

# lamaPLC Communication: CAN

A *controller area network* (**CAN** bus) is a vehicle bus standard designed to allow microcontrollers and devices to communicate with each other. It is a message-based protocol, designed originally for multiplex electrical wiring within automobiles to save on copper, but it can also be used in many different contexts. For each device, the data in a frame is transmitted serially but in such a way that if more than one device transmits simultaneously, the highest priority device can continue while the others back off. All devices, including the transmitting device, receive frames.



Development of the CAN bus started in 1983 at Robert Bosch GmbH. The protocol was officially released in 1986 at the Society of Automotive Engineers (SAE) conference in Detroit, Michigan. The first CAN controller chips were introduced by Intel in 1987, and shortly thereafter by Philips. Released in 1991, the Mercedes-Benz W140 was the first production vehicle to feature a CAN-based multiplex wiring system.

The CAN bus history in short:

- Pre-CAN: Car ECUs relied on complex point-to-point wiring
- 1986: Bosch developed the CAN protocol as a solution
- 1991: Bosch published CAN 2.0 (CAN 2.0A: 11 bit, 2.0B: 29 bit)
- 1993: CAN is adopted as an international standard (ISO 11898)
- 2003: ISO 11898 becomes a standard series
- 2012: Bosch released the CAN FD 1.0 (flexible data rate)
- 2015: The CAN FD protocol is standardized (ISO 11898-1)
- 2016: The physical CAN layer for data rates up to 5 Mbit/s standardized in ISO 11898-2

The CAN bus physical layer defines cable types, electrical signal levels, node requirements, cable impedance, etc. For example, ISO 11898-2 dictates several things, including the following:

- Baud rate: CAN nodes must be connected via a two-wire bus with baud rates up to 1 Mbit/s (Classical CAN) or 5 Mbit/s (CAN FD)
- Cable length: Maximal CAN cable lengths should be between 500 meters (125 kbit/s) and 40 meters (1 Mbit/s)
- Termination: The CAN bus must be properly terminated using a 120-ohm CAN bus termination resistor at each end of the bus

## CANopen

CANopen is a higher-layer protocol that is used for embedded control applications. Because it is based on the CAN messaging protocol, DAQ systems and data loggers that can read and record CAN data can also access data from CANopen.

CANopen was invented to provide easy interoperability among devices in motion control systems. Communication among and between devices is implemented at a high level, and device configuration is also supported. It's heavily used in motion control, robotics, and motor control applications.

CANopen is managed by the international organization CAN in Automation - CiA. Established in

Germany in 1992, CiA is a non-profit international users/manufacturers group for CAN. They take pride in being an unbiased platform for developing the CAN protocol and promoting the image of CAN technology.

## Low-Speed CAN Bus

The low-speed CAN Bus, a fault-tolerant CAN, supports bit rates between 40 kbit/s and 125 kbit/s. It provides a robust communication system that allows communication to continue even when one of the two wires experiences a fault. Each CAN node's dedicated CAN termination further enhances the resilience, ensuring seamless data transfer.

## High-Speed CAN Bus

The high-speed CAN Bus is based on the ISO 11898 standard and is widely used in modern applications. It supports bit rates between 40 kbit/s and 1 Mbit/s and offers simple cabling solutions, making it the most popular choice in the industry today. It is the foundation for various higher-layer protocols such as OBD2, CANopen, and J1939, further solidifying its significance in communication systems.

## CAN FD

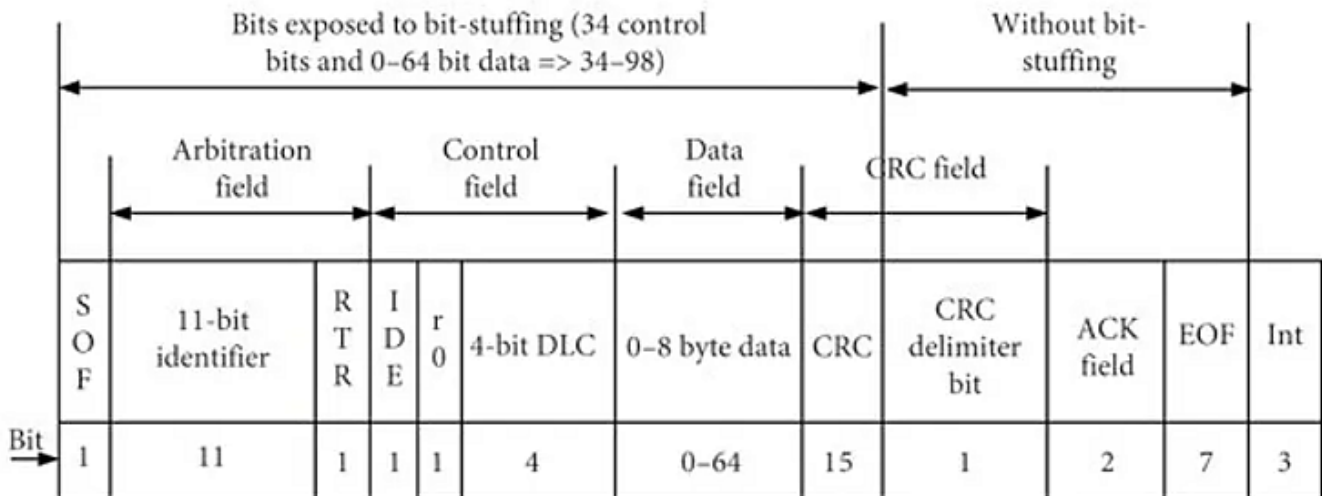
CAN FD, or CAN Flexible Data Rate, is an advanced communication protocol in modern high-performance vehicles.

CAN FD (*Controller Area Network Flexible Data-Rate*) is an extension of the original CAN bus protocol specified in ISO 11898-1. Developed in 2011 and released in 2012 by Bosch, CAN FD was designed to meet the need for increased data transfer rates up to 5 times faster and larger frame/message sizes for use in modern automotive Electronic Control Units (ECUs). As with the classic CAN, the CAN FD protocol is designed to transmit and receive sensor data and control commands reliably, and to detect data errors between electronic sensor devices, controllers, and microcontrollers.

ISO11898-1 CAN FD shares the physical layer, with the CAN protocol as defined in the BOSCH CAN Specification 2.0, can dynamically switch to different data rates and with larger or smaller message sizes.

Enhanced features in CAN FD include the capability to dynamically select and switch to faster or slower data rate, as and when required, and to pack more data within the same CAN frame/message and transport it over the CAN BUS/network in less time. Faster data speeds and greater data capacity enhancements result in several operational advantages over the classic CAN. Using CAN FD, sensor and control data can be sent and received by the ECU (Electronic Control Unit) software much quickly. Commands issued by the executing ECU software reach the output controller much faster. CAN FD is typically used in high-performance ECUs of modern vehicles. A modern car can have more than 70 ECUs that use CAN FD to exchange information over the CAN Bus when the engine is running or when the vehicle is moving.

## CAN Frame



- **SOF:** The Start of Frame is a 'dominant 0' to tell the other nodes that a CAN node intends to talk
- **ID:** The ID is the frame identifier - lower values have higher priority
- **RTR:** The Remote Transmission Request indicates whether a node sends data or requests dedicated data from another node
- **Control:** The Control contains the Identifier Extension Bit (IDE), a 'dominant 0' for 11-bit. It also includes the 4-bit Data Length Code (DLC) that specifies the length of the data bytes to be transmitted (0 to 8 bytes)
- **Data:** The Data contains the data bytes, aka payload, which includes CAN signals that can be extracted and decoded for information
- **CRC:** The Cyclic Redundancy Check is used to ensure data integrity
- **ACK:** The ACK slot indicates if the node has acknowledged and received the data correctly
- **EOF:** The EOF marks the end of the CAN frame

## Related communication buses

In addition to CAN and the protocols that run on it, described in the previous sections, other communication buses are used for vehicle applications:

- MOST (Media-Oriented Systems Transport)
- Automotive Ethernet
- SENT SAE-J2716
- FlexRAY
- LIN Bus - Local Interconnect Network

## Sources

CSS Electronic: <https://www.csselectronics.com/pages/can-bus-simple-intro-tutorial>  
 DEWESoft: <https://dewesoft.com/blog/what-is-can-bus>

# CAN topics on lamaPLC

Page	Date	Tags
• <a href="#">lamaPLC Communication: CAN</a>	2026/04/23 21:51	<a href="#">communication</a> , <a href="#">bus</a> , <a href="#">can</a> , <a href="#">canopen</a> , <a href="#">most</a> , <a href="#">automotive ethernet</a> , <a href="#">sent sae-j2716</a> , <a href="#">flexray</a> , <a href="#">lin bus</a>
• <a href="#">lamaPLC Communication: ControlNet</a>	2026/04/23 21:51	<a href="#">bus</a> , <a href="#">communication</a> , <a href="#">devicenet</a> , <a href="#">cip</a> , <a href="#">can</a> , <a href="#">ethernet ip</a> , <a href="#">ethernet</a> , <a href="#">controlnet</a> , <a href="#">mac</a> , <a href="#">peer-to-peer</a> , <a href="#">allen-bradley</a> , <a href="#">rockwell automation</a> , <a href="#">fieldbus</a> , <a href="#">rg-6</a> , <a href="#">coaxial cable</a>
• <a href="#">lamaPLC Communication: DeviceNet</a>	2024/11/16 20:43	<a href="#">bus</a> , <a href="#">communication</a> , <a href="#">devicenet</a> , <a href="#">62026-3</a> , <a href="#">cip</a> , <a href="#">can</a> , <a href="#">ethernet ip</a> , <a href="#">ethernet</a> , <a href="#">controlnet</a> , <a href="#">mac</a> , <a href="#">peer-to-peer</a> , <a href="#">canopen</a> , <a href="#">allen-bradley</a> , <a href="#">rockwell automation</a> , <a href="#">bosch</a>
• <a href="#">LamaPLC: Arduino MKR Family</a>	2025/08/24 12:19	<a href="#">arduino</a> , <a href="#">board</a> , <a href="#">mkr fox 1200</a> , <a href="#">mkr zero</a> , <a href="#">ethernet</a> , <a href="#">shield</a> , <a href="#">m-bus</a> , <a href="#">rs-485</a> , <a href="#">can</a> , <a href="#">env</a> , <a href="#">sd card</a> , <a href="#">rgb</a> , <a href="#">gps</a> , <a href="#">iot</a> , <a href="#">wifi</a> , <a href="#">sigfox</a> , <a href="#">lora</a> , <a href="#">gsm</a> , <a href="#">lte-m</a>
• <a href="#">lamaPLC: TWAI basics</a>	2026/04/23 21:51	<a href="#">bus</a> , <a href="#">communication</a> , <a href="#">as-i</a> , <a href="#">basic</a> , <a href="#">can</a> , <a href="#">twai</a> , <a href="#">espressif</a> , <a href="#">esp32</a>

[communication](#), [bus](#), [CAN](#), [CANopen](#), [MOST](#), [Automotive Ethernet](#), [SENT SAE-J2716](#), [FlexRAY](#), [LIN Bus](#)

This page has been accessed for: Today: 3, Until now: 128

From:

<https://lamaplc.com/> - **lamaPLC**

Permanent link:

[https://lamaplc.com/doku.php?id=com:basic\\_can](https://lamaplc.com/doku.php?id=com:basic_can)

Last update: **2026/04/21 20:46**

