

```

#include <Servo.h>

Servo base; //Servo object for the base servo
Servo shoulder; //Servo object for the shoulder servo
Servo elbow; //Servo object for the elbow servo

int base_current = 93; //Initial vlue for base_current position
int shoulder_current = 90; //Initial vlue for shoulder_current
position
int elbow_current = 125; //Initial vlue for elbow_current
position
int serial_num = 0; //Initial value for which serial event is
happening
int angles[3] = {93,90,125}; //Initial values for the future
angles
int magnet = 3; //Pin for the electromagnet output
int side = 0; //Initial value for which side to drop (0 =
triangle/right, 1 = square/left)
bool ready = false; //Initial state for if the serial has read
enough inputs

void setup() {
  Serial.begin(115200);
  base.attach(9); //Attach the base servo to pin 9
  shoulder.attach(10); //Attach the shoulder servo to pin 10
  elbow.attach(11); //Attach the elbow servo to pin 11

  pinMode(magnet, OUTPUT); //Make pin 3 an output

  base.write(93); //Write initial 'home' position for the base
  shoulder.write(90); //Write initial 'home' position for the
shoulder
  elbow.write(125); //Write initial 'home' position for the elbow
}

void loop() {

  if(ready) //Only do this if there have been enough serial

```

```

bytes read
{
    ready = false;

    angles[0] = 180 - angles[0]; //Adjust the value for the base
future angle
    if(angles[0] > 180) //Base cannot rotate further than 180
degrees
        angles[0] = 180;
    if(angles[1] > 60) //Shoulder should never go beyond 45
degrees when picking
        angles[1] = 60;

    angles[2] = (2 * angles[1]) + 35; // Set the value for the
elbow angle with an offset for the servo

    angles[0] += 1; //Adjustment made during testing for better
accuracy
    angles[1] -= 7; //Adjustment made during testing for better
accuracy
    angles[2] -= 5; //Adjustment made during testing for better
accuracy

    if(angles[2] > 125) //Angle for elbow should never go above
125 (90 degrees with the offset) when picking
        angles[2] = 125;

    digitalWrite(magnet, HIGH); //Turn the magnet on before
dropping ontop of the piece
    delay(50);

    setAngles(); //Set the angles for the pick
    delay(25);

    angles[1] = 45; //Set the angles for the lift
    angles[2] = 90; //Set the angles for the lift
    setAngles(); //Set the angles for the lift

```

```

    //The below if statement determines which side the shape
needs to be placed
    //depending on the value of the side (0 = right, 1 = left)
    if(side)
        angles[0] = 180;
    else
        angles[0] = 0;

    angles[1] = 90; //Set the angles for the place
    angles[2] = 150; //Set the angles for the place
    setAngles(); //Set the angles for the place

    digitalWrite(magnet, LOW); //Turn off the magnet
    delay(100);

    angles[0] = 90; //Go to home position
    angles[1] = 90; //Go to home position
    angles[2] = 125; //Go to home position
    setAngles(); //Go to home position

    delay(100);
    Serial.write(1); //Write to the serial port that the part
has been placed
    delay(100);
}

}

//This function takes the desired angles as well as the current
angles of the servos and
//rotates the servos through the necessary angles.
//It rotates 1 degree every 20 ms for the base and elbow, and it
rotates 1 degree every 2 seconds
//for the shoulder. This allows for a smooth path of the robot
arm that isn't super jerky
void setAngles()

```

```

{
  for(int i = 0; i < 180; i++)
  {

    if(base_current < angles[0])
    {
      base_current++;
      base.write(base_current);
    }
    else if(base_current > angles[0])
    {
      base_current--;
      base.write(base_current);
    }

    if(shoulder_current < angles[1] && !(i % 2))
    {
      shoulder_current++;
      shoulder.write(shoulder_current);
    }
    else if(shoulder_current > angles[1]&& !(i % 2))
    {
      shoulder_current--;
      shoulder.write(shoulder_current);
    }

    if(elbow_current < angles[2])
    {
      elbow_current++;
      elbow.write(elbow_current);
    }
    else if(elbow_current > angles[2])
    {
      elbow_current--;
      elbow.write(elbow_current);
    }

    delay(20);
  }
}

```

```
}  
}
```

```
//This function handles a serial event (ie when a serial byte is  
received)
```

```
//It takes the value it receives and assigns it to a parameter  
depending on its value and
```

```
//which order they are received.
```

```
void serialEvent()
```

```
{
```

```
  if(Serial.available() > 0)
```

```
  {
```

```
    int temp = Serial.read();
```

```
    if(temp == 255) //Side = 0, so it's a triangle and all  
angles have been read;
```

```
    {
```

```
      serial_num = 0;
```

```
      side = 0;
```

```
      ready = true;
```

```
    }
```

```
    else if(temp == 254) //Side = 1, so it's a square and all  
angles have been read;
```

```
    {
```

```
      serial_num = 0;
```

```
      side = 1;
```

```
      ready = true;
```

```
    }
```

```
  else
```

```
  {
```

```
    angles[serial_num] = temp; //read the value for the servo  
angles
```

```
    serial_num++;
```

```
  }
```

```
    delay(100);
```

```
  }
```

