

Hw6

目的:利用Raspberry Pi獲取溫濕度資料並將其顯示於前端

Raspberry Pi 4 刷機

Arduino IDE 安裝

運行Flask server

▼ Raspberry Pi 4 & Arduino IDE 安裝

選擇安裝環境(Raspberry Pi)



安裝DHT11庫

README Code of conduct MIT license Security

DHT11 Arduino Library

Author Dhruba Saha release v2.1.0 license MIT contributors 1 downloads 305 DOI 10.5281/zenodo.10633701

Table of Contents

- [DHT11 Arduino Library](#)
 - [Table of Contents](#)
 - [Introduction](#)
 - [Updates](#)
 - [v2.1.0 Hotfix Latest](#)
 - [v2.0.0 Update](#)
 - [How It Works](#)
 - [Internal Protocol Handling](#)
 - [Features](#)
 - [Installation](#)
 - [Arduino IDE Library Manager](#)
 - [Installing Manually from GitHub](#)
 - [Usage](#)
 - [Basic Usage](#)
 - [New Methods](#)
 - [Wiring Details](#)
 - [Examples](#)

設定開發版連上sensor

get sensor的data



```
#include <DHT11.h>
```

```
DHT11 dht11(2);
```

```

void setup() {

    Serial.begin(115200);
    dht11.setDelay(500);
}

void loop() {
    int temperature = 0;
    int humidity = 0;
    int result = dht11.readTemperatureHumidity(temperature

    if (result == 0) {
        Serial.print(temperature);
        Serial.print(" ");
        Serial.print(humidity);
        Serial.println();
    } else {
        Serial.println(DHT11::getErrorString(result));
    }
}

```

- ESP32傳輸溫濕度（在Arduino IDE中SerialPort觀測量測到之數據）

將資料傳至server

遇到困難→wifi顯示"connection refused"

解決方法→修改serverAddress對外端

```

#include <ArduinoJson.h>
#include <DHT11.h>
#include <HTTPClient.h>
#include <WiFi.h>

const char *ssid = "我的網路名稱";
const char *password = "秘密";
const char *serverAddress = "http://140.120.233.157:5000";

DHT11 dht11(2);

```

```

void setup() {
    Serial.begin(115200);
    WiFi.begin(ssid, password);

    while (WiFi.status() != WL_CONNECTED) {
        delay(1000);
        Serial.println("Connecting to WiFi...");
    }

    Serial.println("Connected to WiFi");
    Serial.println(WiFi.localIP());
}

void loop() {
    ///////////////////////////////Collect data /////////////////////
    int temperature = 0;
    int humidity = 0;

    // Attempt to read the temperature and humidity value
    int result = dht11.readTemperatureHumidity(temperature);

    if (result == 0) {
        Serial.print("Temperature: ");
        Serial.print(temperature);
        Serial.print(" °C\tHumidity: ");
        Serial.print(humidity);
        Serial.println(" %");
    } else {
        // Print error message based on the error code.
        Serial.println(DHT11::getErrorString(result));
    }

    ///////////////////////////////Send to server ///////////////////
    // Create an object JSON
    DynamicJsonDocument jsonDoc(200);
    jsonDoc["temperature"] = temperature;
}

```

```
jsonDoc["humidity"] = humidity;

String payload;
serializeJson(jsonDoc, payload);

HTTPClient http;
Serial.println("Server Address: " + String(serverAddress));

http.begin(serverAddress);
http.addHeader("Content-Type", "application/json");
int httpResponseCode = http.POST(payload);

if (httpResponseCode > 0) {
    Serial.printf("HTTP Response code: %d\n", httpResponseCode);
    String response = http.getString();
    Serial.println(response);
} else {
    Serial.printf("HTTP Request failed: %s\n",
                  http.errorToString(httpResponseCode));
}

http.end();

delay(5000);
Serial.println();
}
```

問chatGPT哪裡可能有問題

SI You
這個程式執行後為什麼會顯示"connection refused"?
要改哪邊?

```
#include <ArduinoJson.h>
#include <DHT11.h>
#include <HTTPClient.h>
#include <WiFi.h>

const char *ssid = "4110056030";
const char *password = "4110056030";
const char *serverAddress = "http://127.0.0.1:5000/post_data";

DHT11 dht11(2);

void setup() {
    Serial.begin(115200);
    WiFi.begin(ssid, password);

    while (WiFi.status() != WL_CONNECTED) {
        delay(1000);
        Serial.println("Connecting to WiFi...");
    }
}
```

Message ChatGPT

ChatGPT can make mistakes. Consider checking important information.

GPT的回答 → 成功了

ChatGPT
在你的程式中，`serverAddress` 變數設置為 `http://127.0.0.1:5000/post_data`。`127.0.0.1` 是本地主機地址，而 `5000` 是預設的 Flask 開發伺服器端口號。當你嘗試連接到 `http://127.0.0.1:5000` 時，你實際上是試圖連接到你的本地計算機，而不是一個在互聯網上運行的伺服器。
如果你想要發送數據到一個在互聯網上運行的伺服器，你應該將 `serverAddress` 設置為該伺服器的實際 IP 地址或主機名稱，以及對應的端口號。
例如，如果你有一個在互聯網上運行的 Flask 伺服器，並且它的 IP 地址是 `203.0.113.10`，端口號是 `8080`，那麼你的 `serverAddress` 應該被設置為 `http://203.0.113.10:8080/post_data`。請確保你的目標伺服器是運行並且可以接受來自你的裝置的請求。

Message ChatGPT

ChatGPT can make mistakes. Consider checking important information.

▼ 實作Code(修改自https://github.com/sc0210/3311_Lab6_student)

- 使用Flask製作後端API設計 (Backend)

app.py

```
import json
import time

from flask import (
    Flask,
    Response,
    jsonify,
    render_template,
    request,
    stream_with_context,
)

from db_method import DB_NAME, get_all_data, get_single

app = Flask(__name__, template_folder="templates")

data = {"temperature": [], "humidity": []}

@app.route("/")
def index():
    return render_template("index.html")

@app.get("/visualize-realtime")
def visualize():
    return render_template("realtime-data.html")

@app.get("/visualize-specifiedtime")
def visualize2():
    return render_template("specifiedtime-data.html")

@app.get("/visualize-camera_view")
def visualize3():
    return render_template("camera_view.html")

@app.post("/post_data")
```

```

def receive_data():
    try:
        content = request.get_json()

        temperature = content["temperature"]
        humidity = content["humidity"]

        data["temperature"].append(temperature)
        data["humidity"].append(humidity)

        # Call function to initialize database if not already done
        # initdb()

        # Call function to insert sensor data into the database
        insert_sensor_data(temperature, humidity)

        print(f"Received data: temperature={temperature}, humidity={humidity}")
        return jsonify({"success": True})

    except Exception as e:
        print(f"Error receiving data: {str(e)}")
        return jsonify({"success": False, "error": str(e)})
    
```



```

@app.get("/get_data")
def get_data():
    try:
        # Extract start_time and end_time parameters from URL
        start_time_str = request.args.get("start_time")
        end_time_str = request.args.get("end_time")

        if not start_time_str or not end_time_str:
            return (
                jsonify({"error": "start_time and end_time parameters are required"}),
                400,
            )

        # Call the function to retrieve temperature data
        # ...
    
```

```

        data_result = get_single_data(start_time_str, end_time_str)

        if data_result is None:
            return jsonify({"error": "Failed to retrieve data"})

        # Convert data to a list of dictionaries for JSON
        formatted_data = [
            {"temperature": row[0], "humidity": row[1],
             "date": row[2]} for row in data_result
        ]

        return jsonify(formatted_data)
    except Exception as e:
        return jsonify({"error": str(e)}), 500

@app.get("/get_all_data")
def get_all_data_route():
    try:
        # Call the function to retrieve all temperature
        all_data = get_all_data(DB_NAME)

        if all_data is None:
            return jsonify({"error": "Failed to retrieve data"})

        # Convert data to a list of dictionaries for JSON
        formatted_data = [
            {"temperature": row[0], "humidity": row[1],
             "date": row[2]} for row in all_data
        ]

        return jsonify(formatted_data)
    except Exception as e:
        return jsonify({"error": str(e)}), 500

class camera_view:

```

```

def __init__(self):
    self.a_camera_frame = {"image": "NC", "timestamp": 0}

def update_camera_frame(self, frame):
    self.a_camera_frame = frame

def get_camera_frame(self):
    return self.a_camera_frame

a_camera_view = camera_view()

@app.post("/post_camera_frame")
def receive_camera_frame():
    try:
        content = request.get_json()
        a_camera_view.update_camera_frame(content)
        return "camera_frame updated successfully"
    except Exception as e:
        return jsonify({"success": False, "error": str(e)})

@app.route("/get_camera_stream")
def make_stream():
    # 與相機網頁前端建立event-stream
    @stream_with_context
    def generate():
        try:
            while True:
                yield "data:" + json.dumps(a_camera_view.a_camera_frame)
                time.sleep(0.1)
        except GeneratorExit:
            print("closed")

    # 用stream發給前端
    return Response(generate(), mimetype="text/event-stream")

```

```
if __name__ == "__main__":
    initdb(DB_NAME)
    app.run(host="0.0.0.0", port=5000, debug=True)
```

- 使用Flask製作前端頁面 (Frontend&UI)

顯示在前端

index.html

```
<!DOCTYPE html>
<html lang="en">

<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width,
    <title>Sensor Data Collection</title>
    <link rel="stylesheet" href="/static/style.css">
</head>

<body>
    <!-- header.html -->
    {% include 'header.html' %}

    <div class="container">
        <h1>Welcome</h1>
        <h2>~AIOT HW~</h2>
        <p>Monitor temperature and humidity with our se
        <div>
            <a href="/visualize-realtime" class="button"
                style="color: #2ddcff0">real-time<
            <a href="/visualize-specifiedtime" class="b
                style="color: #ffa735">specific-t
        </div>

    </div>

    <!-- footer.html -->
    {% include 'footer.html' %}
```

```

</body>

</html>

```

```

File Edit Selection View Go Run Terminal Help ⌘ my
EXPLORER MY app.py style.css index.html header.html
app.py > get_data
    1 import json
    2 import time
    3
    4 from flask import (
    5     Flask,
    6     Response,
    7     jsonify,
    8     render_template,
    9     request,
   10     stream_with_context,
   11 )
   12
   13 from db_method import DB_NAME, get_all_data, get_single_data, initdb, insert_sensor_data
   14
   15 app = Flask(__name__, template_folder="templates")
   16
   17 data = {"temperature": [], "humidity": []}
   18
   19 # -----前端頁面-----
   20 # 主頁
   21
   22
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
* Serving Flask app 'app'
* Debug mode: on
WARNING: This is a development server. Do not use it in a production deployment. Use a production WSGI server instead.
* Running on all addresses (0.0.0.0)
* Running on http://127.0.0.1:5000
* Running on http://140.120.233.157:5000
Press CTRL+C to quit
* Restarting with watchdog (windowlapi)
(initdb) database initialized successfully.
* Debugger is active
* Dockerfile PIN: 989-066-129
127.0.0.1 - [08/May/2024 11:19:27] "GET / HTTP/1.1" 200 -
127.0.0.1 - [08/May/2024 11:19:27] "GET /static/style.css HTTP/1.1" 304 -

```

header.html

```

<! -- header.html -->
<div class="header-container">

    <header>
        <nav>
            <div class="logo">
                <a href="https://github.com/sc0210/3311">
                    
                </a>
                <a href="https://lms2020.nchu.edu.tw/index.php">
                    
                </a>
            </div>
            <ul class="navigation">
                <li><a href="/">Home</a></li>
                <li><a href="/visualize-realtime">Lab6</a></li>
            </ul>
        </nav>
    </header>
</div>

```

```

<li><a href="/visualize-camera_view">Lab7</a>
<li><a href="http://awin.cs.nchu.edu.tw">About</a>
<li><a href="https://www.cwa.gov.tw/V8/">Central Weather</a>
<li><a href="https://github.com/dhrubashah08/DHT11">DHT11</a>
    <!-- Add more navigation links as needed -->
</ul>
</nav>
</header>

</div>

```

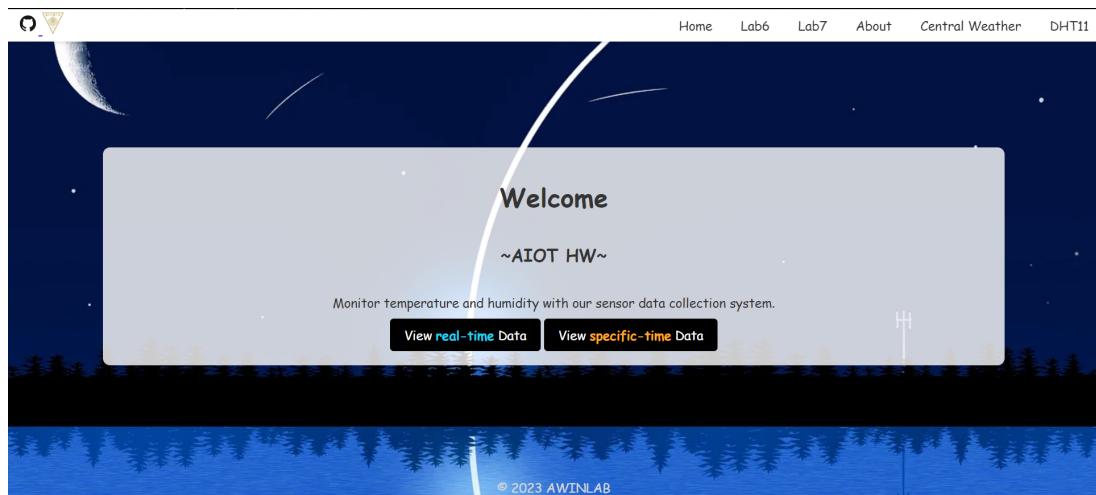
增加navigation至中央氣象局與檢視DHT11

```

<File Edit Selection View Go Run Terminal Help <- > my > my>
EXPLORER templates > header.html > index.html > header.html
MY app.py style.css
> _pycache_
> ArduinoSampleCode
> figures
> static
templates > footer.html
> header.html
> index.html
> realtime-data.html
> specifiedtime-data.html
> .gitignore
app.log
app.py
db_method.py
README.md
SetupArduinoIDE.sh
temperature_data.sqlite

<div class="header">
    <div class="logo">
        <a href="https://github.com/sc0210/3311_Lab6-student.git">
            
        </a>
        <a href="https://lmss2020.nchu.edu.tw/index/login?next=%2Fdashboard">
            
        </a>
    </div>
    <ul class="navigation">
        <li><a href="#">Home</a></li>
        <li><a href="/visualize-realtime">Lab6</a></li>
        <li><a href="/visualize-camera_view">Lab7</a></li>
        <li><a href="http://awin.cs.nchu.edu.tw">About</a></li>
        <li><a href="https://www.cwa.gov.tw/V8/">Central Weather</a></li>
        <li><a href="https://github.com/dhrubashah08/DHT11">DHT11</a></li>
        <!-- Add more navigation links as needed -->
    </ul>
</div>
</header>
</div>

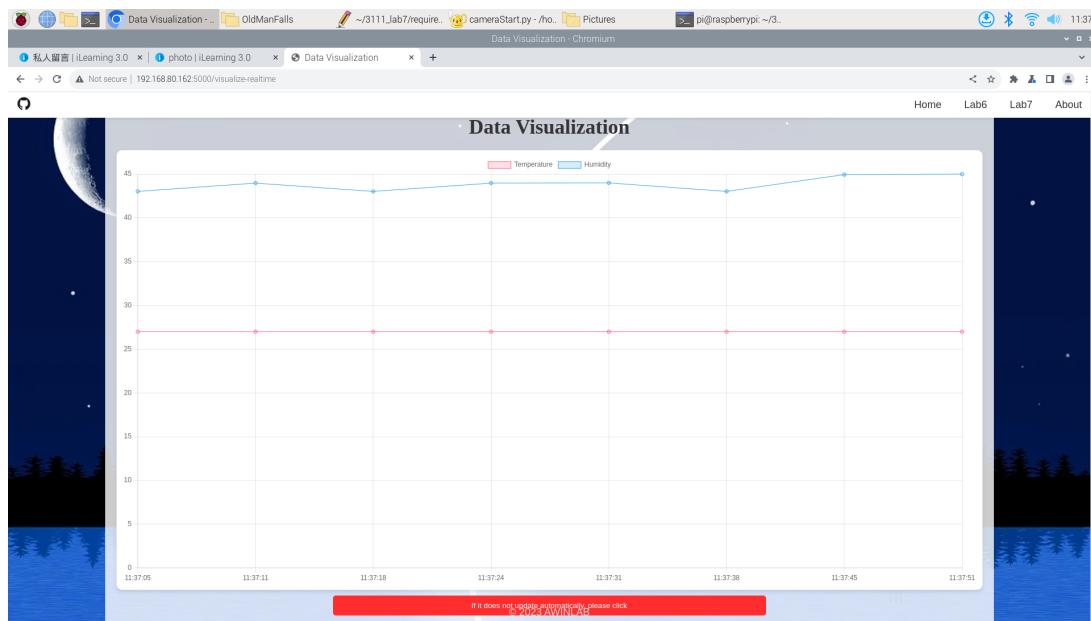
```



((此前端和影片gif影片不同時間錄製因此稍有不同

- 呈現即時數據表現（每五秒更新一次數據）

樹莓派上運行(課堂上的螢幕)



- 可觀看過去某個時間區段資料

