

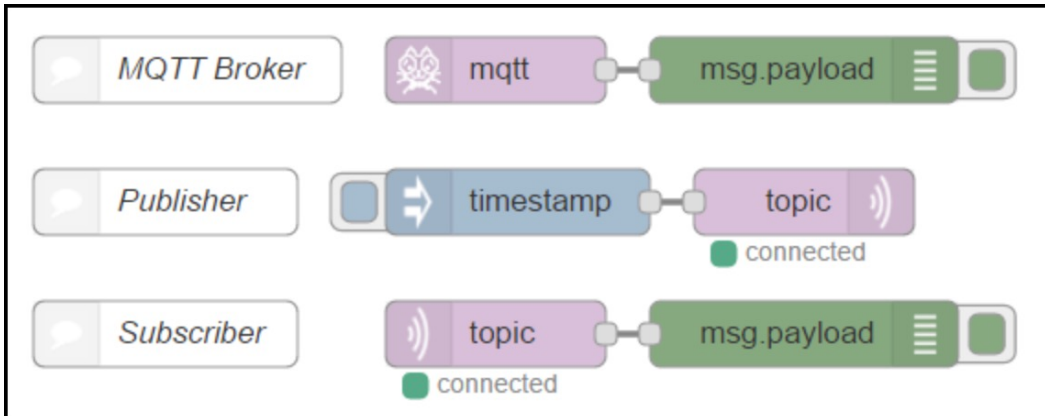
IoT MQTT with NodeRED Monitoring Server Room Temperature & Humidity Level

v1 mar2021

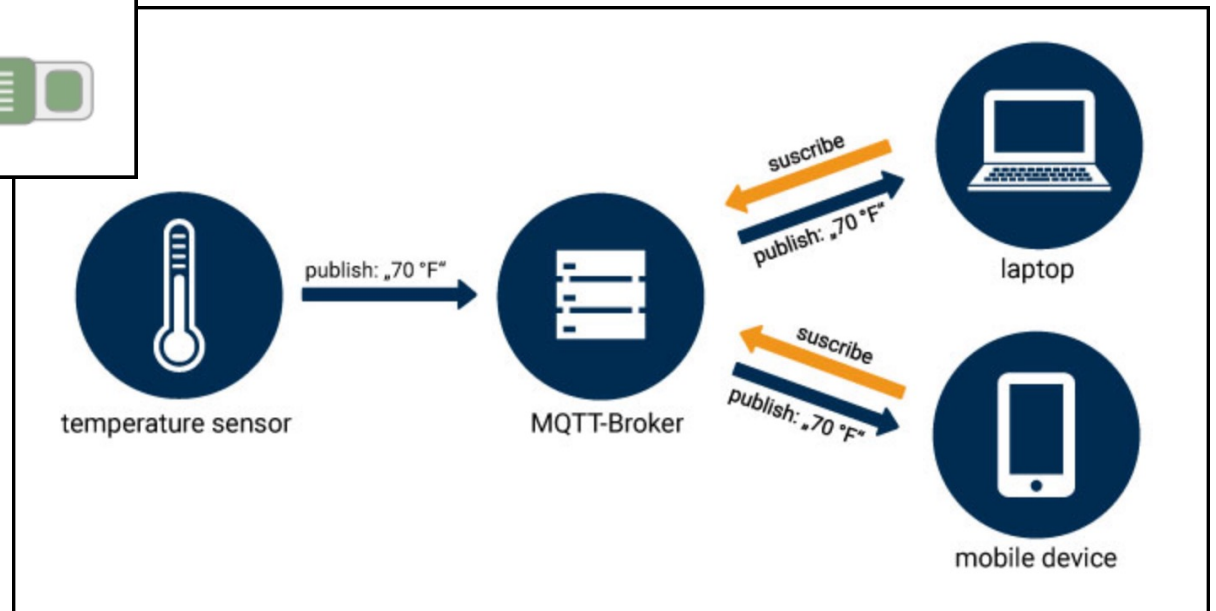
iezan74@gmail.com



Scenario: Display temperature & humidity data on Node-RED using MQTT communication protocol.

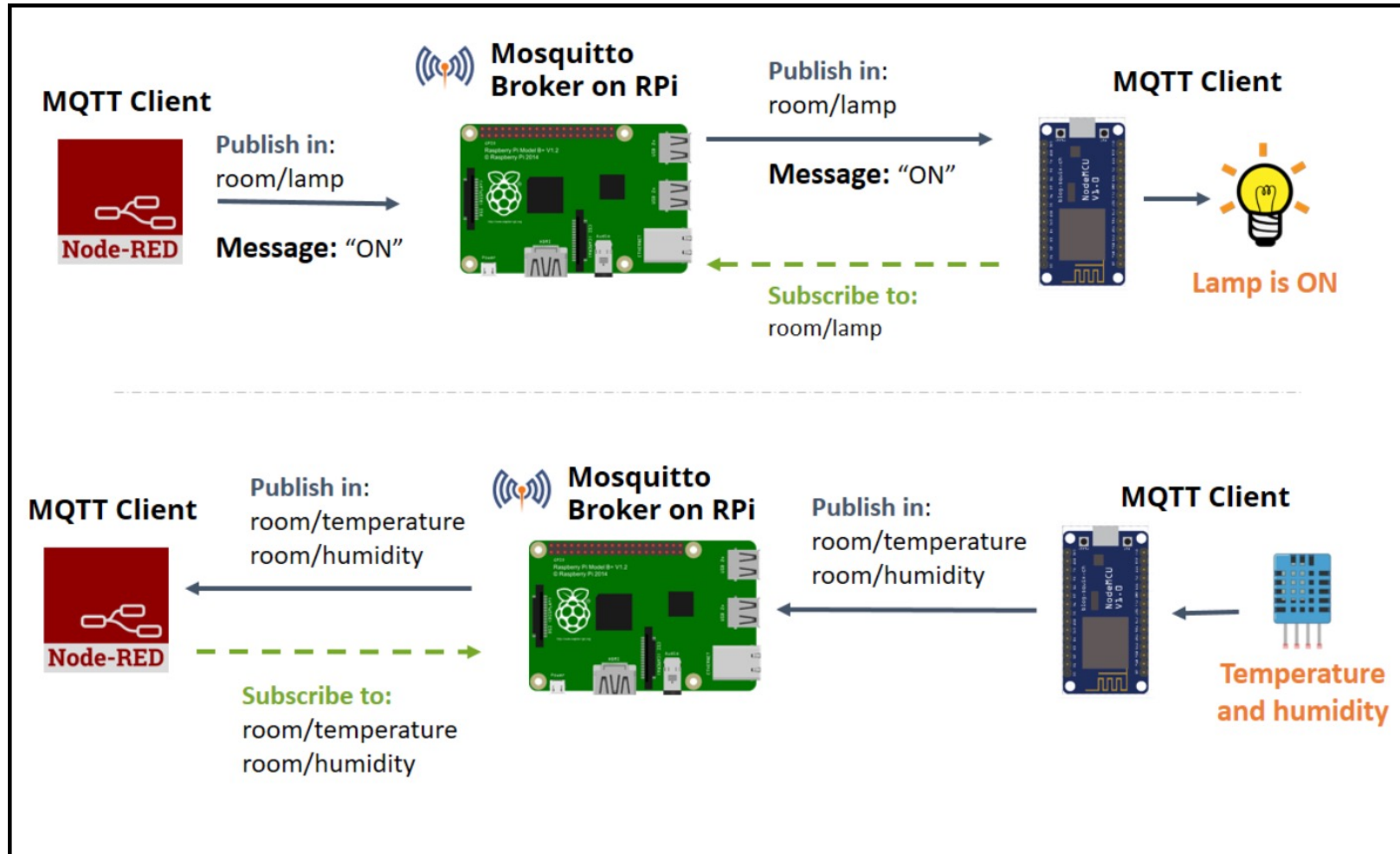


<https://flows.nodered.org/node/node-red-contrib-mqtt-broker>



<https://www.opc-router.com/what-is-mqtt/>

MQTT



Requirement:

- i. Experience with Node-RED, DHT11 & ESP32 would be an advantage.
- ii. DHT 11 & ESP32 (with its accessories).
- ii. PC/Laptop/Raspberry Pi.

To Do:

A. Hardware Section

- i. Download & Install Arduino IDE
- ii. Install ESP32 board
- iii. Download DHT11 library & client library for MQTT messaging – PubSubClient
- iv. Connect/assemble & test hardware <https://www.arduino.cc/reference/en/libraries/pubsubclient/>
- v. Download sketch from xxxx & test.

B. Node-RED Section

- i. Install Aedes Broker.

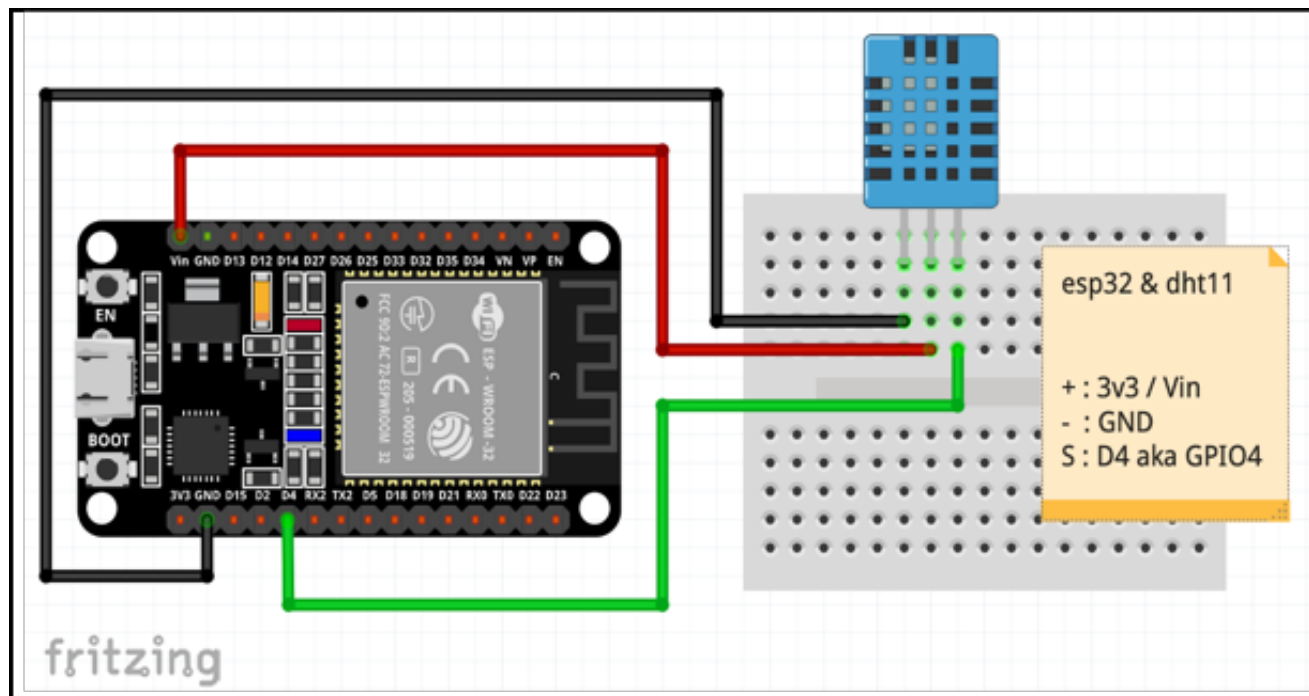
C. System Deployment

- i. Part A + Part B.

D. MySQL (if time permits)

A. HARDWARE SECTION

i. Board Connection



B. PROGRAMMING SECTION

i. Download Ardiono's PubSubClient Library

PubSubClient - Arduino Reference

arduino.cc/reference/en/libraries/pubsubclient/

PROFESSIONAL EDUCATION STORE

Search on Arduino.cc

SIGN IN

HARDWARE SOFTWARE CLOUD DOCUMENTATION COMMUNITY BLOG ABOUT

Reference > Libraries > Pubsubclient

PubSubClient

Communication

A client library for MQTT messaging.

MQTT is a lightweight messaging protocol ideal for small devices. This library allows you to send and receive MQTT messages. It supports the latest MQTT 3.1.1 protocol and can be configured to use the older MQTT 3.1 if needed. It supports all Arduino Ethernet Client compatible hardware, including the Intel Galileo/Edison, ESP8266 and TI CC3000.

Author: Nick O'Leary

Maintainer: Nick O'Leary

[Read the documentation](#)

[Go to repository](#)

Compatibility

This library is compatible with **all** architectures so you should be able to use it on all the Arduino boards.

Releases

To use this library, open the [Library Manager](#) in the Arduino IDE and install it from there.

- 2.8.0 (latest)
- 2.7.0
- 2.6.0

10% OFF

Find anything that can be improved? Suggest corrections and new documentation via [GitHub](#).

Doubts on how to use Github? Learn everything you need to know in [this tutorial](#).

Arduino Day Special Deals

Help

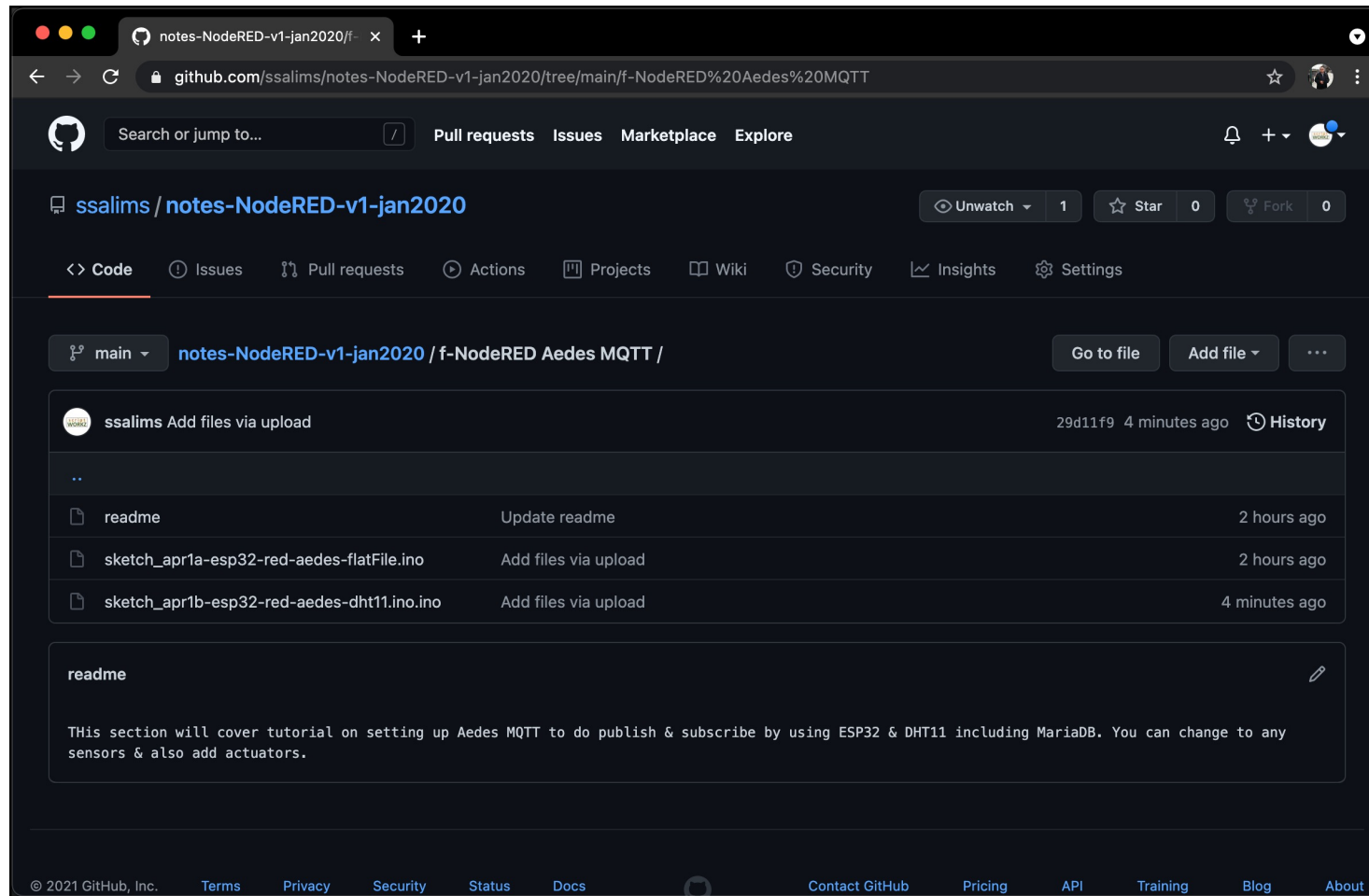
Link: [PubSubClient](#)
Sketch>Include Library>Add .Zip Library...

<https://www.arduino.cc/reference/en/libraries/pubsubclient/>

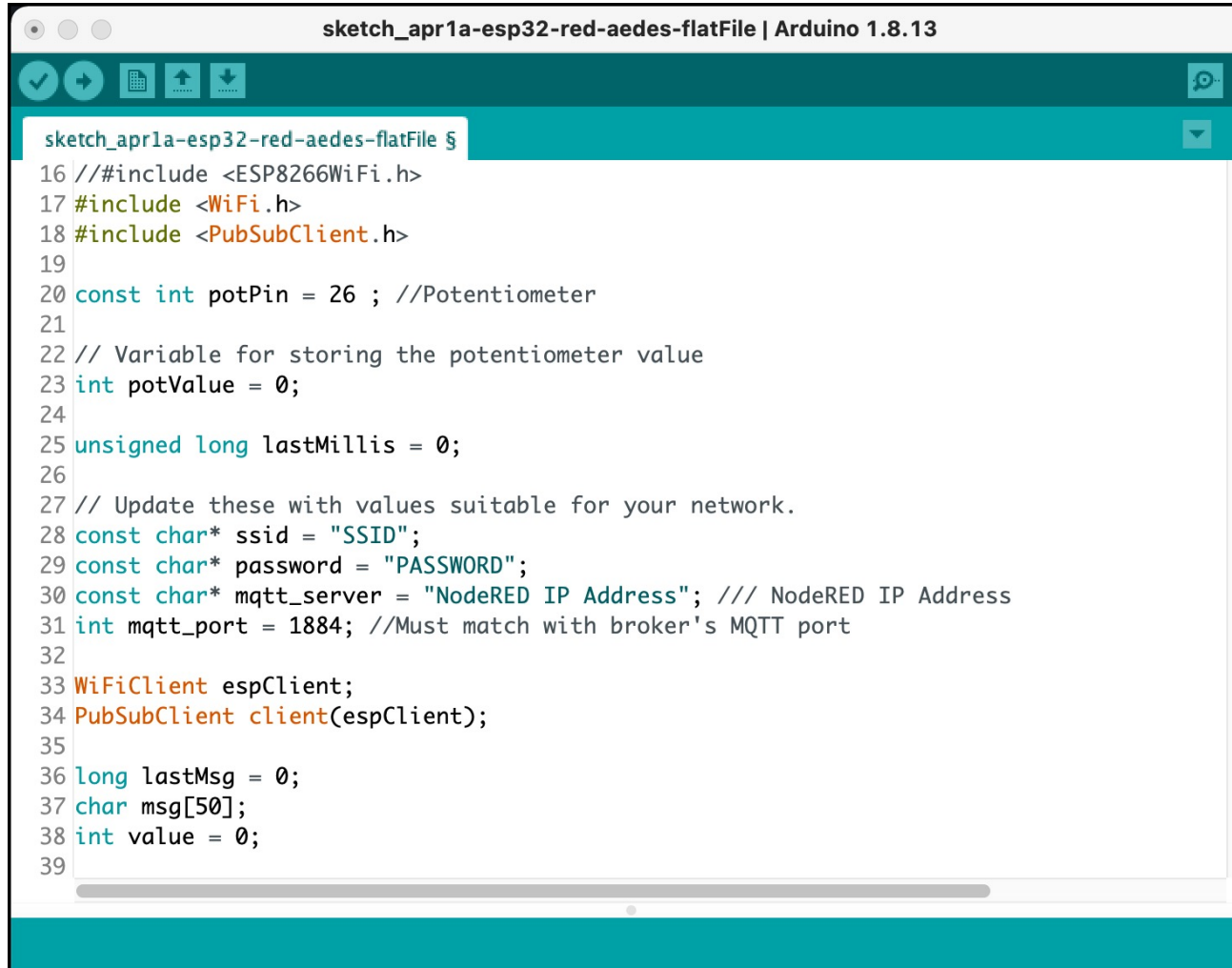
v1-mar-21

ii. Download Sketches from File Repo

<https://github.com/ssalims/notes-NodeRED-v1-jan2020/tree/main/f-NodeRED%20Aedes%20MQTT>



iii. Arduino Code Walkthrough

A screenshot of the Arduino IDE interface. The title bar at the top reads "sketch_apr1a-esp32-red-aedes-flatFile | Arduino 1.8.13". Below the title bar is a toolbar with icons for checking, running, uploading, and saving. The main text area contains C++ code for an ESP32. The code includes headers for ESP8266 WiFi, WiFi, and PubSubClient. It defines a potentiometer pin (26) and a variable for its value. It also sets up network credentials (SSID, password) and MQTT server details (server address, port). The code declares a WiFiClient, a PubSubClient, and a message buffer. The code is as follows:

```
sketch_apr1a-esp32-red-aedes-flatFile $
16 //#include <ESP8266WiFi.h>
17 #include <WiFi.h>
18 #include <PubSubClient.h>
19
20 const int potPin = 26 ; //Potentiometer
21
22 // Variable for storing the potentiometer value
23 int potValue = 0;
24
25 unsigned long lastMillis = 0;
26
27 // Update these with values suitable for your network.
28 const char* ssid = "SSID";
29 const char* password = "PASSWORD";
30 const char* mqtt_server = "NodeRED IP Address"; /// NodeRED IP Address
31 int mqtt_port = 1884; //Must match with broker's MQTT port
32
33 WiFiClient espClient;
34 PubSubClient client(espClient);
35
36 long lastMsg = 0;
37 char msg[50];
38 int value = 0;
39
```

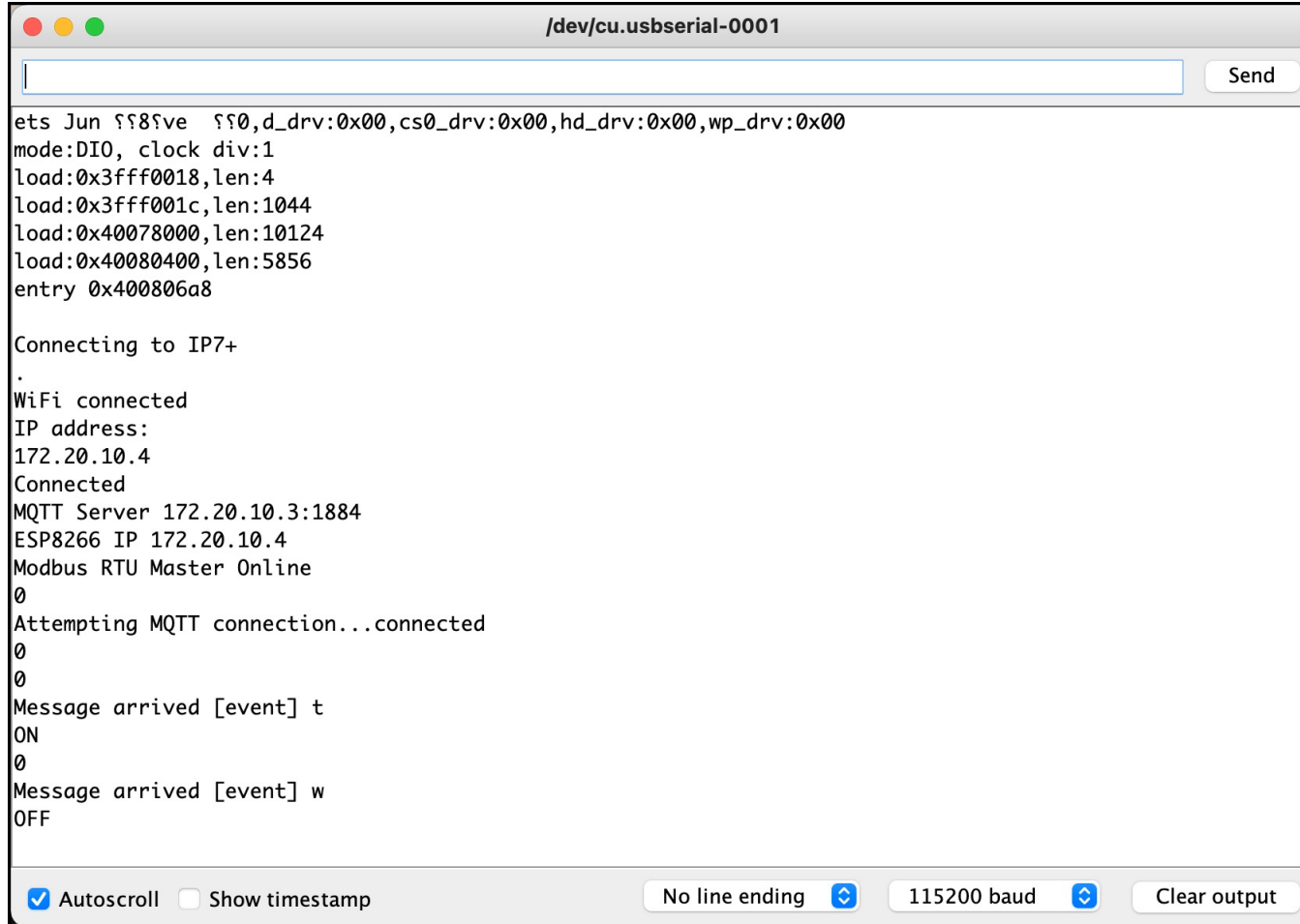
iii. Arduino Code Walkthrough

```
75 void callback(char* topic, byte* payload, unsigned int length) {  
76     //Serial.println("Callback"); // Flag  
77     /* To display message sent from broker */  
78     Serial.print("Message arrived [");  
79     Serial.print(topic);  
80     Serial.print("] ");  
81  
82     for (int i = 0; i < length; i++) {  
83         Serial.print((char)payload[i]);  
84     }  
85     /* ----- */  
86  
87     /* This section is for controlling output: LED, servo etc */  
88     Serial.println();  
89     if ((char)payload[0] == 't') {  
90         Serial.println("ON");  
91         digitalWrite(LED_BUILTIN, HIGH);  
92     } else {  
93         Serial.println("OFF");  
94         digitalWrite(LED_BUILTIN, LOW);  
95     }  
96  
97     /* ----- */  
98 }
```

iii. Arduino Code Walkthrough

```
99 void reconnect() {
100
101 // Loop until we're reconnected
102 while (!client.connected()) {
103     Serial.print("Attempting MQTT connection...");
104 // Attempt to connect
105     if (client.connect("ESP32Client")) {
106         Serial.println("Broker connected to ESP32");
107         client.subscribe("event"); // Topic at ESP32
108     } else {
109         Serial.print("failed, rc=");
110         Serial.print(client.state());
111         Serial.println(" try again in 5 seconds");
112         // Wait 5 seconds before retrying
113         delay(5000);
114     }
115 }
116 }
```

iii. Arduino Code Walkthrough



The screenshot shows a serial monitor window titled "/dev/cu.usbserial-0001". The output text is as follows:

```
ets Jun  8 15:56:12, d_drv:0x00,cs0_drv:0x00,hd_drv:0x00,wp_drv:0x00
mode:DIO, clock div:1
load:0x3fff0018,len:4
load:0x3fff001c,len:1044
load:0x40078000,len:10124
load:0x40080400,len:5856
entry 0x400806a8

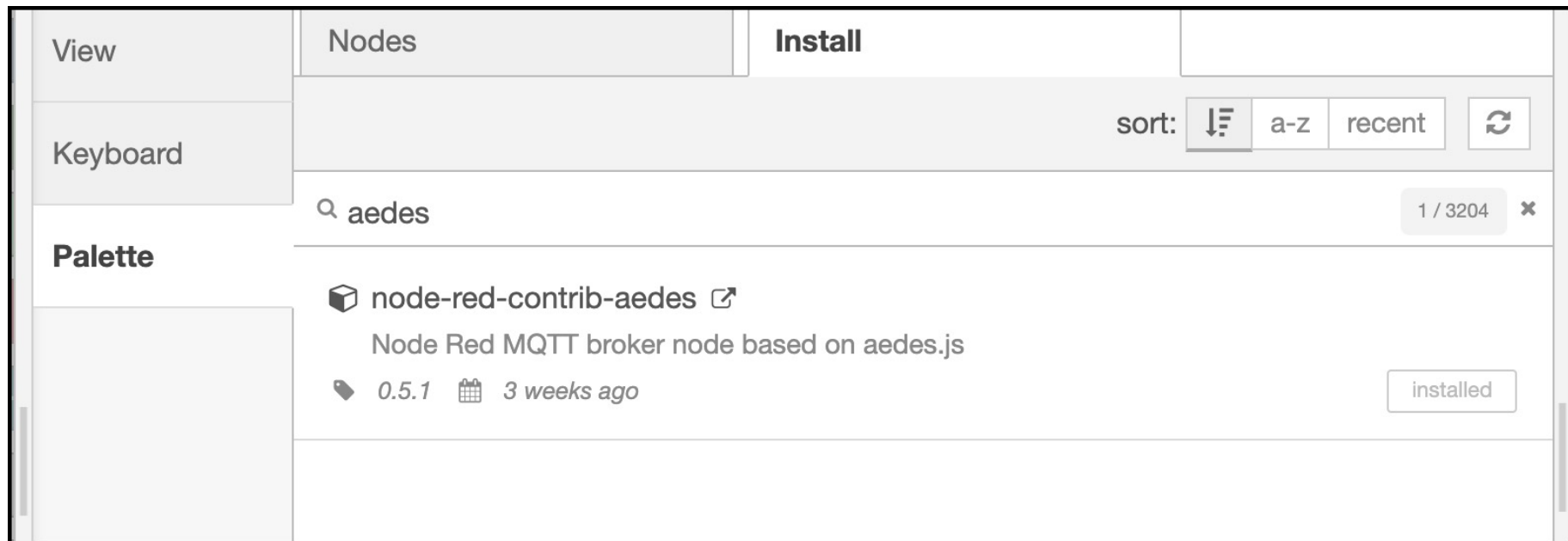
Connecting to IP7+
.
WiFi connected
IP address:
172.20.10.4
Connected
MQTT Server 172.20.10.3:1884
ESP8266 IP 172.20.10.4
Modbus RTU Master Online
0
Attempting MQTT connection...connected
0
0
Message arrived [event] t
ON
0
Message arrived [event] w
OFF
```

At the bottom of the window, there are controls for the serial monitor: ☒ Autoscroll, ☐ Show timestamp, a dropdown menu set to "No line ending", a dropdown menu set to "115200 baud", and a "Clear output" button.

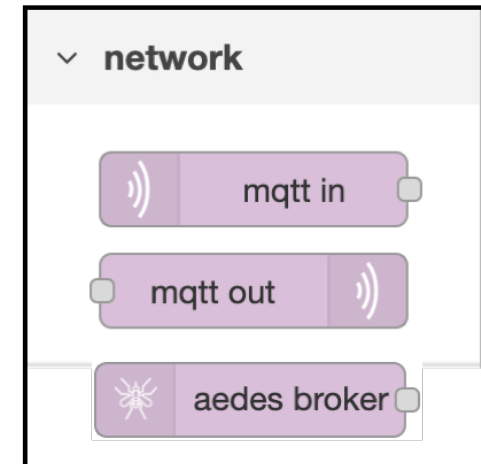
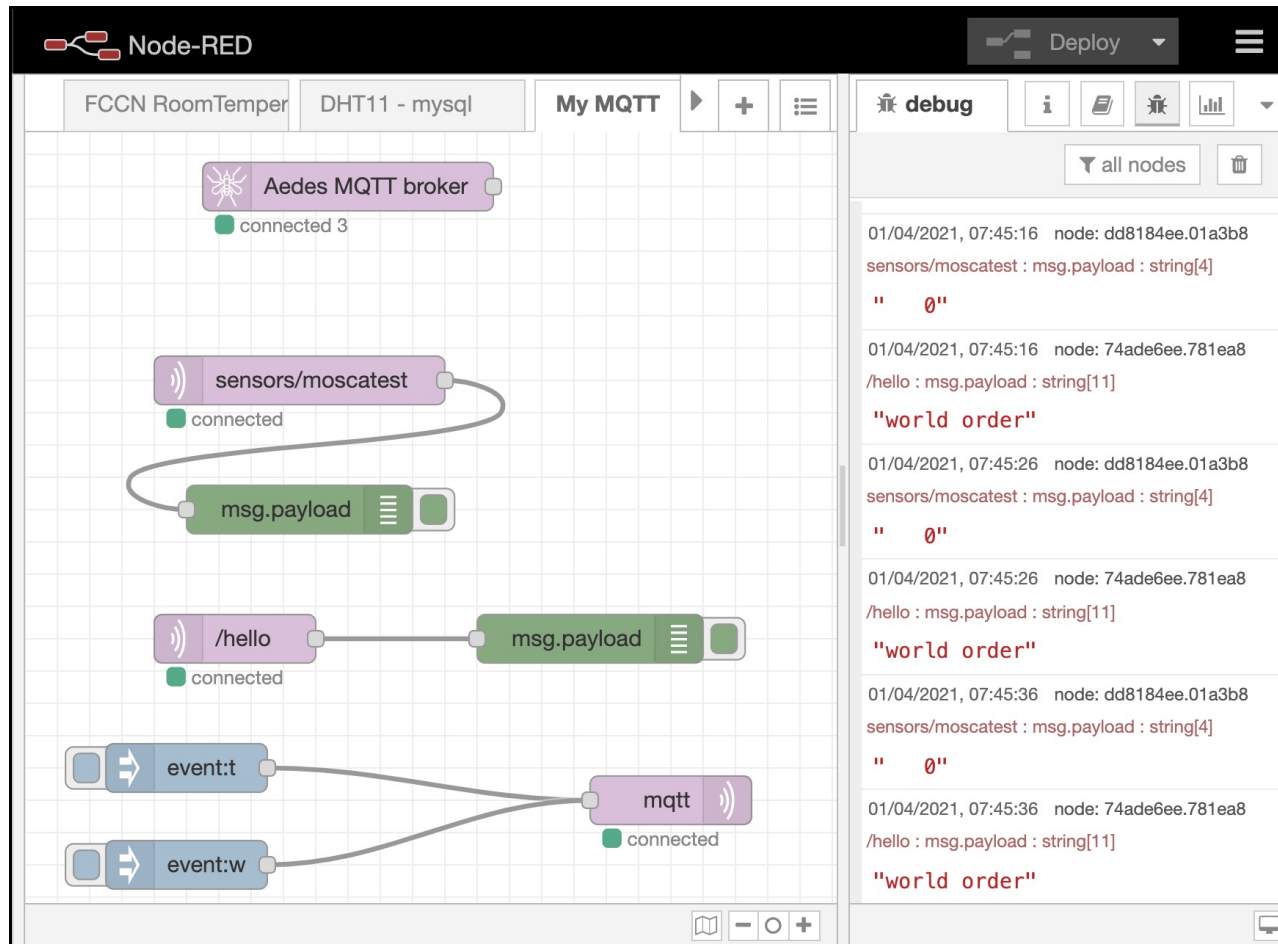
v1-mar-21

C. NodeRED SECTION

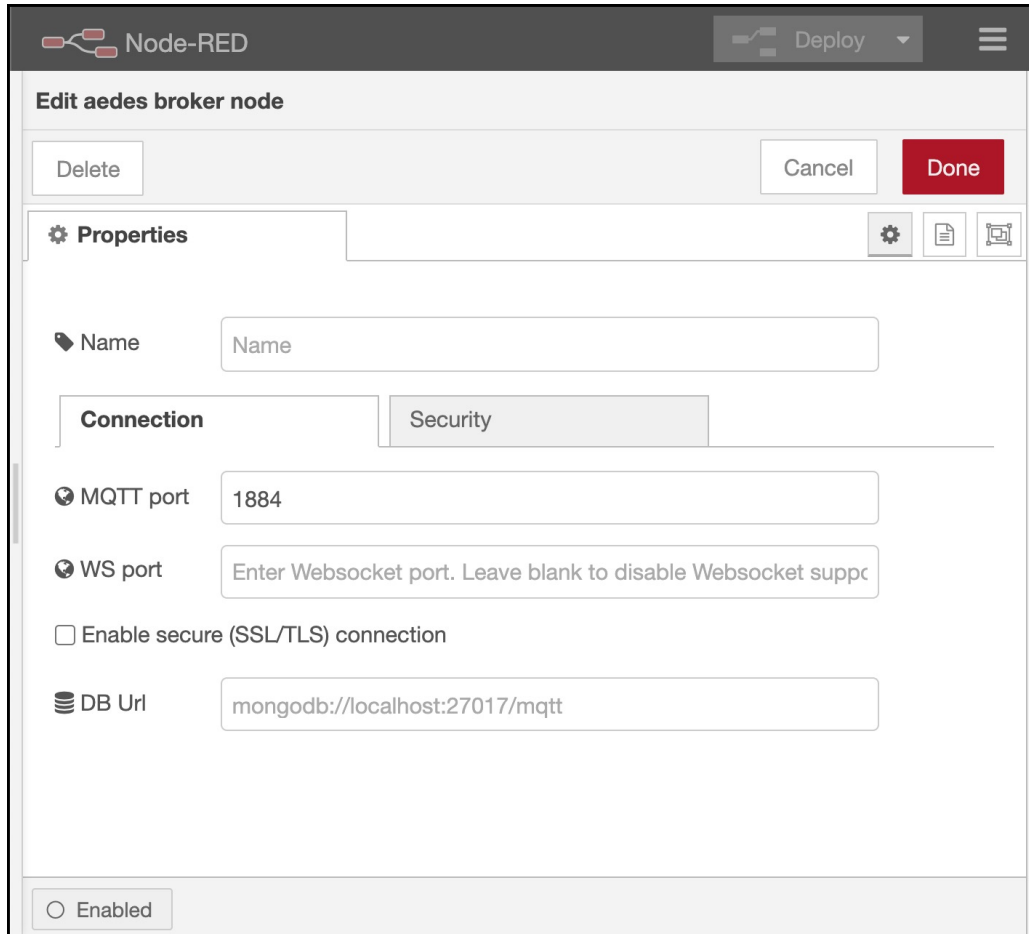
i. Install *Aedes* Palette



ii. Setup Workspace



iii. Edit Aedes Broker Properties



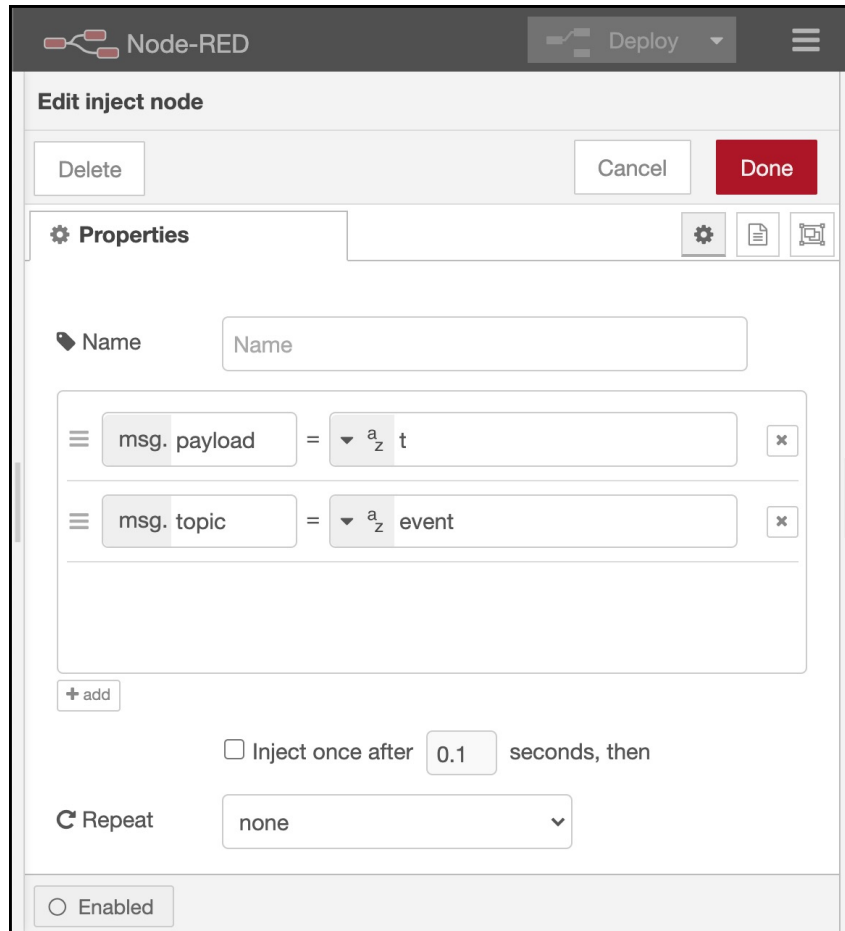
The screenshot shows the 'Edit aedes broker node' configuration window in Node-RED. The window has a dark header with the Node-RED logo and a 'Deploy' button. Below the header, there are 'Delete', 'Cancel', and 'Done' buttons. The main area is titled 'Edit aedes broker node' and contains a 'Properties' tab. Under the 'Properties' tab, there are two sub-tabs: 'Connection' and 'Security'. The 'Connection' sub-tab is active and shows the following fields: 'Name' (with a placeholder 'Name'), 'MQTT port' (with the value '1884'), 'WS port' (with the placeholder 'Enter Websocket port. Leave blank to disable Websocket supp'), 'Enable secure (SSL/TLS) connection' (unchecked), and 'DB Url' (with the value 'mongodb://localhost:27017/mqtt'). At the bottom of the window, there is a radio button labeled 'Enabled' which is selected.

iv. Edit 2 x MQTT In Nodes Properties

The screenshot shows the 'Edit mqtt in node' dialog in Node-RED. At the top, there are 'Delete', 'Cancel', and 'Done' buttons. Below is a 'Properties' section with a settings icon, a document icon, and a preview icon. The properties are: 'Server' (localhost:1884), 'Topic' (sensors/moscatest), 'QoS' (2), 'Output' (auto-detect (string or buffer)), and 'Name' (Name). At the bottom, there is an 'Enabled' checkbox.

The screenshot shows the 'Edit mqtt in node' dialog in Node-RED. At the top, there are 'Delete', 'Cancel', and 'Done' buttons. Below is a 'Properties' section with a settings icon, a document icon, and a preview icon. The properties are: 'Server' (localhost:1884), 'Topic' (/hello), 'QoS' (2), 'Output' (auto-detect (string or buffer)), and 'Name' (Name). At the bottom, there is an 'Enabled' checkbox.

v. Edit 2 x Inject Nodes Properties



The screenshot shows the 'Edit inject node' dialog in Node-RED. At the top, there are 'Delete', 'Cancel', and 'Done' buttons. Below is a 'Properties' section with a 'Name' field. The main area contains two property rows: 'msg. payload' set to 't' and 'msg. topic' set to 'event'. Each row has a menu icon on the left and a delete 'x' icon on the right. An '+ add' button is at the bottom left. At the bottom, there is a checkbox for 'Inject once after 0.1 seconds, then' and a 'Repeat' dropdown menu set to 'none'. The 'Enabled' radio button is selected at the very bottom.

Node-RED

Deploy

Edit inject node

Delete Cancel Done

Properties

Name

msg. payload = t

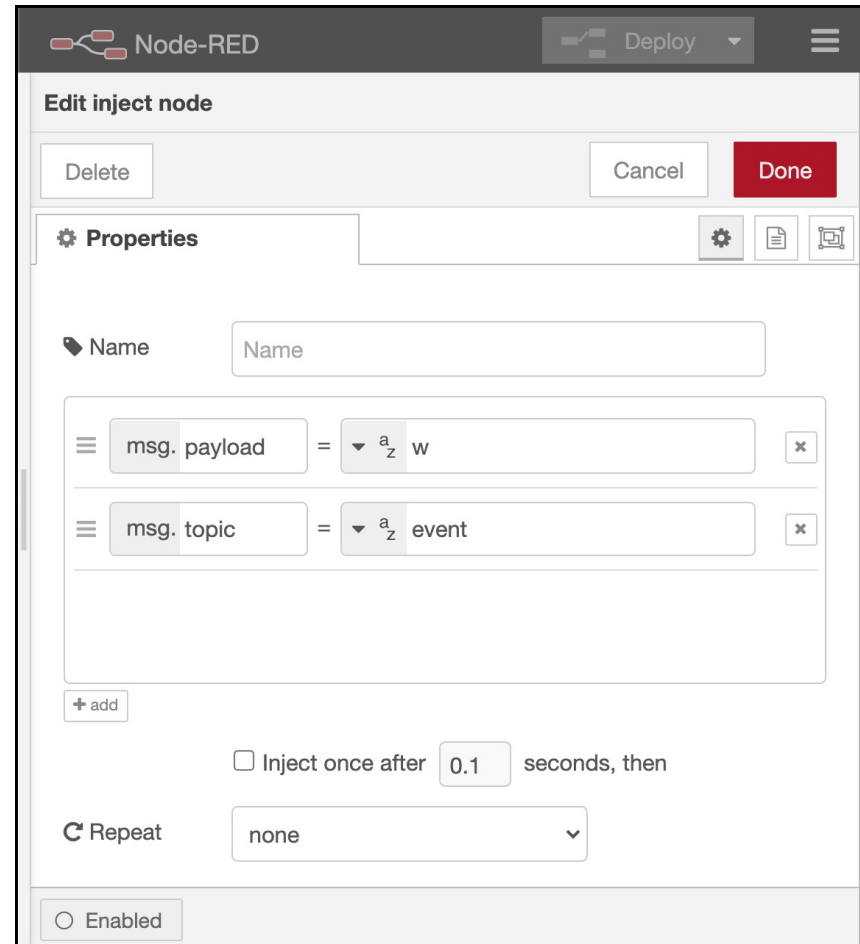
msg. topic = event

+ add

☐ Inject once after 0.1 seconds, then

Repeat none

☒ Enabled



This screenshot is identical to the previous one, but the 'msg. payload' property is set to 'w' instead of 't'. All other elements, including the 'msg. topic' property, timing settings, and the 'Enabled' status, are the same.

Node-RED

Deploy

Edit inject node

Delete Cancel Done

Properties

Name

msg. payload = w

msg. topic = event

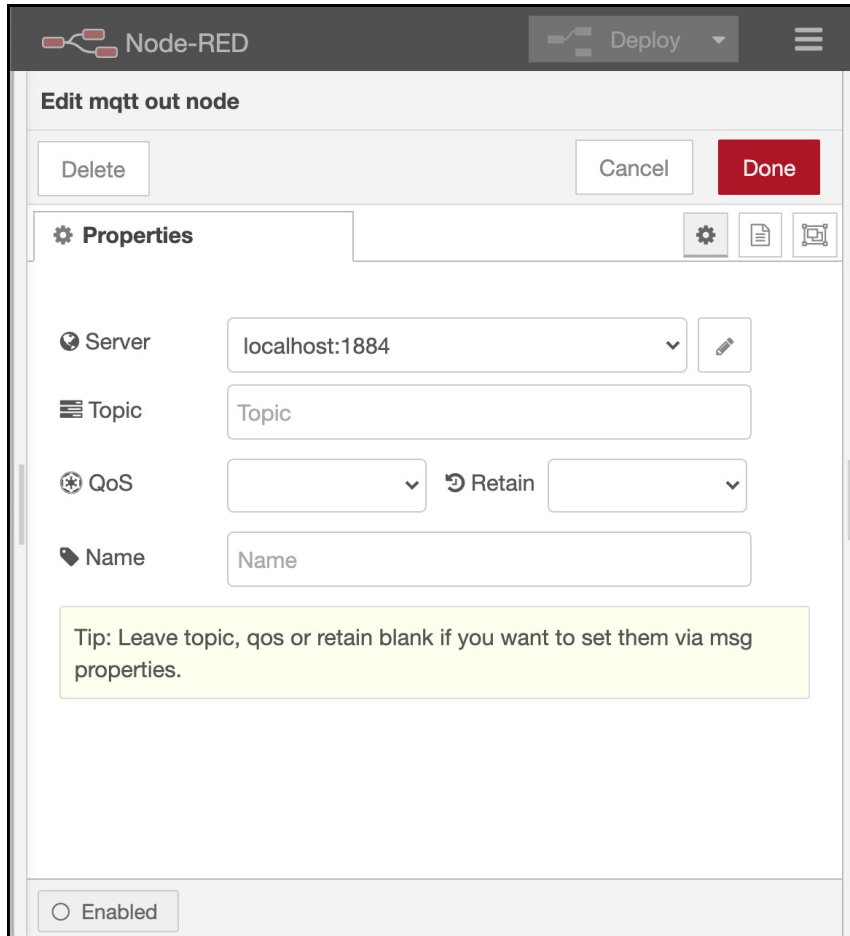
+ add

☐ Inject once after 0.1 seconds, then

Repeat none

☒ Enabled

vi. Edit MQTT Out Node Properties



The screenshot shows the 'Edit mqtt out node' dialog box in Node-RED. The dialog has a title bar with the Node-RED logo and a 'Deploy' button. Below the title bar are three buttons: 'Delete', 'Cancel', and 'Done'. The main section is titled 'Properties' and contains several input fields: 'Server' (a dropdown menu showing 'localhost:1884'), 'Topic' (a text input field), 'QoS' (a dropdown menu), and 'Name' (a text input field). There is also a 'Retain' checkbox. A yellow tip box at the bottom states: 'Tip: Leave topic, qos or retain blank if you want to set them via msg properties.' At the bottom left, there is an 'Enabled' checkbox.

Node-RED

Deploy

Edit mqtt out node

Delete Cancel Done

Properties

Server localhost:1884

Topic Topic

QoS Retain

Name Name

Tip: Leave topic, qos or retain blank if you want to set them via msg properties.

Enabled

D. phpMyAdmin Section

d. phpMyAdmin

Database Name	datacenters_activities		
Table Name	room_dc230		
3 Columns			
NAME	TYPE	LENGTH	ADDITIONAL SETTING
id	INT	11	Index: PRIMARY
			A.I.: <input checked="" type="checkbox"/> **A.I. = Auto Increment
logDateTime	DATETIME	NA	Default: CURRENT_TIMESTAMP
rackID	VARCHAR	255	NA
tempValue	Varchar	255	NA
humidValue	Varchar	255	NA
rackStatus	Varchar	255	NA

QnA

END