

input

inject

catch

status

link

mqtt

http

websocket

tcp

udp

output

debug

link

IOT เบื้องต้น

ด้วยบอร์ด ESP32

นำเสนอด้วย
ESL



Node-RED

Topic

What we will learn today

01

Install Node-RED

02

connect wifi (ESP32)

03

connect mqtt (ESP32)

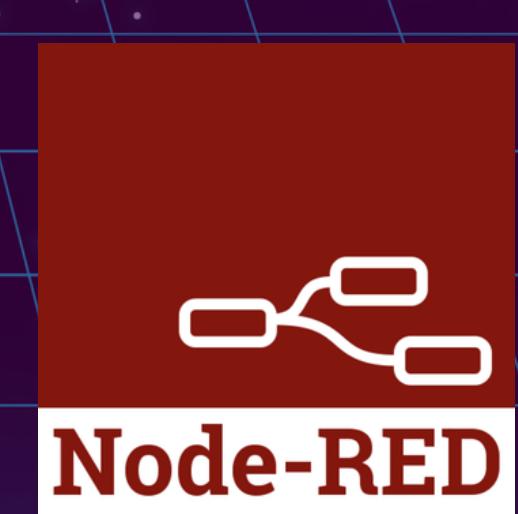
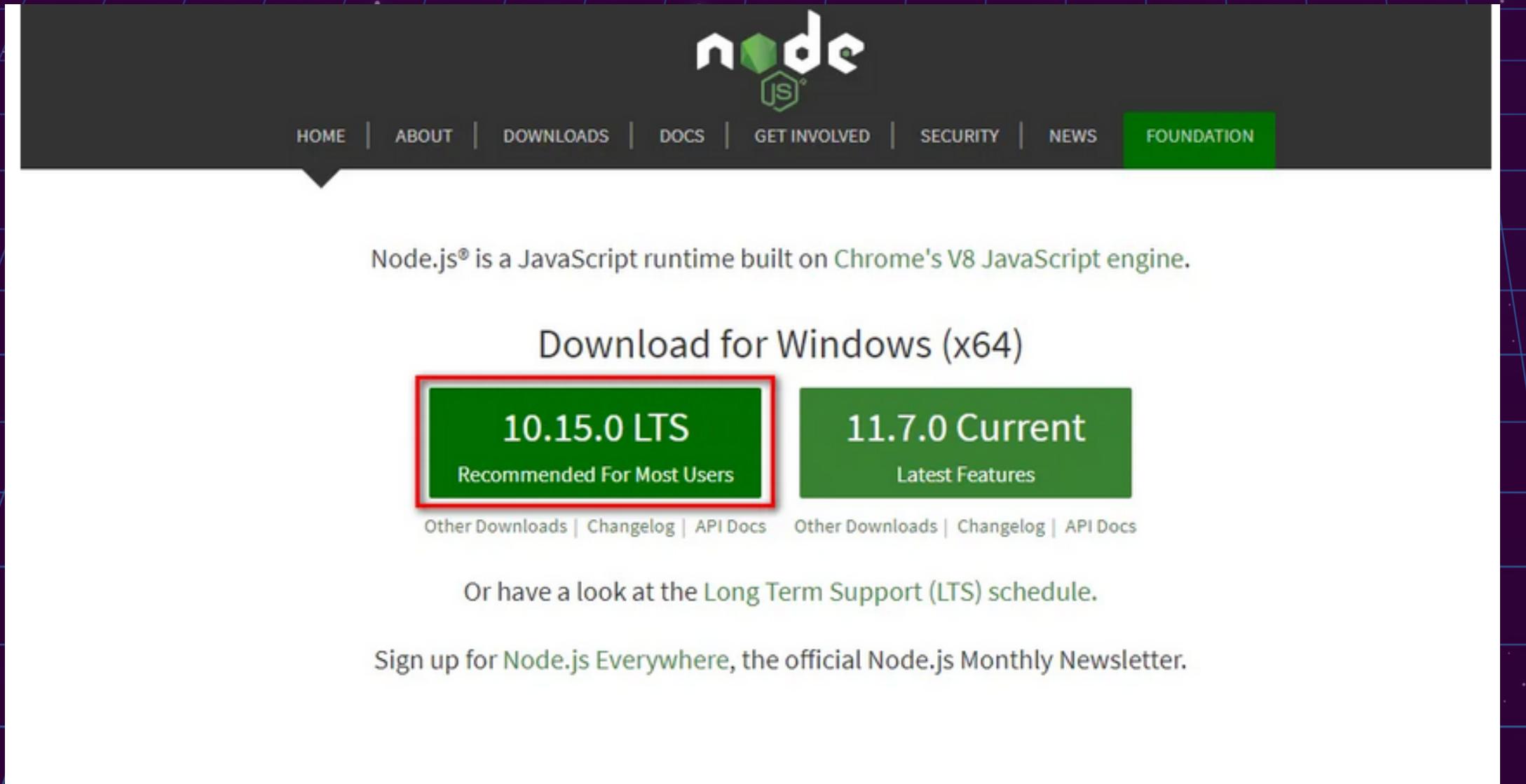
04

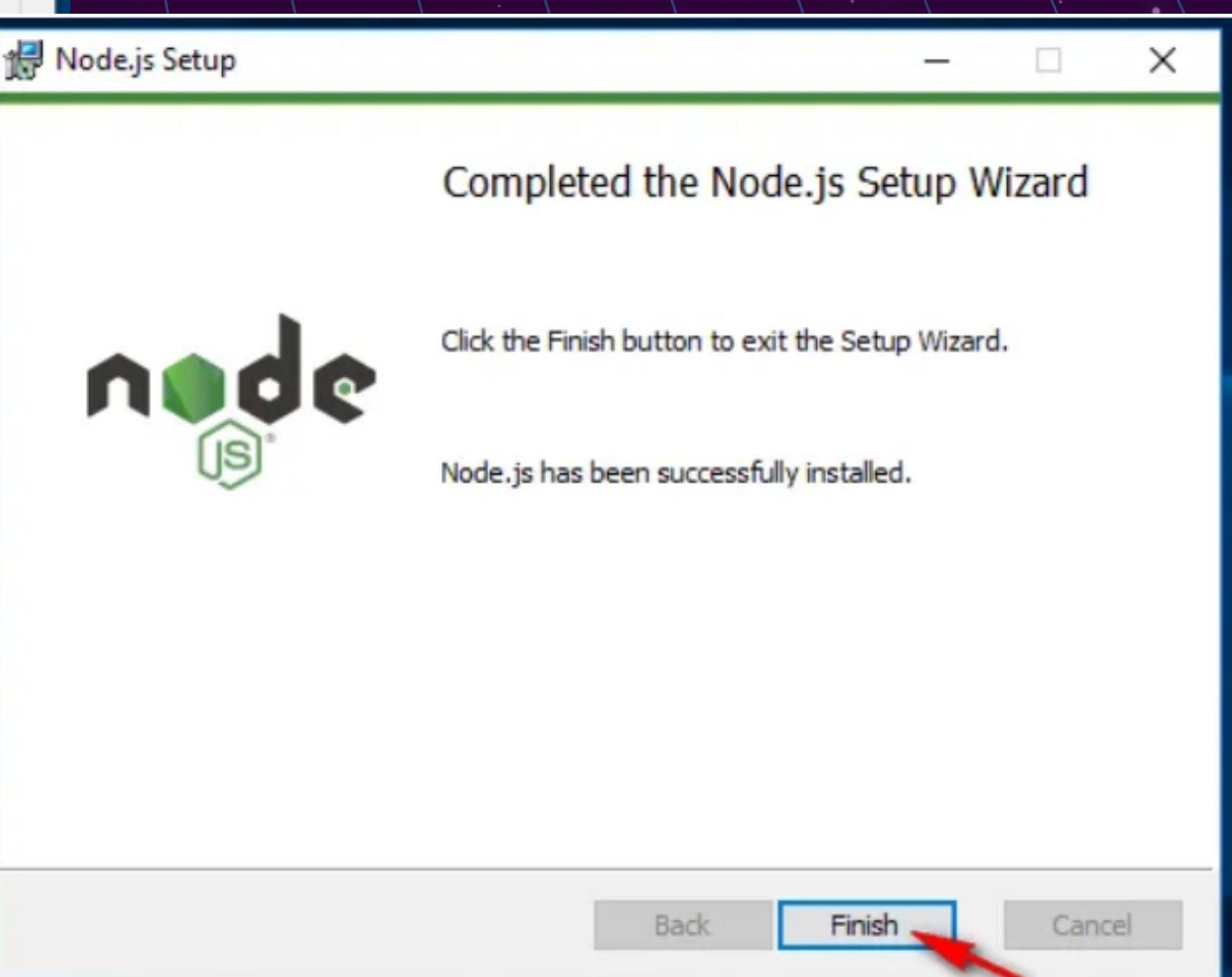
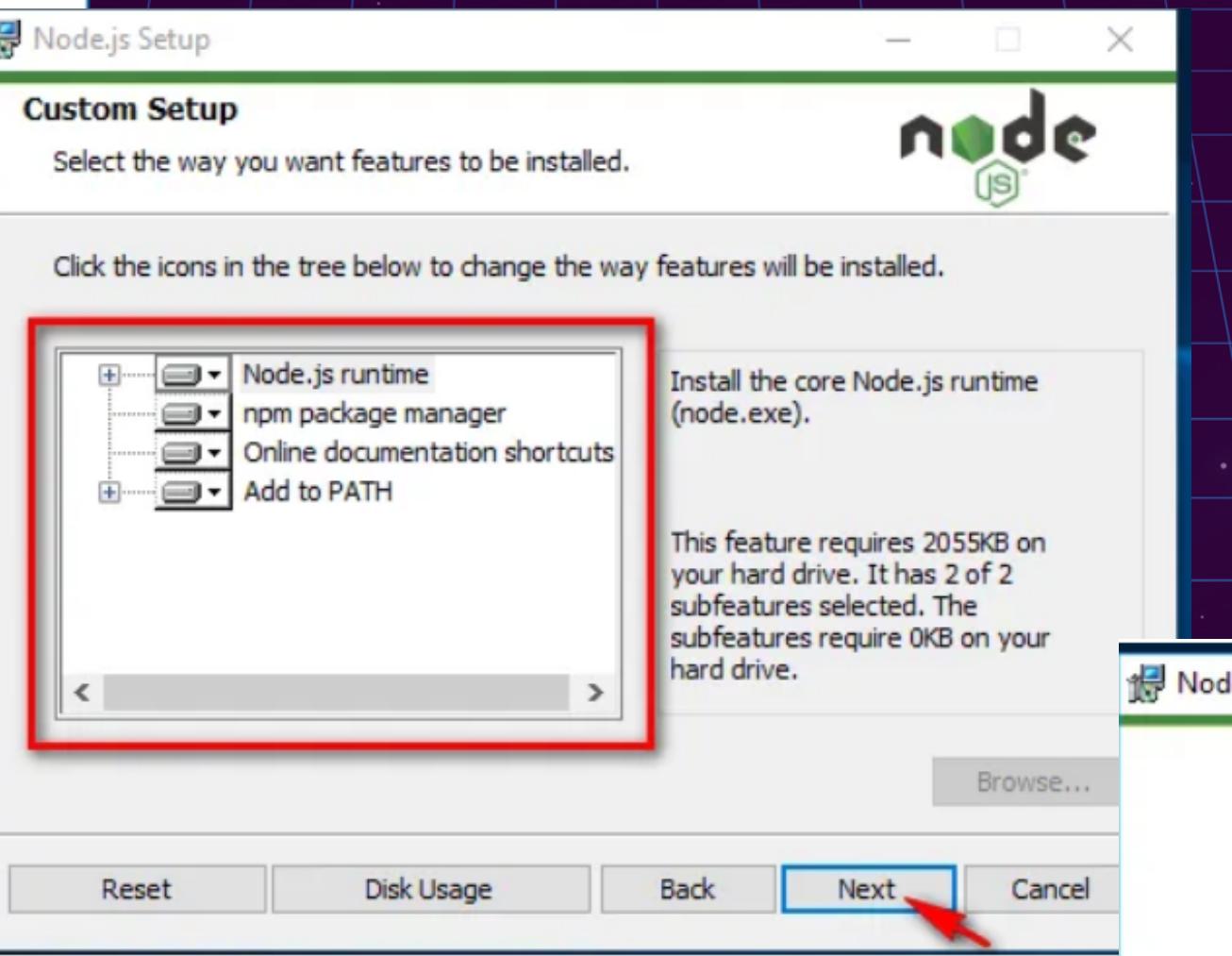
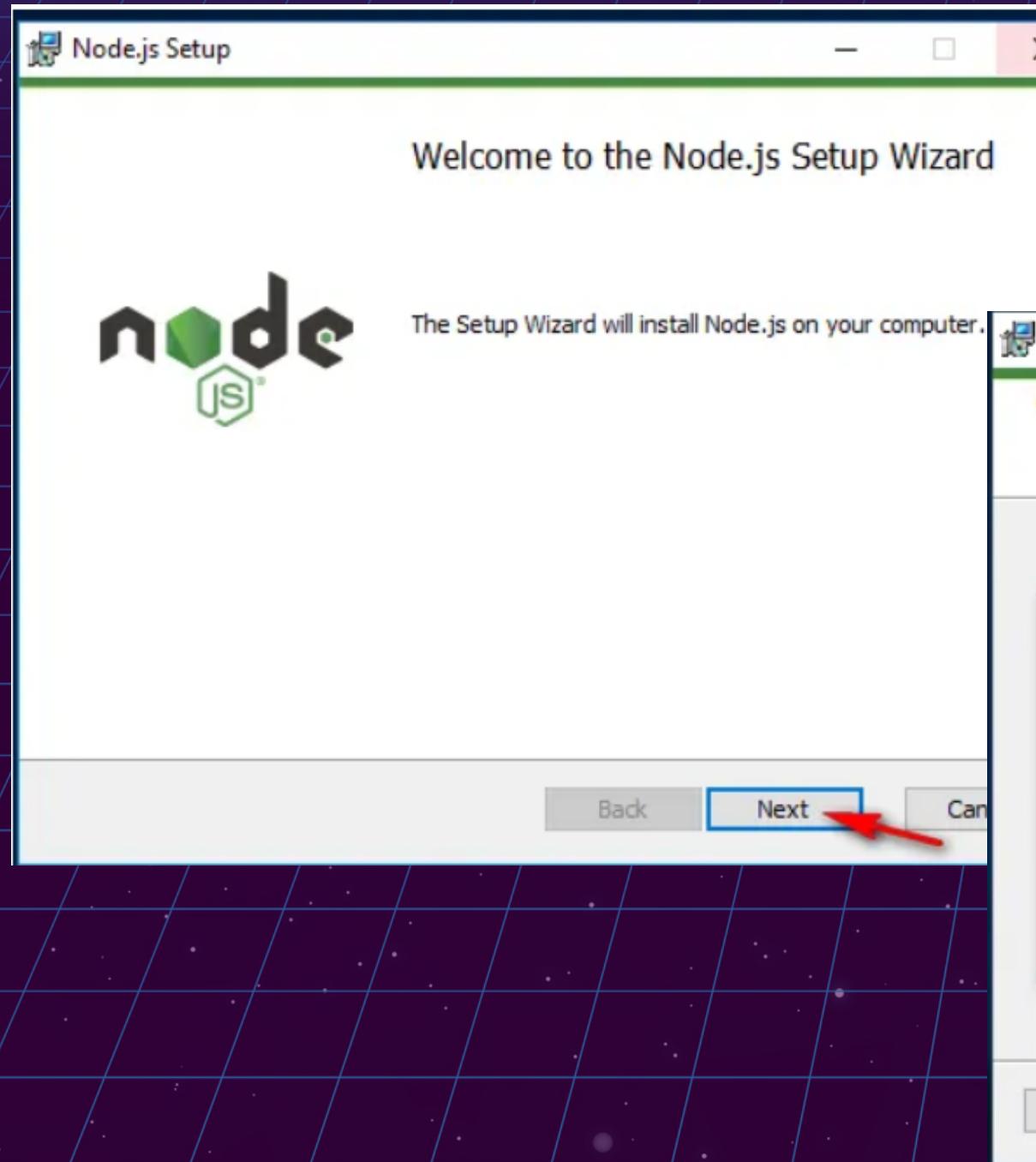
Node-RED



1. Install Node.JS (<https://nodejs.org/en/>)

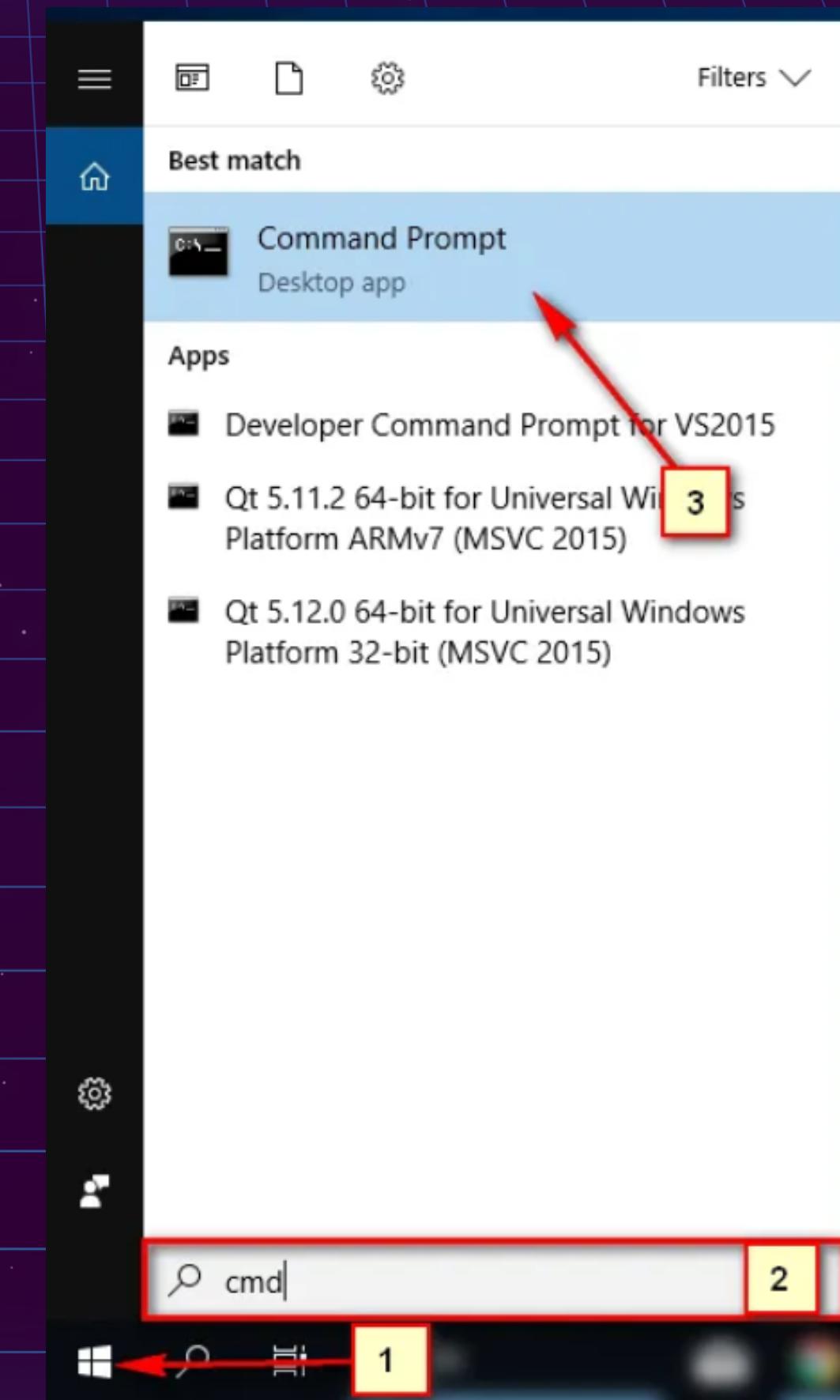
Go to the nodejs.org website and choose to download the LTS version.





2. Check Version Node.js with CMD

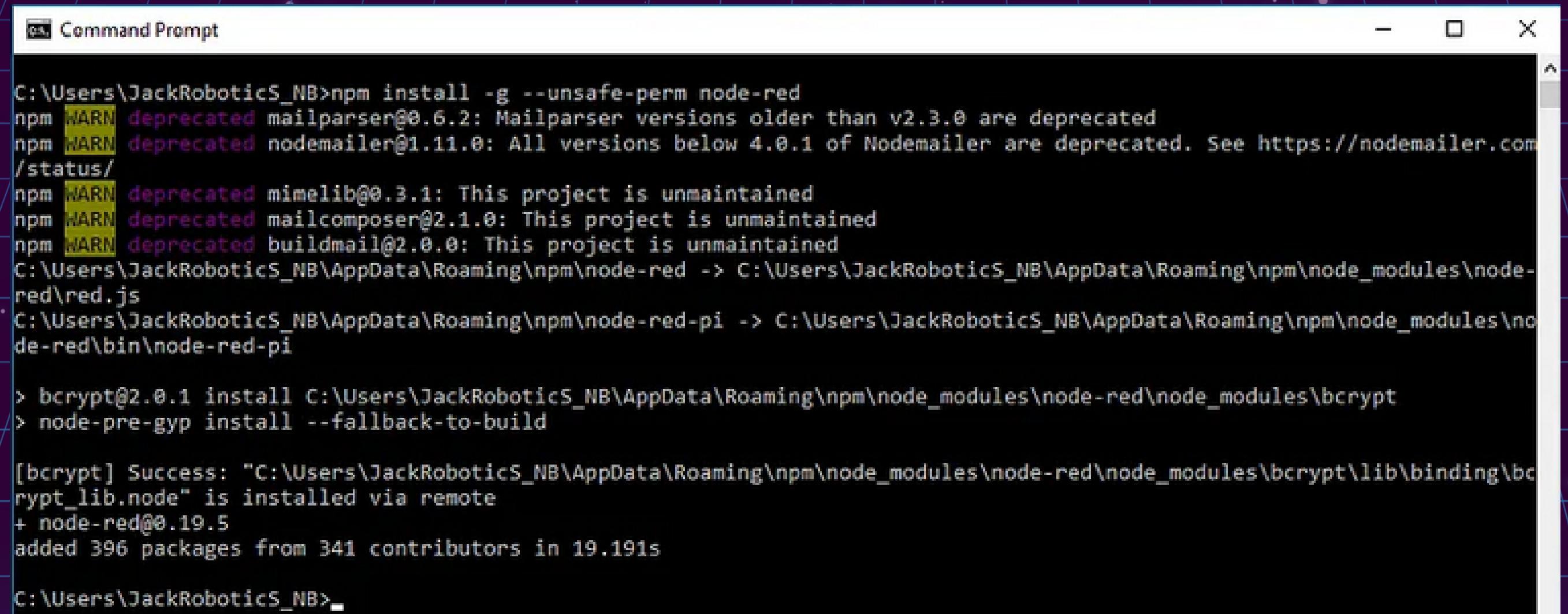
- #1. Press the start menu button
- #2.Type cmd to search
- #3.Select Command Prompt



3. Install Node-Red with npm

```
npm install -g --unsafe-perm node-red
```

When Node-Red is finished installing, it will show the screen as shown below. indicates that the installation is complete



The screenshot shows a Windows Command Prompt window titled "Command Prompt". The window displays the terminal output of the command "npm install -g --unsafe-perm node-red". The output includes several "WARN" messages about deprecated packages like mailparser, nodemailer, mimelib, mailcomposer, and buildmail. It also shows the creation of symbolic links for node-red and node-red-pi in the user's AppData\Roaming\npm directory. The final message indicates that 396 packages were added from 341 contributors in 19.191s.

```
C:\Users\JackRobotics_NB>npm install -g --unsafe-perm node-red
npm WARN deprecated mailparser@0.6.2: Mailparser versions older than v2.3.0 are deprecated
npm WARN deprecated nodemailer@1.11.0: All versions below 4.0.1 of Nodemailer are deprecated. See https://nodemailer.com/status/
npm WARN deprecated mimelib@0.3.1: This project is unmaintained
npm WARN deprecated mailcomposer@2.1.0: This project is unmaintained
npm WARN deprecated buildmail@2.0.0: This project is unmaintained
C:\Users\JackRobotics_NB\AppData\Roaming\npm\node-red -> C:\Users\JackRobotics_NB\AppData\Roaming\npm\node_modules\node-red\node-red.js
C:\Users\JackRobotics_NB\AppData\Roaming\npm\node-red-pi -> C:\Users\JackRobotics_NB\AppData\Roaming\npm\node_modules\node-red\bin\node-red-pi

> bcrypt@2.0.1 install C:\Users\JackRobotics_NB\AppData\Roaming\npm\node_modules\node-red\node_modules\bcrypt
> node-pre-gyp install --fallback-to-build

[bcrypt] Success: "C:\Users\JackRobotics_NB\AppData\Roaming\npm\node_modules\node-red\node_modules\bcrypt\lib\binding\bcrypt_lib.node" is installed via remote
+ node-red@0.19.5
added 396 packages from 341 contributors in 19.191s

C:\Users\JackRobotics_NB>
```

4. Run Node-Red

node-red

Type the command node-red
into cmd, it will look like the picture.

```
node-red

C:\Users\JackRobotics_NB>node-red
22 Jan 20:04:21 - [info]

Welcome to Node-RED
-----
22 Jan 20:04:21 - [info] Node-RED version: v0.19.5
22 Jan 20:04:21 - [info] Node.js version: v10.15.0
22 Jan 20:04:21 - [info] Windows_NT 10.0.17134 x64 LE
22 Jan 20:04:21 - [info] Loading palette nodes
22 Jan 20:04:22 - [warn] rpi-gpio : Raspberry Pi specific node set inactive
22 Jan 20:04:22 - [warn] -----
22 Jan 20:04:23 - [warn] [node-red/tail] Not currently supported on Windows.
22 Jan 20:04:23 - [warn] -----
22 Jan 20:04:23 - [info] Settings file : C:\Users\JackRobotics_NB\.node-red\settings.js
22 Jan 20:04:23 - [info] Context store : 'default' [module=memory]
22 Jan 20:04:23 - [info] User directory : C:\Users\JackRobotics_NB\.node-red
22 Jan 20:04:23 - [warn] Projects disabled : editorTheme.projects.enabled=false
22 Jan 20:04:23 - [info] Flows file : C:\Users\JackRobotics_NB\.node-red\flows_JackRobotics_NB.json
22 Jan 20:04:23 - [info] Creating new flow file
22 Jan 20:04:23 - [warn]

-----
Your flow credentials file is encrypted using a system-generated key.

If the system-generated key is lost for any reason, your credentials
file will not be recoverable, you will have to delete it and re-enter
your credentials.

You should set your own key using the 'credentialSecret' option in
your settings file. Node-RED will then re-encrypt your credentials
file using your chosen key the next time you deploy a change.

-----
22 Jan 20:04:23 - [info] Server now running at http://127.0.0.1:1880/
22 Jan 20:04:23 - [info] Starting flows
22 Jan 20:04:23 - [info] Started flows
-----
```

ESP32 เชื่อมต่อ wifi



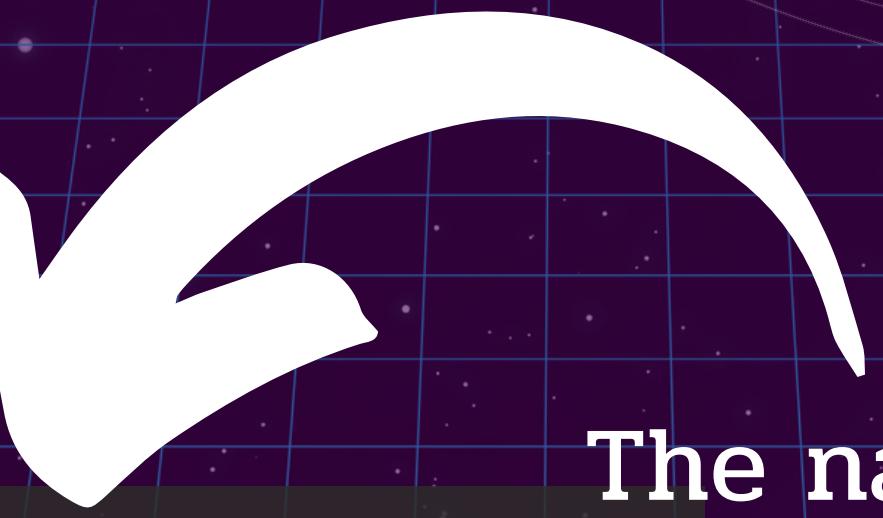
code example for wifi setup

library in use

#include <WiFi.h>

(Comes with esp32 board, no need to add more)

Config wifi and password



The name of the wifi to
(connect to).

```
const char* ssid = "ESL-Net";
```

```
const char* password = "ceeslonly";
```



wifi password to connect

wifi connection function

```
void setup_wifi() {  
    WiFi.mode(WIFI_STA); Choose a wifi connection mode.  
    WiFi.begin(ssid, password); Connect to wifi according to the name and password set.  
    while (WiFi.status() != WL_CONNECTED) {  
        delay(500); Wait until the wifi connects  
        Serial.println("Connecting to WiFi..."); successfully.  
    }  
    Serial.println("WiFi connected");  
}
```

call function **setup_wifi** active

void setup()

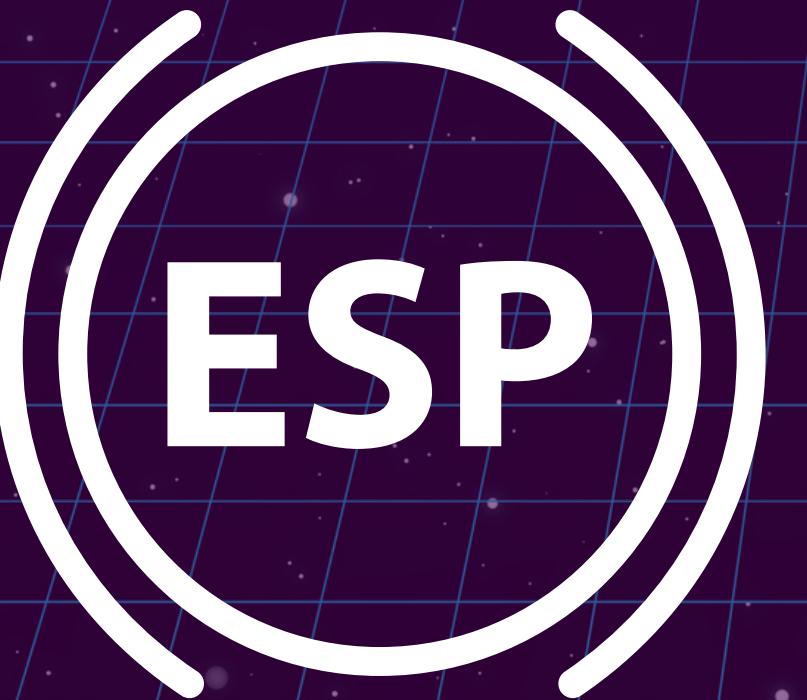
{

Serial.begin(115200);

setup_wifi();

}

ESP32 connect to Mqtt

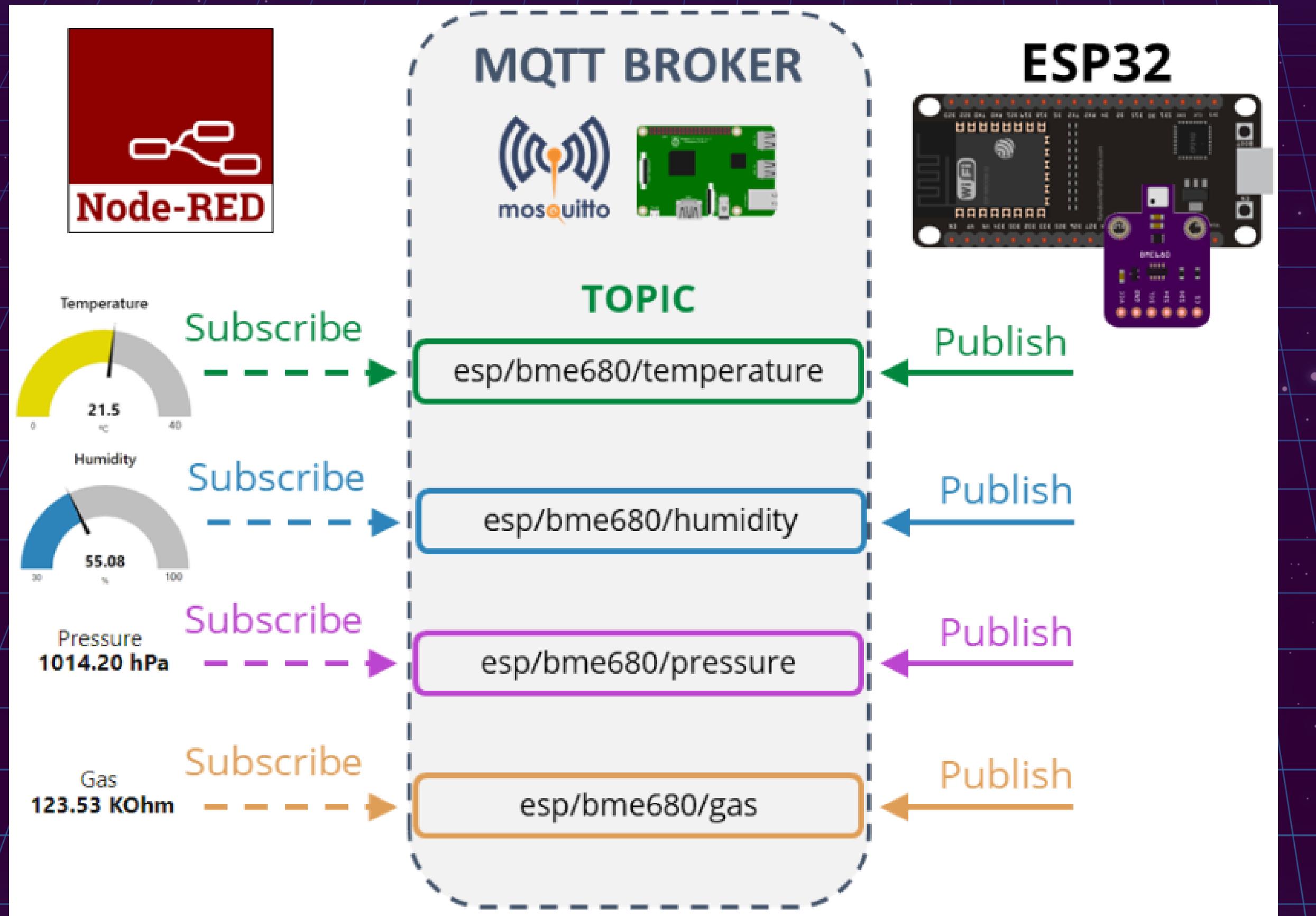


EMQX



code example for mqtt setup

MQTT



libraries in use

#include <PubSubClient.h>
download from github



P'Oat

IP mqtt : broker.emqx.io
port mqtt : 1883

Connect mqtt

```
const char* mqtt_broker = "broker.emqx.io";           //IP mqtt server
const char* mqtt_username = "mqtt_username";          //mqtt username as distributed
const char* mqtt_password = "mqtt_password";          //mqtt password as distributed
const int mqtt_port = 1883;                            //port mqtt server

WiFiClient espClient;
PubSubClient client(espClient);                       //Create an object for mqtt connection.
```

Function subscribe

receive value from mqtt

```
topic incoming | data incoming | length data incoming
void callback(char *topic, byte *payload, unsigned int length) {
    Serial.print("Message arrived in topic: ");
    Serial.println(topic);
    Serial.print("Message:");
    for (int i = 0; i < length; i++)      Loop along the length to print the incoming data.
        Serial.print((char) payload[i]);
    Serial.println();
    Serial.println("-----");
}
```

Function connect Mqtt

```
void reconnect() {
    client.setServer(mqtt_broker, mqtt_port);          //set mqtt server according to IP and port set
    client.setCallback(callback);                     //Select the subscribe function from page 17.
    while (!client.connected())                      //wait until Mqtt connect successfully
    {
        String client_id = "esp32-client-";
        client_id += String(WiFi.macAddress());
        Serial.printf("The client %s connects to the public mqtt broker\n", client_id.c_str());
        connect mqtt device name --> username      password --> return connection status
        if (client.connect(client_id.c_str(), mqtt_username, mqtt_password))
            Serial.println("Public emqx mqtt broker connected");
        else {
            Serial.print("failed with state ");
            Serial.print(client.state());
            delay(2000);
    }
}
```

call function to active void setup()

```
reconnect(); //connect mqtt  
client.subscribe("hero/user/ทำอะไร"); //define topic will be subscribe  
client.publish("hero/user/ทำอะไร", "hello"); //define topic will publish and value
```

call function to active in void loop()

```
void loop() {  
    client.loop();  
}
```



NODERED

THANK YOU
FOR YOUR ATTENTION

