



南臺科技大學
Southern Taiwan University of Science and Technology

AHT15 溫濕度感測器 數據回傳序列埠



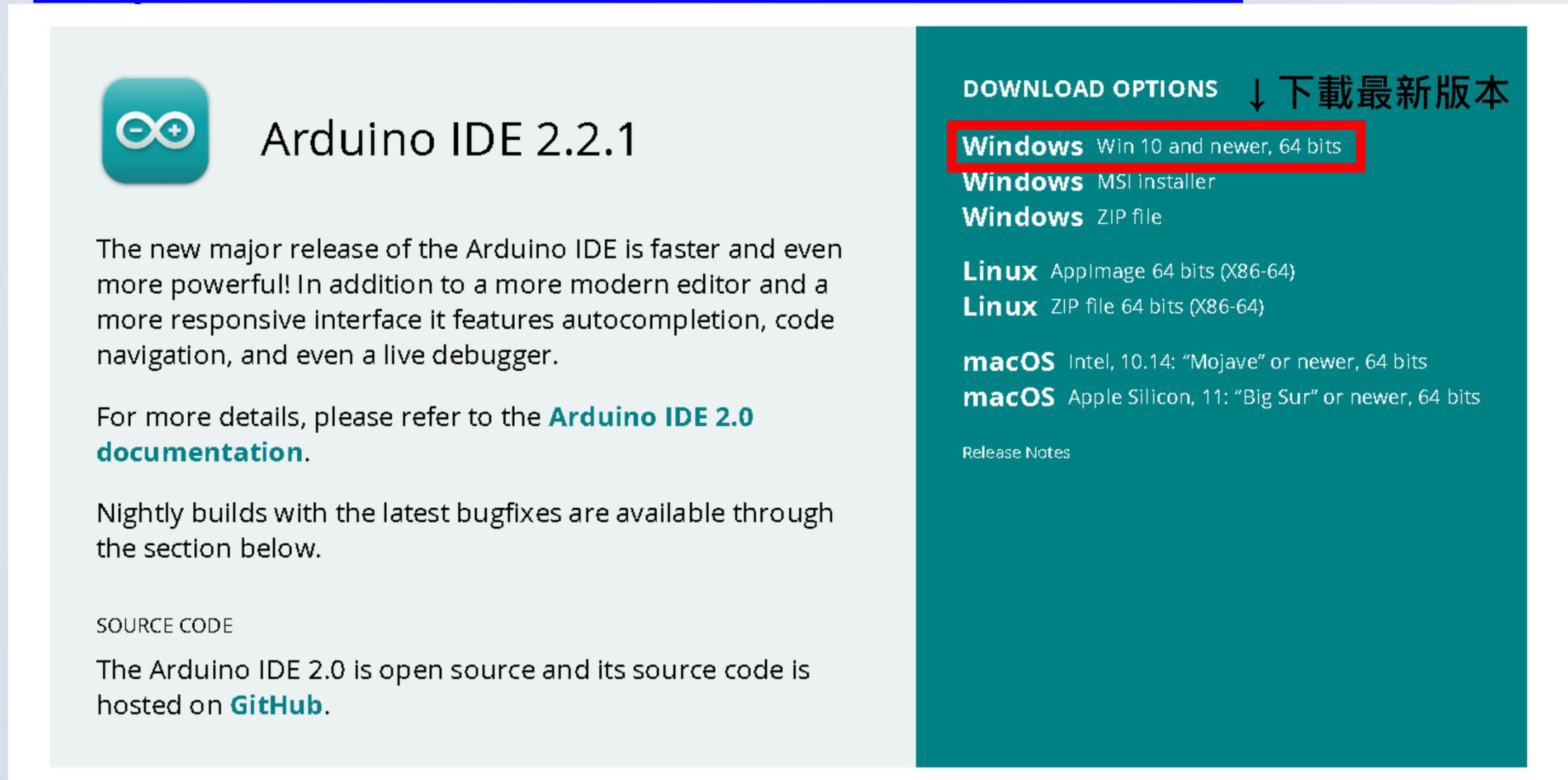
古都土城仔綠電創能與智動養殖
之跨界整合永續淨零發展計畫

目錄

- ESP 32 環境設定
- AHT15(溫濕度感測器)介紹
- ESP32 v.s AHT15 電路連接：
- 全部程式碼
- 程式講解
- 最終執行成果

ESP 32 環境設定

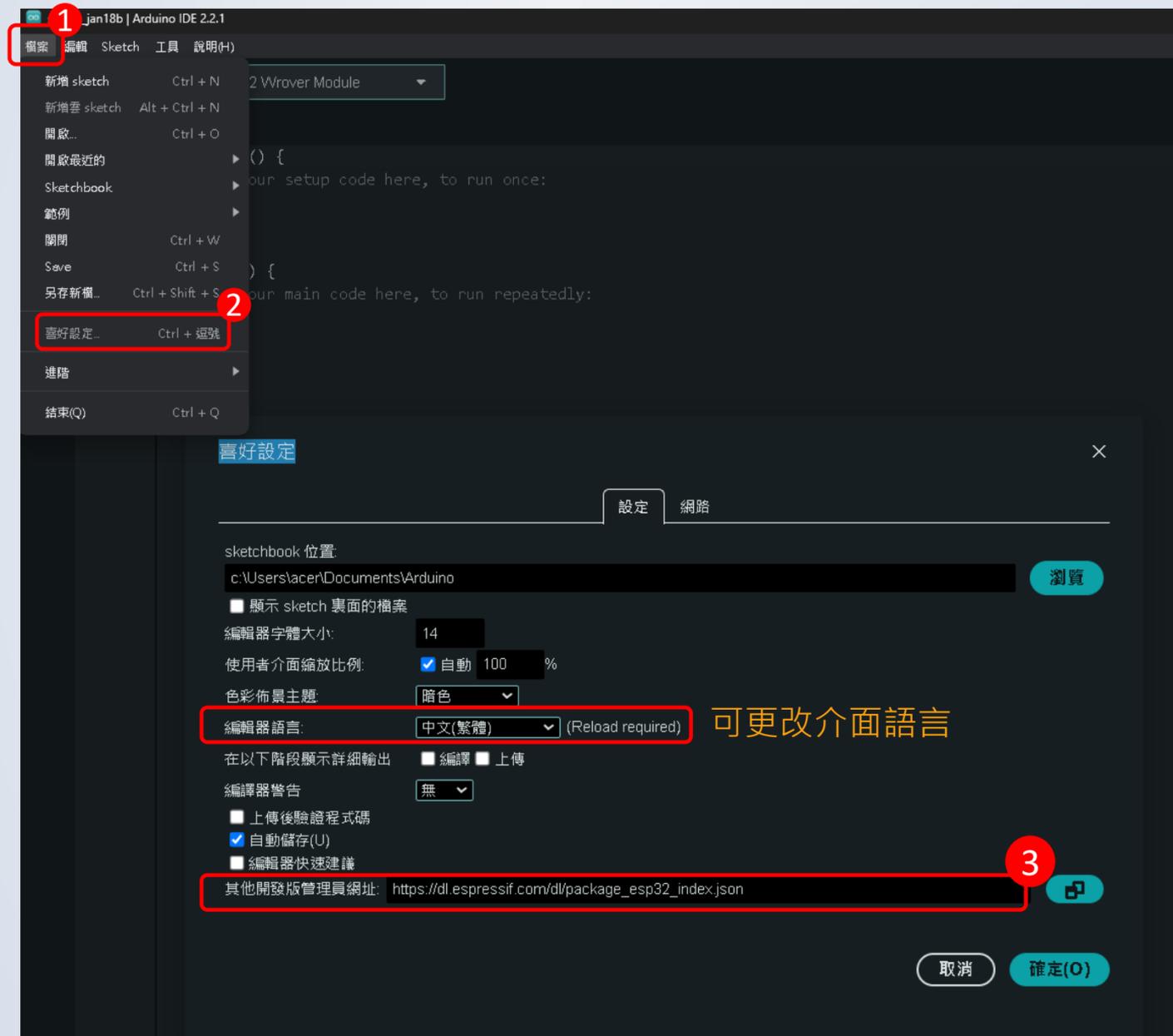
- 本次所使用的程式開發環境為Arduino IDE
- <https://www.arduino.cc/en/software>



The screenshot shows the Arduino IDE 2.2.1 download page. On the left, there is a teal icon with a minus and plus sign, followed by the text "Arduino IDE 2.2.1". Below this, a paragraph describes the new major release as faster and more powerful, featuring a modern editor, autocompletion, code navigation, and a live debugger. It also mentions that for more details, users should refer to the "Arduino IDE 2.0 documentation". A section for "Nightly builds" indicates that the latest bugfixes are available through a specific section. At the bottom, it states that the Arduino IDE 2.0 is open source and its code is hosted on GitHub.

On the right, the "DOWNLOAD OPTIONS" section is highlighted in teal. It includes a downward arrow and the text "下載最新版本". Below this, there are three rows of download options for Windows, Linux, and macOS. The first row for Windows is highlighted with a red box and includes the text "Win 10 and newer, 64 bits". The other rows list "MSI installer", "ZIP file", "ApplImage 64 bits (X86-64)", and "Apple Silicon, 11: 'Big Sur' or newer, 64 bits". A "Release Notes" link is also present at the bottom of this section.

由於ESP32 並非Arduino 原廠製作的裝置，而是樂鑫開發的相容裝置，因此必須再安裝ESP32 相容性套件，才能讓Arduino IDE 認識ESP32 晶片。



下載好後，找到主程式arduino.exe，直接點兩下執行Arduino IDE 開啟主程式後，選擇功能表的檔案/ 偏好設定，開啟偏好設定視窗。在額外開發板管理員輸入以下網址：

https://dl.espressif.com/dl/package_esp32_index.json



接著 點選工具/ 開發板/ 開發板管理員，即出現開發板管理員視窗，輸入關鍵字 ESP32，即可篩選ESP32 相容性套件。找到ESP32 Arduino 並往下拉即可在下方找到最新的ESP32 系列主板。大部分的ESP32 開發板都可以擇通用型號「ESP32WroverModule」來開發即可。上傳速度「Upload speed」請務必保持921600。

CH341SER.EXE

CH341SER.EXE

适用范围	版本	上传时间	资料大小
CH340G, CH340T, CH340C, CH340N, CH340K, CH340E, CH340B, CH341A, CH341F, CH341T, CH341B, CH341C, CH341U	3.8	2023-03-16	642KB

USB转串口Windows一键式安装驱动程序, 支持CH340和CH341, 支持32/64位Windows 11/10/8.1/8/7/VISTA/XP, SERVER 2022/2019/2016/2012/2008/2003, 2000/ME/98, 通过微软数字签名认证, 支持USB转UART的3线和9线SERIAL 串口等, 用于随产品发行到最终用户。

下载

完成環境設定後，可以透過MicroUSB 線連接ESP32 及電腦USB 插槽，當插入USB 後，出現需要安裝USB 驅動程式，目前ESP32 主要有CP210x 或CH340 兩種版本，而傳統UNO 則是CH340，目前windows 或MAC 系統沒有內建驅動程式，因此當接上ESP32 的USB 後，尚需安裝驅動程式。

驅動程式網址:

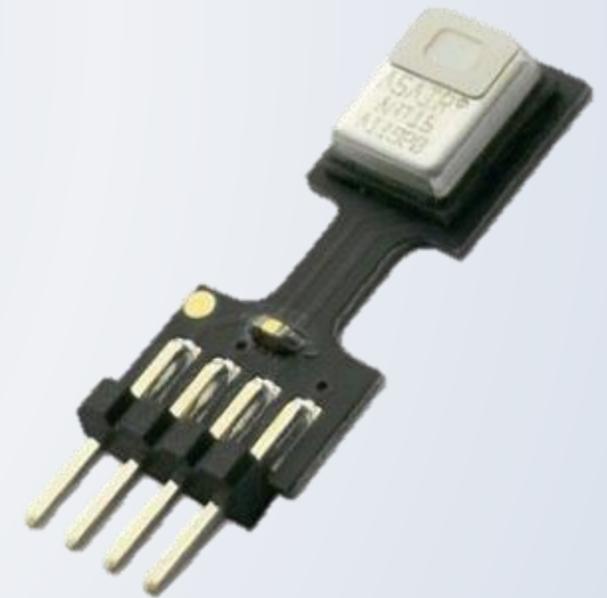
https://www.wch.cn/downloads/CH341SER_EXE.html

完成上述的設定之後，ESP32 的Arduino 開發環境算是安裝完畢。

AHT15(溫濕度感測器)介紹

ATH15溫濕度感測器擁有以下特點:

- 1.具有防水防塵性能，使得它能在惡劣環境下使用。
- 2.有數字輸出，並使用I²C介面。
- 3.擁有優異的長期穩定性。
- 4.溫度的精度 $\pm 0.3^{\circ}\text{C}$ ，濕度在 25°C 環境下精度 $\pm 2\%$ 。



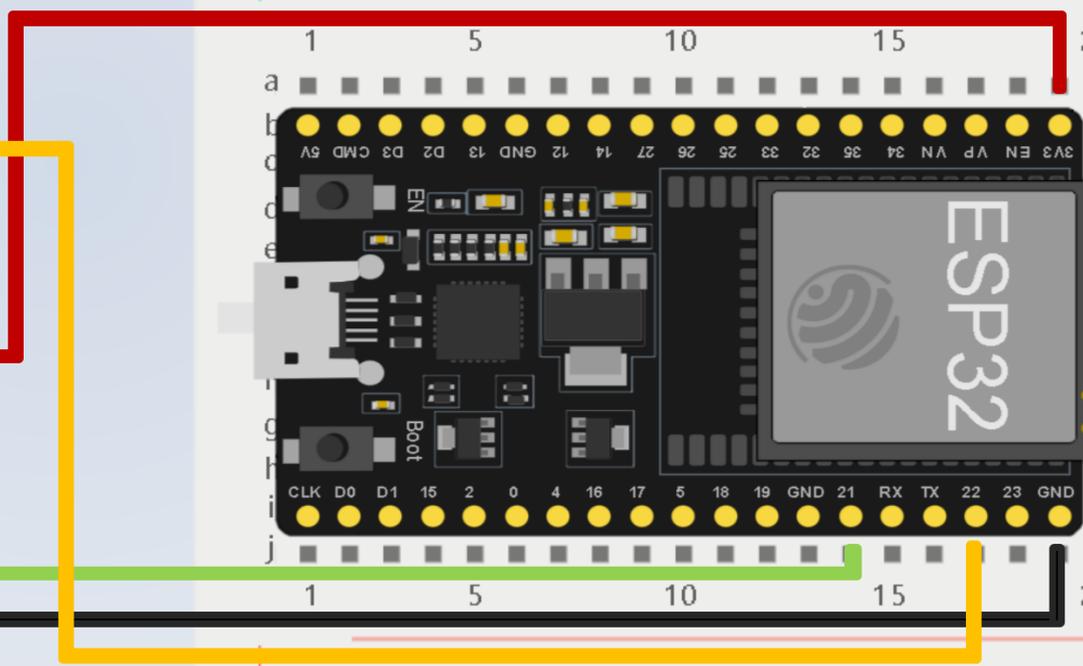
ESP32 v.s AHT15 電路連接：

1. 連接電源：連接VCC腳位到 ESP32 的 3.3V 腳位。
2. GND 腳位連接到 ESP32 的 GND腳位。
3. I²C 連接：連接SDA 腳位到 ESP32 的 SDA 腳位。
4. SCL 腳位連接到 ESP32 的 SCL 腳位。

		3.3V	19		19	GND		
	RESET	EN	18		18	GPIO23	VSPI MOSI	
		GPIO36	17		17	GPIO22	Wire SCL	
	ADC0		GPIO39	16	16	GPIO1	TX0	Serial TX
	ADC3		GPIO34	15	15	GPIO3	RX0	Serial RX
	ADC6		GPIO35	14	14	GPIO21	Wire SDA	
	ADC7		GPIO32	13	13	GND		
TOUCH9	ADC4		GPIO33	12	12	GPIO19	VSPI MISO	
TOUCH8	ADC5		GPIO25	11	11	GPIO18	VSPI SCK	
DAC1	ADC18		GPIO26	10	10	GPIO5	VSPI SS	
DAC2	ADC19		GPIO27	9	9	GPIO17		
TOUCH7	ADC17		GPIO14	8	8	GPIO16		
TOUCH6	ADC16	HSPI SCK	GPIO12	7	7	GPIO4		ADC10 TOUCH0
TOUCH5	ADC15	HSPI MISO	GND	6	6	GPIO0	BOOT	ADC11 TOUCH1
						GPIO2		ADC12 TOUCH2
TOUCH4	ADC14	HSPI MOSI	GPIO13	5	5	GPIO15	HSPI SS	ADC13 TOUCH3
		FLASH D2	GPIO9	4	4	GPIO8	FLASH D1	
		FLASH D3	GPIO10	3	3	GPIO7	FLASH D0	
		FLASH CMD	GPIO11	2	2	GPIO6	FLASH SCK	
		5V		1	1			



SCL
GND
SDA
VCC



注意事項：

1. AHT15 的工作電壓一般是 3.3V，如果連接到 5V 的電源，可能會導致感測器損壞。
2. ESP32 的 SDA 和 SCL 腳位通常是 GPIO21 (SDA) 和 GPIO22 (SCL)。

全部程式碼如下：

```
#include <Wire.h> // 匯入Wire函式庫，用於I2C通訊

#define AHT15_ADDRESS 0x38 // 定義AHT15感測器的I2C地址
#define AHT15_RESET 0xBA // 定義重置命令的值
#define AHT15_INIT 0xBE // 定義初始化命令的值
#define AHT15_START_MEASUREMENT 0xAC // 定義開始測量命令的值

void setup() { // 設定函數，在程式開始時執行一次
  Wire.begin(); // 啟用Wire函式庫
  Serial.begin(115200); // 啟動序列通訊鮑率115200

  // Reset sensor
  Wire.beginTransmission(AHT15_ADDRESS); // 開始向AHT15_ADDRESS發送數據
  Wire.write(AHT15_RESET); // 發送重置命令
  if (Wire.endTransmission() != 0) { // 如果傳輸結束並返回錯誤
    Serial.println("重置錯誤"); // 顯示錯誤信息
    while(1); // 進入無窮迴圈
  }
  delay(20); // 等待20毫秒
```

全部程式碼如下：

```
// Initialize sensor
Wire.beginTransmission(AHT15_ADDRESS); // 開始向AHT15_ADDRESS發送數據
Wire.write(AHT15_INIT); // 發送初始化命令
Wire.write(0x08); // 發送額外的初始化數據
Wire.write(0x00); // 發送額外的初始化數據
if (Wire.endTransmission() != 0) { // 如果傳輸結束並返回錯誤
    Serial.println("初始化錯誤"); // 顯示錯誤信息
    while(1); // 進入無窮迴圈
}
delay(10); // 等待10毫秒
}
void loop() { // 主迴圈，會不斷重複執行
    // Start measurement
    Wire.beginTransmission(AHT15_ADDRESS); // 開始向AHT15_ADDRESS發送數據
    Wire.write(AHT15_START_MEASUREMENT); // 發送開始測量命令
    Wire.write(0x33); // 發送額外的開始測量數據
    Wire.write(0x00); // 發送額外的開始測量數據
    if (Wire.endTransmission() != 0) { // 如果傳輸結束並返回錯誤
        Serial.println("測量失敗"); // 顯示錯誤信息
        delay(1000); // 等待1000毫秒
        return; // 返回，跳過此次迴圈的剩餘部分
    }
    delay(80); // 等待80毫秒以完成測量
```

全部程式碼如下：

```
// Read data
Wire.requestFrom(AHT15_ADDRESS, 6); // 從AHT15_ADDRESS請求6個字節的數據
if (Wire.available() != 6) { // 如果可用數據不是6個字節
    Serial.println("讀取錯誤"); // 顯示錯誤信息
    return; // 返回，跳過此次迴圈的剩餘部分
}
uint8_t data[6]; // 創建一個用於存儲數據的陣列
for (int i = 0; i < 6; i++) { // 對於每個字節
    data[i] = Wire.read(); // 讀取並存儲數據
}

// Calculate humidity and temperature
// 計算原始濕度值
uint32_t rawHumidity = ((uint32_t)data[1] << 12) | ((uint32_t)data[2] << 4) | (data[3] >> 4);
float humidity = rawHumidity * 100.0 / (1 << 20); // 轉換原始濕度值為百分比

// 計算原始溫度值
uint32_t rawTemperature = (((uint32_t)data[3] & 0xF) << 16) | ((uint32_t)data[4] << 8) |
data[5];

float temperature = ((200.0 * rawTemperature) / (1 << 20)) - 50; // 轉換原始溫度值為攝氏度
```

全部程式碼如下：

```
// Print results
Serial.print("Humidity:");
Serial.print(humidity); // 顯示濕度值
Serial.print("% , ");
Serial.print("Temperature: ");
Serial.print(temperature); // 顯示溫度值
Serial.println("°C");

delay(2000); // 等待2秒
}
```

程式講解

```
1 #include <Wire.h>
2
3 #define AHT15_ADDRESS 0x38
4 #define AHT15_RESET 0xBA
5 #define AHT15_INIT 0xBE
6 #define AHT15_START_MEASUREMENT 0xAC
```

`#include <Wire.h>`

這行代碼引入了Wire程式庫，該庫允許ESP32通過 I²C 通訊協議與 I²C 設備進行通信。

`#define AHT15_ADDRESS 0x38`

`#define AHT15_RESET 0xBA`

`#define AHT15_INIT 0xBE`

`#define AHT15_START_MEASUREMENT 0xAC`

這段程式碼分別代表定義了AHT15的重置、初始化、測量、開始測量等命令的十六進制值。

程式講解

```
8 void setup() {
9   Wire.begin(); // 啟用Wire函式庫
10  Serial.begin(115200); // 啟動序列通訊鮑率115200
11
12  // Reset sensor
13  Wire.beginTransmission(AHT15_ADDRESS);
14  Wire.write(AHT15_RESET);
15  if (Wire.endTransmission() != 0) {
16    Serial.println("重置錯誤");
17    while(1);
18  }
19  delay(20);
20
21  // Initialize sensor
22  Wire.beginTransmission(AHT15_ADDRESS);
23  Wire.write(AHT15_INIT);
24  Wire.write(0x08);
25  Wire.write(0x00);
26  if (Wire.endTransmission() != 0) {
27    Serial.println("初始化錯誤");
28    while(1);
29  }
30  delay(10);
31 }
```

重置程序

偵錯程序

- void setup() 副程式只會在啟動板子後執行一次。
- **Serial.begin** 代表啟動序列監控視窗，能顯示文字或數據。
- 板子與序列監控視窗的傳輸鮑率（Baud）須設定同為115200，否則會出現亂碼。
- 程式碼包含重置及初始化的偵錯程序。



```
8 void setup() { // 設定函數，在程式開始時執行一次
9   Wire.begin(); // 啟用Wire函式庫
10  Serial.begin(115200); // 啟動序列通訊鮑率115200
11 }
```

輸出 序列埠監控窗 ×

訊息 (按 Enter 鍵將訊息發送到 COM5 上的 ESP32 Wrover Module) 換行 115200鮑率

```
16:07:57.768 -> Humidity: 57.03% , Temperature: 22.56°C
16:07:59.857 -> Humidity: 57.14% , Temperature: 22.53°C
16:08:01.942 -> Humidity: 57.22% , Temperature: 22.51°C
16:08:04.019 -> Humidity: 57.31% , Temperature: 22.49°C
16:08:06.101 -> Humidity: 57.35% , Temperature: 22.47°C
```

程式講解

```
33 void loop() {
34     // Start measurement
35     Wire.beginTransmission(AHT15_ADDRESS);
36     Wire.write(AHT15_START_MEASUREMENT);
37     Wire.write(0x33);
38     Wire.write(0x00);
39     if (Wire.endTransmission() != 0) {
40         Serial.println("測量失敗");
41         delay(1000);
42         return;
43     }
44     delay(80);
45
46     // Read data
47     Wire.requestFrom(AHT15_ADDRESS, 6);
48     if (Wire.available() != 6) {
49         Serial.println("讀取錯誤");
50         return;
51     }
```

`void loop()` 迴圈副程式在執行完`setup` 之後，就會執行`loop` 內"`{ }`" 的程式碼，要注意的是Arduino的`loop` 是沒有停止條件的，也就是說這是一個無窮迴圈。

第一段是在寫板子給感測器的開始測量指令是否正常，當數據錯誤將顯示錯誤訊息。

第二段程式碼是在讀取從感測器返回的數據。先請求6個字節的數據，再檢查是否有6個字節的數據可用。如果不是，它會打印出一個錯誤訊息。

程式講解

上半段程式碼的目的是從感測器讀取數據並將其存儲在一個陣列中。

中段的程式碼是將感測器的原始濕度值數據進行轉換

`float humidity = rawHumidity * 100.0 / (1 << 20);` 是在將原始濕度值轉換為百分比。

下半段程式碼是將感測器的原始溫度值轉換為攝氏度。

`float temperature = ((200.0 * rawTemperature) / (1 << 20)) - 50;`

是在將原始溫度值映射到一個範圍為-50到150的範圍內。

```
52  uint8_t data[6];
53  for (int i = 0; i < 6; i++) {
54      data[i] = Wire.read();
55  }
56
57  // Calculate humidity and temperature
58  uint32_t rawHumidity = ((uint32_t)data[1] << 12) | ((uint32_t)data[2] << 4) | (data[3] >> 4);
59  float humidity = rawHumidity * 100.0 / (1 << 20);
60
61  uint32_t rawTemperature = (((uint32_t)data[3] & 0xF) << 16) | ((uint32_t)data[4] << 8) | data[5];
62  float temperature = ((200.0 * rawTemperature) / (1 << 20)) - 50;
```

程式講解

這段程式碼是再輸前面所運算出的溫濕度，並顯示在序列埠監視窗，等待兩秒後使整個程式重新執行，無限循環。

```
64 // Print results
65 Serial.print("Humidity: ");
66 Serial.print(humidity);
67 Serial.print("% , ");
68 Serial.print("Temperature: ");
69 Serial.print(temperature);
70 Serial.println("°C");
71
72 delay(2000);
```

最終執行成果

```
12:50:44.643 -> Humidity: 47.79% , Temperature: 22.80°C
12:50:46.727 -> Humidity: 47.89% , Temperature: 22.79°C
12:50:48.820 -> Humidity: 48.08% , Temperature: 22.81°C
12:50:50.879 -> Humidity: 48.08% , Temperature: 22.80°C
12:50:52.965 -> Humidity: 48.62% , Temperature: 22.80°C
```



感謝聆聽 給予指 導

減碳漁

古都土城仔綠電創能與智動養殖
之跨界整合永續淨零發展計畫