**Details**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Ver. Rel. No. | Release Date | Prepared By | Reviewed By | To Be Approved | Remarks/Revision Details |
| 1.0 | 16/02/2022 | Ragini Bhandare  40021048 |  |  |  |

Contents

[Miniproject – 1: Mini Voting System [Individual] 5](#_Toc95931104)

[Modules: 5](#_Toc95931105)

[Requirements 5](#_Toc95931106)

[High Level Requirements 6](#_Toc95931107)

[Low Level Requirements 6](#_Toc95931108)

[Design 7](#_Toc95931109)

[Test Plan 8](#_Toc95931110)

[High Level Test Plan 8](#_Toc95931111)

[Low Level Test Plan 9](#_Toc95931112)

[Implementation and Summary 9](#_Toc95931113)

[Git Link: 9](#_Toc95931114)

[Git Dashboard 10](#_Toc95931115)

[Summary 10](#_Toc95931116)

[Git Inspector Summary 10](#_Toc95931117)

[Miniproject 2 – Embedded PC Control Using TV Remote [Individual] 11](#_Toc95931118)

[Modules 11](#_Toc95931119)

[Requirements 11](#_Toc95931120)

[High Level Requirements 11](#_Toc95931121)

[Low Level Requirements 12](#_Toc95931122)

[Design 12](#_Toc95931123)

[Test Plan 14](#_Toc95931124)

[High Level Test Plan 14](#_Toc95931125)

[Low Level Test Plan 15](#_Toc95931126)

[Implementation and Summary 15](#_Toc95931127)

[Git Link: 15](#_Toc95931128)

[Git Dashboard 15](#_Toc95931129)

[Miniproject 3 – Scientific Calculator [Team] 16](#_Toc95931130)

[Modules 16](#_Toc95931131)

[Requirements 16](#_Toc95931132)

[High Level Requirements 17](#_Toc95931133)

[Low Level Requirements 17](#_Toc95931134)

[Design 18](#_Toc95931135)

[Test Plan 19](#_Toc95931136)

[High Level Test Plan 19](#_Toc95931137)

[Low Level Test Plan 20](#_Toc95931138)

[Implementation and Summary 20](#_Toc95931139)

[Git Link: 20](#_Toc95931140)

[Individual Contribution and Highlights 20](#_Toc95931141)

[Summary 20](#_Toc95931142)

[Miniproject 4 – Calender Automation[Team] 21](#_Toc95931143)

[Modules 21](#_Toc95931144)

[Requirements 21](#_Toc95931145)

[High Level Requirements 21](#_Toc95931146)

[Low Level Requirements 21](#_Toc95931147)

[Test Plan 22](#_Toc95931148)

[High Level Test Plan 22](#_Toc95931149)

[Low Level Test Plan 22](#_Toc95931150)

[Implementation and Summary 23](#_Toc95931151)

[Git Link: 23](#_Toc95931152)

[Git Dashboard 24](#_Toc95931153)

[Git Inspector Summary 24](#_Toc95931154)

[Individual Contribution and Highlights 24](#_Toc95931155)

[Miniproject 5 – Ford Project[Team] 25](#_Toc95931156)

[Modules 25](#_Toc95931157)

[Requirements 25](#_Toc95931158)

[Design 25](#_Toc95931159)

[Miniproject 6 – Wiper Control[Team] 26](#_Toc95931160)

[Modules 26](#_Toc95931161)

[Requirements 26](#_Toc95931162)

[High Level Requirements 26](#_Toc95931163)

[Low Level Requirements 27](#_Toc95931164)

[Design 28](#_Toc95931165)

[Test Plan 30](#_Toc95931166)

[High Level Test Plan 30](#_Toc95931167)

[Low Level Test Plan 30](#_Toc95931168)

[Implementation and Summary 31](#_Toc95931169)

[Git Link: 31](#_Toc95931170)

[Individual Contribution and Highlights 31](#_Toc95931171)

[Miniproject 7 – Ford Project[Team] 32](#_Toc95931172)

[Modules 32](#_Toc95931173)

[Requirements 32](#_Toc95931174)

[Design 33](#_Toc95931175)

[Implementation and Summary 33](#_Toc95931176)

[Git Link: 33](#_Toc95931177)

[Individual Contribution and Highlights 33](#_Toc95931178)

[Miniproject 8 – EV Truck[Team] 34](#_Toc95931179)

[Modules 34](#_Toc95931180)

[Requirements 34](#_Toc95931181)

[Implementation and Summary 35](#_Toc95931182)

[Individual Contribution and Highlights 35](#_Toc95931183)

[Miniproject 9 – Power Mirror[Individual] 36](#_Toc95931184)

[Modules 36](#_Toc95931185)

[Requirements 36](#_Toc95931186)

[Design 36](#_Toc95931187)

[Implementation and Summary 37](#_Toc95931188)

[Git Link: 37](#_Toc95931189)

[Individual Contribution and Highlights 37](#_Toc95931190)

## 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: Mini Voting System [Individual]

## Modules: C Programming On Multiple Platforms

### Requirements

## Introduction

Electronic voting (also known as e-voting) is voting that uses electronic means to either aid or take care of casting and counting votes.

Depending on the particular implementation, e-voting may use standalone electronic voting machines (also called EVM) or computers connected to the Internet. It may encompass a range of Internet services, from basic transmission of tabulated results to full-function online voting through common connectable household devices. The degree of automation may be limited to marking a paper ballot, or may be a comprehensive system of vote input, vote recording, data encryption and transmission to servers, and consolidation and tabulation of election results.

# Objective:-

Electronic voting technology intends to speed the counting of ballots, reduce the cost of paying staff to count votes manually and can provide improved accessibility for disabled voters. Also in the long term, expenses are expected to decrease.[6] Results can be reported and published faster.[7] Voters save time and cost by being able to vote independently from their location. This may increase overall voter turnout. The citizen groups benefiting most from electronic elections are the ones living abroad, citizens living in rural areas far away from polling stations and the disabled with mobility impairments.

* Security
* Accuracy
* Integrity
* Swiftness
* E voting which is physically supervised by representatives of governmental or independent electoral authorities (e.g. electronic voting machines located at polling stations);

## SWOT Analysis

## Strengths:

* Electronic voting technology intends to speed the counting of ballots
* reduce the cost of paying staff to count votes manually and can provide improved accessibility for disabled voters.
* Also in the long term, expenses are expected to decrease. Results can be reported and published faster.

## Weakness:

* They may require an additional application or registration. -Observing remote voting solutions may be more complex/difficult to organise than iperson voting
* There may be information asymmetry between voters who vote in advance and those who vote on Election Day
* Remote voting solutions which take place in an uncontrolled environment may present a higher risk of fraud, coercion, family voting, impersonation, violation of ballot secrecy or other compromises to the integrity of the vote.

## Opportunities:

* In practice, the impact on turnout may depend also on other features of the electoral system, such as the existing remote voting options available to the voter or the specific design and implementation of such options.
* It is important to stress that there are several factors related to turnout and, thus, one should be wary in assuming that simply adding a voting option leads to a great increase in participation.

## Threats:

* Votes usually need to be cast in advance. From this moment until Election Day the voter may change their electoral decision if new information becomes available.
* Due to the limitations on the number of proxy votes per person, there might not be enough proxies.

# High Level Requirements

| **ID** | **Description** | **Status** |
| --- | --- | --- |
| HLR\_1 | It records the basic information of the voter | Implemented |
| HLR\_2 | It can display all records of voter | Implemented |
| HLR\_3 | It can add delete and modify the basic information of voter | Implemented |

# Low Level Requirements

| **ID** | **Description** | **Status** |
| --- | --- | --- |
| LLR\_1 | Basic information of voter displayed | Implemented |
| LLR\_2 | Input from the user | Implemented |
| LLR\_3 | Exit the program | Implemented |

## Design

# Behavioral Diagram[behavioural diagram](https://user-images.githubusercontent.com/83066731/142928969-b7124a3e-2283-4587-af2f-523747d76fe7.png)

# Structural Diagram

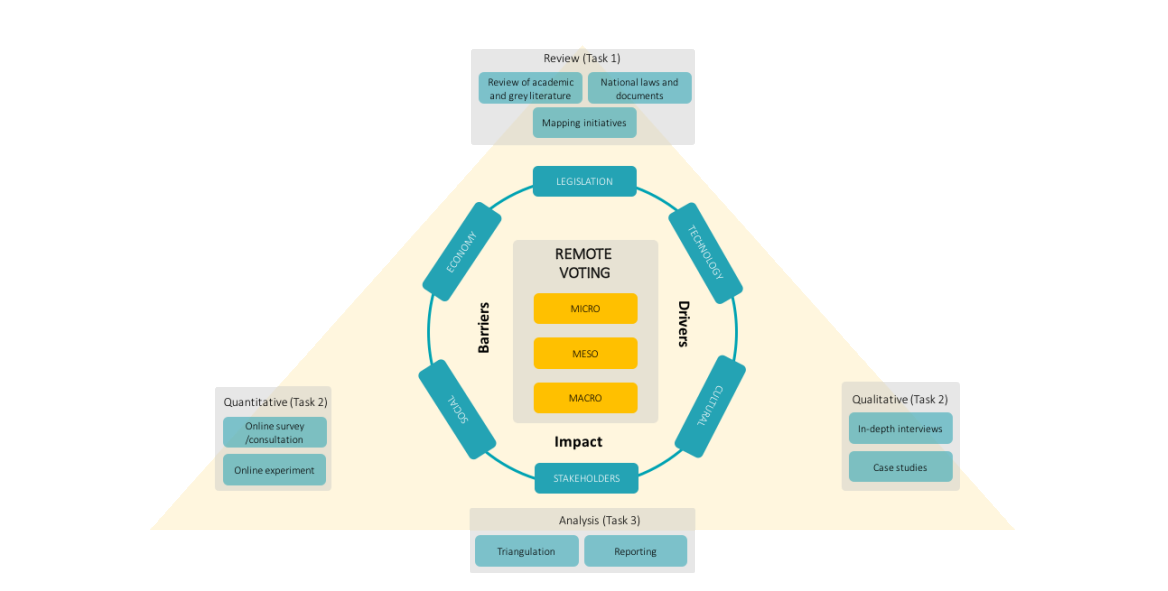
](https://user-images.githubusercontent.com/83066731/142928424-f7b06636-e9d0-444f-9213-d55263b217cd.png)

Figure 2 Flow Chart

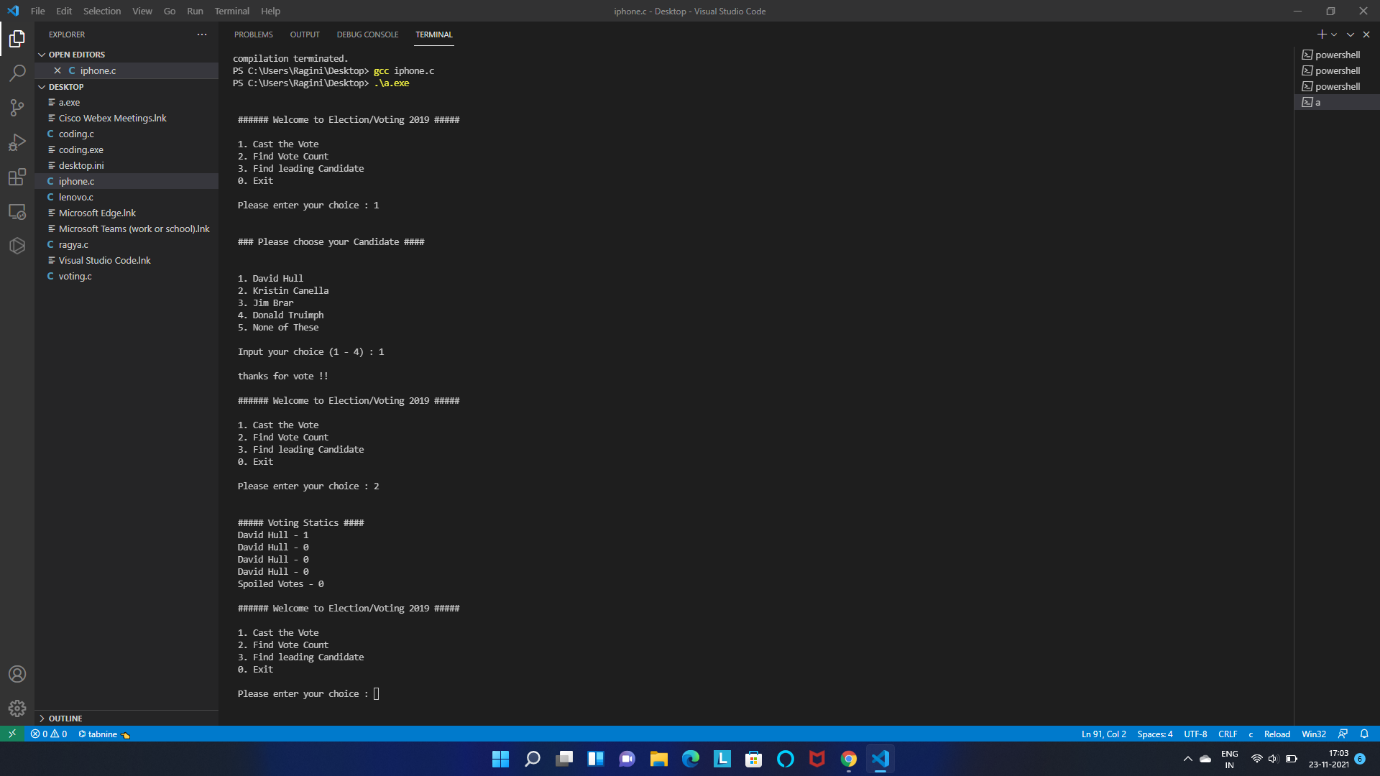
## Test Plan

| **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 |

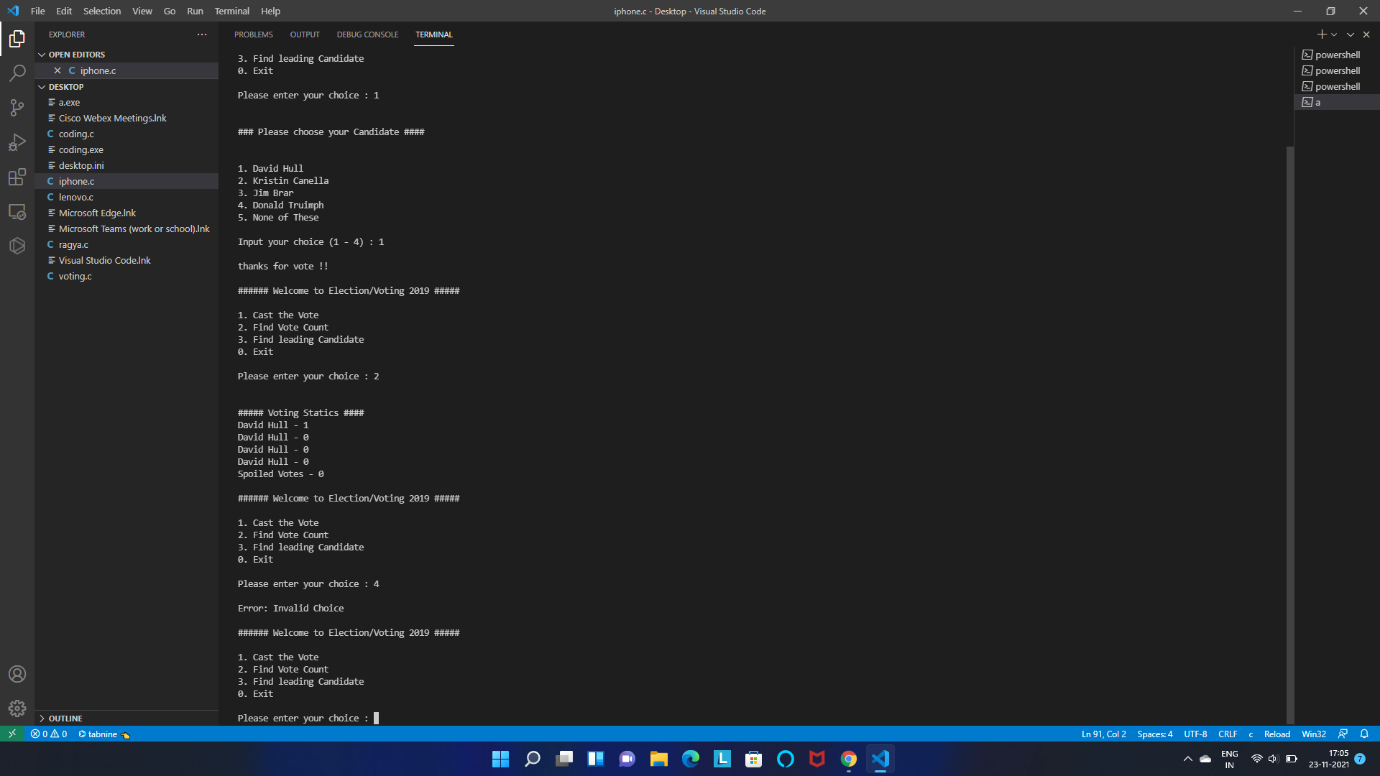
## Test Case

# 

## First displaying name of the candidates

](https://user-images.githubusercontent.com/83066731/143098340-ee52f91a-f344-481b-b9b7-596479d1e7f6.png)

## Displaying voting statistics

](https://user-images.githubusercontent.com/83066731/143098494-34da844a-f116-4c79-a7c5-06d79dbaff25.png)

## Implementation and Summary

### Git Link:

Link: https://github.com/raginibhandare/m1\_App\_mini-voting-system.git

**CERTIFICATION DONE IN MODULE**

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

# Mini project 2 – PC Control Using TV Remote[Individual]

## Module: - Essentials of Embedded System

## Topic: - PC Control Using TV Remote

# REQUIREMENTS

## Title : PC Control Using TV Remote.

### Introduction

* The objective of our project is to develop a microcontroller based system to control computer mouse with an infrared remote. To accomplish the task we have used a Sony TV remote, an Atmega8 microcontroller, a 38 KHz IR receiver with an external circuit. For connecting the system with PC we have chosen USB interfacing. The program code is developed in AVR Studio. Our circuit is very simple and easy to understand. We have tried to make the device user friendly as much as we can. It is compatible in almost all PC with operating system Microsoft®Windows XP, Windows Vista or Windows 7. We hope anyone will have fun using our device.

### State of Art

* The main goal of the project is to control mouse cursor and windows media player with TV remote. This is done with the implementation of RC5-remote on an 8052microcontroller.
* Here the IR receiver is connected to the microcontroller. The microcontroller is connected to the pc through RS232.
* When a certain key is pressed in the remote, it sends infrared signal through its IR transmitter to the IR receiver which is connected to the 8052 microcontroller the received infrared signal is decoded by using the program written on the ROM of the microcontroller
* Hence the operations of cursor and windows media player are performed according to the key pressed.

### SWOT (Strengths,Weakness,Opportunities,Threats) Analysis

#### Strengths

* Two in one facility (TV, PC mouse)

#### Weakness

* The circuit is very much dependent on the sensitivity of IR receiver

#### Opportunities

* Just changing the code we can also interface the Keyboard using the same circuit and remote.

#### Threats

* The TV remote available in market is not good enough, so sometimes it may be needed to press a button repeatedly.

### 4W's

Where

* In Industries,Homes

Why

* To maintain ease and comfort.

When

* When watching movie and chilling you need not have to get up.

### 1H's

How

* Using ADC,RS 232, ATMega328

### Tools Used

* avr-gcc
* Visual Studio Code
* Makefile
* C/C++

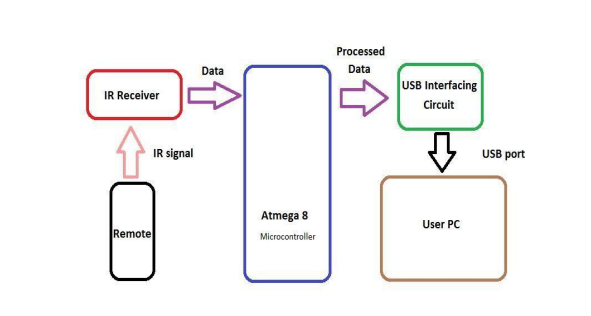
### High Level Requirements

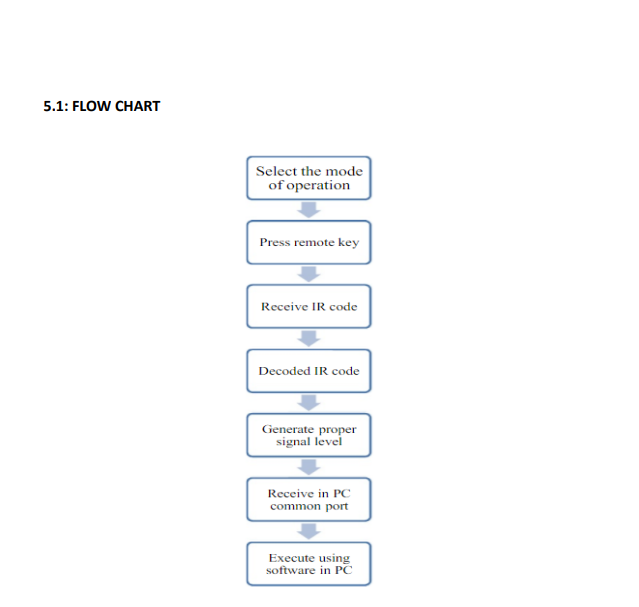
| **HLR\_no** | **Requirement** |
| --- | --- |
| HLR1 | Simulide |
| HLR2 | LCD Display |
| HLR3 | ATMega328 |

### Low Level Requirements

| **LLR\_no** | **Requirement** |
| --- | --- |
| LLR1 | Button |
| LLR2 | LED |
| LLR3 | Resistors |

## Design

](https://user-images.githubusercontent.com/83066731/144302739-6f248ddc-1f63-441e-a904-e5592842e479.png)

[](https://user-images.githubusercontent.com/83066731/144301804-ade236cb-0cf4-47f3-b40d-fe0fb8b1d0d6.png)

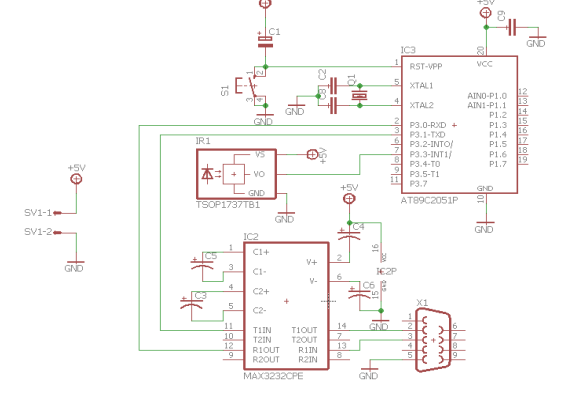
[](https://user-images.githubusercontent.com/83066731/144301887-1c1f4331-dd08-4b46-a338-e1f3185cd47b.png)

Figure 7 Structural Diagram

Figure 8 Simulation

## Test Plan

#### TEST PLAN:

### High Level Test plan:

| **ID** | **Description** | **Expected O/P** | **Actual O/P** |
| --- | --- | --- | --- |
| H\_01 | Cursor Moving | PASSED | SUCCESS |
| H\_01 | Sensing | PASSED | SUCCESS |
| H\_03 | enable blinking led | PASSED | SUCCESS |

### Low Level Test Plan:

| **ID** | **Description** | **Expected O/P** | **Actual O/P** |
| --- | --- | --- | --- |
| L\_01 | Open the app | PASSED | SUCESS |
| L\_03 | Reserve seat | PASSED | SUCCESS |

## 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/raginibhandare/M2-embedded\_project-goal.git

# Mini project 3 – Scientific Calculator

## Modules: - Applied SDLC and Software Testing

# Requirements

## Introduction

It is a advanced calculator that will allow users to perform operations in mathematics ( Arithimetic, Trigonometric, Matrices, Conversion etc) . However, the input has to be provided by user. The input values can be from any integer to even a number with decimals. Moreover, this calculator is smart enough to operate all the operations.

# **Features**

* It can do all operations (Addition, Subtraction, Multiplication, Division).
* Logarithmic operations, Exponential operations are also available.
* Power functions, Factorial and Conversions which are helpful for students are added.
* Basic Trigonometric Operations are also available.
* Basic Matrix Operations are also available.
* It has double precision.

### Cost v/s features

[](https://user-images.githubusercontent.com/78853987/107947612-a2bea200-6fb8-11eb-9838-2495a79aabf7.PNG)

## SWOT ANALYSIS

# **SWOT- Strengths, and Weakness, Opportunities Threats**

**Strengths**

• User Friendly

• All basic operations

• Double Precision

• Trigonometric operations

• Matrix Operations

**Weakness**

• Limited Operations

• Memory Wastage

**Opportunities**

• It can be expanded by adding additional features like Inverse Trigonometric operations, Equations etc.

**Threats**

• There are other programmable calculators which may affect our product marketing.

# 4W's and 1'H

**4W1H**

**What** : Scientific Calculator.

**Where** : Used in Provisional Stores and Students.

**When** : For Complex Calculations.

**Why** : For Easy and faster Calculations.

**How** : Operates by using User Inputs.

# Detail Requirements

# **High Level Requirements**

|  |  |
| --- | --- |
| **High level Requirement** | **Description** |
| HLR1 | Basic Operations |
| HLR2 | Trigonometric Operations |
| HLR3 | |  | | --- | | Matrix Operations | |  | |
| HLR4 | Conversions |
| HRL5 | Advanced operations |

# **Low-Level Requirements**

|  |  |  |
| --- | --- | --- |
| Low Level Requirement | Description | **Related HLR** |
| LLR1 | Addition | HLR1 |
| LLR2 | Subtraction | HLR1 |
| LLR3 | Multiplication | HLR1 |
| LLR4 | Division | HLR1 |
| LLR5 | Sine | HLR1 |
| LLR6 | Cos | HLR1 |
| LLR7 | Tan | HLR1 |
| LLR8 | Cosec | HLR1 |
| LLR9 | sec | HLR1 |
| LLR10 | Cot | HLR1 |
| LLR11 | matrix addition | HLR1 |
| LLR12 | matrix subtraction | HLR1 |
| LLR13 | matrix multiplication | HLR1 |
| LLR14 | Binary - Decimal Conversion | HLR1 |
| LLR15 | Decimal - Binary Conversion | HLR1 |
| LLR16 | Decimal - Octal Conversion | HLR1 |
| LLR17 | Octal - Decimal Conversion | HLR1 |
| LLR18 | Length unit Conversion | HLR1 |
| LLR19 | Temperature unit Conversion | HLR1 |
| LLR20 | Current AC-DC Conversion | HLR1 |
| LLR21 | Log | HLR1 |
| LLR22 | Exponential | HLR1 |
| LLR23 | Modulus (remainder) | HLR1 |
| LLR24 | Factorial | HLR1 |
| LLR25 | Square root | HLR1 |
| LLR26 | Cube root | HLR1 |
| LLR27 | LCM | HLR1 |
| LLR28 | GCD | HLR1 |
| LLR29 | Permutation | HLR1 |
| LLR30 | Combination | HLR1 |

# Design

# High Level Design

# **OBJECT DIAGRAM FOR HIGH LEVEL REQUIREMENTS**

# 

# **CLASS DIAGRAM FOR HLR**

# 

# **ACTIVITY DIAGRAM for HLR**

# HLR_Activity

# **SEQUENCE DIAGRAM FOR HIGH LEVEL REQUIREMENTS**

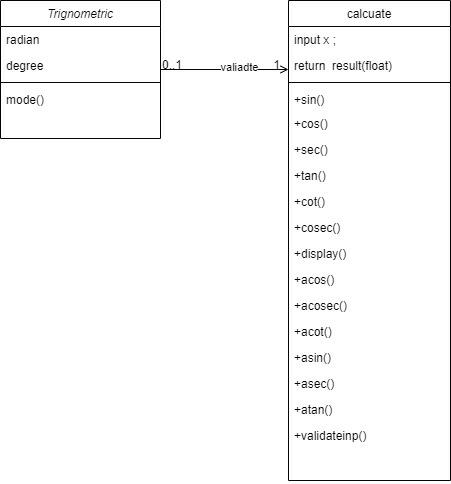
# HLR_sequence_dia

# Low Level Design

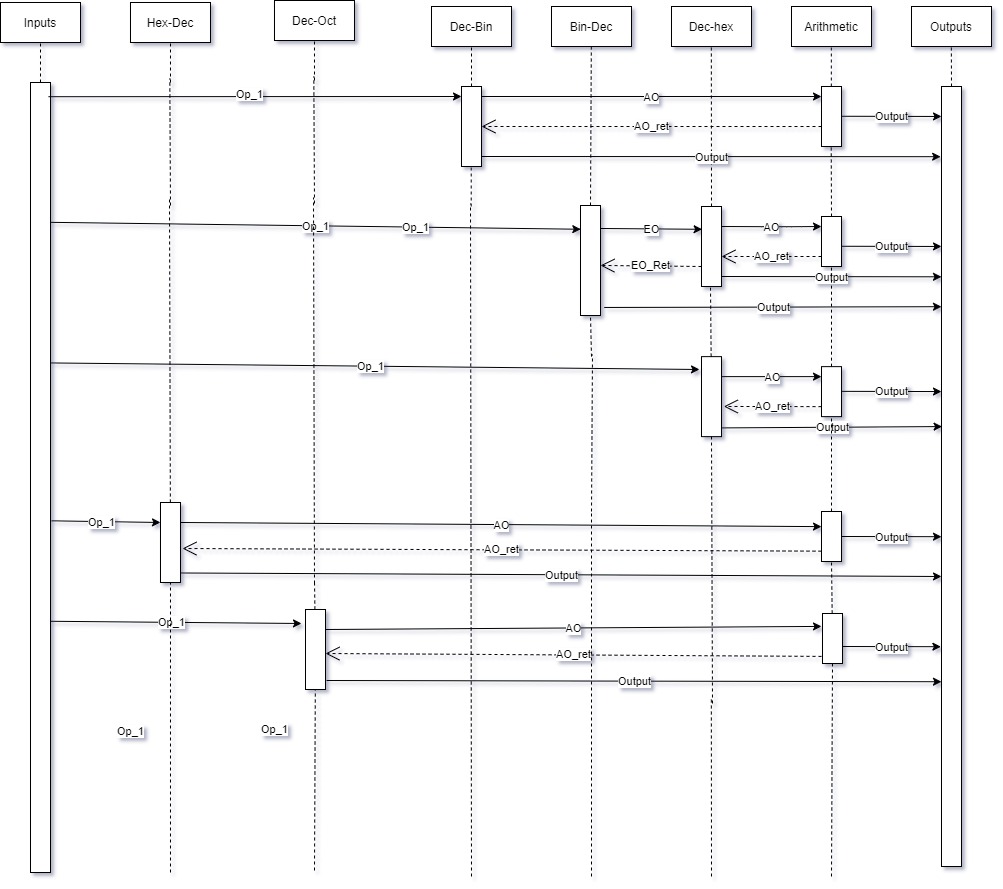
# **ACTIVITY DIAGRAM for TRIGNOMETRIC**

# B_LLR_Trigno

# **CLASS DIAGRAM FOR TRIGNOMETRIC**



# **CLASS DIAGRAM FOR CONVERSION**LLR_class_diagram

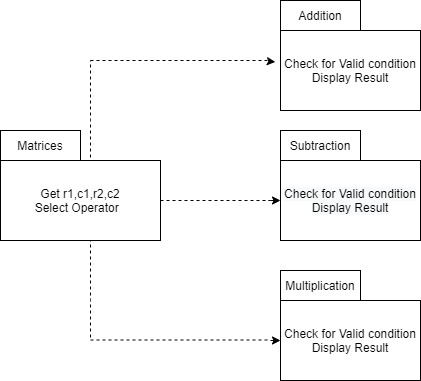
**SEQUENCE DIAGRAM FOR CONVERSI**

**USE CASE DIAGRAM FOR BASIC**

# **OBJECT DIAGRAM FOR BASIC**

# llr-basic-object

# **PACKAGE DIAGRAM FOR MATRIX OPERATIONS**



# **ACTIVITY DIAGRAM FOR MATRIX OPERATION**Matrices_activity

### Git Link:

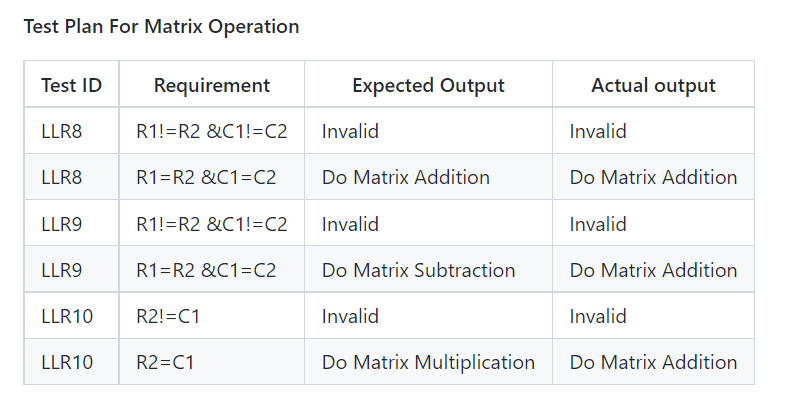
Link: <https://github.com/GENESIS2021Q1/Applied_SDLC-Dec_Team_47>

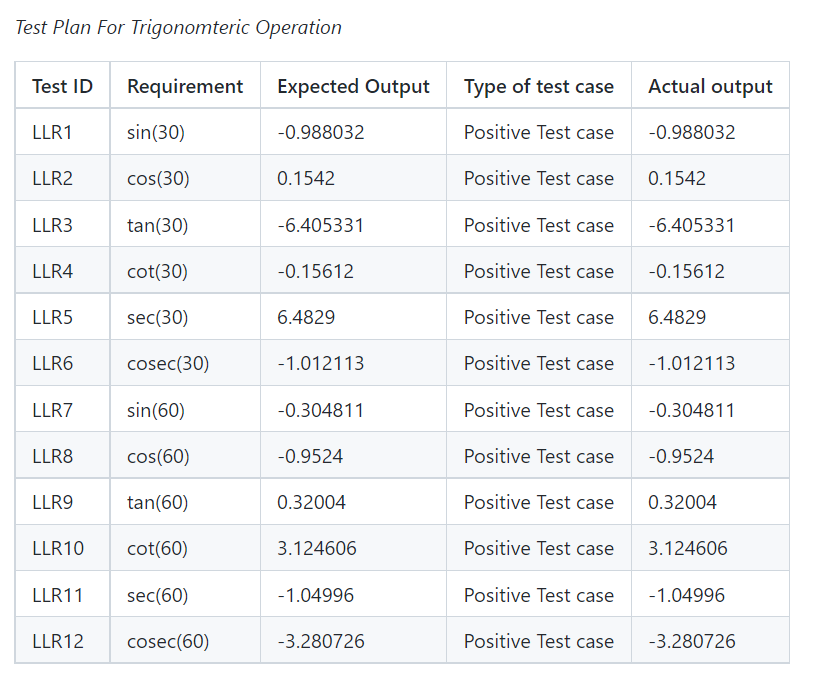
### Individual Contribution and Highlights

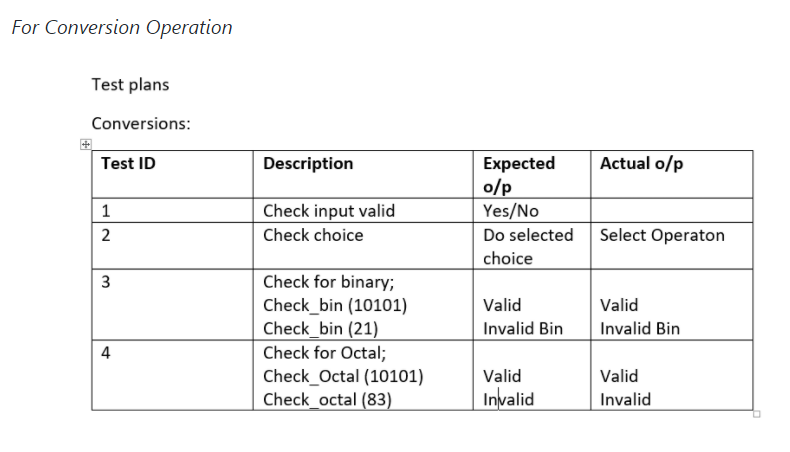
* Requirements
* Implementation (Add Order in System)
* Created Unity File

# TEST PLAN:

# **Test Plan For Matrix Operation**

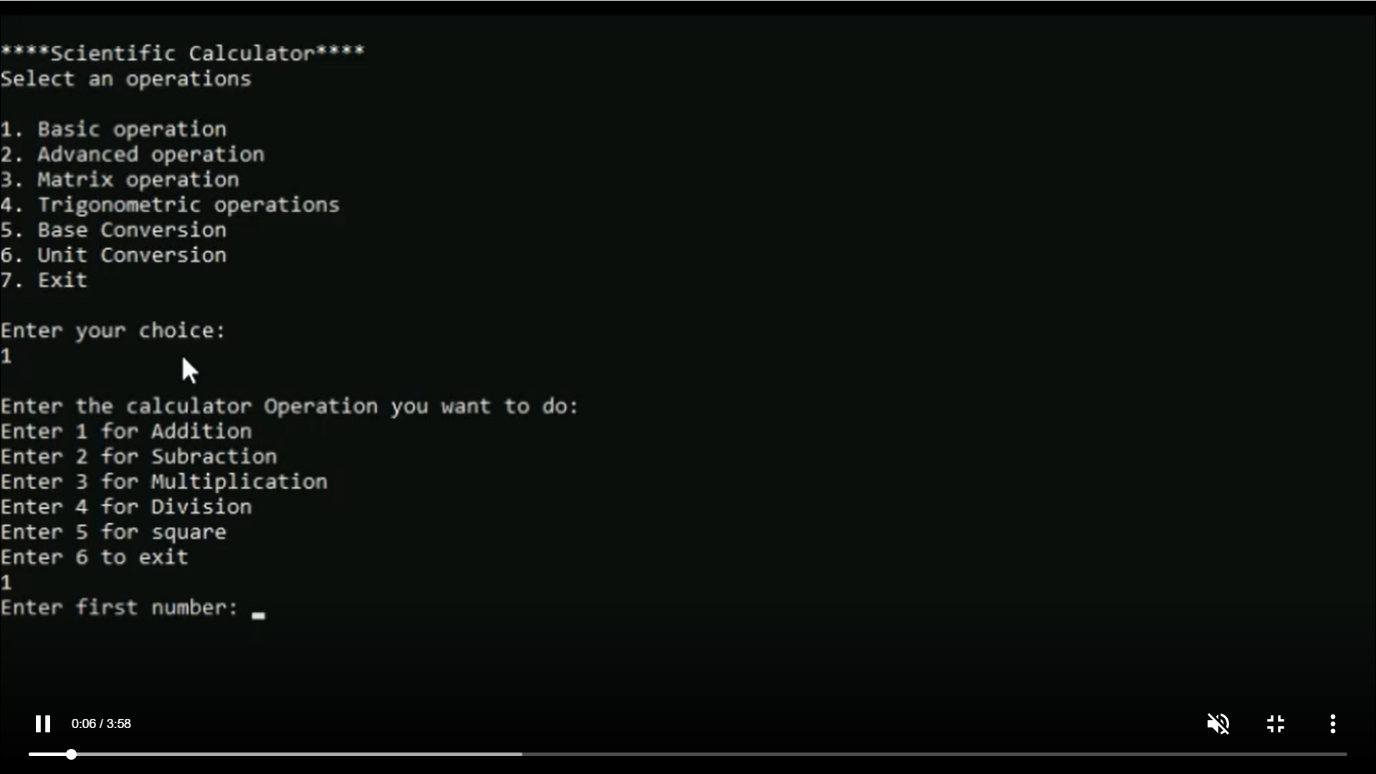






# 

# **6\_Images\_And\_Videos**



# 

# 

# 

## Challenges Faced and How Was It Overcome

1. Running the make file as its resolved by defining its correct path(.out for linux and -lm for math functions)
2. Syncronizing the VS code to github , colleague help to resolve the issue
3. Making the function call in correct path
4. Open git log while committing, thus went to github desktop and pulled origin and then pushed origin.
5. Test case code for the boundary problem. Added code with the help of internet

## Learning Resources

1. [markdownCheatsheet](https://github.com/adam-p/markdown-here/wiki/Markdown-Cheatsheet)
2. [markdownBasics](https://guides.github.com/features/mastering-markdown/)
3. [git inspector](https://github.com/ejwa/gitinspector.git)
4. [github workflow](https://docs.github.com/en/actions/learn-github-action)

# 

# 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 FORD [Team]

## Module: - Applied Model Based Design Module

**Individual Topic: - Boot Control System**

Name: Ragini Sandip Bhandare.

PS No. 40021048

Team Name: Team Ford

Module: Body Control Module

Feature: Boot System

**Boot System**

**Introduction**

When a vehicle is booted, a device is attached to a wheel of your vehicle to prevent you from moving it. Your vehicle can be booted for owing the city more than $350 in parking ticket or camera violations that are in judgment.

The trunk (North American English) or boot (British English) of a car is the vehicle's main storage or cargo compartment, often a hatch at the rear of the vehicle. It is also called a tailgate.

In Indian English the storage area is known as a dickey (also spelled dicky or diggy), and in South-East Asia as a compartment.

A wheel clamp, also known as wheel boot, parking boot, or Denver boot,[1][2] is a device that is designed to prevent motor vehicles from being moved. In its most common form, it consists of a clamp that surrounds a vehicle wheel, designed to prevent removal of both itself and the wheel.

**Overview**

• Cold boot: The device boots from a state in which the processor and peripherals are completely powered off.

• Warm boot: The device boots from a state in which the processor or peripherals still have power or some form of residual information

Multi-stage booting

Multi-stage booting allows a telematics or infotainment system to meet device-activation times after a cold boot. It consists of dividing the software boot process into multiple device-activation stages so that the fastest path to each stage can be achieved.

Beyond carrying luggage, the trunk of most passenger vehicles commonly contains various other components often behind the trimmed surfaces of the interior. These components may be accessed by the customer or the service personnel through (in some cases lockable) hatches in the trim, or by removing carpet and support boards etc.

a. The occupant should not be thrown out of the seat excessively during the crash.

b. Parts of the seat system shall not hurt the occupant before, during, or after a vehicle crash.

|  |  |
| --- | --- |
| ID | Description |
| HLR1 | Receive signals from the user, and adjust the door. |
| HLR2 | All conditions should pass. One condition should pass at a time. |
| HLR3 | Run a system diagnostic test sequence at ignition and determine if any errors are present in the system. |
| HLR | Set the door open or close as per user requirement. |

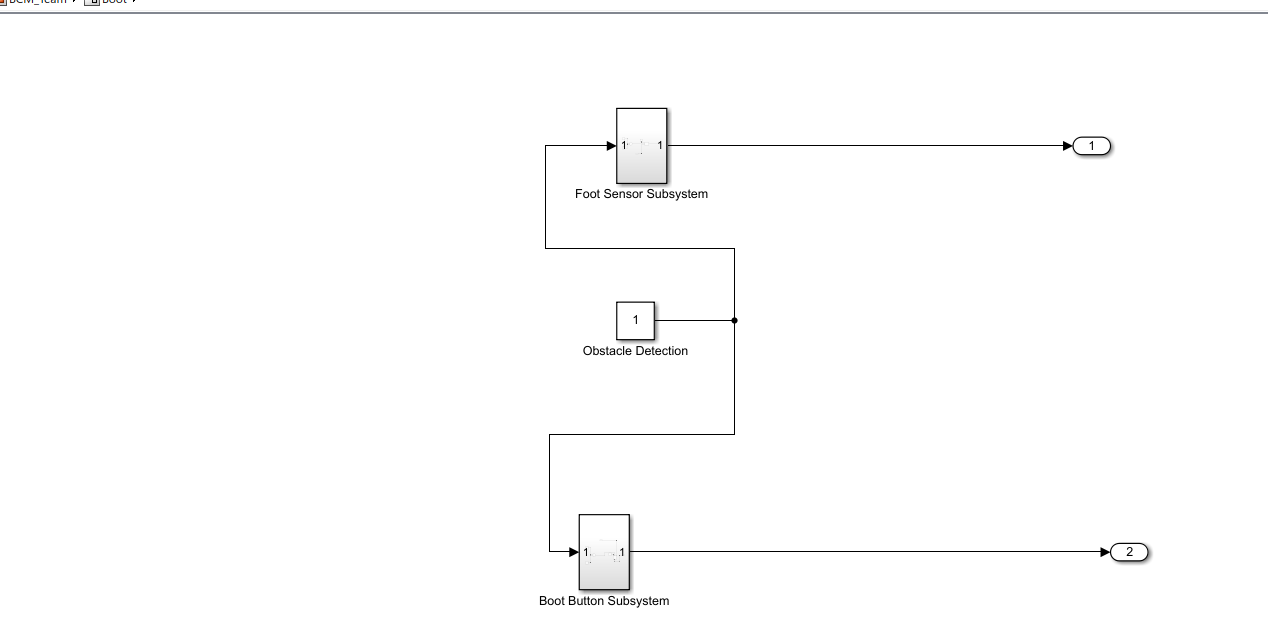
**High Level Requirements:-**

## Low Level Requirements:-

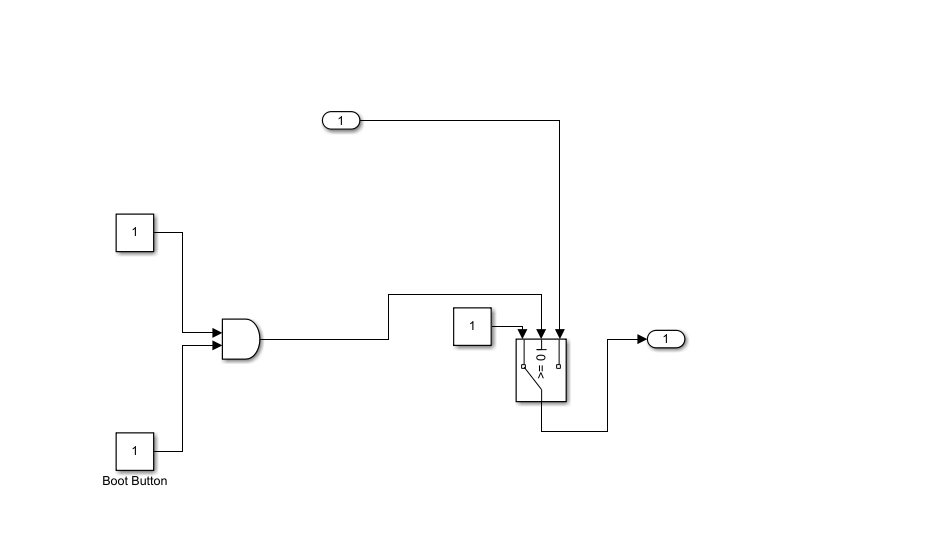
|  |  |
| --- | --- |
| ID | Description |
| LLR1 | Put the input value. |
| LLR2 | The system test will engage when the car is turned on. |
| LLR3 | Compare the conditions. |

**MODELING**

Boot System



Boot Button Subsystem

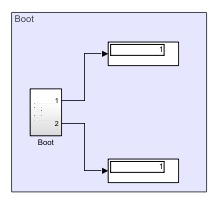


Foot Sensor System



**Output**

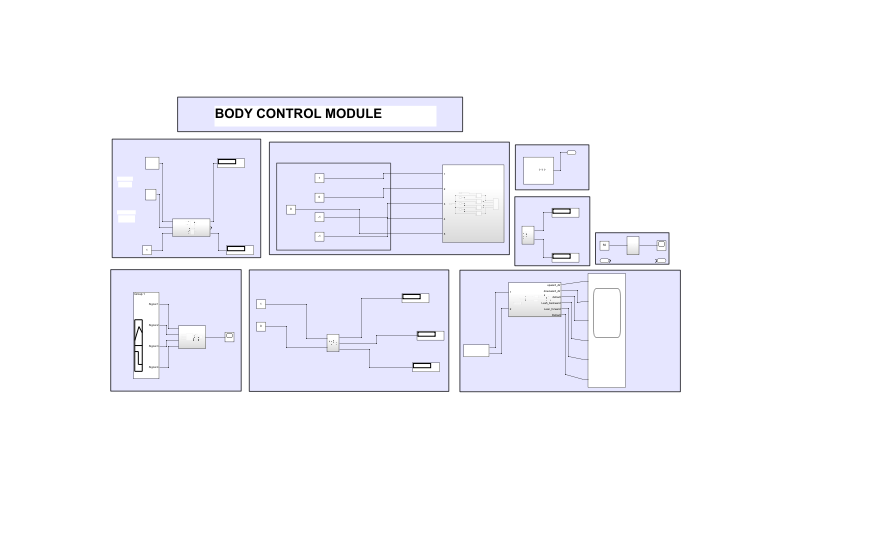
### Running the Simulation in Seat control Mode



**Conclusion**

This model shows how you can use Simulink to simulate a boot system. 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 do this by generating real-time C code for this model using the Simulink Coder. You can then test an actual boot controller by interfacing it to the real-time hardware, which runs the generated code. In this scenario, the real-time model would send the user input to the controller, and the controller would adjust it according to the conditions set to the model.

**Team Activity**



# Mini project 6 – Wiper Control[Team]

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

## WIPER CONTROL SYSTEM

# Wiper Control System using STM32

## Introduction

In this project, Wiper Control System is designed using STM32 Microcontroller.The STM32 family of 32-bit microcontrollers based on the Arm Cortex -M processor is designed to offer new degrees of freedom to MCU users. It offers products combining very high performance, real-time capabilities, digital signal processing, low-power / low-voltage operation, and connectivity, while maintaining full integration and ease of development.

## Objective

The main objective of the system is to create a wiper control system with low power consumption and high performance, which can be achieved by STM 32 Microcontroller.

## Components used

1. STM 32
2. Four LED
3. Switch

## Research

The STM32 family of 32-bit microcontrollers based on the Arm Cortex M processor is designed to offer new degrees of freedom to MCU users. It offers products combining very high performance, real-time capabilities, digital signal processing, low-power / low-voltage operation, and connectivity, while maintaining full integration and ease of development.

The unparalleled range of STM32 microcontrollers, based on an industry-standard core, comes with a vast choice of tools and software to support project development, making this family of products ideal for both small projects and end-to-end platforms.

## Features

1. Low power Consumption
2. High performance
3. real-time capabilities

## 4W's and 1 H's

### Why

1. To understand basic concepts in STM32
2. To control wiper system by switching LED in STM32

### Where

1. It can be used for many mini projects.
2. Its has too many realistic features in this STM32 microcontroller

### Who

1.It can be used by students. 2.It can be used by anyone who are new to embedded programming language.

### When

1. It can be used for both small projects and end-to-end platforms. 2.It is easy to access the emmbedded programing language.

### How

1.By using softwares to exceute the program. 2.By loading the program in STM32 and execute.

## SWOT Analysis

### Strengths

1.Simple program to understand. 2.time effective.

### Weakness

1. Single operating program is used here

### Opportunities

1.Program can be made more complex by adding more components.

### Threats

1.There are advanced programs which are simple to learn is out already.

## High Level Requirements

| **Id** | **Description** | **Status** |
| --- | --- | --- |
| HLR\_1 | Microcontroller | Implemented |
| HlR\_2 | Swtich | Implemented |
| HLR\_3 | Four LED | Implemented |
| HLR\_4 | Software | Implemented |

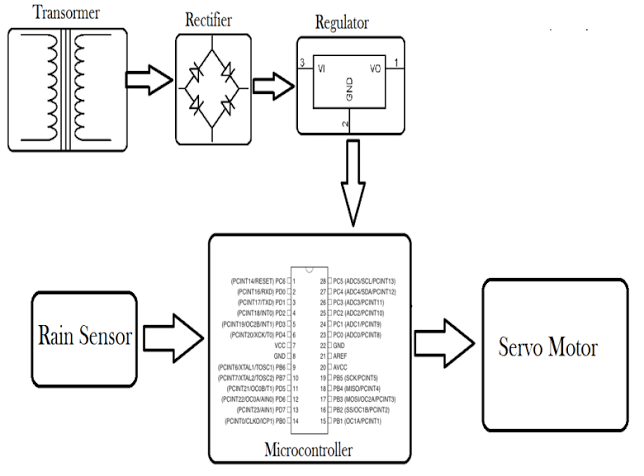
## Low Level Requirements

| **Id** | **Description** | **Status** |
| --- | --- | --- |
| LLR\_1 | STM32 | Implemented |
| LLR\_2 | Swtich | Implemented |
| LLR\_3 | Four LED | Implemented |

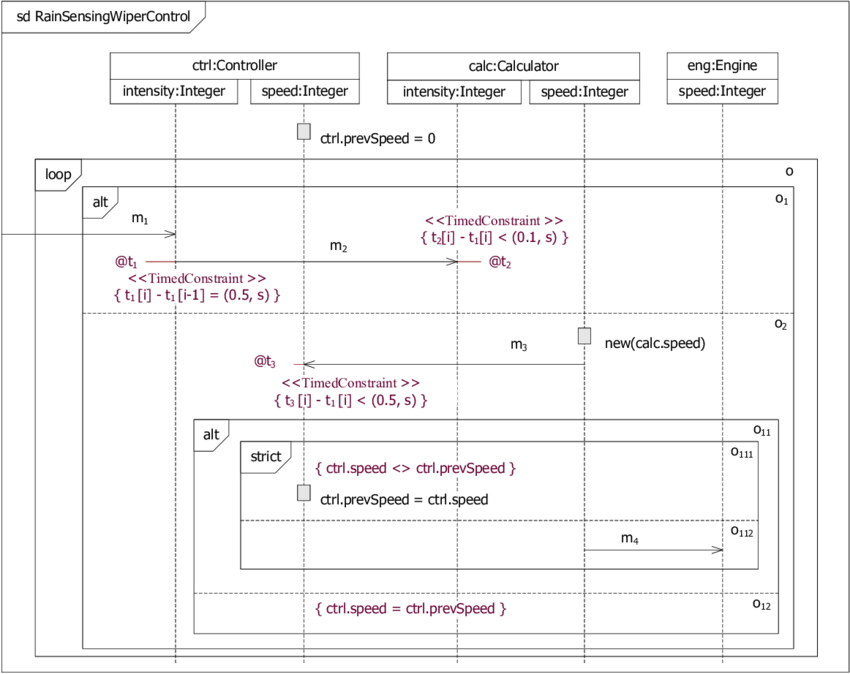
## FLOWCHART

[](https://user-images.githubusercontent.com/78525873/148489749-111fe8ab-d88e-43f7-b556-b6c365690169.png)

## BLOCK DIAGRAM

[](https://user-images.githubusercontent.com/78525873/148491772-9e992aae-dcc8-4680-8158-59bdd11f3b3e.png)

## UML DIAGRAM

[](https://user-images.githubusercontent.com/78525873/148493232-6562f599-46ac-4b3b-abb0-32e0e56b9452.png)

## High Level Requirements

| **Id** | **Description** | **Status** |
| --- | --- | --- |
| HLR\_1 | Microcontroller | Implemented |
| HlR\_2 | Swtich | Implemented |
| HLR\_3 | Four LED | Implemented |
| HLR\_4 | Software | Implemented |

## Low Level Requirements

| **Id** | **Description** | **Status** |
| --- | --- | --- |
| LLR\_1 | STM32 | Implemented |
| LLR\_2 | Swtich | Implemented |
| LLR\_3 | Four LED | Implemented |

## Implementation and Summary

### Git Link:

Link: https://github.com/GENESIS-2022/MasteringMCU-Team76.git

### 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**: - **Power Mirror**

# Power Mirror

## Introduction

A power side-view mirror (power side mirror, power wing mirror, or simply power mirror) is a side-view mirror equipped with electrical means for vertical and horizontal adjustment from the inside of the automobile.

](https://user-images.githubusercontent.com/83066731/150640170-5656010f-927b-4238-bcb2-7ccad1a73e55.jpg)

## 4W's and 1H

**What:** A power side-view mirror (power side mirror, power wing mirror, or simply power mirror) is a side-view mirror equipped with electrical means for vertical and horizontal adjustment from the inside of the automobile.

**Where:** It allows you to connect and mirror your phone to PC via USB or WiFi, and you can remotely control Android from PC or Mac using your mouse and keyboard.

**Why:** A side-view mirror (or side mirror), also known as a wing mirror, is a mirror placed on the exterior of motor vehicles for the purposes of helping the driver see areas behind and to the sides of the vehicle, outside the driver's peripheral vision (in the "blind spot").

**When:** Power windows were first introduced by Ford Motors in 1941. The first cars to get the power windows were the Lincoln Custom and the Packard Custom Super 180.

\*\*How:\*\*Each rear view mirror has two DC motors. ... Electricity then flows through the switch to the DC motor and the mirror head moves in the intended direction. If you press the same switch to the opposite direction, you are reversing the electricity to the mirror motor and the mirror moves in the opposite direction

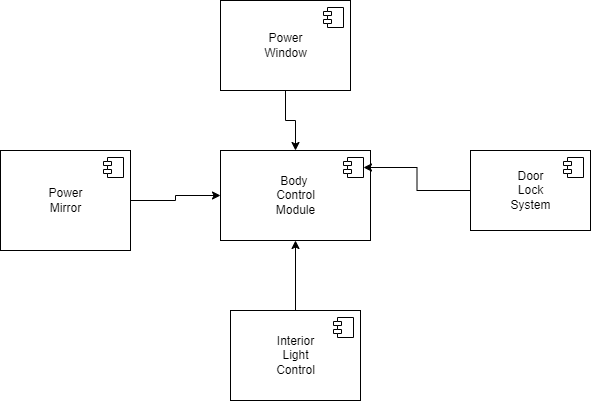
## POWER MIRROR

| **High level Requirement** | **Description** |
| --- | --- |
| HLR\_1 | Power folding mirrors automatically fold inwards while the car is parked to avoid accidental damage |
| HLR\_2 | The mirrors will fold back out when the car door is opened |
| HLR\_3 | There will be a clicking sound when the mirrors are resynchronising |
| HLR\_4 | Due to the high beam of electrons the glass of a power mirror may also be electrically heated to prevent from fogging and icing |

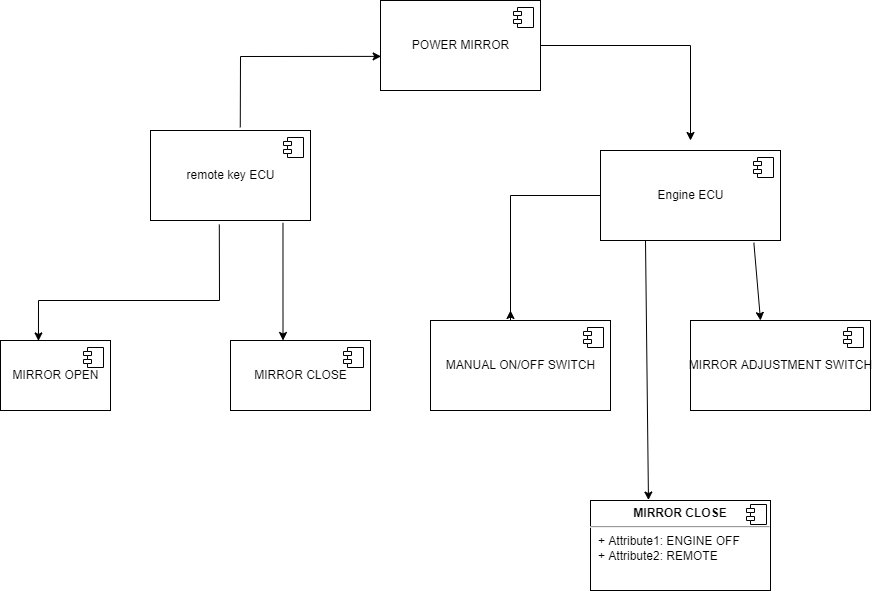
| **Low level Requirement** | **Description** |
| --- | --- |
| LLR\_1 | Driver can manually adjust the switch to change the position of the mirror |
| LLR\_2 | Adjust the mirror direction according to the user requirement |
| LLR\_3 | When the engine is on the mirrors has to automatically open |

## Design

# Body Control Module

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

## Power Mirror

[](https://user-images.githubusercontent.com/83066731/150649199-a86bbd29-673d-49d4-b6d0-30690f29e9e3.png)

## Implementation and Summary

### Git Link:

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

### Individual Contribution and Highlights

* 1. Power Mirror 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, The electric grid cannot support growth in electric vehicles

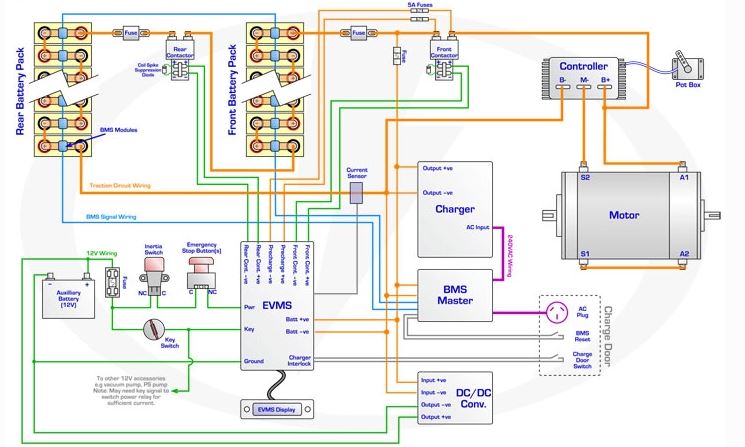
# Research

 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/raginibhandare/Team11\_EV\_TRUCK.git

# Mini project 9 – Parking System [Individual]

## Module:- Classic Autosar Basic to Intermediate

# Power Mirror

## Introduction

A power side-view mirror (power side mirror, power wing mirror, or simply power mirror) is a side-view mirror equipped with electrical means for vertical and horizontal adjustment from the inside of the automobile.

](https://user-images.githubusercontent.com/83066731/150640170-5656010f-927b-4238-bcb2-7ccad1a73e55.jpg)

## 4W's and 1H

**What:** A power side-view mirror (power side mirror, power wing mirror, or simply power mirror) is a side-view mirror equipped with electrical means for vertical and horizontal adjustment from the inside of the automobile.

**Where:** It allows you to connect and mirror your phone to PC via USB or WiFi, and you can remotely control Android from PC or Mac using your mouse and keyboard.

**Why:** A side-view mirror (or side mirror), also known as a wing mirror, is a mirror placed on the exterior of motor vehicles for the purposes of helping the driver see areas behind and to the sides of the vehicle, outside the driver's peripheral vision (in the "blind spot").

**When:** Power windows were first introduced by Ford Motors in 1941. The first cars to get the power windows were the Lincoln Custom and the Packard Custom Super 180.

\*\*How:\*\*Each rear view mirror has two DC motors. ... Electricity then flows through the switch to the DC motor and the mirror head moves in the intended direction. If you press the same switch to the opposite direction, you are reversing the electricity to the mirror motor and the mirror moves in the opposite direction

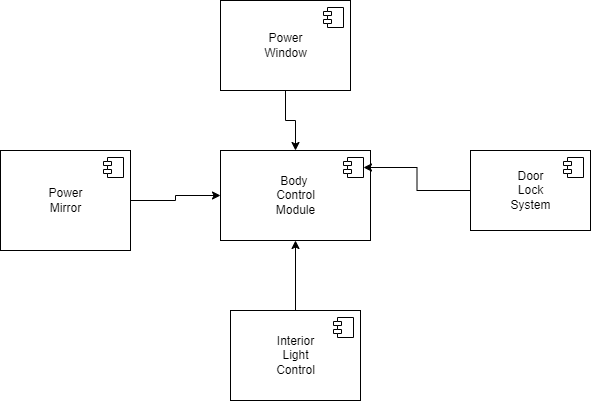
## POWER MIRROR

| **High level Requirement** | **Description** |
| --- | --- |
| HLR\_1 | Power folding mirrors automatically fold inwards while the car is parked to avoid accidental damage |
| HLR\_2 | The mirrors will fold back out when the car door is opened |
| HLR\_3 | There will be a clicking sound when the mirrors are resynchronising |
| HLR\_4 | Due to the high beam of electrons the glass of a power mirror may also be electrically heated to prevent from fogging and icing |

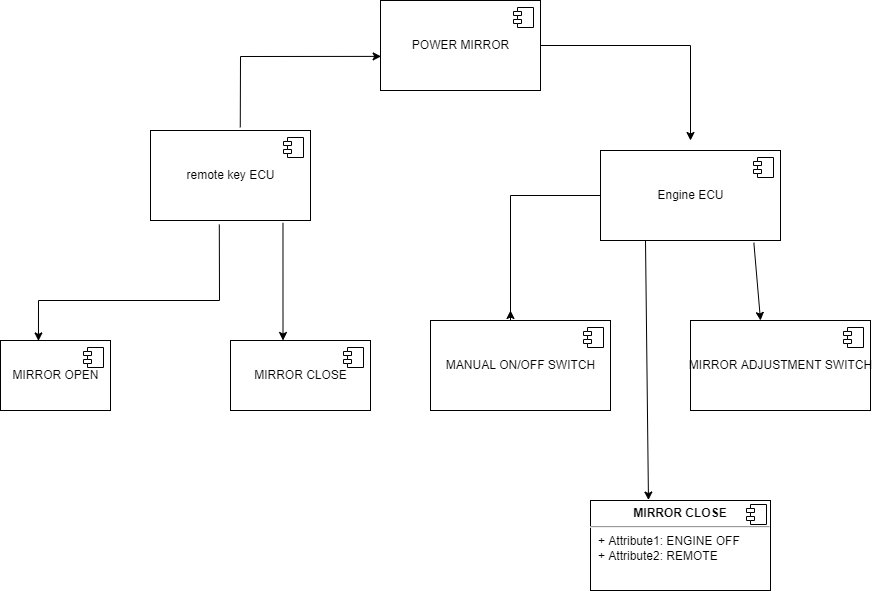
| **Low level Requirement** | **Description** |
| --- | --- |
| LLR\_1 | Driver can manually adjust the switch to change the position of the mirror |
| LLR\_2 | Adjust the mirror direction according to the user requirement |
| LLR\_3 | When the engine is on the mirrors has to automatically open |

## Design

# Body Control Module

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

## Power Mirror

[](https://user-images.githubusercontent.com/83066731/150649199-a86bbd29-673d-49d4-b6d0-30690f29e9e3.png)

## 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