import I2C\_LCD\_driver

import time

import os

import RPi.GPIO as GPIO

import smbus

os.system('modprobe w1-gpio')

os.system('modprobe w1-therm')

import threading #시간,os,GPIO,smbus 등 라즈베리파이 연결을 위한 라이브러리

from flask import Flask, request, jsonify, render\_template, Response #WAS서버 생성 라이브러리

from werkzeug.utils import redirect #FLASK에서 URL 파싱해서 이동시켜주는거

import Adafruit\_DHT

from PIL import ImageFont, ImageDraw, Image

import datetime

from picamera2 import Picamera2

import numpy as np

import cv2

from time import sleep

app = Flask(\_\_name\_\_)

GPIO.setwarnings(False)

GPIO.setmode(GPIO.BCM)

GPIO.setup(5, GPIO.OUT) #물순환모터

GPIO.setup(6, GPIO.OUT) #물순환모터

GPIO.setup(13, GPIO.OUT) #물공급(수중)모터

GPIO.setup(19, GPIO.OUT) #물공급(수중)모터

GPIO.setup(26, GPIO.OUT) #LED

servo\_pin = 21

GPIO.setup(servo\_pin, GPIO.OUT)

GPIO.setwarnings(False)  #모터 구동을 위한GPIO 핀번호 배치 및 설치

servo\_pwm = GPIO.PWM(servo\_pin, 50)

servo\_pwm.start(6.0)

motor\_pwm\_pin\_2 = 5 # 물순환모터 회전속도 조절

GPIO.setup(motor\_pwm\_pin\_2, GPIO.OUT)

motor\_pwm\_2 = GPIO.PWM(motor\_pwm\_pin\_2, 1000)

motor\_pwm\_pin\_3 = 13 # 물공급모터 회전속도 조절

GPIO.setup(motor\_pwm\_pin\_3, GPIO.OUT)

motor\_pwm\_3 = GPIO.PWM(motor\_pwm\_pin\_3, 1000)

#모터 속도제어

def set\_motor\_speed\_2(speed):

    motor\_pwm\_2.start(speed)

def set\_motor\_speed\_3(speed):

    motor\_pwm\_3.start(speed)

#모터 속도제어

DHT\_SENSOR = Adafruit\_DHT.DHT11

DHT\_PIN = 25  # DHT 센서가 연결된 GPIO 핀 번호입니다.

temp\_sensor='/sys/bus/w1/devices/28-fdd4451f64ff/w1\_slave' #수온 센서 구동 위치 지정, GPIO 4

address=0x48 #pcf8591

AIN2=0x42 #수위센서

AIN3=0x43 #탁도센서

AIN0 = 0x40 #조도센서

value = 0 #수위

value1 = 0 #수온

value2 = 0 #조도

value3 = 0 #탁도

value4 = 0 #온습도

humidity, temperature = Adafruit\_DHT.read\_retry(DHT\_SENSOR, DHT\_PIN)

bus=smbus.SMBus(1) #PCF8591 모듈 사용을 위한 smbus의 주소 및 변수 지정

mylcd = I2C\_LCD\_driver.lcd() #LCD 클리어 함수

def temp\_raw():

    f = open(temp\_sensor,'r')

    lines = f.readlines()

    f.close()

    return lines       #수온 센서 파일의 내용을 읽어오는 함수

# 온습도 값을 읽어오는 함수

def read\_dht\_sensor():

    global humidity, temperature

    while True:

        humidity, temperature = Adafruit\_DHT.read\_retry(DHT\_SENSOR, DHT\_PIN)

        time.sleep(10)  # 10초마다 온습도 값을 업데이트

# 온습도 값을 업데이트하는 스레드 생성 및 실행

dht\_thread = threading.Thread(target=read\_dht\_sensor)

dht\_thread.daemon = True  # 메인 스레드가 종료되면 함께 종료

dht\_thread.start()

def read\_temp():

    lines = temp\_raw()

    while lines[0].strip()[-3:] != 'YES':

        time.sleep(0.2)

        lines = temp\_raw()

    temp\_output = lines[1].find('t=')

    if temp\_output != -1:

        temp\_string = lines[1].strip()[temp\_output+2:]

        temp\_c = float(temp\_string) / 1000.0

        return temp\_c      #수온 센서 읽어온 파일의 구문을 분석해 온도부분을 반환하는 함수

# Picamera2 객체 생성

piCam = Picamera2()

# 미리보기 설정: 크기 640x480

piCam.preview\_configuration.main.size = (640, 480)

piCam.preview\_configuration.main.format = "RGB888"

piCam.preview\_configuration.align()

# 카메라 시작

piCam.start()

font = ImageFont.truetype('fonts/SCDream6.otf', 20)

def gen\_frames():

    while True:

        now = datetime.datetime.now()

        nowDatetime = now.strftime('%Y-%m-%d %H:%M:%S')

        frame = piCam.capture\_array()  # 현재 프레임을 캡처

        frame = Image.fromarray(frame)

        draw = ImageDraw.Draw(frame)

        draw.text(xy=(10, 15), text="스마트어항 " + nowDatetime, font=font, fill=(255, 255, 255))

        frame = np.array(frame)

        ref, buffer = cv2.imencode('.jpg', frame)

        frame = buffer.tobytes()

        yield (b'--frame\r\n'

               b'Content-Type: image/jpeg\r\n\r\n' + frame + b'\r\n')  # 그림파일들을 쌓아두고 호출을 기다림

@app.route('/')

def index():

    global value, value1, value2, value3, temperature, humidity

    # 센서 데이터를 필요한 대로 읽고 수집합니다.

    messagetosend = ''.join([str(x) for x in ['{0}mA\n\n{1}dark\n\n{2:.1f}°C\n\n{3}\n\n{4}°C, {5}%'.format(value,value1,value2,value3,temperature,humidity)]])

    return messagetosend

@app.route('/jadong', methods=['GET'])

def jadong():

    global value, value1, value2, value3, temperature, humidity

    #어플에서 자동모드 실행시

    mylcd.clear()#LCD에 남는 잔상 정리

    bus.write\_byte(address, AIN2)

    bus.read\_byte(address)

    lvwater=bus.read\_byte(address)

    print("value:%d" %lvwater) #수위센서 값 받아와서 화면에 출력

    bus.write\_byte(address,AIN0)

    bus.read\_byte(address)

    llux=bus.read\_byte(address)

    print(":{0}".format(llux)) #조도센서 값 받아와서 화면에 출력

    watertemp = read\_temp() #수온센서 변수 지정

    bus.write\_byte(address,AIN3)

    bus.read\_byte(address)

    pol=bus.read\_byte(address)

    print(":{0}".format(pol)) #탁도센서 값 받아와서 화면에 출력

    if humidity is not None and temperature is not None:

        print("온도={0:0.1f}°C 습도={1:0.1f}%".format(temperature, humidity)) #온습도 센서 값 출력

    mylcd.lcd\_display\_string("{0:.1f}C/{1:.1f}mA".format(watertemp, lvwater),1)

    mylcd.lcd\_display\_string("{0}da/NTU={1:.1f}".format(llux, pol),2) #LCD에 수온, 수위, 조도, 탁도 센서값 출력

#탁도, 조도, 수위센서 환경에 따라 조절해서 사용하세요. 0920 수정

    if pol < 60:

        GPIO.output(5,1)

        set\_motor\_speed\_2(40)

    elif pol > 80:

        GPIO.output(5,0)

        motor\_pwm\_2.stop()

    if llux > 236:

        GPIO.output(26, 1)

    elif llux < 234:

        GPIO.output(26, 0)

    if lvwater < 10:

        GPIO.output(13, 1)

        set\_motor\_speed\_3(70)

    elif lvwater > 0:

        GPIO.output(13, 0)

        motor\_pwm\_3.stop()

    return redirect('/')

@app.route('/stop', methods=['GET'])

def stop():

    GPIO.output(5, 0)

    GPIO.output(6, 0)

    GPIO.output(13, 0)

    GPIO.output(19, 0)

    GPIO.output(26, 0)#모든 GPIO에 연결된 모터, LED 정지

    motor\_pwm\_2.stop()

    motor\_pwm\_3.stop()

    global value, value1, value2, value3, temperature, humidity

    value = 0

    value1 = 0

    value2 = 0

    value3 = 0

    temperature = 0

    humidity = 0

    return redirect('/')

@app.route('/re', methods=['GET'])

def re():

    global value, value1, value2, value3, temperature, humidity#, jodo, takdo, suwui

    mylcd.clear()#LCD에 남는 잔상 정리

    bus.write\_byte(address, AIN2)

    bus.read\_byte(address)

    value=bus.read\_byte(address)

    print("value:%d"%value) #수위센서 값 받아와서 화면에 출력

    bus.write\_byte(address,AIN0)

    bus.read\_byte(address)

    value1=bus.read\_byte(address)

    print(":{0}".format(value1)) #조도센서 값 받아와서 화면에 출력

    value2 = read\_temp()

    print(value2) #수온센서 값 받아와서 화면에 출력

    bus.write\_byte(address,AIN3)

    bus.read\_byte(address)

    value3=abs((bus.read\_byte(address)))

    print("value:{0}".format(value3)) #탁도센서 값 받아와서 화면에 출력

    if humidity is not None and temperature is not None:

       print("온도={0:0.1f}°C 습도={1:0.1f}%".format(temperature, humidity)) #온습도 센서 값 출력

    mylcd.lcd\_display\_string("{0:.1f}C/{1:.1f}mA".format(value2, value),1)

    mylcd.lcd\_display\_string("{0}da/NTU={1:.1f}".format(value1, value3),2) #LCD에 수온, 수위, 조도, 탁도 센서값 출력

    return redirect('/')

@app.route('/ledon', methods=['GET'])

def ledon():     #어플에서 LED on

    GPIO.output(26, 1)

    return redirect('/')

@app.route('/ledoff', methods=['GET'])

def ledoff():   #어플에서 LED off

    GPIO.output(26, 0)

    return redirect('/')

@app.route('/saryoon', methods=['GET'])

def saryon():  #어플에서 사료 모터 on

    servo\_pwm.ChangeDutyCycle(3.5)

    time.sleep(1.0)

    servo\_pwm.ChangeDutyCycle(6.0)

    sleep(1)

    return redirect('/')

@app.route('/takon', methods=['GET'])

def takon(): #어플에서 물순환 모터 on

    GPIO.output(5, 1)

    GPIO.output(6, 0) #모터 동작

    set\_motor\_speed\_2(40)

    return redirect('/')

@app.route('/takoff', methods=['GET'])

def takoff(): #어플에서 물순환 모터 off

    GPIO.output(5, 0)

    GPIO.output(6, 0) #모터 정지

    motor\_pwm\_2.stop()

    return redirect('/')

@app.route('/wtron', methods=['GET'])

def wtron(): #어플에서 물공급 모터 on

    GPIO.output(13, 1)

    GPIO.output(19, 0) #모터 동작

    set\_motor\_speed\_3(70)#0920 수정

    return redirect('/')

@app.route('/wtroff', methods=['GET'])

def wtroff(): #어플에서 물공급 모터 off

    GPIO.output(13, 0)

    GPIO.output(19, 0) #모터 정지

    motor\_pwm\_3.stop()#0920 수정

    return redirect('/')

#cctv 코드

@app.route('/video')

def index2():

    return render\_template('index4#2.html')  # index4#2.html 템플릿을 렌더링하여 웹 페이지 반환

@app.route('/video\_feed')

def video\_feed():

    return Response(gen\_frames(), mimetype='multipart/x-mixed-replace; boundary=frame')

if \_\_name\_\_ == "\_\_main\_\_":

    app.run(host="0.0.0.0", port="8000")