THE SHOES SORTING ROBOT

By

Jinghan Guo

Quanhua Huang

Mingxi Zou

Design Document for ECE 445, Senior Design, Spring 2019

TA: Zhen Qin

02 February 2019

Project No. 12

Contents

1 Introduction

1.1 Objective

Nowadays robots are becoming more and more important in our daily lives. During hurricane Katrina, robots like isensys UAV and precisionhawk can collect data needed by rescue team in 7 hours that usually take people 3 days to get any other ways. During the Japan Tsunami, by using the sonar of Sarbot Interface, engineers were able to reopen a fishing port in 4 hours, when the fishing port was told that it was going to be six months before they could get a manual team of divers in. During these six months fisherman in Japan will miss the fishing season, which is the major economy for them. There are many more examples like these two that shows the improvement robots bring to people's life. Robots go places people cannot go and do things that people don't want to spend time doing, robots even do things new by assist us in innovative ways.

Our goal is to bring a house-use robot that helps people organize shoes. Our group comes up with this idea because it is really annoying when people get tripped over a shoe in the doorway. Disorganized shoes can be a mess but sometimes people just don't get time to organize shoes after they take them off. A clear entrance in the doorway can not only make it easy for people to go through but it can make house look nice and clean in the first sight. Organizing shoes is the principal goal of our senior design, but after we successfully build the robot, this application can be extended into organizing all kinds of things like dishes, toys, and clothes.

Therefore, we aim to build a shoe sorting robot to free people from spending time organizing shoes. We will build a mobile robot with chasis and arm, the robot will load a camera for vision and a load cell for sensing the weight. Our robot will initially awaiting next to the shelf, after people take shoes off, robot will come to the shoe, pick it up and place one shoe next to the other on the shelf.

1.2 Background

Our robot is built based on the scenario that shoes are scattered on a 30*100cm entrance mat with a rectangular shoe organizer next to it. A camera will be held above a certain height to capture the whole mat. Our robot is constituted of a 6 degrees of freedom end effector and a mobile chassis. It sits quietly next to the organizer when no shoe appears on the mat, after the camera detects any object, the car moves to the mat and start working.

The robot moves to the shoe that is closest and distinguish each shoe by color filtering and weight filtering. The image captured by camera records the size and color of the object. If that object size lies in a certain range we assign it as a "shoe". Otherwise we just ignore it if the size of the object appears to be too large or too small. And the load cell on the robot records the weight of every object. In this way we can decide if two things are a pair depends on their color, size and weight.

So far there is no such robot on the market, therefore this project will be novel. The hit of mobile vacuum cleaner in the market is an inspiration for us and we believe our shoe sorting robot will be popular if we make it affordable and reliable.

1.3 High-Level Requirements Lists

- The camera must be able to detect correct number of shoes on the 30cm*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.

2 Design

2.1 Block Diagram

Figure 1 shows that our overall design contains three modules: the robot module, the power module and the peripheral module.

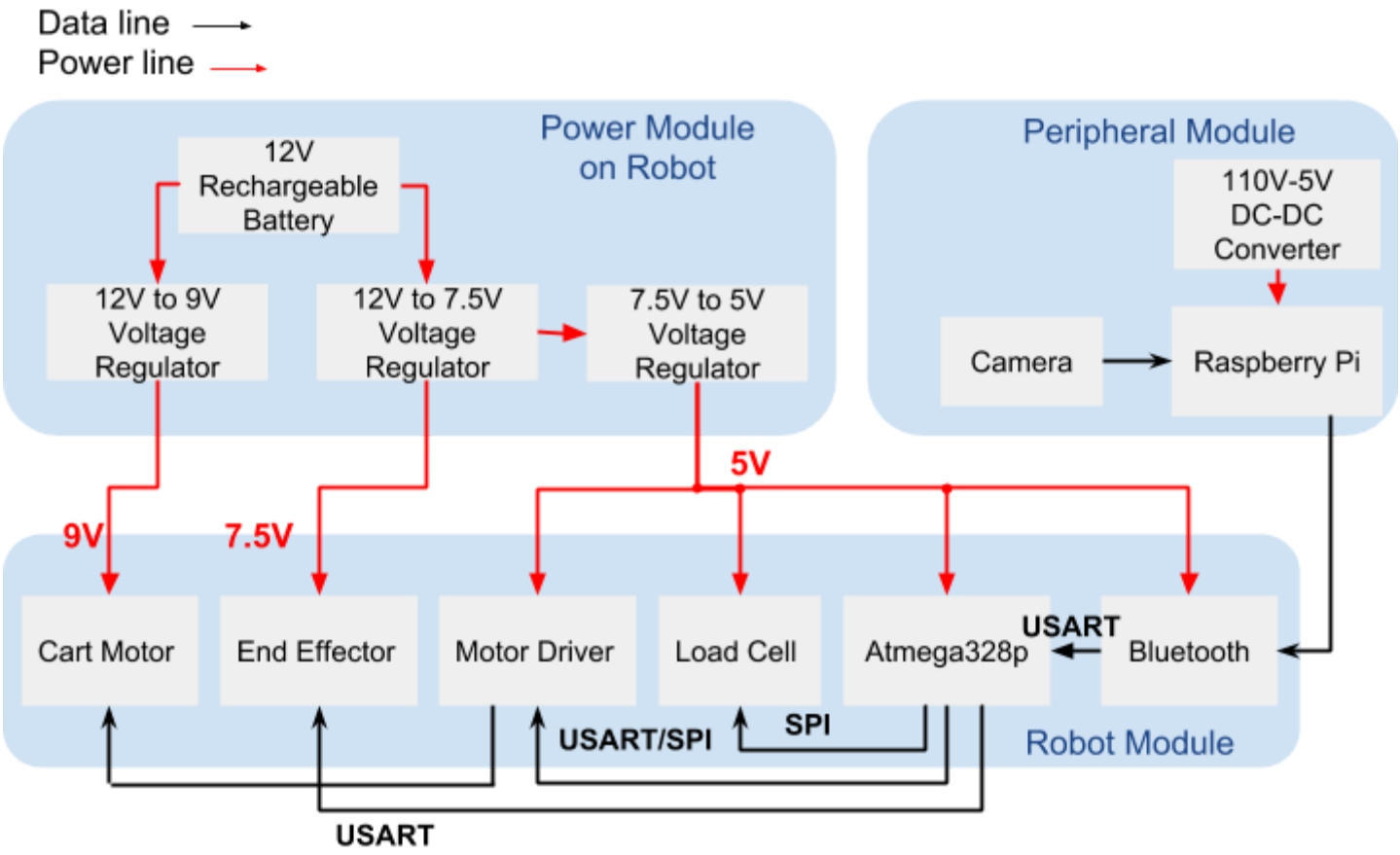


Figure 1. Block Diagram

2.2 Physical Design

The physical design of our project includes a moving cart with a robotic arm, load sensor, microcontroller and motor driver on it, as indicated in figure 2. The picture 2 below shows the circumstance when the robot gets the signal of working. The light orange rectangle on the top right is the shoe mat(30cm*100cm) with three pairs of shoes with different colors. The rectangle on the left represents the shoe organizer and robot sits on the side will move along the path by the mat to reach the closest shoe when start working. We will assign four properties to each shoe: **color, object number, weight,** and **check condition**. The object number indicate the number of each shoe and check condition indicates if the shoe is paired or not. In our experiment setting, we are going to have three pairs of shoes, 2 pairs of white, 1 pair of black shoes. The color will be recorded based on its RGB value, thus the color and object number will be assigned before the robot starts working. Each time the robot picks up a shoes, it will record the weight and mark the shoe as PAIRED or NOT PAIRED. The robot will organize shoes based on their color and pairs of same color will be placed next to each other on the organizer.

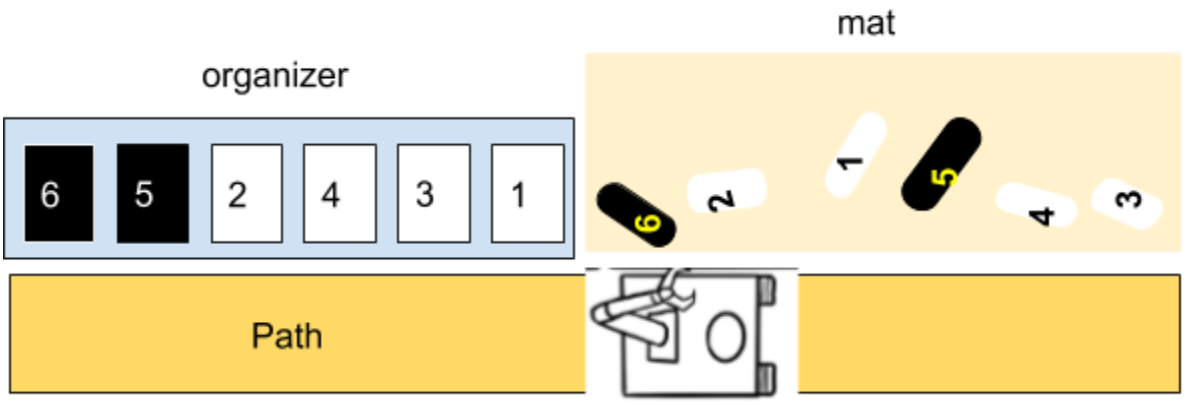


Figure 2. Physical setting

Figure 3 is the physical form of our robotic arm and it shows that the dimension of the arm is 465*120*120mm. The chassis of our robot has a dimension of 270*255*90mm(L*W*H).

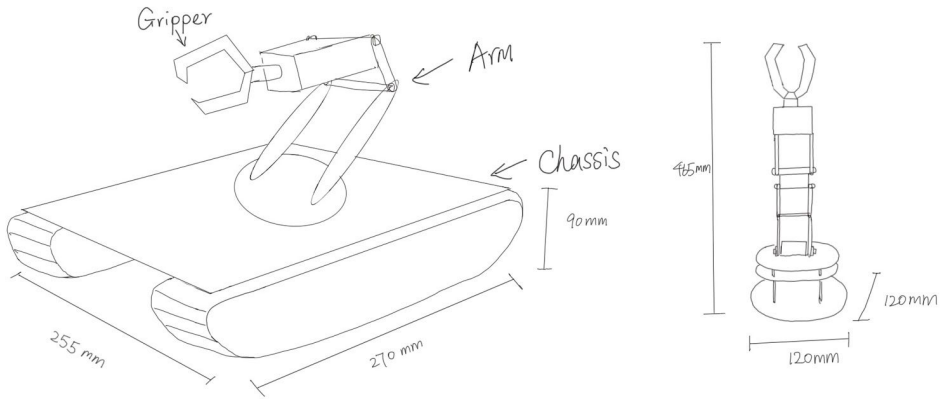


Figure 3. Physical Scale

Figure 4 demonstrates the physical structure of our robot in the side view.

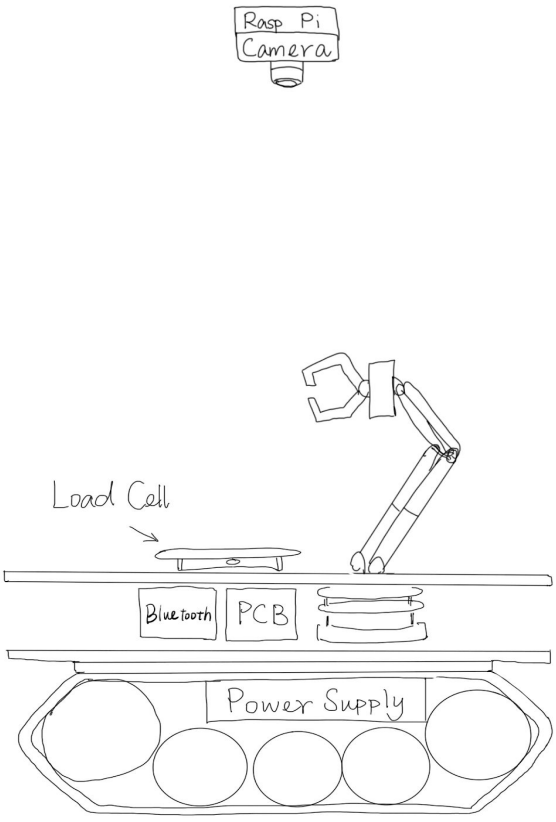


Figure 4. Physical Structure (side view)

2.3 Robot Module

2.3.1 Control Module

The control module will control all of the logic of the system from capturing the image, generating the path, accepting the instructions on the PCB, and carrying out the DC motor and servo motor instructions.

2.3.1.1 Raspberry Pi 3 b and Camera Module v 2

The reason that we use Raspberry Pi 3 b[4] instead of microcontroller (such as Arduino Uno) to do process image is for the following reasons[12]:
(1) Raspberry Pi 3 b has a GPU and a camera port, which is essential for receiving stream frame lines from the camera module.
(2) Raspberry Pi 3 b has much greater processing speed, 64-Bit 1.2 GHz compared to 16 MHz of Arduino
(3) Raspberry Pi 3 b is a Single Board Computer that is more powerful than microcontroller
(4) Raspberry Pi 3 b can connect to bluetooth device with its built-in bluetooth module while Arduino cannot achieve that without extra modules[13].
The Raspberry Pi is capable of processing 1G pixels per second with filtering and DMA in the GPU. The memory used inside the Raspberry Pi is 1GB LPDDR2 RAM, which increases the battery life. To power the Raspberry Pi, we will use a 110V to 5V DC-DC Converter.

Since the processing speed of Raspberry Pi 3 b is 1.25 Ghz, we choose the Raspberry Pi Camera Module V2 which has a smaller processing speed so they can work fine without timer messed up[6]. The camera will be connected to Pi through the Camera Serial Interface 2-Port as in figure 5. There are 4 data lanes that transmits streaming frame lines from camera to Pi via ribbon cable. The image buffer in RAM assemble all the frame lines and CPU of Pi can get access to the image through Direct Memory Access[11]. Then we will achieve all the image processing in Raspbian. After receiving the processed image from Pi through the HC-05 bluetooth module, ATMega328p will then give instructions to motors and servos so our robot can perform grabbing shoes autonomously.

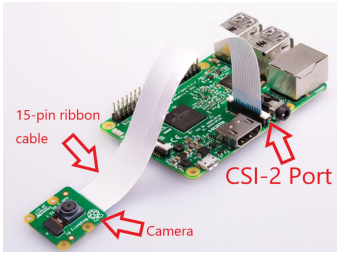


Figure 5. Raspberry Pi and Camera

The Raspberry Pi Camera V2 has a dimension of 23.86 x 25 x 9mm. We will place the camera 1m above the top left corner of the mat so it can take image of the whole mat upon instruction and then send the image back to the Raspberry Pi. After implementing image processing in the Raspberry Pi, all the objects will be distinguished from the background. The still picture resolution of this camera is 8 Megapixels (sensor resolution 3280 x 2464), which is suffice to record the color and size of the objects on the mat. The maximum frame rate is 60fps.[6]

The image captured by the camera is read rows by rows from the sensor and the data of each rows are stored inside the camera registers. The frame lines are then streamed through the 15-pin ribbon cable to the Camera Serial Interface-2 port on the VideoCore IV GPU. The CSI-2 has 4 data lanes, each of which has a max of 1 Gbps bandwidth. These frame lines are stored in the image buffer and is combined to a complete frame image after post-processing. In this way, the CPU can get access to the image frame through Direct Memory Access (DMA) and VideoCore Host Interface (VCHI), as illustrated in figure 6.

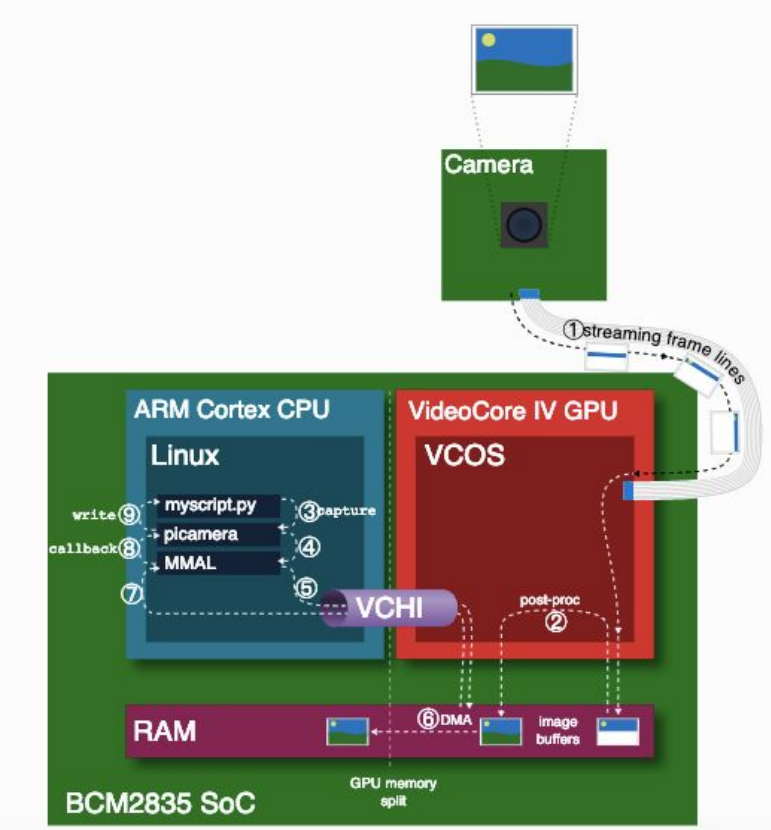


Figure 6. Raspberry Pi and Camera Module

Requirement	Verification
1. The Raspberry Pi must have at least 1GB memory storage to store the image captured by camera. 2. The Raspberry Pi must have processing speed greater than 1G Hz to implement image processing.	1. We will verify the RAM size by by running the command on Raspbian[16] 2. We will verify the processing speed by running the command to get the current frequency of CPU[15]

2.3.1.2 Bluetooth Module

Raspberry pi has a built-in bluetooth 128-bit key 4.1 module with max range of 50 m and max transfer speed of 723 kb/s. In order to avoid inconvenient cable transmission, we will use bluetooth to transfer data from Raspberry pi to microcontroller. Thus ATmega328p needs to connect with a serial bluetooth module HC-05 which has operating voltage 4V to 6V and operating current at 30mA. As shown in the figure 7 below, the RXD and TXD pins on HC-05 will connected with those on ATmega. RXD and TXD are USART input/output pin which are 8-bit bidirectional I/O port. We will 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.

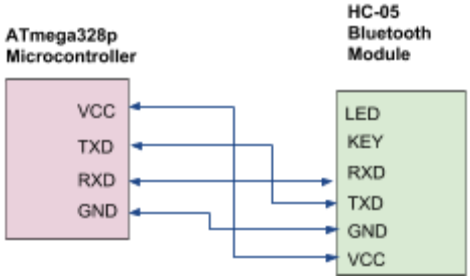


Figure 7. Bluetooth Connection

Requirement	Verification
1. The bluetooth module must be able to receive data in the range of 2.5m which is the max distance between robot and raspberry pi.	1. We will setup the connection between the bluetooth module and the bluetooth on raspberry pi, and then adjust the distance between to test if the data is being transmitted and received.

2.3.1.3 ATMega328p Microcontroller

Microcontroller receives data from the Raspberry Pi, and then converts the command to motor rotations and sends information to activate robot arm and end effector. The supported instructions must include start/stop motors, raise/lower robot arm, change the angle of 6 servos, and open/close end effector.

The interface between microcontroller and Raspberry Pi is handled by HC-05 bluetooth module. The load cell will interface with microcontroller through amplifier HX711. To drive the DC motors, we use the TB6612FNG motor controller.

The ATMega328p microcontroller will be programmed by Arduino Nano, which uses arduino board as an ISP. The arduino board is connected to the computer through USB first, and we will upload ArduinoISP to our arduino nano. Before we could upload our own program to ATMega328p, we need to burn the bootloader first.

Next, we will connect Arduino nano and our microcontroller ATMega328p directly, as illustrated in the figure 8. The 8-bit UART RX and TX digital communication ports are wired with RXD (PD0) and TXD (PD1) respectively. The other pins such as RESET, 5V and GND in arduino are connected with the corresponding pins in ATMega328p.

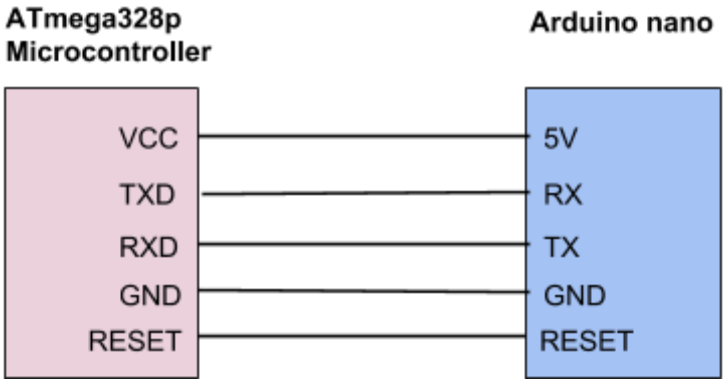


Figure 8. Microcontroller a and Arduino Nano

Figure 9 shows the schematic of our microcontroller, robot arm, Bluetooth, motor drivers and motors. We only have one microcontroller, but the one illustrated on the top right is just a duplicate microcontroller that is for wiring simplicity.

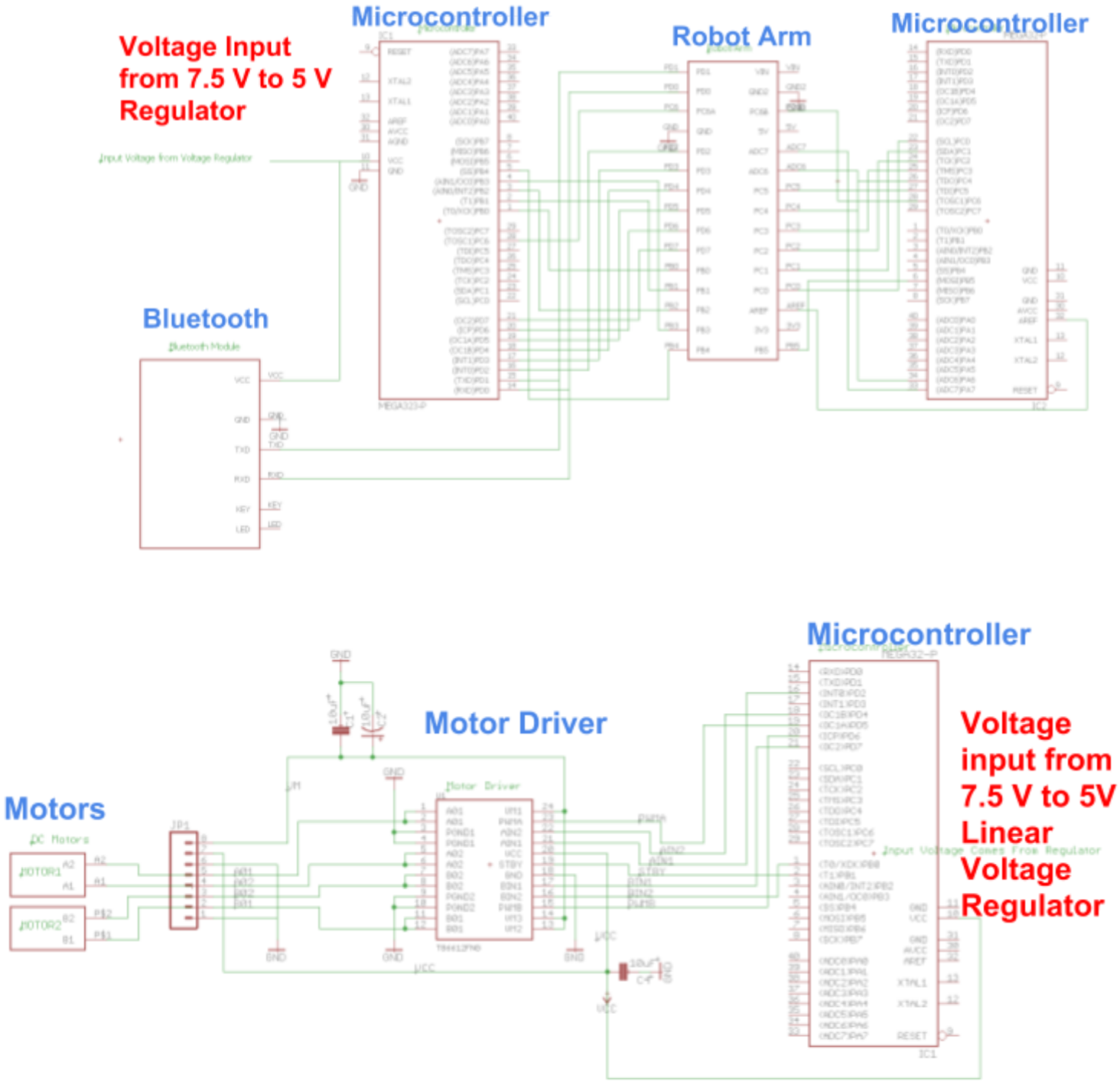


Figure 9. Schematic

The ATMega328p microcontroller also implements the shoe sorting algorithm in figure 10.

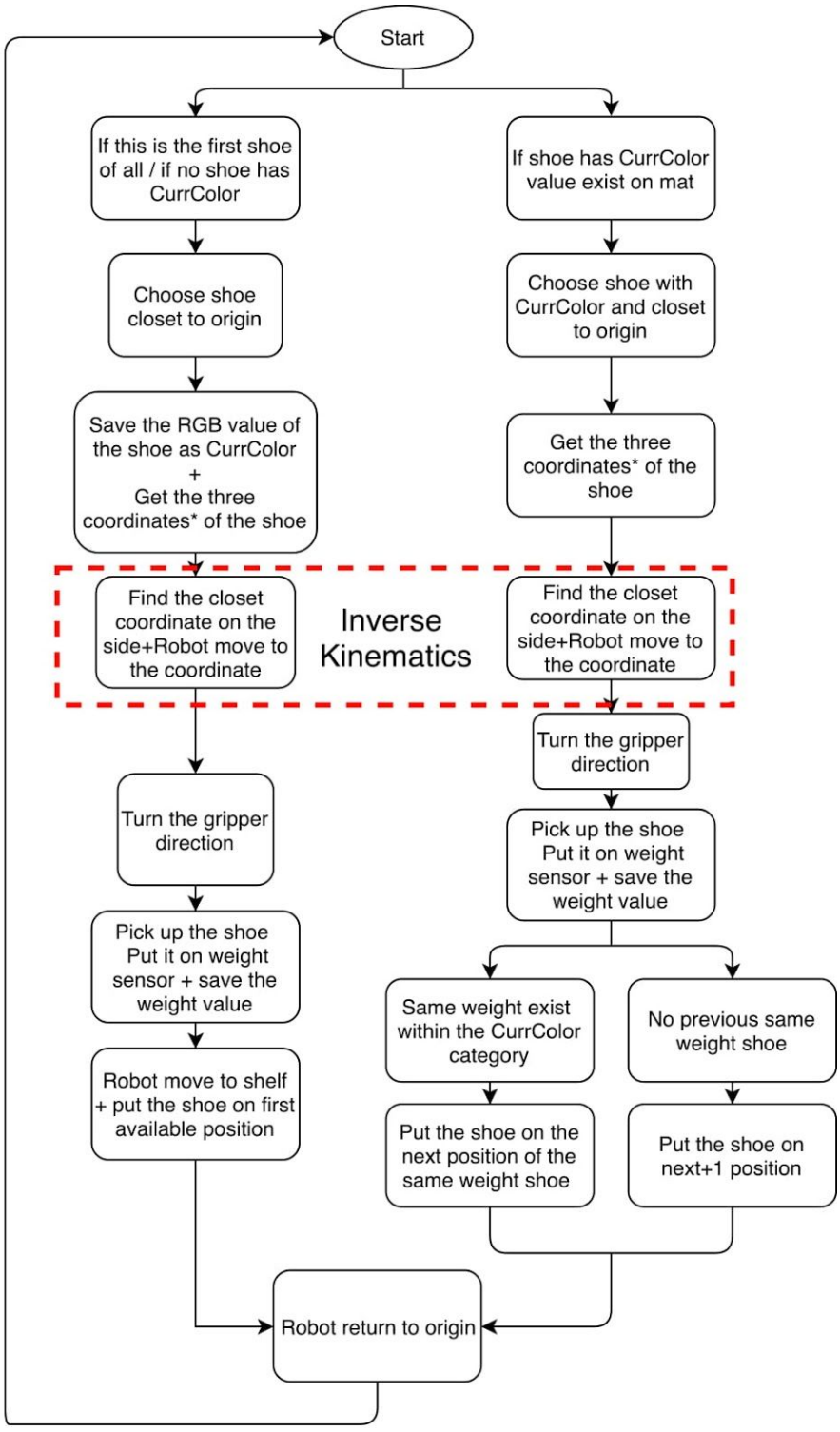


Figure 10. Shoe Sorting Algorithm

Requirement	Verification
1. The microcontroller must have at least 17 GPIOs to interface with robot arm, cart motor, bluetooth module and load cell 2. The microcontroller must have Clock Frequency of at least 1000 kHz to interface with robot arm, cart motor, bluetooth module and load cell 3. The microcontroller must have around 20 kB memory size to store instructions we will use to control robot arm and cart motor	1. ATMega328p has 23 GPIOs to 3 8-bits GPIO ports [17]. 2. ATMega328p has an internal clock of 1 MHz, and we can always add a 16 MHz crystal oscillator to ATMega328p in case that greater clock frequency is needed. 3. ATMega328p has 32 kB program memory size which is more than our requirement.

2.3.2 Robot Module

Robot Module receives demands and data from the microcontroller and then controls the robot movement through two motors and controls six servos of robot arm to reach the shoes.

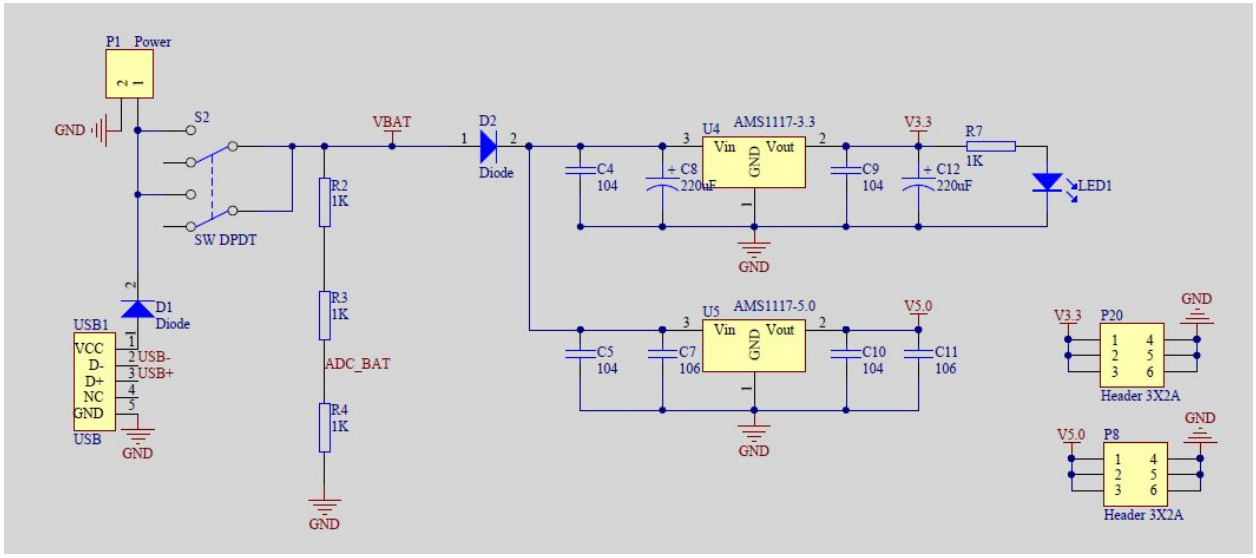
2.3.2.1 Robot Arm

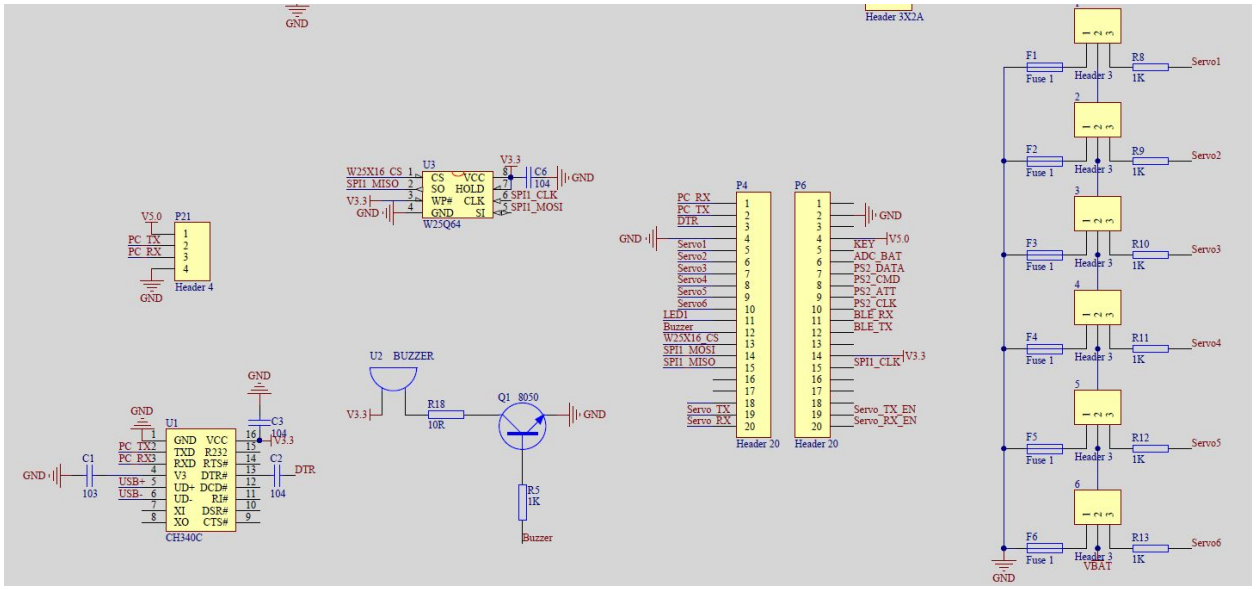
Our robot arm has six servos that control its movement as indicated in the figure 11. The servos have different models: servo 1 is LDX-335, servo 2 and 3 are LFD-06, servo 4 and 5 are LDX-2188 and servo 6 is LD-1501. Among all the servos, servo 1, 3 and 4 receive PWM signals. We place the end effector in the image below with a bigger one(stretch up to 14 cm) so that it is big enough to pick shoes up.

Figure 11. 6 Servos of Robot Arm

Our group will implement the circuit below to control the six servos of robot arm. There is a demo board that comes with the robot arm we purchased, but we will not use it. Instead we will figure out what components there are on the demo board we need in order to control servos and integrate them with our PCB.

The plan for our group is that we will have one person building the circuit below and the other person building the rest of PCB at the same time. In this way, we can use the demo board that comes with the robot arm to test if the rest of PCB is functioning properly and then insert our own servo control to see if it works.





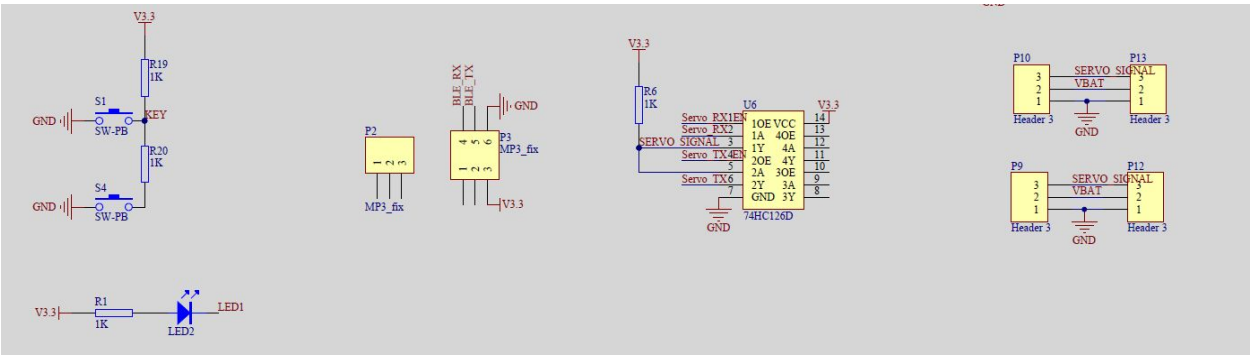


Figure 12. Schematic for Servo Control

Pin layout for 6 servos and corresponding pin on ATMega328p. The connections between the pins are labelled in red in figure 13. In ATMega328p, the label inside purple blocks labelled 2-7 refers to the eservo 1-6.

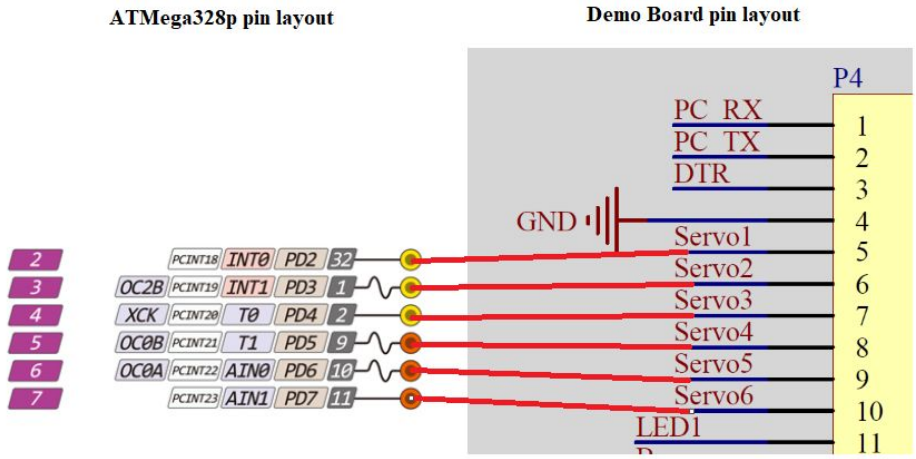


Figure 13. Demo Board Connection with ATMega328p

The input microcontroller gives to servo are **{servo_id, position, and time}**. The servo_id corresponds to which servo we are looking at, such as servo 1 to servo 6; the position ranges from 500 to 2500 and corresponds to 0 degree to 180 degree that each joint can rotate. However, the servo 6 on end effector can only rotate from 500 to 1500 which correspond to 90 degree. The time indicates how much time it will take the servo to rotate some specific angle. The shorter the time, the greater the speed the servos will have.

Requirement	Verification
1. Must have a load bearing of at least 400 grams which is the maximum weight of 10 shoes we have weighted	1. Our robot arm is able to lift all the shoes we prepare for our experiment setting

2.3.2.3 Robot Cart Motor

The two motors come with the cart kit we bought are two DC motor with 9V working voltage and load current 1200mA(max). We use a TB6612FNG driver IC to control those two motors. The maximum power supply voltage that motors can bear is 15V and output current is 1.2A in average. It has a standby system which can save power and CW/CCW/short brake/stop function modes. The circuit of motor driver control is shown in figure 14:

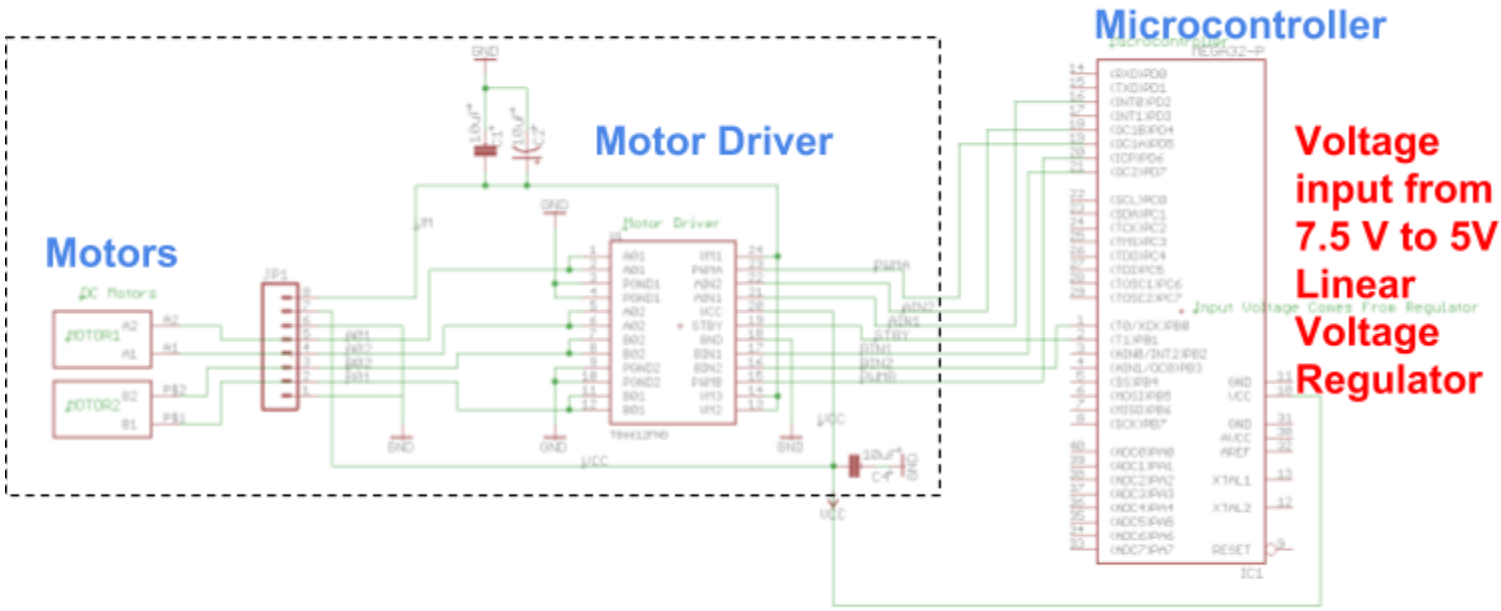


Figure 14. Schematic of Robot Cart Motor and Motor Driver

2.3.2.4 Load Cell (HX711 weight sensor)

The load cell will be implemented on the robot to weight each shoe. This load cell includes a weight sensor and a HX711 analog-to-digital converter module, as shown in figure 15. This module's current consumption under normal operation is smaller than 1.5mA and supply voltage range 2.6~5.5V. It has 4 pins to connect with microcontroller as shown in the graph below.When the robot gripper put shoe on load cell plate, it will send digital value of the shoe's weight to microcontroller.

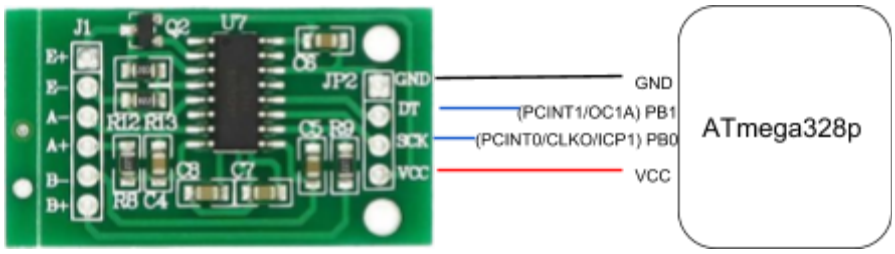


Figure 15. Connection of ATMega328p and Load Cell

2.3.3 Power Module

The power module will power the major circuit with the 12V IK-229679 rechargeable Battery. The power module will be able to provide sufficient power to other parts with 8Ah, considering the voltage decay during and the power consumption at the same time. Besides, to fulfill different voltage requirements, the power supply will be connected to a 7.5V switch regulator to activate the end effector, a 5V linear regulator to power the load cell, motor driver and PCB, and a 9V linear regulator for the motor. In addition to using the batteries, there will also be a 110V to 5V DC-DC converter powering the Raspberry Pi computer that is next to the camera, and the camera connected to the Raspberry Pi will then get powered automatically. The schematic of the power module is illustrated in figure 16. The 12V to 9V linear voltage regulator is parallel to the 12V to 7.5V switch voltage regulator. The 12V to 7.5V voltage regulator is in series with the 7.5V to 5V linear voltage regulator. During the meet with Professor and TA on Tuesday, our group realize that we should replace the 12V to 9V linear voltage regulator to a 12V to 9V switching regulator because compare to linear regulator, switching regulator has higher efficiency. Even though compare to linear voltage regulator, switch regulator will cause more more noise, but since motors don't cause many ripples at switching rate, the switch voltage regulator is the better solution for motors and in our later design, we will replace the 12V to 9V linear voltage regulator to a switch voltage regulator.

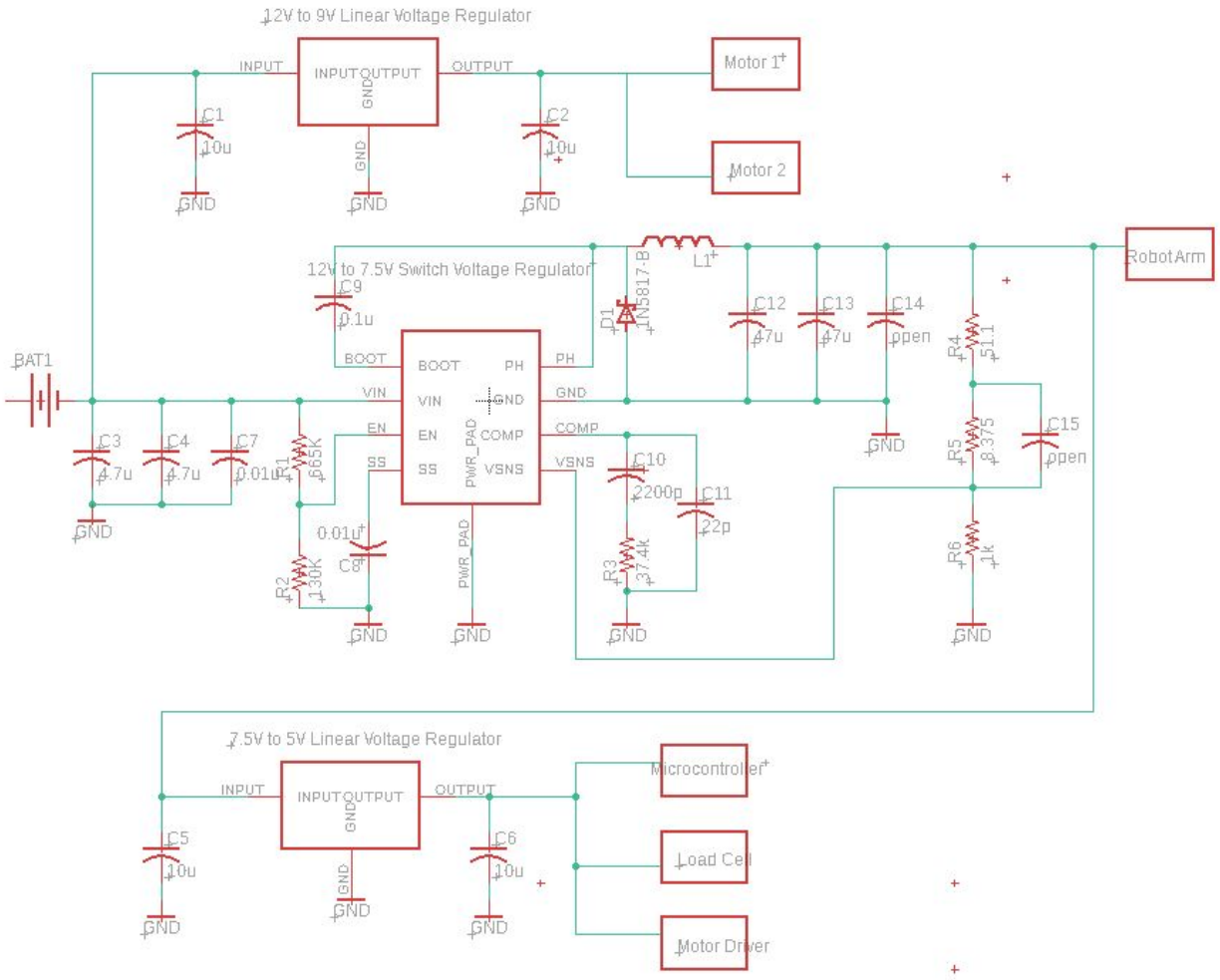


Figure 16. Power Schematic

2.3.3.1 Battery

We will use a 12V IK-229679 rechargeable Battery as the major power supply in our design. We use rechargeable battery because first of all, it is very convenient to use. We can always change it with a simple battery charger. Also, compare with disposable batteries whose voltage gets progressively lower, by using rechargeable batteries, we can get peak performance the whole time. we can always This 12V battery has a capacity of 8Ah so it is sufficient to provide enough power to other parts. Besides, according to our power design, we need 6A from the battery in total, and that is one of the reason why we choose this battery. Since the battery is rechargeable, we could always recharge the battery and keep the battery working at a good condition. The requirement and verification of this battery is provided in the following chart.

The reason that we use 12V instead of 24V is that, for the switch regulator TPS54531, the efficiency is lower and the power dissipation is less at 25 Celsius. As the voltage drop between

the input and the output of the switch regulator increases, more heat and power are lost so we decide to use 12V.

Requirement	Verification
1. Must be able to provide steady power to other parts. 2. The output voltage is at least 12V 6A. 3. The battery could still output steady power to other parts	1. Connect the battery to the oscilloscope and see if the voltage and current of the output of the battery is steady. 2. Use a multimeter to measure the voltage and current through this 12V battery. 3. Recharge the battery and connect the battery to oscilloscope and verify the output of this battery.

2.3.3.2 Power Supply for Raspberry and Camera

To power the camera, first we use the 110V to 5V DC-DC wall adaptor to power the Raspberry Pi computer and then we connect the camera through Raspberry Pi with the CSI-2 Camera Connector. The Raspberry Pi operates at 5V so there is no need to use voltage regulator here. Besides, the current through the Raspberry Pi is 2.5A, while the current through the camera 250mA so the Raspberry Pi is sufficient to supply the camera. The camera module and the raspberry pi is wired through a 15-pin ribbon cable.

2.3.3.3 Voltage Regulator

Different parts of our design require distinct voltage so we include three types of voltage regulators. The robot motor needs exactly 9V so we connect it with a 9V linear voltage regulator. To power the robot arm, we come up with an idea to use the 12V to 7.5V switch voltage regulator, which is more efficient than the linear voltage regulator. For other components, such as the microcontroller, load cell and motor driver, they both need 5V as input voltage. Therefore, we will use a 7.5V to 5V linear regulator in series with the switch voltage regulator, and then output of this 5V regulator is connected to the microcontroller, load cell and the motor driver.

For the 12V to 9V and 7.5V to 5V regulator, the voltage drop is relatively low and the linear regulators offer increased efficiency. Besides, the currents through the motors, microcontroller, motor driver and load cell are not very high so we choose linear regulators. However, the current through the 12V to 7.5V regulator is as high as 4A so it is not efficient to use the linear regulator because the linear regulator has larger heat loss due to high current. In addition, the voltage drop is higher, the effective power is relatively low in this case.

9V Voltage Regulator

This voltage regulator will convert 12V DC into 9V DC, which powers the robot motors. Since the speed of the motor depends on the current and the voltage through the motors, we decide to provide 1.2A current and 9V voltage to make sure the motors work properly. The chip LM7809 accepts 11.5-26V as input and then outputs 9V with a maximum current 2.2A[8], which is sufficient to run the motors with a moderate speed.

Requirement	Verification
1. The regulator should output 9V +/-5% and 2A 2. The regulator accepts 12V as input voltage	1. Connect the regulator output to a voltmeter and measure the voltage and the current. 2. The same method as verification 1.

7.5 V Voltage Regulator

With a TPS54531 7.5V switch voltage regulator, we can make sure that the robot arm works efficiently within its working voltage 6V-7.5V. This TPS54531 switch voltage regulator has a input voltage range of 3.5-28V and the output voltage could be as low as 0.8V with a continuous current 5A[9].

. The robot arm needs 3A current as input and this voltage regulator satisfy this requirement. In addition, the output of this switch regulator can be designed to a specific number such as 5V in our case, with the equations (1) and (2) below. The Vref is 0.8V, so it can be easily solved that the ratio of R5 and R6 is 8.375. We then choose R6 to be 1k ohm and R5 to be 8.375k ohm.

$$R6 = \frac{R5 \times V\,ref}{V\,OUT - V\,ref} \quad \textbf{(1)}$$

$$V_{OUT} = V_{ref} \times \left[\frac{R5}{R6} + 1\right] \quad \textbf{(2)}$$

Requirement	Verification
1. The regulator should output 7.5V +/-5% and 3A 2. The regulator accepts 12V as input voltage	1. Connect the regulator output to a voltmeter and measure the voltage and the current. 2. The same method as verification 1.

5V Voltage Regulator

This L7805A 5V voltage regulator is in series with the above switch regulator so it takes 7.5V as input and outputs 5V to the microcontroller, load cell and the motor driver. The maximum output current is 1.5A[14], which is sufficient to power the other three parts. The load cell requires a voltage input in the range of 3.3 to 5V and a maximum current input 30mA, so this voltage regulator works well. The microcontroller requires 5V as input so this voltage regulator also meet the requirements.

Requirement	Verification
1. The regulator should output 5V +/-5% and 1.5A 2. The regulator accepts 7.5V as input voltage	1. Connect the regulator output to a voltmeter and measure the voltage. 2. The same method as verification 1.

2.3.3.4 Power Calculation

The motor driver has an input of 5V with a maximum current 1.2A and consumes 0.89W while working. Each motor takes 9V as input and consumes 1.18W. In addition, ATMega328p uses 81.8mW with input 5V and 16.43mA. For the load cell, it consumes 150mW and the robot arm consumes 22.5W. First we calculate the power consumed by all three voltage regulator, as in the following calculation.

$$P_{12V\ to\ 9V\ linear\ voltage\ regulator} = V_{12V\ to\ 9V} \times I_{motors}$$
$$= (12-9) \times 2.2$$
$$= 6.6\ \text{W}$$
$$P_{7.5V\ to\ 5V\ linear\ voltage\ regulator} = V_{7.5V\ to\ 5V} \times (I_{robot\ arm} + I_{load\ cell} + I_{motor\ driver})$$
$$= (7.5-5) \times (1 + 0.03 + 0.01643)$$
$$= 2.616075\ \text{W}$$

To calculate the power dissipation of the switch voltage regulator, we find out that the efficiency of the switch regulator at 4A output in 25 Celsius is 90%. The power consumed by this voltage regulator is the output power minus the input power. The output power is calculated by the input power divided by the efficiency, as indicated in the calculation below.

$$P_{12V\ to\ 7.5V\ switch\ regulator} = \frac{V_{OUT} \times I_{OUT}}{efficiency} = \frac{7.5 \times 4}{90\%} = 33.33W$$

$$P_{voltage\ regulator} = P_{12V\ to\ 9V\ linear\ voltage\ regulator} + P_{12V\ to\ 7.5V\ switch\ regulator} + P_{7.5V\ to\ 5V\ linear\ voltage\ regulator}$$

$$= 42.55\ W$$

Next, we add up all the power consumed by all the parts as in the following calculation.

$$Total\ Power\ P = P_{motor\ driver} + P_{motors} + P_{microcontroller} + P_{robot\ arm} + P_{load\ cell} + P_{voltage\ regulator}$$
$$= 0.89 + 1.18 \times 2 + 0.0818 + 22.5 + 0.15 + 42.55$$
$$= 67.3538\ W$$

2.3.5 Software module

The software module in our design takes the image captured by the camera as input from Raspberry Pi and then process the image and then sends the property of each shoe such as color, coordinates, object number to ATMega328p.The software module flowchart is shown in figure 17. We will associate our object in the image using two raster scans.

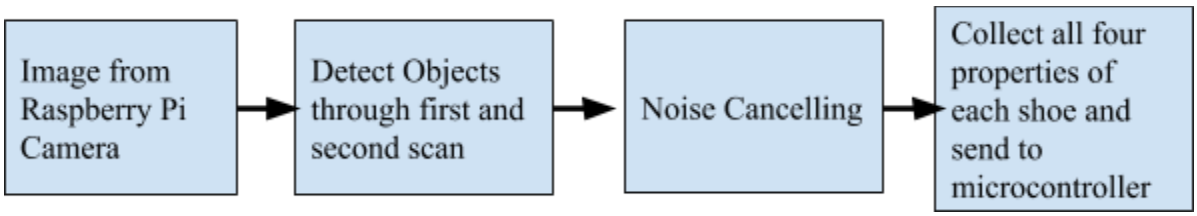


Figure 17. Software Module Flowchart

After running the first raster scan, we seperate shoes(label 1) from background(label 0) and assign all the shoes with one label. In the second raster scan, we assign the same label to all pixels in the same objects. In this way, all the pixels inside the same object share the same label.

After running two raster scans, we are able to distinguish shoes from the background. After we have identifies all the objects with unique labels, we will perform “noise elimination” and discard objects that are too big or too small. For objects lie outside of range, we change its color into white and thereby forcing the objects into background. After noise elimination, the number of legitimate objects and their corresponding properties will be reported to microcontroller. And figure 15 is a simulation of software module after complete objects association and noise

cancelling, each square represents an individual wooden block and we assign each block a different color. The simulation result is in figure 18.

Figure 18. Simulation After Object Recognition and Noise Cancelling

2.4 Tolerance Analysis

Position Deviation: (1) Speed Control

As the robot starts to work, it needs to get to the location on the mat side that is closest to target shoe depending on its coordinates with Euclidean distance. Therefore we need to figure out the speed of robot so that we can power the motor with certain voltage and current, and control the motor to that specific location. Our robot motor has an angular velocity of 100 revolution per minute when it is loaded under active mode. By applying the equation (3), we are able to convert the angular velocity of motors into linear velocity in unit of meters per second.

$$100\ \frac{rev}{min} * \frac{1min}{60\ seconds} * \frac{circumference\ \ of\ our\ motor\ in\ m}{rev} = 1.667\ \text{of motor circumference} \quad (3)$$

Also, since the motor has a stable speed, we are planning to do trials and record the time it takes the motor to travel for example 15 cm, 30 cm, 45 cm, 60 cm up to 100 cm and try to find its accurate speed using distance/time. After we know speed, we can deduce how much time we should power the robot and decide what the voltage and current we need to make it move to the destination. The more precise the velocity is, the less deviation there will be from robot's ideal position.

Position Deviation: (2) Closed Loop Control

Even after we find a velocity that is accurate enough, in order to compensate the deviation from ideal position, we can keep track of where the robot is around the mat. The solution our group

come up with is we are going to place a red mark on the load cell of our car. It will be in a different size and color from the shoes so the robot will not get confused and recognize the mark as a shoe.

Tracking the robot involves a closed loop control and the block diagram of closed loop control is in figure 19. During the process of picking up shoes, we first calculate a position and control the robot to move to that location. After robot stops moving, the camera will update the current image so we can always know the updated position of the mark. Then we can always use the mark on the car to track the position of the robot and check if our robot reaches that destination. The Raspberry Pi receives images from the camera module and analyzes the coordinates of that mark/robot and sends this information to ATMega328p. With the feedback from Raspberry Pi, we can then verify if the car moves to the desired location. If the car fails to move to the correct location, we then calculate the distance between the current position and destination and design a new path. If the car reaches the correct position, we then move to the next instruction to pick up shoes.

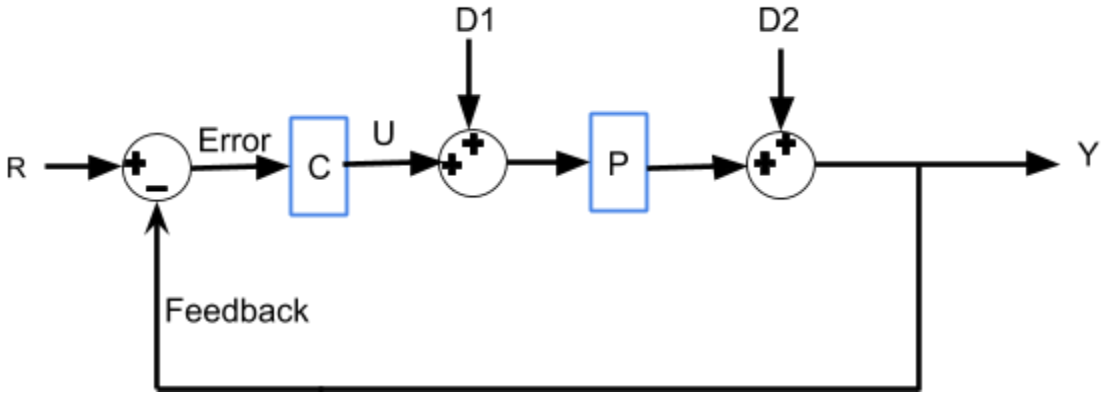


Figure 19. Closed Loop Control

R(Reference): The destination coordinates of the robot

E(Error): The difference of the destination coordinates and the current coordinates

C(Controller): Calculate the desired voltage to drive the motor so that the robot could reach the destination, which is done in ATMega328p

U(Input): The actual voltage driven to the motor

D1(Disturbance 1): The voltage loss between the ideal voltage we want and the actual voltage to the motors

P(Plant): The motor of the cart

D2(Disturbance 2): The friction encountered during running the cart motor.

Y(Output): The current coordinates of the robot

Here, D1 and D2 are acted as compensations to the closed loop control.

Via Point

After the robot arrives at the ideal position, in order to get to position on the the side view graph 18 below, the the base servo 6 need to rotate 90 degree from the original direction which is aligned with robot moving direction. After the first rotation, there are three stop points in the process of picking up shoe which are represented as the three red circles in the graph. The joint with servo 2 will shortly stop on circle position 1, then move to position 2, and finally move to circle position 3, which is the position of picking the shoe. Those three points are on a straight line perpendicular to ground and go through the center of shoe. The reason we set the three stop points is to let the gripper approach the shoe from top of the shoe without touching it, and that touching it before closing gripper may change the shoe position.

In our algorithm, we need to calculate the joint angle values of position 3 first. In the graph, d is the distance between shoe and robot(mat side) which is already got from image processing. 22.5cm is the height of robot base and the 17.5 cm is the height of servo 2 (red circle 3) when the gripper touches ground, the length of robot arms are represented as L1 and L2. First of all, we can find length c from known length d: $c = \sqrt{5^2 + d^2}$. Next from Law of Cosine $c^2 = l_1^2 + l_2^2 - 2 l_1 l_2 \angle C$ we can calculate the angle $\angle C = cos^{-1}(\frac{-c^2 + l_1^2 + l_2^2}{2 l_1 l_2})$. Therefore, we can calculate the three joint angles θ1, θ2, θ3: $\theta 1 = \square - [tan^{-1}(\frac{d}{5}) + cos^{-1}(\frac{-l_1^2 + l_2^2 + c^2}{2\, c\, l_2})]$, $\theta 2 = 180° - \angle C$, and $\theta 3 = 180° - \theta 2 - \theta 1$

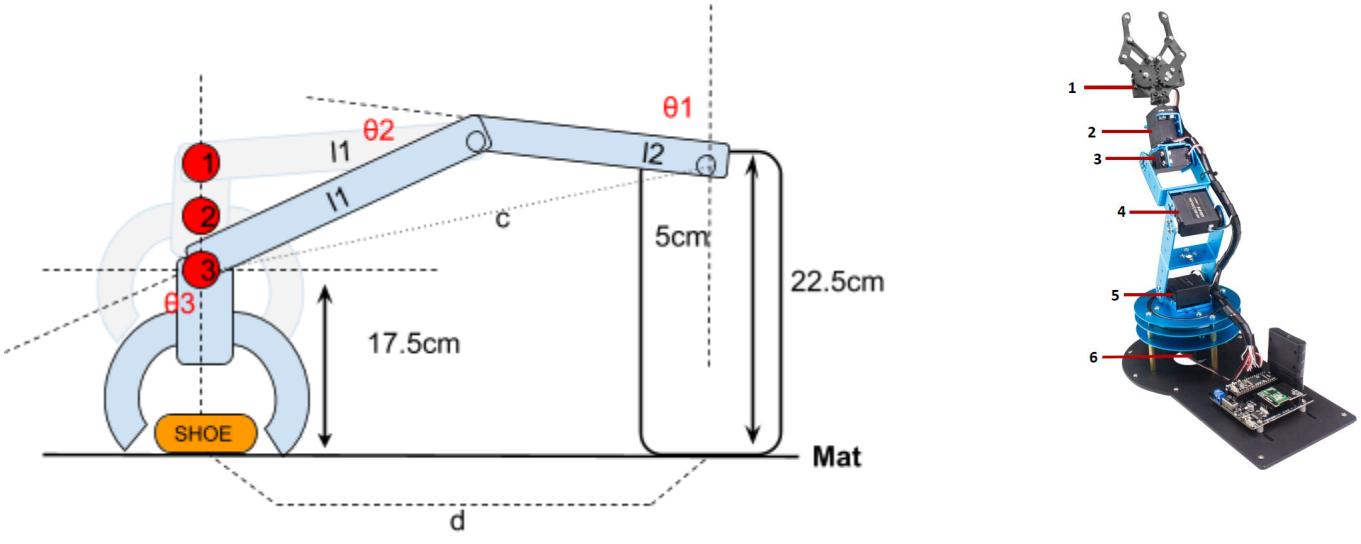


Figure 20. Robot arm side view

Considering about position 2 and 1, the only difference is the height of servo 2, we now set the height of position 2 to be (17.5+1)cm and position 3 to be (17.5+2)cm with respect to ground. This difference may be adjusted in future trials in order to make the robot move smoothly. After the same calculation of position 2 and 1 as above, we can obtain three sets of θ values. θ1 is applied to servo 5, θ2 is applied to servo 4, and θ3 is applied to servo 3.

At the position 1, the robot arm need to rotate the gripper in order to catch the shoe in a proper direction. The graph below is the top view of gripper and shoe and the graph of left is the original gripper position. We represent the angle between shoe center line and mat side as theta. Since servo 2 control the gripper direction can only rotate 180 degrees, we let the servo only rotate in counter clockwise direction. For case 1, angle θ is smaller than 90 degrees, the gripper need to rotate θ+90 degrees, for case 2 the angle θ is larger than 90 degrees the gripper rotate θ-90 degrees.

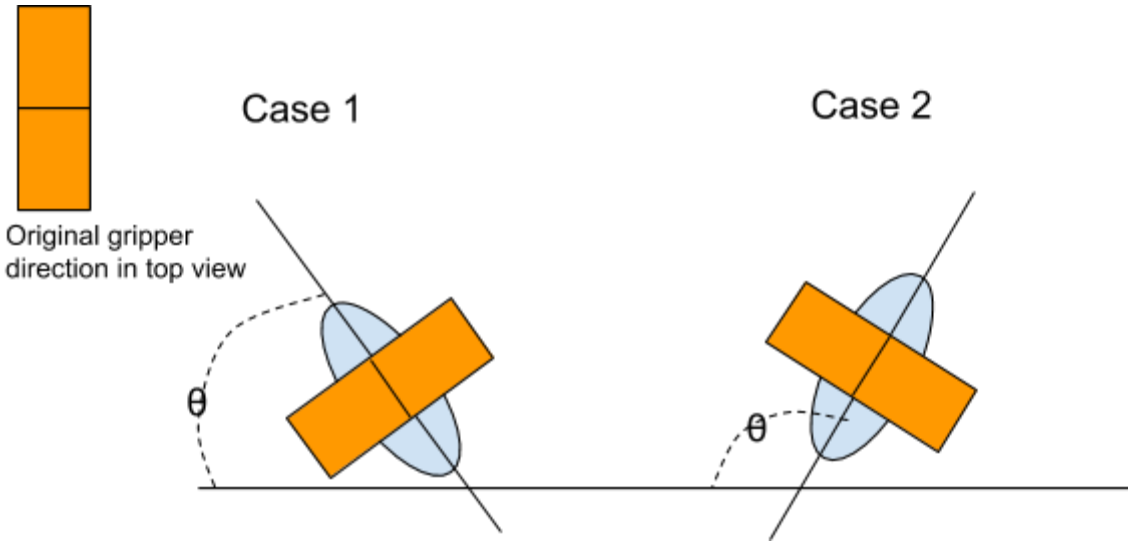


Figure 21. Gripper Direction Adjustment

Under our experiment setting, we think that as long as the gripper can grab the shoe between +/-15% of length A-B from center point, the gripper is able to grab the shoe without falling.

Via Point Tolerance Range

The ideal destination for the robot is the point where it is closest to the shoe, however, we set a tolerance range that even if the robot has not stand exactly at the ideal destination, it can still reach the shoe as long as it stays within tolerance range.

i) When the angle between the line A-B and X-axis is less than 90 degree (graph of version 1 below), Raspberry Pi needs to calculate shortest the **distance d1 and d2** between the point that lies at **15% length of A-B** above center point(represented as an **orange circle**) and robot path. We have two versions of graphs below, one representing angle Alpha smaller than 90 degree and one representing angle alpha greater than 90 degree. In both versions we are able to calculate the distance d1 and d2.

ii) Then Pi will transmit the **coordinates of the orange circle and the calculated distance d** to microcontroller, then we can figure out a tolerance analysis delta X1 and X2 of robot's position by setting the inequalities: $\sqrt{d1^2 + delta\ X1^2} \leq 400mm$ and $\sqrt{d2^2 + delta\ X2^2} \leq 400mm$

iii)Since the length of robot arm is fixed and distance d is known, we can calculate the **deviation delta_X** that is acceptable for the robot to function properly. We then try to figure out the **projection length** of the 0.3 A-B onto the robot path, as indicated by the **orange segment** shown on the graph.

From all the calculation above,even if the robot is not standing at the ideal destination, it still is able to rotate joint angles and reach the shoe as long as it stays in the tolerance range:

$$delta\ X_1 + delta\ X_2 + 0.3L * sin(90 - angle\ Alpha)$$

Shoes Falling Condition

After the gripper grabs the shoes, it will carry it to the load cell. However, we cannot be sure if the shoe will not fall during this process. We will determine the presence of shoe from the feedback of the load cell to microcontroller:

(i) If the load cell has a digital signal back to microcontroller that indicates some weight, we decide that the shoe has been placed on the load cell.

(ii) Otherwise, if the signal indicates very small or zero weight, we assume the shoe must fall during the process.

The figure 22 demonstrate the calculation of the tolerance analysis of the robot position.

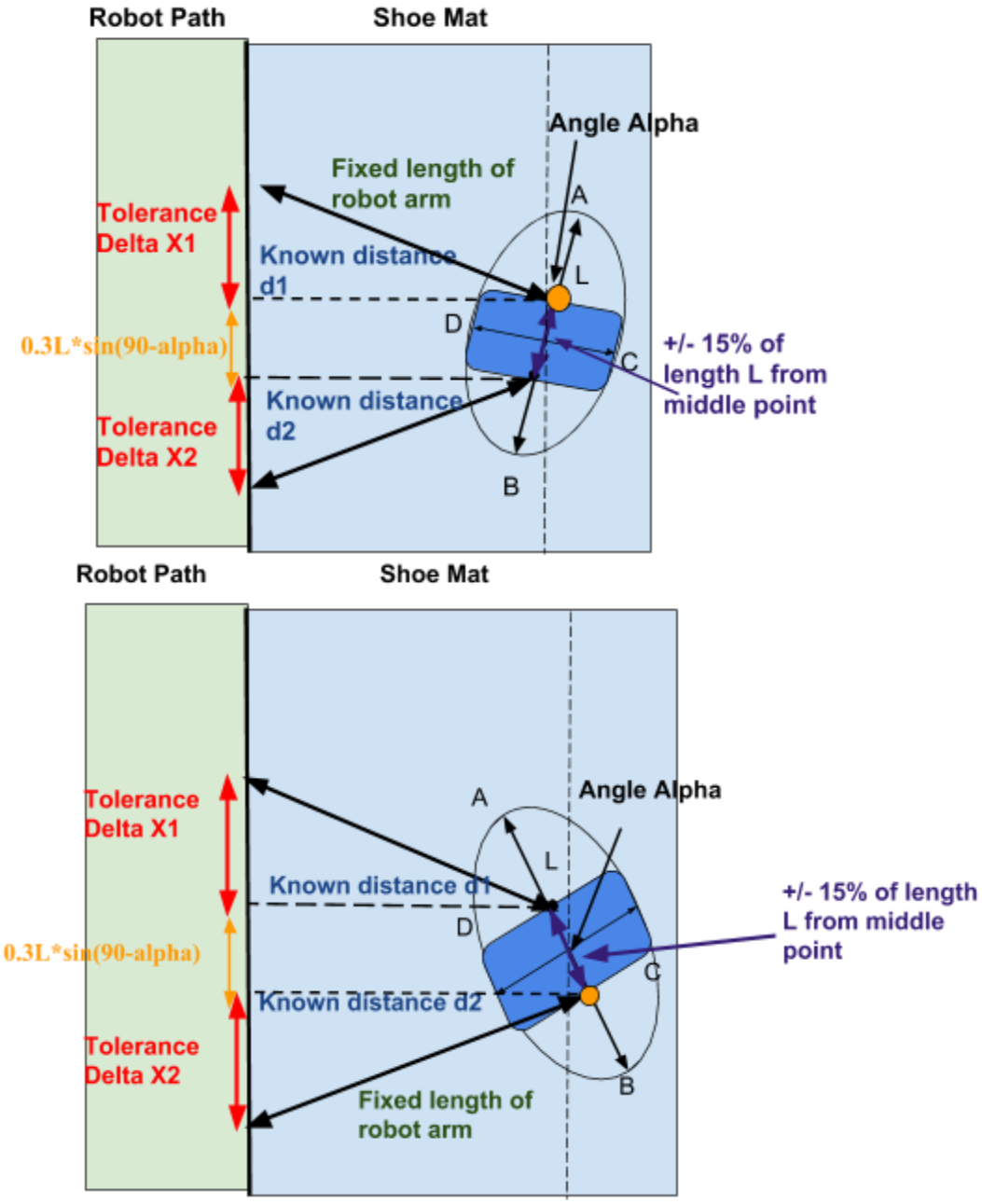


Figure 22. Tolerance Analysis of Robot Position

As long as the load cell doesn't detect any weight, the camera will update the image and Pi will calculate where the current shoe is. If the shoe has fall back to the mat, we make the robot go back to its starting position and start all over again (start finding the closest shoe to the robot path). However, if the shoe has fallen on the robot path, we come up with a new algorithm to pick up shoes. Our robot arm can stretch up to 400 mm and down to 150mm. Therefore as long

as inequalities below satisfy, the robot arm is guaranteed to catch up the shoe.

$$d + \frac{0.3L*sin(angle\ Alpha-90)}{2} < 400 \text{ and } d - \frac{0.3L*sin(angle\ Alpha-90)}{2} > 400$$

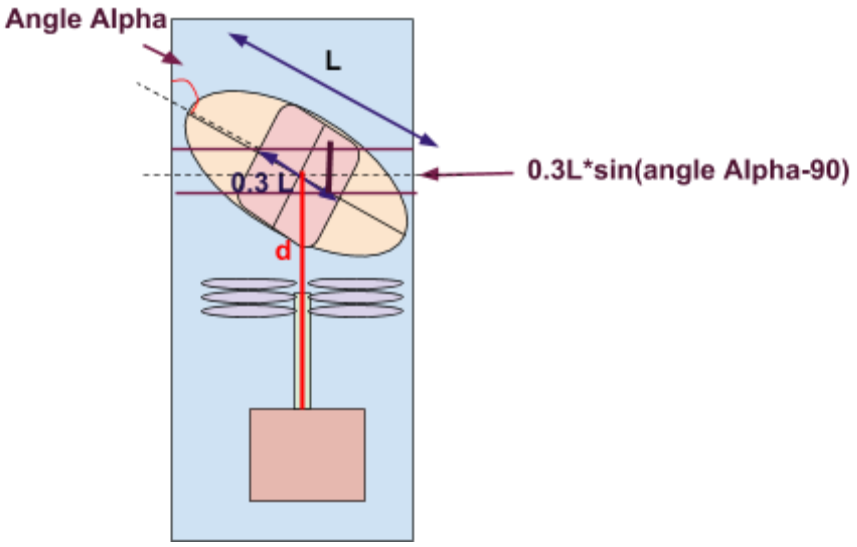


Figure 23. Tolerance Analysis of Gripper Angle

After the robot successfully place the shoe on the load cell and then pick it up, it will know that its following mission is to place the shoe in the storage area as long as the weight signal from load cell goes to low from high. After the robot arm place the shoe on the corresponding storage area following the pairing instruction from microcontroller, the camera will update the image it captures and see if we have place the shoe in correct area. If the shoe falls during the process, the camera will detect its new location and decide if the robot should follow the algorithm to pick the shoe up from mat or from robot path.

The figure 24 below discuss about the situation when control module recognize the robot car and the center of shoe are on a horizontal line, but in reality, there is an offset that the car doesn't arrive on the destination exactly. In this case, the robot arm will still stretch out in the direction perpendicular with mat side and do the pick up task. In the graph, red point is the standard position of picking shoe, pink points are position of gripper when the robot car are L away from the wanted position. To calculate the max L which still allow the gripper to pick shoe up. We assume the angle between shoe and mat side is θ. Then we calculate the green segment to be half of the gripper(7cm) minus half of the shoe(4cm) and minus 1 cm (we need to keep at least 1 cm gap between the gripper and shoe to avoid conflict). Therefore, the green segment has a

length of 2cm. By using the properties of right triangle, we could get the L =2/sin(θ). In conclusion, as long as the offset between robot and shoe on horizontal line is smaller than L, the pick up movement can still success.

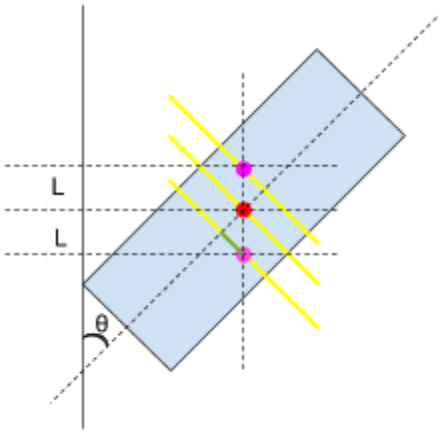


Figure 24. Robot Position Offset

3. Cost and Schedule

3.1 Cost Analysis

3.1.1 Parts and Components Cost

Description	Quantity	Unit Price	Cost
A4988 Stepper Motor Driver Carrier	2	5.95	11.9
CSTCE16M0V53-R0 16MHz Ceramic Oscillator	1	0.5	0.5
100 uF Electrolytic Capacitor	2	0.35	0.7
ATmega328-P	1	2.01	2.01
Raspbery Pi 3 Model B	1	35.8	35.8
Energizer A23 Battery	4	9.71	38.84
Energizer A23 Battery Holder(2 pc)	1	3.18	3.18
Raspberry Pi Camera Module V2	1	29.95	29.95

Raspberry Pi 3b Power Supply	1	10.99	10.99
9V Voltage regulator	1	0.15	0.15
5V Voltage regulator	1	1.59	1.59
6.4V Voltage regulator	1	1.21	1.21
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
Total			478.08

3.1.2 Labor Cost

We have chosen an hourly wage from a salary info sheet provided by engineering at Illinois for Electrical Engineering graduate[10].

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

3.1.3 Grand Total

Grand Total = Parts and Components Cost + Labor Cost = 478.08 + 52650 = 53128.08 dollars

3.2 Schedule

Week	Jinghan	Quanhua	Mingxi
2/25	Prepare for Design Review	Prepare for Design Review	Prepare for Design Review
3/4	Help verify all the components and begin the software module with Quanhua.	Begin the framework for software module.	Purchase needed components and begin verifying that all components work properly.
3/11	Complete software module to receive image, filter and raster scan images.	Work on software and bluetooth module to generate instructions to microcontroller.	Complete verification process of all needed components.
3/18	Work with Quanhua and Mingxi on assembling the physical design and hooking up the electrical components.	Start assembling the physical design .	Start hooking up the electrical components.
3/25	Complete physical and electrical design	Complete physical and electrical design	Complete physical and electrical design
4/1	Start final testing of the complete system and focus on any issues related to image processing	Start final testing of the complete system and focus on any issues related to physical design of robot.	Start final testing of the complete system and focus on any issues related to data transfer from Raspberry Pi
4/8	Iron out any software application and data transfer performance issues	Stabilize physical design by making sure that all physical components and circuitry are safely secured and demo-ready.	Iron out any performance issues related to physical components and circuitry
4/15	Do numerous executions for testing	Do numerous executions for testing	Do numerous executions for testing
4/22	Complete entire system check and begin final report	Complete entire system check and begin final report and final presentation	Complete entire system check and begin final presentation
4/29	Finish final report	Finish final report and final presentation	Finish final presentation.

4. Ethics and Safety

4.1 Ethics

After reading the IEEE Code of Ethics[1] and the ACM Code of Ethics and Professional Conduct[2], 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"[1].

4.2 Safety

There are two underlying safety issues that we will face. The first one is that in our robot design, we need to convert 12V power to 5V 3A by a regulator. The 3A current might be relatively high with respect to human being and people will get hurt easily by improper touching. The maximum current an average man can grasp and let go is 16 mA and 2A current can cause cardiac standstill and internal organ damage.[3] In addition, the Lithium Battery with a poor wiring and bad connection will lead to safety issues as well. Therefore, we must make sure that the connections between elements are stable and are in good conditions.

The second issue is the risk of soldering. Soldering can be dangerous if we inappropriately manipulate the soldering tools. Also we need to be careful of the high temperature during soldering and remember to turn the tools off when we are done.

5. References

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

[2] 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].

[3] Raymond M. Fish, L. (2019). Conduction of Electrical Current to and Through the Human Body: A Review. [online] PubMed Central (PMC). Available at: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC2763825/ [Accessed 7 Feb. 2019].

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

[5] 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=EAIaIQobChMIyYvm99Dn4AIVlYxpCh3DQwzGEAQYAyABEgKdcvD_BwE&gclsrc=aw.ds.

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

[7] Raspberry pi 3b power supply
https://www.amazon.com/Miuzei-Raspberry-Heatsinks-Supply-Compatible/dp/B07BTHNW9W/ref=sr_1_7?keywords=raspberry+pi+3b+5V&qid=1550715692&s=gateway&sr=8-7

[8] 9V Voltage regulator
Continental Device India Limited,”3-TERMINAL POSITIVE VOLTAGE REGULATOR”, LM7809 datasheet
https://static1.squarespace.com/static/5416a926e4b09de8832655bc/t/54427037e4b03de3b67b895a/1413640247188/lm7809.pdf

[9] 7.5V Switch Voltage regulator
Texas Instruments,”, Step-Down SWIFT™ DC-DC Converter With Eco-mode”,TPS54531 datasheet, May.2013[Revised October.2014]
http://www.ti.com/lit/ds/symlink/tps54531.pdf

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

[11]Raspberry Pi Camera Module v2
Camera Hardware¶." *Picamera - Picamera 1.13 Documentation*, picamera.readthedocs.io/en/release-1.13/fov.html.

[12] Arduino Uno or Raspberry Pi 3
How to Decide. (2018, September 10). Retrieved from https://www.arrow.com/en/research-and-events/articles/comparing-arduino-uno-and-raspberry-pi-3

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

[14] 5V voltage regulator
STMicroelectronics, "Positive voltage regulator ICs", L78 datasheet, Revised Sep.2018 https://www.mouser.com/datasheet/2/389/l78-974043.pdf

[15] Raspbian Software Command for testing CPU processing speed
Raspberry Pi Stack Exchange, "How to Get the Exact Clock Speed at the Raspberry Pi Is Running?" raspberrypi.stackexchange.com/questions/8201/how-to-get-the-exact-clock-speed-at-the-raspberry-pi-is-running.

[16] Raspbian Software Command for testing RAM size
Raspberry Pi Forums, www.raspberrypi.org/forums/viewtopic.php?t=20602. https://www.raspberrypi.org/forums/viewtopic.php?t=20602

[17] Learn About Atmega328p Fuse Bits and How To Use Them with an External Crystal Oscillator
Charles Hampton
https://www.allaboutcircuits.com/projects/atmega328p-fuse-bits-and-an-external-crystal-oscillator

Appendix

Code for First Raster Scan

```
int label[height*width];
int *equiv[height*width];
for(int i=0; i<height; i++)
{
    for(int j=0; j<width; j++)
    {
        equiv[i*width+j]=&label[i*width+j];
        if (bw_img.data[i*width+j]==255)
            pixellabel[i][j] = -1;
        else if (bw_img.data[i*width+j] ==  0)
            pixellabel[i][j] = 0;
    }
}
```

```
int bg=-1;
int obj=0;
int labelnum = 1;
// FIRST raster scan
for(int i=0; i<height; i++)
{
  for(int j=0; j<width; j++)
  {
       int Left, Above;
       double smallerbaselabel;
       int min, max;
       int Pixel = pixellabel[i][j];
       if (i>0){
            Above = pixellabel[i-1][j];

       }else{
            Above=bg;
       }
       if(j>0){
         Left=pixellabel[i][j-1];

       }else{
          Left = bg;
       }
       if (Pixel != bg){
          if (Left == bg  && Above==bg){
              pixellabel[i][j] = labelnum;
              label[labelnum]=labelnum;
              labelnum++;
          }
          else if (Left!=bg && Above == bg){
              pixellabel[i][j]=Left;
          }
          else if (Left ==bg && Above !=bg){
              pixellabel[i][j]=Above;
          }
          else if (Left !=bg && Above !=bg){
                smallerbaselabel = fmin(*equiv[Left],*equiv[Above]);
                if(smallerbaselabel==*equiv[Left]){
                      min = Left;
                      max = Above;
                 }else{
                      min = Above;
                      max = Left;
                 }
           pixellabel[i][j] = smallerbaselabel;
           *equiv[max] = *equiv[min];
           equiv[max] = equiv[min];
         }
       }
    }
  }
```

Code for Second Raster Scan

```
//SECOND raster scan
  for(int i=0; i<height; i++)
  {
    for(int j=0; j<width; j++)
    {
      int Pixel = pixellabel[i][j];
      if (Pixel != bg){
        pixellabel[i][j] = *equiv[Pixel];
      }
    }
  }
```