

Programme: Higher Diploma in Artificial Intelligence and Robotics

Programme code: EG114728

Module: AI and Programming

Module Code: MBS 3523

Assessment: Assignment 1 (Part c)

Official Use		
	Full Mark	Mark
Total	60%	

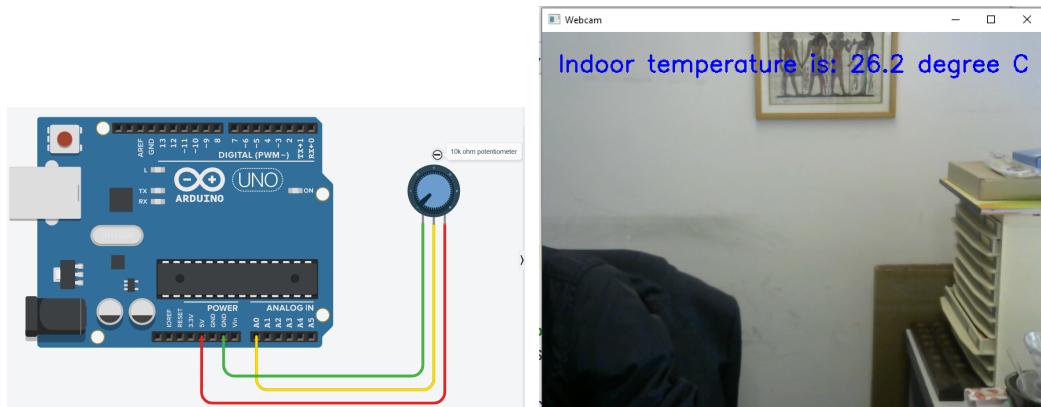
Submission deadline: 11 Mar 2025, 9pm

Name: Li Sing Fai	Student no.:240241059
-------------------	-----------------------

Assignment 1 (Part c)

Question 1 (20%)

Connect ONE sensor (e.g. light sensor, temperature sensor) to your Arduino board and write a code to send the measurement result to a PC running a Python code. The sensing signal should be displayed on a Window while the webcam is streaming.



Requirements:

Save the code on your Github as **MBS3523-A1CQ1_ChantaiMan.py**

Print the codes and paste onto this paper

Screen capture a screen with your own set up (Arduino +sensor).

NOTE: You may use a potentiometer to simulate a sensing signal

Paste your Python code here:

```
import serial
import cv2
import time

ser = serial.Serial('COM22', baudrate=9600, timeout=1)
time.sleep(3)

cap = cv2.VideoCapture(0)

temperature = "N/A"
humidity = "N/A"

while True:
    if ser.in_waiting > 0:
        data = ser.readline().decode('utf-8').strip()
        print(f"Received data: {data}")

    try:
        parts = data.split(' ')
        humidity = parts[1]
        temperature = parts[3]
        print(f"Parsed - Temperature: {temperature}, Humidity: {humidity}")
    except (IndexError, ValueError) as e:
        print(f"Error parsing data: {e}")
        temperature = "N/A"
        humidity = "N/A"

    ret, frame = cap.read()
    if not ret:
        print("Failed to grab frame")
        break

    cv2.putText(frame, f'Temperature: {temperature}', (10, 30), cv2.FONT_HERSHEY_SIMPLEX, 1, (255, 0, 0), 2)
    cv2.putText(frame, f'Humidity: {humidity}', (10, 70), cv2.FONT_HERSHEY_SIMPLEX, 1, (255, 0, 0), 2)

    cv2.imshow('Temperature and Humidity', frame)

    if cv2.waitKey(1) & 0xFF == 27:
        break

cap.release()
ser.close()
cv2.destroyAllWindows()
```

Paste your Arduino code here:

```
#include "DHT.h"

#define DHTPIN 6
#define DHTTYPE DHT22
DHT dht(DHTPIN, DHTTYPE);

void setup() {
    Serial.begin(9600);
    dht.begin();
}

void loop() {
    delay(2000);

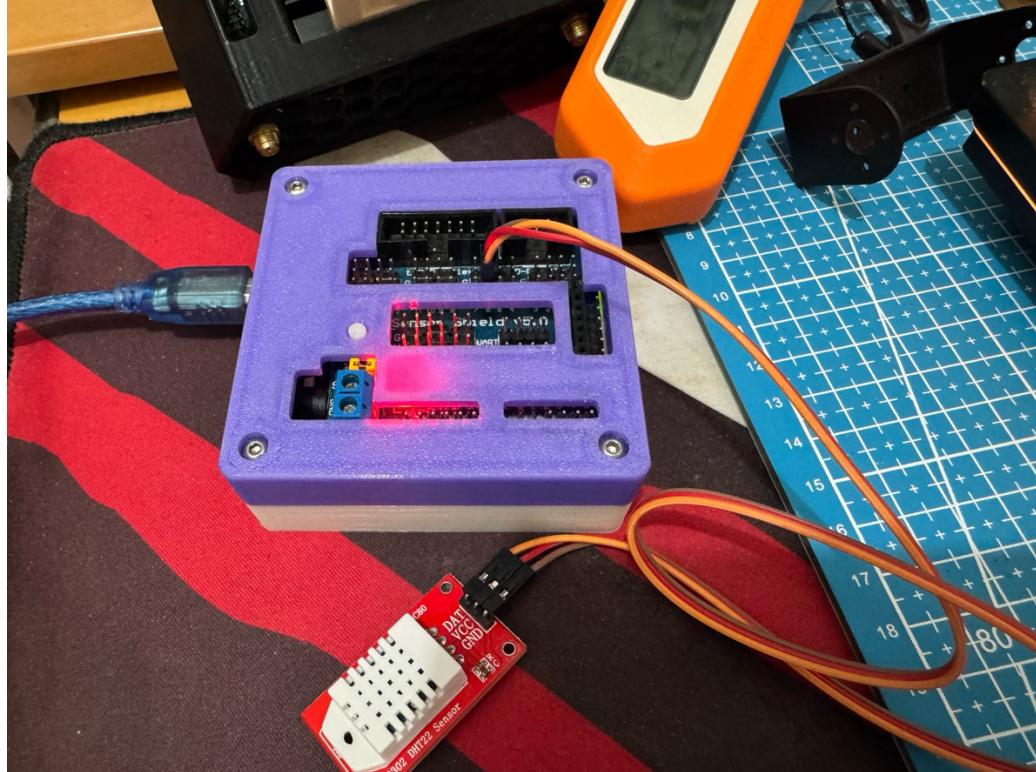
    float h = dht.readHumidity();
    float t = dht.readTemperature();

    if (isnan(h) || isnan(t)) {
        Serial.println(F("Failed to read from DHT sensor!"));
        return;
    }

    Serial.print(F("Humidity: "));
    Serial.print(h);
    Serial.print(F("% Temperature: "));
    Serial.print(t);
    Serial.println(F("Degree C"));

    if (Serial.available()) {
        String input = Serial.readStringUntil('\r');
        input.trim();
    }
}
```

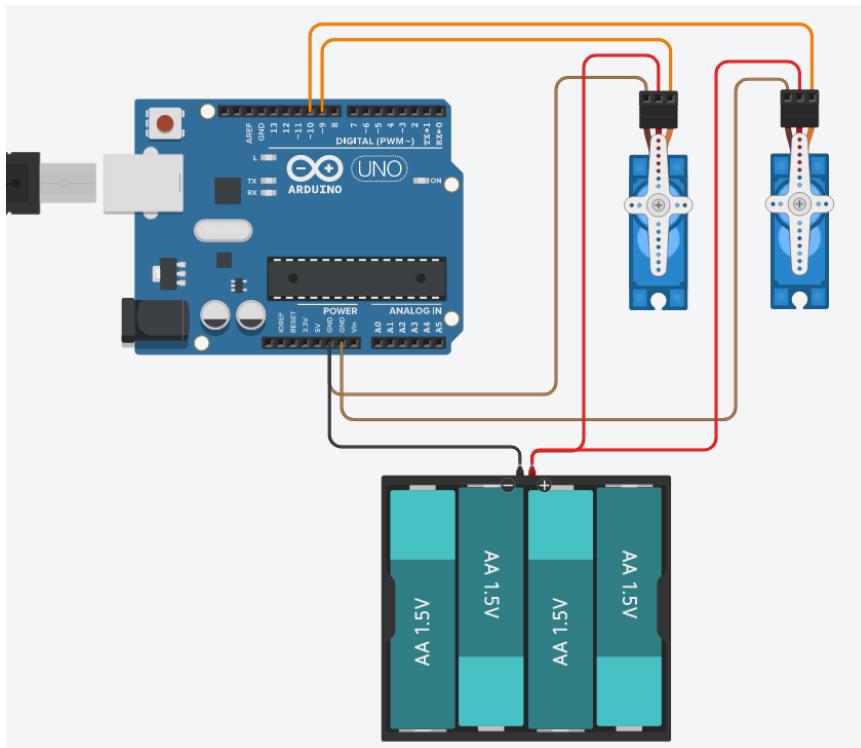
Paste your hardware connection photo here:

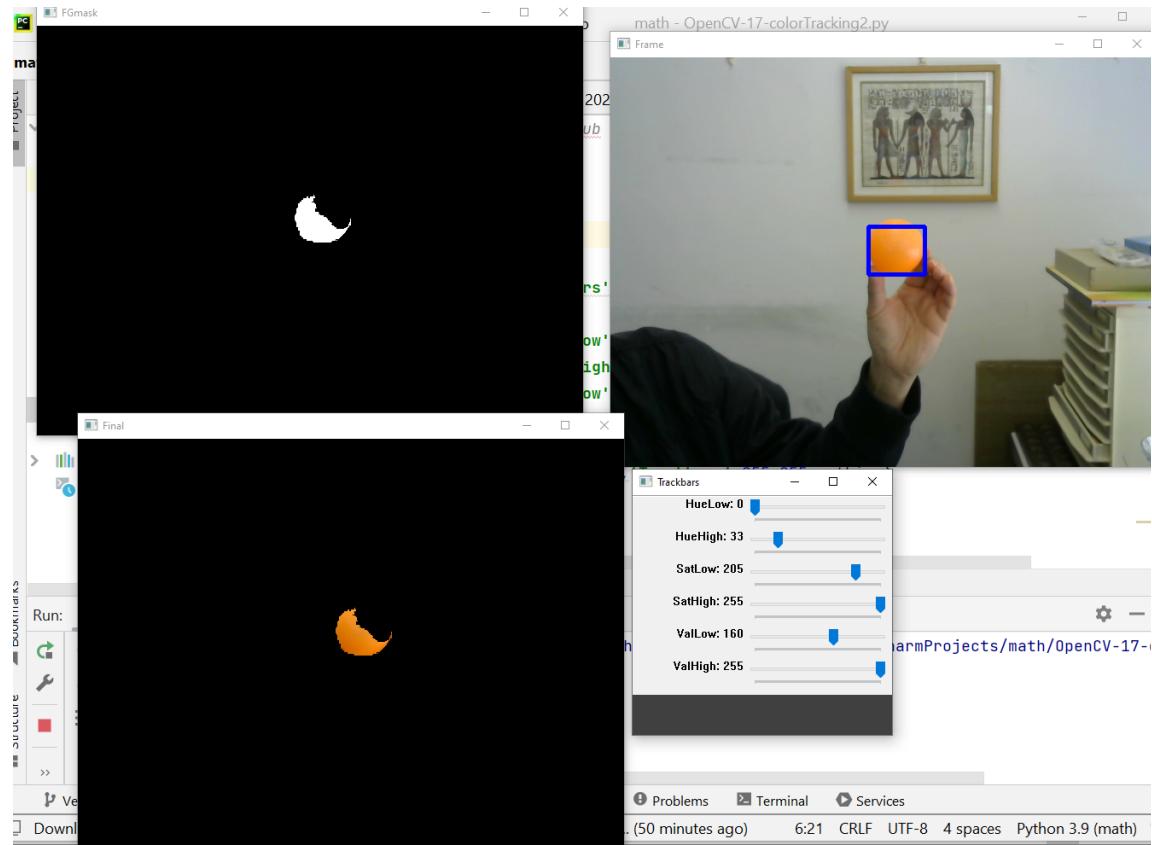


```
servo test.py colorTracking.py Q1.py Q2.py
1 import serial
2 import cv2
3 import time
4
5 ser = serial.Serial( port='COM3', baudrate=9600, timeout=1)
6 time.sleep(3)
7
8 cap = cv2.VideoCapture(0)
9
10 temperature = "N/A"
11 humidity = "N/A"
12
13 while True:
14     if ser.in_waiting > 0:
15         data = ser.readline().decode('utf-8').strip()
16         print(f"Received data: {data}")
17
18     try:
19         parts = data.split(' ')
20         humidity = parts[1]
21         temperature = parts[3]
22         print(f"Parsed - Temperature: {temperature}, Humidity: {humidity}")
23     except (IndexError, ValueError) as e:
24         print(f"Error parsing data: {e}")
25         temperature = "N/A"
26         humidity = "N/A"
27
28     ret, frame = cap.read()
29     if not ret:
30         print("Failed to grab frame")
31         break
32
33     cv2.putText(frame, text=f'Temperature: {temperature}', org=(10, 30), cv2.FONT_HERSHEY_SIMPLEX, fontScale: 1, color: (255, 0, 0), thickness: 2)
34     cv2.putText(frame, text=f'Humidity: {humidity}', org=(10, 70), cv2.FONT_HERSHEY_SIMPLEX, fontScale: 1, color: (255, 0, 0), thickness: 2)
35
36     cv2.imshow(winname='Temperature and Humidity', frame)
37
38     if cv2.waitKey(1) & 0xFF == 27:
39         break
40
41 cap.release()
42 ser.close()
43 cv2.destroyAllWindows()
```

Question 2 (40%)

Select a color ball (or any single color object) and track it with a pan-tilt servo assembly with a webcam mounted onto it.





NOTE:

You need to connect your Arduino Uno board with two servos and a webcam to the PC/notebook computer. After tuning your code, the servo/webcam combo should track the selected color when it is within the FOV (field of view).

```
# Other Requirements:  
# Save the code on your Github as MBS3523-A1CQ2_ChantaiMan.py  
# Print the codes and attach to this paper  
# Take a picture with the object, the code running and the  
object is being tracked.  
# NOTE: If you are unable to drive two servos, you may  
submit your result with only PAN servo operating (mark  
reduction is 10%).
```

Paste your Python code here:

```
import serial
import numpy as np
import cv2
import time

ser = serial.Serial('COM23', baudrate=115200, timeout=1)
time.sleep(2)

pos = 90

def nothing(x):
    pass

cv2.namedWindow('Trackbars')
cv2.createTrackbar('HueLow', 'Trackbars', 100, 179, nothing)
cv2.createTrackbar('HueHigh', 'Trackbars', 140, 179, nothing)
cv2.createTrackbar('SatLow', 'Trackbars', 150, 255, nothing)
cv2.createTrackbar('SatHigh', 'Trackbars', 255, 255, nothing)
cv2.createTrackbar('ValLow', 'Trackbars', 50, 255, nothing)
cv2.createTrackbar('ValHigh', 'Trackbars', 255, 255, nothing)

cam = cv2.VideoCapture(3)

while True:
    success, img = cam.read()
    if not success:
        print("Failed to capture image")
        break

    hsv = cv2.cvtColor(img, cv2.COLOR_BGR2HSV)

    hueLow = cv2.getTrackbarPos('HueLow', 'Trackbars')
    hueHigh = cv2.getTrackbarPos('HueHigh', 'Trackbars')
    satLow = cv2.getTrackbarPos('SatLow', 'Trackbars')
    satHigh = cv2.getTrackbarPos('SatHigh', 'Trackbars')
    valLow = cv2.getTrackbarPos('ValLow', 'Trackbars')
    valHigh = cv2.getTrackbarPos('ValHigh', 'Trackbars')

    FGmask = cv2.inRange(hsv, (hueLow, satLow, valLow), (hueHigh, satHigh, valHigh))
    cv2.imshow('FGmask', FGmask)

    print(f'HueLow: {hueLow}, HueHigh: {hueHigh}, SatLow: {satLow}, SatHigh: {satHigh}, ValLow: {valLow}, ValHigh: {valHigh}')

    contours, hierarchy = cv2.findContours(FGmask, cv2.RETR_EXTERNAL, cv2.CHAIN_APPROX_SIMPLE)
```

```
contours = sorted(contours, key=lambda x: cv2.contourArea(x), reverse=True)

print(f'Contours detected: {len(contours)}')

if contours:

    cnt = contours[0]
    area = cv2.contourArea(cnt)
    if area > 100:
        (x, y, w, h) = cv2.boundingRect(cnt)
        cv2.rectangle(img, (x, y), (x + w, y + h), (255, 0, 0), 3)

    errorPan = (x + w / 2) - (640 / 2)

    if abs(errorPan) > 20:
        pos = pos - errorPan / 200 # 調整方向

    pos = max(10, min(pos, 170))

    servoPos = str(int(pos)) + '\r'
    ser.write(servoPos.encode())
    print(f'Servo Position: {pos}')

    cv2.imshow('Frame', img)

    if cv2.waitKey(1) == 27:
        break
cam.release()
cv2.destroyAllWindows()
```

Paste your Arduino code here:

```
#include <Servo.h>

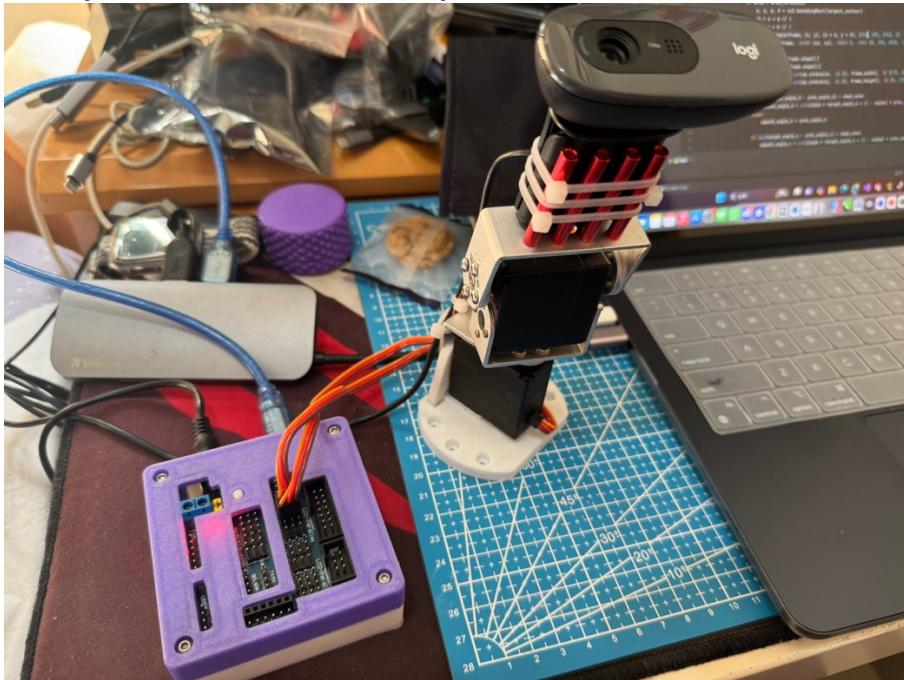
Servo myServo;
Servo myServo2;
int servoPin = 10;
int servoPin2 = 11;
String servoPos;
int pos;

void setup() {
    Serial.begin(115200);
    myServo.attach(servoPin);
    myServo2.attach(servoPin2);
    myServo.write(90);
    myServo2.write(90);
}

void loop() {
    if (Serial.available() > 0) {
        servoPos = Serial.readStringUntil('\r');
        pos = servoPos.toInt();
        myServo.write(pos);
        delay(50);

        if (Serial.available() > 0) {
            servoPos = Serial.readStringUntil('\r');
            pos = servoPos.toInt();
            myServo2.write(pos);
            delay(50);
        }
    }
}
```

Paste your hardware connection photo here:



Paste your screen capture photo here:

