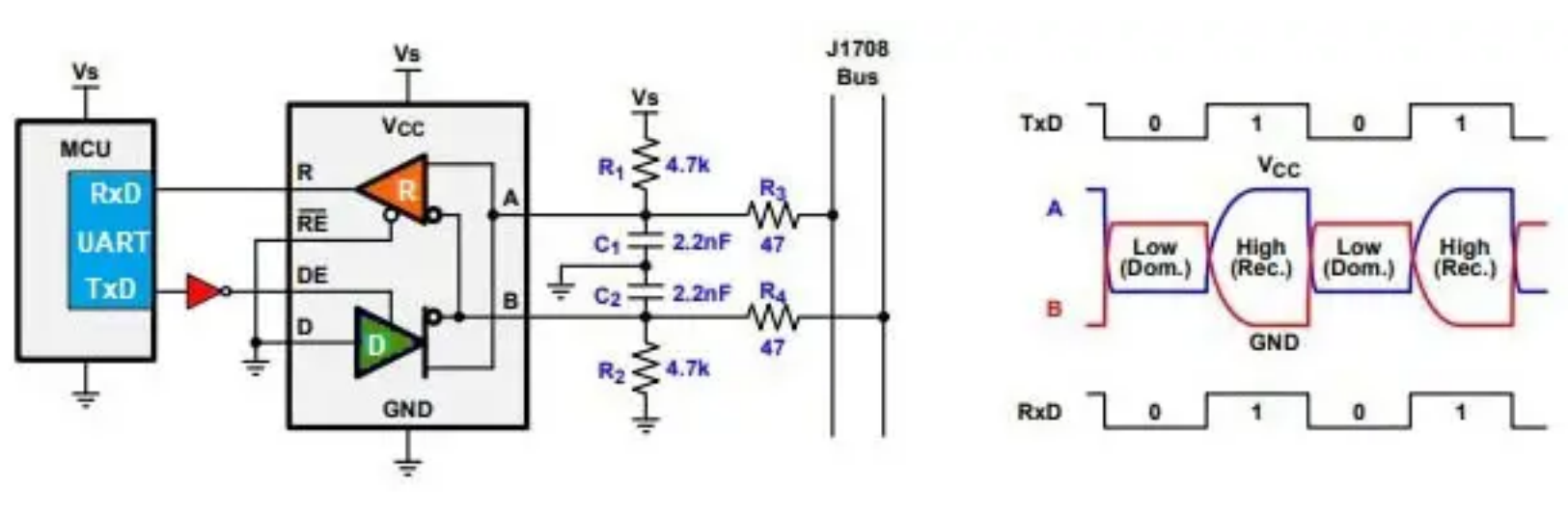


SAE J1708 Protocol

The SAE J1708 Protocol is an Enhanced serial type synchronous or asynchronous protocol. This standard developed by using the Enhanced Universal Synchronous Asynchronous Receiver Transmitter. Mostly it is called EUSART in short. In this article, we will discuss this protocol, including basic to advance with frame format. So that you can learn in a smart way to learn, work, and enjoy your work environment without requesting your neighbor teammate.

Introduction To EUSART Protocol

The J1708 serial communications link specification was issued by SAE in 1986. It was the first network bus that was accepted in the real industry-wide standard for heavy vehicles or commercial trucks. Later it is being integrated with J1587 so it is named with **SAE J1708** protocol. Basically, the J1708 is used for the physical layer and the Common higher-layer protocols that operate on top of J1708 are SAE J1587 and SAE J1922. It is mostly equivalent to the general protocol. Due to its long-distance with high-speed data rate and networking capability, it is named as Enhanced Universal Synchronous and Asynchronous Receiver Transmitter Protocol. If you know the CAN protocol that is having only asynchronous type serial protocol. But in the case of EUSART is having both the Synchronous and Asynchronous, but in the vehicle, we are using the asynchronous type.

SAE J1708 Protocol Definition:

The SAE J1708 protocol bus is a very simple, robust, low-speed, multi-master serial type data bus Protocol used in heavy trucking vehicles. If the MID value is more than 127, then it will come under the J1587 standard. It will have one extra byte called PID. How the J1939 protocol is working on CAN for the heavy vehicles, the SAE J1587 is working with J1708 in EUSART protocol.

SAE J1708 Protocol Bit Time

The bit time is the time required to send 1 bit of data. For the **SAE J1708** protocol specification, the 1-bit time is 104.16 ± 0.5% (±500 ns) microseconds. This is nearly equivalent to a baud rate of 9600 bits per second. So this time corresponds to the 9600 baud rate range in the serial standard. All other times specified by the J1708 specification are multiples of bit times.

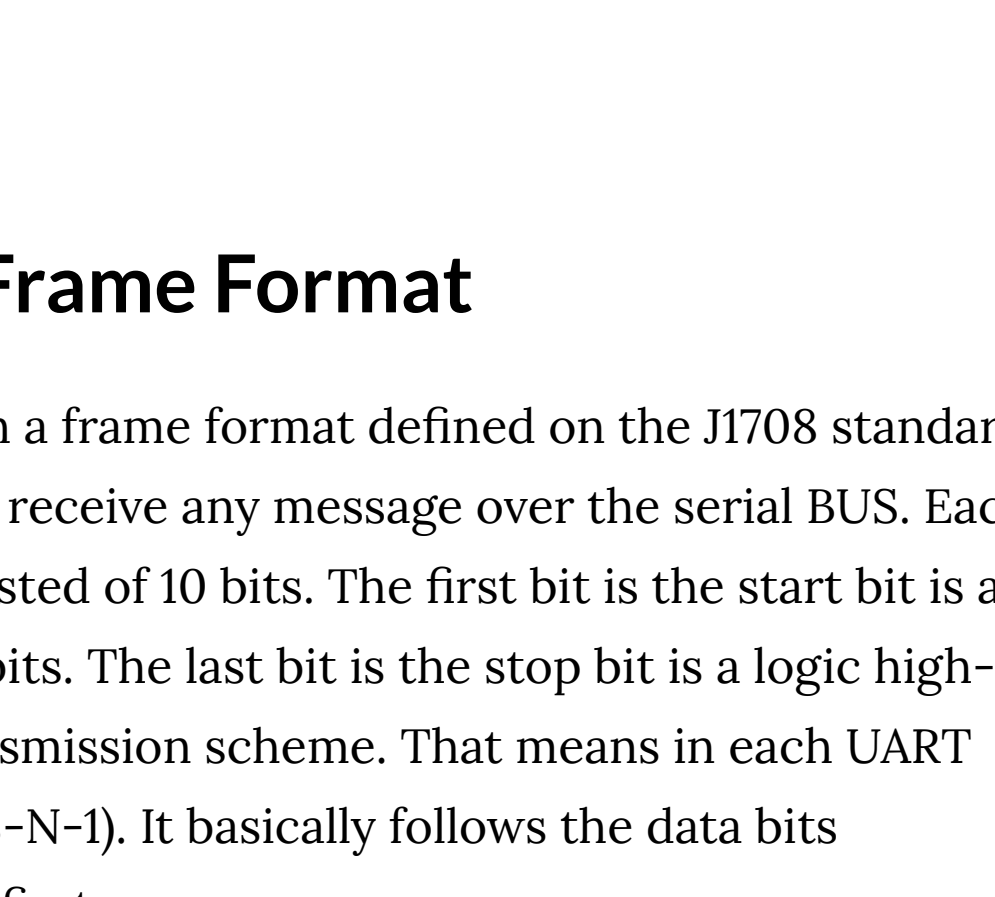
SAE J1708 Protocol Features

- It Minimizes the hardware cost.
- Offers flexibility and the possibility of further expansion of a network bus.
- It is mostly used in conjunction with the application layer protocol SAE J1587 for enhanced networking features.
- The basic RS-485 bus transceiver for its physical layer having a low cost.
- It can have a maximum network length of up to 40 meters.
- J1708 network is based on a bus topology.
- It is a Serial byte-oriented communication protocol with the least significant byte (LSB) first.
- J1708 works on serial standard 9600 baud rate.
- It has a simple packet with MID, Data bytes, and Checksum.
- A single J1708 data frame can have up to 21 bytes long.
- It has also Error detection and handling is handled at the time of message transmission.

J1708 Protocol Physical Layer

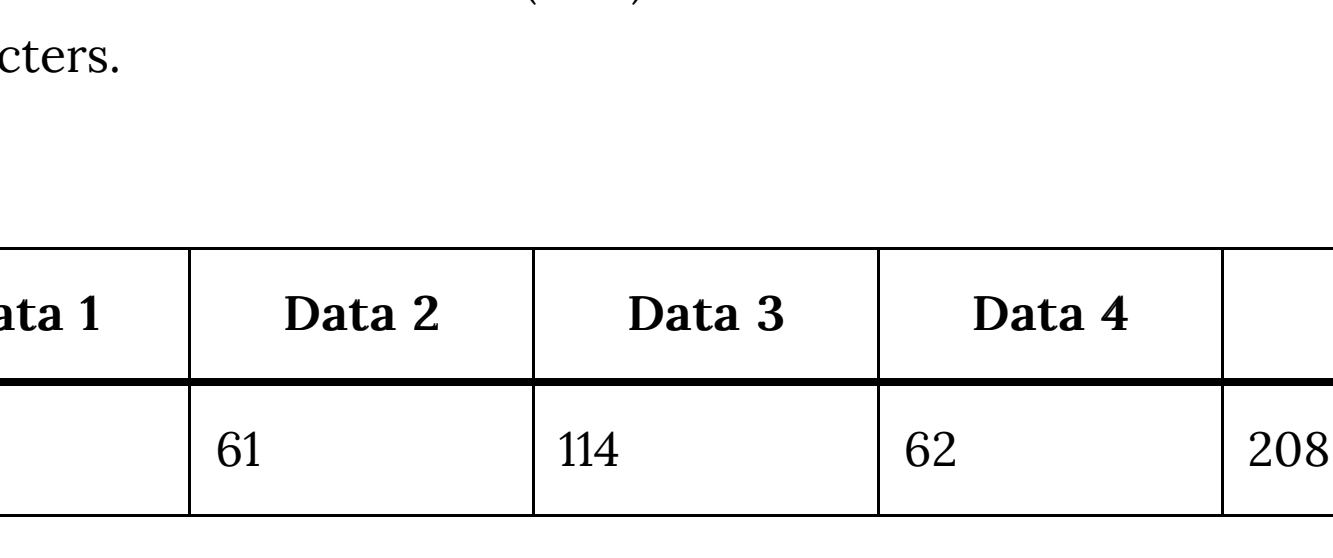
The J1708 protocol uses the RS-485 chip as its transceiver. The **SAE J1708 Protocol** bus network supports at least 20 nodes with RS-485 transceivers. The difference is that the J1708 does not use the bus termination resistors used by RS-485. The j1708 data link layer is used for data communication on the network with the help of the RS-485 serial bus.

The SAE J1708 protocol bus consists of two wires with at least one twist (360°) per inch (2.54 cm) and a total length of up to 40 meters. This bus works the same as the CAN bus. The data bits are high or low can be determined by the difference voltage between the two wires. if the difference voltage between the two-wire is +200mV, then it is called logic High (1). if the difference voltage between the two-wire is -200mV, then it is called logic low (0). The transceiver should be connected with DC power with +6V to -6V in relation to common ground. **[This Image is designed as per the J1708 standard]**



SAE J1708 Protocol Message Frame Format

The J1708 protocol communication happens with a frame format defined on the J1708 standard. Basically, it uses the character format to send or receive any message over the serial BUS. Each character in the SAE J1708 protocol having consisted of 10 bits. The first bit is the start bit is a logic low-level. The next eight bits are the data bits. The last bit is the stop bit is a logic high-level. This mechanism is the standard UART transmission scheme. That means in each UART data character, 8 data bits, no parity, 1 stop bit (8-N-1). It basically follows the data bits transmission with the Least Significant Bit (LSB) first.



SAE J1708 Standard Frame Format

J1708 is having 3 fields, such as:

- Message Identification Characters (MID).
- Data Characters.
- Checksum.

MID	Data 1	Data 2	Data 3	Data 4	Checksum
44	23	61	114	62	208

Basic J1708 Standard Message Format

J1708 Message Identification (MID) Character

The first character of every SAE J1708 protocol message is MID. It is always a byte character. The MID is used to identify who is sending the message and the message priority and collision detection. The main purpose of this byte is to give a unique identifier value to each message transmitted on the BUS. This MID is ranged from 0 – 255. The MIDs are categorized into different modules for uninterrupted serial communication purposes. existing systems, or systems that may presently be under development, and to avoid conflicts, which otherwise might arise if indiscriminate use of MIDs were permitted.

1. **MIDs from 0 – 68:** Transmitter Message Identifier:
2. **MID 69 – 86:** Set aside for J1922 standard uses.
3. **MIDs from 87 – 110:** Reserved for ELECTRONIC INTERFACE SUBCOMMITTEE (They can assign it as per requirement)
4. **MID 111:** Reserved for FACTORY ELECTRONIC MODULE TESTER (FEMT). It is used for Diagnostic Test Tool interface with the Off-Vehicle for diagnostic purposes.
5. **MIDs from 112–127:** Generic OEM (Unassigned, it can be used by OEM).
6. **MIDs from 128–255:** Reserved by Data Format Sub-Committee. It is further used by the SAE J1587 standard.

The transmitter message Identifier is again divided into multiple sub-field categorized. So that you can identify the particular message is from which module.

J1708 MID Range	Transmitter Message Category
00-07	Engine related MIDs
08-09	Tractor Brakes
10-11	Trailer Brakes
12-13	Tractor Tires
14-15	Trailer Tires
16-17	Tractor Suspension
18-19	Trailer Suspension
20-27	Transmission
28-29	Electrical Charging System
30-32	Electrical
33-35	Cargo Refrigeration or Heating
36-40	Instrument Cluster
41-45	Driver Information Center
46-47	CAB Climate Control
48-55	Diagnostic Systems
56-61	Trip Recorder
62-63	Turbocharger
64-68	Off-Board Diagnostics (OBD-II)

SAE J1708 MIDs

J1708 Data Character Format

The **SAE J1708** protocol Data characters are the real data packets sent by the transmitter ECUs over the EUSART BUS. It consists of 0-19 data characters. As you already know each character is having 10 bits. The first bit (bit-0)of this byte is called the start bit. It is always a low-logic level for start bit identification. The last bit (bit-9) is the stop bit. The bit-1 to bit-8 is the data byte ranging from 0 – 255 value. The OEM shall have a proper document that will define the parameters, parameter order, scaling, and error detection or correction coding if it is required by them.

J1708 Protocol Checksum Format

The last character of each **SAE J1708** protocol message is a Checksum byte. It always getting added in the last byte of each message by the J1708 physical layer. You no need to do it in your application layer. It is calculated by the addition of each byte. Basically, it is an addition of MID, PID, and data with the 2's complement of the SUM. If the 8-bit checksum is 0 by neglecting the carry bit, then you can assure that the data byte received is correct.

J1708 Checksum calculation

- Add all the bytes including MID and PID from **SAE J1708** data frame.
- Invert the sum.
- Add 1 with the Sum

SAE J1708 Protocol Message Length Format

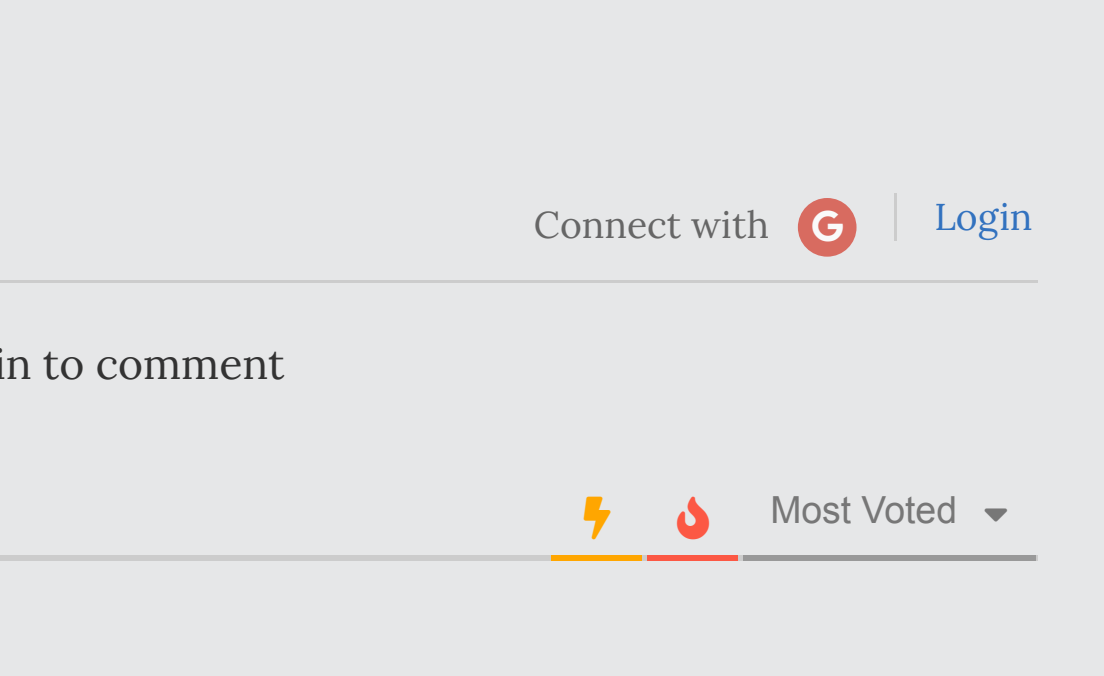
The **SAE J1708** message lengths can be maximum up to 21 bytes including the Message Identifier (MID) and a Checksum byte. If you have some data that is more than that then you can use the multi-frame format. That means you can be broken into multiple consecutive 21-bytes or lesser packets when the engine is not running or the vehicle is not moving. So that practically you are following the J1708 standard by following the 21-bytes format and by implementing the Enhanced Universal Synchronous Asynchronous Receiver Transmitter Transport Protocol (EUSART-TP) on the application you can send more data bytes.

SAE J1708 Message Priority

```
SAE J1708 message priority formula
Pd= Tb*2*P
Pd = Priority Delay in µS
Tb = Bit time or 104.16 µS
P = Message Priority
```

J1708 Connector Diagram

This j1708 connector diagram is used for Off-Board Diagnostic. If there is any fault or any software update is required, then in the service center the service or diagnostic engineer will use this female port for connecting his external computer system for diagnostic purposes. This figure shows the j1708 pinout for understanding how to connect the OBDII port with the vehicle network.



Subscribe
Connect with G | Login
Please login to comment
2 COMMENTS
Most Voted

swarup kumar nath

very nice explanation of J1708 Protocol.

0

Saswat Parida

very nice explanation about the J1708 Protocol.
Really thank you so much for this knowledge and for making it open source.

0

Search Your Idea Here...
Piest Forum - Android App Link
Gift A Cup Of Coffee To PiEmbSystech
JOIN TELEGRAM FOR TECH. DISCUSSION

Recent Posts
Understanding Inheritance in CPP Programming Language
Objects of a Class in CPP Language
Classes in CPP Programming Language
How Crystal Oscillators Work in Microcontrollers: Ensuring Precise Timing and Stable Performance
Download and Install Turbo C++ Compiler
Archives
Select Month
Embedded Research Forum

Table Of Contents

Automotive Electronics
Computer Science
Electronics Technology
Linux System
Programming Language
C
C++
Python
Robotics Technology
VLSI