



TEMPERATURLOGGER

HANDBUCH

Inhalt

Verwendung	4
Ersteinrichtung	5
Materialien	6
Vorbereitung	7
Installation des Betriebssystems Raspbian.....	7
Aktivieren des SSH-Services.....	8
Konfiguration.....	9
Standard Anmeldedaten.....	9
Grundkonfiguration.....	10
RAID-Konfiguration	12
W-LAN Access Point einrichten.....	13
Installation	16
IOT-Stack Installation	16
Container Konfiguration.....	18
Pi-Kiosk.....	28
Automatisierung: Einhängen und Backups	29
Sicherheit.....	32
Standardbenutzer sperren.....	32
Nutzername und Passwort für Node-RED anlegen (optional)	32
Bedienung	33
Backup.....	34
Eichung der Sensoren	34
Erweiterung	35
Arduino Image	36
Arduino Konfigurieren.....	36
Hochladen	36
Debugging mit Node-Red	37
Problemsuche	38
Initialisierungsprobleme:	39
Boot Probleme.....	39

Das Gerät bootet nicht.....	39
Probleme mit Raspberry Pi	40
Es werden keine Dateien auf den USB-Stick übertragen	40
GRAFANA reagiert nicht mehr oder hat sich geschlossen.....	40
Alle Docker Container herunterfahren/starten.....	40
Unbekannte oder falsche Einträge in der Datenbank.....	40
Probleme mit dem Microcontroller.....	41
DEBUG am Temperatur-Display	41
Temperatur: -127.0.....	41
Display funktioniert nicht.....	42
Zugangsdaten	43
Ansprechpartner.....	Fehler! Textmarke nicht definiert.
Notizen	44

Verwendung

Mehrere Mikrocontroller lesen jede Sekunde die Temperatur aus und senden die jeweilige Messung, nach einer festgelegten Zeitperiode, an den Raspberry Pi. Dieser verwaltet die Messdaten in einer Datenbank, zeigt sie grafisch auf einem Bildschirm an (Grafana) und stellt sie zur Sicherung via USB Plug 'n' Play bereit.

Ersteinrichtung

Materialien

Bezeichnung	Verwendungszweck
Raspberry Pi 4B, 4 GiB RAM Version	Basisstation; Verwaltung, Verarbeitung und Darstellung der ausgewerteten Dateien
USB-C Netzteil	Netzteil für Raspberry Pi
7" touch Display	Anzeigen aller Messungen als Graph
Mini HDMI zu Mikro HDMI Kabel	Verbindet den Raspberry Pi mit dem Display
2x Mikro USB zu USB-A Kabel	Stromversorgung des Displays und Touch-Ansteuerung
microSD Karte, 16 GB	Speicherplatz für Dateisystem und Applikationen
WLAN USB 2.0 Dongle	Verbesserter W-LAN Access Point
USB 3.0 HUB mit externem 5V Netzteil	Mehr USB-Ports und zusätzliche Spannungsversorgung
2x SSD 120 GB	Speicherplatz für Datenbank und Logdateien
2x USB zu SATA Kabel	Anschluss für SSDs
2x USB-Verlängerung	Anschluss für USB-Stick und W-LAN Antenne
USB-Stick 64 GB	Für Plug'n Play Backups
5x ESP8266 Mikrocontroller	Auswertung der Daten am Standort und Übertragung an die Basisstation
5x DS18B20 Temperatursensor	Zum Messen der Temperatur
5x Display 1.3 Zoll	Anzeigen der Messwerte der Mikrocontroller
5x Widerstand 4.7k Ohm	Schützt den Datenbus des Sensors und Mikrocontrollers
20x JST-XH 3-Pin Female/Male	Verbindungsstück von Sensor und PCB
5x PCB im Eigendesign	Platine für die Anbindung von Microcontroller, Sensor, Widerstand und Display
5x USB-A zu micro USB 5m	Netzanschluss für die Mikrocontroller
5x Antenne U.FL	Für Wifi-Konnektivität am Mikrocontroller
5x Microcontrollergehäuse im Eigendesign	Schutz der Komponente
1x Raspberrypi-Gehäuse im Eigendesign	Schutz der Komponente

Vorbereitung

Installation des Betriebssystems Raspbian

Herunterladen des „Raspberry Pi Imager“

<https://www.raspberrypi.com/software/>

Mit dem Button „OS Wählen“ das Image „Raspberry Pi OS (x32)“ auswählen

Einstellungen (Zahnrad):

- ➔ Host-Name setzen
- ➔ SSH Aktivieren
- ➔ Benutzername und Passwort setzen
- ➔ Wifi einrichten
- ➔ Sprache/Keyboard setzen

SD-Karte Wählen

Abschließend auf „Schreiben“ drücken.

SD-Karte entfernen und in den Raspberry Pi einlegen. Das Gerät nun mit Strom versorgen.

Aktivieren des SSH-Services

Am einfachsten ist die Erstkonfiguration des Raspberry Pi via Tastatur, Maus und Bildschirm direkt am Endgerät. Hat man diese nicht, dann:

Der SSH-Service kann auf zwei unterschiedliche Weisen gestartet werden

1. Aktivierung vor der Inbetriebnahme des Raspberry Pi's

Nach dem Schreiben des Images auf die SD-Karte, kann im Datei-Explorer auf dem Laufwerk eine Datei mit dem Namen „ssh“ erstellt werden (ohne Dateiendung)

2. Aktivierung am Raspberry Pi selbst

Nach dem Laden der Desktopoberfläche wird ein Konfigurationsmenü gestartet, hier kann auch der SSH Service eingeschaltet werden.

Alternativ kann dieser auch unter dem Menüpunkt „Einstellungen“ unter Start aktiviert werden.

Verwendung:

```
ssh user@<ip-of-raspberry>
```


Konfiguration

Standard Anmeldedaten

Benutzername	pi
Passwort	raspberry

Veraltet, neue Versionen des Raspberry Pis setzt keinen Benutzernamen und Passwort voraus und wird stattdessen beim Hochladen des Images auf die SD-Karte gesetzt oder manuell mittels Ersteinrichtung.

Grundkonfiguration

NEUEN BENUTZER ANLEGEN

```
useradd admin  
passwd <passwort>
```

GRUPPENRICHTLINIE BEARBEITEN

```
groupadd docker  
usermod -aG sudo,docker admin
```

SYSTEM AKTUALISIEREN

```
sudo apt update  
sudo apt upgrade
```

RASPI-CONFIG

```
sudo raspi-config
```

Unter „Localisation Options“

Locale

Hier wählen wir die folgenden Optionen (mit Leertaste).

```
de_DE.UTF-8 UTF-8  
de_DE ISO-8859-1  
de_DE@euro ISO-8859-15  
en_US.UTF-8 UTF-8
```

Die Standard-Standorteinstellung für das System ist „de_DE.UTF-8“

Timezone

Europe / Berlin

Keyboard

Generische PC-Tastatur mit 105 Tasten (Intl) / Deutsch / Standard

WLAN-Country

DE Germany

HOSTNAME ÄNDERN

Hier setzen wir den Gerätenamen des Raspberry Pi's

```
nano /etc/hostname
```

```
temperaturlogger
```

ENERGIEEINSTELLUNGEN ÄNDERN

Einstellungen -> Energieeinstellungen

Hier den Ruhezustand und das Ausschalten des Displays deaktivieren.

Falls das Display vom Raspberry Pi getrennt wird, - muss sichergestellt werden, dass das Display wieder ein Bild anzeigt, wenn es wieder verbunden wird:

```
nano /boot/config.txt
```

```
hdmi_force_hotplug = 1
```

RAID-Konfiguration

Zum Erstellen und Verwalten von Raiden benötigen wir das Programm „mdadm“, dass wir mit folgendem Befehl herunterladen:

```
sudo apt install mdadm
```

RAID 1 ERSTELLEN

Zuerst machen wir uns einen Überblick der angeschlossenen Massenspeicher und notieren uns die Geräte, mit denen wir unser RAID planen.

```
lsblk
```

Hier erstellen wir ein Software-RAID Level 1 mit den beiden Geräten sda und sdb.

```
mdadm --create /dev/md0 --level=1 --raid-devices=2 /dev/sda1 /dev/sdb1
```

Nun können wir uns den Konfigurationsfortschritt und die Konfiguration ansehen:

```
cat /proc/mdstat
```

```
mdadm -D /dev/md0
```

DATEISYSTEM FORMATIEREN

```
sudo mkfs.ext4 /dev/md0
```

AUTOMATISCHES EINHÄNGEN AKTIVIEREN

Zuerst benötigen wir einen Einhängpunkt im Dateisystem:

```
sudo mkdir /mnt/raid
```

Der Eintrag unter /etc/fstab sollte dann so aussehen:

```
/dev/md0    /mnt/raid   ext4        defaults,auto,users,rw,nofail,exec 0 0
```

W-LAN Access Point einrichten

SOFTWARE INSTALLATION

```
sudo apt install -y dnsmasq hostapd
```

```
sudo DEBIAN_FRONTEND=noninteractive apt install -y \  
netfilter-persistent iptables-persistent
```

```
sudo systemctl stop dnsmasq  
sudo systemctl unmask hostapd  
sudo systemctl enable hostapd
```

NETZWERK ROUTER EINRICHTEN

Statische IP-Adresse setzen

```
sudo nano /etc/dhcpd.conf
```

```
interface wlan0  
    static ip_address=10.0.1.1/24  
    nohook wpa_supplicant
```

DHCP und DNS-Dienst Konfigurieren

```
sudo mv /etc/dnsmasq.conf /etc/dnsmasq.conf.backup
```

```
sudo nano /etc/dnsmasq.conf
```

```
interface=wlan0 # Listening interface
dhcp-range=10.0.1.2,10.0.1.128,255.255.255.0,24h
                # Pool of IP addresses served via DHCP
domain=wlan     # Local wireless DNS domain
address=/temperaturlogger.wlan/10.0.1.1
                # Alias for this router
```

WiFi-Blockierung aufheben

```
sudo rfkill unblock wlan
```

Konfigurieren der „Access Point“ Software

```
sudo nano /etc/hostapd/hostapd.conf
```

```
country_code=DE
interface=wlan0
ssid=tempLOG
hw_mode=g
channel=7
macaddr_acl=0
auth_algs=1
ignore_broadcast_ssid=0
wpa=2
wpa_passphrase=Wasserwaage
wpa_key_mgmt=WPA-PSK
wpa_pairwise=TKIP
rsn_pairwise=CCMP
```

Mit „sudo reboot“ neustarten

INSTALL FIRMWARE (W-LAN STICK)

In diesem Fall wird ein Wifi-Dongle mit dem Chipset Realtek RTL8811AU

Mit folgendem Kommando kann man das Gerät und dessen Chipset ausmachen:

```
lsusb
```

Installation der benötigten Pakete:

```
apt install -y raspberrypi-kernel-headers build-essential libncurses5-dev  
bc python git
```

```
wget https://raw.githubusercontent.com/notro/rpi-source/master/rpi-source
```

```
chmod +x rpi-source
```

```
./rpi-source
```

```
git clone https://github.com/gnab/rtl8812au.git  
cd rtl8812au/
```

```
nano Makefile
```

```
CONFIG_PLATFORM_I386_PC = n  
CONFIG_PLATFORM_ARM_RPI = y
```

```
chmod +x install.sh  
./install.sh
```

Installation

IOT-Stack Installation

Die Installation des IOT-Stacks für einfache Container-Virtualisierung

Zuerst benötigen wir ein paar Grundlegende Softwarepakete

```
sudo apt install -y curl  
curl -fsSL  
https://raw.githubusercontent.com/SensorsIot/IOTstack/master/install  
.sh | bash
```

Wir wählen <Ja> zum Installieren von „Docker“ und „docker-compose“

Nach der Installation starten wir das System neu

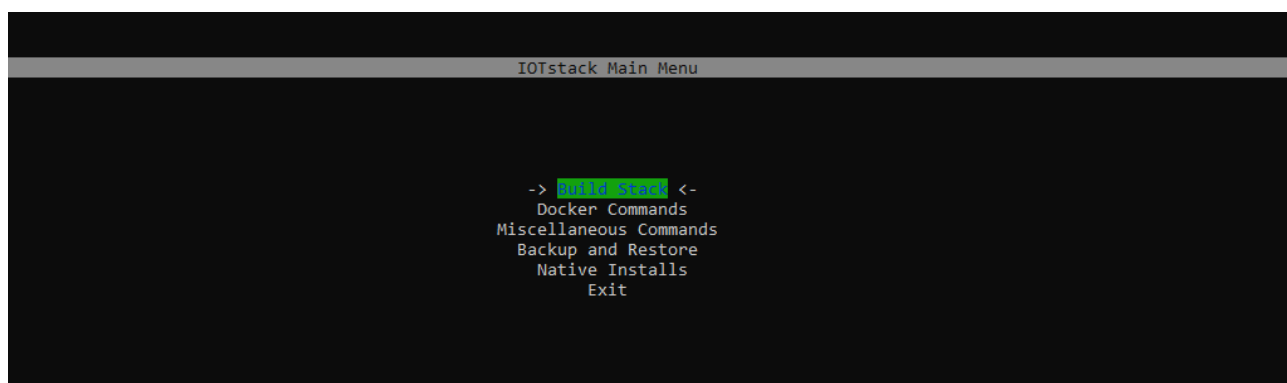
```
reboot
```

Jetzt starten wir die Installation des IOT-Stack's

```
cd ~/IOTstack  
./menu.sh
```

Wir werden nach zusätzlichen Abhängigkeiten gefragt, welche installiert werden müssen. Wir wählen <Ja>.

Jetzt sollten wir im Hauptmenü sein:



Wir bestätigen < Build Stack >

Und wählen folgende Container aus:

- grafana
- influxdb
- mosquitto
- nodered

Und bestätigen mit [Enter]

Nun können wir im Menü-Eintrag Docker mit <Start Stack> die ausgewählten Container herunterladen und starten.

Haben wir alles richtig gemacht, bekommen wir die folgende Benachrichtigung zum Schluss:

```
Successfully built 427e12f3a9b9
Successfully tagged iotstack_nodered:latest
WARNING: Image for service nodered was built because it did not already exist.
Creating influxdb ... done
Creating grafana ... done
Creating mosquitto ... done
Creating nodered ... done

Stack Started
Process terminated. Press [Enter] to show menu and continue.
```

Wir bestätigen mit [Enter]

Um Statusinformationen zu den Containern zu bekommen, geben wir folgendes Kommando ein

```
docker ps
```

Container Konfiguration

INFLUXDB

Damit wir mit influxdb arbeiten können, verwenden wir folgenden Befehl:

```
docker exec -it influxdb influx
```

Hier legen wir die Datenbank „temperature“ an und lassen sie uns nachträglich anzeigen

```
create database temperature
```

```
show databases
```

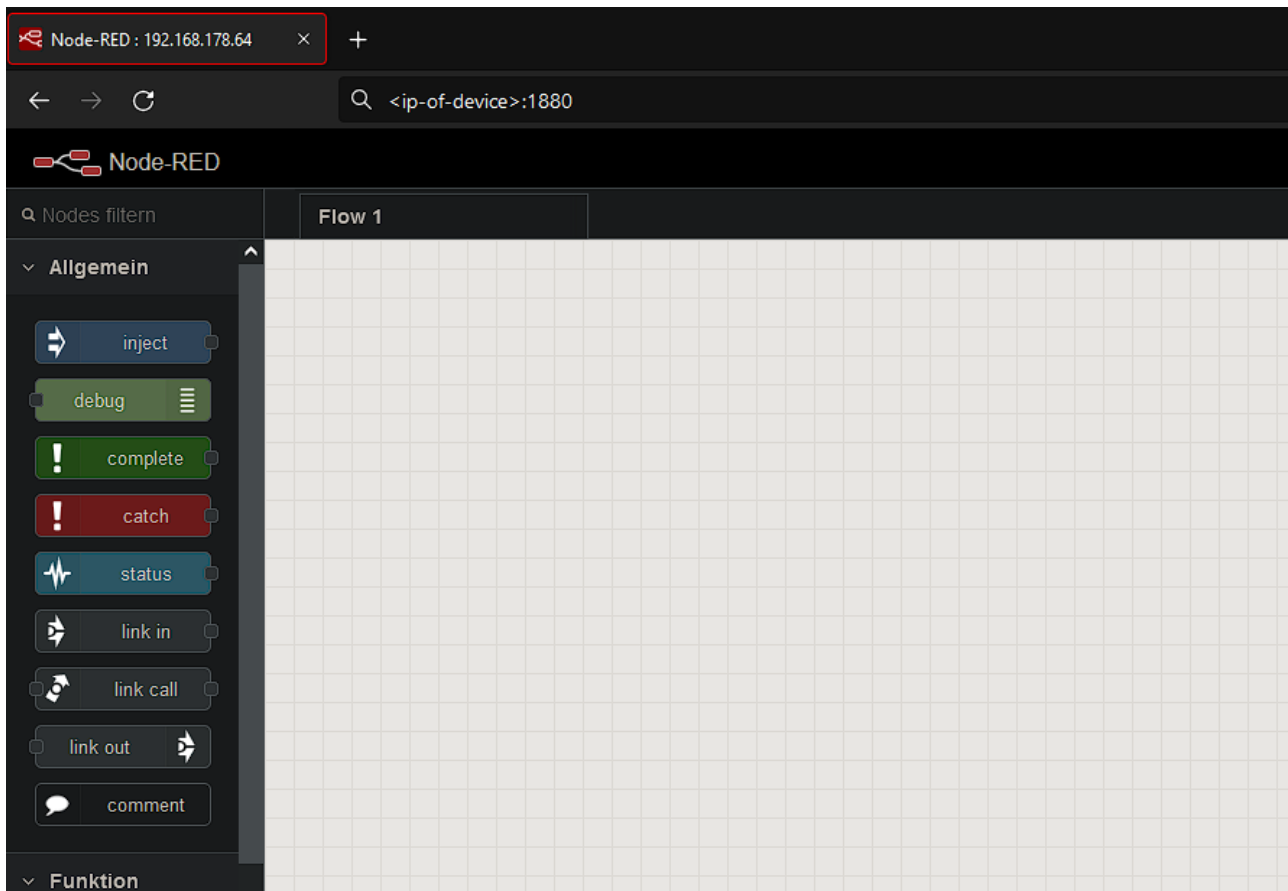
Danach logen wir uns mit dem Befehl „exit“ wieder aus dem Container aus.

MOSQUITTO

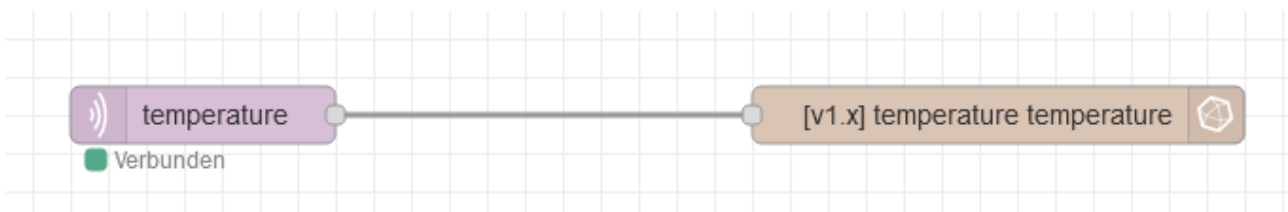
Standardmäßig wird hier nichts eingestellt. Um eine Authentifikation zu fordern, siehe unten in der Kategorie Ersteinrichtung/Sicherheit.

NODE-RED

Im Browser kann mit der Geräte IP (bzw. localhost) und dem Port 1880 auf die Website zur Konfiguration zugegriffen werden:




Wir nehmen die Nodes „mqtt in“ und „influxdb out“ und verbinden Sie wie unten gezeigt:












Konfiguration der Nodes




MQTT


Hier geben wir den Namen, die Topic und die Serveranbindung (Host mit Port, Version und [falls vorhanden] Login) an.

 **Eigenschaften**

 Server	<div>mosquitto:1883</div> <div>▼</div>	
Action	<div>Subscribe to single topic</div> <div>▼</div>	
 Topic	<div>temperature</div>	
 QoS	<div>2</div> <div>▼</div>	
 Ausgang	<div>Auto-Erkennung (string oder buffer)</div> <div>▼</div>	
 Name	<div>sensor</div>	


 **Eigenschaften**  

 **Name**

Verbindung



Sicherheit


Nachrichten



 **Server** **Port**


☒ Connect automatically

☐ TLS

 **Protokoll** 

 **Client-ID**


 **Keep-Alive** 

 **Session** ☒ Bereinigte Sitzung (clean session) verwenden


InfluxDB

Hier beschriften wir den Namen der Node, den Tabellennamen (Measurement) und die Datenbankverbindung (Version, Host mit Port, Datenbank sowie Login). [Benutzername nodered, Passwort nodered]


Eigenschaften


 Name

temperature

 Server

[v1.x] docker_influxdb



 Measurement

temperature

☐ Advanced Query Options

Tip: If no retention policy is specified, **autogen** will be assumed.

Eigenschaften

 Name

docker_influxdb

 Version

1.x



 Host

influxdb

Port

8086

 Database

temperature



Benutzername

nodered



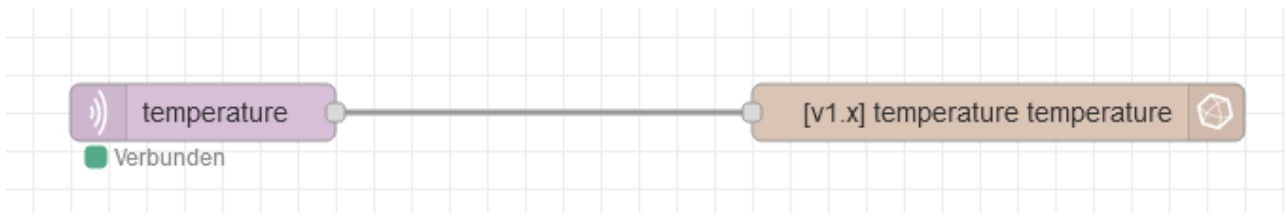
Passwort

●●●●●●●●

☐ Enable secure (SSL/TLS) connection

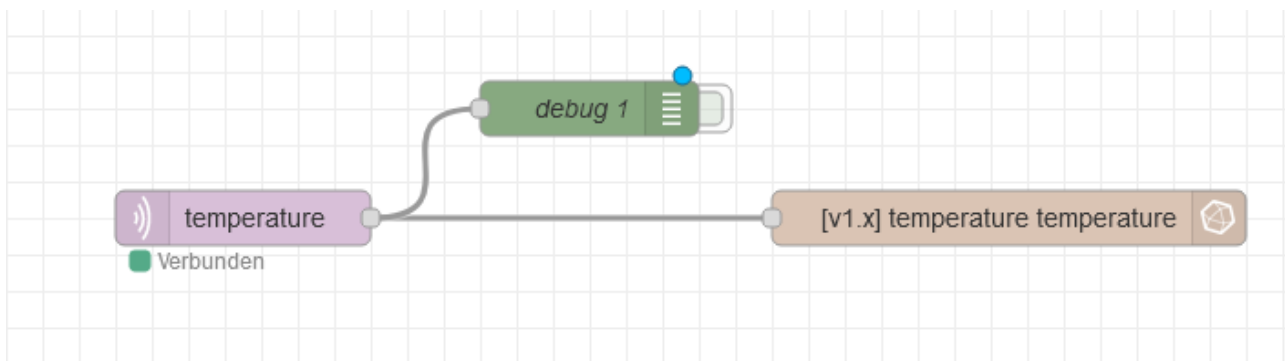
Die JSON-Node bleibt unverändert.

Danach sollten wir folgende Umgebung vorfinden (MQTT Client sollte auf verbunden stehen):

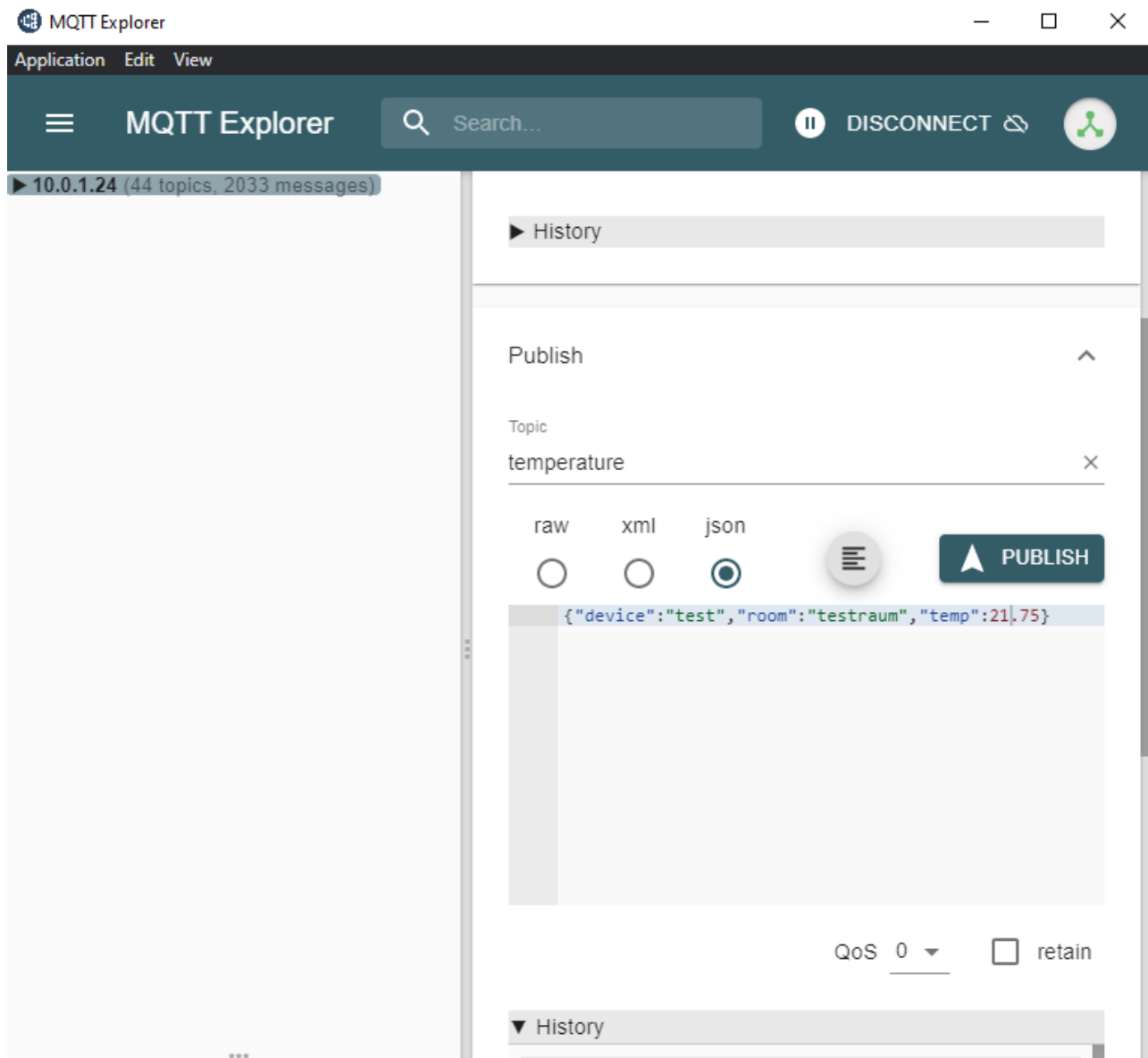


Debugging

Um herauszufinden ob unser Vorhaben vorab funktioniert, fügen wir 2 debug Notes hinzu:



Und senden ein Payload via mqtt Client an das Gerät:

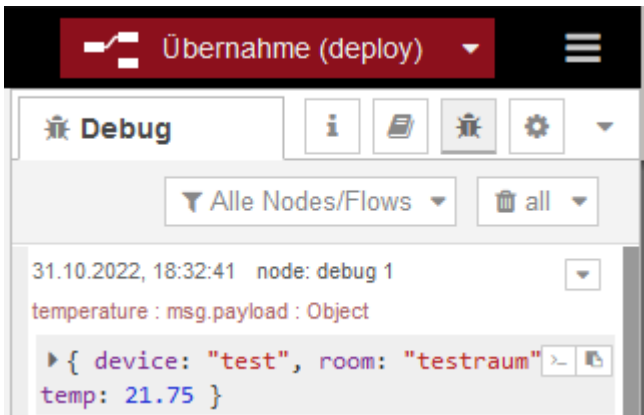


Der verwendete Client kann von der Website <http://mqtt-explorer.com/> heruntergeladen werden.

Json-Payload:

```
{"device": "test", "room": "testraum", "temp": 21.75}
```


Wenn alles funktioniert, sollte rechts bei Node-Red im Debug Reiter folgendes angezeigt werden:



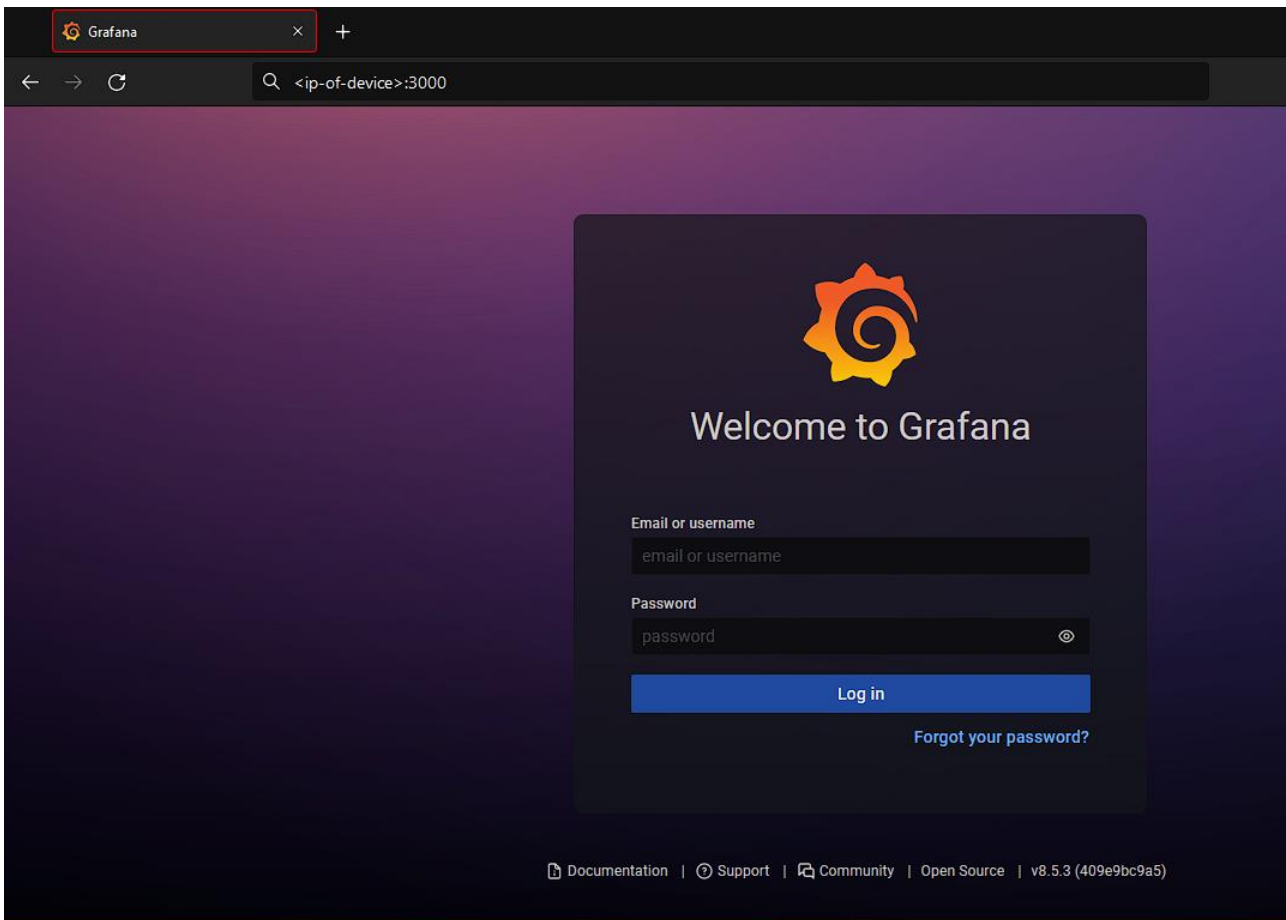
Und in InfluxDB:

```
admin@temperaturlogger:~ $ docker exec -it influxdb influx
Connected to http://localhost:8086 version 1.8.10
InfluxDB shell version: 1.8.10
> show databases
name: databases
name
----
temperature
> use temperature
Using database temperature
> select * from temperature
name: temperature
time                device room    temp
----
1654161522619856737 test   testraum 24.75
> _
```

Sind die Daten in influxdb angekommen, ist das Sammeln der Daten erfolgreich gewesen.

GRAFANA

Im Browser kann mit der Geräte IP (bzw. localhost) und dem Port 3000 auf die Website zur Konfiguration zugegriffen werden:



Default-Login:

Nutzername: admin

Passwort: admin

Konfiguration

Klicke auf `Data Sources` -> `Add data source` -> `InfluxDB` und trage folgende Einstellungen ein

Name: Temperatur Sensor

HTTP

URL: <http://influxdb:8086>

InfluxDB Details

Database: temperature

User: nodered

Password: nodered

Panel erstellen

The screenshot shows the InfluxDB query builder interface. At the top, the 'Data source' is set to 'Temperatur Sensoren'. The 'Query options' section shows 'MD = auto = 1275' and 'Interval = 15s'. Below this, the query is built using a table-like structure:

Clause	Value	Operator	Value
FROM	default		temperature
WHERE	device	=	Strawberry
SELECT	field(temperature)	mean()	+
GROUP BY	time(\$_interval)	fill(null)	+
TIMEZONE	(optional)	ORDER BY TIME	ascending
LIMIT	(optional)	SLIMIT	(optional)
FORMAT AS	Time series	ALIAS	Strawberry

In **FROM** wählen wir die Tabelle (unter Influx auch Measurement genannt) temperature aus. Da wir für jedes Gerät eine Messung benötigen, wählen wir bei **WHERE** das „Device“ z.B. „Strawberry“ aus.

Unter **SELECT** wählen wir die Temperatur, welchen wir auf der X-Achse (Zeit) als Y-Wert für den Verlauf anzeigen möchten.

Bei **GROUP BY**, können wir auch noch anzeigen lassen, wie die Zeitachse gefüllt wird, wenn kein Wert gegeben ist.

Pi-Kiosk

Zuerst wird die Abhängigkeit „unclutter“ installiert:

```
sudo apt install unclutter -y
```

Wir laden die Datei grafana-kiosk.linux.armv7 herunter und platzieren Sie unter /usr/bin als grafana-kiosk

<https://github.com/grafana/grafana-kiosk/releases>

```
sudo mv grafana-kiosk.linux.armv7 grafana-kiosk
```

```
sudo cp grafana-kiosk /usr/bin/
```

```
sudo chmod 755 /usr/bin/grafana-kiosk
```

Damit beim Starten des Gerätes Grafana auf dem Bildschirm erscheint müssen wir eine Autostart-Datei anlegen:

```
nano /home/admin/.config/autostart/grafana-kiosk.desktop
```

```
[Desktop Entry]
Type=Application
Exec=/usr/bin/grafana-kiosk -URL=http://localhost:3000 -login-method=local -username=admin -password=Lagrafa -kiosk-mode=tv -ignore-certificate-errors -lxde
```

Zum Testen - Kioskmode im Betrieb ausführen:

```
/usr/bin/grafana-kiosk --URL http://localhost:3000 --login-method local --user admin --password admin --kiosk-mode tv
```

Automatisierung: Einhängen und Backups

Anlegen des db-csv.sh scripts

```
nano /home/admin/scripts/db-csv.sh
```

```
docker exec -it influxdb influx -database 'temperature' -execute 'SELECT \
* FROM 'temperature' -format csv > Auswertung.csv
mv /mnt/raid/influxdb/databases/temperature /mnt/raid/Gradlog
```

Anlegen der Datei 99-usbhook.rules

```
nano /etc/udev/rules.d/99-usbhook.rules
```

```
ACTION=="add",KERNEL=="sd*", SUBSYSTEMS=="usb", ATTRS{product}=="*",
RUN+="/mnt/raid/Gradlog/usbhook.sh %k"
```

Anlegen des usbhook.sh Scripts

```
#!/bin/bash

# Konfiguration
LOG_FILE=/var/log/usbhook
exec > >(tee -a ${LOG_FILE} )
exec 2> >(tee -a ${LOG_FILE} >&2)
SRC="/mnt/raid/Gradlog"

DEVICE="$1" # the device name
NUMBER="${DEVICE: -1}"

if [ "$NUMBER" = "1" ]
then
    exit 0
fi

mkdir /tmp/"$DEVICE"

# NTFS usb drive
mount -t ntfs-3g /dev/"$DEVICE"1 /tmp/"$DEVICE"
cp -R "$SRC"/* /tmp/"$DEVICE"
chmod 777 -R /tmp/"$DEVICE"

umount /tmp/"$DEVICE"
rm -r /tmp/"$DEVICE"
echo "Copy to $DEVICE >> Fertig !"
export DISPLAY=:0
sudo -u admin notify-send "Copy to $DEVICE >> Fertig !" -t 5000

exit 0
```

Neustarten der udev-Regeln

```
sudo udevadm control --reload-rules
```

Sicherheit

Standardbenutzer sperren

Der Nutzer „pi“ stellt ein Sicherheitsrisiko dar, wird jedoch von einigen Scripts und Programmen vorausgesetzt. Aus diesem Grund sperren wir das Konto, um das Einloggen zu verhindern.

```
sudo passwd --lock pi
```

Nutzername und Passwort für Node-RED anlegen (optional)

Es wird ein Passworthash benötigt:

```
docker exec nodered node -e \  
"console.log(require('bcryptjs').hashSync(process.argv[1], 8));" \  
PASSWORD
```

Kopieren Sie den erhaltenen Hash und fügen Sie diesen in der settings.js ein.

```
adminAuth: {  
  type: "credentials",  
  users: [  
    {  
      username: "admin",  
      password: "PASSWORD",  
      permissions: "*"   
    }  
  ]  
}
```


Bedienung

Backup

USB-Stick mit **NTFS**-Formatierung an das USB-Kabel anschließen.
Das Backup (*wöchentlich und einmal täglich um 6:30 Uhr*) wird automatisch nach einstecken des Datenträgers übertragen.

Eichung der Sensoren

Ändern Sie den Wert Abweichung in der Datei Temperaturlogger.ino.

```
39 //Geräteeinstellungen
40 String Name = "Debug Geraet";
41 String Raum = "Debug Raum";
42 float Abweichung = 0.0;
43
```

Bei dem Wert handelt es sich um den Datentyp „float“ welcher eine Fließkommazahl darstellt. Achten Sie darauf, dass in der englischen Schreibweise ein Punkt anstelle eines Kommas für die Nachkommastellen verwendet wird.

Erweiterung

Wenn Sie das System mit einem weiteren ESP8266-Microcontroller erweitern wollen, folgen Sie der hier beschriebenen Kurzanleitung.

Arduino Image

Arduino Konfigurieren

Dieser Bereich im Quellcode kann vom Endnutzer verändert werden, wenn ein neues Gerät installiert werden soll.

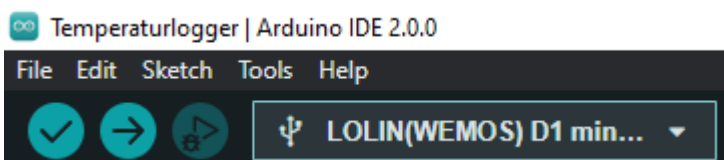
```

38
39 //Geräteeinstellungen
40 String Name = "Debug Geraet";
41 String Raum = "Debug Raum";
42 float Abweichung = 0.0;
43
44 // WiFi-Zugriff Einstellungen
45 const char* ssid = "tempLOG";
46 const char* wifi_password = "Wasserwaage";
47
48 // MQTT-Zugriff Einstellungen
49 const char* mqtt_server = "10.0.1.1";
50 const char* topic = "temperature";
51
52 const char* mqtt_username = "";
53 const char* mqtt_password = "";
54 const char* clientID = "";
55
56 //Ender der Benutzereinstellungen
57
58

```

Hochladen

Mit dem rechtszeigenden Pfeil wird der Quellcode, kompiliert auf das Gerät geladen.



Debugging mit Node-Red

Mit der `msg.payload` Funktion können die empfangenen MQTT Daten angezeigt werden.



Problemsuche

Initialisierungsprobleme:

Boot Probleme

ROOT ACCOUNT LOCKED!

Fehlermeldung:

```
"Cannot open access to console, the root account is locked."
```

In diesem Fall müssen wir die SD-Karte an einen anderen Computer anschließen und bei der Datei cmdline.txt die Zeile init=/bin/bash löschen.

Das Gerät bootet nicht

Prüfen Sie die Stromzufuhr. Ist das Gerät angeschlossen? Blinken die LEDs am Raspberry Pi? Ist das Netzteil des Raspberry Pi's vom Hersteller oder liefert mindestens 2A bei 5V? Wurde die SD-Karte richtig eingelegt oder ist diese falsch formatiert worden?

Probleme mit Raspberry Pi

Es werden keine Dateien auf den USB-Stick übertragen

Prüfen Sie zuerst den USB-Stick auf einen Defekt und schließen Sie ihn gegeben falls an einen anderen Computer an. Wurde das Speichermedium mit NTFS formatiert?

GRAFANA reagiert nicht mehr oder hat sich geschlossen

Öffnen Sie die Verknüpfung „Klick mich“ auf dem Desktop per Touchscreen Anweisung, wenn die Applikation geschlossen wurde. Sollte nichts mehr helfen, können Sie das Gerät kurz vom Strom nehmen. Nach einem Neustart sollte alles wieder funktionieren.

Alle Docker Container herunterfahren/starten

```
docker-compose down
```

```
docker-compose up -d
```

Unbekannte oder falsche Einträge in der Datenbank

Ein Microcontroller wurde falsch formatiert, bzw. gibt falsche Werte weiter. Prüfen Sie das zuletzt formatierte Gerät bzw. ihren derzeitigen Quellcode und vergleichen Sie ihn mit dem Original.

Probleme mit dem Microcontroller

DEBUG am Temperatur-Display

Wifi:

O = Verbunden

X = Verbindung ist abgebrochen / Schlechtes Signal

Bei längerem Ausfall:

- *(Befinden sich Störelemente in Unmittelbarer Umgebung von Sender oder Empfänger?)*
- *(Sollten Sie die Verbindungsinformationen geändert haben, müssen diese auch beim Microcontroller im Skript abgeändert werden)*

MQTT:

O = Verbunden

X = Keine Verbindung mit MQTT-Broker

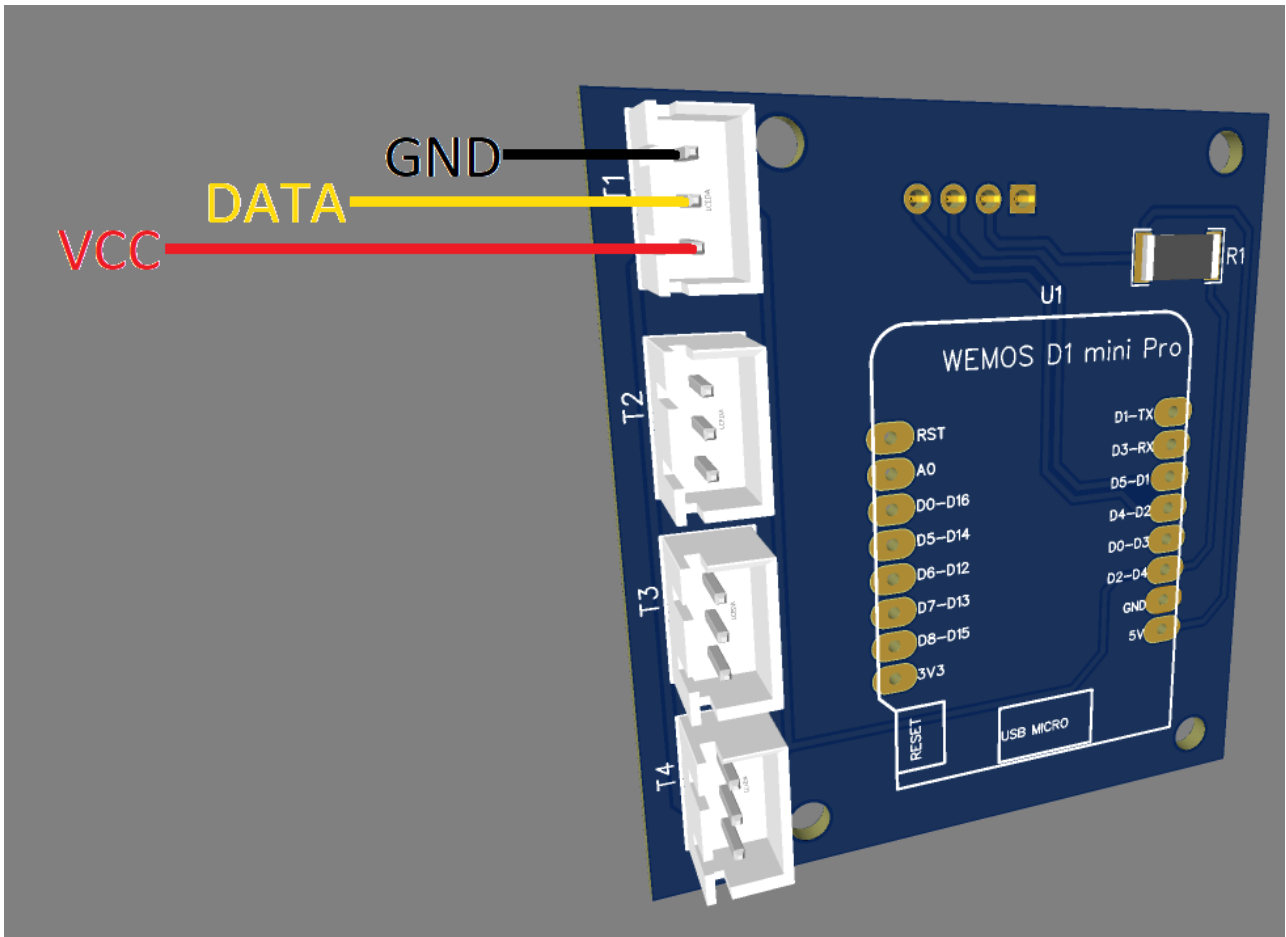
Bei längerem Ausfall:

- *(Läuft der MQTT-Broker am Raspberry Pi? Wurde das Gerät zuletzt vom Netz genommen und wartet auf eine Verbindung?)*
- *(Wurde zuletzt der Port des MQTT-Broker verändert?)*

Temperatur: -127.0

Das Kabel des Temperatursensors hatte sich gelöst oder der Sensor ist defekt.

Vergewissern Sie sich das die einzelnen Kabelstränge wie folgt angeschlossen sind und Kontakt mit der den Pins besitzen.



Abweichendes Anschließen kann das Gerät und den Sensor beschädigen!

Display funktioniert nicht

Überprüfen Sie zuerst die Stromzufuhr des Geräts. Ist das Gerät angeschlossen? Leuchtet die Blaue LED am Controller (periodisch) auf (entfernen Sie das Gehäuse)?

Sitzt das Kabel am Display locker oder ist sogar gelöst?

Sieht das Display beschädigt aus?

Zugangsdaten

Raspberry Pi - 10.0.1.10

Benutzername	admin
Passwort	PASSWD

WLAN (n / 2,4 Ghz) - 10.0.1.11

Passwort	PASSWD
----------	--------

SSID	SSID
WPA2	PASSWD

Webzugänge (IP in den Adressbereich des Browsers eingeben)

Nur im lokalen und AP-Netzwerk des Raspberry Pi's möglich

Raspberry Pi: 10.0.1.10

NodeRED: 10.0.1.10:1880

Grafana: 10.0.1.10:3000

Benutzername	admin
Passwort	PASSWD

TP-LINK Access-Point: 10.0.1.11

Notizen

[illegible]

[illegible]