MECHATRONICS PROJECT

SPRING SORTER

PRESENTED BY GAURAV SHETTY

DR. VIKRAM KAPILA

NYU TANDON SCHOOL OF ENGINEERING

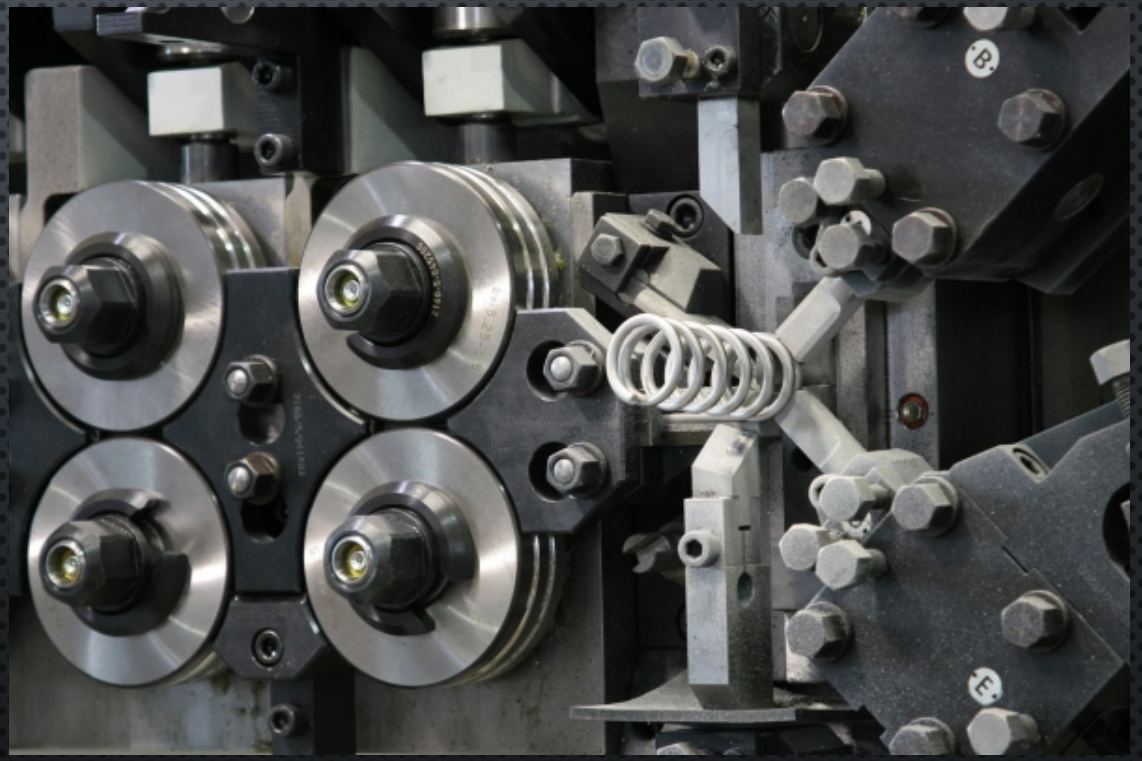
PROJECT OBJECTIVE -

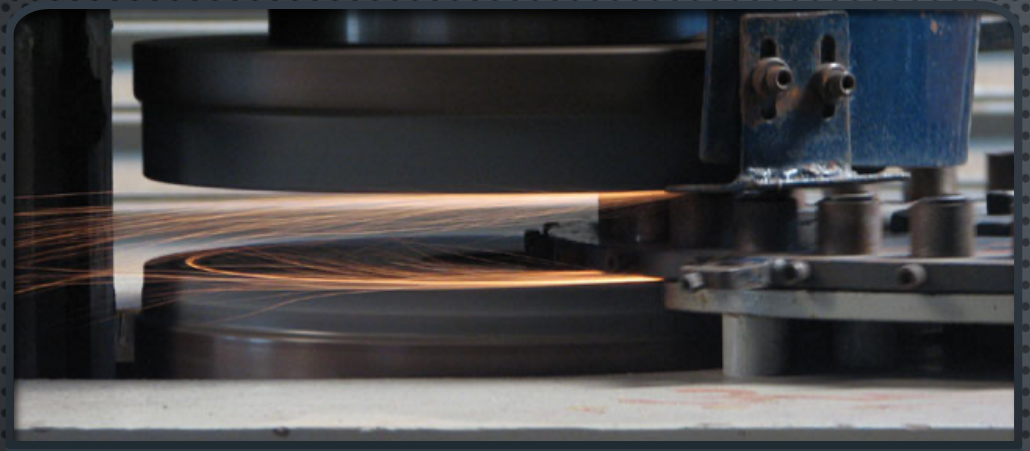
- DESIGN AND BUILD A MECHATRONICS SOLUTION TO AUTOMATE THE PROCESS OF LOADING COIL SPRINGS INTO A CARTRIDGE AND TRANSPORTING THEM TO THE NEXT STEP OF PRODUCTION, WHICH IS THE GRINDING MACHINE.
- CARTRIDGE IS DESIGNED TO OFFSET AT THE BOTTOM TO HOLD THE SPRINGS IN PLACE DURING TRANSPORTATION TO THE GRINDING MACHINE

SPRING PRODUCTION PROCESS

- 1. COIL SPRING PRODUCTION MACHINE

  - Rate of spring production varies from 80 pcs/min to 100 pcs/min
  - Depends on spring density, materials used.
  - Springs are collected and loaded into cartridges, which are placed on the grinding machine

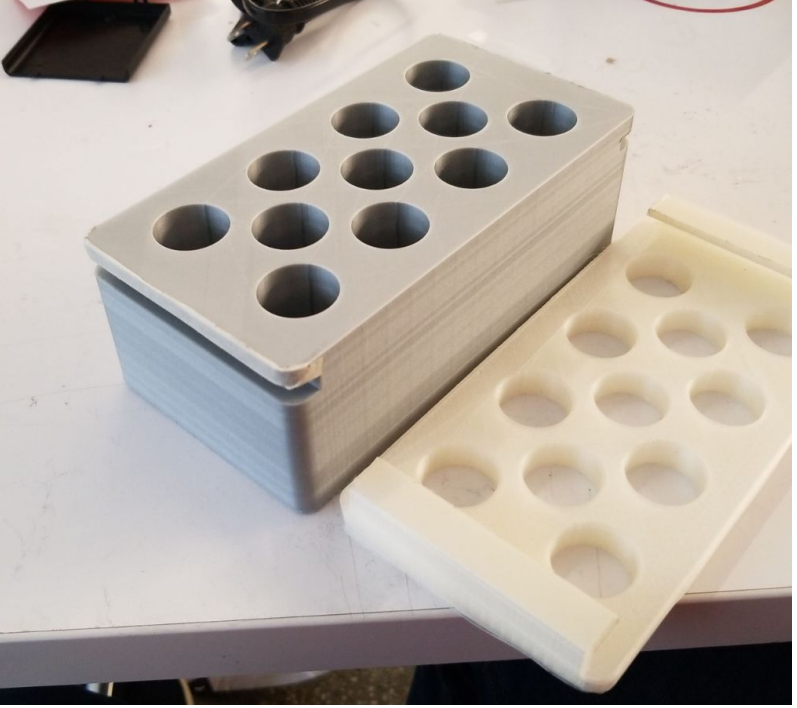


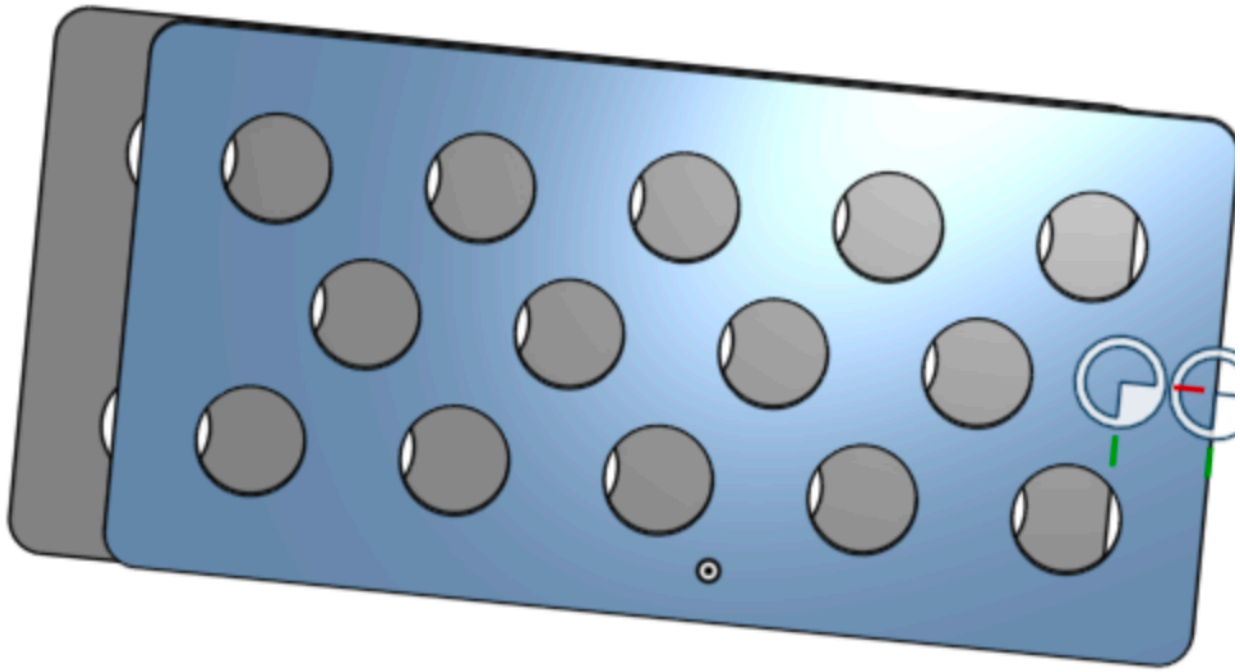


SPRING PRODUCTION PROCESS

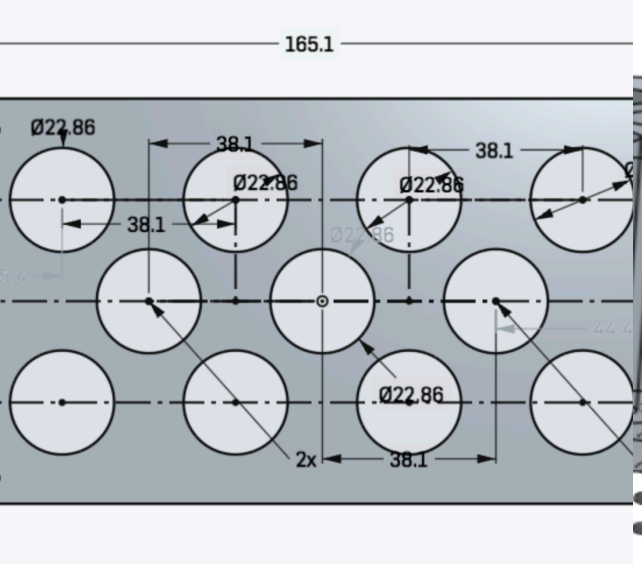
SPRING GRINDING MACHINE –

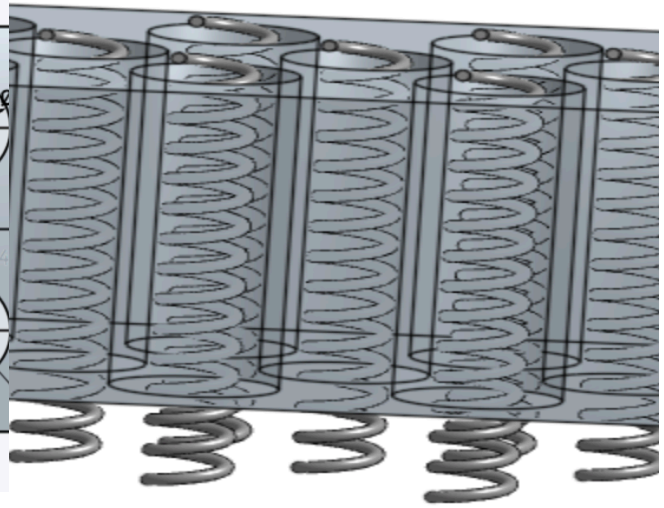
- ROTATING PLATE WITH CARTRIDGES
- TOP AND BOTTOM SURFACE OF SPRINGS WILL BE GRINDED DOWN.

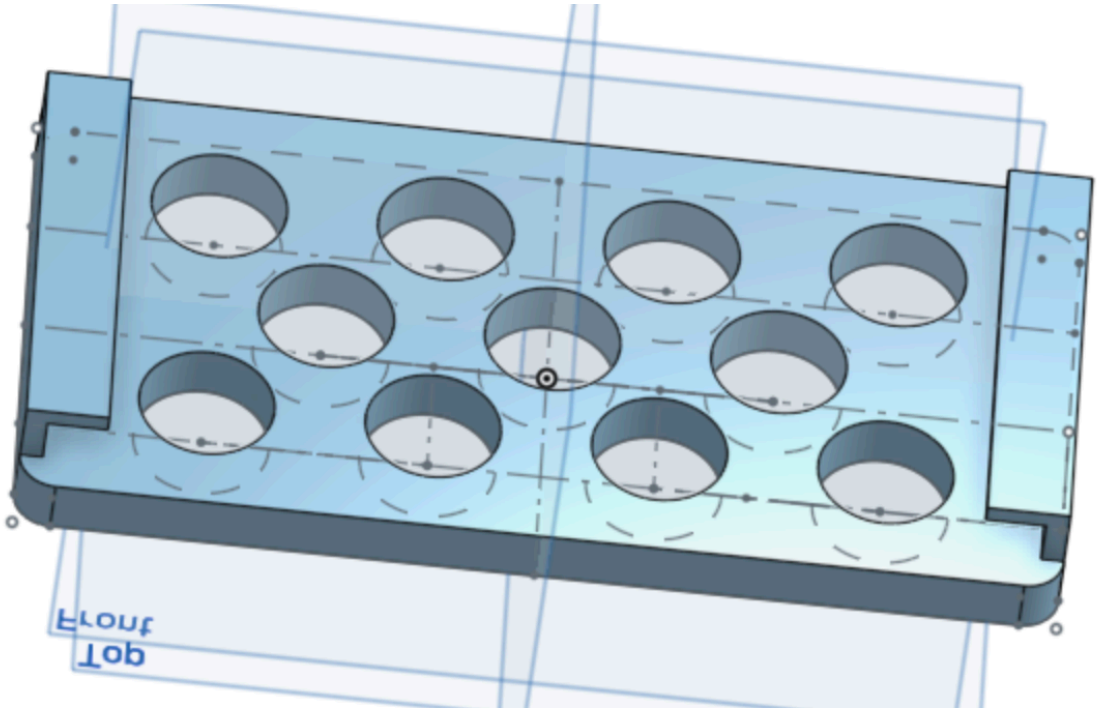




MODIFIED CARTRIDGE

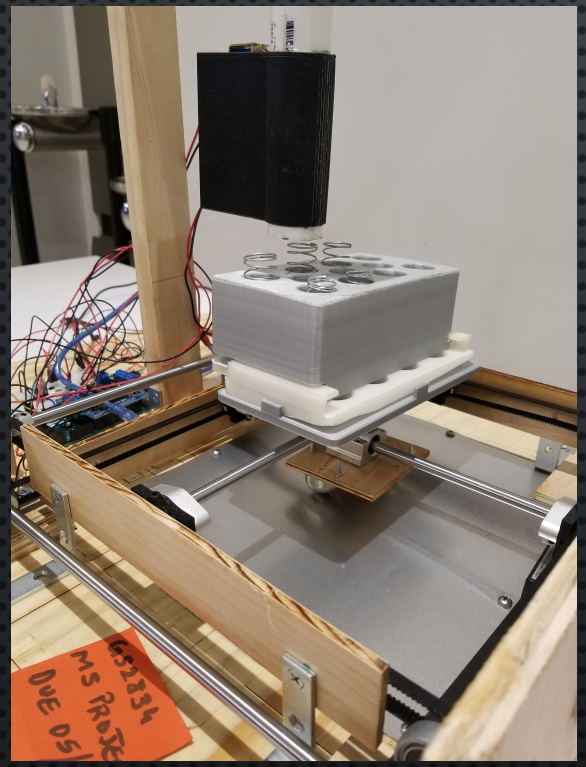


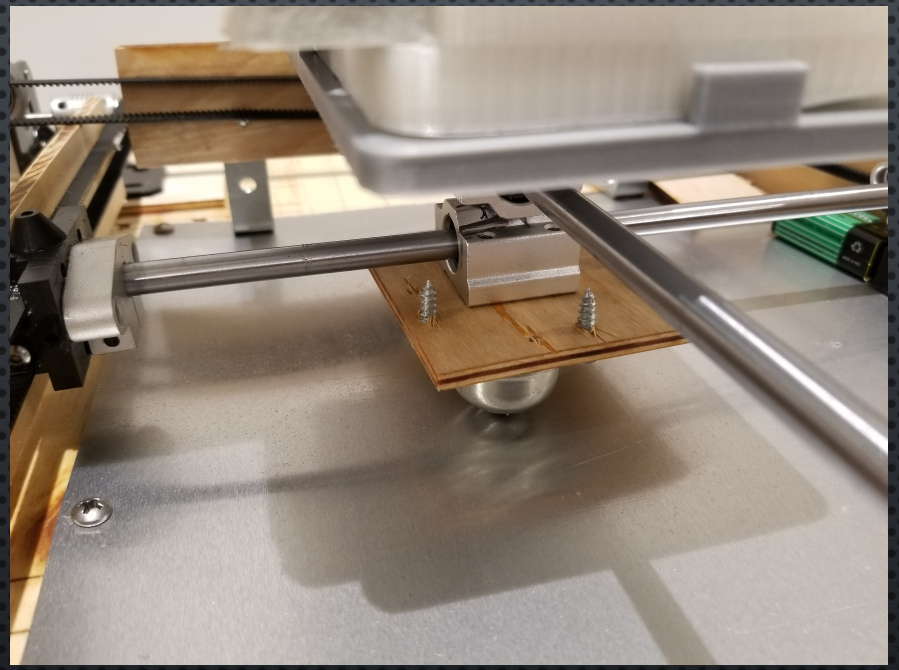




SPRING LOADING INTO CARTRIDGE

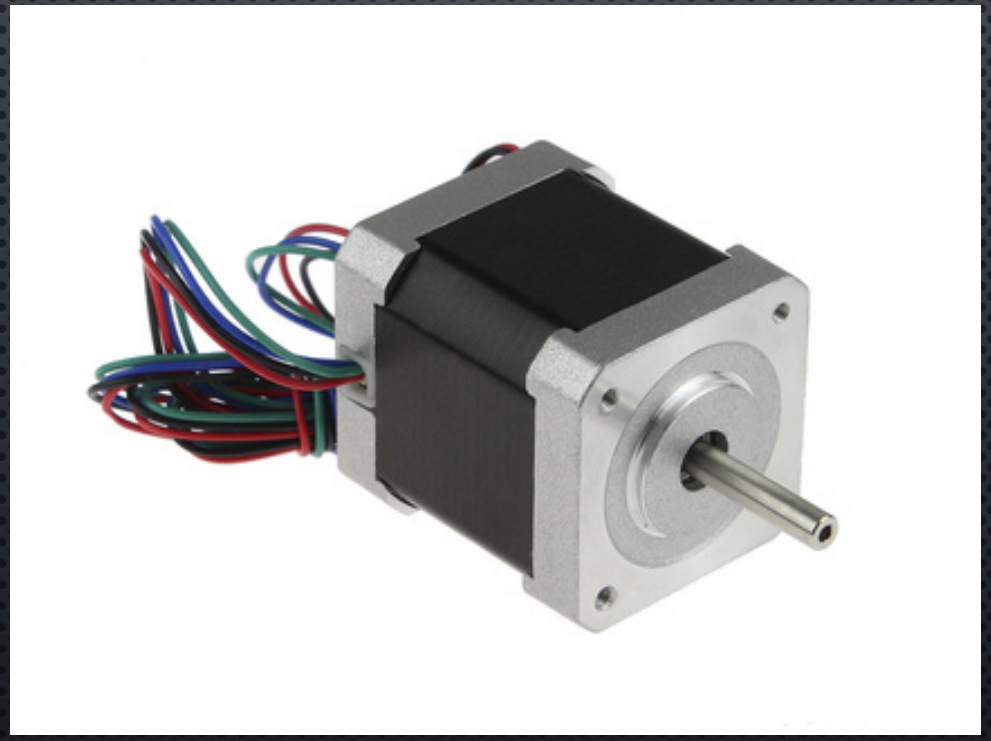
SPRING LOADING STRUCTURE





STRUCTURAL ELEMENTS

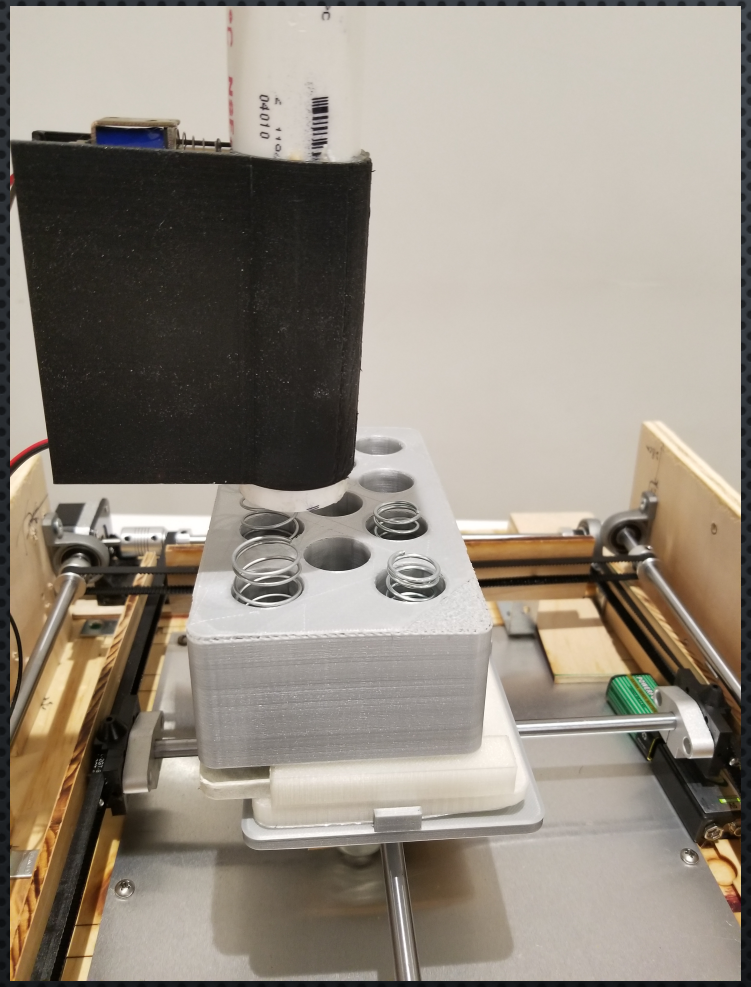
PULLEYS AND BELTS

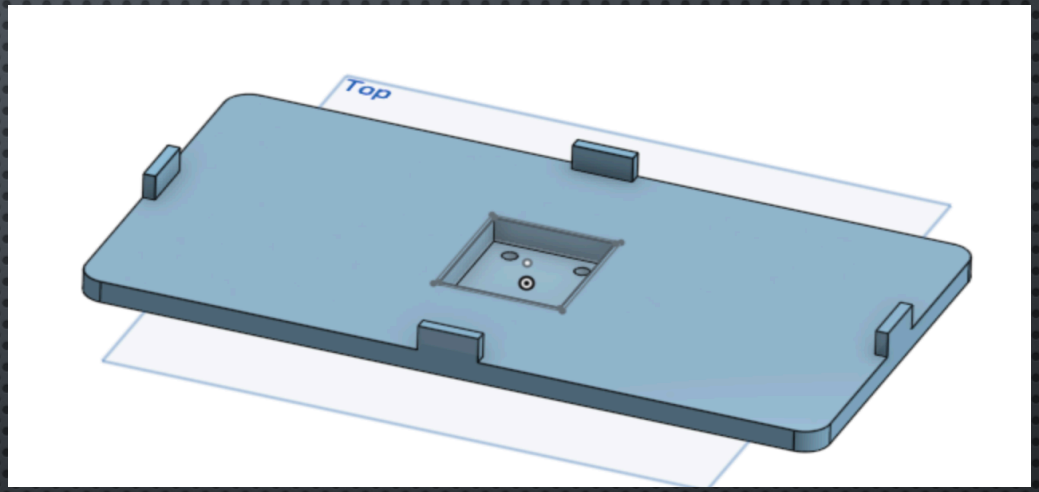


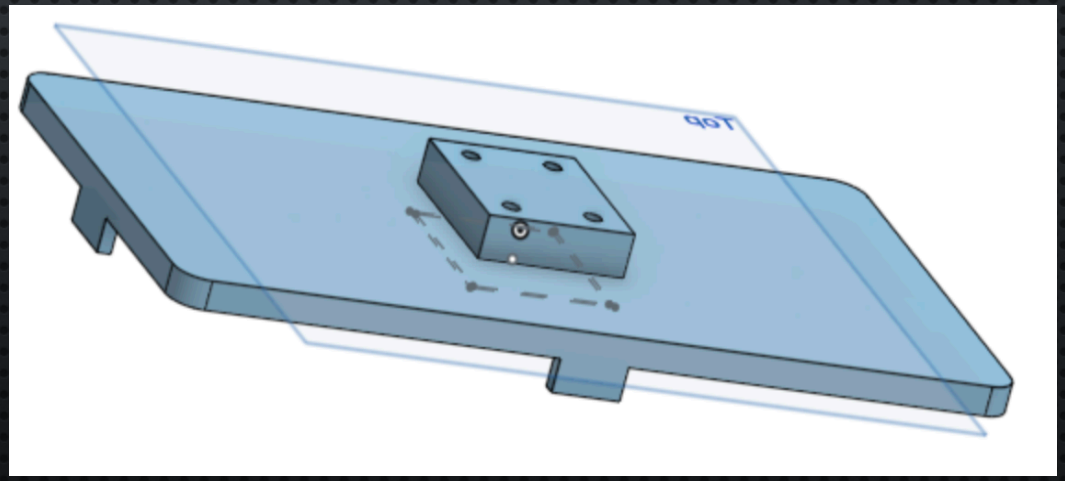
NEMA 17 BIPOLAR STEPPER MOTOR

BELT TENSIONER

STEPPER MOTOR INTERFACE







ARDUINO CODE FOR PRECISE STEPPING

STEPPER_XY_GRID

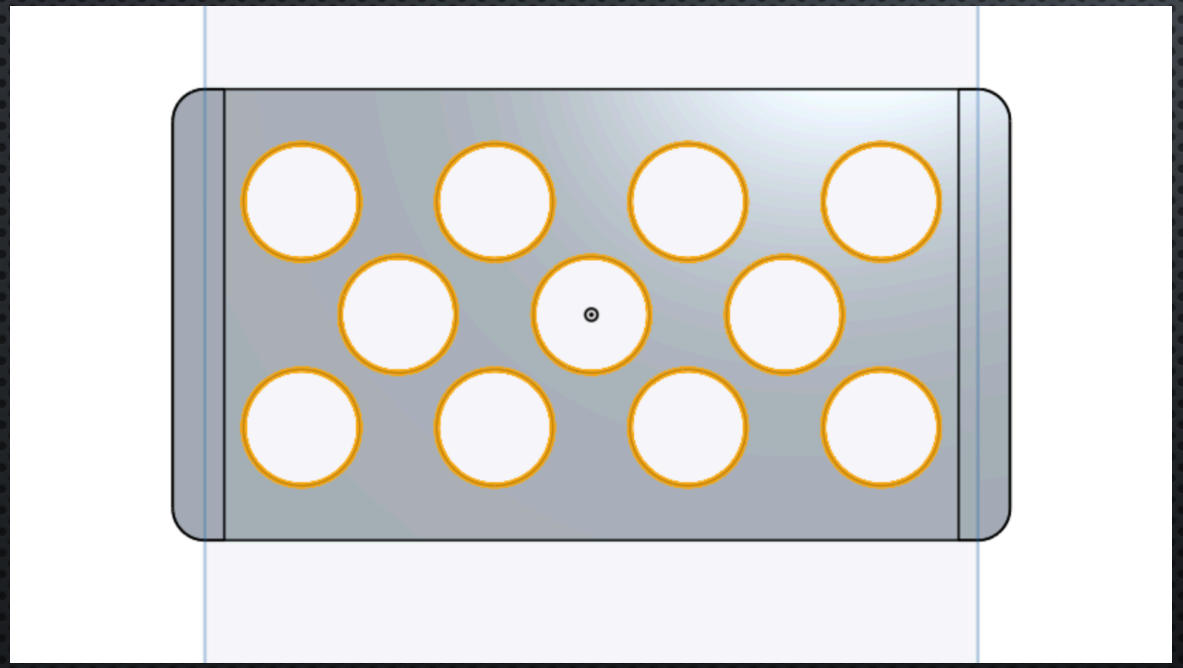
```
#include <AFMotor.h>

AF_Stepper motorXX(200,1);
AF_Stepper motorYY(200,2);

void setup() {
  // put your setup code here, to run once:
  Serial.begin(9600);
  Serial.println("Stepper test!");

  motorXX.setSpeed(40);  // 30 RPM
  motorYY.setSpeed(40); // 30 RPM
  delay(1000);
}

void loop() {
  // put your main code here, to run repeatedly:

  // MOVE TO START //
  motorXX.step(1183, FORWARD, DOUBLE);  // 6 REVOLUTIONS
  delay(100);
  motorYY.step(1623, FORWARD, DOUBLE);  // 8.1 REVOLUTIONS
  delay(1000);

  //  [ROW 1] //

 // STEP 1: ACTUATE SOLENOIDS,  DROP SPRING IN HOLE 1


//MOVE TO NEXT HOLE
  motorXX.step(484, FORWARD, DOUBLE); // 2.4 REVOLUTIONS
// DROP SPRING IN HOLE 2
  delay(1000);
```

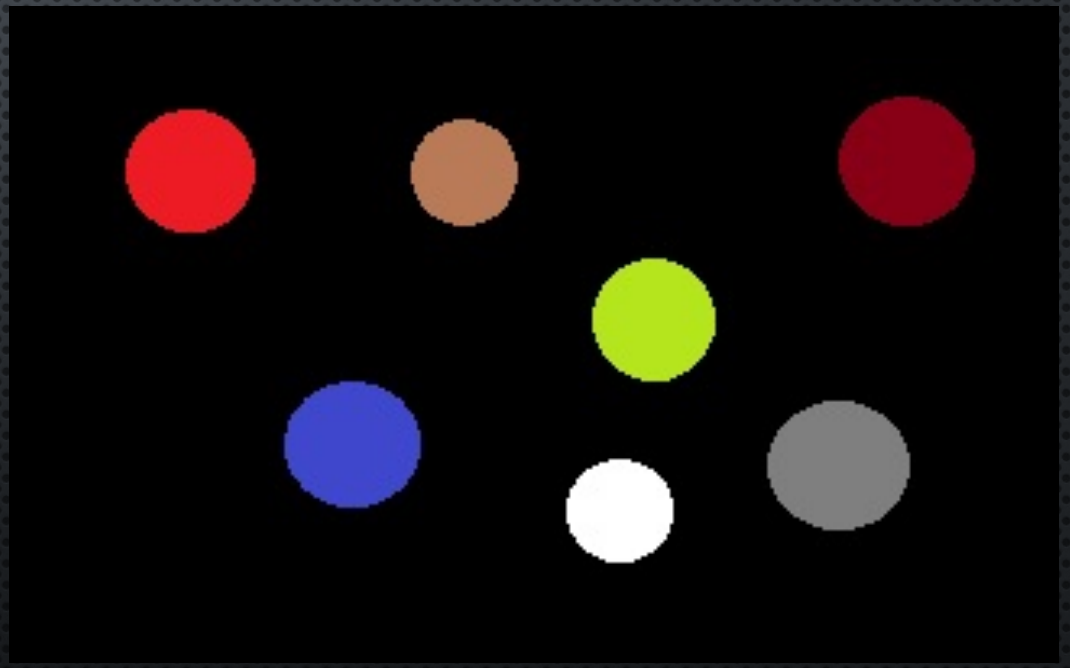
```
//MOVE TO NEXT HOLE
  motorXX.step(484, FORWARD, DOUBLE); // 2.4 REVOLUTIONS
// DROP SPRING IN HOLE 2
  delay(1000);


  // MOVE TO NEXT
  motorXX.step(484, FORWARD, DOUBLE);  // 2.4 REVOLUTIONS
  delay(1000);
  // DROP SPRING

  //MOVE TO NEXT
  motorXX.step(484, FORWARD, DOUBLE);  // 2.4 REVOLUTIONS
  delay(1000);
  // DROP SPRING

  //MOVE TO NEXT
  motorXX.step(484, FORWARD, DOUBLE);  // 2.4 REVOLUTIONS
  delay(1000);
  //DROP SPRING


//MOVE Y-AXIS MOTOR TO NEXT ROW
  motorYY.step(283, FORWARD, DOUBLE);  // 1.4 REVOLUTIONS
  delay(100);
  motorXX.step(243, BACKWARD, DOUBLE); // 1.2 REVOLUTIONS
  delay(1000);


// [ROW 2] //

 // STEP 2: ACTUATE SOLENOIDS,  DROP SPRING IN HOLE 1
```

```
// [ROW3] //

  //MOVE TO NEXT ROW
  motorXX.step(243, BACKWARD, DOUBLE);  // 1.2 REVOLUTIONS
  delay(100);
  motorYY.step(283, FORWARD, DOUBLE);   // 1.4 REVOLUTIONS
  delay(1000);
  // DROP SPRINGS

 //MOVE TO NEXT
  motorXX.step(484, FORWARD, DOUBLE);  // 2.4 REVOLUTIONS
// DROP SPRING IN HOLE 2
  delay(1000);

  // MOVE TO NEXT
  motorXX.step(484, FORWARD, DOUBLE);  // 2.4 REVOLUTIONS
  delay(1000);
  // DROP SPRING

  //MOVE TO NEXT
  motorXX.step(484, FORWARD, DOUBLE);  // 2.4 REVOLUTIONS
  delay(1000);
  // DROP SPRING

  //MOVE TO NEXT
  motorXX.step(484, FORWARD, DOUBLE);  // 2.4 REVOLUTIONS
  delay(1000);
  //DROP SPRING

  motorXX.release();
  delay(2000);
  motorYY.release();
  delay(2000);

}
```

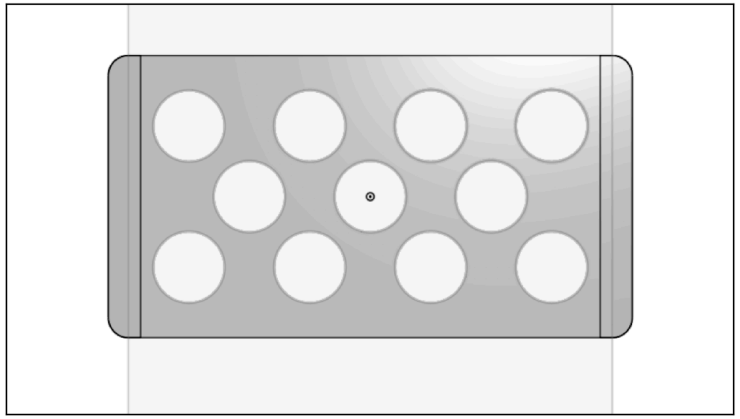
OPENCV TO DETECT EMPTY HOLES

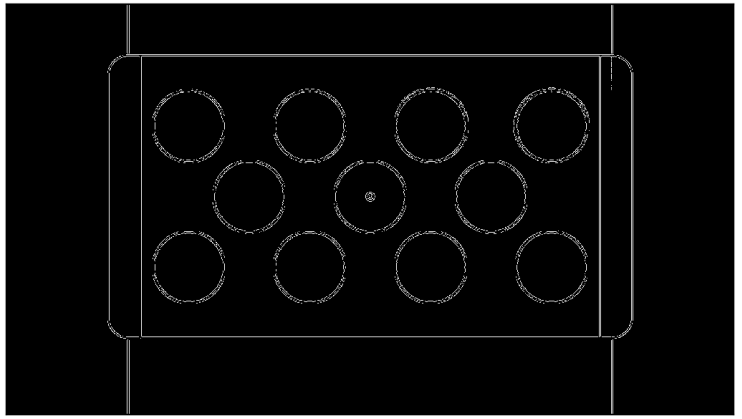
- SIMPLE BLOB DETECTOR WITH LOGITECH CAMERA
- PYTHON SCRIPT
- SERIAL COMMUNICATION WITH ARDUINO



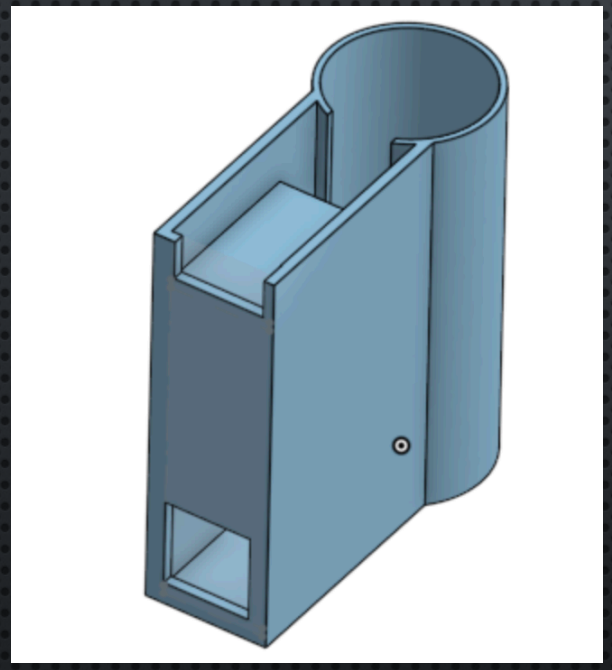


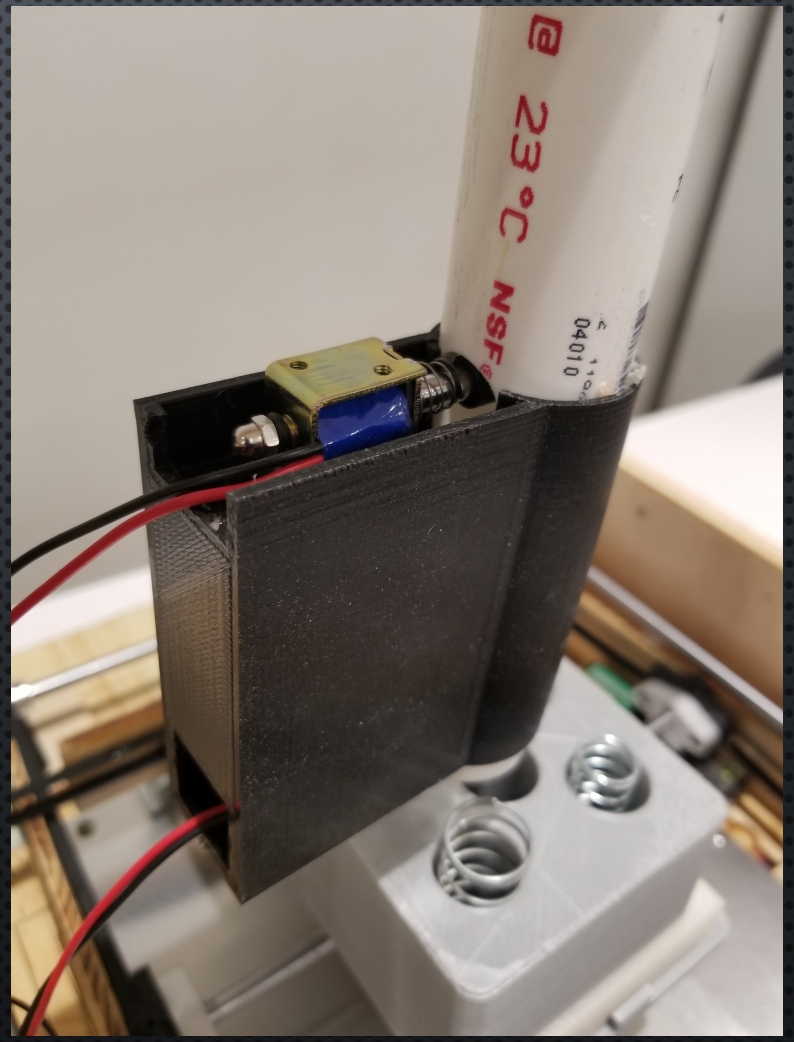
1.EDGE DETECTION
2. HOUGH CIRCLES FUNCTION





FEEDER TUBE FOR SPRINGS





NEXT STEPS-

1

1. Structural support for the tensioners to reduce stutter motion

2

2. Reducing friction on the bottom to ensure smooth motion

3

3. opencv interface with Arduino via serial comm

4

4. Cartridge feeding conveyor to keep up with production machine.