



Santiago Escalante López

www.linkedin.com/in/santiago-escalante-lópez-851011361

CAN (Control Area Network)

¿Qué es?

CAN es un sistema de bus serial, en el cual los datos se transmiten bit a bit, y es de alta integridad (confiable y resistente a errores). Está diseñado para la comunicación entre dispositivos inteligentes en red. Los dispositivos y buses CAN son componentes comunes en los sistemas automotrices e industriales. Básicamente, la función principal es facilitar la comunicación entre diferentes unidades de control electrónico (ECU), quitando la necesidad de que cada componente tenga un cableado dedicado.

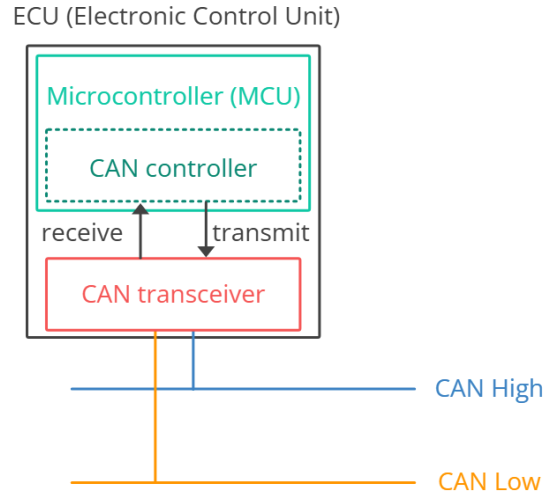
ECU

Es un componente que controla ciertas funciones, como la unidad de control del motor, la transmisión, los frenos, la dirección, las temperaturas, etc. (Un automóvil moderno suele tener más de 70). Un ECU está conformada por tres partes principales:

Microcontrolador: Se encarga de interpretar los mensajes CAN entrantes y decidir qué mensajes transmitir.

Controlador CAN: El controlador está usualmente integrado en el microcontrolador y se encarga de garantizar que todas las comunicaciones se adhieran al protocolo CAN, es decir todo lo relacionado con la codificación de mensajes, detección de errores; esto con el fin de eliminar las complicaciones del microcontrolador.

Transceptor CAN: Se encarga de conectar el controlador CAN a los cables CAN físicos, convirtiendo los datos del controlador en señales diferenciales para el sistema de bus CAN y viceversa. También proporciona protección eléctrica.

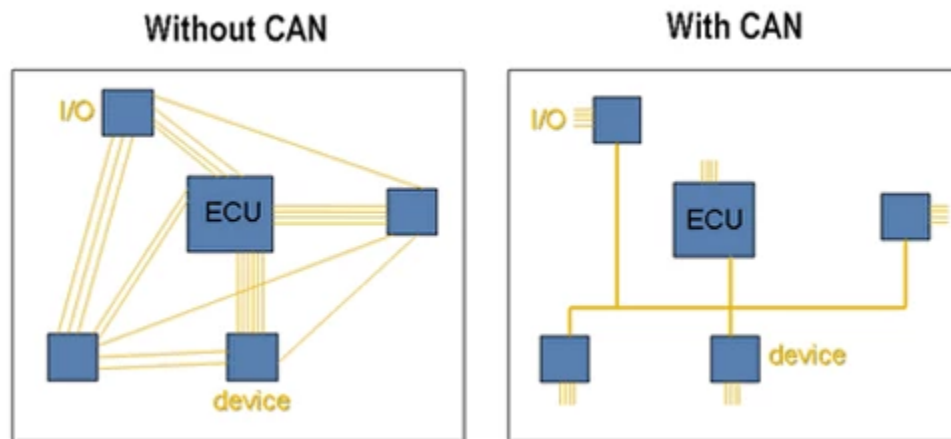


Beneficios

- **Red liviana y de bajo costo**
 - Proporciona una red económica que permite la comunicación entre varios dispositivos CAN a través de un único bus, lo que simplifica considerablemente el cableado. Además, una de sus principales ventajas es que las unidades de control electrónico (ECU) requieren solo una interfaz CAN, en lugar de múltiples entradas analógicas y digitales para cada componente del sistema.
- **Comunicación de radiodifusión**
 - Cada dispositivo conectado en la red es inteligente y tiene un chip controlador CAN, lo que los hace inteligentes y les permite a cada componente visualizar todos los mensajes transmitidos, además de decidir si un mensaje es importante o debe filtrarse. Esto permite que se puedan agregar nuevos dispositivos que solo reciban información sin necesidad de modificar el sistema.
- **Prioridad de mensajes**
 - Los mensajes transmitidos por cada dispositivo tienen un cierto nivel de prioridad, donde si dos elementos intentan transmitir un mensaje, el de mayor prioridad se transmite y el de menor prioridad se pospone.
- **Capacidad de error**
 - La especificación CAN incluye un código de redundancia cíclica (CRC) para realizar la verificación de errores en el contenido de cada frame (los mensajes que se transmiten a través de la red). Todos los nodos (dispositivos conectados al bus que pueden enviar, recibir y procesar datos) ignoran los frames que contienen errores. Además, se puede enviar un frame de error para notificar a la red sobre la falla. El controlador es capaz de distinguir entre errores locales y globales, y si se detecta un exceso de errores, los nodos afectados pueden interrumpir su

transmisión de mensajes defectuosos o incluso desconectarse completamente de la red.

- **Escalabilidad y flexibilidad**
 - Es posible agregar más ECUs sin tener la necesidad de rediseñar todo el sistema.
- **Gran velocidad de transmisión de datos**
 - CAN permite una transmisión de datos rápida y en tiempo real, lo que posibilita un monitoreo constante y una respuesta eficiente en sistemas críticos, como el control del motor o los frenos.



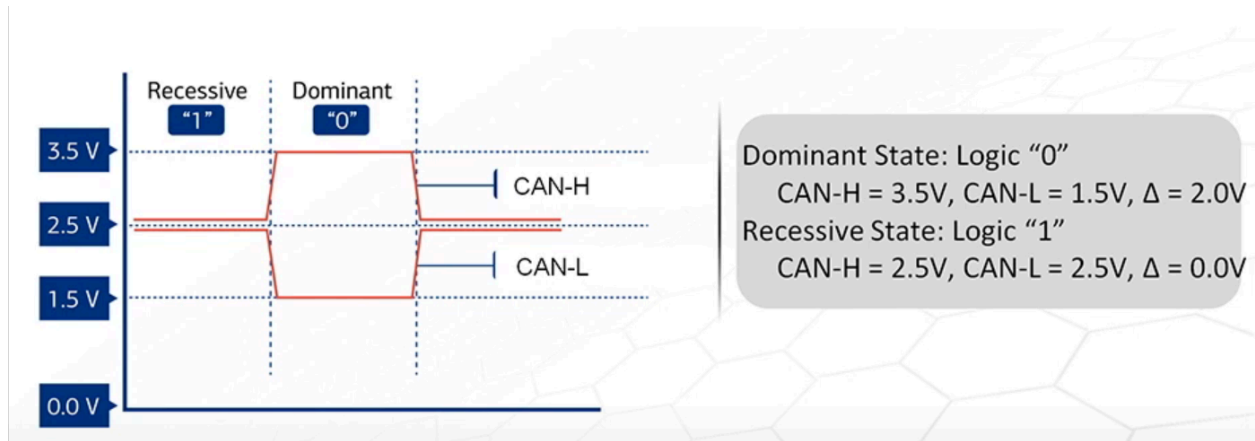
Arquitectura del bus

En primer lugar, un bus es una vía electrónica que facilita la transferencia de datos, señales de control y direcciones entre los componentes de un sistema, como procesadores, memorias, y otros dispositivos. En cuanto a la estructura del CAN bus, este está compuesto por 2 cables que se encuentran entrelazados para poder reducir las interferencias electromagnéticas, los cuales son los siguientes:

CAN HIGH (suele ser un cable amarillo): Señales de nivel lógico alto (voltaje mayor). Los sistemas más importantes son parte de la red del CAN HIGH

CAN LOW (suele ser un cable verde): Señales de nivel lógico bajo (voltaje menor)

State	CAN High voltage	CAN Low voltage	Bit type
Recessive	2.5 volts	2.5 volts	1
Dominant	3.5 volts	1.5 volts	0



*Cuando un bit recesivo es transmitido, el diferencial de voltaje es de 0 V, mientras que cuando se transmite un bit dominante, el diferencial de voltaje es de 2 V.

¿En que se podría usar en el coche?

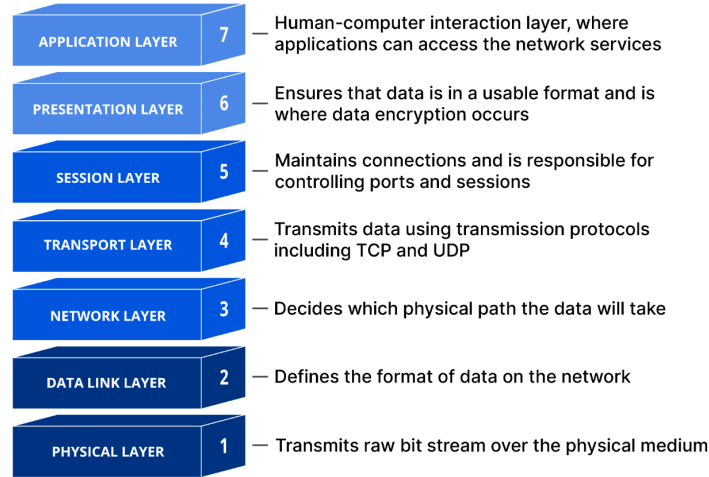
Gracias a que CAN permite la comunicación entre sistemas críticos, en la industria se utiliza en aplicaciones como los sistemas antibloqueo de frenos (ABS), el control de tracción y el control electrónico de estabilidad, los cuales dependen de intercambios de datos instantáneos para que funcionen correctamente.

Capas OSI involucradas

CAN está basado en las dos primeras capas del modelo OSI (Open System Interconnection), el cual es un modelo conceptual que permite que diversos sistemas de comunicación se conecten usando protocolos estándar:

Capa física: Se encarga de las conexiones físicas, especifica los niveles o características eléctricas de las señales, así como la codificación, decodificación, temporización y sincronización de los bits del frame.

Capa de enlace de datos: Se encarga de la transmisión de los bits en frames de información, encargándose de que estos lleguen a su destino sin errores. De igual forma, controla las secuencias de transmisión y retransmisión (en caso de que no se haya recibido el mensaje correctamente).



Tipos de frames

Data Frame:

- Envía datos reales
- Tiene un identificador (ID) para prioridad y un bit RTR dominante (0).
- Campo de datos que puede tener de 0 a 8 bytes.

Remote Frame:

- Solicita que otro nodo envíe un Data Frame con cierto ID.
- Bit RTR recesivo (1) que indica solicitud y no lleva campo de datos.

Error Frame:

- Enviado por un nodo cuando detecta un fallo en el frame, para avisar al resto que algo está mal.

Overload Frame:

- Pide una pausa o indica que un nodo está sobrecargado o buscando tiempo antes de continuar con otro frame.

Estructura del frame CAN (Standard)

Start of Frame (SOF): 1 bit dominante (0).

Campo de arbitraje (Arbitration Field): 12 bits en total.

- 11 bits de identificador (ID)
- 1 bit RTR (Remote Transmission Request)

Campo de control (Control Field): 6 bits:

- 1 bit IDE (Identifier Extension)
- 1 bit r0 (reservado)
- 4 bits DLC (Data Length Code)

Campo de datos (Data Field): Información transmitida que puede ser de 0 a 8 bytes.

Secuencia CRC (CRC Sequence):

- 15 bits de CRC
- 1 bit delimitador recesivo.

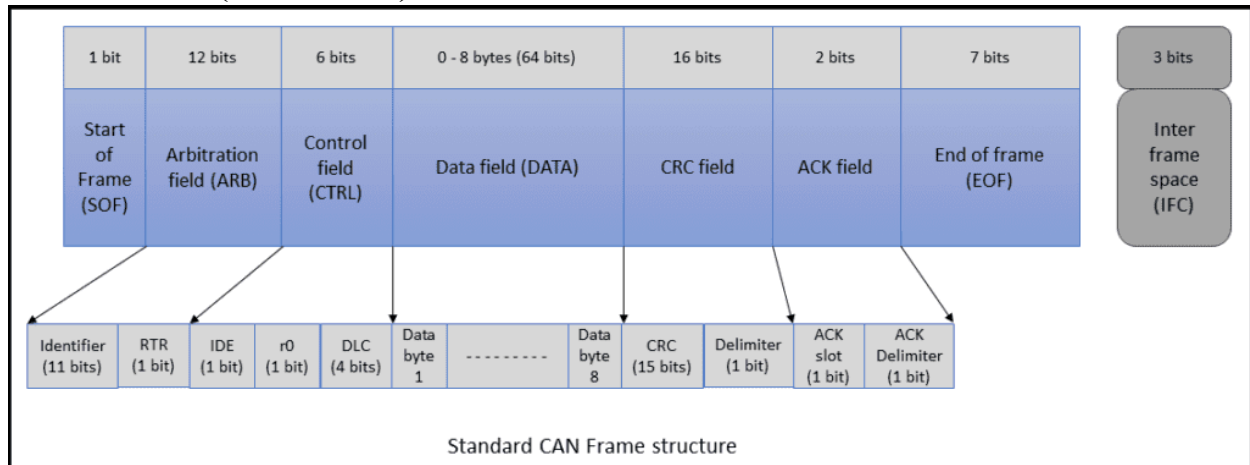
Campo de Acknowledgement (ACK Field): 2 bits:

- 1 bit ACK Slot (los receptores dominan si lo recibieron correctamente)
- 1 bit delimitador recesivo

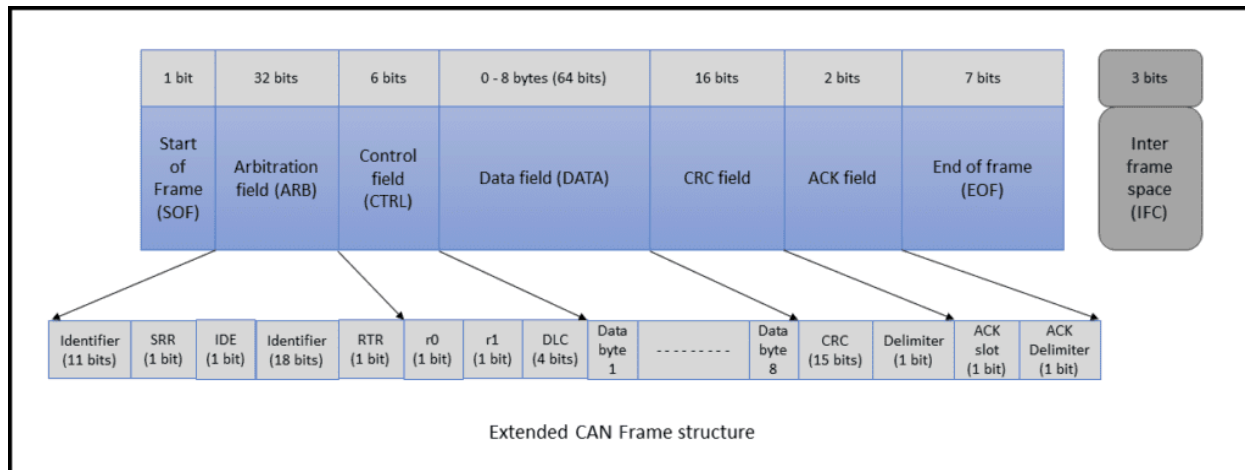
End of Frame (EOF): 7 bits recesivos (1111111)

Inter-Frame Space (IFS): mínimo 3 bits recesivos (111).

Standard CAN (ID de 11 bits)



Extended CAN (ID de 29 bits)



Standard CAN frame format (11-bit CAN ID)	Extended CAN frame format (29-bit CAN ID)
11 bit CAN-Identifier	29 bit CAN-Identifier
Arbitration field (ARB) is 12 bit long.	ARB field is 32 bit long.
If IDE (Identifier Extension) is dominant (0) then its a standard 11-bit frame. Control field (CTRL) contains IDE.	If IDE is recessive (1) then its a Extended 29-bit frame. ARB field contains IDE (Identifier Extension).
It does not contain SRR (Substitute Remote Request) bit.	It contains SRR bit and is always transmitted as a recessive (1) bit.
CTRL field contains 1 reserve bit (r0).	CTRL field contains 2 reserve bits (r0 and r1).
It can contain up to $2^{11} = 2048$ unique CAN-IDs.	Extended CAN frame can contain up to $2^{29} = 536,870,912$ unique CAN-IDs.

Standard CAN frame vs Extended CAN frame

Arbitraje de mensajes

El arbitraje de mensajes permite resolver conflictos de acceso al bus mediante una comparación bit por bit. Cada mensaje CAN posee un identificador (ID) que también determina su prioridad: cuanto menor es el valor del ID, mayor es la prioridad del mensaje.

El acceso al bus en CAN es descentralizado; los nodos transmiten sus mensajes sin coordinación previa. Cuando dos o más nodos intentan transmitir simultáneamente, el proceso de arbitraje decide cuál de ellos mantiene el control del bus.

El campo de arbitraje incluye el identificador del mensaje (de 11 bits en formato estándar o 29 bits en formato extendido) y el bit RTR (Remote Transmission Request), que distingue entre data frames (RTR=0) y remote frames (RTR=1).

El mecanismo se basa en dos niveles físicos: dominante (0) y recesivo (1). Debido a que el nivel dominante sobrescribe al recesivo, el mensaje del cual su identificador contiene un bit dominante en la primera posición donde difiere de los demás (ID más bajo), gana el arbitraje y continúa la transmisión.

*El arbitraje es un proceso **NO destructivo** (no corrompe los frames).

Estados de los nodos y tolerancia a fallos

Además de la verificación de errores mediante CRC y otras técnicas, el sistema CAN incorpora un mecanismo de tolerancia a fallos que protege la estabilidad de toda la red. Cada nodo CAN mantiene dos contadores internos, los cuales son el Transmit Error Counter (TEC) y el Receive Error Counter (REC).

Con base en el valor de estos contadores, el nodo puede encontrarse en uno de tres estados operativos:

Error Active: es el estado normal. El nodo puede enviar y recibir frames y tiene la capacidad de generar “Active Error Flags” en caso de detectar una anomalía.

Error Passive: si el TEC o el REC superan un cierto valor (REC o $TEC > 127$), el nodo pasa a este estado. Aún puede participar en la red, pero sus acciones están restringidas: genera “Passive Error Flags” que no dominan el bus, y además debe esperar tiempos adicionales antes de volver a transmitir, con el fin de dar prioridad al resto de la red.

Bus Off: si el TEC excede otro valor límite ($TEC > 255$) el nodo se aísla automáticamente del bus: deja de transmitir y de generar flags de error para evitar bloquear el sistema.

Gracias a este mecanismo, cuando un nodo presenta fallos continuos o sistemáticos, la red lo aísla progresivamente para que no comprometa al sistema.

Variantes del CAN

CAN de baja velocidad: Versión económica y tolerante a fallas, usada donde la comunicación debe seguir aunque haya errores. Hoy se reemplaza por el bus LIN.

CAN de alta velocidad: El tipo más común en autos y maquinaria; conecta módulos como motor, frenos o dirección.

CAN FD: Versión mejorada del CAN clásico, con más datos por mensaje y mayor velocidad, aunque aún poco usada.

CAN XL: La más avanzada; transmite mucha más información y más rápido, acercándose a la velocidad del Ethernet automotriz.

CAN bus versions

Property	Fault-tolerant CAN (low-speed CAN)	Classical CAN 2.0 (high-speed CAN)	CAN FD (Flexible Data-rate)	CAN XL
Max baud rate speed	0.125 Mbit/s	1 Mbit/s	8 Mbit/s (data phase)	20 Mbit/s
Max data payload size	8 bytes	8 bytes	64 bytes	2048 bytes
Baud rate type	Fixed	Fixed	Variable (faster data field)	Variable (higher rates)
Use cases	Fault-tolerant, body control modules	Real-time automotive applications	High data throughput, ADAS, EV applications	Future high-data applications
Key features	Fault-tolerant operation, continue even if one bus line is damaged	Low cost, robust error detection, most commonly deployed	Increased payload, speed and reliability	Increased payload, speed and reliability

Referencias

Descripción general del protocolo de la red de controladores de área (CAN). (2006, August 31). NI.

<https://www.ni.com/es/shop/seamlessly-connect-to-third-party-devices-and-supervisory-system/controller-area-network--can--overview.html?srsId=AfmBOop3YgTmQvafig3fKUHeTvsKg6sXRhB0UXb3dcMCY6KM1BDkhC9M>

Sital Technology. (2025, April 29). GBP: How did CAN Bus revolutionize the automotive industry?

<https://sitaltech.com/gbp-how-did-can-bus-revolutionize-the-automotive-industry/>

CAN Frames - NI. (2023, November 6). <https://www.ni.com>.

<https://www.ni.com/docs/en-US/bundle/ni-xnet/page/can-frames.html>

Electronics, C. (2025, January 23). CAN Bus Explained - A simple intro [2025]. CSS

Electronics. <https://www.csselectronics.com/pages/can-bus-simple-intro-tutorial>

¿Qué es y cómo funciona el sistema CAN-BUS? - MTE-THOMSON. (2022, November 24).

MTE-THOMSON. <https://mte-thomson.com/es/?noticias=sistema-can-bus>

¿Qué es un bus en informática? | Lenovo México. (n.d.).

<https://www.lenovo.com/mx/es/glosario/bus/?orgRef=https%253A%252F%252Fwww.google.com%252F&srsId=AfmBOooC1iV3CFRL2LrvzPRxqJF1nGfer8MdAjbI37vDqeUpIHxjtzam>

Kvaser. (2019, May 6). CAN Basics (bit-wise arbitration, differential signals) - Kvaser.

<https://kvaser.com/lesson/can-basics-bit-wise-arbitration-differential-signals/>

Automotive Communication Networks, Part II CAN Bus. (n.d.).

<https://www.snapon.com/EN/US/Diagnostics/News-Center/CAN-Bus>

“Que es la red CAN-Bus.” (n.d.). YouTube. https://www.youtube.com/shorts/rm74rYVM-_w
What is CAN arbitration and how does this work? (2024, June 18).

<https://www.hms-networks.com/tech-blog/blogpost/hms-blog/2024/06/18/what-is-can-arbitration-and-how-does-this-work>

¿Qué es el modelo OSI?| Ejemplos de modelos OSI | Cloudflare. (n.d.).

<https://www.cloudflare.com/es-es/learning/ddos/glossary/open-systems-interconnection-model-osi/>

Capítulo 2: Definición y teoría de operación del BUS CAN [Capítulo]. Universidad de las Américas Puebla (UDLAP).

https://catarina.udlap.mx/u_dl_a/tales/documentos/lmt/pacheco_h_je/capitulo2.pdf

Electronics, C. (2025, January 6). CAN bus errors Explained - A simple intro [2025]. CSS Electronics. <https://www.csselectronics.com/pages/can-bus-errors-intro-tutorial>

Electronoobs. (2023, July 9). CAN Bus: Serial Communication - How it works? [Video]. YouTube. <https://www.youtube.com/watch?v=JZSCzRT9TTo>