ESP32 GATWAY LoRa receiver with thingspeak code

```
#include <LoRa.h>
#include <SPI.h>
#include <Wire.h>
#include <Adafruit_GFX.h>
#include <Adafruit_SSD1306.h>
#include <WiFi.h> // Include the WiFi library
#include <ThingSpeak.h> // Include the ThingSpeak library

#define SCREEN_WIDTH 128    // OLED display width, in pixels
#define SCREEN_HEIGHT 64    // OLED display height, in pixels
#define OLED_RESET -1       // Reset pin # (or -1 if sharing Arduino reset
pin)

Adafruit_SSD1306 display(SCREEN_WIDTH, SCREEN_HEIGHT, &Wire, OLED_RESET);

#define ss 5
#define rst 14
#define dio0 2
String LoRaData;
// WiFi and ThingSpeak settings
const char *ssid = "Redmi 12 5G";
const char *password = "8207747517";
const unsigned long channelID = 2506303;
const char *apiKey = "GDDNTHM4VPHSIZTM";

WiFiClient client; // Declare the WiFiClient object

void setup() {
  Serial.begin(115200);

  if (!display.begin(SSD1306_SWITCHCAPVCC, 0x3C)) {
    Serial.println(F("SSD1306 allocation failed"));
    while (true);
  }
  delay(2000);
  display.clearDisplay();

  display.setTextSize(1);
  display.setTextColor(WHITE);
  display.setCursor(0, 10);
  display.println("LoRa Receiver");
  display.display();

  while (!Serial);
```

```
  Serial.println("LoRa Receiver");

  LoRa.setPins(ss, rst, dio0);    // Setup LoRa transceiver module

  while (!LoRa.begin(433E6)) {    // 433E6 - Asia, 866E6 - Europe, 915E6 -
North America
    Serial.println(".");
    delay(500);
  }
  LoRa.setSyncWord(0xA5);
  Serial.println("LoRa Initializing OK!");

  // Connect to WiFi
  WiFi.begin(ssid, password);
  while (WiFi.status() != WL_CONNECTED) {
    delay(1000);
    Serial.println("Connecting to WiFi...");
  }
  Serial.println("Connected to WiFi.");

  // Initialize ThingSpeak
  ThingSpeak.begin(client); // client is an instance of WiFiClient
}

void loop() {
  int packetSize = LoRa.parsePacket();    // Try to parse packet
  if (packetSize) {
    Serial.println("Received packet");

    while (LoRa.available()) {            // Read packet
      LoRaData = LoRa.readString();
      Serial.print(LoRaData);
    }
    Serial.print("RSSI: ");               // Print RSSI of packet
    Serial.println(LoRa.packetRssi());
    Serial.println("");

    // Display on OLED
    display.clearDisplay();
    display.setTextSize(1);
    display.setTextColor(WHITE);
    display.setCursor(20, 0);
    display.println("LoRa Receiver");

    display.setTextSize(1);
    display.setTextColor(WHITE);
    display.setCursor(0, 20);
    display.println(LoRaData);
```

```
    display.print("RSSI: ");
    display.println(LoRa.packetRssi());
    display.display();

    // Parse LoRaData to extract sensor values
    float flowRate, temperature, tdsValue, voltage, pressure;
    if (sscanf(LoRaData.c_str(), "Flow rate: %f L/min Temperature: %f C TDS
Value: %f ppm Voltage: %f V Pressure: %f kPa",
               &flowRate, &temperature, &tdsValue, &voltage, &pressure) == 5)
{
      // Send data to ThingSpeak
      ThingSpeak.setField(1, flowRate);    // Field 1: Flow rate
      ThingSpeak.setField(2, temperature); // Field 2: Temperature
      ThingSpeak.setField(3, tdsValue);    // Field 3: TDS Value
      ThingSpeak.setField(4, pressure);    // Field 4: Pressure

      int status = ThingSpeak.writeFields(channelID, apiKey);
      if (status == 200) {
        Serial.println("Data sent to ThingSpeak successfully.");
      } else {
        Serial.println("Error sending data to ThingSpeak. HTTP error code: " +
String(status));
      }
    } else {
      Serial.println("Error parsing LoRaData.");
    }
  }
}
```