Details

Ver. Rel. No.	Release Date	Prepared By	Reviewed By	To Be Approved	Remarks/Revision Details
1.0	16/02/2022	Aarti Bhagat 40021161			

Contents

List of Figures

Miniproject – 1: Employee Record System [Individual]

Modules:

1. C Programming
2. Git

Requirements

4W's and 1 H's

What:

Employee record system is to add, record, modify and delete the information about employee in the organization. The purposed of the system is to save time, To reduce paper and file works, To speed up the procedure, To have a flexible system that can adjust changes in future, To make the decision making process easy for management.

Who:

This system will help the organization to build record of the employee's.

Where:

The system can be accessed easily by organization anywhere anytime at their own comfort.

When:

The user can access it anytime , anywhere.

How:

By giving simple information it will record all that data in the system.

High Level Requirements

ID	Description	Status
HLR_1	It records the basic information of the employee	Implemented
HLR_2	It can display all records of employee	Implemented
HLR_3	It can add delete and modify the basic information of employee	Implemented

Low Level Requirements

ID	Description	Status
LLR_1	Basic information of employee displayed	Implemented
LLR_2	Input from the user	Implemented
LLR_3	Exit the program	Implemented

Design

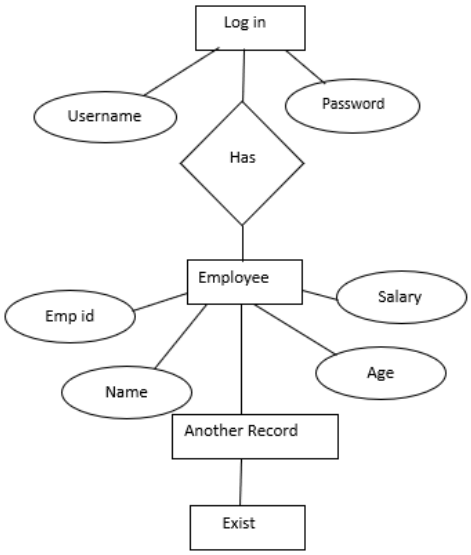


Figure 1 Behavior Diagram

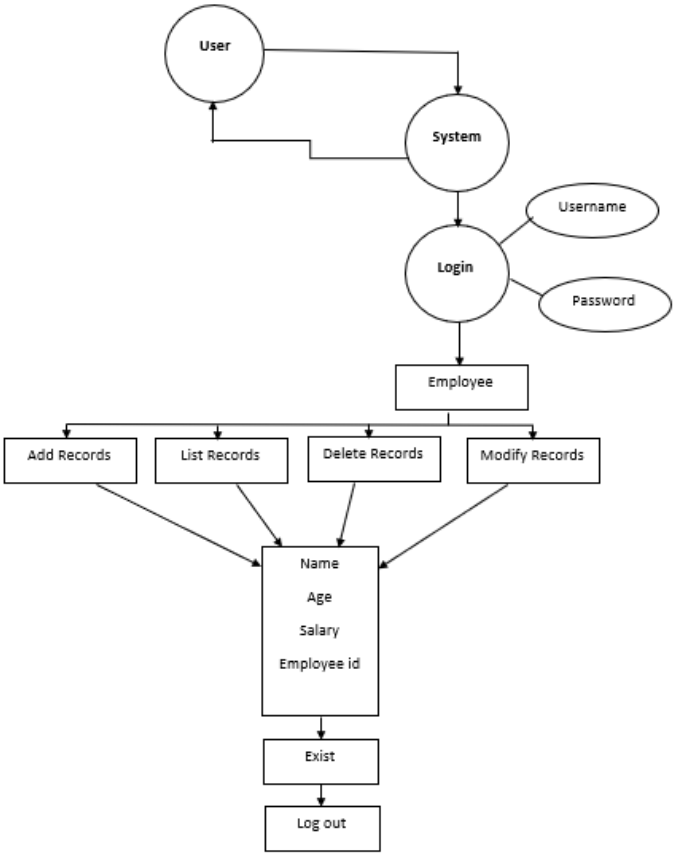


Figure 2 Structure Diagram

Test Plan

High Level Test Plan

Test ID	Description	Exp I/P	Exp O/P	Actual O/P	Type Of Test
H_01	check all information related employee put correctly	Id no	record of employee	record Displayed	Requirement Based
H_02	Check whether same employee record added again	Id no	record of employee	record displayed	Requirement Based

Low Level Test Plan

Test ID	Description	Exp I/P	Exp O/P	Actual O/P	Type Of Test
L_01	Check whether it adding, modifying, or deleting the record of employee	Id no	record of employee	records displayed	Scenario Based
L_02	Check whether there mistakes in records	Id no	record of employee	record displayed	Scenario Based
L_03	Check all employee have a different employee id	Id no	record of employee	record displayed	Requirement Based

Implementation and Summary

Git Link:

Link: https://github.com/aartibhagat/M1_Project_Employee_record_system

Git Dashboard

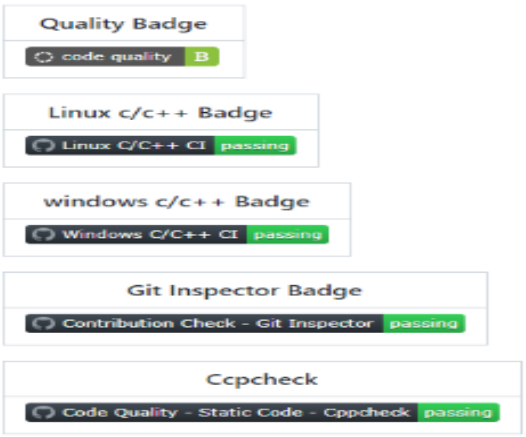


Figure 3 Git Dashboard

Miniproject 2 – Temperature Measurement using Atmega328 [Individual]

Modules

1. C Programming
2. Embedded System
3. SimulIDE
4. Git

Requirements

4W's and 1 H's

Why: To maintain the heat in the vehicles in cold weather.

What: Temperature measurement system to measure, control and generate heat.

Where: Automotive Industry.

When: In vehicles at low temperature areas.

How: By installing the system in vehicles.

High Level Requirements

ID	Description
HLR1	When the two switches are closed, the first LED glows indicating the actuation of the system and the heater.
HLR2	Analog input from the temperature sensor
HLR3	Display.

Low Level Requirements

ID	Description
LLR1	ADC with Pulse Width Modulation.
LLR2	Compatible on different Operating Systems.

Design

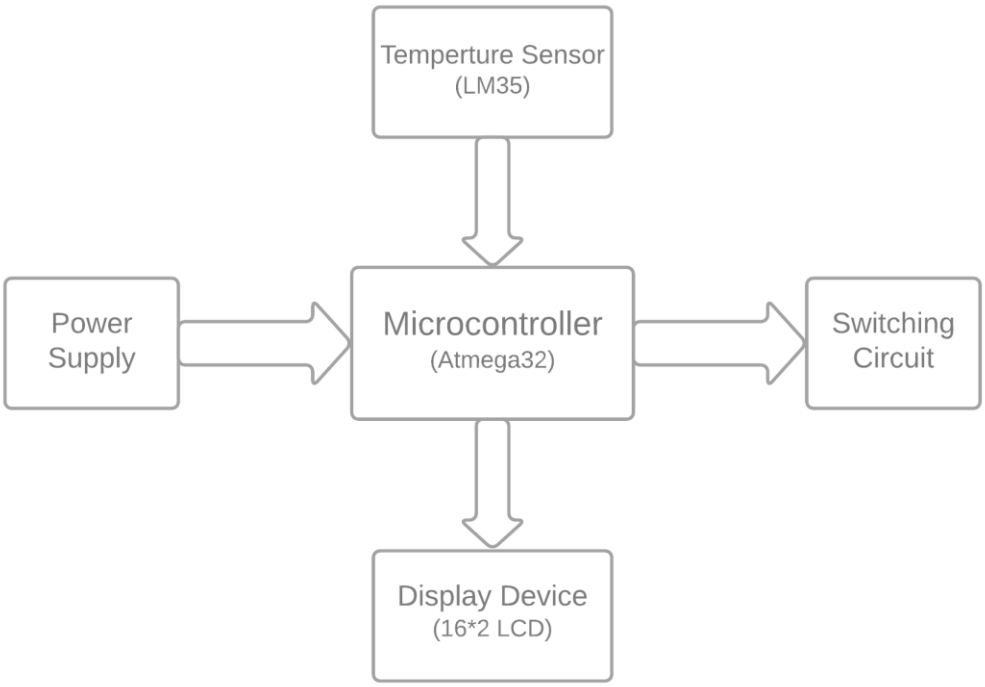


Figure 4 Behavior Diagram

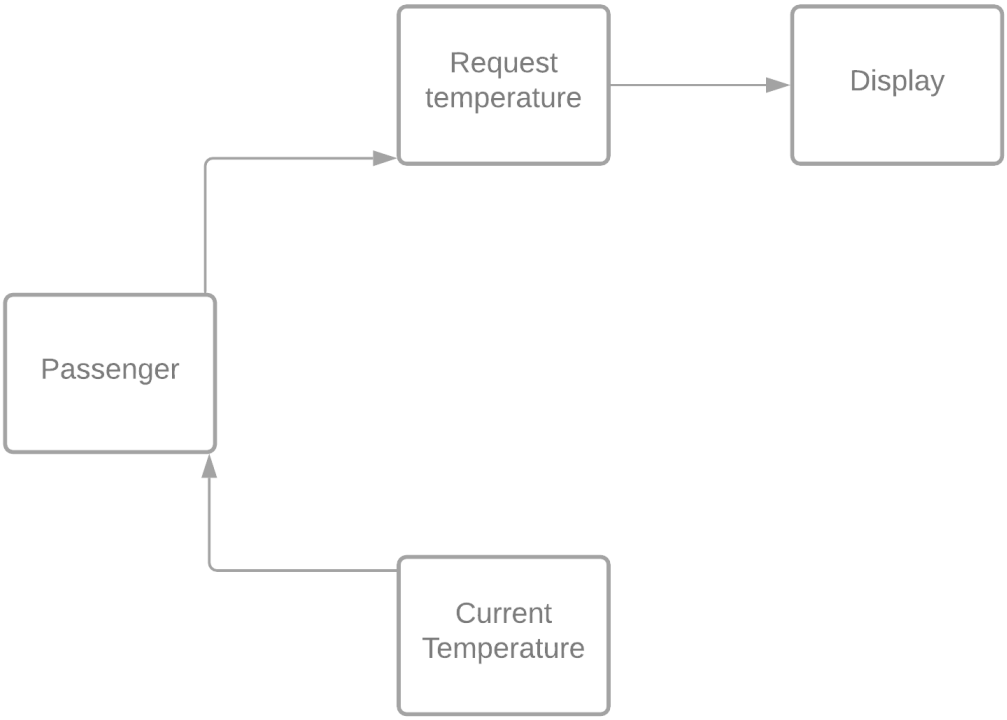


Figure 5 Structure Diagram

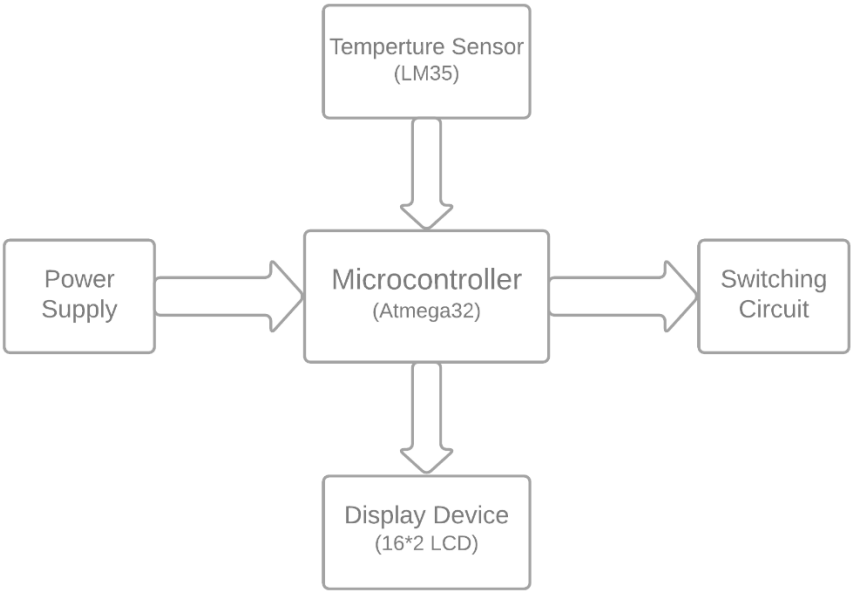


Figure 6 Block Diagram

TEST PLAN:

High Level Test plan:

ID	Description	Expected O/P	Actual O/P
H_01	Read temperature	PASSED	SUCCESS
H_02	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_02	Reserve seat	PASSED	SUCCESS

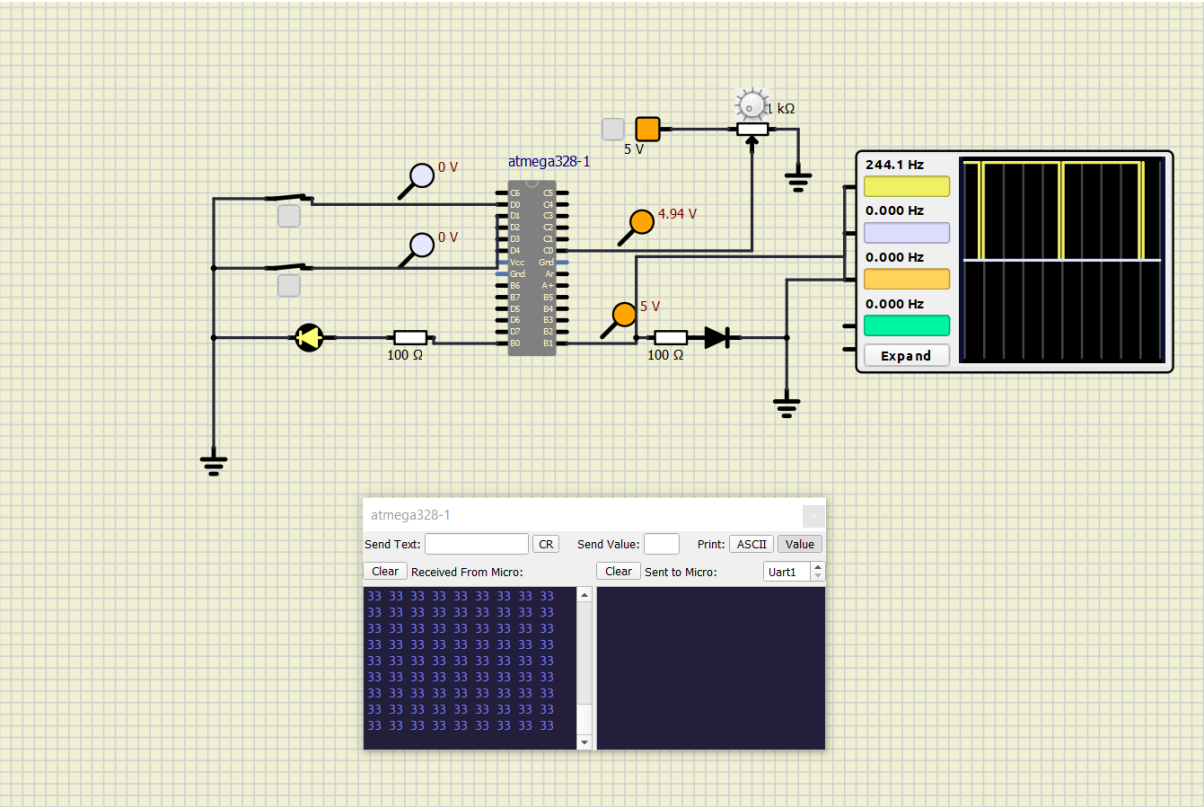


Figure 7 Simulation

Implementation and Summary

Git Link:

Link: https://github.com/aartibhagat/M2-Embedded_Temperature-measurement-using-Atmega328

Git Dashboard

code quality B

Figure 8 Git Dashboard

Miniproject 3 – Virtual Costume Advisor [Team]

Modules

1. SDLC
2. Git

Requirements

4W's and 1 H's

Who:

People who want to look good by getting targeted outfit ideas for their body shape.

What:

Calculates the body shape and occasion they are addressing then suggests them the best suitable outfit for their body.

When:

Anytime they want to get themselves dressed well for particular occasions.

Where:

In the Application/system which has this program.

How:

By entering the measurements of the individuals bust size, waist size, high hip size, hip size.

High Level Requirements

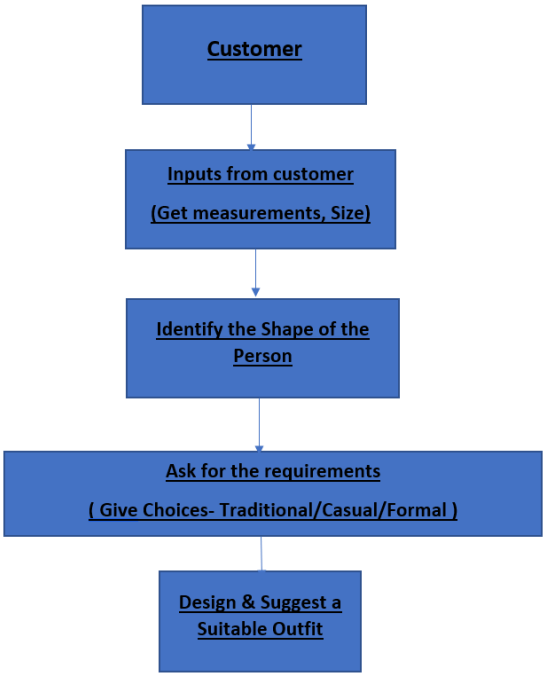
ID	Description	Status
HLR_1	Getting the measurements from the user	Application
HLR_2	Calculating the body type	Vs code
HLR_3	Getting the choice of outfit type from the user	Application
HLR_4	Getting the choice of listed costume from the user	Application

Low Level Requirements

ID	Description	Platform
LLR_1	The measurements should be properly taken and entered correctly by the user	Application
LLR_2	Coding formula to calculate body type should be accurate	Vs Code
LLR_3	The Choice of outfit type should be properly Chosen and entered correctly by the user	Application
LLR_4	The Choice of costume should be properly taken and given correctly by the user	Application

Design

Behavioural Diagram Low Level



Behavioural Diagram High Level

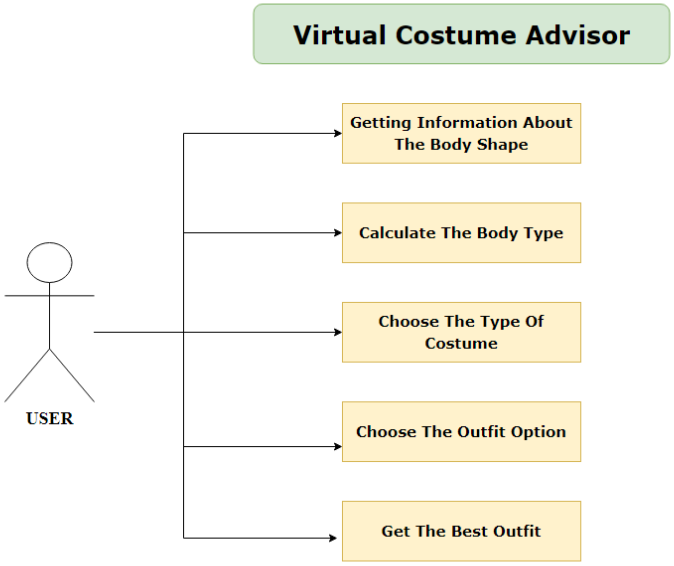
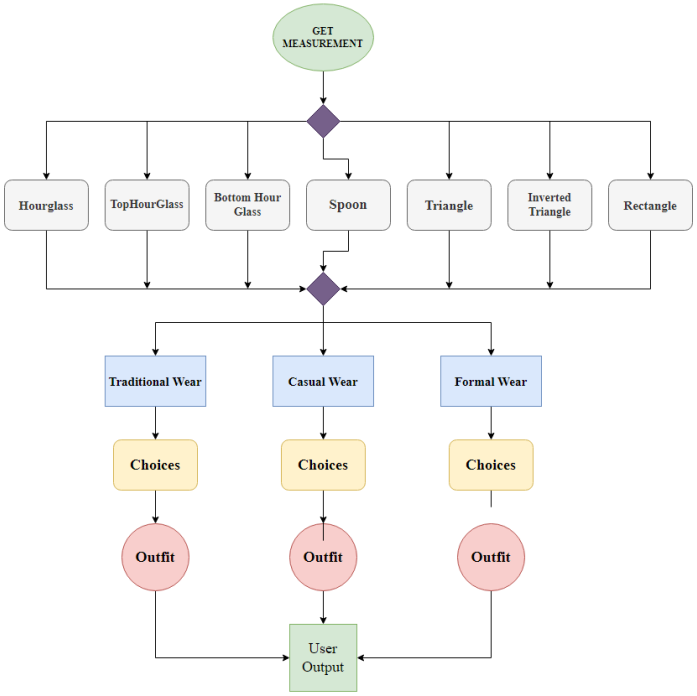


Figure 9 Behavior Diagram

Structural Diagram Of High Level



Structural Diagram Of Low Level

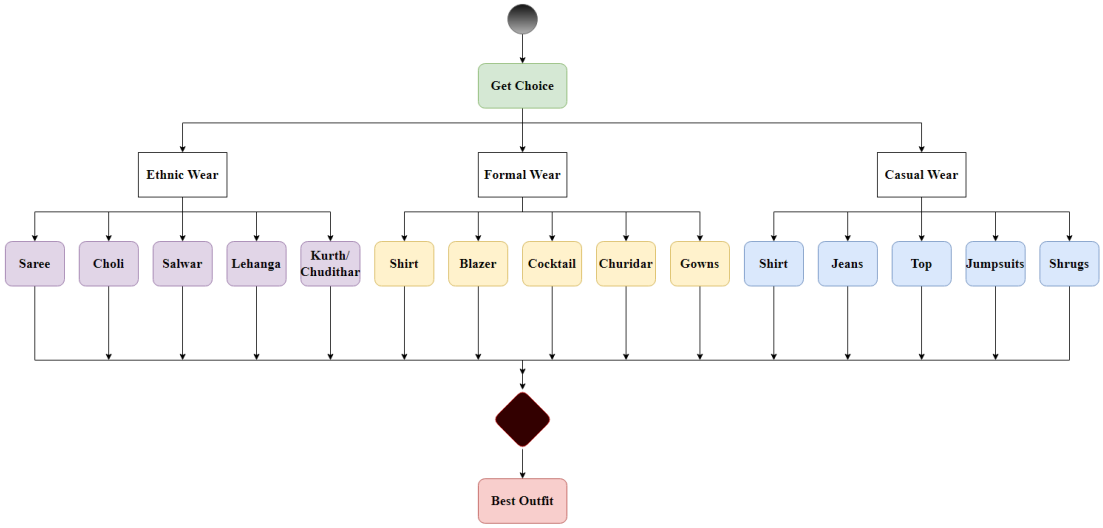


Figure 10 Structure Diagram

Test Plan

High Level Test Plan

ID	Description	Expected I/P	Expected O/P	Actual O/P	Type Of Test
HLTP_1	Create NFT	Click	SUCCESS	SUCCESS	Requirement Based
HLTP_2	Sell NFT	Click	SUCCESS	SUCCESS	Requirement Based
HLTP_3	Bid NFT	Click	SUCCESS	SUCCESS	Requirement Based
HLTP_4	Buy NFT	Click	SUCCESS	SUCCESS	Requirement Based
HLTP_5	Contact	Click	SUCCESS	SUCCESS	Requirement Based

ID	Description	Expected I/P	Expected O/P	Actual O/P	Type Of Test
HLTP_6	Sign In	Click	SUCCESS	SUCCESS	Requirement Based
Test ID	**Description**	**Exp I/P**	**Exp O/P**	**Type of Test**	**Test ID**

Low Level Test Plan

ID	Description	Expected I/P	Expected O/P	Actual O/P	Type Of Test
LLTP_1	Connect Wallet	Click	SUCCESS	SUCCESS	Requirement Based
LLTP_2	Activity	Click	SUCCESS	SUCCESS	Requirement Based
LLTP_3	Forgot password	Click	SUCCESS	SUCCESS	Requirement Based
LLTP_4	To check whether none of the fields should be empty	Empty value in the input module	Prompt message mandatory field missing	SUCCESS	Requirement Based
LLTP_5	E-mail ID should be in the perfect format i.e. group2@gmail.com	group2@gmail.com	Prompt message invalid E-mail ID	SUCCESS	Requirement Based

Implementation and Summary

Git Link:

Link: https://github.com/GENESIS2021Q1/Applied_SDLC-Dec_Team_50

Individual Contribution and Highlights

Summary

1. requirements contents, research on the topic,
2. added the test plan, HLR and LLR of project
3. Help to do code

Miniproject 4 – Calendar Automation[Team]

Modules

1. Python
2. Git

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

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
LR11	Master Calendar: Course title correction	HR02	Not Available	Not Available

ID	Feature	High Level ID	MATLAB v0 Status	Python v0 Status
LR12	Master Calendar: display the dates that were not analysed	HR02	Implemented	Not Available
LR12	Faculty Calendar: display Month wise	HR03	Implemented	Implemented
LR13	Faculty Calendar: display Initiative wise	HR03	Implemented	Not Available
LR14	Faculty Calendar: Appending	HR03	Implemented	Not Available
LR15	Faculty Calendar: Differentiate Initiatives (Color Codes/Numbers)	HR03	Implemented	Implemented
LR16	Faculty name correction/validation in faculty calendar	HR03	Not Available	Not Available
LR17	Faculty Calendar: Highlight conflicts (Red highlight/pop-up/Concatenated Numbers)	HR03	Not Available	Not Available
LR18	Faculty Load Sheet: display Month wise	HR04	Implemented	Not Available
LR19	Faculty Load Sheet: display Initiative wise	HR04	Implemented	Not Available
LR20	Faculty name correction/validation in Faculty Load Sheet	HR04	Not Available	Not Available
LR21	Faculty Load Sheet: Display Available Slots Faculty wise	HR04	Implemented	Not Available

ID	Feature	High Level ID	MATLAB v0 Status	Python v0 Status
LR22	Faculty Load Sheet: Warn User if Available Slots goes Negative	HR04	Not Available	Not Available
LR23	Faculty load insight : OVERLOAD, UNDERLOAD, OPTIMUM	HR04	Not Available	Not Available
LR24	Faculty Load Sheet: Appending	HR04	Implemented	Not Available
LR25	Let User know that the Output has been Successfully Updated	HR02, HR03, HR04	Implemented	Implemented
LR26	Validate correct number of days in month	HR02, HR03	Not Available	Not Available
LR27	Let the User/Faculty book their Slots Themselves	HR05	Not Available	Not Available
LR28	Accessible by everyone with login Credentials	HR06	Not Available	Not Available
LR29	Bar graph indicating the number of slots planned for each faculty	HR07	Not Available	Not Available
LR30	Bar Graph indicating the sessions per initiative	HR07	Not Available	Not Available
LR31	Pie chart indicating the % of sessions being taken according to initiatives	HR07	Not Available	Not Available

Implementation and Summary

Git Link:

Link: https://github.com/Ramki17/Calendar_Automation-Genesis21_Team49

Individual Contribution and Highlights

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

Mini project 5 –Team BMW [Team]

Module: - Applied Model Based Design Module

Individual Topic:- Seat Control System

Requirements

Introduction

With the increased need for safety, aesthetics, infotainment, and comfort in automobiles, automakers are concentrating on introducing maximum automation into vehicles that can give both comfort and safety. One such feature is the seat control, which allows the driver to adjust his or her seat.

Seat control/adjustment is a mechanism for self-adjusting a seat to the comfort of the user. When a person enters 0, 1, -1 as an input, the seat will change according to this design.

Based on the ergonomics of human body, the simulation is created. The feature controls the actions related to adjustment.

Overview

The seat and its functions are an important aspect of cabin comfort and safety, and as one of the direct interactions between the driver and passengers, it contributes to the user experience and serves as a differentiator for car manufacturers.

SEAT PRIMARY FUNCTION: The seat system is the most significant portion of the vehicle since it is in constant contact with the occupant and is directly responsible for the occupant's comfort and safety.

1. Occupant Assistance:

a. The occupant should have consistent support for a long time.

b. The Seat should be able to accommodate people of various weights, sizes, and dimensions.

2. Position of Occupant:

a. The occupant's posture is critical for the vehicle's safe operation.

b. The occupant should be ergonomically positioned to have a clear field of vision.

c. There should be adequate head, leg, and arm room for the occupant.

3. Protect the Occupant:

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.

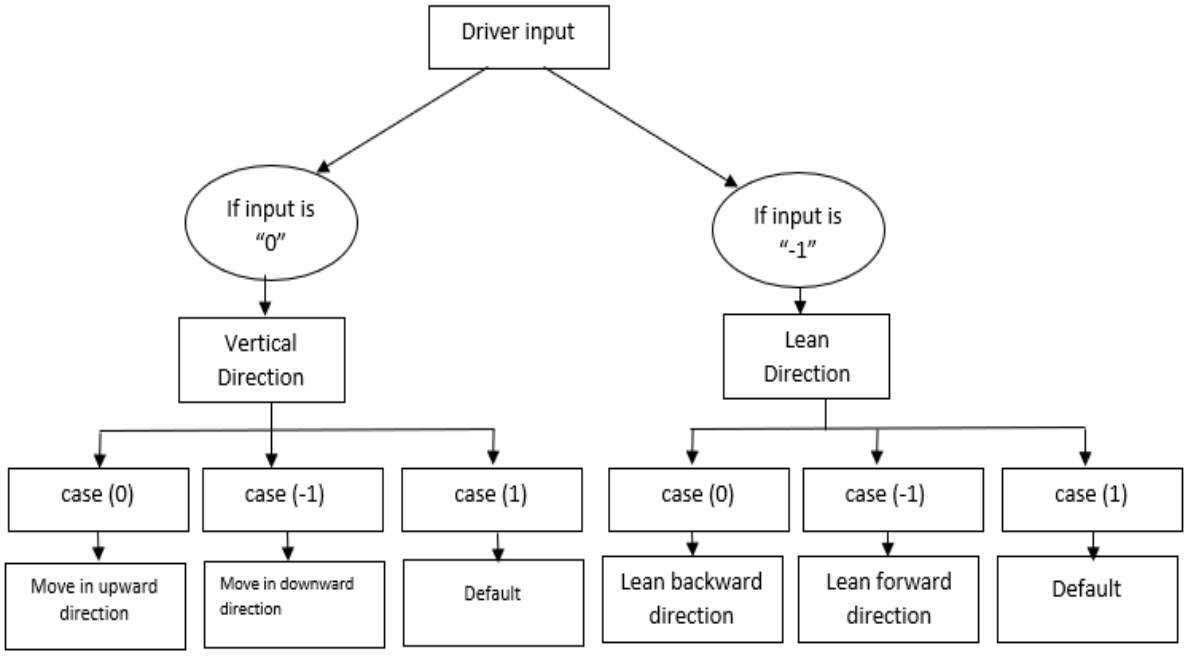
High Level Requirements:-

ID	Description
HLR1	Receive signals from the user, and adjust the seat.
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.
HLR4	Set the seat in the direction as per user requirement.

Low Level Requirements:-

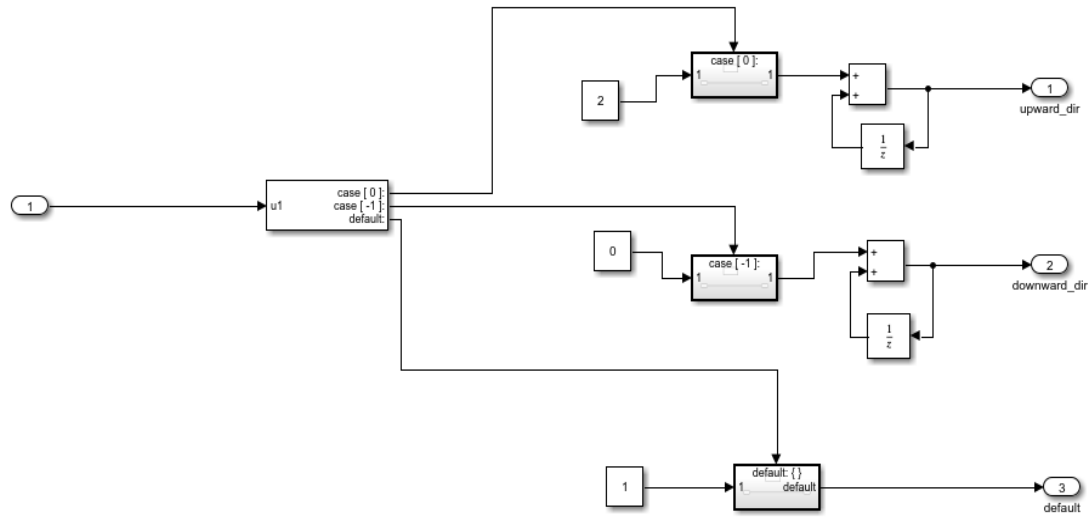
ID	Description
LLR1	Put the input value.
LLR2	The system test will engage when the car is turned on.
LLR3	Compare the conditions.

DESIGN



MODELING

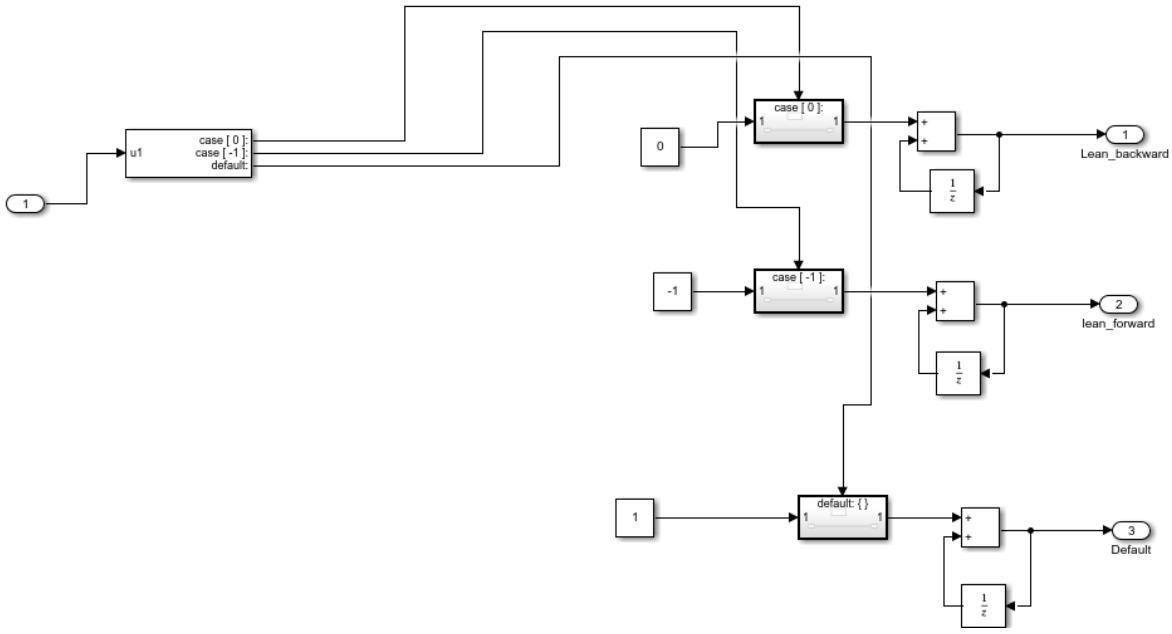
Vertical direction



Vertical Direction

Test Case no	Input	Condition	Result
1	0	move in upward direction	Pass
2	1	Default	Pass
3	-1	Move in Downward Direction	Pass

Lean Direction

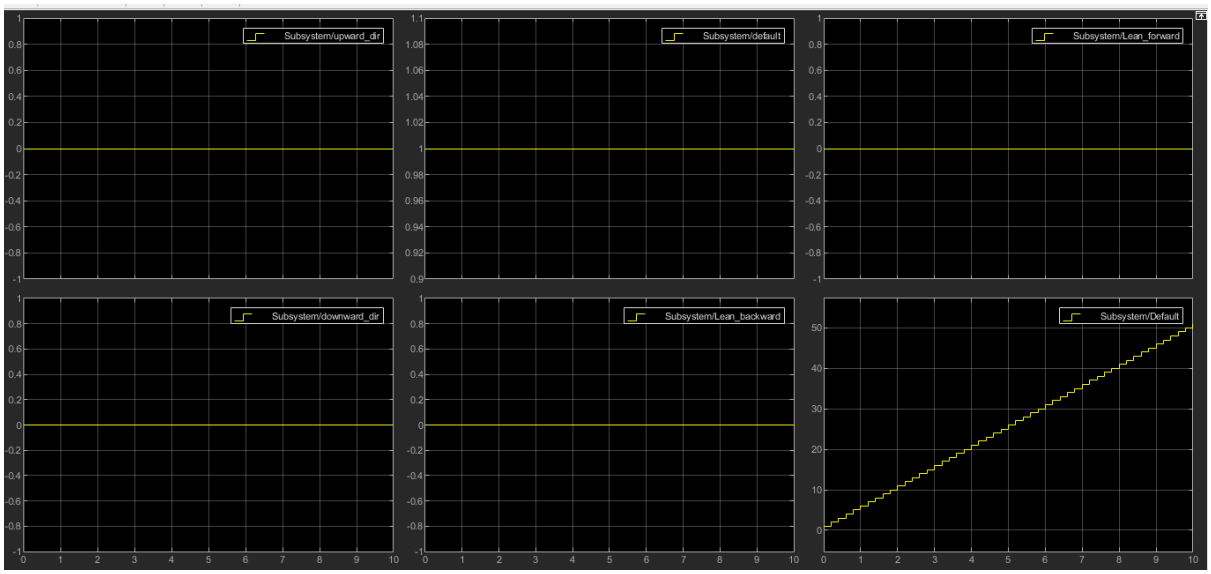


Lean Direction

Test Case no	Input	Condition	Result
1	0	lean backward direction	Pass
2	1	Default	Pass
3	-1	Lean in forward Direction	Pass

Output

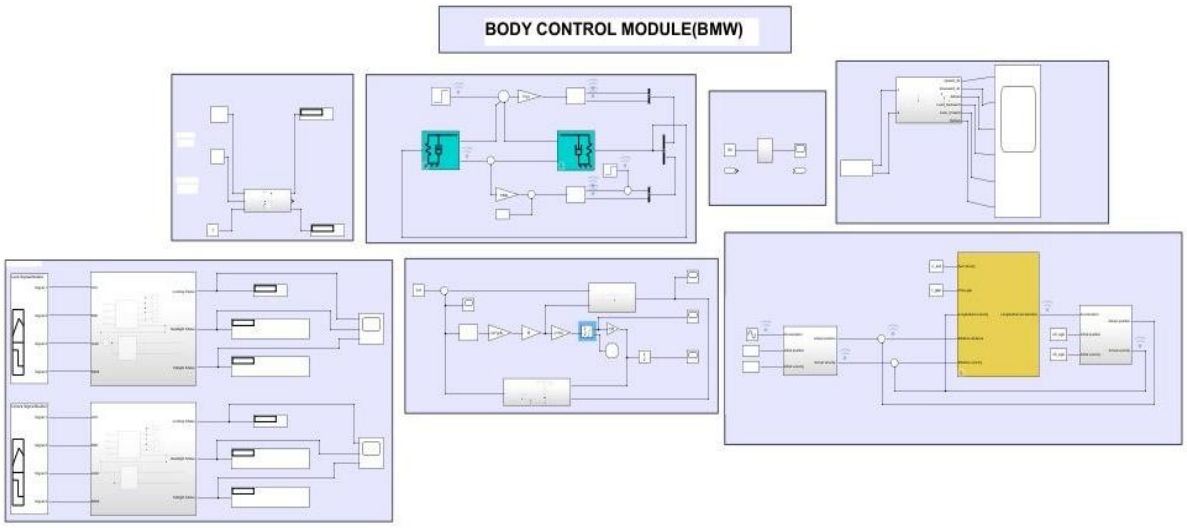
Running the Simulation in Seat control Mode



Conclusion

This model shows how you can use Simulink to simulate a seat control 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 seat 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



Miniproject 6 – Wiper Control[Team]

Modules

1. C Programming
2. STM32

4W's and 1H

Who:

A wiper control system for an car wiper. It is mainly used in cars.

Where:

It is implemented through STM32F4 microcontroller in embedded system.

When:

Whenever the water hit a devoted sensor that found on windscreen, it'll send a flag to move on the wiper motor. Once water isn't identified by sensor, the wiper will naturally halt. This will offer assistance the driver to donate more concentration and decrease the car mishap probability.

What:

Vehicles are presently accessible with driver-programmable brilliantly windscreen wipers that identify the presence and sum of rain employing a rain sensor.

How:

Windshield wipers are controlled by the stalk on the proper side of your controlling wheel. Essentially moving the stalk down will turn your windshield wipers on. Moving the stalk down will turn your you wipers on.

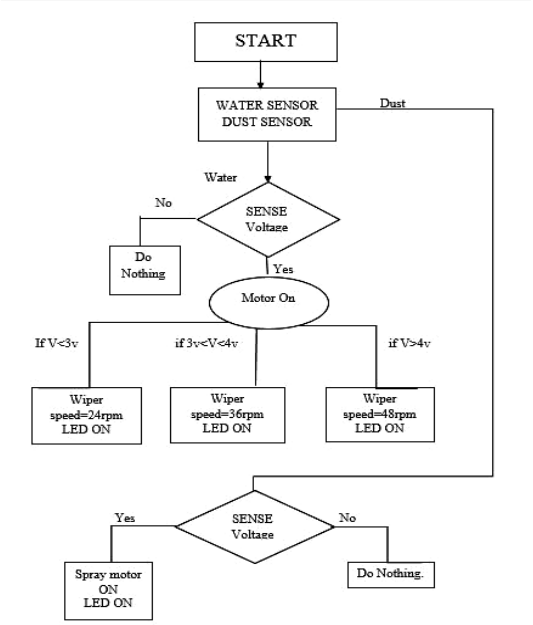
High Level Requirements

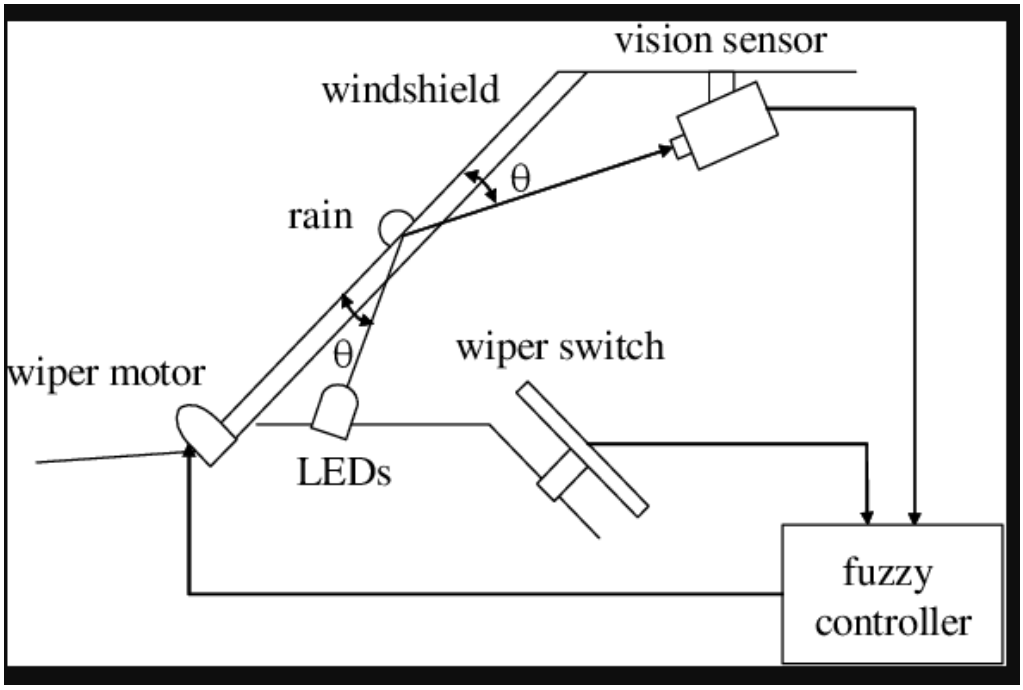
id	Description	Status
HLR1	A windscreen wiper is a device used to do away with rain, snow, ice and particles from a windscreen or windshield.	Implemented
HLR2	The quality and reliability wiper systems meet the highest technical requirements and are the basis for vehicles with state-of-the-art capabilities.	Implemented
HLR3	Our assignment brings ahead this device to automate the wiper gadget having no need for manual intervention.	Implemented

Low Level Requirements

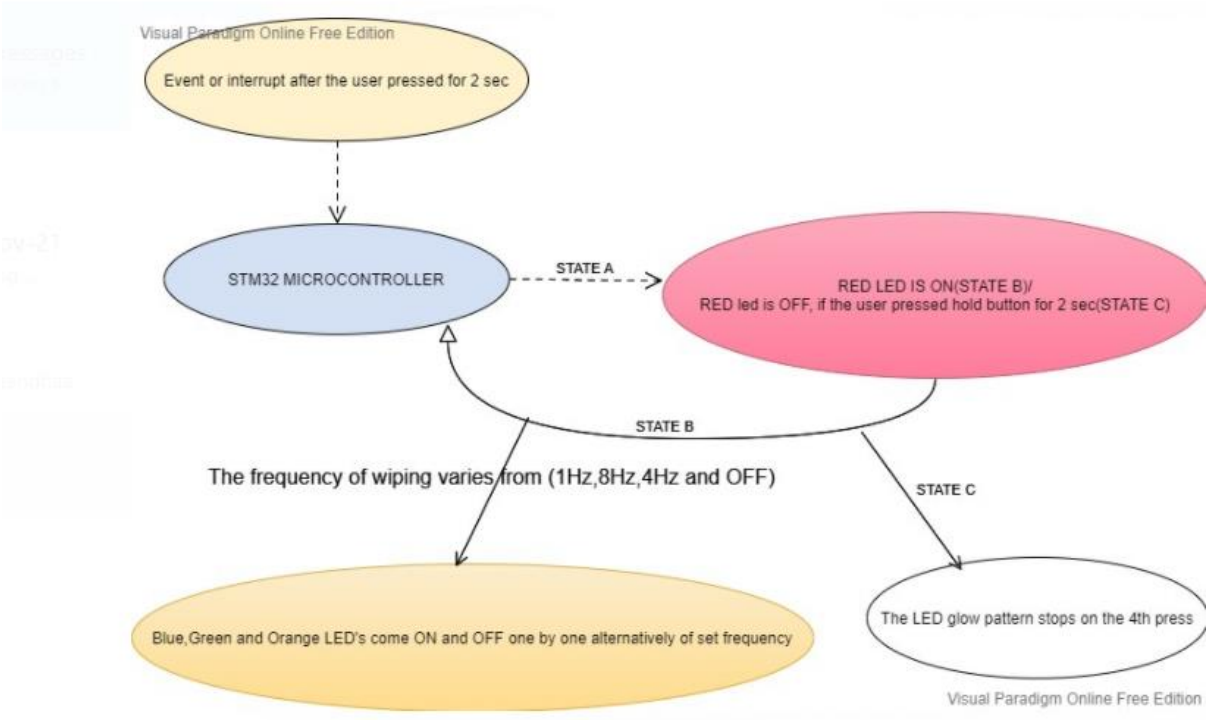
id	Description	Status
LLR1	Lower degree parsing , below the hood, Requirement elegance does maximum of the heavy lifting. magnificence necessities.	Implemented
LLR2	These structures discover droplets of rain at the windshield and robotically activate and adjust the wiper machine	Implemented
LLR3	These structures discover droplets of rain at the windshield and robotically activate and adjust the wiper machine.	Implemented

Design: Flowchart Diagram





Usecase Diagram



TEST PLAN:

High level test plan

HLR_ID	Description	Expected i/p	Expected o/p	Actual o/p	Test type
HLR_1	Check Every is working or not	Call the function	All functions working correctly	All functions working properly	Requirement
HLR_2	Checking individual functionality working	Call individual function	Individual function is called	Individual function is called	Scenario
HLR_3	Check For not available function	Choosing different values	No response from the program	No response from the program	Boundary
HLR_4	Rain Sensing Devices	When Rain fall	Automatic Wiper On	not Implemented	Future

Low level test plan

LLR_ID	Description	Expected i/p	Expected o/p	Actual o/p	Test type
LLR_1	Testing system is ON or Off by pressing button for 2 seconds	Press button for 2 Seconds	RED Led will glow	RED Led will glow	Requirement

LLR_ID	Description	Expected i/p	Expected o/p	Actual o/p	Test type
LLR_2	Testing the modes of the Wiper control system	Pressing the button one time	Blue Led will glow	Blue Led will glow	Requirement
LLR_3	Testing the modes of wiper control system	Pressing the button for 1 second	Blue will Off and Green will be glow	Blue will Off and Green will be glow	Requirement

Implementation and Summary

Git Link: https://github.com/GENESIS-2022/MasteringMCU-Team2

Individual Contribution and Highlights

1. Wiper System using C Programming
2. Source code management using GitHub

Role in Project Team

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

Miniproject 7 – Ford Project[Team]

Modules

1. Automotive Systems
2. Git

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 Topic :- Power Window

Introduction

The windows on cars which can be opened or shut with the help of buttons, are called power windows. 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. Power windows have replaced the traditional manual handles. It can be a built-in feature or installed in cars, using aftermarket accessories.

4W's and 1H

What: Power windows are automobile windows which can be raised and lowered by pressing a button or switch, as opposed to using a crank handle.

Where: whenever there is dust or rain the windows will close automatically. The tint of the window will change when there is abundance of sunlight.

Why: The power windows eliminates the hurdle of the traditional handles which needed a lot of effort and time to open or shut the windows. It allows the driver to control the windows with just the touch of his fingers.

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: Power windows are controlled by switches and wires and are powered by battery or electricity. Power windows do not function if the ignition of the car is not turned on. Unlike the traditional windows, power windows do not have manual handles. They do not work manually.

High level Requirement

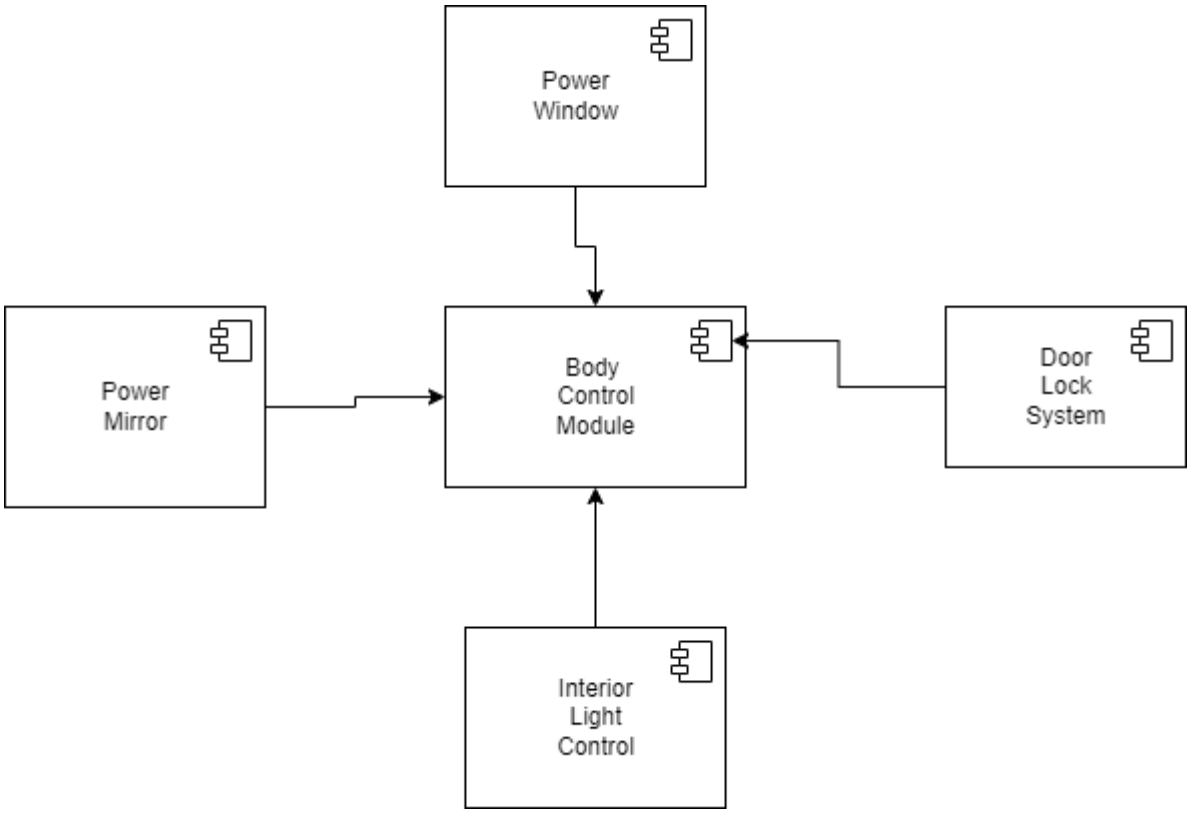
	Description
HLR_1	When there is dust or rain the windows will close automatically.
HLR_2	The tint of the window will change when there is abundance of sunlight.
HLR_3	When the ac is on it will get detected and the windows will close automatically.
HLR_4	When the safety feature is on, only the driver will be able to control the positions of the windows.
HLR_5	When anything gets in the way of the window while goes up, the window will go down again.

Low level Requirement

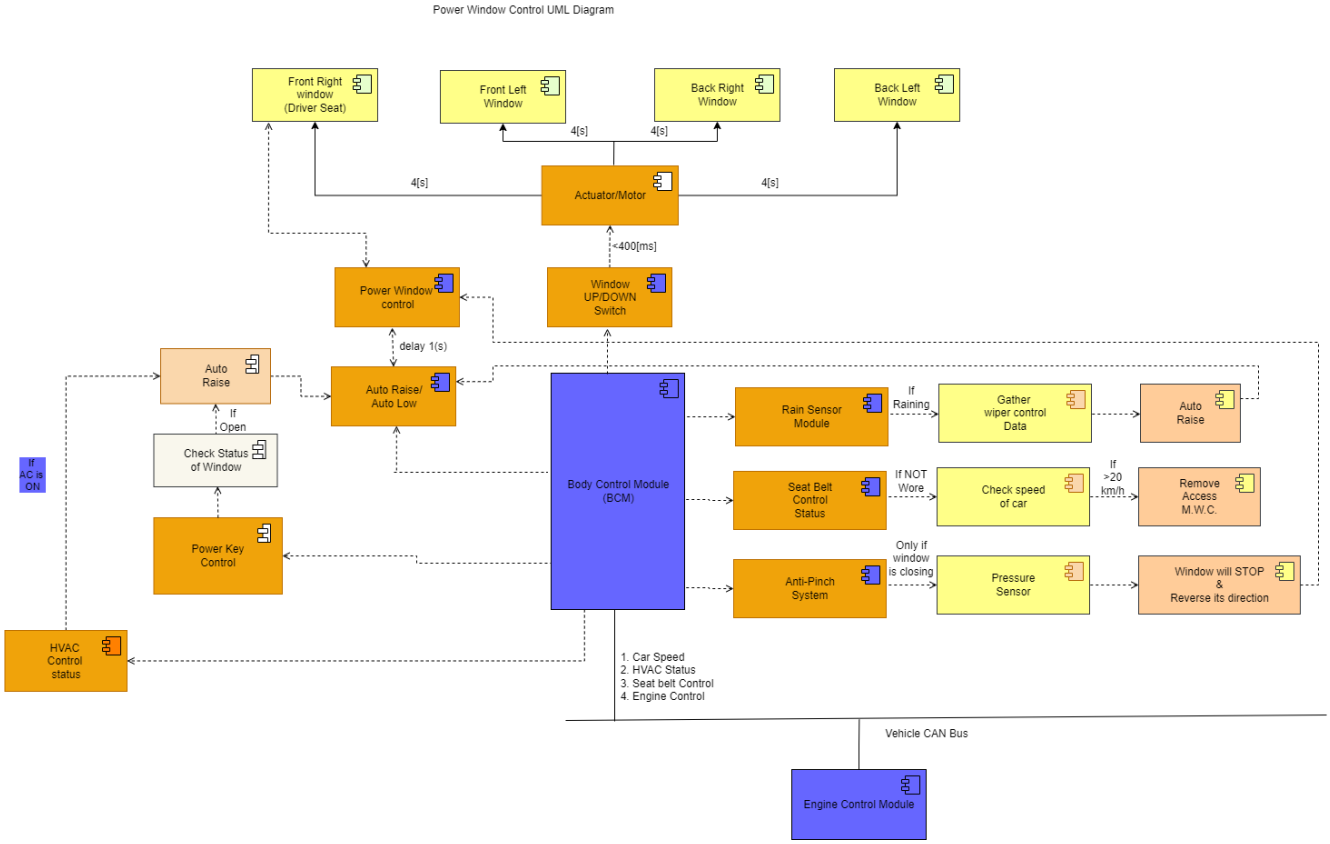
	Description
LLR_1	The power window go all the way down by just tapping the button once. (Automatic down)
LLR_2	The power windows will go all the way up like the automatic down.(Automatic Up)
LLR_3	When the door is locked the windows will close automatically.

Design

Body Control Module



Power Window



Implementation and Summary

Git Link:

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

Miniproject 8 – MINI AIRCRAFT[Team]

Modules

1. Matlab
2. Matlab Script

Requirements :

Introduction:

An electrical Energy management based on fixed priorities of the loads is considered a conventional implementation as applied in today's aircraft systems. It can cut and reconnect loads depending on their importance. Further implementations are depicted that are able to eliminate certain drawbacks of such a typical load management

Objective:

The main objective of the project is to Reduce the energy consumption of the Aircraft

Features:

This project supports the following types of Energy Management in Aircraft

1.Source Management:

If sources are available that can be connected in parallel one can apply a source management, that controls the different sources or generators in an energy efficient way. An intelligent source management will regulate the several sources to reach the overall power losses

2.Electrical Storage Device:

The degree of freedom of an energy management method increases considerably if electrical storage devices like bat-terries or super caps are available. Storages can be used to smooth out the power consumption of load groups. This in turn enables to design lighter generators, feeders, and convert especially in case of many non-constant loads. However, the batteries or supercaps will add weight. Thus, there will bean optimal trade off between installed battery-capacity and installed power of e.g. generators to minimize weight

3.Exploit Slow Responding Roads:

In today's aircraft systems there is a number of slow responding loads. That is systems and components with large time constants like heaters. Since electrical storages will add weight, one can also try to decrease power peaks by exploiting such slow responding loads (SRL). Thus, they can be handled like an electrical storage since they store energy in their respective physical state like the heat of a galley oven.

4.Variable Priorities:

To consider the changing importance of loads during a flight one can simply use variable priorities instead of fixed ones. Thus, the priority can be determined by the loads themselves depending on their current importance.

5.Supervise Reconnection

Instead of shedding loads if an overload occurs, one can also prevent loads from being reconnected if a dedicated power level is reached.

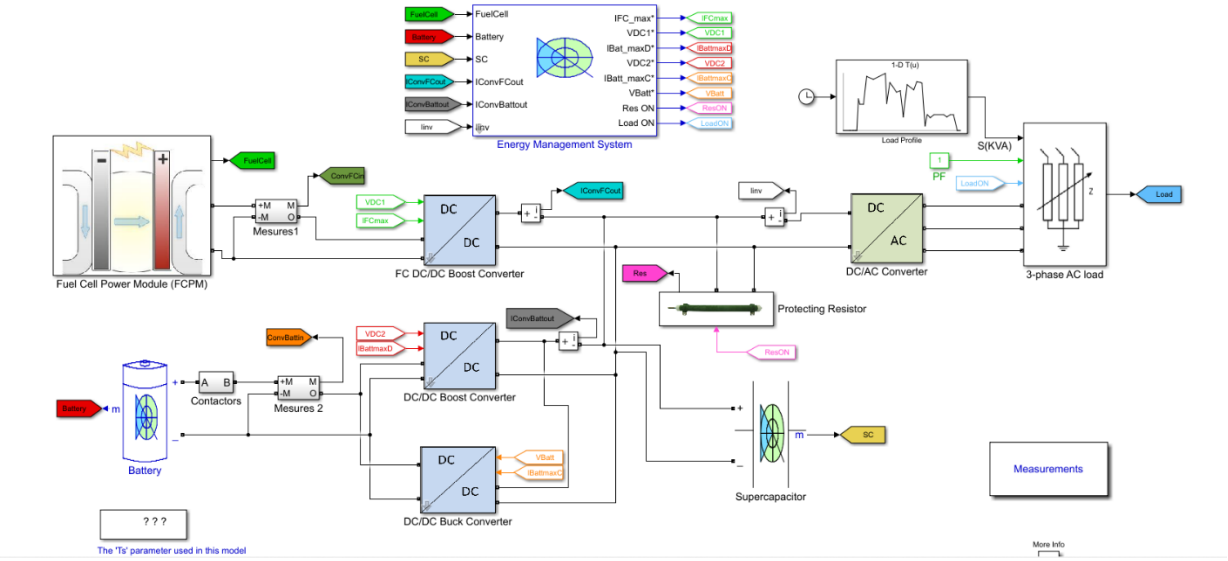
Eviation Alice:

COMPONENTS	E-FLYER 2	EVIATION ALICE
CREW	1	2
CAPACITY (passenger)	1	9
WING SPAN	38 ft (12m)	56 ft (18m)
POWER (kw)	90	640
SPEED (km/hr)	250	407

COMPONENTS	E-FLYER 2	EVIATION ALICE
Propeller	3- Blade composite	3- Blade composite
Manufacturer	Bye Aerospace	Eviation Aircraft
Range(km)	420	815
Endurance(hours)	3.5	8.15
Motor type	Safran electric motor	magniX 650
No. of Battery Packs	6	10
Gross Weight	862	1100
Battery Type	li-ion battery	li-ion battery

COMPONENTS	HAWK AIRCRAFT
CREW	2
CAPACITY (passenger)	10
WING SPAN	52
POWER	520
SPEED (km/hr)	480
Propeller	3- Blade composite
Manufacturer	HAWK AEROSPACE
Range(km)	750
Endurance(hours)	6
No. of Battery Packs	12
Gross Weight	850
Battery Type	li-ion battery

Simulation:



Implementation and Summary

Submission: Submitted in GEALearn

Individual Contribution and Highlights

1. Done in Matlab Script

Role in Project Team

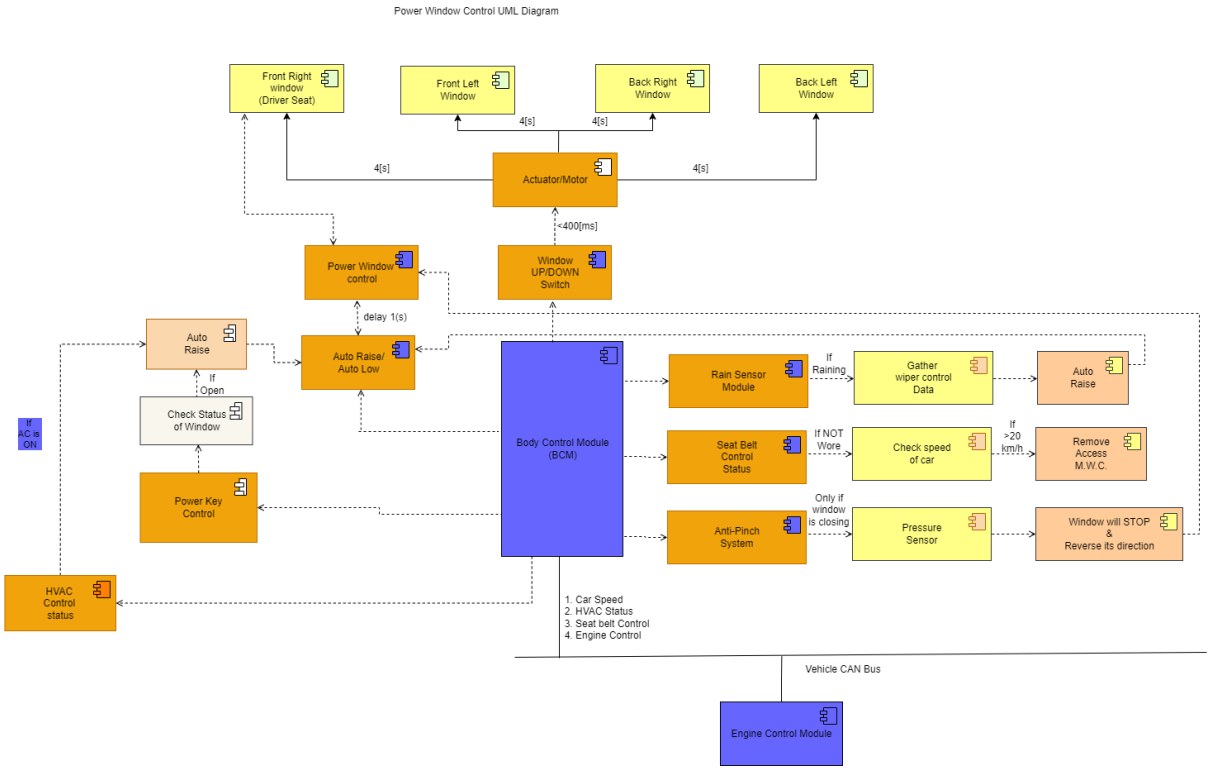
1. Done Matlab scripting for Mini Aircraft Bike
2. Researcher: Done case study for Mini Aircraft Bike

Miniproject 9 – Power Window [Individual]

Modules

1. Autosar
2. Git

Design



Implementation and Summary

Git Link:

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

Individual Contribution and Highlights

1. Power Window Case Study
2. Source code management using GitHub
3. AtomicSwComponent
4. SWCInternalBehavior
5. SWCImplementation