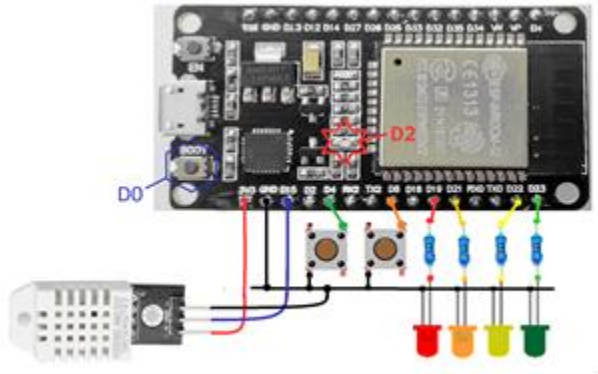
แนวทางการใช้งานอินเทอร์เน็ตของสรรพสิ่งในระบบการผลิต

IoT Approaches to Manufacturing System

ชื่อ-สกุล : นายปราชญา ธนพิบูลผล **รหัสนักศึกษา : B6323059**

5/5. คำถามท้ายบทเพื่อทดสอบความเข้าใจ

Quiz_401 – Ubidots: Monitor DHT22, Monitor Digital Switch and Control 4 LED




```
#include <WiFi.h>

#include <PubSubClient.h>

#include "DHTesp.h"

const char *My_SSID = "It's bad day not a bad life";

const char *My_Pass = "0641453596";

const char *MQTT_Server = "industrial.api.ubidots.com";

const char *MQTT_User = "BBFF-RgUJDWTeZdrMKJmwvoxp5W6ePSMYif";

const char *MQTT_Pass = "BBFF-RgUJDWTeZdrMKJmwvoxp5W6ePSMYif";

const char *PTopic1 = "/v2.0/devices/test/BBFF-GaUujMiW5i0BBGT6g7gBLHEmkO7i3W/lv";

const char *STopic1 = "/v2.0/devices/test/humid";

const char *STopic2 = "/v2.0/devices/test/tempp";

const char *STopic3 = "/v2.0/devices/test/led1";

const char *STopic4 = "/v2.0/devices/test/led2";
```

```
const char *STopic5 = "/v2.0/devices/test/led3";

const char *STopic6 = "/v2.0/devices/test/led4";

const char *STopic7 = "/v2.0/devices/test/sw1";

const char *STopic8 = "/v2.0/devices/test/sw2";

#define MQTT_Port 1883

#define Test_LED1 19

#define Test_LED2 21

#define Test_LED3 22

#define Test_LED4 23

#define Test_SW1 2

#define Test_SW2 4

#define Pin_DHT22 15

DHTesp dht;

WiFiClient espClient;

PubSubClient client(espClient);

long lastMsg = 0;

char msg[50];

int value = 0;

void Setup_Wifi() {

delay(10);

Serial.println();

Serial.print("Connecting to ");

Serial.println(My_SSID);

WiFi.begin(My_SSID, My_Pass);
```

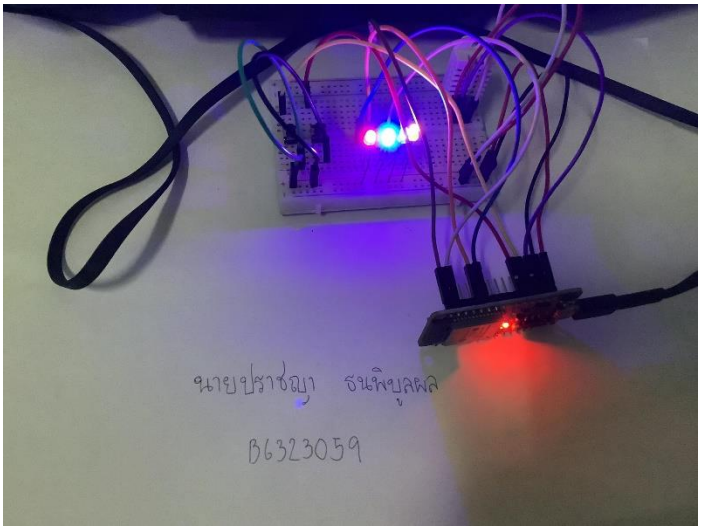
```
while (WiFi.status() != WL_CONNECTED) {
delay(500); Serial.print(".");
}
randomSeed(micros());
Serial.println("");
Serial.println("WiFi connected");
Serial.println("IP address: ");
Serial.println(WiFi.localIP());
}
void reconnect()
{ while (!client.connected()) // Loop until we're reconnected
{ Serial.print("Attempting MQTT connection...");
String clientId = "ESP32 Client-";
clientId += String(random(0xffff), HEX); // Create a random client ID
if (client.connect(clientId.c_str(), MQTT_User, MQTT_Pass)) // Attempt to connect
{ Serial.println("connected"); // Once connected, publish an announcement...
client.subscribe(STopic1);
client.subscribe(STopic2);
client.subscribe(STopic3);
client.subscribe(STopic4);
client.subscribe(STopic5);
client.subscribe(STopic6);
client.subscribe(STopic7);
client.subscribe(STopic8);
```

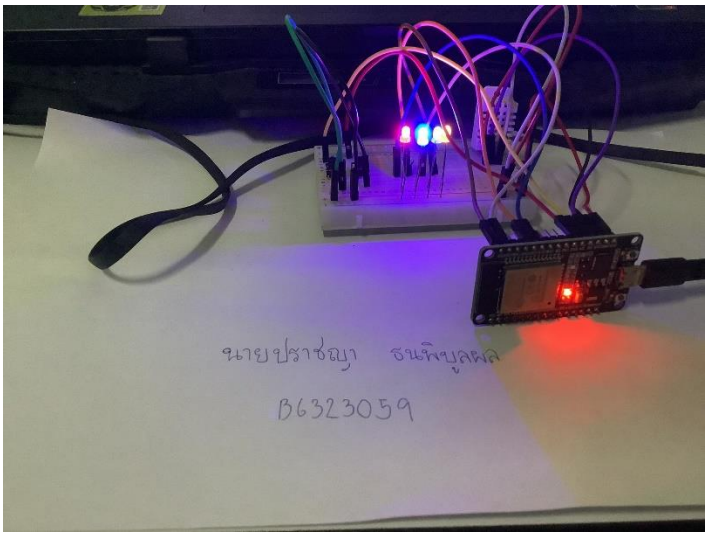
```
} else
{ Serial.print("failed, rc=");
Serial.print(client.state());
Serial.println(" try again in 5 seconds");
delay(5000);
}
}
}
void callback(char *topic, byte *payload, unsigned int length)
{ Serial.print("Message arrived [");
Serial.print(topic);
Serial.print("] ");
for (int i = 0; i < length; i++)
{ Serial.print((char)payload[i]);
}
if (topic[24] == STopic3[24]) {
Serial.print(" -LED1->> ");
Serial.print((char)payload[10]);
if (payload[10] == '1')
digitalWrite(Test_LED1, HIGH);
else
digitalWrite(Test_LED1, LOW);
}
if (topic[24] == STopic4[24]) {
```

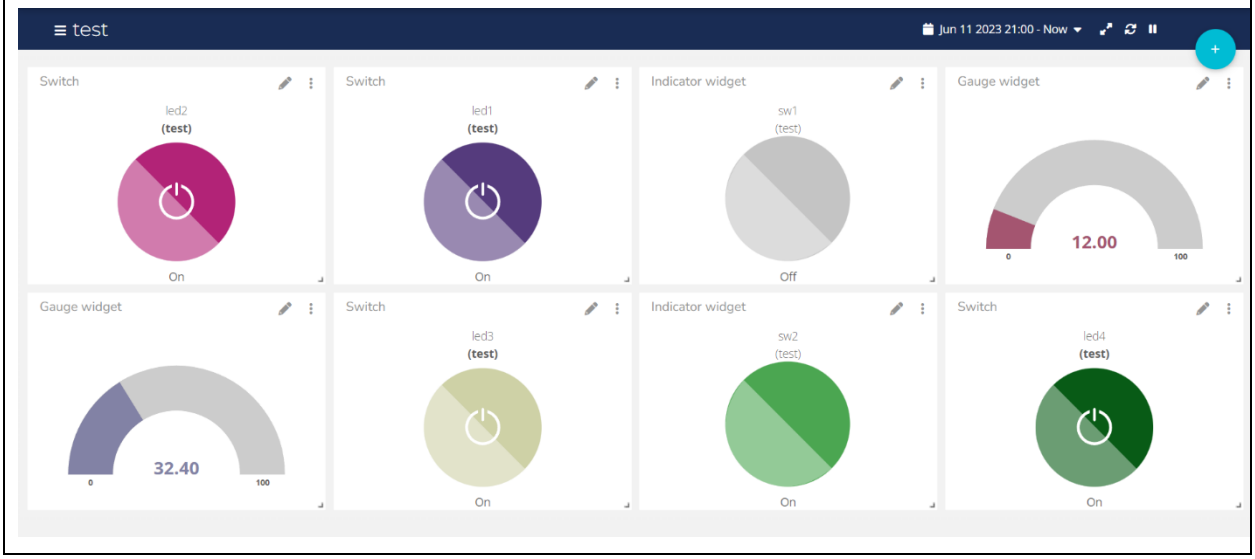
```
Serial.print(" -LED2->> ");
Serial.print((char)payload[10]);
if (payload[10] == '1')
digitalWrite(Test_LED2, HIGH);
else
digitalWrite(Test_LED2, LOW);
}
if (topic[24] == STopic5[24]) {
Serial.print(" -LED3->> ");
Serial.print((char)payload[10]);
if (payload[10] == '1')
digitalWrite(Test_LED3, HIGH);
else
digitalWrite(Test_LED3, LOW);
}
if (topic[24] == STopic6[24]) {
Serial.print(" -LED4->> ");
Serial.print((char)payload[10]);
if (payload[10] == '1')
digitalWrite(Test_LED4, HIGH);
else
digitalWrite(Test_LED4, LOW);
}
Serial.println();
```

```
}
void setup()
{ pinMode(Test_LED1, OUTPUT);
pinMode(Test_LED2, OUTPUT);
pinMode(Test_LED3, OUTPUT);
pinMode(Test_LED4, OUTPUT);
pinMode(Test_SW1, INPUT_PULLDOWN);
pinMode(Test_SW2, INPUT_PULLDOWN);
dht.setup(Pin_DHT22, DHTesp::DHT22);
Serial.begin(115200);
Setup_Wifi();
client.setServer(MQTT_Server, MQTT_Port);
client.setCallback(callback);
}
void loop()
{ if (!client.connected()) reconnect();
client.loop();
long now = millis();
if (now - lastMsg > 5000)
{ lastMsg = now;
float humidity = dht.getHumidity();
float temperature = dht.getTemperature();
int sw1 = 0;
int sw2 = 0;
```

```
if (digitalRead(Test_SW1)== HIGH) sw1 = 1;

else sw1 = 0;

if (digitalRead(Test_SW2)== LOW) sw2 = 1;

else sw2 = 0;

snprintf (msg, 75, "{ \"humid\" : %.2f, \"tempp\": %.2f, \"sw1\": %d, \"sw2\": %d }",

humidity, temperature, sw1, sw2);

Serial.print("Publish message: ");

Serial.println(msg);

client.publish(PTopic1, msg);

} }
```

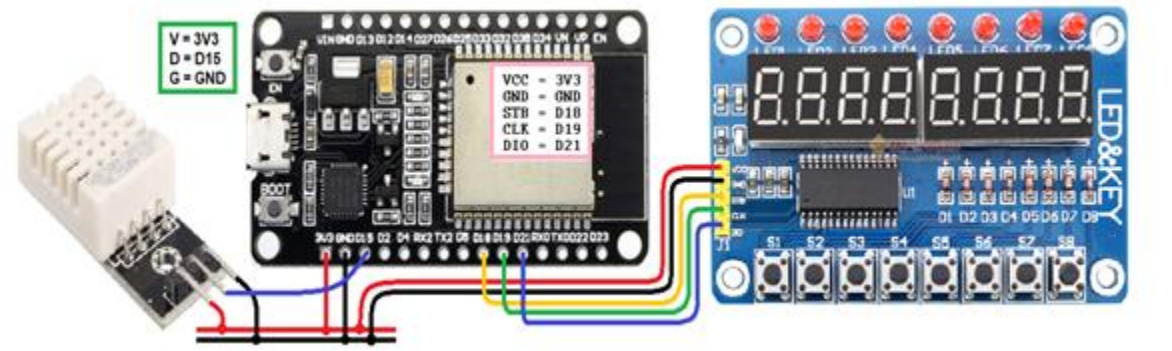



นายปราชญา ธนพิบูลผล
B6323059


test
Jun 11 2023 21:00 - Now
Switch
led2
(test)
On
Switch
led1
(test)
On
Indicator widget
sw1
(test)
Off
Gauge widget
12.00
0
100
Gauge widget
32.40
0
100
Switch
led3
(test)
On
Indicator widget
sw2
(test)
On
Switch
led4
(test)
On

Quiz_402 – Ubidots: Monitor DHT22 with TM1638 Display and LINE Alert

- ส่งข้อมูลอุณหภูมิไปยัง Ubidots
- หากอุณหภูมิที่อ่านได้เกิน 28'C ให้แจ้งเตือนผ่าน LINE และบอกด้วยว่าอุณหภูมิเท่าใด
- แสดงอุณหภูมิที่ 7_Segment Display TM1638 Board




```
#include <WiFi.h>

#include <PubSubClient.h>

#include <HTTPClient.h>

#include <TM1638plus.h>

#include "DHTesp.h"

const char *My_SSID = "It's bad day not a bad life";

const char *My_Pass = "0641453596";

const char *MQTT_Server = "industrial.api.ubidots.com";

const char *MQTT_User = "BBFF-RgUJDWTeZdrMKJmwvoxp5W6ePSMYif";

const char *MQTT_Pass = "BBFF-RgUJDWTeZdrMKJmwvoxp5W6ePSMYif";

#define WebHooksKey "jSJmaCRRDH7ibUiNBL7rTf2ZQiPdCSaq7Xkckg0C9KJ"

#define WebHooksEventName "Sheet"

#define WebHooksEventName_line "LINE"

const char *PTopic1 = "/v2.0/devices/test";

const char *STopic1 = "/v2.0/devices/test/humid";
```

```
const char *STopic2 = "/v2.0/devices/test/tempp";

#define Brd_STB 18 // strobe = GPIO connected to strobe line of module

#define Brd_CLK 19 // clock = GPIO connected to clock line of module


#define Brd_DIO 5 // data = GPIO connected to data line of module

bool high_freq = true; //default false,, If using a high freq CPU > ~100 MHZ set to true.

TM1638plus tm(Brd_STB, Brd_CLK , Brd_DIO, high_freq);

#define MQTT_Port 1883

#define Pin_DHT22 15

#define My_NAME "B6323059 Pratchaya Tanapiboonphol"

DHTesp dht;

WiFiClient espClient;

PubSubClient client(espClient);

long lastMsg = 0;

char msg[50];

int value = 0;

void Setup_Wifi() {

delay(10);

Serial.println();

Serial.print("Connecting to ");

Serial.println(My_SSID);

WiFi.begin(My_SSID, My_Pass);

while (WiFi.status() != WL_CONNECTED) {

delay(500); Serial.print(".");
```

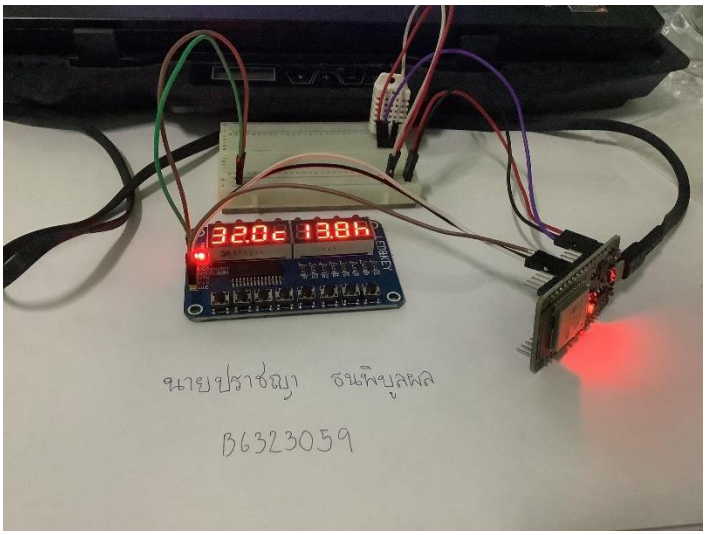
```
}
randomSeed(micros());
Serial.println("");
Serial.println("WiFi connected");
Serial.println("IP address: ");
Serial.println(WiFi.localIP());
}
void reconnect()
{ while (!client.connected()) // Loop until we're reconnected
{ Serial.print("Attempting MQTT connection...");
String clientId = "ESP32 Client-";
clientId += String(random(0xffff), HEX); // Create a random client ID
if (client.connect(clientId.c_str(), MQTT_User, MQTT_Pass)) // Attempt to connect
{ Serial.println("connected"); // Once connected, publish an announcement...
client.subscribe(STopic1);
client.subscribe(STopic2);
} else
{ Serial.print("failed, rc=");
Serial.print(client.state());
Serial.println(" try again in 5 seconds");
delay(5000);
}
}
}
```

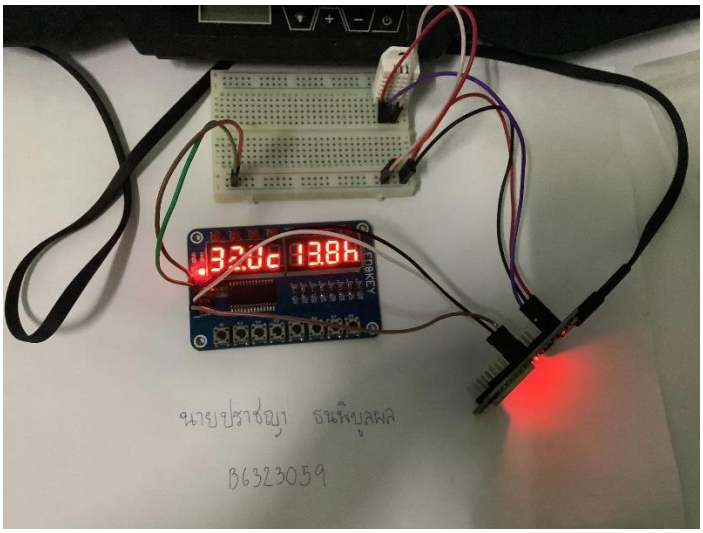
```
void setup()
{
tm.displayBegin();
dht.setup(Pin_DHT22, DHTesp::DHT22);
Serial.begin(115200);
Setup_Wifi();
client.setServer(MQTT_Server, MQTT_Port);
}
void loop()
{ if (!client.connected()) reconnect();
client.loop();
long now = millis();
if (now - lastMsg > 5000)
{ lastMsg = now;
float humidity = dht.getHumidity();
float temperature = dht.getTemperature();
snprintf (msg, 75, "{ \"humid\" : %.2f, \"tempp\": %.2f}", humidity, temperature);
Serial.print("Publish message: ");
Serial.println(msg);
client.publish(PTopic1, msg);
Serial.println();
Serial.print("\nTemperature('C) = ");
Serial.print(temperature, 1);
Serial.print("\tHumidity(%) = ");
```

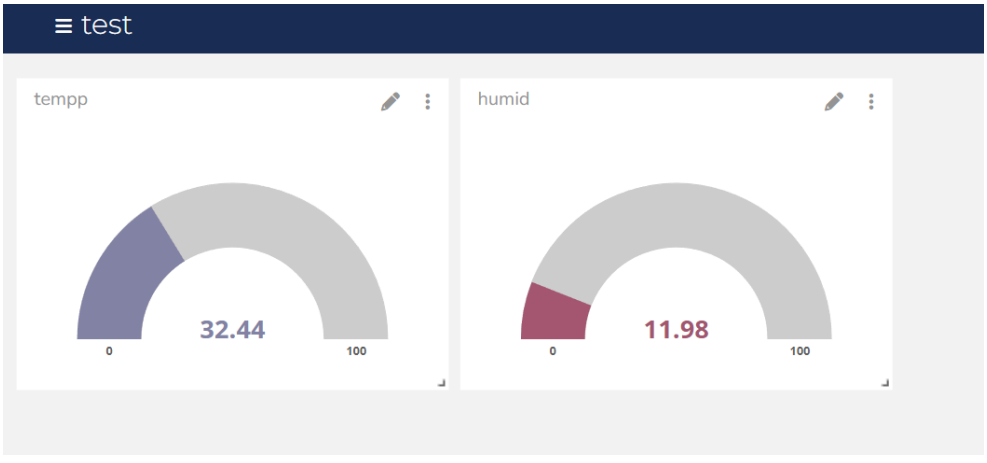
```
Serial.print(humidity, 1);
String serverName = "http://maker.ifttt.com/trigger/" +
String(WebHooksEventName) + "/with/key/" + String(WebHooksKey);
String httpRequestData = "value1=" + String(My_NAME) + "&value2=" +
String(temperature) + "&value3=" +
String(humidity);
Serial.println();
Serial.println("Server Name >> " + serverName);
Serial.println("json httpRequestData >> " + httpRequestData);
if (WiFi.status() == WL_CONNECTED) {
HTTPClient http;
http.begin(serverName);
http.addHeader("Content-Type", "application/x-www-form-urlencoded");
int httpResponseCode = http.POST(httpRequestData);
Serial.print("HTTP Response code: ");
Serial.println(httpResponseCode);
http.end();
if (httpResponseCode == 200)
Serial.println("[Google sheet] --> Successfully sent");
else
Serial.println("[Google sheet] --> Failed!");
}
else {
Serial.println("WiFi Disconnected");
```

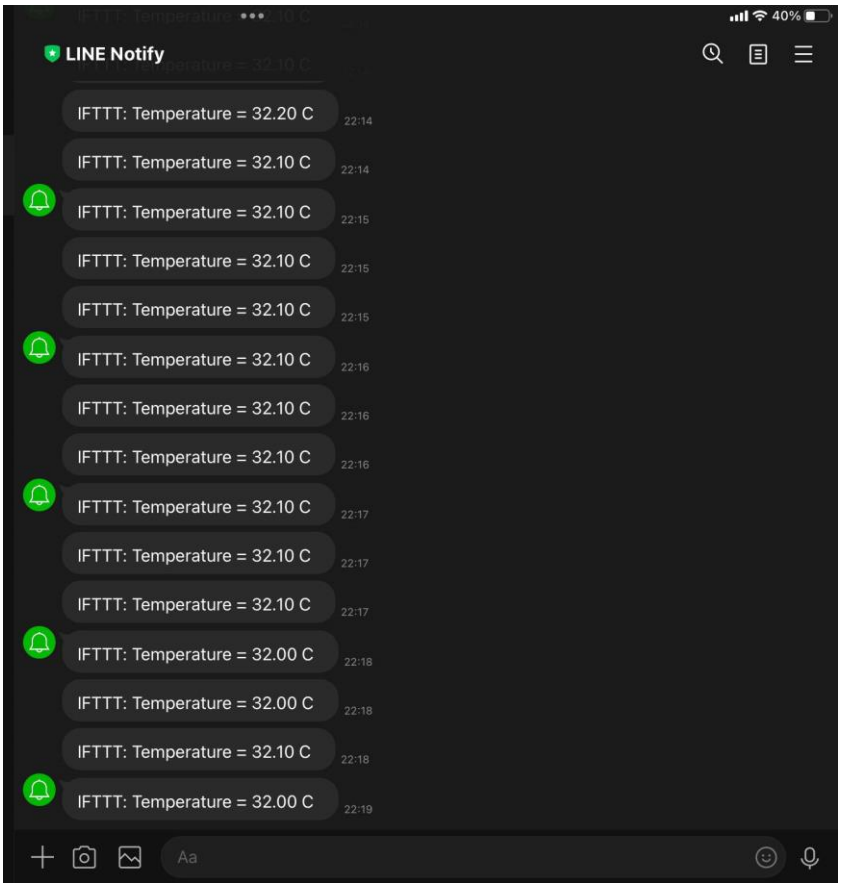
```
}

/// if temp > 28 C send notifications >> line

if (temperature > 28) {

String serverName = "http://maker.ifttt.com/trigger/" +

String(WebHooksEventName_line) + "/with/key/" + String(WebHooksKey);

String httpRequestData = "value1=" + String(temperature);

Serial.println();

Serial.println("Server Name >> " + serverName);

Serial.println("json httpRequestData >> " + httpRequestData);

if (WiFi.status() == WL_CONNECTED) {

HTTPClient http;

http.begin(serverName);

http.addHeader("Content-Type", "application/x-www-form-urlencoded");

int httpResponseCode = http.POST(httpRequestData);

Serial.print("HTTP Response code: ");

Serial.println(httpResponseCode);

http.end();

if (httpResponseCode == 200)

Serial.println("[Line] --> Successfully sent");

else

Serial.println("[Line] --> Failed!");

}

else {

Serial.println("WiFi Disconnected");
```

```
}
}
/*Display */
int t = int(temperature * 100);
int Tempp2 = (int)temperature/10; int Tempp1 = (int)temperature%10; int Tempp0 =
(int)(temperature*10)%10;
int Humi2 = (int)humidity/10; int Humi1 = (int)humidity%10; int Humi0 =
(int)(humidity*10)%10;
tm.displayHex(0, Tempp2);
tm.displayASCIIwDot(1, Tempp1 + '0'); // turn on dot
tm.displayHex(2, Tempp0);
tm.display7Seg(3, B01011000); // Code=tgfedcba
tm.displayHex(4, Humi2);
tm.displayASCIIwDot(5, Humi1 + '0'); // turn on dot
tm.displayHex(6, Humi0);
tm.display7Seg(7, B01110100); // Code=tgfedcba
delay(2000);
int WaitTime = 10;
Serial.print(" >> Wait for next time --> ");
for (int i = WaitTime; i >= 0; i -= 5) {
Serial.print(",");
Serial.print(i);
delay(5000);
}}}
```


นายปราชญา ธนพิบูลผล
B6323059


นายปราชญา ธนพิบูลผล
B6323059


test
tempp
32.44
0
100
humid
11.98
0
100


LINE Notify
IFTTT: Temperature = 32.20 C 22:14
IFTTT: Temperature = 32.10 C 22:14
IFTTT: Temperature = 32.10 C 22:15
IFTTT: Temperature = 32.10 C 22:15
IFTTT: Temperature = 32.10 C 22:15
IFTTT: Temperature = 32.10 C 22:16
IFTTT: Temperature = 32.10 C 22:16
IFTTT: Temperature = 32.10 C 22:16
IFTTT: Temperature = 32.10 C 22:17
IFTTT: Temperature = 32.10 C 22:17
IFTTT: Temperature = 32.10 C 22:17
IFTTT: Temperature = 32.00 C 22:18
IFTTT: Temperature = 32.00 C 22:18
IFTTT: Temperature = 32.10 C 22:18
IFTTT: Temperature = 32.00 C 22:19
Aa