```python
import cv2
import dlib
from imutils.video import VideoStream
from threading import Thread
import time
import winsound
import imutils
from scipy.spatial import distance as dist
from imutils import face_utils
import serial

alarm_sound_path = "C:\\Windows\\Media\\Alarm01.wav"
shape_predictor_path = "C:\\Users\\DELL\\Desktop\\New folder\\myrevanth\\shape_predictor_68_face_landmarks.dat"

try:
    ser = serial.Serial('COM11', 9600)  # Change COM11 to your Arduino port
except serial.SerialException as e:
    print(f"Error opening serial port: {e}")
    ser = None

def calculate_EAR(eye):
    A = dist.euclidean(eye[1], eye[5])
    B = dist.euclidean(eye[2], eye[4])
    C = dist.euclidean(eye[0], eye[3])
    EAR = (A + B) / (2.0 * C)
    return EAR

def play_sound(path):
    winsound.PlaySound(path, winsound.SND_FILENAME)

EAR_threshold = 0.25
EAR_consec_frames = 50
COUNTER = 0
ALARM_ON = False

print("[INFO] loading facial landmark predictor...")
detector = dlib.get_frontal_face_detector()
predictor = dlib.shape_predictor(shape_predictor_path)

(left_Start, left_End) = face_utils.FACIAL_LANDMARKS_IDXS["left_eye"]
(right_Start, right_End) = face_utils.FACIAL_LANDMARKS_IDXS["right_eye"]

print("[INFO] starting video stream thread...")
vs = VideoStream(src=0).start()
time.sleep(1.0)

while True:
    frame = vs.read()
    frame = imutils.resize(frame, width=450)
    gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)
    rects = detector(gray, 0)

    for rect in rects:
        shape = predictor(gray, rect)
        shape = face_utils.shape_to_np(shape)

        leftEye = shape[left_Start:left_End]
        rightEye = shape[right_Start:right_End]
        leftEAR = calculate_EAR(leftEye)
        rightEAR = calculate_EAR(rightEye)

        EAR = (leftEAR + rightEAR) / 2.0

        if EAR < EAR_threshold:
            COUNTER += 1
            if COUNTER >= EAR_consec_frames and not ALARM_ON:
                ALARM_ON = True
                play_sound(alarm_sound_path)
                # Additional actions when the alarm is triggered
                if ser:
                    ser.write(b'1')  # Send '1' to Arduino
        else:
            COUNTER = 0
            ALARM_ON = False
            if ser:
                ser.write(b'0')  # Send '0' to Arduino when not drowsy

        # Additional actions if needed based on EAR value

    cv2.imshow("Frame", frame)
    key = cv2.waitKey(1) & 0xFF
    if key == ord("e"):
        break

# Closing the serial port if it was opened
if ser:
    ser.close()

cv2.destroyAllWindows()
vs.stop()
```