

逢甲大學學生報告 ePaper

智能小溫室 intelligent greenhouse

作者：羅世皓、黃敬皓

系級：電子二甲

學號：D0876874、D0817828

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(IC 設計學程)

開課學年：一百零九學年度 第一學期



摘要:

目的:

現代人上班忙碌，沒有時間照顧植物，那些上班族喜歡種植物就沒空去照顧它，使得植物死亡。所以我們製作了一個小溫室來幫忙照顧植物。

原理:

1. 土壤濕度感測器讀取土壤的濕度，如果濕度太低, ESP8266 則會讓抽水馬達啟動，給植物水分。
2. 溫溼度感測器則讀取溫室內的溫度及濕度，植物不可能在溫度及濕度過高及過低的情況下生存。如果溫度太高或濕度太高 ESP8266 會驅動風扇來達到降溫的效果。
3. 使用 ESP32 啟動 OV7670 鏡頭並連接到網絡，以便在網絡上觀測植物的生長期況。
4. 光敏電阻，當光線不足時 ESP8266 使 LED 燈亮。
5. 使用 ESP8266 的 WIFI 功能連接到互聯網，將數據上傳至 ThingSpeak。
6. 使用降壓模組把 9Vinput 電源降壓至 5V，為 ESP32, ESP8266 等各項設備供電。
7. 使用 ADS1115 擴充 ESP8266 的 analog input，讓 ESP8266 可以連接更多類比輸出的感測器。
8. 用 ESP8266 控制 RELAY 模組，控制需要較高電壓驅動的設備，如風扇，抽水馬達。

關鍵字：土壤溼度感測器、植物觀察、溫溼度感測器

Abstract

purpose:

Modern people are busy at work and do not have time to take care of plants. Those office workers who like to grow plants have no time to take care of them, causing the plants to die. So we made a small greenhouse to help take care of the plants.

principle:

1. The soil moisture sensor reads the moisture of the soil. If the moisture is too low, ESP8266 will start the pumping motor to provide moisture to the plants.
2. The temperature and humidity sensor reads the temperature and humidity in the greenhouse. It is impossible for plants to survive under the conditions of too high or too low temperature and humidity. If the temperature is too high or the humidity is too high, ESP8266 will drive a fan to achieve the effect of cooling down.
3. Use ESP32 to start the OV7670 lens and connect to the network to observe the growth period of plants on the network.
4. Photoresistor, ESP8266 will make the LED light bright when the light is insufficient.
5. Use the WIFI function of ESP8266 to connect to the Internet and upload the data to ThingSpeak.
6. Use a step-down module to step down the 9Vinput power supply to 5V to supply power for various devices such as ESP32 and ESP8266.
7. Use ADS1115 to expand the analog input of ESP8266, so that ESP8266 can connect more sensors with analog output.
8. Use ESP8266 to control the RELAY module to control devices that require higher voltage drive, such as fans and pumping motors.

Keyword: Moisture sensor, Temperature-Humidity sensor, ThingSpeak

目錄：

摘要：	1
Abstract	2
目錄：	3
一、 前言：	4
二、 軟硬體設備介紹：	5
三、 程式碼：	7
四、 成品：	16
五、 實驗結果與分析	17
1. 實驗設置：	17
2. 實驗結果：	17
3. ThingSpeak 觀察數據：	18
六、 結論：	19
七、 參考資料：	20

一、前言：

物聯網(Internet of Things, IoT):1999年美國麻省理工學院成立AUTO-ID中心提出物聯網概念，物聯網系統基本建立在三層架構上「感測層」、「網路層」、「應用層」；感測層-將各種具有感測能力的原件嵌入物件中，擷取各種實用資訊，而這些數據經過分析處理就成了我們生活上的最佳參考，例如:生活中常見的超速照相機運用到移動感測器、自動照明系統運用紅外線感測器等；網路層-IoT的智慧物件都具有的功能「聯網」，代表著收發資訊的功能，比較常見的無線通訊技術是Wi-fi、藍芽等。

雲端技術(Cloud Technology):指的是一種新型態的網路運算，在過去舊有觀念個人電腦一般做為被動式終端機(dumb terminal)接收並顯示大型主機或伺服器的運算結果；雲端技術象徵程式化的控制，使用者可以對伺服器進行個人化的指令，選擇服務項目，修改伺服器回傳結果；這樣的技術強化了個人電腦的功能，能有助於數位文化的發展，同時讓資源分享更為廣泛。雲端是服務，也是技術，我們利用各式雲端平台達成資料傳遞、開發、管理與使用，雲端平台的好處是:人們不用下載大量的軟體，伺服器硬體設備方面要求也大幅下降，我們可以藉由雲端服務進行資料存取及運算，同時它打破距離屏障世界各地的人可以很便利的共享資訊。

在現在忙碌的社會下，越來越少人投入種植花朵或是青菜等植物。所以許多科技工司已經開始使用溫室來種植作物。有些人可能希望可以在自己家中種植植物來美化家中環境。所以我們利用最平價的方式來製作了一個簡單的小溫室。

我們利用Arduino的感測器來製作，首先從瞭解此元件的功用後，利用Arduino uno板來進行單方面實驗。Arduino uno板實驗完成後，我們使用ESP32及ESP8266開發板來進行所有元件的結合，由於ESP8266只有單個傳輸，為了這問題我們決定多添購一個擴充板來解決問題。為了可以在手機上可以觀察到植物的生長狀況及溫室內的變化，ESP32連接鏡頭在溫室內，只是畫面會延遲1到2分鐘，畫質也沒有計畫中的好。

二、軟硬體設備介紹：

1. ESP32



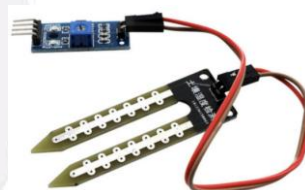
2. ESP8266



3. 風扇



4. 土壤濕度感測器



5. 溫溼度感測器



6. 抽水馬達



7. RELAY 模組



8. 盆栽(示意圖)



9. LED



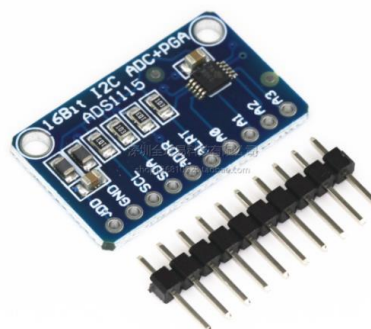
10. 光敏電阻



11. 溫室箱



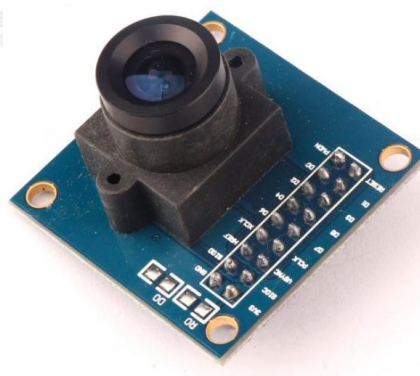
12. ADS1115 ADC



13. DC-DC 降壓模組



14. OV7670 鏡頭模組



三、程式碼：

2-1. ESP8266-土壤溼度、溫溼度、光敏電阻

```
#include <Adafruit_NeoPixel.h>
#ifdef __AVR__
  #include <avr/power.h> // Required for 16 MHz Adafruit Trinket
#endif
#include <ESP8266WiFi.h>
#include "DHT.h"
#define DHTPIN 4
#define PIN 15
#define NUMPIXELS 20 // Popular NeoPixel ring size
#include <Wire.h>
#include <Adafruit_ADS1015.h>
#define DHTPIN 2
DHT dht(DHTPIN, 11);
Adafruit_NeoPixel pixels(NUMPIXELS, PIN, NEO_GRB + NEO_KHZ800);
Adafruit_ADS1115 ads(0x48);

char ssid[] = "Sam'sZenFone6"; // your network SSID (name)
char pass[] = "69331535"; // your network password
WiFiClient client;
unsigned long myChannelNumber = 1259435;
unsigned long myTalkBackID = 40547;
const char * myTalkBackKey = "Y82WP0ZY5ELWJM6L";
const char * myWriteAPIKey = "NVP6SF8W4UIBDTEP";
// Initialize values for ThingSpeak updates
```



```
void setup() {
  #if defined(__AVR_ATtiny85__) && (F_CPU == 16000000)
  clock_prescale_set(clock_div_1);
  #endif
  pinMode(A0, INPUT);
  pinMode(12, OUTPUT);
  pinMode(14, OUTPUT);
  digitalWrite(12, HIGH);
  digitalWrite(14, HIGH);
  ads.begin();
  dht.begin();
  pixels.begin();
  Serial.begin(9600); // Initialize serial
}

void loop() {
  int16_t light;
  light = ads.readADC_SingleEnded(0)/16;
  float h = dht.readHumidity();
  float t = dht.readTemperature();
  if(int(t)>30){
    digitalWrite(14, LOW);
  }
  else digitalWrite(14, HIGH);
  if(analogRead(A0)>670){
    digitalWrite(12, LOW);
  }
  else digitalWrite(12, HIGH);
  if(WiFi.status() != WL_CONNECTED){
    // Connect or reconnect to Wi-Fi
    Serial.print("Attempting to connect to SSID: ");
    Serial.println(String(ssid));
    while(WiFi.status() != WL_CONNECTED){
      WiFi.begin(ssid, pass);
      Serial.print(".");
      delay(5000);
    }
    Serial.println("\nConnected.");
  }
}
```

```
pixels.clear();
if(light<500){
  pixels.setPixelColor(0, pixels.Color(229, 0, 40));
  pixels.setPixelColor(1, pixels.Color(229, 0, 40));
  pixels.setPixelColor(2, pixels.Color(229, 0, 40));
  pixels.setPixelColor(3, pixels.Color(229, 0, 40));
  pixels.setPixelColor(4, pixels.Color(229, 0, 40));
  pixels.setPixelColor(5, pixels.Color(229, 0, 40));
  pixels.setPixelColor(6, pixels.Color(229, 0, 40));
  pixels.setPixelColor(7, pixels.Color(229, 0, 40));
  pixels.setPixelColor(8, pixels.Color(229, 0, 40));
  pixels.setPixelColor(9, pixels.Color(229, 0, 40));
  pixels.setPixelColor(10, pixels.Color(229, 0, 40));
  pixels.setPixelColor(11, pixels.Color(229, 0, 40));
  pixels.setPixelColor(12, pixels.Color(229, 0, 40));
  pixels.setPixelColor(13, pixels.Color(229, 0, 40));
  pixels.setPixelColor(14, pixels.Color(229, 0, 40));
  pixels.setPixelColor(15, pixels.Color(229, 0, 40));
  pixels.setPixelColor(16, pixels.Color(229, 0, 40));
  pixels.setPixelColor(17, pixels.Color(229, 0, 40));
}

void loop() {
  int16_t light;
  light = ads.readADC_SingleEnded(0)/16;
  float h = dht.readHumidity();
  float t = dht.readTemperature();
  if(int(t)>30){
    digitalWrite(14,LOW);
  }
  else digitalWrite(14,HIGH);
if(analogRead(A0)>670){
  digitalWrite(12,LOW);
}
else digitalWrite(12,HIGH);
if(WiFi.status() != WL_CONNECTED){
  // Connect or reconnect to Wi-Fi
  Serial.print("Attempting to connect to SSID: ");
  Serial.println(String(ssid));
  while(WiFi.status() != WL_CONNECTED){
    WiFi.begin(ssid, pass);
    Serial.print(".");
    delay(5000);
  }
  Serial.println("\nConnected.");
}
```

```
else{
pixels.setPixelColor(0, pixels.Color(0, 0, 0));
pixels.setPixelColor(1, pixels.Color(0, 0, 0));
pixels.setPixelColor(2, pixels.Color(0, 0, 0));
pixels.setPixelColor(3, pixels.Color(0, 0, 0));
pixels.setPixelColor(4, pixels.Color(0, 0, 0));
pixels.setPixelColor(5, pixels.Color(0, 0, 0));
pixels.setPixelColor(6, pixels.Color(0, 0, 0));
pixels.setPixelColor(7, pixels.Color(0, 0, 0));
pixels.setPixelColor(8, pixels.Color(0, 0, 0));
pixels.setPixelColor(9, pixels.Color(0, 0, 0));
pixels.setPixelColor(10, pixels.Color(0, 0, 0));
pixels.setPixelColor(11, pixels.Color(0, 0, 0));
pixels.setPixelColor(12, pixels.Color(0, 0, 0));
pixels.setPixelColor(13, pixels.Color(0, 0, 0));
pixels.setPixelColor(14, pixels.Color(0, 0, 0));
pixels.setPixelColor(15, pixels.Color(0, 0, 0));
pixels.setPixelColor(16, pixels.Color(0, 0, 0));
pixels.setPixelColor(17, pixels.Color(0, 0, 0));
}
pixels.show();
```



```
// Create the message body for the POST out of the values
String postMessage = String("field1=") + String(h) +
String("&field2=") + String(t) + String("&field3=") +
String(light) + String("&field4=") + String(analogRead(A0)) +
String("&api_key=") + String(myWriteAPIKey) + String("&talkback_key=") + String(myTalkBackKey);
// Make a string for any commands in the queue
String newCommand = String();
// Make the POST to ThingSpeak
int x = httpPOST(postMessage, newCommand);

client.stop();
// if(x != 200){
//   Serial.println("Problem checking queue. HTTP error code " + String(x));
// }
delay(1000); // Wait 20 seconds to update the channel again
}
```

```
int httpPOST(String postMessage, String &response){
    bool connectSuccess = false;
    connectSuccess = client.connect("api.thingspeak.com",80);
    if(!connectSuccess){
        return -301;
    }
    postMessage += "&headers=false";
    String Headers = String("POST /update HTTP/1.1\r\n") +
        String("Host: api.thingspeak.com\r\n") +
        String("Content-Type: application/x-www-form-urlencoded\r\n") +
        String("Connection: close\r\n") +
        String("Content-Length: ") + String(postMessage.length()) +
        String("\r\n\r\n");
    client.print(Headers);
    client.print(postMessage);
    long startWaitForResponseAt = millis();
    while(client.available() == 0 && millis() - startWaitForResponseAt < 5000){
        delay(100);
    }
    if(client.available() == 0){
        return -304; // Didn't get server response in time
    }
    if(!client.find(const_cast<char *>("HTTP/1.1"))){
        return -303; // Couldn't parse response (didn't find HTTP/1.1)
    }

    int status = client.parseInt();
    if(status != 200){
        return status;
    }
    if(!client.find(const_cast<char *>("\n\r\n"))){
        return -303;
    }
    String tempString = String(client.readString());
    response = tempString;
    return status;
}
```

2-2. ESP32 鏡頭

```
#include "OV7670.h"
#include <Adafruit_GFX.h> // Core graphics library
#include <Adafruit_ST7735.h> // Hardware-specific library
#include <WiFi.h>
#include <WiFiMulti.h>
#include <WiFiClient.h>
#include "BMP.h"
const int SIOD = 21; //SDA
const int SIOC = 22; //SCL
const int VSYNC = 34;
const int HREF = 35;
const int XCLK = 32;
const int PCLK = 33;
const int D0 = 27;
const int D1 = 17;
const int D2 = 16;
const int D3 = 15;
const int D4 = 14;
const int D5 = 13;
const int D6 = 12;
const int D7 = 4;
const int TFT_DC = 2;
const int TFT_CS = 5;
#define ssid1 Sam'sZenFone6 "YOUR_WIFI_SSID"
#define password1 69331535 "YOUR_PASSWORD"
Adafruit_ST7735 tft = Adafruit_ST7735(TFT_CS, TFT_DC, 0/*no reset*/);
OV7670 *camera;
WiFiMulti wifiMulti;
WiFiServer server(80);
unsigned char bmpHeader[BMP::headerSize];
```

智能小溫室

```
void serve()
{
  WiFiClient client = server.available();
  if (client)
  {
    //Serial.println("New Client.");
    String currentLine = "";
    while (client.connected())
    {
      if (client.available())
      {
        char c = client.read();
        //Serial.write(c);
        if (c == '\n')
        {
          if (currentLine.length() == 0)
          {
            client.println("HTTP/1.1 200 OK");
            client.println("Content-type:text/html");
            client.println();
            client.print(
              "<style>body{margin: 0}\nimg{height: 100%; width: auto}</style>"
              "<img id='a' src='/camera' onload='this.style.display=\"initial\"; var b = document.getElementById(\"b\"); b.style.display=\"none\"; b.src=\"/camera?\"+Date.now(); '>"
              "<img id='b' style='display: none' src='/camera' onload='this.style.display=\"initial\"; var a = document.getElementById(\"a\"); a.style.display=\"none\"; a.src=\"/camera?\"+Date.now(); '>");
            client.println();
            break;
          }
        }
      }
    }
  }
}
```

智能小溫室

```
        else
        {
            currentLine = "";
        }
    }
    else if (c != '\r')
    {
        currentLine += c;
    }

    if(currentLine.endsWith("GET /camera"))
    {
        client.println("HTTP/1.1 200 OK");
        client.println("Content-type:image/bmp");
        client.println();

        client.write(bmpHeader, BMP::headerSize);
        client.write(camera->frame, camera->xres * camera->yres * 2);
    }
}
}
// close the connection:
client.stop();
//Serial.println("Client Disconnected.");
}
}

void setup()
{
    Serial.begin(115200);

    wifiMulti.addAP(ssid1, password1);
    //wifiMulti.addAP(ssid2, password2);
    Serial.println("Connecting Wifi...");
    if(wifiMulti.run() == WL_CONNECTED) {
        Serial.println("");
        Serial.println("WiFi connected");
        Serial.println("IP address: ");
        Serial.println(WiFi.localIP());
    }

    camera = new OV7670(OV7670::Mode::QQVGA_RGB565, SIOD, SIOC, VSYNC, HREF, XCLK, PCLK, D0, D1, D2, D3, D4, D5, D6, D7);
    BMP::construct16BitHeader(bmpHeader, camera->xres, camera->yres);

    tft.initR(INITR_BLACKTAB);
    tft.fillScreen(0);
    server.begin();
}
```

```
void displayY8(unsigned char * frame, int xres, int yres)
{
    tft.setAddrWindow(0, 0, yres - 1, xres - 1);
    int i = 0;
    for(int x = 0; x < xres; x++)
        for(int y = 0; y < yres; y++)
            {
                i = y * xres + x;
                unsigned char c = frame[i];
                unsigned short r = c >> 3;
                unsigned short g = c >> 2;
                unsigned short b = c >> 3;
                tft.pushColor(r << 11 | g << 5 | b);
            }
}

void displayRGB565(unsigned char * frame, int xres, int yres)
{
    tft.setAddrWindow(0, 0, yres - 1, xres - 1);
    int i = 0;
    for(int x = 0; x < xres; x++)
        for(int y = 0; y < yres; y++)
            {
                i = (y * xres + x) << 1;
                tft.pushColor((frame[i] | (frame[i+1] << 8)));
            }
}

void loop()
{
    camera->oneFrame();
    serve();
    displayRGB565(camera->frame, camera->xres, camera->yres);
}
```


四、成品：



(成品圖)



(網路鏡頭影像)

五、實驗結果與分析

1. 實驗設置：

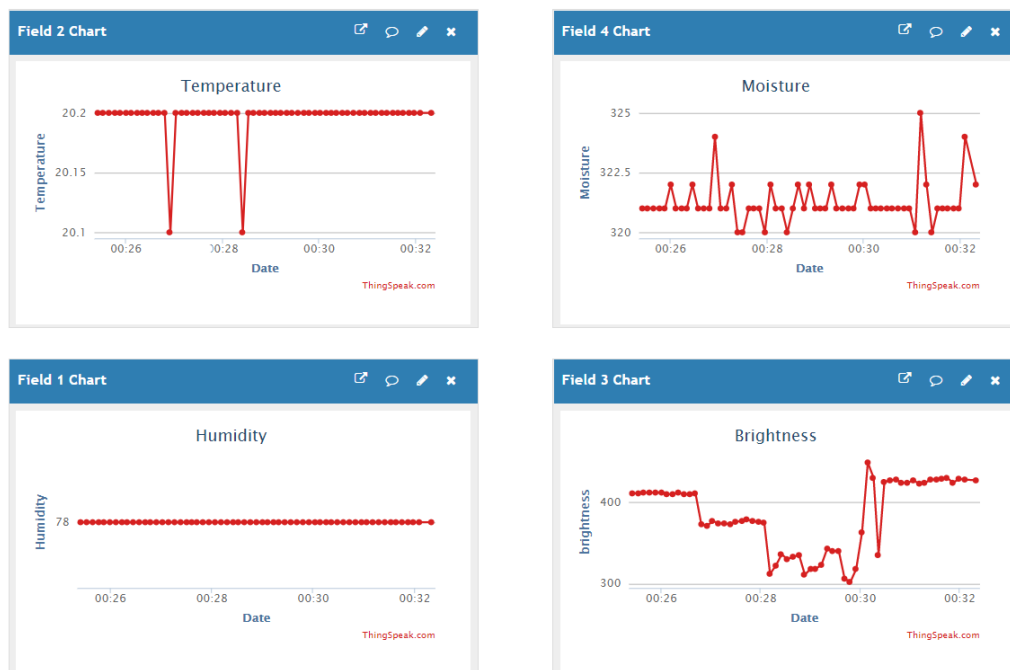
為了實驗本文的有效性，架設一個 Thingspeak 頻道。Thingspeak 是一個網路付費平台，主要功能是做為物聯網的雲端資料庫，接收、整合、展示感測器回傳的數據，此平台支持數據計算軟體 MATLAB，Thingspeak 用戶分析和使用 MATLAB 上傳的數據是可以的，不須而外購買 MATLAB 許可證；甚至，用戶可以在 Thingspeak 網站上登入已註冊的 Mathworks 用戶帳號做使用。雲端平台的選擇可以根據使用者的條件與需求做改變，確認平台後，開始撰寫 Arduino 程式，由於嵌入式系統的執行程式一旦上傳至開發板，若要做更動需重新連結開發板進行上傳，同時一個完整的物聯網系統，除了有傳感器的架設，還需要合適終端來接收資料，所以在編寫程式碼前須先確定雲端平台的架設已完成。程式部分主要會分成三個功能：感測器輸入、數據回傳頻率、資料上傳雲端，本文使用 Arduino 軟體平台進行編程，程式語言為 C 語言，下面將以圖片型式展示部分程式碼。

以上步驟完成後將設置好的感測元件及植物放入溫室中，大致上已完成。甚下就是觀察溫室內的變化。

2. 實驗結果：

在大約一個禮拜的觀察下，我們發下溫度跟濕度沒有變化但風扇一直轉動，在經過檢查後我們發現，極電器會讓 ESP8266 跳電導致數據顯示溫度太高或濕度太濕，將問題找到後發現是焊接時不小心短路。之後也沒發生什麼問題，接下來就是在 ThingSpeak 上觀察數據。

3. ThingSpeak 觀察數據：



(上傳至雲端數據:左上:溫度、右上:土壤濕度、左下:濕度、右下:光亮度)

六、結論：

用科技的方式來照顧植物雖然很方便，但缺少了對植物的愛戴，長出來的花花草草感覺沒什麼生命力。所以要種植物還是等你老了退休再來做這種事吧。

在研究過程中，我們把植物放在室內，所以有些光感測數值每有很準確。建議有興趣的同學，可以將溫室放在室外且沒有燈光乾擾下來觀察。我們在挑選植物也是想了很久，建議不要種植花朵。因為花在一段時間後會凋謝死亡。可以選用一些生命力堅強的植物，最好可以用菜類。影像部分畫面因為適用 WIFI 傳輸，會有所延遲。

希望優興趣的同學可以參考，有新的發現可以與我們分享，希望可以做出更好的作品。



七、參考資料：

裴有恆、陳玟錡(2018)。AIoT 人工智慧在物聯網的應用與商機。台北市:基峰資訊

CEILING TSAI 2017 年 6 月 8 日。Arduino 筆記(27)：土壤濕度檢測 YL-38 + YL-69。2021 年 01 月 06 日，取

自：<https://atceiling.blogspot.com/2017/06/arduinoyl-38-yl-69.html>

Jason Chu 2020 年 2 月 20 日。Arduino & 樹莓派的專家，創客的好朋友！(2020)。2021 年 01 月 06 日，取自：<https://blog.jmaker.com.tw/dht11-lcd/>

阿喵就像家 2015 年 07 月 19 日。深入淺出 Wifi 晶片 ESP8266 with Arduino 2021 年 01 月 06 日，取

自：<https://mlwmlw.org/2015/07/%E6%B7%B1%E5%85%A5%E6%B7%BA%E5%87%BA-wifi-%E6%99%B6%E7%89%87-esp8266-with-arduino/>

