
GUÍA DE ESTUDIO TEMA 15.

CAN

OBJETIVOS

- Ofrecer un conocimiento profundo sobre el funcionamiento básico de CAN: trama, acceso al medio, detección de errores, gestión de fallos y las características de la capa física.
- Conocer las características más importantes del bus DeviceNet.

ORIGEN HISTÓRICO Y EVOLUCIÓN

- Origen (80's): fabricantes de automóviles que buscaban simplificar el conexionado del número creciente de unidades electrónicas de los automóviles.
- Es utilizado en el automóvil como bus de comunicaciones de lo que se denomina sistemas distribuidos embebidos.
- En un automóvil pueden existir varias redes CAN:
 - La red CAN de alta velocidad enlaza unidades de control en tiempo real como las de inyección de combustible, frenado ABS o unidad de encendido en motores de encendido provocado.
 - La red CAN de baja velocidad enlaza dispositivos electrónicos tales como los elevalunas, conjuntos de luces, sistema de climatización, etc.
 - Red CAN de 125 kbit/s con transceptores tolerantes a fallo para unir las unidades controladoras de los *airbags*.
- Características buscadas: robustez, facilidad de uso y alto grado de capacidad de tiempo real.
- Extensión a otros ámbitos industriales, automatización naval, edificios inteligentes, ferrocarriles, aviación, medicina, etc.

BUS DE COMUNICACIONES CAN

- Protocolo de comunicaciones serie asíncrono y multimaestro para conectar unidades de control electrónicas, sensores y actuadores en aplicaciones de automoción e industriales.
- La especificación de Bosch (ISO-11898-1), define del modelo ISO/OSI la capa de enlace de datos y algunos aspectos de la capa física independientes del medio de transmisión, por lo que posteriormente se completó la capa física en distintas especificaciones.
- Red de libre difusión (broadcast) \Rightarrow cualquier nodo puede escuchar las tramas enviadas por la estación transmisora.
- En el medio físico se definen dos estados complementarios: dominante y recesivo.
 - Durante la transmisión simultánea de bits dominantes y recesivos el valor resultante en el medio de transmisión sea dominante.
- Acceso al medio:
 - Permite el acceso simultáneo al bus desde diferentes nodos (capacidad multimaestro)
 - Colisiones: CSMA/CD + AMP (*Carrier Sense Multiple Access with Collision Detection and Arbitration on Message Priority*): acceso múltiple con detección de portadora, detección de colisión más resolución de colisión con arbitraje en la prioridad del mensaje. Método no destructivo con arbitraje bit a bit.
- Versiones:
 - 2.0A: formato estándar de mensaje con identificador de 11 bits
 - 2.0B: formato de mensaje extendido con identificador de 28 bits.
- Controladores:
 - 2.0B pasivo: recibe y transmite mensajes de la versión 2.0A y admite mensajes 2.0B pero no los almacena.
 - 2.0B activo: además almacena y recibe mensajes en formato extendido.

NODO CAN

- **Nodo CAN** = unidad de control electrónica que contiene un microprocesador o microcontrolador con el correspondiente programa de aplicación y el software para las capas altas del protocolo, un controlador CAN como mínimo 2.0B pasivo y un transceptor CAN de alta velocidad sobre par de hilos con retorno.
- Implementación: cada uno en un circuito integrado por separado, o bien integrando el controlador CAN y el microcontrolador en el mismo encapsulado.
 - Las ventajas e inconvenientes de estas dos opciones incluyen el precio de los componentes, el espacio físico ocupado, la fiabilidad del conjunto, la carga que para la CPU supone el acceso al controlador CAN y la reusabilidad del software.
- El software para las capas de la 3 a la 7 de ISO/OSI puede ajustarse a la correspondiente especificación de la capa de aplicación de un bus de campo como DeviceNet o CANOpen o ser una solución propietaria.

CAPA de ENLACE de DATOS.

- Dos subcapas: LLC (control de enlace lógico) y MAC (control de acceso al medio).
- Utiliza mensajes de estructura definida (tramas).
- MAC es responsable del formato y codificación de las tramas. También de la detección y gestión de errores.
- **Tipos de tramas:**
 - **Trama de datos** (*Data Frame*), por la que el nodo transmisor envía datos a los receptores.
 - **Trama remota** (*Remote Frame*), por que la un nodo solicita la transmisión de una trama de datos con el mismo identificador.
 - **Trama de error** (*Error Frame*), que se transmite por un nodo cuando detecta un error en el bus
 - **Trama de sobrecarga** (*Overload Frame*), que transmite un nodo cuando el nodo requiere un retardo extra antes de recibir la próxima trama de datos/remota.

CAPA de ENLACE de DATOS.

■ Formato de la trama de datos y remota (versión 2.0A):

- ❑ Un nodo CAN ha de esperar a que el bus esté en reposo (nivel recesivo), para comenzar su transmisión.
- ❑ Cuando un nodo CAN desea transmitir datos, transmite una trama de datos:
 - SOF: bit de inicio dominante. Sincronización
 - Bits de campo de arbitraje: prioridad (11 bits)
 - Bit RTR (Remote Transmission Request): dominante en trama de datos y recesivo en remota.
 - ❑ Un nodo receptor (consumidor) de una trama de datos puede iniciar la transmisión de dicha trama por el nodo transmisor (productor) del mismo, enviando una trama denominada remota.
 - ❑ La trama remota tiene el mismo identificador que la trama de datos, pero su correspondiente bit RTR es recesivo y además no tiene campo de datos.
 - ❑ El correspondiente nodo productor enviará como respuesta a la trama remota la trama de datos
 - Campo de control: nº de octetos de datos (6 bits)
 - ❑ El bit IDE:
 - si dominante ⇒ trama de datos estándar (identificador de 11 bits)
 - si recesivo ⇒ trama en formato extendido (identificador de 28 bits).
 - ❑ El bit RBO (*reserved bit*) es siempre recesivo.
 - ❑ Los cuatro bits de código de longitud (DLC *Data Length Code*) indican en binario el número de octetos de datos en el mensaje (0 a 8). Cada byte de datos se transmite con el bit más significativo primero.
 - Campo de datos (0 a 64 bits)
 - CRC: 15 bits para detectar errores
 - Campo de reconocimiento: 2 bits
 - Bits de fin de trama: 7 bits recesivos.

CAPA de ENLACE de DATOS.

■ Codificación de la trama

- ❑ CAN es asíncrono \Rightarrow los controladores CAN tienen que recibir una señal de reloj local de frecuencia más elevada que la de la transmisión en el bus y sincronizan este reloj cuando en la línea de bus aparecen flancos de transición entre estados.
- ❑ La secuencia de bits en un mensaje CAN se codifica de acuerdo con el método NRZ (*Non-Return-to-Zero*).
- ❑ Para asegurar la sincronización de todos los nodos del bus, se utiliza la técnica de los **bits de relleno** (*bit stuffing*):
 - Durante la transmisión de un mensaje CAN como máximo 5 bits consecutivos pueden tener la misma polaridad, introduciendo el transmisor un bit de relleno de polaridad contraria cuando se dé este hecho.

■ Detección y gestión de errores

- ❑ **Error de bit:** Cualquier bit que se reciba con polaridad inversa a la que se ha transmitido se considera error de bit, excepto cuando se recibe durante el campo de arbitraje o en el bit de reconocimiento.
- ❑ **Error de relleno:** 6 bits consecutivos del mismo signo
- ❑ **Error de CRC:** el CRC calculado por el receptor no coincide con el recibido en la trama.
- ❑ **Error de forma:** cuando un campo de formato fijo se recibe alterado en algún bit
- ❑ **Error de reconocimiento:** cuando ningún nodo receptor transmite un bit dominante durante el bit de reconocimiento.
- ❑ Cuando cualquier nodo detecta un error en la transmisión de un mensaje, transmite la trama de error (transmisión de 6 bits dominantes sucesivos)
 - El nodo transmisor cuyo mensaje ha sido abortado, reintentará transmitir este mensaje automáticamente cuando las condiciones de bus se lo permitan.
 - Para un transmisor el mensaje es válido si no se detecta ningún error hasta Fin de trama.
 - Para un receptor un mensaje es válido cuando no detecta ningún error hasta el antepenúltimo bit del campo de Fin de trama.
- ❑ El resultado es una consistencia total de la información en el sistema distribuido: o todos los nodos reciben la misma información al mismo tiempo o no reciben información válida.

CAPA de ENLACE de DATOS.

■ Aislamiento de nodos con fallo

- ❑ El estado de un nodo respecto al bus puede ser:
 - Activo: transmite y recibe mensajes
 - Pasivo: para transmitir necesita esperar una secuencia adicional de bits recesivos y no puede señalar errores con trama de error activa.
 - Anulado: tiene que deshabilitar su transceptor y no participa en la comunicación
- ❑ Cuando un nodo acumula errores pasa del activo a pasivo y si la degradación continua el nodo pasa al estado anulado donde se autoexcluye de la comunicación evitando perturbar al resto de los nodos de la red.

■ Filtros y máscaras en controladores CAN

- ❑ Una vez recibida la trama de datos es tarea del controlador CAN de cada nodo decidir si el mensaje ha de ser aceptado para su almacenamiento en su buffer de recepción y su posterior traspaso a la CPU del nodo, labor que tiene encomendada el filtro de aceptación del controlador CAN.
- ❑ El filtro de aceptación está incluido en la subcapa de control de enlace lógico (LLC, *Logic Link Control*) del protocolo CAN.
- ❑ Todas las implementaciones de CAN tienen al menos un filtro de aceptación.
- ❑ Los distintos controladores CAN se diferencian en el número y tamaño de los filtros de aceptación de mensajes y filtros de máscara, y en el tamaño y número de los buffers de recepción y transmisión.
- ❑ Un controlador CAN se dice que es "FullCAN" cuando tiene varios buffer de recepción de mensajes y cada uno de ellos tiene un filtro de aceptación asociado, de modo que a cada buffer de recepción se le puede asignar un mensaje con un identificador específico.
- ❑ Un controlador se dice que es "BasicCAN" cuando solo dispone de uno o dos buffers de recepción donde se almacenan todos los mensajes una vez filtrados.

CAPA de ENLACE de DATOS.

■ Características adicionales en controladores CAN

- ❑ Contadores de error: diagnóstico de fallos
- ❑ Estado “sleep mode”: reduce consumo cuando el bus está inactivo.
- ❑ La posibilidad de activar/ desactivar la retransmisión automática de mensajes: CAN exige que un controlador CAN intente retransmitir un mensaje hasta su correcta transmisión.
 - En el modo de transmisión única (*single shot transmisión*) especificado en ISO 11898 en la opción de tramas planificadas, esta retransmisión automática es anulada.
 - La pérdida de arbitrio y el error de transmisión se comunican por el controlador CAN al microcontrolador mediante interrupción.

CAPA FÍSICA

- Tiene que soportar la representación de los estados recesivo y dominante.
- Debe ser capaz de transmitir y recibir señales al mismo tiempo.
- Utiliza transmisión diferencial (ISO11898-2) \Rightarrow es robusta ante EMI ya que ambas líneas se ven afectadas por las interferencias electromagnéticas.
- Dividida en tres subcapas:
 - **PLS** (*physical signalling*):
 - Está implementada en los chips controladores CAN.
 - Trata de las tareas de encodificación y codificación, sincronización y temporización a nivel de bit.
 - **PMA** (*physical médium attachment*):
 - Describe las características del transceptor
 - Realiza la transformación de niveles lógicos a los requeridos por el medio físico sobre el que se establece la transmisión
 - **MDI** (*medium dependent interface*) especifica las características de los cables y conectores a utilizar:
 - ISO 11898-2 define una línea de dos cables con retorno común terminada en ambos extremos por resistencias de valor igual a la impedancia característica de la línea.
 - Los cables del bus se pueden rutar en paralelo, trenzados y/o apantallados, dependiendo de la robustez EMC que se requiera.
 - La topología del cableado ha de ser tan próxima como sea posible a una estructura de línea simple, sin derivaciones con el fin de evitar reflexiones.
 - En el caso de existir derivaciones para conectar los nodos, recomienda longitudes lo más cortas posibles para estas derivaciones (por ejemplo a 1Mb/s la longitud de las derivaciones no debe exceder de 0,3 m).
 - Todas las masas de los transceptores se interconectarán.
 - Longitud máxima recomendada para el bus sin repetidores de 1 km.

CAPA FÍSICA

- **Características adicionales que incluyen los transceptores:**
 - Protección contra cortocircuito a masa y a positivo de batería (normalmente 12 V, aunque existen transceptores para los sistemas de 24 V de camiones y autobuses) de las líneas CAN_H y CAN_L.
 - Protección térmica
 - Control de rampa en las conmutaciones de nivel dominante-recesivo. Permite reducir la energía electromagnética radiada de acuerdo con las normas aplicables.
 - “Fan out” (capacidad del transmisor para actuar sobre un número dado de entradas receptoras) habitual de 32 nodos, aunque existen transceptores que amplían éste hasta 110 nodos.
 - Alta inmunidad contra las interferencias electromagnéticas y protección de las líneas de bus contra transitorios, habituales ambos en el entorno del automóvil.
 - Modo de funcionamiento en bajo consumo (standby mode) ante falta de actividad en el bus.
 - Tensiones de modo común de rango extendido (se extiende fuera del rango de tensiones de alimentación).

BUS de CAMPO DeviceNET

■ CAPA FÍSICA

- El cable de red está formado por un par trenzado apantallado para el bus CAN.
- Utiliza los transceptores de alta velocidad según ISO-11898-2 y otro par trenzado de hasta 8 A de capacidad para la alimentación a 24 V y masa, para alimentar a sensores y actuadores de bajo consumo (células fotoeléctricas, arrancadores de motor, etc.).
- Los dispositivos con alimentación propia (autómatas programables, variadores de velocidad, etc.) han de conectarse al bus CAN a través de optoacopladores que se alimentan de las líneas de 24 V y masa del cable de red.
- Los nodos pueden ser retirados y conectados sin quitar la alimentación

■ CAPA de APLICACIÓN

- Define como se asignan los identificadores CAN (controlando de este modo las prioridades en el proceso de arbitraje) y como el campo de datos de la trama CAN de datos se utiliza para especificar servicios, datos y determinar su significado.
- Cada nodo DeviceNET tiene una dirección de nodo única.
- Dos tipos de mensajes:
 - Mensajes Explícitos: proveen comunicaciones de propósito múltiple en modelo cliente/servidor.
 - Mensajes I/O: transmitir datos de control en tiempo real.

BUS de CAMPO DeviceNET

- Mensajes Explícitos:
 - Desde un nodo maestro o un nodo de configuración se realicen funciones tales como configurar nodos, obtener/programar parámetros de los nodos, realizar diagnóstico de problemas, etc.
 - Contienen datos que no son urgentes.
 - El significado específico del mensaje está en el campo de datos.
 - Si la longitud de datos > 8 bytes \Rightarrow fragmentan.
- Mensajes I/O:
 - Utilizan identificadores de alta prioridad.
 - Dirigidos a uno o a varios destinatarios.
 - Modelo productor-consumidor:
 - Cualquier nodo de la red puede escuchar el mensaje del nodo transmisor que produce el mensaje (nodo productor). Después de recibir el mensaje, es tarea de cada nodo decidir si el mensaje tiene que ser aceptado o no, en función de los 11 bits del campo identificador de la trama CAN de datos recibida, tarea que se encomienda al filtro(s).
 - El modelo requiere que el campo identificador determine el contenido y significado del mensaje del nodo productor y que los nodos consumidores conozcan a priori los identificadores que les interesan.
 - También se fragmentan cuando los datos > 8 bytes.
 - Tipos de mensajes I/O:
 - Cambio de estad
 - Cíclicos
 - Sincronización
 - Interrogación

BUS de CAMPO DeviceNET

■ Modelo maestro/esclavo

- Muchos sensores y actuadores que están diseñados para ejecutar una función predeterminada con tipo y cantidad de datos que producirán y/o consumirán conocidos.
- Típicamente estos dispositivos suministran datos de sus entradas a un único nodo o requieren datos de dicho nodo para sus salidas y configuración ≡ **maestro-esclavo**.
- DeviceNET define el Conjunto Predefinido de conexiones maestro/esclavo para simplificar el movimiento de datos I/O y de datos de configuración.
- Al maestro le pertenecen los esclavos de su lista.
- Un esclavo no puede iniciar comunicación sin permiso del maestro.
- Un esclavo sólo pertenece a un maestro.
- Cada esclavo implementa un mensaje explícito para que su maestro pueda leer y escribir atributos en los esclavos.
- Cada esclavo puede transmitir tres tipos de mensajes I/O: cíclico/cambio de estado, de respuesta de sincronización de bit y de respuesta de interrogación y tiene que implementar al menos un tipo de estos mensajes I/O.
- El esclavo sólo se comporta como servidor de las peticiones del maestro.
- Si el maestro recibe un mensaje de cambio de estado/cíclico desde un esclavo ⇒ envía un mensaje de reconocimiento al nodo esclavo, con la estructura de identificador.
- El maestro transmite un mensaje I/O poll command a cada uno de los esclavos que tiene que interrogar.
 - Cada esclavo interrogado transmite de vuelta un mensaje de respuesta.
- Los mensajes de orden de sincronización de bit (*bit-strobe command message*) y mensaje de respuesta se utilizan para mover pequeñas cantidades de datos de entrada/salida entre el maestro y los esclavos (sin fragmentación).
 - Los esclavos reaccionan a este mensaje ignorándolo (no tenerlo implementado) o utilizar/no utilizar el mensaje como señal de disparo.
 - Tampoco se permite fragmentación en la respuesta de los esclavos.

BUS de CAMPO DeviceNET

■ Los perfiles de dispositivo

- Los perfiles de dispositivo (*Device Profiles*) definen para dispositivos estándar de plantas industriales automatizadas su comportamiento como nodo en una red DeviceNet.
- Se han definido perfiles tanto para dispositivos sencillos tales como células fotoeléctricas, pulsadores, etc. como más complejos como variadores, controladores de posición, etc.
- Este comportamiento como nodo DeviceNet incluye, para cada tipo de dispositivo, los mensajes I/O que ha de tener, especificando para cada uno de ellos su formato y significado de datos, los parámetros configurables y como se accede/modifican los mismos, etc.
- Todos los fabricantes de un determinado tipo de dispositivo han de cumplir con el perfil DeviceNet de dicho dispositivo, por lo que son equivalentes y por tanto intercambiables.
- Los fabricantes pueden añadir funciones adicionales.

■ La hoja electrónica de datos

- Fichero ASCII con formato estándar que está incluido en la documentación proporcionada por el fabricante del dispositivo.
- La EDS contiene la descripción de los atributos del dispositivo (nombre, unidades de ingeniería utilizadas, tipos de datos, etc), los atributos públicos especificados en el perfil del dispositivo y los atributos específicos del fabricante del dispositivo.
- Las EDS de los distintos dispositivos que conforman la red DeviceNet se suministra a la herramienta de configuración (normalmente un computador tipo PC o similar) que se utiliza para configurar la red.
- La información contenida en la EDS, también puede estar almacenada en la memoria no volátil del nodo DeviceNet (*Parameter Object*) y entonces la herramienta de configuración podrá obtener toda la información de la EDS del nodo preguntando a dicho nodo a través de la red con los correspondientes mensajes de tipo explícito.

BUS de CAMPO CANopen

- Resultado de un proyecto de investigación financiado por la Unión Europea.
- Los mensajes I/O y explícitos de DeviceNet se denominan en CANopen PDO (*Process Data Object*) y SDO (*Service Data Object*) respectivamente.
- PDO hasta 8 bytes de datos ⇒ una única trama CAN de datos.
- Permite hacer uso de la trama CAN remota de petición de datos a un nodo productor.
- Los SDOs pueden tener cualquier longitud datos.
- También existen perfiles de dispositivo.
- **Diferencias DeviceNET y CANopen**
 - CANopen permite hasta 127 nodos mientras que DeviceNet permite 64.
 - En CANopen se permiten velocidades de transmisión de 10 k, 20 k, 50 k, 125 k, 250 k, 500 k, 800 k y 1 M bit/s.
 - DeviceNet tiene una especificación mucho más estricta en la capa física ⇒ facilita la integración
 - CANopen es más flexible tanto en la implementación física como en las funciones de comunicación que se incorporan en el nodo. Esto permite construir un nodo CANopen con el conjunto exacto de funciones requeridas y “optimizar” CANopen en cada aplicación para alcanzar el mejor rendimiento posible en las comunicaciones.
 - DeviceNet esta especializada en automatización de plantas industriales. CANopen esta más enfocada a redes internas de comunicaciones en máquinas y a sistemas distribuidos embebidos.
- **Diccionario de Objetos (OD):** contiene información entre otras de todas las variables de la red.
 - Cada entrada se direcciona con un índice de 16 bits (variables sin estructura) o con índice más subíndice de 8 bits (variables con estructura).
 - Todos los objetos son accesibles para lectura o escritura por medio de mensajes SDO.
 - Hay entradas opcionales y otras obligatorias.

CONCLUSIONES

- El bus CAN está muy bien adaptado para interconectar dispositivos de entrada-salida, sensores y actuadores inteligentes.
- Presenta buen comportamiento en tiempo real, de un modo muy seguro dada la sofisticada detección y confinamiento de errores que utiliza.
- Su método de acceso al medio permite los sistemas multimaestro, y la comunicación orientada a eventos (un evento es el cambio de valor de una variable booleana)
- Los nodos son fiables ya que están diseñados para un entorno agresivo como el del automóvil.
- Entre los buses de tiempo real aplicables a sistemas distribuidos en los que la relación prestaciones/coste es determinante, puede decirse que hasta la fecha ninguno ha alcanzado la combinación de aceptación, disponibilidad de dispositivos de diversos fabricantes y robustez que ofrece el bus CAN.
- El abanico de campos de aplicación donde se ha introducido el bus CAN es cada vez más amplio y el uso creciente de buses de campo basados en CAN principalmente DeviceNet y CANopen.
- En 2001 Bosch ha especificado el protocolo TTCAN (Time Trigered CAN) donde se intenta acercar esta nueva versión a las características necesarias que ha de tener un bus de comunicaciones en sistemas de tiempo real estricto.