Faculty of Engineering – Zagazig University
Mechatronics Engineering Program

Graduation Project 2023

Advanced driver-assistance system (ADAS)

Submitted by

- **Ali Adel Mohammed Ali** **20812018100295**
- **Mohamed Osama Elsayed** **20812018101762**
- **Mohamed Alaa Ahmed** **20812018101005**
- **Ahmad Alaa Mohammed Farouk** **20812018100489**
- **Mostafa Magdy Hassanien** **20812018101701**
- **Ahmad Yasser Nafaz** **20812018101317**

Mechatronics Engineering Program

Supervised by:

Dr. Mohamed Atia Abdelatief
Dr. Ahmed Reda Abdlmonem

كلية الهندسة – جامعة الزقازيق برنامج هندسة
الميكاترونيات – بنظام الساعات المعتمدة
Faculty of Engineering – Zagazig University Mechatronics Engineering Program

Advanced driver-assistance system (ADAS)

Graduation Project 2023
Mechatronics Engineering Program

Submitted by

- **Ali Adel Mohammed Ali**
- **Mohamed Osama Elsayed**
- **Mohamed Alaa Ahmed**
- **Mostafa Magdy Hassanien**
- **Ahmad Yasser Nafaz**
- **Ahmad Alaa Mohammed Farouk**

Supervised by:

Dr. Mohamed Atia Abdelatief

Dr. Ahmed Reda Abdlmonem

Start: 1.10.2022
End: 15.6.2023

Abstract

Advanced driver assistance systems (ADAS) are technologies that enhance the safety and performance of vehicles by providing drivers with information and assistance in various driving scenarios. ADAS can reduce the risk of human error, improve traffic flow, and lower fuel consumption and emissions. In this graduation project book, we present the design and implementation of four ADAS on a smart car: Adaptive Light Control (ALC), Adaptive Cruise Control (ACC), Forward Collision Warning (FCW), and Drowsiness Detection (DD).

ALC is a system that automatically adjusts the headlight beam pattern according to the road conditions and the presence of other vehicles. ALC can improve the visibility and comfort of drivers at night and prevent glaring at oncoming traffic. ACC is a system that automatically maintains a safe distance from the vehicle ahead by adjusting the speed and braking. ACC can reduce driver fatigue and stress, as well as prevent rear-end collisions. FCW is a system that warns the driver of an imminent collision with a vehicle or an obstacle in front of the car. FCW can alert the driver to take evasive action or apply the brakes in time to avoid or mitigate the impact. DD is a system that monitors the driver's eye movements and facial expressions to detect signs of drowsiness or distraction. DD can warn the driver to take a break or pay attention to the road, and prevent accidents caused by impaired driving.

We implemented these four ADAS on a smart car using various sensors, cameras, microcontrollers, and software tools. We tested and evaluated the performance and accuracy of each system in different scenarios and environments. We also discussed the challenges and limitations of our implementation, as well as possible future improvements and extensions. The main contribution of this book is to demonstrate the feasibility and benefits of integrating ADAS on a smart car, and to provide a practical guide for students and researchers who are interested in this field.

Acknowledgements

It has been a great opportunity to gain lots of experience in This project, 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 our supervisors Dr. Ahmed Reda Abdlmonem and Dr. Mohamed Atia Abdelatief who made this work possible. there guidance and advices carried us through all the stages of writing project. I would also like to give special thanks to our families as a whole for their continuous support and understanding. there prayer for us was what sustained us this far.

Table of Contents

List of Figure

Ch1 Introduction

1.1 Definition

ADAS (Advanced Driver Assistance Systems) are passive and active safety systems designed to remove human error components when operating vehicles of many types. ADAS systems use advanced technologies to assist the driver during driving as shown in figure 1, and thereby improve drivers' performance. ADAS uses a combination of sensor technologies to perceive the world around the vehicle, and then either provide information to the driver or acts when necessary.

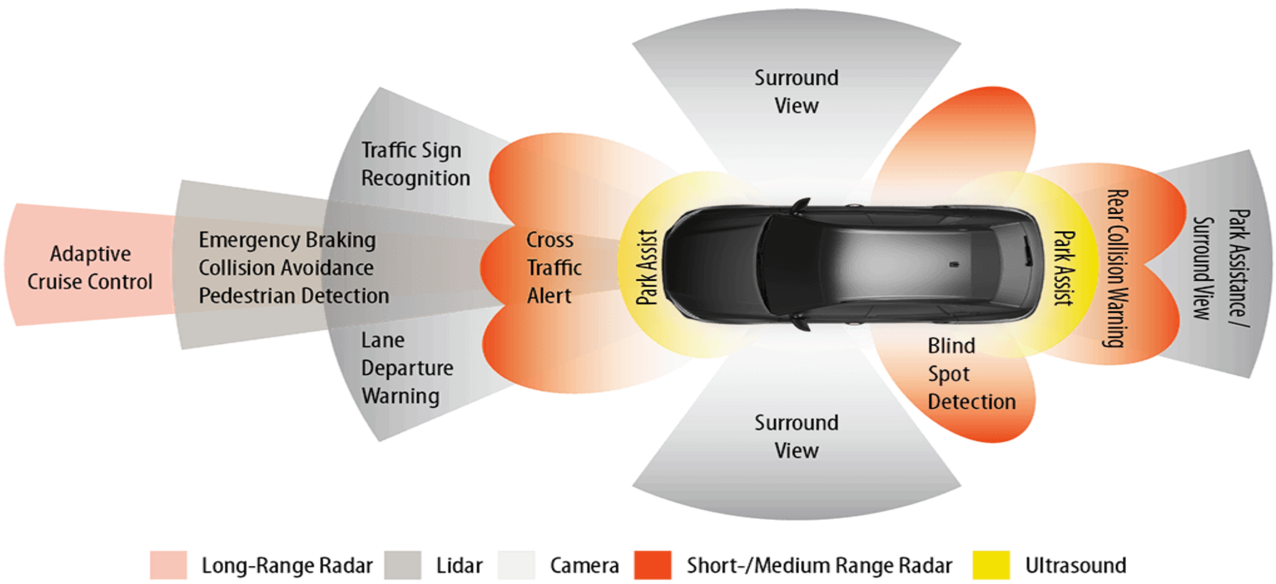


Figure 1: Different types of ADAS sensors used in today's autonomous vehicle

1.2 History

ADAS were first being used in the 1950s with the adoption of the anti-lock braking system. Early ADAS include blind spot information systems, lane departure warning, adaptive cruise control, and adaptive light control, As shown in figure 2 the History of evolved for 20 year.

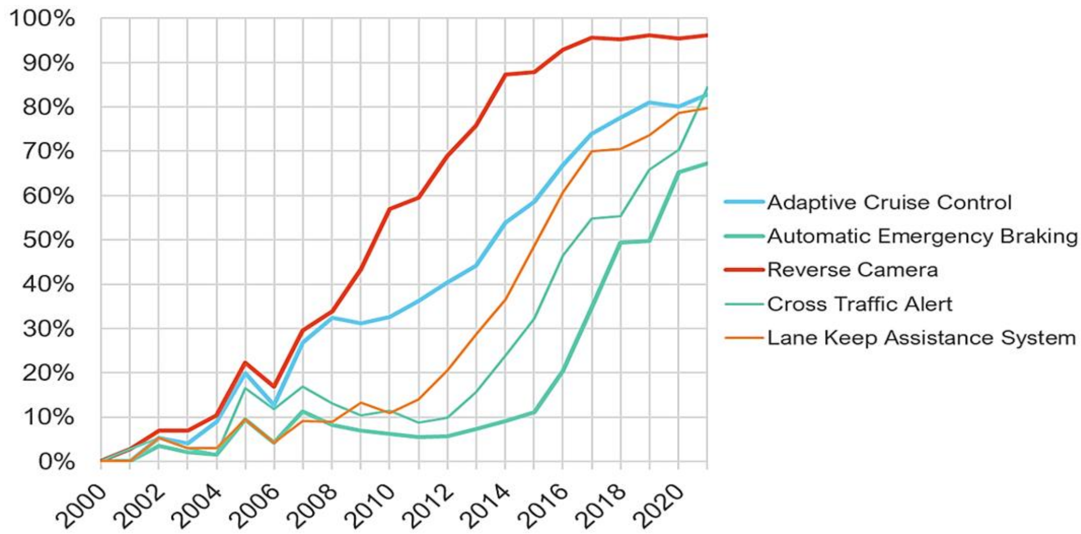


Figure 2: The progression of ADAS feature option

From this analysis it's that current ADAS technologies for level 2 autonomy, adaptive cruise control is the widest adopted, with over 60% of vehicles sold with it as standard. Meanwhile, lane keep assistance systems (LKAS) are the least adopted with nearly one-quarter of vehicles not even having it as an option.

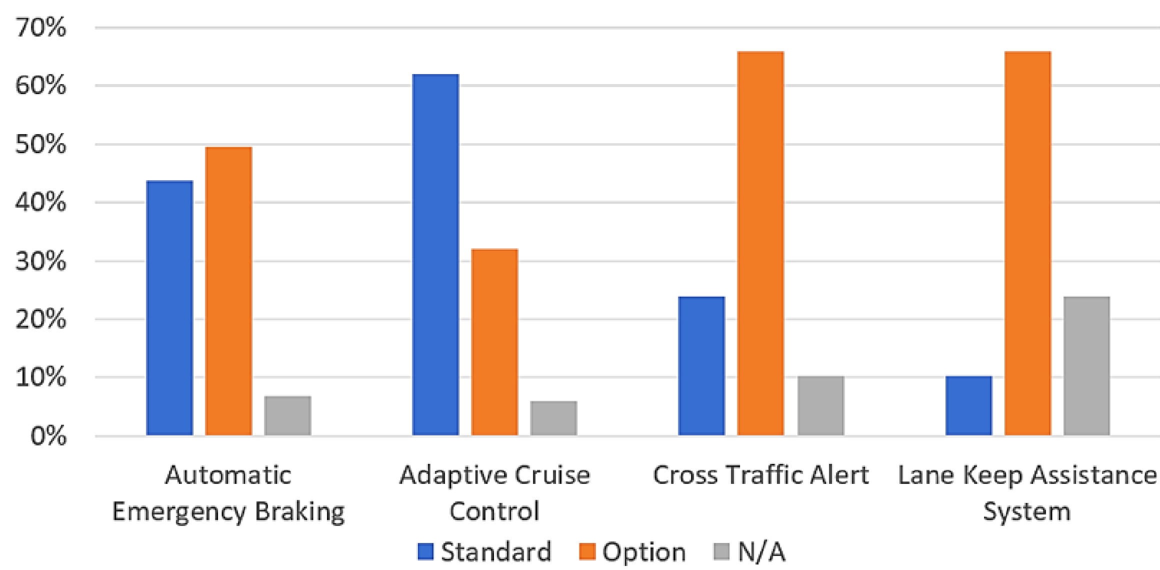


Figure 3: ADAS Feature Adoption

1.3 Safety effects known

The evaluation of ADAS is a young science and their road safety performance is of principal concern to road safety managers. Outcomes can be evaluated in terms of deaths and serious injuries (final outcomes) or any activity which is causally linked to these e.g., the level of seat belt use (intermediate outcomes). In this web text an intervention is deemed to have a 'known positive safety effect' if there are results from more than one study done in a similar road safety context and, where the results are statistically significant and indicate a useful level of effectiveness. Research in the EU and elsewhere has confirmed that the following interventions are likely to make a large contribution towards meeting ambitious safety targets and goals (ETSC 2006 eSafety): Intelligent Speed Adaptation (advisory ISA, Speed Alert); seat belt reminders in all seating positions in new cars, electronic stability control, alcohol interlocks for repeat offenders and fleet drivers, anti-lock braking for motorcycles and event and journey data.

1.4 Safety effects unknown

Systems such as smart keys for young drivers and eCall, that are starting to come on to the market, hold future promise. In general, most of the devices for improvement of braking and handling affect driver behavior, and the questions of driver acceptance, risk compensation and driver reaction, when the system is activated, are important. For example, and not to be confused with Autonomous Emergency Braking Systems, Emergency Brake Assist is often cited as a safety related ADAS. Prospective studies have indicated some benefits, while a study of real accidents has indicated some benefits, though not statistically significant, when Emergency Brake Assist is combined with other measures. However, its contribution to road safety is, as yet not demonstrated. Collision Avoidance systems offer future promise and are receiving much attention though, again, the safety effects are yet unknown.

1.5 Examples of ADAS

1.5.1 ABS

ABS or Anti-Lock Braking System is an important vehicle safety feature that prevents the vehicle from spinning out of control in the event of an emergency, panic, or harsh braking. Sudden braking can cause a wheel to lock up (not spin) and this can result in an immediate loss of traction, particularly over wet or slippery surfaces. This can cause vehicles to skid and meet with an accident. But ABS prevent the wheels from locking up on sudden braking and lets the tyres be controlled by steering input, to safely manoeuvre the vehicle and avoid crashes.

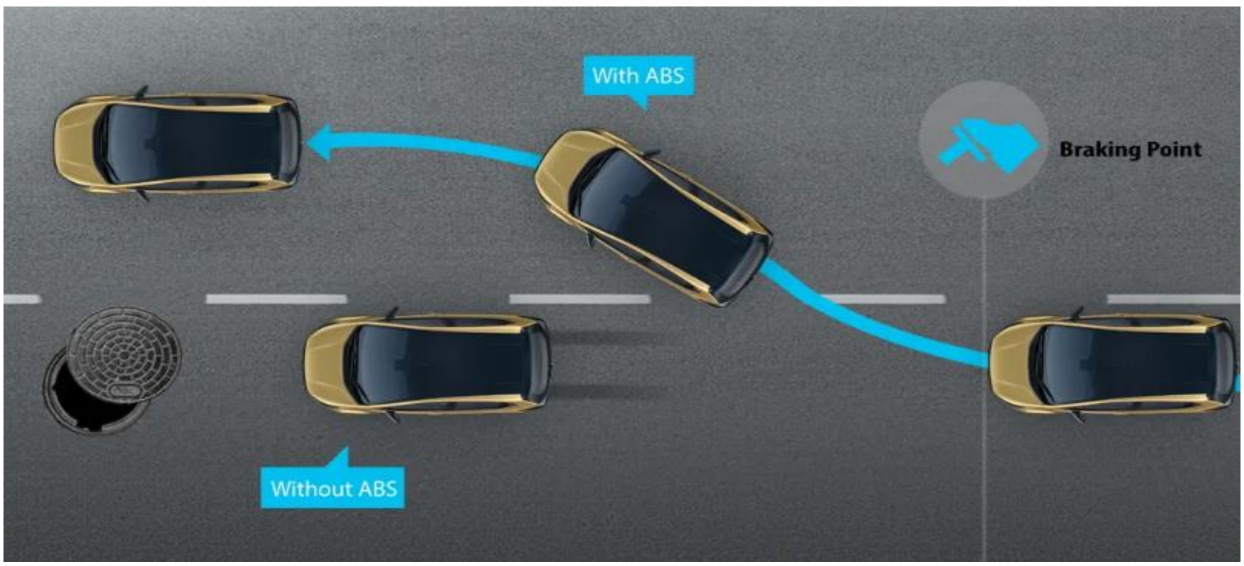


Figure 4: Anti-lock braking system

1.5.2 Adaptive Cruise Control

Adaptive cruise control (ACC) is an active safety system that automatically controls the acceleration and braking of a vehicle. It is activated through a button on the steering wheel and cancelled by driver’s braking and/or another button.

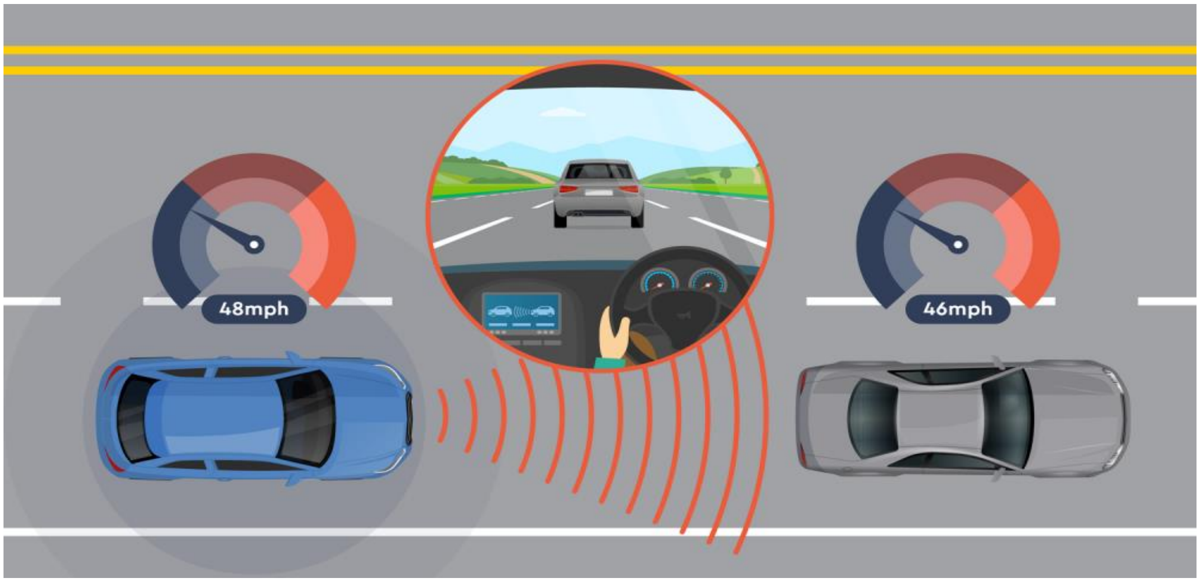


Figure 5: Adaptive Cruise Control System

1.5.3 Adaptive Light Control

Adaptive headlights better illuminate the driving environment compared with traditional headlights, allowing you to see more at night or in low-light conditions.

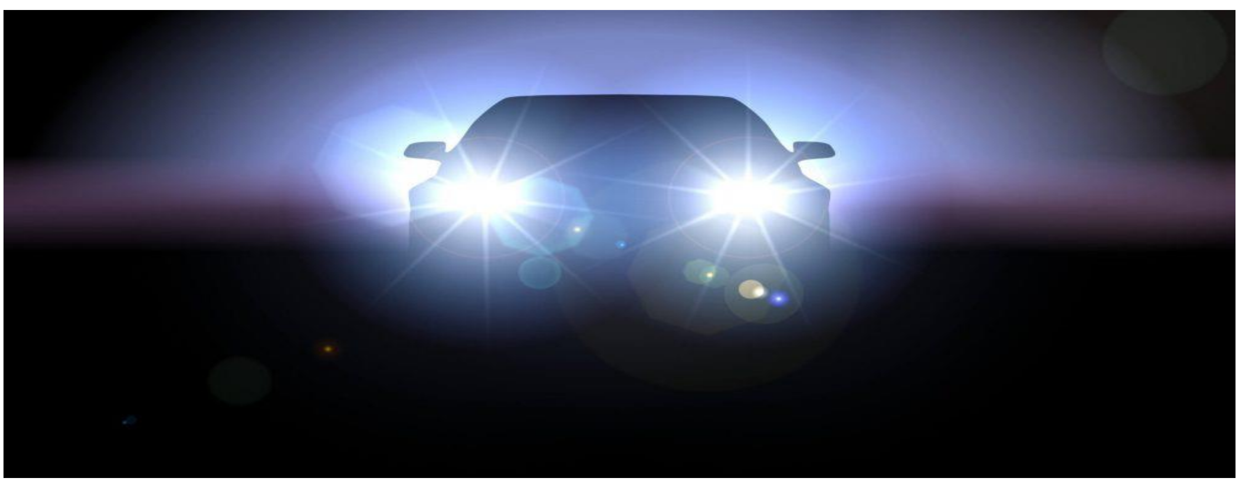


Figure 6: Adaptive Light Control System

1.5.4 Forward collision warning

Forward Collision Warning (FCW) is a driver assistance system that uses sensors to detect when a vehicle is approaching another vehicle or object too quickly. If the system detects an imminent collision, it will alert the driver with an audible warning and/or visual warning on the dashboard.

Figure 7: Forward Collision Warning

1.5.5 Lane Departure System

Lane departure: Lane departure is a safety feature that helps drivers stay in their lane while driving. It uses sensors to detect when a vehicle is drifting out of its lane and then sends an alert to the driver. The alert can be visual, audible, or tactile (vibration). A picture of lane departure is shown below.

Figure 8: Lane Departure System

1.5.6 Drowsiness Detection System

Drowsiness alerts are designed to warn you that you have become drowsy after you have already begun driving; you shouldn't get behind the wheel in the first place if you know you're drowsy.

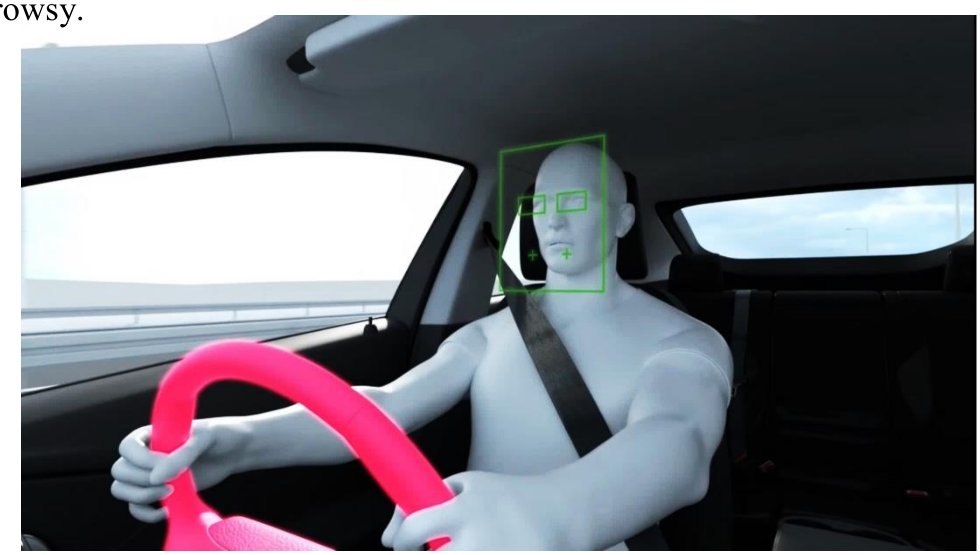


Figure 9: Drowsiness Detection System

1.6 Design

Project design is an early phase of the project lifecycle where ideas, processes, resources, and deliverables are planned out. A project design comes before a project plan as it's a broad overview whereas a project plan includes more detailed information, at early phase of a project where the project's key features, structure, criteria for success, and major deliverables are planned out.

The aim is to develop one or more designs that can be used to achieve the desired project goals. The project design steps might generate various outputs, such as sketches, prototypes, photo impressions, and more.

models were created using Solidworks to design

Figure 10: Solidworks logo

1.6.1 component

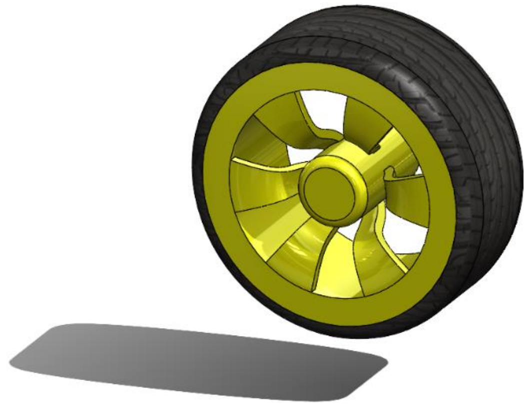


Figure 11: Wheel

vehicle wheel

wheel is simple just tire and rim is the main components.

DC Motor

DC 3V-6V Geared Motor.

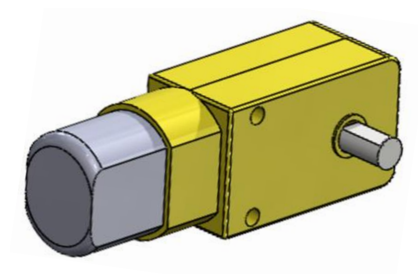


Figure 12: DC Motor

Motor Drive Controller

L298N Board Module Dual H Bridge.

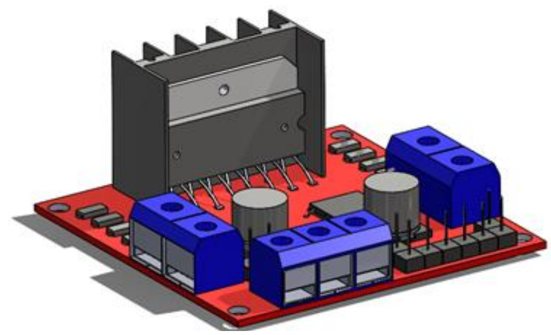


Figure 13: Motor Drive L298N

Led Strip

12v Led Strip.

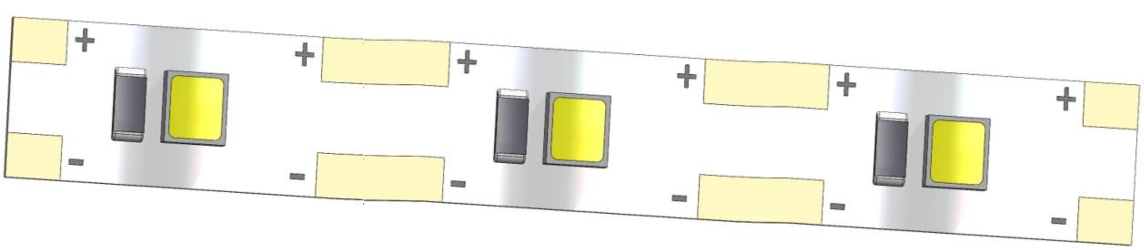


Figure 14: Led Strip

Microcontroller

Blackpill stm32f401.

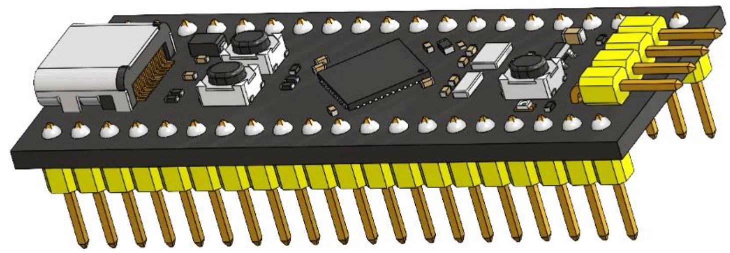


Figure 15: Blackpill stm32f401

Bluetooth Module

HC-05.

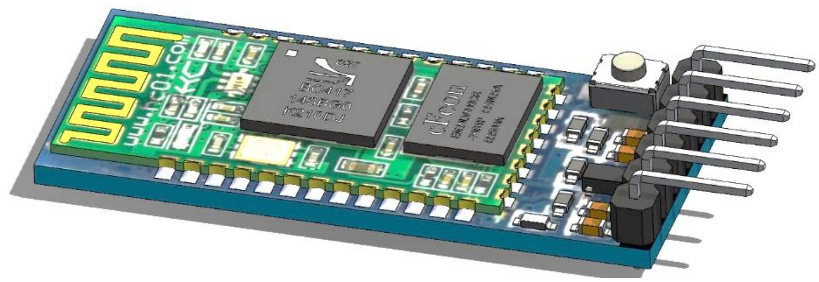


Figure 16: Bluetooth Module

Mini Bread Board

170 tie point breadboard – Green.

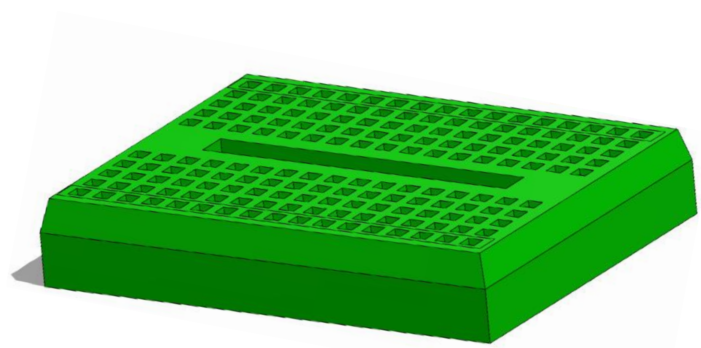


Figure 17: Mini Bread Board

Li-ion Battery and Battery holder

Li-ion Battery 3.7x4v.

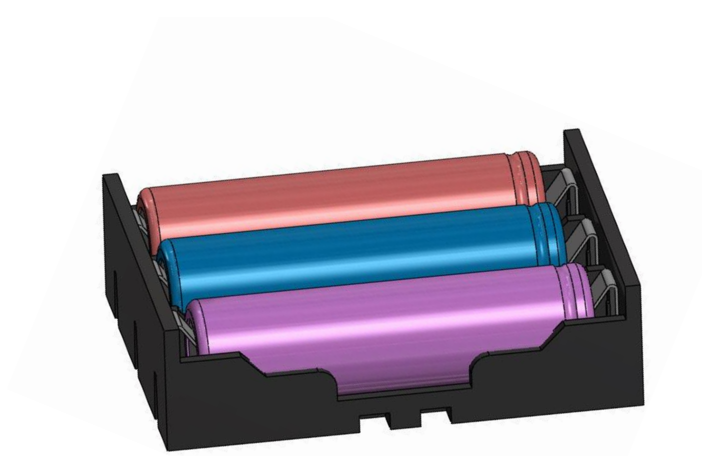


Figure 18: Battery and Battery holder

Breadboard

830 tie point breadboard – white.

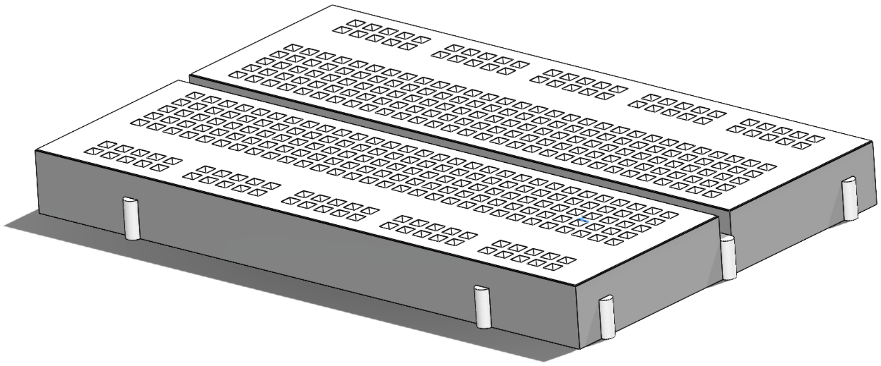


Figure 19: Breadboard

Ultrasonic Sensor Module

Hc-sr04.

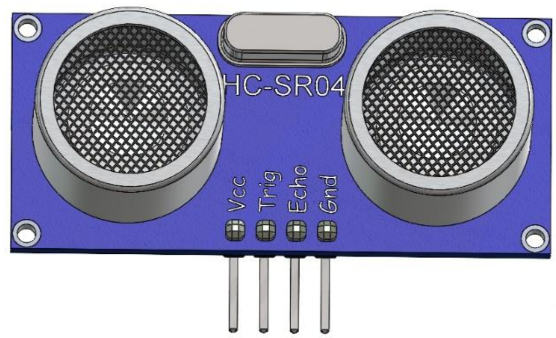


Figure 20: Ultrasonic Sensor

Ultrasonic Sensor Holder

Acrylic –Brown.

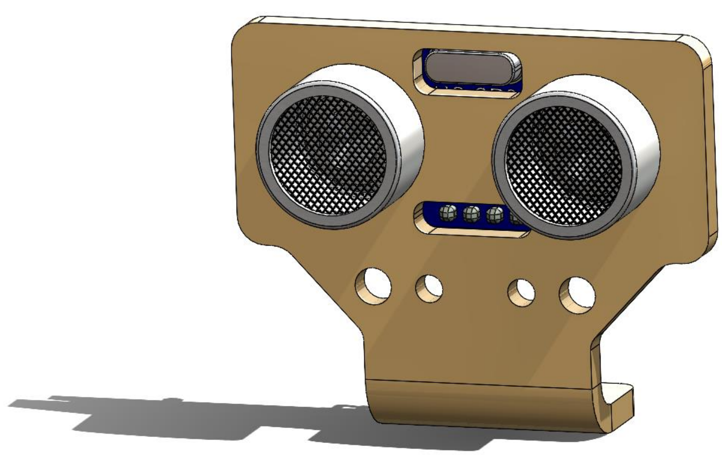


Figure 21: Ultrasonic Sensor Holder

LCD 16X2 Screen

LCD 1602 5V 16×2 Green Screen.

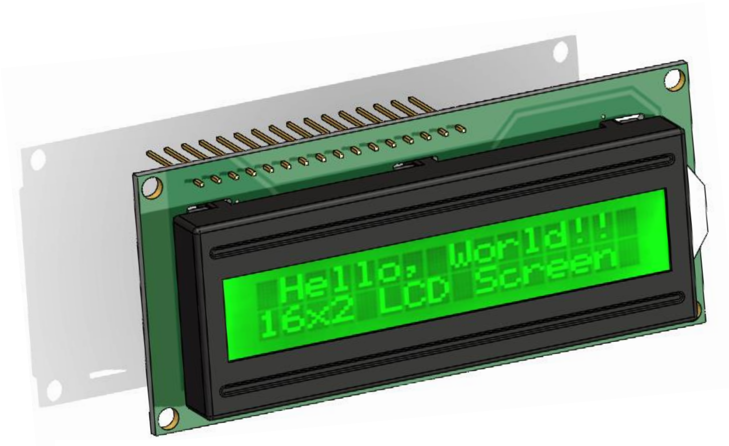


Figure 22: LCD 16X2 Screen

Resistors

10 KΩ Carbon Resistor.

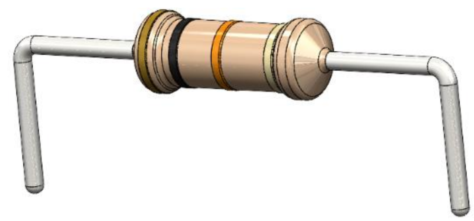


Figure 23: Resistors

Buzzer

5V Buzzer, plastic and metal – black.

Figure 24: Buzzer

LDR

12mm Light Sensitive Photoresistor

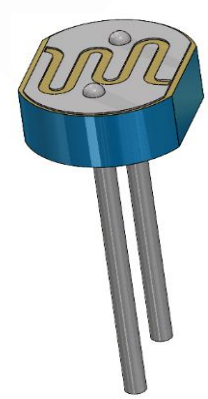


Figure 25: LDR

Camera

Camera Module of raspberry pi

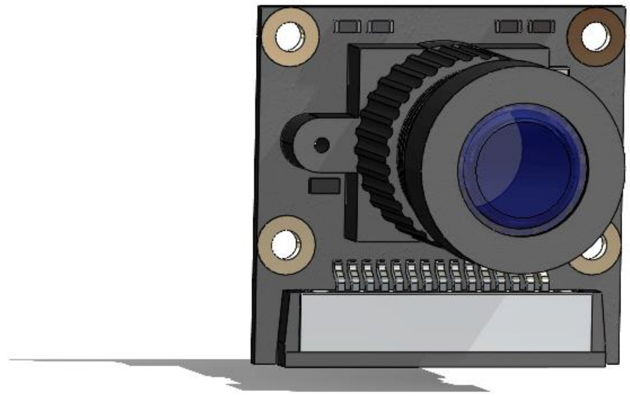


Figure 26: Camera Module

raspberry pi 4

RAM: 4GB; 85x59mm; 9÷12VDC; LPDDR4

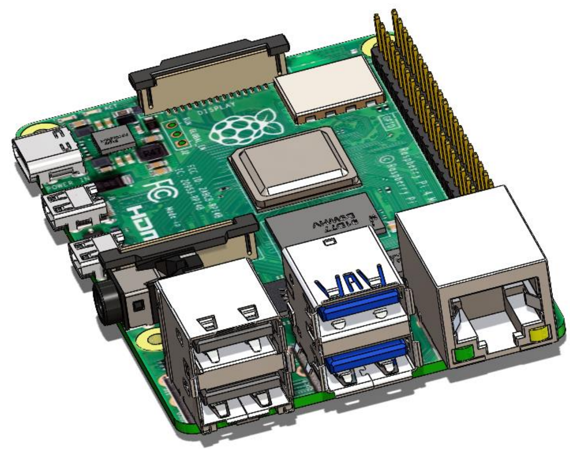


Figure 27: Raspberry pi

internal structure

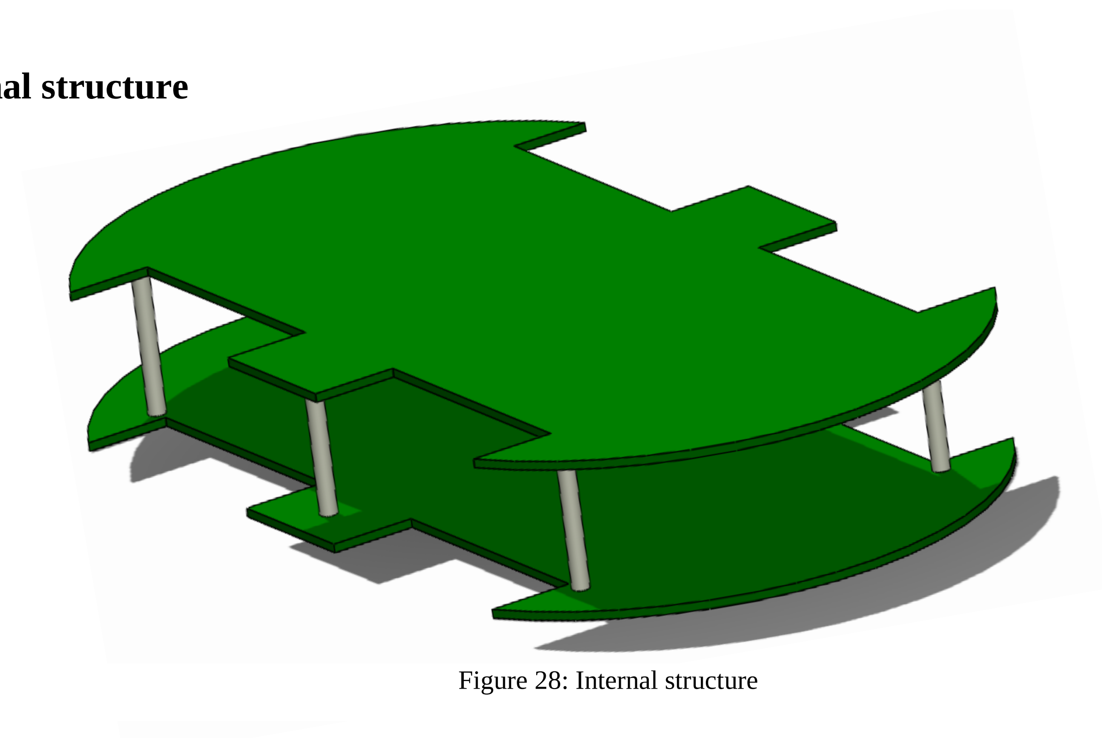


Figure 28: Internal structure

1.6.2 Vehicle full model

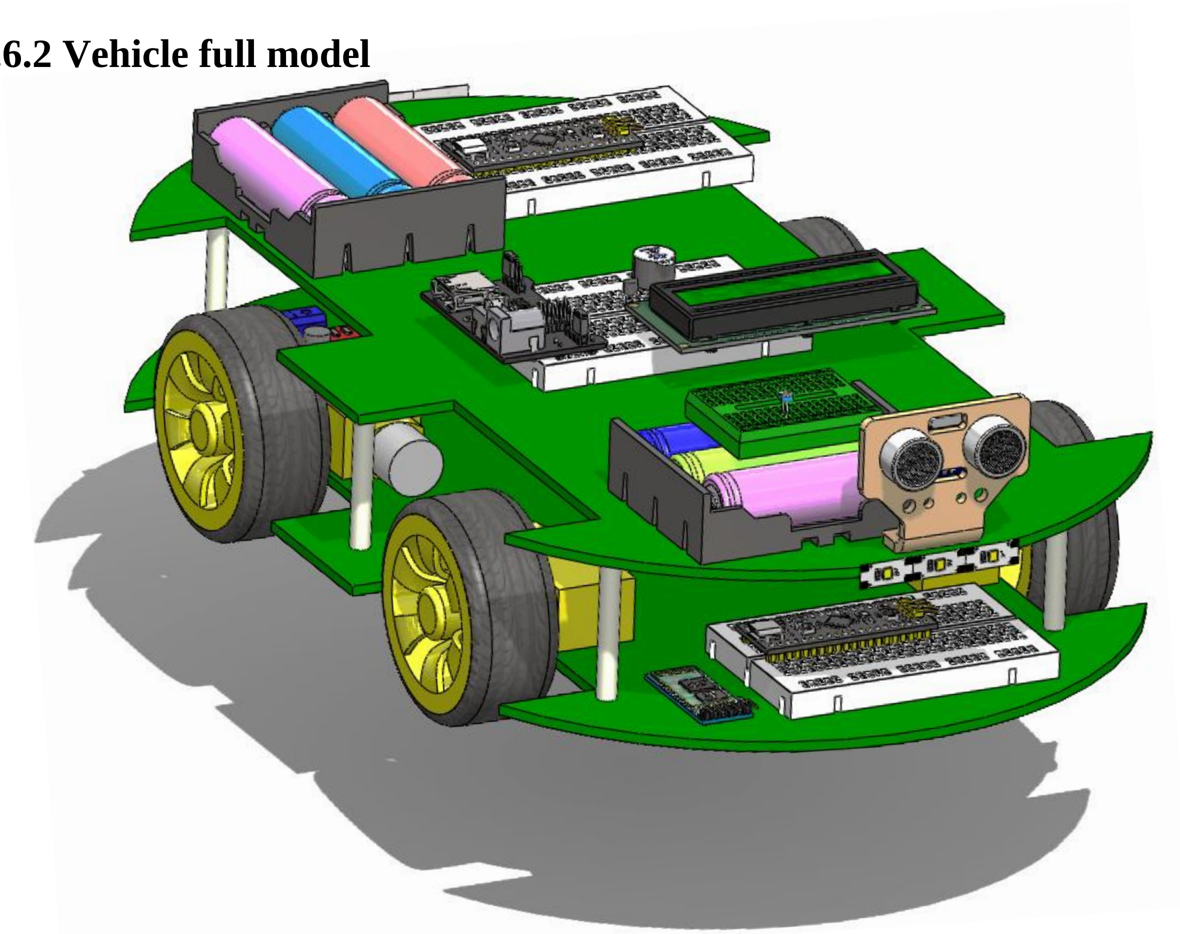


Figure 29 :Vehicle Assembly

1.6.3 Frame

Figure 30: Frame

CH2 Adaptive light control system

2.1 Brief about system

2.1.1 Introduction

The adaptive lighting control system is a safety system designed to help you see more clearly at night without affecting other drivers. The system may also help the driver to see more of the corner when turning. This results in the driver being able to see pedestrians more easily, animals or stopped vehicles along their path of travel. There is a type of ALC that automatically turns on low beams according to vehicle speed and steer to provide better visibility.

2.1.2 The safety benefits of the system

Though driver visibility is improved by the latest headlights still, the majority of the road accidents happen mostly at night due to the road curves and incoming vehicles' head-lights glare. Only 25%-night driving causes 55% fatalities and vehicle crashes occur at night even with less traffic on the road.

2.1.3 How dose System Work

The set of sensors observe/measure the physical variables (changing lighting conditions) that are available in front of the automobile. These variables are converted into an electrical signal. These signals are used as inputs to the processing unit (microcontroller). The microcontroller analyzes the signal from the sensors, adjust the headlights, and illuminate the road concerning the sensed obstacles and glare.

2.2 ALC Components

The ALC consists of 4 modules. These modules are

- sensor module
- Automotive microcontroller unit
- Headlights
- Peripheral circuits

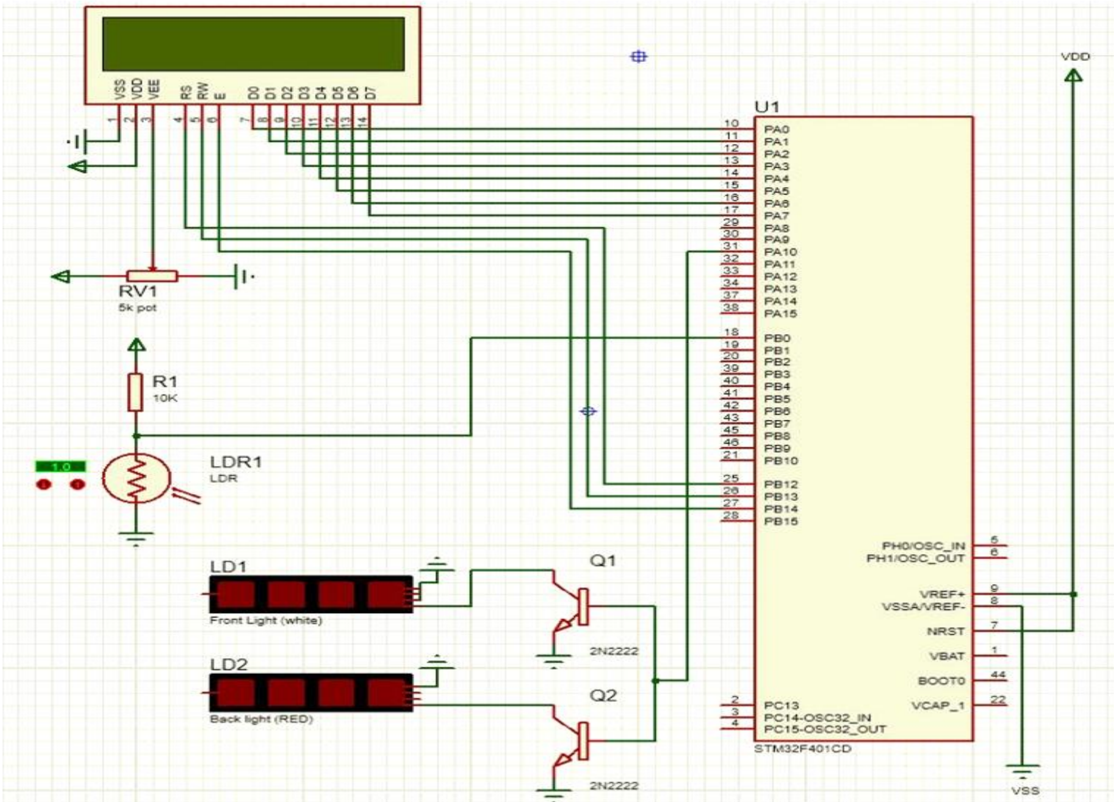


Figure 31: ALC system connection diagram

2.2.1 Sensor Module

The sensor module is a combination of multiple sensors. The notable sensors are steering wheel angle sensor, speed sensor, LDR sensor, and thermal infrared sensors. These sensors measure the changing driving conditions at night and convert into the signal. These signals are used as input data to the microcontroller for adjusting the left and right headlights. The speed sensor monitors the automobile speed. Incoming lights are sensed by the LDR sensor. While, obstacles and objects are determined by the thermal infrared sensor

2.2.2 Automotive microcontroller unit

The automotive microcontroller unit (MCU) is the heart of the ALC. MCU is a STM32F401xD/E advanced Arm®-based 32-bit. The microcontroller contains a program memory (flash), data memory (RAM), EPROM function, watchdog (WDT), analog-to-digital (ADC) converter, and also pulse-width modulation (PWM). While the microcontroller receives the sensor signals as input.

2.2.3 Headlights

Light-Emitting Diodes (LEDs) are PN junction semiconductors that emit monochromatic light with very low power consumption. Semiconductor chips are located in a center of LEDs, these chips consist of two regions that are separated with junctions and these junctions work like a barrier between p and n regions and control the flow of electrons. In our project we will use a front light and a back light.

2.3 Peripheral circuits

2.3.1 LDR Sensor

The Light Dependent Resistor (LDR) is made from a Cadmium-Sulfide compound (CdS).

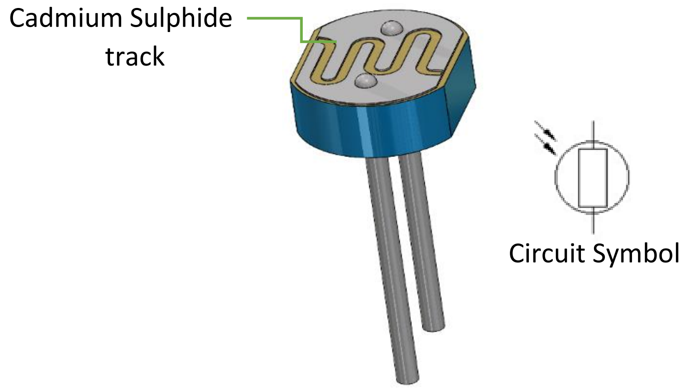


Figure 32: LDR sensor

CdS is a semiconductor material. The resistance of the LDR changes as the light level changes, then More light produces more electron – hole pairs in the material, making the material a better conductor in light and Resistance decreases (conductivity increases) as light levels increase because there are more charge carriers in the material. Note that R = 1/G (G is conductance).

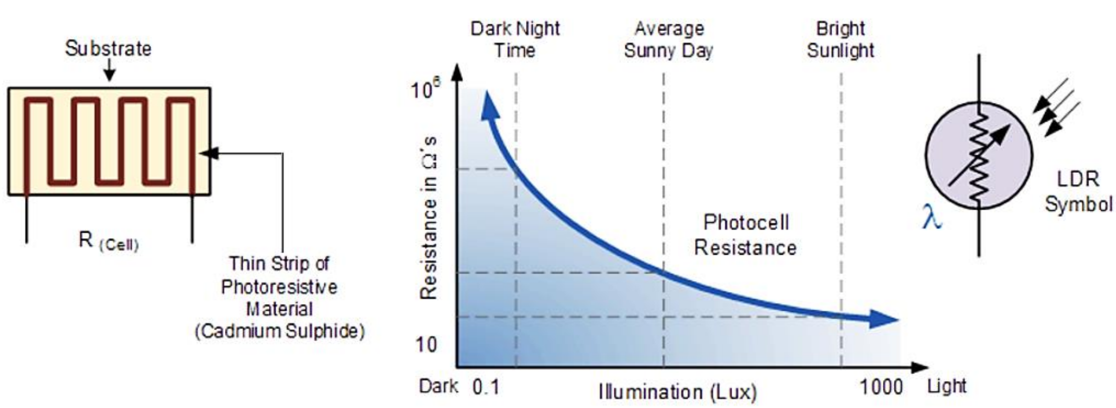


Figure 33: LDR Interfacing

The light dependent resistor uses high resistance semiconductor material. When light falls on such a semiconductor the bound electrons [ie., Valence electrons] get the light energy from the incident photos. We can make a voltage divider circuit using the LDR and a fixed resistor and use the microcontroller's ADC to measure the voltage which indicates the resistance value or the light intensity level. An LDR can have a resistance of 5kΩ in daylight, 8kΩ in room light, and up to 2MΩ in darkness. A pull-up resistor between the output of the LDR sensor and the supply voltage is used to establish an additional loop over the critical components while making sure that the voltage is well-defined. Output voltage equation of the LDR circuit is

$$V_{out} = V_{in} * \frac{R_{LDR}}{R_{LDR} + R_2}$$

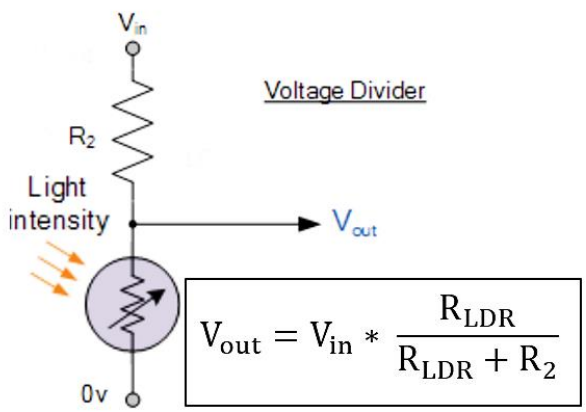


Figure 35: LDR Voltage divider circuit

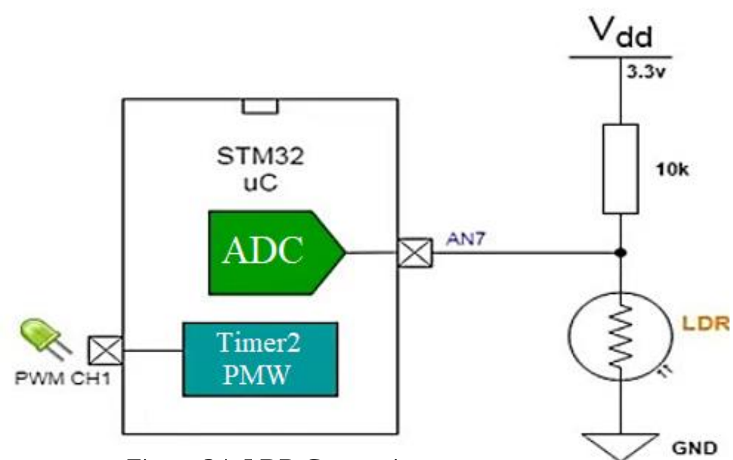


Figure 34: LDR Connection

2.3.2 LCD 2*16 Display

An electronic device that is used to display data and the message is known as LCD 16×2. As the name suggests, it includes 16 Columns & 2 Rows so it can display 32 characters (16×2=32) in total & every character will be made with 5×8 (40) Pixel Dots. So, the total pixels within this LCD can be calculated as 32 x 40 otherwise 1280 pixel.

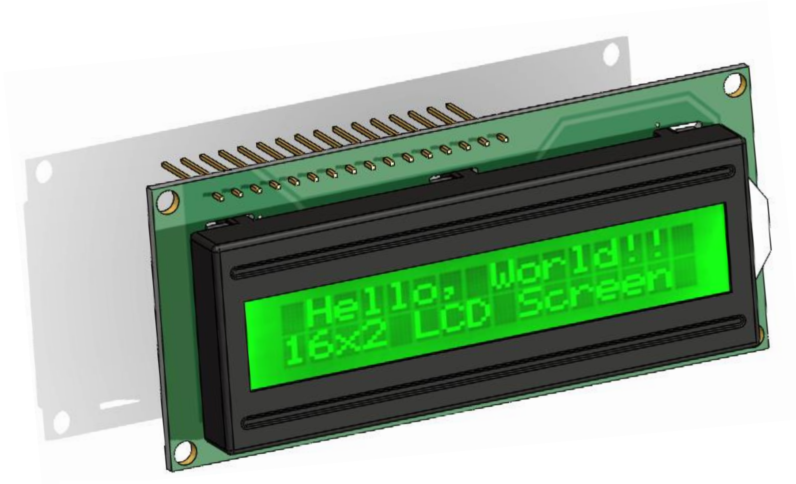


Figure 36: LCD 2*16 Display

Specifications

- The operating voltage of this display ranges from 4.7V to 5.3V
- HD47780 controller
- Number of columns – 16
- Number of rows – 2
- Number of LCD pins – 16
- Characters – 32
- The display bezel is 72 x 25mm
- The operating current is 1mA without a backlight
- PCB size of the module is 80L x 36W x 10H mm
- LED color for backlight is green or blue
- Font size of character is 0.125Width x 0.200height
- It works in 4-bit and 8-bit modes
- Pixel box of each character is 5×8 pixel

Pin Configuration

- Pin1 (Ground/Source Pin): This is a GND pin of display, used to connect the GND terminal of the microcontroller unit or power source.
- Pin2 (VCC/Source Pin): This is the voltage supply pin of the display, used to connect the supply pin of the power source.
- Pin3 (V0/VEE/Control Pin): This pin regulates the difference of the display, used to connect a changeable POT that can supply 0 to 5V.
- Pin4 (Register Select/Control Pin): This pin toggles among command or data register, used to connect a microcontroller unit pin and obtains either 0 or 1(0 = data mode, and 1 = command mode).
- Pin5 (Read/Write/Control Pin): This pin toggles the display among the read or writes operation, and it is connected to a microcontroller unit pin to get either 0 or 1 (0 = Write Operation, and 1 = Read Operation).

- Pin 6 (Enable/Control Pin): This pin should be held high to execute Read/Write process, and it is connected to the microcontroller unit & constantly held high.
- Pins 7-14 (Data Pins): These pins are used to send data to the display. These pins are connected in two-wire modes like 4-wire mode and 8-wire mode. In 4-wire mode, only four pins are connected to the microcontroller unit, whereas in 8-wire mode, 8-pins are connected to microcontroller unit like 0 to 7.
- Pin15 (+ve pin of the LED): This pin is connected to +5V
- Pin 16 (-ve pin of the LED): This pin is connected to GND.

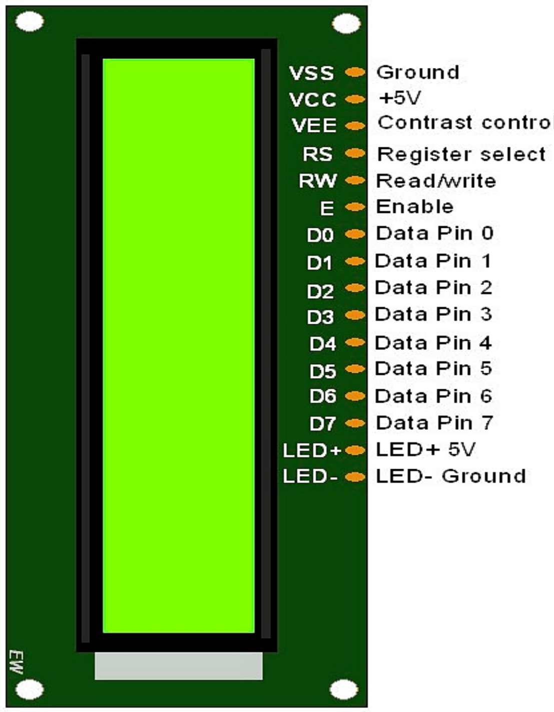


Figure 37: LCD 16*2 Pin Configuration

2.4 Pule Width Modulation (PWM)

Pulse Width Modulation mode allows generating a signal with a frequency determined by the value of the TIMx_ARR register and a duty cycle determined by the value of the TIMx_CCRx register. The PWM mode can be selected independently on each channel (one PWM per OCx output) by writing ‘110’ (PWM mode 1) or ‘111’ (PWM mode 2) in the OCxM bits in the TIMx_CCMRx register. The corresponding preload register must be enabled by setting the OCxPE bit in the TIMx_CCMRx register, and eventually the auto-reload preload register (in upcounting or center-aligned modes) by setting the ARPE bit in the TIMx_CR1 register.

2.4.1 Timer1

The advanced-control timers (TIM1) consist of a 16-bit auto-reload counter driven by a programmable prescaler. It may be used for a variety of purposes, including measuring the pulse lengths of input signals (input capture) or generating output waveforms (output compare, PWM, complementary PWM with dead-time insertion). Pulse lengths and waveform periods can be modulated from a few microseconds to several milliseconds using the timer prescaler and the RCC clock controller prescalers. The advanced-control (TIM1) and general-purpose (TIMx) timers are completely independent, and do not share any resources. They can be synchronized together.

features

- 16-bit up, down, up/down auto-reload counter.
- 16-bit programmable prescaler allowing dividing (also “on the fly”) the counter clock frequency either by any factor between 1 and 65536.
- Up to 4 independent channels for: – Input Capture – Output Compare – PWM generation (Edge and Center-aligned Mode) – One-pulse mode output
- Complementary outputs with programmable dead-time
- Synchronization circuit to control the timer with external signals and to interconnect several timers together.
- Repetition counter to update the timer registers only after a given number of cycles of the counter.
- Break input to put the timer’s output signals in reset state or in a known state.
- Interrupt/DMA generation on the following events: – Update: counter overflow/underflow, counter initialization (by software or internal/external trigger) – Trigger event (counter start, stop, initialization or count by internal/external trigger) – Input capture – Output compare – Break input
- Supports incremental (quadrature) encoder and hall-sensor circuitry for positioning purposes
- Trigger input for external clock or cycle-by-cycle current management

Pulse Width Modulation mode allows generating a signal with a frequency determined by the value of the TIMx_ARR register and a duty cycle determined by the value of the TIMx_CCRx register. The PWM mode can be selected independently on each channel (one PWM per OCx output) by writing ‘110’ (PWM mode 1) or ‘111’ (PWM mode 2) in the OCxM bits in the TIMx_CCMRx register. The corresponding preload register must be enabled by setting the OCxPE bit in the

TIMx_CCMRx register, and eventually the auto-reload preload register (in upcounting or center-aligned modes) by setting the ARPE bit in the TIMx_CR1 register.

2.4.2 PWM selected mode

The selected mode is PWM edge-aligned mode

Upcounting configuration Upcounting is active when the DIR bit in the TIMx_CR1 register is low.

Example

consider PWM mode 1. The reference PWM signal OCxREF is high as long as TIMx_CNT < TIMx_CCRx else it becomes low. If the compare value in TIMx_CCRx is greater than the auto-reload value (in TIMx_ARR) then OCxREF is held at ‘1’. If the compare value is 0 then OCxRef is held at ‘0’. Figure 7 shows some edge-aligned PWM waveforms in an example where TIMx_ARR=8

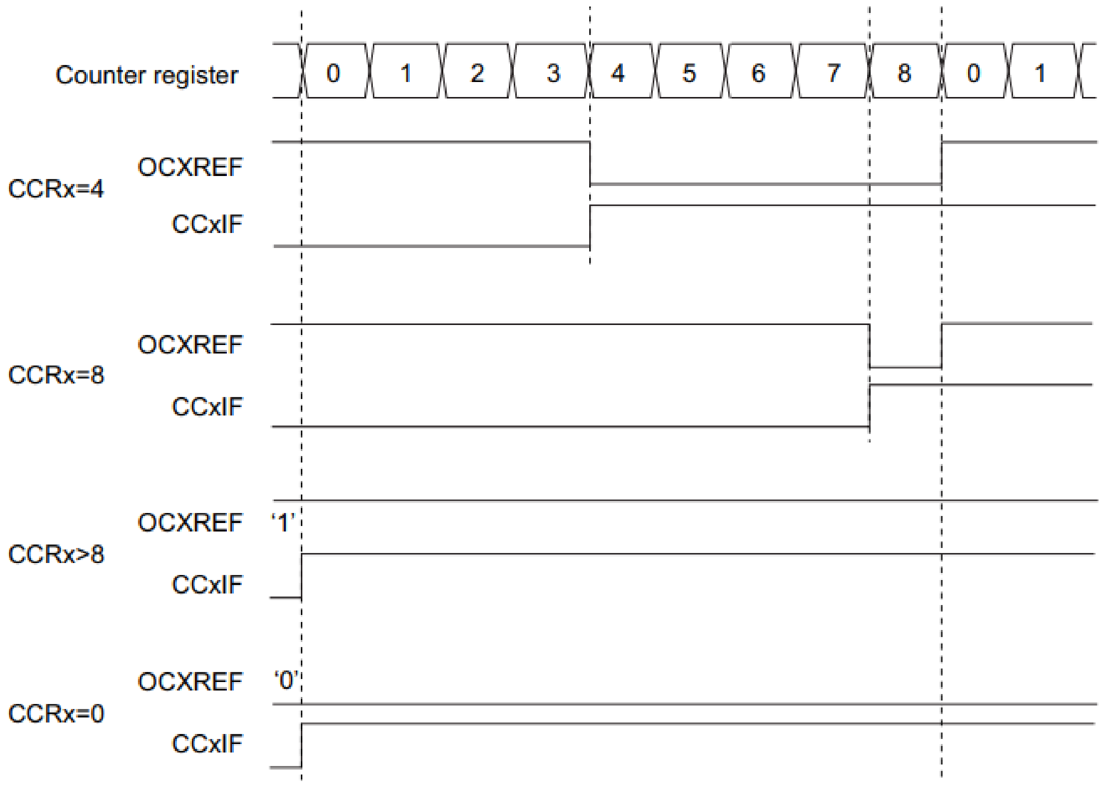


Figure 38: Edge-aligned PWM waveforms (ARR=8)

2.5 Analog digital conversion (ADC)

ADC introduction the 12-bit ADC is a successive approximation analog-to-digital converter. It has up to 19 multiplexed channels allowing it to measure signals from 16 external sources, two internal sources, and the VBAT channel. The A/D conversion of the channels can be performed in single, continuous, scan or discontinuous mode. The result of the ADC is stored into a left or right-aligned 16-bit data register. In our application we used continuous conversion mode, the ADC starts a new conversion as soon as it finishes one. This mode is started with the CONT bit at 1 either by external trigger or by setting the SWSTRT bit in the ADC_CR2 register (for regular channels only). After each conversion: • If a regular group of channels was converted: – The last converted data are stored into the 16-bit ADC_DR register – The EOC (end of conversion) flag is set – An interrupt is generated if the EOCIE bit is set.

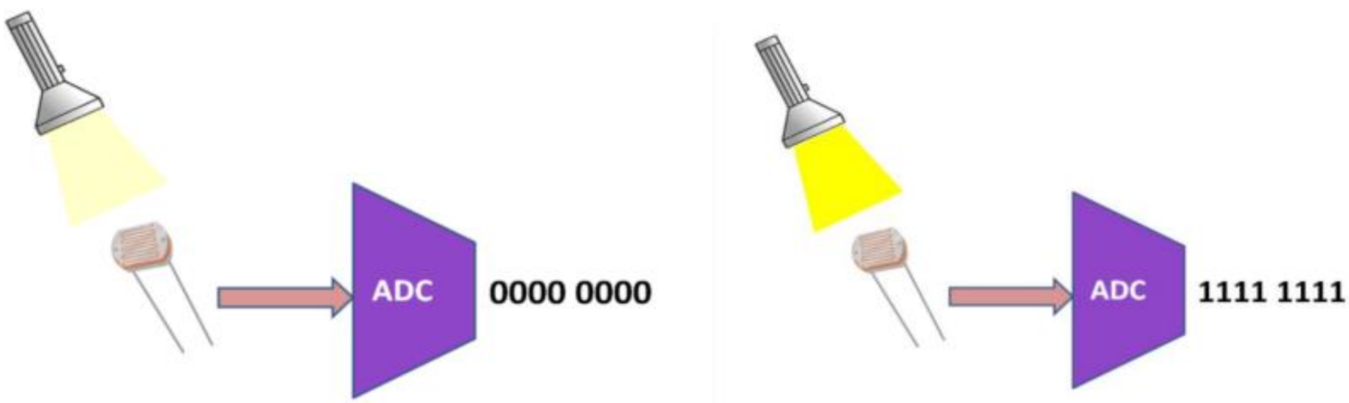


Figure 39: ADC with LDR sensor

2.6 Part of the application code

```
/* Get ADC conversion */
                L_u16BrightnessLevel = HLDR_u16DigitalOutputValue( ) ;


                /* Go to the first line and square4 on the LCD */
                HLCD_vGoTo( HLCD_LINE1, HLCD_Square4 ) ;


                /* Display brightness number */
```

```
HLCD_vDispNumber( L_u16BrightnessLevel ) ;

/* Out the conversion value on the TIM1_CH3 */
MTIM1_vSetCompareReg3Value( L_u16BrightnessLevel ) ;

/* Display brightness bar */
HLCD_vDispBrightnessBar( L_u16BrightnessLevel ) ;

/* Delay 60ms */
MSysTick_vDelayMilliSec( 60 ) ;

/* Clear the old ADC conversion value */
HLCD_vClearChar( HLCD_LINE1, HLCD_Square4 ) ;
HLCD_vClearChar( HLCD_LINE1, HLCD_Square5 ) ;
HLCD_vClearChar( HLCD_LINE1, HLCD_Square6 ) ;
HLCD_vClearChar( HLCD_LINE1, HLCD_Square7 ) ;
HLCD_vClear2ndRow( ) ;

/* Delay 10ms */
MSysTick_vDelayMilliSec( 10 ) ;
```

2.7 Design overview

2.7.1 System Architecture

ALC system is designed using a modular and layered architecture. The system is composed of several main components, including sensors, actuators, processing units, and communication interfaces. These components work together to collect data from the environment, process it to make decisions, and act on those decisions to achieve the desired behavior.

The system is organized into several layers, including a hardware layer, a driver layer, an operating system layer, and an application layer. Each layer is responsible for a specific aspect of the system's operation and provides services to the layers above it. The layered architecture allows us to manage the complexity of the system and to develop and test each layer independently.

In the following sections, we will provide more detailed information about the layered architecture.

2.7.2 Layered Architecture

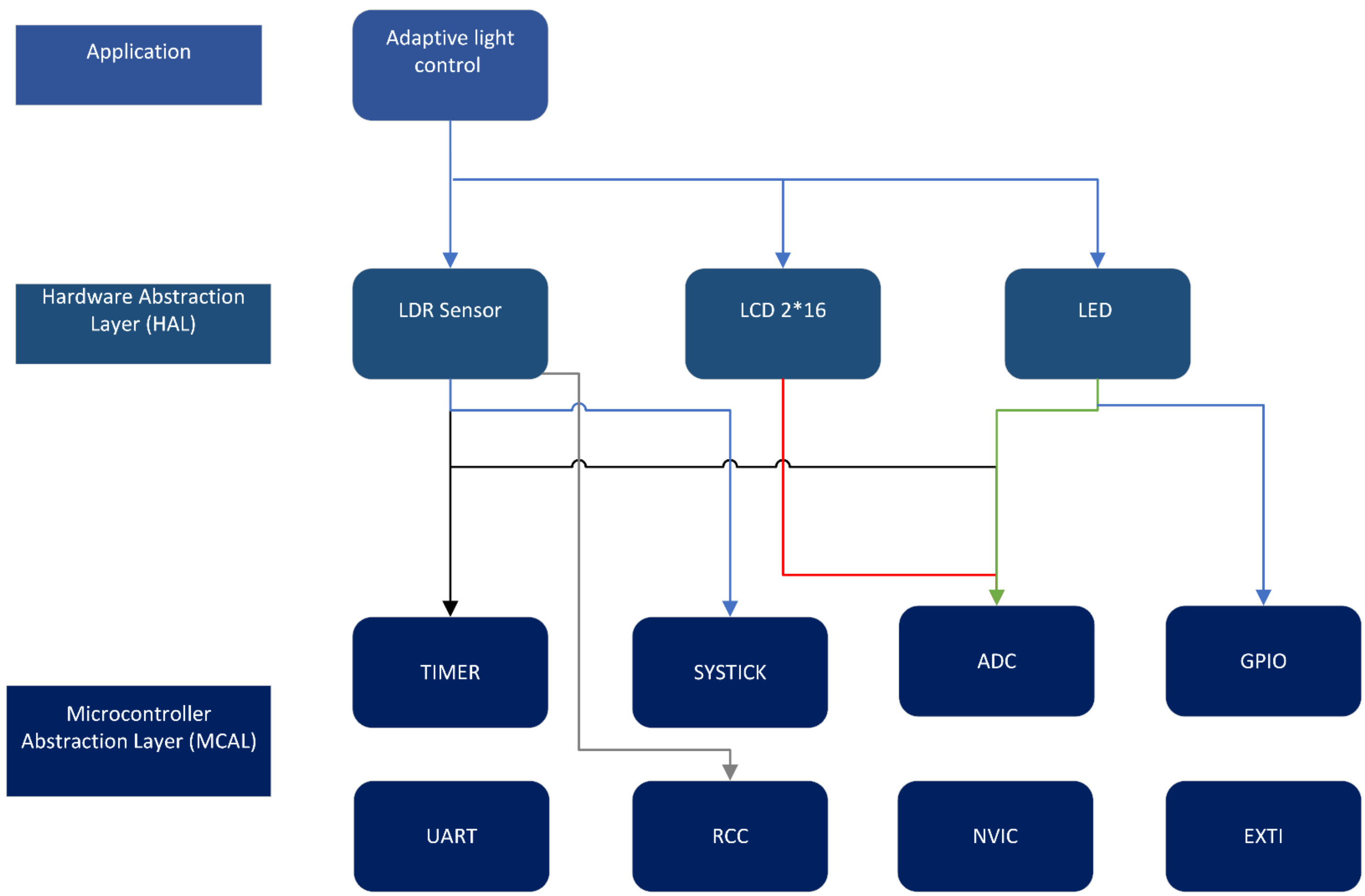
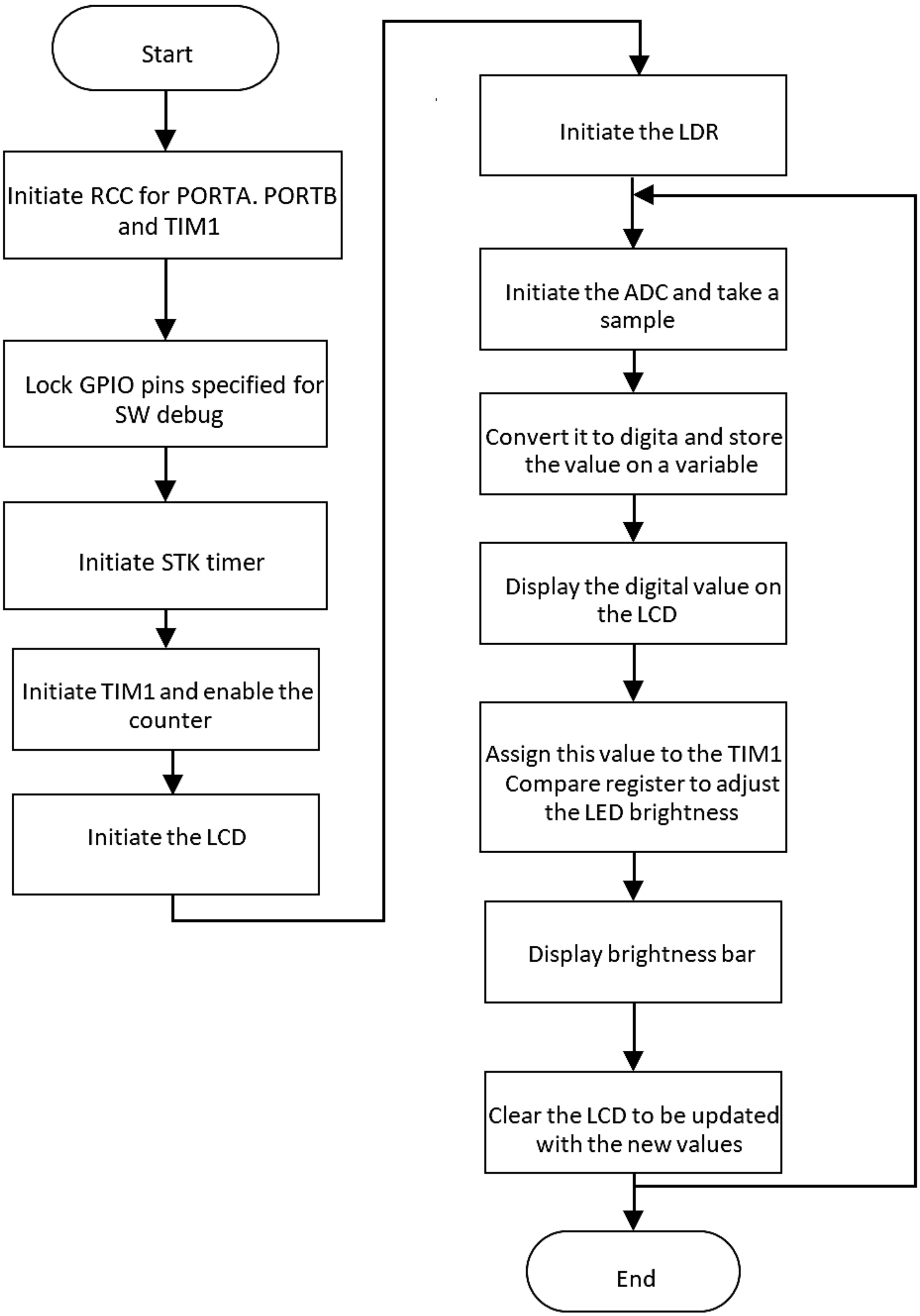


Figure 40: Layered Architecture

2.7.3 System flowchart



CH3 Car Driving Modes

3.1 Introduction

One of the most important aspects of driving a car is the quality of the driving experience that the car offers and the technology it uses to make the driving experience easier and better for the driver. Various systems have been developed to achieve this goal. We are going to talk about Adaptive Cruise Control (ACC) and Forward Collision Warning (FCW).

Adaptive Cruise Control (ACC), which is a very useful driving and driver assistance system that automatically adjusts the speed of the vehicle to keep a safe distance from other vehicles on the road. Forward Collision Control (FCW) is also another helpful system that aims to ease and enhance the driving experience, by alerting the driver with various means if a potential collision is detected ahead. Both systems use sensors such as radar, camera, ultrasonic, or lidar to monitor the surrounding environment and provide feedback to the driver or the vehicle.

ACC and FCW can improve the driving experience by reducing the need for manual braking and accelerating, especially in congested traffic situations. Moreover, it can also help improve fuel efficiency and reduce emissions by minimizing the sudden changes in speed. However, neither of the systems replace the driver's role in monitoring the traffic situation and reacting accordingly. Both systems have some limitations that affect its performance, such as the maximum and minimum speed, the sensor coverage, and the weather visibility. ACC and FCW may also fail to recognize some objects, such as people, bikes, or animals. Therefore, drivers must always be prepared to resume control of the vehicle when using either of the systems.

Some vehicles are equipped with either FCW or ACC or both. The vehicle’s model varies. ACC and FCW are considered to be one of the systems of the transition to more autonomous vehicles, which aim to reduce human error and improve safety and comfort. However, there are still many challenges and ethical issues to be addressed before fully autonomous vehicles can be widely adopted. ACC and FCW are one of the steps towards achieving this goal, but it also requires careful design and evaluation to ensure its effectiveness and reliability.

3.2 Driving Modes

The slow transition from traditional mechanical cars to modern cars with the integration of intelligent digital systems into the car definitely enhanced the driving experience and made it more enjoyable and convenient.

In the following sections, the various car driving modes are going to be explained with stating the control commands for each driving mode**:**

- Traditional Mode.
- Normal Cruise Control Mode.
- Adaptive Cruise Control Mode.
- Forward Collision Warning Mode.

3.2.1 Traditional Mode

Definition

This mode allows the driver to control the car in a simple and direct way. The driver can move the car forward or backward, turn it right or left, speed up or slow down, and stop the car when needed. This mode is suitable for drivers who prefer a conventional driving experience.

Control Commands

Traditional Mode State
[F] = Move forward
[B] = Move backward
[R] = Turn right
[L] = Turn left
[S] = Stop the car
[+] = Speed up the car by 10%
[-] = Slow down the car by 10%
[E] = Turn off the traditional mode

Figure 41: Traditional Mode State, Control Commands

As shown in figure 41, these are the different commands to control the car via Bluetooth. Depending on the data received from the Bluetooth device, the software executes the appropriate action to steer the car.

Advantages

The traditional mode relies solely on the driver's skills and judgment. This means that the driver has full control over the car, but also full responsibility for any mistakes or accidents that may occur. The traditional mode may appeal to some drivers who enjoy the challenge and thrill of driving, but it also poses more risks and requires more attention and effort from the driver.

Disadvantages

One of the drawbacks of the traditional mode of driving is that it depends entirely on the driver's abilities and judgement, which can pose a risk to the driver and the surrounding environment, because any error or mistake could have serious consequences. For example, most car accidents are caused by driver error.

Furthermore, the traditional mode does not help improve fuel efficiency or reduce emissions, especially in congested traffic situations, where the driver has to constantly accelerate and decelerate. This can increase fuel consumption and greenhouse gas emissions, which contribute to climate change and air pollution.

3.2.2 Normal Cruise Control

Definition

In this mode, the car maintains a constant speed while it is operating. The car does not have much flexibility or functionality in this mode. It simply keeps the speed level, without any perception of the environment or impediments in front of it.

Control Commands

Normal Cruise Control Mode State
[+] = Speed up the car by 10%
[-] = Slow down the car by 10%
[E] = Turn off the NCC mode

Figure 42: Normal Cruise Control State, Control Commands

As shown in figure 42, there is not much assistance to the driver over the car in this mode. While it is on, you can speed up or slow down the car, regardless of any obstacles ahead or any blockages in front of it.

Advantages

- It can reduce driver fatigue and stress by allowing the driver to relax their foot and leg muscles.
- It can improve fuel efficiency and reduce emissions by maintaining a steady speed and avoiding unnecessary acceleration and deceleration.
- It can help the driver avoid speeding tickets by keeping the speed within the legal limit.

Disadvantages

- It can reduce driver attention and awareness by making the driving task less engaging and challenging.
- It can increase the risk of collisions by preventing the driver from adjusting the speed according to the traffic flow and road conditions.
- It can malfunction or fail due to technical issues or human errors, causing the driver to lose control of the vehicle.

Therefore, normal cruise control mode can have both positive and negative effects on driving performance and safety. The driver should use this mode with caution and discretion, and always be ready to take over the vehicle if needed.

3.2.3 Adaptive Cruise Control

Definition

Adaptive cruise control (ACC) is a feature that enhances the conventional cruise control system. Unlike the normal mode, ACC adjusts the speed according to the distance from the vehicle ahead. This way, ACC can prevent collisions and ensure safety on the road. ACC can also resume the desired speed after there are not any blockages. ACC is a smart and convenient way to cruise with confidence and comfort.

Control Commands

Adaptive Cruise Control Mode State
[+] = Speed up the car by 10%
[-] = Slow down the car by 10%
[E] = Turn off the NCC mode

Figure 43: Adaptive Cruise Control, Control Commands

As shown in figure 43, the control commands are almost identical to the normal cruise control. However, before any action is taken, the software triggers the sensor (e.g., ultrasonic) to detect any obstacles ahead.

Flowchart Diagram

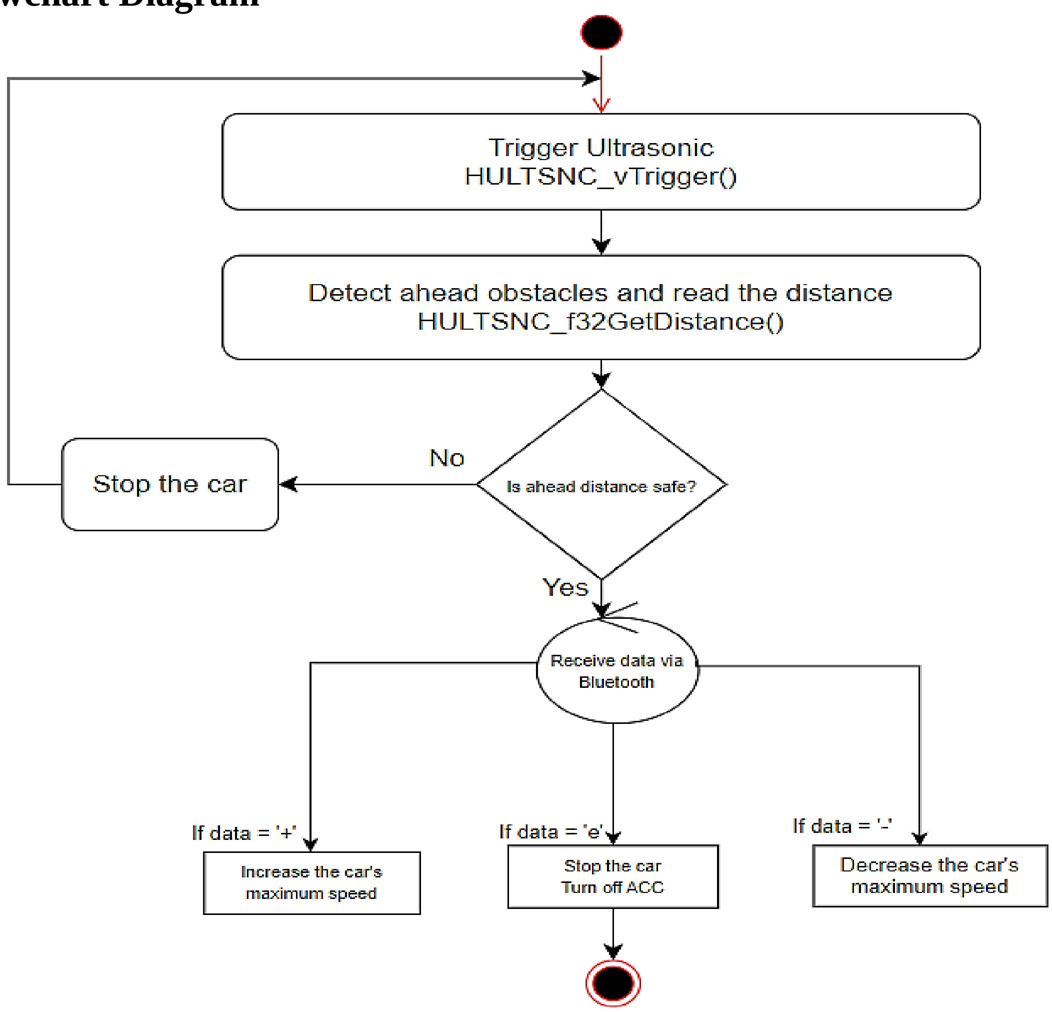


Figure 44: Adaptive Cruise Control, Flowchart Diagram

Advantages

- It can reduce driver fatigue and stress by maintaining a constant speed and following distance without the need for manual intervention.
- It can improve fuel efficiency and reduce emissions by avoiding unnecessary acceleration and braking.
- It can prevent collisions and accidents by reacting faster than human drivers to changes in traffic conditions and applying brakes or accelerating as needed.
- It can enhance comfort and enjoyment by allowing drivers to focus on other aspects of driving, such as navigation, entertainment, or conversation.

Disadvantages

- It can be expensive and complex to install and maintain, requiring sensors.
- It can malfunction or fail due to bad weather, poor visibility, road debris, or system errors, resulting in loss of control or damage to the vehicle or others.
- It can create overconfidence or complacency in drivers, who may rely too much on the system and neglect their own driving skills and awareness.
- If it doesn't match the road rules, traffic patterns, or driving styles, it can confuse or frustrate drivers and other road users.

3.2.4 Forward Collision Warning

Definition

The Forward Collision Warning (FCW) system is a safety feature that alerts drivers of an impending collision with a vehicle or an obstacle in front of them, by uses sensors, such as cameras, ultrasonic, radar, or lidar, to detect the distance and relative speed of the objects ahead. If the system detects a potential collision, it warns the driver by emitting a sound, flashing a light, or vibrating the steering wheel or seat. The driver can then take an action to avoid or mitigate the crash, Also FCW differs from Adaptive Cruise Control (ACC), which also uses sensors to monitor the traffic ahead and adjust the speed accordingly. The main difference is that FCW does not take any action for the driver. It only provides a warning and lets the driver decide how to react. Some FCW systems are also integrated with Automatic Emergency Braking (AEB), which can apply the brakes if the driver fails to respond in time.

Control Commands

Forward Collision Warning State
[F] = Move forward
[B] = Move backward
[R] = Turn right
[L] = Turn left
[S] = Stop the car
[+] = Speed up the car by 10%
[-] = Slow down the car by 10%
[E] = Turn off the traditional mode

Figure 45: Forward Collision Warning, Control Commands

As shown in figure 45, the control commands for the FCW system. Unlike the traditional mode and the NCC, the software activates the ultrasonic sensors before executing any action.

The sensors scan the surroundings for any potential obstacles or vehicles. If an obstacle is detected, the system alerts the driver by different means, such as sound, light, or vibration. The driver reacts to prevent or reduce the impact of a collision.

Flowchart Diagram

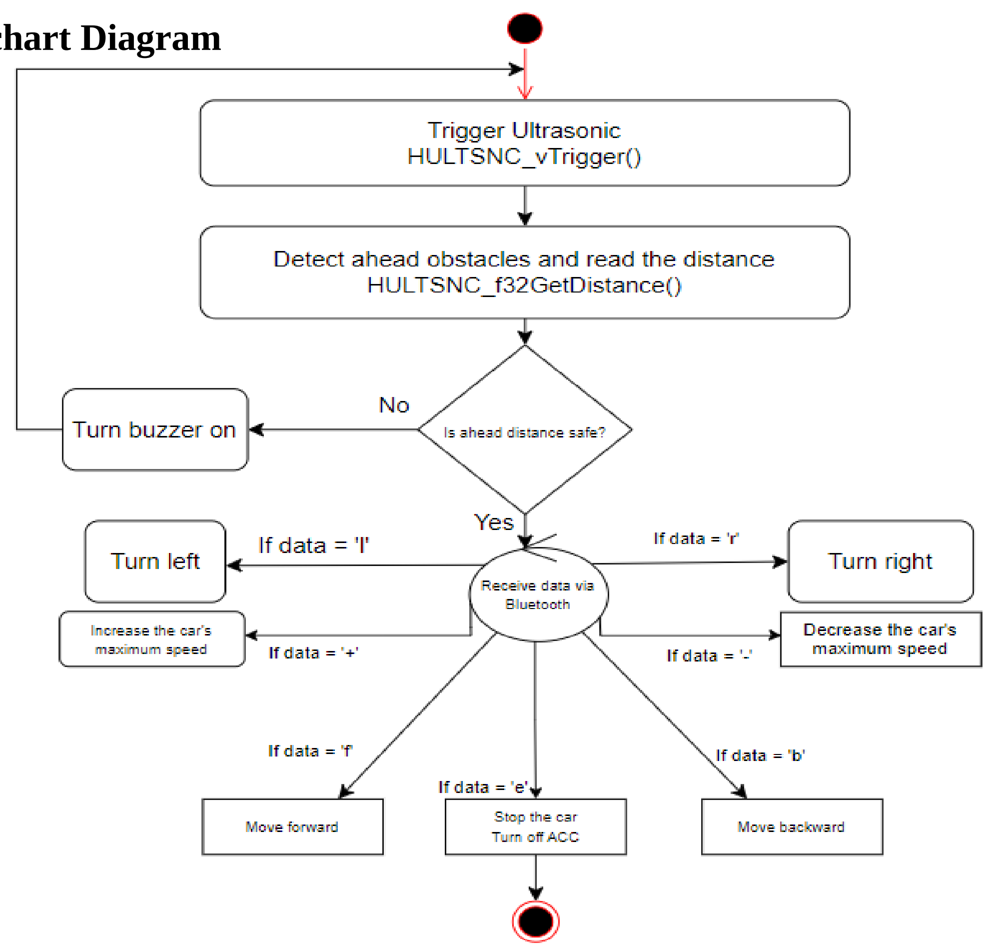


Figure 46: Forward Collision Warning, Flowchart Diagram

Advantages

A forward collision warning system is a safety feature that alerts drivers when they are approaching a vehicle or an obstacle too quickly and may cause a crash. This system can help drivers avoid or mitigate collisions by giving them more time to react and brake.

Disadvantages

- It may not work well in bad weather conditions, such as fog, rain, or snow.
- It may make false alarms or miss some hazards, such as pedestrians or animals.
- It may make drivers over-rely on the system and reduce their attention or vigilance on the road.

Therefore, drivers should not depend on the system entirely and should always drive carefully and responsibly.

3.3 Components

3.3.1 Circuit Wiring Diagram

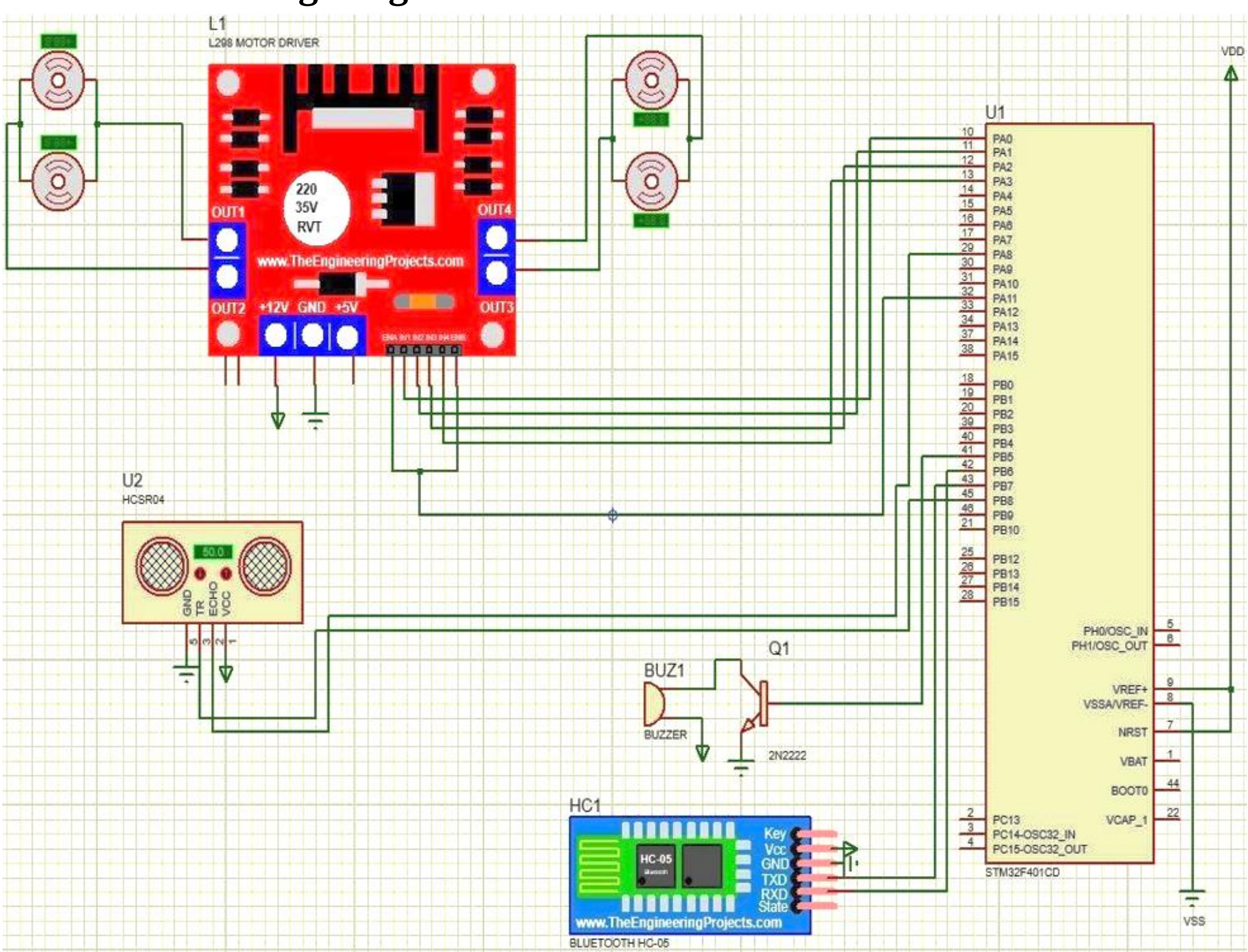


Figure 47: Circuit Wiring

As shown in figure 47, various components were used in this project to implement the desired functionality, such as sensors, ECU components, and actuators. Each of these components will be briefly described in the following sections, along with their role and purpose in the project.

3.3.2 HC-SR04 Ultrasonic Sensors

Figure 48: HC-SR04 Ultrasonic Sensor

The HC-SR04 Ultrasonic is a device that can measure distance by transmitting and receiving ultrasonic waves. Based on the received waves, we can determine whether there are any obstacles or vehicles in front of the car or not.

Working Principle

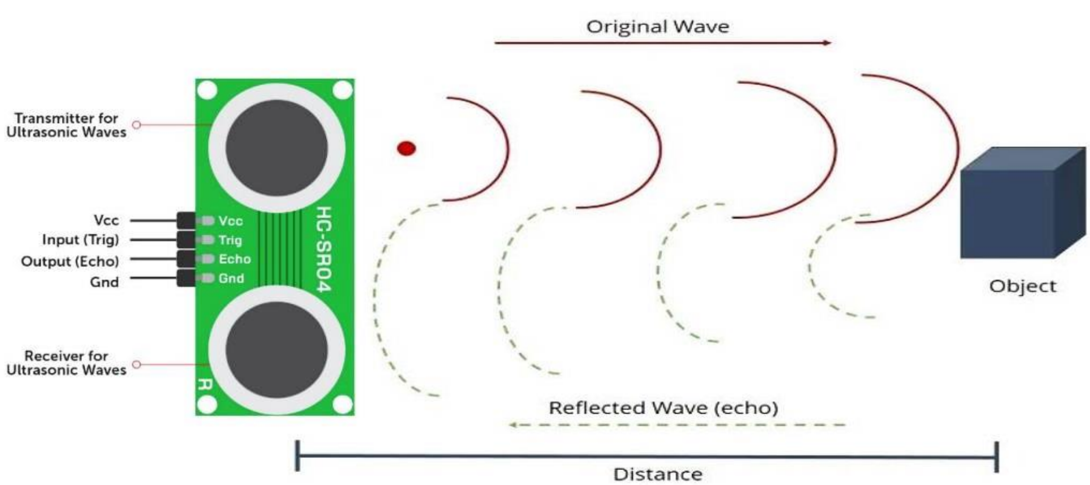


Figure 49: HC-SR04 Ultrasonic Working Principle

The HC-SR04 Ultrasonic consists of two modules: a transmitter and a receiver. The transmitter emits a high-frequency sound wave (40 kHz) beyond the human hearing range that travels through the air until it hits an object and reflects back. The receiver detects the reflected wave and calculates the time it took for the round trip.

The HC-SR04 Ultrasonic has a range of 2 cm to 400 cm, with an accuracy of 3 mm and a measuring angle of less than 15 degrees. It operates with a 5V DC power supply and consumes 15 mA of current. It has four pins: VCC, GND, Trig and Echo. The Trig pin is used to trigger the sound wave emission, and the Echo pin is used to receive the reflected wave signal.

The ultrasonic sensor is activated by triggering the sensor to start operation by sending a short pulse to the TRIGGER pin and it should be anything wider than 2uS (10us recommended). It can be even a few milliseconds.

After triggering, it transmits 8 pulses of sound waves at an ultrasonic frequency of 40KHz. The sound travels in, almost, straight-line path until it hits an object then it reflects back to the sensor module, which sends out a digital pulse on the echo pin that has a width equal to the travel time of sound going back and forth between the module and the hit object.

Then, by taking in the incoming echo pulse from the sensor and measuring its width to calculate the distance value between our sensor and the object in front of it, which is a pretty easy calculation to carry out once the echo pulse width is measured. Given that sound travels at a constant speed in the air, we can easily figure out the distance.

There are so many ways to get the distance reading using a microcontroller with the HC-SR04 ultrasonic sensor. One technique is Differential Double ICU. This works really well especially when you're measuring extremely short pulses. You'll be amazed by the resolution, accuracy, and precision of measurements.

By configuring 2 Input Capture Unit (ICU) channels, one channel is for rising edge detection and the other channel is for falling edge detection. Now, as the number of counts of the reflected pulse width present, multiply by a 1-tick time (1 us), the result is the overall round-trip time is known. By dividing it over 2 results in knowing the reflected wave trip time.

According this formula: $\text{Distance} = \text{time} * \text{speed}$

The speed of the sound in the air is 340 m/s (or 34000 cm/s) and the time is what was just measured, the distance is then can be obtained.

3.3.3 DC Motor Actuator

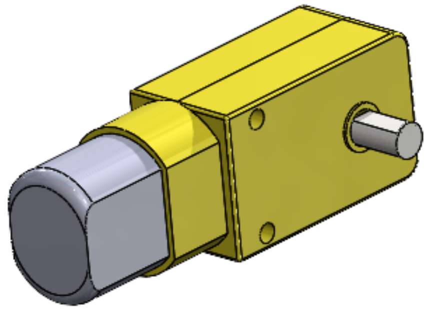


Figure 50: DC Motor

DC motors have a high-speed control capability. They can vary their speed widely by adjusting the voltage of the armature or the field. This allows DC motors to provide the accuracy needed for many industrial applications.

Specifications

- Operating voltage: 3-12 V DC.
- Recommended operating voltage: 6-8 V DC.
- Maximum torque: 800gf cm min.
- Maximum load current: 250 mA.
- Size (Approximately): 7x2.2x1.9 cm.
- Rotate speed 125 RPM.
- Load current: 70 mA.
- No-load speed: 1:48 (3V time).
- Weight: 169 g.

3.3.4 H-Bridge L298 Motor Driver

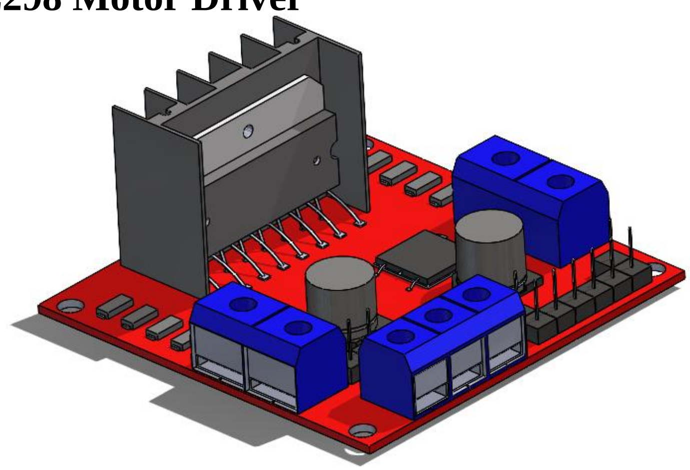


Figure 51: H-Bridge L298 Motor Driver

The H-Bridge L298 motor driver has two H-Bridges where we can control 2 DC motors at the same time. It allows us to control the speed and the direction of each DC motor simultaneously.

Working Principle

Motor Driver Module Pins: • Power Supply Pins • Control Pins • Output Pins

Table 1: Power Supply Pins

VCC	VCC pin is used to supply power to the motor. Its input voltage is between 5 to 35V.
GND	GND is a ground pin. It needs to be connected to the power supply ground(negative).
+5V	+5V pin supplies power for the switching logic circuitry inside the L298N IC. If the 5V-EN jumper is in place, this pin acts as output and can be used to power up a microcontroller or other circuitry (sensor). If the 5V-EN jumper is removed, you need to connect it to the 5V power supply of the microcontroller.

Table 2: Control Pins

IN1 IN2	These pins are input pins of Motor A. These are used to control the rotating direction of Motor A. When one of them is HIGH and the other is LOW, motor A will start rotating in a particular direction. If both the inputs are either HIGH or LOW, the motor A will stop.
IN3 IN4	These pins are input pins of Motor B. These are used to control the rotating direction of Motor B. When one of them is HIGH and the other is LOW, motor B will start rotating in a particular direction. If both the inputs are either HIGH or LOW, the motor B will stop.
ENA	ENA pin is used to control the speed of Motor A. If a jumper is present on this pin, so the pin is connected to +5 V and the motor will be enabled, then motor A rotates maximum speed. If we remove the jumper, we need to connect this pin to a PWM input of the microcontroller. In that way, we can control the speed of Motor A. If we connect this pin to Ground the Motor A will be disabled.
ENB	ENB pin is used to control the speed of Motor B. If a jumper is present on this pin, so the pin is connected to +5 V and the motor will be enabled, then the Motor B rotates maximum speed. If we remove the jumper, we need to connect this pin to a PWM input of the microcontroller. In that way, we can control the speed of Motor B. If we connect this pin to Ground the Motor B will be disabled.

Table 3: Output Pins Table

OUT1 – OUT2	This terminal block will provide the output for Motor A.
OUT3 – OUT4	This terminal block will provide the output for Motor B.

Working Technique

L298n motor driver module uses the H-Bridge technique to control the direction of rotation of a DC motor. In this technique, H-Bridge controlled DC motor rotating direction by changing the polarity of its input voltage.

An H-Bridge circuit contains four switching elements, like transistors, with the motor at the center forming an H-like configuration. Input IN1, IN2, IN3, and IN4 pins actually control the switches of the H-Bridge circuit inside L298N IC.

We can change the direction of the current flow by activating two particular switches at the same time, this way we can change the rotation direction of the motor.

Working Case 1

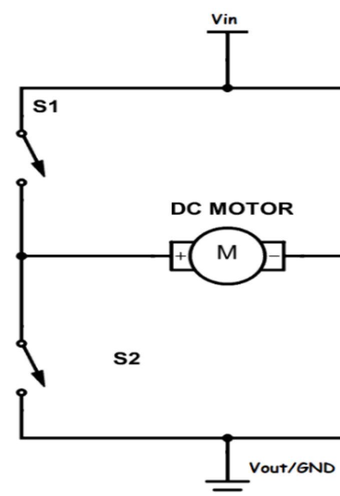


Figure 52: H-Bridge Case 1

As shown in figure 52, when S1, S2, S3, and S4 all switches are open then no current goes to the Motor terminals. So, in this condition, the motor is stopped (not working).

Working Case 2

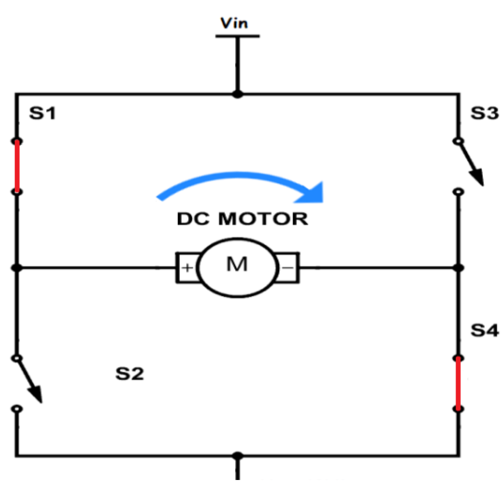


Figure 53: H-Bridge Case 2

As shown in figure 53, When the switch S1 and S4 are closed, then the motor left terminal is getting a positive (+) voltage and the motor right terminal is getting a negative (-) voltage. So, in this condition, the motor starts rotating in a particular direction (clockwise).

Working Case 3

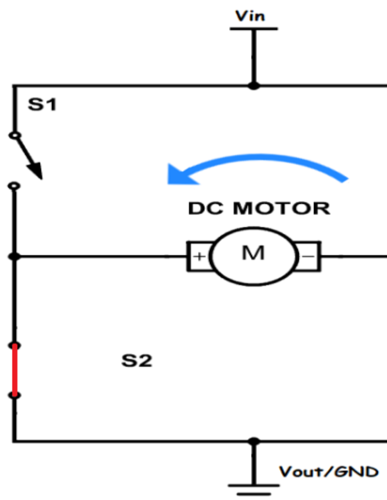


Figure 54: H-Bridge Case 3

As shown in figure 54, when S2 and S3 switches are closed, then the right motor terminal is getting a positive (+) voltage and the left motor terminal is getting a negative (-) voltage. So, in this condition, the motor starts rotating in a particular direction (anti-clockwise).

Specifications

- Double H-Bridge drive.
- Logical voltage: 5V.
- Drive voltage: 5-35 V.
- Logic current: 0-36 mA.
- Drive current: 2A per channel (MAX single bridge).
- Max power: 25W.
- Weight: 30g.
- Size: 43*43*27 mm.

3.3.5 Buzzer

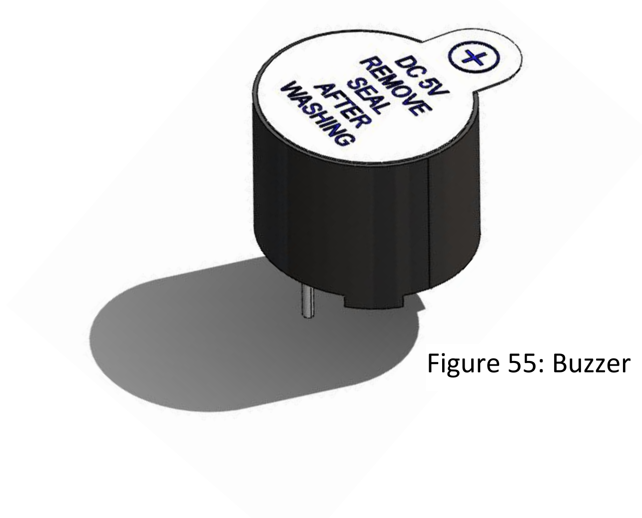


Figure 55: Buzzer

A buzzer is a device that produces sound waves by vibrating a membrane or a metal plate. It can be used as an acoustic actuator to output a warning sound in some situations. For example, if a vehicle is approaching another vehicle too close, a buzzer can alert the driver to take action to avoid a collision. The pitch and volume of the sound depend on the frequency and voltage of the electric signal that drives the buzzer.

Specifications

- Rated voltage: 5V.
- Working voltage: 5V
- Maximum rated current: 30mA.
- Size: Diameter 12 mm* height 9.5 mm.
- The vibration frequency: 2300±300 Hz.
- Working temperature: -30 to +85 °C
- The sound output: 10cm, 85db.

3.3.6 HC-05 Bluetooth Module

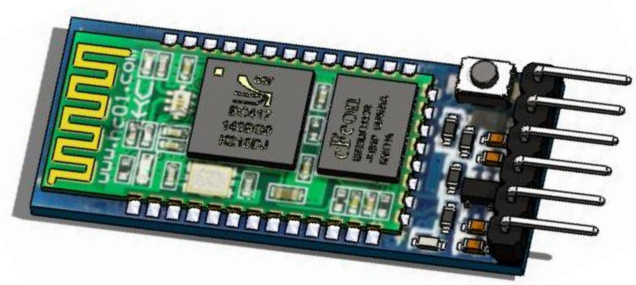


Figure 56: HC-05 Bluetooth Module

The HC-05 Bluetooth module is a communication component that is used to communicate with another Bluetooth-supported devices, such as mobile device applications. This module is a transceiver as it can transmit and receive information to another device.

Specifications

- Permit pairing device to connect as default.
- Auto-pairing PINCODE: “0000” as default.
- UART interface with programmable baud rate.
- Low Power 1.8V Operation, 1.8 to 3.6V I/O.
- Auto-connect to the last device on power as default.
- PIO control.
- With integrated antenna.
- With edge connector.
- Up to +4dBm RF transmit power.

- Typical -80dBm sensitivity.
- Default Baud rate: 38400, Data bits: 8, Stop bit: 1, Parity: No parity, Data control: has.
- Supported baud rate: 9600, 19200, 38400, 57600, 115200, 230400, 460800.
- Given a rising pulse in PIO0, the device will be disconnected.
- Status instruction port PIO1: low-disconnected, high-connected.
- PIO10 and PIO11 can be connected to red, and blue led separately. When master and slave are paired, red and blue led blinks 1time/2s in interval, while disconnected only blue led blinks 2times/s.
- Auto-reconnect in 30 min when disconnected as a result of beyond the range of connection.

3.4 Design Overview

3.4.1 Introduction

We will provide an overview of the design of our ADAS project. We will discuss the system architecture and the main components of the system, as well as the layered architecture and state-machine design used to model and manage its behavior. Our goal is to provide a clear and concise overview of how the system works and how its different parts fit together to achieve the desired functionality.

3.4.2 System Architecture

Our ADAS project is designed using a modular and layered architecture. The system is composed of several main components, including sensors, actuators, processing units, and communication interfaces. These components work together to collect data from the environment, process it to make decisions, and act on those decisions to achieve the desired behavior.

The system is organized into several layers, including a hardware layer, a driver layer, an operating system layer, and an application layer. Each layer is responsible for a specific aspect of the system’s operation and provides services to the layers above it. The layered architecture allows us to manage the complexity of the system and to develop and test each layer independently.

In the following sections, we will provide more detailed information about the layered architecture and state-machine design used in our system.

3.4.3 Layered Architecture

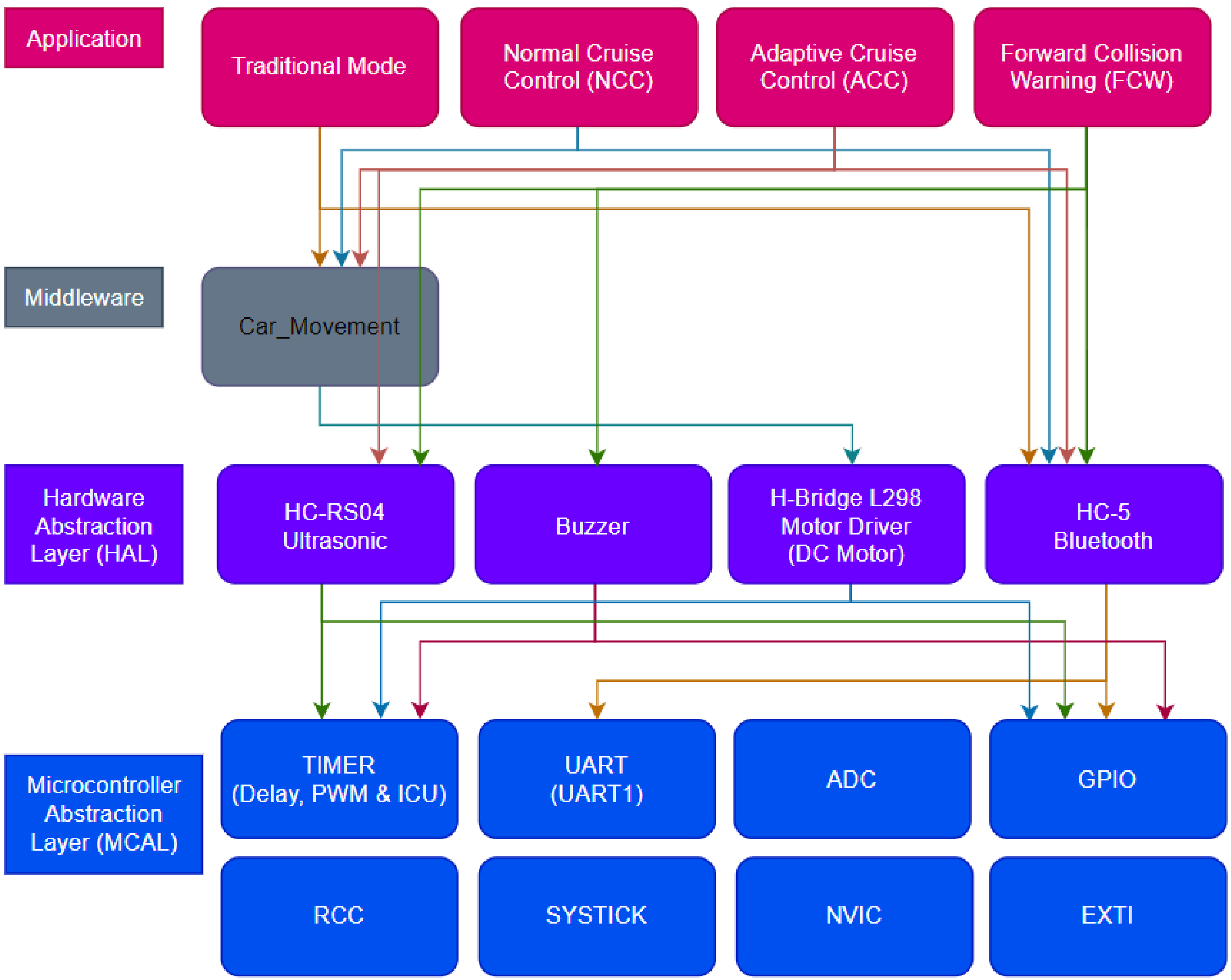


Figure 57: Layered Architecture

As shown in figure 57, These are the different layers we have in the project, with the used modules/peripherals in each layer respectively.

3.4.4 High-Level Design

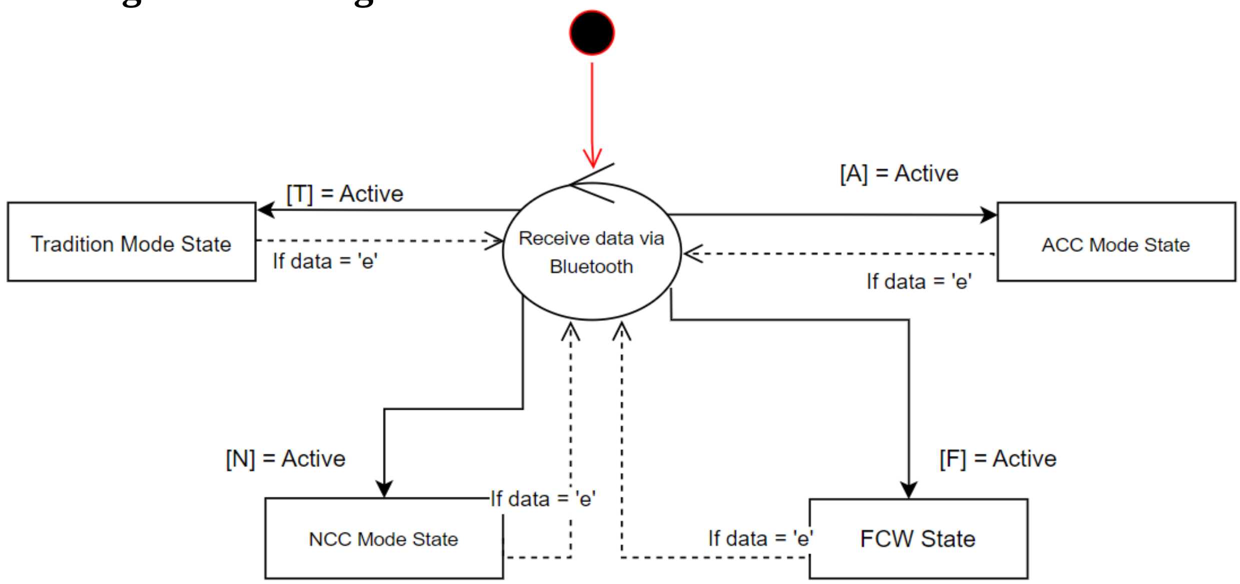


Figure 58: High-Level Design

High-Level Design is a description of the system's architecture, functionality, and interfaces. It defines the components of the system, how they interact with each other and with external entities, and what the key requirements and constraints for each component are.

As shown in figure 58, the project's high-level design of the system's various states as well as the workflow and transitions between different states based on the data received via Bluetooth. As previously discussed in the preceding sections, each mode has specific control commands that are only effective when that mode is active. Any control command that is invalid or unused is simply ignored by the system. The initial and default system's state is **Traditional Mode**.

3.5 API Documentation

3.5.1Doxygen

Introduction

Doxygen is a free tool for generating documentation from annotated source code. It can generate the documentation in various formats, such as HTML (Hyper Text Markup Language), LaTeX, RTF (Rich Text Format), XML (Extensible markup language), and man (manual) pages from comments in various programming languages such as C, C++, Java, PHP, and Python. Doxygen is widely used by software developers who want to create high-quality and easy-to-read documentation for their projects. Doxygen supports a variety of features, such as cross-referencing, inheritance diagrams, class hierarchies, collaboration graphs, and automatic link generation.

Advantages

- It can produce various output formats, such as HTML, PDF, LaTeX, etc.
- It can extract information from comments, annotations, and tags in the code.
- It can create cross-references, diagrams, and graphs to illustrate the code structure and relationships.
- It can be integrated with other tools and frameworks, such as Git, Sphinx, etc.

Disadvantages

- It requires extra effort and discipline to write and maintain the comments and tags in the code.
- It may not support some languages or features that are specific to certain platforms or frameworks.
- It may generate inaccurate or incomplete documentation if the code is not well-written or commented.
- It may not suit the preferences or needs of some users or audiences who prefer different styles or formats of documentation.

3.5.2 Examples

- Example 1

```
/**
 * @brief Sets a certain pin's mode on a speific port
 * @param[in] A_u8PortID The port that the pin belongs to
 * @param[in] A_u8PinID The pin to update its mode
 * @param[in] A_u8Mode The mode to apply the pin
 */
void MGPIOx_vSetPinMode(u8_t A_u8PortID, u8_t A_u8PinID, u8_t A_u8Mode);
```

Figure 59: Source Code of Functions

◆ MGPIOx_vSetPinMode()

```
void MGPIOx_vSetPinMode ( u8_t A_u8PortID,
                          u8_t A_u8PinID,
                          u8_t A_u8Mode
                        )
```

Sets a certain pin's mode on a speific port.

Parameters

[in] **A_u8PortID** The port that the pin belongs to
[in] **A_u8PinID** The pin to update its mode
[in] **A_u8Mode** The mode to apply the pin

Definition at line **71** of file **GPIO_program.c**.

```
{

    switch (A_u8PortID)
    {
        case GPIO_PORTA:
            CLR_BITs(GPIOA->MODERx, 0b11, A_u8PinID, 2);
            SET_BITs(GPIOA->MODERx, A_u8Mode, A_u8PinID, 2);
            break;

        case GPIO_PORTB:
            CLR_BITs(GPIOB->MODERx, 0b11, A_u8PinID, 2);
            SET_BITs(GPIOB->MODERx, A_u8Mode, A_u8PinID, 2);
            break;

        case GPIO_PORTC:
            CLR_BITs(GPIOC->MODERx, 0b11, A_u8PinID, 2);
            SET_BITs(GPIOC->MODERx, A_u8Mode, A_u8PinID, 2);
            break;
    }

}
```

References **CLR_BITs**, **GPIO_PORTA**, **GPIO_PORTB**, **GPIO_PORTC**, **GPIOA**, **GPIOB**, **GPIOC**, and **SET_BITs**.

Referenced by **MGPIOx_vInit()**.

Figure 60: Generated Documentation

-Example 2

```
/**
 * @struct LED_LEDConfiguration
 * @brief LED configuration structure for LED initialization
 * @details This structure is used to initialize the LED with a certain port and pin
 */
typedef struct
{
    /**
     * @brief Initialize the LED on a certain port
     */
    u8_t u8Port;
    /**
     * @brief Initialize the LED on a certain pin
     */
    u8_t u8Pin;
} LED_LEDConfiguration;
```

Figure 61: Source Code of Structures

Data Fields

LED_LEDConfiguration Struct Reference

LED configuration structure for LED initialization. More...

`#include "COTS/HAL/LED/LED_interface.h"`

Data Fields

u8_t **u8Port**
Initialize the LED on a certain port. More...

u8_t **u8Pin**
Initialize the LED on a certain pin. More...

Detailed Description

LED configuration structure for LED initialization.

This structure is used to initialize the LED with a certain port and pin

Definition at line 17 of file LED_interface.h.

Field Documentation

◆ u8Port

u8_t u8Port

Initialize the LED on a certain port.

Definition at line 22 of file LED_interface.h.

◆ u8Pin

u8_t u8Pin

Initialize the LED on a certain pin.

Definition at line 26 of file LED_interface.h.

Figure 62: Generated Documentation

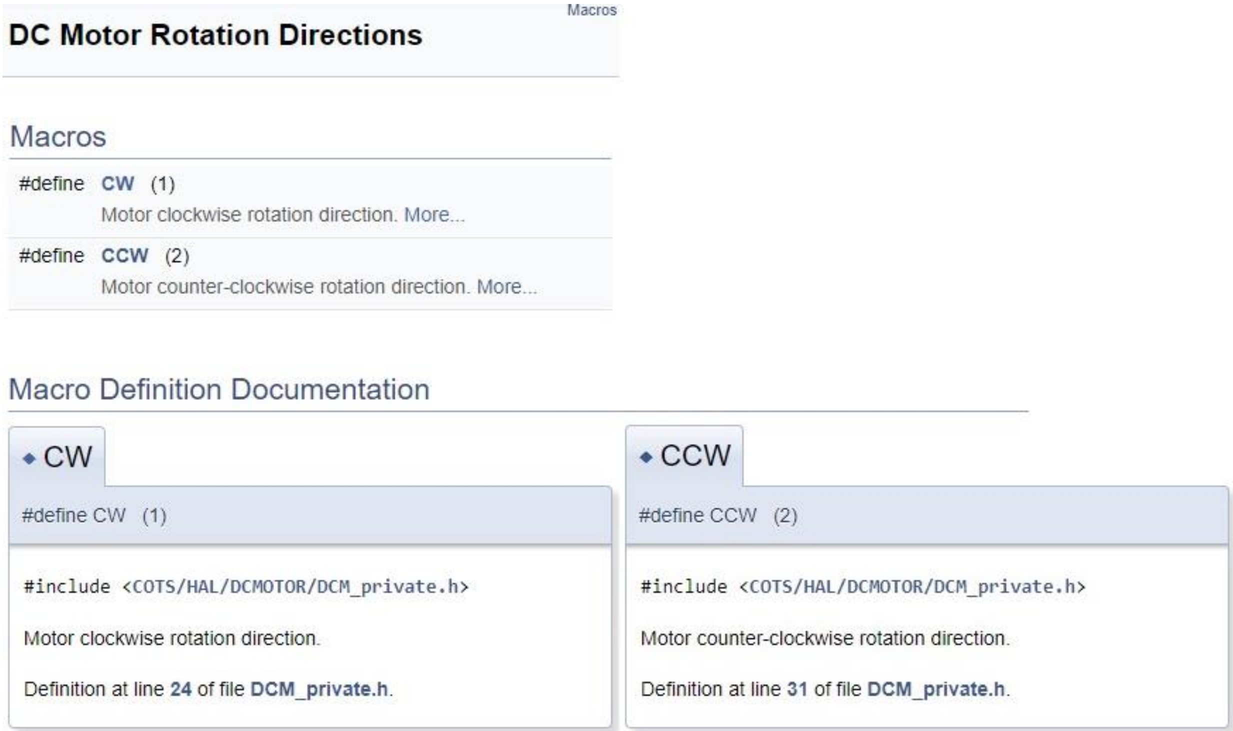
-Example 3

```
/**
 * @defgroup motor_rotation_direction DC Motor Rotation Directions
 *
 * @{
 */

/**
 * @def CW
 * @brief Motor clockwise rotation direction
 *
 */
#define CW (1)

/**
 * @def CCW
 * @brief Motor counter-clockwise rotation direction
 *
 */
#define CCW (2)
/** @} */
```

Figure 63: Source Code of Macros



Macros

DC Motor Rotation Directions

Macros

#define **CW** (1)
Motor clockwise rotation direction. More...

#define **CCW** (2)
Motor counter-clockwise rotation direction. More...

Macro Definition Documentation

◆ CW

#define CW (1)

```
#include <COTS/HAL/DCMOTOR/DCM_private.h>
```

Motor clockwise rotation direction.

Definition at line **24** of file **DCM_private.h**.

◆ CCW

#define CCW (2)

```
#include <COTS/HAL/DCMOTOR/DCM_private.h>
```

Motor counter-clockwise rotation direction.

Definition at line **31** of file **DCM_private.h**.

Figure 64: Generated Documentation

CH4 Driver Monitoring System

4.1 Introduction

This project demonstrates the potential of machine learning-based technologies to significantly improve road safety and prevent accidents caused by human error. The development of the DMS systems represents a step forward in the integration of advanced technologies into future automotive safety systems, and their potential impact on road safety should not be underestimated.

We aim to contribute to the improvement of road safety by applying two powerful APIs (Application Program Interface) that can be integrated into existing systems. The face recognition API and drowsiness detection API have the potential to make a significant impact on road safety, and we hope that they will be widely adopted in the future. In the following sections of this documentation, we will provide detailed information on the development and implementation of these APIs, as well as their potential applications and limitations.

We aim by this project to apply algorithms for the purpose of improving driver safety; a face recognition and a drowsiness detection system and for practical application we used the actuators with a Raspberry Pi.

The face recognition system will be used to identify driver’s face, while the drowsiness detection system will be used to monitor the driver's level of alertness and prevent accidents caused by drowsy driving.

4.2 History of driver monitoring system

DMS was first introduced by Toyota in 2006 for its and Lexus' latest models. It was first offered in Japan on the GS 450h. The system's functions co-operate with the pre-collision system (PCS). The system uses infrared sensors to monitor driver attentiveness.

Specifically, the driver monitoring system includes a CCD camera placed on the steering column which tracks the face, via infrared LED detectors. If the driver is not paying attention to the road ahead and a dangerous situation is detected, the system will warn the driver by flashing lights, warning sounds. If no action is taken, the vehicle will apply the brakes (a warning alarm will sound followed by a brief automatic application of the braking system). This system is said to be the first of its kind.

In 2008, the Toyota Crown system went further and can detect if the driver is becoming sleepy by monitoring the eyelids.

In 2017, Cadillac released their Super Cruise system. Which allowed hands free driving at highway speeds on specially mapped highways. In order to ensure that the driver continued to pay attention to the road, they included Seeing Machines DMS, this was initially only available in the CT6.

In 2019, BMW introduced an Extended Traffic Jam Assistant System in almost its entire range of car models. This allows driving at up to 37 mph.

Figure 65: The driver monitoring system on the Lexus 2006, LS 600h

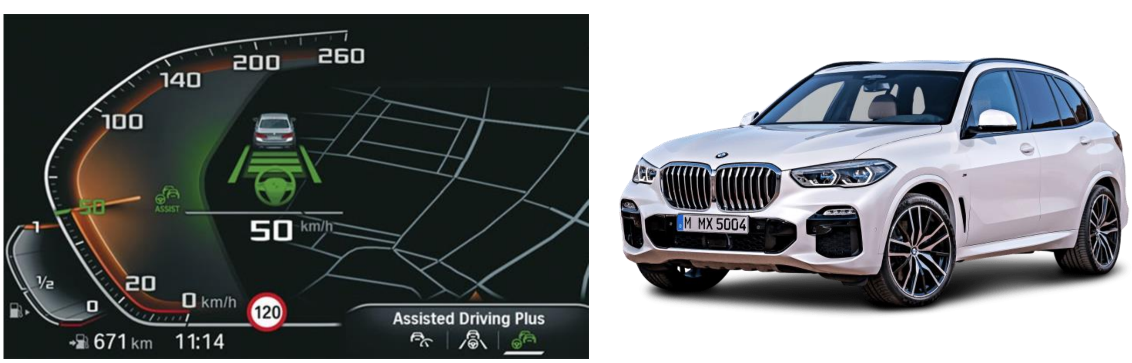


Figure 66: BMW extended traffic jam assistant system, x5 2019

4.3 Importance of driver monitoring system

Facial recognition has become an increasingly popular technology in recent years, with applications ranging from security systems to social media platforms. In this project, we aim to leverage this technology to enhance road safety by identifying individuals in vehicles. By using a camera mounted inside the vehicle, the face recognition system will be able to identify the driver and passengers and ensure that they are authorized to be in the vehicle. This will help prevent unauthorized access to vehicles and improve overall security.

Drowsy driving is a serious problem that affects millions of people around the world. According to the National Highway Traffic Safety Administration, “drowsy driving is a factor in up to 32000/year in our Country giving 8000/year are KILLED, And 7,500 fatal crashes per year in the United States alone”.

Road safety is a critical issue that affects millions of people worldwide. According to the World Health Organization, road traffic accidents are a leading cause of death and disability worldwide, especially among young people. In addition to human suffering, road accidents also have significant economic and social costs.

Distracted driving examples:

- Using Phones and busy hands.
- Very long eye distracted as sleepy eyes.
- High speed in danger roads.
- Dangerous driving habits.

To address this issue, we used some API‘s that uses machine learning algorithms to analyze facial expressions and detect signs of drowsiness. By monitoring the driver's level of alertness and issuing alerts when necessary, this API can help prevent accidents caused by drowsy driving.

4.4 System Main Hardware Components

4.4.1 Raspberry Pi 4 model B

Definition

The Raspberry Pi is a low cost, credit-card sized computer that plugs into a computer monitor or TV, and uses a standard keyboard and mouse. It’s capable of doing everything you’d expect a desktop computer to do, from browsing the internet and playing high-definition video, to making spreadsheets, word-processing, and playing games.

What's more, the Raspberry Pi has the ability to interact with the outside world, and has been used in a wide array of digital maker projects, from music machines and parent detectors to weather stations and tweeting birdhouses with infra-red cameras.

Specifications

- Memory: 4GB LPDDR4
- Environment: Operating temperature 0–50ºC
- Processor: Broadcom BCM2711, quad-core Cortex-A72 (ARM v8) 64-bit SoC @ 1.5GHz
- Connectivity: 2.4 GHz and 5.0 GHz IEEE 802.11b/g/n/ac wireless LAN, Bluetooth 5.0, BLE Gigabit Ethernet 2 × USB 3.0 ports 2 × USB 2.0 ports.
- SD card support: Micro SD card slot for loading operating system and data storage (32 GB)
- Input power: 5V DC via USB-C connector (minimum 3A*) , 5V DC via GPIO header (minimum 3A*)
- GPIO: Standard 40-pin GPIO header

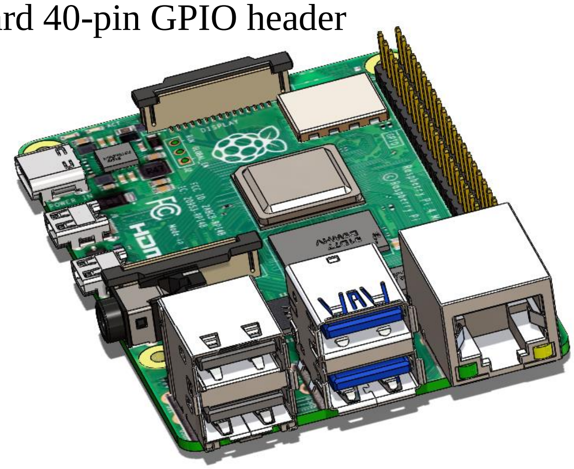


Figure 67: Raspberry pi 4 model B

4.4.2 Camera

Definition

This Camera Module is a custom designed add-on for Raspberry Pi. The interface uses the dedicated CSI (Camera Serial Interface) interface, which was designed especially for interfacing with cameras. The CSI bus is capable of extremely high data rates, and it exclusively carries pixel data. This bus travels along the ribbon cable that attaches the camera board to the Pi.

The sensor itself has a native resolution of 5 megapixels and has an adjustable focus lens onboard. The camera is supported in the latest version of Raspberry Pi’s preferred operating system.

Specifications

- Resolution: 5MP
- Image sensor chip: OV5647
- CCD size: 1/4 inch
- Aperture (F): 1.8
- Come with 15cm length of FFC ribbon cable
- Focal length: 3.6mm
- Field of view (diagonal): 60°
- Sensor resolution: 1080p
- Size: 25mm × 24mm

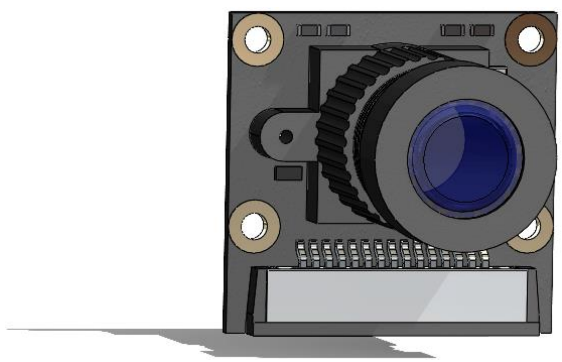


Figure 68: Camera module

4.4.3 Buzzer

Definition

An audio signaling device like a beeper or buzzer may be electromechanical or piezoelectric or mechanical type. The main function of this is to convert the signal from audio to sound. Generally, it is powered through DC voltage and used in timers, alarm devices, printers, alarms, computers, etc. Based on the various designs, it can generate different sounds like alarm, music, bell & siren.

Specifications

- Color is black
- The frequency range is 3,300Hz
- The supply current is below 15mA
- Operating Temperature ranges from – 20° C to +60°Cv
- Operating voltage ranges from 3V to 24V DC
- The sound pressure level is 85dBA or 10cm

Figure 69: Buzzer

4.4.4 Power Supply

Definition

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

Specifications

- Output: 5V, 3 A
- Input frequency: 50 / 60 HZ
- On / Off switch
- Input AC voltage: 100 – 240 V

Figure 70: Power supply

4.4.5 LCD

Definition

The term LCD stands for liquid crystal display. It is one kind of electronic display module used in an extensive range of applications The main benefits of using this module are inexpensive; simply programmable, animations, and there are no limitations for displaying custom characters, special and even animations, etc.

Specifications

- Operating Voltage: 4.7V to 5.3V
- Operating Current 1mA (without backlight)
- Can display (16x2) 32 Alphanumeric Characters
- Works in both 8-bit and 4-bit Mode
- Custom Characters Support

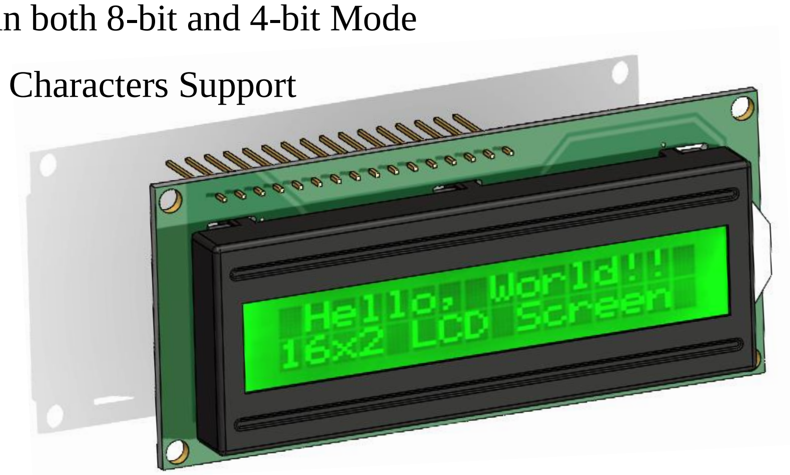


Figure 71: LCD 16x2

4.4.6. Potentiometer

Definition

A potentiometer is a three-terminal resistor with a sliding or rotating contact that forms an adjustable voltage divider.

If only two terminals are used, one end and the wiper, it acts as a variable resistor or rheostat.

Figure 72: Potentiometer

4.4.7. Jumper Wires

Definition

A jump wire is an electrical wire, or group of them in a cable, with a connector or pin at each end.

Specification:

- Male - male ends

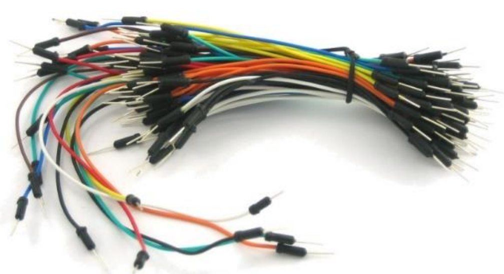


Figure 73: Jumper wire male – male

- Male- female ends

Figure 74: Jumper wire male – female

4.5 Theories behind driver monitoring system

4.5.1 Main Terms

We will review the main concepts in the following sections

Face Detection

Capturing in real time and find human face in it.

When dealing with human face we have to use it, this technology has been available for some years now and is being used all over the place. From cameras that make sure faces are focused before you take a picture, to Facebook when it tags people automatically once you upload a picture (before you did that manually remember?). Some shows like CSI used them to identify “bad guys” from security footage (ENHANCE! – then insert crime pun) or unlocking your phone by looking at it!

Data Gathering

Extract unique characteristics of driver’s face that it can use to differentiate him from another person, like eyes, mouth, nose, etc.

Data Comparison

Despite variations in light or expression, it will compare those unique features to all the features of all the people you know.

Face Recognition

Capturing in real time and find driver’s face in it and determines his name.

Recognition

When you look at an apple, your mind immediately tells you: that is an apple. This process is recognition in the simplest of terms. So, what’s facial recognition? The same, but for faces, obviously. But the real question is: **How can a computer recognize a face?**

To answer, let us take a real-life example. When you meet someone for the first time, you don’t know who that person is at once, right? While he's talking to you or shaking your hand, you’re looking at his face: eyes, nose, mouth, skin tone … This process is your mind gathering data and training for face recognition. Next, that person tells you that his name is Ahmad. So, your brain has already gotten the face data, and now it has learned that this data belongs to Ahmad. The next time you see Ahmad or see a picture of his face, your mind will follow this exact process, as shown in figure below.

It will determine, “Hey, that’s Ahmad!”

Then, the more you meet Ahmad, the more data you will collect about him, and the quicker your mind will be able to recognize him. Or, at least you should. Whether or not you are good with names is another story. Here is when it gets better.

Our human brains are wired to do all these things automatically. In fact, we are very good at detecting faces almost everywhere. Computers aren’t able, yet, to do this automatically, so we need to teach them how to do it step-by-step.

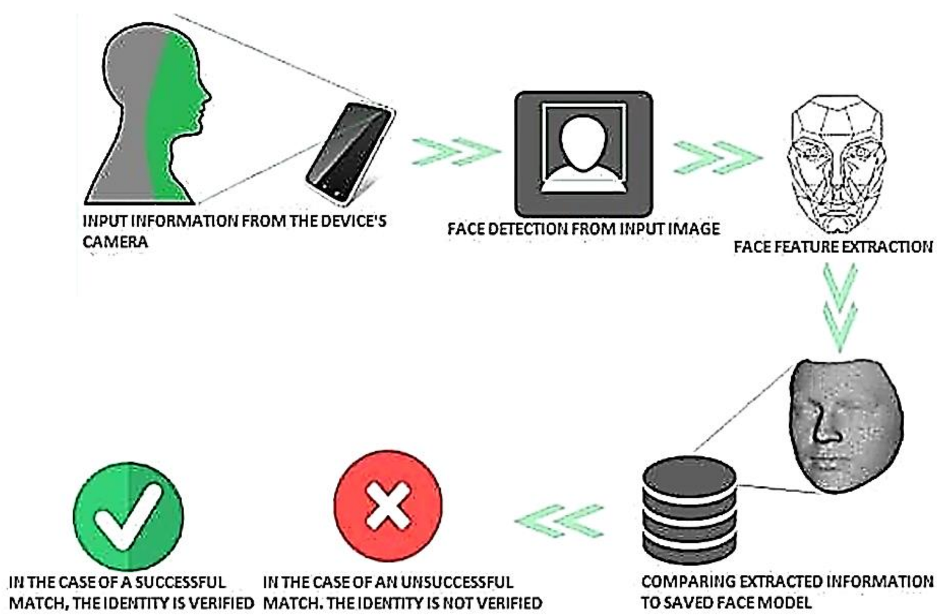


Figure 75: How face recognition system works

Drowsiness

It process the driver’s eyes and label his state to a drowsy or awake by implementing sort of algorithms going to be discussed later.

Yawning

It process the driver mouth and label his state to yawning or not by implementing sort of algorithms going to be discussed later.

4.5.2 Drowsiness detection working principle

Is a technology that uses computer vision and machine learning techniques to analyze facial expressions and movements to detect signs of drowsiness in individuals? It is often used in the context of driver safety, where it can detect and alert drivers who are at risk of falling asleep at the wheel.

Drowsiness detection algorithms typically use computer vision techniques to track facial features such as the eyes, eyebrows, and mouth, and analyze changes in their position or movement over time. These changes can indicate the onset of drowsiness, such as drooping eyelids or head nodding. Machine learning models are then used to classify these changes as indicative of drowsiness or not, As Shown in the below flow chart.

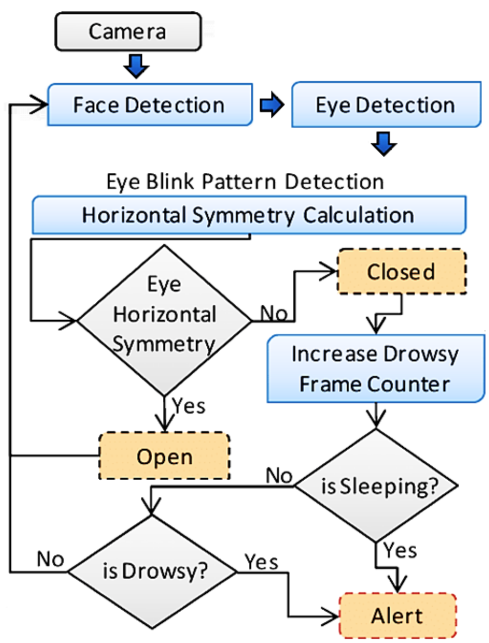


Figure 76: Drowsiness Detection System

4.6 Algorithms

4.6.1 Theory behind Face Detection Classifiers

A computer program that decides whether an image is a positive image (face image) or negative image (non-face image) is called a classifier. A classifier is trained on hundreds of thousands of face and non-face images to learn how to classify a new image correctly.

Now, we will discuss the various Face Detection methods in OpenCV and Dlib and compare the methods quantitatively:

- Haar Cascade Classifiers using OpenCV.
- HOG (Histogram of Oriented Gradients) using Dlib.
- Deep Learning based Face Detector in OpenCV.
- Convolutional Neural Networks using Dlib.
- Eye Aspect Ratio (continued after 1 Haar Cascade).

4.6.2 Haar Cascade Classifiers using OpenCV

The Haar Classifier is a machine learning based approach, an algorithm created by Paul Viola and Michael Jones; which are trained from many positive images (with - faces) and negatives images (without faces).

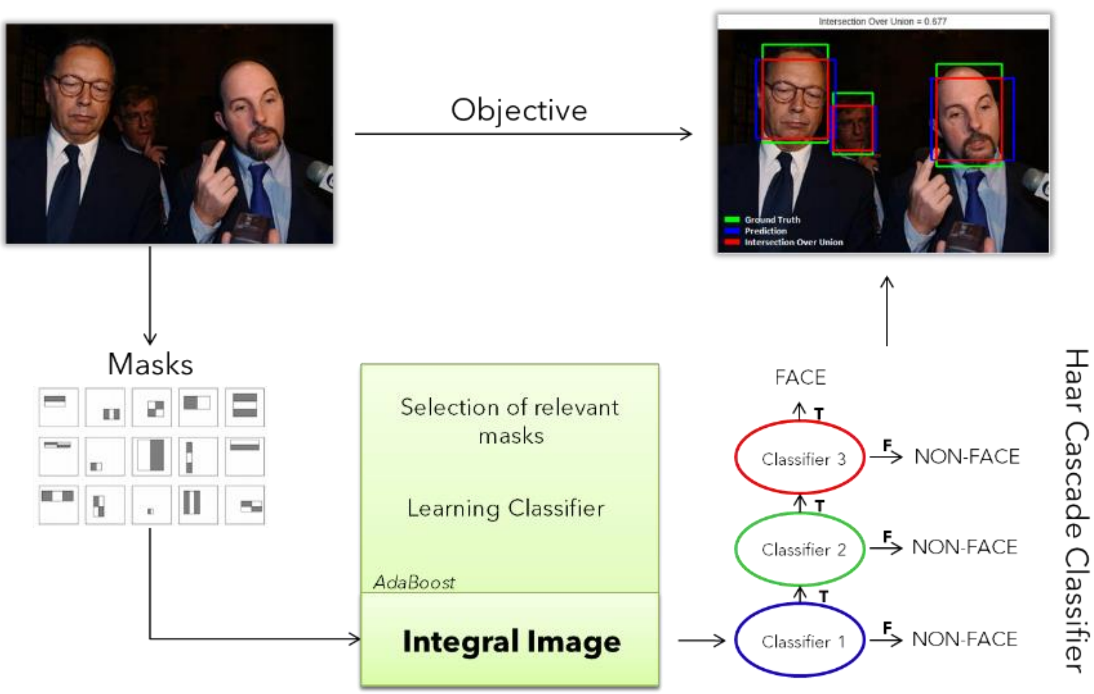
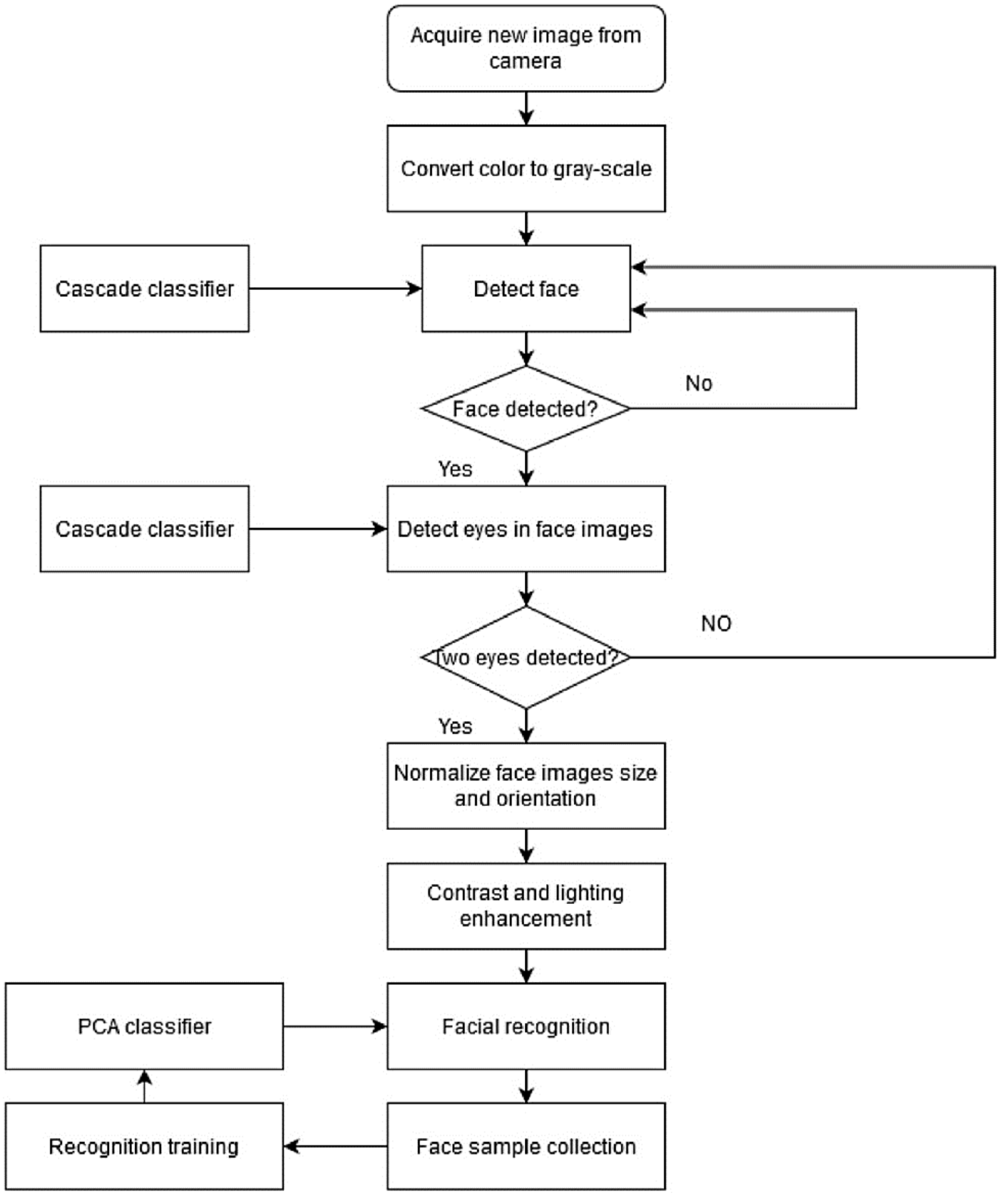


Figure 77: Face Detection using Haar feature-based cascade classifiers

Cascade classifiers are trained on a few hundred sample images of image that contain the object we want to detect, and other images that do not contain those images. It starts by extracting Haar features from each image as shown below:



4.6.3 Histogram of Oriented Gradients (HOG) using Dlib

One of the most popular implements for face detection is offered by Dlib and uses a concept called Histogram of Oriented Gradients (HOG).

The model is built out of 5 HOG filters – front looking, left looking, right looking, front looking but rotated left, and a front looking but rotated right.

The dataset used for training, consists of 2825 images which are obtained from LFW dataset and manually annotated by Davis King, the author of Dlib.

The idea behind HOG is to extract features into a vector, and feed it into a classification algorithm like a Support Vector Machine for example that will assess whether a face (or any object you train it to recognize actually) is present in a region or not.

The features extracted are the distribution (histograms) of directions of gradients (oriented gradients) of the image. Gradients are typically large around edges and corners and allow us to detect those regions.

In the original paper, the process was implemented for human body detection, and the detection chain was the following:

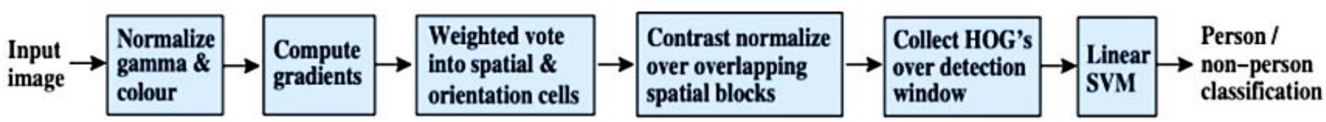


Figure 78: HOG filters used for detection

4.6.4 Deep Learning based Face Detector in OpenCV

This model was included in OpenCV from version 3.3. It is based on Single-Shot-Multibox detector and uses ResNet-10 Architecture as the backbone. The model was trained using images available from the web.

OpenCV provides 2 models for this face detector.

- Floating point 16 version of the original caffe implementation.
- 8-bit quantized version using Tensor flow

4.6.5 Convolutional Neural Networks using Dlib.

CNN stands for Convolutional Neural Network, which is a type of deep learning algorithm that is commonly used in image recognition and classification tasks. A CNN consists of multiple layers of neurons that perform convolution and pooling operations on input images to extract features and learn patterns.

They are feed-forward neural network that are mostly used for computer vision. They offer an automated image pre-treatment as well as a dense neural network part. CNNs are special types of neural networks for processing data with grid-like topology.

It uses a Maximum-Margin Object Detector (MMOD) with CNN based features. The training process for this method is very simple and you don't need a large amount of data to train a custom object detector.

It uses a dataset manually labelled by its Author, Davis King, consisting of images from various datasets like ImageNet, PASCAL VOC, VGG, WIDER, Face Scrub. It contains 7220 images.

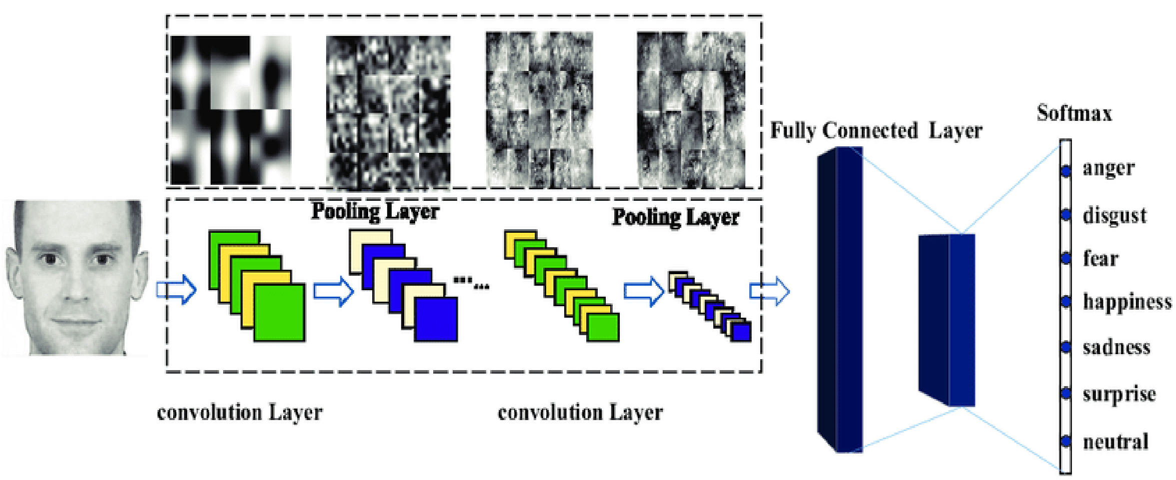


Figure 79: CNN Demonstration

Max-pooling layer: It performs spatial down-sampling of the feature map and retains only the most relevant information. See the picture below for a visual illustration of this operation. From a practical point of view, a pooling of size 2x2 with a stride of 2 gives good results on most applications. Having said that, other types of pooling exist, e.g., average pooling, median pooling, sum pooling.

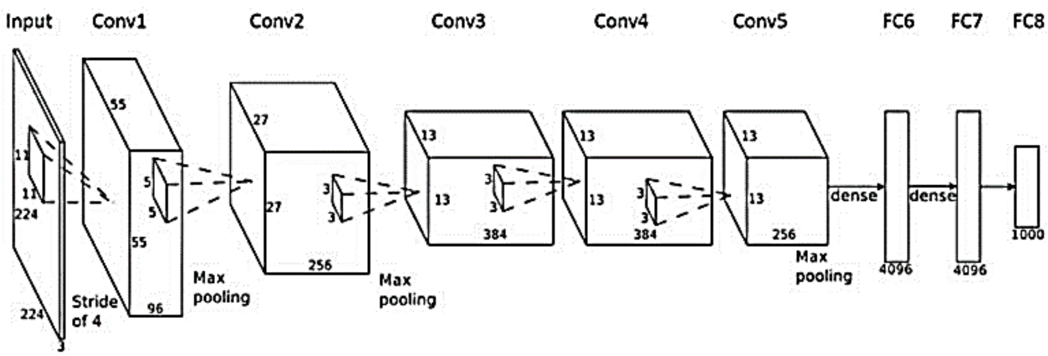


Figure 80: Max-Pooling layer

The architecture of the CNN is inspired by the visual cortex of animals. In previous approaches, a great part of the work was to select the filters in order to create the features in order to extract as much information from the image as possible. With the rise of deep learning and greater computation capacities, this work can now be automated.

The name of the CNNs comes from the fact that we convolve the initial image input with a set of filters. The parameter to choose remains the number of filters to apply, and the dimension of the filters. The dimension of the filter is called the stride length. Typical values for the stride lie between 2 and 5, See the following figure.

4.6.6 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.

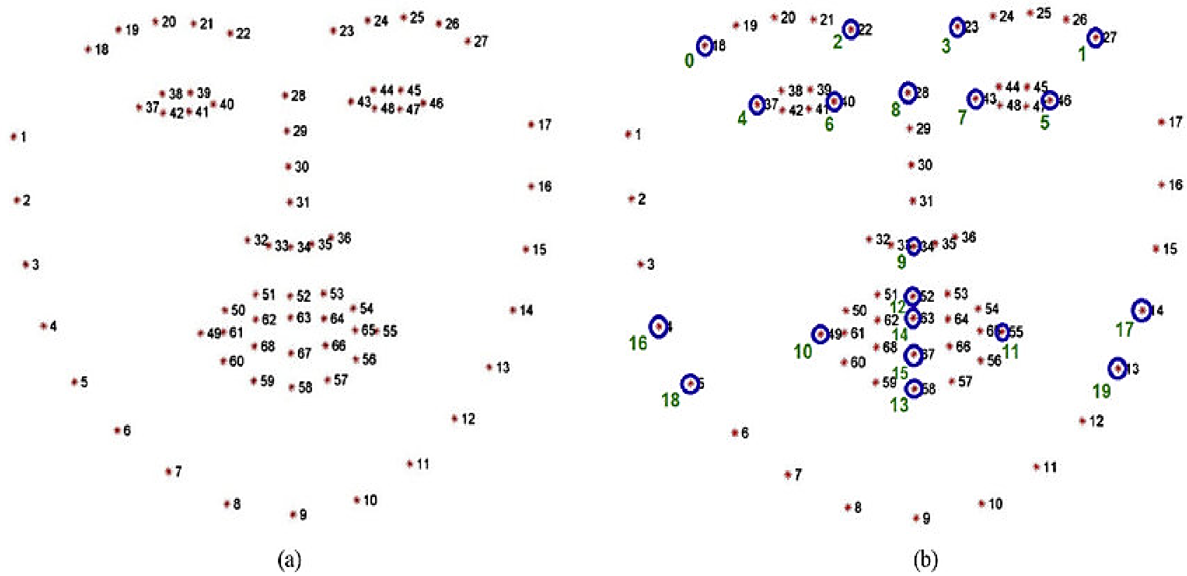
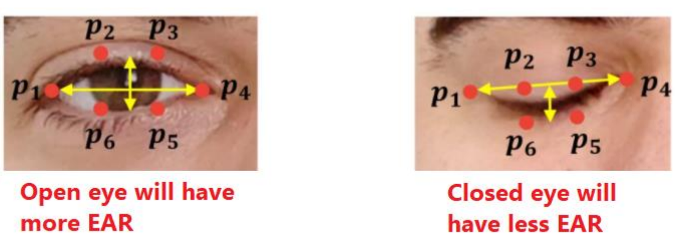


Figure 81: Facial landmark prediction

4.6.7 Eye Aspect ratio (EAR)

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 in the figure below. And by using the EAR we can detect if the eye of the human behind the drive wheel is closed or not or by how present his eyes is closed, as shown in the figure below.



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

Figure 83: Detection using eye aspect ratio algorithm

Figure 82: The EAR for a single eye formula

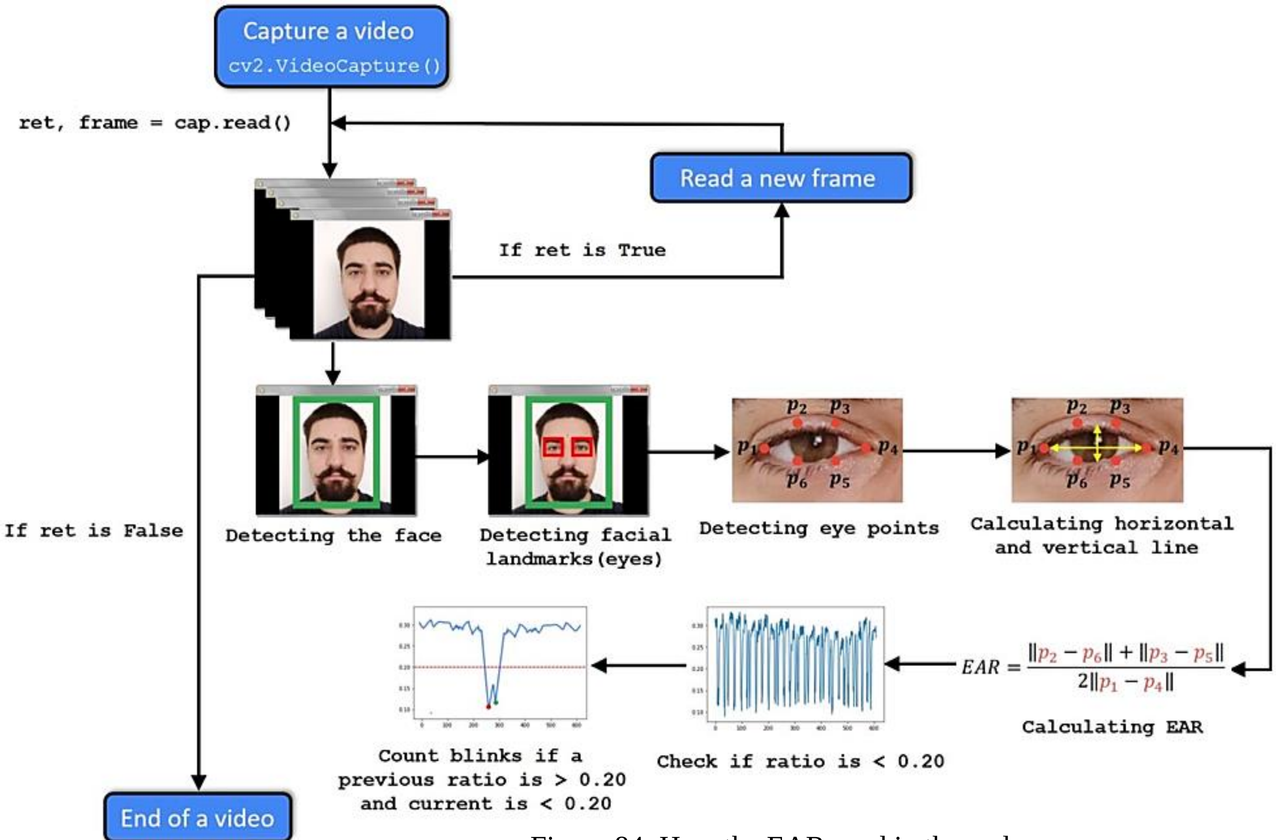


Figure 84: How the EAR used in the code

4.7 Hardware interface

To connect the 16×2 LCD with the Raspberry Pi 4, we will need the previous electronic components connected on breadboard , as the circuit diagram below of interfacing the 16×2 LCD with the Raspberry Pi 4 using fritzing software:

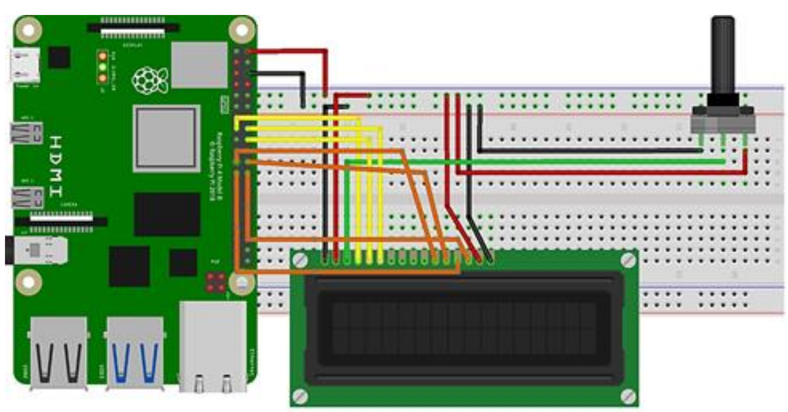


Figure 85: Schematic of Raspberry interfacing

Now with the help of jumper wires, we will connect the 16×2 LCD with the GPIO pins of Raspberry Pi 4 according to the table

Table () : GPIO pins and its BCM pins with LCD pins and Buzzer

GPIO pins of Raspberry Pi	BCM pins of Raspberry Pi	LCD pins
GPIO 22	15	RS
GPIO 24	18	RW
GPIO 23	16	E
GPIO 9	21	D4
GPIO 25	22	D5
GPIO 11	23	D6
GPIO 8	24	D7
Ground	6	K, VSS
5V	2	VDD, A
GPIO 26	17	Buzzer

4.8 Technical Issues and Solutions

As things was not going perfect as we expected, we faced some issues on both Windows and Linux.

4.8.1 Windows issues and solution

- **It was such as impossible to get all libraries API's, it took us very long time to solve,**
  - Finally we installed Visual Studio C++ packages and it had been solved

- **Some bash commands like git was not recognized by the windows**
  - After installing Git Bash program , system recognized it and issue resolved

- **Some commands to be run , virtual environment have to be activated**

- To activate virtual environment on Anaconda prompt , we used this website instructions [1]

4.8.2 Linux issues and solution

- **Controlling raspberry pi was not valid without using monitor, keyboard or mouse**
  - To solve this as we discussed before, we had to install VNC viewer and server software on both windows and raspberry OS (Linux), to be able to communicate with each other on the network.

- **Raspberry pi was not showing GUI on Windows screen**
  - We had to enable the VNC mode on raspberry pi settings.

- **We could not download opencv library easily, it took hours to download without successful trial.**
  - We encountered this website while searching for long times and it helped us to fix the issue [2]

- **Controlling raspberry pi and running code on it , was depending on network stability**
  - We had to ensure that network connection is strong enough to prevent errors.

- **Raspberry pi board temperature always high**
  - We bought heatsink and put on temperature sensitive components as processor.

- **May be VNC cannot view GUI of raspberry pi on windows**
    - Another way to run code remotely on raspberry pi is to use SSH protocol on Putty software.
    - Or, using any IDE as Thonny, and through the pigpio library lets you control the GPIO of one or more networked Pis from a laptop. The laptop may be Windows, Mac, Android, or Linux based - in fact it can run any operating system as long as it can run Python. The pigpio Python module allows control of the remote GPIO.
- **Raspberry pi cannot control LCD or buzzer directly**
    - To interface any LCD either its 16×2 or 16×4, we have to download the library of RPLCD which we can download by using the wget command [3]

4.9 System in action

We had achieved the Face Recognition in the last semester and using our work again and added the Drowsiness Section systems, then we applied the whole system in raspberry pi kit. Even with gloomy light the accuracy still high.

Now, full system including:

1- Face recognition phase
2- Drowsiness detection phase

Before testing both systems on Raspberry Pi OS (Linux), code must be tested on Windows using suitable IDE like Pycharm.

4.9.1 Face recognition testing on Windows

As shown in figure below, red frame around the face showing person's name and accuracy percentage in footer in different light conditions.

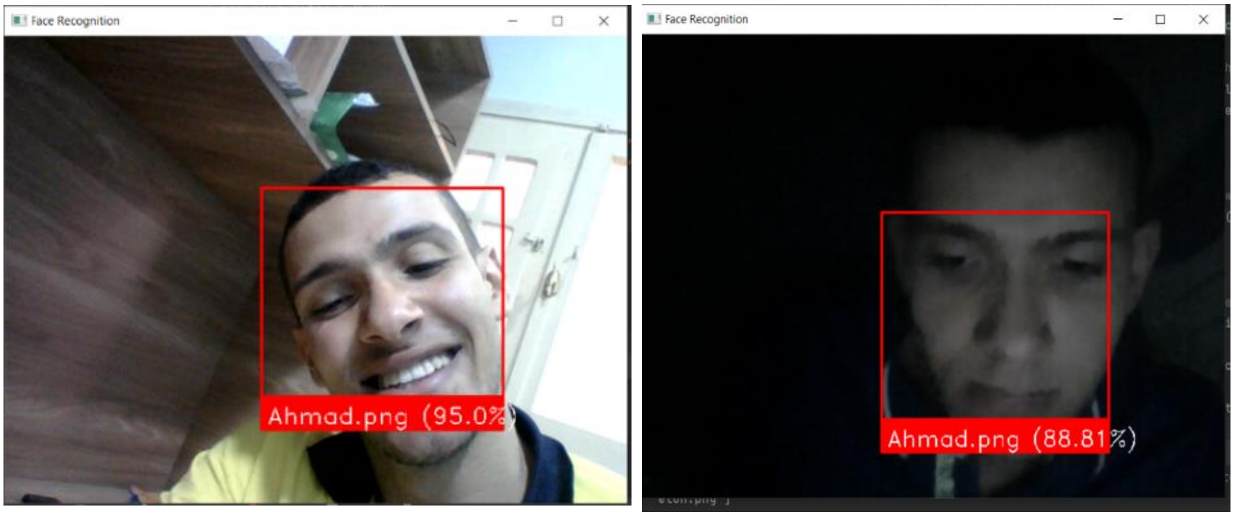


Figure 86: Output of Face recognition testing on Windows

4.9.2 Face recognition testing on Linux

This time some configurations must take place as below:

1- We should connect to raspberry pi remotely, so we could install VNC viewer software as a medium to conduct the operation on both Windows and Linux (raspberry pi), and check that both devices connected to the same network.

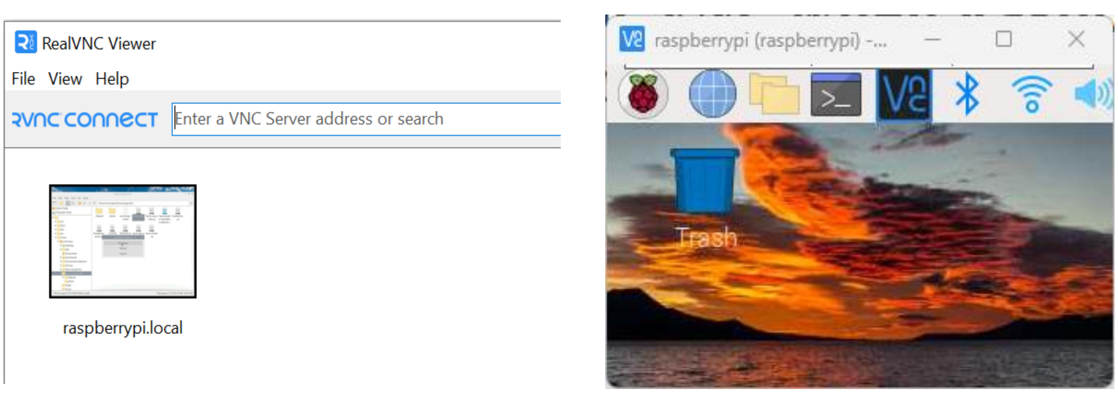


Figure 87: Accessing raspberry pi OS through VNC viewer

2- To access raspberry pi from Windows, we should enter host name in this case raspberrypi.local, then enter username and password. After that, we would be able to run code on raspberry pi IDE such as Thonny

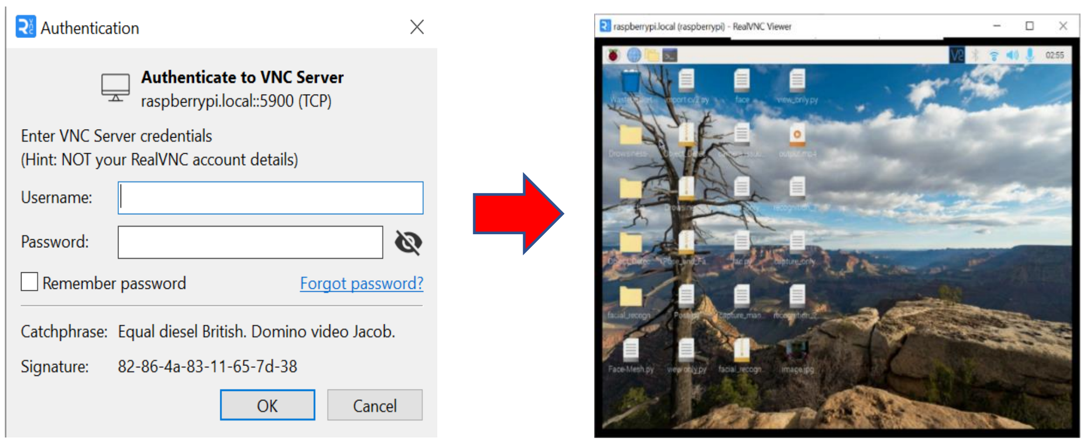


Figure 88: Accessing raspberry pi OS through entering hostname

3- Installing needed packages and code source to run the code, we could run the code successfully.

4- From Raspberry Pi Desktop Open your File Manager by clicking the folder icon, and Navigate to the facial recognition folder and then the dataset folder. Right-Click within the dataset folder and select New Folder.

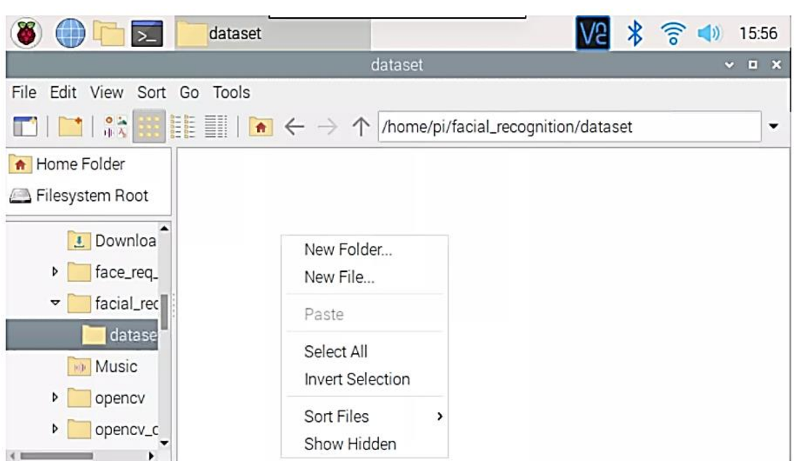


Figure 89: Making new facial recognition folder

5- Put name of person to the folder to be displayed such as Mohamed.

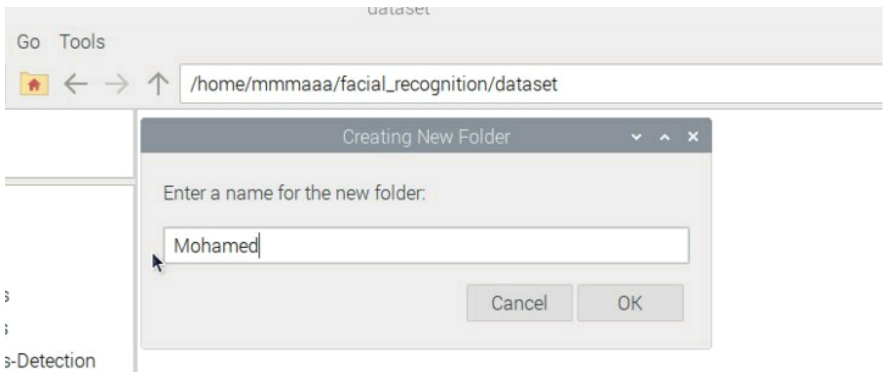


Figure 90: Naming new facial recognition folder

6- Point the webcam at your face and press the spacebar to take a photo of yourself. Each time you press the spacebar you are taking another photo.

Note:

- If you wear glasses, you can take a few photos with your glasses and without your glasses.
- Hats are not recommended for training photos. These photos will be used to train our model.

7- In a new terminal, navigate to facial recognition by typing *cd* facial_recognition.

It takes about 3-4 seconds for the Pi to analyze each photo in your dataset and build the encodings. Pickle file.

8- Run the command to train the model by entering *python train_model.py*.

Now let's test the model we just trained.

9- Run the command to test the model by typing *python facial_req.py*.

In a few seconds, camera window view should open up. Point the camera at your face. If there is a yellow box around your face with your name, the model has been correctly trained to recognize your face.

4.9.3 Visual Output

1- Training phase:

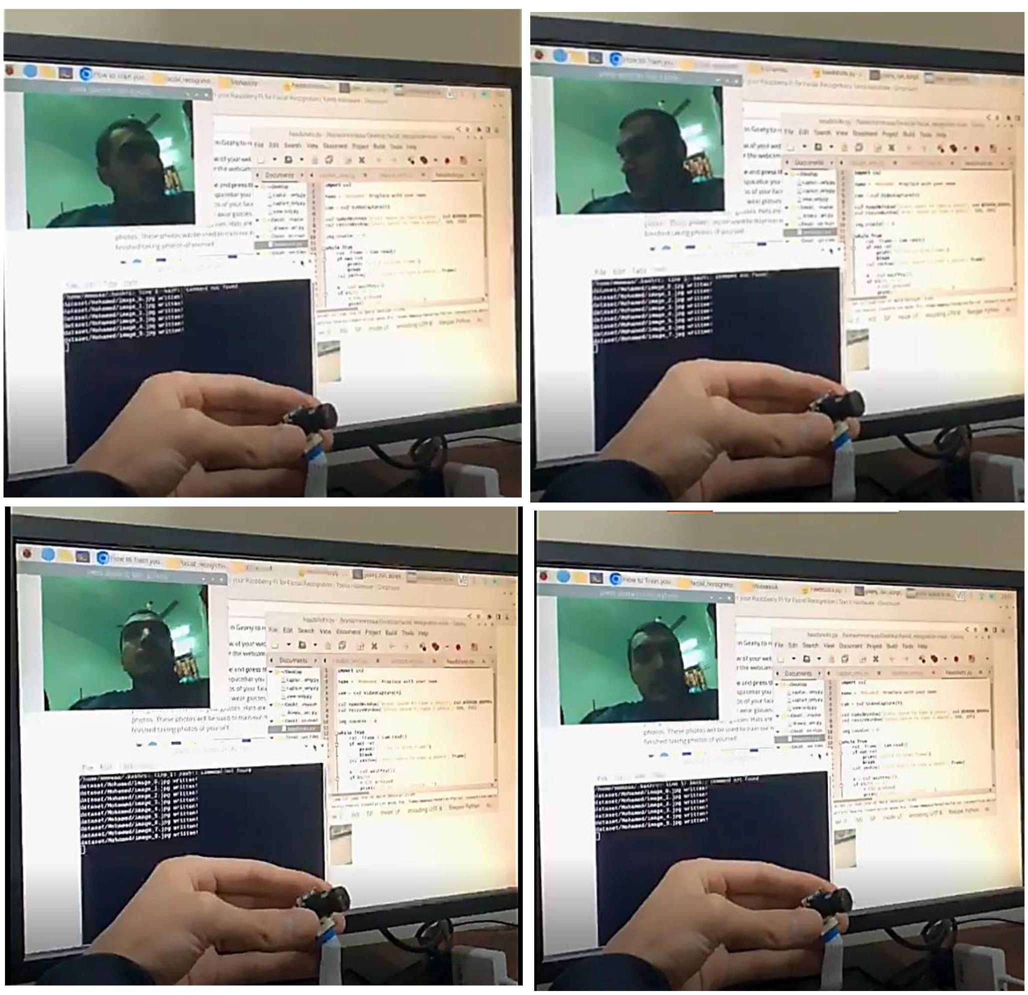


Figure 91: Making training data set for system to better recognize face

Taking pictures of different angles to train model of system to better detect driver in different situations.

Processing procedures through training:

1- Dataset including photos captured within the dataset folder is analyzed through code file called train_model.py and organize these photos into folders by person’s name.

For example, create a new folder named Mohamed and place all photos of Mohamed’s face in the Mohamed folder within the dataset folder.

2- Encodings: train_model.py will create a file named encodings.pickle containing the criteria for identifying faces in the next step.

Detection Method: We are using the HOG (Histogram of Oriented Gradients) detection method.

2- Testing phase:

As shown below, the system recognizes one face - driver’s face - while does not recognize others, until the system train on the other face.

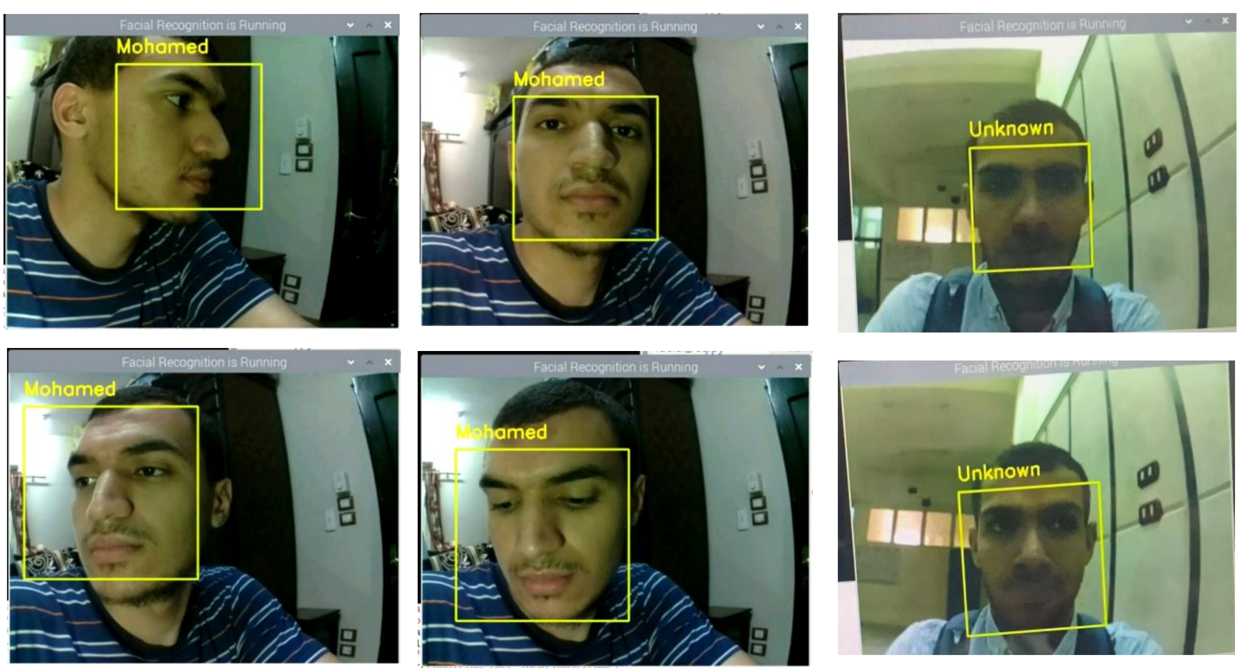


Figure 92: Output of testing phase

- At this phase, system shows yellow label around driver's face showing his name on top of label.
- Label size increase and decreases as driver move back and forth towards camera.
- As facial recognitions process runs it takes few seconds for every image to process as system acts in real time processing, thus no need to capture images and wait to process it separately in many lines of code in traditional manner.
- The system shows yellow label around driver face showing his name on top of label.

4.9.4 Drowsiness detection, Face recognition testing on Windows

Then the Drowsiness output, A text in the program and comes with it an Alarm if the driver dropped for a multi-second, also note the recognition system can recognize your face with one eye but have to unwear the mask if exists, as shown in figure below.

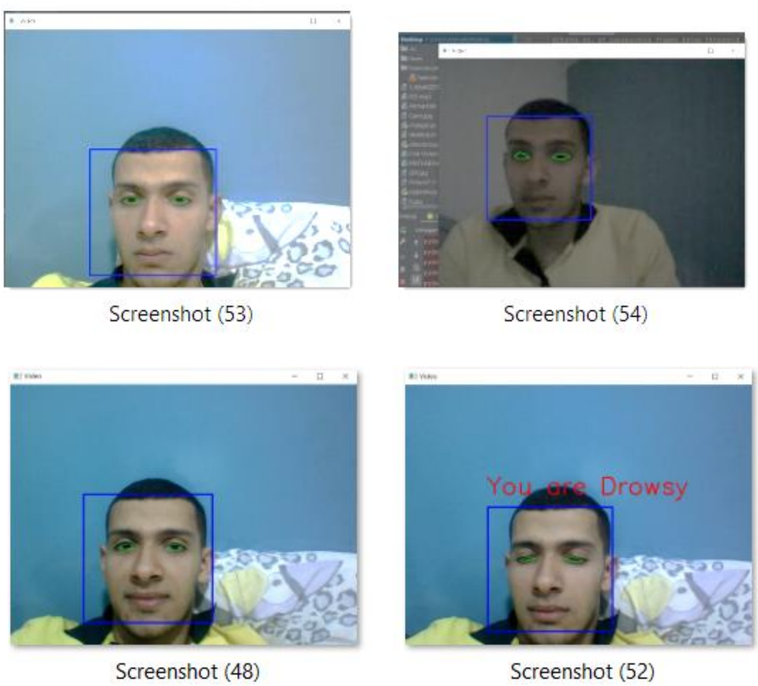


Figure 94: Output of drowsiness detection system

Figure 93: Output of face recognition system

4.9.5 Drowsiness detection testing on Linux

This time with the same configurations, code is trained and ready to be used directly. So, we only need to test it.

We have both outputs on laptop screen and LCD screen with buzzer alert

1- No Detection
   Here it monitor eyes and mouth open or close, if both of eyes are open and mouth is closed, alert is deactivated as shown in figure below.

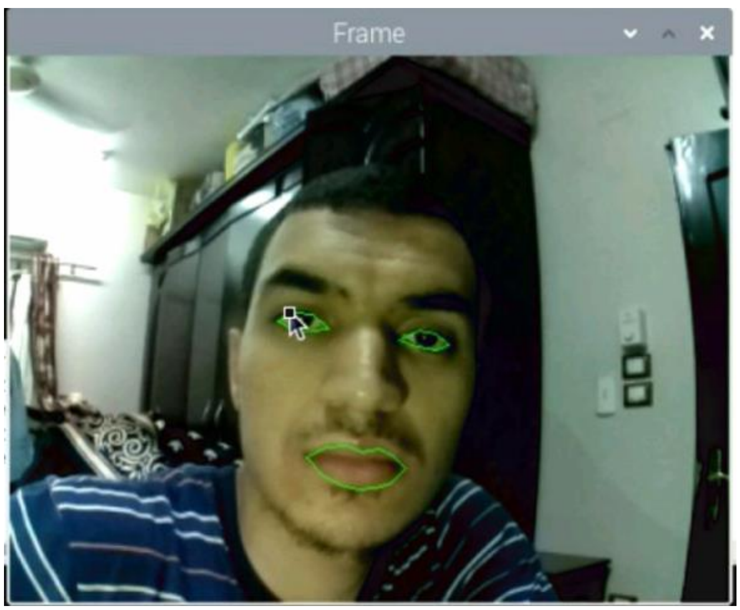


Figure 95: Output on laptop screen

2- Yawn detection

If eyes are open but mouth is open it gives alert on both laptop screen and LCD screen and buzzer alert 2 beeps.

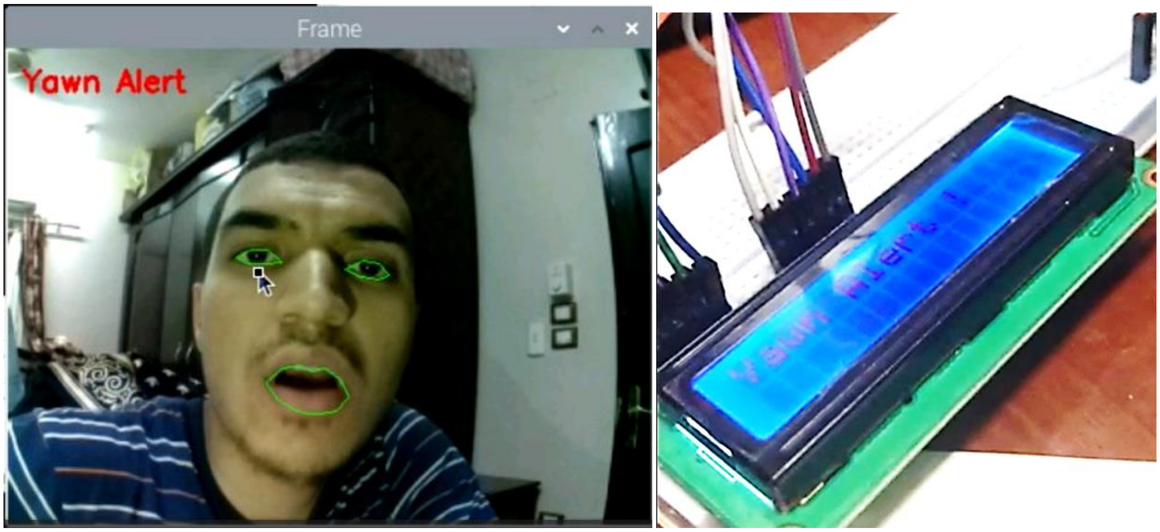


Figure 96: Yawn alert on laptop and LCD screen

3- Drowsiness detection

If eyes and mouth are closed, it waits for about 12 seconds and then gives alert on both laptop screen and LCD screen and buzzer alert 5 beeps.

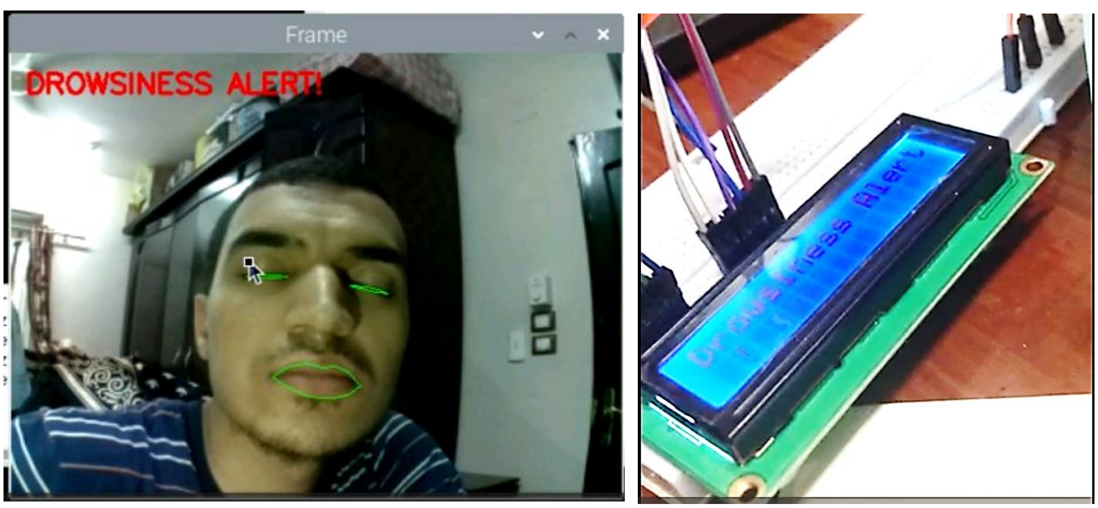


Figure 97: Drowsiness alert on laptop and LCD screen

CH5 Conclusion & Future work

In this graduation project book, we have presented the design and implementation of four advanced driver assistance systems (ADAS) on a smart car. These systems are Adaptive Light Control (ALC), Adaptive Cruise Control (ACC), Forward Collision Warning (FCW) and Drowsiness Detection System (DDS). 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 behavior 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.

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, makes these features not affordable for most cars, however our system is standalone and can be integrated through our integration feature, as well as carrying out the frame design in real, also we prospective improvements in machine learning-based technology that can improve road safety and enhance the effectiveness of Driver Monitoring Systems (DMS). These advances include emotion recognition to detect negative emotions that can impact driving, predictive analytics to prevent potential accidents before they occur, and integrating DMS with autonomous driving technology to monitor the driver's state during manual or semi-autonomous driving. The article highlights the ongoing potential of machine learning technology to prevent accidents caused by human error and enhance road safety.

Appendices

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

Solidworks: to design and create models.

Visual studio code: as an editor.

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

Proteus: PCB Design and Circuit Simulator Software.

Git & GitHub: for version control.

Open On-Chip Debugger (OpenOCD): is a free and open-source software package that provides debugging and programming support for embedded systems.

References

[1] B. Wordenweber, P. Boyce, D. D. Hoffman, and J. Wallaschek,"Automotive lighting and human vision," Springer, 2007, vol. 1.

[2] R. Tamburo, E. Nurvitadhi, A. Chugh, M. Chen, A. Rowe,T. Kanade, and S. G. Narasimhan, "Programmable automotive headlights," in European Conference on Computer Vision, Springer,2014, pp. 750–765.

[3] G. Dhamdhere, S. Chourasia, S. Sasatte, and L. P. Warkey, "Adap- tive front light control system for every vehicle," International Journal of Advanced Research in Electronics and Communication Engineering, pp. 1091–1094.

[4] J. H. Lee, J. Byeon, D. J. Go, and J. R. Park, "Automotive adaptive front lighting requiring only on/off modulation of multi-array leds," Current Optics and Photonics, vol. 1, no. 3, pp. 207–213, 2017.

[5] Raschka, S., & Mirjalili, V. (2017). Python Machine Learning: Machine Learning and Deep Learning with Python, scikit-learn, and TensorFlow. Packt Publishing. (October 2022)

[6] Rosebrock, A. (2018). Facial Recognition Using OpenCV, Python and Deep Learning.PyImageSearch.https://www.pyimagesearch.com/2018/06/18/face-recognition-with-opencv-python-and-deep-learning/ (February 2023)

[7] Heywood, M. (2019). How to Build a Raspberry Pi-based Facial Recognition System. Electronic Design. https://www.electronicdesign.com/embedded-revolution/article/21806133/how-to-build-a-raspberry-pibased-facial-recognition-system (January 2023)

[8] Monk, S. (2020). Raspberry Pi Cookbook: Software and Hardware Problems and Solutions. O'Reilly Media. (February 2023)

[9] Rosebrock, A. (2017). Drowsiness Detection with OpenCV. PyImageSearch. https://www.pyimagesearch.com/2017/05/08/drowsiness-detection-opencv/ (May 2023)