GENESIS - Learning Outcome & Mini-project Summary Report

LTTS
GLOBAL
ENGINEERING
ACADEMY

L&T Technology Services

Details

Ver. Rel. No.	Release Date	Prepared By	Reviewed By	To Be Approved	Remarks/Revision Details
1.0	16/02/2022	Saba Begum 40021049			

Contents

Miniproject – 1: ATM Application [Individual].....5
Modules:.....5
Requirements.....5
High Level Requirement.....7
Low Level Requirement.....8
Design.....9
Test Plan.....12
High Level Test Plan.....12
Low Level Teat Plan.....12
Implementation.....14
Git Link.....20
Mini-project -2: Heat Control System.....21
Modules.....21
Requirements.....21
Design.....22
Implementation.....24
Images and Videos.....30
Git Link: 30
Mini-project -3: Scientific Calculator.....32
Modules.....32
Requirements.....32
High Level Requirements.....33
Low Level Requirements.....34
Design.....35
High Level Design.....35
Low Level Design.....39
Implementation.....44
Test Plan.....45
Images and Videos.....47
Git Link.....53
Mini-project -4 Calendar Automation [Team].....54
Modules.....54

Requirements……....54
High Level Requirements……54
Low Level Requirements……55
Mini-Project - 5 Team-Ford…….58
Introduction…….58
Design…….59
Team Activity……...……60
Mini - project: 6 Wiper Control [Team]……..61
Modules……61
Requirements……....61
High Level Requirements……...……..62
Low Level Requirements……..……62
Design……...64
Test Plan……66
Mini-Project: 7 Golf-Cart [Team]……..……68
Requirements……..……..……69
Report……75
Miniproject 8 – Wiper System [Individual]…….……76
Modules……76
Reqirements……..76
Design……...77
Git hub link……..77

Mini-Project – 1: ATM Application [Individual]

Modules:

1. C Programming
2. Git
3. SDLC

Requirements

Introduction

An ATM is an electronic device which allows a bank's customer to make financial transactions in a public space & check their account balance at any time without the need for a human teller. The customer is identified by a plastic ATM card with a magnetic stripe or a plastic smartcard with a chip. Services available through an ATM

*Withdraw cash.

*Transfer money between accounts.

*Obtain account balance.

*Make deposits of cash and checks

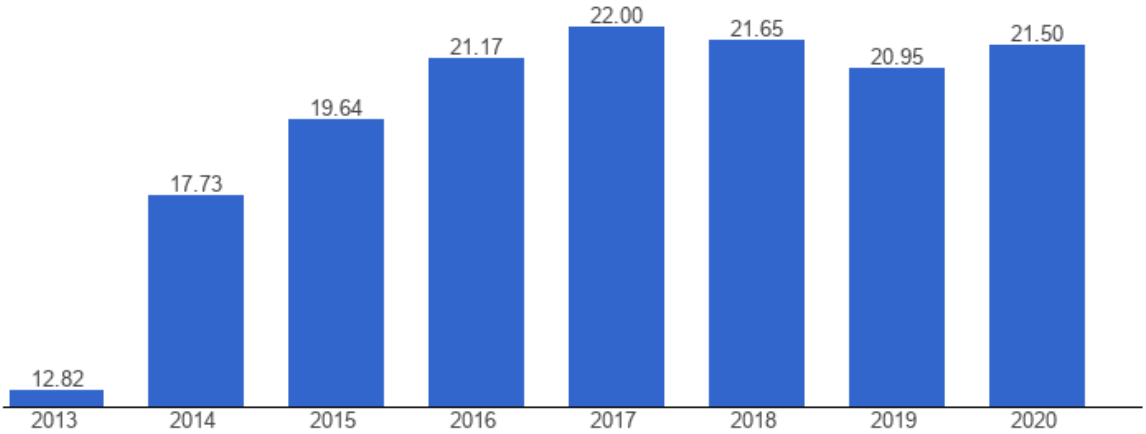
Research

Existing System:

- Existing system is manual system.
- The manual system is prone to error.
- This system involves a lot of manual entries with the application to perform a desired task.
- Usage of papers and records in the process leads to less efficiently less productivity.
- Increase lots of mistakes while writing in paper.
- Time delay between the user and customer is reduced.
- For this a new system in invented.

Proposed System: The system customer transactions, satisfies the requirements of the existing system in full-fledged manner. Through this system, customer can make fast transactions and view the last transactions easily. [https://ieeexplore.ieee.org/document/8783820]

STATE OF ART



The usage of ATMs(in 1000) increases with time(years).

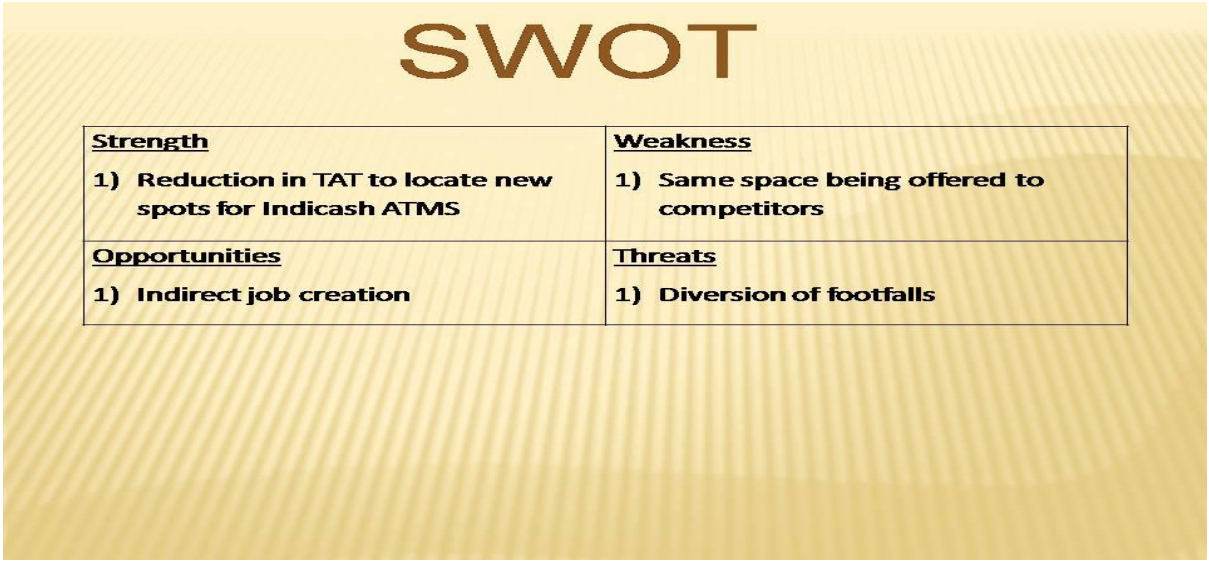
Defining Our System

Automated teller machines (ATMs) are electronic banking outlets that allow people to complete transactions without going into a branch of their bank. Some ATMs are simple cash dispensers while others allow a variety of transactions such as check deposits, balance transfers, and bill payments. The first ATMs appeared in the mid- to late-1960s and have grown in number to over 2 million worldwide. Today's ATMs are technological marvels, many capable of accepting deposits as well as several other banking services. To keep ATM fees down, use an ATM branded by your own bank as often as possible.

Features

- Generating Pin
- Use of ATM
- Transaction can be done in easily

- **SWOT ANALYSIS**



- Strength: -Take less time for efficient and effective service.
- -Easy to use -New capabilities, QR
- Weakness: -There is no plan for proper recycling of machines or used parts. -Lack of aggressive marketing strategies
- Opportunity: -Great possibility for acquisitions and strategic alliances due to strong financial position Cheap work force
- Threat: -High inflation rate when buying spare parts

4W's and 1 H's

Why:

For transaction and need of ATM user.

Where:

ATM at public place.

Who:

ATM user.

When:

As per user need.

How:

By giving input to ATM and it provide the requirements as per user need.

Detail Requirements

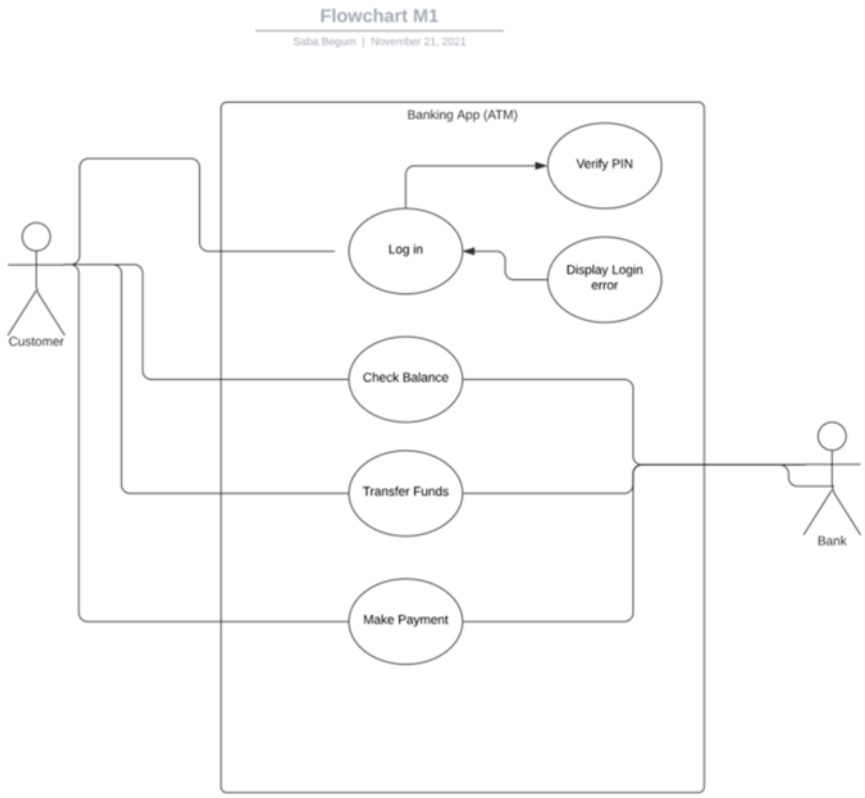
High Level Requirements

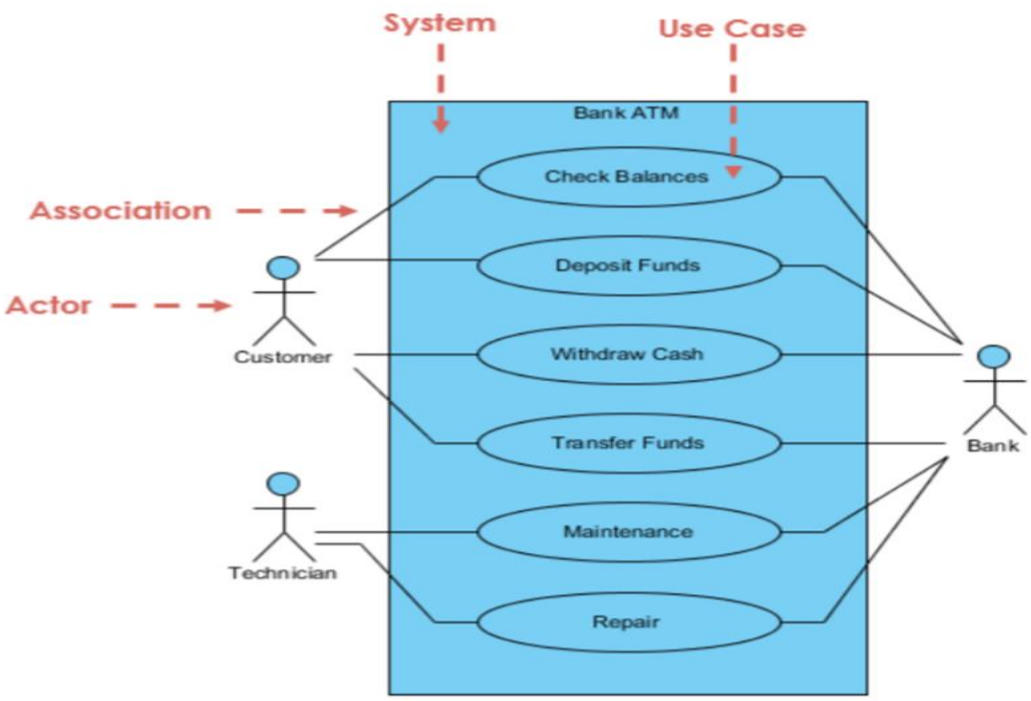
ID	Description	Status
HL1	A customer must be able to make a transfer of money between any two accounts linked to the card.	Implemented
HL2	A customer must be able to make a deposit to any account linked to the card, consisting of cash and/or checks in an envelope	Implemented

Low Level Requirements

ID	Description	Status
LL1	Pin need to be entered in 20sec.	Implemented
LL2	Keypad need to accept input from user.	Implemented

Design


Flowchart M1
Saba Begum | November 21, 2021
Banking App (ATM)
Verify PIN
Log in
Display Login error
Customer
Check Balance
Transfer Funds
Make Payment
Bank


System
Use Case
Bank ATM
Association
Actor
Check Balances
Deposit Funds
Withdraw Cash
Transfer Funds
Maintenance
Repair
Customer
Technician
Bank

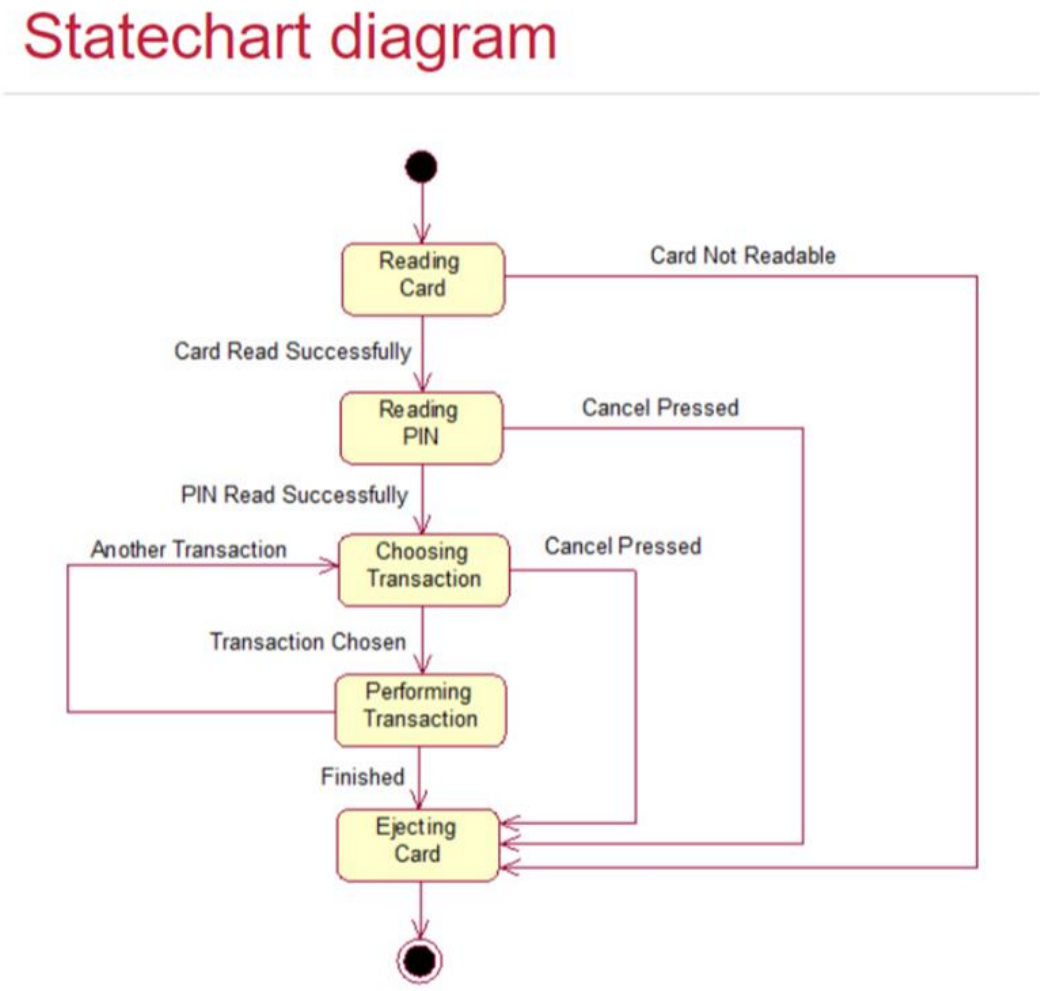
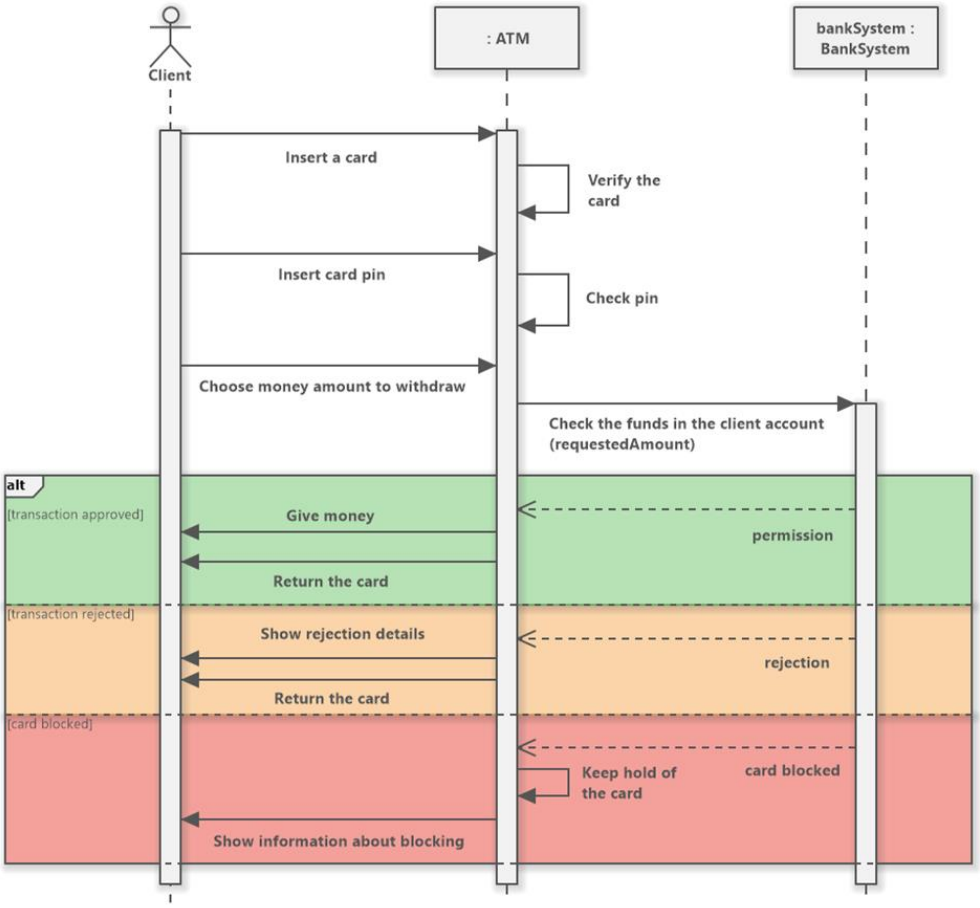


Figure 1 Behavior Diagram



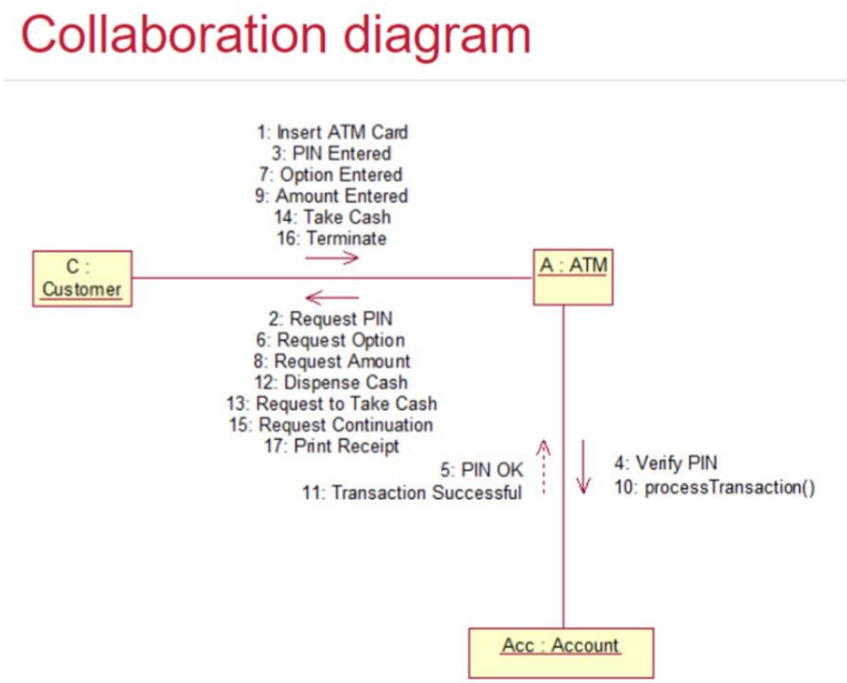


Figure 2 Structure Diagram

Test Plan

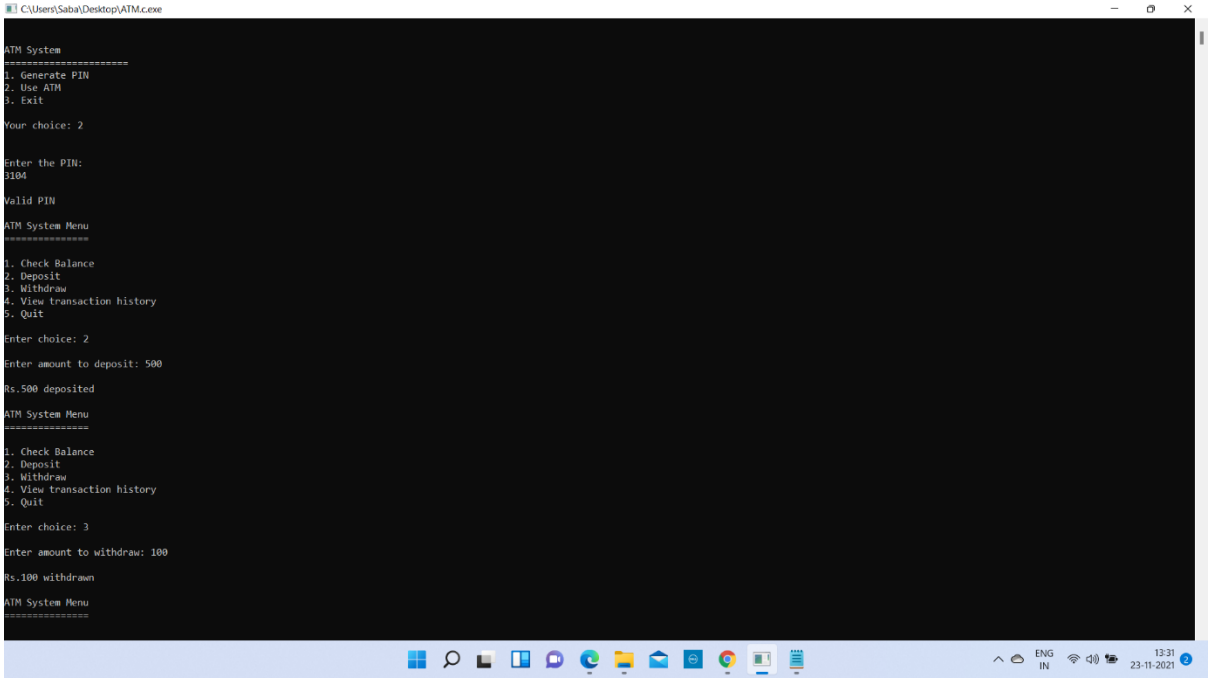
High Level Test Plan

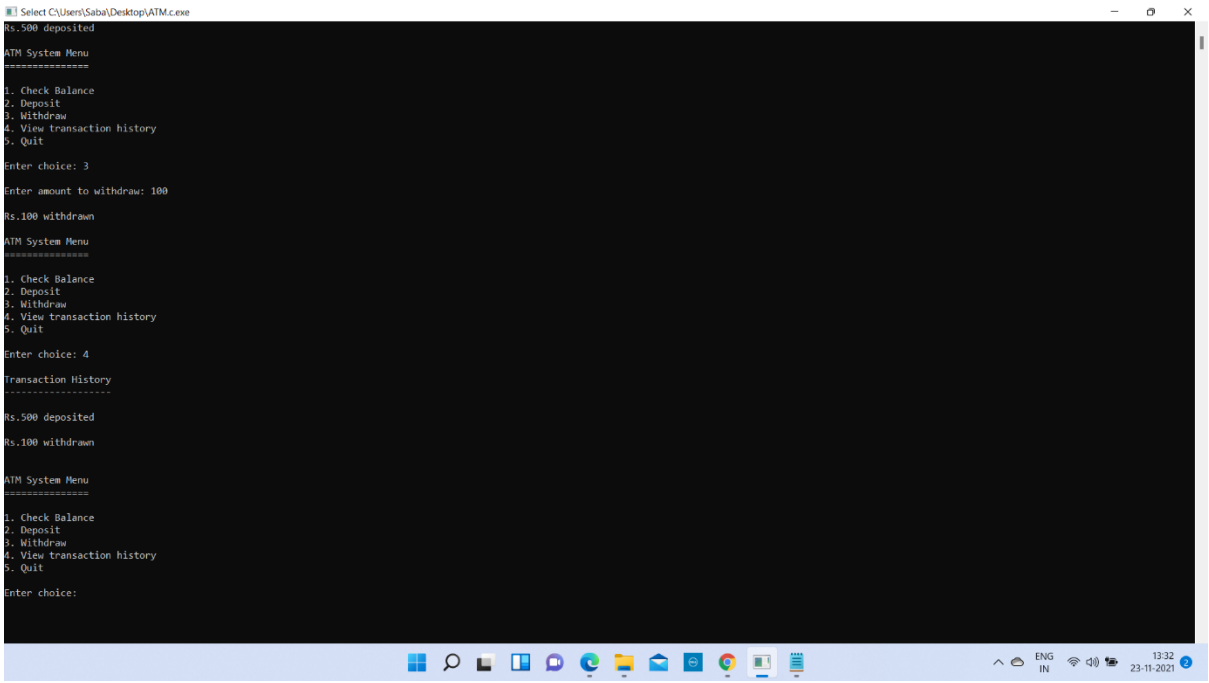
ID	Description	Feature	Result	Type Of Test
HT 1	Components Testing	Sample Input is given	Pass	Requirement Based
HT 2	Software Testing	The inputs required is given	Fail	Scenario Based
HT 3	System Testing	The inputs of the components is given to the system	Fail	Boundary Based

Low Level Test Plan

ID	Description	Feature	Result	Type Of Test
LT 1	The components required for the project is tested for the further process	Sample Input is given	Pass	Requirement Based
HT 2	The software used for the sysytem is tested and if it is passed in the test it is used for the further process	The inputs required is given	Fail	Scenario Based
HT 3	Overall system is tested and then it is approved for the next process	The inputs of the components is given to the system	Fail	Boundary Based

Output:





Features;

1 Simple Interface

2 Accurate output with required values

3 User can check balance

4 User can withdraw money

5 User can deposit money

Implementation

Main.c

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

/*
  node structure for implementing a queue to store transaction history
*/
typedef struct node {
    char statement[50];
    struct node* link;
} node;

/*
  ATM function prototypes
*/
void pinGeneration(void);
int checkPin(void);
void showBalance(int *);
void depositMoney(node **, int *);
void withdrawMoney(node **, int *);
void saveHistory(node **, char *);
void removeHistory(node **);
void showHistory(node **);

int main(void) {
    int choice1, choice2;
    int pinValid = 0, balance = 0;
```

```
    node *head = NULL;

    while (1) {
        printf("\n\nATM System\n=====================\n");
        printf("1. Generate PIN\n2. Use ATM\n3. Exit\n");
        printf("\nYour choice: ");
        scanf("%d", &choice1);

        switch (choice1) {
            case 1: pinGeneration();
                    exit(EXIT_SUCCESS);

            case 2: pinValid = checkPin();

                    if (pinValid) {
                        printf("\nValid PIN\n");
                    } else {
                        printf("\nInvalid PIN. Please generate a PIN if you
don't have one.\n");
                        exit(EXIT_FAILURE);
                    }

                    /*
                                                On valid PIN entry by user,
the ATM Menu is presented to the user
                                                */
                    while(pinValid) {
                        printf("\nATM System
Menu\n===============\n\n");
                        printf("1. Check Balance\n2. Deposit\n3.
Withdraw\n4. View transaction history\n5. Quit\n\n");

                        printf("Enter choice: ");
                        scanf("%d", &choice2);

                        switch(choice2) {
                            case 1: showBalance(&balance);
                                    break;
                            case 2: depositMoney(&head, &balance);
                                    break;
                            case 3: withdrawMoney(&head, &balance);
                                    break;
                            case 4: showHistory(&head);
                                    break;
                            case 5: printf("\nThank you for using the ATM\n");
```

```
                        exit(EXIT_SUCCESS);
                    default: printf("\nInvalid option entered!\n");
                        break;
                }
            }

        case 3: exit(EXIT_SUCCESS);

        default: printf("\nInvalid choice...Try again...\n");
            break;
    }
  }
  return 0;
}

/*
 This function will search for the PIN entered by the user
 in the file where the PIN numbers are stored
 If PIN is found return 1
 Otherwise, return 0
*/
int checkPin(void) {
  FILE *fp;
  // buffer to read PIN and store from file
  char pin[8];
  // buffer to read PIN and store from user
  char keyPin[8];
  int pinValid = 0;

  printf("\n\nEnter the PIN: \n");
  scanf("%s", keyPin);

  fp = fopen("pin.txt", "r");
  if (NULL == fp) {
    printf("\nFile cannot be opened\n");
    exit(EXIT_FAILURE);
  }

  /*
   Search for the PIN entered by user in file pin.txt
  */
  while (fgets(pin, sizeof(pin), fp) != NULL) {
    if (strstr(pin, keyPin)) {
      pinValid = 1;
    }
  }
```

```
    fclose(fp);

    return pinValid;
}

/*
  This function will generate a 4-digit random number that
  is considered as PIN
*/
void pinGeneration(void) {
    FILE *fp;

    /*
            Generate a random 4 digit number
          */
    srand(time(NULL));
    int generatedPin = 1000+rand()%9000;

    printf("\nPIN generated successfully\n");
    printf("\nYour generated PIN: %d\n", generatedPin);
    printf("\nRe-run the program and use ATM with this PIN\n\n");

    fp = fopen("pin.txt", "a");
    if (NULL == fp) {
        printf("\nCannot open file!");
        exit(EXIT_FAILURE);
    }

    /*
      Write PIN to the file
    */
    fprintf(fp, "%d\n", generatedPin);
    fclose(fp);
}

/*
  This function will display the current balance amount
*/
void showBalance(int *balance) {
    printf("\nYour current balance is Rs.%d\n", *balance);
}

/*
  This function will add the money deposited to the balance
*/
void depositMoney(node **head, int *balance) {
```

```
    int depositAmount;
    /*
      buffer to store
    */
    char depositStmt[50];

    printf("\nEnter amount to deposit: ");
    scanf("%d", &depositAmount);

    if (depositAmount > 0) {
        *balance += depositAmount;
        printf("\nRs.%d deposited\n", depositAmount);

                    /*
                      saving formatted string in depositStmt character
array
                    */
        snprintf(depositStmt, sizeof(depositStmt), "Rs.%d deposited\n",
depositAmount);
        saveHistory(head, depositStmt);
    } else {
        printf("\nInvalid amount entered\n.");
    }
}

/*
  This function will deduct the money withdrawn from the balance
*/
void withdrawMoney(node **head, int *balance) {
    int withdrawAmount;
    char withdrawStmt[50];

    printf("\nEnter amount to withdraw: ");
    scanf("%d", &withdrawAmount);

    if (withdrawAmount > 0) {
        if (withdrawAmount > *balance) {
            printf("\nCannot withdraw. Balance Rs.%d\n", *balance);
        } else {
            *balance -= withdrawAmount;
            printf("\nRs.%d withdrawn\n", withdrawAmount);

            /*
                                saving formatted string in withdrawStmt
character array
                            */
```

```
            snprintf(withdrawStmt, sizeof(withdrawStmt), "Rs.%d
withdrawn\n", withdrawAmount);
            saveHistory(head, withdrawStmt);
        }
    } else {
        printf("\nInvalid amount entered\n.");
    }
}

/*
  This function will save a transaction statement
*/
void saveHistory(node **head, char *str) {
    static int count = 0;
    node *temp;
    temp = (node *)malloc(sizeof(node));

    strcpy(temp->statement, str);
    temp->link = NULL;

    if (NULL == *head) {
        *head = temp;
        count++;
    } else {
        if (10 == count) {
            removeHistory(head);
            count--;
        }
        node *p;
        p = *head;
        while (NULL != p->link) {
            p = p->link;
        }
        p->link = temp;
        count++;
    }
}

/*
  This function is used to remove the oldest transaction when
  10 transactions are made
*/
void removeHistory(node **head) {
    node *temp;
    temp = *head;
    *head = (*head)->link;
```

```
    temp->link = NULL;
    free(temp);
}

/*
  This function will display the transaction history
*/
void showHistory(node **head) {
    node *temp;
    temp = *head;

    if (NULL == temp) {
        printf("\nNo transaction history...\n");
    } else {
        printf("\nTransaction History\n-------------------\n\n");
        while (NULL != temp) {
            printf("%s\n", temp->statement);
            temp = temp->link;
        }
    }
}
```

Challenges Faced and How Was It Overcome

1. Working with vsc, learnt from training.
2. Debugging learnt from buddy team.
3. Multifile, got good knowledge from course material.
4. Building workflow found to be difficult task, learnt from online source.

Learning Resources

1. markdownCheatsheet
2. markdownBasics
3. git inspector
4. github workflow

Implementation and Summary

Git Link:

Link: [GitHub - sababegums/M1_App_ATM]

Mini-project 2 – Heat Control System [Individual]

Modules

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

Theory

The heat control system is basically used to control the temperature of a car seat. When a user or driver of the car gets seated on a car, the button sensor gets activated. After that, the user gets access to turn on the heater. The temperature sensor keeps monitoring the temperature and sends the analog value to the microcontroller. The microcontroller processes the analog input of the temperature sensor and outputs a temperature value through serial communication. All the activities of the control system are done on a microcontroller called Atmega328.

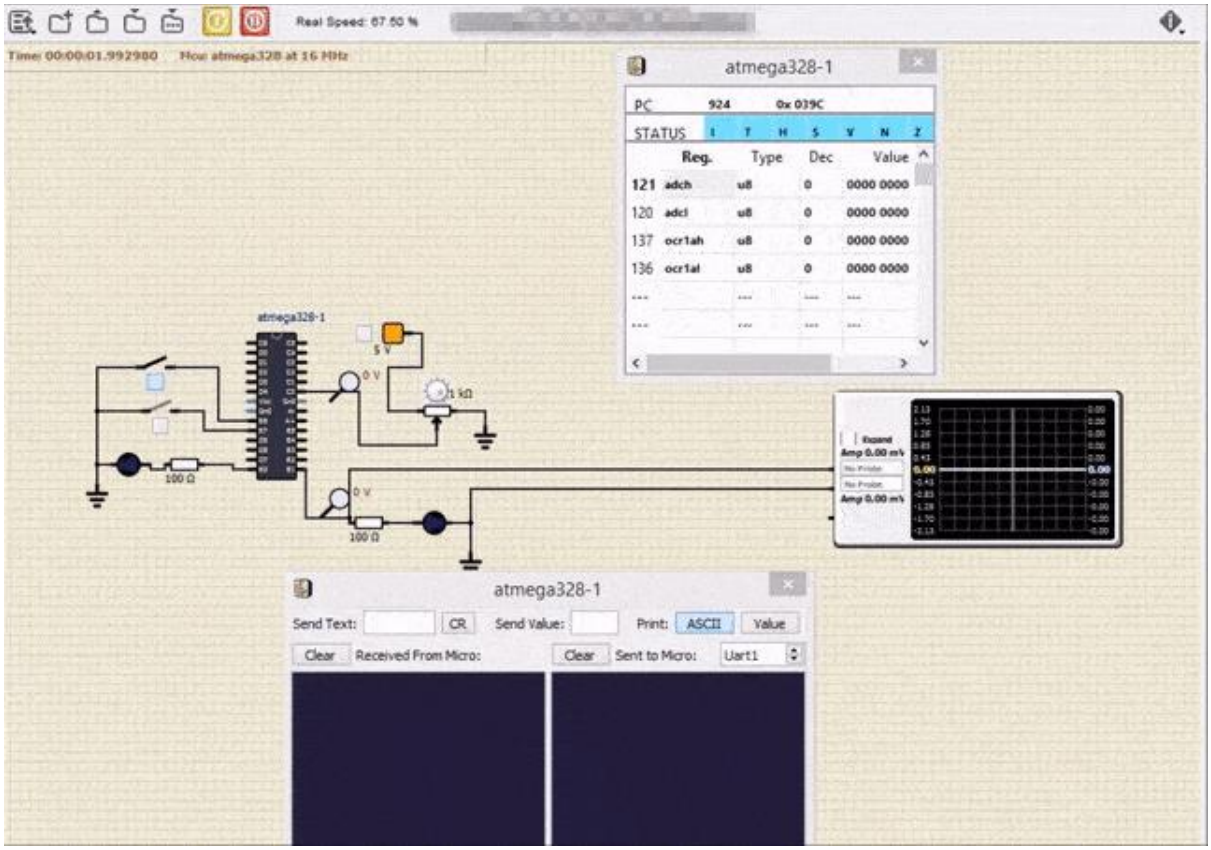
- In the places where snow falls or temperature is less,that is around zero degree centigrade, people feel comfort if the car seats are warm while travelling.
- In this project Microcontroller ATMega 328 is used to control seat temperature.
- Program is designed like Heater starts operation only if person is sitting on seat and heater switch is closed.
- When both above conditions are true Temperature sensor starts sensing tempraure of seat.
- If seat temperature is more than 25 deg centigrade,there will be no supply to heater because seat is warm enough.
- If temperature is less than 25 deg PWM output drives the heater driver circuit.
- PWM output and temperature values are inversely proportional i,e lesser the temperature more value of PWM signal will drive the heater.
- For different range of temperature values different PWM value signal is generated.
- When temperature rises above 25 deg,PWM signal will be 0(low).
- All this opeartion is automatic and also temperature range is transmitted to serial output.

Simulation

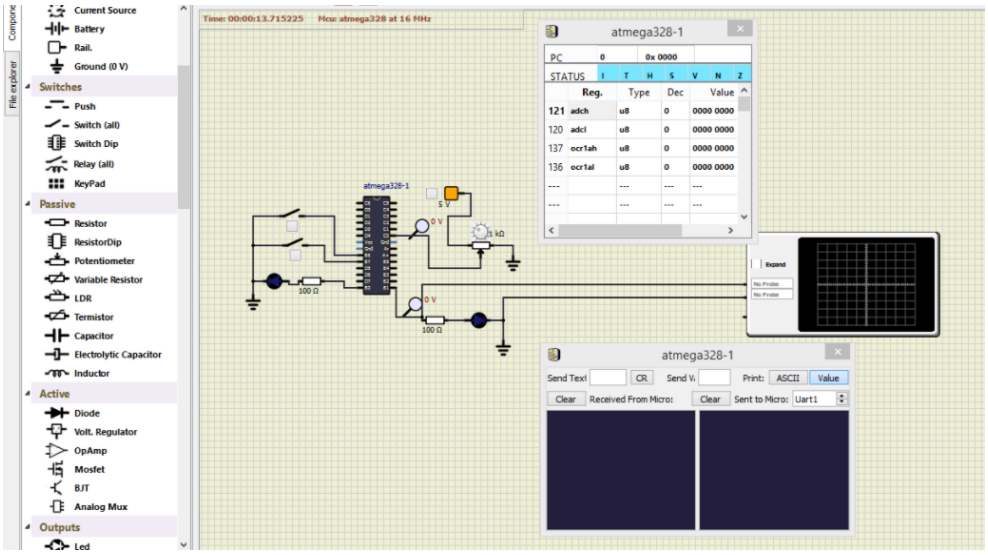
The functionality of the heat control system is coded in embedded c and the working is demonstrated using simuation in a software called SimulIDE. Below shows two images where in the 1st image shows the status of the simulation when the system is ON and the second image shows the status of the system

Design

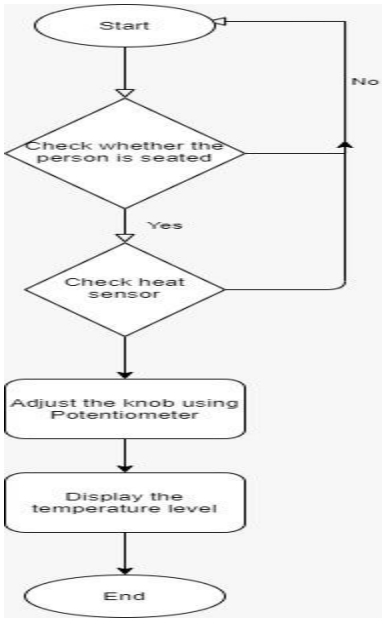
ON



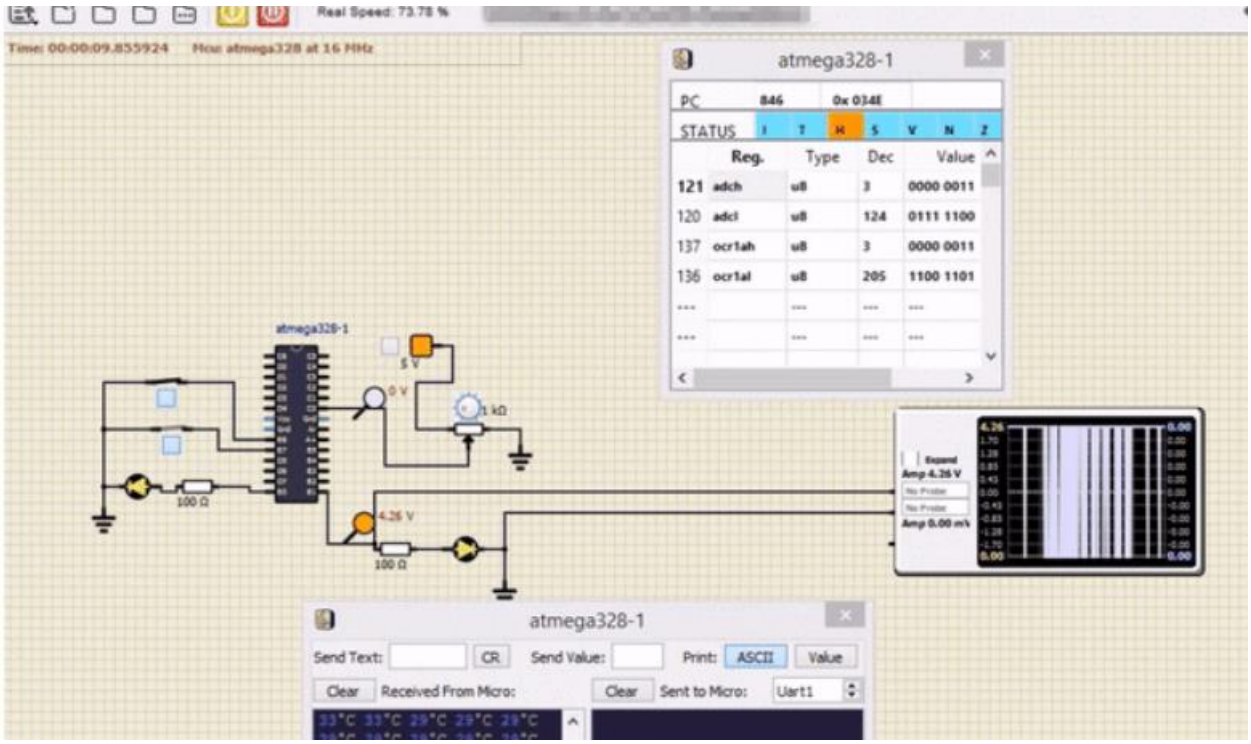
OFF



Design: Structure Diagram



Behaviour Diagram



Implementation:

Code:

Activity1.h

```
#ifndef
__ACTIVITY_1_H__
#define __ACTIVITY_1_H__
/**
 * @file activity1.h
 * @brief Header file for activity1
 *
 */


/**
 * Macro Definitions
 */

#define F_CPU 16000000UL            /**< Clock Frequency of
MCU is 16 MHz */
#define LED_PORT (PORTD)    /**< LED Port Number */
#define LED_PIN  (PORTD2)   /**< LED Pin number  */
#define BUTTON_SENSOR  (PORTD0)   /**< Port for Button
Sensor  */
#define TEMP_SENSOR  (PORTD1)   /**< Port for
Temperature Sensor   */

/**
 * Include files
 */
#include <util/delay.h>
#include <avr/io.h>


/**
 * Function Definitions
 */

/**
 * @brief Initialize all the Peripherals and pin configurations
 *
 */
void peripheral_init(void);
```

```
/**
 * @brief Function to turn LED on
 *
 */
void TurnLED_ON();

/**
 * @brief Function to turn LED off
 *
 */
void TurnLED_OFF();

/**
 * @brief Activity 1 to Turn ON LED if switches for Button
sensor and Heator sensor are pressed
 *
 */
int activity1_LED(void);

#endif /* __ACTIVITY_1_H__ */
```

Activity2.h

```
#ifndef
__ACTIVITY_2_H__
#define __ACTIVITY_2_H__
/**
 * @file activity2.h
 *
 */

/**
 * Include files
 */
#include <util/delay.h>
#include <avr/io.h>

/**
 * Function Definitions
 */

/**
 * @brief Initialize the registers for ADC setup
 *
 */
void InitADC();
```

```
/**
 * @brief Gives the ADC values for the selected channel
 *
 * @param[in] ch Channel number from 0-7
 * @return uint16_t ADC value
 */
uint16_t ReadADC(uint8_t ch);

/**
 * @brief Return the ADC value to main function
 *
 * @return uint16_t The ADC value
 */
uint16_t activity2_GetADC(void);

#endif /* __ACTIVITY_2_H__ */
```

Activity3.h

```
#ifndef
__ACTIVITY_3_H__
#define __ACTIVITY_3_H__
/**
 * @file activity3.h
 *
 */

/**
 * Macro Definitions
 */
#define PWM_20_PERCENT (205) /**< Output PWM for 20%
duty cycle */
#define PWM_40_PERCENT (410) /**< Output PWM for 40%
duty cycle */
#define PWM_70_PERCENT (717) /**< Output PWM for 70%
duty cycle */
#define PWM_95_PERCENT (973) /**< Output PWM for 95%
duty cycle */

/**
 * Include files
 */
#include <util/delay.h>
```

```
#include <avr/io.h>

/**
 * Function Definitions
 */

/**
 * @brief Initialize Timer1 registers
 *
 */
void InitTimer();

/**
 * @brief Produce duty cycle in PWM according to i/p ADC
value
 *
 * @param[in] temp The ADC value from activity2
 */
void activity3_PWM(uint16_t temp);

#endif /* __ACTIVITY_3_H__ */
```

Activity4.h

```
#ifndef
__ACTIVITY_4_H__
#define __ACTIVITY_4_H__
/**
 * @file activity4.h
 *
 */

/**
 * Macro Definitions
 */
#define F_CPU 16000000UL /**< Clock Frequency of MCU is
16 MHz */
#define USART_BAUDRATE 9600 /**< Baud rate for serial
communication  */
#define UBRR_VALUE (((F_CPU / (USART_BAUDRATE *
16UL))) - 1) /**< Formula to calculate UBRR value */

/**
 * Include files
 */
#include <util/delay.h>
#include <avr/io.h>
```

```
/**
 * Function Definitions
 */

/**
 * @brief Setting Baud rate and Enabling Rx and Tx
 *
 */
void USARTInit();

/**
 * @brief Reading data
 *
 * @return int The read data
 */
int USARTRead();

/**
 * @brief Writing data to view in serial monitor
 *
 * @param[in] temp The ADC value
 */
void activity4_USARTWrite(uint16_t temp);

#endif /* __ACTIVITY_4_H__ */
```

main.c

```
/**
 * @file SeatHeatingApp.c
 *
 */

#include "activity1.h"
#include "activity2.h"
#include "activity3.h"
#include "activity4.h"

int main(void)
{
   uint16_t temp;

   while(1)
   {
```

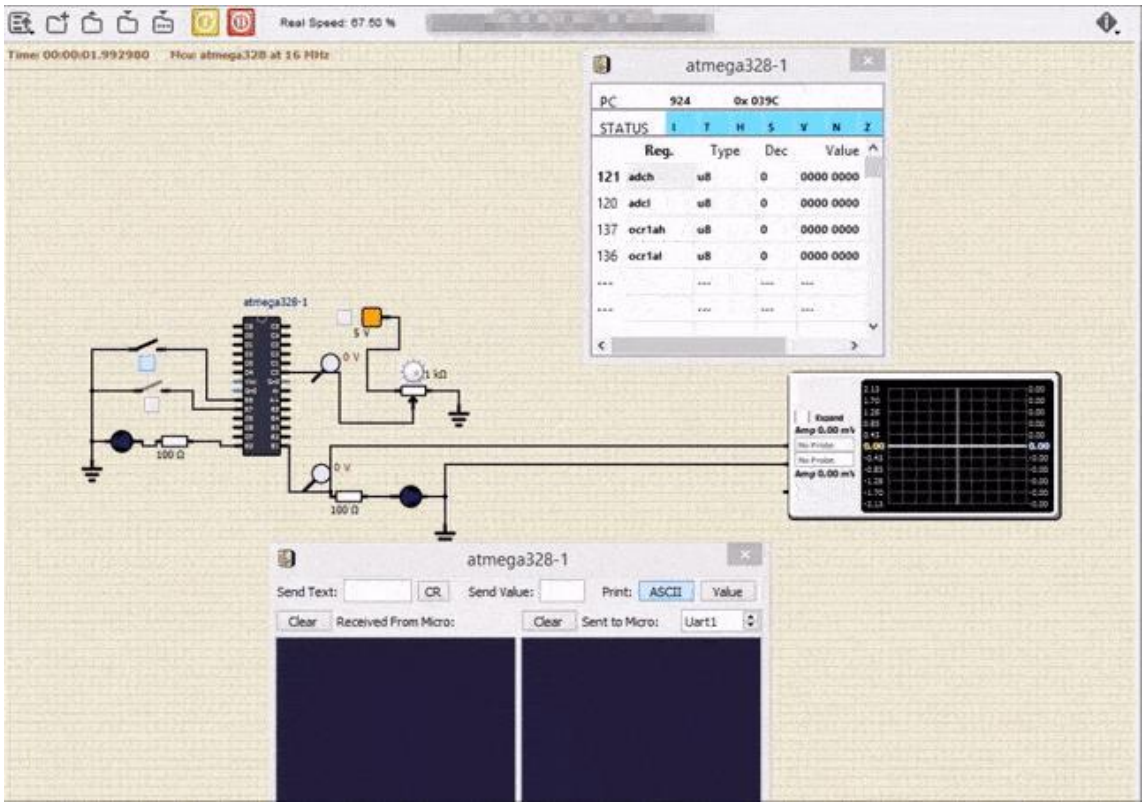
```
        if(activity1_LED()==1) //Check if both the switches are pressed
        {

            TurnLED_ON();//Turn LED ON
            temp=activity2_GetADC(); //Get the ADC value
            activity3_PWM(temp); //PWM output based on temperature
                        activity4_USARTWrite(temp); //To Serial monitor to print
Temperature


        }
        else  //in all other cases
        {
            TurnLED_OFF();//Turn LED OFF
                        _delay_ms(200);
        }

    }
    return 0;
}
```

Images and Videos:

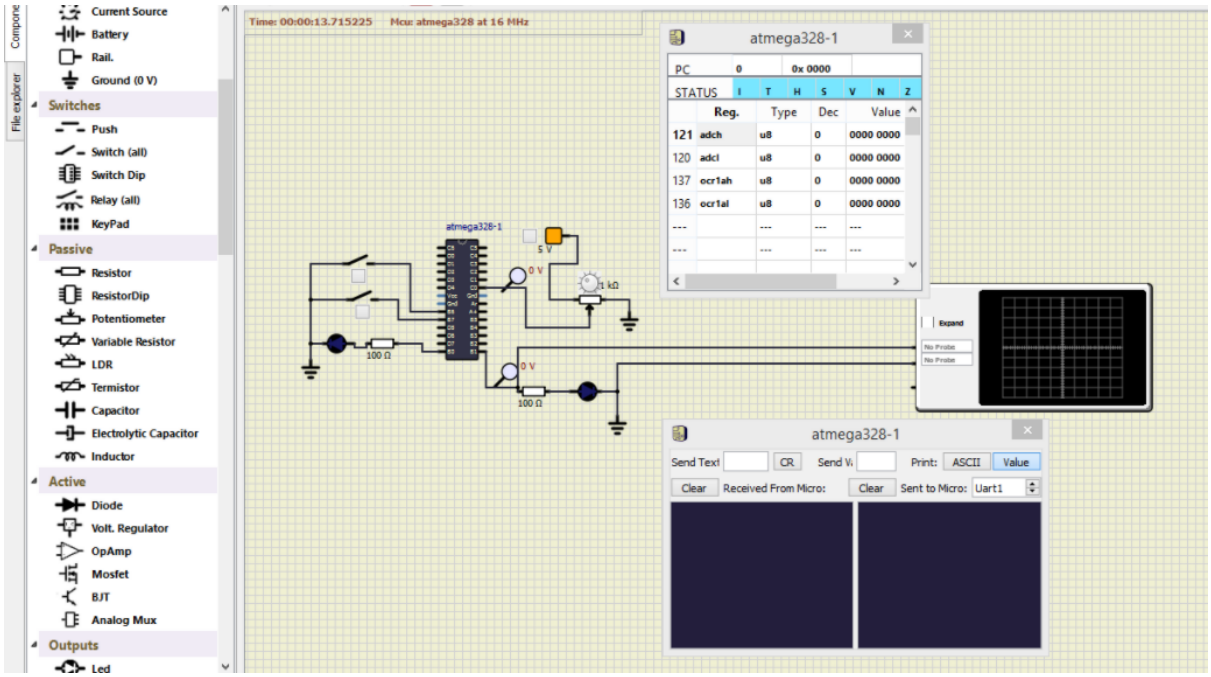
Simulation

The functionality of the heat control system is coded in embedded c and the working is demonstrated using simuation in a software called SimulIDE. Below shows two images where in the 1st image shows the status of the simulation when the system is ON and the second image shows the status of the system when it is OFF

ON



OFF



Implementation and Summary

Git Link:

Link: GitHub - sababegums/M2-Embedded_SEAT-HEAT-MONITOR-AND-CONTROL-SYSTEM

Mini-Project – 3: Scientific Calculator [Team]

Modules:

1. C Programming
2. Git
3. SDLC

Requirements

Introduction

It is a advanced calculator that will allow users to perform operations in mathematics (Arithimetic, Trigonometric, Matrices, Conversion etc) . However, the input has to be provided by user. The input values can be from any integer to even a number with decimals. Moreover, this calculator is smart enough to operate all the operations.

Features

- It can do all operations (Addition, Subtraction, Multiplication, Division).
- Logarithmic operations, Exponential operations are also available.
- Power functions, Factorial and Conversions which are helpful for students are added.
- Basic Trigonometric Operations are also available.
- Basic Matrix Operations are also available.
- It has double precision.

Cost v/s features

Casio FX (Programmable)	Casio FX (Non-color Programmable)	Casio FX (color Programmable)
Casio FX (240 functions)	Casio FX (300 functions)	Casio FX (522 functions)
10 Digit Casio	12 digit Casio	12 digit robust Casio

COST

FEATURES

SWOT ANALYSIS

SWOT- Strengths, and Weakness, Opportunities Threats

Strengths

• User Friendly

• All basic operations

• Double Precision

• Trigonometric operations

• Matrix Operations

Weakness

• Limited Operations

• Memory Wastage

Opportunities

• It can be expanded by adding additional features like Inverse Trigonometric operations, Equations etc.

Threats

• There are other programmable calculators which may affect our product marketing.

4W's and 1'H

4W1H

What : Scientific Calculator.

Where : Used in Provisional Stores and Students.

When : For Complex Calculations.

Why : For Easy and faster Calculations.

How : Operates by using User Inputs.

Detail Requirements

High Level Requirements

High level Requirement	**Description**
HLR1	Basic Operations
HLR2	Trigonometric Operations
HLR3	Matrix Operations
HLR4	Conversions
HRL5	Advanced operations

Low-Level Requirements

Low Level Requirement	**Description**	**Related HLR**
LLR1	Addition	HLR1
LLR2	Subtraction	HLR1
LLR3	Multiplication	HLR1
LLR4	Division	HLR1
LLR5	Sine	HLR1

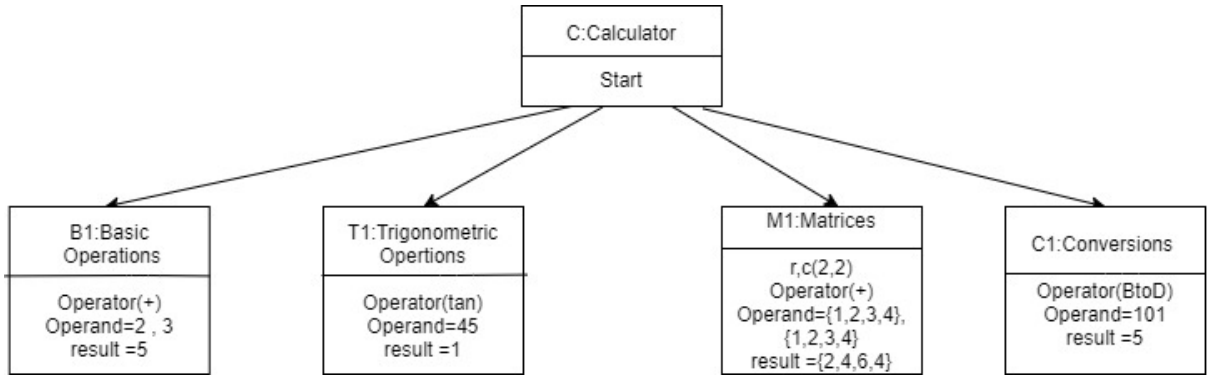
LLR6	Cos	HLR1
LLR7	Tan	HLR1
LLR8	Cosec	HLR1
LLR9	sec	HLR1
LLR10	Cot	HLR1
LLR11	matrix addition	HLR1
LLR12	matrix subtraction	HLR1
LLR13	matrix multiplication	HLR1
LLR14	Binary - Decimal Conversion	HLR1
LLR15	Decimal - Binary Conversion	HLR1
LLR16	Decimal - Octal Conversion	HLR1
LLR17	Octal - Decimal Conversion	HLR1
LLR18	Length unit Conversion	HLR1
LLR19	Temperature unit Conversion	HLR1
LLR20	Current AC-DC Conversion	HLR1
LLR21	Log	HLR1
LLR22	Exponential	HLR1
LLR23	Modulus (remainder)	HLR1
LLR24	Factorial	HLR1
LLR25	Square root	HLR1

LLR26	Cube root	HLR1
LLR27	LCM	HLR1
LLR28	GCD	HLR1
LLR29	Permutation	HLR1
LLR30	Combination	HLR1

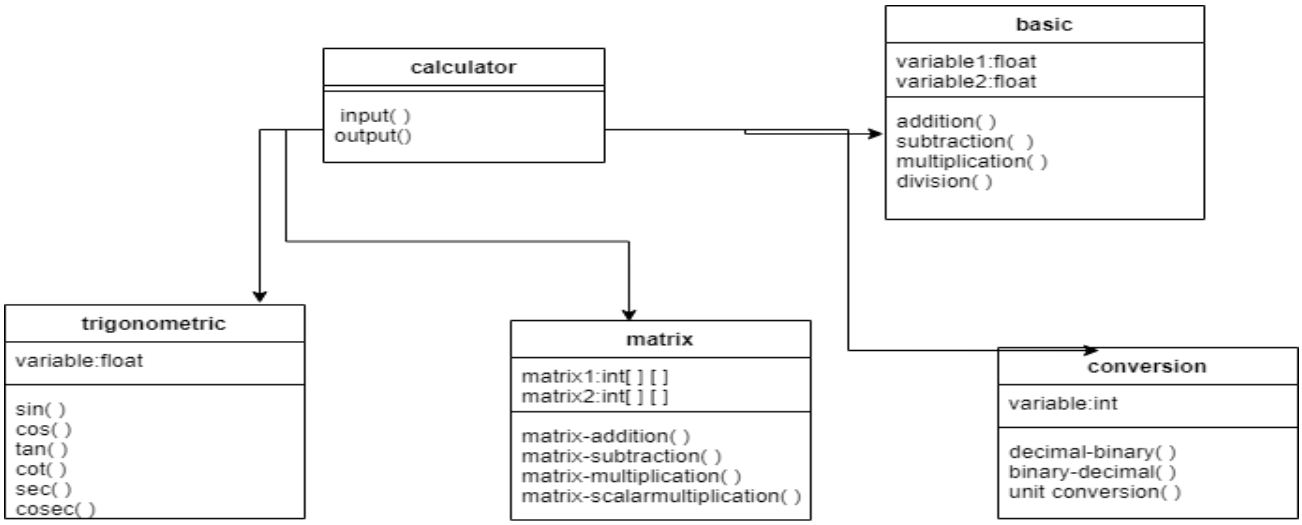
Design

High Level Design

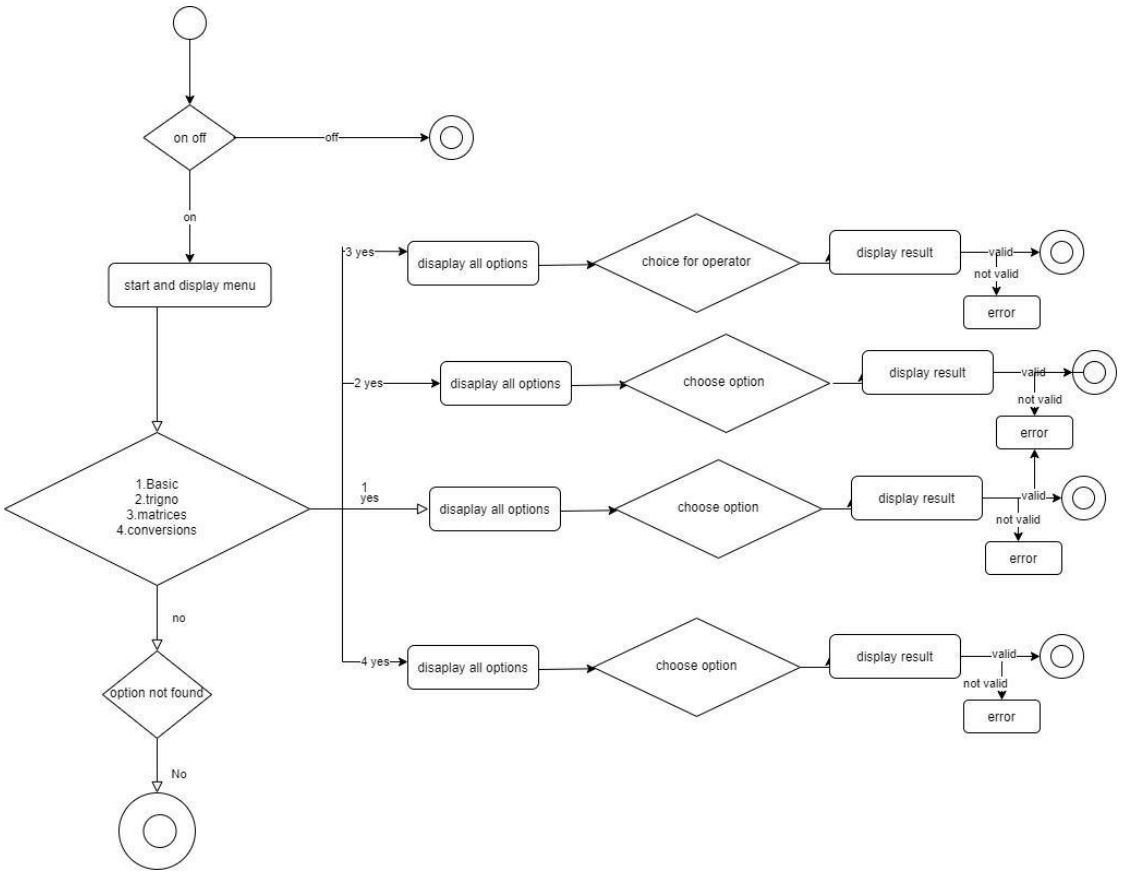
OBJECT DIAGRAM FOR HIGH LEVEL REQUIREMENTS



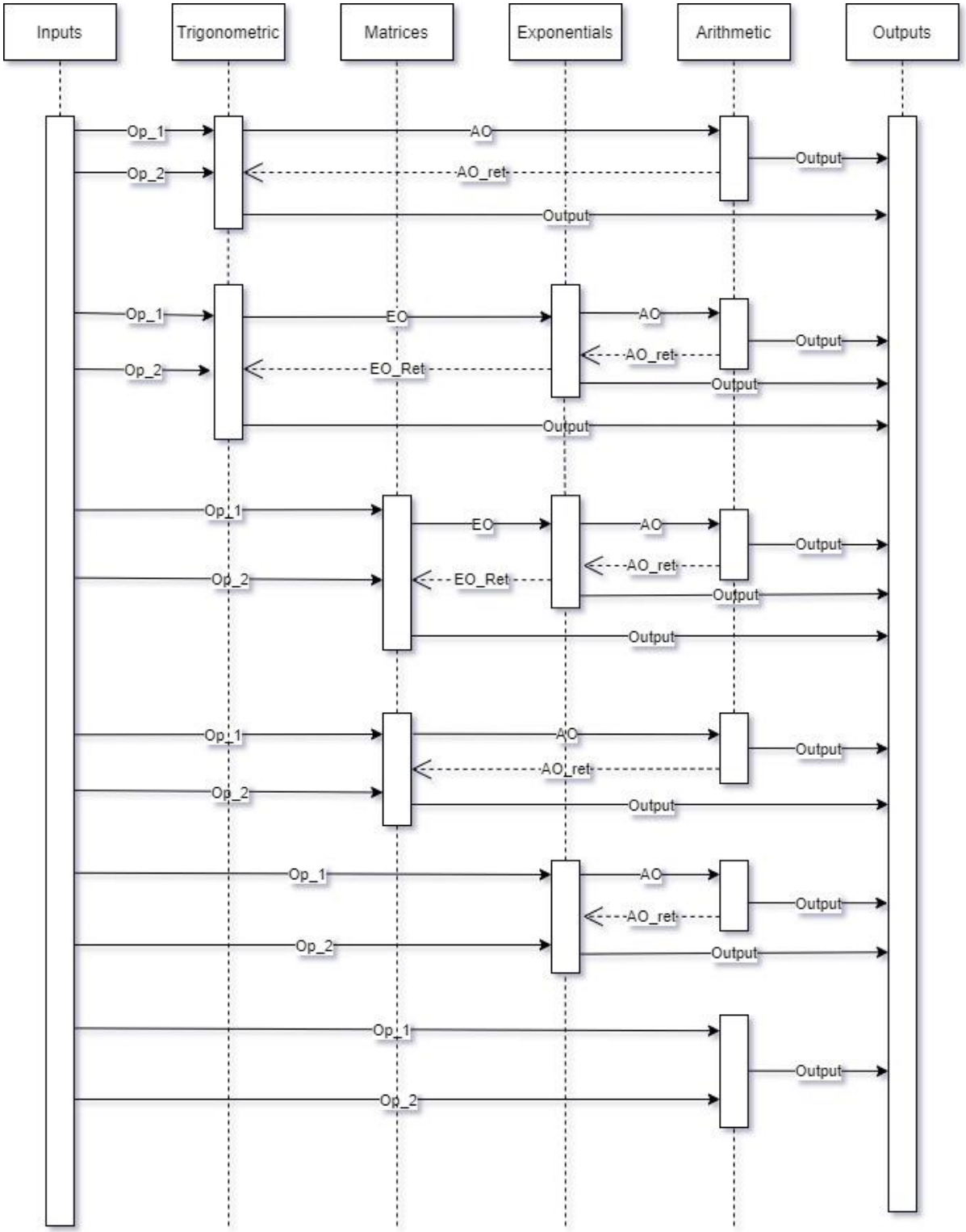
CLASS DIAGRAM FOR HLR



ACTIVITY DIAGRAM for HLR

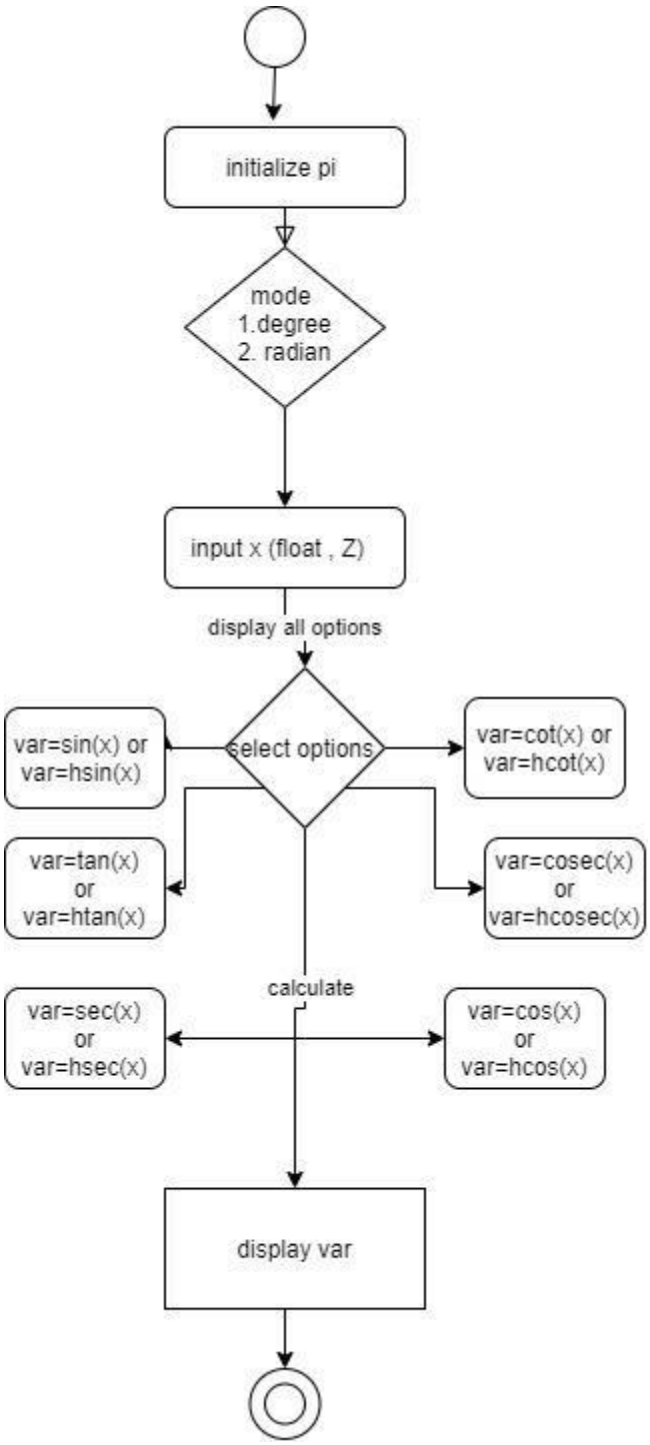


SEQUENCE DIAGRAM FOR HIGH LEVEL REQUIREMENTS

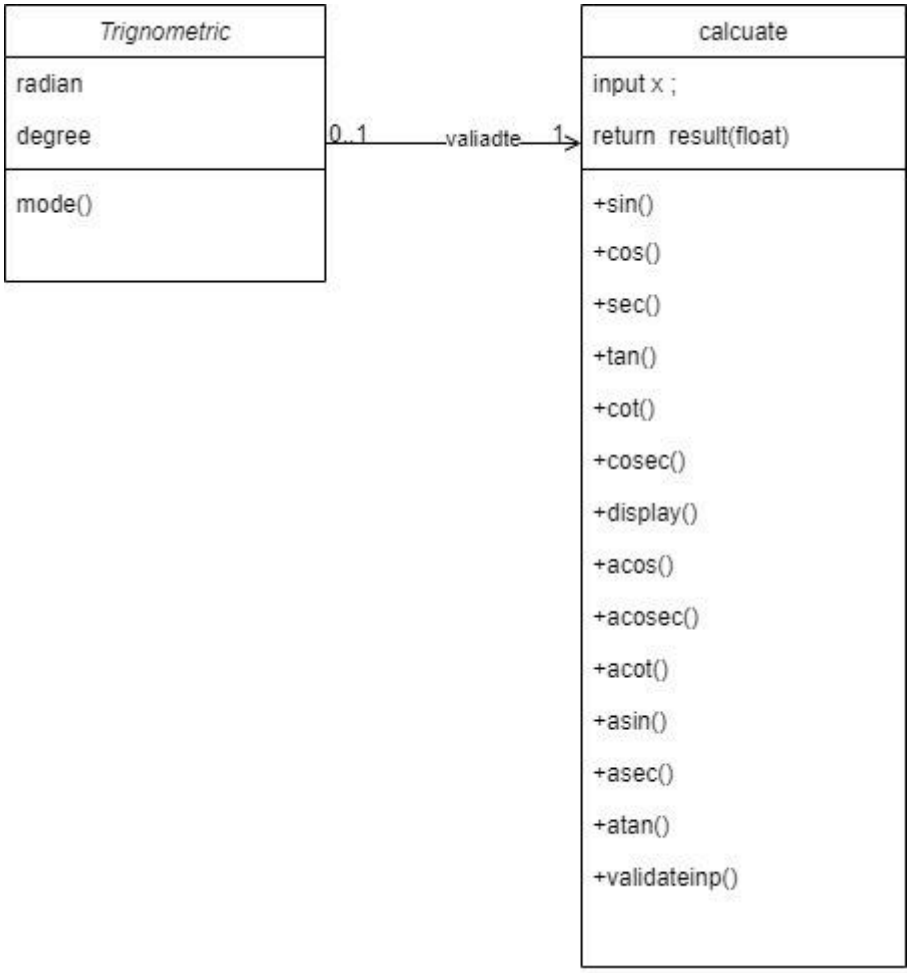


Low Level Design

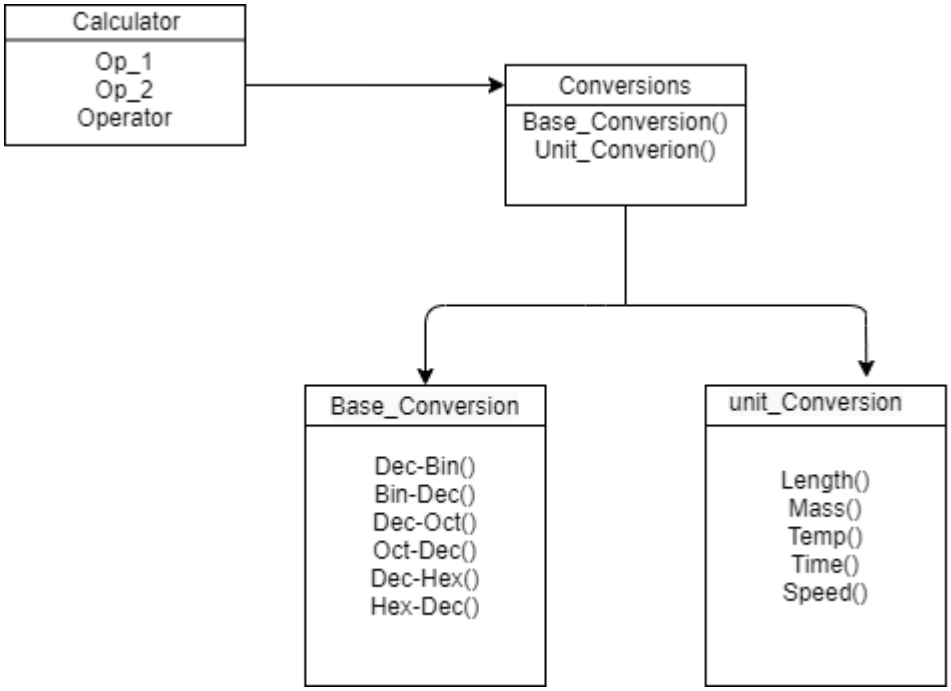
ACTIVITY DIAGRAM for TRIGNOMETRIC



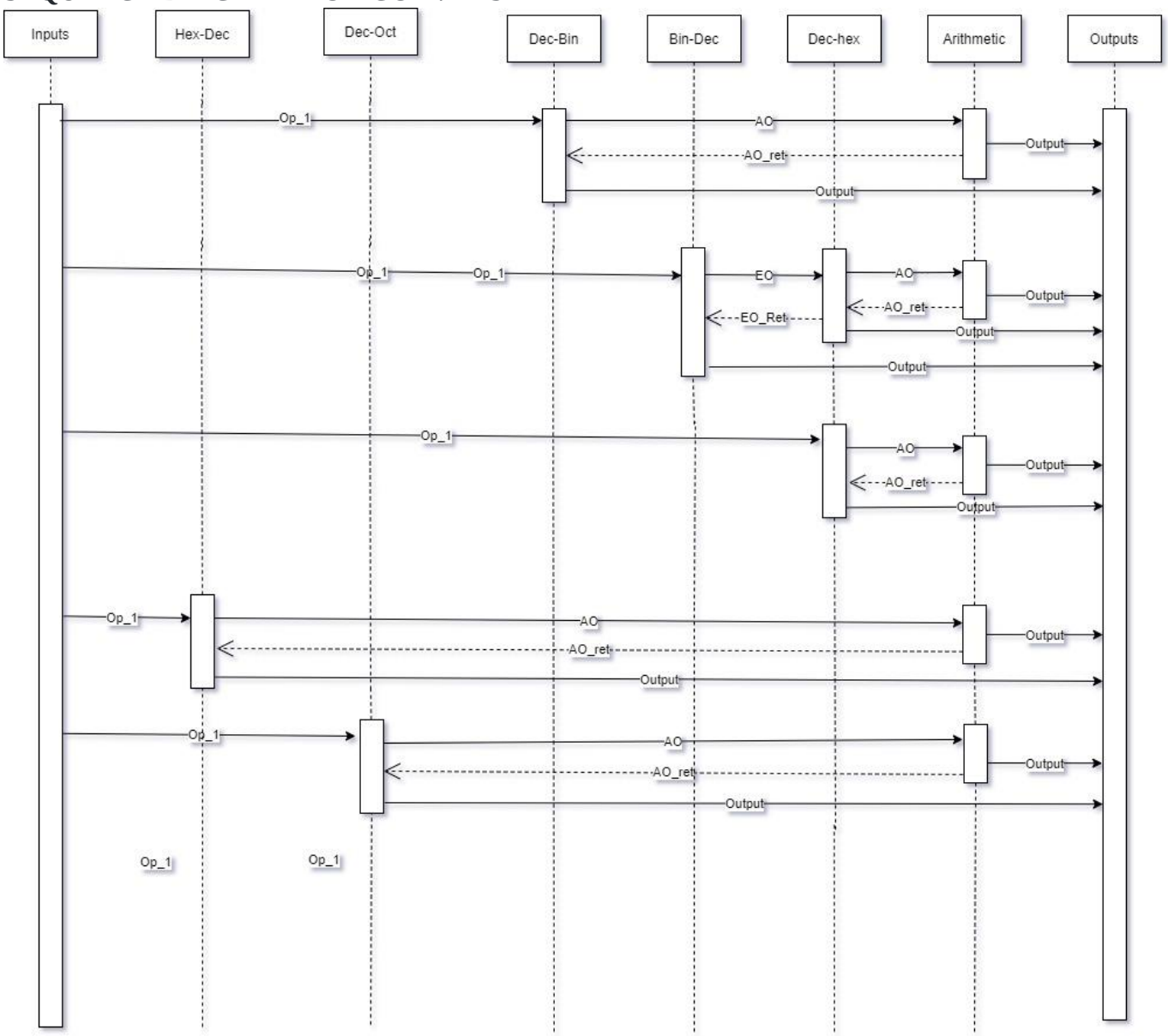
CLASS DIAGRAM FOR TRIGNOMETRIC



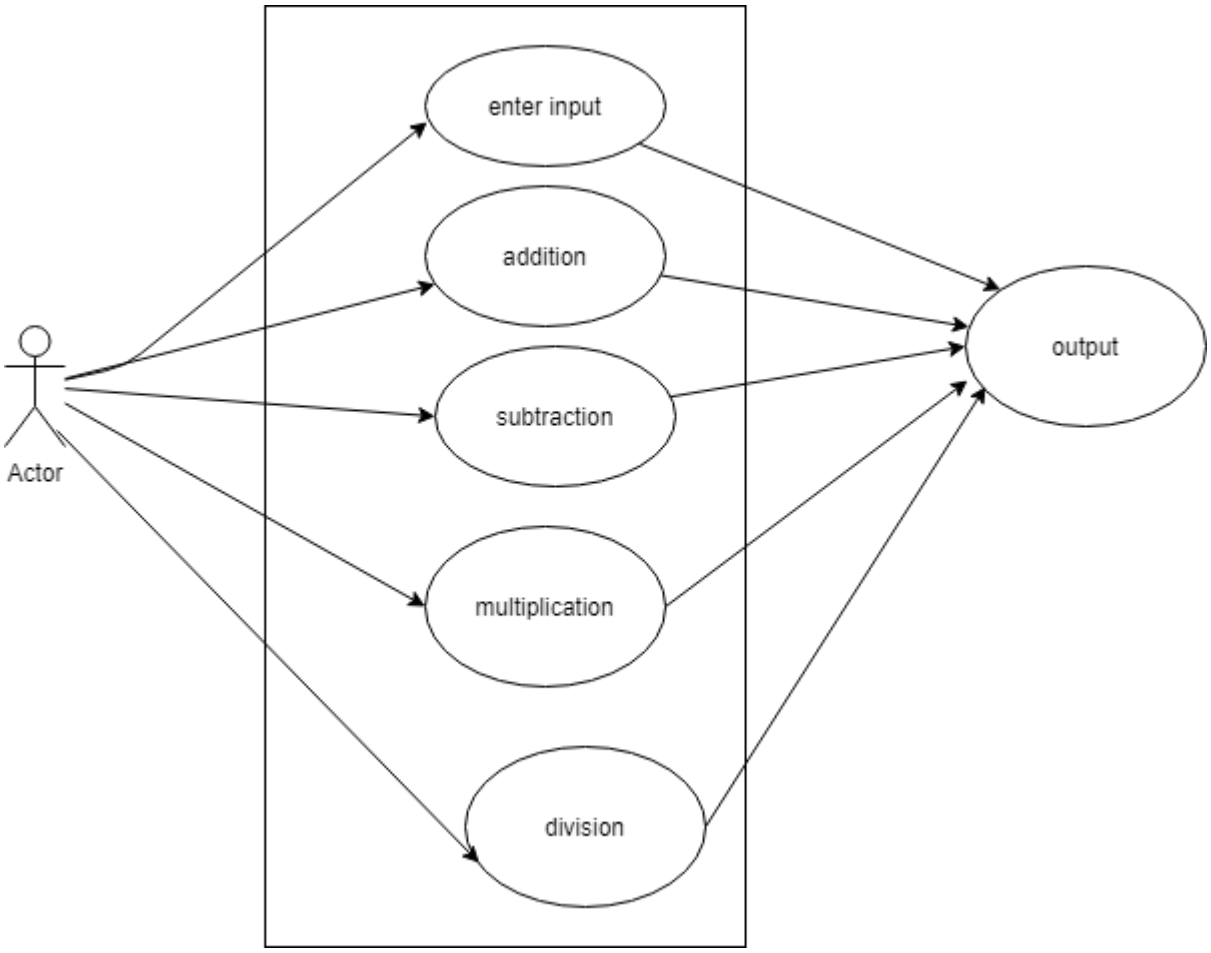
CLASS DIAGRAM FOR CONVERSION



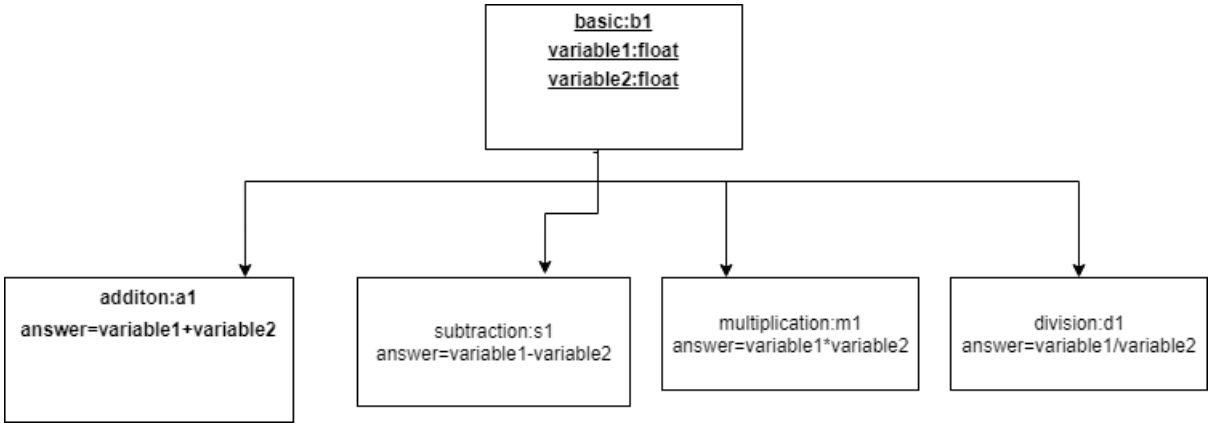
SEQUENCE DIAGRAM FOR CONVERSI



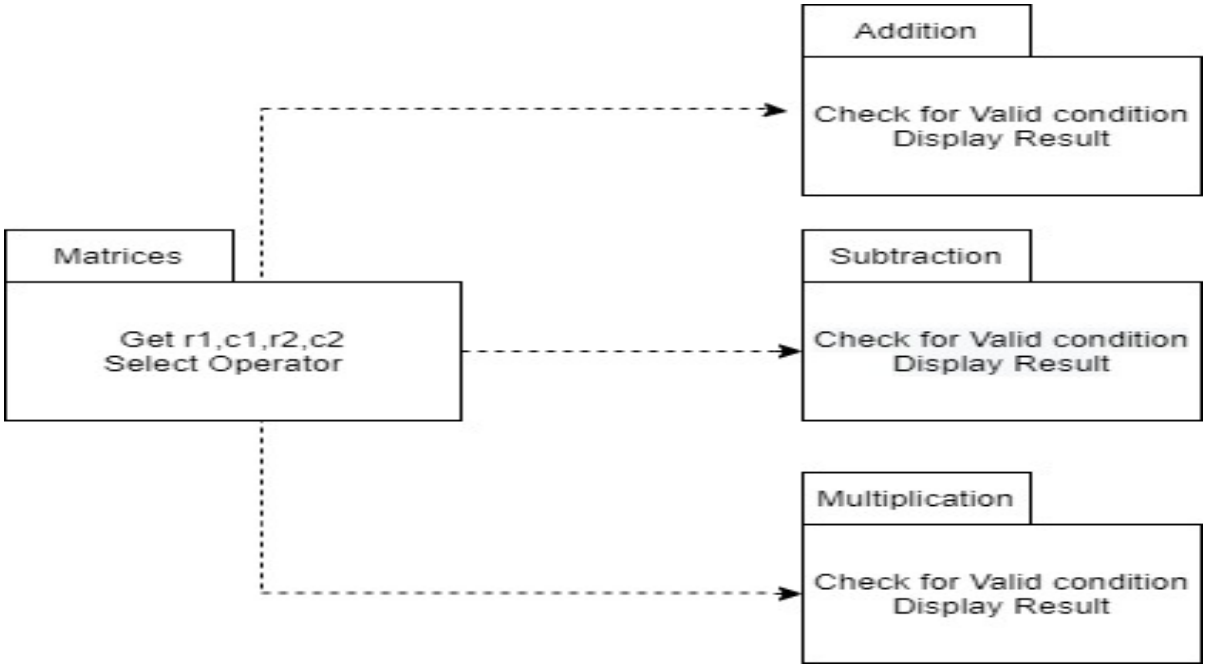
USE CASE DIAGRAM FOR BASIC



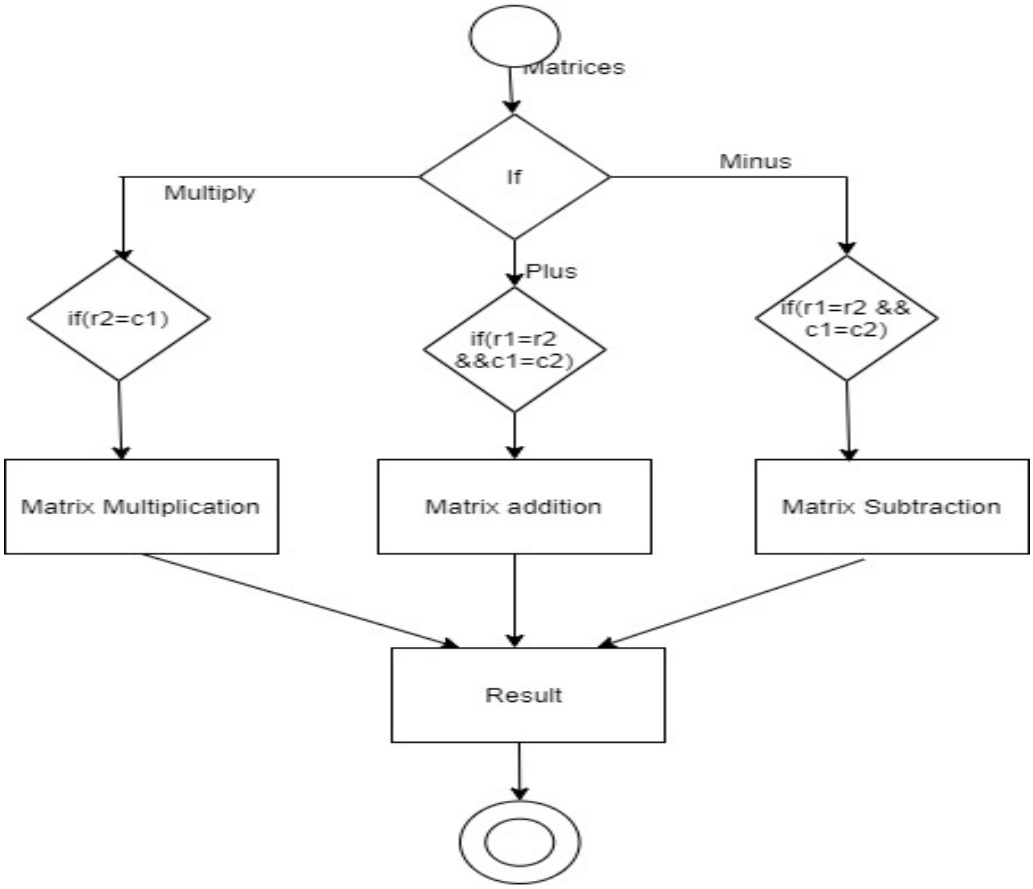
OBJECT DIAGRAM FOR BASIC



PACKAGE DIAGRAM FOR MATRIX OPERATIONS



ACTIVITY DIAGRAM FOR MATRIX OPERATION



Implementation

main.c

```
#include
<trigono.h>
                #include <basic.h>
                #include <conversion.h>


                void setUp(void){}
                void tearDown(void){}

                int main()
                {
                 /*
                float a;
                float sine(float a);
                float cosine(float a);
                */
                }

                /*
                unsigned int calculator_operation = 0;
                int main(int argc, char *argv[])
                {
                   printf("\n****Calculator****\n");
                   while(1)
                   {
                      calculator_menu();
                   }
                }
                void calculator_menu(void)
                {
                   printf("\nAvailable Operations\n");
                   printf("\n1. Basic operation\n2. Matrix operation \n3. Trigonometric
                operations\n4. Conversion \n5. Exit\n");
                   printf("\n\tEnter your choice\n");
                   scanf("%d", &calculator_operation);
                      switch(calculator_operation)
                {
                      case 1:
                        basic_operations();
                         break;
                      case 2:
```

```
                Matrix_operation();
                break;

            case 4:
                conversion_operation();
                break;
            case 5: exit(0);
                    break;

        }
    }
*/
```

TEST PLAN:

Test Plan For Matrix Operation

Test Plan For Matrix Operation

Test ID	Requirement	Expected Output	Actual output
LLR8	R1!=R2 &C1!=C2	Invalid	Invalid
LLR8	R1=R2 &C1=C2	Do Matrix Addition	Do Matrix Addition
LLR9	R1!=R2 &C1!=C2	Invalid	Invalid
LLR9	R1=R2 &C1=C2	Do Matrix Subtraction	Do Matrix Addition
LLR10	R2!=C1	Invalid	Invalid
LLR10	R2=C1	Do Matrix Multiplication	Do Matrix Addition

Test Plan For Trigonomteric Operation

Test ID	Requirement	Expected Output	Type of test case	Actual output
LLR1	sin(30)	-0.988032	Positive Test case	-0.988032
LLR2	cos(30)	0.1542	Positive Test case	0.1542
LLR3	tan(30)	-6.405331	Positive Test case	-6.405331
LLR4	cot(30)	-0.15612	Positive Test case	-0.15612
LLR5	sec(30)	6.4829	Positive Test case	6.4829
LLR6	cosec(30)	-1.012113	Positive Test case	-1.012113
LLR7	sin(60)	-0.304811	Positive Test case	-0.304811
LLR8	cos(60)	-0.9524	Positive Test case	-0.9524
LLR9	tan(60)	0.32004	Positive Test case	0.32004
LLR10	cot(60)	3.124606	Positive Test case	3.124606
LLR11	sec(60)	-1.04996	Positive Test case	-1.04996
LLR12	cosec(60)	-3.280726	Positive Test case	-3.280726

For Conversion Operation

Test plans

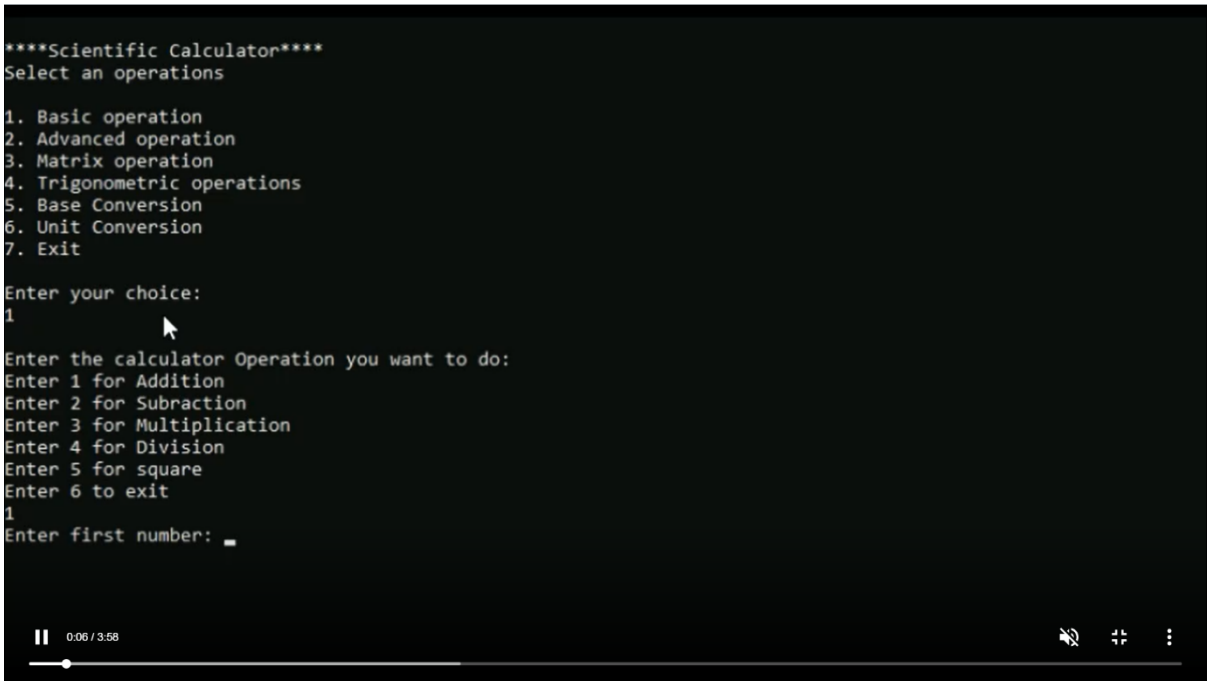
Conversions:

Test ID	Description	Expected o/p	Actual o/p
1	Check input valid	Yes/No	
2	Check choice	Do selected choice	Select Operaton
3	Check for binary; Check_bin (10101) Check_bin (21)	 Valid Invalid Bin	 Valid Invalid Bin
4	Check for Octal; Check_Octal (10101) Check_octal (83)	 Valid Invalid	 Valid Invalid

For Basic Operation

TEST CASE ID	DESCRIPTION	EXPECTED INPUTPUT	EXPECTED OUTPUT	ACTUAL OUTPUT
LLR01	Check the number dived by zero	Given number is divided by zero	Error	Error
LLR02	Check the base of log	Given number is other than 2,10	Error	Error
LLR03	Check the number in square root	Given number is negative not positive	Error	Error
LLR04	Check the number in cube root	Given number is negative	Negative number	Negative number
LLR05	Check divided by big number	Given number is greater	Less than zero	Less than zero
LLR06	Check divided by smaller number	Given number is smaller	Greater than zero	Greater than zero
LLR07	Check divided by negative number	Given number is negative	Negative number	Negative number

6_Images_And_Videos



```
Enter 6 to exit
1
Enter first number: 23
Enter second number: 12
Answer: 35

Enter the calculator Operation you want to do:
Enter 1 for Addition
Enter 2 for Subraction
Enter 3 for Multiplication
Enter 4 for Division
Enter 5 for square
Enter 6 to exit
2

Enter first number  : 10

Enter second number : 5

Answer: 5

Enter the calculator Operation you want to do:
Enter 1 for Addition
Enter 2 for Subraction
Enter 3 for Multiplication
Enter 4 for Division
Enter 5 for square
Enter 6 to exit
```

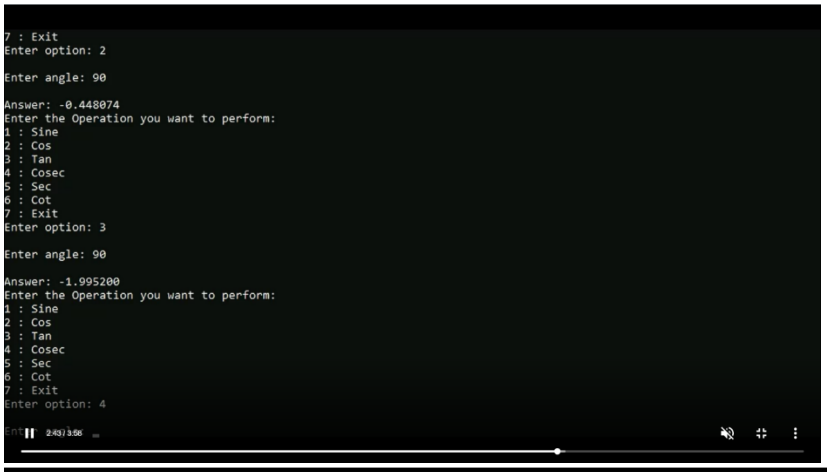
```
Enter 5 to AC to DC
Enter 6 to DC to AC
Enter 7 to exit
Enter option: 3

Enter data: 100

Answer: 328.083984
Enter the Operation you want to perform:
Enter 1 for  Celsius to Fahrenheit
Enter 2 for Farenheit to Celcius
Enter 3 for Meter to Feet
Enter 4 for Feet to Meter
Enter 5 to AC to DC
Enter 6 to DC to AC
Enter 7 to exit
Enter option: 4

Enter data: 50

Answer: 15.240000
Enter the Operation you want to perform:
Enter 1 for  Celsius to Fahrenheit
Enter 2 for Farenheit to Celcius
Enter 3 for Meter to Feet
Enter 4 for Feet to Meter
Enter 5 to AC to DC
Enter 6 to DC to AC
Enter 7 to exit
Enter option:
```

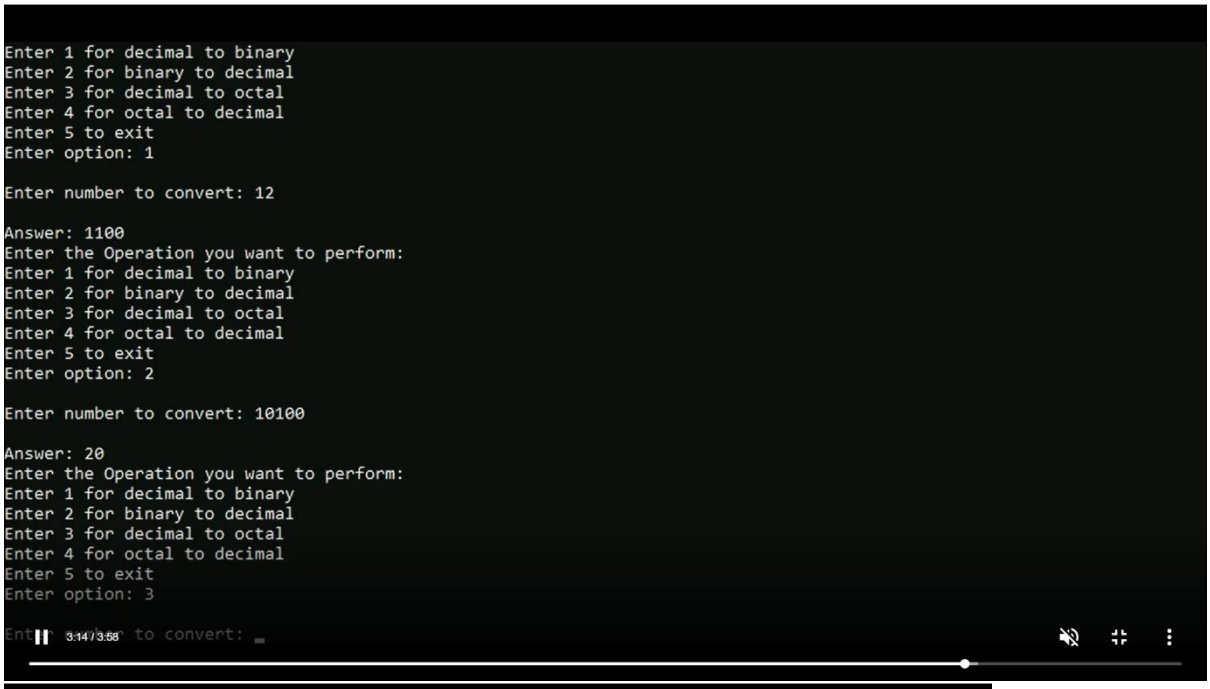
3:42 / 3:58

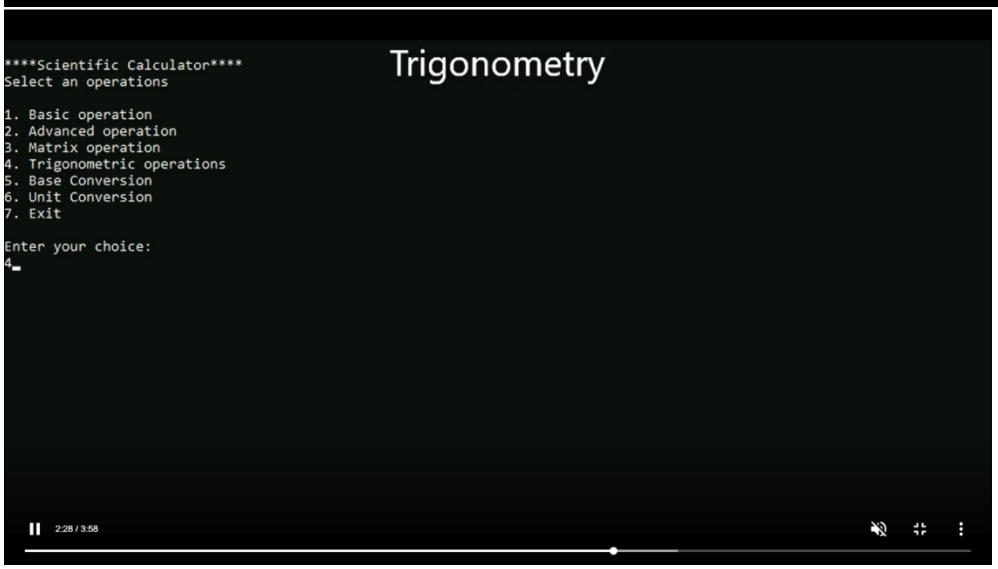
Unit conversion

```
****Scientific Calculator****
Select an operations

1. Basic operation
2. Advanced operation
3. Matrix operation
4. Trigonometric operations
5. Base Conversion
6. Unit Conversion
7. Exit

Enter your choice:
6

Enter the Operation you want to perform:
Enter 1 for  Celsius to Fahrenheit
Enter 2 for Farenheit to Celcius
Enter 3 for Meter to Feet
Enter 4 for Feet to Meter
Enter 5 to AC to DC
Enter 6 to DC to AC
Enter 7 to exit
Enter option:
```

3:27 / 3:58


7 : Exit
Enter option: 2
Enter angle: 90
Answer: -0.448074
Enter the Operation you want to perform:
1 : Sine
2 : Cos
3 : Tan
4 : Cosec
5 : Sec
6 : Cot
7 : Exit
Enter option: 3
Enter angle: 90
Answer: -1.995200
Enter the Operation you want to perform:
1 : Sine
2 : Cos
3 : Tan
4 : Cosec
5 : Sec
6 : Cot
7 : Exit
Enter option: 4


Enter 1 for decimal to binary
Enter 2 for binary to decimal
Enter 3 for decimal to octal
Enter 4 for octal to decimal
Enter 5 to exit
Enter option: 1
Enter number to convert: 12
Answer: 1100
Enter the Operation you want to perform:
Enter 1 for decimal to binary
Enter 2 for binary to decimal
Enter 3 for decimal to octal
Enter 4 for octal to decimal
Enter 5 to exit
Enter option: 2
Enter number to convert: 10100
Answer: 20
Enter the Operation you want to perform:
Enter 1 for decimal to binary
Enter 2 for binary to decimal
Enter 3 for decimal to octal
Enter 4 for octal to decimal
Enter 5 to exit
Enter option: 3
3:14 / 3:58


Trigonometry
****Scientific Calculator****
Select an operations
1. Basic operation
2. Advanced operation
3. Matrix operation
4. Trigonometric operations
5. Base Conversion
6. Unit Conversion
7. Exit
Enter your choice:
4
2:28 / 3:58

```
3 : Multiplication
4 : Exit
Enter option: 1
Enter the number of rows (between 1 and 5): 2
Enter the number of columns (between 1 and 5): 2

Enter elements of 1st matrix:
Enter element a11: 2
Enter element a12: 3
Enter element a21: 4
Enter element a22: 5
Enter elements of 2nd matrix:
Enter element b11: 6
Enter element b12: 7
Enter element b21: 8
Enter element b22: 9

Sum of two matrices:
8   10

12   14


Enter the Operation you want to perform:
1 : Addition
2 : Subraction
3 : Multiplication
4 : Exit
Enter option: 2
Enter the number of rows (between 1 and 5):
```

Matrix

```
****Scientific Calculator****
Select an operations

1. Basic operation
2. Advanced operation
3. Matrix operation
4. Trigonometric operations
5. Base Conversion
6. Unit Conversion
7. Exit

Enter your choice:
3

Enter the Operation you want to perform:
1 : Addition
2 : Subraction
3 : Multiplication
4 : Exit
Enter option: _
```

1:39 / 3:58

```
Enter the Operation you want to perform:
1 : Log
2 : Exponential
3 : Modulus
4 : Factorial
5 : Square root
6 : cube root
7 : GCD
8 : LCM
9 : Combination
10 : Permutation
11 : Exit
Enter option: 6

Enter number: 4

Answer: 1.587401
Enter the Operation you want to perform:
1 : Log
2 : Exponential
3 : Modulus
4 : Factorial
5 : Square root
6 : cube root
7 : GCD
8 : LCM
9 : Combination
10 : Permutation
11 : Exit
Enter option: 4_
```

```
Enter the Operation you want to perform:
1 : Log
2 : Exponential
3 : Modulus
4 : Factorial
5 : Square root
6 : cube root
7 : GCD
8 : LCM
9 : Combination
10 : Permutation
11 : Exit
Enter option: 2

Enter number : 4

Answer: 54.598148
Enter the Operation you want to perform:
1 : Log
2 : Exponential
3 : Modulus
4 : Factorial
5 : Square root
6 : cube root
7 : GCD
8 : LCM
9 : Combination
10 : Permutation
11 : Exit
Enter option:
```

```
Enter the Operation you want to perform:
1 : Log
2 : Exponential
3 : Modulus
4 : Factorial
5 : Square root
6 : cube root
7 : GCD
8 : LCM
9 : Combination
10 : Permutation
11 : Exit
Enter option: 1

Enter number: 5

Answer: 1.609438
Enter the Operation you want to perform:
1 : Log
2 : Exponential
3 : Modulus
4 : Factorial
5 : Square root
6 : cube root
7 : GCD
8 : LCM
9 : Combination
10 : Permutation
11 : Exit
Enter option: 2
```

```
****Scientific Calculator****
Select an operations

1. Basic operation
2. Advanced operation
3. Matrix operation
4. Trigonometric operations
5. Base Conversion
6. Unit Conversion
7. Exit

Enter your choice:
2

Enter the Operation you want to perform:
1 : Log
2 : Exponential
3 : Modulus
4 : Factorial
5 : Square root
6 : cube root
7 : GCD
8 : LCM
9 : Combination
10 : Permutation
11 : Exit
Enter option: 1

Enter number: 5
```

```
****Scientific Calculator****
Select an operations

1. Basic operation
2. Advanced operation
3. Matrix operation
4. Trigonometric operations
5. Base Conversion
6. Unit Conversion
7. Exit

Enter your choice:
2
```

Advanced operations

```
7. Exit

Enter your choice:
1

Enter the calculator Operation you want to do:
Enter 1 for Addition
Enter 2 for Subraction
Enter 3 for Multiplication
Enter 4 for Division
Enter 5 for square
Enter 6 to exit
1
Enter first number: 23
Enter second number: 12
Answer: 35

Enter the calculator Operation you want to do:
Enter 1 for Addition
Enter 2 for Subraction
Enter 3 for Multiplication
Enter 4 for Division
Enter 5 for square
Enter 6 to exit
2

Enter first number  : 10

Enter second number :
```

```
Enter first number  : 10

Enter second number : 5

Answer: 5

Enter the calculator Operation you want to do:
Enter 1 for Addition
Enter 2 for Subraction
Enter 3 for Multiplication
Enter 4 for Division
Enter 5 for square
Enter 6 to exit
3

Enter first number: 5

Enter second number: 4

Answer: 20

Enter the calculator Operation you want to do:
Enter 1 for Addition
Enter 2 for Subraction
Enter 3 for Multiplication
Enter 4 for Division
Enter 5 for square
Enter 6 to exit
```

SDLC Activity Based Learning

Code Quality	[Git Inspector]	linux	windows
Code Quality - Static Code - Cppcheck passing	Contribution Check - Git Inspector passing	Linux C/C++ CI passing	Windows C/C++ CI passing

Challenges Faced and How Was It Overcome

1. Running the make file as its resolved by defining its correct path(.out for linux and -lm for math functions)
2. Syncronizing the VS code to github , colleague help to resolve the issue
3. Making the function call in correct path
4. Open git log while committing, thus went to github desktop and pulled origin and then pushed origin.
5. Test case code for the boundary problem. Added code with the help of internet

Learning Resources

1. markdownCheatsheet
2. markdownBasics
3. git inspector
4. github workflow

Git-hub link: https://github.com/GENESIS2021Q1/Applied_SDLC-Dec_Team_48

Mini-project 4 – Calender-Automation [Team]

Modules

1. Python
2. Git

Requirements

High Level Requirements

ID	Feature	MATLAB v0 Status	Python v0 Status
HR01	GUI	Implemented	Implemented
HR02	Master Calender	Implemented	Implemented
HR03	Faculty calender	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 (Color 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 monthwise 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 geacalendar"
    - 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"

- o 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

Mini Project 5 – Sunroof System [Team]

Sunroof System

Introduction

A sunroof is a movable panel that opens to uncover a window in an automobile roof, allowing light and fresh air to enter the passenger compartment. Sunroofs can be manually operated or motor driven, and are available in many shapes, sizes and styles. While the term sunroof is now used generically to describe any glass panel in the roof, the term "moonroof" was historically used to describe stationary glass panes rigidly mounted in the roof panel over the passenger compartment.

A moonroof has a glass panel that is transparent and usually tinted. Previous terms include Sunshine Roof, Sliding Head and Sliding Roof.

Overview

• It gives the car a stylish interior and exterior, making it look trendy.

• They provide the user with an extra window that enables them to breathe in the more fresh air.

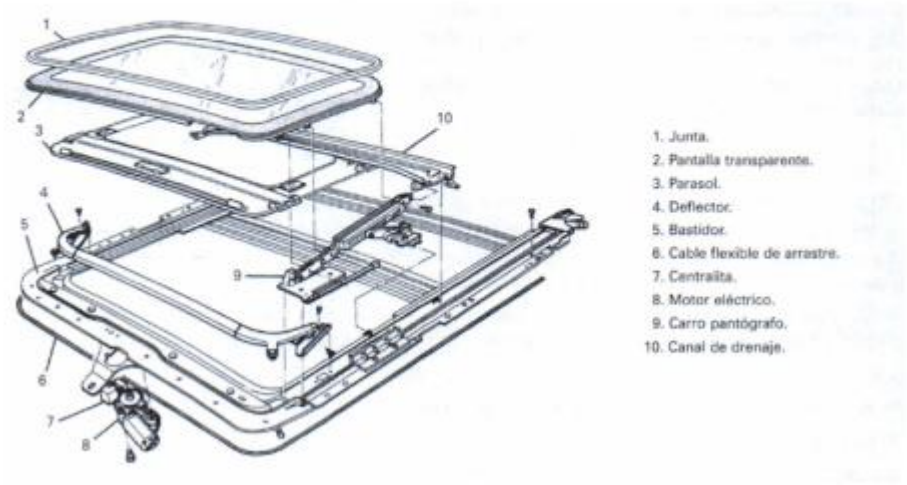
• In traffic, the car sunroofs come in handy as they fairly reduce the noises of vehicles and let the fresh air come in the passenger's cabinet.

• It gives a beautiful view during the rains as the raindrops fall on the glass of the sunroofs. • In cold climates, these sunroofs enable the sun to throw in bright rays and make the driving experience better.

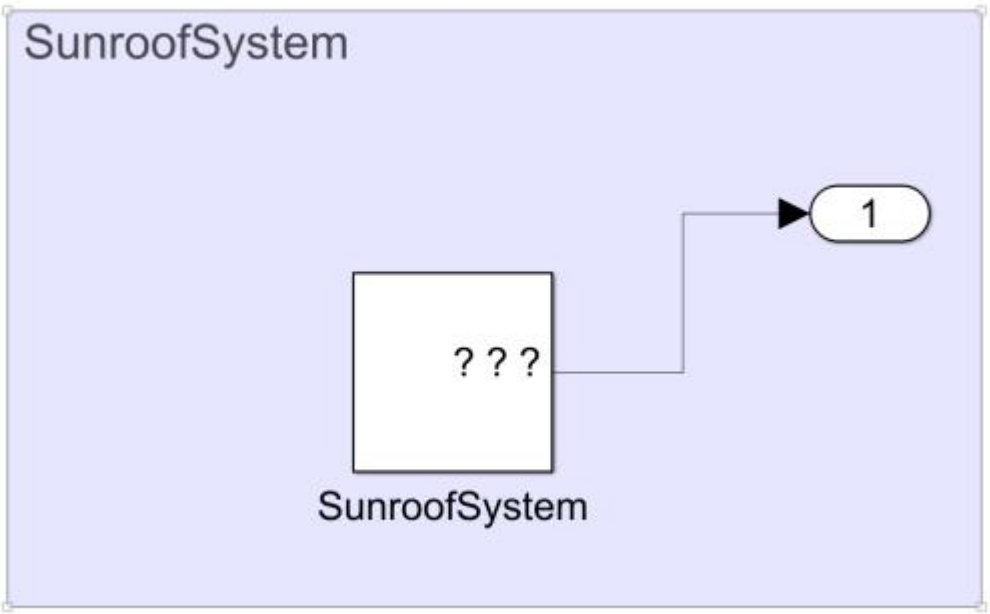
• A Sunroof can operate in dual ways, firstly, the whole roof and the tilt function. With either of the two, one can drastically reduce the car compartment's warmth. This helps with all the warm days and humidity and is a good feature, especially in India

DESIGN

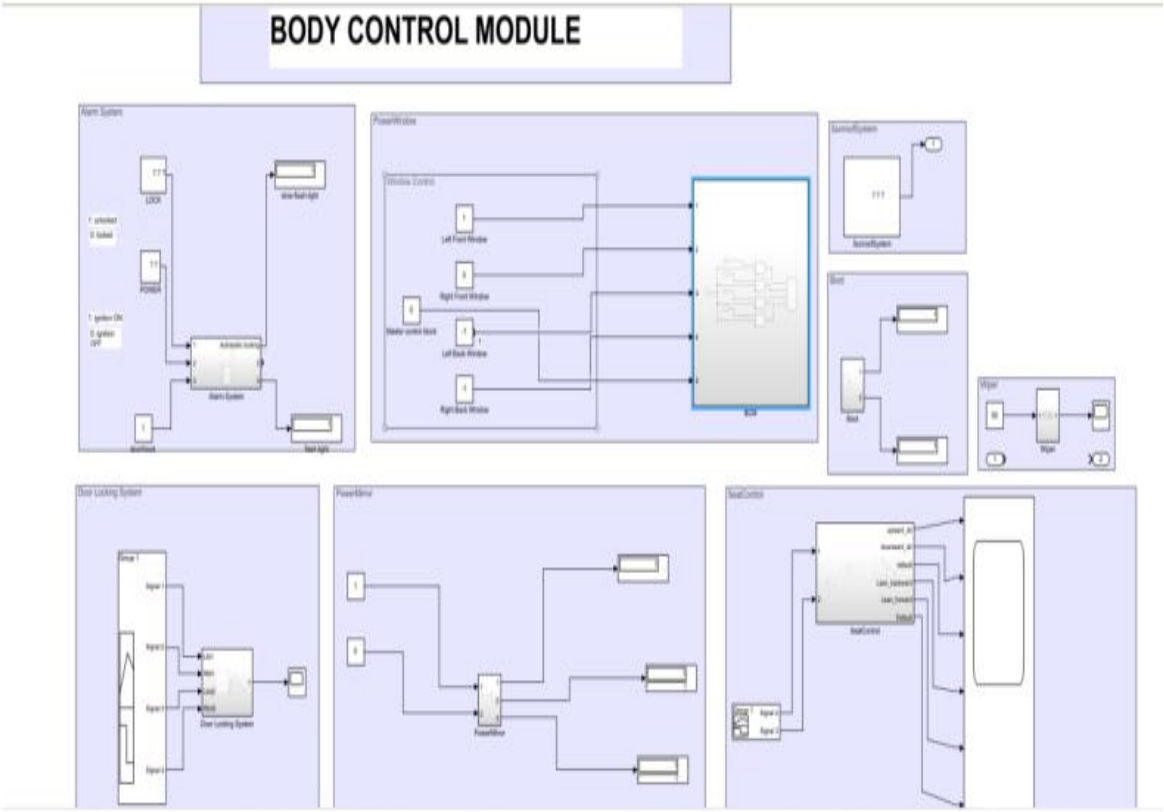
DESIGN



MODELING



Team Activity



Mini-project 6 – Wiper Control [Team]

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
HR 01	User can be able to turn ON/OFF the Wiper control system	Implemented
HR 02	Ignition Key Position at ACC: The Red LED is ON, if the user button is pressed and held for 2 secs	Implemented
HR 03	Wiper ON: Wiper is OFF: On press of the user input, Blue, Green and Orange LEDs come ON one at a time with the set frequency	Implemented
HR 04	Wiper OFF: Wiper is ON: The LED glow pattern stops on the 4th press; the wiper action starts next press onwards as mentioned in step 2	Implemented
HR 05	Ignition Key Position at Lock: The Red LED is OFF, if the user button is pressed and held for 2 secs	Implemented

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

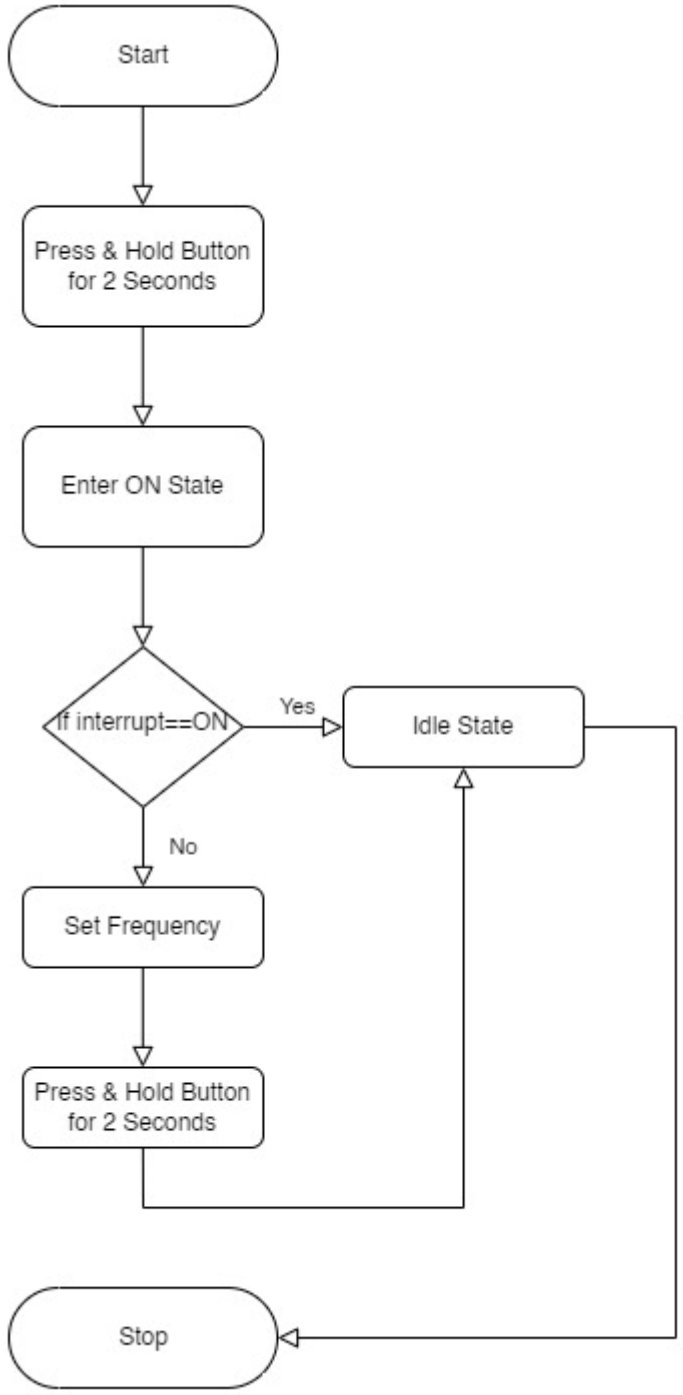


Figure 3 Structure Diagram

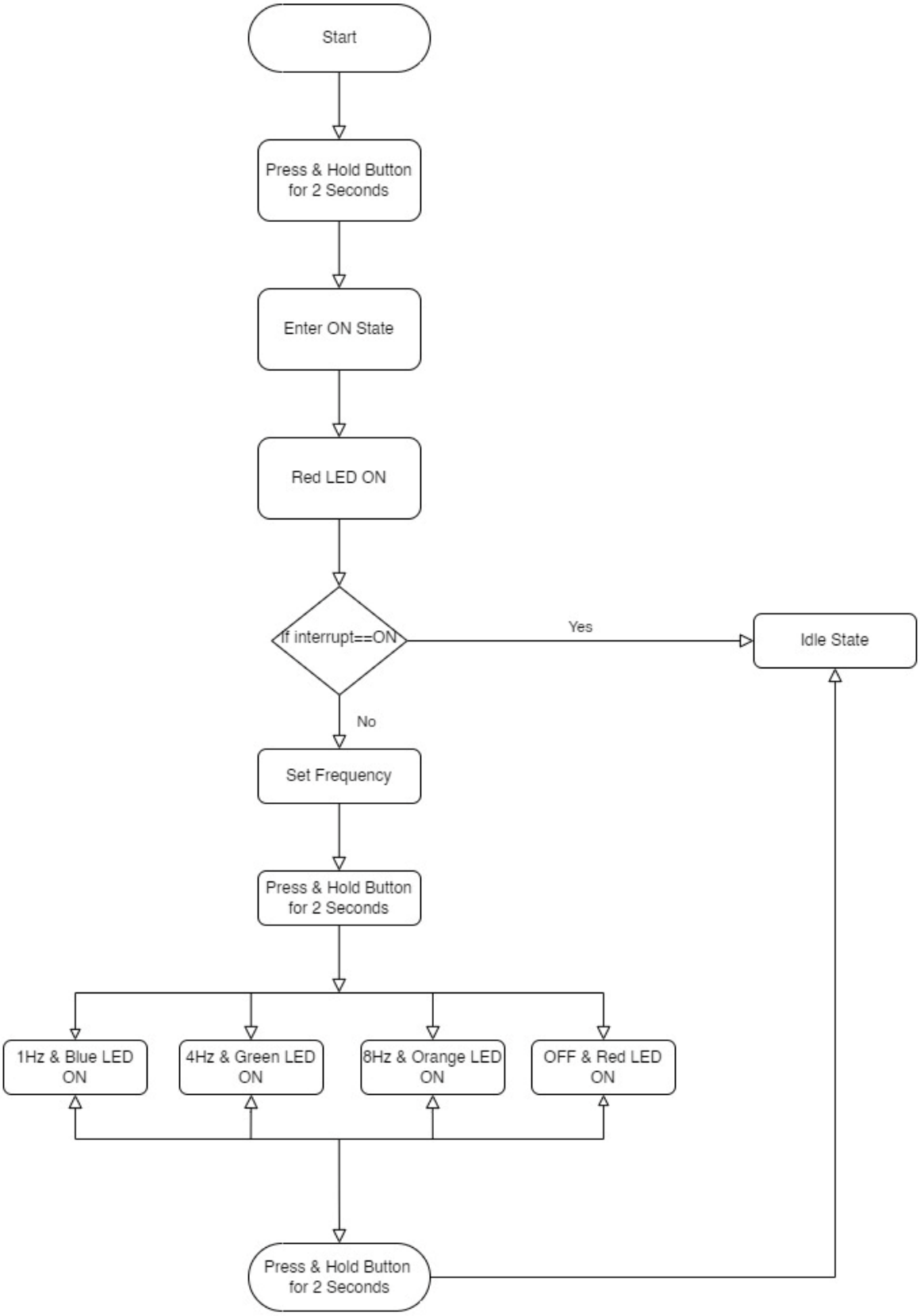


Figure 4 Behavior Diagram

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-Team46

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

Mini-project 7 – Golf-Cart [Team]

GOLF-KART

HISTORY

Reportedly, the first use of a motorized cart on a golf course was by JK Wadley of Texarkana, Texas/Arkansas, who saw a three-wheeled electric cart being used in Los Angeles to transport senior citizens to the grocery store. Later, he purchased a cart and found that it worked poorly on a golf course. The first electric golf cart was custom-made in 1932, but did not gain widespread acceptance. In the 1930s until the 1950s the most widespread use of golf carts was for those with disabilities who could not walk far. By the mid 1950s the golf cart had gained wide acceptance with US golfers, with several manufactures (e.g. Victor Adding Machine Co. and Sears Roebuck) producing various models. Most were electric.

ABSTRACT

In this paper, the Design & Calculations of an Electric golf cart is done, the calculations and process to develop a golf cart chassis, electrical transmission using transaxle, braking, steering geometry, suspension system, solar panel of a long six seat golf cart, battery calculations and controller which monitors the electric system is explained. All the materials used for the calculation are industry standard being utilized currently. Priority of this design was to keep the cart as light weight as possible and rigid with good performance on grassy and concrete terrain's used in either golf course or in hospitality education sectors. A unique tubular chassis is used in the design comprising of circular & rectangular shape. The focus of design was long mileage from the batteries used and how much, adding solar panels can contribute to the existing range of the electric vehicle. A 23 miles range from the battery pack using 6 deep cycle lead acid batteries was achieved and increased12 miles from the solar panels under the constraint assumed for its use (direct sunlight for over 6-7 hours)

Key words:

electric golf cart, design, Electric vehicle, electric transmission, vehicle design, battery calculation, solar golf cart.

INTRODUCTION

This projects aim is to Design an electric golf cart with six people seating capacity which runs on batteries as well as solar panels. The need for a golf cart isn't just using it in the golf course and is nowadays being used in large universities spread over vast acres of land and in expensive hotels and latest complexes for comfort traveling i.e Hospitality industry and providing low cost of charging with increasing mileage as technology advances. In this project the aim has been to put forth an industry standard design for an electric golf cart that can run exceedingly well on grassy / muddy terrains as well as wet conditions and concrete roads. During designing all the

constraints and cost considerations and all technical aspects were set according to the current advancements in the industry in a thoroughly researched manner. A battery electric vehicle (BEV) has far fewer moving parts than a conventional gasoline-powered vehicle. There's no need for a liquid fuels or oil changes. There's no transmission or timing belt to fail you. And most of the maintenance cost associated with ic engine are eliminated.

REQUIREMENTS

Operator Regulations

All operators of golf carts or NEVs shall:

- Be licensed to operate a motor vehicle as provided by Texas Transportation Code, Section 521.021 and carry a valid driver's license and all state law driver's license permissions and restrictions shall apply to the operation of a golf cart or NEV
- Abide by all state and local traffic regulations applicable to vehicular traffic
- Use standard hand signals for turning during daylight if the operator's golf cart is not equipped with turn signals
- Not operate or park on a sidewalk or hike and bike trail at any time
- Not use the vehicle to pull any object or person at any time
- Not exceed the seating capacity of the vehicle as designed by the manufacturer
- Remain seated at all times while the vehicle is in motion and ensure the same for passengers. Passengers can be issued a citation for not remaining seated while vehicle is moving
- Not have or permit a passenger younger than four years of age
- Maintain financial responsibility as defined in the Texas Transportation Code, Section 601.051
- Not intentionally or knowingly allow an unlicensed operator to operate the vehicle

Penalties

In addition to traffic violations for which the owner or driver of the golf cart or NEV may be subject to pursuant to state law, any person who violates this ordinance shall be guilty of a misdemeanor punishable by a fine as follows:

- First offense shall have a minimum fine of $100 and a maximum fine of $500
- Second offense shall have a minimum fine of $200 and a maximum fine of $500
- Third and subsequent offenses shall have a minimum fine of $500
- Each day's violations shall constitute a separate and distinct offense. Any prior conviction under this ordinance will count towards a second and third charge regardless of when it occurred.

Definitions

A **Golf Cart** is a motor vehicle designed by the manufacturer primarily for transporting persons on a golf course. It must have a minimum of four wheels and has an attainable top speed not greater than 25 miles per hour on a paved, level surface and is in compliance with federal motor vehicle safety standards for low-speed vehicles.

Neighborhood Electric Vehicle (NEV) means a vehicle that can attain a maximum speed of 35 miles per hour on a paved, level surface.

Park or parking means the standing or stopping of a vehicle, whether occupied or not, other than temporarily for the purpose of and while actually engaged in loading or unloading merchandise or passengers.

Parking area means those areas accessible to the public by motor vehicular traffic and which are designated for temporary parking of motor vehicles, usually in places referred to as parking lots.

Required Equipment and its installation and maintenance must meet standards provided by the Texas Transportation Code.

City of New Braunfels

Ordinance Number 2015-50 Addressing

Golf Carts and Neighborhood Electric Vehicles (NEVs)

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**

Golf carts use a series of batteries that supply the needed voltage and amperage, so size and power requirements are key features for a replacement. Be aware that golf carts do not use a single battery, but multiples of 6V, 8V, or 12V batteries. Ensure any replacement batteries you purchase use the same combination. Do not upgrade your batteries over the manufacturer's recommendations, which can ruin your cart. Match the brand standard to the standard on your replacement battery packs. The wrong battery can waste money, a hazardous situation, and any combination in between.

ADAPTIVE GOLF KARTS

New technology such as the Solo-Rider, an adaptive golf cart designed for a single user, is allowing disabled persons access to the golf course and the game itself. The cart's seat swivels around, extends to an upright position, and allows the golfer to stand upright, be supported, and swing using both hands.

A common electric golf cart.

An off-road gasoline-powered golf cart.

Customised golf carts

Golf carts may be modified from their original configuration to perform similar to the growing popularity of the Side by Side (UTV). Modifications as minor as suspension upgrades are commonplace, while entire redesigns may include axles and an engine from a full-size

automobile. Some custom golf carts are inspired by the skateboard, with all four wheels being driven and the golfer controlling the cart in a standing upright position as if riding skateboard, leaning left or right to make turns.

Helios Chargers, 130 watts and 205 watts, on solar powered golf carts.

Solar-powered golf carts

Extreme off-road golf cart

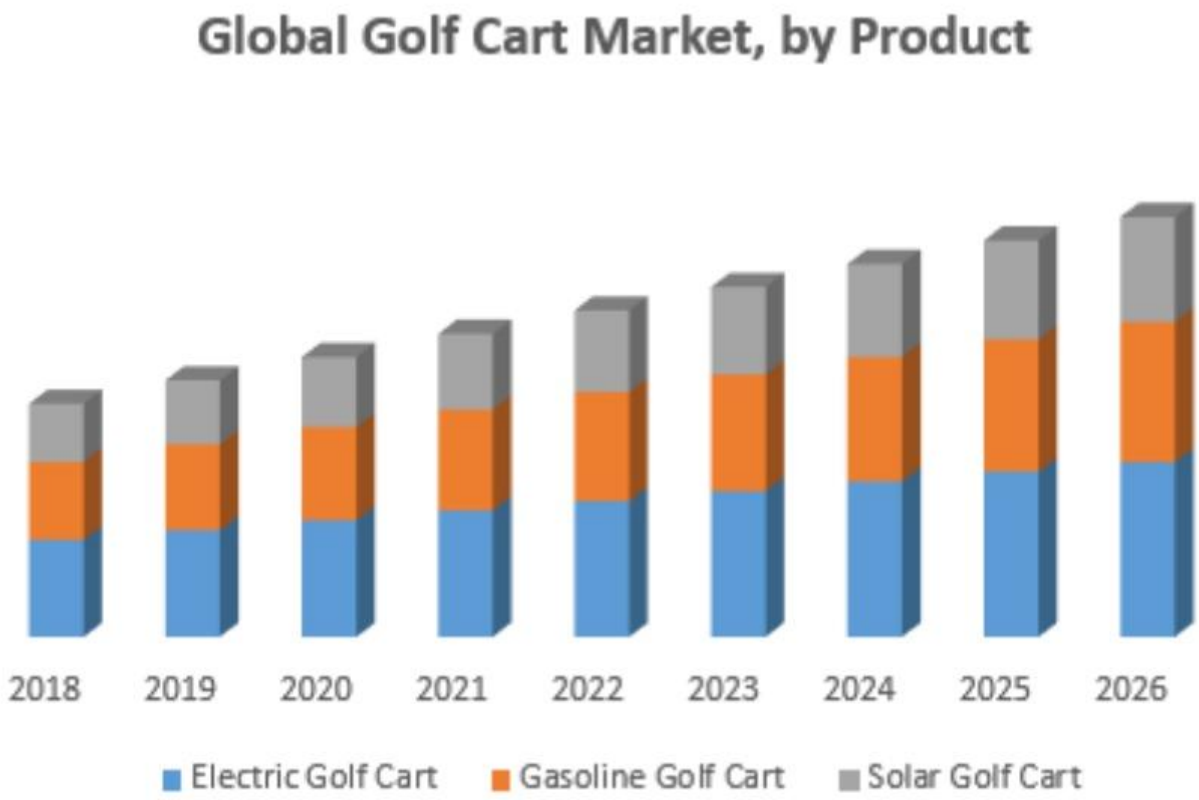
Solar golf carts are powered by mounting a photovoltaic (PV) or thin film panel on top of the existing roof or using a PV panel as the roof itself. A controller converts the sun's energy to charge the golf cart's 36-volt or 48-volt battery bank. Not only does the solar power take the cart off the electric grid, it also increases the driving distance and extends the life of the batteries.

Solar conversion kits have been available for golf carts and Low Speed Vehicles (LSV) for several years. These kits range from low wattage solar battery chargers to a 410-watt array on an 8-passenger transport cart. Kits utilizing flexible solar panels are often preferred on golf carts due to their light weight and ability to conform to the shape of the cart roof. Buyers can take a solar tax credit of 30% of the purchase price on their US Federal Income Tax.

Solar-powered golf carts are popular with owners who drive long distances, such as maintenance workers, golf course fleets, staff at the Detroit Zoo, transportation vehicles in resorts and cities, and drivers of VIP carts on college campuses.

COST ANALYSIS

Figure 2: India Golf Cart Market Value (US$ Mn), 2016 - 2027



REPORT

STRUCTURE AND SPECIFICATIONS

(LT-A627·4)Whole car specifications (standard) :
detailed specification refers to actual contract

Electronic system	controller	Curtis controller 1266A-5201/275A (USA Brand)		
	battery	Lead acid battery 8V x 6pcs 140AH 5Hr Rate		
	Motor	High efficient Sepex ADC 3.7 KW motor		
	charger	Intelligent charger48V,20A		
	Accelerator	Electrical accelerator 48V input, Hall output 0-5V		
Body configuration	Wind shield	Organic glass	Top roof	ABS plastic vacuum forming
	Seat	Leather	Floor	Rubber sliding proof mat
	Side mirror	L & R side mirrors	Steering system	Two way gear steering system,automatically backlash compensation
	Light&signal	48V/12-300W Front light, turning light, rear combo lights, horn, reverse buzz		
	Body	PP plastic injection	chassis	High tensile steel welding structure, electrophoresis treatment for anti-rust
	Dash board	PP plastic injection, power indicator, head light switch, turning light switch, key switch, F/R switch		
	Brake	rear drum + combo brake pedal parking		
	Front Suspension	Spring+hydraulic shock absorber, independent suspension		
	Rear suspension	Excellent transaxle, gear ratio 12.31:1, half independent suspension + hydraulic shock absorber		
	Wheel & tire	Steel wheel 14×8, 5/5×8+tire (452mm)		
	Loading staff	4 persons (including driver)		

1. specification

1.1dimension:L*W*H	3160*1200*1800mm
1.2 wheelbase	2410mm
1.3weight(including batter)	510kg
1.4load capacity	300kg
1.5max speed	22-24km/h
1.6mileage	70-90km/h (level road)
1.7min turning radius	4m
1.8max climbing ability	25%
1.9min clearance	100mm
1.10brake distance	<6m
1.11tread	front900mm/rear 1000mm

2.battery specification

2.1battery type	lead acid battery
2.2volume	140Ah
2.3 voltage	48V

3.motor specification

3.1motor type	DC series motor
3.2rated power	3.7kw
3.3rated rotating speed	2500r/min
3.4 rated voltage	48V
3.5rated currency	105A
3.6ration	15min

STRUCTURE AND SPECIFICATIONS

(LT-A627·4+2)Whole car specifications (standard) :
detailed specification refers to actual contract

Electronic system	controller	Curtis controller 1266A-5201/275A (USA Brand)		
	battery	Lead acid battery 8V x 6pcs 140AH 5Hr Rate		
	Motor	High efficient Sepex ADC 3.7 KW motor		
	charger	Intelligent charger48V,20A		
	Accelerator	Electrical accelerator 48V input, Hall output 0-5V		
Body configuration	Wind shield	Organic glass	Top roof	ABS plastic vacuum forming
	Seat	Leather	Floor	Rubber sliding proof mat
	Side mirror	L & R side mirrors	Steering system	Two way gear steering system,automatically backlash compensation
	Light&signal	48V/12-300W Front light, turning light, rear combo lights, horn, reverse buzz		
	Body	PP plastic injection	chassis	High tensile steel welding structure, electrophoresis treatment for anti-rust
	Dash board	PP plastic injection, power indicator, head light switch, turning light switch, key switch, F/R switch		
	Brake	rear drum + combo brake pedal parking		
	Front Suspension	Spring+hydraulic shock absorber, independent suspension		
	Rear suspension	Excellent transaxle, gear ratio 12.31:1, half independent suspension + hydraulic shock absorber		
	Wheel & tire	Steel wheel 14×8, 5/5×8+tire (452mm)		
	Loading staff	6 persons (including driver)		

1. specification

1.1dimension:L*W*H	3520*1200*1800mm
1.2 wheelbase	2410mm
1.3weight(including batter)	590kg
1.4load capacity	450kg
1.5max speed	22-24km/h
1.6mileage	70-80km/h (level road)
1.7min turning radius	4m
1.8max climbing ability	20%
1.9min clearance	100mm
1.10brake distance	<6m
1.11tread	front900mm/rear 1000mm

2.battery specification

2.1battery type	lead acid battery
2.2volume	140Ah
2.3 voltage	48V

3.motor specification

3.1motor type	DC series motor
3.2rated power	3.7kw
3.3rated rotating speed	2500r/min
3.4 rated voltage	48V
3.5rated currency	105A
3.6ration	15min

STRUCTURE AND SPECIFICATIONS

(LT-A627·2)Whole car specifications (standard) :
detailed specification refers to actual contract

Electric system	controller	Curtis controller 1266A-5201/275A (USA Brand)	
	battery	Lead acid battery 8V x 6pcs 140AH 5Hr Rate	
	Motor	High efficient Sepex ADC 3.7 KW motor	
	charger	Intelligent charger48V,20A	
	Accelerator	Electrical accelerator 48V input, Hall output 0-5V	
Body configuration	Wind shield	Organic glass	Top roof: ABS plastic vacuum forming
	Seat	Leather	Floor: Rubber sliding proof mat
	Side mirror	L & R side mirrors	Steering system: Steering gear meshing system,automatically backlash compensation
	Light&signal	48V/12-300W Front light, turning light, rear combo lights, horn, reverse buzz	
	Body	PP plastic injection	chassis: High tensile steel welding structure, electrophoresis treatment for anti-rust
	Dash board	PP plastic injection, power indicator, head light switch, turning light switch, key switch, F/R switch	
	Brake	rear drum + combo brake pedal parking	
	Front Suspension	Spring+hydraulic shock absorber, independent suspension	
	Rear suspension	Excellent transaxle, gear ratio 12.31:1, half independent suspension + hydraulic shock absorber	
	Wheel & tire	Steel wheel 10×8, 10×8+tire (452mm)	
	Loading staff	2 persons(including driver)	

1. specification

1.1dimension:L*W*H	2400*1200*1800mm
1.2 wheelbase	1650mm
1.3weight(including batter)	470kg
1.4load capacity	150kg
1.5max speed	22-24km/h
1.6mileage	80-100km/h (level road)
1.7min turning radius	2.9m
1.8max climbing ability	25%
1.9min clearance	100mm
1.10brake distance	<6m
1.11tread	front900mm/rear 1000mm

2.battery specification

2.1battery type	lead acid battery
2.2volume	140Ah
2.3 voltage	48V

3.motor specification

3.1motor type	DC series motor
3.2rated power	3.7kw
3.3rated rotating speed	2500r/min
3.4 rated voltage	48V
3.5rated currency	105A
3.6ration	15min

STRUCTURE AND SPECIFICATIONS

(LT-A627·2+2)Whole car specifications (standard) :
detailed specification refers to actual contract

Electric system	controller	Curtis controller 1266A-5201/275A (USA Brand)	
	battery	Lead acid battery 8V x 6pcs 140AH 5Hr Rate	
	Motor	High efficient Sepex ADC 3.7 KW motor	
	charger	Intelligent charger48V,20A	
	Accelerator	Electrical accelerator 48V input, Hall output 0-5V	
Body configuration	Wind shield	Organic glass	Top roof: ABS plastic vacuum forming
	Seat	Leather	Floor: Rubber sliding proof mat
	Side mirror	L & R side mirrors	Steering system: Steering gear meshing system,automatically backlash compensation
	Light&signal	48V/12-300W Front light, turning light, rear combo lights, horn, reverse buzz	
	Body	PP plastic injection	chassis: High tensile steel welding structure, electrophoresis treatment for anti-rust
	Dash board	PP plastic injection, power indicator, head light switch, turning light switch, key switch, F/R switch	
	Brake	rear drum + combo brake pedal parking	
	Front Suspension	Spring+hydraulic shock absorber, independent suspension	
	Rear suspension	Excellent transaxle, gear ratio 12.31:1, half independent suspension + hydraulic shock absorber	
	Wheel & tire	Steel wheel 10×8, 10×8+tire (452mm)	
	Loading staff	4 persons(including driver)	

1. specification

1.1dimension:L*W*H	2760*1200*1800mm
1.2 wheelbase	1650mm
1.3weight(including batter)	550kg
1.4load capacity	300kg
1.5max speed	22-24km/h
1.6mileage	80-90km/h (level road)
1.7min turning radius	2.9m
1.8max climbing ability	25%
1.9min clearance	100mm
1.10brake distance	<6m
1.11tread	front900mm/rear 1000mm

2.battery specification

2.1battery type	lead acid battery
2.2volume	140Ah
2.3 voltage	48V

3.motor specification

3.1motor type	DC series motor
3.2rated power	3.7kw
3.3rated rotating speed	2500r/min
3.4 rated voltage	48V
3.5rated currency	105A
3.6ration	15min

PRE OPERATION

Pre-operation checks should be made each time you use your golf car. Get in the habit of performing the following checks in the same way so that they become second nature.

WARNING

Be sure that the main switch key is removed before performing the pre-operating checks to prevent accidental starting, and apply the parking brake to keep the car from moving.

PRE-OPERATION CHECKLIST

Before each use, please check the following

- √Batteries
- √Tire condition
- √Steering system
- √Back-up buzzer
- √Pedal operation
- √Body and chassis

SEAT

Opening the seat for checking and servicing

PRE OPERATION

BATTERY

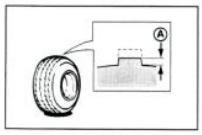
Charge batteries before every use.See charging steps in chapter 6.

Check that the batteries are held securely in place to prevent the batteries from vibration or jarring.Also check that no prevent battery acid from spilling from the battery.Check the battery terminals for corrosion.

TIRE CONDITION

Check the tire air pressure before the operation of car.

Tire pressure
25PSI grass/dirt
36 PSI road



Tire wear limit

Check the tire surface for damage,cracks or embedded objects.When tire tread wears down to0.04(1mm),replace the tire.

Ⓐ Wear limit

OPERATION AND CONTROL

Turning On Main Switch

1,Shifting the switch to RUN position then can turn on the power supply of car.

2,Shifting the switch to OFF position then can turn off the power supply of car.

Key Switch

1、Plugging in the key and turn right then can turn on the key switch(Note:Shifting the F/R switch to the middle position before turning on Main switch).

2、Turning left and then can turn off the switch.Switch can be removed only at this state.

Forward/Reverse Switch

The Forward/Reverse Switch is used to shift the car to forward,reverse or stop.Forward for upward,Reverse for downward,Park for middle.When choosing reverse switch,the reverse buzzer sounds.

Completely stopping car before shifting F/R switch.When shifting F/R switch,please turn switch to middle positon for 2s at first,then choosing Forward switch or Reverse switch.Don't shifting F/R switch in a hurry in case that the sensor will be burnt or switch will be out of use.

OPERATION AND CONTROL

Voltage Indicator

Voltage Indicator is to show the electricity and remains custom to charge on time.The voltage will shows fulled charged when charing enough time.

1、Electricity display reduce with the consumption of electricity during working;

2、When electricity is close to little electricity warning,it was showed by moving between first space and second space;

3、When electricity enters into under-electricity warning,it was displayed by flashing of first space in left.

Head light switch

Turning light switch

Accelerator pedal

Turn on the key switch, choose forward or backward, release brake, put your right foot on the accelerator pedal, soft step down the pedal to start the car.

Attention: don't step down the pedal to the end in rush.

STRUCTURE AND SPECIFICATIONS

(LT-A627·6)Whole car specifications (standard) :
detailed specification refers to actual contract

Electric system	controller	Curtis controller 1266A-5201/275A(USA Brand)	
	battery	Lead acid battery 8V x 6pcs 140AH 5Hr Rate	
	Motor	High efficient Sapex ADC 3.7 KW motor	
	charger	Intelligent charger48V,20A	
	Accelerator	Electrical accelerator 48V input, Hall output 0-5V	
Body configuration	Wind shield	Organic glass	
	Seat	Leather	
	Side mirror	L & R side mirrors	
	Light&signal	48V/12-300W Front light, turning light, rear combo lights, horn, reverse buzz	
	Body	PP plastic injection	
	Dash board	PP plastic injection, power indicator, head light switch, turning light switch, key switch, F/R switch	
	Brake	rear drum + combo brake pedal parking	
	Front Suspension	Spring+hydraulic shock absorber, independent suspension	
	Rear suspension	Excellent transaxle, gear ratio 12.31:1, half independent suspension + hydraulic shock absorber	
	Wheel & tire	Steel wheel 10×8, 10×8+tire (452mm)	
	Loading staff	6 persons (including driver)	
	Top roof	ABS plastic vacuum forming	
	Floor	Rubber sliding proof mat	
	Steering system	Two way gear shifting system, automatically backlash compensation	
	chassis	High tensile steel welding structure, electrophoresis treatment for anti-rust	

1. specification

1.1dimension:L*W*H	3920*1200*1800mm
1.2 wheelbase	3170mm
1.3weight(including batter)	550kg
1.4load capacity	450kg
1.5max speed	22-24km/h
1.6mileage	70-80km/h (level road)
1.7min turning radius	5.1m
1.8max climbing ability	20%
1.9min clearance	100mm
1.10brake distance	<6m
1.11tread	front900mm/rear 1000mm

2.battery specification

2.1battery type	lead acid battery
2.2volume	140Ah
2.3 voltage	48V

3.motor specification

3.1motor type	DC series motor
3.2rated power	3.7kw
3.3rated rotating speed	2500r/min
3.4 rated voltage	48V
3.5rated currency	105A
3.6ration	15min

STRUCTURE AND SPECIFICATIONS

(LT-A627·6+2)Whole car specifications (standard) :
detailed specification refers to actual contract

Electric system	controller	Curtis controller 1268-5403/400A (USA Brand)	
	battery	Lead acid battery 8V x 6pcs 140AH 5Hr Rate	
	Motor	High efficient Sapex ADC 3.7 KW motor	
	charger	Intelligent charger48V,20A	
	Accelerator	Electrical accelerator 48V input, Hall output 0-5V	
Body configuration	Wind shield	Organic glass	
	Seat	Leather	
	Side mirror	L & R side mirrors	
	Light&signal	48V/12-300W Front light, turning light, rear combo lights, horn, reverse buzz	
	Body	PP plastic injection	
	Dash board	PP plastic injection, power indicator, head light switch, turning light switch, key switch, F/R switch	
	Brake	rear drum + combo brake pedal parking	
	Front Suspension	Spring+hydraulic shock absorber, independent suspension	
	Rear suspension	Excellent transaxle, gear ratio 12.31:1, half independent suspension + hydraulic shock absorber	
	Wheel & tire	Steel wheel 10×8, 10×8+tire (452mm)	
	Loading staff	8 persons (including driver)	
	Top roof	ABS plastic vacuum forming	
	Floor	Rubber sliding proof mat	
	Steering system	Two way gear shifting system, automatically backlash compensation	
	chassis	High tensile steel welding structure, electrophoresis treatment for anti-rust	

1. specification

1.1dimension:L*W*H	4280*1200*1800mm
1.2 wheelbase	3170mm
1.3weight(including batter)	630kg
1.4load capacity	600kg
1.5max speed	22-24km/h
1.6mileage	70-80km/h (level road)
1.7min turning radius	5.1m
1.8max climbing ability	20%
1.9min clearance	100mm
1.10brake distance	<6m
1.11tread	front900mm/rear 1000mm

2.battery specification

2.1battery type	lead acid battery
2.2volume	140Ah
2.3 voltage	48V

3.motor specification

3.1motor type	DC series motor
3.2rated power	3.7kw
3.3rated rotating speed	2500r/min
3.4 rated voltage	48V
3.5rated currency	105A
3.6ration	15min

Mini Project 8– Wiper System [Individual]

Modules

1. AUTOSAR
2. Git

Wiper System:

Requirements

Research

Automation is a necessity in our day to day life because it not only seeks to improve the quality of life for humans at both home and work.

- It allows the distribution of both quality products and services to be made available at faster rates, reduces down time and human error.
- This days the technology is developing at very high speed innovation and advancements is their in all the sectors.

SWOT- Strengths, and Weakness, Opportunities Threats:

Strengths

Comfortable, Easy to use, Fast and accuracy operation, Easy transportation

Weakness

High cost,

Opportunities

it's a modern way of approach, High use in modern days

Threat

if sensor is damages, operation of the system will stop

4W's and 1'H:

WHO :-

Any user who is travelling in the car can use this system.

WHAT :-

Its a Body control model of a car which describes different features

WHEN :-

There is a requirement of vehicle these systems are used

WHERE:-

Used in automatic cars

HOW :-

By developing a embedded system which is user friendly and can be implemented without difficulty.

Detail requirements

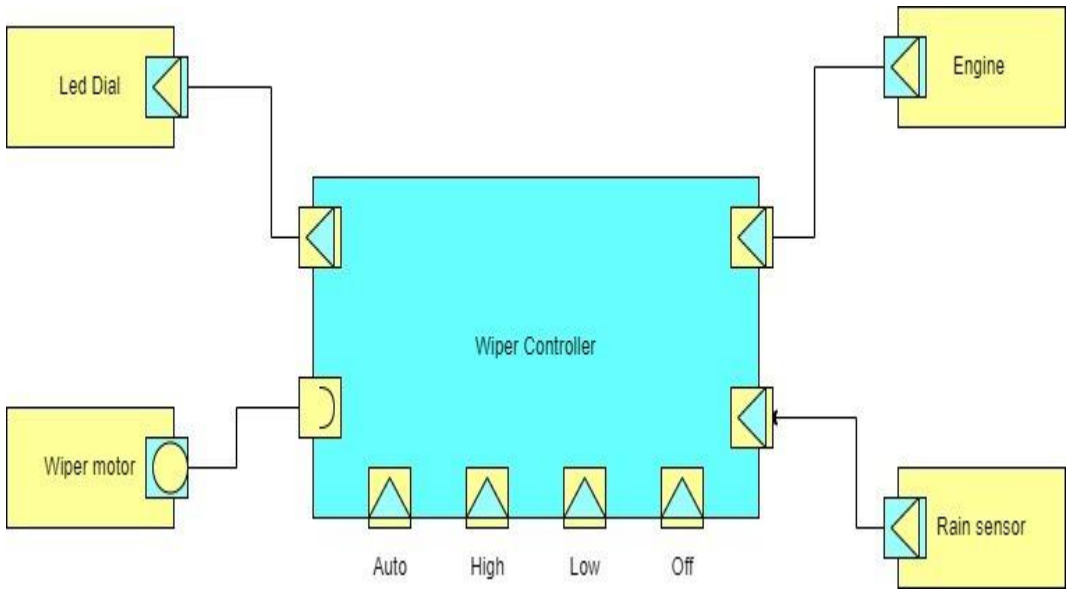
High Level Requirements

Number	Requirements	Description
HLR1	Rain and Humidity sensor	checks the intensity of rain and humidity
HLR2	Semi-Automatic control	Operated automatically or manually
HLR3	Speed control	High, Medium, Low
HLR4	Dry mode	front and back wiper system

Low Level Requirements

Number	Requirement	Description
LLR1	Length of wipers	Depends on the model of the car
LLR2	Direction of wipers	Same and Opposite Directions (180 degrees)

Design



Implementation and Summary

Git Link:
https://github.com/pasumarthiharitha/Automotive_Lamborghini