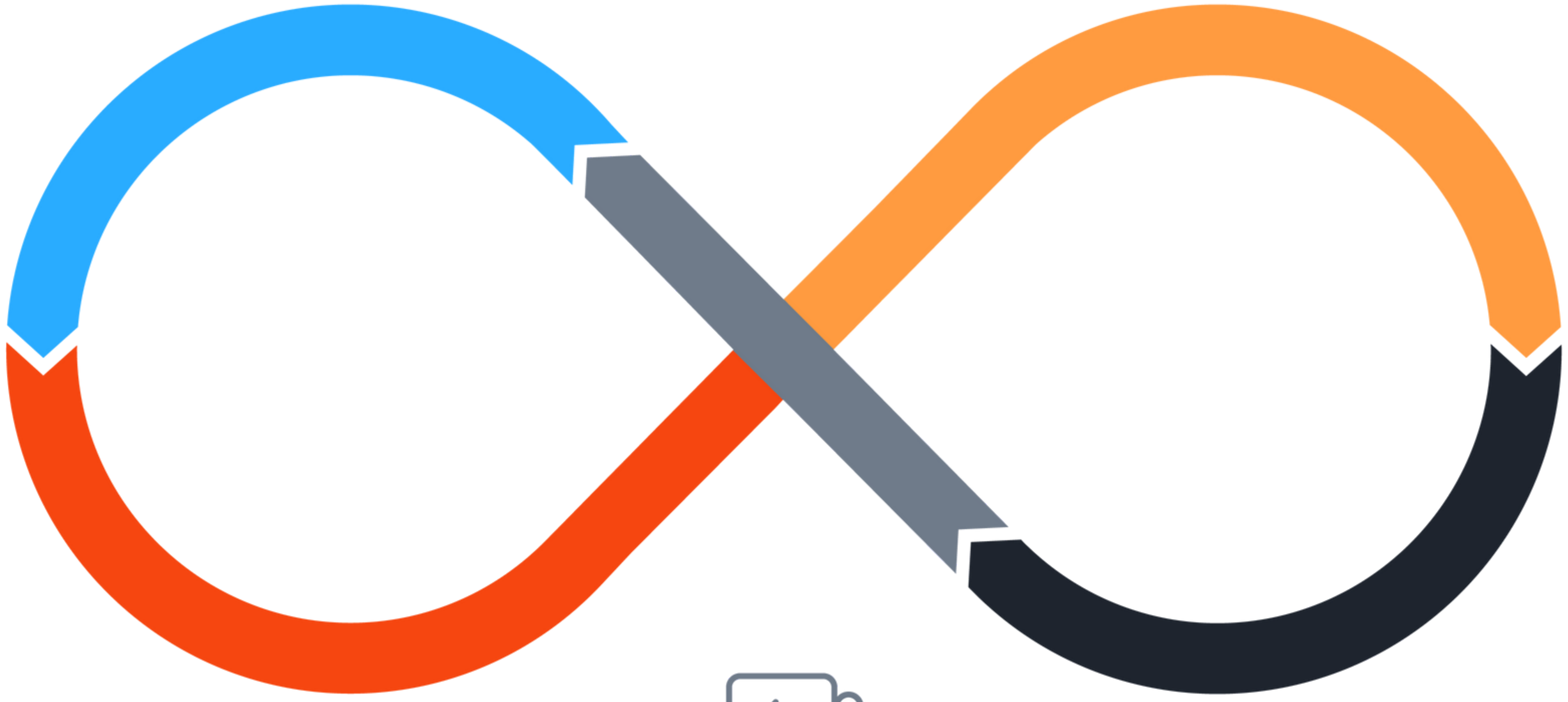
RoboSandy

Logan Jones
Oluwatamilore Ilupeju
Shehryar Ali
Saheel Mohammed

Design Thinking



Empathize
Understanding
people

Define
Figuring out
the problem

Ideate
Generating
your ideas

Prototype
Creation and
experimentation

Test
Refining
the product

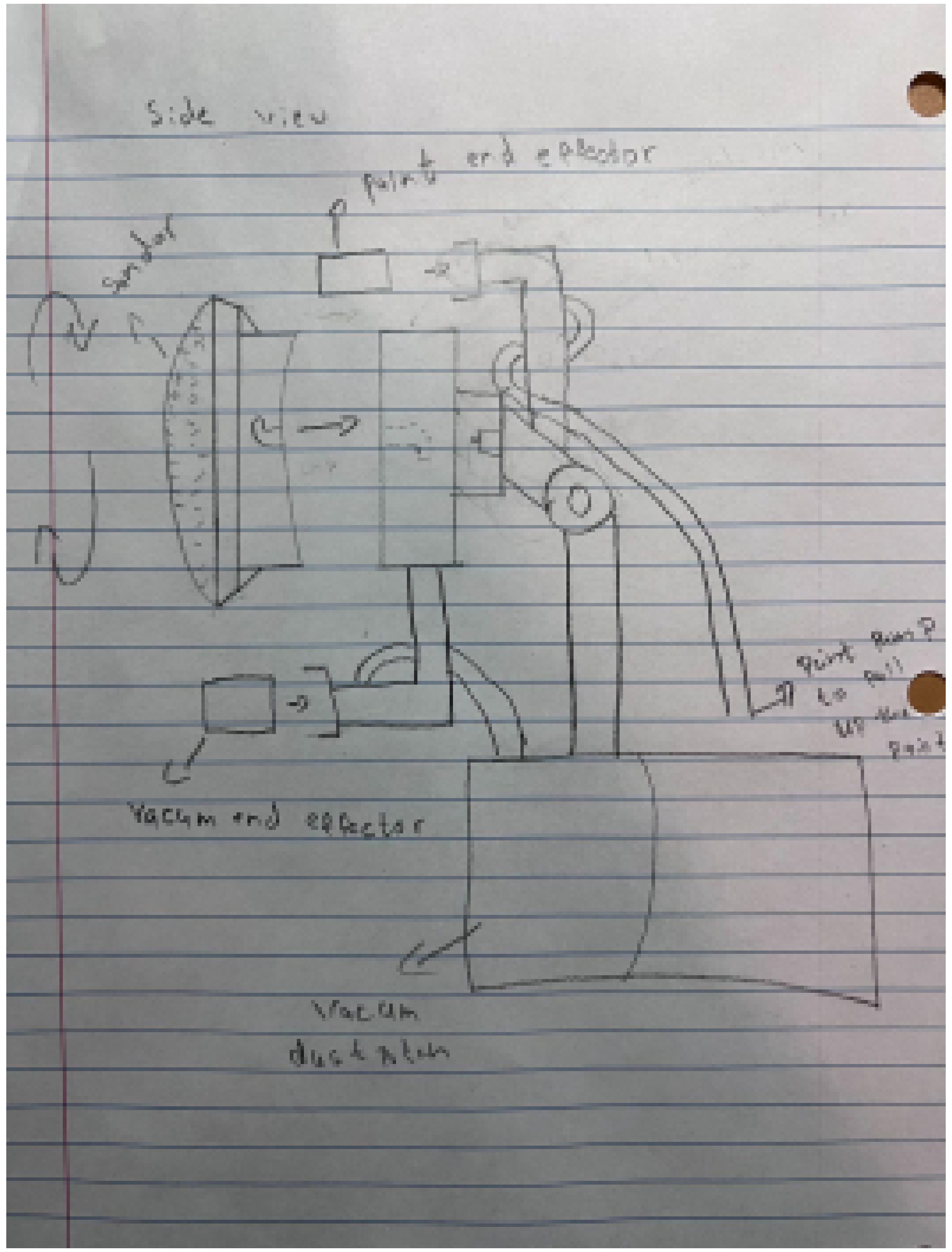
PROBLEM STATEMENT

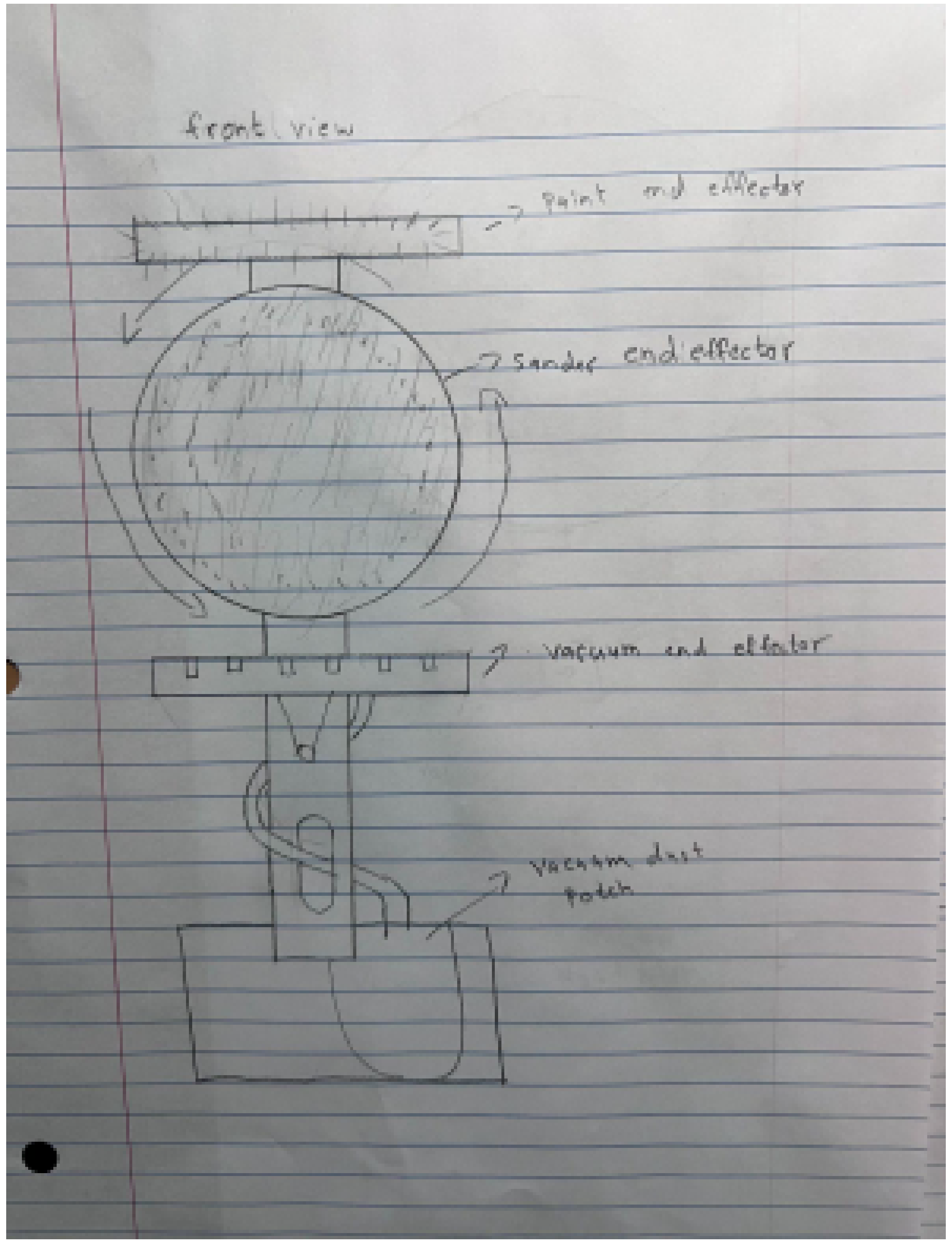
PROBLEM STATEMENT

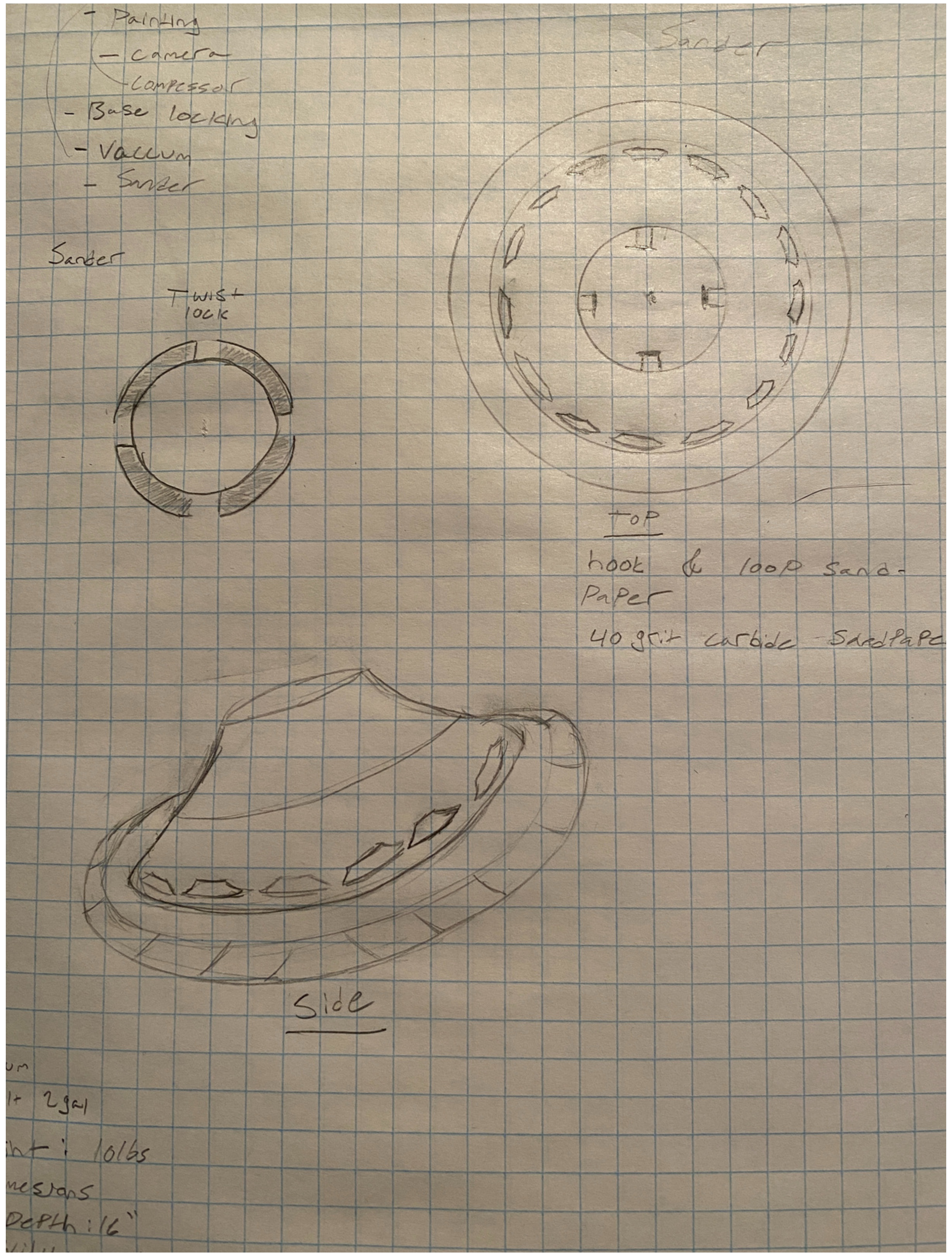
"There exists a need for a lightweight cost-effective 3D printable robotic arm to scan, scrape and repaint over corroded areas of the Halifax class frigate, operated using a simple GUI."

Priority Needs

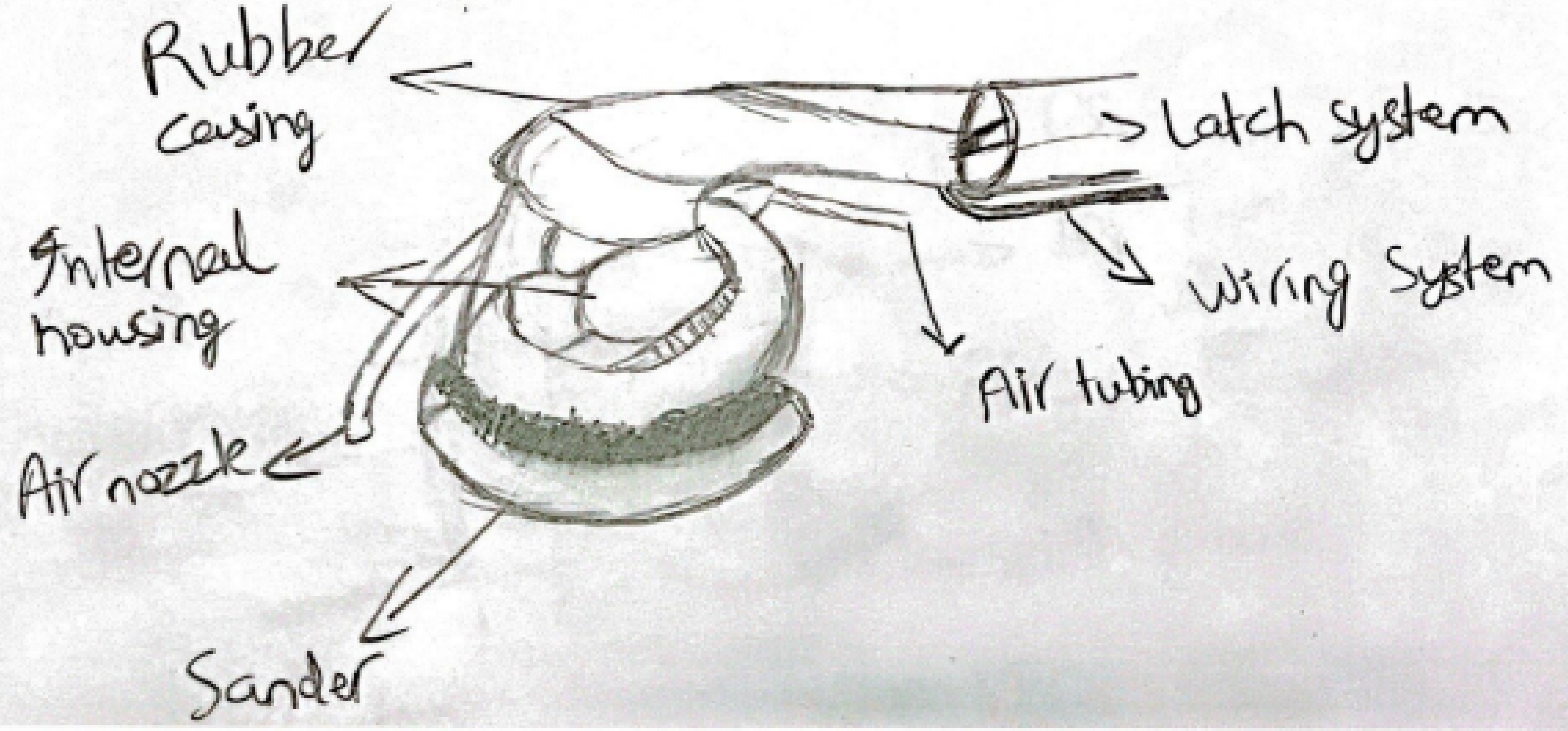
1. Inverse Kinematics
2. GUI
3. Safety
4. End-effectors


Side view
paint end effector
Sander
paint pump to pull up the paint
Vacum end effector
Vacum dust patch

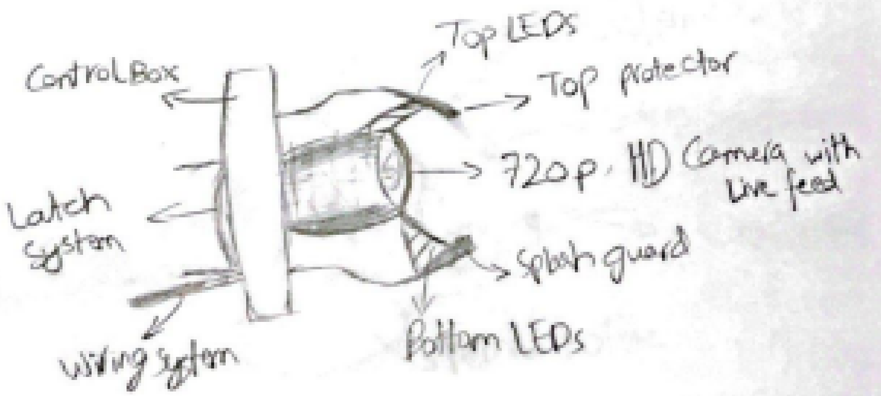

front view
paint end effector
Sander end effector
vacuum end effector
vacuum dust patch


- Painting
- camera
Compessor
- Base locking
- Vaccum
- Sander
Sander
Sander
Twist lock
Top
hook & loop sand-paper
40 grit carbide sandpape
Side
um
lt 2gal
ht : 10lbs
mensions
Depth : 16"

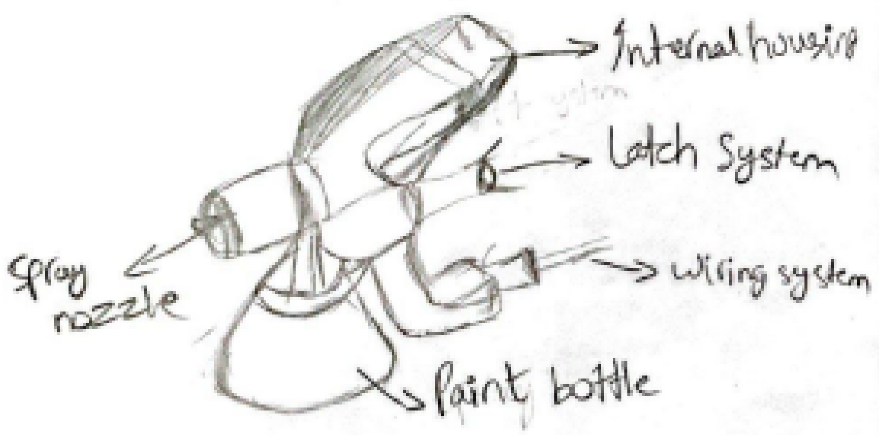
Sander System



Camera System

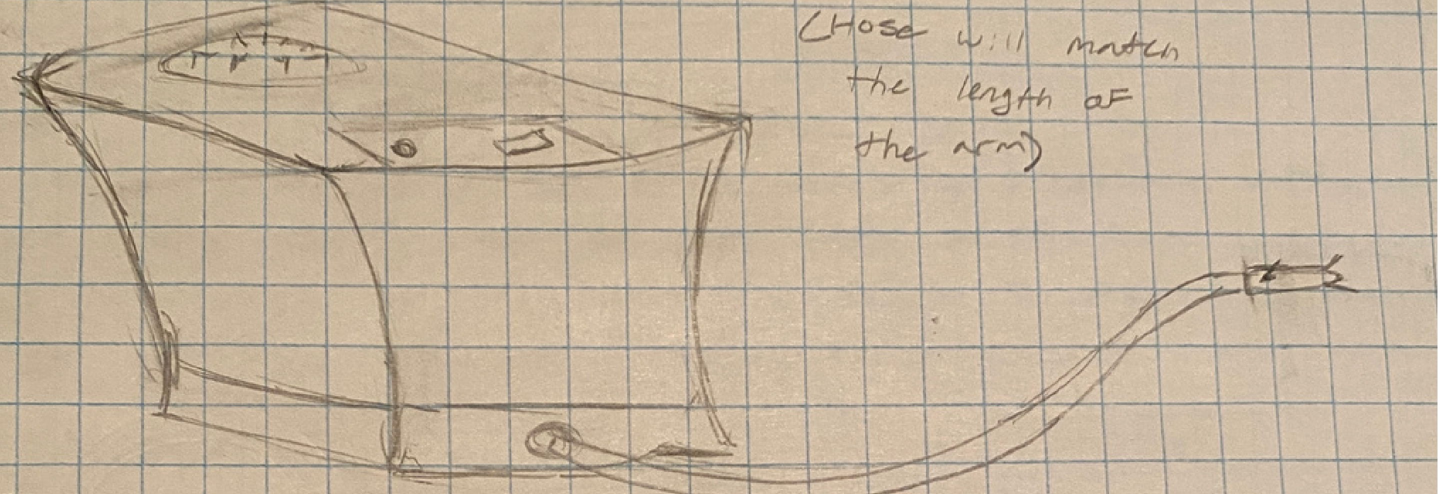


Repainting System



Paint gun (Based on)

- Wagner Control pro 130 Power tank Airless Stand Paint Sprayer



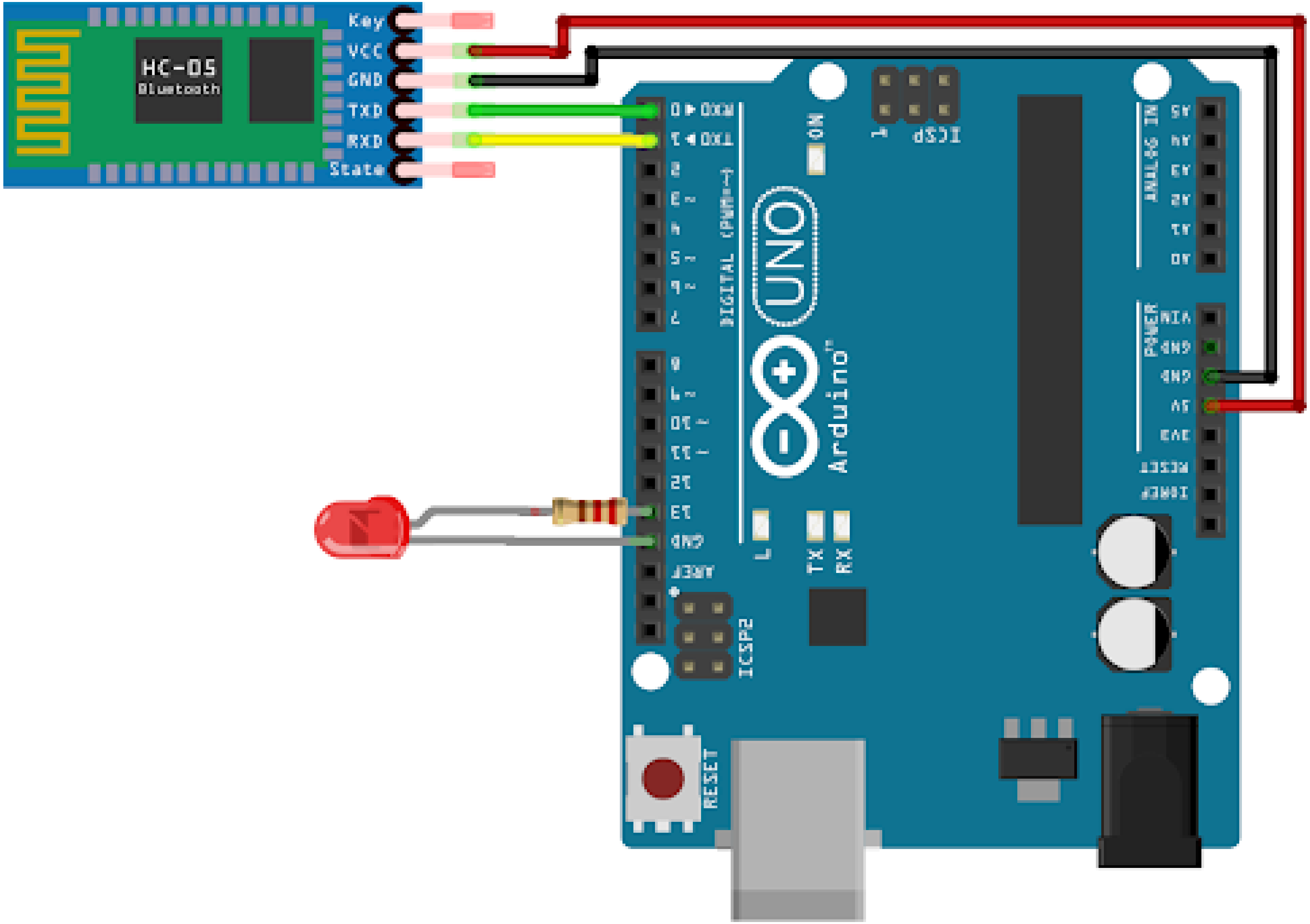
(Hose will match the length of the arm)

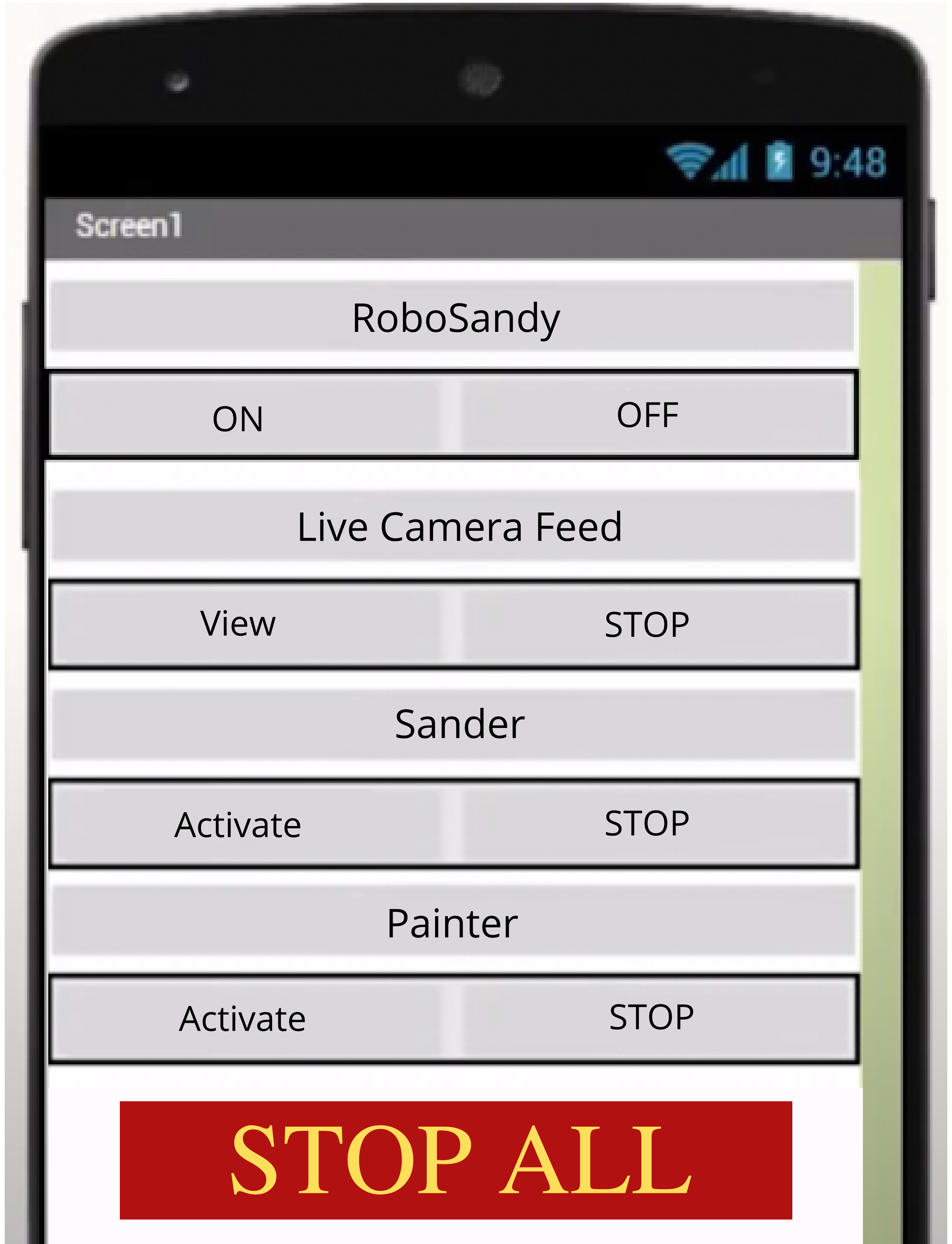
Dimensions
Depth 12"
Width 13.25"
Height 13.25"

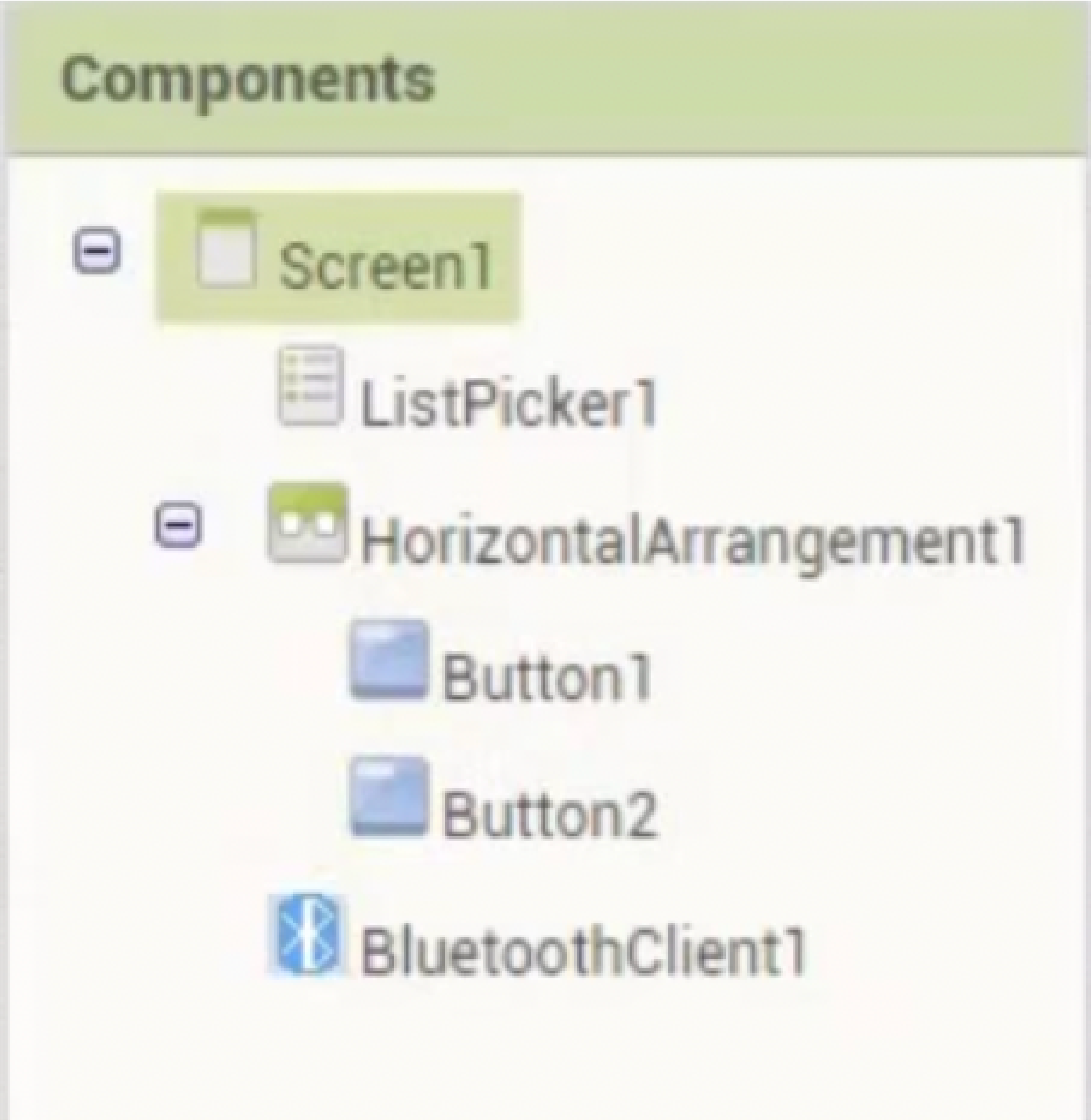
Weight
11 lbs (can be cut down)

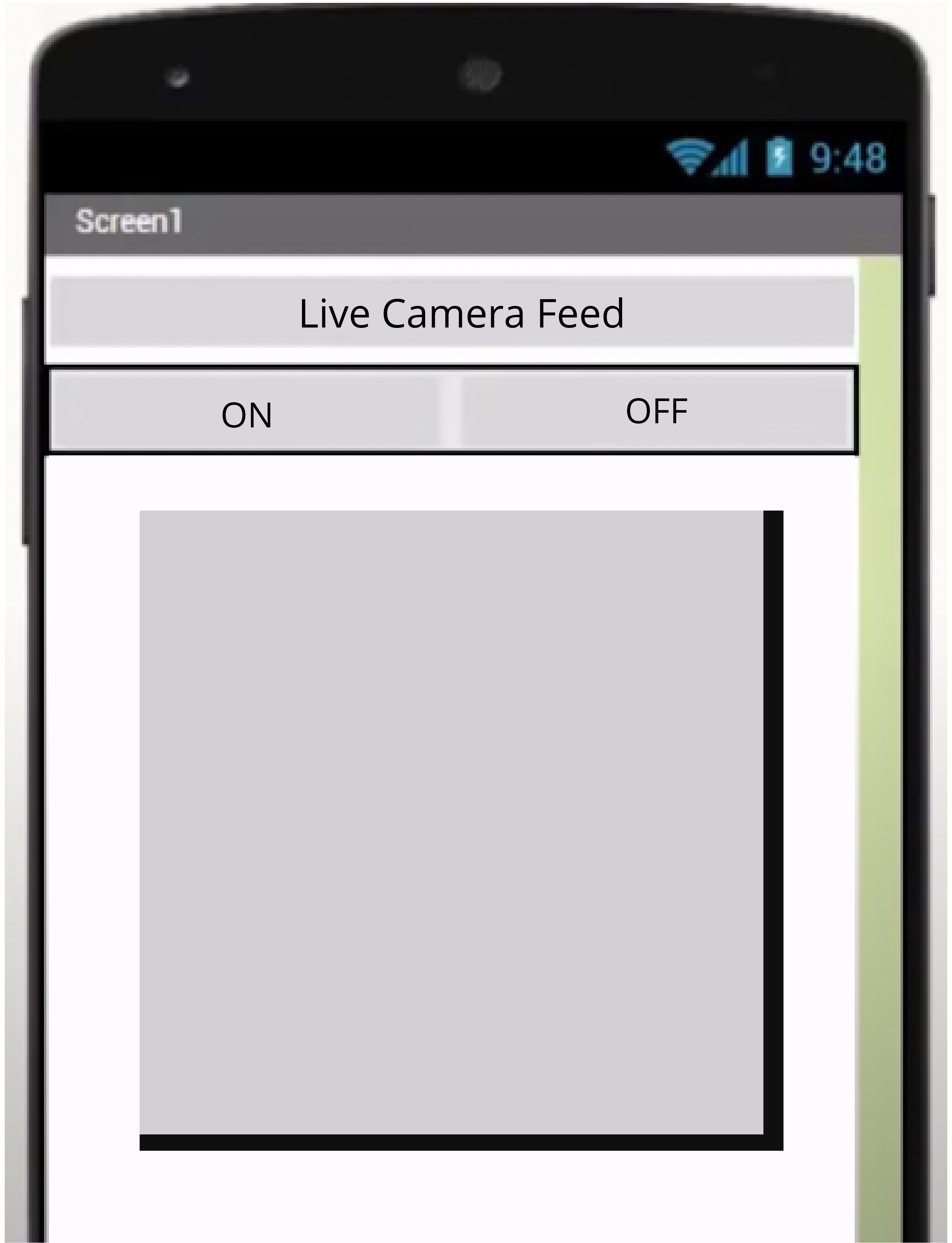
Final Prototype

GUI

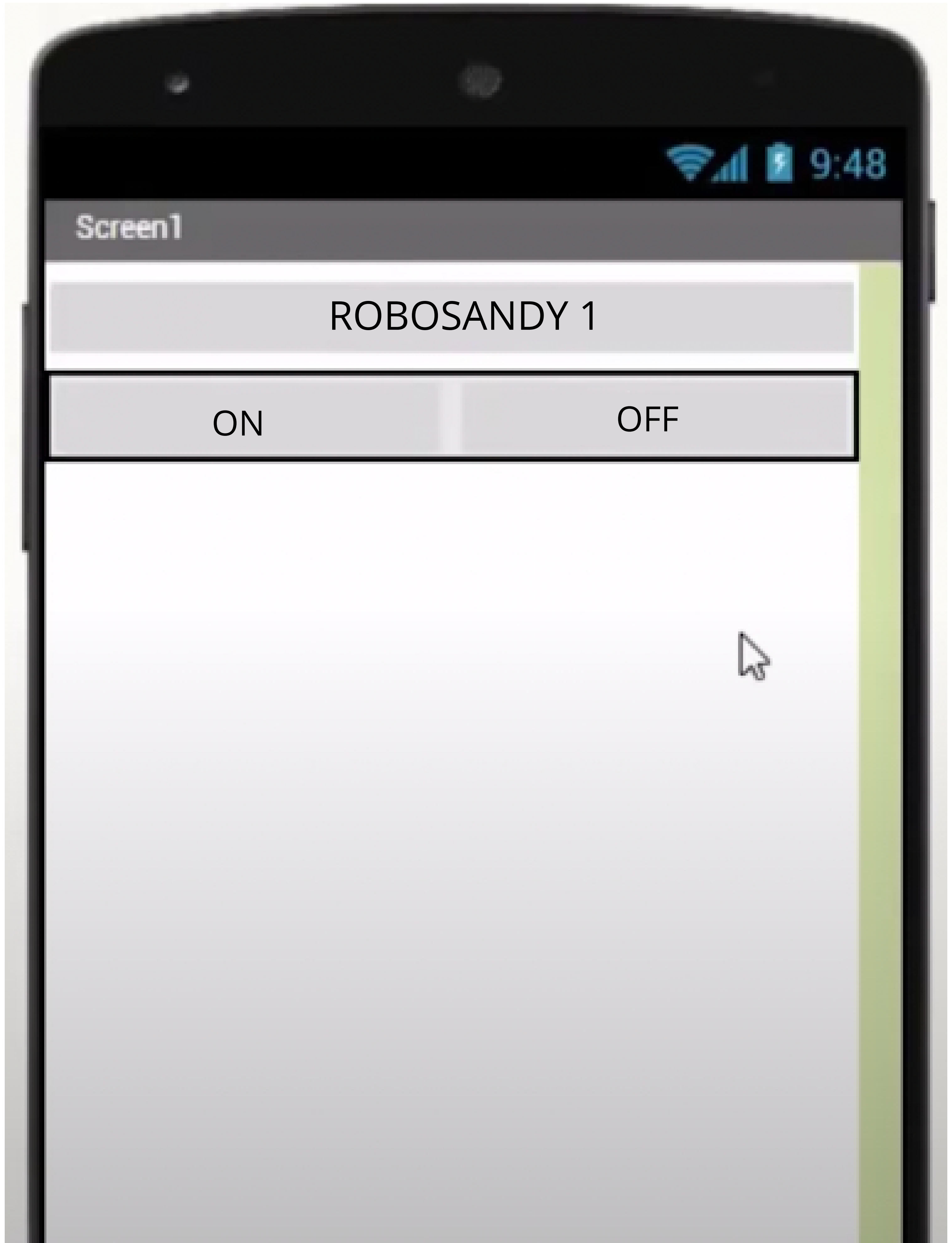

HC-05
Bluetooth
Key
VCC
GND
TXD
RXD
State
RX◀0
TX▶1
DIGITAL (PWM~)
UNO
Arduino
ON
ICSP
ANALOG IN
POWER
VIN
GND
GND
5V
3V3
RESET
IOREF
AREF
GND
13
12
~11
~10
~9
8
L
TX
RX
ICSP2
RESET


9:48
Screen1
RoboSandy
ON
OFF
Live Camera Feed
View
STOP
Sander
Activate
STOP
Painter
Activate
STOP
STOP ALL


Components
Screen1
ListPicker1
HorizontalArrangement1
Button1
Button2
BluetoothClient1



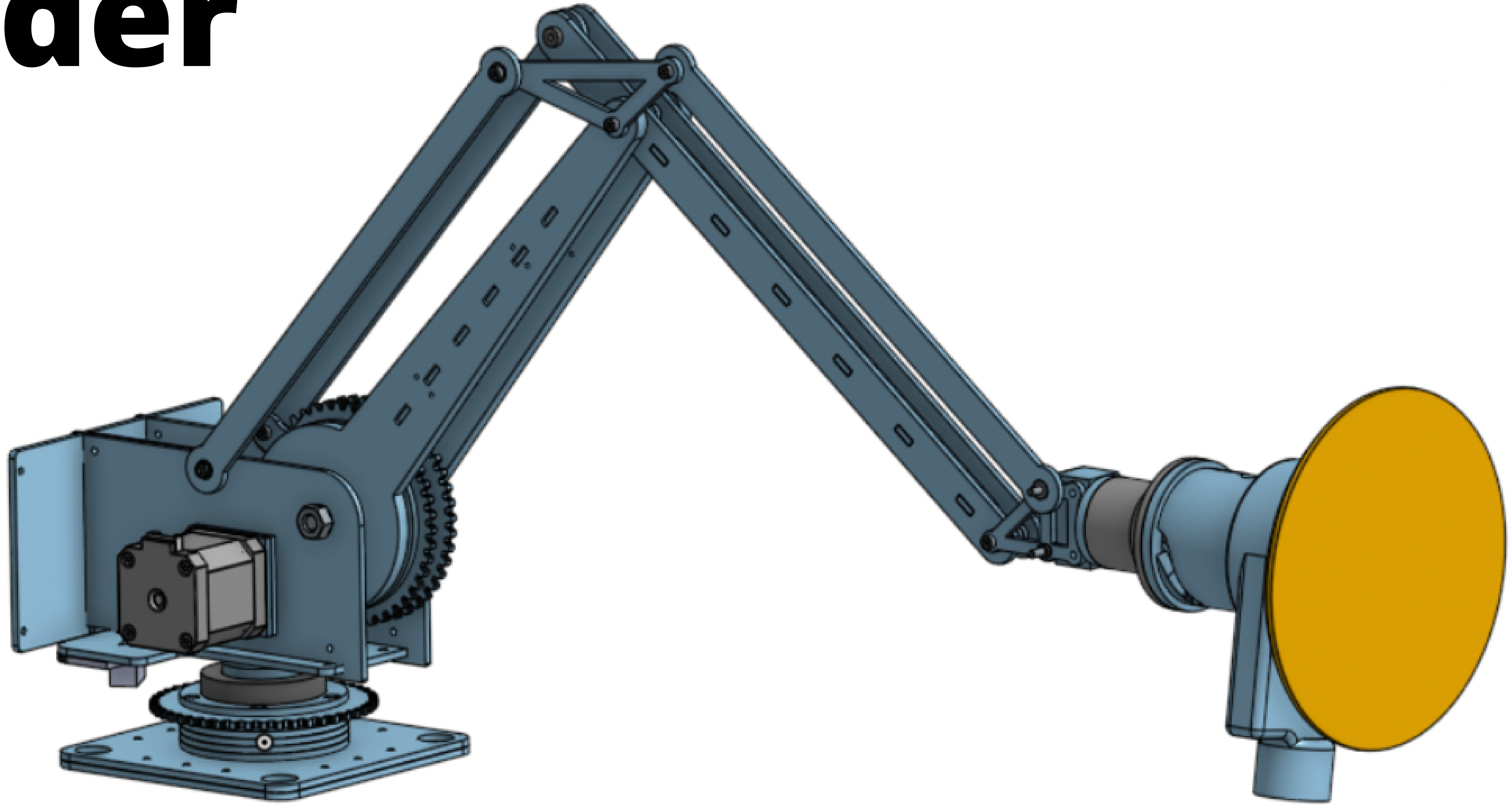
9:48
Screen1
Live Camera Feed
ON
OFF

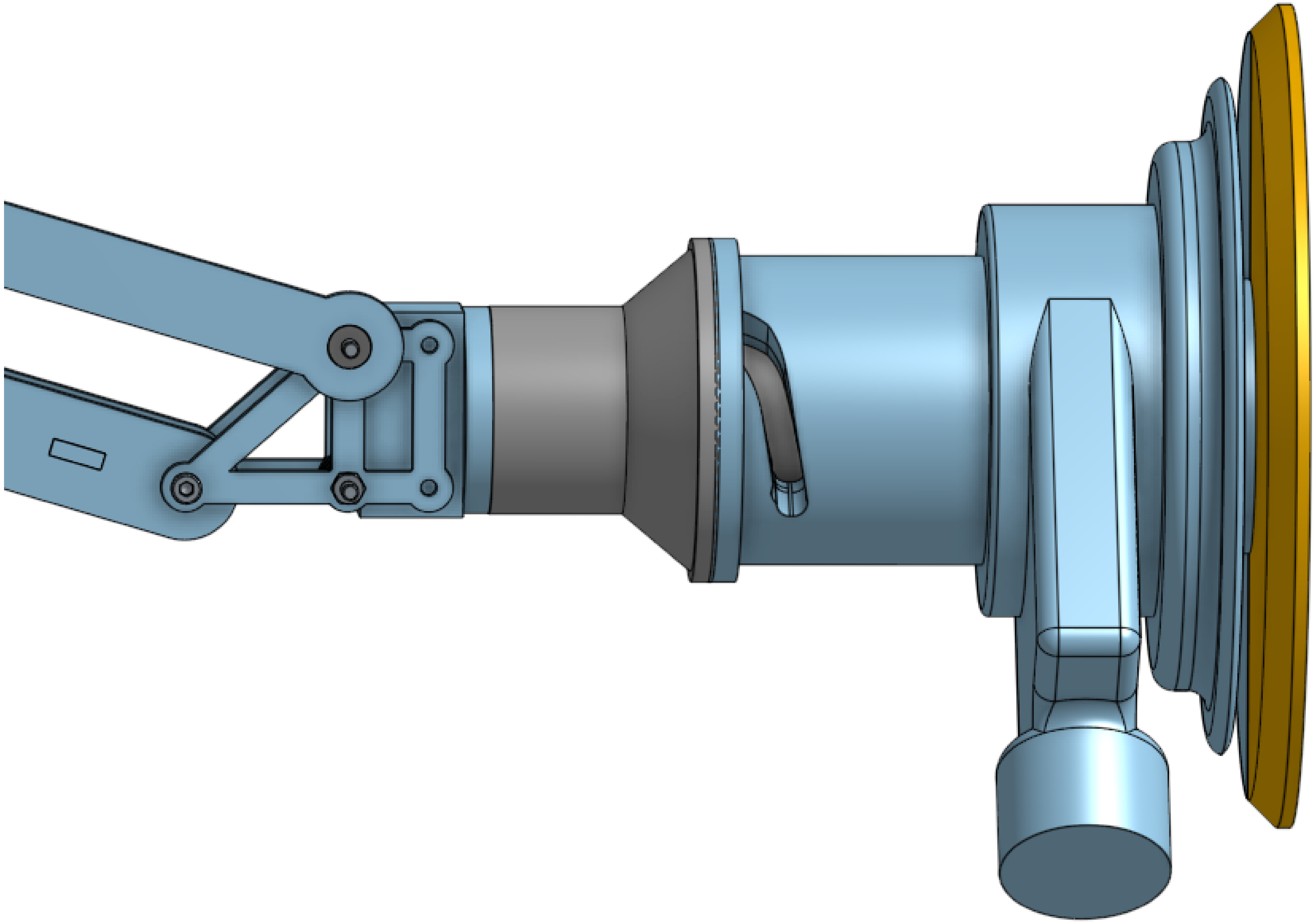


9:48
Screen1
ROBOSANDY 1
ON
OFF

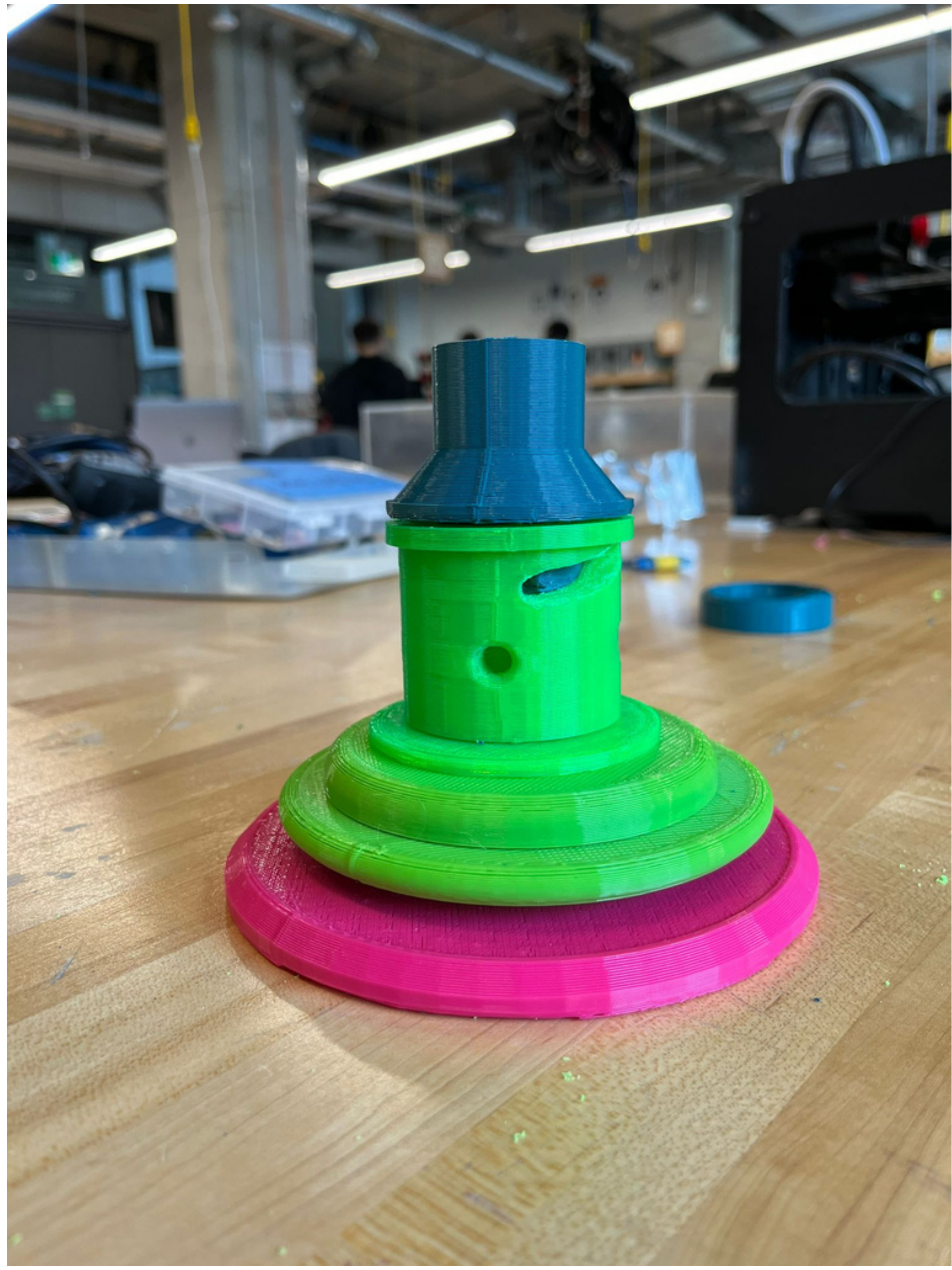
Physical Model

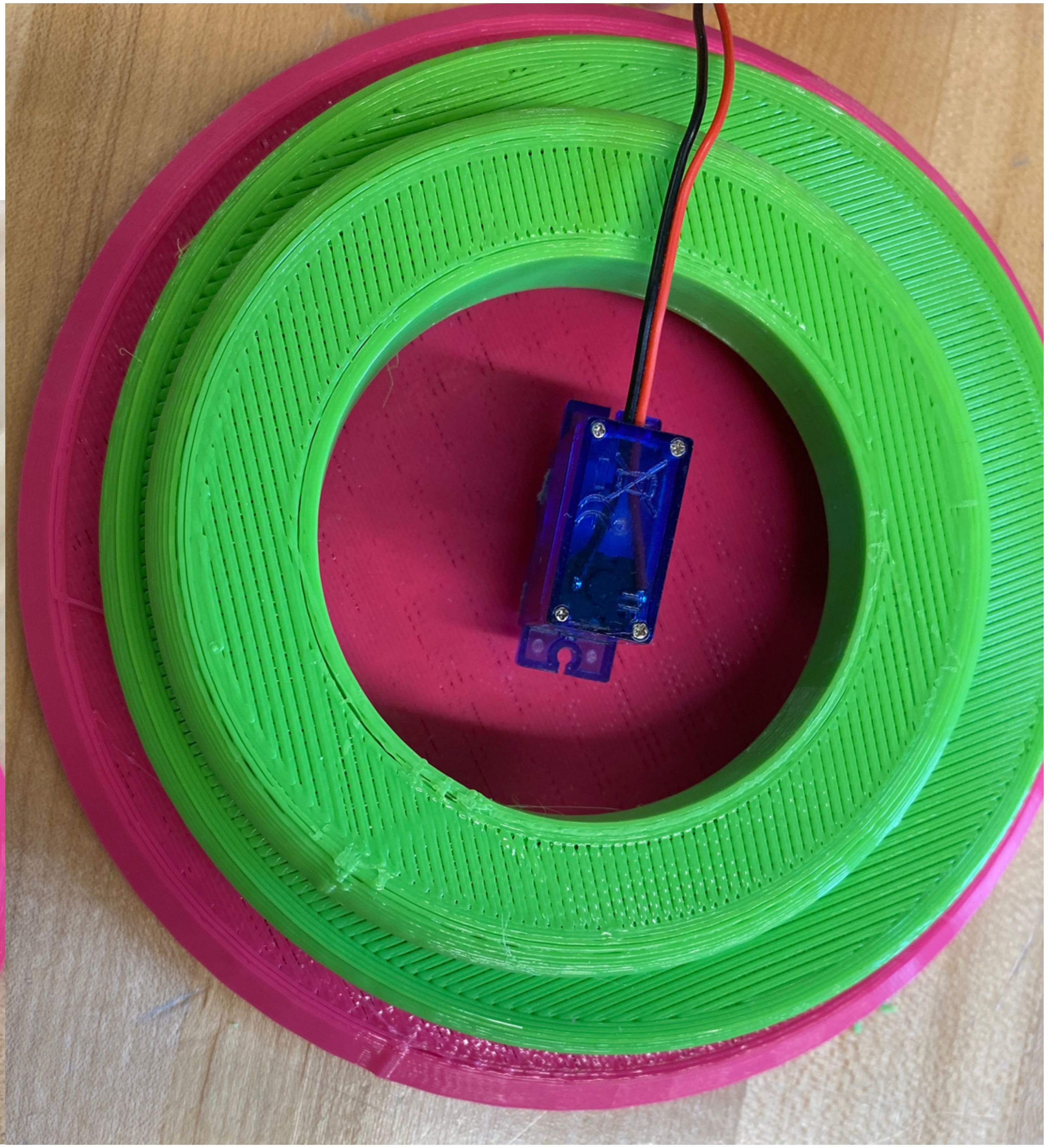
Sander



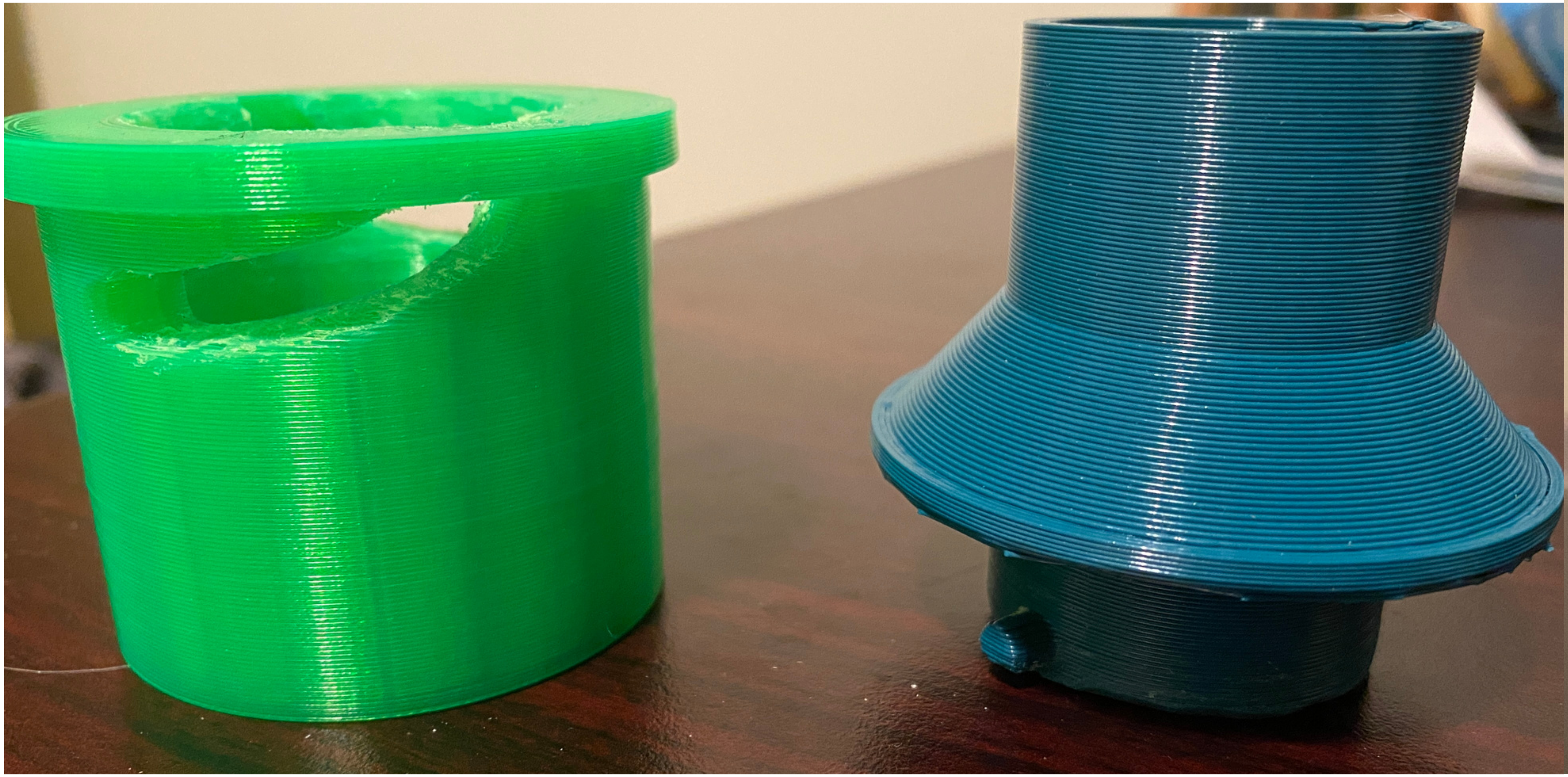


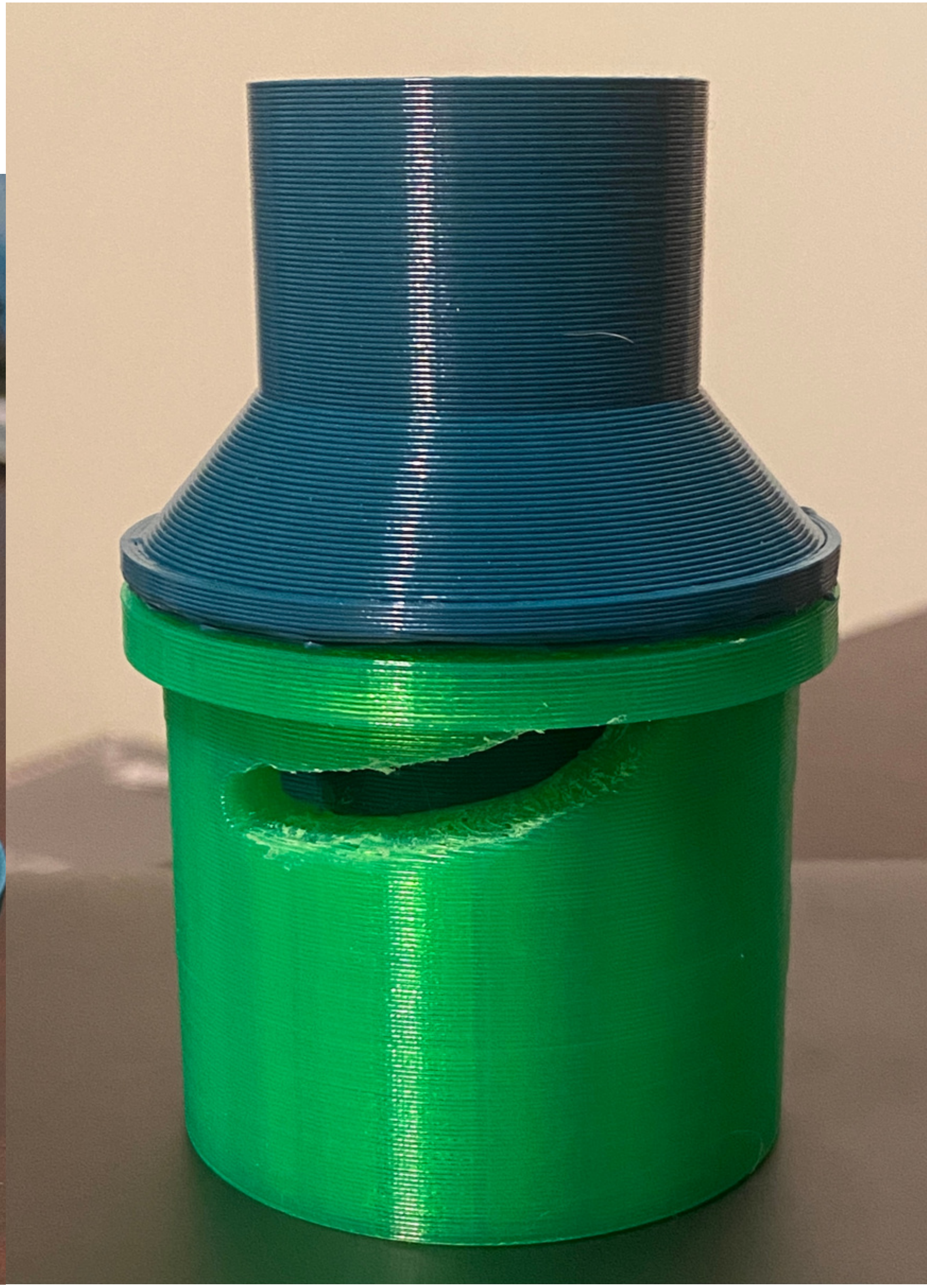
Printed Prototype



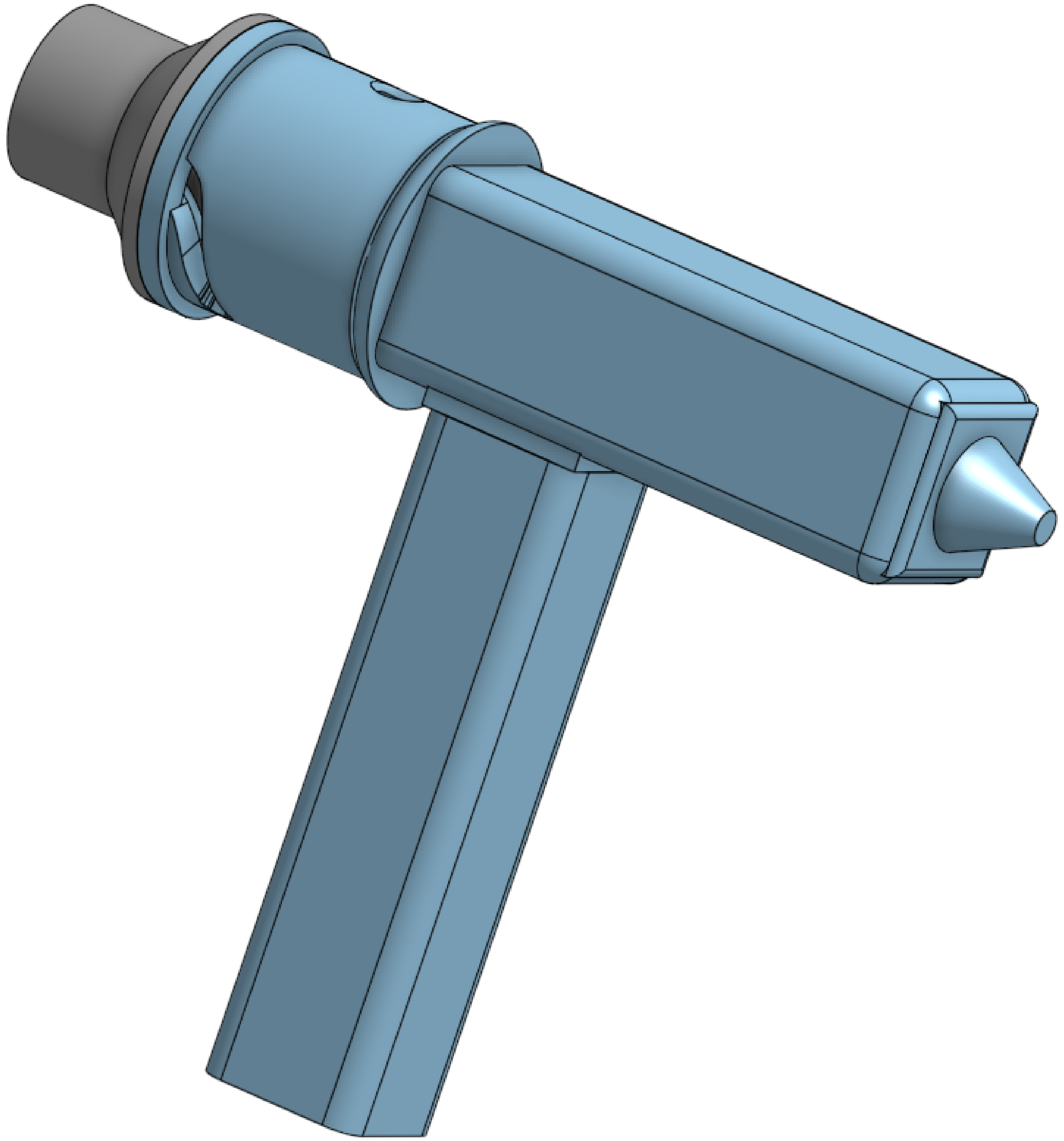


Twist Lock

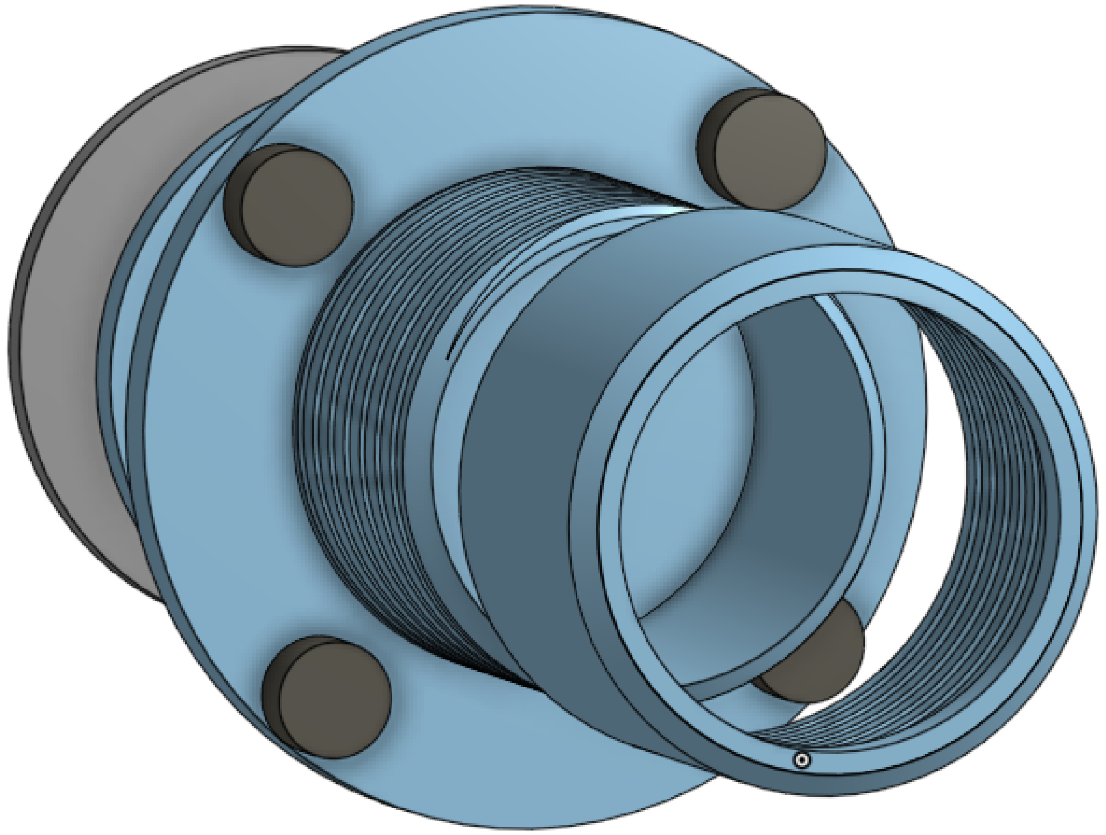


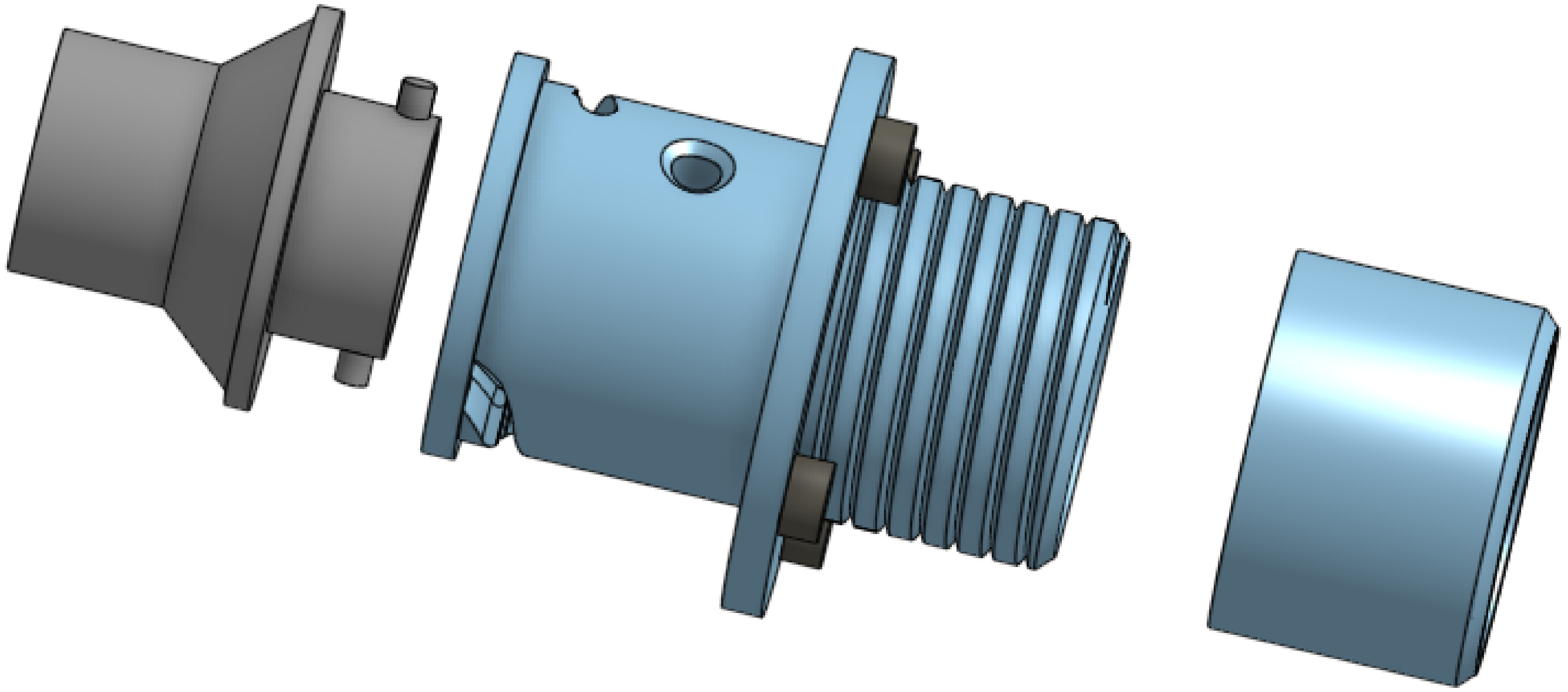


Paint gun



Camera case

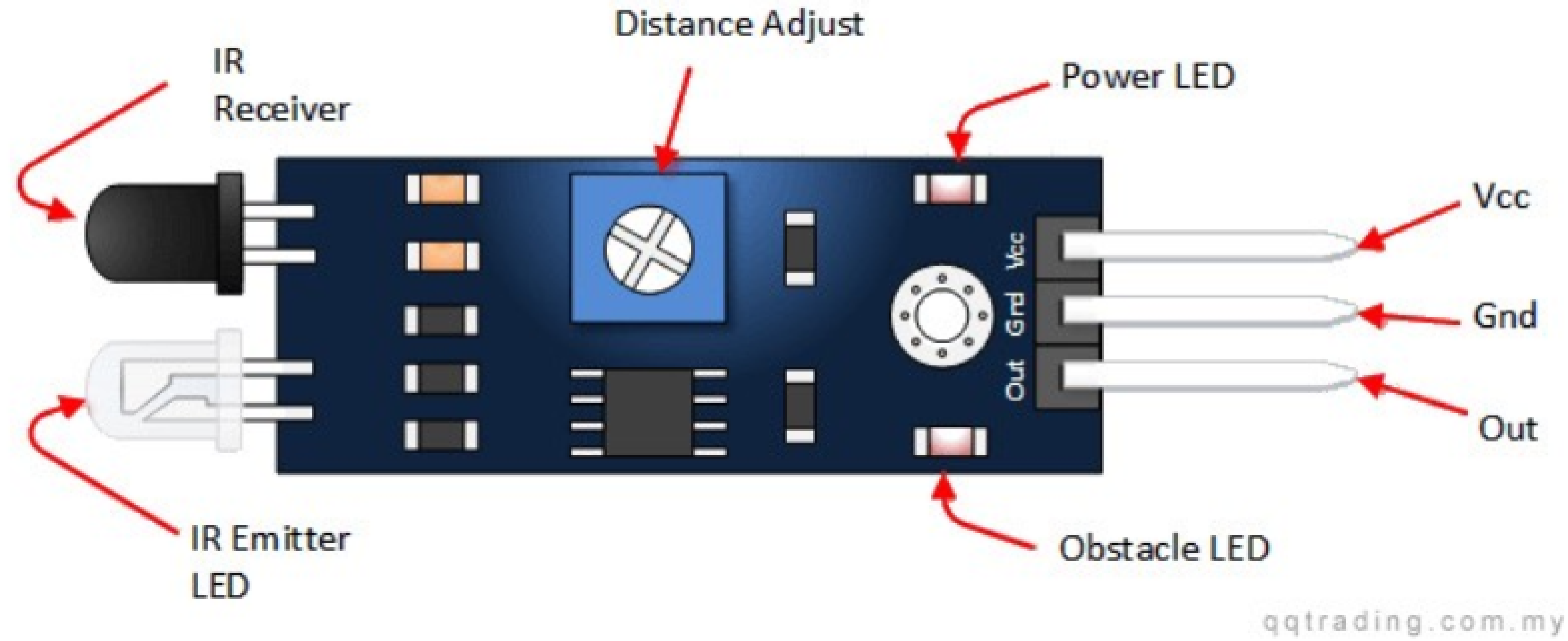




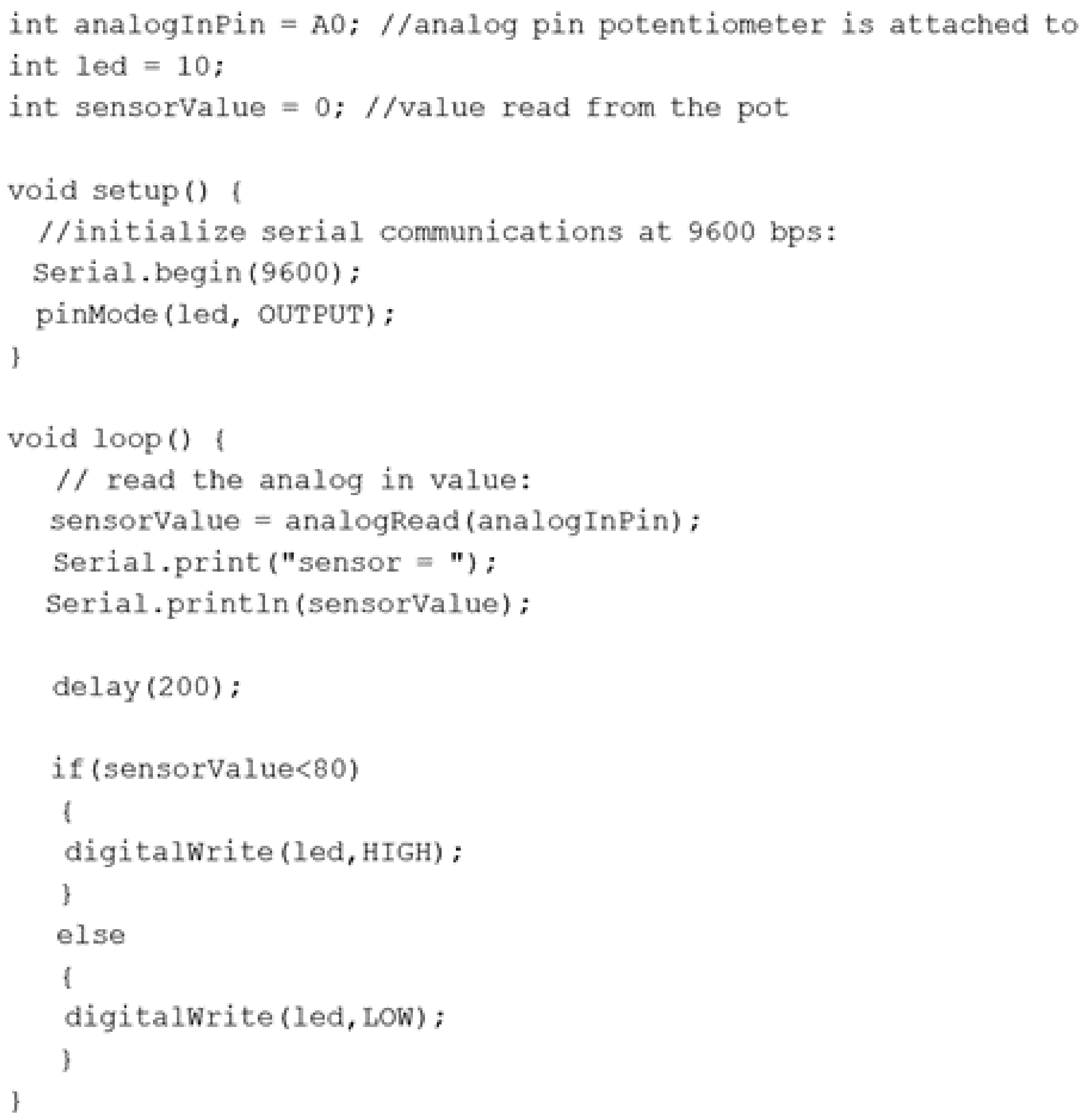
Safety

- Manual force-stop button

- IR sensor mounted at the base of the arm


Distance Adjust
IR
Receiver
Power LED
Vcc
Vcc
Gnd
Gnd
Out
Out
IR Emitter
LED
Obstacle LED
qqtrading.com.my

Obstruction Detection Software



```
int analogInPin = A0; //analog pin potentiometer is attached to
int led = 10;
int sensorValue = 0; //value read from the pot

void setup() {
  //initialize serial communications at 9600 bps:
 Serial.begin(9600);
  pinMode(led, OUTPUT);
}

void loop() {
   // read the analog in value:
   sensorValue = analogRead(analogInPin);
   Serial.print("sensor = ");
  Serial.println(sensorValue);

   delay(200);

   if(sensorValue<80)
    {
    digitalWrite(led,HIGH);
    }
   else
    {
    digitalWrite(led,LOW);
    }
}
```

Inverse Kinematics

```
#include <InverseK.h>
#include <Wire.h>
#include<Adafruit_PWMServoDriver.h>
#include <Servo.h>

Servo motor1;
Servo motor2;
Servo motor3;
// angles in degree and radian

// length of links of robot arm
volatile float L1;
volatile float L2;

// end effector
volatile float pi = 3.14159265359;
Adafruit_PWMServoDriver pca9685 = Adafruit_PWMServoDriver(0X40);

#define SERVOMIN 80
#define SERVOMAX 600

#define SER0 0
#define SER1 1
#define SER2 2

int pwm0;
int pwm1;
int pwm2;
float x=1;
float y=1;
float z=1;


void setup() {
  Serial.begin(115200);
  pca9685.begin();
  pca9685.setPWMFreq(50);
   Serial.println("Enter the length of first arm ");
  while(Serial.available()==0){}
  //L1=Serial.parseFloat();
   L1=6.5;

   Serial.println("Enter the length of second arm ");
  while(Serial.available()==0){}
  //L2=Serial.parseFloat();
   L2=8;
}

void loop() {
   motor1.write(0);
   motor2.write(0);
   motor3.write(0);
   InverseKinematics();
}

void InverseKinematics(){
  float angle1;
  float angle2;
  float angle3;
  double rad_angle1;
  double rad_angle2;
  double rad_angle3;
  float x;
  float y;
  float z;
   Serial.println("Enter the value x ");
      while(Serial.available()==0){}
      x=Serial.parseFloat();


      Serial.println("Enter the value y ");
      while(Serial.available()==0){}
      y=Serial.parseFloat();

      Serial.println("Enter the value z ");
      while(Serial.available()==0){}
      z=Serial.parseFloat();


      rad_angle2 = acos((sq(z)+ sq(y) - sq(L1) - sq(L2)) / (2*L1*L2));
      rad_angle3 = acos((sq(L1) + sq(L2) - sq(x)- sq(z)) / (2*L1*L2));
      rad_angle1= atan2(y , x) - atan2(L1*sin(rad_angle3),L2*sin(rad_angle2))/
(L1*cos(rad_angle3)+ L2*cos(rad_angle2));
      delay(1000);

      angle1= (rad_angle1*180)/pi;
      angle2= (rad_angle2*180)/pi;
      angle3= (rad_angle3*180)/pi;

  Serial.print("x is ");
  Serial.println(x);
  Serial.print("y is ");
  Serial.println(y);
  Serial.print("z is ");
  Serial.println(z);
  Serial.print("angle1 is  ");
  Serial.println(angle1);
  Serial.print("angle2 is ");
  Serial.println(angle2);
  Serial.print("angle3 is ");
  Serial.println(angle3);
  motor1.write(angle1);
  motor2.write(angle2);
  motor3.write(angle3);

  pwm0 = map(angle1, 0, 180, SERVOMIN, SERVOMAX);
  pwm1 = map(angle1, 0, 180, SERVOMIN, SERVOMAX);
  pwm2 = map(angle1, 0, 180, SERVOMIN, SERVOMAX);
  pca9685.setPWM(SER0, 0, pwm0);
  pca9685.setPWM(SER1, 0, pwm1);
  pca9685.setPWM(SER2, 0, pwm2);
  delay(2000);
  x=x+0.1;
  y=y+0.1;
  z=z+0.1;

}
```

Lessons Learned

- Communication
- Time-management
- Responsibility
- Conflict-resolution

Challenges Faced

- Implementation of Inverse Kinematics
- 3D printing
- Budget
- GUI

Future Plans

- 3D printing paint gun
- Working on the functionality of GUI
- Testing all software/hardware individually
- Minimum Viable Product (MVP) Testing
- User Manual

THANK YOU!

Questions?