試験1：「2020　ESP32ボードとLoRaを利用したambientによるグラフ化のスケッチ。」

/\*

 1.　arduinoをPCにインストールしスケッチを描く準備をする。

 2.　arduinoで必要なライブラリーをインクルードして、ボードをマネージャーから登録する。

 3.　ambientに登録しチャネルIDとライトキーを取得する

 4.　下記スケッチの必要な個所を上書き修正しボードに書き込む。

 5. 引用スケッチ： https://github.com/Heltec-Aaron-Lee/WiFi\_kit\_series　、SimpleDHT　、

DallasTemperature sensors　、Ambient

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。

 \*/

送信機側のスケッチ

#include "heltec.h"

#include <SimpleDHT.h>

int pinDHT22 = 23; //23.17が可

SimpleDHT22 dht22(pinDHT22);

#include <OneWire.h> //DS18B20のライブラリ

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

const int oneWireBus = 17; // DS18B20 をつなぐピン 16.22.21.17

OneWire oneWire(oneWireBus); //　OneWireのセットアップ

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

int PinSoilMoist = 36; //CapSoilMoistSensorのセット　ピン25.33.32.39.38.37.36

float temperature, humidity, soilMoist, soilTemp;

#define BAND 915E6 //you can set band here directly,e.g. 868E6,915E6

int counter = 0;

//DHTとCapSoilMoistSensorとDS18B20の変数を設定

String readingID, strTemp, strHumi, strSoilmoist, strSoiltemp, strMeasure;

void getTemHumSoi() {

 //DHT22で温湿度測定

 int err = SimpleDHTErrSuccess;

 if ((err = dht22.read2(&temperature, &humidity, NULL)) != SimpleDHTErrSuccess) {

 Serial.print("Read DHT22 failed, err="); Serial.println(err); delay(2000);

 return;

 }

 // CapSoilMoistSensorでアナログ測定

 soilMoist = analogRead(PinSoilMoist);

 soilMoist = map(soilMoist, 2750, 1110, 0, 100); //0％：2700＝100％：1110を参考に調整

 // DS18B20 センサーを開始

 sensors.begin();

 sensors.requestTemperatures();

 float soilTemp = sensors.getTempCByIndex(0);

 //測定値を一括化

 strMeasure = (readingID + "/" + temperature + "&" + humidity + "#" + soilMoist + "%" + soilTemp + "!");

 Serial.print( strMeasure );

 Serial.print("Sending packet: ");

 Serial.println(counter);

}

void sendLora() {

 //LoRa送信

 LoRa.beginPacket();

 LoRa.setTxPower(14, RF\_PACONFIG\_PASELECT\_PABOOST);

 LoRa.println( strMeasure );

 LoRa.print(counter);

 LoRa.endPacket();

 counter++;

}

void setup() {

 Serial.begin(115200);

 //WIFI Kit series V1 not support Vext control

 Heltec.begin(true /\*DisplayEnable Enable\*/, true /\*Heltec.LoRa Disable\*/, true /\*Serial Enable\*/, true /\*PABOOST Enable\*/, BAND /\*long BAND\*/);

}

void loop() {

 // start working...

 Serial.println("=================================");

 getTemHumSoi();

 sendLora();

 delay(3000);

}

受信機側のスケッチ

#include "heltec.h"

//Ambient(Blynk)のライブラリ

#include <Ambient.h> // Ambient.h をインクルード

//#include <BlynkSimpleEsp32.h>

//Wifiのライブラリ

#include <WiFi.h>

#include <WiFiClient.h>

#include <stdlib.h>

#define PERIOD 5 // delay の値を指定 (例：30 -> 30秒間間隔でデーターをAmbientに送信)

WiFiClient client; // WiFiClientを使うための設定(インスタンス生成)

Ambient ambient; // Ambientを使うための設定(インスタンス生成)

//LoRaの周波数を設定

#define BAND 915E6 //you can set band here directly,e.g. 868E6,915E6

//sendSenserの変数を設定

String readingID, temp, humid, soilmoi, soiltem;

float fltTemp, fltHumi, fltSoilmoi, fltSoiltem;

//Wifiの設定

char ssid[] = "SSID";　//　WifiのSSIDを指定

char pass[] = "PASS";　//　　　　PASSを指定

void startWifi() {

 WiFi.begin(ssid, pass); // Wi-Fiの初期化

 Serial.println("WiFi.begin");

 while (WiFi.status() != WL\_CONNECTED) { // Wi-Fiアクセスポイントへの接続待ち

 Serial.print(".");

 delay(500);

 }

 Serial.print("/r/nWi-Fi connected/r/nIP address: ");

 Serial.println(WiFi.localIP());

}

void startAmbient() {

 unsigned int channelId = チャネルID; // AmbientのチャネルID指定

 const char\* writeKey = "ライトキー"; // ライトキー指定

 ambient.begin(channelId, writeKey, &client);

}

void getLoRaData() {

 // try to parse packet

 int packetSize = LoRa.parsePacket();

 if (packetSize) {

 // received a packet

 Serial.print("Received packet '");

 // read packet

 while (LoRa.available()) {

 //Serial.print((char)LoRa.read()); //charだと文字でシリアル表示、Stringだと数列でシリアル表示。

 String LoRaData = LoRa.readString();

 int pos1 = LoRaData.indexOf('/');

 int pos2 = LoRaData.indexOf('&');

 int pos3 = LoRaData.indexOf('#');

 int pos4 = LoRaData.indexOf('%');

 int pos5 = LoRaData.indexOf('!');

 temp = LoRaData.substring(pos1 + 1, pos2 - 1);

 humid = LoRaData.substring(pos2 + 1, pos3 - 1);

 soilmoi = LoRaData.substring(pos3 + 1, pos4 - 1);

 soiltem = LoRaData.substring(pos4 + 1, pos5 - 1);

 }

 }

}

//数列を数字（小数点込み）に換算

void conversionSensor() {

 fltTemp = temp.toFloat();

 fltHumi = humid.toFloat();

 fltSoilmoi = soilmoi.toFloat();

 fltSoiltem = soiltem.toFloat();

}

void writeAmbient() {

 ambient.set(1, fltTemp); // 1番目のデータに温度を指定。

 //Serial.println("ambient.set1");

 ambient.set(2, fltHumi);

 ambient.set(3, fltSoilmoi);

 ambient.set(4, fltSoiltem);

 ambient.send(); // Ambientにデータを送信

 //Serial.println("ambient.send");

 delay(PERIOD \* 1000); // \*ミリ秒間待つ

}

void printSerial() {

 Serial.println("=================================");

 Serial.println(fltTemp);

 Serial.println(fltHumi);

 Serial.println(fltSoilmoi);

 Serial.println(fltSoiltem);

 Serial.print("' with RSSI ");

 Serial.println(LoRa.packetRssi());}

void setup() {

 // WIFI Kit series V1 not support Vext control

 Heltec.begin(true /\*DisplayEnable Enable\*/, true /\*Heltec.LoRa Disable\*/, true /\*Serial Enable\*/, true /\*PABOOST Enable\*/, BAND /\*long BAND\*/);

 Serial.begin(115200);

 startWifi();

 //Serial.println("startWifi");

 startAmbient();

 //Serial.println("startAmbient");

 // startBlynk();}

void loop() {

 getLoRaData();

 //Serial.println("getLoRaData");

 conversionSensor();

 //Serial.println("conversionSensor");

 writeAmbient();

 //Serial.println("writeAmbient");

 // writeBlynk();

 printSerial();

 delay(5000) ;}