LoraWan UNO

mit MQ2 Sensor

fpf2000/22

Grundlagen

https://github.com/TheThingsNetwork/workshops

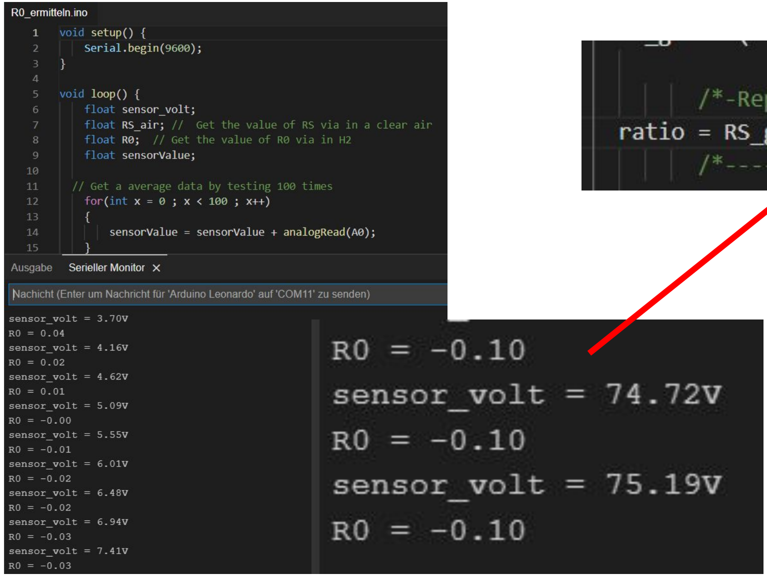
https://www.thethingsindustries.com/docs/devices/the-things-uno/

https://docs.arduino.cc/learn/communication/lorawan-101

https://www.pololu.com/file/0J309/MQ2.pdf

Sensorwerte Kalibrieren

Mit dem Programm R0ermittel.ino den Wert ermitteln und aufschreiben



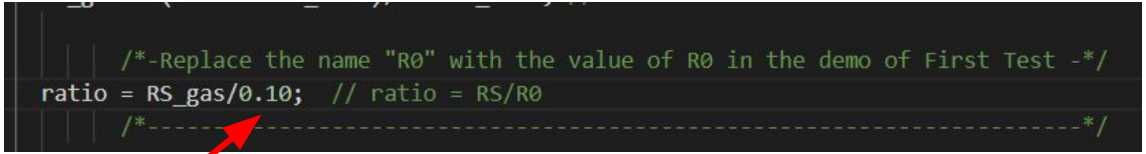
R0_ermitteln.ino

```
void setup() {
    Serial.begin(9600);
}

void loop() {
    float sensor_volt;
    float RS_air; //  Get the value of RS via in a clear air
    float R0;  // Get the value of R0 via in H2
    float sensorValue;

  // Get a average data by testing 100 times
    for(int x = 0 ; x < 100 ; x++)
    {
        sensorValue = sensorValue + analogRead(A0);
    }
```

Ausgabe
Serieller Monitor ×
Nachricht (Enter um Nachricht für 'Arduino Leonardo' auf 'COM11' zu senden)

```
sensor_volt = 3.70V
R0 = 0.04
sensor_volt = 4.16V
R0 = 0.02
sensor_volt = 4.62V
R0 = 0.01
sensor_volt = 5.09V
R0 = -0.00
sensor_volt = 5.55V
R0 = -0.01
sensor_volt = 6.01V
R0 = -0.02
sensor_volt = 6.48V
R0 = -0.02
sensor_volt = 6.94V
R0 = -0.03
sensor_volt = 7.41V
R0 = -0.03
```

```
R0 = -0.10
sensor_volt = 74.72V
R0 = -0.10
sensor_volt = 75.19V
R0 = -0.10
```

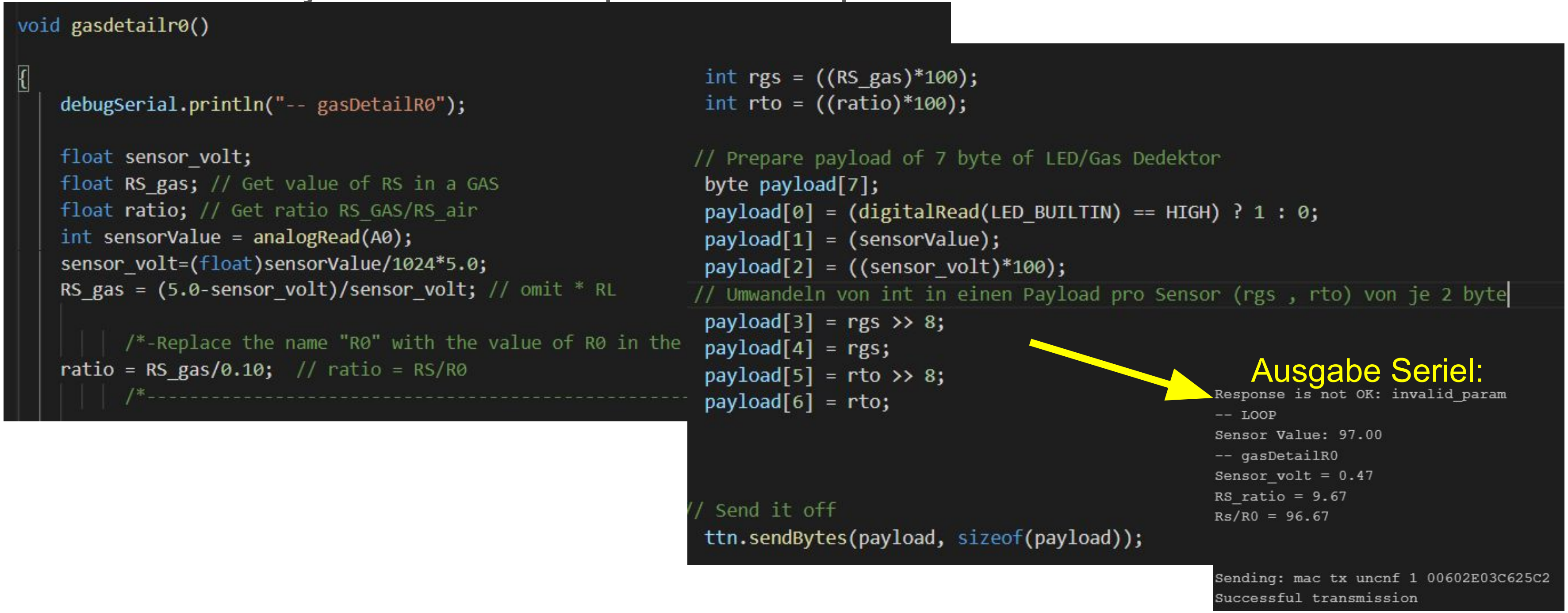


```
      /*-Replace the name "R0" with the value of R0 in the demo of First Test -*/
ratio = RS_gas/0.10;  // ratio = RS/R0
      /*-----------------------------------------------------------------------*/
```

Anschließend in das programm einfügen

Payload im Programm einrichten

Im Code das Paylod format entsprechend anpassen



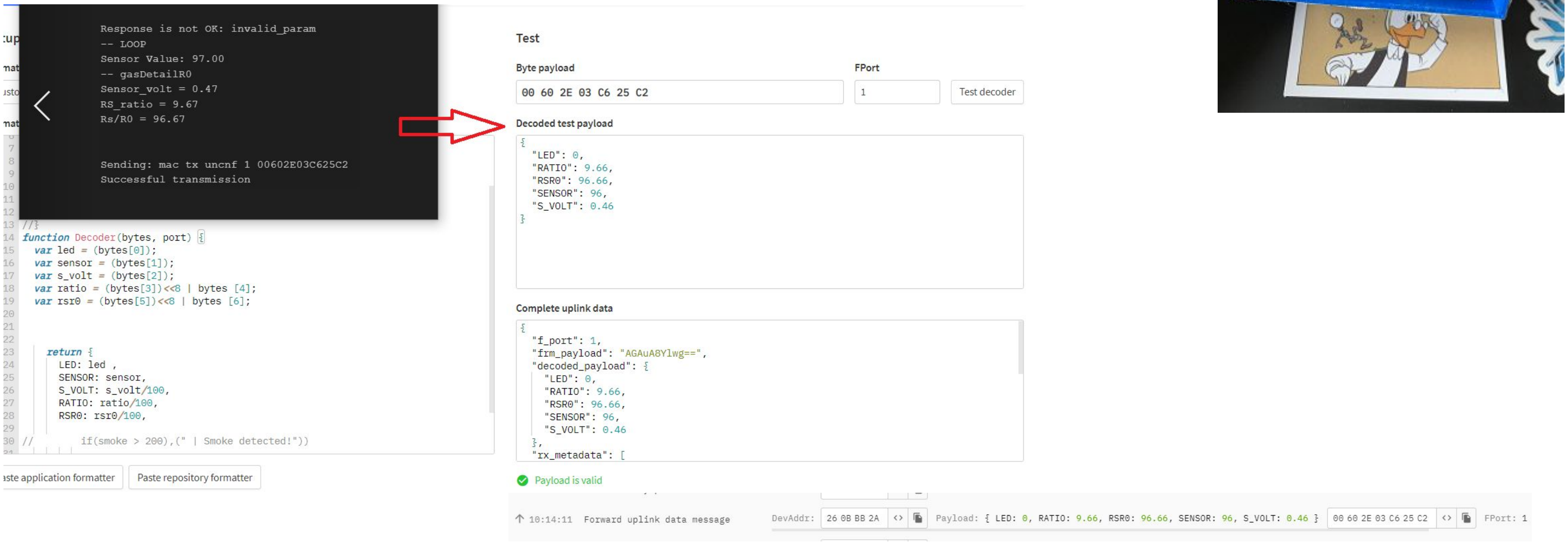
```
void gasdetailr0()

{
    debugSerial.println("-- gasDetailR0");

    float sensor_volt;
    float RS_gas; // Get value of RS in a GAS
    float ratio; // Get ratio RS_GAS/RS_air
    int sensorValue = analogRead(A0);
    sensor_volt=(float)sensorValue/1024*5.0;
    RS_gas = (5.0-sensor_volt)/sensor_volt; // omit * RL

          /*-Replace the name "R0" with the value of R0 in the
    ratio = RS_gas/0.10;  // ratio = RS/R0
          /*-----------------------------------------------
```

```
 int rgs = ((RS_gas)*100);
 int rto = ((ratio)*100);

// Prepare payload of 7 byte of LED/Gas Dedektor
 byte payload[7];
 payload[0] = (digitalRead(LED_BUILTIN) == HIGH) ? 1 : 0;
 payload[1] = (sensorValue);
 payload[2] = ((sensor_volt)*100);
// Umwandeln von int in einen Payload pro Sensor (rgs , rto) von je 2 byte
 payload[3] = rgs >> 8;
 payload[4] = rgs;
 payload[5] = rto >> 8;
 payload[6] = rto;



// Send it off
 ttn.sendBytes(payload, sizeof(payload));
```

Ausgabe Seriel:

```
Response is not OK: invalid_param
-- LOOP
Sensor Value: 97.00
-- gasDetailR0
Sensor_volt = 0.47
RS_ratio = 9.67
Rs/R0 = 96.67


Sending: mac tx uncnf 1 00602E03C625C2
Successful transmission
```

Payload auf TTN Dekodieren



```
Response is not OK: invalid_param
-- LOOP
Sensor Value: 97.00
-- gasDetailR0
Sensor_volt = 0.47
RS_ratio = 9.67
Rs/R0 = 96.67


Sending: mac tx uncnf 1 00602E03C625C2
Successful transmission
```

```
//}
function Decoder(bytes, port) {
  var led = (bytes[0]);
  var sensor = (bytes[1]);
  var s_volt = (bytes[2]);
  var ratio = (bytes[3])<<8 | bytes [4];
  var rsr0 = (bytes[5])<<8 | bytes [6];



    return {
      LED: led ,
      SENSOR: sensor,
      S_VOLT: s_volt/100,
      RATIO: ratio/100,
      RSR0: rsr0/100,

//          if(smoke > 200),(" | Smoke detected!"))
```

Test

Byte payload: 00 60 2E 03 C6 25 C2

FPort: 1

Decoded test payload

```
{
  "LED": 0,
  "RATIO": 9.66,
  "RSR0": 96.66,
  "SENSOR": 96,
  "S_VOLT": 0.46
}
```

Complete uplink data

```
{
  "f_port": 1,
  "frm_payload": "AGAuA8Ylwg==",
  "decoded_payload": {
    "LED": 0,
    "RATIO": 9.66,
    "RSR0": 96.66,
    "SENSOR": 96,
    "S_VOLT": 0.46
  },
  "rx_metadata": [
```

✓ Payload is valid

```
↑ 10:14:11  Forward uplink data message    DevAddr: 26 0B BB 2A    Payload: { LED: 0, RATIO: 9.66, RSR0: 96.66, SENSOR: 96, S_VOLT: 0.46 }  00 60 2E 03 C6 25 C2    FPort: 1
```