

# Tecnología y Protocolos de Comunicación: Controlador de Red de Zona (CAN)

López Fernandez, Dryan Alonso - Madrigal Sánchez, Clara Catalina  
dryanlf@estudiantec.cr catmasa@estudiantec.cr  
Área académica de Ingeniería Mecatrónica  
Instituto Tecnológico de Costa Rica

**Resumen**—CAN es un protocolo estandarizado de red diseñada para permitir que varios microcontroladores y dispositivos sean capaces de comunicarse entre sí. En este estándar, se definen las primeras dos capas del modelo OSI, tal que no exista ningún controlador “maestro” ni “esclavo”; en su lugar, todos los dispositivos transmiten mensajes a la vez, con el dispositivo de más alta prioridad siendo el único capaz de finalizar su mensaje. El mensaje, codificado de manera serial, es recibido por todos los dispositivos de la red, incluyendo el transmisor. Este tipo de redes, diseñadas para ser embebidas en sistemas más grandes, son muy utilizadas en sistemas embebidos y ECUs interconectados, especialmente en la industria automotriz y manufactura, gracias a su bajo uso de cables, robustez, velocidad y alta tolerancia a la distorsión.

**Palabras clave**—Bus de datos, CAN, nodos, protocolo de comunicación, señal digital.

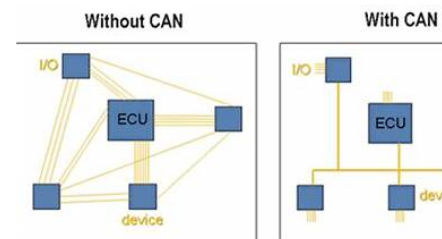


Figura 1. Ejemplo de la simplificación de las conexiones de una red por medio del protocolo CAN [4].

Luego, en 1992, se funda la asociación CiA (*CAN in Automation*), una asociación sin fines de lucro de usuarios y fabricantes de productos para el protocolo [1]. Gracias a sus esfuerzos, en 1993, el protocolo fue estandarizado internacionalmente como la norma ISO 11898-1 [5].

## I. CONTROLADOR DE RED DE ZONA COMO PROTOCOLO ESTANDARIZADO

### I-A. Creación de CAN

El protocolo CAN (*Controller Area Network*, por sus siglas en inglés) en su modalidad clásica fue desarrollado en 1986 por la empresa *Robert Bosch*, con el fin de reemplazar otras redes de buses comunes en ese entonces [1]. El objetivo principal de este desarrollo fue crear un control para sistemas de automóviles utilizando un único bus digital, en vez de las conexiones punto a punto usadas hasta entonces, pues éstas últimas se estaban volviendo cada vez más complejas, pesadas y costosas con cada nuevo componente electrónico agregado al sistema [2]. Utilizando únicamente únicamente dos cables como buses globales, y un acceso de “multimaestro”, **se reduce drásticamente la cantidad de cables necesarios para la conexión**, y no se requiere de ningún microcontrolador centralizado [3].



Figura 2. Ilustración promocional del protocolo CAN [1].

Desde entonces, usando de base la red CAN original, se han desarrollado una gran variedad de protocolos y adiciones adaptados para propósitos específicos, algunos inclusive agregados a las revisiones del estándar que le corresponde. **En general, la CiA es la entidad que define de la manera más detallada el hardware, codificación y recomendaciones requeridas para implementar el protocolo.**

### I-B. Arquitectura y descripción del protocolo

El protocolo CAN se enfoca en el estandarizar, en específico, **las primeras dos capas del modelo OSI**; es decir, la capa física y la de enlace de datos.

### I-C. Capa física

El estándar CAN, si bien no define el tipo de cables requeridos para la comunicación ni las tensiones y corrientes requeridas, establece el codificado de los bits y el sincronizado.

Al realizar un *broadcast* de un mensaje, un transmisor CAN envía una señal a los dos cables que conforman el bus, *CAN-H* y *CAN-L*, de manera simultánea. [3].

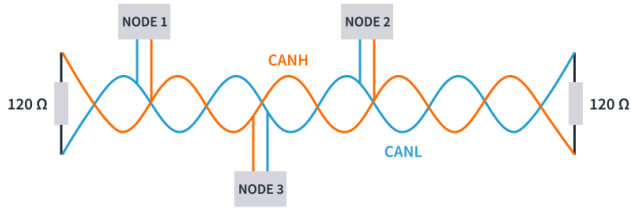


Figura 3. Ilustración de un ejemplo de una interfaz de la capa física de CAN [6].

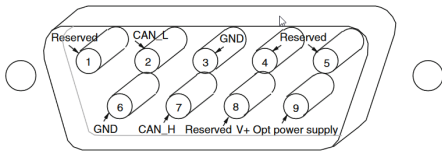


Figura 4. Ilustración de la conexión de bus, según lo definido por la CiA [3].

En los dos cables, ambos buses están conectados por terminales, para así generar la diferencia el potencial eléctrico requerida para la codificación. Las señales lógicas se podrían ver reflejadas en estos extremos si no se maneja, por lo cual existen varios tipos de terminales de bus. En los casos más simples, consta de una resistencia simple, pero existen otras variaciones. Por ejemplo, para una red CAN de alta velocidad, suelen usarse o una única resistencia, o resistencias rodeando un capacitor de valor intermedio [3]. Mientras tanto, para redes CAN de más bajas velocidades, se usan cargas resistivas simples. También se pueden incluir puentes, repetidores, etc, según la topología.

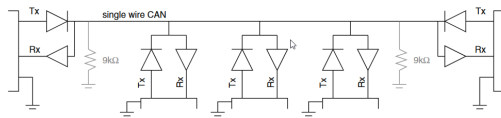


Figura 5. Ejemplo de una interfaz para CAN de baja velocidad [3].

Cabe también mencionar que los microcontroladores que conforman los nodos deben tomar en cuenta el retardo que ocurre al enviar una señal eléctrica a través de un medio físico. Por tanto, el estándar del protocolo define tiempos de transmisión de datos específicos, según las propiedades físicas de los buses y cuánto tiempo sea requerido para asegurar la correcta transmisión de los datos a lo largo de toda la red [3].

Bus speed	Cable type	Cable resistance/m	Terminator	Bus length
50 kb/s at 1,000 m	0.75–0.8 mm <sup>2</sup> (AWG18)	70 m Ω	150–300 Ω	600–1,000 m
100 kb/s at 500 m	0.5–0.6 mm <sup>2</sup> (AWG20)	<60 m Ω	150–300 Ω	300–600 m
500 kb/s at 100 m	0.34–0.6 mm <sup>2</sup> (AWG22, AWG20)	<40 m Ω	127 Ω	40–300 m
1,000 kb/s at 40 m	0.25–0.34 mm <sup>2</sup> (AWG23, AWG22)	<26 m Ω	124 Ω	0–40 m

Figura 6. Valores comunes de frecuencia de transmisión de redes con protocolo CAN [3].

Con respecto a lo que es otros aspectos físicos y electrónicos, el estándar ISO-11898 define valores aceptables de transmisión de datos y características físicas del hardware, dado que los principios del protocolo se podrían implementar de casi cualquier manera, mientras cumpla con los requerimientos de la red: la sincronización de los nodos a nivel de bits cuando ocurre la transmisión, que se cumpla con el arbitraje de la codificación y que todos los nodos reciban la misma señal del bus [3].

Parameter	Min	Max	Unit
DC Voltage on CANH and CANL	−3	+32	V
Transient voltage on CANH and CANL	−150	+100	V
Common mode bus voltage	−2.0	+7.0	V
Recessive output bus voltage	+2.0	+3.0	V
Recessive differential output voltage	−500	+50	mV
Differential internal resistance	10	100	kΩ
Common mode input resistance	5.0	50	kΩ
Differential dominant output voltage	+1.5	+3.0	V
Dominant output voltage (CANH)	+2.75	+4.50	V
Dominant output voltage (CANL)	+0.50	+2.25	V
Output current		100	mA

Figura 7. Requerimientos del hardware CAN según la norma ISO-1189 [3].

Para evitar la distorsión por ruido, CAN tiene una interpretación de señales eléctricas diferencial: la señal digital extraída se obtiene con la diferencia en potencial eléctrico entre los dos cables, de manera que requiere un cierto umbral para leerse como un bit HIGH [6]. Por esto, se suelen enroscar los dos cables de bus: si la distorsión es la misma en ambos, ambas señales tendrán la misma interferencia, causando que la diferencia entre las dos señales no se vea afectada [2].

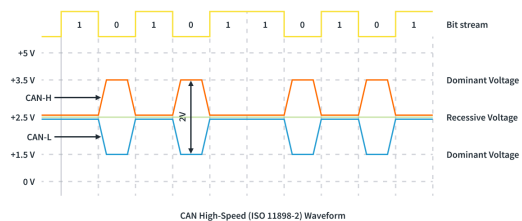


Figura 8. Ilustración de la interpretación de señales eléctricas para CAN [6].

Cada bit está codificado como un NRZ *Non Return to Zero* para aumentar la robustez del código, y por razones que se repasarán al discutir la segunda capa, se le considera a un estado HIGH como un 0 lógico, y a un LOW como 1, tal que

un 0 sobrescribe fácilmente un 1 si ocurren simultáneamente. En otras palabras, se define el 0 una *tensión dominante*, y un 1 como una *tensión recesiva*. Para asegurar la integridad de los datos, y dada la naturaleza asíncrona de la comunicación, se solicita constantemente a los nodos que se sincronicen con los ejes de los bits que estén siendo transmitidos en el momento, utilizando unos bits especiales ingresados en la codificación. [3].

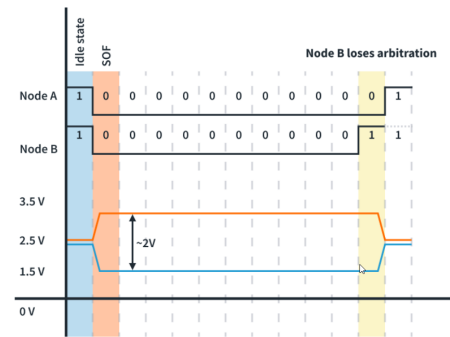


Figura 10. Ilustración de la etapa de arbitraje [6].

#### I-D. Capa de Enlace de datos

Con lo que respecta a la segunda capa del modelo OSI, el protocolo CAN define, en específico, la codificación de los data frames a transmitir.

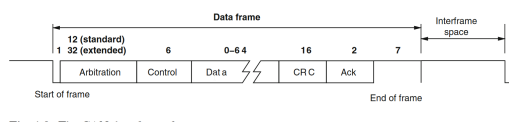


Figura 9. Ilustración del formato de data frame de CAN [3].

Lo primero que realiza una transmisión es comenzar con un bit 0, es decir, con una tensión dominante, para indicar el inicio de la transmisión. A esta transmisión se le llama SOF (*Start Of Frame*) y tiene el fin de permitirle a cada nodo el sincronizarse con la señal transmitida [?].

La primera sección de información es lo que es la capa de arbitraje. En éste, el microcontrolador encargado de un nodo transmite el valor, en bits, que lo identifica, junto con un bit RTR (*Remote Transmission Request*) [2]. Debido a que el sistema es asíncrono, es muy probable que más de un nodo quiera transmitir mensajes de manera simultánea a otro, a pesar de ser un único bus. Para evitar esto, se utiliza un sistema de jerarquía basado en los bits dominantes. Dado que:

- Un número de identificador menor contendrá más ceros.
- Un bit "0" es tensión dominante.

El protocolo establece que tienen prioridad de transmisión aquellos nodos con menor número identificador, y gracias a cómo funcionan los buses, todo microcontrolador es capaz de leer su propio mensaje. Por tanto, si en la etapa de arbitraje, un nodo lee un bit que no corresponde a su número identificador, es debido a que un cero, correspondiente a un número de identificador menor, sobrescribió el bit de su identificador, en cuyo caso, el nodo detiene su transmisión y deja terminar a la señal con mayor prioridad.

Una vez que ya queda únicamente un nodo transmitiendo, se transmite lo que es la sección de control, cuyos 6 bits contienen información acerca del formato y contenido de la información a transmitir [3].

- El bit de IDE, que indica la codificación que presenta el mensaje: la base (11 bits) o la extendida (29 bits).
- Un bit reservado 0.
- Los bits de DLC, que indican la longitud del mensaje, varía entre 0 y 64 bits.

Seguido a eso, está la fase de transmisión de información, según el DLC de la etapa de control [6].

Luego, existe un campo CRC (*Cyclic Redundancy Check*), en donde se utilizan 15 bits para detectar errores ocurridos durante la transmisión. Luego, un bit ACK (*Acknowledgement*), el cual se usa para verificar que los nodos receptores recibieron correctamente el mensaje (0 si exitoso) [6].

Finalmente, se pasa al EOF (*End of Frame*), el cual consta de 7 bits de 1. Esto es para asegurar que el sistema se encuentre en estado recesivo, para así poder captar la siguiente señal sin problemas [3].

Cabe mencionar que, durante todo el data frame, ocurren dos fenómenos:

- Los IFS (*Inter-Frame Spacing*), que constan de 7 bits utilizados para separar frames contiguas.
- Bits que se ingresan en medio de los frames, para darle a los nodos receptores un chance para verificar si están sincronizados con la transmisión del mensaje. A esto se le llama *Bit stuffing* [6]

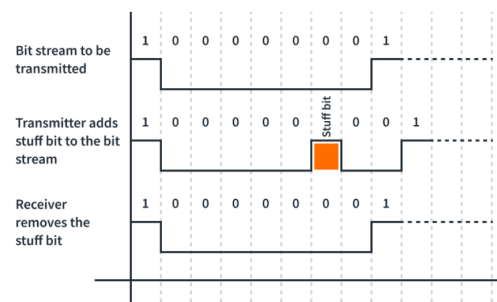


Figura 11. Ilustración del "bit stuffing" [6].

## II. APLICACIONES DE LA TECNOLOGÍA CAN

Para este proyecto, se realizó una implementación simple del protocolo CAN básico, el cual se puede observar siguiendo el siguiente enlace: <https://youtu.be/HthmyCX7gJU>

### II-A. Casos de uso típicos

En general, el protocolo se utiliza para diseñar redes embebidas que conectan los módulos input/output genéricos de un sistema, en especial las unidades ECU (*Electric Control Units* que controlan individualmente cada grupo de actuadores [1].

Gracias a su robustez y velocidad, es muy usado en aplicaciones críticas que involucran poco espacio para error. En general, se utiliza dentro de la lógica interna de automóviles, especialmente en lo que es el sistema de frenado anti bloqueo ABS y el programa de estabilidad electrónica [2].

Aún así, dada la etapa de verificación requerida para iniciar todo mensaje, junto con el espaciado al final de cada señal y que el modelo básico solo permite la transmisión de un mensaje a la vez, causa que la transmisión de mensajes sea más lenta de lo que otros protocolos más complejos son capaces. Además, debido a la naturaleza al azar de la activación de cada nodo y la comunicación asíncrona, el comportamiento de la red es relativamente impredecible [2]. Agregado a eso, cabe mencionar que la detección de errores depende de que el transmisor de un mensaje también lo reciba, es poco confiable a largas distancias, donde puede que la señal se haya distorsionado demasiado. Finalmente, el estándar del protocolo establece que la conexión máxima ha de ser de 64 nodos para asegurar la integridad de los mensajes, lo cual limita el posible tamaño del sistema al cual la red pertenece [7].

### II-B. Tipos y variaciones del protocolo CAN

La norma ISO-11898, la cual establece el estándar del protocolo CAN, tiene varias modalidades de red, para las cuales existe una sección diferente de la norma.

- CAN de alta velocidad: definido por el estándar ISO 11898-2, utiliza una topología de bus y soporta velocidades de transmisión de hasta 1Mbit/s. [7]
- CAN de baja velocidad o tolerante a fallos: definido por el estándar ISO 11898-3, puede utilizar topología de bus, estrella o una mezcla de ambas, se diferencia al de alta velocidad en que los cambios de voltaje de los pines CANH y CANL son más grandes, soporta velocidades de hasta 125Kbit/s. [7]
- CAN de un solo cable: definido por el estándar SAE 2411, utiliza un solo cable de comunicación, esto hace que la inmunidad al ruido sea inferior y se necesite utilizar velocidades más bajas de comunicación, las cuales pueden llegar hasta 83.33 Kbit/s. [7]

Esas son las normadas, pero existen otras, las cuales han sido desarrolladas y verificadas, en su mayoría, por la CiA (*Flexible time-triggered Communication on CAN*), los cuales varían en la manera en que se coordinan los nodos, inclusive nombrando a un nodo como "maestro encargado de la sincronización de todo el resto de nodos [2].

Además, según la aplicación, se le suele agregar varios filtros y registros a los nodos, para que así sean capaces

de interpretar, ignorar o retener ciertos tipos de información, según sea conveniente.s[3].

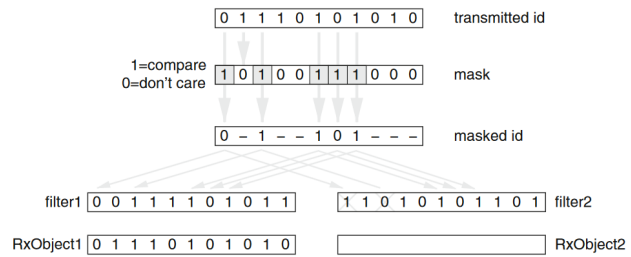


Figura 12. Ilustración de una posible configuración de filtros y registros en CAN [3].

## REFERENCIAS

- [1] CiA, "CAN in Automation:International users' and manufacturers' group", folleto informativo, Setiembre 2022 [Online]. Recuperado de: [https://www.can-cia.org/fileadmin/cia/documents/brochures/cia\\_image\\_2022.pdf](https://www.can-cia.org/fileadmin/cia/documents/brochures/cia_image_2022.pdf)
- [2] J. Ferreira and J. Fonseca, "Controller Area Network", *Informa Uk Limited*, 28 de Febrero, 2011 [Online serial]. Recuperado de: <https://core.ac.uk/works/6491441>
- [3] M. Di Natale, H. Zeng, et al., *Understanding and Using the Controller Area Network Communication Protocol*, Aufl. New York, NY: Springer-Verlag, 2012.
- [4] H. Bernier, "Automotive Can Bus System Explained", *Wiring and Engine Fix Colecion*, 25 de Abril, 2023 [Online]. Recuperado de: <https://schematicpartclaudia.z19.web.core.windows.net/automotive-can-bus-system-explained.html>
- [5] *Road Vehicles - Controller Area Network (CAN)*, ISO 11898, International Organization for Standarization, 2015.
- [6] J. Shah, "Understanding CAN: A Beginner's Guide to the Controller Area Network Protocol", 21 de Febrero, 2024 [Online]. Recuperado de: <https://www.circuitbread.com/tutorials/understanding-can-a-beginners-guide-to-the-controller-area-network-protocol>
- [7] WatElectronics, "What is CAN Protocol : Architecture, Working and Types", *watelectronics.com*, 7 de Abril, 2022 [Online]. Recuperado de: <https://www.watelectronics.com/controller-area-network-protocol/>