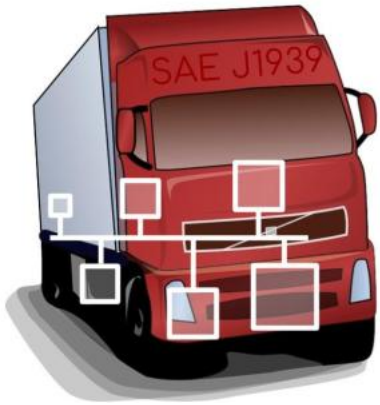
J1939 EXPLAINED - A SIMPLE INTRO (2019)

SAE J1939 is a key protocol in CAN bus data logging, yet it's **difficult to find a really simple intro to J1939**.

But we've found a solution:

We got one of our *non-engineers* to write this intro - and hey, if he gets it, you'll get it!



In this article, we cover:

1. **What is J1939?** Basics & applications
2. How those J1939 **PGNs** and J1939 **SPNs** work
3. ... and **top 6 critical considerations** for logging J1939 data

In short, we'll get you up to speed - and ready to log J1939 data!

Also **check out our 20 min intro video above** with a smooth voice over by Sam F - the #1 YouTube video on the J1939 protocol! (30k+ views)

Alright, let's start with the basics...

1 WHAT IS SAE J1939?

J1939 is a set of standards defining how ECUs communicate, e.g. in heavy-duty vehicles

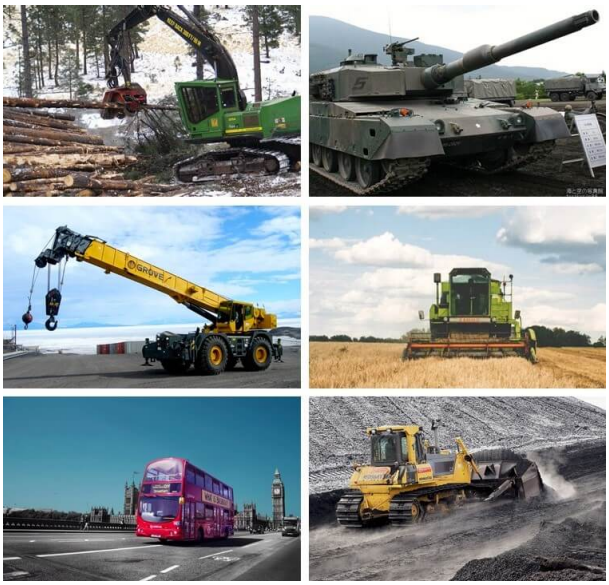
- Applications include trucks, buses, agriculture, maritime, construction and more
- It is based on the high-speed Controller Area Network (CAN) bus, ISO11898
- J1939 is standardized, i.e. ECUs can communicate across manufacturers

However, CAN bus only provides a "tool" for communication (like a telephone) - not the "language" you need to have a conversation.

In most commercial vehicles, this language is the SAE J1939 standard defined by the Society of Automotive Engineers (SAE).

In more technical terms, J1939 provides a higher layer protocol (HLP) using CAN as the "physical layer" basis.

What does that really mean, though?



J1939: ONE LANGUAGE

In simple terms, this means that J1939 offers a **standardized** method for communication across ECUs, or in other words:

J1939 provides ONE language across manufacturers.

In contrast, passenger cars typically rely on manufacturer specific protocols.

Heavy-duty vehicles (e.g. trucks and buses) is one of the most well-known applications. However, several other key industries leverage SAE J1939 today either directly or via derived standards (e.g. ISO 11783, MilCAN, NMEA 2000, FMS):

- Foresting machinery (e.g. delimbers, forwarders, skidders, ...)
- Mining vehicles (e.g. bulldozers, draglines, excavators, ...)
- Military vehicles (e.g. tanks, transport vehicles, ...)
- Agriculture (e.g. tractors, harvesters, ...)
- Construction (e.g. mobile hydraulics, cranes, ...)
- Fire & Rescue (e.g. ambulances, fire trucks, ...)
- Other (e.g. ships, pumping, e-buses, power generation, ...)

From a data logging perspective, SAE J1939 provides an overlay to CAN including a set of **standardized messages** and **conversion rules** that apply across a wide array of vehicles within the above areas.

For this reason, a good understanding of the J1939 protocol is core in e.g. building fleet management systems.

J1939 HISTORY & OUTLOOK

Where did J1939 come from?

- The first docs were released in 1994 (J1939-11, J1939-21, J1939-31)
- In 2000, the initial top level document was published
- Today, the SAE J1939 standard has replaced former standards SAE J1708 and J1587

... and where is it going?

Today, we see massive growth in IoT (Internet of Things) and 'connected mobility' will be a huge market. This will be enabled via scalable fleet solutions using affordable WiFi data loggers, but the heart of such applications will remain the SAE J1939 protocol.

In other words, **the J1939 standard will only grow in importance going forward:**

"**The market for in-vehicle connectivity** - the hardware and services bringing all kinds of new functionality to drivers and fleet owners - is expected to reach **€120 billion by 2020.**"

- Boston Consulting Group, Connected Vehicles and the Road to Revenue

---

KEY CHARACTERISTICS OF J1939

To dig a bit deeper, let's look at some key characteristics of SAE J1939:

- **Speed:** The speed is typically 250 kbit/s, though recently with support for 500 kbit/s
- **Extended:** J1939 uses CAN 2.0B, i.e. an extended 29 bit identifier

use

- **Special Values:** A data byte of 0xFF (255) reflects N/A data, while 0xFE (254) reflects an error
- **Multibyte:** Multibyte variables are sent least significant byte first (Intel byte order)
- **Multi-Packet:** J1939 supports PGNs with up to 1785 bytes using a transport protocol

We will go more in-depth on some of the above characteristics in the sections below.

For a bit of technical details on the 'higher layer protocol' aspect of J1939 and the link to CAN bus, click below:

J1939 - A Higher Layer Protocol +

2 J1939 PGNS & SPNS - HOW TO INTERPRET J1939 MESSAGES

If you're reading this article, your end goal is likely to **analyse decoded SAE J1939 data in human-readable form**.

To do so, you need to interpret the SAE J1939 message format, which requires an understanding of **PGNs** and **SPNs**.

In short: Each J1939 message is identified via a PGN and contains 8 data bytes, split into parameters, SPNs.

Below we go into detail on these concepts.

PARAMETER GROUP NUMBER (PGN)

Let's start with the SAE J1939 identifier, the **PGN**.

First of all: **What is a J1939 PGN?**

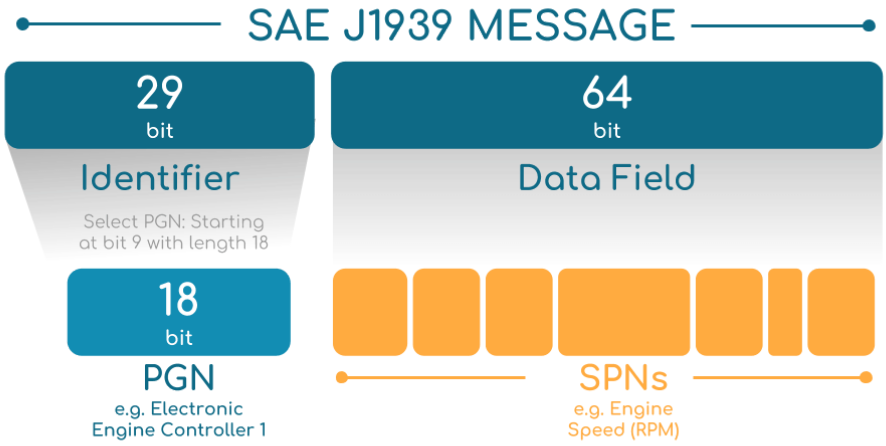
A PGN is a unique ID for looking up the function of a J1939 message and the associated data parameters (i.e. the SPNs)

You can look this up e.g. in the J1939-71 document, which lists PGNs and SPNs, as well as how to interpret and convert the data.

For an illustration of what the various PGN names may look like, see our J1939 PGN list.

A word of caution:

You cannot match PGNs vs the full 29 bit CAN identifier. Instead, you need to separate out the 18 bit PGN as below.



Let's look at a **PGN example:**

PGN61444 - Electronic Engine Controller 1 - EEC1

Transmission Repetition Rate: Engine speed dependent

Let's say we logged a J1939 message with ID **0x0CF00401**.

Here, the **PGN starts at bit 9, with length 18** (indexed from 1).

Bit Start/Byte	Length	SPN ID	SPN Description
1.1	4 bits	899	Engine Torque Mode
2	1 byte	512	Driver's Demand Engine - % Torque
3	1 byte	513	Actual Engine - Percent Torque
4-5	2 bytes	190	Engine Speed
6	1 byte	1483	SA of Controlling Device for Engine Control
7.1	4 bits	1675	Engine Starter Mode
8	1 byte	2432	Engine Demand - Percent Torque

Further, the document will have details on the PGN including priority, transmission rate and a list of the associated SPNs - cf. the example below.

For this PGN, there are seven SPNs (e.g. Engine Speed, RPM), each of which can be looked up in the J1939-71 documentation for further details.

ARE J1939 MESSAGES REALLY THAT SIMPLE?

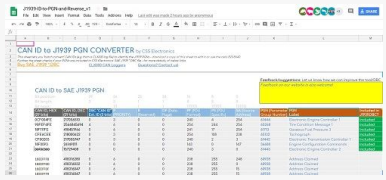
Not completely: The above is a bit simplified as the J1939 29-bit identifier can be broken further down.

Specifically, the ID comprises the **Priority** (3 bits), **PGN** (18 bits) and **Source Address** (8 bits).

Further, the PGN can be broken into four parts: Reserved (1 bit), Data Page (1 bit), PDU Format (8 bits) and PDU Specific (8 bits).

For more on this, we host a popular online converter that lets you convert CAN IDs to PGNs - and see the formulas behind the PGN sub component calculations. This is quite educational and also useful if you're interested in J1939 DBC files.

Check it out!



SUSPECT PARAMETER NUMBER (SPN)

You may be wondering: "Are the PGNs the data parameters I'm looking for?"

Not exactly. That's where SPNs come in!

But what is a J1939 SPN?

The SPNs of a J1939 message reflect data parameters - such as speed and RPM

Let's assume we've identified a PGN (e.g. 61444) based on a raw 29 bit message ID (e.g. 0x0CF00401).

For a given entry of this message ID, we also logged 8 bytes of raw data - now, **how do we interpret and convert this?**

0CF00401 FF FF FF 68 13 FF FF FF

Here we need to look at the SPN, which reflects the ID of a specific parameter contained within the data bytes of a given PGN.

For example consider SPN 190, Engine Speed, mentioned in the previous example (cf. below).

SPN190 - Engine Speed

Actual engine speed which is calculated over a minimum crankshaft angle of 720 degrees divided by the number of cylinders

Data Length	2 bytes
Resolution	0.125 rpm/bit, 0 offset
Data Range	0 to 8,031.875 rpm
Type	Measured
SPN	190
PGN	61444

For simplicity, let's assume we're only interested in converting and analysing this particular parameter.

In that case we see from the PGN info that relevant data is in byte 4 and 5, i.e. 0x68 and 0x13. Taking the decimal form of 0x1368 (Intel byte order), we get 4968 decimal. To arrive at the RPM, we conduct a scaling of this value using the offset 0 and the scale 0.125 RPM/bit.

The result is 621 RPM.

Note how some data bytes in the above are FF or 255 decimal, i.e. not available.

While the PGN may support SPNs in this range, this specific application does not support these parameters.

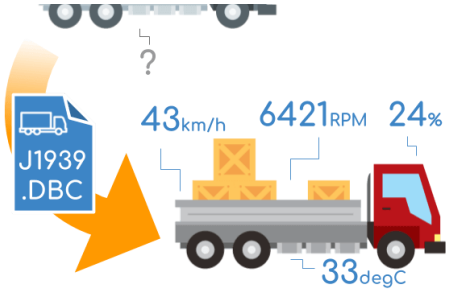
USING A DBC TO DECODE J1939 DATA

convert logged or streamed J1939 data.

In a DBC context, PGNs are often called "Messages" and SPNs are called "Signals". For more on this, check out our DBC conversion article which uses SAE J1939 as a case example.

Further, if you lack an up-to-date J1939 DBC file, you can get a low cost up-to-date version below:

LEARN MORE



ADVANCED: J1939 REQUEST & MULTI-PACKET MESSAGES

The SAE J1939 protocol supports a number of more advanced operations. These include requests, multi-packet messages, multiplexing and more. For a light intro to these concepts, click to expand the below:

J1939 Request Messages

PGN59904 - J1939 Request Message

Transmission Repetition Rate: User specified
Data Length: 3 bytes
Data Page: 0
PDU Format: 234
PDU Specific: Destination address
Default Priority: 6
Parameter Group Number: 59904 (0x00EA00)

Bit Start/Byte	Description
1-3	PGN of message to be requested

Most J1939 messages are broadcast to the CAN bus, but some need to be requested (e.g. some J1939 diagnostic trouble codes). This is achieved using the 'request message' (PGN 59904), which is the only J1939 message with only 3 bytes of data. It has priority 6, a variable transmit rate and can either be sent as a global or specific address request. The data bytes 1-3 should contain the requested PGN (Intel byte order). Examples of requested J1939 messages include the diagnostic messages (DM). As for OBD2, you can use our Transmit feature to set up SAE J1939 request messages.

J1939 Multi-Packet Messages

The PGN & SPN examples are based on a J1939 message of 8 data bytes, which is the size of most messages.

However, **two other possible sizes exist**: The **3 bytes request message** above - and **variable size** messages.

The latter allows communication of data packets beyond the usual 8 bytes limit of the CAN bus format.

These are referred to as **J1939 multi-frame or multi-packet messages**. The J1939 protocol specifies how to **deconstruct**, **transfer** and **reassemble** the packets - a process referred to as the Transport Protocol (cf. J1939-21). Two types exist:

1. The **Connection Mode** (intended for a specific device)
2. The **BAM** (Broadcast Announce Message) which is intended for the entire network.

In simple terms, the **BAM** works by the transmitting ECU sending an initial BAM packet to set up the transfer.

The BAM specifies the multi-packet PGN identifier as well as the number of data bytes and packets to be sent. It is then followed by up to 255 packets of data. Each of the 255 packets use the first data byte to specify the sequence number (1 up to 255), followed by 7 bytes of data. The max number of bytes per multi-packet message is therefore 7 bytes x 255 = 1785 bytes. The final packet will contain at least one byte of data, followed by unused bytes set to FF. In the BAM type scenario, the time between messages is 50-200 ms. Finally, a conversion software can reassemble the multiple entries of 7 data bytes into a single string and handle it according to the multi-packet PGN and SPN specifications.

3 PRACTICAL TIPS FOR J1939 DATA LOGGING

Ready to log data from your truck, bus or excavator?

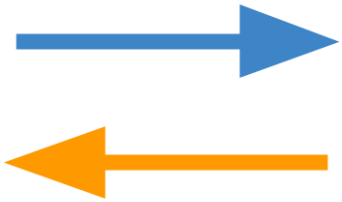
To round things off, this section provides **6 critical considerations** when choosing a J1939 data logger solution.

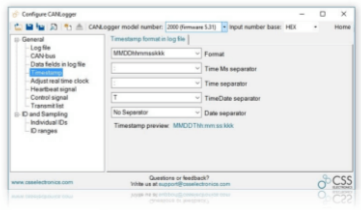
Are you managing a heavy duty vehicle fleet? Then being able to log your J1939 data effectively - and at low cost - is vital

1 **LOGGER VS INTERFACE:** Standalone J1939 data loggers with SD cards are ideal for logging data from e.g. vehicle fleets over weeks or months. A WiFi J1939 logger also enables telematics use cases (see tip 3#). In contrast, a J1939 USB-PC interface requires a PC to stream data from the CAN bus in real-time. This is e.g. useful for diagnostic purposes - or analysing physical events. The CLX000 enables both modes of operation.

2 **ADAPTER VS INDUCTION:** To connect the CAN analyzer to a J1939 asset (e.g. a truck) you can typically use the 9-pin J1939 connector. For the CLX000, we offer a DB9-J1939 adapter which fits the 9-pin deutsch connector found in many heavy duty vehicles. Alternatively, you may prefer to connect the the CLX000 directly to the CAN bus via e.g. a CANCrocodile. This method uses induction to record data silently without cutting any CAN wiring.

3 **WIFI & CELLULAR:** For vehicle fleet management & telematics, WiFi or cellular is vital remotely transfering the data. The CL3000 is a low cost option and allows transfer via WiFi hotspots - incl. cellular. If you e.g. need data from a truck on-the-road, you can connect the CL3000 to a 3G/4G USB WiFi hotspot. However, in cases where data only needs to be retrieved rarely, a lower cost solution can work (e.g. the CL2000).





4 **J1939 DBC SOFTWARE:** When logging or streaming J1939 data, software for post processing is key. In particular, the software should support DBC-based data conversion to allow quick access to human-readable data. Our free CANvas software suports DBC conversion of e.g. J1939 data, while our free Wireshark plugin allows DBC conversion of live-streamed data. Further, we offer a low cost J1939 DBC file with 1,000+ PGNs & 5,000+ SPNs.

5 **REQUEST PGNS:** Some J1939 PGNs are only available on-request, meaning that you need to "poll" the CAN bus to log these. The CLX000 is able to transmit up to 20 custom CAN messages, which can be used to send periodical PGN requests. Note that this is not possible in "silent mode" (i.e. it is not possible if the logger is connected via e.g. a CANCrocodile).

6 **ADVANCED CONFIGS:** To optimize your J1939 data logging, a number of advanced configurations can be helpful. In particular, the CLX000's PGN filters and sampling rate options help optimize the amount of data logged, which is key for e.g. managing cellular bandwidth usage. Other useful options include silent mode and cyclical logging, with the latter enabling the logger to always prioritize the latest data (useful in e.g. blackbox logging).

We hope this helps you get ready to log your J1939 data! If you need sparring just contact us - we'll get back in <24 hours!

WHERE CAN I LEARN MORE?

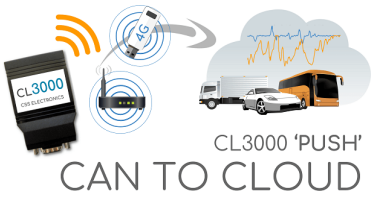
Check out our GUIDES section for more intros, guides & use cases - incl. our J1939 telematics intro below.

Want to log data from your truck, harvester or other J1939 application?
Then learn more about our CLX000 CAN logger below!

GET YOUR CLX000 | J1939 TELEMATICS | CONTACT US

RECOMMENDED FOR YOU

J1939 LOGGER - SIMPLE TELEMATICS



AUTO-PUSH J1939 DATA TO CLOUD

DBC CAN BUS CONVERSION IN WIRESHARK: J1939 EXAMPLE

CONTACT

CSS Electronics | VAT: DK 36711949
Soeren Frichs Vej 38K (Office 35), 8230
Aabyhoej, Denmark
contact[AT]csselectronics.com
+45 91252563

JOIN OUR NEWSLETTER

SIGN UP