

**Details**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Ver. Rel. No. | Release Date | Prepared By | Modules | To Be Approved | Remarks/Revision Details |
| 1.0 | 19/02/2022 | Vishnu Kumar.V  40020488 | C Programming on Multiple Platforms |  |  |
| 1.0 | 19/02/2022 | Vishnu Kumar.V  40020488 | Essentials of Embedded System |  |  |
| 1.0 | 19/02/2022 | Vishnu Kumar.V  40020488 | Applied SDLC and Software Testing |  |  |
| 1.0 | 19/02/2022 | Vishnu Kumar.V  40020488 | OOPS with Python |  |  |
| 1.0 | 19/02/2022 | Vishnu Kumar.V  40020488 | Applied Model Based Design Module |  |  |
| 1.0 | 19/02/2022 | Vishnu Kumar.V  40020488 | Mastering Microcontrollers with Embedded Driver Development Module |  |  |
| 1.0 | 19/02/2022 | Vishnu Kumar.V  40020488 | Overview of Automotive Systems |  |  |
| 1.0 | 19/02/2022 | Vishnu Kumar.V  40020488 | Applied Control Systems and Vehicle Dynamics |  |  |
| 1.0 | 19/02/2022 | Vishnu Kumar.V  40020488 | Classic Autosar Basic to Intermediate |  |  |

Contents

[Miniproject – 1: Phonebook Management System [Individual] 5](#_Toc95931104)

[Modules: 5](#_Toc95931105)

[Requirements 5](#_Toc95931106)

[High Level Requirements 6](#_Toc95931107)

[Low Level Requirements 6](#_Toc95931108)

[Design 7](#_Toc95931109)

[Test Plan 8](#_Toc95931110)

[High Level Test Plan 8](#_Toc95931111)

[Low Level Test Plan 9](#_Toc95931112)

[Implementation and Summary 9](#_Toc95931113)

[Git Link: 9](#_Toc95931114)

[Git Dashboard 10](#_Toc95931115)

[Summary 10](#_Toc95931116)

[Git Inspector Summary 10](#_Toc95931117)

[Miniproject 2 :Functioning LED [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 – NFT Marketplace [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 – Kia Project[Team] 25](#_Toc95931156)

[Modules 25](#_Toc95931157)

[Requirements 25](#_Toc95931158)

[Design 25](#_Toc95931159)

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

[Modules 26](#_Toc95931161)

[Requirements 26](#_Toc95931162)

[High Level Requirements 26](#_Toc95931163)

[Low Level Requirements 27](#_Toc95931164)

[Design 28](#_Toc95931165)

[Test Plan 30](#_Toc95931166)

[High Level Test Plan 30](#_Toc95931167)

[Low Level Test Plan 30](#_Toc95931168)

[Implementation and Summary 31](#_Toc95931169)

[Git Link: 31](#_Toc95931170)

[Individual Contribution and Highlights 31](#_Toc95931171)

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

[Modules 32](#_Toc95931173)

[Requirements 32](#_Toc95931174)

[Design 33](#_Toc95931175)

[Implementation and Summary 33](#_Toc95931176)

[Git Link: 33](#_Toc95931177)

[Individual Contribution and Highlights 33](#_Toc95931178)

[Miniproject 8 – EV 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 Lock 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)

# Miniproject – 1: Phonebook Management System [Individual]

## Modules:

1. C Programming
2. Git

### Requirements

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

**Why:**

1. To store all information under a single contact number

**Where:**

1. This can be used in our daily lives to search contacts.

**Who:**

1. It can be used by each and every individual.

**When:**

1. One is in need to search their very important contact.

**How:**

1. By giving different functions one can find their desired output

### High Level Requirements

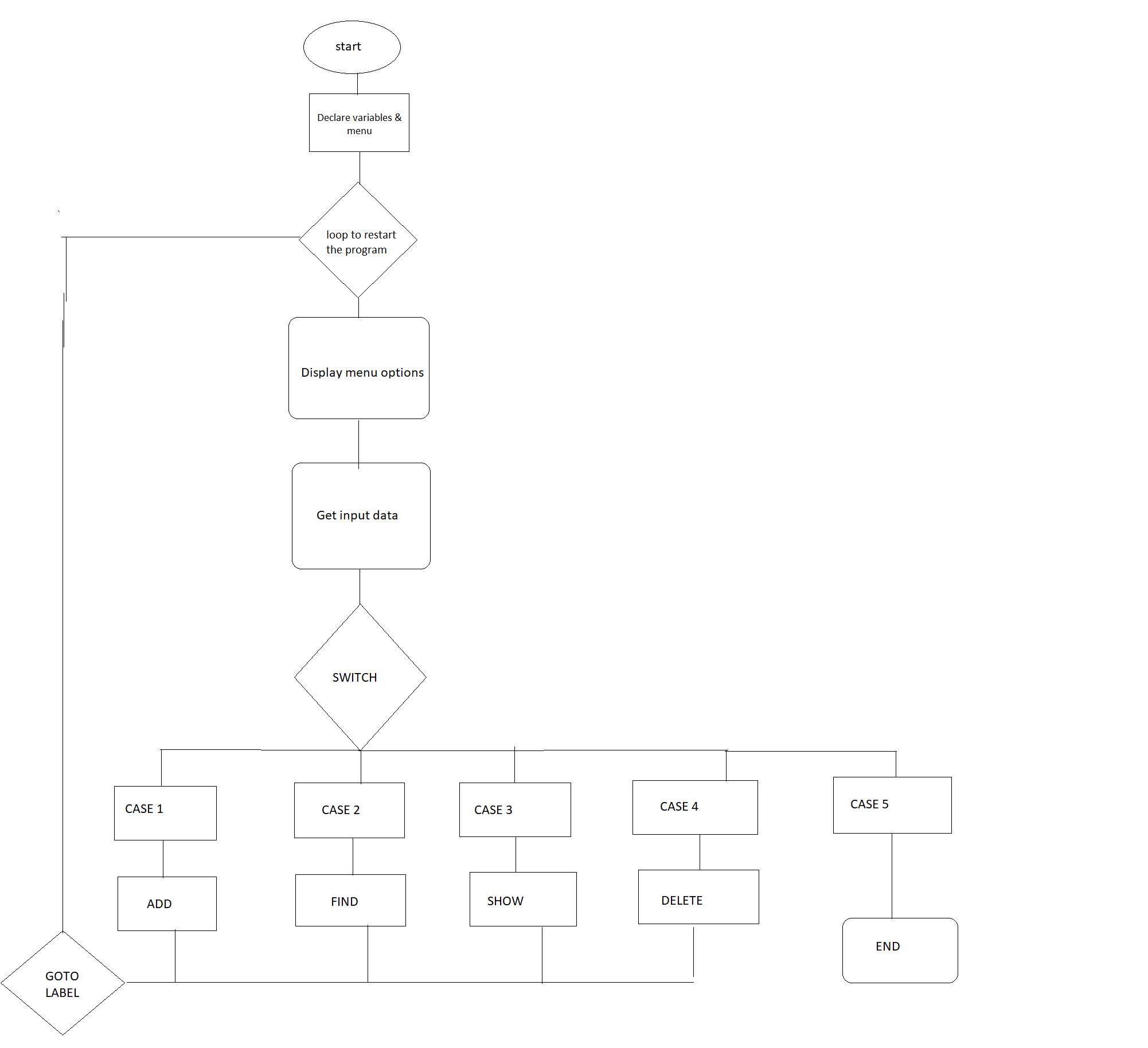
|  |  |  |
| --- | --- | --- |
| **ID** | **Description** | **Status** |
| HLR\_1 | The user can add the contact | Implemented |
| HLR\_2 | The user can search the contact | Implemented |
| HLR\_3 | The user can display the contact | Implemented |
| HLR\_4 | The user can delete the contact | Implemented |

### 

### 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** | **Status** |
| HLR\_1 | The user can add the contact | Implemented |
| HLR\_2 | The user can search the contact | Implemented |
| HLR\_3 | The user can display the contact | Implemented |
| HLR\_4 | The user can delete the contact | Implemented |

### Low Level Test Plan

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

## Implementation and Summary

### Git Link:

Link: [https://github.com/vishnukumar25/M1\_application\_Phonebook-management-system](https://github.com/vishnukumar25/M1_application_Phonebook-management-system.git)

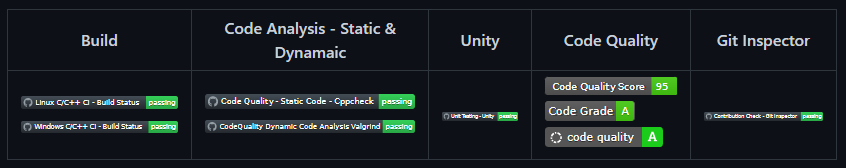
Git Dashboard 

Figure 2 Git Dashboard

### Summary

### Git Inspector Summary

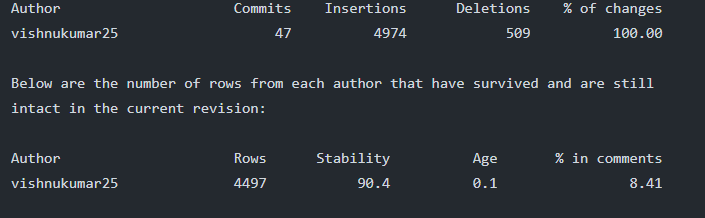


Figure 3 Git Inspector Summary

# Miniproject 2 – Functioning LED [Individual]

## Modules

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

### Requirements

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

**Why:**

1.To Function two LED's using a switch in ATmega328.

2.To understand basic concepts in ATmega328.

**Where:**

1. It can be used anywhere.

2. It can be used for understanding purposes in schools and colleges.

**Who:**

1.It can be used by students and trainees.

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

**When:**

1.People when they are in need to learn embedded programming language.

2.Mostly it can be done as projects in schools and colleges

**How:**

1.By using softwares (C programming) to exceute the program.

2.By uploading the program in ATmega328.

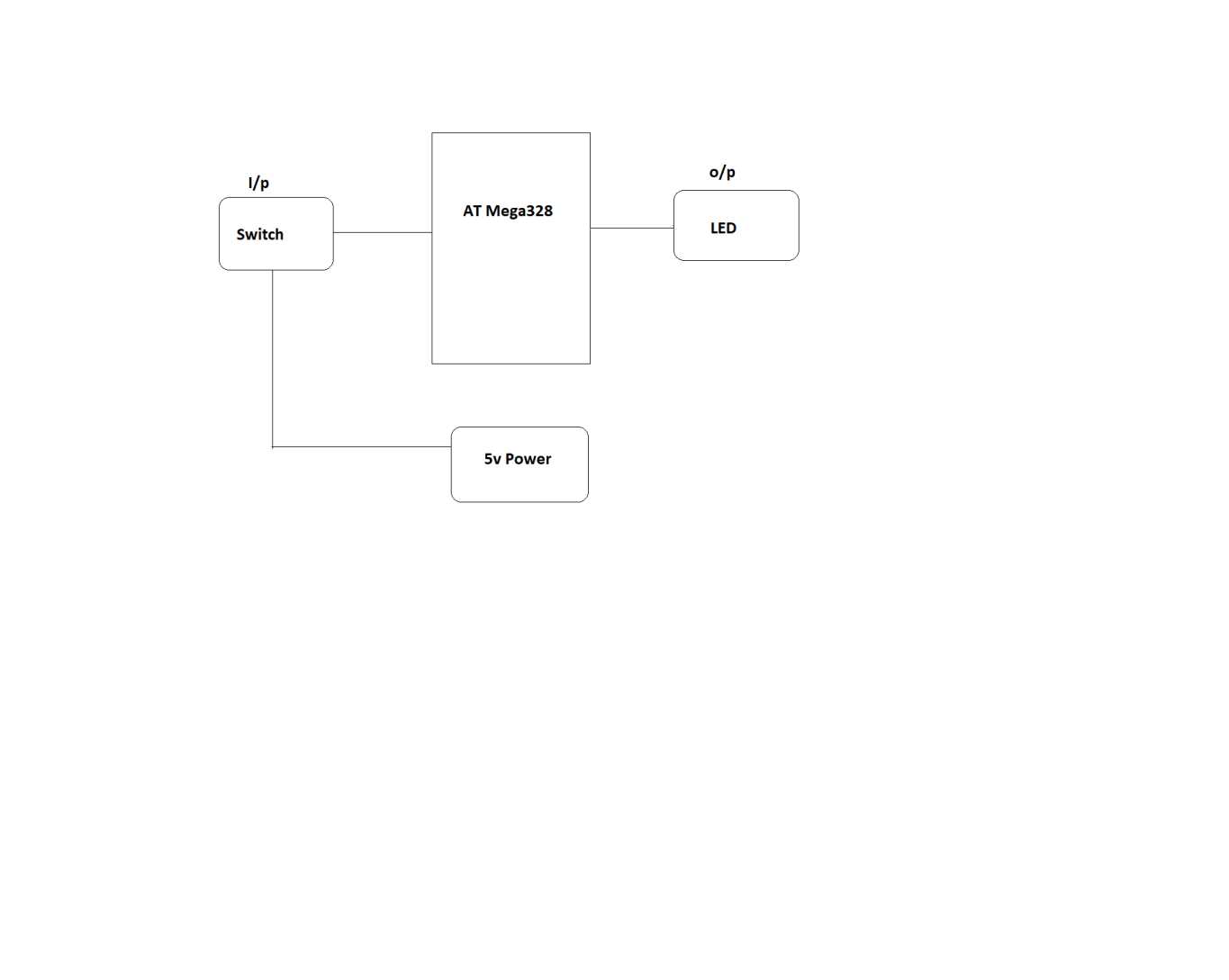
### High Level Requirements

|  |  |  |
| --- | --- | --- |
| ID | Description | Status |
| HLR\_1 | Microcontroller | Implemented |
| HLR\_2 | Switch | Implemented |
| HLR\_3 | LED | Implemented |
| HLR\_4 | Software | Implemented |

### Low Level Requirements

|  |  |  |
| --- | --- | --- |
| ID | Description | Status |
| LLR\_1 | ATmega328 | Implemented |
| LLR\_2 | Switch | Implemented |
| LLR\_3 | LED | Implemented |
| LLR\_4 | Visual studio & SimulIDE | Implemented |

## Design

Figure 4 Block Diagram

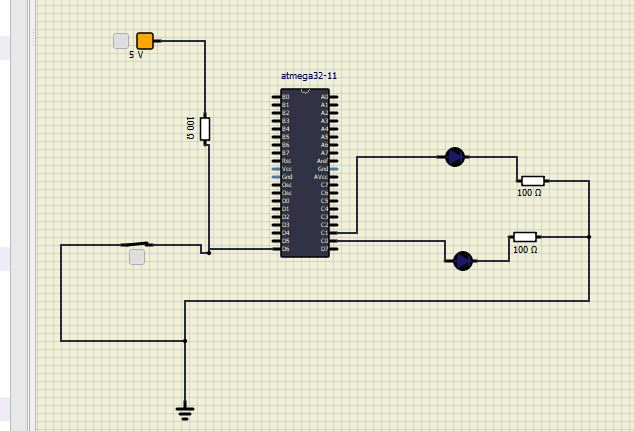


Figure 5 Simulation

## Test Plan

### High Level Test Plan

| **Id** | **Description** | **Expected I/P** | **Expected O/P** | **Actual O/P** | **Type Of Test** |
| --- | --- | --- | --- | --- | --- |
| HLR\_1 | Switch on | High power | LED On | LED On | Requirement |
| HLR\_2 | Switch Off | No power | LED Off | LED Off | Requirement |

### Low Level Test Plan

| **Id** | **Description** | **Expected I/P** | **Expected O/P** | **Actual O/P** | **Type Of Test** |
| --- | --- | --- | --- | --- | --- |
| LLR\_1 | Switch on | value 1 | LED On | LED On | Requirement |
| LLR\_2 | Switch Off | value 0 | LED Off | LED Off | Requirement |

## Implementation and Summary

### Git Link:

Link : https://github.com/vishnukumar25/M2-EMBEDDED\_Functioning\_LED.git

### Git Dashboard

Figure 6 Git Dashboard

# Miniproject 3 – AI ChatBot [Team]

## Modules

1. SDLC
2. Git

### Requirements

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

**Why:**

1. Software application used to conduct an online chat conversation via text or speech.
2. A Computer program which simulates a natural human conversation.

**Where:**

1. Retail and E-Commerce industries.
2. Used in Healthcare.

**Who**

1. Clients who need assistance.
2. Peoples who need support.

**When:**

1. To Provide faster and cheaper assistance to client.
2. To be Increasingly comfortable with Technology.

**How:**

1. Customers who are dealing with their problems late at night, chatbot are blessing as they can work around the clock.
2. During conversations with the customers, chat box provides a bridge between sales and customer team.

### High Level Requirements

| **ID** | **Description** | **Category** | **Status** |
| --- | --- | --- | --- |
| HLR\_1 | Chatterbot | Technical | Implemented |

### Low Level Requirements

| **ID** | **Description** | **HLR ID** | **Status** |
| --- | --- | --- | --- |
| LLR\_1 | Process input | HLR\_1 | Implemented |
| LLR\_2 | Logic adapter 1 | HLR\_1 | Implemented |
| LLR\_3 | Logic adapter 2 | HLR\_1 | Implemented |

## Design

Diagram

Description automatically generated

Figure 10 Behaviour Diagram

Chart, waterfall chart

Description automatically generated

Figure 11 User Flow Diagram

Diagram

Description automatically generated

Figure 12 Structure Diagram

## Test Plan

### High Level Test Plan

| **Test ID** | **Description** | **Exp I/P** | **Exp O/P** | **Actual Out** | **Type Of Test** |
| --- | --- | --- | --- | --- | --- |
| HLTP\_1 | Get input | User input | Return user input to the Process input | SUCCESS | Requirement Based |
| HLTP\_2 | Read input | Process input | Return a response related to the given User input | SUCCESS | Requirement Based |
| HLTP\_3 | Get output | Process input | Return the response from the Process input to the user | SUCCESS | Requirement Based |

### Low Level Test Plan

| **Test ID** | **HLTP ID** | **Description** | **Exp IN** | **Exp OUT** | **Actual Out** | **Type Of Test** |
| --- | --- | --- | --- | --- | --- | --- |
| LLTP\_1 | HLTP\_1 | The inputs can be given only by using console, API, speech recognition, etc. | User input | SUCCESS | SUCCESS | Requirement Based |
| LLTP\_2 | HLTP\_2 | Select a known statement that most closely matches the given User input | Process input | SUCCESS | SUCCESS | Requirement Based |
| LLTP\_2.1 | HLTP\_2 | Return a known response to the selected match and a confidence value based on the matching | Process input | SUCCESS | SUCCESS | Requirement Based |
| LLTP\_3 | HLTP\_3 | Return the response to the user only by using console, API, speech recognition, etc. | User input | SUCCESS | SUCCESS | Requirement Based |

## Implementation and Summary

### Git Link:

Link: <https://github.com/GENESIS2021Q1/Applied_SDLC-Dec_Team_1>

### Individual Contribution and Highlights

### Summary

1. Implementation
2. Testing

Role in Project Team

1. Implementation: Implemented a python code for test file.
2. Testing: Tested the Chatter Bot using spell checking.

# Miniproject 4 – Attendance Automation[Team]

## Modules

1. Python
2. Git

### Requirements

### High Level Requirements

| **ID** | **Description** | **Status** |
| --- | --- | --- |
| HLR\_1 | Attendance Status | Implemented |
| HLR\_2 | User Details | Implemented |
| HLR\_3 | User load Sheet | Implemented |
| HLR\_4 | Output File Generation | Implemented |

### Low Level Requirements

| **ID** | **Description** | **HLR ID** | **Status** |
| --- | --- | --- | --- |
| LLR\_1 | User can get the attendance status | HLR\_1 | Implemented |
| LLR\_2 | User can enter status input to get the attendance status | HLR\_1 | Implemented |
| LLR\_3 | User can get the user details | HLR\_2 | Implemented |
| LLR\_4 | User will get the details after the successful attendance | HLR\_2 | Implemented |
| LLR\_5 | User can load different sheets | HLR\_3 | Implemented |
| LLR\_6 | User can modify the existing sheets as it is dynamic | HLR\_3 | Implemented |
| LLR\_7 | Output file gets generated | HLR\_4 | Implemented |

## Test Plan

### High Level Test Plan

| **ID** | **Description** | **Expected I/P** | **Expected O/P** | **Actual O/P** | **Type Of Test** |
| --- | --- | --- | --- | --- | --- |
| HLTP\_1 | Attendance Status | User Input | SUCCESS | SUCCESS | Requirement Based |
| HLTP\_2 | User details | User Input | SUCCESS | SUCCESS | Requirement Based |
| HLTP\_3 | User load sheet | User Input | SUCCESS | SUCCESS | Requirement Based |
| HLTP\_4 | 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\_1 | HLTP\_1 | User can get Attendance Status | SUCCESS | SUCCESS | Requirement Based |
| LLTP\_2 | HLTP\_1 | User can enter Status input to get the Attendance Status | SUCCESS | SUCCESS | Requirement Based |
| LLTP\_3 | HLTP\_2 | User can get the User details | SUCCESS | SUCCESS | Requirement Based |
| LLTP\_4 | HLTP\_2 | User will get the details after the successful attendance | SUCCESS | SUCCESS | Requirement Based |
| LLTP\_5 | HLTP\_3 | User can load different sheets | SUCCESS | SUCCESS | Requirement Based |
| LLTP\_6 | HLTP\_3 | User can also modify the existing sheets as it is dynamic | SUCCESS | SUCCESS | Requirement Based |
| LLTP\_7 | HLTP\_4 | Output file gets generated | SUCCESS | SUCCESS | Requirement Based |

## Implementation and Summary

### Git Link:

Link: <https://github.com/kavinvignes/GENESIS2021-OOPS_Python-Attendance_Automation-Team_13>

### Git Dashboard

Graphical user interface, website

Description automatically generated

Figure 13 Git Dashboard

### Git Inspector Summary

Graphical user interface

Description automatically generated

Figure 14 Git Inspector Summary

# Miniproject 5 – Toyota 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. Climate Control system
6. Lane Assist System
7. Power Window

## Design

This project was implemented using Matlab.

**Implemenation and Summary**

[https://github.com/RAshwin990/Team\_Toyota](https://github.com/RAshwin990/Team_Toyota.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. |
| 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

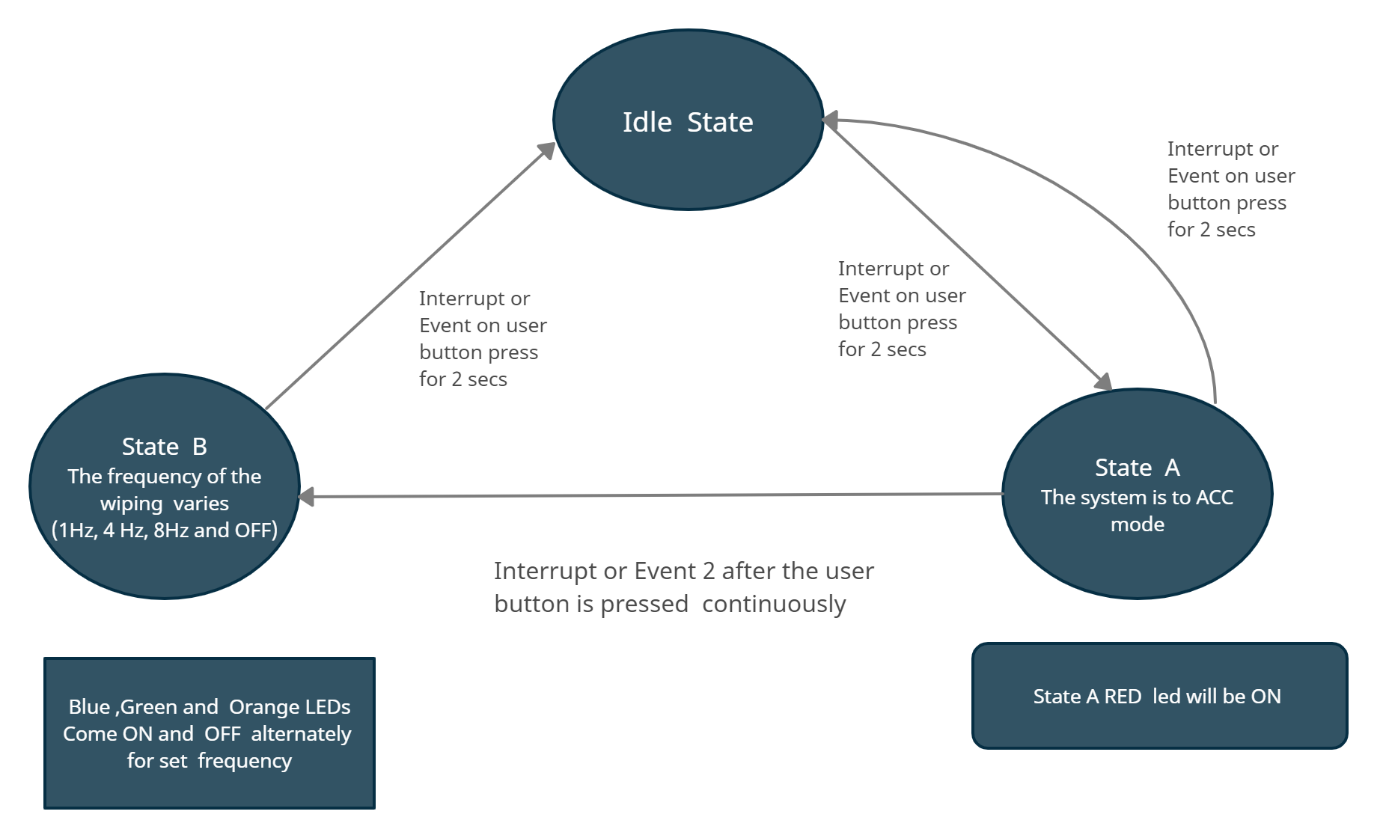


Figure 12 Structure Diagram

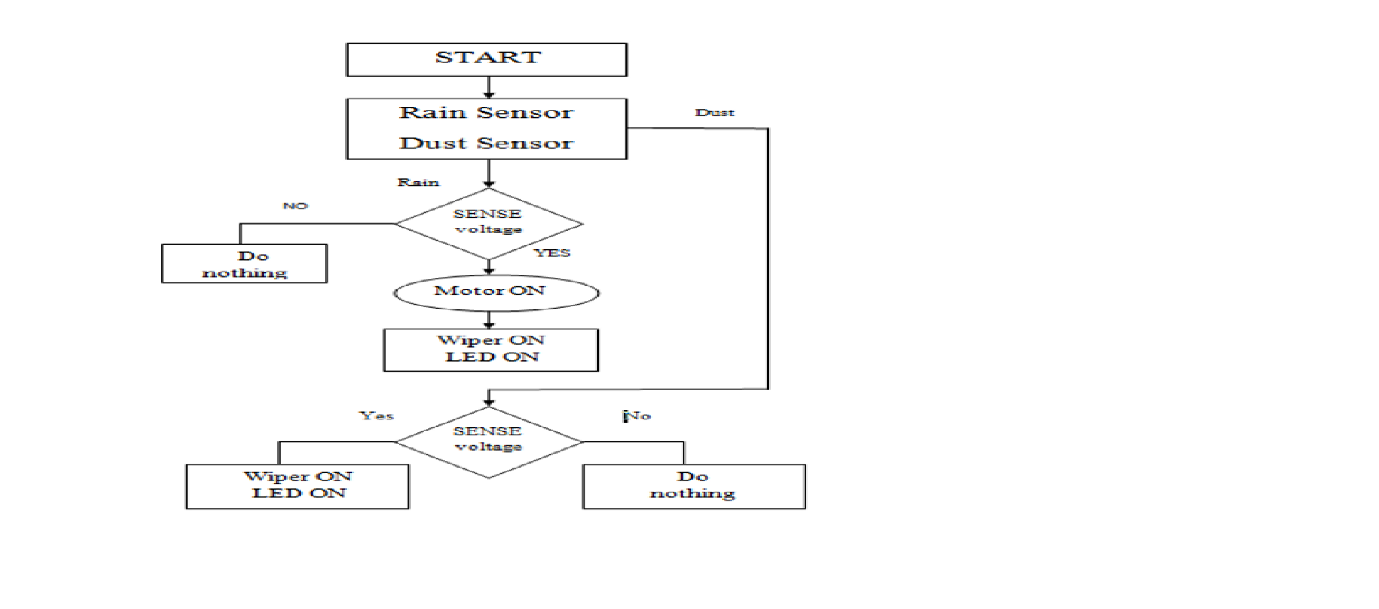


Figure 13 Behavior Diagram

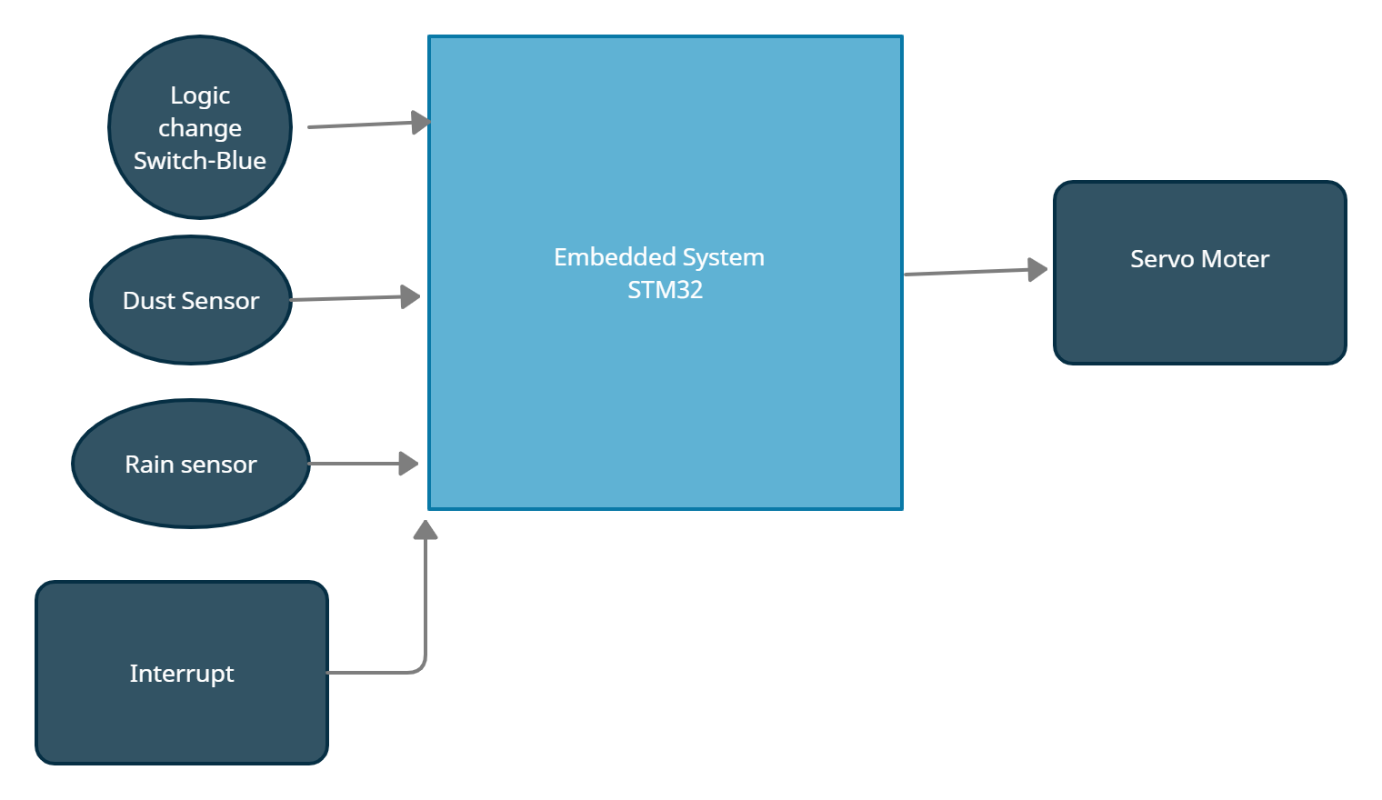


Figure 14 Block Diagram

## 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 – Hyundai Project[Team]

## Modules

1. Automotive Systems
2. Git

### Requirements

In this Jaguar project we have taken following features and I have contributed to Parking System Feature

1. Power Window.
2. Cabin Lights
3. Central Door Lock System.
4. Keyless Entry System.

## 

## Design: Cabin Lights (Individual contribution):

* This feature turns on the cabin lights when the doors are unlocked and turns off the cabin lights when the doors are locked.
* A door lock reminder flashes when the doors are not properly shut.

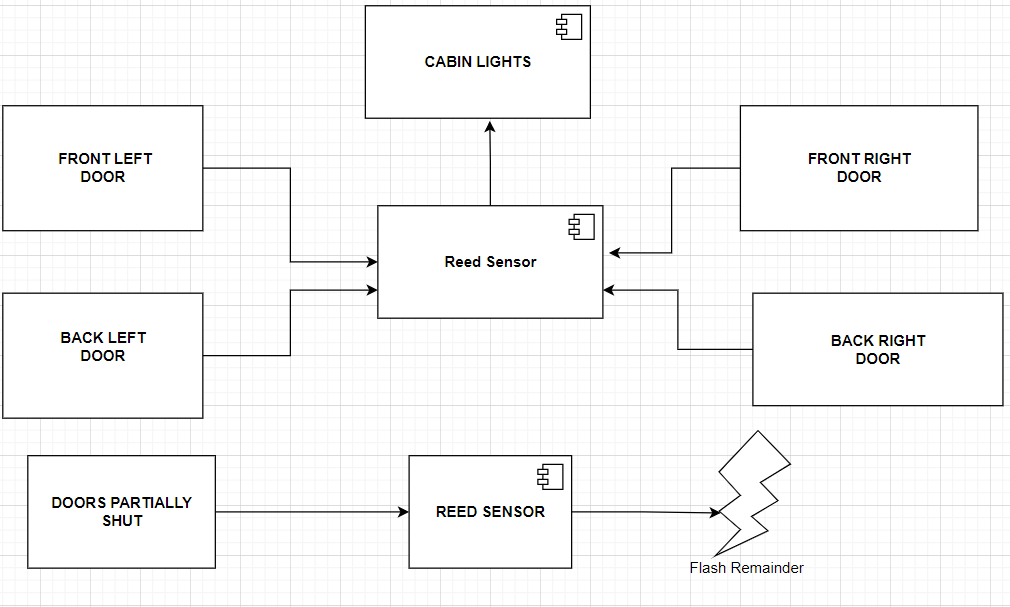


Figure 17 Structure Diagram

## Implementation and Summary

### Git Link:

Link: <https://github.com/SHANMUGAAPRIYANM/TEAM_Hyundai-.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 Keyless Entry

# Miniproject 8 – EV Golf Cart[Team]

## Modules

1. Matlab
2. Matlab Script

### 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 GEALearn

### Individual Contribution and Highlights

1. Done in Matlab Script

Role in Project Team

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

# Miniproject 9 – Cabin Lights [Individual]

## Modules

1. Autosar
2. Git

### Requirements

* This feature turns on the cabin lights when the doors are unlocked and turns off the cabin lights when the doors are locked.
* A door lock reminder flashes when the doors are not properly shut.

## Design

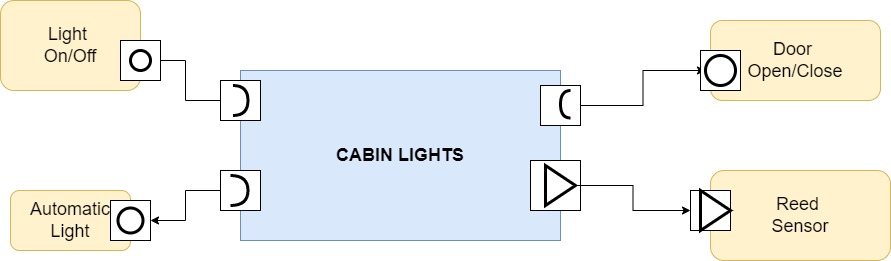


Figure 16 VFB Diagram

## Implementation and Summary

### Git Link:

Link: [https://github.com/vishnukumar25/DoorSystem\_40020488\_DPS](https://github.com/vishnukumar25/DoorSystem_40020488_DPS.git)

### Individual Contribution and Highlights

1. Studied cabin lights Case Study
2. Source code management using GitHub
3. AtomicSwComponent
4. SWCInternalBehavior
5. SWCImplementation