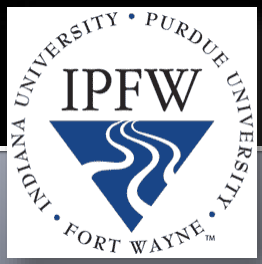
Senior Design Project II: Sorting Machine with Color Light to Digital Convertor

Andrew Costin, Tom Syster, Ryan Cramer
Advisor: Professor Hack
Instructor: Professor Lin
May 5th, 2014



1

Presentation Outline

- Problem Statement
- Introduction
- Executive Summary
- Requirements
- Project Design Activities
- Project Design Overview
- Subsystems
- Design Diagrams
- Conclusions

2

Problem Statement

- This project was meant to demonstrate the ability to electronically detect and used the color of an object to control a system.

3

Introduction

- Phase 1
  - Formed Project Team
  - Planned Project Schedule
  - Planned Project Cost
  - Identified Project Risks
  - Preliminary Requirements
  - Project Approval
- Phase 2
  - Requirement Updates
  - Document Updates
  - Detailed Research / Design
  - Hardware Implementation
  - Software Implementation
  - System Testing

4

Executive Summary

- Project Schedule
  - Design / Testing : 80 days
  - Report / Presentation : 5 days
- Cost
  - Total Cost Budget : $150
  - Total Time Budget : 450 Hours
- Deliverables
  - Functional Prototype, Project Report, and Presentation

5

Operational Requirements

- System shall sort items by color.
  - Test
- System shall have 1 mode for sorting items.
  - Inspect
- System shall sort items into separate containers.
  - Test

6

Performance Requirements

- System shall sort ≥ 12 items per minute.
  - Test
- System shall have an error rate ≤ 50%.
  - Test
- System shall not sort > 5 colors.
  - Test
- System shall have a hopper that will hold ≤ 3 oz (one standard bag of skittles) of candy.
  - Inspect

7

Physical Requirements

- System shall be ≤ 10 lbs.
  - Inspect
- System shall be ≤ 2 foot tall.
  - Inspect
- System shall be ≤ 1 foot wide.
  - Inspect
- System shall be ≤ 1 foot deep.
  - Inspect
- System shall accept skittle sized objects
  - Test

8

Environmental Requirements

- System shall operate in a dry environment.
  - Inspect
- System shall operate at room temperatures Test
  - Test

9

Functional Requirements (Part 1)

- System shall use an Arduino microcontroller.
  - Inspect
- System shall use a color detecting sensor.
  - Inspect
- System shall use a bin/hopper.
  - Inspect
- System shall use Arduino servos.
  - Inspect
- System shall use copper tubing.
  - Inspect

10

Functional Requirements (Part 2)

- System shall use Plexi-Glass housing.
  - Inspect
- System shall use hardwired interfaces.
  - Inspect
- System shall compensate for color variablility. ("S" on Skittle).
  - Inspect
- System should be powered via USB to PC connection
  - Inspect
- System should output the number of sorted items to a display.
  - Test

11

Project Design Activities

- Review and/or Update Requirements and Architecture
- Research hardware/software.
- Component Selection
- Write microprocessor software.
- Integrate parts into a system.
- Troubleshoot system.
- Test system to ensure requirements are met.
- Prepare report.
- Prepare presentation.
- Deliver presentation, report, and prototype.

12

Design Overview

- Arduino Microcontroller Subsystem
  - "Brains" of the overall system
  - Receives data from color sensor
  - Controls mechanical subsystem
- Color Sensor Subsystem
  - Integrated color light to digital convertor
  - Adafruit Color Sensor
- Mechanical Subsystem
  - Collector Servo, Output Servo, Distribution System
  - Parallax Servos

13

Arduino Microcontroller Subsystem

- System Configuration: Connected to Laptop PC via USB connection
- C++ Program implemented in Arduino Sketchpad
  - Initializes variables
  - Initializes serial communication with color sensor
    - I^2C serial Communication
  - Initializes collector rotation
  - Predefined variables for color comparison
- Controls output servo position
- Converts RGBC data to a color temperature

14

Setup Variables

- **tcs**
  - Specifies integration time and gain of the signal
- **diskServo**
  - Used to initialize the collector servo
- **dispenserServo**
  - Used to initialize the output servo
- **pos1, pos2, pos3, pos4, pos5**
  - Used to define the output positions

15

Color Sensing Variables

- **r,g,b,c**
  - Used to store the 16-bit values receive from the CLDC
- **runningTotal**
  - Used to store a running total
- **colorTempAvg**
  - Used to store a 3 sample color temperature average
- **colorTemp**
  - Used to store a single color temperature reading.

16

Counter Variables

- **redCount**
- **yellowCount**
- **purpleCount**
- **greenCount**
- **orangeCount**
- **errorCount**
- **totalCount**

17

Servo Functions

- **.attach()**
  - Connects a servo variable to a specific pin output
- **.write()**
  - Sends movement commands to a servo
- **.writeMicroseconds()**
  - Defines the pulse width sent to the servo
  - Determines the speed and direction that the servo rotates

18

CLDC Functions

- **.begin()**
  - Starts serial communication with CLDC
- **.getRawData()**
  - Takes color readings
  - Stores the color information in the r,g,b,c variables
- **.calculateColorTemperature()**
  - Converts r,g,b,c values to a single color temperature

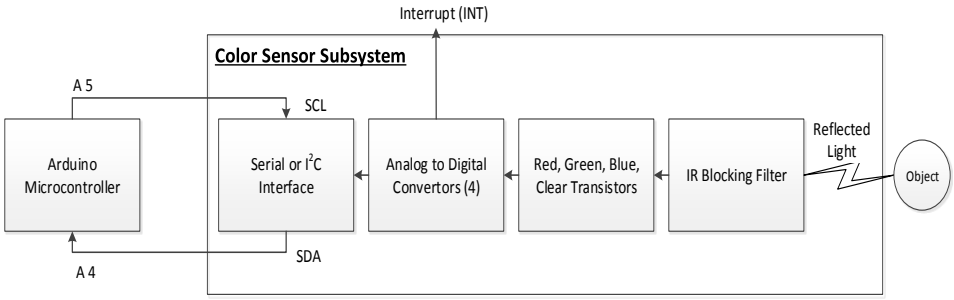
19

Color Sensor Subsystem

- Functional Blocks within CLDC
  - Color Light to Digital Convertor (CLDC)
  - LED's to illuminate detection area
  - Analog to Digital Conversion
  - RGBC data transmitted to MCU

20

Color Sensor Subsystem and Interface to Arduino Microcontroller



21

Mechanical Subsystem

- Collector Servo (Continuous Rotation)
  - Rotating Disc

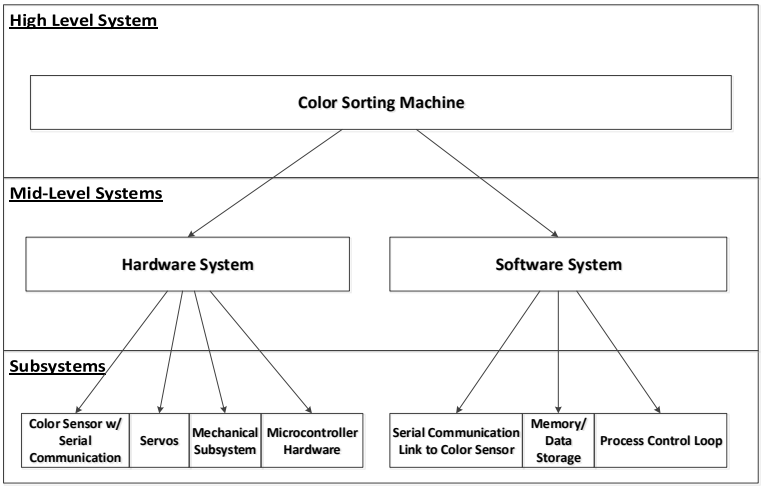
22

Mechanical System (Cont.)

- Output Servo (180 Degree)
  - Sweeping Arm (position arm)

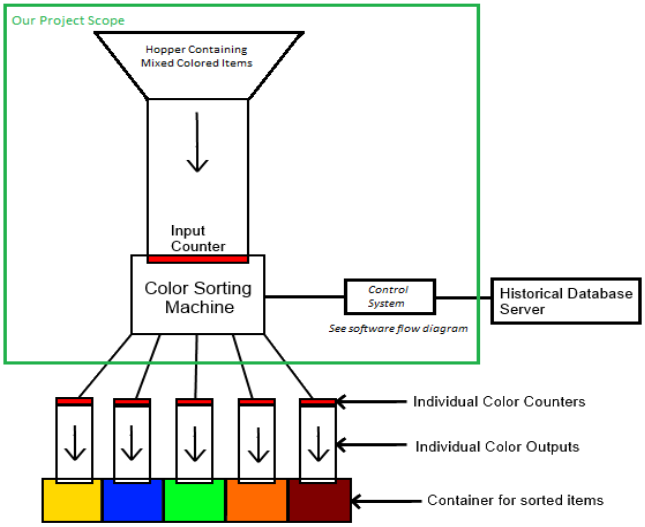
23

System Diagram



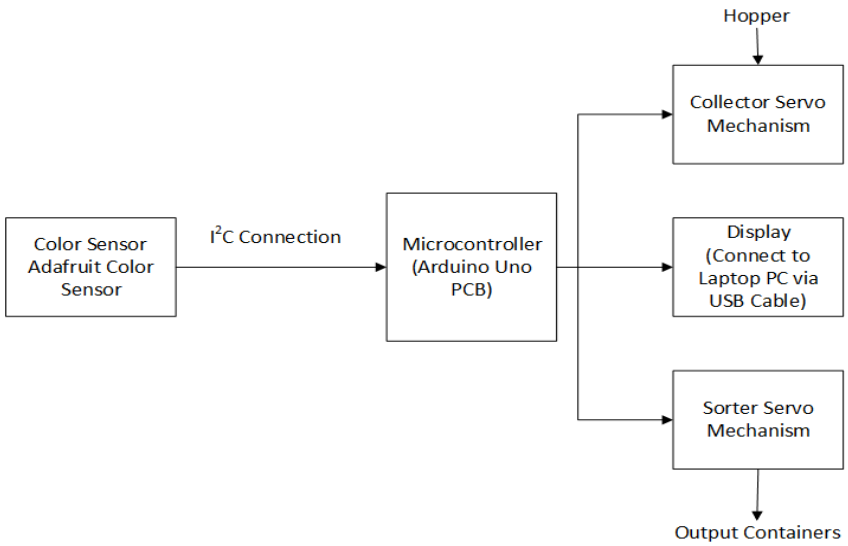
24

Architecture Diagram



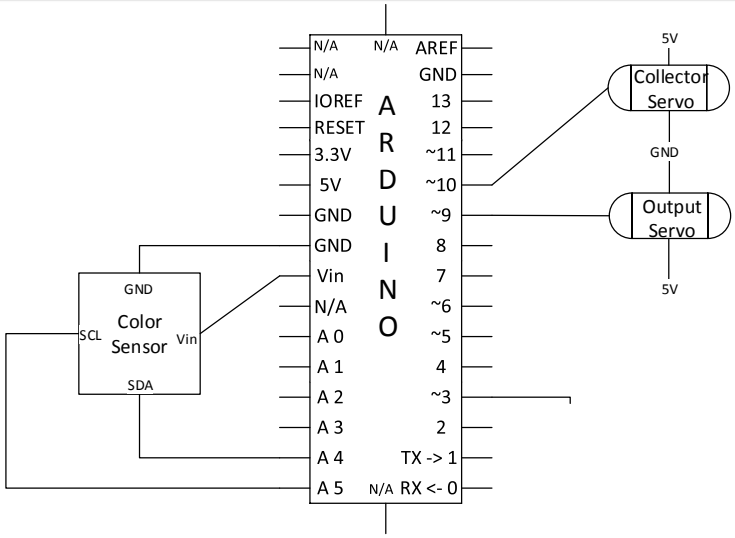
25

Block Diagram



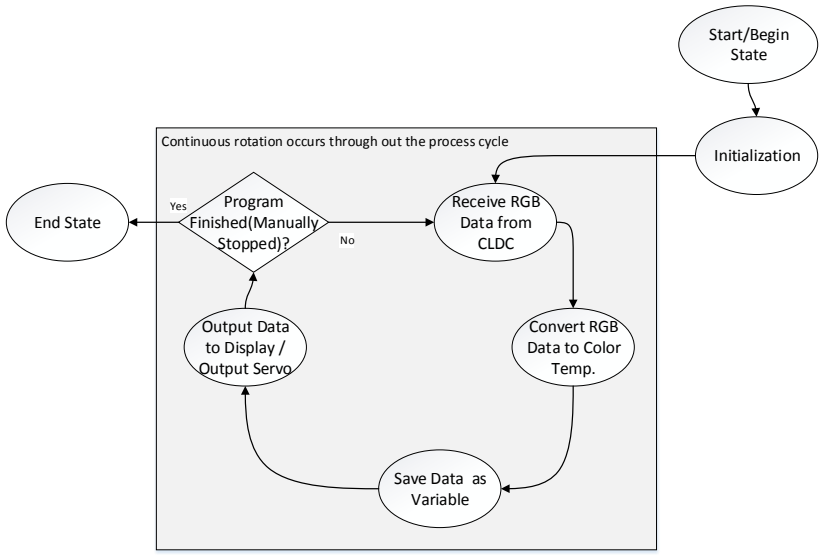
26

Circuit Diagram



27

State Diagram



28

Control Software Algorithm

Header Files
Sensor.h
- Define Sensor addresses
- Sensor Function Prototypes
  1.Get RGB Values
  2.Convert RGB Values to Color Temperature
- Will access sensor.cpp file for function execution

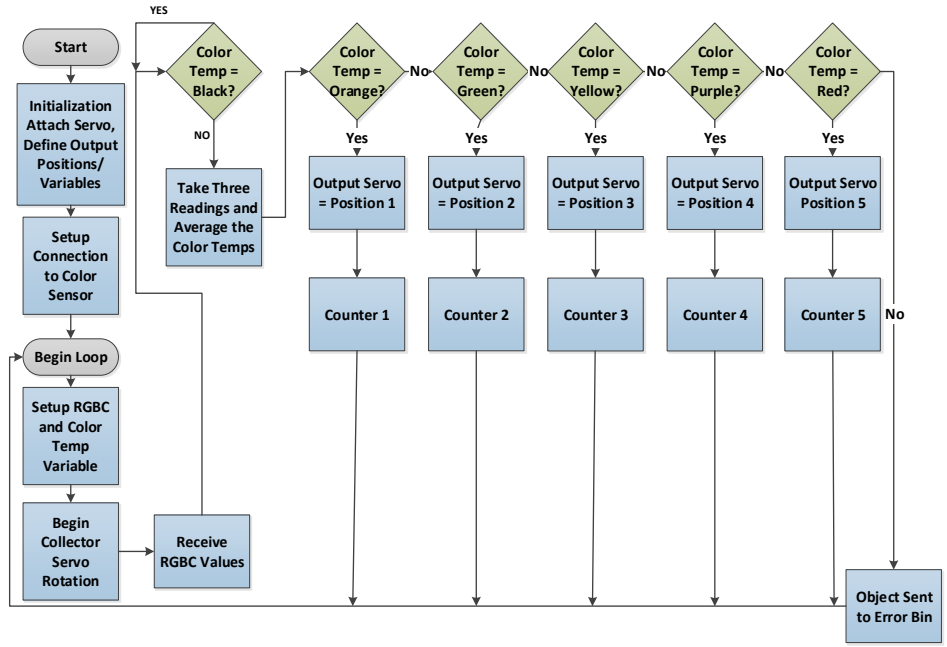
Servo.h
- Servo Functions
  1.Read Servo Position
  2.Write Servo Position
  3.Delay Servo Movement

Main File

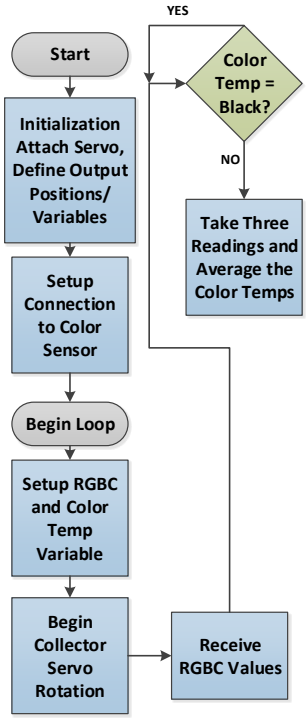
#Include Sensor.h
#Include Servo.h

- Initialization
- Servo Setup
- Sensor Setup
- Get RGB Values
- Convert RGB to Color Temperature
- Decision Making based on
  color Temperature / Color Tolerance determined from testing
- Read servo position
- Write servo position based on
  color temperature and current servo position

29



30



- Initialization
- Connect Color Sensor
- Setup RGBC
- Begin Rotation
- Check for any non-black reading
- Calculate Average Color Temperature

31

Testing Approach

Test Program Algorithm

- Get the values for Red, Green, Blue and Clear variables.
- Convert the RGB values to a color temperature in Kelvin
- The color temperatures will then be displayed on the screen for recording purposes

	Test 1	Test 2	Test 3	Test 4	Test 5
Color 1	*Result 1*	*Result 2*	*Result 3*	*Result 4*	*Result 5*
Color 2	*Result 1*	*Result 2*	*Result 3*	*Result 4*	*Result 5*
Color 3	*Result 1*	*Result 2*	*Result 3*	*Result 4*	*Result 5*
Color 4	*Result 1*	*Result 2*	*Result 3*	*Result 4*	*Result 5*
Color 5	*Result 1*	*Result 2*	*Result 3*	*Result 4*	*Result 5*

Each result is the output color temperature in Kelvin.

32

"S" Facing CLDC Test

	Test 1	Test 2	Test 3	Test 4	Test 5	Average	Variance Average %	Max
Color 1 - Red	1642	1632	1683	1695	1658	1662.0	*1.30%*	1.99%
Variance	*20*	*30*	*-21*	*-33*	*4*			
% Variance	*1.20%*	*1.81%*	*1.26%*	*1.99%*	*0.24%*			
Color 2 - Green	3663	3380	3346	3353	3262	3400.8	*3.08%*	7.71%
Variance	*-262.2*	*20.8*	*54.8*	*47.8*	*138.8*			
% Variance	*7.71%*	*0.61%*	*1.61%*	*1.41%*	*4.08%*			
Color 3 - Yellow	2469	2511	2481	2493	2449	2480.6	*0.70%*	1.27%
Variance	*11.6*	*-30.4*	*-0.4*	*-12.4*	*31.6*			
% Variance	*0.47%*	*1.23%*	*0.02%*	*0.50%*	*1.27%*			
Color 4 - Purple	2960	2982	3028	3083	2878	2986.2	*1.86%*	3.62%
Variance	*26.2*	*4.2*	*-41.8*	*-96.8*	*108.2*			
% Variance	*0.88%*	*0.14%*	*1.40%*	*3.24%*	*3.62%*			
Color 5 - Orange	1624	1641	1632	1766	1644	1661.4	*2.52%*	6.30%
Variance	*37.4*	*20.4*	*29.4*	*-104.6*	*17.4*			
% Variance	*2.25%*	*1.23%*	*1.77%*	*6.30%*	*1.05%*			

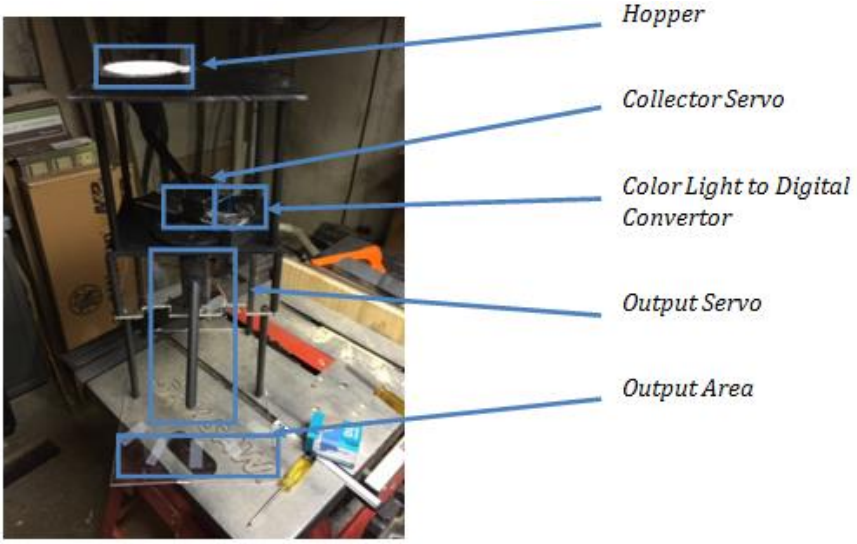
33

"S" Facing away from CLDC Test

	Test 1	Test 2	Test 3	Test 4	Test 5	Average	Variance Average %	Max
Color 1 - Red	1972	1817	1954	1845	1876	1892.8	*2.97%*	4.18%
Variance	*-79.2*	*75.8*	*-61.2*	*47.8*	*16.8*			
% Variance	*4.18%*	*4.00%*	*3.23%*	*2.53%*	*0.89%*			
Color 2 - Green	3459	3299	3305	3407	3236	3341.2	*2.55%*	4.93%
Variance	*-58.2*	*101.8*	*95.8*	*-6.2*	*164.8*			
% Variance	*1.74%*	*3.05%*	*2.87%*	*0.19%*	*4.93%*			
Color 3 - Yellow	2384	2395	2537	2506	2471	2458.6	*2.25%*	3.19%
Variance	*74.6*	*63.6*	*-78.4*	*-47.4*	*-12.4*			
% Variance	*3.03%*	*2.59%*	*3.19%*	*1.93%*	*0.50%*			
Color 4 - Purple	2998	2856	2849	3118	2848	2933.8	*3.39%*	6.28%
Variance	*-64.2*	*77.8*	*84.8*	*-184.2*	*85.8*			
% Variance	*2.19%*	*2.65%*	*2.89%*	*6.28%*	*2.92%*			
Color 5 - Orange	1687	1628	1652	1621	1778	1673.2	*2.84%*	6.26%
Variance	*-13.8*	*45.2*	*21.2*	*52.2*	*-104.8*			
% Variance	*0.82%*	*2.70%*	*1.27%*	*3.12%*	*6.26%*			

34

Prototype



35

Project Cost

(1) Arduino Microprocessor Development Board - $60
(2) Arduino Servos -$40
(?) Misc Components Etc -$20
(?) Misc Items - $30

36

Work Breakdown Structure

	Andrew	Tom	Ryan	Total
Review and/or Update Requirements and Architecture	4			4
Research hardware/software.	4	4	4	12
Design color sensing circuit.		20	10	30
Design servo control circuit.			20	20
Design mechanical system.		20		20
Write microprocessor software.	10		10	20
Integrate parts into a system.	40	40	40	120
Troubleshoot system.	40	40	40	120
Test system to ensure requirements are met.	20	20	20	60
Prepare report.	15	2	2	19
Prepare presentation.	15	2	2	19
Deliver presentation, report, and prototype.	2	2	2	6
Total	**150**	**150**	**150**	**450**

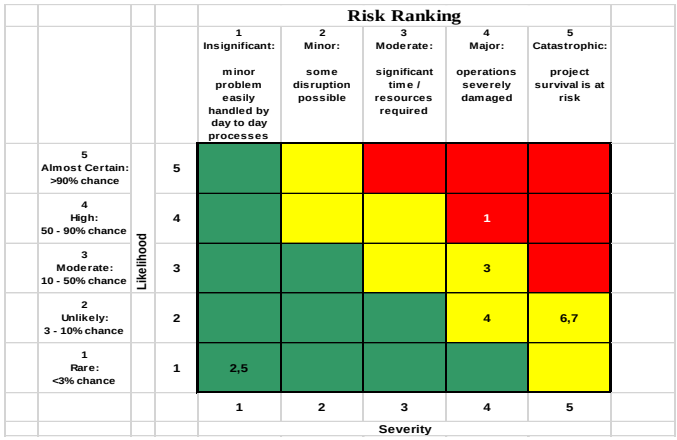
Project Advisor : Professor Hack
Project Advisor Meeting Time : Tuesdays, 5:00pm – 6:00pm

37

Top Project Risks

Project Risks

1. Color sensing errors during color detection
2. Servo position errors



38

Conclusions

- Color Light to Digital Convertor
  - Highly Sensitive
  - Changed original requirement for error rate from <5% to < 50% because of the high sensitivity
- Collector Servo Rotation
  - Originally design to rotate in 90 degree steps, but needed change to continuously rotate.
  - The collector servo would not stop accurately in 90 degree steps. Each rotation was slightly off eventually causing the hole and CLDC to be misaligned.

39

Questions

40

Demo