./

GENESIS - Learning Outcome & Mini-project Summary Report



|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Ver. Rel. No.** | **Release Date** | **Prepared. By** | **Reviewed By** | **To be Approved** | **Remarks/Revision Details** |
| 1 | 05/05/2021 | Aditya Mewara(99003733) |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |

**Details**

Contents

[Contents 3](#_Toc71795662)

[Table of Figures 4](#_Toc71795663)

[Table of Tables 4](#_Toc71795664)

[Miniproject -1 [Team: 99003731, 99003733, 99003734, 99003735] 6](#_Toc71795665)

[Module 6](#_Toc71795666)

[Topic and Subtopics 6](#_Toc71795667)

[Objectives & Requirements 6](#_Toc71795668)

[**Design** 10](#_Toc71795669)

[Test Plan 15](#_Toc71795670)

[Implementation Summary 18](#_Toc71795671)

[Video Summary 18](#_Toc71795672)

[Git Link 18](#_Toc71795673)

[Git Dashboard 19](#_Toc71795674)

[Summary 19](#_Toc71795675)

[Individual Contribution & Highlights 26](#_Toc71795676)

[Summary 26](#_Toc71795677)

[Challenges faced and how were they overcome 27](#_Toc71795678)

[Future Scope 27](#_Toc71795679)

[Miniproject -2 [Team] – Embedded C 28](#_Toc71795680)

[Module/s 28](#_Toc71795681)

[Topic and Subtopics 28](#_Toc71795682)

[Objectives & Requirements 28](#_Toc71795683)

[Design 30](#_Toc71795684)

[Test Plan 33](#_Toc71795685)

[Implementation Summary 34](#_Toc71795686)

[Git Link 34](#_Toc71795687)

[Git Dashboard 34](#_Toc71795688)

[Summary 35](#_Toc71795689)

[Individual Contribution & Highlights: 35](#_Toc71795690)

[Challenges faced and how were they overcome 35](#_Toc71795691)

[Future Scope (If applicable): 35](#_Toc71795692)

[Miniproject -3 [Individual] 36](#_Toc71795693)

[Module 36](#_Toc71795694)

[Topic and Subtopics 36](#_Toc71795695)

[Objectives & Requirements 37](#_Toc71795696)

[Design 39](#_Toc71795697)

[Test Plan 41](#_Toc71795698)

[Implementation Summary 43](#_Toc71795699)

[Video Summary 43](#_Toc71795700)

[Git Link 43](#_Toc71795701)

[Git Dashboard 44](#_Toc71795702)

[Summary 44](#_Toc71795703)

[Challenges faced and how were they overcome 44](#_Toc71795704)

[Future Scope 44](#_Toc71795705)

[Miniproject -4 [Individual-Activity] 45](#_Toc71795706)

[Module 45](#_Toc71795707)

[Topic and Subtopics 45](#_Toc71795708)

[Objectives & Requirements 45](#_Toc71795709)

[Test Plan 47](#_Toc71795710)

[Implementation Summary 48](#_Toc71795711)

[Video Summary 48](#_Toc71795712)

[Git Link 48](#_Toc71795713)

[Summary 49](#_Toc71795714)

[Challenges faced and how were they overcome 49](#_Toc71795715)

[Miniproject -5 [Individual]: Networking 50](#_Toc71795716)

[Module/s 50](#_Toc71795717)

[Topic and Subtopics 50](#_Toc71795718)

[Objectives & Requirements 50](#_Toc71795719)

[Design 50](#_Toc71795720)

[Test Plans: 51](#_Toc71795721)

[Testing Results: 53](#_Toc71795722)

[Implementation Summary 55](#_Toc71795723)

[Individual Contribution & Highlights: 55](#_Toc71795724)

[Challenges faced and how were they overcome 55](#_Toc71795725)

[Future Scope (If applicable): 55](#_Toc71795726)

Table of Figures

[Figure 1. Class-Diagram 9](#_Toc69548261)

[Figure 2. Use-Case Diagram 10](#_Toc69548262)

[Figure 3.Object-Diagram – (Basic Arithmetic, Matrix, Combinatorics) 11](#_Toc69548263)

[Figure 4.Class-Diagram – (Physics) 11](#_Toc69548264)

[Figure 5. Class Diagram – (Financial, Profit loss) 12](#_Toc69548265)

[Figure 6.Use-Case Diagram 13](#_Toc69548266)

[Figure 7.Sequential Diagram 13](#_Toc69548267)

[Figure 8.Workflow-Badges 18](file:///C:\Users\99003731\Downloads\Gen%20LR.docx#_Toc69548268)

[Figure 9.High-Level Design(Activity-Diagram) -Embedded C 29](file:///C:\Users\99003731\Downloads\Gen%20LR.docx#_Toc69548269)

[Figure 10.Low-Level Design (Use-Case Diagram) Embedded C 30](file:///C:\Users\99003731\Downloads\Gen%20LR.docx#_Toc69548270)

[Figure 11.Implemented Circuit of Door Latch-Detection System 34](file:///C:\Users\99003731\Downloads\Gen%20LR.docx#_Toc69548271)

[Figure 12.Implemented Circuit of Parking Assistance System 34](file:///C:\Users\99003731\Downloads\Gen%20LR.docx#_Toc69548272)

[Figure 13.High-Level Design (Activity Diagram) Python 39](file:///C:\Users\99003731\Downloads\Gen%20LR.docx#_Toc69548273)

[Figure 14. Low-Level Design (Activity Diagram) Python 40](#_Toc69548274)

Table of Tables

[Table 1. High-Level Requirements(SDLC) 6](#_Toc69548506)

[Table 2.Low-Level Requirements(SDLC) 8](#_Toc69548507)

[Table 3. High-Level Testing(SDLC) 14](#_Toc69548508)

[Table 4. Low -Level Testing(SDLC) 17](#_Toc69548509)

[Table 5. High Level Requirements (Embedded C) 28](#_Toc69548510)

[Table 6. Low-Level Requirements (Embedded C) 28](#_Toc69548511)

[Table 7. High-Level Testing (Embedded C) 31](#_Toc69548512)

[Table 8. Low -Level Testing (Embedded C) 32](#_Toc69548513)

[Table 9.High-Level Requirements (Python) 37](#_Toc69548514)

[Table 10.Low-Level Requirements (Python) 38](#_Toc69548515)

[Table 11.High-Level Testing (Python) 41](#_Toc69548516)

[Table 12.Low-Level Testing (Python) 42](#_Toc69548517)

[Table 13.Requirements (Kernel driver development) 46](#_Toc69548518)

[Table 14.Test-Plan (Kernel Driver Development) 47](#_Toc69548519)

# Miniproject -1 [Team: 99003731, 99003733, 99003734, 99003735]

## Module

“SDLC (System Development Life-Cycle)”

### Topic and Subtopics

#### Core -Topics

* V-Model

##### Sub-Topic

* C Programming
* MakeFile
* Unit Testing
* Version Control (via GitHub)
* Agile-Methodology

##### Sub-Topic

* Theme
* Epic
* User-Story

All these Core-topics and sub-topics are implemented through V-Model Technique.

## Objectives & Requirements

#### Objective

To design a calculator, which has features according to the specific requirements given by following V-Model flow.

#### Requirements

##### High Level Requirements

|  |  |  |  |
| --- | --- | --- | --- |
| **ID** | **Requirements** | **Description** | **Status** |
| HLRAM001 | Physics Calculation | Several Physics functions are implemented | Implemented, Tested |
| HLRAM002 | Profit and Loss Calculation | Day to Day use of profit and loss function make it more useful | Implemented, Tested |
| HLRSS003 | Basic Arithmetic Calculations | Very basic operations like +, -, /, \* | Implemented, Tested |
| HLRSS004 | Matrix operation | Several Matrix functions | Implemented, Not Tested |
| HLRSS005 | Combinatorics Calculation | Permutation and Combination | Implemented, Tested |
| HLRSA006 | Special math functions | functions like logarithm, exponential, square root | Implemented, Tested |
| HLRSA007 | financial calculation | Simple interest, Compound interest, EMI calculation | Implemented, Tested |
| HLRPK008 | Mensuration | Area, Volume | Implemented, Tested |
| HLRPK009 | AP, GP, Sum | Mean | Implemented, Tested |
| HLRAM001 | Physics Calculation | Several Physics functions are implemented | Implemented, Tested |
| HLRAM002 | Profit and Loss Calculation | Day to Day use of profit and loss function make it more useful | Implemented, Tested |

Table 1. High-Level Requirements(SDLC)

##### Low Level Requirements

|  |  |  |  |
| --- | --- | --- | --- |
| ID | Requirements | Description | Status |
| HLRAM001 – LR1 | Acceleration Calculation | By taking v, u, t as an input | Implemented |
| HLRAM001 – LR2 | Capacitive reactance Calculation | By taking C, f as an input | Implemented |
| HLRAM001 – LR3 | Circular velocity Calculation | By taking r, t as an input | Implemented |
| HLRAM001 – LR4 | Coulombs law Calculation | Making 'K' constant as well as taking q1, q2 and distance between them as an input | Implemented |
| HLRAM001 – LR5 | Projectile motion Calculation | Initial Velocity, Angle and g as an input | Implemented |
| HLRAM002 – LR1 | Profit calculation | Taking SP and CP as user input and Check SP > CP | Implemented |
| HLRAM002 – LR2 | Loss calculation | Taking SP and CP as user input and Check CP > SP | Implemented |
| HLRAM002 – LR3 | Discount calculation | Taking input as CP and enter the percentage of discount as an input | Implemented |
| HLRSS003 - LR1 | Addition | Taking two numbers as input | Implemented |
| HLRSS003 - LR2 | Subtraction | Taking two numbers as input | Implemented |
| HLRSS003 - LR3 | Multiply | Taking two numbers as input | Implemented |
| HLRSS003 - LR4 | Divide | Taking two numbers as input | Implemented |
| HLRSS004 - LR1 | Matrix Addition | Taking two matrices as input and their dimension | Implemented |
| HLRSS004 - LR2 | Matrix Subtraction | Taking two matrices as input and their dimension | Implemented |
| HLRSS004 - LR3 | Matrix Multiplication | Taking two matrices as input and their dimension | Implemented |
| HLRSS004 - LR4 | Matrix Determinant | Taking a matrix as input and its dimension | Implemented |
| HLRSS004 - LR5 | Matrix Row or Column sum | Taking a matrix as input and its dimension with specific row or column | Implemented |
| HLRSS005 - LR1 | Permutation | Taking input for total no. of objects and for no. of permutations | Implemented |
| HLRSS006 - LR2 | Combination | Taking input for total no. of objects and for no. of permutations | Implemented |
| HLRPK007 - LR1 | Area | Initiating the required input | Implemented |
| HLRPK007 - LR2 | Volume | Initiating the required input | Implemented |
| HLRPK008 - LR1 | Mean | Initiating the required input | Implemented |
| HLRSA009 - LR1 | logarithm | Taking one number as input with base as 10 | Implemented |
| HLRSA009 - LR2 | exponential | Taking one number as input | Implemented |
| HLRSA009 - LR3 | Square root | Taking one number as input | Implemented |
| HLRSA0010 - LR1 | Simple interest calculator | Taking Principal amount, Time-period, Rate of interest as input | Implemented |
| HLRSA0010 - LR2 | Compound interest calculator | Taking Principal amount, Time-period, Rate of interest as input | Implemented |
| HLRSA0010 - LR3 | EMI Calculator | Taking Principal amount, Time-period, Rate of interest as input | Implemented |

Table 2.Low-Level Requirements(SDLC)

## **Design**

#### System Level Design (High-Level)

##### Structural

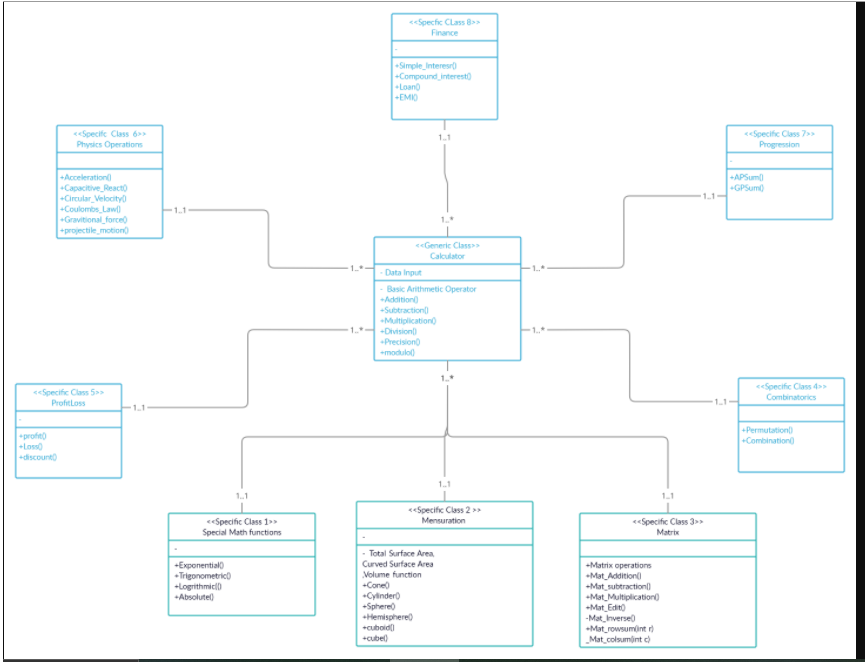


Figure 1. Class-Diagram

##### Behavioral

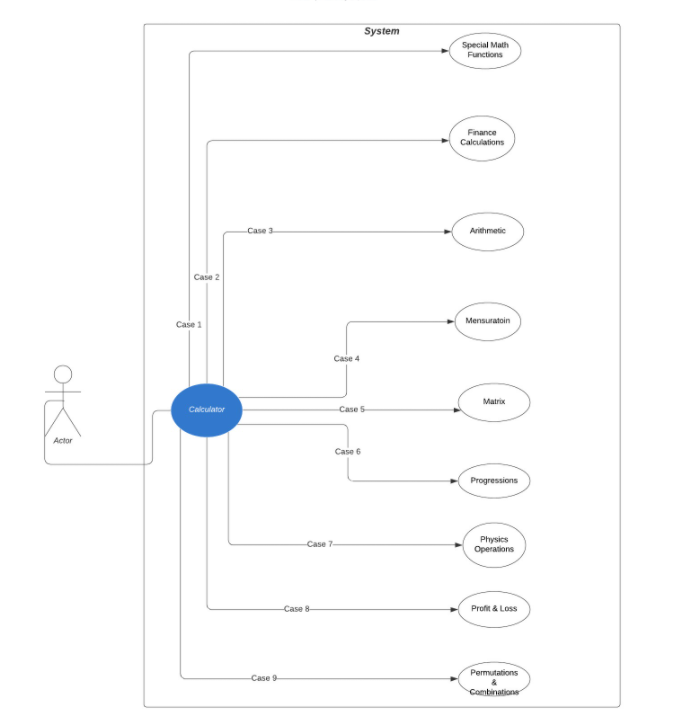


Figure 2. Use-Case Diagram

#### Sub-System Level (Low-Level Design)

##### Structural

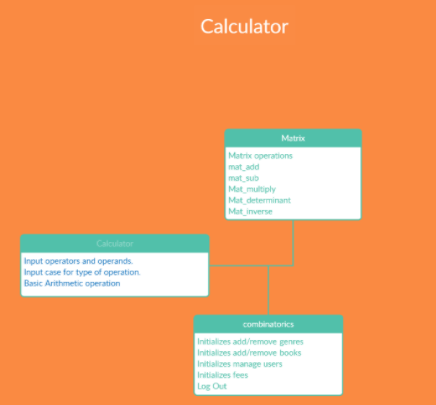


Figure 3.Object-Diagram – (Basic Arithmetic, Matrix, Combinatorics)

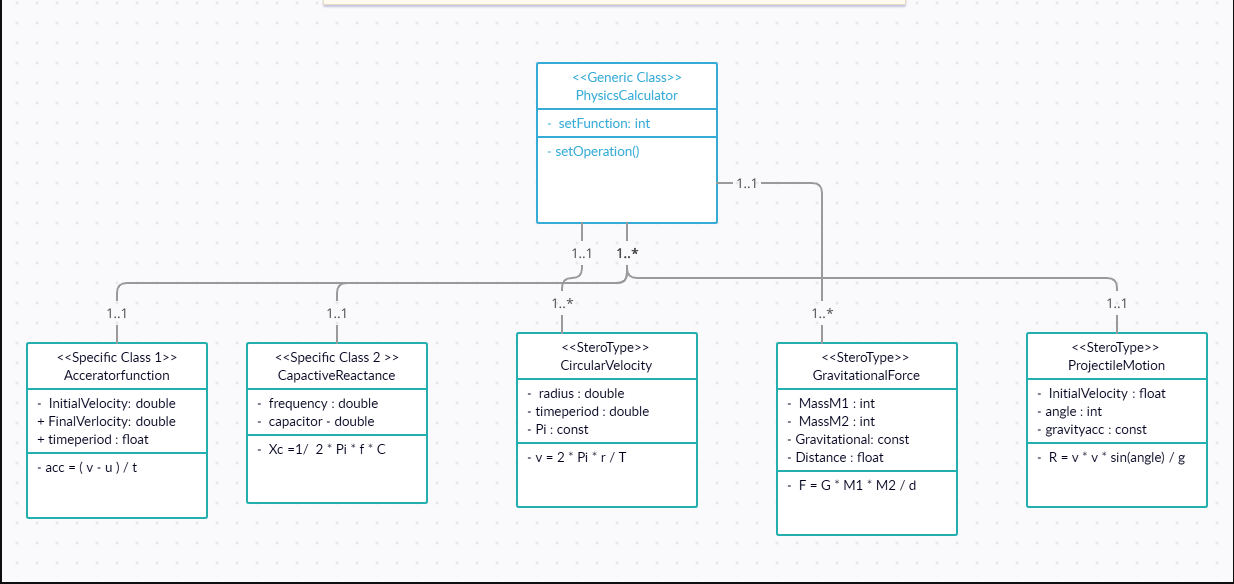


Figure 4.Class-Diagram – (Physics)

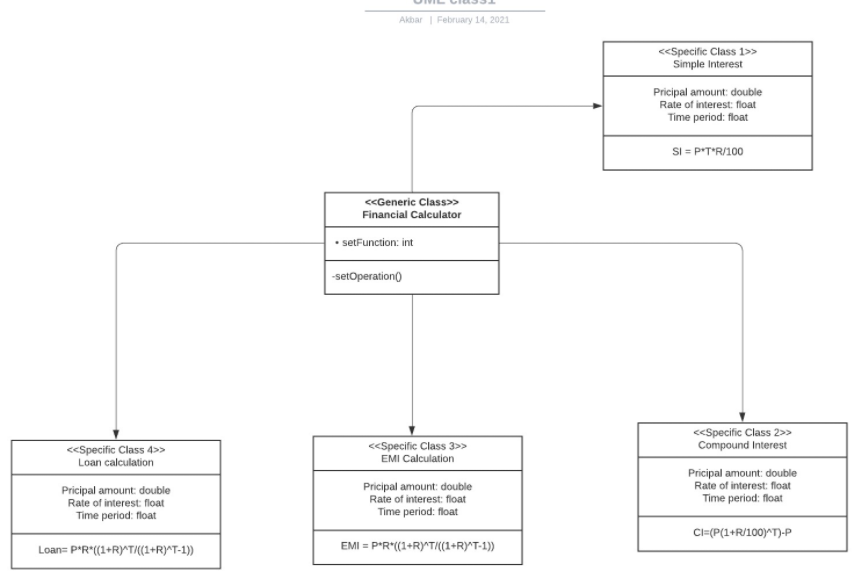


Figure 5. Class Diagram – (Financial, Profit loss)

##### Behavioral

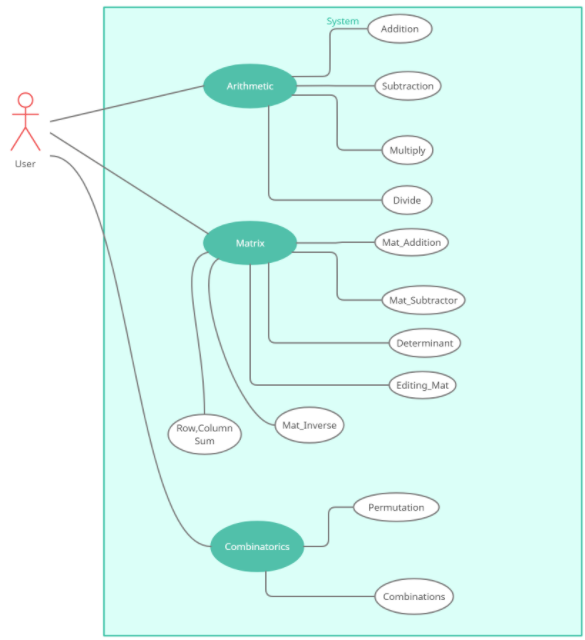


Figure 6.Use-Case Diagram

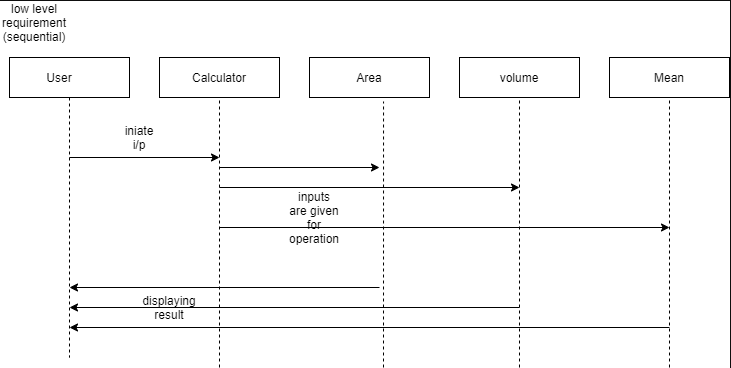


Figure 7.Sequential Diagram

## Test Plan

#### Integration Level Testing

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Test ID** | **Description** | **Expected Input** | **Expected Output** | **Actual Out** | **Type of Test** |
| H\_01 | mode switch working for special operations | mode==1 | shows mode menu | shows mode menu | Requirement based |
| H\_02 | mode switch working | mode==2 | Error | Error | Scenario based |

Table 3. High-Level Testing(SDLC)

#### Unit-Level Testing

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Test ID** | **Description** | **Expected Input** | **Expected Output** | **Actual Out** | **Type of Test** |
| L\_0111 | Adding two numbers | num1= 30, num2=20 | result=50 | result=50 | Requirement based |
| L\_0121 | Subtracting two numbers | num1= 0, num2=3 | result=-3 | result=-3 | Requirement based |
| L\_0131 | Multiplying two numbers | num1= 1, num2=0 | result=0 | result=0 | Requirement based |
| L\_0141 | Dividing two numbers | num1= 27, num2=9 | result=3 | result=3 | Requirement based |
| L\_0211 | Adding two negative numbers | num1=-750, num2=-7500 | result=-8250 | result=-8250 | Scenario based |
| L\_0212 | Adding one negative number and one positive number | num1=10, num2=-20 | result=-10 | result=-10 | Scenario based |
| L\_0213 | Adding one positive number and one negative number | num1=-98, num2=20 | result=-78 | result=-78 | Scenario based |
| L\_0221 | Subtracting two positive numbers | num1=13, num2=4 | result=9 | result=9 | Scenario based |
| L\_0222 | Subtracting positive number from a negative number | num1=-45, num2=6 | result=-51 | result=-51 | Scenario based |
| L\_0223 | Subtracting two negative numbers | num1=-7985, num2=-7745 | result=-240 | result=-240 | Scenario based |
| L\_0231 | Multiplying one positive number with one negative number | num1=-20, num2=3 | result=-60 | result=-60 | Scenario based |
| L\_0232 | Multiplying one negative number with one positive number | num1=8745, num2=-83 | result=-725835 | result=-725835 | Scenario based |
| L\_0233 | Multiplying two negative numbers | num1=-99, num2=-999 | result=98901 | result=98901 | Scenario based |
| L\_0241 | Dividing by zero | num1=1, num2=0 | result=-1(for error) | result=-1 | Scenario based |
| L\_0242 | Dividing two negative number | num1=-135, num2=-3 | result=45 | result=45 | Scenario based |
| L\_0243 | Dividing negative number by positive number | num1=-28, num2=14 | result=-2 | result=-2 | Scenario based |
| L\_0244 | Dividing positive number by negative number | num1=96, num2=-12 | result=-8 | result=-8 | Scenario based |
| L\_0311 | Multiplying two 8 digit numbers | num1=100000000, num2=99999999 | result=-1 | result=-1(for Error) | Boundary based |
| L\_0411 | Calculating Permutations | n=5, r=1 | permutations=5 | permutations=5 | Requirement based |
| L\_0421 | Calculating Combinations | n=2, r=1 | combinations=2 | combinations=2 | Requirement based |
| L\_0511 | If number of required permutations is greater than total number of objects | n=4, r=6 | permutations=-1(for Error) | permutations=-1 | Scenario based |
| L\_0512 | If entered inputs are negative for permutation function | n=-3, r=1 | combinations=-1(for Error) | combinations=-1 | Scenario based |
| L\_0521 | If number of required combinations is greater than total number of objects | n=4, r=10 | combinations=-1(for Error) | combinations=-1 | Scenario based |
| L\_0522 | If entered inputs are negative for combination function | n=4, r=10 | combinations=-1(for Error) | combinations=-1 | Scenario based |
| L\_0611 | Calculating Acceleration | v=10, u=5, t=5 | accefunc = 1 | accefunc = 1 | Scenario based |
| L\_0621 | Finding capacitive Reactance | f=100, C= 5 | react= 0.00031830 | react = 0.00031830 | Requirement based |
| L\_0631 | Finding Circular Velocity | r = 100, t = 2 | velocity= 314.159265 | velocity = 314.159265 | Requirement based |
| L\_0641 | Finding Circular Velocity | M1 = 1000, M2 = 1000, D = 2 | Force= 0.000016675 | Force = 0.000016675 | Requirement based |
| L\_0651 | Finding Projectile motion | V = 25, angle=15 | Range = 31.887755 | Range = 31.887755 | Requirement based |
| L\_0711 | Calculating profit and loss | sp = 120, cp=12 | res = 108 | res = 108 | Requirement based |
| L\_0721 | Calculating Discount | mp = 100, d=15 | disc\_price = 85 | disc\_price = 85 | Requirement based |
| L\_0811 | For Matrix Determinant | [10 ,20 ,30; 5, 6, 7; 1, 0, 0] | Determinant=80 | Determinant=80 | Requirement based |
| L\_0821 | Matrix Addition | MAT1= [9 ,30 ,10; -2 ,8 ,10 ;10,-5 ,50 ], MAT2=[10 ,9 ,-5 ;8 ,2 ,0 ;30 ,4 ,10] | ADD\_MAT= [19 ,39 ,5 ;6 ,10 ,10 ;40 ,-1, 60] | ADD\_MAT= [19 ,39 ,5 ;6 ,10 ,10 ;40 ,-1, 60] | Requirement based |
| L\_0831 | Matrix Subtraction | MAT1= [9 ,3 ,10; -2 ,8 ,10 ;10, -5 ,50], MAT2= [1 ,9 ,-5 ;8 ,2 ,0 ;30 ,4 ,10] | SUB\_MAT= [8,-6 ,15 ;-10 ,6 ,10 ;-20 ,-9 , 40] | SUB\_MAT= [8,-6 ,15 ;-10 ,6 ,10 ;-20 ,-9 , 40] | Requirement based |
| L\_0911 | Matrix addition or subtraction | am=2, an=3, bm=2, bn=2 | Error | Error | Scenario based |
| L\_0912 | Matrix multiply | am=1, an=3, bm=2, bn=1 | Error | Error | Scenario based |
| L\_1011 | Calculate simple interest | principal amount=10000, Time-period=2, rate of interest=8 | result=1600 | result=1600 | Requirement based |
| L\_1021 | Calculate compound interest | principal amount=10000, Time-period=2, rate of interest=8 | result=1728 | result=1728 | Requirement based |
| L\_1031 | Calculate EMI amount | principal amount=10000, Time-period=2, rate of interest=8 | result=10855 | result=10855 | Requirement based |
| L\_1041 | Calculate Loan amount | principal amount=10000, Time-period=2, rate of interest=8 | result=10855 | result=10855 | Requirement based |
| L\_1051 | Calculate simple interest | principal amount=10000, Time-period=3, rate of interest=6 | result=1800 | result=2000 | Scenario based |
| L\_1061 | Calculate compound interest | principal amount=30000, Time-period=4, rate of interest=9 | result=12942 | result=14000 | Scenario based |
| L\_1071 | Calculate EMI amount | principal amount=25000, Time-period=2, rate of interest=8 | result=1131 | result=1500 | Scenario based |
| L\_1081 | Calculate Loan amount | principal amount=100000, Time-period=4, rate of interest=7 | result=2395 | result=2000 | Scenario based |
| L\_1111 | Calculate exponent | value=3 | result=20.08 | result=20.08 | Requirement based |
| L\_1121 | Calculate logarithm | value=10 | result=1 | result=1 | Requirement based |
| L\_1131 | Calculate square root | value=144 | result=12 | result=12 | Requirement based |
| L\_1211 | Calculate exponent | value=5 | result=148.41 | result=100 | Scenario based |
| L\_1221 | Calculate logarithm | value=-10 | result=Error | result=1 | Scenario based |

Table 4. Low -Level Testing(SDLC)

## Implementation Summary

Implementation has all the source files, header files, test files for different features of the calculator such as Basic Arithmetic, Combinatorics, Matrix, Profitloss, Trigonometry, financial, special math functions, physics and mensuration.

Here, inc folder holds all header files with “. h” extension which contains prototype of all functions, structure definition, macro definition and definition of all the enumerators.

The src folder holds all source files with “.c” extension which has definition of all functions whose prototype is defined in header file.

The test folder holds the test\_main.c file for cumulative testing of the source codes on based on requirement, scenario and boundary.

Other than these folders outside there is a predefined unity folder which holds prototypes and definition of the standard unity test case functions.

Also, there is Makefile which builds, debugs using valgrind, checks static and dynamic code quality, performs overall unit testing for all the codes together with the execution of single commands based on different defined targets.

There is project\_main.c using which you can run the calculator designed by us in a menu-driven approach.

### Video Summary

It contains a little walkthrough to our GitHub repository. It gives an overview for different blocks of SDLC based V-model. This video has description of what is there in requirements, design, Implementation and TestPlan folder.

### Git Link

<https://github.com/99003733/SDLC_N2_Calculator.git>

### Git Dashboard



#### Badges

Figure 8.Workflow-Badges

### Summary

In this project, we mainly-focused on how to design a calculator which is bit different and cost-effective as compared to other calculators present in the market.

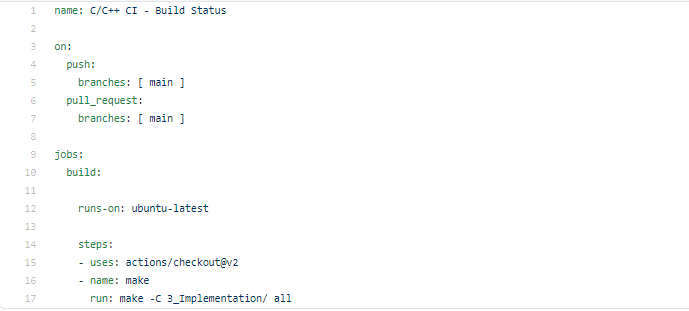
So, for this we first analyzed all the other calculators from low-end feature low cost calculator to high-end feature high-cost calculator and prepared a list of features to include in our modified cost-effective more featured calculator.

Features included in the calculator are basic arithmetic operations (addition, subtraction, multiplication, division), Physics operations (Acceleration, capacitive reactance, circular velocity, Coulomb’s Law, Gravitational force, projectile motion), Matrix Operations (Matrix Addition, Matrix Subtraction, Matrix Multiplication, Matrix row sum , Matrix Column Sum, Matrix Determinant), Combinatorics (Permutations and Combinations), Mensuration (Area and Volumes ), Profit and Loss operations (Profit, Loss, Discount) , Finance operations (Simple Interest, Compound Interest, EMI calculations) , AP,GP sum (AP sum , GP sum) and Special Math Functions (Exponential, Trigonometric, Logarithm).

This calculator is implemented through C programming. This calculator will be mainly used by the school, college students, scientists, businessman, engineers for various purpose. Also, this calculator will be cost effective and if implemented over hardware it cost around 1000 – 1500 INR.

#### Git inspector summary

#### Build



#### Code quality and Issues or Bug Tracking

##### Static Code Quality

##### Dynamic Code Quality

#### Unit Testing

## Individual Contribution & Highlights

#### Individual Contribution

My Contribution to this project in designing features like basic arithmetic functions, combinatorics feature and matrix operation feature. I have implemented all the three features and tested ‘arithmetic and combinatorics using unity.

Also, I have performed the research analysis of different calculators present in market based on features and cost.

I have integrated the code of the whole team in project\_main.c and Calculator.c which is present inside src folder. Also, I have integrated the whole test code.

I have modified Makefile so that program should be able to build as well as it is able to perform unity based unit testing.

#### Highlights

### Summary

This project was our first step to work in coordination as a team while working towards our development of our individual skills.

Also, we’re unorganized at the beginning but after planning through V-Model we’re able to analyze and design each scenario and perfectly implemented whole project in the given duration. So, we also learned a great skill called time-management.

Technical skills developed: -

* Advanced C programming
* Make File
* Unit Testing through Unity
* Version Control using GitHub

Soft-Skills Developed: -

* Team-work
* Team-management
* Time Management
* Assertiveness

### Challenges faced and how were they overcome

1. Issue in running the make file – it was resolved by defining the correct file path in the correct syntax
2. Synchronizing the vs code to GitHub, a colleague helped to resolve the issue
3. Linking source files to header files was an issue that was resolved using MakeFile.

### Future Scope

Test cases for 2D-array based Matrix operations were not tested using unity. So that can be designed also. So more graphical calculations and polynomial based functions can be added as the features in the calculator.

# Miniproject -2 [Team] – Embedded C

## Module/s

Modules linked to this miniproject is Embedded C

### Topic and Subtopics

Driver Designing

GPIO Programming

I2C, SPI, UART and ADC Protocols

Hardware Abstraction Layer (HAL) Application interface

## Objectives & Requirements

Objective:

The main objective of the project is design a Body Control Module (BCM) with some features using STM32F407VG microcontroller with 32-bit ARM cortex-M4 FPU core processor.

A control module is a computer component in a car that monitors, controls and uses electrical devices throughout the vehicle. When electric devices began to be used in cars and trucks, each tool was controlled by a different electrical component. There was a cooling module, an interior lighting module, a door lock module, and so on. The body control module combines all these different modules under a single system to work together instead of partitioning, a simple format for both production and problem solving. While there are many different types of body control modules, they are generally the same, the sensors for connecting, switching and automatic switching together in a single computer system. These components are subdivided into inputs, such as sensor data about temperature or speed, and the effects, or the way a computer control system responds to control motor performance. Inputs and outputs are further divided into analog and digital information types - analog signals used by modular continuous modules (such as oil pressure) and digital signals used for modules that can either turn on or off (such as headlights or oil indicator light).

Requirement:

In our project we have implemented six features on a microcontroller STM32f4 discovery board. In the project we integrated IR Sensor, thermal sensor, Gas sensor (mq7), LDR sensor, Moisture sensor,  ignition sensor with microcontroller STM32f4 using GPIO protocol.

* IR Sensor:-An IR sensor is an electronic device, which illuminates light to detect the surrounding environment. The IR sensor can detect movement. Usually, in the infrared spectrum, all substances emit a certain type of radiation. These types of radiation are invisible to our eyes, but the infrared sensor can detect these rays.
* The gas sensor receives the attention of the liquor gas in the air and the analog voltage reading output. The sensor can operate at temperatures ranging from -10 to 50 ° C with an electrical power of less than 150 Ma to 5V. Sensitivity range ranges from 0.04 mg / L to 4 mg / L, suitable for breathalyzers.
* LDR or light-resistant light resistor is also known as photo resistor, photocell, photoconductor. One type of opposition its resistance varies depending on the amount of light falling on its surface. When light falls on an opponent, the resistance changes.
* Moisture sensors measure the amount of water in the soil. Since direct gravimetric measurement of free soil moisture requires removal, drying, and weight of the sample, soil moisture sensors measure water content indirectly using other earth materials, such as electricity resistance, dielectric constant, or neutron contact, as representative of moisture content.
* The ignition system produces sparks or burns the electrode at high temperatures to heat a mixture of gasoline gas in combustion engines fire engines, oil-fired boilers and gas, rocket engines, etc.
* Thermal sensors detect a change in a physical parameter such as resistance or output voltage that corresponds to a temperature change.

High Level Requirements:

Table : HLR of hardware sensor project

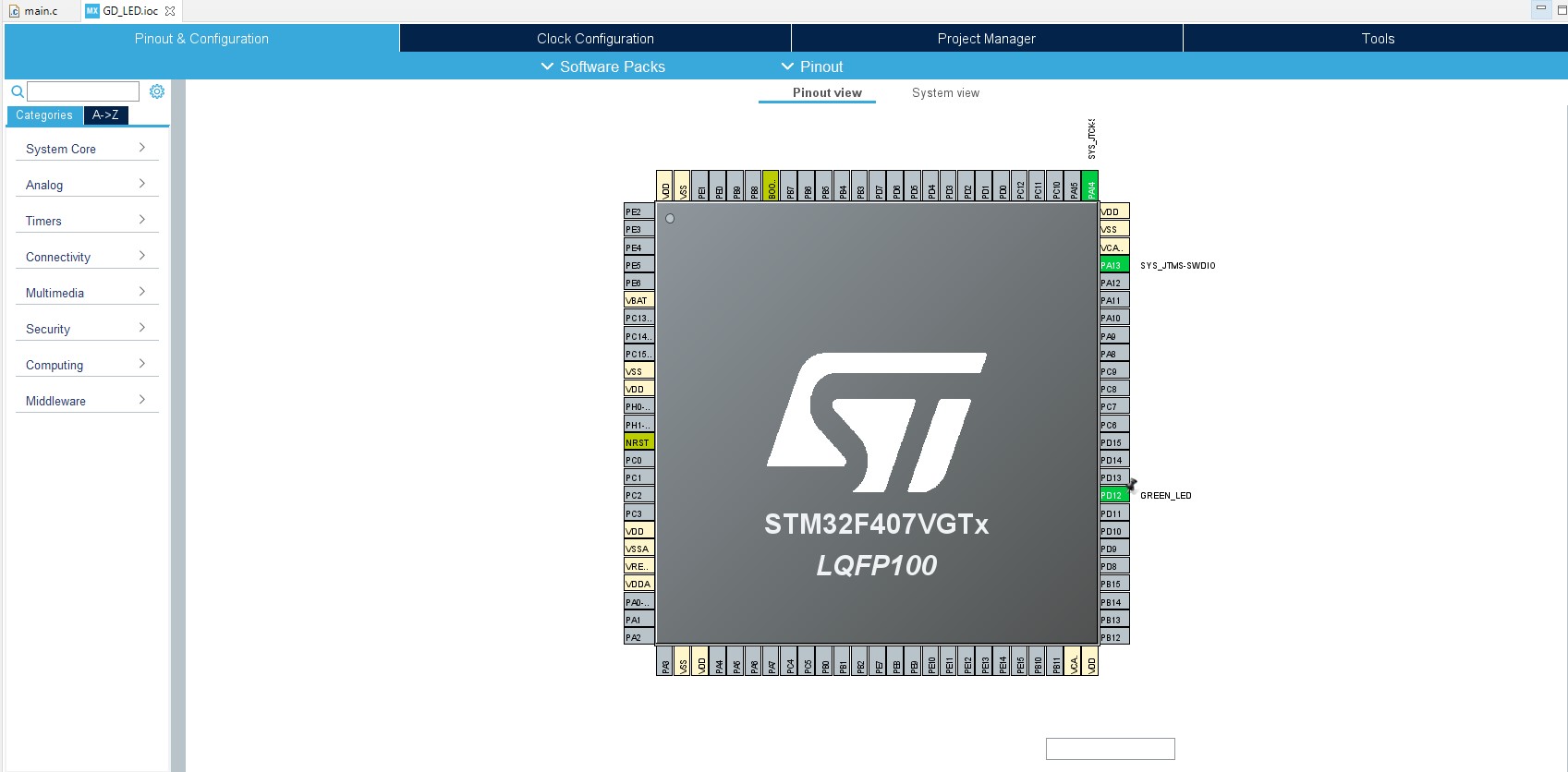
|  |  |  |  |
| --- | --- | --- | --- |
| ID | Requirements | Description | Status |
| HLR\_01 | Enable control Module | Control Module should be enabled in order to detect different sensors. | Implemented |
| HLR\_02 | Temperature sensor | Able to detect temperature | Implemented |
| HLR\_03 | Gas Sensor | Able to detect gas leakage | Implemented |
| HLR\_04 | LDR (light dependent resistor) Sensor | Able to detect higher intensity of light | Implemented |
| HLR\_05 | Moisture Sensor | Able to detect moisture | Implemented |
| HLR\_06 | IR Sensor | Able to detect obstacles | Implemented |

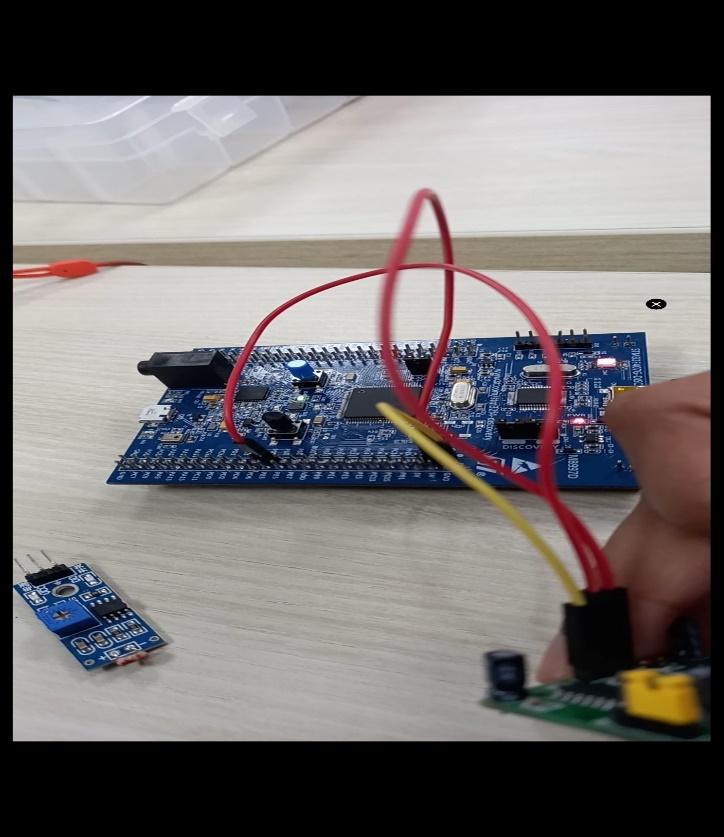
Low Level Requirements:

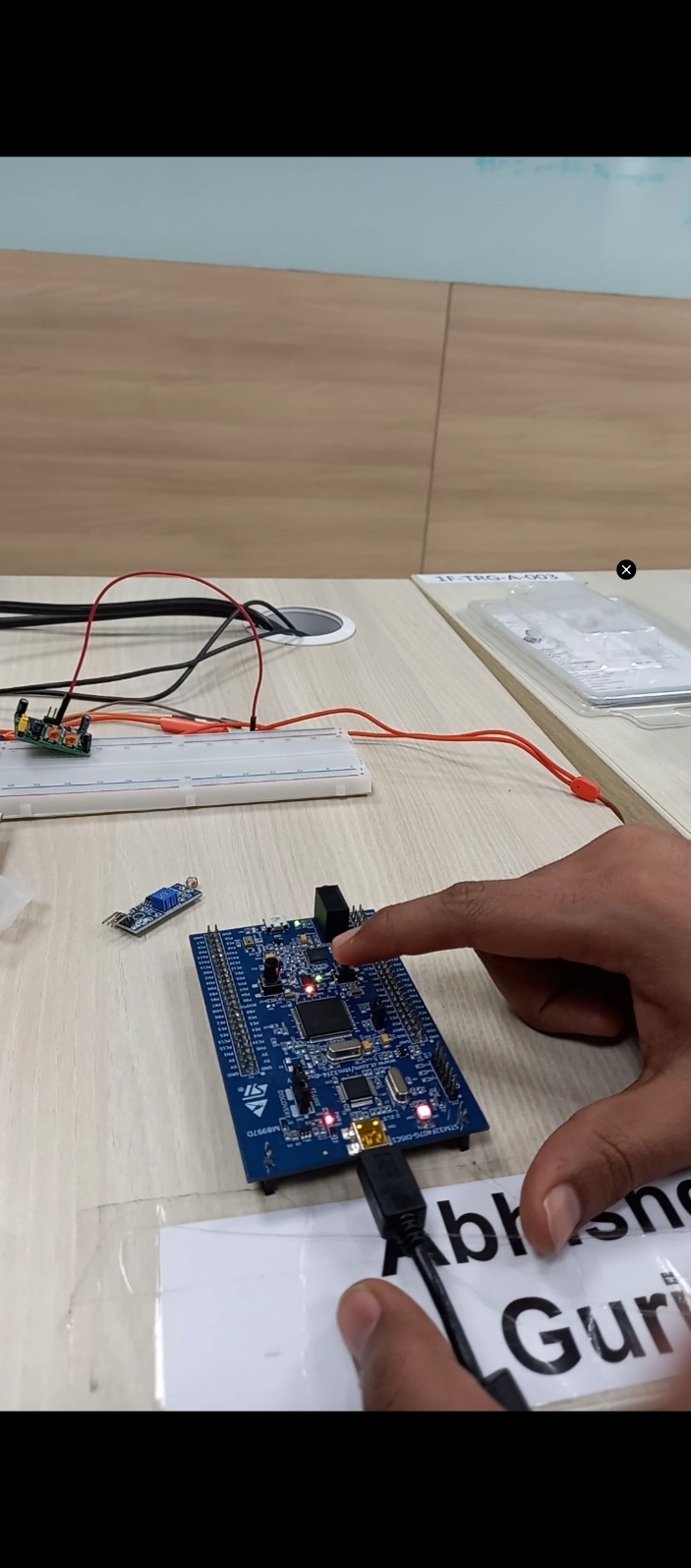
Table : LLR of hardware sensor project

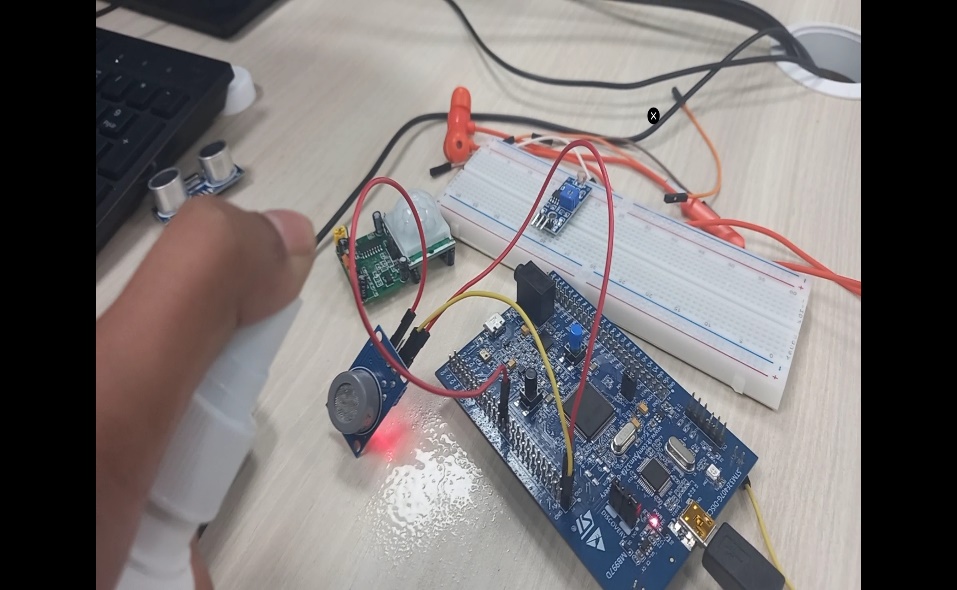
|  |  |  |  |
| --- | --- | --- | --- |
| ID | Requirements | Description | Status |
| LLR\_01 | Implementation of air Conditioning Control module | Thermal sensor detects the high temperature and AC switches ON. | Implemented |
| LLR\_02 | Implementation of gas leakage control module | Gas sensor detects gas leakage and glows LED which prevents spontaneous car explosion. | Implemented |
| LLR\_03 | Implementation of shed controller module | LDR sensor enables the intensity of sunlight and the shed is enabled when the sunlight is too bright | Implemented |
| LLR\_04 | Implementation of Headlight control module | LDR sensor senses darkness and headlight turns ON. | Implemented |
| LLR\_05 | Implementation of Wiper control module | Moisture sensor senses water in wiper and LED switches ON | Implemented |
| LLR\_06 | Implementation of obstacle detector control module | LED glows when some obstacle is detected | Implemented |

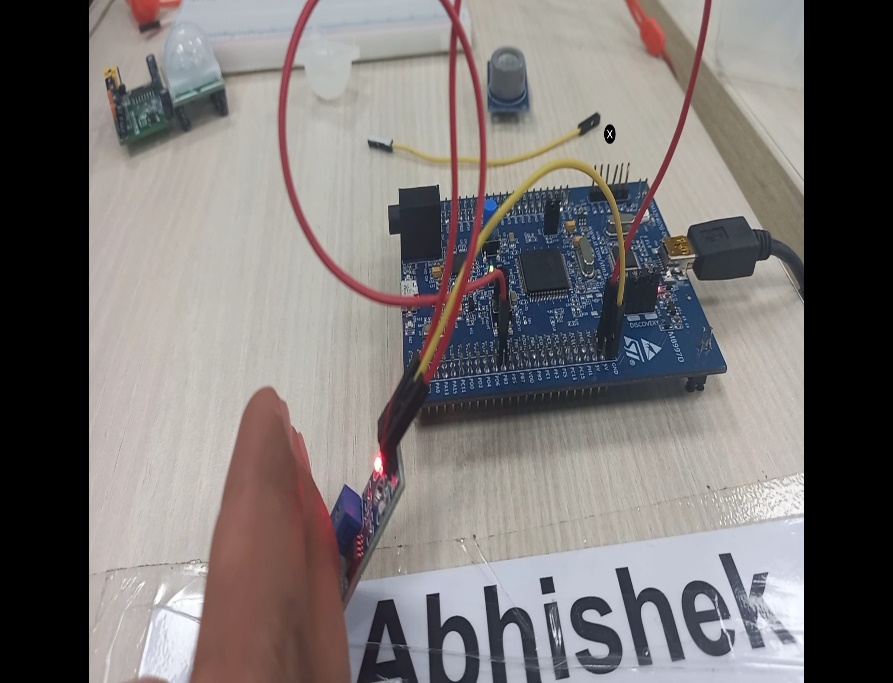
## Design

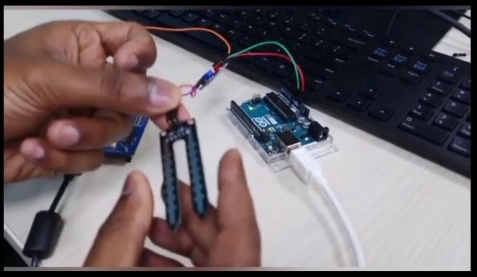


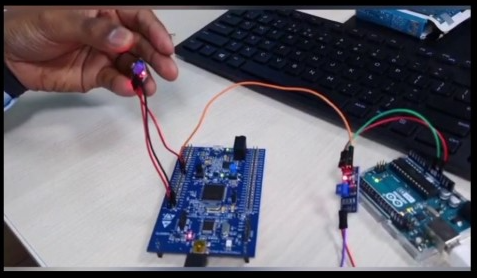












## Test Plan

High Level Plan

Table : High Level Test Plan of Hardware Project

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ID | Description | Expected Input | Expected Output | Actual Output |
| HLP\_01 | Enable control Module should be enabled in order to detect different sensors. | Module is turned on | Module gets enabled | Module enabled |
| HLP\_02 | Temperature sensor should be able to detect temperature | High Temperature is provided | LED glows | LED glows indicating AC switches ON. |
| HLP\_03 | Gas Sensor should be able to detect gas leakage | Excess gas leakage | LED glows and system shut down | LED glows and car shut down |
| HLP\_04 | LDR Sensor should be able to detect higher intensity of light | Excess/low sunlight/light | Sun shed enabled or headlight switches ON | Sun shed enabled or headlight switched ON |
| HLP\_05 | Moisture sensor should be able to detect moisture | Water droplets | Moisture sensor senses water droplets and wiper starts moving | Moisture sensor senses water droplets and wiper starts moving |
| HLP\_06 | IR sensor should be able to detect obstacles | Sensor occupied by any obstacle | Red LED blinks | Red LED blinks |

Low Level Plan:

Table : Low Level Test Plan of Hardware project

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ID | Description | Expected Input | Expected Output | Actual Output |
| LLP\_01 | Thermal sensor detects the high temperature and AC switches ON. | High Temperature provided at the input of thermal sensor | AC switches ON | AC Switched ON |
| LLP\_02 | Gas sensor detects gas leakage and glows LED which prevents spontaneous car explosion. | Hand sanitizer containing alcohol | LED is switched ON and Buzzer starts beeping. | LED is ON |
| LLP\_03 | LDR sensor senses the intensity of sunlight and the shed is enabled when the sunlight is too bright | Torch Light (at small scale) or sunlight | Red LED turns ON | Red LED turns ON indicating Shed has been open |
| LLP\_04 | LDR sensor senses darkness and headlight turns ON. | Push button is pressed indicating seat shifts front or back | Green LED Glows. | Green LED Glows indicating headlight is ON |
| LLP\_05 | Moisture sensor senses water in wiper and LED switches ON | Water droplets | LED turned ON | LED turned ON indicating wiper is active |
| LLP\_06 | LED glows when some obstacle is detected | IR sensor detects obstacle | LED glows | Green LED Glows. |

## Implementation Summary

In this project, six functions of a Body Control Module has been implemented on a STM microcontroller using different sensors and Arduino board.

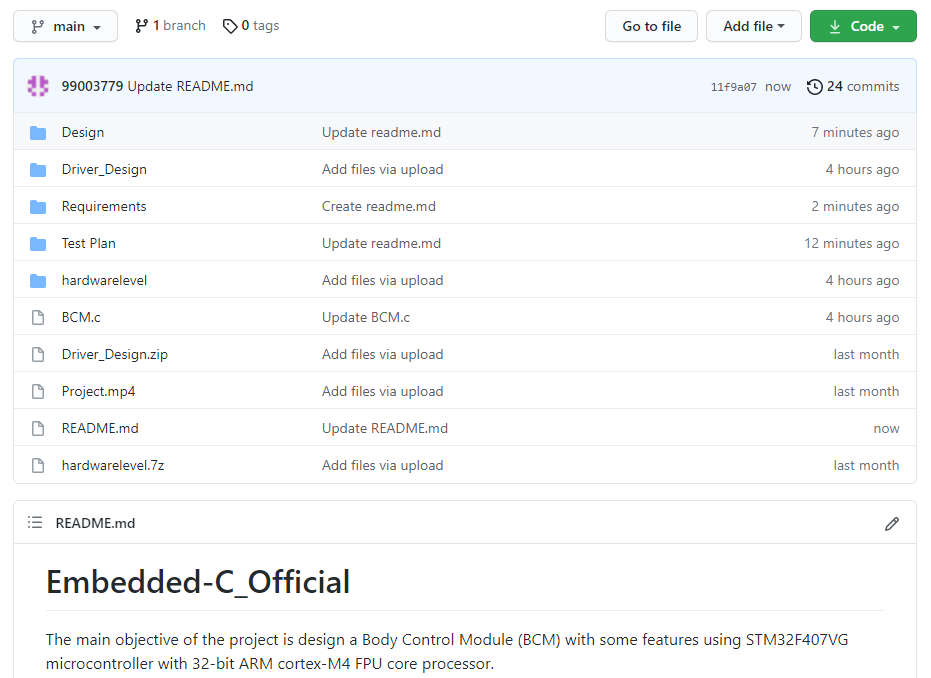
Following six features has been implemented:

1. **Air Conditioner Control Module:** In this module, temperature sensor has been used. Thermal sensor detects the high temperature and AC switches ON. Thermal sensors detect a change in a physical parameter such as resistance or output voltage that corresponds to a temperature change.
2. **Gas Leakage control Module/ Car shut down control Module:** In this module, Gas sensor has been used. Gas sensor detects gas leakage and glows LED which shuts down the car and prevents spontaneous car explosion. The gas sensor receives the attention of the liquor gas in the air and the analog voltage reading output.
3. **Sun Shed Control Module:** In this module, LDR sensor has been used. LDR sensor senses the intensity of sunlight and the shed is enabled when the sunlight is too bright. When light falls on it, the resistance changes and hence it works.
4. **Headlight Control Module:** In this module, again LDR sensor has been used. LDR sensor senses darkness and headlight turns ON.
5. **Wiper Control Module:** In this module, moisture sensor has been used. Moisture sensor senses water in wiper and LED switches ON indicating that wiper is working. Moisture sensor measures water content indirectly with the help of electricity resistance, dielectric constant, or neutron contact, as representative of moisture content.
6. **Obstacle Detector Control Module:** In this module, IR sensor has been used. LED glows when some obstacle is detected by the sensor. An IR sensor is an electronic device, which illuminates light to detect the surrounding environment. The IR sensor can detect movement.

### Git Link

<https://github.com/99003779/Embedded-C_Official.git>

### Git Dashboard



### Summary

In this project, we have implemented a body control module showing six features on a STM32 microcontroller board having a core processor of 32-bit ARM cortex-M4 FPU. Following six features has been implemented:

* Air Conditioner Control Module
* Gas Leakage Control Module/ Car shut down control Module
* Sun Shed Control Module
* Headlight Control Module
* Wiper Control Module
* Obstacle Detector Control Module

## Individual Contribution & Highlights:

Features implemented by me in the team hardware project are:

* **Air Conditioner Control Module:** Thermal sensor detects the high temperature and AC switches ON.
* **Gas Leakage control Module/ Car shut down control Module:** The gas sensor receives the attention of the liquor gas in the air and the analog voltage reading output.

### Challenges faced and how were they overcome

* Integration of all the features on a single STM32 microcontroller board.
* Hardware implementation on STM32 IDE platform.

### Future Scope (If applicable):

NA

# Miniproject -3 [Individual]

## Module

“Advance Python Programming”

### Topic and Subtopics

#### Core-Topics

* Basic Python

##### Sub-Topics

* Data Types
* Arithmetic operations
* String operations
* Control structures

##### Sub-Topics

* If-else statements
* While loops
* For loops
* Nested Loops
* Functions

##### Sub-Topics

* Defining custom functions
* Pass by value
* Pass by reference
* Introduction to Library functions
* Data Structures

##### Sub-Topics

* List
* Tuple
* Set
* Dictionary
* Exceptional Handling

##### Sub-Topics

* Try, Except
* Finally
* pass keyword
* Date and time Library

##### Sub-Topics

* Date comparison
* Execution time analysis
* Excel file library (openpyxl)

##### Sub-Topics

* Open multiple excel files
* Reading multiple excel files sheet by sheet.
* Mastersheet creation
* Writing to excel file

All these Core-topics and sub-topics are implemented through V-Model .

## Objectives & Requirements

#### Objective

To analyze data from multiple-excel spreadsheets of multiple Excel files stored over different directory and writing only the specific data on the Mastersheet or Master-workbook using Python Programming.

#### Requirements

##### High Level Requirements

|  |  |  |  |
| --- | --- | --- | --- |
| **ID** | **Requirements** | **Description** | **Status** |
| HLR01 | Multiple Excel Workbook | Multiple Workbook should be read over different directory | Implemented |
| HLR02 | Reading Worksheet | Reading multiple worksheet from multiple workbook | Implemented |
| HLR03 | Searching & Writing specific data | Searching in existing sheets and writing to Mastersheet or Master-workbook | Implemented |

Table 9.High-Level Requirements (Python)

##### Low-Level Requirements

|  |  |  |  |
| --- | --- | --- | --- |
| **ID** | **Requirements** | **Description** | **Status** |
| HLR01 - LR1 | Taking Multiple workbooks | Accessing multiple workbooks | Implemented |
| HLR01 - LR2 | Finding Path of file | Locating various Excel files based on filename from anywhere on the system and loading them | Implemented |
| HLR01 – LR3 | Excel file format | Excel file should be of .xlsx format | Implemented |
| HLR02 – LR1 | Data Collection | Data of share market for different shares collected in data different sheets of different workbook | Implemented |
| HLR02 – LR2 | Data size | Minimum number of rows=40, Minimum number of Columns=10 | Implemented |
| HLR03 – LR1 | Reading Data | Reading all multiple sheets from multiple workbook | Implemented |
| HLR03 - LR2 | Searching Data | Search for specific data based on user specific input | Implementing |
| HLR03 - LR3 | Writing only particular data | Writing data based on the searched data to Mastersheet or Master-workbook | Implemented |

Table 10.Low-Level Requirements (Python)

### Design

#### System Level Design

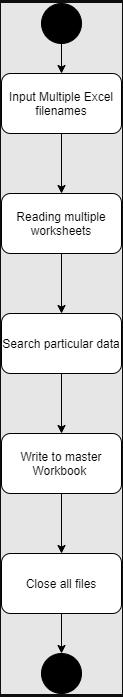


Figure 13.High-Level Design (Activity Diagram) Python

#### Unit Level Testing

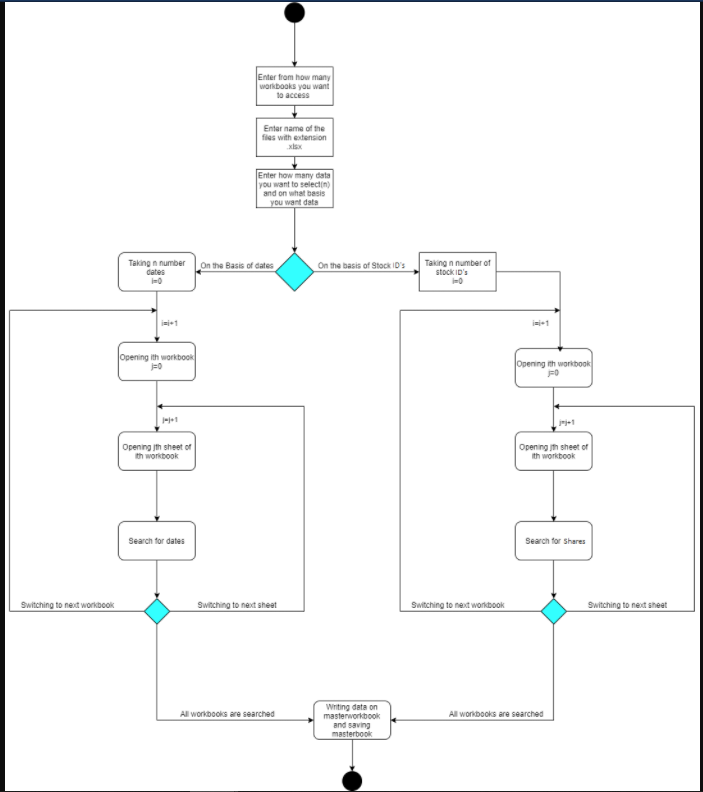


Figure 14. Low-Level Design (Activity Diagram) Python

### Test Plan

#### High level Testing

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Test ID** | **Description** | **Expected Input** | **Expected Output** | **Actual Out** | **Type of Test** |
| HLR\_01 | To Access multiple workbooks stored in different locations | Choice:2  Number of worksheets=2, C:\Users\99003733\Desktop\apps\New folder2\share\_axis.xlsx  C:\New folder1\share\_asian.xlsx | ‘<Workbook 1>’  ‘<Workbook 2>’ | ‘<Workbook1>’  ‘<Workbook2>’ | Requirement based |
| HLR\_02 | To Read multiple worksheets of a workbook | Choice:2  Number of worksheet =1,  Enter file path: D:\New folder\share\_adani.xlsx | ‘<ADANIPORTS>,  <Sheet 1>’ | ‘<ADANIPORTS>,  <Sheet 1>’ | Requirement based |
| HLR\_03 | To Search by date | Choice:1 (to take excel files path from file)  Choice1 :1 (To search by date)  Number of dates:2  Date :2/2/200  Date: 3/1/2000 | All the Data of 2/2/2000 and 3/1/2000 copied to master-workbook | All the data of 2/2/2000 and 3/1/2000 copied to master-workbook | Requirement based |

Table 11.High-Level Testing (Python)

#### Low Level Testing

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Test ID** | **Description** | **Expected Input** | **Expected Output** | **Actual Out** | **Type of Test** |
| HLR01-LR01 | To access multiple worksheets by provide their path one by one | Choice: 2  Number of worksheets = 3  Enter file path: C:\Users\99003733\Desktop\apps\New folder2\share\_axis.xlsx  Enter File path:  C:\New folder1\share\_asian.xlsx  Enter File path:  D:\New folder\share\_adani.xlsx   |  | | --- | |  | | ‘<Workbook 1>’  ‘<Workbook 2>’  ‘<Workbook 3>’ | ‘<Workbook1>’  ‘<Workbook2>’  ‘<Workbook 3>’ | Requirement based |
| HLR01-LR02 | To access multiple worksheets whose paths are present in text file | Choice:1  File.txt | Starts reading every worksheet | Starts reading every worksheet | Scenario based |
| HLR02-LR01 | Searching data by only dates | Choice:1 (to take excel files path from file)  Choice1 :1 To search by date  Number of dates:2   1. 29/2/200 2. 31/1/2000 | All the Data of 29/2/2000 and 31/1/2000 copied to master-workbook | All the data of 29/2/2000 and 31/1/2000 copied to master-workbook | Requirement based |
| HLR02-LR02 | Search Data by only ShareID | Choice:1 (to take excel files path from file)  Choice1 :2 To search by ShareID  Number of shares:2   1. HD855 2. AP575 | All the Data related to ShareID’s HD855, AP575 is copied to master-workbook | All the Data related to ShareID’s HD855, AP575 is copied to master-workbook | Requirement based |
| HLR02-LR03 | Search data by both date and ShareID | Choice:1 (to take excel files path from file)  Choice1 :3 To search by both ShareID  Number of dates: 2   1. 4/2/2000 2. 17/1/2000   Number of shares:1   1. HD855 | All the data on 4/2/2000,  17/1/2000 for  ShareID HD855 is copied to master-workbook | All the data on 4/2/2000,  17/1/2000 for  ShareID HD855 is copied to master-workbook | Requirement based |
| HLR03-LR01 | To Skip date which is not present in the sheets and search for all the other data | Choice:1 (to take excel files path from file)  Choice1 :1 To search by date  Number of dates:2   1. 29/2/200 2. 26/1/2000 (Not a valid date) | Only data of 29/2/2000 copied to master-workbook | Only data of 29/2/2000 copied to master-workbook | Scenario based |
| HLR03-LR02 | To Skip ShareID which is not present in the sheets and search for all the other data | Choice:1 (to take excel files path from file)  Choice1 :2 To search by ShareID  Number of shares:2   1. AX700 (Invalid ShareID) 2. AP575 | Only data of AP575 copied to master-workbook | Only data of AP575 copied to master-workbook | Scenario based |
| HLR02-LR03 | To Skip date/ShareID which is not present in the sheets and search for all the other data | Choice:1 (to take excel files path from file)  Choice1 :3 To search by both ShareID  Number of dates: 2   1. 4/2/2000 2. 17/1/2000   Number of shares:1   1. AB878 | Empty Master-workbook | Empty Master-workbook | Scenario based |

Table 12.Low-Level Testing (Python)

## Implementation Summary

Here I have programmed in such a way that the code is able to read various excel files stored in different locations and extract only particular data from the excel files based on date and ShareID and display it on a separate excel file called masterbook.

All the source files are present in /3\_Implementation/src folder

Here I have divide the code into multiple files where the modules related for multiple input’s and their validation in the given excel files are taken care. Also, there is separate module file to access excel files from various location which has two ways to access the excel file first by accessing file where path of excel file is already stored and secondly by accessing the file where user provide their paths one by one.

Also, exceptions are used to avoid traceback at certain places and situations.

Follow these steps to execute the code: -

* To Run the code run->main.py file (present in Implementation/src folder)
* Enter -1 If you want to open workbooks present in file"

(for path through file you have all share datasets in the folder named data inside src folder copy those .xlsx file various directory and copy the path in file.txt)

* Enter -2 If you want to enter location and open workbooks

(Here also copy all data files from data folder in src.)

Then enter 3 cases by which means you want search or filter data

* Enter -1 To search data by Dates
* Enter -2 To search data by ShareID
* Enter -3 To search data by both Dates and ShareID
* Then enter number of dates or shares you want and then enter dates or shares one by one
* At the end, all data will be printed on masterbook.xlsx file present in src folder

Predefined python module libraries used: -

1. openpyxl
2. time

### Video Summary

It contains a little walkthrough to my GitHub repository. It gives an overview for different blocks of SDLC based V-model. This video has description of what is there in requirements, design, Implementation and TestPlan folder. Also, it contains a bit more description of python files in implementation folder.

### Git Link

<https://github.com/99003733/Python_Mini_Project.git>

### Git Dashboard

### Summary

In this project, I have worked on data of 5 different shares stored in 5 different excel files which stored in different directories in all over the PC. The share data in excel file is from Jan-2000 to Feb-2000.

Here, first every workbook is opened as soon as their paths were provided (5 different pointer stores base address of every workbook). Then when we enter the criteria through which we want to segregate the data and populate it over the masterbook.

After this code searches for the data, sheet by sheet and data related to that criteria (date, ShareID) in all the sheets gets populated to the masterbook.

Also, the data here is searches in each cell of the excel sheet and copy the data cell by cell to the masterbook

### Challenges faced and how were they overcome

* Accessing excel file from different location and automating it for multiple sheets – This issue was overcome by using nested loops and using openpyxl module load\_workbook method.
* Searching common data by two unique data’s simultaneously - This was again solved by analyzing flow of multiple loops using exceptions at right place.

### Future Scope

* Searching in huge amount of data needed to be analyzed.
* Graphs of the extracted data can be analyzed for better analyses.
* Code can be more optimized and automatic generation of Mastersheet or masterbook can be added.
* This code as an API or GUI can be used in analysis of data in share market by plotting scatter plots and histograms.

# Miniproject -4 [Individual-Activity]

## Module

“Embedded Linux and Kernel , Device Drivers”

### Topic and Subtopics

#### Core-Topics

* System Calls

##### Sub-Topics

* C programming
* Custom Kernel Image Generation
* Cross-Compilation (using Qemu)
* IPC (Inter- Process Communication)

##### Sub-Topics

* C programming
* Process
* Race Condition
* Semaphores
* Mutex
* Spinlock
* Wait Queue
* Threads
* Custom Modules

##### Sub-Topics

* Kfifo API
* List API
* Module development
* IOCTL

## Objectives & Requirements

#### Objective

* System calls -- echo back the given string
* System calls -- Traverse process list, print pid and ppid
* System calls -- Retrieve attributes of calling process
* IOCTL - echo the back the string, implement two operations,

string passed by one operation will be retrieved by other operation

* To implement IOCTL code as follows

One operation receives an integer, create a list node and append

Other operation traverses the list

Exit method will clean-up the nodes

* To implement three kernel threads

First two threads will write N chars each into kfifo

Third thread wait for first two threads and retreive data from kfifo

Apply mutual exclusion between first two threads

* To implement three kernel threads

First two threads will write N nodes into a kernel list

Third thread wait for first two threads and traverse the list

Apply mutual exclusion between first two threads

exit method will clean-up the list nodes

#### Requirements

|  |  |  |  |
| --- | --- | --- | --- |
| **ID** | **Requirements** | **Description** | **Status** |
| HLR01 | System call to echo string | The system call echo backs the given string. | Implemented |
| HLR02 | System to traverse process list and print pid and ppid | System call traverses through process list and print pid and ppid of all the running process. | Implemented |
| HLR03 | System to traverse process list and get attributes of the calling process | System call traverses through process list and retrieves attributes of calling process. | Implemented |
| HLR04 | IOCTL to echo back string using two separate operation (Read & write) | Using IOCTL to echo back the string, one operation will read the string and other operation writes back the string | Implemented |
| HLR05 | IOCTL operations to create list, traverse through list and cleanup nodes. | Using IOCTL create list of N nodes and traversing through list. In exit method clean up all the nodes. | Implementing |
| HLR06 | Thread Implementation on Kfifo and avoiding race by mutual exclusion. | 2 threads will write N bytes of data on Kfifo and then only 3rd thread should be able to read data through kfifo. Apply mutual exclusion between two writing threads | Implemented |
| HLR07 | Thread implementation on list, traversing through it and applying mutual exclusion. | 2 threads will write N nodes into list then only 3rd thread should be able to traverse the list. Apply mutual exclusion between two writing threads. In exit method clean-up all list nodes. | Implementing |

Table 13.Requirements (Kernel driver development)

## 

## Test Plan

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Test ID** | **Description** | **Expected Input** | **Expected Output** | **Actual Out** | **Type of Test** |
| HLR\_01 | Given string echo’s back | “Hello welcome to kernel” passed as a command line argument | “Hello welcome to kernel” (on serial console (Qemu)) | “Hello welcome to kernel” | Requirement based |
| HLR\_02 | PID and PPID of all the running process is to be printed | Invoking executable file of user-space code on serial console | PID, PPID of all the running process. (on terminal) | PID, PPID of all the running process. | Requirement based |
| HLR\_03 | Attributes such as state, priority, PID, PPID of calling process is to be printed | Invoking executable file of user-space code on serial console | PID, PPID, State, Priority of the calling process. (on terminal) | PID, PPID, State, Priority of the calling process. | Requirement based |
| HLR\_04 | Using IOCTL to echo back the string, one operation will read the string and other operation writes back the string. | Writing string use echo command | String printed on the serial monitor (Qemu) using cat operation | String printed on the serial monitor (Qemu) using cat operation | Requirement based |
| HLR\_05 | Using IOCTL create list of N nodes and traversing through list. In exit method clean up all the nodes. | Invoking executable file of user-space code on serial console | String printed on the serial monitor (Qemu) using cat operation | String printed on the serial monitor (Qemu) using cat operation | Requirement based |
| HLR\_06 | 2 threads will write N bytes of data on Kfifo and then only 3rd thread should be able to read data through kfifo. Apply mutual exclusion between two writing threads | Writing string use echo command two times | String printed on the serial monitor (Qemu) using cat operation | String printed on the serial monitor (Qemu) using cat operation | Requirement based |
| HLR\_07 | 2 threads will write N nodes into list then only 3rd thread should be able to traverse the list. Apply mutual exclusion between two writing threads. In exit method clean-up, all list nodes. | Invoking executable file of user-space code on serial console | String printed on the serial monitor (Qemu) using cat operation | String printed on the serial monitor (Qemu) using cat operation | Requirement based |

Table 14.Test-Plan (Kernel Driver Development)

## Implementation Summary

For System Calls: -

* Generate new zImage after adding system calls definition and prototype. Adding its definition file name in Makefile.

-> make ARCH=arm CROSS\_COMPILE=arm-linux-gnueabi- zImage

* Then mount the SD card and copy the output file of user space code in it and then unmount it by following commands: -

-> sudo mount –o loop, rw, sync rootfs.img /mnt/rootfs

-> sudo cp a.out /mnt/rootfs/home/root

->sudo umount rootfs.img /mnt/rootfs

* Then run Qemu using following command

-> qemu-system-arm -M vexpress-a9 -m 1024 -serial stdio \

-kernel zImage -dtb vexpress-v2p-ca9.dtb \

-sd rootfs.img -append "console=ttyAMA0 root=/dev/mmcblk0 rw"

* Then run the ./a.out file on Qemu and output is display either on VGA console or in serial console based on the system call. (give command line input only for first system call)
* System call displays the required output.

For IOCTL:-

Echo back string

* Two different functions are used one for writing to the device and other one reading from the device.
* Here input is string through as echo command run on the serial console and cat command is used to read from the file.
* Here IOCTL is used for reading and writing from the file (using IOCTL macros).

IOCTL list

* Two different functions are used here one for writing to the list nodes and other function is to traverse through the list.
* At last in exit operation use list\_for\_each\_safe () to clean-up all the nodes.

For Kthreads:-

* Here at first I have created three kernel threads so first two threads will write to kfifo/list and the third thread can only write when first two thread has performed their operation.
* To implement this scenario I have used wait queue which will wait for first two writer threads then only wakes-up the third thread.
* Now there is a chance of race condition between first two threads so I have used mutex for mutual exclusion in both the threads. So by addition mutual exclusion race condition is avoided.
* For list operation in exit module list\_for\_each\_safe () is used to clean-up all the nodes.

### Video Summary

It contains a little walkthrough to my GitHub repository. It contains the description of what kinds of files stored in activity folder and little a bit information about how they are implemented.

### Git Link

<https://github.com/99003733/Linux_Kernel_Drivers.git>

### Summary

In this project custom system calls for a particular kernel is made by modifying internal syscalls.h , syscall.tbl , kernel /Makefile and its definition in c file in kernel folder of kernel source.

In user-space code of the system call a special system call number is mentioned to use the custom system call which is defined system call table (syscall.tbl). Finally it’s test on serial console and VGA console according to expected input and output.

In ioctl and kernel threads the code is written as a module driver in kernel space and invoked in user space using file operations such as read & writing from file/kfifo/list.

### Challenges faced and how were they overcome

* Unable to directly access string in kernel space from userspace and vice-versa – Using copy\_from\_user () and copy\_to\_user () solved this issue.
* Traversing through system process list was an issue- It was solved by using for\_each\_process () and task\_struct.
* Traversing through node list was issue that was resolved using list\_for\_each () method.
* Synchronizing two competing threads issue was solved by using flags in two different threads and checking them for condition in wait queue.

# Miniproject -5 [Individual]: Networking

## Module/s

Modules linked to this miniproject is Data Communication and Networking

### Topic and Subtopics

IPv4 and IPv6 addresses

Routers

Routing Protocols

Networking

Packet Tracer Tool

## Objectives & Requirements

Objective:

The objective of this project is to design a network consisting of various servers connecting two or more sub networks and is able to send data packets throughout the network. Cisco Packet tracer is the tool that has been used in network designing

Requirements:

Cisco Packet Tracer: It is a strong network simulation program or the software that allows many esearchers, scholars and students across the globe to experiment with the behavior of network. It supplements physical equipments so that students will be able to create networks immense number of devices. Following is a small network demonstrating many networks interconnected by routers and switch.

Each server should be able to send packet to every other server. Whether the server is in the same network or different network.

## Design

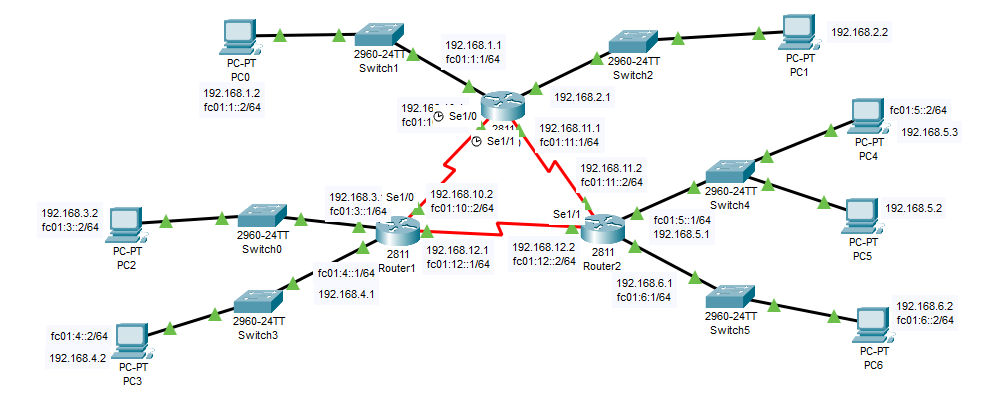


Figure 22: Serial Connection of routers for packet transmission from one server to the other

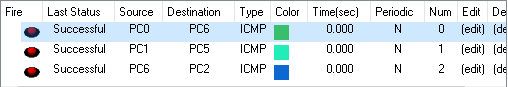


Figure 23:Transfer of Packets from different sources to different destination

## Test Plans:

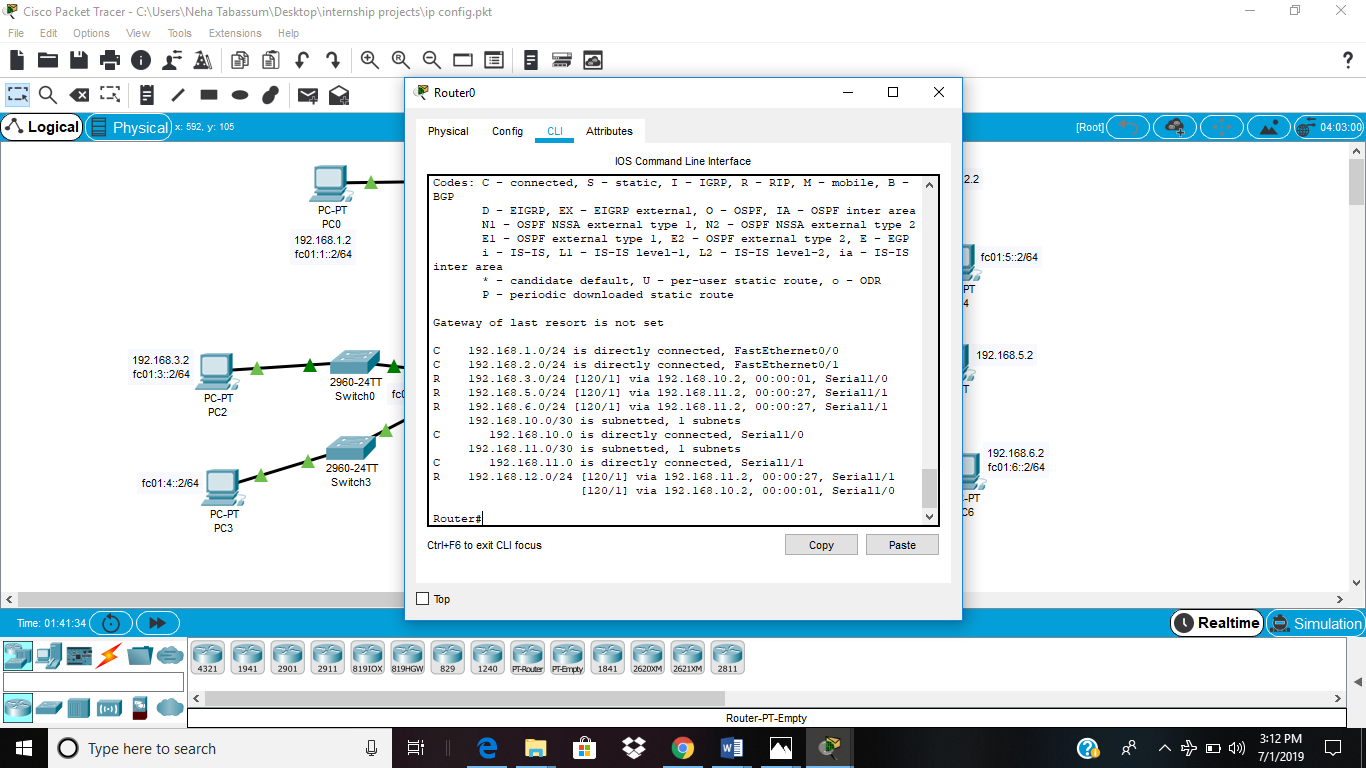


Figure 24: IPv4 routes on router 0 (show ip route)

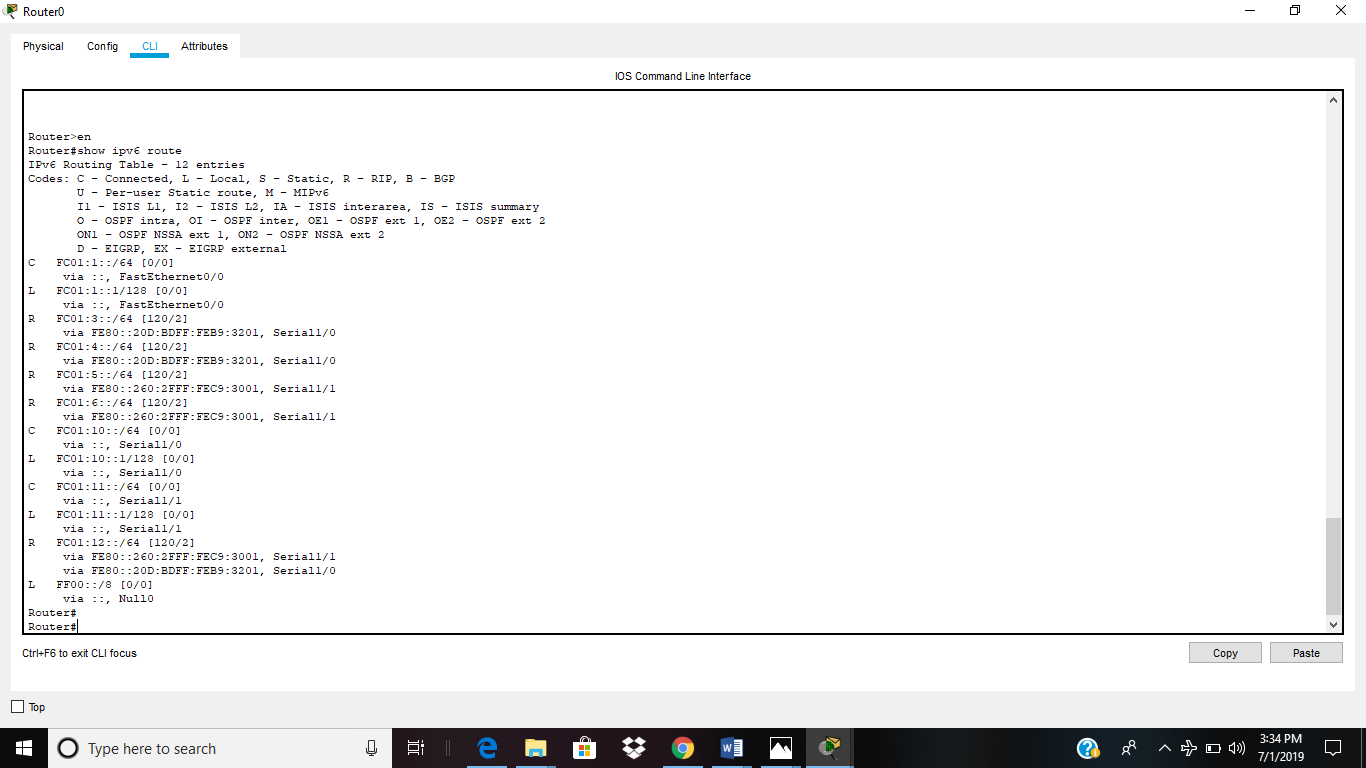


Figure 25: IPv6 routes on router 0 (show ipv6 route)

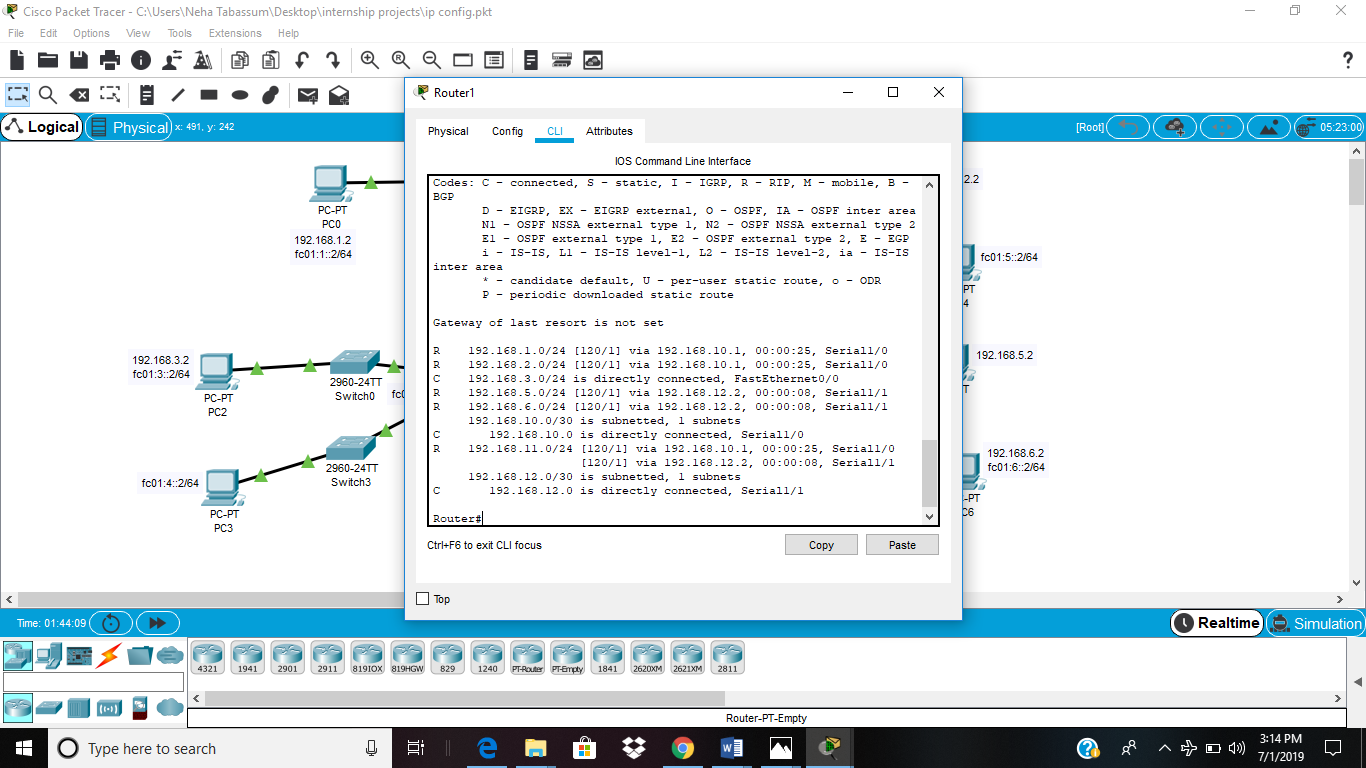


Figure 26: IPv4 routes on router 1 (show ip route)

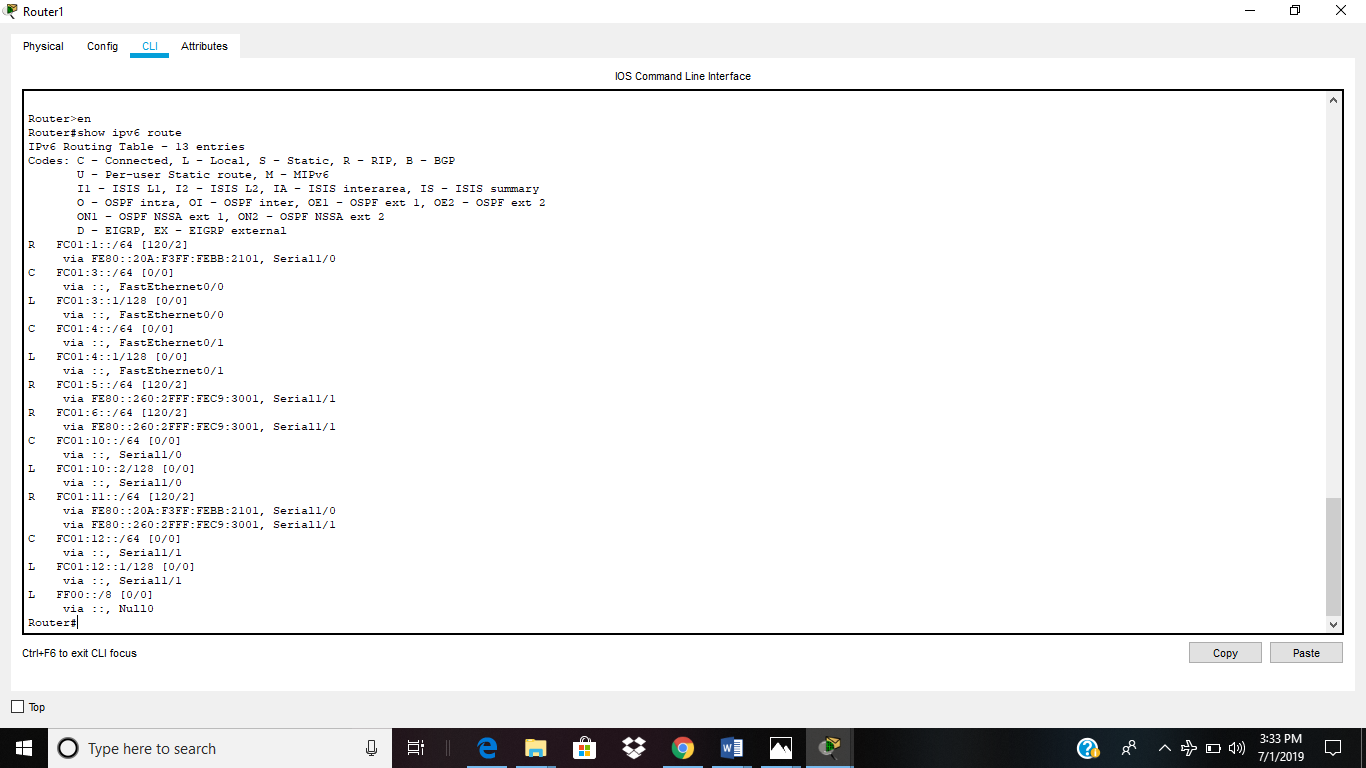


Figure 27: IPv6 routes on router 1 (show ipv6 route)

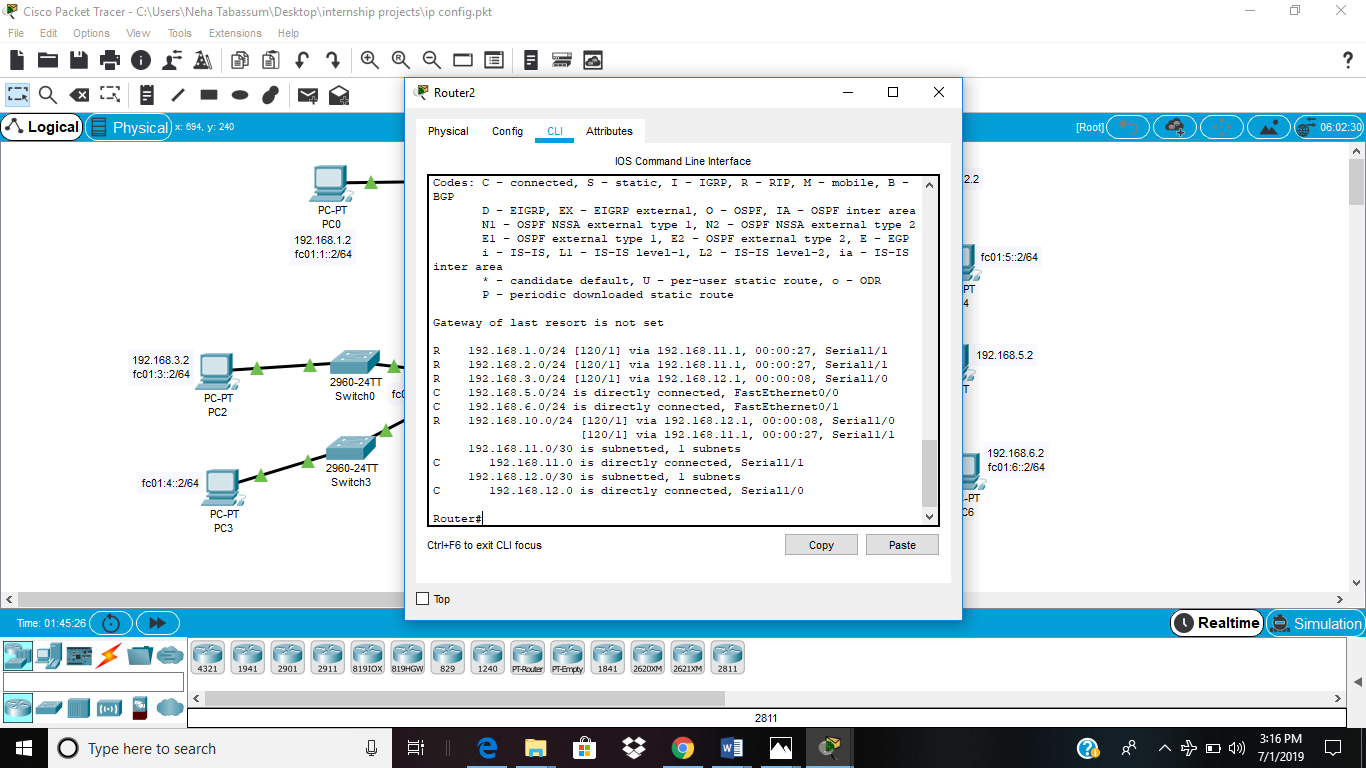


Figure 28: IPv4 routes on router 2 (show ip route)

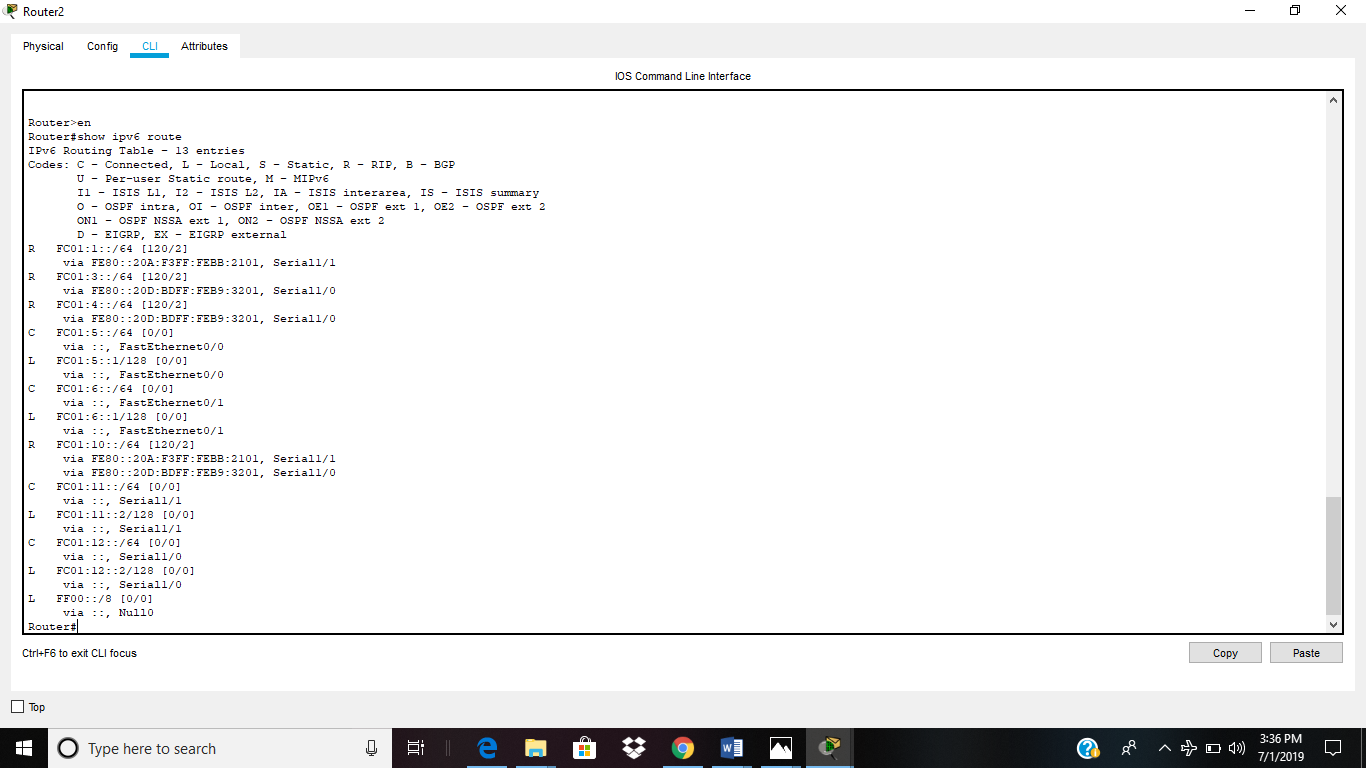
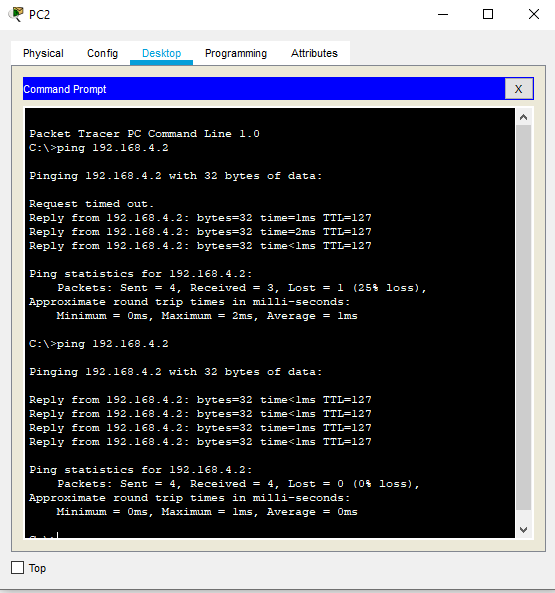
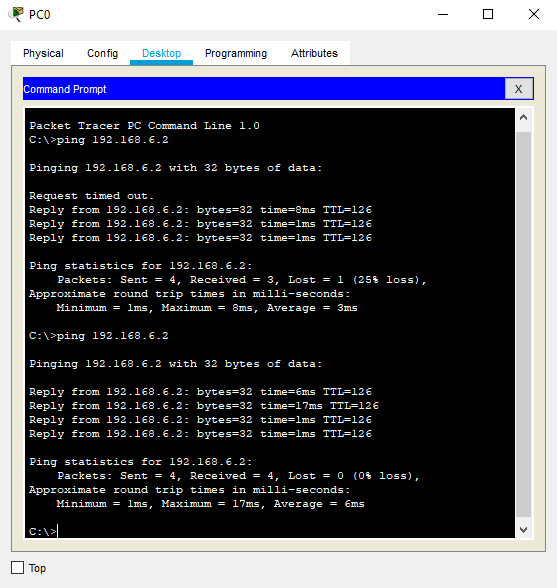


Figure 29: IPv6 routes on router 2 (show ipv6 route)

## Testing Results:

To test whether a network is working properly or not, a ping command is used. Ping is a tool or a network utility program which allows you to check weather a particular host is reachable or not. Loop back address is another IP address which used to check working of the self server. Example of loop back ip address is 127.0.0.1 and it will always return a reply unless a network security system prevents it. Eg firewall.





## Implementation Summary

The implemented design consist of three networks interconnected by router which follows RIP routing protocol. Each Data packets from every server is able to send data to every other server.

## Individual Contribution & Highlights:

The whole project was the individual project.

### Challenges faced and how were they overcome

* Interconnection of different networks using routers but using routing protocols it was interconnected
* Implementation of routing protocols but by learning and reading different routing protocols it was implemented
* Sending of data packets from one network to another but with proper integration data packets were able to send to different networks

### Future Scope (If applicable):

NA