IOT เบื้องต้น

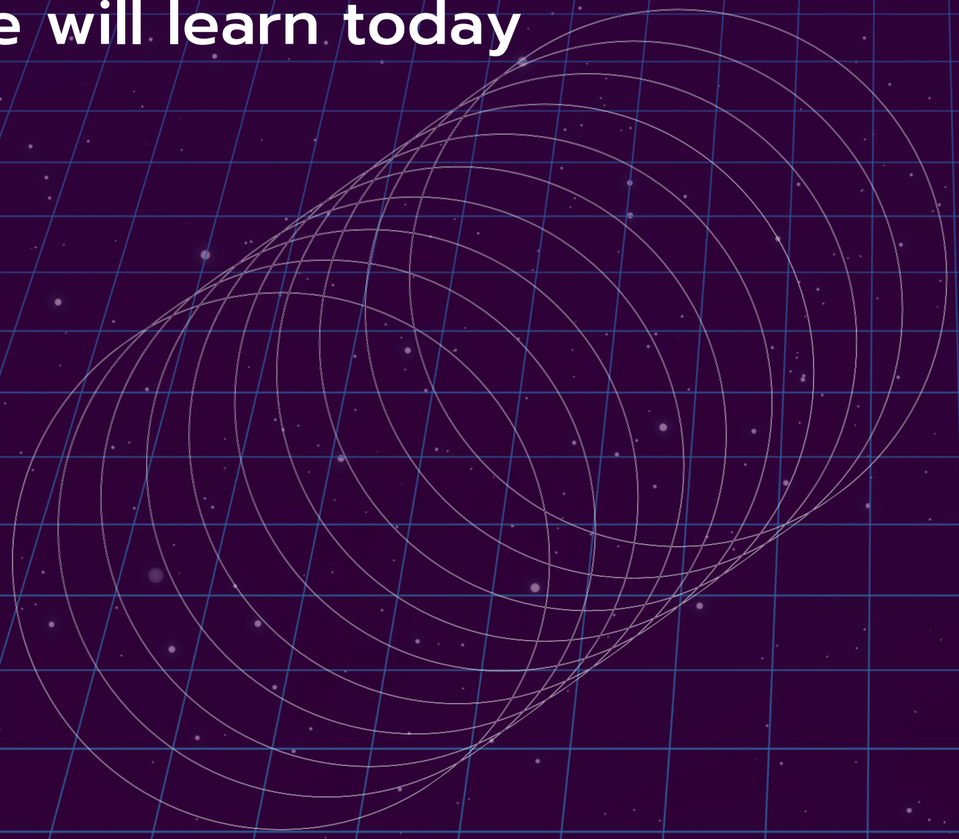
ด้วยบอร์ด ESP32

นำเสนอโดย

ESL

Topic

What we will learn today



01

Install Node-RED

02

connect wifi (ESP32)

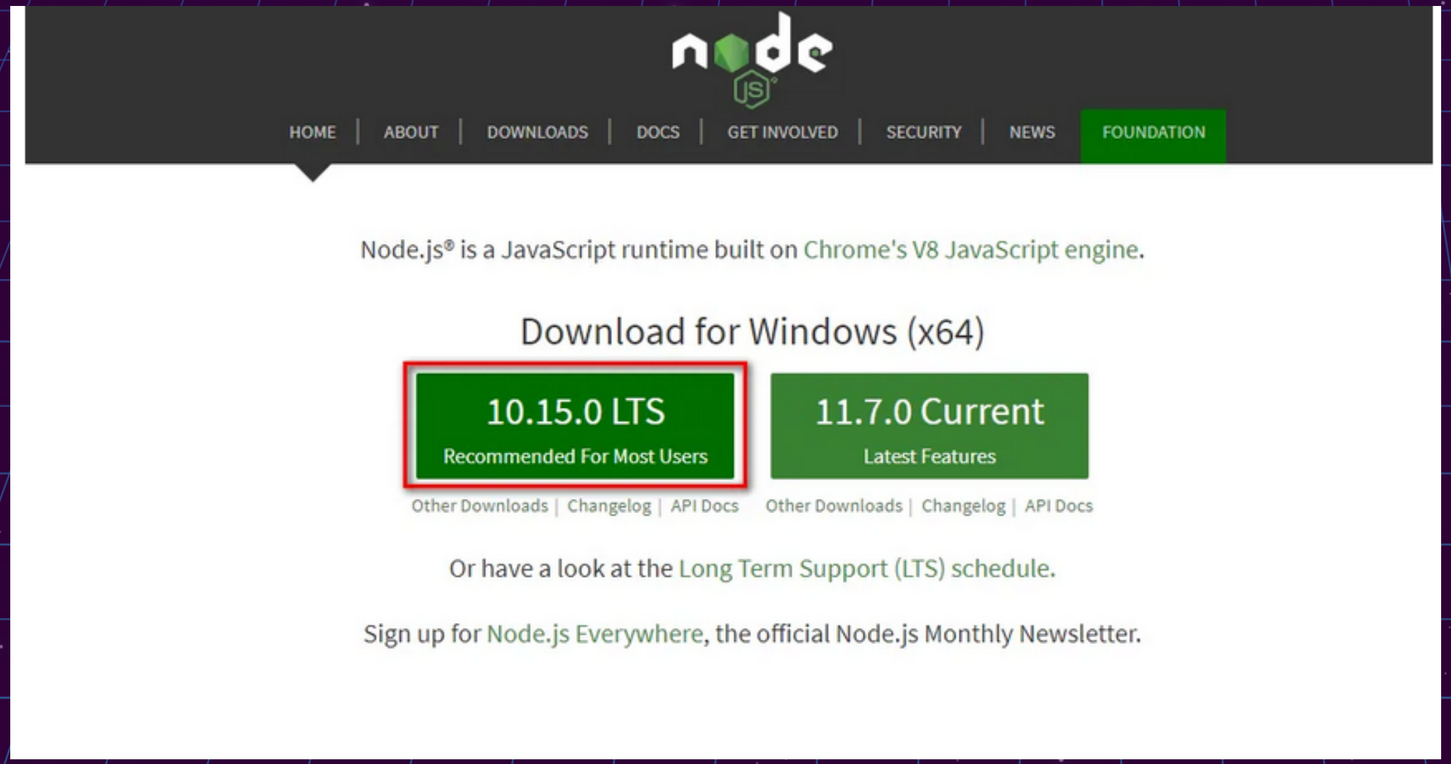
03

connect mqtt (ESP32)

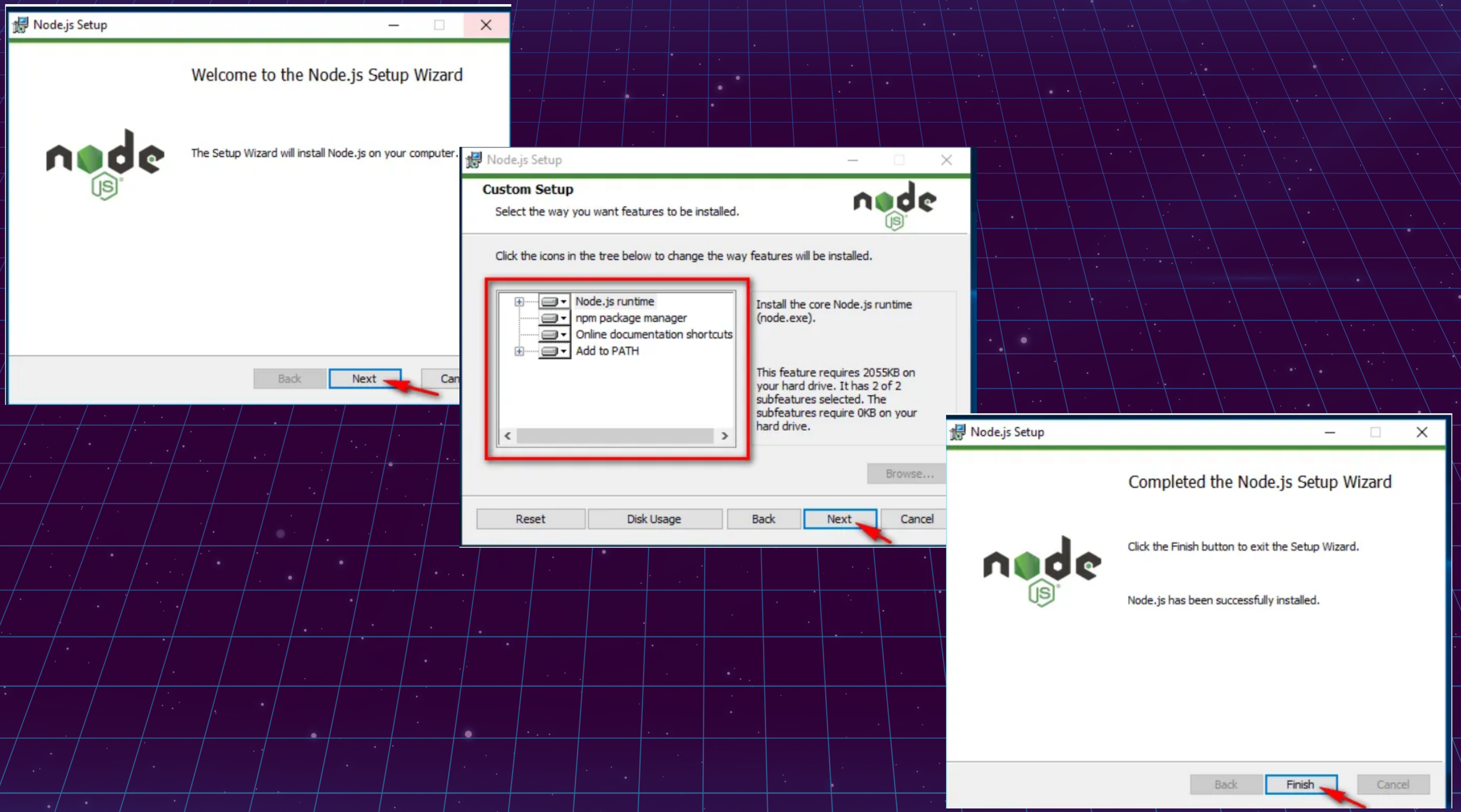
04

Node-RED

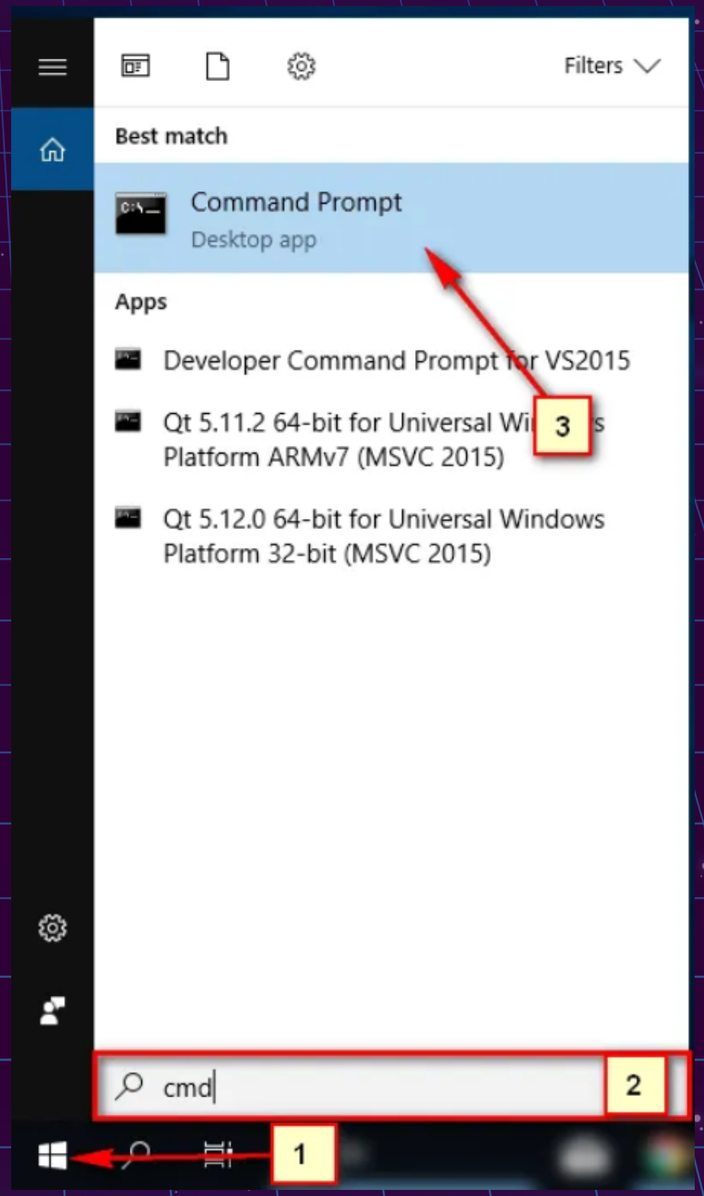
1. Install Node.JS (https://nodejs.org/en/)
Go to the nodejs.org website and choose to download the LTS version.



Node-RED


Node.js Setup
Welcome to the Node.js Setup Wizard
The Setup Wizard will install Node.js on your computer.
Back
Next
Node.js Setup
Custom Setup
Select the way you want features to be installed.
Click the icons in the tree below to change the way features will be installed.
Node.js runtime
npm package manager
Online documentation shortcuts
Add to PATH
Install the core Node.js runtime (node.exe).
This feature requires 2055KB on your hard drive. It has 2 of 2 subfeatures selected. The subfeatures require 0KB on your hard drive.
Browse...
Reset
Disk Usage
Back
Next
Cancel
Node.js Setup
Completed the Node.js Setup Wizard
Click the Finish button to exit the Setup Wizard.
Node.js has been successfully installed.
Back
Finish
Cancel

2. Check Version Node.JS with CMD



#1. Press the start menu button

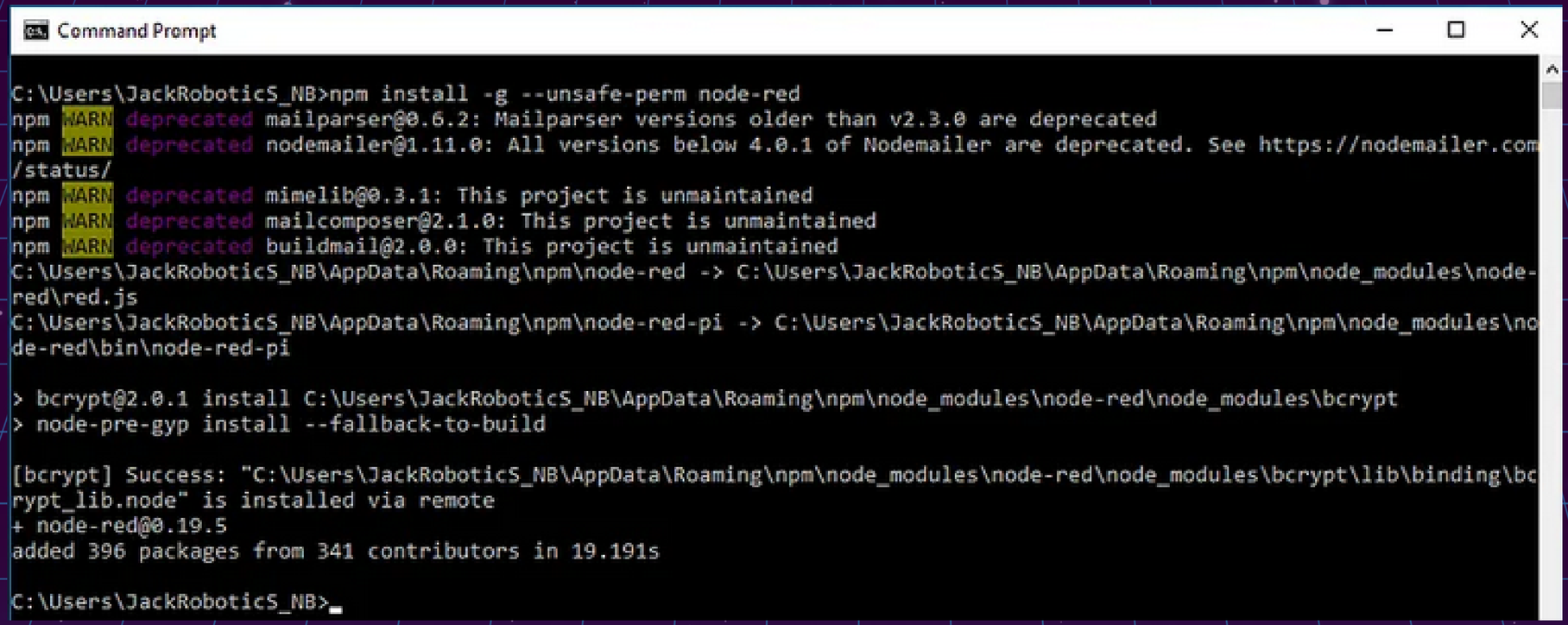
#2.Type cmd to search

#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

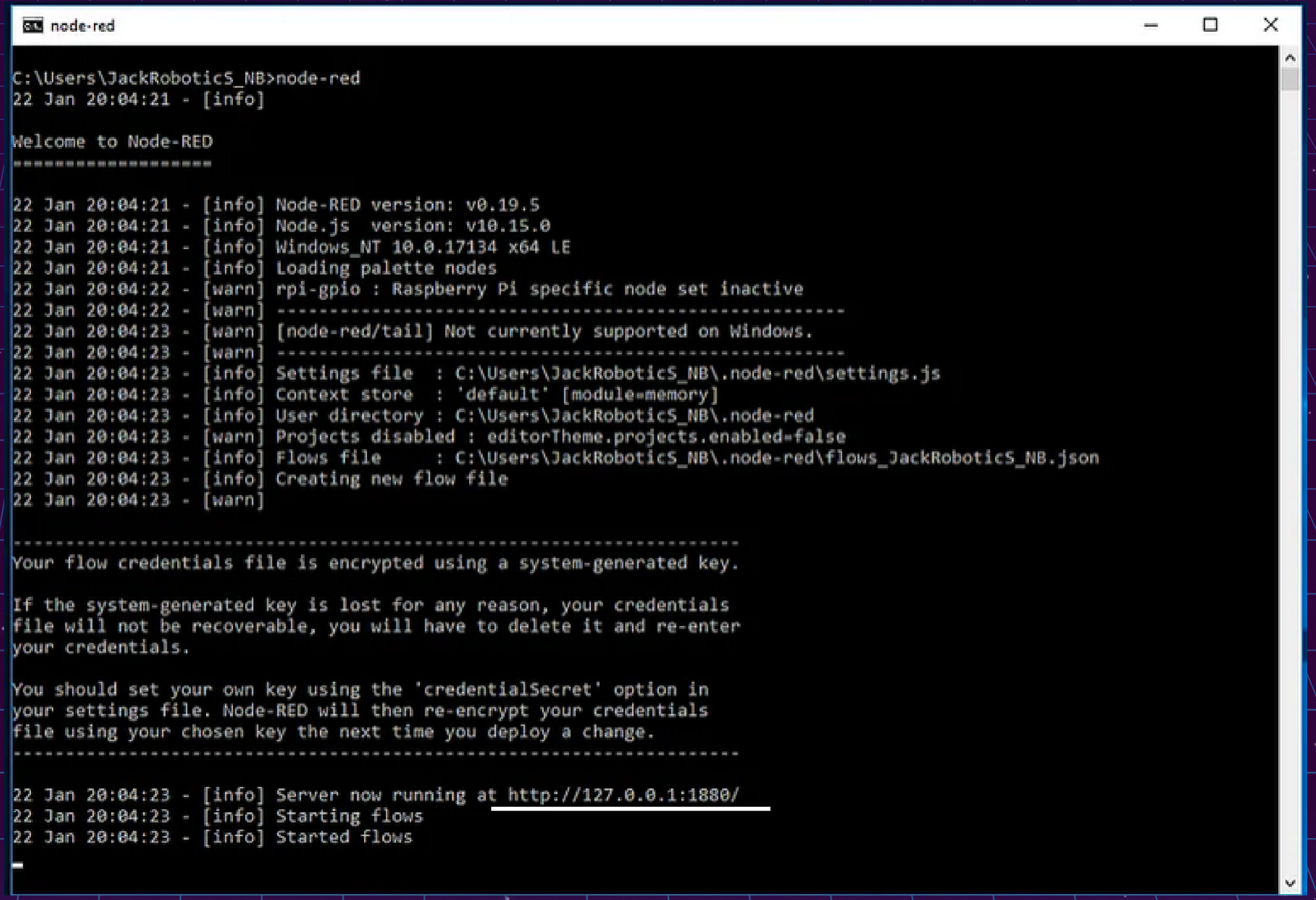


4. Run Node-Red

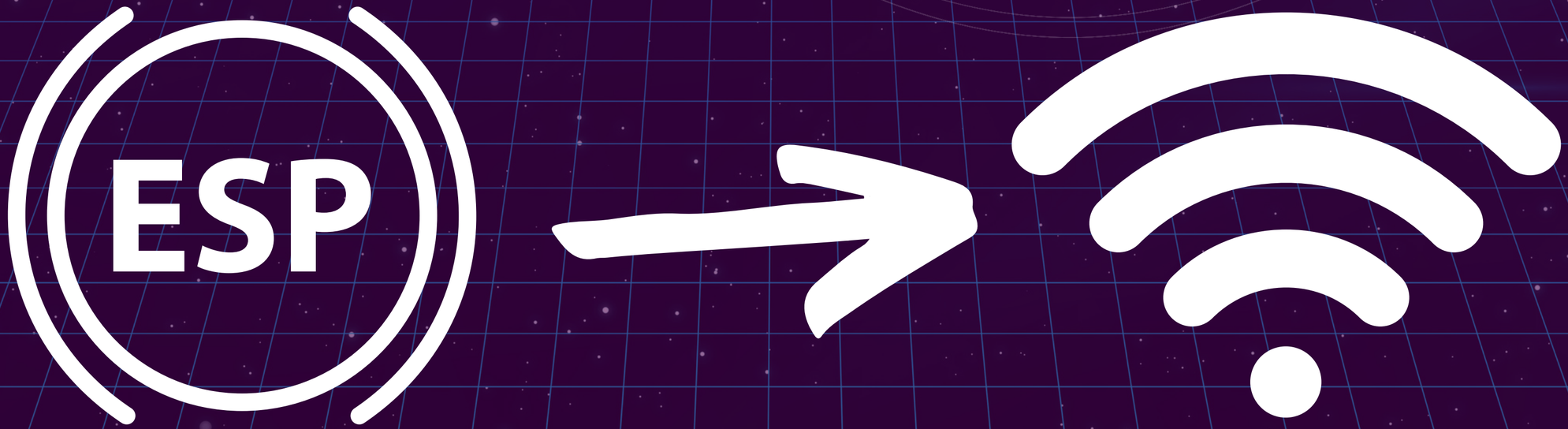
```
node-red
```

Type the command node-red

into cmd, it will look like the picture.



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);
    WiFi.begin(ssid, password);
    while (WiFi.status() != WL_CONNECTED) {
        delay(500);
        Serial.println("Connecting to WiFi...");
    }
    Serial.println("WiFi connected");
}
```

Choose a wifi connection mode.

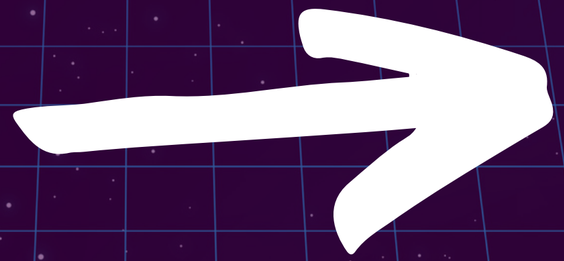
Connect to wifi according to the name and password set.

Wait until the wifi connects successfully.

call function setup_wifi active

```
void setup()
{
        Serial.begin(115200);
        setup_wifi();
}
```

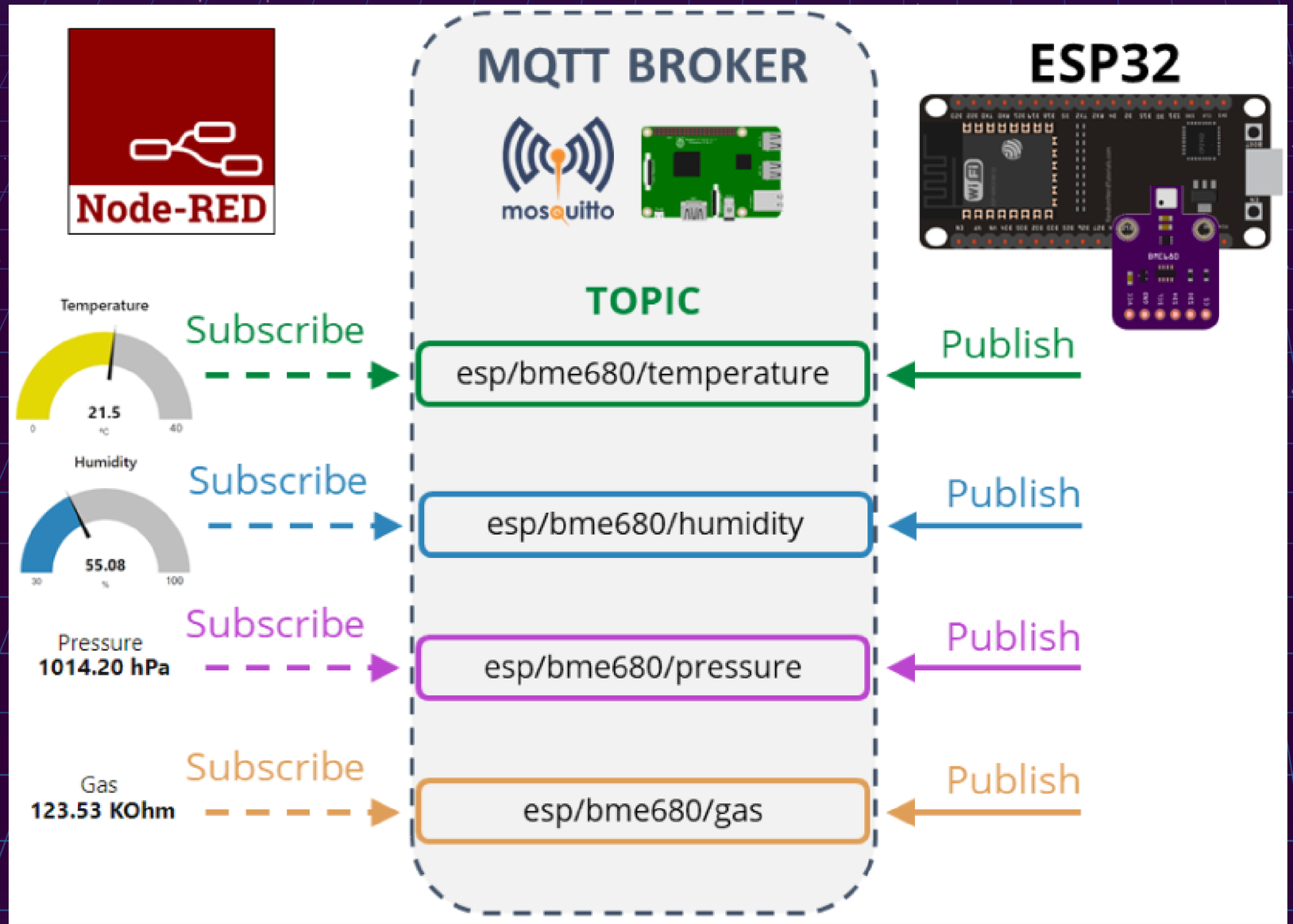
ESP32 connect to Mqtt



EMQX

code example for mqtt setup

MQTT


Node-RED
MQTT BROKER
mosquitto
ESP32
TOPIC
Temperature
21.5
0
°C
40
Subscribe
esp/bme680/temperature
Publish
Humidity
55.08
30
%
100
Subscribe
esp/bme680/humidity
Publish
Pressure
1014.20 hPa
Subscribe
esp/bme680/pressure
Publish
Gas
123.53 KOhm
Subscribe
esp/bme680/gas
Publish

libraries in use

#include <PubSubClient.h>
download from github

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

P'Oat

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 vaule 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++)
        Serial.print((char) payload[i]);
    Serial.println();
    Serial.println("-----------------------");
}
```

Loop along the length to print the incoming data.

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