

Neuinstallation ioBroker auf einem R-Pi 3 B+

Quellen:

- [Raspberry Pi OS und ggf. Image Writer](#)
- [Raspi-Config Kommandozeilen Optionen](#)
- [ioBroker](#)
- [InfluxDB Installation](#)
- [InfluxDB 1.x Installation](#)
- [Grafana Installation](#)
- [Grafana ohne Login](#)
- [Verbindung InfluxDB 2.0 <-> Grafana](#)
- [Jeelink Klon](#)
- [LaCrosse Temperatur- & Feuchtesensor TX29DTH-IT, 868 MHz](#)
- [FHEM Wiki zu JeeLink/Lacrosse Klon im Eigenbau](#)
- [Arduino Sketch für JeeLink Empfänger](#)
- [Verdrahtung Arduino Nano und 868 MHz RF-Modul](#)

Das Grundsystems

Installation

Die Verwendung von InfluxDB 2.x setzt ein 64-bit Betriebssystem voraus. Ein Desktop bzw. eine vollständige grafische Oberfläche wird nicht benötigt. Daher bezeichnet man ein solches System als „Headless“. Somit basiert dieses Heimautomatisierungsprojekt auf einem Raspberry Pi OS (32 oder 64-bit) Lite.

```
# Raspberry Pi Image besorgen
#wget
https://downloads.raspberrypi.org/raspios_lite_armhf/images/raspios_lite_armhf-2023-02-22/2023-02-21-raspios-bullseye-armhf-lite.img.xz
wget
https://downloads.raspberrypi.com/raspios_lite_armhf/images/raspios_lite_armhf-2023-10-10/2023-10-10-raspios-bookworm-armhf-lite.img.xz
oder
#wget
https://downloads.raspberrypi.org/raspios_lite_arm64/images/raspios_lite_arm64-2023-02-22/2023-02-21-raspios-bullseye-arm64-lite.img.xz
wget
https://downloads.raspberrypi.com/raspios_lite_arm64/images/raspios_lite_arm64-2023-10-10/2023-10-10-raspios-bookworm-arm64-lite.img.xz
# R-Pi Imager herunterladen und intallieren:
wget https://downloads.raspberrypi.org/imager/imager_latest_amd64.deb
sudo dpkg -i imager_latest_amd64.deb
rpi-imager
#sudo touch /media/<benutzername>/bootfs/ssh
```

```
#sync
# SD-Karte aushängen und damit den R-Pi booten
```

Konfiguration

Während des Bootvorganges erhält der Raspberry eine IP Adresse vom Router. Diese kann über die entsprechende Weboberfläche des Routers ermittelt werden.

```
# Grundsystem auf den aktuellen Stand bringen:
ssh -l pi <IP>
sudo su
apt-get update && sudo apt-get upgrade
apt-get install aptitude mc apt-transport-https software-properties-common
raspi-config nonint do_hostname iobroker
raspi-config nonint do_owire 0
raspi-config nonint do_change_locale de_DE.UTF-8 UTF-8
raspi-config nonint do_expand_rootfs
raspi-config nonint do_update
reboot
# Feste IP für eth0 einstellen:
mcedit /etc/dhcpd.conf
interface eth0
static ip_address=192.168.10.xx/24
static routers=192.168.10.1
static domain_name_servers=192.168.10.1
```

InfluxDB 1.x

Installation

```
sudo su
#wget -q0- https://repos.influxdata.com/influxdata-archive_compat.key | sudo
apt-key add -
#source /etc/lsb-release
#echo "deb https://repos.influxdata.com/${DISTRIB_ID,,} ${DISTRIB_CODENAME}
stable" | sudo tee /etc/apt/sources.list.d/influxdb.list
curl https://repos.influxdata.com/influxdata-archive.key | gpg --dearmor |
sudo tee /usr/share/keyrings/influxdb-archive-keyring.gpg >/dev/null
echo "deb [signed-by=/usr/share/keyrings/influxdb-archive-keyring.gpg]
https://repos.influxdata.com/debian $(lsb_release -cs) stable" | sudo tee
/etc/apt/sources.list.d/influxdb.list
apt update
apt install influxdb
systemctl unmask influxdb
systemctl enable influxdb
systemctl start influxdb
```

```
systemctl status influxdb
```

Konfiguration

```
sudo su
influx
> CREATE USER "admin" WITH PASSWORD 'influxdbadmin' WITH ALL PRIVILEGES
> CREATE USER "iobroker" WITH PASSWORD 'iobroker'
> CREATE DATABASE "iobrokerdb"
> GRANT ALL ON "iobrokerdb" TO "iobroker"
> exit
#
mcedit /etc/influxdb/influxdb.conf
[http]
  enabled = true
  bind-address = ":8086"
  auth-enabled = true
  log-enabled = true
  write-tracing = false
  pprof-enabled = false
  https-enabled = false
#
systemctl restart influxdb
```

Datenbankgröße ermitteln

```
sudo su
du -sh /var/lib/influxdb/data/iobrokerdb/
```

Backups

Quelle:

https://docs.influxdata.com/influxdb/v1/administration/backup_and_restore/#back-up-all-databases

Erstellen

```
sudo su
influxd backup -portable /path/to/backup-destination
```

Wieder herstellen

```
sudo su
influxd restore -portable /path/to/backup-destination
```

InfluxDB 2.x

Installation

```
sudo su
cd ~
wget -q https://repos.influxdata.com/influxdata-archive_compat.key
echo '393e8779c89ac8d958f81f942f9ad7fb82a25e133faddaf92e15b16e6ac9ce4c
influxdata-archive_compat.key' | sha256sum -c && cat influxdata-
archive_compat.key | gpg --dearmor | sudo tee
/etc/apt/trusted.gpg.d/influxdata-archive_compat.gpg > /dev/null
echo 'deb [signed-by=/etc/apt/trusted.gpg.d/influxdata-archive_compat.gpg]
https://repos.influxdata.com/debian stable main' | sudo tee
/etc/apt/sources.list.d/influxdata.list
apt-get update
apt-get install influxdb2
systemctl start influxdb
systemctl status influxdb
```

Konfiguration

Die Konfiguration der InfluxDB erfolgt über den Browser via <http://RASPI-IP:8086>.

Grafana

Installation

```
sudo su
wget -q -O /usr/share/keyrings/grafana.key https://apt.grafana.com/gpg.key
echo "deb [signed-by=/usr/share/keyrings/grafana.key]
https://apt.grafana.com stable main" | sudo tee -a
/etc/apt/sources.list.d/grafana.list
apt-get update
# Install the latest OSS release:
apt-get install grafana
# Start System Service aka. Server
systemctl daemon-reload
systemctl start grafana-server
systemctl status grafana-server
systemctl enable grafana-server.service
```

Grafana Server: <http://IP:3000>

Grafana ohne Login

```
mcedit /etc/grafana/grafana.ini
# [auth.anonymous]
enabled = true
systemctl restart grafana-server
```

Backup

Quelle: <http://cactusprojects.com/backup-restore-grafana/>

Erstellen

```
sudo su
cp /var/lib/grafana/grafana.db /path/to/backup-destination
cp /etc/grafana/grafana.ini /path/to/backup-destination
```

Wieder herstellen

```
sudo su
cd /path/to/backup-destination
cp grafana.db /var/lib/grafana/
cp grafana.ini /etc/grafana/
```

ioBroker

Installation

```
sudo su
# automatische Installation:
curl -sLf https://iobroker.net/install.sh | bash -
```

Nach der Installation ist die ioBroker Instanz erreichbar unter <http://<Raspberry-Pi-IP>:8081>.

Adapter installieren

Als Adapter werden alle Schnittstellen zu externen aber auch zu internen Datenquellen (z.B. Shelly Sensoren, Temperatur, Luftfeuchte, GPIO, 1-Wire, etc.) und Senken (Shelly Aktoren) bezeichnet.

- influxDB (Datenbank zur Speicherung von Messwerten)
- DS18B20 (Auslesen von 1-Wire Temperatursensoren)
- HABpanel (Steuerung von Aktoren vis Webseite)

- Jeelink-Geräte (Empfangen von 868mHz LaCrosse Temperatur- und Feuchtesensoren (hier z.B. TX29DTH-IT) mit Hilfe eines Jeelink Klones
- MQTT Broker (zur Datenverarbeitung von MQTT-fähigen Sensoren und Aktoren)
- Philips Luftreiniger (für die Anbindung eines Philips Air Purifier 4000i Series (Typ: AC4236/10)
- RPI-Monitor (Nutzung von GPIOs z.B. zum Einlesen eines Reed-Kontaktes für den Gaszähler)
- Skriptausführung (Javascript und Blockly)
- WEB-Server (für HABpanel)

admin ★★★★★ Admin Installierte Instanzen: 1 Verfügbare Version: 6.3.5 Installierte Version: 6.3.5	backupup ★★★★★ BackItUp Installierte Instanzen: 1 Verfügbare Version: 2.5.12 Installierte Version: 2.5.12	influxdb ★★★★★ Daten mit InfluxDB protokollieren Installierte Instanzen: 1 Verfügbare Version: 3.2.0 Installierte Version: 3.2.0	ds18b20 ★★★★★ DS18B20 Installierte Instanzen: 1 Verfügbare Version: 1.6.1 Installierte Version: 1.6.1	simple-api ★★★★★ Einfache RESTful API Verfügbare Version: 2.7.2 Installierte Version: 2.7.2
discovery ★★★★★ Gerätesuche Installierte Instanzen: 1 Verfügbare Version: 3.1.0 Installierte Version: 3.1.0	habpanel ★★★★★ HABpanel Installierte Instanzen: 1 Verfügbare Version: 0.5.0 Installierte Version: 0.5.0	jeelink ★★★★★ Jeelink-Geräte Installierte Instanzen: 1 Verfügbare Version: 0.1.4 Installierte Version: 0.1.4	mqtt ★★★★★ MQTT Broker/Client Installierte Instanzen: 1 Verfügbare Version: 4.0.7 Installierte Version: 4.0.7	philips-air ★★★★★ Philips Luftreiniger Installierte Instanzen: 1 Verfügbare Version: 1.0.1 Installierte Version: 1.0.1
rpi2 ★★★★★ RPI-Monitor Installierte Instanzen: 1 Verfügbare Version: 1.3.2 Installierte Version: 1.3.2	javascript ★★★★★ Skriptausführung Installierte Instanzen: 1 Verfügbare Version: 6.1.4 Installierte Version: 6.1.4	socketio ★★★★★ socket.io Verfügbare Version: 4.2.0 Installierte Version: 4.2.0	ws ★★★★★ Web socket Verfügbare Version: 1.3.0 Installierte Version: 1.3.0	web ★★★★★ WEB-Server Installierte Instanzen: 1 Verfügbare Version: 4.3.0 Installierte Version: 4.3.0

Adapter konfigurieren

Die oben aufgeführten Adapter sind nach der Installation unter Instanzen zu finden und müssen dort konfiguriert werden.

1-Wire

Instanzeinstellungen: ds18b20.0 v2.0.4										
ALLGEMEINE EINSTELLUNGEN SENSOREN REMOTE-SYSTEME										
Sensoren										
+ Adresse	Remote-System-ID	Name	Abfrageintervall	Einheit	Faktor	Offset	Dezimalstellen	Null bei Fehler	Aktiviert	
10-00080359c712	keins	Vorlauftemperatur	Standard	°C	1	0	2	☒	☒	↓
10-00080359e7fe	keins	Rücklauftemperatur	Standard	°C	1	0	2	☒	☒	↑

DS18B20

TBC

JeeLink

TBC - Empfänger bauen und programmieren → TBC

MQTT

TBC - für Shelly und IKEA Sensoren -> angepasste Firmware dokumentieren

Philips

TBC

GPIO für Gaszähler

TBC

Blockly

TBC

InfluxDB

TBC

Update und Bugfixing

```
iobroker stop  
iobroker update  
iobroker fix  
# oder curl -sL https://iobroker.net/fix.sh | bash -  
iobroker upgrade self  
iobroker start
```

Last update:
2023/10/24
18:47

wiki:projekte:iobroker:uebersicht <https://www.von-thuelen.de/doku.php/wiki/projekte/iobroker/uebersicht?rev=1698173242>

From:

<https://www.von-thuelen.de/> - **Christophs DokuWiki**

Permanent link:

<https://www.von-thuelen.de/doku.php/wiki/projekte/iobroker/uebersicht?rev=1698173242>



Last update: **2023/10/24 18:47**