

BỘ CÔNG THƯƠNG
TRƯỜNG ĐẠI HỌC CÔNG NGHIỆP TP HỒ CHÍ MINH

CHUYÊN ĐỀ IOT



BÀI BÁO CÁO THỰC HÀNH BUỔI 5

Giảng viên: TS. Cao Văn Kiên

Sinh viên thực hiện: Nhóm 6

Họ và tên	Mã số sinh viên
Đinh Tiến Thuận (100%)	20124381
Nguyễn Tấn Phong (100%)	20124691
Trần Phú Quý (80%)	20101441

Mỗi nhóm chỉ chọn 1 trong 3 mức độ tương ứng với số điểm tối đa có thể đạt được để thực hiện, viết báo cáo. Chương trình phải chạy đúng yêu cầu thì mới chấm điểm.

Mức độ 3 (10 điểm):

→ Web: Viết giao diện WEB với 4x2 nút nhấn và giao diện đồ thị thể hiện 2 giá trị nhiệt độ, độ ẩm từ server. 1 Nút nhấn chọn chế độ Auto/Manual; 3 nút nhấn điều khiển LED, Buzzer, Relay

→ Raspberry: Đọc cảm biến nhiệt độ, độ ẩm và gửi lên server mỗi 20s trong liên tục 30p. Hiển thị giá trị thời gian trên LCD 16x2. Giao tiếp server, đọc về giá trị 04 nút nhấn thực hiện chức năng sau:

- o 1 Nút nhấn để chọn chế độ Auto/Manual.
- o Ở chế độ Auto, LED sáng trong khoảng 18h đến 22h, còn lại LED tắt. 1 Buzzer kêu khi giá trị nhiệt độ trên 37 độ, tắt khi giá trị dưới 31 độ, còn lại giữ nguyên trạng thái. 1 Relay bật khi độ ẩm trên 90, tắt khi độ ẩm dưới 60, còn lại giữ nguyên.
- o Ở chế độ Manual, LED, Buzzer và relay được điều khiển bằng 3 nút nhấn trên web.

Yêu cầu:

- Thời gian từ lúc bấm nút nhấn khác nhau trên web đến khi có sự thay đổi trên thiết bị phải dưới 5s.
- Trong báo cáo trình bày rõ Code chương trình, lưu đồ giải thuật (20%).
- Giao diện WEB chạy được trên máy tính và điện thoại (20%).
- Minh chứng dạng 01 video (SV tự ghép thành 1 file video) được upload lên youtube và gửi link kèm trong báo cáo. Trong video phải có mô tả thí nghiệm, trình bày kết nối dây giữa module với board, trình bày chức năng từng dòng code Python trong thí nghiệm, trình bày kết quả thí nghiệm (ít nhất 3 lần thay đổi khác nhau) và nhận xét về kết quả thí nghiệm. (40%).

- **Code:**

```
import paho.mqtt.client as mqtt
from time import sleep
from grove.display.jhd1802 import JHD1802
from seeed_dht import DHT
from urllib import request, parse
import json
from gpiozero import LED
from datetime import datetime, time
import time as tg
from grove.grove_relay import GroveRelay

rl = GroveRelay(22)
lcd = JHD1802()
start_time = time(18, 0) # 18:00
end_time = time(22, 0) # 22:00
red = LED(5)
bz = LED(18)
sensor = DHT('11',16)

led = 0
buzzer = 0
relay = 0

def make_param_thingspeak(humi,temp):
    params = parse.urlencode({'field5':humi,'field6':temp}).encode()
    return params

def thingspeak_post(params):
    api_key_write = "6TY1GM4LJODNQ7LD"
    req = request.Request('http://api.thingspeak.com/update',method="POST")
    req.add_header("Content-Type","application/x-www-form-urlencoded")
    req.add_header("X-THINGSPEAKAPIKEY",api_key_write)
    r = request.urlopen(req,data= params)
    response_data = r.read()
    return response_data
# đọc dữ liệu từ thingspeak
def thingspeak_get_data():
    api_key_read = "48AZQL097YBSOS0T"
    channel_ID = "2268895"
    # Lấy dữ liệu từ field_num
    req =
request.Request("https://api.thingspeak.com/channels/2268895/feeds.json?api_key=48AZQL097YBSOS0T", method="GET")
    r = request.urlopen(req)
```

```

response_data = r.read().decode()
response_data = json.loads(response_data)
# Dữ liệu lịch sử là một danh sách các mục
data = response_data["feeds"]
data_c = response_data["channel"]

while True:
    b = tg.strftime("%H"+"h"+":'+"%M"+"m")
    current_time = datetime.now().time()
    humi, temp = sensor.read()
    print(humi)
    print(temp)
    lcd.setCursor(0, 0)
    lcd.write('ND:{0:2}c'.format(temp))
    lcd.setCursor(0,7)
    lcd.write('DA:{0:5}%'.format(humi))
    lcd.setCursor(1,0)
    lcd.write('tg: '+b)
    params_thingspeak = make_param_thingspeak(humi,temp)
    thingspeak_post(params_thingspeak)

    data,data_c = thingspeak_get_data()
    try:
        led = data[len(data)-1]['field%s'%(2)]
        led = int(led)
        print(0)
    except:
        led = led
        print(1)
    print("led: ", led)

    try:
        buzzer = data[len(data)-1]['field%s'%(3)]
        print(buzzer)
        buzzer = int(buzzer)
    except:
        buzzer = buzzer
    print("buzzer: ", buzzer)

    try:
        relay = data[len(data)-1]['field%s'%(4)]
        relay = int([relay])
    except:
        relay = relay
    print("relay: ", relay)

    try:
        a_m =data[len(data)-1]['field%s'%(1)]

```

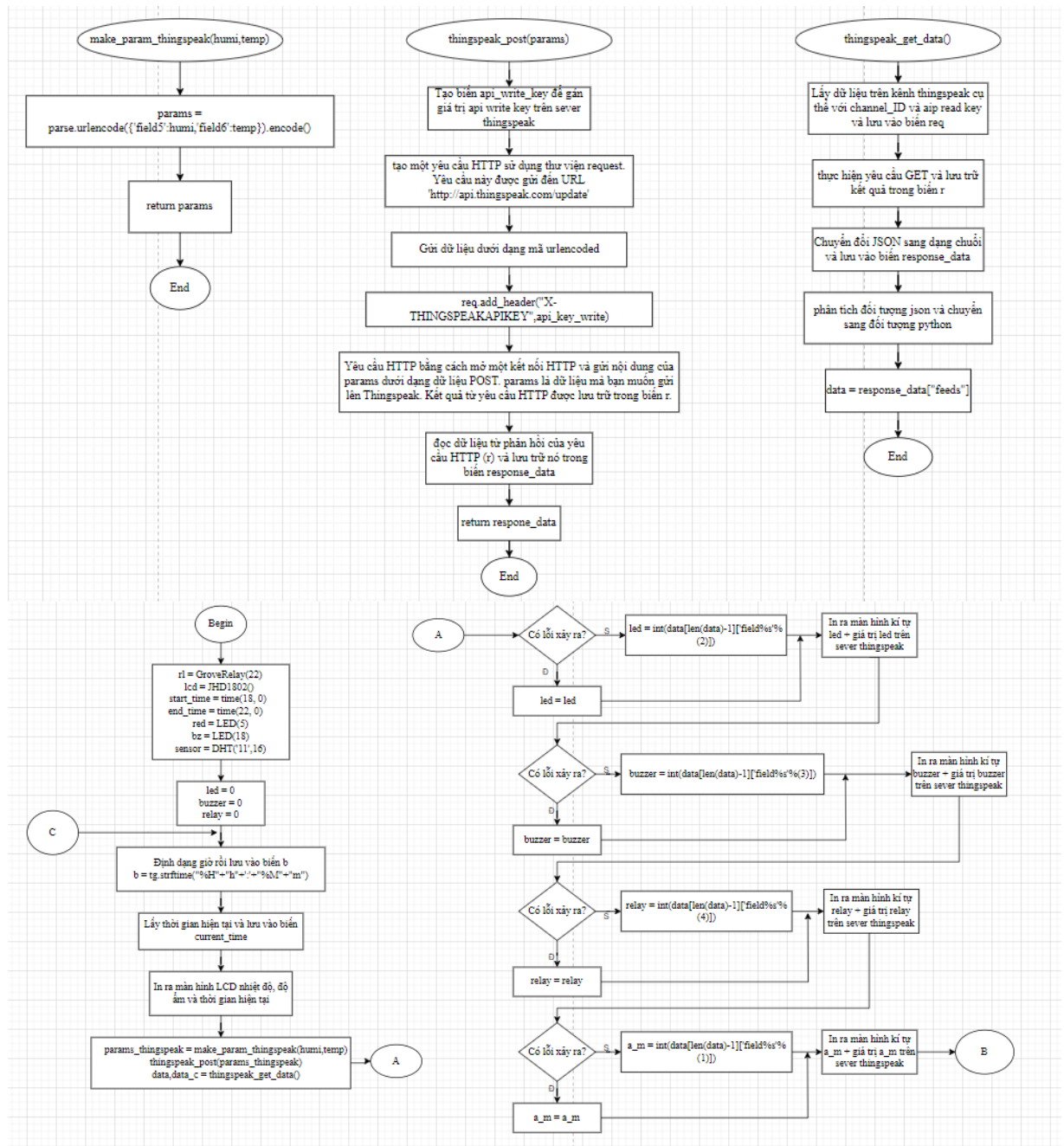
```

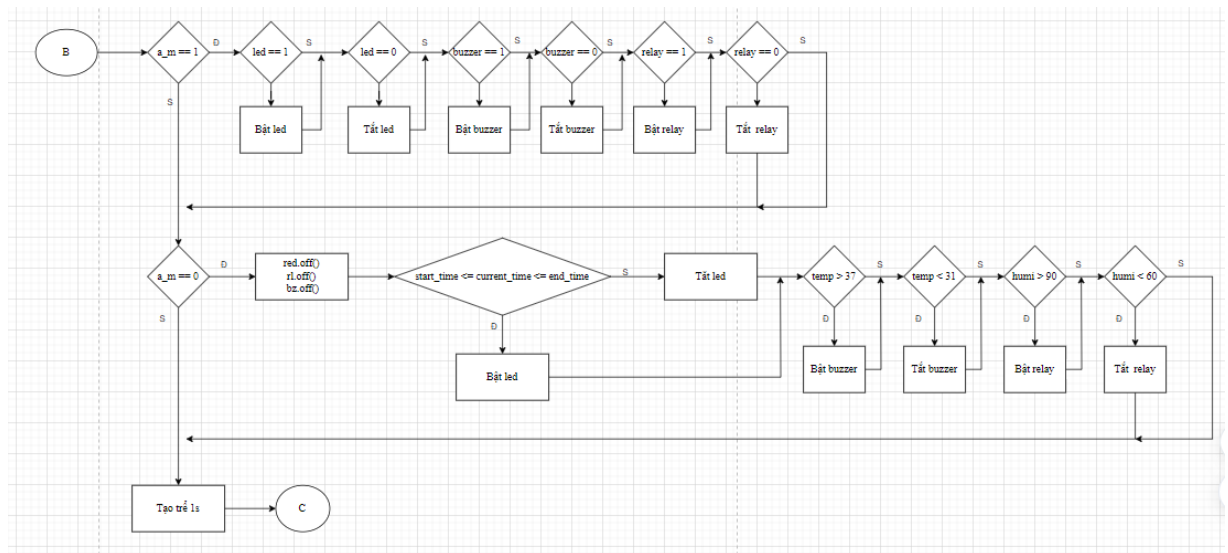
        a_m = int(a_m)
    except:
        a_m = a_m
    print("a_m: ", a_m)
    if a_m == 1 :
        if led == 1:
            red.on()
        if led == 0:
            red.off()
        if buzzer == 1:
            print(buzzer)
            bz.on()
        if buzzer == 0:
            bz.off()
        if relay == 1:
            print(relay)
            rl.on()
        if relay == 0:
            rl.off()

    if a_m == 0:
        red.off()
        rl.off()
        bz.off()
        if start_time <= current_time <= end_time:
            red.on()
        else:
            red.off()
        if temp > 37:
            bz.on()
        if temp < 31:
            bz.off()
        if humi > 90:
            rl.on()
        if humi < 60:
            rl.off()
    sleep(1)

```

- Lưu đồ:





- **Link youtube:**

<https://youtu.be/iee2n2d5vMM?si=grnwztARN2-DzVnT>