GENESIS-Learning

Outcome & Mini-project Summary Report

**Details**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Ver. Rel. No. | Release Date | Prepared By | Module Name | Reviewed By | To Be Approved | Remarks/Revision Details |
| 1.0 | 16/02/2022 | Anusha Upendar Gurram | C Programming on Multiple Platforms |  |  |  |
| 1.0 | 02/12/2021 | Anusha Upendar Gurram | Essentials of Embedded System |  |  |  |
| 1.0 | 16/12/2021 | Anusha Upendar Gurram | Applied SDLC and Software Testing |  |  |  |
| 1.0 | 23/12/2021 | Anusha Upendar Gurram | OOPS with Python |  |  |  |
| 1.0 | 31/12/2021 | Anusha Upendar Gurram | Applied Model Based Design Module |  |  |  |
| 1.0 | 07/01/2022 | Anusha Upendar Gurram | Mastering Micro controllers with Embedded Driver Development Module |  |  |  |
| 1.0 | 21/01/2022 | Anusha Upendar Gurram | Overview of Automotive Systems |  |  |  |
| 1.0 | 03/02/2022 | Anusha Upendar Gurram | Applied Control Systems and Vehicle Dynamics |  |  |  |

Contents

[Miniproject – 1: TicTacToe [Individual] 5](#_Toc95931104)

[Modules: 5](#_Toc95931105)

[Requirements 5](#_Toc95931106)

[High Level Requirements 6](#_Toc95931107)

[Low Level Requirements 6](#_Toc95931108)

[Design 7](#_Toc95931109)

[Test Plan 8](#_Toc95931110)

[High Level Test Plan 8](#_Toc95931111)

[Low Level Test Plan 9](#_Toc95931112)

[Implementation and Summary 9](#_Toc95931113)

[Git Link: 9](#_Toc95931114)

[Git Dashboard 10](#_Toc95931115)

[Summary 10](#_Toc95931116)

[Git Inspector Summary 10](#_Toc95931117)

[Miniproject 2 – Embedded Object Sensing System [Individual] 11](#_Toc95931118)

[Modules 11](#_Toc95931119)

[Requirements 11](#_Toc95931120)

[High Level Requirements 11](#_Toc95931121)

[Low Level Requirements 12](#_Toc95931122)

[Design 12](#_Toc95931123)

[Test Plan 14](#_Toc95931124)

[High Level Test Plan 14](#_Toc95931125)

[Low Level Test Plan 15](#_Toc95931126)

[Implementation and Summary 15](#_Toc95931127)

[Git Link: 15](#_Toc95931128)

[Git Dashboard 15](#_Toc95931129)

[Miniproject 3 – Applied SDLC Dec Team 47 [Team] 16](#_Toc95931130)

[Modules 16](#_Toc95931131)

[Requirements 16](#_Toc95931132)

[High Level Requirements 17](#_Toc95931133)

[Low Level Requirements 17](#_Toc95931134)

[Design 18](#_Toc95931135)

[Test Plan 19](#_Toc95931136)

[High Level Test Plan 19](#_Toc95931137)

[Low Level Test Plan 20](#_Toc95931138)

[Implementation and Summary 20](#_Toc95931139)

[Git Link: 20](#_Toc95931140)

[Individual Contribution and Highlights 20](#_Toc95931141)

[Summary 20](#_Toc95931142)

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

[Modules 21](#_Toc95931144)

[Requirements 21](#_Toc95931145)

[High Level Requirements 21](#_Toc95931146)

[Low Level Requirements 21](#_Toc95931147)

[Test Plan 22](#_Toc95931148)

[High Level Test Plan 22](#_Toc95931149)

[Low Level Test Plan 22](#_Toc95931150)

[Implementation and Summary 23](#_Toc95931151)

[Git Link: 23](#_Toc95931152)

[Git Dashboard 24](#_Toc95931153)

[Git Inspector Summary 24](#_Toc95931154)

[Individual Contribution and Highlights 24](#_Toc95931155)

[Miniproject 5 – Scorpio [Team] 25](#_Toc95931156)

[Modules 25](#_Toc95931157)

[Requirements 25](#_Toc95931158)

[Design 25](#_Toc95931159)

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

[Modules 26](#_Toc95931161)

[Requirements 26](#_Toc95931162)

[High Level Requirements 26](#_Toc95931163)

[Low Level Requirements 27](#_Toc95931164)

[Design 28](#_Toc95931165)

[Test Plan 30](#_Toc95931166)

[High Level Test Plan 30](#_Toc95931167)

[Low Level Test Plan 30](#_Toc95931168)

[Implementation and Summary 31](#_Toc95931169)

[Git Link: 31](#_Toc95931170)

[Individual Contribution and Highlights 31](#_Toc95931171)

[Miniproject 7 – MahindraXUV500 Project [Team] 32](#_Toc95931172)

[Modules 32](#_Toc95931173)

[Requirements 32](#_Toc95931174)

[Design 33](#_Toc95931175)

[Implementation and Summary 33](#_Toc95931176)

[Git Link: 33](#_Toc95931177)

[Individual Contribution and Highlights 33](#_Toc95931178)

[Miniproject 8 – Golf Cart[Team] 34](#_Toc95931179)

[Modules 34](#_Toc95931180)

[Requirements 34](#_Toc95931181)

[Implementation and Summary 35](#_Toc95931182)

[Individual Contribution and Highlights 35](#_Toc95931183)

[Miniproject 9 – Door Locking System [Individual] 36](#_Toc95931184)

[Modules 36](#_Toc95931185)

[Requirements 36](#_Toc95931186)

[Design 36](#_Toc95931187)

[Implementation and Summary 37](#_Toc95931188)

[Git Link: 37](#_Toc95931189)

[Individual Contribution and Highlights 37](#_Toc95931190)

## List of Figures

[Figure 1 Behavior Diagram 8](#_Toc95933213)

[Figure 2 Structure Diagram 9](#_Toc95933214)

[Figure 3 Git Dashboard 11](#_Toc95933215)

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

[Figure 5 Behavior Diagram 13](#_Toc95933217)

[Figure 6 Structure Diagram 14](#_Toc95933218)

[Figure 7 Block Diagram 14](#_Toc95933219)

[Figure 8 Simulation 15](#_Toc95933220)

[Figure 9 Git Dashboard 16](#_Toc95933221)

[Figure 10 Behavior Diagram 19](#_Toc95933222)

[Figure 11 UserFlow Diagram 19](#_Toc95933223)

[Figure 12 Structure Diagram 20](#_Toc95933224)

[Figure 13 Git Dashboard 25](#_Toc95933225)

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

[Figure 15 Structure Diagram 29](#_Toc95933227)

[Figure 16 Behavior Diagram 30](#_Toc95933228)

[Figure 17 Structure Diagram 34](#_Toc95933229)

[Figure 18 VFB Diagram 37](#_Toc95933230)

# 

# Mini-project – 1: Tic-Tac-Toe [Individual]

**Date: -** Nov 15th to 25th.

**Team/Individual:-**Individual

**Project topic:- Tictactoe Game**

## Modules:

1. C Programming
2. Git

### Requirements

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

**Why:**

1. The Project will help student and children to develop with concentration.
2. **This is a game anybody can play*.***
3. It can be used by anyone at any place.
4. **This is a purely leisure game. Because there are so many different outcomes in this game, businesses can utilise it to design strategies.**

**Where: A simple tic tac toe game is accessible on a number of websites. In addition, corporations and organisations use it.**

**Who:** It can be played by anyone.

**When: This game can be played if you're bored or want to learn more about the game's methods, consequences, and scenarios game.**

**How: Blocks your opponent from winning as you try to win.**

### High Level Requirements

| **ID** | **Description** | **Status** |
| --- | --- | --- |
| HLR\_1 | Users can use a web browser to obtain the information | Implemented |
| HLR\_2 | From the landing page, the user should choose the game's difficulty level and begin playing | Implemented |
| HLR\_3 | When a user moves, the game page allows them to move | Implemented |
| HLR\_4 | The user can see the opponent's movements in real time on the game page | Implemented |
| HLR\_5 | |  | | --- | | The user can pick up where they left off in a game that isn't yet finished | |  | | Implemented |
| HLR\_6 | When one player gets three symbols in a row, the game should be over | Implemented |
| HLR\_7 | After the game, the user sees the results | Implemented |

### 

### Low Level Requirements

| **ID** | **Description** | **Status** |
| --- | --- | --- |
| LLR-1 | Players personal details like gender, contact number. | Implemented |
| LLR-2 | After the game the user see the results. | Implemented |

## 

## Design

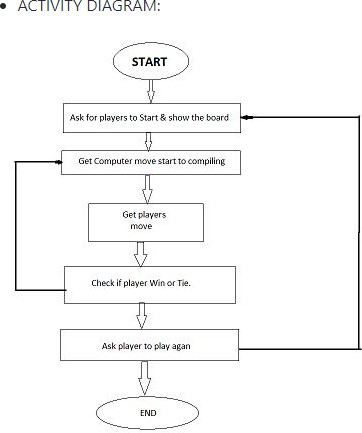


Figure Behavior Diagram

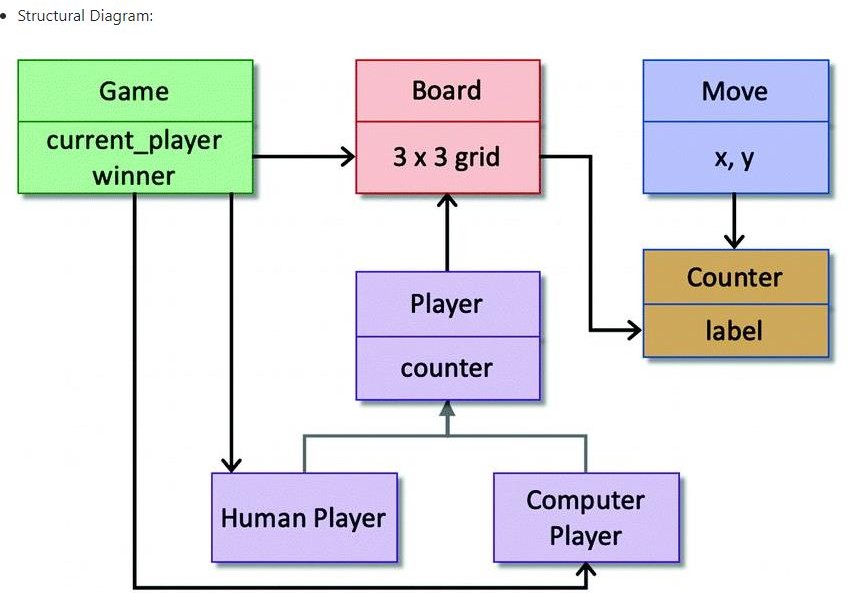


Figure Structure Diagram

## Test Plan

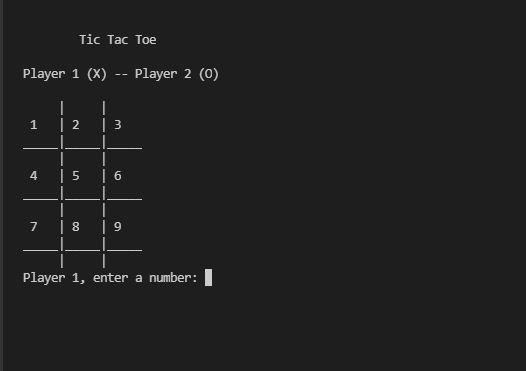
### High Level Test Plan

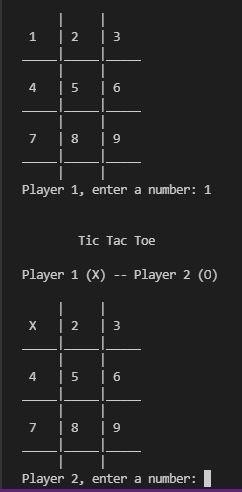
|  |  |
| --- | --- |
| HLR-1 | Users can use a web browser to obtain the information. |
| HLR-2 | From the landing page, the user should choose the game's difficulty level and begin playing |
| HLR-3 | When a user moves, the game page allows them to move |
| HLR-4 | The user can see the opponent's movements in real time on the game page |
| HLR-5 | The user can pick up where they left off in a game that isn't yet finished |
| HLR-6 | When one player gets three symbols in a row, the game should be over |
| HLR-7 | After the game, the user sees the results |

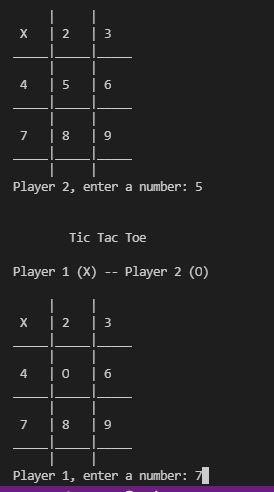
### Low Level Test Plan

|  |  |
| --- | --- |
| LLR-1 | Name of the player |
| LLR-2 | Players personal details like gender, contact number. |

## Implementation and Summary







### Git Link:

Link: <https://github.com/GurramAnnu/M1_application_TicTacToe.git>

**Certification Done: -**

* SOLO Learn Certification.
* Linux Certification.
* GitHub Learning Certification.

# 

# Miniproject 2 – Embedded Calculator [Individual]

**Date: -** Nov 26th to 2nd.

**Team/Individual:-** Individual.

**Project topic: -** Embedded Object Sensing System

## Modules

1. C Programming
2. Embedded System
3. Simul-IDE
4. Git

### Requirements

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

## Who:

**The importance of the project is to detect and calculate accurate distance from any obstacle that we want to measure. Ultrasonic sensors are used primarily as proximity sensors. They can be found in automobile self-parking technology, medical applications and anti-collision safety systems**

## What:

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

## When:

**This will be useful to user when they need assistance in dark light and whenever we want to measure the particular distance from the moving object.**

## Where:

**It measures accurate distance using a non-contact technology.**

**How:**

**Ultrasonic sensors generate high frequency sound waves and evaluate the echo which is received back by the sensor if obstacle detected. Sensors calculate the time interval between sending the signal and receiving the echo to determine the distance to an object.**

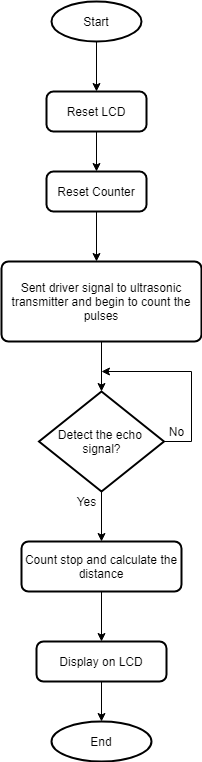
### High Level Requirements

| ID | Description | Status |
| --- | --- | --- |
| HLR\_1 | Interfacing Ultrasonic Sensor with controller | Implemented |
| HLR\_2 | Interfacing LCD display with controller | Implemented |
| HLR\_3 | Installing required software on the PC/Laptop | Implemented |

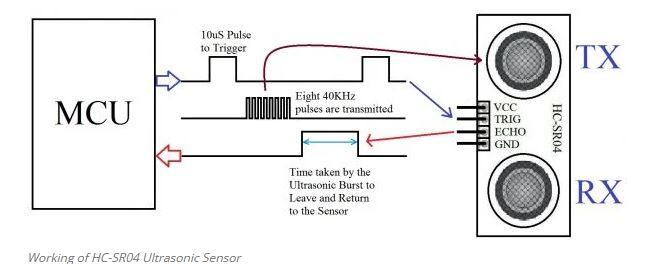
### Low Level Requirements

| ID | Description | Status |
| --- | --- | --- |
| LLR1 | Setting the range up-to 80 cm | Implemented |
| LLR2 | The high level signal is sent to 10ms using Trigger | Implemented |

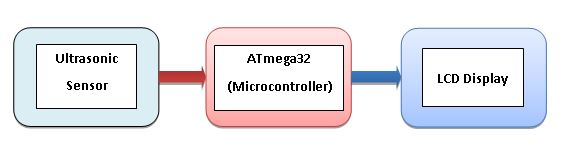
## Design



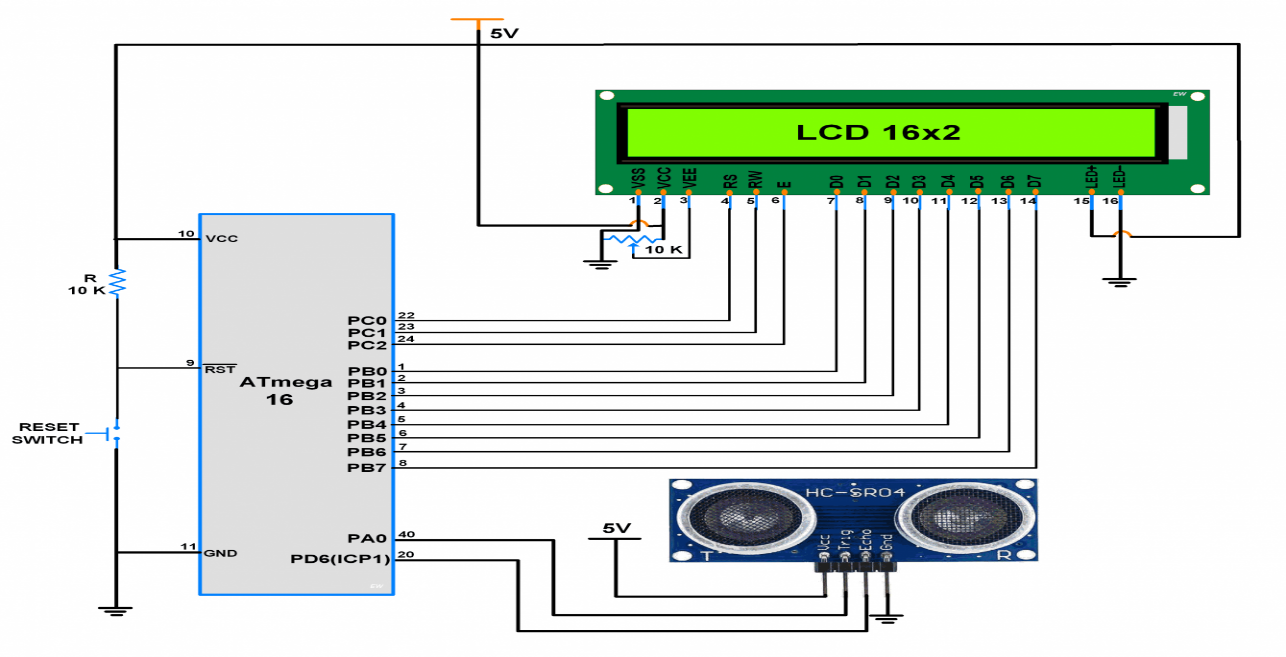
Flow chart



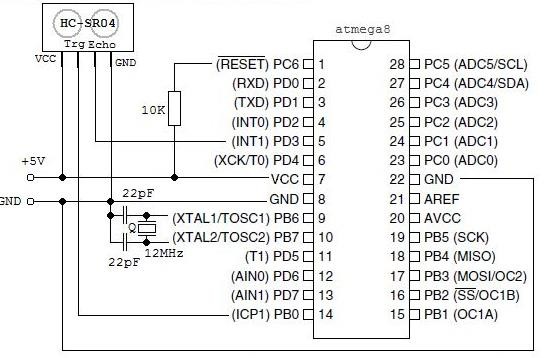
Behavioural Diagram



Block Diagram



Structural Diagram



Circuit Diagram

## Test Plan

### High Level Test Plan

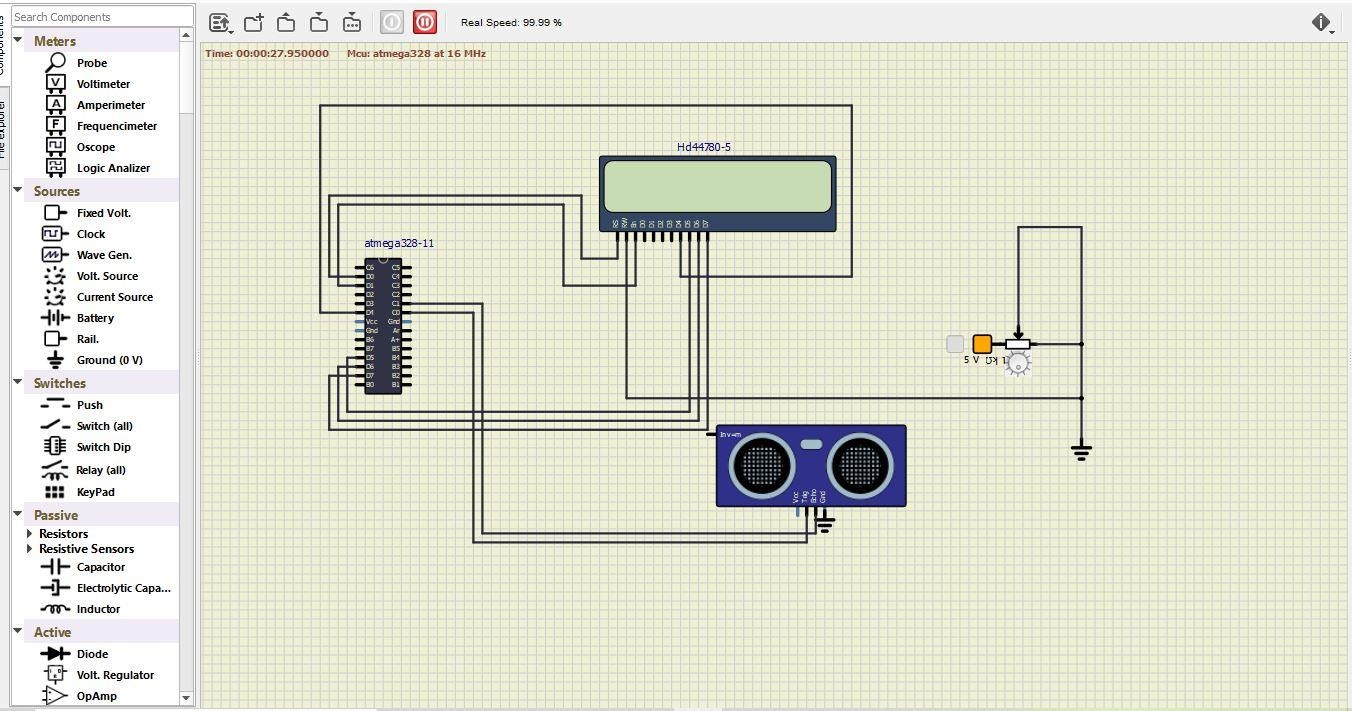
| **ID** | **Description** | **Expected I/P** | **Expected O/P** | **Actual O/P** | **Type Of Test** |
| --- | --- | --- | --- | --- | --- |
| HLTP\_1 | Sensor Working | Obstacle | Obstacle Detected | SUCCESS | Requirement Based |
| HLTP\_2 | High accuracy | Object in the range | Accurate distance from object on Display | SUCCESS | Scenario Based |
| HLTP\_3 | Measuring time lapses between the sending and receiving of the ultrasonic pulse | Object in the range | Display | SUCCESS | Requirement  based |

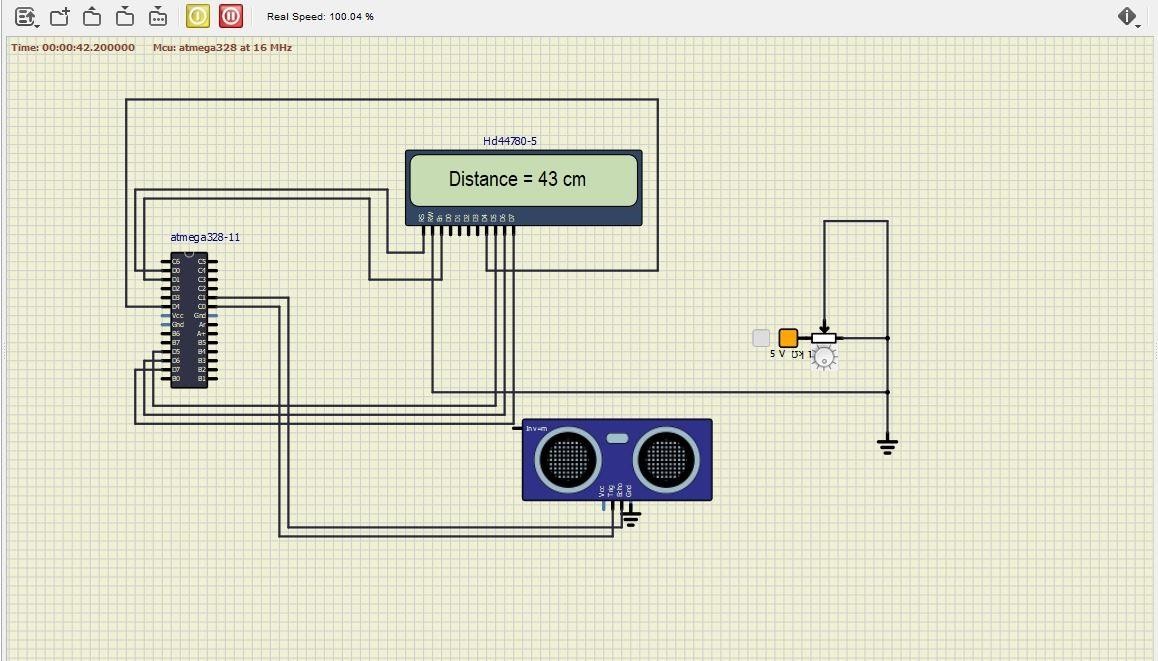
### Low Level Test Plan

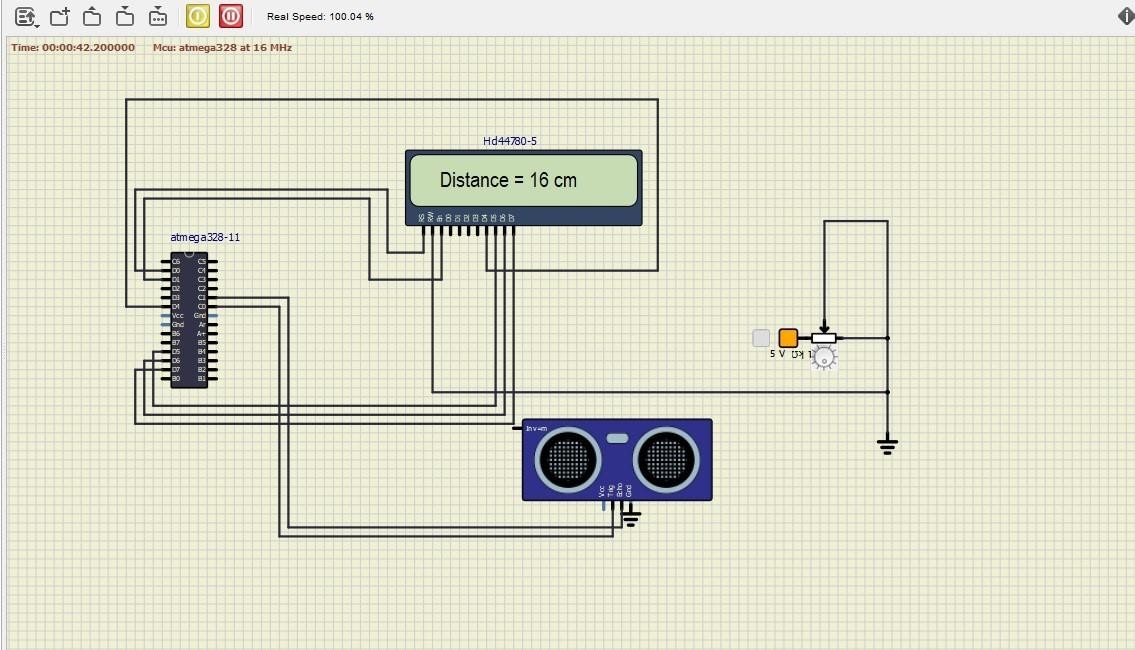
| ID | Description | Expected I/P | Expected O/P | Actual O/P | Type Of Test |
| --- | --- | --- | --- | --- | --- |
| LLTP\_1 | Detection of clear objects | Obstacle | Display | SUCCESS | Scenario Based |
| LLTP\_2 | Provide multiple range measurements | Moving Obstacle | Display | SUCCESS | Requirement Based |

## 

## Implementation and Summary







### Git Link:

Link: <https://github.com/GurramAnnu/M2-Embedded_ObjectSensingSystem.git>

**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.

## 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.

## Opportunity:

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

## 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. |

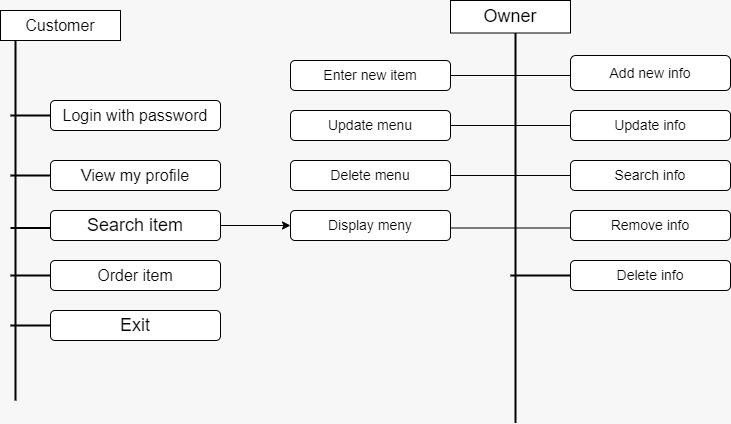
|  |  |
| --- | --- |
| **ID** | **Description** |
| HLR5 | Customer should able to search item via name or item code. |
| HLR6 | Application should able to do the all calculation that are required to generate bill amount. |
| HLR7 | Customer should be able to delete the particular item from ordered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

## Low Level Requirements

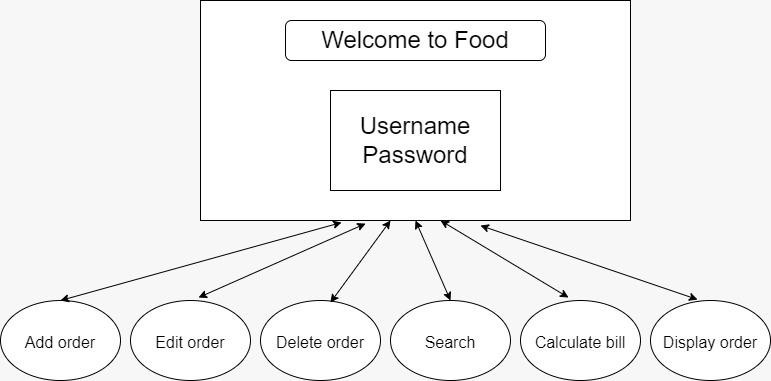
|  |  |
| --- | --- |
| **ID** | **Description** |
| LLR1 | Login Page off Food Court. |
| LLR2 | Enter user and password. |
| LLR3 | Newly added details should be display. |
| LLR4 | Item name, quantity, rate should be removed. |
| LLR5 | Item name, item number and item rate should be there while generating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

**Design:**

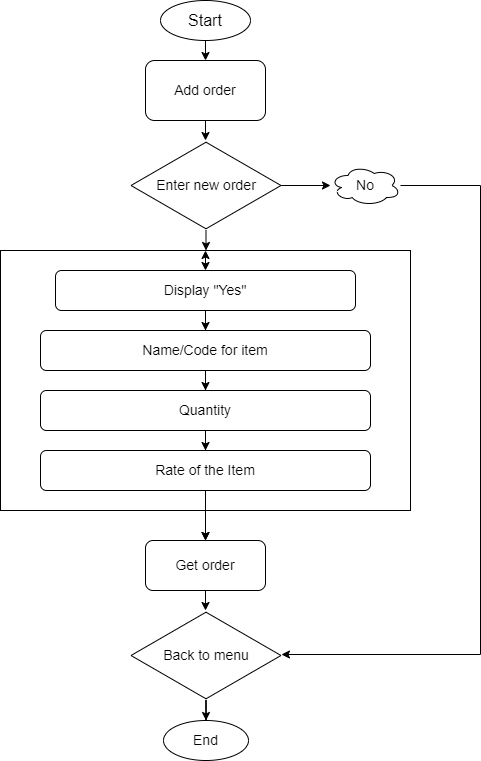
## Block diagram:

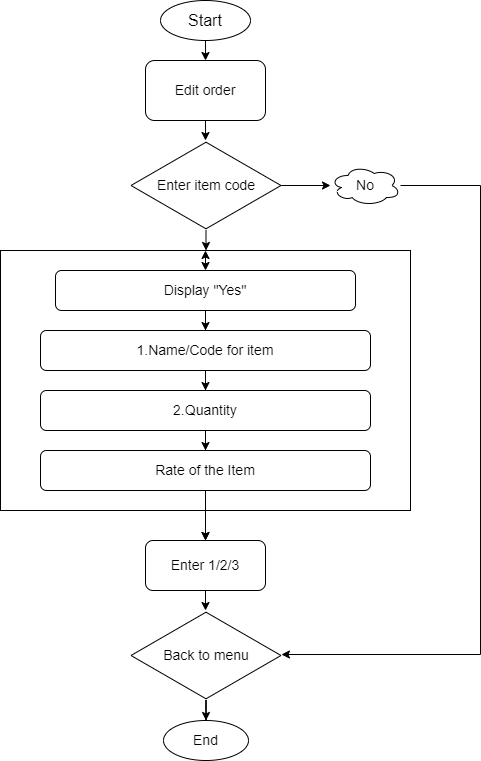


## Structural diagram:

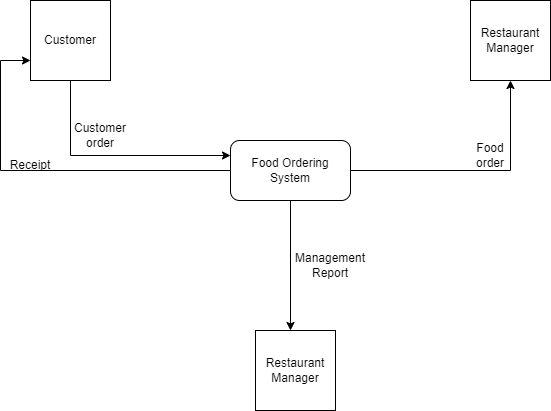


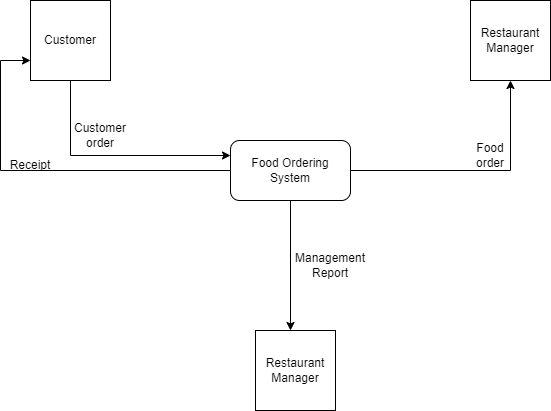
**Behavioural diagram:** Flow chart: 1





## High level diagram:

Low level diagram:



## Test Plan:

## High Level Test Plan

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **TEST\_I D** | **Descripti on** | **Expecte d I/P** | **Expected O/P** | **Actual O/P** | **Type Of Test** |
| HLR\_1 | Login to system | Provide proper Userna me with characte r length of 10 | entered username | entered username | Requirem ent Based |
| HLR\_2 | Login to system | Provide proper passwor d with characte r length of 10 | Login successful | Login successful | Requirem ent Based |
| HLR\_3 | Providing items that you want to add | User Choice | Added Successfu lly | Added Successfu lly | Requirem ent Based |
| HLR\_4 | Display the menu | ---- | Added Items is Displayed | Added Items is Displayed | Requirem ent Based |
| HLR\_5 | Bill Calculatio n | Choice | Customer' s Bill | Customer’ s Bill | Requirem ent Based |

## Low Level Test Plan

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **TEST\_ ID** | **Descripti on** | **Expect ed I/P** | **Expected O/P** | **Actual O/P** | **Type Of Test** |
| LLR\_1 | Login to system | userna me and passwo rd is  Incorre ct (in case number s or  exceed length) | Login is Unsuccess ful | Login is Unsuccess ful | Requirem ent Based |
| LLR\_2 | Edit Item | Provide Item code | Item Edited | Item Edited | Requirem ent Based |
| LLR\_3 | Delete Item | Provide Item | Item Deleted | Item Deleted | Requirem ent Based |
| LLR\_4 | Search Item | Provide Item Code | Item Searched | Item Searched | Requirem ent Based |
| LLR\_5 | exit operation | ---- | Exit Successful ly | Exit Successful ly | Requirem ent Based |

## Summary

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

## Git Link:

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

# 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

## Modules

1. Python
2. Git

## Link for template

2 Slots format -

M/A: [https://docs.google.com/spreadsheets/d/1jtKnXV12VE1fH20CGDo4B3u](https://docs.google.com/spreadsheets/d/1jtKnXV12VE1fH20CGDo4B3uNWRTAhQCWz-hHUDWUe3I/edit?usp=sharing) [NWRTAhQCWz-hHUDWUe3I/edit?usp=sharing](https://docs.google.com/spreadsheets/d/1jtKnXV12VE1fH20CGDo4B3uNWRTAhQCWz-hHUDWUe3I/edit?usp=sharing)

4 Slots format -

M1/M2/A1/A2: [https://docs.google.com/spreadsheets/d/1jVheSPZkOtfNKRNo](https://docs.google.com/spreadsheets/d/1jVheSPZkOtfNKRNoc_858nwk2UaHCe0gExTNZfZ8vxA/edit?usp=sharing) [c\_858nwk2UaHCe0gExTNZfZ8vxA/edit?usp=sharing](https://docs.google.com/spreadsheets/d/1jVheSPZkOtfNKRNoc_858nwk2UaHCe0gExTNZfZ8vxA/edit?usp=sharing)

### 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 successfully attendance entry | HLR\_03 | Implemented |
| 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

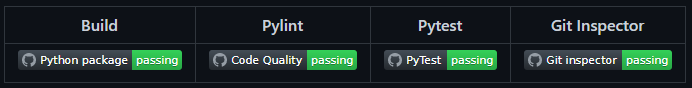
| ID | HLTP ID | Description | Expected I/P | Actual O/P | Type Of Test |
| --- | --- | --- | --- | --- | --- |
| LLTP\_01 | HLTP\_01 | User can get Attendance Status | SUCCESS | SUCCESS | Requirement Based |
| LLTP\_02 | HLTP\_01 | User can enter Status input to get the Attendance Status | SUCCESS | SUCCESS |  |
| 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/Pradnya579/GENESIS2021-OOP-Python_Team_46.git>

### Git Dashboard



Git Dashboard

### 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

**Module: -** Applied Model Based Design Module.

**Date: -** Dec 27th to 31st.

**Team/Individual: -** ScorpioTeam.

**Project topic: -** Air Conditioner System**.**

## Modules

1. Matlab
2. Git

## Requirements

We have implemented following features

1. Power Windows
2. Anti-Lock Breaking System
3. Wiper System
4. Air Conditioner
5. Sunroof Control

## Design

This project was implemented using Matlab.

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

**Date: -** Jan 3rd to 7th.

**Team/Individual: -** Team.

**Project topic: -** Wiper Control System Using STM32F407VG.

## Modules

1. C Programming
2. STM32

## Requirements

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

**Who:**

Users who drives the vehicles can use this.

**What:**

This project is concerned is about automatic wiper system in vehicles.

**When:**

When there is a change in the weather the wipers work automatically.

**Where:**

This projects helps the users to achieve the clear path when there is a change of weather.

**How:**

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

### High Level Requirements

| ID | Description | Status |
| --- | --- | --- |
| HLR\_1 | Press and hold the button to put the Ignition key position in ACC mode | Implemented |
| HLR\_2 | Different wiper frequencies to be set (1Hz, 4Hz & 8Hz) | Implemented |
| HLR\_3 | Hold the button to put the system in Idle state | Implemented |

### Low Level Requirements

| ID | Description | HLTP ID | Status |
| --- | --- | --- | --- |
| LLR\_1 | Hold the button for 2 sec to bring the ignition key position at ACC mode | HLR\_1 | Implemented |
| LLR\_2 | Hold the button for 2 sec to go back to the Idle state | HLR\_1, HLR\_3 | Implemented |
| LLR\_3 | Press the button one time to set frequency to 1Hz | HLR\_2 | Implemented |
| LLR\_4 | Press the button second time to set frequency to 4Hz | HLR\_2 | Implemented |
| LLR\_5 | Press the button third time to set frequency to 8Hz | HLR\_2 | Implemented |
| LLR\_6 | Press the button fourth time to turn OFF the wiper action | HLR\_2 | Implemented |
| LLR\_7 | Hold the button for 2 sec to bring ignition key position at Lock state | HLR\_3 | Implemented |

## Design

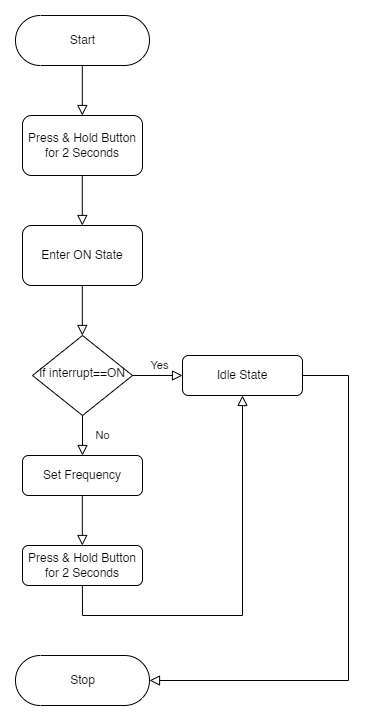


Figure Structure Diagram

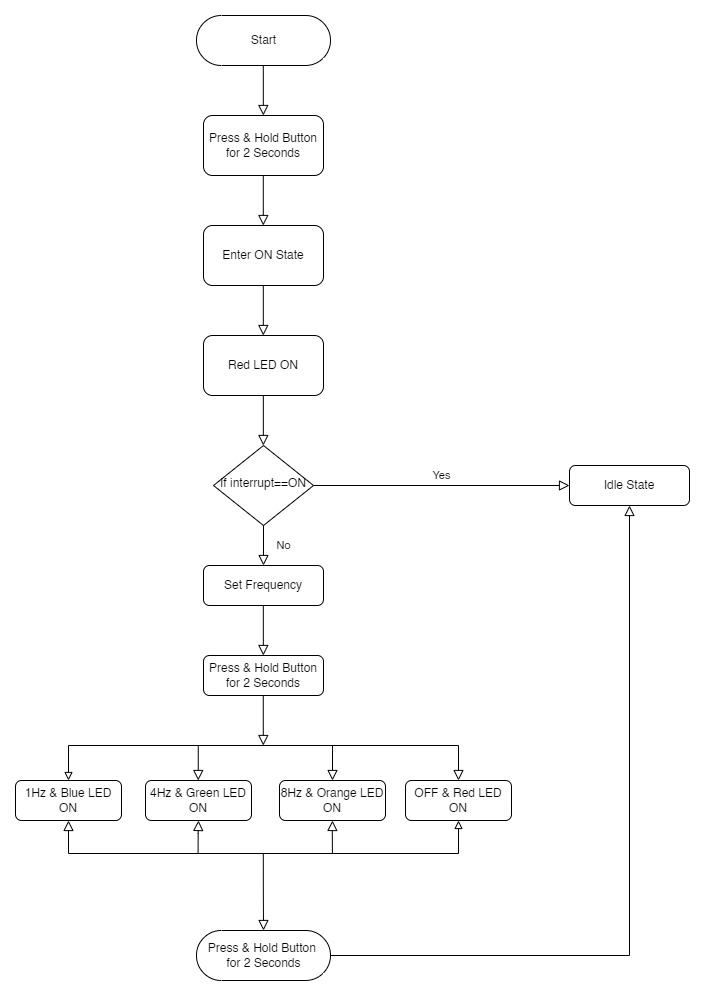


Figure Behavior Diagram

## Test Plan

### High Level Test Plan

| ID | Description | Output | Type of Test |
| --- | --- | --- | --- |
| HLTP\_1 | Press and hold the button to put the Ignition key position in ACC mode | System Enters ACC State | Requirement Based |
| HLTP\_2 | Different wiper frequencies to be set (1Hz, 4Hz & 8Hz) | Responds Based on Input | Requirement Based |
| HLTP\_3 | Hold the button to put the system in Idle state | Enters Idle State | Requirement Based |

### Low Level Test Plan

| ID | Description | Output | HLTP ID | Type of Test |
| --- | --- | --- | --- | --- |
| LLTP\_1 | Hold the button for 2 sec to bring the ignition key position at ACC mode | Red LED-ON | HLTP\_1 | Requirement Based |
| LLTP\_2 | Hold the button for 2 sec to go back to the Idle state | Red LED-OFF | HLTP\_1, HLTP\_3 | Requirement Based |
| LLTP\_3 | Press the button one time to set frequency to 1Hz | Blue LED-ON | HLTP\_2 | Requirement Based |
| LLTP\_4 | Press the button second time to set frequency to 4Hz | Green LED-ON | HLTP\_2 | Requirement Based |
| LLTP\_5 | Press the button third time to set frequency to 8Hz | Orange LED-ON | HLTP\_2 | Requirement Based |
| LLTP\_6 | Press the button fourth time to turn OFF the wiper action | All LED OFF except Red | HLTP\_2 | Requirement Based |
| LLTP\_7 | Hold the button for 2 sec to bring ignition key position at Lock state | Red LED-OFF | HLTP\_3 | Requirement Based |

## Implementation and Summary

### Git Link:

Link: <https://github.com/GENESIS-2022/MasteringMCU-Team17.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. Integrator: Integrated all the codes
3. Tester: Writing Testcases and testing the integrated code

**Module: -** Overview of Automotive Systems.

# Date: - Jan 18th to 21st. Team/Individual: - Team.

**Project topic: - MahindraXUV500.**

## Modules

1. Automotive Systems
2. Git

### Requirements

In this Mahindra project we have taken following features. I have contributed to Door Lock System

| **Name** | **Ps No.** | **Topic** |
| --- | --- | --- |
| Tharageshwari Babu | 40021061 | Anti-Lock Braking System |
| Tamildurga Pari | 40021023 | Wiper Control System |
| Monisha Guruchandiran | 40021054 | Climate Control System |
| Anusha Upendar Gurram | 40021029 | Door Lock Control System |

## Design

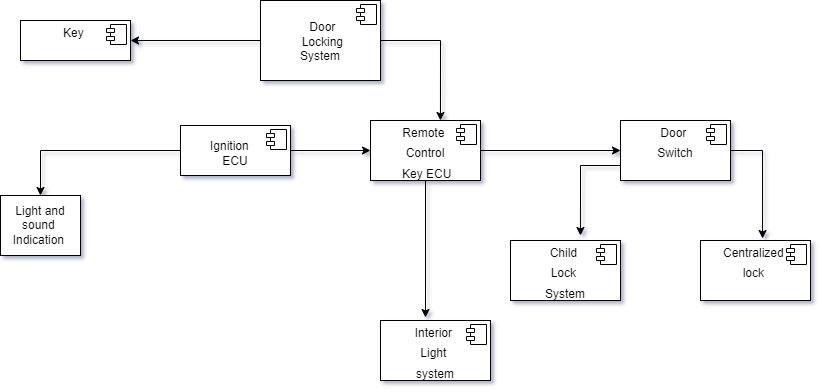


Figure Structure Diagram

## Implementation and Summary

### Git Link:

Link: <https://github.com/Tamil-durga-Pari/M1_Automotive-System-MahindraXUV500.git>

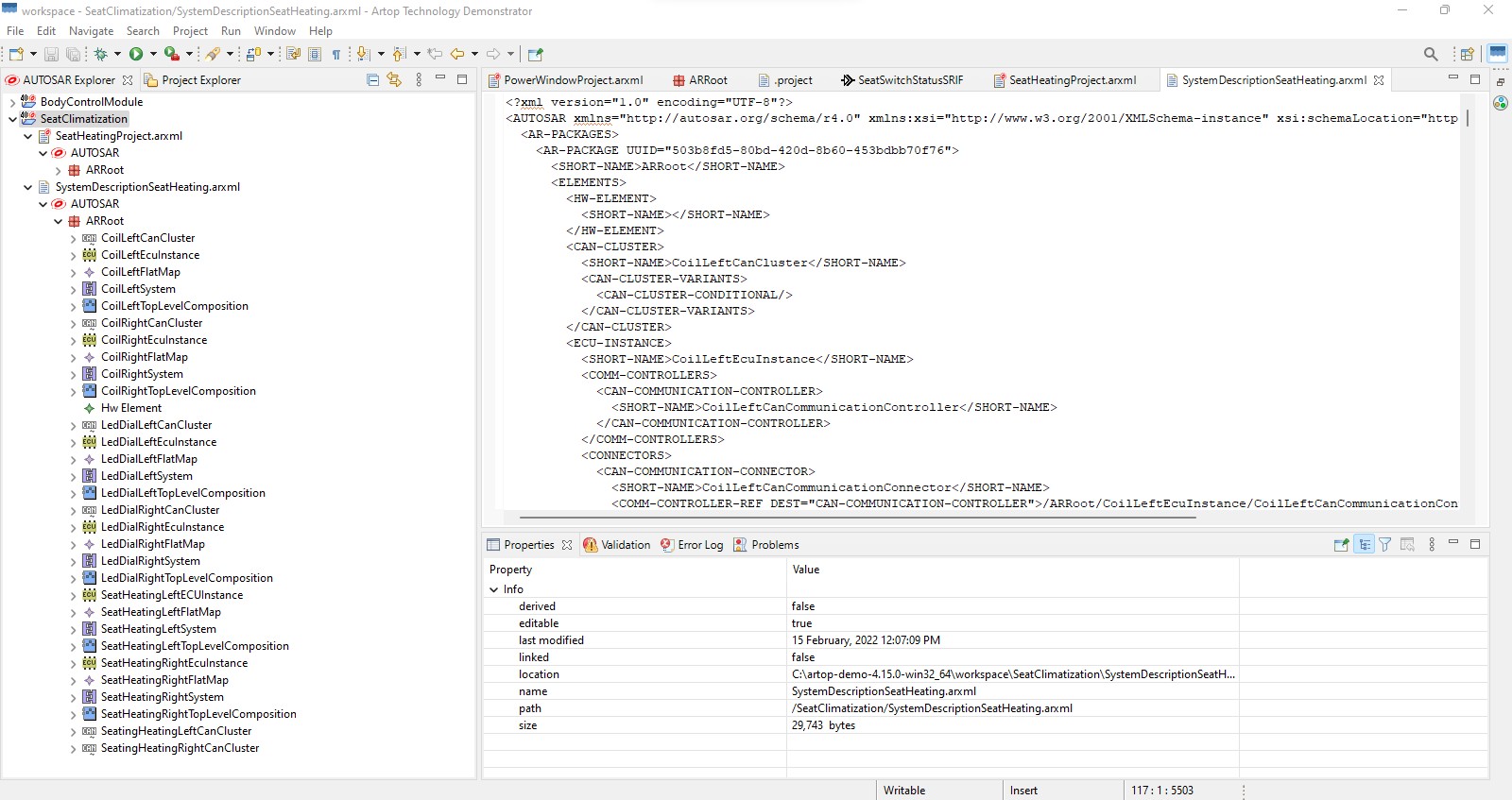
### Individual Contribution and Highlights

* 1. Central Door Lock 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 Central Door Lock System

ASSESSMENT:-

Seat Climatization:-

# 

**Module: -** Applied Control Systems & Vehicle Dynamics.

**Date: -** Feb 1st to 3rd.

**Team/Individual: -** Team.

**Project topic: - GOLF CART.**

## Modules

1. Mat lab
2. Applied Vehicle dynamics

### Requirements

**EZGO marathon Vs YAMAHA UMAX rally 2+2:**

**Motor Specifications:**

|  |  |  |
| --- | --- | --- |
| **Component** | **EZGO marathon** | **Y- UMAX rally 2+2** |
| Engine Type | 350cc twin cylinder unit  18 cubic inches | 402cc low-emission single cylinder 60 degree incline OHV |
| Fuel Tank Capacity | Twin cylinder unit | 5.2US GAL (20 LITERS) |
| Top speed | 12-14 Nm | 15 mph (24.1 km/hr) |
| Minimum Turning Radius | 4.24m | 3.98m |
| Maximum Forward Speed | 12mph (19.3 km/h) | 15mph (24.1 km/h) |

**Battery Specifications:**

|  |  |  |
| --- | --- | --- |
| **Component** | **EZGO marathon** | **Y- UMAX rally 2+2** |
| Battery Type | Works on gas cylinder | Lithium-ion |
| Range | 25-30 miles | 35 miles |
| Battery Charging Time | Works on gas cylinder | 4 Hours |
| Battery Capacity | Works on gas cylinder | 2 kWh |
| No of Cells | Works on gas cylinder | 20,500 |

**Wheel Specifications:**

|  |  |  |
| --- | --- | --- |
| **Wheel Type** | **EZGO marathon** | **Y- UMAX rally 2+2** |
| Front Wheel Size | 10 inches | 12 inches |
| Rear Wheel Size | 11 inches | 12 inches |
| Front Tyre Size | 22\*9-11\*10.5 - ply monitor k272 | 23\*10.5-12\*4- ply monitor k389 |







**Range**

Vehicle Model Buggy

Colour Wide range of Colour options

Usage/Application Eco-friendly drive within enclosed campuses.

Seating Capacity 4-12 persons

Running Distance 40 Km/Charge

Maximum Speed 11KM/Hour

Brand Maini

Material Powder coated tubular steel Chassis,

Body frame dent proof ABS

Number Of Battery 6 no, 8 Volt each, 150AH

Voltage 48V

Motor Rated 48 Volt AC Motor

Power 4 Kw Continuous

Speed 11KM/Hour

Seater 2-14 seater

Capacity Kg 200-1100 KG

**Battery**

Brand Trojan

Capacity @20Hr - 170Ah

Model Name/Number Trojan Motive T-875 with Bayonet Cap

Voltage 8V

Battery Type Deep-Cycle Flooded/Wet Lead-Acid Battery

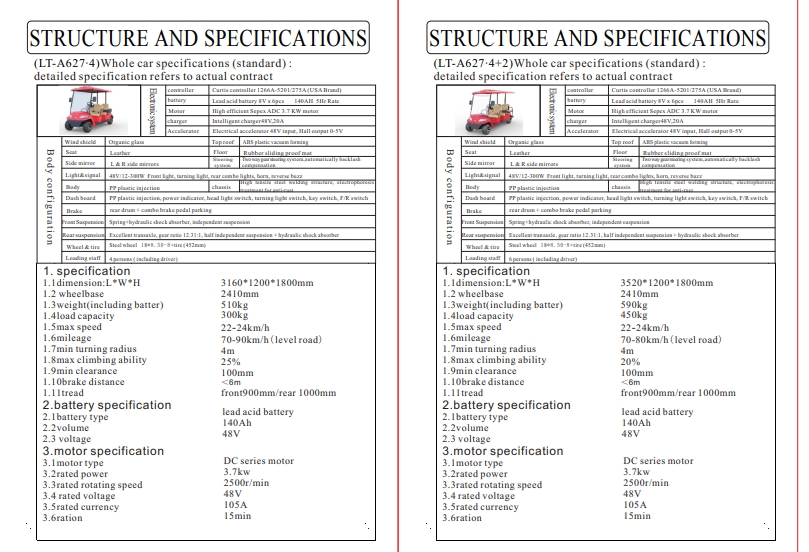
Dimensions 10.27 x 7.10 x 11.14 Inches

Weight 29 Kg

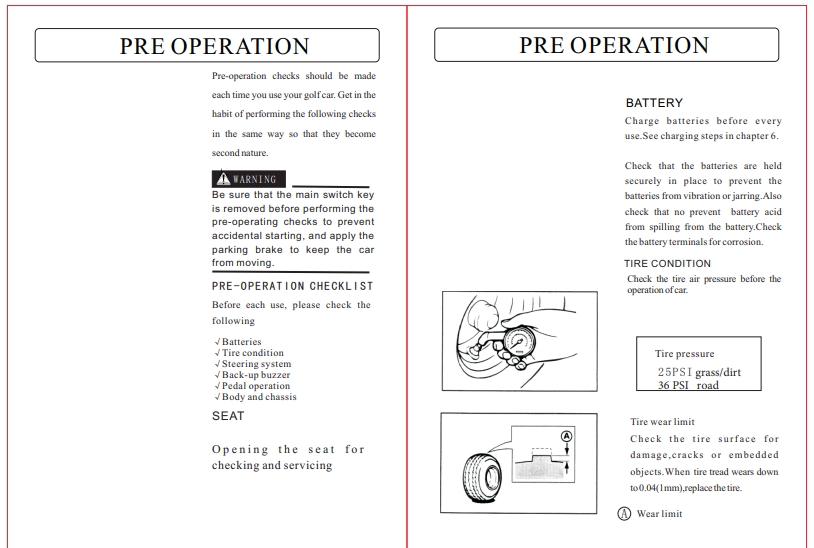
Material Polypropylene

Application/Usage Golf cart, Low Speed Electric Vehicle

**Report**



## 



## 

## Implementation and Summary

Submission: Submitted in GEA Learn

### Individual Contribution and highlights

### 1. Done in Matlab Script

# 

## Module: - Autosar basic to Intermediate.

**Date: -** Feb 1st to 15th.

**Team/Individual: -** Individual.

**Project topic: - MahindraXUV500.**

## Modules

1. Autosar
2. Git

### Requirements

| S.NO | Function | Description |
| --- | --- | --- |
| 1 | Engine | The Engine Should be ON to Use the Parking System |
| 2 | Reverse Gear | Car Should be in Reverse Gear to Enable Parking System |
| 3 | Assistive Mode | Assistive Mode Can Be Enabled or Disabled |
| 4 | Assistive Mode ON | If Assistive Mode is ON the Sensor Assists |
| 5 | Assistive Mode OFF | If Assistive Mode is OFF the Sensor Will not Assist |
| 6 | Sound Frequency | Sound Frequency Varies Based On The Object Distance |

## Design

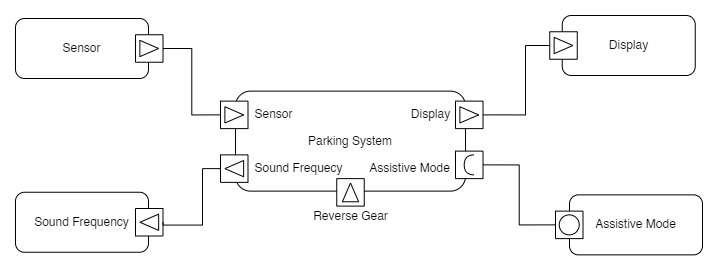


Figure VFB Diagram

## Implementation and Summary

### Git Link:

Link: <https://github.com/Tamil-durga-Pari/M1_Automotive-System-MahindraXUV500.git>

### 

### Individual Contribution and Highlights

|  |  |  |
| --- | --- | --- |
| Tharageshwari Babu | 40021061 | Anti-Lock Braking System |
| Tamildurga Pari | 40021023 | Wiper Control System |
| Monisha Guruchandiran | 40021054 | Climate Control System |
| Anusha Upendar Gurram | 40021029 | Door Lock Control System |