COLOR DETECTING AUTOMATIC PAINT DISPENSER

ECE 445 INDIVIDUAL PROGRESS REPORT - FALL 2024

Lucky Konatham

Professor: Arne Fliflet

TA: Chi Zhang

Table of Contents

1 Introduction

1.1 Project Description:

The Color Detecting Automatic Paint Dispenser is designed to automate the process of mixing paint by detecting a color and then dispensing the required colors in precise proportions. The device enables users to “scan” a surface color using an RGB color sensor. The sensor detects RGB values using photodiodes sensitive to red, green, blue, and clear light. This data is sent to the microcontroller (MCU), which calculates the CMYK values from the RGB input using formulas. These CMYK values translate to specific amounts of cyan, magenta, yellow, black, and white paint to be dispensed through peristaltic pumps. Each pump is controlled by a motor driver linked to the MCU, which outputs PWM signals based on the calculated time for each color proportion. By automating these steps, the dispenser reduces the complexity and inaccuracy artists face in traditional color mixing.

1.1 Individual Responsibilities:

My responsibilities in this project span multiple areas, including the design and assembly of the PCB, component procurement, programming the microcontroller, and creating project documentation. Each of these areas plays a key role in the overall functionality and integration of the project.

- PCB Schematic Design: I created the schematic for our custom PCB, focusing on designing a centralized board to connect the RGB sensor, microcontroller, and motor drivers. The PCB enables seamless data flow between these components, ensuring that the color detection and dispensing functions are integrated.

- Component Procurement: I selected and ordered all necessary components, choosing parts that met reliability and performance requirements. I prioritized parts with stable specifications and long-term availability to avoid issues during assembly and testing.
- Microcontroller Programming: I developed the code for the STM32 microcontroller, which receives RGB values from the sensor, converts these to CMYK proportions, and sends PWM signals to the motor drivers. These signals control the paint dispensing based on the exact color proportions required.
- Documentation: I wrote the proposal and most of the design document, detailing each subsystem and providing a high-level overview of requirements and objectives. This work has helped maintain clear goals and communication across the team.

2 Design

2.1 Design Considerations:

During the design of the PCB, I prioritized several aspects to ensure efficient operation and ease of testing. Below are some of the major considerations I addressed.

- Signal Integrity: With various data and power signals on the PCB, preventing interference was a priority. I added capacitors near the microcontroller and motor drivers to reduce noise and stabilize voltage. Signal traces were kept as short as possible to reduce potential issues, particularly between the high-power motor drivers and the sensitive RGB sensor.

- Component Layout: I arranged high-current components, such as motor drivers, and low-current components, such as the RGB sensor, to avoid interference. The layout also minimizes trace lengths between critical components to improve signal integrity.
- Power Management: The system operates with a 24V power supply, but different components require different voltage levels. I incorporated voltage regulators to convert 24V to 5V and 3.3V, providing a stable power supply for the MCU and motor drivers. Each motor driver has individual capacitors to improve stability, especially during operations that demand high power.

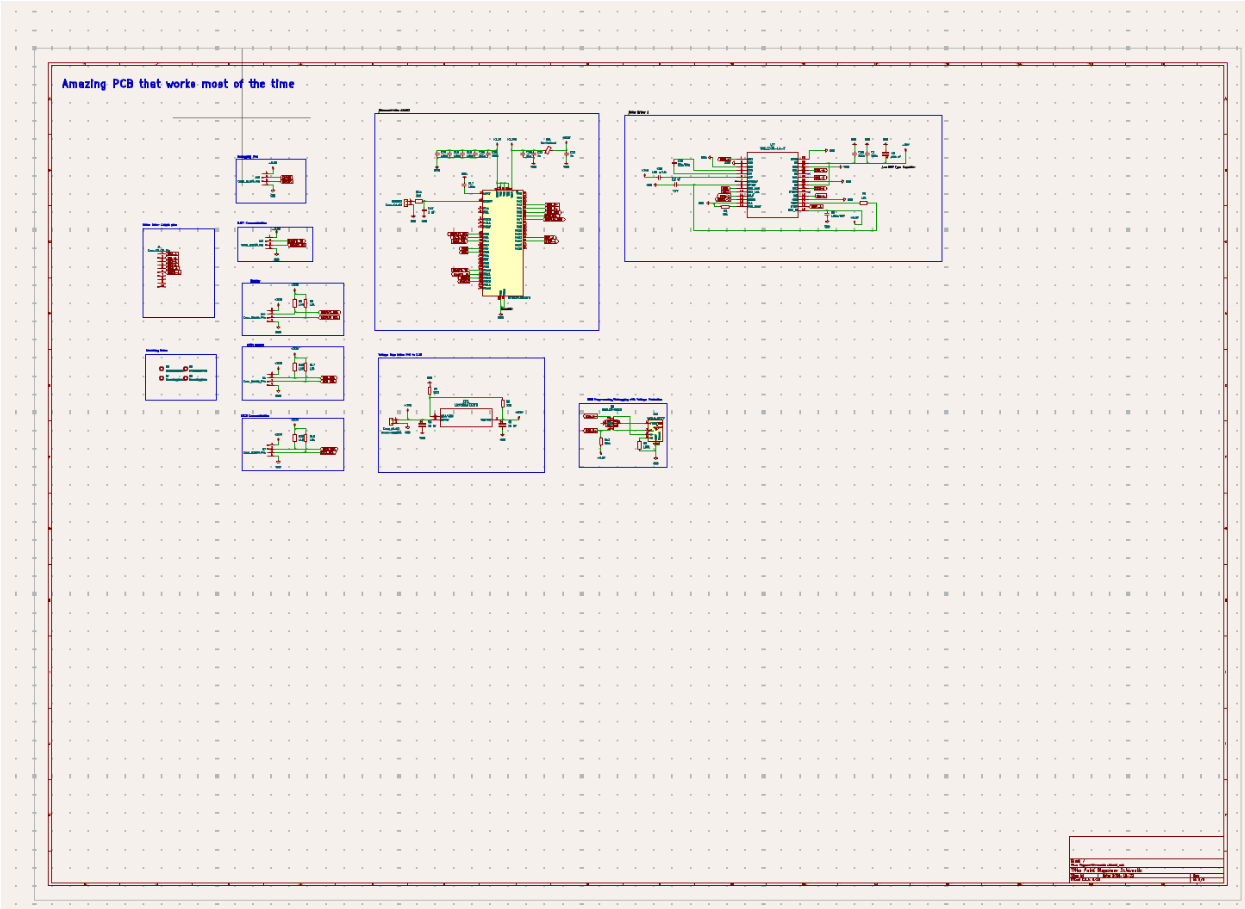
After designing the initial schematic, I refined the layout based on test results. For example, I added additional ground paths to stabilize the system and adjusted component placements to improve accessibility for testing.

2.2 Diagrams and Schematics:

The PCB schematic and code flow were structured to support efficient operation and accurate color dispensing.

- PCB Schematic

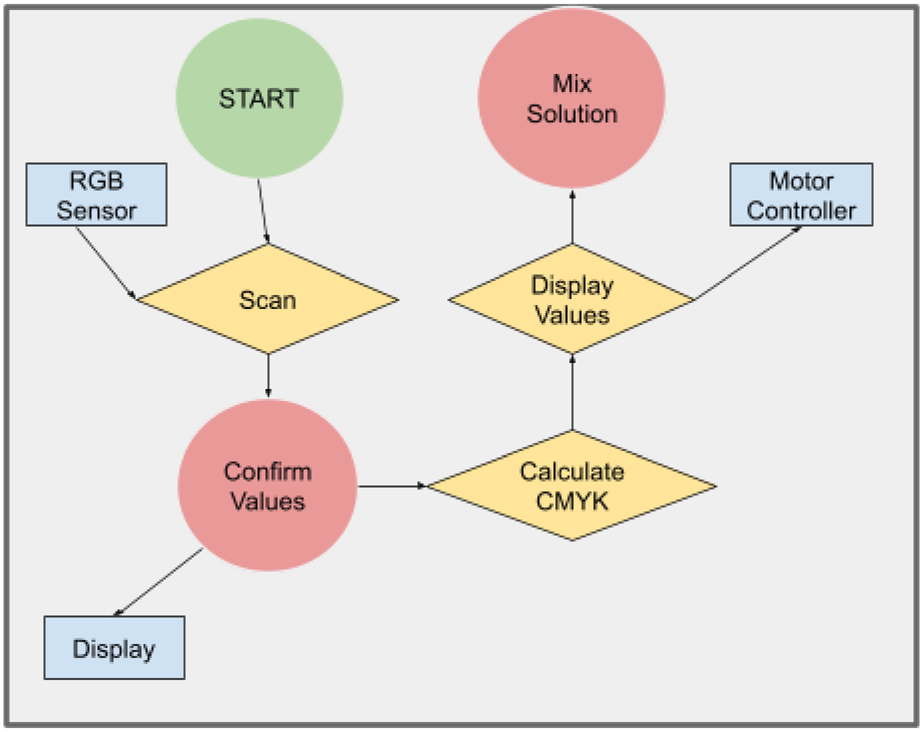
  The schematic includes the RGB color sensor, STM32 microcontroller, motor drivers, and voltage regulators, each connected through dedicated traces and connectors. The design also separates power and ground planes, reducing interference and improving overall stability. The PCB is compact and logical, with a focus on signal isolation.



- Microcontroller Flow and Code

  The microcontroller code is organized into two main parts: a color detection part and a motor control part.

  - Color Detection: The RGB sensor scans the target color, converting it to RGB values. These values are then converted into CMYK using standard formulas.
  - Motor Control: Based on the CMYK values, the MCU sends PWM signals to each motor driver, specifying how long each pump should operate. The PWM duty cycle is calculated to ensure that each motor dispenses paint according to the required ratios.



Microcontroller Code

The microcontroller programming encompassed several tasks essential to the operation of the paint dispenser:

- RGB to CMYK Conversion

  The conversion from RGB to CMYK follows an algorithmic approach to accurately map detected colors. For each scan, RGB values are first normalized, then applied to calculate CMYK values using the following formulas:

  - $R' = R / 255$, $G' = G / 255$, $B' = B / 255$
  - $K = 1 - \max(R', G', B')$
  - $C = (1 - R' - K) / (1 - K)$, similarly for M and Y
  - These CMYK values determine the ratios of each primary color.

```cpp
#include <Wire.h>
#include <Adafruit_TCS34725.h>

// Define pins for motor control (adjust according to your STM32 setup)
#define CYAN_MOTOR_PIN 2
#define MAGENTA_MOTOR_PIN 3
#define YELLOW_MOTOR_PIN 4
#define BLACK_MOTOR_PIN 5
#define WHITE_MOTOR_PIN 6

// Initialize the color sensor
Adafruit_TCS34725 tcs = Adafruit_TCS34725(TCS34725_INTEGRATIONTIME_50MS, TCS34725_GAIN_4X);

// Variables for RGB and CMYK values
float R, G, B;
float C, M, Y, K;

// Setup function for initialization
void setup() {
  Serial.begin(9600);
  if (!tcs.begin()) {
    Serial.println("No TCS34725 found");
    while (1);
  }
  pinMode(CYAN_MOTOR_PIN, OUTPUT);
  pinMode(MAGENTA_MOTOR_PIN, OUTPUT);
  pinMode(YELLOW_MOTOR_PIN, OUTPUT);
  pinMode(BLACK_MOTOR_PIN, OUTPUT);
  pinMode(WHITE_MOTOR_PIN, OUTPUT);
}

// Function to read RGB values from color sensor
void readColor() {
  uint16_t clear, red, green, blue;
  tcs.getRawData(&red, &green, &blue, &clear);
  R = red / 255.0;
  G = green / 255.0;
  B = blue / 255.0;
}
```

```cpp
// Function to convert RGB to CMYK
void convertToCMYK() {
  K = 1 - max(R, max(G, B));
  C = (1 - R - K) / (1 - K);
  M = (1 - G - K) / (1 - K);
  Y = (1 - B - K) / (1 - K);
}

// Function to control motor PWM for paint dispensing
void dispensePaint(float c, float m, float y, float k, float white) {
  analogWrite(CYAN_MOTOR_PIN, c * 255);
  analogWrite(MAGENTA_MOTOR_PIN, m * 255);
  analogWrite(YELLOW_MOTOR_PIN, y * 255);
  analogWrite(BLACK_MOTOR_PIN, k * 255);
  analogWrite(WHITE_MOTOR_PIN, white * 255);

  delay(5000); // Adjust delay based on required paint volume and calibration
  analogWrite(CYAN_MOTOR_PIN, 0);
  analogWrite(MAGENTA_MOTOR_PIN, 0);
  analogWrite(YELLOW_MOTOR_PIN, 0);
  analogWrite(BLACK_MOTOR_PIN, 0);
  analogWrite(WHITE_MOTOR_PIN, 0);
}

// Main loop function
void loop() {
  readColor();
  convertToCMYK();
  float whiteVolume = 1.0; // Adjust white base volume as per requirements
  dispensePaint(C, M, Y, K, whiteVolume);
  delay(10000); // Wait before next dispensing cycle
}
```

- Motor Control Implementation

  PWM signals control each motor’s duty cycle, adjusted to achieve the correct paint volume based on the calculated ratios. Each motor’s run time is precisely timed, considering the pump’s flow rate to ensure that the dispensed volume is accurate to within a 5% margin of error.

- Error Handling and Calibration

  Error-checking routines were implemented to verify sensor readings and motor performance. Calibration steps included:

  - Verifying sensor accuracy against standard color samples.
  - Testing motor consistency by dispensing timed amounts of paint and adjusting PWM as needed for variations in motor speed or response.

3 Verification

Testing and verification were conducted in multiple stages, from individual component tests to system integration.

I verified the PCB’s functionality by checking connections and confirmed that all connections were intact across the PCB.

The microcontroller code was tested by inputting different RGB values to simulate the color detection process. Key steps included timing and flow rate adjustment: adjusted motor timings and PWM values based on flow rate tests, ensuring that each motor dispenses paint volumes within a 5% error margin.

To confirm the system’s overall functionality, I conducted integrated tests on the color detection subsystems for color detection consistency: repeated tests with various color samples to ensure the sensor accurately detected RGB values.

In the future, I will still have to conduct tests on the following:

- Dispensing Accuracy: Using a colorimeter to measure the color accuracy of the dispensed paint, confirming it matched the input color within a defined tolerance level.
- Latency Monitoring: Measuring the time delay between color detection and motor response to ensure prompt dispensing, reducing delays that could impact precision.
- I2C Communication: Testing communication between the RGB sensor and microcontroller, ensuring accurate data transfer.

4 Conclusion

1. Self-Assessment

   My role involved a significant portion of both hardware and software responsibilities, from schematic design and programming to system testing. Although the workload has been intensive, I have managed to stay on track, even as testing required more time than anticipated. This additional time has proven valuable, yielding important data for refining the system.

2. Remaining Work and Timeline

   The remaining tasks include final PCB refinements, completion of full-system tests, and a thorough accuracy check. These are scheduled to be completed within the next two weeks, followed by a final calibration phase and user testing to gather feedback on usability and functionality.

3. Ethical Considerations

   The project adheres to the IEEE and ACM Codes of Ethics, focusing on reliability, quality, and environmental responsibility. The primary ethical considerations are:

   - Accuracy and Reliability: Extensive testing ensures reliable operation across conditions, minimizing the risk of errors affecting users' work.
   - User Safety: Safety precautions in both hardware and software help prevent malfunctions and protect users from potential hazards.
   - Environmental Responsibility: The dispenser minimizes paint waste through accurate dispensing, supporting sustainable practices in line with ethical standards on environmental conservation.

5 Citations

A. West, *Treaty of Paris*, 1783. [Online]. https://commons.wikimedia.org/wiki/File:Treaty_of_Paris_by_Benjamin_West_1783.jpg . [Accessed: Oct. 1, 2024].

Robu, robu.in/wp-content/uploads/2024/03/10021-1.pdf. Accessed 6 Nov. 2024.

St, www.st.com/content/ccc/resource/technical/document/datasheet/53/db/00/58/09/98/4b/36/CD00001884.pdf/files/CD00001884.pdf/jcr:content/translations/en.CD00001884.pdf. Accessed 6 Nov. 2024.

STM32F103x8 STM32F103xB - -Based 32-Bit MCU with 64 Or ..., www.st.com/resource/en/datasheet/stm32f103cb.pdf. Accessed 6 Nov. 2024.

TMC2209 Datasheet, www.analog.com/media/en/technical-documentation/data-sheets/TMC2209_datasheet_rev1.09.pdf. Accessed 6 Nov. 2024.