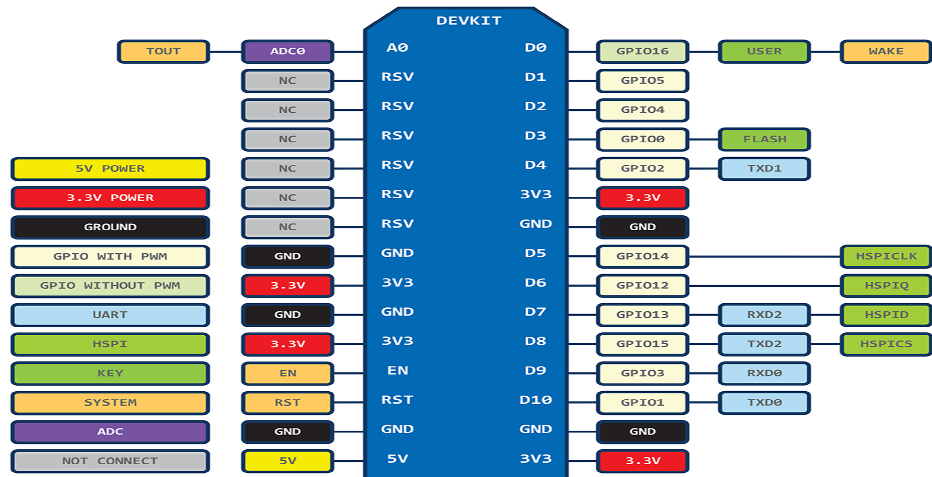


Part03a - ESP32 Connect to Network

1. เนื้อหา

NodeMCU - ESP8266



NodeMCU V0.9

ESP-12 Module

Arduino IDE = Node0.9



- USB-SERIAL CH340
- ใช้ Serial LED ที่ GPIO1 ได้ แต่ต้องไม่ใช้พร้อม Serial Communication
- มี LED Buid in ที่ GPIO16“BUILTIN_LED”

NodeMCU V1.0

ESP-12E Module

Arduino IDE = Node1.0



- Silicon Labs CP210x USB to UART Bridge
- ใช้ Serial LED ที่ GPIO2 ได้ แต่ต้องไม่ใช้พร้อม Serial Communication
- มี LED Buid in ที่ GPIO16“BUILTIN_LED”

NodeMCU V3.0

ESP-12E Module

Arduino IDE = Node1.0

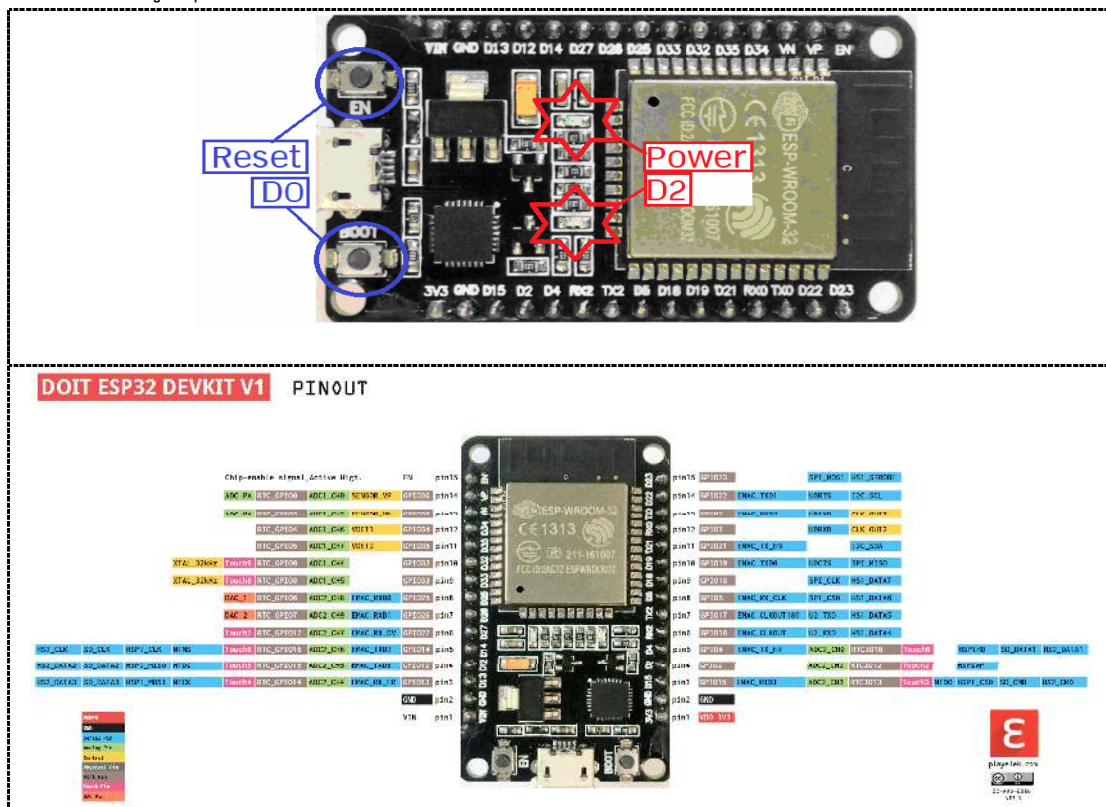


- USB-SERIAL CH340
- ใช้ Serial LED ที่ GPIO2 ได้ แต่ต้องไม่ใช้พร้อม Serial Communication

ESP-32 Dev Kit V1 Board

ก่อนหน้านี้มีการใช้งาน NodeMCU V2 ซึ่งเป็น ESP8266 อย่างแพร่หลาย แต่ด้วยเสียงลือเสียงเล่าอ้างเรื่องความสามารถของ ESP32 ที่พัฒนาความสามารถเพิ่มมาแก้จุดด้อยของ ESP8266 ทั้งรองรับการเชื่อมต่อแบบ Hybrid ทั้ง WiFi และ Bluetooth มีพอร์ตรองรับ I/O ได้เพิ่มขึ้น รองรับ touch sensor มี hardware เข้ารหัสสำหรับ HTTPS และอีกมากมาย ด้วยเหตุผลที่ว่าไปแล้วและราคาที่ไม่แพง รอบนี้เลยได้ ESP32 Development Board ที่มีชื่อเต็มคือ DOIT ESP32 DevKit V1 ใช้โมดูล ESP-WROOM-32 มาทำการทดสอบ

รายละเอียดเพิ่มเติมของ DOIT ESP32 DevKit V1 ลองเข้าไปดูใน SmartArduino (<https://github.com/SmartArduino/SZDOITWiki/wiki/ESP8266---ESP32>) หน้าตาคล้าย ESP32 DevKit C V2 ของ Espressif และ Development Board ตระกูลเดียวกันกับเจ้าอื่นๆ เลย มีเจาะรูสี่มุมมาด้วย แต่ pinout ไม่เหมือนกัน



DOIT ESP32 DEVKIT V1 PINOUT

Chip-enable signal, Active High.									
				EN		pin15			
ADC_PA	RTC_GPIO0	ADC1_CH0	SENSOR_VP	GPIO36		pin14			
ADC_PA	RTC_GPIO3	ADC1_CH3	SENSOR_VN	GPIO39		pin13			
	RTC_GPIO4	ADC1_CH6	VDET1	GPIO34		pin12			
	RTC_GPIO5	ADC1_CH7	VDET2	GPIO35		pin11			
XTAL_32kHz	Touch9	RTC_GPIO9	ADC1_CH4	GPIO32		pin10			
XTAL_32kHz	Touch8	RTC_GPIO8	ADC1_CH5	GPIO33		pin9			
DAC_1	RTC_GPIO6	ADC2_CH8	EMAC_RXD0	GPIO25		pin8			
DAC_2	RTC_GPIO7	ADC2_CH9	EMAC_RXD1	GPIO26		pin7			
	Touch7	RTC_GPIO17	ADC2_CH7	EMAC_RX_DV	GPIO27	pin6			
HS2_CLK	SD_CLK	HSPI_CLK	MTMS	Touch6	RTC_GPIO16	ADC2_CH6	EMAC_TXD2	GPIO14	pin5
HS2_DATA2	SD_DATA2	HSPI_MISO	MTDI	Touch5	RTC_GPIO15	ADC2_CH5	EMAC_TXD3	GPIO12	pin4
HS2_DATA3	SD_DATA3	HSPI_MOSI	MTCK	Touch4	RTC_GPIO14	ADC2_CH4	EMAC_RX_ER	GPIO13	pin3
							GND		pin2
							VIN		pin1

POWER
GND
Serial Pin
Analog Pin
Control
Physical Pin
Port Pin
Touch Pin
DAC Pin



pin15	GPIO23	SPI_MOSI	HS1_STROBE			
pin14	GPIO22	EMAC_TXD1	U0RTS	I2C_SCL		
pin13	GPIO1	EMAC_RXD2	U0TXD	CLK_OUT3		
pin12	GPIO3	U0RXD	CLK_OUT2			
pin11	GPIO21	EMAC_TX_EN	I2C_SDA			
pin10	GPIO19	EMAC_TXD0	U0CTS	SPI_MISO		
pin9	GPIO18	SPI_CLK	HS1_DATA7			
pin8	GPIO5	EMAC_RX_CLK	SPI_CS0	HS1_DATA6		
pin7	GPIO17	EMAC_CLKOUT180	U2_TXD	HS1_DATA5		
pin6	GPIO16	EMAC_CLKOUT	U2_RXD	HS1_DATA4		
pin5	GPIO4	EMAC_TX_ER	ADC2_CH0	RTCIO10	Touch0	HSPIHD SD_DATA1 HS2_DATA1
pin4	GPIO2	ADC2_CH2	RTCIO12	Touch2		HSPIWP
pin3	GPIO15	EMAC_RXD3	ADC2_CH3	RTCIO13	Touch3	MTD0 HSPI_CS0 SD_CMD HS2_CMD
pin2	GND					
pin1	VDD_3V3					

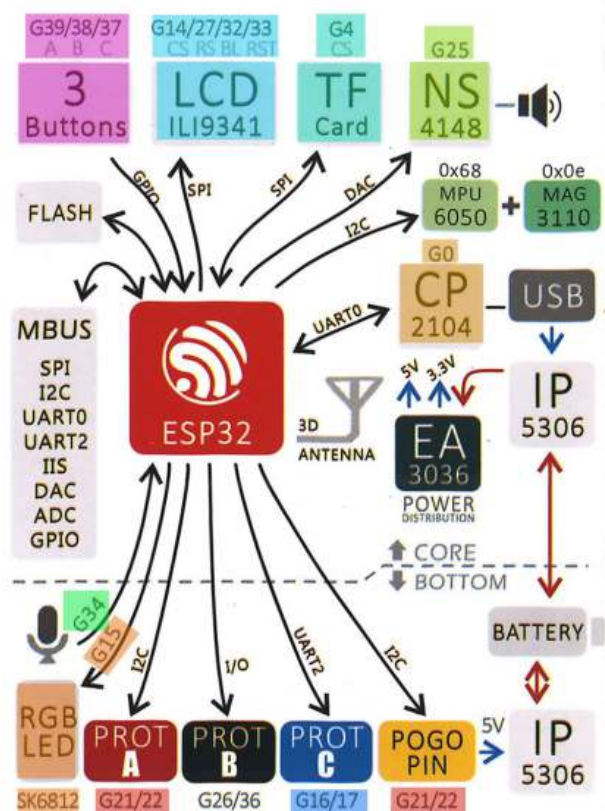
M5-Stack



M BUS					
GND	ADC	G35			
GND	ADC	G36			
GND	RST	EN			
G23	MOSI	DAC/SPK	G25		
G19	MISO	DAC	G26		
G18	SCK	3.3V			
G3	RXD0	TXD0	G1		
G16	RXD2	TXD2	G17		
G21	SDA	SCL	G22		
G2	GPIO	GPIO	G5		
G12	IIS_SK	IIS_WS	G13		
G15	IISOUT	IIS_MK	G0		
HPWR		IIS_IN	G34		
HPWR		5V			
HPWR		BATTERY			

GROVE PORT		CABLE		
A	I2C	GND	5V	SDA SCL
B	I/O	GND	5V	G26 G36 In/Out Input
C	UART	GND	5V	TXD2RXD2

Schematic



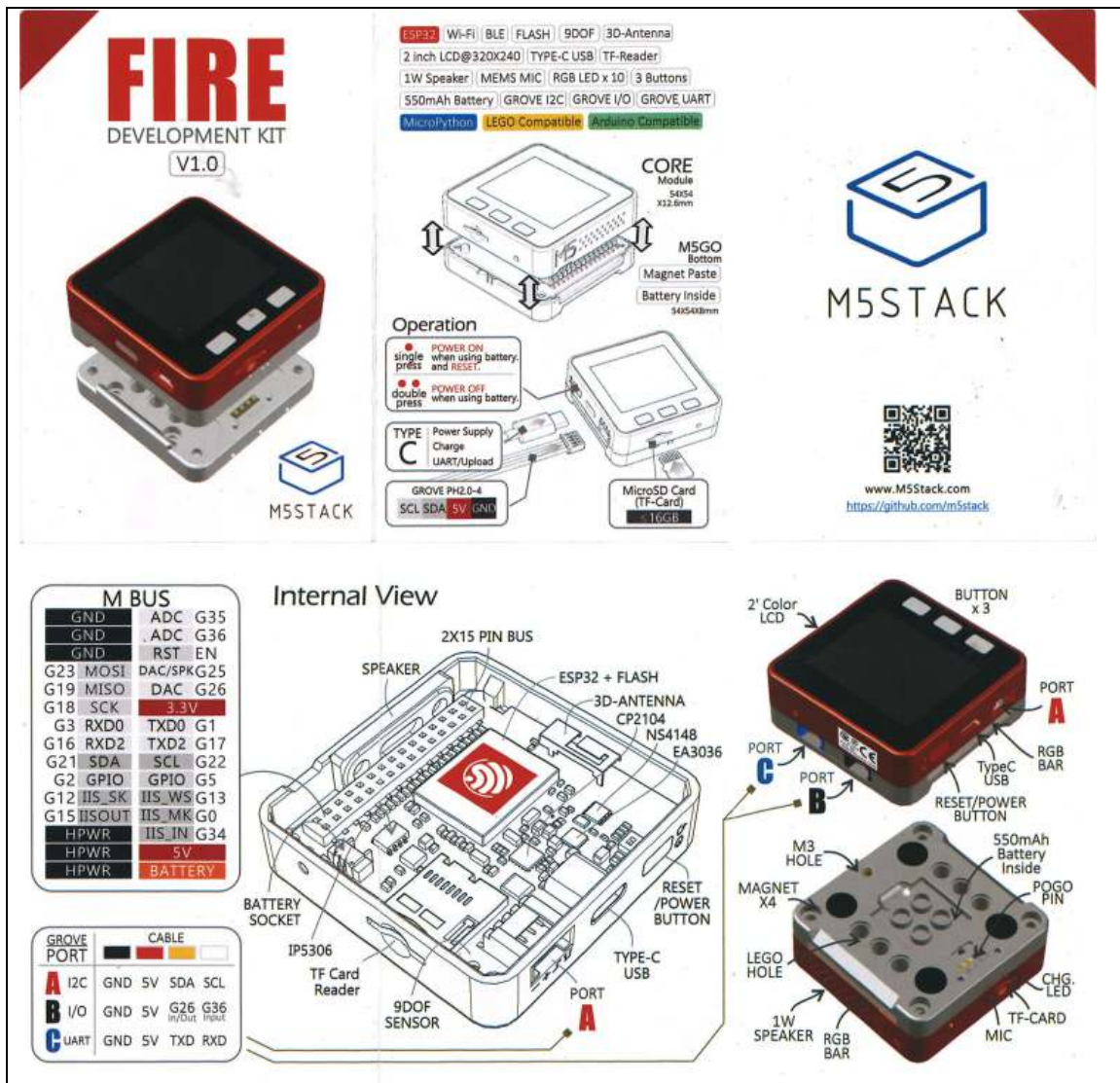
5Stack ESP32 คอมพิวเตอร์จิ๋วที่ใช้สร้างงานต้นแบบที่มาพร้อมกับหน้าจอ 2 นิ้ว ความละเอียด 320x240 pixel และมีหน้ากากให้เปลี่ยนเป็นพิมพ์ได้ 3 แบบ ตามการใช้งานที่ออกแบบ Keyboard panel, Gameboy panel และ Number Panel พร้อมกับฐานชาร์ตและแบตเตอรี่ LiPo ขนาด 650mAh

M5Stack คืออุปกรณ์ที่ใช้สร้างตัวต้นแบบของอุปกรณ์ WiFi+Bluetooth โดยใช้ชิป ESP32 ของ Espressif โปรแกรมได้ทั้ง Arduino, Micro-Python หรือ Web-IDE ตัวบอร์ด

หลัก M5 Core จะอยู่ด้านบนติดกับจอ LCD จะมีเสาอากาศแบบ 3D มาให้ (3D Antenna), Grove connector สำหรับ I2C, microSD card socket, JST battery socket, สวิตช์ เปิด/ปิด/reset, 3ปุ่มกดบนหน้าจอ

บอร์ดด้านหลังจะเป็น M5 Faces ซึ่งทำหน้าที่ต่อเข้ากับแป้นพิมพ์ Panel ต่างๆ และยังมีแบตเตอรี่ LiPo ขนาด 650mAh ที่สามารถชาร์จผ่าน charging Base ได้

- <https://github.com/m5stack/M5Stack>
- <http://forum.m5stack.com/topic/360/m5stack-fire-pinout-leaflet>



2. อ่านเพิ่มเติม

- NodeMCU GitHub: [https:// github.com/nodemcu](https://github.com/nodemcu)
- NodeMCU Driver: [https://
www.silabs.com/products/mcu/Pages/USBtoUARTBridgeVCPDrivers.aspx](https://www.silabs.com/products/mcu/Pages/USBtoUARTBridgeVCPDrivers.aspx)
- ThaiEasyElect: [http:// www.thaieasyelec.com/products/internet-of-things/nodemcu-development-kit-v2-detail.html?gclid=CjoKEQjwl-e4BRCwqeWkv8TWqOoBEiQAMocbPytjm40atWOSYlaQI7V00Op-7asSWryeJ9tCQJNxnpoaAk2-8P8HAQ](http://www.thaieasyelec.com/products/internet-of-things/nodemcu-development-kit-v2-detail.html?gclid=CjoKEQjwl-e4BRCwqeWkv8TWqOoBEiQAMocbPytjm40atWOSYlaQI7V00Op-7asSWryeJ9tCQJNxnpoaAk2-8P8HAQ)
- AiyaraFun: [http:// www.ayarafun.com/2015/08/introduction-arduino-esp8266-nodemcu/](http://www.ayarafun.com/2015/08/introduction-arduino-esp8266-nodemcu/)
- Firmware Build and Example: [http:// nodemcu-build.com/](http://nodemcu-build.com/)
- Read This <https://playelek.com/doit-esp32-devkit-v1/>
- Read This <http://esp32.net/>
- Read This <https://www.arduitronics.com/product/1329/doit-esp32-development-board-esp-wroom-32-wifiblueetooth-esp-32s>
- Read This <https://www.mcucity.com/product/1144/doit-esp32-wifiblueetooth-ultra-low-power-consumption-dual-core-esp-32-esp-32s-esp-32-similar-esp8266>

3. การทดลอง – Digital Input/Output, Analog Input and Serial Input/Output

การทดลอง 1: Start with Arduino IDE in Hello World

1. Install Arduino IDE and Add ESP32 Board
2. Test Ex00_Blink

```
// ESP-32

int nloop = 24;
char DispBuff[] = {1, 1, 0, 0, 1, 1, 1, 0, 1, 1, 0, 0, 1, 1, 0, 0, 1, 0, 0, 0, 0, 0, 0};

void setup() {
  pinMode(LED_BUILTIN, OUTPUT);
}

void loop() {
  for (int i = 0; i < nloop; i++)
  { digitalWrite(LED_BUILTIN, DispBuff[i]);      delay(120);
    digitalWrite(LED_BUILTIN, LOW);              delay(120);
  }
}
```

```
#include <M5Stack.h>
int nloop = 24;
char DispBuff[] = {1, 1, 0, 0, 1, 1, 1, 0, 1, 1, 0, 0, 1, 1, 0, 0, 1, 0, 0, 0, 0, 0, 0};

void setup() {
  M5.begin();
  M5.Lcd.printf("M5Stack Speaker");
}

void loop() {
  for (int i = 0; i < nloop; i++)
  { if (DispBuff[i] == 1)
    M5.Speaker.beep(); //beep;
    else
    M5.Speaker.mute(); //no beep;
    delay(100);
    M5.Speaker.mute(); //no beep;
    delay(100);
  }
}
```

การทดลอง 2: My MAC Address

3. โหลดโปรแกรม My MAC = _____

```
uint64_t chipid;

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

void loop() {
  chipid = ESP.getEfuseMac(); //The chip ID is essentially its MAC address(length: 6 bytes).
  Serial.printf("ESP32 Chip ID = ");
  Serial.printf("%02X:", (uint8_t)(chipid >> 0)); //print 1 bytes
  Serial.printf("%02X:", (uint8_t)(chipid >> 8));- //print 1 bytes.
  Serial.printf("%02X:", (uint8_t)(chipid >> 16)); //print 1 bytes.
  Serial.printf("%02X:", (uint8_t)(chipid >> 24)); //print 1 bytes.
  Serial.printf("%02X:", (uint8_t)(chipid >> 32)); //print 1 bytes.
  Serial.printf("%02X:", (uint8_t)(chipid >> 40)); //print 1 bytes.
  Serial.println();
  delay(3000);
}
}
```

4. ทดสอบ File → Example → WiFi → WifiScan หรือโปรแกรมต่อไปนี้

```
#include "WiFi.h"

void setup()
{
  Serial.begin(115200);

  // Set WiFi to station mode and disconnect from an AP if it was previously connected
  WiFi.mode(WIFI_STA);
  WiFi.disconnect();
  delay(100);
  Serial.println("Setup done");
}

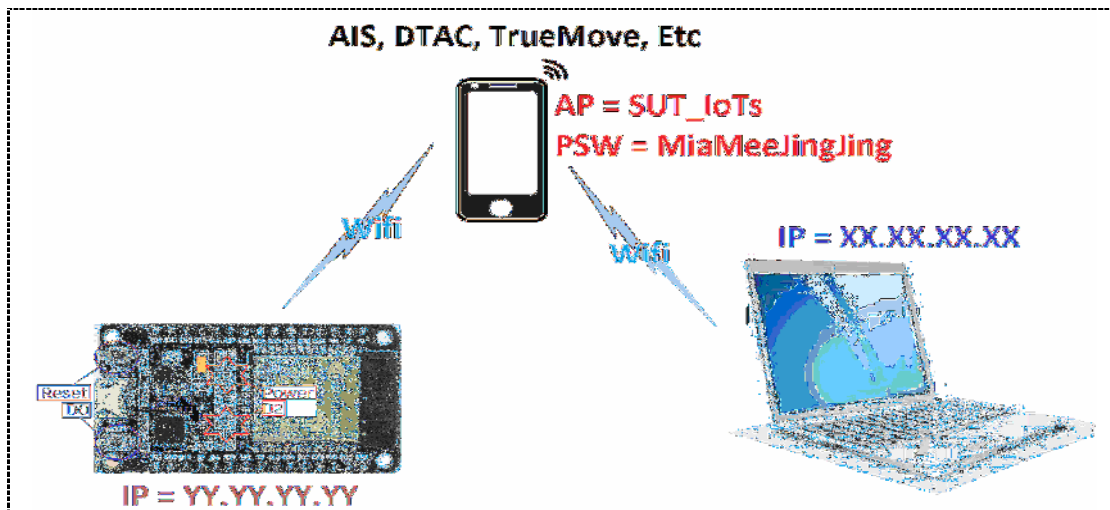
void loop()
{
  Serial.println("scan start");

  // WiFi.scanNetworks will return the number of networks found
  int n = WiFi.scanNetworks();
  Serial.println("scan done");
  if (n == 0) {
    Serial.println("no networks found");
  } else {
    Serial.print(n);
    Serial.println(" networks found");
    for (int i = 0; i < n; ++i) {
      // Print SSID and RSSI for each network found
      Serial.print(i + 1);
      Serial.print(": ");
      Serial.print(WiFi.SSID(i));
      Serial.print(" (");
      Serial.print(WiFi.RSSI(i));
      Serial.print(")");
      Serial.println((WiFi.encryptionType(i) == WIFI_AUTH_OPEN)? " ":"*");
      delay(10);
    }
  }
  Serial.println("");

  // Wait a bit before scanning again
  delay(5000);
}
}
```


การทดลอง 3: Connect to Network

5. โหลดโปรแกรมและ อย่าลืมแก้เป็นชื่อ SSID และ Password เป็นของตัวเอง



6. ทดสอบการเชื่อมต่อด้วยคำสั่ง Ping ในหน้าต่าง command

```
#include <WiFi.h>
const char* ssid = "SUT_IoTs";
const char* password = "MiaMeeJingJing";
void setup() {
  Serial.begin(115200);
  delay(10);
  Serial.println(); Serial.println();
  Serial.print("Connecting to ");
  Serial.println(ssid);
  WiFi.begin(ssid, password);
  while (WiFi.status() != WL_CONNECTED)
  { delay(500);
    Serial.print(".");
  }
  Serial.println("");
  Serial.println("WiFi connected");
  Serial.println("IP address: ");
  Serial.println(WiFi.localIP());
}

void loop() {}
```

COM3

```
load:0x3f10018, len:4
load:0x3fff001c, len:808
load:0x40078000, len:6084
load:0x40080000, len:6696
entry 0x400802e4

Connecting to SUT_IoT
..
WiFi connected
IP address:
192.168.43.87
```

Select Command Prompt

```
Microsoft Windows [Version 10.0.16299.547]
(c) 2017 Microsoft Corporation. All rights reserved.

C:\Users\Pk007_Bit32>ping 192.168.43.87

Pinging 192.168.43.87 with 32 bytes of data:
Reply from 192.168.43.87: bytes=32 time=780ms TTL=255
Reply from 192.168.43.87: bytes=32 time=377ms TTL=255
Reply from 192.168.43.87: bytes=32 time=386ms TTL=255
Reply from 192.168.43.87: bytes=32 time=395ms TTL=255

Ping statistics for 192.168.43.87:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 377ms, Maximum = 780ms, Average = 484ms

C:\Users\Pk007_Bit32>
```

การทดลอง 4: Web Server-Command

7. Test Ex20_WebServer_Cmd

```
#include <WiFi.h>

const char* ssid = "SUT_IoTs";
const char* password = "MaiMeeJingJing";
int pinTest = 2;

WiFiServer server(80);

void setup() {
  Serial.begin(115200);
  pinMode(pinTest, OUTPUT);          // set the LED pin mode
  delay(10);
  Serial.print("\n\nConnecting to ");
  Serial.println(ssid);
  WiFi.begin(ssid, password);

  while (WiFi.status() != WL_CONNECTED) {
    delay(500);
    Serial.print(".");
  }

  Serial.println("");
  Serial.println("WiFi connected.");
  Serial.println("IP address: ");
  Serial.println(WiFi.localIP());

  server.begin();
}

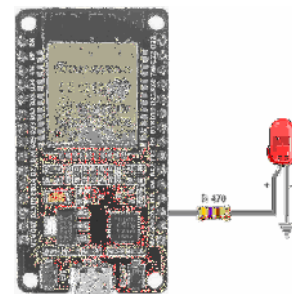
int value = 0;
bool LED_Status = LOW;

void loop() {
  digitalWrite(pinTest, LED_Status);
  WiFiClient client = server.available();   // listen for incoming clients

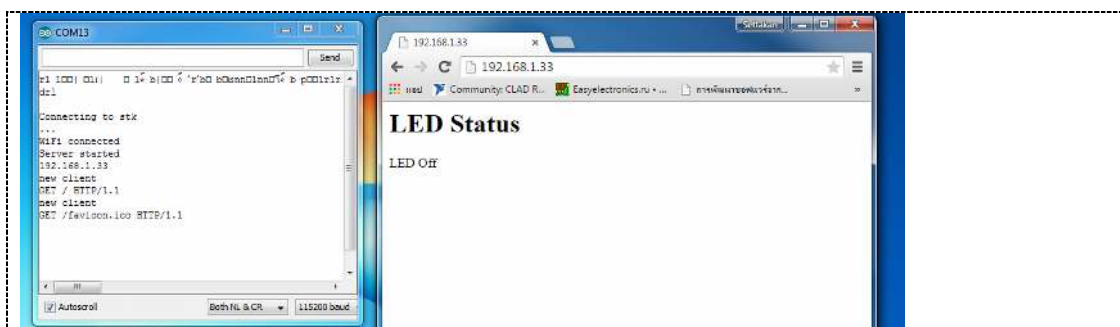
  if (client) {                             // if you get a client,
    Serial.println("New Client.");           // print a message out the serial port
    String currentLine = "";                 // make a String to hold incoming data from the client
    while (client.connected()) {             // loop while the client's connected
      if (client.available()) {               // if there's bytes to read from the client,
        char c = client.read();               // read a byte, then
        Serial.write(c);                     // print it out the serial monitor
        if (c == '\n') {                     // if the byte is a newline character
          if (currentLine.length() == 0) {
            client.println("HTTP/1.1 200 OK");
            client.println("Content-type:text/html");
            client.println();
            client.println("<html>");
            client.println("<body>");
            client.println("<h1>LED Status</h1>");
            client.println("<p>");
            if (LED_Status == HIGH)
              client.println("LED On");
            else
              client.println("LED Off");
            client.println("<p>");
            client.println("<body>");
            client.println("<html>");
            break;
          } else {
            currentLine = "";
          }
        } else if (c != '\r') {
          currentLine += c;
        }
        if (currentLine.endsWith("GET /ledon")) LED_Status = HIGH;
        if (currentLine.endsWith("GET /ledoff")) LED_Status = LOW;
      }
    }
    client.stop(); // close the connection:
    Serial.println("Client Disconnected.");
  }
}
```

การทดลองนี้เป็นการนำเอา ESP-32 มาสร้างเป็น Web Server โดยเมื่อมีการร้องขอหน้าเว็บไซต์มาเป็นตัวกำหนดให้หลอด LED ติดดับ

ต่อวงจร โหลดโปรแกรมและ อย่าลืมแก้เป็นชื่อ SSID และ Password เป็นของตัวเอง



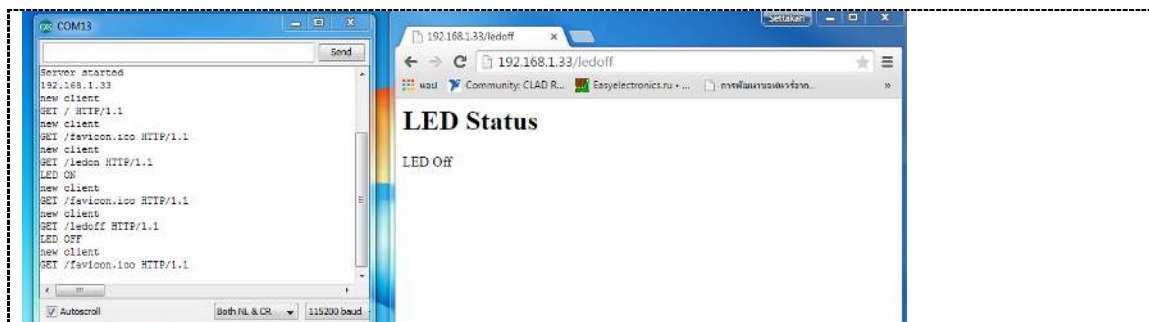
เปิด Web Browser แล้วกำหนด url ไปที่ IP ของ ESP-32 → YY.YY.YY.YY



เรียกหน้าเว็บไปที่ YY.YY.YY.YY/ledon สังเกต >> หลอด LED จะติด



เรียกหน้าเว็บไปที่ YY.YY.YY.YY/ledoff สังเกต >> หลอด LED จะดับ



การทดลอง 5: Web Server-Button

8. Test Ex21_WebServer_Button

```
#include <WiFi.h>

const char* ssid = "SUT_IoTs";
const char* password = "MaiMeeJingJing";
int pinTest = 2;

WiFiServer server(80);

void setup() {
  Serial.begin(115200);
  pinMode(pinTest, OUTPUT); // set the LED pin mode
  delay(10);
  Serial.print("\n\nConnecting to "); Serial.println(ssid);
  WiFi.begin(ssid, password);

  while (WiFi.status() != WL_CONNECTED) {
    delay(500); Serial.print(".");
  }

  Serial.println("");
  Serial.println("WiFi connected."); Serial.println("IP address: ");
  Serial.println(WiFi.localIP()); server.begin();
}

int value = 0;
bool LED_Status = LOW;

void loop() {
  digitalWrite(pinTest, LED_Status);
  WiFiClient client = server.available(); // listen for incoming clients

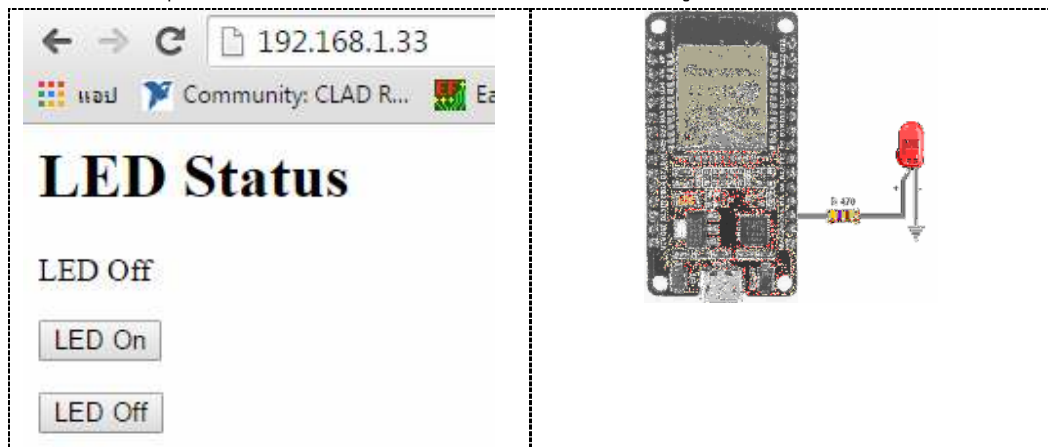
  if (client) { // if you get a client,
    Serial.println("New Client."); // print a message out the serial port
    String currentLine = ""; // make a String to hold incoming data from the client
    while (client.connected()) { // loop while the client's connected
      if (client.available()) { // if there's bytes to read from the client,
        char c = client.read(); // read a byte, then
        Serial.write(c); // print it out the serial monitor
        if (c == '\n') { // if the byte is a newline character
          if (currentLine.length() == 0) {
            client.println("HTTP/1.1 200 OK");
            client.println("Content-type:text/html");
            client.println();
            client.println("<html>");
            client.println("<body>");
            client.println("<h1>LED Status</h1>");

            client.println("<p>");
            if (LED_Status == HIGH)
              client.println("LED On");
            else
              client.println("LED Off");

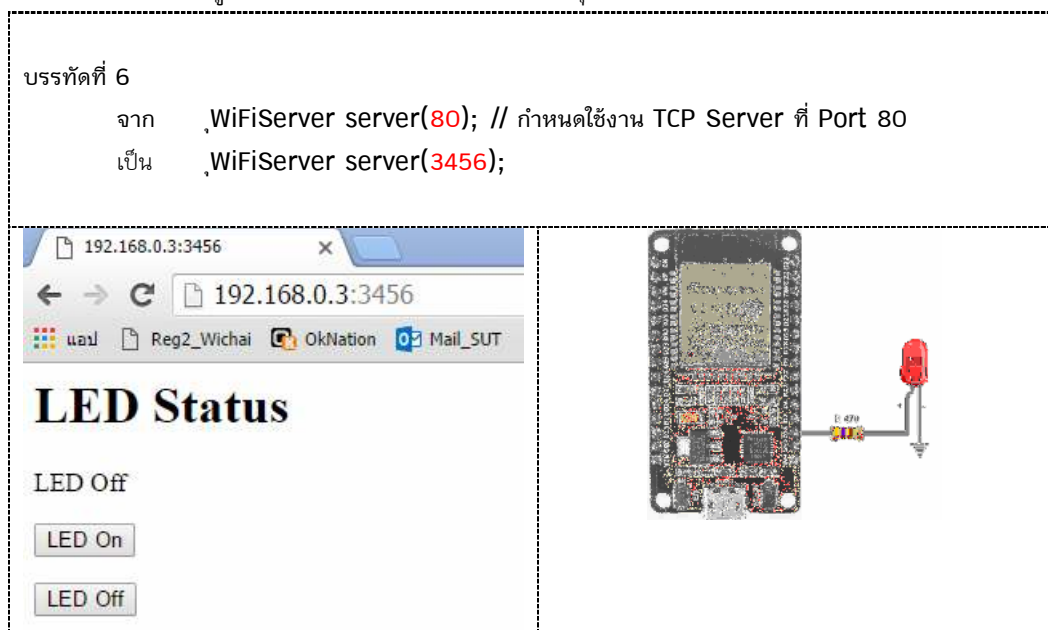
            client.println("<p>");
            client.println("<a href='\"/ledon\"'><button>LED On</button></a>");
            client.println("</p>");
            client.println("<a href='\"/ledoff\"'><button>LED Off</button></a>");

            client.println("<body>");
            client.println("<html>");
            break;
          } else {
            currentLine = "";
          }
        } else if (c != '\r') {
          currentLine += c;
        }
      }
      if (currentLine.endsWith("GET /ledon")) LED_Status = HIGH;
      if (currentLine.endsWith("GET /ledoff")) LED_Status = LOW;
    }
    client.stop(); // close the connection:
    Serial.println("Client Disconnected.");
  }
}
```

- ควบคุมการทำงานของ LED D1 โดยต่อวงจรตามรูป



- ลองปรับแก้จากพอร์ต 80 เป็นพอร์ตอื่น เช่น 3456
- การเรียกดูผ่าน Web Browser ก็ต้องระบุพอร์ตด้วย เช่น **192.168.0.3:3456**



การทดลอง 6: Web Server-Digital Read

9. Test Ex30_WebServer_DigitalRead

```
#include <WiFi.h>
#define pinTest 2
#define SW_Test 39

const char* ssid = "SUT_IoTs";
const char* password = "MaiMeeJingJing";
WiFiServer server(80);

void setup() {
  Serial.begin(115200);
  pinMode(SW_Test, INPUT_PULLUP);
  pinMode(pinTest, OUTPUT);
  delay(10);
  Serial.print("\n\nConnecting to "); Serial.println(ssid);
  WiFi.begin(ssid, password);

  while (WiFi.status() != WL_CONNECTED) {
    delay(500); Serial.print(".");
  }

  Serial.println("");
  Serial.println("WiFi connected."); Serial.println("IP address: ");
  Serial.println(WiFi.localIP()); server.begin();
}

int value = 0;
bool LED_Status = LOW;

void loop() {
  digitalWrite(pinTest, LED_Status);
  WiFiClient client = server.available(); // listen for incoming clients

  if (client) { // if you get a client,
    Serial.println("New Client."); // print a message out the serial port
    String currentLine = ""; // make a String to hold incoming data from the client
    while (client.connected()) { // loop while the client's connected
      if (client.available()) { // if there's bytes to read from the client,
        char c = client.read(); // read a byte, then
        Serial.write(c); // print it out the serial monitor
        if (c == '\n') { // if the byte is a newline character
          if (currentLine.length() == 0) {
            client.println("HTTP/1.1 200 OK");
            client.println("Content-type:text/html");
            client.println();
            client.println("<html>");
            client.println("<body>");
            client.println("<h1>LED Status</h1>");

            client.println("<p>");
            if (LED_Status == HIGH)
              client.println("LED On");
            else
              client.println("LED Off");

            client.println("<p>");
            client.println("<a href='\"/ledon\"'><button>LED On</button></a>");
            client.println("<a href='\"/ledoff\"'><button>LED Off</button></a>");
            client.println("</p>");

            client.println("<h1>Read Switch</h1>");
            client.println("<style>");
            client.println(".circle-green,.circle-red");
            client.println("{width: 100px; height: 100px; border-radius: 50%;}");
            client.println(".circle-green {background-color: green;}");
            client.println(".circle-red {background-color: red;}");
            client.println("</style>");

            client.println("<meta http-equiv='\"refresh\"' content='\"1\"'>");
            client.println("<p>");
            if (digitalRead(SW_Test) == HIGH)
              { client.println("<div class='\"circle-red\"'></div>");
                client.println("<p>SW = 1</p>");
              }
            else
              { client.println("<div class='\"circle-green\"'></div>");
                client.println("<p>SW = 0</p>");
              }
            client.println("</p>");
            client.println("<body>");
            client.println("<html>");
            break;
          }
          currentLine += c;
        }
      }
    }
  }
}
```

```
} else {
  currentLine = "";
}
} else if (c != '\r') {
  currentLine += c;
}
if (currentLine.endsWith("GET /ledon")) LED_Status = HIGH;
if (currentLine.endsWith("GET /ledoff")) LED_Status = LOW;
}
}
client.stop(); // close the connection:
Serial.println("Client Disconnected.");
}
}
```

- วงจรตามรูป

LED Status

LED On

LED On LED Off

Read Switch

SW = 1

- เปิด Web Browser แล้วเรียกหน้า Page ไปยัง IP ของ Node MCU

เมื่อไม่กด Switch	ทดลองกด Switch
<div><h2>Read Switch</h2><div></div><p>SW = 1</p></div>	<div><h2>Read Switch</h2><div></div><p>SW = 0</p></div>

การทดลอง 7: Web Server-Sensor

10. Test Ex40_DHT22 Sensor ทดสอบโปรแกรมนี้

- ให้แน่ใจว่าใช้ DHT-22 library ของ [beegee_tokyo Ver 1.0.5](#)

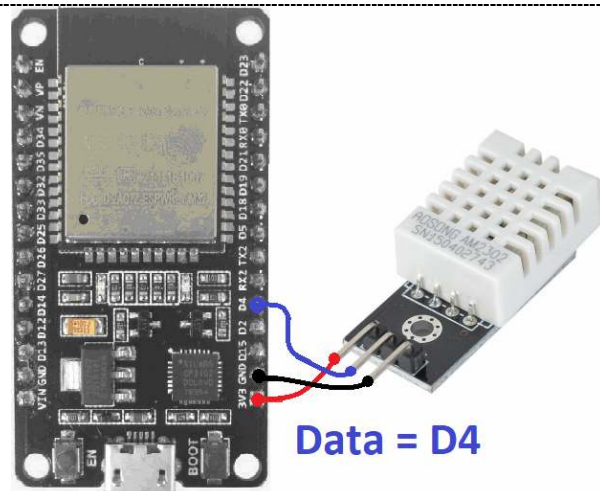
DHT sensor library for ESPx by beegee_tokyo Version 1.0.5 INSTALLED
Arduino ESP library for DHT11, DHT22, etc Temp & Humidity Sensors Op
 changes: Use correct field separator in keywords.txt.
[More info](#)

- DHT-22 Test Code

```
#include "DHTesp.h"
DHTesp dht;

void setup()
{ Serial.begin(115200);
  Serial.println();
  Serial.println("Status\tTemperature
(C)\tHumidity (%)");
  dht.setup(4); // data pin D4
}

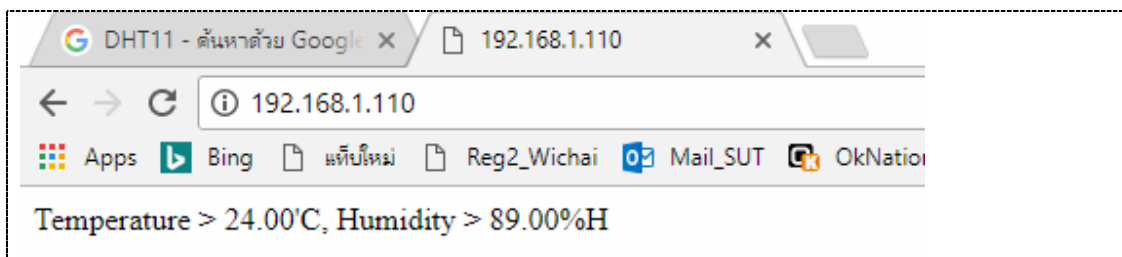
void loop()
{ delay(2000);
  float humidity = dht.getHumidity();
  float temperature = dht.getTemperature();
  Serial.println();
  Serial.print(dht.getStatusString());
  Serial.print("\t");
  Serial.print(temperature, 1);
  Serial.print("\t\t");
  Serial.print(humidity, 1);
}
```



Status	Temperature (C)	Humidity (%)
OK	27.6	46.2
OK	27.6	46.3
OK	27.6	46.3
OK	27.5	45.9
OK	27.5	45.4
OK	27.5	45.1
OK	27.5	45.2
OK	27.5	45.4

11. Test Ex41_WebServer_DHT22 Sensor

- ผลของการทดสอบการทำงาน



● WebServer Test Code

```
#include <WiFi.h>
#include "DHTesp.h"
#define DHT_Pin 5
const char* ssid = "SUT_IoTs";
const char* password = "MaiMeeJingJing";

DHTesp dht;
WiFiServer server(80);

void setup() {
  Serial.begin(115200);
  Serial.print("\n\nConnecting to ");
  Serial.println(ssid);
  WiFi.begin(ssid, password);
  while (WiFi.status() != WL_CONNECTED)
  { delay(500); Serial.print(".");
  }
  Serial.println();
  Serial.println("WiFi connected");
  server.begin();
  Serial.println("Server started");
  Serial.println(WiFi.localIP());
  dht.setup(DHT_Pin);
}

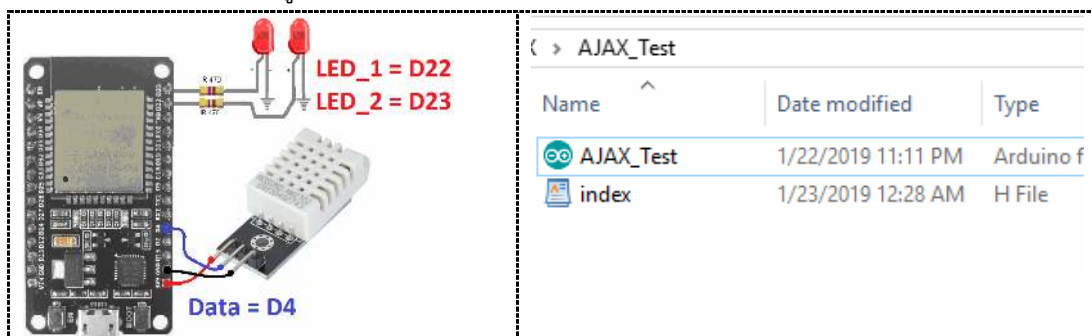
void loop() {
  WiFiClient client = server.available(); // wait Client request
  if (client) {
    Serial.println("new client"); // an http request ends with a blank line
    Serial.println("Requesting temperatures...");
    Serial.print("Temperature is: ");
    float Humid = dht.getHumidity();
    float cTemp = dht.getTemperature();
    Serial.print(cTemp, 2); Serial.print("C, ");
    Serial.print(Humid, 2); Serial.println("%H");
    boolean currentLineIsBlank = true;
    while (client.connected()) {
      if (client.available()) {
        char c = client.read();
        Serial.write(c);
        if (c == '\n' && currentLineIsBlank) // send a standard http response header
        { client.println("HTTP/1.1 200 OK");
          client.println("Content-Type: text/html");
          client.println("Connection: close"); // the connection will be closed after completion of the response
          client.println("Refresh: 5"); // refresh the page automatically every 5 sec
          client.println();
          client.println("<!DOCTYPE HTML>");
          client.println("<html>");
          client.print("Temperature > "); client.print(cTemp, 2);
          client.print("C, Humidity > "); client.print(Humid, 2);
          client.print("%H");
          client.println("<br />");
          client.println("</html>");
          break;
        }
      }
      if (c == '\n') // you're starting a new line
      { currentLineIsBlank = true;
      }
      else if (c != '\r') // you've gotten a character on the current line
      { currentLineIsBlank = false;
      }
    }
  }
  delay(1); // give the web browser time to receive the data
  client.stop(); // close the connection:
  Serial.println("client disconnected");
}
}
```

การทดลอง 8: Web Server-AJAX

12. Test Ex50_WebServer AJAX_Monitor by AJAX

ในตัวอย่างก่อนหน้านี้ การ Update สถานะของการกด Button จะใช้การ refresh หน้า web ทั้งหมดทุกๆ 1 วินาที ทำให้ทั้งหน้ากระพริบ และ เป็นการรับ/ส่ง Data ที่ค่อนข้างสิ้นเปลือง เนื่องจากในบางส่วนไม่ได้มีการเปลี่ยนแปลงแต่เราต้อง update ทั้งหมด ในการทดลองนี้เราได้นำเอา Ajax เข้ามาช่วยให้สามารถ update ข้อมูลมาแสดงเฉพาะส่วนที่มีการเปลี่ยนแปลง ทำให้ไม่ต้อง refresh ทั้งหมด และลดภาระของ Web server ให้ทำงานน้อยลง

- Read More <https://circuits4you.com/2018/11/20/web-server-on-esp32-how-to-update-and-display-sensor-values/>
- Read More <https://circuits4you.com/2018/02/04/esp8266-ajax-update-part-of-web-page-without-refreshing/>
- ตัวอย่างดังรูป ดังนี้



- Create Program “Test_AJAX_01.ino”
- Create “index.h” file and save to Test_AJAX_01 folder
- Upload “Test_AJAX_01.ino” to ESP-32
- เปิด Web Browser แล้วเรียกหน้า Page ไปยัง IP ของ ESP-32



- Code “Test_AJAX_01.ino” – WebServer AJAX Monitor by AJAX

```
#include <WiFi.h>
#include <WiFiClient.h>
#include <WebServer.h>
#include "index.h" //Web page header file

#include "DHTesp.h"
#define DHT_Pin 17 // DHT-11 Pin D4

const char* ssid = "SUT_IoTs";
const char* password = "MaiMeeJingJing";

DHTesp dht;
WebServer server(80);

void handleRoot() {
  String s = MAIN_page; //Read HTML contents
  server.send(200, "text/html", s); //Send web page
}

void handleADC() {
  float h = dht.getHumidity();
  float t = dht.getTemperature();
  String Value = String(t) + " C, ";
  Value += String(h) + " %";
  Serial.print("DHT-22 >> ");
  Serial.println(Value);
  server.send(200, "text/plain", Value);
}

void setup(void) {
  Serial.begin(115200);
  Serial.println("\nBooting Sketch...");
  WiFi.mode(WIFI_STA); //Connect to your wifi
  WiFi.begin(ssid, password);
  Serial.println("Connecting to ");
  Serial.print(ssid);

  while (WiFi.waitForConnectResult() != WL_CONNECTED) {
    Serial.print(".");
    delay(10);
  }

  Serial.print("\nConnected to ");
  Serial.println(ssid);
  Serial.print("IP address: ");
  Serial.println(WiFi.localIP()); //IP address assigned to your ESP

  server.on("/", handleRoot); //This is display page
  server.on("/readADC", handleADC); //To get update of ADC Value only
  server.begin(); //Start server
  Serial.println("HTTP server started");
  dht.setup(DHT_Pin);
}

void loop(void) {
  server.handleClient();
  delay(1);
}
```

- Test_AJAX_01 = "index.h"

```
const char MAIN_page[] PROGMEM = R"=====(
<!DOCTYPE html>
<html>
<style>
.card{
  max-width: 500px;
  min-height: 250px;
  background: #02b875;
  padding: 30px;
  box-sizing: border-box;
  color: #FFF;
  margin: 20px;
  box-shadow: 0px 2px 18px -4px rgba(0,0,0,0.75);
}
</style>
<body>

<div class="card">
  <h4>The ESP32 Update web page without refresh</h4><br>
  <h1>DHT-22 >> :<span id="ADCValue">0</span></h1><br>
  <br><a href="https://www.facebook.com/groups/311747285898180/">551402 Microcontroller System 4(4-0-8)</a>
</div>
<script>

setInterval(function() {
  // Call a function repetatively with 2 Second interval
  getData();
}, 2000); //2000mSeconds update rate

function getData() {
  var xhttp = new XMLHttpRequest();
  xhttp.onreadystatechange = function() {
    if (this.readyState == 4 && this.status == 200) {
      document.getElementById("ADCValue").innerHTML =
        this.responseText;
    }
  };
  xhttp.open("GET", "readADC", true);
  xhttp.send();
}
</script>
</body>
</html>
)=====";
```

13. Test Ex51_WebServer AJAX_Control Monitor by AJAX

- Create Program “Test_AJAX_02.ino”
- Create “index.h” file and save to Test_AJAX_02 folder
- Upload “Test_AJAX_02.ino” to ESP-32
- เปิด Web Browser แล้วเรียกหน้า Page ไปยัง IP ของ ESP-32



- Code “Test_AJAX_02.ino” – WebServer AJAX Monitor by AJAX

```
#include <WiFi.h>
#include <WiFiClient.h>
#include <WebServer.h>
#include "DHTesp.h"

#include "index.h" //Our HTML webpage contents with javascripts
#define DHT_Pin 21
#define testLED1 22
#define testLED2 23

//SSID and Password of your WiFi router
const char* ssid = "SUT_IoTs";
const char* password = "MaiMeeJingJing";

WebServer server(80); //Server on port 80
DHTesp dht;

String ledState1 = "NA";
String ledState2 = "NA";

//=====
// This routine is executed when you open its IP in browser
//=====
void handleRoot() {
  String s = MAIN_page; //Read HTML contents
  server.send(200, "text/html", s); //Send web page
}

void handleADC() {
  float h = dht.getHumidity();
  float t = dht.getTemperature();
  String tmpValue = "Temp = ";
  tmpValue += String(t) + " C, Humidity = ";
  tmpValue += String(h) + " %";
  server.send(200, "text/plain", tmpValue); //Send value to client ajax request
}

void handleLED() {
  String t_state = server.arg("LEDstate"); //Refer xhttp.open("GET", "setLED?LEDstate="+led, true);
  Serial.println(t_state);
  if (t_state == "11") { digitalWrite(testLED1, HIGH); ledState1 = "ON"; } //Feedback parameter
  if (t_state == "10") { digitalWrite(testLED1, LOW); ledState1 = "OFF"; } //Feedback parameter
  if (t_state == "21") { digitalWrite(testLED2, HIGH); ledState2 = "ON"; } //Feedback parameter
  if (t_state == "20") { digitalWrite(testLED2, LOW); ledState2 = "OFF"; } //Feedback parameter
  server.send(200, "text/plain", ledState1+" "+ledState2); //Send web page
}

void setup(void) {
  Serial.begin(115200);
  dht.setup(DHT_Pin);
  pinMode(testLED1, OUTPUT);
  pinMode(testLED2, OUTPUT);
  Serial.print("\n\nConnect to ");
  Serial.println(ssid);
  WiFi.begin(ssid, password);
  while (WiFi.status() != WL_CONNECTED) {
    delay(500); Serial.print(".");
  }
  Serial.print("\nConnected "); Serial.println(ssid);
  Serial.print("IP address: "); Serial.println(WiFi.localIP());

  server.on("/", handleRoot);
  server.on("/setLED", handleLED);
  server.on("/readADC", handleADC);

  server.begin();
  Serial.println("HTTP server started");
}

void loop(void) {
  server.handleClient(); //Handle client requests
}
```

- Test_AJAX_02 = "index.h"

```
const char MAIN_page[] PROGMEM = R"=====(
<!DOCTYPE html>
<html>
<body>

<div id="demo">
<h1>The ESP-32 Update web page without refresh</h1>
<button type="button" onclick="sendData(11)" style="background: rgb(202, 60, 60);">LED1 ON</button>
<button type="button" onclick="sendData(10)" style="background: rgb(100,116,255);">LED1 OFF</button><br><br>
<button type="button" onclick="sendData(21)" style="background: rgb(202, 60, 60);">LED2 ON</button>
<button type="button" onclick="sendData(20)" style="background: rgb(100,116,255);">LED2 OFF</button><br><br>
State of [LED1, LED2] is >> <span id="LEDState">NA</span><br>
</div>

<div>
<br>DHT-22 sensor : <span id="ADCValue">0</span><br>
</div>

<script>
function sendData(led) {
var xhttp = new XMLHttpRequest();
xhttp.onreadystatechange = function() {
if (this.readyState == 4 && this.status == 200) {
document.getElementById("LEDState").innerHTML =
this.responseText;
}
};
xhttp.open("GET", "setLED?LEDstate="+led, true);
xhttp.send();
}

setInterval(function() {
// Call a function repetatively with 2 Second interval
getData();
}, 2000); // 2000mSeconds update rate

function getData() {
var xhttp = new XMLHttpRequest();
xhttp.onreadystatechange = function() {
if (this.readyState == 4 && this.status == 200) {
document.getElementById("ADCValue").innerHTML =
this.responseText;
}
};
xhttp.open("GET", "readADC", true);
xhttp.send();
}
</script>

<br><a href="https://www.facebook.com/groups/311747285898180/">551402 Microcontroller System 4(4-0-8)</a>

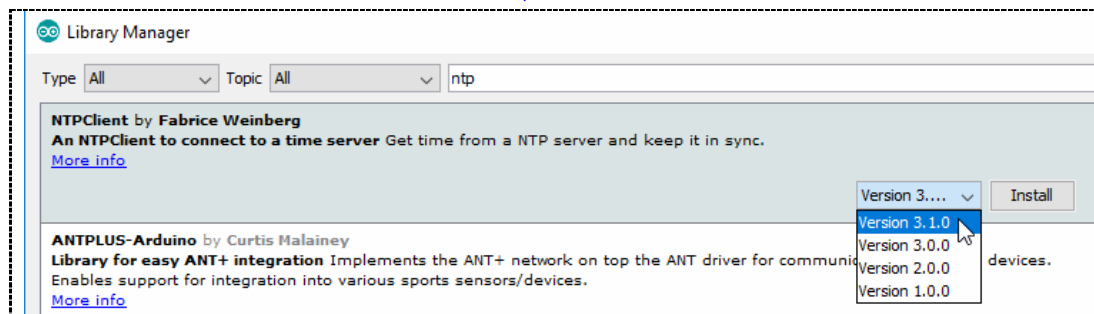
</body>
</html>
)=====";
```


การทดลอง 9: Read Time via NTP

11. Read Time via NTP

14. Test Ex60_WebClient_GetTime via NTP

- Install NTPClient Arduino Library Ver 3.1.0 < <https://github.com/arduino-libraries/NTPClient/archive/master.zip> >



- ทดสอบโปรแกรม

```
// ESP-32 Board

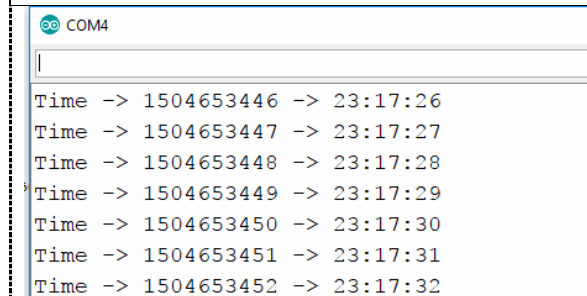
#include <NTPClient.h>
#include <WiFi.h>
#include <WiFiUdp.h>

const char* ssid = "SUT_IoTs";
const char* password = "MaiMeeJingJing";

WiFiUDP ntpUDP;
NTPClient timeClient(ntpUDP);

void setup() {
  Serial.begin(115200);
  WiFi.begin(ssid, password);
  while ( WiFi.status() != WL_CONNECTED )
  { delay ( 500 );
    Serial.print ( "." );
  }
  timeClient.begin();
  timeClient.setTimeOffset(25200); // +7Hour = (7H x 60Min x 60Sec = 25200 Sec)
  Serial.println();
}

void loop() {
  timeClient.update();
  Serial.print("Time -> "); Serial.print(timeClient.getEpochTime());
  Serial.print(" -> "); Serial.println(timeClient.getFormattedTime());
  delay(1000);
}
```



```

// M5Stack Board

#include <M5Stack.h>
#include <WiFi.h>
#include <WiFiUdp.h>
#include <NTPClient.h>

const char* ssid = "SUT_IoTs";
const char* password = "MaiMeeJingJing";

WiFiUDP ntpUDP;
NTPClient timeClient(ntpUDP);

void setup() {
  M5.begin();
  M5.Lcd.setTextSize(3);
  M5.Lcd.print("Wifi Connecting.");
  Serial.begin(115200);
  WiFi.begin(ssid, password);
  while ( WiFi.status() != WL_CONNECTED )
  { delay ( 500 );
    M5.Lcd.print ( "." );
    Serial.print ( "." );
  }
  timeClient.begin();
  timeClient.setTimeOffset(25200); // +7Hour = (7H x 60Min x 60Sec = 25200 Sec)
  Serial.println();
  M5.Lcd.clear();
  M5.Lcd.setTextSize(5);
  M5.Lcd.setTextColor(RED);
  M5.Lcd.setCursor(10, 10);
  M5.Lcd.print("Time");
}

void loop() {
  timeClient.update();
  String formattedTime = timeClient.getFormattedTime();
  unsigned long epochTime = timeClient.getEpochTime();
  Serial.print("Time -> "); Serial.print(epochTime);
  Serial.print(" -> "); Serial.println(formattedTime);
  M5.Lcd.setTextColor(BLUE);
  M5.Lcd.setCursor(20, 80); M5.Lcd.print(epochTime);
  M5.Lcd.setCursor(20, 150); M5.Lcd.print(formattedTime);
  delay(1000);
  M5.Lcd.setTextColor(BLACK);
  M5.Lcd.setCursor(20, 80); M5.Lcd.print(epochTime);
  M5.Lcd.setCursor(20, 150); M5.Lcd.print(formattedTime);
}

```

- <http://forum.m5stack.com/topic/41/lesson-1-1-lcd-graphics>

```

// M5Stack Board

#include <M5Stack.h>
#include "DHTesp.h"
#include <WiFi.h>
#include <WiFiUdp.h>
#include <NTPClient.h>
#define DHT22_Pin 5

const char* ssid = "SUT_IoTs";
const char* password = "MaiMeeJingJing";

DHTesp dht;
WiFiUDP ntpUDP;
NTPClient timeClient(ntpUDP);

void setup() {
  M5.begin();
  M5.Lcd.setTextSize(3);
  M5.Lcd.print("Wifi Connecting.");
  Serial.begin(115200);
  WiFi.begin(ssid, password);
  while ( WiFi.status() != WL_CONNECTED )
  { delay ( 500 );
    M5.Lcd.print ( "." );
    Serial.print ( "." );
  }
  timeClient.begin();
  timeClient.setTimeOffset(25200); // +7Hour = (7H x 60Min x 60Sec = 25200 Sec)
  dht.setup(DHT22_Pin);
  Serial.println();
  M5.Lcd.clear();
  M5.Lcd.setTextSize(5);
  M5.Lcd.setTextColor(RED);
  M5.Lcd.setCursor(10, 10); M5.Lcd.print("Time");
  M5.Lcd.setCursor(10, 105); M5.Lcd.print("DHT-22");
}

void loop() {
  timeClient.update();
  String formattedTime = timeClient.getFormattedTime();
  String temperature = (String)dht.getTemperature() + " C";
  String humidity = (String)dht.getHumidity() + " %";
  Serial.print("Time = "); Serial.print(formattedTime);
  Serial.print(", Temp = "); Serial.println(temperature);
  Serial.print(", Humidity = "); Serial.println(humidity);
  M5.Lcd.setTextColor(BLUE);
  M5.Lcd.setCursor(40, 60); M5.Lcd.print(formattedTime);
  M5.Lcd.setCursor(40, 145); M5.Lcd.print(temperature);
  M5.Lcd.setCursor(40, 185); M5.Lcd.print(humidity);
  delay(1000);
  M5.Lcd.setTextColor(BLACK);
  M5.Lcd.setCursor(40, 60); M5.Lcd.print(formattedTime);
  M5.Lcd.setCursor(40, 145); M5.Lcd.print(temperature);
  M5.Lcd.setCursor(40, 185); M5.Lcd.print(humidity);
}

```

PCเลขที่ _____ รหัส _____ ชื่อ-สกุล _____

4. คำถามท้ายการทดลอง- ตรวจสอบการทำงานตามการทดลอง

การทดลอง 1: Start LED Blink

การทดลอง 2: My MAC Address

การทดลอง 3: Connect to Network

การทดลอง 4: Web Server-Command

การทดลอง 5: Web Server-Button

การทดลอง 6: Web Server-Digital Read

การทดลอง 7: Web Server-Sensor

การทดลอง 8: Web Server-AJAX

การทดลอง 9: Read Time via NTP

จากเอกสาร Word ทำการปรับแก้ไขเป็นของตัวเอง
แปลงเป็น pdf ไฟล์ แก้ไขชื่อเป็น รหัส-ชื่อ.pdf

- เช่น B3706585 - วิชัย ศรีสุรักษ์.pdf

Upload ก่อนวันจันทร์ที่ 28 มกราคม 2562 ที่

- กลุ่ม วันอังคาร ส่งที่ <https://goo.gl/ySifm3>
- กลุ่ม วันเสาร์ ส่งที่ <https://goo.gl/kLxDgz>