A Project report on

**Vehicle Accident Prevention (CAN) and Reporting System**

**using GPS and stm32f407 Microcontroller**



Submitted in partial fulfillment for the award of

**Post Graduate Diploma in EMBEDDED SYSTEMS AND DESIGN**

From C-DAC, Sunbeam (Pune)

**Guided by**

Mr. Ankush Tembhurnikar

**Centre for Development of Advanced Computing (C-DAC), Pune**

**CERTIFICATE**

TO WHOMSOEVER IT MAY CONCERN

This is to certify that

MR. Ashutosh Phegade 240844230058

MR. Bhimashankar Bhosle 240844230015

MISS. Pranjali Vaidya 240844230062

MISS. Pritambra Verma 240844230063

Have completed their project on **Vehicle Accident Prevention (CAN) and Reporting System** **using GPS and stm32f407 Microcontroller** Under the guidance of Mr. Ankush Tembhurnikar.

Project Guide:

Mr. Ankush Tembhurnikar.

**Acknowledgment**

The success and outcome of this project “Vehicle Accident Prevention (CAN) and Reporting System using GPS and stm32f407 Microcontroller” required a lot of guidance and assistance from many people and I am extremely privileged to have got this all along with the completion of my project. All that I have done is only due to such supervision and assistance and I would not forget to thank them.

We take this opportunity to thank Ms. Devendra Dhande and all staff members for the cooperation provided by them in many ways. It is our great pleasure in expressing sincere and deep gratitude toward Ms. for her valuable guidance and constant support throughout this work. I would like to express my gratitude towards my parents & member of CDAC Sunbeam, Pune for their kind co-operation and encouragement which help me in the completion of this project.

My most heartfelt thanks go to Mrs. Nilesh Ghule (Course Co-Ordinator, PG-DESD) who gave all the required support to me.

My thanks and appreciations also go to my batchmates in developing the project and people who have willingly helped me out with their abilities.

Mr. Ashutosh Phegade 240844230058

Mr. Bhimashankar Bhosle 240844230015

Miss. Pranjali Vaidya 240844230062

Miss. Pritambra Verma 240844230063

**Abstract**

The Vehicle Accident Prevention and Reporting System using GPS and the STM32F407 Microcontroller aims to enhance road safety by leveraging advanced technologies for real-time accident detection and efficient emergency response. This system integrates the Controller Area Network (CAN) protocol, which communicates with the vehicle's onboard systems to monitor key parameters such as speed, acceleration, and crash forces. In the event of a collision or dangerous driving behavior, the STM32F407 microcontroller processes sensor data and triggers an automatic accident alert. The system uses GPS to capture the vehicle's precise location and communicates this data to emergency services or pre-configured contacts for immediate response. Additionally, the system can store accident-related information for post-incident analysis and prevention strategies. The proposed solution aims to reduce response times, minimize injury risks, and improve overall vehicle safety, making it a valuable tool in modern vehicular safety systems.

**Contents**

|  |  |
| --- | --- |
| **Certificate** | **ii** |
| **Acknowledgment** | **iii** |
| **Abstract** | **iv** |
| **Contents** | **v** |
| **List of Figures** | **vi** |

|  |  |
| --- | --- |
| 1. Introduction | vii |
| * 1. About Project | vii |
| * 1. System requirements | vii |
| * + 1. Hardware requirements | viii |
| * + 1. Software requirements | Viii |
| 1. CAN controller in STM32 | ix |
| 1. Hardware | xii |
| * 1. STM32f407 Discovery Board | xii |
| * 1. Ultrasonic sensor | xiv |
| * 1. L298N motor driver | xv |
| * 1. DC motor | xvi |
| * 1. Pizo-electric sensor | xvii |
| * 1. GPS NEO 7M | xviii |
| * 1. Wifi module ESP8266 | xix |
| 1. Result | xx |
| 1. Conclusion | xxii |
| 1. Reference | xxiii |
|  |  |

1. **INTRODUCTION**
   1. About Project

Advancement in technologies to have a great vehicular experience, safety system is very essential in automobiles. Accidents can occur anywhere anytime hence there is a need to save human lives from an accident by detecting a mishap before it happens. As traffic hazards and road accidents are increasing day by day it causes huge loss of life and property because of the poor emergency facilities. This project is aimed at advancements in cars for making it more interactive and intelligent for avoiding accidents on roads. As an improvement to safety systems a multi-sensor, control Area Network (CAN) based system is interfaced with Engine Control Unit (ECU) using ARM cortex-M based STM32F407G microcontroller. In order to prevent from accidents Ultrasonic sensors and piezoelectric sensor are used for obstacle detection and also sudden collisions. Global positioning system module and CAN technologies for faster communications make the system completely reliable, safe, and stable and it attains the expected result of real-time analysis of data very effectively to provide a safer drive.

* 1. System Requirements
     1. Hardware Requirements:
* Ultrasonic sensor
* Piezo-electric sensor
* DC motor
* L298N Motor driver
* 2 STM32f407G Discovery Board
* CAN Trans-receiver MCP2551
* GPS Neo7M Module
* ESP8266 (Nodemcu)
* hd44780 (16X2) LCD
* pcf8574 I2C Device
* Registers 120 ohm-2, 1k ohm
  + 1. Software Requirements:

The software used for implementing the project was STM32CubeIDE 1.28.0 for STM32F407VGT6 Discovery Board.

**Testing Environment**

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®/CDTTM framework and GCC toolchain for the development, and GDB for the debugging. It allows the integration of the hundreds of existing plug-ins that complete the features of the Eclipse®IDE.

STM32CubeIDE integrates STM32 configuration and project creation functionalities from STM32CubeMX to offer all-in-one tool experience and save installation and development time. After the selection of an empty STM32 MCU or MPU, or preconfigured microcontroller or microprocessor from the selection of a board or the selection of an example, the project is created and initialization code generated. At any time during the development, the user can return to the initialization and configuration of the peripherals or middleware and regenerate the initialization code with no impact on the user code.

STM32CubeIDE includes build and stack analyzers that provide the user with useful information about project status and memory requirements. STM32CubeIDE also includes standard and advanced debugging features including views of CPU core registers, memories, and peripheral registers, as well as a live variable watch, Serial Wire Viewer interface, or fault analyzer.

All features:

• Integration of services from STM32CubeMX: STM32 microcontroller, microprocessor, development platform and example project selection Pinout, clock, peripheral, and middleware configuration Project creation and generation of the initialization code Software and middleware

completed with enhanced STM32Cube Expansion Packages

• Based on Eclipse®/CDTTM, with support for Eclipse® add-ons, GNU C/C++ for Arm®toolchain and GDB debugger

• STM32MP1 Series: Support for Opens Linux projects: Linux Support for Linux

• Additional advanced debug features including CPU core, peripheral register, memory views Live variable watch view System analysis, and real-time tracing (SWV)CPU fault analysis toolrooms-aware debug support including Azure

• Support for ST-LINK (STMicroelectronics) and J-Link (SEGGER) debug probes

• Import project from Atollic® TrueSTUDIO® and AC6 System Workbench for STM32 (SW4STM32)

• Multi-OS support: Windows®, Linux®, and macOS®, 64-bit versions only.

1. CAN protocol in STM32

The Basic CAN protocol in STM32. Here we will see, how to communicate between two STM32 boards using the CAN protocol. CAN (Controlled Area Network) Protocol is a way of communication between different devices but under certain rules. These rules must be followed when a message is transmitted over the CAN bus. Shown below is the Standard CAN Frame.

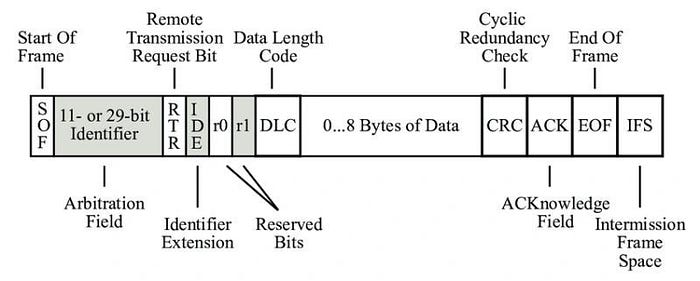


Fig.

CRC and ACK will be handled by the HAL Library.

Here, Identifier is the ID of the transmitting Device RTR (Remote Transmission Request) Specifies if the data is Remote frame or Data frame IDE specifies if we are using Standard ID or Extended ID r is the Reserved bit DLC specifies the data length in Bytes Data Field is where we can send the data, which should be up to 8 bytes Checksum and DEL are the CRC data and it’s Delimiter

ACK and DEL is the acknowledgment bit and its Delimiter

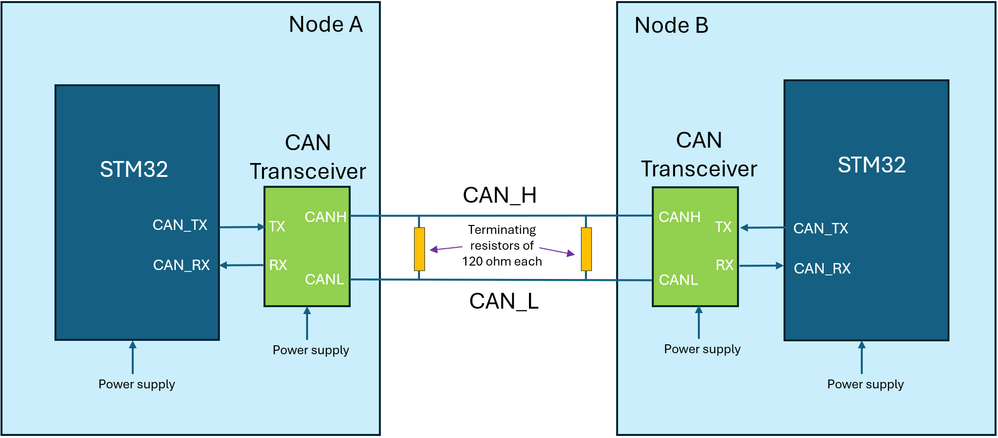
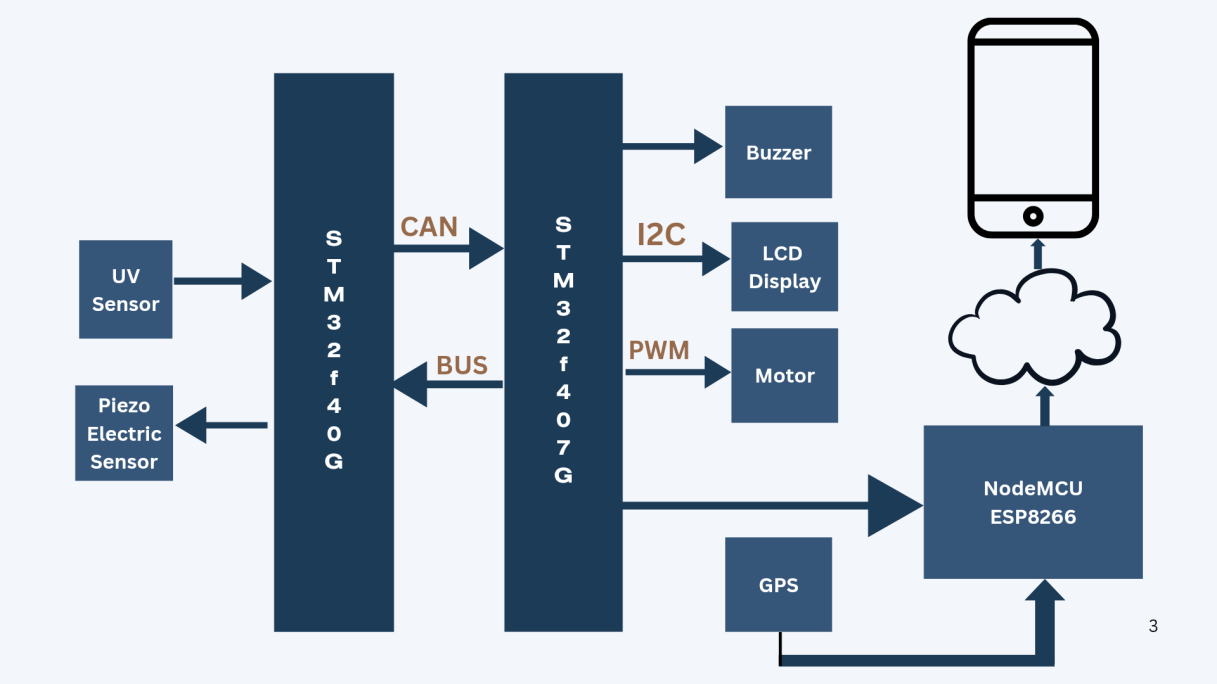


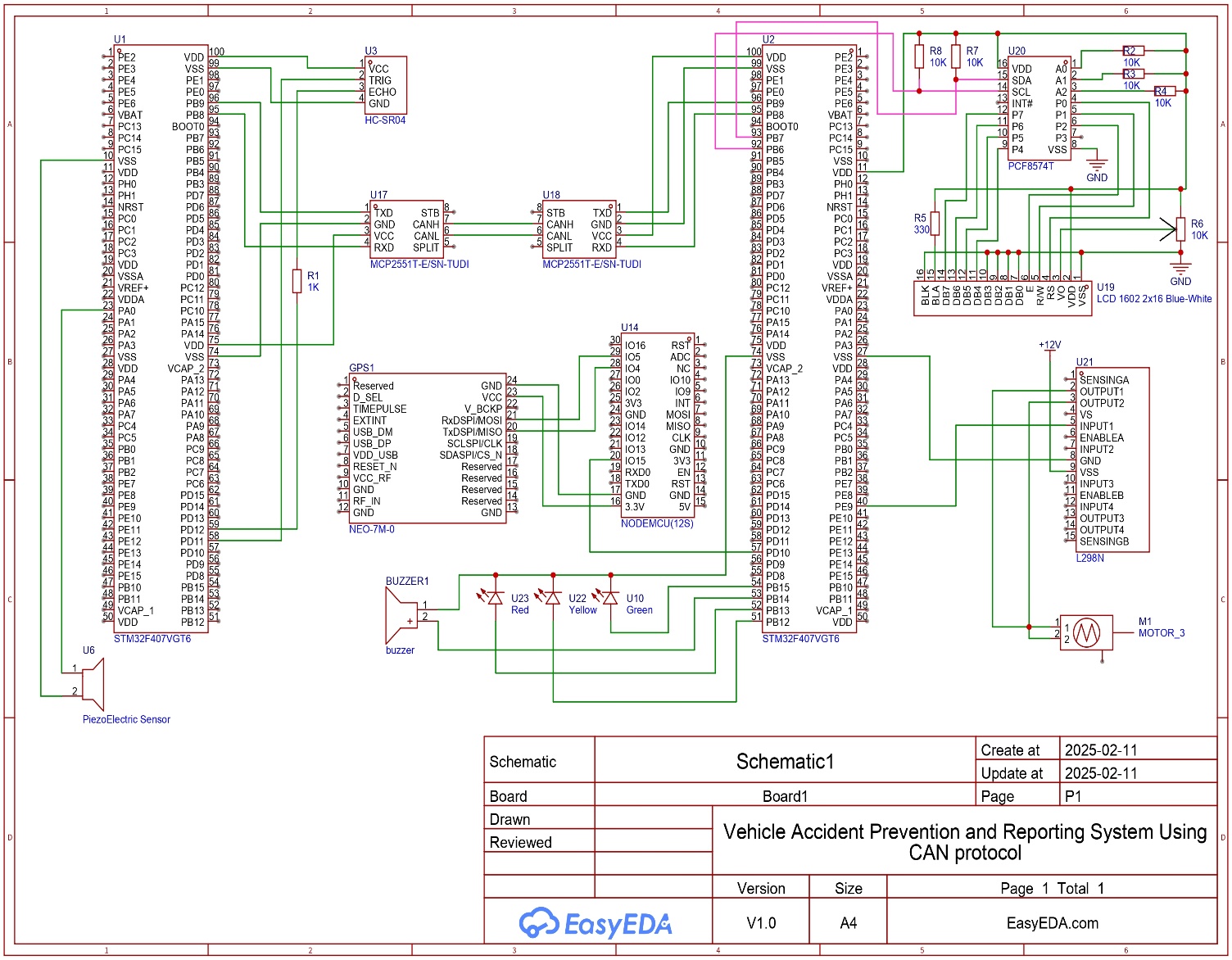
Fig. Connection between STM32f3 board and CAN transceiver

Here the Tx and Rx from the Transceivers are connected to PB9 and PB10 of the Respective controllers CANH and CANL are connected to each other. Here the BAUD RATE is set to 9600 bps. The Operating Mode is NORMAL Mode.



Block Diagram

Circuit Diagram



1. **HARDWARE**

3.1 STM32f407 Discovery Board



Fig. 3.1 STM32f407 Discovery Board

The STM32F4DISCOVERY Discovery kit allows users to easily develop applications with the STM32F407VG high-performance microcontroller with the Arm® Cortex®-M4 32-bit core. It includes everything required either for beginners or experienced users to get started quickly. Based on STM32F407VG, it includes an ST-LINK/V2-A embedded debug tool, one STMEMS digital accelerometer, one digital microphone, one audio DAC with integrated class D speaker driver, LEDs, push-buttons and a USB OTG Micro-AB connector. Specialized add on boards can be connected by means of the extension header connectors. The STM32F4DISCOVERY Discovery kit comes with the STM32 comprehensive free software libraries and examples available with the STM32CubeF4 MCU Package.

Features:

The STM32F4DISCOVERY offers the following features:

• STM32F407VGT6 microcontroller featuring 32-bit Arm®(a) Cortex®-M4 with FPU core, 1-Mbyte Flash memory, 192-Kbyte RAM in an LQFP100 package

• USB OTG FS

• ST MEMS 3-axis accelerometer

• ST-MEMS audio sensor omni-directional digital microphone

• Audio DAC with integrated class D speaker driver

• User and reset push-buttons

• Eight LEDs: – LD1 (red/green) for USB communication – LD2 (red) for 3.3 V power on – Four user LEDs, LD3 (orange), LD4 (green), LD5 (red) and LD6 (blue) – Two USB OTG LEDs, LD7 (green) VBUS and LD8 (red) over-current

• Board connectors: – USB with Micro-AB – Stereo headphone output jack – 2.54 mm pitch extension header for all LQFP100 I/Os for quick connection to prototyping board and easy probing

• Flexible power-supply options: ST-LINK, USB VBUS, or external sources

• External application power supply: 3 V and 5 V

• Comprehensive free software including a variety of examples, part of STM32CubeF4 MCU Package, or STSW-STM32068 for using legacy standard libraries

• On-board ST-LINK/V2-A debugger/programmer with USB re-enumeration capability: mass storage, Virtual COM port, and debug port

• Support of a wide choice of Integrated Development Environments (IDEs) including IAR Embedded Workbench®, MDK-ARM, and STM32CubeIDE

3.2 Ultrasonic Sensor HC-SR04 :

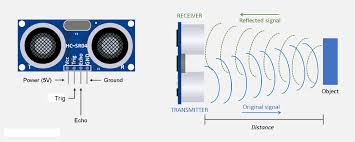
 

Fig. 3.2

Features:

An HC-SR04 ultrasonic distance sensor actually consists of two [ultrasonic transducers](https://en.wikipedia.org/wiki/Ultrasonic_transducer). One acts as a transmitter that converts the electrical signal into 40 KHz ultrasonic sound pulses. The other acts as a receiver and listens for the transmitted pulses. When the receiver receives these pulses, it produces an output pulse whose width is proportional to the distance of the object in front. This sensor provides excellent non-contact range detection between 2 cm to 400 cm (~13 feet) with an accuracy of 3 mm. Since it operates on 5 volts, it can be connected directly to an Arduino or any other 5V logic microcontroller.

The HC­SR04 has 4 pins: VCC, GND, TRIG and ECHO.

1. VCC is a 5v power supply. This should come from the microcontroller

2. GND is a ground pin. Attach to ground on the microcontroller.

3. TRIG should be attached to a GPIO pin that can be set to HIGH

4. ECHO is a little more difficult. The HC­SR04 outputs 5v, which could destroy many microcontroller GPIO pins (the maximum allowed voltage varies). In order to step down the voltage use a single resistor or a voltage divider circuit. Once again this depends on the specific microcontroller you are using you will need to find out its GPIO maximum voltage.

● Power Supply: +5V DC

● Quiescent Current: <2mA

● Working current: 15mA

● Effectual Angle: <15o

● Ranging Distance: 2­400 cm

● Resolution: 0.3 cm

● Measuring Angle: 30 degree

● Trigger Input Pulse width: 10uS

● Dimension: 45mm x 20mm x 15mm

● Weight: approx. 10 g

3.3 L298N Motor Driver

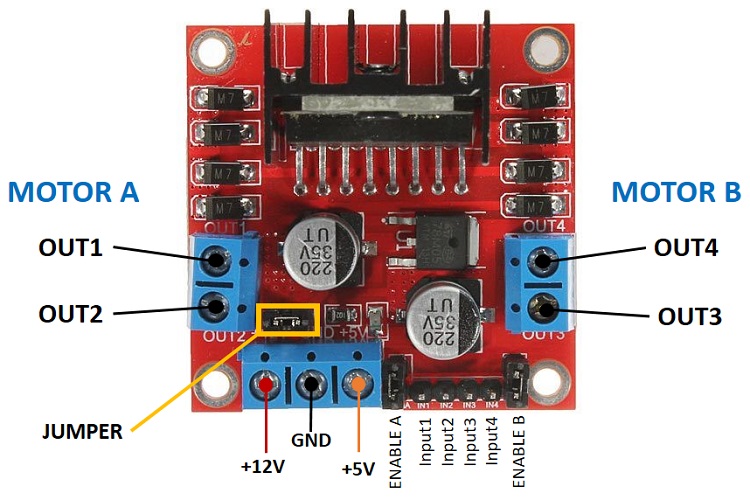


Fig. 3.3

The L298N Motor Driver module consists of an L298 Motor Driver IC, 78M05 Voltage Regulator, resistors, capacitor, Power LED, 5V jumper in an integrated circuit. 78M05 Voltage regulator will be enabled only when the jumper is placed. When the power supply is less than or equal to 12V, then the internal circuitry will be powered by the voltage regulator and the 5V pin can be used as an output pin to power the microcontroller. The jumper should not be placed when the power supply is greater than 12V and separate 5V should be given through 5V terminal to power the internal circuitry. ENA & ENB pins are speed control pins for Motor A and Motor B while IN1& IN2 and IN3 & IN4 are direction control pins for Motor A and Motor B. The L298N is a dual H-Bridge motor driver which allows speed and direction control of two DC motors at the same time. The module can drive DC motors that have voltages between 5 and 35V, with a peak current up to 2A.

Features

* Driver Model: L298N 2A
* Driver Chip: Double H Bridge L298N
* Motor Supply Voltage (Maximum): 46V
* Motor Supply Current (Maximum): 2A
* Logic Voltage: 5V
* Driver Voltage: 5-35V
* Driver Current:2A
* Logical Current:0-36mA
* Maximum Power (W): 25W
* Current Sense for each motor
* Heat-sink for better performance
* Power-On LED indicator

3.4 DC Motor:

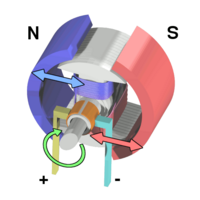
 

Fig. 3.4

A **DC motor** is an [electrical motor](https://en.wikipedia.org/wiki/Electrical_motor) that uses direct current (DC) to produce mechanical force. The most common types rely on magnetic forces produced by currents in the coils. Nearly all types of DC motors have some internal mechanism, either electromechanical or electronic, to periodically change the direction of current in part of the motor. DC motors were the first form of motors widely used, as they could be powered from existing direct-current lighting power distribution systems. A DC motor's speed can be controlled over a wide range, using either a variable supply voltage or by changing the strength of current in its field windings. Small DC motors are used in tools, toys, and appliances. The [universal motor](https://en.wikipedia.org/wiki/Universal_motor), a lightweight [brushed](https://en.wikipedia.org/wiki/Brush_(electric)) motor used for portable power tools and appliances can operate on direct current and alternating current. Larger DC motors are currently used in propulsion of electric vehicles, elevator and hoists, and in drives for steel rolling mills. The advent of [power electronics](https://en.wikipedia.org/wiki/Power_electronics) has made replacement of DC motors with [AC motors](https://en.wikipedia.org/wiki/AC_motors) possible in many applications.

3.5 Piezo-electric Sensor



Fig. 3.5

Pierre Curie founded the piezoelectric effect in 1880, but it started to be used for industrial sensing application in 1950. **Piezoelectric sensor** is used to convert the mechanical stress into electric charge, it gives AC at output. The ability of a piezoelectric material to convert a mechanical stress into electrical charge is called a **Piezoelectric Effect**. The word Piezoelectric derived from the Greek word ‘piezein’ which means to push, press and squeeze. Piezoelectric effect is reversible effect means when we applied mechanical stress to the piezoelectric material we get some electrical charge at output. Same as when we feed electrical charge to the sensor it gets stretch or compresses.

To use a piezoelectric sensor is the easiest task, just connect the positive and negative terminal to your circuit and press the top of sensor. By pressing, due to mechanical pressure it create voltage at output which is further feed to the circuit. You can simply connect a LED to the piezoelectric sensor, as shown in the circuit below. Whenever you press the sensor the LED will give a flash.

3.6 GPS NEO-7M:



Fig. 3.6

The **u-blox NEO-7M** is a **Global Navigation Satellite System (GNSS) receiver module** designed to provide accurate positioning data by receiving signals from satellite systems like GPS, GLONASS, and QZSS. It is part of the **NEO-7 series** of u-blox's compact and low-power GPS modules, specifically tailored for applications that require location tracking and navigation capabilities.

The NEO-7M receives signals from satellites in orbit (such as GPS, GLONASS, and QZSS) to calculate the device's position on Earth. It continuously tracks satellite signals and uses their time data to compute the receiver's location, speed, and time. The module then outputs this data to a connected system via communication interfaces like UART, USB, or SPI.

This sensor has 4 pins:

* **VIN:**Module power supply – 5 V
* **GND:** Ground
* **RX:** Receive data via serial protocol
* **TX:** Sending data via serial protocol

Features

* **GNSS Compatibility**: The NEO-7M supports **GPS (Global Positioning System)**, **GLONASS (Russian Navigation System)**, and **QZSS (Quasi-Zenith Satellite System)**, allowing for global coverage and improved positioning accuracy, especially in challenging environments.
* **Compact Design**: It has a small form factor, measuring only 16 x 12.2 x 2.4 mm, which makes it suitable for integration into small devices, including drones, wearables, and automotive systems.
* **Low Power Consumption**: The NEO-7M is optimized for low energy use, making it ideal for battery-powered devices and portable applications.
* **Multiple Interfaces**: The module supports various communication interfaces, including **UART**, **USB**, **SPI**, and **I2C**, for easy integration with different systems.
* **High Accuracy**: It offers horizontal position accuracy down to **2.5 meters** (when using SBAS assistance) and high velocity and heading accuracy, making it suitable for applications requiring precise geolocation.
* **Fast Time to First Fix (TTFF)**: The module provides quick start-up times for faster location fixes after power-up (about 30 seconds in a cold start).

Top of Form

3.7 Wifi Module ESP8266:



Fig. 3.7

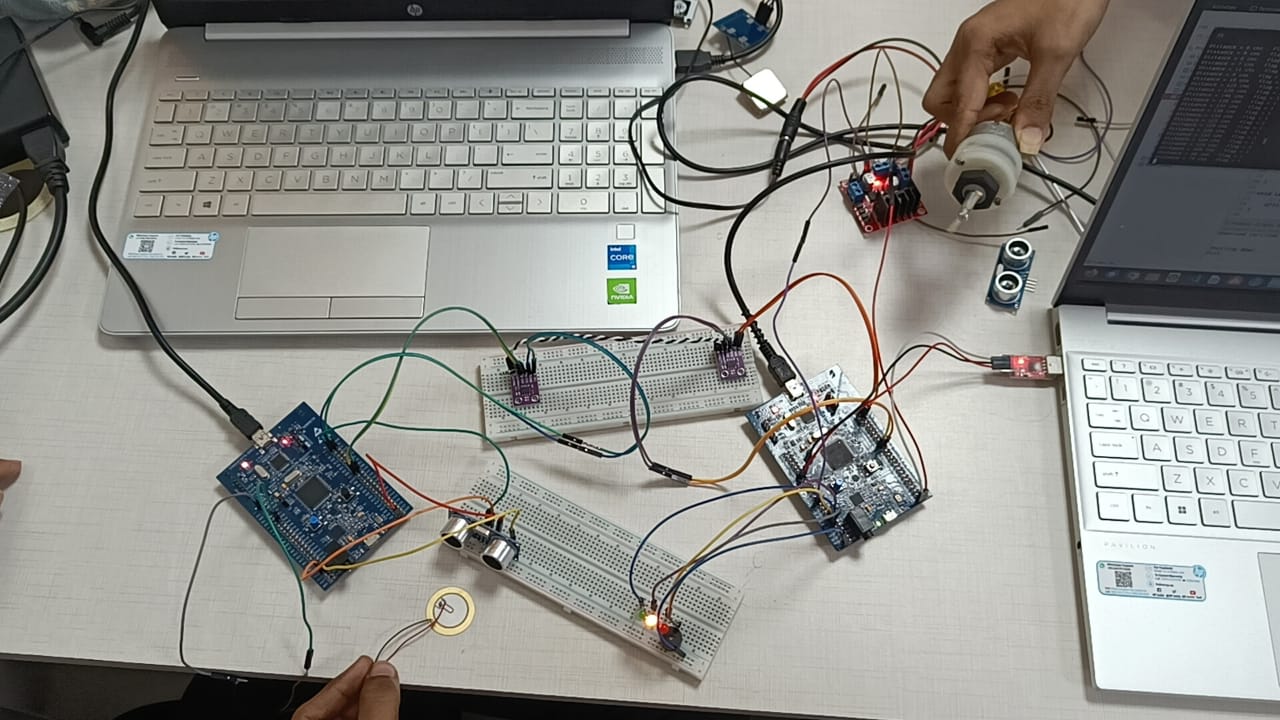
The ESP8266 Wi-Fi module is highly integrated with RF balun, power modules, RF transmitter and receiver, analog transmitter and receiver, [amplifiers](https://www.elprocus.com/power-amplifier-design-for-fm-transmitter/), filters, digital baseband, power modules, external circuitry, and other necessary components. The ESP8266 Wi-Fi module is a microchip.The ESP8266 Wi-Fi module comes with a boot ROM of 64 KB, user data RAM of 80 KB, and instruction RAM of 32 KB. It can support 802.11 b/g/n Wi-Fi network at 2.4 GHz along with the features of I2C, SPI, I2C interfacing with DMA, and 10-bit [ADC](https://www.elprocus.com/analog-to-digital-conversion-using-pic-microcontroller/).  Interfacing this module with the microcontroller can be done easily through a serial port. An external [voltage converter](https://www.elprocus.com/frequency-to-voltage-converter-using-555-ic/) is required only if the operating voltage exceeds 3.6 Volts. It is most widely used in robotics and IoT applications due to its low cost and compact size.

The **ESP8266 Wi-Fi module specifications or features** are given below.

* It is a powerful Wi-Fi module available in a compact size at a very low price.
* It is based on the L106 RISC 32-bit microprocessor core and runs at 80 MHz
* It requires only 3.3 Volts power supply
* The current consumption is 100 m Amps
* The maximum Input/Output (I/O) voltage is 3.6 Volts.
* It consumes 100 mA current
* The maximum Input/Output source current is 12 mA
* The frequency of built-in low power 32-bit MCU is 80 MHz
* The size of flash memory is 513 kb
* It is used as either an access point or station or both
* It supports less than 10 microAmps deep sleep
* It supports serial communication to be compatible with several developmental platforms such as Arduino
* It is programmed using either AT commands, Arduino IDE, or Lua script
* It is a 2.4 GHz Wi-Fi module and supports WPA/WPA2, WEP authentication, and open networks.
* It uses two serial communication protocols like I2C (Inter-Integrated Circuit) and SPI ( Serial Peripheral Interface).
* It provides 10- bit analog to digital conversion
* The type of modulation is PWM (Pulse Width Modulation)
* UART is enabled on dedicated pins and for only transmission, it can be enabled on GPIO2.
* It is an IEEE 802.11 b/g/n Wi-Fi module with LNA, power amplifier, balun, integrated TR switch, and matching networks.
* GPIO pins – 17
* Memory Size of instruction RAM – 32 KB
* The memory size of instruction cache RAM – 32 KB
* Size of User-data RAM- 80 KB
* Size of ETS systems-data RAM – 16 KB

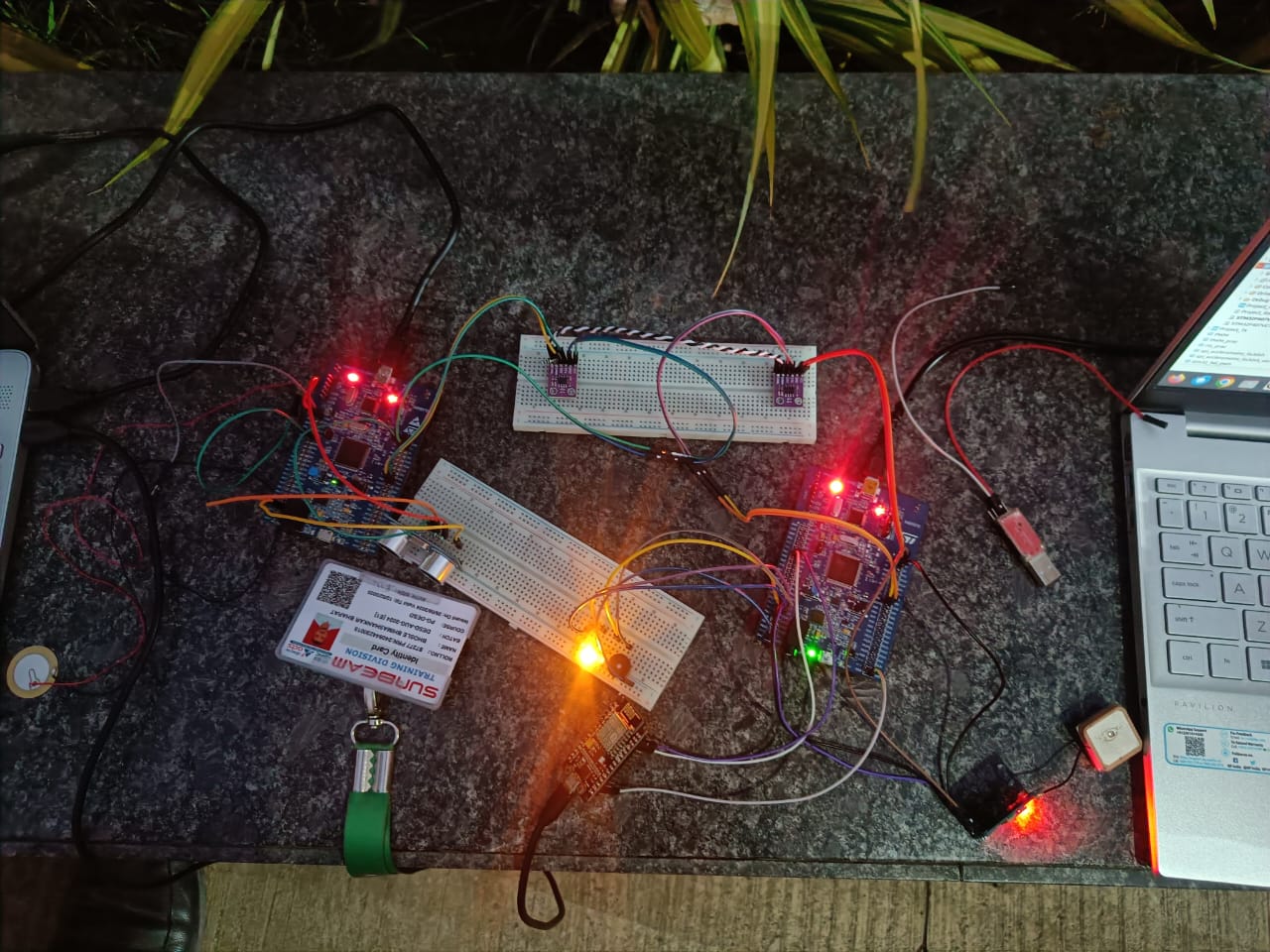
Bottom of Form

1. RESULT:



Hardware connection with Motor operation

We are successful to transmit data from one board to another board using CAN protocol. Ultrasonic sensor sense the distance between vehicle and frontend object and sends the data to another microcontroller board over the CAN bus. The accident is detected using low distance sensed by ultrasonic sensor and piezoelectric sensor. The data is sent from one board to another board over CAN bus and speed of vehicle is controlled using PWM technique. An alert message is also printed on LCD board (dashboard) as warning sign for driver. As the accident is detected, motor operation will be stopped and alert message will be sent to emergency services and the registered contact person. This will help in reduced response time to take appropriate action once the accident happened.



Hardware connection with GPS NEO-7M module

1. **CONCLUSION**

The Vehicle Accident Prevention and Reporting System using GPS and the STM32F407 Microcontroller provides an innovative approach to enhancing road safety and improving emergency response efficiency. By integrating the CAN protocol with real-time data processing, this system offers the ability to detect accidents promptly and automatically relay crucial information, including GPS coordinates, to emergency responders. This technology not only facilitates faster assistance in critical situations but also contributes to reducing the severity of accidents by providing timely intervention. Moreover, the system's capability to log incident details can help improve future safety measures and vehicle designs. Overall, this system represents a significant step forward in accident prevention and road safety, offering both immediate and long-term benefits to drivers, emergency services, and society as a whole.

1. **REFERENCE**

[1] Othman, H.F.; Aji, Y.R.; Fakhreddin, F.T.; Al-Ali, A.R. Controller Area Networks: Evolution and Applications, 2nd Information and Communication Technologies, 2006, vol. 2, pp. 3088 - 3093.

[2] Robert Bosch GmbH, “CAN Specification”, Version 2.0, 1991.

[3] Vehicle Control System using Controller Area Network [Can] Protocol T. Rajasekar1, K. Bhaskar 21 Student, M.Tech, Embedded System Technologies, Vel Tech Dr RR & Dr SR Technical University 2Assistant Professor, Department of EEE.

[4] 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

[5] J.M.Irazabel & S.Blozis, Philips Semiconductors, “I2CManual, Application Note, ref. AN10216-0” March24, 2012.

[6] Vehicle Parameter Monitoring Using CAN Protocol IJCST Jan-Feb 2015 Pratiksha Nawale, Anjali Vekhande, Priyanka Waje

[7] Vehicle Control System using Controller Area Network [Can] Protocol T. Rajasekar1, K. Bhaskar2 1Student, M.Tech, Embedded System Technologies, Vel Tech Dr RR & Dr SR Technical University 2Assistant Professor, Department of EEE.

[8] Vehicle Control Using CAN Protocol for Implementing the Intelligent System (IBS) IJAREEIE March 2014

[9] Venkatesh H. and Rajashri Y Manakwad, “Driver Alerting System Using CAN Protocol”, International Journal of Electrical and Electronics Research, Vol. 3, Issue 1, pp. 218-223, 2015.

[10] N.Q. B. M. Noor and A. Saparon, "FPGA implementation of high speed serial peripheral interface for motion controller," in Proc. 2012 IEEE Symposium on Industrial Electronics and Applications (ISIEA), pp.78-83, Sept.2012