

## **PROBLEM STATEMENT**

We are tasked with developing a MATLAB program for a company that facilitates the calculation and display of resistor values and tolerances based on user-input color codes. Resistors are essential components in electric circuits, regulating current flow. The relationship between voltage, current, and resistance is governed by Ohm's Law: ( $V = IR$ ), where  $V$  is voltage (Volts),  $I$  is current (Amperes), and  $R$  is resistance (Ohms).

For through-hole resistors, where legs are inserted into a printed circuit board (PCB) and soldered, a color code system aids in determining the resistor's value and tolerance. Two common types of color codes are used: 4-band and 5-band.

### **Program Requirements:**

- **Menu System:** The program should start with an initial menu system providing instructions on entering resistor color bands.
- **User Input:** Users should input resistor band colors for either 4-band or 5-band resistor types.
- **Error Handling:** Implement error checking and input validation. Notify and reprompt users for input if they enter an invalid menu selection or color.
- **Calculation:** Based on the entered color bands, calculate and display the decoded resistor value and tolerance in a user-friendly format on the Command Window.

### **Resistor Decoding:**

- **4-band Resistors:** The first and second colors represent the first two numbers of the resistor value, the third band indicates the multiplier, and the fourth band signifies the tolerance.
- **5-band Resistors:** The first, second, and third color bands represent the first three numbers in the resistor value, the fourth band is the multiplier, and the fifth band indicates the tolerance.

### **Program Implementation:**

- Develop a text-based user interface within MATLAB.
- Allow users to select resistor type (4-band or 5-band) and enter color bands accordingly.
- Decode the resistor value and tolerance using the provided color code information.
- Display the decoded values to the user in a clear and understandable format.

Our program should be efficient, accurate, and capable of handling inputs for both 4-band and 5-band resistor types.

## INPUTS

- Number of color bands (params.bandNo)

If the number of color bands is 4

- Color of the first band (params.col1)
- Color of the second band (params.col2)
- Color of the third band (params.colMul)
- Color of the fourth band (params.colTol)

If the number of color bands is 5

- Color of the first band (params.col1)
- Color of the second band (params.col2)
- Color of the third band (params.col3)
- Color of the fourth band (params.colMul)
- Color of the fifth band (params.colTol)

## OUTPUTS

- Calculated resistor value
- Tolerance of the resistor

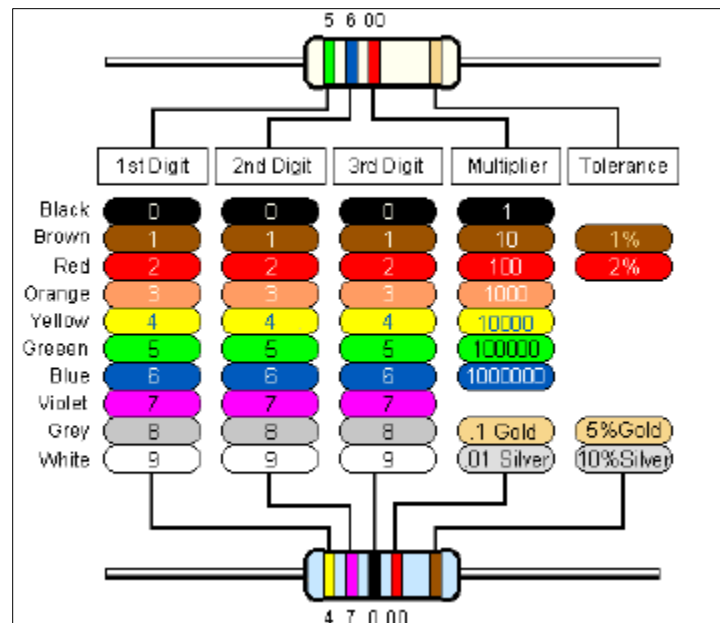


Figure 01: Resistor Color Codes

## ALGORITHM DESIGN

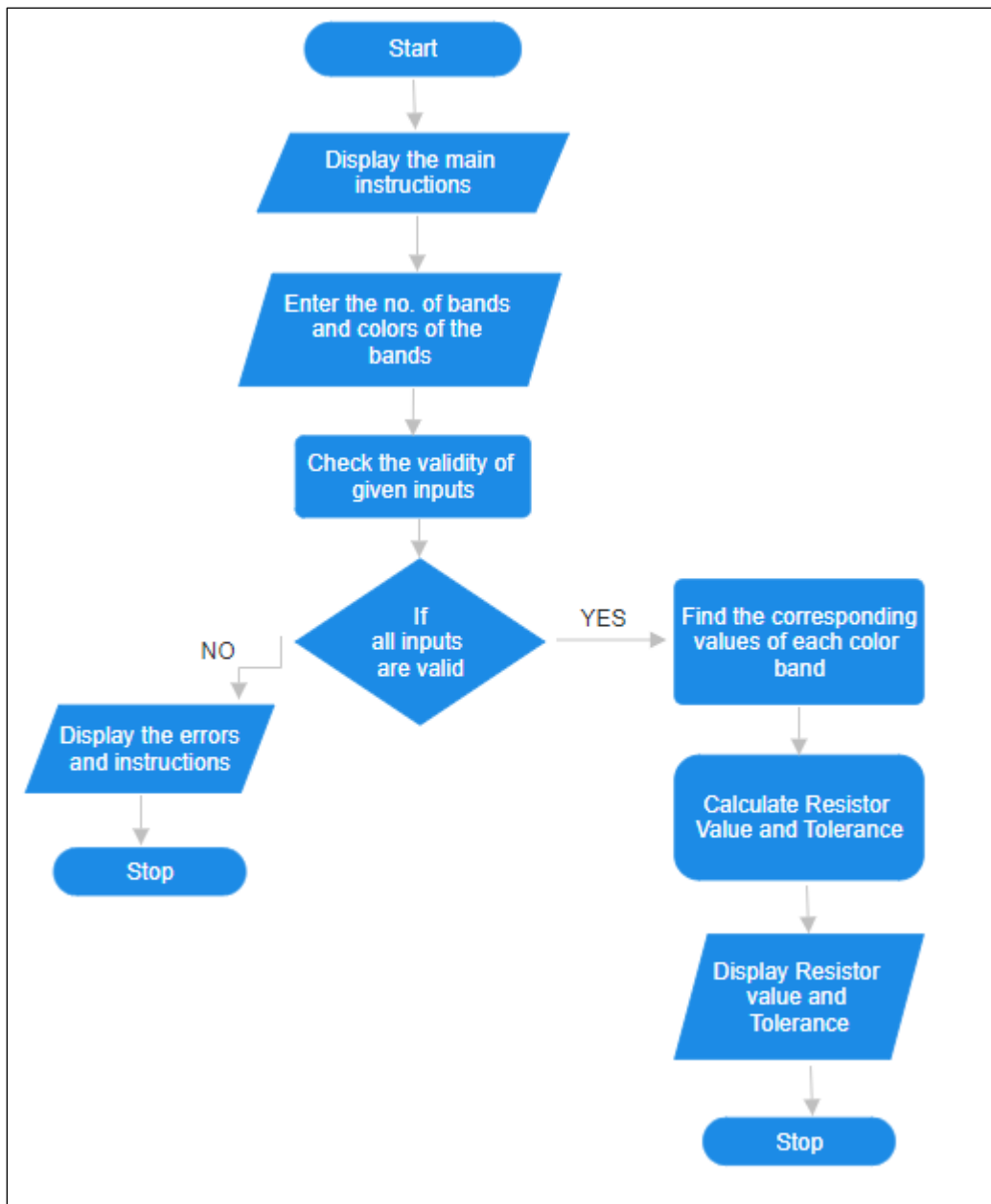


Figure 02: Flow Chart of the MatLab Algorithm

## TESTING

1. **Unit Testing:** The program underwent extensive unit testing to verify the accuracy of individual functions such as getColorValue, getColorMulValue, getColorTolValue, and checkValidity. Unit tests were designed to cover various scenarios, including valid and invalid input color names, to ensure robust error handling and accurate conversion of color names to corresponding values.
2. **Integration Testing:** After confirming the correctness of individual functions, integration testing was conducted to assess the overall functionality of the program. Integration tests focused on scenarios where multiple functions interacted, such as merging user-provided parameters with default parameters and calculating resistor values based on the entered color bands.
3. **User Interface Testing:** The text-based user interface was evaluated for usability and clarity. Test cases included entering valid and invalid input color names, testing the error handling mechanism, and verifying the correctness of output messages displayed to the user.

- **Testing case 01:** [4 bands, green, blue, red, gold]  
Manually calculated values: 5600 Ohms    +/-5%

```
>> params.bandNo = 4;
>> params.col1 = 'green';
>> params.col2 = 'blue';
>> params.colMul = 'red';
>> params.colTol = 'gold';
>> resistorValue(params)
[4]      'green'    'blue'    'red'    'gold'
Resistor Value = 5600 Ohms  Tolerance = +/-5%
```

Figure 03: Inputs and outputs of testing case 01

- **Testing case 02:** [5 bands, yellow, violet, black, red, brown]  
Manually calculated values: 47,000 Ohms    +/-1%

```
>> params.bandNo = 5;
>> params.col1 = 'yellow';
>> params.col2 = 'violet';
>> params.col3 = 'black';
>> params.colMul = 'red';
>> params.colTol = 'brown';
>> resistorValue(params)
[5]      'yellow'    'violet'    'black'    'red'    'brown'
Resistor Value = 47000 Ohms  Tolerance = +/-1%
```

Figure 04: Inputs and Outputs of testing case 02

- **Testing case 03:** [5 bands, brown, green, orange, green, silver]  
Manually calculated values: 15,300,000 Ohms    +/-10%

```

>> params.bandNo = 5;
>> params.col1 = 'brown';
>> params.col2 = 'green';
>> params.col3 = 'orange';
>> params.colMul = 'green';
>> params.colTol = 'silver';
>> resistorValue(params)
      [5]      'brown'      'green'      'orange'      'green'      'silver'

Resistor Value = 15300000 Ohms  Tolerance = +/-10%

```

Figure 05: Inputs and Outputs of testing case 03

- **Testing case 04:** [4 bands, grey, white, yellow, brown]  
Manually calculated values: 890,000 Ohms    +/-1%

```

>> params.bandNo = 4;
>> params.col1 = 'gray';
>> params.col2 = 'white';
>> params.colMul = 'yellow';
>> params.colTol = 'brown';
>> resistorValue(params)
      [4]      'gray'      'white'      'yellow'      'brown'

Resistor Value = 890000 Ohms  Tolerance = +/-1%

```

Figure 06: Inputs and Outputs of testing case 04

- **Testing case 05:** Invalid input for the band number parameter (params.bandNo)  
This parameter only accepts 4 and 5. It will give the below error for any other value.

```

>> params.bandNo = 6;
>> params.col1 = 'yellow';
>> params.col2 = 'violet';
>> params.col3 = 'black';
>> params.colMul = 'red';
>> params.colTol = 'brown';
>> resistorValue(params)
Error using resistorValue (line 59)
Invalid input for param.bandNo, see the instructions using mainInfo() function

```

Figure 07: Inputs and Outputs of testing case 05

- **Testing case 06:** Invalid input for color bands parameters (params.col1, params.col2 etc.)  
These parameters only accept the pre-defined colors only. Any other colors or any words with spelling mistakes will not be accepted.

```

>> params.bandNo = 5;
>> params.col1 = 'yellow';
>> params.col2 = 'purple';
>> params.col3 = 'black';
>> params.colMul = 'red';
>> params.colTol = 'brown';
>> resistorValue(params)
Error using resistorValue (line 43)
Invalid input fot param.col2, see the instructions using mainInfo() function

```

Figure 08: Inputs and Outputs of testing case 06 (part I)

```

>> params.bandNo = 5;
>> params.col1 = 'yellow';
>> params.col2 = 'violet';
>> params.col3 = 'blak';
>> params.colMul = 'red';
>> params.colTol = 'brown';
>> resistorValue(params)
Error using resistorValue (line 45)
Invalid input fot param.col3, see the instructions using mainInfo() function

```

Figure 09: Inputs and Outputs of testing case 06 (part II)

```

>> mainInfo()
-----Define the Input Arguments-----
->If the resistor has 4 colour bands
params.bandNo = 4
params.col1 = color of the 1st color band
params.col2 = color of the 2nd color band
params.colMul = color of the 3rd color band
params.colTol = color of the 4th color band

->If the resistor has 5 colour bands
params.bandNo = 5
params.col1 = color of the 1st color band
params.col2 = color of the 2nd color band
params.col3 = color of the 3rd color band
params.colMul = color of the 4th color band
params.colTol = color of the 5th color band

-----Call the Function-----
resistorValue(params)

-----valid colors for input arguments-----
col1/col2/col3 : black, brown, red, orange, yellow, green, blue, violet, grey, white
colMul : black, brown, red, orange, yellow, green, blue, gold, silver
colTol : brown, red, gold, silver

-----examples-----
params.bandNo = 4;
params.col1 = 'green';
params.col2 = 'blue';
params.colMul = 'red';
params.colTol = 'gold';
resistorValue(params)

params.bandNo = 5;
params.col1 = 'yellow';
params.col2 = 'violet';
params.col3 = 'black';
params.colMul = 'red';
params.colTol = 'brown';
resistorValue(params)

```

Figure 10: Instructions of the mainInfo() Function

## **DISCUSSION**

In developing the MATLAB program for calculating resistor values and tolerances based on user-input color codes, several steps were taken to ensure the software solution was fit for purpose and produced the expected results. This discussion will delve into analyzing the working program, and highlight its strengths and weaknesses.

The program demonstrates robust error handling and input validation mechanisms, ensuring effective management of invalid user inputs. Informative error messages guide users in correcting input formats and offer clear instructions for rectifying errors. Its modular design enhances code readability, maintainability, and reusability by segmenting functionality into separate functions, facilitating easier troubleshooting and future enhancements. Users benefit from the program's flexibility, as it accommodates both 4-band and 5-band resistor types, allowing seamless switching between configurations without the need for code modifications. The mainInfo function provides comprehensive instructions on defining input arguments, calling the main function (resistorValue), and listing valid color names for each input, enhancing user understanding and utilization of the program.

### **Strengths:**

1. **Accuracy:** The program accurately calculates resistor values and tolerances based on user-provided color codes, adhering to the specifications outlined in the problem statement. Unit testing ensures the correctness of individual functions, contributing to overall accuracy.
2. **User-Friendly Interface:** The text-based user interface is intuitive and user-friendly, guiding users through the input process and providing informative feedback. Clear instructions and error messages enhance the user experience, making the program accessible to a wide range of users.
3. **Error Handling:** Robust error handling mechanisms prevent runtime errors and ensure graceful handling of invalid user inputs. Error messages are descriptive and actionable, empowering users to correct input mistakes efficiently.

### **Weaknesses:**

1. **Limited Color Recognition:** The program only recognizes a predefined set of color names for input arguments, potentially limiting its applicability in scenarios involving custom or non-standard color codes. Enhancements could be made to support a broader range of color names or accommodate user-defined color mappings.
2. **Minimal Error Recovery:** In cases where invalid input colors are provided, the program terminates with an error message, requiring users to re-enter correct inputs from the beginning. Implementing error recovery mechanisms to allow users to correct individual input errors without restarting the entire process could enhance usability.

In conclusion, the developed MATLAB program effectively fulfills the requirements outlined in the problem statement by accurately calculating resistor values and tolerances based on user-provided color codes. Through rigorous testing and analysis, strengths such as accuracy, user-friendliness, and robust error handling were identified, along with weaknesses related to color recognition, input sanitization, and error recovery. Overall, the program provides a solid foundation for further refinement and future enhancements to meet evolving user needs and requirements.