

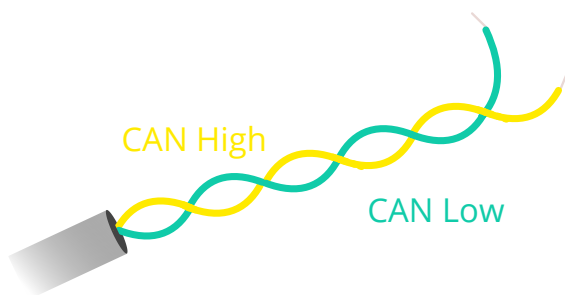
Controller Area Network

Controller Area Network (CAN) is a communication protocol commonly used in automotive and industrial automation industries. It is known for its reliability, high data transmission rate, and robustness in noisy environments.

Physical Implementation

The CAN bus consists of two wires, commonly colored yellow and green, which are intertwined.

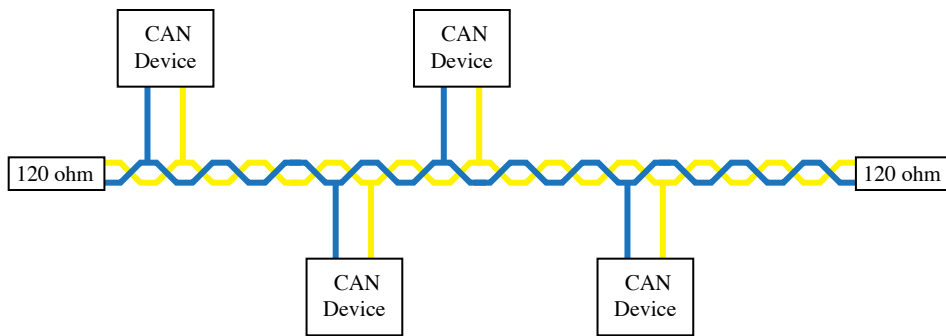
- One wire represents CAN High (CAN_H) and is typically colored yellow.
- The other wire represents CAN Low (CAN_L) and is typically colored green.



At the ends of the two wires, they are connected via termination resistors (usually 120 ohms).

The devices on the CAN bus are then connected to the bus at any point.

- Note that the distance from the CAN device (node), which is referred to as stub length, to the CAN bus should be under 30 cm.



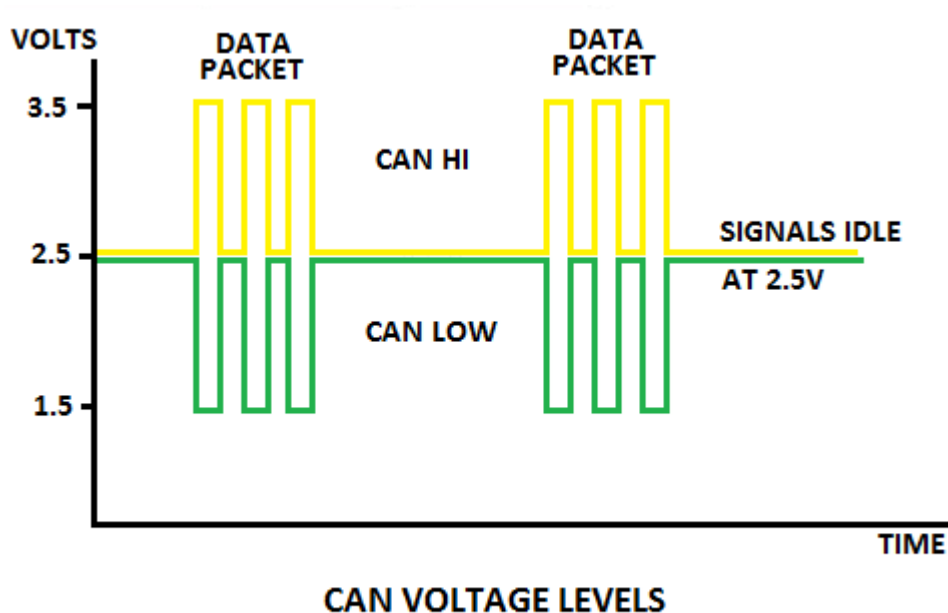
Why use two wires compared to one?

- **Resistance to External Noise:** Two wires help to reduce electrical interference and noise.
- **Reduction of Noise Emitted by the Bus**

For example, a voltage spike on a single wire could cause the receiver to read incorrect high bits. On a two-wire system, the high and low wires will go opposite of each other, with the difference between CAN_H and CAN_L indicating the bits. This is called differential signaling.

- Because communication is based on the difference between the two voltages, a voltage spike occurring on the CAN bus will not affect the difference between the two voltages.
- Note that both wires at idle voltage represent a 1, and the wires at different voltages indicate a 0.
- A digital 1 is considered to be the recessive state, and a 0 is considered the dominant state. This means that the bus will default to 1 when idle, and 0 is used to transmit data.

Another reason for using two intertwined cables is to reduce the electrical noise generated by the CAN bus. Since the two wires transmit opposite signals, the electromagnetic waves they emit cancel each other out. To an external observer, it appears as if the CAN bus generates no electrical noise.



Termination Resistors

When the transceiver is not transmitting on the bus, we want the bus to return to the recessive state. This is one reason why termination resistors are placed at either end of the bus.

Another reason is to prevent bus reflections. When a signal travels down the CAN bus, it meets the resistor, which prevents the signal from being reflected back onto the bus.

- This is why the stub length should be kept under 30 cm—because the nodes do not have termination resistors, signals are subject to reflections, which can return to the main bus.

120-ohm resistors are commonly used because:

- They match the characteristic impedance of the CAN bus wires, preventing bus reflections.
- They help control the slew rate of the signal, creating sharp rising and falling edges.

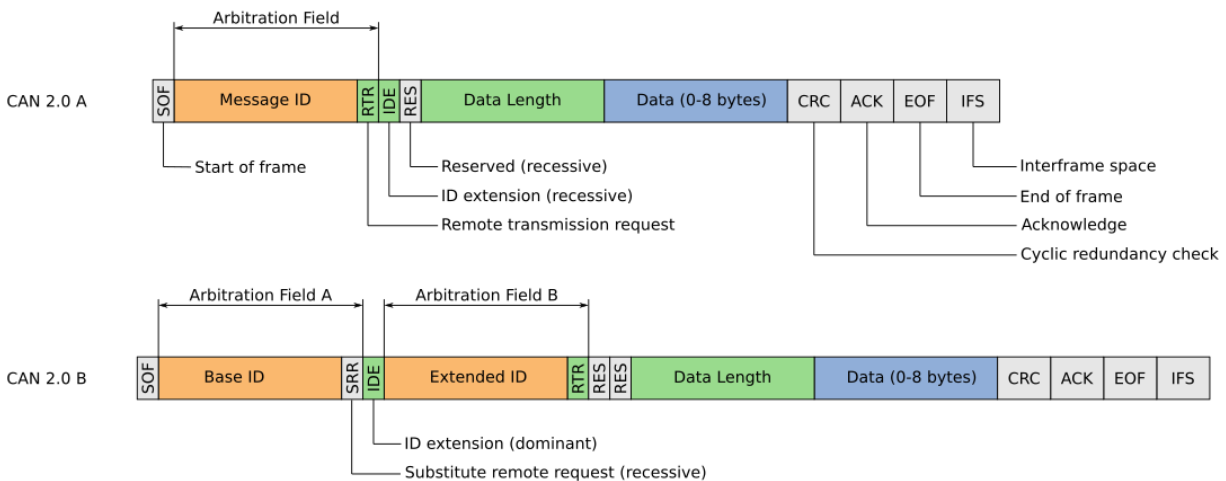
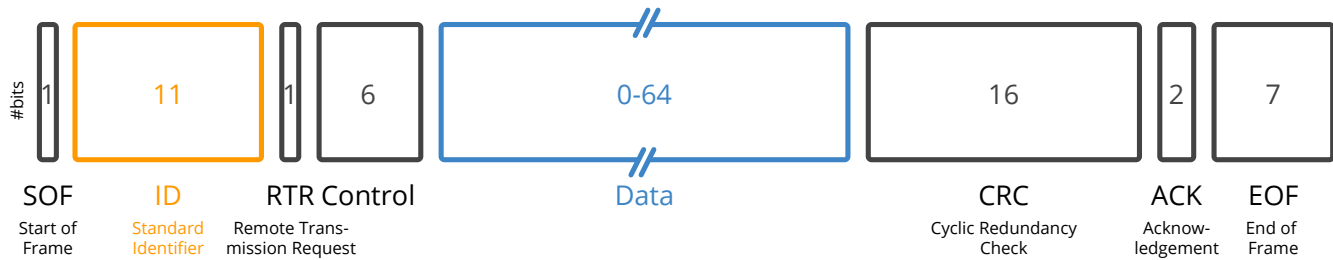
CAN Packet Structure

Every CAN message has the following structure:

- **SOF (Start of Frame)**: 1 bit (0) which indicates the start of the CAN message.
- **ID (Identifier)**: 11 bits that are used to identify the information contained in the CAN message. Sometimes, a 29-bit identifier is used for buses requiring more identification.
- **RTR (Remote Transmission Request)**: 1 bit that is used to identify whether it is a data frame or a remote frame. Remote frames are used to request the transmission of a data frame with the same identifier.
- **IDE (Identifier Extension)**: 1 bit that is used to indicate the use of either the 11-bit or 29-bit identifier.
- **R0 (Reserved Bit)**: An unused bit that should always be set to 0.
- **DLC (Data Length Code)**: A 4-bit field that specifies the number of bytes (0-8) carried by the CAN message.
- **Data Bytes**: 0-8 bytes containing the data to be sent in the message.
- **CRC (Cyclic Redundancy Check)**: A 15-bit checksum that is calculated based on the content of the message. This is used to guard against any bit flips by verifying the contents of the message.
- **ACK (Acknowledgment)**: 2 bits that are used to indicate if the message was successfully received by at least one node on the bus. The receiving nodes pull the bus to a dominant state to acknowledge the successful reception of the message.
- **EOF (End of Frame)**: 7 consecutive recessive (1) bits that are used to indicate the end of the CAN message.
- **IFS (Inter-frame Space)**: 3 recessive bits used to add a small delay between messages, ensuring that all nodes have time to process

the previous message.

Standard CAN frame



Bus Arbitration

Bus arbitration is a system that automatically decides which node has the right to transmit on the CAN bus.

It works by comparing the IDs of the messages being transmitted. The CAN protocol uses a priority system where a lower ID has a higher priority because a 0 bit (dominant) overrides a 1 bit (recessive).

When a node detects that its ID has been overwritten by a more dominant 0 bit, it stops transmitting its message and goes into receive mode, allowing the node with the higher priority message to continue transmitting.

This effectively creates a priority system among the nodes, where the node with the lowest ID number has priority over other nodes on the

CAN bus.

Note that this system works because the CAN node is always listening to the bus, even when transmitting, allowing it to stop its own transmission if a higher priority message is detected. This also means that there must be at least two participating nodes on the bus for data to be transmitted.