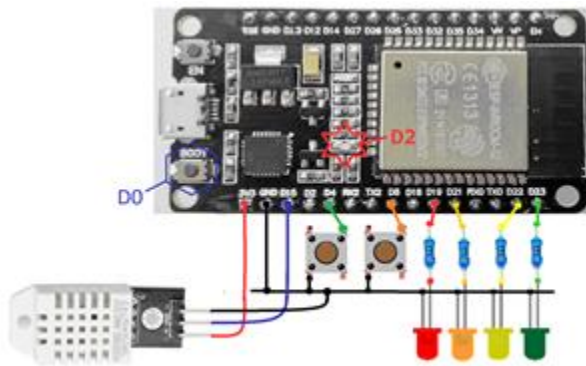


แนวทางการใช้งานอินเทอร์เน็ตของสรรพสิ่งในระบบการผลิต
IoT Approaches to Manufacturing System

ชื่อ-สกุล : นางสาวณัฐชยา ผ่องกุล B6226718

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 = "BOOK";
const char *My_Pass = "book1017";
const char *MQTT_Server = "things.ubidots.com";
const char *MQTT_User = "BBFF-KK5XRLxBJDyhXkQoKFMCDV2S3io619";
const char *MQTT_Pass = "BBFF-KK5XRLxBJDyhXkQoKFMCDV2S3io619";
const char *PTopic1 = "/v2.0/devices/tphappy";
const char *STopic1 = "/v2.0/devices/tphappy/humid";
const char *STopic2 = "/v2.0/devices/tphappy/tempp";
const char *STopic3 = "/v2.0/devices/tphappy/led1";
const char *STopic4 = "/v2.0/devices/tphappy/led2";
const char *STopic5 = "/v2.0/devices/tphappy/led3";
const char *STopic6 = "/v2.0/devices/tphappy/led4";
const char *STopic7 = "/v2.0/devices/tphappy/sw1";
```

```
const char *STopic8 = "/v2.0/devices/tphappy/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();
}

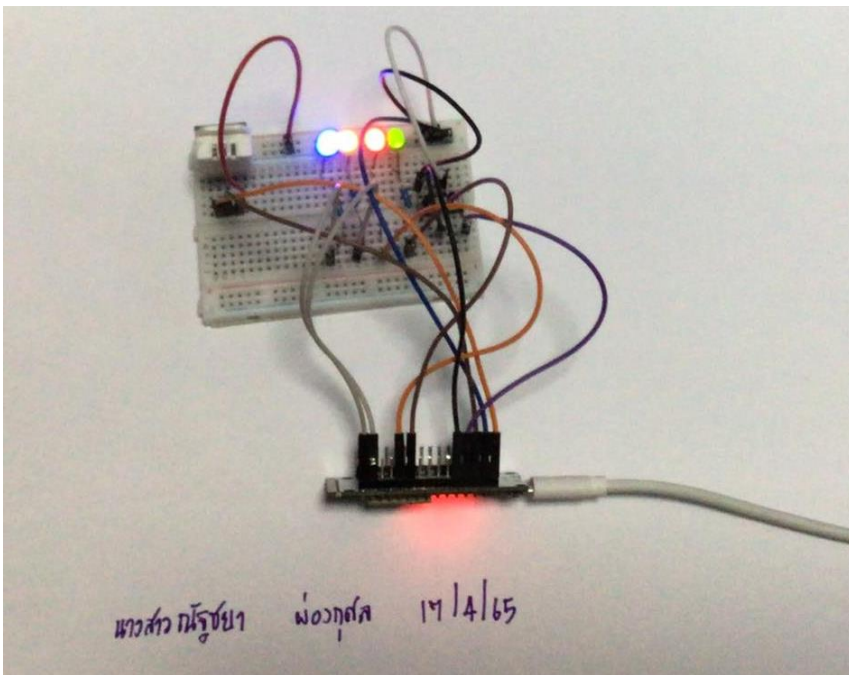
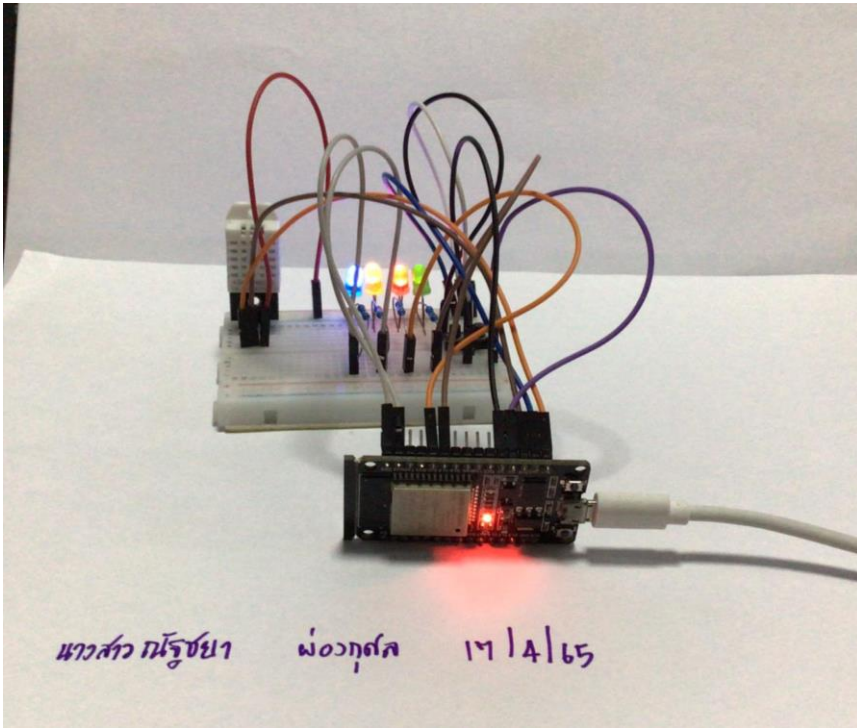
void setup()
{ pinMode(Test_LED1, OUTPUT);
  pinMode(Test_LED2, OUTPUT);
  pinMode(Test_LED3, OUTPUT);
  pinMode(Test_LED4, OUTPUT);
  pinMode(Test_SW1, INPUT_PULLDOWN);
  pinMode(Test_SW2, INPUT_PULLDOWN);
  dht.setup(Pin_DHT22, DHTesp::DHT22);
  Serial.begin(115200);
  Setup_Wifi();
  client.setServer(MQTT_Server, MQTT_Port);
  client.setCallback(callback);
}

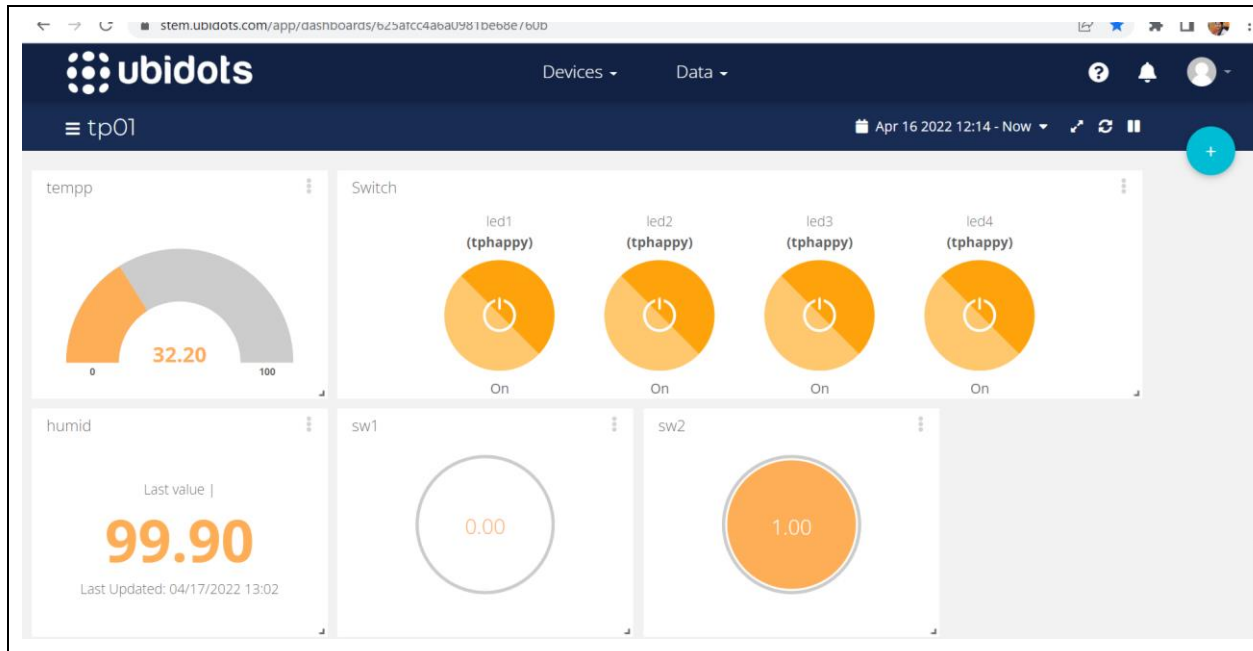
void loop()
{ if (!client.connected()) reconnect();
  client.loop();
  long now = millis();
```

```
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;

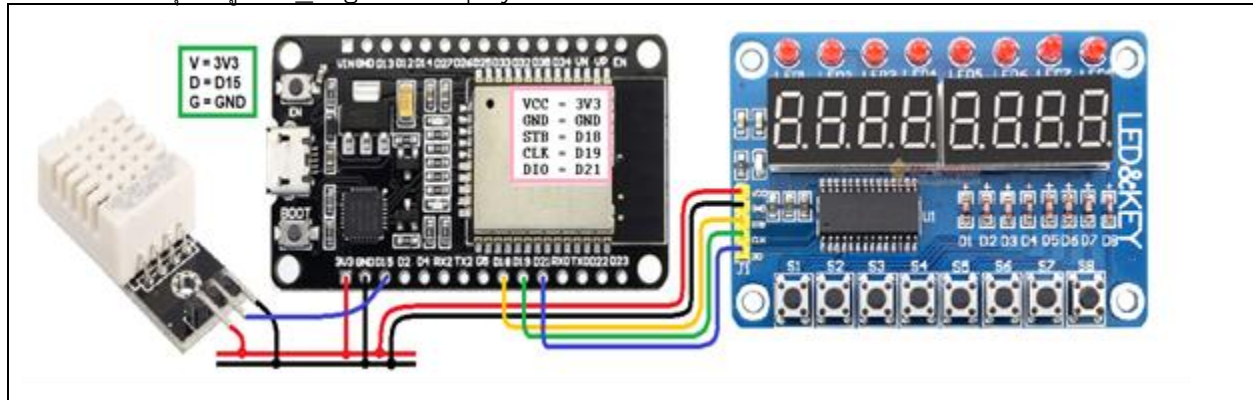
snprintf (msg, 75, "{ \"humid\" : %.2f, \"tempp\": %.2f, \"sw1\": %d, \"sw2\": %d }",
humidity, temperature, sw1, sw2);
Serial.print("Publish message: ");
Serial.println(msg);
client.publish(PTopic1, msg);
} }
```





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 = "BOOK";
const char *My_Pass = "book1017";
const char *MQTT_Server = "things.ubidots.com";
const char *MQTT_User = "BBFF-KK5XRLxBJDyhXkQoKFMCDV2S3io619";
const char *MQTT_Pass = "BBFF-KK5XRLxBJDyhXkQoKFMCDV2S3io619";
#define WebHooksKey "y2TRnqZxW1XjxELy7-rO-"
#define WebHooksEventName "test_GSheet"
#define WebHooksEventName_line "Test_Key"
const char *PTopic1 = "/v2.0/devices/tphappy";
const char *STopic1 = "/v2.0/devices/tphappy/humid";
const char *STopic2 = "/v2.0/devices/tphappy/temp";
#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 "B6226718 Natchaya Phongkuson"
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)

```

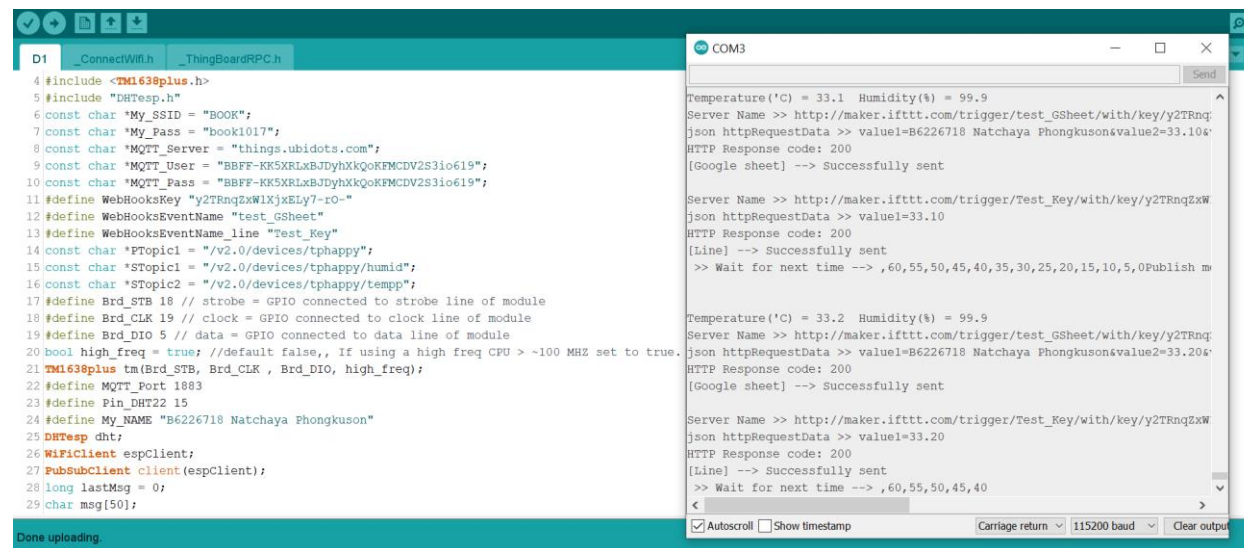
```

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 Temppo =
(int)(temperature*10)%10;
int Humi2 = (int)humidity/10; int Humi1 = (int)humidity%10; int Humio =
(int)(humidity*10)%10;
tm.displayHex(0, Tempp2);
tm.displayASCIIDot(1, Tempp1 + '0'); // turn on dot
tm.displayHex(2, Temppo);
tm.display7Seg(3, B01011000); // Code=tgfedcba
tm.displayHex(4, Humi2);
tm.displayASCIIDot(5, Humi1 + '0'); // turn on dot
tm.displayHex(6, Humio);
tm.display7Seg(7, B01110100); // Code=tgfedcba
delay(2000);

int WaitTime = 60;
Serial.print(" >> Wait for next time --> ");
for (int i = WaitTime; i >= 0; i -= 5) {
Serial.print(",");
Serial.print(i);

```

```
delay(5000);
}}< Test Code >
```



The screenshot shows the Arduino IDE interface. The left pane displays the code for a project named 'D1' with tabs for 'ConnectWiFi.h' and 'ThingBoardRPC.h'. The code includes headers for 'TM1638plus.h' and 'DHT.h', defines MQTT server and user credentials, and sets up a DHT sensor and a WiFi client. The right pane shows the 'COM3' serial monitor with the following output:

```
Temperature('C) = 33.1 Humidity(%) = 99.9
Server Name >> http://maker.ifttt.com/trigger/test_GSheet/with/key/y2TRnq2xW
json httpRequestData >> value1=B6226718 Natchaya Phongkuson&value2=33.10&
HTTP Response code: 200
[Google sheet] --> Successfully sent

Server Name >> http://maker.ifttt.com/trigger/Test_Key/with/key/y2TRnq2xW
json httpRequestData >> value1=33.10
HTTP Response code: 200
[Line] --> Successfully sent
>> Wait for next time --> ,60,55,50,45,40,35,30,25,20,15,10,5,0Publish m

Temperature('C) = 33.2 Humidity(%) = 99.9
Server Name >> http://maker.ifttt.com/trigger/test_GSheet/with/key/y2TRnq2xW
json httpRequestData >> value1=B6226718 Natchaya Phongkuson&value2=33.20&
HTTP Response code: 200
[Google sheet] --> Successfully sent

Server Name >> http://maker.ifttt.com/trigger/Test_Key/with/key/y2TRnq2xW
json httpRequestData >> value1=33.20
HTTP Response code: 200
[Line] --> Successfully sent
>> Wait for next time --> ,60,55,50,45,40
```

At the bottom of the IDE, a status bar indicates 'Done uploading.' and the serial monitor settings are set to 'Autoscroll', 'Show timestamp', 'Carriage return', '115200 baud', and 'Clear output'.

