

A
PROJECT REPORT ON

Vehicle Speed Control and Accident Avoidance using CAN Protocol

SUBMITTED IN
PARTIAL FULFILLMENT
OF

DIPLOMA IN EMBEDDED SYSTEM DESIGN (PG-DESD)



BY

- 1. Pooja Sambhaji Garade (80443)**
- 2. Tushar Manohar Khonde (80836)**
- 3. Pavan Pandrinath Karpe (80807)**
- 4. Aditya Bhattacharya (80350)**

AT

**SUNBEAM INSTITUTE OF INFORMATION TECHNOLOGY,
HINJAWADI, PUNE**

**SUNBEAM INSTITUTE OF INFORMATION TECHNOLOGY,
HINJAWADI.**



CERTIFICATE

This is to certify that the project

Vehicle Speed Control and Accident Avoidance using CAN Protocol

Has been submitted by

Pooja Sambhaji Garade (80443)

Tushar Manohar Khonde (80836)

Pavan Pandrinath Karpe (80807)

Aditya Bhattacharya (80350)

In partial fulfillment of the requirement for the Course of **PG Diploma in Embedded System Design (PG-DESD Sept- 2023)** as prescribed by The **CDAC ACTS, PUNE.**

Place: Hinjawadi(PUNE)

Date: 22 Feb 2024

Mr. Sohail Inamdar

Project Guide

ACKNOWLEDGEMENT

It is indeed a matter of great pleasure and proud privilege to be able to present this project on “**Vehicle Speed Control and Accident Avoidance Using CAN Protocol**”. The completion of this project work is great experience in student’s life and its execution is inevitable in hands of guide. We are highly indebted to the Project Guide **Mr. Sohail Inamdar** for his invaluable guidance and appreciation for giving form and substance to this project. It is due to his enduring efforts, patience and enthusiasm which has given sense of direction and purposefulness to this project and alternately made it a success.

I also wish to express my sincere thanks to the CEO of Sunbeam Institute of Information Technology Pune **Mr. Nitin Kudhale** for their immense planning, continuous guidance and talk-shows which kept us involved even in this remote learning batch. Also, I am grateful to all the faculty members specially **Mr. Devendra Dhandhe, Mr. Nilesh Ghule**, for their consistent support and assistance.

I am also grateful to the Lab mentors: **Mr. Vrushabh Patil** and **Mr. Ankush Tembhurnikar** for their consistent support and assistance.

Finally, last but by no means least; also to everyone in the institute, fellow batch-mates it was great sharing a learning platform with all of you during last six months.

Thanks for all your encouragement!

ABSTRACT

The main objective of this project is to enhance safety and convenience by integrating special sensors and smart cruise control mode into a motor control system. Picture a car that not only avoids obstacles but also allows users to select different driving modes based on their preferences! In this project we have used two STM32F407 development boards as two nodes, Node 1 consists of an ultrasonic sensor i.e. HC-SR04 sensor to detect the distance between vehicle and an obstacle. Node 2 consists of actual breaking system. We have used DC motor and L293D as wheel drive and used Minicom via USART as a dashboard. These two nodes are communicating with each other using CAN protocol. Both nodes are connected with one CAN trans-receiver, which is used to adapt signals to CAN bus levels.

The micro-controller will collect data from ultrasonic sensor and via CAN bus it will send to the Node 2 where breaking will happen according to the distance received from Node 1. For this purpose we used ARM based micro-controller STM32F407 and ultrasonic sensor HC-SR04 to measure the distance between the vehicle and an obstacle. The distance measured by an ultrasonic sensor will be displayed on the Minicom.

INDEX

1.	INTRODUCTION	6
2.	LITERATURE SURVEY	8
3.	BLOCK DIAGRAM	10
4.	FLOW CHART	11
5.	HARDWARE DESCRIPTION	12
6.	CONTROLLER AREA NETWORK	22
7.	SOFTWARE REQUIREMENT AND SPECIFICATIONS	27
8.	OUTPUT	28
9.	CONCLUSION	29

1. INTRODUCTION

In today's world, the frequency of accidents is alarmingly high and unpredictable. Accidents can happen at any time and anywhere, often resulting in severe injuries and fatalities. A significant proportion of these accidents occur due to delayed driver reactions in applying the brakes. Our project not only enhances safety by avoiding obstacles but also give users the flexibility to customize their driving experience. Its like having a car that adapts to your preferences, making every journey safer and more reliable. This system relies on an ultrasonic ranging sensor, which operates based on the principles of ultrasonic waves.

The system operates continuously, monitoring the distance between the vehicle and any potential obstacles. When an obstacle is detected within the predefined safety range, the system raises a flag which is transmitted to the motor via CAN communication which promptly applies the brakes.

A **Controller Area Network (CAN)** is main heart of our project. A **Controller Area Network (CAN bus)** is a robust vehicle bus standard designed to allow micro-controllers and devices to communicate with each other's applications without a host computer. It is a message-based protocol, designed originally for multiplex electrical wiring within automobiles to save on copper, but it can also be used in many other contexts. For each device, the data in a frame is transmitted sequentially but in such a way that if more than one device transmits at the same time, the highest priority device can continue while the others back off. Frames are received by all devices, including by the transmitting device. CAN protocol is not an address-based protocol, but message-oriented protocol, wherein the embedded message in CAN has the contents and priority of data being transferred. Up on the reception of data on the bus, each node decides whether to discard or to process the data – and then depending on the system, the network message is destined to single node or many other nodes.

The STM32 have inbuilt CAN controller so we are use MCP2551 (CAN trans-receiver) to establish CAN communication. The sensors are interfaced with STM32 board. In that we use three stm32 board one is master controller and two are slave controllers. Sensors are connected to

the slave controllers. The sensors data transferred from controller through CAN bus. On master controller the received data is compared with threshold value, if the data is beyond the threshold value, then motor will turn on or off to control the speed of vehicle and fan turn on or off to maintain the temperature. Also control the fuel level. The data is received based on the message ID. The whole to be send on the LCD interfacing to the master controller. This whole purpose of our to implementing the vehicle automating system using CAN protocol.

2. LITERATURE SURVEY

Embedded System of DC Motor Speed Control Based on ARM:

Authors of this paper are Helei Wu, Xueqiang Chen, Lingyan Hu. This paper designed a DC motor speed control system. The controller is ARM S3C2410, and the operating system is μ C/OS-II, a real time operating system. Designed a closed loop system of motor speed control, adopted the algorithm of PWM to control the armature voltage, and motor speed is controlled by regulating, of armature voltage. The system has a good response.

Controller area network for vehicle automation :

This book by prof. Ashwini Shinde, Prof. Vidhyadhar and B. Dharmadhikari Based on requirements of modern vehicle, in vehicle Controller Area Network (CAN) architecture has been implemented. In order to reduce point to point wiring harness in vehicle automation, CAN is suggested as a means for data communication within the vehicle environment. The benefits of CAN bus-based network over traditional point to point schemes will offer increased flexibility and expandability for future technology insertions. This paper describes the ARM7 based design and implementation of CAN Bus prototype for vehicle automation. We referred this book to understand the mechanism of CAN bus. It also refers to the understanding of inputs from sensors and further operations the sensor values.

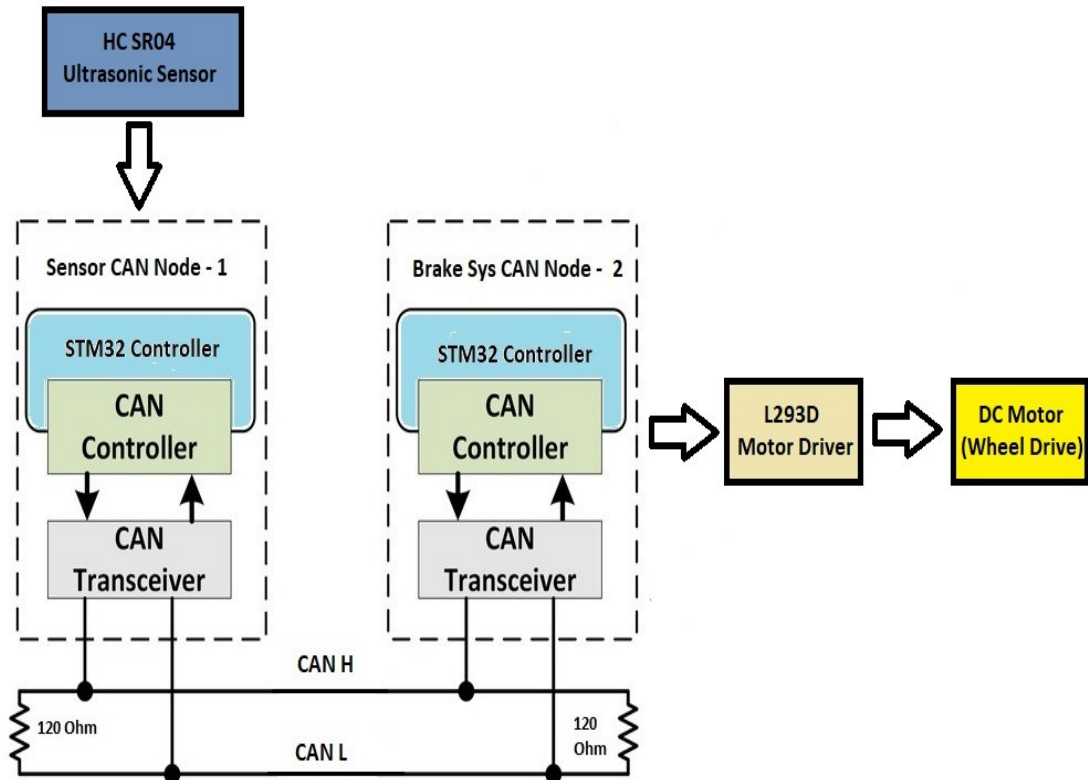
Implementation of Can protocol in automobiles using advanced Embedded system:

Author of this paper Vikash Kumar Sing and Kumari Archana 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. Further it also includes the explanation of temperature sensor mechanism, which we had included into our project.

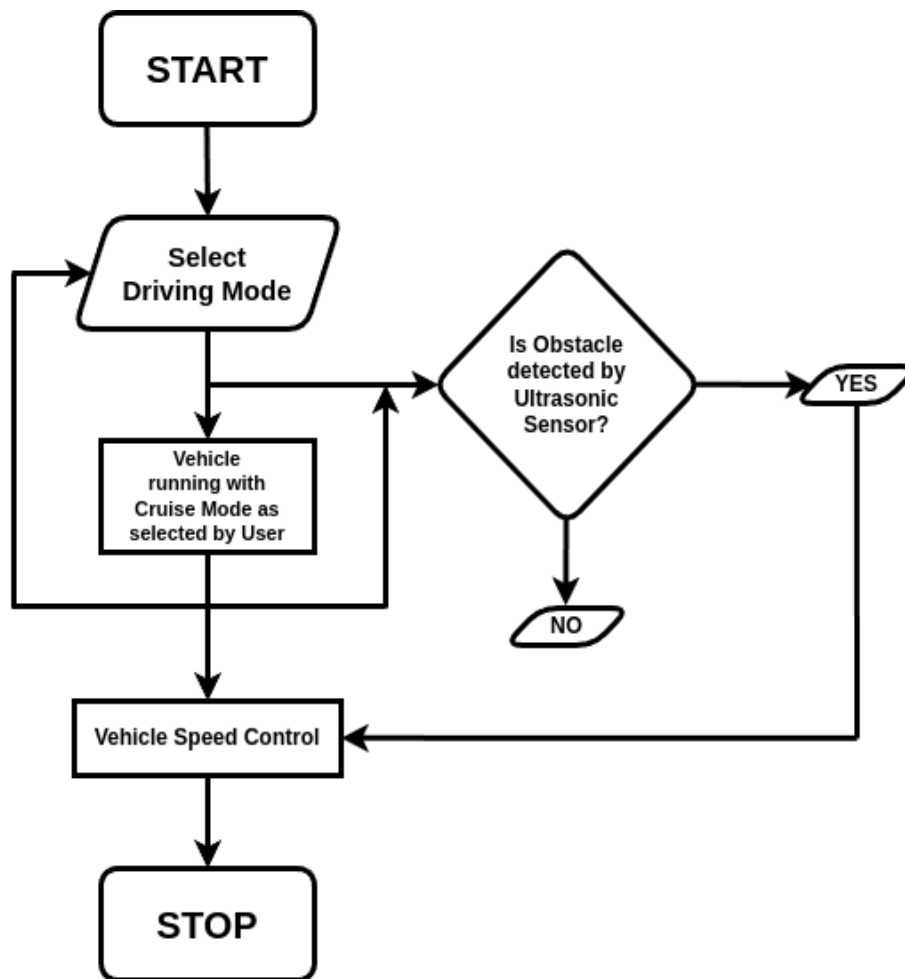
Automotive

The modern automobile may have as many as 70 electronic control units (ECU) for various subsystems. Traditionally, the biggest processor is the engine control unit. Others are used for Autonomous Driving, Advanced Driver Assistance System (ADAS), transmission, airbags, antilock braking/ABS, cruise control, electric power steering, audio systems, power windows, doors, mirror adjustment, battery and recharging systems for hybrid/electric cars, etc. Some of these form independent subsystems, but communications among others are essential. A subsystem may need to control actuators or receive feedback from sensors. The CAN standard was devised to fill this need. One key advantage is that interconnection between different vehicle systems can allow a wide range of safety, economy and convenience features to be implemented using software alone - functionality which would add cost and complexity if such features were "hard wired" using traditional automotive electrics.

3. BLOCK DIAGRAM



4. Flow Chart



5. HARDWARE DESCRIPTION

5.1 STM32F407

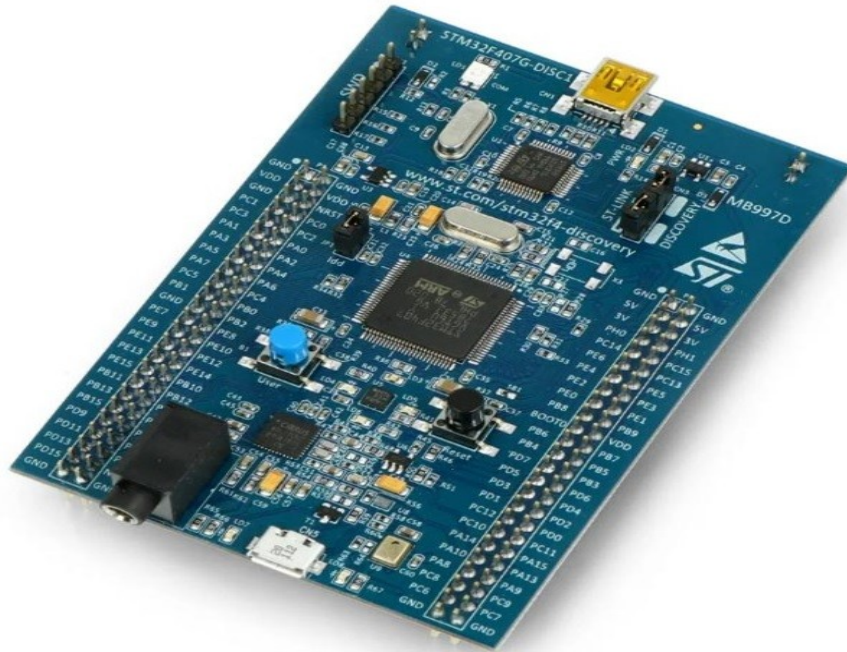


Fig 1: STM32F407

The **STM32F407VG Discovery board** is small devices based on ARM Cortex™-M4 micro-controller, which is a high-performance microcontroller. This board allows users to develop and design applications. It has multiple modules within itself which allows the user to communicate and design the interface of different kinds without relying on any third device. The board has all the modern system modules peripherals like CAN, DAC, ADC, audio port, UART, I2C, etc which makes it one of the best-developing devices.

The STM32F407 lines are designed for medical, industrial and consumer applications where the high level of integration and performance, embedded memories and rich peripheral set inside packages as small as 10 x 10 mm are required. The STM32F407 offers the performance of the Cortex™-M4 core (with floating point unit) running at 168 MHz.

Performance: At 168 MHz, the STM32F407 delivers 210 DMIPS/566 CoreMark performance executing from Flash memory, with 0-wait states

using ST's ART Accelerator. The DSP instructions and the floating-point unit enlarge the range of addressable applications.

- 2x USB OTG (one with HS support)
 - Audio: dedicated audio PLL and 2 full-duplex I²S
 - Up to 15 communication interfaces (including 6x USARTs running at up to 11.25 Mbit/s, 3x SPI running at up to 45 Mbit/s, 3x I²C, 2x CAN, SDIO)
 - Analog: two 12-bit DACs, three 12-bit ADCs reaching 2.4 MSPS or 7.2 MSPS in interleaved mode
 - Up to 17 timers: 16- and 32-bit running at up to 168 MHz
 - Easily extendable memory range using the flexible static memory controller supporting Compact Flash, SRAM, PSRAM, NOR and NAND memories
 - Analog true random number generator
- The STM32F407 product lines provide from 512 Kbytes to 1 MByte of Flash, 192 Kbytes of SRAM, and from 100 to 176 pins in packages as small as 10 x 10 mm.

5.2 CAN Transceiver (MCP 2551)



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.

Transmitter Function

The CAN bus has two states: Dominant and Recessive. A dominant state occurs when the differential voltage between CANH and CANL is greater than a defined voltage (e.g., 1.2V). A recessive state occurs when the differential voltage is less than a defined voltage (typically 0V).

Receiver Function

The RXD output pin reflects the differential bus voltage between CANH and CANL. The low and high states of the RXD output pin correspond to the dominant and recessive states of the CAN bus, respectively.

Operating Modes

The RS pin allows three modes of operation to be selected:

- High-Speed
- Slope-Control
- Standby

When in High-speed or Slope-control mode, the drivers for the CANH and CANL signals are internally regulated to provide controlled symmetry in order to minimize EMI emissions. Additionally, the slope of the signal transitions on CANH and CANL can be controlled with a resistor connected from pin 8 (RS) to ground, with the slope proportional to the current output at RS, further reducing EMI emissions.

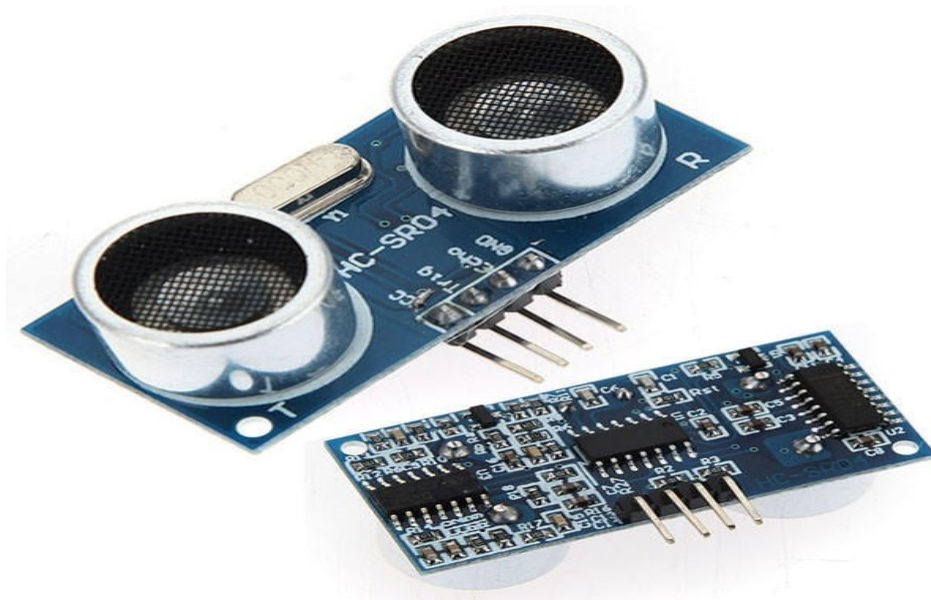
Features

- a) Supports 1 Mb/s operation
- b) Implements ISO-11898 standard physical layer requirements
- c) Suitable for 12V and 24V systems
- d) Externally-controlled slope for reduced RFI emissions
- e) Detection of ground fault (permanent dominant) on TXD input
- f) Power-on reset and voltage brown-out protection

Pin Function

1. VCC :- Supply Voltage
2. GND :- Ground
3. CTX :- Transmit Data Input
4. CRX:- Receive Data Output
5. CANH :- CAN High-Level Voltage I/O
6. CANL :- CAN Low-Level Voltage I/O

5.3 Ultrasonic Sensor



Ultrasonic ranging module HC–SR04 provides 2cm – 400cm non-contact measurement function, the ranging accuracy can reach to 3mm. The modules includes ultrasonic transmitters, receiver and control circuit.

- Working of hcsr04 is pretty simple and straight.
- The module emits an ultrasound at 40 KHz which, after reflecting from obstacle, bounces back to the module.
- By using the **travel time** and the speed of the sound, we can calculate the **distance** between the sensor and the obstacle.

Pin description

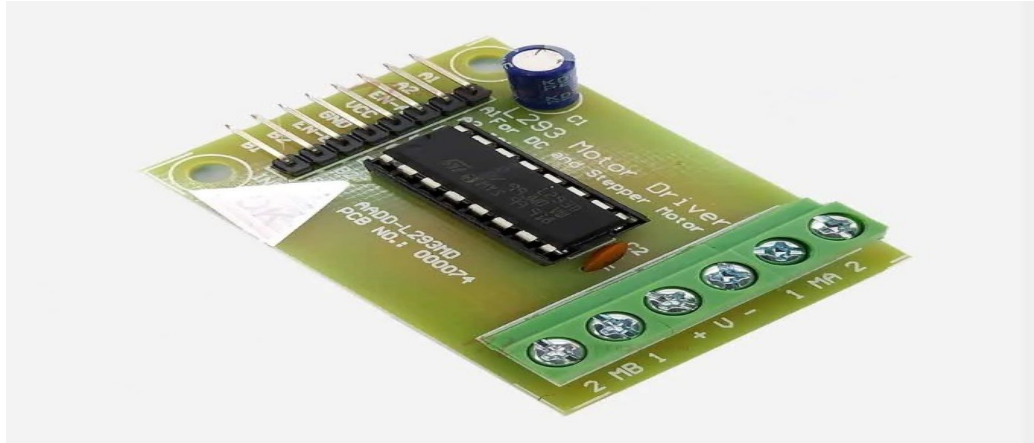
- **VCC** : supplies power to the HC-SR04 ultrasonic sensor. You can connect it to the 5V output from your Arduino.
- **Trig** : Trigger pin is used to trigger ultrasonic sound pulses. By setting this pin to HIGH for 10 μ s, the sensor initiates an ultrasonic burst.
- **Echo** : Echo pin goes high when the ultrasonic burst is transmitted and remains high until the sensor receives an echo, after which it goes low. By measuring the time the Echo pin stays high, the distance can be calculated.
- **GND** : Ground

5.4 DC Motor



A DC motor or direct current motor is an electrical machine that transforms electrical energy into mechanical energy by creating a magnetic field that is powered by direct current. When a DC motor is powered, a magnetic field is created in its stator. The field attracts and repels magnets on the rotor; this causes the rotor to rotate. To keep the rotor continually rotating, the commutator that is attached to brushes connected to the power source supply current to the motors wire windings.

5.5 L293D Motor Driver IC



The Device is a monolithic integrated high voltage, high current four channel driver designed to accept standard DTL or TTL logic levels and drive inductive loads (such as relays solenoids, DC and stepping motors) and switching power transistors. To simplify use as two bridges each pair of channels is equipped with an enable input. A separate supply input is provided for the logic, allowing operation at a lower voltage and internal clamp diodes are included. This device is suitable for use in switching applications at frequencies up to 5 kHz.

The L293D is assembled in a 16 lead plastic package which has 4 center pins connected together and used for heatsinking. The L293DD is assembled in a 20 lead surface mount which has 8 center pins connected together and used for heatsinking.

Specification:

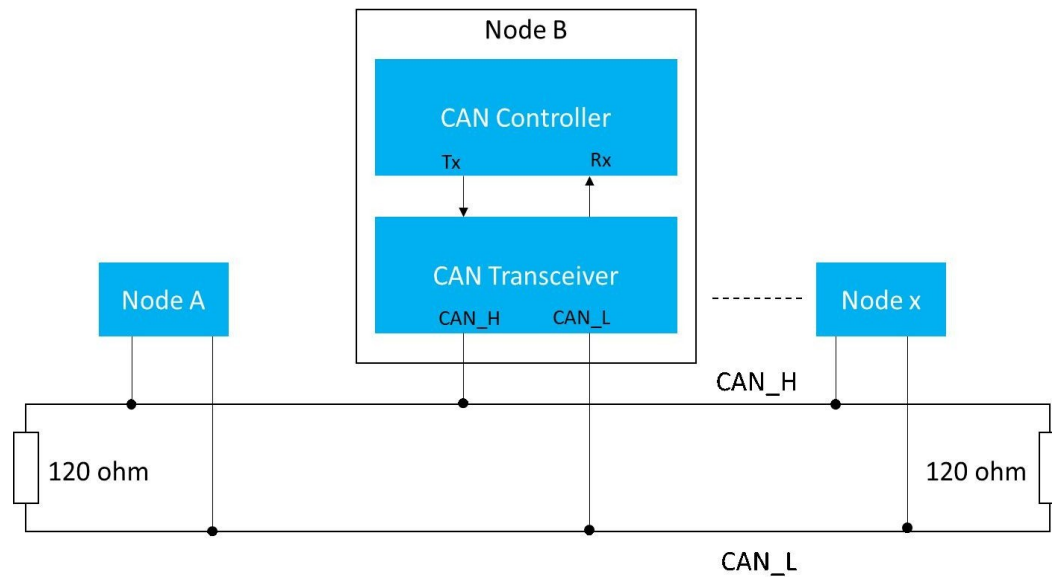
- 600mA OUTPUT CURRENT CAPABILITY PER CHANNEL
- 1.2A PEAK OUTPUT CURRENT (non repetitive) PER CHANNEL
- ENABLE FACILITY
- OVERTEMPERATURE PROTECTION
- LOGICAL "0" INPUT VOLTAGE UP TO 1.5 V (HIGH NOISE IMMUNITY)
- INTERNAL CLAMP DIODES

6. Controller Area Network (CAN)

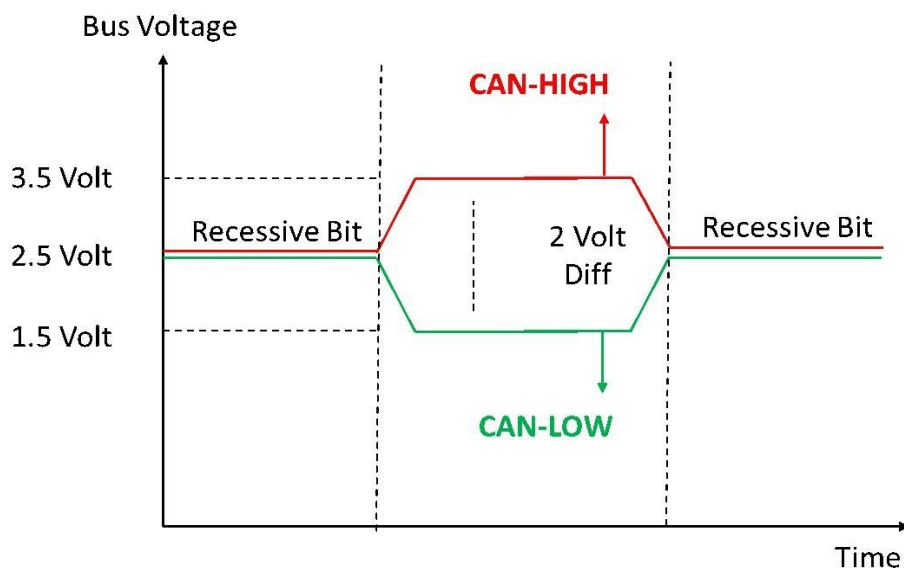
CAN stands for controller area network. They are designed specially to meet the Automobile Industry needs. Before CAN was introduced, each electronic device is connected to other devices using many wires to enable communication. But when the functions in the automobile system increased, it was difficult to maintain because of the tedious wiring system. With the help of the CAN bus system, which allows ECUs to communicate with each other without much complexity by just connecting each ECU to the common serial bus. Hence when compared with the other protocols used in automotive systems i.e., CAN vs LIN, CAN is robust due to less complexity.

CAN Protocol can be defined as a set of rules for transmitting and receiving messages in a network of electronic devices connected through a serial bus. Each electronic device in a CAN network is called a node. Each node must have hardware and software embedded in them for data exchange. Every node of a CAN bus system has a host microcontroller unit, CAN controller and, CAN transceiver in it. CAN controller is a chip that can be embedded inside the host controller or added separately, which is needed to manage the data and sends data via transceiver over the serial bus and vice versa. CAN Transceiver chip is used to adapt signals to CAN bus levels. CAN is a message-based protocol where every message is identified by a predefined unique ID. The transmitted data packet is received by all nodes in a CAN bus network, but depending on the ID, CAN node decides whether to accept it or not. CAN bus follows the arbitration process when multiple nodes try to send data at the same time.

6.1 CAN Bus Electrical Specification:



CAN signals processed by CAN transceiver are single-ended signals and differential signals (CANH and CANL). CAN High and CAN Low lines are at 2.5v in ideal condition. CAN defines logic “zero” as the dominant bit and logic “one” as the recessive bit. When the dominant bit is transmitted, CAN High goes to 3.5v, and CAN low goes to 1.5v i.e., the differential voltage of the dominant bit is 2v. When the recessive bit is transmitted CAN High and CAN Low lines are driven to 2.5v, indicating the differential voltage of the recessive bit is 0v. CAN bus terminal resistor of 120 ohms should be added at the physical end of CANH and CANL lines to avoid any signal reflections.



6.2 Frame types of CAN:

Frame is a defined structure or format that carries meaningful data(bytes) within the network. CAN has four frame types:

1. Data Frame
2. Remote Frame
3. Error Frame
4. Overload frame

Data Frame

Data Frame contains the actual data for transmission. Data Frames consist of fields that provide additional information about the message i.e., Arbitration Field, Control Field, Data Field, CRC Field, a 2-bit Acknowledge Field, and an End of Frame. There are two types of Data frames

1. Standard Frame or Base frame format
2. Extended Frame format

The only difference between the two formats is standard frame supports an 11-bit identifier, and the extended frame supports a 29-bit identifier made up of an 11-bit identifier and extended 18-bit identifier. IDE bit is dominant in a standard frame and recessive in an extended frame.

Base Frame Format:

Standard CAN protocol is also known as the Base frame format. The standard frame is mainly used to send data.

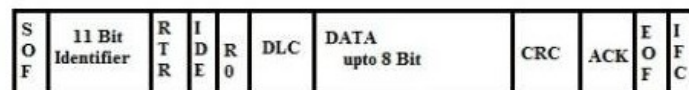


Figure: STANDARD CAN Frame Format

- **SOF** - Start of Frame. Denotes the start of frame transmission.
- **Identifier** - 11-bit unique id and also represents message priority Lower the value, higher is the priority.

- **RTR** - Remote Transmission Request. It is dominant for data frames and recessive for remote frames.
- **IDE** - Single Identification Extension. It is dominant for standard frames and recessive for extended frames.
- **R0** - reserved bit.
- **DLC** - Data Length Code. Defines the length of the data being sent. It is of 4-bit size.
- **Data** - Data to be transmitted and length is decided by DLC.
- **CRC**– Cyclic Redundancy Check. It contains the checksum of the preceding application data for error detection.
- **ACK**– Acknowledge. It is 2 bits in length. It is dominant if an accurate message is received.
- **EOF**– end of the frame and must be recessive.
- **IFS**– Inter Frame Space. It contains the time required by the controller to move a correctly received frame to its proper position.

Extended FRAME:

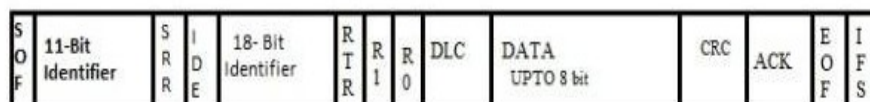


Figure:EXTENDED CAN Frame Format

It is the same as the Standard Frame with some additional fields.

- **SRR**- Substitute Reverse Request. The SRR bit is always transmitted as a recessive bit to ensure that the standard Data frame has high priority when compared to the extended Data frame if both messages have the same 11-bit identifier. It also contains an 18-bit identifier other than an 11-bit identifier.
- **r1**- Reserved bit.

6.3 Advantages of CAN bus Protocol

- Low cost because of reduced wiring
- Saves time due to simple wiring
- Auto retransmission of Lost messages
- Supports Error Detection
- Flexible Data transmission rates

6.4 Disadvantages of CAN bus Protocol

- **CAN Bus Hacking** : CAN bus Hacking is a threat to consumers. CAN Bus hacking may occur through many means, including physical access to the vehicle's diagnostic port, wireless access through vulnerabilities in onboard systems like infotainment units, or through compromised devices such as dongles(OBD-II). Once access is gained, attackers can execute various attacks, including spoofing messages, altering sensor data, injecting malicious commands, or even taking control of steering, brakes or acceleration.

Unauthorized control over critical systems can lead to accidents or vehicle malfunction, privacy breaches, thefts and hijacking.

Many wireless technologies are adopted in CAN bus vehicles like Bluetooth for attending calls or playing music. When onboard system access CAN bus in a car and are capable of Wi-Fi connectivity, it is easy for the hackers to get CAN bus access and be able to control the car. Wi-Fi hotspots are popular in cars, this allows people who know cars' IP address to track the car.

This leads the Car manufacturers to secure transmitted data on the CAN bus network by:

1. Implementing strong encryption and authentication mechanism within CAN Bus network

2. Deploying Intrusion Detection System(IDS) can detect unauthorized access attempts in real-time.
3. Ensuring all the vehicle components employ secure boot mechanism and regularly receive firmware updates.

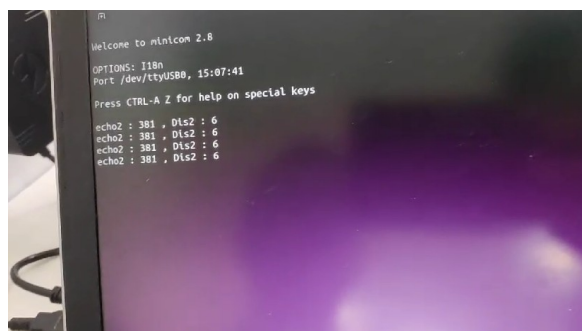
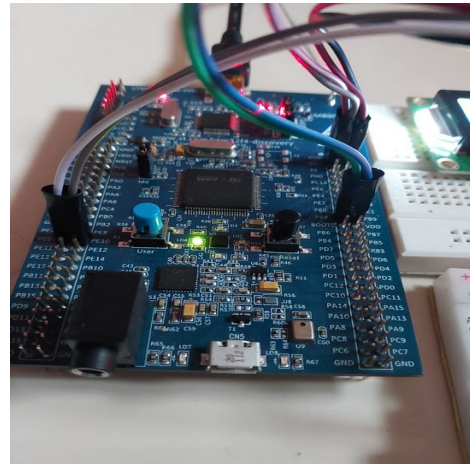
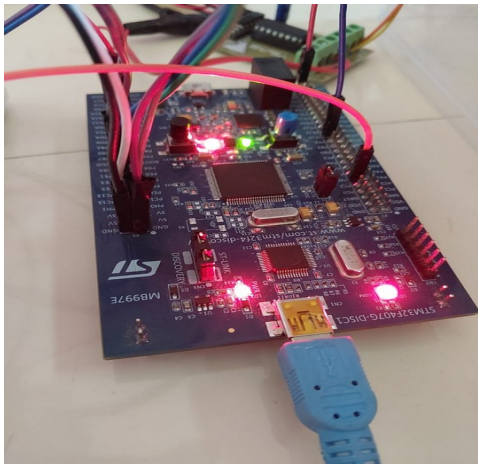
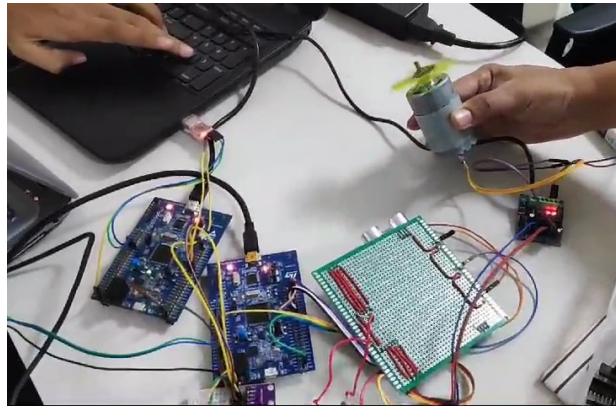
7. SOFTWARE REQUIREMENTS AND SPECIFICATIONS

7.1 STM32CubeIDE



STM32CubeIDE is an all-in-one multi-OS development tool, which is part of the STM32Cube software ecosystem. STM32CubeIDE is an advanced C/C++ development platform with peripheral configuration, code generation, code compilation, and debug features for STM32 microcontrollers and microprocessors. It is based on the Eclipse®/CDT™ framework and GCC toolchain for the development, and GDB for the debugging. It allows the integration of the hundreds of existing plugins that complete the features of the Eclipse.

8. OUTPUT



9. CONCLUSION

This project indicates that utilizing this technology may offer safety advantages, such as reducing driver stress, minimizing lane changes, and promoting longer following distances, especially among younger drivers. However, a significant concern arises from the fact that many drivers lack awareness regarding the limitations of these systems. Our research has shed light on common misconceptions among drivers, including the unfounded belief that these systems can fully prevent collisions with stationary vehicles. To enhance safety, it is imperative to prioritize comprehensive driver education on the circumstances in which these systems may not perform optimally.

Given the nuanced interplay of potential benefits and concerns, further research is indispensable to provide a holistic assessment of the overall safety impact of these systems. Looking forward, the future development of the system could encompass hardware enhancements, including the incorporation of auditory alerts, visual indicators, and the ability to set desired speeds via a keyboard, using Lidar sensors or even bypassing the system over Real Time Operating System.

In future, we might explore adding more advanced features, such as automatic mode selection based on user habits to further improve safety and convenience for users.