

```

#include "WiFi.h"
#include "ThingSpeak.h"
#define RE 4
#define RXD2 17
#define TXD2 16
WiFiClient client;
unsigned long myChannelNumber = 1;
const char * myWriteAPIKey = "Put in your API key here";

const byte humi[] = {0x01, 0x03, 0x00, 0x00, 0x00, 0x01, 0xe84, 0x0a};
const byte temp[] = {0x01, 0x03, 0x00, 0x01, 0x00, 0x01, 0xd5, 0xca};
const byte cond[] = {0x01, 0x03, 0x00, 0x02, 0x00, 0x01, 0x25, 0xca};
const byte phph[] = {0x01, 0x03, 0x00, 0x03, 0x00, 0x01, 0x74, 0x0a};
const byte nitro[] = {0x01, 0x03, 0x00, 0x04, 0x00, 0x01, 0xec5, 0xcb};
const byte phos[] = {0x01, 0x03, 0x00, 0x05, 0x00, 0x01, 0xe94, 0x0b};
const byte pota[] = {0x01, 0x03, 0x00, 0x06, 0x00, 0x01, 0xe64, 0x0b};
const byte sali[] = {0x01, 0x03, 0x00, 0x07, 0x00, 0x01, 0xe35, 0xcb};
const byte tds[] = {0x01, 0x03, 0x00, 0x08, 0x00, 0x01, 0xe05, 0xc8};
byte values[11];
// ThingSpeak information.
#define NUM_FIELDS 8
#define HUMIDITY_FIELD 1
#define TEMPERATURE_FIELD 2
#define CONDUCTIVITY_FIELD 3
#define pH_FIELD 4
#define NITROGEN_FIELD 5
#define PHOSPHORUS_FIELD 6
#define POTASSIUM_FIELD 7
#define RSSI_FIELD 8

const char* ssid = "????????"; // your network SSID (name)
const char* password = "????????"; // your network password

void setup() {
  Serial.begin(4800);
  Serial1.begin(4800, SERIAL_8N1, RXD2, TXD2);

  WiFi.mode(WIFI_STA);
  ThingSpeak.begin(client); // Initialize ThingSpeak
  pinMode(RE, OUTPUT);
  // digitalWrite(RE, LOW);
  //delay(3000);
  digitalWrite(RE, LOW);
  delay(1000);

  WiFi.mode(WIFI_STA);

  delay(3000);
}

void loop() {
  // 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(5000);
    }
  }
}

```

```

Serial.println("\nConnected.");
}

float val1, val2, val3, val4, val5, val6, val7, val8, val9;

//***** HUMIDITY *****
Serial.print("Humidity: ");
digitalWrite(RE, HIGH);
delay(10);

for (uint8_t i = 0; i < sizeof(humi); i++ ) Serial1.write( humi[i] );
Serial1.flush();
digitalWrite(RE, LOW);
delay(100);
for (byte i = 0; i < 7; i++) {
    values[i] = Serial1.read();
}
val1 = int(values[3]<<8|values[4]);
val1 = val1/10;
Serial.print(" = ");
Serial.print(val1,1);
Serial.println(" %");
delay(200);

//***** TEMPERATURE *****

Serial.print("Temperature: ");
digitalWrite(RE, HIGH);
delay(10);
for (uint8_t i = 0; i < sizeof(temp); i++ ) Serial1.write( temp[i] );
Serial1.flush();
digitalWrite(RE, LOW);
delay(100);
for (byte i = 0; i < 7; i++) {
    values[i] = Serial1.read();
}
val2 = int(values[3]<<8|values[4]);
val2=val2/10;
Serial.print(" = ");
Serial.print(val2,1);
Serial.println(" deg.C");
delay(200);

//***** CONDUCTIVITY *****
Serial.print("Conductivity: ");
digitalWrite(RE, HIGH);
delay(10);
for (uint8_t i = 0; i < sizeof(cond); i++ ) Serial1.write( cond[i] );
Serial1.flush();
digitalWrite(RE, LOW);
delay(100);
for (byte i = 0; i < 7; i++) {
    values[i] = Serial1.read();
}
val3 = int(values[3]<<8|values[4]);
Serial.print(" = ");
Serial.print(val3);
Serial.println(" uS/cm");

```

```

delay(200);

//***** Ph *****
Serial.print("pH: ");
digitalWrite(RE, HIGH);
delay(10);

for (uint8_t i = 0; i < sizeof(phph); i++ ) Serial1.write( phph[i] );
Serial1.flush();
digitalWrite(RE, LOW);
delay(100);
for (byte i = 0; i < 7; i++) {
    values[i] = Serial1.read();
    // Serial.print(values[i], HEX);
//    Serial.print(' ');
}
val4 = int(values[3]<<8|values[4]);
val4 = val4/10;
Serial.print(" = ");
Serial.println(val4,1);
delay(200);
//***** NITROGEN *****
Serial.print("Nitrogen: ");
digitalWrite(RE, HIGH);
delay(10);
for (uint8_t i = 0; i < sizeof(nitro); i++ ) Serial1.write( nitro[i] );
Serial1.flush();
digitalWrite(RE, LOW);
delay(100);
for (byte i = 0; i < 7; i++) {
    values[i] = Serial1.read();
    // Serial.print(values[i], HEX);
//    Serial.print(' ');
}
val5 = int(values[3]<<8|values[4]);
Serial.print(" = ");
Serial.print(val5);
Serial.println(" mg/L");
delay(200);

//***** PHOSPHORUS *****
Serial.print("Phosphorus: ");
digitalWrite(RE, HIGH);
delay(10);

for (uint8_t i = 0; i < sizeof(phos); i++ ) Serial1.write( phos[i] );
Serial1.flush();
digitalWrite(RE, LOW);
delay(100);
for (byte i = 0; i < 7; i++) {
    values[i] = Serial1.read();
    // Serial.print(values[i], HEX);
//    Serial.print(' ');
}
val6 = int(values[3]<<8|values[4]);
Serial.print(" = ");
Serial.print(val6);
Serial.println(" mg/L");

```

```

delay(200);

//***** POTASSIUM *****

Serial.print("Potassium: ");
digitalWrite(RE, HIGH);
delay(10);

for (uint8_t i = 0; i < sizeof(pota); i++ ) Serial1.write( pota[i] );
Serial1.flush();
digitalWrite(RE, LOW);
delay(100);
for (byte i = 0; i < 7; i++) {
    values[i] = Serial1.read();
}
val7 = int(values[3]<<8|values[4]);
Serial.print(" = ");
Serial.print(val7);
Serial.println(" mg/L");
delay(200);

//***** SALINITY *****
Serial.print("Salinity: ");

digitalWrite(RE, HIGH);
delay(10);

for (uint8_t i = 0; i < sizeof(sali); i++ ) Serial1.write( sali[i] );
Serial1.flush();
digitalWrite(RE, LOW);
delay(100);
for (byte i = 0; i < 7; i++) {
    values[i] = Serial1.read();
    // Serial.print(values[i], HEX);
    // Serial.print(' ');
}
val8 = int(values[3]<<8|values[4]);
Serial.print(" = ");
Serial.print(val8);
Serial.println(" g/L");
delay(200);
//***** TDS *****
Serial.print("TDS: ");
digitalWrite(RE, HIGH);
delay(10);
for (uint8_t i = 0; i < sizeof(tds); i++ ) Serial1.write( tds[i] );
Serial1.flush();
digitalWrite(RE, LOW);
delay(100);
for (byte i = 0; i < 7; i++) {
    values[i] = Serial1.read();
    // Serial.print(values[i], HEX);
    // Serial.print(' ');
}
val9 = int(values[3]<<8|values[4]);
Serial.print(" = ");
Serial.print(val9);
Serial.println(" mg/L");

```

```
    delay(200);

Serial.println(WiFi.RSSI());

    Serial.println("");
    Serial.println("");
    Serial.println("");

    delay(1000);

// set the fields with the values
ThingSpeak.setField(1, val1);
ThingSpeak.setField(2, val2);
ThingSpeak.setField(3, val3);
ThingSpeak.setField(4, val4);
ThingSpeak.setField(5, val5);
ThingSpeak.setField(6, val6);
ThingSpeak.setField(7, val7);
ThingSpeak.setField(8, (WiFi.RSSI()));

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));
    }
    delay(600000); // send data every 10 minutes
}
```