UM–SJTU Joint Institute

Design and Manufacturing I
(VM250)

Robot Arm with a Soft Silica gel Gripper

Group 8

2019.4.26

Contents

Acknowledgement

We would like to express our deep gratitude to Professor Ju, our instructor, for his patience, wise guide and useful knowledge. Without his teaching, we won't have the idea about how to complete the project.

We would also like to express our gratitude to Lab Technician Otto Zhang for his practical advise and help. He gave us a lot of suggestions and help us fix the questions. In addition, Teaching Assistant Yisi Wu and Kai Xiao teach us a lot, too.

We would also want to extend our gratitude to our teammates. Without them help, discussion and tolerance, we won't build such a device.

At last, we would like to express our gratitude to non-ideal performance on the game day, because learning how to fail is actually one of the most important things that VM250 tries to deliver. As Friedrich Nietzsche said, "What Doesnt Kill You Makes You Stronger." We will regard this experience as a precious lesson, and apply what we learned this time to later projects.

1 Abstract

Grasping and moving objects is an essential problem in robotics, especially for those with different shapes. This function is significant in some specific application, such as the delivery. The objective of this project is to develop a robotic arm with a soft gripper to grasp objects, and move them from one place to another using remote control. We using a mechanical structure with rotational robot arm and platform to implement the motion, and soft gripper with oblique fingers, uneven cavity size, tough surface, "fingernail" and "web" to grasp objects. The resulting device is flexible to grasp many kinds of objects with different shapes and size, with to be improved operations. We find two ways to improve the curvature of soft gripper under fixed pressure. The first way is to increase the number of cavity at a given length, and another way is to make cavity with uneven cavity size. In addition, we find limit of hardware and actual deviation from ideal situation are necessary to be taken into the consideration when designing. Analyzing theoretically first is necessary rather than practicing in hurry.

No.	Method	Major findings	Refs.
L1	Rapidly Actuating Pneumatic-Nets	pneumatic-net that is better suited for high-speed, large-amplitude motion	"Pneumatic Networks for Soft Robotics that Actuate rapidly"
L2	Universal robotic gripper	hardware,software complexities more controllable joints	"Universal robotic gripper based on the jamming of granular material"
L3	Our approach	Diamond structure uneven internal cavity structure traditional craft skills	Proposed work

Table 1: literature review

2 Introduction

Grasping objects can be a very important problem in robotics. How to grasp objects with different shapes and size is significant to the application, such as the delivery, because it is rare to meet the situation that the objects have the identical shape and size. Therefore, we manage to create a complex structure to grab objects and move from one place to the other by remote control in this project. We are exploring some ways to create a robot arm and gripper to finish this task.

The relevant literature is shown on Table 1.In literature 1 ,the author developed a new design for a pneumatic-net that is better suited for high-speed, large-amplitude motion than the design described previously.

In literature two,the approach in the article introduces hardware and software complexities. There are large numbers of controllable joints to reach the need for force sensing if objects are to be handled securely without crushing them, and the computational overhead to decide how much stress each finger should apply and where.

Our method is to use diamond structure and uneven internal cavity structure with traditional craft skills.A diamond structure helps to catch the bottle when the gripper is vertical and uneven internal cavity structure is easy to inflate. What's more, some traditional craft skills like finger nail or fins structure will also help to grab the things.

What we need to achieve the goal are:3 DoF to move within the zone, 1 DoF to decide whether to grab or drop and 1 more DoF to pour the paper stars. So we choose a robot arm and a rotating platform to move, a soft gripper to catch and a wrist to rotate and pour.

3 Design

According to the game rules, our robot need to grab objects with different shapes and sizes within a ring to a basket. Plus, we have a special task which is to pour the luck stars in a bottle into the basket.

Therefore, we come up with the following basic design ideas. Generally speaking, our robot should have 5 degrees of freedom, 3 to move within the zone, 1 to pour the luck stars (rotate the bottle) and 1 to grab and drop objects. The first four degrees of freedom can be achieved by motors and servos, while the last one requires flexibility to catch objects with different shapes, which leads to the soft gripper. Thus, we mainly divide our robot into the following two parts.

3.1 Robot Arm

Since the game zone is a ring with inner diameter 0.35 m and outer diameter 1 m, we decide to use a moving strategy similar to polar coordinates, namely, a rotating platform and an arm to move forth and back and up and down. In order to pour the luck stars into the basket, we add a wrist (revolution joint) at the end of the gripper.

Then we divide our robot arm into 3 major parts, the rotating platform, the arm and the wrist to rotate the bottle and install the gripper and the suction cup (Figure 1, 2 and 3).

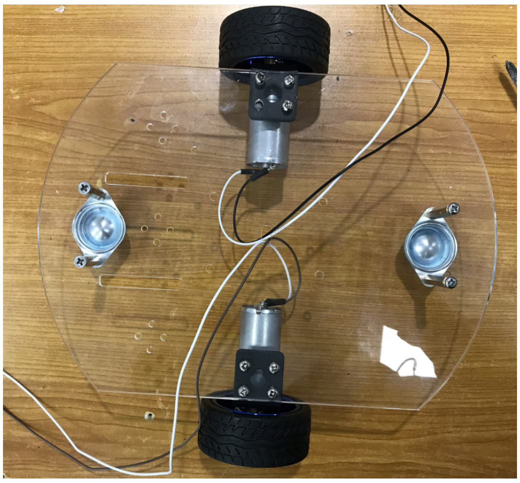


Figure 1: Rotating platform

The rotating platform is driven by two DC motors (and we add 2 universal wheels to keep it horizontal). It is designed large enough to install all the components and prevent tipping. It has two decks. The higher one is for the pneumatic system while the lower one is for the electric control system and the power supply. Also, the arm is installed at the edge of the platform to shorten the arm length and reduce the torque load on the bottom servos.

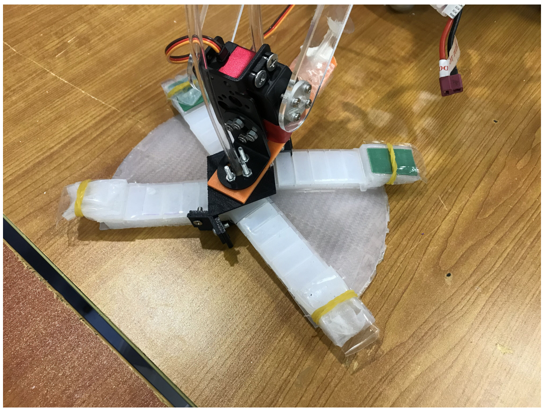


Figure 2: Wrist

The wrist is designed to fit the soft gripper. Four small holes are drilled to connect the servo bracket by screws and nuts and one big hole for the pipe.

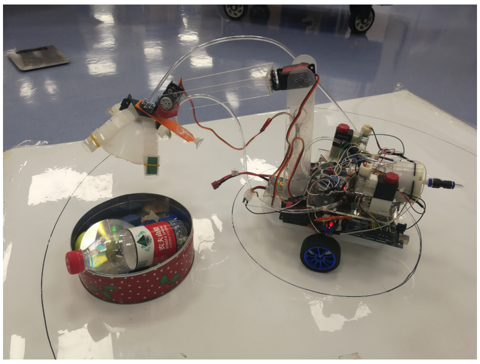


Figure 3: Whole assembly

The arm can be seen in the whole assembly. We design the two revolution joints in the same vertical plane of the wrist for the sake of convenience. (Otherwise, a specially designed component to change the rotation axis is needed.) It is a simple imitation of human arms. The components and assembly processes will be shown in Manufacturing/Assembly section.

3.2 Soft Gripper

The soft gripper is made of silica gel and use the mould printed by 3D printer to build its shape.In order to have a more curved gripper at a given pressure, we design cavity with uneven width to allocate the pressure to somewhere more efficient–the tip of each finger. Thus, the more fingertips bend during inflation, the easier it is to hold an object. Also, we design some traditional handicrafts to help grab objects: each finger is equipped with so-called ”fingernail” to stuff it under the object and hold it against it; we add two fan-shaped so-called ”web” with tough surface to increase the friction; we set the angle between neighbor finger by 60 or 120 degrees instead of 90 degrees to grab the bottle more tightly.

4 Manufacturing

4.1 Selection of Materials

Different components have different functions so that we should choose materials respectively.

4.1.1 Component Materials

Since we want to minimize the torque load on servos, we should choose a material with a low density and enough tensile strength for the arm and rotating platform (the analysis will be shown later). Therefore, Acrylic is a good choice. Another advantage is its low price.

The brackets of servos are made of aluminum which is also strong and light. Also, they are manufactured in a large scale. Thus, the price is also cheap.

As for the support between the decks of the platform and the universal wheels, we use copper columns. Those slender bars can carry a large load with small deflection.

The soft gripper is made of silica which has a good flexibility and can be cast at room temperature.....(maybe more advantages)

4.1.2 Actuators

We need 1 actuator to pump air, 1 system to keep or release the air, 1 for the rotating platform, 2 for the arm and 1 for the wrist.

The air pump and the solenoid valves are provided by the instructor (which we can not choose). They require a 12 voltage DC power supply. Thus, we choose 12V DC motors as our actuators for the rotating platform so that they can share the same power supply with the pneumatic components. Another two reasons for our choice is the cheap price and the fast rotation speed (which means a high efficiency).

As for the servos, we choose the most common ones. They are cheap and stable. Plus, we can get the brackets to install them onto our platform easily on Taobao. The only drawback is the torque one servo provides may be not enough. So we double the number.

4.2 Manufacturing Procedure

Since we need to install our chips, servos and motors on the Acrylic boards and take the wiring into account, there are many holes in them. Also, the outer shape is circular. Therefore, we choose laser cutting as the manufacturing procedure.

The connection between the soft gripper and the wrist is customized. Hence, we 3D print it. Thus, the material is PLA.

The other components are all products manufactured in large scale in factories (e.g. servo brackets, copper columns). We can only choose proper types but not the manufacturing procedure.

The manufacturing procedure of the soft gripper can mainly be divided into four steps: First, design and print the 3D mould using 3D printer;Second, use the silicon rubber to make the body part of the gripper; Third, make a silicon rubber base and cut it into the desired shape; Finally, equip fingers with fingernail.

4.3 Assembly

The assembly processes are as follows. First, we assemble the rotating platform. The components and the assembly are shown in Figure 4 and Figure 5.

Figure 4: Components

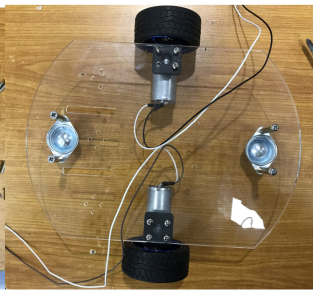


Figure 5: Assembly

Then we install the rear arm and the servos onto it.

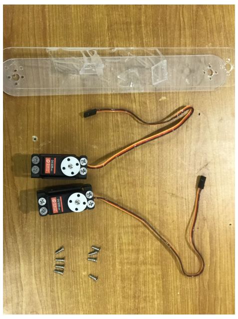


Figure 6: Components

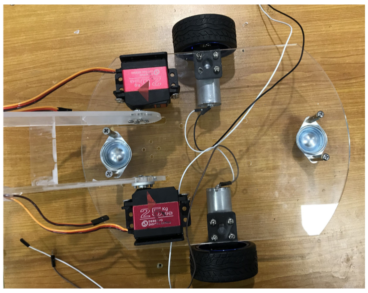


Figure 7: Assembly

After that, we install our control system onto the lower deck.

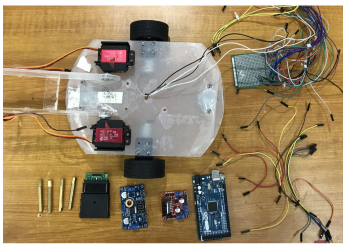


Figure 8: Components

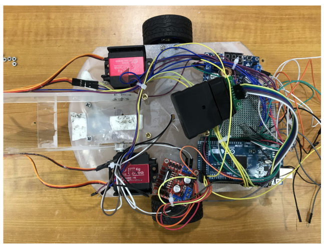


Figure 9: Assembly

Then we install all the pneumatic components on the upper deck.

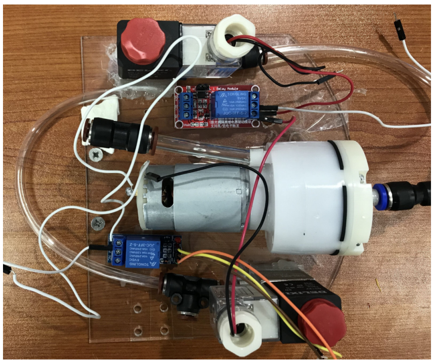


Figure 10: Upper deck

Next step is to install the front arm and the wrist.

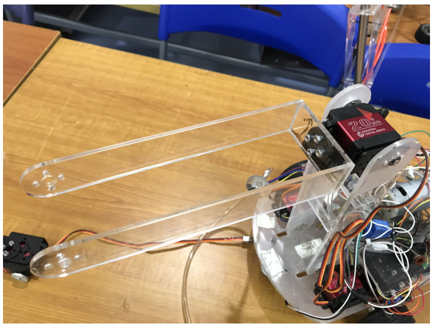


Figure 11: Front arm

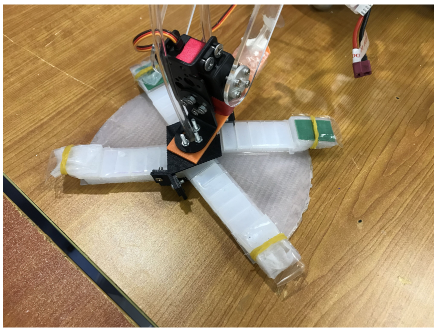


Figure 12: Wrist

The final step is to add the upper deck and gripper to the assembly and connect the pipes.

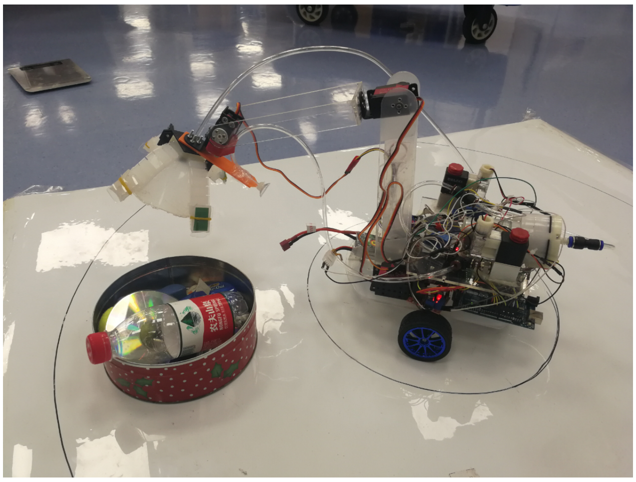


Figure 13: Whole assembly

5 Analysis and Test

5.1 Static Analysis

The free body diagram of the whole machine can be expressed as followed:

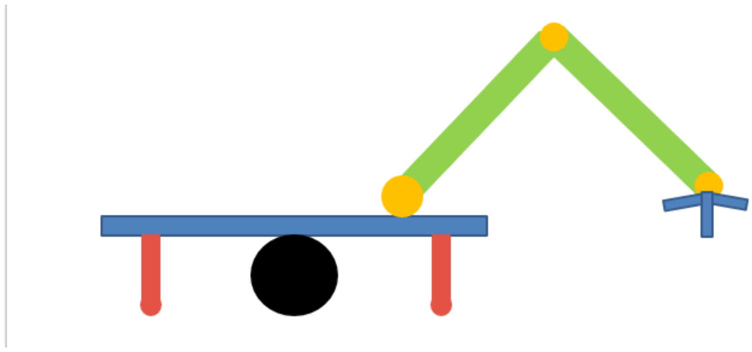


Figure 14: Sketch of the device

We can divide the whole system into three different sections, which from left to right are the platform, the lower arm and the upper arm. The black part are two rubber wheels and the red parts are two universal wheels. The yellow joints shown in the free body diagram are servo motors, with the maximum bearing torque to be 25 kg × cm, 20 kg × cm, 20 kg × cm respectively.

We then focus on the free body diagram of both two arms. The free body diagram of both two arms are shown in the below picture and the corresponding tested mass and arm length are also labelled in Figure 15.

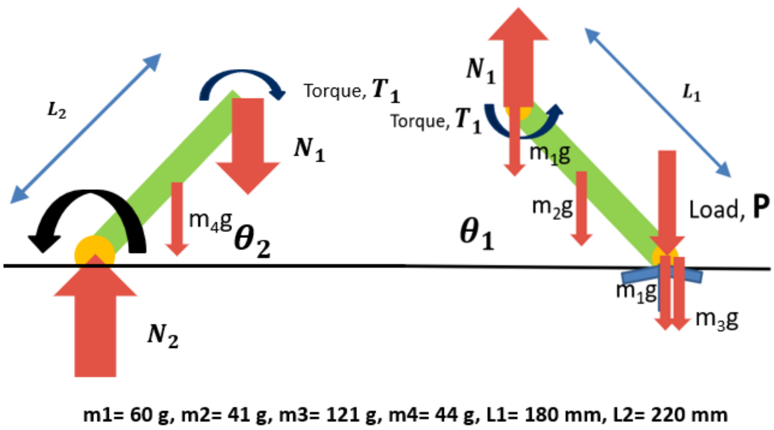


Figure 15: Free Body Diagram

Denotations in Figure 15 have the meaning listed below.

Denotation	Meaning
m_1	Mass of servo motor
m_2	Mass of the upper arm
m_3	Mass of the gripper
m_4	Mass of the lower arm
L_1	Length of the upper arm
L_2	Length of the lower arm

We then draw the relation between the safety factor and the load to ensure that our design is safe and will not face a fracture failure.

To find the stress of the acrylic board, we solve for the following equations.

$$M = \frac{1}{2} m_4 l_2 g cos\theta_2 + m_1 g(2l_2 \cos\theta_2 + l_1 \cos\theta_1) + m_2 g(l_2 \cos\theta_2 + \frac{1}{2} l_1 \cos\theta_1) + (P + m_3 g)(l_2 \cos\theta_2 + l_1 \cos\theta_1)$$

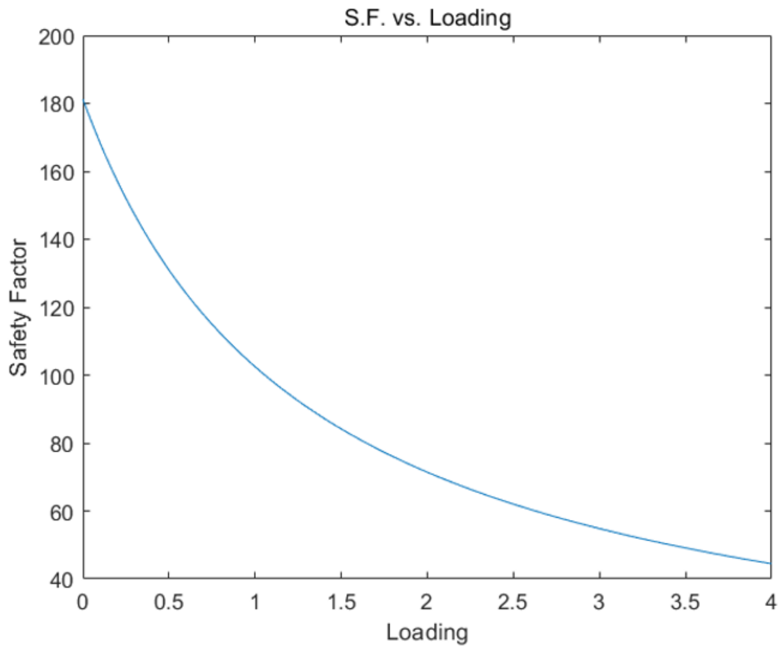
$$\sigma = \frac{Mc}{2I}$$

There are three unknown variables in the equation and it is not easy for understanding to plot a 4-dimensional view in matlab, so we consider the extreme case: Let both the angles θ_1, θ_2 to be zero. This is feasible because in this case we measure the maximum torque to get the maximum stress. Now we simplify the equation to be:

$$M = \frac{1}{2} m_4 l_2 g + m_1 g(2l_2 + l_1) + m_2 g(l_2 + \frac{1}{2} l_1) + (P + m_3 g)(l_2 + l_1)$$

We then plot the relation between the safety factor and the loading. The safety factor can be measured as:

$$S.F. = \frac{\sigma_{material}}{\sigma_{real}}$$



The design is safe if S.F. is larger than 1, and the higher the safety factor, the less likely our machine will suffer a fracture failure. In the graph we consider the maximum loading in the game to be around 2-3 N (tennis ball) and we can find the safety factor is far larger than 1.

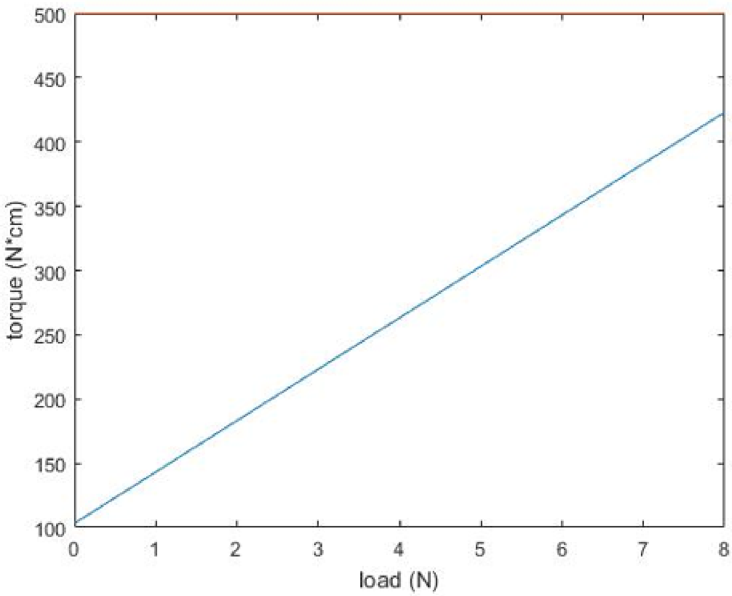
We then calculate the required torque for servo motors to find whether the provided torque is sufficient.

We still consider the extreme case when both the angles are zero.

The torque needed for the joint between the platform and the lower arm is calculated as:

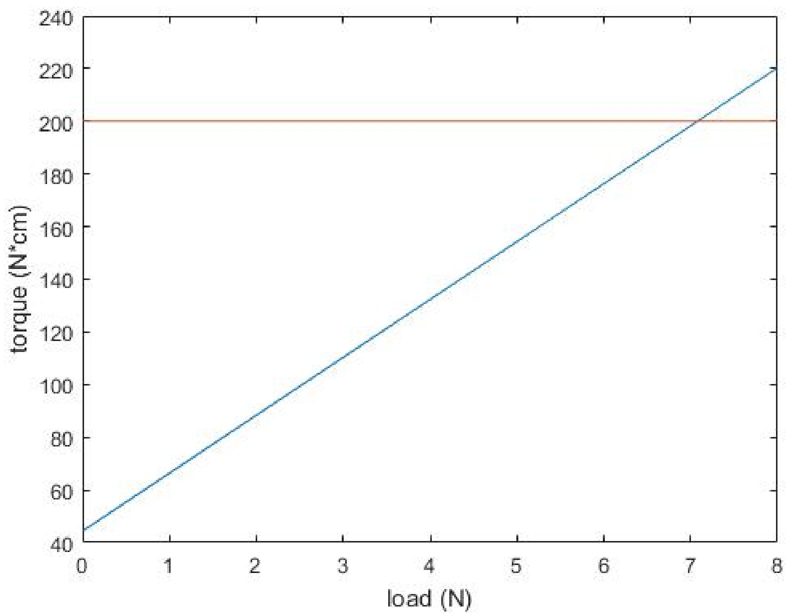
$$\tau_1 = \frac{1}{2}m_4 l_2 g + m_1 g(2l_2 + l_1) + m_2 g(l_2 + \frac{1}{2}l_1) + (P + m_3 g)(l_2 + l_1) = 103.15 + 40P$$

The torque at this part is provided by two servos of total torque to be 500N·cm. We plot a horizontal line to show this value:



Compared with the torque the servo can provide, we find that the servo torque is sufficient. The servo at the joint between the upper arm and the lower arm is

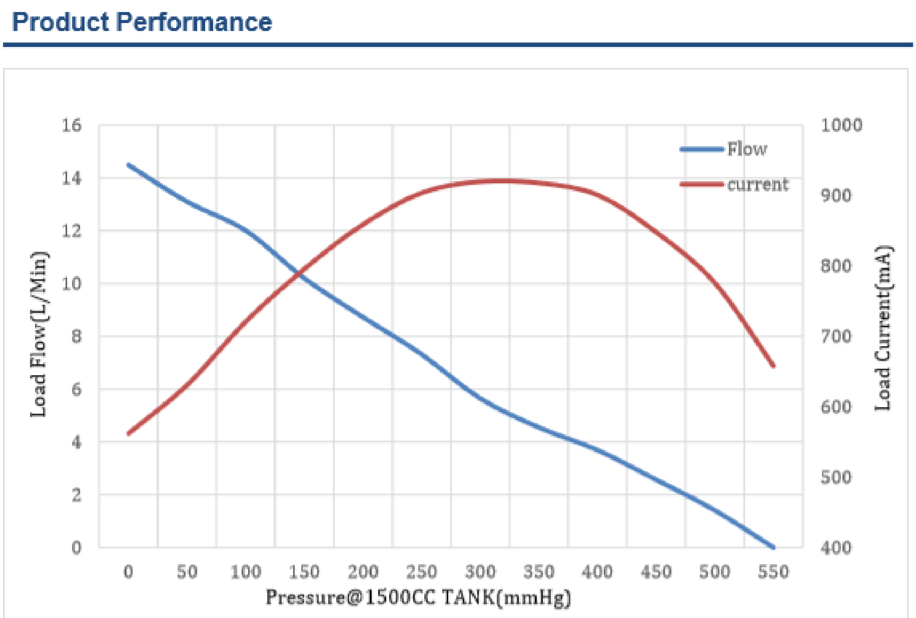
$$\tau_2 = \frac{1}{2}m_2 g l_1 + (P + m_3 g + m_1 g)l_1 = 44.33 + 22P$$



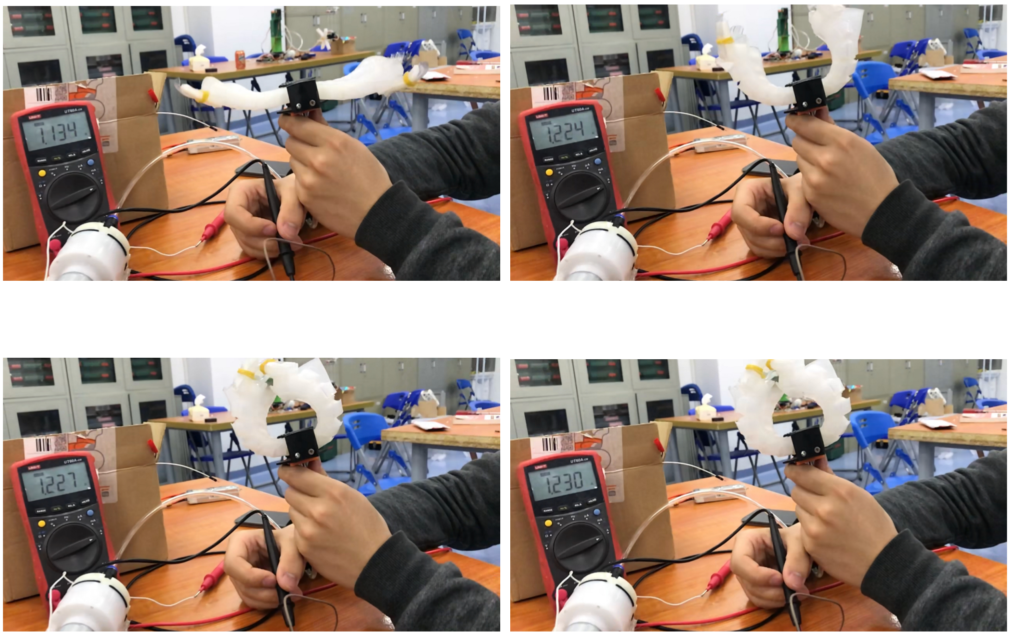
From the figure we know within the range of 3N, our servo can provide sufficient torque.

5.2 Pneumatic Gripper Test

From the performance curve the air pump works under 12V voltage and maximum working current is 1A.



In the lab we fail to get any air-pressure testing machine with a proper measurement range. So we compare the working current with gripper geometry.



From the testing graph we can see the gripper can inflate when the pump is on. And the pressure is enough for the gripper to fully inflate before the current begins to drop. However, we still find the performance curve is contradicted to the testing analysis in term of working current. In the performance curve the current is always smaller than 1A, but in our test the pump works with 1.23A current when the voltage reaches the wording voltage 12V.

6 Demonstration

Our soft-robot gripper can grip things (0-5N) tightly with diverse shape and size. Here are six screen shots from our validation video showing we can take:

1. small sphere (pingpang ball)
2. large sphere (tennis ball)
3. moderate size solid (correction tape)
4. small size pen
5. rotate a plastic bottle

We also use suctorial cup to grab the CD and rubber.

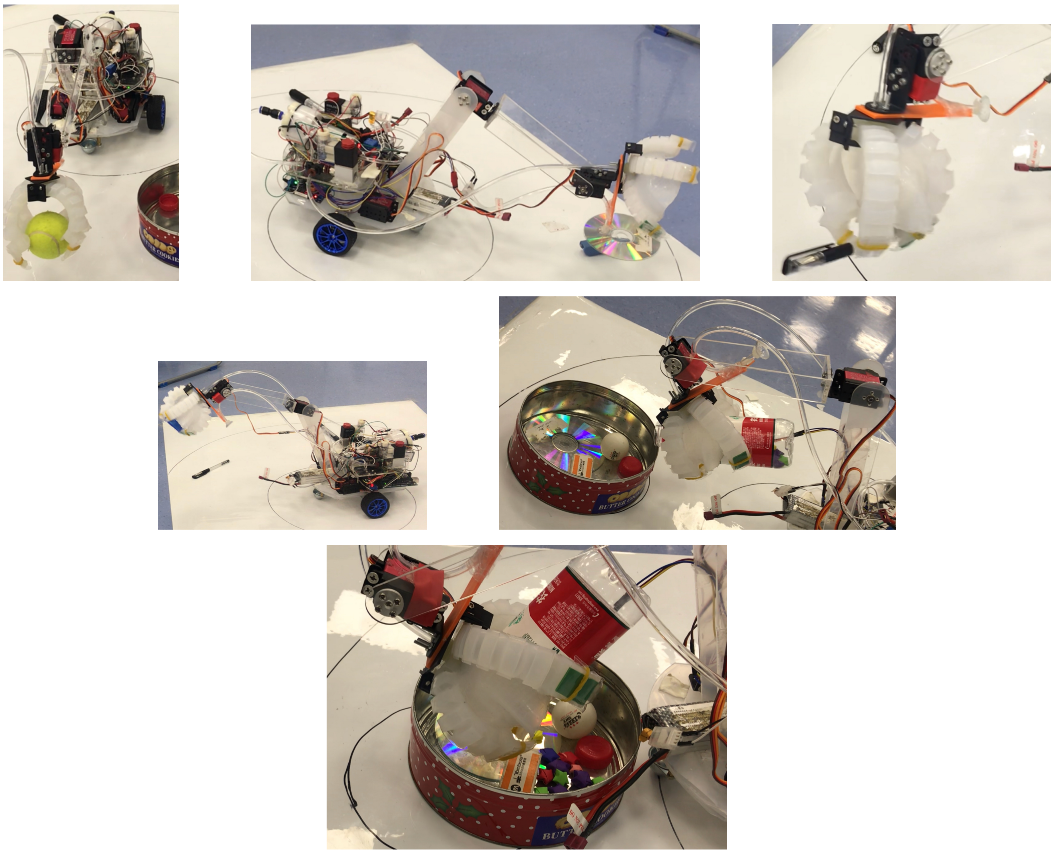


Figure 16: Grip all kinds of objects

For further demonstration, you can check our validation video which is added in the APPENDIX part.

7 Control and Circuit

7.1 Flow Chart of the Arduino Code

Figure 17 shows an overview of the whole program. The program is mainly divided into three parts.

First, we define some parameters before the setup. For example, we define pins, operations, objects and so on in the program. Therefore, it can be convenient for later on test and refinement. We needn't adjust parameters one by one.

Then, we build the setup, which prepares for the servos, motors and air pump.

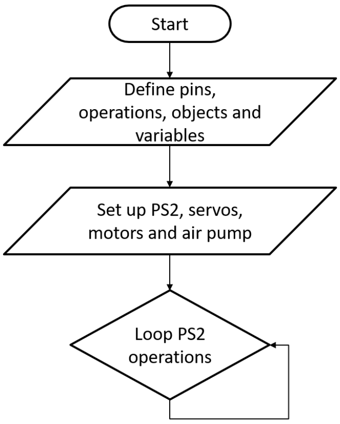


Figure 17: Overview of the whole program

At last, we build the loop. The detailed flowchart for loop is shown in Figure 18. The program will ask for the condition round by round. If the condition is true, specific operations, such as turning on the air pump will be conducted.

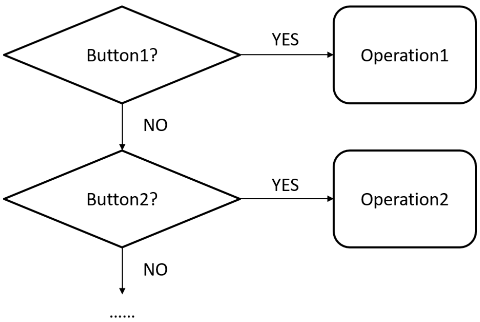


Figure 18: Detailed flowchart for loop

7.2 Circuit Diagram

Our circuit is as shown in Figure 19. With our Arduino code, you can build the whole circuit and run it. Although there are many components, our circuit is very ordered and flexible to adjust.

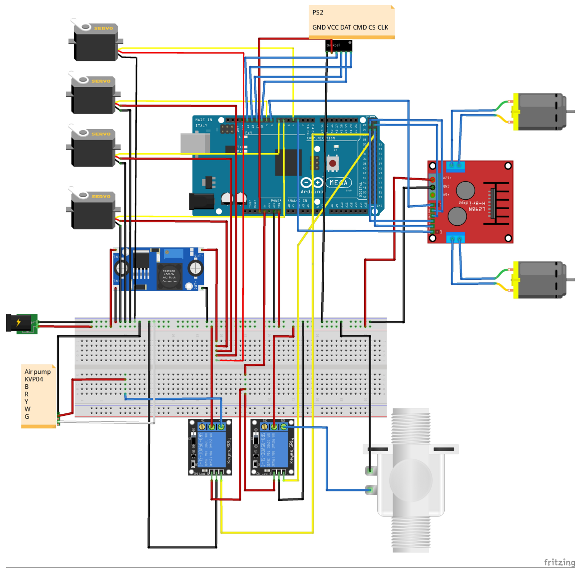


Figure 19: Circuit

8 Further Discussion

8.1 Discussion on test result

- Performance of air pump
  The performance curve for air pump implies that the current is always below 1000 mA. However, according to the video attached, it is obvious that the current is over 1000 mA. In addition, when we ran the air pump to a constant voltage source with 12V, the current is about 1300 mA. We think the performance curve may be not correct.

- Performance of servos
  The parameters of servos on the base is 25 kg·cm. With two servos, the device should be even able to hold object with 2 kg at the edge. Similarly, we can come to the conclusion that the other two servos are sufficient to carry all objects. However, the actual situation is that servos will vibrate. There are two possible reasons for it. First, the quality of servos is suspected. Second, there are four servos in the circuit, along with other components. Therefore, the total current can excess the capacity of the battery and Arduino, which results in unstable servos.

- Performance of motors
  We use two motors to rotate the platform, and we expect it to rotate in a circle. However, the actual movement is not a perfect circle. In fact, the device will move outward due to not absolutely same angular velocity of both sides.

8.2 Discussion on our work

- Excellent time-management
  During the project, we perform very well in time management. We almost

finish the project just before the Design Review. Due to our early test and refinement, we firstly found the problem of old air pump and reported it to Otto Zhang, which resulted in the replacement of the air pump.

- Brave and simple design
  We don't choose the traditional design for the mechanical structure, but choose a simpler structure, which can be seen as a trial to find a new way to solve the problem.

- Flexible and creative gripper
  As shown in the video, we can actually grasp almost every object, which means that it is flexible to objects with different shapes and size. In addition, our gripper is very creative, with tough surface to increase the friction, oblique fingers to grasp the bottle easily, uneven cavity size to increase curvature, so-called "fingernail" to grasp tiny things and so-called "web" to grasp bottle tightly.

8.3 Discussion on implications

- Design of mechanical structure
  Compared with other groups' design, our design, using rotational motion to drive the gripper, is simpler to develop. Besides, it is simpler to integrate this kind of structure with other kinds of design, because robot arms can be a very common design.
  However, to use this kind of design more smoothly, the good performance of program is required. For example, to prevent the operator adjusting three servos at the same time and getting nonlinear output, it is necessary to develop a kind of operation that enables the gripper to move in the horizontal direction and vertical direction. Otherwise, the operator will lead to being stuck in complex operation.

- Design of soft gripper
  To design of useful soft gripper, researchers and engineers must pay a lot of attention to theoretical analysis before the manufacturing. There are several reasons to do so. First, the influence of parameters of soft gripper can be abstract and indirect. Second, the performance of soft gripper is not as good as most people expected, if there is no careful and analysis and consideration. Third, the performance of soft gripper is sensitive to parameters. At last, the manufacturing process of soft gripper is tedious and time consuming. Without the general idea of design and just following the previous path without thinking, students can fall into the result that they pay a lot of efforts but earn little.
  In addition, the criterion for whether a gripper is feasible is not whether it is **can** be pumped by the **inflator**, which commonly used by students. In fact, a more practical way to judge it is to see **how curved** it can be when it is connected to the **air pump**. Because the highest pressure in the air pump is limited.
  When it comes to the design for this kind of "finger" soft gripper, we find some ways to increase the curvature at a given pressure.

  - At a given length, increase the number of cavity.
  - The size of cavity close to the center should be lower, while that away from the center should be larger.

- Design of circuit
  It is very easy for students to neglect the capacity of battery and Arduino. In another word, when design the circuit, it is very necessary to take the maximum current in the circuit into consideration. Otherwise, the circuit will vanish or the performance of the device will not reach the expected level.

- Implementation of whole project
  As Professor Ju said in the beginning of the course, VM250 is a course of "integration", which precisely describes the key to complete the project. Sometimes, every part of the device can work well, but it can't perform well after the integration.
  Besides, refinement and test will take much long time than expected. Therefore, it is very important for project manager to start earlier.

9 Conclusion

9.1 Objectives

- Design and manufacture a robotic arm with a soft gripper to grasp objects
- Move from one place to the other using remote control

9.2 Methods

- Use motors to rotate the platform
- Use servos to rotate the robot arm to specific location
- Use soft gripper with oblique fingers, uneven cavity size, tough surface, ”fingernail” and ”web” to grasp objects with different shapes tightly

9.3 Findings

- Our device perform well on grasping objects with different shapes
- The operation of our device is needed to be improved
- The performance curve of air pump may be wrong
- The current capacity of the circuit is a significant factor to the performance of the device
- Using motors to drive the platform can cause the problem of driving away from the center
- Using rotational motion to drive the robot arm can be convenient to build, but requires more on the control program and the ability of operator
- Design of soft gripper is the key for the project, and theoretical analysis is vital
- It is necessary to leave sufficient time to refine and test the project

9.4 Generalized ideas

- Take the constraints of hardware into consideration
- Think about the actual situation but not the ideal situation
- Analyze theoretically first rather than practice in hurry
- Analyze the key point in one project before doing it
- Start earlier

10 Reference

1.Eric Browna,1, Nicholas Rodenberga, John Amendb, Annan Mozeikac, Erik Steltzc, Mitchell R. Zakind,Hod Lipsonb, and Heinrich M. Jaegera"Universal robotic gripper based on the jamming of granular material",2010

2.Bobak Mosadegh , Panagiotis Polygerinos , Christoph Keplinger , Sophia Wennstedt , Robert F. Shepherd , Unmukt Gupta , Jongmin Shim , Katia Bertoldi , Conor J. Walsh , and George M. Whitesides,"Pneumatic Networks for Soft Robotics that Actuate Rapidly",2013

3.Ramses V. Martinez , Jamie L. Branch , Carina R. Fish , Lihua Jin , Robert F. Shepherd , Rui M. D. Nunes , Zhigang Suo , and George M. Whitesides,"Robotic Tentacles with Three-Dimensional Mobility Based on Flexible Elastomers",2013

11 Appendix

11.1 Project Contribution

Figure 20: Group Member (from left to right: Yixuan Wang, Tianle Liu, Yibo Chen, Jiwen Chen)

Yixuan Wang is responsible for the mechanical part of the robot arm.
Tianle Liu is responsible for the coding part of the robot arm.
Yibo Chen is responsible for the mechanical part of the soft gripper.
Jiwen Chen is responsible for the coding part of the soft gripper.

11.2 Gantt chart

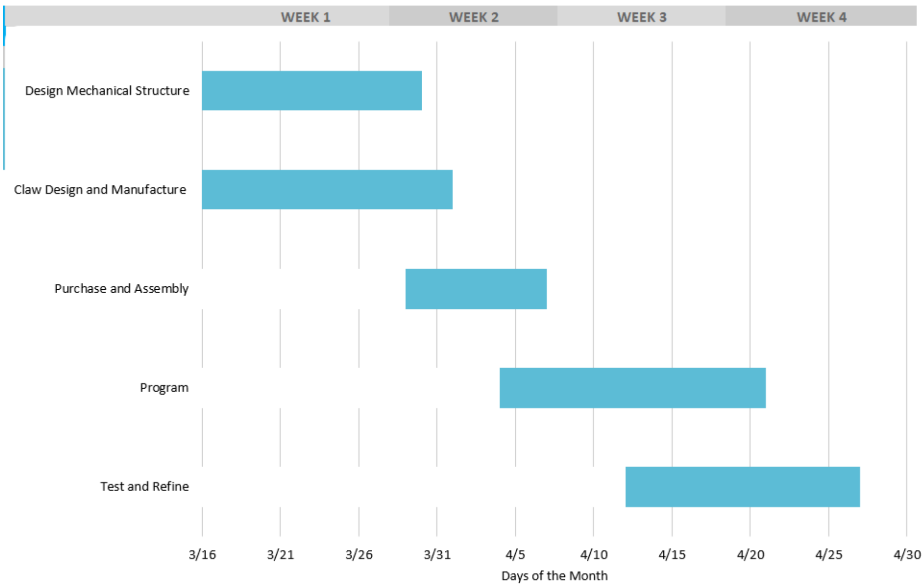


Figure 21: Gantt chart

11.3 Budget table

11.4 Video Link

There are several videos in the folder.

- "air pump": it shows how our air pump and soft gripper works

item	description	price	number	total
coupler	connect wheels and motors	3.9	2	7.8
wheels	Phi 65mm	12	2	24
universal wheels	15A	2.1	4	8.4
20kg servos	/	78	2	156
25kg servos	/	88	2	176
servo brackets	/	2.8	3	8.4
L shape servo brackets	/	2.3	2	4.6
servo bearings	/	2.5	3	7.5
motors with brackets	12V	30	2	60
motor driving board	L298N	10.94	1	10.94
wires	for power	10	many	10
copper column	support	16.7	many	16.7
Acrylic boards	customized	6.67	6	40
nuts and screws	M4/M3	15	many	15
battery	1800 mA 20 C 11.1V	89	1	89
Arduino mega	control chip	49	1	49
Buck module	LM2596S DC-DC	3.54	1	19
air pump	provided by the instructor	0	1	0
solenoid valve	1 given by instructor, 1 bought	47.58	1	47.58
total				749.92

- ”bottle”: it shows how our device grasp the bottle
- ”pen”: it shows how our device grasp the pen
- ”stapler”: it shows how our device grasp the stapler
- ”whole process”: it shows the whole process of the movement of our device.

11.5 Arduino Programming Code

```
#include <PS2X_lib.h>
#include <Servo.h>

#define PS2_DAT      13
#define PS2_CMD      11
#define PS2_SEL      10
#define PS2_CLK      12
#define servo1Pin    9
#define servo2Pin    7
#define servo3Pin    6
#define servo4Pin    4
#define enA          8
#define enB          3
#define in1          22
#define in2          23
#define in3          24
#define in4          25

#define pumpRelay    26
#define valveRelay   27
//#define AirPWM       4  delete and substituted by pumpRelay
//#define pinRelay     5  delete and substituted by valveRelay

#define pressures false
#define rumble    false

//Following part is to define the operation

#define servo1Up   PSB_TRIANGLE
#define servo1Down PSB_CROSS
#define servo2Up   PSB_CIRCLE
#define servo2Down PSB_SQUARE
#define servo3Up   PSB_R1
#define servo3Down PSB_R2
#define motorCCW   PSB_PAD_LEFT
#define motorCW    PSB_PAD_RIGHT
#define fp1        PSB_L1// the corresponding function is based on
   specific mechanical configration, which is shown in the directory
#define fp2        PSB_L2
#define forward    PSB_PAD_UP
#define back       PSB_PAD_DOWN
#define airPumpOn PSB_L3
#define airPumpOff PSB_R3

//Following part is to determine the parameter of the system

#define servo1MaxPos  170
#define servo1MinPos  110
#define servo2MaxPos  180
```

```
#define servo2MinPos  0
#define servo3MaxPos  180
#define servo3MinPos  0
#define l1            22// this is the length of lower robot arm
#define l2            22// this is the length of upper robot arm
#define WorkingV      25// Input voltage is 0.5V
#define StopingV      200// Used to stop the air pump

PS2X ps2x; // create PS2 Controller Class
Servo servo1; // create servo class
Servo servo2;
Servo servo3;
Servo servo4;

int error = 0;
byte type = 0;
byte vibrate = 0;
int servoDelay=15;
int flag=0;//define the situation of the air pump.


double pos1 = 90; // this is to store the position of servo
double pos2 = 90;
double pos3 = 90;
double pos4 = 90;
double fp1pos1 = 110;
double fp1pos2 = 100;
double fp1pos3 = 80;
double fp2pos1 = 110;
double fp2pos2 = 120;
double fp2pos3 = 180;
double sum14 = 185;

void (* resetFunc) (void) = 0;

void setup(){

  Serial.begin(115200);

  delay(500); //added delay to give wireless ps2 module some time to
      startup, before configuring it

  //CHANGES for v1.6 HERE!!! **************PAY ATTENTION*************

  //setup pins and settings: GamePad(clock, command, attention, data,
      Pressures?, Rumble?) check for error
  error = ps2x.config_gamepad(PS2_CLK, PS2_CMD, PS2_SEL, PS2_DAT,
      pressures, rumble);

  if(error == 0){
    Serial.print("Found Controller, configured successful ");
```

```
  Serial.print("pressures = ");
if (pressures)
  Serial.println("true ");
else
  Serial.println("false");
Serial.print("rumble = ");
if (rumble)
  Serial.println("true)");
else
  Serial.println("false");
  Serial.println("Try out all the buttons, X will vibrate the controller, faster as you press harder;");
  Serial.println("holding L1 or R1 will print out the analog stick values.");
  Serial.println("Note: Go to www.billporter.info for updates and to report bugs.");
}
else if(error == 1)
  Serial.println("No controller found, check wiring, see readme.txt to enable debug. visit www.billporter.info for troubleshooting tips");

else if(error == 2)
  Serial.println("Controller found but not accepting commands. see readme.txt to enable debug. Visit www.billporter.info for troubleshooting tips");

else if(error == 3)
  Serial.println("Controller refusing to enter Pressures mode, may not support it. ");

type = ps2x.readType();
switch(type) {
  case 0:
    Serial.println("Unknown Controller type found ");
    break;
  case 1:
    Serial.println("DualShock Controller found ");
    break;
  case 2:
    Serial.println("GuitarHero Controller found ");
    break;
case 3:
    Serial.println("Wireless Sony DualShock Controller found ");
    break;
 }

// Following part is to set up the servo
servo1.attach(servo1Pin);
servo2.attach(servo2Pin);
servo3.attach(servo3Pin);
```

```
  servo4.attach(servo4Pin);

  // Following part is to set up the motor
  pinMode(enA, OUTPUT);
  pinMode(in1, OUTPUT);
  pinMode(in2, OUTPUT);
  pinMode(enB, OUTPUT);
  pinMode(in3, OUTPUT);
  pinMode(in4, OUTPUT);
  digitalWrite(in1, LOW);
  digitalWrite(in2, LOW);
  digitalWrite(in3, LOW);
  digitalWrite(in4, LOW);
  pinMode(pumpRelay,OUTPUT);
  pinMode(valveRelay,OUTPUT);
}

void loop() {
  analogWrite(enA, 140);
  analogWrite(enB, 140);
  if(error == 1){ //skip loop if no controller found
    resetFunc();
  }

  if(type == 2){ //Guitar Hero Controller
    ps2x.read_gamepad();        //read controller

    if(ps2x.ButtonPressed(GREEN_FRET))
      Serial.println("Green Fret Pressed");
    if(ps2x.ButtonPressed(RED_FRET))
      Serial.println("Red Fret Pressed");
    if(ps2x.ButtonPressed(YELLOW_FRET))
      Serial.println("Yellow Fret Pressed");
    if(ps2x.ButtonPressed(BLUE_FRET))
      Serial.println("Blue Fret Pressed");
    if(ps2x.ButtonPressed(ORANGE_FRET))
      Serial.println("Orange Fret Pressed");

    if(ps2x.ButtonPressed(STAR_POWER))
      Serial.println("Star Power Command");

    if(ps2x.Button(UP_STRUM))        //will be TRUE as long as button is
       pressed
      Serial.println("Up Strum");
    if(ps2x.Button(DOWN_STRUM))
      Serial.println("DOWN Strum");

    if(ps2x.Button(PSB_START))       //will be TRUE as long as button is
       pressed
      Serial.println("Start is being held");
    if(ps2x.Button(PSB_SELECT))
```

```
    Serial.println("Select is being held");

  if(ps2x.Button(ORANGE_FRET)) {  // print stick value IF TRUE
    Serial.print("Wammy Bar Position:");
    Serial.println(ps2x.Analog(WHAMMY_BAR), DEC);
  }
}
else { //DualShock Controller
  ps2x.read_gamepad(false, vibrate); //read controller and set large
      motor to spin at 'vibrate' speed
  digitalWrite(in2, LOW);
  digitalWrite(in1, LOW);
  digitalWrite(in3, LOW);
  digitalWrite(in4, LOW);

  if(ps2x.Button(servo1Up) && pos1+1<= servo1MaxPos){
    Serial.println("servo1 rotates up");
    pos1=pos1+1;
    servo1.write(pos1);
    pos4 = sum14 - pos1;
    servo4.write(pos4);
    delay(servoDelay);
  }

  if(ps2x.Button(servo1Down) && pos1-1 >= servo1MinPos){
    Serial.println("servo1 rotates down");
    pos1=pos1-1;
    servo1.write(pos1);
    pos4 = sum14 - pos1;
    servo4.write(pos4);
    delay(servoDelay);
  }

  if(ps2x.Button(servo2Up) && pos2+1<= servo2MaxPos){
    Serial.println("servo2 rotates up");
    pos2=pos2+1;
    servo2.write(pos2);
    delay(servoDelay);
  }

  if(ps2x.Button(servo2Down) && pos2-1 >= servo2MinPos){
    Serial.println("servo2 rotates down");
    pos2=pos2-1;
    servo2.write(pos2);
    delay(servoDelay);
  }

  if(ps2x.Button(servo3Up) && pos3+1<= servo3MaxPos){
    Serial.println("servo3 rotates up");
    pos3=pos3+1;
    servo3.write(pos3);
```

```
  delay(servoDelay);
}

if(ps2x.Button(servo3Down) && pos3-1 >= servo3MinPos){
  Serial.println("servo3 rotates down");
  pos3=pos3-1;
  servo3.write(pos3);
  delay(servoDelay);
}

if(ps2x.ButtonPressed(fp1)){
  Serial.println("To fixed point 1");
  pos1 = fp1pos1;
  pos4 = sum14 - pos1;
  pos2 = fp1pos2;
  pos3 = fp1pos3;
  servo1.write(pos1);
  servo2.write(pos2);
  servo3.write(pos3);
  servo4.write(pos4);
}

if(ps2x.ButtonPressed(fp2)){
  Serial.println("To fixed point 2");
  pos1 = fp2pos1;
  pos4 = sum14 - pos1;
  pos2 = fp2pos2;
  pos3 = fp2pos3;
  servo1.write(pos1);
  servo2.write(pos2);
  servo3.write(pos3);
  servo4.write(pos4);
}

if(ps2x.Button(forward)){
  Serial.println("moves forward");
  digitalWrite(in1, LOW);
  digitalWrite(in2, LOW);
  digitalWrite(in4, HIGH);
  digitalWrite(in3, HIGH);
}

if(ps2x.Button(back)){
  Serial.println("moves backward");
  digitalWrite(in2, HIGH);
  digitalWrite(in1, HIGH);
  digitalWrite(in3, LOW);
  digitalWrite(in4, LOW);
}

if(ps2x.Button(motorCCW)){
```

```
    Serial.println("rotates CCW");
    digitalWrite(in1, HIGH);
    digitalWrite(in2, LOW);
    digitalWrite(in4, HIGH);
    digitalWrite(in3, LOW);
  }

  if(ps2x.Button(motorCW)){
    Serial.println("rotates CW");
    digitalWrite(in2, HIGH);
    digitalWrite(in1, LOW);
    digitalWrite(in3, HIGH);
    digitalWrite(in4, LOW);
  }

  if(flag==0){
    Serial.println("1111");
    digitalWrite(valveRelay,LOW);
    digitalWrite(pumpRelay,LOW);
  }

  if(ps2x.Button(airPumpOn)){
    Serial.println("air pump on");
    digitalWrite(valveRelay,LOW);
    digitalWrite(pumpRelay,HIGH);
    delay(100);
    flag=1;
  }

  if(ps2x.Button(airPumpOff)){
    Serial.println("air pump off, keep the inflation");
    digitalWrite(valveRelay,HIGH);
    digitalWrite(pumpRelay,LOW);
    delay(100);
    flag=1;
  }

  if(ps2x.ButtonReleased(airPumpOff)){
    flag=0;
  }

  if(ps2x.ButtonReleased(airPumpOn)){
  Serial.println("air pump off, inflation off");
  digitalWrite(pumpRelay,LOW);
  digitalWrite(valveRelay,LOW);
  delay(100);
  flag=0;
  }

}
delay(50);
```

```
}
```