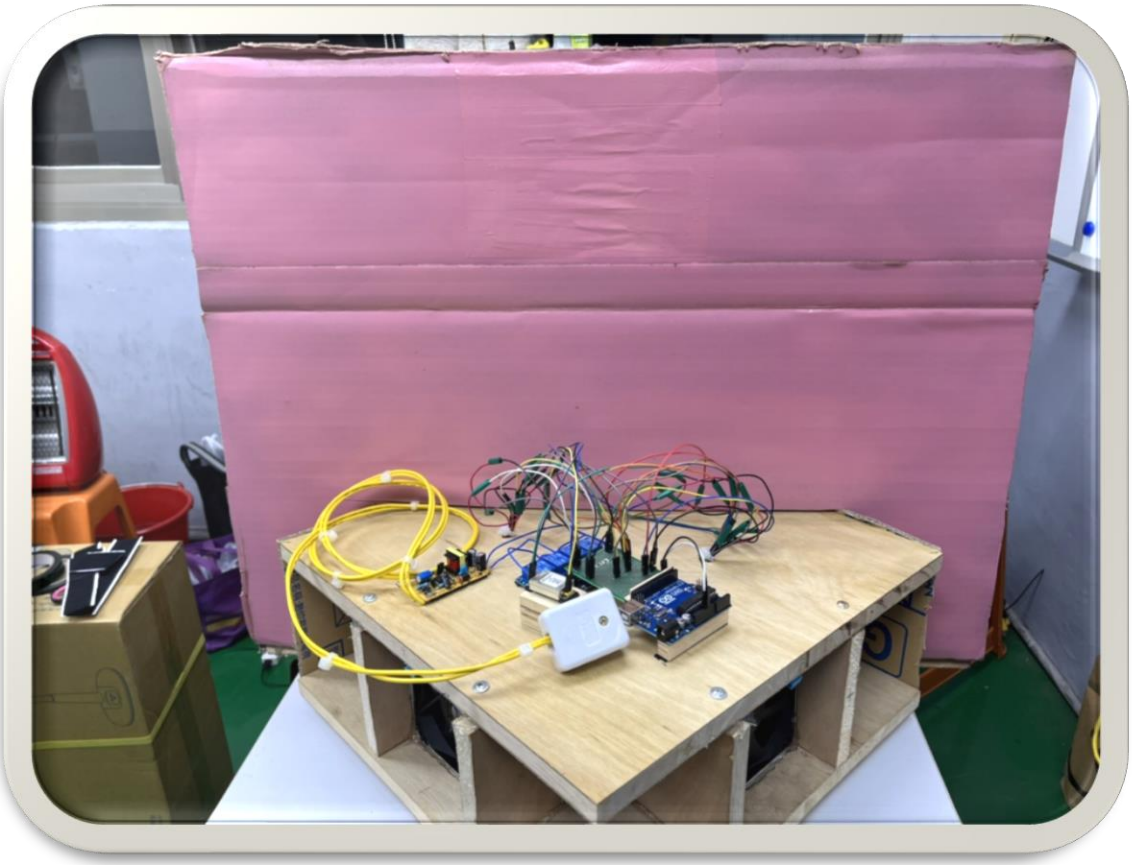
臺北市立大安高級工業職業學校專題製作競賽

「專題組」作品說明書封面



群別:電機電子群

作品名稱:自動化全熱交換器

關鍵字:二氧化碳、熱交換、自動化

目錄

壹、摘要

此專題是模擬夏季開冷氣時的密閉教室空間，自動化偵測空間中二氧化碳濃度，若濃度超標則自動開啟風扇。但引進外氣會大大增加冷氣的耗能，因此我們製作了小型的全熱交換器來進行室內外氣體的熱交換，以減少冷氣的耗能達到省電的目的。

貳、研究動機

炎炎夏日，大多數教室都開著冷氣且緊閉門窗，長時間待在教室上課時，常常發現有人昏昏欲睡，我們推測可能是通風不足造成室內二氧化碳濃度過高。室內空間中二氧化碳濃度只要超過 700ppm 就會產生胸悶頭暈、注意力分散、嗜睡、記憶力減退等症狀，如圖 1 所示。在缺氧的環境下，影響學生的學習效率，為了解決這些問題，勢必要引進新鮮空氣。

圖 1 二氧化碳濃度對照表

參、主題與課程之相關性或教學單元之說明

一、裝置製作

此全熱交換器結合了高一所學的冷凍空調原理，裝置中間的風管利用熱傳導的原理，將冷熱空氣利用交錯風管來降低溫度。二氧化碳濃度是結合了人體舒適度，利用二氧化碳感測器讀取空氣中二氧化碳的濃度，進一步改善空氣品質。

二、電路設計

在高一的基本電學實習及高二的電子學實習中，學習了電子元件的使用和電子電路的配線，這次的專題我們使用洞洞板燒焊電路以及排針使其能插上杜邦線，相較於使用麵包版更穩定、美觀且省空間。

三、程式撰寫

在高三的程式設計課程中，學習如何用程式語言使各式家電達到自動化控制。我們運用這門課的內容控制 DHT-11 溫溼度傳感器和二氧化碳感測器數值紀錄，且讓繼電器在二氧化碳濃度達到標準時啟動風扇。

肆、研究方法(過程)

一、研究流程

我們在 7 月決定了專題題目後，便開始收集資與購買材料，接著同時配合熱交換裝置製作、外觀設計、電路設計、程式撰寫後，將全熱交換器與自動化電路控制完全整合，最後製作模擬空間進行功能測試，完成專題作品。其專題製作時間分配及研究步驟分別如下表 1 及圖 2 所示：

表 1 時間分配表

	8 月	9 月	10 月	11 月	12 月
1. 蒐集資料					
2. 購買材料					
3. 全熱交換器製作					
4. 程式撰寫					
5. 電路設計					
6. 模擬空間製作					
7. 成品測試					

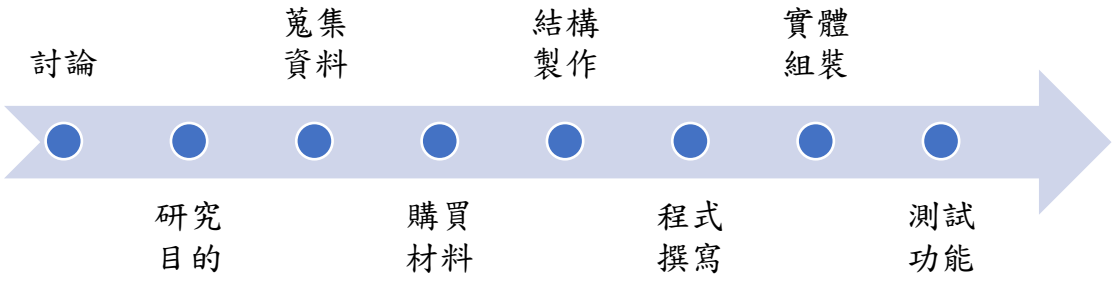


圖 2 研究步驟

二、使用材料及工具

(一)零件介紹

1、繼電器

繼電器是一種電子控制器件，它具有控制系統和被控制系統，通常應用於自動控制電路中，是用較小的電流去控制較大電流的一種「自動開關」，在電路中起著自動調節、安全保護、轉換電路等作用，如圖 3 所示。

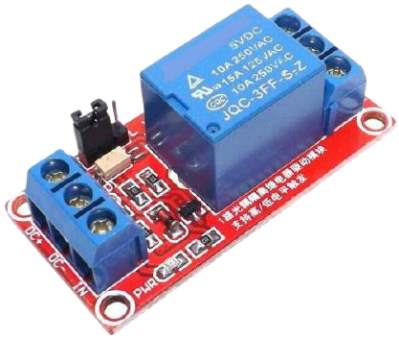


圖 3 繼電器

2、Arduino UNO

Arduino Uno 是一款基於 ATmega328P 的微控制器板。它有 14 個數位輸入/ 輸出接腳，6 個類比輸入，16 MHz 石英晶體，USB 連接孔，電源插孔，ICSP 接頭和重置按鈕，基本規格如表 1.1 所示。 Arduino Uno 板可通過 USB 連接或外部電源供電，如圖 4 所示。

圖 4　Arduino UNO

3、DHT11 溫溼度傳感器

DHT11 是一款經過校准過且直接以數字訊號輸出的溫濕度感測器。內含一個電阻式感濕元件和一個 NTC 測溫元件，並與一個 8bit 單晶片相連接。體積小、功耗低，傳輸距離最遠可達 20 公尺以上，其溫度測量範圍為 0~50℃，誤差在±2℃;溼度的測量範圍為 20%~90%RH(相對溼度),誤差在±5%RH，如圖 5 所示。

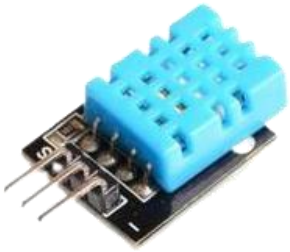


圖 5 DHT11 溫溼度傳感器

4、SenseAir S8 二氧化碳濃度傳感器

SenseAir S8 是一個高精準度的二氧化碳濃度感測元件，尺寸:33.5 x 20 x 8.5 mm、重量:＜8 克、電源:4.5V-25V、偵測範圍:400-2000ppm、工作溫度:0-50℃，如圖 6 所示。

圖 6 SenseAir S8 二氧化碳濃度傳感器

伍、研究結果

一、運作流程

自動化全熱交換器開機後一直重複執行以下動作，自動偵測空間內二氧化碳度是否超過人體不舒適設定值(700ppm)，若超過則全熱交換器的風扇啟動進行氣體交換，待空間中二氧化碳濃度降到人體可接受設定值(450ppm 以下)，則停止動作，其動作流程如下圖 7 所示。

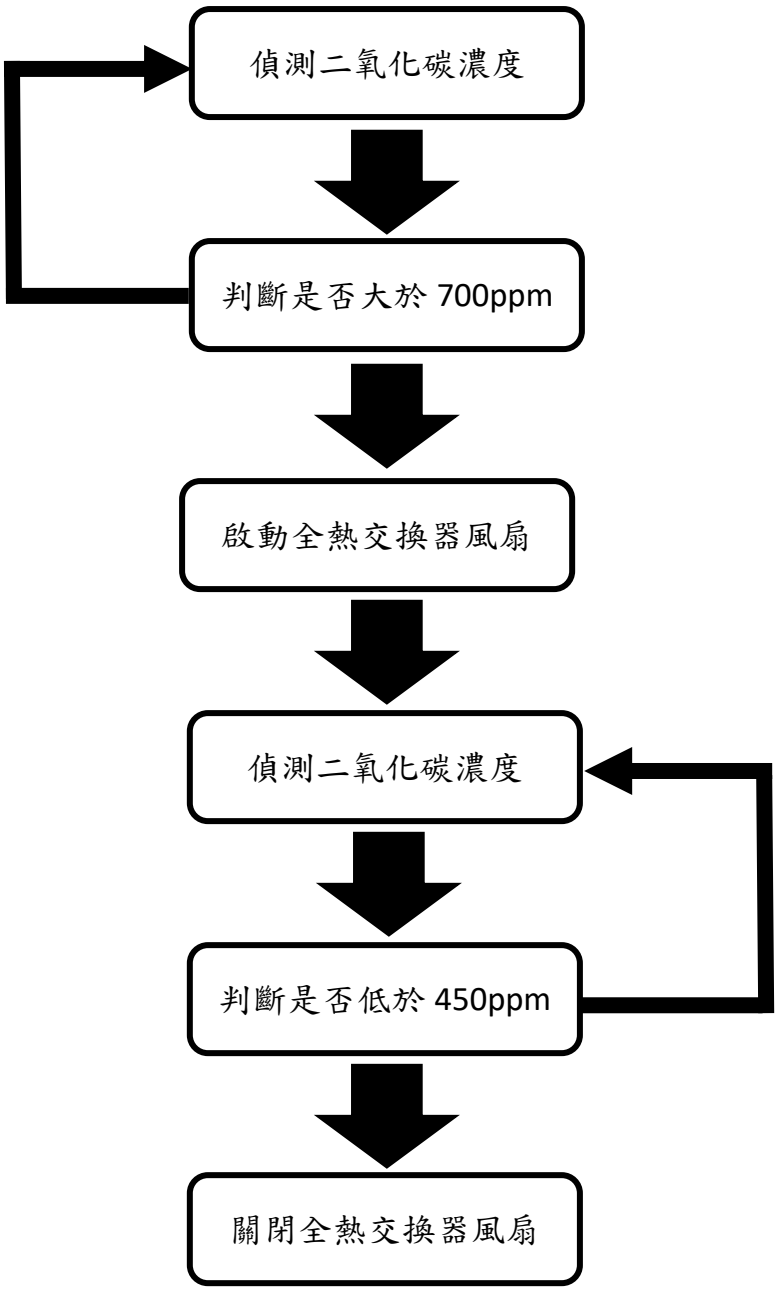


圖 7 運作流程圖

二、結構製作

(一)熱交換裝置

我們用「鋁箔紙」以圓柱捲筒的方式作為熱交換的風道，左右交會的方式堆疊而成，室內室外兩入兩出的換氣設計讓冷熱空氣在中間交會時，排出室內的冷空氣能帶走從室外引進的熱空氣的熱量，使進入室內的空氣溫度有降溫的效果，藉此來降低冷氣的耗能，如圖 8 所示。

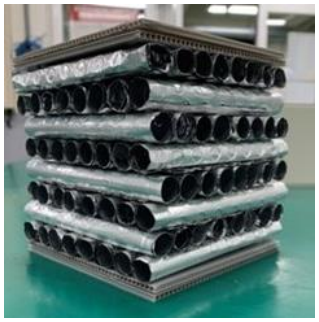


圖 8 鋁箔紙製作熱交換裝置

(二)全熱交換器

在上下部分選擇 40*40(cm^2)的木板並在四個邊中間預留 12cm 可放置風扇，風扇兩側用 L 型木板做出十字風道，並將熱交換裝置放在正中間，如圖 9 及圖 10 所示。

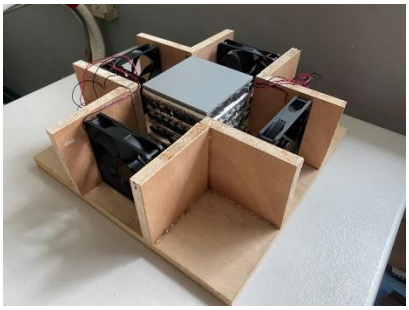


圖 9 全熱交換器(未放置頂板)

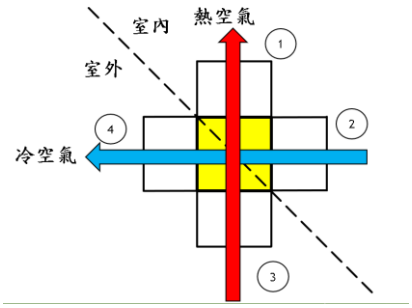


圖 10 內部示意圖

(三)自動化控制電路

我們選用 SenseAir S8 二氧化碳濃度傳感器這來偵測我們室內空間中的二氧化碳濃度，並且分別在四個進出風口的位置裝上 DHT11 溫溼度傳感器來觀測 4 個風口的溫度，以及使用繼電器來控制風扇的起停，最後搭配 Arduino UNO 開發版及程式撰寫來達成自動化，如圖 11 所示。

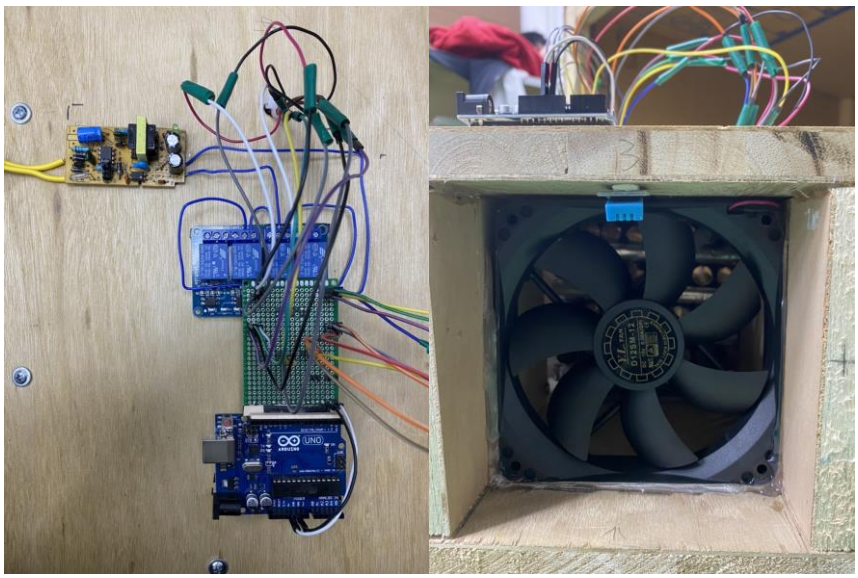


圖 11 電子電路

(四)模擬空間

為了模擬教室的空間，等比例的縮小，教室的大小為 8.5*7*5(m^3)，模擬空間的大小約為 85*70*50(cm^3)，長寬高分別縮小 10 倍，總體積縮小 1000 倍。設計上底板材質為木頭，四腳分別鎖上高為 50cm 的木條，以方便固定；側邊和頂部的材質為紙板，其中一面挖出一個洞來放入全熱交換器，側面挖一個洞以方便對裡面吹氣，來進行測試，如圖 12 所示。

圖 12 模擬空間

三、電路設計

電路主要分成 DC12V、DC5V 兩個迴路，DC12V 是供應給我們四個風口的風扇，而 DC5V 是供應給四個風口的溫度感測器、繼電器以及二氧化碳濃度傳感器作使用，另外除了風扇外的各個元件還個別需要一條數據線，傳輸讀取的數值或給予繼電器高低電位資訊，其中電路圖如下圖 13 及圖 14 所示。

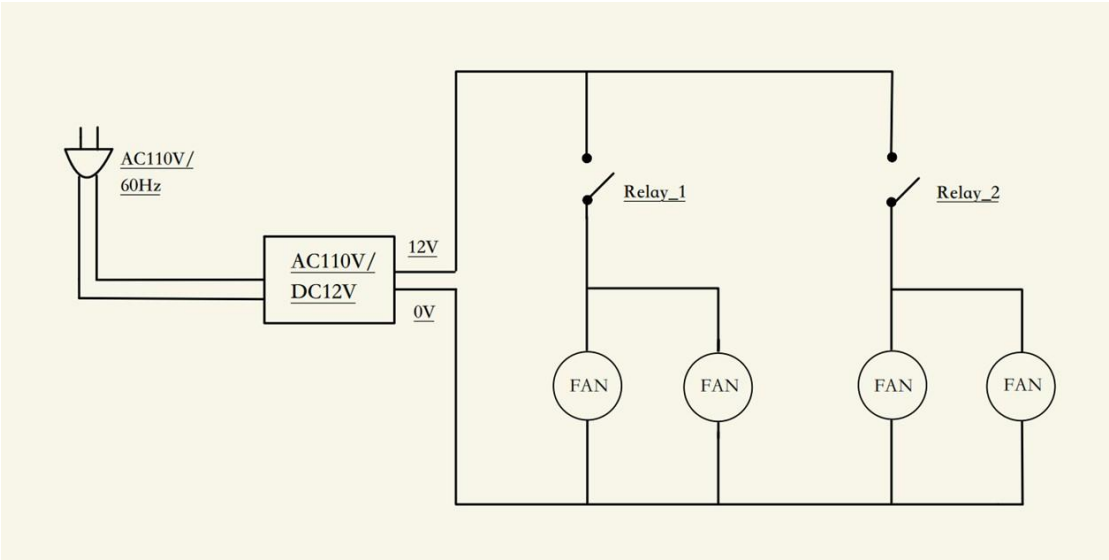


圖 13 風扇迴路電路圖

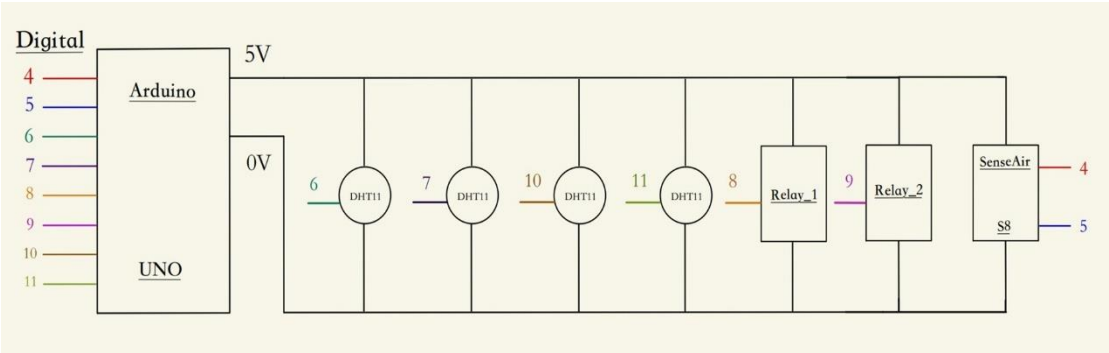


圖 14 感測器迴路電路圖

四、軟體介紹

我們使用 Arduino 這個軟體來撰寫程式並搭配 Arduino UNO 開發版及其他元件達成自動化控制，判讀二氧化碳濃度是否到達指標來控制繼電器，且偵測四個進出風口的溫度，確保熱交換功能正常，其中程式設計代碼如附件 1 所示。

五、實驗數據

等待二氧化碳濃度上升至 700ppm 時，自動化全熱交換器裝置會自行啟動來進行室內室外氣體交換，待濃度降至 450ppm 時會自動停機，如下圖 15 所示，可得知整個運作過程的四個風口溫度變化及二氧化碳濃度變化的曲線，整個換氣過程約為 11 分 30 秒。可以看到引進的氣體溫度室內側(風口 1)明顯低於室外測(風口 3)約 3~4 度; 排出的氣體溫度室內側(風口 2)明顯低於室外測(風口 4)約 3~4 度，得知引進的外氣有達到降溫的效果，且排出的氣體有達到升溫的效果，印證了我們製作的全熱交換器確定可以達到氣體交換及降低進氣的溫度的功能。

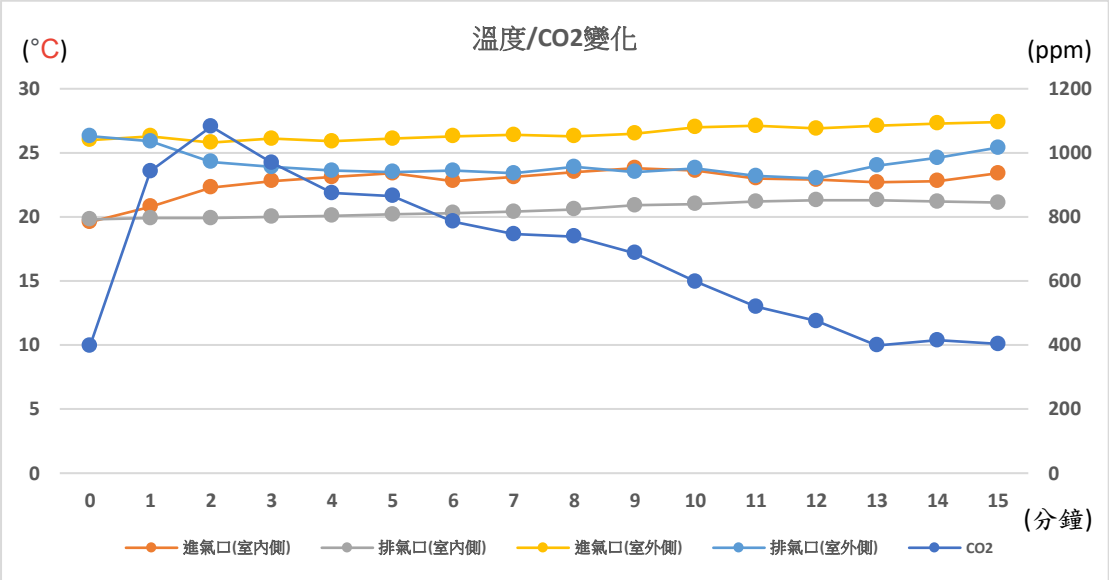


圖 15 實驗數據

陸、問題與討論

- 問題一: 熱交換裝置不穩固

  原因:

  一開始所設計的熱交換器風道為三角形，在黏貼組裝時容易脫落。

  解決方法:

  改變設計，將風道改為圓形，黏貼面積較大，較容易固定。

- 問題二: 穩定室外熱源的問題

  原因:

  因為專題實驗時間為冬天,我們利用吹風機模擬夏天室外的溫度,但進出風口溫度相差太多且供熱溫度不固定，這樣數據紀錄並不合理。

  解決方法:

  更正熱源供應為電暖器,並取適當距離,將入風側溫度定在 30°C。

- 問題三: 穩定室內冷房的問題

  原因:

  原先設定的環境是開著冷氣且門窗緊閉的教室，但實驗開始後發現我們忽略了模擬空間中並沒有安裝空調，模擬空間中溫度隨之上升，一段時間後排出和進入的氣體無法達成熱交換。

  解決方法:

  在模擬空間中安裝製冷晶片，來穩定冷房效果。

柒、結論

根據此專題研究，我們的全熱交換器確定可以達到氣體交換及降低進氣的溫度，在模擬空間下，二氧化碳濃度下降大約為 60%，下降幅度是明顯的，因此可以改善空氣品質;並利用熱交換器使出風口氣體與入風口的溫度大約相差 3~5°C，來降低冷氣的耗能。

由於市面上只有單純的熱交換器，因此我們使它自動化並加上二氧化碳濃度偵測，來改善教室內的二氧化碳濃度，讓熱交換器擁有許多功能。

過程中雖然遇到許多困難，但我們從中學到了解決問題的能力，也了解到團隊合作的重要性，以及把所學的課程學以致用，來完成此專題。

捌、參考文獻

[1]空調吸附除濕節能應用技術，shorturl.at/duFMR。

[2]二氧化碳濃度對人的影響，https://reurl.cc/dX0VLg。

[3]Arduino 空氣品質監測應，https://reurl.cc/g0784X。

[4]SenseAir S8 參考程式，shorturl.at/tvBU2 及 shorturl.at/orDF1。

附件 1 Arduino Uno 程式自動判斷代碼

```
#include <CRCx.h>
#include <SoftwareSerial.h>
#include <DHT.h>
#define SenseAir_S8
#define DHTTYPE DHT11
//=====Pin=====
const byte PIN_TX = 4;  //SenseAir S8
const byte PIN_RX = 5;  //SenseAir S8
int  DHTPIN1 = 11;
int  DHTPIN2 = 7;
int  DHTPIN3 = 6;
int  DHTPIN4 = 10;
int  Relay1 = 8;
int  Relay2 = 9;
//=====CO2 Senser=====
const byte MB_PKT_7 = 7;
const byte MB_PKT_8 = 8;
const byte cmdReSA[MB_PKT_8] = {0xFE, 0X04, 0X00, 0X03, 0X00, 0X01,
0XD5, 0XC5};
const byte cmdOFFs[MB_PKT_8] = {0xFE, 0x06, 0x00, 0x1F, 0x00, 0x00,
0xAC, 0x03};
const byte cmdCal1[MB_PKT_8] = {0xFE, 0x06, 0x00, 0x00, 0x00, 0x00,
0x9D, 0xC5};
const byte cmdCal2[MB_PKT_8] = {0xFE, 0x06, 0x00, 0x01, 0x7C, 0x06,
0x6C, 0xC7};
const byte cmdCal3[MB_PKT_8] = {0xFE, 0x03, 0x00, 0x00, 0x00, 0x01,
0x90, 0x05};
const byte cmdCalR[MB_PKT_7] = {0xFE, 0x03, 0x02, 0x00, 0x20, 0xAD,
0x88};
static byte response[MB_PKT_8] = {0};
SoftwareSerial co2sensor(PIN_RX, PIN_TX);
#define BAUDRATE 9600
byte ConnRetry = 0;
int CO2 = 0;
unsigned int crc_cmd;
int sensor_state=0;
```

```
void CO2iniSenseAir()  //初始化
{
  co2sensor.write(cmdOFFs, MB_PKT_8);
  co2sensor.readBytes(response, MB_PKT_8);
  Serial.print(F("Deactivate Automatic Self-Calibration SenseAir S8:
"));
  CheckResponse(cmdOFFs, response, MB_PKT_8);
  delay(2000);
  while (co2SenseAir() == 0 && (ConnRetry < 5)){
    BadConn();
  }
  if (ConnRetry == 5){
  Serial.println(F(" Air sensor detected AirSense S8 Modbus"));
  delay(10);
  Serial.println(F("SenseAir read OK"));
  delay(4000);
  }
}
void CheckResponse(uint8_t *a, uint8_t *b, uint8_t len_array_cmp)  //
完成與失敗修訂
{
  bool check_match = false;
  for (int n = 0; n < len_array_cmp; n++){
    if (a[n] != b[n]){
      check_match = false;
      break;
    }
    else{
      check_match = true;
    }
  }
  if (check_match){
    Serial.println(F("CheckResponse done"));
  }
  else{
    Serial.println(F("CheckResponse failed"));
  }
}
```

```
int co2SenseAir()  //讀取
{
  int co2=0;
  static byte responseSA[MB_PKT_7] = {0};
  memset(responseSA, 0, MB_PKT_7);
  co2sensor.write(cmdReSA, MB_PKT_8);
  co2sensor.readBytes(responseSA, MB_PKT_7);
  co2 = (256 * responseSA[3]) + responseSA[4];
  if (co2 != 0){
    crc_cmd = crcx::crc16(responseSA, 5);
    if (responseSA[5] == lowByte(crc_cmd) && responseSA[6] ==
highByte(crc_cmd)){
      return co2;
    }
    else{
      while (co2sensor.available() > 0){
        char t = co2sensor.read();
      }
      Serial.println(F("FAIL CRC_CO2"));
      return 0;
    }
  }
  else{
    Serial.println(F("FAILED"));
    return 0;
  }
}
void BadConn()  //訊號檢測
{
  Serial.println("Air sensor not detected. Please check wiring...
Try# " + String(ConnRetry));
  if (ConnRetry > 1){
    delay(20);
  }
  delay(2500);
  ConnRetry++;
}
void CalibrationSenseAir()  //校準
```

```
{
  sensor_state=1;
  byte responseSA[MB_PKT_7] = {0};
  delay(100);
  //Step 1 Calibration
  co2sensor.write(cmdCal1, MB_PKT_8);
  co2sensor.readBytes(response, MB_PKT_8);
  Serial.print(F("Calibration Step1: "));
  CheckResponse(cmdCal1, response, MB_PKT_8);
  delay(1000);
  //Step 2 Calibration
  co2sensor.write(cmdCal2, MB_PKT_8);
  co2sensor.readBytes(response, MB_PKT_8);
  Serial.print(F("Calibration Step2: "));
  CheckResponse(cmdCal2, response, MB_PKT_8);
  delay(4000);
  //Step 3 Calibration
  co2sensor.write(cmdCal3, MB_PKT_8);
  co2sensor.readBytes(responseSA, MB_PKT_7);
  Serial.print(F("Resetting forced calibration factor to 400: "));
  CheckResponse(cmdCalR, responseSA, MB_PKT_7);
  delay(1000);
  sensor_state=0;
}
void Preheat()  //預熱
{
  Serial.print(F("Preheat: "));
  for (int i = 30; i > -1; i--){
    Serial.println("HEAT");
    delay(1000);
    Serial.println(i);
    delay(1000);
    i--;
  }
}

//=====Dht11=====
void view1()
```

```
{
  DHT dht(DHTPIN1,DHTTYPE);
  dht.begin();
  float t = dht.readTemperature();
  //Serial.print("DHT11_1:");
  Serial.print(t);
  //Serial.print("℃");
  Serial.print(",");
}
void view2()
{
  DHT dht(DHTPIN2,DHTTYPE);
  dht.begin();
  float t = dht.readTemperature();
  //Serial.print("DHT11_2:");
  Serial.print(t);
  //Serial.print("℃");
  Serial.print(",");
}
void view3()
{
  DHT dht(DHTPIN3,DHTTYPE);
  dht.begin();
  float t = dht.readTemperature();
  //Serial.print("DHT11_3:");
  Serial.print(t);
  //Serial.print("℃");
  Serial.print(",");
}
void view4()
{
  DHT dht(DHTPIN4,DHTTYPE);
  dht.begin();
  float t = dht.readTemperature();
  //Serial.print("DHT11_4:");
  Serial.print(t);
  //Serial.print("℃");
  Serial.println(",");
```

```
}
//=====Relay=====
void RelayON()
{
  Serial.println("FAN_State:Turning_ON  Wait...");
  digitalWrite(Relay1,LOW);
  delay(7000);
  digitalWrite(Relay2,LOW);
  Serial.println("==================================");
}
void RelayOFF()
{
  Serial.println("FAN_State:Turning_OFF");
  digitalWrite(Relay1,HIGH);
  digitalWrite(Relay2,HIGH);
  Serial.println("==================================");
}
//=====Test=====
void test()
{
  RelayON();
  int num =1;
  while(num<1){
    view1();
    view2();
    view3();
    view4();
    Serial.println("==================================");
    delay(10000);
  }
}
//=====Main=====
void setup(){
  Serial.begin(9600);
  co2sensor.begin(BAUDRATE);
  pinMode(DHTPIN1,INPUT);
  pinMode(DHTPIN2,INPUT);
  pinMode(DHTPIN3,INPUT);
```

```
  pinMode(DHTPIN4,INPUT);
  pinMode(Relay1,OUTPUT);
  pinMode(Relay2,OUTPUT);
  RelayOFF();
  CO2iniSenseAir();
  //Preheat();
  //CalibrationSenseAir();
  //test();
  //RelayON();
}
void loop(){
  int x =co2SenseAir();

  if(x>=700){
    RelayON();
    while(x>=450){
      x = co2SenseAir();
      //Serial.print("CO2:");
      Serial.print(",");
      Serial.print(x);
      //Serial.print("ppm");
      Serial.print(",");
      //Serial.println("FAN_State:ON");
      view1();
      view2();
      view3();
      view4();
      //Serial.println("==================================");
      delay(10000);
    }
    RelayOFF();
  }
  else{
    x = co2SenseAir();
    //Serial.print("CO2:");
    Serial.print(",");
    Serial.print(x);
    //Serial.print("ppm");
```

```
    Serial.print(",");
    //Serial.println("FAN_State:OFF");
    view1();
    view2();
    view3();
    view4();
    //Serial.println("==================================");
    delay(10000);
  }
}
```