GENESIS - Learning Outcome & Mini-project Summary Report

L&T Technology Services

Details

Ver. Rel. No.	Release Date	Prepared By	Module Name	To Be Approved	Remarks/Revision Details
1.0	16/02/2022	Franklin Cedric.J 40020514	C Programming On Multiple Platforms		
1.0	16/02/2022	Frankliln Cedric.J 40020514	Essesntials of Embedded Systems		
1.0	16/02/2022	Frankliln Cedric.J 40020514	Applied SDLC and Software Testing		
1.0	16/02/2022	Frankliln Cedric.J 40020514	Applied Model Based Design Module		
1.0	16/02/2022	Frankliln Cedric.J 40020514	OOPS with Python		
1.0	16/02/2022	Frankliln Cedric.J 40020514	Mastering Microcontrollers with Embedded Driver Development Module		
1.0	16/02/2022	Frankliln Cedric.J 40020514	Overview of Automotive Systems		
1.0	16/02/2022	Frankliln Cedric.J 40020514	Applied Control Systems and Vehicle Dynamics		
1.0	16/02/2022	Frankliln Cedric.J 40020514	Classic Autosar Basic to Intermediate		

Contents

List of Figures

Miniproject – 1: Scientific Calculator [Individual]

Modules:

1. C Programming
2. Git

Requirements

4W's and 1 H's

Why:

1. To reduce the complexcity of the mathemetical calculations
2. To help user in viewing his data and privileges
3. It can be used by anyone.

Where:

1. It can be used in our daily lives.
2. We can use it in sovlving complex problems in Mathemathics and Physics

Who:

1. It can be used by anyone(EX: students,teachers etc.).
2. Can be used as a reference for advanced mathemathi calculations.

When:

1. One can calculate the mathemathical problems anywhere.
2. The project can be used when the complex mathematical calculations are required and the result will be obtained fastly.

How:

1. They can sovlve any calculations by this calculator and it gives them a correct result.
2. It will be helpful in performing various calculations like Natural Logarithm and PI.

High Level Requirements

ID	Description	Status
HLR_1	The user can calculate the Addition	Implemented
HLR_2	The user can calculate the Subraction	Implemented

ID	Description	Status
HLR_3	The user can calculate the Multiplication	Implemented
HLR_4	The user can calculate the Division	Implemented
HLR_5	The user can calculate the Power	Implemented
HLR_6	The user can calculate the Sin	Implemented
HLR_7	The user can calculate the Cos	Implemented
HLR_8	The user can calculate the Root	Implemented
HLR_9	The user can calculate the Tan	Implemented
HLR_10	The user can calculate the Sec	Implemented
HLR_11	The user can calculate the Cosec	Implemented
HLR_12	The user can calculate the Cot	Implemented
HLR_13	The user can calculate the Exponential	Implemented
HLR_14	The user can calculate the Factorial	Implemented
HLR_15	The user can calculate the Logritham	Implemented
HLR_16	The user can calculate the Exit	Implemented

Low Level Requirements

ID	Description	Status
LLR_1	List of operations displayed	Implemented
LLR_2	Input from the user	Implemented
LLR_3	Exit the program	Implemented

Design

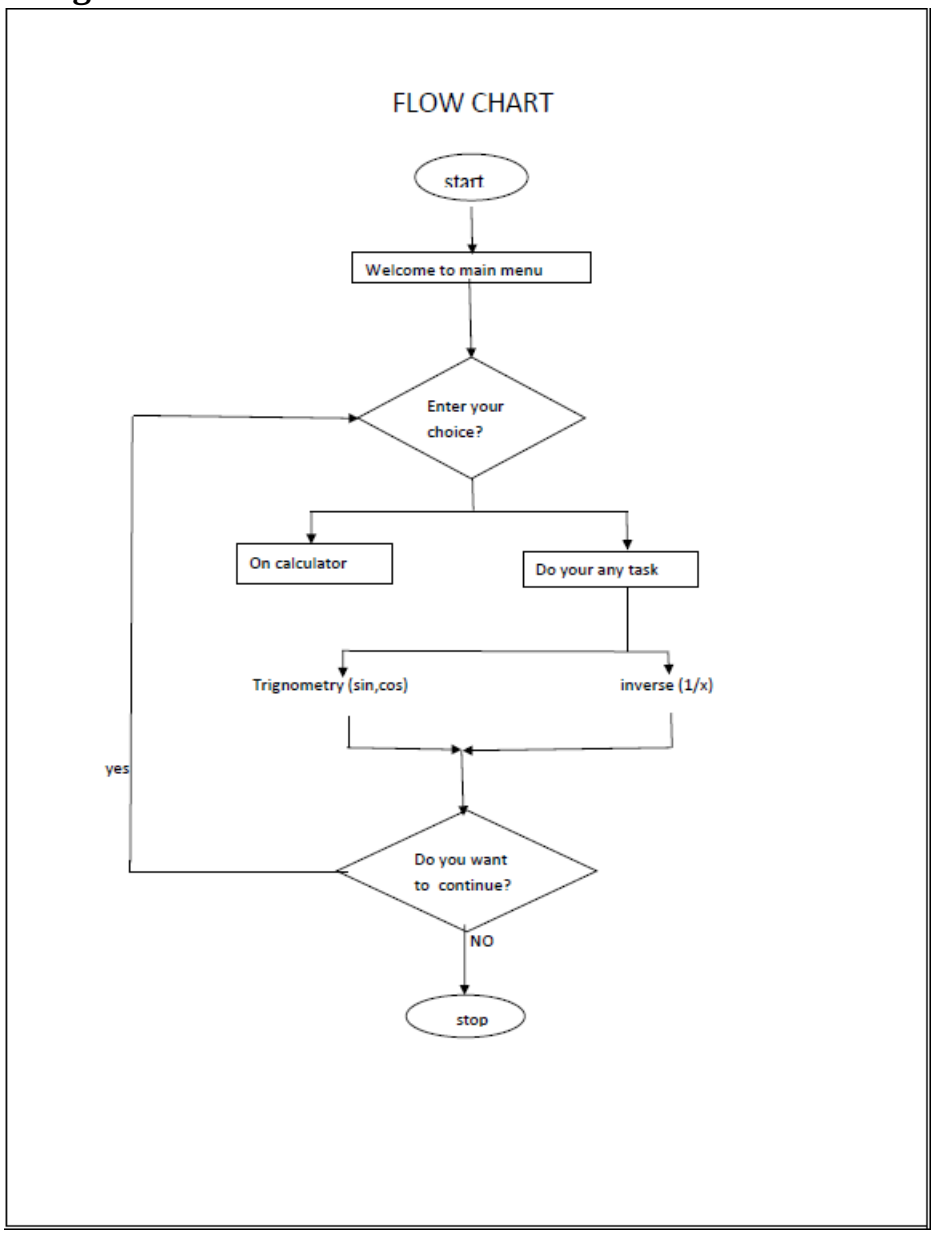

Figure 1 Behavior Diagram

Test Plan

High Level Test Plan

ID	Description	Expected o/p
HLTP_1	Addition Calculation	Implemented
HLTP_2	Subraction Calculation	Implemented
HLTP_3	Multiplication Calculation	Implemented
HLTP_4	Division Calculation	Implemented
HLTP_5	Power Calculation	Implemented
HLTP_6	Sin Calculation	Implemented
HLTP_7	Cos Calculation	Implemented
HLTP_8	Root Calculation	Implemented
HLTP_9	Tan Calculation	Implemented
HLTP_10	Sec Calculation	Implemented
HLTP_11	Cosec Calculation	Implemented
HLTP_12	Cot Calculation	Implemented
HLTP_13	Exponential Calculation	Implemented
HLTP_14	Factorial Calculation	Implemented
HLTP_15	Logritham Calculation	Implemented

Low Level Test Plan

ID	Description	Expected I/P	Expected O/P	Actual O/P	Type Of Test
LLTP_1	Addition Calculation	(444 + 444)	888	888	Requirement Based

ID	Description	Expected I/P	Expected O/P	Actual O/P	Type Of Test
LLTP_2	Subraction Calculation	(476 - 323)	153	153	Requirement Based
LLTP_3	Multiplication Calculation	(25 * 35)	875	875	Requirement Based
LLTP_4	Division Calculation	(72 / 2)	Quotient=36 Remainder=0	Quotient=36 Remainder=0	Requirement Based
LLTP_5	Power Calculation	(10 , 5)	100000	100000	Requirement Based
LLTP_6	Sin Calculation	333	-0.00882116611	-0.00882116611	Requirement Based
LLTP_7	Cos Calculation	555	-0.487	-0.487	Requirement Based
LLTP_8	Root Calculation	81	9	9	Requirement Based
LLTP_9	Tan Calculation	878	13.282	13.282	Requirement Based
LLTP_10	Sec Calculation	777	-1.932	-1.932	Requirement Based
LLTP_11	Cosec Calculation	444	-1.163	-1.163	Requirement Based
LLTP_12	Cot Calculation	768	0.120	0.120	Requirement Based
LLTP_13	Exponential Calculation	4	54.598	54.598	Requirement Based
LLTP_14	Factorial Calculation	5	120	120	Requirement Based

ID	Description	Expected I/P	Expected O/P	Actual O/P	Type Of Test
LLTP_15	Logritham Calculation	45	1.531	1.531	Requirement Based

Implementation and Summary

Git Link:

Link: https://github.com/cedricxavi/M1_Application_Scientific_Calculator.git

Git Dashboard

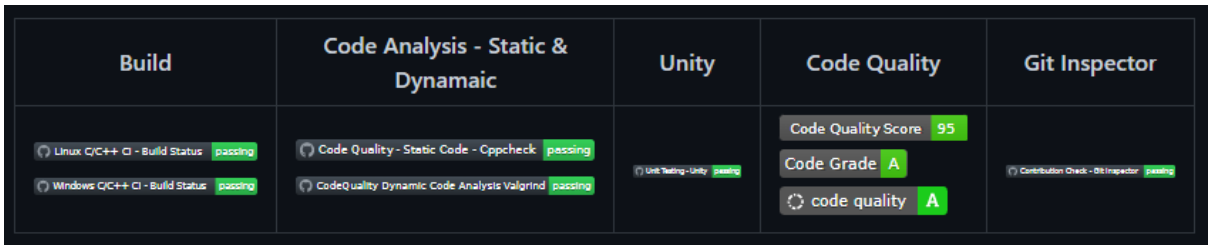

Figure 2 Git Dashboard

Summary

Git Inspector Summary

```
Author                    Commits    Insertions      Deletions    % of changes
cedricxavi                     50          4944            203          100.00

Below are the number of rows from each author that have survived and are still
intact in the current revision:

Author                    Rows      Stability          Age       % in comments
cedricxavi                4763           96.3          0.4               11.61

The following history timeline has been gathered from the repository:

Author                 2021W46    2021W47
cedricxavi            -+++++++   ++++++++
Modified Rows:              68       5079
```

Figure 3Git Inspector Summary

Miniproject 2 – Fire Alarm System [Individual]

Modules

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

Requirements

4W's and 1 H's

Why:

1. To reduce the fire accidents in House and Industries.
2. It can be used at one place.

Where:

1. This can be used in our Houses.
2. We can use it in the Banks, Industies, Shopping Malls etc.

Who:

1. Can be used as a reference for advanced fire alarm System.

When:

1. When fire accidents is going to occur these alarm systems can be used.
2. This project is used to prevent people from fire accidents.

How:

1. It sense the fire and gives an alarm sound as an output.
2. It will be helpful in preventing fire accidents in any places

High Level Requirements

ID	Description	Status
HLR_1	Microcontroller	Implemented
HLR_2	Servo motor	Implemented
HLR_3	Buzzer	Implemented
HLR_4	Resistor	Implemented

ID	Description	Status
HLR_5	Capacitor	Implemented
HLR_6	Button	Implemented
HLR_7	AVR-ISP PROGRAMMER	Implemented
HLR_8	Sensor	Implemented

Low Level Requirements

ID	Description	Status
LLR_1	ATmega328	Implemented
LLR_2	Servo motor (sg90)	Implemented
LLR_3	10KΩ resistor	Implemented
LLR_4	1KΩ resistor	Implemented
LLR_5	220Ω resistor	Implemented
LLR_6	100nF capacitor	Implemented
LLR_7	AVR-ISP PROGRAMMER	Implemented
LLR_8	Buzzer	Implemented
LLR_9	Button	Implemented
LLR_10	Fire Sensor	Implemented

Design

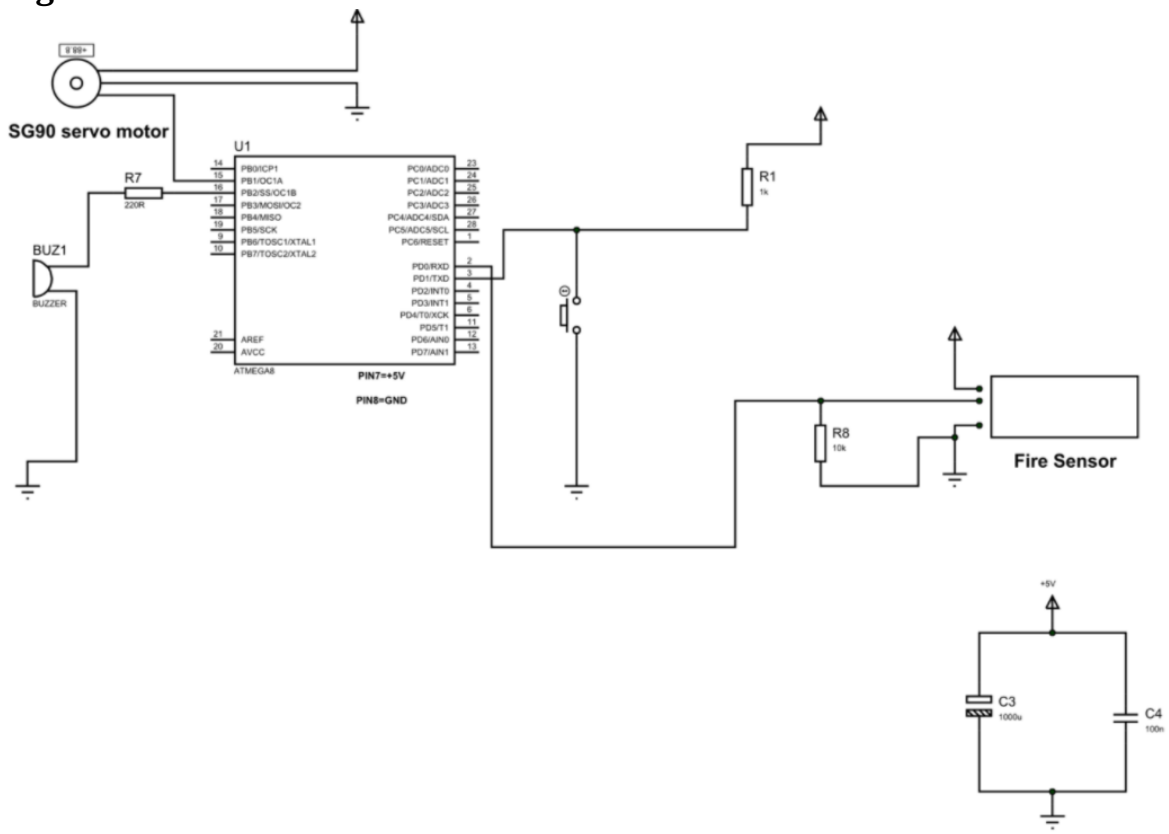

Figure 4 Block Diagram

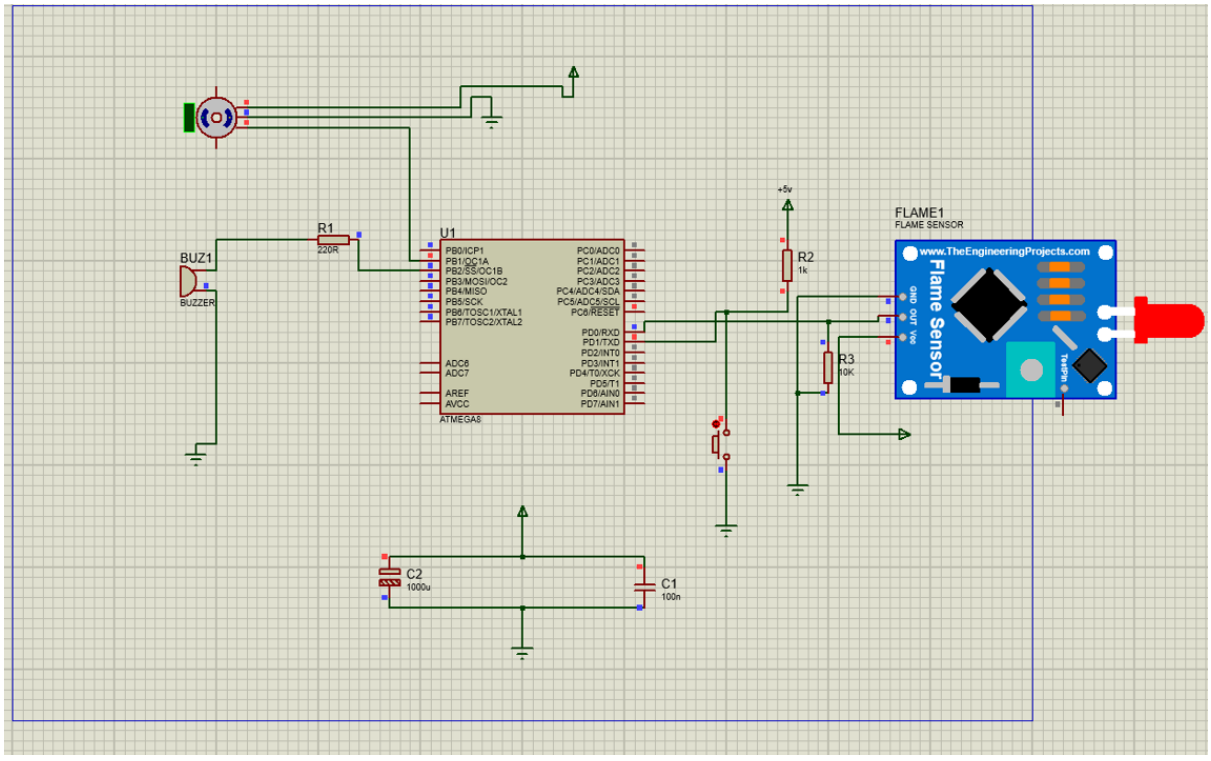

Figure 5 Simulation

Test Plan

High Level Test Plan

Test ID	Description	Exp I/P	Exp O/P	Actual Out	Type of Test
HLT_01	Power ON	Power	Display ON	SUCCESS	Requirement Based
HLT_01	Fire Sensor	900 Degree Clesius	Fire sensed	SUCCESS	Requirement Based

Low Level Test Plan

Test ID	Description	Exp O/P	Actual Output
LLR_01	Buzzer	alram sound	alram sound
LLR_02	Fire sensor	Fire Sensed	Fire Sensed

Implementation and Summary

Git Link:

Link: https://github.com/cedricxavi/M2_Fire_Alarm_System.git

Git Dashboard

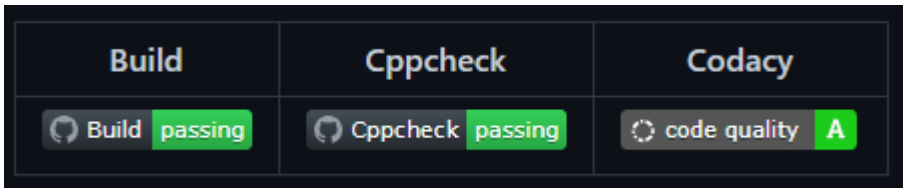

Figure 6 Dash Board

Git Summary

```
Author                    Commits    Insertions      Deletions    % of changes
cedricxavi                     37           610             38          100.00

Below are the number of rows from each author that have survived and are still
intact in the current revision:

Author                    Rows      Stability         Age       % in comments
cedricxavi                 487           79.8         8.5                3.49

The following history timeline has been gathered from the repository:

Author                 2021W48    2022W07
cedricxavi            ++++++++  +++++++++
Modified Rows:             597         51
```

Figure 7Git Summary

Miniproject 3 – NFT Marketplace [Team]

Modules

1. SDLC
2. Git

Requirements

4W's and 1 H's

Why:

1. It can be used by anyone at any place.
2. Digital Items are the Future
3. Individual Creators can use this platform to sell the Digital products.

Where:

1. This can be used in our daily lives.
2. Can be used for international transactions

Who

1. It can be used by anyone.
2. It can be used as a reference for marketplace.

When:

1. One can buy, sell or create anytime.
2. The project can be used when the anyone wants to buy an NFT.
3. Can be used without any centralised authority

How:

1. By using a crypto wallet anyone can Buy or Bid on NFT.
2. It will be helpful for Digital Creators.

High Level Requirements

ID	Description	Status
HLR_1	Create NFT	Implemented
HLR_2	Sell NFT	Implemented

ID	Description	Status
HLR_3	Bid NFT	Implemented
HLR_4	Buy NFT	Implemented
HLR_5	Contact	Implemented

Low Level Requirements

ID	Description	Status
LLR_1	Sign In	Implemented
LLR_2	Register	Implemented
LLR_3	Connect Wallet	Implemented
LLR_4	Activity	Implemented
LLR_5	Forgot password	Implemented
LLR_6	Signup	Implemented

Design

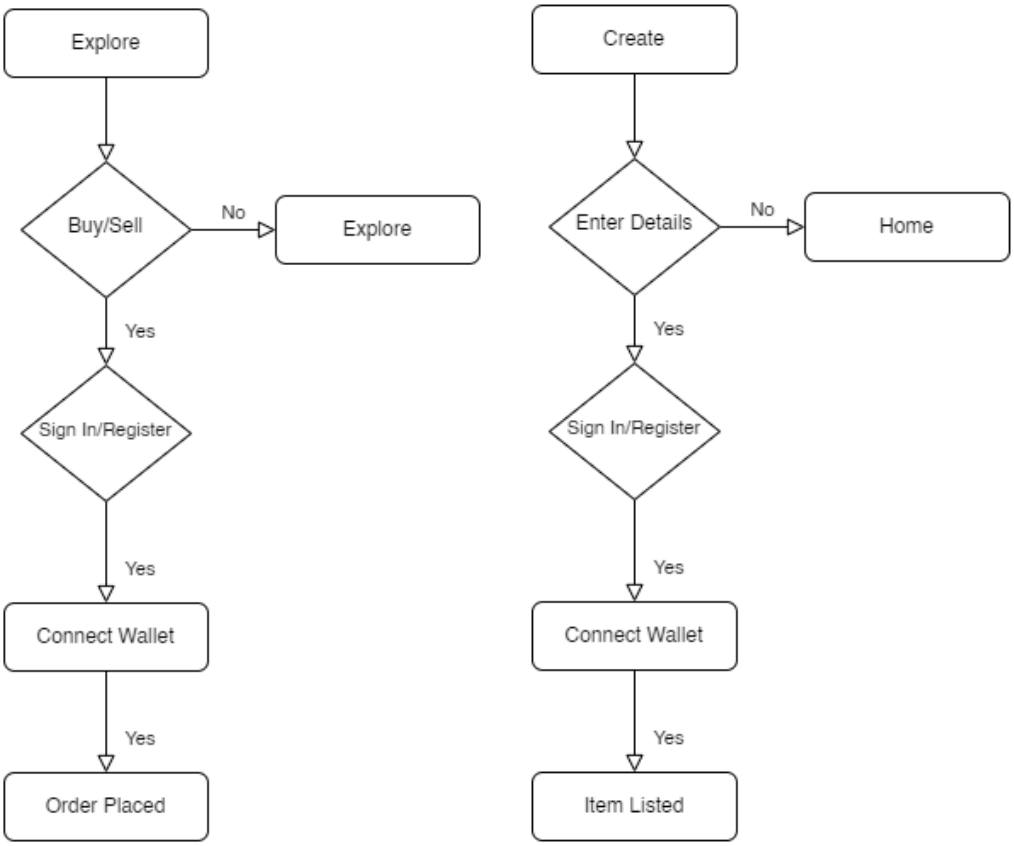

Figure 8 Behavior Diagram

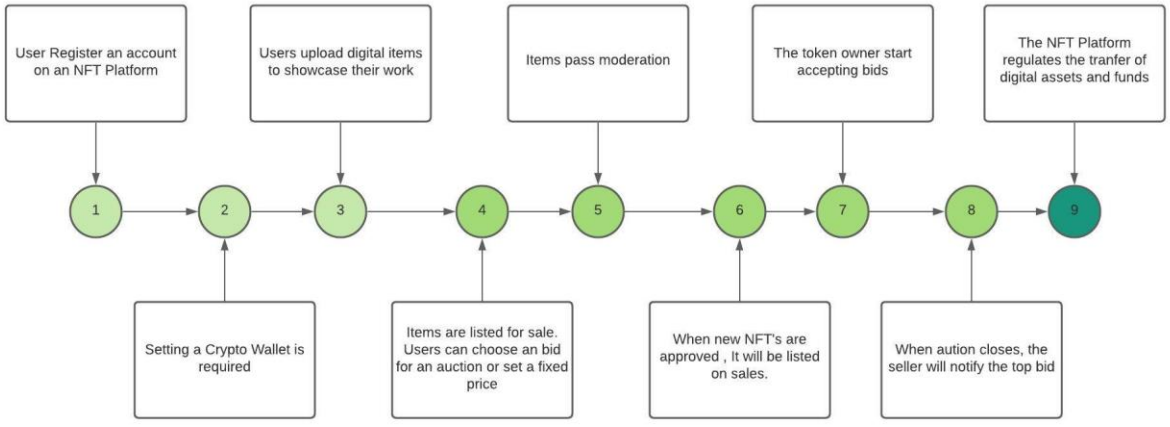

Figure 9 UserFlow Diagram

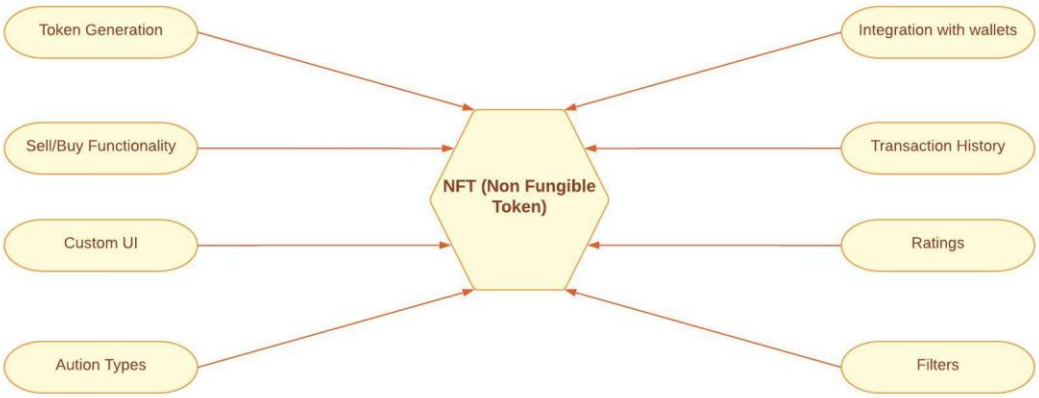

Figure 10 Structure Diagram

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
HLTP_6	Sign In	Click	SUCCESS	SUCCESS	Requirement Based
HLTP_7	Register	Click	SUCCESS	SUCCESS	Requirement Based

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_2

Live Project Link: https://alrichroshan.com/nft

Individual Contribution and Highlights

Summary

1. Designed Homepage
2. Search Option
3. Header
4. Footer
5. Integrating All Pages Together

Role in Project Team

1. Designer: Designed Webpages Using HTML, CSS, JavaScript
2. Integrator: Integrated All the Pages Together
3. Tester: Testing the Webpage Performance and Bugs

Miniproject 4 – Attendance Automation[Team]

Modules

1. Python
2. Git

Requirements

High Level Requirements

ID	Feature	Status
HLR_01	GUI	Not Implemented
HLR_02	Attendance Status	Implemented
HLR_03	User Details	Implemented
HLR_04	User load sheet	Implemented
HLR_05	Output file generation	Implemented

Low Level Requirements

ID	Feature	High Level ID	Status
LLR_01	GUI should allow user to enter inputs	HLR_01	Not Implemented
LLR_02	Input Files For Different Sessions	HLR_01	Not Implemented
LLR_03	User can get the Attendance Status	HLR_02	Implemented
LLR_04	User can enter status input to get the Attendance Status	HLR_02	Implemented
LLR_05	User can get the user details	HLR_03	Implemented
LLR_06	User will get the details after the successfull attendance entry	HLR_03	Implemented

ID	Feature	High Level ID	Status
LLR_07	User can load different sheets	HLR_04	Implemented
LLR_08	User can also modify the existing sheets as it is dynamic	HLR_04	Implemented
LLR_09	Output file gets generated	HLR_05	Implemented
LLR_10	Multiple files can be generated with different inputs	HLR_05	Implemented

Test Plan

High Level Test Plan

ID	Description	Expected I/P	Expected O/P	Actual O/P	Type Of Test
HLTP_01	Attendance Status	User Input	SUCCESS	SUCCESS	Requirement Based
HLTP_02	User details	User Input	SUCCESS	SUCCESS	Requirement Based
HLTP_03	User load sheet	User Input	SUCCESS	SUCCESS	Requirement Based
HLTP_04	Output file generation	User Input	SUCCESS	SUCCESS	Requirement Based

Low Level Test Plan

ID	HLTP ID	Description	Expected I/P	Actual O/P	Type Of Test
LLTP_01	HLTP_01	User can get Attendance Status	SUCCESS	SUCCESS	Requirement Based
LLTP_02	HLTP_01	User can enter Status input to	SUCCESS	SUCCESS	

ID	HLTP ID	Description	Expected I/P	Actual O/P	Type Of Test
		get the Attendance Status			
LLTP_03	HLTP_02	User can get the User details	SUCCESS	SUCCESS	Requirement Based
LLTP_04	HLTP_02	User will get the details after the successful attendance	SUCCESS	SUCCESS	Requirement Based
LLTP_05	HLTP_03	User can load different sheets	SUCCESS	SUCCESS	Requirement Based
LLTP_06	HLTP_03	User can also modify the existing sheets as it is dynamic	SUCCESS	SUCCESS	Requirement Based
LLTP_07	HLTP_04	Output file gets generated	SUCCESS	SUCCESS	Requirement Based
LLTP_08	HLTP_04	Multiple files can be generated with different inputs	SUCCESS	SUCCESS	Requirement Based

Implementation and Summary

Git Link:

Link: https://github.com/alrichroshan/Attendance_Automation_Team_14

Git Dashboard

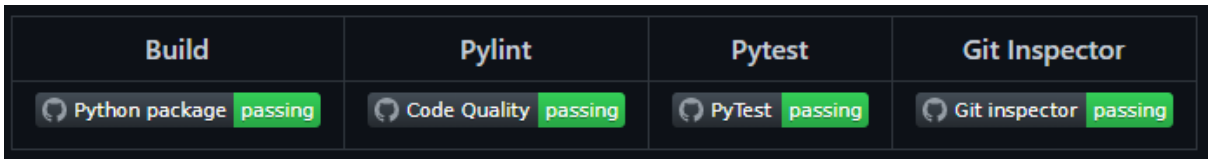

Figure 11 Git Dashboard

Git Inspector Summary

Author	Commits	Insertions	Deletions	% of changes
Alrich Roshan	4	190	1	37.97
Dileep Kumar Varadar	1	1	1	0.40
Sreenithy Thayanithy	3	53	20	14.51
Vishnu-prasath	3	2	2	0.80
alrichroshan	9	105	63	33.40
cedricxavi	2	25	3	5.57
gulamsuhail00	2	9	9	3.58
muthupbalag1310	1	5	5	1.99
vanisreekathirvel	5	8	1	1.79

Below are the number of rows from each author that have survived and are still intact in the current revision:

Author	Rows	Stability	Age	% in comments
Alrich Roshan	88	46.3	0.1	0.00
Dileep Kumar Varadar	1	100.0	0.1	0.00
Sreenithy Thayanithy	7	13.2	0.1	0.00
alrichroshan	67	63.8	0.1	0.00
cedricxavi	15	60.0	0.1	0.00
gulamsuhail00	2	22.2	0.1	0.00
vanisreekathirvel	1	12.5	0.1	0.00

Figure 12 Git Inspector Summary

Individual Contribution and Highlights

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

Role in Project Team

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

Miniproject 5 – Kia Project[Team]

Modules

1. Matlab
2. Git

Requirements

We have implemented following features

1. Adaptive Cruise Control System
2. Anti Lock Braking System
3. Automatic Transmission Control System
4. Door Locking system
5. Engine Braking System
6. Lane Assist System
7. Power Window

Adaptive Cruise Control

Adaptive cruise control (ACC) is an intelligent form of cruise control that allows vehicles to speed up and slow down automatically in order to keep pace with the traffic ahead.

ACC is also known as autonomous cruise control, active cruise control, intelligent cruise control and radar cruise control. But regardless of what it's called, it's becoming an increasingly common feature in new cars.

Design

This project was implemented using Matlab.

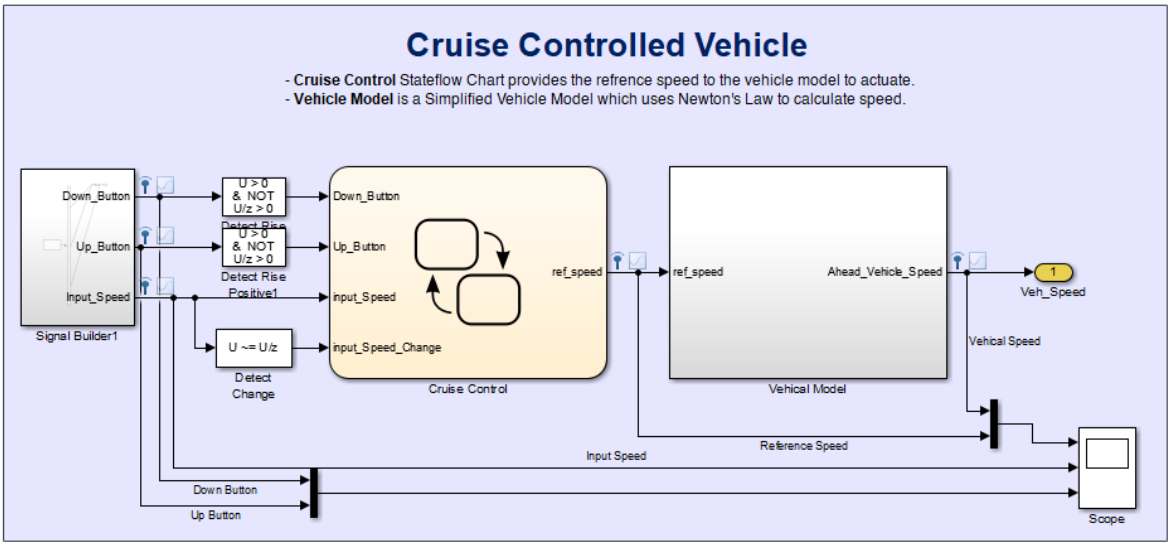

Figure 13 Circuit Diagram

Individual Contribution and Highlights

1. Improved implementation of Adaptive Cruise Control in Matlab
2. Source code management using GitHub

Role in Project Team

1.Implemetation of Adaptive cruise control system

Implementation and Summary

Git Link:

Link: https://github.com/karthikeyans99/Genesis2021_Applied_Mbd-Kia_Project_Team.git

Miniproject 6 – Wiper Control[Team]

Modules

1. C Programming
2. STM32

Requirements

4W's and 1'H

Who:

Person who is driving the vehicle can able to use the wiper system.

What:

Wipers may be powered by a variety of means, although most in use today are powered by an electric motor through a series of mechanical components, typically two 4-bar linkages in series or parallel.

Why:

1. Used to remove rain,snow from a vehicles front windows

2. To ensure the driver's safety.

Where:

1. Used in four wheelers .

2. Used in heavy vehicles.

How:

The wiper system is controlled using rain sensor, temperature sensor and SMT32 microcontroller

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 dust 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.

ID	Description
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 and dust
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,similerly the dust particals detected and wiped off.

Design

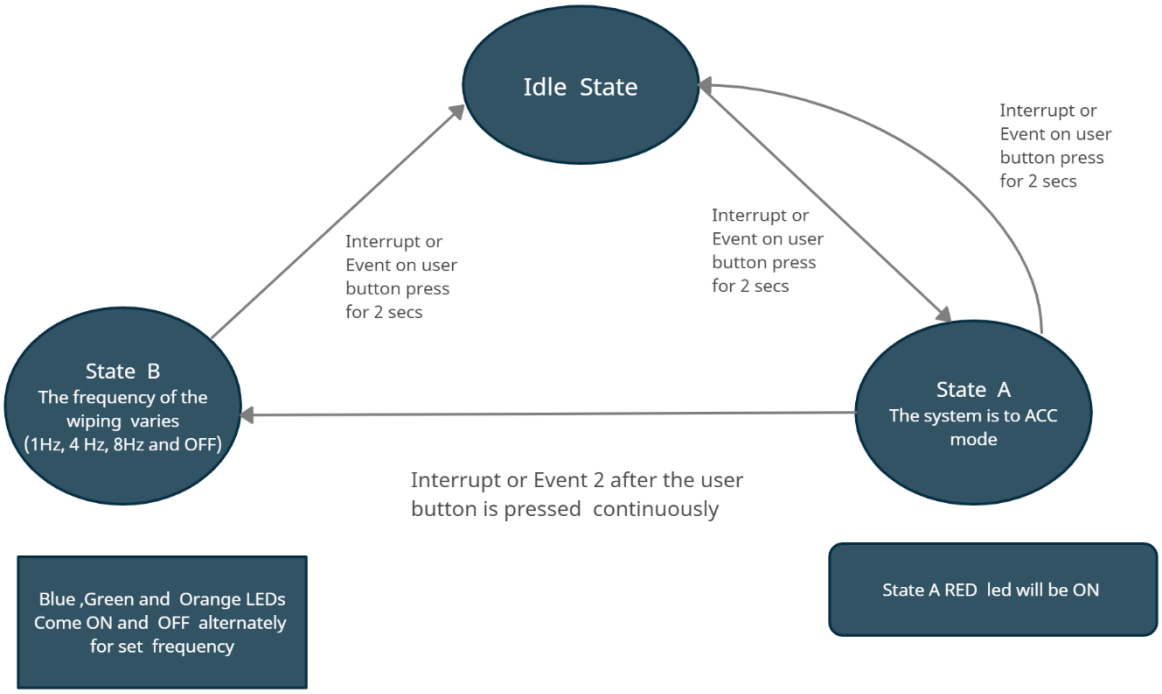

Figure 14 Structure Diagram

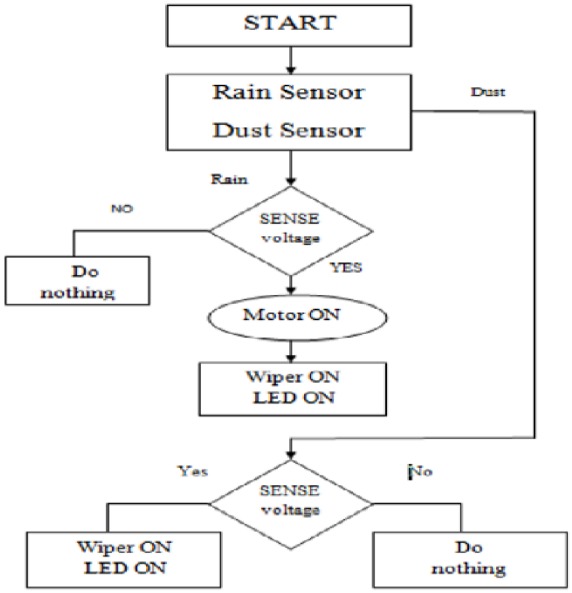

Figure 15 Behavior Diagram

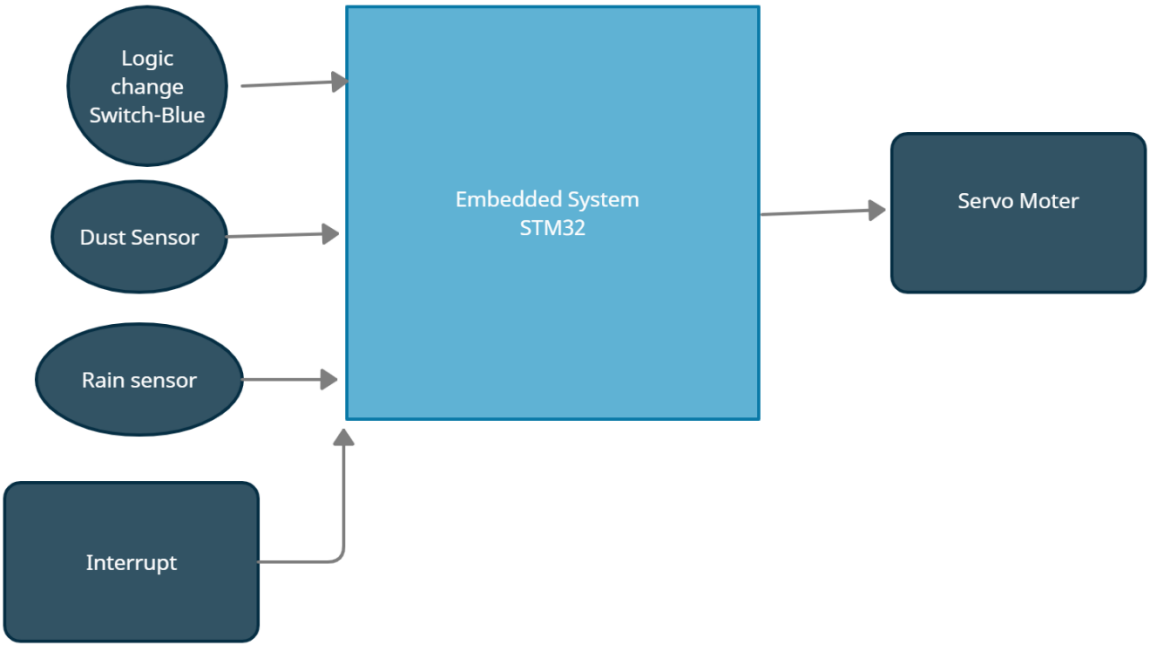

Figure 16 Block Diagram

Test Plan

High Level Test Plan

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 dust 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 Test Plan

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 and dust
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,similerly the dust particals detected and wiped off.

Implementation and Summary

Git Link:

Link: https://github.com/GENESIS-2022/MasteringMCU-Team77.git

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. Tester: Writing Testcases and testing the integrated code

Miniproject 7 – Jeep Compass Project[Team]

Modules

1. Automotive Systems
2. Git

Requirements

Door System is a type of door opening, typically hinged on its front edge, but sometimes attached by other mechanisms such as tracks, for entering and exiting a vehicle. Doors most often integrate side windows for visibility from inside the car and can be locked to secure the vehicle. The door system available in this car are,

1. Power Door Locks
2. Passive Keyless Entry
3. Automatic Unlock Doors on Exit
4. Power Window System

High Level Requirements

S.No.	Feature	Description
HLR_1	Power Door Lock	Driver or front passenger can lock or unlock the doors.
HLR_2	Passive Keyless Entry	Allows you to unlock and lock the doors to a vehicle without using a key.
HLR_3	Automatic Unlock Door on	The door locks will unlocked automatically.
HLR_4	Power Window System	All windows can accessed by switching on the passenger door trim panel.

Low Level Requirements

S.No.	Feature	Description
LLR_1	Power Door Lock	It automatically lock doors at certain speeds.
LLR_1.1	Power Door Lock	To secure all the doors at the same time.

S.No.	Feature	Description
LLR_2	Passive Keyless Entry	Doors will be lock or unlock by using a radio frequency remote keyless system.
LLR_2.1	Passive Keyless Entry	If the vehicle is unlocked by Passive Entry and no door is opened within 60 seconds, the vehicle will re-lock.
LLR_3	Automatic Unlock Door on	If the vehicle is unlocked by Passive Entry and no door is opened within 60 seconds, the vehicle will re-lock.
LLR_4	Power Window System	The windows will be open and close within 4 seconds.

Design

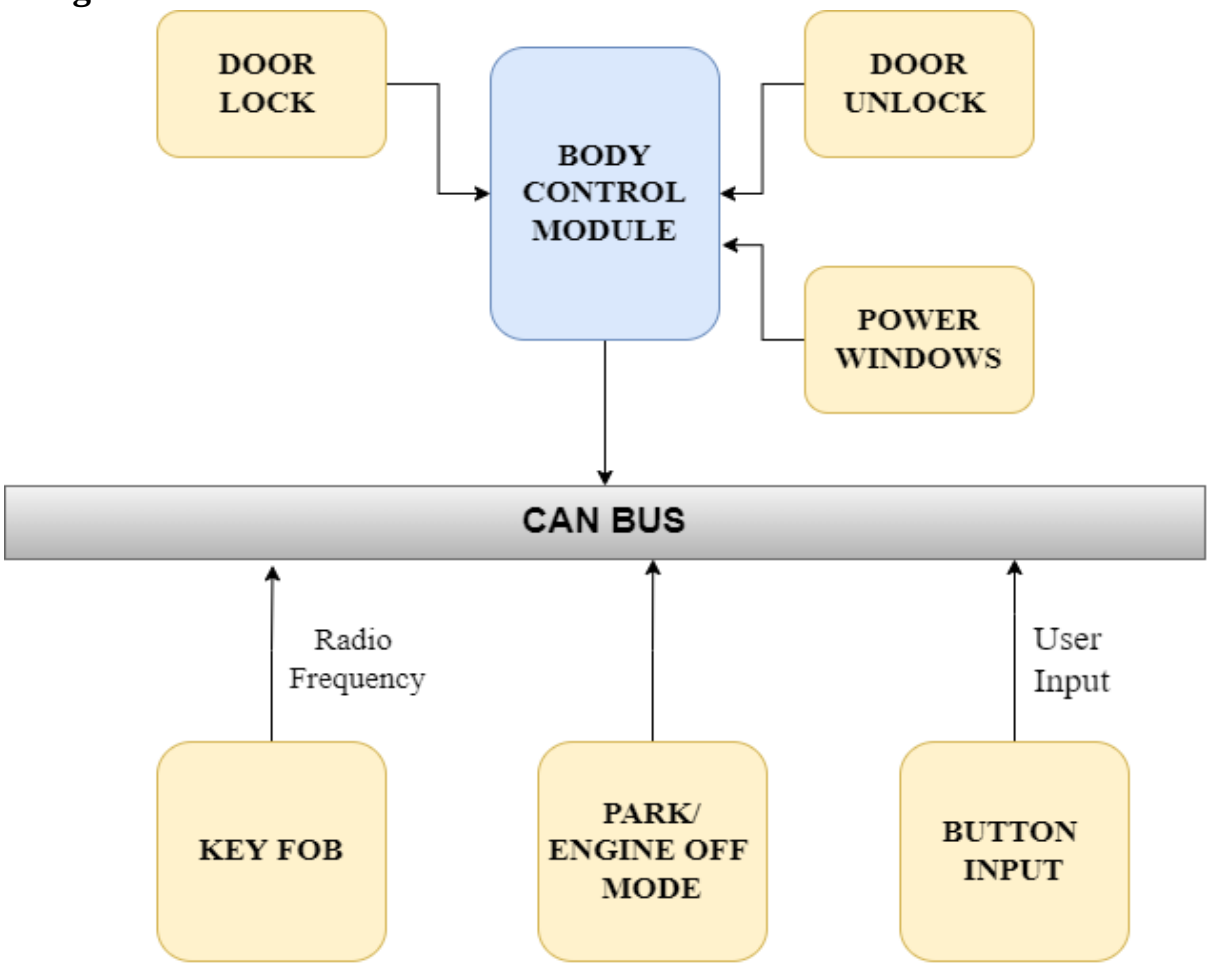

Figure 17 Structure Diagram

Implementation and Summary

Git Link:

Link: https://github.com/karthikeyans99/Jeep_Compass_Team.git

Individual Contribution and Highlights

1. Door System Case Study
2. Source code management using GitHub

Role in Project Team

1. Designer: Done Designing for Project
2. Researcher: Done case study for Door System

Miniproject 8 – EV Golf Cart[Team]

Modules

1. Matlab
2. Matlab Script

Requirements

Introduction

A golf cart or golf buggy is a small vehicle designed originally to carry two golfers and their golf clubs around a golf course or on desert trails with less effort than walking.They are generally around 4 feet (1.2 m) wide × 8 feet (2.4 m) long × 6 feet (1.8 m) high and weigh 900 pounds (410 kg) to 1,000 pounds (450 kg).Electric golf

carts work by sending power from the battery to the motor, Upon receiving electric energy from the power bank, the motor turns the wheel, and the car will start to run

The components of golf cart are:

1. Battery Pack
2. Solenoid
3. Speed Controller
4. Throttle
5. ELectric Motor
6. Electrical Accessories

Research

Reportedly, the first use of a motorized cart on a golf course was by JK Wadley of Texarkana, Texas/Arkansas, who saw a three-wheeled electric cart being used in Los Angeles to transport senior citizens to the grocery store. The first electric golf cart was custom-made in 1932, but did not gain widespread acceptance.Merle Williams of Long Beach, California was an early innovator of the electric golf cart. He started with knowledge gained from production of electric cars due to World War II gasoline rationing.They are many types of golf carts such as adaptive, customised, Solar-Powered golf carts.A golf cart community is a place where you can get from one place to other place by taking your golf cart. Some golf cart communities will have a rule that no cars are allowed, and you can only drive your cart.Major golf cart communities such as Peachtree City in Georgia and the Villages in Florida

Comparision of 2 Models

Feature	YAMAHA GOLFCART 4 SEATER	EZ GO Golf Cart S4
ELECTRICAL SYSTEM	48 Volt	56 Volt
HORSEPOWER (KW)	3.5 HP (2.6 kw) Make: HITACHI DC / 6.7 HP (5.0 kw) Make: HITACHI AC	11.7 hp (8.7 kW) Peak
BATTERIES (QTY/TYPE)	GS BV 150Z 12 VOLT X 4 PCS	Single, 56 V Li-Ion (Standard Configuration)
DRIVETRAIN	DC Shunt Wound Motor	Motor Shaft Direct Drive
BATTERY CHARGER	LT1510	56VDC, 120/230 VAC, UL/CSA, CE
OVERALL LENGTH	112.3" (2850 mm)	115 in (292 cm)
OVERALL WIDTH	47.2" (1200 mm)	49.5 in (124 cm)
OVERALL HEIGHT (W/O ROOF)	70.4" (1790 mm)	52 in (132 cm)
Wheel Base	64.6" (1640 mm)	67.5 in (171 cm)
GROUND CLEARANCE @ DIFFERENTIAL	4.3" (110 mm)	7.3 in (18.5 cm)

Cost and Features

COST	YAMAHA	EZ-GO	COST
$ 10000	Four Seater	Four Seater	$ 13699
$ 15000	Six Seater	Six Seater	$15599

Defining Our System

Feature	Specifications
ELECTRICAL SYSTEM	53 Volt
HORSEPOWER (KW)	7.3 HP(4.3 kW)
BATTERIES (QTY/TYPE)	DCM 0035 12V 35Ah Deep Cycle
DRIVETRAIN	DC Motor
OVERALL LENGTH	113.3"
OVERALL WIDTH	48.2"
Wheel Base	65.6"
GROUND CLEARANCE @ DIFFERENTIAL	6.3"

Design:

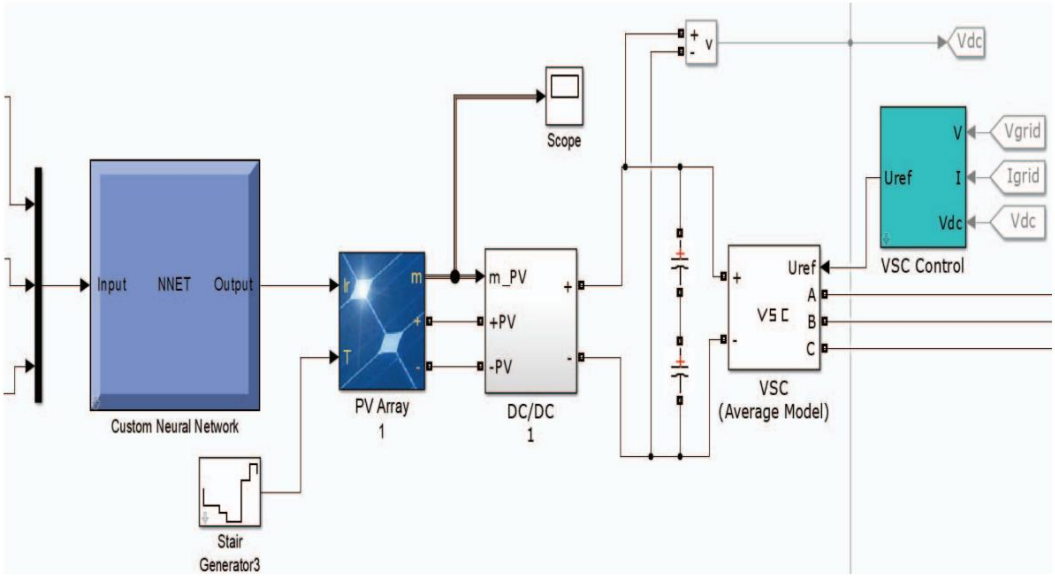

Figure 18 Circuit Diagram

Git Hub Repo

Git Link: https://github.com/SHANMUGAAPRIYANM/TEAM-1-EV_Golf-Cart.git

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 EV Golf Cart
2. Researcher: Done case study for EV Golf Cart

Miniproject 9 – Automatic Door Unlock[Individual]

Modules

1. Autosar
2. Git

Requirements

Automatic Door Unlock :

The doors will unlock automatically on vehicles with power door locks if the Automatic Unlock Doors On Exit feature is enabled and all doors are closed. If the transmission gear selector was not in PARK, then is placed in PARK.

Design

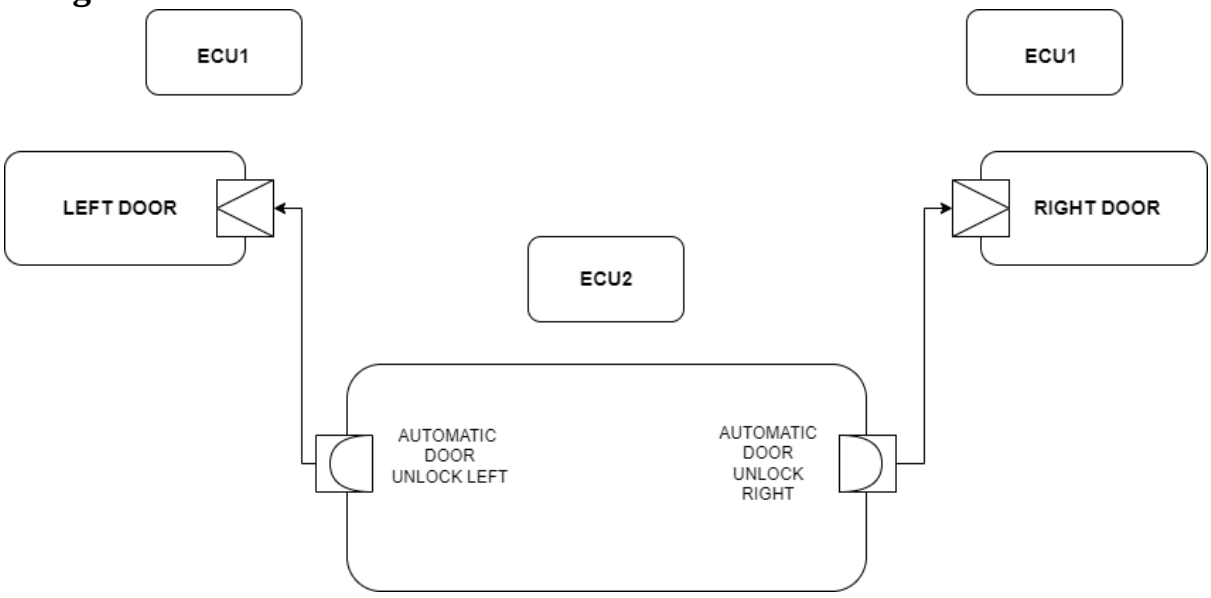

Figure 19 VFB Diagram

Implementation and Summary

Git Link:

Link: https://github.com/cedricxavi/AutomaticDoorUnlock_40020514_DPS.git

Individual Contribution and Highlights

1. Automatic Door Unlock Case Study
2. Source code management using GitHub
3. AtomicSwComponent
4. SWCInternalBehavior
5. SWCImplementation