試験2：「2020　M5stackとSigfoxを利用したambientによるグラフ化のスケッチ。」

/\*

1.M5stackBasic+sigfoxモジュールにDht22とCapacitiveSoilMoistureSensorとDs18b20を5v入力で動作確認。3vは0.5vしかなくエラーに注意。

2.CapacitiveSoilMoistureSensorは電線の長さ（抵抗）で値が変化する。

arduinoをPCにインストールしスケッチを描く準備をする。

3.arduinoで必要なボードをボードマネージャーで登録して、ライブラリーをインクルードする。

4.ambientに登録しチャネルIDとライトキーを取得する

5.sigfoxに登録しグループ・ユーザー・デバイスタイプ・デバイスの管理を行い、デバイスIDとPACを登録する。

参照HP：M5stack COM.X Sigfoxの使い方(kccs-iot.jp/20200918-technical/

6.下記スケッチのID・パス等必要な個所を上書き修正しボードに書き込む。

7.sigfoxの16進数データは偶数でないと受信できない。

本プログラムのすべて、または一部の利用により被った損害等について、当方は一切の責任を負いかねます。利用者本人の責任において利用してください。

\*/

#include <M5Stack.h>

#include "DHT.h" //DHT22のライブラリを指定

#define DHTPIN 13 // BasicでDHT22 をつなぐピン

#define DHTTYPE DHT22

DHT dht(DHTPIN, DHTTYPE);

#include <OneWire.h> //DS18B20のライブラリ

#include <DallasTemperature.h> //DS18B20 のライブラリ

#define ONE\_WIRE\_BUS 0 // BasicでDS18B20 をつなぐピン 、4.7ｋΩ\_pin21\_DS18B20の順に結線

OneWire oneWire(ONE\_WIRE\_BUS); //　OneWireのセットアップ

DallasTemperature sensors(&oneWire);// Dallas Temperature sensor　のセット

#define PinSoilMoist 35 //CapSoilMoistSensorのセット

const int dry = 3700; //土壌水分”0％”の値

const int wet = 1400; //土壌水分”100％”の値

int inttemp , inthumid ; //温湿度の変数（整数）を設定

int intsoilmoist , soilval , intsoiltemp; //土壌水分・地温の変数（整数）を設定

float temperature, humidity, soilMoist, soilTemp; //温湿度・土壌水分・地温の変数（小数)を設定

void getData() {

delay(2000);

dht.begin();

float temperature = dht.readTemperature();

float humidity = dht.readHumidity();

if (isnan(temperature) || isnan(humidity) ) {

Serial.println(F("Failed to read from DHT sensor!"));

return;

}

inttemp = int ( temperature ); //温度の変数(整数）を設定

inthumid = int ( humidity );

Serial.print( inttemp ); Serial.println(" \*C, "); //温度の変数(整数）をPCに表示

Serial.print( inthumid ); Serial.println(" RH% ");

soilMoist = analogRead(PinSoilMoist); // 土壌水分をアナログ測定

soilval = map(soilMoist, dry, wet, 0, 100); //上記の0％=AirValueと100％=WaterValueを調整

if(soilval >= 99)

{

intsoilmoist = 99;

}

else if(soilval <=20)

{

intsoilmoist = 20;

}

else if(soilval >21 && intsoilmoist < 99)

{

intsoilmoist = soilval ; //土壌水分の変数(整数）を設定

}

// Serial.print( soilMoist ); Serial.println(" Pin "); //土壌水分の変数(整数）をPCに表示

Serial.print( intsoilmoist ); Serial.println(" soil% "); //土壌水分の変数(整数）をPCに表示

delay(250);

sensors.requestTemperatures(); // DS18B20 センサーで地温を測定

float soilTemp = sensors.getTempCByIndex(0);

intsoiltemp = int ( soilTemp ); //地温の変数(整数）を設定

Serial.print( intsoiltemp ); Serial.println(" soil\*C, "); //土壌水分の変数(整数）をPCに表示

}

void sendSigfox() {

String msg = String (inttemp, HEX) //整数値を、16進文字列に変換

+ String (inthumid, HEX)

+ String (intsoilmoist, HEX)

+ String (intsoiltemp, HEX);

//Sigfoxでデータを送る場合は、"AT$SF="の後に最大24文字（12バイト）で送る

Serial2.println("AT$SF=" + msg);

Serial.println("AT$SF=" + msg);

}

void setup() {

M5.begin(true, false, true); //M5stackを設定

M5.Speaker.write(0); // Basicでスピーカーを無効にする＝0

Serial2.begin(9600, SERIAL\_8N1, 16, 17); // Basicでsigfox通信の設定

Serial.begin(9600); //PCとの通信設定

}

void loop() {

getData();

sendSigfox();

M5.Power.deepSleep(SLEEP\_SEC(60\*15)); // BasicでDeepsleepの設定

/\*backend.sigfox.comのcallbackｓの設定

Type: data/uplink

channel: URL

custom

payload

config: temp::uint:8 humid::uint:8 temp2::uint:8 humid2::uint:8

Url

pattern: http://ambidata.io/api/v2/channels/チャネルID/data

UseHTTP

Method: POST

Send SNI:チェック入れる

Content

type: application/json

Body:

{

"writeKey":"ライトキー",

"deviceID":"{device}",　　以下書き換え無し

"time":"{ time}",

"seqNumber":"{seqNumber}",

"d1":"{customData#temp}",

"d2":"{customData#humid}",

"d3":"{customData#humid2}",

"d4":"{customData#temp2}"

}

\*/