

```

#include <SPI.h>
#include <LoRa.h>
#include <Wire.h>
#include <OneWire.h>
#include <DallasTemperature.h>
#include <GravityTDS.h>

//LoRa Pin
String outgoing;

#define ss 10
#define rst 9
#define dio0 2

byte msgCount = 0;
byte MasterNode = 0xFF;
byte Node2 = 0xCC;

//Sensor Pin
const int turbidityPin = A0;    // Analog pin for turbidity sensor
const int temperaturePin = A1;  // Analog pin for temperature sensor
const int tdsPin = A2;         // Analog pin for TDS sensor
const int oneWireBus = 2;      // Digital pin for OneWire bus
const int pHpin = A3;
float temperatureValue = 0;
float turbidityValue = 0;
float tdsValue = 0;
float pHvalue = 0;
unsigned long int avgValue;
float b;
int buf[10];
OneWire oneWire(oneWireBus);
DallasTemperature sensors(&oneWire);
GravityTDS gravityTds;

String Mymessage = "";
String incoming = "";

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

  while (!Serial);

  Serial.println("LoRa Node2");

  LoRa.setPins(ss, rst, dio0);

  if (!LoRa.begin(433E6)) {
    Serial.println("LoRa Fail!!!!");
    while (1);
  }

  gravityTds.begin();
  sensors.begin();
  pinMode(2, OUTPUT);
}

void loop() {
  for (int i = 0; i < 10; i++) {
    buf[i] = analogRead(pHpin);
    delay(10);
  }

  for (int i = 0; i < 9; i++) {

    for (int j = i + 1; j < 10; j++) {

      if (buf[i] > buf[j]) {
        temperatureValue = buf[i];
        buf[i] = buf[j];
        buf[j] = temperatureValue;
      }
    }
  }

  avgValue = 0;

```

```

    for (int i = 2; i < 8; i++)
        avgValue += buf[i];
    float pHvalue = (float)avgValue * 5.0 / 1024 / 6;
    pHvalue = 3.5 * pHvalue;

    // Read sensor values
    float turbidityValue = analogRead(turbidityPin);
    float temperatureValue = getTemperature();
    float tdsValue = gravityTds.getTdsValue();

    // Print values to Serial Monitor
    Serial.print("Turbidity: ");
    Serial.println(turbidityValue);

    Serial.print("Temperature: ");
    Serial.print(temperatureValue);
    Serial.println(" °C");

    Serial.print("TDS: ");
    Serial.print(tdsValue);
    Serial.println(" ppm");

    Serial.print("pH: ");
    Serial.print(pHvalue);

    onReceive(LoRa.parsePacket());
}

float getTemperature(){
    sensors.requestTemperatures();
    return sensors.getTempCByIndex(0);
}

void onReceive(int packetSize){
    if (packetSize == 0) {
        int recipient = LoRa.read();
        byte sender = LoRa.read();
        byte incomingMsgId = LoRa.read();
        byte incomingLength = LoRa.read();

        String incoming = "";

        while (LoRa.available()) {
            incoming += (char)LoRa.read();
        }

        if (incomingLength != incoming.length()){
            ;
            return;
        }

        if (recipient != Node2 && recipient != MasterNode) {
            ;
            return;
        }

        Serial.println(incoming);
        int Val = incoming.toInt();
        if (Val == 20) {
            Mymessage = Mymessage + turbidityValue + ',' + temperatureValue + ',' + tdsValue + ',' + pHvalue;
            sendMessage(Mymessage, MasterNode, Node2);
            delay(100);
            Mymessage = "";
        }
    }
}

void sendMessage(String outgoing, byte MasterNode, byte Node2){
    LoRa.beginPacket();
    LoRa.write(MasterNode);
    LoRa.write(Node2);
    LoRa.write(msgCount);
    LoRa.write(outgoing.length());
}

```

```
LoRa.print(outgoing);  
LoRa.endPacket();  
msgCount++;  
}
```