**Prototype Report**

ME 2900

Instructor: Dr. Sandra Metzler

Date of Submission: 10/16/2017

Zhaoyi Jiang(.1364)

**I, Introduction**

The current industries demand robot heavily. To meet the requirement of low cost and certain accuracy, the 3D printed robot arm is made. The arm should be able to adjust to certain angle by press the button and by hand control. It should be accurate enough to grab a ping-pang ball.

(In fact, it is a gift for one of my friend who needs a robot arm to start his Machine Learning project)

**II, Idea Development Process**

Actually I did not really brainstorming about what project I will choose because my friend kept asking me about making him a robot arm for weeks. I automatically choose the robot arm as my project. I did brainstorming on how to design and make my arm work. I was wondering using Aluminums as my major material. However, considering about the accuracy and compatibility, servo is my primary choice for the joints of the arm, which means the weight should be reduced to minimum. Stepper motor is one option but the motors are too large and heavy. Besides, six degrees of freedom is required since it is the only solution to make it swift. Consequently, 3D printing is used for my arm. Solidworks is used to dram the sketches and models.

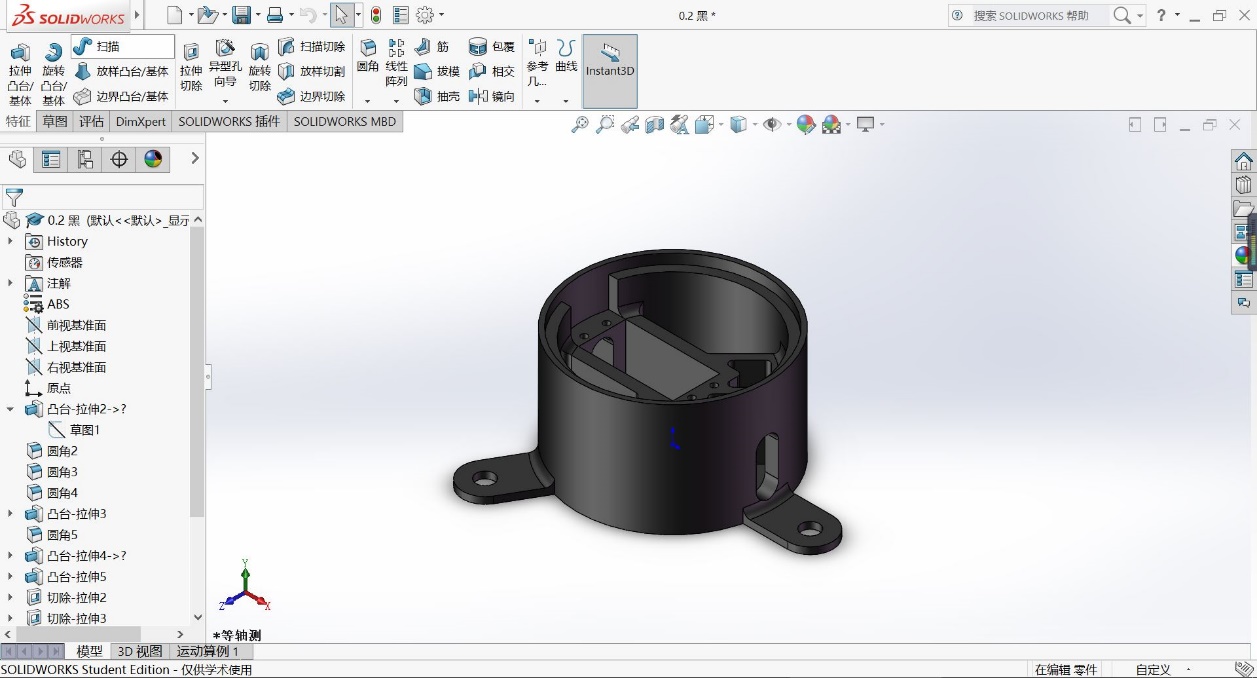


Figure 1：Solidworks

According to “the guy”, the error of the 3D printing is 0.2mm. I left some spaces for every part to ensure that the servos would fit into the sockets. I also left some holes for wires to come out.

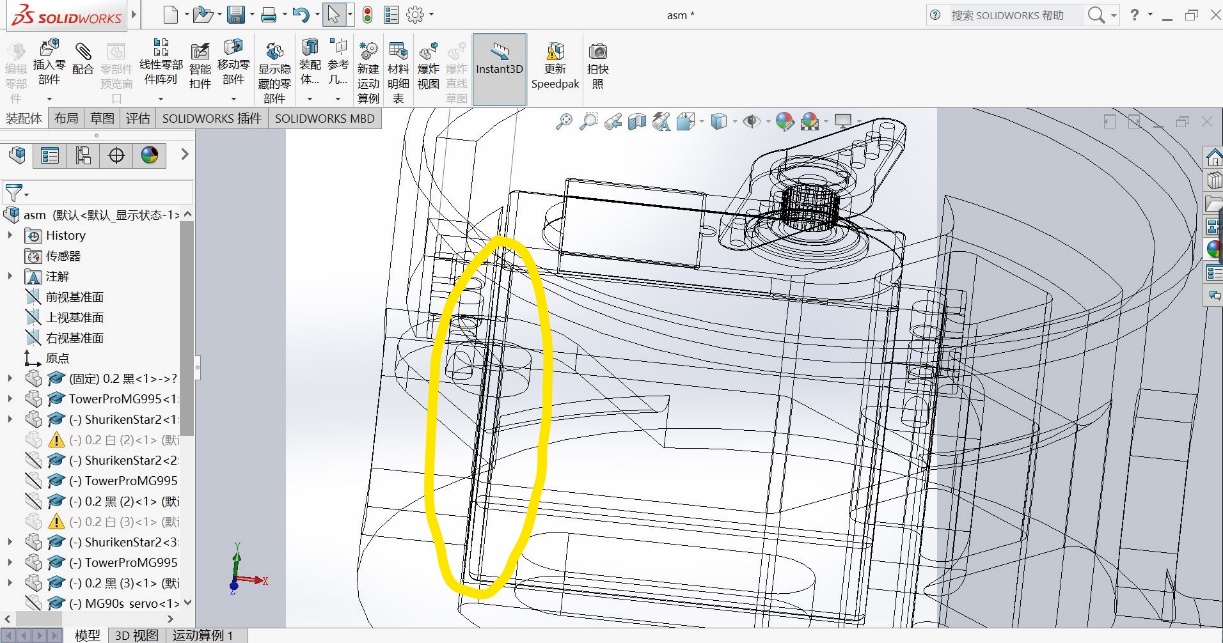


Figure 2: Gap between servo and socket

Due to the limited accuracy of 3D printing machine, hot glue is used to stick some parts, but most connectors are screws. To make sure that the servo could be installed into the sockets properly, I made a slot of every single part that connects to servo.

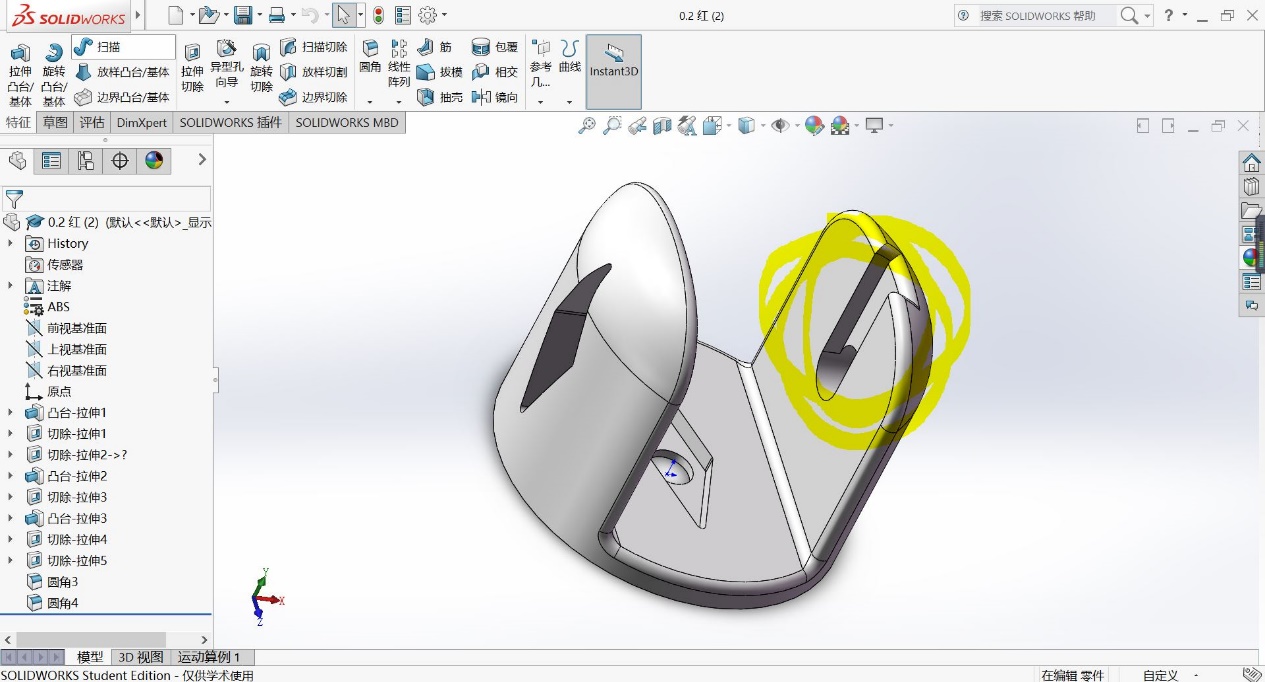
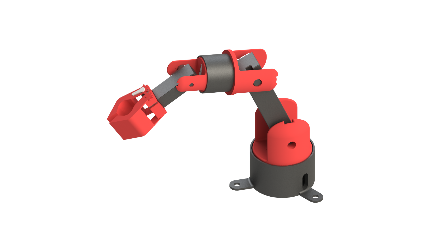
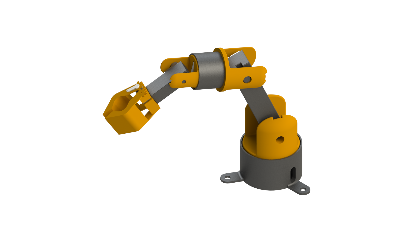
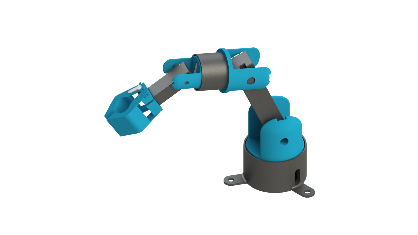


Figure 3: Slot for servo and beam

The robot arm is designed without any big problems, it is just the matter or time for me. Several color combos are listed below, and I finally chose the black-white combo.



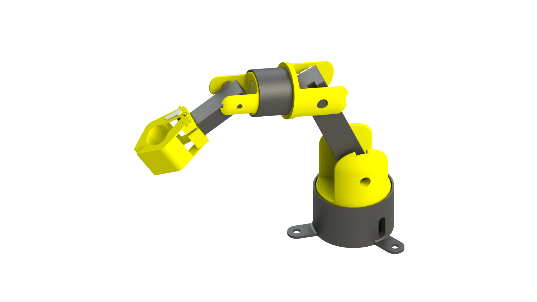
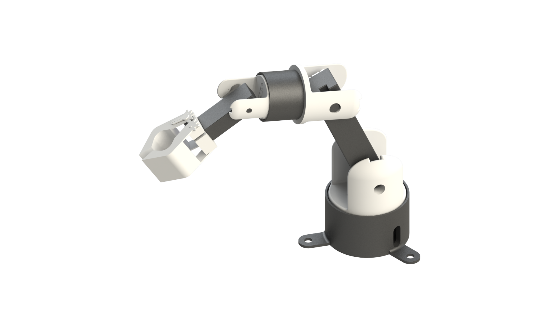


Figure 4: Color Combos

The use or potentiometer almost take part at first. However, the final decision is the joystick, because the potentiometer is too small to be handled and sometimes inaccurate. The joysticks are used to change the angles of servos by one degree every 10ms if they are moved. A for loop is enough. The initial thought of use of keypad is input a number and press “#” as “Enter”. However, that will only remain five press buttons to select servos. So I decided to use “A”,”B”,”C”,”D”,”#” and ”\*” to select the servo. Number one to nine will correspond to 0, 20, 40……180 degrees, which makes the program easier and more user-friendly. While writing the program of “Choose servo”, I is wondering about using bunch of Booleans and while loops to do it, but that will storage too many variables in the memory. I then further find out that the command of switchcase. By using that, the code is shortened and easier to understand.

**III, Discussion**

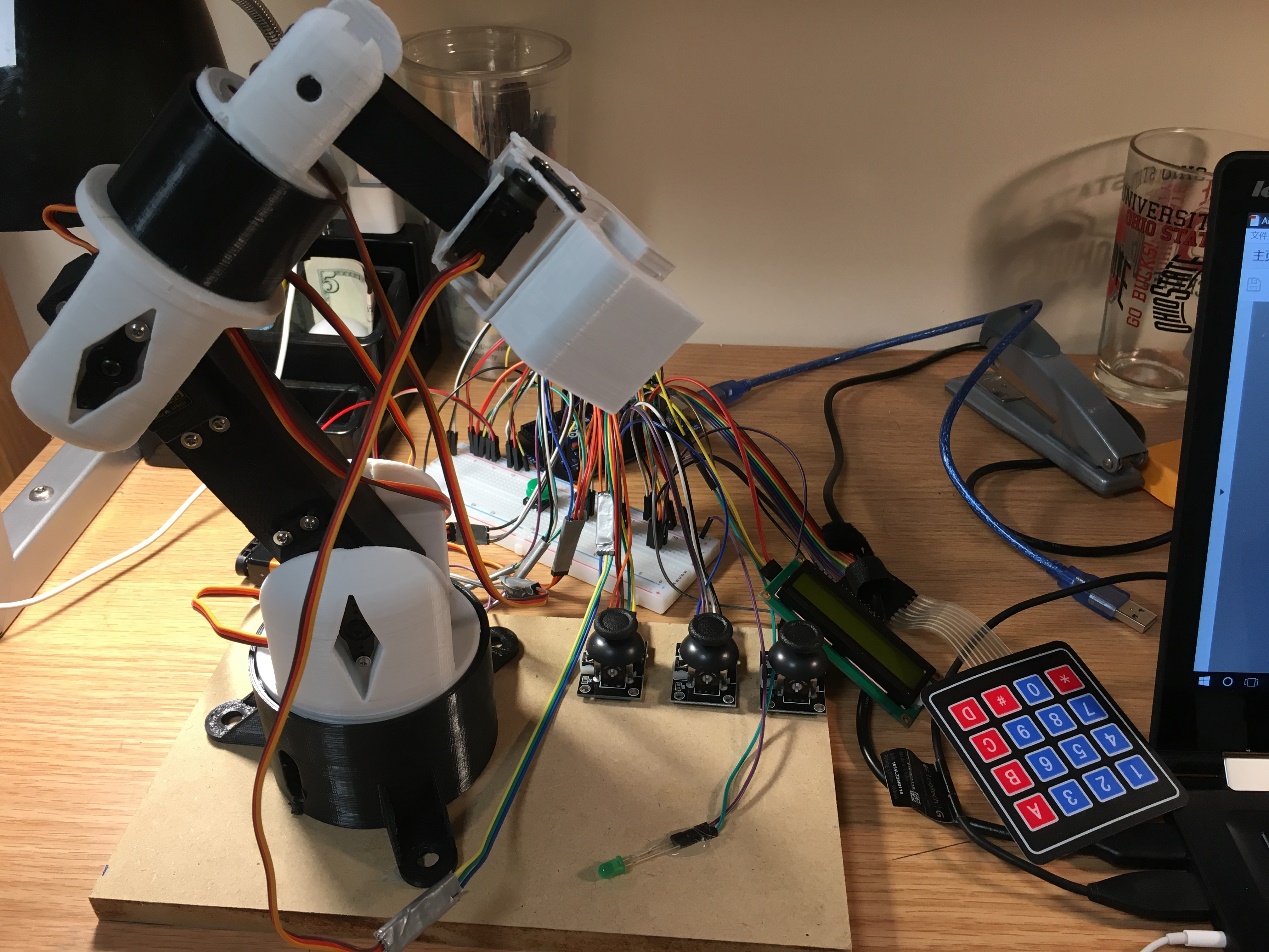
The prototype is simply a six degrees of freedom robot arm that is able to grab items. The robot arm is driven by six servos. The robot arm could be controlled by keypad and joysticks. By pressing “A”,”B”,”C”,”D”,”#” and ”\*” on the keypad, you first select the servo you want to control. The basic idea of code for this part is “Switchcase”. Then by pressing 0 to 9 on the keypad, you choose the angle of the selected servo. The name of the servo and angle you choose are displayed on the LCD. Each number refers to 20 degrees of increment. Using the “map” function, this part is easy to obtain. By keeping pressing the pressbutton, you can switch to hand control mode. When switch to hand control mode, the green LED will be on to notice you. I used an” if” to achieve it because I wrote two major parts of my code as two functions. By playing around with the joysticks. You can adjust the angle of servos. Every time you move the joystick, the angle increase or decrease by one. A “map” function is used to convert the analog value to value between -10 to 10. Then if the value is less than -2 or greater than 2, the change of angle will equal to the value divided by the absolute value of the value. So, the sign is kept.

**IIII, Conclusion**

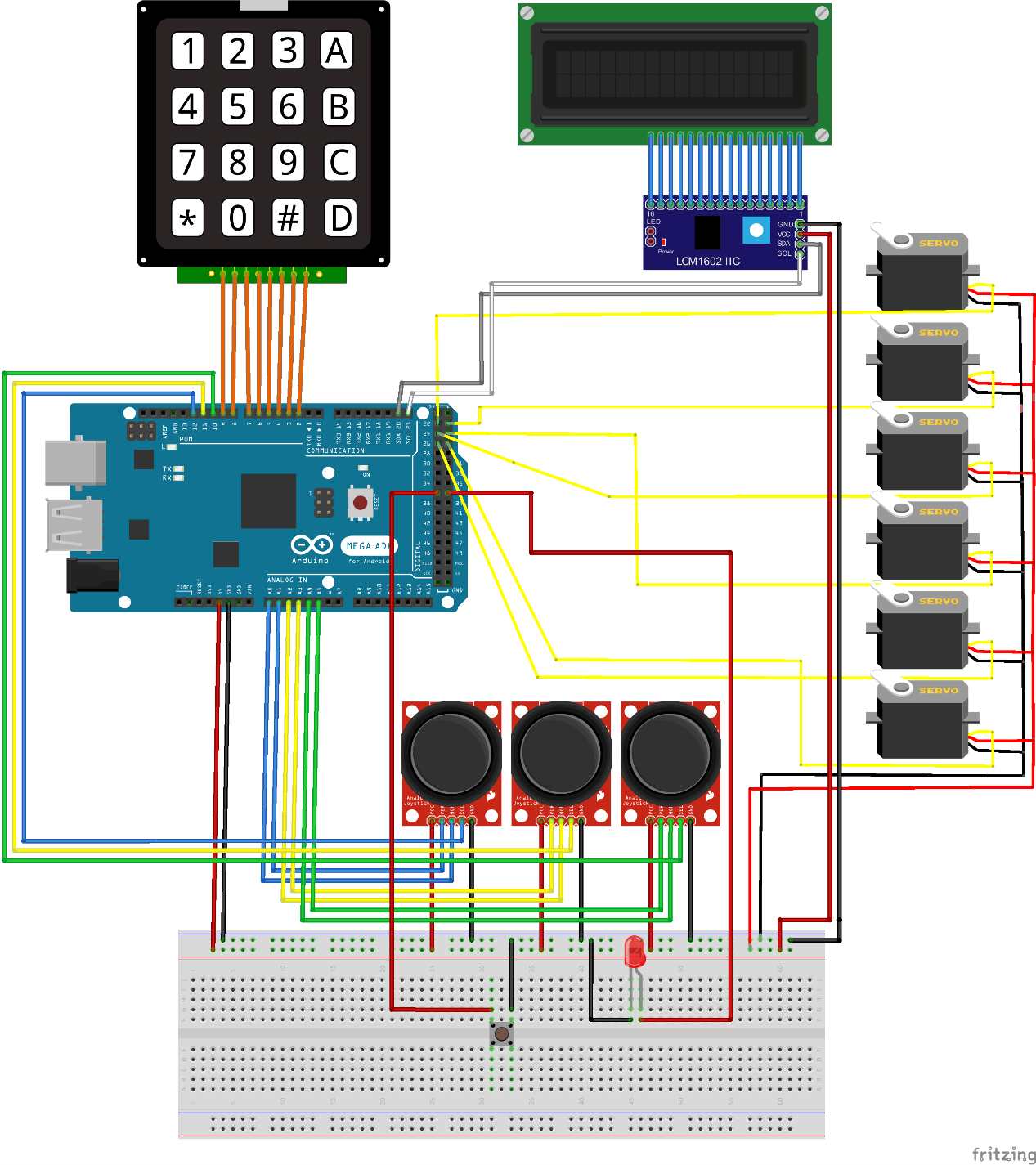
My most successful feature is my 3D printed parts. I designed it by solidworks and found a friend who can print those parts in super-low price. The 3D printed parts are stronger and lighter then I used to think they are. They fit each other almost perfectly, which means my design is successful. I also think that my design of the grab is exquisite. I draw the gears by myself and they work functionally. The difficulty is that I had some troubles with LCD display and switchcase command, because I have never used that before, plus the LCD is connected to a I2C bus. Google is the best tool in the world. I solve all of my troubles by searching online. I will work on the safety and accuracy in the future. I did not set a switch for my arm, which means as far as I connect the battery. The program will be automatically on without any hesitate. I will probably add more codes on that part. The 3D printed part is not that accurate because some gap between the parts and the servos’ rods. I will try to fix that in the future by redesign the parts.

**Appendices**

**Photo**



**Circuit**



**Code(Control):**

//////////////////////Setup initial values for joysticks/////////////////////////////////////////

int pressed = -1;

int x = -1;

int y = -1;

int w = -1;

int z = -1;

int m = -1;

int n = -1;

int angle = 90;

////////////////////////////setup LCD I2C bus//////////////////////////////////////

#include <Wire.h>

#include <LiquidCrystal\_I2C.h>

LiquidCrystal\_I2C lcd(0x3F, 16, 2);

//////////////////////////////Setup all servos///////////////////////////////////////

#include <Servo.h>

Servo base;

Servo BMain;

Servo TMain;

Servo TRotate;

Servo TMi;

Servo Grab;

//create an array to name the servo

String Snames[6] = {"Base", "Base Main", "Top Main", "Top Rotate", "Top Main", "Grab"};

/////////////////////////////setup the keypad/////////////////////////////////////////

#include <Keypad.h>

char keymap[4][4] =

{

{'1', '2', '3', 'A'},

{'4', '5', '6', 'B'},

{'7', '8', '9', 'C'},

{'\*', '0', '#', 'D'}

};

byte rPins[4] = {9, 8, 7, 6};

byte cPins[4] = {5, 4, 3, 2};

Keypad kpd = Keypad(makeKeymap(keymap), rPins, cPins, 4, 4);

/////////////////////////////////////////////////////////////////////////////////

////////////////////////////////////////////////////////////////////////////////////

void setup() {

pinMode(37,OUTPUT);

Serial.begin(9600);

//////////////////////setup the pin for a pushbutton////////////////////////////////////////////

pinMode(36, INPUT\_PULLUP);

////////////////////////////setup the pins for joystick/////////////////////////////////

pinMode(12, INPUT);

digitalWrite(12, HIGH);

pinMode(11, INPUT);

digitalWrite(11, HIGH);

pinMode(10, INPUT);

digitalWrite(10, HIGH);

////////////////////////////setup pins for servos/////////////////////////////

base.attach(22);

BMain.attach(23);

TMain.attach(24);

TRotate.attach(25);

TMi.attach(26);

Grab.attach(27);

base.write(90);

BMain.write(90);

TMain.write(80);

TRotate.write(150);

TMi.write(20);

Grab.write(40);

////////////////////////////////setup the initial display for LCD///////////////////////////////////

lcd.init();

lcd.init();

lcd.backlight();

lcd.setCursor(0, 0);

lcd.print("Angle");

lcd.setCursor(0, 1);

lcd.print("########");

}

//////////////////////////////////////////////////////////////

////////////////////////////////////////////////////////////////////////

void loop() {

if (digitalRead(36)==LOW){

joystick();//controled by joystick

digitalWrite(37,HIGH);

}

else{

motorControl(); //controled by keypad

digitalWrite(37,LOW);

}

}

////////////////////////////////////////////////////////////////////////

///////////////////////////////////////////////////////////////////////////

int getservonames(char input) { //Function to relate cases to servos' names

int index=0;

switch (input) {

case 'A' : {

index = 0;

break;

}

case 'B': {

index = 1;

break;

}

case'C': {

index = 2;

break;

}

case'D' : {

index = 3;

break;

}

case'#' : {

index = 4;

break;

}

case'\*': {

index = 5;

break;

}

}

return index;

}

void motorControl() { //Main part of keypad control

int id=0;

char forcase = kpd.getKey();

id = getservonames(forcase);

if (forcase != NO\_KEY)

{

Serial.println(id);

switch (forcase) {

case 'A': {

lcd.clear();

lcd.setCursor(0, 0);

lcd.print(Snames[id]);

int a = kpd.waitForKey() - 48;

int angle1 = map(a, 0, 9, 5, 179);

constrain(angle1, 5, 179);

base.write(angle1);

lcd.setCursor(0, 1);

lcd.print(angle1);

break;

}

case 'B': {

lcd.clear();

lcd.setCursor(0, 0);

lcd.print(Snames[id]);

int b = kpd.waitForKey() - 48;

int angle2 = map(b, 0, 9, 5, 179);

constrain(angle2, 5, 179);

BMain.write(angle2);

lcd.setCursor(0, 1);

lcd.print(angle2);

break;

}

case 'C': {

lcd.clear();

lcd.setCursor(0, 0);

lcd.print(Snames[id]);

int c = kpd.waitForKey() - 48;

int angle3 = map(c, 0, 9, 5, 179);

constrain(angle3, 5, 179);

TMain.write(angle3);

lcd.setCursor(0, 1);

lcd.print(angle3);

break;

}

case 'D': {

lcd.clear();

lcd.setCursor(0, 0);

lcd.print(Snames[id]);

int d = kpd.waitForKey() - 48;

int angle4 = map(d, 0, 9, 5, 179);

constrain(angle4, 5, 179);

TRotate.write(angle4);

lcd.setCursor(0, 1);

lcd.print(angle4);

break;

}

case '#': {

lcd.clear();

lcd.setCursor(0, 0);

lcd.print(Snames[id]);

int e = kpd.waitForKey() - 48;

int angle5 = map(e, 0, 9, 5, 179);

constrain(angle5, 5, 179);

TMi.write(angle5);

lcd.setCursor(0, 1);

lcd.print(angle5);

break;

}

case '\*': {

lcd.clear();

lcd.setCursor(0, 0);

lcd.print(Snames[id]);

int f = kpd.waitForKey() - 48;

int angle6 = map(f, 0, 9, 5, 179);

constrain(angle6, 5, 179);

Grab.write(angle6);

lcd.setCursor(0, 1);

lcd.print(angle6);

break;

}

}

}

}

void joyControl(Servo servoname, int le) { //Joystick control

if (le < -2 || le > 2) {

angle = angle + le/abs(le);

angle = constrain(angle, 1, 179);

servoname.write(angle);

delay(10);

}

}

void joystick() { //main part of joystick control

x = map(analogRead(0), 0, 1023, -10, 10);

y = map(analogRead(1), 0, 1023, -10, 10);

w = map(analogRead(2), 0, 1023, -10, 10);

z = map(analogRead(3), 0, 1023, -10, 10);

m = map(analogRead(4), 0, 1023, -10, 10);

n = map(analogRead(5), 0, 1023, -10, 10);

joyControl(base, x);

joyControl(BMain, y);

joyControl(TMain, w);

joyControl(TRotate, z);

joyControl(TMi, m);

joyControl(Grab, n);

}

**Code(PingPang ball & Test):**

#include <Servo.h>

Servo base;

Servo BMain;

Servo TMain;

Servo TRotate;

Servo TMi;

Servo Grab;

void setup() {

base.attach(22);

BMain.attach(23);

TMain.attach(24);

TRotate.attach(25);

TMi.attach(26);

Grab.attach(27);

base.write(90);

delay(300);

BMain.write(90);

delay(300);

TMain.write(90);

delay(300);

TRotate.write(150);

delay(300);

TMi.write(20);

delay(300);

Grab.write(10);

delay(3000);

for(int i=90;i>0;i--){

base.write(i);

delay(10);

}

for(int i=150;i>50;i--){

TRotate.write(i);

delay(10);

}

for(int i=90;i<135;i++){

BMain.write(i);

delay(10);

}

for(int i=80;i>30;i--){

TMain.write(i);

delay(10);

}

for(int i=20;i<40;i++){

TMi.write(i);

delay(10);

}

for(int i=10;i<35;i++){

Grab.write(i);

delay(10);

}

for(int i=135;i>90;i--){

BMain.write(i);

delay(10);

}

for(int i=40;i<75;i++){

TMi.write(i);

delay(10);

}

for(int i=0;i<80;i++){

base.write(i);

delay(5);

}

delay(1000);

for(int i=35;i>10;i--){

Grab.write(i);

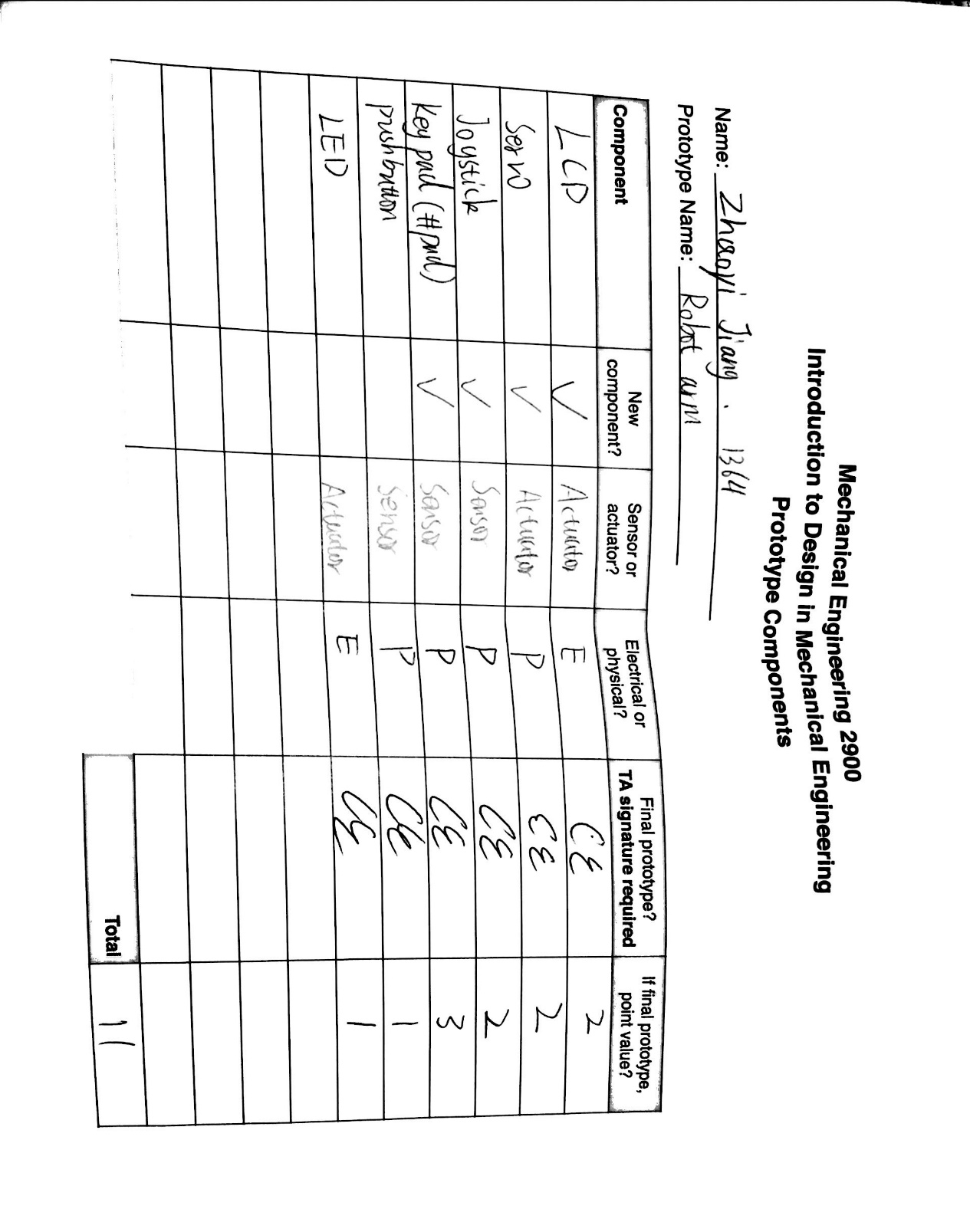
delay(10);

}

}

void loop(){}

**Checkout sheet:**

****

**Video link:**

<https://youtu.be/LaFlYgAL9kE>