International Journal of Modern Trends in Engineering And Research
www.ijmter.com
e-ISSN No.:2349-9745, Date: 2-4 July, 2015

Communication over Automobile Instrument Assembly using CAN Bus –A Review

Nakade M. J.[1], Shrivastava K. N.[2], Dhumal S. B.[3], Wakle R. P.[4]

[1]*Department of Electronics and Telecommunication,*
Maharashtra Institute of Technology, Aurangabad
[1]*mayurnakade009@gmail.com,* [2]*kritishrivastava29@gmail.com,*
[3]*sanjaydhumal26@gmail.com,* [4]*rattuu13@gmail.com*

Abstract: The modern automobile has progressed greatly in the last few years. Just as our everyday consumer life has become sophisticated, our transportation has followed suit. Intensive electronification and higher complexity of vehicles necessitates the use of data buses in vehicles. These buses are used for information exchange between control units (ECUs) containing commonly information about a vehicle operating parameters such as vehicle speed, engine speed, fuel consumption etc. CAN bus (Controller Area Network) is the most widely used internal digital network in vehicle on which data transmission takes placed using CAN protocol specified by the CAN standard ISO-11898. This paper will give the detailed analysis of how communication over automobile instrument assembly takes placed using CAN Bus.

Keywords: Automobile Instrument Assembly, CAN Bus, IVI, ECU, Node, CSMA/CA access, Bit arbitration, CAN transceiver.

I. INTRODUCTION

The vehicles have become the indivisible asset of every individual's life. Now days we see high tech gadgets, high speed internet, high speed computers and fancy entertainment systems etc. these concepts are all in the modern car. Introduction of IVI (In-Vehicle Infotainment) is a good example [10]. Increasingly complex electronic systems provide for a high level of safety and comfort in car driving. The CAN (Controller Area Network) serial bus system makes a crucial contribution here with to its specific properties. It assures reliable data exchange even under harsh environmental conditions for example. CAN bus (Controller Area Network) is the most widely used internal digital network in vehicle on which data transmission takes placed using CAN protocol specified by the CAN standard ISO-11898 [11]. Information on the CAN bus is in the form of data messages whose structure is defined by the CAN protocol. The data on the communication line contain commonly information about a vehicle operating parameters such as vehicle speed, fuel consumption, engine performance etc.

Electronic system and those function requires more intensive exchange of information. Using a digital network controller (ECU) can efficiently and with the desired speed of data exchange can be possible. ECU is an Electronic Control Unit consisting of CAN controller and micro-controller to which various I/O devices as sensors, actuators are attached, identifies and processes the required data. Modern motor vehicles can contains over hundred ECUs which are interconnected by the CAN Bus network [5]. Some of the ECUs used are: Engine Control Module (ECM), Electronic Brake

Control Module (EBCM), Body Control Module (BCM), Telematics Module, Remote Control Door Lock Receiver (RCDLR), Heating, Ventilation, Air Conditioning (HVAC), Theft Detection Module (TDM) etc. The structure of automobile instrument assembly could be well illustrated by following diagram (refer Figure 1) showing In-Vehicle CAN Bus Network joining various ECUs.

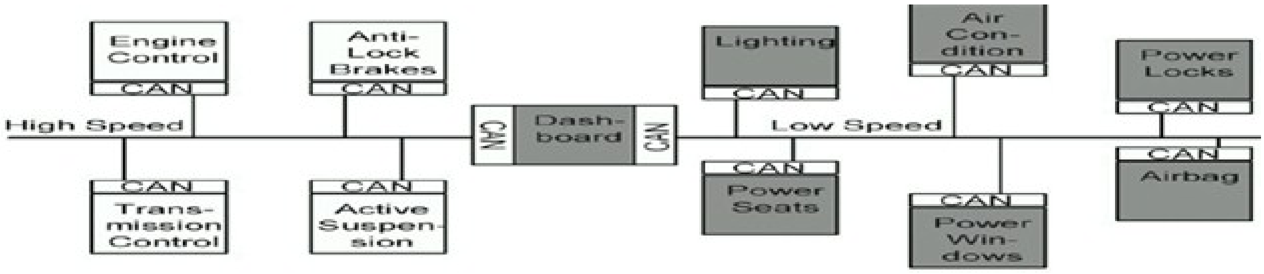

Fig 1. System Model

II. CAN ARCHITECTURE

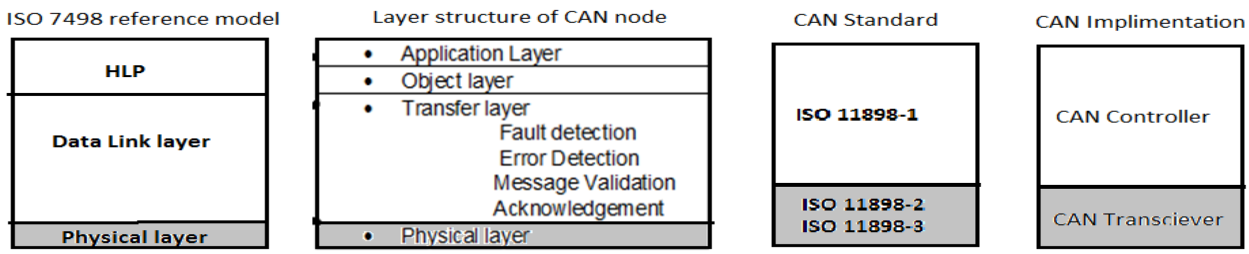

Fig 2. CAN Architecture

The CAN controller implements only three layers of the ISO/OSI Reference model. It mainly comprises of physical layer and data link layer (as shown in Figure 2) [3], [9] in order to limit the resources and improve the performance. The other layers i.e. Layers 3 to Layers 6 are implemented in higher layer protocols like CANOpen, J1939 and DeviceNet. The first white shaded part contains the CAN protocol that covers the entire data link layer (framing, addressing, bus access, data assurance-Fault detection, Error detection, Message validation, Acknowledgement) and gray shaded part of the physical layer (physical signaling) of the standardized reference model for data communication (ISO 7498). The physical layer and data link layer are integrated on the CAN controller chips. ISO 11898 restructured into two parts; ISO 11898-1 which covers the data link layer, and ISO 11898-2 which covers the CAN physical layer for high-speed CAN. ISO 11898-3 was released later and covers the CAN physical layer for low-speed, fault-tolerant CAN.

III. INFORMATION EXCHANGE

The entire communication scheme in the vehicle over the CAN bus could be well understand by organizing into two parts: Data link layer covering CAN protocol and Physical layer that are standardized by CAN standard 2.0 [8]. In Data link layer limelight is fallen on topic as CAN message frames, Data Transmission - CSMA/CA, synchronization, Error detection and Fault confinement while in Physical layer limelight on CAN bus signal levels and bus topology, CAN Node and CAN Transceiver module.

3.1 Data Link Layer

A. CAN Massage Frame

An important difference between CAN and Ethernet, for example, is that CAN does not specify a transmitting station or node, only a message so Message Oriented Message Broadcasting protocol. As a result, an Identifier Field is included in each message identifier has to be unique within the whole network and it defines not only the content but also the priority of the message. CAN protocol defines four different types of message frames: Data Frame, Remote Frame, Error Frame and Overload Frame [1], [2].

Data Frame is used when a node transmits information to any or all other nodes in the system. There are two formats to this frame; Standard and extended. The standard frame has an 11 bit id in the arbitration field that allows up to 2048 unique messages onto the BUS and the extended frame has a 29 bit id that allows up over 536 million unique messages. Remote frames are used when one node needs to request data from another node. Remote Frame is basically same as Data Frame with no data payload and RTR is being set 1. Error frame and Overload frame these two frame types are for handling errors. Error Frames are generated by nodes that detect any one of the many protocol errors defined by CAN. Overload errors are generated by nodes that require more time to process messages already received.

SOF	Identifier	RTR	Control
One dominant bit	11 OR 29 BITS	1-Bit	4-Bit
DATA	CRC	ACK	EOF
0-8 bytes	15- bits	2- bits	7 recursive bits

Fig3. CAN Message Frame

Above Figure 3 represents CAN message frame in tabular form as above. The First field of 12 bits - Arbitration field consist of first 11-bit that specifies destination device address and a RTR bit (Remote Transmission Request). If RTR bit is at '1', it means this packet is for the device at destination address. If this bit is at '0' (dominant state) it means, this packet is a request for the data from the device. Second field of 6 bits - Control field whose first bit IDE is for the identifier's extension, second bit is always '1' and the last 4 bits specify data length code. Third field of 0 to 8byte i.e. upto 64 bits is the data field containing actual data whose length depends on the data length code in the control field. Fourth field (third if data field has no bit present) of 16 bits is CRC (Cyclic Redundancy Check) bits. The receiver node uses it to detect the errors, if any, during the transmission. Fifth field is of 2 bits first bit 'ACK slot'. Sender sends '1' in ack field and receiver sends back '0' in this slot when the receiver detects an error in the reception. While transmission sender after sensing '0' in the ACK slot, generally retransmits the data frame and second bit 'ACK delimiter' bit, it signals the end of ACK field. The error frame consists of two different fields: The first field is of ERROR FLAGS formed of 6–12 dominant/recessive bits generated by node and second field is the ERROR DELIMITER formed of 8 recessive bits. Overload Frame is a special error frame having format similar to that of error frame and is not used very often.

B. Data Transmission over CAN

CAN uses CSMA/CA bus access and Bus Arbitration principle for the data transmission. Every module (node) on the network is capable of sending and receiving signals and has its own unique address on the network. Bus access begins when a CAN node wishing to send first listens to the CAN bus (Carrier Sense – CS). The CAN specification defines two bus states: dominant and recessive based on priority. Dominant Bus Level = 0, Recessive Bus Level = 1. The dominant bit has given priority over recessive bit. If there are multiple CAN nodes wishing to send, bitwise arbitration prevents collisions from occurring in spite of simultaneous bus access (Multiple Access – MA). Based on the ID CAN bus uses bus arbitration principle not only to prevent collisions (Collision Avoidance – CA), they also provide priority-controlled bus access [3], [9].

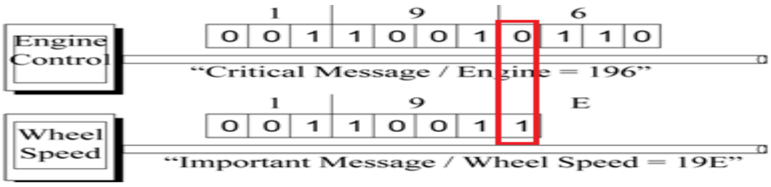

Fig4. Bus Arbitration Principle

There is no explicit address in the messages, instead, each message carries a numeric value which controls its priority on the bus, and may also serve as an identification of the contents of the message. When several messages access the bus at the same time, the one with the higher priority "wins" the arbitration. The identifier with the lowest binary number has the highest priority. "Losers" automatically become receivers and they won't try to send another message until the bus becomes available again and so is called as non-destructive arbitration which could be well illustrated by example stated above (refer Figure 4). CAN uses bit coding according to "Non-Return-to-Zero" principle for the synchronization as NRZ provides highest transport capacity. But NRZ format also violets when continuous 0's or 1's are to be transfer then "Bit-Stuffing" is also used.

C. Error Detection and Fault Confinement

There are following types of errors [1] occurs as

i. Stuff Error: It is generated when more than 5 Bits of same polarity outside of "bit-stuffed" segment are present.
ii. CRC Error: If the sum of received CRC sequence and that of transmitted sequence is not same then CRC error occurs.
iii. Form Error: Occurs when violation of fixed format Bit fields.
iv. Acknowledgement Error: occurs when transmitted message receives no acknowledgement. ACK confirms only the successful transmission. It is used for error confinement.

After the error detection an Error Frame is generated which tend to repeat original message after a proper intermission time. CAN guarantees proper network operation even in cases where malfunctioning nodes produce continuous error condition. CAN error detection can pinpoint to "perpetrator" which distinguish between temporary and permanent node failures. After identification of such nodes CAN prepares to removal (self-retirement) of malfunctioning nodes from the bus.

3.2 Physical Layer

CAN specifies following Bus Lines

i. High Speed CAN - It uses 2 wires and a transfer rate of up to 1 Mb/s (dependent upon wire length). It is the most commonly used physical layer
ii. Low-Speed/Fault-Tolerant CAN Hardware - It uses 2 wires and a transfer rate of up to 125 kbit/s. It is used in door wiring and in brake lights
iii. Single-Wire CAN Hardware - It uses 1 wire and a transfer rate of up to 33 kbit/s. It is used in comfort devices (mirrors, seats, etc).

Message length is short with a maximum of 8 data bytes per message and Max. transfer rate of 1000 kilobits per second at a maximum bus length of 40 meters or 130 feet when using a twisted wire pair which is the most common bus medium used for CAN.

A. Bus Signal Level and Bus Topology

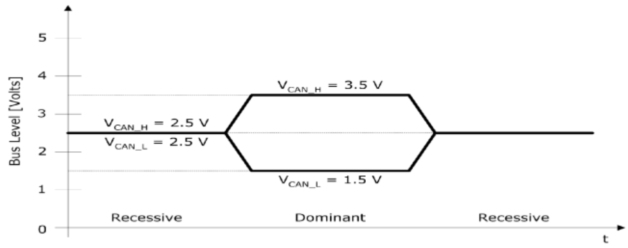

Fig 5. CAN Bus Levels

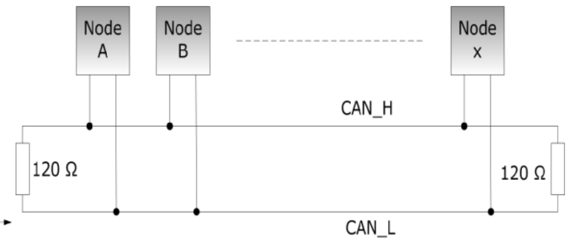

Fig 6. CAN Bus Topology

CAN bus uses two dedicated wires for communication. The wires are called CAN high and CAN low. When the CAN bus is in idle mode, both lines carry 2.5V. When data bits are being transmitted, the CAN high line goes to 3.75V and the CAN low drops to 1.25V due to the resistance of 120ohm used at the termination as shown in above in Figure 6, thereby generating a 2.5V differential voltage between the lines. A dominant bit is represented by CAN_H going to about 3.5 V and CAN_L going to about 1.5 V (refer Figure 5) [8], [11]. Since communication relies on a voltage difference between the two bus lines, the CAN bus is NOT sensitive to inductive spikes, electrical fields or other noise. This makes CAN bus a reliable choice for networked communications on mobile equipment. CAN power can be supplied through CAN bus or a power supply for the CAN bus modules can be arranged separately.

B. CAN Node and Transceiver Module

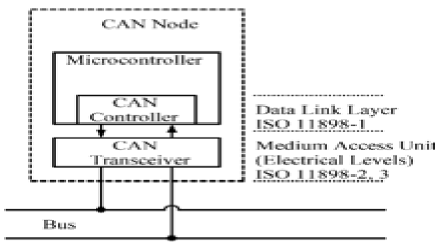

Fig 7. CAN Node

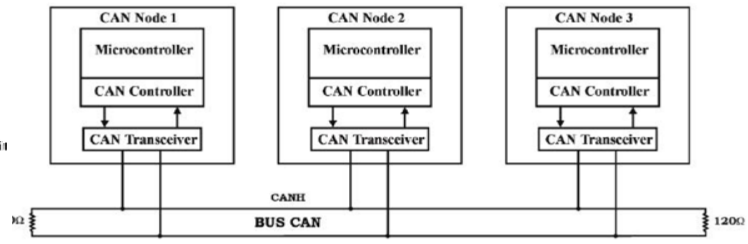

Fig 8. CAN Transceiver Module

Each node consist of Central processing unit i.e. microprocessor, or host processor, CAN Controller and CAN transceiver chip [12]. The host processor decides what the received messages mean and what messages it wants to transmit. Sensors, actuators and control devices can be connected to the host processor. CAN controller often an integral part of the microcontroller. The CAN controller stores the received serial bits from the bus until an entire message is available, which can then be fetched by the host processor. The transceiver is a transmitter and receiver amplifier. It converts the serial bit stream of the CAN module into electrical voltage values and vice versa.

CONCLUSION

When taken into account that CAN is still at the beginning of a global market penetration, even conservative estimates show further growth for this bus system for the next ten to fifteen years [8]. So the study communication over automobile instrument assembly using CAN bus makes major significance here. This paper presented the complete overview of how communication takes placed reliably using CAN in the automobile instrument assembly.

REFERENCES

[1] B. Praveen Kumar, V.Dhana Raj, B V Satyanarayana,Dr. N S Murthy Sarma, "Design of Automatic Automobile using CAN Bus", International Journal of Research in Computer and Communication technology, IJRCCT, ISSN 2278-5841, Vol 1, Issue 7, December 2012.
[2] Renjun Li, Chu Liu and Feng Luo, "A Design for Automotive CAN Bus Monitoring System", IEEE Vehicle Power and Propulsion Conference (VPPC), September 3-5, 2008.
[3] Mayer, E., "Serial bus systems in the automobile – Architecture, tasks and advantages", Elektronik Automotive 7/2006, pp. 70ff.
[4] Jukl M., Čupera J., "Analysis of Operating Parameters of the Vehicle Via CAN-Bus", MENDELNET 2013
[5] Benjamin C. K., M. Farid Golnaraghi, "Automatic Control Systems, Eight Edition", John Wiley & Sons, Inc. 2003.
[6] Ashwini S. Shinde, Prof. Vidhyadhar B. Dharmadhikari, "Controller Area Network for Vehicle Automation", International Journal of Emerging Technology and Advanced Engineering (ISSN 2250-2459), Vol 2,, Feb 2012
[7] Wilfried Voss, A Comprehensive Guide to Controller Area Network, Copperhill Media Corporation, 2005-2008.
[8] http://en.wikipedia.org/wiki/CAN_bus
[9] www.vector-informatik.de
[10] http://www.interfacebus.com/CAN-Bus-Description-Vendors-Canbus-Protocol.html
[11] http://www.can-cia.org
[12] http://www.sae.org

