Controller Area Network for Automobile Application using ASIC Based on PSoC and Analysing through Vector CANoe

Divya Sharma, Mayank Gupta

Abstract— In the automotive industry, embedded control has grown from stand-alone systems to highly integrated and networked control systems. By networking electro-mechanical subsystems, it becomes possible to modularize functionalities and hardware, which facilitates reuse and adds capabilities. With the increasing number of distributed microcontrollers and intelligent peripherals used in today's electronic systems, such as vehicle controls, networking protocols between the units have become extremely important. A wide range of these applications are using CAN (Controller Area Network) for network communication. The CAN bus was developed by BOSCH as a multi-master, message broadcast system that specifies a maximum signaling rate of 1M bit per second (bps). Unlike a traditional network such as USB or Ethernet, CAN does not send large blocks of data point-to-point from node A to node B under the supervision of a central bus master. In a CAN network many short messages like temperature or RPM are broadcast to the entire network, which allows for data consistency in every node of the system [1].

Index Terms— Controller Area Network, Cypress PSoC, CANoe, CANalyzer.

I. INTRODUCTION

Some ten years ago the technology of micro-controllers had reached a level that made it possible to design machines constructed of micro processor controlled stand alone actuators. These actuators are nodes in a serial network, a Controller Area Network, and are coordinated by a master computer. Such a maschine architecture has several advantages, not only technical but also project managerial. In spite of this the development within this area has made little progress until recent time. The main reason for this has been the lack of a reliable standard protocol suitable for fast hard real-time communication. Such a protocol, designated CAN, was however developed for the automotive industry by Bosch and Intel and is now implemented in chips by the leading chip manufacturers.[1] The technology is continuing to play an increasing role in automobiles. The industry has recently seen the advent and development of many types of autonomous vehicles. Since the 1970s, one observes an exponential increase in the number of electronic systems that have gradually replaced those that are purely mechanical or hydraulic. The growing performance and reliability of hardware components and the possibilities brought by software technologies enabled implementing complex functions that improve the comfort of the vehicle's occupant as well as their safety[2].

In the early 1980s, engineers at Bosch were evaluating existing serial bus systems regarding their possible use in passenger cars. Because none of the available network protocols were able to fulfill the requirements of the automotive engineers, Uwe Kiencke started the development of a new serial bus system in 1983. The new bus protocol was mainly supposed to add new functionality – the reduction of wiring harnesses was just a by-product, but not the driving force behind the development of CAN. Engineers from Mercedes-Benz got involved early on in the specification phase of the new serial bus system, and so did Intel as the potential main semiconductor vendor. Professor Dr. Wolfhard Lawrenz from the University of Applied Science in Braunschweig-Wolfenbüttel, Germany, who had been hired as a consultant, gave the new network protocol the name 'Controller Area Network'. Professor Dr. Horst Wettstein from the University of Karlsruhe also provided academic assistance. In February of 1986, CAN was born: at the SAE congress in Detroit[3].

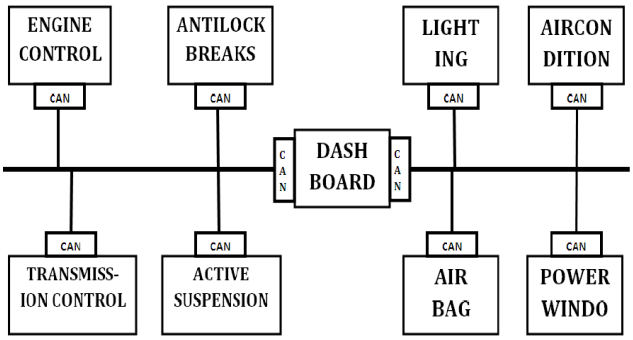

Figure 1. Automobile Subsystem With CAN

A. The need for serial communication in vehicles

Many vehicles already have a large number of electronic control systems. The growth of automotive electronics is the result partly of the customer's wish for better safety and greater comfort and partly of the government's requirements for improved emission control and reduced fuel consumption. Control devices that meet these requirements have been in use for some time in the area of engine timing, gearbox and carburetor throttle control and in anti-block systems (ABS) and acceleration skid control (ASC).

Manuscript received on April, 2013.
DivyaSharma , School of Electronics Engineering, Lovely Professional University, Phagwara, India.
Mayank Gupta, School of Electronics Engineering, Lovely Professional University, Phagwara, India.

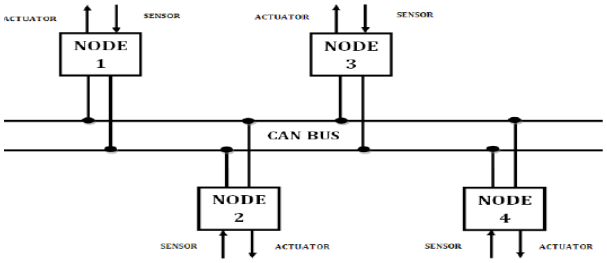

Figure 2. Distributed CAN Network

The complexity of the functions implemented in these systems necessitates an exchange of data between them. With conventional systems, data is exchanged by means of dedicated signal lines, but this is becoming increasingly difficult and expensive as control functions become ever more complex. In the case of complex control systems (such as Motronic) in particular, the number of connections cannot be increased much further. Moreover, a number of systems are being developed which implement functions covering more than one control device. For instance, ASC requires the interplay of engine timing and carburettor control in order to reduce torque when drive wheel slippage occurs. Another example of functions spanning more than one control unit is electronic gearbox control, where ease of gear changing can be improved by a brief adjustment to ignition timing If we also consider future developments aimed at overall vehicle optimization, it becomes necessary to overcome the limitations of conventional control device linkage. This can only be done by networking the system components using a serial data bus system. Bosch developed a system for this purpose, the "Controller Area Network" (CAN), which has since been standardized internationally (ISO 11898) and has been "cast in silicon" by several semiconductor manufacturers[4].Controller Area Network (CAN) is the latest communication system within the automotive world. At its simplest level, it can be thought of as a means of linking all of the electronic systems within a car together to allow them to communicate with each other. As computerisation within a car increases, so to does the number of different electronic systems. The information recorded and processed by each one is often used by one or more others - hence the requirement for a standardised means of quickly passing information between them. This requirement led to the development of CAN[5] The CAN protocol now forms part of the OBDII and EOBD standards. The CAN protocol describes the method by which information is passed between devices.

It conforms to the OSI model and defines the lowest two layers: the data link and the physical layer. The application layers are linked to the physical layer by various emerging protocols or a proprietary user defined scheme.

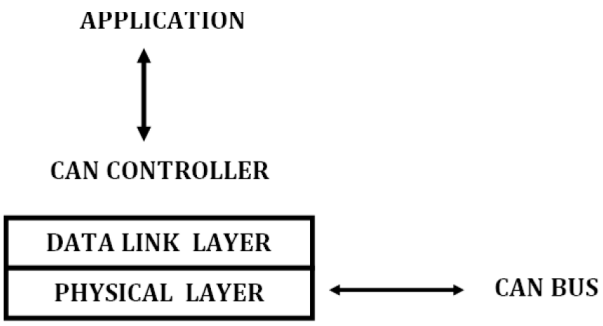

Figure 3. CAN Physical Layer

Two varieties of CAN exists today, basic CAN or a higher level Full CAN with acceptance filtering hardware. In the basic CAN implementation all messages broadcast on the network have to be individually checked by the host controller. This results in the CPU often being tied up checking messages rather than processing them. The introduction of an acceptance filter masks out irrelevant messages, using identifiers (ID´s) leaving the CPU to process only those messages that are of interest. The CAN protocol allows for two identifier field lengths: part A specifies 11 bits, which allows 2032 different ID´s out of 2048, whilst extended CAN (part B) has 29 bits giving over 536 million unique identifiers .Today a wide range of applications use CAN for distributed networks. CAN 2.0B is now used in the power train of many midrange and high-end vehicles for communication between engine management, ABS, transmission control, ASR, active suspension and electronic throttle. This high speed bus with up to 500 baud transmission speed exchanges information such as motor RPM, vehicle speed and throttle control. In case of braking for instance this information can be used to automatically switch off the fuel injection to reduce the stopping distance. In a new car radio application for Mercedes Benz the volume control can automatically adjust by up to 10dB based on the speed information broadcast on the CAN bus or switch off. Those components like car radio, airbag, multifunctional display for satellite navigation, mobile phones or intelligent sensors are connected to a second CAN bus which uses 40 - 125kBaud transmission speed. The dashboard is the control panel which displays all relevant information and its instrumentation controller acts in many cases as the bridge between high- and low-speed buses. This intensive use of CAN in the automotive industry together with rapidly decreasing pricing for CAN components attracted the attention of industrial users. Decisions in favour of CAN have being made not only by manufacturers of agricultural machinery and marine engineering equipment but also by manufacturers of medical equipment, textile machines and elevator monitoring systems[6].

There is today more than 20 years of experience in automotive CAN applications, and CAN has certainly proven very successful as a robust, cost effective and all-around network technology

C. CAN protocol advantages over other serial communication protocols.

1) CAN is a message based protocol. In CAN, nodes are not assigned specific addresses in CAN networks. This provides flexibility to add or remove a node to/from the network without affecting the rest of the network.

2) CAN protocol ensure that even if one of the nodes fails, others continue to work and communicate properly.

3) Messages are prioritized so that critical messages are delivered first.

4) CAN network offers system wide data consistency In CAN networks, corrupted messages are automatically retransmitted as soon as the bus is idle again.

5) CAN incorporate a five level error checking capability to ensure reliable traffic and data integrity [6].

D. Benefits of CAN-based diagnostics

The CAN protocols and interface run at roughly fifty times the speed of the earlier OBDII/EOBD protocols. This speed improvement, combined with the addition of new diagnostic parameters, will give technicians the ability to see data faster and gain more information from newer cars than is currently available.[2] This paper include communication between two nodes using ASIC based on Cypress PSoC and then implementing CAN control in Vector CANoe software. The design has been implemented on CANoe/CANalyzer software which include database, panel designer and the graphical results obtained during tested.

II. METHODLOGY

This paper, introduces the basic concepts of CAN protocol and demonstrates how CAN bus communication can be implemented using PSoC 3 and PSoC 5LP (hereafter referred to as PSoC) and software implementation of control system in automobiles through Vector CANoe /CANalyzer software for three nodes.

A. Use of PSoC

PSoC integrates the CAN functionality along with configurable analog, programmable digital, memory, and a central processor on a single chip. Built-in functions called APIs (Application Programming Interfaces) abstracts the commonly used tasks and allows user to focus more on to the core of the problem. Moreover, all components in the PSoC including CAN has a set of options which can be easily configured using GUI, rather than writing C codes for them. CAN component in PSoC is compliant with the CAN 2.0a and 2.0b standards. However, it requires an external transceiver to level shift the output voltages and to make it compatible with the CAN protocol. TJA1050 can be used as an external transceiver[7].

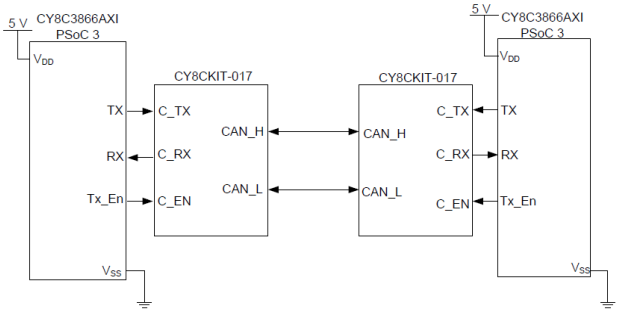

Figure 3. Hardware Connections for Implementing CAN

B. Configuring in PSoC

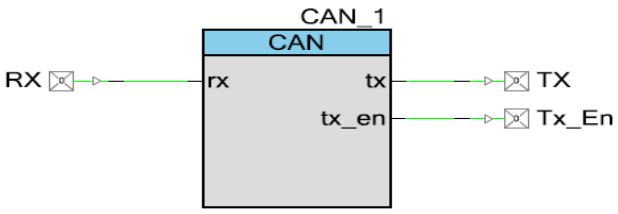

Figure 4. CAN Component in PSoC Creator

PSoC offers two different modes for communication .Full CAN and Basic CAN. A full CAN communication can be easily set up with the help of GUI with very limited amount of programming, whereas Basic CAN requires all the parameters to be set in firmware. it uses hardware for message filtering. Basic CAN requires that the CPU is interrupted every time a message is received to determine whether it is accepted or not. Full CAN only be used for receiving a single type of message per mailbox whereas Basic CAN configuration can accept message with a range of identifiers per mailbox[7].

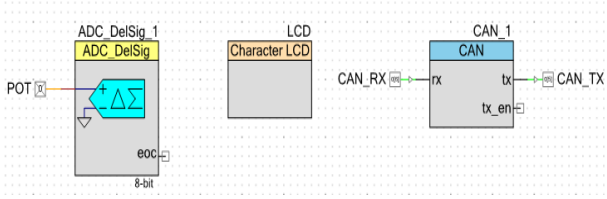

Figure 5. CAN design in PSoC Creator (Transmitter side)

C. Implementation of hardware

The PSoC outputs from the CAN component are TX and RX. These outputs need to be level translated in order to obtain the CANH and CANL signals. TJA1050 is the CAN transceiver IC used CAN requires only two wires a CAN bus. A standard male- to –male DB9 connector may be used to connect two CAN nodes. Three components are used in the hardware design Delta Sigma ADC, LCD, and CAN controller[7].

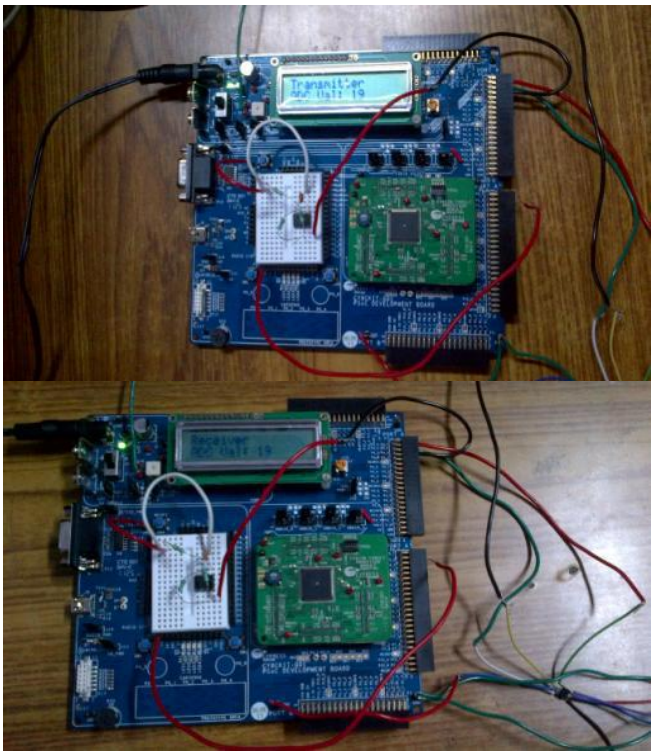

Figure 6. Two node CAN Communication

An Automobile System receives for example the ignition input from the user and vehicle inputs through a CAN transceiver and drives the three-phase brushless automotive motor. The following figure provides the ADC acquiring the vehicle input signal, taken using the DAQ assistant, monitoring the changing inputs applied to ADC on PSoC device.

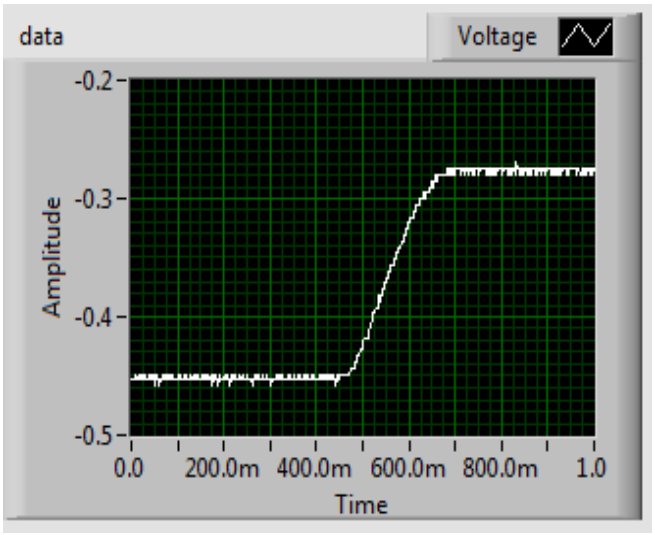

Figure7. Amplitude Versus Time Curve For Analog Vehicle Inputs

D. Working with Vector CANoe/CANalyzer

The control system for automobile using three CAN node is designed on a software tool. CANoe/CANalyzer is the software tool which used for the system design.

The system design using CANoe/CANalyzer involves six steps.

- Create a database.
- Create node.
- Associate the database.
- Panel design.
- Place the panel into the CANoe environment.
- Create node behaviour

Database of the system defines the overall functionality of the system among different network nodes which include the input as well as output variables of the system. It include defining messages and selecting baud rate of the bus of the bus. Simulation tool evaluate this information first to provide initial estimates of bus load and the latency times to be expected at the prescribed baud rate. Afterwards, this specification can also be utilized for testing in subsequent phases. The figure given below shows the database of the design[8].

1) Database

Database of the system defines the overall functionality of the system among different network nodes which include the input as well as output variables of the system. It include defining messages and configuring the baud rate of the bus. Database of the system define two important aspects of the system:

- Exchange of the information or message frame between two network nodes via communication bus i.e. CAN bus
- Input / output interface of the peripheral or we can saywiring between each node and its I/O units.

Each peripheral element is "wired" to an environment variable or system variable i.e. it is connected to the CAPL program for the network node. Simulation tool evaluate this information first to provide initial estimates of bus load and the latency times to be expected at the prescribed baud rate. Afterwards, this specification can also be utilized for testing [7].

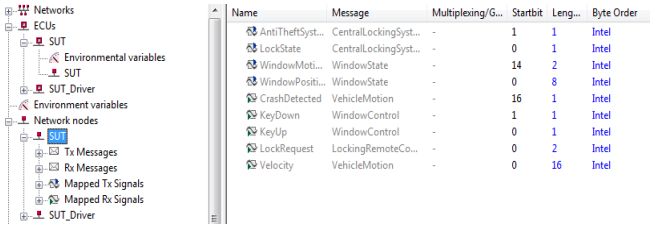

Figure8. Database Of The Design

After database creation the design and development of individual network node is done independently and in parallel. The tool require additional hardware for real bus simulation but the design here is tested in an offline mode without hardware.[8]

2) Node

Network node is created in the simulation setup. In this system design three node has been created. After creating the network node, these node are configured. Each node has its own editor, compiler and panel so each node can be configured independently. This adds to the advantage in the system design because if a node new node is added to the system then the system remains unaffected [8].

The figure below shows the CAN network node of the design.

Figure9. CAN node of the design.

Each of these node has its own editor, compiler and node panel, so a node can either be added or can be removed from the bus without effecting the overall system which is the major advantage of CAN. The database is now associated with the node of the design. The control panel provides a user friendly interface to the environment or the system variables. The panel is created using panel designer. During simulation values of environment or system variables displayed and modified through control panel. The panel of the system design is shown in the figure below[8].

3) Associate the database

The database created at the initial stage of the system design is now associated with the network node so that the network node can interact with the periphery and respond to the input of periphery and perform the desired operation in real time[8].

4) Panel design

The software provides an option of creating a user friendly interface to the environment or the system variables and integrating them to the program. Here external events of the system design are described with the help of symbols, whose name and type are defined in the database.

The panel is created using panel designer [5]. During simulation values of environment or system variables are displayed and modified through panel. Values of the symbols can be configured whenever needed. The change in the value can be can be visualized on the panel using display element. Figure 7 shows the panel created for the system design of the control system of the automobile [8].

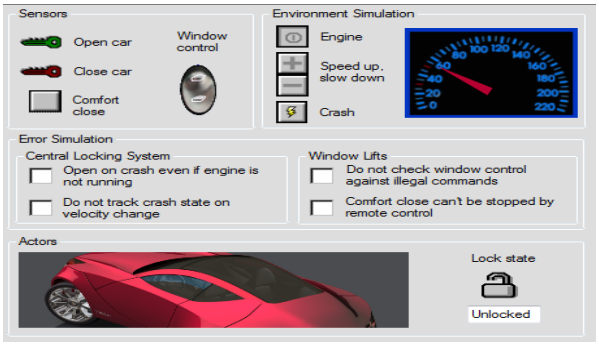

Figure 10. Display and Control Panel

As show in the panel the car is in running mode where its speedometer shows the running status displaying number of RPM and the car is open and window motion is taking place. state. The above panel shows the control panel for the control system designed for automobile. This panel comprises of engine ignition control, speed control window control and crash detected of the system. After panel is designed, the panel configuration is then added to the node. This panel act as an interface between peripherals and the system. Now these panel is added with network node and act as interface for the system variable or environment variable[8].

5) Node Behaviour

The node behaviour is created through programming. We use CAPL (Communication Application Programming Language) programming language feature of CANoe to give behaviour to our node. Now since each now has its own compiler and editor so each of the node is programmed individually on its own editor such if any of the node is removed the system remain unaffected and work efficiently. CAPL programs enables user to interface with wide variety of input, outputs and other functions. Start-stop events, keyboard entry events, the ability to transmit and receive CAN messages, interaction with the serial port and parallel port, the use of timers, and the ability to interconnect to customer specific DLLs are some of the interface choices available inside the CAPL programming environment [8].

II. SIMULATION RESULT

The above section describes the control system design using CAN standard. The system is then successfully simulated and this section describes the result obtained. The result is obtained in terms of graphs, trace and data analysis.

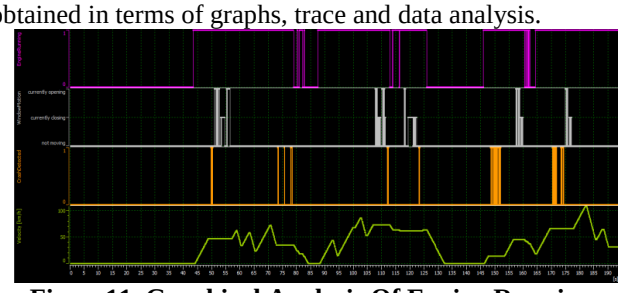

Figure11. Graphical Analysis Of Engine Running, Window Motion, Crash Detected And Velocity.

The graph of engine speed represents variation of engine speed and velocity in terms of (km/hr) with respect to time. At the start of the ignition the engine speed is at zero state but with the gradual variation in the speed we can see the proportional variation in the plot with respect to time. At the maximum speed i.e. at 220 km/hr the plot attains its maximum and when brake is applied we can see the gradual decrease in the curve, further there is no crash detected with velocity change. The graph of window control(lifts) indicate the state of the window position either UP or DOWN when window control is operated.

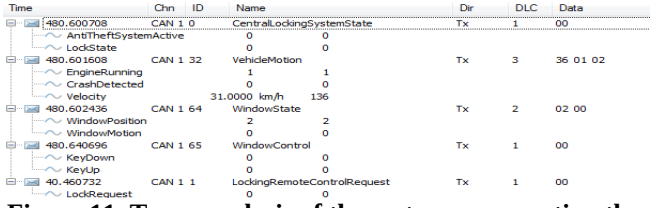

Figure 11. Trace analysis of the system representing the frame and data that are transmitted by the node.

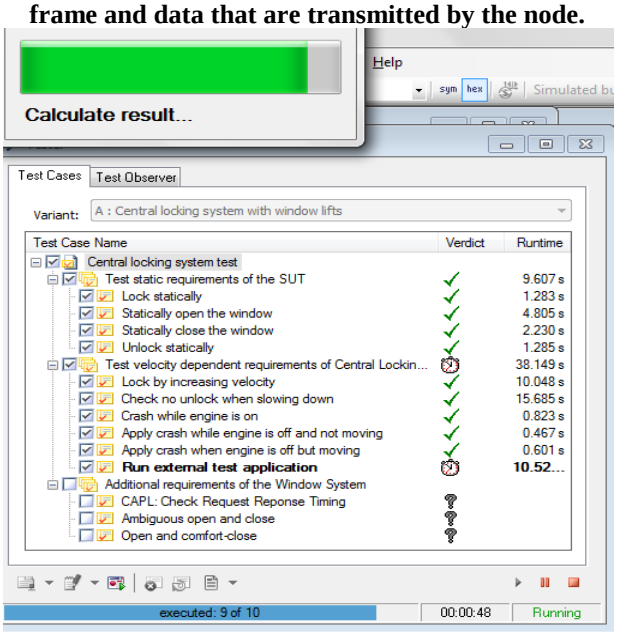

Figure 12. Test design analysis

III. CONCLUSION

CAN bus has advantage over the complex wiring system since it is more reliable network with efficient error handling mechanism, automatic recovery, cost reduction and provide an efficient data interchange mechanism among different nodes. 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 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. CAN standard has been adopted by many of the automotive industries such as Mercedes, BMW, Volkswagen, etc. CAN bus is still under development stage and with the continuous development of CAN bus, it will be more widely applied in the field of the automotive industries.

ACKNOWLEDGMENT

I like to express my sincere and humble thanks to my teacher "Mayank Gupta" for his kind cooperation and able guidance. I deeply express one indebtedness and thanks to all my faculty members of ECE for their invaluable guidance which enabled me to bring out this paper in a presentable manner.

REFERENCES

1. Lars-Berno Fredriksson, Controller Area Networks And The Protocol CAN For Machine Control Systems.
2. Matthew John, University of Kentucky, Development And Evaluation Of A Controller Area Network Based Autonomous Vehicle.
3. Steve Corrigan, Introduction to the Controller Area Network, Application Report SLOA101 - August 2002.
4. Controller-Area-Network- CAN. Available:http://www.esd-electronics-usa.com
5. What is CAN (controllerAreaNetwork). Available:http://www.gendan.co.uk/article_9.html
6. Embedded_tutorials/can_tutorial.htm
7. Ranjhit M , PSoC 3 and PSoC 5LP –getting started with CAN, AN52701, Software Version :PSoC Creator 2.1 SPI.
8. CANoe 75 manual "Vector Informatik GmbH"

AUTHORS PROFILE

Divya Sharma Educational Detail: University Name: Lovely Professional University, Phagwara, Punjab **Qualification:** Mtech (pursuing), last semester. Graduated in Electronics and Communication Engineering from Lala Lajpat Rai Institure Of Engineering and Technology, Moga (Punjab),2010. **Research Work**: CAN for Automobiles Application.

Achievements: Paper published in Bhartiya Vigyan Sammelan and Expo, (Conference) 2012. **Area Of Interest**: Automotive Embedded Systems.

Mayank Gupta Educational Detail: M.Tech in Nanotechnology from VIT University, Vellore (T.N.) B.E. in Electronics & Communication from RGTU, Bhopal (M.P.) **Research Work:** Chemical based sensor devices, Material Science, Embedded systems and Nano electronics. **Achievements :** Highly selective low power ammonia sensor at room temperature, in Sensor Actuator B: Chemical Journal. **Area Of Interest**: Nano Electronics, Material Science.