



PES UNIVERSITY

(Established under Karnataka Act No. 16 of 2013)

100 Feet Ring Road, BSK 3rd Stage, Bengaluru-560085

Department of Electronics and Communication Engineering

NEXTGEN COMMUNICATION SYSTEMS

(Summer Skill Enhancement Course)

Project Topic

“Sensor data communication between systems using ESP8266”

Under the Guidance of

Prof. Prajeesha

(Dept of ECE)

Submitted by

Deepa Kulkarni (PES1UG22EC908) - 4 'C'

Divyansh Sharma (PES1UG21EC093) - 6 'B'

Aim:

To establish connection between 2 ESP8266 systems:client and server; share & display sensor data values over Wi-Fi connection.

Components required:

- 2 ESP8266 nodemcu boards
- DHT11 sensor
- Jumper wires and breadboard
- Micro USB Data Cables
- LCDi2C display

Code:

Server Code:

```
#include <ESP8266WiFi.h>

#include "DHT.h"

const char* ssid = "xyz";

const char* password = "abcd1234";

WiFiServer server(80);

#define DHTPIN 4

#define DHTTYPE DHT11

DHT dht(DHTPIN, DHTTYPE);

void setup() {

    Serial.begin(9600);

    delay(10);

    dht.begin();

    // Connect to Wi-Fi

    WiFi.begin(ssid, password);

    while (WiFi.status() != WL_CONNECTED) {
```

```
    delay(1000);

    Serial.println("Connecting to WiFi...");

}

// Start the server

server.begin();

Serial.println(WiFi.localIP());

Serial.println("Server started");

}

void loop() {

    WiFiClient client = server.available();

    if (client) {

        Serial.println("Client connected");

        while(client) {

            float temperature = dht.readTemperature();

            float humidity = dht.readHumidity();

            if (isnan(temperature) || isnan(humidity)) {

                Serial.println("Failed to read from DHT sensor!");

                client.println("Failed to read from DHT sensor!");

            }

            else {

                String temperatureData = "Temp: " + String(temperature) + " C";

                String humidityData = "Hum: " + String(humidity) + " %";

                Serial.println(temperatureData);

                Serial.println(humidityData);

                client.println(temperatureData);

            }

        }

    }

}
```

```
        client.println(humidityData);

    }

    delay(3000);

}

}
```

Client Code:

```
#include <ESP8266WiFi.h>

#include <Wire.h>

#include <LiquidCrystal_I2C.h>

const char* ssid = "xyz";

const char* password = "abcd1234";

const char* serverIP = "192.168.137.222";

LiquidCrystal_I2C lcd(0x27, 16, 2);

void setup() {

    Serial.begin(9600);

    lcd.init();

    lcd.backlight();

    delay(10);

    // Connect to Wi-Fi

    WiFi.begin(ssid, password);
```

```
while (WiFi.status() != WL_CONNECTED) {

    delay(1000);

    Serial.println("Connecting to WiFi...");

}

Serial.println("Connected to WiFi");

}

void loop() {

    if (WiFi.status() == WL_CONNECTED) {

        WiFiClient client;

        if (client.connect(serverIP, 80)) {

            while (client.connected()) {

                if (client.available()) {

                    String temperature = client.readStringUntil('\n');

                    String humidity = client.readStringUntil('\n');

                    Serial.print("Server temperature: ");

                    Serial.println(temperature);

                    Serial.print("Server humidity: ");

                    Serial.println(humidity);

                    temperature.trim();

                    humidity.trim();

                }

            }

        }

    }

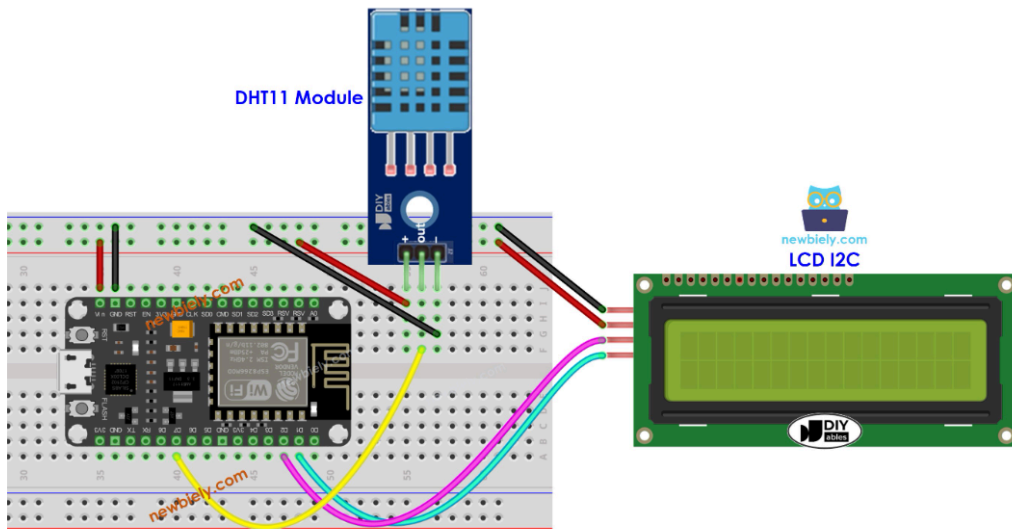
}
```

```
    if (!temperature.isEmpty() && !humidity.isEmpty()) {  
        lcd.clear();  
        lcd.setCursor(0, 0);  
        lcd.print(temperature);  
        lcd.setCursor(0, 1);  
        lcd.print(humidity);  
    }  
}  
}  
}  
}  
}
```

Both the server and the client get connected to the same hotspot with the name 'xyz' and password 'abcd1234'. As soon as the connection is established, the server side circuit starts recording the sensor values and transfers it to the client side via wifi connection, where the client side circuit displays the sensor values in the powershell as well as the LCD Display.

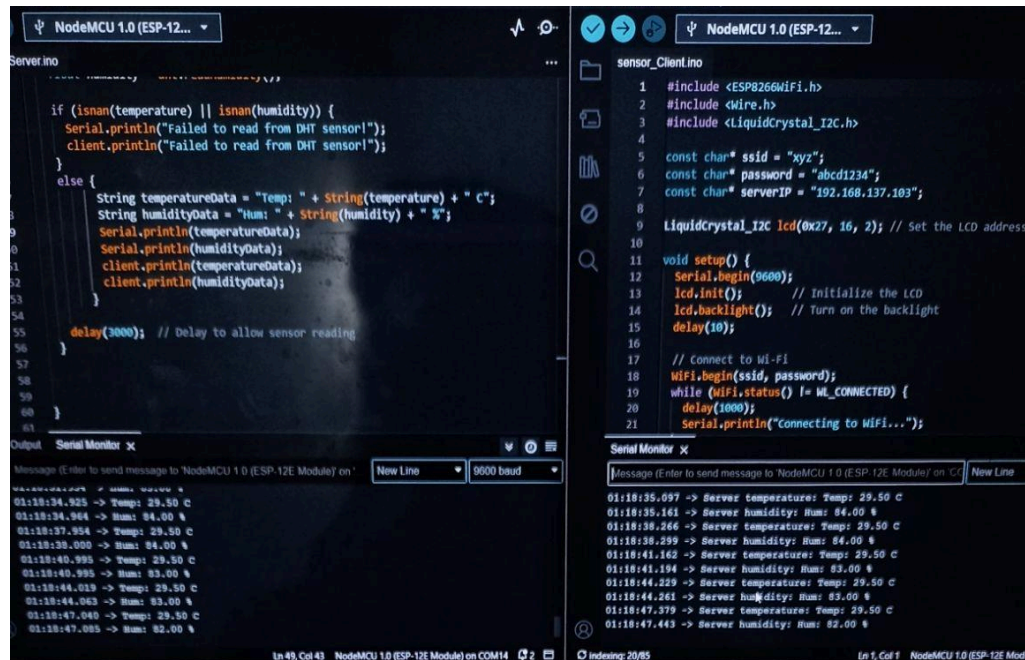
Circuit diagram:

ESP8266 - DHT11 and LCD Wiring Diagram



The above circuit diagram shows a sample connection of the ESP to DHT11 sensor and the LCD display. Similar connection is to be done on 2 ESP boards, one on the server side with the DHT11 module, which is responsible for sending the sensor's data, and one on the client side with the LCD display which will be displaying the values. Please refer to the next (outputs) section for a better understanding of the connections made and outputs received.

Outputs:



```
Server.ino
// ...
if (isnan(temperature) || isnan(humidity)) {
  Serial.println("Failed to read from DHT sensor!");
  client.println("Failed to read from DHT sensor!");
}
else {
  String temperatureData = "Temp: " + String(temperature) + " C";
  String humidityData = "Hum: " + String(humidity) + " %";
  Serial.println(temperatureData);
  Serial.println(humidityData);
  client.println(temperatureData);
  client.println(humidityData);
}
delay(3000); // Delay to allow sensor reading
}

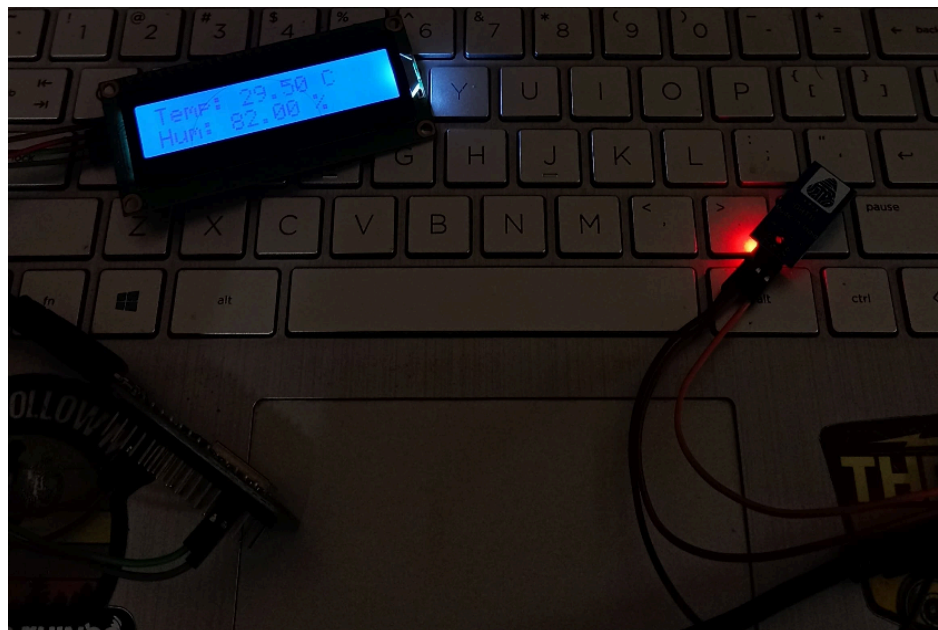
sensor_Client.ino
1 #include <ESP8266WiFi.h>
2 #include <Wire.h>
3 #include <LiquidCrystal_I2C.h>
4
5 const char* ssid = "xyz";
6 const char* password = "abcd1234";
7 const char* serverIP = "192.168.137.103";
8
9 LiquidCrystal_I2C lcd(0x27, 16, 2); // Set the LCD address
10
11 void setup() {
12   Serial.begin(9600);
13   lcd.init(); // Initialize the LCD
14   lcd.backlight(); // Turn on the backlight
15   delay(10);
16
17   // Connect to Wi-Fi
18   WiFi.begin(ssid, password);
19   while (WiFi.status() != WL_CONNECTED) {
20     delay(1000);
21     Serial.println("connecting to WiFi...");
22   }
23 }
```

Serial Monitor

Message (Enter to send message to 'NodeMCU 1.0 (ESP-12E Module)' on COM14)

01:18:34.925 -> Temp: 29.50 C
01:18:34.964 -> Hum: 84.00 %
01:18:37.954 -> Temp: 29.50 C
01:18:38.000 -> Hum: 84.00 %
01:18:40.985 -> Temp: 29.50 C
01:18:40.985 -> Hum: 83.00 %
01:18:44.029 -> Temp: 29.50 C
01:18:44.063 -> Hum: 83.00 %
01:18:47.040 -> Temp: 29.50 C
01:18:47.085 -> Hum: 82.00 %

In the above image we can see both the server and client codes running simultaneously, connected to the same hotspot and sharing the same sensor values in the powershell.



In the above image the LCD Display can be seen displaying the Temperature and Humidity values captured by the sensor.

Conclusion:

We successfully established a connection between the client-server system using the WiFi module connection of the ESP8266. A temperature and humidity sensor was used to record and share the values from server side to the client, over the wireless connection, and was successfully displayed on the client side on a LCD display.

This project is a significant step towards making wireless sensor data communication more accessible, efficient, and user-friendly. It truly exemplifies the potential of technology in making our lives safer and more comfortable.

Acknowledgement:

We thank Prof. Prajeesha and the Department of ECE, PES University, Bengaluru for offering the summer course 'NextGen Communication Systems' and providing us with the opportunity to present our working model based on its applications.