SHOE SORTING ROBOT

By

Jinghan Guo

Quanhua Huang

Mingxi Zou

Final Report for ECE 445, Senior Design, Spring 2019

TA: Qin Zhen

1 May 2019

Project No. 12

Abstract

This document outlines the design and construction of the Shoe Sorting Robot, with two primary areas of focus: hardware and mechanical design, and processing software. The final product is meant to be used as a house-use automatic robot that help us pick up baby shoes, clean the doorway, organize shoes on shelf, and most importantly, free our hands and save our time. The robot consists of a camera eye, a cart motor, a robot arm, a load cell, a bluetooth and a microcontroller unit. Organizing shoes is the principal goal of our senior design project, but after we successfully build the robot, this application can be extended into organizing all kinds of things like dishes, toys, and clothes.

Contents

1. Introduction

1.1 Objective

Have you ever get tripped over a shoe in you doorway? Disorganized shoes can be a mess but sometimes people just don't get time to organize shoes. Our goal is to build a house-use robot that helps people organize shoes. The main components of our robot include a camera eye for vision, a cart motor for movement, and a robot arm for grabbing the shoes. Our robot will initially awaiting next to the shelf, after people take shoes off, robot first detect the shoe, next come to the shoe, and then transfer it to the load cell, after microcontroller records the weight of current shoe, the robot will move to organizing area and place the shoe in the designated place.

1.2 High-level Requirement

- The camera must be able to detect correct number of shoes on the 20cm*100cm mat with an accuracy of 80%.
- The robot can move to the calculated position along the shoe mat, pick the shoe up, and put it on load cell with an accuracy of 80%.
- The robot can pick the shoe up from load cell and move to shelf to put the shoe in correct position with an accuracy of 80% then return to original location.

1.3 Block Diagram

Our project consists of three modules which is shown in figure 1. Power module supplies stable 5 V to every components of robot, peripheral module detects shoe and send data to robot module through bluetooth. Robot module contains all the components physically on robot.

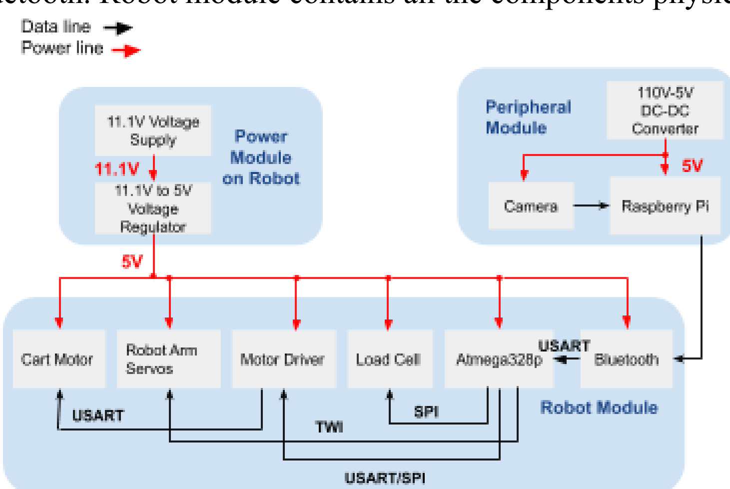

Figure 1. Block Diagram

2 Design

2.1 Mechanical Design

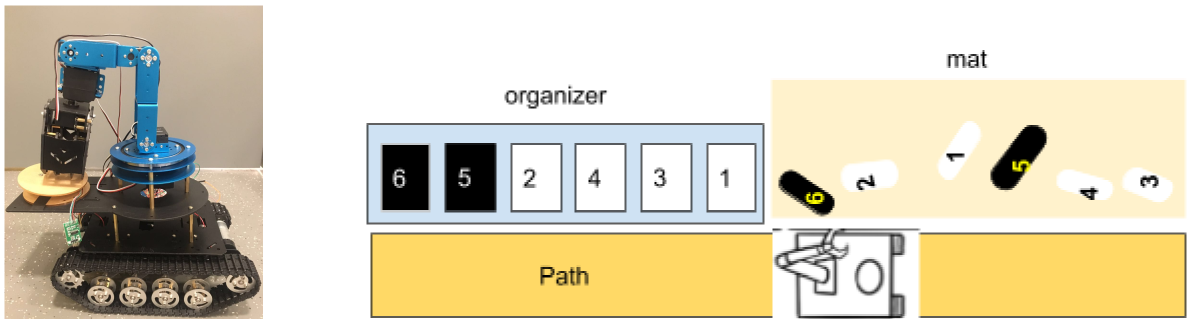

Figure 2. Mechanical Design

2.2 Design Procedures

Power, hardware, and software are three main clocks of our project. Based on the datasheet for all components of robot, due to high currents supply to motor and servos of robot arm, our group decide to use a 11V-to-5V switching regulator because of it high efficiency and low power assumption. The calculation of power assumption for our project is: $P_{12V\ to\ 7.5V\ switch\ regulator} = \frac{V_{OUT} \times I_{OUT}}{efficiency} = \frac{7.5\times4}{90\%} = 33.33W$. For hardware, we decide to use ATMega328p because it contains enough I/O pins required by our project. The I/O peripherals of our microcontroller include motor driver TB6612fng, bluetooth unit HC-05, load cell, and six servos of robot arm.

2.3 Design Details

2.3.1 Power Supply and Voltage Regulator Circuit

The power supply on our robot is a 11.1V rechargeable Lipo battery, with a discharging rate 35C and a capacitance of 1500mAh. Our robot arm needs around 3A to run the six servos, and our motor cart requires 2A when running, therefore it is necessary to have a large discharging rate battery along with 1500mAh to ensure that the power source can always provides enough current to power up servos and motors. The reason for choosing 11.1V instead of using higher input voltage source is that, for switch voltage regulator circuit, a smaller input voltage increases the efficiency, which is crucial to decrease the power consumption in the voltage regulation part. The battery is also rechargeable, reducing the cost of buying disposable battery every time the robot runs out of power.

We used to have three switch voltage regulators, which were 11.1V to 9V for the motors, 11.1V to 7.5V for the servos and 11.1V to 5V for the microcontroller, the load cell, bluetooth module and motor driver. After doing experiment, we find out that 5V is enough to power the servos and also the motors move in a moderate speed under this voltage, which improve the robustness of our project. Besides, considering the power dissipation, we finally agree to use only one switch voltage regulator, which is 11.1V to 5V. From our experiment results, the 5V voltage works pretty well with other components. The schematic of our switch voltage regulator circuit is shown in figure 3. The open capacitors in the figure are 47uF. Besides, the diode we include has a forward voltage of 0.55V, which is small and improves the efficiency of the switch voltage regulator circuit.

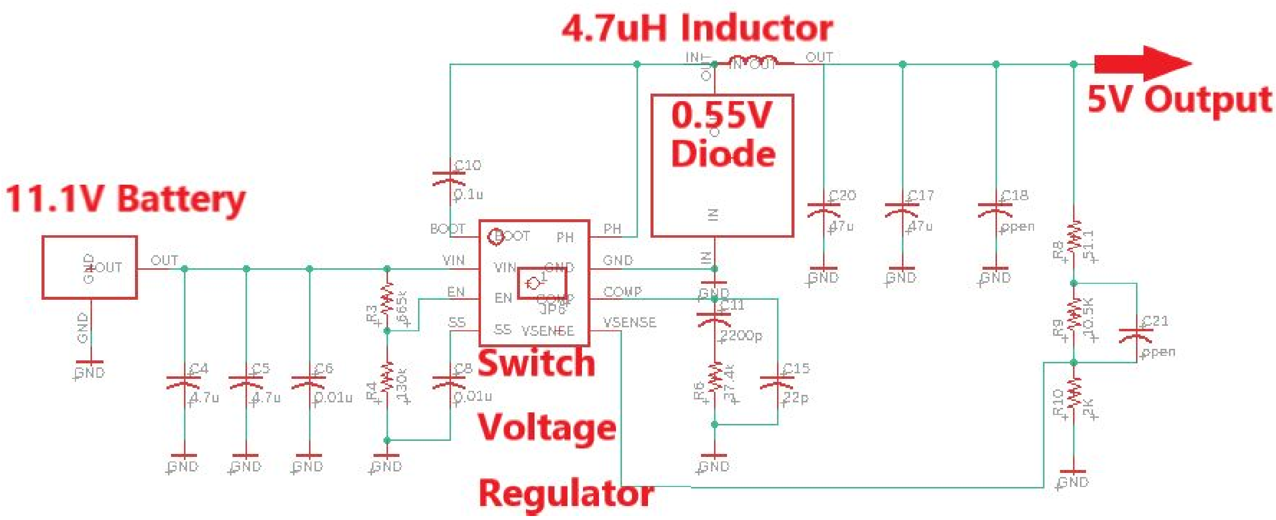

Figure 3. Power circuit schematic

2.3.2 Robot Arm

Our robot arm has six servos that control its movement as indicated in the figure 4. The servos have different models: servo 1 is LDX-335, servo 2 and 3 are LFD-06, servo 4 and 5 are s and servo 6 is LD-1501. All servos have three input: Vcc, gound, and PMW. These servos are connected to 6 analog pins on microcontroller. For servo1, it can rotate from 0 degree to 90 degree. For the rest of the servos, each can rotate from 0 degree to 180 degree.

Figure 4. Six-Servo Robot Arm

2.3.3 Microcontroller

ATMega328p microcontroller is the control unit of our robot. Bluetooth is connected to PD0/PD1, which are RXD,TXD 8-bit Universal Synchronous/Asynchronous Receiver/Transmitter(USART) bidirectional I/O port. Load cell is connected to 2 8-bit Serial Peripheral Interface(SPI) port. Motor driver is connected to 5 USART port and 2 SPI port [5]. And six servos are connected to 6 7-bit Inter-Integrated Circuit bidirectional serial bus port.

Microcontroller receives data transmitted from Raspberry Pi through bluetooth HC-05. Microcontroller reads serial input that contains the object number, center coordinates, slope and color of each shoe. Based on x-axis center coordinates passed in, the microcontroller sends command to motor and makes motor move right next to the target shoe. And then based on the y-axis center coordinates passed in, microcontroller starts calculating the angle required by six servos of robot arm to get to the target shoe. The setup of our robot and shoes is demonstrated in figure 5.

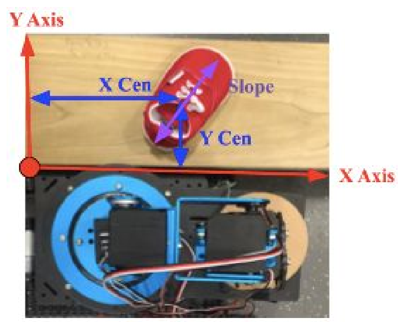

Figure 5. Setup of Robot and shoes

First of all we assign the value of y-axis center coordinates to d: $d = ycen$. The side view of our robot arm is in figure 6, and from that figure we can find the value of length beta by using Pythagorean theorem $beta = \sqrt{d^2 + (h - l_3)^3}$, angle beta based on law of cosine $\angle beta = cos^{-1}(\frac{-beta^2 + l_1^2 + l_2^2}{2l_1 l_2})$, and angle l2 $\angle l_2 = cos^{-1}(\frac{-l_2^2 + l_1^2 + beta}{2l_1 beta})$. The value of theta 4 is therefor $\theta_4 = 180° - \angle beta$. The value of angle_alpha is $\angle alpha = tan^{-1}(\frac{l_3 - h}{d})$. And then we can find the value of theta 5 to be 90- $\angle l_2$ - $\angle alpha$. In the end we can find angle for servo 5 to be $90° + \theta_5$, the angle for servo 4 to be $90° - \theta_4$. The value of theta 3 is then $\theta_3 = 180° - \theta_2 - \theta_1$, and value of servo 3 is $90° - \theta_3$.

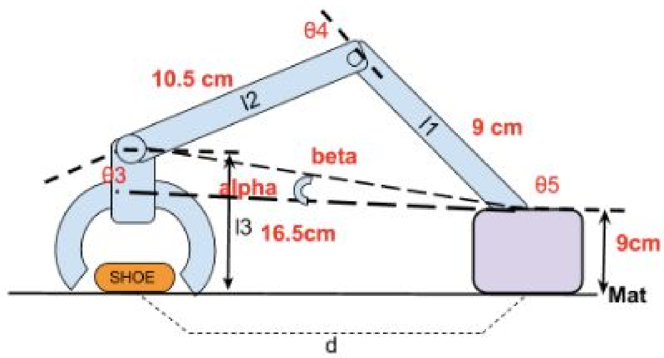

Figure 6. Robot Arm side view

The slope passed in from Raspberry Pi determines the rotation of servo 2 and there are two cases, which are shown in figure 7. If slope of the shoe is positive, servo 2 will rotate from $0°$ to $180° - tan^{-1}(slope)$, amd if slope is negative, servo 2 will rotate from $0°$ to $90° - tan^{-1}(slope)$.

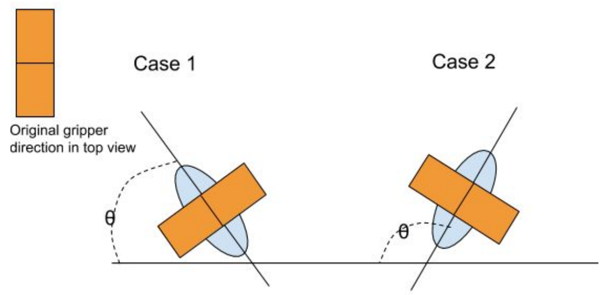

Figure 7. Two cases for servo 2

The output of calculation unit should be the rotation values of six servos that control the robot arm to get to the target shoe and grab it. And then microcontroller sends signal to robot arm to transfer the shoe back to the load cell located on robot platform. After several delay duration, microcontroller finishes recording the weight of current shoe, and then microcontroller sends signal to motor to move to the organizing area and then robot arm places the shoe down. Last but not least, microcontroller sends signal to motor to move backward to the starting point, and wait for another signal transmitted from Raspberry Pi.

When microcontroller tries to control motor, we want to fix one unit cycle of command to be around 1 cm of movement for motor. After a lot of experiments, we come up with a conclusion that by using the forward command with inputs of motor1, motor 2, speed of 100, following by delay duration of 0.2s, and a brake function following by another 0.2s, we can fix one unit cycle of motor movement to precise 1 cm.

When microcontroller tries to control six servos of robot arm, we also have down a lot of experiments to test the stability of all the servos. And the result we get is that only servo 2 has 0 offset. Servo 6 has the greatest offset of about 20 degrees. Therefore when we write command to control six servos, we also account for all the offsets. For example, if we want servo 6 to rotate from $0°$ to $90°$, we need to actually implement the rotation to be from $0°$ to $110°$ to account for the servo offset.

2.3.4 Bluetooth

We used bluetooth module HC-05 as shown in the figure.8 to do the data transmitting work. Since the Raspberry Pi has built-in bluetooth, we connected HC-05 with our microcontroller - the RXD and TXD pins on HC-05 was connected with TXD and RXD on ATmega. RXD and TXD are USART input/output pin which are 8-bit bidirectional I/O port. We left the key pin unwired since the key pin is used to change the operating mode but we only need the default mode - data mode in which it can send and receive data from other Bluetooth device, like Raspberry Pi. For VCC and GND pins, we powered this module by supply it the 5V output from voltage regulator. Then we paired with Raspberry Pi to help microcontroller receive the result file from the image processing program.

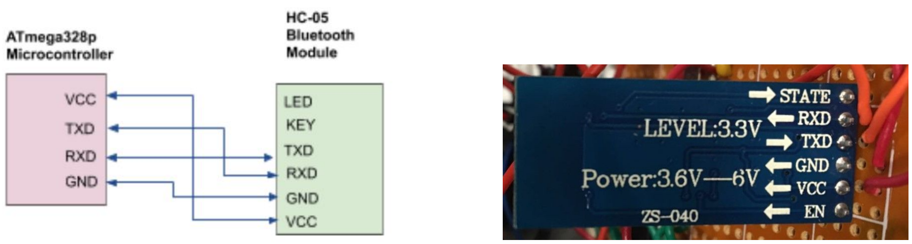

Figure 8. Bluetooth module

2.3.5 Cart Motor with Driver

The base cart of our robot has two DC motors and we use TB6612FNG driver IC to control those two motors. H-bridge is the main function of this chip and we used it to control the forward and backward movement. It also has a standby system which can save power and CW/CCW/short brake/stop function modes, those could help us in future improvement of letting the robot move in changing path. The circuit of motor driver control is shown in figure 9.

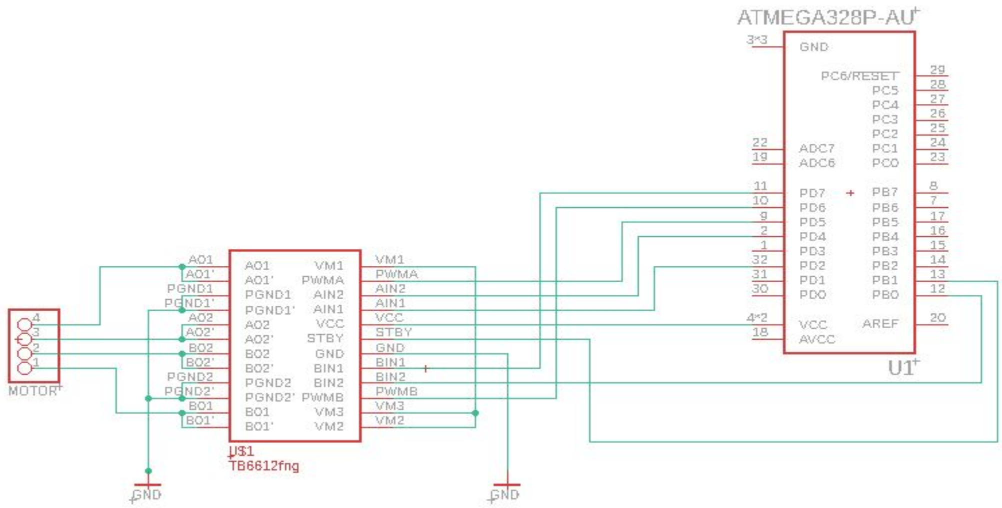

Figure 9. ATMega328p and Motor Driver Schematic

2.3.6 Load Cell

Our load cell contains a weight sensor and a HX711 analog-to-digital converter module. We place our load cell on the robot platform, so every time after the robot grab the shoe it will transfer the shoe to the load cell, which will then send digital value of the shoe's weight to microcontroller. We set a delay duration of 8 seconds for load cell to detect the weight of the shoe for four rounds, and then microcontroller will store the average value. Figure 10 below is the connection between load cell and ATMega328p.

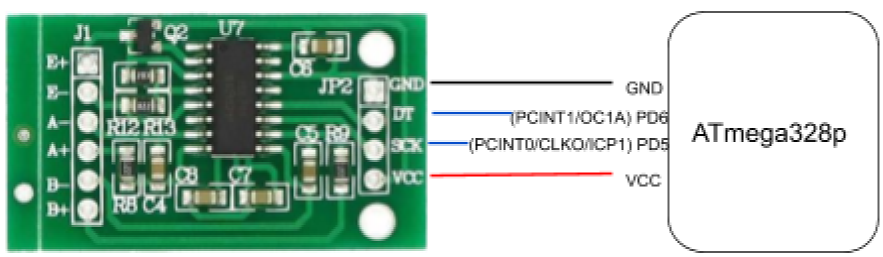

Figure 10. Bluetooth and ATMega328p connection

2.3.7 Raspberry Pi with Camera

We selected Raspberry Pi and Pi camera as the image processing unit. As shown in the figure 11, the Pi Camera was connected by a 15-pin ribbon cable with Pi by its CSI-2 port and directly powered by the Pi[1]. The camera was used to take a still picture before each round of work, the picture was saved as an 480*640 pixels PNG file and was readed as the input of image processing program. [3]

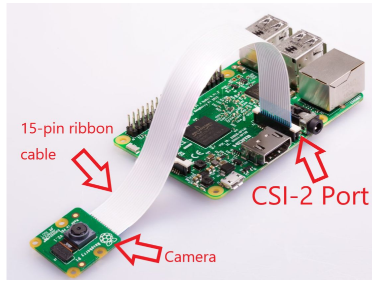

Figure 11. Raspberry Pi with Camera

We find that the x-axis coordinate is very accurate, and the offset between image captured by camera and reality is trivial. However, there is an offset of -3 cm on y axis between the image captured by camera and real life setting, therefore when we calculate the y-axis coordinates of all the shoes, we need to take this -3 offset into account.

2.3.8 Image Processing

The input of our program is the 480*640 PNG files figure 13 and the output of the program is in the format shown in the figure 12. We output 4 types of element for each shoe. The object number is used to decide the shoe picking-up order which means the robot always pick up the object 1 first. The coordinate correspond to the center point coordinate of each shoe, the way we find the center coordinate is to find the tail and toe coordinates first then calculate the midpoint coordinate. The slope is also calculated by the tail and toe coordinates and this value help to decide the rotation angle of gripper(servo 2). For the convenience of data transmitting, we used number to represent colors, 0 represents red, 1 represents yellow and 2 represents black. The way we recognized colors is to check if the HSV value of pixels matched the range we set

object	coordinate	slope	color
1	(514, 153)	8.69	0
2	(362, 170)	1.36	0
3	(434, 74)	-0.75	1
4	(121, 99)	-2.28	1
5	(267, 66)	200	2
6	(227, 184)	1.12	2

Figure 12. Result txt file generated in Raspberry Pi

Figure 13. Input Image

To have a more accurate control, we took the offset between image and real distance into account. Since the output coordinates are in pixel unit whereas the robot need command of real life distance in unit ‘cm’, we need to find the ratio between image and real setting.

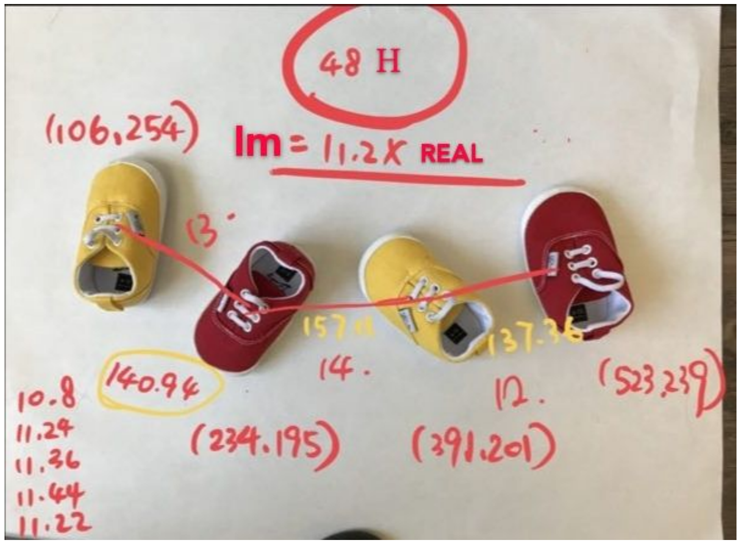

Figure.14 Ratio of Image

Figure.14 is an example of ratio calculation, the coordinates(106,254) and coordinate(234,195) are the center coordinates of the yellow shoe and red shoe on the left. Those coordinates are in pixel units which are the result from image processing program, therefore we calculated the distance by using the Pythagorean theorem and get the value 140.94. In addition we measured the real distance by ruler as 13cm so we find the ratio is 10.8 since 140.94/13=10.8. Then we did the same calculation on the other shoes and got 5 values of ratio, by calculating the average of those 5 values we decided to use the ratio 11.2 to transfer image distance to real life distance.

3. Design Verification

We unit tested off-the-shelf components and verified that they met our operating requirements. And the following section outlines the verification results for all the blocks of our design.

3.1 Power Module

The simulation for the 11.1V to 5V voltage regulator circuit is shown in the following figure 15.

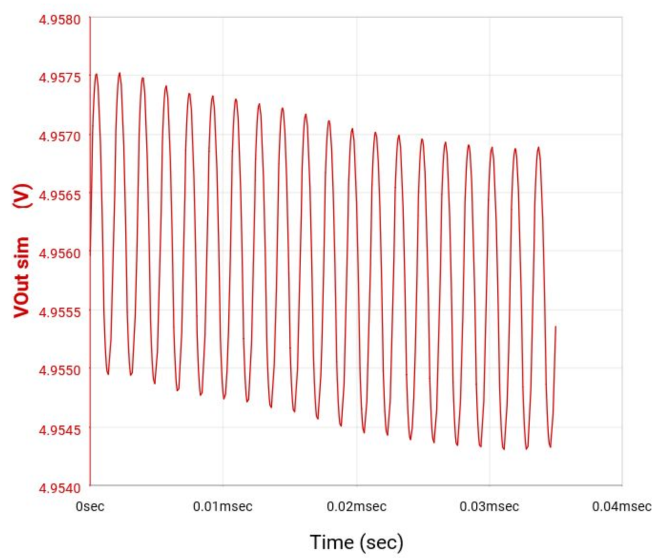

Figure 15. Output voltage of voltage regulator circuit

The following figure 16 indicates the output voltage versus time, with an input voltage of around 11.1V. The results from the simulation matches our expectations.

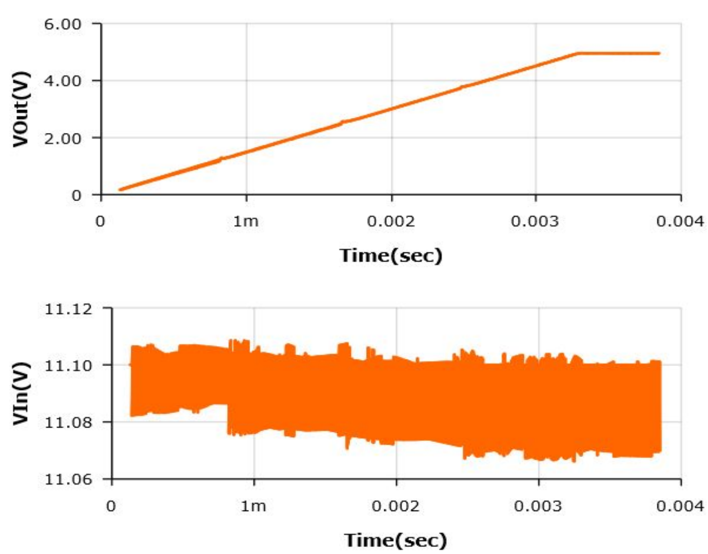

Figure 16. Input voltage and Output voltage versus time

3.2 Mechanical Design

Robot arm and cart motor are two major parts in our mechanical design. First we modularly tested the stability of these two. During rotation, offsets exist between ideal situation and reality for six servos. For servo one, there is a $10°$ offset; for servo 2, the offset is $0°$; for servo 3, 4 and 5, offset is about $10°$; and for servo 5, offset is about $20°$. After we came up with the values of these offsets, we took them into account when we write command to control the rotation of robot arm.

We achieve the stability of motor movement by trying forward command and brake command with different input speeds and delay durations. And the best result we get is to use a forward command with input speed of 100 and delay duration of 0.2s, following by a brake command with delay duration of 0.2s. In this way for every unit cycle of command we can precisely control the motor movement to be 1cm. In figure 17, we did four test cases to check whether our motor behaves correctly by measuring the physical distance traveled by the motor with rulers, comparing with the ideal distance that the motor should travel. The results show that the accuracy of the motor movement is about 75%.

Test Case	Ideal Distance	Measured Distance
1	13cm	11cm
2	5cm	5cm
3	17cm	17cm
4	42cm	42cm

Figure 17. Motor Test

After assembly these two major blocks together we verified that when taking all the offsets from ideal situation into account, our cart motor can always move right next to the shoe, and our robot arm can get to the target shoe and grab it successfully.

3.3 Control Module

We tested that our control module was operational under a 5V power supply and the robot was attempting to move the DC motors at first. We know that the circuit of ATMega328p and motor driver is working correctly once the movement of motor matches with our command. For example, figure 1 indicates the parameters for two shoes and the first shoe needs to travel 91 pixel distance, which is 8cm in real distance. We then measured the distance that motor travels to the first shoe, which is exactly 8cm, indicating that our control module is successful. Furthermore, we also assembled the six servos of robot arm to our control circuit and verified that each servo can rotate correct angles based on commands from microcontroller.

```
============================== RESTART ========================
>>> %Run wholeprocess.py
 object    coordinate      slope     color
 1          (91, 91)      1.31       0
 2          (335, 95)      200       1
```

Figure 18. Result in Raspberry Pi

3.4 Bluetooth Unit

We verified the functionality of our bluetooth module HC-05 by sending input from Raspberry Pi manually and print the serial input received by microcontroller. We know bluetooth is working correctly by checking that all values are matched according to figure 19. The letter c in figure 18 refers to the object number.

object	coordinate	slope	color
1	(514, 153)	8.69	0
2	(362, 170)	1.36	0

Result.txt in Raspberry Pi

```
c is:
1
x is:
514
y is:
153
slope is:
8.69
color is:
0
c is:
2
x is:
362
y is:
170
slope is:
1.36
color is:
0
```

Data received in Serial Port

Figure 19. Bluetooth Verification

3.5 Load Cell

We tested our load cell was operational by printing the weight of shoe detected by load cell. And then we manually weigh the shoe on scales. The value we get from load cell and value we get from scales is almost the same. Figure 20 shows the measurements for three shoes: Red1, Red2 and Yellow 1. And then we tested the weight of one pair of shoes detected by load cell and find the offset is within 5%.

Weighing Times	Experiment	Real
1	19.1g	18.7g
2	18.9g	18.7g
3	18.5g	18.7g
4	18.6g	18.7g

Average Weight for Red1: 18.775g

Weighing Times	Experiment	Real
1	18.7g	18.7g
2	18.1g	18.7g
3	18.8g	18.7g
4	18.1g	18.7g

Average Weight for Red2: 18.425g

Weighing Times	Experiment	Real
1	20.6g	20.5g
2	20.3g	20.5g
3	20.5g	20.5g
4	20.9g	20.5g

Average Weight for Yellow1: 20.575g

Figure 20. Load Cell Measurements

3.6 Image Processing

All of the major requirements of the Image Processing Module were met. The execution time of object detection was at most 10 seconds respectively, for all images tested.

4. Costs

4.1 Parts and Components Cost

Description	Quantity	Unit Price	Cost
CSTCE16M0V53-R0 16MHz Ceramic Oscillator	1	0.5	0.5
100uF Electrolytic Capacitor	2	0.35	0.7
ATmega328-P	1	2.01	2.01
Raspberry Pi 3 Model B	1	35.8	35.8
Energizer A23 Battery	4	9.71	38.84
Raspberry Pi Camera Module V2	1	29.95	29.95
Raspberry Pi 3b Power Supply	1	10.99	10.99
5V Voltage regulator	1	1.59	1.59
Robot arm+shipping	1	163.35	163.35
Gripper	3	22.5	67.6
Digital load cell	1	11.29	11.29
Robot chassis	1	98.99	98.99
SMD ¼ watt 1.8 Kohms 5%	2	0.1	0.2
Capacitors MLCC - SMD/SMT 22pF 50V NPO 10%	2	0.13	0.26
TB6612FNG Motor Driver	2	2.11	4.22
Fixed Inductors Fixed Inductors 4.7uH	6	0.91	5.46
Diodes & Rectifiers SCHOTTKY DIODE 5A, 40V	6	0.7	4.2
Capacitors MLCC - SMD/SMT 22pF 3K V NPO 5%	6	2.33	13.98
Capacitors MLCC - SMD/SMT 2200pF 16V 10% 0805	6	0.13	0.78
Capacitors MLCC - SMD/SMT X6S 4Vdc 47uF	24	0.491	11.78
Capacitors MLCC - SMD/SMT 0.1uF 10% 50V	6	0.67	4.02

Capacitors MLCC - SMD/SMT 4.7uF 10% 25V	12	0.34	4.08
Capacitors MLCC - SMD/SMT 0.01uF 10% 50V	12	0.243	2.92
Sockets & Adapters Sockets & Adapters SO Prototyp Adaptor 8 contact SOIC	5	3.09	15.45
SMD 665K OHM 1%	6	0.23	1.38
SMD 1/4watt 130 kohms 1%	6	0.1	0.6
SMD 1K ohm 0.1% 15 ppm Thin Film	2	0.68	1.36
PCB	5	8.02	41
Total			573.3

4.2 Labor Cost

We have chosen an hourly wage from a salary info sheet provided by engineering at Illinois for Electrical Engineering graduate[4].

Team Member	Hourly Rate	Hours	Cost ✕ 2.5
Jinghan Guo	39 $/hr	180 hrs	17550
Quanhua Huang	39 $/hr	180 hrs	17550
Mingxi Zou	39 $/hr	180 hrs	17550
Total			52650

4.3 Grand Total

Grand Total = Parts and Components Cost + Labor Cost = 573.3 + 52650 = 53223.3 dollars

5. Conclusion

5.1 Accomplishments

Our project can successfully implement a whole process of locating the shoes, moving robot to the specific position, picking up shoes and putting them in the desired places. The image processing parts also captures the images clearly and produces the correct outputs that are sent to ATMega328p through bluetooth.

5.2 Uncertainties

There are a lot of mechanical limits exist in our project. First of all, the length of the robot arm is 36cm, which is relatively short and the size of our end effector is not big enough to grab real size shoes. Also I want to point out that for six servos, serve 2 to 6 has rotation range from $0°$ to $180°$ while servo 1 has rotation range from $0°$ to $90°$. After a lot of experiments, we came up with a conclusion our robot arm can only reach objects that lies between 11 cm to 17 cm away from the base of the robot.

5.2 Ethics

After reading the IEEE Code of Ethics[7] and the ACM Code of Ethics and Professional Conduct[8], I think our project is compliance with all the Ethics requirements mentioned above. In our design, the shoes sorting robot is small and does not take much space. The maximum height of the robot is about 55.5cm but usually it only takes around 30cm. In addition, the robot will only pick shoes and will not harm people. The speed of the car chassis is slow and the voltage that needed for this robot is only 5 Volts, which is acceptable for human body and will not hurt others. The IEEE Code of Ethics #9, "to avoid injuring others, their property, reputation, or employment by false or malicious action", also support this idea. In addition, we will try our best to reduce any possible risk when using this robot, as according to IEEE Code of Ethics #1, "to hold paramount the safety, health, and welfare of the public, to strive to comply with ethical design and sustainable development practices, and to disclose promptly factors that might endanger the public or the environment"[7].

5.3 Future work

5.3.1 Feedback loop

To further improve our project, we should implement feedback loop to always keep track of robot and shoes in case the robot deviates from working path or any shoe falls off during transferring. which is designed to track the position of robot cart and position of shoe in transition. By adding the feedback loop,

5.3.2 Improvement of image processing

We can improve the image processing program by adding functions to discriminate shoes and other object on the shoe mat and we should try to output a more accurate coordinates of each objects by eliminating the offset due to the position of camera.

5.3.3 Variability of working path

We only have a straight single working path right now, but we should enable the robot to move in more directions to cover a large working area.

References

[1]Raspberry Pi Datasheet
https://static.raspberrypi.org/files/product-briefs/Raspberry-Pi-Model-Bplus-Product-Brief.pdf [Accessed 5-March-2019]

[2]12V Rechargeable Battery
"12-Volt Prong-Top Rechargeable Battery." *Our History : Cabela's*, www.cabelas.com/product/Volt-Prong-Top-Rechargeable-Battery/741254.uts?productVariantId =1400479&WT.tsrc=PPC&WT.mc_id=GoogleProductAds&WT.z_mc_id1=02761176&rid=20 &ds_rl=1252079&gclid=EAIaIQobChMIyYvm99Dn4AIVlYxpCh3DQwzGEAQYAyABEgKdc vD_BwE&gclsrc=aw.ds.

[3]Raspberry Pi Camera Module v2 Datesheet
https://cdn.sparkfun.com/datasheets/Dev/RaspberryPi/RPiCamMod2.pdf [Accessed 5-March-2019]

[4]Salary.Info.Sheet.pdf
https://engineering.illinois.edu/documents/Salary.Info.Sheet.pdf [Accessed 5-March-2019]

[5]Arduino
Arduino - Introduction, www.arduino.cc/en/Tutorial/ArduinoToBreadboard.

[6]5V voltage regulator
STMicroelectronics, "Positive voltage regulator ICs", L78 datasheet, Revised Sep.2018
https://www.mouser.com/datasheet/2/389/l78-974043.pdf

[7] "IEEE Code of Ethics", Ieee.org. (2019).[online]. Available:
https://www.ieee.org/about/corporate/governance/p7-8.html [Accessed 5 Feb. 2019].

[] Acm.org. (2019). The Code affirms an obligation of computing professionals to use their skills for the benefit of society. [online] Available:
https://www.acm.org/code-of-ethics [Accessed 5 Feb. 2019].