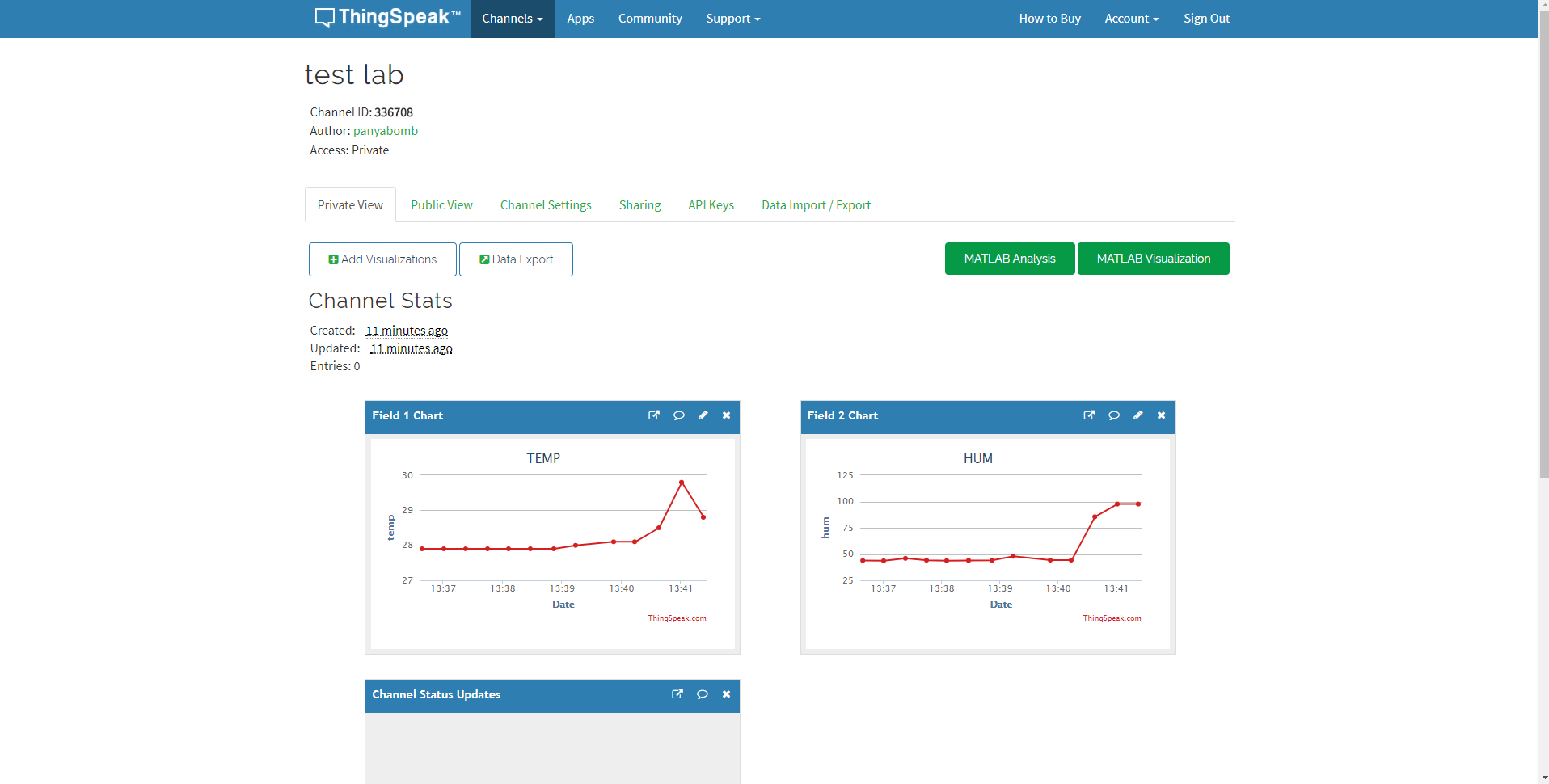
LAB 8 Thingspeak

ภาพการทำงาน



Code

#include <LiquidCrystal\_I2C.h>

#include "DHT.h"

#include <ESP8266WiFi.h>

#define PUMP\_RLY 4 // output drive relay for pump GPIO4 (D2)

#define DHTPIN 2 // what pin we're connected to GPIO2 (D4)

#define DHTTYPE DHT22 // DHT 11

#define DEBUG

#define DEBUG\_PRINTER Serial

#ifdef DEBUG

#define DEBUG\_PRINT(...) { DEBUG\_PRINTER.print(\_\_VA\_ARGS\_\_); }

#define DEBUG\_PRINTLN(...) { DEBUG\_PRINTER.println(\_\_VA\_ARGS\_\_); }

#else

#define DEBUG\_PRINT(...) {}

#define DEBUG\_PRINTLN(...) {}

#endif

const char\* ssid = "FITM WiFi";

const char\* password = "";

DHT \*dht;

void connectWifi();

void reconnectWifiIfLinkDown();

void initDht(DHT \*\*dht, uint8\_t pin, uint8\_t dht\_type);

void readDht(DHT \*dht, float \*temp, float \*humid);

void uploadThingsSpeak(float t, float h);

void setup() {

Serial.begin(115200);

delay(10);

pinMode(PUMP\_RLY, OUTPUT); // Initialize the PUMP\_RLY(4) pin as an output

digitalWrite(PUMP\_RLY, HIGH);// Make sure relay is normal off

connectWifi();

initDht(&dht, DHTPIN, DHTTYPE);}

void loop() {

static float t\_dht;

static float h\_dht;

readDht(dht, &t\_dht, &h\_dht);

if(t\_dht > 29) // condition for make relay on

{

digitalWrite(PUMP\_RLY, HIGH); //If condition true do this!

} else

{

digitalWrite(PUMP\_RLY, LOW);

}

uploadThingsSpeak(t\_dht, h\_dht);

// Wait a few seconds between measurements.

delay(10 \* 1000);

reconnectWifiIfLinkDown();

}

void reconnectWifiIfLinkDown() {

if (WiFi.status() != WL\_CONNECTED) {

DEBUG\_PRINTLN("WIFI DISCONNECTED");

connectWifi();

}

}

void connectWifi() {

DEBUG\_PRINTLN();

DEBUG\_PRINTLN();

DEBUG\_PRINT("Connecting to ");

DEBUG\_PRINTLN(ssid);

WiFi.begin(ssid, password);

while (WiFi.status() != WL\_CONNECTED) {

delay(500);

DEBUG\_PRINT(".");

}

DEBUG\_PRINTLN("");

DEBUG\_PRINTLN("WiFi connected");

DEBUG\_PRINTLN("IP address: ");

DEBUG\_PRINTLN(WiFi.localIP());

}

void initDht(DHT \*\*dht, uint8\_t pin, uint8\_t dht\_type) {

// Connect pin 1 (on the left) of the sensor to +5V

// NOTE: If using a board with 3.3V logic like an Arduino Due connect pin 1

// to 3.3V instead of 5V!

// Connect pin 2 of the sensor to whatever your DHTPIN is

// Connect pin 4 (on the right) of the sensor to GROUND

// Connect a 10K resistor from pin 2 (data) to pin 1 (power) of the sensor

// Initialize DHT sensor for normal 16mhz Arduino

// NOTE: For working with a faster chip, like an Arduino Due or Teensy, you

// might need to increase the threshold for cycle counts considered a 1 or 0.

// You can do this by passing a 3rd parameter for this threshold. It's a bit

// of fiddling to find the right value, but in general the faster the CPU the

// higher the value. The default for a 16mhz AVR is a value of 6. For an

// Arduino Due that runs at 84mhz a value of 30 works.

// Example to initialize DHT sensor for Arduino Due:

//DHT dht(DHTPIN, DHTTYPE, 30);

\*dht = new DHT(pin, dht\_type, 30);

(\*dht)->begin();

DEBUG\_PRINTLN(F("DHTxx test!")) ;

}

void uploadThingsSpeak(float t, float h) {

static const char\* host = "api.thingspeak.com"; /////////////////

static const char\* apiKey = "96GE1TH5YPRL2GRM"; ///////////////////////

// Use WiFiClient class to create TCP connections

WiFiClient client;

const int httpPort = 80;

if (!client.connect(host, httpPort)) {

DEBUG\_PRINTLN("connection failed");

return;

}

// We now create a URI for the request

String url = "/update/";

// url += streamId;

url += "?key=";

url += apiKey;

url += "&field1=";

url += t;

url += "&field2=";

url += h;

DEBUG\_PRINT("Requesting URL: ");

DEBUG\_PRINTLN(url);

// This will send the request to the server

client.print(String("GET ") + url + " HTTP/1.1\r\n" +

"Host: " + host + "\r\n" +

"Connection: close\r\n\r\n");

}

void readDht(DHT \*dht, float \*temp, float \*humid) {

if (dht == NULL) {

DEBUG\_PRINTLN(F("[dht11] is not initialised. please call initDht() first."));

return;

}

// Reading temperature or humidity takes about 250 milliseconds!

// Sensor readings may also be up to 2 seconds 'old' (its a very slow sensor)

float h = dht->readHumidity();

// Read temperature as Celsius

float t = dht->readTemperature();

// Read temperature as Fahrenheit

float f = dht->readTemperature(true);

// Check if any reads failed and exit early (to try again).

if (isnan(h) || isnan(t) || isnan(f)) {

DEBUG\_PRINTLN("Failed to read from DHT sensor!");

return;

}

// Compute heat index

// Must send in temp in Fahrenheit!

float hi = dht->computeHeatIndex(f, h);

DEBUG\_PRINT("Humidity: ");

DEBUG\_PRINT(h);

DEBUG\_PRINT(" %\t");

DEBUG\_PRINT("Temperature: ");

DEBUG\_PRINT(t);

DEBUG\_PRINT(" \*C ");

DEBUG\_PRINT(f);

DEBUG\_PRINT(" \*F\t");

DEBUG\_PRINT("Heat index: ");

DEBUG\_PRINT(hi);

DEBUG\_PRINTLN(" \*F");

\*temp = t;

\*humid = h;

}