De La Salle University – Manila

Gokongwei College of Engineering

Department of Electronics and Computer Engineering

Resistor Color Coded Image Generator

LBYEC4A

Term-End Project

by

CASTILLO, Seth Lawrence D.

ROSANES, Marc Gabriel L.

VALDERAMA, Ruskin Antoine C.

LBYEC4A – EK2

April 16, 2023

I. Abstract

This Project presents a program that generates an image of a resistor with appropriate color bands based on the inputted resistance value and tolerance level. The program was created in MATLAB using various image processing techniques, and MATLAB commands and functions. This program determines the first two bands of the resistor represented by the first two digits of the input resistance value, then calculates the multiplier of the resistance value to determine the third band, and finally sets the fourth color band based on the inputted tolerance level. The program accepts commercially available resistors which are present on the E12 and E24 series resistors. If deemed unavailable, the program prompts the user to input a new set of values again. The successful execution of this project aims to demonstrate the feasibility of automating the process of resistor color band generation, which can save time and effort for electronics hobbyists and professionals alike especially when it comes to circuit building.

II. INTRODUCTION

Resistors are one of the most common yet one of the most important components of an electronic circuit. Resistors are the ones responsible for limiting the flow of electrons in a circuit. Without a resistor or without properly applying the correct resistance value in a circuit, the device will either be shorted or will experience overvoltage and might even explode. Since resistors are widely used, various innovations have been made. Different series of resistors differ in their tolerance, the number of bands, and even their uses. According to research [#], a resistor is essential to protect electronic circuits against voltage spikes and it can provide the proper voltage that is fit and compatible with the other components in the electronic circuit.

Since resistors are one of the most common components used by electrical and electronic engineers around the world, every circuit contains one; and with this being a fundamental component in almost every electronic circuits, it is important that the resistors connected to a circuit are precise or accurate to have a fully functioning circuit. With this, the students aim to create a program that will aid Electronics and Electrical Engineers to easily identify and locate the proper resistance value of a resistor by having the program display the desired resistor given/inputted a resistance value. The program will be done under MATLAB; together with its toolboxes mainly focusing on image processing. The program to be designed would be able to simplify the identification of resistors in real-life for engineers and students, which they can benefit from, thereby improving their productivity and efficiency in circuit design and analysis.

III. THEORETICAL CONSIDERATION

- **Resistors & Resistor Color Coding**

Type of Resistor	Description
Carbon Film	These are low-cost resistors made by depositing a carbon film on a ceramic substrate. Carbon Film resistors are available in a range of resistance values and power ratings and they are the most common resistors that an engineering student would see.
Metal Film	Similar to carbon film resistors, Metal Film resistors uses a thin layer of metal instead of carbon. These have better temperature stability and lower noise than carbon film resistors as metal is more a stable component. With that being said,
Wirewound type	The Wirewound resistors name come from the method it was created, it was made by winding a wire around a ceramic core. They can handle high power and have high precision and stability.
Thick Film	These are made by screen-printing a resistive ink onto a ceramic substrate. Generally, Thick films are low-cost and can be made in a range of shapes and sizes.
Thin Film	Similar to thick film resistors, Thin films use a thinner layer of resistive material, therefore having better stability and accuracy than thick film resistors.
Surface Mount	These are small resistors that are designed to be mounted directly onto a circuit board. They are available in a range of sizes and power ratings.
Variable Type	These are resistors that can be adjusted to change their resistance value. They are commonly used for volume and tone controls in audio equipment.

IV. METHODOLOGY

The project focuses on the concept of image processing while utilizing different MATLAB commands in order to output the correct and precise color bands of the resistor which depends on the inputted resistance value by the user. To prevent memory overload and prolonged codes, the group implemented function files to have a more readable and understandable code.

The first section of the program asks the user to input a resistance value and a tolerance level which is the standard for all real-world resistors. From this user input, the program gets the first two digits of the inputted resistance value as this value determines what color will be the first 2 colors of the 4 band resistor image to be generated. From here, the program now checks whether the resistance value together with its tolerance level inputted are available in real life based on the list of available resistors of the E12 and E24 series resistors (Keim, 2023). Once the input values are validated, the program now develops or generates the resistor image with the correct color bands. For the last 2 color bands of the 4 band resistor, the 3rd band would be based on the multiplier value of the resistance value inputted which can be calculated by dividing the resistance value from its first two digits. Then for the 4th band, the color would depend on the tolerance level the user has inputted.

The second section of the program focuses on the image processing of the resistor to be generated. The program would first read the resistor template with blank bands or colorless resistor bands. With the function files created, it would consist of the codes that would change the color of the bands of the resistor based on the values gathered from the first section of the program. The color changes were done by getting the RGB values of the resistor template, then prompted by conditional statements that would allow RGB value changes that would put colors on the blank bands of the resistor image. The colors to be displayed on the bands are based on the RGB value of the 12 possible colors of the 4 bands on a resistor. By using the concept of matrix or arrays, the image's RGB colors would be possible to change as the image read by the program, the resistor template, is converted to a 3-Dimensional image wherein the 3rd dimension would be used to determine the intensity of the Red, Green, and Blue to mix colors and output the appropriate colors needed to produce the desired resistor. From this, the last part needed for the project to be finished is to compile these new RGB values into the resistor template to be displayed on the program afterwards.

Table 1. MATLAB Codes to Generate a Color-Coded Resistor

```
check1 = 1;
check2 = 1;
multiplier = 1; % Initializing multiplier checks for the commercial
while (mod(multiplier,10)~=0)&&(check1==1 || check2==1)
res = input("Enter Resistance: ");
t = input("Enter a Tolerance Level: ");

% Convert Number to String
res_str = num2str(res);

% Get First 2 Numbers
if length(res_str) >= 2
    first_two_digits = str2double(res_str(1:2));
    multiplier = res/first_two_digits;
else
    disp("The number has less than 2 digits");
end

% Tolerance and Resistance Validity Checker
if (t==10)
   E12 = [10 12 15 18 22 27 33 39 47 56 68 82];
   check1 = ismember(first_two_digits,E12);
   if (check1==1)
      check2 = 0;
      disp("Valid Resistance Value!")
      fprintf('Resistance = %0.0f', res)
      fprintf("Multiplier is %0.0f", multiplier);
      fprintf('Tolerance = %0.0f%%', t)
   else
      disp("Invalid Resistance Value!")
      fprintf("Multiplier = %0.0f", multiplier)
      disp("Not A Commercially Available Resistor.")
   end
elseif (t==1||t==5)
   E24 = [10 11 12 13 15 16 18 20 22 24 27 30 33 36 39 43 47 51 56 62 68 75 82 91];
   check2 = ismember(first_two_digits,E24);
   if (check2==1)
      check1 = 0;
      disp("Valid Resistance Value!")
      fprintf('Resistance = %0.0f', res)
      fprintf("Multiplier = %0.0f", multiplier);
      fprintf('Tolerance = %0.0f%%', t)
   else
      disp("Invalid Resistance Value!")
      fprintf("Multiplier = %0.0f", multiplier)
      disp("Not A Commercially Available Resistor.")
   end
else
   disp("Enter Valid Tolerance Level.")
end
end
```

```
% Resistor Display
Resistor = imread('resistor.png'); %168x584x3

%Checks the multiplier
basemultiplier = log10(multiplier);

%separates the first two digits
[digit1, digit2] = separateDigits(first_two_digits);
```

```
%takes the first digit and convert it to a color band
%takes the Second digit and convert it to a color band
%takes the multiplier and convert it to a color band
%takes the tolerance value and convert it to a color band
[RES1_2_3_4] = Band1234Generator(digit1,digit2,basemultiplier,t);

%assigns the colorbands
Resistor = RES1_2_3_4;

%shows the image-generated resistor
if(check1~=0 || check2~=0) %shows only valid resistance values
imshow(Resistor)
end
```

```
function [RES1_2_3_4] = Band1234Generator(color1,color2,color3,color4)
Resistor = imread('resistor.png'); %168x584x3
%-----------------------------------------------------
if (color1 == 0)
    Resistor(11:159,87:110,1) = 0;
    Resistor(11:159,87:110,2) = 0; %black
    Resistor(11:159,87:110,3) = 0;
elseif (color1 == 1)
    Resistor(11:159,87:110,1) = 150;
    Resistor(11:159,87:110,2) = 75; %Brown
    Resistor(11:159,87:110,3) = 0;
elseif (color1 == 2)
    Resistor(11:159,87:110,1) = 255;
    Resistor(11:159,87:110,2) = 0; %red
    Resistor(11:159,87:110,3) = 0;
elseif (color1 == 3)
    Resistor(11:159,87:110,1) = 255;
    Resistor(11:159,87:110,2) = 165; %orange
    Resistor(11:159,87:110,3) = 0;
elseif (color1 == 4)
    Resistor(11:159,87:110,1) = 255;
    Resistor(11:159,87:110,2) = 255; %yellow
    Resistor(11:159,87:110,3) = 0;
elseif (color1 == 5)
    Resistor(11:159,87:110,1) = 0;
    Resistor(11:159,87:110,2) = 255; %green
    Resistor(11:159,87:110,3) = 0;
elseif (color1 == 6)
    Resistor(11:159,87:110,1) = 0;
    Resistor(11:159,87:110,2) = 0; %blue
    Resistor(11:159,87:110,3) = 255;
elseif (color1 == 7)
    Resistor(11:159,87:110,1) = 255;
    Resistor(11:159,87:110,2) = 0; %violet
    Resistor(11:159,87:110,3) = 255;
elseif (color1 == 8)
    Resistor(11:159,87:110,1) = 128;
    Resistor(11:159,87:110,2) = 128; %gray
    Resistor(11:159,87:110,3) = 128;
elseif (color1 == 9)
    Resistor(11:159,87:110,1) = 255;
    Resistor(11:159,87:110,2) = 255; %white
    Resistor(11:159,87:110,3) = 255;
end
%-----------------------------------------------------
if (color3 == 0)
    Resistor(21:149,243:265,1) = 0;
    Resistor(21:149,243:265,2) = 0; %black
    Resistor(21:149,243:265,3) = 0;
elseif (color3 == 1)
    Resistor(21:149,243:265,1) = 150;
    Resistor(21:149,243:265,2) = 75; %brown
    Resistor(21:149,243:265,3) = 0;
elseif (color3 == 2)
    Resistor(21:149,243:265,1) = 255;
    Resistor(21:149,243:265,2) = 0; %red
    Resistor(21:149,243:265,3) = 0;
elseif (color3 == 3)
    Resistor(21:149,243:265,1) = 255;
    Resistor(21:149,243:265,2) = 165; %orange
    Resistor(21:149,243:265,3) = 0;
elseif (color3 == 4)
    Resistor(21:149,243:265,1) = 255;
    Resistor(21:149,243:265,2) = 255; %yellow
    Resistor(21:149,243:265,3) = 0;
elseif (color3 == 5)
    Resistor(21:149,243:265,1) = 0;
    Resistor(21:149,243:265,2) = 255; %green
    Resistor(21:149,243:265,3) = 0;
elseif (color3 == 6)
    Resistor(21:149,243:265,1) = 0;
    Resistor(21:149,243:265,2) = 0; %blue
    Resistor(21:149,243:265,3) = 255;
elseif (color3 == 7)
    Resistor(21:149,243:265,1) = 255;
    Resistor(21:149,243:265,2) = 0; %violet
    Resistor(21:149,243:265,3) = 255;
elseif (color3 == 8)
    Resistor(21:149,243:265,1) = 128;
    Resistor(21:149,243:265,2) = 128; %gray
    Resistor(21:149,243:265,3) = 128;
elseif (color3 == 9)
    Resistor(21:149,243:265,1) = 255;
    Resistor(21:149,243:265,2) = 255; %white
    Resistor(21:149,243:265,3) = 255;
end
%-----------------------------------------------------
if (color4 == 1)
   Resistor(10:160,478:498,1) = 150;
   Resistor(10:160,478:498,2) = 75; %brown
   Resistor(10:160,478:498,3) = 0;

elseif (color4 == 5)
   Resistor(10:160,478:498,1) = 255;
   Resistor(10:160,478:498,2) = 215; %gold
   Resistor(10:160,478:498,3) = 0;
elseif (color4 == 10)
   Resistor(10:160,478:498,1) = 192;
   Resistor(10:160,478:498,2) = 192; %silver
   Resistor(10:160,478:498,3) = 192;
end
%-----------------------------------------------------
RES1_2_3_4 = Resistor; % Assign resulting image to RES variable
end
```

```
function [digit1, digit2] = separateDigits(number)
% This function separates a two-digit number into its individual digits
% and stores them in separate variables

digit1 = floor(number/10); % Extract the tens digit
digit2 = mod(number, 10); % Extract the ones digit
end
```

```
%-----------------------------------------------------
if (color2 == 0)
    Resistor(20:150,175:197,1) = 0;
    Resistor(20:150,175:197,2) = 0; %black
    Resistor(20:150,175:197,3) = 0;
elseif(color2 == 1)
    Resistor(20:150,175:197,1) = 150;
    Resistor(20:150,175:197,2) = 75; %Brown
    Resistor(20:150,175:197,3) = 0;
elseif (color2 == 2)
    Resistor(20:150,175:197,1) = 255;
    Resistor(20:150,175:197,2) = 0; %red
    Resistor(20:150,175:197,3) = 0;
elseif (color2 == 3)
    Resistor(20:150,175:197,1) = 255;
    Resistor(20:150,175:197,2) = 165; %orange
    Resistor(20:150,175:197,3) = 0;
elseif (color2 == 4)
    Resistor(20:150,175:197,1) = 255;
    Resistor(20:150,175:197,2) = 255; %yellow
    Resistor(20:150,175:197,3) = 0;
elseif (color2 == 5)
    Resistor(20:150,175:197,1) = 0;
    Resistor(20:150,175:197,2) = 255; %green
    Resistor(20:150,175:197,3) = 0;
elseif (color2 == 6)
    Resistor(20:150,175:197,1) = 0;
    Resistor(20:150,175:197,2) = 0; %blue
    Resistor(20:150,175:197,3) = 255;
elseif (color2 == 7)
    Resistor(20:150,175:197,1) = 255;
    Resistor(20:150,175:197,2) = 0; %violet
    Resistor(20:150,175:197,3) = 255;
elseif (color2 == 8)
    Resistor(20:150,175:197,1) = 128;
    Resistor(20:150,175:197,2) = 128; %gray
    Resistor(20:150,175:197,3) = 128;
elseif (color2 == 9)
    Resistor(20:150,175:197,1) = 255;
    Resistor(20:150,175:197,2) = 255; %white
    Resistor(20:150,175:197,3) = 255;
end
%-----------------------------------------------------
```

V. RESULTS

The Results of the project as it is enabled by the user are as follows. The process can be divided into three sequences. First is the default program interface. When the program is running, the program interface is set at a default asking the user to input a certain resistance value. As the user inputs the correct resistance value; given that it is commercially available and is appropriate with the E12 and E24 series, the program will proceed and will ask the user for the resistor's tolerance value. If otherwise, the program will reask the user to input another resistance value.

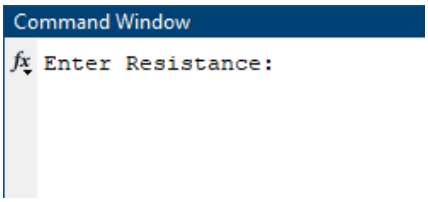

Figure 1. Program Interface

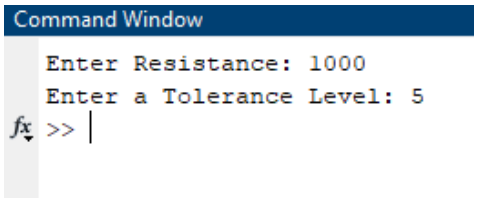

Figure 2. Program Interface upon user's input

Upon the program's acceptance of the inputted resistance and tolerance value, the program will proceed into reading the data and check if the 1st band, 2nd band, multiplier, and tolerance value are valid and appropriate to the E12 and E24 series. Lastly, the final output is where the resistance value that was inputted by the user will reflect. Through MATLAB's image processing tools, the resistor image will be presented and the colors of the bands will reflect as well.

```
Valid Resistance Value!
Resistance = 1000
Multiplier = 100
Tolerance = 5%
```

Figure 3. Appropriate Resistance Value.

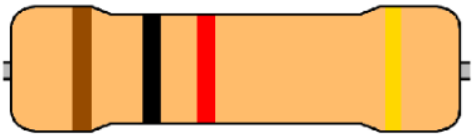

Figure 4. Generated Resistor Image.

Another ability that the program can do is to detect when the inputted resistance value is inappropriate or is not “Commercially Available”. On this instance, a message will appear stating that the resistance value is invalid, then, the program will reask the user again for another input of resistance and tolerance as it is looped.

```
Invalid Resistance Value!
Multiplier = 1
Not A Commercially Available Resistor.
```

Figure 5. Invalid Resistance Input Sequence.

VI. DISCUSSION

The generated image was processed through the use of image processing and various MATLAB commands. After the program has determined what is the first two digits of the inputted resistance value, the program then separates this digit into two in order to have a basis for the first two bands of the resistor. As an example seen above, the input was 1000 ohms, and the program have determine the first two digits which are 1 and 0. From the function file created for generating colors for the 4 bands, the number 1 would be equal to a brown color band, while 0 is a black color band. From here, the next step the program would do is to get the multiplier of the resistance value by dividing the resistance value by the first two digits. With this, the quotient of 1000 and 10 is 100, which means that the multiplier of the resistance is 100 which is equivalent to a red color band. Lastly, the 4th and last band of the resistor will be changed depending on the tolerance level inputted. With a tolerance level of 5%, a gold color band is implemented to the resistor. This process will be the same for all inputted resistance values as long as the input value is valid, and therefore, commercially available. The sample result from this process is seen in Figures 1, 2, 3, and 4 of Section V.

The program isn't limited to the generation of the image of a commercially available resistor. Another operation that the program can do is to detect whether the inputted resistance is valid or invalid or check whether that resistance is available commercially. If the resistance is valid, then the program would proceed with image processing, but if the value inputted is invalid, the program would prompt the user to input another resistance value together with a tolerance level. This prompt would repeat until the user has inputted a valid resistance value. Now this instance would be prompted if the first two digits of the inputted resistance are not included on the E12 and E24 series resistors. Another possible instance that the input would be invalid is when the available resistor is not available when combined with the inputted tolerance level. Meaning, some resistance values included in the E24 series may not be viable with a 10% tolerance level since the E12 series has a 10% tolerance level while the E24 series is available with a 1% and 5% tolerance level. With this, the program wouldn't just generate a resistor image even when the inputted resistance is not on the E12 and E24 series resistors. A sample of this invalid display text can be seen in Figure 5 of Section V.

VII. CONCLUSION

As Resistors are one of the most common components used in Electronic Devices, a proper resistance value is needed to attain electric stability in the circuit. Resistance plays an important role. With that being said, the conventional use of the correct resistor is significant. The student's objective was to create a MATLAB program in which the image of a resistor will be displayed and will portray the inputted resistance value from the user. Following MATLAB's toolboxes and functions, the project has been operative and is functional. The program was able to produce an image of the Commercially Available E12 and E24 series resistors upon the input of the user. The program interface is user-friendly, and the sequence of the program depends on the user's input. The program ensures that the inputted resistance value is commercially available and appropriate to the E12 and E24 series before proceeding to the final output. Through MATLAB's image processing tools, the resistor image is presented, and the colors of the bands reflect the resistance value accurately. The student's MATLAB program which displays an image of a resistor and portrays the inputted resistance value is a useful tool for Electrical and Electronics engineers. By using the program, users can quickly identify the appropriate resistor for their circuits, therefore saving time and effort. Additionally, the program's use of commercially available resistor series ensures that the selected resistor is suitable for a wide range of applications. Overall, the program is an excellent example of how MATLAB's functions and toolboxes can be used to simplify and optimize electronic design and analysis.

VIII. AUTHOR'S CONTRIBUTIONS

CASTILLO, Seth Lawrence D.

- Programming/Coding
- Video Presentation
- Abstract
- Discussion

ROSANES, Marc Gabriel L.

- Programming/Coding
- Poster
- Methodology & Results
- Discussion

VALDERAMA, Ruskin Antoine C.

- Programming/Coding
- Poster
- Introduction
- Theoretical Considerations
- Conclusion

IX. REFERENCES

Dorf, R. C. (2006). The electrical engineering handbook, [Vol. 2] Circuits, signals, and speech and image processing. Crc/Taylor And Francis.

Haque, S. (2019, October 23). Resistor Color Code/ Value Detector from image. Www.mathworks.com. https://www.mathworks.com/matlabcentral/fileexchange/73099-resistor-color-code-value-detector-from-image

Khondker Shihabul Hoque (2019). Resistor Color Code/ Value Detector from image (https://www.mathworks.com/matlabcentral/fileexchange/<...>), MATLAB Central File Exchange. Retrieved October 23, 2019.

Learning Outcomes Real resistor values (the E12 and E24 series). (n.d.). https://ccea.org.uk/downloads/docs/Support/Factfile/2019/Fact%20File%3A%202.5%20Resistors%20E12%20-%20E24.pdf

Lobontiu, N. (2018, January 1). Chapter 8 - State-Space Modeling (N. Lobontiu, Ed.). ScienceDirect; Academic Press. https://www.sciencedirect.com/science/article/pii/B9780128045596000087

Nikolaos Ploskas, & Nikolaos Samaras. (2016). GPU programming in MATLAB. Morgan Kaufmann.

Resistor - Energy Education. (n.d.). Energyeducation.ca. https://energyeducation.ca/encyclopedia/Resistor#:~:text=It

Resistor values | Resistor standards and codes | Resistor Guide. (n.d.). Eepower.com. https://eepower.com/resistor-guide/resistor-standards-and-codes/resistor-values/#

Scherf, K. (2016, August 4). What Does A Resistor Do And Why Is It Important? Del City Blog. https://blog.delcity.net/what-does-a-resistor-do-and-why-is-it-important/

Series resistors (article). (n.d.). Khan Academy. https://www.khanacademy.org/science/electrical-engineering/ee-circuit-analysis-topic/ee-resistor-circuits/a/ee-series-resistors