|  |
| --- |
| #include <SPI.h>  #include <LoRa.h>  #include <Adafruit\_Sensor.h>  #include <Adafruit\_BME280.h>  #include <BH1750.h> // adds BH1750 library file    //#define SS 10  //#define RST 9  //#define DI0 2    //#define TX\_P 17  //#define ENCRYPT 0x78    #define BAND 433E6  #define rain\_sensor A0    #define SEALEVELPRESSURE\_HPA (1013.25)  Adafruit\_BME280 bme;    BH1750 lightMeter;    String LoRaMessage = "";  char device\_id[12] = "MyDevice123";      void setup()  {  Serial.begin(115200);  Wire.begin();  lightMeter.begin();    pinMode (rain\_sensor, INPUT);  while (!Serial);    Serial.println(F("LoRa Sender"));    //LoRa.setPins(SS, RST, DI0);  //LoRa.setTxPower(TX\_P);  //LoRa.setSyncWord(ENCRYPT);    if (!LoRa.begin(BAND))  {  Serial.println(F("Starting LoRa failed!"));  while (1);  }  if (!bme.begin(0x76))  {  Serial.println("Could not find a valid BME280 sensor, check wiring!");  while (1);  }  }    void loop()  {  float temperature = bme.readTemperature();  float pressure = bme.readPressure() / 100.0F;  float altitude = bme.readAltitude(SEALEVELPRESSURE\_HPA);  float humidity = bme.readHumidity();    double dewPoint = dewPointFast(temperature, humidity);    int rainfall = map(analogRead(rain\_sensor), 780, 0, 0, 100);  if (rainfall >= 100)  {  rainfall = 100;  }  if (rainfall <= 0)  {  rainfall = 0;  }    float lux = lightMeter.readLightLevel();    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("Humidity = "));  Serial.print(humidity);  Serial.println(F("%"));    Serial.print(F("Dew point = "));  Serial.print(dewPoint);  Serial.println(F(" \*C"));    Serial.print(F("Rainfall = "));  Serial.print(rainfall);  Serial.println(F("%"));    Serial.print(F("Light = "));  Serial.print(lux);  Serial.println(F(" lx"));      Serial.println();    LoRaMessage = String(device\_id) + "/" + String(temperature) + "&" + String(pressure)  + "#" + String(altitude) + "@" + String(humidity) + "$" + String(dewPoint)  + "^" + String(rainfall) + "!" + String(lux);    // send packet  LoRa.beginPacket();  LoRa.print(LoRaMessage);  LoRa.endPacket();  delay(10000);  }      double dewPointFast(double celsius, double humidity)  {  double a = 17.271;  double b = 237.7;  double temp = (a \* celsius) / (b + celsius) + log(humidity \* 0.01);  double Td = (b \* temp) / (a - temp);  return Td;  }  **Source Code/Program for Gateway: WebServer**  Here is the **gateway**code for the ESP32 board. Using this code you can view weather station data in a web browser. The ESP32 web server creates a webpage where all weather station data is logged.  #include <LoRa.h>  #include <WiFi.h>  #include <WebServer.h>      #define SS 5  #define RST 14  #define DI0 2    //#define TX\_P 17  #define BAND 433E6  //#define ENCRYPT 0x78    String device\_id;  String temperature;  String pressure;  String altitude;  String humidity;  String dewPoint;  String rainfall;  String lux;    const char\* ssid = "Bynark\_Airtel";  const char\* password = "bynark@123";    WebServer server(80);    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());    server.on("/", handle\_OnConnect);  server.onNotFound(handle\_NotFound);    server.begin();  Serial.println("HTTP server started");  }    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("Humidity = "));  Serial.print(humidity);  Serial.println(F("%"));    Serial.print("Dew point = ");  Serial.print(dewPoint);  Serial.println(" \*C");    Serial.print(F("Rainfall = "));  Serial.print(rainfall);  Serial.println(F("%"));    Serial.print(F("Light = "));  Serial.print(lux);  Serial.println(F(" lx"));    Serial.println();    server.handleClient();  }  }    void handle\_OnConnect()  {  server.send(200, "text/html", SendHTML(temperature.toFloat(), humidity.toFloat(), pressure.toFloat(), altitude.toFloat(), dewPoint.toFloat(), rainfall.toFloat(), lux.toFloat()));  }      void handle\_NotFound()  {  server.send(404, "text/plain", "Not found");  }    String SendHTML(float temperature, float humidity, float pressure, float altitude, float dewPoint, float rainfall, float lux)  {  String ptr = "<!DOCTYPE html> <html>\n";  ptr += "<head><meta name=\"viewport\" content=\"width=device-width, initial-scale=1.0, user-scalable=no\">\n";  ptr += "<title>Wireless Weather Station</title>\n";  ptr += "<style>html { font-family: Helvetica; display: inline-block; margin: 0px auto; text-align: center;}\n";  ptr += "body{margin-top: 50px;} h1 {color: #444444;margin: 50px auto 30px;}\n";  ptr += "p {font-size: 24px;color: #444444;margin-bottom: 10px;}\n";  ptr += "</style>\n";  ptr += "</head>\n";  ptr += "<body>\n";  ptr += "<div id=\"webpage\">\n";  ptr += "<h1>Wireless Weather Station</h1>\n";    ptr += "<p>Temperature: ";  ptr += temperature;  ptr += "°C</p>";    ptr += "<p>Humidity: ";  ptr += humidity;  ptr += "%</p>";    ptr += "<p>Pressure: ";  ptr += pressure;  ptr += "hPa</p>";    ptr += "<p>Altitude: ";  ptr += altitude;  ptr += "m</p>";    ptr += "<p>Dew Point: ";  ptr += dewPoint;  ptr += "°C</p>";    ptr += "<p>Rainfall: ";  ptr += rainfall;  ptr += "%</p>";    ptr += "<p>Light: ";  ptr += lux;  ptr += "lx</p>";    ptr += "</div>\n";  ptr += "</body>\n";  ptr += "</html>\n";  return ptr;  } |