

[Products
About us](#)[Applications
Careers](#)[Support
Training](#)

What Is CAN Bus (Controller Area Network) and How It Compares to Other Vehicle Bus Networks

19. Feb 2021

We're here to help!

NEWS CATEGORIES

Written by *Grant Maloy Smith*, the data acquisition expert



All categories

Dewesoft Corporate News

Dewesoft Events

DewesoftX Software News

Product News

Application Notes

Case Studies

Data Acquisition

In this article we will discuss the CAN bus (Controller Area Network) and other vehicle bus interfaces so you will be able to:

See what CAN bus really is

Learn about the background and future of CAN bus systems

Understand how Dewesoft data acquisition systems interface with CAN bus

Jump to a section

[What Is the CAN Bus Protocol?](#)

[What Is the CAN FD?](#)

[Advantages of CAN bus](#)

[Popular CAN Bus Applications](#)

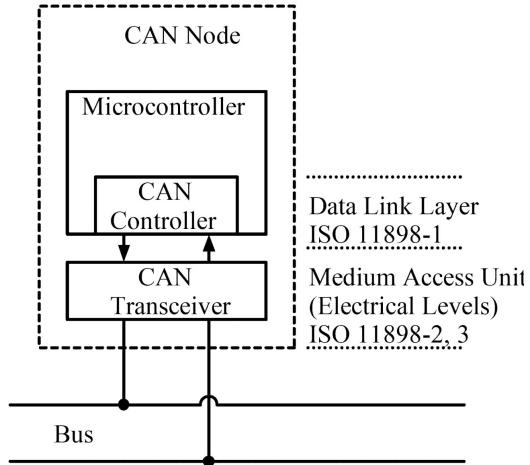
[A Brief History of CAN Bus](#)

[How Does CAN Messaging Work?](#)

[\(DAQ\)](#)[Authors](#)[Variations of the CAN Bus](#)[Additional CAN Standards and Protocols](#)[Related Communication Buses](#)[Comparing CAN with Other Vehicle Buses](#)[Dewesoft CAN Bus DAQ Systems](#)[More Reading](#)

What Is the CAN Bus Protocol?

The **Controller Area Network - CAN bus** is a message-based protocol designed to allow the Electronic Control Units (ECUs) found in today's automobiles, as well as other devices, to communicate with each other in a reliable, priority-driven fashion. Messages or "frames" are received by all devices in the network, which does not require a host computer. CAN is supported by a rich set of international standards under [ISO 11898](#).



CAN bus network schematic

What Is the CAN FD?

CAN FD is a “Flexible Data (Rate)” version of the CAN bus. The standard length of each message has been increased 800% to 64 bytes, and the maximum data rate has been similarly increased from 1 Mbps to 8 Mbps. The “flexible” part refers to the fact that ECUs can dynamically change their transmission rates and select larger or smaller message sizes, based on real-time requirements.

Despite all of these advances, CAN FD is still completely backwardly compatible with standard CAN 2.0. Today, CAN

FD is found in very high-performance vehicles, but it is expected to migrate across all or most vehicles eventually.

This video provides excellent background information about vehicle data buses, including CAN:

Advantages of CAN bus

The CAN bus standard is widely accepted and is used in practically all vehicles and many machines. This is mainly due to below key benefits:

Simple and Low Cost: ECUs communicate via a single CAN system instead of via direct complex analog signal lines - reducing errors, weight, wiring, and costs. CAN chipsets are readily available and affordable.

Fully Centralized: 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 numbers. The top priority data gets immediate bus access, without causing interruption of other frames.

Reduced Vehicle Weight: by the elimination of kilometers of heavily insulated electrical wires and their weight from the vehicle.

Easy Deployment: a proven standard with a rich support ecosystem.

Resistant to EMI: this makes CAN ideal for critical applications in vehicles.

CAN has excellent control and fault detection capabilities. Detecting an error is easily done, and thus transmitted data gets to where it needs to go.

It is an ideal protocol when distributed control of a complex system is required. It reduces heavy wiring and thus cost and weight. The cost of the chips is low, and implementing CAN is relatively easy because of the clean design of the protocol.

Another advantage to using CAN is that the first two layers: the physical layer and the data link layer, are implemented in inexpensive microchips, available in several configurations.

Popular CAN Bus Applications

Today, applications for CAN are dominated by the automotive and motor vehicle world, but they are not limited to that. CAN is found across virtually every industry. You can find the CAN protocol being used in:

Every kind of vehicle: motorcycles, automobiles, trucks...

Heavy-duty fleet telematics

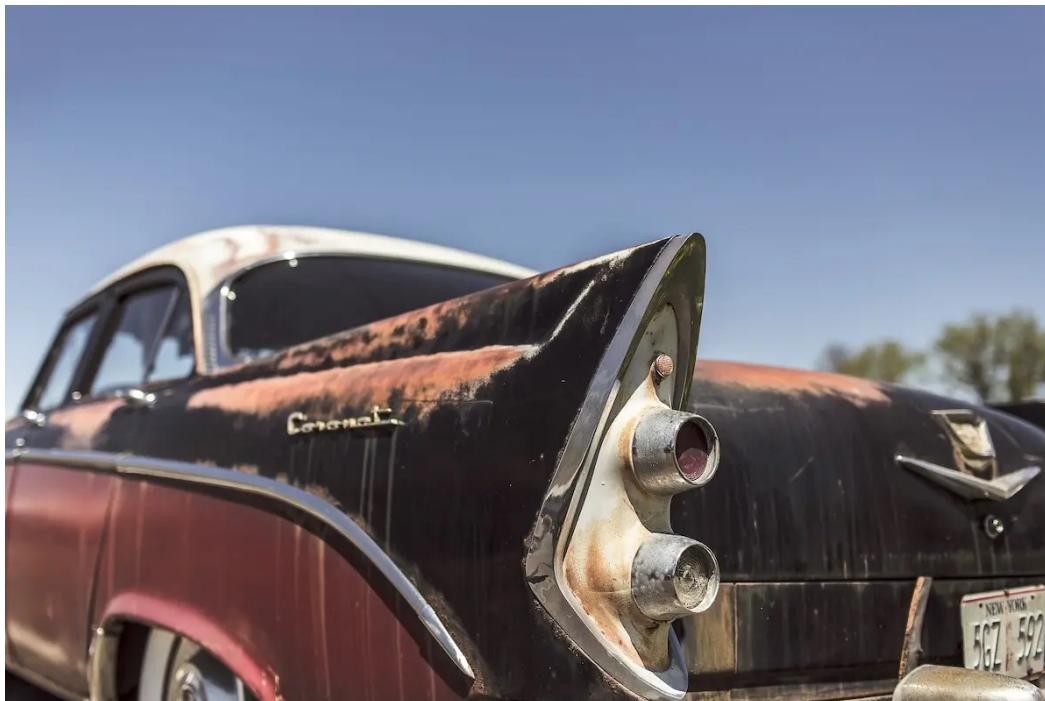
- Airplanes
- Elevators
- Manufacturing plants of all kinds
- Ships
- Medical equipment
- Predictive maintenance systems
- Washing machines, dryers, and other household appliances.

A Brief History of CAN Bus

When you flip a switch in your house to turn on the lights, electricity flows through the switch to the lights. As a result, the switches and wiring need to be heavy and insulated enough to handle the maximum expected current. The walls of your house are filled with this heavy, insulated wiring.

Cars and trucks used to be wired exactly the same way: ever since Henry Ford got the idea to add lights and an electric horn to his cars in 1915, electricity flowed from the battery through switches to the lights and other devices. By the 1960s there were thousands of heavy wires running throughout every vehicle. Every bit of extra weight reduces a

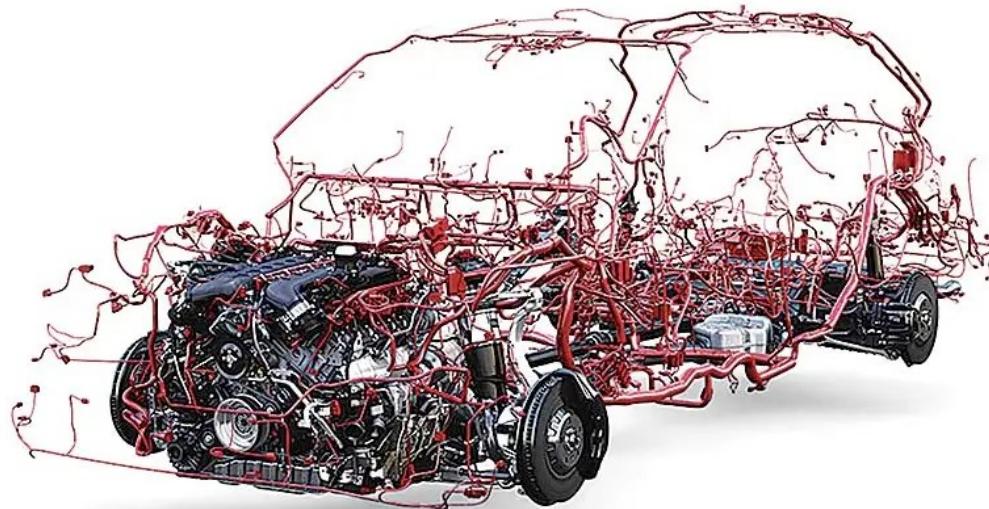
vehicle's fuel efficiency.



Pre-CAN bus car with miles/kilometers of heavy wires inside.

Copyright Ryan McGuire from Pixabay.

Following the oil embargoes of the 1970s, there was increasing pressure on automobile manufacturers to improve their fuel efficiency. So they started looking for ways to reduce the weight of the cars they were making.

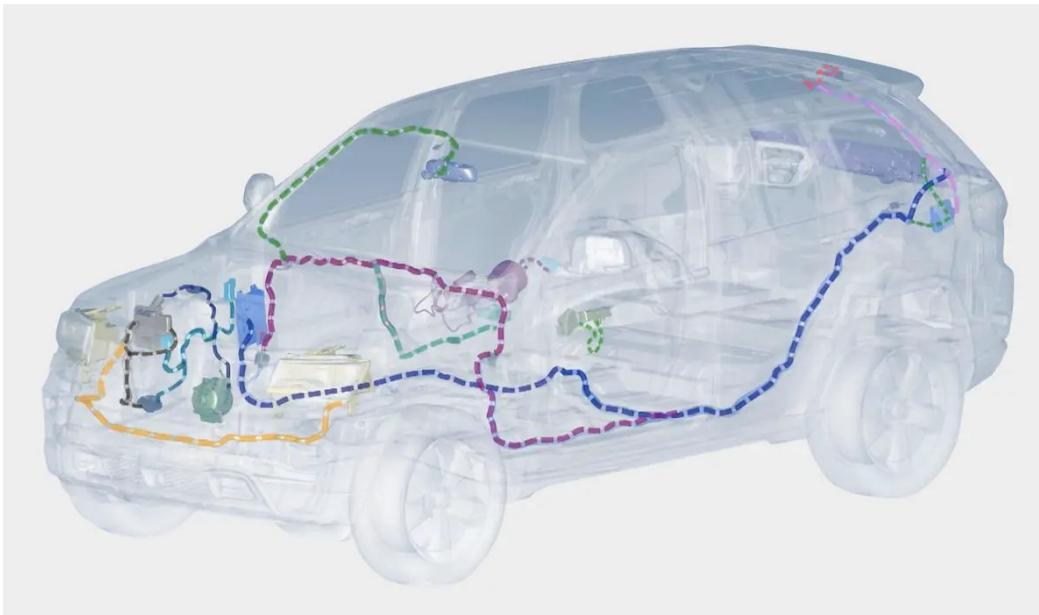


Typical electrical wiring in a passenger car

Picture courtesy of Transparency Market Research

By the early 1980s, cars had more and more **ECUs** (electronic control units) inside them, and companies like **Robert Bosch** company of Germany were looking for a type of bus communication system that could be used as a communication system between multiple ECUs and vehicle systems. They searched the market but couldn't find exactly what they needed, so they began developing the "Controller Area Network" in partnership with automobile manufacturer Mercedes-Benz and semiconductor maker Intel®, and several universities in Germany.

In 1986, Bosch introduced the CAN standard at the SAE Congress in Detroit. One year later Intel Corporation began shipments of the first CAN controller chips, and the automotive world was changed forever. Looking back, the weight savings that resulted from the development of CAN were almost a lucky by-product, but very real nonetheless.



The heavy cable is replaced with lightweight 2-wire CAN in today's cars and trucks

How Does CAN Messaging Work?

Devices on a CAN bus are called “nodes.” Each node

consists of a CPU, CAN controller, and a transceiver, which adapts the signal levels of both data sent and received by the node. All nodes can send and receive data, but not at the same time.

Nodes cannot send data directly to each other. Instead, they send their data onto the network, where it is available to any node to which it has been addressed. The CAN protocol is lossless, employing a bitwise arbitration method to resolve contentions on the bus.

All of the nodes are synchronized so that they all sample data on the network simultaneously. However, data is not transmitted with clock (time) data, so CAN is not truly a synchronous bus, such as **EtherCAT**, for example.

With CAN, all data are sent in frames, and there are four types:

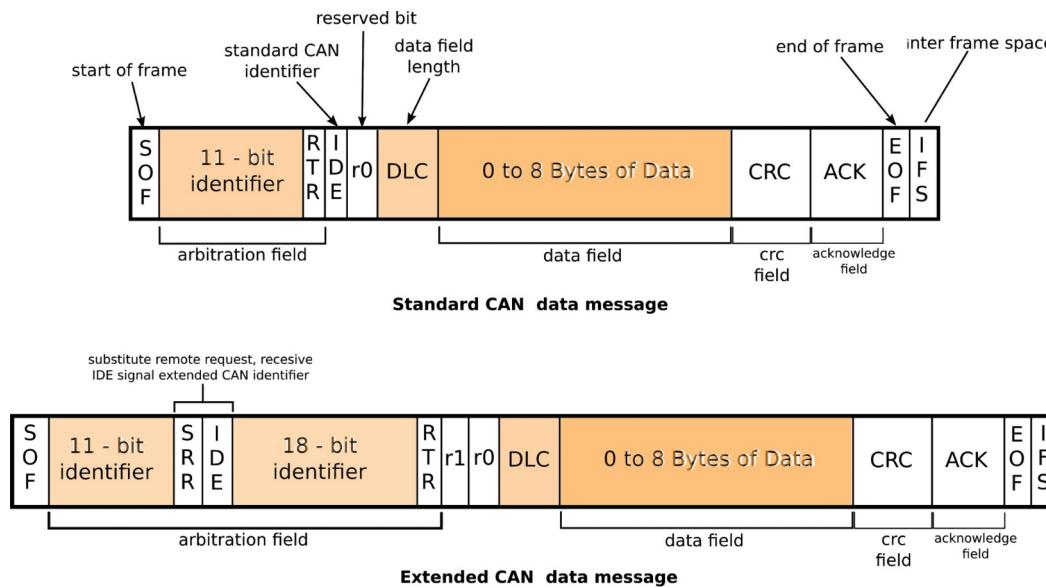
Data frames transfer data to one or many receiver nodes

Remote frames ask for data from other nodes

Error frames report errors

Overload frames report overload conditions

There are two variants of the message length: standard and extended. The real difference is the additional 18-bit identifier in the arbitration field.



Standard and Extended frame of the CAN data message architecture

CAN Data Message Structure (CAN Frame)

Field	Bits	Description
SOF	1	The single dominant Start Of Frame. This bit marks the start of a message. It synchronizes the nodes after an idle period.
Identifier	11	The CAN 11-bit identifier data field sets the message

Field	Bits	Description
		priority. Lower values mean higher priorities.
RTR	1	Remote Transmission Request. This bit is dominant when information is requested by another node. All nodes will receive the request, but the identifier determines the desired node.
IDE	1	The Identifier Extension bit indicates that a standard CAN identifier (not an extended one) is being transmitted.
R0	1	Reserved for future use.
DLC	4	The Data Length Code contains the number of bytes in the transmission.
Data	0 - 64	The actual data being transmitted.
CRC	16	The 16-bit (15 bits plus delimiter) cyclic redundancy check (CRC) contains the checksum (number of bits transmitted) of the preceding application data for transmitting error detection.

Field	Bits	Description
ACK	2	When a node successfully receives a message, it ACKnowledges it by overwriting this bit with a dominant bit. On the other hand, if a node finds an error in a message, it allows this bit to remain recessive and ignores the message. ACK slot and ACK delimiter are each one bit long.
EOF	7	End Of Frame is a 7-bit field that denotes the end of every CAN frame (message).
IFS	3+	Inter Frame Space (IFS) is the time that the controller needs to move a frame (message) into position in the buffer area. Note that IFS contains a minimum of three consecutive recessive (1) bits. After three recessive bits have passed, when a dominant bit is detected, it becomes the SOF bit of the next frame.

Looking closer the bit fields of CAN data transmit messages

The arbitration field contains the message identification number and remote transmission request bit. More important messages have lower ID numbers.

If multiple nodes transmit at the same time, they start a simultaneous arbitration. The node with the lowest message ID number gets priority. Dominant bits overwrite

recessive bits on the CAN bus.

The message identifier can be **11-bit** (Standard CAN, 2048 different message identifiers) or 29 bit in length (Extended CAN, 537 million different message identifiers). The remote transmission request bit is dominant and indicates that data is being transmitted.

In most systems, logical 1 represents a high, and logical 0 represents a low. But this is the other way around on the CAN bus. CAN transceivers therefore typically use a pull-up on the driver inputs and receiver outputs, so that devices have defaulted to a recessive bus state.

Variations of the CAN Bus

The **ISO 11898 standard** defines several versions of CAN. The dominant CAN types used within the automobile industry are:

Low Speed CAN

High Speed CAN

CAN FD (Flexible Data Rate CAN)

Low Speed CAN

Used for fault-tolerant systems that do not require high update rates. The maximum data transfer rate is 125 kbps, but the wiring is thus more economical than high-speed CAN. In automotive applications, low-speed CAN is used for diagnostics, dashboard controls and displays, power windows, etc.

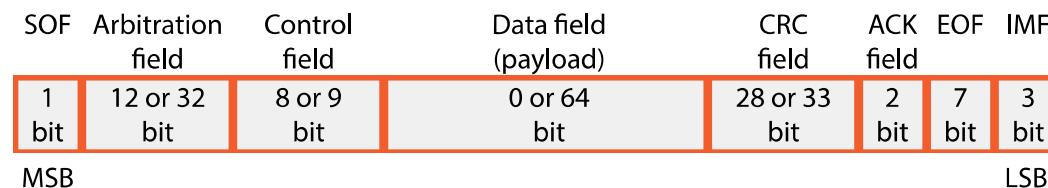
High Speed CAN

Used for communications between critical subsystems that require high update rates and high data accuracy (e.g., anti-lock braking system, electronic stability control, airbags, engine control units, etc). Data transfer speeds of high-speed CAN ranges from 1 kbit to 1 Mbit per second.

High-speed CAN is faster than low speed, but the bandwidth requirement of new automotive applications is increasing every year, so automobile OEMs are now installing CAN FD into new cars. CAN FD has been described tongue-in-cheek as “CAN on steroids.”

CAN FD (Flexible Data Rate CAN)

The latest version of CAN introduces a flexible data rate, more data per message, and much higher speed transmissions. The data length within each standard (low speed and high speed) CAN message is 8 bytes, but with **CAN FD** this has been increased 800% to 64 bytes of data. In addition, the maximum data rate has also been increased dramatically from 1 Mbps to 8 Mbps.



CAN FD data frame format

CAN FD is also backwardly-compatible and supports the CAN 2.0 communication protocol as well as special protocols such as **SAE J1939**, where CAN out is used as read-only. CAN FD is essentially an extension of the original CAN standard as specified in ISO 11898-1, and is fully compatible with classical CAN systems.

CAN FD is an important step forward because it allows ECUs to dynamically change their transmission rates and select larger or smaller message sizes, based on real-time requirements. It is found now in high-performance vehicles,

but as ECU performance rises and CAN FD hardware costs fall, it is only a matter of time before CAN FD makes its way into virtually all vehicles.

Many Dewesoft products have low/high speed CAN bus interfaces built directly into them, including SIRIUS (and SIRIUS-based instruments including the R1, R2, R3, R4, R8, R8rt), DEWE-43A, and the MINITAURs. These models all include one CAN bus, except for the DEWE-43A which has two. Additional CAN bus interfaces can be added to any Dewesoft system using the 1, 2, 4, and even 8 port interfaces that are available.



CAN bus interfaces are available in almost every Dewesoft DAQ system

If CAN FD is required, **KRYPTONi-1xCAN-FD** is a single port CAN FD device that uses EtherCAT as a data interface. It supports high-speed CAN interfaces with data rates up to 8 Mbps. In addition, CAN FD supports the CAN2.0 communication protocol as well as special protocols such as J1939, where CAN out is used as read-only. The KRYPTONi-1xCAN-FD uses galvanically isolated communication lines and an isolated sensor supply of +5 V and +12 V. Power limit for the sensor supply is 1.4 W.



The rugged and water-proof KRYPTONi-1xCAN-FD with EtherCAT interface

This very small CAN FD module can be added to any Dewesoft DAQ instrument that has an EtherCAT port, which

includes the entire **SIRIUS line**, and of course, the **KRYPTON line** itself.

Learn more:

[Dewesoft CAN interfaces](#)

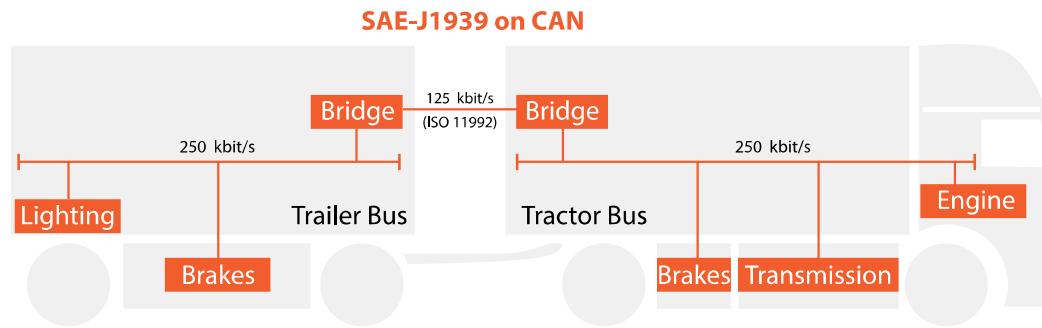
Additional CAN Standards and Protocols

Why do we need additional standards and protocols “on top” of CAN? It’s simply because while CAN is an elegant and reliable protocol, that’s really all it is. It is a messaging system, but it does not include any way to analyze or understand the data within the messages. This is why several companies have created additional standards and protocols that run within or on top of CAN, providing additional functionalities. The best known of these include:

SAE J1939 on CAN

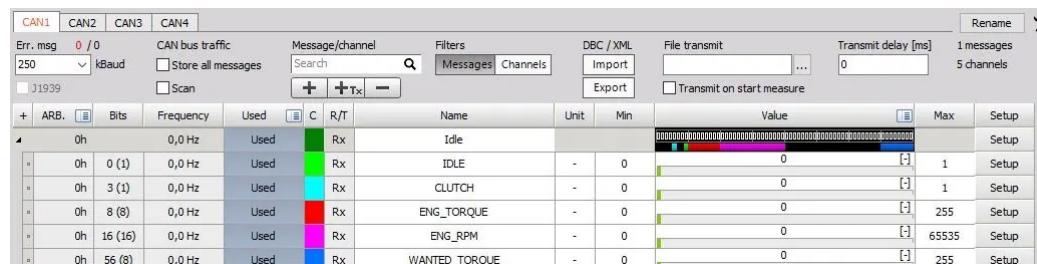
The SAE J1939 protocol was developed originally to be

used by heavy trucks and tractor-trailer rigs in the USA. Today it is used by diesel engine makers all over the world. J1939 is a higher-level protocol that runs on the CAN physical layer. It provides some useful functions specific to heavy trucks such as 18-wheel trucks.



SAE on CAN schematic

The protocol has a few restrictions that were intentionally put in place to promote the highest possible reliability, including limiting the message identifier to 29-bits and limiting the bus speed to either 250 or 500 kbps.



CAN bus setup screen in DewesoftX software. Notice the J1939 checkbox near the top-left.

DewesoftX software allows the engineer to select J1939 decoding via a checkbox on the CAN setup screen for any available CAN port. Of course, this assumes that the messages on the CAN bus are formatted according to the J1939 standard. Data messages are the same length as the extended CAN standard.

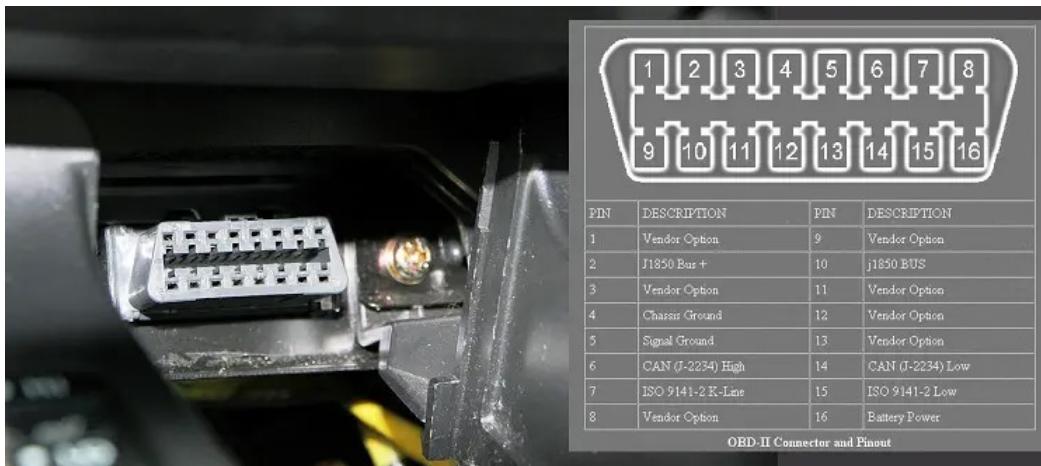
The arbitration field contains an additional source and destination address, and the baud rate is limited to 250 kbps or 500 kbps, depending on the J1939 standard version being used. J1939 is a selection on the standard Dewesoft X CAN setup screen - no additional hardware or software are required.

Learn more:

[Setting up CAN, including J1939 data, with DewesoftX](#)

OBD II (aka “OBD 2”)

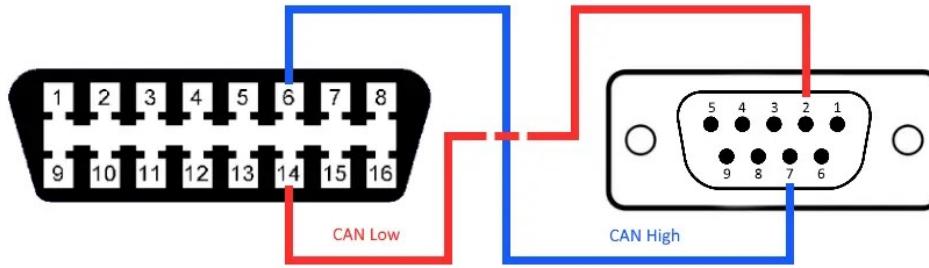
This on-board diagnostics port is found in all cars made since 1989. Usually located within 2 feet (0.61 meters) of the steering wheel, it's an interface that allows automobile repair shops as well as vehicle owners to diagnose vehicle problems by connecting a scanning tool to its 16-pin connector. (Pictured here under the steering wheel in the 2016 Toyota 4Runner).



OBD II connector on a vehicle

Scanning tools can read the DTC (diagnostic trouble codes) reported by the vehicle. The **OBD II interface** is required to carry dozens of channels of real-time data, such as RPM, vehicle speed, coolant temperature, and more. Dewesoft CAN interfaces can be connected to this OBD II connector as shown below, and can read out, display, and record any

or all of these channels in sync with the other data being recorded.



OBD II connector (left) connected to a Dewesoft CAN interface connector (right)

Test: Can 1														
Message / Signal		Scan:	Import / Export messages		Vehicle identification number		Diagnostic trouble codes							
Add	Remove	<input type="checkbox"/> Scan all messages	<input type="checkbox"/> Import	<input type="checkbox"/> Export	IVPOZZ29ZT5391223		Show DTCs							
PID	Mode	DLC	Start bit	Bit length	Date type	Frequency	Sending freq.	Receiving freq.	Supported	Used	C	Name	Description	Units
4	1	1	0	8	Unsigned	5	5,00	5,00	yes	Used		Calculated engine load	-	%
5	1	1	0	8	Unsigned	5	5,00	5,00	yes	Used		Engine coolant temperature	-	°C
6	1	1	0	8	Unsigned	5	0,00	0,00	yes	Used		Short term fuel trim -B1	-	%
7	1	1	0	8	Unsigned	5	0,00	0,00	yes	Used		Long term fuel trim -B1	-	%
8	1	1	0	8	Unsigned	5	0,00	0,00	yes	Used		Short term fuel trim -B2	-	%
9	1	1	0	8	Unsigned	5	0,00	0,00	yes	Used		Long term fuel trim -B2	-	%
10	1	1	0	8	Unsigned	5	1,00	5,00	yes	Used		Fuel pressure	-	psi
11	1	1	0	8	Unsigned	5	0,00	0,00	yes	Used		Intake manifold absolute pressure	-	psi
12	1	2	0	16	Unsigned	5	5,00	5,00	yes	Used		Engine RPM	-	rpm
13	1	1	0	8	Unsigned	5	5,00	5,00	yes	Used		Vehicle speed	-	kph
14	1	1	0	8	Unsigned	5	0,00	0,00	yes	Used		Timing advance	-	°A
15	1	1	0	8	Unsigned	5	5,00	5,00	yes	Used		Intake air temperature	-	°C
16	1	2	0	16	Unsigned	5	0,00	0,00	yes	Used		MAP air flow rate	-	grams/sec

Just part of the ODB II setup screen in DewesoftX software

Decoding, displaying, and recording ODB II messages in Dewesoft systems require an additional ODB II software plugin. You can scan DTC (diagnostic trouble codes) and much more with this system.

Learn more:

[ODB II in DewesoftX software](#)

XCP/CCP on CAN and Ethernet

The **Universal Measurement and Calibration Protocol**

([XCP](#)) was designed to connect ECUs to calibration systems. The “universal” in its name refers to the fact that it can run on top of the CAN bus, CAN FD, FlexRay, Ethernet, and more. It is the successor to the original CAN Calibration Protocol (CCP) developed in the 1990s.

Dewesoft supports XCP/CCP protocols via [XCP/CCP Master](#) and [XCP/CCP Slave plugins](#) that run in [DewesoftX DAQ software](#). Standard Dewesoft CAN (and ethernet) interface hardware can be used.

Dewesoft XCP presentation video

In addition to these XCP Slave and Master plugins, Dewesoft's **SIRIUS XHS** and **IOLITE LX** data acquisition systems can natively serve data via XCP on Ethernet without the need for any additional software. Please watch this short introductory video for more information about Dewesoft XCP Data Acquisition Systems and XCP Data Loggers:

Learn more:

[XCP Slave DewesoftX Manual](#)

[XCP Master DewesoftX Manual](#)

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 the development of the CAN protocol, and for promoting the image of CAN technology.

CANopen provides several additional concepts, including:

Three Basic Communication Models

Master/Slave - where one node is the “master” and all others are slaves.

Client/Server - somewhat similar to master/slaves, except that the nodes are servers of data upon request to a client node.

Producer/Consumer - where certain nodes are configured to produce certain kinds of data automatically, while other nodes are configured to consume it.

Two basic Communication Protocols

SDOs for node configuration

PDOs for sending real-time data

Device Profiles

CiA 401 Input/output modules

CiA 402 motion-control for vendor independence

Object Dictionary

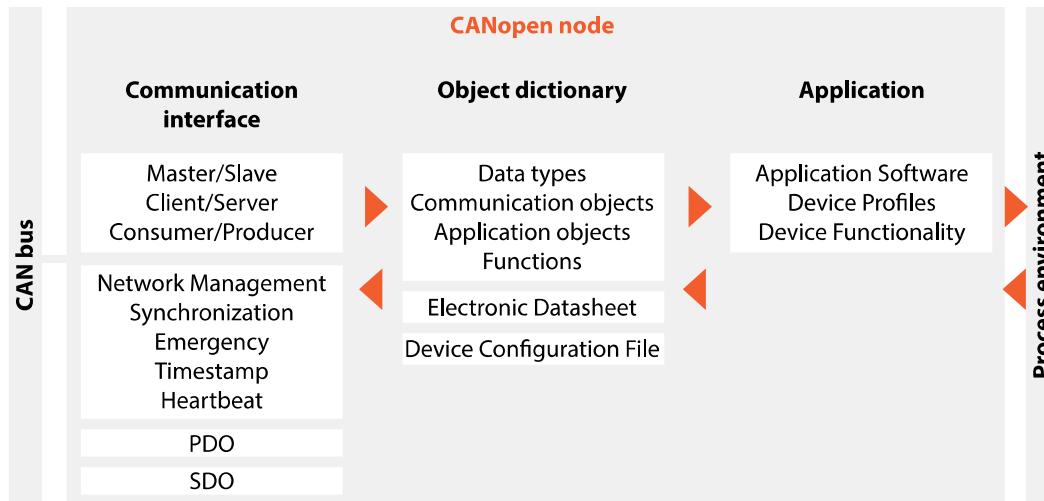
There is an OD (Object Dictionary) for each device on the network. The OD has a standard configuration for the data that defines the configuration of each device on the network.

Device States

A master node is capable of changing or resetting the state of devices on the network.

Electronic Data Sheets (EDS)

The EDS is a standard file format for OD entries - allowing e.g. service tools to update devices



Connections among CANopen concepts and capabilities

Related Communication Buses

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

MOST (Media-Oriented Systems Transport)

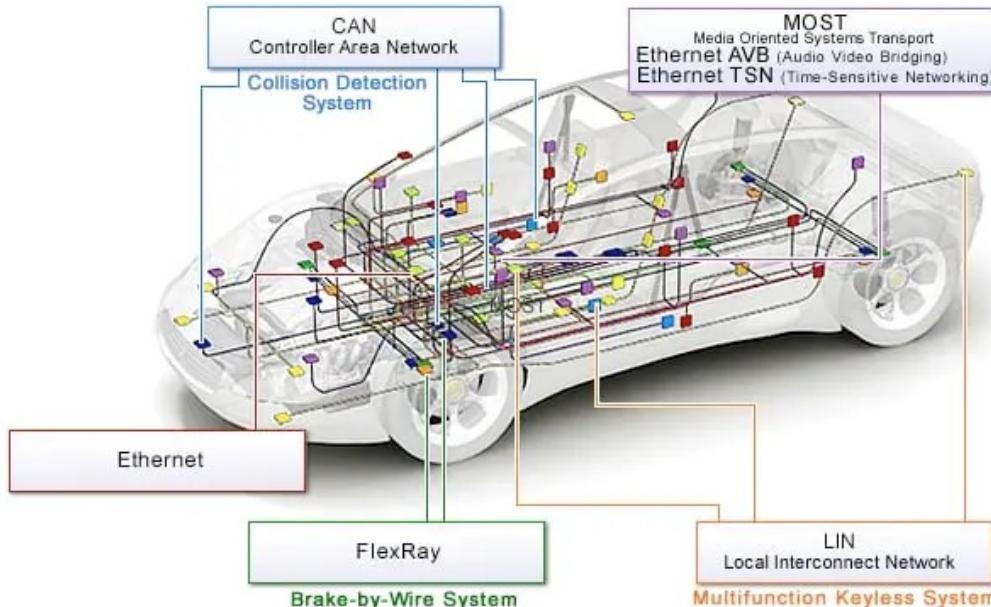
Automotive Ethernet

SENT SAE-J2716

FlexRAY

LIN Bus - Local Interconnect Network

Today's modern vehicles use a combination of multiple data buses. Let's take a look at each one of these and see how they compare to a CAN bus.



Multiple buses used in today's vehicles

Image © 2020 Renesas Electronics Corporation

MOST (Media-Oriented Systems Transport)

Everyone expects their new car to have a better, more capable entertainment system than their previous car. The old-fashioned AM/FM radio that was standard for more than 50 years has been transformed to accept removable media, from the old days of cassette and 8-track tapes to

compact discs (CD) and removable flash media. Today the focus is on streaming content from mobile devices as well as satellite radio (SIRIUS/XM® in North America).



MOST bus - Media-orientated Systems Transport

Image courtesy of Pixabay

Media-Oriented Systems Transport (MOST) is a standard bus used to interconnect vehicle entertainment and information systems developed by a partnership of carmakers called **MOST Cooperation**. It offers data rates of 25, 50, and 150 Mb/s. But it should be noted that these are

aggregate rates that are divided among all of the nodes on the bus.

MOST is used in nearly every car brand around the world. Up to 64 devices can be connected to a **MOST** ring network, which allows devices to be connected or disconnected easily. Other topologies are also possible, including virtual stars. There are various versions of **MOST**, including:

MOST25 offers a 23 MB maximum streaming rate, which is really limited to about 10 kB/s due to protocol overhead and other limitations.

MOST50 doubles the bandwidth of MOST25.

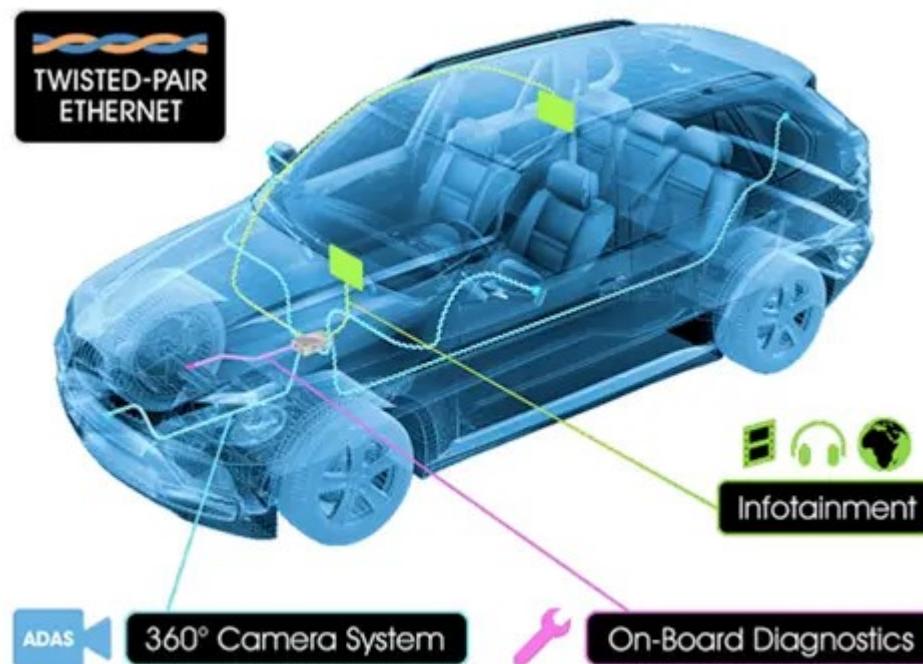
MOST150 triples the bandwidth of MOST50 and adds a physical layer that allows ethernet to be added.

MOST is facing increasing competition from Automotive Ethernet, which is discussed below.

Automotive Ethernet

New technologies such as driver assistance and even self-driving/autonomous vehicle functionalities require higher and higher bandwidth in order to work. This need for speed,

coupled with the low cost of Ethernet hardware, has been a big factor in promoting Automotive Ethernet among carmakers. Other motivations for automotive ethernet include the transfer rates needed for LIDAR and other sensors, raw camera data, GPS data, map data, and higher and higher resolution flatscreen displays.



*Automotive Ethernet
Image © 2017 OPEN Alliance SIG*

But unlike our comfortable home and office environments, a vehicle is subject to a much wider range of temperatures,

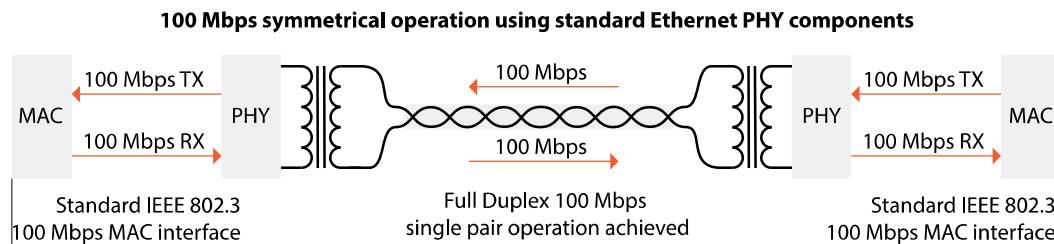
shocks, and continuous vibrations. In addition, there is EMI and RFI that must be blocked so that critical data is not interfered with, especially those related to driver assistance and collision avoidance.

The term “Automotive Ethernet” does not refer to a specific standard per se. It includes any Ethernet-based network used within vehicles. It is also meant to refer to the OPEN Alliance BroadR-Reach standard developed by Broadcom, and to IEEE 802.3bw-2015 aka 100Base-T1.

Despite its obvious advantages of speed and worldwide popularity, until recent years ethernet was only used for diagnostic applications in cars - in other words when the car was under service and not moving. Why? Because of its susceptibility to EMI (electromagnetic interference) and RFI (radio frequency interference), lack of inherent deterministic time synchronization, and susceptibility to connector failure due to the vibration. Standard CAT5 connectors, for example, cannot survive in automobiles under normal use.

However, these issues are being addressed by the IEEE 802.3 and 802.1 working groups. In the meantime, chipmaker Broadcom has developed BroadR-Reach™, which adapts Ethernet technology for automotive use. BroadR-

Reach provides 100 Mb/s speed using unshielded twisted-pair cabling up to 15 meters, or up to 40 meters when a shield is added to the cables.



*BroadR-Reach Automotive Ethernet topology. Broadcom's
PHY chips
simultaneously send and receive data bi-directionally*

BroadR-Reach has been adopted by some carmakers for infotainment systems, driver assistance, on board-diagnostics, and even ADAS applications. It offers 100 Mbps data rates per port, which is much higher than MOST's 150 Mbps aggregate rate, for example.

The BroadR-Reach standard is overseen by the OPEN (One-Pair Ether-Net) Alliance, which advocates for the adoption of Automotive Ethernet.

Ethernet AVB (Audio Video Bridging) is the IEEE standard AVB1.0. It is moving toward acceptance for use with

cameras and multimedia systems. AVB2.0 is aimed at vehicle control applications. AVB is promoted by the AVnu Alliance.

Ethernet TSN (Time-Sensitive Networking) is the IEEE 802.1 standard designed to allow deterministic messaging across standard Ethernet. Not a protocol per se, TSN is a standard implemented at Layer 2 of the Ethernet OSI - i.e., the Data layer (AVB is also a Layer 2 standard).

As an extension of Ethernet AVB described above, TSN focuses on the kind of time synchronization, scheduling, and packet shaping that are necessary for self-driving vehicle applications. Because TSN is all about “time,” Precision Time Protocols (PTP) IEEE 802.1AS and IEEE 802.1ASRev have been selected to provide a shared concept of time across devices.

Learn more:

[Automotive Ethernet OPEN Alliance](#)

[IEEE TSN/AVB Standards](#)

[BroadR-Reach](#)

[Ethernet TSN \(includes AVB\)](#)

According to Gartner, in 2017 there were a total of 19.3 million ethernet ports installed in consumer vehicles. By 2020 this has risen to 122.8 million, a number that is projected to double by 2023.

SENT SAE-J2716

SENT SAE-J2716 (Single Edge Nibble Transmission) was designed to be a low-cost protocol alternative to CAN or LIN. It's a one-way transmission protocol that allows sensors to send their data to ECUs.

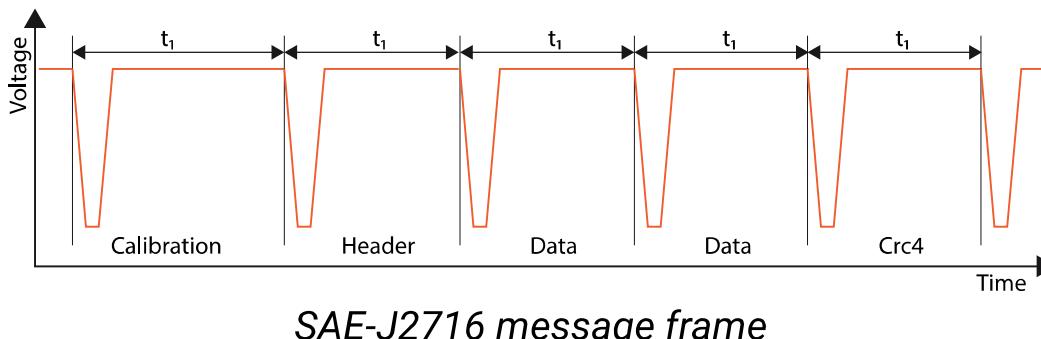
Data is encoded using pulse code modulation (PCM) and transmitted on a single wire. There are three wires in total: signal, ground, and power. Data is encoded in 4-bit "nibbles."

A typical SENT message is 32 bits (8 nibbles), comprised of:

Signal data: 24 bits (6 nibbles)

CRC error detection: 4 bits (1 nibble)

Status info: 4 bits (1 nibble)



It is also possible to configure messages of 20 bits (5 nibbles), where the data is only 12-bits (3 nibbles).

Due to its modulated data design, SENT is ideal for use in electrically noisy environments.

Dewesoft systems support SENT SAE-J2716 data using counter channels to read a SENT signal that is being transmitted by a sensor. Two fast channels and any number of slow channels, which can be detected automatically. Engineers can decode SENT signals from multiple sensors

simultaneously where each sensor is using a different counter, by adding multiple module windows. SENT channels are available as Dewesoft channels.

Learn more:

[SENT channels in DewesoftX software](#)

FlexRAY

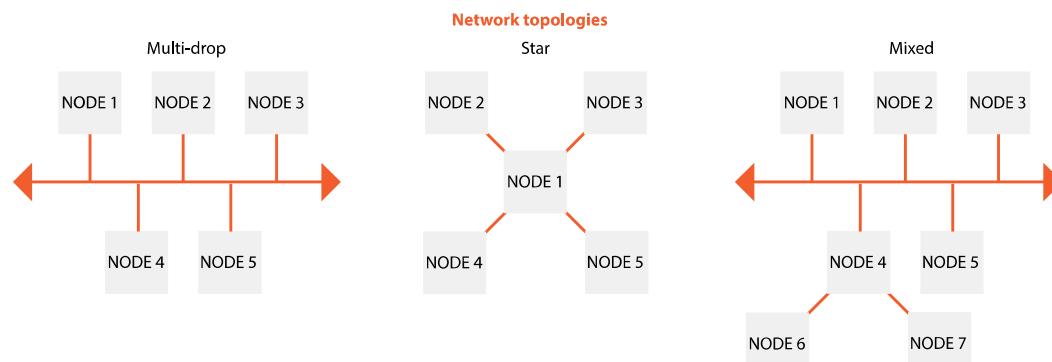
FlexRAY is a protocol used for dynamic automotive applications such as chassis control. It was created in 2005 by the FlexRAY Consortium but has since been standardized under ISO 17458-1 to 17458-5.

FlexRAY transmits data over one or two unshielded, twisted pair cables. It runs at 10 Mbps and supports one or two-wire configurations. Bus, star, and hybrid network topologies are supported, at speeds up to 10 Mbps. Differential signaling keeps noise low without the need for shielded cables, which adds cost and weight.

As with CAN, only one node can write to a FlexRAY bus at

the same time. CAN uses an arbitration bit to determine which data gets priority and is allowed to proceed. FlexRAY, on the other hand, uses a Time Division Multiple Access (TDMA) method where each time-synchronized node must wait for its turn to send a message. This avoids collisions and allows higher overall throughput of data across the bus due to the high overall data rate of the bus.

FlexRAY is often implemented in the classic multi-drop topology shared by LIN and CAN, however, it can also be configured in a star topology. Star topology has the advantage of not allowing a wiring fault to affect more than one node. FlexRAY can also be implemented in a mixed topology, as shown below.



Network topologies: Left: Multi-drop, Center: Star, Right: Mixed

FlexRAY is used most often for high-performance powertrain, safety, and active chassis control applications.

FlexRAY is more expensive than a CAN bus implementation.

However, when dual pairs of parallel data lines are used, this provides redundancy: when a line is damaged, the second line can take over. This is important in mission-critical applications like steering and braking. FlexRAY applications that are not mission-critical typically use a single twisted pair.

Dewesoft systems can easily acquire FlexRAY data using the FIBEX library import option. A software plugin is available to support all **Vector FlexRay** interface cards.

Learn more:

[DewesoftX I/O interfaces, including FlexRAY](#)

LIN Bus - Local Interconnect Network

LIN bus is an inexpensive alternative to the CAN bus. It's

very simple but necessarily limited to one master and 15 slave nodes. LIN is a serial unidirectional messaging system, where the slaves listen for message identifiers addressed to them.

Because of its lower bandwidth and node count limitations, LIN is normally used to control small electric motors and controls. LIN is limited to 19.2 kbps or 20 kbps data rate.



*Adjustable car seat controls in a Mercedes-Benz
Image courtesy of Pixabay*

LIN is a single-wire network defined by ISO 9141. It is used for low-bandwidth applications such as electric windows, lights, door locks, keycard entry systems, electric mirrors, power seats, and similar.

The LIN bus plugin for DewesoftX allows engineers to connect and listen to communication on multiple LIN networks. Using Vector brand LIN BUS hardware, it mimics listen-only slaves that listen to all data transmission on the bus. Decoding can be done in three different forms:

Analog data with extensive scaling options

Discrete data

A mixture of both

The plugin supports importing the configuration from LIN description files (LDF). To read the LIN bus, a **Vector LIN BUS** card is required.

Learn more:

[Dewesoft interfaces, including LIN BUS](#)

DewesoftX manual LIN BUS section

Comparing CAN with Other Vehicle Buses

	LIN	CAN	CAN FD	FlexRay	MOST	Ethernet
Speed	10-20 kbps	1 Mbps	8 Mbps	10 Mbps	150 Mbps (shared)	100 Mbps (per node)
Data size	8 B	8B	64 B	254 B	370 B	1500 B
Cabling	Single wire	UTP*	UTP	UTP	UTP or fiber optic	UT
Topology	Bus	Bus	Bus / passive star	Bus / Star / Mixed	Ring	Star / Tree / Ring
Where Used	Sensors, Actuators (lights, mirrors, etc.)	Backbone, Body, Chassis, Powertrain	Body, Powertrain, Distributed Control, Chassis	High-performance powertrain, Backbone, Drive-by-wire, Chassis	Information & Entertainment Systems	Diagnostics, ECU Programming, Information & Entertainment

	LIN	CAN	CAN FD	FlexRay	MOST	Ethernet
Error Detection	8-bit CRC	15-bit CRC	17 or 21-bit CRC	24-bit CRC	CRC	32-bit CRC
Redundancy	N/A	N/A	N/A	Yes	Yes	N/A
Determinism	N/A	N/A	N/A	Yes	Yes	Not inherent
Cost	\$	\$\$	\$\$\$	\$\$\$	\$\$	\$\$

A high-level comparison of vehicle buses

Notes: * UTP = unshielded twisted pair

As with any networking and interoperable system, automotive bus choice is best driven by the requirements of the application, while keeping an eye on cost and projected industry requirements and trends. Clearly, carmakers don't want to implement old buses in new designs, when there are better buses available at an equivalent or better deployment cost.

Dewesoft CAN Bus DAQ Systems

The CAN bus interfaces provided as standard or optional with Dewesoft systems provide a high level of capability, as well as extensibility to additional protocols.



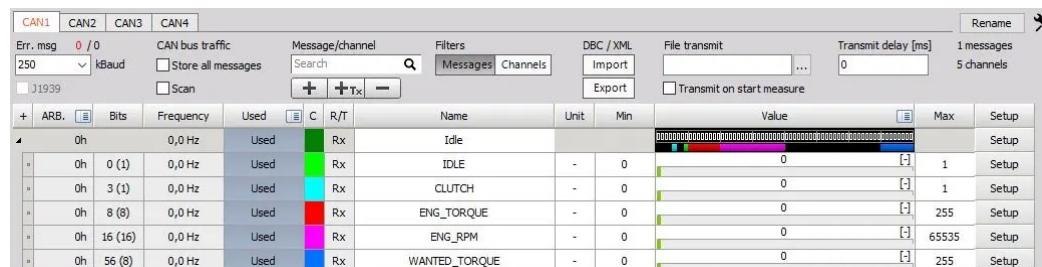
Dewesoft SIRIUS DAQ module recording analog, digital, and CAN bus vehicle data

All **Dewesoft CAN interface** are **galvanically isolated**, protecting the instrument and connected devices from

ground loops and other electrical disturbances. All Dewesoft CAN interface utilize the high-speed **CAN 2.0b standard**. Dewesoft also offers a **CAN FD** device.

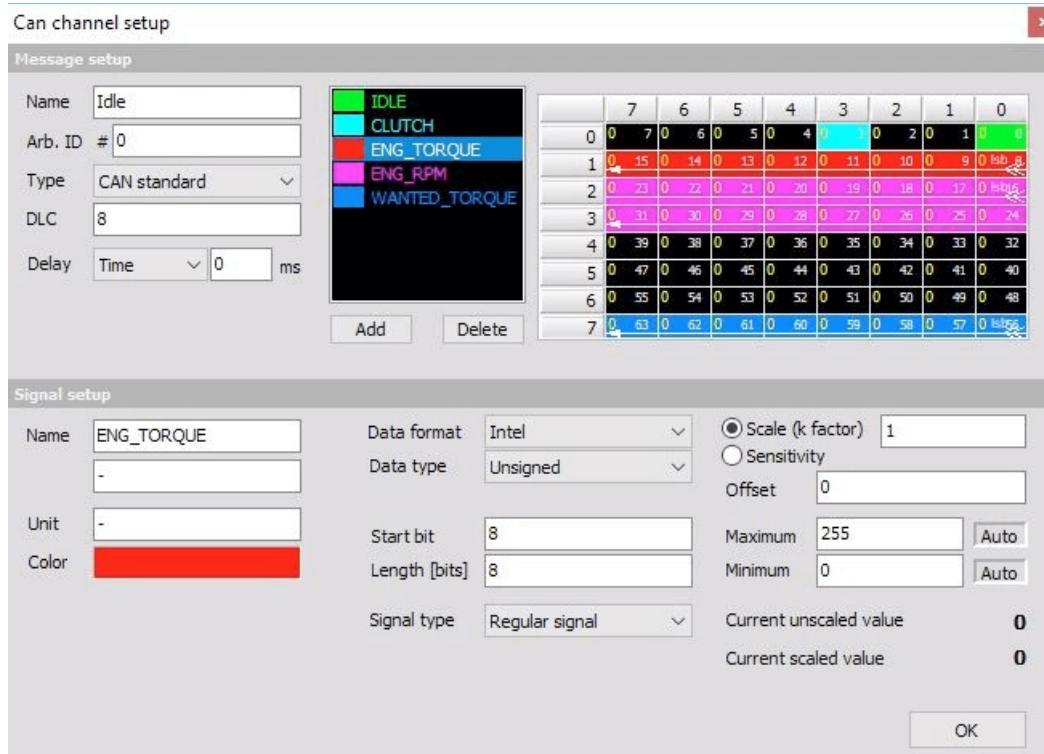
All Dewesoft CAN interfaces can both **read and write** CAN messages, allowing engineers to put messages onto the bus, in order to request data from CAN devices on the network and more.

All Dewesoft CAN interface can be configured in seconds, because the included DewesoftX data acquisition software can **import DBC files**. DBC files are a standard format that allows engineers to parse the data stream into individual channels with names, scaling, proper engineering units, and more.



DewesoftX CAN main setup screen

Clicking the “Setup” button on the far right of any message row opens the CAN channel setup screen shown below:



DewesoftX CAN bus channel setup screen, showing five different channels contained within a single message

DewesoftX makes it extremely easy to configure CAN channels. The software can import and export CAN DBC or XML files. DBC files are common files for CAN message and channel definition. After the import software will automatically set up all available CAN channels and decode CAN messages.

DewesoftX CAN Software Capabilities

- Advanced CAN recording, storing, and analysis
- On-line monitoring and decoding of CAN messages
- Off-line CAN message decoding
- Visual display for displaying CAN data
- Online and offline math analysis of CAN channels
- CAN DBC file import and export
- OBDII on CAN, J1939, and XCP/CCP support
- CAN transmit functionality

Dewesoft CAN Bus Interfaces

Dewesoft was among the first DAQ system makers to fully implement CAN bus interfaces with their analog data acquisition system. Nearly every **Dewesoft DAQ system** has at least one CAN bus interface built-in as standard, and an additional **dedicated CAN interface** can be added internally, externally, or both, while still maintaining perfect synchronization.



Dewesoft multi-channel CAN and CAN FD interfaces

As far back as 2008, the original **DEWE-43** was introduced, featuring two high-speed CAN bus interfaces as a standard feature. A very important thing to know is that the CAN data coming into these ports is fully hardware synchronized with the analog data, as well as with the counter/digital input data. Dewesoft CAN interface is also galvanically isolated, protecting both the instrument and the bus itself from ground loops and other electrical problems.



The DEWE-43A, with 8 multi-function analog inputs, 8 counter/timer/digital inputs, and 2 isolated, high-speed CAN bus interfaces

Less than a year after its introduction, the DEWE-43 was awarded the **NASA TECH BRIEFS Reader's Choice Product of the Year** for its innovations in combining sheer DAQ power and capability into an incredibly small instrument.

Today, Dewesoft offers support for several standard automotive interfaces for analyzing and inspecting vehicle bus data. Data can be captured from all the supported interfaces and synchronized with other sources like analog, video, and others.

Dewesoft DAQ Systems with Built-in CAN Interfaces

As mentioned before, nearly every Dewesoft DAQ system has at least one CAN port built-in as standard, and all systems can be outfitted with CAN if it's not standard, according to this table:

Model	CAN port(s) standard?	Additional CAN ports?
DEWE-43A	Yes, 2 CAN ports standard	Externally*
MINITAURs	Yes, 1 CAN port is standard	Externally*
SIRIUS XHS	Yes, 1 CAN port is standard	Externally*
SIRIUS modular	Yes, 1 CAN port is standard	Externally*
SIRIUS rack series	Yes, 1 CAN port per rack slice	Externally*
SIRIUS mini	No	Externally*
SIRIUS Waterproof	Yes	Externally*
KRYPTON	No. Dedicated KRYPTON CAN modules	Externally*
IOLITE	No	Externally*
IOLITE LX	Yes, 1 CAN port is standard	Externally*
IOLITED	No	Externally*

**Externally means that one or more external and synchronizable CAN interfaces can be added. These include*

the DS-CAN2, SIRIUSim-4X-CAN, SIRIUSf-8x-CAN, and KRYPTONi-1xCAN-FD.

External, Synchronizable CAN Interfaces

In addition to the CAN port(s) built into nearly every Dewesoft DAQ system, separate synchronizable 2, 4, and 8 port CAN interfaces are available. These connect to either a Windows computer running Dewesoft X software or to a Dewesoft DAQ system via USB and a sync cable.



From left to right: DS-CAN2, SIRIUSim-4xCAN, SIRIUSf-8x-CAN

These modules provide 2, 4, and 8 CAN port, respectively

KRYPTONi-1xCAN-FD Module

The latest member of the Dewesoft CAN bus family is the **KRYPTONi-1xCAN-FD**. This is a single port CAN FD device that connects to the DAQ system via EtherCAT®. It supports

high-speed CAN data rates up to 8 Mbps. In addition, CAN FD supports the CAN 2.0 communication protocol as well as special protocols such as J1939.



*The KRYPTONi 1xCAN-FD module is only 62 x 56 x 29 mm
(2.44 x 2.20 x 1.14 in.)*

KRYPTONi-1xCAN-FD has **galvanically isolated** communication lines, and an **isolated sensor supply** of +5 V and +12 V. The power limit of the sensor supply is 1.4 watts.

Because this module is a member of the award-winning **KRYPTON ONE product line**, it can withstand extreme environmental conditions, such as an operating temperature range of -40°C and +85 °C (-40 to +185° F), and sealed against dust and liquids according to IP67.

KRYPTONi-1xCAN-FD is supplied in a standard KRYPTON ONE chassis with a DSUB9 input connector.

More Reading:

[Learn more about Dewesoft CAN and CAN FD interfaces](#)

[Get Dewesoft CAN bus training](#)

Share this article:

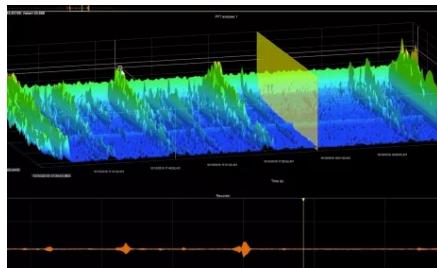
Related Articles



09.03.2022

The Top
150
Electric
Cars and
Other
Electric...

[Read more »](#)



21.02.2022

What is
Signal
Processing?

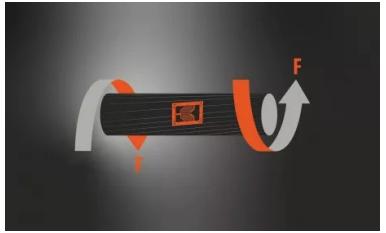
[Read more »](#)



14.02.2022

The Most
Common
Electricity
Source in
Each
Country

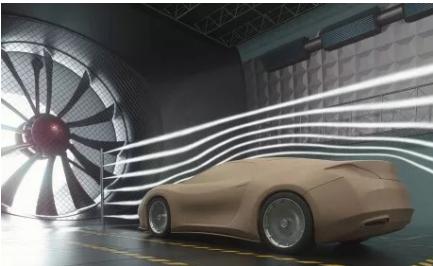
[Read more »](#)



12.02.2022

How To Measure Torque?

[Read more »](#)



19.01.2022

The List of Wind Tunnel Testing Facilities

[Read more »](#)



18.01.2022

Every Satellite Orbiting Earth and Who Owns Them

[Read more »](#)

News

Products	Applications	Support	Company	Newsletter
Overview	Data Recording	Contact us	About us	Signup for the Dewesoft newsletter and be informed of the latest
DAQ systems	Vehicle Testing	Find dealers	Careers	innovations, product updates and
Rugged DAQ systems	NVH Testing	Downloads	Dewesoft Blog	events.
DAQ and control systems	Power & Energy	Forum	Company News	
DAQ software	Vibration Analysis discussions		Events	
Interfaces & sensors	Structural	Software registration	DewesoftX News	
Accessories Services	Dynamics Acoustics Monitoring	Helpdesk RMA service	Product News Warranty	Enter your e-mail SIGN UP
	Aerospace & Defense	Developer portal		Stay in touch

Training	Articles	Case Studies	Application Notes
Training Portal	What Is DAQ?	Biogas Engine	LED Testing
Online PRO Training	Types of DAQ Systems	Testing Monitoring	The Influence of Shielded Cables
Getting Started with Dewesoft	Data Acquisition History	longest bridge Simulation of the	Energy Flow Analysis
How to Measure Signals	How To Choose the Right DAQ?	Heart Rate NASA PCM	Force Measurements on
How to	Dewesoft DAQ	Telemetry	Trampoline

AnalyzeData	Technology	Tractor	Jumpers
How to Develop for Dewesoft Webinars	Explained What Is a Data Logger?	performance tuning	Condition Monitoring of Spaceplane
DewesotX Manual Application Notes Case Studies DAQ Blog	Data Logger vs DAQ System What Is ADC Converter?	Thermal Protection Combustion Analysis	Integrated Power Measurement GNSS on a Mountain Bike
Basic Concepts of Electricity	What Is Signal Conditioning?	Electric	Road Load Data Logger
	What Is Signal Amplifier?	Powertrain	HBV Testing on a Motorbike
	What is a Sensor?	Measurement	Hybrid Boat
	What is TEDS?	Power Analysis	ADAS Testing
	Measure Position	Marine	With Velodyne
	and Displacement	Performance	Lidar
	Testing		Industrial Energy
DAQ Isolation	ROPS Testing	Consumption	
DAQ	Rocket Engine	Vibrations in	
Synchronisation	Test	Buildings	
What is Power	Emergency	Energy	
Analysis?	Ventilator Testing	Consumption of	
What is Modal	Lift Capacity	Electric	
Analysis?	Measurement	Motorcycle	
What Is a	Monitoring of	Bridge Collapses	
Spectrum	Fruit Ripeness	Prevention	
Analyzer?	Scaffolding Test	Wireless Mesh	
What is EtherCAT? Monitoring on the Network			

What Is CAN Bus? NASA's Crawler Innovation in
What Is Condition Aerospace Supervisory
Monitoring? Structural Testing Control
Performing Automated Sound Economizer
Condition Intensity Testing
Monitoring Measurement Automated Motor
Measuring RPM, Vibration Analysis Gear Testing
Angle, and Speed of MRI Scanner Sound Power
List of Automotive Car Seat Crash Level on a
Proving Grounds Test Notebook
Merging DAQ and Wind Turbine Computer
a Real-time Tower Monitoring Electric Vehicle
Control Systems Testing of Real-drive Test
What is ADAS? Automotive Measuring Effects
Types of ADAS Components of Vibration on
Sensors Race Car Exhaust Buildings
How Are ADAS Noise Testing
Systems Tested? Dewesoft DAQ
ADAS Standards Integration
and Safety Vibration on High-
Protocols Voltage Reactor
History of ADCs High-Voltage
FFT Analysis: The Switch Testing
Ultimate Guide ISO 8528
What is Order Certification of
Analysis Diesel Generators
Sound and Noise Industrial Timber

Measurement
with Microphone
Sensors
Strain Gauge
Applications
Satellites Orbiting
Earth
How To Measure
Torque?
The Most
Common
Electricity Source
What is Signal
Processing?
Top 150 Electric
Cars by Range

Packaging
Construction
Noise and
Vibration
Monitoring
Monitoring of the
Railway Viaduct
OMA on a Jetty
Structure
Frequency
Stabilization

Copyright © 2000 - 2022 Dewesoft d.o.o., all rights reserved. [Sitemap](#)

[Privacy Policy](#) [Terms & Conditions](#)
[Legal imprint](#)