Zagazig University

Faculty of Engineering, Zagazig, Egypt

A Graduation Project on

Automotive Safety

by

Mahmoud Ayman Fahmy Abd El-Aziz

Hussam Hamdy Mohamed Metwally

Mohamed Abd El-Rahman Abd El-Aziz Habash

Daniel Magdy Abd Allah Hanna

Samuel Mefreh Kamal

Ahmed Mohamed Moheb

Mostafa Samy Fahmy Hamza

Mohamed Abd El-Salam Abd ElGwad

Supervised by

Assoc. Prof. Hanaa Shaker Dr. Rania Ahmed

Dept. of ECE, ZU

July 2022

Acknowledgment

It has been a great opportunity to gain lots of experience in real time projects, followed by the knowledge of how to design analyze real projects. For that we want to thank all the people who made it possible for students like us. Special thanks to the graduation Project Unit for the e orts they didto provide us with all useful information and making the path clear for thestudents to implement all the education periods in real-time project design and analysis. Furthermore, we all the professors and visiting industry for the interesting lectures they presented which had great bene t for all of us. We would like to express our deepest gratitude to our graduation project supervisor Dr. Hanaa Shaker and Dr. Rania Ahmed for your patience and guidance along the semester. Moreover, it is our dutyto thank all the evaluation committee members for their generous discussions and encouragement. At last, we would like to thank all the people who helped, supported, and encouraged us to successfully finish the graduation project whether they were in the university or in the industry.

Table of contents

Table of figures

Abstract

The number of vehicle accidents in the society increases continuously for several reasons which are mostly because of reckless driving and speeding or even driver drowsiness due to long hours of work. Thus, the vehicle driver needs a tool to monitor his performance as a driver and his family. Nearly 1.2 million people die in road crashes every year, on average 3,287 deaths a day. An additional 20-50 million are injured or disabled. More than half of all road traffic deaths occur among young adults ages 15-44. This project provides a low-budget and time free solution to this problem compared to other solutions, it also aims to prioritise the safety of users and system performance. We do this by providing three solutions in our system. First solution is the accident detection system, which is a system to detect an accident when it happens and report the location in an emergency message. Second solution is the sleeping detection system, which detects the driver state, if he is fell asleep or closed his eyes for too long for any reason with strict time constraints and AI algorithms for better detection and operate an alarm or gives a signal to the car system to park. Third solution is the controlling of the motors during Adaptive cruise control and during Parking to help reduce accidents during these two options in cars. To make the system affordable and portable to any kind of cars, we implemented two features for this purpose. first feature is system integration. System integration works independently on car CAN bus of the car so that you can diagnose it as a part of the car and not a separate thing. Second feature is the extended support, firmware is updating and diagnosing remotely, this is available as we provide our flash over the air (FOTA) technology, that makes the system up to date for years without any extra expenses for the user.

Chapter 1
Introduction

1.1 Introduction

In this chapter we'll be explaining why we need this project.

We'll also talk about the Architecture, design, and will give a brief about our main features, and explain how our system compete with other existing solutions.

Cities are getting more and more crowded every day. The overwhelming number of vehicles led to increase in traffic, which led to increase in the number of road accidents. A recent World Health Organization (WHO) report stated that road accidents are ranked as the eighth leading cause of death every year, it even predicted that it may rise to the fifth leading cause of deaths.

There has been a global increase in the annual number of road traffic deaths, even in developed countries with good road safety measures. In third world countries it remains a great danger due to the poorly designed roads and the delay of delivering emergencies.

In Egypt many people are losing their lives every day.

With such a rate of fatality it is particularly important that road safety is improved in developed, and more importantly, in developing countries. With the help of the Internet of Things (IOT) it gives us a promise for the development of intelligent traffic management systems.

Global Navigation Satellite Systems (GPS) are being increasingly used for vehicle positioning and navigation many vehicles that are shipped today have GPS devices that sense the position of the vehicle and send this information to cloud servers. Problems we aim to solve are accidents that are lately reported to or not reported at all if in remote places. How to prevent people from sleeping during driving, and more importantly how to take an action if that happened, and to reduce distractions caused by using mobile apps for maps.

This project provides a low-budget and time free solution to this problem compared to other solutions, it also aims to prioritise the safety of users and system performance. We do this by providing three solutions in our system.

First solution: An accident detection system, which is a system to detect whenever an accident happens and report the location in an emergency message. We used MPU to detect acceleration changes, and angles to detect accidents, and a GPS to get the current location of the accident to be reported, and finally we used GSM to send the location in SMS message to the emergency.

Second solution: A sleeping detection system, which detects whenever the driver is sleeping with strict time constraints and AI algorithms for better detection and sound an alarm or give a signal to the car system to park. We implemented this by training our model on sleeping people's dataset.

Third solution is the controlling of the motors during Adaptive cruise control and during Parking to help to reduce accidents during these two options in cars.

To make the system affordable and portable to any kind of cars we implemented two features for this purpose. The first feature is system integration. System integration works independently on car can bus of the car so that you can diagnose it as a part of the car and not a separate thing. The second feature is the extended support, firmware updates and diagnosing remotely, this is available as we provide our flash over the air (FOTA) technology, that makes the system up to date for years without extra expenses for the user. We implemented our own bootloader to receive new firmware over Wi-Fi, and to burn the code to the non-volatile memory remotely.

Because the system is time critical, we used **Our Own RTOS** as real time operating system to avoid unnecessary delays and to make every task is time constrained so that the whole system is predictable.

1.2 Main features

Our system aims to increase driver safety with least cost and maximum functionality by implementing all of the hardware interfaces to minimize code footprint and by using the best out of our hardware to increase performance.

The system can be embedded in a real car system and communicate with it using CAN protocol.

Existing solutions:

What makes our system competitive with other existing solutions is that even if these systems exist, they mostly come within the car and these categories of cars are too expensive which makes these features not affordable for most cars. However, our system is standalone and can be integrated through our integration feature.

Another competitive feature is the firmware update over the air (FOTA) which makes the system valuable and up to date even a long time after the user buy it. As long as the hardware exists the software and many other features is updated and remotely diagnosed, and support is extended.

1.3 Design

1.3.1 Software design Process

Software design is the process of envisioning and defining software solutions to one or more sets of problems.

Figure 1 Software Design Process

1.3.2 Software design approaches

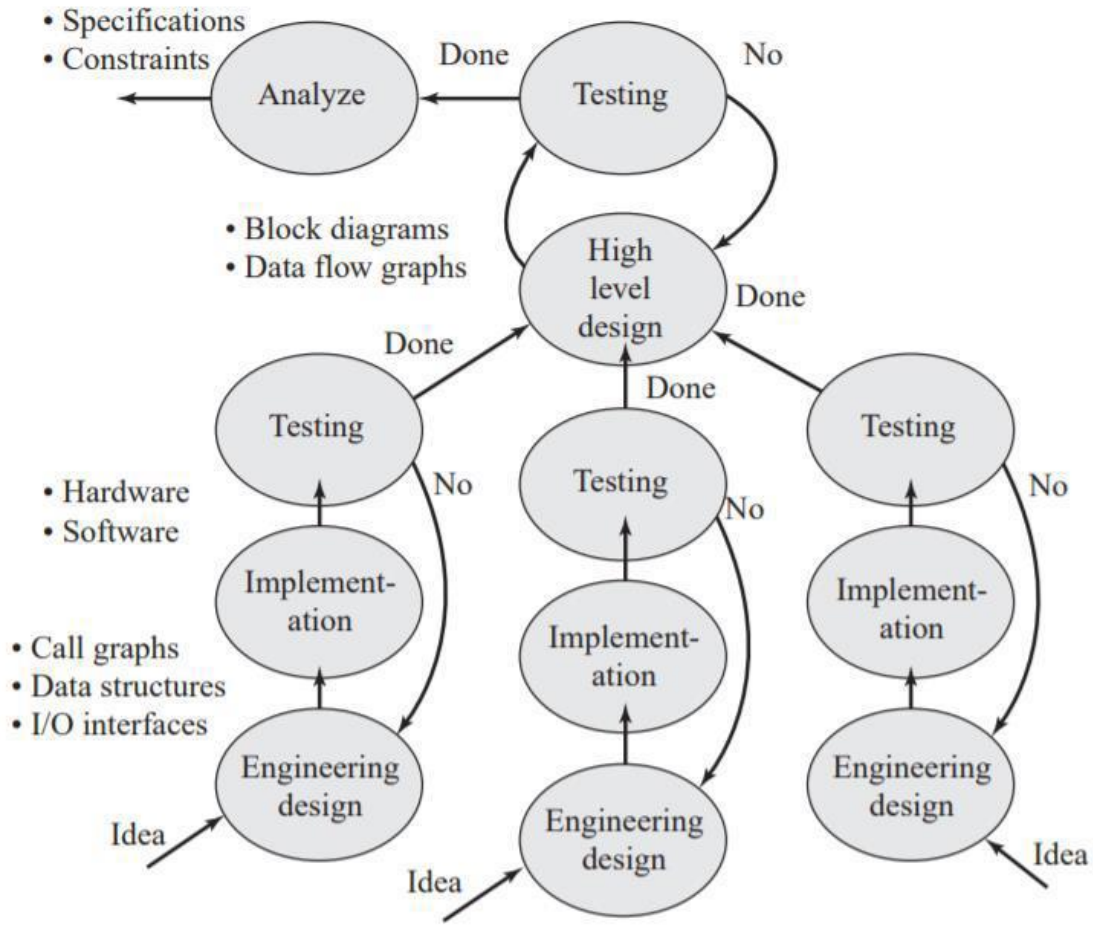


Figure 3 Bottom-up design

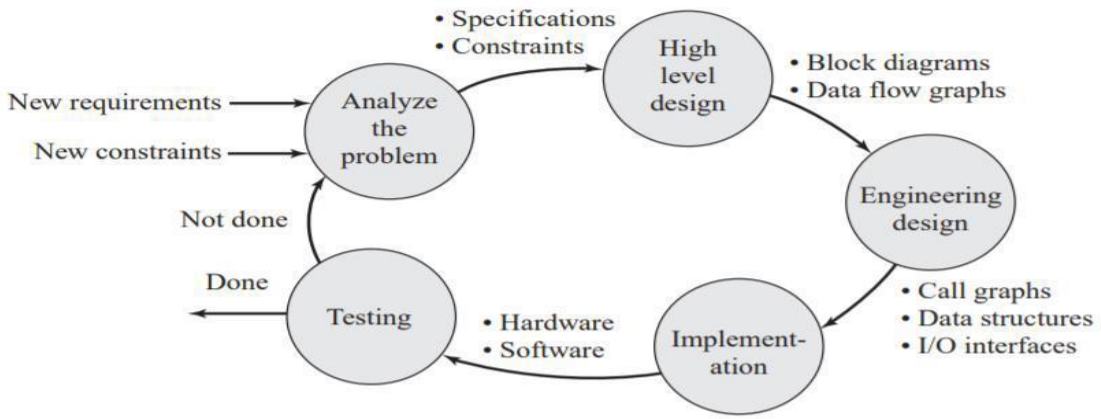


Figure 2 Top-down design

1.3.3 Software Characteristics

- Compatibility **(**backward compatible**)**
- Extendibility (add new code)
- Modularity (drag and drop)
- Maintainability (easy for debug)
- Reusability (Do you can use it in different project?)
- Robustness (work hard)
- Usability (user interface)
- Scalability (increase input data)

1.3.4 Software design Types

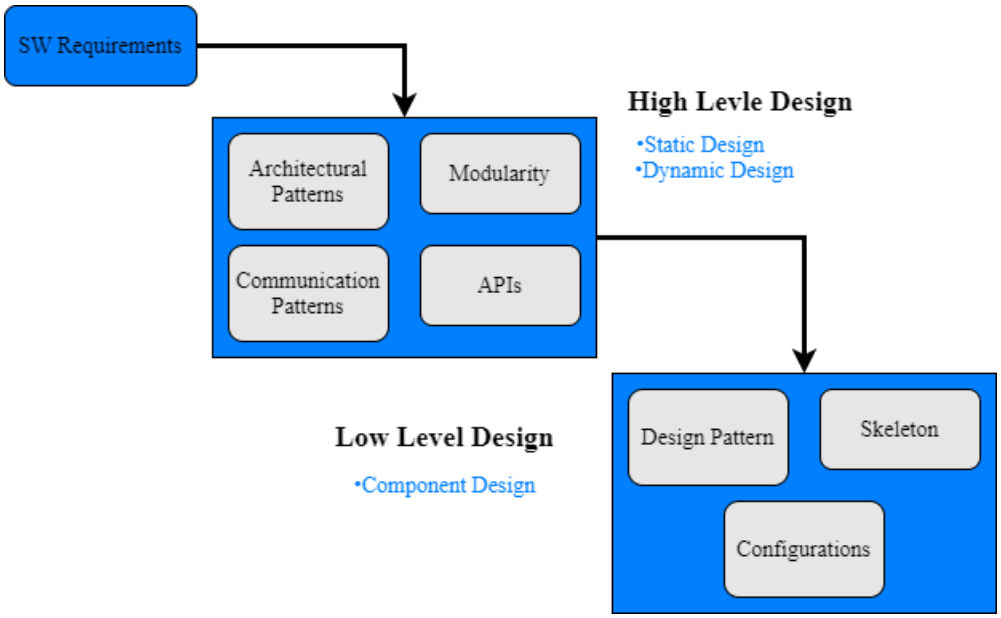


Figure 4 Design Types

1.3.4.1 High Level design

The system engineer analysis customer requirements and extract the software specifications and writes it in a document called SRS.

The architect takes the SRS and put his imagination about the system software from the point view of static design, physical design, and dynamic design.

1- Static design

It represents some important concepts like modularity and architectural patterns.

- Modular programming define how SW system is divided into SW components.
- Architectural patterns: define how to organize modules.
  - Layered architecture
  - Event triggered
  - Micro services

2- Physical design

The role of physical design?

- Design folder structure for the project.
- Design make system for the project.
- Design the output from the project.
- Define files dependencies and implement constraints on them.

3- Dynamic design

- Dynamic Design always answer how the modules talk with each other by implementing communication pattern
- It defines when module APIs' will be triggered.
  - Event or Time triggered
  - Client server
  - Service per Signal oriented

1.3.4.1 Low Level design (Logic design)

Logical design is that which pertains to language constructs such as classes, operators, functions, and so on. For example, whether a particular class should or should not have a copy constructor is a logical design issue.

- Module skeleton: getter API to get data, setter API to set data and main function that run module state machine.
- Design pattern: implement the actual logic of the module.
- Configuration: configure configuration parameters.

The designer writes CDD that explain the logic of the module.
The developer takes the CDD and writes the code that match this logic.

1.3Architecture and Design

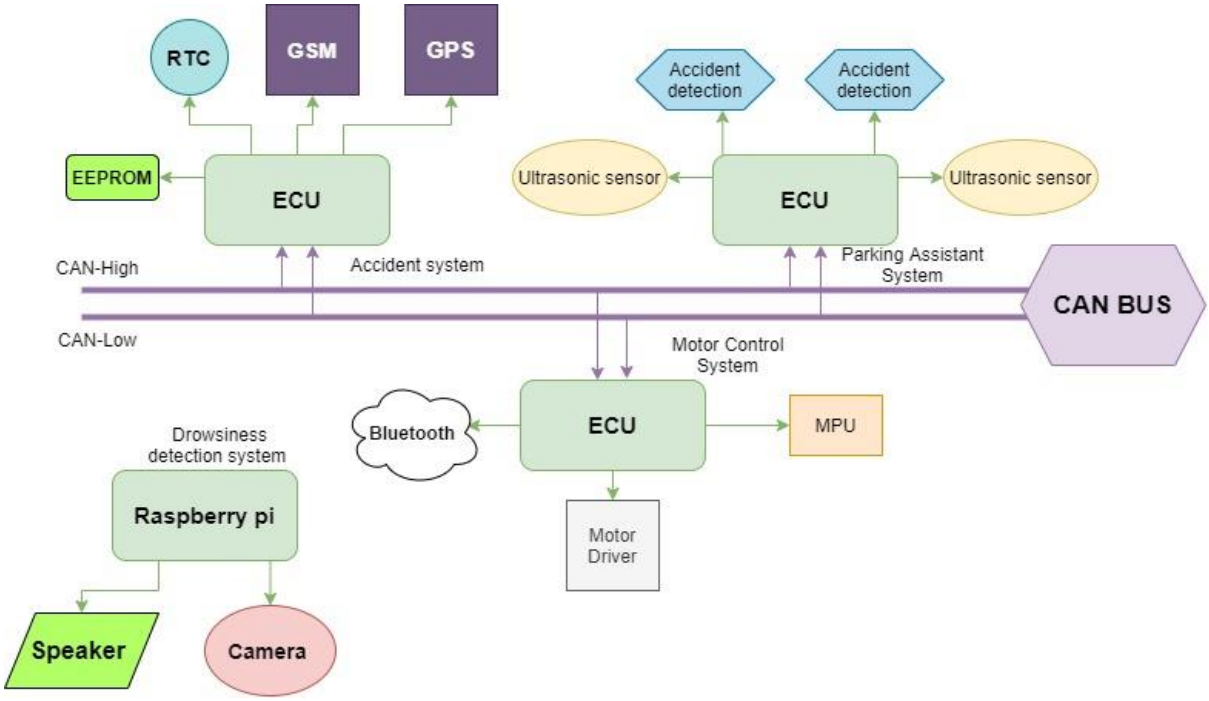


Figure 5 Block Diagram

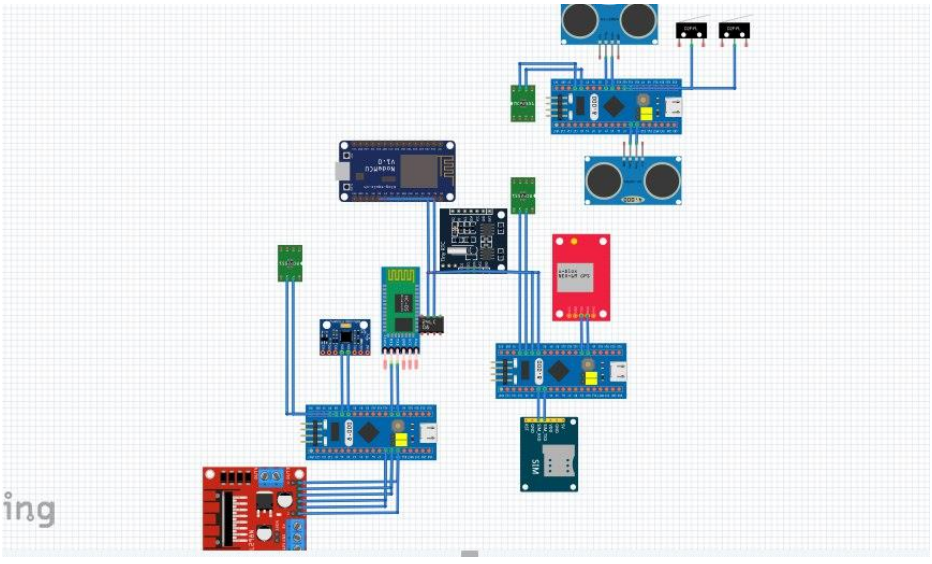


Figure 6 Hardware Design

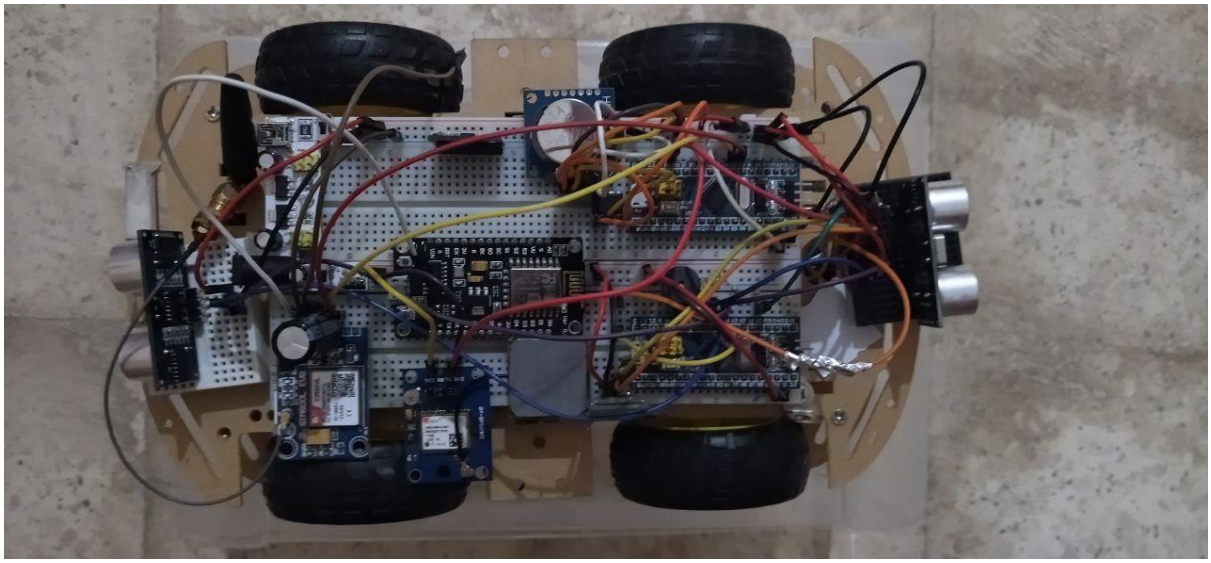


Figure 7 Hardware Connection

Figure 8 Hardware Connection

Chapter 2
Motor Control

2.1 Introduction

Using a prototype for the project to test the system using a motor drive. By measuring the speed and direction of the vehicle. Using a motor driver in order to control the direction and speed of the DC motor.

The idea of working is using L298N chip, which is used to vary PWM according to the desired speed of the motor. This happens through the duty cycle which represent the percentage or the ratio between the ON period to the total period time (ON+OFF period)

2.2 Motors

Without doubt the motors topic or idea is a wide topic, but There are specific types which are common in the embedded systems field such as DC motor, Servo motor and stepper motor.

2.2.1 Motor operation in a Car

Generally, in small light duty DC motors the stator consists of a pair of fixed permanent magnets producing a uniform and stationary magnetic flux inside the motor giving these types of motors their name of “permanent-magnet direct-current” (PMDC) motors.

The motor's armature consists of individual electrical coils connected together in a circular configuration around its metallic body producing a North-Pole then a South-Pole then a North-Pole etc, type of field system configuration.

The current flowing within these rotor coils produces the necessary electromagnetic field. The circular magnetic field produced by the armature windings produces both north and south poles around the armature which are repelled or attracted by the stator’s permanent magnets producing a rotational movement around the motor's central axis. [1]

2.2.2 Motors in our project

In our project we use 4 DC motors to drive the car, one motor on each wheel. controlling the speed and the direction of the vehicle. This happens by varying the speed of on, two, three or the 4 wheels. Speeding up one wheel at the front while keeping the other with exact the same speed, makes the car move in this wheel direction. Or we can speed up the whole 4 wheels for straight line acceleration. So, it is a full control over the car.

2.2.3 Why DC Motor?

There are some advantages taken into consideration when we use a DC motor such as good speed control, high torque, seamless operation and free from harmonics. We will mention every advantage with short details.

Figure 9 DC Motor

- Good Speed Control

DC motors offer highly controllable speed. By changing the armature or field voltage it's possible to achieve wide speed variation and with this level of controllability, DC motors offer the precision required by a wide range of industry applications.

- High torque

A DC motor also offers a high starting torque, which makes it perfect for use in applications that are designed to move heavier loads, such as wiper systems and in industrial automation applications, such as conveyor systems or materials handling equipment. The consistent drive power that DC motors deliver means they're ideal for maintaining a constant torque whilst an application is in use, making them an excellent choice for a geared motor solution.

- Seamless Operation

As DC motors operate with high levels of controllable power across a range of speeds, they offer the benefit of seamless operation. In some industries, it is vital that DC motors can start and stop efficiently to cope with the requirements of the application. If you are looking for a solution that offers rapid acceleration, an option to reverse direction and start/stop efficiency, a DC motor is a good choice.

- Free from Harmonics

In any electric power system, a harmonic is a voltage or current at a multiple of the fundamental frequency of the system, typically produced by the action of non-linear loads such as rectifiers or saturated magnetic devices. Harmonic frequencies in the power grid can be the cause of power quality problems and harmonics in some AC motors can cause torque pulsations, resulting in a decrease in torque. DC motors are free from issues associated with harmonics. [2]

2.2.4 DC motor control

To have complete control over the DC motor we have to control its speed and rotation direction. This can be achieved by combining these two techniques:

- PWM – to control speed
- H-Bridge – to control the rotation direction

➢ PWM – to control speed

The speed of a DC motor can be controlled by changing its input voltage. A common technique to do this is to use PWM (Pulse Width Modulation).

A DC motor consists basically of two parts, the stationary body of the motor called the “Stator” and the inner part which rotates producing the movement called the “Rotor”. For D.C. machines the rotor is commonly termed the “Armature”. [3]

➢ Pulse Width Modulated Waveform

The use of pulse width modulation to control a small motor has the advantage in that the power loss in the switching transistor is small because the transistor is either fully “ON” or fully “OFF”. As a result, the switching transistor has a much-reduced power dissipation giving it a linear type of control which results in better speed stability.

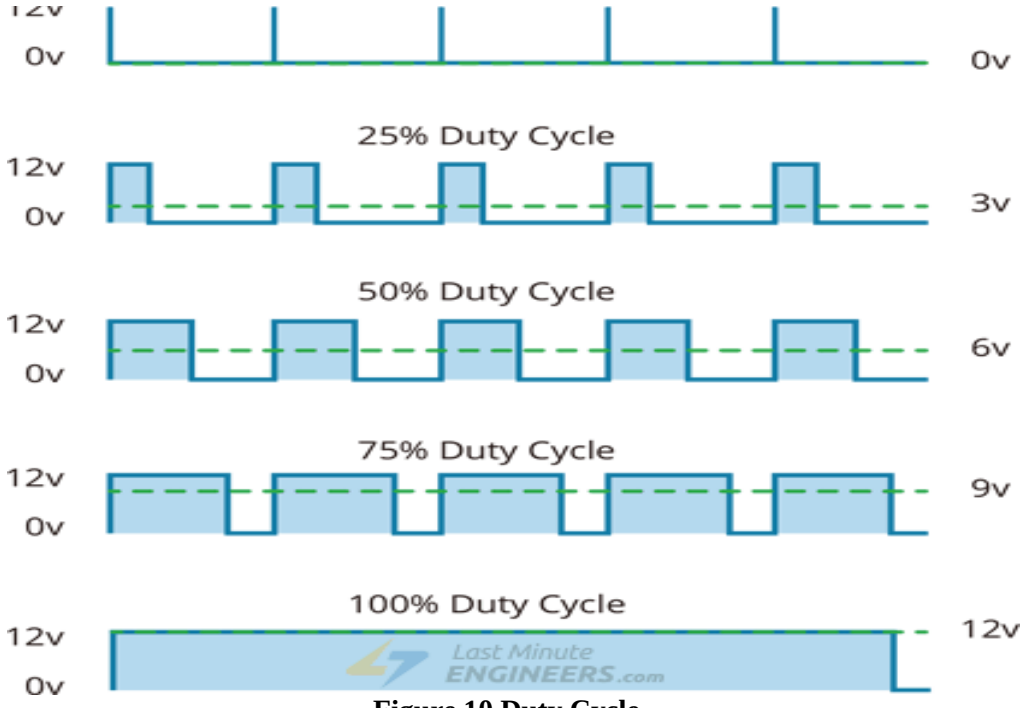


Figure 10 Duty Cycle

Also, the amplitude of the motor voltage remains constant, so the motor is always at full strength. The result is that the motor can be rotated much more slowly without it stalling. By varying the value of the signal, the PWM works accordingly

2.2.5 H-Bridge-to control the rotation direction

The spinning direction of a DC motor can be controlled by changing the polarity of its input voltage. A common technique for doing this is to use an H-bridge.

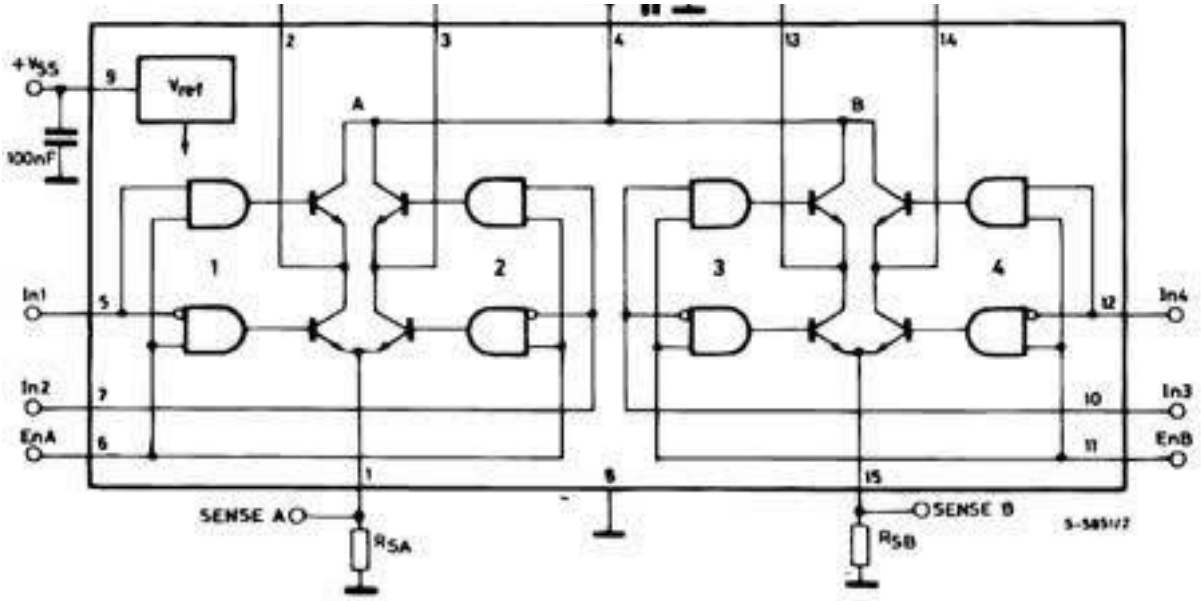


Figure 11 H bridge Circuit

An H-Bridge circuit contains four switching elements, transistors or MOSFET, with the motor at the centre forming an H-like configuration. By activating two particular switches at the same time, we can change the direction of the current flow, thus changing the rotation direction of the motor.

➢ L298 Diver

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. The module has two screw terminal blocks for the motor A and B, and another screw terminal block for the Ground pin, the VCC for motor and a 5V pin which can either be an input or output. This depends on the voltage used at the motors VCC. The module has an onboard 5V regulator which is either enabled or disabled using a jumper. If the motor supply voltage is up to 9V we can enable the 5V regulator and the 5V pin can be used as output, for example for powering our Arduino board. But if the motor voltage is greater than 9V we must disconnect the jumper because those voltages will cause damage to the onboard 5V regulator. In this case the 5V pin will be used as input as we need to connect it to a 5V power supply in order for the IC to work properly.

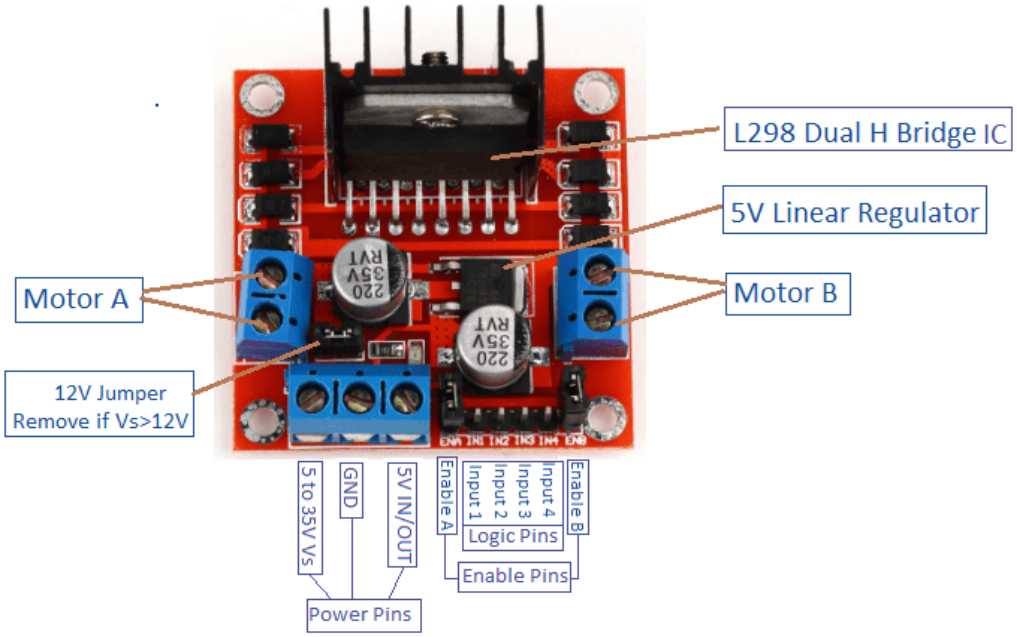


Figure 12 L298 Motor Driver

We can note here that this IC makes a voltage drop of about 2V. So, for example, if we use a 9V power supply, the voltage at motors terminals will be about 10V, which means that we won't be able to get the maximum speed out of our 9V DC motor.

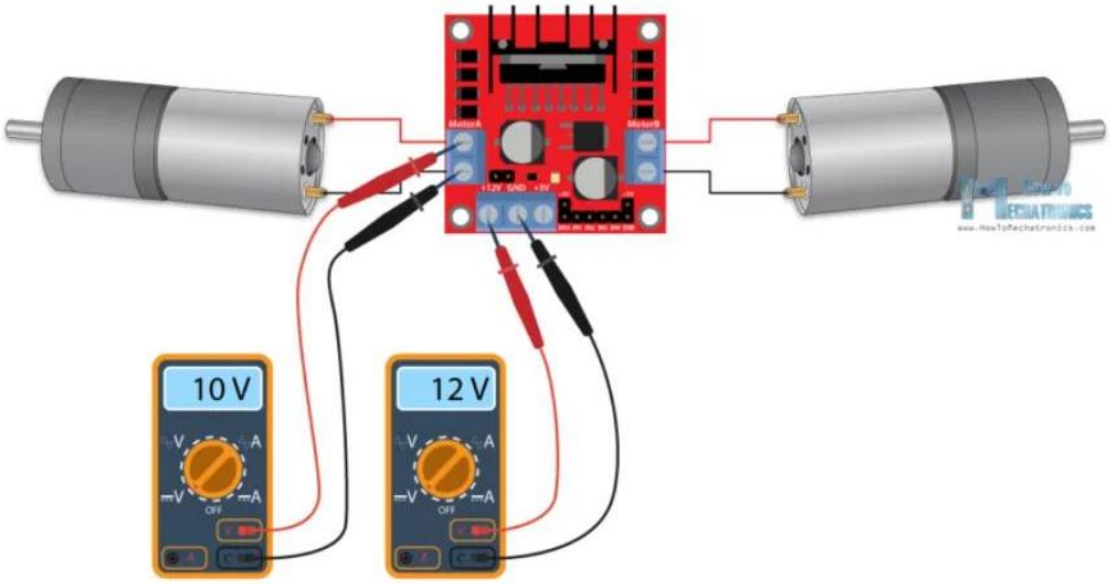


Figure 13 Motors Connections

Next are the logic control inputs. The Enable A and Enable B pins are used for enabling and controlling the speed of the motor. If a jumper is present on this pin, the motor will be enabled and work at maximum speed, and if we remove the jumper, we can connect a PWM input to this pin and in that way control the speed of the motor. If we connect this pin to a Ground the motor will be disabled.

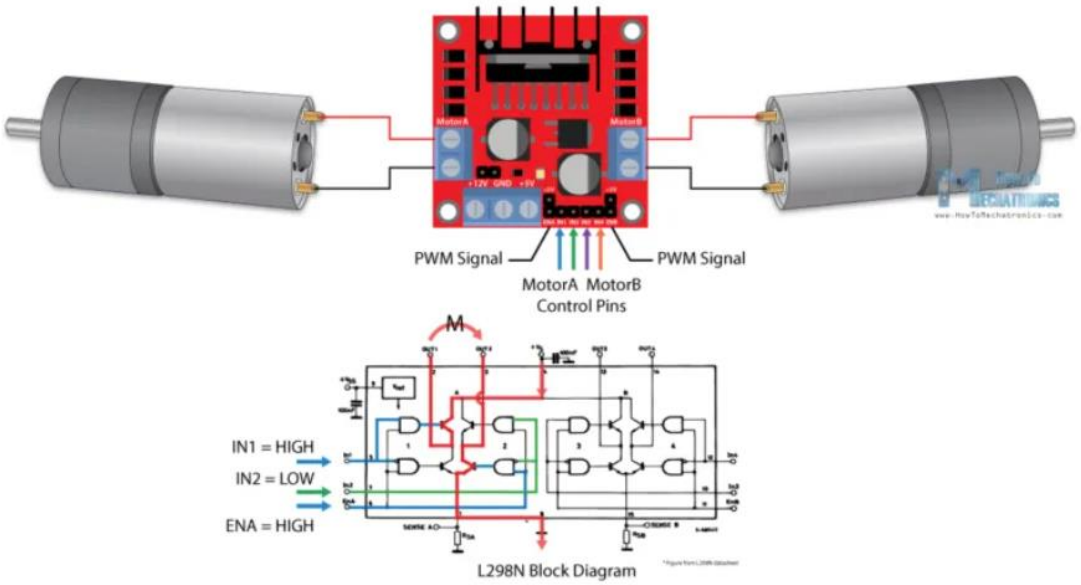


Figure 14 Motors Connections

Next, the Input 1 and Input 2 pins are used for controlling the rotation direction of the motor A, and the inputs 3 and 4 for the motor B. Using these pins, we actually control the switches of the H-Bridge inside the L298N IC. If input 1 is LOW and input 2 is HIGH the motor will move forward, and vice versa, if input 1 is HIGH and input 2 is LOW the motor will move backward. In case both inputs are the same, either LOW or HIGH the motor will stop. The same applies for the inputs 3 and 4 and the motor B. [4]

2.3 Bluetooth

2.3.1 Introduction

Bluetooth technology allows devices to communicate with each other without cables or wires. Bluetooth relies on short-range radio frequency, and any device that incorporates the technology can communicate as long as it is within the required distance

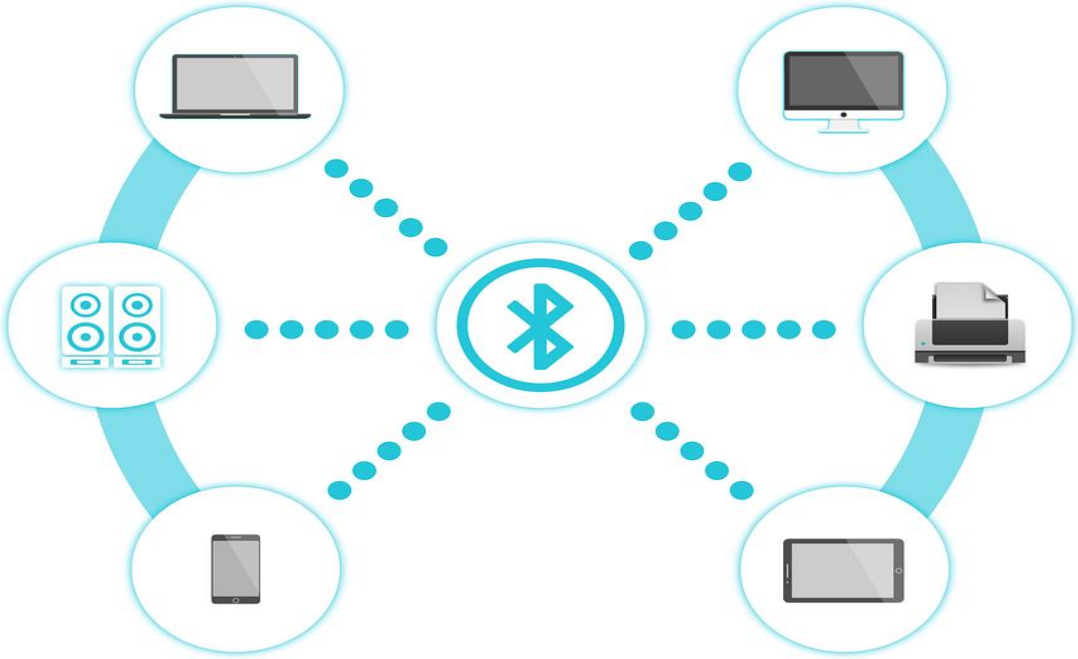


Figure 15 Bluetooth

2.3.2 Advantages of Bluetooth

- it avoids interference from other wireless devices.
- It has lower power consumption.
- It is easily upgradeable.
- It has a better range than infrared communication.
- Bluetooth is used for voice and data transfer.
- Bluetooth devices are available at a very cheap cost.
- No line of sight. Hence, can connect through any obstacles.
- Free to use if the device is installed with Bluetooth.
- The technology is adopted in many products such as headset, in car system, printer, web cam, GPS system, keyboard, and mouse.

2.3.3 Disadvantages of Bluetooth

- it can lose connection in certain conditions.
- It has low bandwidth as compared to Wi-Fi.
- It allows only short-range communications between devices.

2.3.4 Bluetooth module HC-05

- It is used for many applications like wireless headset, game controllers, wireless mouse, wireless keyboard and many more consumer applications.
- It has a range up to <100m which depends upon transmitter and receiver, atmosphere, geographic & urban conditions.
- It is IEEE 802.15.1 standardized protocol, through which one can build wireless Personal Area Network (PAN). It uses frequency-hopping spread spectrum (**FHSS**) radio technology to send data over air.
- It uses serial communication to communicate with devices. It communicates with a microcontroller using a serial port (**USART**). [5]

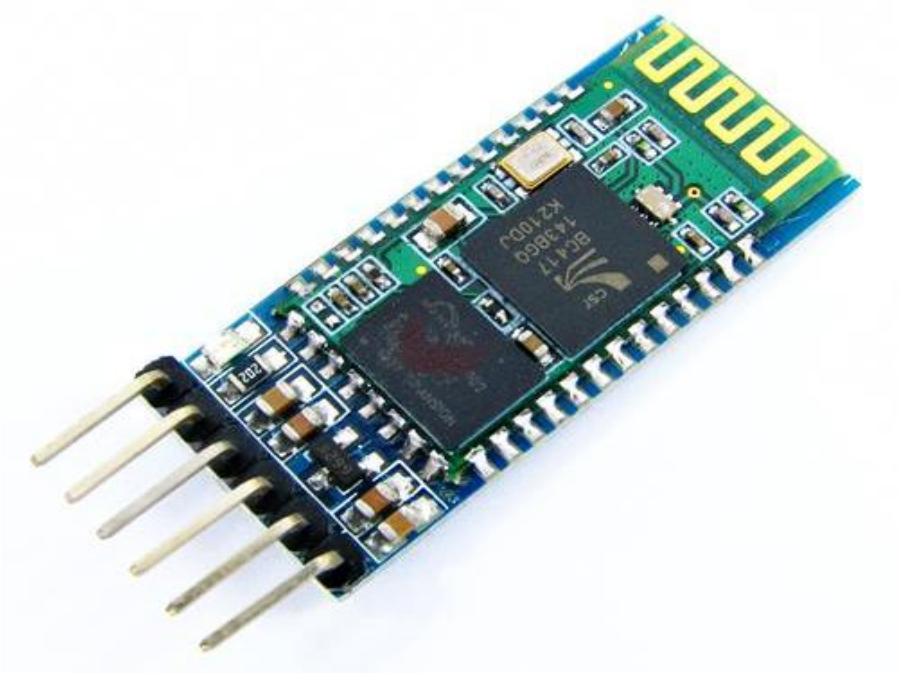


Figure 16 Bluetooth Module

Pin Description

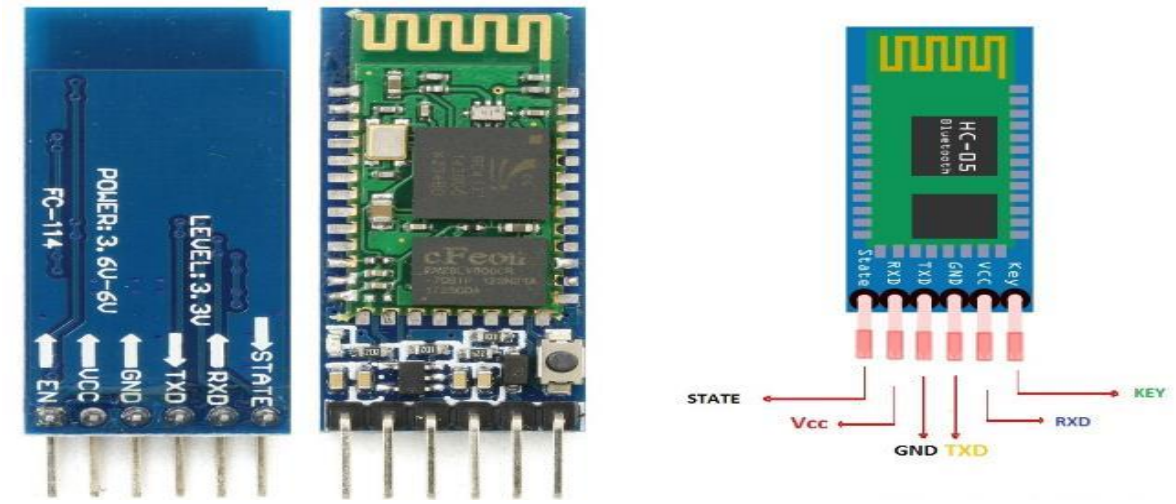


Figure 17 Bluetooth Pin Description

Bluetooth serial modules allow all serial enabled devices to communicate with each other using Bluetooth.

It has 6 pins,

1. **Key/EN**: It is used to bring Bluetooth modules in AT commands mode. If the Key/EN pin is set to high, then this module will work in command mode. Otherwise by default it is in data mode. The default baud rate of HC-05 in command mode is 38400bps and 9600 in data mode.

HC-05 module has two modes,

1. **Data mode**: Exchange of data between devices.
2. **Command mode**: It uses AT commands which are used to change settings of HC-05. To send these commands to the module serial (USART) port is used.

2. **VCC**: Connect 5 V or 3.3 V to this Pin.
3. **GND**: Ground Pin of module.
4. **TXD**: Transmit Serial data (wirelessly received data by Bluetooth module transmitted out serially on TXD pin)
5. **RXD**: Receive data serially (received data will be transmitted wirelessly by Bluetooth module).
6. **State**: It tells whether a module is connected or not.

HC-05 module Information

- HC-05 has a red LED which indicates connection status, whether the Bluetooth is connected or not. Before connecting to the HC-05 module this red LED blinks continuously in a periodic manner. When it gets connected to any other Bluetooth device, its blinking slows down to two seconds.
- This module works on 3.3 V. We can connect 5V supply voltage as well since the module has on board 5 to 3.3 V regulator.
- As HC-05 Bluetooth module has 3.3 V level for RX/TX and microcontroller can detect 3.3 V level, so, no need to shift transmit level of HC-05 module. But we need to shift the transmit voltage level from the microcontroller to the RX of the HC-05 module.

2.3.5 USART

The USART stands for universal synchronous and asynchronous receiver and transmitter. It is a serial communication protocol. This protocol is used for transmitting and receiving the data bit by bit

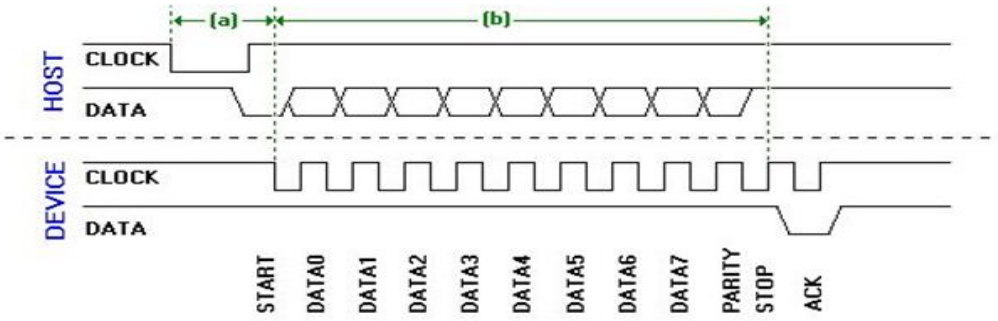


Figure 18 UART DATA

Data transmission

In UART, the mode of transmission is in the form of a packet. The piece that connects the transmitter and receiver includes the creation of serial packets and controls those physical hardware lines. A packet consists of a start bit, data frame, a parity bit, and stop bits.

Figure 19 UART Frame

Start Bit

The UART data transmission line is normally held at a high voltage level when it's not transmitting data. To start the transfer of data, the transmitting UART pulls the transmission line from high to low for one (1) clock cycle. When the receiving UART detects the high to low voltage transition, it begins reading the bits in the data frame at the frequency of the baud rate.

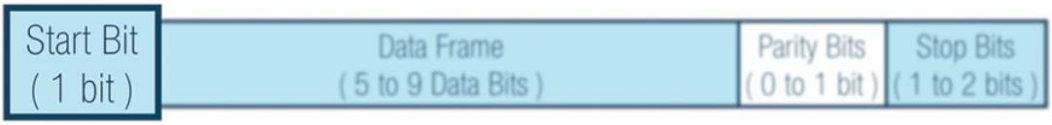


Figure 20 UART Start Bit

Data Frame

The data frame contains the actual data being transferred. It can be five (5) bits up to eight (8) bits long if a parity bit is used. If no parity bit is used, the data frame can be nine (9) bits long. In most cases, the data is sent with the least significant bit first.

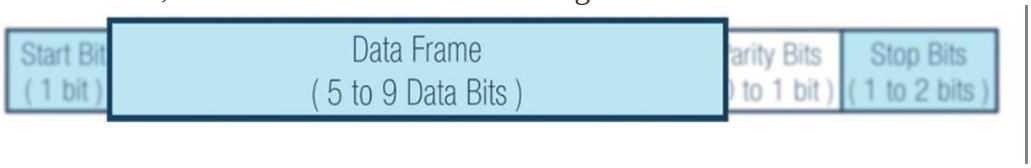


Figure 21 UART Data bits

Parity

Parity describes the evenness or oddness of a number. The parity bit is a way for the receiving UART to tell if any data has changed during transmission. Bits can be changed by electromagnetic radiation, mismatched baud rates, or long-distance data transfers.
After the receiving UART reads the data frame, it counts the number of bits with a value of 1 and checks if the total is an even or odd number. If the parity bit is a 0 (even parity), the 1 or logic-high bit in the data frame should total to an even number. If the parity bit is a 1 (odd parity), the 1 bit or logic highs in the data frame should total to an odd number.

When the parity bit matches the data, the UART knows that the transmission was free of errors. But if the parity bit is a 0, and the total is odd, or the parity bit is a 1, and the total is even, the UART knows that bits in the data frame have changed.

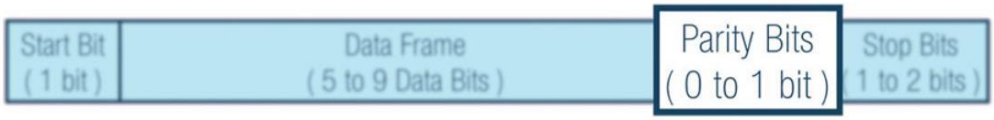


Figure 22 UART Parity Bits

Stop Bits

To signal the end of the data packet, the sending UART drives the data transmission line from a low voltage to a high voltage for one (1) to two (2) bit(s) duration.

Steps of UART transmission

First: The transmitting UART receives data in parallel from the data bus.

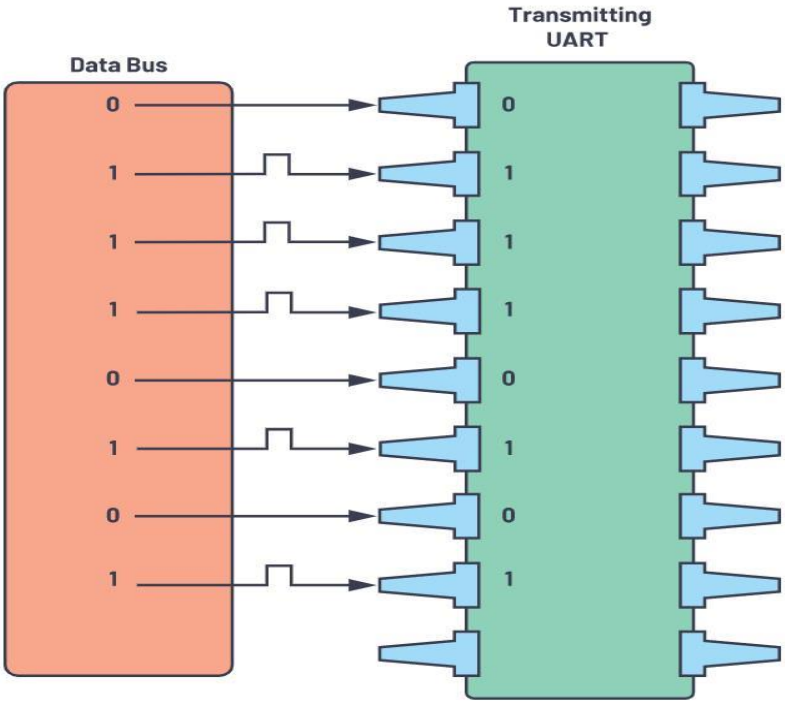


Figure 23 UART Internal to Tx Buffer

Second: The transmitting UART adds the start bit, parity bit, and the stop bit(s) to the data frame.

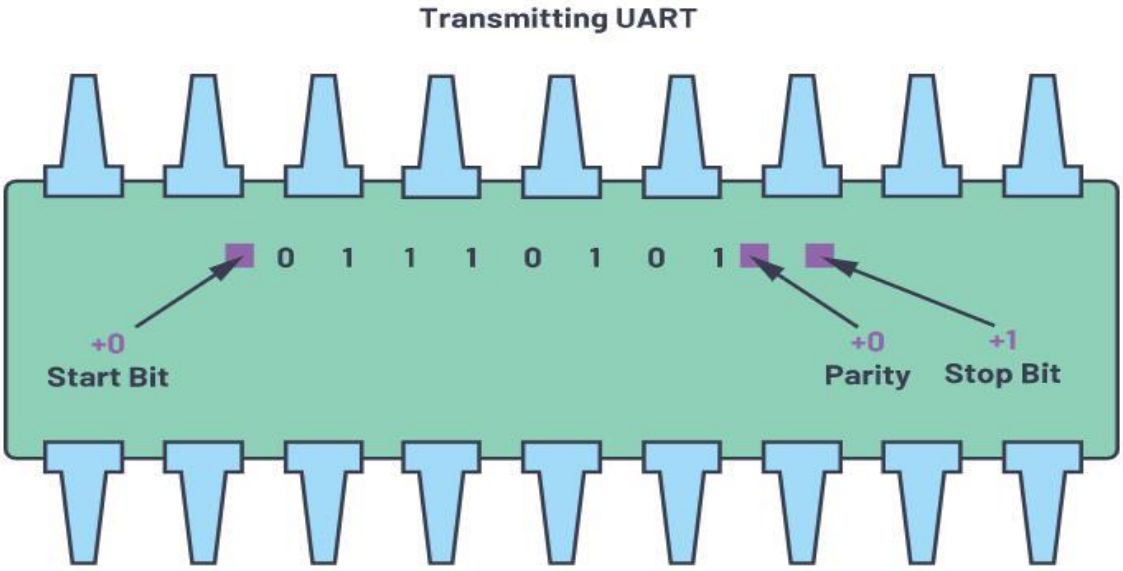


Figure 24 UART Buffer

Third: The entire packet is sent serially starting from start bit to stop bit from the transmitting UART to the receiving UART. The receiving UART samples the data line at the preconfigured baud rate.

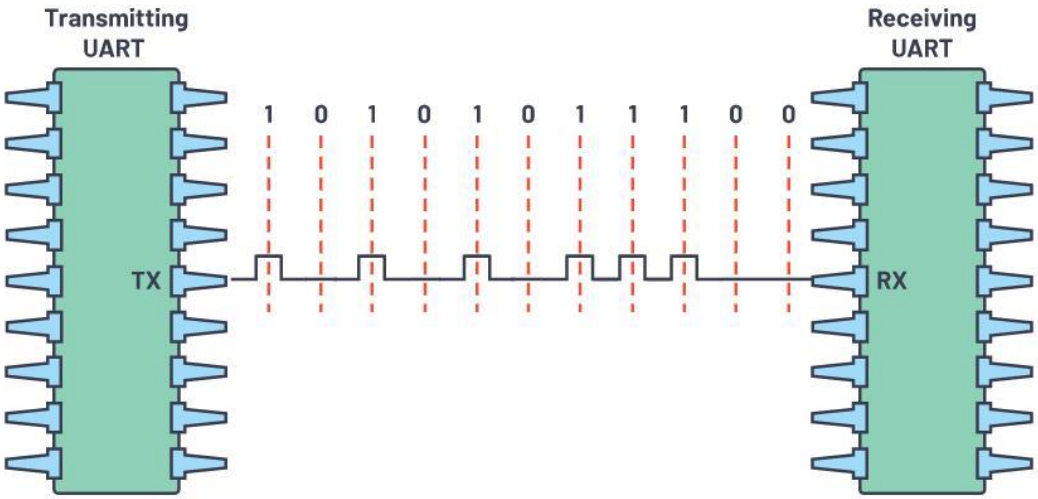


Figure 25 UART Transmission

Fourth: The receiving UART discards the start bit, parity bit, and stop bit from the data frame.

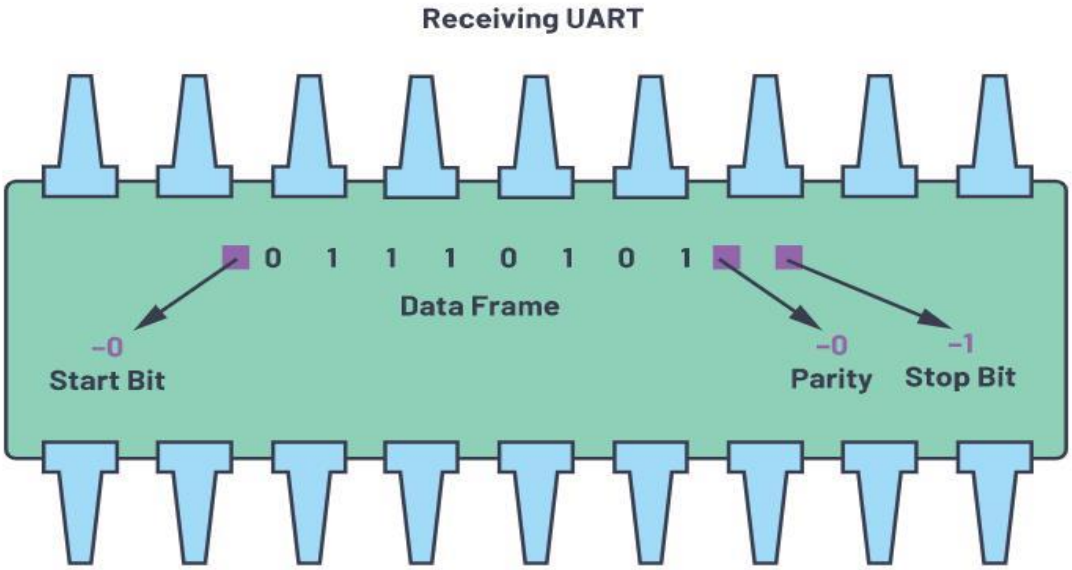


Figure 26 UART Reception

Fifth: The receiving UART converts the serial data back into parallel and transfers it to the data bus on the receiving end.

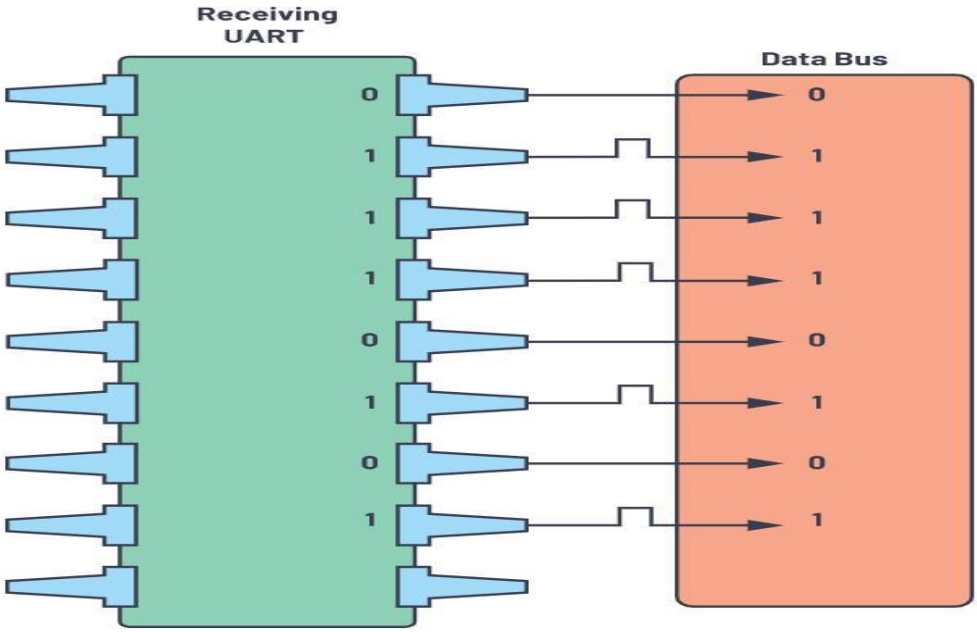


Figure 27 UART to Buffer

Rules of UART

As mentioned earlier, there is no clock signal in UART, and the transmitter and receiver must agree on some rules of serial communication for error free transfer of data. The rules include:

- Synchronization Bits (Start and Stop bits)
- Parity Bit
- Data Bits
- Baud Rate

Baud Rate: The speed at which the data is transmitted is mentioned using Baud Rate. Both the transmitting UART and Receiving UART must agree on the Baud Rate for a successful data transmission.

- Baud Rate is measured in bits per second. Some of the standard baud rates are 4800 bps, 9600 bps, 19200 bps, 115200 bps etc. Out of these 9600-bps baud rates is the most commonly used one. [6]

2.3.6 Security Bluetooth network

Bluetooth is extremely secure in that it employs several layers of data encryption and user authentication measures. Bluetooth devices use a combination of the Personal Identification Number (PIN) and a Bluetooth address to identify other Bluetooth devices. Data encryption (i.e., 128-bit) can be used to further enhance the degree of Bluetooth security.

The transmission scheme (FHSS) provides another level of security in itself. Instead of transmitting over one frequency within the 2.4 GHz band, Bluetooth radios use a fast frequency-hopping spread spectrum (FHSS) technique, allowing only synchronized receivers to access the transmitted data.

2.3.7 Frequency hopping spread spectrum FHSS

Bluetooth utilizes frequency-hopping spread spectrum technology to avoid interference problems. The ISM 2.4 GHz band is 2400 to 2483.5 MHz, and Bluetooth uses 79 radio frequency channels in this band, starting at 2402 MHz and continuing every 1MHz It is these frequency channels that Bluetooth technology is "hopping" over. The signal switches carrier
channels rapidly, at a rate of 1600 hops per second, over a determined pattern of channels. There are six defined types of hopping sequences.
Information is conveyed by modulating the carrier channel frequency, using one of several modulation schemes. Gaussian frequency-shift keying (GFSK) modulation was initially the only type available, but recently other varieties have been enabled. GFSK is simply a type of frequency-shift keying (FSK), which is a modulation scheme where the bits of the transferred information correspond to discrete frequency changes in the carrier signal. The carrier signal is whichever band the device happens to be using at that moment (before it hops to another), and the modified signal is broadcast out.

Because Bluetooth uses this frequency-hopping scheme, it is very unlikely that there will be much interference from other devices, be they Bluetooth or not. Given that the hopping patterns are pseudo-random, the chances that another Bluetooth device would use the same pattern and disrupt a large amount of dataflow is very low. Additionally, other devices that simply broadcast at a fixed frequency can only have a minimal impact on the data transferred using Bluetooth. [7]

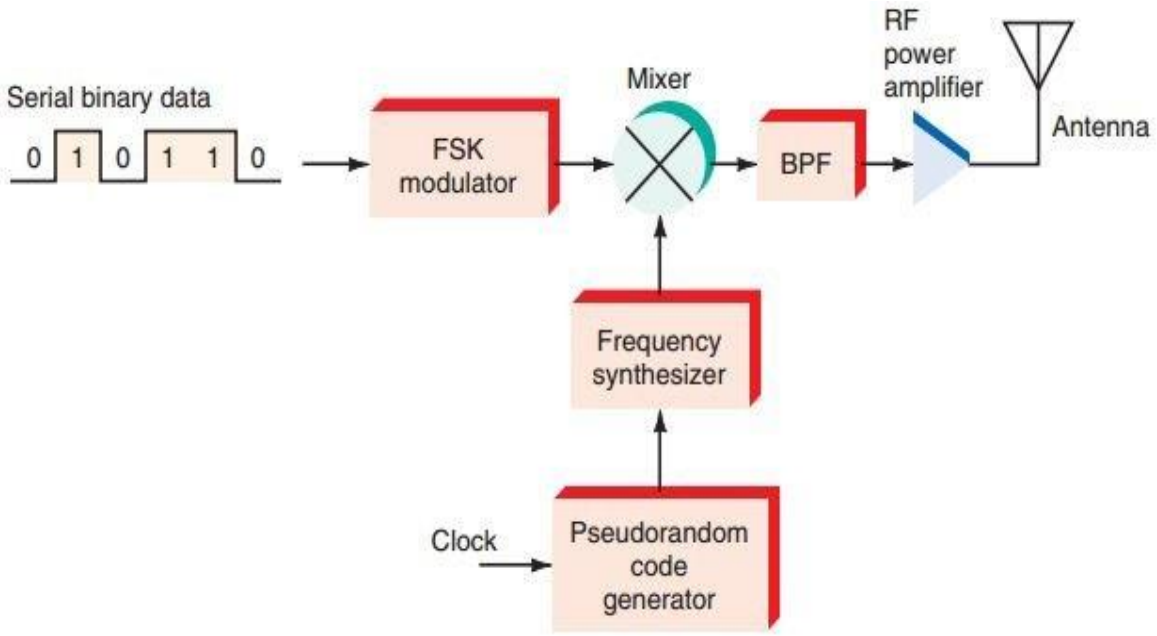


Figure 28 FHSS

2.3.8 Bluetooth communication between Devices

E.g., Send data from the Smartphone terminal to the HC-05 Bluetooth module and see this data on the PC serial terminal and vice versa.

To communicate with the HC-05 Bluetooth module, the smartphone requires a Bluetooth terminal application for transmitting and receiving data. You can find Bluetooth terminal applications for android and windows in respective applications stores.

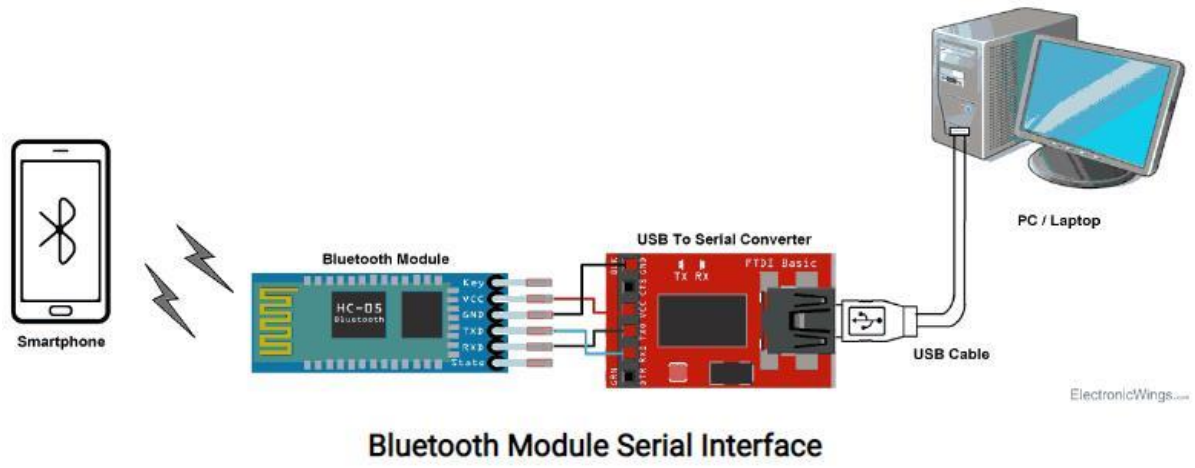


Figure 29 Bluetooth serial interface

So, when we want to communicate through a smartphone with the HC-05 Bluetooth module, connect this HC-05 module to the PC via serial to USB converter.

Before establishing communication between two Bluetooth devices, 1st we need to pair the HC-05 module to a smartphone for communication.

➢ Pair HC-05 and smartphone:

1. Search for a new Bluetooth device from your phone. You will find Bluetooth devices with "HC-05" name.
2. Click on connect/pair device option; default pin for HC-05 is 1234 or 0000.

After pairing two Bluetooth devices, open terminal software (e.g., Teraterm, Realterm etc.) on the PC, and select the port where we have connected USB to the serial module. Also select default baud rate of 9600 bps.
In the smartphone, open the Bluetooth terminal application and connect to paired device HC-05.
It is simple to communicate, we just have to type in the Bluetooth terminal application of our smartphone. Characters will get sent wirelessly to Bluetooth module HC-05. HC-05 will automatically transmit it serially to the PC, which will appear on the terminal. Same way we can send data from PC to smartphone.

➢ Command Mode

- When we want to change settings of the HC-05 Bluetooth module like change password for connection, baud rate, Bluetooth device's name etc.
- To do this, HC-05 has AT commands.
- To use the HC-05 Bluetooth module in AT command mode, connect the "Key" pin to High (VCC).
- Default Baud rate of HC-05 in command mode is 38400bps.
- Following are some AT commands generally used to change the setting of the Bluetooth module.
- To send these commands, we have to connect the HC-05 Bluetooth module to the PC via serial to USB converter and transmit these commands through serial terminal of the PC.

Command	Description	Response
AT	Checking communication	OK
AT+PSWD=XXXX	Set Password e.g. AT+PSWD=4567	OK
AT+NAME=XXXX	Set Bluetooth Device Name e.g. AT+NAME=MyHC-05	OK
AT+UART=Baud rate, stop bit, parity bit	Change Baud rate e.g. AT+UART=9600,1,0	OK
AT+VERSION?	Respond version no. of Bluetooth module	+Version: XX OK e.g. +Version: 2.0 20130107 OK
AT+ORGL	Send detail of setting done by manufacturer	Parameters: device type, module mode, serial parameter, passkey,etc.

Chapter 3
Accident Monitoring & Parking Assistant

3.1 Introduction

Accidents take many forms and shapes, there is huge variety of how a user can have an accident. Some accidents happen during driving, or maybe during parking.

To avoid accidents of any form we have a system only to help avoid these accidents.

This system is composed of many phases that are interdependent, and every system will be discussed very specifically in chapter 4,5, and 6.

In this chapter we will have an overall idea of these phases and how they interact with each other.

The detecting phase. This phase detects whether the accident happened or not yet. And for this phase we have two ultrasonic sensors. One on the rear and the other on the front. These two detectors purpose is to try to avoid the accident by measuring the distance in front of the car and at the rear. Also, we have two more sensors, collision sensors. The collision sensors job comes in when the accident took place.

The alerting phase. This phase takes place when one of the sensors or the collision sensor detect either a near collision or a collision already took place. There is a broadcast through the intercommunication system that there is an accident so that every system behaves accordingly.

Taking an action phase. In this phase, all the systems received the broadcast. So that they got to take action accordingly.

If there is a near collision at the rear, it sounds an alarm. If there is a near collision at the front or the collision already happened, it takes an action (We'll get to it in the next chapters)

Parking Assistant

3.2 Introduction

The parking assistant purpose is to help the user park efficiently and safely. Parking is a very disturbing thing for many people, that's why this system is important. Parking assistant will help the user avoid any obstacle at the rear.

It consists of Microcontroller STM32f103c8T6, two ultrasonic one on the rear and another in the front and two Accident sensor. If The car is in cruise control mode, if the Front ultrasonic read a distance below a certain distance it starts to make a sound using the Buzzer, then sends through intercommunication system to the motor control system to stop the car. If the car is in not cruise control mode, if the front or rear ultrasonic read a distance below a certain distance it starts to make a sound using the Buzzer .The accident sensor if it reads that there is an accident it broadcast through intercommunication system that there is an accident so, the motor control system stop the motor, the accident alert system starts to send the message to save a life of the car driver, the accident recording system record the co-ordinates of the accident and time in non-volatile memory (EEPROM) and the IOT system display on the application that there is an accident, show the co-ordinates as a message and the location on the map.

3.3 Ultrasonic

The HC-SR04 Ultrasonic Distance Sensor consists of two ultrasonic transducers. The device acts as a transmitter that converts the electrical signal into 40 kHz ultrasonic sound pulses. The receiver listens to the transmitted pulses. If it receives it, it produces an output pulse whose width can be used to determine the distance travelled by the pulse.

The HC-SR04 ultrasonic distance sensor. This economical sensor provides 2cm to 400cm of non-contact measurement functionality with a ranging accuracy that can reach up to 3mm. Each HC-SR04 module includes an ultrasonic transmitter, a receiver, and a control circuit. [8]

3.2.1 Pin Assignment

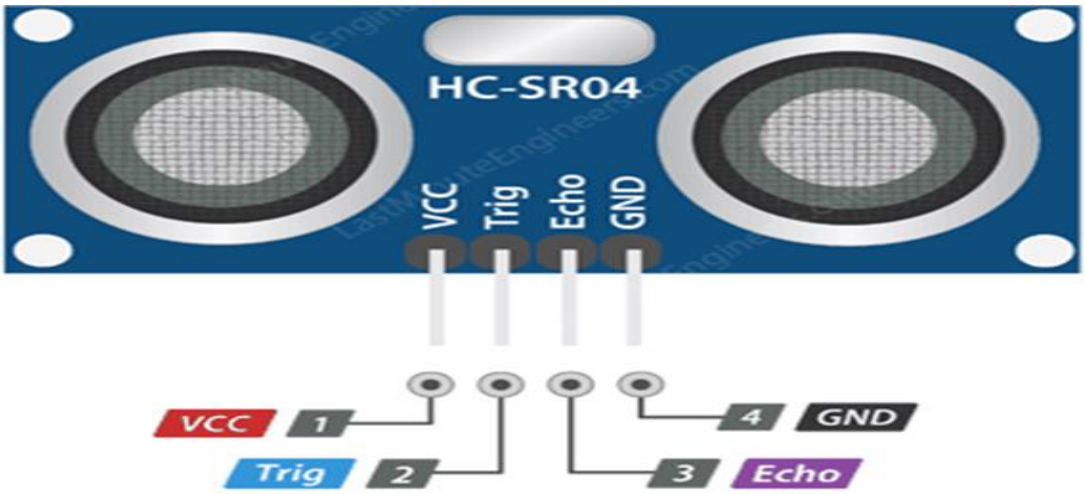


Figure 30 HC-SR04 Pin Assignment

There are only four pins that you need to worry about on the HC-SR04: VCC (Power), Trig (Trigger), Echo (Receive), and GND (Ground). You will find this sensor very easy to set up

VCC is the power supply for HC-SR04 Ultrasonic distance sensor which we connect the 5V pin on the Arduino.

Trig (Trigger) pin is used to trigger the ultrasonic sound pulses.

Echo pin produces a pulse when the reflected signal is received. The length of the pulse is proportional to the time it took for the transmitted signal to be detected.

GND should be connected to the ground of Microcontroller. [9]

3.2.2 HC-SR04 Sensor Features

- Operating voltage: +5V
- Theoretical Measuring Distance: 2cm to 450cm
- Practical Measuring Distance: 2cm to 80cm
- Accuracy: 3mm
- Measuring angle covered: <15°
- Operating Current: <15mA
- Operating Frequency: 40Hz [9]

3.2.3 Ultrasonic Sensor Used

Ultrasonic sensors are used primarily as proximity sensors. They can be found in automobile self-parking technology and anti-collision safety

systems. Ultrasonic sensors are also used in robotic obstacle detection systems, as well as manufacturing technology. In comparison to infrared (IR) sensors in proximity sensing applications, ultrasonic sensors are not as susceptible to interference of smoke, gas, and other airborne particles (though the physical components are still affected by variables such as heat).

An ultrasonic sensor is a sensor which measures the distance of the respective object by sending the sound wave of a specific frequency. This sound wave is reflected after the collision with the respective object and this wave is received by the ultrasonic receiver. Distance is measured by calculating sending and receiving time of this sound wave. Here this post gives information about the pros and cons of ultrasonic to better understand this topic. [10]

3.2.4 Advantages of Ultrasonic sensor

- This sensor could have easily sensed the nature, shape and orientation of that specific object which is within the area of these sensors so sensing capability to sense all the material types.
- These sensors have greater accuracy than another method for measuring the thickness and depth of the parallel surface.
- This sensor easily interfaces with a microcontroller or any type of controller.
- This sensor has high frequency, high sensitivity and high penetrating power; therefore, it can easily detect the external or deep object.
- It can work in any adverse conditions.
- Their sensor is easy to use, not dangerous during operation for nearby object person, equipment, or material.
- This sensor is not affected due to atmospheric dust, rain, and snow.
- It provides a good reading in sensing large-sized objects with hard surfaces.
- It has a higher sensing distance compared to inductive or capacitive proximity sensor types.

3.2.6 Disadvantages of Ultrasonic sensor:

- It is very sensitive to variation in the temperature.
- It has more difficulties in reading reflections from soft, curved and thin as well as a small object.
- These sensors have a base detecting distance.
- It required careful attention for an experienced technician.
- Change in nature for example temperature, airborne particles, weight, air turbulence, influence ultrasonic reaction.
- In this sensor, the main focuses of low thickness similar to froth and fabric tend to assimilate sound vitality these materials may be hard to sense at long range.
- When these sensors are interfaced with a microcontroller, or any controller then experienced person or programmer is required.
- When these sensors are used for inspection purpose then these could be water-resistive otherwise they could be damaged. [11]

3.2.5 HC-SR04 Ultrasonic Sensor – Working:

- It all starts, when a pulse of at least 10 µS (10 microseconds) in duration is applied to the Trigger pin. In response to that the sensor transmits a sonic burst of eight pulses at 40 KHz. This 8-pulse pattern makes the "ultrasonic signature" from the device unique, allowing the receiver to differentiate the transmitted pattern from the ambient ultrasonic noise. [9]
- The eight ultrasonic pulses travel through the air away from the transmitter. Meanwhile the Echo pin goes HIGH to start forming the beginning of the echo-back signal.
- If those pulses are not reflected back, then the Echo signal will timeout after 38 mS (38 milliseconds) and return low. Thus a 38 mS pulse indicates no obstruction within the range of the sensor.
- If those pulses are reflected back the Echo pin goes low as soon as the signal is received. This produces a pulse whose width varies between 150 µS to 25 mS, depending upon the time it took for the signal to be received

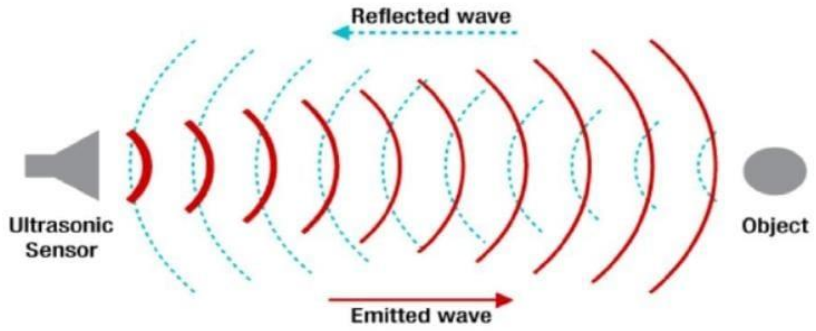


Figure 31 Ultrasonic wave reflection

- Ultrasonic sensors operate on the principle of measuring the time between usually sending a few very short pulses and receiving the reflection of the transmitted signal. The basic building blocks are the transmitter and receiver. Transmitter block may be composed of two types of transducers: Magneto strictive transducers - are working at low frequencies and their principle is based on a mechanical change in the length of magnetic material. Piezoelectric transducers - operate at high frequencies and the principle is based on the inverse piezoelectric effect. Ultrasonic receiver is based on the principle of the transfer of mechanical waves reflected back to an electrical signal. [10]
- As shown above the HC-SR04 Ultrasonic (US) sensor is a 4-pin module, whose pin names are VCC, Trigger, Echo and Ground respectively. This sensor is a very popular sensor used in many applications where measuring distance or sensing objects are required. The module has two eyes like projects in the front which forms the Ultrasonic transmitter and Receiver. The sensor works with the simple high school formula that. [10]

Distance = Speed × Time

- The Ultrasonic transmitter transmits an ultrasonic wave, this wave travels in air and when it gets objected by any material it gets reflected back toward the sensor this reflected wave is observed by the Ultrasonic receiver module
- Now, to calculate the distance using the above formulae, we should know the Speed and time. Since we are using the Ultrasonic wave, we know the universal speed of US wave at room conditions which is 330m/s. The circuitry inbuilt on the module will calculate the time taken for the US wave to come back and turns on the echo pin high for that same particular amount of time, this way we can also know the time taken. Now simply calculate the distance using a microcontroller or microprocessor.
- The width of the received pulse is then used to calculate the distance to the reflected object. This can be worked out using simple distance-speed-time equation.
- Let's take an example to make it clearer. Suppose we have an object in front of the sensor at an unknown distance and we received a pulse of width 500 µS on the Echo pin. Now let's calculate how far the object from the sensor is. We will use the below equation.

Distance = Speed x Time

- Here, we have the value of Time i.e., 500 µs and we know the speed. What speed do we have? The speed of sound, of course! Its 340 m/s. We have to convert the speed of sound into cm/µs in order to calculate the distance. A quick Google search for "speed of sound in centimetres per microsecond" will say that it is 0.034 cm/µs. You could do the math, but searching it is easier. Anyway, with that information, we can calculate the distance!

Distance = 0.034 cm/µs x 500 µs

- But this is not done! Remember that the pulse indicates the time it took for the signal to be sent out and reflected back so to get the distance so, you'll need to divide your result in half.

Distance = (0.034 cm/µs x 500 µs) / 2

Distance = 8.5 cm

Chapter 4
Accident Recording

4.1 Introduction

As the vehicle is moving in the road we need a way to track the vehicle velocity , location and update the time in the system , we also need to save this data in the case of an accident in an external memory for the system to have a record of the accident , and to be able to have saved data to be sent to the authorities and relatives of the car driver to be able to save them in the nearest time possible.

This System deals with recording data of the car periodically & in the case of an accident. The Systems takes the readings from GPS module, RTC & MPU then the MCU saves the data (Latitude, Longitude, Time & Velocity) on an I^2C EEPROM.

To track the location, we use GPS module (GPS NEOM module) It tracks over 22 satellites to identify location using signals transmitted from 3 satellites to calculate the location receiver. we receive the coordinates and save them to be sent.

As most MCU are unaware of the surrounding time so we need RTC chip. the DS-1307 manges time keeping function, with I2C interfacing. It keeps date, hours, minutes, seconds with automatic adjustment. It comes with built-in battery holder for backup power in the case of failure of main power supply. The chip needs constant power to keep incrimination of time for accurate time keeping

We use MPU data mainly to calculate the velocity of the vehicle using simple linear Newtonian motion equations & to know if the vehicle was overturned to employ the accident alert.

4.2 Motion Processing Unit (MPU)

4.2.1 Introduction

Mpu6050 is a Micro Electro-mechanical (MEMS), it consists of a three-axis accelerometer and a three-axis gyroscope. It is used to measure velocity, orientation, acceleration, displacement, and other motion-like features. Mpu6050 has a Digital Motion Processor (DMP) which has the property to solve complex problems. Mpu6050 consists of 16 bits Analog to Digital converter hardware. Which captures 3-dimensional motion at the same time and provides high accuracy. Mpu6050 operating voltage is 3 to 5 Volt. Mpu6050 uses I2C communication protocol to communicate with the microcontroller. [12]

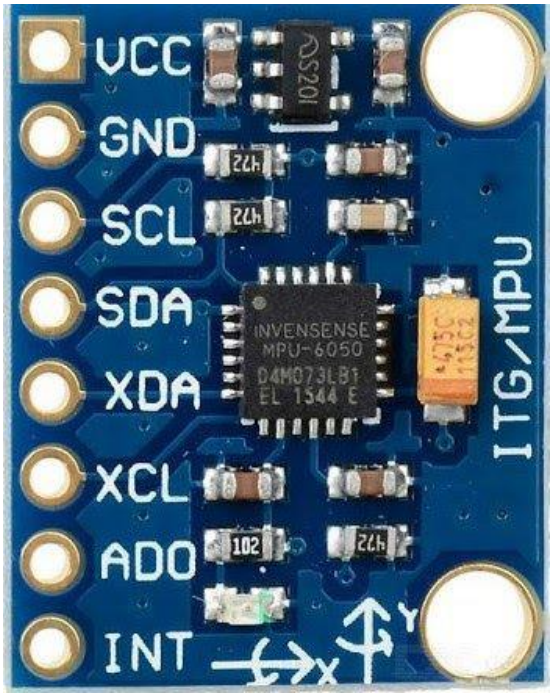


Figure 32 MPU

4.2.2 Three-Axis Accelerometer

Mpu6050 consists of a Three-Axis Accelerometer which can measure gravitational acceleration.

Using some trigonometry maths, we can calculate the angle at which the sensor is positioned known as "Tilt Angle" along the X, Y and Z axes as shown in the figure. The full-scale range of acceleration are +/- 2g, +/- 4g, +/- 8g, +/- 16g, and the measuring unit is g (gravity force).

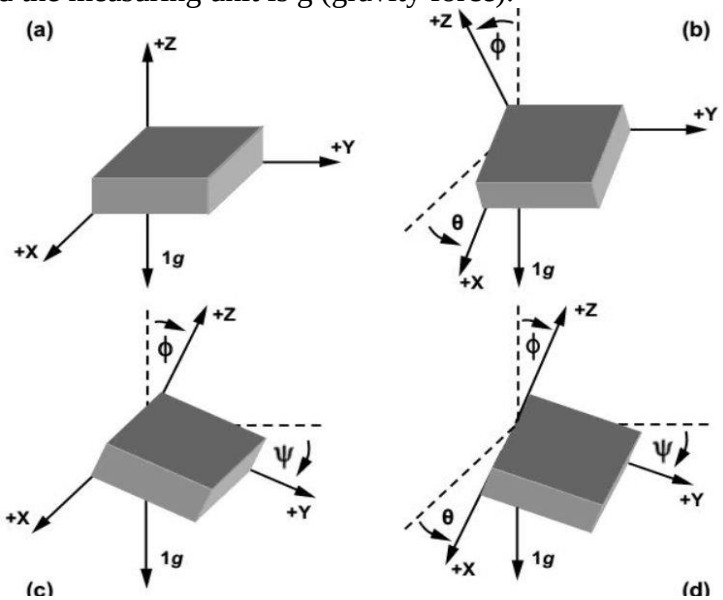


Figure 33 MPU coordinates

Acceleration is measured by measuring change in capacitance. It has a mass attached to a spring so when acceleration is applied in any particular direction the mass will be

moved and the capacitance between the plates and the mass will change. The change in capacitance measured, processed and converted to a particular value.

4.2.3 Three-Axis Gyroscope

Mpu6050 consists of a 3-Axis Gyroscope to measure rotational velocity or rate of change of the angular position over time along the X, Y and Z axis as shown in the figure. The full-scale range of gyroscope is +/-250, +/-500, +/-1000, +/-2000 measured in degrees per second unit.

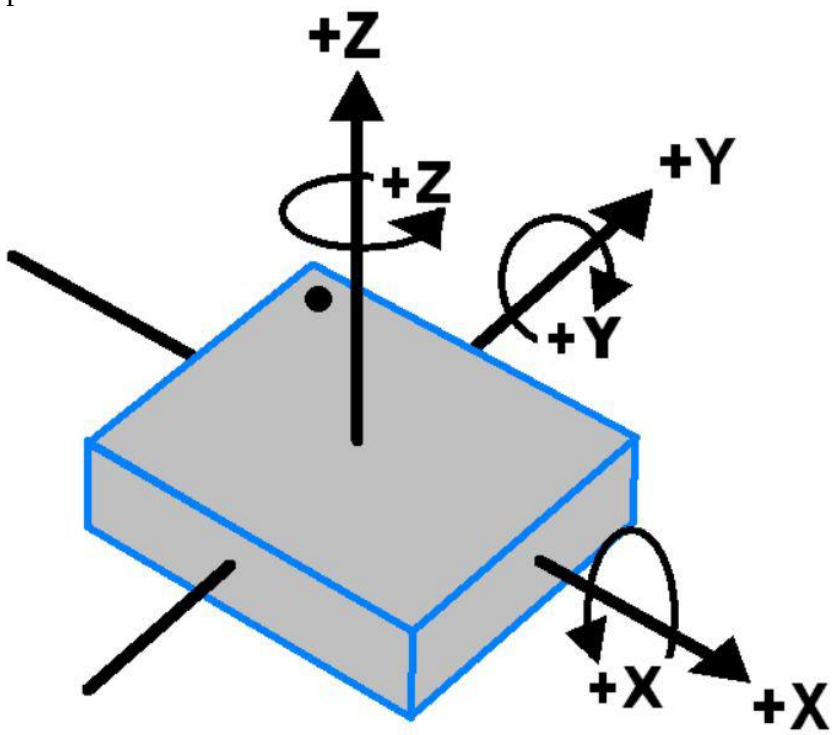


Figure 34 MPU 3-axis

The gyroscope measures angular rate using the Coriolis Effect. when a mass is moving in a particular direction with a particular velocity and when external angular rate is applied, a force will occur causing perpendicular displacement. This displacement will cause change in capacitance which is measured, processed, and converted to a particular angular rate. [12]

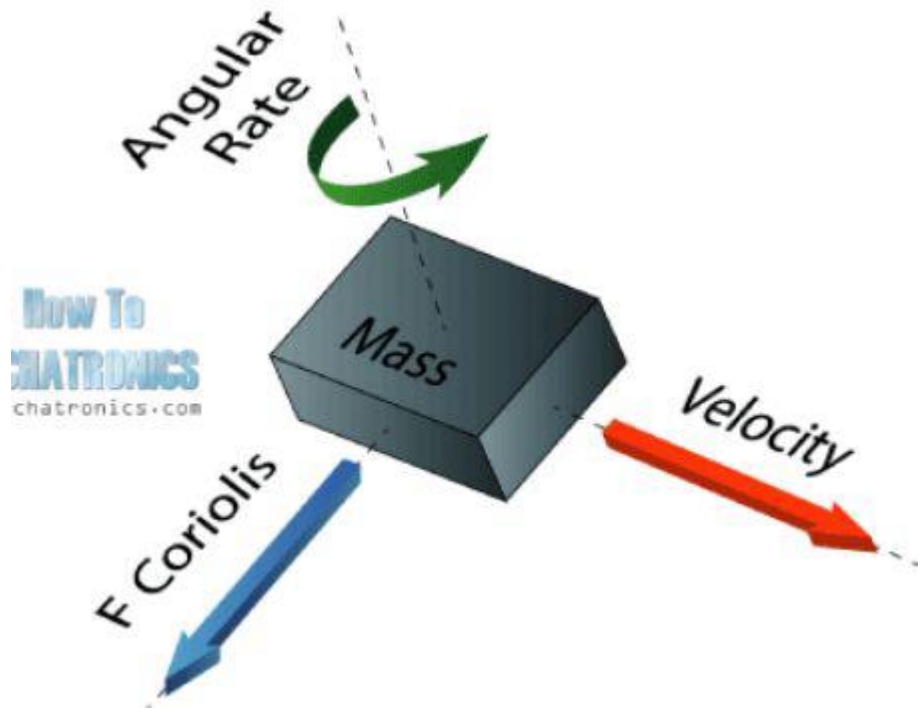


Figure 35 MPU force diagram

4.2.4 Inter Integrated Circuit (I^2C)

Mpu6050 communicates with ECUs through I2C Protocol. So, the communication goes through two algorithms polling algorithm and interrupt handling algorithm. We used some of the functions in the standard library and we developed our own transmitter and reception functions to go with our project needs as we used Real Time Operating System "RTOS" we need to receive and transmit data without waiting for them to be transmitted or received so we don't waste CPU time and don't block our system to wait for some data, so we used an interrupt mechanism for sending and receiving. We used the pulling mechanism to initiate the module and the interrupt mechanism for receiving data. The following Three designs for the sending and receiving explains how they work and how they fit in with our project.

4.2.5 Gyroscope Angles

The gyroscope measures angular rates of change around each axis with respect to the body frame. To get the angular position to the frame we should integrate these raw measurements of the gyroscope. Unfortunately, the problem here is that we are integrating which means that we are adding up many small, computed intervals to find orientation and also add up all the small errors due to noise, which then would accumulate over time and become unbounded in magnitude. This is commonly known as gyroscopic drift. In conclusion, we cannot use this approach either as the gyroscope provides accurate data about changing orientation in the short term, but the necessary integration causes the results to drift over longer time scales. The figure shows an estimate of the roll angle acquired by integrating the noisy gyroscope data. The drift can be clearly seen, even after such a short period of time. [11]

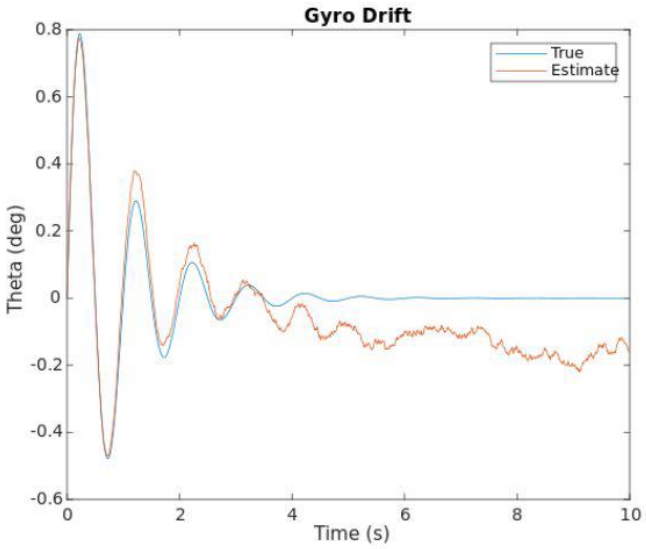


Figure 36 MPU output

4.2.6 MPU filtration

Accelerometers are fine for when the system is not in motion and gyroscopes are fine for short periods of time, but over longer periods of time – individually – both sensors will not give reliable estimates of pitch and roll. The solution to these problems is to fuse the accelerometer and gyroscope data together in such a way that the errors cancel out this is known as Sensor fusion. We are using for this approach a filter called Complementary filter. [12]

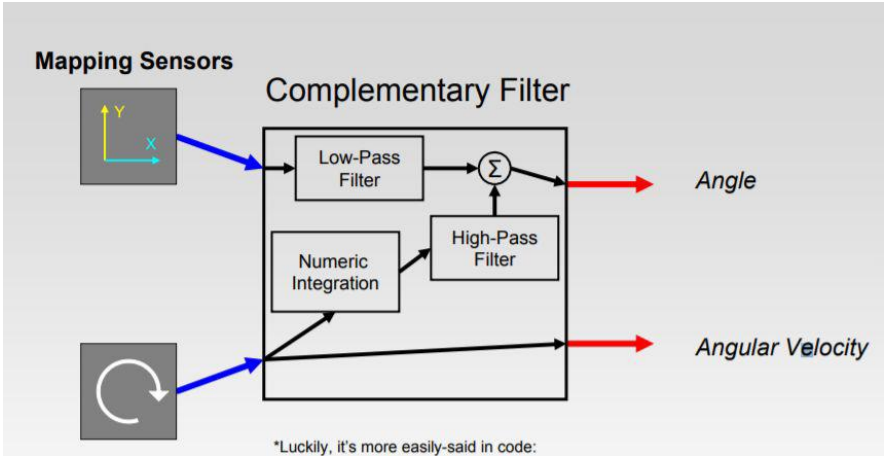


Figure 37 MPU filters

4.2.7 MPU calibration

Before using the IMU it has to be calibrated. The IMU will not be perfectly aligned with the ground, so we need to take a series of data measurements from the accelerometer and gyroscope to produce offsets. From physics perspective the offsets provide a translation from the body frame to the inertial frame. The body Frame is the

IMU mounted on the robot, whereas the Inertial Frame is the frame from which all angle and velocity calculations are done. We used two methods to calibrate the IMU and get very close values.

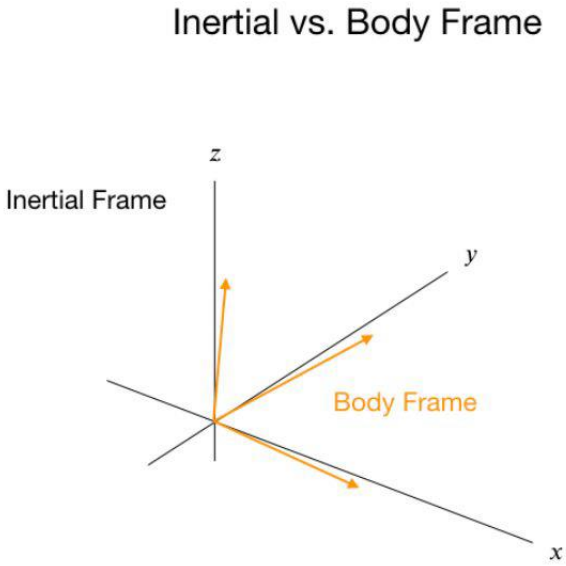


Figure 38 MPU inertial frame vs. Body frame

First Calibration Method

We used Arduino for the first method, and I got the values in the below figure. We used a built-in example to determine offsets of the IMU called “IMU Zero”.

Figure 39 MPU Calibration

Accident detection

The flowchart below shows the accident detection task and the algorithms we used to detect accident crash and accident roll-over. To check for an Accident crash we calculate the accelerometer deceleration and check if it is bigger than 5g and velocity is less than 24 km then an accident has occurred. To check for roll-over accidents, we check for tilt angles coming from the calculations of complementary filters. As usual the system could be wrong so we

decided to delay for 10 sec before taking any actions and to put a push button that will cancel the texting emergency if an accident was falsely detected. The flowchart below illustrates the accident task within RTOS and also algorithms for detecting accidents.

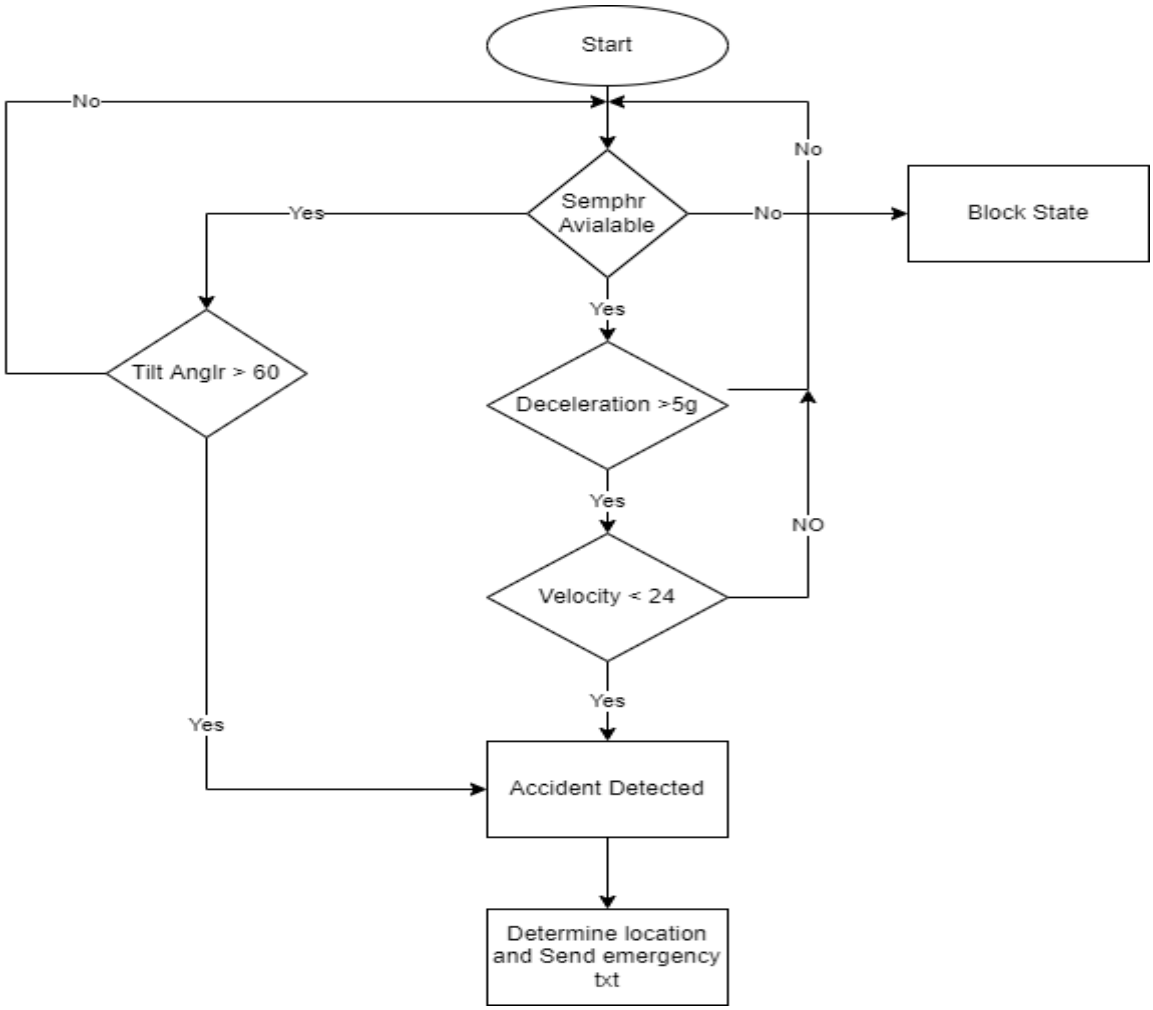


Figure 40 Accident Detection Flow Diagram

4.3 GPS

4.3.1 Introduction

GPS is a system of 30+ navigation satellites circling Earth. It's made up of three parts: satellites, ground stations, and receivers. Satellites act like the stars in constellations—we know where they are supposed to be at any given time. The ground stations use radar to make sure they are actually where we think they are. We know where they are because they constantly send out signals. A GPS receiver listens for these signals. Once the receiver calculates its distance from four or more GPS satellites, it can figure out where exactly you are.

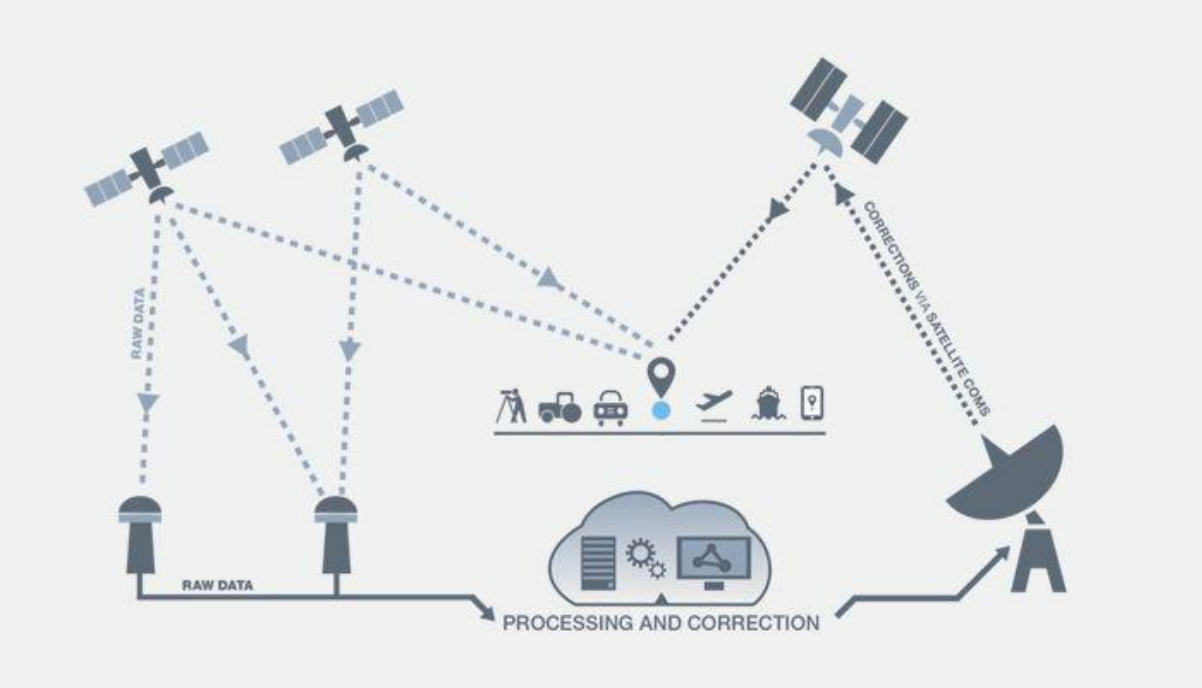


Figure 41 GPS

To locate the vehicle's location, we used ublox neo-6m GPS module

Figure 42 U-blox Module

4.3.2 GPS frame

NEO-6 modules include one configurable UART interface for serial communication, configured to operate on baud rate of 9600 b/s Available messages are in the NMEA standard, we can interpret the following messages using our module:

$GPBOD - Bearing, origin to destination $GPBWC - Bearing and distance to waypoint, great circle $GPGGA - Global Positioning System Fix Data $GPGLL - Geographic position, latitude / longitude $GPGSA - GPS DOP and active satellites $GPGSV - GPS Satellites in view $GPHDT - Heading, True $GPR00 - List of waypoints in currently active route $GPRMA - Recommended minimum specific Loran-C data $GPRMB - Recommended minimum navigation info $GPRMC - Recommended minimum specific GPS/Transit data ...

Every sentence starts with $ is extracted to multiple pieces of information. Here is an example of extracted data we used to get position and other information: NMEA has its own version of essential GPS PVT (position, velocity, time) data. It is called RMC, The Recommended Minimum, which will look similar to:
$GPRMC,123519, A,4807.038, N,01131.000, E,022.4,084.4,230394,003.1, W*6A
Where: [13]

RMC ---> Recommended Minimum sentence C 123519 ---> Fix taken at 12:35:19 UTC A ---> Status A=active or V=Void. 4807.038, N ---> Latitude 48 deg 07.038' N 01131.000, E ---> Longitude 11 deg 31.000' E 022.4 ---> Speed over the ground in knots 084.4 ---> Track angle in degrees True 230394 ---> Date - 23rd of March 1994 003.1, W ---> Magnetic Variation *6A ---> The checksum data, always begins with *

4.3.3 GPS testing

After finishing and testing the module and its driver the output is 003.1, W accurate as the following test shows

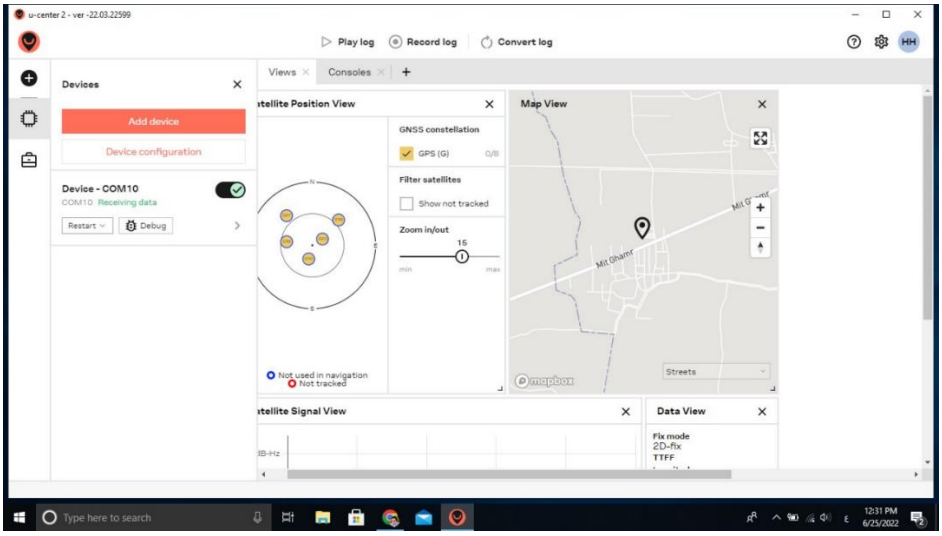


Figure 43 GPS test

4.4 Real Time Clock (RTC)

4.4.1 Introduction

We all know that most MCUs we use for our projects are time-agnostic; simply put they are unaware of the time around them. It's OK for most of our projects but once in a while when you come across an idea where keeping time is a prime concern, DS1307 RTC module is a saviour. It's perfect for projects containing data-logging, clock-building, time stamping, timers and alarms.

4.4.2 DS1307 RTC chip

At the heart of the module is a low-cost, quite accurate RTC chip from Maxim – DS1307. It manages all timekeeping functions and features a simple two-wire I2C interface which can be easily interfaced with any microcontroller of your choice.

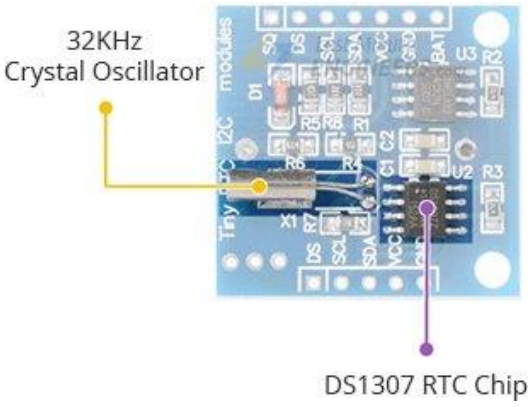


Figure 44 MPU Crystal Oscillator

The chip maintains seconds, minutes, hours, day, date, month, and year information. The date at the end of the month is automatically adjusted for months with

fewer than 31 days, including corrections for leap year (valid up to 2100). The clock operates in either the 24-hour or 12-hour format with an AM/PM indicator.

The other cool feature of this board comes with SQW pin, which outputs one of four square-wave frequencies 1Hz, 4kHz, 8kHz or 32kHz and can be enabled programmatically. [14]

4.4.3 Battery Backup

The DS1307 incorporates a battery input, and maintains accurate timekeeping when main power to the device is interrupted.

The built-in power-sense circuit continuously monitors the status of VCC to detect power failures and automatically switches to the backup supply. So, you need not worry about power outages, your MCU can still keep track of time.

Figure 45 RTC Battery

The bottom side of the board holds a battery holder for 20mm 3V lithium coin cells. Any CR2032 battery can fit well.

4.5 EEPROM

EEPROM: Electrically Erasable/ Programmable Read Only Memory. EEPROM is a non-volatile memory. EEPROM module we are using 24C16 an (8*2048) byte addressable EEPROM with i2c interfacing with addressing to access each byte.

Data block	A data block may contain 1..n bytes and is used within the API of the EEPROM driver. Data blocks are passed with - Address offset in EEPROM - Pointer to memory location - Length to the EEPROM driver.
Data unit	The smallest data entity in EEPROM. The entities may differ for read/write/erase operation.

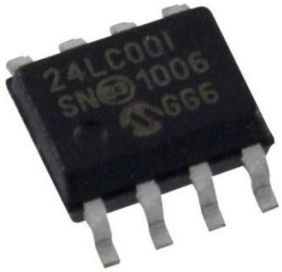


Figure 46 EEPROM

Chapter 5
Accident Alert

5.1 Introduction

when Collision sensor OR MPU OR drowsiness system sends a control massage within CAN PASS when there is a failure in the system.

GSM module takes store location from GPS (Latitude, longitude and time)

Then GSM module sent massage and make alarm call to family members, hospital and insurance company.

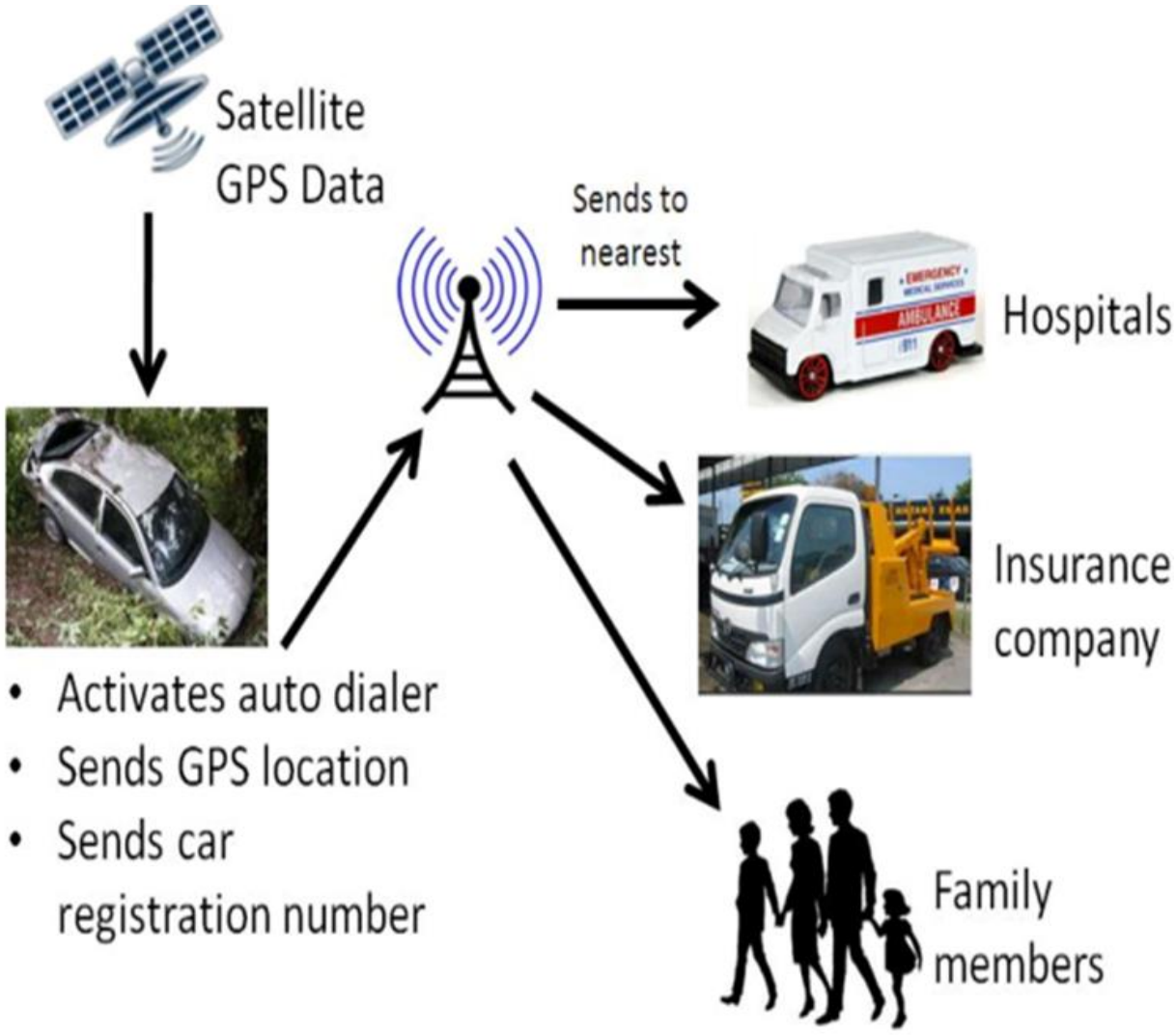


Figure 47 Accident Alert Workflow

The system consists of

-Microcontroller STM 32F103

-GSM Module SIM 800L

-GPS Module

-EEPROM 24C16

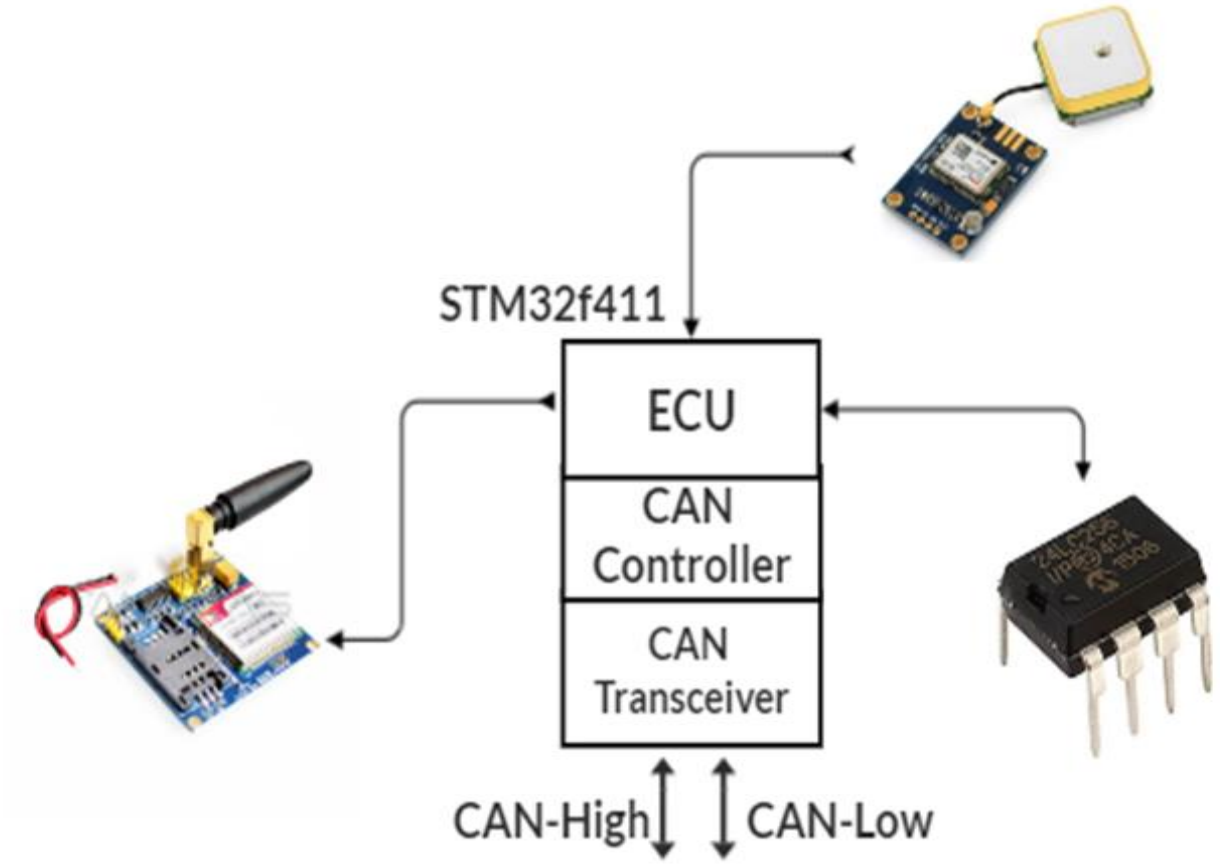


Figure 48 Accident Alert Connections

5.2 GSM

5.2.1 Introduction

SIM800L is a GSM module from Simcom that provides any GSM function for fine control, which means it can connect to the mobile network to receive calls, send and receive text messages, and also connect to the Internet using GPRS, TCP or IP. Another advantage is that the board takes advantage of existing mobile frequencies, which means it can be used anywhere in the world [15].

Figure 49 GSM Module

GSM module is Considered an antenna receive and transmitting signals. And put on its SIM card to Identifies the frequency band which use, each operator (Vodafone, Etisalat, orange, we) has a part of frequency band on any version of mobile communication (2G, 3G ,4G ,5G). SO, when use a sim card for example Vodafone use a frequency band of Vodafone on 2G version of mobile [5.2].

In this moment we can say the GSM module (SIM 800L) is consider a mobile hand set.

So:

sim800l + microcontroller by using UART protocol and set AT command = mobile hand set

5.3 Structure of GSM moduel 800L :

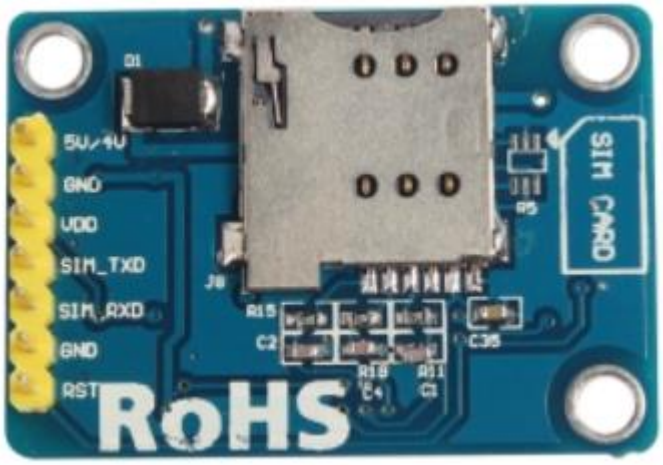


Figure 50 GSM Module

This moduel consists of 7 pins :

1 - pin : is Vcc 5 volt (input voltage source)

2 - pin : is ground

3 – pin : VDD

4 – pin : TXD - Serial data output

5- pin: RXD -Serial data input

6 – pin: ground

7 - pin: RST - Reset pin, pull low for 100ms to perform hard reset

- It is preferred to use capacitor 100u on pin 1.

There is a Gard for sim card [16].

5.4 Modulations technology

One of the ways in which 2G GSM evolution is able to provide higher data rates is to use a different modulation scheme for higher data rates. However, the GMSK modulation scheme used for the basic GSM system is still used for the lower data rates.

GMSK was chosen for the original GSM system for a variety of reasons:

It is resilient to noise when compared to many other forms of modulation. Radiation outside the accepted bandwidth is lower than other forms of phase shift keying.

It has a constant power level which allows higher efficiency RF power amplifiers to be used in the handset, thereby reducing current consumption and conserving battery life.

The GMSK modulation format is is used for the lower data rate transfers. The advantages mean that it is well suited for situations where lower data rates can be tolerated.

GMSK, Gaussian Minimum Shift Keying is a form of signal modulation that is used in a number of portable radio and wireless applications. It has advantages in terms of spectral efficiency as well as having an almost constant amplitude which allows for the use of more efficient transmitter power amplifiers, thereby saving on current consumption, a critical issue for battery power equipment. It gains its name from the fact it is filtered using a Gaussian filter [17].

5.5 Transmission technology:

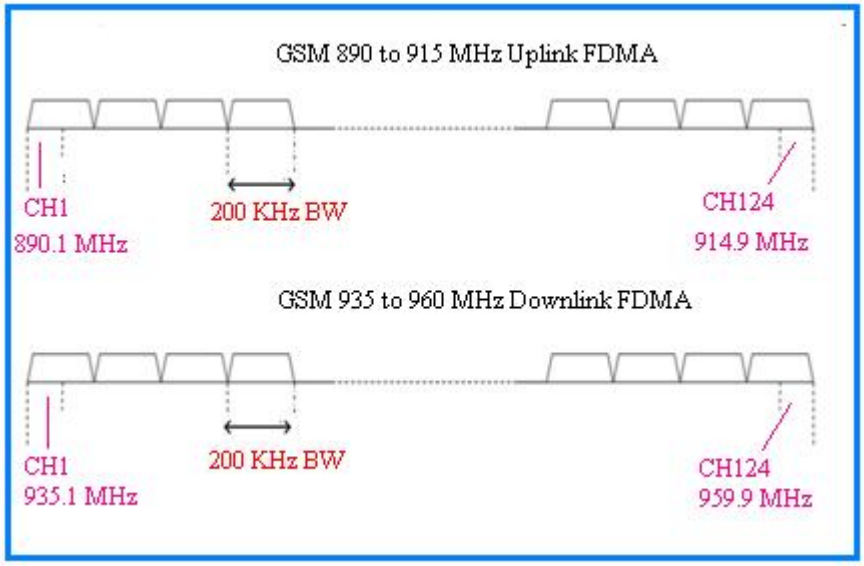


Figure 51 Transmission technology

In GSM, large frequency band (25 MHz) is divided into smaller frequency bands (200 KHz) known as channels. Moreover, separate frequency bands are allocated for uplink (890 to 915 MHz) and downlink (935 to 960 MHz)

Total of 124 channels are available with each having 200KHz bandwidth in each direction (uplink and downlink). For communication between users and Base station, one dedicated frequency is used for uplink and one for downlink. Hence simultaneous transmissions are possible in GSM. This process is known as FDMA (Frequency Division Multiple Access) over TDMA (Time Division Multiple Access) [18].

5.6 AT command:

The module recognizes AT commands, which can be used to do things like check signal strength, get the SIM card number, check the network connection and battery state, in addition to reading texts, making and receiving calls.

AT commands with a GSM/GPRS MODEM or mobile phone can be used to access following information and services:

- Information and configuration pertaining to mobile device or MODEM and SIM card.

- SMS services.

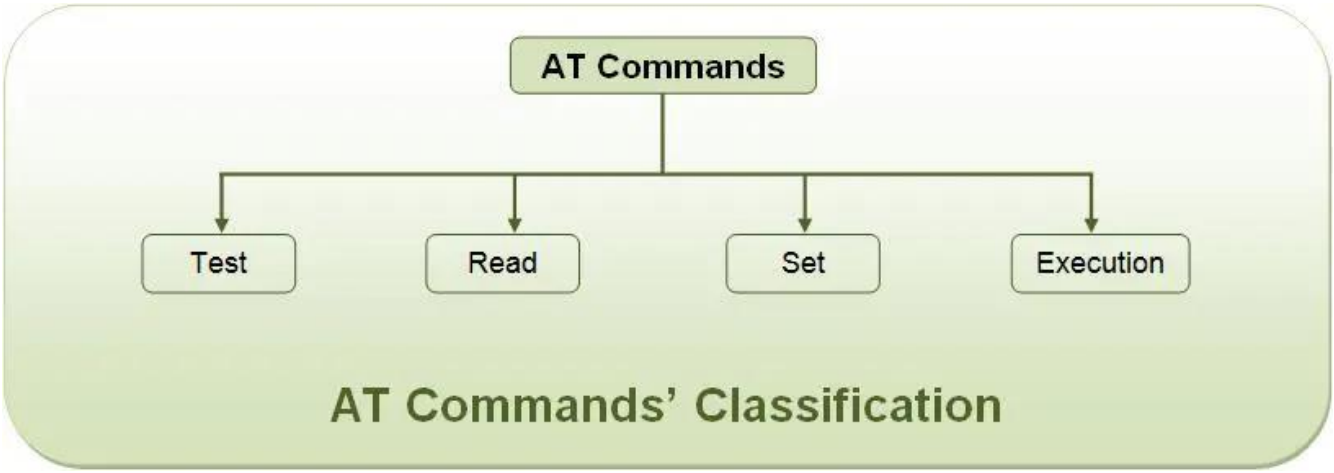
- MMS services.

- Fax services.

- Data and Voice link over mobile network. [19]

5.7 Types of AT Commands:

There are four types of AT commands:



Testing:

Command	Description
AT	Checking communication between the module and computer.

Call control:

Command	Description
ATA	Answer command
ATD	Dial command
ATH	Hang up call
ATL	Monitor speaker loudness
ATM	Monitor speaker mode
ATO	Go on-line
ATP	Set pulse dial as default
ATT	Set tone dial as default
AT+CSTA	Select type of address
AT+CRC	Cellular result codes

Data card Control for registration the network fist time:

Command	Description
ATI	Identification
ATS	Select an S-register
ATZ	Recall stored profile
AT&F	Restore factory settings
AT&V	View active configuration
AT&W	Store parameters in given profile
AT&Y	Select Set as power up option
AT+CLCK	Facility lock command
AT+COLP	Connected line identification presentation
AT+GCAP	Request complete capabilities list
AT+GMI	Request manufacturer identification
AT+GMM	Request model identification
AT+GMR	Request revision identification
AT+GSN	Request product serial number identification (IMEI)

Phone control for paging:

Command	Description
AT+CBC	Battery charge
AT+CGMI	Request manufacturer identification
AT+CGMM	Request model identification
AT+CGMR	Request revision identification
AT+CGSN	Request product serial number identification
AT+CMEE	Report mobile equipment error
AT+CPAS	Phone activity status
AT+CPBF	Find phone book entries
AT+CPBR	Read phone book entry
AT+CPBS	Select phone book memory storage
AT+CPBW	Write phone book entry
AT+CSCS	Select TE character set
AT+CSQ	Signal quality

SMS Text mode:

Command	Description
AT+CSMS	Select message service
AT+CPMS	Preferred message storage
AT+CMGF	Message format
AT+CSCA	Service centre address
AT+CSMP	Set text mode parameters
AT+CSDH	Show text mode parameters
AT+CSCB	Select cell broadcast message types
AT+CSAS	Save settings
AT+CRES	Restore settings
AT+CNMI	New message indications to TE
AT+CMGL	List messages
AT+CMGR	Read message
AT+CMGS	Send message
AT+CMSS	Send message from storage
AT+CMGW	Write message to memory
AT+CMGD	Delete message

AT+CMGL	List Messages
AT+CMGR	Read message
AT+CMGS	Send message
AT+CMGW	Write message to memory

5.8 PCB Circuit design:

Using layout software, the PCB design process combines component placement and routing to determine electrical conductivity on a manufactured circuit board.

By using Altium program design this circuit to minimize it and make it more accuracy of connection.

First, design a schematic circuit which put all component and connection it

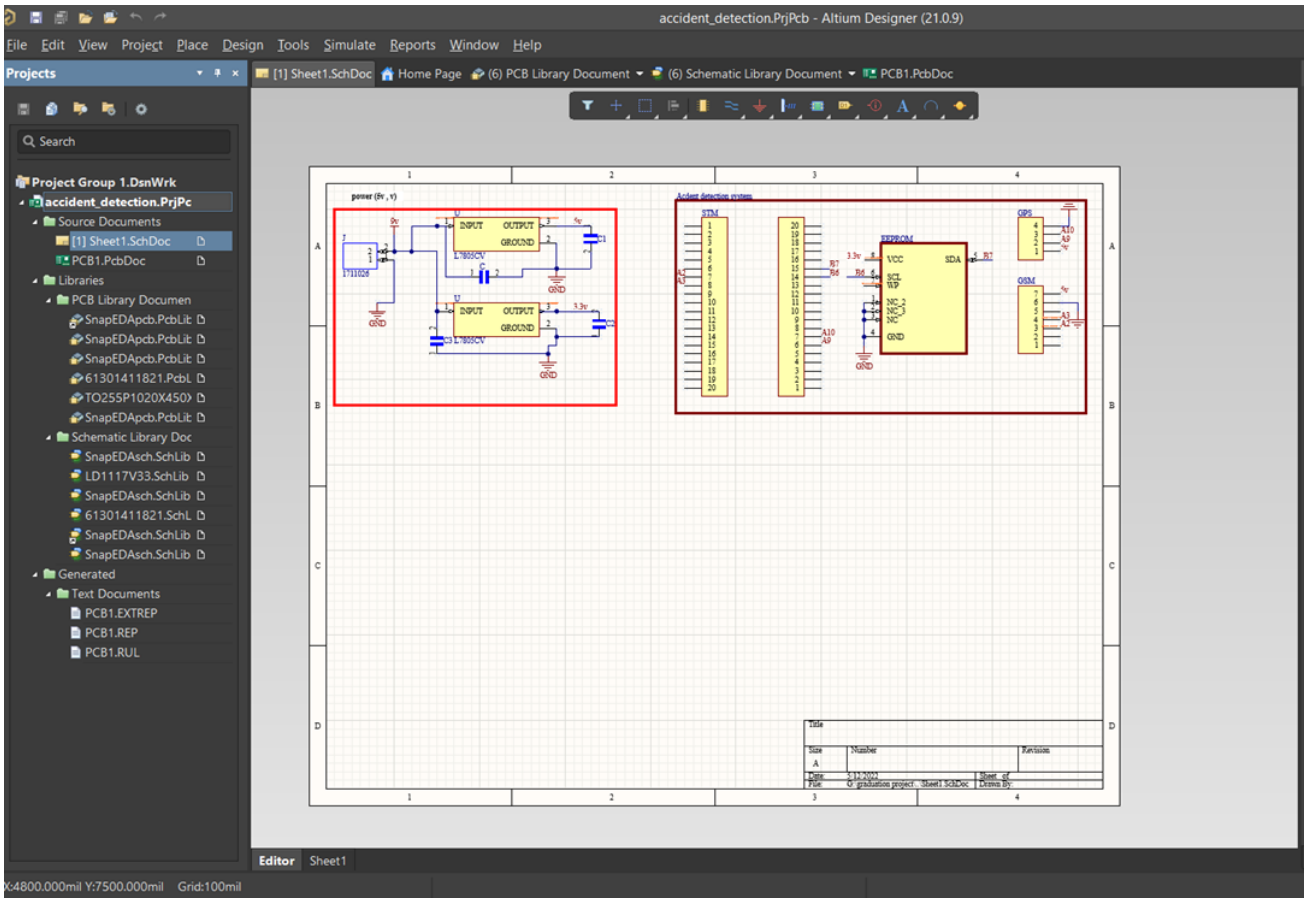


Figure 52 PCB Circuit design

Then, select all compounds as groups and move it PCB

2D

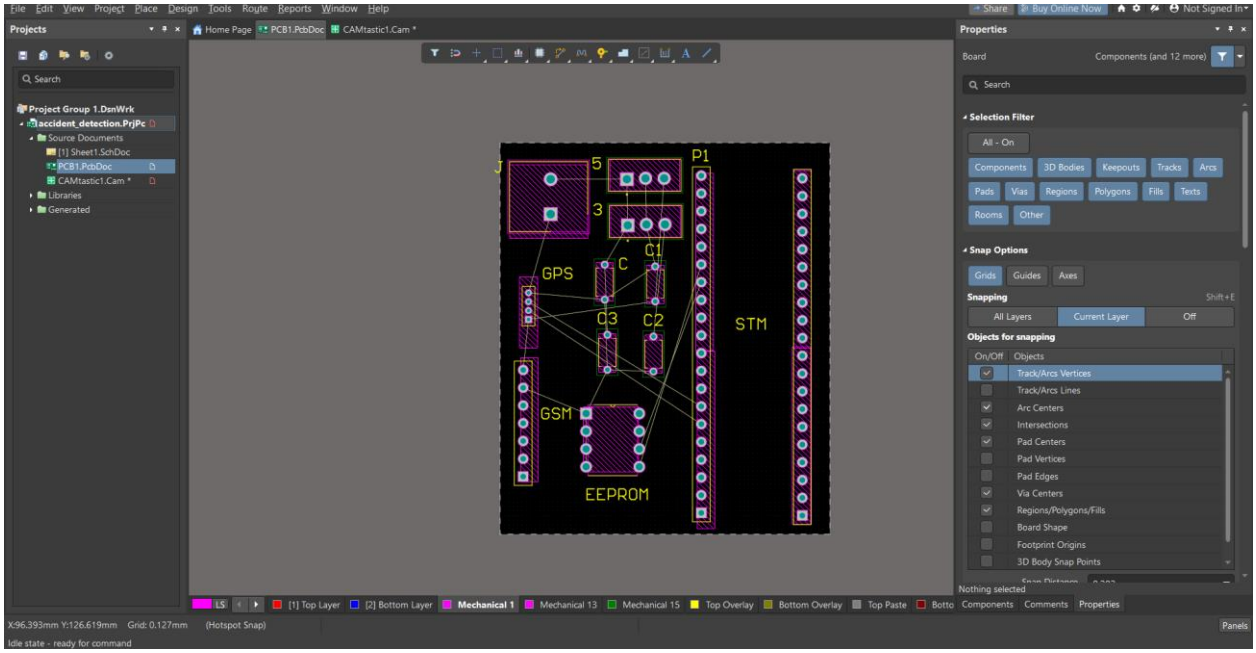


Figure 53 PCB 2D design

3D

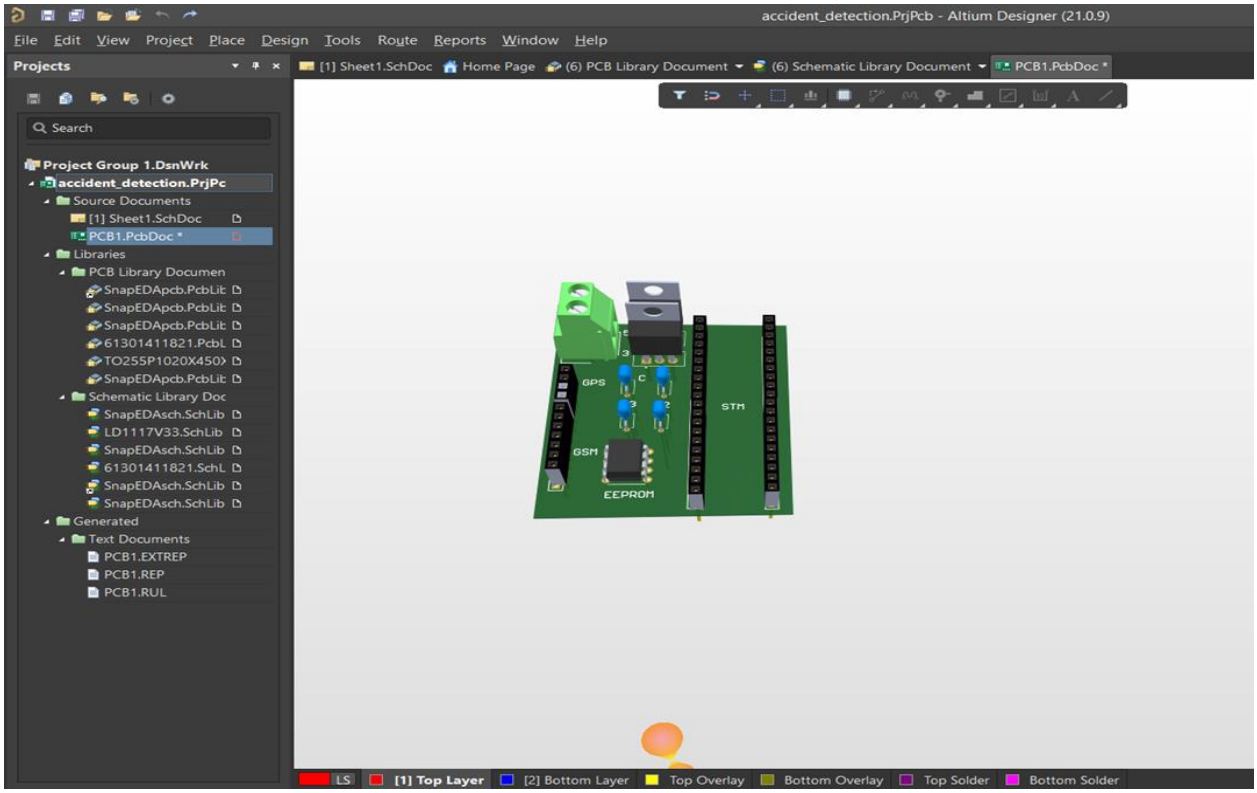


Figure 54 PCB 3D design

Chapter 6 Intercommunication System

6.1 Introduction

In order to make the car smart and more safety new ECUs were added to the cars, it is not a problem. The problem was how to connect between these different ECUs to deliver the data from ECU to another. In order to solve the problem of wiring in the cars, Bosch started to develop a new serial communication protocol in the early1980s. In 1986, Bosch introduced The Controller Area Network (CAN) bus to the world, The first use of CAN bus was in 1988 BMW 8 Series cars. The last version of CAN bus is CAN with Flexible Data-Rate (CAN FD) and it released from Bosch in 2012, this version speed reach to 8 Megabit/sec and allows to sending 64 bytes of data in every message. The CAN bus is also using to run diagnostics on the cars to report the status of each ECU in the car this feature makes the fix of car's problem easier. Today all cars in the world use CAN bus as a main serial communication protocol between most of ECUs for several reasons:

• Message priority assignment and guaranteed maximum latencies. • Multicast communication with bit-oriented synchronization.

• System-wide data consistency.

• Bus multi master access.

• Error detection and signalling with automatic retransmission of corrupted messages.

• Detection of permanent failures in nodes, and automatic switch-off to isolate faulty node.

6.2 Layers Structure of CAN

Many network protocols are described using the seven-layer Open System Interconnection (OSI) model, as shown in Figure (4-1). The Controller Area Network (CAN) protocol defines the Data Link Layer and part of the Physical Layer in the OSI model. The remaining physical layer (and all of the higher layers) are not defined by the CAN specification. These other layers can either be defined by the system designer, or they can be implemented using existing non-proprietary Higher Layer Protocols (HLPs) and physical layers. [20]

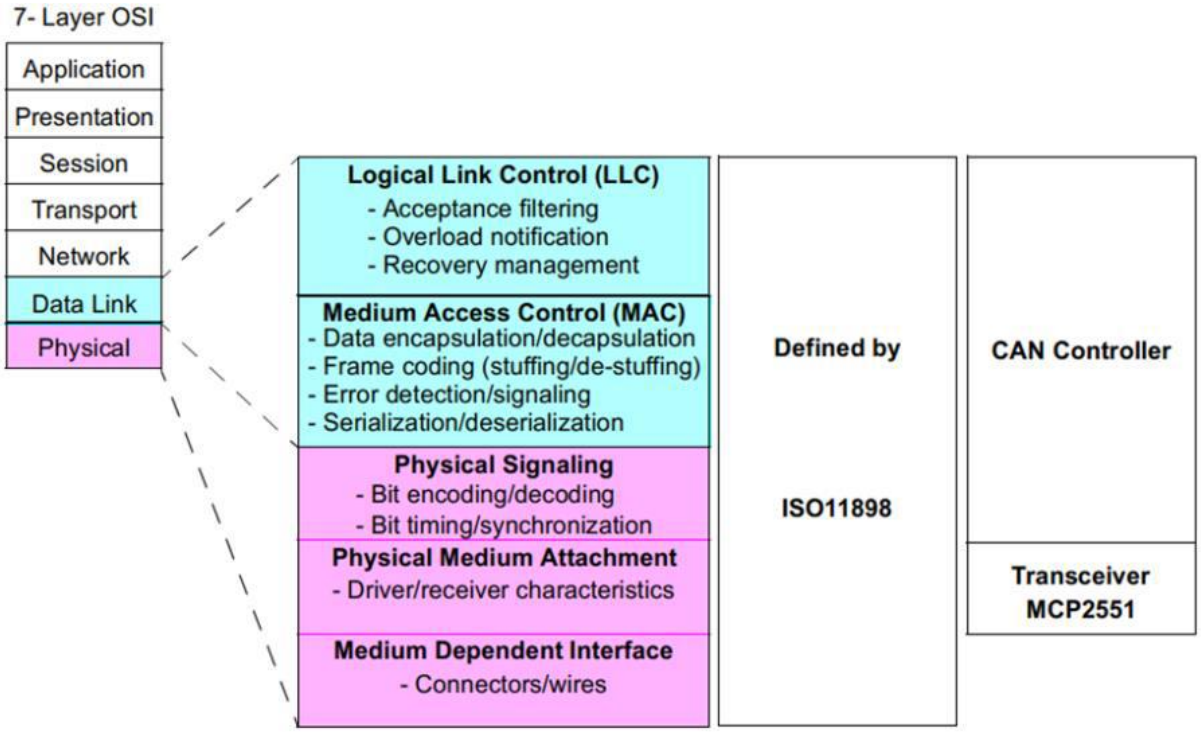


Figure 55 Layers Structure of CAN

6.2.1 Physical layer

Is the basic hardware required for a CAN network, i.e., the ISO 11898 electrical specifications? It converts 1‘s and 0‘s into electrical pulses leaving a node, then back again for a CAN message entering a node. Although the other layers may be implemented in software or in hardware as a chip function, the Physical Layer is always implemented in hardware The impact on the physical, which includes the transceiver, the network and the interface between microcontroller and transceiver, will be discussed in this part. [21]

6.2.1.1 Bus Levels

CAN specifies two logical states: recessive and dominant. ISO-11898 defines a differential voltage to represent recessive and dominant states (or bits)

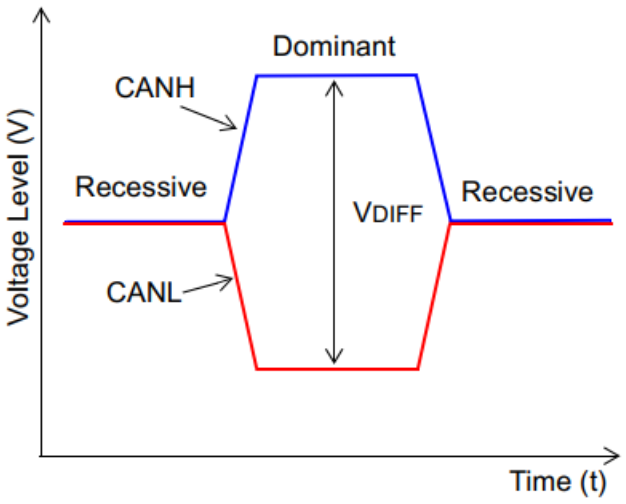


Figure 56 Bus Levels

In the recessive state:

(logic_1‘on the MCP2551 TXD input), the differential voltage on CANH and CANL is less than the minimum threshold (<0.5V receiver input or <1.5V transmitter output).

In the dominant state:

(logic_0‘on the MCP2551 TXD input), the differential voltage on CANH and CANL is greater than the minimum threshold. A dominant bit overdrives a recessive bit on the bus to achieve node strictive bitwise arbitration.

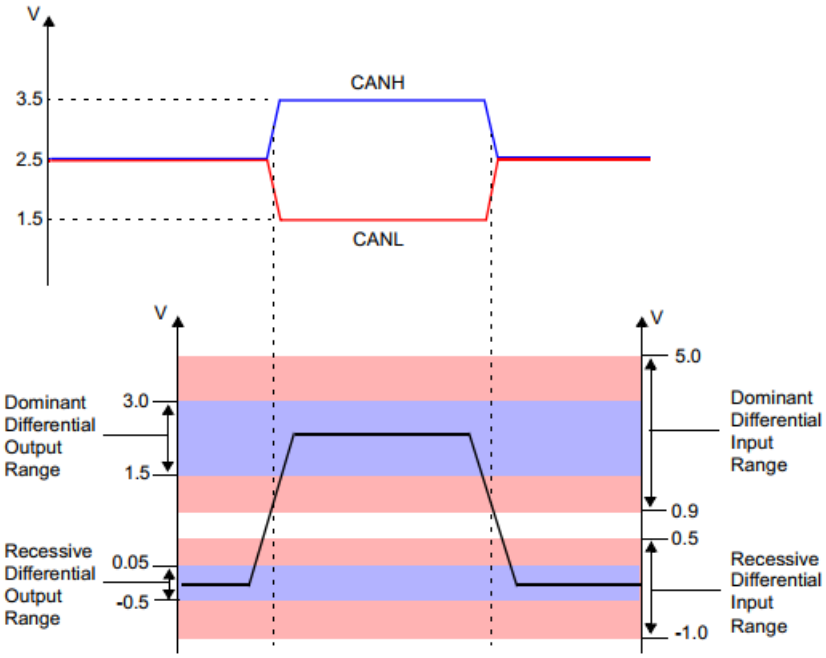


Figure 57 ISO11898 NOMINAL BUS LEVELS

6.2.1.2 Connectors and Wires

ISO-11898-2 does not specify the mechanical wires and connectors. However, the specification does require that the wires and connectors meet the electrical specification. The specification also requires 120Ω (nominal) terminating resistors at each end of the bus. Transceiver used: MCP2551

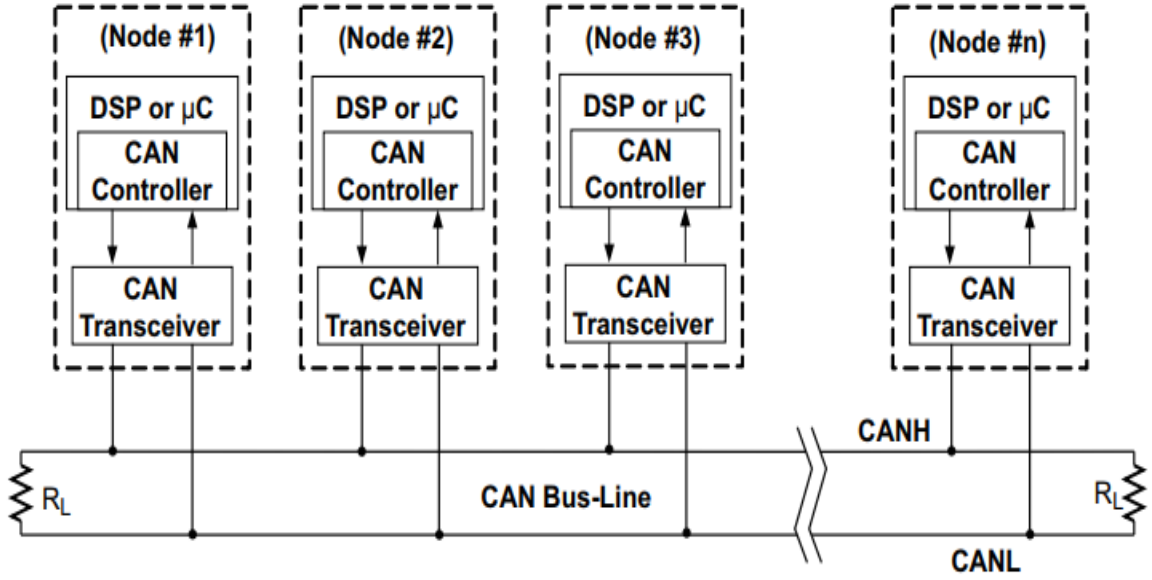


Figure 58 Details of a Typical CAN Node

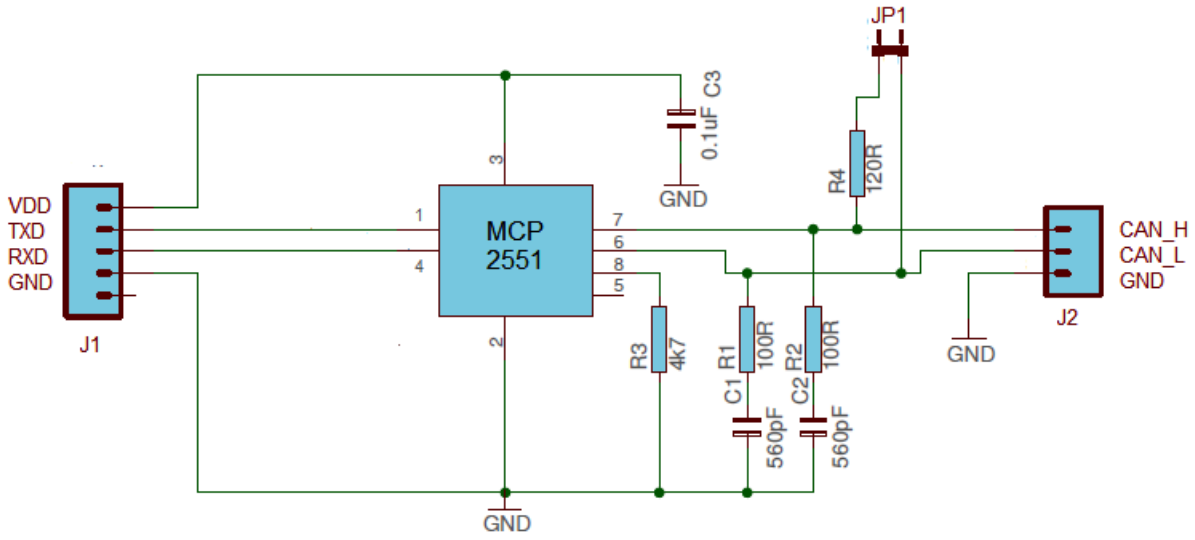


Figure 59 MCP Wiring schematic

6.2.2 Data Link Layer

Is responsible for transferring messages from a node to the network without errors. It handles bit stuffing and checksums, and after sending a message, waits for acknowledgment from the receivers.

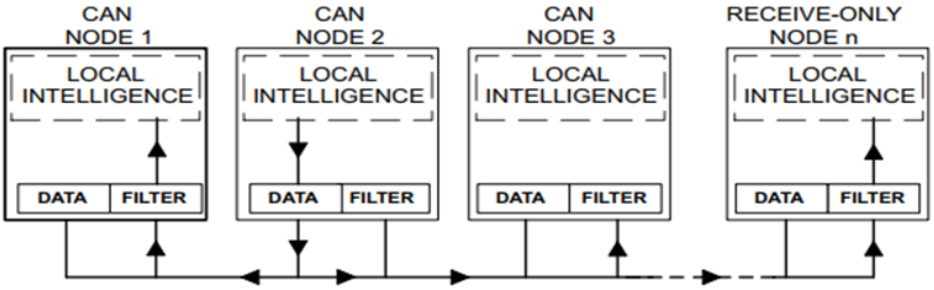


Figure 60 The CAN Data-Flow Model

since CAN is a broadcast system, a transmitting node places data on the network for all nodes to access. As shown in Figure, only those nodes requiring updated data allow the message to pass through a filter that is set by the network designer messages from certain nodes can pass, and all others are ignored.

6.3 Message Frame Format

In CAN, there are four different types of frames, defined according to their content and function.

Data frames: which contain data information from a source to (possibly) multiple receivers.

Remote frames: which are used to request transmission of a corresponding (same identifier) Data frame.

Error frame: which are transmitted whenever a node on the network detects an error.

Overload frames: which are used for flow control to request an additional time delay before the transmission of a Data or Remote frame. [21]

6.3.1 Data Frame Format

Data frames are used to transmit information between a source node and one or more receivers.

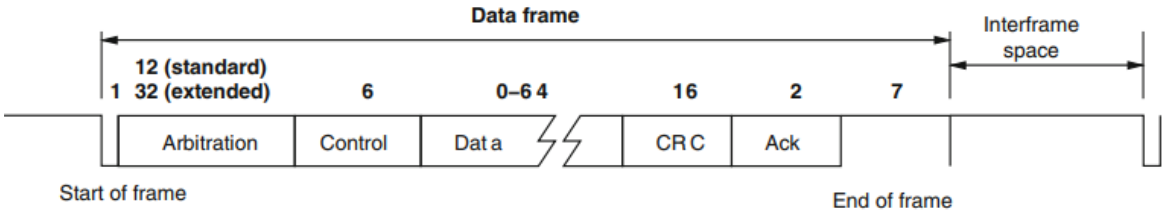


Figure 61 Data Frame Format

There are two different formats for CAN messages shown in Figure (58), according to the type of message identifier that is used by the protocol.: - Standard frames are frames defined with an 11-bit Identifier field. - Extended frames have been made available from version 2.0 of the protocol as frames with a 29-bit Identifier field.

Standard and extended frames can be transmitted on the same bus by different nodes or by the same node. The arbitration part of the protocol works regardless of the identifier version of the transmitted frames, allowing 29-bit identifier messages to be transmitted on the same network together with others with an 11-bit identifier.

In our project we used the both to identify messages IDs. [20]

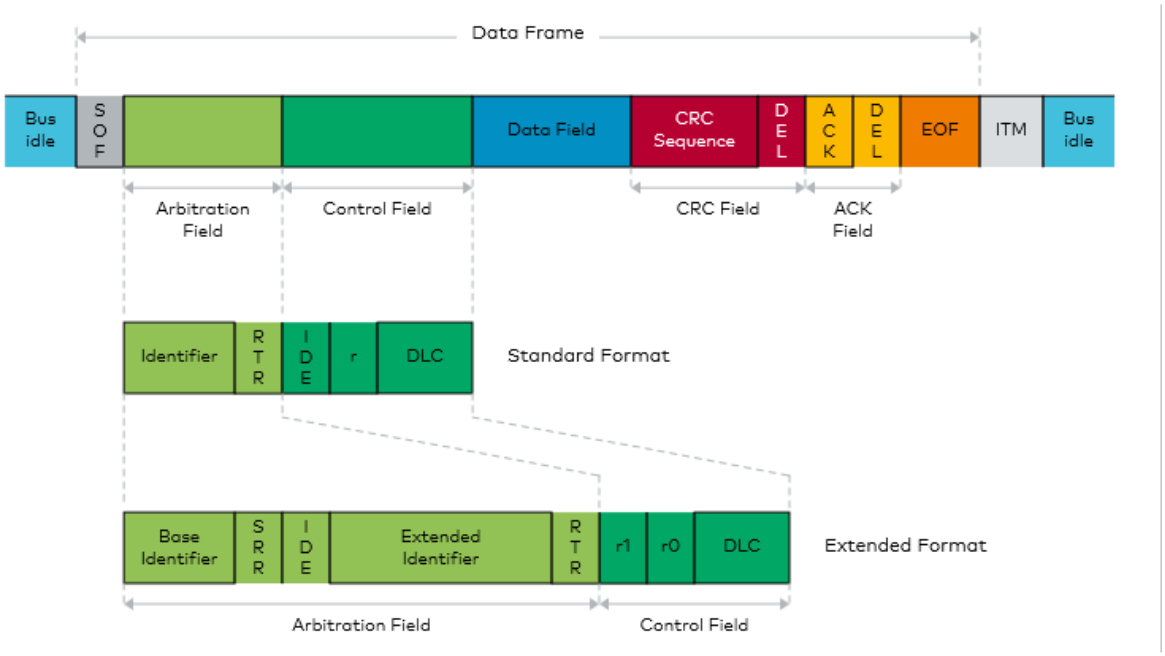


Figure 62 Standard and Extended Frame Format

Segments of the frame:
SOF: - (start of frame) it's always a dominant bit to mark the start of any CAN frame. Arbitration Field: - In standard Frame it's consists of 11-bit identifier, so it allows variety of 2048 identifier this 11-bit followed remote transmission request bit (RTR) which is dominant in Data Frame and recessive in Remote Frame. - In Extends Frame It's consists of 29-bit identifier, so it allows identifier the first 11 bits base identifier is followed by substitute Remote Request bit (SRR) it's a recessive bit. Identifier extension bit (IDE) and it's part of arbitration field which is dominant in Standard Frame and recessive in Extended Frame.

Control Field: - consists of IDE bit followed by 4 DLC bits Data length code to indicate how many data bytes are transmitted and it may take value from 0 to 8.
Data Field: - it's comprising the actual information to be transmitted. number of data bytes may vary from 0 to 8 byte.
CRC &ACK Field: - The payload is protected by a checksum using a cyclic redundancy check (CRC) which is ended by a delimiter bit. Based on the results of the CRC, the receivers acknowledge positively or negatively in the ACK slot (acknowledgement) which also is followed by a delimiter bit.
EOF Field: After this the transmission of a data frame is terminated by seven recessive bits (End of Frame — EOF).

6.3.2 Error Frame Format

The Error frame is not a real frame, but rather the result of error signalling and recovery action(s). The Error frame given that it consists of 6 dominant or recessive bits in a sequence is of fixed form with no bit stuffing. The CAN protocol identifies and defines management for five different error types:
• Bit error: A Bit error is detected when the bit value read from the bus is different from the bit value that is sent.
• Stuff error: A Stuff error is detected at the sixth consecutive occurrence of the same bit in a message field that is subject to bit stuffing.

• CRC error: If the CRC computed by the receiver differs from the one stored in the message frame.

• Form error: A Form error occurs when a fixed-form bit field contains one or more illegal bits. There's some fixed form of fields in CAN: CRC, DEL, ACK, EOF, Ifs These bits must be recessive bits, if receiver find any dominant bits in these fields it's a frame error.

• Acknowledgement error: It is detected by a transmitter if a recessive bit is found on the ACK slot.

First, we need to know that for each node there are two error counters Transmit error counter & Receive error counter. CAN protocol includes a fault confinement mechanism that detects faulty units and places them in passive or off states, so that they cannot affect the bus state with their outputs. The protocol assigns each node to one of three states: 1- Error Active State: units in this state are assumed to function properly. Units can transmit on the bus and signal errors with an Active Error flag. 2- Error Passive State: units in this state are suspected of faulty behaviour (in transmission or reception). They can still transmit on the bus, but their error signalling capability is restricted to the transmission of a Passive Error flag. 3- Buss Off State:

units in this state are very likely corrupted and cannot have any influence on the bus. (Their output drivers should be switched off). [22]

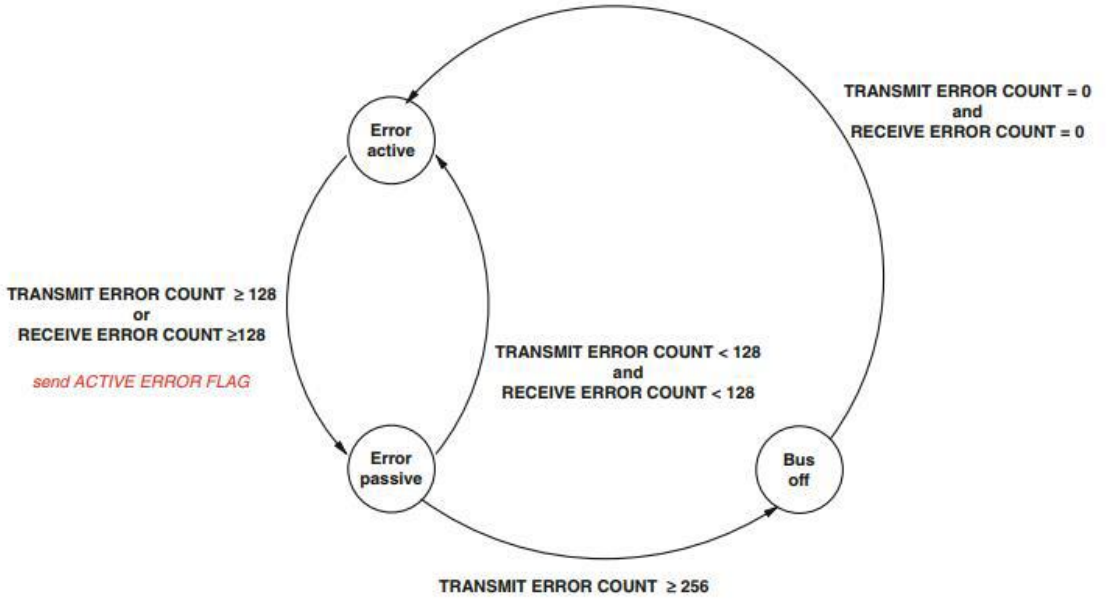


Figure 63 Bit Sequence after detection of an error

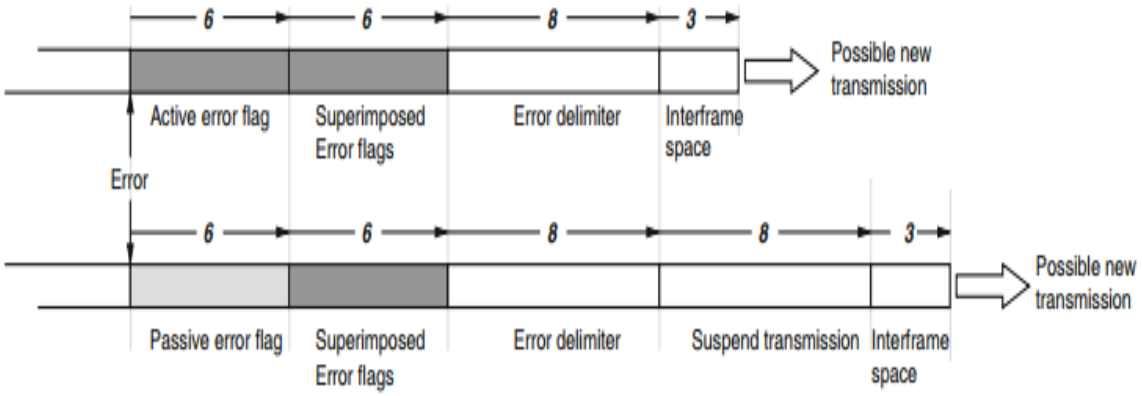


Figure 64 Detection of an error

6.4 Bit Timing

The Controller Area Network (CAN) is a serial, asynchronous, multi-master communication protocol for connecting electronic control modules in automotive and industrial applications. A feature of the CAN protocol is that the bit rate, bit sample point and number of samples per bit are programmable. This gives the opportunity to optimize the performance of the network for a given application.

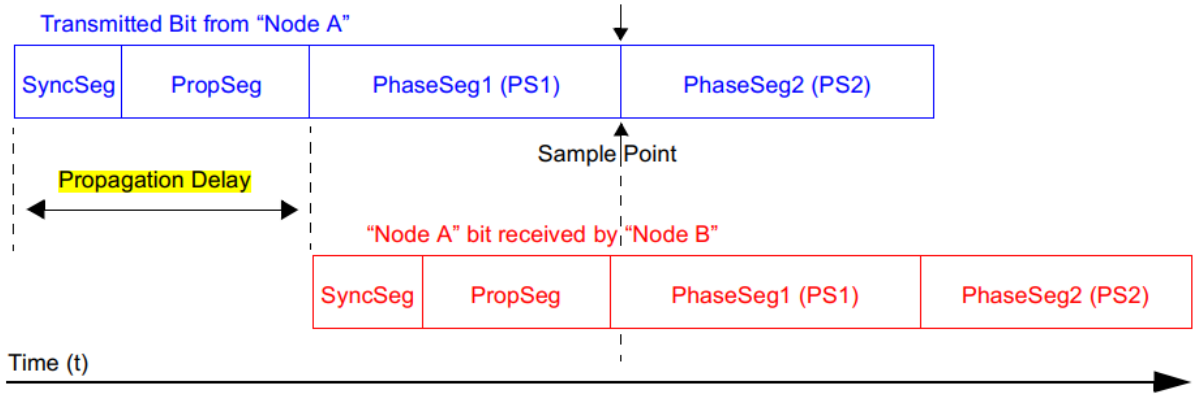


Figure 65 One-Way Propagation Delay

- Synchronization segment (SYNC SEG): This is a reference interval, used for synchronization purposes. The leading edge of a bit is expected to lie within this segment.

- Propagation segment (PROP SEG): This part of the bit time is used to compensate for the (physical) propagation delays within the network. It is twice the sum of the signal propagation time on the bus line, plus the input comparator delay, and the output driver delay.

- Phase segments (PHASE SEG1 and PHASE SEG2): These phase segments are time buffers used to compensate for phase errors in the position of the bit edge. These segments can be lengthened or shortened to resynchronize the position of SYNCH SEG with respect to the following bit edge.

- Sample point (SAMPLE POINT): The sample point is the point of time at which the bus level is read and interpreted as the value of that respective bit. The quantity information processing time is defined as the time required to convert the electrical state of the bus, as read at the SAMPLE POINT into the corresponding bit value. [22]

6.5 Developing CAN Interface Driver

Using STM32f103c8T6 Microcontroller we implement CAN Interface Driver. the discussion of steps presented in Appendix Code part.

6.5.1 CAN Interface In Microcontroller

The protocol controller transfers and receives the serial data from the CAN bus and passes the data on to the message handler. -The message handler then loads this information into the appropriate message object based on the current filtering and identifiers in the message object memory. -The message handler is also responsible for generating interrupts based on events on the CAN bus. - The message object memory is a set of 32 identical memory blocks that hold the current configuration, status, and actual data for each message object. These memory blocks are accessed via either of the CAN message object register interfaces. - The message memory is not directly accessible in the STM32f103c8T6 memory map, so the STM32f103c8T6 CAN controller provides an interface to communicate with the message memory via two CAN interface register sets for communicating with the message objects. These two interfaces used to read or write to each message object. - The two message object interfaces allow parallel access to the CAN controller message objects when multiple objects may have new information that must be processed. In general, one interface is used for transmitting data and one for receive data.

STM32f103c8T6 microcontroller includes two CAN units with 32 message Objects with individual identifier Masks we can use them individually or using FIFO Mode which enables Storage of multiple message objects. - The whole message (including all arbitration bits, data-length code, and eight data bytes) is stored in the message object. If the Identifier Mask is used, the arbitration bits that are masked to "don't care" may be overwritten in the message object. - The transmission of any number of message objects may be requested at the same time; they are transmitted according to their internal priority, which is based on the message identifier for the message object, with 1 being the highest priority and 32 being the lowest priority.

6.5.2 CAN Filters

The filtering is done by arbitration identifier of the CAN frame. This technique is also used when monitoring a CAN bus, in order to focus in on messages of importance using an identifier and mask. These allow a range of IDs to be accepted with a single filter rule. When a CAN frame is received, the mask is applied. This determines which bits of the identifier will be used to determine if the frame matches the filter. -If a mask bit is set to a zero, the corresponding ID bit will automatically be accepted, regardless of the value of the filter bit. - If a mask bit is set to a one, the corresponding ID bit will be compared with the value of the filter bit; if they match it is accepted otherwise the frame is rejected. [23]

. For Example:

```
identifier = 0x123, mask = 0x7FF -> matches only frames with identifier 0x123

identifier = 0x123, mask = 0x7F0 -> matches frames with identifiers 0x120, 0x121, 0x122, 0x123, ...,
```

6.5.3 FIFO Buffer:

First in First Out concept using in CAN to Store multiple Messages Received in RAM. as Receiving each message Individually keeps interrupting the CPU for every single Message receives. It will make system overhead especially in multitasking systems. When the CPU transfers the contents of a message object from a FIFO buffer. To assure the correct function of a FIFO buffer, the CPU should read out the message objects starting with the message object with the lowest message number. When reading from the FIFO buffer, should be aware that a new received message is placed in the message object with the lowest message number. As a result, the order of the received messages in the FIFO is not guaranteed.

6.5 Flow Charts

6.5.1 Transmission Flow Chart

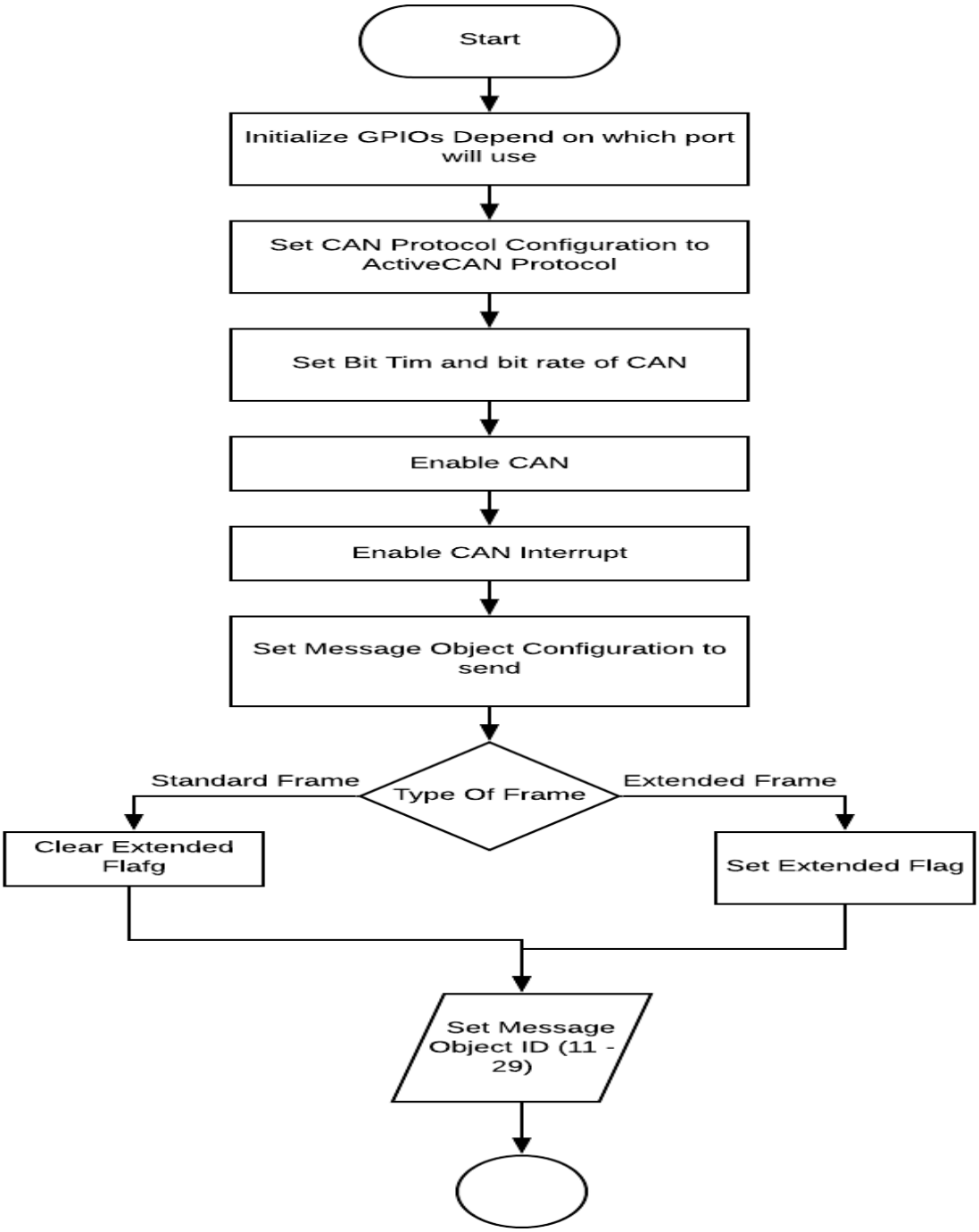


Figure 66 Transmission Flow Chart

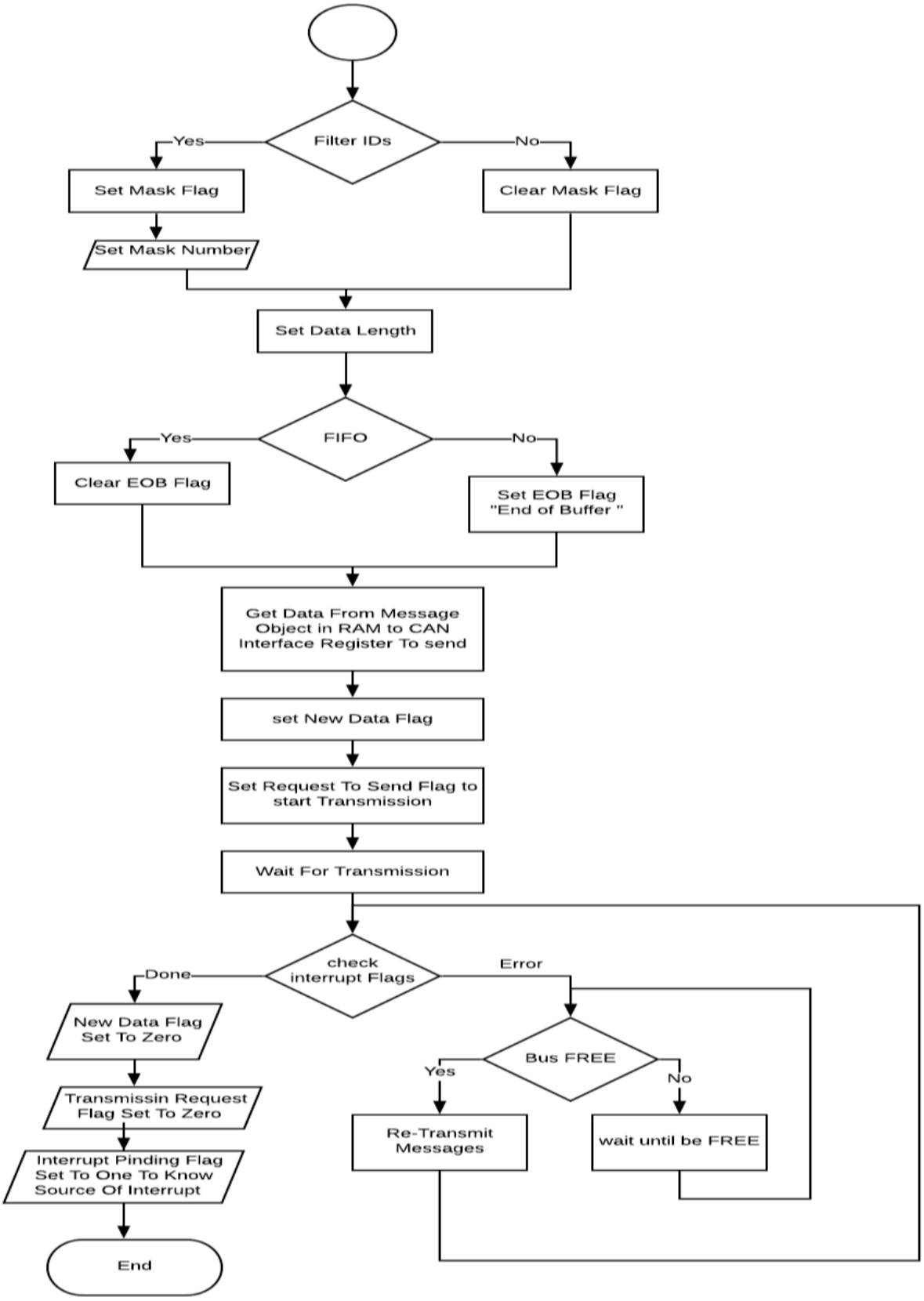


Figure 67 Transmission Flow Chart

6.5.2 Receiving Flow Chart

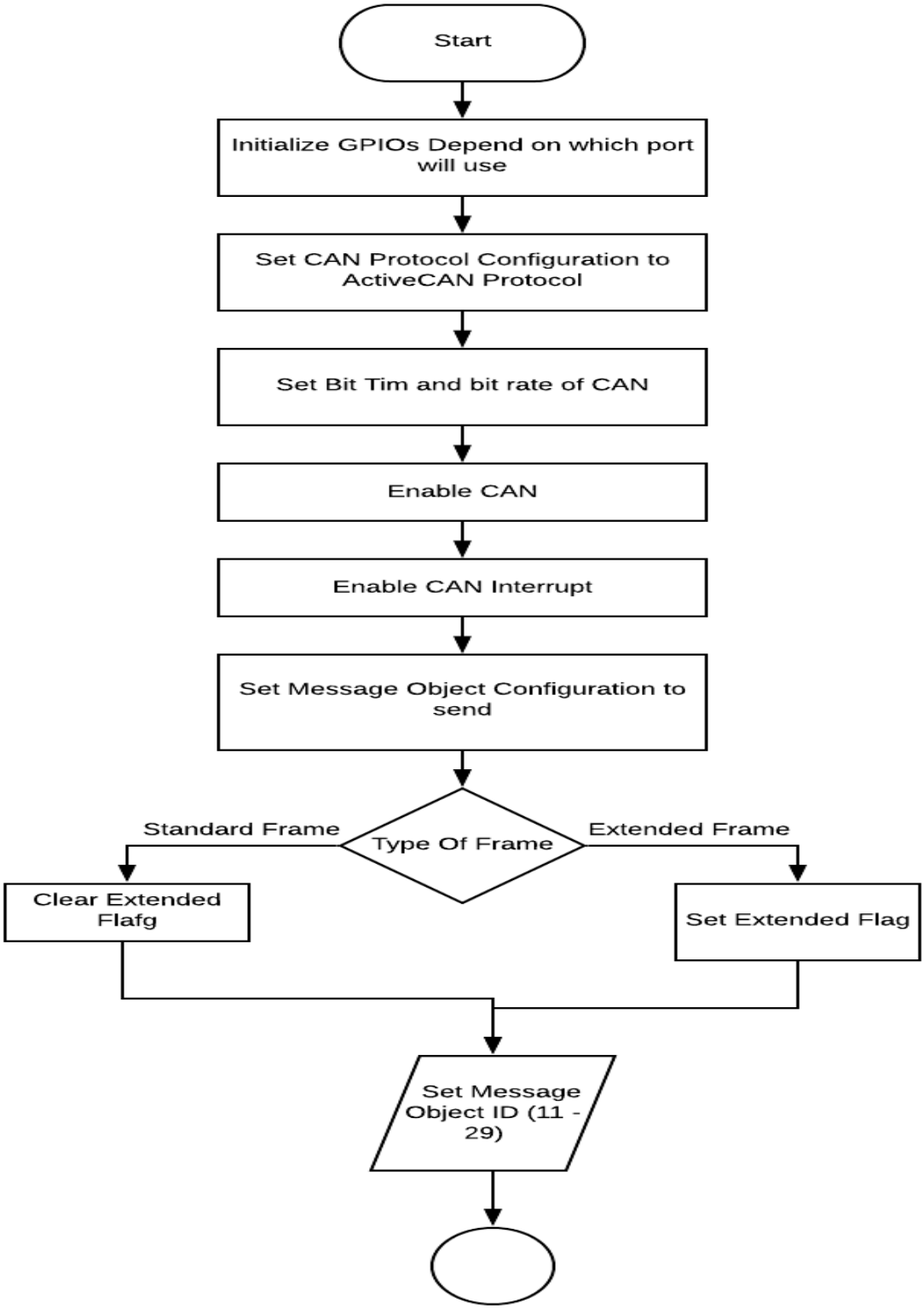


Figure 68 Receiving Flow Chart

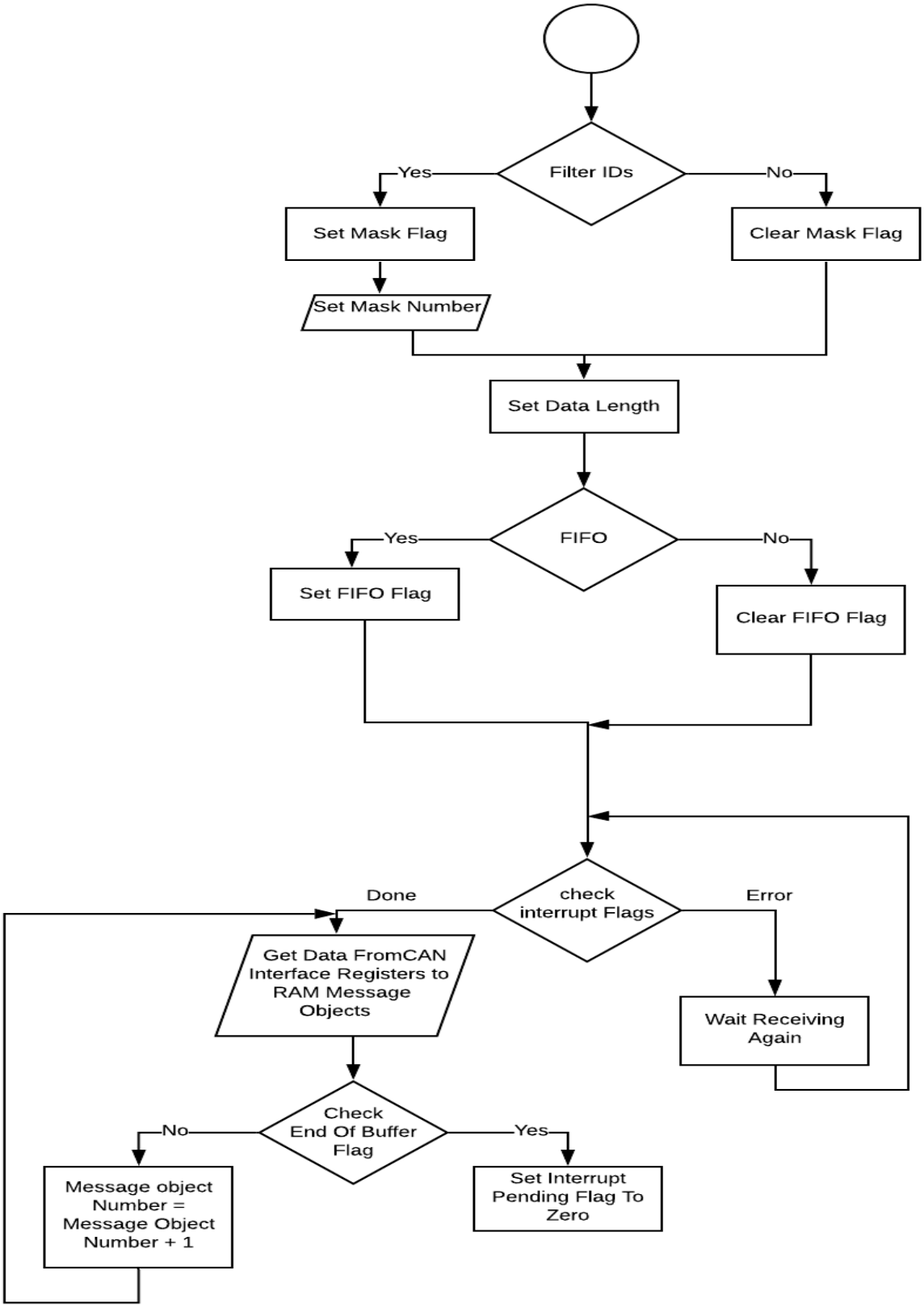


Figure 69 Receiving Flow Chart

Chapter 7
Drowsiness Detection

7.1 Abstract

Distracted driving is the reason for large number of accidents and death in the whole world. It includes many reasons like mobile texting, eating, being drowsy and falling asleep. Driving a vehicle when sleepy is known as drowsy driving. Drowsy driving significantly increases the risk of accidents, leading to a troubling number of injuries and deaths every year.

Research indicates that it is disturbingly common. The National Sleep Foundation found that through 4 years 2005 to 2009 drowsy driving killed off too many people also in America Poll found that 60% of adult drivers reported driving while drowsy in the past year. Survey data from the CDC indicated that one in every 25 adults had fallen asleep behind the wheel in the past month [24]

The Drowsy driving is a major contributor to motor vehicle collisions. According to the National Highway Traffic Safety Administration (NHTSA), in 2017 the drowsiness while driving led to at least 91,000 crashes, resulting in roughly 50,000 injuries and 800 deaths.

Other studies calculate that drowsy driving causes up to 6,000 deadly crashes every year. Researchers estimate that around 21% of fatal car crashes involve a person driving while drowsy.

Also, as an example there are nearly over 6 million car accidents in the United States every year. The road crashes are the leading cause of death in the country, resulting in more than 38,000 people losing their lives each year.

In our project we want to prevent or at least decrease the number of crashes and accidents that are because of drowsiness.

There is a large number of accidents that occurs on roads due to sleeping while driving or that the driver gets drowsy due to working for a large number of hours for long distances without sleeping or resting and as shown below that the last years the probability of sleeping while driving due to fatigue and drowsiness has increased and if the driver get drowsy for only a few seconds unfortunately he will make an accidents and make his life his family in a large danger also the surrounding cars will be in danger. So also, the probability of injury & death has increased due to this large number of accidents due to the drowsiness [25]

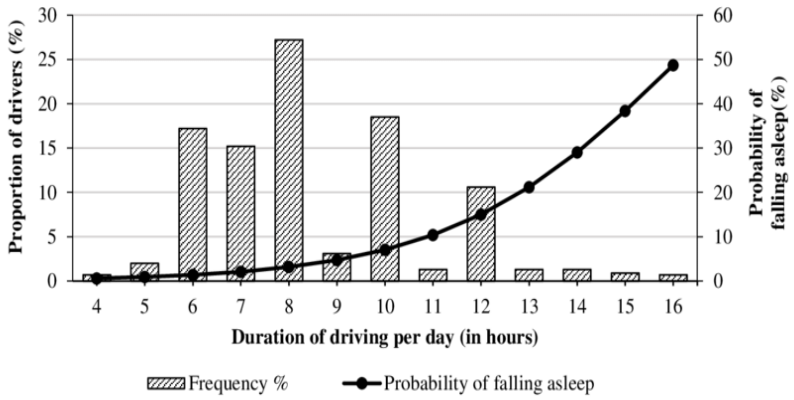


Figure 70 Car driver chart

7.2 Introduction

Our time has come, and we have made a machine learning developed system in the future Automotive safety cars and we hope that our system gets known and be implemented in all the new cars so it is wide developed in the whole world that will surely decrease the probability of accidents and protect human behind the wheel and the passengers.

We have made a safety system that can be easily integrated in the cars system that aims to prevent the accidents that occurs because of the sleeping and of course to protect the human being foremost.

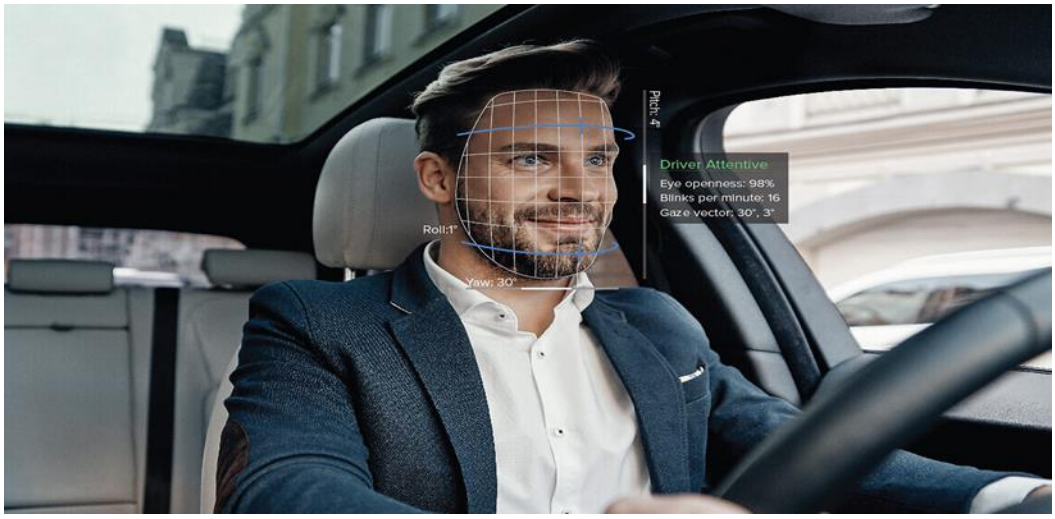


Figure 71 Face detection

7.2.1 System Main Components

- Raspberry Pi 4
- Camera
- Speaker
- Power Supply

7.2.1.1 Raspberry Pi 4

Raspberry Pi 4 is the most important component; it is the processor or the brain of the system. It organizes all the data and control signal and make all components interact with each other. Based on some algorithms and techniques that'll be discussed more specifically, we sound an alarm for the driver when he is danger.

7.2.1.2 Camera

The camera is directed all the time towards the driver face to monitor and capture his physical state through his facial expressions. It is the input of our system that takes video frames and passes it to the processor in order to be analysed and check if the driver is sleeping or showing any sleepy behaviour.

7.2.1.3 Speaker

The speaker is the output; the system sounds an alarm through the speaker if the driver is drowsy or sleeping.

7.2.1.4 Power Supply

The power supply is used to boot up and initialize the system.

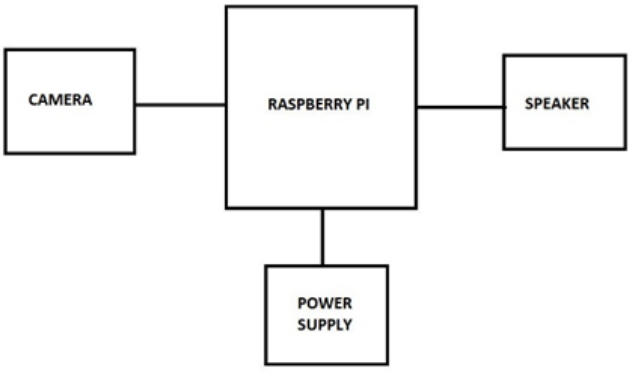


Figure 72 Drowsiness system block Diagram

7.3 Drowsiness System Operation

The system is operated through 3 stages that will be explained.

We use Driver eye/face capturing technology starting when the user starts up the car and begin his journey and ending when the user reaches his destination safely.

This technology is based on detecting the driver's face specially the eyes and the mouth with a camera and based on some algorithms and calculations the system can detect if the driver is awake and in good situation or his eyes is closed and he is sleepy, so the system alarms him. Even if the driver is yawning and It's a very natural response to being drowsiness, tiredness, or fatigue. So, we take the yawning as evidence that he is in his way to sleep so it is also a danger situation.

So, in all this cases the system must detect this situation and take a fast and correct decision to protect the life of the driver or even the passengers in the car.

The system is provided with an alarming speaker that immediately produce the driver an alarm to warn him in both situations whether his eyes is closed, or he is yawning.

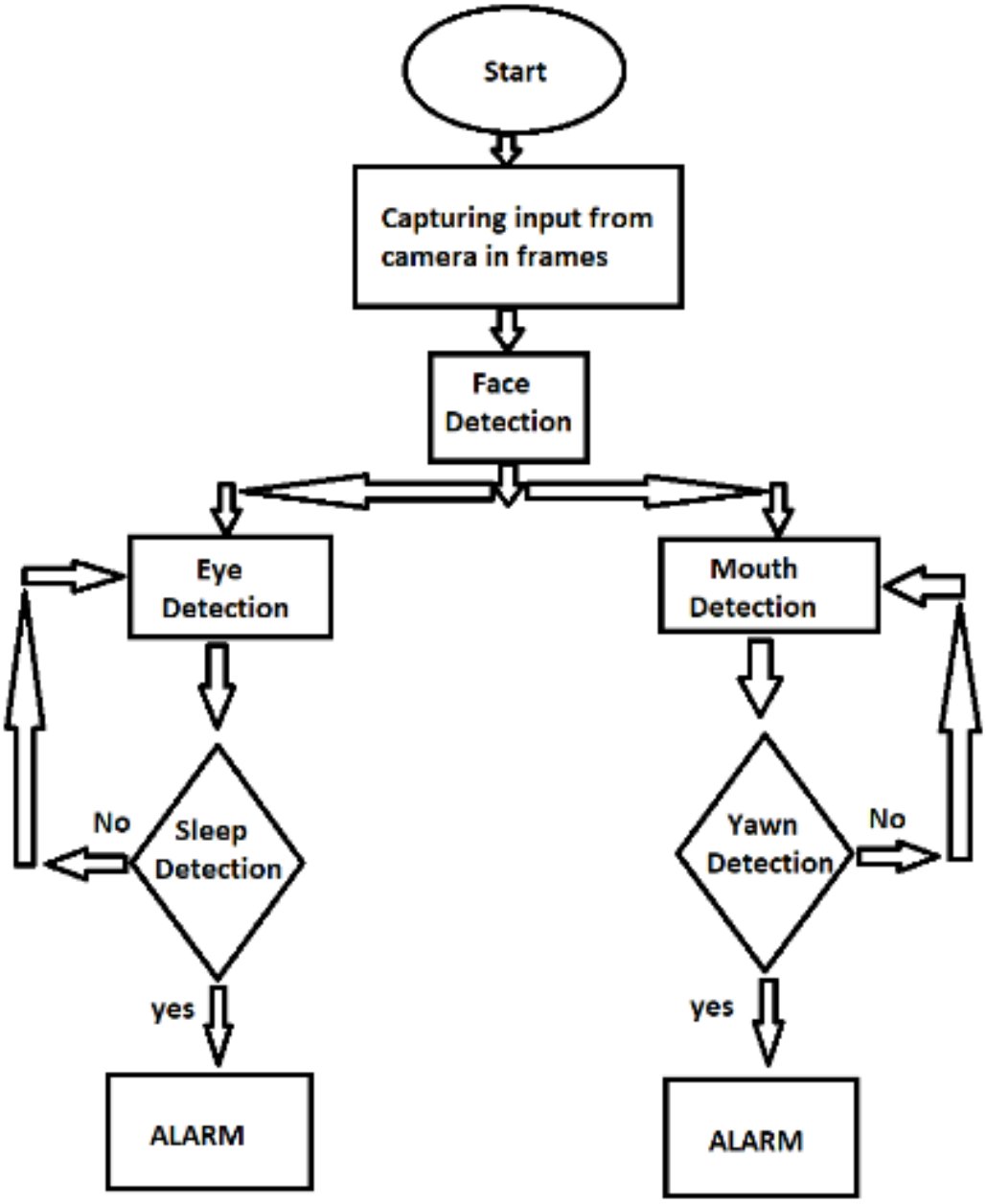


Figure 73 Drowsiness detection flow chart

7.3.1 First Stage (Detecting Unit)

The detecting unit is the Camera. The Camera takes video and send it to the processor, and this takes us to the next stage

7.3.2 Second Stage (Processing Unit)

The processing unit is the Raspberry Pi. It checks if there is any problem with the driver facial state in case he is yawning or had his eyes close. It moves to the next stage. If he isn't yawning or his eyes are closed, it goes back in a loop until something happen.

7.3.3 Third Stage (Alarming Unit)

The Alarming unit is the speaker. It receives signal from the processing unit if there the driver closes his eyes or yawning. If not, it loops back until something happen.

7.4 Operation

Science and technology are improving and improving every day, face detection and recognition turned from a science fiction to reality, and being used on daily basis for many applications, verification of identity, Mobile phone unlocking system, And finally Alipay's face-brushing tech. These applications all have to go through face detection and recognition. There is a huge variety of technology these days, face detection and recognition of all technologies became very close to us, as we use it every. Face detection and recognition, makes our life easier, and everything is faster and smoother

Face detection is performed to determine the position of the face in the picture

7.4.1 OpenCV

OpenCV Method is a common method in face detection. Computers doesn't understand our faces, they've no clue about it, that's why we use image processing. Image processing helps the computer to understand what this image is about. Image processing processes the input which could be video of photo, while the output is a set of characteristics based on the imgae.

It firstly extracts the features of the image and enlarge it into a large sample set by extracting the face Haar features in the image and then uses AdaBoost algorithm as the face detector. [26]

7.4.2 Haar Cascade

Haar cascades, first introduced by Viola and Jones in their seminal 2001 publication, Rapid Object Detection using a Boosted Cascade of Simple Features, are arguably OpenCV's most popular object detection algorithm. One of the primary benefits of Haar cascades is that they are just so fast — it's hard

to beat their speed and scale in an image. Furthermore, this algorithm can run in real-time, making it possible to detect objects in video streams.

Specifically, they focus on detecting faces in images. Still, the framework can be used to train detectors for arbitrary "objects," such as cars, buildings, kitchen utensils, and even bananas.

While the Viola-Jones framework certainly opened the door to object detection, it is now far surpassed by other methods, such as using Histogram of Oriented Gradients (HOG) + Linear SVM and deep learning. We need to respect this algorithm and at least have a high-level understanding of what's going on underneath the hood.

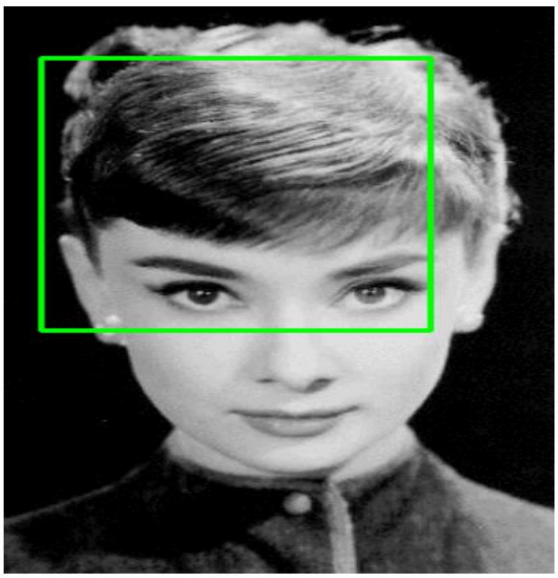


Figure 74 Eye Detection

In the shown Figure, we can see that we are sliding a window across our image at multiple scales. At each of these phases, our window stops, computes some features, and then classifies the region as Yes, this region does contain a face, or no, this region does not contain a face.

This requires a bit of machine learning. We need a classifier that is trained in using positive and negative samples of a face:

- Positive data points are examples of regions containing a face
- Negative data points are examples of regions that *do not* contain a face

Given these positive and negative data points, we can "train" a classifier to recognize whether a given region of an image contains a face.

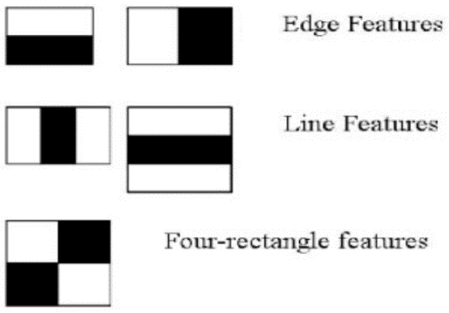


Figure 75 OpenCV

OpenCV can perform face detection using a pre-trained Haar cascade. This ensures that we do not need to provide our own positive and negative samples, train our own classifier, or worry about getting the parameters tuned exactly right. Instead, we simply load the pre-trained classifier and detect faces in images.

OpenCV is doing something quite interesting. For each of the stops along the sliding window path shown above, there is five rectangular features are computed as you see in this figure 75

If you are familiar with wavelets, you may see that they bear some resemblance to Haar basis functions and Haar wavelets (where Haar cascades get their name).

Haar-like features are digital image features used in object recognition. They owe their name to their intuitive similarity with Haar wavelets and were used in the first real-time face detector. Paul Viola and Michael Jones adapted the idea of using Haar wavelets and developed the so-called Haar-like features.

To obtain features for each of these five rectangular areas, we simply subtract the sum of pixels under the white region from the sum of pixels under the black region. Interestingly enough, these features have actual real importance in the context of face detection:

1. Eye regions tend to be darker than cheek regions.

2. The nose region is brighter than the eye region.

Therefore, given these five rectangular regions and their corresponding difference of sums, we can form features that can classify parts of a face.

Then, for an entire dataset of features, we use the AdaBoost algorithm to select which ones correspond to facial regions of an image.

However, as you can imagine, using a fixed sliding window and sliding it across every *(x, y)*-coordinate of an image, followed by computing these Haar-like features, and finally performing the actual classification can be computationally expensive.

To combat this, Viola and Jones introduced the concept of cascades or stages. At each stop along the sliding window path, the window must pass a series of tests where each subsequent test is more computationally expensive than the previous one. If any one test fails, the window is automatically discarded.

From the advantages of Haar cascade benefits are that they’re very fast at computing Haar-like features due to the use of integral images. They are also very efficient for feature selection through the use of the AdaBoost algorithm.

The integral images: is a data structure and algorithm for quickly and efficiently generating the sum of values in a rectangular subset of a grid. In the image processing domain, it was introduced to computer graphics in 1984 by Frank Crow for use with mipmaps. In computer vision it was popularized by Lewis and

then given the name "integral image" and prominently used within the Viola–Jones object detection framework in 2001. [27], [28]

The face training set that was used consists of 4916 positive image that contain faces they have been chosen randomly from the world wide web, while the negative images are 5944 image that doesn't contain any faces and are chosen also randomly from world wide web.

The testing data set that viola and johnes used to test their system is MIT+CMU frontal face test set that contain 130 image that contain 507 faces. [27]

Ok now, but how we you use Haar cascades with OpenCV?

The OpenCV library provide a group of pre-trained Haar cascades. Most of these Haar cascades are used for eyes, mouth, frontal face, full/partial body and plate number detection.

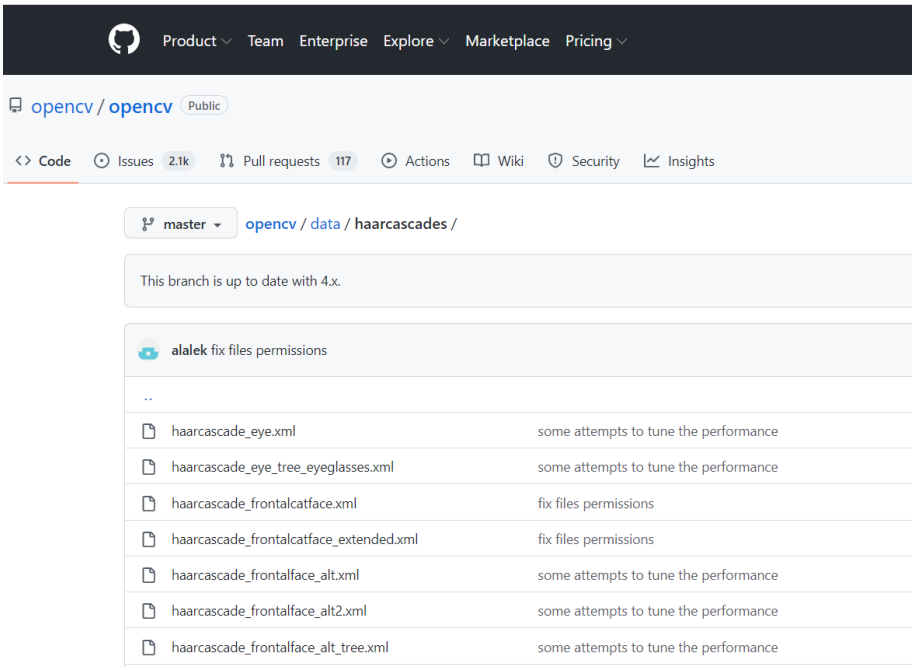


Figure 76 OpenCV library

You download any of these provided trained Haar cascades and we can load any pre-trained Haar cascade from the disk using the cv2.CascadeClassifer function.

7.4.3 Python Packages

- VideoStream
- Argparse
- Imutils
- Time
- Cv2
- Os

We must import our required Python packages. We need VideoStream to access our webcam, argparse for command line arguments, imutils for our OpenCV convenience functions, time to insert a small sleep statement, cv2 for our OpenCV bindings, and os to build file paths, agnostic of which operating system you are on (Windows uses different path separators than Unix machines, such as MacOS and Linux).

Imutils is a package based on OpenCV. It can easily realize any operations on the image. We use video as these operations will be done on a video. The video stream displays video from local stream (webcam) allows accessing the streamed video data from Python

Threading is used to execute multiple tasks at the same time, without any waiting.

NumPy is a library used for working with arrays, similar to list. However, NumPy is much faster as the arrays are stored at one continuous place in memory. NumPy helps to handle the data. And OpenCV will help us in gathering the frames from webcam and write over them. NumPy job is to handle the data brought by the Dlib

Argparse is a part of Python that allows to write CLI (command line interfaces) for your code. CLI connects user to a computer program or operating system.

Import time is essential as it helps representing time in code.

7.4.4 Dlib Library

The Dlib is used for detecting the facial landmarks and extract features using pre-trained face landmark predictor. It is used to detect 68 landmark points on the face, and it is trained on I-Bug 300-W dataset.

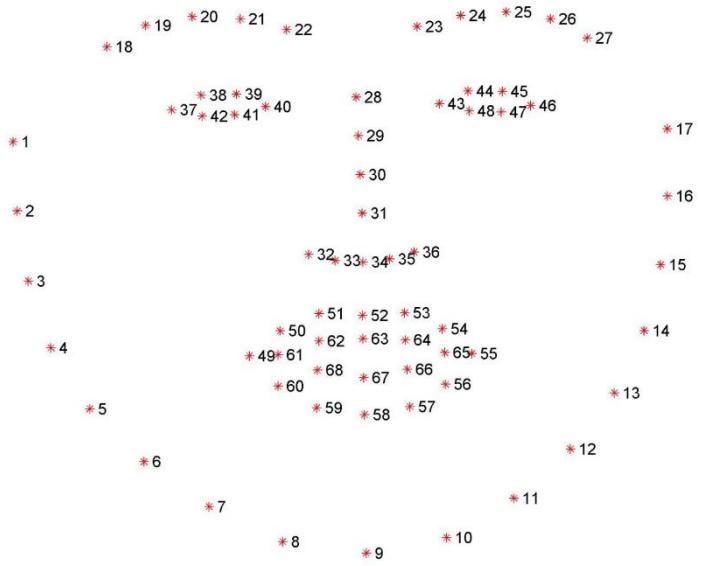


Figure 77 Dlib Points

The I-Bug 300-W It is a dataset consists of 7764 images the researchers have manually labelled the 68-point (x, y) coordinates on each image of them to determine the face landmarks location on the face.

We have used (Dlib-shape predictor) function to open the pre-trained model which is "shape predictor-68-facelandmarks.dat" that has been trained on the I-Bug 300-W dataset. So, it is able to detect the 68 landmark points in a new face image.

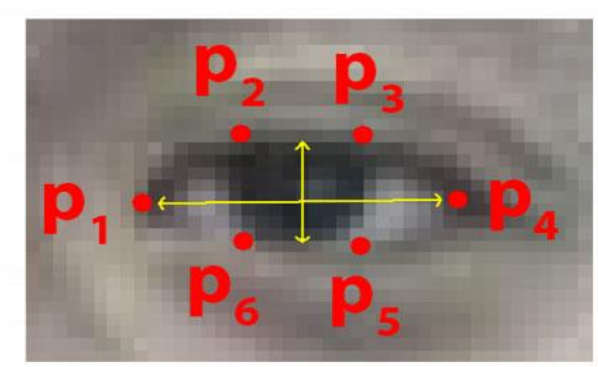
This pre-trained model is available and can be downloaded from GitHub and the owner is Davis King.

Python's Dlib library uses Kazemi and Sullivan's One Millisecond Face Alignment with an Ensemble of Regression Trees to implement this feature.

Now it's the time for detecting the eyes and determine if it is closed or not so this introduces to us the Eye Aspect Ratio.

By using the points from (37 to 42) in the 68-point shown above. And by using the eye aspect ratio (EAR). We can detect if the eye of the human behind the wheel is closed or not or by how present his eyes is closed. [27], [28]

The Ear function we have used has been published in 2016 by Tereza Soukupova and Jan Cech in Real-Time Eye Blink Detection using Facial Landmarks paper.



$$EAR = \frac{\|p_2 - p_6\| + \|p_3 - p_5\|}{2\|p_1 - p_4\|}$$

Figure 78 Eye Aspect Ratio

The Eye Aspect Ratio is a constant value when the eye is open, but rapidly falls to 0 when the eye is closed as shown below

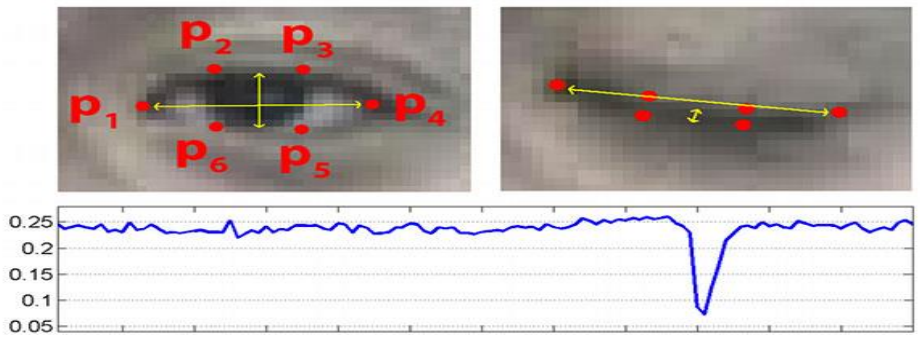


Figure 79 Eye Open and Closed

So, by using this technique and by determining a threshold value for the eye if it decreased than this threshold value then the eye is closed. by determining also, the time that his eyes are closed for, then we can surely determine that he is sleeping or not.

This threshold is not randomly determined. We don't know how much yawning too long or how much blinking is too much and based on Harvard biological database we determined this threshold.

Harvard biological database contains normal range for many things. What is important for us is the blinking and yawning. For example, a normal human being blink takes duration of 100 milliseconds to 400 milliseconds. The system was oriented with these normal ranges in order to sound an alarm when the input is by far different than the normal range. [29]

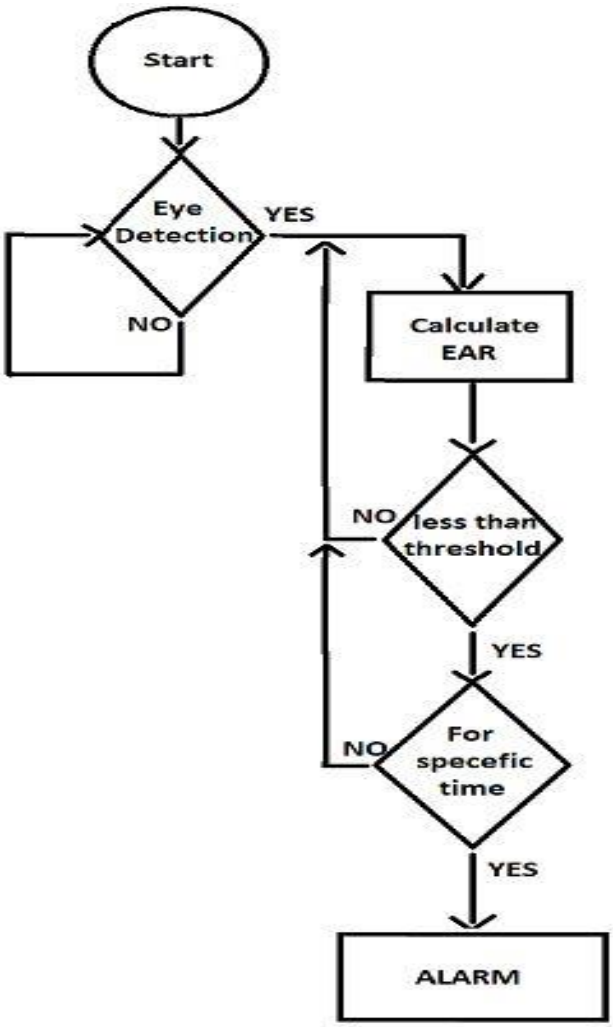


Figure 80 Detection Algorithm

Also, by using the points that determine the mouth we can determine the distance between these points if it exceeds a certain limit then we can say that he is yawning but below this limit no problem as the user may open his mouth while talking to anyone in the car so there is no problem in this case.

All these conditions we considered it as by testing we found that there are many scenarios that may occur so after many testing this code was the most reliable and give the most accuracy for the system.

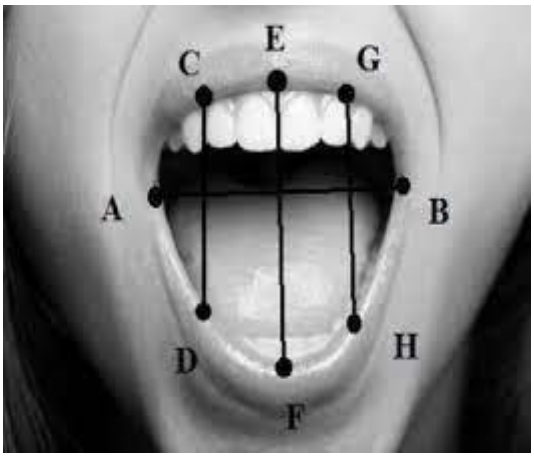


Figure 81 Mouth points

We have used also Espeak function: it is a module used to convert text to speech or audio, it is a compact open-source software speech synthesizer for English and other languages, for Linux and Windows.

We control the output of the speaker (the alarm) to be as we want, in our project when the driver gets drowsy and about to fall asleep the system alarms him by "YOU ARE SLEEPING, PLEASE GET OFF THE ROAD SIR" and when he is yawning, we alarm him by "TAKE SOME FRESH AIR SIR".

We have companied many libraries and functions in our project like OpenCV, Dlib, eye aspect ratio (EAR), Espeak and other individual functions and make them work together to be able to make one useful project is used to protect human life and to prevent accidents and crashes as possible as we could. [30]

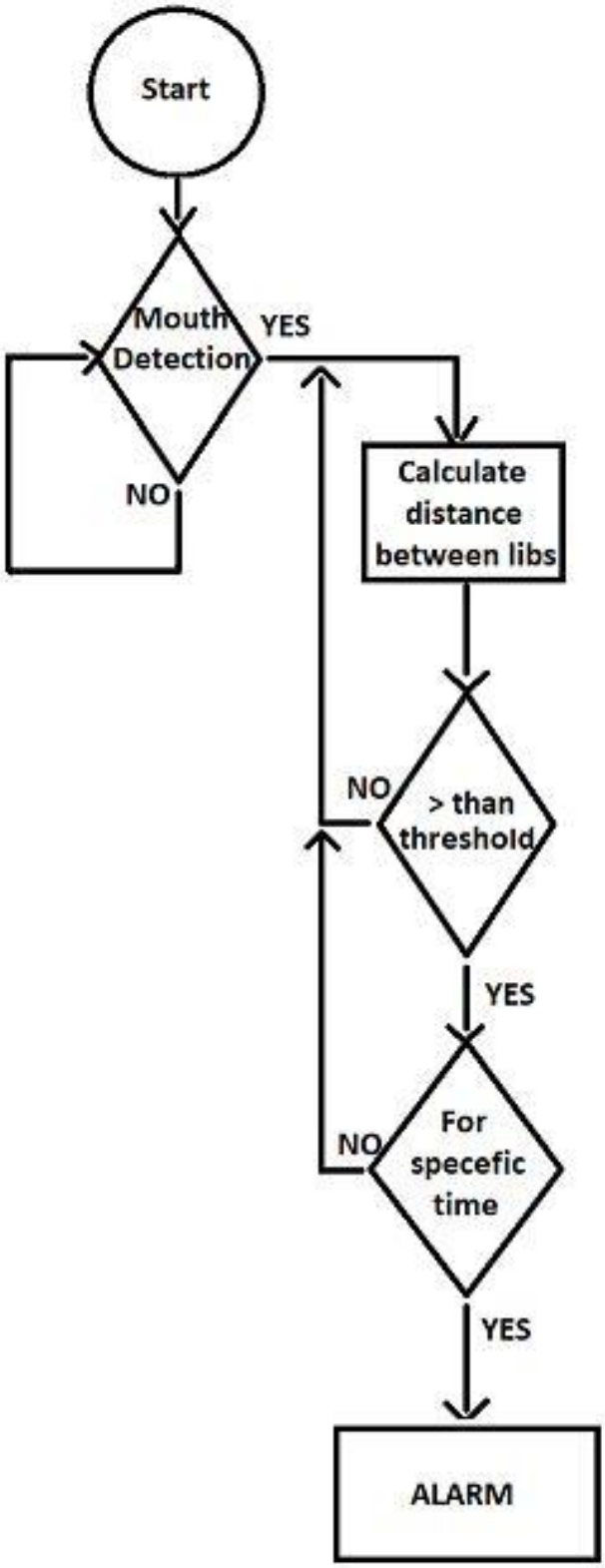


Figure 82 Yawning Algorithm flowchart

7.5 Challenges & Future developments

7.5.1 Challenges

- At first, we were using a natural human voice in alarming the driver by using a function called google text to speech (GTTS) that convert a given text into a human voice. but it caused a delay in the alarming unit. As it converts the text that is encoded into an mp3 file and store in the hard disk and after detecting the user sleeping or yawning, the system loads the alarm from the hard disk and play it. This process takes more time. But as every second matter, we have used espeak function that is much faster but it is a robotic voice instead.

- We had in the beginning some challenges in installing some libraries and functions on raspberry pi as all installing processes is done on command line (cmd) not Graphical user interface (GUI) which is much easier.

7.5.2 Future development

- The system has some defects which is if the driver is wearing glasses that has reflection lenses it may make a little misunderstanding as the library we used is trained on the faces without glasses, also the area of focus in this project is the face and the eyes especially and any change in this area would make some errors in detecting the eyes. **For project development** this problem can be solved by implement your own system and train it to be able to detect faces with glasses or without, but it is not easy to do it as it requires large number of pictures and many times for training.

- If the brightness of the face is not high enough there may be errors also as the system depends on the bright and dark areas on the face to detect it from the surroundings. **For project development** this problem can be solved by using a high-quality night vision camera to be able to detect faces even in a poor brightness situation or at night.

The system can be developed in the future by making the car park by itself on the side of the road if the driver is drowsy or sleeping to protect his life and the surroundings cars and people from crashes and accidents.

7.6 Testing and Results

We have tested the system in real application by setting the camera at a distance nearly equal the distance between the driving wheel and the driver (0.5 meter), the camera is connected to the raspberry pi and the speaker also is also connected to it.

As we see once the raspberry pi is powered on the camera will start capturing the face and detected that there is a face and also detected the most two important parts in the face which are the eyes and the mouth.

Figure 83 Testing Output 1

Now the user has closed his eyes and the EAR has fallen below the threshold value so the system detected it immediately and produce him an alarm to wake up.

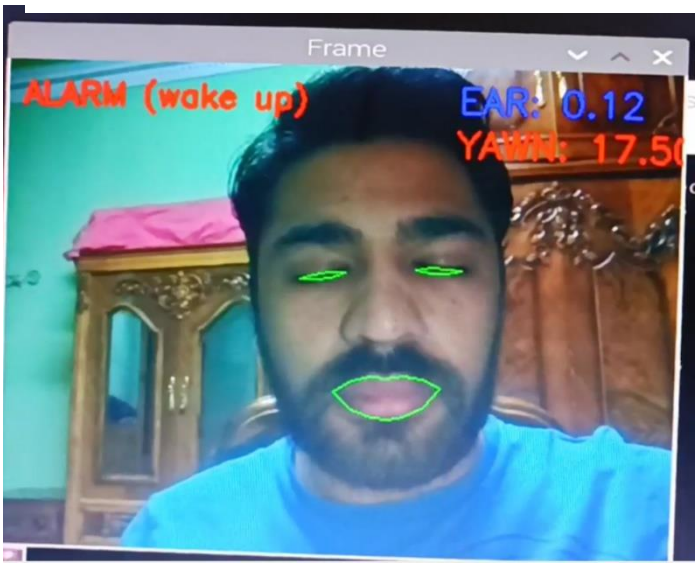


Figure 84Testing Output 2

Now the user has opened his mouth and start yawning so the distance between the upper lip and down lip has increased so the system also detected it correctly and produce him an alarm to warn him.

Figure 85 Testing Output 3

7.7 System efficiency

We have tested the system efficiency and it was very good results as the eye detection for sleeping alert is nearly 96% and for the yawning detection the efficiency was 85% but must be provided with a suitable environment and conditions which are:

1- good illumination.

2- suitable distance between camera and the user (30cm-80cm).

3- suitable camera height corresponding to the eyes.

Chapter 8
Vehicle Monitoring IOT

8.1 Introduction

Sometimes the human element must be aware of what is happening in the systems, such as the location of the vehicle and its instantaneous condition, and this will allow a lower error rate and a faster error detection and resolution.

In addition, by using (IOT) technology, any other user using the Arduino IOT application on smartphones can see the current location of the car in it, and this will be very useful in the event of theft of the vehicle or for any other purpose

8.2 IOT

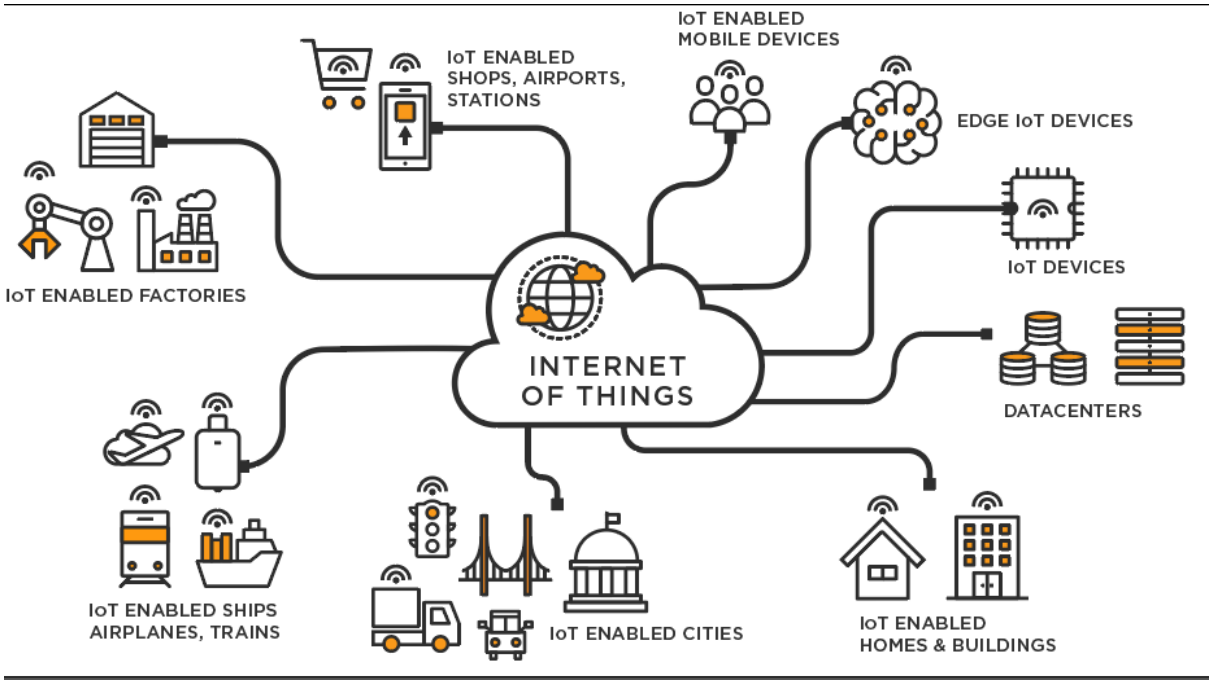


Figure 86 IOT

The internet of things, or IoT, is a system of interrelated computing devices, mechanical and digital machines, objects or people that are provided with unique identifiers (UIDs) and the ability to transfer data over a network without requiring human-to-human or human-to-computer interaction.

A thing in the internet of things can be a person with a heart monitor implant, a farm animal with a biochip transponder, an automobile that has built-in sensors to alert the driver when tire pressure is low or any other natural or man-made object that can be assigned an Internet Protocol (IP) address and is able to transfer data over a network.

Increasingly, organizations in a variety of industries are using IoT to operate more efficiently, better understand customers to deliver enhanced customer service, improve decision-making and increase the value of the business.

8.2.1 How does IoT work?

An IoT ecosystem consists of web-enabled smart devices that use embedded systems, such as processors, sensors and communication hardware, to collect, send and act on data they acquire from their environments. IoT devices share the sensor data they collect by connecting to an IoT gateway or other edge device where data is either sent to the cloud to be analyzed or analyzed locally. Sometimes, these devices communicate with other related devices and act on the information they get from one another. The devices do most of the work without human intervention, although people can interact with the devices -- for instance, to set them up, give them instructions or access the data. [31]

8.2.2 Why is the Internet of Things (IoT) so important?

By means of low-cost computing, the cloud, big data, analytics, and mobile technologies, physical things can share and collect data with minimal human intervention. In this hyperconnected world, digital systems can record, monitor, and adjust each interaction between connected things. The physical world meets the digital world—and they cooperate.

8.2.3 NodeMCU

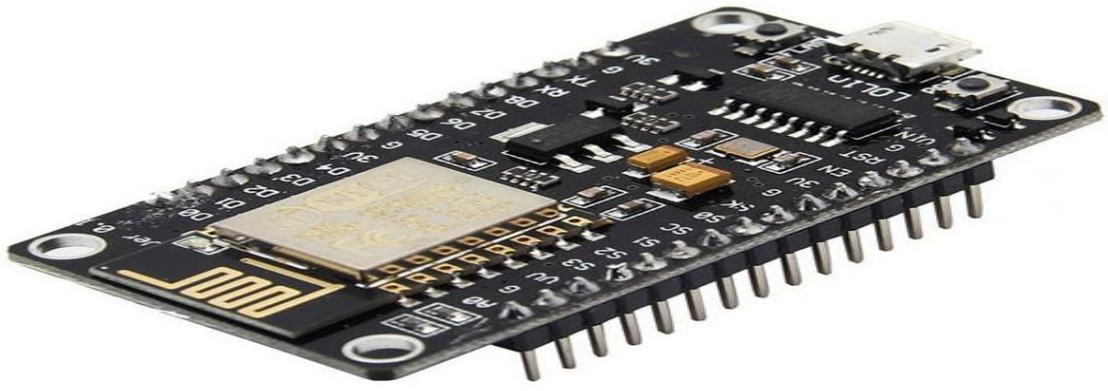


Figure 87 NodeMCU Module

The prototype is the first step in building an Internet of Things (IoT) product. An IoT prototype consists of user interface, hardware devices including sensors, actuators and processors, backend software and connectivity. IoT microcontroller unit (MCU) or development board is

used for prototyping. IoT microcontroller unit (MCU) or development board contain low-power processors which support various programming environments and may collect data from the sensor by using the firmware and transfer raw or processed data to a local or cloud-based server. NodeMCU V3 is an open-source firmware and development kit that plays a vital role in designing your own IoT product using a few Lua script lines. [32]

Multiple GPIO pins on the board allow you to connect the board with other peripherals and are capable of generating PWM, I2C, SPI, and UART serial communications.

The interface of the module is mainly divided into two parts including both Firmware and Hardware where former runs on the ESP8266 Wi-Fi SoC and later is based on the ESP-12 module.

The firmware is based on Lua - A scripting language that is easy to learn, giving a simple programming environment layered with a fast-scripting language that connects you with a well-known developer community.

And open-source firmware gives you the flexibility to edit, modify and rebuild the existing module and keep changing the entire interface until you succeed in optimizing the module as per your requirements.

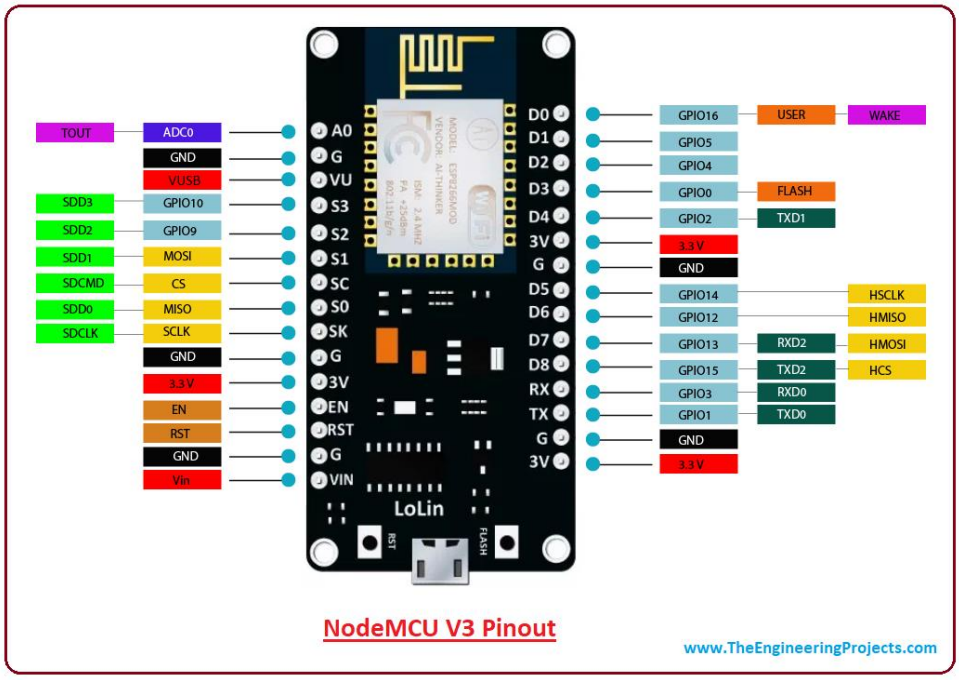


Figure 88 NodeMCU Pinout

A USB to UART converter is added on the module that helps in converting USB data to UART data which mainly understands the language of serial communication.

Instead of the regular USB port, Micro-USB port is included in the module that connects it with the computer for dual purposes: programming and powering up the board.

The board incorporates a status LED that blinks and turns off immediately, giving you the status of the module if it is running properly when connected with the computer.

8.2.3.1 NodeMCU V3 Features

- Open source
- Arduino-like hardware
- Status LED
- MicroUSB port
- Reset/Flash buttons
- Interactive and Programmable
- Low cost
- ESP8266 with inbuilt Wi-Fi
- USB to UART converter [33]

8.3 How to program NodeMCU using Arduino IDE

In order to use Arduino IDE to program the NodeMCU, you have to introduce it to the software at first.

Step1. Choose Preferences in the File menu and enter the copied code in Additional Board Manager URLs part. Then press OK.

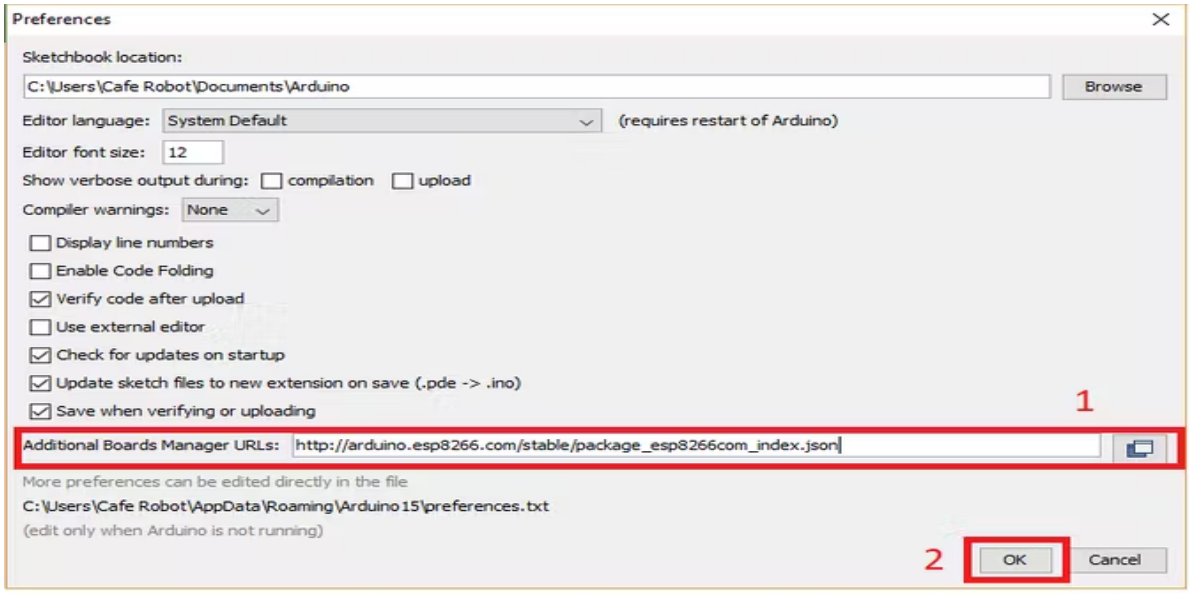


Figure 89 Include Library step 1

Step2. Search the word ESP8266 in Boards>boards manager from the Tools menu. Then install ESP8266 boards. After complete installation, you will see the INSTALLED label on ESP8266 boards.

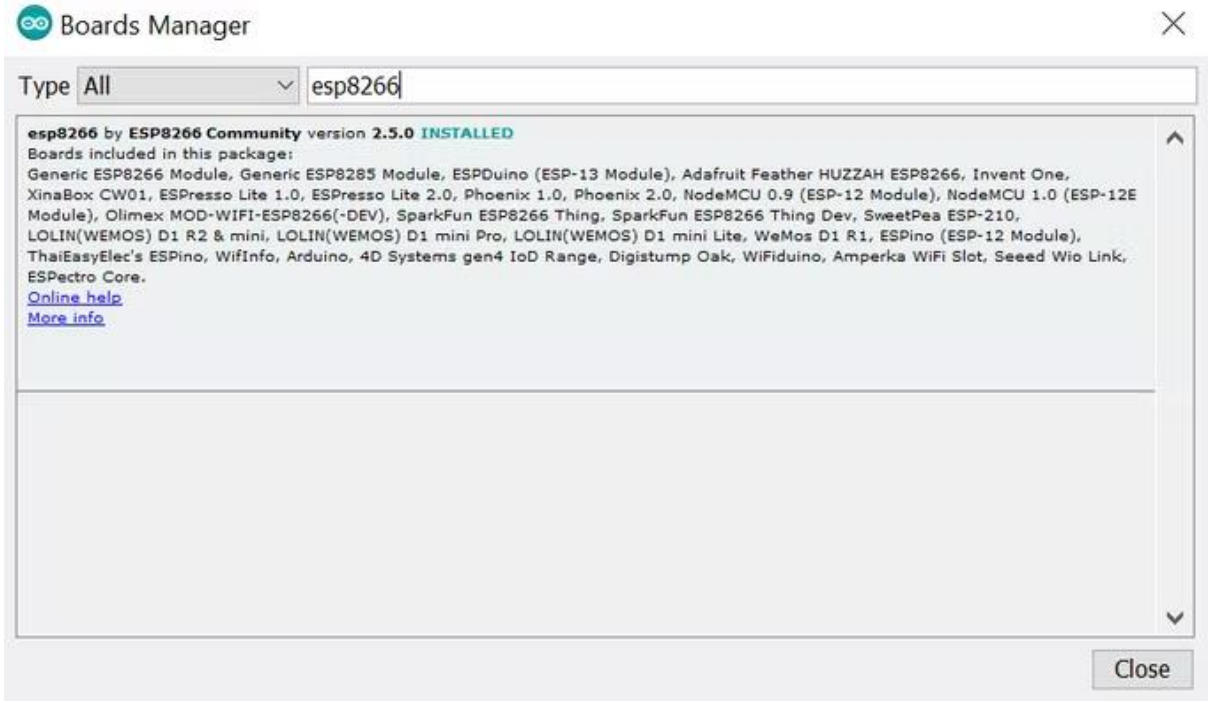


Figure 90 Include Library step 2

After these two steps, you can see ESP8266 based boards such as NodeMCU in your Arduino IDE boards list, and you can choose your desired board to upload the code. [34]

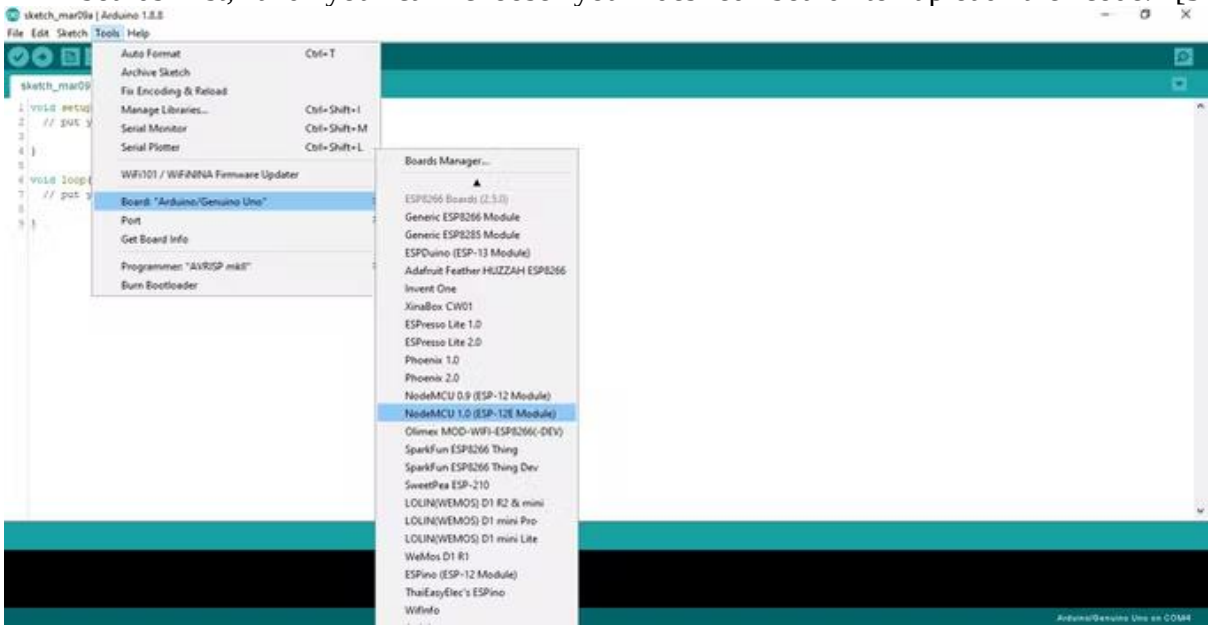


Figure 91 Include Library step 3

8.4 MQTT

MQTT (Message Queue Telemetry Transport) is a publish-subscribe messaging protocol widely used in IoT applications. This protocol is designed for data transfer between devices with limited network bandwidth and power. Thus, it is highly recommended for microcontroller projects that send data over the internet.

MQTT is a lightweight publish-subscribe-based messaging protocol.

- It is quicker (faster) than other request-response based APIs like HTTP.
- It is developed on the base of the TCP/IP protocol.
- It allows remote location devices to connect, subscribe, publish, etc. to a specific topic on the server with the help of a message broker.
- MQTT Broker/Message broker is a module in between the sender and the receiver. It is an element for message validation, transformation, and routing.
- The broker is responsible for distributing messages to the interested clients (subscribed clients) of their interested topic. [35]

8.4.1 Install required libraries

First, refer to Getting Started with NodeMCU using Arduino IDE if you have not installed NodeMCU board packages in Arduino IDE.

Here we are using Adafruit libraries for the above example. We will need to install the Adafruit IO, Adafruit MQTT, and Arduino http Client libraries using the Arduino Library Manager. [36]

Open the Arduino IDE and navigate to Sketch -> Include Library -> Manage Libraries

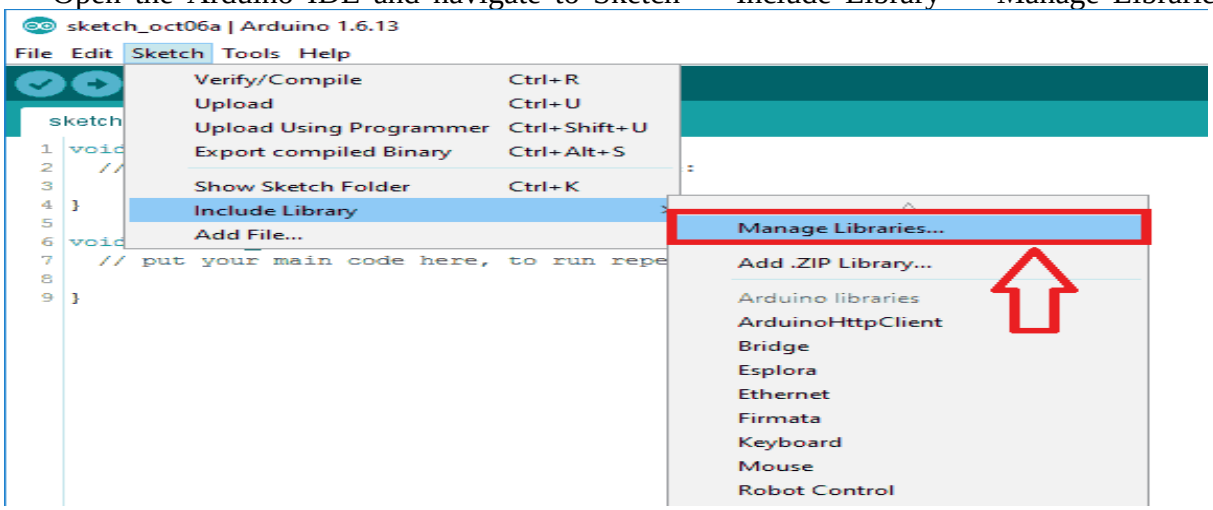


Figure 92 MQTT library

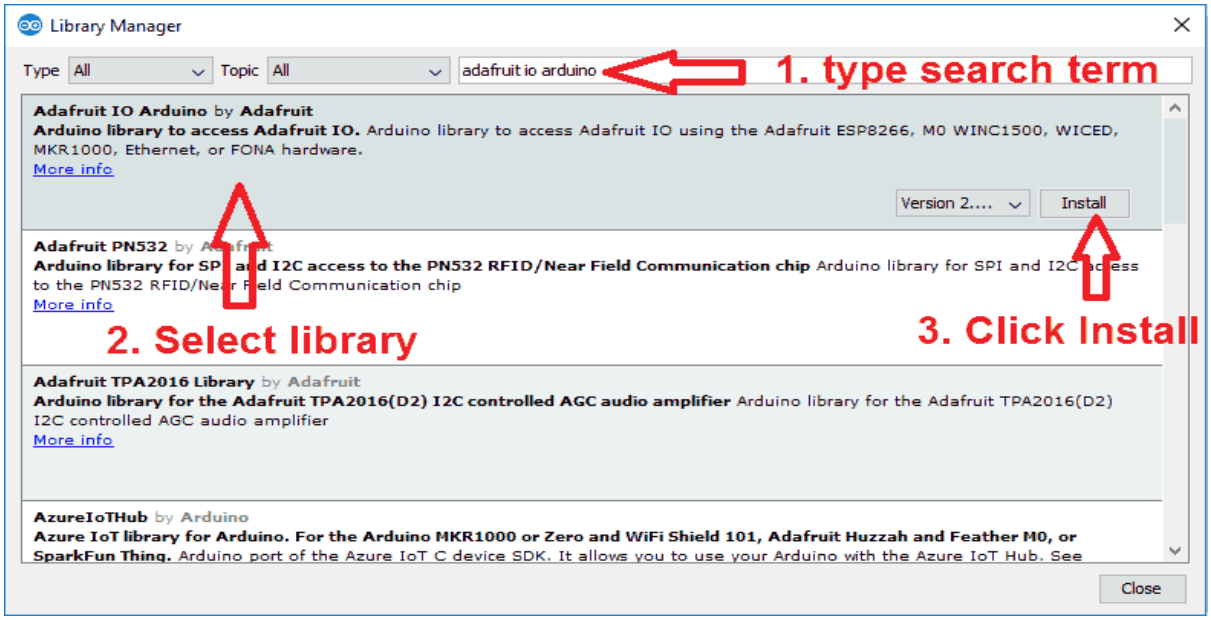


Figure 93 MQTT library

The library Manager window will pop up. Now enter Adafruit IO Arduino into the search box and click Install on the Adafruit IO Arduino library option to install version 2.6.0 or higher. Now enter Adafruit MQTT into the search box and click Install on the Adafruit MQTT library option to install version 0.17.0 or higher.

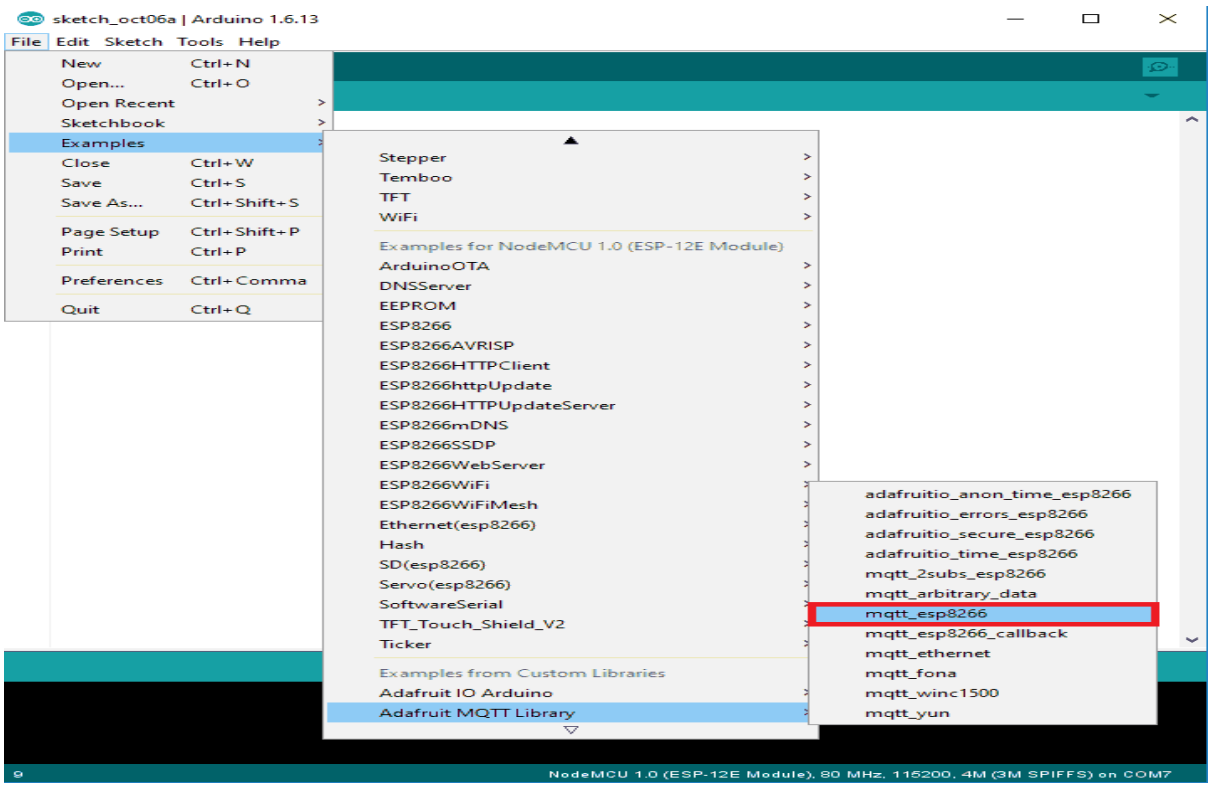


Figure 94 MQTT library

8.5 MQTT broker

An MQTT broker, which is the heart of the MQTT Publish/Subscribe protocol, is a server that receives all messages from the MQTT clients and then routes the messages to the appropriate subscribing clients.

In this project, the server used will be Arduino IOT. The Arduino IoT server is an online platform that makes it easy for you to create [37]

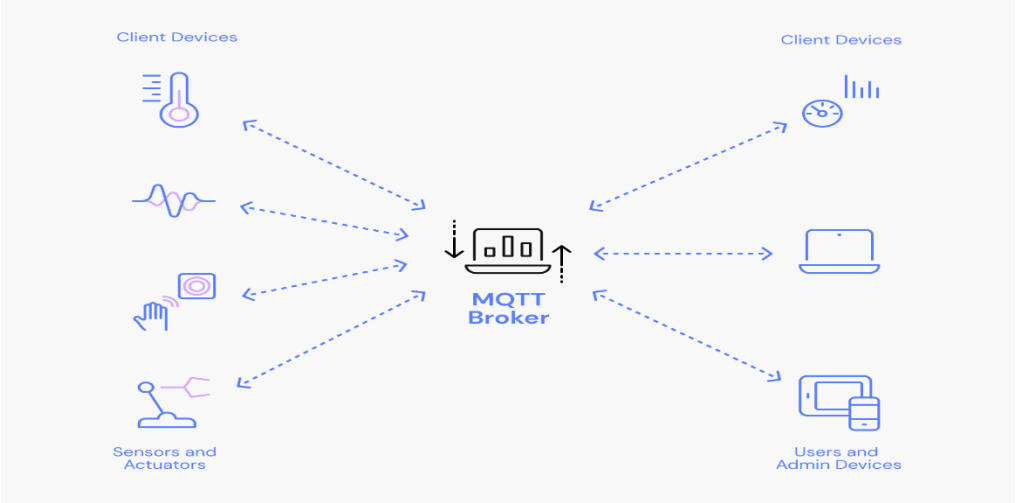


Figure 95 MQTT broker

8.6 Outputs

After knowing the parts of the system separately and how they work
It is the role of displaying the information saved on the EEPROM memory such as

1. the current vehicle speeds
2. the current location of the vehicle
3. the GPS status
4. the message sent by the GSM module

Thus, the interface will be in the "Arduino IOT" application.

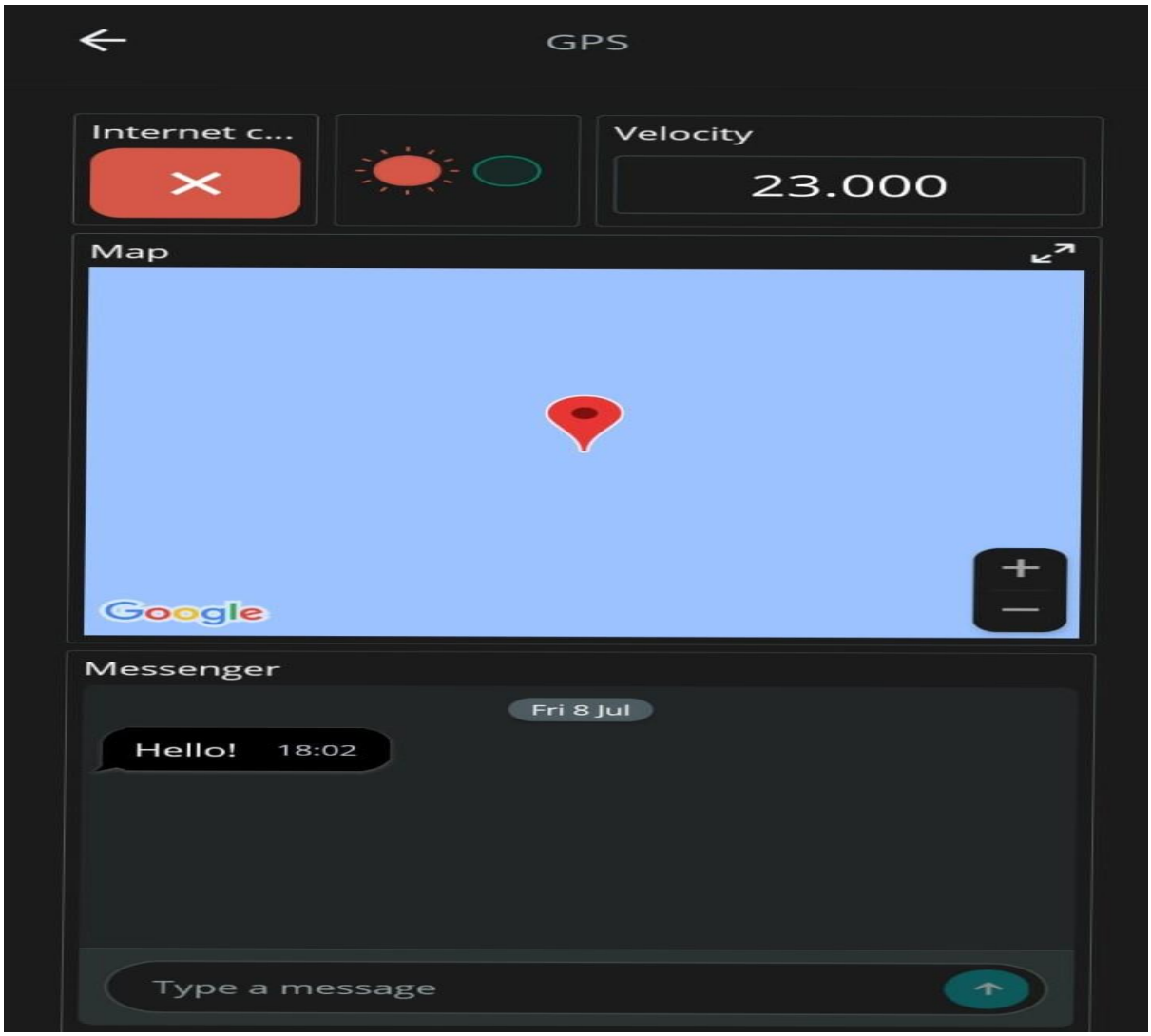


Figure 96 IOT system Output

Chapter 9
FOTA

9.1 FOTA System View

9.1.1 An Abstract View

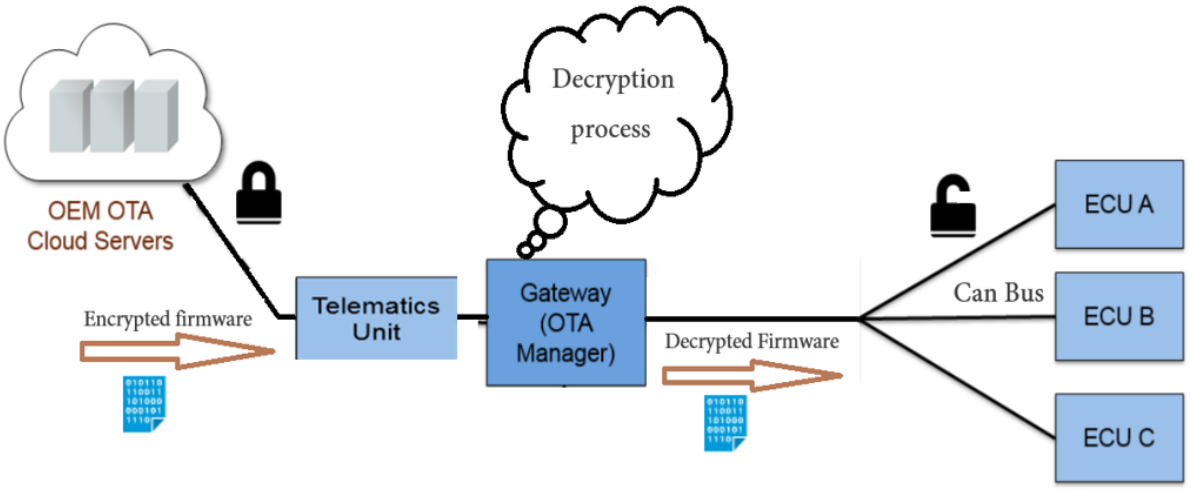


Figure 97 FOTA Block Diagram

1 - OEM cloud server

It is where we will upload the new update through a defined user interface and will have the needed information about the products and the new update.

2 - Telematics unit

It is our way to connect to server and download the code it has the capability to connect to Wi-Fi for example to has access to internet and will access the server easily.

3 - Gateway

Will receive the update from telematics unit and important data about the update like the CRC and target ECU ID and will decrypt the code before sending it to the target ECU.

4 - The target ECU

It is the ECU which the update is directed to and connected to the gateway through a communication bus like UART or CAN.

There is a fifth hidden component which is crucial for the whole process, it is the bootloader. A bootloader which supports FOTA updates must be installed in the ECU before shipping the product, the definition and mechanism of bootloader will be discussed later.

The OEM will have to a defined interface to access the server and manage the updates easily. A graphical user interface will make it easier and more reliable.

9.1.2 The View Implemented

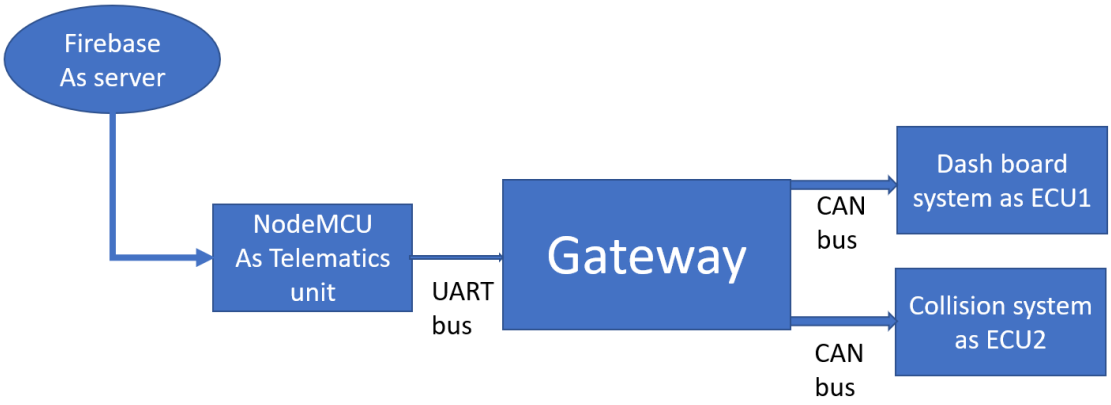


Figure 98 FOTA Implemented View

For the server firebase is used as it provides real time database and free large storage, and it is a google platform.
NodeMCU is used as a telematics unit it has a Wi-Fi module which will make it easy to connect to the server and will be connected to gateway through a UART bus.

For test cases we implemented two systems used in cars a dashboard system and collision system both ECUs is connected to CAN bus to access the gateway. And both use stm32f103 microcontroller.
Gateway also will be implemented using the same microcontroller as it has the required communication protocols, and the micro has the required processing power.

9.2 Cloud Connection

The first fundamental step to apply FOTA technology is to connect to the internet, to do so we will need some prerequisites, we first need a server to upload the update to and handle the backend services, and a GUI to interface with this server and control it more easily and to make standard interface with cloud, GUI can be accessed from any remote PC and finally the embedded device must have a hardware that can connect to internet like a Wi-Fi module or GSM module the connection must be stable and secure.

So, the three main components to accomplish a connection to the cloud are:

1. Server for that we used firebase.
2. GUI for that we implemented simple GUI using python.
3. Wi-Fi hardware for that we used NodeMCU.

9.2.1 Firebase

Firebase is a Backend-as-a-Service (Baas). It provides developers with a variety of tools and services to help them develop quality apps, grow their user base, and earn profit. It is built on Google's infrastructure.

Firebase is google platform, free and easy to use, it have a real time database.

Figure 99 Firebase logo

Firebase lets you automatically run backend code in response to events triggered by Firebase features and HTTPS requests. Your code is stored in Google's cloud and runs in a managed environment.

Firebase is categorized as a NoSQL database program, which stores data in JSON-like documents. Firebase is powerful and offers several services, including:

- Analytics – Google Analytics for Firebase offers free, unlimited reporting on as many as 500 separate events. Analytics presents data about user behavior in iOS and Android apps, enabling better decision-making about improving performance and app marketing.

- Authentication – Firebase Authentication makes it easy for developers to build secure authentication systems and enhances the sign-in and onboarding experience for users. This feature offers a complete identity solution, supporting email and password accounts, phone auth, as well as Google, Facebook, GitHub, Twitter login and more.

- Cloud messaging – Firebase Cloud Messaging (FCM) is a cross-platform messaging tool that lets companies reliably receive and deliver messages on iOS, Android, and the web at no cost.

- Realtime database – the Firebase Realtime Database is a cloud-hosted NoSQL database that enables data to be

stored and synced between users in real time. The data is synced across all clients in real time and is still available when an app goes offline.

- Performance – Firebase Performance Monitoring service gives developers insight into the performance characteristics of their iOS and Android apps to help them determine where and when the performance of their apps can be improved.

- Storage – Firebase provide large storge where we can store data like the software update to access it later and download it.

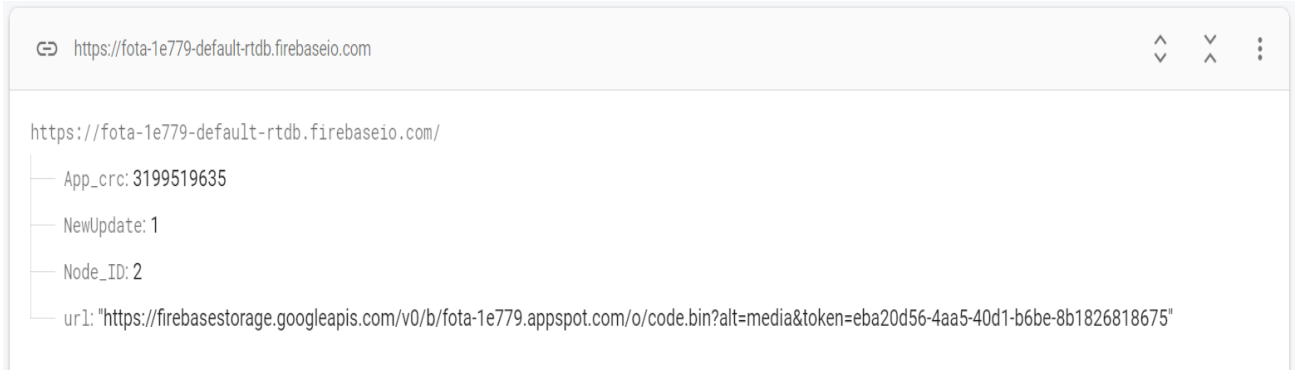


Figure 100 Real time Database

Used to store important data about the update like CRC which is used to check the validity of the code, the ECU ID in case the system has many ECUs, a flag to notify the device if there is a new update and the download link of the update, these variables will be changed through the GUI without the need to access the database directly through the firebase interface.

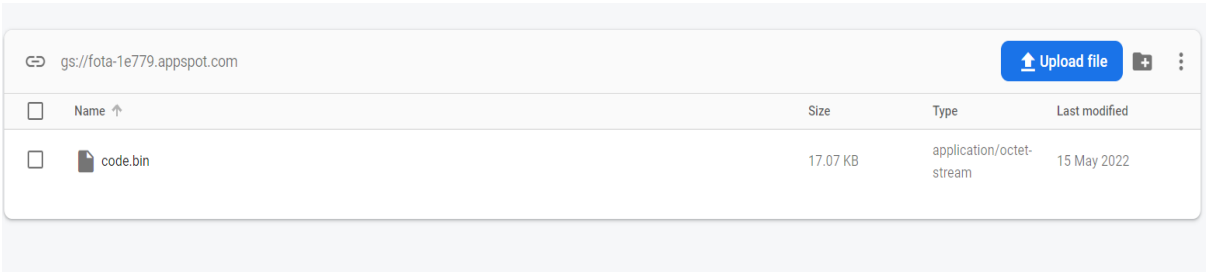


Figure 101 Firebase Storage

Used to store the update itself to download it later in the embedded device we can upload to it directly through the firebase interface or through the GUI provided to access the firebase to abstract the backend work.

9.3 GUI

Used to abstract the backend implementation with firebase and create standard interface to deal with the update, GUI is implemented using python language.

Python is an interpreted, object-oriented, high-level programming language with dynamic semantics. Its high-level built-in data structures, combined with dynamic typing and dynamic binding, make it very attractive for Rapid Application Development, as well as for use as a scripting or glue language to connect existing components together. Python's simple, easy to learn syntax emphasizes readability and therefore reduces the cost of program maintenance.

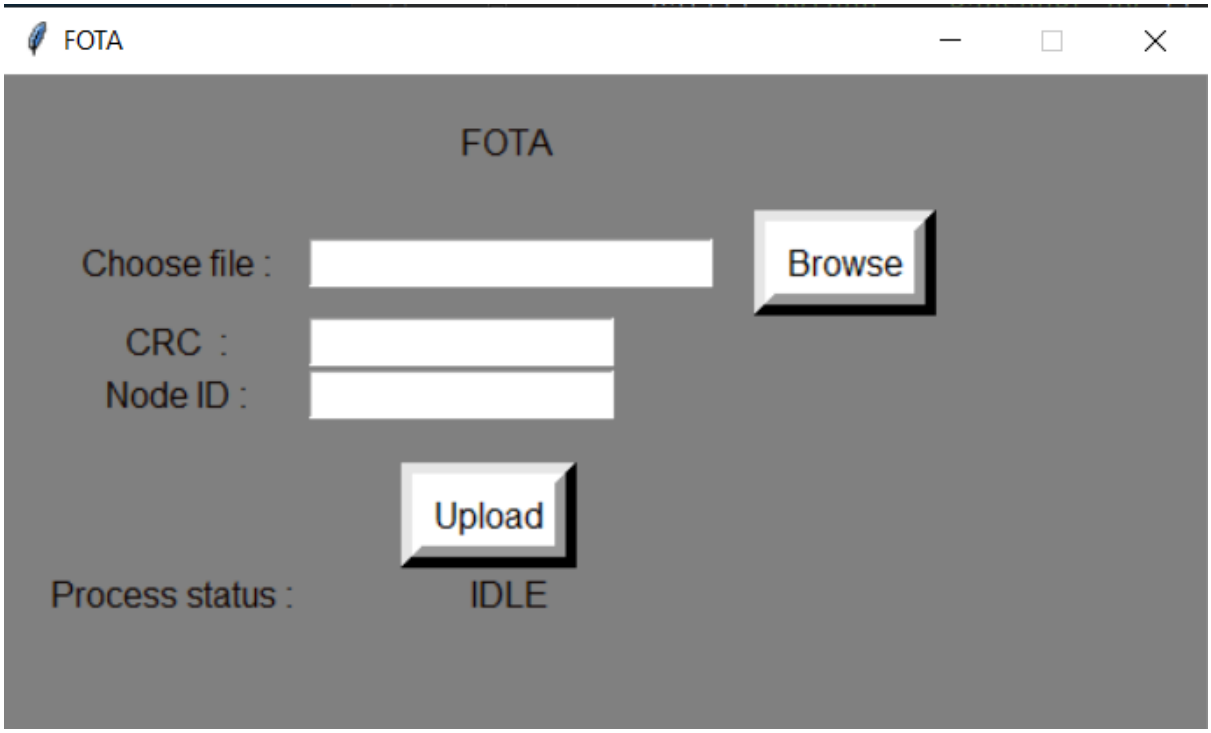


Figure 102 FOTA GUI

We enter three important fields first the code file and must be in binary form, the binary form is efficient and less in size than other forms, second field is CRC of the code and lastly the node ID.

After we enter all the fields, we press the upload file and status bar indicates the progress of the update or block the update and if the server is busy with another update.

9.4 NodeMCU

It is an open-source firmware and development kit that helps you to prototype or build IoT products. It includes firmware that runs on the ESP8266 Wi-Fi, and hardware which is based on the ESP-12 module. The firmware uses the Arduino IDE.

NodeMCU is an Arduino based microcontroller with additional feature of ESP8266 Wi-Fi chip with full TCP/IP stack, main advantage of Nodemcu is that it can directly connect to the internet without using any additional peripheral and can connect to cloud using HTTP or MQTT protocols to send and receive data to help IoT systems to be up to date and can monitor systems as well analyze data for the future developments.

We used NodeMCU to connect to the firebase cloud and download the update then will forward the update to the gateway through UART protocol.

To connect to the firebase NodeMCU must have a secret key to authenticate to the cloud to ensure security, and to prevent any other device to access the cloud without permeation.

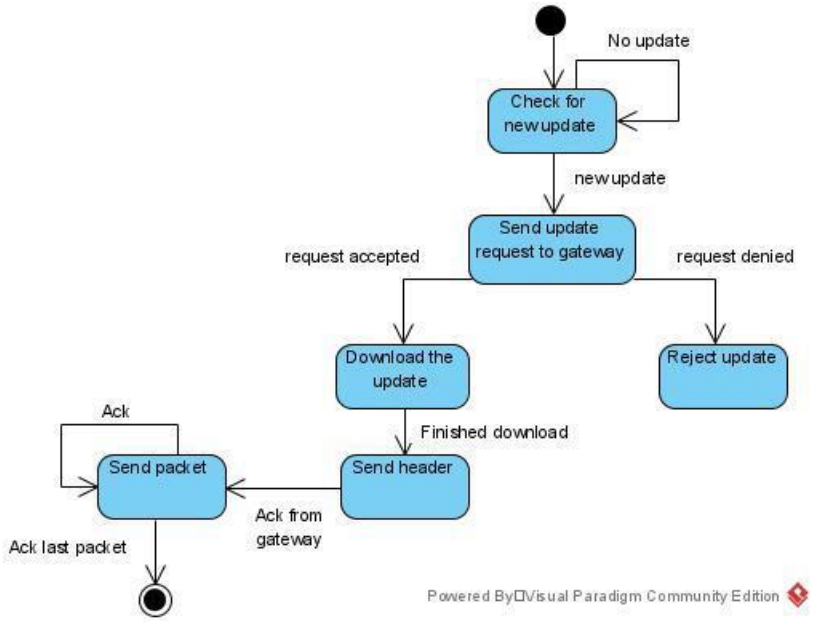


Figure 103 NodeMCU state machine

In the state machine of the sequence happening in the NodeMCU, as we see the code will not be downloaded unless the user accepts the update first, when the request is accepted the code will be downloaded and then will be sent to

the gateway the header then the packet, the header includes data about the code like its size, CRC, and ECU ID.
The code will be divided into packets and each packet will be sent after the gateway acknowledge the node till all the code is sent.
After the code is sent to gateway and verified, NodeMCU will update the cloud that the update is successful, and the device now will be ready to receive other updates.

9.5 Security

There are two main encryption methods. First, Asymmetric Encryption in which the encryption key is published for anyone to use and for encrypting messages. Only the receiving party has access to the decryption key that enables messages to be read. Public-key encryption was first described in a secret document in 1973. Before that, all encryption schemes were symmetric-key (also called private-key). Second, Symmetric Encryption in which the encryption and decryption keys are the same. Communicating parties must have the same key to achieve secure communication. We choose AES Symmetric method to encrypt the firmware.

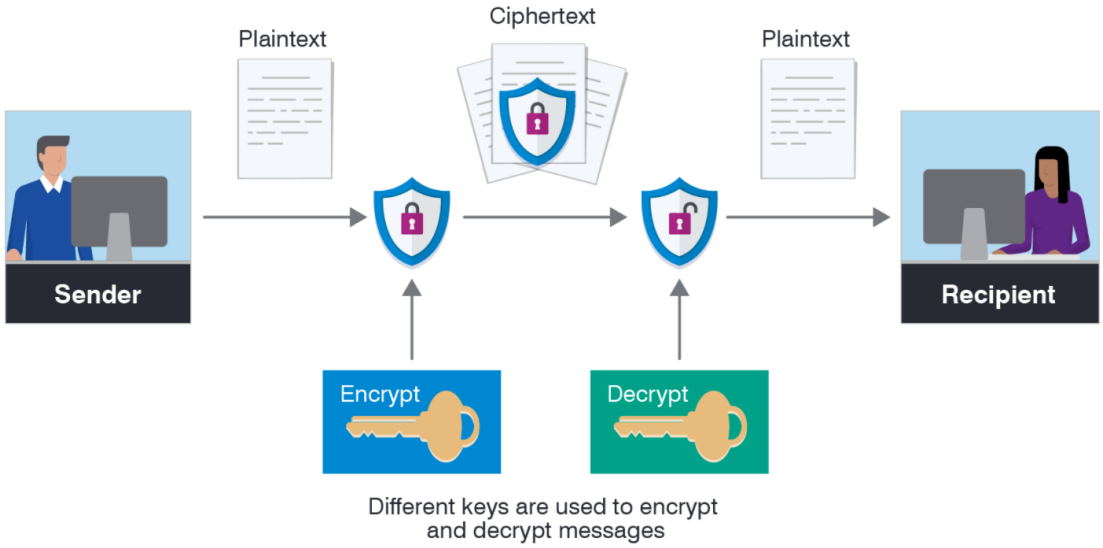


Figure 104 FOTA Security

9.6 Gateway

Gateway is the bridge between the telematics unit and the target ECUs, it is very important especially with application having many ECUs like cars, gateway receive the code from NodeMCU through UART protocol and forward it to the target ECU through a CAN bus.

A gateway is a central hub that securely and reliably interconnects and processes data across these heterogeneous networks. It provides physical isolation and protocol translation to route data between functional domains (powertrain, chassis and safety, body control, infotainment, telematics, ADAS) that share data to enable new features. Gateways allow engineers to design more robust and functional vehicle networks.

9.6.1 Gateway functions

- Save information about connected ECUs, this information includes ECUs IDs, software version of every ECU and any important data related to the connected ECUs.

- Decrypt the code before sending it to the target ECU, gateway in most cases will be more powerful microcontroller than the other ECUs so it will hold the decryption algorithm and make sure the code had been decrypted correctly before forwarding it, and like stated before decryption is very necessary step to as the code will be encrypted in the server to security reasons.

- Check if code is valid, it is important to check before sending to ECU to avoid sending buggy code or incorrect data.

- Determine which ECU this code is for, and that is one of the main tasks to make sure the code is delivered to the correct ECU, as stated before the gateway has information about all the ECUs including ECUs IDs so it can determine which ECU the update is targeted to.

- Resolve dependencies in case of rollback, and that is why the gateway save information about every ECU's software version, sometime a rollback will be needed due to the failure of the current software, and that software will be relying on other ECUs, so if this ECU will execute a rollback should

inform first the gateway to send a command to rollback all the ECUs depending on that ECU.

- Notify the node when the update is complete so the node can notify the server too that the code update is done.

9.6.2 Gateway state machine

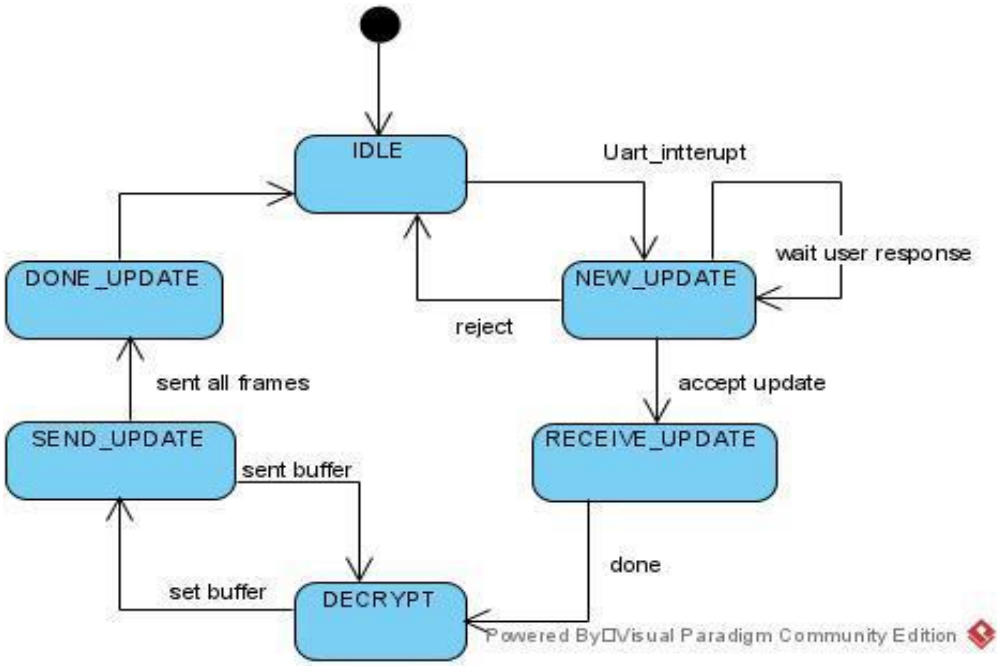


Figure 105 Gateway state machine

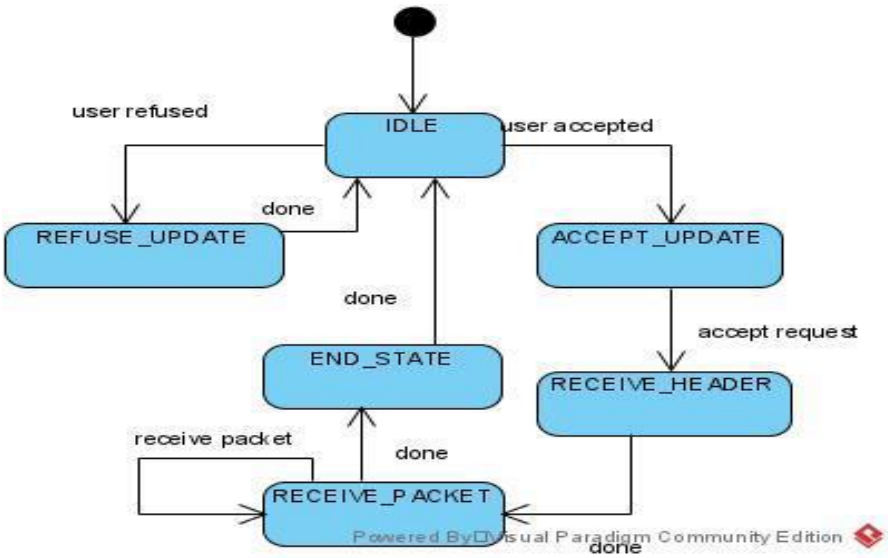


Figure 106 Gateway state machine 2

9.6.2 Gateway sequence

1. Gateway remains in idle state waiting a notification from NodeMCU in (Figure 104) we can see the TFT screen in idle state

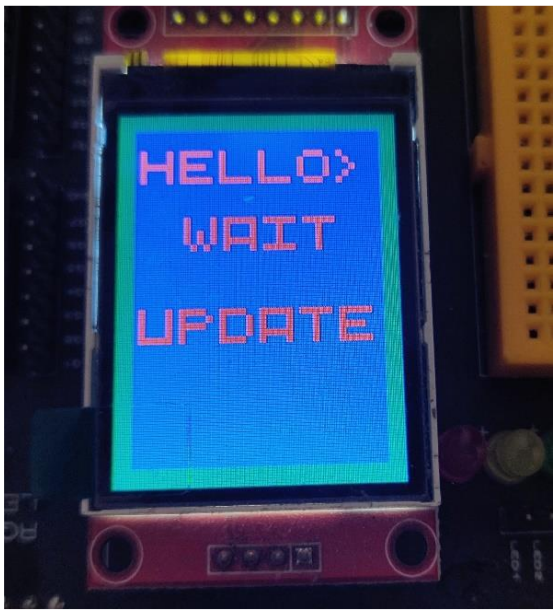


Figure 107 Gateway idle state

2. When gateway receive UART interrupt from NodeMCU indicating that a new update available the gateway will notify the user as we see in (Figure 105) the user can accept or reject the update

Figure 108 Gateway receiving state

3. If the user rejected the update the gateway will return to the idle state but if the user accepted the update the code will be downloaded

and sent to gateway and the progress will be shown like in (Figure 106).

Figure 109 Gateway Loading state

9.7 State machine

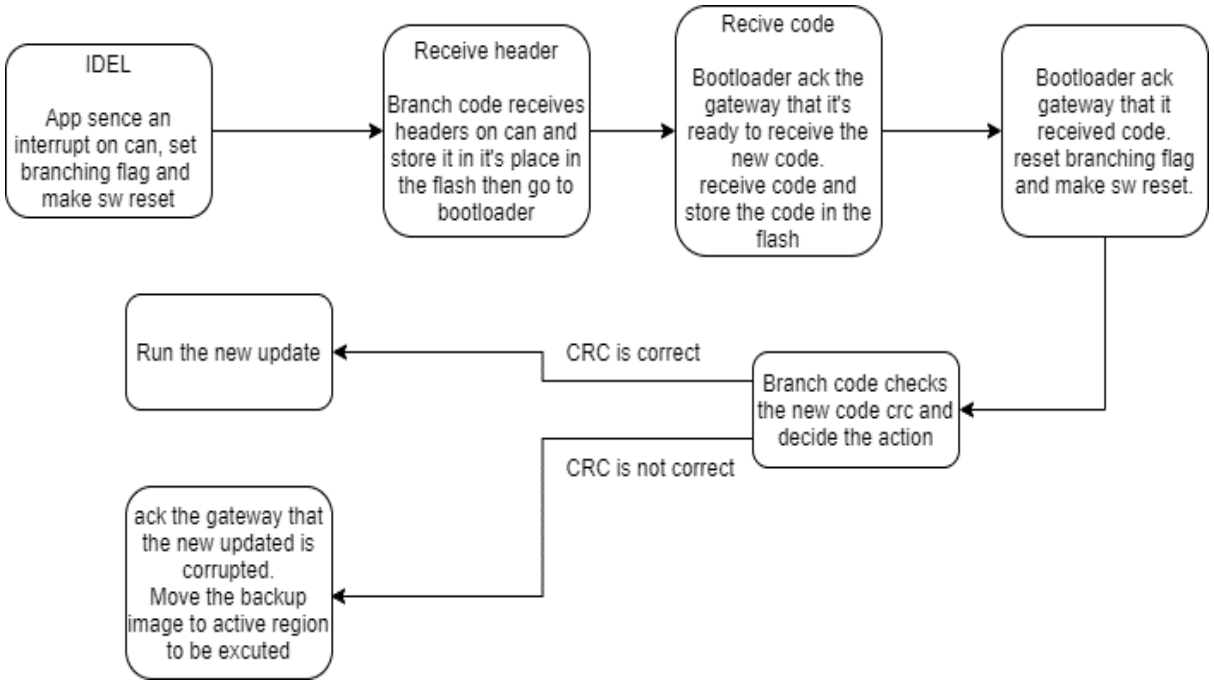


Figure 110 FOTA State machine

Appendices

Next steps

Our next steps are to reduce the hardware size and to develop the code to enhance time critical systems, to enhance driver performance analysis systems to be more accurate and precise, to develop a predictive maintenance system to predict any failure in the sensors and recommend changes to change it.
And to reduce cost and make the system more portable.

Conclusion

We can decrease the number of vehicle accidents caused by sleeping while driving and furthermore, we can save many lives and prevent many persons from being fatally injured due to wrong behaviour from observers or the delay of emergency.

Thus, the vehicle driver now has a tool to keep an eye on him and to save him from any situation, with low cost and high compatibility.

What makes our system competitive with other existing solutions is that even if these
systems exist, they mostly come within the car and these categories of cars are too expensive which makes these features not affordable for most cars, however our system is standalone and can be integrated through our integration feature.
Another competitive feature is the firmware over the air **(FOTA)** which makes the system valuable and up to date even a long time after the user buy it as long as the hardware exists the software many other features is updated and remote diagnose and support is extended.

Tools

1. STM cube: complete software development environment for a range of Arm Cortex-M based microcontroller devices, helps with debugging.

2. Arduino IDE: it is an open-source IDE makes it easy to write code and upload it to the board, used for NodeMCU.

3. Python 3.8: used to create the GUI as it is a stable release of Python.

4. Visual studio code: as an editor.

5. STM32 ST-LINK Utility: used to burn code to microcontroller.

6. Draw.io: to draw software diagrams.

7. Git & GitHub: for version control.
8. Altium design for PCB.

References

[1] https://www.electricmotorengineering.com/an-electric-motor-works-car/

[2] https://www.parvalux.com/advantages-of-dc-motors/

[3] https://lastminuteengineers.com/l298n-dc-stepper-driver-arduino-tutorial/

[4] https://howtomechatronics.com/tutorials/arduino/arduino-dc-motor-control-tutorial-l298n-pwm-h-bridge/

[5] https://www.electronicwings.com/sensors-modules/bluetooth-module-hc-05-

[6] https://www.analog.com/en/analog-dialogue/articles/uart-a-hardware-communication-protocol.html

[7] https://www.bluetooth.com/blog/how-bluetooth-technology-uses-adaptive-frequency-hopping-to-overcome-packet-interference/

[8] Carullo, Alessio, and Marco Parvis. "An ultrasonic sensor for distance measurement in automotive applications." *IEEE Sensors journal* 1.2 (2001): 143.

[9] Morgan, Elijah J. "HC-SR04 ultrasonic sensor." *Nov* (2014).

[10] Gabriel, Mutinda Mutava, and Kamweru Paul Kuria. "Arduino uno, ultrasonic sensor HC-SR04 motion detector with display of distance in the LCD." *International Journal of Engineering Research and Technical Research* 9 (2020).

[11] Prafanto, Anton, and Edy Budiman. "A water level detection: IoT platform based on wireless sensor network." *2018 2nd East Indonesia Conference on Computer and Information Technology (EIConCIT)*. IEEE, 2018.

[12] https://www.instructables.com/MPU6050-Setup-and-Calibration-Guide/

[13] http://www.geekmomprojects.com/gyroscopes-and-accelerometers-on-a-chip/

[14] https://www.u-blox.com/sites/default/files/products/documents/NEO-6_DataSheet_(GPS.G6-HW-09005).pdf

[15] https://circuitdigest.com/microcontroller-projects/interfacing-sim800l-module-with-esp32

[16] https://www.3gpp.org/

[17] SIM800L datasheet.

[18] Principles of communication: systems, modulation, and noise reference ,Author: Rodger E Ziemer & William H Tranter

[19] Transmission Techniques for Emergent Multicast and Broadcast Systems 1st Edition Author: Mario Marques da Silva & Américo M. C. Correia

[20] Understanding and Using the Controller Area Network Communication Protocol by A. Ghosal, Haibo Zeng, and Marco Di Natale.

[21] Computation of CAN Bit Timing Parameters Simplified Meenanath Taralkar, OTIS ISRC PVT LTD, Pune, India.

[22] The configuration of CAN Bit Timing BY: 6th International CAN Conference 2nd to 4th November, Turin (Italy).

[23] BOSCH CAN Specification.

[24] https://www.cdc.gov/sleep/about_sleep/drowsy_driving.html

[25] https://www.sleepfoundation.org/drowsy-driving

[26] Mahamkali, Naveenkumar & Ayyasamy, Vadivel. (2015). OpenCV for Computer Vision Applications.

[27] Viola, Paul, and Michael Jones. "Rapid object detection using a boosted cascade of simple features." *Proceedings of the 2001 IEEE computer society conference on computer vision and pattern recognition. CVPR 2001*. Vol. 1. Ieee, 2001.

[28] Mehtab, S., and J. Sen. "Face Detection Using OpenCV and Haar Cascades Classifiers. Mar. 2020." *ECC: No Data (logprob:-57.959)*.

[29] Kazemi, Vahid, and Josephine Sullivan. "One millisecond face alignment with an ensemble of regression trees." *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2014.

[30] Cech, Jan, and Tereza Soukupova. "Real-time eye blink detection using facial landmarks." *Cent. Mach. Perception, Dep. Cybern. Fac. Electr. Eng. Czech Tech. Univ. Prague* (2016): 1-8.

[31] https://www.oracle.com/eg/internet-of-things/what-is-iot

[32] https://nodemcu.readthedocs.io/en/master

[33]https://www.researchgate.net/publication/328265730_NodeMCU_V3_For_Fast_IoT_Application_Development

[34] https://create.arduino.cc/projecthub/electropeak/getting-started-w-nodemcu-esp8266-on-arduino-ide-28184f

[35] https://www.opc-router.com/what-is-mqtt

[36] https://www.electronicwings.com/nodemcu/nodemcu-mqtt-client-with-arduino-ide

[38] https://docs.arduino.cc/tutorials/uno-wifi-rev2/uno-wifi-r2-mqtt-device-to-device