

TA212

MANUFACTURING PROCESSES II



GROUP NUMBER : 21

Instructor in - charge :
Shantanu Bhattacharya

Project Guide :
Dheeraj K. Soni

GROUP MEMBERS

Anmoldeep Singh Dhillon-220163

Kuldeepak Dhar Dwivedi-220558

Riyanshu Kumar-220904

Anukalp Rai-220182

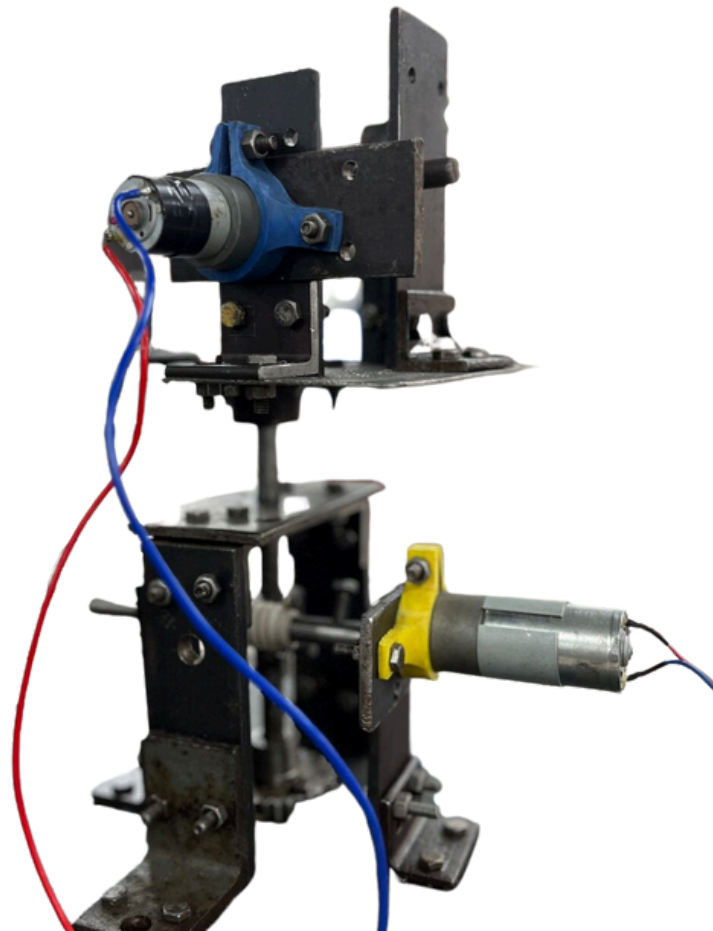
Shorya Agarwal-221023

Pranjali Singh-220796

Aditya Gautam-220064

Tanay Kesharwani-221122

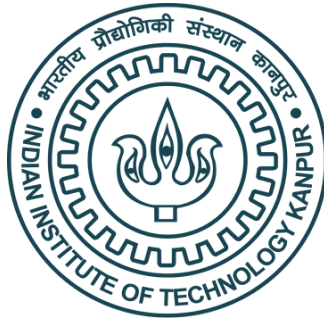
Chiranshu Kataria-220315



**FACE TRACKER USING 2
DEGREE FREEDOM
MECHANISM**

TA212

MANUFACTURING PROCESSES II



GROUP NUMBER : 21

Instructor in - charge :
Shantanu Bhattacharya

Project Guide :
Dheeraj K. Soni

GROUP MEMBERS

Anmoldeep Singh Dhillon-220163

Kuldeepak Dhar Dwivedi-220558

Riyanshu Kumar-220904

Anukalp Rai-220182

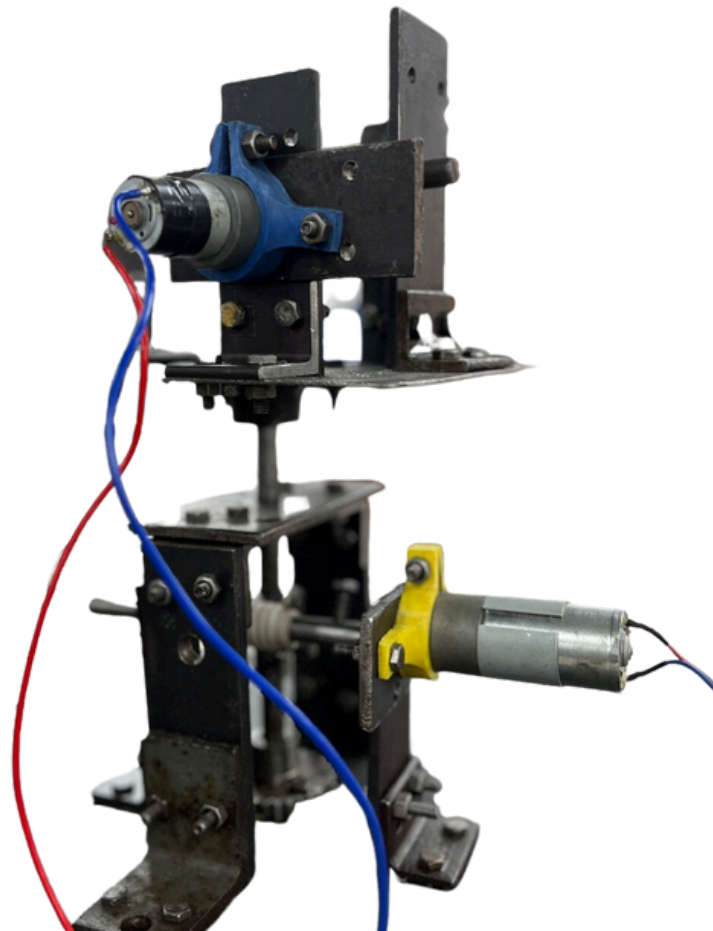
Shorya Agarwal-221023

Pranjali Singh-220796

Aditya Gautam-220064

Tanay Kesharwani-221122

Chiranshu Kataria-220315



**FACE TRACKER USING 2
DEGREE FREEDOM
MECHANISM**

TABLE OF CONTENTS

S. NO.	TOPIC
1	PROJECT OVERVIEW
2.	ISOMETRIC VIEW
3..	LIST OF PARTS
4.	DETAILED PART DRAWINGS
5.	MOTOR AND GEAR CALCULATIONS
6.	ARDUINO CODE AND FACE TRACKING CODE
7.	COST ANALYSIS

PROJECT OVERVIEW

Project Objective	The objective of our project is to track human faces using a mechanism that can move in two degrees of freedom.
Manufactured Parts	We have manufactured 10 different parts including worm, worm wheel, disc, U support, L- joints, motor holder, phone holder.
Cost Analysis	The cost of the project is ₹18,975.
Improvements that can be made	We can aim at any dynamic target or select any fix target by improving face detection code. Also, it can be attached on machine artillery to aim perfectly.

LIST OF PARTS

Part No.	Part Name	Quantity	Material Used	Manufactured/ Bought	Manufacturing Technique
1	Base Plate	1	Mild Steel	Manufactured	Cutting, Drilling
2	Top Plate	1	Mild Steel	Manufactured	Cutting, Drilling
3	L-Joints	8	Mild Steel	Manufactured	Cutting, Drilling
4	Support (68x50mm)	2	Mild Steel	Manufactured	Cutting, Drilling
5	Disc	2	Mild Steel	Manufactured	Milling, Drilling
6	Support (120x50mm)	4	Mild Steel	Manufactured	Cutting, Drilling
7	U Support	1	Mild Steel	Manufactured	Cutting, Drilling
8	Axle 12.70d 120mm	2	Mild Steel	Manufactured	Cutting
9	Worm and Worm Wheel	2	Mild Steel and PVC	Manufactured	Turning, Milling
10	Phone Holder	1	Plastic	Manufactured	3D Printing
11	Motor Holder	2	Plastic	Manufactured	3D Printing

ARDUINO CODE

```
int PWM_LOWER = 5; //Lower motor
// int PWM_UPPER = 6 //Upper motor
int INA_LOWER = 8;
int INB_LOWER = 7;
int PWM_UPPER = 6; //Upper motor
int INA_UPPER = 12;
int INB_UPPER = 13;
// int INA_UPPER =
// int INB_UPPER =
int x;
int y;
int prevX;
int prevY;
int speedX;
int speedY;
void setup()
{
  Serial.begin(9600);
  // servoVer.attach(9); //Attach Vertical Servo to Pin 5
  // servoHor.attach(10); //Attach Horizontal Servo to Pin 6
  // servoVer.write(90);
  // servoHor.write(90);
  pinMode(PWM_LOWER,OUTPUT);
  pinMode(INA_LOWER,OUTPUT);
  pinMode(INB_LOWER,OUTPUT);
}
void Pos()
{
  if(prevX != x || prevY != y)
  {
    speedX = x-320;
    speedY = y-240;
  }
  else{
    speedX = 0;
    speedY= 0;
  }
  if(speedX<10 || speedX>-10){
    speedX=0;
  }
  if(speedY<5 || speedY>-5){
    speedY=0;
  }
}
```

```

    prevX=x;
    prevY=y;
    // Serial.Print(speedX);
    // Serial.Printnl;
}
void loop()
{
    if(Serial.available() > 0)
    {
        if(Serial.read() == 'X')
        {
            x = Serial.parseInt();
            if(Serial.read() == 'Y')
            {
y = Serial.parseInt();
            Serial.print(x);
            Serial.print(" ");
            Serial.print(y);
            Serial.println();
            //Pos();
            if(x-320>0){
            analogWrite(PWM_LOWER,x-320);
            digitalWrite(INA_LOWER,HIGH);
            digitalWrite(INB_LOWER,LOW);
            }
            else{
            analogWrite(PWM_LOWER,fabs(x-320));
            digitalWrite(INA_LOWER,LOW);
            digitalWrite(INB_LOWER,HIGH);
            }

            if(y-240>0){
            analogWrite(PWM_UPPER,y-240);
            digitalWrite(INA_UPPER,HIGH);
            digitalWrite(INB_UPPER,LOW);
            }
            else{
            analogWrite(PWM_UPPER,fabs(y-240));
            digitalWrite(INA_UPPER,LOW);
            digitalWrite(INB_UPPER,HIGH);
            }
            //delay(100);
        }
        }
        // while(Serial.available() > 0)
        // {
        //   Serial.read();
        // }
    }
}

```

FACE TRACKING CODE

```
import cv2
from cvzone.FaceDetectionModule import FaceDetector

# import pyfirmata
import numpy as np
import serial

cap = cv2.VideoCapture(0)
address= "http://192.168.11.109:8080/video"
cap.open(address)
ws, hs = 1280, 720
cap.set(3, ws)
cap.set(4, hs)

if not cap.isOpened():
    print("Camera couldn't Access!!!")
    exit()
ArduinoSerial=serial.Serial('com6',9600,timeout=0.1)

# port = "COM7"
# board = pyfirmata.Arduino(port)
# servo_pinX = board.get_pin('d:9:s') #pin 9 Arduino
# servo_pinY = board.get_pin('d:10:s') #pin 10 Arduino

detector = FaceDetector()
servoPos = [90, 90] # initial servo position

while True:
    success, img = cap.read()
    img, bboxes = detector.findFaces(img, draw=False)

    if bboxes:
        #get the coordinate
        fx, fy = bboxes[0]["center"][0], bboxes[0]["center"][1]
        pos = [fx, fy]
        #convert coordinat to servo degree
        servoX = np.interp(fx, [0, ws], [0, 180])
        servoY = np.interp(fy, [0, hs], [0, 180])
```



```

# if servoX < 0:
    #     servoX = 0
    # elif servoX > 180:
    #     servoX = 180
    # if servoY < 0:
    #     servoY = 0
    # elif servoY > 180:
    #     servoY = 180

string='X{0:d}Y{1:d}'.format(fx,fy)

servoPos[0] = servoX
servoPos[1] = servoY
print(string)
ArduinoSerial.write(string.encode('utf-8'))

cv2.circle(img, (fx, fy), 80, (0, 0, 255), 2)
cv2.putText(img, str(pos), (fx+15, fy-15), cv2.FONT_HERSHEY_PLAIN,
2, (255, 0, 0), 2 )
cv2.line(img, (0, fy), (ws, fy), (0, 0, 0), 2) # x line
cv2.line(img, (fx, 0), (fx, hs), (0, 0, 0), 2) # y line
cv2.circle(img, (fx, fy), 15, (0, 0, 255), cv2.FILLED)
cv2.putText(img, "TARGET LOCKED", (850, 50), cv2.FONT_HERSHEY_PLAIN,
3, (255, 0, 255), 3 )

else:
    cv2.putText(img, "NO TARGET", (880, 50), cv2.FONT_HERSHEY_PLAIN, 3,
(0, 0, 255), 3)
    cv2.circle(img, (640, 360), 80, (0, 0, 255), 2)
    cv2.circle(img, (640, 360), 15, (0, 0, 255), cv2.FILLED)
    cv2.line(img, (0, 360), (ws, 360), (0, 0, 0), 2) # x line
    cv2.line(img, (640, 0), (640, hs), (0, 0, 0), 2) # y line

    cv2.putText(img, f'Servo X: {int(servoPos[0])} deg', (50, 50),
cv2.FONT_HERSHEY_PLAIN, 2, (255, 0, 0), 2)
    cv2.putText(img, f'Servo Y: {int(servoPos[1])} deg', (50, 100),
cv2.FONT_HERSHEY_PLAIN, 2, (255, 0, 0), 2)

# servo_pinX.write(servoPos[0])
# servo_pinY.write(servoPos[1])

cv2.imshow("Image", img)
cv2.waitKey(1)

```

