

## Major project

### Arduino board

```
#include <Servo.h>

// Declare constants for pins
const int in = 2;
const int out = 3;
const int buzz = 4;
const int led = 5;

// Initialize variables
int temp = 0;
int count = 0;
String a = "";

// Create servo objects for controlling
motors
Servo servo1;
Servo servo2;
Servo servo3;
Servo servo4;

void setup()
{
    // Start serial communication
    Serial.begin(9600);

    // Set input and output modes for pins
    pinMode(in, INPUT);
    pinMode(out, INPUT);
    pinMode(buzz, OUTPUT);
}
```

```
pinMode(led, OUTPUT);

// Attach servo motors to their respective pins
servo1.attach(6);
servo2.attach(9);
servo3.attach(10);
servo4.attach(11);

// Set initial position for servo motors
servo1.write(180);
servo2.write(180);
servo3.write(180);
servo4.write(180);
}

void loop()
{
    // Print temperature to serial monitor
    Serial.println(temp);

    // Wait for 1 second
    delay(1000);

    // Check if input or output button is pressed
    if (digitalRead(in) == 0)
    {
        // If input button is pressed, increment temperature
        temp = temp + 1;
    }
}
```

```

    // Wait for 2 seconds to prevent multiple
inputs
    delay(2000);
}
else if (digitalRead(out) == 0)
{
    // If output button is pressed, decrement
temperature
    temp = temp - 1;

    // If temperature goes below 0, set it to
0
    if (temp < 0)
    {
        temp = 0;
    }

    // Wait for 2 seconds to prevent multiple
inputs
    delay(2000);
}

// Check for any incoming messages from
serial monitor
while (Serial.available())
{
    a = Serial.readString();
}

// If message is "lift"
if (a == "lift")
{

```

```
// If count is less than 5, turn off
buzzer and LED, and increment count
if (count < 5)
{
    digitalWrite(buzz, LOW);
    digitalWrite(led, LOW);
    count += 1;

    // Wait for 1 second to prevent
multiple inputs
    delay(1000);
}
// If count is greater than or equal to 5
else if (count >= 5)
{
    // Turn on buzzer and LED for 10
seconds
    digitalWrite(buzz, HIGH);
    digitalWrite(led, HIGH);
    delay(10000);

    // Move servo motors to 0 degree
position
    servo1.write(0);
    servo2.write(0);
    servo3.write(0);
    servo4.write(0);
}
}
}
```

## Python

```
import cv2 as cv # importing OpenCV library
import numpy as np
import telegram # importing Telegram library
import time # importing time library
import requests # importing requests library
for sending requests to APIs
import serial # importing pyserial library
for serial communication with Arduino

ser = serial.Serial('COM8',9600) # setting up
serial communication with Arduino board
time.sleep(2) # wait for 2 seconds to
establish the connection

videoCapture = cv.VideoCapture(0) # capturing
video from the default webcam
prevCircle = None # setting previous circle
to None
dist = lambda x1,y1,x2,y2: (x1-x2)**2 + (y1-
y2)**2 # defining distance function using
lambda function

while True: # loop for video capturing and
processing
    ret, frame = videoCapture.read() #
reading frame from the webcam
    if not ret: break # if there is no frame,
break the loop
```

```
    grayFrame = cv.cvtColor(frame,
cv.COLOR_BGR2GRAY) # converting the frame to
grayscale
    blueFrame = cv.GaussianBlur(grayFrame,
(17,17), 0) # blurring the frame using
GaussianBlur function

    circles = cv.HoughCircles(blueFrame,
cv.HOUGH_GRADIENT, 1.2, 100,param1=100,
param2=30, minRadius=75, maxRadius=400) #
detecting circles using HoughCircles function

    if circles is not None: # if circles are
detected
        circles =
np.uint16(np.around(circles)) # rounding the
coordinates and radius of circles
        circles1 =
np.round(circles[0,:]).astype("int") #
extracting the coordinates and radius of
circles and converting them to integers
        global no # declaring the number of
circles as global variable
        no = len(circles1) # counting the
number of circles detected
        print(no) # printing the number of
circles detected
        b = ser.readline() # reading the
input from the Arduino board
        print(b.decode()) # decoding and
printing the input from the Arduino board
```

```

        if no < int(b): # if the number of
circles detected is less than the input from
the Arduino board
            ser.write("lift".encode()) # send
the command to the Arduino board to lift the
mesh
            time.sleep(10) # wait for 10
seconds
            telegram =
"https://api.telegram.org/bot6109980937:AAHwP
q-T-
xJTEcMmm9hKXmopeLBvKwMyQag/sendMessage?chat_i
d=1102735525&text="+ "Alert! Alert!! Alert!!!
\nSomeone is drowning" # creating the message
for Telegram API
            requests.post(telegram) # sending
the message to the specified chat ID using
Telegram API
            chosen = None # setting chosen circle
to None
            for i in circles[0,:]: # loop for
iterating through all detected circles
                if chosen is None: chosen = i #
If chosen circle is None, set chosen circle
to the first detected circle
                if prevCircle is not None:
                    if dist(chosen[0], chosen[1],
prevCircle[0], prevCircle[1]) > dist(i[0],
i[1], prevCircle[0], prevCircle[1]):
                        chosen = i # if the
distance between the current circle and
previous circle is less than the distance

```

```
between the chosen circle and previous  
circle, set chosen circle to current circle  
    cv.circle(frame,(chosen[0],chosen[1])  
, 1, (0,100,
```