

**Process Description & Project Planning**

Course: CPE-4700

Report By: Ezequiel Villanueva, Logan Doizaki

12/3/2024

Table of Contents

[Process Overview 3](#_Toc184223363)

[Detailed Process Description 4](#_Toc184223364)

[Budget Preparation 12](#_Toc184223365)

[Project Scope and Timeline 16](#_Toc184223366)

# Process Overview

The purpose of this project is to develop a device which processes an image and applies RGB manipulation through user-based input via hardware.

A diagram of a computer

Description automatically generated

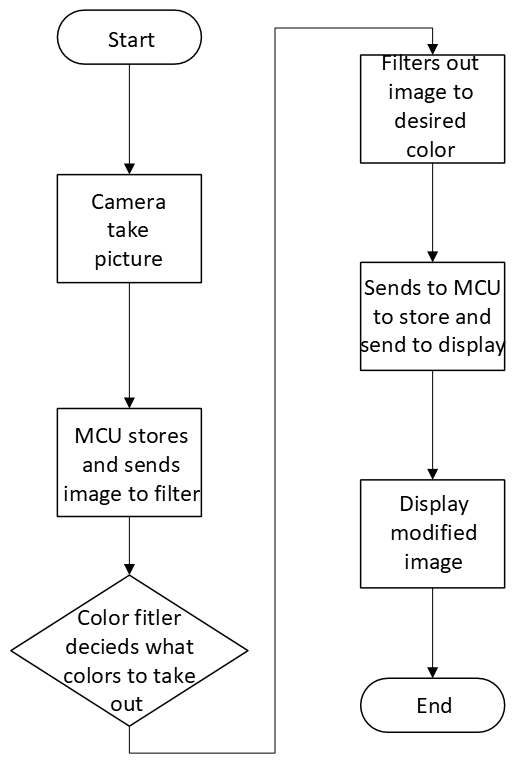
**How the project works:**

1. The device is connected to a microcontroller (MCU) with the ability to take, store, and display images.
2. The MCU captures a picture (storing it on the MCU) using the camera attached and displays the unedited image on the display the MCU is attached to. The image is saved as a JPEG and converted to a Bitmap image.
3. Using the MCU, the whole image file is sent to the Color Device filter in pieces where it will be stored temporarily.
4. On the color filter device, the user can select between 3 buttons. The 3 buttons control the red, green, and blue filters. The buttons, when enabled, take out 100% of the selected color.
5. After the user selects their modifiers, the color filter works by utilizing multiplexers, multipliers, adders, and shifters to apply the change to the image segments stored locally.
6. After digitally altering the image segment, the processed image is stored on the device temporarily.
7. The image is then sent back to the MCU after each segment is modified.
8. The MCU is used to display modified images on the display.

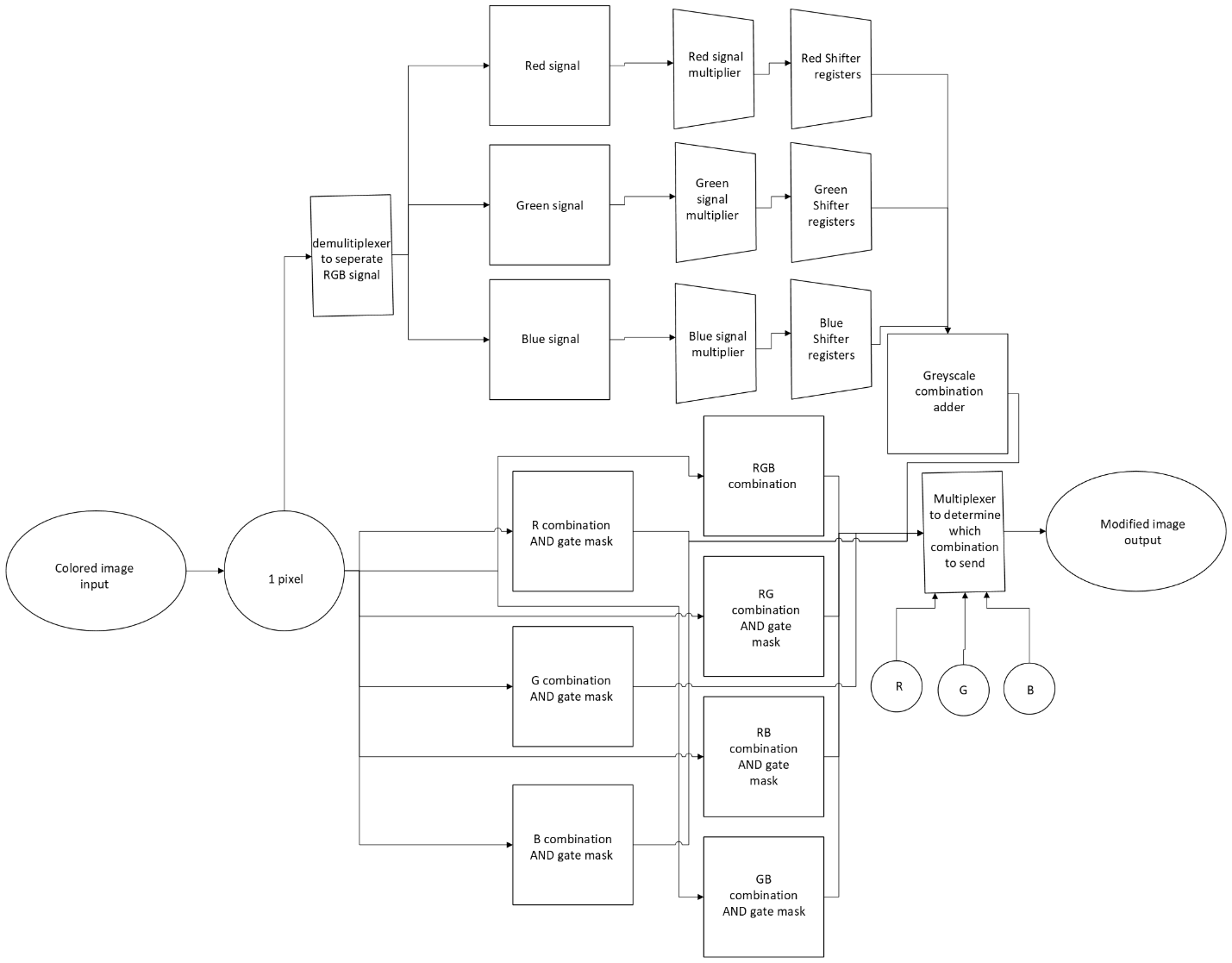
# Detailed Process Description

*Block Diagrams:*

Basic Functionality:

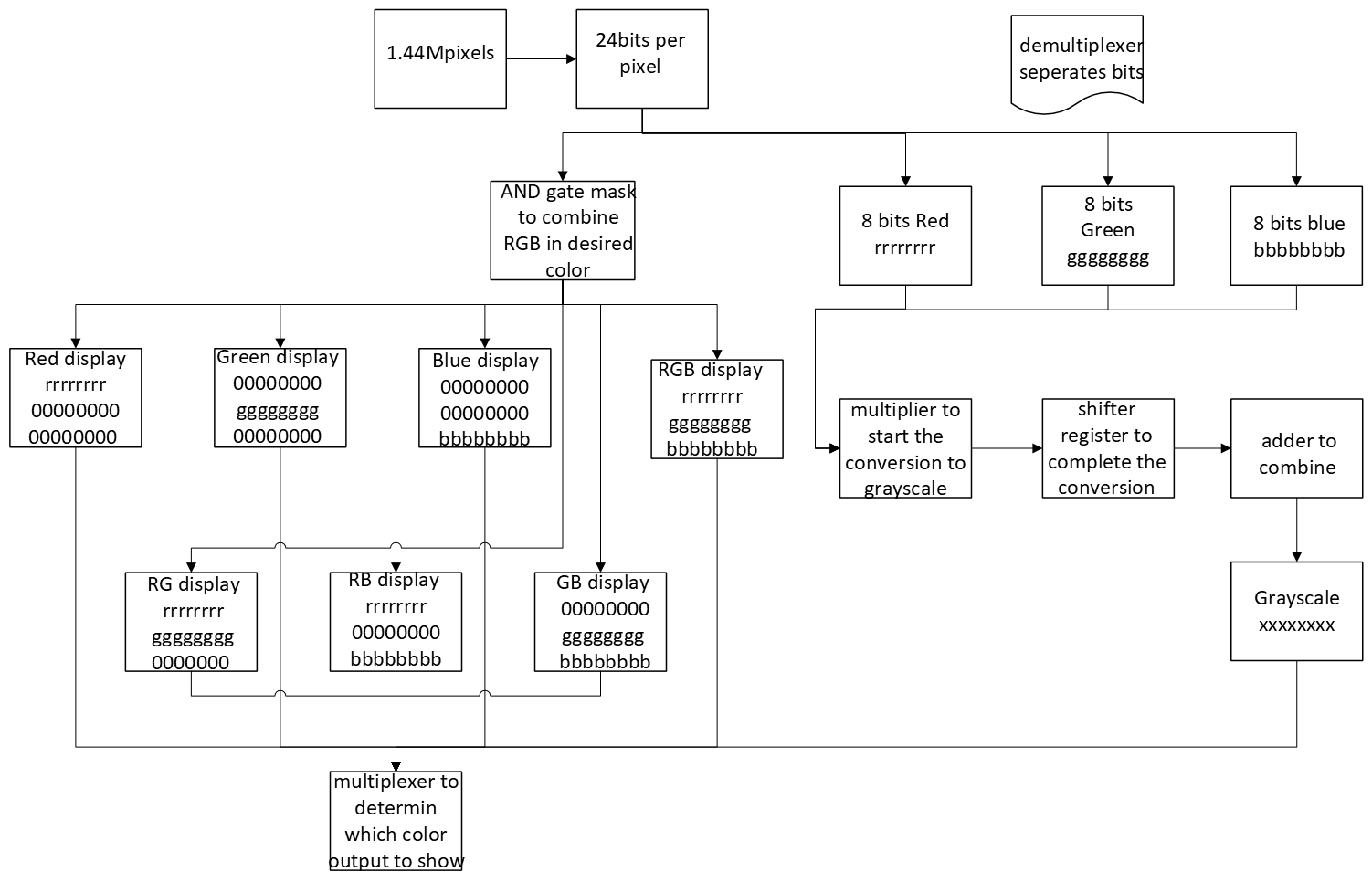


Color Filter



1. The camera takes a 1600x900 JPEG image using a simple hardware interface on the MCU to “Take pictures” and stores them on a MicroSD card.
2. The MCU converts the saved image to a Bitmap image on the microcontroller next to the original on the MicroSD via software.
3. The user selects the colors to remove (red, green, and/or blue) on the color filter interface. Users can choose up to 8 modifications. The filter must be configured before receiving the image file.
4. Assuming the user selects the “Modify Image” option on the MCU interface, the image is then sent in byte sized segments.
5. Each segment is analyzed sequentially using logic filters. In BMP images, the file contains headers, offsets, and other useful information to identify the properties of the image. It will be translated to HEX file.
   1. For example, shift registers can be used to reach offsets. If a 24-byte segment is sent, a shift register can be used to remove the 12 bytes of information not containing header data.
6. After gathering necessary information, the hardware utilizes components to manage the information.
   1. Full adders will be used to reconstruct the image segments to a full bpm image file.
   2. For instance, based on the user color selection. A multiplexer is used to determine the output. An 8-1 MUX would be used. Based on the inputs 23 = 8 for 1 output.
7. The modified image is sent to MCU where it will be stored.
8. The MCU will open the image on display and the modified image on the display using the MCU interface.

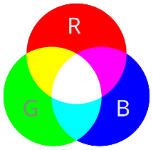
Example of Information going through Color Filter:



* MCU takes a JPEG file and converts it to Bitmap file. Why BMP?
  + Uncompressed
  + File structure is understandable:
    - Header
    - Pixels saved in an array and corresponding RGB value (24 bits total).
  + The memory system works by sending the image segments at a time processed within the filter.

Example of Images Being Filtered:

The images below are based on RGB manipulation.



Unmodified Image:



Red Removed



Green Removed



Blue Removed



Red and Green Removed



Red and Blue Removed



Green and Blue Removed



Red, Green, and Blue Removed. It should be noted that removing all colors (RGB) will result in a solid black image. In the color filter device, the image will be desaturated instead of solid black.



# Budget Preparation

The component list outlined below can change as the development process progresses. However, the list outlined below should allow the students to begin developing the project. Below is a list of the components and their corresponding cost.

*ESP32 w/ Camera Module:*

This is the microcontroller to be used in this project. This device utilizes an OV2640 (a 2 MP camera) and has the corresponding board attachment to function with the ESP32. This board also can use an external power source which would be useful for using the device without being directly connected to power.

<https://www.amazon.com/SunFounder-Kit-Development-Rechargeable-Compatiple/dp/B0D7ZY6ZS8?crid=3UPJNXY39YMTQ&dib=eyJ2IjoiMSJ9.yDUZ4sdK2BVim_7v4ziWIkNupSlmltyORIkRuiM7Ogy53TgIwI44D-8wUpfBvQW6i0f1h7j5cK1IhPnR6x9Sj-rID9vy-S7KgB5MzZI_RGRsdLD59uLNaMj_yVTWwPOVCZc02pWKEVXqgorQIWpYxed18mfY3VH4XxF4go1em3BgsoWzZLIBH-bhrkCX-4sh.owZbeSnFe4OwyflBBMHTcL8jkZ5H0sUOqafUEsOUxBk&dib_tag=se&keywords=esp32+camera+model&qid=1731988372&sprefix=ov2640+esp32+camera+model%2Caps%2C265&sr=8-17>

*DSD TECH 1.8 Inch TFT LCD Display Module with SPI Interface for Arduino and MCU:*

This is the display attached to the MCU to show the images taken from the camera and the modified images from the color filter.

<https://www.amazon.com/dp/B07WDJ3TV6?ref=ppx_yo2ov_dt_b_fed_asin_title>

*Button Potentiometer*:

This piece allows the user to interact with the color filter device. Initially, these will be used solely for their button mechanism. Since they can be used as knobs, they could later be used to specify the RGB amount outlined in the following section.

Link: <https://www.amazon.com/Tnuocke-Vertical-Position-Switches-SS12D10-G5/dp/B099MRCDG8?crid=BRNXGF7KTHCP&dib=eyJ2IjoiMSJ9.MQ2nhxvnhZhW2kZnfccF13qEOMHaLYvQcj1Jglq4EoaxtDR4-lMzmVCpe0h-mY6R3z3in-nvCuwp_CITv7icodYF18heISfHmJpAv-P9cFWY0q5imPxh_1lyQGCxlVIhl3aZh_FPcvRopOPed7otsx3Cu2PYt3fkUAWpXt7qg8XbpGgFCyNVNJ-kmVNoYEIO1bMwJcY3yS4Ajoe2z1_w7xU4gjW6bwnu8879DUIMKlg.xyvrAgClB2llRvpGZeeFCvy2-0hKN8hZpiMw2_LnA04&dib_tag=se&keywords=electronic%2Bswitches&qid=1733436598&sprefix=electronic%2Bswit%2Caps%2C204&sr=8-4&th=1>

*MicroSD:*

The ESP32 can be stored onto MicroSD cards. These will be how the images will be stored on the MCU and on the color filter device.

<https://www.amazon.com/Adapter-Memory-Tablet-Console-TF162/dp/B0CYT2KL98?crid=36I1XBJ441YEZ&dib=eyJ2IjoiMSJ9.NxBLjwblV3x-TOUO9wDC7qFudSTbAGC1JwRzton8Nn5rGLUpqfdHn-U6SIUHmyo-wlMSks8oMpvdtSAeDWJlTCdHYwwymXJ2FpxRy9CC2JB59Z4hJ36gNJ3m4uAIt3xuhAXguLQwspqoMWu-6yU2Emqzsv1FxMZtKuFC36QFbffYB-o1K1HYW62oYrYxoKUXaCmvlesODVxd1G4nkj-WXOL1XqmCHKzHzR_fufBRr9M.xHxpiVB2JDhrkYXcitq3KaPN-XgbBTOXZMcCY8SQruw&dib_tag=se&keywords=8%2Bgb%2Bsmall%2Bmicrosd%2Bcard&qid=1732336862&sprefix=8%2Bgb%2Bsmall%2Bmicrosd%2Bcard%2Caps%2C130&sr=8-1&th=1>

The components outlined below are possible components to be used in the Color filter device. A brief description is also given. The amount of the components to be used depends on the development of the Color filter device in Step 2. A rough draft of the amount needed is multiplied to the cost below. All the parts in that section are more placeholders and references than the exact parts that we will be using.

*De/Multiplexors:* These components are going to be used to separate and combine the color signals of red, green, and blue. Each color is separated into 8 bits and will be 24-bit binary number that then will be broken into its separate parts to do conversion then put back together.

<https://www.jameco.com/z/74HC4051-Major-Brands-8-Channel-Analog-Multiplexer-Demultiplexer-DIP-16_88102.html?gQT=1>

*Multipliers:* These will be used to do part of the conversion from full color to greyscale. Staying in binary to be able to multiply a fraction a multiplier is needed. These will manipulate the values of red, green, and blue. With the help of a shift register it will make the picture into grayscale

<https://www.mouser.com/ProductDetail/Texas-Instruments/CD4521BEE4?qs=gb35HGp1gQL0Y9JWxiK0lA%3D%3D&mgh=1&gQT=1>

*Adders:* These will be used to combine the red, green, and blue signal. After doing the conversion to greyscale you need to combine the numbers back together to produce the greyscale color.

<https://www.digikey.com/en/products/detail/rochester-electronics,-llc/CD74AC283E/13505834?gQT=1>

*Shift Register:* These will be used to do part of the conversion from full color to greyscale. Staying in binary to be able to multiply a fraction a shift register is need. This will complete the conversion from color to grayscale by shifting the multiplier output. The shift registers and multipliers together complete the full conversion.

<https://www.sparkfun.com/products/13699>

AND gates: These will be used as filters. Taking a mask and putting the red, green, and blue signals through the AND gate with the mask will filter out the colors that are not wanted.

<https://www.gocarelectronic.es/en/producto/semiconductors/ic-standard-logic-gates/texas-instruments/cd4081be?gQT=3>

*Resistor Variety:*

Ideally the school will give us some registers to use for this project. In that case we cannot use them. This variety box will suffice.

Link: <https://www.amazon.com/BOJACK-Values-Resistor-Resistors-Assortment/dp/B08FD1XVL6?crid=1DBOGVYWHBTN7&dib=eyJ2IjoiMSJ9.8EDLQctdt9NzqYQqREI7ygaqitd_OjXmmDyTfflxOH0oLB0G2fEG8yzkmv7QPSS9qFZ0mIdDCNLBFzWLA9TldjUjfNPXVFYOWMoi6GysEwDORDZ77WdvLnA_8awMd1bVNG4HB3h-Z2izRucijS2rlJ1Z3nB20Vwy7EA5ufGXKSOt-azQvjucIvk_1Nt3ya_uF5SxZuxjoup5-4VvpmAniZMlfCXBmyu4sDXglTQPMIA.1705gDATowg8W9H4nJ4WMwxcBD8neAOlmLic3tlCoxw&dib_tag=se&keywords=resistors&qid=1731559203&sprefix=resistors%2Caps%2C156&sr=8-3>

*Breadboard:*

A breadboard would be used to place all the components during the prototyping phase. After the basic functions are done, a PCB will be created.

Link: <https://www.amazon.com/ELEGOO-Breadboard-Solderless-Breadboards-Electronics/dp/B0CXF1B6GB?crid=1FK4FA6W3G3D8&dib=eyJ2IjoiMSJ9.rc8qs1F3n1h1lbbxRPJcmV0zEGiLfOk12Nk_YWb-tnG0ofqfosSSQU6QBxHJ-B2WO2eByc3kQ83JtEPwXwLr74rO8L0QkENJgxN4IDjZtRfl_tel1zc0ciedHFmcPn5ci4osdbV1kZF5bSB1Z7ugBkeD66ByhAIFFex0QPidpJ1geVGb5PNEv8hqH50zu-atupQtdUplN5gLAetooDv8Dv6Ot5py9nBqNTw_13clyrw.m9c-14pf0aEv6I5nbIODbAqsy-IJqeGBzsTyZq6PwKA&dib_tag=se&keywords=Breadboard&qid=1731559649&sprefix=breadboar%2Caps%2C170&sr=8-3>

*PCB Printing*: According to our professor, we can design and print a PCB on campus. This would just reduce the cost of the materials. However, if we are not able to, one would be commissioned to be created.

<https://www.pcbway.com/>

Price of parts as of writing 11/22/2024

|  |  |
| --- | --- |
| Part | Cost |
| *ESP32 w/ Camera Module* | 27 |
| *DSD TECH 1.8 Inch TFT LCD Display Module* | 9.99 |
| *Switches/Buttons* | 7.99 |
| *MicroSD* | 6.79 |
| *Multiplexers* | 2(2.95) = 5.9 |
| *Multipliers* | 3(.94) =2.82 |
| *Adders* | 4(5.12) =20.48 |
| *Shifters* | 3(1.05) =3.15 |
| *AND* | 36(0.42) =15.12 |
| *Resistor Variety* | 9.99 |
| *Breadboard* | 8.99 |
| *PCB* | Refer To Above Description??? (20-50) |
| Total | 118.22 |

In summary the total cost for a single device is $118.22 not including the PCB and tax/shipping. With the PCB the cost could increase to approximately $160. It should be noted that the total price will vary as the components used in the color filter device change. In this chart, we are assuming these will be the parts list in the color filters device. Something else to mention is, two of these devices would be created to allow efficient collaboration. By having two devices, this would allow work to be done simultaneously. The cost would double to $320. Based on this table, development will be expensive and the cost changes as development progresses. These estimates assume multiple factors such as product pricing are set, and component list is set. Ideally, the price is below this and the total cost will be noted while in development.

# Project Scope and Timeline

The project has been recently created and is still under review. A graph of the timeline done for the semester of Fall 2024 is seen below. The previous projects denied were:

* Automated Greenhouse
* Smart House
* Real-Time Clock with Alarm Moisture Sensor

Once this project is approved, the next sections can be completed for the Spring 2025 semester.

**A screenshot of a project

Description automatically generated**

**Why this project?**

This project introduces students to digital image processing and file formatting. Since images are stored digitally, students would have to learn in detail how images are stored to be able to manipulate them. Furthermore, since the device performs the color processing on a dedicated device, students would have to understand dataflow. Alongside this, embedded systems would be used to develop a user-friendly interface and interconnect components like the MCU, Camera, Storage devices, and the color filter device.

**Development Process:**

The development process can be broken down into 4 parts, assuming all the components are collected.

1. The Microcontroller implementation.

In this part of the development process, the developers will be focusing on setting up the microcontroller. Since the microcontroller being used is the ESP32, the Arduino IDE using Python and C will be used. The microcontroller should be able to attach to an external camera, store images locally (on board memory/external memory), and display the image on an external display. Furthermore, a simple interface should be created for the MCU for users to use. The interface would allow users to take images, send images to the color filter device, and open received files from the filter.

1. The Color Filter Device

To be able to develop this device multiple programs could be used. Such as VHDL and ModelSim to simulate the Color filter device. Since VHDL is a Hardware Description Language, the components used in the program can be converted to their physical counterparts and/or the physical components can be simulated in the programs. When starting out, only one-color filter will be developed. Simultaneously, a memory management system will be created to allow the image file received from the MCU to be accessed, modified, and transferable. This will be done with a sizeable storage device internal/external. Afterwards, a button interface will be created on the device itself to allow the user to toggle the filter developed and send the image to the MCU. Assuming a single filter function, the remaining ones will be added along with the corresponding interface. Finally, based on the software prototypes, a list of exact components can be created. These will then be set up according to their prototype. Testing will be done to make sure the device works as expected.

1. Testing

Testing will be done concurrently with the process of step 1 and 2, but extensive testing will be done once both steps are completed. After developing the Microcontroller and the Color filter device, both will be connected to each other. When connected, the interface of both devices will be tested by making sure they achieve the expected results. Pressing the red filter button on the color filter device enables/disabling. Besides testing the interface, we would make sure that the RGB filter is working as expected. This will be done by reviewing the images displayed and the image files directly. One example of a test we will conduct is enabling the red and blue filter on the color filter device. The expected result should be a green image.

1. Optimization

After completing the previous steps, optimization can begin. Optimization could include pricing where we choose better or cheaper components. Another optimization could be done in the processing, this can be found by using simpler components, advanced components, redesigning the code/hardware and more. Clearly multiple optimizations can be done, but it would depend on when devices are completed.

**Future Project Idea:**

Outlined below is a brief description of future add-ons.

* **User Interface:** After successfully implementing the button interface, more controllability over the RGB can be added by using knobs.
* **Live Feed:** The version being implemented is image processing. By being able to manipulate a single image, it should be achievable to translate the same process for a live feed since a video is just a series of images.