//Feather

#include <HardwareSerial.h>

#include <WiFi.h>

#include "ThingSpeak.h"

const char\* ssid="SENGEEE8092GW"; //Type wifi name

const char\* password="pin-9201.double";//TYpe wifi password

HardwareSerial SerialPort(2); //use UART2

#define UART\_TX\_PIN 43

#define UART\_RX\_PIN 44

#define BAUD\_RATE 115200

int led = 13;

typedef struct UART\_message {

    int id;

    float b;

    float c;

    float d;

    float e;

    float f;

    float g;

    float h;

    float i;

    uint8\_t CRC\_checksum; //CRC for error checking

} UART\_message;

WiFiClient client;

unsigned long myChannelNumber = 1; //The channel number in ThingSpeak (e.g.Channel 1)

const char\* myWriteAPIKey = "Q4DC7LHJQ4KM4ETF"; //Write API Key, can be obtained from ThingSpeak,differenct channel has different API keyTimer variables

unsigned long lastTime = 0;

unsigned long timerDelay = 5000;

void setup() {

  Serial.begin(BAUD\_RATE);

  SerialPort.begin(BAUD\_RATE, SERIAL\_8N1, UART\_RX\_PIN, UART\_TX\_PIN);

  WiFi.mode(WIFI\_STA);

  WiFi.begin(ssid, password);

  if(WiFi.status() != WL\_CONNECTED){

      Serial.print("Attempting to connect Wi-Fi");

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

        WiFi.begin(ssid,password);

        delay(1000);

      }

      Serial.println("\nWiFi Connected.");

  }

  Serial.print("RSSI: ");

  Serial.println(WiFi.RSSI());

  pinMode(led,OUTPUT);

  ThingSpeak.begin(client);  // Initialize ThingSpeak

  delay(1000);

}

void loop() {

// Variable definition for uploading data to ThingSpeak

    float Lux\_Y;

    float Tem\_Y;

    float Lux\_F;

    float Tem\_F;

    float WATER;

    float moisture;

    float Humidity;

    float Temperature;

// Serial Port Scanning

  if (SerialPort.available()) {

    // String message = SerialPort.readString();

    // Serial.println("Received message: " + message);

    size\_t byteSize = sizeof(UART\_message);

    byte\* byteArray = new byte[byteSize];

    SerialPort.read(byteArray, byteSize);

    // Deserialize the byte array back into the structure

    UART\_message UARTReceivedData;

    memcpy(&UARTReceivedData, byteArray, byteSize);

/\*--------Process different types of data----------\*/

    switch(UARTReceivedData.id){

       case 1:

              // Print the received structure data and check the data valid or not

              if (UARTReceivedData.b > 0 && (UARTReceivedData.c > -50 && UARTReceivedData.c < 50)){

                  Serial.print("Received from Board ID: ");

                  Serial.println(UARTReceivedData.id);

                  Serial.print("   Lux\_Y: ");

                  Serial.print(UARTReceivedData.b);   // b is linked to humidity

                  Serial.println(" LUX ");

                  Serial.print("    Tem\_Y: ");    // c to temperature

                  Serial.print(UARTReceivedData.c);

                  Serial.println(" C ");

                  Serial.print("    Lux\_F: ");    // c to temperature

                  Serial.print(UARTReceivedData.d);

                  Serial.println(" LUX ");

                  Serial.print("    Tem\_F: ");    // c to temperature

                  Serial.print(UARTReceivedData.e);

                  Serial.println(" C ");

                  Serial.print("    WATER: ");    // c to temperature

                  Serial.print(UARTReceivedData.f);

                  //Serial.println(" C ");

                  Serial.print("    CRC = ");  // CRC=1 means data transmission right

                  Serial.println(UARTReceivedData.CRC\_checksum);

                  Lux\_Y=UARTReceivedData.b;

                  Tem\_Y=UARTReceivedData.c;

                  Lux\_F=UARTReceivedData.d;

                  Tem\_F=UARTReceivedData.e;

                  WATER=UARTReceivedData.f;

                  ThingSpeak.setField(1, Lux\_Y);

                  ThingSpeak.setField(2, Tem\_Y);

                  ThingSpeak.setField(3, Lux\_F);

                  ThingSpeak.setField(4, Tem\_F);

                  ThingSpeak.setField(5, WATER);

              }

              break;

       case 2:

               //if (UARTReceivedData.b > 0 && UARTReceivedData.c > 0 )

               {

                  Serial.print("Received from Board ID: ");

                  Serial.println(UARTReceivedData.id);

                  Serial.print("   moisture: ");

                  Serial.print(UARTReceivedData.g);   // b is linked to moisture

                  Serial.print("   Humidity: ");    // c to temperature

                  Serial.print(UARTReceivedData.h);

                  Serial.println(" % ");

                   Serial.print("  Temperature: ");    // c to temperature

                  Serial.print(UARTReceivedData.i);

                  Serial.println(" C ");

                  Serial.print("    CRC = ");  // CRC=1 means data transmission right

                  Serial.println(UARTReceivedData.CRC\_checksum);

                  moisture=UARTReceivedData.g;

                  Humidity=UARTReceivedData.h;

                  Temperature=UARTReceivedData.i;

                  ThingSpeak.setField(6, moisture);

                  ThingSpeak.setField(7, Humidity);

                  ThingSpeak.setField(8, Temperature);

              }

              break;

      /\*default:

               if (UARTReceivedData.b > 0 ){

                  Serial.print("Received from Board ID: ");

                  Serial.println(UARTReceivedData.id);

                  Serial.print("    TSL2591 Light Density: ");

                  Serial.print(UARTReceivedData.b);   // b is linked to light desnity

                  Serial.println(" lux ");

                  Serial.print("    CRC = ");  // CRC=1 means data transmission right

                  Serial.println(UARTReceivedData.CRC\_checksum);

                  LightDensity=UARTReceivedData.b;

                  ThingSpeak.setField(5, LightDensity);

              }

              break;\*/

    }

    digitalWrite(led, HIGH);

    delay(100);

  //ThingSpeak Data UIploading

  if ((millis() - lastTime) > timerDelay) {

    // Connect or reconnect to WiFi

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

      Serial.print("Attempting to connect");

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

        WiFi.begin(ssid,password);

        delay(1000);

      }

      Serial.println("\nConnected.");

    }

    // Get a new temperature reading

      // ThingSpeak.setField(1, humidity);

      // ThingSpeak.setField(2, temperature);

      // ThingSpeak.setField(3, TempVoltage);

      // ThingSpeak.setField(4, LightVoltage);

      // ThingSpeak.setField(5, LightDensity);

    // Write to ThingSpeak. There are up to 8 fields in a channel, allowing you to store up to 8 different

    // pieces of information in a channel.  Here, we write to field 1.

    int x = ThingSpeak.writeFields(myChannelNumber, myWriteAPIKey);

    if(x == 200){

      Serial.println("Channel update successful.");

    }

    else{

      Serial.println("Problem updating channel. HTTP error code " + String(x));

    }

    lastTime = millis();

  }

  }

  digitalWrite(led, LOW);//digitalWrite(LED, HIGH);

}