

```

#include <FS.h>                //this needs to be first, or it all crashes and burns...
#include <SPIFFS.h> //เพิ่ม

#include <WiFi.h>              //https://github.com/esp8266/Arduino

#include <WiFiClient.h>

#include <iSYNC.h>

WiFiClient client;

iSYNC iSYNC(client);

String iSYNC_USERNAME = "stanutty1123";

String iSYNC_KEY = "5fb2aa195e614c07a2d8a206";

String iSYNC_AUTH = "5fb2aa015e614c07a2d8a204"; //auth project

//needed for library

#include <DNSServer.h>

#include <WebServer.h>

#include <WiFiManager.h>      //https://github.com/tzapu/WiFiManager

#include <ArduinoJson.h>      //Ver 5.13.4

//-----

#include "SparkFun_SGP30_Arduino_Library.h" // 0x58 Click here to get the library:

```

```

#include <Wire.h>
#include <LiquidCrystal_I2C.h>
#include "TinyGPS++.h"
#include "SoftwareSerial.h"
#include "SimpleTimer.h"
#include <TinyGPS++.h>
#include <TridentTD_LineNotify.h>
char line_token1[45] = "";//
char line_token2[45] = "";//
char line_token3[45] = "";//
//----- กำหนดหมายเลขของขาของ Node MCU ESP32 -----
const int LedblynkGreen = 19;
const int LedblynkRed = 18;          // ใช้ไฟ LED สีฟ้า ของบอร์ด MCU ESP32 ให้
มีสัญญาณไฟกระพริบ ตาม Code ที่เขียน
const int AP_Config = 5;            // ใช้เป็นปุ่มกด เพื่อเข้า AP Config ได้ตามความ
ต้องการของผู้ใช้
#define buzzer 23
bool shouldSaveConfig = false;

//callback notifying us of the need to save config
void saveConfigCallback () {
  Serial.println("Should save config");
  shouldSaveConfig = true;
}

LiquidCrystal_I2C lcd(0x3F, 20, 4); //Module IIC/I2C Interface บางรุ่นอาจจะใช้ 0x3f
SGP30 mySensor; //create an object of the SGP30 class
SoftwareSerial serial_connection(16, 17); //RX=pin 10, TX=pin 11
TinyGPSPlus gps; //This is the GPS object that will pretty much do all the grunt
work with the NMEA data

```

```

SimpleTimer timer; // ประกาศให้ฟังก์ชัน ทำงาน
float ppm;
float RS_gasc = 0;
float ratioc = 0;
float sensorValuec = 0;
float sensor_voltc = 0;
float R0c = 7200.0;
float xc=0;

float sensor_volt;
float RS_gas; // Get value of RS in a GAS
float R0 = 15000; //example value of R0. Replace with your own
float ratio; // Get ratio RS_GAS/RS_air
float LPG_PPM;
float x=0;

//                      ฟังก์ชัน MQTT
void callback(char* topic, byte* payload, unsigned int length) {
  String msg = "";
  Serial.print("[iSYNC]-> ");
  for (int i = 0; i < length; i++){
    msg+=(char)payload[i];
    Serial.print((char)payload[i]);
  }
  Serial.println();
  if(msg.startsWith("LINE:"))msg = msg.substring(5);
  if(msg.equals("co")){
    iSYNC.mqPub(iSYNC_KEY,"CO ตอนนี้ปริมาณ "+String(ppm)+" ppm ถ้ายังไม่ถึง 30-35ppm ถือว่ายังปลอดภัยครับ"); //Publish
  }else if(msg.equals("lpg")){

```

```

        iSYNC.mqPub(iSYNC_KEY,"LPG ตอนนี้มีปริมาณ "+String(LPG_PPM)+" ppb ครับ ถ้ายังไม่ถึง 500ppb ถือว่ายังปลอดภัยครับ"); //Publish
    }else if(msg.equals("co2")){
        iSYNC.mqPub(iSYNC_KEY,"Co2 ตอนนี้มีปริมาณ "+String(mySensor.CO2)+" ppm ครับ ถ้ายังไม่ถึง 1500-1800ppm ถือว่ายังปลอดภัยครับ"); //Publish
    }else if(msg.equals("tvoc")){
        iSYNC.mqPub(iSYNC_KEY,"TVOCS ตอนนี้มีปริมาณ "+String(mySensor.TVOC)+" ppb ครับ"); //Publish
    }else if(msg.equals("พิกัด")){
        iSYNC.mqPub(iSYNC_KEY,"ตอนนี้คุณอยู่ที่ "+String(gps.location.lat(), 6)+" , "+String(gps.location.lng(), 6)+" ครับ ถ้ายังไม่ถึง 1700-2008ppm ถือว่ายังปลอดภัยครับ");
        //Publish
    }else if(msg.equals("!พิกัด!!co!!tvoc!!co2!!lpg")){
        iSYNC.mqPub(iSYNC_KEY,"กรุณากรอกใหม่อีกครั้งครับ เราไม่ทราบว่าคุณหมายถึงอะไร");
    }
}

//          Connect MQTT การเชื่อมต่อกับ Mqtt
void connectMQTT(){
    while(!iSYNC.mqConnect()){
        Serial.println("Reconnect MQTT...");
        delay(3000);
    }
    iSYNC.mqPub(iSYNC_KEY,"พร้อมรับคำสั่งแล้วจะพีจ๋า"); //Publish on Connect
    // iSYNC.mqSubProject(); //subscribe all key in your project
    iSYNC.mqSub(iSYNC_KEY); //subscribe key
}
}

```

```

void setup() {
  //-----IO NODE MCU Esp32-----//
  // ให้ LED ทั้งหมดดับก่อน
  pinMode(buzzer, OUTPUT);
  pinMode(LedblynkGreen, OUTPUT);
  pinMode(LedblynkRed, OUTPUT);    //กำหนดโหมดใช้งานให้กับขา Ledblynk เป็นขา
สัญญาณไฟ ในสถานะต่างๆ
  pinMode(AP_Config, INPUT_PULLUP);
  digitalWrite(LedblynkRed, LOW);//ให้หลอด LED สีฟ้าดับก่อน
  digitalWrite(LedblynkGreen, LOW);//ให้หลอด LED สีฟ้าดับก่อน
  digitalWrite(buzzer, HIGH);
  //-----//
  Serial.begin(115200);
  //-----//
  timer.setInterval(3000, readSensor); // ตั้งเวลาให้อ่านทุก 3 วิ
  timer.setInterval(60000, toline); // ตั้งเวลาให้อ่านทุก 60 วิ
  timer.setInterval(500,recon);
  timer.setInterval(500,calco);
  timer.setInterval(500,callpg);
  lcd.begin();
  lcd.backlight();
  Wire.begin();
  serial_connection.begin(9600);//This opens up communications to the GPS
  //*****          การ อ่าน เขียนค่า WiFi + Password ]ลงใน Node MCU
ESP32  *****//
  //read configuration from FS json
  Serial.println("mounting FS...");//แสดงข้อความใน Serial Monitor

  if (SPIFFS.begin(true)) {

```

```

Serial.println("mounted file system");
if (SPIFFS.exists("/config.json")) {
    //file exists, reading and loading
    Serial.println("reading config file");
    File configFile = SPIFFS.open("/config.json", "r");
    if (configFile) {
        Serial.println("opened config file");
        size_t size = configFile.size();
        // Allocate a buffer to store contents of the file.
        std::unique_ptr<char[]> buf(new char[size]);

        configFile.readBytes(buf.get(), size);
        DynamicJsonBuffer jsonBuffer;
        JsonObject& json = jsonBuffer.parseObject(buf.get());
        json.printTo(Serial);
        if (json.success()) {
            Serial.println("\nparsed json");
            strcpy(line_token1, json["line_token1"]);
            strcpy(line_token2, json["line_token2"]);
            strcpy(line_token3, json["line_token3"]);
        } else {
            Serial.println("failed to load json config");//แสดงข้อความใน Serial Monitor
        }
    }
}
} else {
    Serial.println("failed to mount FS");//แสดงข้อความใน Serial Monitor
}
//end read

```

```

Serial.println(line_token1);

//***** จอการ อ่าน เขียนค่า WiFi + Password ]ลงใน Node MCU
ESP32 *****//

// การตั้งค่าช่องการใส่โทเค้น

WiFiManagerParameter custom_text0("<p> </p>");
WiFiManagerParameter custom_text1("<label>1: ID Line 1 </label>");
WiFiManagerParameter custom_text2("<label>2: ID Line 2 </label>");
WiFiManagerParameter custom_text3("<label>3: ID Line 3 </label>");

//***** AP AUTO CONNECT *****

WiFiManagerParameter custom_line_token1("line_token1", "line_token1",
line_token1, 45);
WiFiManagerParameter custom_line_token2("line_token2", "line_token2",
line_token2, 45);
WiFiManagerParameter custom_line_token3("line_token3", "line_token3",
line_token3, 45);
//WiFiManager
//Local intialization. Once its business is done, there is no need to keep it around
WiFiManager wifiManager;

//set config save notify callback
wifiManager.setSaveConfigCallback(saveConfigCallback);

wifiManager.addParameter(&custom_text0);

```

```

wifiManager.AddParameter(&custom_text1);
wifiManager.AddParameter(&custom_line_token1);

wifiManager.AddParameter(&custom_text0);
wifiManager.AddParameter(&custom_text2);
wifiManager.AddParameter(&custom_line_token2);

wifiManager.AddParameter(&custom_text0);
wifiManager.AddParameter(&custom_text3);
wifiManager.AddParameter(&custom_line_token3);

for (int i = 5; i > -1; i--) { // นับเวลาถอยหลัง 5 วินาทีก่อนกดปุ่ม AP Config
    digitalWrite(LedblynkRed, HIGH);
    delay(500);
    digitalWrite(LedblynkRed, LOW);
    delay(500);
    lcd.setCursor(1,1);
    lcd.print("Wait for Config : " +String(i));
    Serial.print (String(i) + " ");//แสดงข้อความใน Serial Monitor
}

if (digitalRead(AP_Config) == LOW) {
    digitalWrite(LedblynkRed, HIGH);
    Serial.println("Button Pressed");//แสดงข้อความใน Serial Monitor
    lcd.setCursor(1,2);
    lcd.print(" กดปุ่มค้าง 5วิ ");

    // wifiManager.resetSettings();//ให้ล้างค่า SSID และ Password ที่เคยบันทึกไว้

```



```

wifi_init_config_t cfg = WIFI_INIT_CONFIG_DEFAULT(); //load the flash-saved
configs

esp_wifi_init(&cfg); //initiate and allocate wifi resources (does not matter if
connection fails)

delay(2000); //wait a bit

if (esp_wifi_restore() != ESP_OK)

{
    Serial.println("WiFi is not initialized by esp_wifi_init ");
} else {
    Serial.println("WiFi Configurations Cleared!");
}

//continue
//delay(1000);

//esp_restart(); //just my reset configs routine...
}

wifiManager.setTimeout(90);
//ใช้ได้ 2 กรณี
//1. เมื่อกดปุ่มเพื่อ Config ค่า AP แล้ว จะขึ้นชื่อ AP ที่เรที่ตั้งขึ้น
// ช่วงนี้ให้เราทำการตั้งค่า SSID+Password หรืออื่นๆทั้งหมด ภายใน 60 วินาที ก่อน AP จะ
หมดเวลา
// ไม่เช่นนั้น เมื่อครบเวลา 60 วินาที MCU จะ Reset เริ่มต้นใหม่ ให้เราตั้งค่าอีกครั้งภายใน
60 วินาที
//2. ช่วงไฟดับ Modem router + MCU จะดับทั้งคู่ และเมื่อมีไฟมา ทั้งคู่ก็เริ่มทำงานเช่นกัน
// โดยปกติ Modem router จะ Boot ช้ากว่า MCU ทำให้ MCU กลับไปเป็น AP รอให้เรา
ตั้งค่าใหม่
// ดังนั้น AP จะรอเวลาให้เราตั้งค่า 60 วินาที ถ้าไม่มีการตั้งค่าใดๆ เมื่อครบ 60 วินาที MCU
จะ Reset อีกครั้ง

```

```

// ถ้า Modem router Boot และใช้งานได้ภายใน 60 วินาที และหลังจากที่ MCU Reset
และเริ่มทำงานใหม่
// ก็จะสามารถเชื่อมต่อกับ Modem router ที่ Boot และใช้งานได้แล้ว ได้ ระบบจะทำงาน
ปกติ

if (!wifiManager.autoConnect("PUYIOT ESP32 AP CONFIG")) {
    Serial.println("failed to connect and hit timeout");//แสดงข้อความใน Serial Monitor
    delay(3000);
    //reset and try again, or maybe put it to deep sleep
    ESP.restart();//แก้ เดิม ESP.reset(); ใน Esp8266
    delay(5000);
}

Serial.println("Connected.....OK!");//แสดงข้อความใน Serial Monitor
strcpy(line_token1, custom_line_token1.getValue());
strcpy(line_token2, custom_line_token2.getValue());
strcpy(line_token3, custom_line_token3.getValue());
//save the custom parameters to FS
if (shouldSaveConfig) {
    Serial.println("saving config");
    DynamicJsonBuffer jsonBuffer;
    JsonObject& json = jsonBuffer.createObject();

    json["line_token1"] = line_token1;
    File configFile = SPIFFS.open("/config.json", "w");

    json["line_token2"] = line_token2;
    File configFile2 = SPIFFS.open("/config.json", "w");

```

```

    json["line_token3"] = line_token3;
    File configFile3 = SPIFFS.open("/config.json", "w");
    if (!configFile) {
        Serial.println("failed to open config file for writing");//แสดงข้อความใน Serial
Monitor
    }
    json.printTo(Serial);
    json.printTo(configFile);
    configFile.close();

    json.printTo(configFile2);
    configFile2.close();

    json.printTo(configFile3);
    configFile3.close();
    //end save
}

//***** จบ AP AUTO CONNECT **//

if (mySensor.begin() == false) {
    Serial.println("No SGP30 Detected. Check connections.");
    while (1);
}
//Initializes sensor for air quality readings
//measureAirQuality should be called in one second increments after a call to
initAirQuality
mySensor.initAirQuality();

```

```

iSYNC.mqInit(iSYNC_USERNAME,iSYNC_AUTH);
    iSYNC.MQTT->setCallback(callback);
    connectMQTT();

Serial.println("local ip"); //แสดงข้อความใน Serial Monitor
delay(100);
Serial.println(WiFi.localIP()); //แสดงข้อความใน Serial Monitor
Serial.println(WiFi.gatewayIP());
Serial.println(WiFi.subnetMask());

Serial.println(line_token1);
Serial.println(line_token2);
Serial.println(line_token3);
lcd.clear();
}
//***** จบ void setup
//          ฟังก์ชันคำนวณ sensor MQ7
void calco(){
    sensorValuec = analogRead(A5);
    sensor_voltc = sensorValuec/1024*5.0;
    RS_gasc = (5.0-sensor_voltc)/sensor_voltc;
    ratioc = RS_gasc/R0c; //Replace R0 with the value found using the sketch above
    xc = 1538.46 * ratioc;
    ppm = pow(x,-1.709);
}
//          ฟังก์ชัน MQ6
void callpg(){
    int sensorValue = analogRead(A0);

```

```

    sensor_volt=(float)sensorValue/1024*5.0;
    RS_gas = (5.0-sensor_volt)/sensor_volt;
    ratio = RS_gas/R0;
    x = 1000*ratio ;
    LPG_PPM = pow(x,-1.431); //LPG PPM
  }
void loop() {

    while(serial_connection.available())//While there are characters to come from
the GPS
    {
        gps.encode(serial_connection.read()); //This feeds the serial NMEA data into the
library one char at a time
    }
    if(gps.location.isUpdated())//This will pretty much be fired all the time anyway
but will at least reduce it to only after a package of NMEA data comes in
    {
        //Get the latest info from the gps object which it derived from the data sent by
the GPS unit
        Serial.println("Satellite Count:");
        Serial.println(gps.satellites.value());
        Serial.println("Latitude:");
        Serial.println(gps.location.lat(), 6); // ส่งค่า ลติจูด
        Serial.println("Longitude:");
        Serial.println(gps.location.lng(), 6); //ส่งค่าลองจิจูด
        Serial.println("Speed MPH:");
        Serial.println(gps.speed.mph());
        Serial.println("Altitude Feet:");
        Serial.println(gps.altitude.feet());
    }
}

```

```

Serial.println("");
}

//----- ส่วนของการแจ้งเตือนเมื่อแก๊สเกินกำหนดทางอุปกรณ์ -----

if(mySensor.CO2>1500 || mySensor.TVOC>2000 ||LPG_PPM>500 || ppm>30){
digitalWrite(buzzer, LOW);
delay(500);
digitalWrite(buzzer, HIGH);
delay(500);
digitalWrite(LedblynkRed, HIGH);
digitalWrite(LedblynkGreen, LOW);
// lcd.setCursor(0,3);lcd.print(" STATUS : UNSAFE  ");
}else{
digitalWrite(buzzer, HIGH);
digitalWrite(LedblynkGreen, HIGH);
digitalWrite(LedblynkRed, LOW);
// lcd.setCursor(0,3);lcd.print(" STATUS : SAFE  ");
}

//----- ส่วนของการแจ้งเตือนเมื่อแก๊สเกินกำหนดไปที่ line -----

if(mySensor.CO2>1500 || mySensor.TVOC>2000 ||LPG_PPM>500 || ppm>32){
LINE.setToken(line_token1);
LINE.notify("Longitude: "+String(gps.location.lng(), 6));
LINE.notify("Latitude: "+String(gps.location.lat(), 6));
LINE.notify(" CO2: "+String (mySensor.CO2)+" ppm");
LINE.notify(" TVOC: "+String (mySensor.TVOC)+" ppb");
LINE.notify(" LPG: "+String (LPG_PPM) +" ppb");
LINE.notify(" CO: "+String (ppm) +" ppm");
LINE.notify("อันตรายมีแก๊สเกินกำหนด");
}

```

```

LINE.setToken(line_token2);

    LINE.notify("Longitude: "+String(gps.location.lng(), 6));
    LINE.notify("Latitude: "+String(gps.location.lat(), 6));
    LINE.notify(" CO2 : "+String (mySensor.CO2)+" ppm");
    LINE.notify(" TVOC: "+String (mySensor.TVOC)+" ppb");
    LINE.notify(" LPG: "+String (LPG_PPM) +" ppb");
    LINE.notify(" CO: "+String (ppm) +" ppm");
    LINE.notify("อันตรายมีแก๊สเกินกำหนด");

    LINE.setToken(line_token3);

    LINE.notify("Longitude: "+String(gps.location.lng(), 6));
    LINE.notify("Latitude: "+String(gps.location.lat(), 6));
    LINE.notify(" CO2 : "+String (mySensor.CO2)+" ppm");
    LINE.notify(" TVOC: "+String (mySensor.TVOC)+" ppb");
    LINE.notify(" LPG: "+String (LPG_PPM) +" ppb");
    LINE.notify(" CO: "+String (ppm) +" ppm");
    LINE.notify("อันตรายมีแก๊สเกินกำหนด");

}

timer.run();

}

// ----- ฟังก์ชันการรีการเชื่อมต่อระหว่าง Mqtt -----

void recon(){
    if (!iSYNC.mqConnected())connectMQTT();
    iSYNC.mqLoop();
}

```

```
//          ฟังก์ชันการอ่านค่า
void readSensor() {

    //First fifteen readings will be
    //CO2: 400 ppm  TVOC: 0 ppb
    //measure CO2 and TVOC levels

    mySensor.measureAirQuality();
    Serial.print("CO2: ");                //แสดงค่า co2
    Serial.print(mySensor.CO2);
    Serial.print(" ppm\tTVOC: ");        //แสดงค่า tvoc
    Serial.print(mySensor.TVOC);
    Serial.println(" ppb");
    Serial.print("LPG PPM = ");          //แสดงค่า lpg
    Serial.println(LPG_PPM);
    Serial.print("CO = ");
    Serial.println(ppm);
    lcd.setCursor(0,0);
    lcd.print(" CO2: ");
    lcd.print(mySensor.CO2);
    lcd.print(" ppm");lcd.print("");
    lcd.setCursor(0,1);
    lcd.print(" TVOC: ");
    lcd.print(mySensor.TVOC);
    lcd.print(" ppb");lcd.print("");
    lcd.setCursor(0,2);
    lcd.print(" LPG : "); lcd.print(LPG_PPM); lcd.print(" ppb");
    lcd.setCursor(0,3);
```



```

    lcd.print(" CO :"); lcd.print(ppm); lcd.print(" ppm");
}
// ----- ฟังก์ชันการส่งค่าไปที่ line -----

void toline() //timer notify 60min
{
    // การแจ้งเตือน line notify
    if(mySensor.CO2<1500 || mySensor.TVOC<2000 ||LPG_PPM<500||ppm<32 ){
    LINE.setToken(line_token1); LINE.notify("ปลอดภัยดี");
    LINE.notify("Longitude: "+String(gps.location.lng(), 6));
    LINE.notify("Latitude: "+String(gps.location.lat(), 6));
    LINE.notify(" CO2: "+String (mySensor.CO2)+" ppm");
    LINE.notify(" TVOC: "+String (mySensor.TVOC)+" ppb");
    LINE.notify(" LPG: "+String (LPG_PPM) +" ppb");
    LINE.notify(" CO: "+String (ppm) +" ppm");

    LINE.setToken(line_token2); LINE.notify("ปลอดภัยดี");
    LINE.notify("Longitude: "+String(gps.location.lng(), 6));
    LINE.notify("Latitude: "+String(gps.location.lat(), 6));
    LINE.notify(" CO2: "+String (mySensor.CO2)+" ppm");
    LINE.notify(" TVOC: "+String (mySensor.TVOC)+" ppb");
    LINE.notify(" LPG: "+String (LPG_PPM) +" ppb");
    LINE.notify(" CO: "+String (ppm) +" ppm");

    LINE.setToken(line_token3); LINE.notify("ปลอดภัยดี");
    LINE.notify("Longitude: "+String(gps.location.lng(), 6));
    LINE.notify("Latitude: "+String(gps.location.lat(), 6));

```

