

Temperaturkontrolle mit dem ESP8266 über WLAN

Übersicht: Elf Schritte für die Hard- und Software-Einrichtung

Die folgenden Seiten sind auf den ersten Blick für Einsteiger wahrscheinlich kaum zu entschlüsseln, mit einiger Geduld sollte sich aber hoffentlich erkennen lassen, dass sich mit 10 € für den **Board ESP8266** (z.B. Bauform LoLin NodeMcu V3 – wie hier verwendet), den **Sensor (DS1820 b)**, einem **10 Kiloohm Widerstand** und das **Relais** (1-Kanal Relais-Modul 5V für Arduino) etc. das beschriebene System in seinen Grundfunktionen relativ zügig zum Laufen bringen lässt.

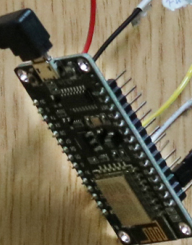
Unverzichtbar die Vertrautheit mit der Arduino-Entwicklungsumgebung (IDE)!

Wenn die entsprechende Vorbereitungen getroffen sind, dann sind folgende Schritte auszuführen:

1. Rechner starten und die darauf installierte Arduino-IDE starten.
2. Nachschauen, ob die IDE für das Board vom Typ „Generic ESP8266 Module“ vorbereitet ist, das ggf. über die „Voreinstellungen“ nachholen, siehe folgende Seiten.
3. Den Code in das IDE-Fenster hineinkopieren und mit „Überprüfen/Kompilieren“ testen, ob alle Bibliotheken schon in die IDE integriert sind. Die Informationen dazu erhält man in roter Schrift im schwarzen Feld oder dem Code-Fenster
4. Fehlendes über „Sketch => Bibliothek einbinden“ installieren.
5. Wenn das Kompilieren ohne Fehlermeldungen (Warnungen muss man erst einmal nicht beachten) ausgeführt wird, das Board über USB anschließen und den richtigen Port einstellen.
6. Richtiges WLAN mit Namen und Schlüssel in den Code eintragen.
7. Sketch (Code) hochladen.
8. Seriellen Monitor öffnen.
9. Reset-Taster auf dem ESP drücken. Wenn die IP-Adresse im Monitorfenster ausgegeben wird, diese kopieren und in den Browser eingeben. Die Website müsste sich öffnen mit -127°C als „Ist-Temperatur“. Falls im Monitorfenster keine IP-Adresse erscheint - oder die Website sich nicht öffnet, auf Fehlersuche gehen.
10. DS1820 anbringen – siehe folgende Seiten, kontrollieren.
11. Relais anbringen und kontrollieren, siehe folgende Seiten.



Lüsterklemme mit
10-Kilohm-Widerstand



Regelung mit dem ESP8266

**Soll-Temperatur
in °C eingeben:**

25

Button
Einstellen mit Sollwert
Abfragen ohne Wert

Aktuelle Werte:

Soll-Temp.: 25.00 °C
Ist-Temp.: 25.25 °C
Differenz: 0.25 °

Vom ESP als Server bereitgestellte
HTML-Seite, aufgerufen mit Browser
auf Smartphone oder Rechner

Fernsteuerung über WLAN

Soll-Temperatur

Ist-Temperatur

Temperatur-
Sensor
DS18B20

Lüsterklemme

Ist-Temperatur

Schaltbefehl

Relais

schaltet Heizung

ESP8266

Bauform LoLin
Node Mcu V3

Mikrocontroller mit Regelprogramm

Ggf. die notwendige „Zusätzliche Boardverwalter-URL hinzufügen“

The image shows the Arduino IDE interface with the 'Voreinstellungen' (Preferences) menu open. A yellow box labeled '1.' highlights the 'Voreinstellungen' menu item. A yellow box labeled '2.' highlights the 'Voreinstellungen' menu item. A yellow box labeled '3.' highlights the 'Voreinstellungen' menu item. A yellow box labeled '4.' highlights the 'Zusätzliche Boardverwalter-URLs' dialog box.

The 'Zusätzliche Boardverwalter-URLs' dialog box is open, showing a list of URLs. The URL http://arduino.esp8266.com/stable/package_esp8266com_index.json is highlighted in blue. The dialog box also contains the text 'Geben Sie weitere URLs ein, eine in jeder Zeile' and 'Klicken Sie für eine Liste mit inoffiziellen Boardunterstützungs-URLs'.

The 'Voreinstellungen' dialog box is also visible, showing various settings. The 'Zusätzliche Boardverwalter-URLs' field is highlighted in blue, showing the URL http://arduino.esp8266.com/stable/package_esp8266com_index.json.

http://arduino.esp8266.com/stable/package_esp8266com_index.json

Das richtige Board auswählen!

1.

TempLoLinHtmlDS1820R | aisFritzSchrift2 | Arduino 1.8.12

Datei Bearbeiten Sketch Werkzeuge Hilfe



TempLoLinHtmlDS182

```
#include<ESP8266W
#include<ESP8266W
#include<ESP8266W
#include<ESP8266W
#include<WiFiServ
#include<Wire.h>
#include<OneWire.
#include<DallasTe
#include<WiFiServ
#include<BearSSLH
//LoLin NodeMcu:2
float tempNeu = 0
OneWire oneWire(S
String TempSoll=
```

2.

Board: "Generic ESP8266 Module"

Builtin Led: "2"

Upload Speed: "115200"

CPU Frequency: "80 MHz"

Crystal Frequency: "26 MHz"

Flash Size: "1MB (FS:64KB OTA:~470KB)"

Strg+T

Automatische Formatierung

Sketch archivieren

Kodierung korrigieren & neu laden

Bibliotheken verwalten...

Strg+Umschalt+I

Serieller Monitor

Strg+Umschalt+M

Serieller Plotter

Strg+Umschalt+L

WiFi101 / WiFinINA Firmware Updater

ArduBlock

Boardverwalter...

VintLabs ESP32 Devkit

Robot Zero One

ESP8266 Boards (2.6.2)

• Generic ESP8266 Module

Generic ESP8285 Module

ESPDuino (ESP-13 Module)

Ad-Fruit Feather-LU77411-ESP8266

3.

Generic ESP8266 Module

Das gelingt nur, wenn die dazu notwendige Software bereits in die Entwicklungsumgebung aufgenommen worden ist. Andernfalls muss die Aufnahme über die Option „Boardverwalter“ durchgeführt werden.

Den Code kopieren und in das Codefenster einfügen und versuchen zu kompilieren!

```
TempLoLinHtmlDS1820Relais | Arduino 1.8.12
Datei Bearbeiten Sketch Werkzeuge Hilfe

TempLoLinHtmlDS1820Relais

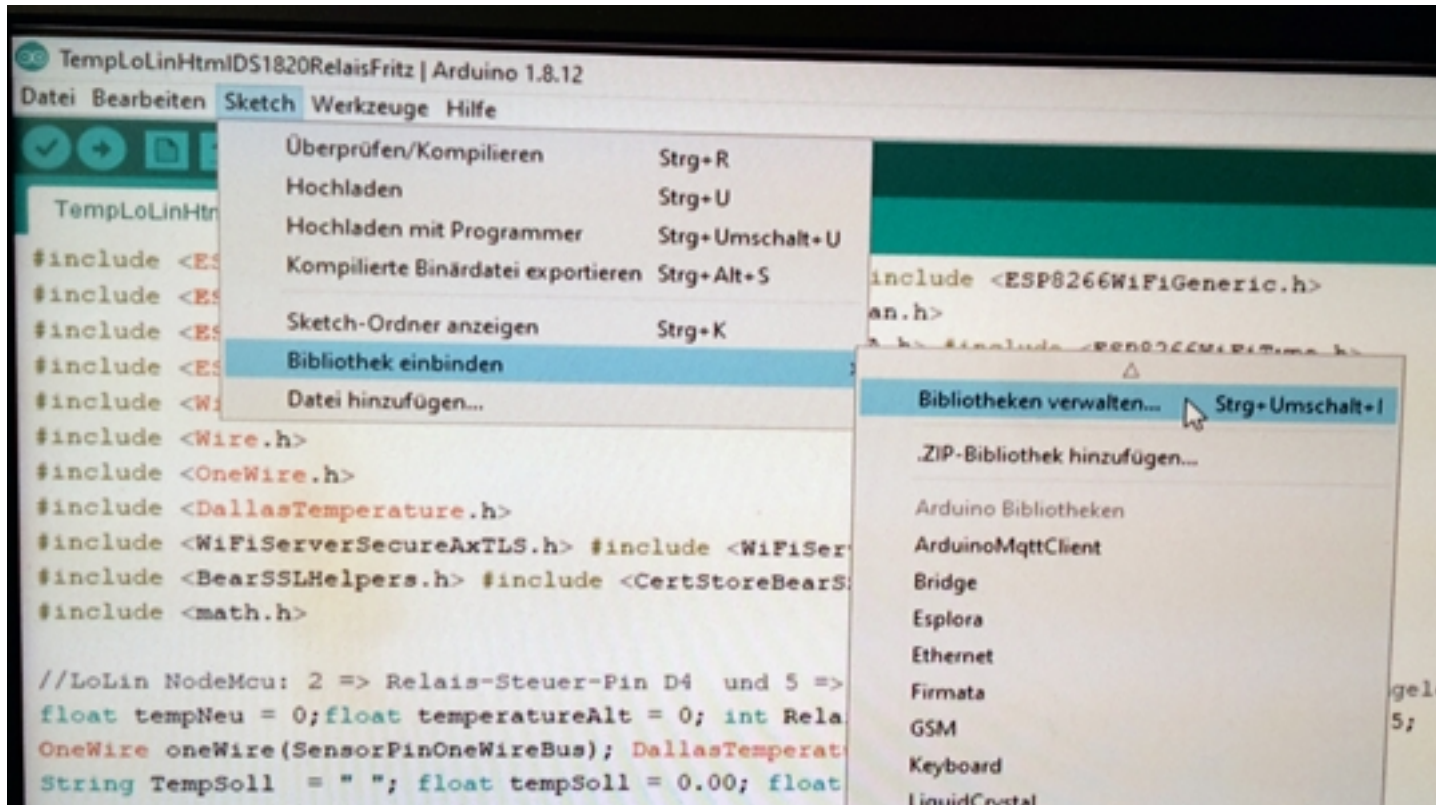
#include<ESP8266WiFi.h> #include<ESP8266WiFiAP.h> #include<ESP8266WiFiGeneric.h> #include<ESP8266WiFiMulti.h> #include<ESP8266WiFiScan.h>
#include<ESP8266WebServer.h> #include<ESP8266WiFiSTA.h> #include<ESP8266WiFiType.h>
#include<ESP8266WiFi.h> #include<WiFiClient.h> #include<WiFiClientSecure.h>
#include<WiFiServer.h> #include<WiFiServerSecure.h> #include<WiFiUdp.h>
#include<Wire.h>
#include<OneWire.h>
#include<DallasTemperature.h>
#include<WiFiServerSecureAxTLS.h> #include<WiFiServerSecureBearSSL.h>
#include<BearSSLHelpers.h> #include<CertStoreBearSSL.h> #include<math.h>
//LoLin NodeMcu:2=>Relais-Steuer-Pin D4, 5=>Temperatur-Daten-Pin=D1(intern umgeleitet)
float tempNeu=0;float temperatureAlt=0;int RelaisPin=2;int SensorPinOneWireBus=5;
OneWire oneWire(SensorPinOneWireBus);DallasTemperature sensors(&oneWire);
String TempSoll= " ";float tempSoll=21.00;float deltaTemp=0;int delta=0;
ESP8266WebServer server(80);//Sollwert-Eingabe
const char HTML_ANFANG[]="<!DOCTYPE html><head>"
"<meta name = 'viewport' content = 'width=device-width,initial-scale=1.0,maximum-scale=1.0'>"
"<meta http-equiv='refresh'content='20'><title>ESP8266 mit DS1820-Temperatursensor und Relais</title></head><body><center>"
"<FORM style='font-family:arial'action='method='post'<h4>Regelung mit dem ESP8266</h4>"
"<h2>Soll-Temperatur<br>in &degC eingeben:</h2><h1><INPUT = text style='text-align:center;font-size:30px'size='10'name='message'><br><br>"
"<Button style='font-size:18px'class='Button'name='submit'>Button<br>Einstellen mit Sollwert<br>Abfragen ohne Wert</button></P></h1></FORM>";
const char HTML_ENDE[]="</body></center></html>";
String messwertTabelle() //Messwert-Ausgabe:
{String tabl="<h2 style='font-family:arial'>Aktuelle Werte:</h2>";tabl=tabl+"<table style='font-family:arial'>";
tabl=tabl+"<tr><td>Soll-Temp.:</td><td>"+String(tempSoll)+"</td><td>&degC</td></tr>";
tabl=tabl+"<tr><td> Ist-Temp.:</td><td>"+String(tempNeu)+"</td><td>&degC</td></tr>";
tabl=tabl+"<tr><td>Differenz:</td><td>"+String(deltaTemp)+"</td><td>&degC</td></tr>";tabl=tabl+"</table>";return tabl;};
void HTML_Server()//Wenn Text vom Client eingetroffen, HTML-Seite neu zusammenbauen:
{if (server.hasArg("message")&& server.arg("message")!="")
(TempSoll=server.arg("message");tempSoll=TempSoll.toFloat();deltaTemp=tempNeu-tempSoll;};
String htm=String(HTML_ANFANG)+messwertTabelle()+String(HTML_ENDE);server.send(200,"text/html",htm);}
void setup()//WLAN, Server und I2C-Bus starten
{pinMode(RelaisPin, OUTPUT);digitalWrite(RelaisPin,LOW);WiFi.persistent(false);WiFi.mode(WIFI_STA);delay(100);
WiFi.begin("hegburk","202187130877");Serial.begin(9600);while(WiFi.status()!=WL_CONNECTED){delay(500);Serial.print(".");};
WiFi.mode(WIFI_STA);WiFi.hostname("ESP8266");Serial.println("\nIP: "+ WiFi.localIP().toString());delay(1000);
server.on("/", HTML_Server); server.begin();Wire.begin(); WiFi.persistent(false);WiFi.mode(WIFI_STA);delay(100);}
void loop() {sensors.requestTemperatures(); temperatureAlt = tempNeu;tempNeu=sensors.getTempCByIndex(0);//Dauerschleife
deltaTemp=tempNeu-tempSoll;if(abs(deltaTemp)>1)Serial.println(deltaTemp);Serial.println(deltaTemp);
String htm=String(HTML_ANFANG)+messwertTabelle()+String(HTML_ENDE);
if (deltaTemp > 0){digitalWrite(RelaisPin, HIGH);}; if (deltaTemp <= 0){digitalWrite(RelaisPin, LOW);};
delay(1000); server.handleClient();delay(100);}
```

Speichern abgeschlossen.

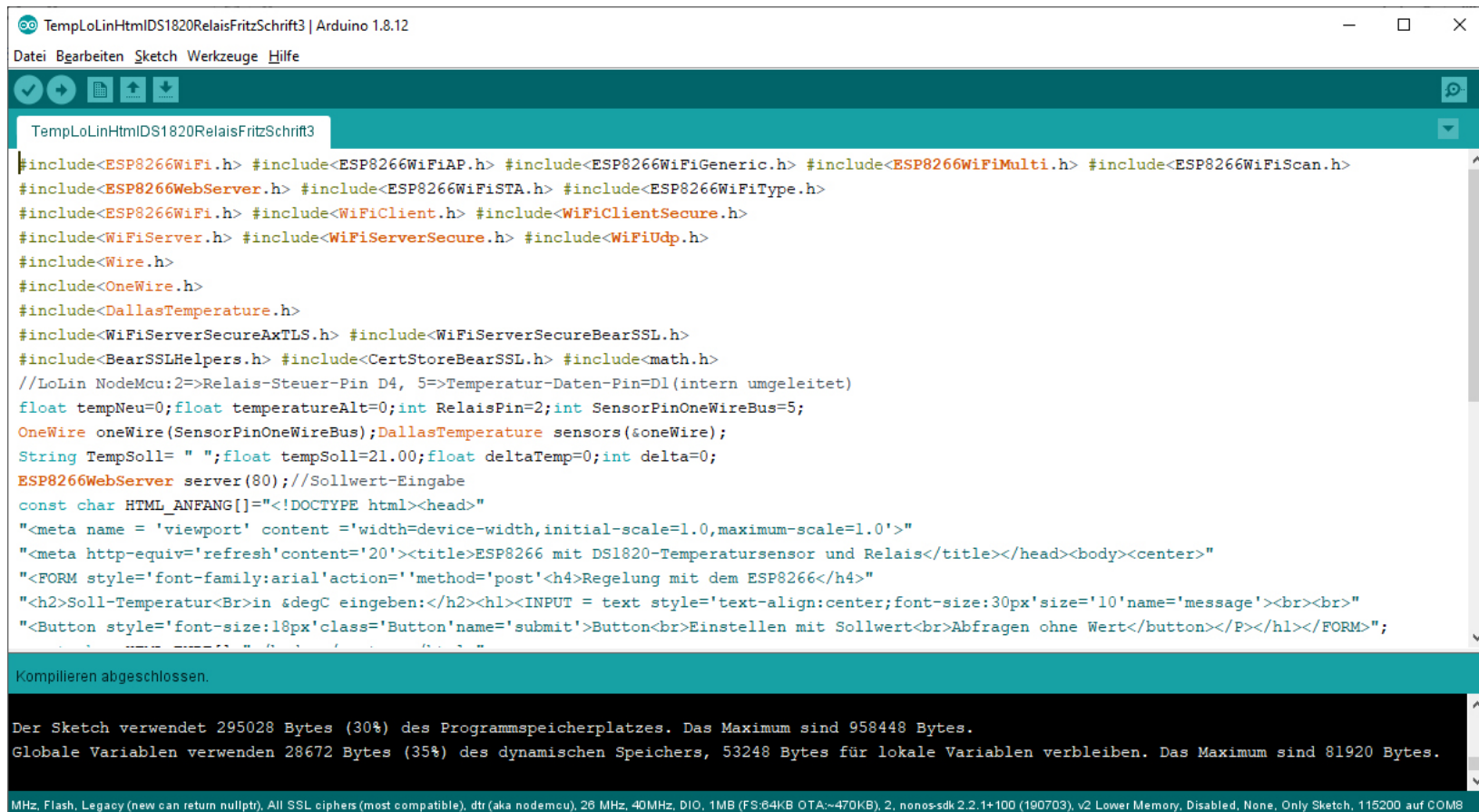
Dieser Code kann aus dem Internet heruntergeladen oder aus dem im Anhang kopiert werden.

Fehlende Bibliotheken notieren!

Fehlende Bibliotheken einbinden!



Erfolgreich hochgeladen!



```
TempLoLinHtmIDS1820RelaisFritzSchrift3 | Arduino 1.8.12
Datei Bearbeiten Sketch Werkzeuge Hilfe

TempLoLinHtmIDS1820RelaisFritzSchrift3

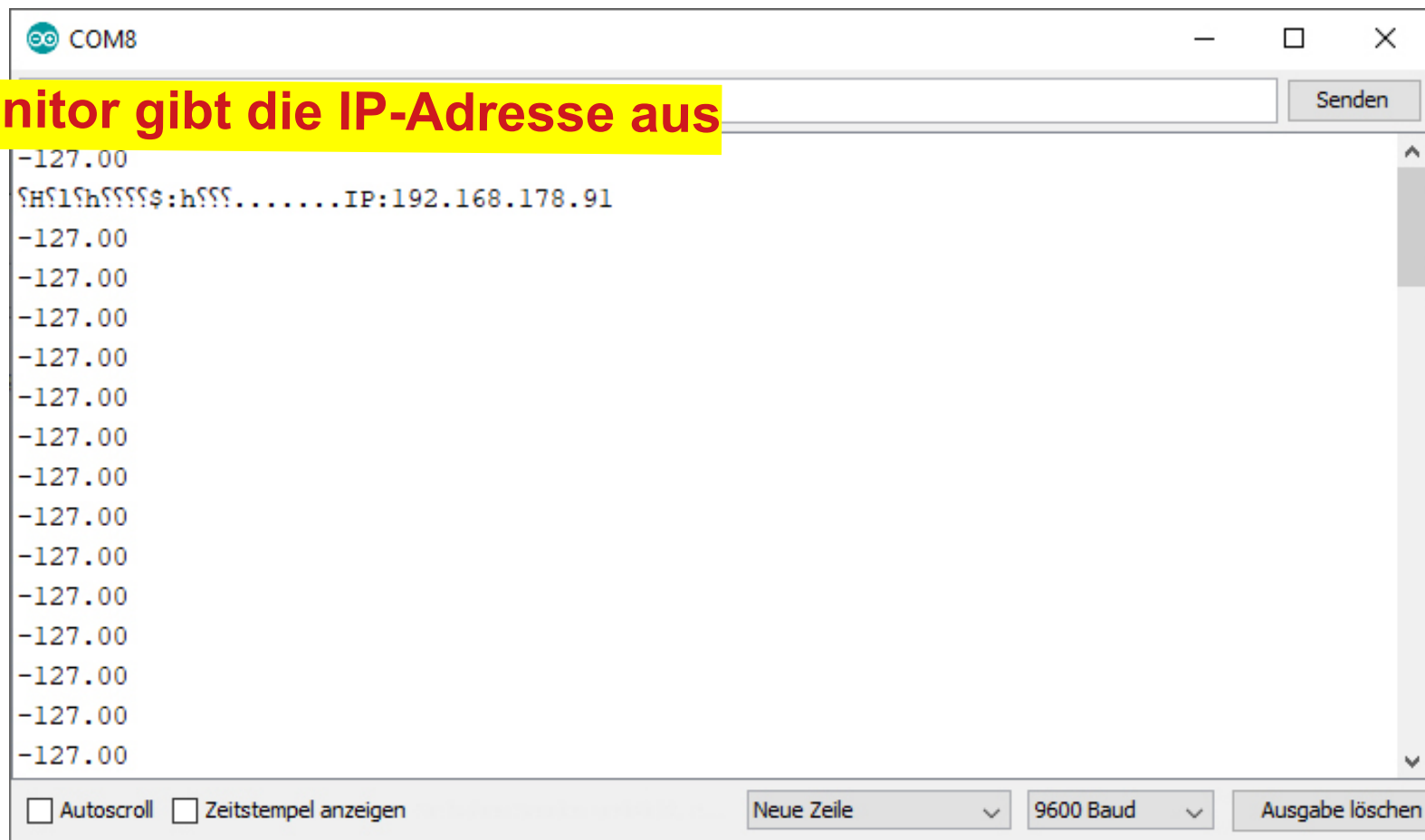
#include<ESP8266WiFi.h> #include<ESP8266WiFiAP.h> #include<ESP8266WiFiGeneric.h> #include<ESP8266WiFiMulti.h> #include<ESP8266WiFiScan.h>
#include<ESP8266WebServer.h> #include<ESP8266WiFiSTA.h> #include<ESP8266WiFiType.h>
#include<ESP8266WiFi.h> #include<WiFiClient.h> #include<WiFiClientSecure.h>
#include<WiFiServer.h> #include<WiFiServerSecure.h> #include<WiFiUdp.h>
#include<Wire.h>
#include<OneWire.h>
#include<DallasTemperature.h>
#include<WiFiServerSecureAxTLS.h> #include<WiFiServerSecureBearSSL.h>
#include<BearSSLHelpers.h> #include<CertStoreBearSSL.h> #include<math.h>
//LoLin NodeMcu:2=>Relais-Steuer-Pin D4, 5=>Temperatur-Daten-Pin=D1(intern umgeleitet)
float tempNeu=0;float temperatureAlt=0;int RelaisPin=2;int SensorPinOneWireBus=5;
OneWire oneWire(SensorPinOneWireBus);DallasTemperature sensors(&oneWire);
String TempSoll= " ";float tempSoll=21.00;float deltaTemp=0;int delta=0;
ESP8266WebServer server(80);//Sollwert-Eingabe
const char HTML_ANFANG[]="<!DOCTYPE html><head>"
"<meta name = 'viewport' content = 'width=device-width,initial-scale=1.0,maximum-scale=1.0'>"
"<meta http-equiv='refresh'content='20'><title>ESP8266 mit DS1820-Temperatursensor und Relais</title></head><body><center>"
"<FORM style='font-family:arial'action='method='post'<h4>Regelung mit dem ESP8266</h4>"
"<h2>Soll-Temperatur<br>in &degC eingeben:</h2><hl><INPUT = text style='text-align:center;font-size:30px'>size='10'>name='message'><br><br>"
"<Button style='font-size:18px'class='Button'>name='submit'>Button<br>Einstellen mit Sollwert<br>Abfragen ohne Wert</button></P></hl></FORM>";

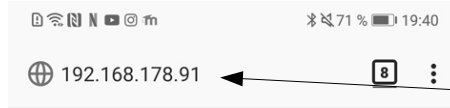
Kompilieren abgeschlossen.

Der Sketch verwendet 295028 Bytes (30%) des Programmspeicherplatzes. Das Maximum sind 958448 Bytes.
Globale Variablen verwenden 28672 Bytes (35%) des dynamischen Speichers, 53248 Bytes für lokale Variablen verbleiben. Das Maximum sind 81920 Bytes.

MHz, Flash, Legacy (new can return nullptr), All SSL ciphers (most compatible), dtr (aka nodemcu), 26 MHz, 40MHz, DIO, 1MB (FS:64KB OTA:~470KB), 2, nonos-sdk 2.2.1+100 (190703), v2 Lower Memory, Disabled, None, Only Sketch, 115200 auf COM8
```

Serieller Monitor gibt die IP-Adresse aus





**Website im lokalen WLAN
- noch ohne Sensor**

Temperatur-Regelung mit dem ESP8266

Soll-Temperatur Eingabe:

Senden!

Aktuelle Werte:

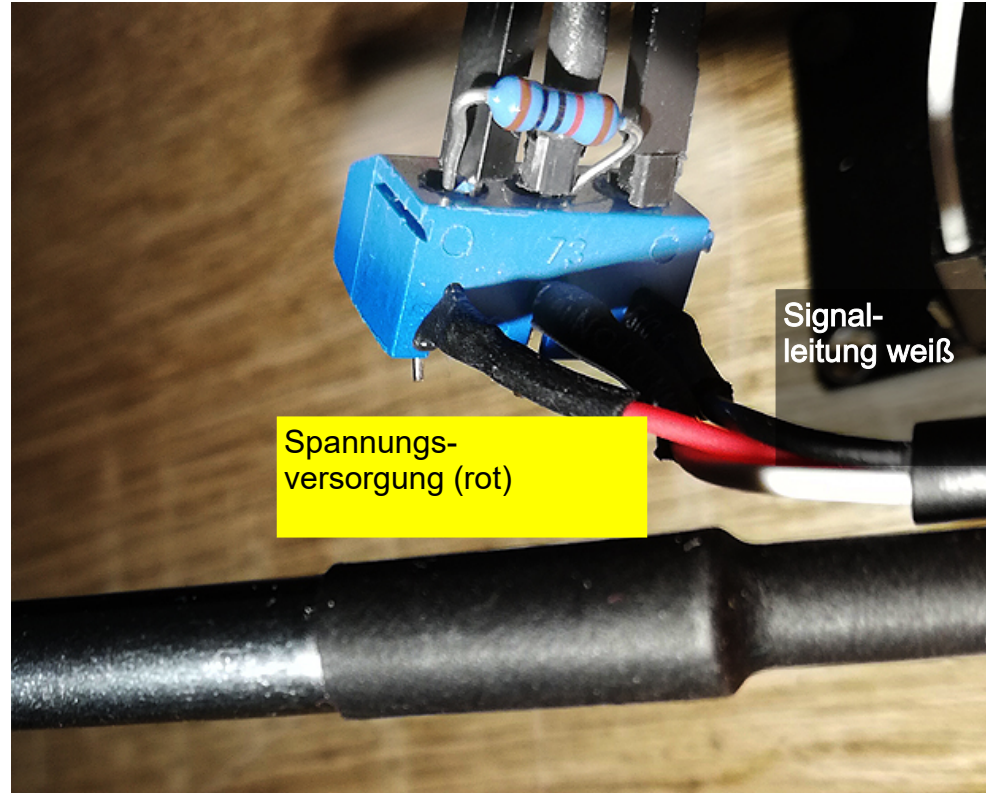
Temp-soll: 0.00 Grad Celsius

Temp-ist: -127.00 Grad Celsius

Differenz: -127.00 Grad



Zwischen der Spannungsversorgung und der Signalleitung muss ein 10 k Ω montiert werden.



Das Relais wird über Spannungsversorgung, Ground (Masse, 0V) und Steuerleitung, hier gelb, angeschlossen.



Anhang: Der Code zum Kopieren

```
#include<ESP8266WiFi.h> #include<ESP8266WiFiAP.h> #include<ESP8266WiFiGeneric.h> #include<ESP8266WiFiMulti.h> #include<ESP8266WiFiScan.h>
#include<ESP8266WebServer.h> #include<ESP8266WiFiSTA.h> #include<ESP8266WiFiType.h>
#include<ESP8266WiFi.h> #include<WiFiClient.h> #include<WiFiClientSecure.h>
#include<WiFiServer.h> #include<WiFiServerSecure.h> #include<WiFiUdp.h>
#include<Wire.h>
#include<OneWire.h>
#include<DallasTemperature.h>
#include<WiFiServerSecureAxTLS.h> #include<WiFiServerSecureBearSSL.h>
#include<BearSSLHelpers.h> #include<CertStoreBearSSL.h> #include<math.h>
//LoLin NodeMcu:2=>Relais-Steuer-Pin D4, 5=>Temperatur-Daten-Pin=D1(intern umgeleitet)
float tempNeu=0;float temperatureAlt=0;int RelaisPin=2;int SensorPinOneWireBus=5;
OneWire oneWire(SensorPinOneWireBus);DallasTemperature sensors(&oneWire);
String TempSoll= " ";float tempSoll=21.00;float deltaTemp=0;int delta=0;
ESP8266WebServer server(80);//Sollwert-Eingabe
const char HTML_ANFANG[]="<!DOCTYPE html><head>"
"<meta name = 'viewport' content = 'width=device-width,initial-scale=1.0,maximum-scale=1.0'>"
"<meta http-equiv='refresh'content='20'><title>ESP8266 mit DS1820-Temperatursensor und Relais</title></head><body><center>"
"<FORM style='font-family:arial'action='method='post'<h4>Regelung mit dem ESP8266</h4>"
"<h2>Soll-Temperatur<br>in &degC eingeben:</h2><h1><INPUT = text style='text-align:center;font-size:30px'size='10'>name='message'><br><br>"
"<Button style='font-size:18px'class='Button'>name='submit'>Button<br>Einstellen mit Sollwert<br>Abfragen ohne Wert</button></P></h1></FORM>";
const char HTML_ENDE[]="</body></center></html>";
String messwertTabelle() //Messwert-Ausgabe:
{String tabl="<h2 style='font-family:arial'>Aktuelle Werte:</h2>";tabl=tabl+"<table style='font-family:arial'>";
tabl=tabl+"<tr><td> Soll-Temp.:</td><td>"+String(tempSoll)+"</td><td>&degC</td></tr>";
tabl=tabl+"<tr><td> Ist-Temp.:</td><td>"+String(tempNeu)+"</td><td>&degC</td></tr>";
tabl=tabl+"<tr><td>Differenz:</td><td>"+String(deltaTemp)+"</td><td>&degC</td></tr>";tabl=tabl+"</table>";return tabl;};
void HTML_Server()//Wenn Text vom Client eingetroffen, HTML-Seite neu zusammenbauen:
{if (server.hasArg("message")&& server.arg("message")!="")
{TempSoll=server.arg("message");tempSoll=TempSoll.toFloat();deltaTemp=tempNeu-tempSoll;}
String htm=String(HTML_ANFANG)+messwertTabelle()+String(HTML_ENDE);server.send(200,"text/html",htm);}
void setup()//WLAN, Server und I2C-Bus starten
{pinMode(RelaisPin, OUTPUT);digitalWrite(RelaisPin,LOW);WiFi.persistent(false);WiFi.mode(WIFI_STA);delay(100);
WiFi.begin("hegburk","202187130877");Serial.begin(9600);while(WiFi.status()!=WL_CONNECTED) {delay(500); Serial.print(".");};
WiFi.mode(WIFI_STA);WiFi.hostname("ESP8266");Serial.println("IP:"+ WiFi.localIP().toString());delay(1000);
server.on("/", HTML_Server); server.begin();Wire.begin(); WiFi.persistent(false);WiFi.mode(WIFI_STA);delay(100);}
void loop() {sensors.requestTemperatures(); temperatureAlt = tempNeu;tempNeu=sensors.getTempCByIndex(0);//Dauerschleife
deltaTemp=tempNeu-tempSoll;if(abs(deltaTemp)>1)Serial.println(deltaTemp);Serial.println(deltaTemp);
String htm=String(HTML_ANFANG)+messwertTabelle()+String(HTML_ENDE);
if (deltaTemp > 0){digitalWrite(RelaisPin, LOW);}; if (deltaTemp <= 0){digitalWrite(RelaisPin, HIGH);};
delay(1000); server.handleClient();delay(100);}
```