

Mini Project Report – Applied System Development Life Cycle and Software Testing

Multipurpose Electrical & Electronics Calculator



L&T Technology Services



GENESIS Learning Report – Module Name

Document History

Ver. Rel. No.	Release Date	Prepared By	Reviewed By	To be approved By	Remarks/Revision Details
1	24/08/21	99005716	99005716	Kartik Mudaliar	
2	24/08/21	99005756	99005756	Kartik Mudaliar	
3	24/08/21	99005749	99005749	Kartik Mudaliar	
4	24/08/21	99005705	99005705	Kartik Mudaliar	
5	24/08/21	99005739	99005739	Kartik Mudaliar	
6	24/08/21	99005722	99005722	Kartik Mudaliar	
7	24/08/21	99005718	99005718	Kartik Mudaliar	
8	24/08/21	99005747	99005747	Kartik Mudaliar	
9	24/08/21	99005752	99005752	Kartik Mudaliar	
10	24/08/21	99005719	99005719	Kartik Mudaliar	

TABLE OF CONTENTS

Document History.....	3
Introduction	5
Product Built: Multi-functional Electronics Calculator	5
SWOT Analysis.....	6
4W's and 1'H.....	7
Who.....	7
What.....	7
When.....	7
Where	7
How.....	7
Requirements.....	8
Research	8
Ohm's Law.....	8
KCL.....	8
KVL	8
Duty Cycle	8
Motor Parameters	8
Half Adder.....	8
Half Subtractor	9
PCB Trace Width	9
Resistor color code	9
Current Division & Voltage Division	9
Star & Delta Conversion	9
Equivalent Resistance, Capacitance and Inductance of Circuits	10
Selection of multi & single core wire	10
Circular Convolution.....	10
Tensile Strength	10
Cost & Feature	11
Defining the System	11
How to Run	11
Detailed Requirements.....	12
High Level Requirements	12
Low Level Requirements.....	13
Design	15
Test Plan	40
High Level Test Plan	40
Low Level Test Plan.....	41
Implementation Screenshots:-	43

Introduction

In this project, a new system is designed to perform various electrical and electronics calculations and conversions. This system allows the user to solve problems quickly and in an efficient manner. In this project the programming language used is C.

Product Built: Multipurpose Electrical & Electronics Calculator



SWOT Analysis



4W's and 1'H

Who

- Any person who intends to play with some fundamentals of Electrical and Electronics arena.

What

- A virtual and command line interface based project which does basic calculation and conversions of basic electrical concepts, analysis and monitoring of various parameters of electrical machines.

When

- The project is developed as a part of "Applied SDLC Module" of Genesis internship program.
- The project can be employed when there is need in calculation of basic Electrical concepts either in day-to-day apllication or for labrotary procedures.

Where

- The program is developed in Visual StudioCode IDE.
- Users can use this application on their desktop or laptop terminal

How

- The module was build solely by using C and concepts like "Make Files" has been implemented.
- Testing has been carried out by "Unity"
- And the code quality is ensured by screening the code in various code reviewing entities such as "Codacy", "Git Inspector", etc.

Requirements

Research

Need and Importance of Electronics formula Calculations

The requirements of the project was to analyse and find out what is already available and what improvement can be made so that its useful to most to of the people.

Ohm's Law

- In electronics, Ohm's law is vitally important to describe the circuits, as it relates voltage to the current with the resistance value.
- This law was derived by the German Physicist, George Simon Ohm.
- Ohm's law is used to maintain the desired voltage drop across the electronic components and it is used in DC ammeter and other DC shunts to divert the current. Its application also ranges from household appliances to massive projects.

Kirchoff's Current Law (KCL)

- It is first Kirchoff's law that deals with the conservation of charge.
- The law states that total current entering a junction or node is exactly equal to the charge leaving the node as it has no place to go except to leave, as no charge is lost within the node.

Kirchoff's Voltage law (KVL)

- It is the second law that deals with the conservation of energy around a closed circuit path.
- It states that for a closed loop series path the algebraic sum of all the voltages around any closed loop in a circuit is equal to zero.
- It is used for voltage measurement in circuits.

Duty cycle

- Duty cycle is the ratio of time a load or circuit is ON compared to the time the load or circuit is OFF.
- It is expressed as a percentage of ON time.

Motor parameters

- The motor parameters include frequency, output power, speed and torque. The speed of a motor depends on the rotational velocity of the motor shaft and torque is a rotating force produced by an engine's crankshaft.
- The relation between torque and speed are inversely proportional to each other.
- Frequency is the rate at which current changes its direction per second in a machine. The output power is calculated with the help of input power and losses in the motor.

Half Adder

- It is a arithmetic combinational logic circuit designed to perform addition of two single bits.

GENESIS Learning Report – Module Name

- It contains two inputs and produces two outputs.
- Inputs are called Augend and Addend bits and Outputs are called Sum and Carry.
- Sum = A XOR B
- Carry = A AND B

Half Subtractor

- It is a combinational logic circuit designed to perform subtraction of two single bits.
- It contains two inputs (A and B) and produces two outputs (Difference and Borrow-output).
- Difference = A XOR B
- Borrow = !A AND B

PCB Trace Width

- The trace is a conducting track that connects components electrically and allows electric current to flow with little resistance.
- The single trace on Printed circuit Board(PCB) is the equivalent of a wire for conducting signals. Each trace consists of flat, narrow part of the copper foil that remains after etching.
- The PCB trace widths are dependent on the needs of the signal. The thinner traces are for general purpose Transistor-Transistor Logic(TTL) signals and the thicker traces are optimized for current carrying capacity and used for high power-related functions

Resistor Colour Code

- In 1920, the resistor colour code was developed by Radio Manufacturers Association (RMA) as a fixed resistor colour coding. This colour coding is used to quickly identify the resistive value and its percentage of tolerance.
- The most popular mnemonic created to easily memorize the sequence of colours is BBROYGBVW.
- The composition has 3 to 6 resistor colour bands. The three band resistor is for multiplier with no tolerance, 4 band resistor with one tolerance band, 5 band resistor with 4 multipliers and one tolerance bond, 6 band resistor have an additional band as temperature coefficient.

Current division & Voltage division

- Current Divider Circuits are parallel circuits in which the source or supply current divides into a number of parallel paths.
- All the components have their terminals connected together sharing the same two end nodes.
- A voltage divider is a simple circuit which turns a large voltage into a smaller one.
- Voltage division is the result of distributing the input voltage among the components of the divider.

Star and Delta Conversions

- In a 3-phase, 3-wire supply or 3-phase load is connected in one type (Star/Delta), it can be easily changed into an equivalent type (Delta/Star) by using either Star-Delta transformation or Delta-Star transformation.

GENESIS Learning Report – Module Name

- When three branches are connected nose to tail, they form a triangular closed loop, it is referred as Delta connection. Whereas the terminals are connected to a common point to form a Y like pattern, is is referred as Star connection.
- This conversion is used to establish equivalence for the networks within the terminals.

Selection of multi and single core wire

- The conductor cross-section shall be chosen such that the conductor withstands prospective short circuit current for a specified duration of time.
- Area of cross-section of the conductor shall be sized to carry estimated load current continuously such that the temperature rise of the conductor is within the acceptable limits for the installation conditions foreseen
- Voltage drop within the cable is within the permissible limits so that the functionality of the connected load by the cable remains unaffected

Equivalent Capacitance and Inductance of Circuits

- The equivalent capacitance and inductance of a circuit can be calculated by their equivalent formulas
- Passive components in parallel and series circuits add up in ratios relating to their properties

Tensile strength

- Tensile strength is the amount of load or stress that can be handled by a material before it stretches and breaks.
- It is the material's resistance to tension that is caused by mechanical loads applied to the material.
- It measures the force required to pull something such as rope, wire or a structural beam to the point where it breaks.

Circular Convolution

- Circular convolution is a special case of periodic convolution, which is the convolution of two periodic functions that have the same period.
- It is the mathematical way of combining two signals to form a third signal.
- Linear Convolution is the basic operation to calculate the output for any linear time invariant system. Circular convolution is the same thing but considering that the support of the signal is periodic.

Implementation with Programming Language C

- C defines datatypes like double which have high precision and can be used to implement the algorithms with high accuracy.
- The loops and precision can be used to iterate the algorithms in an efficient manner.

Cost and Features

- This system saves a lot of time for the user and results in quick and accurate manner.
- The heavy calculations are made simple with this system and it can be accessed easily.

Defining the System

In this system, the essential parameters are obtained from the user as input and this system will compute the values depending on the choice made by the user for calculation.

Steps to test and run the program:

1. clone the repo https://github.com/tanmaya191/tanmaya191-SDLC_8_Thunderbirds.git
2. open 3_Implementation folder
3. Run command: **make** (To build the code)
4. Run command: **make test** (To test the code)
5. Run command: **make run** (To run the code)

Detailed Requirements

High Level Requirements

ID	Description	Status
HA01	Calculation based on color code of the resistance	Implemented
HA02	Calculation of Current and Voltage using Current Division and Voltage Division Rule	Implemented
HA03	Star & Delta Conversions	Implemented
HA04	Calculation of maximum current in the cable across the Load	Implemented
HA05	Determining the voltage and current using ohms law	Implemented
HA06	Obtaining Series and Parallel resistance	Implemented
HA07	Finding the Traces of PCB designing and calculation of its Traces	Implemented
HA08	Calculation of Duty cycle	Implemented
HA09	Determing the Parameteres of Electrical Machines	Implemented
HA10	Obtaining the Expression for Half Adder	Implemented
HA11	Obtaining the Expression for Half Subtractor	Implemented
HA12	Calculating Equivalent capacitance and Inductance both in series and parallel	Implemented
HA13	Finding Electrical parameters using ohm's law	Implemented
HA14	Finding the total voltage in a system of many loops using Kirchoff's voltage law	Implemented
HA15	Calculating the amount of cuurent entering the circuit and leaving the circuit using Kirchoff's current law	Implemented
HA16	Calculating the tensile strength of conductors	Implemented
HA17	Calculating Circular Convolution	Implemented

Low Level Requirements

ID	Description	HLR	Status
LA01	Calculation of Resistance from the resistor color band	HA01	Implemented
LA02	Current Divsion Calculation using combination of resistance values	HA02	Implemented
LA03	Voltage Divsion Calculation using combination of resistance values	HA02	Implemented
LA04	Conversion of Star Network to Delta Network by evaluating value of Ra Rb RC	HA03	Implemented
LA05	Conversion of Delta Network to Delta Network by evaluating value of Ra Rb RC	HA03	Implemented
LA06	Maximizing the Value of Current and finding the Voltage Drop Across the Load	HA04	Implemented
LA07	Calculating Current, Resistance, Voltage using Ohms Law	HA05	Implemented
LA08	Equivalent Resistance(Series) when Current accross the Resistor are same. (Decreasing of Current)	HA06	Implemented
LA09	Equivalent Resistance(Parallel) when Current accross the Resistor are same. (Increasing of Current)	HA06	Implemented
LA10	We have to give current value to "trace" which results in increase in trace temperature below a specified limit.	HA07	Implemented
LA11	By giving the length of the trace, total resistance, voltage drop, and power loss due to trace resistance are also calculated.	HA07	Implemented
LA12	Duty Cycle is Being calculated by calculating Ton Period and Toff Period of the wave	HA08	Implemented
LA13	Calculating Power Input and Output using Voltage and Current	HA09	Implemented
LA14	Calculating Speed, Torque of Motor/Generator	HA09	Implemented
LA15	Input of 4 Sequence Given 0 0, 0 1, 1 0, 1 1 For Half Adder	HA10	Implemented
LA16	Performing XOR and AND Operation in all the 4 Sequence 0 0, 0 1, 1 0, 1 1 to get Half Adder Result	HA10	Implemented
LA17	Input of 4 Sequence Given 0 0, 0 1, 1 0, 1 1 For Half Subtractor	HA11	Implemented
LA18	Performing XOR and (!)Not operation with AND Operation combined to all the 4 Sequence 0 0, 0 1, 1 0, 1 1 to get Half Subtractor Result	HA11	Implemented

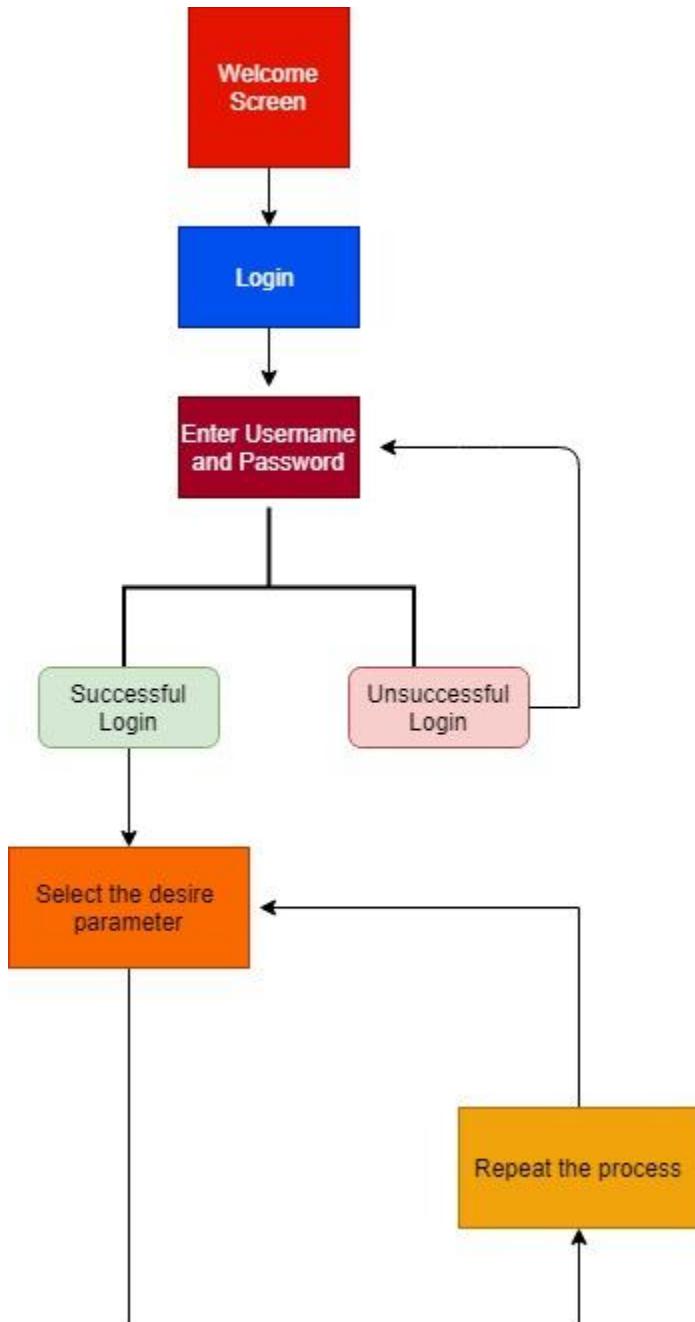
GENESIS Learning Report – Module Name

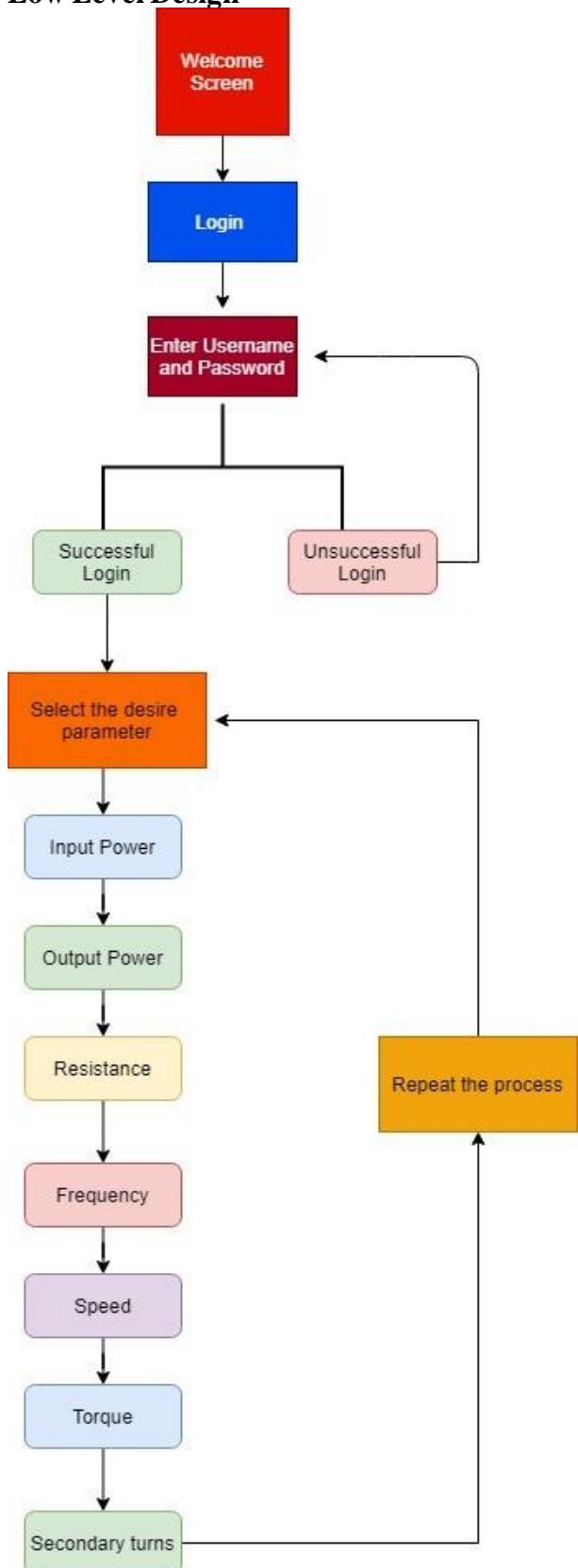
LA19	Calculating resistance using voltage and current	HA13	Implemented
LA20	Calculating current using voltage and resistance	HA13	Implemented
LA21	Calculating voltage using current and resistance	HA13	Implemented
LA22	Confirmation of total nodes is always equals to sum of nodes where current comes in and nodes where current flows away	HA15	Implemented
LA23	Calculating area of cylindrical conductors to find tensile strength	HA16	Implemented
LA24	Calculating area of shaped conductors to find tensile strength	HA16	Implemented
LA25	Giving Input sequence of 2 periodic functions	HA17	Implemented
LA26	Calculating Circular Convolution for 2 periodic input functions	HA17	Implemented

Design

**Application 1 - To find the Machine Parameters like:-
Input Power, Output Power, Resistance, Speed, Frequency and Torque.**

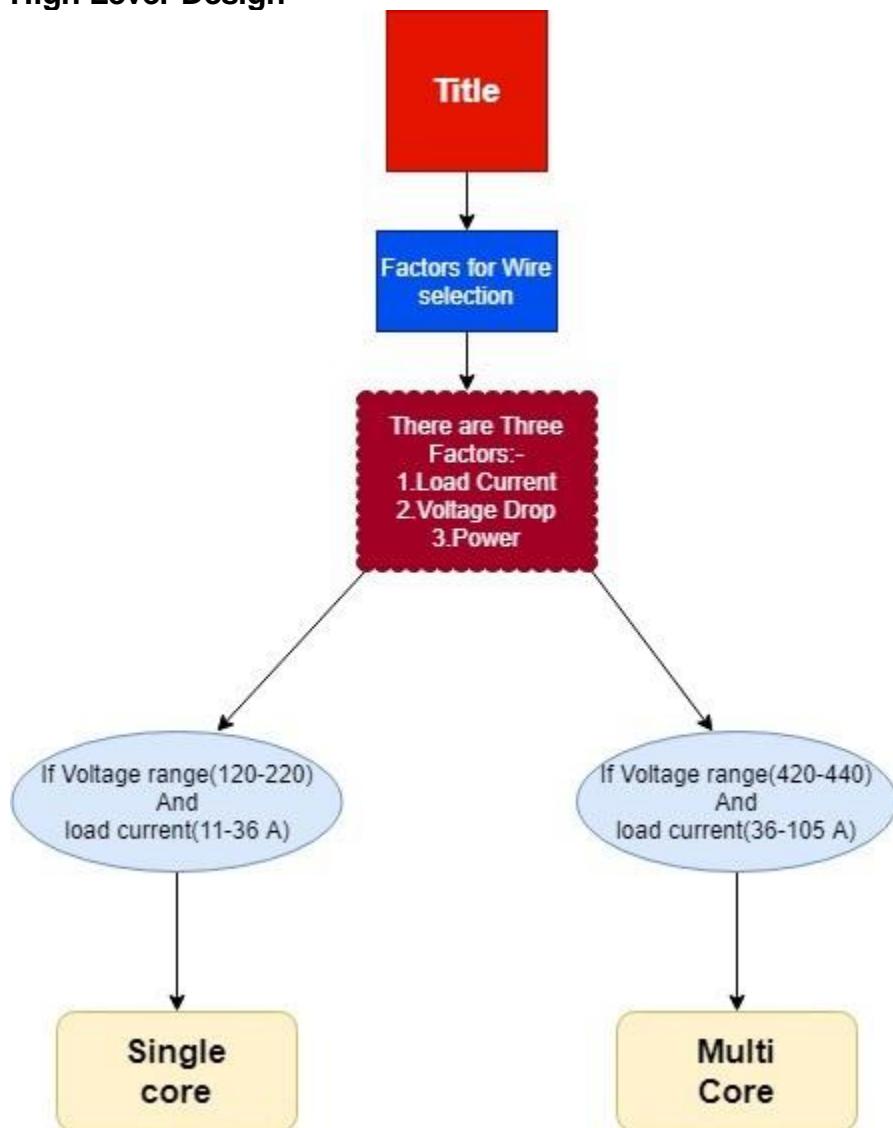
High Level Design



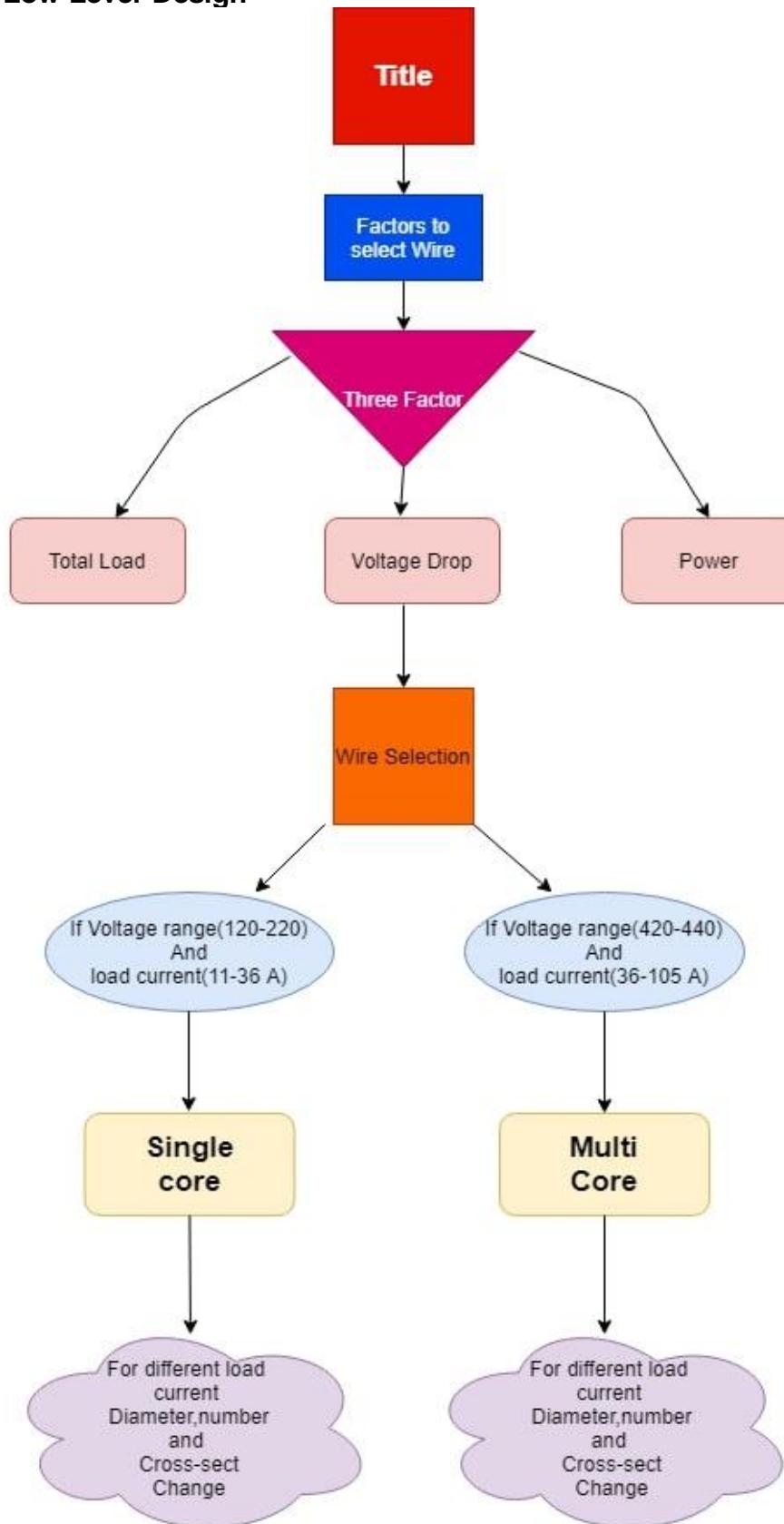
Low Level Design

Application 2 - How to Select Multi and Single core Wires.

High Level Design



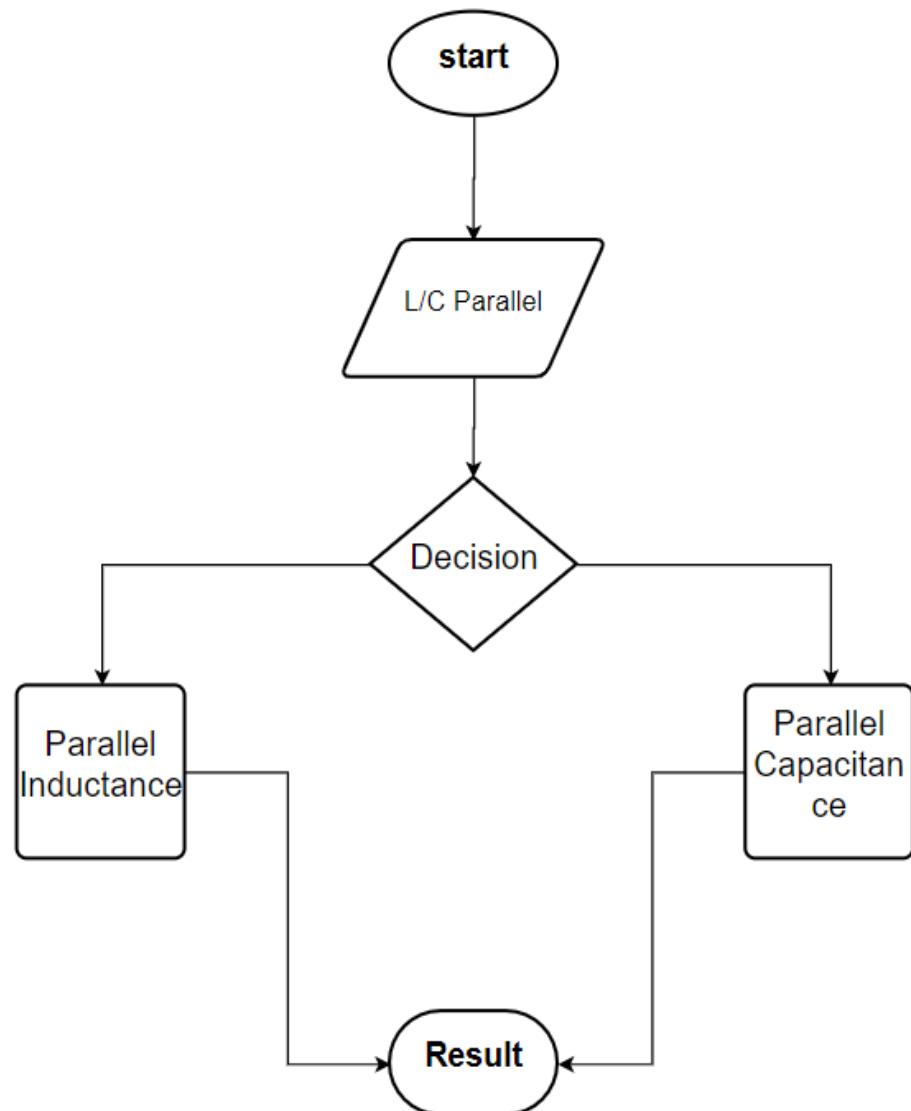
GENESIS Learning Report – Module Name
Low Level Design



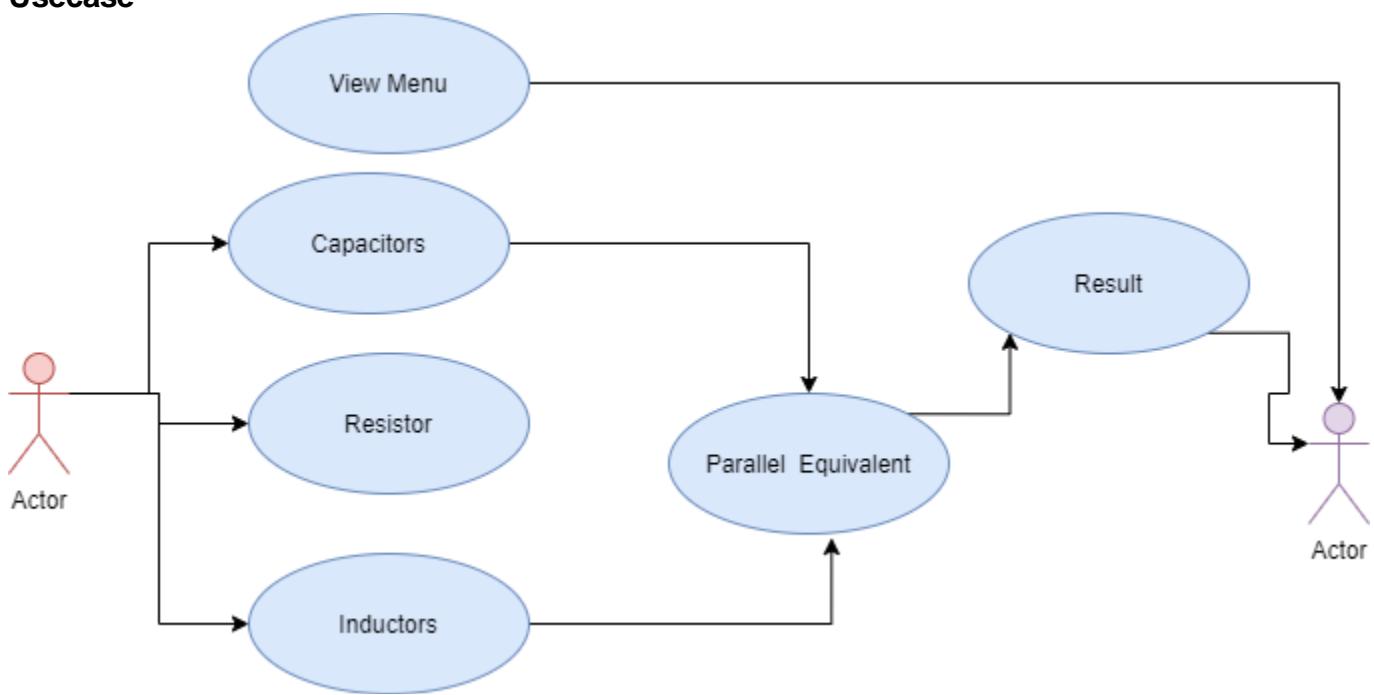
Application 3 - Calculating equivalent capacitance and inductance in parallel connection

Architecture

100 200 300 400 500 600 700 800



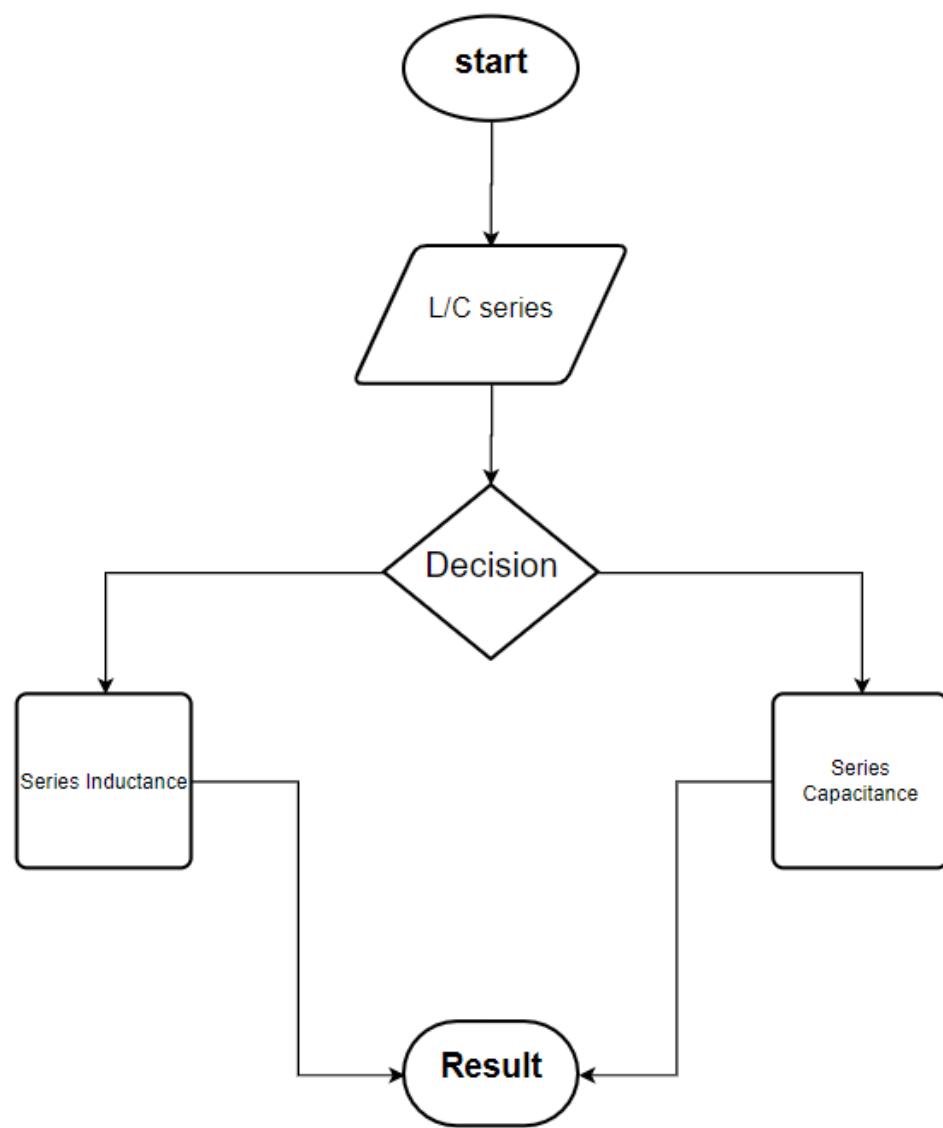
GENESIS Learning Report – Module Name
Usecase



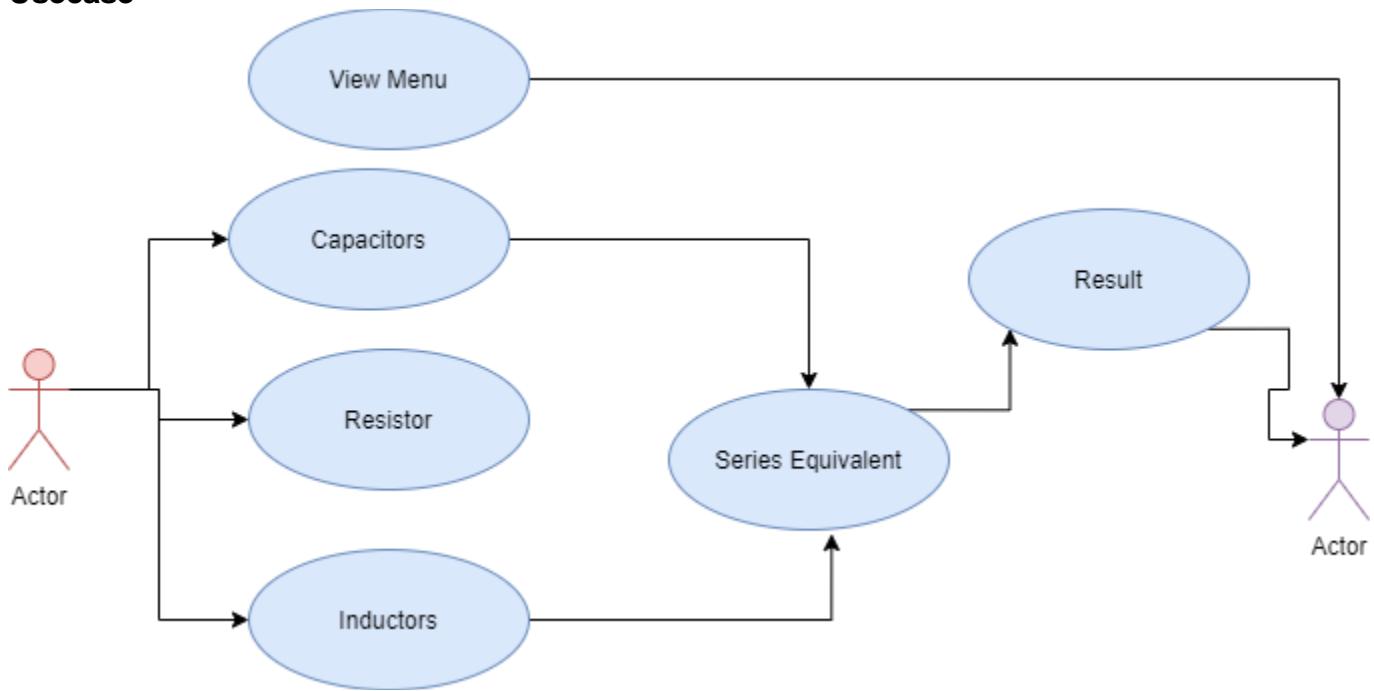
Calculating equivalent capacitance and inductance in series connection

Architecture

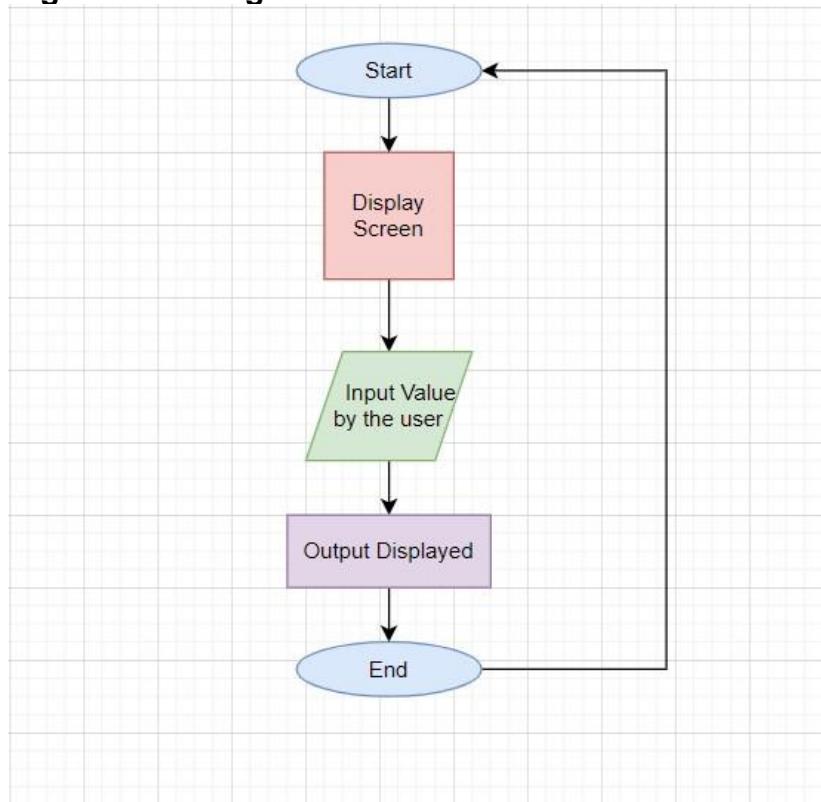
100 200 300 400 500 600 700 800



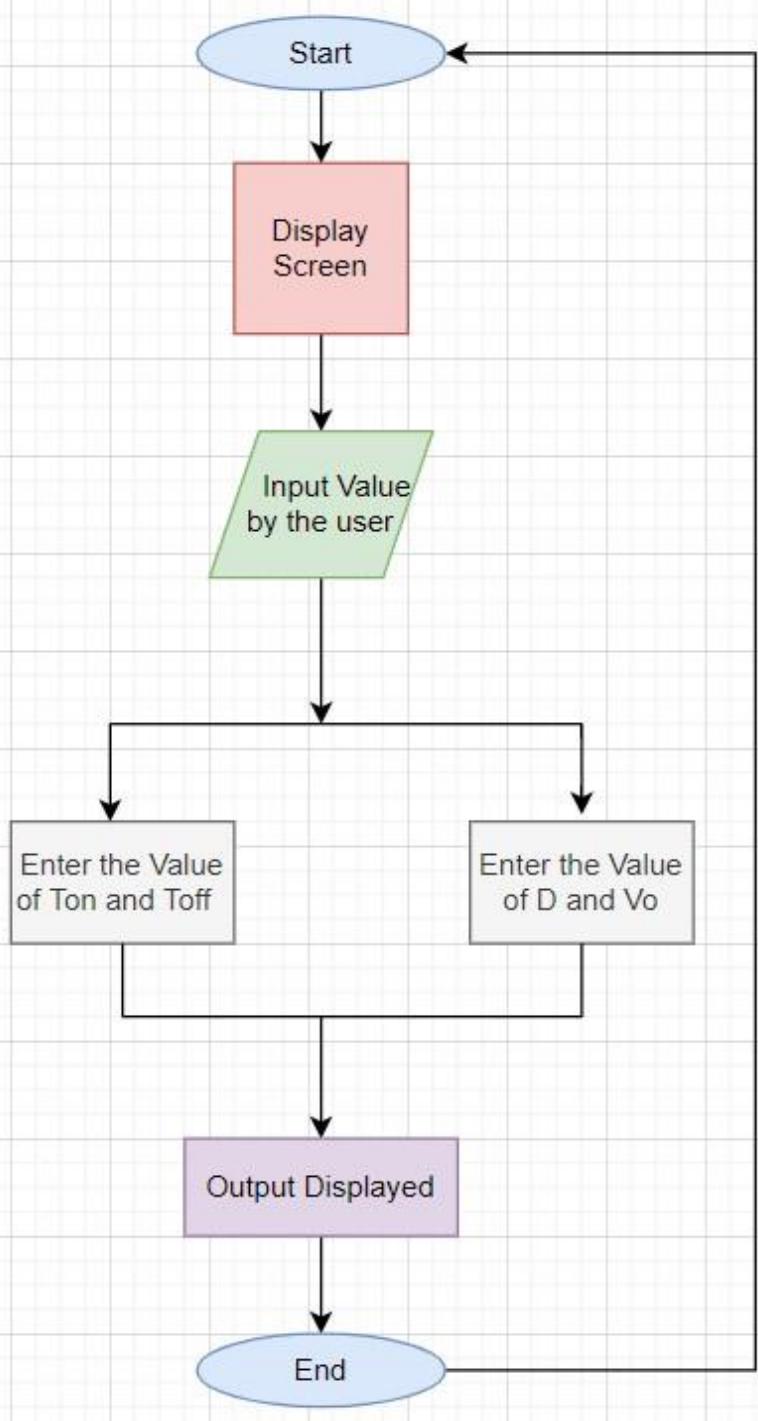
GENESIS Learning Report – Module Name
Usecase



Application 4 To Find Duty Cycle and Voltage of the Output waveform
High Level Design

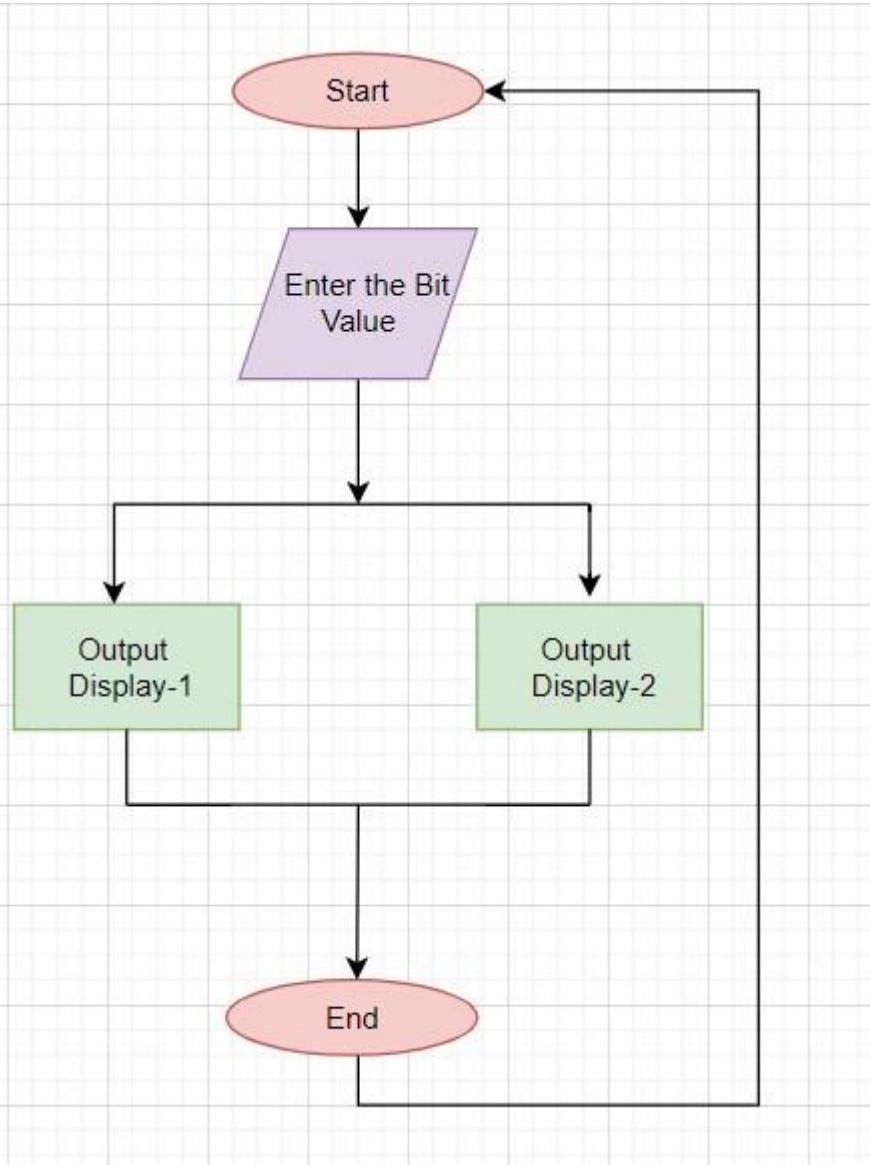


GENESIS Learning Report – Module Name
Low Level Design

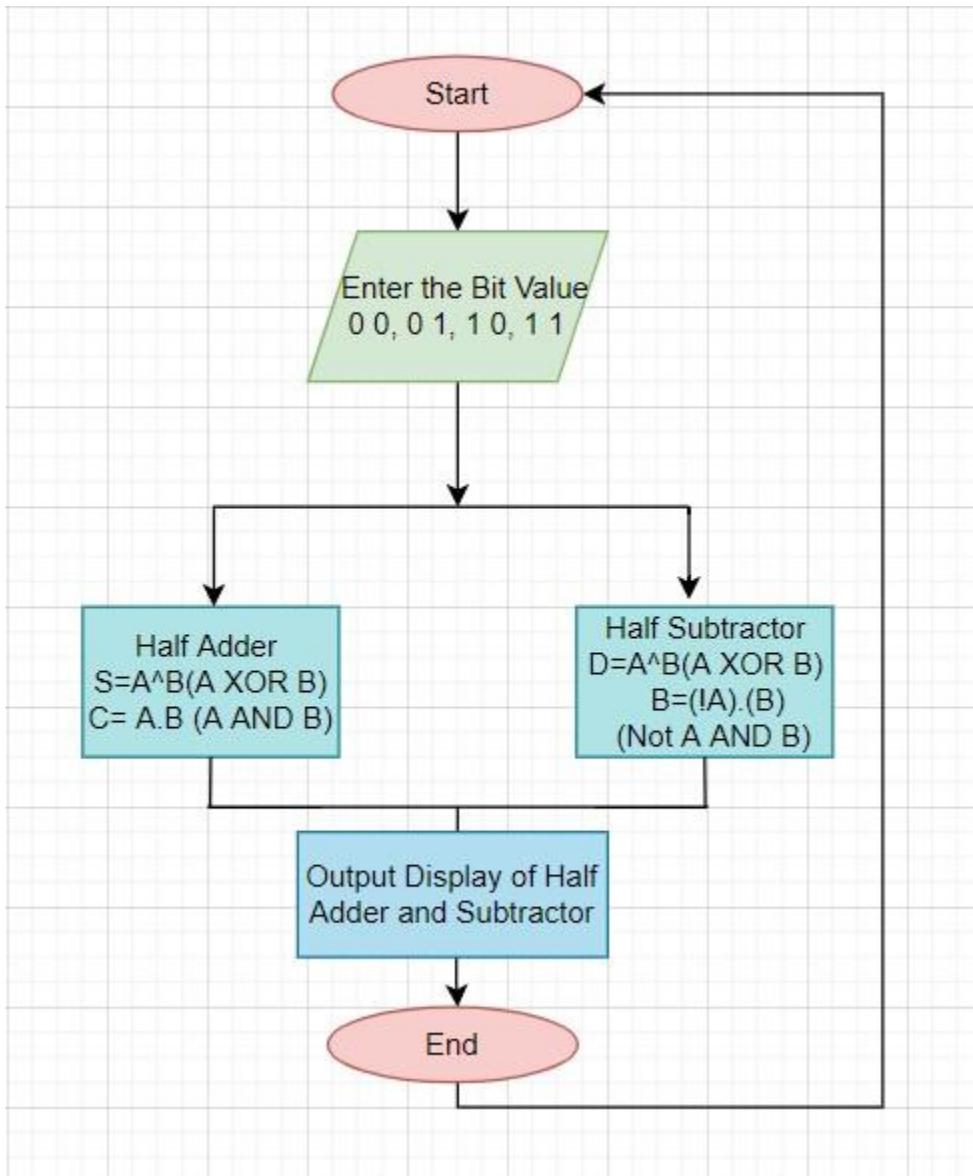


Application 5 To Display Half Adder and Subtractor

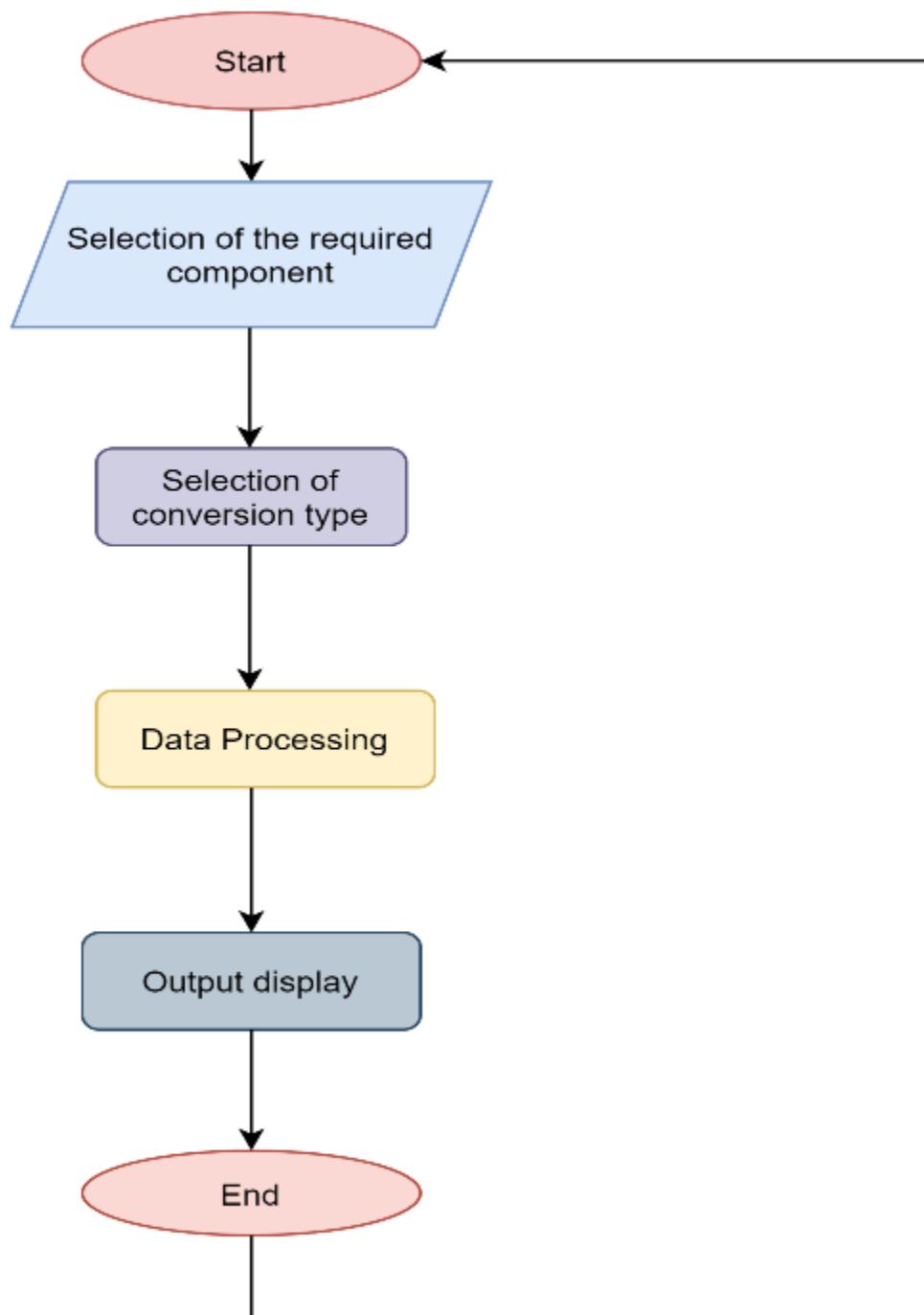
High Level Design



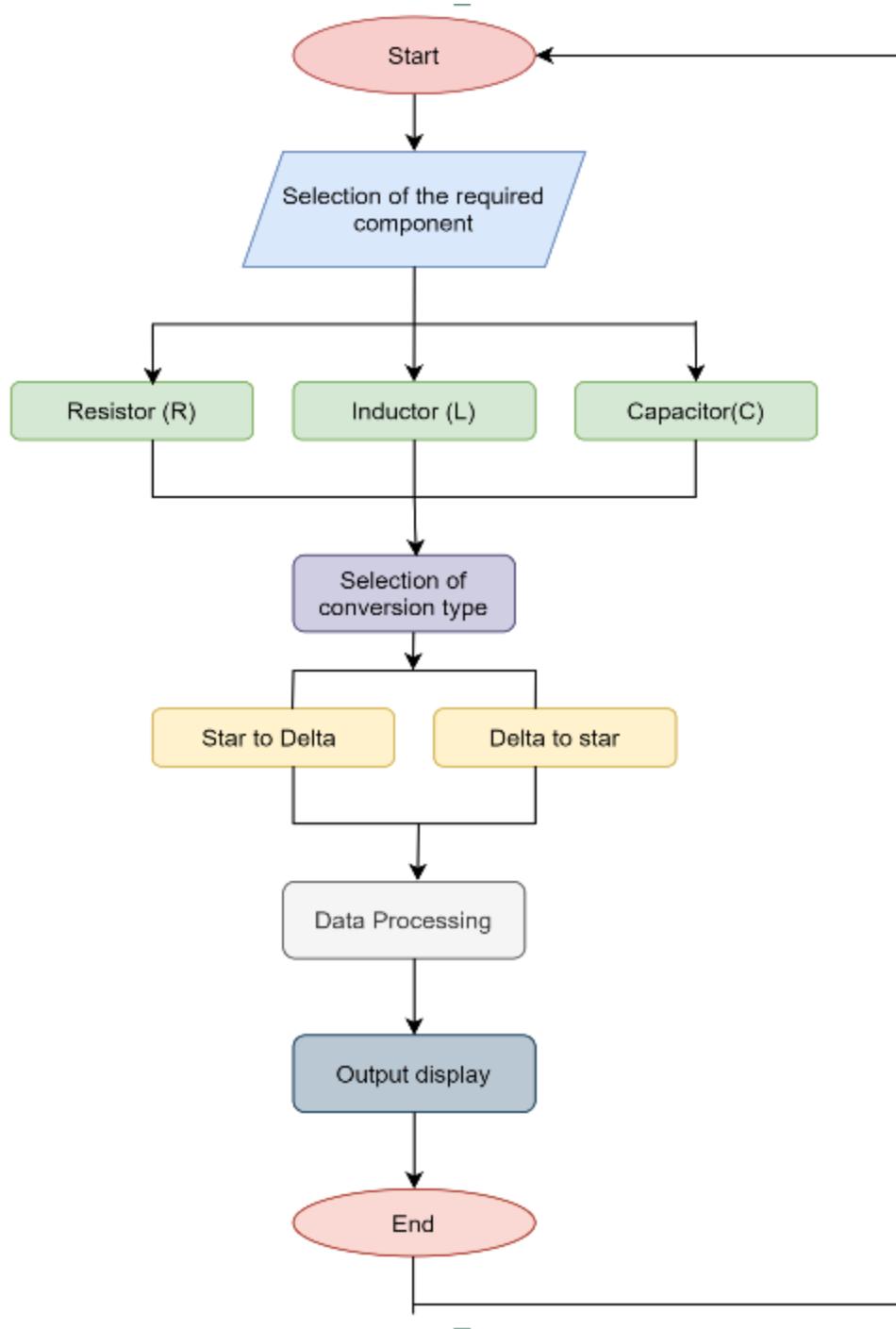
GENESIS Learning Report – Module Name
Low Level Design



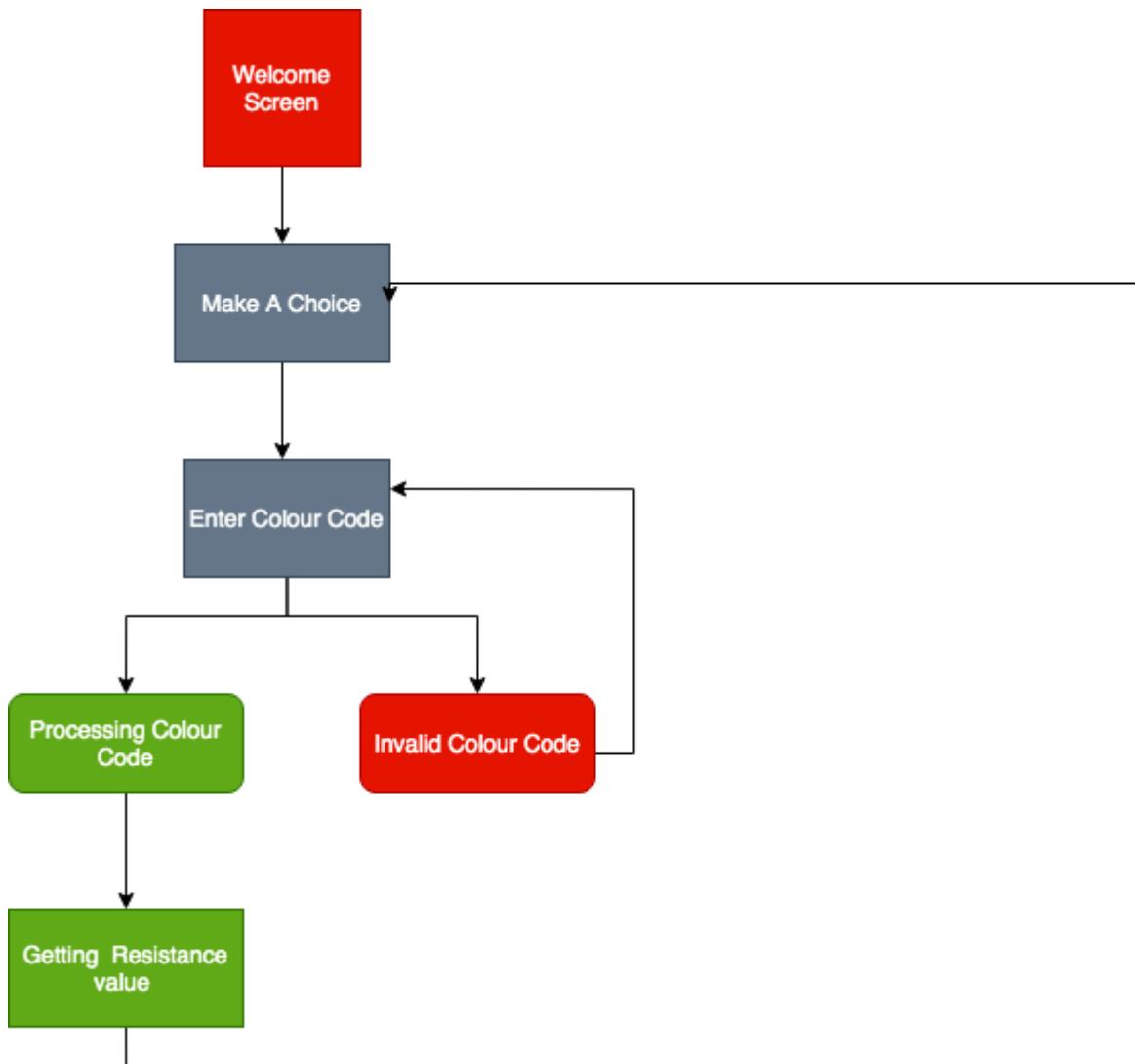
Application 6 - Star and Delta conversions



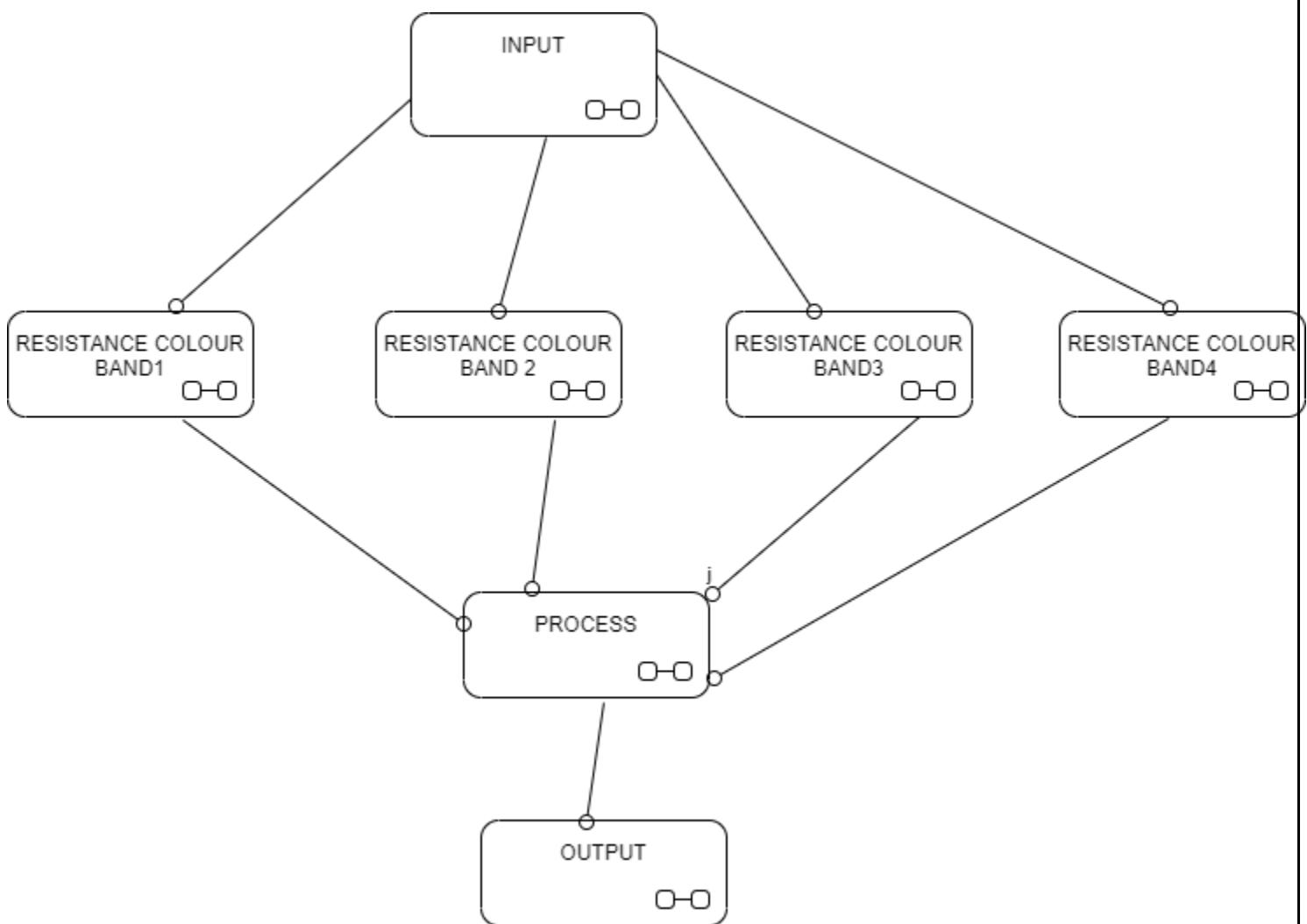
GENESIS Learning Report – Module Name
Low Level Design



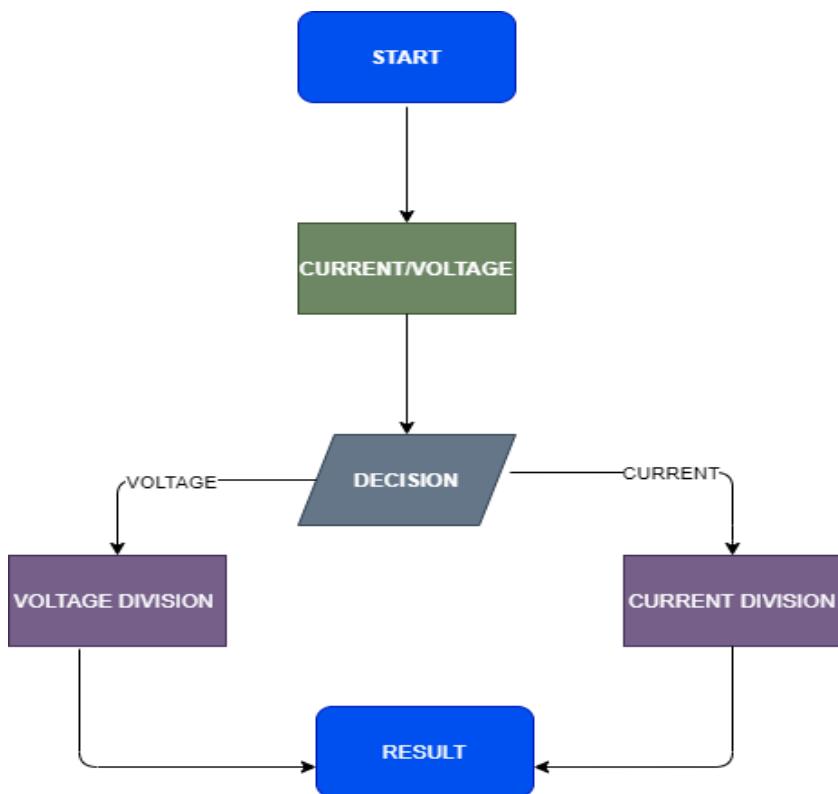
APPLICATION 7 - REGISTER COLOUR CODE



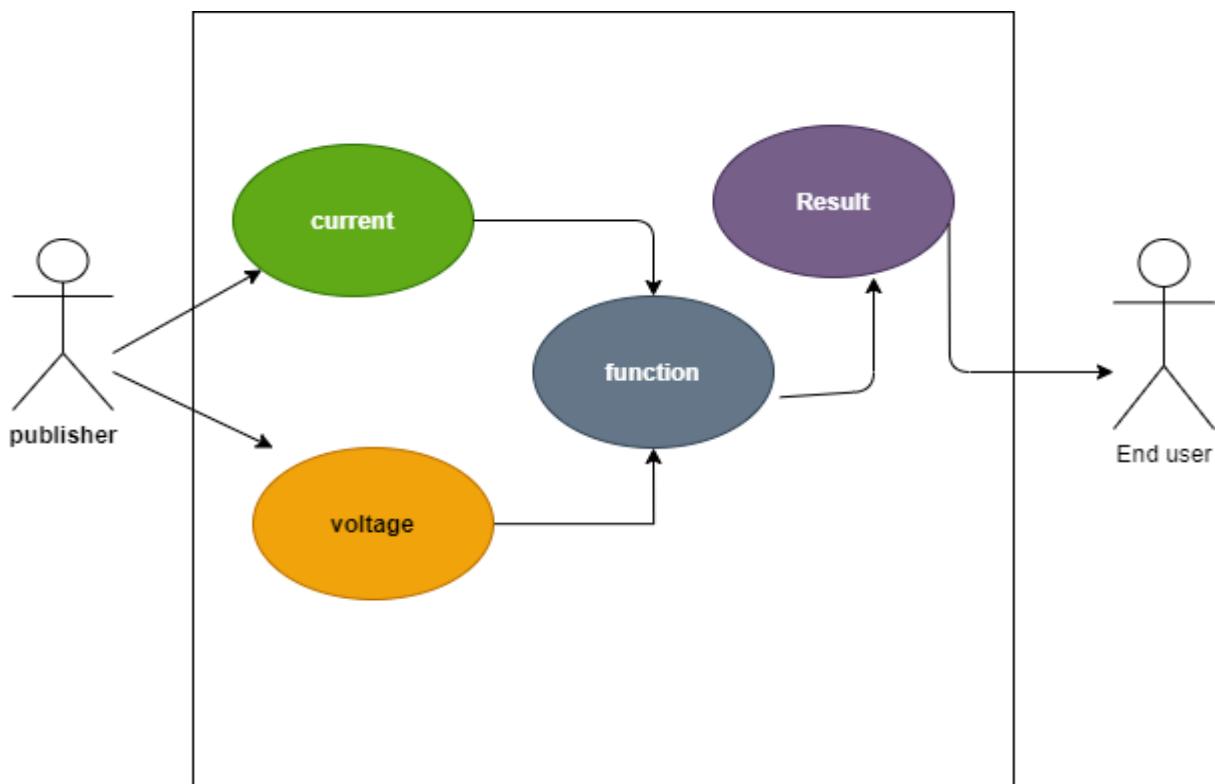
BEHAVIOURAL DIAGRAM



APPLICATION8 - CURRENT AND VOLTAGE DIVISION

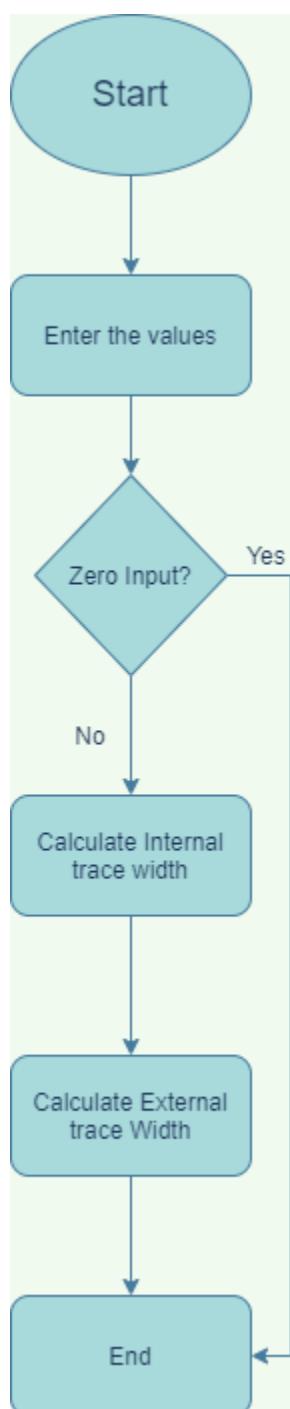


usecase diagram



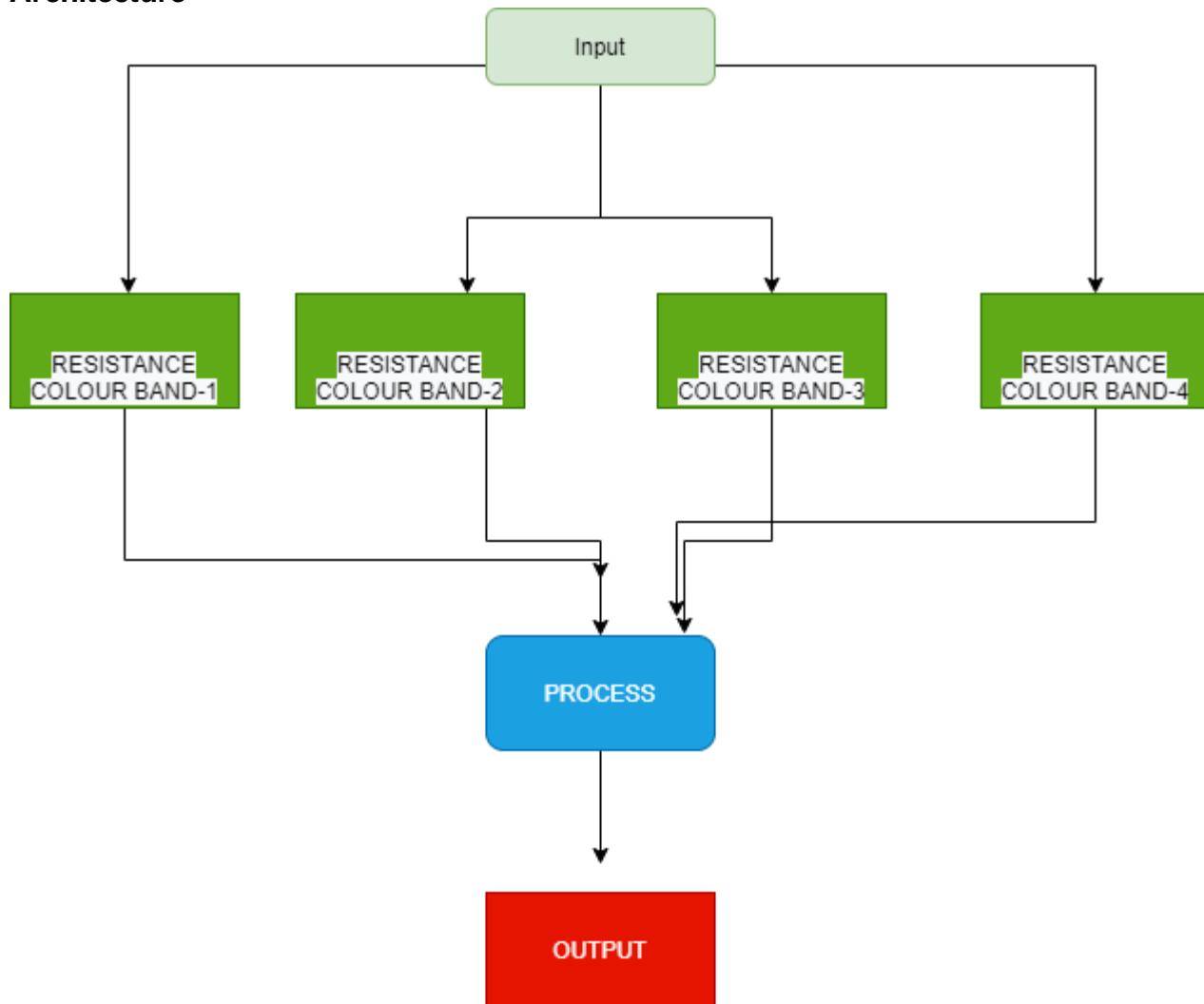
Application 9 - Calculating trace width of a PCB

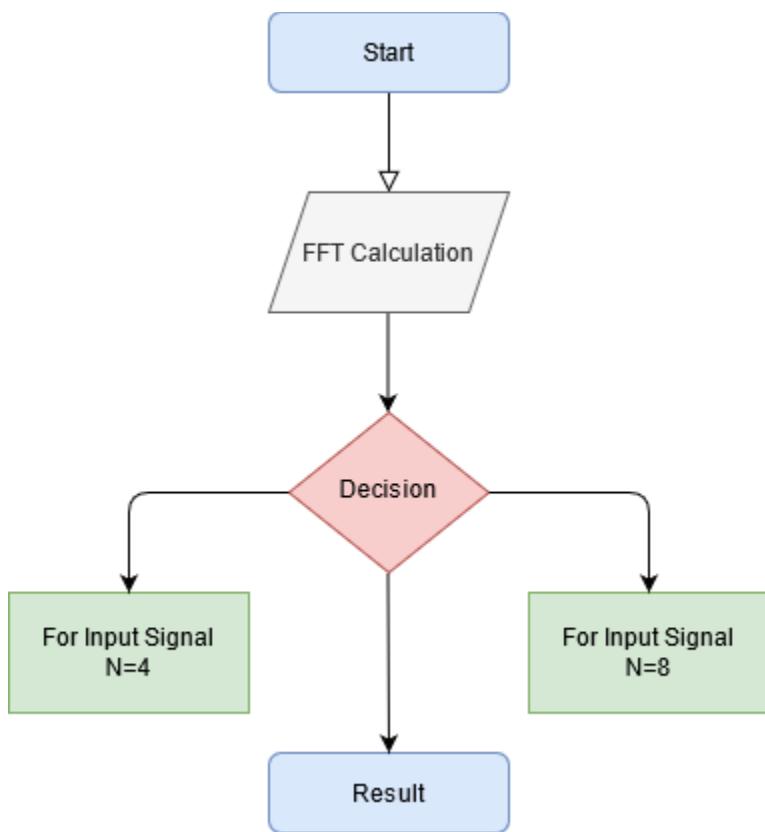
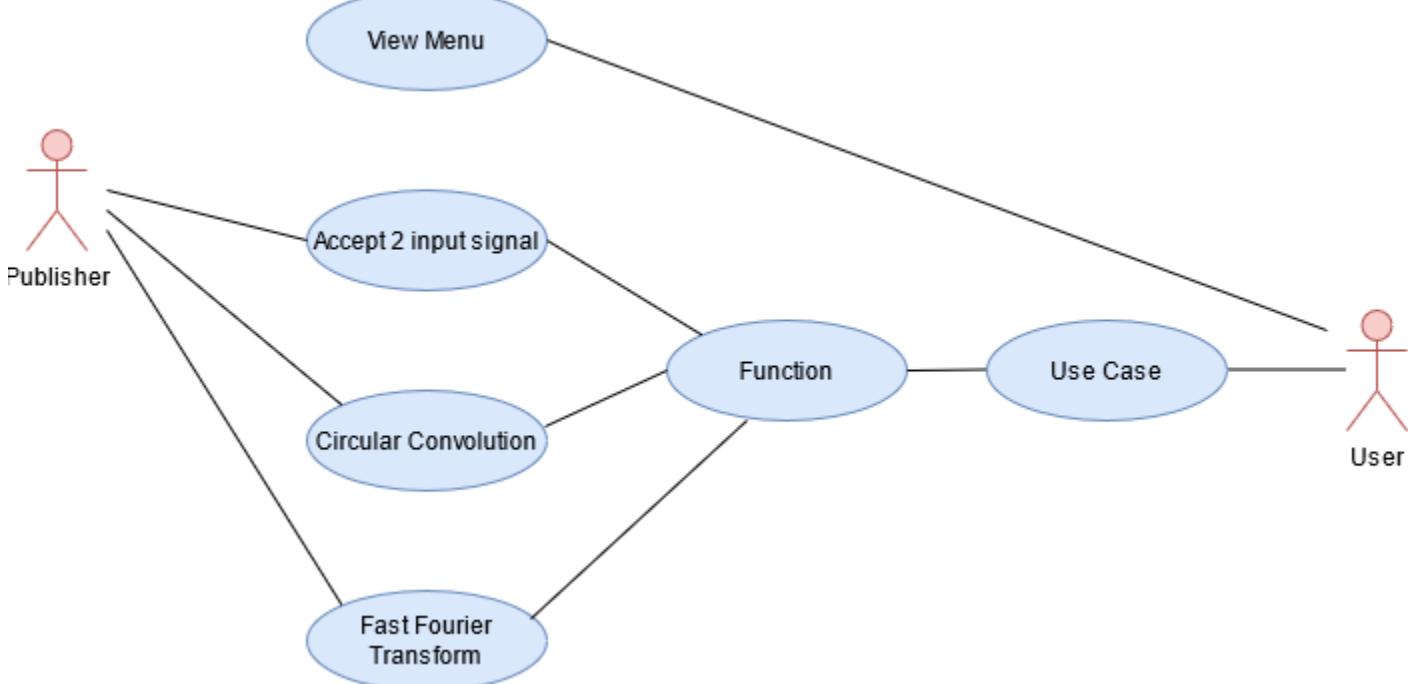
Architecture



Application 10 - FINDING REGISTER COLOR CODE

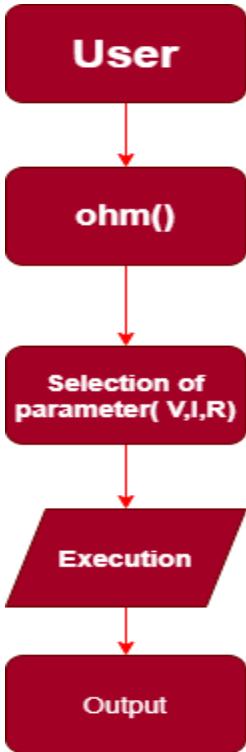
Architecture



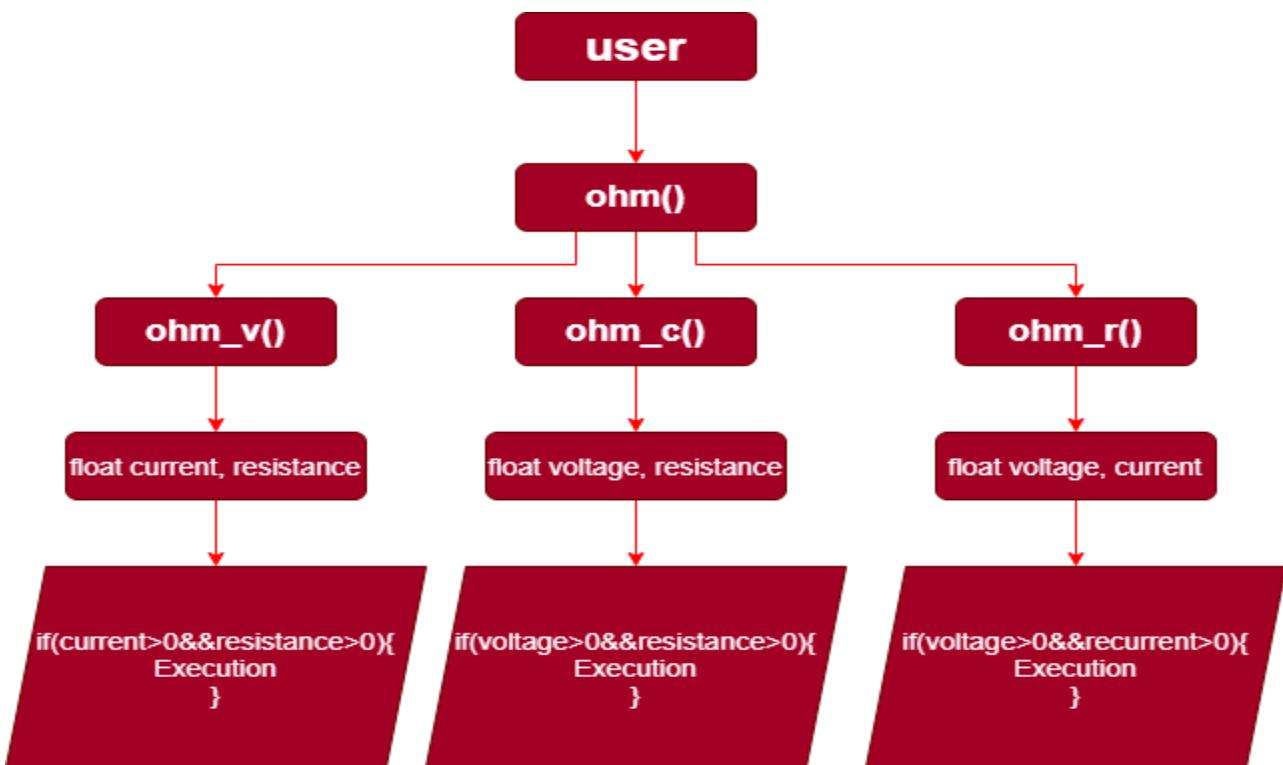
APPLICATION 16 - Calculating Circular Convolution**Architecture****Use Case**

GENESIS Learning Report – Module Name
Application 17 - Ohm's Law

High Level

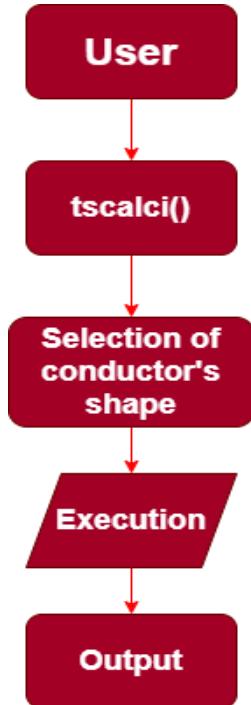


Architecture

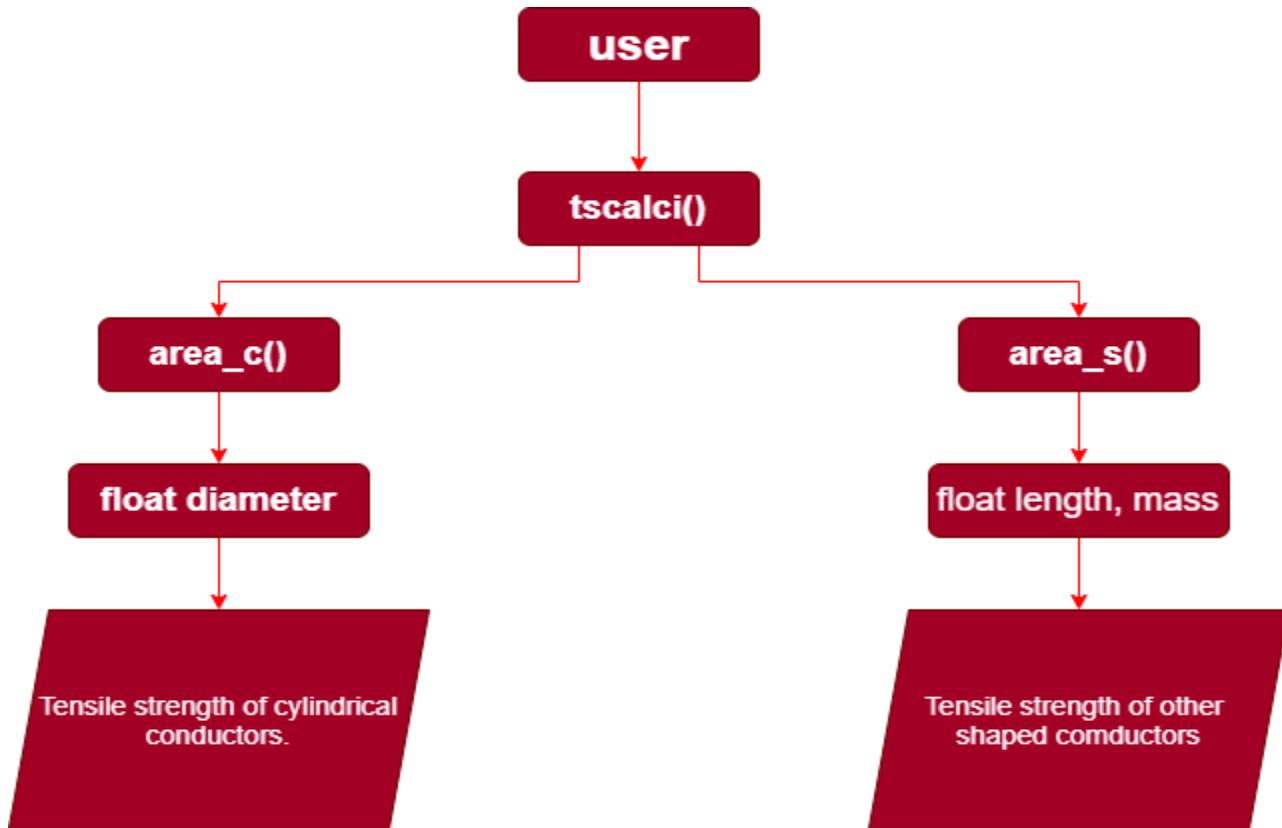


Application 18 - Findig Tensile Strength of Conductors

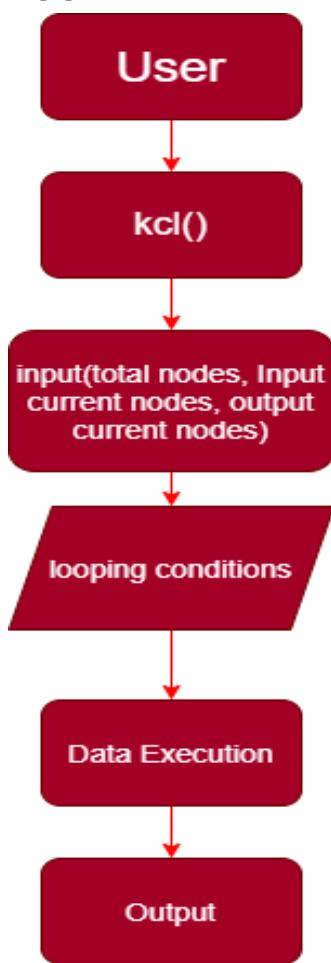
High Level



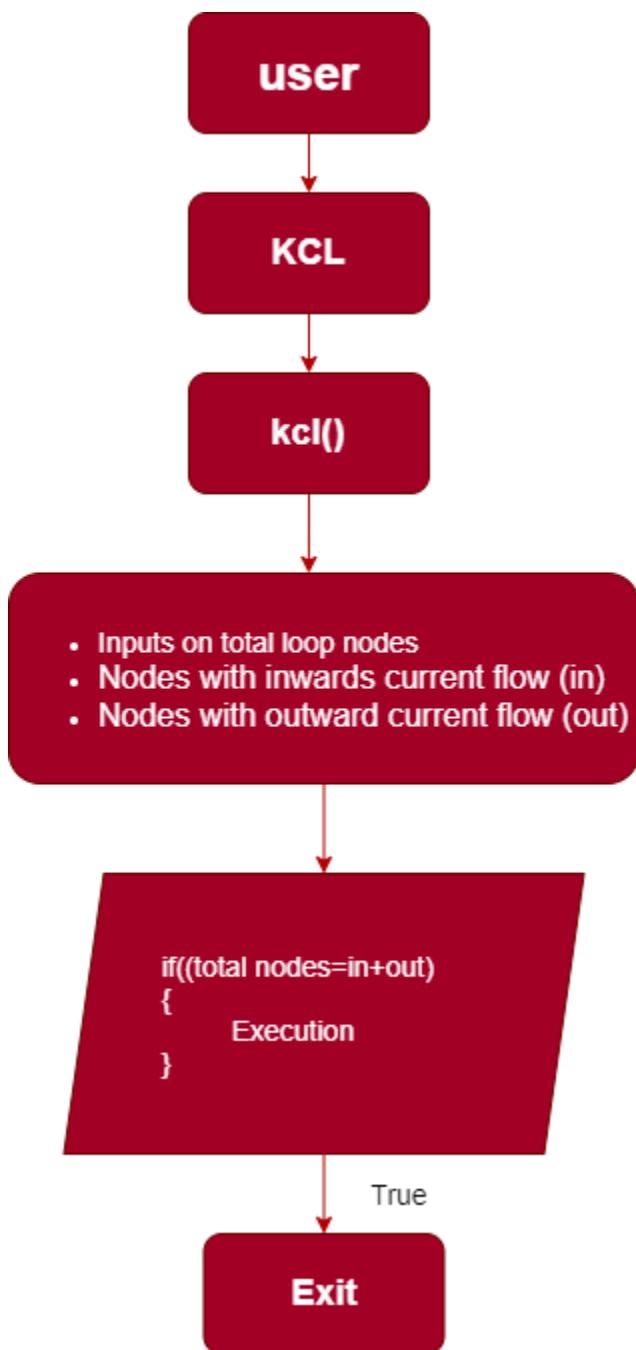
Architecture



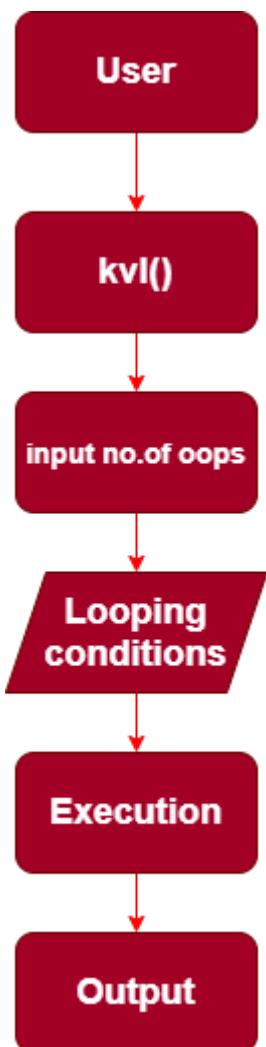
Application 19 - Kirchoff's Current Law



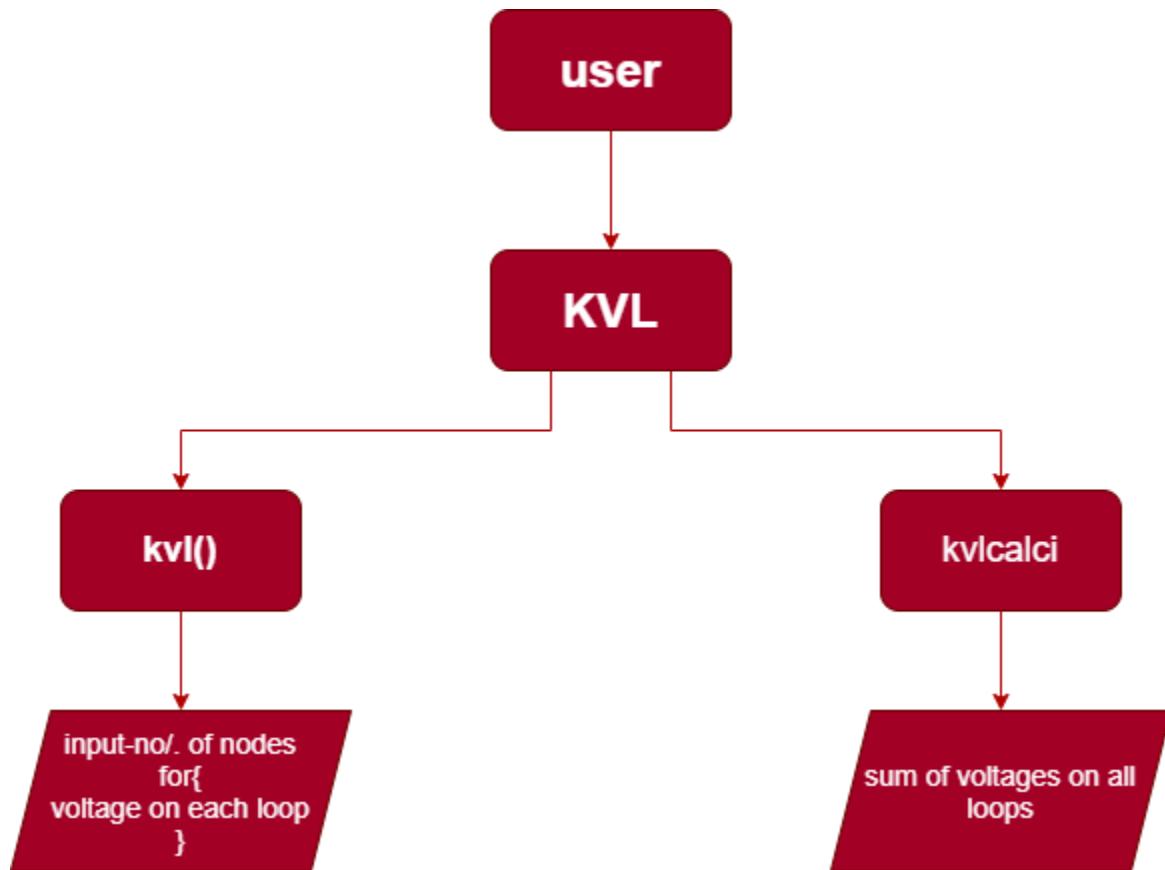
Architecture



GENESIS Learning Report – Module Name
Application 20 - Kirchoff's Voltage Law



Architecture



TEST PLAN:

Table no: High level test plan

Test ID	Description	Exp I/P	Exp O/P	Actual Out	Type Of Test
H_01	Circuit Calculator passive element parameters	Success	Success	Success	Application based
H_02	Calculator for Calculating the Duty Cycle and the output voltage of the waveform	Success	Success	Success	Application based
H_03	Calculator for Displaying the Half Adder and Subtractor (via Truth Table)	Success	Success	Success	Application based
H_04	Calculate the Different Machine Parameters	Success	Success	Success	Application based
H_05	Calculation to find Multi or Single core Wires	Success	Success	Success	Application based
H_06	Calculator for the conversion of star and delta	Success	Success	Success	Application based
H_07	Calculation of electrical parameters using ohm's law	Success	Success	Success	Application based
H_08	Kirchoff's voltage law	Success	Success	Success	Application based
H_09	Kirchoff's current law	Success	Success	Success	Application based
H_10	Tensile strength of conductors	Success	Success	Success	Application based

GENESIS Learning Report – Module Name
Table no: Low level test plan

Test ID	Description	Exp I/P	Exp O/P	Actual Out	Type Of Test
L_01	Calculate the Different Machine Parameters	Success	Success	Success	Requirement based
L_02	Calculate the Different Machine Parameters	Success	Success	Success	Scenario based
L_03	Calculate the Different Machine Parameters	Success	Success	Success	Boundary based
L_04	Calculation to find Multi or Single core Wires	Success	Success	Success	Requirement based
L_05	Calculation to find Multi or Single core Wires	Success	Success	Success	Scenario based
L_06	Calculation to find Multi or Single core Wires	Success	Success	Success	Boundary based
L_07	Calculation of Duty Cycle By giving input as Ton and Toff	Success	Success	Success	Requirement based
L_08	Calculation of output voltage with respect to given input voltage (v_0) and calculated Duty Cycle value	Success	Success	Success	Scenario based
L_09	Input of Duty Cycle should always be less than 1 as its a ratio it cant be $>=1$	Success	Success	Success	Boundary based
L_10	Display of the Half adder and Half Subtractor	Success	Success	Success	Requirement based
L_11	Operations like XOR operation and AND operation in the given input gives the output as Sum Carry Difference Borrow	Success	Success	Success	Scenario based
L_12	The Input Value has boundary constrain as it is Half adder and Subtractor binary input range is 0-4	Success	Success	Success	Boundary based
L_13	The passive elements required for the conversion are displayed	Success	Success	Success	Requirement based
L_14	Conversion of output (star/delta) for each element with respective inputs (delta/star)	Success	Success	Success	Scenario based
L_15	Finding Voltage from Voltage Division Circuit	Success	Success	Success	requirement based
L_16	Finding Current from Current Division Circuit	Success	Success	Success	requirement based
L_17	Passing current and resistance to find voltage using ohm's law	Success	Success	Success	requirement based
L_18	Passing voltage and resistance to find current using ohm's law	Success	Success	Success	requirement based
L_19	Passing voltage and current to find resistance using ohm's law	Success	Success	Success	requirement based
L_20	Electrical Parameters must not be negative	Success	Success	Success	Boundary based
L_21	Passing Length, Mass for finding tensile strength of shaped conductors	Success	Success	Success	requirement based
L_22	Passing diameter for finding tensile strength of cylindrical	Success	Success	Success	requirement

GENESIS Learning Report – Module Name

Test ID	Description	Exp I/P	Exp O/P	Actual Out	Type Of Test
	conductions				based
L_23	Parameter passed cannot be negative	Success	Success	Success	Boundary based
L_24	Number of total nodes must be passed to verify Kirchoff's current law	Success	Success	Success	requirement based
L_25	Number of total nodes where current is inwards towards circuits and away from circuit must be separately passed	Success	Success	Success	requirement based
L_26	Number of total nodes must be equal to nodes where current flows towards circuits and away from circuit	Success	Success	Success	Boundary based
L_27	Number of loops in the given system must be passed	Success	Success	Success	Requirement based
L_28	Voltage values of each loops must be passed	Success	Success	Success	Requirement based

Screenshots of output and test cases

Image 1

The screenshot shows a Visual Studio Code interface with the following details:

- File Explorer:** Shows a project structure under 'TANMAYA191-SDLC_8_THUNDERBIRDS' containing .github, .vscode, 1_Requirements, 2_Design, 3_Implementation (selected), 4_TestPlan, 5_Daily_Update, .gitignore, LICENSE, and README.md.
- Terminal:** Running Windows PowerShell. The session starts with a 'make' command in the directory H:\Github\tanmaya191-SDLC_8_Thunderbirds\3_Implementation. It then runs 'make run'. The output shows the program 'WELCOME' followed by a list of 16 calculation options numbered 1 to 16. The user types '11' and the program responds with '--PCB trace width--' and 'Welcome to Trace Width Calculator'. It prompts for 'Enter value of Thickness' and 'Enter value of temperature rise' both set to 5.
- Status Bar:** Shows 'Ln 25, Col 41' and other system information like date and time.
- Bottom Bar:** Includes a search bar, taskbar icons for Google Chrome, File Explorer, Task View, and VS Code, and system status icons.

GENESIS Learning Report – Module Name

Image 2

```

File Edit Selection View Go Run Terminal Help main.c - tanmaya191-SDLC_8_Thunderbirds - Visual Studio Code
EXPLORER PROBLEMS OUTPUT TERMINAL DEBUG CONSOLE
powershell + x
resistance_calculator.c main.c -Iinc -Itest -o build\Thunderbirds.exe -lm
PS H:\Github\tanmaya191-SDLC_8_Thunderbirds\3_Implementation> make test
gcc src/freq.c src/parameter.c src/power_input.c src/power_output.c src/resistance.c src/secondary_turns.c src/speed.c src/torque.c src/load_current.c src/Factor_s.c src/parallel.c src/series.c src/halfadder.c src/halbsubtractor.c src/DutyCycle.c src/kVL.c src/kCL.c src/ohm.c src/tensiles.c src/trace_width.c src/starDeltaConversion.c src/division.c src/CircularConvolution.c src/resistance_calculator.c test/test_Thunderbirds.c test/unity.c -Iinc -Itest -o build\Test_Thunderbirds.exe -lm
build\Test_Thunderbirds.exe
test/test_Thunderbirds.c:244:test_power_input:PASS
test/test_Thunderbirds.c:245:test_power_output:PASS
test/test_Thunderbirds.c:246:test_resistance:PASS
test/test_Thunderbirds.c:247:test_freq:PASS
test/test_Thunderbirds.c:248:test_speed:PASS
test/test_Thunderbirds.c:249:test_torque:PASS
test/test_Thunderbirds.c:250:test_secondary_turns:PASS
test/test_Thunderbirds.c:251:test_load_current:PASS
test/test_Thunderbirds.c:252:test_series_capacitance:PASS
test/test_Thunderbirds.c:253:test_series_inductance:PASS
test/test_Thunderbirds.c:254:test_parallel_capacitance:PASS
test/test_Thunderbirds.c:255:test_parallel_inductance:PASS
test/test_Thunderbirds.c:256:test_series_resistance:PASS
test/test_Thunderbirds.c:257:test_parallel_resistance:PASS
test/test_Thunderbirds.c:261:test_halfadd1:PASS
test/test_Thunderbirds.c:262:test_halfadd2:PASS
test/test_Thunderbirds.c:263:test_halbsub1:PASS
test/test_Thunderbirds.c:264:test_halbsub2:PASS
test/test_Thunderbirds.c:265:test_Dutycycle:PASS
test/test_Thunderbirds.c:266:test_VoltageOutputofDutyCycle:PASS
test/test_Thunderbirds.c:269:testohm_v:PASS
test/test_Thunderbirds.c:270:testohm_c:PASS
test/test_Thunderbirds.c:271:testohm_r:PASS
test/test_Thunderbirds.c:275:test_trace_width_external:PASS
test/test_Thunderbirds.c:276:test_trace_width_internal:PASS

-----
25 Tests 0 Failures 0 Ignored
OK
PS H:\Github\tanmaya191-SDLC_8_Thunderbirds\3_Implementation> make

```

Ln 25, Col 41 Spaces: 4 UTF-8 CRLF C Win32

Type here to search

Image 3

```

File Edit Selection View Go Run Terminal Help main.c - tanmaya191-SDLC_8_Thunderbirds - Visual Studio Code
EXPLORER PROBLEMS OUTPUT TERMINAL DEBUG CONSOLE
powershell + x
2. Wire Selection
3. Star Delta Conversion
4. Ohms law
5. KVL
6. KCL
7. Tensile calculator
8. DutyCycle
9. Adder
10. Subtractor
11. PCB Trace Width
12. Division
13. Parallel
14. Series
15. Resistor colour code
16. Circular convolution
Type : 1
-----Motor Parameters-----
1.power_input
2.power_output
3.resistance
4.freq
5.speed
6.torque
7.secondary_turns
Enter the parameter: 1
Enter the value of voltage and current:
23 24
The Input Power of the machine is : 552
PS H:\Github\tanmaya191-SDLC_8_Thunderbirds\3_Implementation> make run
gcc src/freq.c src/parameter.c src/power_input.c src/power_output.c src/resistance.c src/secondary_turns.c src/speed.c src/torque.c src/load_current.c src/Factor_s.c src/parallel.c src/series.c src/halfadder.c src/halbsubtractor.c src/DutyCycle.c src/kVL.c src/kCL.c src/ohm.c src/tensiles.c src/trace_width.c src/starDeltaConversion.c src/division.c src/CircularConvolution.c src/resistance_calculator.c main.c -Iinc -Itest -o build\Thunderbirds.exe -lm
build\Thunderbirds.exe
-----WELCOME-----
Select the type of calculation

```

Ln 25, Col 41 Spaces: 4 UTF-8 CRLF C Win32

Type here to search

GENESIS Learning Report – Module Name

Image 4

```

File Edit Selection View Go Run Terminal Help main.c - tanmaya191-SDLC_8_Thunderbirds - Visual Studio Code
EXPLORER PROBLEMS OUTPUT TERMINAL DEBUG CONSOLE powershell + - X
TANMAYA191-SDLC_8_THUNDERBIRDS
.github
.vscode
1_Requirements
2_Design
Readme.md
trace_width.png
3_Implementation
4_TestPlan
5_Daily_Update
Readme.md
.gitignore
LICENSE
README.md

11. PCB Trace Width
12. Division
13. Parallel
14. Series
15. Resistor colour code
16. Circular convolution
-----Wire selection-----
*****The selection of wire Depent on three Factors*****
~~~~~Total Load~~~~~
~~~~~Total Current~~~~~
~~~~~Voltage Drop~~~~~
The type is:2
++If the voltage is in range of 420-440 for industrial purposes++
Enter the value of Power and Voltage
74 25
The selection of wire on the bases of Load_current = 2
The load is too high such wire are not available in market
PS H:\Github\tanmaya191-SDLC_8_Thunderbirds\3_Implementation> make run
gcc src/freq.c src/parameter.c src/power_input.c src/power_output.c src/resistance.c src/secondary_turns.c src/speed.c src/torque.c src/Load_current.c src/Factor_s.c src/parallel.c src/series.c src/halfadder.c src/half_subtractor.c src/DutyCycle.c src/kv1.c src/kcl.c src/ohm.c src/tensiles.c src/trace_width.c src/starDeltaConversion.c src/division.c src/CircularConvolution.c src/resistance_calculator.c main.c -Iinc -Itest -o build\Thunderbirds.exe -lm
build\Thunderbirds.exe
-----WELCOME-----
Select the type of calculation
Ln 25, Col 41 Spaces: 4 UTF-8 CRLF C Win32 F Q
Type here to search
15:15 24-08-2021

```

Image 5

```

File Edit Selection View Go Run Terminal Help main.c - tanmaya191-SDLC_8_Thunderbirds - Visual Studio Code
EXPLORER Makefile main.c M X powershell + - X ...
TANMAYA191-SDLC_8_THUNDERBIRDS
.github
.vscode
1_Requirements
2_Design
Readme.md
trace_width.png
3_Implementation
4_TestPlan
5_Daily_Update
Readme.md
.gitignore
LICENSE
README.md

3_Implementation > C main.c > main()
19 |     printf("4. Ohms law\n"); //Viswak
20 |     printf("5. KVl\n"); //Viswak
21 |     printf("6. KCL\n"); //Viswak
22 |     printf("7. Tensile_calculator\n"); //Viswak
PROBLEMS OUTPUT TERMINAL DEBUG CONSOLE
11. PCB Trace Width
12. Division
13. Parallel
14. Series
15. Resistor colour code
16. Circular convolution
Type : 3
-----Star and Delta Conversion-----
Select the component required
-----
2-Inductor
3-Capacitor
Select the component: 2
Select the transformation
-----
1.Star-->Delta conversion
2.Delta-->Star conversion
-----
The transformation is: 2
Enter the values of delta inductors
L_a= 5
L_b= 8
L_c= 9
The equivalent star config. are
L_1= 1.82 mH
L_2= 3.27 mH
L_3= 2.05 mH
PS H:\Github\tanmaya191-SDLC_8_Thunderbirds\3_Implementation> make run
gcc src/freq.c src/parameter.c src/power_input.c src/power_output.c src/resistance.c src/secondary_turns.c src/speed.c src/torque
Ln 25, Col 41 Spaces: 4 UTF-8 CRLF C Win32 F Q
Type here to search
15:14 24-08-2021

```

GENESIS Learning Report – Module Name

Image 6

```

File Edit Selection View Go Run Terminal Help main.c - tanmaya191-SDLC_8_Thunderbirds - Visual Studio Code
EXPLORER TANMAYA191-SDLC_8 THUNDERBIRDS .github .vscode 1_Requirements 2_Design Readme.md trace_width.png 3_Implementation 4_TestPlan 5_Daily_Update Readme.md .gitignore LICENSE README.md
Makefile main.c M X
3_Implementation > C main.c > main()
19 printf("4. Ohms law\n"); //Viswak
20 printf("5. KVLn"); //Viswak
21 printf("6. KCL\n"); //Viswak
22 printf("7. Tensile calculator\n"); //Viswak
PROBLEMS OUTPUT TERMINAL DEBUG CONSOLE
powerShell + x ^ x
14. Series
15. Resistor colour code
16. Circular convolution
Type : 4
<-----Ohms law-----
*****Calculation of Voltage, Current, Resistance using Ohm's Law*****
Parameter to be found:
1. Voltage
2. Current
3. Resistance
Enter the option : 2

Enter the voltage value : 58

Enter the resistance value : 6
current : 9.666667
PS H:\Github\tanmaya191-SDLC_8_Thunderbirds\3_Implementation> make run
gcc src/freq.c src/parameter.c src/power_input.c src/power_output.c src/resistance.c src/secondary_turns.c src/speed.c src/torque.c src/load_current.c src/Factor_s.c src/parallel.c src/series.c src/halfadder.c src/halfsubtracter.c src/DutyCycle.c src/kvl.c src/kcl.c src/ohm.c src/tensiles.c src/trace_width.c src/starDeltaConversion.c src/division.c src/CircularConvolution.c src/resistance_calculator.c main.c -Iinc -Itest -o build\Thunderbirds.exe -lm
build\Thunderbirds.exe
-----WELCOME-----
Select the type of calculation
1. Motor parameters
2. Wire Selection
3. Star Delta Conversion
4. Ohms law
Ln 25, Col 41 Spaces: 4 UTF-8 CRLF C Win32 ⚡ Q
Type here to search
15:14 24-08-2021

```

Image 7

```

File Edit Selection View Go Run Terminal Help main.c - tanmaya191-SDLC_8_Thunderbirds - Visual Studio Code
EXPLORER TANMAYA191-SDLC_8 THUNDERBIRDS .github .vscode 1_Requirements 2_Design Readme.md trace_width.png 3_Implementation 4_TestPlan 5_Daily_Update Readme.md .gitignore LICENSE README.md
Makefile main.c M X
3_Implementation > C main.c > main()
19 printf("4. Ohms law\n"); //Viswak
20 printf("5. KVLn"); //Viswak
21 printf("6. KCL\n"); //Viswak
22 printf("7. Tensile calculator\n"); //Viswak
PROBLEMS OUTPUT TERMINAL DEBUG CONSOLE
powerShell + x ^ x
16. Circular convolution
Type : 5
<-----KVL-----
Kirchoff Voltage Law Calculation
Enter the number of loops : 5

Enter the voltage in loop1 : 6
Enter the voltage in loop2 : 8
Enter the voltage in loop3 : 9
Enter the voltage in loop4 : 4

Enter the voltage in loop5 : 6
The voltage is 33.00
PS H:\Github\tanmaya191-SDLC_8_Thunderbirds\3_Implementation> make run
gcc src/freq.c src/parameter.c src/power_input.c src/power_output.c src/resistance.c src/secondary_turns.c src/speed.c src/torque.c src/load_current.c src/Factor_s.c src/parallel.c src/series.c src/halfadder.c src/halfsubtracter.c src/DutyCycle.c src/kvl.c src/kcl.c src/ohm.c src/tensiles.c src/trace_width.c src/starDeltaConversion.c src/division.c src/CircularConvolution.c src/resistance_calculator.c main.c -Iinc -Itest -o build\Thunderbirds.exe -lm
build\Thunderbirds.exe
-----WELCOME-----
Select the type of calculation
1. Motor parameters
2. Wire Selection
3. Star Delta Conversion
4. Ohms law
Ln 25, Col 41 Spaces: 4 UTF-8 CRLF C Win32 ⚡ Q
Type here to search
15:14 24-08-2021

```

GENESIS Learning Report – Module Name

Image 8

```

main.c - tanmaya191-SDLC_8_Thunderbirds - Visual Studio Code

File Edit Selection View Go Run Terminal Help
main.c - tanmaya191-SDLC_8_Thunderbirds - Visual Studio Code

EXPLORER TANMAYA191-SDLC_8_THUNDERBIRDS .github .vscode 1_Requirements 2_Design README.md trace_width.png 3_Implementation 4_TestPlan 5_Daily_Update Readme.md .gitignore LICENSE README.md

Makefile main.c M X
3_Implementation > C main.c > main()
19 printf("4. Ohms law\n"); //Viswak
20 printf("5. KVl\n"); //Viswak
21 printf("6. KCL\n"); //Viswak
22 printf("7. Tensile_calculator\n"); //Viswak

PROBLEMS OUTPUT TERMINAL DEBUG CONSOLE
powershell + x

15. Resistor colour code
16. Circular convolution
-----KCL-----
*****Kirchoff's current law*****
Enter the number of nodes: 3
Enter the number of nodes where current flows inwards: 2
Enter the number of nodes where current flows outwards: 1
Inward Current1 : 4
Inward Current2 : 6
Outward Current1 : 2

Voltage entering the junction : 10.00
Voltage leaving the junction : 2.00
PS H:\Github\tanmaya191-SDLC_8_Thunderbirds\3_Implementation> make run
gcc src/freq.c src/parameter.c src/power_input.c src/power_output.c src/resistance.c src/secondary_turns.c src/speed.c src/torque.c src/Load_current.c src/Factor_s.c src/parallel.c src/series.c src/halfadder.c src/halfsubtracter.c src/DutyCycle.c src/kvl.c src/kcl.c src/ohm.c src/tensiles.c src/trace_width.c src/starDeltaConversion.c src/division.c src/CircularConvolution.c src/resistance_calculator.c main.c -Iinc -Itest -o build\Thunderbirds.exe -lm
build\Thunderbirds.exe
-----WELCOME-----
Select the type of calculation
1. Motor parameters
2. Wire Selection
-----tensileCalci-----
Type : 7
Tensile test of conductors
1. Cylindrical conductors
2. Shaped conductors
Enter the option:1
Enter the diameter:5
The area of the conductor is:19.625000
Enter the value of breaking load:
The tensile strength of the conductor is:0.46
PS H:\Github\tanmaya191-SDLC_8_Thunderbirds\3_Implementation> make run
gcc src/freq.c src/parameter.c src/power_input.c src/power_output.c src/resistance.c src/secondary_turns.c src/speed.c src/torque.c src/Load_current.c src/Factor_s.c src/parallel.c src/series.c src/halfadder.c src/halfsubtracter.c src/DutyCycle.c src/kvl.c src/kcl.c src/ohm.c src/tensiles.c src/trace_width.c src/starDeltaConversion.c src/division.c src/CircularConvolution.c src/resistance_calculator.c main.c -Iinc -Itest -o build\Thunderbirds.exe -lm
build\Thunderbirds.exe
-----WELCOME-----
Select the type of calculation
1. Motor parameters
2. Wire Selection
3. Star Delta Conversion
4. Ohms law
-----
```

Ln 25, Col 41 Spaces: 4 UTF-8 CRLF C Win32 15:14 24-08-2021

Image 9

```

main.c - tanmaya191-SDLC_8_Thunderbirds - Visual Studio Code

File Edit Selection View Go Run Terminal Help
main.c - tanmaya191-SDLC_8_Thunderbirds - Visual Studio Code

EXPLORER TANMAYA191-SDLC_8_THUNDERBIRDS .github .vscode 1_Requirements 2_Design README.md trace_width.png 3_Implementation 4_TestPlan 5_Daily_Update Readme.md .gitignore LICENSE README.md

Makefile main.c M X
3_Implementation > C main.c > main()
19 printf("4. Ohms law\n"); //Viswak
20 printf("5. KVl\n"); //Viswak
21 printf("6. KCL\n"); //Viswak
22 printf("7. Tensile_calculator\n"); //Viswak

PROBLEMS OUTPUT TERMINAL DEBUG CONSOLE
powershell + x

14. Series
15. Resistor colour code
16. Circular convolution
Type : 7
-----tensileCalci-----
Tensile test of conductors
1. Cylindrical conductors
2. Shaped conductors
Enter the option:1
Enter the diameter:5
The area of the conductor is:19.625000
Enter the value of breaking load:
The tensile strength of the conductor is:0.46
PS H:\Github\tanmaya191-SDLC_8_Thunderbirds\3_Implementation> make run
gcc src/freq.c src/parameter.c src/power_input.c src/power_output.c src/resistance.c src/secondary_turns.c src/speed.c src/torque.c src/Load_current.c src/Factor_s.c src/parallel.c src/series.c src/halfadder.c src/halfsubtracter.c src/DutyCycle.c src/kvl.c src/kcl.c src/ohm.c src/tensiles.c src/trace_width.c src/starDeltaConversion.c src/division.c src/CircularConvolution.c src/resistance_calculator.c main.c -Iinc -Itest -o build\Thunderbirds.exe -lm
build\Thunderbirds.exe
-----WELCOME-----
Select the type of calculation
1. Motor parameters
2. Wire Selection
3. Star Delta Conversion
4. Ohms law
-----
```

Ln 25, Col 41 Spaces: 4 UTF-8 CRLF C Win32 15:14 24-08-2021

GENESIS Learning Report – Module Name **Image 10**

File Edit Selection View Go Run Terminal Help main.c - tanmaya191-SDLC_8_Thunderbirds - Visual Studio Code

EXPLORER ...

TANMAYA191-SDLC_8_THUNDERBIRDS

- > .github
- > .vscode
- > 1_Requirements
- > 2_Design
 - ① Readme.md
 - trace_width.png
 - > 3_Implementation
 - 4_TestPlan
 - 5_Daily_Update
 - ① Readme.md
 - .gitignore
 - LICENSE
 - README.md

Makefile main.c M X

3_Implementation > C main.c > main()

```
19 printf("4. Ohms law\n"); //Viswak
20 printf("5. KVL\n"); //Viswak
21 printf("6. KCL\n"); //Viswak
22 printf("7. Tensile calculator\n"); //Viswak
```

PROBLEMS OUTPUT TERMINAL DEBUG CONSOLE

powerShell + x

15. Resistor colour code
16. Circular convolution
Type : 8

-----DutyCycle-----

1.Duty_cycle
2.Output_voltage
Enter your choice:2
Note That Duty Cycle is a Ratio so its Value should be less than 1:
Enter the Value of D (Duty cycle):
Enter the Value of Input Voltage (V0):
0.4
6
Voltage output of the waveform is:4.00000
PS H:\Github\tanmaya191-SDLC_8_Thunderbirds\3_Implementation> make run
gcc src/freq.c src/parameter.c src/power_input.c src/power_output.c src/resistance.c src/secondary_turns.c src/speed.c src/torque.c src/load_current.c src/Factor_s.c src/parallel.c src/series.c src/halfadder.c src/halfsubtracter.c src/DutyCycle.c src/kvl.c src/kcl.c src/ohm.c src/tensiles.c src/trace_width.c src/starDeltaConversion.c src/division.c src/CircularConvolution.c src/resistance_calculator.c main.c -Iinc -Itest -o build\Thunderbirds.exe

-----WELCOME-----

Select the type of calculation

1. Motor parameters
2. Wire Selection
3. Star Delta Conversion
4. Ohms law
5. KVL
6. KCL
7. Tensile calculator

main* 0 △ 0 Type here to search Ln 25, Col 41 Spaces: 4 UTF-8 CRLF C Win32

Image 11

The screenshot shows a Visual Studio Code interface with the following details:

- File Explorer:** Shows a project structure for "TANMAYA191-SDLC_8_THUNDERBIRDS" containing files like .github, .vscode, 1_Requirements, 2_Design, 3_Implementation, 4_TestPlan, 5_Daily_Update, Readme.md, trace_width.png, .gitignore, LICENSE, and README.md.
- Terminal:** The terminal window displays the command "make run" being executed in the directory "H:\Github\tanmaya191-SDLC_8_Thunderbirds\3_Implementation". The output shows the program running through several source files including freq.c, parameter.c, power_input.c, power_output.c, resistance.c, secondary_turns.c, speed.c, torque.c, load_current.c, factor_s.c, parallel.c, series.c, halfadder.c, halfsubtractor.c, DutyCycle.c, kvl.c, ohm.c, tensile.c, trace_width.c, starDeltaConversion.c, division.c, circularConvolution.c, and resistance_calculator.c. It also shows the main.c file being compiled with -Iinc and -Isrc options, linking build\Thunderbirds.exe, and finally building Thunderbirds.exe.
- Code Editor:** The main editor area shows the "main.c" file with code related to Ohms law, KVL, KCL, and a tensile calculator. A cursor is positioned at line 19. Below the editor are tabs for PROBLEMS, OUTPUT, TERMINAL, and DEBUG CONSOLE.
- Status Bar:** The status bar at the bottom shows the current file is "main.c", the line and column are "Ln 25, Col 41", the encoding is "UTF-8", and the file was last modified on "24-08-2021 15:14".

GENESIS Learning Report – Module Name

Image 12

The screenshot shows a Visual Studio Code interface with the following details:

- File Explorer:** Shows a project structure under "TANMAYA191-SDLC_8 THUNDERBIRDS" containing files like .github, .vscode, Requirements, Design, Implementation, TestPlan, Daily Update, Readme.md, trace_width.png, .gitignore, LICENSE, and README.md.
- Terminal:** Displays the output of a C program named "main.c". The code includes comments for various calculations (e.g., Ohms law, KVL, KCL, Tensile calculator). The terminal shows the user entering values for current, thickness, temperature rise, ambient temperature, and trace length, followed by the calculated internal and external trace widths.
- Status Bar:** Shows the line number (Ln 25), column number (Col 41), spaces (Spaces: 4), encoding (UTF-8), and date/time (24-08-2021 15:14).

Image 13

The screenshot shows a Visual Studio Code interface with the following details:

- File Explorer:** Shows a project structure under "TANMAYA191-SDLC_8 THUNDERBIRDS" containing files like .github, .vscode, Requirements, Design, Implementation, TestPlan, Daily Update, Readme.md, trace_width.png, .gitignore, LICENSE, and README.md.
- Terminal:** Displays the output of a C program named "main.c". The code includes comments for various calculations (e.g., Ohms law, KVL, KCL, Tensile calculator). The terminal shows the user being prompted to choose between Current Division and Voltage Division, then entering input current and resistances, and finally displaying the results for current across resistors.
- Status Bar:** Shows the line number (Ln 25), column number (Col 41), spaces (Spaces: 4), encoding (UTF-8), and date/time (24-08-2021 15:14).

GENESIS Learning Report – Module Name

Image 14

```
main.c - tanmaya191-SDLC_8_Thunderbirds - Visual Studio Code

File Edit Selection View Go Run Terminal Help main.c - tanmaya191-SDLC_8_Thunderbirds - Visual Studio Code

EXPLORER Makefile main.c M X
TANMAYA191-SDLC_8_THUNDERBIRDS
3.Implementation > C main.c > main()
19 printf("4. Ohms law\n"); //Viswak
20 printf("5. KVl\n"); //Viswak
21 printf("6. KCL\n"); //Viswak
22 printf("7. Tensile calculator\n"); //Viswak

PROBLEMS OUTPUT TERMINAL DEBUG CONSOLE
powerShell + x

12. Division
13. Parallel
14. Series
15. Resistor colour code
16. Circular convolution
Type : 13
-----
Parallel-----

Enter which passive element's parallel equivalent you want to find
Press 1 'C' for capacitance
Press 2 'I' for inductance
Press 3 'R' for resistance
1

Enter the number of Capacitors : 3
Enter Value of Each Capacitance in microFarads:
Capacitance value1 : 5
Capacitance value2 : 10
Capacitance value3 : 9

Equivalent Parallel Capacitance : 24.000000 microFarads
PS H:\Github\tanmaya191-SDLC_8_Thunderbirds\3_Implementation> make run
gcc src/freq.c src/parameter.c src/power_input.c src/power_output.c src/resistance.c src/secondary_turns.c src/speed.c src/torque.c src/Load_current.c src/Factor_s.c src/parallel.c src/series.c src/halfadder.c src/halfsubtracter.c src/DutyCycle.c src/kvl.c src/kcl.c src/ohm.c src/tensiles.c src/trace_width.c src/starDeltaConversion.c src/division.c src/CircularConvolution.c src/resistance_calculator.c main.c -Iinc -Itest -o build\Thunderbirds.exe -lm
build\Thunderbirds.exe

Ln 25, Col 41 Spaces: 4 UTF-8 CRLF C Win32 ⌂ 15:14 ENG 24-08-2021
```

Image 15

```
main.c - tanmaya191-SDLC_8_Thunderbirds - Visual Studio Code

File Edit Selection View Go Run Terminal Help main.c - tanmaya191-SDLC_8_Thunderbirds - Visual Studio Code

EXPLORER Makefile main.c M X
TANMAYA191-SDLC_8_THUNDERBIRDS
3.Implementation > C main.c > main()
19 printf("4. Ohms law\n"); //Viswak
20 printf("5. KVl\n"); //Viswak
21 printf("6. KCL\n"); //Viswak
22 printf("7. Tensile calculator\n"); //Viswak

PROBLEMS OUTPUT TERMINAL DEBUG CONSOLE
powerShell + x

15. Resistor colour code
16. Circular convolution
Type : 14
-----
Series-----

Enter which passive element's series equivalent you want to find
Press 1 'C' for capacitance
Press 2 'I' for inductance
Press 3 'R' for resistance
2

Enter the number of Inductors : 3
Enter Value of Each Inductance in milli Henry :
Inductor value1 : 5
Inductor value2 : 9
Inductor value3 : 8

Equivalent Series Inductance : 22.000000 milli Henry
PS H:\Github\tanmaya191-SDLC_8_Thunderbirds\3_Implementation> make run
gcc src/freq.c src/parameter.c src/power_input.c src/power_output.c src/resistance.c src/secondary_turns.c src/speed.c src/torque.c src/Load_current.c src/Factor_s.c src/parallel.c src/series.c src/halfadder.c src/halfsubtracter.c src/DutyCycle.c src/kvl.c src/kcl.c src/ohm.c src/tensiles.c src/trace_width.c src/starDeltaConversion.c src/division.c src/CircularConvolution.c src/resistance_calculator.c main.c -Iinc -Itest -o build\Thunderbirds.exe -lm
build\Thunderbirds.exe

Ln 25, Col 41 Spaces: 4 UTF-8 CRLF C Win32 ⌂ 15:14 ENG 24-08-2021
```

GENESIS Learning Report – Module Name

Image 16

The screenshot shows a Visual Studio Code interface with the following details:

- File Explorer:** Shows a project structure for "TANMAYA191-SDLC_8_THUNDERBIRDS" containing files like .github, .vscode, Requirements, Design, TestPlan, Daily Update, README.md, trace_width.png, and .gitignore.
- Terminal:** Displays the command "make run" being executed in the directory "H:\Github\tanmaya191-SDLC_8_Thunderbirds\3_Implementation". The output shows the resistor color code conversion logic and the result for a resistor described as "BYRO".
- Code Editor:** The main.c file contains code for calculating resistor values based on color codes. It includes a mapping of colors to numerical values (N=Brown, R=Red, O=Orange, Y=Yellow, G=Green, L=Blue, V=Violet, E=Gray, W=White) and a series of printf statements for different calculations.
- Status Bar:** Shows the current file is "main.c", the line and column are 25, Col 41, and the encoding is UTF-8.

Image 17

The screenshot shows a Visual Studio Code interface with the following details:

- File Explorer:** Shows a project structure for "TANMAYA191-SDLC_8_THUNDERBIRDS" containing files like .github, .vscode, Requirements, Design, TestPlan, Daily Update, README.md, trace_width.png, and .gitignore.
- Terminal:** Displays the command "make run" being executed in the directory "H:\Github\tanmaya191-SDLC_8_Thunderbirds\3_Implementation". The output shows the circular convolution logic and the result for two input arrays, x[n] and h[n].
- Code Editor:** The main.c file contains code for performing circular convolution. It prompts for the lengths of the input arrays and their values, then prints the resulting array y[n].
- Status Bar:** Shows the current file is "main.c", the line and column are 25, Col 41, and the encoding is UTF-8.