

逢甲大學學生報告 ePaper

物聯網智能家居
IoT Smart Home

作者：林承翰、楊采妮

系級：電子二乙

學號：D0949661、D0949866

開課老師：王通溫

課程名稱：IOT 智慧電子系統

開課系所：電子工程學系

開課學年：110 學年度 第一 學期



中文摘要

在現代生活中網路已是人們不可或缺的一部份，隨時隨地都有網路，目前來說網路都是我們再主動使用他，而我們希望能夠更加普及的使用在我們生活中看不到的地方，所以我們設想以物聯網的方式讓居家中的家電以連上網路的方式進行操控以達到生活上的便利，此企劃內主要設備含燈光、監控、溫溼度的及時測量和窗簾及暖爐等設備的控制，而控制這些設備的板子是 esp8266，透過 esp8266，接收感應元件的各項數據，去對各個設備做出最適當的操控，並加以適時的觀察數據的變化。結論來說我們成功完成物聯網，我們能夠在各個地方隨時的觀察物品的情況及各項數據，但目前還是有一些問題需要克服，至於問題文內有詳細說明及可能可以處理的辦法。應用範圍:自能居家設備，無人工廠等。

關鍵字:物聯網、數據觀測、自啟動、esp8266

Abstract

In modern life, the Internet has become an indispensable part of people. Internet is available anytime, anywhere. At present, we actively use the Internet, and we hope to use it more widely in our daily life. We envisage using the Internet of Things to control the home appliances by connecting to the Internet to achieve the convenience of life. The main equipment in this plan includes lighting, monitoring, real-time measurement of temperature and humidity, and curtains. And the control of equipment such as heaters, and the board that controls these equipment is esp8266, which passes through esp8266, receives various data of the sensing element, makes the most appropriate control of each equipment, and observes the data changes in a timely manner. In conclusion, we have successfully completed the Internet of Things, and we can observe the status of items and various data at any time in various places, but there are still some problems to be overcome. As for the problems, there are detailed explanations and possible solutions. Application range: self-energy home equipment, unmanned factories, etc.

Keywords: data observation, esp8266,internet of things, self-starting

目錄

中英文摘要	1.2
壹、研究動機	4
貳、研究方法	4
一、項目	4
二、概要	5
三、所需材料	5
四、內容	6
參、研究成果	13
肆、結論	15
伍、參考文獻	16

壹、研究動機

現代人生活過於忙碌，工作後回到家希望能有個舒適的居家環境，不用再因為各種小事而勞累，所以我們希望製作能自動調整家裡部分電器產品程式，讓我們可以在任何時間和地點觀測數據，並適時地自動做出調整。



貳、研究方法

一、項目：

測量項目->燈光、溫溼度(溫度)、監控

控制項目->窗簾、暖爐、風扇、LED 燈

二、概要：

我們以三塊 ESP8266 為核心，並用 Thonny 編寫程式，藉由房子內部的各個傳感系統所獲的各項數據，回傳給 ESP8266，再由 ESP8266 上的 wifi 模組傳至 thinkspeak，同時由獲得的數據適時的自動控制家中設備運作。

我們將其分為四大部分，溫溼度、內光、外光和攝像監控，而每部分又分為程式及硬體。

(溫溼度、內光、外光各使用一塊 ESP8266 板子)

三、所需材料：

ESP8266 x3、ESP32 CAM 320 萬畫素相機模組 x1、麵包板、跳線、電池組、光敏電阻 x2、馬達 x2、LED 燈 x1、溫溼度傳感器 x1、暖爐 x1、風扇 x1、繼電器 x7

四、內容：

(一)溫溼度

1. 溫溼度程式：修改老師上課教我們使用 DHT11 這一顆傳感器的程式及我們自行查閱資料並編寫完成的繼電器驅動程式合併而成。

```
import machine
from machine import Pin
import dht
import time
import urequests
import network

p1 = Pin(0, Pin.OUT)
p1.value(0)
p2 = Pin(4, Pin.OUT)
p2.value(0)

d = dht.DHT11(machine.Pin(2)) #pin link D4

ssid = "123456"
password = "qwertyuiop"
wifi = network.WLAN(network.AP_IF) # 创建一个热点
wifi.active(True) # 激活热点
try:
    wifi.connect('ssid','password')
    print('start to connect wifi')
    for i in range(10):
        print('try to connect wifi in {}s'.format(i))
        utime.sleep(1)
        if wifi.isconnected():
            break
```

```

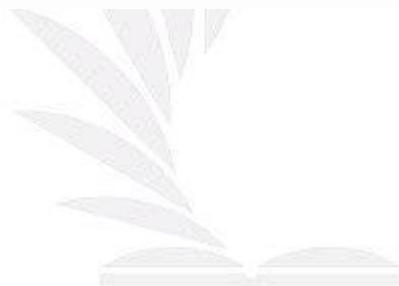
except Exception as e: print(e)

p0=Pin(2, Pin.IN)
d=dht.DHT11(p0)
host='http://api.thingspeak.com'
api_key='Y9PCKREKN0SVCUF2'

while True:
    d.measure() #重新測量溫溼度
    t=d.temperature() #讀取攝氏溫度
    f=round(t * 9/5 +32)
    h=d.humidity() #讀取相對溼度
    print('Temperature=', t, 'C', '/', f, 'F', 'Humidity=', h, '%')
    time.sleep(2) #暫停 2 秒
    if t>27: #高室溫5度
        p1.value(1)
    elif t<25: #hIGH
        p1.value(0)
    if t<24: #高室溫2度
        p2.value(1) #LOW
    elif t>25:
        p2.value(0)
    p1.value
    p2.value

```

2. 溫溼度硬體：



	S	VCC	GND
DHT11	D4	3V	G
風扇繼電器	D3	3V	G
暖爐繼電器	D2	3V	G

註：表格為連接到 ESP8266 的腳位

(二)內光

1. 內光程式：

```
except Exception as e: print(e)

#p0=Pin(2, Pin.IN)
#d=dht.DHT11(p0)
host='http://api.thingspeak.com'
api_key='Y9PCKREKNØSVCUF2'

while True:
    adc=machine.ADC(0)    #建立 ADC 物件
    value=adc.read()      #讀取類比數位轉換器輸出
    print("value=",value, "str=",str(round(value*100/1024)) + '%')
    url='%s/update?api_key=%s&field4=%s' %(host, api_key, value)
    #print('url=', url)
    r=urequests.get(url)
    print('response=', r.text)
    time.sleep(3)
    if value<=360:
        p0.value(1)
        time.sleep(5)
    else:
        p0.value(0)
```

```
import machine
from machine import Pin
import dht
import time
import urequests
import network

p0 = Pin(5, Pin.OUT)
p0.value(0)
ssid = "123456"
password = "qwrtyuiop"
wifi = network.WLAN(network.AP_IF) # 创建一个热点
wifi.active(True)                 # 激活热点
try:
    wifi.connect('ssid','password')
    print('start to connect wifi')
    for i in range(10):
        print('try to connect wifi in {}'.format(i))
        utime.sleep(1)
        if wifi.isconnected():
            break
    if wifi.isconnected():
```

2. 內光硬體：

	S	VCC	GND
光敏電阻	A0	3V	G
繼電器	D1	3V	G

註：表格為連接到 ESP8266 的腳位

(三)外光

1. 外光程式：

```
import machine
from machine import Pin
import dht
import time
import urequests
import network

p0 = Pin(5, Pin.OUT)
p0.value(1)
p1 = Pin(4, Pin.OUT)
p1.value(0)
ssid = "123456"
password = "qwrtyuiop"
wifi = network.WLAN(network.AP_IF) # 创建一个热点
wifi.active(True) # 激活热点
try:
    wifi.connect('ssid','password')
    print('start to connect wifi')
    for i in range(10):
        print('try to connect wifi in {}'.format(i))
        utime.sleep(1)
        if wifi.isconnected():
            break
    if wifi.isconnected():
        print('WiFi connection OK!')
        print('Network Config=',wifi.ifconfig())
```

```

except Exception as e: print(e)

#p0=Pin(2, Pin.IN)
#d=dht.DHT11(p0)
host='http://api.thingspeak.com'
api_key='Y9PCKREKN0SVCUF2'

while True:
    adc=machine.ADC(0)    #建立 ADC 物件
    value=adc.read()      #讀取類比數位轉換器輸出
    print("value=",value, "str=",str(round(value*100/1024)) + '%')    #輸出值
    url='%s/update?api_key=%s&field4=%s' %(host, api_key, value)
    #print('url=', url)
    r=urequests.get(url)
    print('response=', r.text)
    time.sleep(3)
    if value>=900:
        p0.value(0)
        time.sleep(6)
        p0.value(1)
        time.sleep(5)
    elif value<=200:
        p1.value(1)
        time.sleep(6)
        p1.value(0)

```

2. 外光硬體

	S	VCC	GND
光敏電阻	A0	3V	G
繼電器啟動	D1	3V	G
繼電器收回	D2	3V	G

註：表格為連接到 ESP8266 的腳位

(四) 攝像監控

1. 攝像監控程式：arduino 內部相機基本範例稍作修改

```
config.pixel_format = PIXFORMAT_JPEG;

// if PSRAM IC present, init with UNGA resolution and higher JPEG quality
// for larger pre-allocated frame buffer.
if(psramFound()){
  config.frame_size = FRAMESIZE_UMGA;
  config.jpeg_quality = 10;
  config.fb_count = 2;
} else {
  config.frame_size = FRAMESIZE_SVGA;
  config.jpeg_quality = 12;
  config.fb_count = 1;
}

#ifdef CAMERA_MODEL_ESP_EYE
pinMode(13, INPUT_PULLUP);
pinMode(14, INPUT_PULLUP);
#endif

// camera init
esp_err_t err = esp_camera_init(&config);
if (err != ESP_OK) {
  Serial.printf("Camera init failed with error 0x%x", err);
  return;
}

sensor_t * s = esp_camera_sensor_get();
// initial sensors are flipped vertically and colors are a bit saturated
if (s->id.PID == OV3660_PID) {
  s->set_vflip(s, 1); // flip it back
  s->set_brightness(s, 1); // up the brightness just a bit
  s->set_saturation(s, -2); // lower the saturation
}

// drop down frame size for higher initial frame rate
s->set_framesize(s, FRAMESIZE_QVGA);

#ifdef CAMERA_MODEL_MSSTACK_WIDE || defined(CAMERA_MODEL_MSSTACK_ESP32CAM)
s->set_vflip(s, 1);
s->set_hmirror(s, 1);
#endif

WiFi.begin(ssid, password);

while (WiFi.status() != WL_CONNECTED) {
```



```
config.pixel_format = PIXFORMAT_JPEG;

// if PSRAM IC present, init with UXGA resolution and higher JPEG quality
// for larger pre-allocated frame buffer.
if(psramFound()){
  config.frame_size = FRAMESIZE_UXGA;
  config.jpeg_quality = 10;
  config.fb_count = 2;
} else {
  config.frame_size = FRAMESIZE_SVGA;
  config.jpeg_quality = 12;
  config.fb_count = 1;
}

#ifdef CAMERA_MODEL_ESP_EYE
pinMode(13, INPUT_PULLUP);
pinMode(14, INPUT_PULLUP);
#endif

// camera init
esp_err_t err = esp_camera_init(&config);
if (err != ESP_OK) {
  Serial.printf("Camera init failed with error 0x%x", err);
  return;
}

sensor_t * s = esp_camera_sensor_get();
// initial sensors are flipped vertically and colors are a bit saturated
if (s->id.PID == OV3660_PID) {
  s->set_vflip(s, 1); // flip it back
  s->set_brightness(s, 1); // up the brightness just a bit
  s->set_saturation(s, -2); // lower the saturation
}
// drop down frame size for higher initial frame rate
s->set_framesize(s, FRAMESIZE_QVGA);

#ifdef CAMERA_MODEL_MSSTACK_WIDE || defined(CAMERA_MODEL_MSSTACK_ESP32CAM)
s->set_vflip(s, 1);
s->set_hmirror(s, 1);
#endif

WiFi.begin(ssid, password);

while (WiFi.status() != WL_CONNECTED) {
```

2. 攝像監控硬體：ESP32 CAM 320 萬畫素相機模組

參、研究成果

一、溫溼度:當溫度過低引響到人的活動時，屋內的暖爐便會自行啟動，以確保溫度適中當溫度達 25 度時，暖爐便會停止，反之當屋內溫度高時送風系統運作，增加屋內的空氣流通，以達到散熱的目的。

二、屋外光線：窗戶旁有一個光敏電阻，當檢測到屋外有強光時，會使板子驅動馬達以關上窗簾遮蔽光線，如果屋外光下降後，窗簾便會自動打開

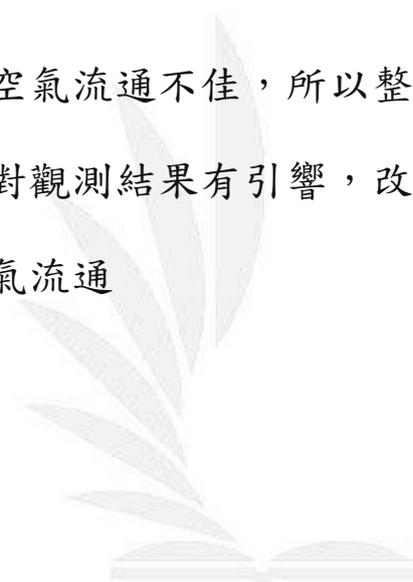
三、屋內光線：當屋內的光線小於 360(經測量得出數據)時屋內光源啟動，而大於 360 則關閉，但屋內光源較不容易影響其數值，主要變動為戶外的光造成的影響

四、攝像監控：即時顯示當下影像

五、數據統計：測量實驗途中各項數據紀錄於 thinkspeak 並即時顯示數據。

六、需要改進的地方：由於我們分成三塊版子雖然在製作放較為方便，但在最後測試時，發現沒有辦法三台同時接上電腦一起啟動，最後只能一個一個的展示而改進的方法有以下幾種，一 更換有更多腳位的板子把程式寫在一起，但在程式可能就要再做更多的調整 二 使用 arduino 寫直接讓板子全部一起上電就可以了

使用暖爐時，由於空氣流通不佳，所以整棟房子的溫度上升的很不平均這樣會對觀測結果有引響，改善方法在暖爐後方加設風扇以增加空氣流通



伍、結論

對於我們來說這是真正的第一次自己完成一份程式及硬體上的結合，雖然大一時，有玩過自走車，但都是學長說審麼，我們就做什麼，也不了解這些程式碼要幹嘛，只知道改這個車會動就這樣，但這次我們一步步從程式的編寫，雖然用現成的作為模板再做修改，但我們有去了解每一行程式對於硬體或網路上的聯結有何作用。而這次最大的問題也是程式，因為我們只有學 c 沒有學過 python，所以只由一邊上網找一邊麻煩老師了，最後才完成全部的程式，而在硬體上自己實際動手做才知道理論真的不等於實際，常常想的在實際動手知道無法成功實行，連帶程式都要一起稍作改，成品也事在最後的時間才完成出來的。

陸、參考文獻

Alan Wang B.C.(2020).Simple MicroPython IoT Smart Home (ESP8266).Retrieved January 18, 2020, from <https://www.hackster.io/alankrantas/simple-micropython-iot-smart-home-esp8266-d83f8e>

小狐狸事務所 (2017) 。MicroPython on ESP8266 (十五)：光敏電阻與 ADC 測試，2017 年 7 月 29 日，取自：
<http://yhhuang1966.blogspot.com/2017/07/micropython-on-esp8266-adc.html>

米羅科技文創學院(2021) 。【物聯網 入門教學課程 2】ESP32-CAM 打造無線影像串流主機，2021 年 4 月 15 日，取自：
<https://shop.mirotek.com.tw/iot/iot-start-2/>