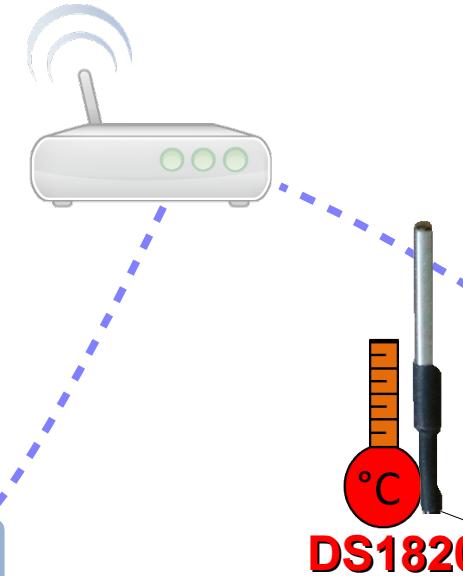
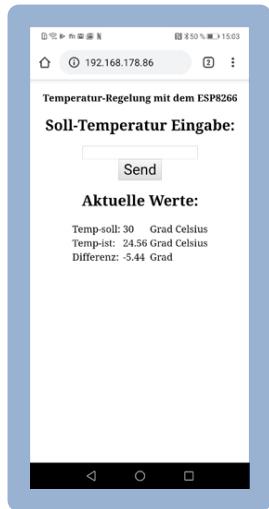


Smarthome-Modelle
für Maker und Unterricht:
Temperatur-Kontrolle
über WLAN mit
ESP8266 und DS1820



ESP8266



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 Kilohm Widerstand** und das **Relais** (1-Kanal Relais-Modul 5V für Arduino) etc. das beschriebene System in seinen Grundfunktionen relativ zügig zum Laufen bringen lässt.

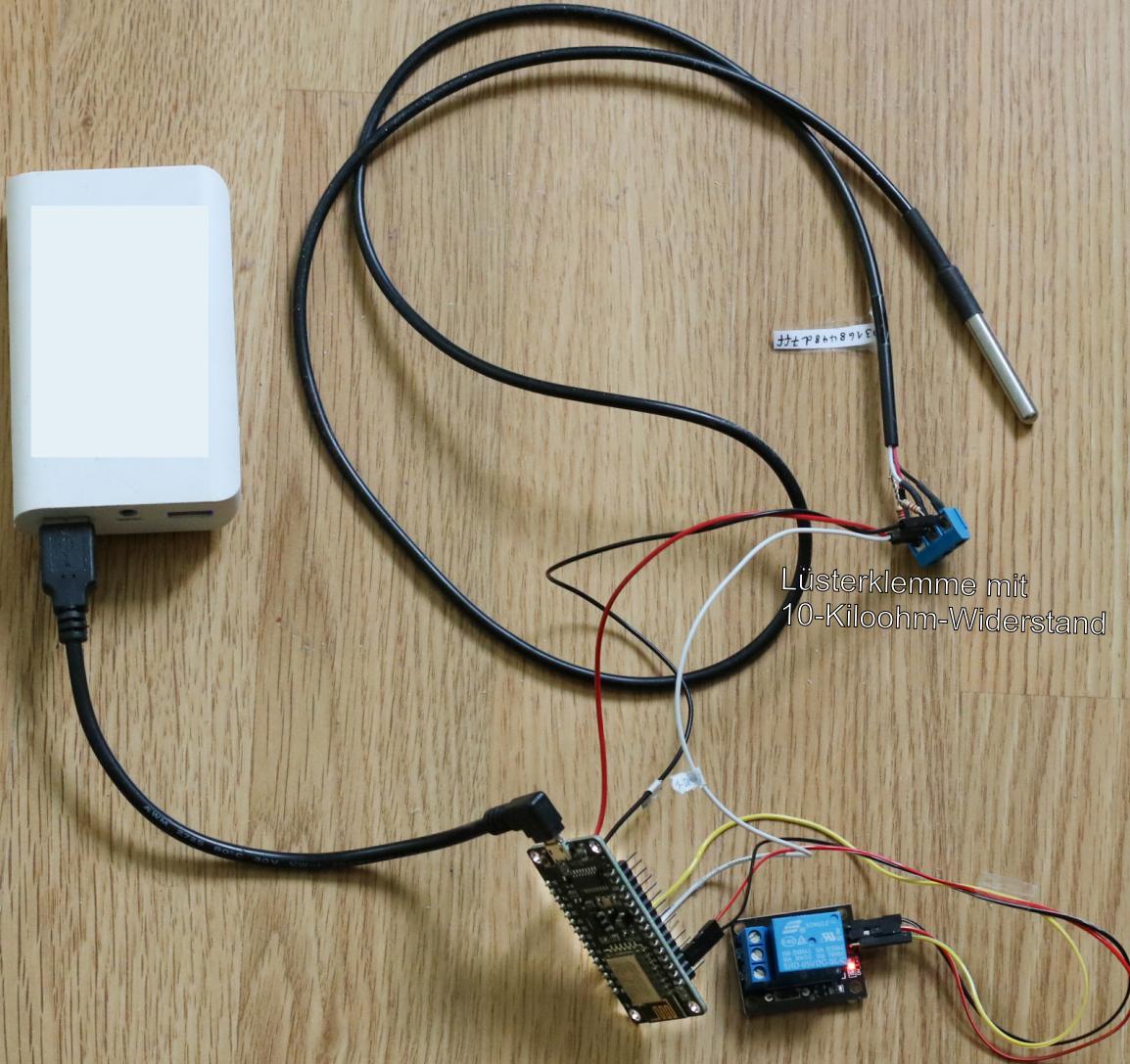
Unverzichtbar die Vertrautheit mit der Arduino-Entwicklungsumgebung (IDE)!

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.

3A6844847FF

Lüsterklemme mit
10-Kilohm-Widerstand



71% 4:12 PM

esp8266

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

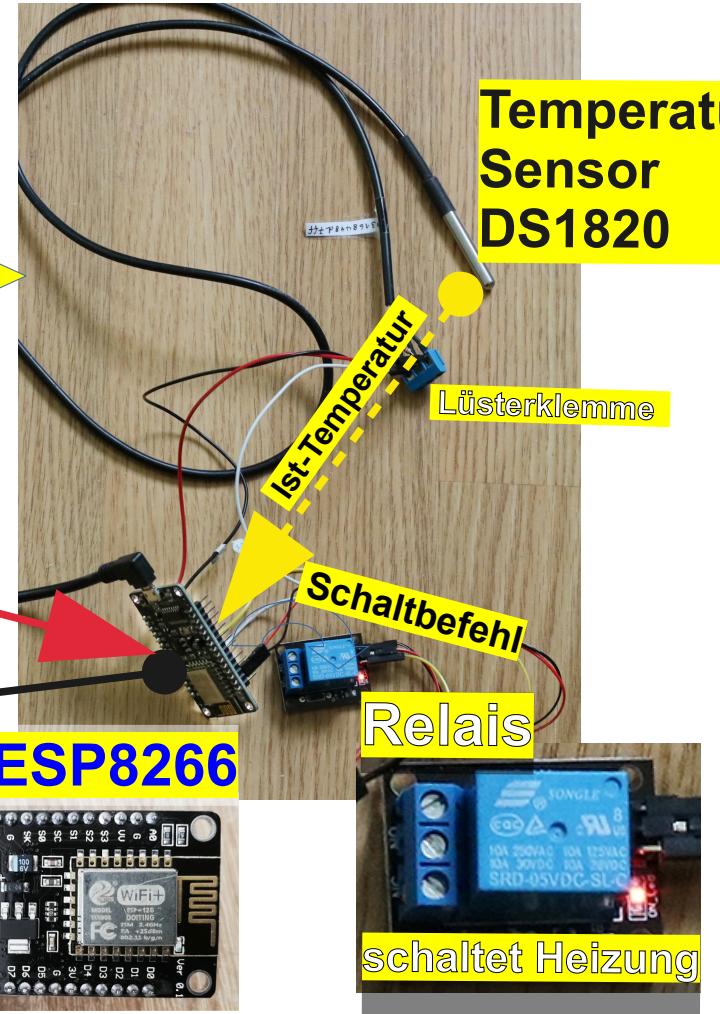
Fernsteuerung über WLAN

Soll-Temperatur

Ist-Temperatur

Bauform LoLin
Node Mcu V3

Mikrocontroller mit Regelprogramm



Temperatur-Sensor DS1820

Lüsterklemme

Ist-Temperatur

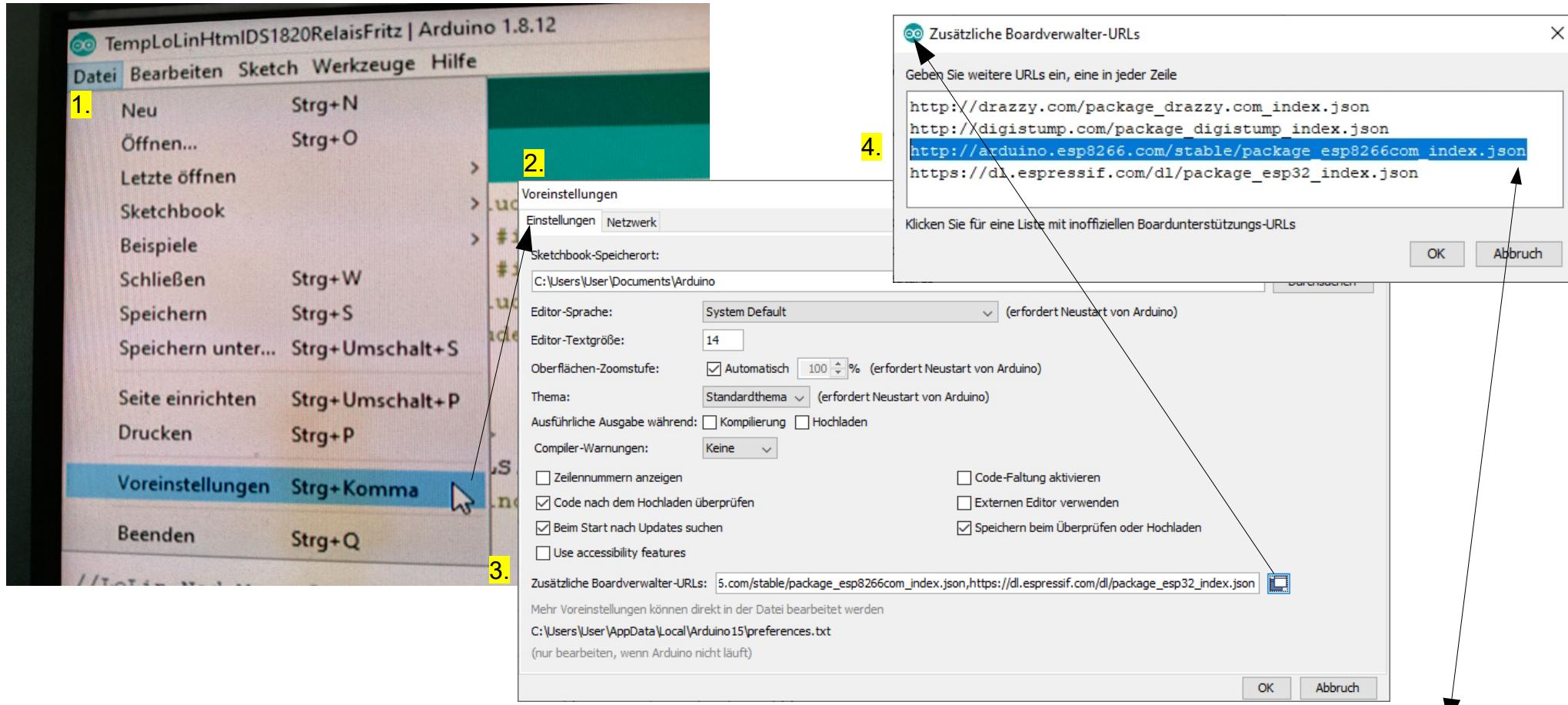
Schaltbefehl

Relais

ESP8266

schaltet Heizung

Ggf. die notwendige „Zusätzliche Boardverwalter-URL“ hinzufügen“



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

Das richtige Board auswählen!

TempLoLinHtmlDS1820RaisFritzSchrift2 | Arduino 1.8.12

Datei Bearbeiten Sketch Werkzeuge Hilfe



Automatische Formatierung Strg+T

Sketch archivieren

Kodierung korrigieren & neu laden

Bibliotheken verwalten...

Strg+Umschalt+I

FiGeneric.h>

Serieller Monitor

Strg+Umschalt+M

Serieller Plotter

Strg+Umschalt+L

WiFi101 / WiFiNINA Firmware Updater

ArduBlock

Board: "Generic ESP8266 Module"

Builtin Led: "2"

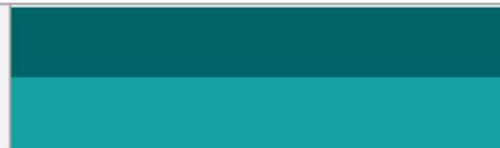
Upload Speed: "115200"

CPU Frequency: "80 MHz"

Crystal Frequency: "26 MHz"

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

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.



Boardverwalter...

VintLabs ESP32 Devkit

Robot Zero One

ESP8266 Boards (2.6.2)

Generic ESP8266 Module

Generic ESP8285 Module

ESPDuino (ESP-13 Module)

AUFGABE FÜR DEN KÜNSTLER

2.

3.

Generic ESP8266 Module

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

The screenshot shows the Arduino IDE interface with the title bar "TempLoLinHtmlIDS1820Relais | Arduino 1.8.12". The menu bar includes "Datei", "Bearbeiten", "Sketch", "Werkzeuge", and "Hilfe". Below the menu is a toolbar with icons for file operations. The main window displays the C++ code for the sketch "TempLoLinHtmlIDS1820Relais". The code includes numerous #include directives for WiFi libraries like ESP8266WiFi.h, WiFiClient.h, WiFiServer.h, WiFiUDP.h, WiFiOneWire.h, DallasTemperature.h, WiFiServerSecureAxTLS.h, WiFiServerSecureBearSSL.h, and BearSSLHelpers.h. It also defines constants for pins (RelaisPin=2, SensorPinOneWireBus=5, tempNeu=0, temperatureAlt=0, RelaisPin=2, SensorPinOneWireBus=5), initializes OneWire and DallasTemperature objects, and sets up an ESP8266WebServer on port 80. The code then handles HTTP requests to display current values and set a target temperature, and manages a relay via digital pin RelaisPin. A status message "Speichern abgeschlossen." is visible at the bottom.

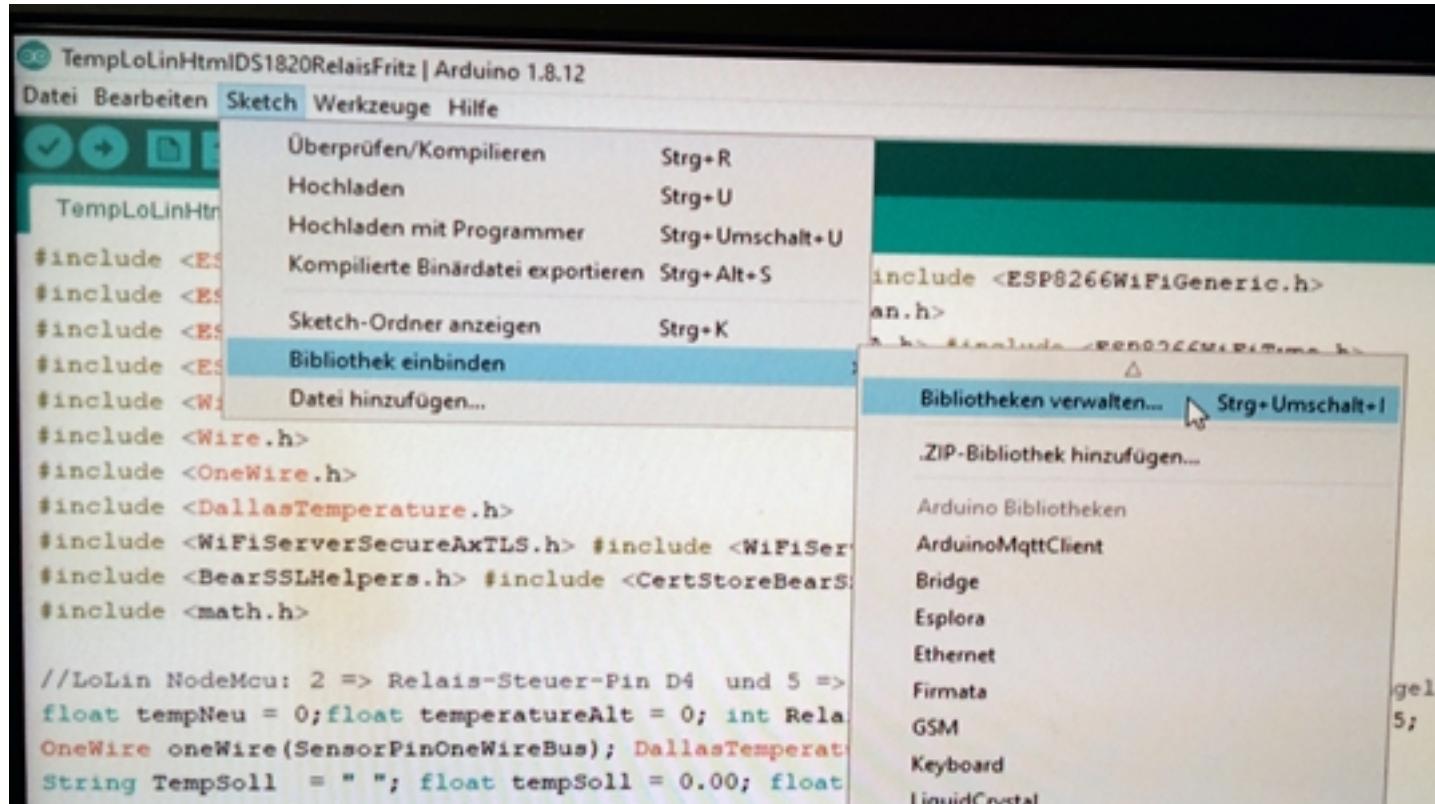
```
#include<ESP8266WiFi.h> #include<ESP8266WiFiAP.h> #include<ESP8266WiFiGeneric.h> #include<ESP8266WiFiMulti.h> #include<ESP8266WiFiScan.h>
#include<ESP8266WebServer.h> #include<ESP8266WiFiSTA.h> #include<ESP8266WiFiType.h>
#include<ESP8266WiFi.h> #include<WiFiClient.h> #include<WiFiClientSecure.h>
#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='post'><h4>Regelung mit dem ESP8266</h4>
<h2>Soll-Temperatur<br>in &degC eingeben:</h2><h1><input type='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></h1></form>";
const char HTML_ENDE[]="</body></center></html>";
String messwertTabelle() //Messwert-Ausgabe:
{String tabl=<h1 style='font-family:arial'>Aktuelle Werte:</h1>;tabl=tabl+<table style='font-family:arial'>;
tabl=tabl+<tr><td>Soll-Temp.</td><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, HIGH);} if (deltaTemp <= 0){digitalWrite(RelaisPin, LOW);}
delay(1000); server.handleClient();delay(100);}

```

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

Fehlende Bibliotheken notieren!

Fehlende Bibliotheken einbinden!



Erfolgreich hochgeladen!

COM8

- □ ×

Senden

Serieller Monitor gibt die IP-Adresse aus

-127.00
SH?1?h????\$:h????.....IP:192.168.178.91
-127.00
-127.00
-127.00
-127.00
-127.00
-127.00
-127.00
-127.00
-127.00
-127.00
-127.00
-127.00
-127.00
-127.00
-127.00
-127.00

Autoscroll Zeitstempel anzeigen

Neue Zeile

9600 Baud

Ausgabe löschen

71 % 19:40

192.168.178.91 8 :

Temperatur-Regelung mit dem ESP8266

Soll-Temperatur Eingabe:

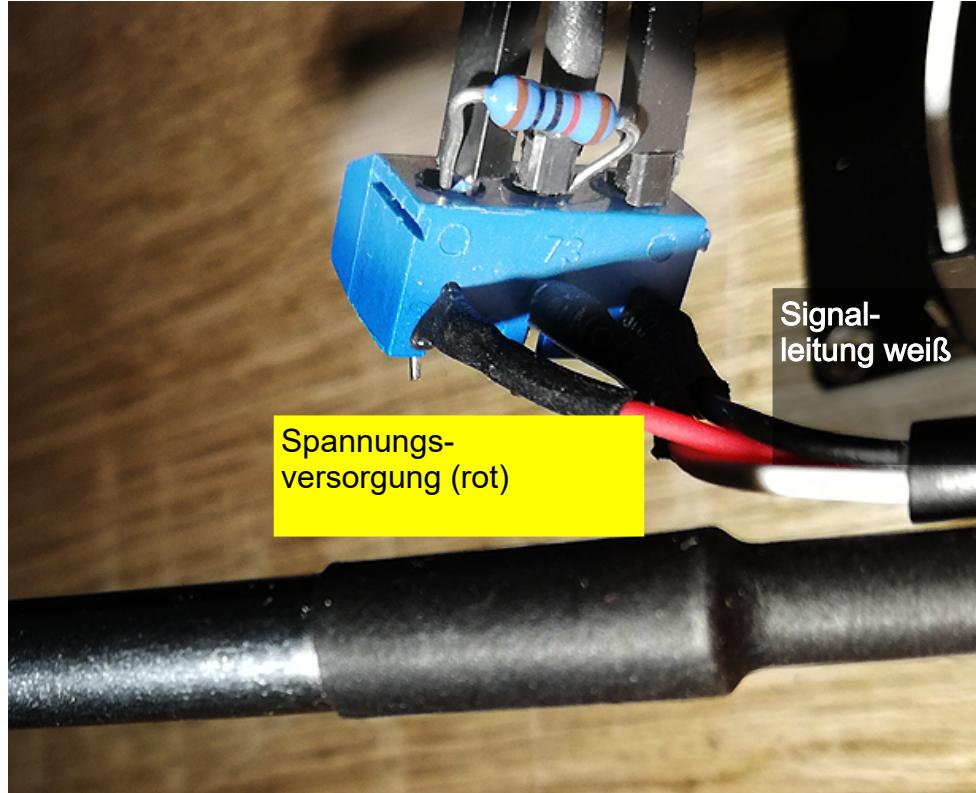
Senden!

Aktuelle Werte:

Temp-soll: 0.00 Grad Celsius
Temp-ist: -127.00 Grad Celsius
Differenz: -127.00 Grad

Website im lokalen WLAN
- noch ohne Sensor

Zwischen der Spannungsversorgung und
der Signalleitung muss ein $10\text{ k}\Omega$ 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'><input type='post'><h4>Regelung mit dem ESP8266</h4>
<h2>Soll-Temperatur<br>in &degC eingeben:</h2><h1><input style='text-align:center;font-size:30px' size='10' name='message'><br><br>
<button style='font-size:18px'>Submit</button>Einstellen mit Sollwert<br>Abfragen ohne Wert</button></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>&deg</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);}
```