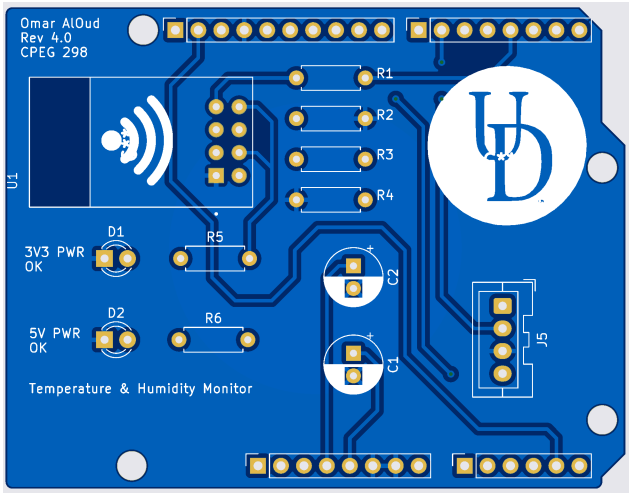
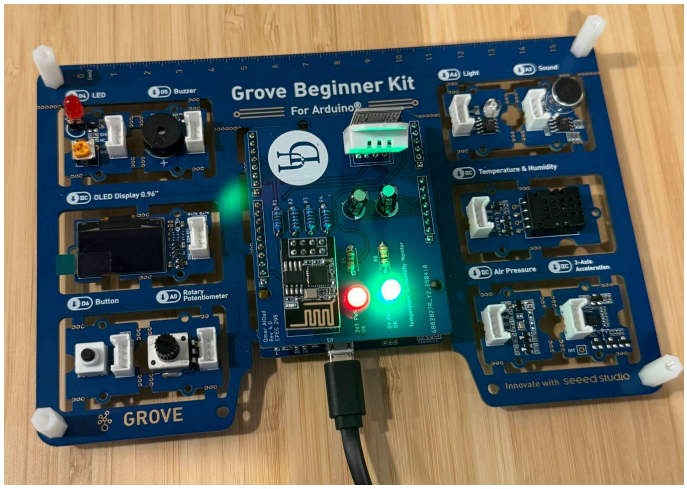
Omar AlOud
CPEG298: Final Design Report
16 May 2025

IoT-Enabled Temperature & Humidity Monitoring Device

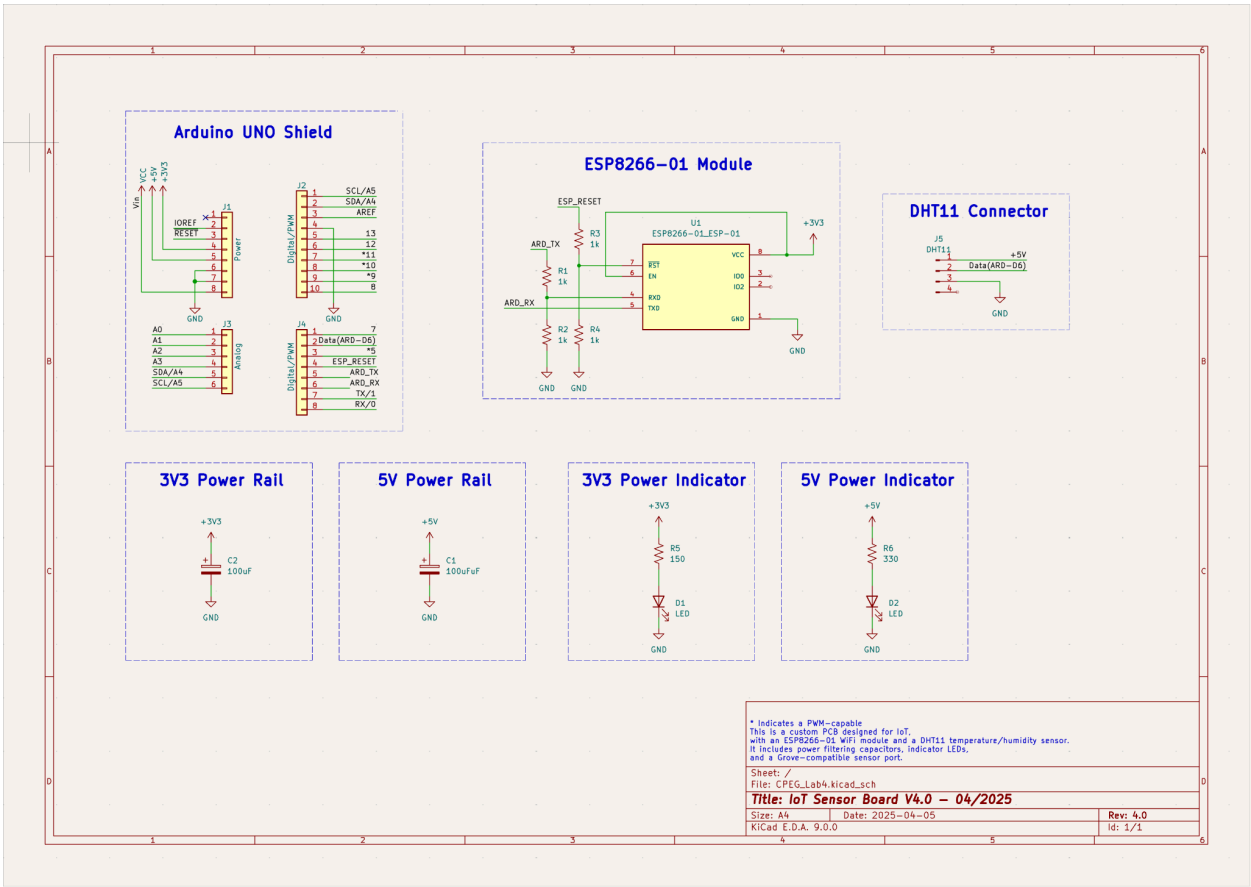




1. Circuit Functionality

The custom-designed IoT PCB is built to measure and display ambient temperature and humidity using a DHT22 sensor. The data is displayed in real time by sending it over Wi-Fi using an ESP-01 (ESP8266) module to the Adafruit IO cloud platform. The Arduino acts as the main microcontroller. The PCB interfaces with the Arduino via pin headers, drawing power and signal connections. LEDs indicate power and activity status.

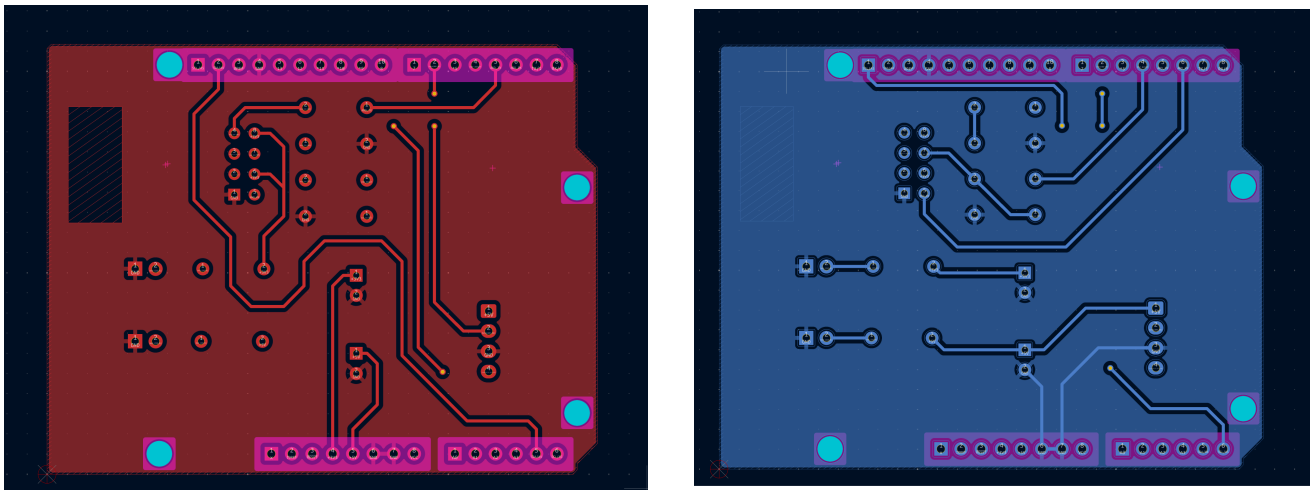
2. Schematics



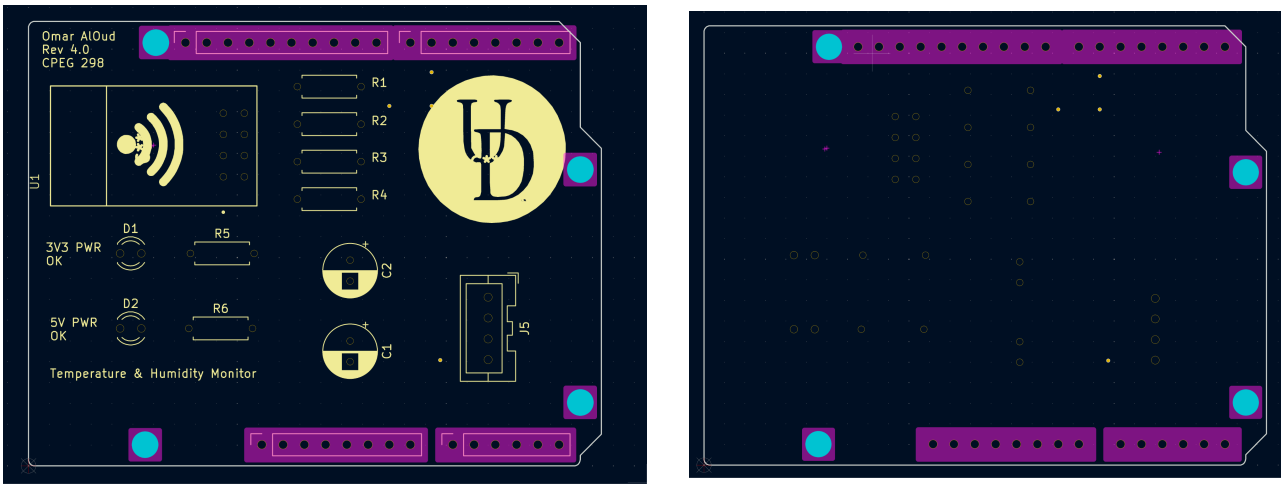
The above schematic includes:

- DHT22 temperature and humidity sensor.
- ESP-01 Wi-Fi module (connected via TX/RX and reset pin).
- 2x LEDs with series resistors for power/activity indication.
- Capacitors for voltage stabilization.

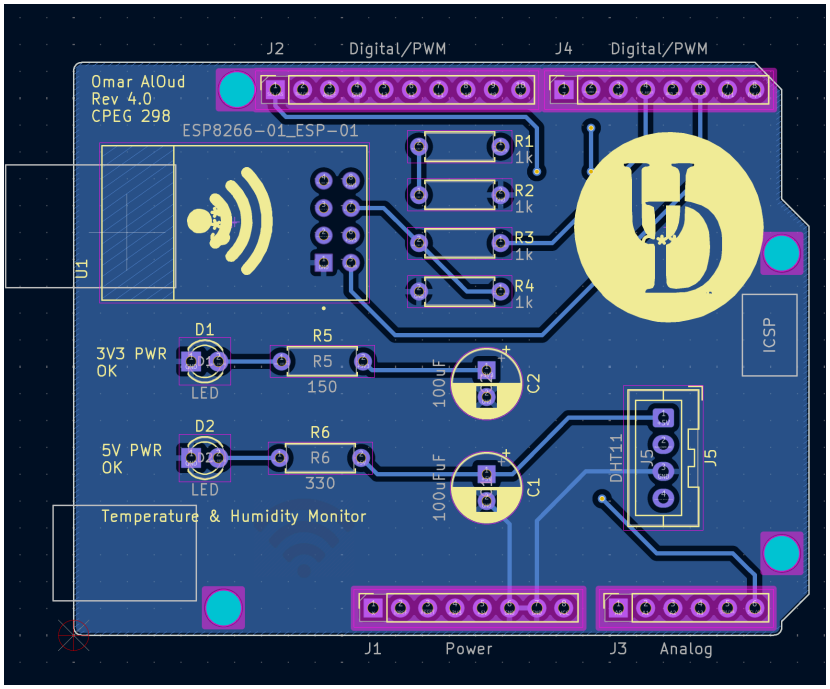
3. Manufacturing (CAM) Images



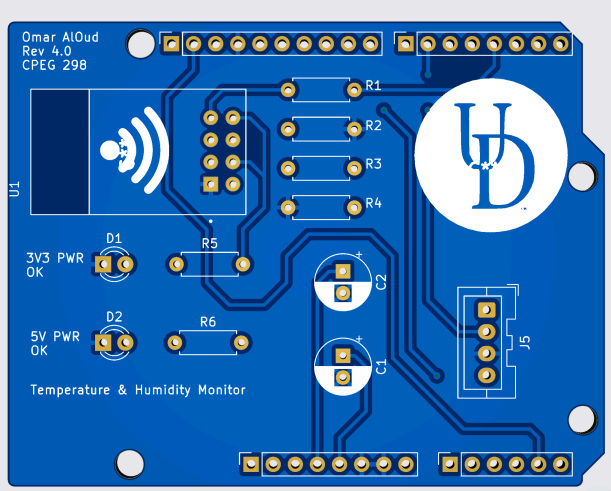
Top Copper Layer (left) & Bottom Copper Layer (Right).

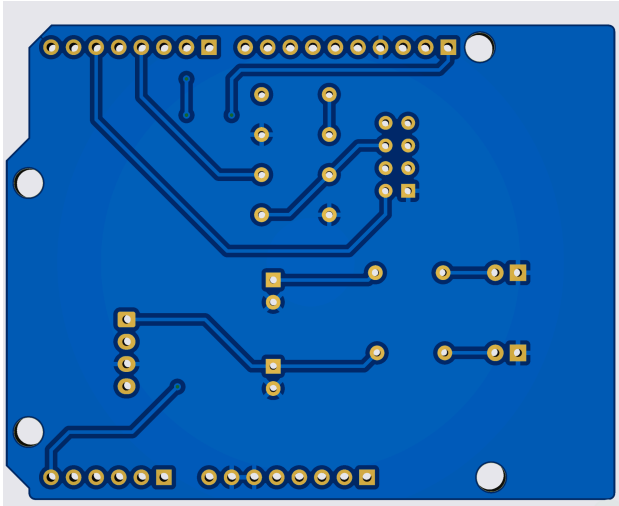


Front Silkscreen (Left), Back Silkscreen (Right).



Full PCB Image





3D image of PCB (front and back).

---

4. Bill of Materials

Component	Value	Footprint	Link	Qty	Cost
J1	Power	PinSocket_1x08_P2.54mm_Vertical	https://www.digikey.com/en/products/detail/metz-connect-usa-inc/PR20208VBNN/13402440	1	0.12
J2	Digital/PWM	PinSocket_1x10_P2.54mm_Vertical	https://www.digikey.com/en/products/detail/sullins-connector-solutions/PPPC101LFBN-RC/810182	1	0.64
J3	Analog	PinSocket_1x06_P2.54mm_Vertical	https://www.digikey.com/en/products/detail/metz-connect-usa-inc/PR20208VBNN/13402440	1	0.12
J4	Digital/PWM	PinSocket_1x10_P2.54mm_Vertical	https://www.digikey.com/en/products/detail/sullins-connector-solutions/PPPC101LFBN-RC/810182	1	0.64
D1, D2	LEDs (R&G)	LED_D3.0mm_FlatTop	https://www.digikey.com/en/products/detail/marktech-optoelectronics/MT7403A-UR-A/4214630	2	0.82
R1-R4	1kΩ	R_Axial_DIN0207_L6.3mm_D2.5mm_P10.16mm_Horizontal	https://www.digikey.com/en/products/detail/vishay-beyschlag-draloric-bc-components/MBB0207VD1001BC100/7350409	4	6.8
R5	150Ω	R_Axial_DIN0207_L6.3mm_D2.5mm_P10.16mm_Horizontal	https://www.digikey.com/en/products/detail/vishay-beyschlag-draloric-bc-components/SFR2500001500FR500/595677	1	0.14

Component	Value	Footprint	Link	Qty	Cost
R6	330Ω	R_Axial_DIN0207_L6.3mm_D2.5mm_P10.16mm_Horizontal	https://www.digikey.com/en/products/detail/yageo/CFR-25JR-52-330R/11962	1	0.10
C1, C2	100µF	CP_Radial_D6.3mm_P2.50mm	https://www.digikey.com/en/products/detail/rubycon/35YXJ100M6-3X11/3563127	2	0.58
J5	DHT22	JST_XA_B04B (4-pin vertical)	https://www.adafruit.com/product/385	1	9.95
U1	ESP8266-01 WiFi Module	XCVR_ESP8266-01_ESP-01	https://www.microcenter.com/product/616024/ESP8266_WiFi_Transceiver_Receiver_Module_-_2_Pack?gQT=2	1	5.99

Total estimated cost = $25.9

5. Arduino Code & Libraries

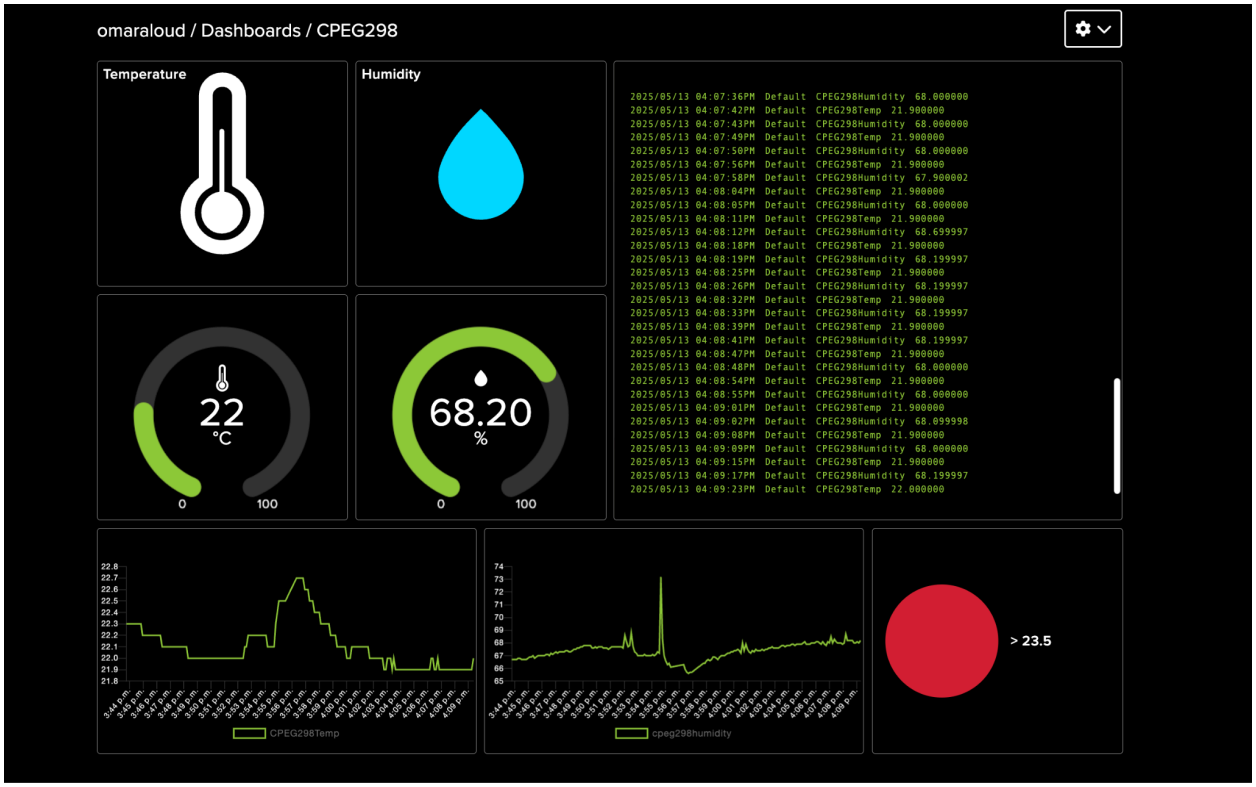
Arduino code:

- Reads temperature and humidity from the DHT22 sensor.
- Sends data to Adafruit IO using SoftwareSerial to communicate with the ESP-01.

Libraries used:

- DHT.h
- SoftwareSerial.h

6. Adafruit IO Dashboard

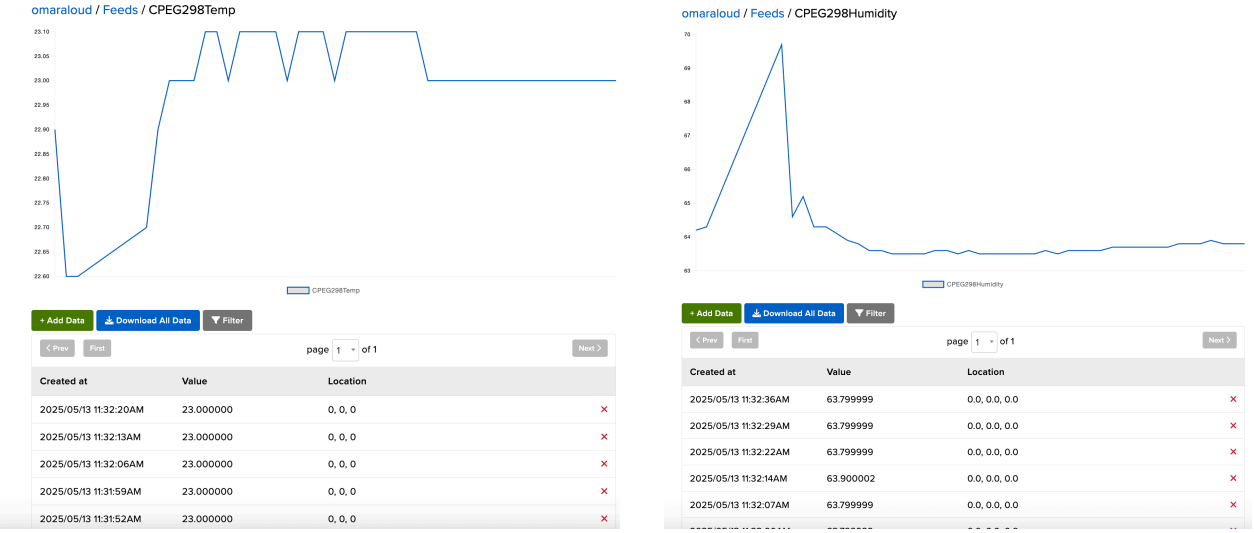


This custom Adafruit IO dashboard provides real-time remote monitoring of temperature and humidity using the DHT22 sensor and ESP8266 module.

Widgets:

- Two gauge indicators show the live temperature (°C) and humidity (%) values.
- Icons for quick visual identification of each measurement.
- 2 graphs plot changes in temperature and humidity over time.
- A data log panel displays sensor data with timestamps.
- A status indicator shows whether the temperature exceeds 23.5°C (red = false).

Data is sent from the device to Adafruit IO every 5 seconds via the ESP-01 module over Wi-Fi, this allows remote viewing from any device connected to the internet.



Temperature Feed (Left) and Humidity feed (Right).

---

7. Improvements

If I was to redesign my device I would explore the following ideas:

- Header for LiPo battery connection, making the device more portable.
- Design/3D print an enclosure.
- Label silkscreen more clearly as I do not have labels for J1-J4.
- Think about future expansion, extra GIPO pins etc.

Appendix A: Arduino Code

```
#include "Arduino.h"
#include <SoftwareSerial.h>    //  use two GPIO pins for a second UART
#include "DHT.h"               // Include DHT library

#define RESET_PIN 4            // pin used to bring ESP-01 module up in UART mode
#define UART_TX_PIN 3         // UART software serial transmit pin
#define UART_RX_PIN 2         // UART software serial receive pin
#define DHTPIN 6               // DHT22 data pin connected to D6
#define DHTTYPE DHT22          // Using DHT22 sensor

DHT dht(DHTPIN, DHTTYPE);      // Create DHT object
SoftwareSerial espSerial(UART_RX_PIN, UART_TX_PIN);  // Create software UART to talk
to the ESP8266

//  credentials
String IO_USERNAME = "omaraloud";
String IO_KEY = "aio_KswB755xCpPJbU0fPD4ZCSmKTjZv";
String WIFI_SSID = "UD Devices";
String WIFI_PASS = "";

// Internal state
String response;
String subresp;
long int time, updatetime;

void setup() {
 Serial.begin(115200);
 espSerial.begin(9600);

 pinMode(RESET_PIN, OUTPUT);
 digitalWrite(RESET_PIN, LOW);
 delay(1000);
 digitalWrite(RESET_PIN, HIGH);

 dht.begin(); // Start DHT sensor

 response = "";
```

```
 subresp = "";
 time = millis();

 // Clear UART buffer
 while ((time + 2000) > millis()) {
   while (espSerial.available()) {
     char c = espSerial.read();
     response += c;
     Serial.print(c);
   }
 }

 espSerial.flush();
 Serial.println("\n\nCPEG/ELEG Demo S25");
 Serial.println("setting up Adafruit IO");

 response = espSend("get_macaddr", 2000, true);
 response = espSend("get_version", 2000, true);
 response = espSend("wifi_ssid=" + WIFI_SSID, 2000, true);
 response = espSend("wifi_pass=" + WIFI_PASS, 2000, true);
 response = espSend("io_user=" + IO_USERNAME, 2000, true);
 response = espSend("io_key=" + IO_KEY, 2000, false);
 response = espSend("setup_io", 30000, true);

 if (response.indexOf("connected") < 0) {
   Serial.println("\nAdafruit IO Connection Failed");
   while (1);
 }

 response = espSend("setup_pubfeed=1,CPEG298Temp", 2000, false);     // Temperature
 response = espSend("setup_pubfeed=2,CPEG298Humidity", 2000, false);  // Humidity

 Serial.println("------ Setup Complete ----------");
}

void loop() {
 float temp = dht.readTemperature(); // Celsius
 float hum = dht.readHumidity();

 if (isnan(temp) || isnan(hum)) {
   Serial.println("Failed to read from DHT sensor!");
```

```
    return;
  }

  Serial.print("Temp: ");
  Serial.print(temp);
  Serial.print("°C, Humidity: ");
  Serial.print(hum);
  Serial.println("%");

  response = espSend("send_data=1," + String(temp), 2000, false); // Temp to feed 1
  response = espSend("send_data=2," + String(hum), 2000, false);  // Humidity to feed 2

  delay(5000);
}

String espSend(String command, const int timeout, boolean debug) {
  response = "";
  char c;
  espSerial.println(command);
  long int time = millis();

  while ((time + timeout) > millis()) {
    while (espSerial.available()) {
      c = espSerial.read();
      response += c;
    }
    if (c == '\n') break;
  }

  response.trim();
  if (debug) {
    Serial.println("Resp: " + response);
  }

  return response;
}
```