CAN Protocol

Very Important in Automotive Embedded Systems

Written By,

IIT Bombay Alumni Foundation's

Embedded Technosolutions

This document will give the detail idea about CAN bus protocol, its functionality and different features of it.

This document is divided into mainly five sections as listed below,

1. Introduction to CAN
2. CAN layers

3. CAN Properties
4. Types of CAN
5. CAN messages

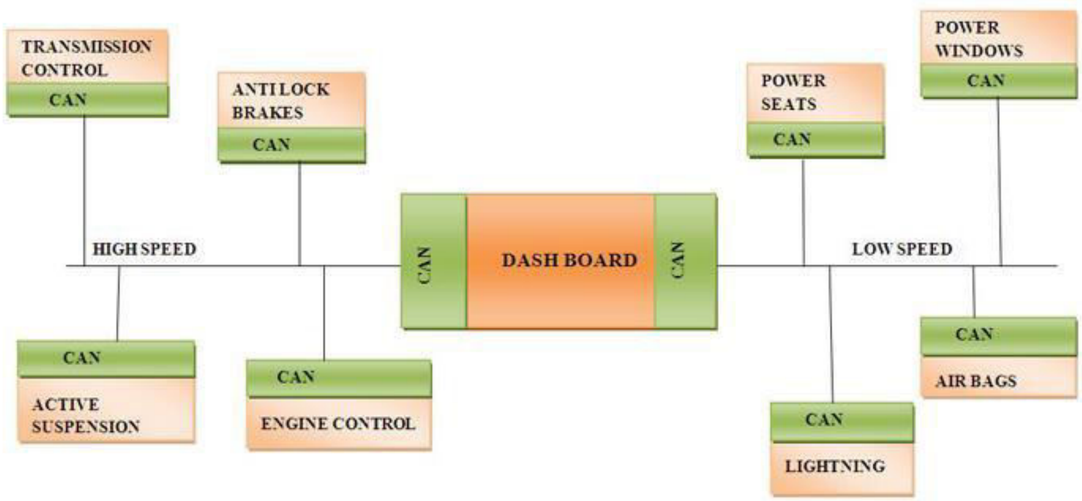
Introduction to CAN

Controller Area Network or CAN **protocol** is a method of communication between various electronic devices like engine management systems, active suspension, ABS, gear control, lighting control, air conditioning, airbags, central locking etc embedded in an automobile. An idea initiated by Robert Bosch GmbH in 1983 to improve the quality of automobiles thereby making them more reliable, safe and fuel efficient. With the developments taking place in the electronics and semiconductor industry the mechanical systems in an automobile were being replaced by more robust electronics system which had an improved performance. New technologies, products and inventions with added or improved functions started to shape a complete new era for the automobile industry which promised more robust vehicles with use of electronics. The increasing number of electronic devices used communication signals with more complex interrelations between them. Thereby making the life difficult for automobile engineers when they designed systems wherein one electronic device needs to communicate with others to operate. Realizing the problem of communication between different electronic modules Robert Bosch came up with this new protocol called CAN which was first released in 1986. CAN provide a mechanism which is incorporated in the hardware and the software by which different electronic modules can communicate with each other using a common cable.

Need for CAN

A vehicle contains a network of electronic devices to share information/data with each other. For example A spark ignition engine requires a spark to initiate the combustion chamber at the correct time so it communicates with engine control unit that adjusts the exact timing for ignition to provide better power and fuel efficiency.. Another example is of a transmission control unit that changes the ratio of gear automatically with the changing speed. It uses information from engine control unit and various sensors in the system. Every electronic device has an ECU/MCU (electronic/microcontroller control unit) with its own set of rules to share/transfer information. For two or more devices to interact they should have the necessary hardware and software which allows them to communicate with each other. Before CAN was introduced in the automobiles, each electronic device was connected to other device using the wires (point to point wiring) which worked fine, when the functions in the system were limited. The figure below is the pictorial view of the point to point wiring connection.

One of the major problems for automotive engineers was linking the ECUs of the different devices so that real time information can be exchanged. CAN protocol was designed to address this problem. It laid down the rules through which the various electronic devices can exchange information with each other over a common serial bus. This in turn reduced the wiring connections to a great extend thereby reducing the bulkiness and complexity of the system. The image below shows how the different devices are connected using the CAN protocol.



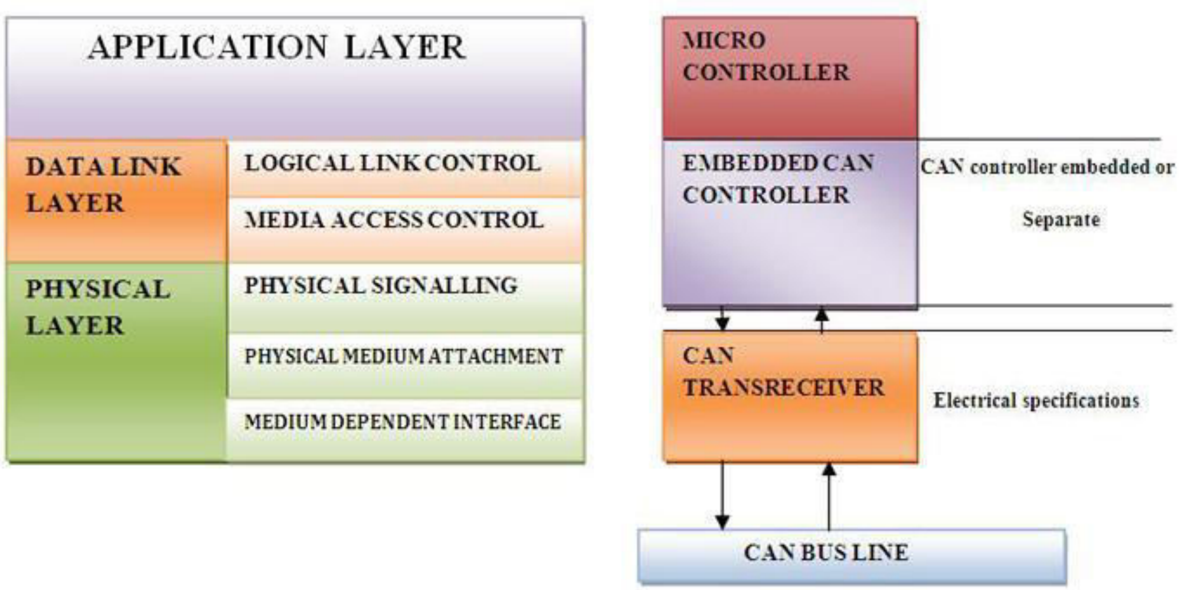
Also the standard technology of time, asynchronous transmitter/receiver did not support multi domain communications. Domain is a group of electronic devices that have almost similar requirements to work in the system. For example CD/DVD PLAYER, GPS system, monitors and displays etc. form a single domain. Similarly air conditioning and climate control, dashboards, wipers, lights doors etc. form another domain. Hence the electronic devices implanted in a vehicle can be classified under different domain. CAN facilitates multi-domain communication for the engineers.

Layers of CAN

The CAN functionality is divided into two layers,

1) Data link layer.

2) Physical layer.



Data link layer

Data link layer is subdivided into two layers

1) Logical link layer - Accept the messages by filtering process, overload notification and recovery management tasks will be taken care by this layer.

2) MAC (medium access control) layer - This layer will do the data encapsulation; frame coding, media access management, error detection and signaling and acknowledgment tasks.

Physical layer

This layer deals with the bit encoding and de coding, bit timing, synchronization Processes

CAN properties

- Multicast reception with time synchronization: Simultaneously multiple nodes can receive the frame.
- Multimaster : When the bus is free any unit may start to transmit a message. The unit with the message of higher priority to be transmitted gains bus access.
- Prioritization of messages: Depending on the importance of messages the priorities will be given to the different messages.
- Arbitration: Whenever the bus is free, any unit may start to transmit a message. If 2 or more units start transmitting messages simultaneously, which unit gets the bus access that will be depend on bitwise arbitration using the identifier. The mechanism of arbitration guarantees that neither information nor time is lost.
- System wide data consistency.

- Message Routing: An identifier names the content of a message. The identifier does not indicate the destination of the message, but describes the meaning of the data, so that all nodes in the network are able to decide by message filtering whether the data is to be acted upon by them or not.
- Remote Data Request: By sending a remote frame a node requiring data may request another node to send the corresponding data frame. The same identifier names the data frame and the corresponding remote frame.
- Error detection and signaling.
- Automatic retransmission of corrupted messages as soon as the bus is idle again.
- There is a distinction between temporary errors and permanent failures of nodes and autonomous switching off of defect nodes.
- Configuration flexibility: Any number of nodes can be added to or removed from the system without doing any modification in software or hardware parts.

- Bit rate: The speed of CAN message transfer may be different in different systems. But for a given system the bitrate is uniform and fixed.
- Single Channel: The bus consists of a single channel that carries bits. The way in which this channel is implemented is not fixed. E.g. single wire, two differential wires, optical fibres, etc.
- Bus values: The bus can have one of two complementary logical values: ‘dominant’ (‘0’) or ‘recessive’ (‘1’). During simultaneous transmission of ’dominant’ and ’recessive’ bits, the resulting bus value will be ’dominant’.
- Acknowledgment: All receivers check the consistency of the message being received and will acknowledge the transmitter by sending dominant bit in acknowledgement field in case of proper message reception
- Sleep/Wakeup mode: To reduce the power consumption CAN-device may be set into sleep mode without any internal activity and with disconnected bus drivers.

- Inter frame spacing data frames and remote frames are separated from preceding frames by inter frame spacing field.

Different Types of CAN

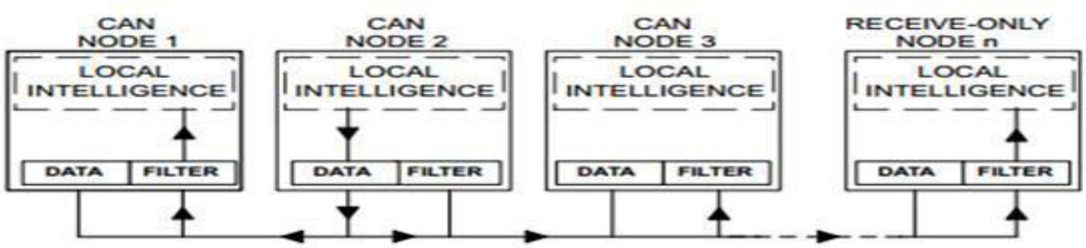
There are two types of CAN implementations depending in the size of the identifier field.

1) STANDARD: 11-bit wide identifier field.

2) EXTENDED: 29-bit wide identifier field.

CAN message types

A Message is packet of data that carries the information to be exchanged between the nodes. Each message in CAN has a unique identification number. The identification number is specified according to the content of the message and stored in message identifier. This identification number is also unique within the network so when the transmitting node places the data on the network for access to all nodes it checks unique ID number to allow the message to pass through the filter and rest are ignored. This is done to save the time spent on sorting. With message based protocol other nodes can be added without re-programming since the units connected to the bus have no identifying information like node addressing. So there is no change needed in the software and hardware of any of the units connected on the bus.



1. Data frame

Description - Carries data from a transmitter to the receivers.

Fields - Start of Frame, Arbitration Field, Control Field, Data Field, CRC Field, ACK Field, End of Frame.

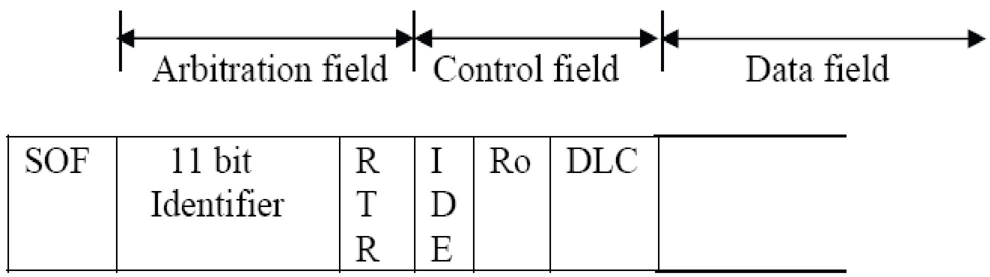
Format

Bits:	1	12/32	6	0-64	16	2	7	
	SOF	Arbitration Field	Control field	Data field	CRC field	ACK field	EOF	Inter frame space

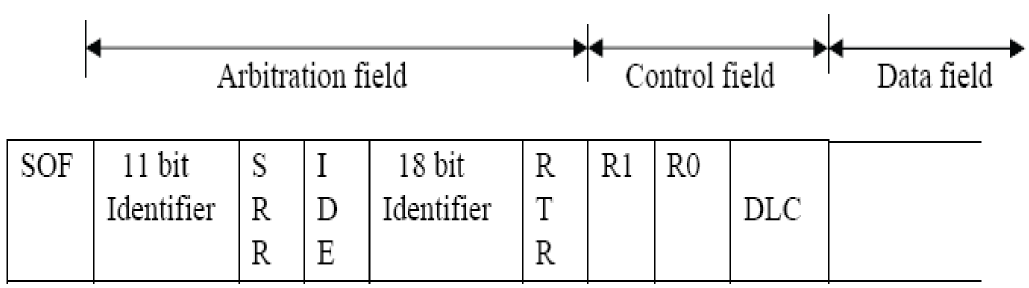
SOF : Start of frame bit. It is the indication of start of frame when a dominant bit is detected after the bus idle condition.

Arbitration : The arbitration field is depending on the type of frame.

STANDARD frame format: -



EXTENDED frame format: -



Arbitration field contains

- **Identifier field:** contains information's about message. In case of standard frame the length of this field is 11 bits and in case of extended 18 more bits are added to it .So in case of extended frame total size of identifier field it is 29 bits.
- **SRR:** one bit wide. Used in case of extended frame. SRR bit will be recessive and in case standard frame this bit replaces the RTR bit. .
- **RTR:** one bit wide. Indicates that incoming frame is a data frame or a remote frame depending on the value of this bit. If it's a dominant bit implies incoming bit stream is of data frame else if recessive then remote frame.
- **IDE:** one bit wide. Indicates that the incoming frame is of standard format if dominant bit is received in this field or else extended one (if recessive).

Control field : six bit wide. Two bits r0 and r1 are reserved. In standard frame r1 is replaced by IDE bit. And remaining 4 bits called as data length control (DLC) describes total length of data field.

Data field: contains the actual information within the limit of {0…8} bytes.

CRC field: 16 bit wide. Divided into two parts:

CRC sequence: 15 bit length. Calculated over SOF to data field.

CRC delimiter: 1 bit recessive bit.

ACK field: 2 bit wide and contain two fields ACK flag and ACK delimiter.

ACK flag: being successful reception of frame the receiver will indicate the transmitter by putting a dominant bit in this place. Otherwise recessive.

ACK delimiter: A recessive bit.

END OF FRAME: 7 bit wide. Consists of 7 recessive bits.

Inter Frame Space: 3 bit wide. Two continuous frames are separated by interframe pace.Normal interfame bit values should be recessive. In case of other values the next state will be predicted depending on the bit position. Detailed discussion of this will be done in later part of this tutorial.

2. Remote frame

Description - Transmitted by a bus unit to request the transmission of the data frame with the same identifier.

Fields - Start of Frame, Arbitration Field, Control Field, CRC Field, ACK Field, End of Frame.

Format

Bits:	1	11/29	6	16	2	7
	SOF	Arbitration field	Control field	CRC field	ACK field	EOF

Remote frame is similar to data frame except that it will not contain data field. And RTR bit is used to indicate the remote frame.

3. Error frame

Description - Any unit on detecting a error transmits an error frame.

Fields - Error flag and Error delimiter.

Bits:	6	8
	Error flag	Error delimiter

Format

As shown in above figure there will be two fields in error frame, error flag and error delimiter.

There are two types of error frames are there in CAN

1) Active error frame: Error flag -6 dominant bits.
2) Passive error frame. Error flag -6 recessive bits.

And in both the cases all the bits of error delimiter will be all recessive. Depending on the node internal error count register condition the respective type of error frame will be transmitted.

Types of error

There are five types of error are there in CAN and are listed below. And in case any one of these error is observed the error frame is transmitted.

- **Bit error**: During transmission the node transmits the bit at transmit time region and receives the bit at receive time and two bits are compared if they are not equal then that is considered as bit error.
- **Stuff error:** In a frame if continuous 5 recessive or dominant bits are transmitted, the sixth bit should be of opposite to that. While receiving if continuous 5 recessive or dominant bits are received then the next incoming bit is of same value that of previous then it will be considered as stuff error
- **CRC error**: While receiving data or remote frame the CRC value will be calculated and is compared with the received CRC values. In case of proper frame transmission both will be equal. Else that will be considered as CRC error and error frame will be transmitted.
- **Form error:** Form error will be related to the error in forming the frame. For e.g. delimiter (CRC, ACK) missing or improper end of frame, these conditions are taken as form error and immediately error frame is transmitted.
- **Acknowledgement error:** During the transmission of data or remote frame, in the ACKs flag field the transmitter will put recessive bit and expect dominant bit

from receive pin. If dominant bit is observed then that is considered as proper transmission. In other case it is acknowledgement error. The error frame is transmitted if acknowledgment error is observed.

Depending on node state the active error frame or passive error frame is transmitted. The error frame is transmitted immediately if error found is of type other than CRC error. And in case of CRC error the error frame will be transmitted immediate next bit of ACK delimiter.

4. Overload frame

Description - Detection of any one of overload condition make node to transmit overload frame.

Fields - Overload flag and Overload delimiter.

Format

Bits:	6	8
	Overload flag	Overload delimiter

As shown in above figure there will be two fields in overload frame, overload flag and overload delimiter.

Conclusion

CAN protocol initially developed for in-vehicle networking of automobiles has expanded its applications in various other industries. The application started for luxurious cars is now being used in heavy duty vehicles like trucks, buses, trains and rail vehicles. The unique feature of CAN that allows various electronic units to communicate with each other made it important in healthcare domain. For example intensive care units and operating rooms where time and communication is of utmost importance. Entertainment industry also used CAN protocols to improve features in studios to control lights and door system and to control stage of theatres, event halls etc. Gambling machines and toys are other examples in entertainment field. In the field of science the high energy experiments and astronomical telescope use CAN in embedded network.