# Understanding Controlled Area Network (CAN)

CAN stands for controller area network. They are designed specially to meet the Automobile Industry needs. Before CAN was introduced, each electronic device is connected to other devices using many wires to enable communication. But when the functions in the automobile system increased, it was difficult to maintain because of the tedious wiring system. With the help of the CAN bus system, which allows ECUs to communicate with each other without much complexity by just connecting each ECU to the common serial bus. Hence when compared with the other protocols used in automotive systems i.e., CAN vs LIN, CAN is robust due to less complexity.

CAN Protocol can be defined as a set of rules for transmitting and receiving messages in a network of electronic devices connected through a serial bus. Each electronic device in a CAN network is called a node. Each node must have hardware and software embedded in them for data exchange. Every node of a CAN bus system has a host microcontroller unit, CAN controller and, CAN transceiver in it. CAN controller is a chip that can be embedded inside the host controller or added separately, which is needed to manage the data and sends data via transceiver over the serial bus and vice versa. CAN Transceiver chip is used to adapt signals to CAN bus levels.

CAN is a message-based protocol where every message is identified by a predefined unique ID. The transmitted data packet is received by all nodes in a CAN bus network, but depending on the ID, CAN node decides whether to accept it or not. CAN bus follows the arbitration process when multiple nodes try to send data at the same time.

## CAN Bus Electrical Specification:

CAN signals processed by CAN transceiver are single-ended signals and differential signals (CANH and CANL). CAN High and CAN Low lines are at 2.5v in ideal condition. CAN defines logic “zero” as the dominant bit and logic “one” as the recessive bit. When the dominant bit is transmitted, CAN High goes to 3.5v, and CAN low goes to 1.5v i.e., the differential voltage of the dominant bit is 2v. When the recessive bit is transmitted CAN High and CAN Low lines are driven to 2.5v, indicating the differential voltage of the recessive bit is 0v.CAN bus terminal resistor of 120 ohms should be added at the physical end of CANH and CANL lines to avoid any signal reflections.

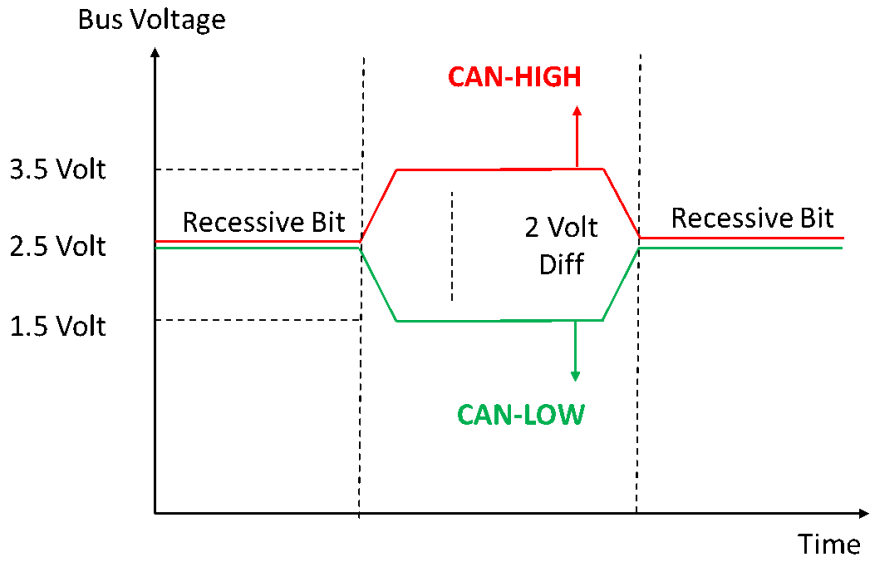


*Figure 1. CAN bus differential signals*

# Frame types of CAN:

Frame is a defined structure or format that carries meaningful data(bytes) within the network. CAN has four frame types:

- Data Frame
- Remote Frame
- Error Frame
- Overload frame

**Data Frame**

Data Frame contains the actual data for transmission. Data Frames consist of fields that provide additional information about the message i.e., Arbitration Field, Control Field, Data Field, CRC Field, a 2-bit Acknowledge Field, and an End of Frame.

There are two types of Data frames

1. Standard Frame or Base frame format
2. Extended Frame format

The only difference between the two formats is standard frame supports an 11-bit identifier, and the extended frame supports a 29-bit identifier made up of an 11-bit identifier and extended 18-bit identifier. IDE bit is dominant in a standard frame and recessive in an extended frame.

**Base Frame Format:**

Standard CAN protocol is also known as the Base frame format. The standard frame is mainly used to send data.

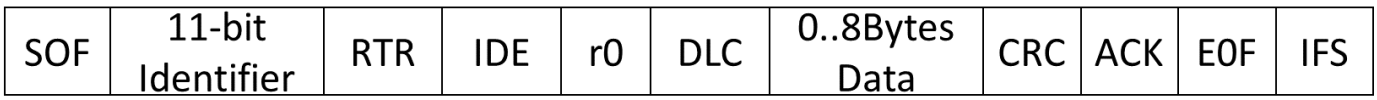


*Figure 2. Standard Frame*

***Terminologies***

- SOF - Start of Frame. Denotes the start of frame transmission.
- Identifier - 11-bit unique id and also represents message priority Lower the value, higher is the priority.
- RTR - Remote Transmission Request. It is dominant for data frames and recessive for remote frames.
- IDE - Single Identification Extension. It is dominant for standard frames and recessive for extended frames.
- R0 - reserved bit.
- DLC - Data Length Code. Defines the length of the data being sent. It is of 4-bit size.
- Data - Data to be transmitted and length is decided by DLC.
- CRC– Cyclic Redundancy Check. It contains the checksum of the preceding application data for error detection.
- ACK– Acknowledge. It is 2 bits in length. It is dominant if an accurate message is received.
- EOF– end of the frame and must be recessive.
- IFS– Inter Frame Space. It contains the time required by the controller to move a correctly received frame to its proper position.

**Extended FRAME:**

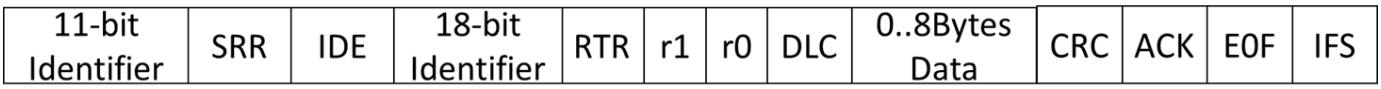


*Figure 3. Extended Frame*

It is the same as the Standard Frame with some additional fields.

SRR- Substitute Reverse Request. The SRR bit is always transmitted as a recessive bit to ensure that the standard Data frame has high priority when compared to the extended Data frame if both messages have the same 11-bit identifier. It also contains an 18-bit identifier other than an 11-bit identifier.
r1- Reserved bit.

**Remote Frame**

The remote frame is similar to a data frame with two differences. The remote frame is sent by the receiver to request data from the transmitter. The differences between the remote frame and data frame are remote frame does not contain any data field in it since it is not used for the data transfer. The second difference is RTR bit in the arbitration field is recessive for the remote frame. The data frame wins the arbitration if both are ready to be transmitted at the same time because of the dominant RTR bit in the data frame.

**Error frames**

If transmitting or receiving nodes detect an error, it will immediately stop transmission and send an error frame consisting of an error flag made up of six dominant bits and error flag delimiter made up of eight recessive bits.

Error flags:

1.Active Error flag

2.Passive Error flag

Active Error Flag: Transmitted by the node when an error is detected on a CAN network

Passive error flag: Transmitted by the node when an active error is detected on a CAN network.

Error Counters: If an error is detected on a bus, then TEC or REC count increases.

1.Transmit Error Counter (TEC)

2.Receive Error Counter (REC)

- When TEC and REC is lesser than 128, an active error frame is transmitted
- When TEC or REC is greater than 127 and less than 255, a passive frame is transmitted
- When TEC is greater than 255, the node enters into bus off state, then no frames can be transmitted

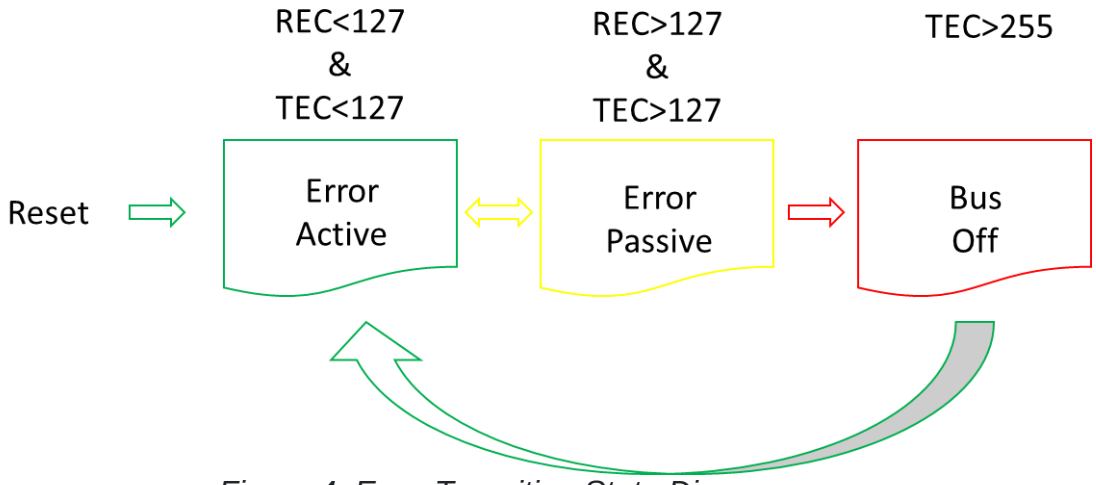


*Figure 4. Error Transition State Diagram*

**Overload Frame**

The overload frame contains two fields Overload flag and Overload Delimiter. Overload flag consists of six dominant bits followed by overload flags generated by other nodes. Overload delimiter consists of eight recessive bits. Overload conditions that lead to the transmission of overload frame are:

1.when the receiver needs a delay of the next frame.

2.when a dominant bit is detected during intermission.

## Arbitration

Arbitration is a mechanism to resolve the conflict when more than one node is ready to transmit the message at the same time. Whenever bus is free, any node can transmit the data. If multiple nodes are ready to transmit the data at the same time access to the bus is conflicted and can be resolved by arbitration using an identifier. During arbitration process, every transmitter compares the transmitted bit value with bit value on the bus. If the bit value is same, the node continues to transmit the bits. If the transmitted bit value is not the same as bus value then, the dominant bit overwrites the recessive bits.

The Arbitration field of the CAN message consists of either 1-bit or 29-bit identifiers and RTR bit. The identifier with the lowest value has the highest priority. If both data frame and the remote frame have the same identifier and ready to send data at the same time data frame has high priority since remote transmission bit (RTR) is dominant in the data frame and recessive in the remote frame.

## Message level CAN bus Error Control Mechanisms

When compared to other protocols i.e. CAN vs LIN, CAN vs MOD bus, CAN protocol is robust Error checking makes CAN protocol robust. Through these mechanisms if an error is detected, the node transmits an error frame and destroys the transmitted frame.

**CRC check**

The cyclic redundancy check value is calculated by transmitting nodes and is transmitted via the CRC field and this value is received by all nodes. Then all the received nodes calculate CRC value and match the value with the transmitted value. If values are different then an Error Frame is generated.

### ACK slots

When a transmitting node sends a message, a recessive bit is sent in the acknowledgement slot. If a message is received, then the acknowledge slot is replaced by the dominant bit which would acknowledge that at least one node correctly received the message. If this bit is recessive, then no node has received the message properly.

### Form Error

End of frame, Inter-frame space, Acknowledge Delimiter are fields that are always recessive, if any node detects a dominant bit in any one of these fields then, a Form Frame is generated and the original message is present after certain period.

## Layered architecture of CAN

It consists of three layers i.e. Application layer, Data link layer, and Physical layer.

- **Application Layer**: This layer interacts with the operating system or application of a CAN device.
- **Data Link Layer:** It connects actual data to the protocol in terms of sending, receiving, and validating data.
- **Physical Layer:** It represents the actual Hardware i.e. CAN Controller and Transceiver.

### CAN physical layer Characteristics

The CAN physical layer is divided into three parts: Physical coding implemented in CAN controller chips, physical media attachment specifying Transceiver characteristics, physical media dependent sublayer which is application-specific and not standardized.

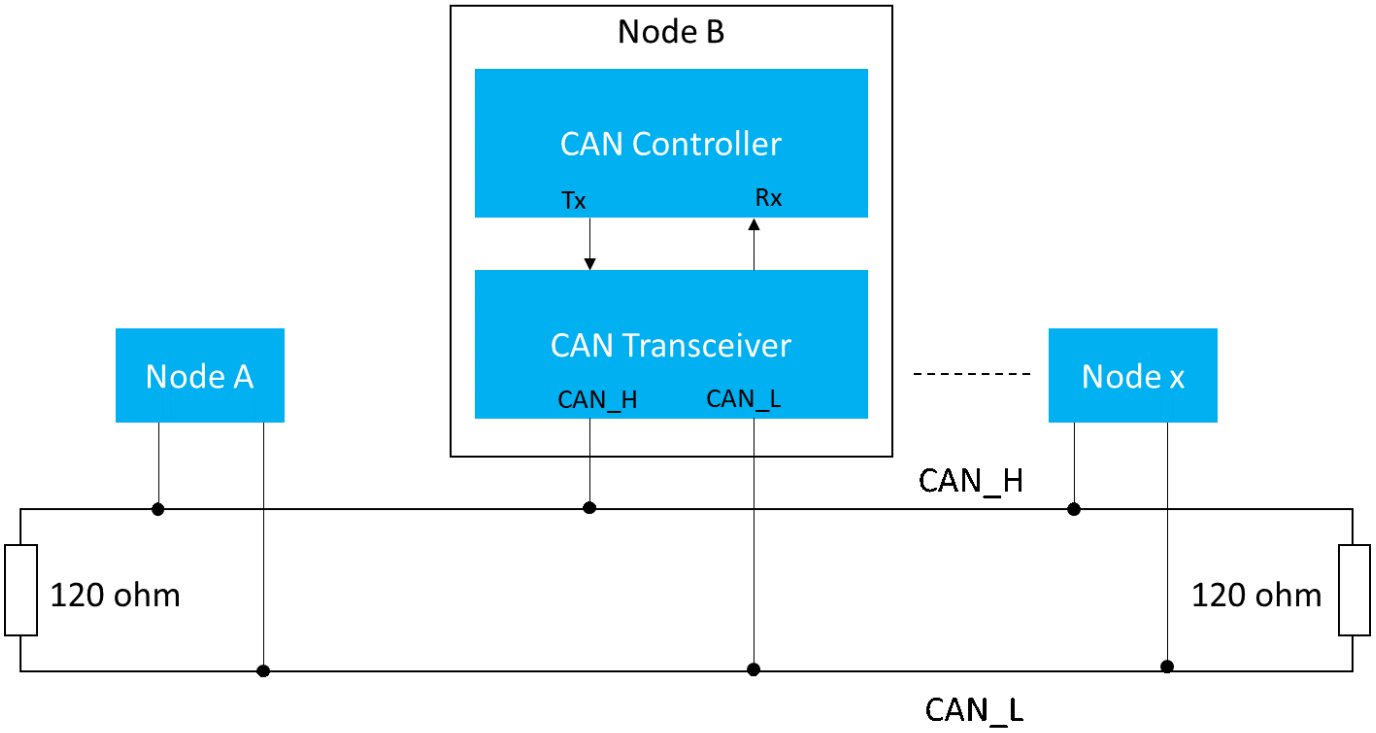


*Figure 5. CAN bus Wiring Diagram*

**The Physical coding sublayer**

The PCS comprises the bit encoding and decoding, bit timing. It provides an attachment unit interface to the transceiver chips and contains Tx and Rx pins. Bit-level errors are also handled through bit stuffing.

**Bit Timing**

Each bit of CAN bus is divided into four segments. Each segment is made up of Quanta. Segments:

1.Synchronization Segment
2.Propagation Segment
3.Phase Segment 1
4.Phase Segment 2

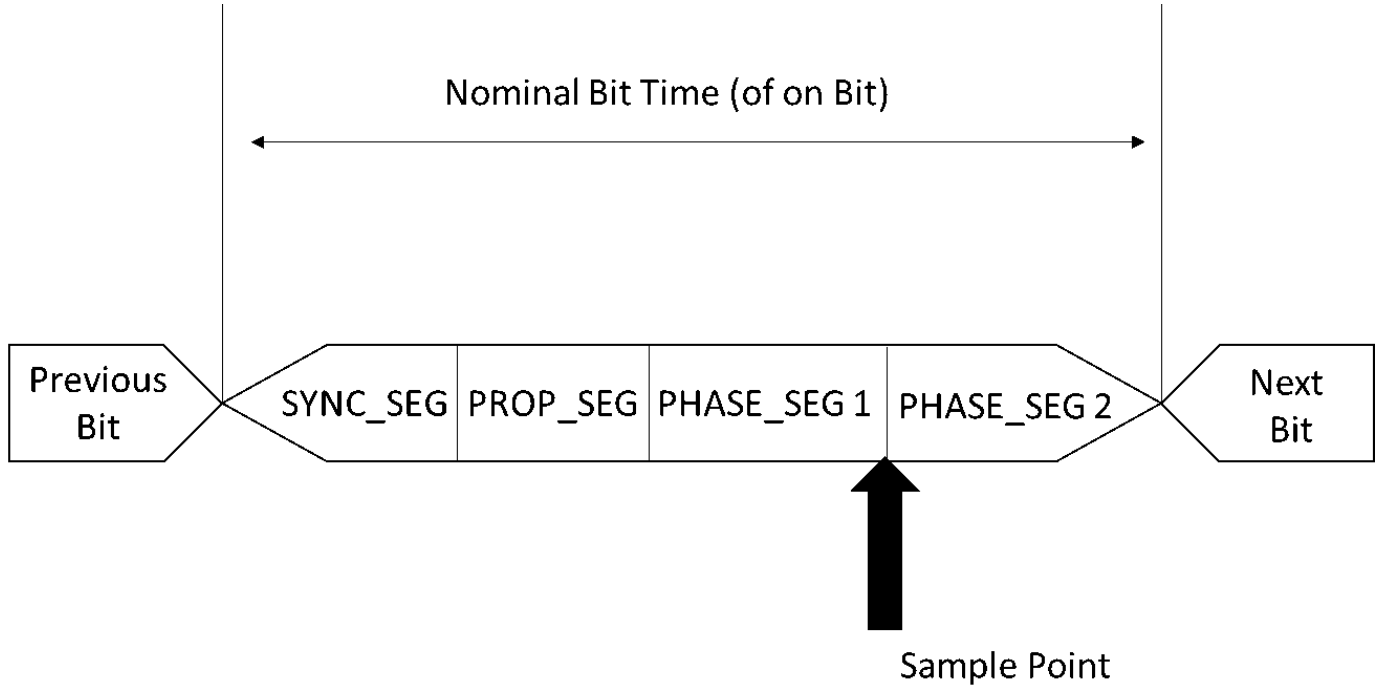


*Figure 6. CAN bit timing*

**Synchronization Segment**

- Segment is one quanta long
- Synchronization Segment is used to synchronize the nodes on the CAN bus system and for synchronization of clocks.
- A bit edge is expected to occur in this segment when the bus voltage level changes (Dominant to recessive or recessive to dominant)

**Propagation Segment**

Propagation segment is needed for the compensation of physical delays between the nodes

**Phase Segments**

Phase Segments lengthened or shortened based on the delay or early occurrence of bit edge outside of synchronization segment

### Sampling Point

- Sampling Point inside the bit period determines CAN bus voltage is either recessive or dominant.
- Represented in the percentage of the bit period and position is calculated from the starting point of the bit period and lies in between phase1 and phase 2.

### Handling Bit Level Errors

### Bit stuffing

CAN protocol follows NRZ encoding for transmission. The logic level does not change between bit interval. CAN requires a transition in logic level for re-synchronization. Hence 1 bit of opposite logic level is sent after 5 same consecutive bits. This is known as stuff bit and the receiver identifies it.

### Bit error

A node that is sending the bit always monitors the bus. If the bit sent by the transmitter differs from the bit value on the bus then an error frame is generated.

## The Physical media attachment Sublayer

This layer is implemented in the CAN Transceiver chip and gets input from the CAN Controller through Tx and Rx pins. Output drives the CANH and CANL lines. Transceivers are responsible for the different bit rates. CAN bus speed refers to CAN bus communication rate. The maximum CAN bus communication rate is 1Mbit/sec. For special applications some of the CAN controllers will handle higher speeds more than 1Mbit/sec. Low speed CAN communication rate is 125kbits/sec.

### The medium dependent sublayer

The medium dependent sublayer is highly application-specific. Standardization for pin assignment for different connectors belongs to this layer. Various Connectors are DB9, OBD II.

### CAN Bus DB9 Pinout

CAN bus is usually accessed via connector

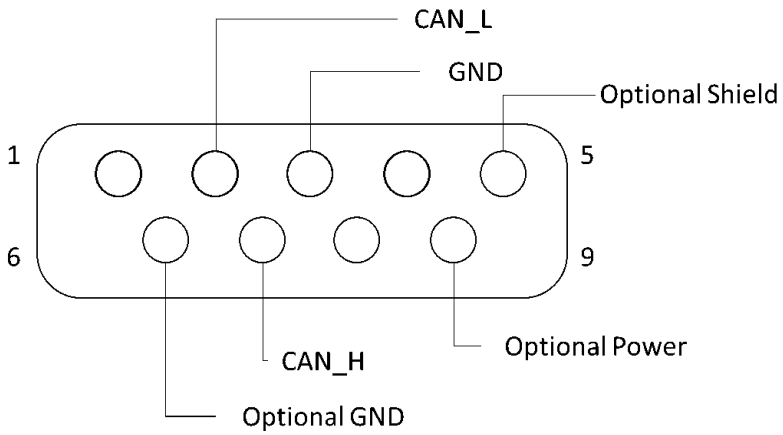


*Figure 7. CAN bus DB9 pinout*

Pin2: CAN_L
Pin3: CAN GND
Pin5: CAN_SHLD
Pin6: GND
Pin7: CAN_H
Pin9: CAN_V+

## CAN bus support in various Microcontrollers:

Microcontrollers should have CAN Hardware and Software which should provide CAN drivers to enable communication. Python-CAN libraries can also be used in the drivers which provide an abstraction to hardware components of Microcontroller and used to send and receive messages over the CAN network. Python CAN bus is also used to test the hardware and CAN bus data logging.

**CAN Bus shield for Arduino**

- CAN bus Shield adopts CAN bus controller with SPI interface and CAN transceiver and provides CAN bus capabilities to Arduino.
- Arduino with CAN bus helps to get the information like vehicle speed, fuel consumption, Temperature from the ECU's
- An Arduino CAN library is used to send and receive CAN messages over CAN bus.

**Raspberry Pi CAN bus:**

Raspberry Pi does not have specific hardware i.e. CAN controller and CAN transceiver to support CAN protocol. CAN bus was not supported by Raspberry Pi Software. Raspberry Pi supports CAN communication through the SPI interface.

Raspberry Pi is connected to an external CAN Controller supported by the board through the SPI interface and CAN controller is connected to CAN transceiver through Rx and Tx lines.

Examples of CAN controller: SJA100, MCP2515
Examples of CAN Transceiver: TJA1040, MCP2551

**STM32 CAN bus:**

Unlike Arduino and Raspberry Pi, STM32 has a CAN controller embedded in it. STM provides CAN bus drivers and those API's can be used when GPIO configured either to CAN1, CAN2.

## How to read CAN bus data?

Access to CAN bus data is possible when CAN bus is interfaced with external tools like Microchip CAN bus analyzer, CAN bus Wire shark through CAN USB adapter which gives instant connectivity to the USB port of computer or PC. CAN USB adapter is also controlled from anywhere via Ethernet, Internet, Intranet. CAN bus Wireshark is a tool used in Linux systems especially known from Ethernet network analysis that offers to display CAN messages by using SocketCAN which is a set of drivers and a networking Stack and hence called as Linux CAN bus. CAN to USB helps external tools to get the messages from the CAN network and the tools then used for monitoring and debugging the received or transmitted messages. But these messages are in raw format. Hence data collected from these data loggers are converted into scaled Engineering values using CAN bus decoders. Data collected from Data loggers can also be stored in the SD card which helps to control vehicle settings for more efficiency. Collected CAN bus data is useful in vehicle fleet management, R&D, Diagnostics, etc.

## Testing CAN bus with a multimeter

Testing is necessary to check any occurrence of CAN bus failures like a failure of wiring, ECU, the voltage supply to any one of the components in CAN network. CAN bus troubleshooting like adding 120-ohm terminator resistor at the physical end of CAN bus lines diagnoses the problem.

Ensure terminal resistor is 120 ohms and resistors are fitted and not broken by testing with a multimeter and also test the data being transmitted by switching multimeter to AC voltage.

## How to tell if a car has a CAN bus?

CAN bus equipped vehicles contain CAN bus led and CAN-BUS HID kit. CAN bus led communicates with the car advanced system and the Vehicle warns out when this Led is turned off. CAN BUS HID KIT acts as DC to AC Converter and helps to turn on lights initially with a high voltage current. Once the light has started, it requires a lower voltage current. But when HID uses low power, the CAN bus system assumes that the light is turned off and gives a warning. To avoid this condition HID conversion kit is used which communicates with the CAN bus system to tell that there is a working bulb. These warnings tell us that the car is equipped with a CAN bus.

## CAN Bus Hacking

CAN bus Hacking is a threat to consumers. Many wireless technologies are adopted in CAN bus vehicles like Bluetooth for attending calls or playing music. When onboard system access CAN bus in a car and are capable of Wi-Fi connectivity, it is easy for the hackers to get CAN bus access and be able to control the car. Wi-Fi hotspots are popular in cars, this allows people who know cars' IP address to track the car. This leads the Car manufacturers to secure transmitted data on the CAN bus network.

## Advantages of CAN bus Protocol

- Low cost because of reduced wiring
- Saves time due to simple wiring
- Auto retransmission of Lost messages
- Supports Error Detection
- Flexible Data transmission rates