*Color Reader*

Product Design Specification

Designer: *Joe Ma / Tung Ho*

Version: *0.1*

*04/22/2020*

TABLE OF CONTENTS

[1 Introduction 3](#_Toc38378770)

[1.1 Purpose of The Product 3](#_Toc38378771)

[2 General Overview and Design Guidelines/Approach 3](#_Toc38378772)

[2.1 Assumptions / Constraints / Standards 3](#_Toc38378773)

[3 Architecture Design 3](#_Toc38378774)

[3.1 Hardware Architecture 4](#_Toc38378775)

[3.2 Software Architecture 4](#_Toc38378776)

[3.3 Communication Architecture 4](#_Toc38378777)

[3.4 Sensor/Parts Description 4](#_Toc38378778)

[4 System Design 5](#_Toc38378779)

[4.1 Bill of material (BOM) 5](#_Toc38378780)

[4.2 Calibration and test procedures 5](#_Toc38378781)

[5 Conclusion on Product Design Specification 6](#_Toc38378782)

[Appendix A: References 7](#_Toc38378783)

[Appendix B: Key Terms 8](#_Toc38378784)

# Introduction

## Purpose of The Product

The purpose of this product is to convert the color of real-world items into hex equivalent values that can be used in graphic design. The TFT display shows the RGB in 16bit hex values and the approximate color in RGB565.

# General Overview and Design Guidelines/Approach

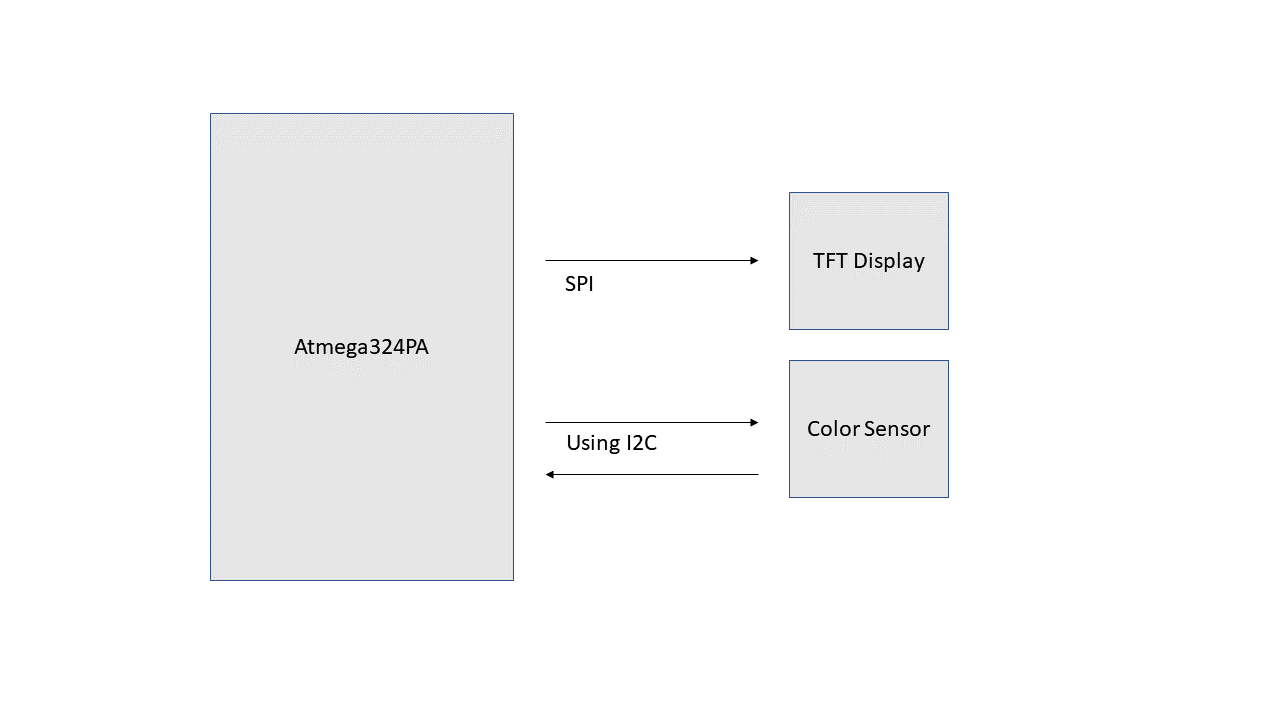
The general use of this product is stated in 1.1. When designing the product, ease of Implementation was the top priority.

## Assumptions / Constraints / Standards

Constraints: There is a minimum height between the sensor and the color. The height is roughly 0.5 inches. Any less will result in an overly white color sensing, and any more will result in an overly black color sensing. Also, the integration time for the color is 700ms, so when the display displays one value it is already integrating another value.

Standards: Colors are displayed in 16bit hex values. Using the higher byte of each can create RGB888. The display will show an approximate color in RGB565.

# Architecture Design



The MCU initializes the color sensor through I2C and the display through SPI. It then calls for the color sensor’s data through I2C. Each color is stored on separate high and low byte registers within the sensor. Accessing that information, the data is also stored separately within the MCU as different variables. The high and low bytes are converted to a single 16bit hex value and sent to the display. A conversion takes place to make those color values into RGB565 and is also sent to the display. A matching color to that RGB565 value is generated as well.

## Hardware Architecture

The color sensor is an I/O device. The TFT display is used strictly as an output.

## Software Architecture

The color sensor sends data to the MCU after being initiated. Conversions from individual 16bit colors to RGB565 is done in the MCU. The MCU then sends all that data to the TFT display.

## Communication Architecture

The color sensor communicates to the MCU through I2C, and the display communicates using SPI. Those are the only supported methods of communication for the sensors. The MCU acts as the middleman where the color sensor will feed the MCU I2C data, and the MCU will send that data through SPI to the display.

## Sensor/Parts Description

Color Sensor: Flora Color Sensor with White LED – Adafruit Product ID: 1356

Link: https://www.adafruit.com/product/1356

The color sensor featured is the TCS34725. It communicates via I2C and runs on a 3V rail. It features IR filters and a dynamic range. It can sense RGB and clear light. Adafruit says this is the best color sensor on the market.

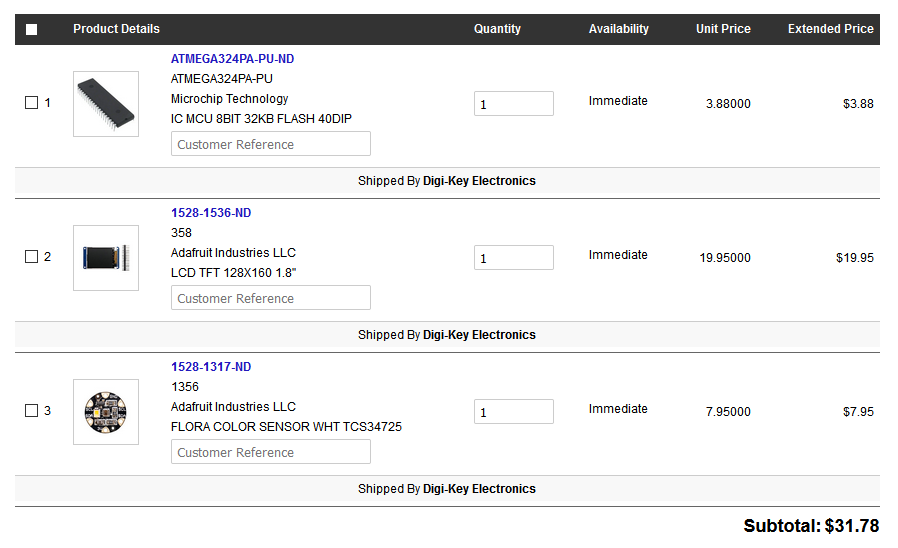
TFT Display: 1.8” TFT Display – Adafruit Product ID: 358

Link: https://www.adafruit.com/product/358

The TFT display features the ST7735R driver. This display is 128\*160 and features 18bit color. This certain board contains internal regulators and step-downs so that 3.3 and 5V work with it. It communicates via SPI.

# System Design

## Bill of material (BOM)



## Calibration and test procedures

The sensor used was factor calibrated. Testing only consisted of having the sensor “look” at a few colors I had lying around like white, blue, and green.

# Conclusion on Product Design Specification

There was another color sensor that was supposed to be used. There was a communication issue with it, so that is why the Flora was used. The Flora had a more in-depth datasheet and was easier to communicate to. There was a library for the display which made things go way better. The experience was not bad if it were not for that first sensor. Some challenges that were faced was the TWI/I2C not stopping. The datasheet asked for an acknowledgment, and everything just breaks. That was interesting.

The project met a good chunk of the specifications that were first laid out. The sensor was able to communicate with the MCU and all that data was fed to a display.

The main thing that could be improved is by using more interrupts. The initial plan was to use a button and have it cause an interrupt which will turn on the color sensor and works its magic to the display. This fell through since the display library used implicit interrupts and would have issues when sei() was on. This will also help with power consumption. There is a better workaround, but we were happy to get the project running.

The last thing that could be improved is more solid connections. Using jumper wires is not ideal since there was a lot of flickering but soldering it might help a lot.

Appendix A: References

The following table summarizes the documents referenced in this document.

|  |  |  |
| --- | --- | --- |
| **Document Name and Version** | **Description** | **Location** |
| *RGB565 Color Picker* | *RGB565 Color Picker* | *http://www.barth-dev.de/online/rgb565-color-picker/* |

Appendix B: Key Terms

The following table provides definitions for terms relevant to this document.

|  |  |
| --- | --- |
| **Term** | **Definition** |
| *RGB565* | *Digital color code for 16bit color. There are 5 bits of red, 6 bits of green, and 5 bits of blue. Ex: 0xFFFF is white.* |
| *RGB888* | *Digital color code for 24bit color. There are 8 bits of red, 8 bits of green, and 8 bits of blue. Ex: 0xFFFFFF is white.* |