

Projekt

„Austausch einer bestehenden Thermischen Solarregelung kurz VRS Replace“



Oben Alt / unten Neu (seit November 2024)

Hinweise / Disclaimer

- Die im Folgenden mögliche Schaltung arbeitet mit 240V!
- Es handelt sich hierbei um kein Breadboard Projekt!
- Sollten Begriffe wie „Duspol“, „Schutzleiter“, „Phase(n)“, „FI“ und „Neutralleiter“ Fremdworte sein, so bitte die Idee an eine mögliche Umsetzung direkt und komplett verwerfen!
- Sollte die Stromleitung zum bisherigen/zukünftigen/auszutauschenden Solar-Regler auch nicht komplett stromlos geschaltet werden können, bitte im Allgemeinen erst für eine korrekte Installation des bestehenden/auszutauschenden Systems sorgen!
- Das im Folgenden beschriebene Projekt, protokolliert im gewissen Umfang Temperaturdaten. Diese Daten könnten u.A. Rückschlüsse auf Gewohnheiten einzelner im Haushalt lebender Personen ermöglichen! Es ist daher nicht auszuschließen, dass bei Verwendung in vermieteten Wohnräumen die DSGVO zu berücksichtigen ist!
- In meinem/unseren Fall wurde von dem damaligen Heizungsmonteuren kein abgeschirmtes Kabel bei der Installation der Temperaturfühler verwendet. Bitte berücksichtigen!
- Stand April 2025 beinhaltet der mitgelieferte Code (Version 0.9.8) keinerlei Authentifizierungsmechanismen und auch keinerlei Verschlüsselungen (SSL). Der mögliche Betrieb sollte daher nur in einer abgesicherten Umgebung in Betracht gezogen werden!
- Bei der im Folgenden beschriebenen Umsetzung handelt der Autor rein aus privatem Interesse. Es werden daher keinerlei Garantien für eine korrekte Funktionsweise und/oder irgendwelche Schäden übernommen!
- Die beschriebene Umsetzung bezieht sich rein auf die beim Autor existierende Heizungsinstallation! Es gibt kein „Schwimmbad“, „nur zwei Kollektoren“, „keine Zirkulationspumpe“, „keine Umschaltventile“, „keinen Wärmemengenzähler“, ... (siehe Heizungsinstallation)
- Dieses Dokument und der mit diesem Dokument veröffentlichte Quellcode bezieht sich derzeit einzig und allein auf die Version 0.9.8! Zukünftige Erweiterungen, Programmanpassungen, Änderungen, ... werden vorrangig auf dem zugehörigen Github Repository (siehe Link github) veröffentlicht.
- Die Beschreibung und der Quellcode unterliegen der Open Source Lizenz GPLv3 (siehe Lizenz)



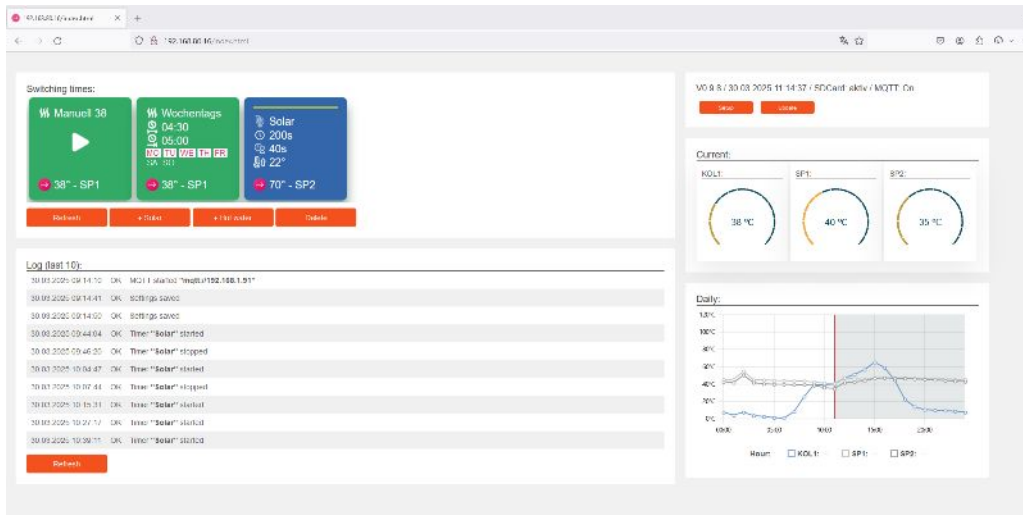
Inhaltsverzeichnis

Beschreibung Allgemein / Warum.....	4
Link github.....	4
Heizungsinstallation.....	4
Aufbau / Schaltplan.....	5
Bauteile.....	6
ADC Kalibrierung / Temperatursensoren.....	7
Installation.....	8
Kompilierung.....	8
*.bin Files.....	10
Betrieb.....	12
Konfiguration / Administration.....	12
Timer für Solarpumpe.....	14
Timer für Warmwasser.....	15
Block „Schaltzeiten“.....	16
Block „Protokoll (letzten 10)“.....	17
Block „Status“.....	17
Block „Aktuell“.....	17
Block „Tagesverlauf“.....	18
REST.....	19
/status.....	19
/log – Hier Rechtschreibung.....	20
/downloadtemperaturelog.....	21
/temperature.....	22
/config.....	23
/system.....	24
/changeconfig.....	25
/timer.....	26
/timerchange.....	28
/timerdelete.....	29
/timerdeleteall.....	29
/timerlist.....	30
/timerstart.....	31
/timerstop.....	31
/testrelay.....	32
MQTT.....	33
Prometheus metrics.....	34
Statuscodes.....	35
Zusätzlich verwendete Softwarekomponenten.....	37
Zukunft.....	37
Lizenz.....	38



Beschreibung Allgemein / Warum

Dieses „einfache“ Projekt wurde dafür erdacht, die Funktionen eines Solar-Differenztemperaturreglers (hier VRS auroMATIC 560) abzubilden und einige Dinge zu verbessern. Neben der Steuerung der Solarpumpe, können auch beliebig viele Zeitprogramme für die Warmwasseraufbereitung hinterlegt werden, es erfolgt eine Protokollierung der einzelnen Messwerte direkt auf einer SD-Karte und das System ist mittels Webbrowser per Wi-Fi steuerbar. Ferner werden die bereits verbauten Temperatursensoren (VR10 und VR11) weiterverwendet. Der Austausch durch PT1000 und/oder DS18B20 ist somit nicht notwendig und derzeit auch nicht vorgesehen.



Link github

Zusätzliche Informationen, sowie Binaries zukünftige Änderungen und Anpassungen können unter <https://github.com/kleinekuh/vrsreplace> abgerufen bzw. eingesehen werden.

Heizungsinstallation

Unsere derzeitige Heizungsanlage besteht aus folgenden Komponenten.

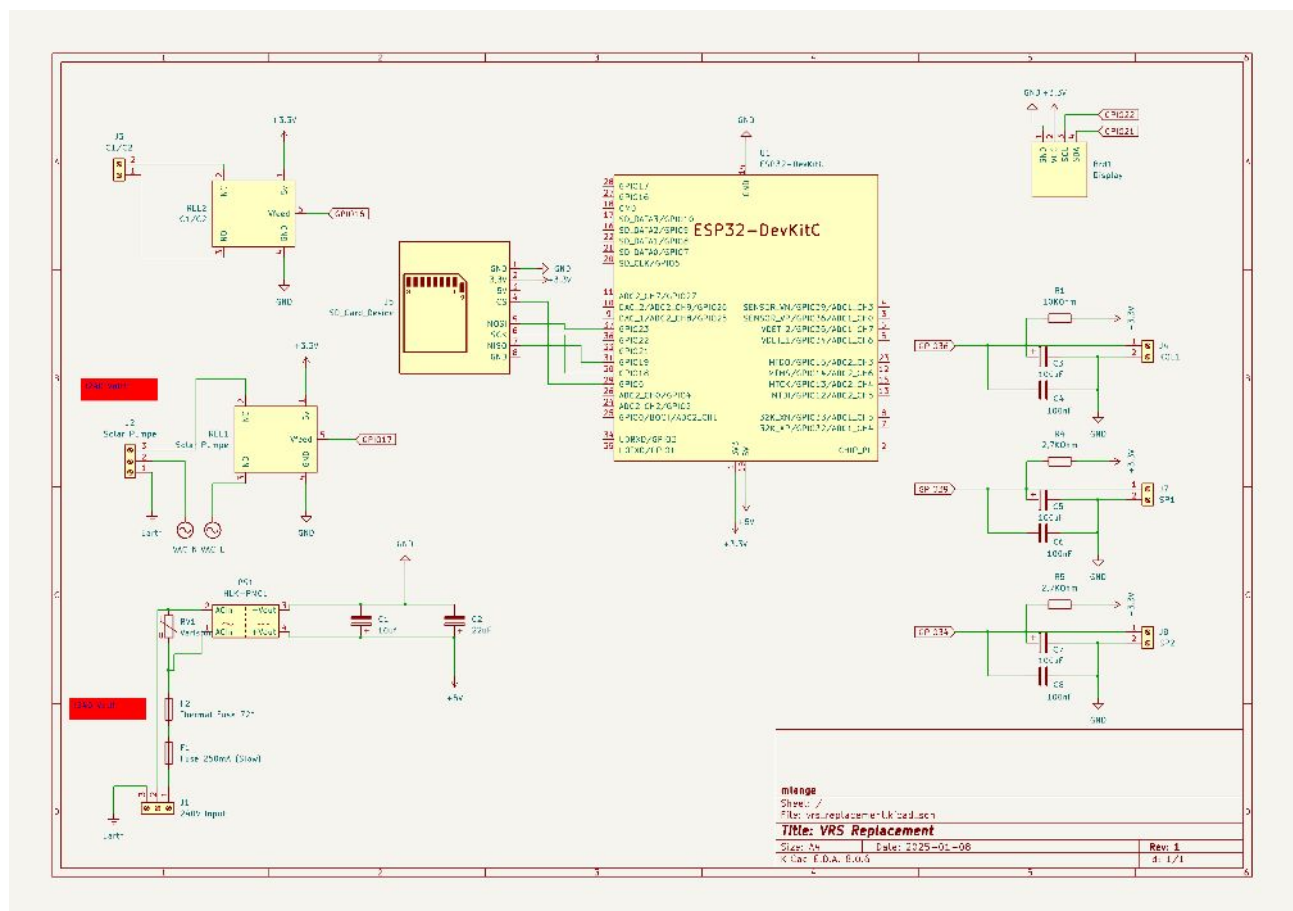
Anzahl	Bezeichnung	Beschreibung
1	VIH S 300	Bivalenter Warmwasserspeicher für Solaranlagen
1	VKK 226/2	Gas-Brennwertkessel (Heizung und Warmwasser)
1	auroMATIC 560	Solardifferenzregler
1	VR11	Temperatursensor
2	VR10	Temperatursensor
2	auroTHERM VFK 155	Solar-Flachkollektoren
1	WILO Yonos-PICO-STG	Solar-Umwälzpumpe

Wie unter „Hinweise / Disclaimer“ bereits erwähnt, besteht unser System aus keinen weiteren Komponenten. Es ist zwar eine Zirkulationsleitung vorhanden, allerdings ist hier keine entspr. Pumpe verbaut. Die projektspezifische Berücksichtigung einer Ansteuerung von Zirkulationspumpen wurde daher als nicht notwendig angenommen.



Aufbau / Schaltplan

Wie bereits unter „Hinweise / Disclaimer“ erwähnt, handelt es sich hierbei um kein Breadboard-Projekt. Aus diesem Grund wurde auch keine Steck-Verbindungs-Schaltung mittels Fritzing, o.Ä. erstellt. Der Schaltplan ist als KiCad Datei verfügbar.



Bauteile

Folgende elektronischen Bauteile wurden für die Umsetzung verwendet.

Anzahl	Bauteile
1	ESP32 DevKit C (siehe Bild)
1	SD Card Modul
1	0,96“ I2C Display (nicht unbedingt notwendig, sieht aber gut aus)
2	Vorkonfektioniertes Relais Boards. Diese können sowohl mit 3,3V aber auf 5V betrieben werden und verfügen bereits über die entspr. Elektronik.
2	Schraubterminal (3 Pins)
4	Schraubterminal (2 Pins)
1	Varistor 250V 250mw
1	Thermische Sicherung 72°C
1	Sicherung 250mA (langsam)
1	Sicherungshalter
1	10uF Kondensator
1	22uF Kondensator
3	100nF Kondensator (Keramik)
3	100uF Kondensator
2	2,7KOhm Widerstand
1	10KOhm Widerstand
1	Platinennetzteil HLK-PM01 AC DC 220V

Als Gehäuse wurde ein einfacher IP65 Abzweigkasten (Sonderpreisbaumarkt) genutzt. Die Öffnung für das Display wurde mittels einer Laubsäge erstellt und das Display mittels Schmelzkleber befestigt (siehe Bild). Diverse Kleinteile wie Draht, Platine, Steckleisten, etc. finden hier keine gesonderte Erwähnung, da diese Dinge im Fundus vorhanden waren.

Ein befreundeter Elektroniker gab den Hinweis, am Ausgang J2 auch noch einen Varistor mit einzubauen.

Sämtliche Bauteile wurden über den Onlinehandel bezogen.



ADC Kalibrierung / Temperatursensoren

Je nach verwendeten ESP kann der DAC Probleme bei der Umwandlung der jeweiligen Widerstandswerte machen. Während der Umsetzung wurden bei diversen ESP unterschiedliche Ergebnisse geliefert, was wiederum zu unterschiedlichen Temperaturen führte. Die Berechnung (Glättung) der jeweiligen Werte erfolgt nun mittels einer im Vorfeld generierten Lookup-Tabelle. Für eine passende Erzeugung der Tabellen kann entweder die Lösung von „e-thinkers“ (<https://github.com/e-tinkers/esp32-adc-calibrate>) oder eine etwas genauere Logik von „Kveri“ (<https://github.com/Kveri/esp32-adc-calibrate>) verwendet werden. Alternativ können aber auch die mit dem Quellcode-Archiv (Ordner „/sensordata“) bereitgestellten Tabellen „lut34.txt“, „lut36.txt“ und „lut39.txt“ genutzt werden. Die Tabellen beinhalten hierzu die jeweils gemessenen Mittelwerte von 8 unterschiedlichen ESP. Wichtig, bei einer eigenen Messung ist, dass die Anzahl von 4096 (Auflösung DAC) Zeilen einzuhalten sind und die Werte alle als Integer vorliegen müssen.

Bei der Zuordnung der jeweiligen Widerstandswerte von den Temperatursensoren (VR10 / VR11) zum DAC wurden die herstellerspezifischen Tabellen (zu finden auf Seite 26 im Handbuch „Für den Betreiber/für den Fachhandwerker Bedienungs- und Installationsanleitung auroMATIC 560“) herangezogen. Die gemessenen und gemittelten Werte wurden mittels einer Widerstandsdekade nachgebildet und im Abschluss in die Dateien „vrs10.txt“ und „vrs11.txt“ überführt.

Die Berechnung der jeweiligen Temperaturen erfolgt dann im Programm mittels einer Linearen Interpolation (siehe Quellcode tempsensors.ino).

Die generierten Dateien (egal ob selbst oder die aus dem mitgelieferten Archiv) müssen auf die SD-Karte in das Verzeichnis „/sensordata“ kopiert werden. Auf die Namenskonformität ist hierbei zu achten.

Gpio Pin	Name
34	lut34.txt
36	lut36.txt
39	lut39.txt



Installation

Es besteht die Möglichkeit den Quellcode selbst zu kompilieren oder die bereitgestellten „.bin“ Dateien aus dem Github-Repository (siehe Link github) zu verwenden.

Unabhängig ob Kompilierung oder „.bin“ Nutzung, es muss eine SD-Karte (FAT32) mit folgender Ordnerstruktur vorbereitet werden.

Ordner	Verwendung
/sensordata	Speicherort der Lookup-Dateien
/log	Speicherort für die Logfiles
/webserver	Speicherort der html, js und css Dateien

Im Anschluss sollten die Inhalte aus den gleichnamigen Ordnern des ZIP-Archivs 1:1 in den jeweiligen Ordner überführt werden. Für den Fall das eigenen Lookup-Dateien generiert wurden, sind Diese natürlich zu berücksichtigen.

Kompilierung

Für die Entwicklung wurde Arduino in der Version 2.3.2 verwendet. Die Dateien (*.ino, *.h) können daher 1:1 in den Ordner eines bestehenden Arduino Projekts kopiert werden. Bitte auf Namensgleichheit („VRS-Replace“) achten.

- Neuen Sketch erstellen
- Sketch unter dem Name „VRS-Replace“ speichern
- Sketch Ordner öffnen
- *.ino & *.h Files in den Ordner aus dem ZIP „/src“ kopieren

Folgende externe Bibliotheken wurden zusätzlich über den „LIBRARY MANAGER“ in das Projekt aufgenommen.

- ArduinoJson
- PsychicHttp
- U8g2lib

Die Kompilierung erfolgt mit dem ausgewählten Board „ESP32-WROOM-DA Module“ und mit dem Partition Scheme „Minimal SPIFFS (1.9MB APP with OTA/190KB SPIFFS)“. Weitere Einstellungen sollten hierfür nicht relevant sein, können aber dem nachfolgenden Screenshot entnommen werden.



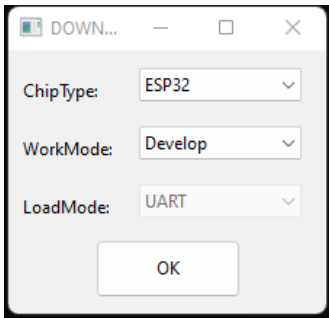


Ein Wechsel der Entwicklungsumgebung inkl. dem Wechsel auf C++ ist in der Planung.



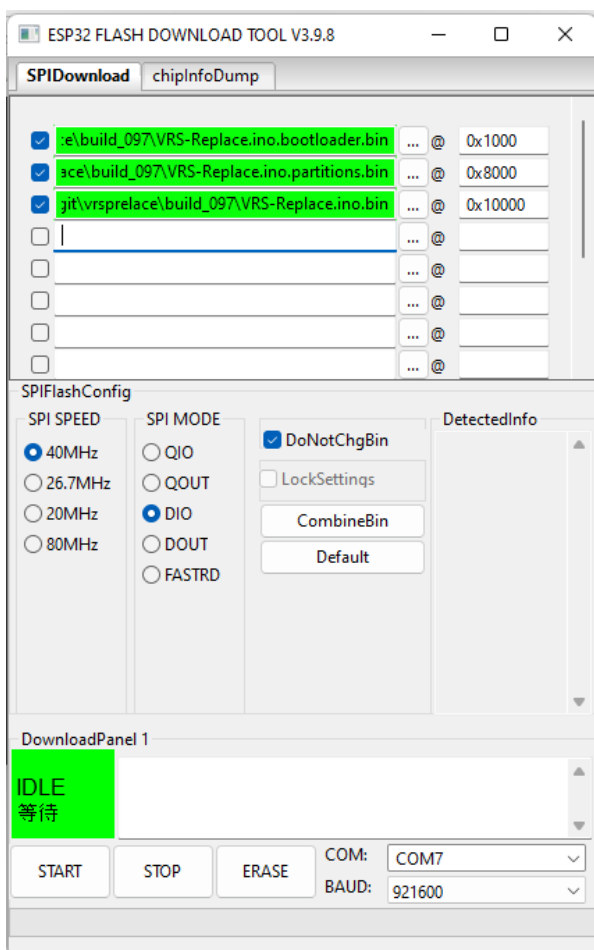
*.bin Files

Für die Installation der *.bin Dateien, kann das ESP32 FLASH DOWNLOAD TOOL (hier verwendet Version v3.9.8) verwendet werden. Bitte auf die richtigen Parameter achten.



Die *.bin Dateien müssen entsprechen der folgende Tabelle auch an die jeweiligen Bereiche übertragen werden:

Datei	Address
VRS-Replace.ino.bootloader.bin	0x1000
VRS-Replace.ino.partitions.bin	0x8000
VRS-Replace.ino.bin	0x10000



Der ESP sollte danach neu starten. Für den Fall das keine Konfiguration vorhanden ist, startet das System als AP (siehe Konfiguration / Administration)



Betrieb

Der mögliche Betrieb kann in folgende Bereiche gegliedert werden.

Konfiguration / Administration

Ohne irgendwelche Einstellungen, startet das System als Access-Point unter der SSID „VRS-Replace“. Nach Anmeldung mit dem Key „12345678“ kann die Konfigurationsseite durch Eingabe der IP „192.168.4.1“ aufgerufen werden.

Config

SSID: WLAN1 Passwort:

NTP Server: de.pool.ntp.org Zeitzone: Europe/Berlin

Sprache: de

Verbindungsversuche: 10 Wartezeit(ms): 60000

☐ MQTT aktiv?

MQTT-Server: mqtt://192.168.1.91 MQTT-Send intervall(ms): 5000

Small log write(ms): 60000 Large log write(ms): 10000

Sensors Refresh(ms): 10000 Maximale Temperatur(°C): 120

Logpath: /log

Webserverpath: /webserver

Lutpath: /sensordata

☒ CORS zulassen ☐ HTML von SD

Speichern Neustart Zurück

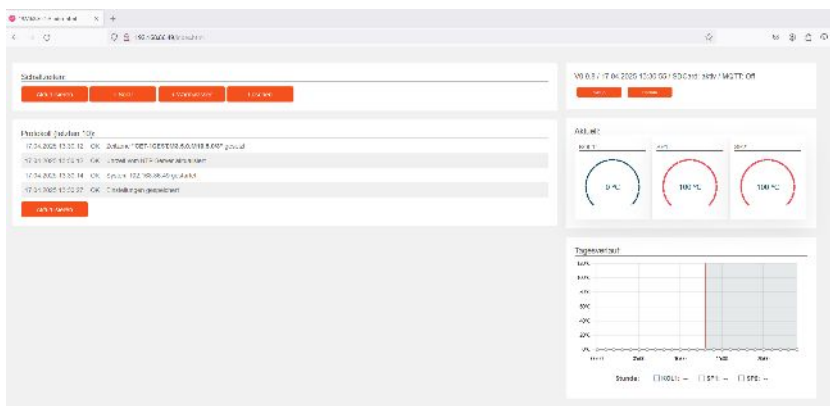
Parameter	erfordert Neustart	Beschreibung
SSID:	x	SSID des WiFi-Netzwerks
Passwort:	x	Password für die Anmeldung am WiFi-Netzwerk
NTP Server:		IP oder Domain des NTP-Servers
Zeitzone:		Auswahl unterschiedlicher europäischer Zeitzonen. Für Deutschland „Europe/Berlin“
Sprache:		Derzeit werden „de“ = Deutsch und „en“ = Englisch angeboten
Verbindungsversuche:		Anzahl der Verbindungsversuche bis das Projekt sich wieder als AP meldet.
Wartezeit(ms):		Wartezeit zwischen den einzelnen Verbindungsversuchen.
MQTT aktiv?		MQTT aktiv
MQTT-Server:		URL des MQTT-Servers. Der Term „mqtt://“ muss am Anfange der URL stehen.
MQTT-Send intervall(ms):		Intervall für den Versand von MQTT-Nachrichten.
Small log write(ms):		In welchen Zeitabständen soll das „Small“-Log geschrieben werden.



Large log write(ms):		In welchen Zeitabständen soll das „Large“-Log geschrieben werden.
Sensors Refresh(ms):		Aktualisierungszeit der Temperatursensoren
Maximale Temperatur(°C):		Maximal zulässige Temperatur des Kollektorfühler.
Logpath:		Ablageort auf der SD Karte für die Logfiles.
Webserverpath:		Ablageort auf der SD Karte für die HTML/JS/CSS Dateien.
Lutpath:		Ablageort für die LUT-Tabellen. (siehe ADC Kalibrierung / Temperatursensoren)
CORS zulassen?		Soll der HTTP-Server CORS-Anfragen zulassen.
HTML von SD?		Sollen HTML/JS/CSS-Dateien von der SD-Karte geliefert werden.

Nachdem die Konfiguration vorgenommen wurde, kann mittels „Speichern“ und „Neustart“ das System neu gestartet werden. Sollte bei der Konfiguration ein Fehler unterlaufen sein, so startet das System erneut als Access-Point nach (Verbindungsversuche * Wartezeit(ms)).

Mit korrekten Konfigurationswerten startet das System in einer ähnlichen Darstellung.




Obige Darstellung wurde der Entwicklungsumgebung entnommen und hier sind keine Temperaturfühler angeschlossen.



Timer für Solarpumpe

Ein Druck auf „+ Solar“ öffnet den Dialog für die Erstellung eines neuen Timer für die Solarpumpe.



Solarpumpe

Name:

Laufzeit (s):

Wartezeit (s):

Einschalttemperatur (°C):

Ausschalttemperatur (°C):

Temperaturdifferenz (°C):

Temperatur Sensor: SP1 ▼

☐ Aktiv

Speichern

Zurück

Parameter:	Beschreibung:
Name:	Name des Timer
Laufzeit (s):	Laufzeit der Pumpe in Sekunden (Timer blinkt rot)
Wartezeit (s):	Wartezeit der Pumpe in Sekunden (Timer blinkt gelb)
Einschalttemperatur (°C):	Ab welcher Temperatur (Kol1) soll die Pumpe aktiviert werden
Ausschalttemperatur (°C):	Zieltemperatur, abhängig von „Temperatur Sensor“
Temperaturdifferenz (°C):	Temperaturdifferenz Kol1 zu „Temperatur Sensor“ zum Starten der Pumpe. Falls die Temperatur kleiner als die Ausschalttemperatur ist und der Timer bereits gestartet wurde, schaltet der Timer in den Wartemodus.
Temperatur Sensor:	Relevanter Temperatursensor Sp1, Sp2 oder Mittelwert
Aktiv	Timer aktiv oder nicht

Timer für Warmwasser

Ein Druck auf „+ Warmwasser“ öffnet den Dialog für die Erstellung eines neuen Warmwasser-Timer.

Parameter	relevantes „Zeiten“ Programm	Beschreibung
Name:	X	Name des Timer
Zeiten:	X	Auswahl der jeweiligen Zeitsteuerung (Täglich/Wochentags/Manuell)
MO, DI, MI, DO, FR, SA, SO	Wochentags	An welchem Wochentag soll der Timer gestartet werden.
Einschaltzeit:	Täglich/Wochentags	Wann soll der Timer gestartet werden HH24:mm
Ausschaltzeit:	Täglich/Wochentags	Wann soll der Timer gestoppt werden HH24:mm
Ausschalttemperatur (°C)	Täglich/Wochentags	Bei welcher Temperatur soll der Timer gestoppt bzw. beendet werden.
Hysterese (°C)	Täglich/Wochentags	Hysterese
Temperatur Sensor:	X	Relevanter Temperatursensor Sp1, Sp2 oder Mittelwert
Aktiv	X	Timer aktiv oder nicht



Block „Schaltzeiten“



Der Block „Schaltzeiten“ zeigt sämtliche definierten Programme an. Bei inaktiven Programmen wird der Name mit einem roten Strich durchgestrichen dargestellt. Aktive Timer blinken je nach Status „gelb“ oder „rot“.

Timer vom Typ „Manuell“ könnten durch einen Druck auf das entsp. Symbol „gestartet“ oder „gestoppt“ werden.

Start Timer	Stop Timer

* In der Chart wurde der Timer auf eine Zieltemperatur von 130°C gesetzt. Das ist natürlich absoluter Quatsch und wird auch abgefangen.

Button(s)	Beschreibung
Aktualisieren	Aktualisiert die Timerliste
+ Solar	Öffnet den Dialog für das Hinzufügen eines Solar Timer's
+ Warmwasser	Öffnet den Dialog für das Hinzufügen eines Warmwasser Timer's
Löschen	Markierte Timer werden nach Druck auf diesen Button gelöscht. Es findet keine Warnung statt.

Ein Doppelklick bzw. ein „Longtouch“ auf einen Timer öffnet den Dialog mit den konfigurierten Werten.



Block „Protokoll (letzten 10)“

Protokoll (letzten 10):		
17.04.2025 13:59:01	OK	Heizungstimer "Wochenende 10:00" hinzugefügt
17.04.2025 13:59:34	OK	Heizungstimer "Manuell 38" hinzugefügt
17.04.2025 14:59:49	OK	Timer für Solarsteuerung "SolarPumpe" hinzugefügt
17.04.2025 15:00:58	OK	Heizungstimer "Täglich 08:00" geändert
17.04.2025 15:07:13	OK	Einstellungen gespeichert
17.04.2025 15:07:32	OK	Heizungstimer "Manuell 38" geändert
17.04.2025 15:08:46	OK	Heizungstimer "Manuell 38" geändert
17.04.2025 15:08:50	OK	Timer "Manuell 38" gestartet
17.04.2025 15:08:54	OK	Timer "Manuell 38" gestoppt
17.04.2025 15:09:45	OK	Timer "Manuell 38" gestartet
Aktualisieren		

Zeigt die letzten 10 Protokolleinträge.

Button(s)	Beschreibung
Aktualisieren	Aktualisiert die letzten 10 Einträge

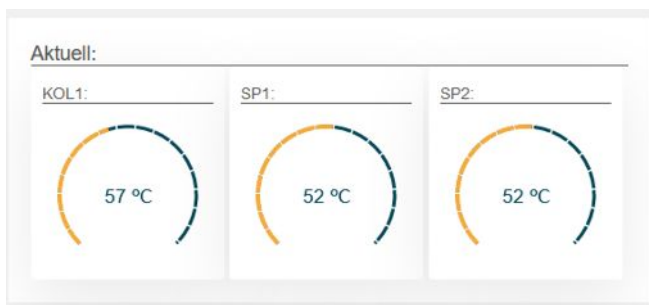
Block „Status“

V0.9.8 / 17.04.2025 15:32:12 / SDCard: aktiv / MQTT: Off	
Setup	Update

Zeigt den aktuellen Systemstatus. Die Version, Datum mit Uhrzeit, ob die SD-Karte aktiv ist und ob die MQTT-Übertragung aktiviert wurde.

Button(s)	Beschreibung
Setup	Öffnet das Konfigurationsmenü
Update	Öffnet den Dialog für das Einspielen einer neuen Version

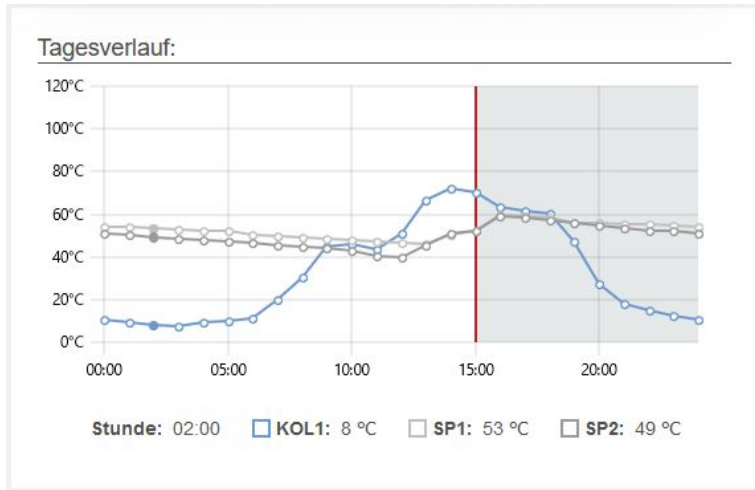
Block „Aktuell“



Zeigt die aktuell gemessene Temperatur des jeweiligen Sensors.



Block „Tagesverlauf“



Zeigt die aktuelle Temperaturkurve verteilt über den Tag. Der rote Balken markiert die aktuelle Stunde und der graue Bereich zeigt den Temperaturverlauf des gestrigen Tags.



REST

Die Kommunikation zwischen Front- und Backend erfolgt mittels den aufgelisteten REST Aufrufen. Die verwendete IP (192.168.80.16) bei einzelnen Beispielen, bezieht sich auf meine Umgebung. Die IP ist daher an die eigene Umgebung anzupassen.

/status

Bereitstellung aktueller Systeminformationen

Input:	Type:	Beschreibung
NA		
Output:		
timestamp	string	Zeitstempel im Format
sdmount	bool	Ist die SD-Karte verfügbar(mount)
version	string	Version der Software
sd	bool	Ist das SD-Interface verfügbar
nbtimers	int	Anzahl der Timer
heap	int	Verfügbarer Heap-Memory (bytes)
mqttactive	bool	Wurde MQTT aktiviert

http://192.168.80.16/status

```
{"timestamp":"18.04.2025  
12:10:26","sdmount":true,"version":"0.9.8","sd":true,"mqttactive":false,"nbtimers":3,"nbtimersactive":0,"heap":91416}
```



/log – Hier Rechtschreibung

Liefert die letzten 10 Protokolleinträge

Input:	Type:	Beschreibung
NA		
Output:	Array[10]	
id	int	Fortlaufender Nummer
priocode	int	Priorität des Eintrags (Veraltet)
date	int	Datum im Format yyyyMMdd
time	int	Uhrzeit im Format HH24mmss ohne führende Null
code	int	Statuscode (siehe Statuscodes)
message	string	Zusammengesetzte Nachricht. Beispiel: code=201 / message=5!morgens Übersetzt: Timer mit der id=5 und dem Namen=morgens wurde gestoppt

http://192.168.80.16/log

```
[{"id":5523,"priocode":10,"date":20250418,"time":10629,"code":182,"message":"CET-1CEST,M3.5.0,M10.5.0/3"}, {"id":5524,"priocode":10,"date":20250418,"time":10629,"code":180,"message":""}, {"id":5525,"priocode":10,"date":20250418,"time":10726,"code":125,"message":"192.168.80.16"}, {"id":5526,"priocode":10,"date":20250418,"time":10730,"code":126,"message":"192.168.80.16"}, {"id":5527,"priocode":10,"date":20250418,"time":10730,"code":182,"message":"CET-1CEST,M3.5.0,M10.5.0/3"}, {"id":5528,"priocode":10,"date":20250418,"time":10730,"code":180,"message":""}, {"id":5529,"priocode":10,"date":20250418,"time":32045,"code":125,"message":"192.168.80.16"}, {"id":5530,"priocode":10,"date":20250418,"time":32150,"code":126,"message":"192.168.80.16"}, {"id":5531,"priocode":10,"date":20250418,"time":32150,"code":182,"message":"CET-1CEST,M3.5.0,M10.5.0/3"}, {"id":5532,"priocode":10,"date":20250418,"time":32150,"code":180,"message":""}]
```



/downloadtemperaturelog

Ermöglicht den Download von protokollierten Temperaturwerten. Der Aufbau der Dateien existiert in zwei unterschiedlichen Formaten „small“ und „full“. Das Schreiben und somit die Größe des Temperaturlogs „full“ wird durch den Parameter „Large log write(ms):“ (siehe Konfiguration / Administration) beeinflusst.

Input:	Type:	Beschreibung
logfile	int	Datum des gewünschten Logfiles. Format: yyyyMMdd
type	string	„small“ oder „full“
Output:	Startet den Download-Dialog	
NA		

CSV Dokumentformat small

Spalte:	Type:	Beschreibung
1	int	Temperatursensor 0=Kol1, 1=Sp1, 2=Sp2
2-25	double	Maximal gemessene Temperatur zur vollen Stunde
26-49	double	Minimal gemessene Temperatur zur vollen Stunde

CSV Dokumentformat full

Spalte:	Type:	Beschreibung
1	int	Datum im Format yyyyMMdd
2	string	Uhrzeit im Format HH24mmss mit führender Null
3	int	Id eines aktiven Warmwassertimer (-1 falls kein Programm gestartet)
4	int	Id eines aktiven Solartimer (-1 falls kein Programm gestartet)
5	int	Id eines aktiven Zirkulationspumpentimer (Derzeit keine Verwendung, daher immer -1)
6	string	Name des Temperatursensors (Kol1, Sp1, Sp2)
7	int	Id des Temperatursensors
8	double	Aktuell gemessene Temperatur
9	double	Maximal gemessene Temperatur (pro Tag)
10	double	Minimal gemessene Temperatur (pro Tag)



/temperature

Liefert die aktuell gemessenen Temperaturwerte innerhalb eines JSON-Arrays.

Input:	Type:	Beschreibung
NA		
Output:	Array[3]	
bez	int	Name des Sensors
deviceid	int	Id des Sensors Kol1=0 / Sp1 = 1 / Sp2 = 2
sensortype	int	Sensortyp VR10 = 1 / VR11=2
Ta	double	Aktuelle Temperatur
TaAvg	double	Durchschnittliche Temperatur (Anpassbar im Code sensorAvgTemperatureCalcSize)
TaMin	double	Minimum Temperatur (pro Stunde)
TaMax	double	Maximal Temperatur (pro Stunde)
TaMinH	double Array[24]	Array mit den Minimum Temperaturen pro Stunde
TaMaxH	double Array[24]	Array mit den Maximal Temperaturen pro Stunde
TaMinY	double Array[24]	Array mit den Minimum Temperaturen pro Stunde / Vortag
TaMaxY	double Array[24]	Array mit den Maximal Temperaturen pro Stunde / Vortag

http://192.168.80.16/temperature

```
[{"bez": "Kol1", "deviceid": 1, "sensortype": 2, "Ta": 25.53, "TaAvg": 25.59, "TaMin": 25.48, "TaMax": 26.08, "TaMinH": [8.64, 8.48, 8.34, 8.22, 7.97, 7.83, 7.83, 9.58, 12.49, 16.62, 28.76, 26.26, 31.47, 26.01, 25.48, 55.22, 63.67, 56.91, 21.28, 13.23, 10.65, 9.58, 9.07, 8.76], "TaMaxH": [9.11, 8.93, 8.88, 8.81, 8.69, 8.41, 9.72, 13.81, ..., "TaMaxHY": [51.27, 50.19, 49.36, 48.51, 47.73, 47.06, 46.38, 45.31, 44.93, 44.14, 43.15, 40.32, 40.06, 45.31, 50.75, 52.09, 55.11, 56.87, 56.79, 55.49, 54.18, 53.02, 52.31, 51.87]]}]
```



/config

Liste mit den vergebenen Konfigurationswerten

Input:	Type:	Beschreibung
NA		
Output:		
ssid	string	SSID des genutzten Wi-Fi Netzwerks
password	string	Immer ***
ntpserver	string	IP oder Domain des NTP-Servers
lutpath	string	Order auf der SD-Karte für die vorkonfigurierten Sensordaten.
logpath	string	Order auf der SD-Karte für die Ablage der Logfiles
webserverpath	string	Order auf der SD-Karte für die Bereitstellung von html, js, css Dateien.
temperaturesrt	int	Zeit in ms für das messen von Temperaturwerten
temperaturelwt	int	Zeit in ms für das Schreiben von Temperaturwerten (small)
temperatureswt	int	Zeit in ms für das Schreiben von Temperaturwerten (full)
wificonnectionretries	int	Anzahl der Wi-Fi Verbindungsversuche
wifireconnectinterval	int	Zeit zwischen den einzelnen Verbindungsversuchen
maxtemperature	int	Maximale Kollektortemperatur
version	string	Version der installierten Software
allowcors	bool	CORS Zugriffe zulässig?
deliverfromsd	bool	Webdaten werden von der SD-Karte bereitgestellt?
mqttactive	bool	MQTT aktiv?
mqttserver	string	URL des MQTT-Server
mqttsendinterval	int	Intervall (ms) für das Senden von MQTT-Nachrichten
language	string	Sprache (de, en)
timezone	string	Eingestellte Zeitzone

http://192.168.80.16/config

```
{"ssid":"WLAN1","password":"*****","ntpserver":"de.pool.ntp.org","lutpath":"/sensordata","logpath":"/log","webserverpath":"/webserver","temperaturesrt":10000,"temperaturelwt":10000,"temperatureswt":60000,"wificonnectionretries":10,"wifireconnectinterval":60000,"maxtemperature":120,"version":"0.9.8","allowcors":true,"deliverfromsd":false,"mqttactive":false,"mqttserver":"mqtt://192.168.1.91","mqttsendinterval":5000,"language":"de","timezone":"+4;CET-1CEST,M3.5.0,M10.5.0/3"}
```



/system

Ermöglicht Systemänderungen durch hinzufügen einer der folgenden Parameter an die URL.

Input:	Type:	Beschreibung
reboot	string	Startet das System neu
ntp	string	Führt einen Abgleich der Uhrzeit über den eingetragenen NTP-Server durch.
eraseconfig	string	Löscht die komplette Konfiguration und startet als AP neu.
Output:	JSON Status	
timestamp	int	DatumUhrzeit im Format dd.MM.yyyy HH24:mm:ss
id	int	Id des Timer. Falls der Aufruf sich nicht auf einen Timer bezieht, ist der Wert immer -1.
status	int	1=Ok / 2=Error
code	int	Statuscode (siehe Statuscodes)

```
http://192.168.80.16/system?ntp
```

```
{"timestamp":"18.04.2025 14:40:07","id":-1,"status":1,"code":112}
```

Entsprechender Eintrag in dem Block „Protokoll (letzten 10)“

```
18.04.2025 14:40:06 OK Zeitzone "CET-1CEST,M3.5.0,M10.5.0/3" gesetzt
```

```
18.04.2025 14:40:07 OK Uhrzeit vom NTP-Server aktualisiert
```

```
18.04.2025 14:40:07 OK NTP Uhrzeit aktualisiert
```

Alarmeinstellungen



/changeconfig

Ermöglicht eine Änderung der eingestellten Konfigurationsparameter. Eine Änderung von SSID und/oder vom Password, erfordert einen Neustart.

Input:	Type:	Beschreibung
config	JSON Dokument	
JSON Dokument:		
ssid	string	SSID des genutzten Wi-Fi Netzwerks
password	string	Immer ***
ntpserver	string	IP oder Domain des NTP-Servers
temperaturesrt	int	Zeit in ms für das messen von Temperaturwerten
temperaturelwt	int	Zeit in ms für das Schreiben von Temperaturwerten (small)
temperatureswt	int	Zeit in ms für das Schreiben von Temperaturwerten (full)
wificonnectionretries	int	Anzahl der Wi-Fi Verbindungsversuche
wifireconnectinterval	int	Zeit zwischen den Verbindungsversuchen
maxtemperature	int	Maximale Kollektortemperatur
Output:		
timestamp	int	DatumUhrzeit im Format dd.MM.yyyy HH24:mm:ss
id	int	Id des Timer. Falls der Aufruf sich nicht auf einen Timer bezieht, ist der Wert immer -1.
status	int	1=Ok / 2=Error
code	int	Statuscode (siehe Statuscodes)



/timer

Liefert die konfigurierten Werte eines Timer zurück. Wird keine gültige TimerId beim Aufruf als Id übergeben, so wird eine entsp. JSON Status Meldung zurückgeliefert.

Input:	Type:	Timerprogramm(e)	Relais	Beschreibung
id	int	Alle	Alle	Id des Timer
Output:	JSON Dokument			
pos	string	Alle	Alle	
id	int	Alle	Alle	Eindeutige ID des Timer
bez	string	Alle	Alle	Bezeichnung des Timer
gpiopin	int	Alle	Alle	Verwendeter ESP-Pin
relais	int	Alle		Zugeordneter Relais-Type: 1=Warmwasser, 2=Solar
time_on	int	1,2	1	Startzeit Format: H24mmss
time_off	int	1,2	1	Endzeit Format: H24mmss
weekdays	int Array[7]	2	1	Wochentage, an denen der Timer gestartet werden soll
running_period	int		2	Wie lange soll der Timer laufen (ms)
waiting_period	int		2	Wie lange soll der Timer warten (ms)
timer_program	int		1	Typ des Timer 1=Täglich, 2=Wochentags, 3=Manuell
temperature_on	int		2	Kol1 Temperatur für das Starten des Timer
temperature_off	int	1,2,3	1,2	Zieltemperatur
tempsensor	int	Alle	Alle	Relevanter Temperatursensor für das Erreichen der Zieltemperatur Keiner=0, Sp1=1, Sp2=2, Avg_Sp1_Sp2=3, Kol1=4
temperature_difference	int		2	Temperaturdifferenz Kol1 zu „tempsensor“ für das Starten des Timer
hysteresis	int	1,2	1	Hysteresis
active	bool	Alle	Alle	Timer aktiv?
state	bool	Alle	Alle	Timer gestartet?
timerstatetype	int	Alle	Alle	Derzeitiger Timer-Status 1=On, 2=Off, 3=Wait
runnings	int	Alle	Alle	Anzahl wie oft der Timer bisher gestartet wurde.
waitings	int	Alle	Alle	Anzahl wie oft der Timer bisher gewartet hat.
laststartdate	int	Alle	Alle	Datum des letzten Starts. Format: yyyyMMdd
laststarttime	int	Alle	Alle	Uhrzeit des letzten Starts. Format: H24mmss
publishhomekit	bool	3	1	Nicht mehr relevant, da keine Unterstützung für Homekit mehr geplant ist. → MQTT
avgtempincreas epermin	int	1,2,3	1,2	Durchschnittlicher Temperaturanstieg pro Minute (Future)



http://192.168.80.16/timer?id=45 (Ungültig)

```
{"timestamp":"18.04.2025 15:33:37","id":-1,"status":2,"code":312}
```

http://192.168.80.16/timer?id=1

```
{"pos":"0","id":1,"bez":"Manuell 38","gpiopin":16,"relais":1,"time_on":-1,"time_off":-1,"weekdays":  
[0,0,0,0,0,0],"running_period":-1,"waiting_period":-1,"timer_program":3,"temperature_on":-  
1,"temperature_off":38,"tempsensor":1,"temperature_difference":-1,"hysteresis":-  
1,"active":true,"state":false,"timerstatetype":2,"runnings":44,"waitings":0,"laststartdate":20250316,"laststartti  
me":84931,"publishhomekit":false,"avgtempincreasepermin":-1}
```



/timerchange

Ermöglicht das Hinzufügen eines neuen oder das Ändern eines existierenden Timer. Hierzu muss folgendes JSON-Dokument gesendet werden.

Input:	Type:	Timerprogramm(e)	Relais	Beschreibung
timer	JSON Dokument	Alle	Alle	
Input JSON Dokument:				
id	int	Alle	Alle	Id des Timer oder -1 für einen neuen Timer. Im zurück gegebenen Dokument kann ggf. die neue TimerId entnommen werden.
bez	string	Alle	Alle	Bezeichnung des Timer
gpiopin	int	Alle	Alle	Verwendeter ESP-Pin
relais	int	Alle		Zugeordneter Relais-Type: 1=Warmwasser, 2=Solar
time_on	int	1,2	1	Startzeit Format: H24mmss
time_off	int	1,2	1	Endzeit Format: H24mmss
weekdays	int Array[7]	2	1	Wochentage, an denen der Timer gestartet werden soll
running_period	int		2	Wie lange soll der Timer laufen (ms)
waiting_period	int		2	Wie lange soll der Timer warten (ms)
timer_program	int		1	Typ des Timer 1=Täglich, 2=Wochentags, 3=Manuell
temperature_on	int		2	Kol1 Temperatur für das Starten des Timer
temperature_off	int	1,2,3	1,2	Zieltemperatur
tempsensor	int	Alle	Alle	Relevanter Temperatursensor für das Erreichen der Zieltemperatur Sp1=1, Sp2=2, Avg_Sp1_Sp2=3
temperature_difference	int		2	Temperaturdifferenz Kol1 zu „tempsensor“ für das Starten des Timer
hysteresis	int	1,2	1	Hysterese
active	bool	Alle	Alle	Timer aktiv?
Output:	JSON Status		1,2	Durchschnittlicher Temperaturanstieg pro Minute (Future)
timestamp	int			DatumUhrzeit im Format dd.MM.yyyy HH24:mm:ss
id	int	.		Id des Timer. Falls der Aufruf sich nicht auf einen Timer bezieht, ist der Wert immer -1.
status	int			1=Ok / 2=Error
code	int			Statuscode (siehe Statuscodes)



/timerdelete

Der Aufruf löscht einen oder mehrere Timer. Die zu löschenden Timer werden mittels einem Arrays unter der jeweiligen TimerId übergeben. Gestartete Timer werden im Vorfeld beendet.

Input:	Type:	Beschreibung
timers	JSON Array	Array mit den Id's der zu löschenden Timer.
Output:	JSON Status	
timestamp	int	DatumUhrzeit im Format dd.MM.yyyy HH24:mm:ss
id	int	Id des Timer. Falls der Aufruf sich nicht auf einen Timer bezieht, ist der Wert immer -1.
status	int	1=Ok / 2=Error
code	int	Statuscode (siehe Statuscodes)

```
http://192.168.80.16/timerdelete?timers={'id':[1,2]}
{"timestamp":"18.04.2025 16:36:34","id":-1,"status":1,"code":311}
```

18.04.2025 16:36:34 OK Timer "**Täglich 08:00**" wurde gelöscht

18.04.2025 16:36:34 OK Timer "**Wochenende 10:00**" wurde gelöscht

Aktualisieren

/timerdeleteall

Löscht alle existierenden Timer. Gestartete Timer werden im Vorfeld beendet.

Input:	Type:	Beschreibung
NA		
Output:	JSON Status	
timestamp	int	DatumUhrzeit im Format dd.MM.yyyy HH24:mm:ss
id	int	Id des Timer. Falls der Aufruf sich nicht auf einen Timer bezieht, ist der Wert immer -1.
status	int	1=Ok / 2=Error
code	int	Statuscode (siehe Statuscodes)



/timerlist

Liefert eine Liste als JSON Array mit allen bestehenden Timer.

Input:	Type:	Timerprogramm(e)	Relais	Beschreibung
NA				
Output:	JSON Array			
pos	string	Alle	Alle	
id	int	Alle	Alle	Eindeutige ID des Timer
bez	string	Alle	Alle	Bezeichnung des Timer
gpiopin	int	Alle	Alle	Verwendeter ESP-Pin
relais	int	Alle		Zugeordneter Relais-Type: 1=Warmwasser, 2=Solar
time_on	int	1,2	1	Startzeit Format: H24mmss
time_off	int	1,2	1	Endzeit Format: H24mmss
weekdays	int Array[7]	2	1	Wochentage, an denen der Timer gestartet werden soll
running_period	int		2	Wie lange soll der Timer laufen (ms)
waiting_period	int		2	Wie lange soll der Timer warten (ms)
timer_program	int		1	Typ des Timer 1=Täglich, 2=Wochentags, 3=Manuell
temperature_on	int		2	Kol1 Temperatur für das Starten des Timer
temperature_off	int	1,2,3	1,2	Zieltemperatur
tempsensor	int	Alle	Alle	Relevanter Temperatursensor für das Erreichen der Zieltemperatur Keiner=0, Sp1=1, Sp2=2, Avg_Sp1_Sp2=3, Kol1=4
temperature_difference	int		2	Temperaturdifferenz Kol1 zu „tempsensor“ für das Starten des Timer
hysteresis	int	1,2	1	Hysteresis
active	bool	Alle	Alle	Timer aktiv?
state	bool	Alle	Alle	Timer gestartet?
timerstatetype	int	Alle	Alle	Derzeitiger Timer-Status 1=On, 2=Off, 3=Wait
runnings	int	Alle	Alle	Anzahl wie oft der Timer bisher gestartet wurde.
waitings	int	Alle	Alle	Anzahl wie oft der Timer bisher gewartet hat.
laststartdate	int	Alle	Alle	Datum des letzten Starts. Format: yyyyMMdd
laststarttime	int	Alle	Alle	Uhrzeit des letzten Starts. Format: H24mmss
publishhomekit	bool	3	1	Nicht mehr relevant, da keine Unterstützung für Homekit mehr geplant ist. → MQTT
avgtempincreasepermin	int	1,2,3	1,2	Durchschnittlicher Temperaturanstieg pro Minute (Future)



/timerstart

Startet einen Timer vom Type „Manuell“ durch Übergabe der entspr. TimerId.

Input:	Type:	Beschreibung
id	int	Id des Timer der gestartet werden soll.
Output:	JSON Status	
timestamp	int	DatumUhrzeit im Format dd.MM.yyyy HH24:mm:ss
id	int	Id des Timer. Falls der Aufruf sich nicht auf einen Timer bezieht, ist der Wert immer -1.
status	int	1=Ok / 2=Error
code	int	Statuscode (siehe Statuscodes)

`http://192.168.80.16/timerstart?id=3`

```
{"timestamp":"18.04.2025 16:47:05","id":3,"status":1,"code":305,"message":"3|Manuell 38"}
```

/timerstop

Stoppt den Timer durch Übergabe der entspr. TimerId. Nur relevant für Timer vom Type „Manuell“.

Input:	Type:	Beschreibung
id	int	Id des Timer der gestoppt werden soll.
Output:	JSON Status	
timestamp	int	DatumUhrzeit im Format dd.MM.yyyy HH24:mm:ss
id	int	Id des Timer. Falls der Aufruf sich nicht auf einen Timer bezieht, ist der Wert immer -1.
status	int	1=Ok / 2=Error
code	int	Statuscode (siehe Statuscodes)

`http://192.168.80.16/timerstop?id=3`

```
{"timestamp":"18.04.2025 16:47:28","id":3,"status":1,"code":307,"message":"3|Manuell 38"}
```

18.04.2025 16:47:05 OK Timer "**Manuell 38**" gestartet

18.04.2025 16:47:28 OK Timer "**Manuell 38**" gestoppt

Aktualisieren



/testrelay

Ermöglicht das Testen der angeschlossenen Relais, unabhängig von irgendwelchen Programmen. Timer haben Vorrang und können daher den Status (high, low) überschreiben.

Input:	Type:	Beschreibung
gpio	int	Relevanter GPIO Pin für Low/High Aktivierung. Derzeit GPIO 16 = Warmwasser, GPIO 17 = Solar
onoff	int	On=1 / Off=2
Output:	JSON Status	
timestamp	int	DatumUhrzeit im Format dd.MM.yyyy HH24:mm:ss
id	int	Id des Timer. Falls der Aufruf sich nicht auf einen Timer bezieht, ist der Wert immer -1.
status	int	1=Ok / 2=Error
code	int	Statuscode (siehe Statuscodes)



MQTT

Das System ist in der Lage mittels MQTT-Messages über einen MQTT-Brokers und anhand von verschiedenen Topics zu kommunizieren.

```
▼ 192.168.1.91
▼ vrsreplace = online
status = {"timestamp":"31.03.2025 13:45:11","sdmount":true,"version":"0.9.8","sd":true,"mqttactive":true,"nbtimers":3,"nbtimersactive":0,"heap":76768}
▼ sensor
  Kol1 = 27.27
  Sp1 = 34.49
  Sp2 = 29.14
▼ timer
  1 = {"bez":"","Manuell":38,"relaistype":1,"timerstatetype":2,"active":true}
  2 = {"bez":"","Solar":"","relaistype":2,"timerstatetype":2,"active":true}
  3 = {"bez":"","Wochentags":"","relaistype":1,"timerstatetype":2,"active":true}
message = {"timestamp":"31.03.2025 12:35:42","id":2,"status":1,"code":200,"message":"2(Solar)"}
```

Readable Topics

Topic:	Type:	Beschreibung:
/vrsreplace	string	Generelle Statusinformationen. Diese Topic beinhaltet auch den LWT (online/offline).
/vrsreplace/status	json	Statusinformationen analog wie beim REST-Server „/status“
/vrsreplace/sensor/[Sensor]	double	Aktuelle gemessene Temperatur für den jeweiligen Sensor „[Kol1, Sp1, Sp2]“
/vrsreplace/timer/[Timer ID]	json	Status der verfügbaren Timer
/vrsreplace/message	string	Zuletzt publizierter Logeintrag

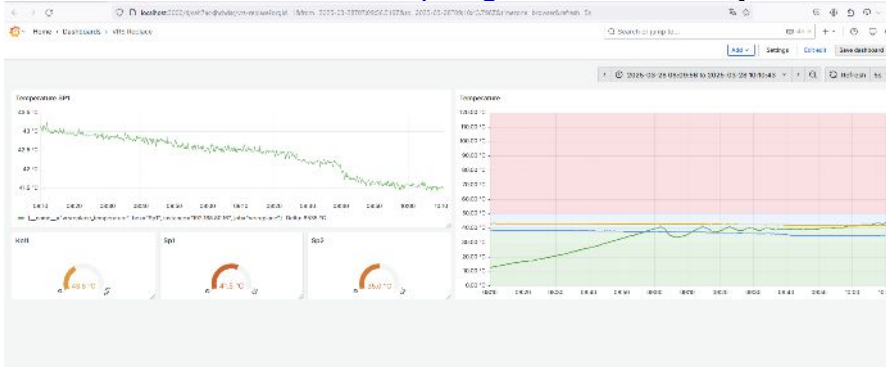
Writeable Topic

Topic:	Type:	Beschreibung:
/vrsreplace/timer/[Timer ID]/cmd	string	Alle Timer können aktiviert oder deaktiviert werden. Hierzu wird unter der jeweiligen „Timer ID“ der Befehl „activate“ oder „deactivate“ gesendet. Manuelle Timer (Type 3) können zusätzlich gestartet „cmd=start“ oder gestoppt „cmd=stop“ werden.



Prometheus metrics

Einzelne Werte können auch über Prometheus <https://prometheus.io> protokolliert und im Anschluss mittels Grafana <https://grafana.com/> analysiert werden.



Die Einbindung in Prometheus erfolgt durch einen Eintrag in der Datei „prometheus.yml“ innerhalb der Sektion „scrape_configs“. Bitte unbedingt auf das notwendige Format (Anzahl Blanks) achten!

```
- job_name: 'vrsreplace'
  scrape_interval: 10s # >= Depending on your measurement interval
  static_configs:
    - targets: ['<IP of VRS-Replace>']
      metrics_path: /metrics
```

Folgende Metriken werden geliefert

Metrik	Beschreibung
vrsreplace_info	{name="vrsreplace",arch="ESP32-D0WD",mac="84:0D:8E:E9:80:E0",version="0.9.8"} 1
vrsreplace_uptime	Zeit in Sekunden seit Start
vrsreplace_heap_size	Größe des Heap
vrsreplace_free_heap_size	Größe des freien Heap
vrsreplace_biggest_heap_block	Größe des größten freien Heap Blocks
vrsreplace_heap_min_free	Minimale Heap Größe seit Start
vrsreplace_wifi_rssi	Wert RSSI
vrsreplace_wifi_station	BSSID des Routers
vrsreplace_temperature	Aktuell gemessene Temperatur für alle Sensoren
vrsreplace_timer	Alle derzeit existierenden Timer mit aktuellem Status. id=TimerId, bez=Bezeichnung, relaistype=Solar/2 oder Warmwasser/1, timerprogram=Vergebenes Timerprogramm, tempsensor=Zugeordneter Temperatursensor



Statuscodes

Code	Status	Beschreibung
100	OK	System gestartet
101	OK	Tageswechsel
110	Error	Unbekannter Systemaufruf
111	OK	System wird neu gestartet
112	OK	NTP Uhrzeit aktualisiert
113	OK	Firmware aktualisiert
114	ERROR	Firmware konnte nicht aktualisiert werden
115	OK	ADC-Wandler erfolgreich deaktiviert
116	ERROR	ADC-Wandler konnten nicht deaktiviert werden
117	OK	Einstellungen gespeichert
118	OK	Einstellungen wurden erfolgreich gespeichert. Neustart erforderlich.
119	OK	System wurde erfolgreich zurückgesetzt
120	ERROR	Einstellungen konnten nicht gespeichert werden
121	ERROR	Lut file konnte nicht gefunden werden
122	ERROR	Keine SD Karte vorhanden und/oder nicht beschreibbar
125	ERROR	Wi-Fi Verbindung unterbrochen
126	OK	Wi-Fi Verbindung wieder hergestellt
127	OK	System wurde im Config-Modus gestartet
180	OK	Uhrzeit vom NTP-Server aktualisiert
181	ERROR	Uhrzeit konnte nicht gesetzt werden
182	OK	Zeitzone gesetzt
200	OK	Timer gestartet
201	OK	Timer gestoppt
202	OK	Timer wurde in den Wait-State versetzt
203	OK	Timer wurde restartet
204	OK	Alle Timer wurden gelöscht
205	ERROR	Timer konnten nicht gelöscht werden
206	OK	Timer wurde gelöscht
207	OK	Timer für Warmwasser hinzugefügt
208	OK	Timer für Solarsteuerung hinzugefügt
209	OK	Timer für Zirkulationspumpe hinzugefügt
210	OK	Timer für Warmwasser geändert
211	OK	Parameter für Solarsteuerung geändert
212	OK	Schaltzeiten für Zirkulationssysteme geändert
300	ERROR	Timer konnte nicht gestartet werden, da kein ID Parameter im Aufruf vorhanden



301	ERROR	Timer konnte nicht gestartet werden, da ID nicht vorhanden
302	ERROR	Timer mit der übergebenen ID, wurde bereits gestartet
303	ERROR	Ein Timer mit dem gewählten Relais-Type, wurde bereits gestartet
304	OK	Timer wurde deaktiviert
305	OK	Timer gestartet
306	ERROR	Timer konnte nicht gestoppt werden, kein ID Parameter im Aufruf vorhanden
307	OK	Timer gestoppt
308	ERROR	Timer konnte nicht gestoppt werden, da kein laufender Timer mit der ID gefunden wurde
309	ERROR	Timer(s) konnten nicht gelöscht werden, keine ID's übergeben
310	ERROR	Genereller Fehler beim Löschen der Timer
311	OK	Timer(s) erfolgreich gelöscht
312	ERROR	Timer nicht gefunden. ID nicht vorhanden
313	ERROR	Timer nicht gefunden. Keine ID beim Aufruf übergeben
314	ERROR	Timer konnte nicht geändert werden. Es wurde kein Timer übergeben
315	ERROR	Genereller Fehler beim Ändern des Timer
316	OK	Timer hinzugefügt
317	OK	Timer geändert
340	OK	Webservice Connect
341	OK	Erfolgreiche Abmeldung von Client
350	OK	Firmwareupdate gestartet
351	ERROR	Fehler beim Firmwareupdate
352	OK	Firmwareupdate erfolgreich ausgeführt.
353	ERROR	Parameter konnten nicht geladen werden.
360	ERROR	Config konnte nicht geändert werden
361	ERROR	Conifg nicht vorhanden
370	OK	MQTT gestartet
371	OK	MQTT gestoppt
372	ERROR	MQTT Fehler
373	OK	MQTT wurde initialisiert
374	ERROR	MQTT konnte nicht initialisiert werden



Zusätzlich verwendete Softwarekomponenten

Das System verwendet folgende zusätzliche OpenSource Komponenten.

- HTMX

<https://htmx.org/> <https://github.com/bigskysoftware/htmx>

- svg-gauge

<https://github.com/naikus/svg-gauge>

- uPlot

<https://github.com/leeoniya/uPlot>

- PsychicHttp

<https://github.com/hoeken/PsychicHttp>

Zukunft

Folgende „Futures“ sind geplant.

- Umstellung Entwicklungsumgebung
- Umstellung auf C++
- Neues FE ggf. auf Basis von „vue.js“
- Ethernet
- SSL
- Auth User/Password
- Dynamische Programme („Solar“/ „Legionellen“)
- mDNS
- Ertragskalkulation
- ggf. Verzicht auf SD-Card und nur noch Logging über externe Systeme.
- Unterstützung weiterer Temperaturfühler über Regelung mittels AD5270.
- Bessere Fehlererkennung und Benachrichtigung u.A. von „vergammelten Sensoren“.



Lizenz

GNU GENERAL PUBLIC LICENSE

Version 3, 29 June 2007

Copyright (C) 2007 Free Software Foundation, Inc. <<https://fsf.org/>>
Everyone is permitted to copy and distribute verbatim copies
of this license document, but changing it is not allowed.

Preamble

The GNU General Public License is a free, copyleft license for
software and other kinds of works.

The licenses for most software and other practical works are designed
to take away your freedom to share and change the works. By contrast,
the GNU General Public License is intended to guarantee your freedom to
share and change all versions of a program--to make sure it remains free
software for all its users. We, the Free Software Foundation, use the
GNU General Public License for most of our software; it applies also to
any other work released this way by its authors. You can apply it to
your programs, too.

When we speak of free software, we are referring to freedom, not
price. Our General Public Licenses are designed to make sure that you
have the freedom to distribute copies of free software (and charge for
them if you wish), that you receive source code or can get it if you
want it, that you can change the software or use pieces of it in new
free programs, and that you know you can do these things.

To protect your rights, we need to prevent others from denying you
these rights or asking you to surrender the rights. Therefore, you have
certain responsibilities if you distribute copies of the software, or if
you modify it: responsibilities to respect the freedom of others.

For example, if you distribute copies of such a program, whether
gratis or for a fee, you must pass on to the recipients the same
freedoms that you received. You must make sure that they, too, receive
or can get the source code. And you must show them these terms so they
know their rights.

Developers that use the GNU GPL protect your rights with two steps:
(1) assert copyright on the software, and (2) offer you this License
giving you legal permission to copy, distribute and/or modify it.

For the developers' and authors' protection, the GPL clearly explains
that there is no warranty for this free software. For both users' and
authors' sake, the GPL requires that modified versions be marked as
changed, so that their problems will not be attributed erroneously to
authors of previous versions.

Some devices are designed to deny users access to install or run
modified versions of the software inside them, although the manufacturer
can do so. This is fundamentally incompatible with the aim of
protecting users' freedom to change the software. The systematic
pattern of such abuse occurs in the area of products for individuals to
use, which is precisely where it is most unacceptable. Therefore, we
have designed this version of the GPL to prohibit the practice for those



products. If such problems arise substantially in other domains, we stand ready to extend this provision to those domains in future versions of the GPL, as needed to protect the freedom of users.

Finally, every program is threatened constantly by software patents. States should not allow patents to restrict development and use of software on general-purpose computers, but in those that do, we wish to avoid the special danger that patents applied to a free program could make it effectively proprietary. To prevent this, the GPL assures that patents cannot be used to render the program non-free.

The precise terms and conditions for copying, distribution and modification follow.

TERMS AND CONDITIONS

0. Definitions.

"This License" refers to version 3 of the GNU General Public License.

"Copyright" also means copyright-like laws that apply to other kinds of works, such as semiconductor masks.

"The Program" refers to any copyrightable work licensed under this License. Each licensee is addressed as "you". "Licensees" and "recipients" may be individuals or organizations.

To "modify" a work means to copy from or adapt all or part of the work in a fashion requiring copyright permission, other than the making of an exact copy. The resulting work is called a "modified version" of the earlier work or a work "based on" the earlier work.

A "covered work" means either the unmodified Program or a work based on the Program.

To "propagate" a work means to do anything with it that, without permission, would make you directly or secondarily liable for infringement under applicable copyright law, except executing it on a computer or modifying a private copy. Propagation includes copying, distribution (with or without modification), making available to the public, and in some countries other activities as well.

To "convey" a work means any kind of propagation that enables other parties to make or receive copies. Mere interaction with a user through a computer network, with no transfer of a copy, is not conveying.

An interactive user interface displays "Appropriate Legal Notices" to the extent that it includes a convenient and prominently visible feature that (1) displays an appropriate copyright notice, and (2) tells the user that there is no warranty for the work (except to the extent that warranties are provided), that licensees may convey the work under this License, and how to view a copy of this License. If the interface presents a list of user commands or options, such as a menu, a prominent item in the list meets this criterion.

1. Source Code.

The "source code" for a work means the preferred form of the work for making modifications to it. "Object code" means any non-source form of a work.



A "Standard Interface" means an interface that either is an official standard defined by a recognized standards body, or, in the case of interfaces specified for a particular programming language, one that is widely used among developers working in that language.

The "System Libraries" of an executable work include anything, other than the work as a whole, that (a) is included in the normal form of packaging a Major Component, but which is not part of that Major Component, and (b) serves only to enable use of the work with that Major Component, or to implement a Standard Interface for which an implementation is available to the public in source code form. A "Major Component", in this context, means a major essential component (kernel, window system, and so on) of the specific operating system (if any) on which the executable work runs, or a compiler used to produce the work, or an object code interpreter used to run it.

The "Corresponding Source" for a work in object code form means all the source code needed to generate, install, and (for an executable work) run the object code and to modify the work, including scripts to control those activities. However, it does not include the work's System Libraries, or general-purpose tools or generally available free programs which are used unmodified in performing those activities but which are not part of the work. For example, Corresponding Source includes interface definition files associated with source files for the work, and the source code for shared libraries and dynamically linked subprograms that the work is specifically designed to require, such as by intimate data communication or control flow between those subprograms and other parts of the work.

The Corresponding Source need not include anything that users can regenerate automatically from other parts of the Corresponding Source.

The Corresponding Source for a work in source code form is that same work.

2. Basic Permissions.

All rights granted under this License are granted for the term of copyright on the Program, and are irrevocable provided the stated conditions are met. This License explicitly affirms your unlimited permission to run the unmodified Program. The output from running a covered work is covered by this License only if the output, given its content, constitutes a covered work. This License acknowledges your rights of fair use or other equivalent, as provided by copyright law.

You may make, run and propagate covered works that you do not convey, without conditions so long as your license otherwise remains in force. You may convey covered works to others for the sole purpose of having them make modifications exclusively for you, or provide you with facilities for running those works, provided that you comply with the terms of this License in conveying all material for which you do not control copyright. Those thus making or running the covered works for you must do so exclusively on your behalf, under your direction and control, on terms that prohibit them from making any copies of your copyrighted material outside their relationship with you.

Conveying under any other circumstances is permitted solely under the conditions stated below. Sublicensing is not allowed; section 10



makes it unnecessary.

3. Protecting Users' Legal Rights From Anti-Circumvention Law.

No covered work shall be deemed part of an effective technological measure under any applicable law fulfilling obligations under article 11 of the WIPO copyright treaty adopted on 20 December 1996, or similar laws prohibiting or restricting circumvention of such measures.

When you convey a covered work, you waive any legal power to forbid circumvention of technological measures to the extent such circumvention is effected by exercising rights under this License with respect to the covered work, and you disclaim any intention to limit operation or modification of the work as a means of enforcing, against the work's users, your or third parties' legal rights to forbid circumvention of technological measures.

4. Conveying Verbatim Copies.

You may convey verbatim copies of the Program's source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice; keep intact all notices stating that this License and any non-permissive terms added in accord with section 7 apply to the code; keep intact all notices of the absence of any warranty; and give all recipients a copy of this License along with the Program.

You may charge any price or no price for each copy that you convey, and you may offer support or warranty protection for a fee.

5. Conveying Modified Source Versions.

You may convey a work based on the Program, or the modifications to produce it from the Program, in the form of source code under the terms of section 4, provided that you also meet all of these conditions:

- a) The work must carry prominent notices stating that you modified it, and giving a relevant date.
- b) The work must carry prominent notices stating that it is released under this License and any conditions added under section 7. This requirement modifies the requirement in section 4 to "keep intact all notices".
- c) You must license the entire work, as a whole, under this License to anyone who comes into possession of a copy. This License will therefore apply, along with any applicable section 7 additional terms, to the whole of the work, and all its parts, regardless of how they are packaged. This License gives no permission to license the work in any other way, but it does not invalidate such permission if you have separately received it.
- d) If the work has interactive user interfaces, each must display Appropriate Legal Notices; however, if the Program has interactive interfaces that do not display Appropriate Legal Notices, your work need not make them do so.

A compilation of a covered work with other separate and independent works, which are not by their nature extensions of the covered work,



and which are not combined with it such as to form a larger program, in or on a volume of a storage or distribution medium, is called an "aggregate" if the compilation and its resulting copyright are not used to limit the access or legal rights of the compilation's users beyond what the individual works permit. Inclusion of a covered work in an aggregate does not cause this License to apply to the other parts of the aggregate.

6. Conveying Non-Source Forms.

You may convey a covered work in object code form under the terms of sections 4 and 5, provided that you also convey the machine-readable Corresponding Source under the terms of this License, in one of these ways:

- a) Convey the object code in, or embodied in, a physical product (including a physical distribution medium), accompanied by the Corresponding Source fixed on a durable physical medium customarily used for software interchange.
- b) Convey the object code in, or embodied in, a physical product (including a physical distribution medium), accompanied by a written offer, valid for at least three years and valid for as long as you offer spare parts or customer support for that product model, to give anyone who possesses the object code either (1) a copy of the Corresponding Source for all the software in the product that is covered by this License, on a durable physical medium customarily used for software interchange, for a price no more than your reasonable cost of physically performing this conveying of source, or (2) access to copy the Corresponding Source from a network server at no charge.
- c) Convey individual copies of the object code with a copy of the written offer to provide the Corresponding Source. This alternative is allowed only occasionally and noncommercially, and only if you received the object code with such an offer, in accord with subsection 6b.
- d) Convey the object code by offering access from a designated place (gratis or for a charge), and offer equivalent access to the Corresponding Source in the same way through the same place at no further charge. You need not require recipients to copy the Corresponding Source along with the object code. If the place to copy the object code is a network server, the Corresponding Source may be on a different server (operated by you or a third party) that supports equivalent copying facilities, provided you maintain clear directions next to the object code saying where to find the Corresponding Source. Regardless of what server hosts the Corresponding Source, you remain obligated to ensure that it is available for as long as needed to satisfy these requirements.
- e) Convey the object code using peer-to-peer transmission, provided you inform other peers where the object code and Corresponding Source of the work are being offered to the general public at no charge under subsection 6d.

A separable portion of the object code, whose source code is excluded from the Corresponding Source as a System Library, need not be included in conveying the object code work.



A "User Product" is either (1) a "consumer product", which means any tangible personal property which is normally used for personal, family, or household purposes, or (2) anything designed or sold for incorporation into a dwelling. In determining whether a product is a consumer product, doubtful cases shall be resolved in favor of coverage. For a particular product received by a particular user, "normally used" refers to a typical or common use of that class of product, regardless of the status of the particular user or of the way in which the particular user actually uses, or expects or is expected to use, the product. A product is a consumer product regardless of whether the product has substantial commercial, industrial or non-consumer uses, unless such uses represent the only significant mode of use of the product.

"Installation Information" for a User Product means any methods, procedures, authorization keys, or other information required to install and execute modified versions of a covered work in that User Product from a modified version of its Corresponding Source. The information must suffice to ensure that the continued functioning of the modified object code is in no case prevented or interfered with solely because modification has been made.

If you convey an object code work under this section in, or with, or specifically for use in, a User Product, and the conveying occurs as part of a transaction in which the right of possession and use of the User Product is transferred to the recipient in perpetuity or for a fixed term (regardless of how the transaction is characterized), the Corresponding Source conveyed under this section must be accompanied by the Installation Information. But this requirement does not apply if neither you nor any third party retains the ability to install modified object code on the User Product (for example, the work has been installed in ROM).

The requirement to provide Installation Information does not include a requirement to continue to provide support service, warranty, or updates for a work that has been modified or installed by the recipient, or for the User Product in which it has been modified or installed. Access to a network may be denied when the modification itself materially and adversely affects the operation of the network or violates the rules and protocols for communication across the network.

Corresponding Source conveyed, and Installation Information provided, in accord with this section must be in a format that is publicly documented (and with an implementation available to the public in source code form), and must require no special password or key for unpacking, reading or copying.

7. Additional Terms.

"Additional permissions" are terms that supplement the terms of this License by making exceptions from one or more of its conditions. Additional permissions that are applicable to the entire Program shall be treated as though they were included in this License, to the extent that they are valid under applicable law. If additional permissions apply only to part of the Program, that part may be used separately under those permissions, but the entire Program remains governed by this License without regard to the additional permissions.

When you convey a copy of a covered work, you may at your option remove any additional permissions from that copy, or from any part of it. (Additional permissions may be written to require their own



removal in certain cases when you modify the work.) You may place additional permissions on material, added by you to a covered work, for which you have or can give appropriate copyright permission.

Notwithstanding any other provision of this License, for material you add to a covered work, you may (if authorized by the copyright holders of that material) supplement the terms of this License with terms:

- a) Disclaiming warranty or limiting liability differently from the terms of sections 15 and 16 of this License; or
- b) Requiring preservation of specified reasonable legal notices or author attributions in that material or in the Appropriate Legal Notices displayed by works containing it; or
- c) Prohibiting misrepresentation of the origin of that material, or requiring that modified versions of such material be marked in reasonable ways as different from the original version; or
- d) Limiting the use for publicity purposes of names of licensors or authors of the material; or
- e) Declining to grant rights under trademark law for use of some trade names, trademarks, or service marks; or
- f) Requiring indemnification of licensors and authors of that material by anyone who conveys the material (or modified versions of it) with contractual assumptions of liability to the recipient, for any liability that these contractual assumptions directly impose on those licensors and authors.

All other non-permissive additional terms are considered "further restrictions" within the meaning of section 10. If the Program as you received it, or any part of it, contains a notice stating that it is governed by this License along with a term that is a further restriction, you may remove that term. If a license document contains a further restriction but permits relicensing or conveying under this License, you may add to a covered work material governed by the terms of that license document, provided that the further restriction does not survive such relicensing or conveying.

If you add terms to a covered work in accord with this section, you must place, in the relevant source files, a statement of the additional terms that apply to those files, or a notice indicating where to find the applicable terms.

Additional terms, permissive or non-permissive, may be stated in the form of a separately written license, or stated as exceptions; the above requirements apply either way.

8. Termination.

You may not propagate or modify a covered work except as expressly provided under this License. Any attempt otherwise to propagate or modify it is void, and will automatically terminate your rights under this License (including any patent licenses granted under the third paragraph of section 11).

However, if you cease all violation of this License, then your license from a particular copyright holder is reinstated (a)



provisionally, unless and until the copyright holder explicitly and finally terminates your license, and (b) permanently, if the copyright holder fails to notify you of the violation by some reasonable means prior to 60 days after the cessation.

Moreover, your license from a particular copyright holder is reinstated permanently if the copyright holder notifies you of the violation by some reasonable means, this is the first time you have received notice of violation of this License (for any work) from that copyright holder, and you cure the violation prior to 30 days after your receipt of the notice.

Termination of your rights under this section does not terminate the licenses of parties who have received copies or rights from you under this License. If your rights have been terminated and not permanently reinstated, you do not qualify to receive new licenses for the same material under section 10.

9. Acceptance Not Required for Having Copies.

You are not required to accept this License in order to receive or run a copy of the Program. Ancillary propagation of a covered work occurring solely as a consequence of using peer-to-peer transmission to receive a copy likewise does not require acceptance. However, nothing other than this License grants you permission to propagate or modify any covered work. These actions infringe copyright if you do not accept this License. Therefore, by modifying or propagating a covered work, you indicate your acceptance of this License to do so.

10. Automatic Licensing of Downstream Recipients.

Each time you convey a covered work, the recipient automatically receives a license from the original licensors, to run, modify and propagate that work, subject to this License. You are not responsible for enforcing compliance by third parties with this License.

An "entity transaction" is a transaction transferring control of an organization, or substantially all assets of one, or subdividing an organization, or merging organizations. If propagation of a covered work results from an entity transaction, each party to that transaction who receives a copy of the work also receives whatever licenses to the work the party's predecessor in interest had or could give under the previous paragraph, plus a right to possession of the Corresponding Source of the work from the predecessor in interest, if the predecessor has it or can get it with reasonable efforts.

You may not impose any further restrictions on the exercise of the rights granted or affirmed under this License. For example, you may not impose a license fee, royalty, or other charge for exercise of rights granted under this License, and you may not initiate litigation (including a cross-claim or counterclaim in a lawsuit) alleging that any patent claim is infringed by making, using, selling, offering for sale, or importing the Program or any portion of it.

11. Patents.

A "contributor" is a copyright holder who authorizes use under this License of the Program or a work on which the Program is based. The work thus licensed is called the contributor's "contributor version".



A contributor's "essential patent claims" are all patent claims owned or controlled by the contributor, whether already acquired or hereafter acquired, that would be infringed by some manner, permitted by this License, of making, using, or selling its contributor version, but do not include claims that would be infringed only as a consequence of further modification of the contributor version. For purposes of this definition, "control" includes the right to grant patent sublicenses in a manner consistent with the requirements of this License.

Each contributor grants you a non-exclusive, worldwide, royalty-free patent license under the contributor's essential patent claims, to make, use, sell, offer for sale, import and otherwise run, modify and propagate the contents of its contributor version.

In the following three paragraphs, a "patent license" is any express agreement or commitment, however denominated, not to enforce a patent (such as an express permission to practice a patent or covenant not to sue for patent infringement). To "grant" such a patent license to a party means to make such an agreement or commitment not to enforce a patent against the party.

If you convey a covered work, knowingly relying on a patent license, and the Corresponding Source of the work is not available for anyone to copy, free of charge and under the terms of this License, through a publicly available network server or other readily accessible means, then you must either (1) cause the Corresponding Source to be so available, or (2) arrange to deprive yourself of the benefit of the patent license for this particular work, or (3) arrange, in a manner consistent with the requirements of this License, to extend the patent license to downstream recipients. "Knowingly relying" means you have actual knowledge that, but for the patent license, your conveying the covered work in a country, or your recipient's use of the covered work in a country, would infringe one or more identifiable patents in that country that you have reason to believe are valid.

If, pursuant to or in connection with a single transaction or arrangement, you convey, or propagate by procuring conveyance of, a covered work, and grant a patent license to some of the parties receiving the covered work authorizing them to use, propagate, modify or convey a specific copy of the covered work, then the patent license you grant is automatically extended to all recipients of the covered work and works based on it.

A patent license is "discriminatory" if it does not include within the scope of its coverage, prohibits the exercise of, or is conditioned on the non-exercise of one or more of the rights that are specifically granted under this License. You may not convey a covered work if you are a party to an arrangement with a third party that is in the business of distributing software, under which you make payment to the third party based on the extent of your activity of conveying the work, and under which the third party grants, to any of the parties who would receive the covered work from you, a discriminatory patent license (a) in connection with copies of the covered work conveyed by you (or copies made from those copies), or (b) primarily for and in connection with specific products or compilations that contain the covered work, unless you entered into that arrangement, or that patent license was granted, prior to 28 March 2007.

Nothing in this License shall be construed as excluding or limiting



any implied license or other defenses to infringement that may otherwise be available to you under applicable patent law.

12. No Surrender of Others' Freedom.

If conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot convey a covered work so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not convey it at all. For example, if you agree to terms that obligate you to collect a royalty for further conveying from those to whom you convey the Program, the only way you could satisfy both those terms and this License would be to refrain entirely from conveying the Program.

13. Use with the GNU Affero General Public License.

Notwithstanding any other provision of this License, you have permission to link or combine any covered work with a work licensed under version 3 of the GNU Affero General Public License into a single combined work, and to convey the resulting work. The terms of this License will continue to apply to the part which is the covered work, but the special requirements of the GNU Affero General Public License, section 13, concerning interaction through a network will apply to the combination as such.

14. Revised Versions of this License.

The Free Software Foundation may publish revised and/or new versions of the GNU General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Program specifies that a certain numbered version of the GNU General Public License "or any later version" applies to it, you have the option of following the terms and conditions either of that numbered version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of the GNU General Public License, you may choose any version ever published by the Free Software Foundation.

If the Program specifies that a proxy can decide which future versions of the GNU General Public License can be used, that proxy's public statement of acceptance of a version permanently authorizes you to choose that version for the Program.

Later license versions may give you additional or different permissions. However, no additional obligations are imposed on any author or copyright holder as a result of your choosing to follow a later version.

15. Disclaimer of Warranty.

THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM



IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

16. Limitation of Liability.

IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MODIFIES AND/OR CONVEYS THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

17. Interpretation of Sections 15 and 16.

If the disclaimer of warranty and limitation of liability provided above cannot be given local legal effect according to their terms, reviewing courts shall apply local law that most closely approximates an absolute waiver of all civil liability in connection with the Program, unless a warranty or assumption of liability accompanies a copy of the Program in return for a fee.

END OF TERMS AND CONDITIONS

How to Apply These Terms to Your New Programs

If you develop a new program, and you want it to be of the greatest possible use to the public, the best way to achieve this is to make it free software which everyone can redistribute and change under these terms.

To do so, attach the following notices to the program. It is safest to attach them to the start of each source file to most effectively state the exclusion of warranty; and each file should have at least the "copyright" line and a pointer to where the full notice is found.

```
<one line to give the program's name and a brief idea of what it does.>
Copyright (C) <year> <name of author>
```

```
This program is free software: you can redistribute it and/or modify
it under the terms of the GNU General Public License as published by
the Free Software Foundation, either version 3 of the License, or
(at your option) any later version.
```

```
This program is distributed in the hope that it will be useful,
but WITHOUT ANY WARRANTY; without even the implied warranty of
MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
GNU General Public License for more details.
```

```
You should have received a copy of the GNU General Public License
along with this program. If not, see <https://www.gnu.org/licenses/>.
```

Also add information on how to contact you by electronic and paper mail.

If the program does terminal interaction, make it output a short notice like this when it starts in an interactive mode:

```
<program> Copyright (C) <year> <name of author>
This program comes with ABSOLUTELY NO WARRANTY; for details type `show w'.
```



This is free software, and you are welcome to redistribute it under certain conditions; type ``show c'` for details.

The hypothetical commands ``show w'` and ``show c'` should show the appropriate parts of the General Public License. Of course, your program's commands might be different; for a GUI interface, you would use an "about box".

You should also get your employer (if you work as a programmer) or school, if any, to sign a "copyright disclaimer" for the program, if necessary. For more information on this, and how to apply and follow the GNU GPL, see [<https://www.gnu.org/licenses/>](https://www.gnu.org/licenses/).

The GNU General Public License does not permit incorporating your program into proprietary programs. If your program is a subroutine library, you may consider it more useful to permit linking proprietary applications with the library. If this is what you want to do, use the GNU Lesser General Public License instead of this License. But first, please read [<https://www.gnu.org/licenses/why-not-lgpl.html>](https://www.gnu.org/licenses/why-not-lgpl.html).

