Communication Protocol CAN

* Introduction:

Before the year of 1985, the wiring of the automotive system was very hard, complicated, and expensive due to the high number of ECUs in the vehicle. So, the need for a system more simply was necessary, so Bosch originally developed CAN at that time, a high-integrity serial bus system for networking intelligent devices, which replaced automotive point-to-point wiring systems. Diagram

Description automatically generated

*Figure 1. CAN networks significantly reduce wiring.*

In 1993, the CAN emerged as the standard in-vehicle network due to its energetic features for safety and it became the international standard known as ISO-11898.

In 1996 the On-Board Diagnostics OBD-II standard which incorporates CAN becomes mandatory for all cars and light trucks sold in the United States.

* **What is CAN?**

CAN is short for ‘controller area network’. Controller area network is an electronic communication bus defined by the ISO 11898 standards. Those standards define how communication happens, how the wiring is configured, and how messages are constructed, among other things. Collectively, this system is referred to as a CAN bus.

* **Features of CAN protocol**

1. **Low cost:** Since a CAN serial bus uses two wires (with high-volume and low-cost production), it offers a good price-to-performance ratio.
2. **Bit-rate:** the bit rate is uniform and fixed for all the nodes and The speed of the CAN may be different for different networks available in a system.
3. **System flexibility:** it is easy for engineers to integrate new electronic devices into the CAN bus network without significant programming overhead and supports a modular system that is easily modified to suit your specs or requirements.
4. **Message routing:** every message is identified by a special unique identifier that does not indicate the destination of the message but only describes the meaning of the data available in the message. So that all the nodes are connected in the network can decide by message filtering technology using this message ID either to receive the data or not.
5. **Multi-master communication:** Any node can access the bus.
6. **Multicast:** more than one node/ECU in the network able to receive the same transmitted message.
7. **Arbitration:** If two or more nodes start transmitting messages at the same time, the bus access conflict is resolved by the bit-wise arbitration using the Identifier. The mechanism of the arbitration guarantees that neither information nor time is lost. If a data frame and a Remote frame with the same Identifier are initiated at the same time, the Data frame prevails over to the Remote frame.
8. **Priorities:** every message has a priority, so if two nodes try to send messages simultaneously, the one with the higher priority gets transmitted and the one with the lower priority gets postponed. This arbitration is non-destructive and results in non-interrupted transmission of the highest priority message.
9. **Error Capabilities:** the CAN specification includes a Cyclic Redundancy Code (CRC) to perform error checking on each frame's contents.  Frames with errors are disregarded by all nodes, and an error frame can be transmitted to signal the error to the network.  Global and local errors are differentiated by the controller, and if too many errors are detected, individual nodes can stop transmitting errors or disconnect themselves from the network completely.

* **Can Network Message Format:**

CAN devices send data across the CAN network in packets called frames. These frames can be differentiated based on identifier fields. A CAN frame with 11-bit identifier fields is called **Standard CAN** and with 29-bit identifier field called the extended frame.

* + 1. **Standard frame:**

**Graphical user interface, application

Description automatically generated with medium confidence**

Figure 2. The standard CAN frame format.

* **SOF (start-of-frame) bit** – indicates the beginning of a message with a dominant (logic 0) bit.
* **Arbitration ID**– identifies the message and indicates the message's priority. Frames come in two formats -- standard, which uses an 11-bit arbitration ID, and extended, which uses a 29-bit arbitration ID.
* **RTR (remote transmission request) bit** – serves to differentiate a remote frame from a data frame. A dominant (logic 0) RTR bit indicates a data frame. A recessive (logic 1) RTR bit indicates a remote frame.
* **IDE** **(identifier extension) bit** – allows differentiation between standard and extended frames.
* **R0** – Reversed bit.  Not used currently and kept for future use.
* **DLC** **(data length code)** – indicates the number of bytes the data field contains.
* **Data Field** – contains 0 to 8 bytes of data.
* **CRC (cyclic redundancy check)** – contains 15-bit cyclic redundancy check code and a recessive delimiter bit. The CRC field is used for error detection.
* **ACK** **(Acknowledgement) slot** – It compromises of the ACK slot and the ACK delimiter. When the data is received correctly the recessive bit in the ACK slot is overwritten as a dominant bit by the receiver.
* **EOF (End of Frame) –** the 7-bit field marks the end of a CAN frame (message) and disables.

1. **Extended frame:**

**Table

Description automatically generated with low confidence**

Figure 3. The extended CAN frame format

It is the same as 11-bit identifier with some added bits.

**SRR (Substitute Reverse Request) –** The SRR bit is always transmitted as a recessive bit to ensure that, in the case of arbitration between a Standard Data Frame and an Extended Data Frame, the Standard Data Frame will always have priority if both messages have the same base (11 bit) identifier.

**R1–** It is another bit not used currently and kept for future use.

**Message frame:**

There are four different frames that can be used on the bus.

* **Data frames:** These are most commonly used frames and used when a node transmits information to any or all other nodes in the system.  Data Frames consist of fields that provide additional information about the message as defined by the CAN specification. Embedded in the Data Frames are Arbitration Fields, Control Fields, Data Fields, CRC Fields, a 2-bit Acknowledge Field, and an End of Frame.

Diagram

Description automatically generated with low confidence

*Figure 4. CAN Data frames.*

* **Extended frame:** The purpose of the remote frame is to seek permission for the transmission of data from another node. This is similar to the data frame without the data field and the RTR bit is recessive.
* **Error frames:** If the transmitting or receiving node detects an error, it will immediately abort transmission and send an error frame consisting of an error flag made up of six dominant bits and an error flag delimiter made up of eight recessive bits.  The CAN controller ensures that a node cannot tie up a bus by repeatedly transmitting error frame.

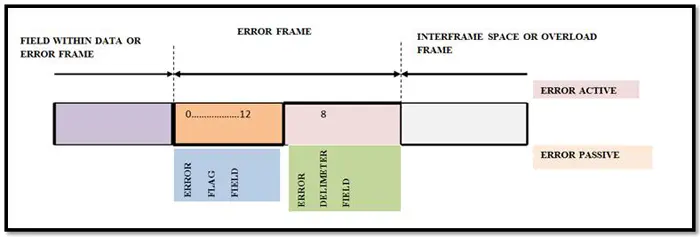
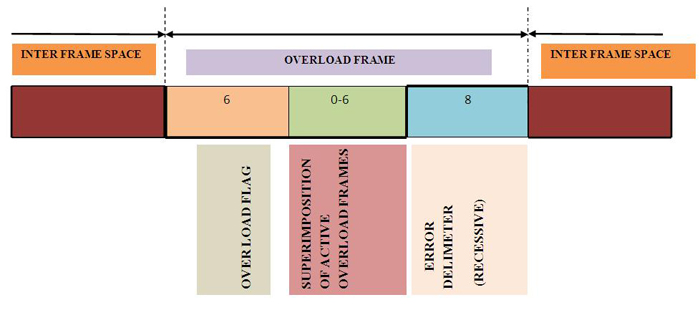


Figure 4. CAN Error Frames.

* **Overload frame:** It is similar to the error frame but used for providing an extra delay between the messages.  An Overload frame is generated by a node when it becomes too busy and is not ready to receive.

****

*Figure 5. CAN Overload frame.*

**CAN Protocol Working Principle**

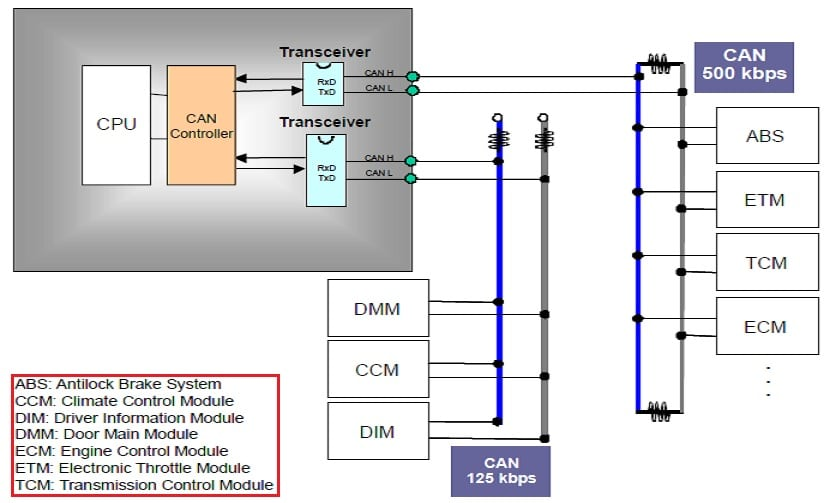
****

Figure 6. CAN Error Frames.

Each node in the CAN bus requires the below modules to work together in a CAN network.

1. **Microcontroller (Host):** It decides what the received messages mean and what messages it wants to transmit.
2. **CAN controller:** They are often an integral part of the microcontroller that handles framing, CRC, etc. like the Data Frame, Remote Frame, Error Frame, Overload Frame, and Inter-Frame Space generation.
3. **CAN Transceiver**: It converts the data from the CAN controller to the CAN bus levels and also converts the data from CAN bus levels to a suitable level that the CAN controller uses.

The transceiver drives or detects the dominant and recessive bits by the voltage difference between the CAN\_H and CAN\_L lines. The nominal dominant differential voltage is between 1.5V to 3V and the recessive differential voltage is always 0V.

The CAN transceiver actively drives to the logical 0 (dominant bits) voltage level and the logical 1 (recessive bits) are passively returned to 0V by the termination resistor. The idle state will always be in the recessive level (logical 1).

Individually, CAN\_H will always be driven towards supply voltage (VCC) and the CAN\_L towards 0V when transmitting to the dominant (0). But in a practical case, supply voltage (VCC) or 0V cannot be reached due to the transceiver’s internal diode drop. CAN H/L will not be driven when transmitting a recessive (1) where the voltage will be maintained at VCC/2.

**Operation Of The CAN Network​**

Each node is then assigned a unique identification number called the Physical Address of the ECU.

All the nodes are interesting to transmit compete for the channel by transmitting a binary signal based on their identification value.

A node will drop out of the competition if it detects a dominant state while transmitting a passive state.

Thus, the node which is having the lowest identification value will win the bus network and start the transmission of the message.

If any error detects either the transmitter or receiver, it stops the sending of the Data Frame and starts sending the Error Frame. The error is having an error state to handle the error in CAN Protocol called Error Handling in CAN Protocol. There are 5 types of error in CAN protocol that can occur.