

HeizungSensor2

Version 2.1.0
3/6/2022 3:05:00 PM

Inhaltsverzeichnis

Datei-Verzeichnis	2
Datei-Dokumentation	3
D:/git/DS18B20MQTT/src/HS2Hilfen.cpp	3
D:/git/DS18B20MQTT/src/main.cpp	4
D:/git/DS18B20MQTT/src/MQTT.cpp	8
Index	10

Datei-Verzeichnis

Auflistung der Dateien

Hier folgt die Aufzählung aller Dateien mit einer Kurzbeschreibung:

D:/git/DS18B20MQTT/src/HS2Hilfen.cpp (Hilfsprogramme für Heizungsensor2)3
D:/git/DS18B20MQTT/src/main.cpp (Heizung-Temperatur-Sensor ESP-01 mit DS18B20-Sensoren für Vorlauf-, Rücklauf- und Warmwasser-Temperatur)4
D:/git/DS18B20MQTT/src/MQTT.cpp (MQTT-Routinen für Heizung-Temperatur-Sensor)8

Datei-Dokumentation

D:/git/DS18B20MQTT/src/HS2Hilfen.cpp-Dateireferenz

Hilfsprogramme für Heizungsensor2.

```
#include <Arduino.h>
#include "HS2.h"
```

Funktionen

- void **BlaueLEDblinkt** ()
Blaue LED blinkt.

Ausführliche Beschreibung

Hilfsprogramme für Heizungsensor2.

Autor

Bernd-Burkhard Borys

Version

1.0

Datum

6 März 15 Jan 2022

Copyright

Copyright (c) 2022

Dokumentation der Funktionen

void **BlaueLEDblinkt** ()

Blaue LED blinkt.

D:/git/DS18B20MQTT/src/main.cpp-Dateireferenz

Heizung-Temperatur-Sensor ESP-01 mit DS18B20-Sensoren für Vorlauf-, Rücklauf- und Warmwasser-Temperatur.

```
#include <Arduino.h>
#include "HS2.h"
#include <math.h>
#include "MQTT.h"
#include <OneWire.h>
#include <DallasTemperature.h>
```

Makrodefinitionen

- `#define GPIO2 2`
alle Sensoren an GPIO2 von ESP-01.

Funktionen

- `OneWire SensorAnschluss (GPIO2)`
Anschluss über OneWire.
- `void setup ()`
Setup-Programm.
- `void loop ()`
das übliche Hauptprogramm

Variablen

- `EspMQTTClient * myClient`
MQTT-Client zur Übertragung. Parameter für WLAN und FHEM-MQTT-Broker.
- `DallasTemperature sensorenTemperatur & SensorAnschluss`
DS18B20.
- `float WwGesendet = (-200)`
- `float temp1 = (-200)`
- `float VIGemessen = (-200)`
- `float RIGemessen = (-200)`
- `float deltaGesendet = (-200)`
Temperaturen bei letzter MQTT-Nachricht. Initialisiert mit unmöglichem Wert.
- `unsigned long LetzteMQTT = 0`
Zeitpunkt letzte Übertragung.
- `unsigned long LetzteMuss = 0`
Zeitpunkt letzte Zwangsübertragung.
- `unsigned long LetzteRuecklaufMessen = 0`

Zeitpunkt letzte Messung der Rücklauftemperatur, wird jeder Minute gemessen, nach 5 Messungen gesendet.

- unsigned long **LetzteVorlaufMessen** = 0
Zeitpunkt letzte Messung der Vorlauftemperatur, wird jeder Minute gemessen, nach 5 Messungen gesendet.

Ausführliche Beschreibung

Heizung-Temperatur-Sensor ESP-01 mit DS18B20-Sensoren für Vorlauf-, Rücklauf- und Warmwasser-Temperatur.

Autor

Bernd-Burkhard Borys

Version

2.1.0

Datum

6 März 15 12 Jan 2022 20 10 Feb 2021 1 Nov 31 28 10 9 3 1 Okt Sep 2020

Copyright

Copyright (c) 2022-2021

Makro-Dokumentation

#define GPIO2 2

alle Sensoren an GPIO2 von ESP-01.

Dokumentation der Funktionen

void loop ()

das übliche Hauptprogramm

MQTT-Loop

wenn MQTT-Verbindung besteht

Mindestabstand MQTT-Übertragung 15 Sekunden in ms

Mindestabstand Rücklauf-Messung 30 s in ms

Mindestabstand Vorlauf-Messung 30 s in ms

Mittelung der Differenz über NDELTA Messungen

Mittelung der Rücklauftemperatur über NRL Messungen

Mittelung der Vorlauftemperatur über NVL Messungen

erst Abfrage aller Sensoren

nur für Warmwasser und Vor/Rück-Differenz: messen und senden, wenn

- minimaler Abstand und minimale Änderung überschritten für Differenz: NDELTA
Messungen addieren, nach der NDELTA. Messung den Mittelwert senden, wenn Delta unter 2 K: Heizung ist vermutlich aus

und DELTAABSTAND vorüber: Temperaturen abfragen und senden (funktioniert in dieser Form auch nach Überlauf von millis())

Mindeständerung in K für Warmwasser an und aus

blaue LED aus im Normalfall

dann Temperaturen einzeln und bei Änderung senden

nur für Rücklauf: messen, wenn RUECKABSTAND (1 Minute) vorüber NRL (100)
Messungen addieren nach der NRL. Messung den Mittelwert senden

RUECKABSTAND vorüber: jetzt messen (funktioniert in dieser Form auch nach Überlauf von millis())

nur für Vorlauf: messen, wenn VORABSTAND vorüber NVL Messungen addieren nach der letzten Mittelwert senden

VORABSTAND vorüber: jetzt messen (funktioniert in dieser Form auch nach Überlauf von millis())

wenn keine MQTT-Verbindung: Blaue LED blinkt

OneWire SensorAnschluss (GPIO2)

Anschluss über OneWire.

void setup ()

Setup-Programm.

blaue LED vorbereiten

und einschalten

Enable the web updater. User and password default to values of MQTTUsername and MQTTPassword. These can be overrited with enableHTTPWebUpdater("user", "password").

LWT-Meldung

Sensoren initialisieren

Variablen-Dokumentation

float deltaGesendet = (-200)

Temperaturen bei letzter MQTT-Nachricht. Initialisiert mit unmöglichem Wert.

unsigned long LetzteMQTT = 0

Zeitpunkt letzte Übertragung.

unsigned long LetzteMuss = 0

Zeitpunkt letzte Zwangsübertragung.

unsigned long LetzteRuecklaufMessen = 0

Zeitpunkt letzte Messung der Rücklauftemperatur, wird jeder Minute gemessen, nach 5 Messungen gesendet.

unsigned long LetzteVorlaufMessen = 0

Zeitpunkt letzte Messung der Vorlauftemperatur, wird jeder Minute gemessen, nach 5 Messungen gesendet.

EspMQTTClient* myClient [extern]

MQTT-Client zur Übertragung. Parameter für WLAN und FHEM-MQTT-Broker.

float RIGemessen = (-200)

DallasTemperature sensorenTemperatur& SensorAnschluss

DS18B20.

Rückgabe

DallasTemperature-Objekt

float temp1 = (-200)

float VIGemessen = (-200)

float WwGesendet = (-200)

D:/git/DS18B20MQTT/src/MQTT.cpp-Dateireferenz

MQTT-Routinen für Heizung-Temperatur-Sensor.

```
#include "MQTT.h"
```

Funktionen

- void **onConnectionEstablished** ()
MQTT-Initialisierung, nur Status-Meldung "Bereit".
- float **SendeTemp** (String messung, float temp)
Temperatur mit MQTT senden.
- void **SendeStatus** (String status)
Statusmeldung senden.

Variablen

- unsigned long **LetzteMQTT**
Zeitpunkt letzte Übertragung.
- EspMQTTClient * **myClient**
MQTT-Client zur Übertragung. Parameter für WLAN und FHEM-MQTT-Broker.

Ausführliche Beschreibung

MQTT-Routinen für Heizung-Temperatur-Sensor.

Autor

B. Borys

Version

1.1.5

Datum

6 März 15 12 Jan 2022 20 Feb 2021 31 28 10 9 3 1 Okt Sep 2020

Copyright

Copyright (c) 2022 2021 2020

Dokumentation der Funktionen

void onConnectionEstablished ()

MQTT-Initialisierung, nur Status-Meldung "Bereit".

void SendeStatus (String *status*)

Statusmeldung senden.

Parameter

<i>status</i>	String
---------------	--------

float SendeTemp (String *messung*, float *temp*)

Temperatur mit MQTT senden.

Parameter

<i>messung</i>	Bezeichnung der Messstelle
<i>temp</i>	die Temperatur, die gesendet werden soll

Rückgabe

float die Temperatur, die gesendet wurde

Variablen-Dokumentation

unsigned long LetzteMQTT [extern]

Zeitpunkt letzte Übertragung.

EspMQTTClient* myClient

```
Initialisierung:=new EspMQTTClient (
    WLANSID,
    WLAMPWD,
    MQTTBROKERIP,
    OTAUSER, OTAPASSWD,
    MQTTNAME,
    1883
)
```

MQTT-Client zur Übertragung. Parameter für WLAN und FHEM-MQTT-Broker.

Index

BlaueLEDblinkt
 HS2Hilfen.cpp 3
D:/git/DS18B20MQTT/src/HS2Hilfen.cpp 3
D:/git/DS18B20MQTT/src/main.cpp 4
D:/git/DS18B20MQTT/src/MQTT.cpp 8
deltaGesendet
 main.cpp 6
GPIO2
 main.cpp 5
HS2Hilfen.cpp
 BlaueLEDblinkt 3
LetzteMQTT
 main.cpp 6
 MQTT.cpp 9
LetzteMuss
 main.cpp 7
LetzteRuecklaufMessen
 main.cpp 7
LetzteVorlaufMessen
 main.cpp 7
loop
 main.cpp 5
main.cpp
 deltaGesendet 6
 GPIO2 5
 LetzteMQTT 6
 LetzteMuss 7
 LetzteRuecklaufMessen 7
 LetzteVorlaufMessen 7
 loop 5
 myClient 7
 RIGemessen 7
 SensorAnschluss 6, 7
 setup 6
 temp1 7
 VIGemessen 7
 WwGesendet 7
MQTT.cpp
 LetzteMQTT 9
 myClient 9
 onConnectionEstablished 8
 SendeStatus 9
 SendeTemp 9
myClient
 main.cpp 7
 MQTT.cpp 9
onConnectionEstablished
 MQTT.cpp 8
RIGemessen
 main.cpp 7
SendeStatus
 MQTT.cpp 9
SendeTemp
 MQTT.cpp 9
SensorAnschluss
 main.cpp 6, 7
setup
 main.cpp 6
temp1
 main.cpp 7
VIGemessen
 main.cpp 7
WwGesendet
 main.cpp 7