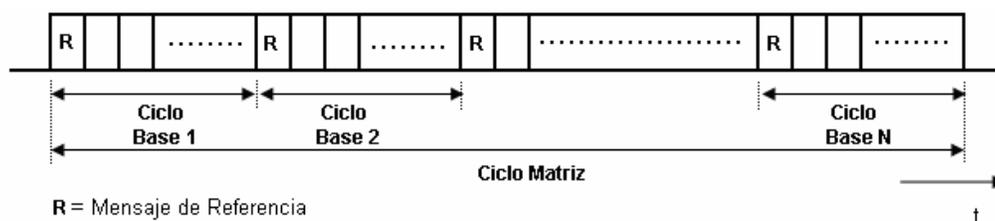


Figura 3.16: Ciclo matriz TTCAN

¹ Ver sección 3.5.2

Cada ciclo base inicia con un mensaje de referencia, sin embargo estos ciclos no deben poseer necesariamente el mismo tipo de ventanas, pero si la misma cantidad, además deben iniciarlas y terminarlas al mismo tiempo en todos los ciclos base que conforman un ciclo matriz (Ver Figura 3.17).

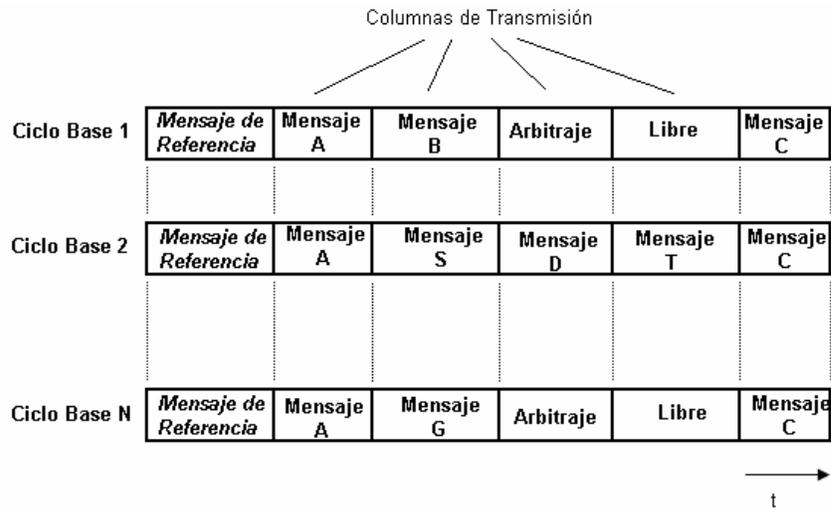


Figura 3.17: Estructura de un ciclo matriz TTCAN. El número de ciclos base por ciclo matriz máximo puede ser de 63. El ancho de cada columna puede ser establecido a cualquier tamaño, sin embargo todos los slots de una columna deben tener el mismo tamaño

El ciclo matriz comprende varios ciclos base, y se repite indefinidamente hasta que el sistema es apagado.

3.5.1.2 Mensajes de Referencia

Los “mensajes de referencia” son usados para sincronizar y calibrar las bases de tiempo de todos los nodos a la base de tiempo del maestro de tiempo, proveyendo un “tiempo global” de red.

Ya que la disponibilidad del maestro de tiempo es crucial para la operación del sistema, se provee un mecanismo para que un maestro con problemas pueda ser sustituido. Un mensaje de referencia está caracterizado por un identificador de 11 bits y es recibido y aceptado por todos los nodos excepto por el time master (que es el que lo transmite). Los primeros 8 bits del identificador caracterizan al

mensaje como uno de referencia, los 3 menos significantes especifican la prioridad¹ de hasta 8 potenciales maestros de tiempo. El primer byte del campo de datos del mensaje de referencia provee el número del ciclo base (cuenta de ciclo), también indica por medio de una “bandera de indicación” si el próximo mensaje de referencia es generado por eventos o por tiempo (Figura 3.18), el resto de bytes de este campo están destinados al uso de la aplicación.

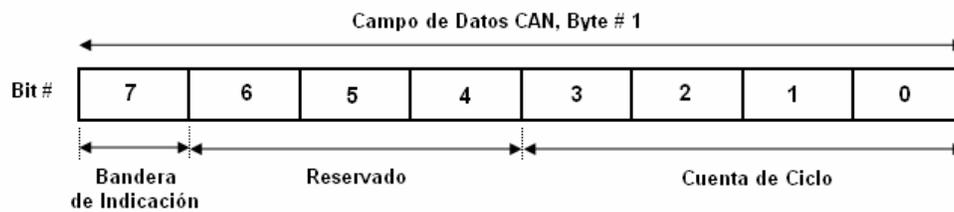


Figura 3.18: Primer byte de datos de un mensaje de referencia de nivel 1

Por su parte un mensaje de referencia de nivel 2, además de las características provistas por el de nivel 1, contiene el tiempo global provisto por el master.

Usualmente el mensaje de referencia es transmitido en intervalos equidistantes de tiempo. Opcionalmente la transmisión del mensaje de referencia puede también ser generado por un evento específico.

3.5.2 ESQUEMA DE TRANSMISIÓN DEFINIDA (*SCHEDULING*)

La matriz del sistema determina cual mensaje es transmitido en una ventana de tiempo. Todas las ventanas de tiempo de la matriz del sistema son continuamente repetidas en el ciclo de matriz. El ciclo de matriz consiste de uno o más ciclos base. Cada ciclo base inicia con un mensaje de referencia. Los mensajes de referencia mantienen la sincronización de todos los nodos en la red. El ciclo de tiempo de un ciclo base es reiniciado en cada nodo en los mensajes de referencia. La secuencia de las ventanas dentro del ciclo base es controlada por “marcas de tiempo” que son comparadas al actual ciclo de tiempo e indican el inicio y terminación de una ventana. En nodos sincronizados a la red TTCAN, la

¹ Ver sección 3.5.3

transmisión y recepción de mensajes es controlada por marcas de tiempo, para envío de mensajes estas marcas son llamadas “Tx_Triggers”. De igual manera se han definido “Rx_Triggers” para revisar puntos específicos de tiempo para la recepción de mensajes periódicos. Una ventana particular de tiempo es abierta cuando su correspondiente marca de tiempo es alcanzada y es cerrada antes de que la siguiente ventana sea abierta.

No es necesario que toda la matriz del sistema sea conocida por cada nodo. Un nodo solamente debe conocer cuando transmitir sus mensajes, cuando verificar la recepción de mensajes y cuando revisar la ausencia de mensajes de referencia.

3.5.3 TOLERANCIA A FALLOS Y DETERMINACIÓN DEL MAESTRO DE TIEMPO ACTIVO

Como el maestro de tiempo juega un papel fundamental, TTCAN provee una función de tolerancia a fallos al respecto. Hasta 8 nodos de una red TTCAN pueden ser potenciales maestros de tiempo, pero solo uno de ellos puede llegar a ser el maestro verdadero cuando se establece la comunicación controlada por tiempo en el bus. Al arrancar el sistema, después de resetear el hardware, todos los potenciales maestros realizarán la función de éste e intentarán transmitir (de acuerdo a su prioridad y tiempo de espera¹) un mensaje de referencia. El nodo con la prioridad más alta de convertirse en el maestro usa el identificador CAN más alto y posee el menor tiempo de espera, convirtiéndose así en el verdadero maestro de tiempo cuando todos los potenciales maestros son reseteados en el mismo punto de tiempo.

Por otro lado, cuando el actual maestro de tiempo fracasa al enviar un mensaje de referencia, el bus permanece desocupado y el potencial maestro con el tiempo de espera más corto transmitirá un mensaje de referencia tan pronto como su tiempo de espera haya terminado y se convertirá en el nuevo maestro. Si dos o más potenciales maestros intentan enviar un mensaje de referencia al mismo

¹ Marca especial de tiempo, relativa al Tx Trigger , aplicada solamente a los potenciales maestros de tiempo.

tiempo, el mecanismo de arbitraje de bit asegurará que el nodo de mayor prioridad se convierta en el maestro.

Durante la inicialización o reseteo de un potencial maestro, es posible que un nodo cuya prioridad no sea la más alta, erróneamente se convierta en el actual maestro de tiempo. Entonces es necesario una re-sincronización que no afecte los ciclos base ya establecidos por este maestro. Para lograrlo, un potencial maestro con prioridad de referencia más alta que el actual, decrementará un tiempo adicional al de espera cada vez que reciba un mensaje de referencia hasta¹ que este tiempo adicional expire, entonces en el próximo intento su tiempo de espera será menor que el del actual maestro por lo que tendrá éxito al enviar su mensaje de referencia y convertirse en el nuevo maestro de tiempo.

3.5.4 SINCRONIZACIÓN LOCAL Y GLOBAL

En sistemas TTCAN de nivel 2 todos los nodos capturan su tiempo local en el punto de muestreo del bit SOF del mensaje de referencia enviado por el maestro de tiempo. Por otro lado el maestro provee su tiempo como el tiempo global correcto como parte del mensaje de referencia (nivel 2). Con esta información cada nodo puede corregir la diferencia existente entre el tiempo global del maestro y su tiempo local, y usar el tiempo global corregido en el próximo ciclo base

Adicionalmente, Sistemas TTCAN de nivel 2 introducen un mecanismo adicional que asegura que los tiempos local y global tengan la misma velocidad², el cual está basado en la recepción de dos tiempos de referencia locales y globales, a partir de los cuales se obtiene un valor que indica el nivel de corrección aplicable en cada controlador de cada nodo.

¹ Hasta que este tiempo adicional expire o hasta que reciba un mensaje de referencia de otro potencial maestro de mayor prioridad a la suya.

² Esto asegura que todos los nodos tienen una visión consistente del tiempo global.

3.5.5 CAMPO DE APLICACIÓN

El principal campo de aplicación del sistema TTCAN tiene lugar en los sistemas x-by-wire (xbws) aplicados al campo automotriz (vehículos de pasajeros).

Con los sistemas X-by-Wire se permite el control coordinado de la dirección y la frenada del vehículo, además de ofrecer integración adicional con otras funciones del chasis y del vehículo. Estos sistemas son programables y proporcionan una oportunidad para optimizar el manejo, la conducción y el rendimiento de los sistemas de seguridad, que incluyen el control de la estabilidad del vehículo y evasión de colisiones.

CAPITULO 4

APLICACIONES

4.1 APLICACIONES Y CARACTERISTICAS DE SISTEMAS CAN

4.1.1 CAN EN AUTOMATIZACIÓN DE LA PRODUCCIÓN

En la actualidad, las plantas modernas que solían usar costosos sistemas de cableado, los han reemplazado por sistemas de bus seriales (sistemas bus de campo) para el intercambio de datos entre dispositivos de campo y dispositivos de control. En el área de automatización de la producción son particularmente útiles para la realización de funciones de control distribuido en tareas tales como ensamblaje, transporte, montaje y manipulación de piezas, así como para el establecimiento de comunicación entre robots industriales, computadores y periféricos, etc.

La instrumentación técnica del dominio de campo de plantas de producción (piso de planta) está caracterizada por el uso de Controladores Lógicos Programables (PLCs) o unidades de control equivalentes de aplicación industrial junto con un número de dispositivos periféricos de entrada y salida. Generalmente todos estos dispositivos están conectados a la respectiva unidad de control a través de sistemas de bus de campo, siendo especialmente importantes aquellos basados en CAN.

La estructura de comunicación entre una unidad de control y los dispositivos conectados a ella generalmente corresponde a una estructura “uno-varios”, sin embargo la creciente asignación de inteligencia a dispositivos periféricos junto con la distribución de funciones de procesamiento a varios de éstos ha provocado la necesidad de estructuras de comunicación “varios-varios” en el área de automatización de la producción.

La Tabla 4.1 muestra las características generales de estos sistemas.

Parámetros de Red	
Máxima extensión de red	Tecnología de procesos: 1000 m Tecnología de manufacturación: 300 m
Número de nodos	32 por segmento
Estructura de red	Estructura de red jerárquica ¹
Topología	Bus, anillo, árbol
Parámetros y Estructura de Comunicación	
Estructura de comunicación	Intercambio de datos entre la unidad de control central y dispositivos periféricos simples: “uno-varios” (un solo maestro). Intercambio de datos entre unidades de control y periféricos inteligentes, además de entre los mismos periféricos: “varios-varios” (multimaestro).
Velocidad de transmisión y tamaño del mensaje	Para la comunicación entre unidades de control: Bajas velocidades de transmisión con mensajes largos (100-200 bytes). Para la comunicación entre la unidad de control y unidades simples de periféricos: Altas velocidades de transmisión con mensajes cortos (varios bytes) Para configuración de unidades de control y dispositivos periféricos inteligentes: altas velocidades de transmisión con longitudes de mensaje en el orden de 10 a 100 bytes
Tiempo de latencia del mensaje	Tecnología de procesos: 100 – 300 ms Tecnología de manufacturación: 10 – 100 ms Sistemas drives: 1 – 10 ms
Requerimientos especiales	Sistemas drives: Mecanismos de sincronización
Integridad de Datos	
Seguridad de transmisión	Altos requerimientos para evitar disturbios ocasionales
Tiempo de detección de falla	10 – 100 ms
Tecnología de transmisión	Tecnología eficiente en costo. Transmisión simultánea de datos y de alimentación cuando sea necesario.
Estandarización	Modelo de comunicación abierta. Ínter cambiabilidad de dispositivos (perfiles de dispositivos)
Herramientas de configuración	Se requieren de herramientas eficientes para configuración

Tabla 4.1: Características generales de sistemas de automatización basados en CAN

4.1.2 CAN EN SISTEMAS MÓVILES

El uso de CAN en vehículos motorizados brinda una significativa mejora en sus sistemas de conducción, al sincronizar los procesos asignados en varias unidades

¹ Ver Anexo B.

de control electrónico a través del intercambio de datos entre éstas. La aplicación de CAN también resulta interesante en el área del networking de componentes electrónicos de confort en el automóvil, tales como: aire acondicionado, ajuste de posición de los asientos y espejos, ascensor de vidrios, sistemas de protección a robos, bloqueo central, control de iluminación, etc.

Parámetros de Red	
Máxima extensión de red	40 m.
Número de nodos	De 10 – 32 por segmento
Estructura de red	Redes separadas para varias áreas de diferente función
Topología	Bus
Parámetros y Estructura de Comunicación	
Estructura de comunicación	Intercambio de datos entre las unidades de control y los componentes periféricos en forma “varios-varios” (multimaestro).
Velocidad de transmisión y tamaño del mensaje	Comunicación entre unidades de control y componentes periféricos: Muy altas velocidades de transmisión con solamente pocos bytes por mensaje. Para configuración de unidades de control durante la inicialización del sistema: Altas velocidades de transmisión con tamaños de mensajes en el orden de 10 a 100 bytes.
Tiempo de latencia del mensaje	Unidades de milisegundos para mensajes de alta prioridad Decenas de milisegundos para el resto
Requerimientos especiales	Mecanismos de sincronización para la comunicación entre dispositivos de control
Integridad de Datos	
Seguridad de transmisión	Altos requerimientos para condiciones extremas de interferencia
Tiempo de detección de falla	Entre 1 a 10 ms
Tecnología de transmisión	Eficiente en costo, detección y desactivación de nodos averiados.
Estandarización	Modelo de comunicación abierta
Herramientas de configuración	Requiere de herramientas eficientes de prueba

Tabla 4.2: Características generales de sistemas de comunicación internos del vehículo basados en CAN

Cabe añadir que CAN no solamente es implementado en automóviles y vehículos comerciales, sino en todo tipo de sistema móvil, como por ejemplo: ascensores, máquinas agrícolas, grúas, camiones elevadores, vehículos de transporte público, camiones de basura, vehículos militares, trenes, buques, aviones y máquinas de construcción.

La Tabla 4.2 muestra las principales características de los sistemas de comunicación interna del vehículo.

4.1.3 CAN EN AUTOMATIZACIÓN DE EDIFICIOS

Los sistemas más modernos de automatización de edificios están basados en sistemas seriales de transmisión de datos para el intercambio de información entre interruptores, sensores, sistemas de iluminación, actuadores y otros sistemas de control, lo que ha permitido una operación segura y económica de grandes edificaciones.

Parámetros de Red	
Máxima extensión de red	Hasta 1000 m por segmento de red
Número de nodos	64 por segmento
Estructura de red	Estructura jerárquica de red
Topología	Bus, árbol
Parámetros y Estructura de Comunicación	
Estructura de comunicación	Intercambio de datos entre la unidad de control central y los componentes periféricos en forma "varios-varios" (multimaestro)
Velocidad de transmisión y tamaño del mensaje	Comunicación entre unidades de control y componentes periféricos: muy altas velocidades de transmisión con pocos bytes por mensaje. Para configuración de dispositivos de control y periféricos inteligentes: Altas velocidades de transmisión con tamaños de mensajes en el orden de 10 a 100 bytes.
Tiempo de latencia del mensaje	100 a 500 ms
Integridad de Datos	
Seguridad de transmisión	Pocos requerimientos
Tiempo de detección de falla	100 ms
Tecnología de transmisión	Tecnología eficiente en costo. Transmisión simultánea de datos y de alimentación.
Estandarización	Modelo de comunicación abierta. Inter cambiabilidad de dispositivos (perfiles de dispositivos)
Herramientas de configuración	Se requieren de herramientas eficientes para configuración

Tabla 4.3: Características generales de sistemas de automatización de edificios basados en CAN

La Tabla 4.3 muestra las principales características que debe poseer un sistema de comunicación para automatización de edificios.

4.1.4 CAN EN SISTEMAS EMBEBIDOS

Aplicaciones importantes en ésta área, incluyen el networking de sensores, actuadores, paneles de operación y unidades de control al interior de máquinas así como el networking de dispositivos dentro de sistemas de monitoreo, control y medición.

Parámetros de Red	
Máxima extensión de red	10 m
Número de nodos	Hasta 32
Estructura de red	Redes independientes
Topología	Bus
Parámetros y Estructura de Comunicación	
Estructura de comunicación	Intercambio de datos entre la unidad de control central y dispositivos periféricos simples: “uno-varios” (un solo maestro). Intercambio de datos entre unidad (es) de control y componentes periféricos inteligentes: “varios-varios” (multimaestro)
Velocidad de transmisión y tamaño del mensaje	Para la comunicación entre unidad (es) y periféricos: La velocidad de transmisión es específica a la aplicación, generalmente con unidades de bytes por mensaje. Para la configuración de unidades de control y periféricos se transmite de 100 a 1000 bytes por mensaje a bajas velocidades. Para la transferencia de datos de medición se envían varios miles de bytes a bajas velocidades.
Tiempo de latencia del mensaje	1 – 100 ms
Requerimientos especiales	Específicos a la aplicación
Integridad de Datos	
Seguridad de transmisión	Altos requerimientos de seguridad para ambientes altamente perturbados
Tiempo de detección de falla	1 – 10 ms
Tecnología de transmisión	Eficiente en costo y a prueba de fallos
Estandarización	Modelo de comunicación abierta (no obligatorio).
Herramientas de configuración	Herramientas específicas para pruebas y servicios requeridos

Tabla 4.4: Características generales de sistemas de comunicación embebidos basados en CAN

Aplicaciones de este tipo son por ejemplo: máquinas de medición y prueba, celdas de producción, robots, máquinas de impresión, máquinas de empaquetado, máquinas textiles, equipo médico de cualquier tipo, etc. Debido a

la diversidad de aplicaciones, no se puede establecer un criterio general de requerimientos para éstas. La Tabla 4.4 resume varios criterios relevantes a este campo de aplicación.

4.1.5 EJEMPLOS DE APLICACIÓN DE SISTEMAS CAN

4.1.5.1 Campo Automotriz

La industria automotriz utiliza CAN como una red IVN (in-vehicle network) para funciones de control del motor, electrónica de cuerpo (control de puertas y de techo) y redes de confort (aire acondicionado, iluminación).

Algunos de los ejemplos más relevantes de éste tipo de redes son:

- Sistemas de seguridad vehicular activos.- Por ejemplo Sistemas “ESC” (Electronic Stability Control) pueden detectar situaciones inestables de conducción y hacer correcciones automáticas para evitar que el conductor pierda el control.
- Nuevos desarrollos en redes “powertrain¹”.- Menos consumo de combustible, combustión más limpia y dinámica de conducción mejorada.
- Sistemas de prevención de accidentes.- Dos sistemas futuristas de asistencia de conducción han sido recientemente introducidos:
 - Sistemas de Asistencia de Parqueo
 - Sistemas de Control de Crucero
- Freno Inteligente.- El sistema calcula la presión ideal de frenado dependiendo de la situación, para lo cual necesita información de presión individual y ángulo de dirección de cada neumático y aceleración. Estos datos son intercambiados a través de redes IVN basadas en CAN.

¹ “Powertrain” en el mundo automotriz consiste de todos los componentes del sistema de conducción de un vehículo, tales como: motor y transmisión entre otros.



La mayoría de constructores europeos de autos usan redes IVN basadas en CAN, mientras que los constructores americanos recientemente han iniciado su implementación.

Muchos de los fabricantes europeos y americanos de buses y camiones ya han implementado CAN como redes IVN para aplicaciones power-train así como para la comunicación entre camión y remolque. Algunos sistemas de control de superestructuras de camiones (tales como camiones de bomberos y mezcladoras de concreto) también usan CAN como red de control embebida.

En camiones agrícolas y de construcción CAN es usado como red embebida y como red IVN power-train. Unidades de control electrónico (ECUs) conectadas a redes CAN pueden controlar motores a diesel o unidades eléctricas si el vehículo es alimentado a batería. El rango de vehículos que emplean CAN incluye vehículos de minería, remolques de aviones, camiones de construcción de carreteras así como vehículos de selvicultura, etc.

CAN también es empleado en una variedad de aplicaciones automotrices que incluyen tranvías, trenes subterráneos y trenes que recorren largas distancias tanto de pasajeros como de carga en áreas tales como: control de motores, sistemas de compuertas y control de frenos, entre otros.



4.1.5.2 Electrónica Marítima

En electrónica marítima, redes CAN son usadas en barcos, naves y buques como redes embebidas en subsistemas y como redes de integración que conectan varios de estos subsistemas.

4.1.5.3 Electrónica Aeroespacial

CAN es usada como red backbone en aeronaves para los sensores de estado de vuelo, sistemas de navegación y exploración instalados en la cabina del piloto. CAN también es empleado en aplicaciones aeroespaciales de control de motores tales como control de combustible, bombas y actuadores lineales.

4.1.5.4 Sistemas de Automatización

En automatización de fábricas CAN es utilizado para interconectar máquinas, unidades de control de procesos y subsistemas de producción. Aplicaciones típicas incluyen por ejemplo:

- **Automatización de sistemas de almacenamiento.**- Un ejemplo lo constituye el Grupo Munz-Magenwirth (Productor de hardware y software para diferentes marcas industriales), cuyo sistema de automatización está caracterizado por un muy rápido acceso a casi 20.000 diferentes artículos.
- **Producción de componentes automotrices.**- Tales como sistemas de ensamblaje de asientos.



- **CAN en el control de sistemas de grúa para contenedores.**- Conveniente para grandes puertos marítimos, en donde es necesario asegurar un adecuado manejo de los contenedores a través de un sistema seguro de bus basado en CAN.

CAN también es utilizado en automatización de edificios, en funciones tales como administración de calefacción, ventilación, iluminación, puertas y aire acondicionado. Funciones adicionales incluyen el control de sistemas de alarma, rociadores de agua y control de audio y video entre las más importantes. Suiza y Alemania son los países más adelantados al respecto.



4.1.5.5 Sistemas Embebidos

En máquinas de control industrial CAN es utilizado como una red embebida, por ejemplo en máquinas textiles, máquinas de impresión, máquinas de moldeo a

inyección o máquinas empacadoras. Estas redes de control embebidas enlazan dispositivos independientes tales como módulos I/O. Muchas de estas aplicaciones son orientadas al control de movimiento.



Elevadores y escaleras mecánicas siempre han utilizado redes CAN embebidas, todos los componentes de los elevadores tales como paneles, controladores, puertas, barreras de luz etc. Son enlazados entre sí y controlados a través de redes CAN. De igual manera existen sistemas de control de escaleras mecánicas enteramente enlazados con CAN.

CAN también es usada como red embebida en dispositivos médicos tales como en máquinas de rayos x. Cuartos completos de operación son equipados con una red CAN que administra todas las funciones. CAN también es usada como red embebida en camas de pacientes. Además sistemas de control de hospitales completos (control de voltaje, unidades de control, software de visualización, etc.) son enlazados a través de redes CAN.

4.2 ASPECTOS DE DISEÑO E IMPLEMENTACIÓN DE SISTEMAS CAN

4.2.1 ALTERNATIVAS DE IMPLEMENTACIÓN

En sus inicios, los sistemas CAN fueron implementados en base al estándar de capa enlace de datos y de señalización física [ISO99-1]. En la actualidad, con la disponibilidad de varios protocolos de capa superior tales como CAL, CANopen o DeviceNet (entre los más importantes), es posible emplear los servicios de

protocolos de capa 7 (capa aplicación) y así permitir comunicación abierta. Al respecto, la inter-cambiabilidad de dispositivos también puede ser asegurada por el uso de perfiles predefinidos de dispositivos.

En general, existen tres tipos de alternativas para la implementación de redes CAN:

- Implementación basada en servicios de capa 2
- Implementación basada en protocolos estandarizados de capa 7
- Implementación basada en perfiles de dispositivos

La decisión de si un sistema debería ser implementado basado en capa 2, protocolo de capa aplicación o en un perfil determinado, depende de varios factores:

4.2.1.1 Implementación basada en servicios de Capa 2

Este tipo de implementación provee los servicios necesarios para el envío y solicitud de mensajes, permitiendo que los procesos de aplicación involucrados operen directamente con los mensajes, cada uno especificado por su identificador (relación directa entre la aplicación y los procesos de comunicación). No satisface los requerimientos hechos en sistemas abiertos, constituyéndose así en sistemas cerrados (es decir que una implementación de este tipo solo servirá al sistema específico para el cual fue diseñada, cualquier cambio en éste requiere de cambios y adaptaciones adicionales en el mismo).

Los identificadores necesarios para la transmisión de mensajes, así como los formatos de datos de procesos, comandos y parámetros son determinados en función de la aplicación. Los servicios de administración requeridos en cada sistema distribuido y cualquier servicio adicional necesario (por ejemplo funciones de fragmentación de mensajes) deben ser provistos por los mismos procesos de aplicación.

Por otro lado constituyen soluciones optimizadas con requerimientos específicos de comportamiento en la comunicación y tiempos de latencia.

Como la aplicación está en contacto directo con las tareas de comunicación, esto provoca que los tiempos de ejecución de software sean mínimos.

4.2.1.2 Implementación basada en protocolos estandarizados de capa aplicación

Su propósito es proveer una comunicación estándar e independiente de la aplicación. A diferencia de la implementación basada en capa 2, gracias a los “servicios de aplicación” definidos por una capa aplicación (capa 7) se obtiene total independencia entre los procesos de comunicación y la aplicación. La ventaja más importante de usar una capa aplicación estandarizada es la provisión de la base para comunicación abierta. Adicionalmente un protocolo de capa superior incluye una serie de características especiales que soportan la implementación y adaptación de redes de control CAN, tales como administración de red, funciones y mecanismos de prueba de consistencia de objetos de comunicación definidos y asignación dinámica de identificadores.

Un protocolo de capa aplicación estandarizado provee, entonces, a la red CAN todos los objetos y servicios necesarios para comunicación y administración de la misma. A diferencia de otros sistemas de bus de campo, especial cuidado fue puesto en la definición de los protocolos de capa 7 de CAN, de tal manera que no sea necesario añadir información adicional de control con el fin de que la característica de tiempo real del sistema no sea disminuida comparada con el uso directo de servicios de capa 2.

4.2.1.3 Implementación basada en un perfil estandarizado

Debido a que las implementaciones basadas en un protocolo de capa 7, son independientes de la aplicación, son necesarias definiciones adicionales específicas a la aplicación de un sistema en particular. Por otro lado, en cambio, con la disponibilidad de perfiles de dispositivos y comunicación estandarizada extensas definiciones para la funcionalidad de la aplicación y de los dispositivos ya están especificadas. Por medio de estas definiciones la estructura de una aplicación implementada en un dispositivo es predefinida. Funciones de sistema

así como comportamiento de la aplicación e interfaces de datos son definidos por el perfil. Como resultado, estructuras de comunicación y posibles procesos de un sistema son también predeterminados. Este tipo de implementación, entonces, limita la flexibilidad del sistema hasta cierto punto, lo cual ocurre en favor de una intercambiabilidad e integrabilidad simplificada de dispositivos de otros fabricantes en estas redes.

La Figura 4.1 muestra los diferentes niveles de esfuerzo en la implementación de las alternativas presentadas anteriormente. La opción A muestra una implementación pura de capa 2, la opción B grafica una implementación basada en una capa aplicación estandarizada tal como CAL o CAN Kingdom¹, y la opción C indica una implementación basada en un perfil de aplicación estandarizado tal como CANopen o DeviceNet. Los bloques sin color son desarrollados por el programador de la aplicación.

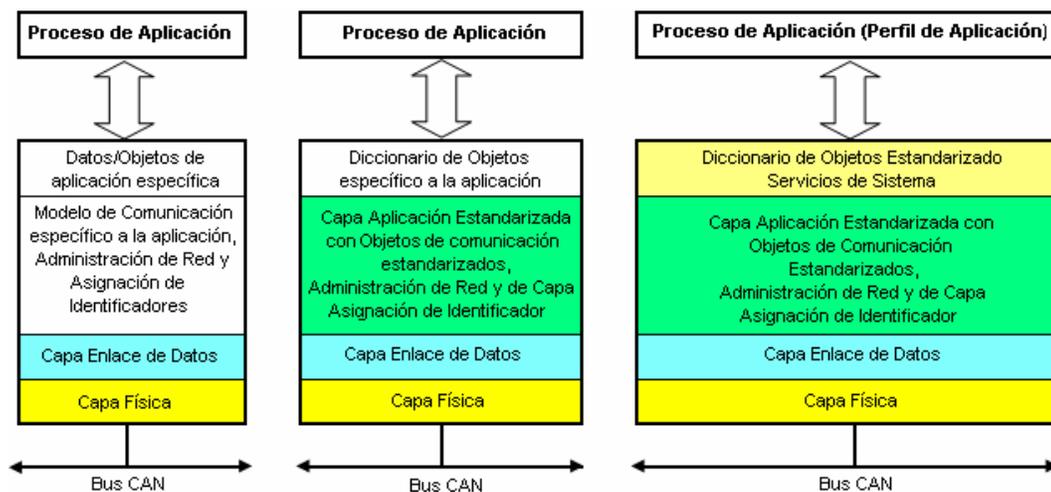


Figura 4.1: Alternativas de implementación de Sistemas CAN

Para el diseño de sistemas CAN basados en protocolos de capas superiores, existen aquellos basados en SAE J1939, definidos para un campo específico de aplicación. Adicionalmente, como ya se ha visto, se cuenta también con otros de

¹ Al igual que SDS constituyen otros protocolos de capa superior basados en CAN, aunque de menor relevancia con respecto a los estudiados.

aplicación universal basados en capa aplicación o en un perfil estandarizado, tales como: CAL, CAN Kingdom, DeviceNet, CANopen y SDS.

CANopen y DeviceNet actualmente constituyen las soluciones más comunes en la implementación de redes o sistemas de control y automatización estandarizados basados en CAN. Además de los mecanismos de comunicación, estos protocolos también definen perfiles para diversos dispositivos de tecnología de automatización, ofreciendo así gran flexibilidad en la implementación de estos sistemas en diferentes campos de aplicación. La interfaz al programa de aplicación, (Diccionario de Objetos en CANopen y Modelo de Objetos en DeviceNet) la cual es estandarizada para cada aplicación, permite una integración simple de componentes tales como módulos I/O, sensores o drives en estos sistemas de control.

A diferencia de CANopen y DeviceNet, la especificación CAL solo define procesos generales de comunicación (objetos y servicios de comunicación, distribución de identificador, administración de capa y red), no define contenidos de datos ni objetos de comunicación específicos de dispositivos. El usuario, entonces no es forzado a seguir una norma específica de implementación y puede adaptar el sistema de comunicación de acuerdo a los requerimientos particulares de su aplicación. CAL es mayormente implementado en campos de aplicación, tales como tecnología médica o de medición, así como en la implementación de sistemas de control con unidades inteligentes descentralizadas.

La selección de un determinado protocolo de capa superior no solo depende de la funcionalidad requerida por el sistema a implementarse, sino también de cuestiones de estrategia de mercado. Mientras CAL y en especial CANopen se han establecido fuertemente en el mercado europeo, DeviceNet ha hecho lo propio en aquellos segmentos de mercado en los que la Compañía Rockwell Automation está representada, esto comprende a los continentes americano y asiático.

4.2.2 DISEÑO DE REDES DE CONTROL DISTRIBUIDO BASADAS EN CAN

La Figura 4.2 ilustra un proceso sistemático para diseñar redes de control basadas en el Protocolo CAN de 2 capas. Ya que CAN utiliza un sistema de comunicación orientado al mensaje, el diseño de estos sistemas está fundamentado en el número de mensajes específicos a su aplicación.

El diseño de una red CAN, por tanto, empezará con el análisis del número y tipo de mensajes de aplicación (datos o señales) y de la relación de comunicación entre los transmisores y receptores de éstos. Como resultado de este análisis, serán conocidos: el número total de mensajes de aplicación específica, una relación de comunicación productor-consumidor junto con sus tiempos de latencia permitidos.

El siguiente paso es mapear los mensajes de aplicación específica a un sistema de grupos de mensajes, lo cual consiste en agrupar mensajes en determinados grupos de éstos y enviarlos en el campo de datos del mensaje CAN de acuerdo al modo de transmisión¹ requerido. Subsecuentemente a los grupos de mensajes obtenidos para el sistema se les asigna su respectivo identificador para la formación del mensaje CAN (objeto CAN) correspondiente.

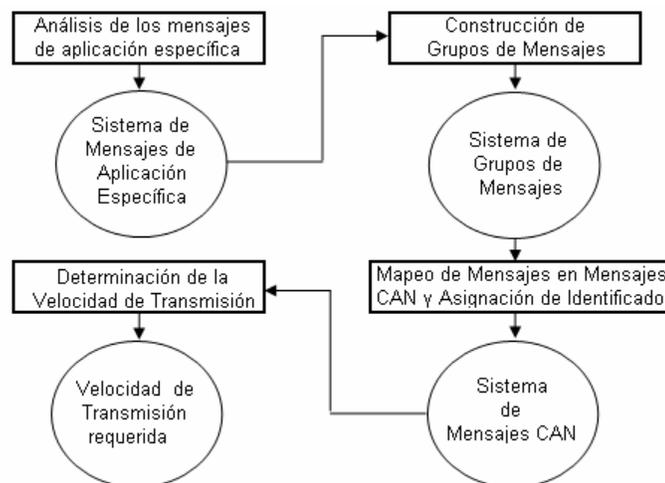


Figura 4.2: Diseño sistemático de redes CAN para implementaciones de basadas en el protocolo CAN de 2 capas

¹ Remotamente solicitado, transmisión cíclica o controlada por eventos.

El sistema de mensajes CAN permite determinar la velocidad de transmisión del sistema, basado en los requerimientos de latencia máxima para los mensajes de alta prioridad¹.

4.2.2.1 Mapeo de Mensajes de Aplicación Específica en Grupos de Mensajes

La efectividad de una red CAN a parte del modo de transmisión de sus mensajes (solicitud remota, transmisión cíclica o por eventos) depende mucho de la agrupación de éstos en grupos de mensajes. En general, mensajes de información cuya transmisión conjunta es importante (por ejemplo información de la misma naturaleza) deberían ser agrupados y transmitidos en un “grupo de mensajes” con el objetivo de minimizar el número de mensajes CAN, la carga del bus y el esfuerzo administrativo.

La Figura 4.3 muestra el principio de agrupación de mensajes individuales en un grupo de mensajes².

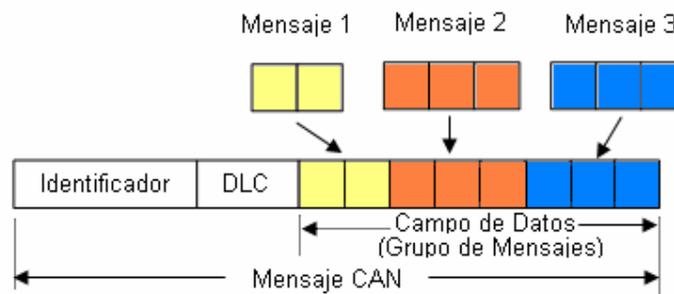


Figura 4.3: Agrupación de Mensajes de aplicación específica en un grupo de mensaje

Como la Figura 4.4 lo ilustra, también es posible el “uso múltiple” (multiplexación) de un mensaje CAN, con ello se puede transmitir diferentes mensajes de información, para lo cual se inserta un “sub-identificador” en el mensaje. Este sub-identificador se ubica próximo al Identificador del mensaje CAN, ocupa 1 byte de su campo de datos y es el que especifica los mensajes individuales transmitidos. Este método es principalmente aplicado en la transmisión de parámetros de configuración como parte del proceso de inicialización del sistema.

¹ Se asume que en un sistema, la totalidad de sus mensajes han sido divididos en dos categorías: de alta y baja prioridad.
² El principio de agrupación de mensajes individuales (datos de aplicación) es aplicado en CANopen como “Mapeo PDO” (Ver sección 3.2.1.1), y en DeviceNet como “Ensamble” de Objetos de aplicación (Ver sección 3.3.1.3)

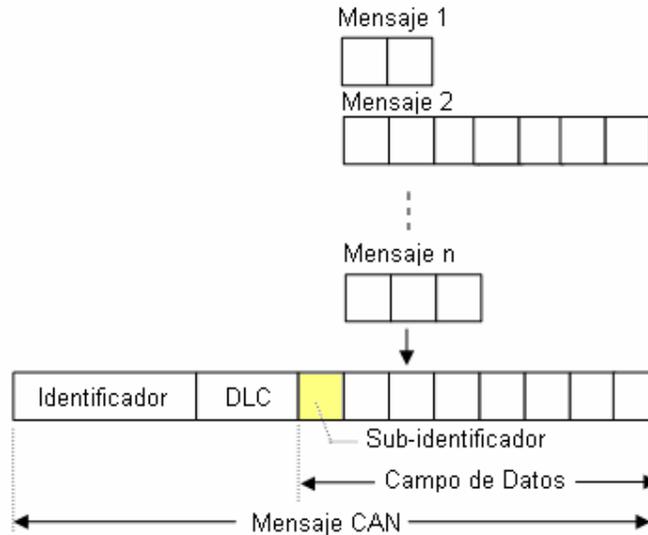


Figura 4.4: “Uso múltiple” de un mensaje CAN por medio de un sub-identificador en el campo de datos

Este principio permite transmitir un gran número de parámetros a través de un solo mensaje y por ende utilizando un solo identificador. Un requerimiento necesario para multiplexar un mensaje CAN es que el nodo receptor tenga capacidad de procesamiento en tiempo real¹ con los mensajes recibidos. CAL provee este método a través de una llamada “Variable múltiple” (sección 3.1.2). Por su parte CANopen aplica este principio en su “PDO Multiplexado” (sección 3.2.1).

4.2.2.2 Asignación de Grupos de Mensajes a Mensajes CAN

La formación de mensajes CAN está determinada por la asignación de un identificador al grupo de mensajes obtenido. Con el formato básico de trama, el identificador cuenta con 11 bits, es decir que pueden ser diferenciados hasta 2048 mensajes CAN. Con una estrategia apropiada de asignación, esta cantidad resulta casi siempre suficiente para aplicaciones industriales en general.

Adicionalmente, también es posible utilizar un identificador extendido, con el cual 29 bits están disponibles para su representación. La gran cantidad de mensajes

¹ Se refiere a que el receptor sea capaz de saber con absoluta certeza, cuando inicia y termina uno de los mensajes transmitidos dentro de un mensaje multiplexado.

disponibles hace posible una definición estática de un extenso número de mensajes específicos a un cierto campo de aplicación.

Sin embargo, esta ventaja es obtenida sacrificando la eficiencia del protocolo, puesto que se deben transmitir 20 bits adicionales de información de control por cada objeto CAN comparado con el formato estándar.

Existen dos estrategias básicas de asignación de identificador:

- **Asignación Orientada a la Prioridad**

Con este método de asignación, el valor del identificador determina la prioridad de acceso al bus, y con ello el tiempo de latencia máximo de un mensaje CAN. Principalmente los mensajes de alta prioridad de un sistema, son los que toman muy en cuenta este método. Basado en esta estrategia se puede emplear asignación dinámica de identificadores, tal como lo hace CAL (ver sección 3.1.3)

- **Asignación Orientada al Procesamiento**

Con este método, la asignación ocurre desde el punto de vista de filtrado y procesamiento simple de los mensajes aceptados. Esta estrategia puede ser aplicada dividiendo su identificador en grupos de bits de acuerdo a las características de los datos que transmite el mensaje del cual forma parte, tales características corresponden por ejemplo a: prioridad, dirección destino u origen, tipo de datos, etc. Esta estrategia, por ejemplo, es aplicada por el estándar SAE J1939 en el campo de vehículos comerciales (ver sección 3.4) en donde el campo identificador es subdividido en subcampos que indican características de sus datos: prioridad, dirección origen o destino y tipo de datos. Con ello se simplifica el preprocesamiento de los mensajes en los nodos.

En sistemas basados en el protocolo CAN de 2 capas, la asignación de identificadores a sus mensajes es realizada por el diseñador del sistema, es decir, que es estáticamente configurada en el programa de aplicación.

4.2.2.3 Determinación de la Velocidad de Transmisión del Sistema

El comportamiento de tiempo real (ver sección 4.2.3) de un sistema distribuido, está determinado por los tiempos de latencia debidos a la transmisión de mensajes a través del sistema de comunicación. Al respecto, los mensajes de alta prioridad juegan un rol importante, ya que podría darse el caso de que éstos continuamente ocupen el bus sin considerar la existencia de mensajes de menor prioridad. Para evitar esta situación se deben establecer intervalos de tiempo definidos durante los cuales solamente se transmitan mensajes de alta prioridad (ver sección 4.2.3) y otros intervalos para mensajes de prioridad baja.

Por tanto, el máximo tiempo de latencia posible de un mensaje es determinado por la duración total de todos los mensajes de alta prioridad, y por consiguiente la velocidad de transmisión es dependiente de los tiempos de latencia requeridos

4.2.3 COMPORTAMIENTO DE TIEMPO REAL

La característica de tiempo real se refiere a que un sistema posee absoluta certeza de en qué momento empieza y termina una transmisión específica a su aplicación, es decir que sea predecible. Un ejemplo de este comportamiento es representado por sistemas token passing y maestro-esclavo, en donde se conoce a ciencia cierta en qué momento exacto de tiempo un nodo podrá transmitir, y cuánto tiempo le tomará aquello.

Para garantizar este comportamiento, un sistema de comunicación debe poseer acceso determinístico al medio de transmisión, lo que le permite conocer los tiempos efectivos de latencia en la transmisión de sus mensajes. Como anteriormente se mencionó, para sistemas basados en CAN estos tiempos de latencia a su vez, están determinados por la duración total de todos los mensajes de alta prioridad que estos sistemas posean. Entonces en una red CAN es posible garantizar la característica de tiempo real, tomando en cuenta la cantidad de mensajes de alta prioridad que posea. Sin embargo se debe garantizar que este tipo de mensajes no acceda continuamente al medio y no permita que otros mensajes sean transmitidos, por ello se ha definido un intervalo de tiempo

(“tiempo mínimo de impedimento”) después del cual se permite una nueva transmisión de mensajes de alta prioridad. Dentro de este intervalo de tiempo, la transmisión de mensajes de baja prioridad es por tanto posible. Para asegurar que estos mensajes ocupen el bus solamente después de que el tiempo de impedimento haya expirado, los procesos de aplicación involucrados emplean funciones de monitoreo¹ de tiempo. Cabe resaltar que la consideración de una ventana adicional de tiempo para permitir la transmisión de mensajes de prioridad baja es necesaria solamente en sistemas cuyos mensajes de alta prioridad provocan un alto promedio de carga sobre el bus.

El siguiente ejemplo, Figura 4.5, muestra una estimación del tiempo máximo posible de reacción para sistemas CAN, es decir la mayor latencia de tiempo que puede poseer un mensaje en el peor de los casos. Esto es cuando se desea transmitir todos los mensajes de alta prioridad al mismo tiempo. Si un sistema posee 16 mensajes de alta prioridad, cada uno con 2 bytes de datos, entonces una trama estará compuesta por 73 bits (considerando los 47 bits del protocolo CAN, 16 bits de datos y 10 bits de relleno como caso extremo).

Entonces con una velocidad de transmisión de 1Mbps, el tiempo de transmisión por mensaje es de $73 \mu\text{seg}$. Luego para transmitir los 16 mensajes de alta prioridad se necesitará de $1,168 \text{ mseg}$. Suponiendo que adicionalmente, es necesario considerar una segunda ventana de tiempo para la transmisión de mensajes de prioridad más baja se asignan $0,332 \text{ mseg}$ para este requerimiento. Obteniendo un tiempo de impedimento total de $1,5 \text{ mseg}$ para los 16 mensajes de alta prioridad². Cabe resaltar que este tiempo de latencia vendría a ser efectivo solamente cuando todos los mensajes de alta prioridad tengan que ser transmitidos al mismo tiempo.

¹ Al utilizar protocolos de capa superior tales como CAL, CANopen o DeviceNet ésta funcionalidad es implementada por protocolo en cuestión.

² Considerando la ventana de tiempo destinada a la transmisión de mensajes de baja prioridad.

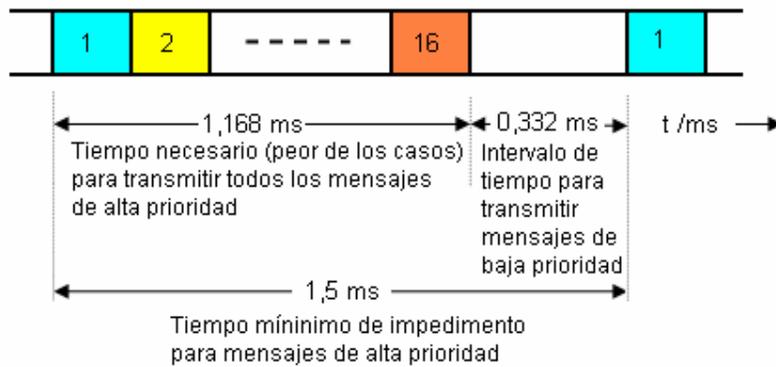


Figura 4.5: Estimación de la latencia máxima de tiempo de los mensajes CAN

4.2.4 IMPLEMENTACIÓN DE NODOS CAN

4.2.4.1 TIPOS DE NODO

Existen varios tipos de nodos CAN los cuales pueden ser agrupados en tres grupos:

- Nodo I/O (Tipo A)
- Nodo Microcontrolador (Tipo B y C)
- Nodo Interfaz a un Sistema Host (Tipo D y E)

En su forma más simple un nodo CAN consiste de un chip SLIO (Enlace Serial I/O) y de un transceiver (Figura 4.6). Este tipo de nodo es usado para proveer interfaces de procesos (Funciones I/O) a través del bus CAN. Sin embargo este tipo de nodo ha dejado de ser tomado en cuenta y por tanto discontinuado, la razón es que el costo por canal I/O obtenido es mayor comparado con soluciones basadas en microcontroladores las cuales a parte de un gran número de canales I/O proveen mayor flexibilidad y permite el pre-procesamiento de datos.

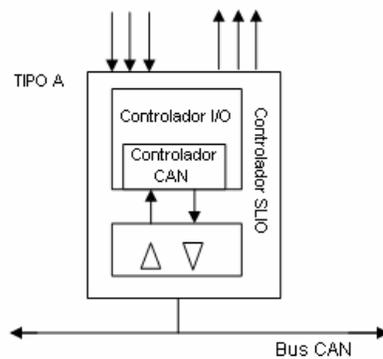


Figura 4.6: Nodo Tipo A

Los tipos B y C (nodos microcontroladores) entre ellos difieren solamente en la forma en que implementan su controlador CAN (Figura 4.7). El Nodo tipo B utiliza una unidad CAN integrada¹ en su microcontrolador, mientras que el tipo C está basado en un controlador “stand alone”². Además de enlazar procesos a través del bus CAN los nodos microcontroladores también permiten la implementación de funciones de procesamiento descentralizadas (nodos inteligentes³).

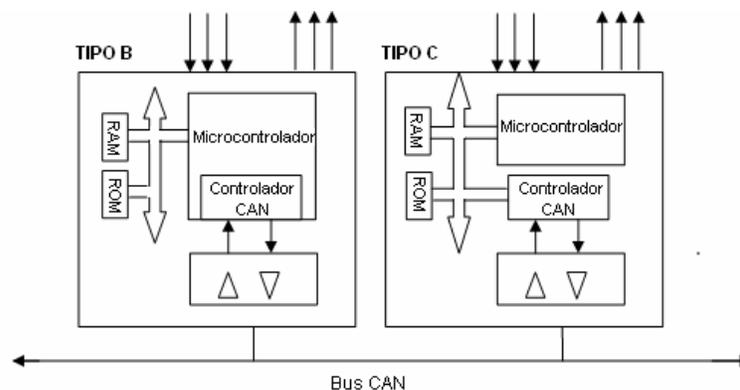


Figura 4.7: Nodo Tipo B y C

El nodo tipo D sirve para conectar el bus CAN a través de un controlador CAN a un “Sistema Host” (PCs, sistemas PLC, etc). Ver Figura 4.8. Este nodo consiste de una tarjeta CAN y del Sistema Host. La tarjeta CAN actúa solamente como interfaz entre el bus CAN y el bus paralelo del Sistema Host, por eso se le conoce

¹ Con este tipo de nodo, el Controlador de Host puede acceder al controlador CAN sin usar un bus externo, por tanto se obtienen tiempos de acceso más rápidos, reduciendo considerablemente la carga al Controlador de Host.

² Este tipo de implementación, provee la funcionalidad del controlador CAN en una unidad separada del microcontrolador.

³ Ver ejemplo de aplicación en sección 4.2.5

como “Interfaz Pasiva”. El sistema Host es quien controla el sistema de comunicación a través del bus CAN, es decir es el encargado de pasar los mensajes al controlador CAN y leer los mensajes enviados por este.

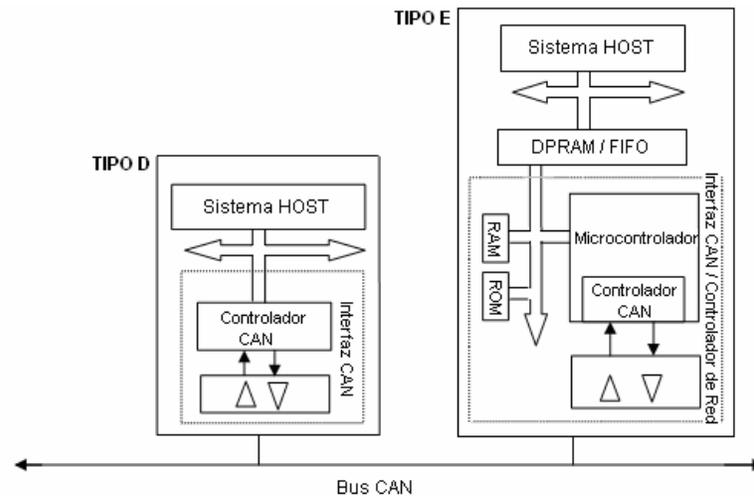


Figura 4.8: Nodo Tipo D y E

Los nodos tipo E constan de una tarjeta interfaz (tarjeta CAN), la cual lleva a cabo funciones tanto de interfaz como de procesamiento (Figura 4.8). Este tipo de nodo, separa completamente el Sistema Host del bus CAN a través de una interfaz simple (por ejemplo DPRAM O FIFO¹). El Sistema Host pasa solicitudes de transmisión a través de esta interfaz a la tarjeta CAN y recibe los mensajes CAN ya en forma procesada a través de las interfaces mencionadas. Con esto, la tarjeta CAN libera al Sistema Host de tareas relacionadas a comunicación por medio del sistema microcontrolador, siendo entonces llamado como “Interfaz activa”. Este tipo de nodo puede ser empleado también en CAL, CANopen o DeviceNet en donde las funciones de maestro son realizadas por la tarjeta CAN. En definitiva, el diseño tipo E es conveniente para sistemas en los que el Sistema Host debería ser aliviado tanto como sea posible de las tareas de comunicación, así como en sistemas cuyas aplicaciones provoquen elevados niveles de carga o de velocidad de transmisión sobre el bus

¹ DPRAM y FIFO actúan como un sistema normal de memoria, permitiéndole al Sistema Host y al Controlador CAN acceder directamente a la información que mutuamente intercambian. Es decir, DPRAM y FIFO actúan como “Interfaz de Datos” entre los procesos de aplicación y de comunicación.

4.2.4.2 Arquitectura del Controlador CAN

Un aspecto importante para el diseño del hardware de un nodo CAN es la selección del controlador CAN. Por un lado es posible escoger entre un extenso número de controladores de diferente precio y fabricante. Por otro, el controlador CAN escogido está en función de si su diseño esta basado en un concepto “BasicCAN” ó “FullCAN”.

La arquitectura de los controladores no es cubierta por el estándar CAN, esto hace que puedan existir diferencias sustanciales entre ellos. Las dos categorías principales (BasicCAN y FullCAN), como ya se mencionó en la sección 2.2.6.1, se diferencian respecto al manejo de mensajes y a su comportamiento frente a solicitudes remotas.

Una característica del principio BasicCAN (Figura 4.9) es que posee un buffer para almacenar los mensajes recibidos y otro para los mensajes a ser transmitidos. Antes de que un mensaje recibido pueda ser almacenado, debe pasar por un “Filtro de Aceptación”, adicionalmente el buffer receptor provee una “sombra de buffer” el cual permite la lectura de un mensaje recibido y la recepción de otro al mismo tiempo. Antes de recibir un tercer mensaje el Controlador de Host debe asegurarse que haya leído el primero.

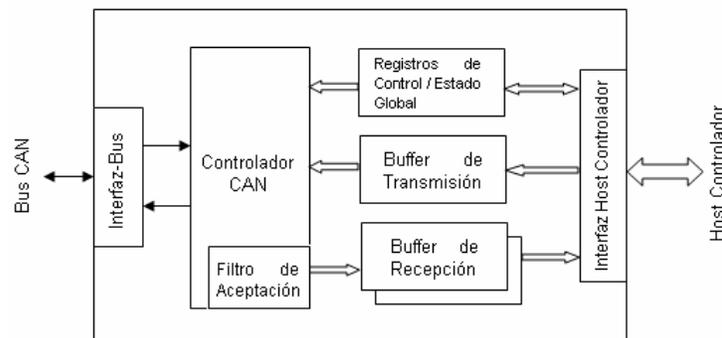


Figura 4.9: Diagrama de Bloques de un Controlador basado en el concepto BasicCAN

BasicCAN esta basado en una interfaz simple que pasa mensajes entre el controlador CAN y el microcontrolador por medio de sus buffers. Con este mecanismo simple las operaciones necesarias para recibir y transmitir mensajes son reducidas al mínimo.

Con la disponibilidad de controladores BasicCAN con buffers de recepción de capacidad extendida, el Controlador de Host puede operar en tiempo real¹.

En contraste al principio BasicCAN, un controlador CAN basado en el concepto FullCAN provee varios buffers de transmisión o recepción. Aquí, cada mensaje que debe ser transmitido o recibido tiene su propio buffer asignado. La configuración y asignación de buffers a los diferentes mensajes es realizada por el Controlador de Host en una etapa de configuración. Solo aquellos mensajes que pasen el filtro de aceptación serán recibidos y almacenados en el buffer² de recepción establecido (Figura 4.10).

Si un buffer receptor ya contiene un mensaje, y recibe otro del mismo identificador especificado al que debe almacenar, sobrescribe el antiguo con este último. Por tanto, los buffers siempre contienen información actualizada. De esta manera el Controlador de Host ya no es forzado a leer cada mensaje recibido, sino solamente aquellos que ocasionalmente necesite accediendo a un determinado buffer, liberándole de esta manera de una carga innecesaria. También el controlador CAN tiene acceso a los buffers todo el tiempo, incluso cuando un mensaje está siendo leído por el Controlador de Host. Adicionalmente, después de la lectura, el Controlador de Host debe revisar que haya leído datos consistentes y no partes de dos mensajes sucesivos (debido por ejemplo a un segundo mensaje que actualiza al primero).

¹ Para esta funcionalidad, el concepto de tiempo real se refiere a que un nodo destino pueda recibir una sucesión de mensajes dirigidos a él, sin que debido a limitaciones del buffer, rechace alguno de ellos.

² En Controladores FullCAN mejorados es posible que un buffer de recepción acepte más de un mensaje.

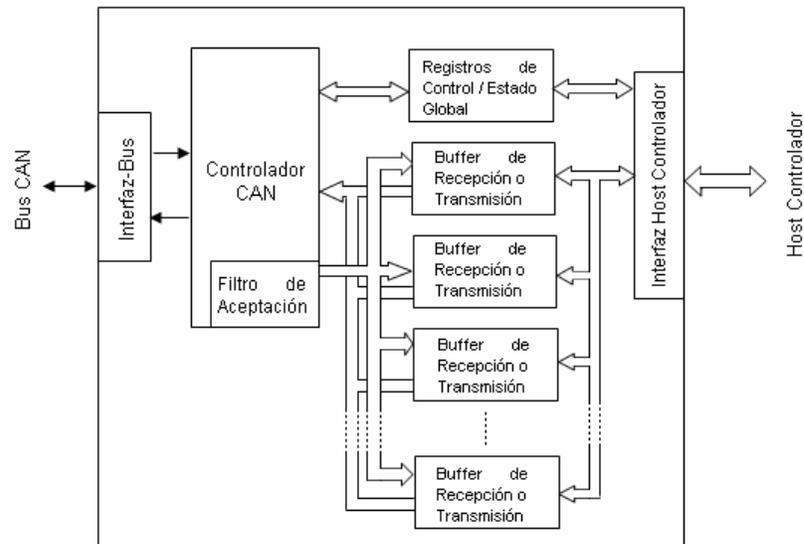


Figura 4.10: Diagrama de Bloques de un Controlador basado en el concepto FullCAN

Debido a los diferentes principios de almacenamiento de mensajes los conceptos BasicCAN y FullCAN también difieren respecto al manejo de mensajes remotamente solicitados. Ya que el almacenamiento de mensajes con un controlador BasicCAN ocurre en la memoria del Controlador de Host, un controlador de este tipo debe pasar una solicitud remota de mensaje a su Controlador de Host para que éste la responda escribiendo el mensaje deseado en el buffer de transmisión del controlador CAN.

A diferencia de BasicCAN y debido al almacenamiento de mensajes provisto en un controlador FullCAN es posible que éste responda inmediatamente a una solicitud remota sin ninguna intervención del proceso de aplicación del Controlador de Host.

El uso del principio FullCAN es necesario y conveniente en sistemas con altas velocidades de transmisión, bajo número de mensajes a ser recibidos por nodo y baja carga permitida sobre el Controlador de Host (Controlador de Host simple). Cabe señalar que este alivio de carga es posible solamente si el Controlador de Host está interesado en los mensajes más actualizados.

Si por otra parte el Controlador de Host debe recibir una elevada cantidad de mensajes¹ o soportar una alta carga, entonces una implementación menos costosa basada en el concepto de BasicCAN resulta ventajosa.

En este punto es importante recordar que los dos conceptos de implementación mencionados varían solamente con respecto al manejo del mensaje, mas no con respecto al procesamiento del protocolo, con esto es posible que diferentes implementaciones de controladores puedan ser operadas en la misma red.

Con chips CAN “stand alone” y microcontroladores con controlador CAN integrado, es posible la implementación de nodos CAN inteligentes. Por ejemplo, poderosos nodos CAN pueden ser implementados basados en simples microcontroladores de 8 bits tales como los de la familia 8051 ó 68HC11.

Por ejemplo, un nodo consistente de un microcontrolador 8051 con una frecuencia de reloj de 16 MHz y de un controlador externo BasicCAN es capaz de leer y almacenar mensajes CAN secuenciales a una velocidad de transmisión de 1Mbps. Sin embargo, con un promedio alto de carga del bus, el microcontrolador será cargado exclusivamente con estas tareas de comunicación y no será posible el pre-procesamiento de los mensajes CAN recibidos (por ejemplo filtrado de mensajes). Por lo tanto, para una velocidad de transmisión mayor a 500 KBits/s, el uso de un microcontrolador con controlador CAN integrado o de un microcontrolador más poderoso es necesario.

4.2.5 EJEMPLO DE IMPLEMENTACIÓN DE UNA RED DE CONTROL DISTRIBUIDO I/O

Una importante área de aplicación de los sistemas de bus de campo es la adquisición descentralizada de datos o el control de actuadores descentralizados llevado a cabo por una unidad de control superior.

Al respecto, y como ejemplo concreto, se presenta la implementación de un sistema distribuido I/O con nodos de red inteligentes (Ver Figura 4.11). El sistema I/O está conectado al Sistema Host a través de un controlador I/O central por

¹ Por ejemplo, una sucesión de mensajes del mismo identificador.

medio de una interfaz DPRAM. El sistema consistirá de hasta 32 nodos I/O distribuidos, cada uno con 16 entradas digitales, 16 salidas digitales y 8 entradas analógicas (cada una de éstas requerirá de 8 bits para su representación). Se requiere una extensión de red máxima de 100 metros. Para la transmisión de una entrada digital el tiempo de latencia debería ser menor a 5 milisegundos (para garantizar un comportamiento de tiempo real. Ver sección 4.2.3). Para cuestiones de carga del bus, se considera que el sistema transmite todos sus mensajes una vez cada 100 ms. La implementación del sistema estará basada directamente en el Protocolo CAN de 2 capas.

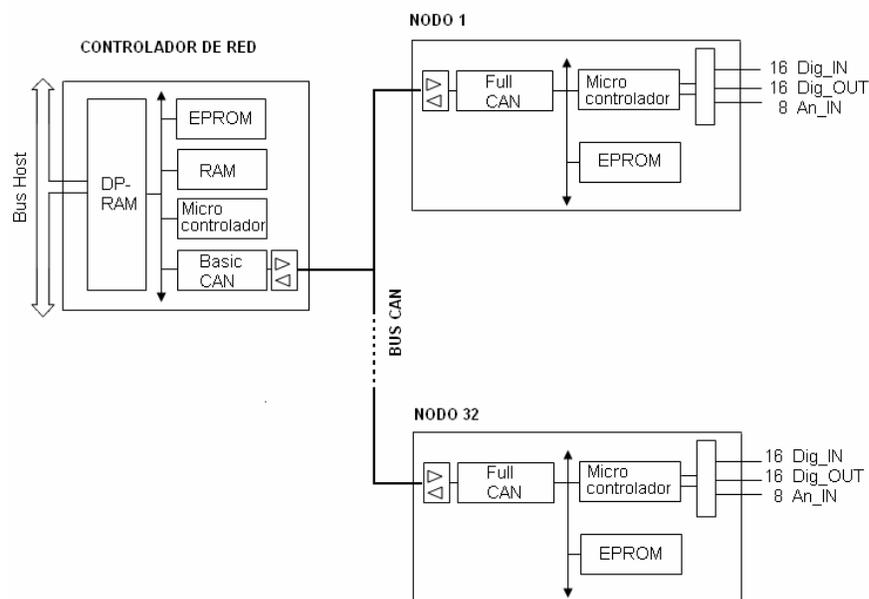


Figura 4.11: Implementación de un Sistema Distribuido I/O

Definición de mensajes y Asignación de Identificadores

Ya que la estructura considerada por el sistema corresponde a una estructura lógica maestro-esclavo, los mensajes solamente son intercambiados entre los nodos I/O y el controlador I/O central, mas no entre los nodos I/O. Para esto, los siguientes mensajes son requeridos:

- **“Solicitud de Estado de Nodo”**. Con este mensaje el Controlador I/O Central solicita a todos los nodos I/O enviarle información acerca de su estado¹. Este mensaje es transmitido en broadcast, sin ningún dato.

Transmisor: Controlador I/O, después de resetear el sistema

Receptor: Todos los nodos I/O

- **“Estado de Nodo”**. Este mensaje es enviado por un nodo I/O al controlador central I/O en respuesta al mensaje anteriormente citado. Cada nodo I/O requiere su propio mensaje CAN sin bytes de datos.

Transmisor: Nodo I/O después de recibir una “Solicitud de Estado de Nodo”

Receptor: Controlador I/O

- **“Entradas Digitales”**. Con este mensaje se comunica el estado de las 16 entradas digitales de un nodo I/O. Cada nodo I/O requiere su propio mensaje CAN. El valor de las 16 entradas digitales pueden ser transmitidas en dos bytes de datos.

Transmisor: Nodo I/O, después del cambio de una entrada digital

Receptor: Controlador I/O

- **“Salidas Digitales”**. Por medio del cual el controlador I/O establece las 16 salidas digitales de un nodo I/O. Para esto, se requiere un mensaje de 2 bytes de datos. Cada nodo I/O requiere su propio mensaje CAN.

Transmisor: Controlador I/O, después del cambio de una salida digital

Receptor: Nodo I/O

- **“Entradas Analógicas”**. Con el cual se transmiten 8 valores de entradas analógicas. Debido a que cada valor de entrada debe ser determinado con

¹ Por simplicidad, se asume que todos los nodos ya se encuentran en estado de comunicación antes de que el controlador central envíe este mensaje.

una exactitud de 8 bits, los 8 valores analógicos pueden ser transmitidos en los 8 bytes del campo de datos.

Transmisor: Nodo I/O, después de que un intervalo determinado de tiempo haya expirado o después de que una entrada analógica haya excedido cierto valor.

Receptor: Controlador I/O

Para la asignación de identificadores a los mensajes anteriormente detallados, se debe considerar que, grupos de mensajes del mismo tipo serán asignados al mismo grupo de prioridad. En este ejemplo las entradas digitales serán transmitidas con el valor más alto de prioridad.

La Figura 4.12 muestra la estructura del identificador de un mensaje definido para el ejemplo. Los 5 bits más significativos del identificador diferencian el tipo de mensaje, mientras que los 6 restantes representan el número del nodo fuente o destino. El número de nodo difícilmente tendrá alguna influencia sobre la prioridad del mensaje. Ésta es determinada primeramente por el tipo de mensaje con el identificador de mayor valor. Ver Tabla 4.5. Con 5 bits pueden ser representados 32 tipos de mensajes, mientras que con 6 bits es posible diferenciar entre 64 diferentes números de nodos.

Si un nodo I/O envía un mensaje al controlador I/O central (entrada digital), éste usa un identificador el cual lleva un "0" en el campo de tipo de mensaje y el número de nodo que lo identifica en el campo respectivo.

Si el controlador central I/O envía un mensaje a un nodo I/O (salida digital), su identificador llevará un "1" en el campo de tipo de mensaje y el número de nodo especificará al nodo I/O receptor; si el mensaje es dirigido a todos los nodos I/O el campo número de nodo llevará un "0".

Como ya se mencionó, la transmisión de una entrada digital por los nodos I/O y el establecimiento de una salida digital por el controlador I/O deben ocurrir con la prioridad más alta comparada a cualquier otra transmisión, por esta razón estos

dos tipos de mensajes poseerán los dos valores más altos de prioridad en su campo Tipo de Mensaje ("0", "1"). Mensajes tipo 2, son utilizados para transmitir valores de entradas analógicas del nodo I/O. Los mensajes de solicitud y envío de información del estado de un nodo tienen menos importancia que los mensajes de aplicación específica, y se les asignará las prioridades más bajas ("30", "31"). Si el controlador de red envía un mensaje de tipo "30" con un número de nodo "0", esto es entendido como una solicitud broadcast. Entonces, cada nodo I/O envía un mensaje tipo "31" con su respectivo número de nodo en su identificador.

Tipo de mensaje	Mensaje	Longitud de Datos (Bytes)
0	Entradas Digitales	2
1	Salidas Digitales	2
2	Entradas Analógicas	8
30	Solicitud de Estado de Nodo	0
31	Estado de Nodo	0

Tabla 4.5: Identificadores de Mensajes para un Sistema Distribuido I/O

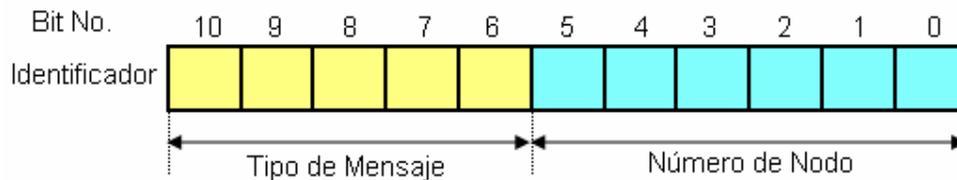


Figura 4.12: Estructura de un Identificador de Mensaje

Los tipos de mensajes restantes, están disponibles para extensiones futuras. Con la finalidad de que sea posible integrar nodos de otra funcionalidad en la red y usar los tipos de mensajes libres para los mensajes requeridos.

Como ejemplo, en la Tabla 4.6 se muestra la construcción de identificadores de los mensajes requeridos por un nodo I/O cuyo número es "1".

Grupo de Mensaje	Tipo de Mensaje	Identificador
Entradas Digitales	Envía	00000000001
Salidas Digitales	Recibe	00001000001
Entradas Analógicas	Envía	00010000001
Solicitud de Estado de Nodo	Recibe	11110000000
Estado de Nodo	Envía	11111000001

Tabla 4.6: Construcción de Identificadores de mensajes para el Nodo I/O con número de nodo: "1"

Determinación de la Velocidad de Transmisión requerida

Para este ejemplo se ha solicitado que el tiempo máximo de latencia para la transmisión de una entrada digital (mensaje de alta prioridad) sea menor a 5 ms. Este requerimiento significa que, en el peor caso que todas las entradas digitales deberán ser transmitidas en 5 ms.

Para la transmisión de una entrada digital de 16 bits, el número total de bits requeridos por mensaje (en el caso extremo) serán:

47 bits del protocolo + 16 bits de datos + 10 bits de relleno (peor de los casos)= 73 bits

Si los 32 nodos desean transmitir el estado de sus entradas digitales simultáneamente, se requeriría en total:

$$32 \text{ mensajes} * 73 \frac{\text{bits}}{\text{mensaje}} = 2336 \text{ tiempos de bit}$$

Para que esto sea posible dentro de 5 ms, el tiempo por bit no debe ser mayor a:

$$\frac{5 \text{ ms}}{2336 \text{ bits}} = 2,14 \mu \text{seg}$$

Es decir que el sistema debería operar a una velocidad de transmisión de:

$$\frac{1}{2,14 \mu \text{seg}} = 500 \text{ kbits / seg}$$

De acuerdo al la sección 2.2.1.5 del capítulo 2 con este valor de velocidad, la longitud máxima de la red puede ser 110metros, entonces el requerimiento de extensión de la red (100 metros) es cumplido.

Como se verá en el siguiente paso, la carga provocada por los mensajes de prioridad más alta sobre el medio de transmisión es baja y no exige la creación de ventanas adicionales de tiempo para la transmisión de mensajes de menor prioridad. Sin embargo, para que el comportamiento de tiempo real solicitado, (el mismo que es asegurado por medio de la velocidad de transmisión seleccionada) sea en realidad provisto por el sistema, es necesario configurar el sistema para prevenir que un nodo I/O envíe el estado de sus entradas digitales continuamente sin interrupción. De otra manera, bloquearía el bus a otros nodos I/O con un número de nodo más bajo (menor prioridad) que también desee transmitir el estado de sus entradas digitales.

Estimación de la Carga promedio del Bus

Para estimar la carga del bus, el sistema permite considerar que todas las 512 entradas digitales, 512 salidas digitales y 256 entradas analógicas son transmitidas una vez cada 100 ms.

Para transmitir esta información se requieren los siguientes tiempos de transmisión (no se han considerado bits de relleno):

- **Entradas Digitales**

Número total de bits= (Número de bits por Nodo I/O)*(Número de Nodos I/O)

Número total de bits= (47 + 16) * 32

Número total de bits= 2016

Tiempo de transmisión= (Número total de bits)*(1 Tiempo de Bit)

$$\text{Tiempo de transmisión} = 2016 * \left(\frac{1}{500 \cdot 10^3} \right)$$

Tiempo de transmisión= 4,032ms

- **Salidas Digitales**

Número total de bits= (Número de bits por Nodo I/O)*(Número de Nodos I/O)

Número total de bits= $(47 + 16) * 32$

Número total de bits= 2016

Tiempo de transmisión= (Número total de bits)*(1 Tiempo de Bit)

Tiempo de transmisión= $2016 * \left(\frac{1}{500 \cdot 10^3} \right)$

Tiempo de transmisión= 4,032ms

- **Canales Analógicos**

Número total de bits= (Número de bits por Nodo I/O)*(Número de Nodos I/O)

Número total de bits= $(47 + 64) * 32$

Número total de bits= 3552

Tiempo de transmisión= (Número total de bits)*(1 Tiempo de Bit)

Tiempo de transmisión= $3552 * \left(\frac{1}{500 \cdot 10^3} \right)$

Tiempo de transmisión= 7,104ms

En total, en el peor de los casos, en 100ms el bus sería ocupado aproximadamente 15ms , lo que corresponde a una carga promedio del 15%.

Implementación de Nodos I/O

Para la implementación de nodos I/O, es posible el uso de hardware muy económico. Como en la comunicación entre un nodo y el controlador de red solamente intervienen pocos mensajes CAN, una combinación de controlador FullCAN junto con un microcontrolador simple es suficiente para implementar el hardware de los nodos I/O. Un ejemplo de este tipo de microcontrolador es el Infineon C515C, el cual provee 8 entradas analógicas además de un controlador interno FullCAN, el mismo que soporta la velocidad de transmisión obtenida (500

kbit/s). Ya que solamente pocos objetos de recepción son requeridos, estos pueden ser configurados individualmente en los buffers del controlador CAN para que éste pueda realizar las funciones de filtrado de mensajes, y de esta manera el microcontrolador sea liberado de estas tareas.

El programa de aplicación para los nodos I/O también es fácil de implementar. Basado en una librería¹ que actúa como interfaz hacia la capa 2 del protocolo CAN, pocas son las líneas necesarias para implementar las funciones de comunicación. Los identificadores son asignados parcialmente en forma dinámica ya que éstos tienen una parte variable correspondiente al número de nodo. De tal forma que, el mismo programa puede ser instalado en cada nodo I/O.

Implementación del Controlador I/O Central

Debido a que el Controlador debe ser capaz de comunicarse con todos los nodos I/O, tendrá que recibir una gran cantidad de mensajes, razón por la cual un controlador BasicCAN debe ser usado. Para controlar el sistema de comunicación es suficiente un sistema microcontrolador de 8 bits (por ejemplo un 8051). Considerando el promedio relativamente bajo de carga del bus, el microcontrolador puede manejar la administración del DP-RAM, la generación y transmisión de solicitudes así como el procesamiento del buffer receptor BasicCAN.

¹ Colección de herramientas que proveen todas las funcionalidades requeridas para la operación y control de un controlador CAN. Estas funciones incluyen inicialización del nodo, envío, recepción y administración de mensajes CAN.

CAPITULO 5

CONCLUSIONES Y RECOMENDACIONES

- La principal ventaja que ofrecen los buses de campo, y la que los hace más atractivos a los usuarios finales, es la reducción de costos. El ahorro proviene fundamentalmente de tres fuentes: instalación, mantenimiento y ahorros derivados de la mejora del funcionamiento del sistema. Una de las principales características de los buses de campo es la significativa reducción del cableado necesario en una instalación. Cada componente sólo requiere un cable para la conexión de los diversos nodos. Se estima que puede ofrecer una reducción de 5 a 1 en los costos de cableado. En comparación con otros tipos de redes, dispone de herramientas de administración del bus que permiten la reducción del número de horas necesarias para la instalación y puesta en marcha.
- El esquema de arbitraje que usa el protocolo CAN es de bit inteligente no destructivo (CSMA/CD+AMP). Esto significa que en caso de un acceso múltiple, los bits componentes de cada mensaje se comparan entre sí, como resultado solamente el mensaje de prioridad más alta será transmitido (no se destruye), mientras que el resto de mensajes se detienen para ser retransmitidos posteriormente. Éste es un punto importante que ayuda a minimizar el tiempo de fuera de servicio del bus y aumentan al máximo el uso eficaz del ancho de banda disponible, además de que se le permite al sistema un manejo sencillo de las colisiones.
- La transmisión rápida y robusta en detección y confinamiento de fallas de CAN, le permite detectar y poner automáticamente fuera de servicio a los nodos defectuosos, evitando que alguno de ellos colapse la red. Esto garantiza que el ancho de banda siempre estará disponible para la transmisión

de mensajes críticos. Con todos estos beneficios implementados en el protocolo CAN y su extendida aceptación en el campo automotriz (sistemas móviles en general), otras áreas tales como: automatización y sistemas embebidos han iniciado la implementación de CAN en sus sistemas.

- CAN provee jerarquía multi-maestro, la cual permite la construcción de sistemas inteligentes y redundantes. Un nodo defectuoso en la red no puede impedir que la red siga operando sin problema. La transmisión orientada al mensaje garantiza integridad de datos ya que todos los dispositivos utilizan la misma información. Cabe añadir que sus sofisticados mecanismos de detección de errores y re-transmisión de mensajes errados también garantizan integridad de datos.
- Los sistemas de control distribuido que utilizan un bus CAN en general pueden modelarse en un modelo simplificado de tres capas: física, enlace y aplicación. La especificación de CAN publicada por Bosch se refiere solamente a las subcapas inferiores: física y enlace. Para la capa física existen varias alternativas en uso, siendo la más difundida la especificada en la norma ISO 11898-2. Otras redes basadas en un protocolo de capa superior proporcionan especificaciones de capa de aplicación sobre la base de CAN, lo cual provee al bus CAN de capacidad “plug and play”, de tal forma que al implementar un bus CAN, si la aplicación lo requiere se pueden añadir o intercambiar dispositivos sin necesidad de configuración adicional.
- CAL describe una capa aplicación (capa 7) independiente de la aplicación en sí. Ya que CAL no hace definiciones específicas a la aplicación, éstas deben ser hechas por un sistema específico de implementación. De esta manera CAL provee una plataforma de comunicación estandarizada para la implementación optimizada de aplicaciones distribuidas específicas. Además CAL mantiene un acceso directo a los servicios de capa 2, es decir que no es necesario introducir información de control adicional para la transmisión de datos de proceso, es decir que su característica de tiempo real no es afectada respecto a aquellas basadas en dos capas (enlace y física).

- Para la implementación de sistemas de automatización distribuida, además de una plataforma de comunicación estandarizada, son necesarias definiciones adicionales. Los objetivos de esta propuesta son la provisión de sistemas del mismo tipo de diseño y comportamiento con la consecuente reducción de esfuerzos necesarios de diseño, instalación y mantenimiento de dichos sistemas, así como la estandarización de funciones de dispositivos, condición básica para el uso de equipos de diferentes fabricantes, obteniendo ínter cambiabilidad de dispositivos e independencia de un solo fabricante. CANopen y DeviceNet constituyen dos de las más importantes soluciones al respecto.
- CANopen y DeviceNet representan dos sofisticados estándares de sistemas abiertos basados en CAN. Mientras que CANOpen, desarrollado por CiA, es aplicado principalmente en cualquier tipo de sistema embebido distribuido en Europa, DeviceNet que fue desarrollado prácticamente al mismo tiempo por OVDA, es empleado en EE.UU. y Asia en la automatización de piso de planta. Con estos dos sistemas de comunicación estandarizados se brindan soluciones, altamente avanzadas, muy confiables y eficientes en costo para la transmisión de datos en tiempo real en sistemas embebidos distribuidos y automatización de piso de planta.
- SAE J1939 es el protocolo serial estandarizado más importante en la comunicación de vehículos comerciales. También especifica una capa aplicación estándar la cual asegura interoperabilidad. Su empleo como protocolo estandarizado en el campo mencionado, entonces, permite que los fabricantes de componentes implementen un solo protocolo, mientras que los fabricantes de vehículos pueden emplear y operar componentes individuales de diferentes proveedores sin realizar modificaciones adicionales, asegurando así interoperabilidad entre estos componentes. SAE J1939 además, también es recomendado como plataforma para áreas específicas adicionales tales como automatización de buques.

- A pesar de las características de robustez y eficiencia en costo de una red de control CAN, ésta no es por naturaleza conveniente para las más severas aplicaciones de control de tiempo real. CAN es esencialmente un protocolo controlado por eventos en el cual la latencia de los mensajes no es constante y se incrementa con altos niveles de tráfico de mensajes sobre el bus. La solución fue el desarrollo de una nueva capa sesión que soporte un modelo controlado por tiempos “TTCAN”, el cual garantiza un comportamiento determinístico sobre el bus y usa de forma mucho más eficiente el ancho de banda físico de una red CAN. Además provee un tiempo global basado en un maestro de tiempo que corrige el tiempo local de cada nodo, con lo que satisface los requerimientos de sincronización de sistemas controlados por tiempo en aplicaciones distribuidas de tiempo real (por ejemplo automatización industrial).

- Respecto a la arquitectura de los controladores CAN, a primera vista el concepto FullCAN parecería ser el principio de implementación más poderoso debido a que su controlador de host no es sobrecargado con tareas de comunicación y a que se dispone de varios buffer para el almacenamiento de mensajes. A su vez, sin embargo, éste principio es limitado respecto al número máximo de mensajes a ser recibidos por un nodo (limitada cantidad de buffers). Por otro lado, ya que el principio BasicCAN solo implementa una parte de la funcionalidad FullCAN permite una implementación más económica, además no tiene límites en cuanto al número de mensajes a ser recibido. Por esto, ambos principios de implementación han sido integrados en controladores CAN de última generación.

- Antes de decidir si un sistema debe ser implementado basado en la propuesta de dos capas o en la de un protocolo de capa superior (protocolo de capa 7 o perfil estandarizado) hay que considerar los siguientes aspectos:
 - Una implementación basada en dos capas es conveniente para sistemas de control distribuido que requieren de un sistema que se ajuste exactamente a los requerimientos particulares de su aplicación (sistemas cerrados).

- Una implementación basada en una capa aplicación satisface los requerimientos de sistemas de control distribuido basados en una capa aplicación que requieren un entorno orientado a objetos e independiente de la aplicación.
- Una implementación basada en un perfil estandarizado provee comunicación abierta asegurando características como interoperabilidad e ínter cambiabilidad de dispositivos. Principalmente empleada en sistemas industriales y de automatización.
- Actualmente existen varias áreas del protocolo CAN, en las que se ha depositado particular interés y que representan potenciales campos de estudio y desarrollo, en especial aquellas referentes a la introducción de nuevas herramientas relacionadas al mejoramiento de los mecanismos frente a fallas y de la característica de tiempo real en redes de control basadas en CAN. Es por esto que se recomienda su estudio y desarrollo.
- Considerando el suficiente fundamento teórico provisto por el presente trabajo, se recomienda la realización de futuros trabajos relacionados a la implementación de sistemas de control basados en cualquiera de las 3 alternativas CAN aquí revisadas.

ANEXO A

REFERENCIAS

- CIA – 200 CiA DS 201 - 207, Version 1.1;
CAN Application Layer for Industrial Applications;
CAN-in-Automation; 1996
- CIA – 301 CiA DS 301 - Version 4.01;
CANopen Communication Profile for Industrial Systems;
CAN-in-Automation; 2000
- CIA – 302 CiA DS 302 – Version 3.0;
CANopen, Framework for Programmabel CANopen Devices;
CAN-in-Automation; 2000
- CIA – 303 CiA DSP 303, Version 0.21;
CANopen Connector Pin Assignment;
CAN-in-Automation; 1999
- CIA – 304 CiA WD 304 – Version 0.94;
CANopen, Framework for Safety-Relevant Communication;
CAN-in-Automation; 2000
- CIA – 401 CiA DSP 401 – Version 1.4;
CANopen Device Profile for I/O Modules;
CAN-in-Automation; 1996
- CIA – 402 CiA DSP 402 – Version 1.0;
CANopen Device Profile for Drives and Motion Control;
CAN-in-Automation; 1997
- CIA – 403 CiA DSP 403 – Version 1.0;
CANopen Device Profile for Human-Machine-Interfaces;

- CAN-in-Automation; 1998
- CIA – 404 CiA DSP 404 – Version 1.12;
CANopen Device Profile for Control and Measurement;
CAN-in-Automation; 1998
- CIA – 405 CiA DSP 405 – Version 1.0;
CANopen Device Profile for IEC 61131-3 Programmable Devices;
CAN-in-Automation; 1998
- CIA – 406 CiA DSP 406 – Version 2.0;
CANopen Device Profile for Encoders;
CAN-in-Automation; 1998
- DaUn92 Dais, S.; Unruh, J.;
Technisches Konzept des seriellen Bussystems CAN;
Teil 1: ATZ Automobiltechnische Zeitschrift 94 (1992) 2, S. 66-77;
Teil 2: ATZ 94 (1992) 3, S. 13-18
- DNISP98 DeviceNet Specifications, Volumes I and II;
Open DeviceNet Vendor Association Inc.; Revision 2.0; 1998
- ISO00-4 ISO – WD 11898-4;
Road vehicles – Controller Area Network –
Part 4: Time Triggered Communication; 2000
- ISO98-1 ISO – WD 16845;
Road vehicles – Controller Area Network (CAN) –
Conformance Test Plan; 1998
- ISO99-1 ISO – WD 11898-1;
Road vehicles – Controller Area Network (CAN) –
Part 1: Data link layer and physical signaling; 1999
- ISO99-2 ISO – WD 11898-2;

- Road vehicles – Controller Area Network (CAN) –
Part 2: High speed medium access Unit; 1999
- ISO99-3 ISO – WD 11898-3;
Road vehicles – Controller Area Network (CAN) –
Part 3: Low speed medium access unit; 1999
- J1939/81 Network Management Protocol;
Society of Automotive Engineers; 1996
- PHI-4 Philips;
P8xC591, Single-chip 8-bit microcontroller with CAN controller
DATA SHEET (Preliminary Specification);
Philips Semiconductors; Jul. 2000
- Rath97 Rathje, J.;
Kommunikation im Feld; In: Handbuch der Prozeautomatisierung;
Oldenbourg Verlag; München 1997
- SAE 00-1 SAE J2411 Single Wire CAN Network for Vehicle Applications;
February 2000;
- Stepper97 Stepper, M. R.;
J1939 Serial Vehicle Network Explanation and Tutorial;
SAE Society of Automotive Engineers; 1997
- UnMa90 Unruh, J.; Mathony, H. J.; Kaiser, K. H.:
Error Detection Analysis of Automotive Communication Protocols;
SAE International Congress, No. 900699; Detroit 1990

ANEXO B

MODELO JERARQUICO DE CAPAS EN AUTOMATIZACION DE PROCESOS

Un proceso de producción puede ser descrito en forma de un modelo jerárquico de acuerdo a la Figura B-1.



Figura B-1: Modelo de capas para automatización de la producción y variación de los requerimientos específicos de las capas

Mientras que las funciones estratégicas para administrar una compañía son ejecutadas en la “Capa de Control de Compañía”, el planeamiento y administración de una compañía son dirigidas por la capa inferior “Capa Control de Automatización”. Esta área por tanto también es designada como “Capa de Control de Fábrica o Compañía”. En la “Capa Control de Procesos” las órdenes de producción son transformadas en los correspondientes procesos técnicos y largas líneas de producción de una planta son controladas y monitoreadas a través de sistemas de control de esta capa.

Los dispositivos requeridos para la adquisición y control de procesos están localizados en la capa más baja del modelo jerárquico, la “capa de campo”. Esta incluye, además de unidades de control, interfaces/terminales hombre-máquina, sensores, transductores, etc.

Ya que los requerimientos de las distintas capas de automatización para el sistema de comunicación son bastante diferentes, es más apropiado proveer una combinación de varios subsistemas específicos correspondientes a los requerimientos de la capa específica que una sola red universal. La Figura B-2 ilustra la estructura general de tales sistemas. De acuerdo a los requerimientos de las diferentes capas de automatización diferentes sistemas de comunicación son usados.

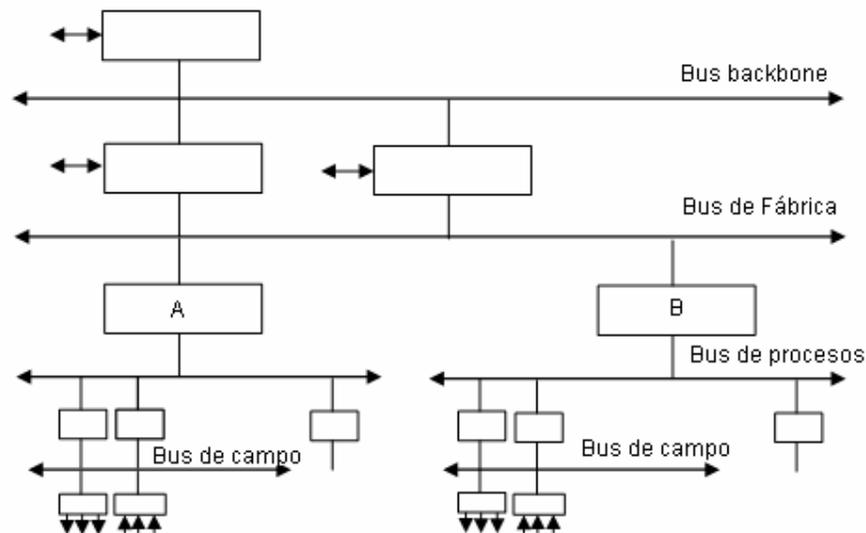


Figura B-2: Estructura jerárquica de red en sistemas de automatización de producción

La instrumentación técnica del dominio de campo de plantas de producción está caracterizado por el uso de PLCs y otras unidades de control de aplicación industrial además de un número de dispositivos periféricos de entrada y salida. Generalmente todos estos dispositivos están conectados a la respectiva unidad de control a través de sistemas de bus de campo, tales como CANOpen y DeviceNet.

ANEXO C

PROPIEDADES DE CAPA FÍSICA ESPECÍFICAS PARA PROTOCOLOS DE CAPA SUPERIOR

Una de las metas más importantes de los protocolos estandarizados de capa superior es la inter-operabilidad e inter-cambiabilidad de dispositivos. Por supuesto, esto presupone compatibilidad total de las propiedades físicas de la red y de los dispositivos. A parte de las características básicas de capa física previamente definidas en ISO 11898-1 y -2, también se definen las siguientes propiedades:

- Topología
- Tipo de línea
- Tecnología de Conexión
- Suministro de potencia
- Velocidad de Transmisión
- Unidad de acceso al medio bus

Las siguientes secciones presentan las propiedades de capa física específicas para CANopen y DeviceNet.

CANopen

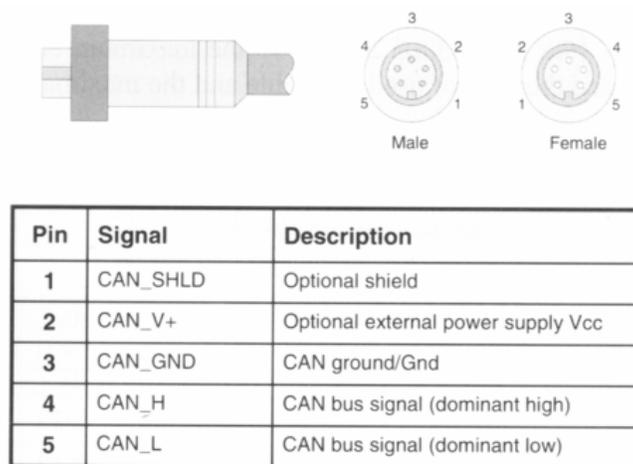
Con CANopen es posible implementar redes de hasta 127 nodos. El medio de transmisión CANopen especifica un bus de dos líneas, terminadas en ambos extremos con una resistencia igual a la impedancia específica de línea. Aislamiento galvánico de los nodos es opcional y es recomendado para longitudes de bus mayores a 200 m.

Todos los dispositivos CANopen deben soportar una velocidad de transmisión de 20 kBit/s. Por ello, CANopen recomienda el uso de velocidades de transmisión y longitudes de línea en conformidad con CiA DS 102.

Una variedad de conectores es sugerida en [CiA DS-303]:

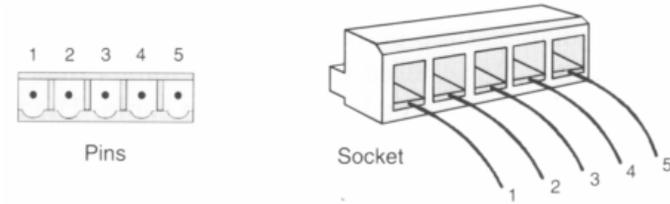
- Conector de 9 pines D-SUB de acuerdo a CiA DS 102
- Conector estilo mini de 5 pines
- Conector estilo abierto
- Conector multi-polo
- Conector estilo micro
- Conector RJ10/RJ45
- Conector redondo DIN 7 pines/8 pines

La Figura C-1 muestra un conector estilo mini de 5 pines y su asignación de pines, la Figura C-2 un conector de estilo abierto y su respectiva asignación de pines.



5-pin mini-style connector according to CANopen

Figura C-1: Conector de 5 pines CANOpen



Pin	Signal	Description
1	CAN_GND	CAN ground/ Gnd
2	CAN_L	CAN Bus Signal (dominant low)
3	CAN_SHLD	Optional shield
4	CAN_H	CAN Bus Signal (dominant high)
5	CAN_V+	Optionale external power supply Vcc

Open-style connector according to CANopen

Figura C-2: Conector "Open-Style" CANOpen

DeviceNet

Como en el caso de CANopen, la especificación de capa física de DeviceNet representa una extensión del estándar ISO11898. DeviceNet especifica redes de hasta 64 nodos con topología de bus. Las líneas del bus son terminadas en ambos extremos con una impedancia de $RT=121 \text{ Ohms } \pm 1\%$. La Figura C-3 muestra la máxima longitud permitida para las "líneas drop" y la máxima extensión del bus dependiendo de la velocidad de transmisión y del tipo de cable usado en el bus.

Data Rate	Trunk Line Length			Cumulative Drop Line Length	Maximum Single Drop Line Length
	Thick Cable	Thin Cable	Flat Cable		
125 kb	500 m (1640')	100 m (328')	420 m (1378')	156 m (512')	6 m (20')
250 kb	250 m (820')	100 m (328')	200 m (656')	78 m (256')	6 m (20')
500 kb	100 m (328')	100 m (328')	100 m (328')	39 m (128')	6 m (20')

Figura C-3: Máxima extensión del bus

Devicenet especifica un par de alambres uno para señal de transmisión y otro para potencia de alimentación. Tres tipos de cables de tamaño variable de

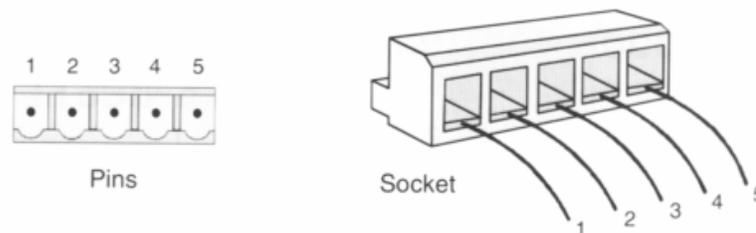
diámetro son permitidos, “cable grueso” (thick cable), “cable delgado” (thin cable) y “cable plano” (flat cable). Los cables grueso y plano son usados para tamaños de red mayores a 100 m. “Drop lines” y redes de tamaño más pequeño pueden ser realizadas con cable delgado. Es obligatorio el uso de código de colores en alambres individuales.

De acuerdo a la especificación DeviceNet, sus dispositivos soportan hasta tres velocidades de transmisión: 125 kBit/s, 250 kBit/s, 500kBit/s¹. La conexión al bus así como los dispositivos electrónicos pueden ser alimentados por la fuente de alimentación (nominal de 24V).

Las Figuras C-4, C-5 y C-6 especifican tres tipos de conectores:

- Estilo Abierto
- Estilo Mini
- Estilo Micro

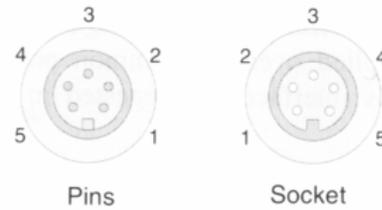
Adicionalmente, DeviceNet especifica múltiples conectores y “taps” requeridos para instalación así como posibilidades de conexión para un conector especial para propósitos de diagnóstico.



Pin	Signal	Description
1	V-	GND (black)
2	CAN_L	CAN bus signal (dominant low) (blue)
3	Drain	Shield
4	CAN_H	CAN bus signal (dominant high) (white)
5	V+	external power supply Vcc (red)

Open-style connector according to DeviceNet

¹ Solamente la velocidad de 125 kBit/s es obligatoria, las otras dos son opcionales. Comúnmente los dispositivos soportan las tres.

Figura C-4: Conector “open style” DeviceNet

Pin	Signal	Description
1	Drain	Shield
2	V+	External power supply Vcc (red)
3	V-	GND (black)
4	CAN_H	CAN bus signal (dominant high) (white)
5	CAN_L	CAN bus signal (dominant low) (blue)

Mini-style connector according to DeviceNet

Figura C-5: Conector “mini style” DeviceNet

Pin	Signal	Description
1	Drain	Shield
2	V+	External power supply Vcc (red)
3	V-	GND (black)
4	CAN_H	CAN bus signal (dominant high) (white)
5	CAN_L	CAN bus signal (dominant low) (blue)

Micro-style connector according to DeviceNet

Figura C-6: Conector “micro style” DeviceNet

Conexión del bus y dispositivo debe ser protegida contra “miswiring”. Posibles medidas para proteger las conexiones son sugeridas en la especificación DeviceNet. La conexión al bus opcionalmente puede tener aislamiento galvánico.

ANEXO D

MEDIDAS PARA MEJORAR LA COMPATIBILIDAD ELECTROMAGNETICA (EMC, Electromagnetic Compatibility)

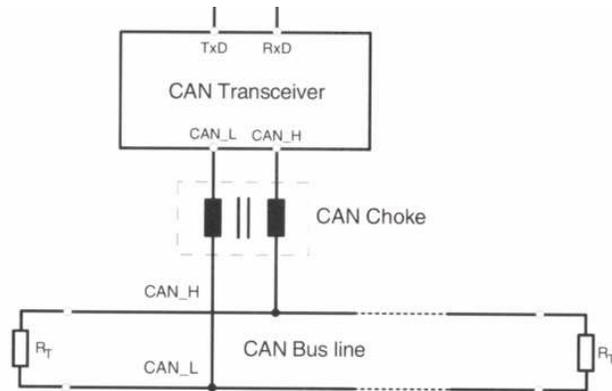
Aspectos de compatibilidad electromagnética de la capa física son de particular interés cuando se usa alambres de bus no blindados. Posibles medidas para mejoramiento de la compatibilidad electromagnética de redes CAN pueden ser divididas en dos grupos:

- Inmunidad contra interferencia electromagnética inducida (protección de “immision”)
- Reducción de la potencia electromagnética emitida (protección de “emission”)

La inmunidad EMC, principalmente se dirige a la capacidad del receptor de detectar apropiadamente los niveles de la señal diferencial bajo condiciones de ruido.

Ciertamente uno de los mejores métodos para mejorar la protección de inmision y emission es el uso de líneas de bus trenzado y blindado. Estos ofrecen un grado muy alto de protección y son independientes de parámetros de aplicación tales como velocidad de transmisión y número de nodos. Medidas adicionales, comunes, para el mejoramiento del EMC con respecto a inmision son:

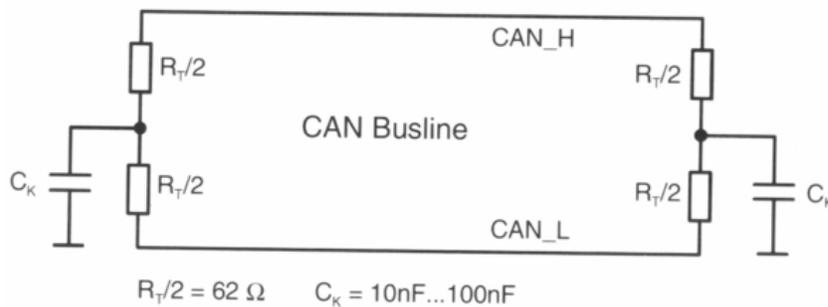
- Incremento de la resistencia contra perturbaciones de modo común por medio de atenuación en componentes en el interfaz del bus (Ver Figura D-1).



Suppression of inductively induced common mode disturbances by means of a CAN choke

Figura D-1: Supresión de perturbaciones inducidas

- Desviando las perturbaciones de alta frecuencia por medio de una terminación de bus dividida (split bus terminations) (Ver Figura D-2).



Improvement of the EMC characteristic by splitted bus termination

Figura D-2: Terminación de bus dividida

ANEXO E

CARACTERISTICAS DE LOS TRANSCEIVERS CAN DE ACUERDO A ESTANDARES DE CAPA FISICA

En la Tabla E-1 se resumen los parámetros más importantes de algunos de los transceivers CAN de alta velocidad disponibles conforme a ISO 11898-2. La Tabla E-2 muestra un vistazo general de transceivers disponibles correspondientes a los estándares de baja velocidad ISO 11898-3, ISO 11992 y SAE J2411.

High Speed CAN transceiver according to ISO 11898-2 (High-speed standard)

Manufacturer	Infineon [INF-6]	Philips [PHI-7]	Philips [PHI-8]	Philips [PHI-9]	Temic (Silico- nix)	Unitrode [UNI-1]
Type	TLE 6250	80C250	80C251	TJA 1050	SI9200EY	UC5350
Max. Data Rate [Mbit/s]	1	1	1	1	1	1
Thermal Shutdown	yes	yes	yes	yes	yes	yes
Slope Control	no	variable	Variable	no	no	variable
Common Mode Range [V]	-2 ... +7	-7 ... +12	-7 ... +12	-7 ... +12	-2 ... +7	-25 ... +18
Delay [ns]	150	170	170	150	120	100
Fan Out	n.a.	110	110	110	32	n.a.
Current Consumption Dominant State [mA]	70	70	80	75	70	70
Current Consumption Standby [uA]	10	170	250	n.a.	n.a.	1000
Packaging	P-DSO-8-3	SO-8	SO-8	SO-8	SO-8	SOIC-8

Tabla E-1: Características de transceivers de alta velocidad

Low Speed CAN transceivers according to the standards ISO 11898-3, SAEJ2411 and ISO 11992

Manufacturer	Infineon [INF-7]	Infineon [INF-8]	Motorola [MOT-6]	Philips [PHI-10]	Philips [PHI-11]	Temic (Siliconix) [TEM-1]
Type	TLE 6254 G	TLE6255 G	MC 33388	82C252	TJA 1054	B10011S
Specification	ISO 11898-3 Low Speed	SAE J2411 Single Wire	ISO 11898-3 Low Speed	ISO 11898-3 Low Speed	ISO 11898-3 Low Speed	ISO 11992
Max. Data Rate [Kbit/s]	125	100	125	125	125	250
Bus Failure Management	automatic	-	automatic	automatic	automatic	automatic detection, mode switch by host
Protection	(1),(2),(3)	(1),(2)	n.a.	(1),(2),(3)	(1),(2),(3),(4)	n.a.
Fan Out	n.a.	n.a.	32	15	32	2
Current Consumption Dominant State [mA]	10 (5)	9	5 (5)	35	27	26
Current Consumption Standby [uA]	55	40	30	75	50	-
Packaging	SO-14	P-DSO-14	SO-14	SO-14	SO-14	SO-16

- (1) short circuit proof in 12V powered systems
- (2) thermal protection
- (3) bus lines protected against transients in an automotive environment
- (4) permanent dominant monitoring of TxD
- (5) no load
- n.a. data not available

Tabla E-2: Características de transceivers de baja velocidad