

# Node-RED: Case Study 2

## Temperature & Humidity Monitoring System using ESP32 & DHT11



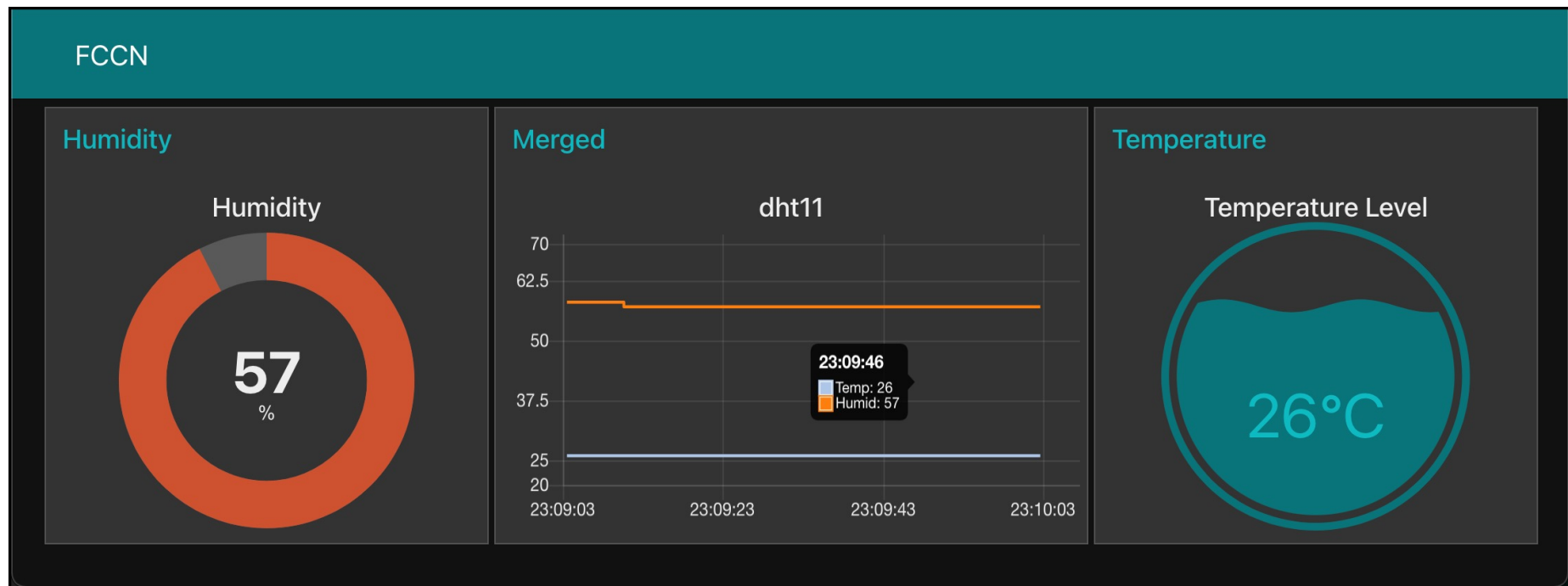
v1 mar2021

iezan74@gmail.com



**Scenario:** To set up temperature & humidity monitoring system dashboard with the following conditions:

- a chart for temperature,
- a chart for humidity,
- a 2 line chart showing temperature & humidity value.



## **Requirement:**

- i. ESP32 x 1,
- ii. DHT11 x1,
- iii. NodeRED – PC or Pi.

## **Methods:**

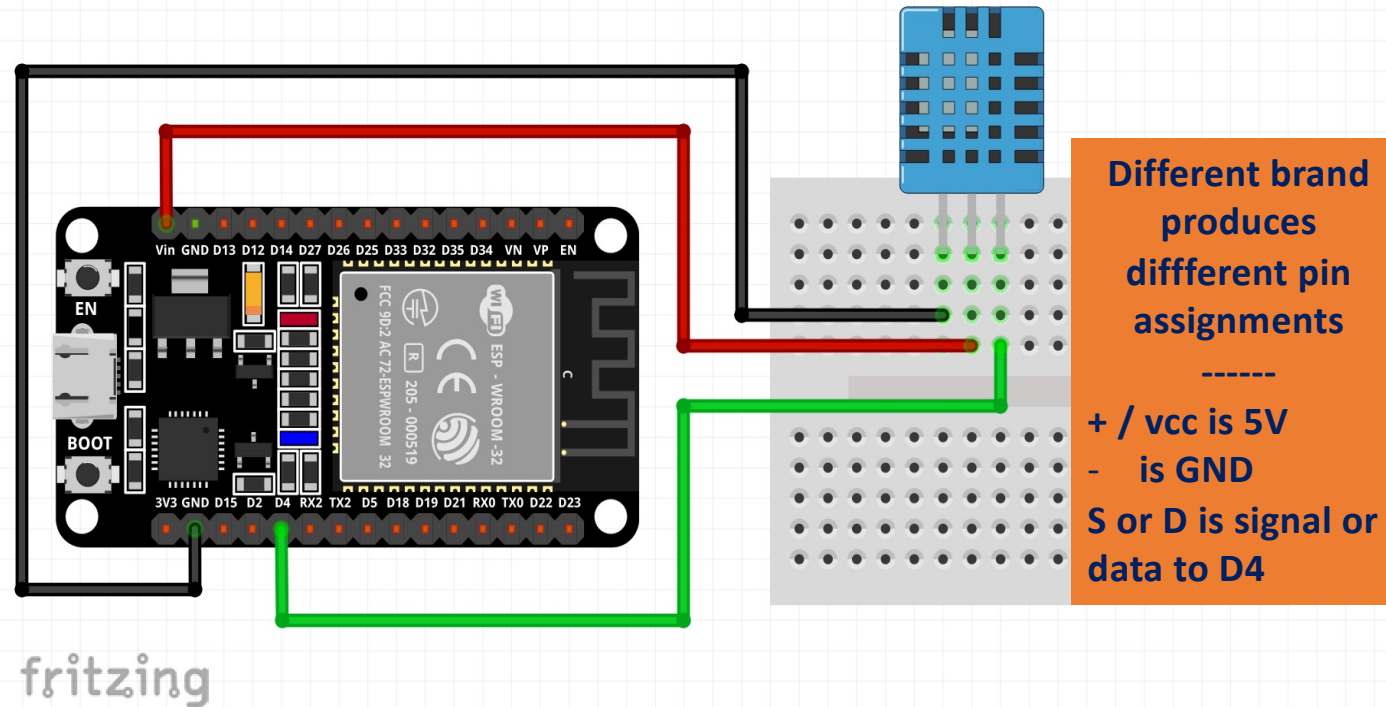
- i. Do wiring connection & upload the sketch into microcontroller.  
Troubleshoot any errors.
- ii. NodeRED configuration: Layout, Nodes & nodes properties
- iii. Test the system

## Microcontroller: a. The Schematic Diagram.

>Connect your board to PC / laptop.

>Make sure correct board name & port is selected.

>Always check your wiring especially the power supply. This might save your money from replacing a burnt device.



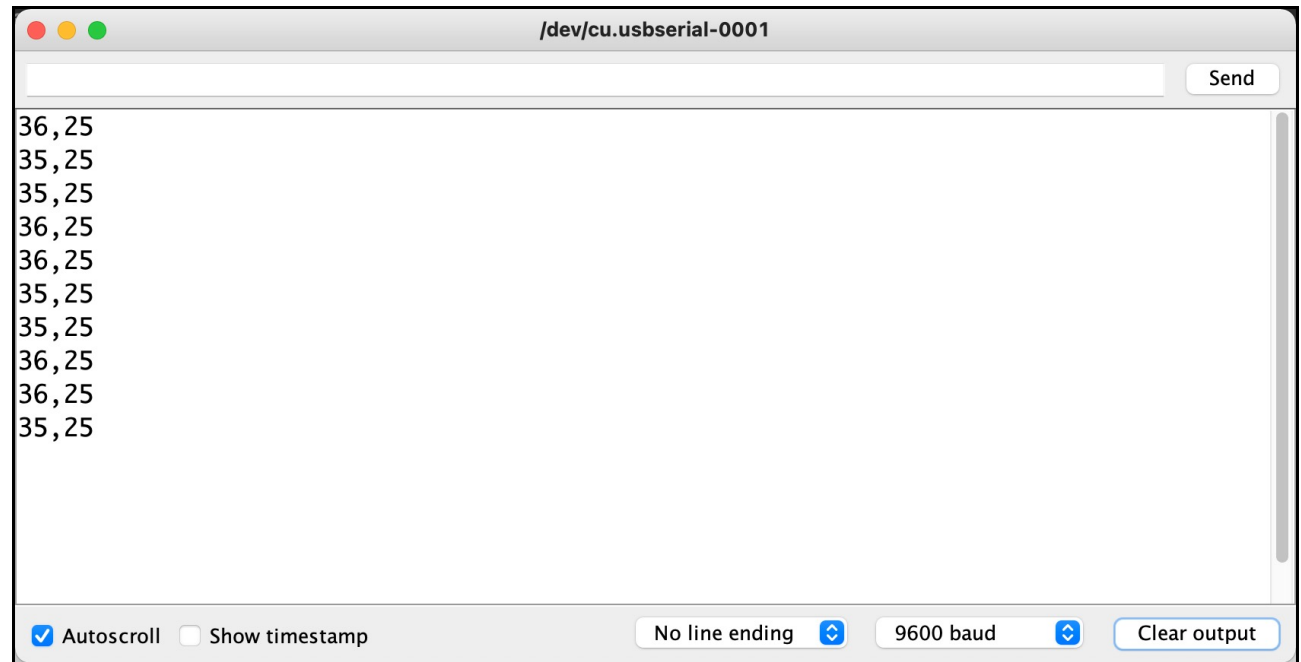
## Microcontroller: b. The sketch.

```
1  #include <dht11.h>
2  dht11 DHT;
3  #define DHT11_PIN 4
4
5  void setup(){
6    Serial.begin(9600);
7  }
8
9  void loop(){
10   int chk;
11   chk = DHT.read(DHT11_PIN);  // READ DATA
12
13   // DISPLAY DATA
14   Serial.print(DHT.humidity,1);
15   Serial.print(",");
16   Serial.println(DHT.temperature,1);
17
18   delay(1000);
19 }
```

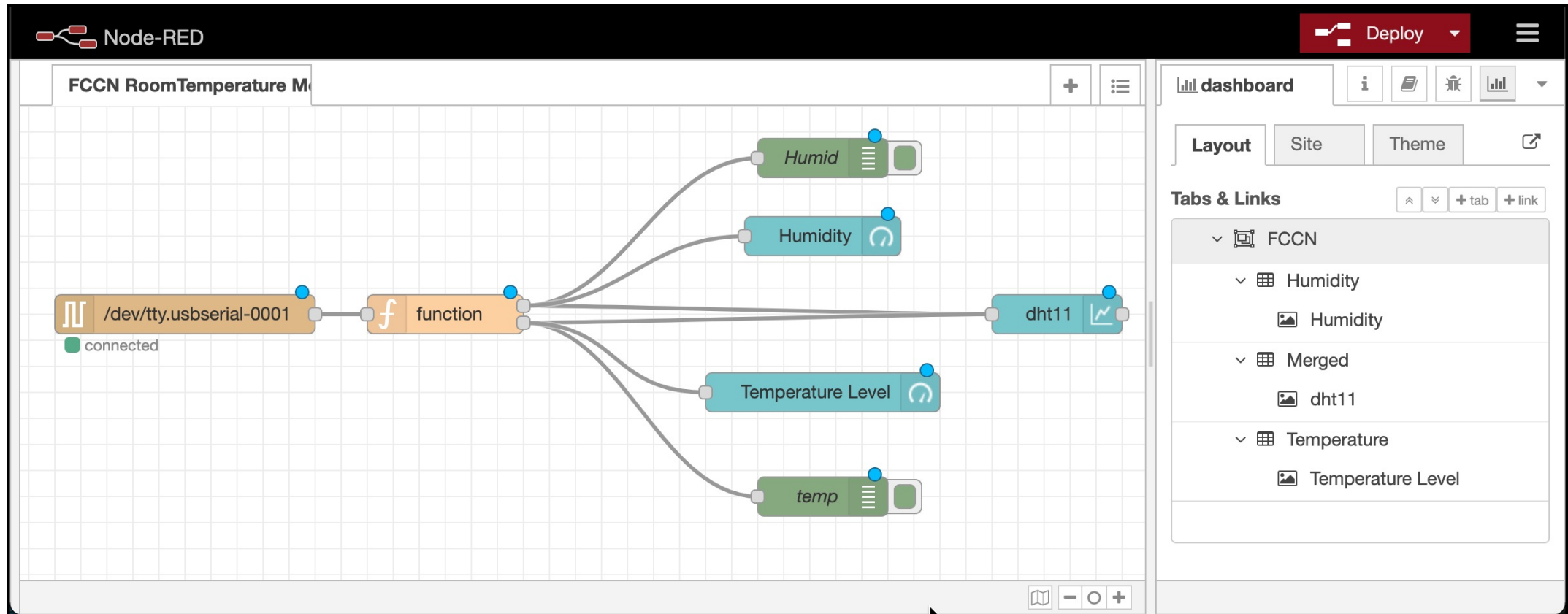
## Microcontroller: c. Expected Output.

> Upload the sketch, open Serial Monitor & adjust the baud rate option.

> Close the Serial Monitor to give a way for NodeRED Serial In node to communicate with ESP32.



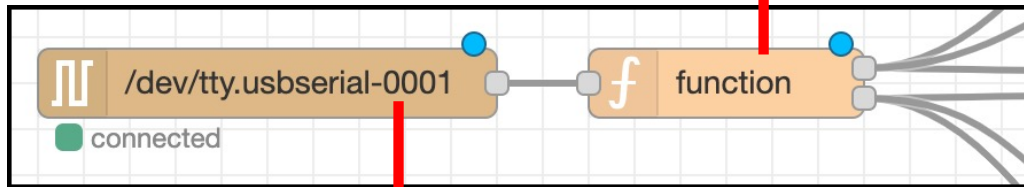
## Node-RED: d. Layout.



[d2-cs2-31mar21-flows-e32-dht11-compressed.json](#)

## BDA CASE STUDY 2: Temperature & Humidity Monitoring System using ESP32 & DHT11

### Node-RED: d. Layout.



#### Edit serial in node

Delete Cancel Done

⚙ Properties

Serial Port

Name

#### Edit function node

Delete Cancel Done

⚙ Properties

Name

Setup Function Close

```
1 var output = msg.payload.split(",");
2 msg={};
3 msg2={};
4
5 var read1 = parseInt(output[0]);
6 var read2 = parseInt(output[1]);
7
8 msg = {payload : read1};
9 msg.topic = 'Humid';
10
11 msg1 = {payload : read2};
12 msg1.topic = 'Temp';
13
14 return [msg,msg1];
```

Outputs 2

```
//COPY ME
var output = msg.payload.split(",");
msg={};
msg2={};
```

```
var read1 = parseInt(output[0]);
var read2 = parseInt(output[1]);
```

```
msg = {payload : read1};
msg.topic = 'Humid';
```

```
msg1 = {payload : read2};
msg1.topic = 'Temp';
```

```
return [msg,msg1];
```

8



## BDA CASE STUDY 2: Temperature & Humidity Monitoring System using ESP32 & DHT11

### Node-RED: d. Layout.

The image displays the Node-RED interface for configuring a Temperature Level gauge node and its associated debug node. The central workspace shows a 'Temperature Level' gauge node (light blue) and a 'temp' node (green). Red arrows indicate the flow of data from the 'temp' node to the 'Temperature Level' gauge node and from the 'Temperature Level' gauge node to the 'Edit debug node' panel.

**Edit gauge node**

Properties:

- Group: [FCCN] Temperature
- Size: auto
- Type: Level
- Label: Temperature Level
- Units: °C
- Range: min 15, max 35
- Name:

☐ Enabled

**Edit debug node**

Properties:

- Output: msg. payload
- To: ☒ debug window, ☐ system console, ☐ node status (32 characters)
- Name: Humid

## BDA CASE STUDY 2: Temperature & Humidity Monitoring System using ESP32 & DHT11

### Node-RED: d. Layout.

The image displays the Node-RED interface for configuring a Temperature & Humidity Monitoring System. The central workspace shows two nodes: 'Humid' (green) and 'Humidity' (teal). Red arrows point from these nodes to their respective configuration panels.


**Edit gauge node (Humidity):**

- Group: [FCCN] Humidity
- Size: auto
- Type: Donut
- Label: Humidity Level
- Value format: {{value}}
- Units: %
- Range: min 20, max 60
- Enabled: ☐

**Edit debug node (Humid):**

- Output: msg. payload
- To: ☒ debug window, ☐ system console, ☐ node status (32 characters)
- Name: Humid

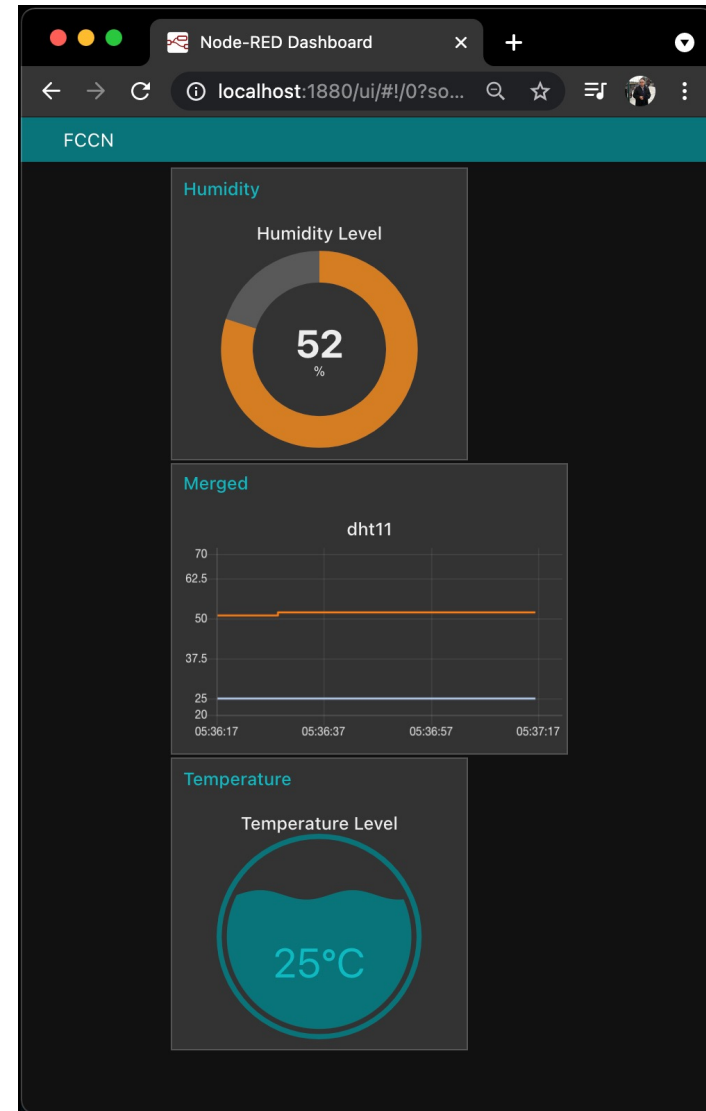
### Node-RED: d. Layout.



The debug console displays a list of messages from the 'all nodes' tab. The messages are organized by node type and timestamp.

Node	Timestamp	Humid (msg.payload : number)	Temp (msg.payload : number)
Humid	31/03/2021, 05:34:49	51	
Temp	31/03/2021, 05:34:49		24
Humid	31/03/2021, 05:34:50	51	
Temp	31/03/2021, 05:34:50		24
Humid	31/03/2021, 05:34:51	51	
Temp	31/03/2021, 05:34:51		24

v1-mar-21



## QUESTIONS?



Questions?

TQ & BYE

END