A
Project
Report On

“COVAS : Collision Avoidance and Vehicle Automation System using CAN Protocol”

Submitted in the partial fulfillment of the requirements for the PG Diploma in

EMBEDDED SYSTEMS & DESIGN (PG - DESD)

by

Shaileshkumar Gangaram Zanjare	250244230049
Shreyash Shashikant Patil	250244230040
Nachiket Vaijnath Sanganwar	250244230036
Vipin Vansing Patil	250244230058

Sunbeam Institute of Information Technology, Pune & Karad.

Year : 2025

Sunbeam Institute of Information Technology,
Pune & Karad.

CERTIFICATE

This is to certify,

Shaileshkumar Gangaram Zanjare	250244230049
Shreyash Shashikant Patil	250244230040
Nachiket Vaijnath Sanganwar	250244230036
Vipin Vansing Patil	250244230058

have satisfactorily completed Project and presented a report on topic titled “COVAS: Collision Avoidance and Vehicle Automation System using CAN Protocol” at Sunbeam Institute of Information Technology in partial fulfilment of requirement of PG Diploma in Embedded Systems & Design (PG-DESD) academic year 2025.

Date : 11/08/2025

ACKNOWLEDGEMENT

This project "COVAS: Collision Avoidance and Vehicle Automation System using CAN Protocol" was a great learning experience for us and we are submitting this work to Sunbeam Institute of Information Technology. We all are very glad to Miss. Utkarsha Mam for his valuable guidance to work on this project. His guidance and support helped us to overcome various obstacles and intricacies during the course of project work.

Our most heartfelt thanks goes to Mr. Devendra Dhande (Course Coordinator, PG - DESD) who gave all the required support and kind coordination to provide all the necessities like required hardware, internet facility and extra Lab hours to complete the project and throughout the course up to the last day here in Sunbeam Institute of Information Technology, Pune.

INDEX

LIST OF FIGURES

LIST OF TABLES

ABSTRACT

This project is about creating an automatic braking system (ABS) for vehicles and system monitoring using the Controller Area Network (CAN) protocol for communication. The system uses an ultrasonic sensor to detect objects in front of the vehicle. The data from the sensor is sent to a microcontroller (STM32F407) that processes the information. If an obstacle is detected, the microcontroller sends a signal over the CAN network to another microcontroller, which controls the vehicle's brakes through a motor drive circuit. Where the Rain Sensor and the Temperature sensor sends the reliable information of the rain alert and the system temperature alert over the CAN network, The CAN protocol helps ensure that all parts of the system communicate reliably and quickly, which is important for safety. The system also has a buzzer that sounds an alert when there's danger. This project shows that a simple and effective ABS can be built using common parts and the CAN protocol, improving vehicle safety and reducing the risk of collisions. In the future, more sensors could be added for better awareness of the surroundings, and smarter control systems could improve braking performance

1. INTRODUCTION

About Project:

This project aims to reduce vehicle accidents by developing an automatic braking system (ABS) that uses an ultrasonic sensor to detect obstacles ahead. The data is processed by an STM32F407 microcontroller, which sends commands over the CAN protocol to a motor circuit that activates the brakes if a collision risk is detected. Additional rain and temperature sensors provide real-time alerts on environmental conditions, all communicated quickly and reliably through the CAN network. A buzzer warns the driver of any imminent danger. By combining sensor data and CAN-based monitoring, the system helps prevent collisions and improves overall vehicle safety.

1.1 Project scope:

The automatic braking system is developed to improve driver safety and comfort by automatically applying brakes when a potential collision is detected. This system helps overcome situations where drivers may misjudge the distance to an obstacle, reducing the risk of accidents and vehicle damage. It utilizes an ultrasonic sensor for obstacle detection, with data processed by a microcontroller that communicates over the CAN protocol to control braking accurately and promptly.

In addition to braking, the system features rain and temperature sensors that provide timely alerts to the driver—through visual indicators like LEDs—about adverse weather conditions and critical system temperatures. These alerts help increase driver awareness and preparedness in varying environments.

By integrating these sensors with reliable CAN-based communication, the system offers an innovative and effective safety solution, aligning with modern automotive demands for enhanced comfort and security. This approach not only aids in accident prevention but also contributes new ideas to the advancement of vehicle safety technologies.

1.2 CAN Protocol:

The CAN bus was developed by BOSCH as a multi-master, message broadcast system that specifies a maximum signaling rate of 1 megabit 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 provides for data consistency in every node of the system. CAN is an International Standardization Organization (ISO)-defined serial communications bus originally developed for the automotive industry to replace the complex wiring harness with a two-wire bus.

The specification calls for high immunity to electrical interference and the ability to self-diagnose and repair data errors.

CAN Frame Format:

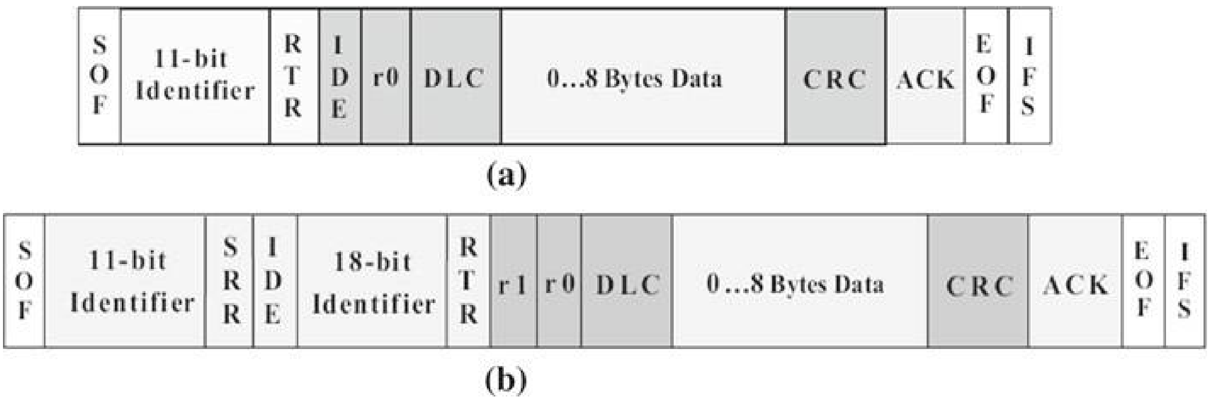

Fig 1.1 CAN Frame Format.

The following figure shows the CAN protocols' frame format.

- SOF: The single dominant start of frame (SOF) bit marks the start of a message and is used to synchronise the nodes on a bus after being idle.
- Identifier: The standard CAN 11-bit identifier establishes the priority of the message. The lower the binary value, the higher its priority.

- RTR: The single remote transmission request (RTR) bit is dominant when information is required from another node. All nodes receive the request, but the identifier determines the specified node. The responding data is also received by all nodes and used by any node interested. In this way, all data being used in a system is uniform.

- IDE: A dominant single identifier extension (IDE) bit means that a standard CAN identifier with no extension is being transmitted.
- ro: Reserved bit (for possible use by future standard amendment).
- DLC: The 4-bit data length code (DLC) contains the number of bytes of data being transmitted.
- Data: up to 64 bits of application data may be transmitted.
- CRC: The 16-bit (15 bits plus delimiter) cyclic redundancy check (CRC) contains the checksum (number of bits transmitted) of the preceding application data for error detection.
- ACK: Every node receiving an accurate message overwrites this recessive bit in the original message with a dominant bit, indicating an error-free message has been sent. Should a receiving node detect an error and leave this bit recessive, it discards the message, and the sending node repeats the message after re-arbitration. In this node acknowledges (ACK) the integrity of its data. ACK is 2 bits; one is the acknowledgement bit and the second is a delimiter. way, each
- EOF: This end-of-frame (EOF), 7-bit field marks the end of a CAN frame (message) and disables bit-stuffing, indicating a stuffing error when dominant. When 5 bits of the same logic level occur in succession during normal operation, a bit of the opposite logic level is stuffed into the data.
- IFS: This 7-bit inter-frame space (IFS) contains the time required by the controller to move a correctly received frame to its proper position in a message buffer area.

The Extended CAN Frame format consists of these additional bits:

- SRR: The substitute remote request (SRR) bit replaces the RTR bit in the standard message location as a placeholder in the extended format.
- IDE-A recessive bit in the identifier extension (IDE) indicates that more identifier bits follow. The 18-bit extension follows IDE.
- r1: Following the RTR and r0 bits, an additional reserve bit has been included ahead of the DLC bit.

System Requirements

Hardware requirements are as follows:

1. 2x STM32F407 Discovery Board.
2. 2x CAN transceivers (MCP2551)
3. 1x Ultrasonic Sensor (HCSR04)
4. 1x Motor Drive module (L298N)
5. 1x DC motor
6. 1x LM35 Temperature Sensor
7. 1x USB 2.0 to TTL UART serial converter (CP2102)
8. 1x 1Kfi Resistor
9. 2x 120fi Resistor

Software requirements are as follows:

1. The software we have used for implementing the project is STM32Cube IDE 1.18.0.

2. HARDWARE REQUIREMENTS

1. STM32F407 Discovery Board

The STM32F407G-DISC1 Discovery Kit is designed to help both beginners and experienced users explore the features of the STM32F407/417 line and develop applications. It's based on the STM32F407VGT6 microcontroller and includes an embedded ST-LINK/V2 debug tool, MEMS sensors, an audio DAC, LEDs, push buttons, and a USB OTG micro-AB connector.

Key Features and Specifications:

- Microcontroller: STM32F407VGT6 featuring a 32-bit ARM Cortex-M4F core, 1 MB Flash, and 192 KB RAM in an LQFP100 package. The core runs at up to 168 MHz and includes a floating-point unit (FPU) and Digital Signal Processing (DSP) instructions for high-performance embedded applications.
- On-board ST-LINK/V2: With a selection mode switch, it can be used as a standalone ST-LINK/V2 with an SWD connector for programming and debugging. Newer boards use ST-LINK/V2-A.
- Power Supply: Can be powered through a USB bus or an external 5 V supply voltage. It also provides an external application power supply at 3 V and 5 V.
- MEMS: Includes a LIS302DL or LIS3DSH ST MEMS 3-axis accelerometer and an MP45DT02 ST MEMS audio sensor, which is an omnidirectional digital microphone.
- LEDs: Eight LEDs are present: LD1 (red/green) for USB communication, LD2 (red) for 3.3 V power on, four user LEDs (orange, green, red, and blue)
- Push Buttons: It has two push buttons for user input and reset.
- USB OTG FS: Comes with a micro-AB connector.
- Debug and Programming: The ST-LINK circuitry allows you to flash code onto the microcontroller and debug it using host software like Keil or STM Cube IDE.

  The board comes in two versions, a newer one labeled "DISC1" and an older one without this designation. The board identification number MB997D is for the newer board, while MB997C is for older versions.

  The STM32F4 Discovery board allows users to develop and design applications, offering modules for communication and interface design without relying on third-party devices. It incorporates modern system modules and peripherals like DAC, ADC, and UART.

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

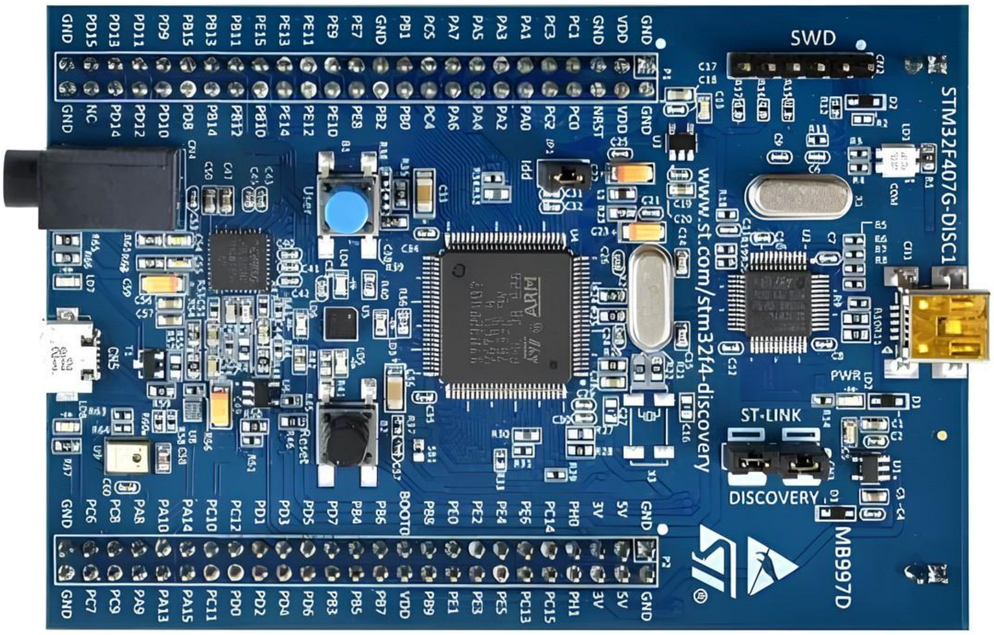

Fig 2.1 STM32F407 Discovery Board

2. MCP2551 CAN Transceivers:

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

Some of its features are:
• Supports 1 Mb/s operation
• Implements ISO-11898 standard physical layer requirements
• Suitable for 12V and 24V systems
• Detection of ground fault (permanent Dominant) on TXD input
• Power-on Reset and voltage brown-out protection
• An unpowered node or brown-out event will not disturb the CAN bus
• Low current standby operation
• Protection against damage due to short-circuit conditions (positive or negative battery voltage)
• Up to 112 nodes can be connected
• High-noise immunity due to differential bus implementation
• Temperature ranges: - Industrial (I): -40°C to +85°C - Extended (E): -40°C to +125°C

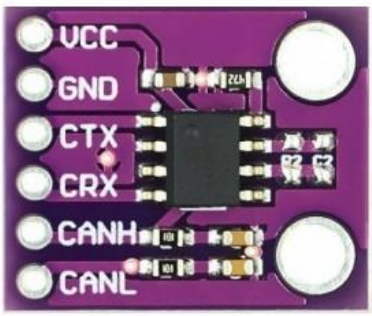

Fig 2.2 MCP255

3. L298N Motor Drive Module.

The L298N Motor Driver Module is a high-power module designed for driving DC and Stepper Motors, containing an L298 motor driver IC and a 78M05 5V regulator. It can control up to four DC motors or two DC motors with directional and speed control.

Pinout Configuration:

- IN1 & IN2 : Motor A input pins for controlling spinning direction.
- IN3 & IN4 : Motor B input pins for controlling spinning direction.
- ENA : Enables PWM signal for Motor A.
- ENB : Enables PWM signal for Motor B.
- OUT1 & OUT2 : Output pins of Motor A.
- OUT3 & OUT4 : Output pins of Motor B.
- 12V : 12V input from DC power source.
- 5V : Supplies power for the switching logic circuitry inside L298N IC.
- GND : Ground pin.

Features & Specifications:

- 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
- Heatsink
- Power-On LED indicator

The control mode and state of motor:

ENA	IN1	IN2	STATE
1	0	0	Brake
1	0	1	Rotate Clockwise
1	1	0	Rotate Anti-Clockwise
1	1	1	Brake

Table 2.1. Motor States.

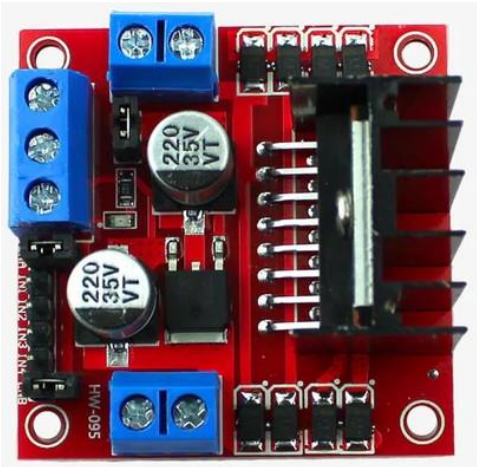

Fig. 2.

4. Ultrasonic Sensor (HC-SR04)

Ultrasonic ranging module HC - SR04 provides 2cm - 400cm non-contact measurement function, the ranging accuracy can reach to 3mm. The modules include ultrasonic transmitters, receiver and control circuit. The basic principle of work:
1) Using IO trigger for at least 10us high level signal,
2) The Module automatically sends eight 40 kHz and detects whether there is a pulse signal back.
3) IF the signal back, through high level, time of high output IO duration is the time from sending ultrasonic to returning.
Test distance = (high level time velocity of sound (340M/S)/2

Wire connecting direct as following:
• 5V Supply
• Trigger Pulse Input
• Echo Pulse Output
• 0V Ground

Working Voltage	DC 5 V
Working Current	15 mA
Working Frequency	40 KHz
Max Range	4m
Min Range	2cm
Measuring Angle	15 degrees
Trigger Input Signal	10uS TTL pulse
Echo Output Signal	Input TTL lever signal and the range in Proportion
Dimension	45*20*15mm

Table 2.2. Electric parameters of HC-SR04

Fig 2.4. HCSR04

Timing diagram:

The timing diagram is shown below. You only need to supply a short 10 µs pulse to the trigger input to start the ranging, and then the module will send out an 8-cycle burst of ultrasound at 40 kHz and raise its echo. The Echo is a distance object that is pulse width and the range in proportion. You can calculate the range through the time interval between sending the trigger signal and receiving the echo signal. Formula: us / 58 = centimeters or us / 148 inches; or: the range = high-level time * velocity (340 m/s) / 2; we suggest using over a 60 ms measurement cycle in order to prevent the trigger signal from reaching the echo signal.

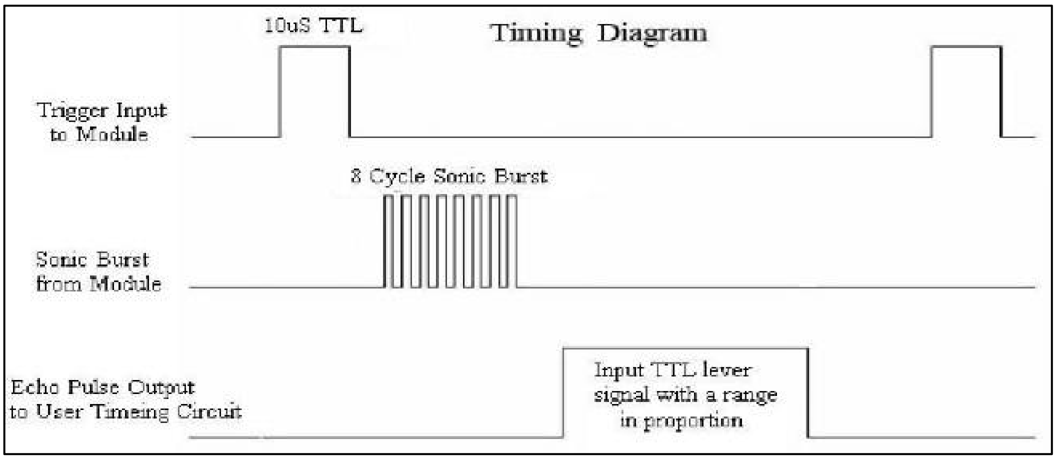

Fig 2.5. Timing Diagram of HCSR-04

5. DC Motor.

Almost every mechanical movement that we see around us is accomplished by an electric motor. Electric machines are a means of converting energy. Motors take electrical energy and produce mechanical energy. Electric motors are used to power hundreds of devices we use in everyday life. Motors come in various sizes. Huge motors that can take loads of 1000s of horsepower are typically used in the industry. Some examples of large motor applications include elevators, electric trains, hoists, and heavy metal rolling mills. Examples of small motor applications include motors used in automobiles, robots, hand power tools, and food blenders. Micro-machines are electric machines with parts the size of red blood cells and find many applications in medicine. Electric motors are broadly classified into two different categories: DC (Direct Current) and AC (Alternating Current). Within these categories are numerous types, each offering unique abilities that suit them well for specific applications. In most cases, regardless of type, electric motors consist of a stat or (stationary field) and a rotor (the rotating field or armature) and operate through the interaction of magnetic flux and electric current to produce rotational speed and torque. DC motors are distinguished by their ability to operate from direct current. There are different kinds of D.C. motors, but they all work on the same principles. In this chapter, we will study their basic principle of operation and their characteristics. It's important to understand motor characteristics so we can choose the right one for our application requirements. The learning objectives for this chapter are listed below.

DC Motor Basic Principles

- Energy Conversion

If electrical energy is supplied to a conductor lying perpendicular to a magnetic field, the interaction of current flowing in the conductor and the magnetic field will produce mechanical force (and therefore, mechanical energy).

- Value of Mechanical Force

There are two conditions that are necessary to produce a force on the conductor. The conductor must be carrying current and must be within a magnetic field. When these two conditions exist, a force will be applied to the conductor, which will attempt to move the conductor in a direction perpendicular to the magnetic field. This is the basic theory by which all DC motors operate. The force exerted upon the conductor can be expressed as follows.

$F = B * i * l$ Newton

where B is the density of the magnetic field, l is the length of the conductor, and i is the value of the current flowing in the conductor. The direction of motion can be found using Fleming's Left Hand Rule.

The first finger points in the direction of the magnetic field (first - field), which goes from the North Pole to the South Pole. The second finger points in the direction of the current in the wire (second - current). The thumb then points in the direction the wire is thrust or pushed while in the magnetic field (thumb: torque or thrust).

Fig 2.6. DC Motor

6. **USB to TTL UART serial converter.**

The CP2102 USB to TTL UART serial converter facilitates a serial connection to a computer for embedded systems. It appears as a standard COM port when attached to a USB bus.

Key features and specifications:

- Chip: Utilizes the CP2102 USB-to-UART bridge controller.
- USB Compliance: USB specification 2.0 compliant, with full-speed 12Mbps.
- UART Interface: Implements all RS-232 signals, including control and handshaking signals.
- Baud Rates: Supports baud rates from 300bps to 921600 bps.
- Voltage: TTL compatible with 5V and 3.3V. Features dual power output at both 3.3V and 5V. An on-chip voltage regulator provides 3.3 V output.

- Connectors: Standard USB type A male and TTL 6-pin connector. The 6 pins are for 3.3V, RST, TXD, RXD, GND, and 5V.

Fig 2.7. CP2120

7. LM35 Temperature Sensor

The LM35 is a precision integrated-circuit temperature sensor that provides an output voltage linearly proportional to the Celsius temperature.

Key features and specifications:

Sensor type: Analog temperature sensor with linear output

Output voltage scale: 10 mV/°C (Linear)

Temperature range: -55°C to +150°C (wide operating range)

Accuracy: ±0.5°C typical at 25°C

Operating voltage: 4 V to 30 V

Current consumption: Low quiescent current, typically 60 µA

Output impedance: Approximately 0.1 Ω

Self-heating: Very low, less than 0.1°C in still air

Pin configuration: 3-pin (Vcc, Output, Ground)

Package types: Available in TO-92, TO-220, TO-CAN, and SOIC packages

Interface: Provides analog voltage output proportional to temperature for easy interfacing with microcontrollers and ADCs

Additional info: No external calibration needed; suitable for remote temperature sensing applications

The LM35 sensor is widely used in applications like environmental monitoring, automotive temperature sensing, HVAC systems, and industrial temperature control due to its simplicity, accuracy, and ease of integration.

This concise format highlights the essential technical details similar to your CP2102 converter format. Let me know if you want it adjusted or expanded!

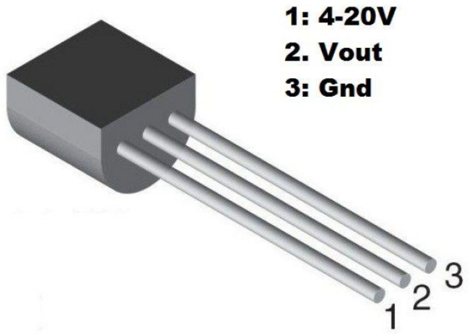

Fig 2.6. LM35 Sensor

8. Rain Sensor Module

The rain sensor module detects the presence and intensity of rain using a separate probe and control board. It provides both digital and analog outputs for flexible interfacing with microcontrollers.

Key features and specifications:

- **Sensor type:** Rain detection module (with sensing pad and control board)
- **Operating voltage:** 5V DC
- **Outputs:** Digital output (DO) for rain/no rain, Analog output (AO) for measuring rain intensity
- **Sensitivity:** Adjustable via onboard potentiometer (trim pot)
- **Current consumption:** 15mA typical for digital output

- **Indicator LEDs:** Power indicator, rain detection indicator
- **Pin configuration:** 4 pins—VCC, GND, DO (digital out), AO (analog out)
- **Board size:** Main module: approximately 3.2cm x 1.4cm; probe: 5cm x 4cm
- **Operating principle:** Conductivity changes between probe tracks detect moisture presence and level
- **Interface compatibility:** Compatible with most microcontrollers (Arduino, Raspberry Pi, STM32, etc.)
- **Comparator IC:** LM393 used for digital processing
- **Other features:** Anti-oxidation probe material, long-term reliability, bolt holes for easy installation

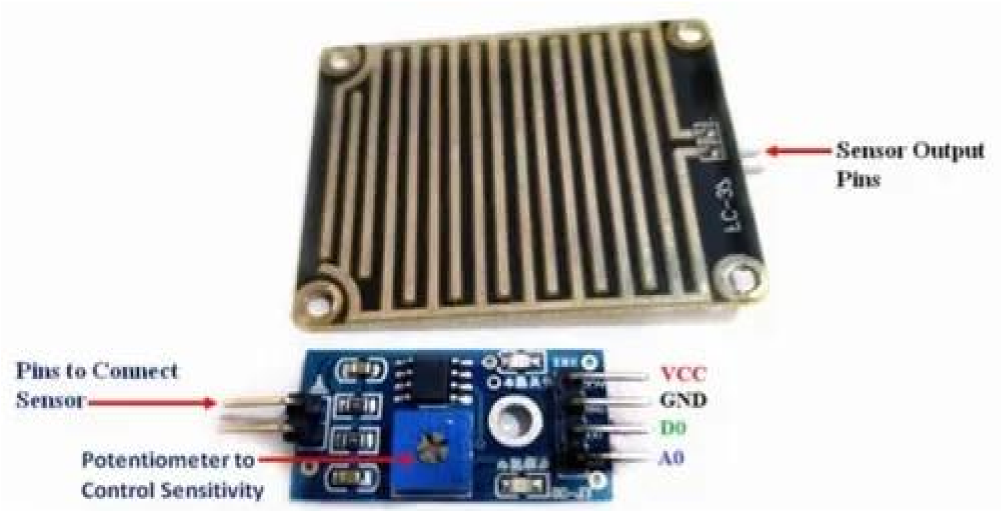

Fig 2.7 Rain Sensor

3. SOFTWARE REQUIREMENTS

STM32CubeIDE is a free, all-in-one integrated development environment (IDE) from STMicroelectronics, designed for developing applications on STM32 microcontrollers and microprocessors. It is part of the STM32Cube software ecosystem and is based on the Eclipse/CDT framework. STM32CubeIDE integrates features from STM32CubeMX for configuration and project management, along with a GNU C/C++ compiler toolchain and GDB debugger.

Key features:

- Integration of STM32CubeMX: Allows for the selection of STM32 microcontrollers, pin assignments, clock and peripheral configuration, and initialization code generation.
- Eclipse-based: Supports Eclipse plug-ins and uses the GNU C/C++ toolchain for ARM and GDB debugger.
- Debugging Tools: Offers advanced debugging features, including CPU core, peripheral register, and memory views, live variable watch, system analysis, real-time tracing, and CPU fault analysis.
- Build and Stack Analyzers: Includes build and stack analyzers that provide information on project status and memory requirements.
- Project Import: Supports importing projects from Atollic TrueSTUDIO and AC6 System Workbench for STM32 (SW4STM32).
- Multi-OS Support: Compatible with Windows, Linux, and MacOS.

STM32CubeIDE helps developers with chip selection, project configuration, code generation, editing, compiling, debugging, and burning. It uses a workspace to manage projects, where each workspace is a folder containing project folders and a “. metadata" folder with project information. For dual-core architecture MCUs, such as STM32H7, STM32L5, and STM32MP1 series, the STM32CubeIDE project has a hierarchical structure with "root" and "child" projects for each core.

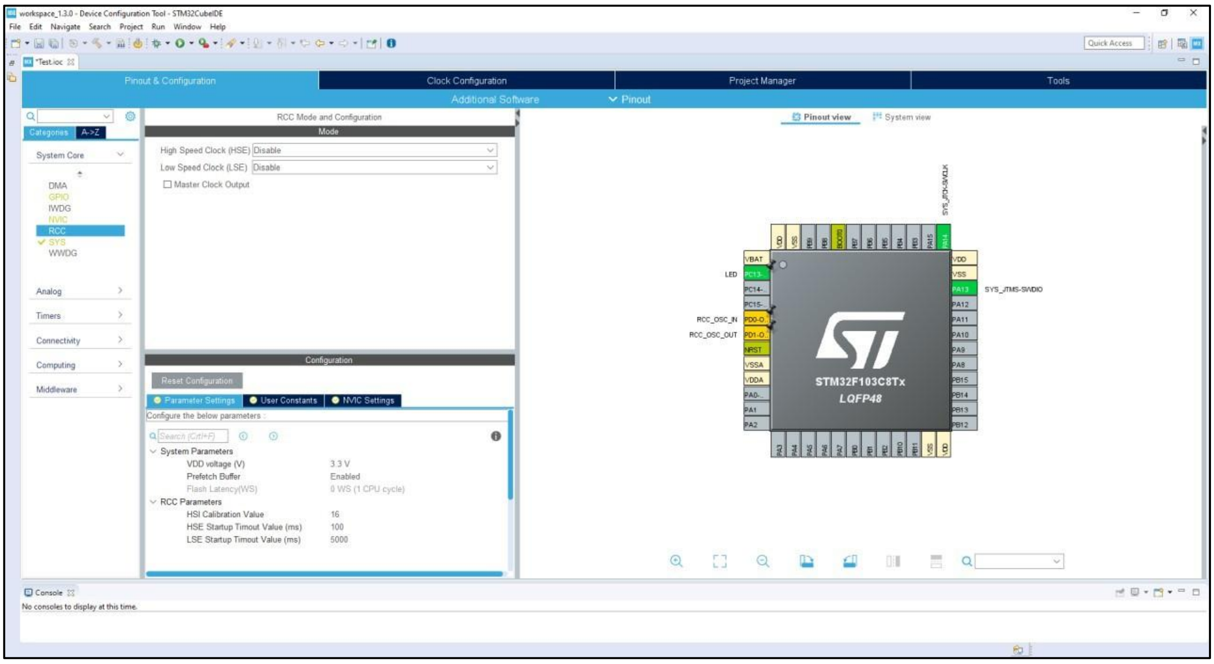

Fig 3.1. STMCubeIDE UI

4. DESIGN AND IMPLEMENTATION

Block Diagram:

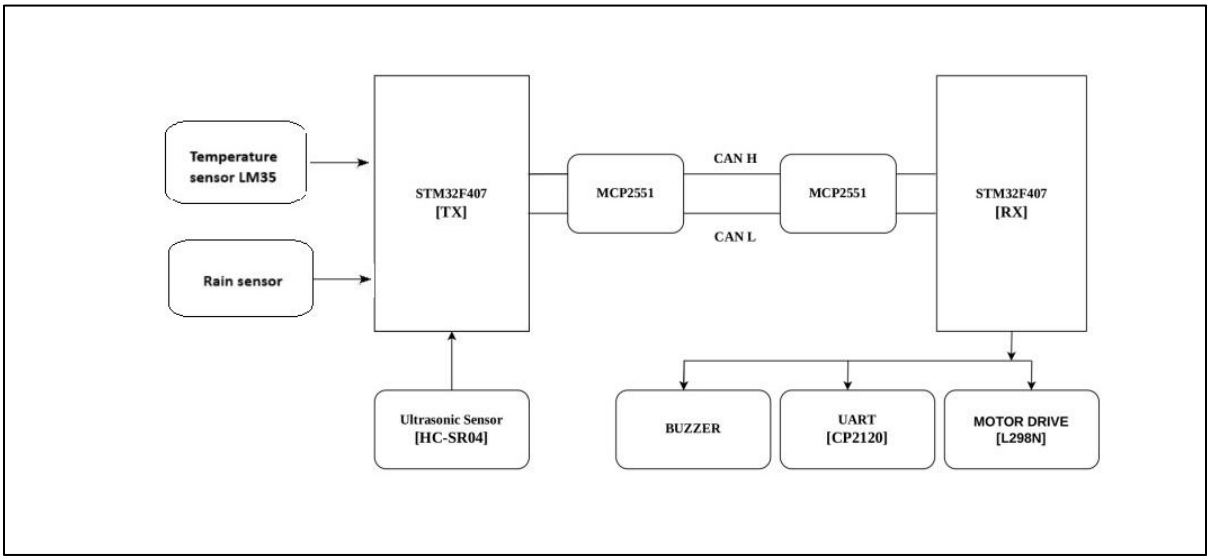

Fig 4.1. Block diagram.

- **STM32F407 [TX]:** This microcontroller acts as the "transmitter" and is responsible for gathering data from the sensor and initiating the braking action. This unit is primarily transmitting data onto the CAN bus.
- **STM32F407 [RX]:** This microcontroller acts as the "receiver" and is responsible for receiving the braking command and controlling the actual braking mechanism. This unit is primarily receiving data from the CAN bus.
- **MCP2551** are transceivers that convert the digital signals from the STM32F407 MCUs into the differential signals required for the CAN bus and vice versa. They act as a bridge between the microcontroller and the physical CAN bus. The two wires that make up the CAN bus. CAN H (High) and CAN L (Low) carry the differential signals that enable communication between the nodes.
- **Ultrasonic Sensor [HC-SR04]:** This sensor measures distance to obstacles by emitting ultrasonic pulses and measuring the time it takes for the echo to return. This data is crucial for the automatic braking system to detect potential collisions.

- **MOTOR DRIVE [L298N]:** This is a motor driver module that can control the speed and direction of a DC motor. In this system, it's used to control the braking mechanism
- **Rain Sensor:** It gives an alert to the user regarding the rain.
- **Temperature Sensor (LM35):** the sensor allows to monitor the real time temperature of the system and alert the user.
- **BUZZER:** This is an audible alarm that can be used to provide warnings or alerts to the user about potential dangers or system status.
- **UART [CP2120]:** This indicates a serial communication interface (Universal Asynchronous Receiver/Transmitter) using a CP2120 converter. This is used for distance for external devices.

System Functionality:

1. The STM32F407 [TX] reads data from the Ultrasonic Sensor [HC-SR04] to determine the distance to any obstacles in front of the vehicle, also the data from the rain sensor and the temperature sensor.
2. The STM32F407 [TX] processes the distance data and determines if there's a risk of collision, the risk of overheating of the system and the rain alert
3. If a potential collision is detected, the STM32F407 [TX] sends a message over the CAN bus via the MCP2551 transceiver. This message contains information about the urgency of the action required.
4. The STM32F407 [RX] receives the CAN message via its MCP2551 transceiver.
5. Based on the received message, the STM32F407 [RX] controls the MOTOR DRIVE [L298N] to apply the brakes.
6. The BUZZER is activated by STM32F407 to warn the user.
7. The UART [CP2120] interface is used for real time temperature, rain status and the distance between a system and an obstacle to an external device.

5. ADVANTAGES

- **Enhanced Safety:** Automatically applies brakes when an obstacle is detected, helping prevent accidents caused by driver error or delayed reaction.
- **Reliable Communication:** Utilizes the CAN protocol to ensure fast and dependable data transfer between sensors and control units, reducing the risk of miscommunication between system components.
- **Real-Time Monitoring:** Integrates ultrasonic, rain, and temperature sensors for immediate detection of road hazards and environmental changes, improving overall situational awareness.
- **Reduced Vehicle Damage:** Early intervention through automatic braking minimizes collision impact, protecting both the vehicle and passengers.
- **Driver Alerts:** The buzzer and sensor-driven alerts increase driver awareness about dangers such as adverse weather or overheating, encouraging timely responses.
- **Scalability and Flexibility:** The modular CAN network allows easy addition of new sensors and smarter control features, adapting to future safety and monitoring requirements.
- **Cost-Effective Implementation:** The system uses widely available, standard hardware components, making it accessible and affordable for integration into various vehicle types.
- **Lower Maintenance and Downtime:** Automated diagnostics and system alerts via CAN network can help identify faults early, leading to quicker maintenance and less downtime.
- **Comfort and Confidence:** By automating safety responses and delivering clear alerts, the system promotes comfortable and confident driving, especially in complex conditions.

6. DISADVANTAGES

- **Complexity in Integration:** Incorporating multiple sensors and CAN-based communication increases system complexity, requiring careful calibration and reliable software to prevent malfunction or false triggers.
- **Potential for False Positives:** Sensors may sometimes detect non-threatening objects or environmental changes (such as sudden rain or temperature fluctuations), leading to unnecessary braking or driver alerts, which could reduce trust in the system.

- **Dependence on Electronic Components:** The system relies heavily on sensors, microcontrollers, and communication networks; a failure in any key component (e.g., sensor malfunction or CAN network disruption) can compromise safety.

- **Maintenance Requirements:** More sophisticated electronics and sensors may require regular maintenance and updates, increasing maintenance costs and potentially causing more frequent downtime.

7. FUTURE SCOPE

- **Integration of Advanced Sensors:** Expand the system by incorporating additional sensors such as LIDAR, radar, or camera modules to enable more accurate detection of obstacles, pedestrians, and lane markings for broader environmental awareness.
- **Artificial Intelligence and Machine Learning:** Implement AI algorithms to analyze sensor data, predict road hazards, and adapt braking responses to different traffic scenarios for smarter decision-making.
- **Connectivity and IoT Features:** Connect the system to cloud platforms for remote monitoring, data logging, and real-time diagnostics. This enables predictive maintenance and fleet management.
- **Vehicle-to-Vehicle (V2V) and Vehicle-to-Infrastructure (V2I) Communication:** Utilize CAN and other communication protocols to allow vehicles to exchange safety information with each other and roadside infrastructure, further reducing collision risks.
- **Enhanced User Interface:** Develop intuitive driver information displays and smartphone apps to provide real-time system alerts, status updates, and driving suggestions.
- **Integration with Autonomous Driving:** Adapt the system to support semi-autonomous or fully autonomous driving features as automotive technology evolves.

8. CONCLUSION

The COVAS system, if implemented can avert lots of accidents and can save invaluable human lives and property. Implementation of such an advanced system can be made compulsory similar to wearing of seat belts so that accidents can be averted to some extent. Our COVAS provides a glimpse into the future of automotive safety, and how much more advanced these individual systems can be for avoiding accidents and protecting vehicle occupants when they are integrated into one system. The future of automotive safety is more than just developing new technology; it is shifting the approach to safety. This approach represents a significant shift from the traditional approach to safety, but it is fundamental to achieving the substantial benefits.

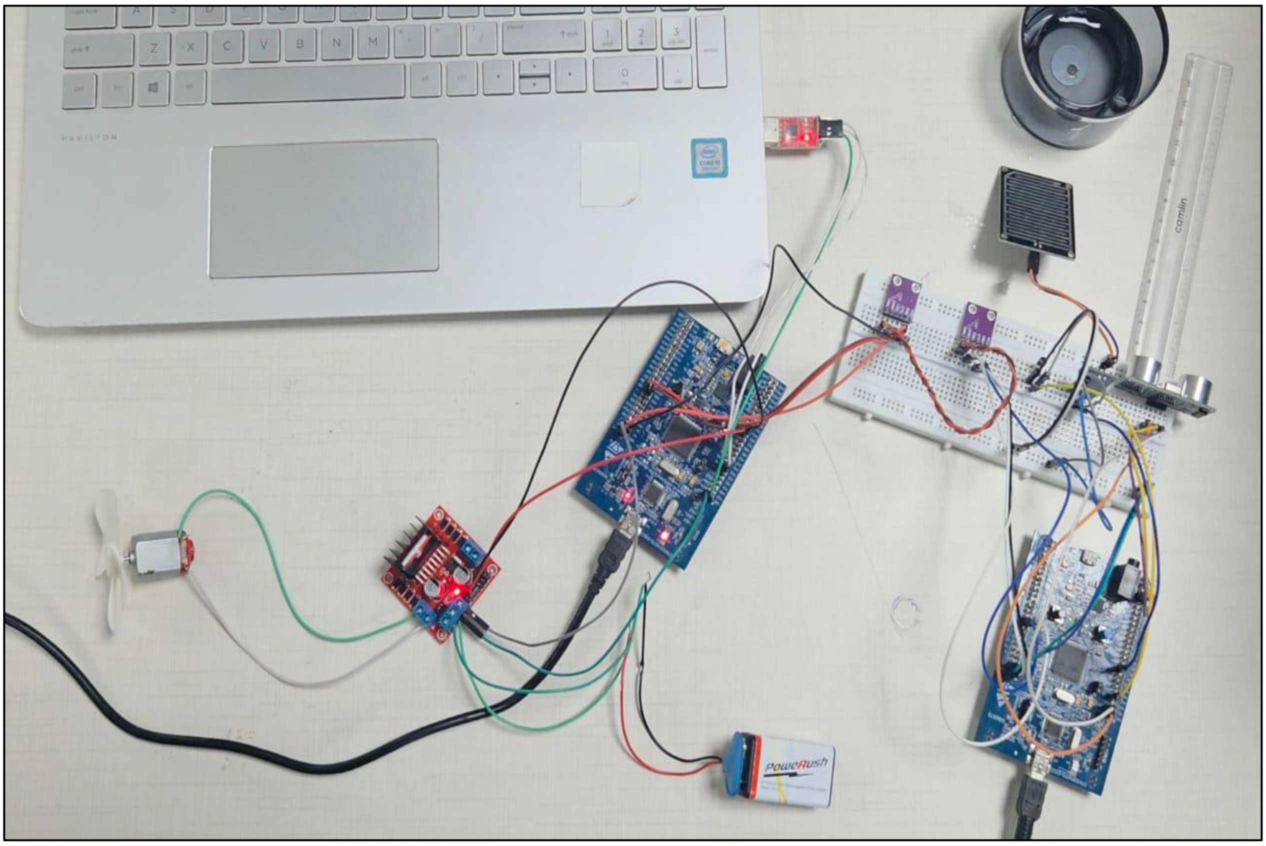

Fig 8.1. COVAS on STM.

9. REFERENCE

1) Ashworth.R, Darkin.D.G, Dickinson.K.W, Hartley.M.G, Wan. W.L, and R.C. Waterfall, APPLICATIONS OF VIDEO IMAGE PROCESSING FOR TRAFFIC CONTROL SYSTEMS. Second International Conference on Road Traffic Control, 14-18 April 1985, London, UK, pp. 119-122.

2) Divyapriya.P, Sasirekha.A, Srilakshmi.B, and Vinodhini. V. EMBEDDED SYSTEM BASED MULTIMODULE PROCESS CONTROL USING CONTROLLER AREA NETWORK International Journal of Communication and Computer Technologies Volume 02 No.5 Issue: 04 April 2014.

3) Manoj Prasanth.R, Raja.S, and Saranya.L, VEHICLE CONTROL USING CAN PROTOCOL FOR IMPLEMENTING THE INTELLIGENT BRAKING SYSTEM (IBS). International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering. Vol. 3, Issue 3, March 2014.

4) Muhammad Ali Mazidi, Rolin D. McKinlay, and Danny Causey: PIC MICROCONTROLLER AND EMBEDDED SYSTEMS, PE India, 01-Sep-2008.

5) Pradhan Suvendu Kedareswar and Venkatasubramanian Krishnamoorthy, AN INTELLIGENT EMBEDDED DIAGNOSTIC SYSTEM ON CAN PROTOCOL TO AVOID REAR-END COLLISION OF VEHICLES. Indian Journal of Science and Technology, Vol 8(19), IPL0124, August 2015.