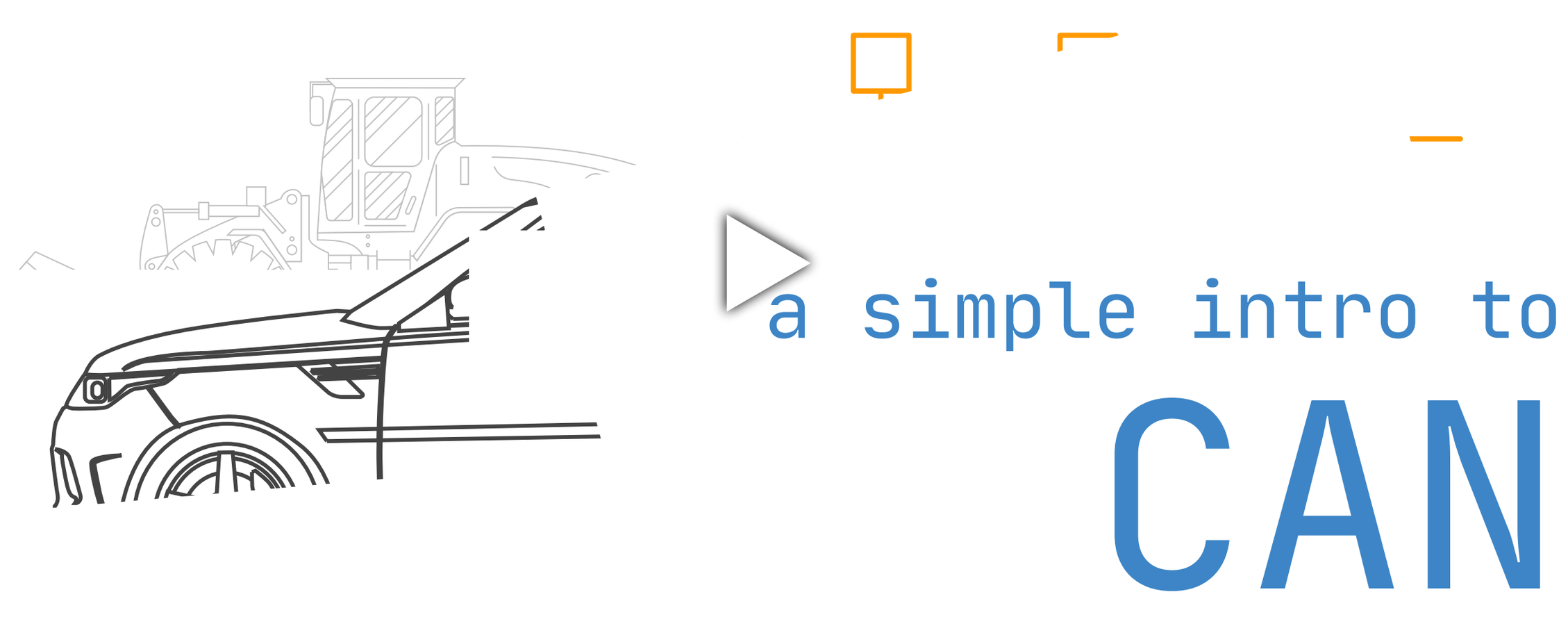
CAN Bus Explained - A Simple Intro [2023]



Need a simple, practical intro to CAN bus?

In this tutorial we explain the Controller Area Network (CAN bus) 'for dummies' incl. message interpretation, CAN logging - and the link to OBD2, J1939 and CANopen.

Read on to learn why this has become the **#1 guide on CAN bus**.

You can also view our CAN protocol intro above or get the PDF.

Author: Martin Falch in (updated April 2022)

Download as PDF

What is CAN bus?

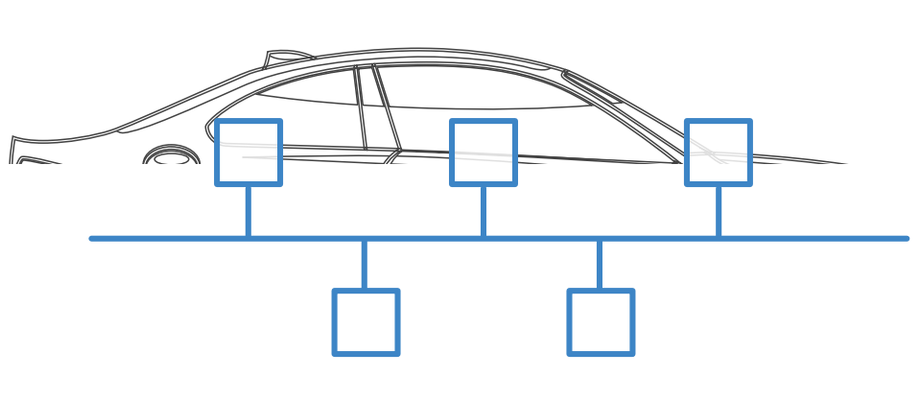
Your car is like a human body:

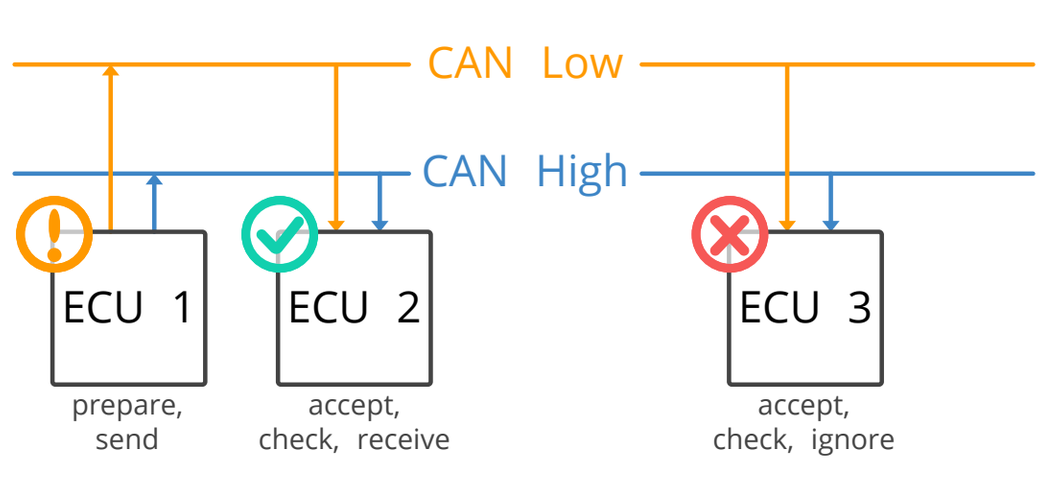
The Controller Area Network (CAN bus) is the **nervous system**, enabling communication.

In turn, 'nodes' or 'electronic control units' (ECUs) are like parts of the body, interconnected via the CAN bus. Information sensed by one part can be shared with another.

So what is an ECU?

In an automotive CAN bus system, ECUs can e.g. be the engine control unit, airbags, audio system etc. A modern car may have **up to 70 ECUs** - and each of them may have information that needs to be shared with other parts of the network.





This is where the CAN standard comes in handy:

The CAN bus system enables each ECU to communicate with all other ECUs - without complex dedicated wiring.

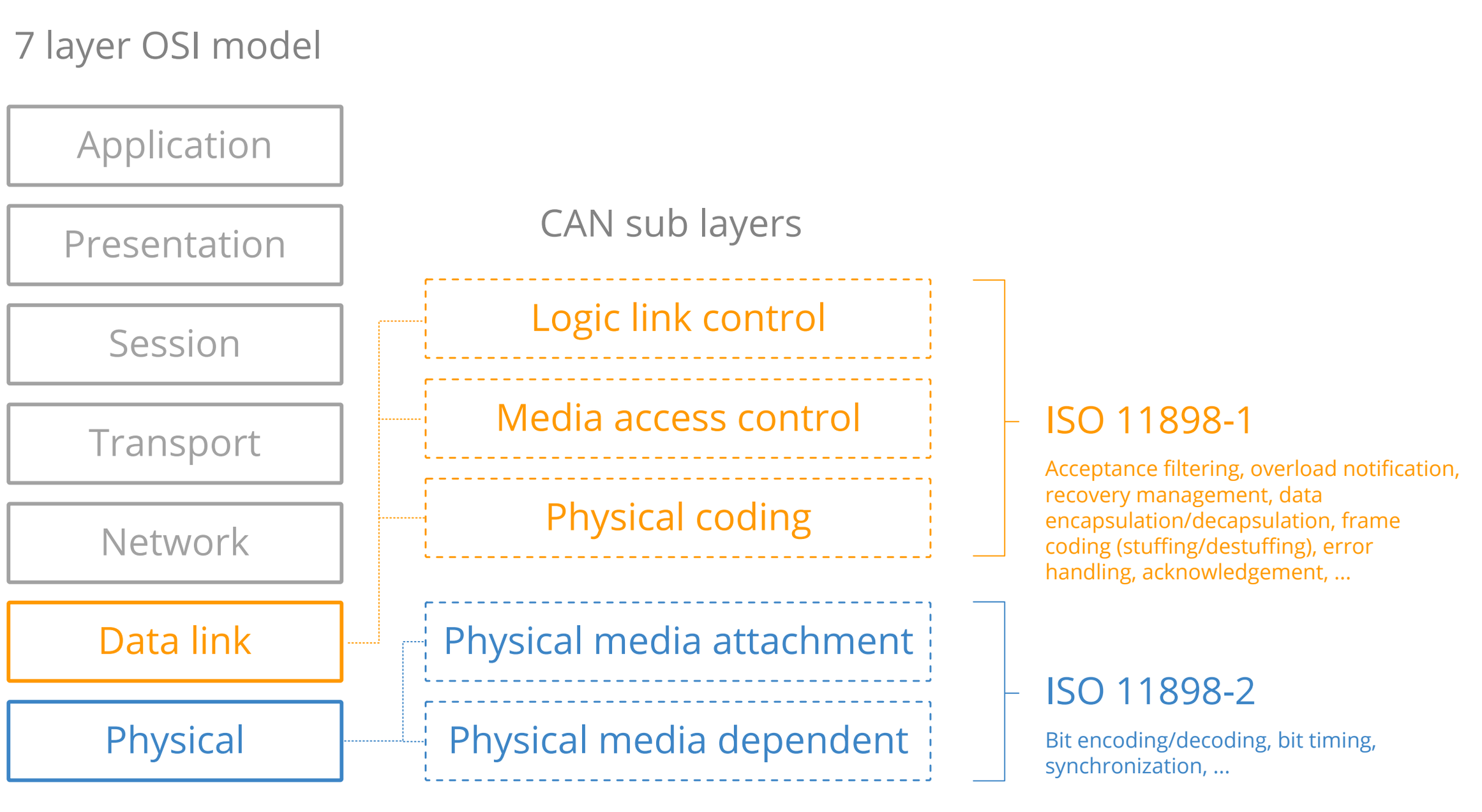
Specifically, an ECU can prepare and broadcast information (e.g. sensor data) via the CAN bus (consisting of two wires, CAN low and CAN high). The broadcasted data is accepted by all other ECUs on the CAN network - and each ECU can then check the data and decide whether to receive or ignore it.

CAN bus physical & data link layer (OSI)

In more technical terms, the controller area network is described by a data link layer and physical layer. In the case of high speed CAN, ISO 11898-1 describes the data link layer, while ISO 11898-2 describes the physical layer. The role of CAN is often presented in the 7 layer OSI model as per the illustration.

The CAN bus physical layer defines things like cable types, electrical signal levels, node requirements, cable impedance etc. For example, ISO 11898-2 dictates a number of things, including below:

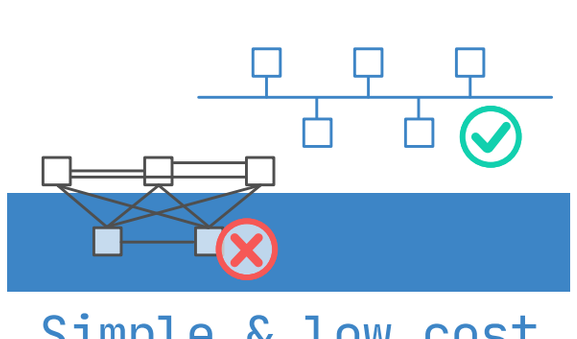
- **Baud rate:** CAN nodes must be connected via a two wire bus with baud rates up to 1 Mbit/s (Classical CAN) or 5 Mbit/s (CAN FD)
- **Cable length:** Maximal CAN cable lengths should be between 500 meters (125 kbit/s) and 40 meters (1 Mbit/s)
- **Termination:** The CAN bus must be properly terminated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



High speed CAN, low speed CAN, LIN bus, ... +

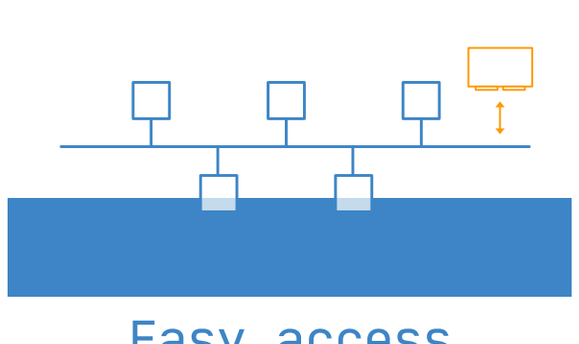
Top 4 benefits of CAN bus

The CAN bus standard is used in practically all vehicles and many machines due to below key benefits:



Simple & low cost

ECUs communicate via a single CAN system instead of via direct complex analogue signal lines - reducing errors, weight, wiring and costs

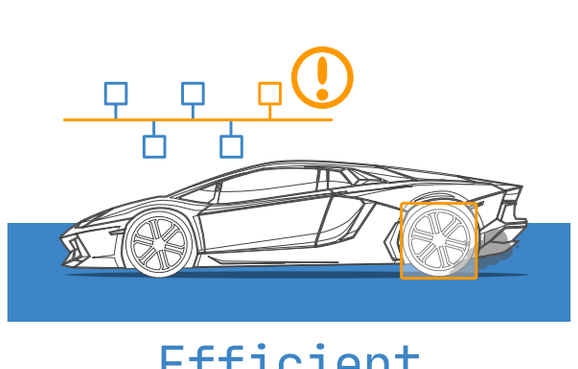


Easy access

The CAN bus provides 'one point-of-entry' to communicate with all network ECUs - enabling central diagnostics, data logging and configuration

Extremely robust

The system is robust towards electric disturbances and electromagnetic interference - ideal for safety critical applications (e.g. vehicles)



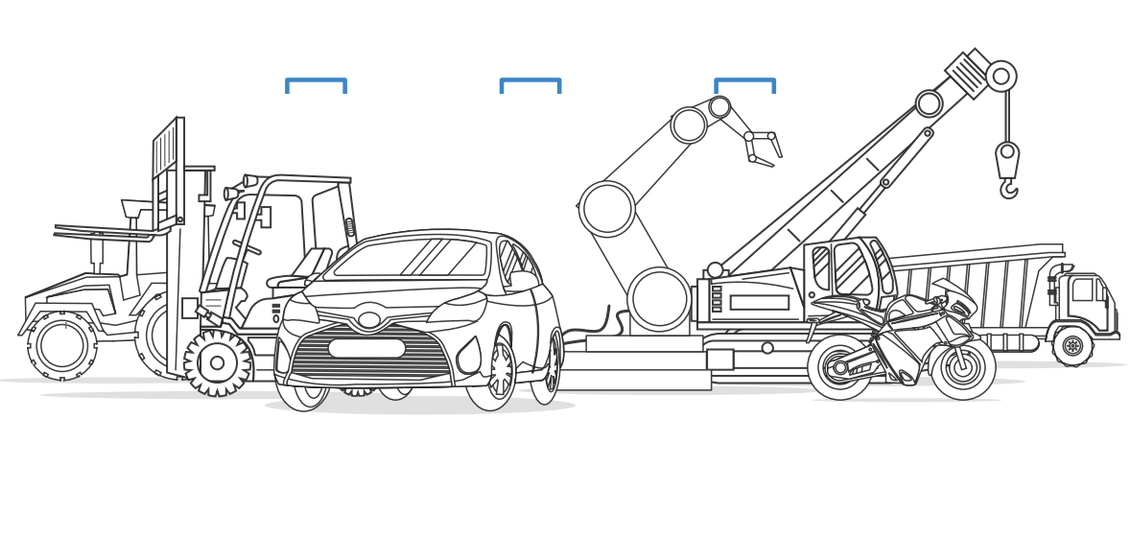
Efficient

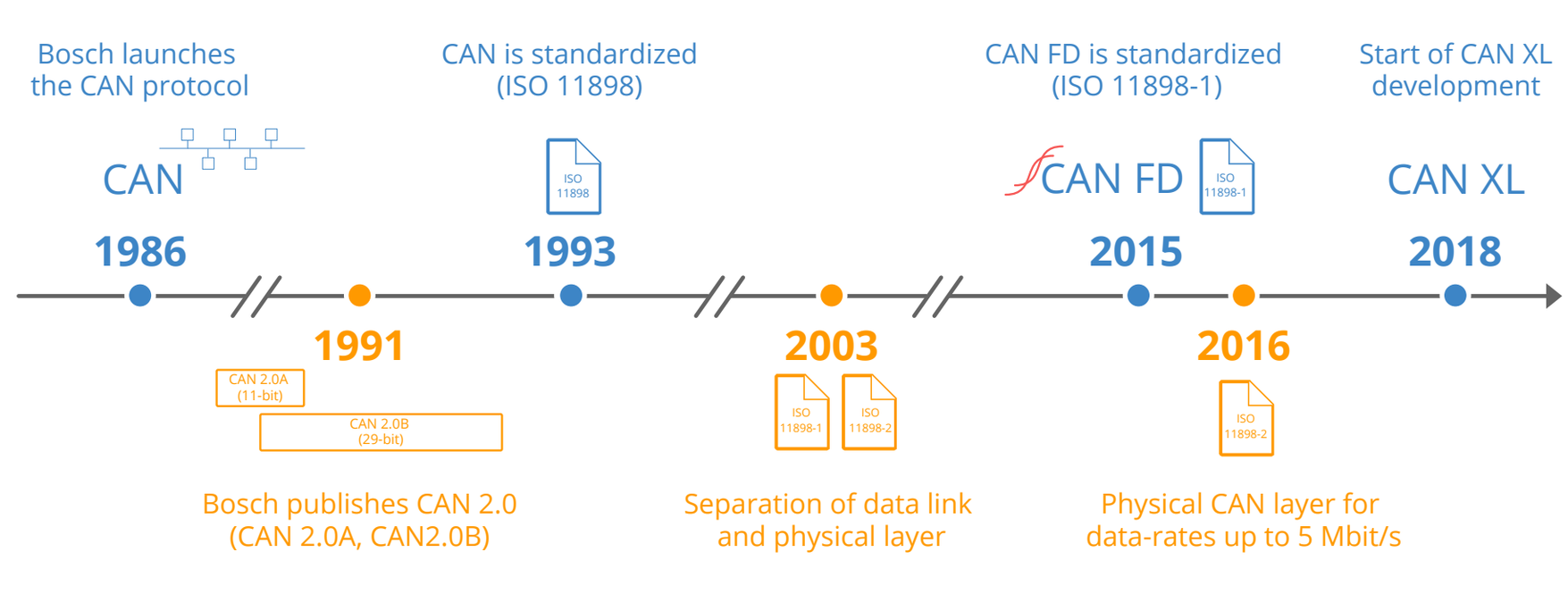
CAN frames are prioritized by ID so that top priority data gets immediate bus access, without causing interruption of other frames or CAN errors

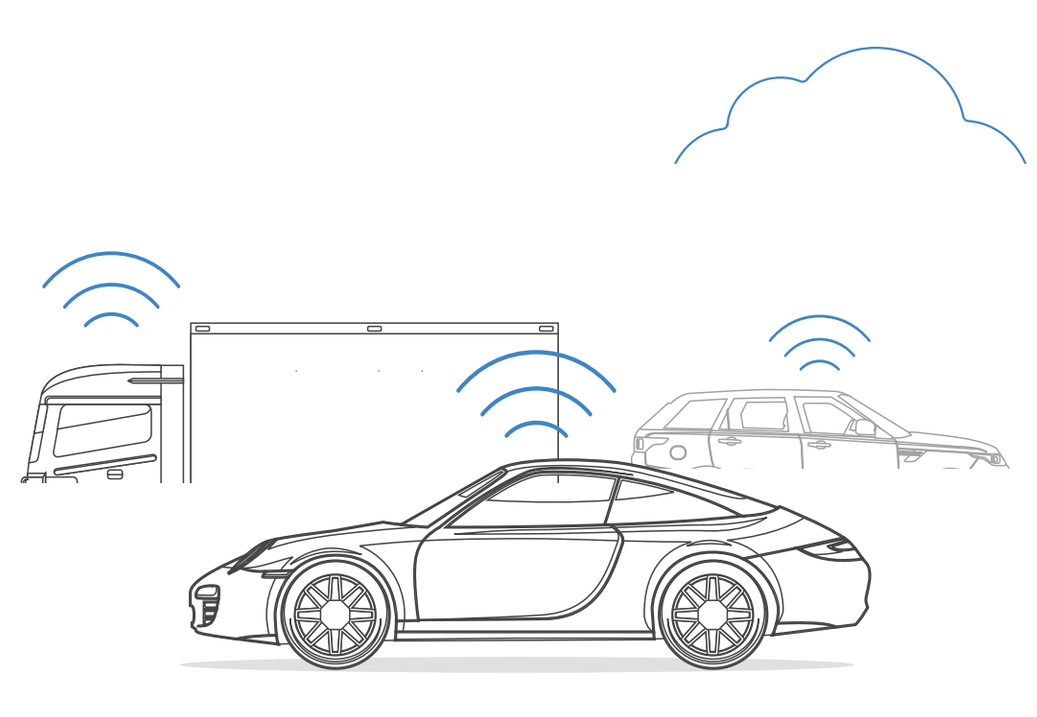
The CAN bus history in short

- **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 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

Today, CAN is standard in automotives (cars, trucks, buses, tractors, ...), ships, planes, EV batteries, machinery and more.







The future of CAN bus

Looking ahead, the CAN bus protocol will stay relevant - though it will be impacted by **major trends**:

- A need for increasingly advanced vehicle functionality
- The rise of cloud computing
- Growth in Internet of Things (IoT) and connected vehicles
- The impact of autonomous vehicles

In particular, the rise in connected vehicles (V2X) and cloud will lead to a rapid growth in vehicle telematics and IoT CAN loggers.

In turn, bringing the CAN bus network 'online' also exposes vehicles to security risks - and may require a shift to new CAN protocols like CAN FD.

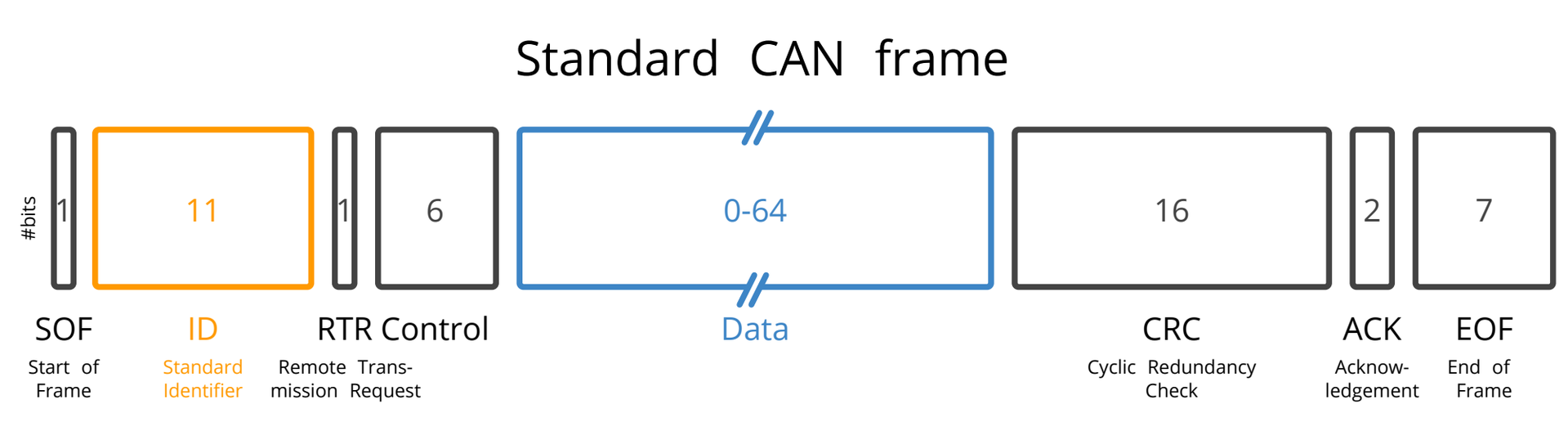
The rise of CAN FD +

What is a CAN frame?

Communication over the CAN bus is done via CAN frames.

Below is a standard CAN frame with 11 bits identifier (CAN 2.0A), which is the type used in most cars. The extended 29-bit identifier frame (CAN 2.0B) is identical except the longer ID. It is e.g. used in the J1939 protocol for heavy-duty vehicles.

Note that the CAN ID and Data are highlighted - these are important when recording CAN bus data, as we'll see below.

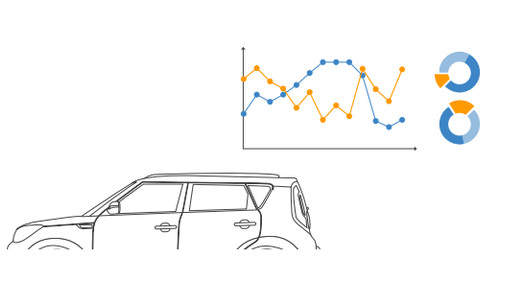


The 8 CAN bus protocol message fields +

CAN bus errors +

Logging CAN data - example use cases

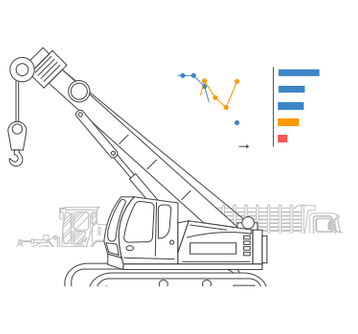
There are several common use cases for recording CAN bus data frames:



Logging/streaming data from cars

OBD2 data from cars can e.g. be used to reduce fuel costs, improve driving, test prototype parts and insurance

obd2 logging →



Heavy duty fleet telematics

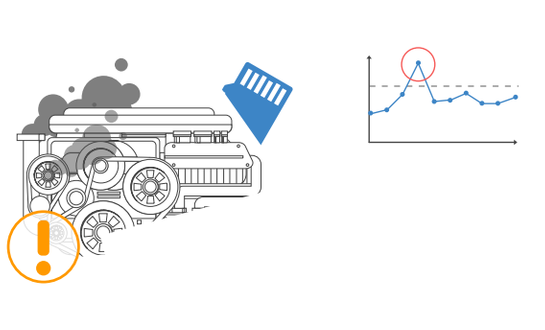
J1939 data from trucks, buses, tractors etc. can be used in fleet management to reduce costs or improve safety

j1939 telematics →

Predictive maintenance

Vehicles and machinery can be monitored via IoT CAN loggers in the cloud to predict and avoid breakdowns

predictive maintenance →



Vehicle/machine blackbox

A CAN logger can serve as a 'blackbox' for vehicles or equipment, providing data for e.g. disputes or diagnostics

can bus blackbox →

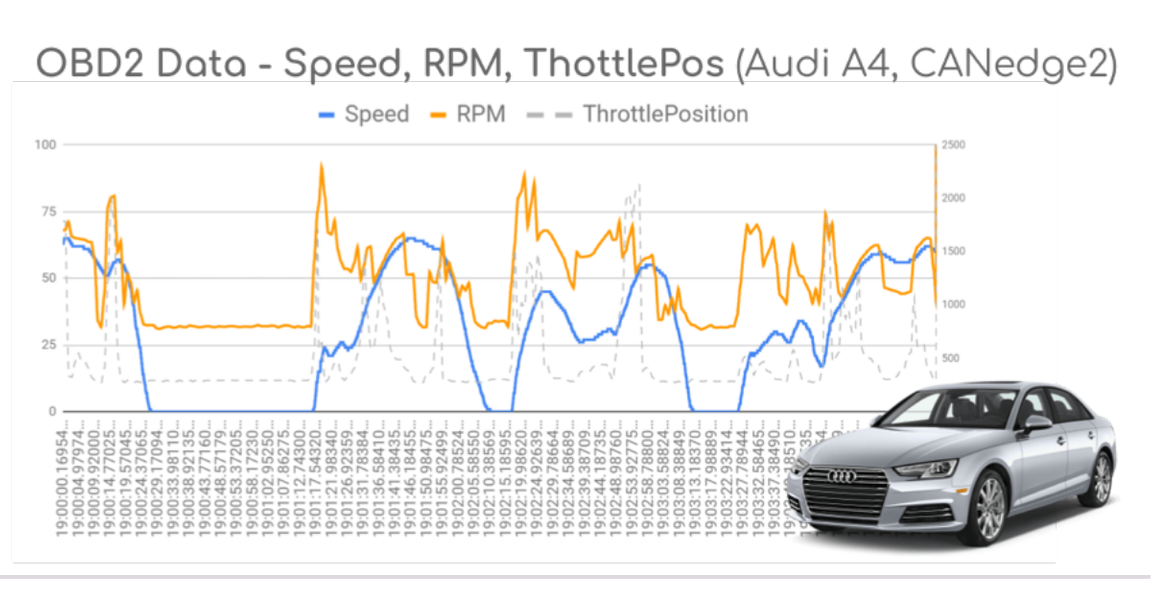
Do you have a CAN logging use case? Reach out for free sparring!

contact us →

How to log CAN bus data

As mentioned, two CAN fields are important for CAN logging: The **CAN ID** and the **Data**.

To record CAN data you need a **CAN logger**. This lets you log timestamped CAN data to an SD card. In some cases, you need a **CAN interface** to stream data to a PC - e.g. for car hacking.



Connecting to the CAN bus +

Example: Raw CAN sample data (J1939) +

Example: CANedge CAN logger

The CANedge1 lets you easily record data from any CAN bus to an 8-32 GB SD card. Simply connect it to e.g. a car or truck to start logging - and decode the data via free software/APIs.

Further, the CANedge2 adds WiFi, letting you auto-transfer data to your own server - and update devices over-the-air.

learn about the canedge

How to decode raw CAN data to 'physical values'

If you review the raw CAN bus data sample above, you will probably notice something:

Raw CAN bus data is not human-readable.

To interpret it, you need to **decode the CAN frames into scaled engineering values** aka physical values (km/h, degC, ...).

Below we show step-by-step how this works:

Extracting CAN signals from raw CAN frames +

The challenge of proprietary CAN data +

CAN database files (DBC) - J1939 example +

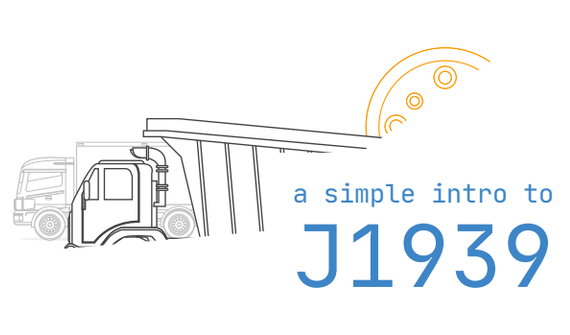
Example: Decoded CAN sample data (physical values) +

What is the link between CAN, J1939, OBD2, CANopen, ...?

The Controller Area Network provides the basis for communication - but not a lot more.

For example, the CAN standard does not specify how to handle messages larger than 8 bytes - or how to decode the raw data. Therefore a set of **standardized protocols** exist to further specify how data is communicated between CAN nodes of a given network.

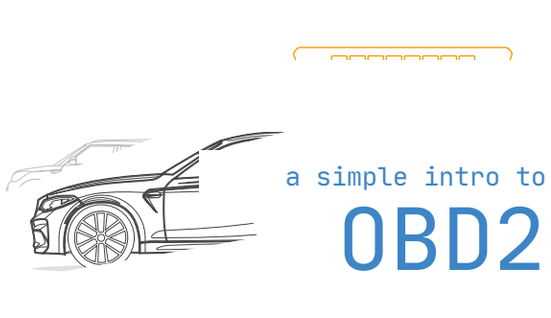
Some of the most common standards include SAE J1939, OBD2 and CANopen. Further, these higher-layer protocols will increasingly be based on the 'next generation' of CAN, CAN FD (e.g. CANopen FD and J1939-17/22).



SAE J1939

J1939 is the standard in-vehicle network for heavy-duty vehicles (e.g. trucks & buses). J1939 parameters (e.g. RPM, speed, ...) are identified by a suspect parameter number (SPN), which are grouped in parameter groups identified by a PG number (PGN).

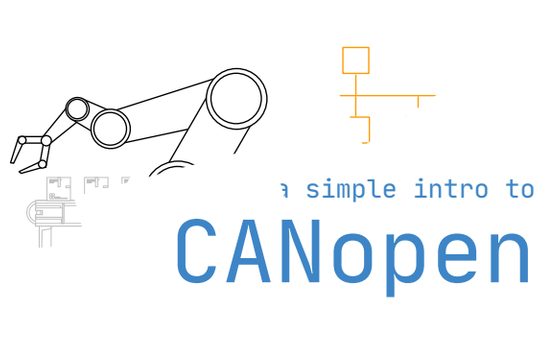
j1939 intro →
j1939 telematics →



OBD2

On-board diagnostics (OBD, ISO 15765) is a self-diagnostic and reporting capability that e.g. mechanics use to identify car issues. OBD2 specifies diagnostic trouble codes (DTCs) and real-time data (e.g. speed, RPM), which can be recorded via OBD2 loggers.

obd2 intro →
obd2 logging →



CANopen

CANopen is used widely in embedded control applications, incl. e.g. industrial automation. It is based on CAN, meaning that a CAN bus data logger is also able to log CANopen data. This is key in e.g. machine diagnostics or optimizing production.

canopen intro →
canopen logger →

CAN FD

CAN bus with flexible data-rate (CAN FD) is an extension of the Classical CAN data link layer. It increases the payload from 8 to 64 bytes and allows for a higher data bit rate, dependent on the CAN transceiver. This enables increasingly data-intensive use cases like EVs.

can fd intro →
can fd logger →

Other CAN based protocols +

For more intros, see our guides section - or download the **'Ultimate Guide'** PDF.

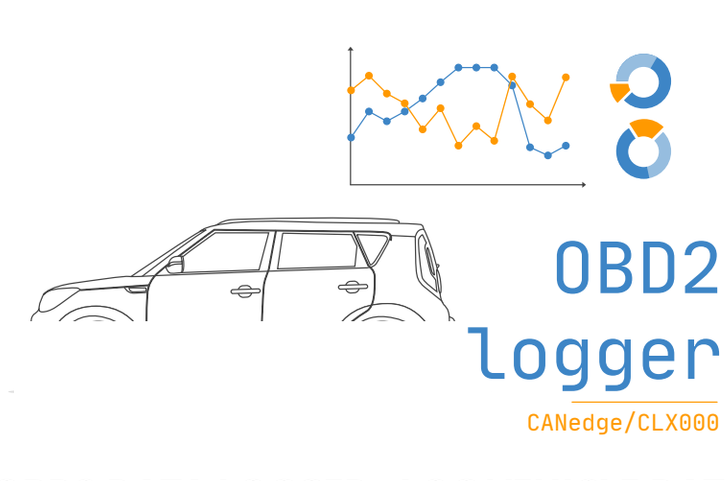
Need to log/stream CAN bus data?

Get your CAN logger today!

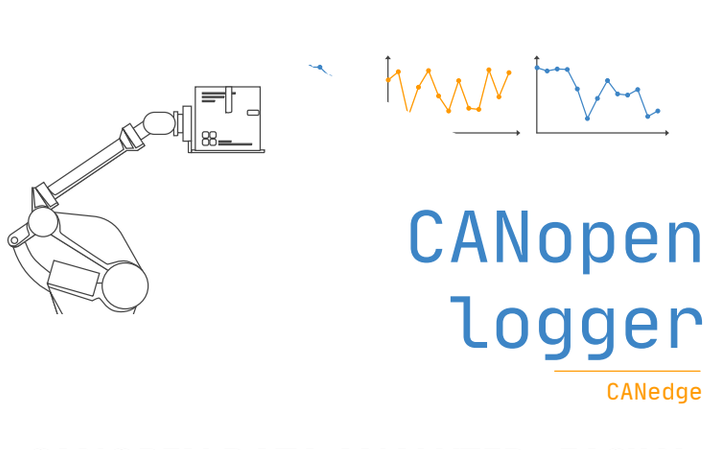
buy now contact us →

Recommended for you

J1939 DATA LOGGER - SIMPLE TELEMATICS



OBD2 DATA LOGGER - LOG VEHICLE DATA EASILY



CANOPEN DATA ANALYZER - EASILY ANALYZE MACHINES