Journal of Management Engineering and Information Technology (JMEIT)
Volume -4, Issue- 3, Jun. 2017, ISSN: 2394 - 8124
Impact Factor : 4.564 (2015)
Website: www.jmeit.com | E-mail: editorjmeit@outlook.com|jmeit@outlook.com

Automated Driver-Vehicle Control System Using CAN Protocol

Mrs. Seema B Hegde, Mohan Kumar N K, Mayank Kumar, Menka Kumari, Rahul Kumar Jha

DOI 10.5281/zenodo.821238

***Mrs. Seema B Hegde**, Assistant Professor, Department of E&CE, SIT, Tumakuru-03, India, Phn: 9916308755*
seemab_hegde@yahoo.co.in
***Mohan Kumar N K**, Department of E&CE, SIT, Tumakuru-03, India, Phn: 9164240838,*
mohanmk0143@gmail.com
***Mayank Kumar**, Department of E&CE, SIT, Tumakur-03u, India, Phn: 8792166392,*
mayankkumarsit@gmail.com
***Menka Kumari**, Department of E&CE, SIT, Tumakuru-03, India, phn: 7411397212,*
menkaku@gmail.com
***Rahul Kumar Jha,** Department of E&CE, SIT, Tumakuru-03, India, phn: 8884502124,*
rahulkumar.1si13ec078@gmail.com

Abstract: Present Automobiles are being developed with more electronics for efficient operation of vehicles. Earlier vehicles were built with an analog driver-vehicle interface for indicating various vehicle status like speed, fuel level, engine temperature etc. The driver-vehicle interface provides an interaction of driver with the dashboard for monitoring various parameters of the vehicle and it is designed in a way to help the driver to drive safely and comfortably. There are two types of driver-vehicle interface i.e, analog and digital. In the existing analog driver-vehicle interface, it is tough to build an automated driving system because most of the processing for decision-making regarding vehicle control uses digital electronic devices (like microprocessors etc) and requires data to be handled in digital format. Hence there is need to develop digital driving system that uses digital data throughout the processing there by making the decision process more transparent as well as easy for design. The automated driving system built using digital interface allows the driver to switch from manual mode to cruise mode (semi-autonomous), so that the vehicle adjust itself with respect to other vehicles in motion, thereby helps in safe driving. The paper develops and implements the digital driving system for a semi-autonomous vehicle to improve the driver-vehicle interface. It uses an ARM based data acquisition system that uses ADC to bring all control data from analog to digital format and visualize through LCD. This paper is based on CAN which uses embedded networking for efficient data transfer. It also takes feedback of vehicle conditions like Vehicle speed, Engine temperature etc, and are controlled by main controller. Additionally this unit equipped with GSM which communicates to the owner during emergency situations.

Keywords— ECU (Engine Control Unit), CAN (Controller Area Network), Embedded C, GSM (Global System for Mobile (communication)).

I. INTRODUCTION

Road accident account for a sever threat to humans lives from both an injury as well as financial perspective. To overcome this issue, there is a need of vehicle automation which can provide an efficient and safe driver-vehicle interface. With rapidly changing computer and information technology and much of the technology finding way into vehicles, they are undergoing dramatic changes in their capabilities and the interaction with the drivers. Although some vehicles have provisions for deciding to either generate warnings for the human driver or controlling the vehicle autonomously, they must make these decisions in real time with only incomplete information. So, it is important that human drivers have some control over the vehicle. There are two types of driver-vehicle interface i.e analog and digital. In the existing analog driver-vehicle interface, it is tough to build an automated driving system because most of the processing for decision-making regarding vehicle control uses digital electronic devices and requires data to be handled in digital format [1]. Hence there is a need to develop digital driving system that uses digital data throughout the processing there by making the decision process more transparent as well as easy for design. Its introduction

into the vehicle design has allowed an almost symbiotic relationship between the driver and vehicle by providing a sophisticated and intelligent driver-vehicle interface through an intelligent information network.

This paper implements the digital driver-vehicle interface using CAN protocol. The controller area network (CAN) is a serial communications protocol which efficiently supports distributed real time control with a very high level of security. Its domain of application ranges from high speed networks to low cost multiplex wiring. In automotive electronics, engine control units, sensors, anti-skid-systems, etc are connected using CAN with bit rates up to 1Mbit/sec [3]. At the same time it is cost effective to build into vehicle body electronics, e.g: lamp clusters, electric windows etc. to replace the wiring harness otherwise required. This paper discusses the development of such a control framework for the vehicle which is called the *digital-driving behavior;* It consists of a joint mechanism between the driver and vehicle for perception, decision making and control.

II. REVIEW OF LITERATURE

Advanced in vehicle information systems provide vehicles with different types and levels of intelligence to assist the driver. The design and development of CAN protocol is widely used in automobile designing concepts. Some of the surveyed references are given below.

1. Navet.N. Loria.,Nancy,France,"Controller area network [automotive applications]" -2002

The controller area network (CAN) is becoming a standard for data transmission in automotive applications. Such applications are characterized by their obligation to respect stringent time and dependability constraints. This is why it is vital, at the design step, to validate their behaviour. The verification is based on the performance evaluation of the operational architecture. The authors' approach involves: (1) building models (both analytic and simulation) and solving them (model-based evaluation); (2) performing network monitoring on a prototype of the system (prototype-based evaluation). Automotive communications systems face an extremely hostile electrical environment (strong electromagnetic disturbances). To assume it is a 100% reliable medium is irresponsible in the face of such reality. So, the authors focus on transmission errors that may cause missed deadlines because of retransmitting corrupted frames. They propose using a flexible error-model for assessing the transmission's reliability and for designing well-suited message scheduling policies.

2. Bayilmis,C.,Kelebekler,E,"Remote control of a CAN-based mobile model car using a voice activated control system"-2008

This paper presents implementation of a remote controlled CAN-based mobile model car using a speech activated control system. The prototype consists of a voice activated control system, a CAN network employed in a model car and a CAN/IEEE 802.11b wireless interworking unit. This integrated approach proposes an effective solution for use in widespread industrial control and automation systems.

3. Xiaoming Li; Mingxiong Li,"An Embedded CAN-BUS Communication Module for Measurement and Control System"-2010

A CAN-BUS communication module for embedded system has been proposed in this paper, together with its hardware and software design. The motivation to design such CAN module is try to ease the development of CAN-BUS based embedded applications. With the CAN module, the engineers do not have to understand all aspects of CAN protocol, CAN controllers and CAN transceivers, therefore an industry automation application can be built conveniently. The CAN module provides simple hardware interface and highly abstracted software communication protocol, due to the abstraction and integration of all CAN related communication functions in an embedded module, and therefore can be easily embedded into any MCU based systems. The CAN module can send/receive data following the pipe based communication model. With the help of the software package, the CAN-BUS communication would be transparent to the micro-controllers of the nodes in the system.

III. OBJECTIVES OF THE PROPOSED SYSTEM

1. Implementation of Cruise control system in which the vehicle responds accordingly with respect to the other vehicles in motion. Ultrasonic sound sensor is used to measure the distance between vehicles for cruise control option.
2. Implementation of fuel level indication of the vehicle is achieved by linear motion of potentiometer and fuel float sensor by increasing or decreasing the fuel level in the tank.
3. Implementation of speed of the vehicle using GPS module in which the speed is measured with respect to the satellites that provides accurate value compared to conventional method, and also the location of vehicle can be tracked easily.
4. Implementation of Parking assistance facility for safe parking by implementing IR sensor in rare bumper.
5. Implementation of GSM module for emergency alert service.

IV. IMPLEMENTATION OF AUTOMATED DRIVER VEHICLE INTERFACE SYSTEM

This paper implements the digital driver-vehicle interface using CAN protocol. It implements the driver-vehicle interface for indicating various parameters of vehicle status like temperature, pressure and speed etc. To improve the driver-vehicle interface, an interactive digital system is designed. A microcontroller based data acquisition system that uses ADC to bring all control data from analog to digital format is used [6]. Since the vehicle information systems are spread out all over the body of a practical vehicle, a communication module that supports to implement a one stop control of the vehicle through the master controller of the digital driving system.

SYSTEM OVERVIEW

The main functional blocks of the system as shown in figure 3.1. Here slave and master unit are ARM 7 LPC2129 which includes inbuilt CAN protocol support.

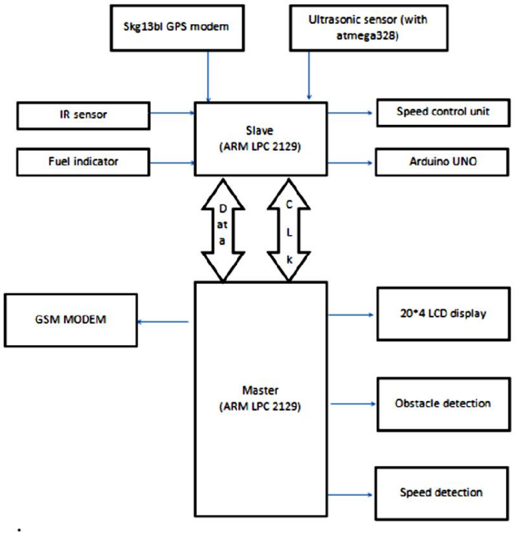

Figure 3.1: Block diagram of the system

The master unit is the main controller that takes the feedback by the slave unit via sub slaves, which is achieved through CAN communication. Master unit consists of *obstacle detection* and *speed detection* unit through which the distance from the obstacle and speed of the vehicle is observed on the LCD. Slave unit is having connection with all major parts in vehicle for identifying front object, to measure the distance between vehicle and also for cruise control option using ultrasonic sensor. The sensor data is acquired by slave device and controlling/indication parameter is send to the master device. IR sensor is used in rare bumper for *parking assistance* and logic data is sent to master. The fuel indication is achieved by linear motion of potentiometer with increase in the fuel level. To calculate the speed of the vehicle *GPS modem* is used so that accessing data is accurate compare to conventional method and also location of vehicle can be tracked easily and *GSM alert* service also can be include for emergency service. The connection of the master and slave units are shown in figure 3.2.

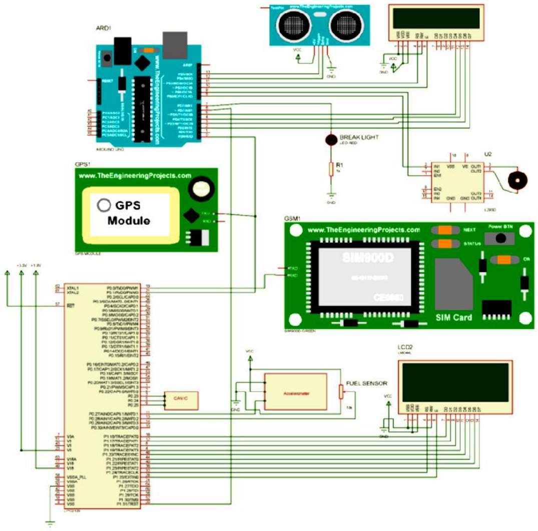

Figure 3.2: Circuit Diagram of the system

SOFTWARE

Programming the ARM7TDMI-S:

Traditionally, microcontrollers have been programmed in assembly language to make maximum use of the limited CPU power. Modern development tools can produce efficient code from compiled languages such as C. KEIL[2] is the Integrated Development Environment (IDE) used in the project for developing the hardware code. Practically, the microcontroller is controlled through direct memory read and writes operations to control and status registers at fixed addresses. While software libraries are available for most peripheral interfaces, in KEIL one can access the peripherals closer to the register level.

KeiluVision4:

Micro vision[2], the popular IDE from keil software, combines project management, source code editing, program debugging and ash programming in a single, powerful environment. The uVision development platform is easy-to-use and helping you quickly create
embedded programs that work. The uVisioned debug are integrated in a single application that provide a seamless embedded project development environment. Keil provides a broad range of development tools like ANSIC compiler, macro assemblers, debuggers and simulators, linkers, IDE, library managers, real-time operating systems and
evaluation boards for Intel 8051, Intel MCS-251, ARM, and XC16x[1]/C16x[2]/ST10 families. It provides an overview of the most commonly use uVision features including:

(a) Project management, device set up and tool configuration.
(b) Reader facilities for creating, modifying and correcting programs.
(c)Target debugging or CPU and Peripheral simulations.
The way to operate an ARM chip using Keil uVision software are as follows,
(a) Launch KeiluVision4.
(b) To Configure default values. Click Project and Components, Environment, Books and then select default value for Compiler as shown below.
(c) Create new Project File by using command Project and New Project and then configure or create position Folder to save new Project File with preferred Project Filename.
(d) Select LPC2129.
(e) Configure option value of Project File by using command Project option for Target, select Target 1 and then select Tab of Target to configure it.
(f) Start writing C Language Source Code, click command File and New.
(g) Add Files into Project File, click command Project and Components Environment, Books and select Tab Project Components and then select desired Add File to add into Project File.
(h) Command to interpret the written program, click command Projects and Re build all target files.

Flash Magic

Flash Magic is a software used to (a) Dump the .hex file to the ARM controller from PC using USB port of the computer through UART communication of the ARM controller. (b) Read the serial data obtained from the ARM controller on PC.

Dumping the code to ARM7

Following steps describes the way to dump an ARM chip via Flash Magic software:
(a) Launch Flash Magic.
(b) On the ARM controller board hold the In-System Programming (ISP) switch and reset switch then first release only reset and then release ISP.
I In Communications select LPC2129, select COM port and give a baud rate as 9600 and 12 Mhz as Oscillator frequency.
(d) Browse the hex file.
(e) Click on Start.
The hexfile is successfully dumped on to the LPC2148 ARM controller.

4. RTAW simulator

RtaW-Sim is a timing accurate simulator of Controller Area Networks (CAN) that provides frame response time distributions and statistics about the frame buffer usage at the microcontroller and communication controller level. RtaW-Sim is able to simulate and predict the performances of CAN 2.0A, CAN2.0B, ARINC825 and CAN FD networks, possibly interconnected through gateways, with a very accurate modeling of the communication stack and communication controller. RtaW-Sim helps the designer compare the impact of different design alternatives, choose the right communication stacks (e.g., waiting queue policy) and communication controllers (e.g., number of buffers), and configure them. RtaW-Sim enables the designer to also perform Simulation Based Fault Injection (SBFI), for instance analyzing the effects of clock drifts orthe impact of transmission errors on transmission latencies.

5. NS-2 simulator

NS-2 stands for network simulator version 2. It is an open-source event-driven simulator
designed specifically for research in computer communication networks.
Features of ns-2 simulator are as follows,
(a) It is a discrete event simulator for networking research.

(b) It provides substantial support to simulate bunch of protocols like TCP, FTP, UDP, HTTP and DSR.
I It simulates wired and wireless network.
(d) It is primarily Unix based.
(e) Uses TCL as its scripting language.
(f) Otcl: Object oriented support.
(g) Tclcl: C++ and otcl linkage.
(h) Discrete event scheduler.

Flow Chart

Figure 5.1, 5.2 and 5.3 shows the flow chart of the paper. First master controller checks whether CAN communication is taking place or not, if it is then CAN1 request for the frame from CAN2 and if CAN2 has accepted the request for frame, it sends the sub sensors data back to CAN1. The data sent takes the necessary value and takes the action accordingly, for eg. If the distance of the obstacle from the ultrasonic sensor is less than the threshold distance (15 cm) the vehicle slows down and stops when this distance is less than 12 cm. If there is any tilt in the axial position of the vehicle, the MEMS sensor takes the value from GPS and sends the GPS value from GSM. If the speed of the vehicle is not within the predefined value, CAN sends the active high signal to arduino. The arduino (sub-slave) takes the value from ultrasonic sensor and matches with the predefined value. If the distance value is lesser than the defined value, the speed of the motor is decreased and receives the value from GPS in kilometer and displays on LCD. If the distance value is greater than the defined value, the speed of the motor is increased.

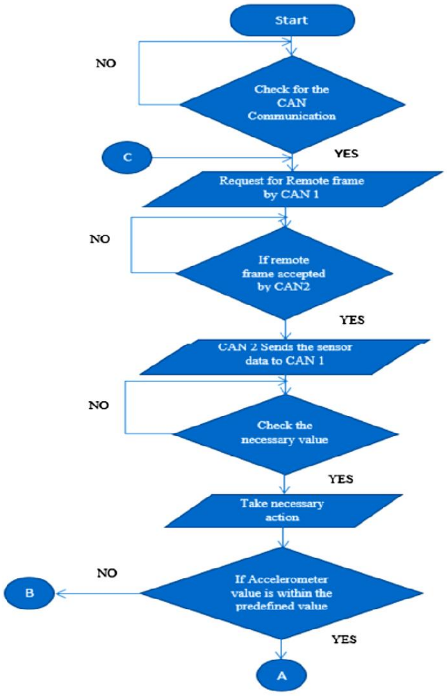
Start
NO
Check for the CAN Communication
C
YES
Request for Remote frame by CAN 1
NO
If remote frame accepted by CAN2
YES
CAN 2 Sends the sensor data to CAN 1
NO
Check the necessary value
YES
Take necessary action
NO
B
If Accelerometer value is within the predefined value
YES
A

Figure 5.1: Flow graph of the system

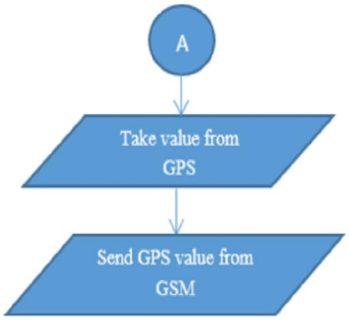

Figure 5.2: Flow graph of the system

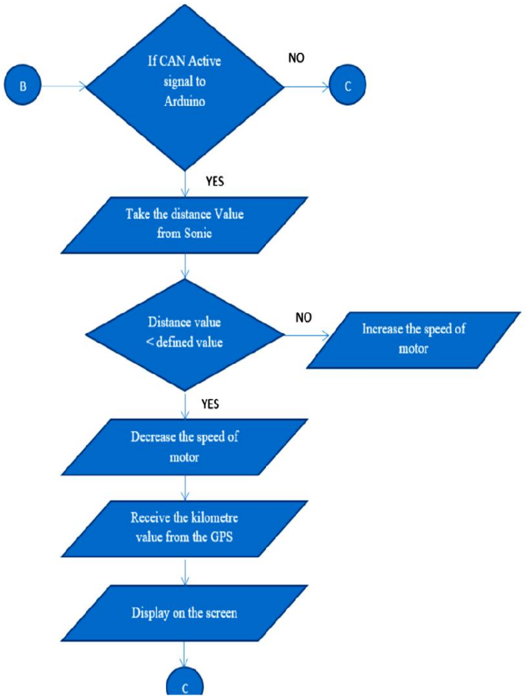
B
If CAN Active signal to Arduino
NO
C
YES
Take the distance Value from Sonic
Distance value < defined value
NO
Increase the speed of motor
YES
Decrease the speed of motor
Receive the kilometre value from the GPS
Display on the screen
C

Figure 5.3: Flow graph of the system

RESULTS

This section gives the results for the implemented parameters. The monitoring parameters are fuel level indication, parking assistance, cruise control, speed. For monitoring the above parameters, fuel oat sensor, IR, ultrasonic sensors are used. The slave sends the data to master, master recon_gures the data and sends it back to the slave and the results are visualize through the LCD. Then the Simulation results are obtained using Kiel micro vision, Rtaw-sim v1.4.7, Flash magic software. Figure 6.1 shows the various parameters of the slave sensors i.e speed, distance, fuel indication, tilt aixs. Figure 6.2 shows the architecture level analysis of CAN with several ECU's, where the frame rate of ECU's are observed.

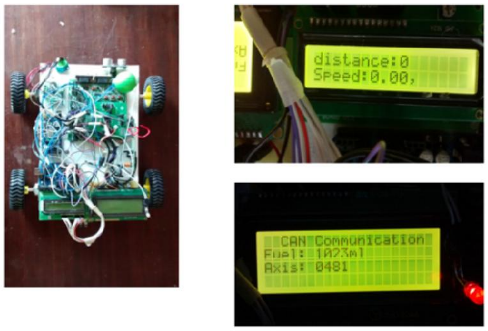

Figure 6.1: (a) Prototype of vehicle (b) Displays the monitoring parameters

The window shows the name and the speed of the bus in kbits/s, as explained by the tool-tip of the label Speed. Furthermore, the periodic bus load is displayed for the chosen bit-stuffng: 10 percent" means that the bit-stuffed part of the frame is 10 percent longer than nominal.

Figure 6.3 represents response time Vs frames in which the collision avoidance of frames from different ECU's. It can be clearly seen that a longer sample implies a higher response time maximum. This phenomenon is the result of the clock drifts that make the inter-ECU frame offsets vary over time and produce other scenarios than those in the periodic behavior encountered in the first simulation. The longer the simulation, the more different trajectories of the system are simulated and the higher the maximal response times of the frames.

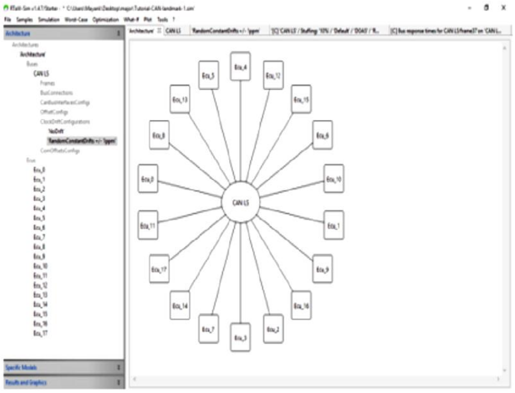

Figure 6.2: Simulations with RtaWsim- CAN simulator

Figure 6.4 represents the transmission of data frames from different ECU's based on the priority. The waiting period (transparent, dotted rectangle) and the transmission period (plain rectangle) of the frame37 are underlined in orange. It can be seen that frame36 and frame42, which are sent by the same ECU1, may be instantiated at the same time than frame37. Furthermore, the clock-drifts lead to a situation where 4 other ECUs (distinguished by colors) instantiate higher priority frames before frame37 is able to win the arbitration.

Figure 6.5 represents the response time from the sender ECU to the receiver ECU with a clock drift. Some drifts are positive (i.e. the clock-drift factor is above 1 and thus the clock faster than nominal) and some are negative (i.e. the clock-drift factor is below 1 and clock slower than nominal) and this within the specified bound of +/- 1 ppm.

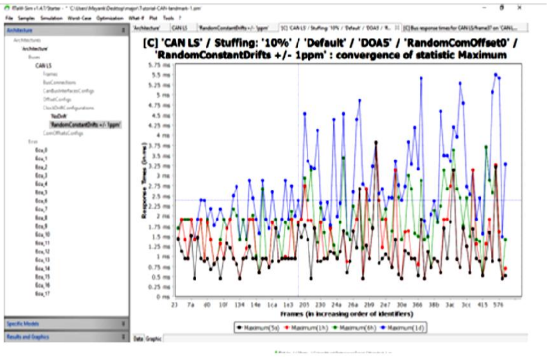

Figure 6.3: Shows the simulated graph of CAN

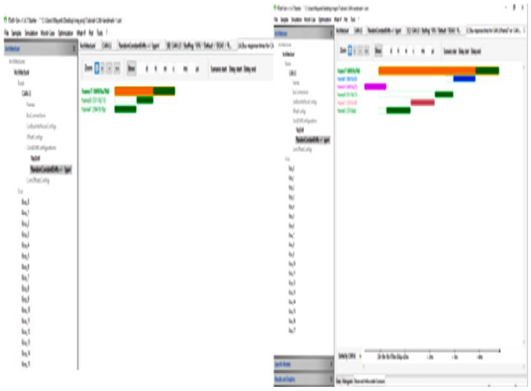

Figure 6.4: Shows the transmission of data.

Figure 6.6 shows the implementation of vector-cannoi software with ABS car engine. By use of NI cabling the ECU's frames are sent and the transmission rate is observed.

Figure 6.7 represents the TCP simulation on ns-2 simulator where loss of packet data is observed between source to destination. Those coloured in blue are low priority packets and has falling rate 10 times of red packet.

Figure 6.8 shows the selection of baud rate on an ABS engine using vector cannoi software.

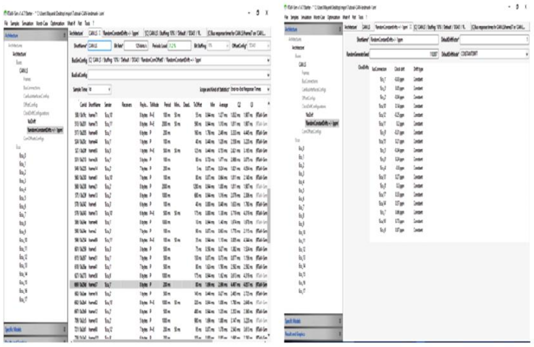

Figure 6.5: Shows the response time on RTaW simulator.

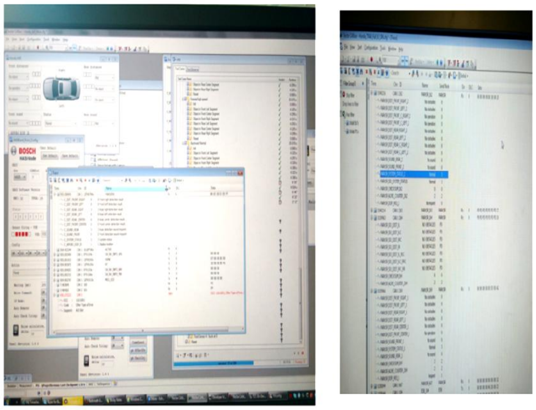

Figure 6.6: Shows the real time simulation on car engine using Vector-Cannoi software Licence tool.

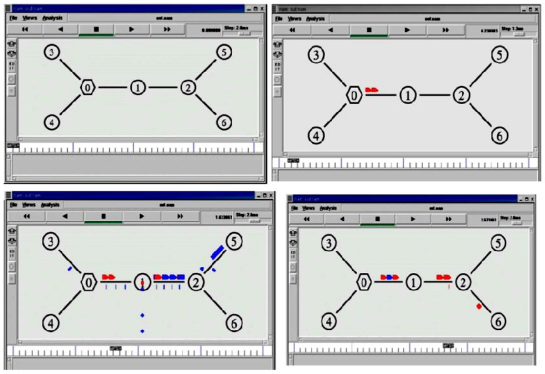

Figure 6.7: Shows TCP simulation on ns-2 Simulator.

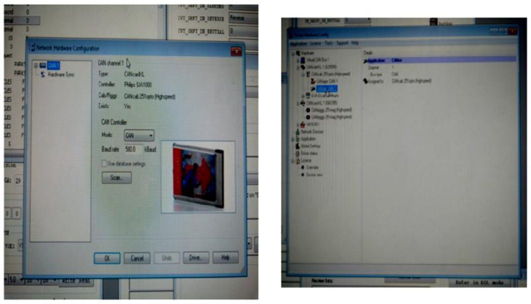

Figure 6.8: Shows real time baud rate selection using vector-cannoi software.

CONCLUSION

This paper implements an embedded system with a combination of CAN bus systems. Digital control of the vehicle is an important criterion of modern technology. The implemented high-speed CAN bus system solves the problem of automotive system applications. The paper has used ARM as the main controller and it makes full use of the high-performance of ARM, high-speed of CAN bus communication control networks and instrument control so as to achieve full sharing of data between nodes and enhance their collaborative work. The paper implements the master and slave communication using CAN protocol in monitoring the parameters like speed control, fuel level indication, parking assistance and location of the vehicle.

FUTURE WORK

This paper can be implemented further with the help of License software on a real car and the wired connection between master and slave can be replaced by zigbee module(wireless communication). Some more additional features can be included like temperature sensor, vibration sensor, pressure sensor, fire alarm sensor and so on.

ACKNOWLEDGMENT

we offer the acknowledgement to the KSCST (Karnataka State Council for Science and Technology) for sponsoring the development fund of our project model and encouraging us.

REFERENCES

[1] Wuhong Wang, 2013."A Digital-Driving system for smart vehicles",IEEE magizine on intelligent transportation system, p.81-83, september/october 2013.

[2] Kumar, M. A.Verma, and A. Srividya, "Response-Time Modeling of Controller Area Network (CAN)". Distributed Computing and Networking, Lecture Notes in Computer Science Volume 5408, p 163-174, 2009.

[3] Tindell, K., A. Burns, and A.J. Wellings, "Calculating controller area network (CAN) message response times". IEEE International Conferenc on Control Engineering Practice, 3(8): p. 1163-1169, 2005.

[4] Jing Yang, Edwin Hou and MengChu Zhou. "Front Sensor and GPS-Based Lateral Control of Automated Vehicles", IEEE Transactions on Intelligent Transportation Systems, Vol. 14, No. 1, March 2013.

[5] B.Gmbh, "CAN speci_cation". vol 1 Version 2.0, 1991.

[6] Bayilmis, C., Kelebekler, E. "Remote control of a CAN-based mobile model car using a voice activated control system",IEEE Transaction on Signal Processing, Communication and Applications Conference,2008.

[7] Navet.N. LORIA.Comput.,Sci.Lab. ,Nancy, France ; Controller area network [automotive applications] Potentials, IEEE Transaction on Volume:17, p. 160-185, 2010.

[8] Prodanov, W., M. Valle, and R. Buzas, A controller area network bus transceiver behavioral model for network design and simulation. IEEE Transactions on Industrial Electronics, 56(9): p. 3762-3770, 2009.

[9] T. Lomas, A. Tuantranont, "Wireless Black Box Using MEMS Accelerometer and GPS tracking for accidental monitoring of vehicles". IEEE International Conference on Biomedical and Health Informatics, 27 Jan 2012.

Author's Details

Mrs. Seema B Hegde *is currently Assistant Professor in the Department of E&CE Siddaganga Institute of Technology, Tumkuru from 13 years. Her area of interest comprises of Embedded systems, Wireless protocols, MANETs/ Opportunistic*

Networks, Cognitive agents based routing solutions, Context aware trust and security in opportunistic computing.

***Mohan Kumar N K**, is currently a final year student in the Department of E&CE Siddaganga Institue of Technology, Tumakuru-03. His area of interest comprises of Micro Electronics, Integrated Electronics & circuits and Digital system processing.*

***Mayank Kumar**, is currently a final year student in the Department of E&CE Siddaganga Institute of Technology, Tumakuru-03. His area of interest comprises of Wireless protocols and mobile wireless networks.*

***Menka Kumari**, is currently a final year student in the Department of E&CE Siddaganga Institue of Technology, Tumakuru-03. Her area of interest comprises of Wireless protocols and Embedded System*

***Rahul Kumar Jha,** is currently a final year student in the Department of E&CE Siddaganga Institue of Technology, Tumakuru-03.His area of interest comprises of Digital system processing, Control and Automation.*