GENESIS-Learning Outcome & Mini-project Summary Report

Details

Ver. Rel. No.	Release Date	Prepared By	Reviewed By	To Be Approved	Remarks/Revision Details
1.0	24/02/2022	Vivek Kushwaha 40021036			

Contents

Module: - C Programming on multiple platforms.

Date: - Nov 15th to 25th.

Team/Individual: - Individual.

Project topic: - Ticktacktoe Game.

Requirements: -

Introduction: -

The miniproject is developed in C Programming Language. We will be going over how we planned on implementing a Ticktacktoe game using C Programming Language and Run in Visual Studio Code using GitHub and a as many various other tools as we could testing or deployment.

About The Project:

1. The game has a basic interface that runs in the terminal.

2. The goals of this project are:

a. Practice with C Programming Language using Visual Studio Code.
b. Learn how to structure, Correct use of logic and interface files to keep code maintainable.
c. Practice of GitHub and GitHub creating branches for the different features.

SWOT Analysis: -

SWOT Analysis is a simple tool that can help you to analyse what your company does best right now, and to devise a successful strategy for the future. SWOT can also reveal areas of the business that are holding you back, or that your competitors could exploit if you don't protect yourself.

a) Strengths:

Tic Tac Toe is in itself a wonderful game for developing planning skills as children need to plan out how they can build their own line of 4 while also watching out for what their opponent is doing so, they can block their opponent from winning

b) Weaknesses:

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

c) Opportunity:

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

d) Threats:

The theft of intellectual property is one of the primary concerns for online gaming companies.

1. Data Breaches.
2. Ransomware and IABs.
3. Phishing and Stolen Accounts.

4W's and 1'H: -

* What:

This helps user to increasing to more players and no monthly subscription.

* Where:

It has to be used easily by the users.

* When:

It has to be deployed 20th of November 2021.

* Why:

I am Developing this basic C Programming Language to perform in Visual studio code in best easy manner and improve my coding skills.

* How:

I am using C programming language for Developing this simple basic Tic Tac Toe Game.

Cost and Features and Timeline: -

Cost:
No cost - Free download

5 years Ago:
Higher dimensional variations are possible.

Detail requirements: -

High Level Requirements:

ID	Description
HLR1	Users can use a web browser to obtain the information.
HLR2	When a user moves, the game page allows them to move.
HLR3	The user can see the opponent's movements in real time on the game page.
HLR4	From the landing page, the user should choose the games difficulty level & begin playing.

Low Level Requirements:

ID	Description
LLR1	Players personal details like gender, contact number.
LLR2	When one player gets three symbols in a row the game should be over.
LLR3	After the game the user sees the results.

DESIGN: -

To begin with we had figure out the basic logic of the game and how we were going to implement that. The following list describe the core elements we needed to implement.

* The board:
  A 3x3 square stored in a two-dimensional array

* Players:
  1. Two players, O and X
  2. Array to keep track of score
  3. Players can be both X or O, depending on game
  4. Player makes a move
  5. 5Take a command from keyboard/mouse and change the board array
  6. Check if that move is valid
  7. Check if player won the game or if it's a draw

* The user interface

  1. Display the board
  2. Ask for input / get input
  3. Give input to controller and make the move.

- Structural Diagram:

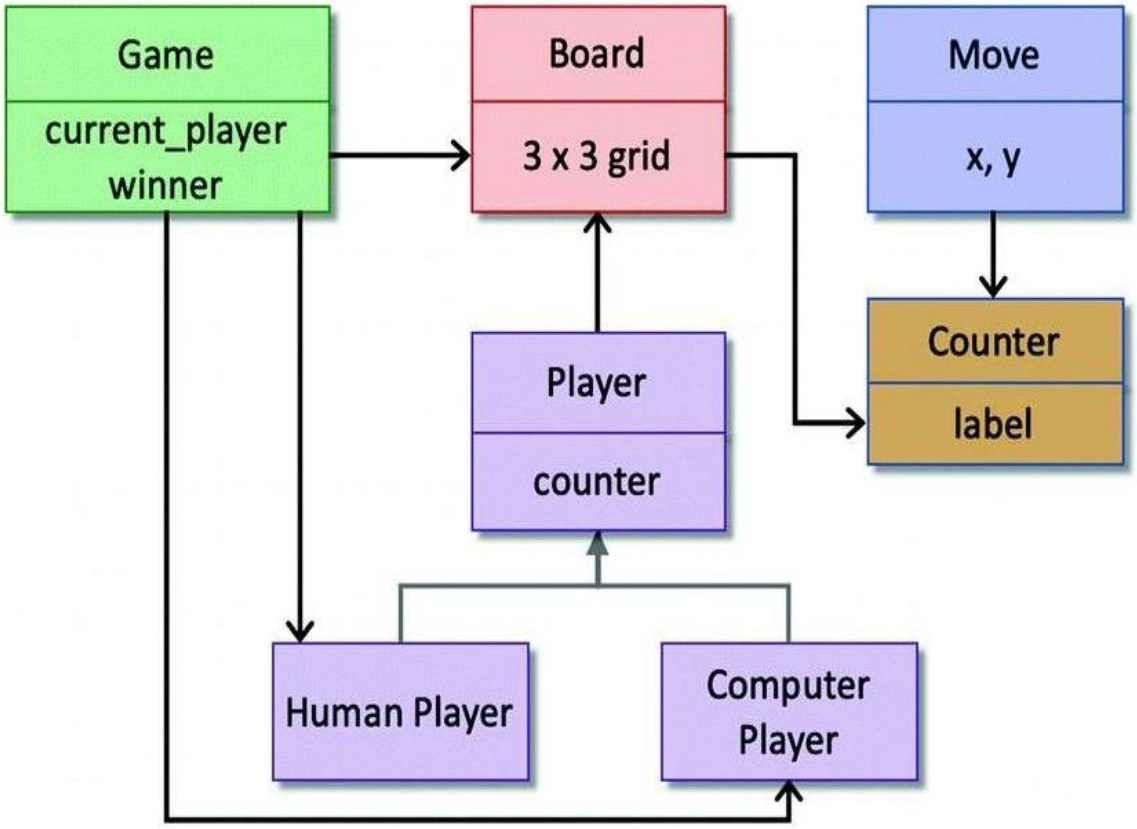


Fig 1. Structural Diagram.

- ACTIVITY DIAGRAM:

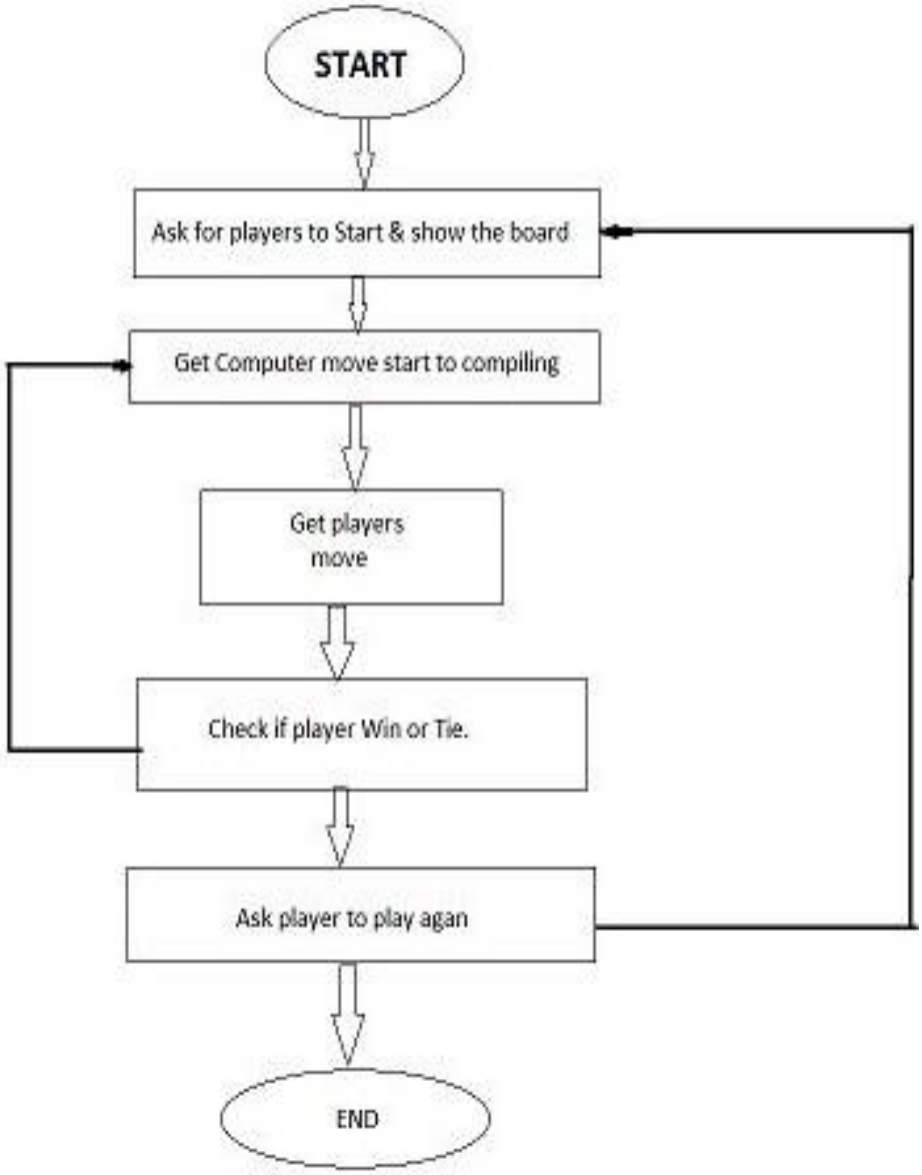


Fig 2. Activity Diagram.

Implementation: -

C code: -

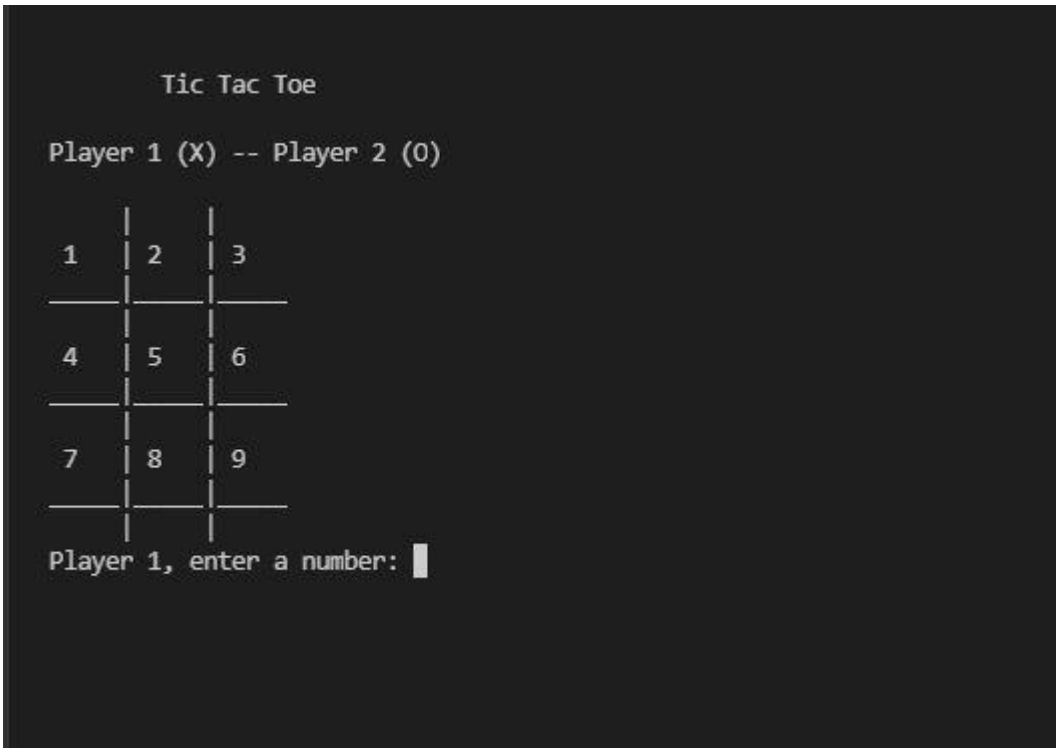
```
#include
<stdio.h>
            #include <stdlib.h>
```

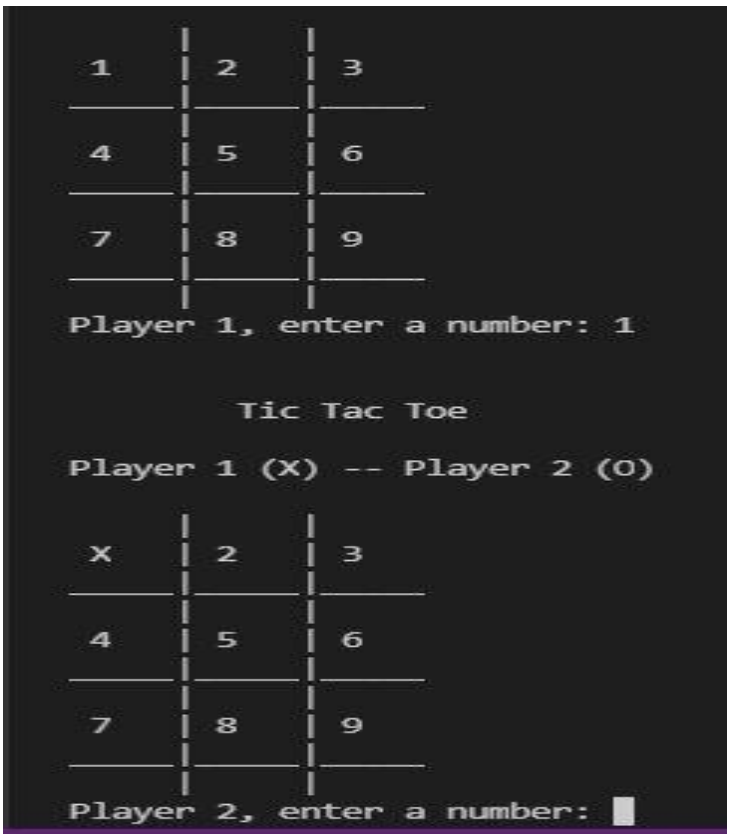
```
#include <ctype.h>
#include <time.h>
char box[10]={"0","1","2","3","4","5","6","7","8","9"};
void creating_board();
void marking_board(int, char);
int check_for_win();
int main()
{
    int choice,player=1,i;
    char mark;
    do{
        creating_board();
        player= (player % 2) ? 1: 2;

        printf("Player %d, enter a number: ",player);
        scanf("%d",&choice);

        mark = (player==1) ? "X" : "O";
        marking_board(choice,mark);

        i=check_for_win();
        player++;

    }while(i == -1);

    creating_board();

    if(i==1)
        printf("Player %d  you have won the game",--player);
    else
        printf("<-------Draw -------->");


    return 0;
}
void creating_board()
{

    printf("\n\n\tTic Tac Toe\n\n");
    printf("Player 1 (X) -- Player 2 (O)\n\n");
    printf("     |     |     \n");
    printf(" %c   | %c   | %c   \n",box[1],box[2],box[3]);
    printf("_____|_____|_____\n");
    printf("     |     |     \n");
    printf(" %c   | %c   | %c   \n",box[4],box[5],box[6]);
```

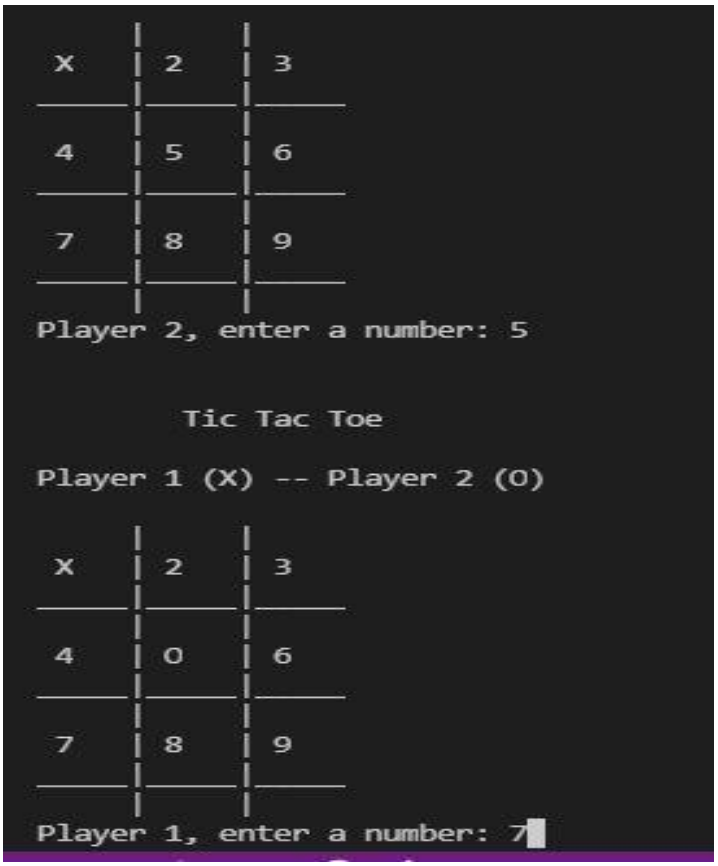
```
    printf("     |_____|_____\n");
    printf("     |     |      \n");
    printf(" %c   | %c   | %c   \n",box[7],box[8],box[9]);
    printf("     |_____|_____\n");
    printf("     |     |      \n");


}
void marking_board( int choice, char mark)
{
    if(choice==1 && box[1]=='1')
        box[1]=mark;
    else if(choice==2 && box[2]=='2')
        box[2]=mark;
    else if(choice==3 && box[3]=='3')
        box[3]=mark;
    else if(choice==4 && box[4]=='4')
        box[4]=mark;
    else if (choice==5 && box[5]=='5')
        box[5]=mark;
    else if (choice==6 && box[6]=='6')
        box[6]=mark;
    else if (choice==7 && box[7]=='7')
        box[7]=mark;
    else if (choice==8 && box[8]=='8')
        box[8]=mark;
    else if (choice==9 && box[9]=='9')
        box[9]=mark;
    else
    {
        printf("Invalid move");
    }
}

int check_for_win()
{
    if(box[1]==box[2] && box[2]==box[3])
        return 1;
    else if(box[4]==box[5] && box[5]==box[6])
        return 1;                                   // horizontal match
    else if (box[7]==box[8] && box[8]==box[9])
        return 1;

    else if (box[1]==box[4] && box[4]==box[7])
        return 1;
```

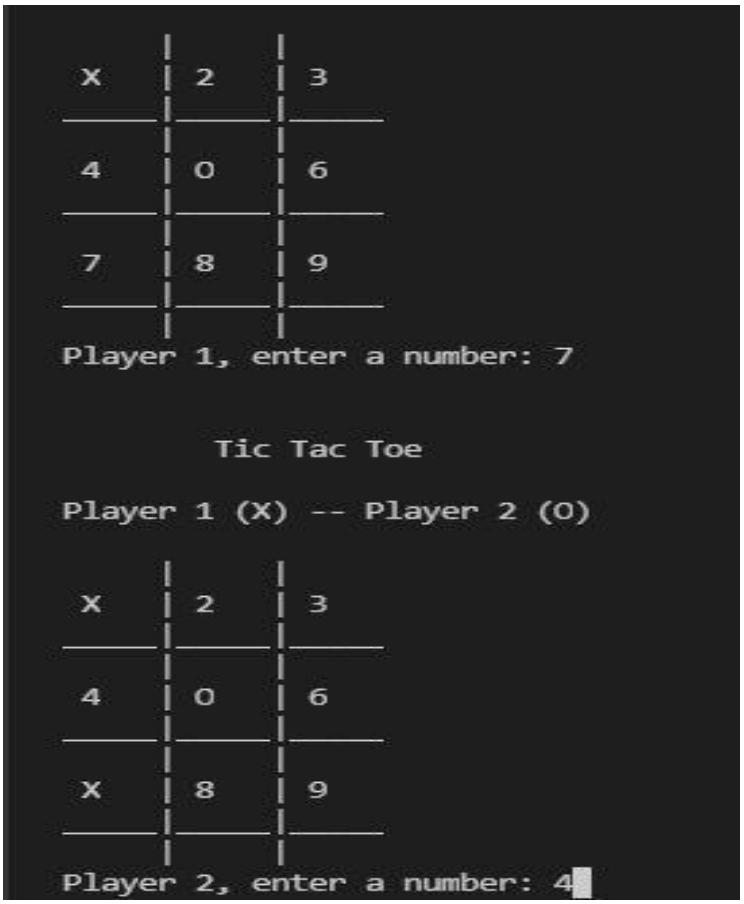
```
    else if(box[2]==box[5] && box[5]==box[8])          // vertical match
        return 1;
    else if(box[3]==box[6] && box[6]==box[9])
        return 1;


    else if(box[1]==box[5] && box[5]==box[9])
        return 1;
    else if(box[3]==box[5] && box[5]==box[7])          //diagonal match
        return 1;

     else if(box[1]!= "1" && box[2]!= "2" && box[3]!= "3" && box[4]!= "4"&&
 box[5]!= "5" && box[6]!= "6"&& box[7]!= "7" && box[8]!= "8" && box[9]!="9")
//no match
        return 0;

    else
        return -1;


}
```

Output: -




1 | 2 | 3
4 | 5 | 6
7 | 8 | 9
Player 1, enter a number: 1
Tic Tac Toe
Player 1 (X) -- Player 2 (O)
X | 2 | 3
4 | 5 | 6
7 | 8 | 9
Player 2, enter a number:


X | 2 | 3
4 | 5 | 6
7 | 8 | 9
Player 2, enter a number: 5
Tic Tac Toe
Player 1 (X) -- Player 2 (O)
X | 2 | 3
4 | O | 6
7 | 8 | 9
Player 1, enter a number: 7


X | 2 | 3
4 | O | 6
7 | 8 | 9
Player 1, enter a number: 7
Tic Tac Toe
Player 1 (X) -- Player 2 (O)
X | 2 | 3
4 | O | 6
X | 8 | 9
Player 2, enter a number: 4

Certification Done: -

1. SOLO Learn Certification.
2. Linux Certification.
3. GitHub Learning Certification.

Module: - Essentials of Embedded System.

Date: - Nov 26th to 2nd.

Team/Individual: - Individual.

Project topic: - Ultrasonic Sound Sensor.

Requirements: -

Introduction: -

Ultrasonic Sound Sensor with ATmega328 Microprocessor.

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

Research: -
The requirements for the program to run or for the code to be in effect are basic and a great solution for the detection of clear objects.

Features Hardware and Software: -

a) HARDWARE: -
1] SimulIDE:
- SimulIDE provides AVR, Arduino and PIC microcontrollers that can be accessed just like other components.
- Features like gpsim and simavr allow you to use PIC and AVR microcontrollers, respectively.

2] AVR:
- An automatic voltage regulator (AVR) is an electronic device that maintains a constant voltage level to electrical equipment on the same load.
- The AVR regulates voltage variations to deliver constant, reliable power supply.

b) SOFTWARE: -
1] ATmega328:

- ATmega328 is commonly used in many projects and autonomous systems where a simple, low-powered, low-cost micro-controller is needed
- Perhaps the most common implementation of this chip is on the popular Arduino development platform.

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

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

Defining Our System: -

Ultrasonic sensors work by sending out a sound wave at a frequency above the range of human hearing. The transducer of the sensor acts as a microphone to receive and send the ultrasonic sound. Our ultrasonic sensors, like many others, use a single transducer to send a pulse and to receive the echo

SWOT Analysis: -

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

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

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

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

4W's and 1'H: -

- What:

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

- Where:

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

- When:

In 1959, Sato Mura created an ultrasonic flowmeter that used doppler technology.

- Why:

I am Developing this project for easily measure the distance between objects

- How:

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

Detail requirements: -

High Level Requirements:

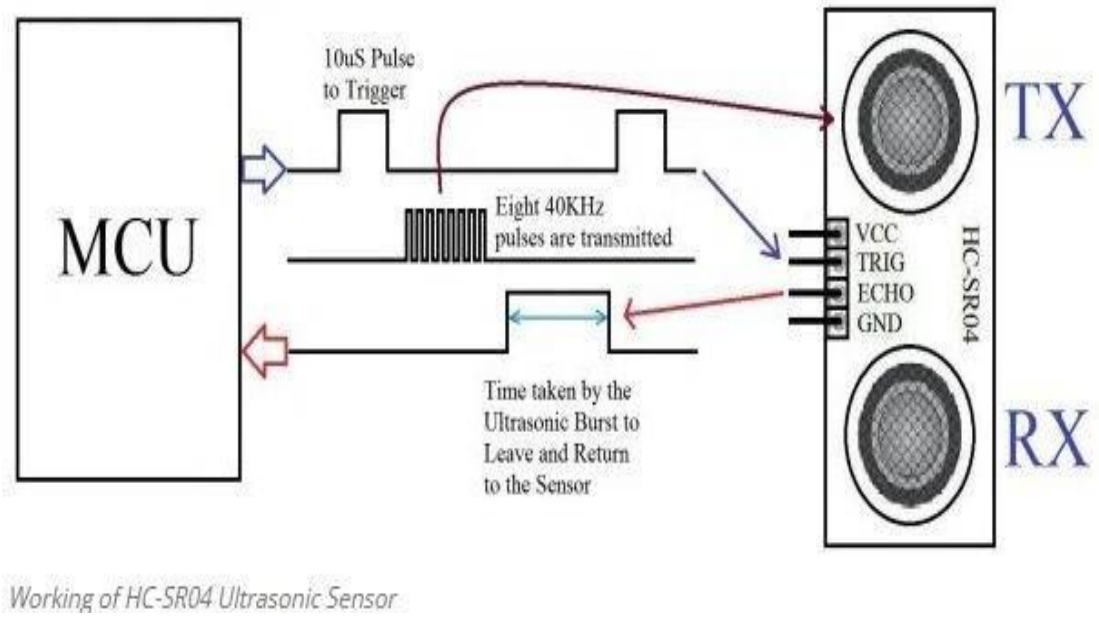
ID	Description
HLR1	Used to avoid and detect obstacles with robots like biped robot, obstacle avoider robot, path finding robot etc.
HLR2	Used to avoid and detect obstacles with robots like biped robot, obstacle avoider robot, path finding robot etc.
HLR3	Depth of certain places like wells, pits etc can be measured since the waves can penetrate through water

Low Level Requirements:

ID	Description
LLR1	Measuring Angle: 30 degrees.
LLR2	Trigger Input Pulse width: 10uS TTL pulse
LLR3	Depth of certain places like wells, pits etc can be measured since the waves can penetrate through water

DESIGN: -

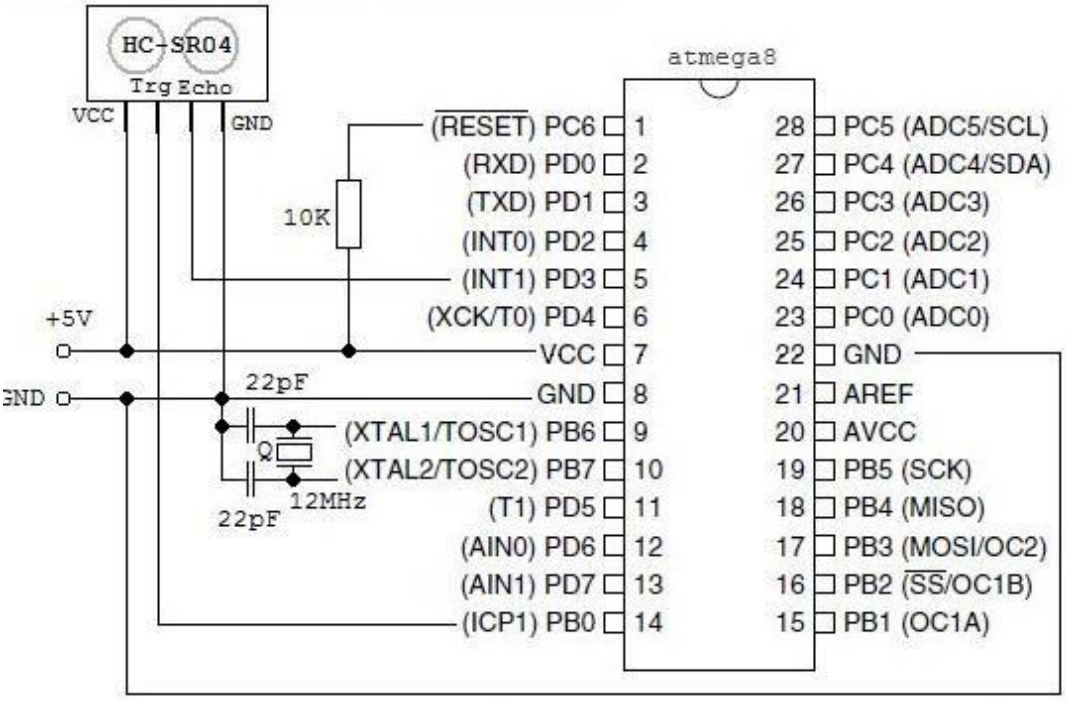
Structural Diagram:



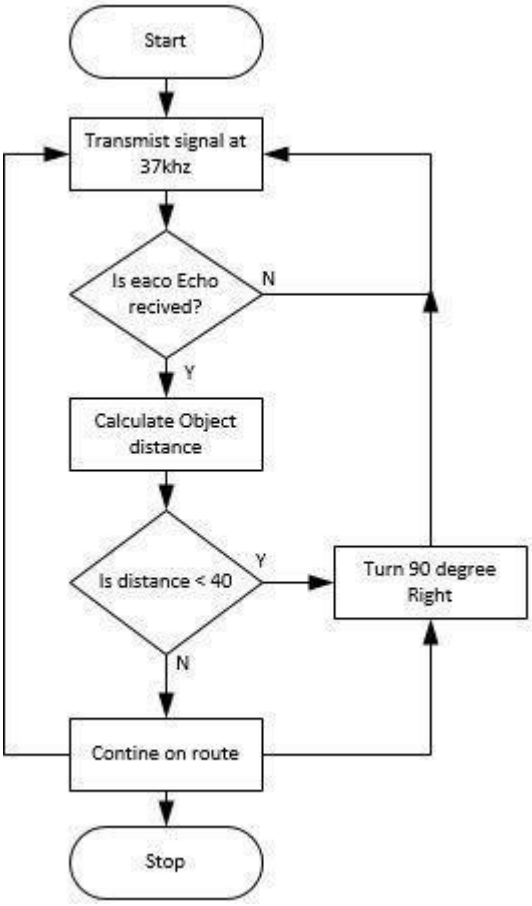
Working of HC-SR04 Ultrasonic Sensor

Block Diagram:

Circuit Diagram:



Behavioural of Project:



Implementation & Test Plan: -

Obstacle Detection:

How: Our implementation for this step requires multiple steps.

Step 1: Find a distance value between each pair of sensors. To test the distance value, we may use the numbers we see for the height and length, as well as the Pythagorean Theorem.

Step 2: Check the angle found between each pair of sensors using the distance value initially found.

Step 3: Using these values, determine what each angle should approximately be to detect different types of obstacles.

Step 4: Detect the obstacles.

Testing cases:

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

RMS error: Root mean square error between actual and sensing distance.
SD: Standard deviation of the RMS errors.

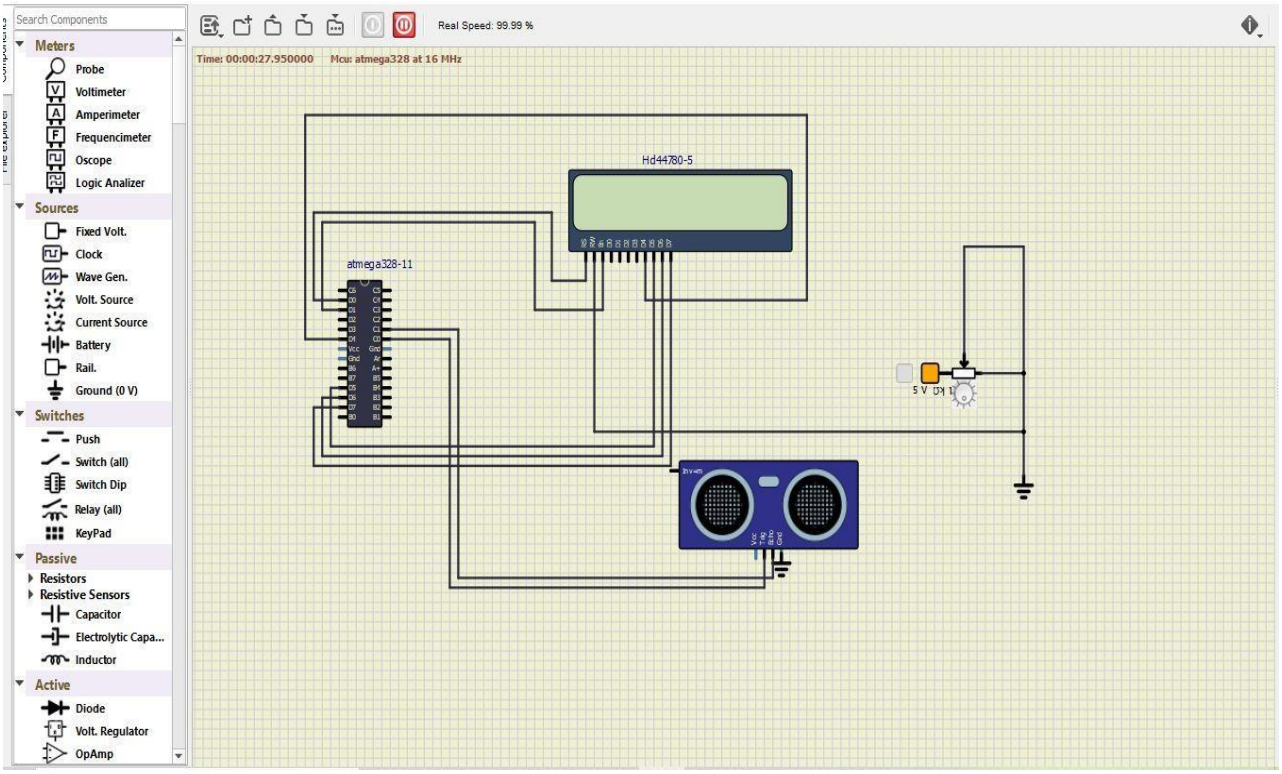
Communication Protocols:

- Wired

In this project we use UART communication protocol.

- UART - TX & RX (2 devices)

1. 2 Wire
2. Individual clocks used by the both parties
3. Standard speed (9600, 115200) - Baud rate
4. Parity



Output: -

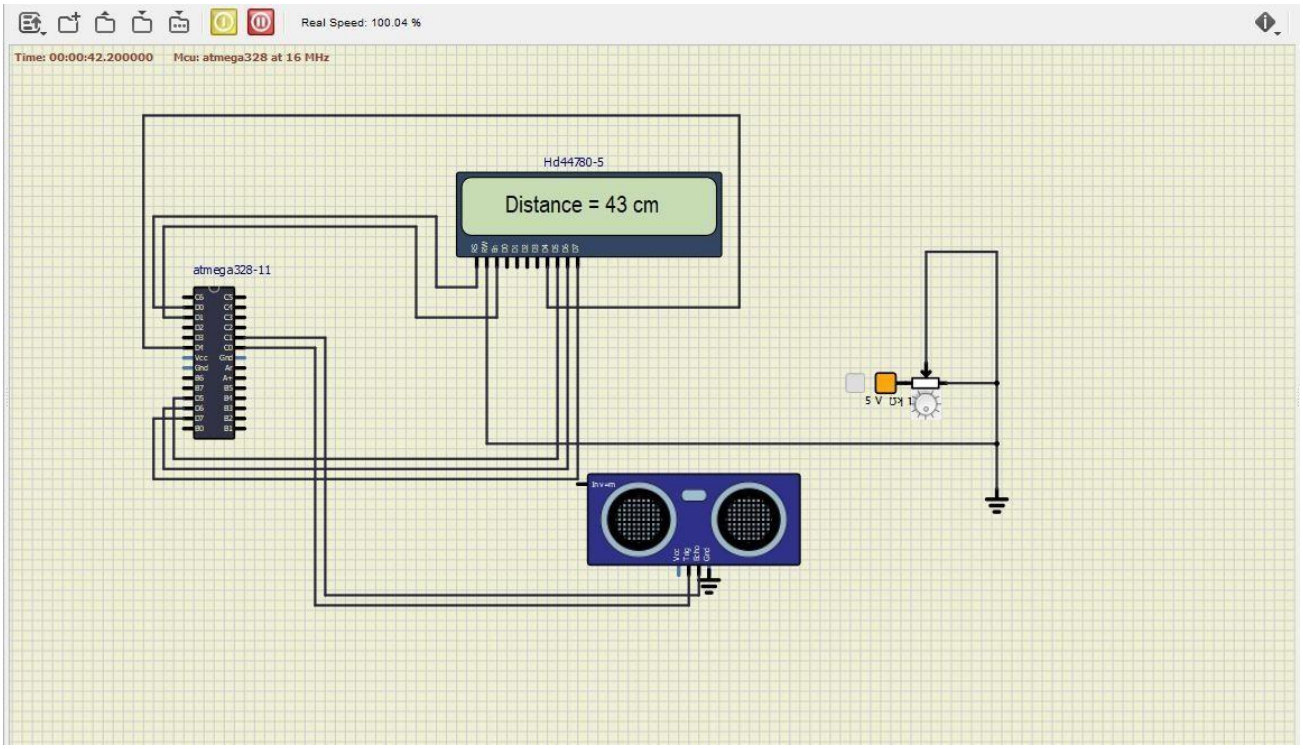
Output: As we had steps for each test, we will again make steps for the expected outputs:

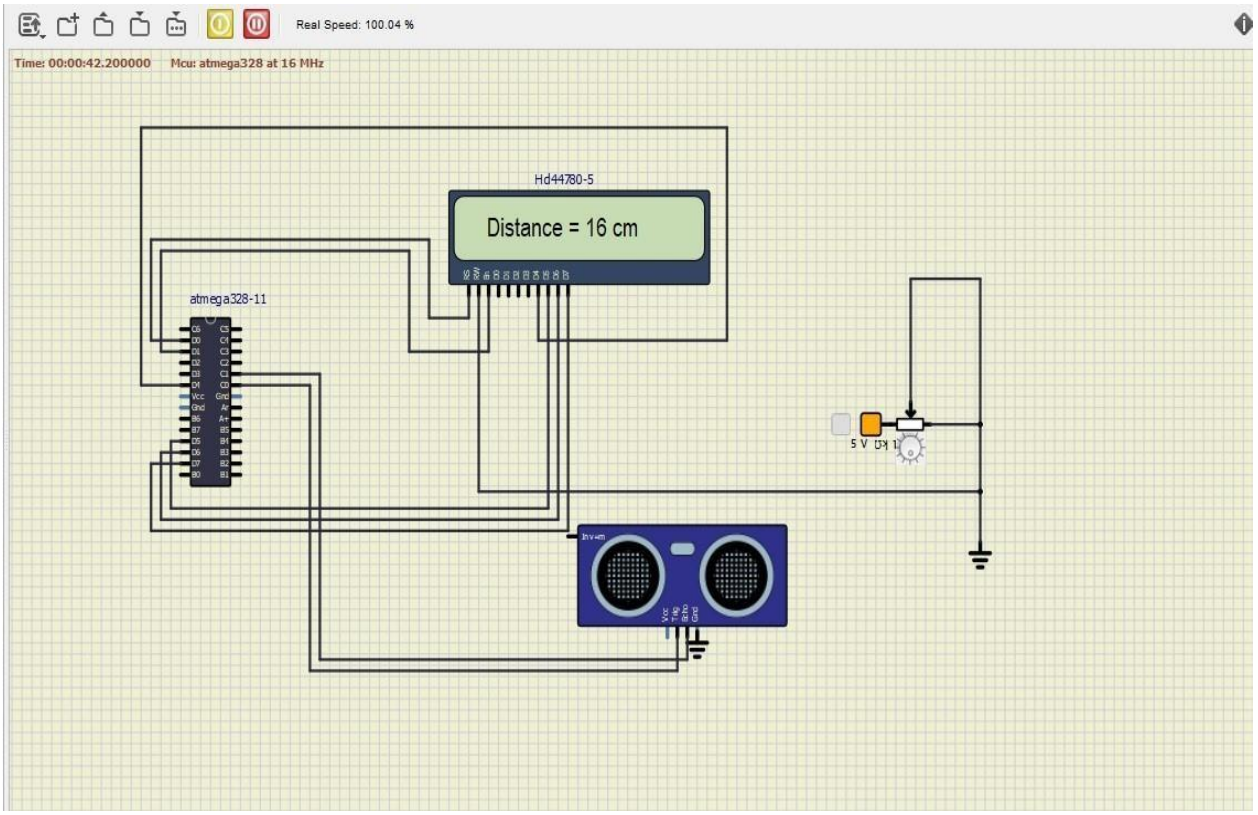
Step 1: Compare the outputted (through serial) value for the hypotenuse to the Pythagorean calculated value. We expect them to be the same.

Step 2: Using the same technique as step 1 except calculating the angle, we should see the same value for this calculation as well.

Step 3: The values and outputs for the "obstacle detected" will be constantly checked and rechecked to make sure the angles determine the correct obstacle.

Step4: Adding Audio to the Ultrasonic Sensors.




Real Speed: 100.04 %
Time: 00:00:42.200000
Mcu: atmega328 at 16 MHz
Hd44780-5
Distance = 16 cm
atmega328-11
5 V

Module: - Applied SDLC and Software Testing.

Date: - Dec 10th to 16th.

Team/Individual: - Team.

Project topic: - Food court billing system.

Aim: -

The main aim of food court billing application is to provide clean and fresh food to the students/employees of the organization. In many organizations, entire mess management and billing calculations are done manually till date. It is very time consuming and increases the chances of performing calculation mistakes. It would be possible to do the same work within a short period of time and without using much efforts and manpower if there existed a software for the same. Thus, there arises a need to create a software for the same. Such a software would make the entire Mess related management an automated system. The software is not only restricted to food items and their billing manipulations, but handling the information of the cadets seeking training in the PTC is also possible in the software. Thus, such a combination in a single software is of great benefits.

Requirements

Introduction: -

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

Features:

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

SWOT analysis:

a) Strength:

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

b) Weakness:

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

c) Opportunity:

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

d) Threat:

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

4W's and 1 H's

Who:

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

What:

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

When:

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

Where:

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

How:

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

Detail requirements: -
High Level Requirements

ID	Description
HLR1	Customer should be able to add item via item.
HLR2	Customer should be able to search items from menu function.
HLR3	Customer should be able to see their order on display function.
HLR4	Customer should able to edit their orders.

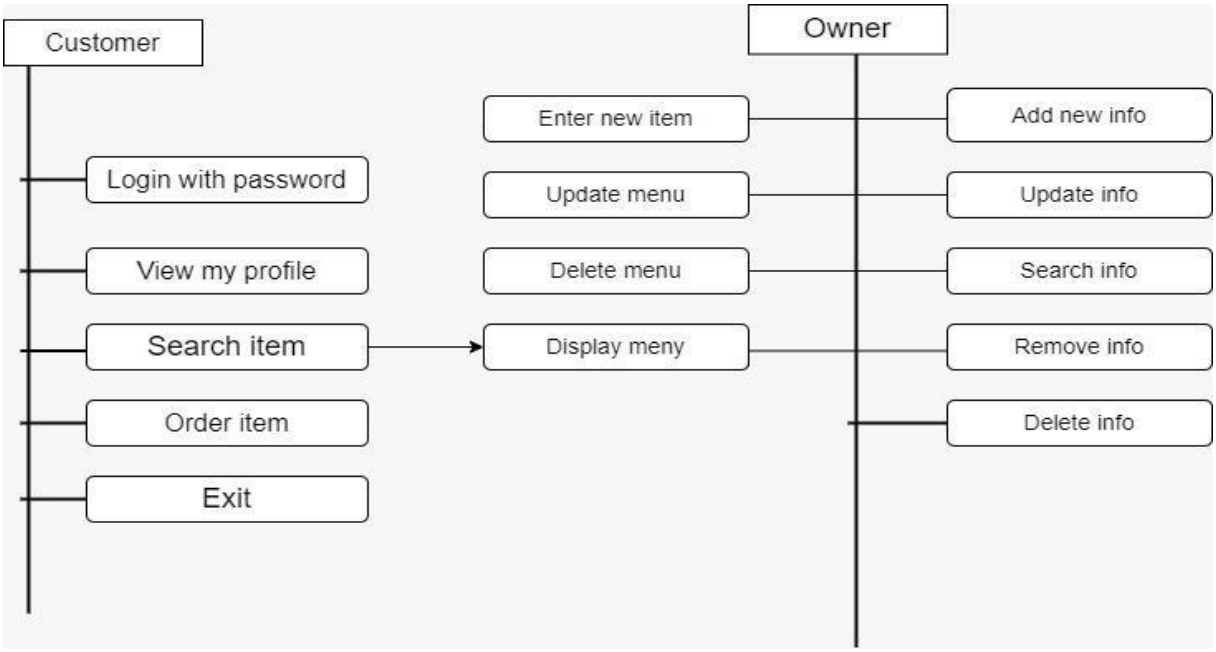
ID	Description
HLR5	Customer should able to search item via name or item code.
HLR6	Application should able to do the all calculation that are required to generate bill amount.
HLR7	Customer should be able to delete the particular item from ordered list.

Low Level Requirements

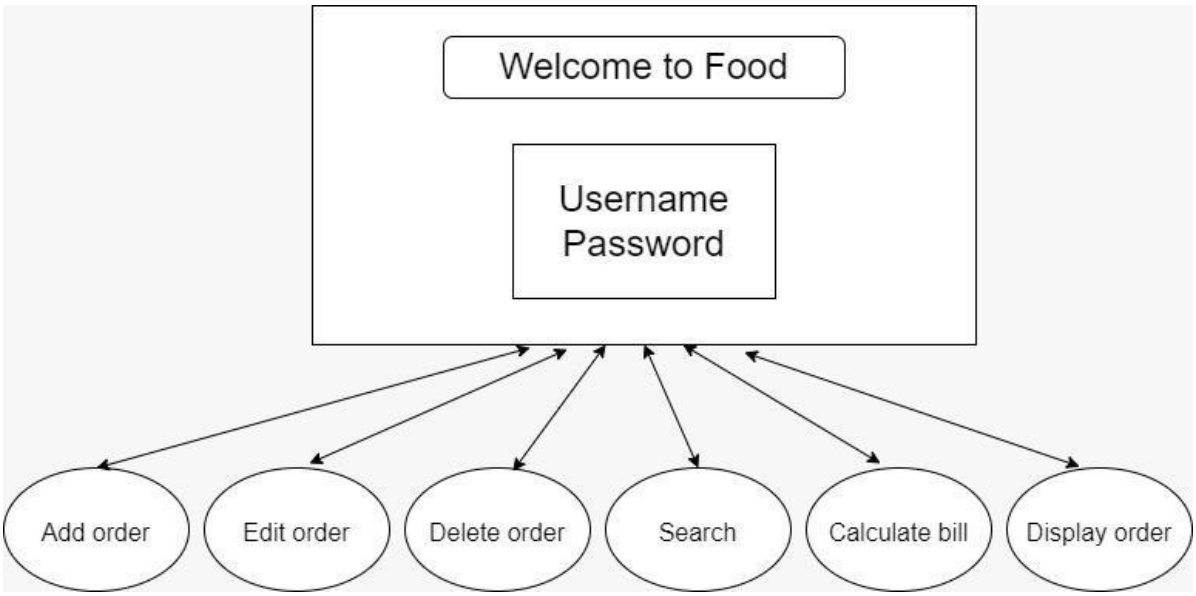
ID	Description
LLR1	Login Page off Food Court.
LLR2	Enter user and password.
LLR3	Newly added details should be display.
LLR4	Item name, quantity, rate should be removed.
LLR5	Item name, item number and item rate should be there while generating bill.
LLR6	Application should return exact final bill.

Design

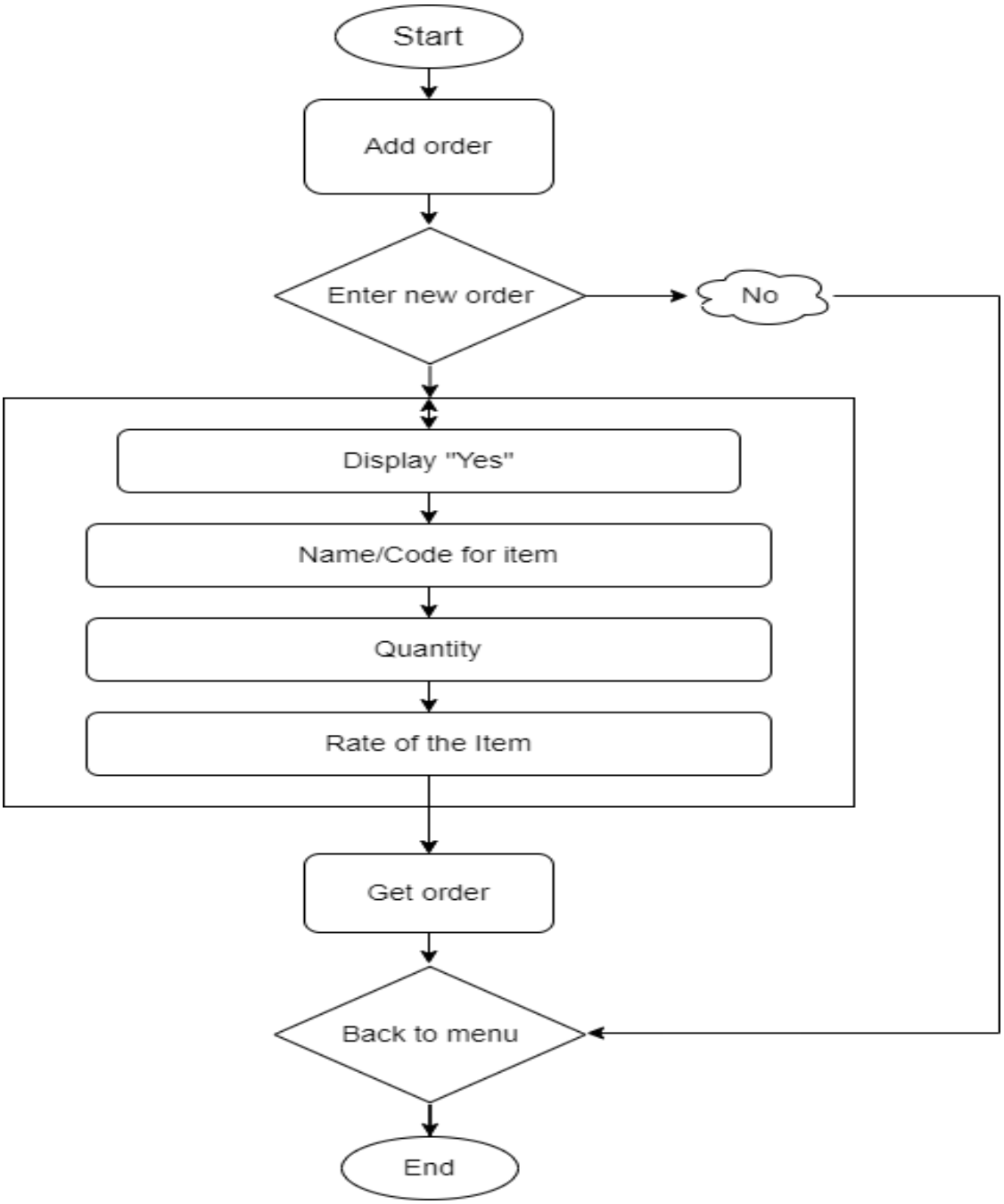
Block diagram:



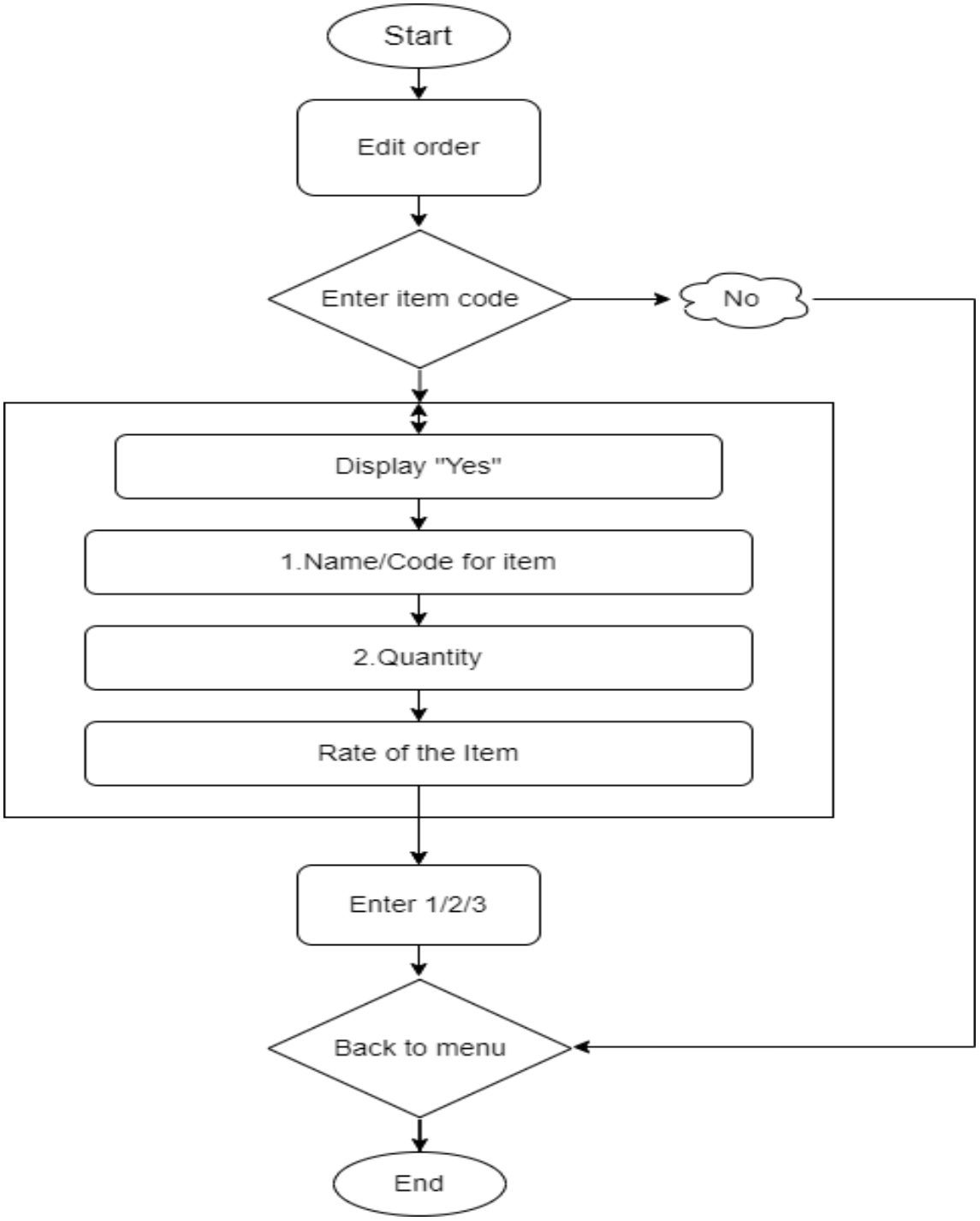
Structural diagram:



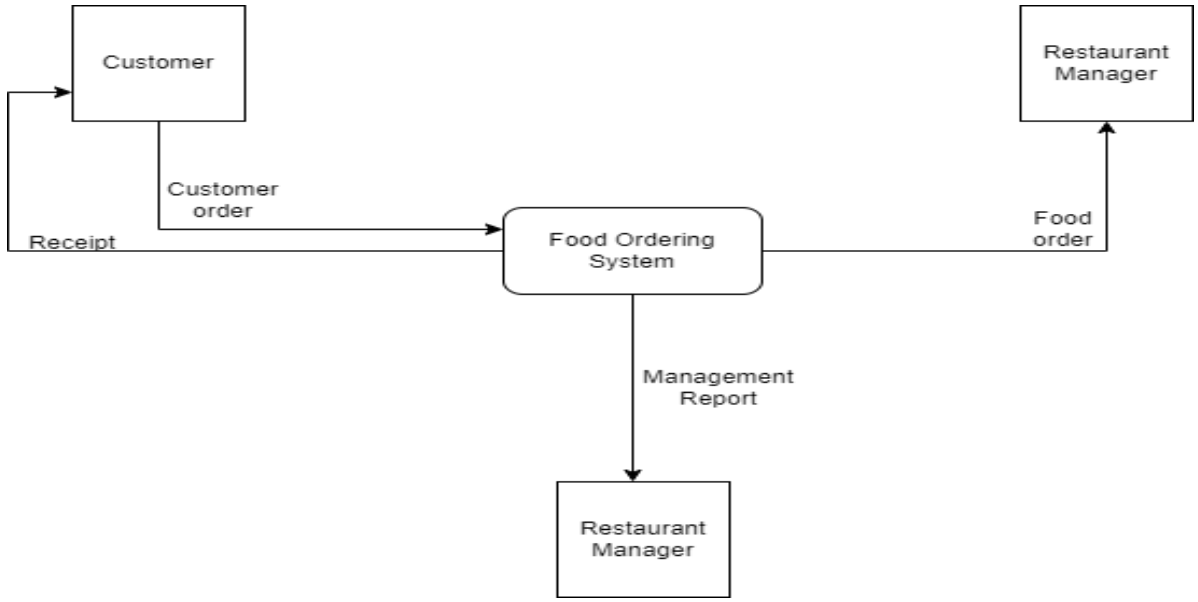
Behavioural diagram: Flow chart: 1



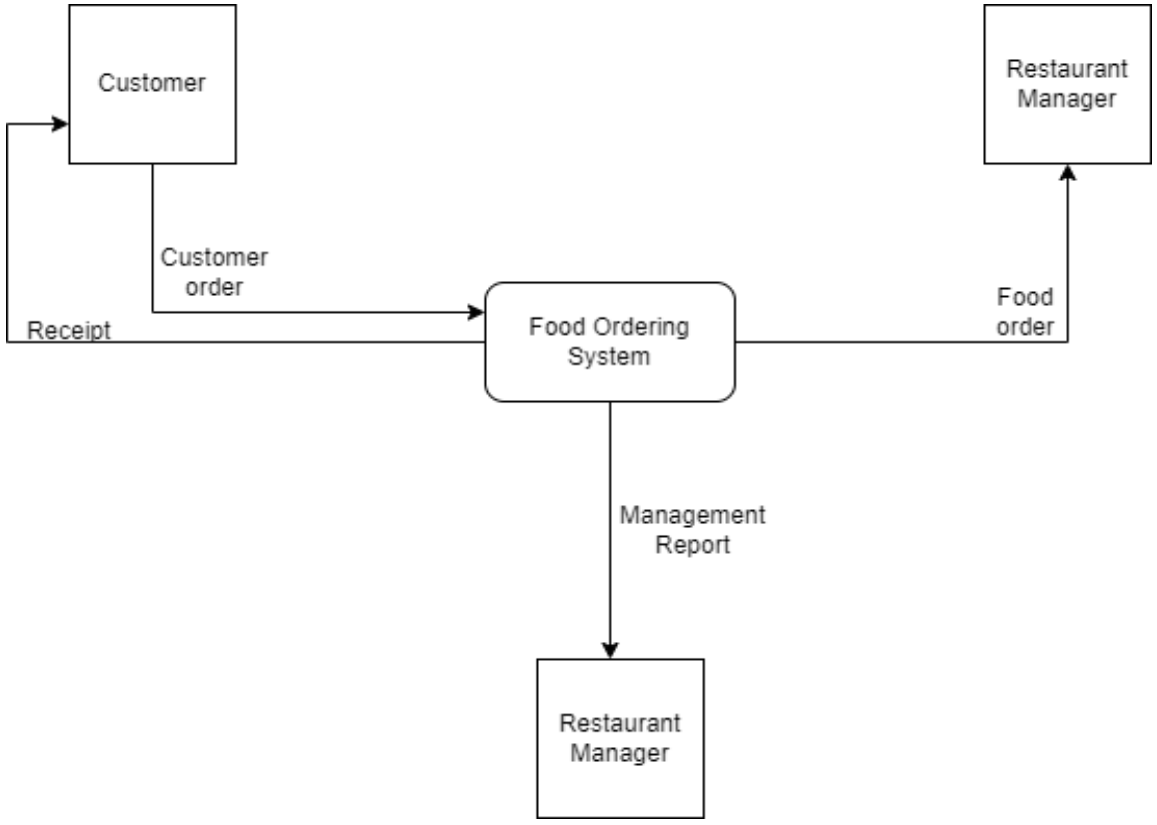
Flow chart: 2


Start
Edit order
Enter item code
No
Display "Yes"
1.Name/Code for item
2.Quantity
Rate of the Item
Enter 1/2/3
Back to menu
End

High level diagram:



Low level diagram:



Test Plan:

High Level Test Plan

TEST_ID	Description	Expected I/P	Expected O/P	Actual O/P	Type Of Test
HLR_1	Login to system	Provide proper Username with character length of 10	entered username	entered username	Requirement Based
HLR_2	Login to system	Provide proper password with character length of 10	Login successful	Login successful	Requirement Based
HLR_3	Providing items that you want to add	User Choice	Added Successfully	Added Successfully	Requirement Based
HLR_4	Display the menu	----	Added Items is Displayed	Added Items is Displayed	Requirement Based
HLR_5	Bill Calculation	Choice	Customer's Bill	Customer’s Bill	Requirement Based

Low Level Test Plan

TEST_ ID	Description	Expected I/P	Expected O/P	Actual O/P	Type Of Test
LLR_1	Login to system	username and password is Incorrect (in case numbers or exceed length)	Login is Unsuccessful	Login is Unsuccessful	Requirement Based
LLR_2	Edit Item	Provide Item code	Item Edited	Item Edited	Requirement Based
LLR_3	Delete Item	Provide Item	Item Deleted	Item Deleted	Requirement Based
LLR_4	Search Item	Provide Item Code	Item Searched	Item Searched	Requirement Based
LLR_5	exit operation	----	Exit Successfully	Exit Successfully	Requirement Based

Summary

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

Daily Scrum Stand Up Meeting Records

Date: Dec 11,2021

Tasks Completed: -
* Decided the project.
* Run the code for Add and Edit order successfully.
* Doubt solving.

Tasks Planned for Future:
* Integrate project.

Challenges Faced:
* Run the Code
* Swelling the error

Date: Dec 12,2021

* All members have ben searching more information about project. (Google, you tub, Pass out college students)

Date: Dec 13,2021

Tasks Completed:
* Code almost done!
* Block diagram
* Structural diagram

Tasks Planned for Future:
* Chalked the High-Level Requirements
* Low Level Requirements

Challenges Faced:
* Correcting the sequence diagram!
* Members are not connected to each other!
* A new member has joined our group- tried to helping out!

Date: Dec 14,2021

Tasks Completed:
* Rechecking all required reports!
* Few errors in during the dot.c code, all issues resolved till now!

Tasks Planned for Future:
* Trying to upload all files and trying to resolve the errors.
* Report making.
* Working on badges.

Challenges Faced:
* Errors occurred in make.c file.
* None so far.

Date: Dec15,2021

Tasks Completed:
* Uploaded the required files.
* Uploaded new design diagram.

Tasks Planned for Future:
* The project can be used for people, so test people knowledge.
* New feature added in the code.
* Add more diagram.

Challenges Faced:
* New feature added when the lots of error is creating.

Date: Dec16,2021

Tasks Completed:
* Uploaded new 2 design diagram.
* Update Homepage.

Git Link:

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

References:

- http://www.organizationaldynamics.upenn.edu/system/files/Ac

Individual Contribution and Highlights

- Requirements
- Folder structure
- Implementation (Add Order in System)
- Created Unity File

Module: - OOPS with Python.

Date: - Dec 17th to 23rd.

Team/Individual: Team.

Project topic: - Calendar Automation

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 (Colour 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 month wise 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
- 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. Architecture (Sequence)
2. Implementation
3. Learning resources
4. Folder structure
5. Report

Module: - Applied Model Based Design Module.

Date: - Dec 27th to 31st.

Team/Individual: - Team.

Project topic: - Anti lock braking system.

Introduction

• ABS prevents locking of wheels during braking.
• During severe braking or on slippery surfaces, wheels approach lockup. At that time, ABS takes over.
• ABS modulates the brake line pressure independent of the pedal force, to bring the wheel speed back to the slip level range that is necessary for optimal braking performance.

Modeling: -

The friction coefficient between the tire and the road surface, mu, is an empirical function of slip, known as the mu-slip curve. We created mu-slip curves by passing MATLAB variables into the block diagram using a Simulink lookup table. The model multiplies the friction coefficient, mu, by the weight on the wheel, W, to yield the frictional force, acting on the circumference of the tire. Ff is divided by the vehicle mass to produce the vehicle deceleration, which the model integrates to obtain vehicle velocity.
In this model, we used an ideal anti-lock braking controller, We set the desired slip to the value of slip at which the mu-slip curve reaches a peak value, this being the optimum value for minimum braking distance (see note below.).

Mathematical Model: -

• The friction coefficient can vary in a very wide range, depending on factors like:
• Road surface conditions (dry or wet)
• Tire side-slip angle
• Tire brand (summer tire, winter tire)
• Vehicle speed
• The slip ratio between the tire and the road.

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

Equation 1

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

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

$\omega_v =$ vehicle speed divided by wheel radius

$V_v =$ vehicle linear velocity

$R_r =$ wheel radius

$\omega_w =$ wheel angular velocity

ABS model

Parameters: -

m	75
g	9.81
I	6
R	1.25
v0	44

Implementation

Discussion: -

• It is obvious that ABS improves the braking performance

• The stopping distance after using ABS system has considerably reduced.

• The error in slip and desired slip is used to manipulate brake pressure in brake cylinder.

• In case of cornering, the side slip ratio would be controlled so that wheels don’t lock and thus ensuring steerability.

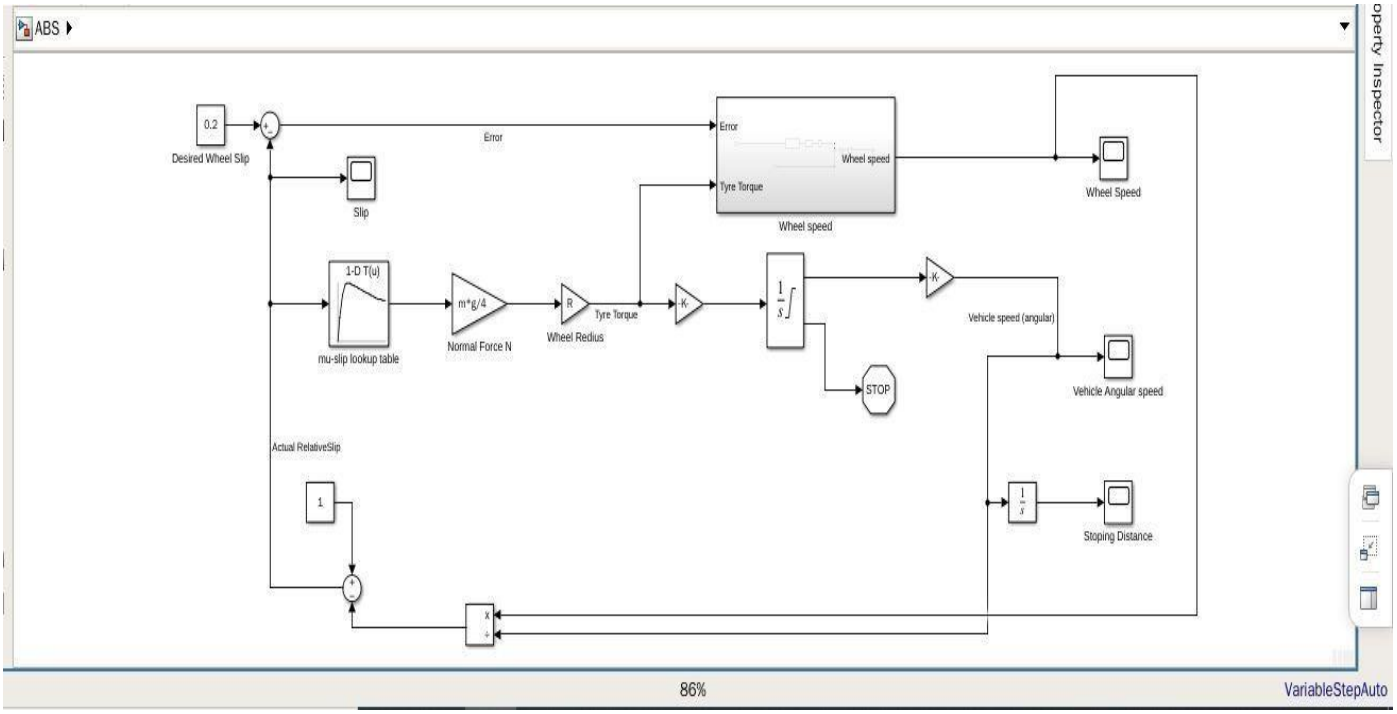


Fig. ABS System

Output: -

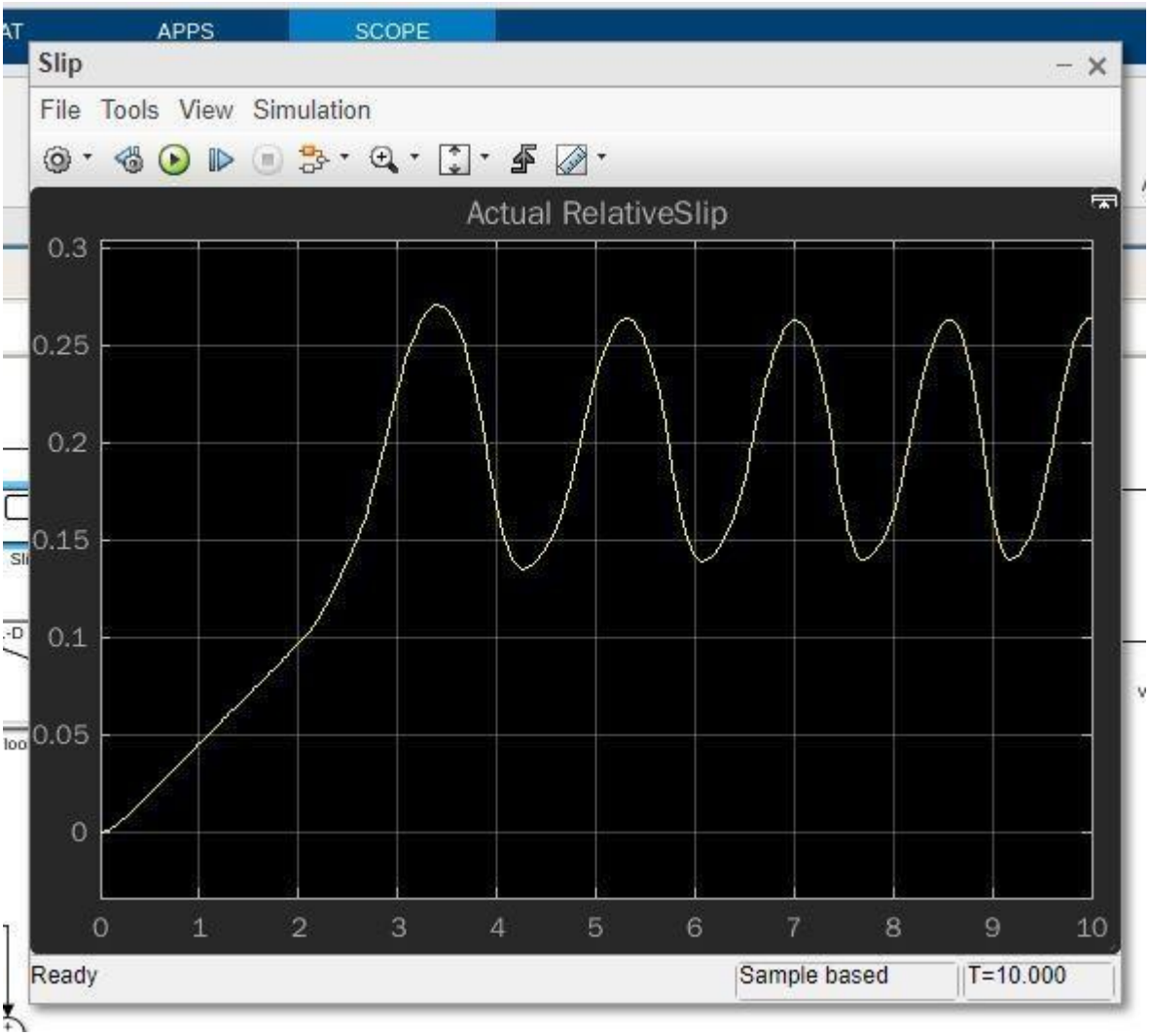


Fig. Slip Graph

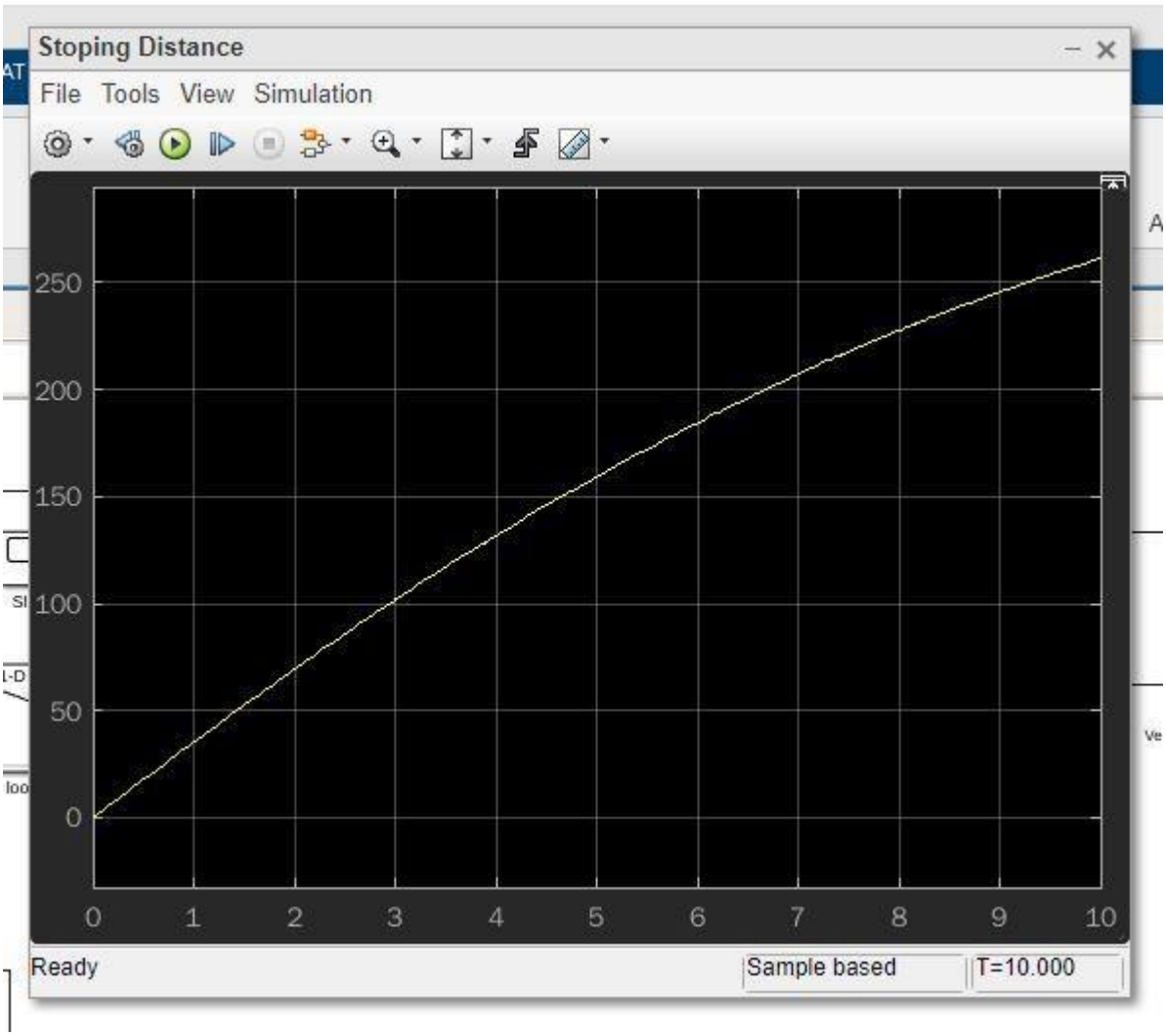


Fig. Stopping Distance

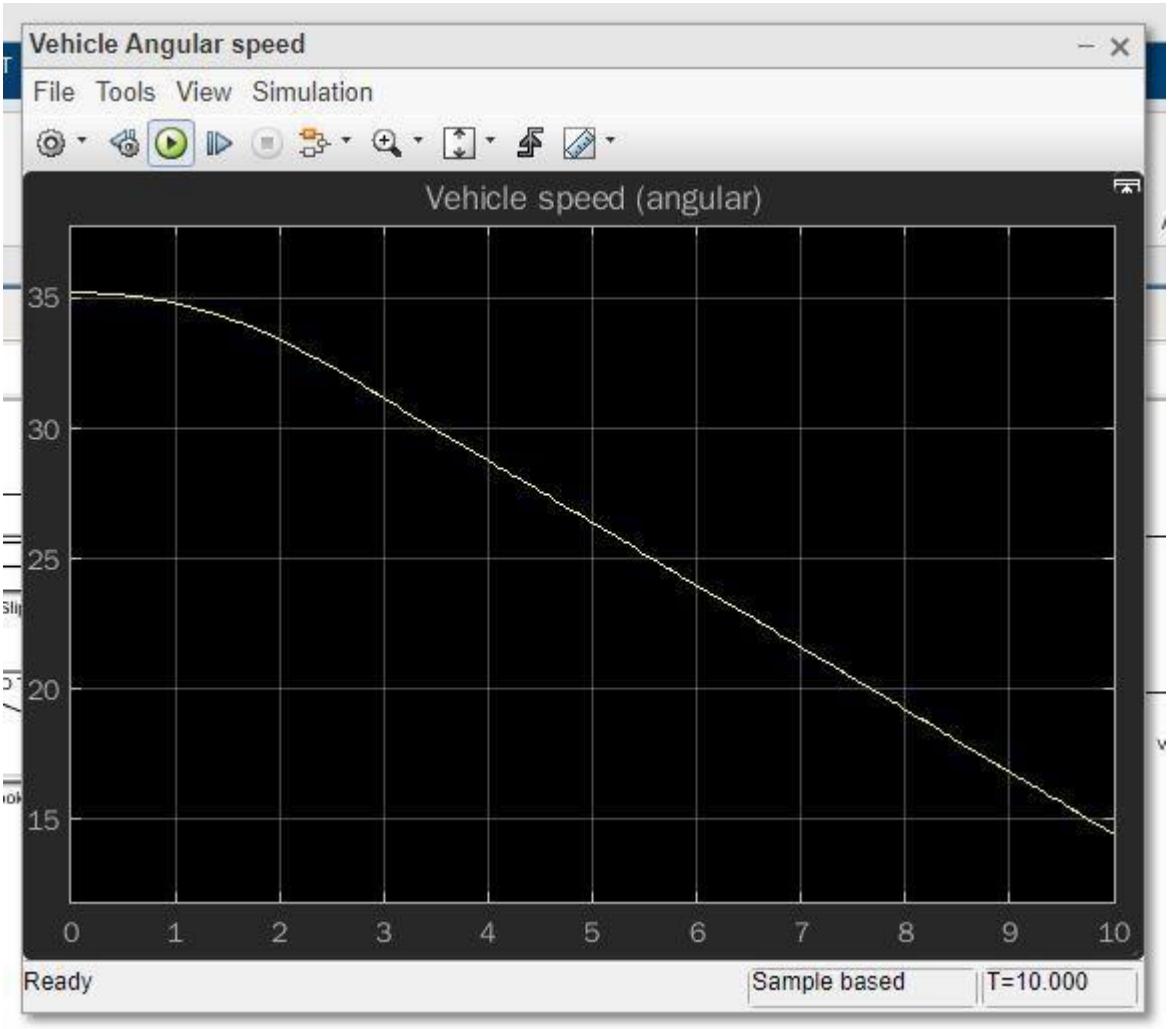


Fig. Vehicle Angular Speed

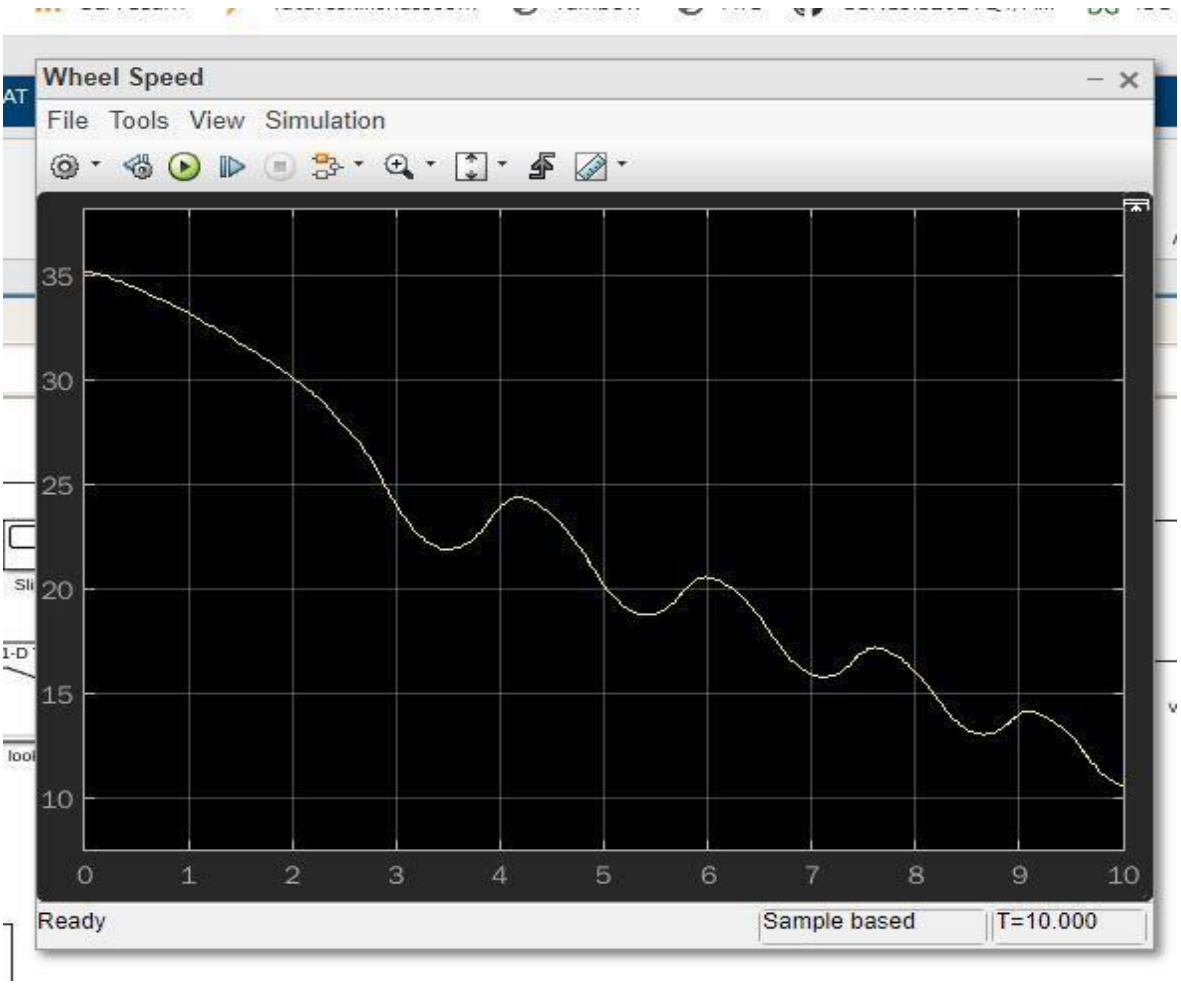


Fig. Wheel Speed

Conclusions

This model shows how you can use Simulink to simulate a braking system under the action of an ABS controller. The controller in this example is idealized, but we can use any proposed control algorithm in its place to evaluate the system's performance.

References

https://www.mathworks.com/help/simulink/slref/modeling-an-anti-lock-braking-system.html

https://www.youtube.com/watch?v=G8VbGtLOmR8

Individual Contribution and Highlights

1. Requirements
2. Implementation
3. Images & Videos (Output video)

Module: - Mastering Microcontrollers with Embedded Driver Development Module

Date: - Jan 3rd to 7th.

Team/Individual: - Team.

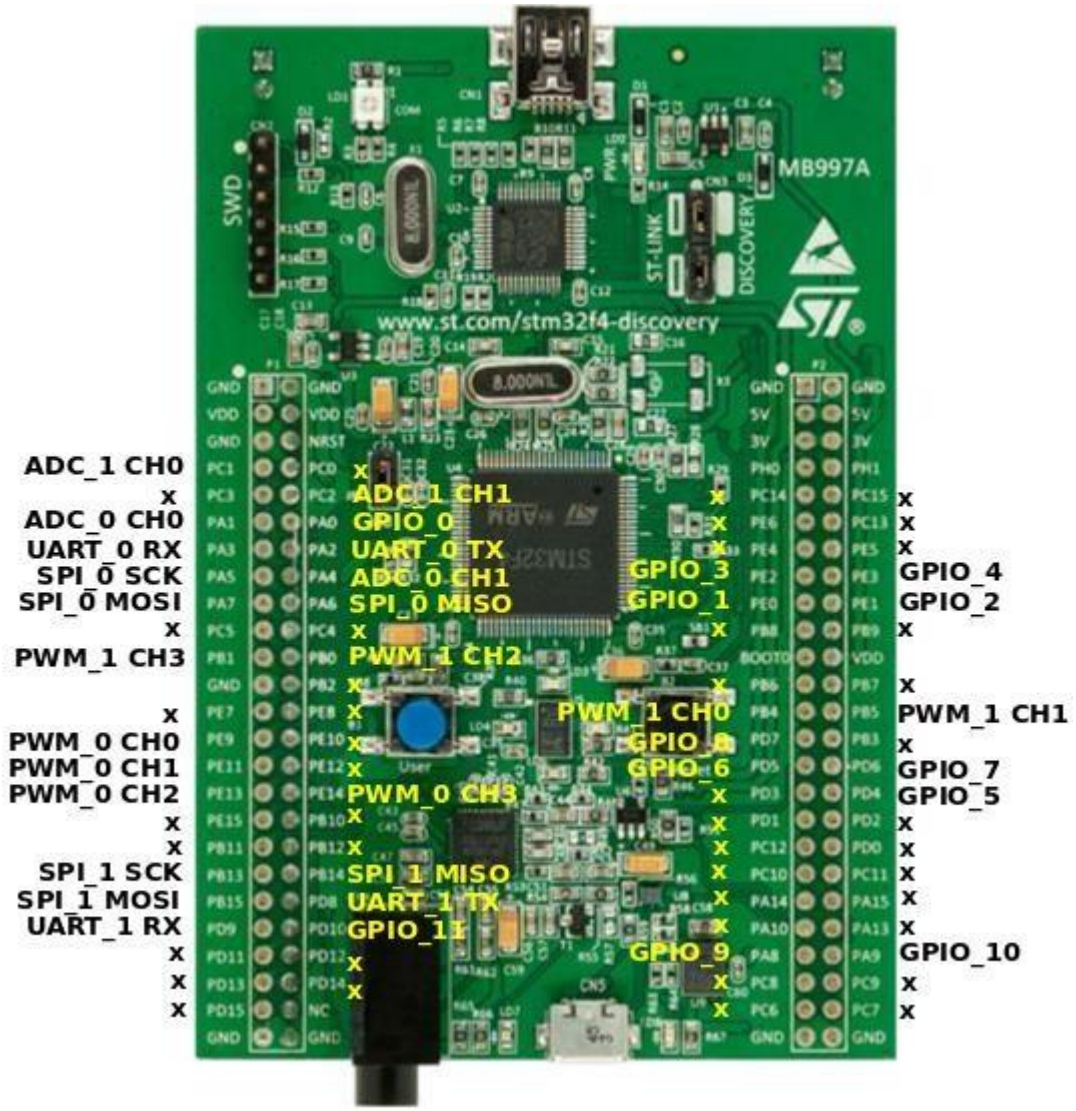
Project topic: - Wiper Control System Using STM32F407VG.

Introduction:

Wiper is an essential component that used to wipe the raindrops or any water from the windscreen. Wipers are designed and made to clear the water from a windscreen. Most of cars have two wipers on the windscreen, one on the rear window and the other on each headlight. The wiper parts visible from outside the car are the rubber blade, the wiper arm holding the blade, a spring linkage, and parts of the wiper pivots. The wiper itself has about six parts called pressure points or claws that are small arms under the wiper. Existing system manually used control stalk to activate wiper and the process of pulling up wiper is difficult to be handled. The driver needs to switch on and off the control stalk and it will reduce the driver's concentration during the driving. Thus, this system is proposed to solve all these problems. The concept of this wiper system is similar with other conventional wiper, yet this system will be upgraded to an automatic control system by using a controller.

Research:

The windshield wiper control module drives the wiper motor based on the the current state of the wiper switch, mostly to support the intermittent wipe feature.



MCU	STM32F4
Family	ARM Cortex
Vcc	2.0V - 3.6V
RAM	192kb
Flash	1024kb
Datasheet	https://www.st.com/resource/en/datasheet/stm32f407vg.pdf

MCU	STM32F4
Reference Manual	https://www.st.com/resource/en/reference_manual/dm00031020-stm32f405-415-stm32f407-417-stm32f427-437-and-stm32f429-439-advanced-arm-based-32-bit-mcus-stmicroelectronics.pdf

4 User controllable LEDs:

NAME	LD1	LD2	LD3	LD4
Colour	Red	Blue	Green	Orange
Pin	PD	PD	PD	PD

Objective:

To build a system would wipe out the rain from any shield with minimum man power having handful of systems automatically, operated and reduce the cost of previously used automatic wiper system and make it available for common man.

Cost and Timeline:

* It is a cost benefit analysis keeping in view the system is economically feasible.

Features:

* The wiper serves to clean the windshield of the car at the front and rear, although not all cars have wipers on the rear side.

* Wiper works by removing oil, dust, rainwater, and dirt that get stuck to the windshield.

SWOT ANALYSIS:

Strength:

* It helps in saving money by switching off the irrigation system when it rains. This saves money by cutting off bills on electricity consumption.

* Operating principle is very easy.

* It consumes less power for operation.

* This saves money by cutting off bills on electricity consumption.

* Almost all motor vehicles, including cars, trucks, etc. are equipped with wipers and are very important for a clear vision.

Weakness:

* The cost of overall system increases as additional components are needed along with rain sensor.

* Rust or corrosion on the ends.

* Dirt and dust stuck in the blades.

* Cracks and tears on the rubber lining.

* Slight separation of the blades from the frame.

Opportunities:

* This project is extending life of rain sensor-based systems such as car wiper, irrigation systems by running them only when it is necessary.

* Using more appropriate rain sensor we can make more precise automatic wiper system.

* By adding microcontroller-based system we can implement some security features for automobiles.

* We can use a capacitor sensor which will enable it work along line of sight.

* The speed controlling mechanism which will make it work according to the intensity and speed of water coming on the sensor.

Threats:

* Too much ice or snow on your windshield may cause your wipers to get jammed and cause circuit overloading.

* Your wipers may also malfunction if components like the control switch, wiper control unit, wiring, or linkage blades fail.

4W's and 1'H

Who:

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

Where:

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

When:

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

What:

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

How:

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

Test Plan

High level test plan: -

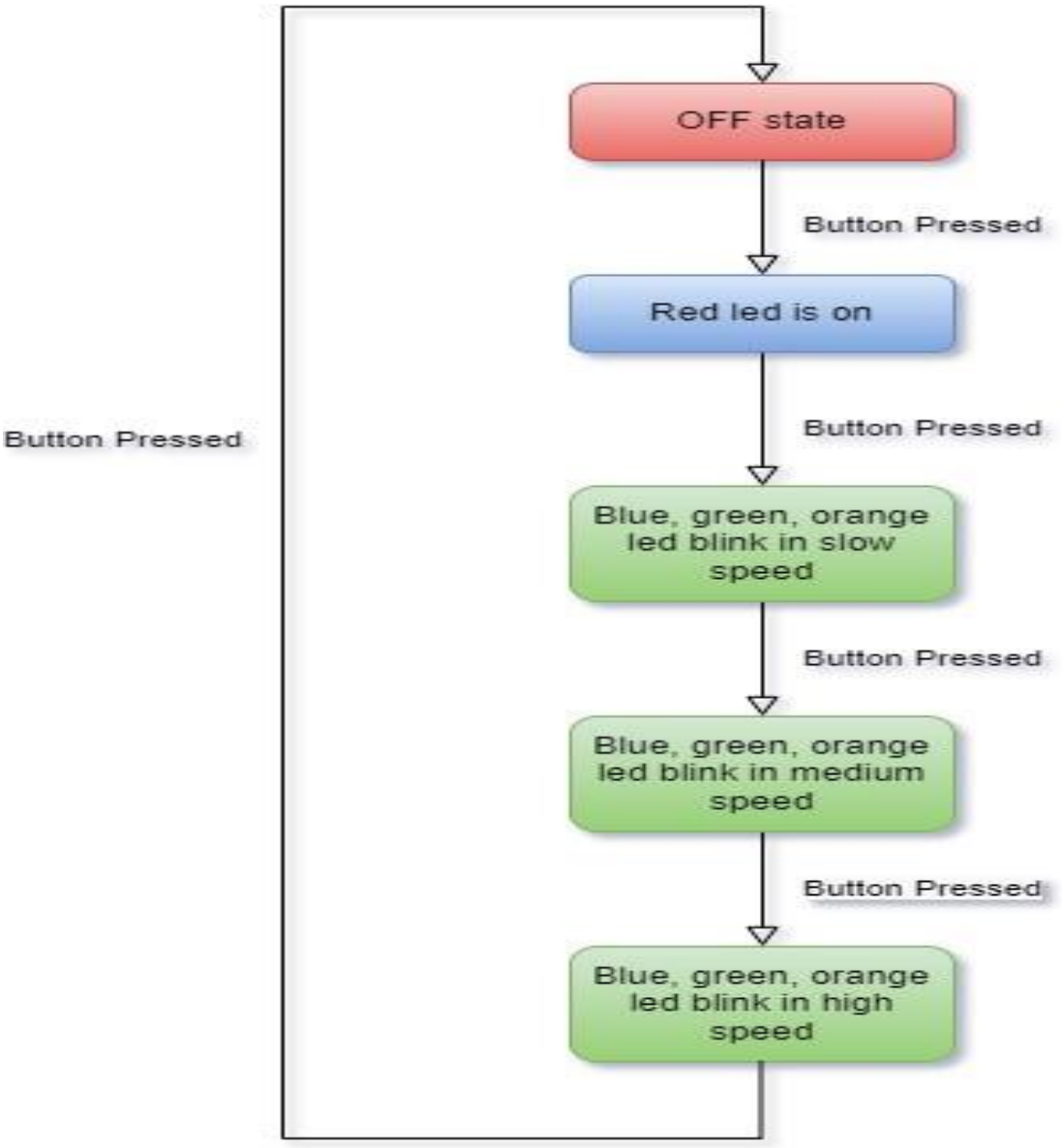
TEST PLAN ID	Description	exp I/P	Exp O/P	Status
HLR01	Check Red LED on & wiper work	Press and held 2sec	Red LED On & Wiper works	pass
HLR02	Press button 1 more time	press button	Blue LED ON	Pass
HLR03	check the frequency of blue LED	1HZ	Wiper revive 1HZ	Pass
HLR04	Press the button again 1 more time	press button	Green LED ON	Pass
HLR05	check the frequency of Green LED	4HZ	wiper receive 4HZ	Pass
HLR06	Press the button 1 more time again	Press Button	Orange LED ON	Pass
HLR07	check the frequency of Orange LED	8HZ	Wiper revive 8HZ	Pass
HLR08	Check the frequency of LED & check wiper work	5,6,7HZ	Wiper revive 5HZ for blue,6HZ for green & 7HZ for orange	Pass
HLR09	Check all LED OFF and wiper stops work	press and held 2secs	All LEDs are OFF & wiper stops	Pass

Low level test plan: -

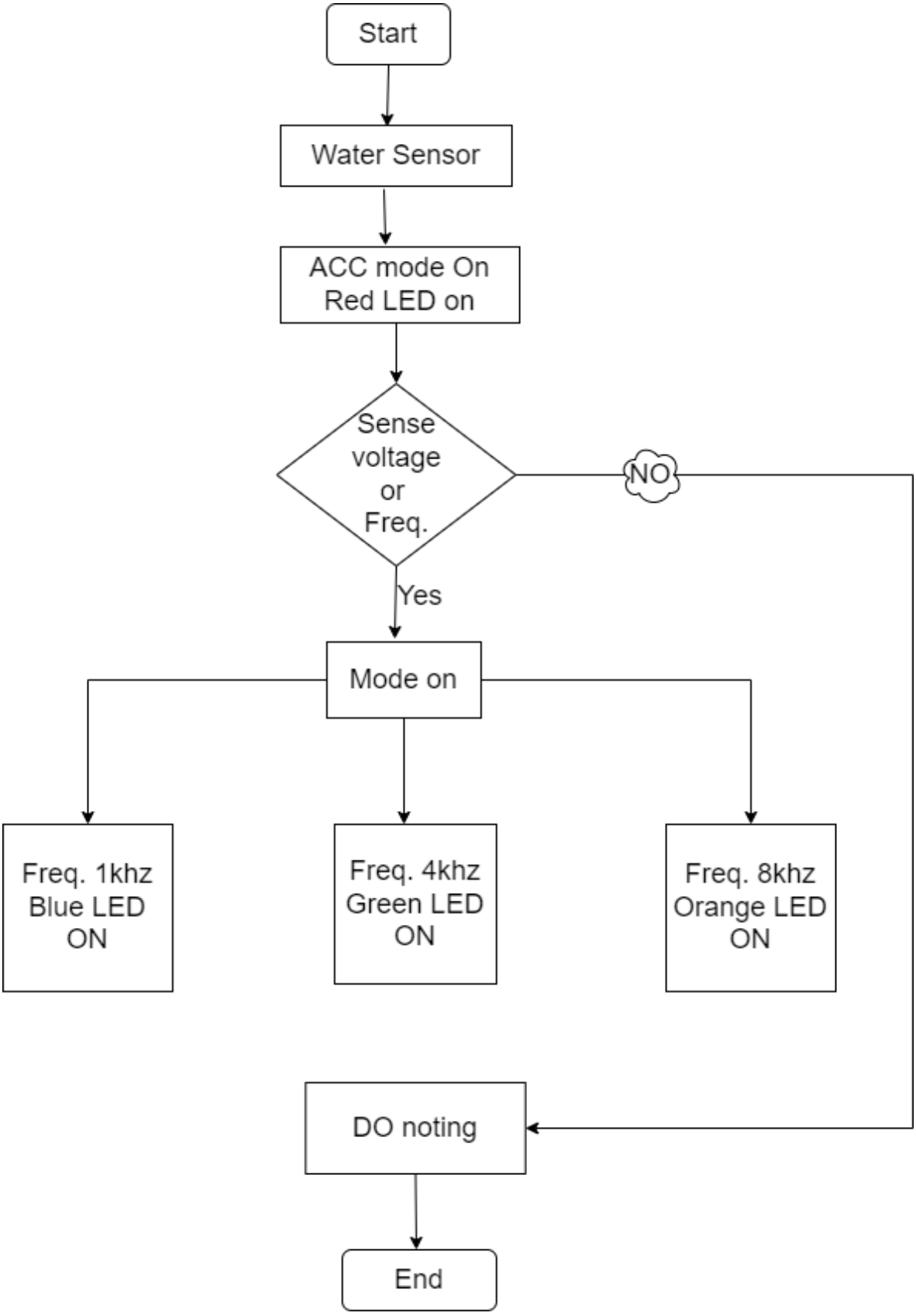
LLR-ID	Description	HLR-ID	exp I/P	Exp O/P	Status
LLR01	Run the system	HLR01	Check the LEDs & wiper	LEDs off & wiper off	Pass
LLR02	Press the button for 2sec	HLR01	Press and held 2sec	Red LED will be ON	Pass
LLR03	Press the button for 1sec	HLR01	Press and held 1sec	Red LED will be OFF	Pass
LLR04	Press the button for 3sec	HLR01	Press and held more then 3sec	Red LED will be OFF	Pass
LLR05	Press the button for 2sec	HLR01	Press and held 2sec	Red LED will be OFF	Fail
LLR06	After red LED ON wiper is also ON	HLR01	Wiper ON	Wiper will start working	Pass
LLR07	After red LED ON wiper is also ON	HLR01	Wiper OFF	Wiper will stop working	Fail
LLR08	Press & check the blue LED	HLR02	press button	Blue LED ON	Pass

Design

Behavioural diagram: -



Flow Chart:


Start
Water Sensor
ACC mode On
Red LED on
Sense
voltage
or
Freq.
NO
Yes
Mode on
Freq. 1khz
Blue LED
ON
Freq. 4khz
Green LED
ON
Freq. 8khz
Orange LED
ON
DO noting
End

Conclusion:

As almost everything described already for this design, we would like to say there are still numerous kinds of enhancements one can implement on this project to make it even more convenient. The project we have made and presented is quite efficient and it is cost effective also. It has great advantage of over the optical sensor covering all the design specifications together with the requirements of common man. The speed controlling mechanism can be added in this project which will make it work according to the intensity and speed of water coming on the sensor. The basic manoeuvring is done only to make it cost effective and reliable.

Implementation and Summary

Git Link:

Link: https://github.com/GENESIS-2022/MasteringMCU-Team35

Individual Contribution and Highlights

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

Role in Project Team

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

Module: - Overview of Automotive Systems.

Date: - Jan 18th to 21st.

Team/Individual: - Team.

Project topic: - Automotive Team Tesla.

Requirements

In this Automotive Team Tesla project, we have taken following features and I have contributed to Power window.

1. Power Window
2. Lighting System
3. Door Locking System.
4. Power Mirror.
5. Wiper Control.

Power Windows: -

High level Requirement: -

ID	Description
HLR1	The window should get raised or lowed after the respective switch of the door is pulled up or pushed.
HLR2	After pressing the door lock button on the key, if any window is opened should get automatically closed.
HLR3	The windows should get automatically raised according to the humidity and cooling status of the AC as per need
HLR4	The obstacle comes in between the window the sensor detect object and reverse the direction of window.
HLR5	While raining, by gathering wiper control data, windows should get raised automatically.

Low level Requirement: -

ID	Description
LLR1	The auto-lower & auto-raised feature is triggered if the driver's window lower switch is held down for more than 1 second.
LLR2	More than one window can be in motion simultaneously.
LLR3	If the up or down command is issued for at least 200ms the window has to be fully opened or closed respectively.
LLR4	The driver command has priority over the passenger command.

UML Design: -

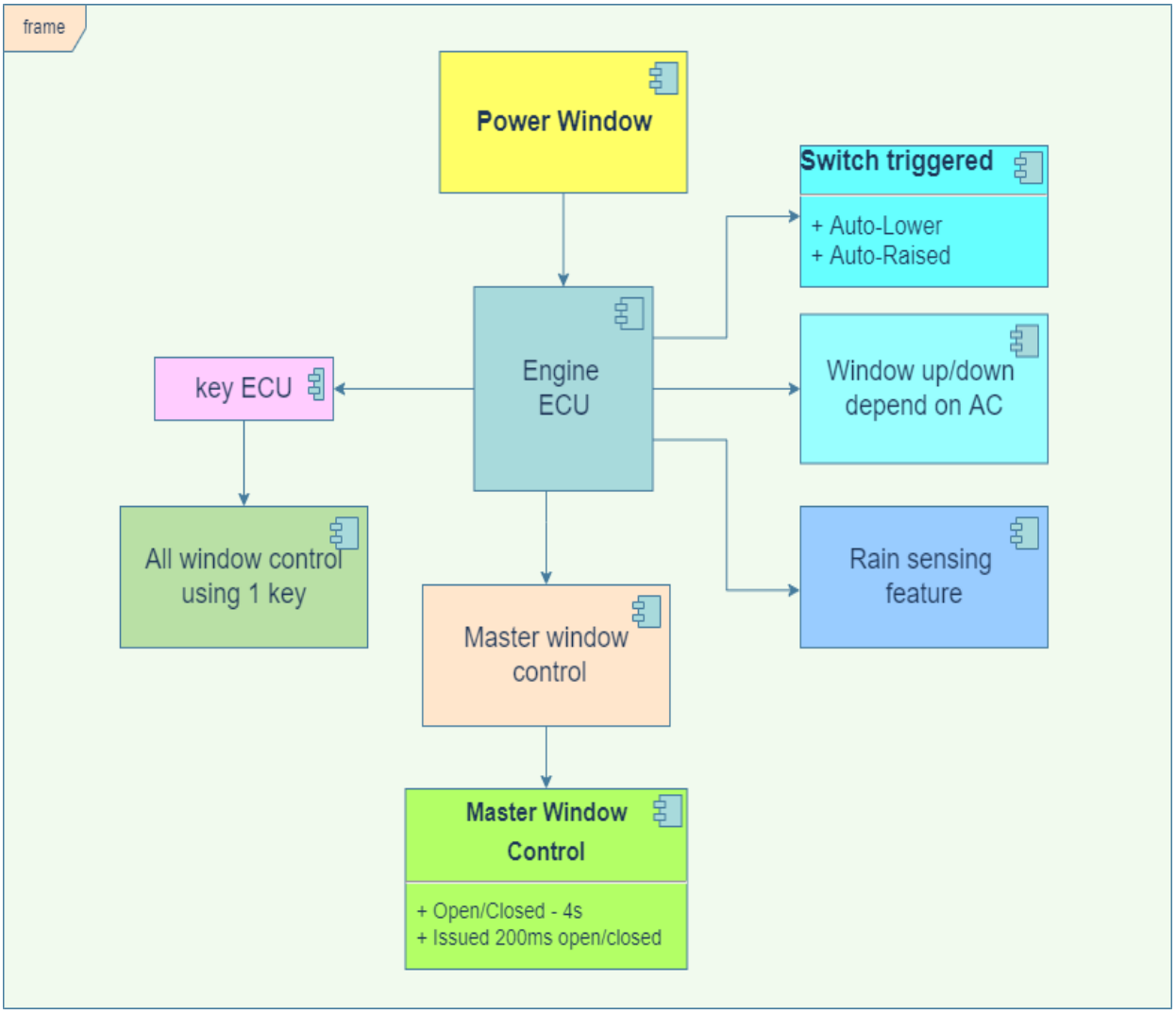


Fig. UML Design.

Assessment: - Door Lock

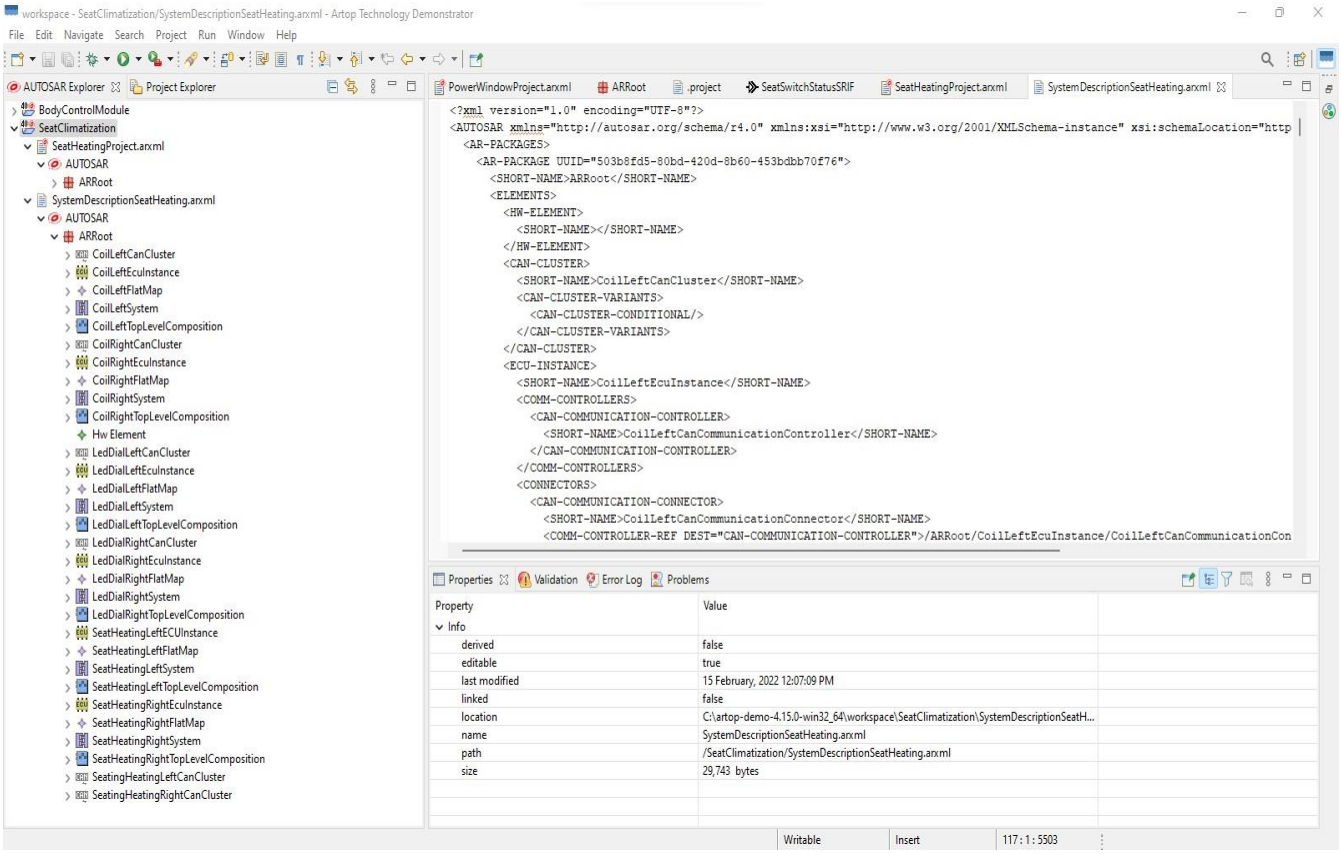


Fig. Door Lock

Group Members: -

Ps No.	Name	Futures
40021041	Vikas Chand	Lightening System
40021032	Vaibhav R. Patil	Door Lock
40021026	Abhishek Bahuguna	Power Mirrors
40021036	Vivek Kushwaha	Seat Climatization
40020574	Mohammad Asif	Wiper Control

Individual Contribution and Highlights

1. Power Window Case Study
2. Source code management using GitHub

Module: - Applied Control Systems & Vehicle Dynamics.

Date: - Feb 1st to 3rd.

Team/Individual: - Team.

Project topic: - EV Bike.

Modules: -

1. MATLAB
2. MATLAB Scrip

Requirements

Motor Performance:

1. Our Arrow M1 has a Mid Drive IPM motor which can produce 7.2 kW power and 40 Nm torque. We find these figures to be a nice balance of drivability and efficiency.
2. Arrow M1 has an acceleration time from 0 to 60 km/hr of 6.5 seconds.
3. Top speed of our Arrow M1 is 100 km/hr.

Battery Performance:

1. We are using a Lithium polymer battery to reduce weight and thereby increase fuel efficiency, performance and handling.
2. A range of 220 km is class leading due to our battery being the biggest at 4.6 kWh.
3. Charging times of our Arrow M1 is higher than the competition at 7.15 hours but we make up for it in the range section.
4. We also offer fast charging.

Braking Performance:

1. Our Arrow M1 also uses combi braking system and use disc brakes for both front and back wheels.
2. Braking performance is on par with the competition.

Wheel Performance:

1. Our Arrow M1 uses Alloy wheels at 12 inches diameter.

2. We use a 90 section, 90 profile tire for a balance between grip, efficiency and ride quality.

Suspension Performance:

1. We use Mono shocks for rear and single fork for front.

Dimensions:

1. Our kerb weight is 110 kg which is just 2 kg heavier than the Ather 450X while having a substantially bigger battery and more powerful motor.

2. Length, Height and Weight are all comparable to the competition.

3. Wheelbase is 1370 mm is the longest in the segment.

4. With a seat height of 782 mm it is accessible for a wide range of people in terms of height

Implementation and Summary

Submission: Submitted in GEA Learn.

Individual Contribution and Highlights

1. Done in MATLAB Script.
2. Done MATLAB scripting for EV Bike
3. Researcher: Done case study for EV Bike

Module: - Autosar basic to Intermediate.

Date: - Feb 1st to 15th.

Team/Individual: - Individual.

Project topic: - Automotive Team Tesla.

Requirements

In this Automotive Team Tesla project, we have taken following features and I have contributed to Power window.

1. Power Window
2. Lighting System
3. Door Locking System.
4. Power Mirror.
5. Wiper Control.

Power Windows: -

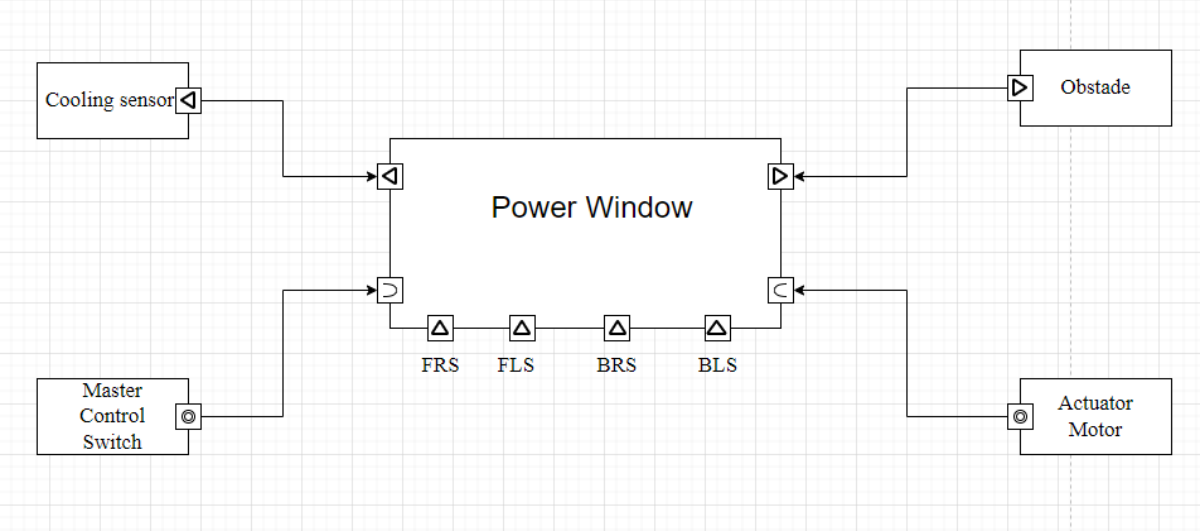
High level Requirement: -

ID	Description
HLR1	The window should get raised or lowed after the respective switch of the door is pulled up or pushed.
HLR2	The windows should get automatically raised according to the humidity and cooling status of the AC as per need
HLR3	The obstacle comes in between the window the sensor detect object and reverse the direction of window.
HLR4	While raining, by gathering wiper control data, windows should get raised automatically.

Low level Requirement: -

ID	Description
LLR1	The auto-lower & auto-raised feature is triggered if the driver's window lower switch is held down for more than 1 second.
LLR2	More than one window can be in motion simultaneously.
LLR3	The driver command has priority over the passenger command.

Design



Individual Contribution and Highlights

1. Power Window Case Study
2. Source code management using GitHub

Learnings of Electrical vehicles

Domain Knowledge Videos

1. Understanding Hill Start Assist!- https://youtu.be/aXEPnWgRnjk?list=PLuUdFsbOK_8rJsh_osoqVKfIRUkb8-rOg
2. Differential | How does it work?- https://youtu.be/nC6fsNXdcMQ?list=PLuUdFsbOK_8rJsh_osoqVKfIRUkb8-rOg
3. Seatbelt | How does it work?- https://youtu.be/uRaU1HMJyCo?list=PLuUdFsbOK_8rJsh_osoqVKfIRUkb8-rOg
4. Understanding Wheel Alignment !- https://youtu.be/7d2K_mKgsZ0?list=PLuUdFsbOK_8rJsh_osoqVKfIRUkb8-rOg
5. How does the Steering Wheel automatically returns to its center?- https://youtu.be/wLbs8kBXgrw?list=PLuUdFsbOK_8rJsh_osoqVKfIRUkb8-rOg
6. Understanding your Car's Steering & Power Steering !- https://youtu.be/em1O8mz7sF0?list=PLuUdFsbOK_8rJsh_osoqVKfIRUkb8-rOg
7. Understanding Anti-lock Braking System (ABS) !- https://youtu.be/98DXe3uKwfc?list=PLuUdFsbOK_8rJsh_osoqVKfIRUkb8-rOg
8. Torque Converter, How does it work ?- https://youtu.be/bRcDvCj_JPs?list=PLuUdFsbOK_8rJsh_osoqVKfIRUkb8-rOg
9. Why you should not PARTIALLY press the Clutch ?- https://youtu.be/_hKvS6xTC0E?list=PLuUdFsbOK_8rJsh_osoqVKfIRUkb8-rOg
10. Clutch, How does it work ?- https://youtu.be/devo3kdSPQY?list=PLuUdFsbOK_8rJsh_osoqVKfIRUkb8-rOg
11. Electric cars vs Petrol cars- https://youtu.be/ewcWN-rHQ6Q?list=PLuUdFsbOK_8rJsh_osoqVKfIRUkb8-rOg
12. How does an Electric Car work ? | Tesla Model S- https://youtu.be/3SAxXUIre28?list=PLuUdFsbOK_8rJsh_osoqVKfIRUkb8-rOg
13. Understanding PLANETARY GEAR set !- https://youtu.be/ARd-

Om2VyiE?list=PLuUdFsbOK_8rJsh_osoqVKfIRUkb8-rOg

14. Automatic vs Manual Transmission- https://youtu.be/auQgOtveQi0?list=PLuUdFsbOK_8rJsh_osoqVKfIRUkb8-rOg
15. Working of Dual Clutch Transmission (DSG)- https://youtu.be/lFAtc-zOKZs?list=PLuUdFsbOK_8rJsh_osoqVKfIRUkb8-rOg
16. Manual Transmission, How it works ?- https://youtu.be/wCu9W9xNwtI?list=PLuUdFsbOK_8rJsh_osoqVKfIRUkb8-rOg
17. Automatic Transmission, How it works ?- https://youtu.be/u_y1S8C0Hmc?list=PLuUdFsbOK_8rJsh_osoqVKfIRUkb8-rOg
18. Petrol (Gasoline) Engine vs Diesel Engine- https://youtu.be/bZUoLo5t7kg?list=PLuUdFsbOK_8rJsh_osoqVKfIRUkb8-rOg
19. Diesel Engine, How it works ?- https://youtu.be/DZt5xU44IfQ?list=PLuUdFsbOK_8rJsh_osoqVKfIRUkb8-rOg
20. How a Differential works ?- https://youtu.be/SOgoejxzF8c?list=PLuUdFsbOK_8rJsh_osoqVKfIRUkb8-rOg

Electrical vehicle basics:

1. EPT trainings learning content

   Video link:EPT Trainings Learning Content | Microsoft Stream (mcas.ms)

   1. EV Architecture and components

   2. Inverter Hardware and software -part 1

   3. Inverter Hardware and software -part 2 4.EV Lab and testing training

   5. Worst case analysis Tolerance analysis

   6. Design calculations -Inverter losses & thermal design

   7. Hardware Simulation and control simulation of DC-DC converter topologies 8.software closed loop control DC -DC Converter topologies

2. BMS (Battery management system)

Video link EV Learning Content | Microsoft Stream (mcas.ms)

1. system requirements, specification feature and DFMEA
2. BMS -software application and Algorithm
3. FUSA-1
4. FUSA-2
5. Wireless BMS
6. BMS testing and BI HIL
7. EV lab Demo and Amaze BMS
8. Overall BMS architecture and platform

3. System level -conventional/EV

Video link: System Level - Conventional/EV | Microsoft Stream (mcas.ms)

1. Inviting for Battery Management System
2. Introduction to Functional Safety
3. Function Safety Session 2
4. Overview of Engine After treatment System, Engine Sensors and transmission System
5. Inviting for Battery Management System – Session 2
6. Overview of different Vehicle architectures
7. DC – DC converter