

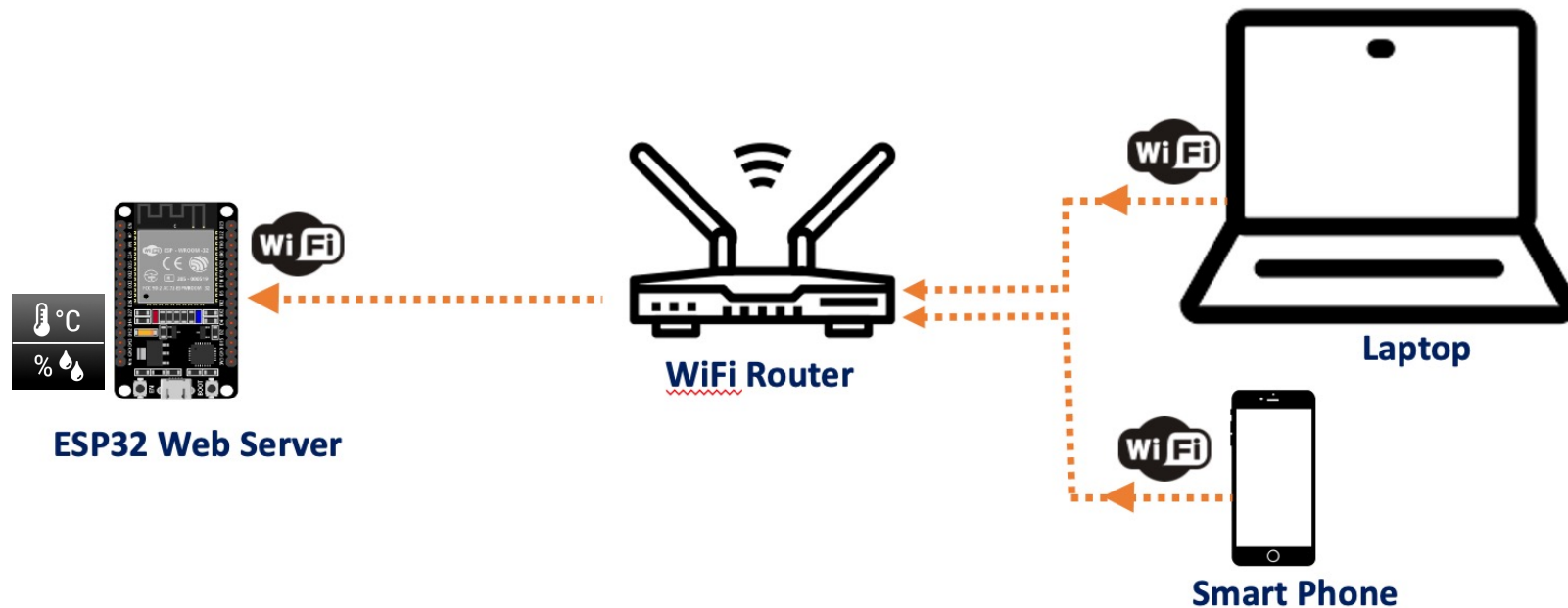
MODULE 2c

Monitoring Data Center Temperature & Humidity Level

safyzan salim
019 622 0575

Project Overview

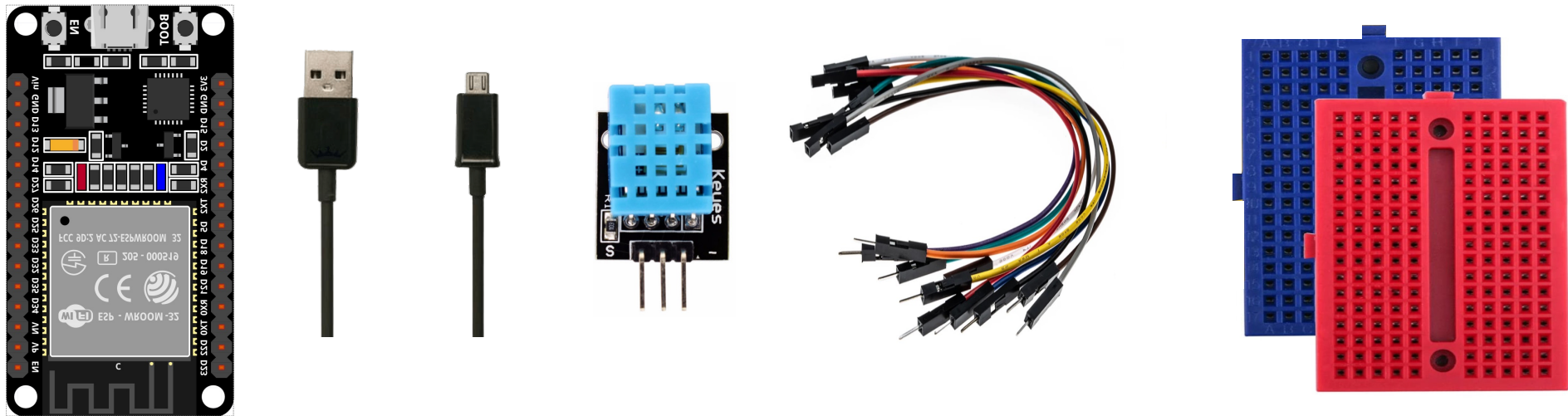
This tutorial will teach you how to build a mobile monitoring system for data center that captures its temperature & humidity level by using DHT11 & ESP32. It is a combination of previous 2 exercises that we have went through; i.e., *ii-esp32+dht.pptx* & *iii-esp32+sta.pptx*.



2c. Monitoring Data Center Temperature & Humidity Level

Components

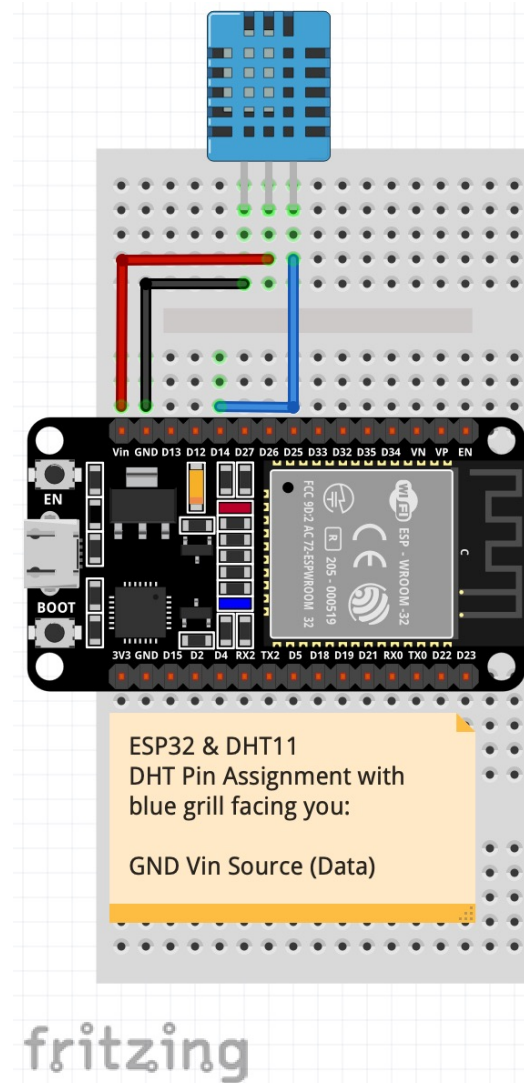
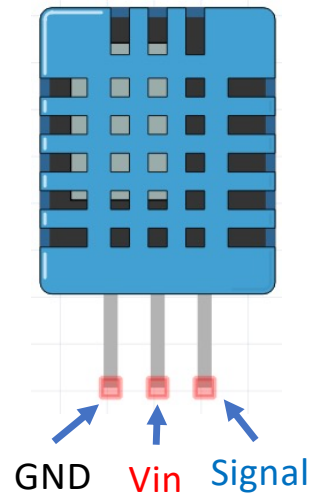
- 1 x ESP-32 Wifi+Bluetooth 2-In-1 Development Board for Arduino (30 pin) + Cable.
- 1 x Temperature & Humidity Sensor, DHT11.
- Jumpers (male to female).
- 2 x Mini Breadboard.




2c. Monitoring Data Center Temperature & Humidity Level

Schematic Diagram

→ Check a few times the connections that have been made before plugging ESP32 to USB port. This might prevent short circuit to DHT11. Watch out on sensors' pin. Same type, doesn't mean same pin configurations.



Working File

- Download the file working file from <http://bit.ly/3aB3WBL> by click at .
- Unzip and find ***sketch-iv_esp32-dht11-sta.ino***. Double click to open the file with Arduino IDE.
- Click **OK** when Arduino pop-up window appears. This action will create a folder that carry same name with sketch (***sketch-iv_esp32-dht11-sta.ino***) and move the sketch into it. Click **Cancel** will not open the sketch at all.
- Before upload the sketch to the board, modify the following two variables and suit with your network credentials. This will allow ESP32 establish connection with your network.

```
20 // enter your network credentials
21 const char* ssid = "air24";
22 const char* password = "polis12345";
```

- Upload the sketch to your ESP32 board. Ensure that you have selected the right board and set the current port. Don't forget to press **BOOT** button. Refer to **Troubleshoot** section if you have problem.

2c. Monitoring Data Center Temperature & Humidity Level

The Output

→ Open a browser and type the ESP32 IP address, **192.168.1.114**

The image shows two overlapping windows. The background window is a terminal window displaying the output of an ESP32 program. The foreground window is a web browser showing the web page of the ESP32.

Terminal Output:

```
entry 0x400806b4  
Connecting to: air24  
.  
WiFi connected successfully  
Your ESP32 IP: 192.168.1.114  
MDNS responder started  
HTTP server started  
Mode: STA  
Channel: 13  
SSID (5): air24  
Passphrase (10): polis12345  
BSSID set: 0  
RSSI:-69
```

Web Page Content:

Server Up Time: 0h:53m:46s

My Portable Data Center

Temperature & Humidity Monitoring System

ID: dc#1a

Temperature: 32 °C

Humidity: 59%

At the bottom of the terminal window, there are controls: ☒ Autoscroll, ☐ Show timestamp, No line ending, 115200 baud, and Clear output.


Red arrows point from the text "IP address released by air24" to the IP address "192.168.1.114" in the terminal output and the browser address bar.

iii. Create a Web Server with ESP32 and Arduino IDE

Code Explanation

- Filename: ***sketch-iv_esp32-dht11-sta.ino***
- Additional library needed was ***dht11.h***.
- ***sketch-iv_esp32-dht11-sta.ino*** is a combination of previous 2 exercises; ***sketch-ii & sketch-iii***.
- First, add dht11 initialization, connect **SIGNAL** pin of the sensor to **D4** at ESP32.

```
8 #include <WiFi.h>
9 #include <WebServer.h>
10 #include <ESPmDNS.h>
11
12 #include <dht11.h>
13 dht11 DHT;
14 #define DHT11_PIN 4
```

A yellow box labeled "dht11 initialization" has three red arrows pointing to the code lines: line 12 (#include <dht11.h>), line 13 (dht11 DHT;), and line 14 (#define DHT11_PIN 4).

Code Explanation

- Write Arduino syntax such as, read sensor value, turn on LED, control servo motor etc. within SendHTML() function. This function is invoke by loop () function. It's a non-stop process. Thus the reading will keep updated depends on time interval set.
- The DHT.read(4) tasks is to gather the DHTLIB health status, the current temperature & also the humidity. You may open dht11.cpp for better understanding.

```
105 String SendHTML( ){  
106     // returns the number of milliseconds that  
107     // your ESP32 board since powered up,  
108     // & become zero when reset or unplugged  
109     int sec = millis() / 1000;  
110     int min = sec / 60;  
111     int hr = min / 60;  
112     int chk = DHT.read(DHT11_PIN); // READ DATA
```

Call dht11.cpp function

Code Explanation

→ Remember, writing a program a top down-process. You have to declare the variables first before using it, get the data, than write the formula for process, finally, display the results. It is similar as cooking. You cannot skip the order.

```
124 ptr += "<div id=\"webpage\">\n";
125 ptr += "Server Up Time: <b>";
126 ptr += hr;
127 ptr += "h:";
128 ptr += min % 60;
129 ptr += "m:";
130 ptr += sec % 60;
131 ptr += "s </b>\n";
132 ptr += "<h1>My Portable Data Center</h1>\n";
133 ptr += "<h1>Temperature & Humidity Monitoring System</h1>\n";
134 ptr += "<h2>ID: dc#1a</h2>\n"; ← Sensor ID
135 ptr += "<p>Temperature: ";
136 ptr += DHT.temperature,1; ← Display temperature in decimal at web page
137 ptr += " ";
138 ptr += char(176); ← Degree symbol
139 ptr += "C</p>";
140 ptr += "<p>Humidity: ";
141 ptr += DHT.humidity,1; ← Display humidity in decimal at web page
142 ptr += "%</p>";
143 ptr += "</div>\n";
```

QUESTIONS?



Questions?

EXERCISE

Add a RED led and GREEN led. When the temperature less than 27°C, only GREEN led goes off. When the temperature greater than 27°C, only RED led goes off.

Hint: -use conditional statement:
 -digital write
[30 minutes activity]

2c. Monitoring Data Center Temperature & Humidity Level

ANSWER

Don't forget to initialize pinMode();

```
if (DHT.temperature>27)
{
    warn = "danger";
    //Serial.println(warn);
    digitalWrite(pinGreen,LOW);
    digitalWrite(pinRed,HIGH);
} else
{
    warn = "normal";
    //Serial.println(warn);
    digitalWrite(pinGreen,HIGH);
    digitalWrite(pinRed,LOW);
}
```

TROUBLESHOOT GUIDE

Failed to upload the sketch to ESP32? Follow these steps:

- Press REBOOT button until the IDE approaches **Connecting...** segment;
- **OR**, check whether you have chosen the correct **Board** and set the correct **PORT**;
 - * Choose the right board by going to **Tools > Boards > ESP32 Dev Module**
 - * Select the correct port at **Tools > Port > choose the appropriate serial port**
- **OR**, unplugging the board from USB and plugging back;
- **OR**, **Verify** to confirm that your sketch is error free;
- **OR**, Swap to other PC / Laptops;
- **OR**, Use different board;
- You are using micro-USB power cable instead of power & data cable.