GENESIS-Learning Outcome & Mini-project Summary Report

**Details**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Ver. Rel. No. | Release Date | Prepared By | Reviewed By | To Be Approved | Remarks/Revision Details |
| 1.0 | 16/02/2022 | Pradnya Chaudhari  40021027 |  |  |  |

Contents

[List of Figures 6](#_Toc96280441)

[Mini project – 1: Department Store Management System [Individual] 7](#_Toc96280442)

[Modules: C Programming On Multiple Platforms 7](#_Toc96280443)

[Requirements 8](#_Toc96280444)

[Introduction 8](#_Toc96280445)

[Department Store Management 8](#_Toc96280446)

[Objective:- 8](#_Toc96280447)

[a) Strengths: 8](#_Toc96280448)

[b) Weaknesses 8](#_Toc96280449)

[d) Threats 9](#_Toc96280450)

[High Level Requirements 9](#_Toc96280451)

[Low Level Requirements 10](#_Toc96280452)

[Design 11](#_Toc96280453)

[Test Plan 12](#_Toc96280454)

[Test Case 12](#_Toc96280455)

[Implementation and Summary 12](#_Toc96280456)

[Git Link: 12](#_Toc96280457)

[Git Dashboard 13](#_Toc96280458)

[Summary 13](#_Toc96280459)

[Git link: 13](#_Toc96280460)

[Mini project 2 – Ultrasonic Sound Sensor with Atmega328 Microprocessor [Individual] 14](#_Toc96280461)

[Module: - Essentials of Embedded System 14](#_Toc96280462)

[Topic: - ULTRASONIC SOUND SENSOR WITH ATmega328 MICROPROCESSOR 14](#_Toc96280463)

[Requirements 14](#_Toc96280464)

[Introduction 14](#_Toc96280465)

[Features, Hardware and Software:- 14](#_Toc96280466)

[a) HARDWARE:- 14](#_Toc96280467)

[b) SOFTWARE:- 15](#_Toc96280468)

[SWOT ANALYSIS:- 15](#_Toc96280469)

[d) Strength: 15](#_Toc96280470)

[b) Weakness: 15](#_Toc96280471)

[c) Opportunity: 16](#_Toc96280472)

[d) Threats: 16](#_Toc96280473)

[4W's a 1H:- 16](#_Toc96280474)

[ What: 16](#_Toc96280475)

[ Where: 16](#_Toc96280476)

[ How: 16](#_Toc96280477)

[Low Level Requirements 17](#_Toc96280478)

[Design 18](#_Toc96280479)

[Test Plan 19](#_Toc96280480)

[Obstacle Detection: 19](#_Toc96280481)

[How: Our implementation for this step requires multiple steps: 19](#_Toc96280482)

[Output: As we had steps for each test, we will again make steps for the expected outputs: 20](#_Toc96280483)

[Testing cases 20](#_Toc96280484)

[Summary 21](#_Toc96280485)

[Git Link: 21](#_Toc96280486)

[Mini project 3 – FOOD COURT BILLING SYSTEM 22](#_Toc96280487)

[Modules: - Applied SDLC and Software Testing 22](#_Toc96280488)

[Requirements 22](#_Toc96280489)

[Introduction: 22](#_Toc96280490)

[Features: 22](#_Toc96280491)

[SWOT analysis: 23](#_Toc96280492)

[a) Strength: 23](#_Toc96280493)

[b) Weakness: 23](#_Toc96280494)

[c) Opportunity: 23](#_Toc96280495)

[d) Threat: 23](#_Toc96280496)

[Who: 24](#_Toc96280497)

[What: 24](#_Toc96280498)

[When: 24](#_Toc96280499)

[Where: 24](#_Toc96280500)

[How: 24](#_Toc96280501)

[High Level Requirements 24](#_Toc96280502)

[Low Level Requirements 25](#_Toc96280503)

[Design 26](#_Toc96280504)

[Block diagram: 26](#_Toc96280505)

[Structural diagram: 26](#_Toc96280506)

[High level diagram : 28](#_Toc96280507)

[Low level diagram : 29](#_Toc96280508)

[Test Plan: 29](#_Toc96280509)

[High Level Test Plan 29](#_Toc96280510)

[Low Level Test Plan 31](#_Toc96280511)

[Summary 32](#_Toc96280512)

[Git Link: 32](#_Toc96280513)

[Individual Contribution and Highlights 32](#_Toc96280514)

[Mini project 4 – Calendar Automation [Team] 33](#_Toc96280515)

[Modules:- OOPS with Python 33](#_Toc96280516)

[Requirements 33](#_Toc96280517)

[High Level Requirements 33](#_Toc96280518)

[Low Level Requirements 34](#_Toc96280519)

[Link for template standard input template : 35](#_Toc96280520)

[Requirements for updating Master calendar using Master calendar as input 35](#_Toc96280521)

[Link for template 35](#_Toc96280522)

[App deployment 36](#_Toc96280523)

[Additional features for V1 to do 36](#_Toc96280524)

[Git Link: 36](#_Toc96280525)

[Individual Contribution and Highlights 37](#_Toc96280526)

[Mini project 5 –Team BMW [Team] 38](#_Toc96280527)

[Module: - Applied Model Based Design Module 38](#_Toc96280528)

[Requirements 38](#_Toc96280529)

[Analysis and Physics 39](#_Toc96280530)

[**High Level Requirements:-** 40](#_Toc96280531)

[Low Level Requirements:- 40](#_Toc96280532)

[Running the Simulation in ABS Mode 42](#_Toc96280533)

[Mini project 6 – Wiper Control [Team] 44](#_Toc96280534)

[Modules: - Mastering Microcontrollers with Embedded Driver Development Module 44](#_Toc96280535)

[WIPER CONTROL SYSTEM 44](#_Toc96280536)

[Introduction 44](#_Toc96280537)

[Features 44](#_Toc96280538)

[State of Art 45](#_Toc96280539)

[SWOT Analysis 45](#_Toc96280540)

[Strength 45](#_Toc96280541)

[Weakness 45](#_Toc96280542)

[Opportunities 45](#_Toc96280543)

[Threats 46](#_Toc96280544)

[4W 1H: 46](#_Toc96280545)

[Who: 46](#_Toc96280546)

[What: 46](#_Toc96280547)

[Where: 46](#_Toc96280548)

[When: 46](#_Toc96280549)

[How: 47](#_Toc96280550)

[BLOCK DIAGRAM 47](#_Toc96280551)

[FLOW CHART 48](#_Toc96280552)

[CASE STUDY 49](#_Toc96280553)

[STATE FLOW 50](#_Toc96280554)

[SYSTEM DESIGNS 51](#_Toc96280555)

[High Level Requirements 51](#_Toc96280556)

[Low Level Requirements 52](#_Toc96280557)

[Implementation and Summary 53](#_Toc96280558)

[Git Link: 53](#_Toc96280559)

[Individual Contribution and Highlights 53](#_Toc96280560)

[Mini project 7 – Ford Project [Team] 54](#_Toc96280561)

[Modules: - Automotive Systems 54](#_Toc96280562)

[Requirements 54](#_Toc96280563)

[Ford Aspire 54](#_Toc96280564)

[Body Control Module 54](#_Toc96280565)

[Features: 54](#_Toc96280566)

[Introduction 55](#_Toc96280567)

[4W's 1H 55](#_Toc96280568)

[WHAT: 55](#_Toc96280569)

[WHERE: 55](#_Toc96280570)

[WHEN: 55](#_Toc96280571)

[WHY: 56](#_Toc96280572)

[How:- 56](#_Toc96280573)

[High Level Requirement:- 56](#_Toc96280574)

[Design 57](#_Toc96280575)

[Body Control Module 57](#_Toc96280576)

[Interior Light Control 58](#_Toc96280577)

[Implementation and Summary 58](#_Toc96280578)

[Git Link: 58](#_Toc96280579)

[Individual Contribution and Highlights 58](#_Toc96280580)

[Mini project 8 – EV Truck [Team] 59](#_Toc96280581)

[Module: - Applied Control Systems and Vehicle Dynamics 59](#_Toc96280582)

[EV TRUCK 59](#_Toc96280583)

[ARE ELECTRIC TRUCKS BETTER? 59](#_Toc96280584)

[BENEFITS OF ELECTRIC TRUCKS 60](#_Toc96280585)

[Argument for Electric Trucks 60](#_Toc96280586)

[Argument against Electric Truck’s 60](#_Toc96280587)

[Research 61](#_Toc96280588)

[Reference: 61](#_Toc96280589)

[Simulation Design: 61](#_Toc96280590)

[Analysis 62](#_Toc96280591)

[TWO MODEL COMPARISON 62](#_Toc96280592)

[EV MODEL DESIGN 64](#_Toc96280593)

[Conclusion 64](#_Toc96280594)

[Implementation 65](#_Toc96280595)

[Git Link: 65](#_Toc96280596)

[Mini project 9 – Parking System [Individual] 66](#_Toc96280597)

[Module:- Classic Autosar Basic to Intermediate 66](#_Toc96280598)

[Requirements 66](#_Toc96280599)

[**High Level Requirement:** 66](#_Toc96280600)

[**Low Level Requirement:** 66](#_Toc96280601)

[Design 67](#_Toc96280602)

[Implementation and Summary 67](#_Toc96280603)

[Git Link: 67](#_Toc96280604)

## 

## List of Figures

[Figure 1 Behavior Diagram 8](#_Toc95933213)

[Figure 2 Structure Diagram 9](#_Toc95933214)

[Figure 3 Git Dashboard 11](#_Toc95933215)

[Figure 4 Git Inspector Summary 11](#_Toc95933216)

[Figure 5 Behavior Diagram 13](#_Toc95933217)

[Figure 6 Structure Diagram 14](#_Toc95933218)

[Figure 7 Block Diagram 14](#_Toc95933219)

[Figure 8 Simulation 15](#_Toc95933220)

[Figure 9 Git Dashboard 16](#_Toc95933221)

[Figure 10 Behavior Diagram 19](#_Toc95933222)

[Figure 11 UserFlow Diagram 19](#_Toc95933223)

[Figure 12 Structure Diagram 20](#_Toc95933224)

[Figure 13 Git Dashboard 25](#_Toc95933225)

[Figure 14 Git Inspector Summary 25](#_Toc95933226)

[Figure 15 Structure Diagram 29](#_Toc95933227)

[Figure 16 Behavior Diagram 30](#_Toc95933228)

[Figure 17 Structure Diagram 34](#_Toc95933229)

[Figure 18 VFB Diagram 37](#_Toc95933230)

# Mini project – 1: Department Store Management System [Individual]

## Modules: C Programming On Multiple Platforms

### Requirements

## Introduction

## Department Store Management

The project as the name suggests is based on the departmental store management system. Departmental store consist commodities of daily human use, so they need to be managed in various ways. The program mainly focuses on manipulation of various functions by a cashier and manager on the basis of goods selected by the costumers.

# Objective:-

The main objectives of this project are listed as follows:

* Proper management of selling items in departmental store.
* Can be useful for both manager and cashier.
* Good security of information for which password facilities are provided.
* Maximum providence of modification and search options.
* User friendly messages in the way for easy application of program.

**SWOT Analysis**

## Strengths:

Strengths are things that your system does particularly well, or in a way that distinguishes you from your competitors

## Weaknesses

Weaknesses, like strengths, are inherent features of your projects, so focus on your people, resources, systems, and procedures.

1. Opportunity

They usually arise from situations outside your organization, and require an eye to what might happen in the future.

## Threats

Threats include anything that can negatively affect your business from the outside, such as supply-chain problems, shifts in market requirements, or a shortage of recruits.

**4W's and 1 H's**

**Why:**

I am developing this basic management system to perform basic operation in best easy manner and improve my coding skills.

**Where:**

It has to be used easily by the users.

**What:**

The main components of the program are to start as a manager or cashier and verification, goods entry and display, bill display, total sales display, etc.

**When:**

It has to be deployed 25th of November 2021.

**How:**

I am using C programming language for Developing this simple basic store management system.

### High Level Requirements

| **ID** | **Description** |
| --- | --- |
| HLR1 | This Department Store Management System is primarily based totally on an idea to offer facts on calculating, adding, viewing goods, and different capabilities too... |
| HLR2 | Provides the searching facilities based on various factors. Such as Products, Payments, Discounts, Stock. |
| HLR3 | Department Store also manage the Inventory details online for Discounts details, Stock details, Products. |

### Low Level Requirements

| **ID** | **Description** |
| --- | --- |
| LLR\_1 | • Create the master and slave database structure to reduce the overload of the database queries |
| LLR\_2 | • We will host the platform on online servers to make it accessible worldwide. |
| LLR\_3 | • Implement the backup mechanism for taking backup of codebase and database on regular basis on different servers. |
| LLR\_4 | • If I consider commercial field it will provide me the different Result. |

## Design

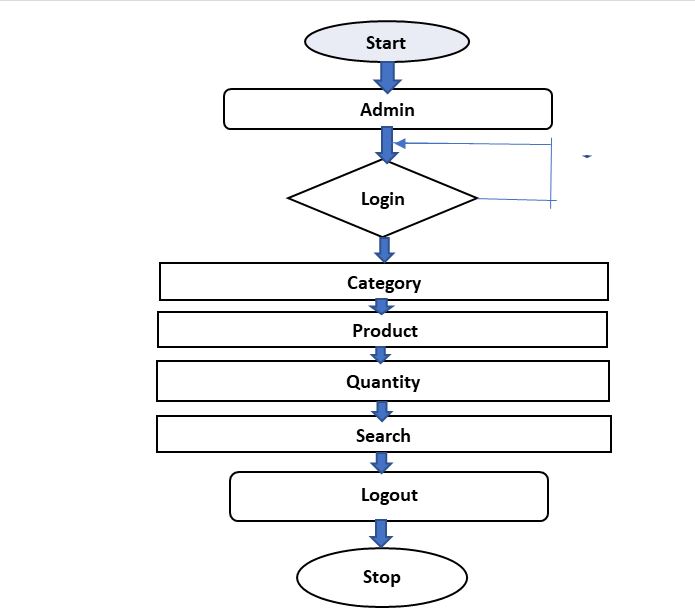


Figure 1 Behaviour Diagram

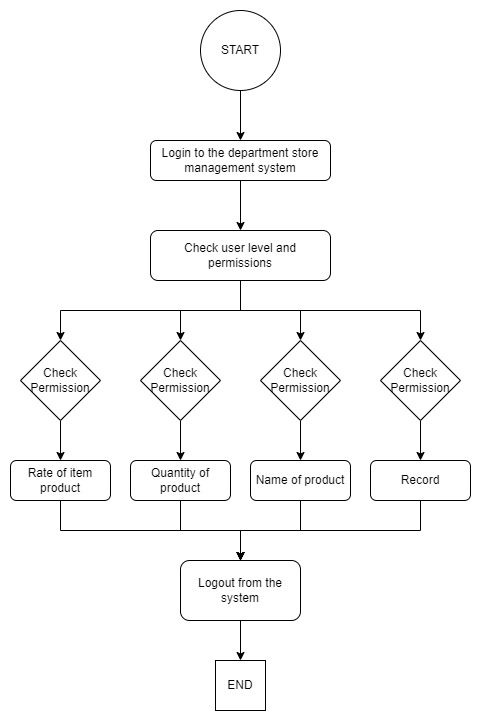


Figure 2 Flow Chart

## Test Plan

| **Test ID** | **Description** | **Exp I/P** | **Exp O/P** | **Actual Out** | **Type Of Test** |
| --- | --- | --- | --- | --- | --- |
| H\_01 | Enter a valid username & password | Username: user Password: pass | login Successfully | login Successfully | Requirement Based |
| H\_02 | Enter invalid username & password | Username: user Password: pass | Login Failed Enter Again Username & Password | Login Failed Enter Again Username & Password | Requirement Based |
| H\_03 | User should be able to Login with valid credentials | Username: user Password: pass | Login Failed Enter Again Username & Password | Login Successfully | Scenario Based |

## Test Case

## Implementation and Summary

### Git Link:

Link: <https://github.com/alrichroshan/M1_App_FinancialCalculator>

Git Dashboard ****

Figure 3 Git Dashboard

### Summary

 We can conclude that the project has been successful for what it aims to perform. This program under the project can be used in a departmental store for the proper management of purchase and sales of the goods along with the billing management. Hence although the project got some of its limitations the maximum effort was targeted for the gain of its aim i.e. departmental store management system.

Git link:

https://github.com/Pradnya579/M1\_ApplicationLab\_DepartmentStoreManagementSystem

**CERTIFICATION DONE IN MODULE**

* SOLO-Learn Certification
* Linux Certification
* Github Learning Certification

# Mini project 2 – Ultrasonic Sound Sensor with Atmega328 Microprocessor [Individual]

## Module: - Essentials of Embedded System

## Topic: - ULTRASONIC SOUND SENSOR WITH ATmega328 MICROPROCESSOR

### Requirements

## Introduction

The project as the name suggests is based on Ultrasonic sensors. Ultrasonic sensors work by sending out a sound wave at a frequency above the range of human hearing. Our ultrasonic sensors, like many others, use a single transducer to send a pulse and to receive the echo. The sensor determines the distance to a target by measuring time lapses between the sending and receiving of the ultrasonic pulse.

# Features, Hardware and Software:-

## a) HARDWARE:-

#### 1] SimulIDE:

- SimulIDE provides AVR, Arduino and PIC microcontrollers that can be accessed just like other components.

- Features like gypsum and simavr allow you to use PIC and AVR microcontrollers, respectively.

#### 2] AVR:

- An automatic voltage regulator (AVR) is an electronic device that maintains a constant voltage level to electrical equipment on the same load.

- The AVR regulates voltage variations to deliver constant, reliable power supply.

## b) SOFTWARE:-

#### 1] ATmega328:

- ATmega328 is commonly used in many projects and autonomous systems where a simple, low-powered, low-cost micro-controller is needed

- Perhaps the most common implementation of this chip is on the popular Arduino development platform.

#### 2] Sound:

- A sound sensor is defined as a module that detects sound waves through its intensity and converting it to electrical signals.

#### 3] Display:

- A display device is an output device for presentation of information in visual or tactile form.

# SWOT ANALYSIS:-

## d) Strength:

The distance to an obstacle can be measured with the low cost ultrasonic sensor. The sensors can measure distances from 2 to 400cm with an accuracy of 3mm. This sensors module includes ultrasonic transmitter, ultrasonic receiver and control circuit.

## b) Weakness:

Although we fully believe in the capability of our sensors, we understand that ultrasonic are not suited for every application. Focuses of low thickness, similar to froth and fabric, have a tendency to assimilate sound vitality; these materials may be hard to sense at long range.

## c) Opportunity:

This project can be used as parking assistance systems in vehicles with high power ultrasonic transmitter. This Project Can be used as burglar alarm with suitable additional software for homes and offices.

## d) Threats:

Ultrasonic sensors must view a surface (particularly a hard, level surface) unequivocally (oppositely) to get adequate sound reverberation. Additionally, solid detecting requires a base target surface range, which is indicated for every sensor sort. If connection is wrong there might be chances of short-circuit.

# 4W's a 1H:-

# What:

We have made a setup based on a microcontroller in which real time distance is sensed by an ultrasonic sensor and displays measured distance on an LCD display.

# Where:

It measures accurate distance using a non-contact technology - A technology that involves no physical contact between sensor and object.

-3 When: In 1959, Satomura created an ultrasonic flow meter that used Doppler technology.

-# Why: I am Developing this project for easily measure the distance between objects

# How:

By using Atmega328 and display an ultrasonic sensor mainly used to determine the distance of the target object.

**High Level Requirements**

| **ID** | **Description** |
| --- | --- |
| HLR1 | Used to avoid and detect obstacles with robots like biped robot, obstacle avoider robot, path finding robot etc. |
| HLR2 | Used to measure the distance within a wide range of 2cm to 400cm |
| HLR3 | Depth of certain places like wells, pits etc can be measured since the waves can penetrate through water |

### Low Level Requirements

| **ID** | **Description** |
| --- | --- |
| LLR\_1 | • Power Supply: +5V DC. |
| LLR\_2 | • Measuring Angle: 30 degree. |
| LLR\_3 | • Trigger Input Pulse width: 10uS TTL pulse. |
| LLR\_4 | • Depth of certain places like wells, pits etc can be measured since the waves can penetrate through water. |

## Design

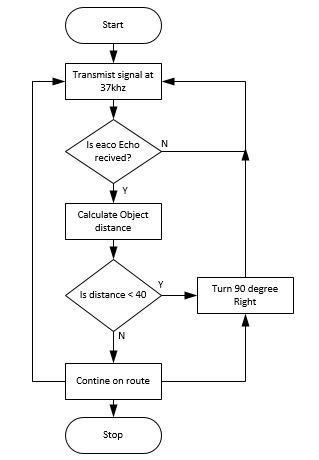


Figure 5 Behaviour Diagram

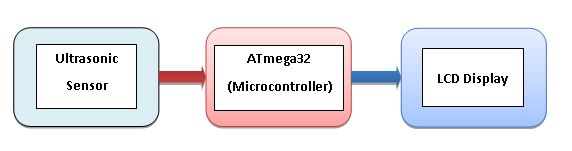
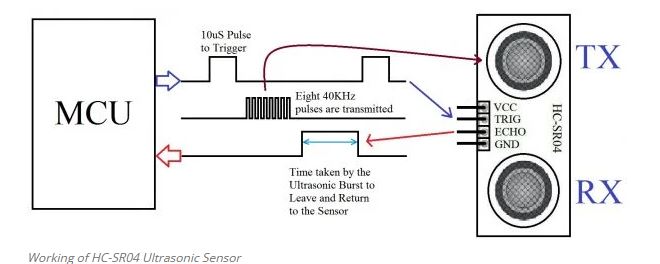


Figure 6 Block Diagram



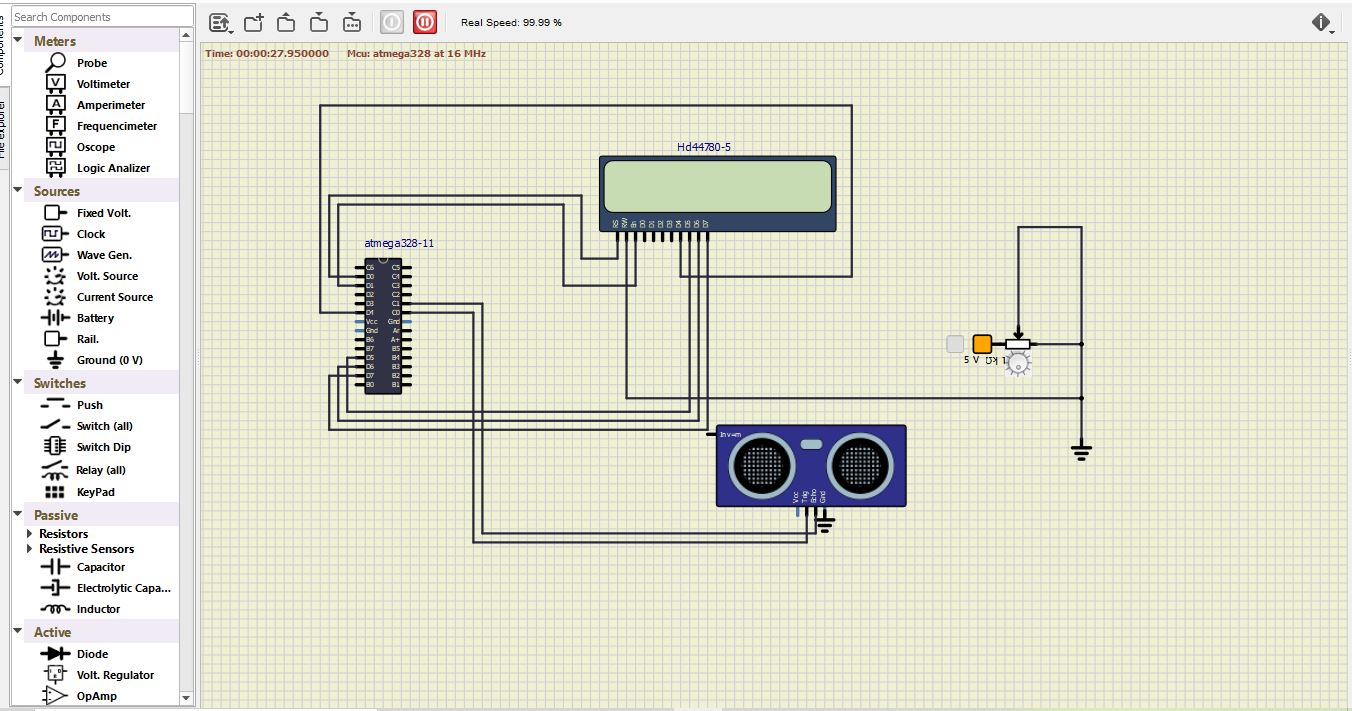
Figure 7 Structural Diagram

Figure 8 Simulation

## Test Plan

## Obstacle Detection:

### How: Our implementation for this step requires multiple steps:

#### Step 1: Find a distance value between each pair of sensors. To test the distance

Value, we may use the numbers we see for the height and length, as well as the Pythagorean Theorem. ####Step 2: Check the angle found between each pair of sensors using the distance value initially found. ####Step 3: Using these values, determine what each angle should approximately be to detect different types of obstacles. ####Step 4: Detect the obstacles.

## Output: As we had steps for each test, we will again make steps for the expected outputs:

#### Step 1: Compare the outputted (through serial) value for the hypotenuse to the

Pythagorean calculated value. We expect them to be the same.

#### Step 2: Using the same technique as step 1 except calculating the angle, we should

See the same value for this calculation as well.

#### Step 3: The values and outputs for the “obstacle detected” will be constantly

Checked and rechecked to make sure the angles determine the correct obstacle.

#### Step4: Adding Audio to the Ultrasonic Sensors.

## Testing cases

| **Average Speed(m/s)** | **0.8** | **1.5** | **2.0** |
| --- | --- | --- | --- |
| Mean RMS error (cm)\* | 19.4 | 12.7 | 10.2 |
| SD\*\* | 11.2 | 14.3 | 13.4 |
| Sensing error (%) | 5.0 | 1.6 | 1.0 |
| ---------------------------------------------------------- |  |  |  |

#### RMS error: Root mean square error between actual and sensing distance.

#### SD\*\*: Standard deviation of the RMS errors.

## Summary

The objective of the project was to design and implement an ultrasonic distance meter. The device described here can detect the target and calculate the distance of the target. The ultrasonic distance meter is a low cost, low a simple device for distance measurement. The device calculates the distance with suitable accuracy and resolution. It is a handy system for non-contact measurement of distance. The device has its application in many fields. It can be used in car backing system, automation and robotics, detecting the depth of the snow, water level of the tank, production line. This device will also have its application in civil and mechanical field for precise and small measurements. For calculating the distance using this device, the target whose distance is to be measured should always be perpendicular to the plane of propagation of the ultrasonic waves. Hence the orientation of the target is a limitation of this system. The ultrasonic detection range also depends on the size and position of the target. The bigger is the target, stronger will be the reflected signal and more accurate will be the distance calculated. Hence the ultrasonic distance meter is an extremely useful device.

### Git Link:

Link: <https://github.com/Pradnya579/M2-Embedded_UltrasonicSoundSensor>

**Git Dashboard**

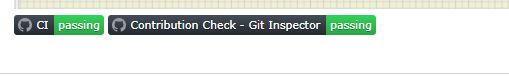


Figure 9 Git Dashboard

# Mini project 3 – FOOD COURT BILLING SYSTEM

## Modules: - Applied SDLC and Software Testing

## Requirements

## Introduction:

For simplicity and better understanding of the owner, this software is designed. It would avoid confusion and help operate the software easily. Also, such a software that is easy to use will reduce the work of owner who still maintain all the logs in registers and files. It would be of great benefit as all calculations would be done easily on the click of a button. This reduces the burden on the owner as the paperwork or calculation work is reduce and other essentials to update.

## Features:

* For the calculate bills, the user can view their bills after ordering a foods.
* For the add orders, the user can add new order of foods.
* For the edit orders, the user can edit their orders information.
* For the display orders, the user can view their orders.
* For the search orders, the user can search their orders.
* For the delete orders, the user can delete their order information.
* For the exit, the user can also exit in the system.

## SWOT analysis:



### a) Strength:

This system is a keeping track of billing records, menus and extra food items.

### b) Weakness:

All the staff needs to be trained on the software. If there is a power failure, the hotel runs a high risk of losing all the stored information.

### c) Opportunity:

This project can be merged with any major projects in future where meals and their monthly calculations need to be done.

### d) Threat:

If there is a virus attack the stored information might get corrupt.

**4W's and 1 H's**

### Who:

It can be used by the owner of the food court to update and to use it freely.

### What:

A user friendly application for used to check update in food court daily.

### When:

As the customers in their recess time use food court inside the company for their food consumption they will need a management system to check today's update.

### Where:

Used in all mess canter’s running inside a company for owner's benefit.

### How:

It can be used in a mobile app easily or can login in a PC.

### High Level Requirements

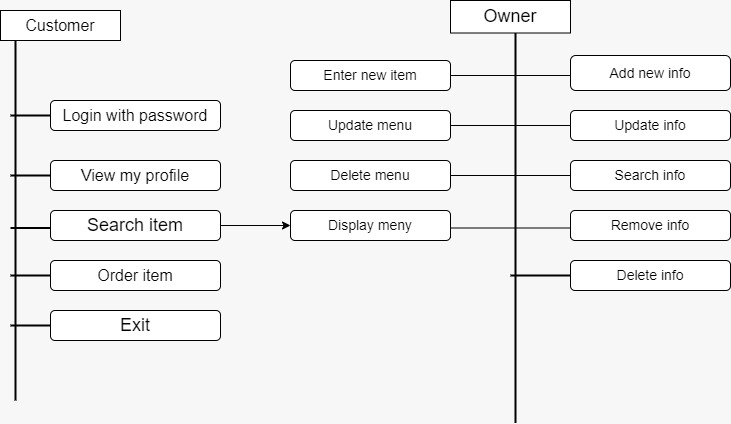
| **ID** | **Description** |
| --- | --- |
| HLR1 | Customer should be able to add item via item. |
| HLR2 | Customer should be able to search items from menu function. |
| HLR3 | Customer shoulde be able to see their order on display function. |
| HLR4 | Customer should able to edit their orders. |
| HLR5 | Customer should able to search item via name or item code. |
| HLR6 | Application should able to do the all calculation that are required to generate bill amount. |
| HLR7 | Customer should br able to delete the perticular item from ordered list. |

### Low Level Requirements

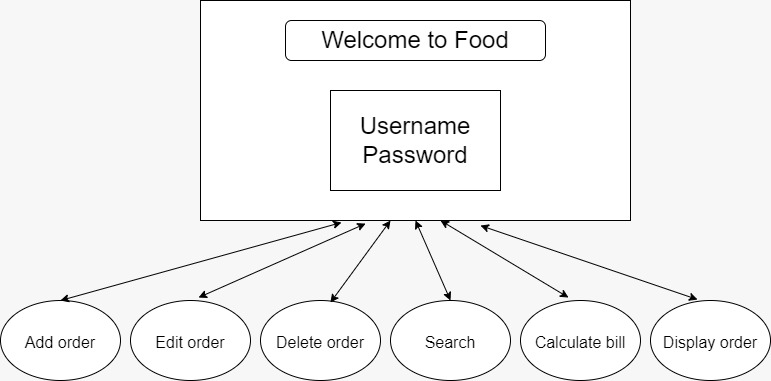
| **ID** | **Description** |
| --- | --- |
| LLR1 | Login Page off Food Court. |
| LLR2 | Enter user and password. |
| LLR3 | Newly added details should be display. |
| LLR4 | Item name, quantity, rate should be removed. |
| LLR5 | Item name, item number and item rate should be there while generating bill. |
| LLR6 | Application should return exact final bill. |

## Design

### Block diagram:

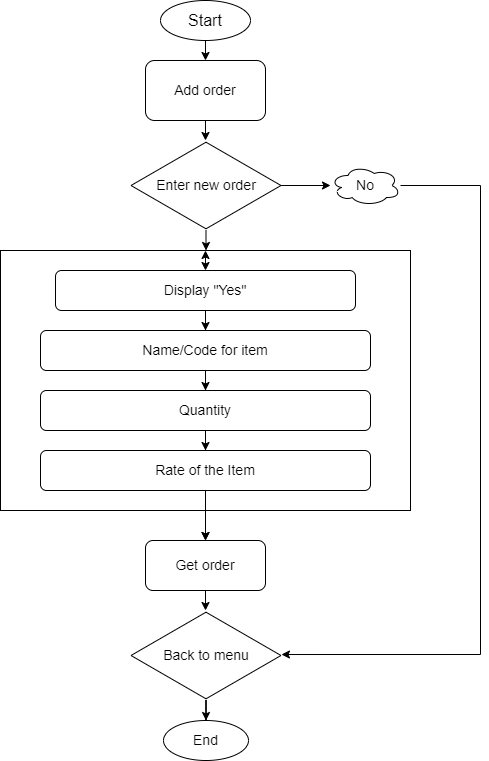
[](https://user-images.githubusercontent.com/94294050/146140854-be886e97-d2c4-4e23-bf37-5ebac4da199f.jpg)

### Structural diagram:

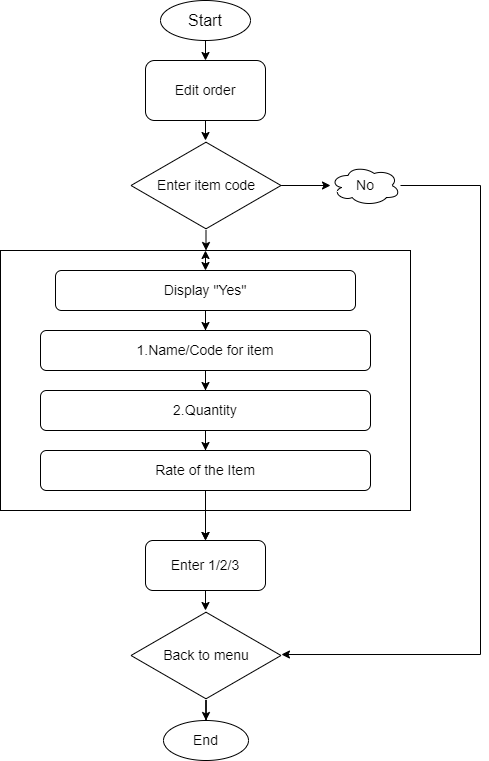
[](https://user-images.githubusercontent.com/94294050/146142432-b9beb024-7a16-4018-9083-dba53918795e.jpg)

Behavioural diagram:

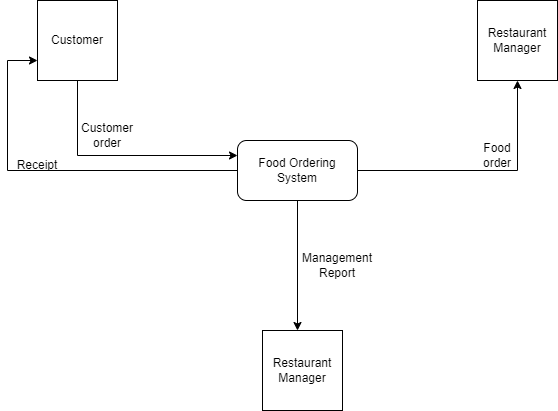
#### Flow chart: 1

[](https://user-images.githubusercontent.com/94240954/146131150-ec18ce18-bc84-4945-b776-ce4426ef4caf.png)

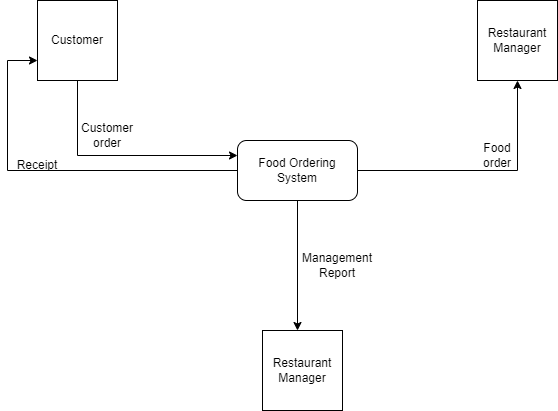
#### Flow chart: 2

[](https://user-images.githubusercontent.com/94240954/146131323-04fa51e2-7b9e-4a07-9a4d-68d1f4c351bd.png)

## High level diagram :

[](https://user-images.githubusercontent.com/94240954/146324220-25ffd32f-6f23-4b31-9e23-277ba4fb061a.png)

## Low level diagram :

[](https://user-images.githubusercontent.com/94240954/146324237-c6903ae2-586b-4592-97f3-9ea2249efbdf.png)

# Test Plan:

### High Level Test Plan

| **TEST\_ID** | **Description** | **Expected I/P** | **Expected O/P** | **Actual O/P** | **Type Of Test** |
| --- | --- | --- | --- | --- | --- |
| HLR\_1 | Login to system | Provide proper Username with character length of 10 | entered username | entered username | Requirement Based |
| HLR\_2 | Login to system | Provide proper password with character length of 10 | Login successful | Login successful | Requirement Based |
| HLR\_3 | Providing items that you wants to add | User Choice | Added Successfully | Added Successfully | Requirement Based |
| HLR\_4 | Display the menu | ---- | Added Items is Displayed | Added Items is Displayed | Requirement Based |
| HLR\_5 | Bill Calculation | Choice | Customer's Bill | Customer"s Bill | Requirement Based |

### Low Level Test Plan

| **TEST\_ID** | **Description** | **Expected I/P** | **Expected O/P** | **Actual O/P** | **Type Of Test** |
| --- | --- | --- | --- | --- | --- |
| LLR\_1 | Login to system | username and password is Incorrect (in case numbers or exceed length) | Login is Unsuccessful | Login is Unsuccessful | Requirement Based |
| LLR\_2 | Edit Item | Provide Item code | Item Edited | Item Edited | Requirement Based |
| LLR\_3 | Delete Item | Provide Item | Item Deleted | Item Deleted | Requirement Based |
| LLR\_4 | Search Item | Provide Item Code | Item Searched | Item Searched | Requirement Based |
| LLR\_5 | exit operation | ---- | Exit Successfully | Exit Successfully | Requirement Based |

## Summary

* Add orders
* Edit orders
* Display orders
* Search orders
* Delete orders
* Calculate bill

### Git Link:

Link: <https://github.com/GENESIS2021Q1/Applied_SDLC-Dec_Team_47>

### Individual Contribution and Highlights

* Requirements
* Implementation (Add Order in System)
* Created Unity File

# Mini project 4 – Calendar Automation [Team]

## Modules:- OOPS with Python

## Requirements

## High Level Requirements

| **ID** | **Feature** | **MATLAB v0 Status** | **Python v0 Status** |
| --- | --- | --- | --- |
| HR01 | GUI | Implemented | Implemented |
| HR02 | Master calendar | Implemented | Implemented |
| HR03 | Faculty calendar | Implemented | Implemented |
| HR04 | Faculty load sheet | Implemented | Implemented |
| HR05 | Showing Available Open Slots based on faculty and modules | Not Available | Not Available |
| HR06 | Output file generated across different computers (windows + Linux) | Not Available | Implemented |
| HR07 | Visualizing data to create Meaningful Insights | Not Available | Not Available |
| HR08 | Calculate Individual Faculty Load | Implemented | Implemented |

## Low Level Requirements

| **ID** | **Feature** | **High Level ID** | **MATLAB v0 Status** | **Python v0 Status** |
| --- | --- | --- | --- | --- |
| LR01 | GUI should allow user to login using credentials | HR01 | Not Available | Not Available |
| LR02 | Input Files Based on Different Initiatives and Timelines | HR01 | Implemented | Not Available |
| LR03 | GUI should get Base Calendar as Input | HR01 | Implemented | Implemented |
| LR04 | GUI should get Month and Initiative as Input | HR01 | Implemented | Implemented |
| LR05 | GUI should be able to show Conflicts/Warnings | HR01 | Implemented | Not Implemented |
| LR06 | Master Calendar: display Month wise | HR02 | Implemented | Implemented |
| LR07 | Master Calendar: display Initiative wise | HR02 | Implemented | Not Available |
| LR08 | Master Calendar: Differentiate Initiatives (Color Codes/Numbers) | HR02 | Implemented | Implemented |
| LR09 | Master Calendar: Appending | HR02 | Implemented | Not Available |
| LR10 | Master Calendar: Course code correction | HR02 | Implemented | Not Available |

# Link for template standard input template :

<https://docs.google.com/spreadsheets/d/1EWYp_1iyK2wLMfKGJOiTJAk5WexZusCP/edit?usp=sharing&ouid=113003694561146884677&rtpof=true&sd=true>

* Using the template above, training schedule can be added monthwise and initiatives wise
* The name of the input excel sheet MUST be named as "Test\_vector"(as shown in template)
* Along with the Test\_vector sheet, "Key" sheet MUST be present under the columns assigned as in the template
* The "Key" sheet must contain all times the 6 fixed initiatives with their respective codes and total list of course code and course title in order to refer for corrections while writing to output files
* Appending additional slots for existing courses is possible by adding just the additional slots in the input file for the same course

## Requirements for updating Master calendar using Master calendar as input

# Link for template

2 Slots format - M/A : <https://docs.google.com/spreadsheets/d/1jtKnXV12VE1fH20CGDo4B3uNWRTAhQCWz-hHUDWUe3I/edit?usp=sharing>

4 Slots format - M1/M2/A1/A2 : <https://docs.google.com/spreadsheets/d/1jVheSPZkOtfNKRNoc_858nwk2UaHCe0gExTNZfZ8vxA/edit?usp=sharing>

* Any of the two templates can be used for updating Master calendar month wise on to the drive
* The blocked slots must have the corresponding initiative code in the cell according to the key as shown in the sample data in the template
* The name of the sheet must be the name of the month to be updated
* The "Key" sheet must be present with the fixed list of initiatives and initiative code

## App deployment

* The app is deployed on heroku servers.
* To add/modify new features, you will be required to install HEROKU CLI [link](https://devcenter.heroku.com/articles/getting-started-with-python#set-up)
* After installation, open terminal in working directory and enter the following commands:
  + "heroku git:clone -a gea calendar"
  + login using heroku credentials
* After pulling and making changes, enter the following commands to push app and deploy on server
  + Git add.
  + git commit -m "commit message"
  + git push heroku master

### Additional features for V1 to do

* Update key sheet by appending new initiatives/courses list
* Check for duplicate course entries in input file
* Using built in libraries to identify number of days in month, current year and highlight weekend and holidays
* Function to remove a course schedule
* Read multiple months data in one sheet as input file (currently takes data one by one month)
* Calculate individual faculty load

### Git Link:

https://github.com/Pradnya579/GENESIS2021-OOP-Python\_Team\_46

### Individual Contribution and Highlights

1. Improved implementation of Python Programming
2. Source code management using Git Hub

Role in Project Team

1. Programmer: Done Programming for calendar Automation

# Mini project 5 –Team BMW [Team]

## Module: - Applied Model Based Design Module

**Individual Topic: - Anti-Lock Braking System**

## Requirements

**INTRODUCTION**

Anti-lock brake systems (ABS) prevent brakes from locking during braking. Under normal braking conditions the driver controls the brakes. However, during severe braking or on slippery roadways, when the driver causes the wheels to approach lockup, the antilock system takes over. ABS modulates the brake line pressure independent of the pedal force, to bring the wheel speed back to the slip level range that is necessary for optimal braking performance. An antilock system consists of wheel speed sensors, a hydraulic modulator, and an electronic control unit. The ABS has a feedback control system that modulates the brake pressure in response to wheel deceleration and wheel angular velocity to prevent the controlled wheel from locking. The system shuts down when the vehicle speed is below a pre-set threshold.

**OVERVIEW**

* ABS was first invented and applied in the aircraft industry and then was introduced to automobile industry in the early 1970's. However, it had not been used popularly until the middle of the 1980's due to technical difficulties and high cost.
* ABS functions in place of the traditional brake system at times of wheel lock-up. A quick test sequence checks all the components of the system.
* If ever the test sequence fails, the normal brake system is in control.
* Although the normal brake system can give instant and efficient braking, it can cause the wheels to be lock up, therefore, the driver cannot steer and would lose control of the car.
* If any of the wheels happen to be skidding, the driver must recognize wheel-skid and manually 'pump the brakes' to avoid a skid. The advantage of ABS lies in its ability to allow the driver retain steering control in order to keep the car moving in the direction that the wheels are turned towards, rather than skidding in the direction of the car's forward momentum.
* ABS has the classic design of an embedded system.

1. controller Sensor
2. Wheel speed.
3. Actuators (valve and ABS reservoir) at each wheel.

### Analysis and Physics

The wheel rotates with an initial angular speed that corresponds to the vehicle speed before the brakes are applied. We used separate integrators to compute wheel angular speed and vehicle speed. We use two speeds to calculate slip, which is determined by Equation 1. Note that we introduce vehicle speed expressed as an angular velocity (see below).

$$\omega_v = \frac{V}{R} \mbox{ (equals the wheel angular speed if there is no slip)}$$

**Equation 1**

$$ \omega_v = \frac{V_v}{R_r}$$

$$slip=1-\frac{\omega_w}{\omega_v}$$

$$\omega_v = \mbox{ vehicle speed divided by wheel radius}$$

$$ V_v = \mbox{ vehicle linear velocity}$$

$$ R_r = \mbox{ wheel radius}$$

$$ \omega_w = \mbox{ wheel angular velocity}$$

From these expressions, we see that slip is zero when wheel speed and vehicle speed are equal, and slip equals one when the wheel is locked. A desirable slip value is 0.2, which means that the number of wheel revolutions equals 0.8 times the number of revolutions under non-braking conditions with the same vehicle velocity. This maximizes the adhesion between the tire and road and minimizes the stopping distance with the available friction.

**REQUIREMENTS**

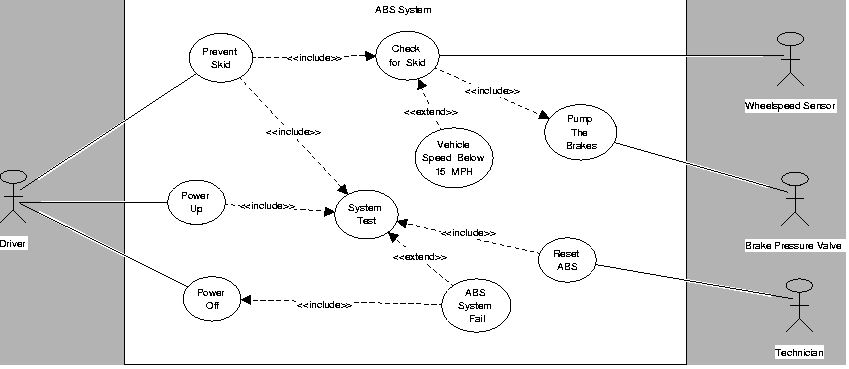
**High Level Requirements:-**

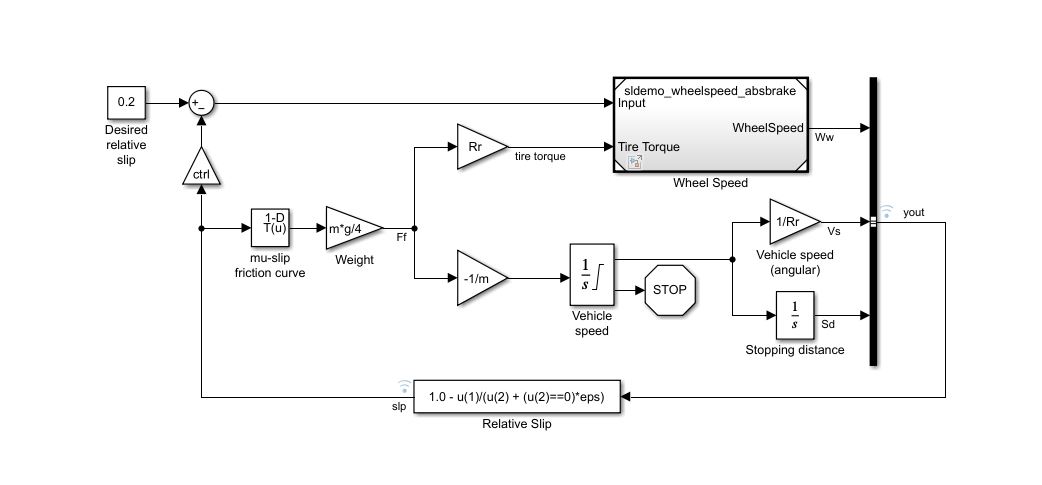
|  |  |
| --- | --- |
| ID | Description |
| HLR1 | Receive on/off signals from the brakes, and use them for engaging and disengaging the ABS system |
| HLR2 | Receive rotational speed data from four wheel speed sensors |
| HLR3 | The same signal that is used to turn on the brake lights is read by the ABS system in order to determine whether or not the brake has been pressed engaged. The ABS will then only be able to become engaged if this signal shows the brakes are being used. |
| HLR5 | The ABS will receive information from a wheel speed sensor for each wheel. Each wheel speed sensor will send the speed of the wheel it is monitoring in meters/second. |
| HLR6 | Run a system diagnostic test sequence at ignition and determine if any errors are present in the system. |

## Low Level Requirements:-

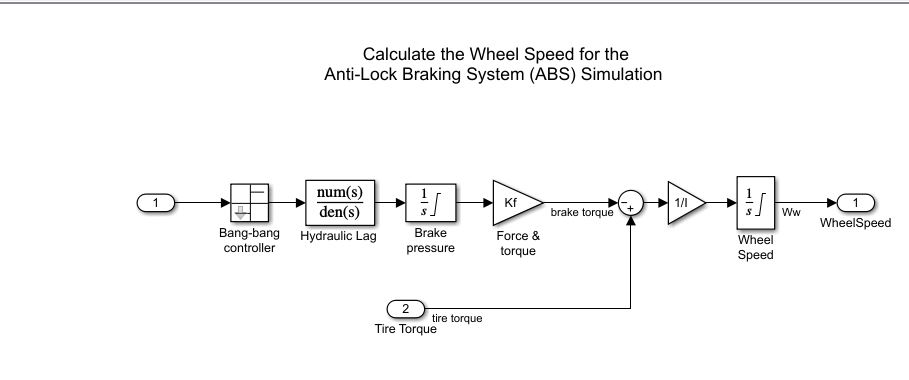
|  |  |
| --- | --- |
| ID | Description |
| LLR1 | The wheel rotates with an initial angular speed that corresponds to the vehicle speed before the brakes are applied. |
| LLR2 | The system test will engage when the car is turned on. |
| LLR3 | Calculate rotational deceleration from the wheel speed data, and determine if wheel lock-up is imminent. |
| LLR4 | The same signal that is used to turn on the brake lights is read by the ABS system in order to determine whether or not the brake has been pressed engaged. The ABS will then only be able to become engaged if this signal shows the brakes are being used. |
| LLR5 | Terminate system execution if any failure occurs form either test. The termination shall not affect normal braking behaviour of the vehicle |

**DESIGN**



**MODELING**

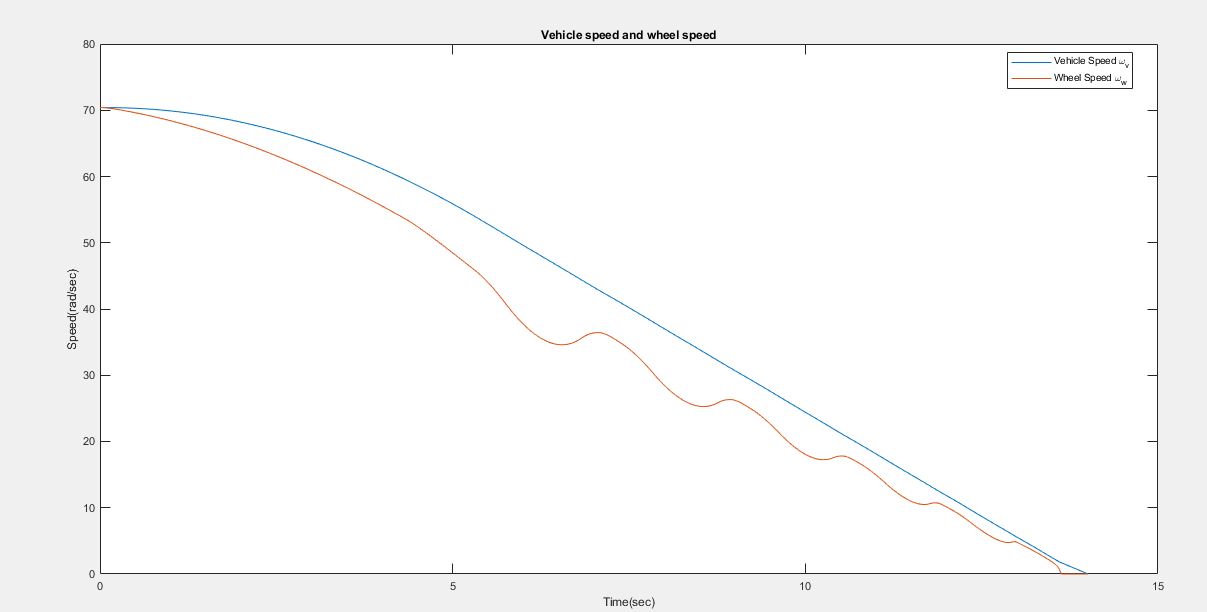
**ABS/Wheel Speed**

****

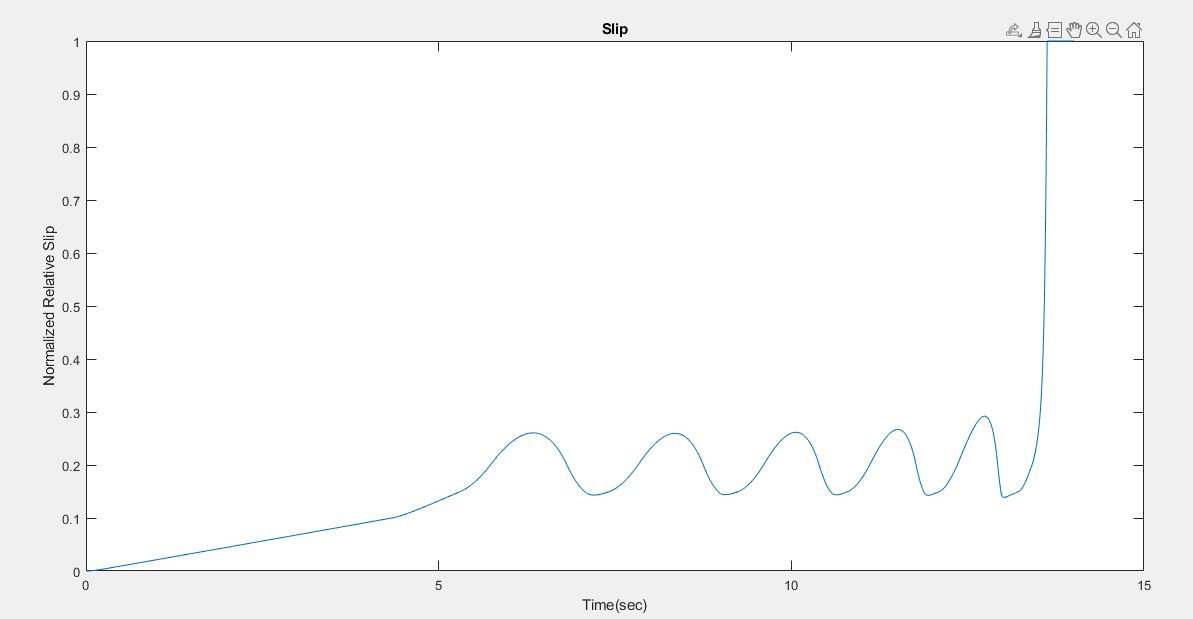
**OUTPUT**

### Running the Simulation in ABS Mode

* **Vehicle Speed and Wheel Speed**

****

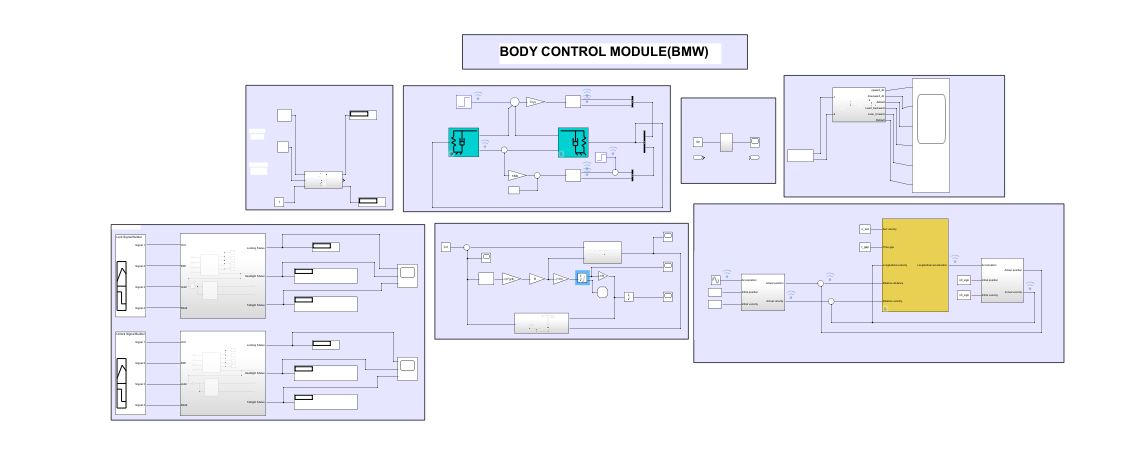
* **Slip**

****

**Conclusion**

This model shows how you can use Simulink to simulate a braking system under the action of an ABS controller. The controller in this example is idealized, but you can use any proposed control algorithm in its place to evaluate the system's performance. You can also use the Simulink® Coder™ with Simulink as a valuable tool for rapid prototyping of the proposed algorithm. C code is generated and compiled for the controller hardware to test the concept in a vehicle. This significantly reduces the time needed to prove new ideas by enabling actual testing early in the development cycle.

For a hardware-in-the-loop braking system simulation, you can remove the 'bang-bang' controller and run the equations of motion on real-time hardware to emulate the wheel and vehicle dynamics. You can do this by generating real-time C code for this model using the Simulink Coder. You can then test an actual ABS controller by interfacing it to the real-time hardware, which runs the generated code. In this scenario, the real-time model would send the wheel speed to the controller, and the controller would send brake action to the model.

**Merging with BMW (BCM MODULE)**

# Mini project 6 – Wiper Control [Team]

## Modules: - Mastering Microcontrollers with Embedded Driver Development Module

## WIPER CONTROL SYSTEM

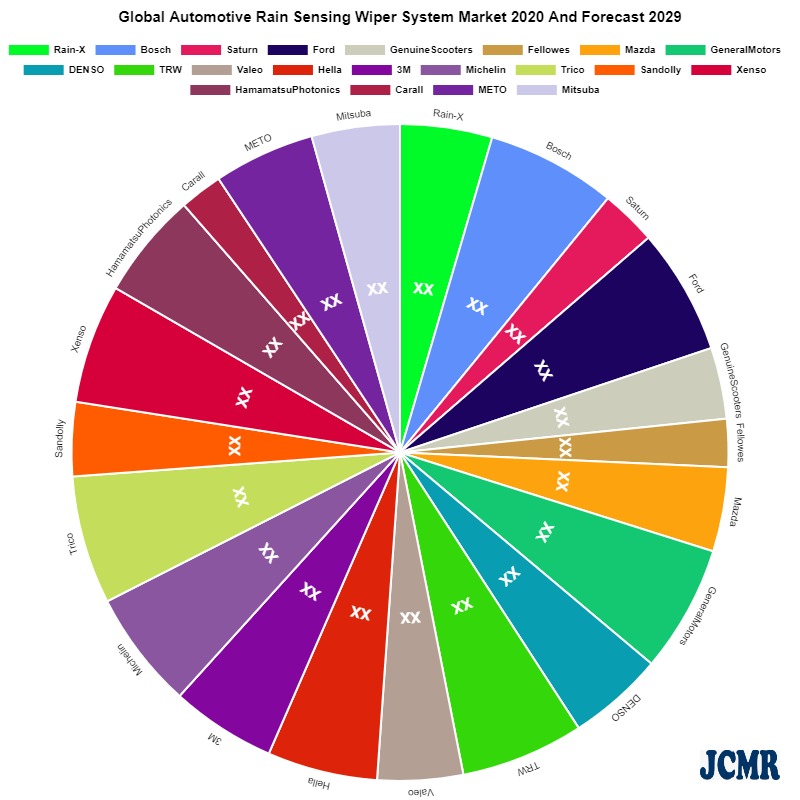
## Introduction

Automotive wipers form an essential part for any vehicle. They perform to remove water, ice, snow, and dust from a windshield of a vehicle. An automotive wiper is either powered by an electric motor or pneumatic power. Almost all motor vehicle including cars, trucks, buses, train locomotives and watercraft with a cabin are equipped with one or more such wipers. The automotive wiper market is multiplying as there is an exponentially increased production of automobiles globally.

## Features

* To achieve high safety
* To reduce man power
* To increase the efficiency of the vehicle
* To reduce the work load
* To reduce the vehicle accident
* To reduce the fatigue of workers
* To high responsibility
* Less Maintenance cost

## State of Art

[](https://user-images.githubusercontent.com/94118726/148398163-c7e29d3e-35cb-4ba6-a0f7-5c2776349f48.jpeg)

## SWOT Analysis

# Strength

* It is possible to operate Manually/automatically by proving On/Off switch
* Improve Visibility of car in rain.Can drive easily in any climatic situation.

# Weakness

* This system applied in the case of water falling on the class only.
* Addition cost is required to install this system to four wheeler.

# Opportunities

* To increase automation in vehicle driving system
* To dispense with troublesome wiper operation needed when rainfall condition change or driving condition change, including the car speed and entry to or exit from tunnels.
* To operate the wiper with response to changing rainfall or driving conditions, thus keeping the driver’s windshield clear.

# Threats

* Dust particles and non-conductive particles accumulated on the surface of sensors cannot be detected by conductive sensors.

# 4W 1H:

## Who:

* A wiper speed control system for an automotive wiper controls the operational speed of a wiper in accordance with rain conditions.

## What:

* Vehicles are now available with driver-programmable intelligent (automatic) windscreen wipers that detect the presence and amount of rain using a rain sensor.

## Where:

* It is located underneath the dashboard, above the brake and accelerator pedal, and is responsible for the complete operation of the windshield wiper system.

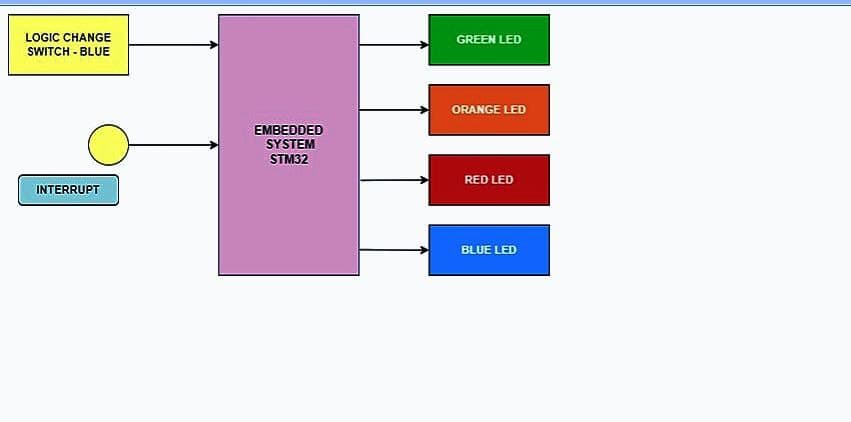
## When:

* Whenever the water hit a dedicated sensor that located on windscreen, it will send a signal to move on the wiper motor. Once water is not detected by sensor, the wiper will automatically stop. This will help the driver to give more concentration and reduce the car accident probability.

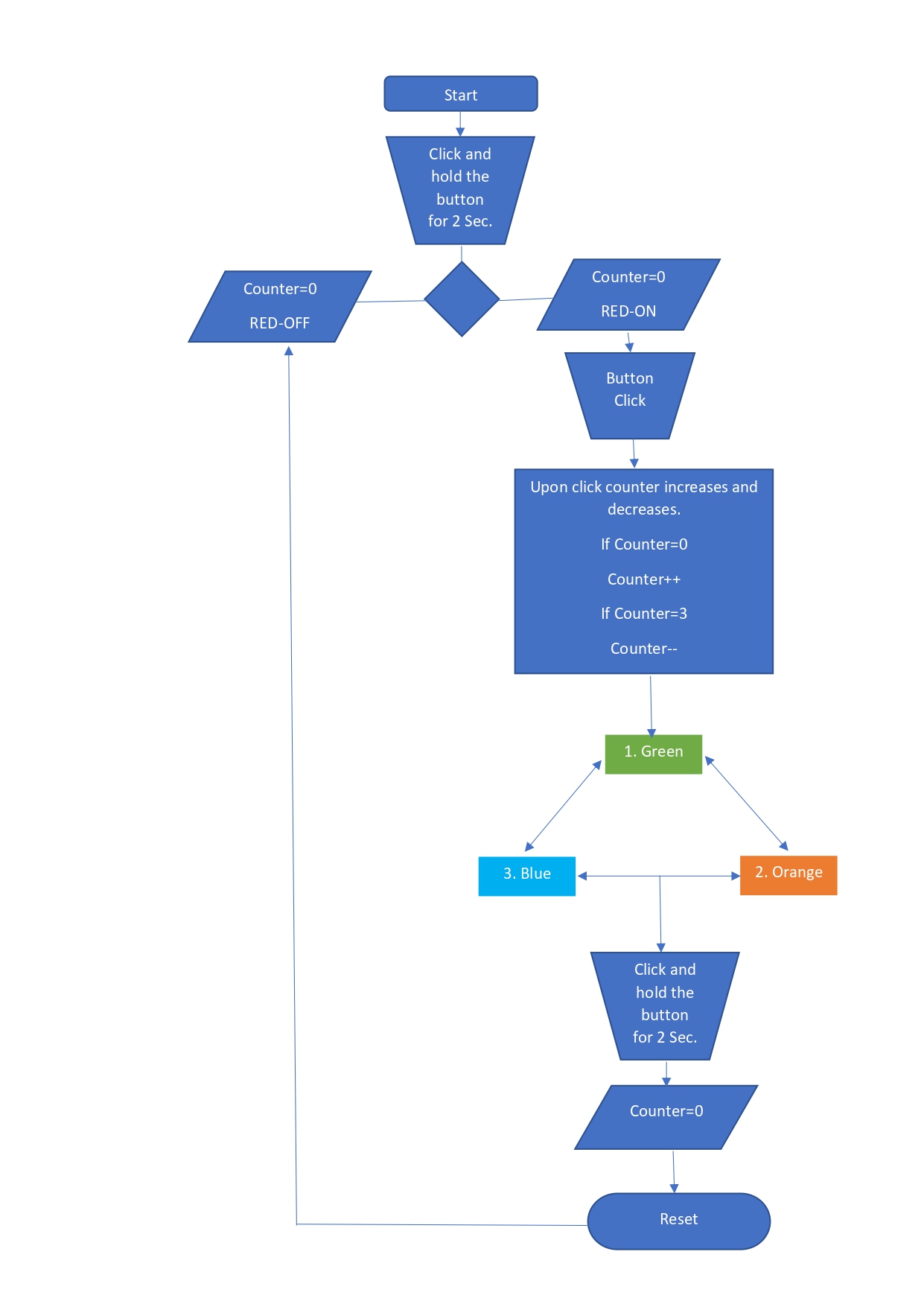
## How:

* Windshield wipers are controlled by the stalk on the right side of your steering wheel. Simply moving the stalk down will turn your windshield wipers on. Moving the stalk down will turn your you wipers on.

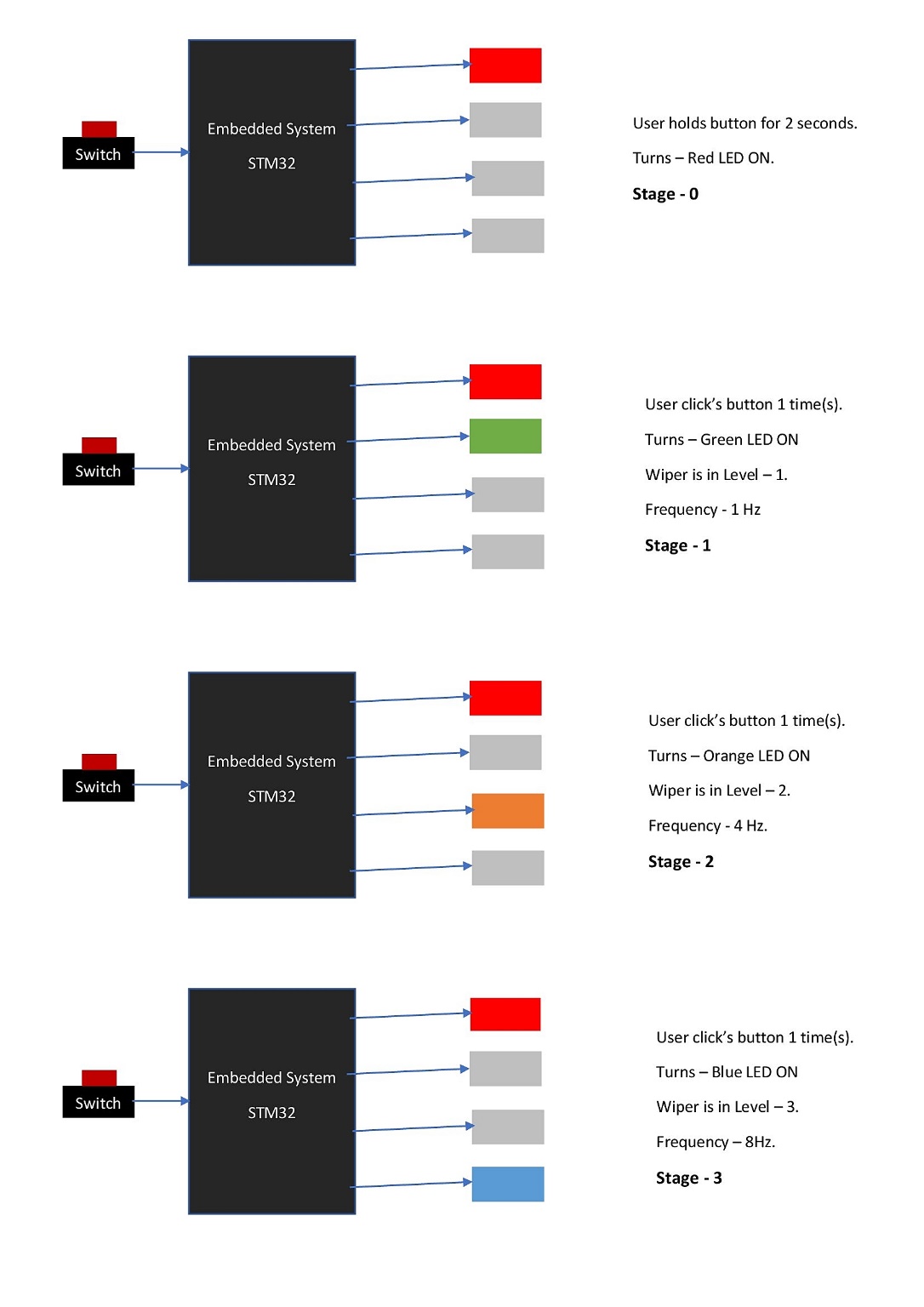
## BLOCK DIAGRAM

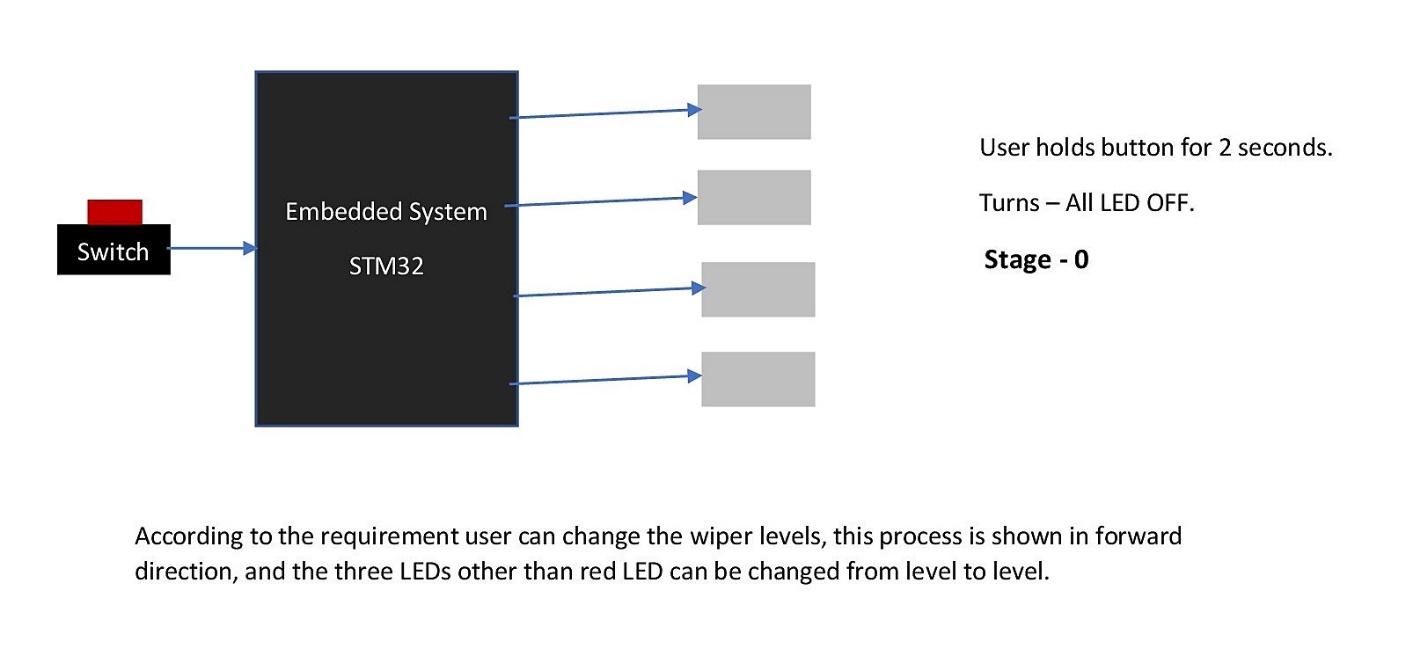
[](https://user-images.githubusercontent.com/95903618/148645700-98d07e9f-5aa4-440d-aab3-cbd3ce8df965.jpg)

## FLOW CHART

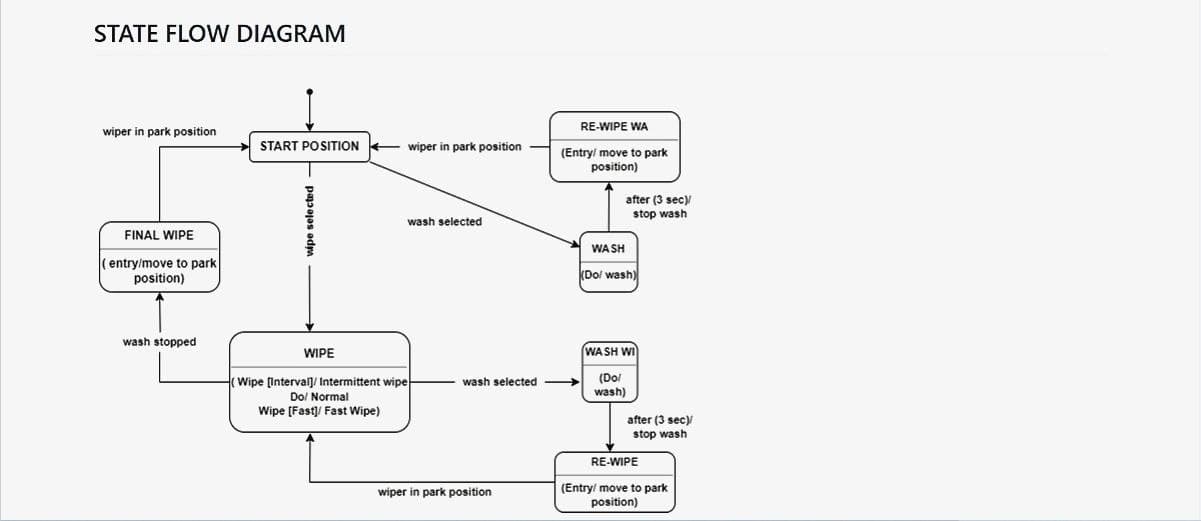
[](https://user-images.githubusercontent.com/94304459/148668701-59d6ab3f-9175-4323-979c-2bcfe6c5b797.jpg)

## CASE STUDY

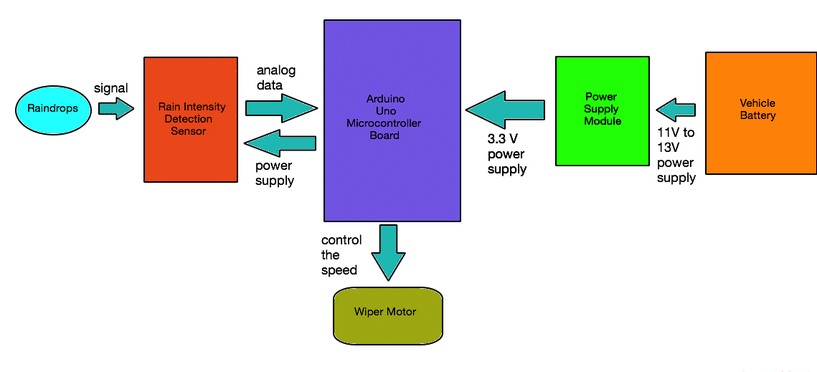
[](https://user-images.githubusercontent.com/94304459/148668959-58aff2ed-7316-4ed9-a3bd-2cc93e2ced36.jpg)

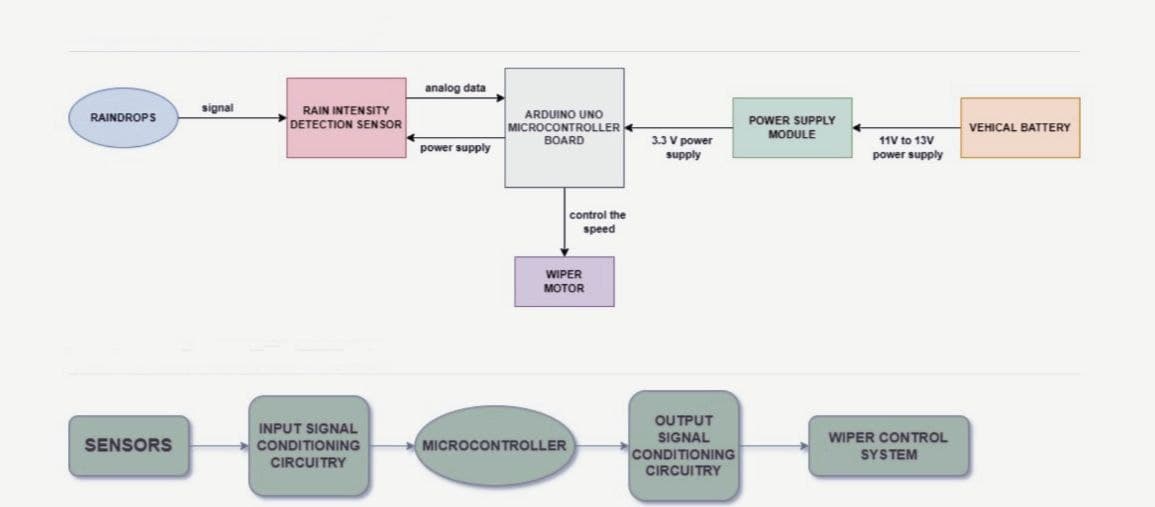
[](https://user-images.githubusercontent.com/94304459/148669088-172d72e0-923b-4256-b936-b8262cea5bf8.jpg)

## STATE FLOW

[](https://user-images.githubusercontent.com/95903618/148645916-4005d5b7-6324-43ae-af86-1acda391d920.jpg)

## SYSTEM DESIGNS

[](https://user-images.githubusercontent.com/95903618/148647630-5474cd5f-8cb2-4585-b01a-8be3d3013266.jpg)

[](https://user-images.githubusercontent.com/95903618/148649206-19629a1d-ea92-451d-931a-f18c47b4074d.jpg)

## High Level Requirements

| **ID** | **Description** |
| --- | --- |
| HLR1 | These systems detect droplets of rain on the windshield and automatically turn on and adjust the wiper system in accordance to the level of precipitation. |
| HLR2 | A windscreen wiper or windshield wiper is a device used to remove rain, snow, ice and debris from a windscreen or windshield. |
| HLR3 | Quality and reliability wiper systems meet the highest technical requirements and are the basis for vehicles with sophisticated features. |
| HLR5 | Almost all motor vehicle, including trains, aircraft and watercraft, are equipped with such wipers, which are usually an essential requirement. |
| HLR6 | Our project brings forward this system to automate the wiper system having no need for manual intervention. |

## Low Level Requirements

| **ID** | **Description** |
| --- | --- |
| LLR1 | A new mechatronic reversing system can now be used to clean the windshield with two wiper arms, whereby one wiper arm is powered directly and the other indirectly using a connection link. |
| LLR2 | Wiper motor is automatically ON during the time of rainfall. |
| LLR3 | Existing system manually used control stalk to activate wiper and the process of pulling up wiper is difficult to be handled. |
| LLR4 | Lower level parsing¶. Under the hood, the Requirement class does most of the heavy lifting... Class requirements. |
| LLR5 | These systems detect droplets of rain on the windshield and automatically turn on and adjust the wiper system. |

## Implementation and Summary

### Git Link:

Link: https://github.com/GENESIS-2022/MasteringMCU-Team38

### Individual Contribution and Highlights

1. Wiper System using C Programming
2. Source code management using GitHub
3. REQUIREMENTS+Implementation(start-up’s , STM32F407XX.H)

Role in Project Team

1. Programmer: Done Programming for Wiper System
2. Integrator: Integrated all the codes
3. Tester: Writing Test cases and testing the integrated code

# Mini project 7 – Ford Project [Team]

## Modules: - Automotive Systems

### Requirements

# Ford Aspire

The Ford Aspire nameplate has been used by the American automobile manufacturer Ford for the following cars, in the following markets: Ford Festiva, in North America from 1993 to 1997. The sedan version of the Ford Figo, a rebadged third generation Ford Ka in India since 2015.

# Body Control Module

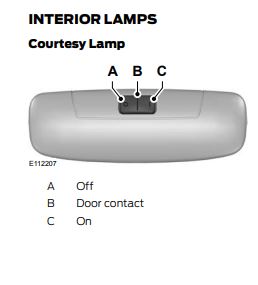
## Features:

* Door Lock System
* Interior Light Control
* Power Mirror
* Power Window

**Individual Feature**: - **Interior Light Control**

## Introduction

The interior lighting in cars consisted of a few incandescent lights that turned on or off in response to micro switches in various doors or simple switches near the light fixture.

[](https://user-images.githubusercontent.com/94118726/150636352-1b524cd2-6b02-47b6-9b41-e99ac291e1c0.JPG)

## 4W's 1H

### WHAT:

The lighting normally stays lit until the vehicle is turned on so the passengers can safely fasten their seat belts. In addition, interior lights can aid in reading maps or finding lost items in the dark and The following are a few things you should know about your car’s interior lights :( 1) Dim Light (2) Flickering Light (3)Light stays on , Etc.

### WHERE:

The interior light is used in dashboard for indication, in safety indicator and headlight of door and foot step, bonnet, boot etc.

### WHEN:

The system is standard or available on many of Ford's models, including the Fiesta, Focus, Fusion, Taurus, and Mustang cars and the Escape, Edge, Flex, and F-150 trucks, depending on the selected trim level.

### WHY:

Most vehicles have interior lights that are also called dome lights or courtesy lights. These can be located on the ceiling of the vehicle and illuminate when people enter or exit the car.

### How:-

When the interior lights in a car are working correctly, they will usually come on when you open your door and then shut off some time after you close the door. This process relies on a switch in the door jamb that opens when you open the door and closes when you close the door.

**Requirement**

## High Level Requirement:-

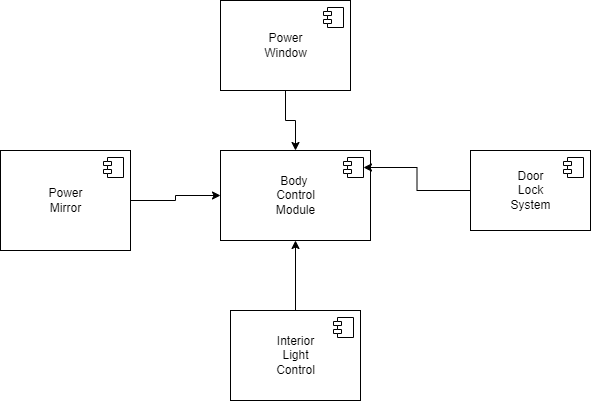
| **High level Requirement** | **Description** |
| --- | --- |
| HLR\_1 | Light will be on when Door is open |
| HLR\_2 | Light will be in on state until all the door is correctly closed |
| HLR\_3 | Dashboard Light will ON when the car is unlocked |
| HLR\_4 | Lights wiil be OFF after 10 sec when the lock button pressed on the key |
| HLR\_5 | In Night Foot step light is automatically ON |
| HLR\_6 | Lights can be turn off and on manually with the switch |

**Low Level Requirement:-**

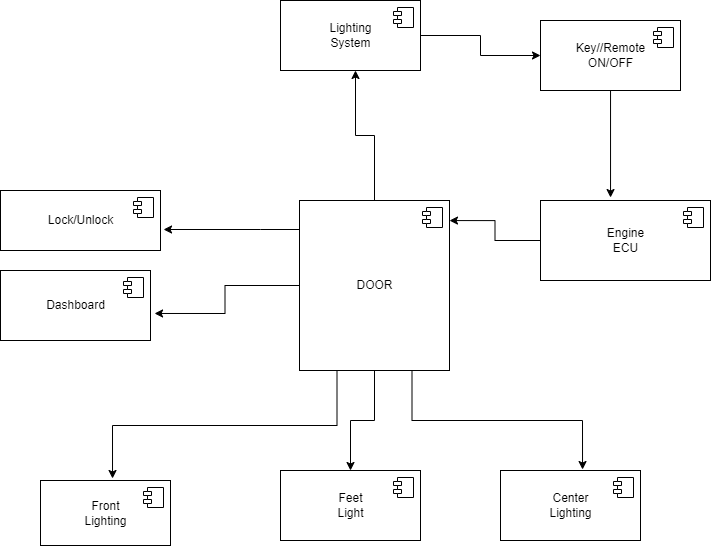
| **Low level Requirement** | **Description** |
| --- | --- |
| LLR\_1 | Voice command to turn on and off the light |
| LLR\_2 | Lights will be turned on when unlockes the car with the remote key from outside |
| LLR\_3 | Add Multicolour LED |
| LLR\_4 | The light will turn-off after 5 seconds when we lock the car through the key |
| LLR\_5 | Safety Indicator |

## Design

# Body Control Module

[](https://user-images.githubusercontent.com/94118726/150649288-0258c06a-5755-4b9c-b4db-26da7486f14e.png)

## Interior Light Control

[](https://user-images.githubusercontent.com/94118726/150648766-a63f203e-bcc9-4fdc-a4c8-9a468ee191b5.png)

## Implementation and Summary

### Git Link:

https://github.com/Pradnya579/Automotive\_Ford\_Team

### Individual Contribution and Highlights

* 1. Interior Light Control System Case Study

Role in Project Team

1. Designer: Done Designing for Project
2. Researcher: Done case study for Body Control Module of Ford.

# Mini project 8 – EV Truck [Team]

## Module: - Applied Control Systems and Vehicle Dynamics

# EV TRUCK

An electric truck is an electric vehicle powered by batteries designed to transport cargo, carry specialized payloads, or perform other utilitarian work. Electric trucks have serviced niche applications like milk floats, pushback tugs and forklifts for over a hundred years, typically using lead-acid batteries, but the rapid development of lighter and more energy-dense battery chemistries in the twenty-first century has broadened the range of applicability of electric propulsion to trucks in many more roles. Electric trucks reduce noise and pollution, relative to internal-combustion trucks. Due to the high efficiency and low component-counts of electric power trains, no fuel burning while idle, and silent and efficient acceleration, the costs of owning and operating electric trucks are dramatically lower than their predecessors.

THE FUTURE OF HEAVY - DUTY VEHICLES = ELECTRIC TRUCKS

Electric trucks are becoming more popular due to its cost efficiency, better performance and lower emissions. Global sales of electric vehicles increased by 43% in 2020 and is expected to keep growing in upcoming years. The same will slowly apply on EV trucks and companies will adapt its business models accordingly.

# ARE ELECTRIC TRUCKS BETTER?

In urban areas, delivery routes with heavy traffic and lots of stops can become very costly. Electric trucks are roughly 50% more effective than diesel trucks, which makes them roughly 20% less expensive than diesel trucks. However, it largely depends on how the trucks will be used. Speeding, braking, exceeding RPM and many other aspects influence the consumption of the EV battery, which leads to a lower battery health and therefore, increased need for charging. Researchers analysed driving behaviour, number of stops, speeding and overall usage of the truck, and electric trucks clearly outperformed diesel trucks. By driving an EV truck, you use around 30% less total energy, reduce greenhouse gas emissions by roughly 50%, and lower down the noise level of the vehicle. The noise disturbance is a real issue, especially with older diesel trucks that produce a lot of noise and emissions. This challenge is typically solved with a right driver management system, which is a set of measurements to improve driving experience and manage fleet drivers safely and effectively.

# BENEFITS OF ELECTRIC TRUCKS

* Lower emissions
* Lower maintenance
* Lower noise disturbance
* Better performance
* Increased efficiency

It is always cheaper to charge your electric truck than spending money on gas. Electric trucks provide businesses with many benefits that primarily aim for the long run. EV trucks do not require fuel, which is already one of the biggest advantages, due to fuel cost and effect on nature. Driving electric trucks reduces CO2 emissions and actually offers better performance for drivers.EV trucks also have less parts, which should lead to less damage and lower maintenance. However, this depends on a truck model and its usage. While driving within urban areas with frequent stops and speeding, it is way more efficient to drive electric truck than diesel truck.

# Argument for Electric Trucks

WEIGHT - Commercial battery electric vehicle (CBEV) weight is not an issue

TECHNOLOGY - CBEV technology is proven and here now it will last beyond 10 years, Maintenance will be less costly

COST - it will be competitively priced, less expensive to operate and command a premium at resale

CHARGING - Trust the market to provide Commercial battery electric vehicle charging solutions, the grid and market will evolve with CBEVs

# Argument against Electric Truck’s

WEIGHT - Vehicle tare weight is too high to support my freight needs

TECHNOLOGY - Technology is not ready, Maintenance may not be less costly and Vehicle life is too short

COST - Vehicle purchase price is too high for a positive ROI, Operating costs are too great for positive ROI and residual value is questionable

CHARGING - Charging infrastructure is not ready, Charging Infrastructure is not fast enough, and the electric grid cannot support growth in electric vehicles

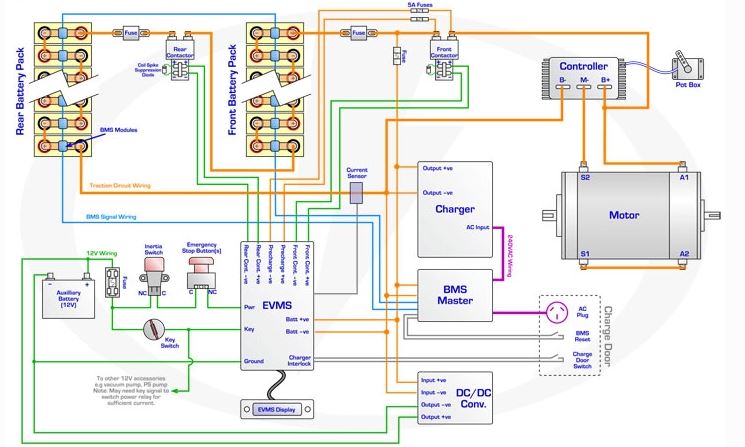
# Research

 The Volvo FE Electric trucks join NFI’s fleet of more than 4,500 heavy-duty tractors that support its dedicated transportation and port drayage services for customers spanning from manufacturing to retail. The pilot trucks will be based out of one of NFI’s warehouse facilities in Southern California that serves as a central distribution centre for the region. “As the future of goods movement in the U.S. changes from more of a long-haul operation to regional and hub and spoke models, not only is that NFI's wheelhouse, it’s an ideal scenario to immerse electrification into our regional hauling strategy,” said Jim O’Leary, vice president, Assets/Fleet Services, NFI Industries. “Our executive team is excited to collaborate with the Volvo LIGHTS team to accelerate our transition to a zero-emission fleet, so that we can lower our carbon footprint, reduce our operating costs and provide a better work environment for our drivers.”

## Reference:

* Integrated, feed-forward hybrid electric vehicle simulation in SIMULINK and its use for power management studies
* Energy management strategy for a parallel hybrid electric truck,
* Energy management strategy for a parallel hybrid electric truck
* <https://ieeexplore.ieee.org/abstract/document/7587102>
* <https://www.tandfonline.com/doi/abs/10.1080/00423110412331291553>

# Simulation Design:

[](https://user-images.githubusercontent.com/94118726/152677873-fbcdd4b0-3733-4d6f-ba50-6f4756168263.JPG)

# Analysis

## TWO MODEL COMPARISON

| **SPECIFICATIONS** | **Volvo Fe** | **Mack LR** |
| --- | --- | --- |
| • COST | 39,900$ | ₹ 15.29 Lakh - ₹ 16.82 Lakh |
| • GROSS COMBINATION WEIGHT | Up to 27 Tonnes | 7300 kg |
| • RANGE | UP to 200 km | More than 100 KMs |
| • BATTERY | Lithium-ion batteries | Lithium-ion batteries |
| • BATTERY CAPACITY | 200-265 kWh, 3,4 batteries | 62.5 kWh-Octillion Make-(10 S1P) |
| • CHARGING TIME(FULL CHARGE) | 11h with AC (22 kW) & 2h with DC (150 kW) | 2 hrs. |
| • DRIVELINE/MOTOR | 2 electric motors, 2-Speed gearbox Max torque electric motors 850 Nm. Max torque rear axle 28 kNm. | The 4SPCR engine that offers 100hp of power along with 300Nm of torque from 1,200 to 2,200rpm and brushless asynchronous induction motor |
| • PERFORMANCE | Up to 225kw power (300hp) | 100hp |
| • ELECTRIC MOTOR TORQUE FOR PTO(PEAK/CONTINUOUS) | 530 Nm/270 Nm | 300Nm |
| • INVERTER | Traction | Traction |
| • WHEEL BASE | From 3 900 mm up to 5000 mm1 | 3310mm |
| • CONTROLLER | PI Controller | PI controller |

# EV MODEL DESIGN

* COST – ₹ 15.29 Lakh to ₹ 16.82 Lakh
* GROSS COMBINATION WEIGHT – 7300 kg
* RANGE – More than 100 KMs
* BATTERY - nickel-metal hydride
* BATTERY CAPACITY – 60–120 wH/kg
* CHARGING TIME(Full Charge) – 3-4hrs
* DRIVELINE/MOTOR – Permanent Magnet Synchronous Motor (PMSM)
* PERFORMANCE – 100hp
* ELECTRIC MOTOR TORQUE PTO(peak/continuous)- 180Nm
* INVERTER – Grid-tied string
* WHEEL BASE - 3310mm
* CONTROLLER – PID Controller

# Conclusion

The progress that the electric vehicle industry has seen in recent years is not only extremely welcomed, but highly necessary in light of the increasing global greenhouse gas levels. As demonstrated within the economic, social, and environmental analysis sections of this webpage, the benefits of electric vehicles far surpass the costs. The biggest obstacle to the widespread adoption of electric-powered transportation is cost related, as gasoline and the vehicles that run on it are readily available, convenient, and less costly. As is demonstrated in our timeline, we hope that over the course of the next decade technological advancements and policy changes will help ease the transition from traditional fuel-powered vehicles. Additionally, the realization and success of this industry relies heavily on the global population, and it is our hope that through mass marketing and environmental education programs people will feel incentivized and empowered to drive an electric-powered vehicle. Each person can make a difference, so go electric and help make a difference!

## Implementation

### Git Link:

Link: https://github.com/robin6119/Team5\_EV\_Truck

# Mini project 9 – Interior Light Control System [Individual]

## Module: - Classic Autosar Basic to Intermediate

### Requirements

**High Level Requirement:**

| **High level Requirement** | **Description** |
| --- | --- |
| HLR\_1 | Light will be on when Door is open |
| HLR\_2 | Light will be in on state until all the door is correctly closed |
| HLR\_3 | Dashboard Light will ON when the car is unlocked |
| HLR\_4 | Lights will be OFF after 10 sec when the lock button pressed on the key |
| HLR\_5 | In Night Foot step light is automatically ON |
| HLR\_6 | Lights can be turn off and on manually with the switch |

**Low Level Requirement:**

| **Low level Requirement** | **Description** |
| --- | --- |
| LLR\_1 | Voice command to turn on and off the light |
| LLR\_2 | Lights will be turned on when unlocks the car with the remote key from outside |
| LLR\_3 | Add Multicolour LED |
| LLR\_4 | The light will turn-off after 5 seconds when we lock the car through the key |
| LLR\_5 | Safety Indicator |

## Design

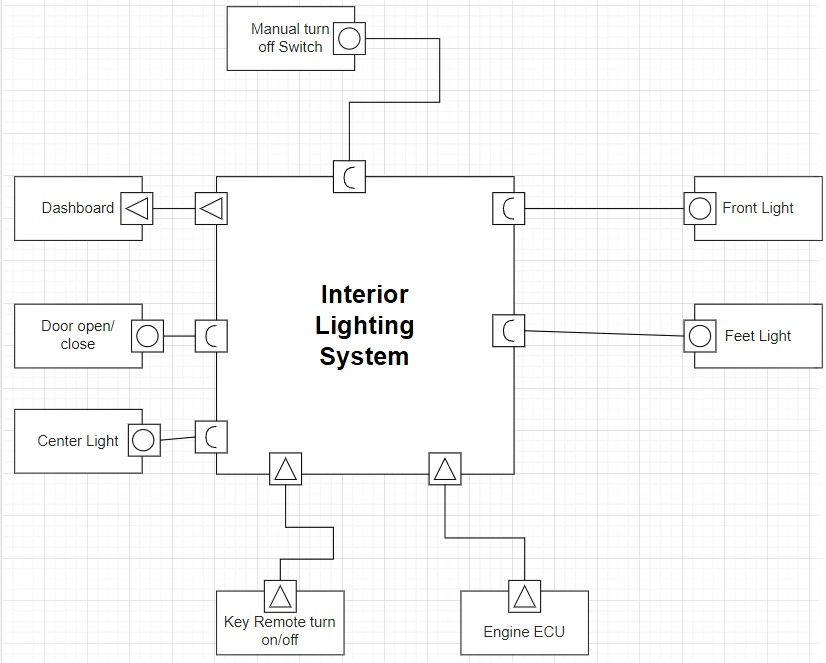


Figure 18 VFB Diagram

## 

## Implementation and Summary

### Git Link:

Link: <https://github.com/Pradnya579/Automotive_Ford_Team>

Individual Contribution and Highlights

1. Interior Light Control Case Study
2. Source code management using GitHub
3. AtomicSwComponent
4. SWCInternalBehavior
5. SWCImplementation