./

GENESIS - Learning Outcome & Mini-project Summary Report



|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Ver. Rel. No.** | **Release Date** | **Prepared. By** | **Reviewed By** | **To be Approved** | **Remarks/Revision Details** |
| 1 | 15-04-2021 | Arikatla Pujitha/99003787 |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |

**Details**

Contents

[Contents 3](#_Toc69890818)

[Mini project -1 [Team/Individual] 5](#_Toc69890819)

[1.Module- 5](#_Toc69890820)

[2.Topic and Subtopics 5](#_Toc69890821)

[3.Objectives & Requirements 5](#_Toc69890822)

[Low Level Requirements 9](#_Toc69890823)

[Design: 10](#_Toc69890824)

[LLR\_UML Diagram 11](#_Toc69890825)

[Test Plan 13](#_Toc69890826)

[Implementation Summary 18](#_Toc69890827)

[Video Summary 19](#_Toc69890828)

[Git Link 19](#_Toc69890829)

[Git Dashboard 19](#_Toc69890830)

[Summary 19](#_Toc69890831)

[Individual Contribution & Highlights 27](#_Toc69890832)

[Summary 28](#_Toc69890833)

[Challenges faced and how were they overcome 28](#_Toc69890834)

[Future Scope 28](#_Toc69890835)

[Mini project -2 Python [Team/Individual] 30](#_Toc69890836)

[1.Modules used: 30](#_Toc69890837)

[Module used in this project is Python and SDLC. 30](#_Toc69890838)

[2.Topic and Subtopics 30](#_Toc69890839)

[3.Objectives & Requirements 30](#_Toc69890840)

[Features that are integrated in this project are 31](#_Toc69890841)

[4W's and 1'H 31](#_Toc69890842)

[Why: 31](#_Toc69890843)

[What: 31](#_Toc69890844)

[When: 31](#_Toc69890845)

[Where: 31](#_Toc69890846)

[How: 32](#_Toc69890847)

[SWOT Analysis 32](#_Toc69890848)

[4.Design: 34](#_Toc69890849)

[Implementation Summary 39](#_Toc69890850)

[Git Link 39](#_Toc69890851)

[Git Dashboard 39](#_Toc69890852)

[Git Summary 40](#_Toc69890853)

[Summary 40](#_Toc69890854)

[Challenges faced and how were they overcome 41](#_Toc69890855)

[ Implementing the library 41](#_Toc69890856)

[ Writing data in master sheet 41](#_Toc69890857)

[Mini Project -3 EMBEDDED C => [Team] 42](#_Toc69890858)

[Modules Used: 42](#_Toc69890859)

[Modules used in this project are Embedded Systems and Embedded C Programming and was implemented on the hardware STM32. 42](#_Toc69890860)

[Objectives & Requirements: 42](#_Toc69890861)

[Summary 50](#_Toc69890862)

[Mini project -4 [Individual] 51](#_Toc69890863)

[Module 51](#_Toc69890864)

[Topic and Subtopics 51](#_Toc69890865)

[Objectives & Requirements 52](#_Toc69890866)

[Test Plan 52](#_Toc69890867)

[Implementation Summary 53](#_Toc69890868)

[Git Link 54](#_Toc69890869)

[Git Dashboard 54](#_Toc69890870)

[Summary 54](#_Toc69890871)

[Challenges faced and how were they overcome 55](#_Toc69890872)

# Mini project -1 [Team/Individual]

## **1.Module-**

The module used in this are SDLC, C Programming.

### 2.Topic and Subtopics

* The core steps of SDLC is being implemented.
* The features of calculator are implemented.
* The testing has been done for each function.
* Introduction about SDLC
* C Programming
* Code Analysis
* Cpp check
* Val grind
* Testing
* Unit testing
* Make file
* V model
* Agile model
* Git hub

## **3.Objectives & Requirements**

**FEATURES:**

1. The calculator will solve all basic arithmetic operations. (addition, subtraction, multiplication and division)

# The calculator will solve all the exponential, logarithmic operations, power of a number, and factorial of a number.

# The calculator is going to solve all trigonometric and inverse trigonometric operations.

# The calculator is going to have a clear function which will clear the calculator display.

# The calculator will have a liquid crystal display i.e. LCD Display.

**SWOT ANALYSIS:**

**STRENGTHS:**

1. Our product is very cost efficient.
2. It is very user friendly since it has limited functions.
3. It will work on solar power and has 1.5v battery as a battery backup and has an automatic power off.

**WEAKNESS:**

1. It has limited operations. People trying to do calculations regarding complex numbers and other things regarding statistics will not be able to do so.
2. The battery inside a solar calculator is mostly alkaline button cell which cannot be recharged.

**OPPORTUNITIES:**

1. The price of the product is less than other products with same features that are available in the market.
2. The product will do very well in shop counters as the calculator is very user friendly and anybody can use it without prior knowledge of that calculator.
3. Students till class 10 will be attracted to this product because of its simplicity in operations and design.
4. The product will also work in banking sectors and other government sectors where they want low price, minimum features, handy products.

**THREATS**

1. Students who are pursuing higher education won’t be interested in this product as it has limited operations.
2. Several other competitors are there who are willing to make the product with more features.
3. One of the threats in the product is its battery which cannot be recharged.

**4W & 1H:**

**WHAT:**

 It is a simple electronic hardware device or software that are capable of performing the simple calculations such as addition, subtraction, multiplication, division, calculating power of number, exponential function, logarithmic function, permutation and combination, trigonometry, inverse-trigonometric functions, factorial of a number, binary to decimal conversion etc.

**WHEN:**

1. Useful during exams, for getting complex calculation in very less time.
2. Calculation of bills in malls, shops, and restaurants.

**WHERE:**

1. Exam hall
2. Shop Counters
3. Colleges and schools.
4. Banking sectors

**WHY:**

1. Complex calculations are very tough to calculate in less time, so this calculator is used for simple operations.
2. We get the result quickly and accurately.
3. We save our valuable time by using the product.
4. Saves human power.

**HOW:**

1. Input will be given by the user from the keyboard and the result will be displayed to the screen.
2. Write the code for all the requirements.
3. Use one programming language to code the functions(C).
4. Use GitHub and visual studio for making and building file for required specification.
5. Check all functionalities.

**High level Requirements-**

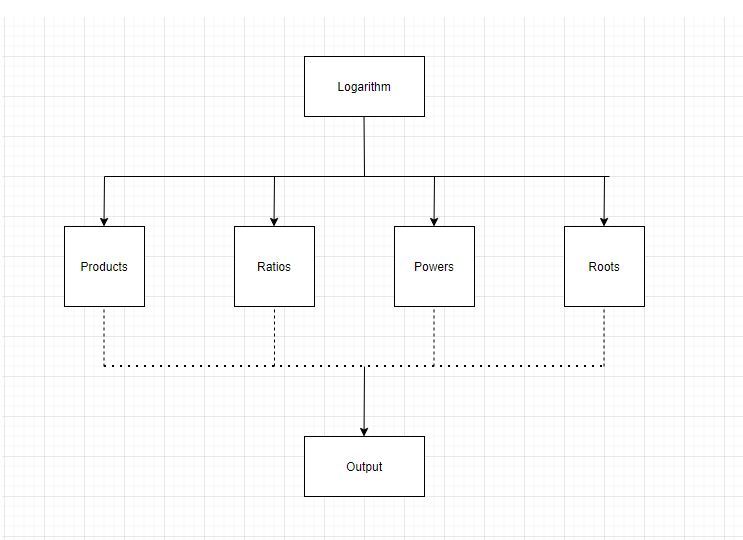
1. Maximum input digits a user can enter is up to 12 digits.
2. Arithmetic Operations
3. Exponential and Logarithmic Operations.
4. Square roots and powers.
5. Permutation and Combinations.
6. Finding area of different geometrical shapes like circle, square, rectangle, triangle.
7. Finding volume of different geometrical shapes like cube, cylinder, cone, sphere.
8. Trignometric And Inverse Trignometric Functions.
9. Measurements conversion (km, cm, inch, etc.)
10. weight conversion (kilogram, liter etc.)
11. Mixed to improper fractions.
12. Cube and cube root.
13. Temperature conversion.
14. Arithmetic operations of fractions. (+,-,\*,/) .
15. Calculation of Reminder.
16. Calculation of simple interest.
17. Binary to decimal conversion.

## **Low Level Requirements**

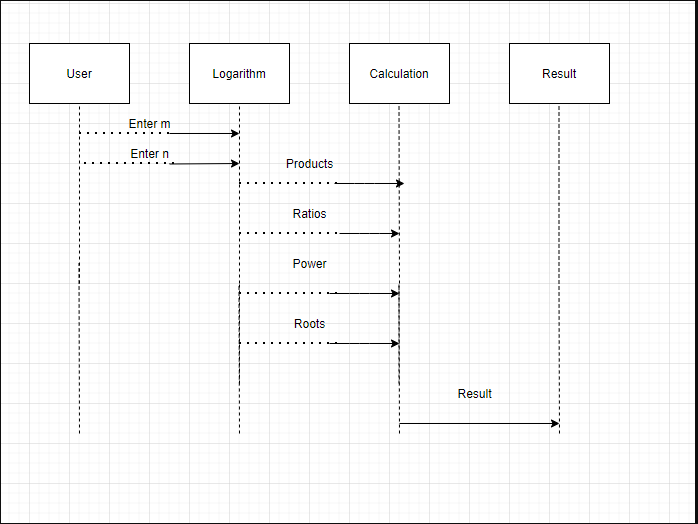
1. Exponential operation (result=b^x where b=base and x=exponent. Input type: integer and float) Logarithmic operation (result=log(x) where the input type of x is integer or float)
2. Area of different geometrical shapes (The input data type used in this scenario are integer, float and long).
3. Volume of different geometrical shapes (The input data type used in the scenario are integer and float.)
4. Permutation and Combination (result=nPr; input type: integer and result=nCr; input type: integer)
5. using float for Decimal values.
6. using int for Integer values.
7. using Double.
8. calculation of reminder using modulus operator.
9. using arithmetic operators for calculation of simple interest.

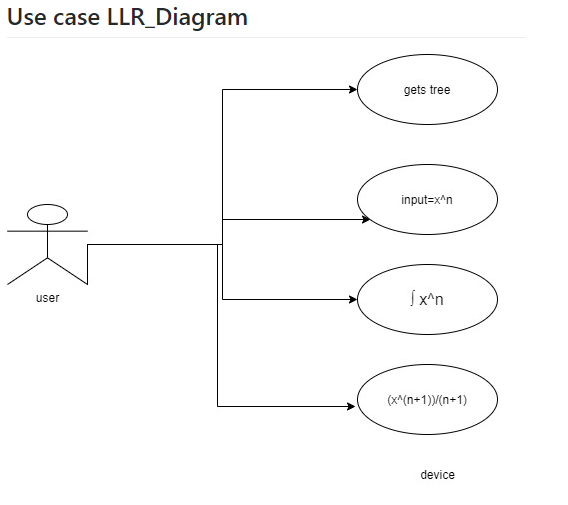
## **Design:**

**HLR\_UML Diagram**



## **LLR\_UML Diagram**





## **Test Plan**

**High level test plan:** -

|  |  |  |
| --- | --- | --- |
| Test ID | Description | Type of Test |
| H\_01 | To perform area calculation of circle | Requirement based |
| H\_02 | To perform area calculation of rectangle | Requirement based |
| H\_03 | To perform area calculation of square | Requirement based |
| H\_04 | To perform area calculation of triangle | Requirement based |
| H\_05 | To perform volume calculation of cube | Requirement based |
| H\_06 | To perform volume calculation of cylinder | Requirement based |
| H\_07 | To perform volume calculation of cone | Requirement based |
| H\_08 | To perform volume calculation of sphere | Requirement based |

|  |  |  |
| --- | --- | --- |
| Test ID | Description | Type of Test |
| L\_01 | To enter a binary number and convert to decimal------------- | Requirement based |
| L\_02 | To enter a binary number and convert to decimal------------- | Requirement based |
| L\_03 | To enter a binary number and convert to decimal------------- | Scenario based |
| L\_04 | To enter a binary number and convert to decimal------------- | Scenario based |
| L\_05 | To enter a binary number and convert to decimal------------- | Scenario based |

|  |  |  |
| --- | --- | --- |
| Test ID | Description | Type of Test |
| H\_01 | To Perform logarithmic operations | Requirement based |
| H\_02 | To Perform exponential operations | Requirement based |
| H\_03 | To Perform Square root operations | Requirement based |
| H\_04 | To Perform Cube root operations | Requirement based |

|  |  |  |
| --- | --- | --- |
| ****Test ID**** | ****Description**** | ****Type of Test**** |
| H\_01 | To perform Basic Arithmetic  Operations | Requirement based |
| H\_02 | To perform Trigonometric functions | Requirements based |

**Low level test plan:** -

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Test ID | Description | Exp IN | Exp OUT | Actual Out | Type of Test |
| L\_01 | To find the area of circle | radius=5 | 78.53 | 78.53 | Requirement based |
| L\_02 | To find the area of rectangle | length=5  breadth=6 | 30 | 30 | Requirement based |
| L\_03 | To find the area of square | length=12 | 144 | 144 | Requirement based |
| L\_04 | To find the area of triangle | base=4  height=3 | 6 | 6 | Requirement based |
| L\_05 | To find the volume of cube | radius=5 | 125 | 125 | Requirement based |
| L\_06 | To find the volume of cylinder | radius=6  height=5 | 565.56 | 565.56 | Requirement based |
| L\_07 | To find the volume of cone | radius=6  height=5 | 186.63 | 186.63 | Requirement based |
| L\_08 | To find the volume of sphere | radius=4 | 267.64 | 267.64 | Requirement based |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Test ID | Description | Exp IN | Exp OUT | Actual Out | Type of Test |
| L\_01 | To enter a binary number and convert to decimal------------- | 1000 | 8 | 8 | Requirement based |
| L\_02 | To enter a binary number and convert to decimal------------- | 1111 | 15 | 15 | Requirement based |
| L\_03 | To enter a binary number and convert to decimal------------- | 152 | Invalid Input | Invalid Input | Scenario based |
| L\_04 | To enter a binary number and convert to decimal------------- | -101 | Invalid Input | Invalid Input | Scenario based |
| L\_05 | To enter a binary number and convert to decimal------------- | -85 | Invalid Input | Invalid Input | Scenario based |

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Test ID | | Description | | EXPIN | | EXPOUT | Actual Out | Type of Test |
| L\_01 | | To Find the logarithmic operations | | 1.product = log(mxn) = log(m)+log(n)  2. Ratio=log(m/n) = log(m)-log(n) m=5 n=6 | | product =16 ratio=300 | product =16 ratio=300 | Requirement based |
| L\_02 | | To Find the exponential operations | | base=2 exp =3 | | 8 | 8 | Requirement based |
| L\_03 | | To Find the Square root operations | | 16 | | 4 | 4 | Requirement based |
| L\_04 | | To Find the cube root operations | | 125 | | 5 | 5 | -Requirement based |
| **Test ID** | | **Description** | | **Type of Test** | | |
| L\_01 | | To perform addition | | Requirement based | | |
| L\_02 | | To perform subtraction | | Requirement based | | |
| L\_3 | | To perform multiplication | | Requirement based | | |
| L04 | | To perform division | | Requirement based | | |
| L\_05 | | To perform sin function | | Requirement based | | |
| L\_06 | | To perform cos function | | Requirement based | | |
| L\_07 | | To perform tan function | | Requirement based | | |
| L\_08 | | To perform sin inverse function | | Requirement based | | |
| L\_09 | | To perform cos inverse function | | Requirement based | | |
| L\_10 | | To perform tan inverse function | | Requirement based | | |

## **Implementation Summary**

# Implementation folder had all source files, header files, test files for different features of the calculator such as Basic Arithmetic, Square root, cube root, exponent, logarithm, etc.

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.

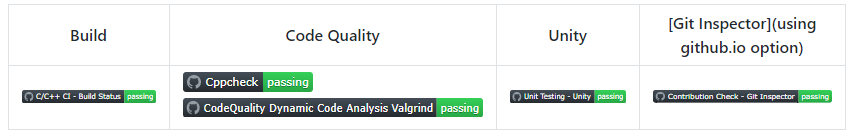
Other than these folders outside there is a predefined unity folder which holds prototypes and definition of the standard unity test case functions.

# Also, there is a **Make file** which builds, debugs using valgrind, check 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.

Mandatory: To add the GitHub user - **stepin654321** as a contributor to the repo”

**BADGES**

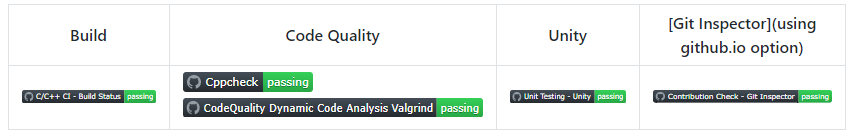
****

### Video Summary

### Git Link

<https://github.com/99003784/N9_SDLC_CALCULATOR.git>

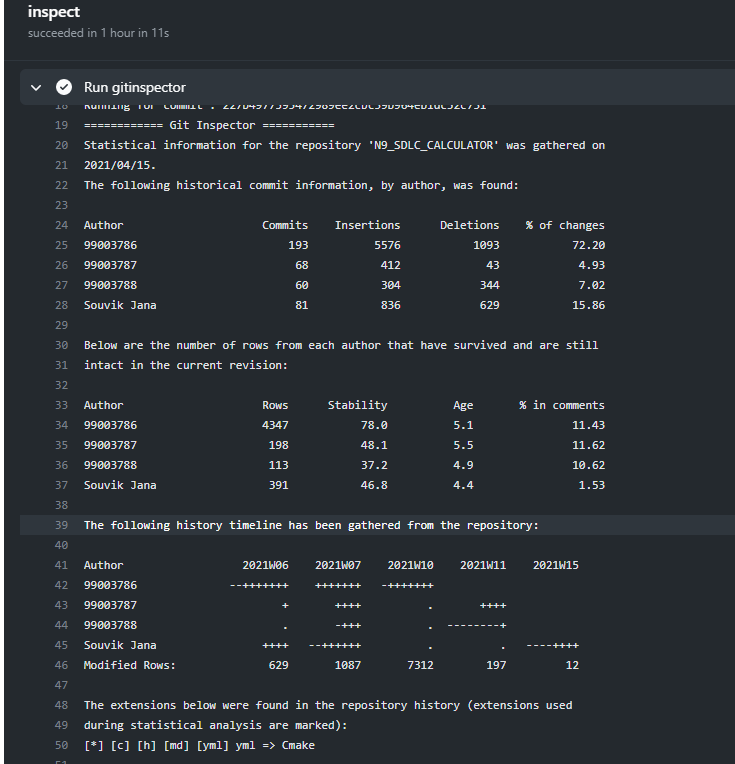
### Git Dashboard

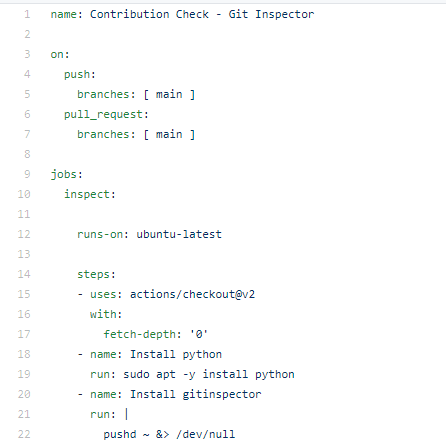
****

### Summary

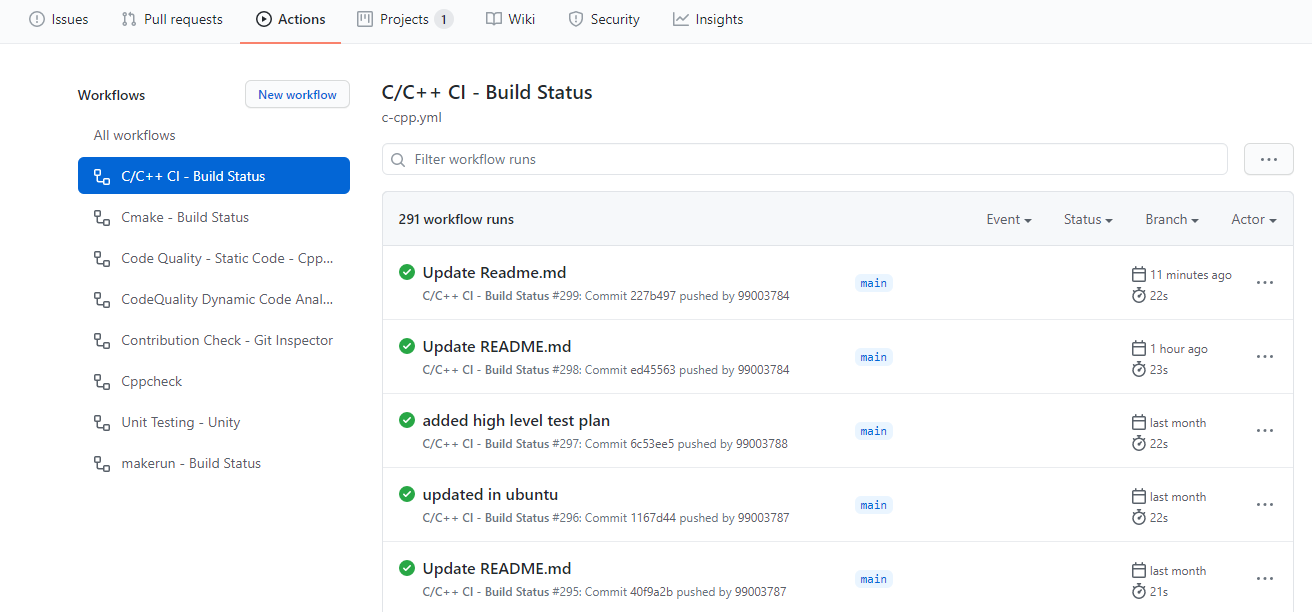
* In this project, we mainly-focused on how to design a calculator which is bit different and cost-effective as compared to other calculators present in the market.
* So, for this we first analyzed all the other calculators from low-end feature low cost calculator to high-end feature high-cost calculator 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.

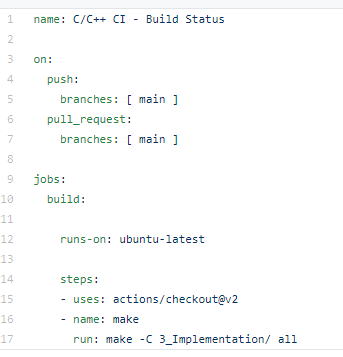
#### Git inspector summary





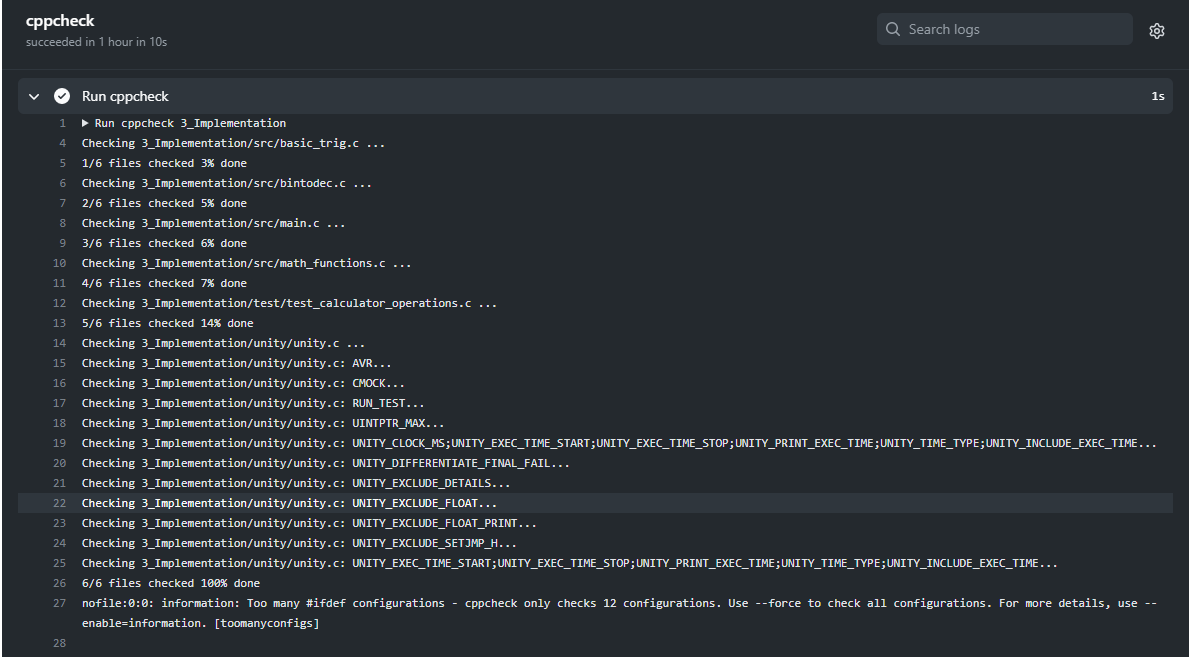
#### Build





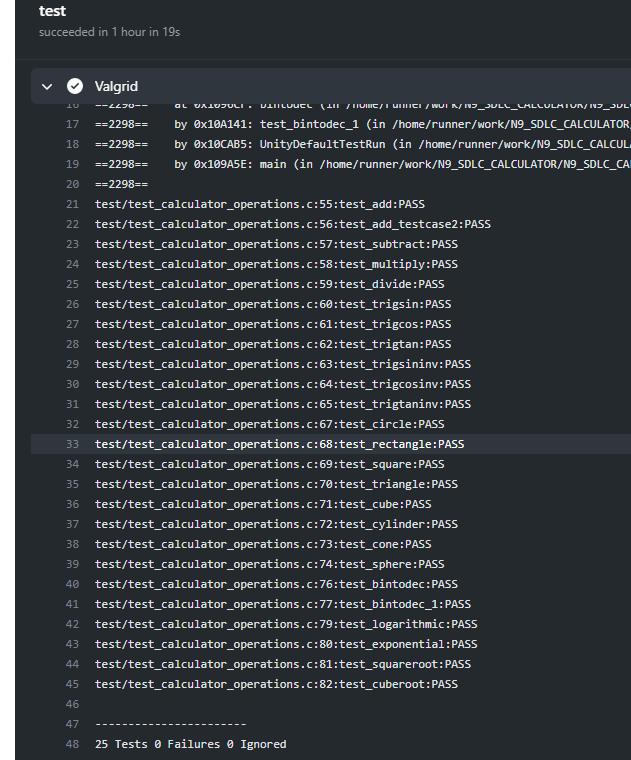
#### Code quality and Issues or Bug Tracking

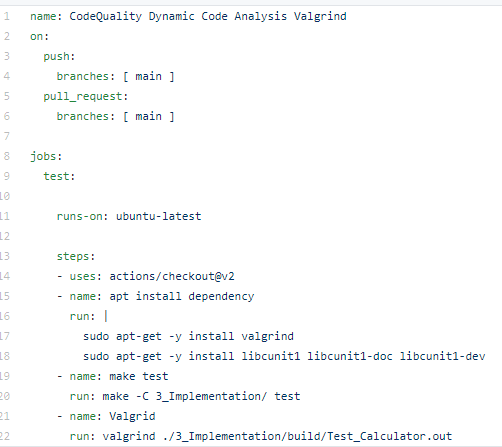
Static Code Quality:



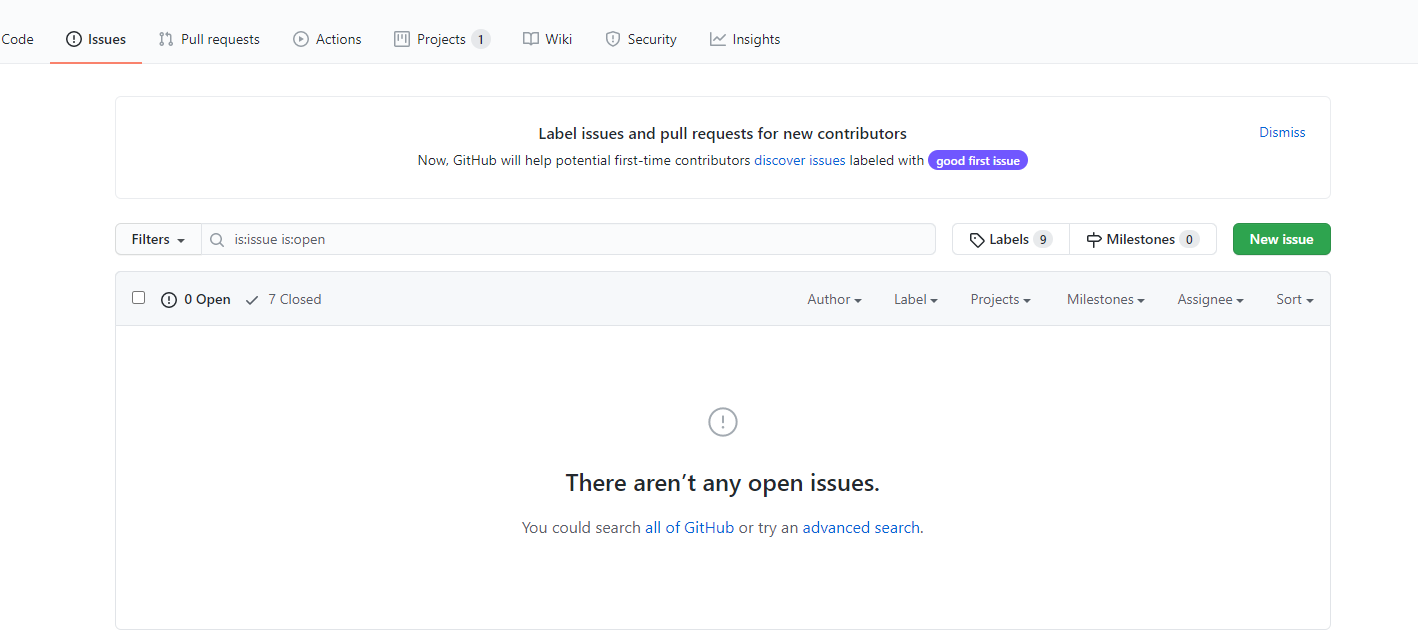


Dynamic Code Quality:

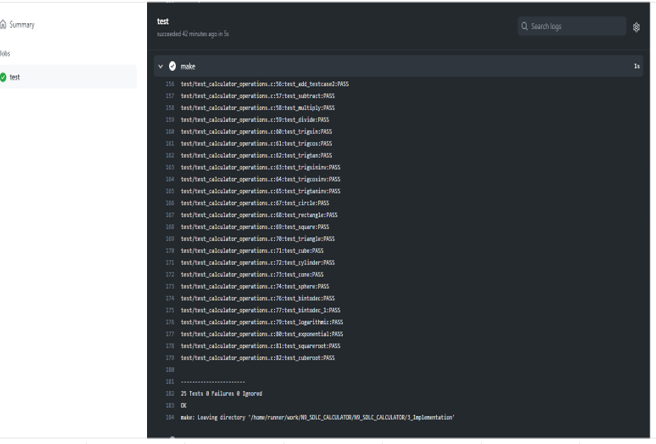


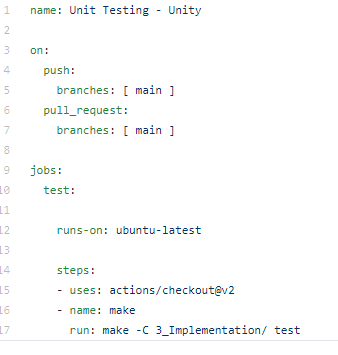


**GIT ISSUES**



#### Unit Testing





## **Individual Contribution & Highlights**

**Individual contribution:**

My Contribution to this project is to design features like Special Math Functions.

* Special math functions like logarithm, exponential, square root, cube root.

These are the features that I have implemented from Special math functions and financial calculations. But before implementing I have gone through some of the basic calculators present in the market according to that I have created a table of features and cost on calculator in order to understand the high-level and low-level requirement. Based on high level and low-level requirement of the calculator I started designing and implementing on the project.

I have written my code in src folder with the file name math\_function.c accordingly I have added special\_Math\_Function.h file in inc folder. I have modified test\_math\_operations.c where I have written my test cases.

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:**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| PS No. | Name | Features | Issues’ Raised | Issues Resolved | Issues Resolved |
| 99003786 | Sangam Patel | Arithmetic, Trigonometric | 5 | 2 | 11 |
| 99003784 | Souvik Jana | Area, volume (diff shapes) | 2 | 2 | 8 |
| 99003788 | Suvradeep Dutta | Binary to Decimal | 1 | 1 | 2 |
| 99003787 | Arikatla pujitha | Log, exp, sqrt, cube root | 2 | 2 | 4 |

### Summary

This was my first project on SDLC, previously the projects that I have done are hardware related so it’s a good experience.

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. Initially I was unable to run test cases but now it’s running.
2. Some workflows problems were there but now it has been solved.
3. Makefile problem while running in windows but it worked with Linux it was extension problem in building but it resolved in linux.
4. Git inspector was not working initially but now it’s working fine.
5. valgrind was not working previously because there was some issue in Makefile but now it’s working fine after resolving Makefile.

### Future Scope

* Features like matrix operations can be added.
* Features such as Physics operations (Acceleration, capacitive reactance, circular velocity, Coulomb’s Law, Gravitational force, projectile motion) can be added.
* Features like AP, GP sum (AP sum, GP sum) can be implemented in the future.

# Mini project -2 Python [Team/Individual]

## **1.Modules used:**

## Module used in this project is Python and SDLC.

### 2.Topic and Subtopics

|  |  |
| --- | --- |
| Folder | Description |
| 1\_Requirements | Documents detailing requirements and research |
| 2\_Design | Documents specifying design details |
| 3\_Implementation | All code and documentation |
| 4\_Test\_plan | Documents with test plans and procedures |

## **3.Objectives & Requirements**

**Introduction-**

* In this python project, we have to read multiple excel sheet data having 40 rows x 10 column that contain 3 similar columns in each sheet i.e. PS number, Name & Email address. Next, we have to take the user input for printing the desired student data in a master sheet. Here we are reading the multiple sheet and storing the similar student data into master sheet.
* Excel files should be placed in different directories and Master file should get the

data from all excel files with respect to user input

**About the project:**

The aim of the project is to extract the data present in different spreadsheets in one excel file as required by the user by different paths given by him. The excel sheet scrolls through all the spreadsheets with the following data common in all the sheets:

* Name:
* Ps Number:
* Email id:

The user defines the data that needs to be searched on the basis of the common data. The python program then reads the data corresponding to the particular data from different spreadsheets of excel. It then creates a master sheet and adds the data from all the sheets to it. In the end, the data to be provided to the user is printed to the console

# Features that are integrated in this project are

* Reading multiple excel sheets each having 40 rows x 10 column
* Searching methods to search details for user input values
* Combining all the similar data in news master sheet

# 4W's and 1'H

# Why:

* We are using to retrieve the data of an individual candidate from the excel workbook of 5 sheets where all the relevant data of 40 candidates is present.
* We can easily access the details of that individual by giving some input such as name, Ps no and email id.

## **What:**

* We are preparing the master excel sheet to search and retrieve data from all the 5 excel sheets that are created.
* It is used for easy search of a particular cell or data of a person
* It provides information of every person details like bio, academics, health and personal details

## **When:**

* Searching for person information
* To get the contact information
* To get the required details of that person educational qualification.

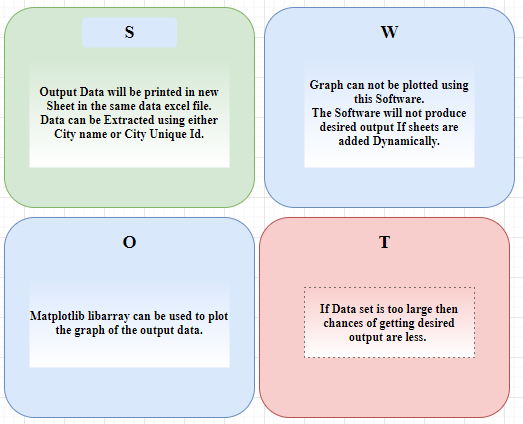
## **Where:**

* To check the information and bio of a person
* Very useful during emergency times like health issues
* We can also use it for evaluation of marks using the mail, search the location of person

## **How:**

* Input: - We need to give 3 inputs such as Name, Ps No and Email Id.
* Output: - We will get all the relevant information of that person whose name, PS no and email id is given.
* source: - All the relevant data will get copied in master sheet.

## **SWOT Analysis**



**High Level Requirements:**

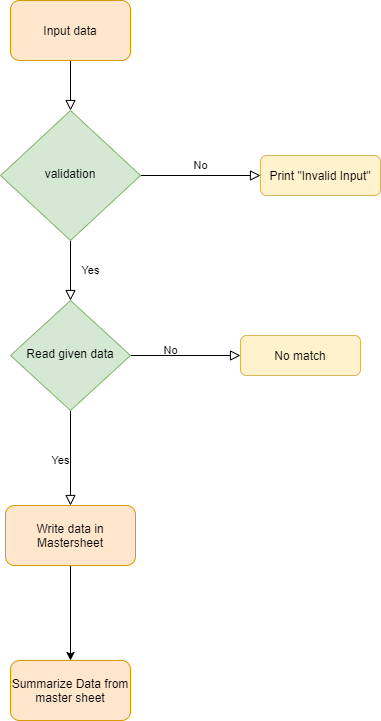
|  |  |  |  |
| --- | --- | --- | --- |
| ID | Requirements | Description | Status |
| HLR01 | Excel Work Book | Work Book contains 5 worksheets with one Master Sheet | Implementing |
| HLR02 | User Inputs | User has to give 3 Inputs like Name, PS No, Email Id | Implementing |
| HLR03 | Read & Write | Reading from existing sheets and writing to Master Sheet | Implementing |

**Low Level Requirements:**

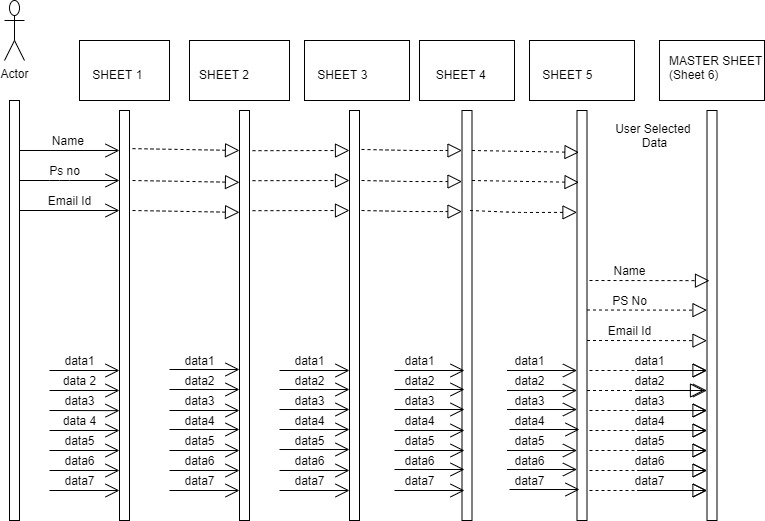
|  |  |  |  |
| --- | --- | --- | --- |
| ID | Requirements | Description | Status |
| HLR01-LR1 | Data Collection | worksheets contain the data of company details and academic details of user input | Implementing |
| HLR01-LR2 | Data Size | Each Worksheet Contains minimum of 40 rows and 10 columns | Implementing |
| HLR01-LR3 | Excel file format | The workbook file should be of .xlsx format | Implementing |
| HLR02-LR1 | Inputs | user can give multiple inputs like name, PS No, Email Id at once | Implementing |
| HLR03-LR1 | Reading Data | Reading all 5 Worksheets from workbook | Implementing |
| HLR03-LR2 | Searching Data | Search for specific data based on user specific inputs | Implementing |
| HLR03-LR3 | Writing Particular Data | Only specific kind of data should be written to Master sheet | Implementing |

## **4.Design:**

**HLR OBJECT DIAGRAM**

****

**LLR\_SEQUENCE DIAGRAM**



**Test Plan**

**High Level Testing**:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| ****Sl No.**** | ****Description**** | ****Expected Input**** | ****Expected Output**** | ****Actual Output**** | ****Type of Test**** |
| 1 | To search by Name, PsNo, EmailId | Number of inputs you want: | To retrieve the data of a particular person from different excels which are present in different directories with inputs like PS number, name, email id. The required output is all the candidate data and stored in the master sheet | To retrieve the data of a particular person from different excels which are present in different directories with inputs like PS number, name, email id. The required output is all the candidate data and stored in the master sheet | Requirement Based |

**Low Level Testing:**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Sl No.** | **Description** | **Expected Input** | **Expected Output** | **Actual Output** | **Type of Test** |
| **1** | To search by Name, PsNo, EmailId | Number of inputs you want:3  **Enter Name:**  Isabella Johnson  **Enter PS Number:**  99003709  **Enter emailId:**  i.johnson@randatmail.com  **Enter Name:**  James Evans  **Enter PS Number:**  99003710  **Enter emailId:**  j.evans@randatmail.com  **Enter Name:**  Arthur Lloyd  **Enter PS Number:**  99003711  **Enter emailId:**  a.lloyd@randatmail.com | All the data of Isabella Johnson, James Evans, Arthur Lloyd from different excel files which are present in different directories should print to mastersheet and also all the headers are copied to mastersheet | All the data of Isabella Johnson, James Evans, Arthur Lloyd from different excel files which are present in different directories should print to mastersheet and also all the headers are copied to mastersheet | Requirement Based |
| **2** | To search by Name, PsNo, EmailId | Number of inputs you want:2  **Enter Name:**  Albert Jones  **Enter PS Number:**  99003712  **Enter emailId:**  a.jones@randatmail.com  **Enter Name:**  David Miller  **Enter PS Number:**  99003713  **Enter emailId:**  d.miller@randatmail.com | All the data of Albert Jones, David Miller from different excel files which are present in different directories should print to mastersheet and also all the headers are copied to mastersheet | All the data of Albert Jones, David Miller from different excel files which are present in different directories should print to mastersheet and also all the headers are copied to mastersheet | Requirement Based |
| **3** | To search by Name, PsNo, EmailId | Number of inputs you want:1  **Enter Name:**  Agata Brown  **Enter PS Number:**  99003714  **Enter emailId:**  a.brown@randatmail.com | All the data of Agata Brown from different excel files which are present in different directories should print to mastersheet and also all the headers are copied to mastersheet | All the data of Agata Brown from different excel files which are present in different directories should print to mastersheet and also all the headers are copied to mastersheet | Requirement Based |
| **4** | To search by Name, PsNo, EmailId | Number of inputs you want:0 | Only the headers from all the sheets will be copied to the mastersheet | Only the headers from all the sheets will be copied to the mastersheet | Requirement Based |
| **5** | To search by Name, PsNo, EmailId | Number of inputs you want:1  **Enter Name:**  Paige Williams  **Enter PS Number:**  99003701  **Enter emailId:**  j.elliott@randatmail.com | Only the headers from all the sheets will be copied to the mastersheet because Name , PS Number , Emailid doesn’t match | Only the headers from all the sheets will be copied to the mastersheet because Name , PS Number , Emailid doesn’t match |  |

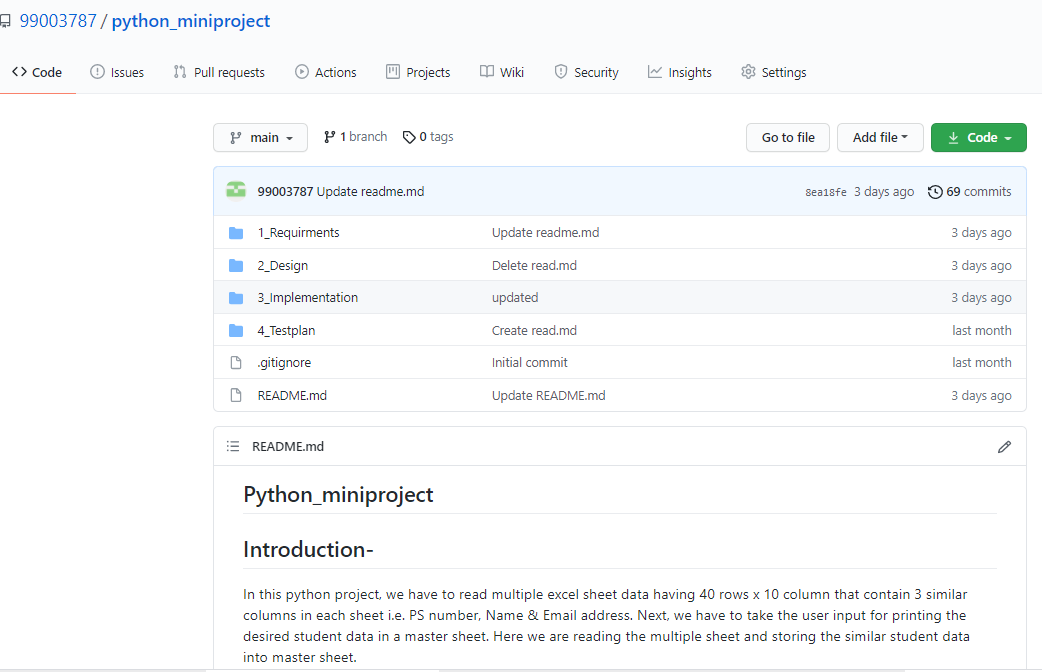
## **Implementation Summary**

Implementation folder has the python file with extension “.py” named **pytho.py**containing the code. It also contains the excel file which consists of 7 sheets namely “Sheet1”, “Sheet2”, “Sheet3”, “Sheet4”, “Sheet5”, “mastersheet” and “summarysheet”. “Sheet1” to “Sheet5” are populated manually. After we run compile and run the **pytho.py** file, data from “Sheet1” to “Sheet5” gets displayed on the mastersheet. The number of rows and columns on the mastersheet are displayed on the summarysheet.

### Git Link

<https://github.com/99003787/python_miniproject.git>

### Git Dashboard



### Git Summary

There are four folders namely:

* Requirements:

There is a readme.md file consisting of:

* + - Introduction
    - HLR - It consists of the high-level requirement table.
    - LLR - It consists of the low-level requirement
    - 4W & 1H
    - SWOT Analysis
* Design

There is a readme.md file consisting of:

* HLR - It consists of the high-level requirement UML diagram.
* LLR – It consists of the low-level requirement UML diagram.
* Implementation

Here there are two files:

* pytho.py file consisting the code.
* Mini\_project.xlsx file consisting of the raw data.
* Test Plan

Here there is a readme.md file consisting of:

* + - * High Level Test Plan
      * Low Level Test Plan

### Summary

This miniproject is based on automated excelsheets where we can retrieve and extract a particular data from a large group of files into a master sheet using python automation. our main aim is to make data entry job easy and We are going to write a Python program that can process thousands of spreadsheets and manage all the calculations under a second for the user.

As per the objective, firstly I created an excel file that has five sheets. Then I started populating the sheets with data like NAME, PS NUMBER, Email, D.O.B, etc. The PS NUMBER, Name, Email column is common throughout all the five sheets. After we run compile and run the **pytho.py** file, data from “Sheet1” to “Sheet5” gets displayed on the mastersheet. After that I am counting the number of rows and columns in the mastersheet and then reflecting the result into a sheet named summarysheet

### Challenges faced and how were they overcome

## Implementing the library

## Writing data in master sheet

* Creating the database
* Reading the file path from multiple directory

# 

# Mini Project -3 EMBEDDED C => [Team]

# Modules Used:

## Modules used in this project are Embedded Systems and Embedded C Programming and was implemented on the hardware STM32.

**Topics and Subtopics:**

1)Driver API Development

a) GPIO

b) ADC

c) SPI, UART, I2C

d) External interrupt

e) Debugging using STM Board

2) Driver Development (Hardware Abstraction Level- HAL)

* 1. GPIO
  2. ADC
  3. External Interrupt
  4. Debugging using STM Board

## **Objectives & Requirements:**

To implement different CAR Module features using STM32f407VG Microcontroller featuring 32-bit ARM-M4 with FPU core.

Car Module Features:

1. Fuel level detecting system (Moisture sensor)
2. Door Lock control system (Motion sensor)
3. Emergency vehicle siren system (RGB sensor)
4. Horn system in vehicles (Buzzer sensor)

**Components Used:**

1. STM32f407VG Microcontroller
2. Breadboard
3. LED
4. LDR sensor
5. Moisture sensor
6. Motion sensor
7. RGB color sensor
8. Ring Buzzer sensor
9. Jumper Wires

**HIGH LEVEL REQUIREMENTS:**

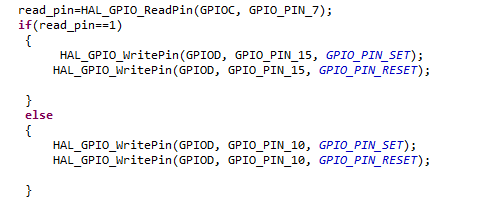
* RGB sensor is used for the emergency siren for vehicles in emergency situations.
* Buzzer sensor is used for vehicle horn and alarm in reverse gear.
* LDR sensor for automatic headlights during night times and interior lights in car when we open doors.
* Humidity sensor for fuel detection in petrol tanks in vehicles.

**LOW LEVEL REQUIREMENTS:**

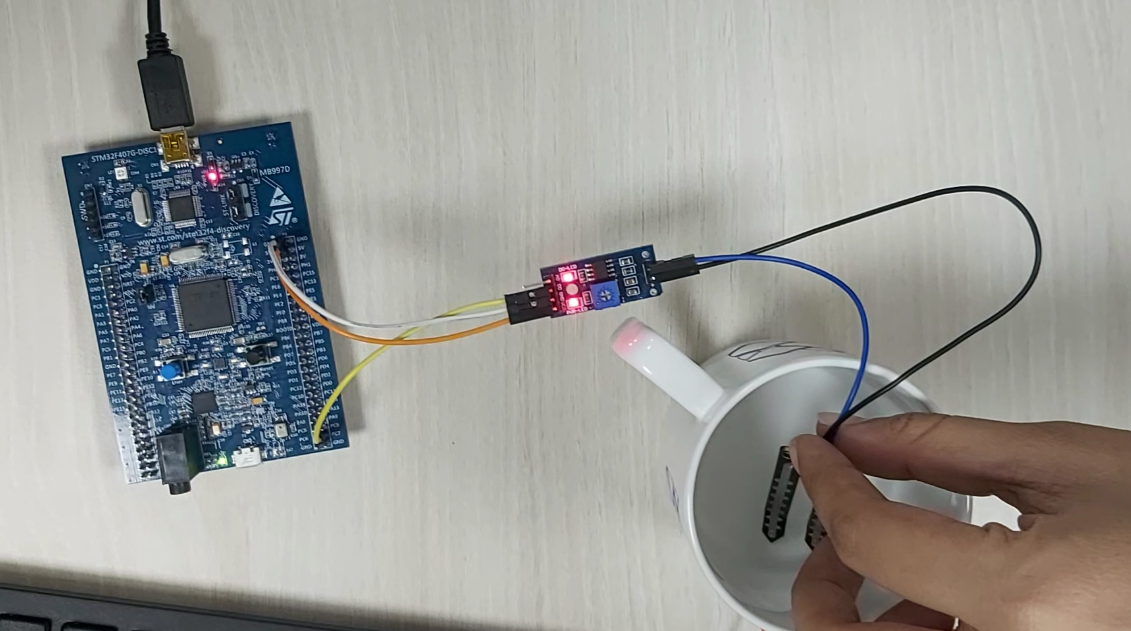
* RGB sensor is used in emergency situations like siren in ambulance, Police vehicles, Parking lights for vehicles, Fire engines. When input is 1, the RGB will glow.
* From STM32 switch takes input and output is connected to buzzer when we press the switch the buzzer will sound. It is used in car horn, parking sounds. When the switch is on the input should be 1, the buzzer will sound.
* LDR sensor for automatic headlights during night times and Interior lights in car when we open doors. When the sensor detects darkness, the LED in the sensor glow.
* Humidity sensor prototype can be used in fuel detection in petrol tanks in vehicles when petrol level in tank reaches below par level then this sensor indicates a warning as LED light.

**Design:**

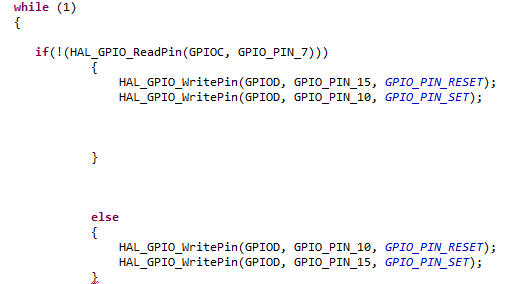
**CODE SNIPPET FOR FUEL LEVEL DETECTING SYSTEM USING MOISTURE SENSOR:**



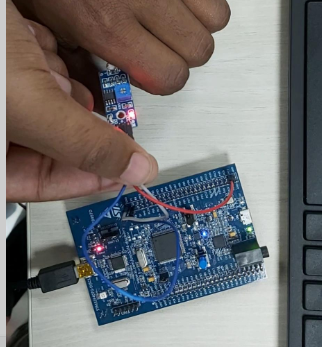
**CIRCUTE DIAGRAM FOR FUEL LEVEL DETECTING SYSTEM USING MOISTURE SENSOR:**



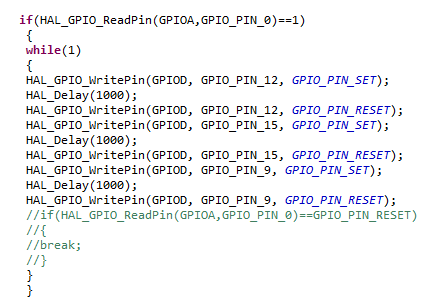
**CODE SNIPPET FOR DOOR LOCK CONTROL SYSTEM USING MOTION SENSOR:**

****

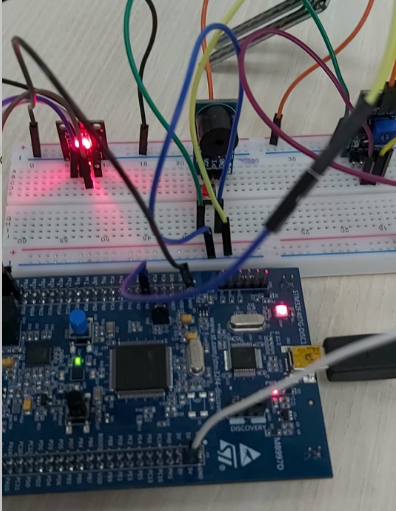
**CIRCUTE DIAGRAM FOR DOOR LOCK CONTROL SYSTEM USING MOTION SENSOR:**



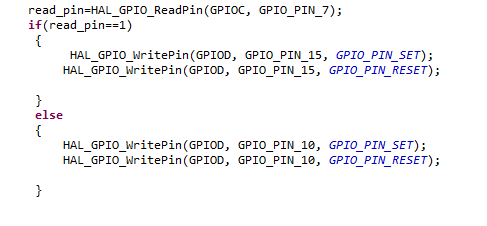
**CODE SNIPPET FOR EMERGENCY VEHICLE SIREN SYSTEM USING RGB COLOR SENSOR:**



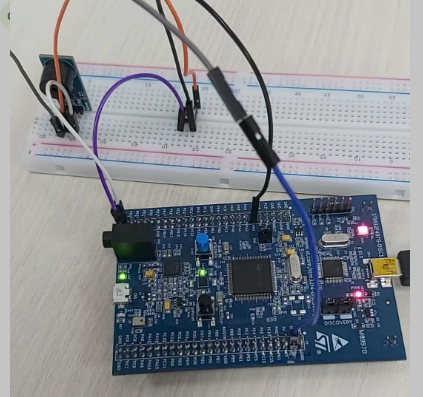
**CIRCUTE DIAGRAM FOR EMERGENCY VEHICLE SIREN SYSTEM USING RGB COLOR SENSOR:**



**CODE SNIPPET FOR HORN SYSTEM IN VEHICLES USING BUZZER SENSOR:**



**CIRCUIT DIAGRAM FOR HORN SYSTEM IN VEHICLES USING BUZZER SENSOR:**



**TEST CASES:**

**HIGH LEVEL TEST PLANS:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| SL No | TEST\_ID | TESTING FUNCTION | EXPECTED INPUT | EXPECTED OUTPUT |
| 1 | HLR\_1 | Emergency Siren lights for Vehicles | Emergency siren for vehicles in emergency situations. | Different lights in RGB will glow. |
| 2 | HLR\_2 | Horn or Reverse gear sensor | When the switch is on, here buzzer gets activated. | Buzzer will sound when the switch is on. |
| 3 | HLR\_3 | Automatic light sensor | LDR sensor sense the presence and absence of light is necessary. | Turn on/off car lights. |
| 4 | HLR\_4 | Fuel level detection | Moisture sensor senses the fuel level. | If the fuel level is below the range the LED will glow. |

**LOW LEVEL TEST PLANS:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| SL No | TEST\_ID | TESTING FUNCTION | EXPECTED INPUT | EXPECTED OUTPUT |
| 1 | HLR\_1-LLR1 | Emergency Siren lights for Vehicles | When the input is 1 i.e. When the RGB sensor is active | The green LED will glow i.e. the GPIO pin with PORT D 12 will be set |
| 2 | HLR\_1-LLR2 | Emergency Siren lights for Vehicles | When the input is 0 i.e. When the RGB sensor is deactivate. | The green LED will stop i.e. the GPIO pin with PORT D 12 will be reset. |
| 3 | HLR\_2-LLR1 | Horn or Reverse gear sensor | When the input Is 1 i.e. when the switch is on | The green LED will glow i.e. the GPIO pin with PORT D 12 will be set |
| 4 | HLR\_2-LLR2 | Horn or Reverse gear sensor | When the input Is 0 i.e. when the switch is off | The green LED will stop i.e. the GPIO pin with PORT D 12 will be reset. |
| 5 | HLR\_3-LLR1 | Automatic light sensor | When the input is 1 i.e. When the darkness is detected around | The green LED will glow i.e. the GPIO pin with PORT D 12 will be set |
| 6 | HLR\_3-LLR2 | Automatic light sensor | When the input is 0 i.e. When the lights are detected around | The green LED will stop i.e. the GPIO pin with PORT D 12 will be reset. |
| 7 | HLR\_4-LLR1 | Fuel level detection | When the input is 1 i.e. When the fuel touches the sensor | The green LED will glow i.e. the GPIO pin with PORT D 12 will be set |
| 8 | HLR\_4-LLR2 | Fuel level detection | When the input is 0 i.e. When the fuel away from the sensor | The green LED will stop i.e. the GPIO pin with PORT D 12 will be reset. |

### Summary

**Challenges faced and how were they overcome**

* In some systems STM32 microcontroller board is working and in some systems, it’s not so we have tested multiple microcontrollers
* Even few sensors didn’t work properly so we have tested with multiple sensors
* There were even few issues with our code at the beginning, after proper debugging we are able to eliminate the errors.

# Mini project -4 [Individual]

## **Module**

“Kernel and 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

#### Requirements

|  |  |  |  |
| --- | --- | --- | --- |
| SL No. | Requirements | Description | Status |
| 1 | System call to echo string | The system call echo backs the given string. | Implemented |
| 2 | System to traverse process list and print pid | System call traverses through process list and print pid of all the running process. | Implemented |
| 3 | System to traverse process list and get attributes like state, priority and pid of the calling process | System call traverses through process list and retrieves state, priority and pid of calling process. | Implemented |

## **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 |

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- 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 rootfs.img /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 \

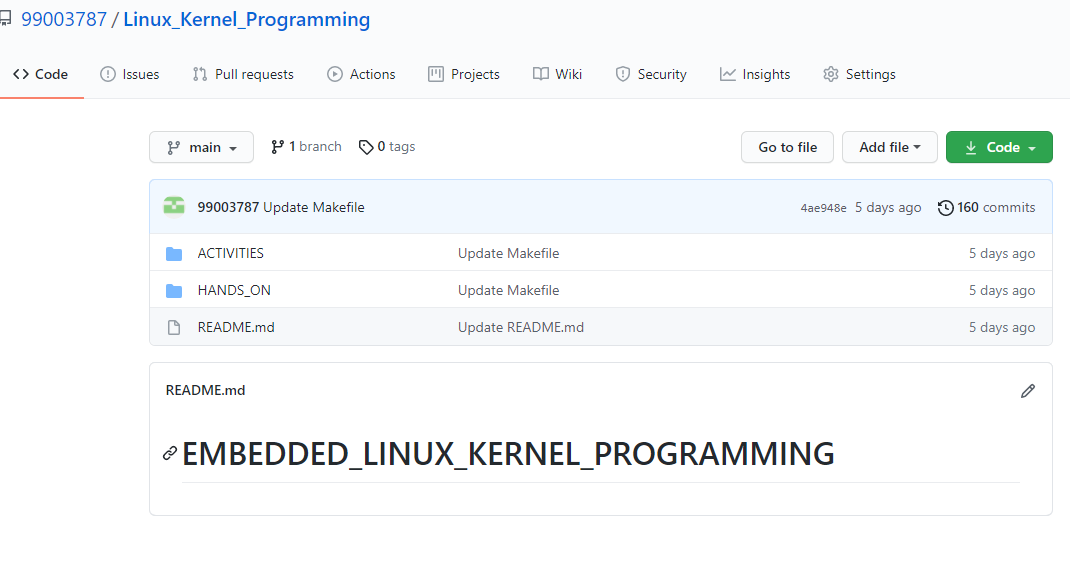
-sd rootfs.img -append "console=ttyAMA0 root=/dev/mmcblk0 rw"

* 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.

### Git Link

<https://github.com/99003787/Linux_Kernel_Programming.git>

### Git Dashboard



### 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.

### 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.