Vehicle Automation Using Controller Area Network

Bhagyashri U.Wani [1], Dr. Sunita P. Ugale [2]

[1]*Post Graduate Student of K.K.Wagh Institute of Engg. Education & Research, Nashik , Maharashtra, India.*
[2]*Associate professor, Department of Electronics Engineering, K.K.Wagh Institute of Engg. Education & Research.*

---***---

Abstract - *As per the high speed requirement in modern vehicle, Controller Area Network (CAN) architecture has been implemented. In order to reduce more numbers of wiring complexity for point to point connection in vehicle automation, CAN is suggested for data communication media within the vehicle environment. CAN have advantage of high speed communication & increased flexibility. This paper describes the ARM7 based design and implementation of CAN Bus prototype for vehicle automation & its advantages with respect to traditional protocol interface for vehicle automation like LIN, Zigbee, Flexray.*

Key Words: **Vehicle Automation, Controller Area Network (CAN), LIN.**

1. INTRODUCTION

For better performance we require automation in vehicle. In Vehicle system automotive electrical architectures consists of electronic control units (ECU) carries a variety of control functions. In vehicle system greater safety, more comfort, convenience, pollution control and less fuel consumption are main concern. In modern vehicle many electronic control units (ECU) are present for various subsystems. these are airbags, antilock braking, engine control, audio systems, windows, doors, mirror adjustment etc. Subsystems are independent or dependent & communications among dependent sub systems is essential.

Traditional systems also satisfy the main need of vehicle automation but they lag to reduce wiring complexity & reliability issue. In vehicles networking protocols must satisfy requirements which include, significant reduction of wiring complexity, reducing body weight and costs, improving the efficiency of fault diagnosis, low latency times and configuration flexibility and enhancing the level of intelligent control [4].

Different protocols are being used in vehicle automation like flexray, zigbee, LIN, CAN. They vary in speed, reliability & flexibility. In the proposed work CAN bus protocol is used for vehicle automation. Various protocols are described below;

1.1 Zigbee:

It is wireless protocol. It is having speed of 250Kbps over the range 50 meter. It can be used for the automation in vehicle. It is designed for low power consumption so that Batteries can last for month & years. It is used in tyre pressure monitoring system as communication media[1]. But this protocol is not suited for large distance communication & not that much reliable as Controller area network.

1.2 LIN:

Local Interconnect Network is a serial network protocol used for communication for the low cost, low speed vehicle network. for the low speed and low cost applications LIN bus can be used in combination with CAN bus due to which reliability of whole system increases. It is having speed of 20Kbps at 40meter length. LIN provides cost efficient communication in application where the bandwidth & versatility of CAN are not required.

Its target application is low speed system, such as switch type equipment and position type system which are including control of the rearview mirrors, locks, car seat, windows etc. LIN bus is having Master/Slave structure, using single wire communication it decreases the weight and wires greatly [5]. To increase the reliability of system LIN should be used in combination with CAN.

1.3 FlexRay:

It is high speed of 10Mbps protocol. This is emerging bus for high speed synchronized data communication in advanced system such as active suspension, high performance powertrain safety. Benefits of the application of FlexRay in automotive control are larger CRC, dual redundancy, fast data rate which allows faster real-time control loops [5].As this is too much expensive protocol, so CAN is preferred over it. To increase the reliability of system Flexray should be used with CAN protocol.

2. CAN protocol Overview

CAN or Controller Area Network or CAN bus is an ISO standard computer network protocol and bus standard, designed for microcontrollers and devices to communicate with each other without a host computer. It is designed earlier for industrial networking but recently it is more adopted for automotive applications, CAN have gained widespread popularity for embedded control in the areas like industrial automation, automotives, mobile machines, medical, military and other harsh environment network applications. Development of the CAN bus started originally in 1983 at Robert Bosch GmbH.

CAN supports data frames of size only up to 8 bytes. these 8 bytes will not take the bus for a long time, so it ensures real-time communication. CAN uses 15-bit CRC which makes CAN very secure and reliable. The communication rate of CAN based network, depends on the physical distances between the nodes. If the distance is less than 40m, the rate can be up to 1Mbps. CAN bus Protocol has the following properties:

1.It support multi-master communication.
2. It has configuration flexibility.
3. It gives guarantee of latency time.
4. It provides error detection.
5. Messages can be prioritized.

2.1 CAN Frame Format

CAN use a specific message frame format for receiving and transmitting the data. The two types of frame format available are:

A)Standard CAN protocol or Base frame format
B) Extended CAN or Extended frame format

figure1 illustrates the standard CAN frame format, which consists of seven different bit fields.

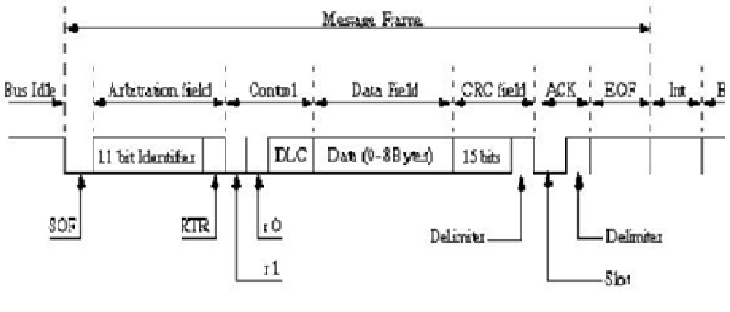

Fig -1: CAN data frame

a) A Start of Frame (SOF) field indicates the beginning of a message frame.
b) An Arbitration field contains a message identifier and the Remote Transmission Request (RTR) bit. The RTR bit is used to distinguish between a transmitted Data Frame and a request for data from a remote node.
c) A Control Field contains six bits in which two are reserved bits (r0 and r1) and a four bits are Data Length Code (DLC). The DLC indicates the number of bytes in the Data Field that follows.
d) A Data Field, consists from zero to eight bytes.
e) The CRC field, contains a fifteen bit cyclic redundancy check code and a recessive delimiter bit.
f) The Acknowledge field, consisting of two bits. The first one is a Slot bit which is transmitted as recessive, but is subsequently written by dominant bits transmitted from any node that successfully receives the transmitted message. The second bit is a recessive delimiter bit.
g) The End of Frame field, consists of seven recessive bits.

2.2 Abstraction layer in CAN protocol:

1.Physical layer:

CAN bus originally specified the link layer protocol with only abstract requirements for the physical layer. This layer defines physical and electrical characteristics of the network. Physical layer is same for all the nodes on the same network. The physical layer defines how the signals are transmitted. Tasks include signal level, bit representation and transmission medium.

2. Data link layer:

This has two sub layers, logical link control sub layer and medium access sub layer (MAC). Logical link control sub layer accept the messages and recovery management. Medium access sub layer (MAC) does message framing, acknowledgment, error detection and signaling.

3.Application Layer:

It is a user interface responsible for displaying received information to the user. This layer specifies the shared protocols and interfaces methods used by hosts in a communication network.CANOPEN, Device- Net, SAEJ1938 are the implementation of CAN application layer.

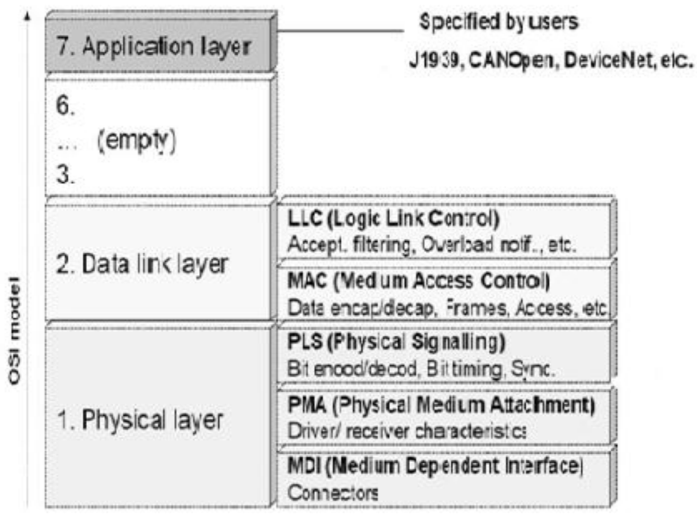

Fig -2: Hierarchical structure of CAN protocol

2.3 Error detection & correction:

This mechanism is used for detecting errors in messages appearing on the CAN bus, so that the transmitter can retransmit message. The CAN protocol has five different ways of detecting errors. Two of these work at the bit level, and the other three at the message level.

1. Bit Monitoring.

2. Bit Stuffing.
3. Frame Check.
4. Acknowledgement Check.
5. Cyclic Redundancy Check.

2.4 Error confinement:

Error confinement is a unique technique in CAN which provides a method to distinguish between temporary errors and permanent failures in the communication network. Temporary errors caused by spurious external conditions. Permanent failures are caused by bad connections, faulty cables, defective transmitters or receivers.

2.5 CAN bus communication:

The CAN is a "broadcast" type of bus. That means there is no address of sending or receiving node. All the nodes in the network can receive or transmit the message, For receiving the message each node perform acceptance test. the messages checked by each node whether it is relevant to that particular node or not. If message is relevant then it is accepted by that node otherwise ignored. For transmission of message the priority node can send first. That priority depend on 11 bit identifier. an identifier that is unique throughout the network is used to label the content of the message. 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. If the bus is free, any node may begin to transmit. But in situations where two or more nodes attempt to transmit message (to the CAN bus) at the same time then identifier field, which is unique throughout the network helps to determine the priority of the message. A "nondestructive arbitration technique" is used to fulfill this, to ensure that the messages are sent in order of priority and that no messages are lost. The lower the numerical value of the identifier, the higher the priority. That means the message with identifier having more dominant bits (i.e. bit 0) will overwrite other nodes' less dominant identifier so that eventually (after the arbitration on the ID) only the dominant message remains and is received by all nodes.

CAN do not use address based format for communication, instead uses a message based Data format.

3. Proposed System

The proposed system shown in figure 2 It consists of Master node is near driver & second is slave node which has sensors connection. LPC 1768 (ARM cortex M3 core) processor used here. The given system used for automation in vehicle lightening [2]. Here the high speed CAN protocol used for communication between two processors.

A. Transmitting and receiving node(LPC 1768):

LPC 1768 processor is used for this purpose. It supports at frequencies up to 120 MHz. The architecture features a multilayer AHB bus that supports multiple high bandwidth data streams running simultaneously from peripherals such as Ethernet, USB, or CAN. It consist of inbuilt 2.0B CAN controller .It has 512 KB Flash memory[3].

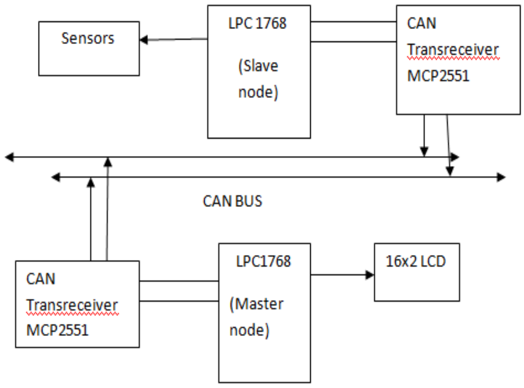

Fig -3: Block Diagram

3.1 CAN transreciever MCP2551:

The MCP2551 is the interface between a CAN protocol controller and the physical bus. It has protective circuitry for CAN controller & adapts signal level from the bus to CAN controller expected level. It converts the signal from CAN controller to signal which transmit on bus.

MCP2515 implements the CAN specification, version 2.0B. It is capable of transmitting and receiving both standard and extended data and remote frames with 0 – 8 byte length of the data field. The MCP2515 has two acceptance masks and six acceptance filters that are used to filter out unwanted messages, thereby reducing the host MCUs overhead.

3.2 Sensors:

As the proposed system is used for Automation vehicle lightning application, rain detection sensor & LDR sensor will be used for Automatically Fog ON/Off light. In proposed system for daylight & night light detection, LDR will be used, along with Rain detection sensor . these sensors will be interfaced with slave node LPC1768 processor. The sensory data transmit through CAN transreciever of slave node towards the Master node LPC 1768. Master node will take decision & that transmit towards slave node. As per the signal send by master node, slave node will ON or OFF the FOG light.

4. CONCLUSIONS

Here high Relaibility,Fexibility & Error handling capacity is achieved in vehiclesusing CAN communication . The high speed CAN protocol can effectively implement automation like vehicle lightening.

REFERENCES

[1] G.Petchinathan, K.Srinivasa Sricharan, “AUTOMATED TYRE PRESSURE MONITORING AND REGULATING SYSTEM” IEEE International Conference on Vehicular Electronics and Safety (ICVES), December 16-17,2014.

[2] Yali Guo; Qinmu Wu; Honglei Wang, "Design and implementation of intelligent headlamps control system based on CAN bus," in Systems and Informatics (ICSAI), 2012 International Conference on , vol., no., pp.385-389, 19-20 May 2012R. Nicole, “Title of paper with only first word capitalized,” J. Name Stand. Abbrev., in press.

[3] ARM cortex M3 core- LPC1768 CortexM3 Microcontroller NXP Semiconductors. Available at- LPC1768 CortexM3 Microcontroller NXP Semiconductors.

[4] Chin E. Lin, Hung-Ming Yen, “A Prototype Dual Can-Bus Avionics System For Small Aircraft Transportation System” 2006 IEEE.

[5] Chris Quigley, Richard McLaughlin, “Electronic System Inte-Gration For Hybrid And Electric Vehicles”.