CAN Bus Explained (2021)

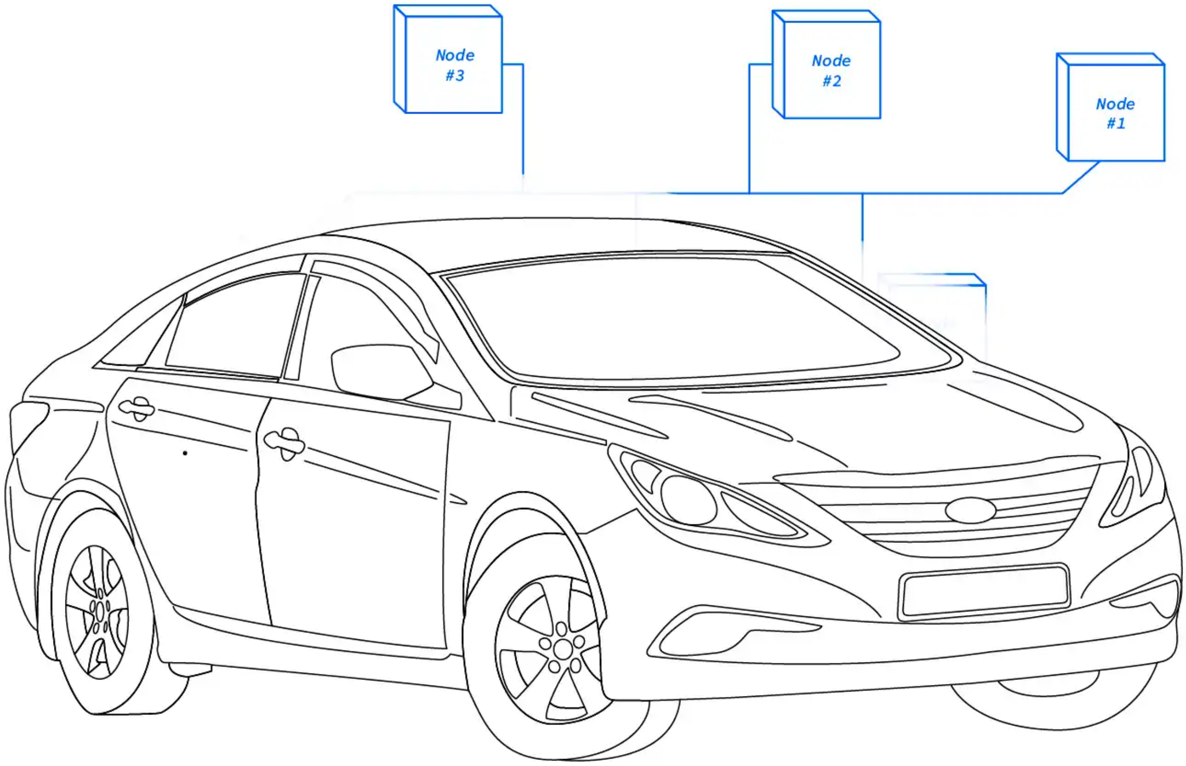
CAN Bus stands for Controller Area Network and consists of two electrical wires called CAN_Low and CAN_High. The information within each vehicle is being transmitted from and to ECUs and CAN Bus is built to handle robust performance within harsh environments.

We have prepared a simple introduction to CAN Bus. Several topics have been covered, in order to give you the best explanation of the (CAN Bus) protocol.

While working on the article, we combined the knowledge from our top experts within the company, as well as other team members, who do not have the expertise.

Why? The idea was to write a professional but simple introduction to CAN Bus for everyone, no matter how much experience you have.

In conclusion, no matter if you don’t have any knowledge about CAN bus whatsoever or you are already a pro. The simple intro to CAN bus will give you all the information needed.



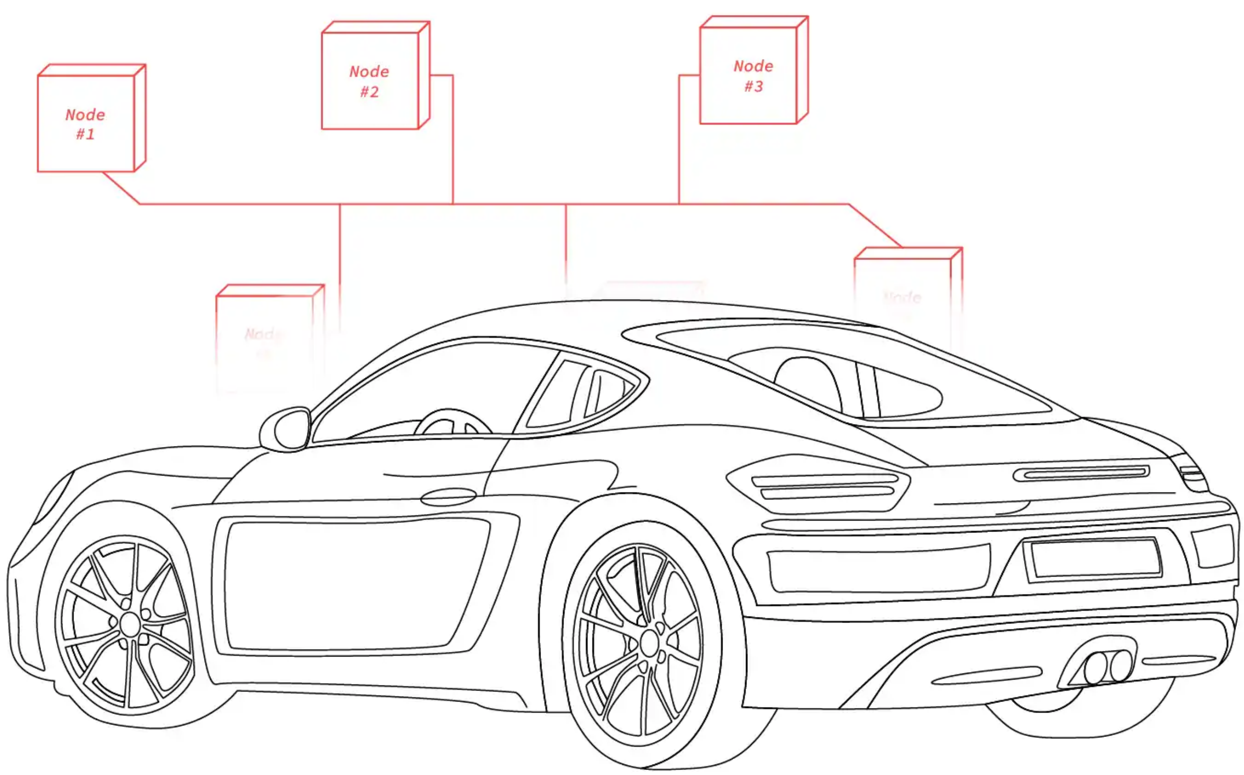
What is CAN Bus?

CAN Bus is a set of two electrical wires in the car network (CAN_Low and CAN_High), where the information is being sent to and from ECUs. The network that allows ECUs to communicate is called Controller Area Network (CAN).

The **CAN bus** is a serial communication bus, designed for robust performance within harsh environments, primarily in industrial and automotive applications.

It is basically a vehicle bus standard that allows microcontrollers and devices to communicate with each other.

CAN bus is one of the protocols used in the On-Board Diagnostics (OBD). OBD-2 is nowadays mandatory in all cars and light trucks all around the globe.



CAN bus easily explained

Now, let's try to look at it from a totally different point of perspective.

Imagine that your car is like a human body and the nervous system in the human body is the **Controller Area Network**

(CAN bus) in the car, which also enables the communication.

Nodes or electronic control units (ECUs) are something like body parts that are interconnected through the CAN bus. Information can be easily shared between parties. That is much easier to understand, isn't it?

CAN Bus system

Depending on the type of the car, it can have up to 70 ECUs (electronic control units) and each of them needs to be shared with other parts of the network.

Some of the examples are for instance; audio system, airbags, engine control unit, door control unit and so on. The CAN bus allows ECUs to communicate to each other.

Think about ECUs as specific people. One ECU can formulate and transmit the information via CAN bus to other ECUs that accept the data. After that, they will check the data and decide if they want to receive it or ignore it.

The CAN bus uses two wires for communication - **CAN low** and **CAN high** (CAN L and CAN H). ISO 11898-2 describes the physical layer of the CAN bus and ISO 11898-1 describes the data link layer.

The physical layer represents types of cables, node requirements, electrical signal levels, cable impedance and so on.

On the other side, ISO 11898-2 represents things such as baud rate, cable length and termination.

- Cable lengths should be 40 meters (1 Mbit/s) or 500 meters (125 kbit/s).
- The CAN bus has to be terminated using a 120 Ohms CAN bus resistor at the end of each bus.
- CAN nodes have to be connected through two wire bus with baud rates upto 1 Mbit/s (CAN) or 5 Mbit/s (CAN FD).

5 Benefits of CAN Bus

CAN bus standard is commonly used in all vehicles because of its key benefits, such as

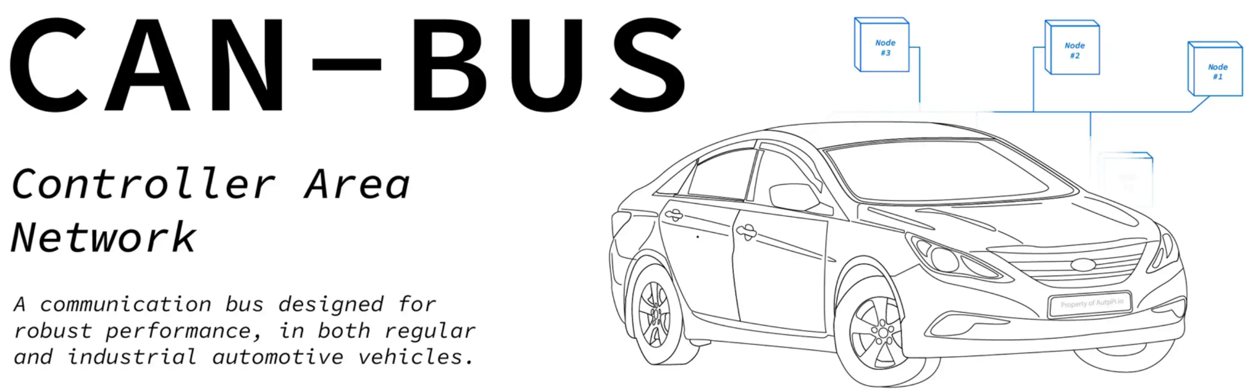
- Robustness
- Low cost
- Speed
- Flexibility
- Efficiency

- **Speed** - Currently defined by two physical layers - High Speed CAN and Low Speed CAN, both with its own advantages and disadvantages.
- **Efficiency** - High priority data will get prioritized by ID, in order to get immediate bus access - not interrupting other frames
- **Low-cost** - When the CAN protocol was created, its purpose was to enable fast communication between electronic devices and modules, while reducing errors,

weight, wiring and costs.

- **Robustness** - CAN bus standard is ideal for safety applications such as vehicles due to its durability and reliability. There are also 5 mechanisms to detect errors in the CAN protocol such as bit stuffing, bit monitoring, frame check, acknowledgment check and cyclic redundancy check.

- **Flexibility** - CAN bus protocol is well-known as a message-based protocol, meaning nodes can easily be added or removed without performing any updates on the system. This makes it really easy for engineers to integrate new electronic devices without significant programming and modify it to your requirements.



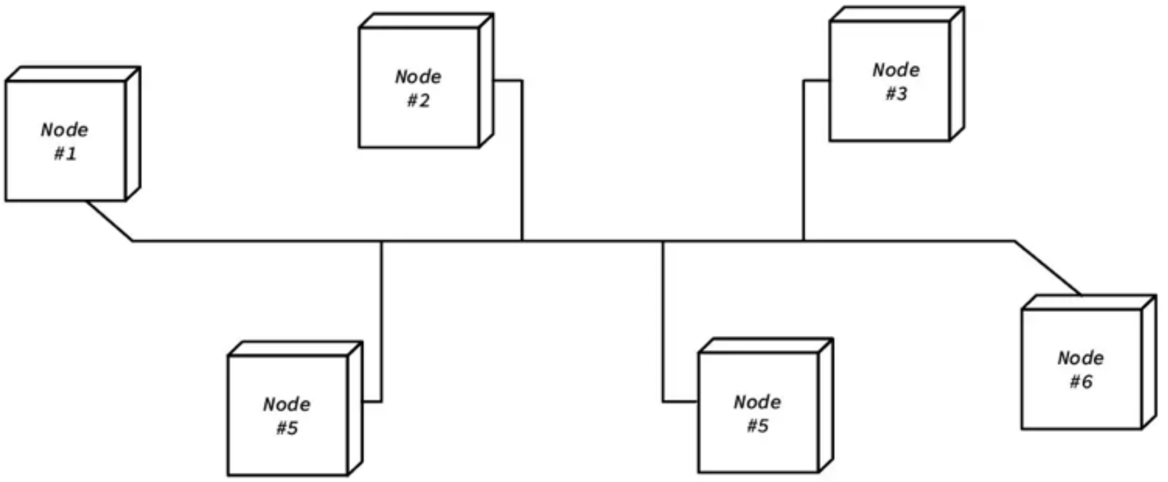
CAN Bus Wiring

One of the best advantages of CAN bus is the reduced amount of wiring in combination with an ingenious prevention of message collision.

In other words, no data will be lost during message transmission.

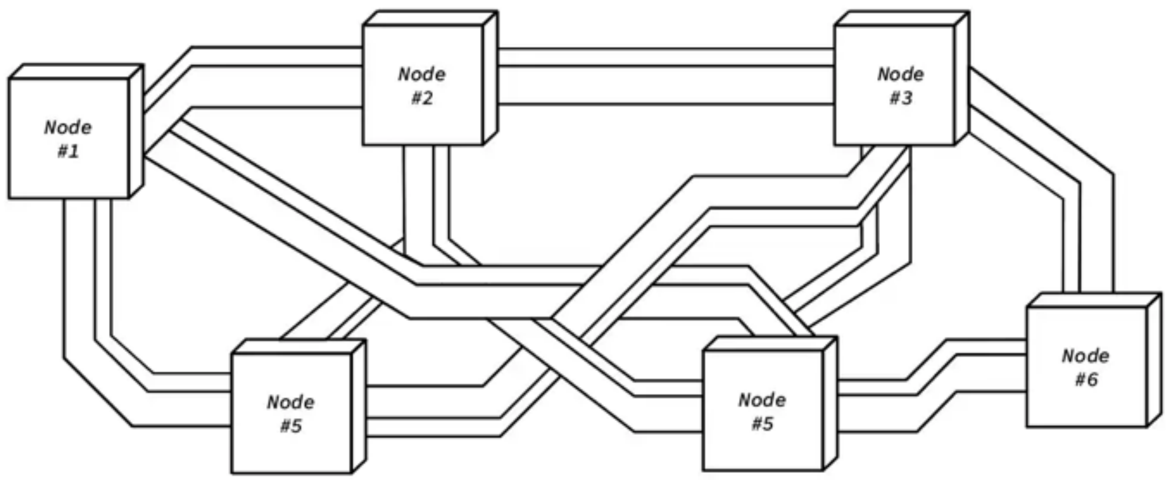
Two examples below show how the **CAN bus protocol** seems like with CAN bus and how it would look like without the CAN system.

Clearly, with CAN bus, it is much easier for nodes to communicate and navigate through.



With CAN System

On the other side, without a CAN bus, it is much harder for nodes to communicate with each other and the communication is ineffective.



Without CAN System

There are several different types of networks. You can find an easy explanation below.

High speed CAN bus (ISO 11898)

- Supports bit rates between 40 kbit/s and 1 Mbit/s
- Simple cabling

- Most commonly used these days
- Basis for higher layer protocols such as OBD2, CANopen, j1939 and more.

Low speed CAN bus

- Supports bit rates between 40 kbit/s and 125 kbit/s
- Allows communication to continue despite the fault in one of the two wires
- Also known as a fault tolerant CAN
- Each CAN node has own CAN termination

LIN bus

- Low-cost supplement
- Less harness
- Cheaper nodes
- Usually consists of a LIN master, which is acting as a gateway - up to 16 slave nodes

- Typically includes vehicle functions such as door functionality or aircondition

Automotive ethernet

- Ethernet supports high bandwidth requirements of Advanced Driver Assistance Systems (ADAS), cameras, infotainment systems and so on
- Provides much higher data transfer rates than CAN bus
- Lacks safety features of CAN and CAN FD
- Most likely will be used commonly in the upcoming years within the automotive industry

CAN FD

- Typically used in modern high performance vehicles
- CAN FD is an extension to the original CAN bus protocol
- Released in 2012 by Bosch
- Developed to meet the need to increase the data transfer

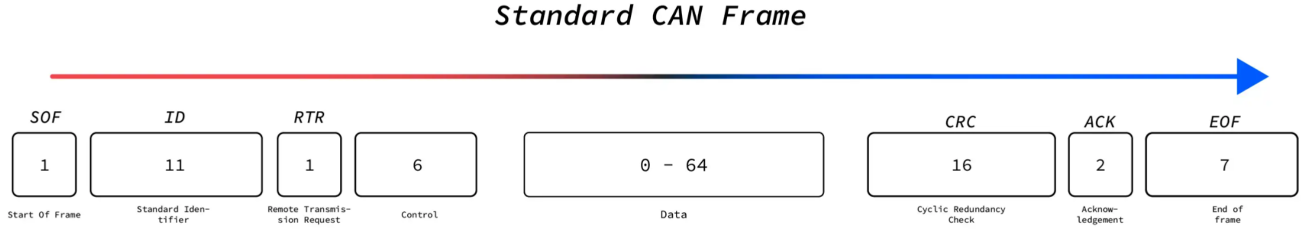
What is a CAN message frame?

CAN frames are used to communicate over CAN bus. CAN uses the differential signal with two logic states - dominant and recessive.

CAN network uses two CAN messages - standard CAN and extended CAN that are described below.

Standard CAN

Typically, a CAN frame with 11 bits identifier (CAN 2.0A) is used in most vehicles, which is also called standard CAN and can be seen in the picture below.



The first bit is the start of the frame (SOF), which represents the start of the **CAN message**. Next one is the 11 bit identifier that organizes the priority of the CAN message. The smaller the identifier is, the higher priority it has.

The Remote Transmission Request (RTR) is typically dominant, but it becomes recessive when nodes are requesting data from each other.

Next one is the identifier extension (IDE) bit, which is dominant when the standard CAN frame is sent - not extended one. The r0 bit is reversed and not currently used.

Next one is the Data Length Code (DLC), which indicates how many bytes of data are in the current message. Another important part is the data itself, where it is the same number of bytes as in DLC bits.

The next one is the cyclic redundancy check (CRC), which is a 16-bit checksum that detects errors and issues in the transmitted data.

In case the message is properly received, the receiving node will overwrite the recessive acknowledge bit (ACK) with a dominant bit. The end of frame (EOF) indicates the end of the CAN message.

It is 7 bits wide and it detects bit stuffing errors. The last part of the CAN message is the interframe space (IFS), which is being used as a time delay.

Extended CAN

Extended CAN uses a 29 bit identifier with a couple additional bits. The extended 29 bit identifier (CAN 2.0B) is identical, but has a longer ID and is usually used in the j1939 protocol - heave-duty vehicles. CAN uses two logic states; dominant and recessive.

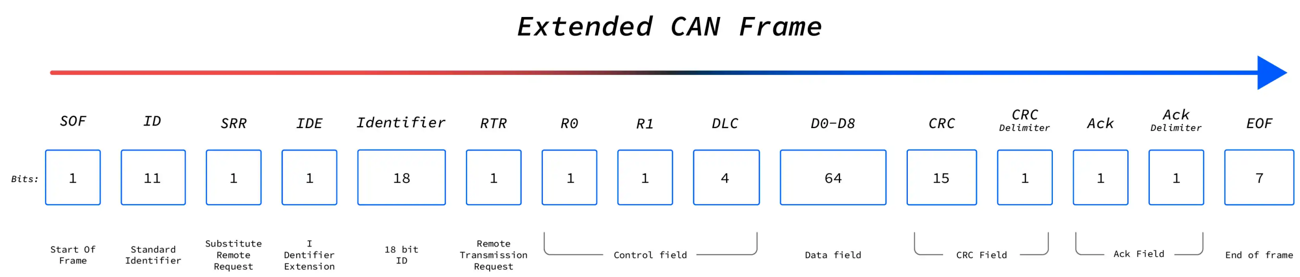
- **Dominant** - pinpoints that the differential voltage is greater than the minimum threshold. In addition, the dominant state is also achieved by driving a logic ‘0’ onto the bus.

- **Recessive** - pinpoints that the differential voltage is less than the minimum threshold. On the other side, recessive state is achieved by a logic ‘1’.

It also has a substitute remote request (SRR) bit, which comes after 11 bit identifier and acts like a placeholder, in order to keep the same structure as a standard CAN.

The identifier extension (IDE) should be recessive and the extended identifier should follow it accordingly.

The remote transmission request (RTR) comes right after the 18 bit ID. The reverse bit r1 follows the path and the rest of the message stays the same.



CAN Bus data logging

Logging CAN data can be done from several types of vehicles such as cars, heavy duty vehicles, predictive maintenance and machine blackbox.

The data from the car are gathered through the OBD2 port and are usually used to reduce fuel costs, improve car mileage and more.

On the other side, data from the heavy duty vehicles are gathered through j1939 and are usually used to improve safety and reduce costs.

Vehicles and machinery can be also monitored through IoT **CAN loggers**. That can be done in the cloud to avoid breakdowns. A CAN logger can provide data for disputes or diagnostics. It is also called blackbox.

```
500#0230632e0f
130#00085e9b229f7ac3
29c#09e30a21ffff04e4
354#04e4000000000000
0c6#813a82cf7ff1aad9
1f8#2a04a8ce980c8a2d
17a#ffffffbb00f02d1d
186#1f402a42fa0020
130#00285ea0229f7f99
12e#ae800e8130ffff00
29a#0a1b09e804e406fb
1f8#2404a8ce9c0c8c2d
0c6#814182bb7ff1ace4
186#1f402a52fa0020
130#00485ea5229f8172
29c#09d80a0affff04db
0c6#814782937ff1ae04
1f8#3d04a8cea10c6e2d
17a#ffffffbb00f02d31
186#1ef02a62fb0020
130#00685eab229f8647
12e#ae800e8140ffff00
29a#0a0a09d204db072a
5d7#04db0022dcd0cc
0c6#814d826b7ff1b024
1f8#4604a8cea60c602d
17a#ffffffbb00f02d3c
186#1ef02a72fb0020
4f8#44100005eb000000
130#00885eb1229f891e
29c#09c709f9ffff04d3
18a#fff000069c00
354#04d3000000000000
0c6#8153826b7ff1b21c
1f8#3e04a8ceac0c622d
17a#ffffffbb00f02d49
186#1ef02a92fb0020
130#00a85eb8229f8df3
12e#af800e8150ffff00
29a#09ee09bb04d30865
0c6#8159826b7ff1b414
1f8#7604a8cebfe1dc068ffff

pi@autopi-aa00b00e1d05:~ $ |
[1] 1:ssh  2:bash- 3:ssh*
```

Decoding raw CAN data

Raw CAN data is not easily readable. Therefore we have prepared a guide for you. Check out the guide on how to log raw CAN messages.
The CAN bus supports the basis for communication, but not more than that. The CAN standard protocol doesn't indicate how to handle message messages greater than 8 bytes, or

how to decode the RAW data.
In order to indicate how data is communicated between CAN nodes of a network, a set of standardized protocols come in handy. There are several higher layer protocols such as; OBD2, CANopen, CAN FD and SAE j1939.

- **OBD2** - OBD has a self-diagnostic capability that mostly mechanics use to analyze car issues and the overall health of the car. OBD2 determines trouble codes (DTCs) and real time data (RPM, speed, etc) that can be recorded through OBD2 loggers.
- **CANopen** - CANopen is typically used in embedded control applications such as industrial automation and is based on a CAN meaning that the CAN bus data logger is also capable to log CANopen data.
- **CAN FD** - CAN FD is a CAN bus with flexible data rate and an extension of the classical CAN data link layer. In comparison with classical CAN protocol, CAN FD increases the payload from 8 to 64 bytes. It also allows a higher data bit rate, depending on the CAN transceiver.
- **SAE J1939** - J1939 is commonly used in heavy duty vehicles. J1939 parameters such as RPM and speed are analyzed by a suspect parameter number (SPN).

Afterwards, they are grouped in parameter groups and classified by a PG number (PGN).

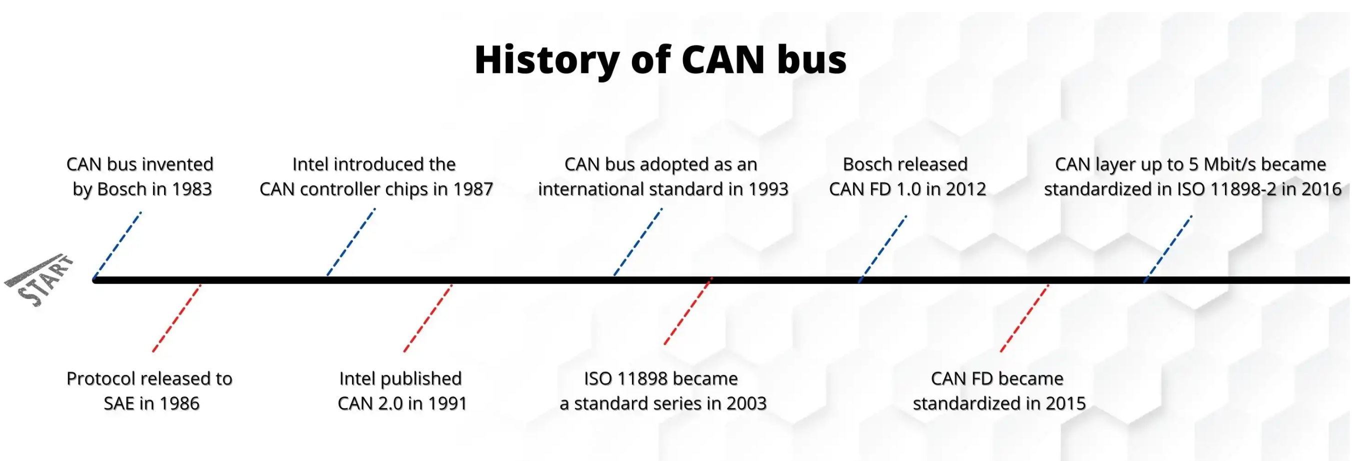
A high speed transfer data rate offers DoIP diagnostics, more precisely approximately 100 times of CAN diagnosis. Read more about what DoIP is here.

History of CAN Bus

Control Area Network (CAN Bus) has a rich history and went through several development stages. The actual development stages within years can be seen below.

- Development of CAN Bus goes all the way back to 1983 when Bosch originally invented Control Area Network and it was later codified into ISO 11898-1 standard.
- The protocol was later released to the Society of Automotive Engineers (SAE) in 1986.
- Intel was the first one to introduce the CAN controller chips in 1987, and Phillips joined Intel shortly after that.

- In 1991, Bosch published CAN 2.0 (CAN 2.0A: 11 bit, 2.0B: 29bit).
- CAN Bus as an international standard in ISO 11898, was adopted in 1993.
- In 2003, ISO 11898 became a standard series
- In 2012, Bosch has released the CAN FD 1.0 - flexible data rate
- In 2015, the CAN FD protocol has become standardized in ISO 11898-1
- Lastly, the physical CAN layer up to 5Mbit/s has become standardized in ISO 11898-2, in 2016



In the future, CAN bus will still be commonly used, but influenced by major automotive industry trends such as;

growth of Internet of Things and connected vehicles, impact of autonomous vehicles, the rise of cloud computing, the need for advanced vehicle functionality and more.

The need of CAN FD increases and many experts assume it will slowly replace the classical CAN bus protocol. Stay updated to see what happens.