Dr. Wei-Chuan Shih

W328 Engineering Building 2

Houston, TX 77204

Dear Dr. Shih:

This document is the final report on our senior design project entitled “μ-Vision” for the Fall semester. The project is progressing according to plan aside from minor setbacks in color recognition. This task has proven to be more involved than anticipated, but has progressed since the previous progress report. Outlined in this report are completed tasks and progressing goals that are available to show in the form of figures and images to assure the project is continuing as planned. Our overview diagram is updated to match the one displayed on our full size poster, and a detail of the sorting mechanism was included.

Sincerely,

Kevin Tazehzadeh

Frank Nnam

Kris Griffith

Noah Shubber

µ-Vision

Frank Chibuisi Nnam

Kris Griffith

Noah Shubber

Kevin Tazehzadeh

Faculty Advisor: Dr. Wei Shih

Abstract

This report details the final progress report for the Fall semester towards the completion of the μ-Vision iOS application and automated conveyor belt system. This project has a dual purpose; the standalone iOS application can identify resistor values individually, and the automated resistor detection system can sort a chosen resistor value from others. This will be done with the help of what will be referred to as a “dot lens”. The dot lens is a small lens that attaches to a smartphone camera for magnification purposes. After a rigorous study of the Objective-C programming language,a user interface has been created, the application can access the phone camera, the application compiles utilizing all required frameworks, and has a refined camera interface. For control of the autonomous sorting system an Arduino Leonardo microcontroller equipped with a Bluetooth module for communication with the iPhone is being used. The semester goal to have interfacing between the Leonardo and the iPhone has been completed, and coding for basic color detection on the application in conjunction with the image processing framework called OpenCV has been completed.

Purpose and Background

The purpose of this project is to bring recognition to the ECE department of the University of Houston, as well as simplify resistor identification. This will be achieved through the utilization of a small dot lens that attaches to the camera of an Apple iPhone in order to magnify small resistors, and the creation of a resistor-identifying iOS Application. The application will feature a standalone resistor identification function, as well as a resistor sorting function. The purpose of the standalone application is to allow electronic circuit designers and those who are colorblind to easily identify values of resistors while utilizing the cheap dot lens. The resistor sorting portion of the application will utilize the same essential resistor identification software, but will include the control of a conveyor belt, a DC servo controlled gate, and two bins for sorting correct and incorrect values. The conveyor will be powered by a DC motor that will be hooked up to the Arduino, and controlled by the iPhone. Ultimately, once combined these elements will allow the user to place a resistor on the conveyor belt, enter the resistor value they would like sorted, and wait as the resistors are sorted into the designated bins.

Problem, Need, and Significance

Currently a quick and effective way of identifying the values of resistors does not exist. In a world changing rapidly by mobile technology, making daily tasks more efficient and less time-consuming has become the norm. Being able to quickly identify and label values of circuit elements such as resistors in a time-pressed environment not only reduces calculation error and improves efficiency, but allows for users who might not have previously been able to identify resistor band colors due to being colorblind or otherwise visually impaired to participate in physical circuit construction.

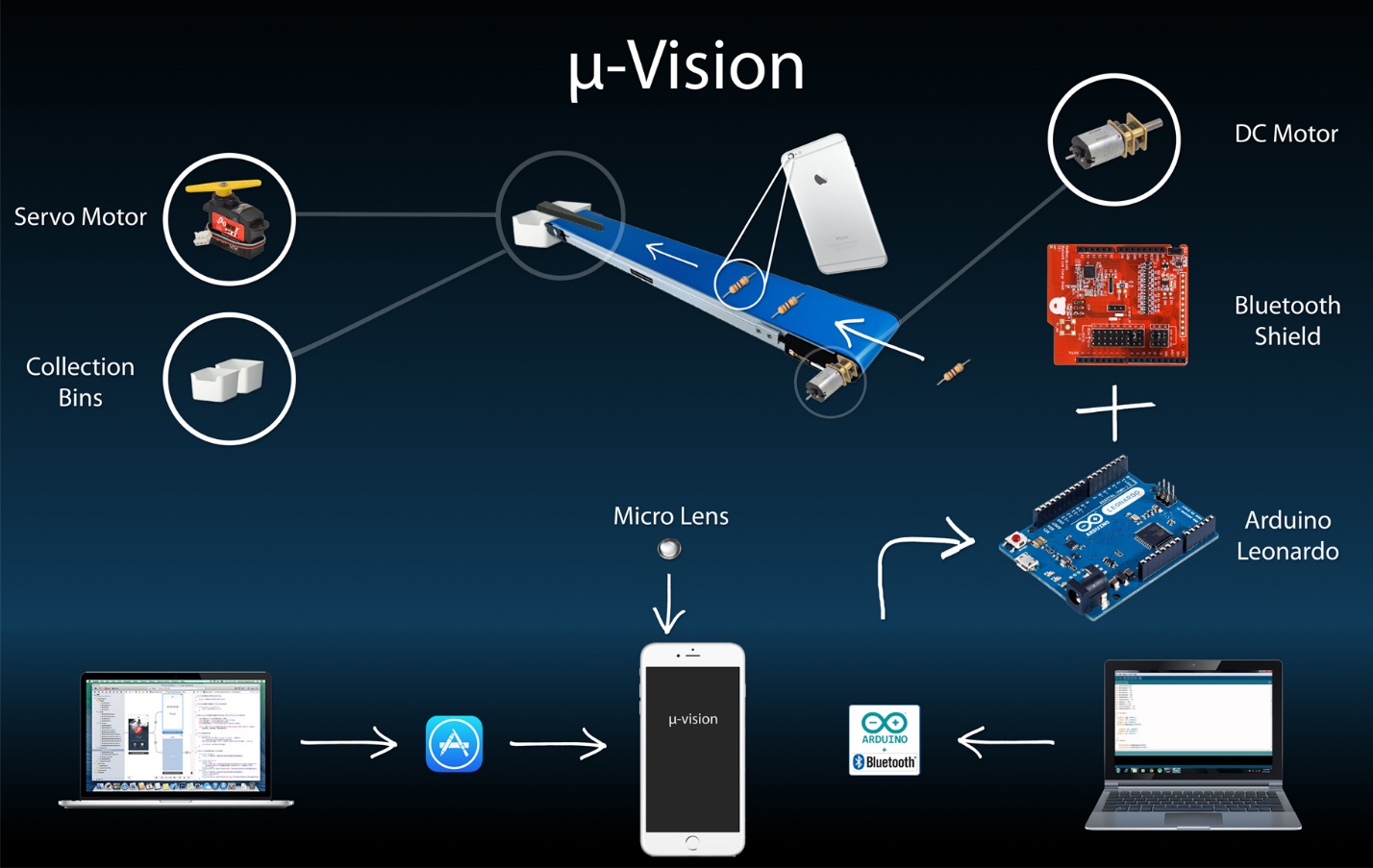
Through creation of the iOS application and automated conveyor belt system for resistor sorting, the user will benefit from faster resistor identification, especially users who are colorblind. Students using the application will save time in circuit construction and mass resistor sorting is severely simplified.

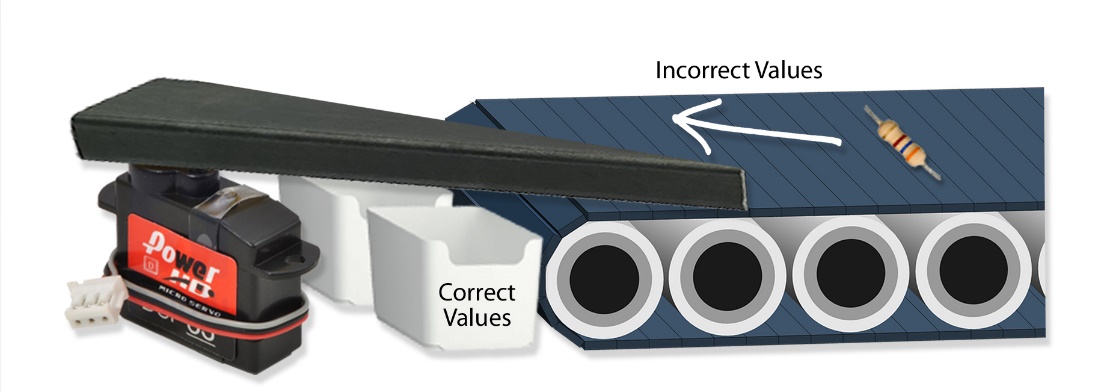
User Analysis

The target audience expected to use the application and sorting system are those interested in physical circuit construction. The expected user is also constrained to those who own iPhones as opposed to other platforms such as Android, since the application software is coded in Xcode for the iOS platform. The dot lens is also necessary to precisely identify resistor colors on small resistor casings. The level of expertise required for use of the application is minimal; the user must download and open the app and place the resistor in front of the camera for automated identification.

Overview Diagram

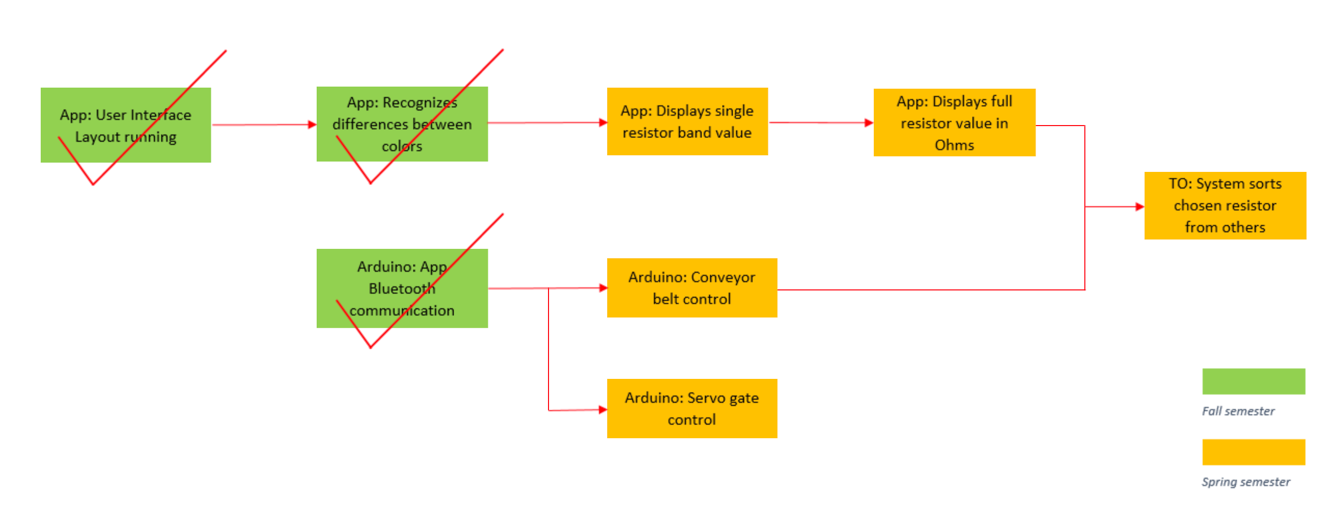
Figure 1, below, is an overview of the components being utilized in the project. The application will be coded on a MacBook using Xcode, as can be seen from the bottom left, and will be submitted through the Apple App store, and downloaded on the iPhone. Also, the Arduino Leonardo and the Bluetooth shield are programmed to receive commands from the iPhone. The Arduino will be controlling the servo motor, and the DC motor that powers the conveyor. A stand will be constructed and added to the conveyor belt, so that once the value is selected by the user, the autonomous sorting can begin. Figure 2 shows a blown up view of the sorting mechanism which receives commands from the Arduino to block off one of the sorting bins so that correct values are sorted out from incorrect values.

  
Figure 1. Overview diagram

  
Figure 2. Detail of Sorting Mechanism

Target Objective and Goal Analysis

The target objective for this project is to sort chosen resistor values autonomously utilizing an iOS application. Shown below, labeled as Figure 3, is the Goal Analysis for the project. The boxes shown in green indicate fall semester goals, while the yellow boxes indicate spring semester goals. A goal with a check indicates successful completion.

  
Figure 3. Goal Analysis

Specifications and Constraints

The specifications and constraints for each goal shown above in Figure 3 are shown below in numerical order for organization.

1. The User Interface (UI) layout running on the application was the first goal of the project.

* Specifications
  + Application runs 12 times without crashing on 3 different iPhones
  + The UI has a start screen, menu screen, and navigational menu options
* Constraints
  + The limitations of the built-in table view controller for the menu
  + For application approval by the App store it must not crash during operation

1. “The Application recognizes differences between colors” is another semester goal of the project, and is currently being worked on.
   * Specifications
     + The application will categorize colors by the following HSV table, labeled Table 1.

Table 1: HSV Values

|  |  |  |  |
| --- | --- | --- | --- |
| Color | Hue | Saturation | Value |
| White | 0 - 180 | 0 - 25 | 0 – 255 |
| Black | 0 – 180 | 0 – 250 | 0 – 50 |
| Brown | 0 – 15 | 90 – 250 | 10 – 100 |
| Red | 0 – 8 | 150 – 250 | 0 – 255 |
| Orange | 8 – 12 | 150 – 250 | 0 – 255 |
| Yellow | 20 – 30 | 130 – 250 | 0 – 255 |
| Green | 45 – 72 | 50 – 250 | 0 – 255 |
| Blue | 90 – 130 | 50 – 255 | 50 – 255 |
| Violet | 130 – 155 | 50 – 255 | 50 – 255 |

* Constraints
  + Light reflection (Camera Flash or Sunlight)
  + Low light environments causing additional color variation

1. The Arduino sends and receives commands from the iPhone
   * Specifications
     + Communication successful within 10 feet
     + Communication via Bluetooth
     + RSSI should be within -26dBm and -80dBm
   * Constraints
     + Coding language differences between Arduino and iOS Application

Test Plan and Progress Description/Statement of Accomplishments

In order to design a user interface that met specifications, Xcode “Table-View” controllers, navigation controllers, and the necessary frameworks were used. Most importantly the OpenCV framework, also known as open source computer vision, was included. This framework allows for the application to be able to recognize that a particular object is in front of it which is in this case a resistor. This was necessary for the formation of the UI, because it must not make the application crash, but is associated with a future goal of the project. The application has a welcome start screen, mode selection option, and a modified full-screen camera layout. The combined interface, shown in Figure 4 and Figure 5, was run on three different iPhones in order to meet the specifications. The application was opened and closed twelve times successfully, and deleted and re-downloaded to ensure functionality.

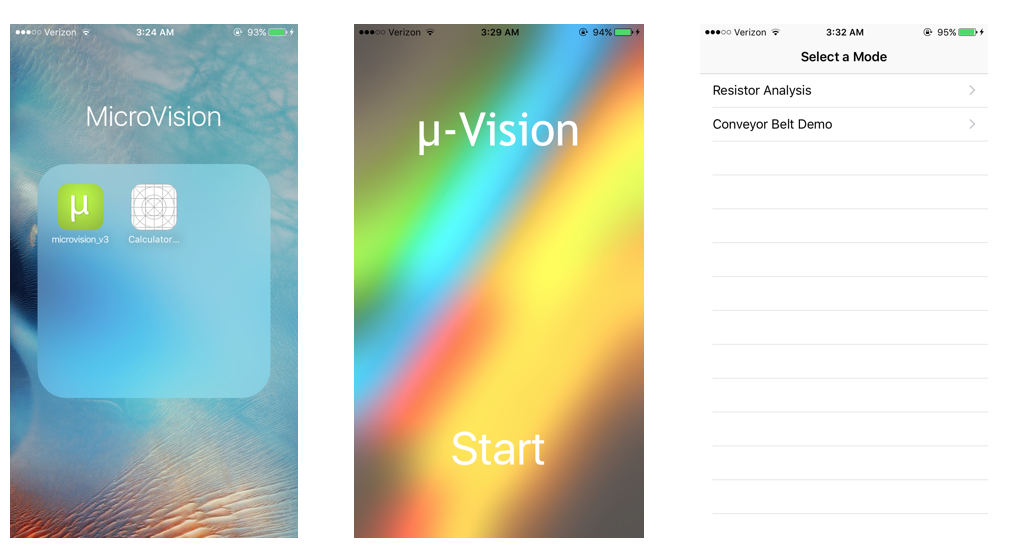


Figure 4. User Interface on iPhone

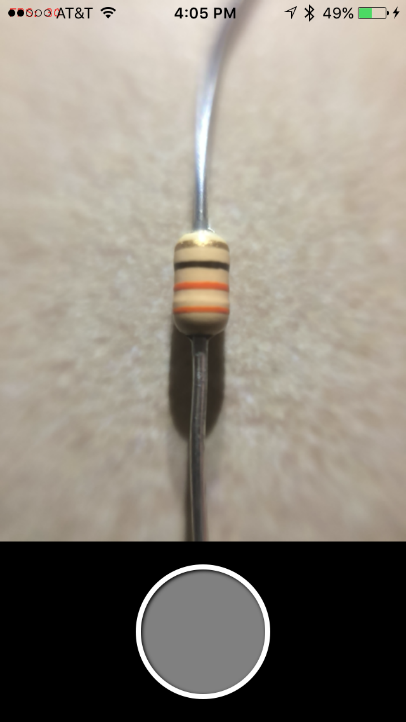


Figure 5. User Interface on iPhone

The second goal, which is the “Arduino Leonardo communicating with the iPhone through Bluetooth”, has been completed. Although the Bluetooth Shield module that would connect with the Leonardo did not arrive at the specified time, the semester goal was met. The Bluetooth module that was chosen to be used for this project is manufactured by Red Bear Labs and is called the BLE shield. The BLE shield board is designed to fit on several Arduino control boards including the Leonardo. The inclusion of this chip enables the Arduino board to receive and send data wirelessly via Bluetooth. To establish and test communication, we obtained the open source Arduino libraries from Red Bear Labs required to initialize the Arduino ports for use with the BLE shield. Arduino uses the C/C++ programming language which all members of the group are familiar with, so understanding which libraries were necessary was easily completed, and it was deduced that only two libraries, RBL\_nRF8001 and BLE, were needed. Once these libraries were added to the standard libraries that come with the Arduino IDE software, example code from the RBL\_nRF8001 library was loaded onto our Arduino Leonardo with the BLE shield connected. The code is written to work with multiple open source applications that are also provided by Red Bear Labs for iOS developers that are creating Bluetooth enabled applications. The specific application named BLE controller was downloaded onto our iPhone to test the connectivity. This will be added to our application in the future in order to complete future goals, but was not done for this specific goal. Figure 6, shown below, shows the BLE controller recognizing the BLE shield, along with the MAC (media access control) address and RSSI (received signal strength indication).

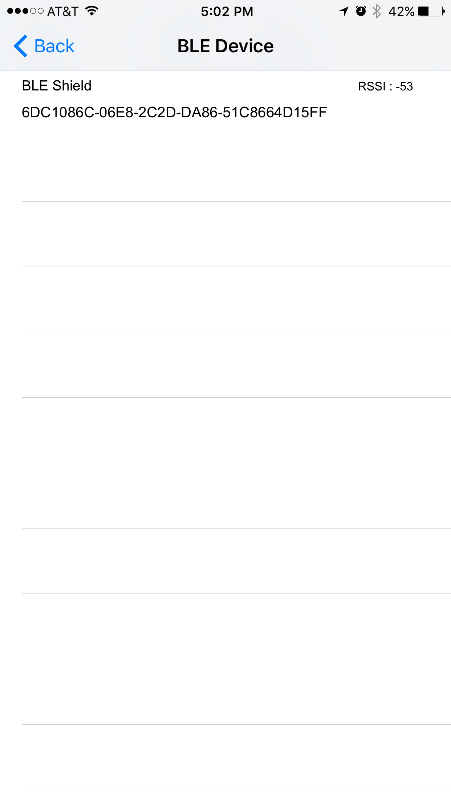
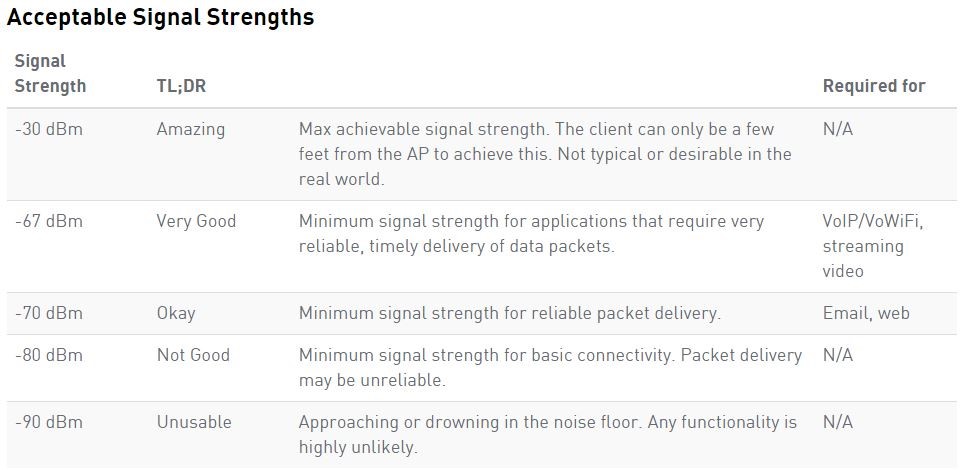


Figure 6. BLE controller application page

RSSI is a scale that is not standardized. The IEEE suggested scale goes from 0-255 [1], however every manufacturer uses their own scale. The Apple RSSI scale is no different, and is based off of this same Decibel-milliwatts [dBm] scale. The more negative the number the weaker the signal is. RSSI on mobile devices are usually between -26 and -100 [3]. Our RSSI was within that range with the weakest signal being at -71dBm when 10ft away with no obstructions and based on the table below our RSSI is sufficient. Table 2, shown below, explains

Table 2. Showing RSSI values and their quality [2]



The screenshot below, labeled Figure 7, shows a replication of the pins on the microcontroller and their real time status. A page, which displays the controls that we can perform on the microcontroller from the phone, is displayed below. We can also perform read/write operations and PWM to the controller pins. During testing we found that only one device can be connected to the BLE shield at a time, this means no other device can interrupt the communication with the Arduino once it is commenced.

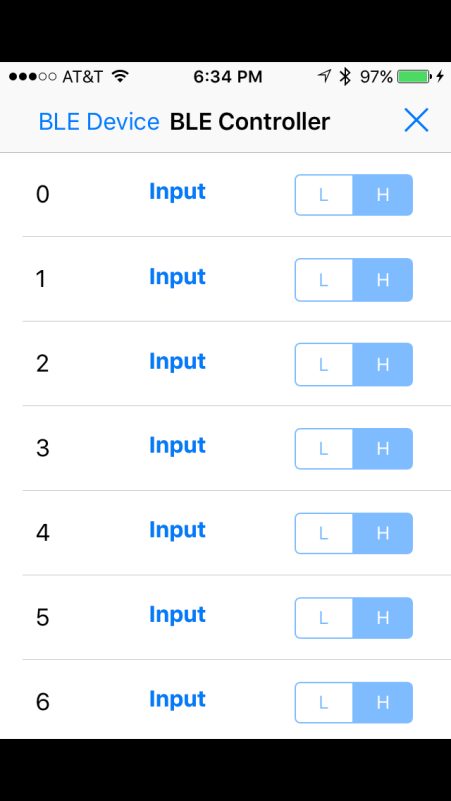


Figure 7. iPhone Input Options

For the final goal of the semester, the application recognizes different colors that are placed in front of it. The utilization of an image detection algorithm which finds colors set within a threshold is currently being worked on. The process of color detection is a lengthy one taking longer than expected, but aided by the use of the OpenCV framework for image processing the camera can be utilized as a sensor for detection. Since the application will need to work with moving objects such as resistors on the conveyor belt, the framework is helpful in recognizing and tracking colors as the resistors pass by. Color isolation and removal is also possible with this framework and will be crucial in separating resistor color bands from each other. For example, if a green sheet of paper that does not encapsulate the entire screen was held in front of the camera, the application would detect that the sheet is indeed green, and locate the green sheet disregarding background images. This effect can be seen in Figure 8 where the camera was held in front of a color spectrum displayed on a computer screen. When the application was coded to detect and filter the color red, the image on the far right shows the result displayed to the screen. The color blue is shown in the middle which comes from the same process. This goal has been completed and will help to achieve future goals set for the project. Currently, threshold levels have been found for various colors, and the OpenCV framework is being studied to incorporate the object/color tracking aspect. Colors can be detected by use of HSV, which stands for hue, saturation and value. Hue is representative of a pure color on the color wheel, saturation relates to how white a color is, and value describes how dark a color is.

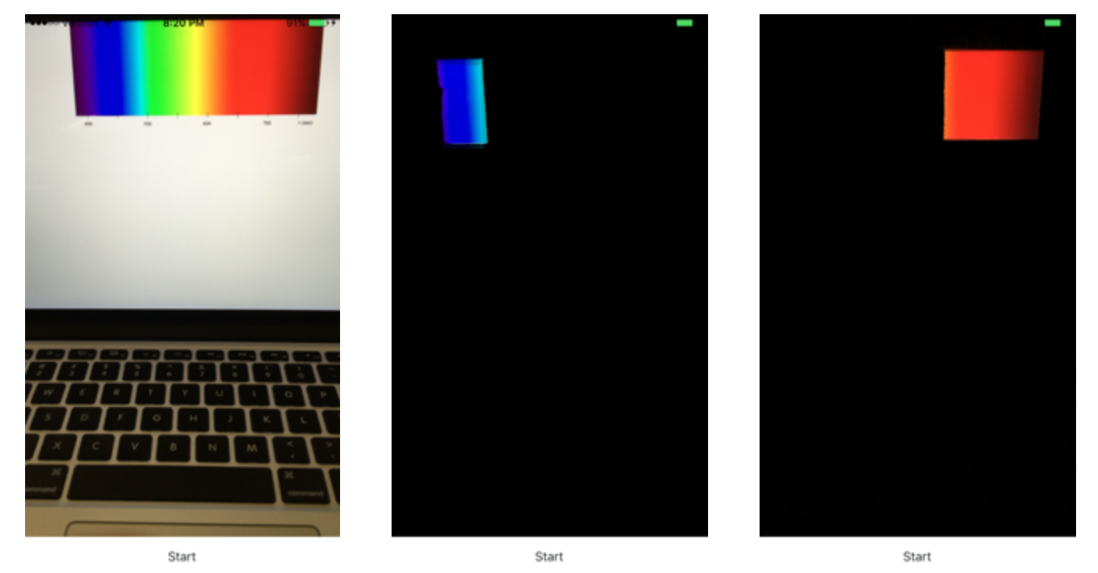


Figure 8. Color Isolation

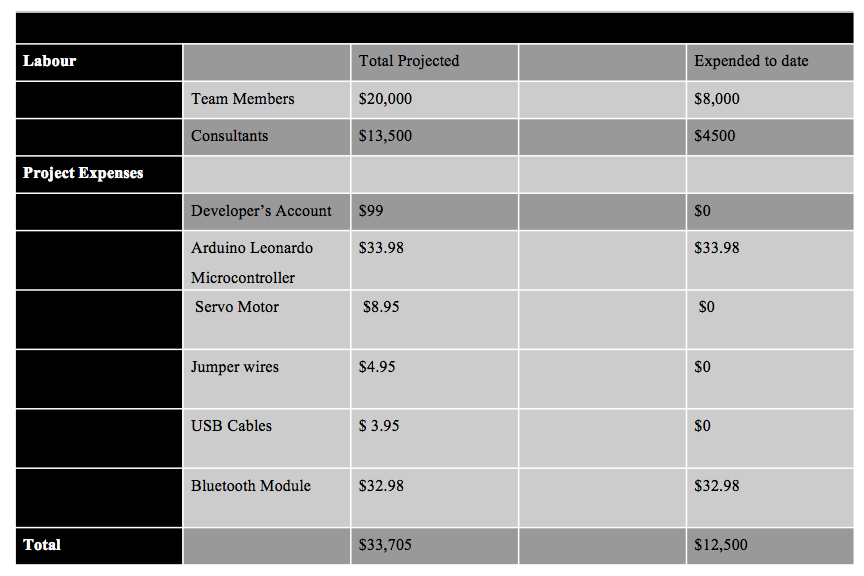
Engineering Standards

1. Bluetooth IEEE 802.15.1: WPAN / Bluetooth: This standard is a communications specification for wireless personal area networks (WPAN). This is relevant to our project due to the fact that the iPhone communicates with the Arduino via Bluetooth.
2. ISO/IEC 9899:2011: This is the current C programming standard. This standard is relevant to our project because we are coding both the Arduino and the iPhone application in C.
3. USB 2.0: This is the current standard USB standard. It provides technical details to understand USB requirements for USB compatible products. This standard relates to our project because the Arduino microcontroller.

Budget

Table 3, below, shows the costs associated with the project.

Table 3. Project budget estimate



Risks

An original risk, from which we were able to recover, was the late shipment of the Bluetooth module. Although the Module arrived two weeks later than anticipated, the coding for the interfacing between the Arduino and the iPhone had been completed in advance. Once the module arrived, the interfacing was completed and the specifications were tested and met. Currently, the main associated risks with the completion of our project arise with trying to achieve the resistor color isolation aspect of the application. Although the semester goal has been achieved, the complexity of the resistor value detection has been realized, but hopefully will be completed by the end of next semester.

Conclusion

Previous to the start of the semester no member of the group had any idea of the complexity associated with the creation of an iPhone application. However, the interfacing of the iPhone with the Arduino through Bluetooth and the complete user interface for the application were completed this semester. The completion of color recognition is progressing and the goal for this semester has been met. The necessary research has been done as well as compiling the HSV values for the different colors that require detection, but full implementation of detecting multiple colors at once is still underway.

References

[1] "IEEE 802®: LOCAL AND METROPOLITAN AREA NETWORK STANDARDS." *IEEE Standards*. IEEE Standards Association, 29 Mar. 2012. Web. 13 Dec. 2015

[2] "Understanding RSSI." *Trying to Understand RSSI Levels and How to Read Them*. MetaGeek, n.d. Web. 13 Dec. 2015.

[3] "How to Check Bluetooth Connection Strength in OS X - CNET." *CNET*. N.p., n.d. Web. 13 Dec. 2015.