|  |
| --- |
| Sender  #include<Wire.h>  #include <SPI.h>  #include <LoRa.h>  #include <Adafruit\_Sensor.h>  #include <Adafruit\_BMP085.h> // Change to Adafruit\_BMP180 library  #define BAND 433E6  #define rain\_sensor A0  #define SEALEVELPRESSURE\_HPA (1013.25)  Adafruit\_BMP085 bmp; // Change to Adafruit\_BMP180  String LoRaMessage = "";  char device\_id[12] = "MyDevice123";  void setup() {  Serial.begin(115200);  Wire.begin();      pinMode (rain\_sensor, INPUT);  while (!Serial);  Serial.println(F("LoRa Sender"));    if (!LoRa.begin(BAND)) {  Serial.println(F("Starting LoRa failed!"));  while (1);  }    if (!bmp.begin()) { // Change to bmp.begin() for BMP180  Serial.println("Could not find a valid BMP180 sensor, check wiring!");  while (1);  }  }  void loop() {  float temperature = bmp.readTemperature(); // Change to bmp.readTemperature() for BMP180  float pressure = bmp.readPressure() / 100.0F; // Change to bmp.readPressure() for BMP180  float altitude = bmp.readAltitude();    int rainfall = map(analogRead(rain\_sensor), 780, 0, 0, 100);  if (rainfall >= 100) {  rainfall = 100;  }  if (rainfall <= 0) {  rainfall = 0;  }      Serial.print(F("Device ID: "));  Serial.println(device\_id);    Serial.print(F("Temperature = "));  Serial.print(temperature);  Serial.println(F("\*C"));    Serial.print(F("Pressure = "));  Serial.print(pressure);  Serial.println(F("hPa"));    Serial.print(F("Approx. Altitude = "));  Serial.print(altitude);  Serial.println(F("m"));        Serial.print(F("Rainfall = "));  Serial.print(rainfall);  Serial.println(F("%"));        Serial.println();    LoRaMessage = String(device\_id) + "/" + String(temperature) + "&" + String(pressure)  + "#" + String(altitude) + "@"  + "^" + String(rainfall) + "!" ;    // send packet  LoRa.beginPacket();  LoRa.print(LoRaMessage);  LoRa.endPacket();  delay(10000);  } |
| Receiver  #include <SPI.h>  #include <LoRa.h>  #include <WiFi.h>    String apiKey = "JL0D44OC494NWMYT"; // Enter your Write API key from ThingSpeak  const char\* ssid = "HUAWEI nova 3e";  const char\* password = "123456789";  const char\* server = "api.thingspeak.com";    #define SS 5  #define RST 14  #define DI0 2    //#define TX\_P 17  #define BAND 433E6  //#define ENCRYPT 0x23    String device\_id;  String temperature;  String pressure;  String altitude;  String rainfall;    WiFiClient client;      void setup()  {  Serial.begin(115200);  Serial.println("LoRa Receiver");  //LoRa.setTxPower(TX\_P);  //LoRa.setSyncWord(ENCRYPT);    LoRa.setPins(SS, RST, DI0);  if (!LoRa.begin(BAND))  {  Serial.println("Starting LoRa failed!");  while (1);  }    Serial.println("Connecting to ");  Serial.println(ssid);    //connect to your local wi-fi network  WiFi.begin(ssid, password);    //check wi-fi is connected to wi-fi network  while (WiFi.status() != WL\_CONNECTED) {  delay(1000);  Serial.print(".");  }  Serial.println("");  Serial.println("WiFi connected..!");  Serial.print("Got IP: ");  Serial.println(WiFi.localIP());    }    void loop()  {  // try to parse packet  int pos1, pos2, pos3, pos4, pos5, pos6, pos7;    int packetSize = LoRa.parsePacket();  if (packetSize)  {  // received a packet  Serial.print("Received packet: ");  String LoRaData = LoRa.readString();  Serial.print(LoRaData);  // read packet  while (LoRa.available())  {  Serial.print((char)LoRa.read());  }  // print RSSI of packet  Serial.print("' with RSSI ");  Serial.println(LoRa.packetRssi());    pos1 = LoRaData.indexOf('/');  pos2 = LoRaData.indexOf('&');  pos3 = LoRaData.indexOf('#');  pos4 = LoRaData.indexOf('@');  pos5 = LoRaData.indexOf('$');  pos6 = LoRaData.indexOf('^');  pos7 = LoRaData.indexOf('!');    device\_id = LoRaData.substring(0, pos1);  temperature = LoRaData.substring(pos1 + 1, pos2);  pressure = LoRaData.substring(pos2 + 1, pos3);  altitude = LoRaData.substring(pos3 + 1, pos4);  humidity = LoRaData.substring(pos4 + 1, pos5);  dewPoint = LoRaData.substring(pos5 + 1, pos6);  rainfall = LoRaData.substring(pos6 + 1, pos7);  lux = LoRaData.substring(pos7 + 1, LoRaData.length());    Serial.print(F("Device ID = "));  Serial.println(device\_id);    Serial.print(F("Temperature = "));  Serial.print(temperature);  Serial.println(F("\*C"));    Serial.print(F("Pressure = "));  Serial.print(pressure);  Serial.println(F("hPa"));    Serial.print(F("Approx. Altitude = "));  Serial.print(altitude);  Serial.println(F("m"));        Serial.print(F("Rainfall = "));  Serial.print(rainfall);  Serial.println(F("%"));          Serial.println();    if (client.connect(server, 80)) // "184.106.153.149" or api.thingspeak.com  {  String postStr = apiKey;  postStr += "&field1=";  postStr += String(temperature);  postStr += "&field2=";  postStr += String(pressure);  postStr += "&field3=";  postStr += String(altitude);  postStr += "&field4=";      postStr += String(rainfall);  postStr += "&field7=";    postStr += "r\n";    client.print("POST /update HTTP/1.1\n");  client.print("Host: api.thingspeak.com\n");  client.print("Connection: close\n");  client.print("X-THINGSPEAKAPIKEY: " + apiKey + "\n");  client.print("Content-Type: application/x-www-form-urlencoded\n");  client.print("Content-Length: ");  client.print(postStr.length());  client.print("\n\n");  client.print(postStr);    Serial.println("Data Send to Thingspeak");  delay(500);  }  client.stop();  Serial.println("Waiting...");    }  } |