Implementation Of 'CAN' Protocol In Automobiles Using Advance Embedded System

Vikash Kumar Singh[#1], Kumari Archana[*2]

[#1 & *2] *Student, M.Tech ES & VLSI-D, Dept. Of ECE, LIET, JNTU-Hyderabad*

Lords Institute of Engineering and Technology, Himayat Sagar, Hyderabad, INDIA

***Abstract*— This paper describes the benefit of control systems with network architecture over traditional systems with a central processor. A suitable standard protocol, CAN, is briefly presented and its current and future use in automobile machines is discussed. An important task is to find a way to make it possible to use standard network modules from different producers in a network specially designed for a specific machine. A solution to this problem is the design rules "CAN Kingdom" and the basics for this are presented. This model consists of the collection and display of data which are independent of each other and remotely executed. The data collection module receives data through CAN bus, and the data display module display these data via GUI designed by Qt/Embedded in Microcontroller. Temperature sensor and IR sensors are used as data collection agents and LCD and motor is used as output agents for an application here to demonstrate the effectiveness of CAN.**

To avoid data corruption, redundancy and reduce the complexity of circuit, MCP2510 is used for extending CAN. The whole CAN bus system is made up of the MCP2510 which is a stand-alone CAN controller with SPI Interface and the MCP2551 which acts as an interface between the CAN controller and the physical bus which carries data. MCP2510 is capable of both acceptance filtering and message management. It includes three transmit buffers and two receive buffers that reduce the amount of microcontroller (MCU) management required. The MCU communication is implemented via an industry standard Serial Peripheral Interface (SPI) with data rates up to 5 Mb/s.

***Keywords*— CAN, Controller Area Network, MCP2551, MCP2510, CAN in Automobile Industry, CAN Kingdom, CAN Architecture**

I. INTRODUCTION

CAN stands for Controller Area Network. It is a protocol which defines a set of rules of data transfer from on point to another point. CAN protocol was developed for making sure data from one node gets transferred to another node between two connection safely and securely without any data corruption and without missing any of the data. CAN protocol was mainly intended for short length data transfer like in automobiles. CAN Protocol is used in automation of factory machinery for example two machines which interacts between each other and transfers data to take some mutual decision and then act accordingly. It is also used in medical automation, marine ships, military applications and at all other places where a simple yet robust data transfer network is needed. CAN is not a complete network. It consists of only the physical layer, the priority scheme and the error checking system. We can say that CAN is just a connection between two nodes of a network.

CAN can broadcast multiple messages at a single point of time. USB and Ethernet protocols uses a block of data to transfer data form point A or node A to pint 2 or node B which makes the speed of the transfers less where as CAN does not uses a block of data and achieves maximum signalling rate In a CAN network, many short messages like temperature or RPM can be broadcast to the entire network, which provides data consistency and accuracy in every node of the system.

The paper mainly describes an application of CAN in automobiles. We have used CAN as a device which can enhance the utility, performance, speed & security of a system. Just to explain how a CAN can be utilized in an automobile we have used two CAN nodes connected by 2 Mbps CAN bus. Temperature & voltage sensor at one node called as NODE A is connected to CAN through Microcontroller A. The other node called NODE B is connected to Microcontroller B through CAN controller. In Microcontroller B an IR sensor & machine control is attached to exchange the automatic of automobiles. When the sensor at NODE A senses the change in temperature it captures the information and passes it to CAN connected at NODE A. CAN transfers this data to NODE B of CAN and then the NODE B microcontroller displays it on LCD. Similarly the IR sensor is used to sense the light and increase or decrease the speed of motor based on intensity of light.

II. CONTROLLER AREA NETWORK(CAN)

CAN is a serial communication bus protocol defined in International Standards Organization (ISO). A serial communication bus is basically used to transfer data from one network or point to another point in duplex mode. It was developed for the automotive industry as an alternate to the complex wiring with a two-wire bus. CAN greatly reduce the electrical interference and noise interference is the transferred signal in the network. CAN introduce an error detection and correction mechanism in the network which is very effective In transferring the correct and authenticate data through the network. CAN is very popular in industries including building automation, medical, and manufacturing. The CAN communications protocol, ISO-11898: 2003 explains how information is passed between devices on a network CAN follows OSI model that is defined in terms of different layers. CAN uses only Physical and data link layer of OSI model as shown below. Physical layer of the CAN and OSI model explains the actual communication between devices connected. Two nodes are connected to each other via a physical wired connection. The ISO 11898 architecture defines the lowest two layers of the seven layer OSI/ISO model as the data-link layer and physical layer in Fig 2.1 and fig 2.2 below. Fig 2.1 shows which all layers of OSI is used in CAN and which all are not used. Fig 2.2 shows in details the usage of DLL and Physical layer in CAN.

Application Layer – Not Used in CAN
Presentation Layer – Not Used in CAN
Session Layer – Not Used in CAN
Transport Layer – Not Used in CAN
Network Layer – Not Used in CAN
Data Link Layer : LLC and MAC
Physical Layer : PLC, PMA and MDI

Fig. 1.1 ISO/OSI Reference Model

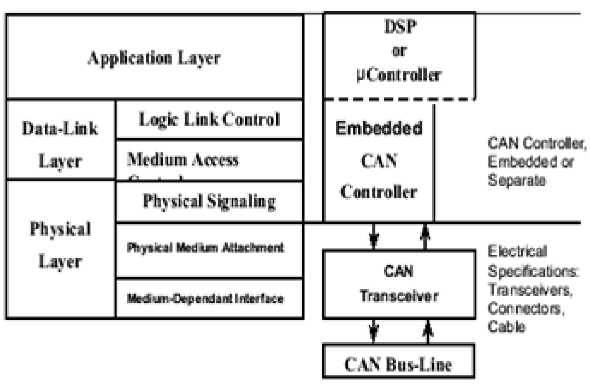

Fig. 2.2 Standard layered ISO 11898 Architecture

A. The Bit Fields of Standard CAN and Extended CAN:

Standard CAN

Fig. 3.3 Standard CAN 11 bit identifier

The abbreviation of the fields of Figure 2.3 is:

- SOF–It is single dominant start of frame (SOF) bit Message in CAN starts with SOF.
- Identifier-The Standard CAN 11-bit identifier. It establishes the priority of the message.
- RTR–Remote Transmission Request (RTR)
- IDE–A dominant single identifier extension (IDE) bit. If this bit is enabled means that a standard CAN identifier with no extension is being transmitted.
- r0–Reserved bit
- DLC–The 4-bit data length code (DLC). It contains the number of bytes of data which needs to be transmitted.
- Data–Up to 64 bits of application data may be transmitted.
- CRC–The 16-bit (15 bits plus delimiter) cyclic redundancy checks (CRC). It contains the checksum of message
- ACK = Acknowledge bit
- EOF– End-of-frame (EOF) Bit ,
- IFS–This 7-bit interframe space (IFS)

Extended CAN

Fig. 4.4 Extended CAN 29 bit identifier

As shown in Figure 2.4, the Extended CAN message is the same as the Standard message with the addition of:

- SRR–Substitute Remote Request (SRR) bit. It replaces the RTR bit in the standard message
- IDE–A recessive bit in the identifier extension (IDE). It indicates that more identifier bits follow. The 18-bit extension follows IDE.
- r1–Following the RTR and r0 bits, an additional reserve bit has been included ahead of the DLC bit.

B. Principles of data exchange in CAN

When data are transmitted by CAN, no stations are addressed, but instead, the content of the message (e.g. rpm or engine temperature) is designated by an identifier that is unique throughout the network. This is shown in fig 2.5. The identifier defines not only the content but also the priority of the message. This is important for bus allocation when several stations are competing for bus access.

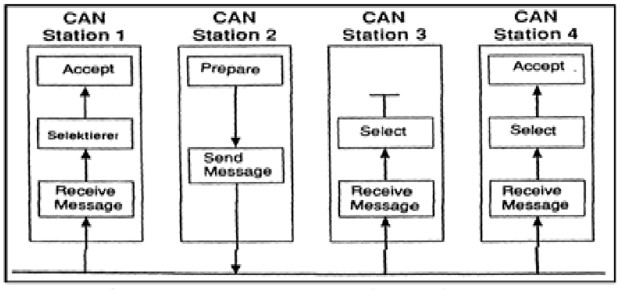

Fig. 5.5 Broadcast transmission and acceptance filtering by CAN

If the CPU of a given station wishes to send a message to one or more stations, it passes the data to be transmitted and their identifiers to the assigned CAN chip (”Make ready”). This is all the CPU has to do to initiate data exchange. The message is constructed and transmitted by the CAN chip. As soon as the CAN chip receives the bus allocation (”Send Message”) all other stations on the CAN network become receivers of this message (”Receive Message”). Each station in the CAN network, having received the message correctly, performs an acceptance test to determine whether the data received are relevant for that station (”Select”). If the data are of significance for the station concerned they are processed (”Accept”), other- wise they are ignored.

The below fig 2.6 shows the principal of non destructive bitwise arbitration.

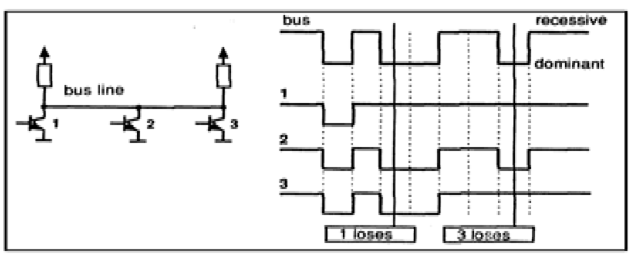

Fig. 6.6 Principal of non-destructive bitwise arbitration

A high degree of system flexibility is achieved as a result of the content-oriented addressing scheme which is used here. It is very easy to add stations to the existing CAN net- work without making any hardware or soft- ware modifications to the existing stations, provided that the new stations are purely receivers. Because the data transmission protocol does not require physical destination addresses for the individual components, it supports the concept of modular electronics and also permits multiple reception (broad- cast, multicast) and the synchronization of distributed processes: measurements needed as information by several controllers can be transmitted via the network, in such a way that it is unnecessary for each controller to have its own sensor

C. Physical CAN connection

The data rates (up to 1 Mbit/s) necessitate a sufficiently steep pulse slope, which can be implemented only by using power elements. A number of physical connections are basically possible. However, the users and manufacturers group CAN in Automation recommends the use of driver circuits in accordance with ISO 11898. Integrated driver chips in accordance with ISO 11898 are available from several companies (Bosch, Philips, Siliconix and Texas Instruments). The international users and manufacturers group (CiA) also specifies several mechanical connections (cable and connectors).

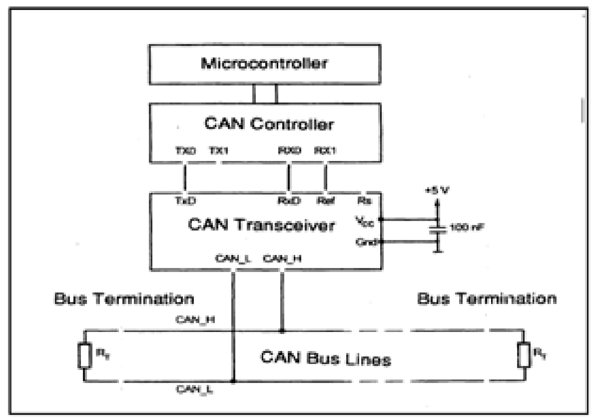

Fig. 7.7 Physical CAN connection according to ISO 11898

III. CAN TRANSCEIVER AND CONTROLLER

A. CAN Controller MCP2510

Microchip Technology’s MCP2510 is a stand-alone Controller Area Network (CAN) controller that implements the CAN specification, version 2.0B. It is capable of transmitting and receiving both standard and extended data and remote frames. The MCP2510 has two acceptance masks and six acceptance filters that are used to filter out unwanted messages, thereby reducing the host MCUs overhead. The MCP2510 interfaces with microcontrollers (MCUs) via an industry standard Serial Peripheral Interface (SPI).

B. CAN Transceiver(MCP2551)

The MCP2551 is a high-speed CAN, fault-tolerant device that serves as the interface between a CAN protocol controller and the physical bus. The MCP2551 provides differential transmit and receive capability for the CAN protocol controller and is fully compatible with the ISO-11898 standard, including 24V requirements. It will operate at speeds of up to 1 Mb/s. typically; each node in a CAN system must have a device to convert the digital signals generated by a CAN controller to signals suitable for transmission over the bus cabling (differential output). It also provides a buffer between the CAN controller and the high-voltage spikes that can be generated on the CAN bus by outside sources (EMI, ESD, electrical transients, etc.).

C. PIC16F877a

For this model, PIC microcontroller is used because of following advantages of PIC

1. Many interrupt sources, so need more interrupt pins
2. Need very high speed
3. Must be fully compatible with CAN
4. It should be having low-power consumption

IV. Model to Implement CAN with PIC

In the model to show the CAN with PIC we have used two CAN nodes connected by 2 Mbps bus. We have used temperature sensor at one node A connected to CAN through Microcontroller A. The other node B is connected to Microcontroller B through CAN controller. In MCB an IR sensor & machine control is attached to exchange the automatic of automobiles.

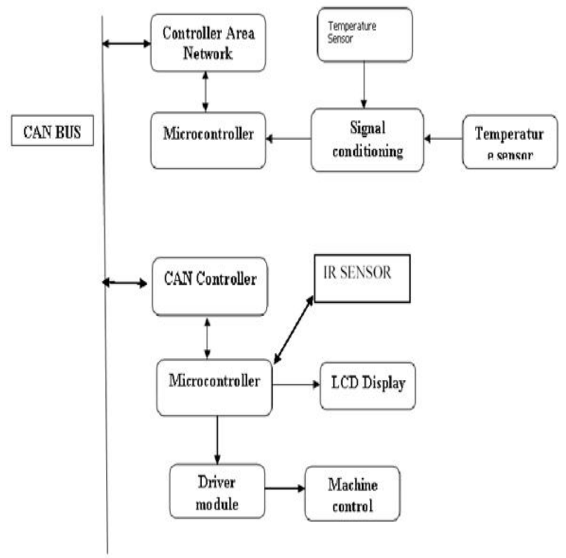

Fig. 4.1 Block Diagram of the Model

The temperature sensors will sense the temperature of different places and it will pass the data to microcontroller. The PIC microcontroller is connected to the CAN transceiver. The CAN will receive data from microcontroller and it will pass the data to node B of project where this information will be displayed on LCD.

At the node B we have used LCD, Infrared sensor, Motor. The LCD will display the data received from node A. IR sensor is used to detect the presence of any object. If the path of IR sensor is blocked then the speed of motor should be decreased.

The CAN controller consists of CAN transceiver and CAN controller. CAN transceiver transmits and receives the data at node from high speed CAN bus and CAN controller is used to interface the CAN transceiver with microcontroller.

The flow of block diagram depicts that the temperature from the temperature sensors will be sensed and will be passed to microcontroller. Then microcontroller will do the signal conditioning and ADC. From there the data will be passed to CAN controller and then to CAN transceiver. CAN transceiver will load the data onto bus so that it can be transferred to other node. At node 2 the reverse action will take place and data will be displayed on screen. At node B also an IR sensor and motor is interfaced to the microcontroller. If IR sensor detects any object then the speed of motor will be decreased.

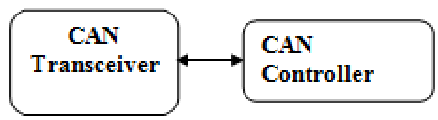

Fig 4.2 CAN controller Block diagram

The below figure shows the actual hardware connections and implemetation of the proptotype model of the descrives CAN on PIC microcontrioller.

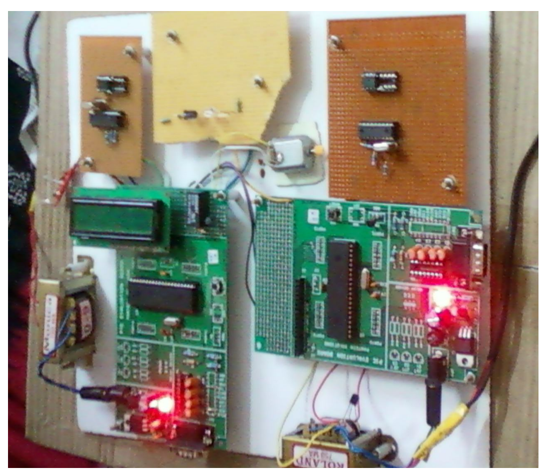

Fig 6.3 Snapshot of Implemented Model

V. Flow of Operation at Transceiver and Receiver Sections

A. *Transmitter Section*

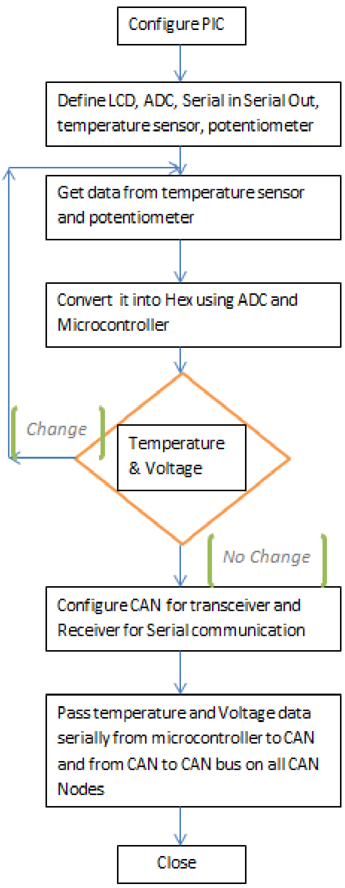

Fig 6.1 Flowchart of transmitter section

The above flow chart is converted in to C++ program and is embedded in the PIC microcontroller.

B. *Software's used :*

1. OrCAD
2. MPLAB IDE
3. Proteus 7.2
4. Flash programmer WIN PIC 800

C. *Major Hardware Components used :*

1. CAN Controller MCP2510
2. CAN Transceiver MCP2551
3. PIC 16F877a
4. Temperature Sensor LM35
5. Liquid Crystal Display
6. Infrared Sensor

D. *Receiver Section*

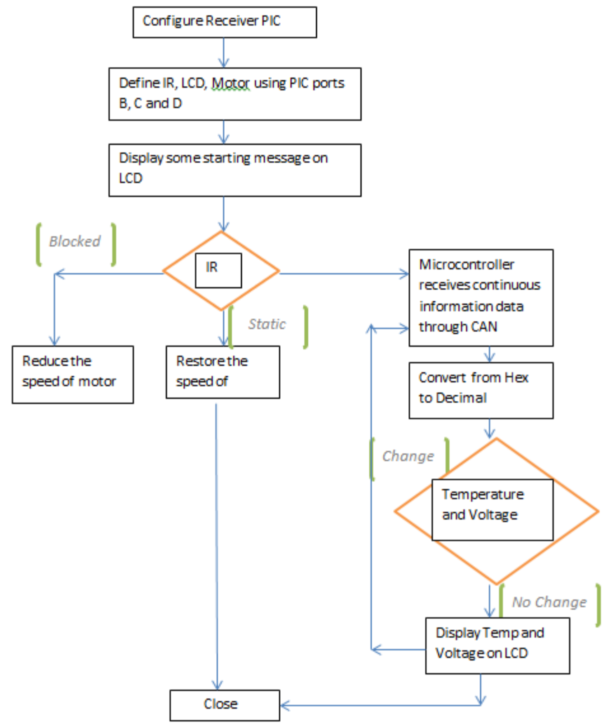

Fig 6.2 Flowchart of Receiver section

VI. Advantages and User Benefits

A. *Advantages:*

1. Intelligent and self-smart systems
2. Effective alert module
3. Data transfer using CAN Protocol
4. Driver-related signals for processing

B. *User Benefits:*

1. CAN can be implemented in a very low cost
2. CAN is reliable
3. CAN means real-time
4. CAN is flexible
5. CAN is fast
6. CAN allows Multi-Master Operation
7. CAN means Broadcast Capability
8. CAN is standardized

C. *Applications*

1. CAN in motor vehicles (cars, trucks, buses)

It Enables communication between ECUs like engine management system, anti-skid braking, gear control, active suspension (power train). It can be used to control units like dashboard, lighting, air conditioning, windows, central locking, airbag, seat belts etc. (body control)

2. CAN in utility vehicles

e.g. construction vehicles, forklifts, tractors etc. CAN can be used for power train and hydraulic control.

3. CAN in trains

High need of data exchange between the different electronic subsystem control units (i.e. acceleration, braking, door control, error messages etc. as well as diagnostics).

4. CAN in industrial automation

CAN provide excellent way of connecting all kinds of automation equipment (control units, sensors and actuators). It can be used for initialization, program and parameter up-/download, exchange of rated values / actual values, diagnosis etc. Machine control (printing machines, paper- and textile machines etc.): Connection of the different intelligent subsystems. It can be used in transport systems.

5. CAN in medical equipment

Example is Computer tomographs, X-ray machines, dentist & wheel chairs

6. CAN in building automation

Example is Heating, air conditioning, lighting, surveillance etc. Elevator and escalator control

7. CAN in household appliances

Example Dishwashers, washing machines, even coffee machines.

8. CAN in office automation

Example is photocopier - interface to document handler, paper feeding systems, sorter, communicates status, allows on-site / field connection or "hot swapping, DocuText Systems, i.e. automatic print, sort and bind on demand

VII. CONCLUSIONS

CAN is ideally suited in applications requiring a large number of short messages with high reliability in rugged operating environments. Because CAN is message based and not address based, it is especially well suited when data is needed by more than one location and system-wide data consistency is mandatory. Fault confinement is also a major benefit of CAN. Faulty nodes are automatically dropped from the bus, which prevents any single node from bringing a network down, and ensures that bandwidth is always available for critical message transmission. This error containment also allows nodes to be added to a bus while the system is in operation, otherwise known as hot-plugging. The many features of the TI CAN transceivers make them ideally suited for the many rugged applications to which the CAN protocol is being adapted. Among the applications finding solutions with CAN are automobiles, trucks, motorcycles, snowmobiles trains, buses, airplanes, agriculture, construction, mining, and marine vehicles. In this model CAN protocol has been implemented and its uses and advantages are shown by live examples. In the model temperature and voltages are sensed at one node and displayed at the other node of the CAN. Also, presence of object is sensed and appropriate action is taken. Overall this model explains the uses and necessity of implementing CAN protocol in modern Life.

ACKNOWLEDGMENT

The authors are thankful to S.V. Altaf, Professor, Lords Institute of Engineering and Technology, JNTU Hyderabad for providing the necessary facilities and guidance for executing the model and the preparation of the paper.

REFERENCES

[1] Controller Area Networks: Evolution and Applications, Othman, H. F.; Aji, Y. R.; Fakhreddin, F. T.; Al-Ali, A.R., "Controller Area Networks: Evolution and Applications," Information and Communication Technologies, 2006. ICTTA '06. 2nd , vol.2, no., pp.3088,3093, 0-0 0

[2] Automobile Control System using Controller Area Network. International Journal of Computer Applications 67(18):34-38, April 2013.

[3] Kiencke U, Kytola T. 1997. "CAN, a Ten Years' Anniversarial Review", Institute of Industrial Information Systems, University of Karlsruhe.

[4] L. Vhzquez , L. Rojas , E. Uribazo. 2000. "Intelligent CAN - ISA Gateway Development to Industrial Networks and Internet Coupling", IEEE.

[5] R. S. Raji, “Smart networks for control,” IEEE Spectrum, vol. 31, pp.49–55, June 1994.

[6] R. Davis, N. Navet, “Traffic Shaping to Reduce Jitter in Controller Area Network (CAN)“, ACM SIGBED Review, Special issue of the 24th Euromicro Conference on Real-Time Systems (ECRTS12), Volume 9, Issue 4, pp37-40, November 2012

[7] N. Navet, Y-Q. Song, F. Simonot, “Worst-Case Deadline Failure Probability in Real-Time Applications Distributed over CAN (Controller Area Network)“, Journal of Systems Architecture, Elsevier Science, vol. 46, n°7, 2000.