

# Beschreibung der Installation für Stick IMST871 und Raspberry PI zum Auslesen Kampstrup Multical 21 Wasserzähler

LINUX-Befehle, die im Terminalfenster (putty) eingegeben wurden, sind blau und kursiv gekennzeichnet: *ls -la*

Am Schluss soll als Ergebnis so etwas herauskommen:

Zählerstand Wasser		
Zähler-Seriennummer:		
Wasserstand absolut:	117,708	m3
Vormonatsendwert:	112,309	m3
Durchschnittsverbrauch pro Tag:	0,1965	m3
Jahresanfangswert Januar:	46,405	m3
Zählerstand Dezember:	147,6775	m3
Jahresverbrauch:	101,272	m3
Wassertemperatur extern:	15	°C
Wassertemperatur intern:	12	°C
Status:	NaN	
Uhrzeit letzte Ablesung:	2021 07 18 18:24:04	



## Vorausbedingung:

1. Beschaffung Raspberry PI mind. 4 GB oder 8 GB (Modell 4 Rev. B)
2. Beschaffung Micro-SD Card mind. 16 GB oder höher
3. Beschaffung IMST iM871A-USB Stick (zu kaufen z.B. hier: <https://www.tekmodul.de/produkt/im871a-usb-drahtloser-m-bus-adapter/> oder hier <https://shop.imst.de/wireless-modules/usb-radio-products/10/im871a-usb-wireless-m-bus-usb-adapter-868-mhz>)



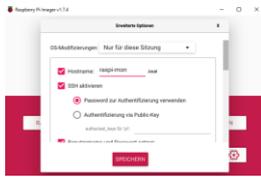
4. Information vom Wasserversorger – er stellt eine XML-Datei oder eine Information aus, in dem der AES-Key zum Entschlüsseln der Datagramme und die ID des Wasserzählers enthalten ist. Diesen braucht man später. Ggfs. braucht der Wasserversorger eine Kostenübernahmeerklärung, da der Zähler bei Auszug getauscht werden muss (die Ableseverschlüsselung ist ja offengelegt einer dann nicht mehr beteiligten 3. Partei).

## Installationsdurchführung

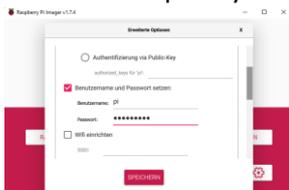
### Raspberry PI mit Betriebssystem versorgen

1. Download des Imager für Windows-PCs: [https://downloads.raspberrypi.org/imager/imager\\_latest.exe](https://downloads.raspberrypi.org/imager/imager_latest.exe)
2. den Imager installieren und starten  
Raspberry PI OS (other) -> Raspberry Pi OS (64-bit)  
OS: RASPBERRY PI OS (64-BIT) auswählen  
SD-Card: die eingesteckte SD-Karte auswählen – Achtung: es werden alle darauf befindlichen Daten gelöscht, da diese neu formatiert wird!

3. Zahnradsymbol für erweiterte Optionen aufrufen
4. Hostname vorgeben
5. SSH aktivieren



6. Name und Passwort setzen:  
Benutzername: pi  
Passwort: raspberry



7. Ggfs. Wifi-Zugangsdaten eingeben & alles Speichern
8. Dann auf Karte schreiben – **Achtung es werden alle Daten auf der SD-Card überschrieben!!!!**

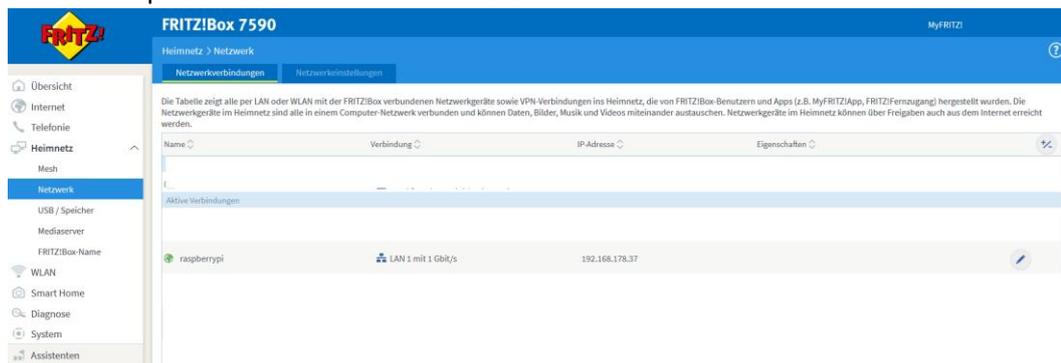


9. SD-Karte herausnehmen
10. SD-Karte in den Raspberry Pi einlegen
11. Netzkabel am Raspberry Pi und Netzwerk/Wifi anschließen (ggfs. direkt an der FritzBox oder dem Internet-Router oder internem Switch anschließen/ggfs. Wifi-Netzwerk konfigurieren)
12. Stromversorgung am Raspberry Pi anschließen (die LEDs gehen an und das System bootet)

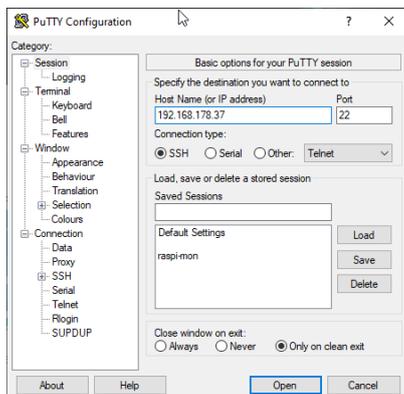
## Grundkonfiguration des Raspberry-Pi

1. Windows-Tools installieren  
putty downloaden (ggfs. mal nachschauen, ob es neuere Versionen gibt  
<https://www.chiark.greenend.org.uk/~sgtatham/putty/latest.html>  
<https://the.earth.li/~sgtatham/putty/latest/w64/putty-64bit-0.78-installer.msi>)
2. putty installieren aus dem msi-Paket

3. Raspberry Pi prüfen, ob SD-Karte eingelegt ist  
wenn ja, dann Stromversorgung herstellen und Netzkabel anschließen – ggfs. direkt an FritzBox oder andere Router (WLAN-Konfiguration kann dann später durchgeführt werden)
4. Jetzt in der FritzBox nachschauen, welche IP-Adresse der Raspberry Pi bekommen hat.
  1. Browser auf dem Windows-PC öffnen
  2. Eingeben <https://fritz.box> (oder andere IP-Adresse zum Management des Routers, meist 192.168.1.1 oder andere – im Handbuch zum Router nachschauen!)  
hier Achtung: der Fehler, dass das Zertifikat ungültig ist, kann ignoriert werden!
  3. Dann in die lokalen Netzwerksetting nachschauen, welche IP-Adresse der Raspberry Pi über DHCP bekommen hat (das Gerät meldet sich mit raspberrypi):
  4. In dem Beispiel ist das die IP-Adresse 192.168.178.37

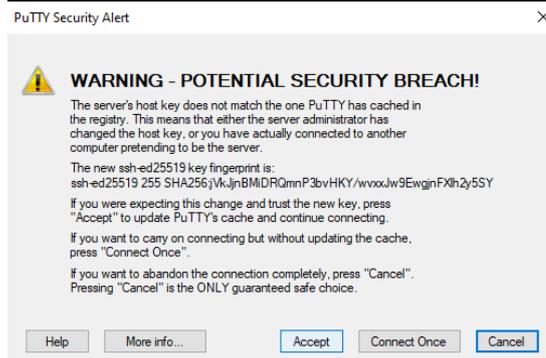


5. Putty auf dem Windows-PC öffnen
6. Folgende Eingaben im Reiter „Session“ vornehmen (hier dann die IP-Adresse eintragen, die oben vom Router vergeben wurde):



7. Dann die Verbindung öffnen mit Klick auf Open

## Die erscheinende Warnmeldung mit Accept bestätigen



Als Login Namen eingeben: pi  
Als Passwort eingeben: raspberry

8. Raspi updaten  
[`sudo apt-get update`](#) <ENTER>  
[`sudo apt-get upgrade`](#) <ENTER>
9. Neu starten  
[`sudo reboot`](#) <ENTER>

## Konfiguration des Rasp-Pi für den Einsatz als Wasserzähler für den Kamstrup Multical 21

1. Putty aufrufen (siehe Beschreibung im vorigen Auswahlmenü Punkt 4. Bis Punkt 7.)
2. Download der wbusmeters-Dateien  
[`wget https://github.com/weetmuts/wbusmeters/archive/refs/heads/master.zip`](#) <ENTER>
3. Nach Eingabe des Befehls `ls -la` <ENTER> müsste die Ausgabe so aussehen

```
pi@raspberrypi:~$ wget https://github.com/weetmuts/wbusmeters/archive/refs/heads/master.zip
--2021-07-18 12:54:27-- https://github.com/weetmuts/wbusmeters/archive/refs/heads/master.zip
Resolving github.com (github.com)... 140.82.121.3
Connecting to github.com (github.com)[140.82.121.3]:443... connected.
HTTP request sent, awaiting response... 302 Found
Location: https://codeload.github.com/weetmuts/wbusmeters/zip/refs/heads/master [following]
--2021-07-18 12:54:28-- https://codeload.github.com/weetmuts/wbusmeters/zip/refs/heads/master
Resolving codeload.github.com (codeload.github.com)... 140.82.121.10
Connecting to codeload.github.com (codeload.github.com)[140.82.121.10]:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: unspecified [application/zip]
Saving to: 'master.zip'

master.zip
[ <-> ]
2021-07-18 12:54:28 (5.85 MB/s) - 'master.zip' saved [583242]

pi@raspberrypi:~$ ls -la
total 608
drwxr-xr-x 5 pi pi 4096 Jul 18 12:54 .
drwxr-xr-x 3 root root 4096 May 7 15:42 ..
-rw-r--r-- 1 pi pi 220 May 7 15:42 .bash_logout
-rw-r--r-- 1 pi pi 3523 May 7 15:42 .bashrc
drwxr-xr-x 2 pi pi 4096 May 7 15:52 Bookshelf
drwx----- 3 pi pi 4096 Jul 18 11:17 .gnupg
drwxr-xr-x 3 pi pi 4096 May 7 15:52 .local
-rw-r--r-- 1 pi pi 583242 Jul 18 12:54 master.zip
-rw-r--r-- 1 pi pi 807 May 7 15:42 .profile
-rw-r--r-- 1 pi pi 209 Jul 18 12:54 .wget-hsts
pi@raspberrypi:~$
```

4. Entpacken der Dateien mit dem Befehl: `unzip master.zip` <ENTER>  
Nach der Eingabe des Befehls `ls -la` <ENTER> müsste das Verzeichnis wbusmeters-master erzeugt worden sein

```
pi@raspberrypi:~$ ls -la
total 612
drwxr-xr-x 6 pi pi 4096 Jul 18 12:56 .
drwxr-xr-x 3 root root 4096 May 7 15:42 ..
-rw-r--r-- 1 pi pi 220 May 7 15:42 .bash_logout
-rw-r--r-- 1 pi pi 3523 May 7 15:42 .bashrc
drwxr-xr-x 2 pi pi 4096 May 7 15:52 Bookshelf
drwx----- 3 pi pi 4096 Jul 18 11:17 .gnupg
drwxr-xr-x 3 pi pi 4096 May 7 15:52 .local
-rw-r--r-- 1 pi pi 583242 Jul 18 12:54 master.zip
-rw-r--r-- 1 pi pi 807 May 7 15:42 .profile
-rw-r--r-- 1 pi pi 209 Jul 18 12:54 .wget-hsts
drwxr-xr-x 15 pi pi 4096 Jul 15 09:06 wbusmeters-master
pi@raspberrypi:~$
```

5. In das Unterverzeichnis wechseln: `cd wbusmeters-master/ <ENTER>`
6. Eine Library nachinstallieren: `sudo apt install librtlsdr-dev <ENTER>`  
Die darauffolgende Abfrage mit „y“ bestätigen.
7. Eine weitere Library installieren: `sudo apt install libncurses-dev <ENTER>`
8. Jetzt die Software konfigurieren mit dem Befehl: `./configure <ENTER>`  
(Achtung „./“ müssen mit eingegeben werden!)

```

pi@raspberrypi:~/wbusmeters-master $ ./configure
checking build system type... armv7l-unknown-linux-gnueabihf
checking host system type... armv7l-unknown-linux-gnueabihf
checking for g++... g++
checking whether the C++ compiler works... yes
checking for C++ compiler default output file name... a.out
checking for suffix of executables...
checking whether we are cross compiling... no
checking for suffix of object files... o
checking whether we are using the GNU C++ compiler... yes
checking whether g++ accepts -g... yes
checking for rtsdr get device count in -lrtlsdr... no
configure: error: Could not find rtsdr library. Try: sudo apt install librtlsdr-dev
pi@raspberrypi:~/wbusmeters-master $

```

9. Aufrufen des Makefiles mit: `sudo make DEBUG=true <ENTER>`  
Achtung: es laufen hier viele Meldungen durch und es dauert etwas, bis es fertig ist – Geduld!
10. Jetzt die Software/Treiber installieren mit: `sudo make install <ENTER>`
11. Nach Abschluss den daemon neu laden: `sudo systemctl daemon-reload <ENTER>`
12. Jetzt müssen die Konfigurationsdateien angelegt werden
  - a) Eingeben des Befehls: `sudo vi /etc/wbusmeters.conf <ENTER>`
  - b) Jetzt i (Insert-Taste drücken): `i`
  - c) Danach den folgenden Text eingeben (Voraussetzung es handelt sich um einen Wasserzähler, der mit „41“ = t1 anfängt)
 

```

loglevel=normal
# set USB-Stick for IMST 871A for a wmbus device and meter-type = 40; set it to c1.
# set USB-Stick for IMST 871A for a wmbus device and meter-type = 41; set it to t1.
device=/dev/ttyUSB0:im871a:t1
# But do not probe this serial tty.
donotprobe=/dev/ttyACM2
logtelegrams=true
format=json
meterfiles=/var/log/wbusmeters/meter_readings
meterfilesaction=overwrite
meterfilesnaming=name
meterfilestimestamp=day
logfile=/var/log/wbusmeters/wbusmeters.log
shell=/usr/local/bin/mosquitto_pub -h localhost -t wbusmeters/$METER_ID -m "$METER_JSON"
alarmshell=/usr/local/bin/mosquitto_pub -h localhost -t wbusmeters_alarm -m "$ALARM_TYPE $ALARM_MESSAGE"
alarmtimeout=1h
          
```

*alarmexpectedactivity=mon-sun(00-23)*

*ignoreduplicates=true*

Folgende Erklärungen zu den Übertragungsmodi des Wasserzählers – diese können hier ausgelesen werden:

<https://products.kamstrup.com/download.php?uid=515d4ab700278>

<http://products.kamstrup.com/ajax/downloadFile.php?uid=5bcecc825f706&display=1>

Wenn die Typbezeichnung mit „41“ anfängt, muss man t1 wählen (obiges Beispiel: *device=/dev/ttyUSB0:im871a:t1*)

Wenn die Typbezeichnung des Wasserzählers mit „40“ anfängt, muss man c1 wählen (obiges Beispiel: *device=/dev/ttyUSB0:im871a:c1*)

13. Nach Eingabe des Textes „ESC“ drücken: *<ESC>*

14. Datei abspeichern mit der Eingabe: *:wq <Enter>*

Achtung: der „.“ muss mit angegeben werden!

15. Verzeichnis anlegen

*sudo mkdir /var/log/wmbusmeters/meter\_readings <ENTER>*

*sudo chown wmbusmeters:wmbusmeters /var/log/wmbusmeters/meter\_readings <ENTER>*

16. Danach muss die Datei MyTapWater angelegt werden

*sudo vi /etc/wmbusmeters.d/MyTapWater <ENTER>*

danach einfügen des Textes durch Eingabe von „i“: *i*

*name=MyTapWater*

*id=<hier die ID des Wasserzählers eingeben>* (steht in der Mail vom Wasserversorger – siehe oben Punkt 4 der Vorausbedingung)

*key=<hier den Key eingeben>* (steht in der Mail vom Wasserversorger – siehe oben Punkt 4 der Vorausbedingung)

danach drücken von: *<ESC>*

abspeichern mit: *:wq <ENTER>*

Die Datei müsste so aussehen:

*name=MyTapWater*

*id=12345678*

*key=00112233445566778899AABBCCDDEEFF*

17. Das gesamte System einmal updaten:

*sudo apt-get update <ENTER>*

*sudo apt-get upgrade <ENTER>*

Achtung: das dauert etwas, zwischendurch kann eine Interaktion erscheinen und man muss z.B. „q“ drücken

18. Eine Library installieren: *sudo apt-get install libuv1 <ENTER>*

19. Den MQTT-Client installieren

Der mosquitto-Client wird benötigt, um die Datagramme später im ioBroker zu verarbeiten.

Eingeben des Befehls:

*cd ~ <ENTER>*

*wget https://mosquitto.org/files/source/mosquitto-2.0.15.tar.gz <ENTER>*

nach Eingabe des Befehls *ls -la >ENTER>* muss die Ausgabe so aussehen:

```

pi@raspi-moni:~$ ls -la
total 1482
drwxr-xr-x 15 pi pi 4096 Mar 5 10:17 .
drwxr-xr-x 3 root root 4096 Feb 21 04:20 ..
-rw-r--r-- 1 pi pi 279 Mar 5 09:47 .bash_history
-rw-r--r-- 1 pi pi 220 Feb 21 03:53 .bash_logout
-rw-r--r-- 1 pi pi 3523 Mar 5 09:49 .bashrc
drwxr-xr-x 2 pi pi 4096 Feb 21 04:20 Bookshelf
drwxr-xr-x 4 pi pi 4096 Feb 21 04:20 .cache
drwxr-xr-x 2 pi pi 4096 Feb 21 04:20 .config
drwxr-xr-x 2 pi pi 4096 Feb 21 04:20 Desktop
drwxr-xr-x 2 pi pi 4096 Feb 21 04:20 Documents
drwxr-xr-x 2 pi pi 4096 Feb 21 04:20 Downloads
drwxr-xr-x 1 pi pi 4096 Feb 21 04:20 .local
-rw-r--r-- 1 pi pi 840375 Mar 5 09:47 master.zip
-rw-r--r-- 1 pi pi 792632 Aug 16 2022 mosquitto-2.0.15.tar.gz
drwxr-xr-x 2 pi pi 4096 Feb 21 04:20 Music
drwxr-xr-x 1 pi pi 4096 Feb 21 04:20 Pictures
-rw-r--r-- 1 pi pi 807 Feb 21 03:53 .profile
drwxr-xr-x 2 pi pi 4096 Feb 21 04:20 Public
drwxr-xr-x 2 pi pi 4096 Feb 21 04:20 Templates
drwxr-xr-x 2 pi pi 4096 Feb 21 04:20 Videos
-rw-r--r-- 1 pi pi 247 Mar 5 10:17 .wget-hsts
drwxr-xr-x 17 pi pi 4096 Mar 5 09:51 .xsession-errors-master
-rw-r--r-- 1 pi pi 54 Mar 5 09:47 .Xauthority
-rw-r--r-- 1 pi pi 2625 Mar 5 09:47 .xsession-errors
-rw-r--r-- 1 pi pi 2625 Feb 21 04:20 .xsession-errors.old
pi@raspi-moni:~$

```

20. Jetzt die Software auspacken:

```
tar xzf mosquitto-2.0.15.tar.gz <ENTER>
```

nach Eingabe des Befehls `ls -la <ENTER>` müsste das Verzeichnis so aussehen:

```

pi@raspi-moni:~$ ls -la
total 1696
drwxr-xr-x 16 pi pi 4096 Mar 5 10:18 .
drwxr-xr-x 3 root root 4096 Feb 21 04:20 ..
-rw-r--r-- 1 pi pi 279 Mar 5 09:47 .bash_history
-rw-r--r-- 1 pi pi 220 Feb 21 03:53 .bash_logout
-rw-r--r-- 1 pi pi 3523 Mar 5 09:49 .bashrc
drwxr-xr-x 2 pi pi 4096 Feb 21 04:20 Bookshelf
drwxr-xr-x 4 pi pi 4096 Feb 21 04:20 .cache
drwxr-xr-x 5 pi pi 4096 Feb 21 04:20 .config
drwxr-xr-x 2 pi pi 4096 Feb 21 04:20 Desktop
drwxr-xr-x 2 pi pi 4096 Feb 21 04:20 Documents
drwxr-xr-x 2 pi pi 4096 Feb 21 04:20 Downloads
drwxr-xr-x 3 pi pi 4096 Feb 21 04:20 .local
-rw-r--r-- 1 pi pi 840375 Mar 5 09:47 master.zip
drwxr-xr-x 19 pi pi 4096 Aug 16 2022 mosquitto-2.0.15
-rw-r--r-- 1 pi pi 792632 Aug 16 2022 mosquitto-2.0.15.tar.gz
drwxr-xr-x 2 pi pi 4096 Feb 21 04:20 Music
drwxr-xr-x 1 pi pi 4096 Feb 21 04:20 Pictures
-rw-r--r-- 1 pi pi 807 Feb 21 03:53 .profile
drwxr-xr-x 2 pi pi 4096 Feb 21 04:20 Public
drwxr-xr-x 2 pi pi 4096 Feb 21 04:20 Templates
drwxr-xr-x 2 pi pi 4096 Feb 21 04:20 Videos
-rw-r--r-- 1 pi pi 247 Mar 5 10:17 .wget-hsts
drwxr-xr-x 17 pi pi 4096 Mar 5 09:51 .xsession-errors-master
-rw-r--r-- 1 pi pi 54 Mar 5 09:47 .Xauthority
-rw-r--r-- 1 pi pi 2625 Mar 5 09:47 .xsession-errors
-rw-r--r-- 1 pi pi 2625 Feb 21 04:20 .xsession-errors.old
pi@raspi-moni:~$

```

21. Libraries installieren

```
sudo apt install libssl-dev <ENTER>
```

```
sudo apt-get install libjson-dev <ENTER>
```

Rückfragen mit „y“ beantworten

22. In das Verzeichnis mosquitto wechseln und Software kompilieren und installieren

```
cd mosquitto-2.0.15/ <ENTER>
```

```
sudo make <ENTER>
```

```
sudo make install <ENTER>
```

Achtung: das dauert etwas

23. Mosquitto Dienst ggfs. deaktivieren und entfernen

```
sudo systemctl stop mosquitto <ENTER>
```

```
sudo update-rc.d mosquitto remove <ENTER>
```

```
sudo systemctl disable mosquitto <ENTER>
```

nach Eingabe des Kommandos

```
ps aux | grep mosquitto <ENTER>
```

muss die Ausgabe so aussehen:

```

pi@raspberrypi:~$ ps aux|grep mosquitto
pi 12397 0.0 0.0 7360 524 pts/0 S+ 16:39 0:00 grep --color=auto mosquitto
pi@raspberrypi:~$

```

24. Jetzt den Linker-Cache updaten

```
sudo /sbin/ldconfig <ENTER>
```

25. Aufrufen der journal-Protokollierung und beobachten, ob der Stick eingebunden wird

```
journalctl -fe <ENTER>
```

## 26. Jetzt den IMST iM871A USB-Stick am Raspberry Pi einstecken

Folgende Ausgabe müsste jetzt erscheinen:

```
Jul 19 14:32:07 raspberrypi kernel: usb 1-1.3: new full-speed USB device number 3 using xhci_hcd
Jul 19 14:32:07 raspberrypi kernel: usb 1-1.3: New USB device found, idVendor=10c4, idProduct=ea60, bcdDevice= 1.00
Jul 19 14:32:07 raspberrypi kernel: usb 1-1.3: New USB device strings: Mfr=1, Product=2, SerialNumber=3
Jul 19 14:32:07 raspberrypi kernel: usb 1-1.3: Product: WIMOD iM871A-usb
Jul 19 14:32:07 raspberrypi kernel: usb 1-1.3: Manufacturer: Silicon Labs
Jul 19 14:32:07 raspberrypi kernel: usb 1-1.3: SerialNumber: 00B93EA5
Jul 19 14:32:07 raspberrypi mtp-probe[9156]: checking bus 1, device 3: "/sys/devices/platform/sch/fd500000.pcie/pci0000:00/0000:00:00.0/0000:01:00.0/usb1/1-1/1-1.3"
Jul 19 14:32:07 raspberrypi mtp-probe[9156]: bus: 1, device: 3 was not an MTP device
Jul 19 14:32:07 raspberrypi kernel: usbcore: registered new interface driver usbserial_generic
Jul 19 14:32:07 raspberrypi kernel: usbserial: USB Serial support registered for generic
Jul 19 14:32:07 raspberrypi kernel: usbcore: registered new interface driver cp210x
Jul 19 14:32:07 raspberrypi kernel: usbserial: USB Serial support registered for cp210x
Jul 19 14:32:07 raspberrypi kernel: cp210x 1-1.3.1.0: cp210x converter detected
Jul 19 14:32:07 raspberrypi kernel: usb 1-1.3: cp210x converter now attached to ttyUSB0
Jul 19 14:32:07 raspberrypi mtp-probe[9162]: checking bus 1, device 3: "/sys/devices/platform/sch/fd500000.pcie/pci0000:00/0000:00:00.0/0000:01:00.0/usb1/1-1/1-1.3"
Jul 19 14:32:07 raspberrypi mtp-probe[9162]: bus: 1, device: 3 was not an MTP device
```

## 27. Abbrechen der Anzeige mittels <CTRL> <C>

## 28. den wmbusmeters-Dienst aktivieren

`sudo systemctl enable wmbusmeters <ENTER>`

`sudo systemctl daemon-reload <ENTER>`

## 29. prüfen, ob der Dienst korrekt läuft:

`systemctl status wmbusmeters <ENTER>`

```
pi@raspberrypi:~/wmbusmeters-master $ systemctl status wmbusmeters
● wmbusmeters.service - "wmbusmeters service"
   Loaded: loaded (/lib/systemd/system/wmbusmeters.service; enabled; vendor preset: enabled)
   Active: inactive (dead)
     Docs: https://github.com/weetmits/wmbusmeters
          man:wmbusmeters(1)
```

## 30. System neu starten

`sudo reboot<ENTER>`

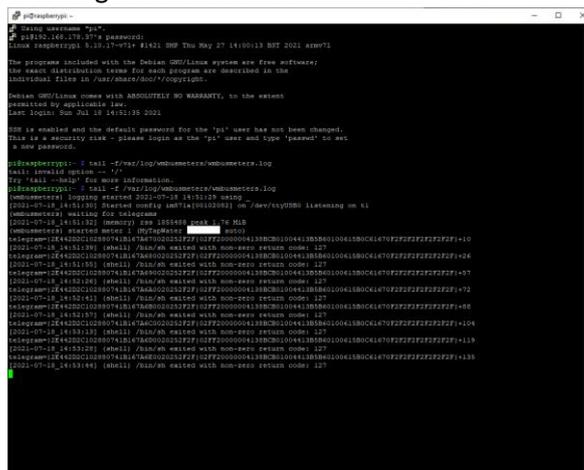
Achtung: damit wird die Putty-Sitzung beendet und es muss nach Neustart des Systems diese wieder neu hergestellt werden (siehe oben)!

## 31. Nach neuer Putty-Sitzung die Logfiles prüfen:

`tail -f /var/log/wmbusmeters/wmbusmeters.log <ENTER>`

die Ausgabe kann man mit <CTRL> <C> abbrechen

die Ausgabe sieht so aus:



Mit dem Befehl: `ls -la /var/log/wmbusmeters/meter_readings/ <ENTER>`

Werden in dem Unterverzeichnis jetzt Dateien angezeigt, die nach Tagen durchnummeriert sind und die aktuellen Daten enthalten.

Damit sollte jetzt der USB-Stick und die Erzeugung der Datenpakete erfolgreich sein.

## ioBroker Installation

Nun muss auf dem Raspberry Pi der ioBroker eingerichtet und konfiguriert werden.

Hinweis: die Befehle sind immer mit <Enter> abzuschließen – dieses wird nicht mehr mitgeschrieben!

1. ioBroker-Installation  
putty auf dem Windows-PC öffnen und mit dem Raspberry Pi verbinden  
Name: pi  
Passwort: raspberry
2. nun auf der Shell die Befehle eingeben (siehe Anleitung hierunter:  
<https://www.iobroker.net/#de/download>):

`cd ~ <ENTER>`

`curl -sLf https://iobroker.net/install.sh | bash - <ENTER>`

3. nun den Raspberry Pi einmal neu starten  
`sudo reboot <ENTER>`  
Achtung: dabei geht wieder die putty-Verbindung verloren!
4. Kurze Zeit warten, danach ist der Rasperry Pi ioBroker erreichbar  
Auf dem Windows PC einen Browser öffnen und die folgende IP-Adresse eingeben (abhängig davon, welche Adresse der Rasperry Pi bekommen hat.  
Wie oben beschrieben, wäre das in dem Beispiel die 192.168.178.37, der Admin-Port ist die 8081

<http://192.168.178.37:8081>



5. Assistent starten
6. Die entsprechenden Meldungen akzeptieren und den Info-Adapter installieren lassen  
Den Lizenzvertrag akzeptieren



## 7. Administratorkennwort festlegen

Name: admin

Passwort: raspberry+123

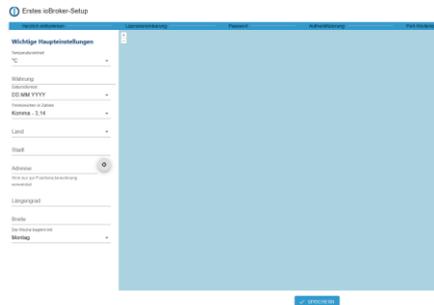
## 8. Kennwortprüfung und Zertifikate auswählen (hier entscheiden, ob die Settings gebraucht werden!)



## 9. Meldung zustimmen



## 10. Dann die Grundeinstellungen übernehmen



Auf **SPEICHERN** klicken

Danach **ABSCHLIESSEN** klicken

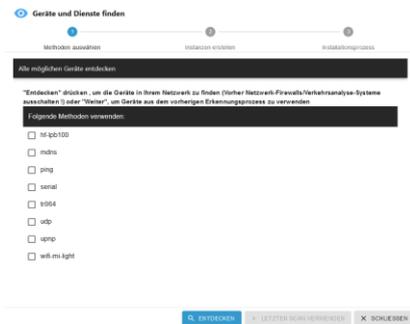
## 11. Die Methoden können alle abgewählt werden.

Diese Methoden werden per Default angezeigt – für den hier beschriebenen Anwendungsfall können sie alle abgewählt werden.



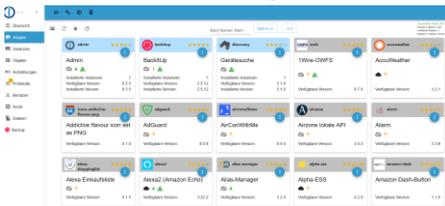
MQTT  mDNS  ping  serial  SSDP  UPnP  UPnP  web-ui.light

Abwählen:



und SCHLIESSEN klicken

Das ist dann die Anzeigeseite, die der ioBroker nach dem Konfigurationsschritt anzeigt:



## Adapter-Installation

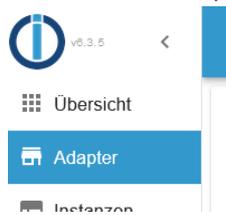
Nun müssen weitere Adapter aktiviert werden.

Folgende Adapter sind wichtig:

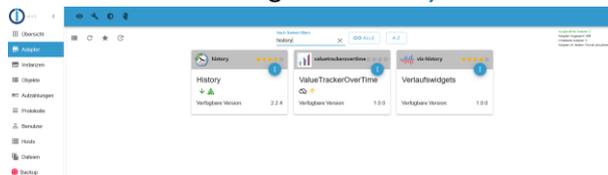
- History
- Visualisierung/Web
- MQTT
- JS Skriptausführung
- Log Parser

## History-Adapter installieren:

a) Auf die Seite Adapter gehen:



b) Oben im Filterfeld eingeben: *History*

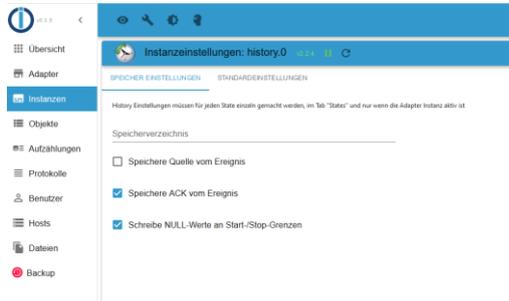


c) Den History-Adapter bei den  anklicken

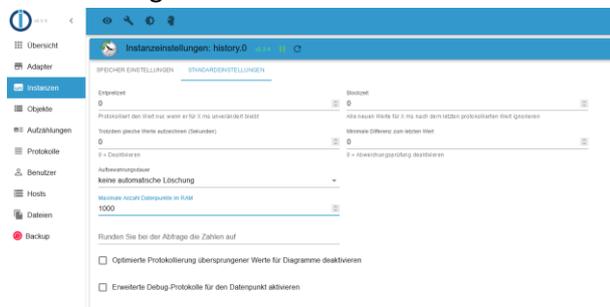
d) Dann auf das „+“-Zeichen den Adapter hinzufügen



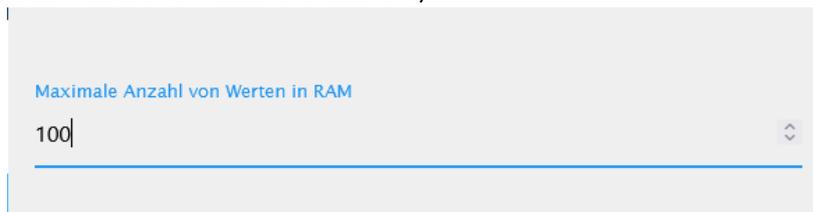
e) Danach öffnet sich die Einstellungen für den Adapter



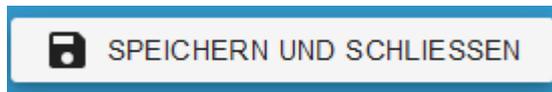
f) Jetzt die STANDARDEINSTELLUNGEN FÜR ZUSTÄNDE prüfen durch anklicken des Menüeintrags



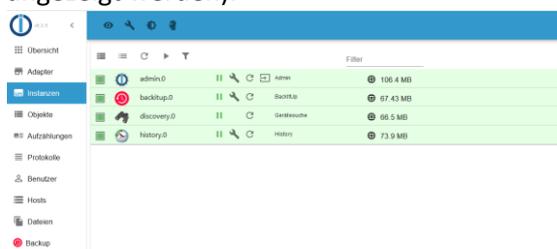
Hier können Einstellungen gemacht werden – z.B. den Wert für „Maximale Anzahl von Werten in RAM“ kann herabgesetzt werden, wenn der Raspberry Pi z.B. relativ wenig RAM-Speicher hat, kann er auf 100 verringert werden. Weiterhin sollte man alle Werte sammeln, so kann man später auf alle Zahlen zugreifen (abhängig von der Größe der Micro-SD-Karte).



g) Danach auf SPEICHERN UND SCHLIESSEN klicken



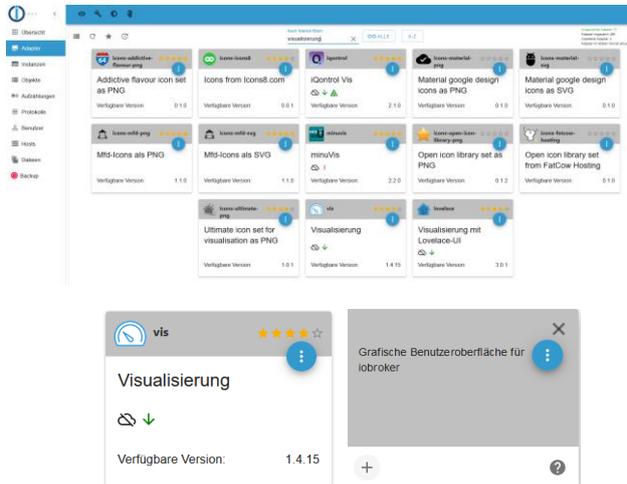
h) Das Ergebnis wird hier angezeigt (der Adapter ist aktiv, wenn die grünen 2 Striche angezeigt werden):



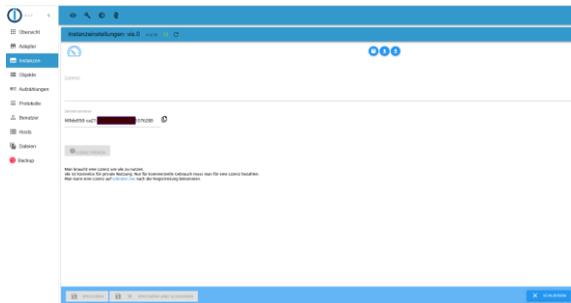
## Web-Adapter installieren:

Im Adapter-Menü im Filter-Feld eingeben: *visualisierung*

Es werden mehrere Adapter angezeigt, es wird der Adapter mit dem Namen Visualisierung benötigt. Diesen, wie oben beschrieben, installieren.

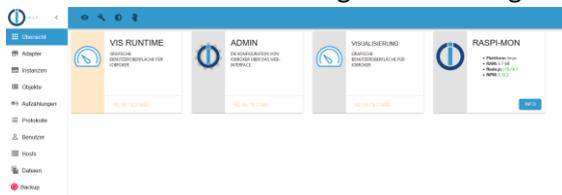


Achtung: die Installation dauert etwas, da einige Pakete nachgeladen und installiert werden müssen. Die Installation endet mit dem Bild:



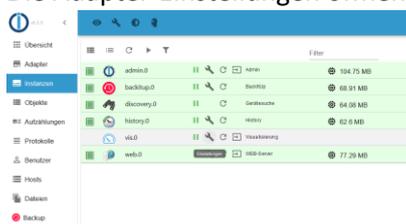
SCHLIESSEN anklicken

Übersicht die aktuelle Konfiguration anzeigen



Bei iobroker.net eine Lizenz für den VIS-Adapter beantragen

1. Die Adapter-Einstellungen öffnen über den Mausschlüssel (= Einstellungen)



## 2. Den Link öffnen (blau hervorgestellter Text)

Man braucht eine Lizenz um vis zu nutzen.

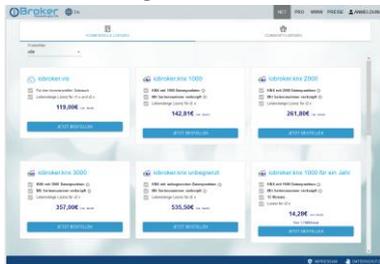
vis ist kostenlos für private Nutzung. Nur für kommerzielle Gebrauch muss man für eine Lizenz bezahlen.

Man kann eine Lizenz auf [iobroker.net](http://iobroker.net) nach der Registrierung bekommen.

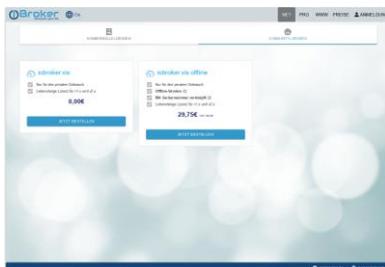
## 3. Im iobroker.net anmelden



## 4. Auf PREISE gehen



## 5. Umstellen auf COMMUNITY-LIZENZEN



## 6. Iobroker.vis für 0 € auswählen und JETZT BESTELLEN klicken

## 7. Jetzt einen Account erstellen und dann einloggen

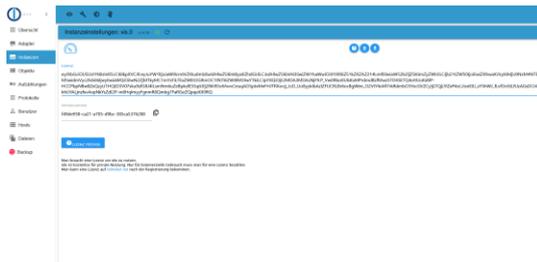
## 8. Die Lizenzen sind unter dem Menüpunkt „LIZENZEN“ hinterlegt – hier auf das Auge klicken

## 9. Dann im aufgehenden Fenster den Button „IN DIE ZWISCHENABLAGE KOPIEREN“ klicken

## 10. SCHLIESSEN klicken

## 11. Zurück auf die Webseite mit der eigenen iobroker-Installation gehen

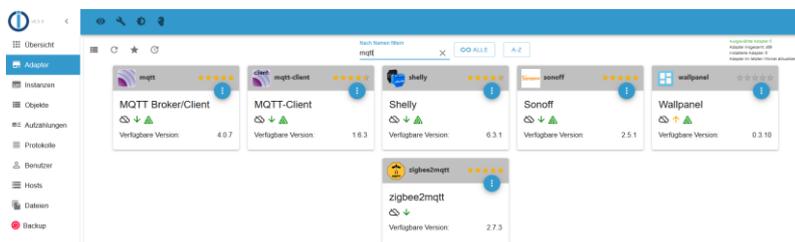
## 12. Dann unter INSTANZEN VIS -> Lizenzfeld anklicken und die Lizenz hineinkopieren (im Feld -> rechte Maustaste -> Einfügen drücken)



13. Dann LIZENZ PRÜFEN klicken -> sollte ein grünes OK kommen.
14. SPEICHERN UND SCHLIESSEN klicken

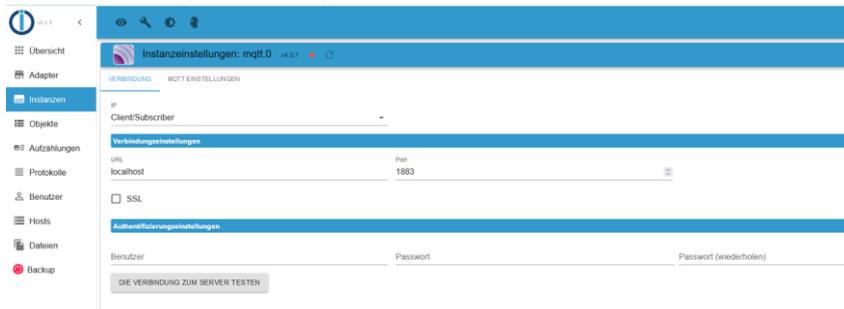
## MQTT-Adapter installieren:

Es wird der Adapter: MQTT Broker/Client benötigt.

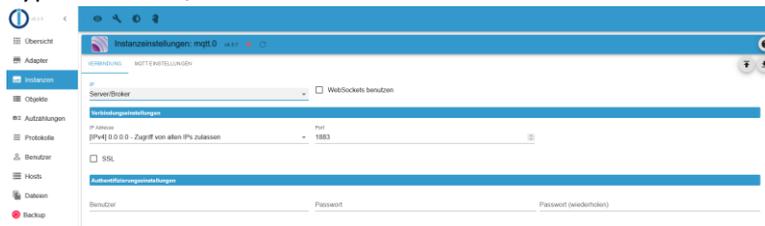


Mit Auswählen des Adapters und „+“ den Adapter installieren

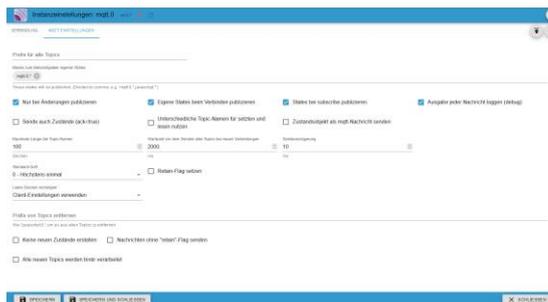
Dann die Konfiguration anpassen:



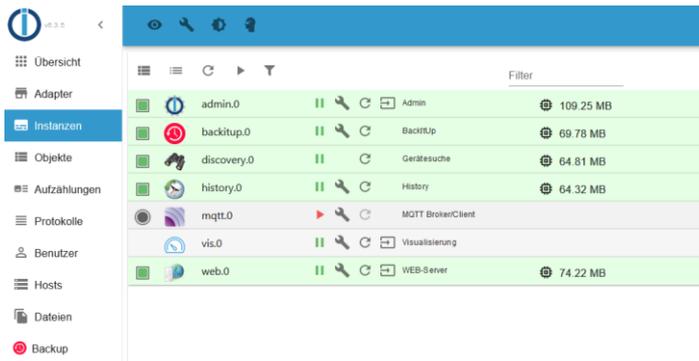
1. Typ: auf Server/Broker setzen



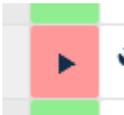
2. MQTT EINSTELLUNGEN abändern



3. danach beenden mit SPEICHERN UND SCHLIESSEN  
es wird dann folgende Grafik angezeigt:



Das besagt, dass der MQTT-Adapter aktuell noch nicht läuft, weil das Start-Symbol rot ist

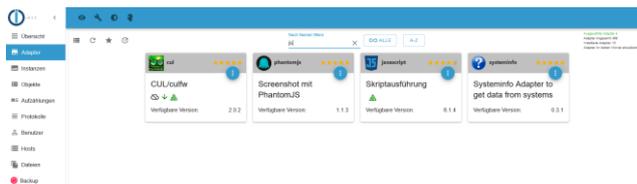


#### 4. MQTT-Broker starten

Hier den roten Play-Button am Adapter drücken – er ändert den Zustand in grün

### **JavaScript (JS)-Adapter installieren:**

Im Filterfeld der Adapter eingeben: `js`



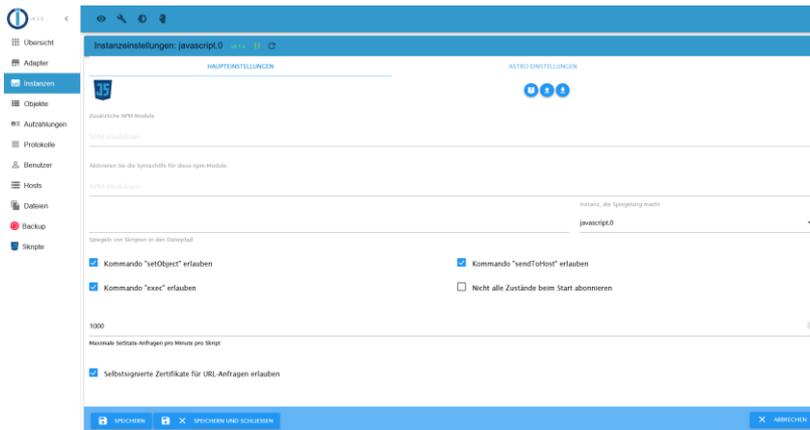
Benötigt wird der Adapter:



Diesen, wie oben beschrieben, installieren.

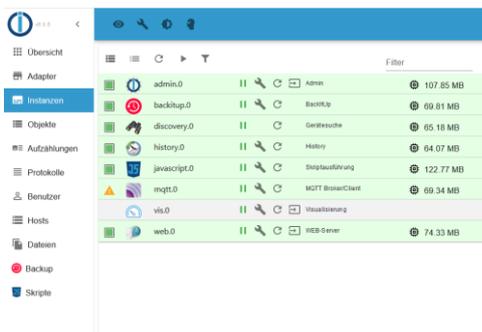
Achtung: das dauert etwas!

Danach wird das Fenster aktiv – hier alle Optionen aktivieren, bis auf „Nicht alle Zustände beim Start abonnieren“:

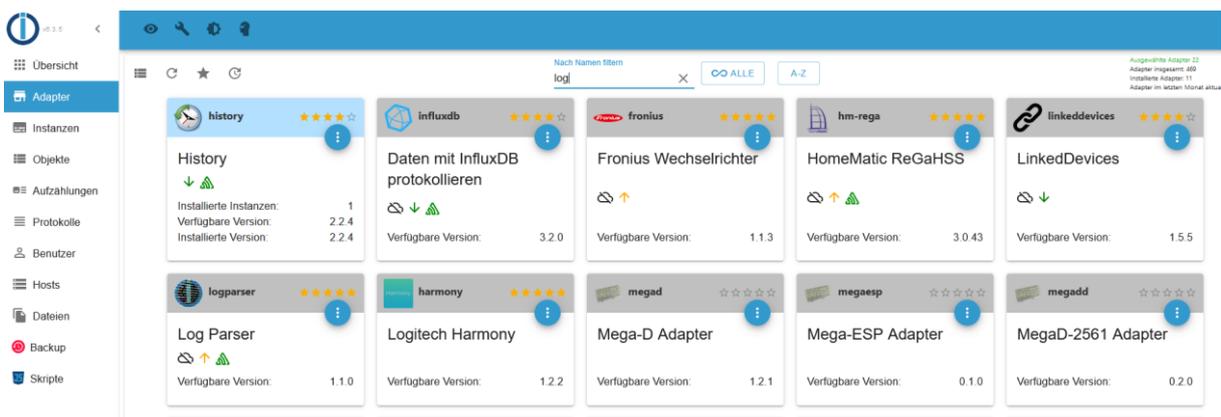


Danach **SPEICHERN UND SCHLIESSEN** das Fenster beenden.

Die Webseite noch einmal neu laden - Nun gibt es einen neuen Menüeintrag <> Skripte

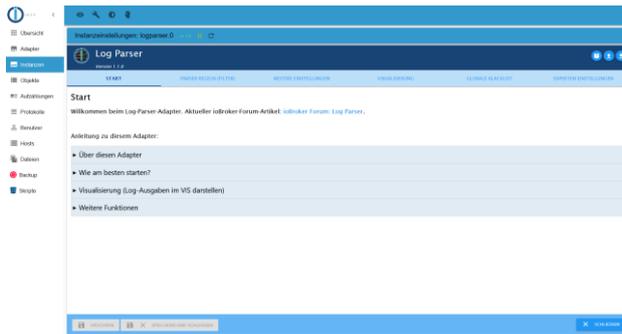


### Log Parser Adapter installieren:



Hier den Log Parser auswählen und installieren.





Mit SCHLIESSEN das Fenster beenden.

## Skripte anpassen

### Skript extract\_watermeters-values.js

Das Skript erzeugt die States im

Auf dem Windows-PC muss per notepad oder anderem Editor (z.B. notepad++) das folgende Script angepasst werden. Besonders wichtig ist hierbei wieder die ID des Wasserzählers, die vom Wasserwerk zugeschickt wurde (siehe oben).

Die Datei extract\_watermeters-values.js erzeugt die Initialwerte zum Anlegen im ioBroker.

Vorgehen:

1. diese Datei herunterladen
2. den Wert in der Datei (hier vorbelegt 12345678 in der Zeile:

```
const varSourceJSON = 'mqtt.0.wmbusmeters.12345678'/*wmbusmeters bei 12345678 bitte die richtige ID einsetzen!/*;
```

) durch die eigene ID des Wasserzählers ersetzen und die so geänderte Datei auf dem Windows-PC abspeichern.

3. den Wert in der Datei (hier vorbelegt mit 45 in der Zeile 20:

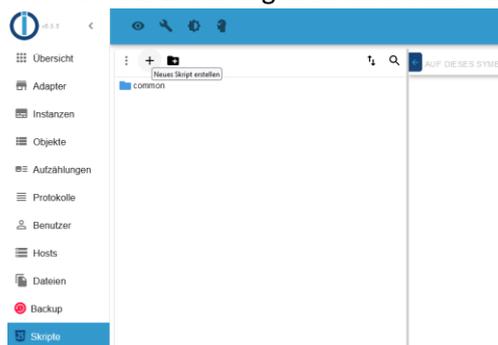
```
20 let january_value = 45; // start value in january
```

) durch den Wert ersetzen, der im Januar am eigenen Zähler gemessen wurde. Dieser Wert ist der Startwert für die Berechnung des prognostizierten Jahresgesamtwertes. In den Folgejahren wird dieser Wert automatisch neu gesetzt. Die Software ermittelt diesen Wert zu dem Zeitpunkt, an dem die ersten Werte des Folgejahres für Januar neu eingelesen werden und setzt diesen automatisch neu. Dieser Wert muss dann nur einmal angegeben werden, wenn die Software unterjährig neu eingespielt und gestartet wird oder wenn im Laufe des Jahres die Erhebung abstürzt und das System neu bootet.

4. Danach das Skript im ioBroker neu anlegen

Im Menü <> Skripte auswählen.

Auf das Plus-Zeichen gehen und damit ein neues Skript anlegen:

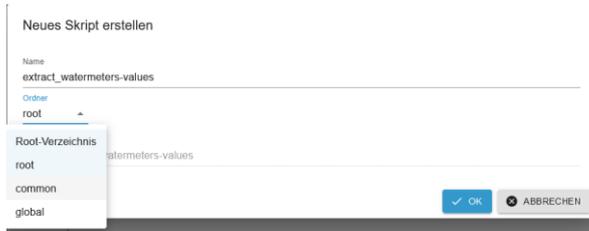


5. Dann Javascript auswählen

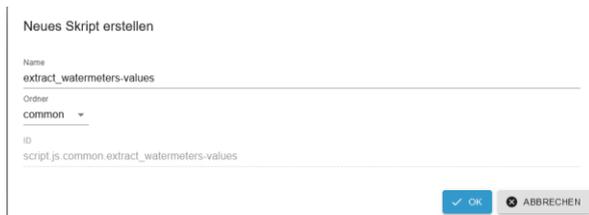


Und HINZUFÜGEN klicken

6. Den Namen extract\_watermeters-values eingeben

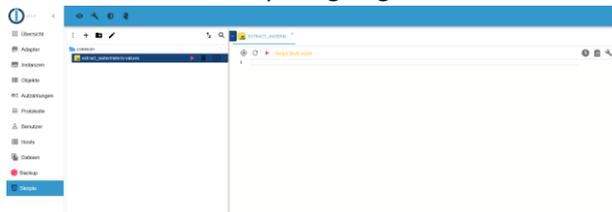


7. Ordner auf common ändern

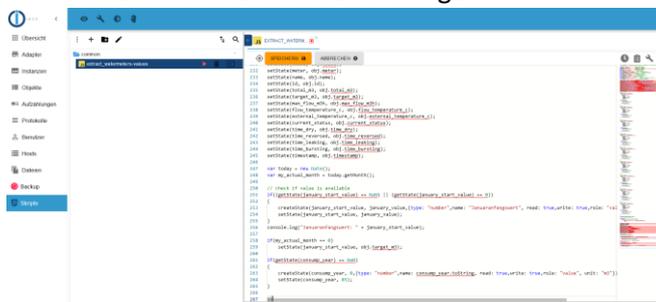


8. Dann OK drücken

9. Es wird ein leeres Script angelegt.



10. Jetzt den kompletten Inhalt des angepassten Skriptes auswählen und kopieren und bei 1 auf der rechten Seite des Fensters einfügen.

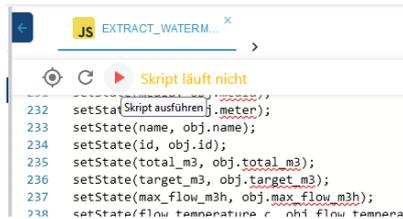


11. Dann Speichern drücken.



## 12. Script ausführen

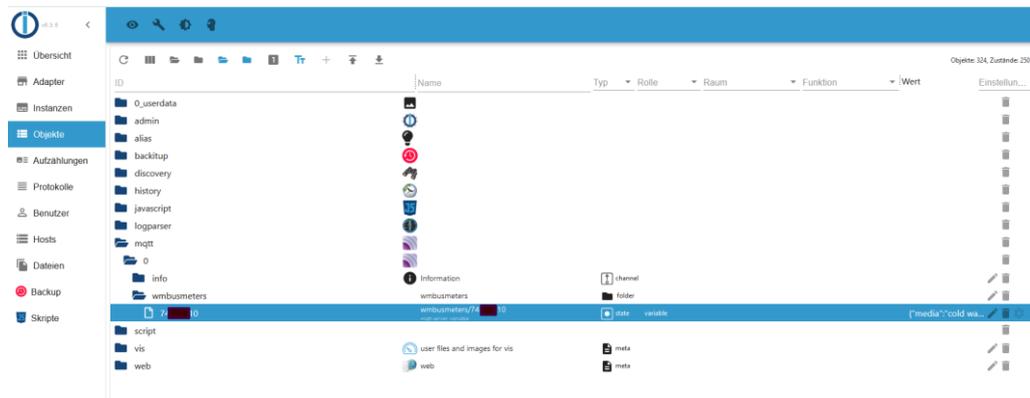
Dabei auf das rote Pfeilsymbol klicken.



```
JS EXTRACT_WATERM...
Skript läuft nicht
232 setState(Skript ausführen.j...meter);
233 setState(name, obj.name);
234 setState(id, obj.id);
235 setState(total_m3, obj.total_m3);
236 setState(target_m3, obj.target_m3);
237 setState(max_flow_m3h, obj.max_flow_m3h);
238 setState(flow temperature, obj.flow temperat
```

Jetzt prüfen, ob ein Objekt angelegt wurde:

Unter Objekte -> mqtt.0 -> wmbusmeters -> gibt es einen entsprechenden Untereintrag, der besagt, dass das Objekt korrekt angelegt wurde.



### Skript calculate\_average\_new.js

Das Skript wird jeden Tag einmal in der Nacht ausgeführt, um die möglichen hochgerechneten Verbrauchswerte bis zum Ende des Jahres zu berechnen. Dabei berechnet das Skript aus den historisierten Verbrauchswerten den Tagesmittelwert und berechnet mit dem aktuellen Zählerstand als Offset den vermeintlichen Jahresendwert.

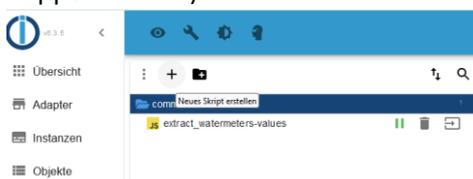
Auf dem Windows-PC muss per notepad oder anderem Editor (z.B. notepad++) das folgende Script angepasst werden.

Vorgehen:

1. die Datei calculate\_average\_new.js herunterladen
2. Danach das Skript im ioBroker neu anlegen

Im Menü <> Skripte auswählen.

Auf das Plus-Zeichen gehen und damit ein neues Skript anlegen (den Ordner common doppelt klicken):



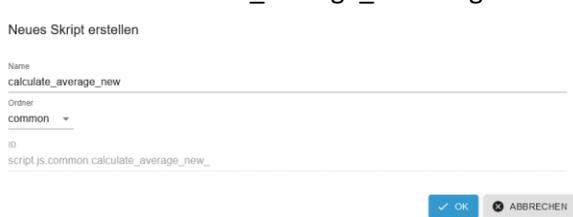
3. dann Javascript auswählen und HINZUFÜGEN klicken

Neues Skript hinzufügen



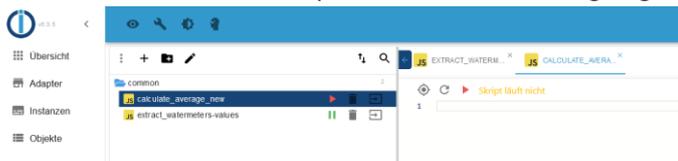
ABBRECHEN

4. Den Namen `calculate_average_new` eingeben

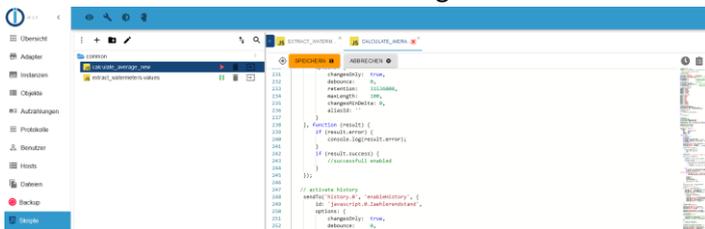


5. Ordner ist auf `common` bereits voreingestellt – das so belassen

6. OK klicken – das neue Skript ist als leere Hülle angelegt.



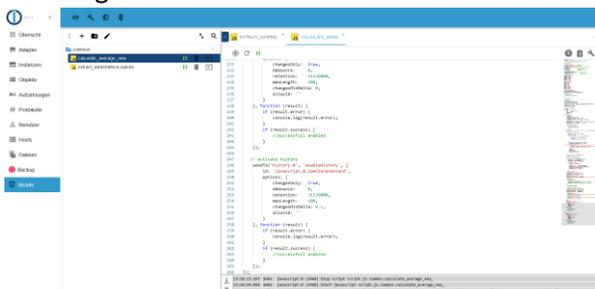
7. Jetzt den kompletten Inhalt des angepaßten Skriptes auswählen und kopieren und bei 1 auf der rechten Seite des Fensters einfügen.



8. **SPEICHERN** klicken

9. Danach auf **starten** klicken

10. Die Ausgabe beachten – hier sollte kein Fehler auftreten (grau eingefärbte Zeilen)



**Kontrolle der Skript-Ausführung:**

Mittels des Log-Adapters kann man prüfen, ob die Skripte korrekt laufen. Die hellbraun eingefärbten Werte werden aktuell noch nicht gesetzt, sondern erst in der Nacht um 01:00 Uhr, wenn das Skript calculate\_average\_new.js das erste Mal durchgelaufen ist.

Quelle	Zeit	debug	Nachricht
mgmt.0	2023-03-05 13:37:29.309	info	Client [] connection closed: disconnected
mgmt.0	2023-03-05 13:37:29.304	info	Client [] connected with secret 1678019640203_8450
admin.0	2023-03-05 13:37:08.364	info	← Disconnected system user/admin from 193.192.168.179:22 javascript
mgmt.0	2023-03-05 13:37:04.253	info	Client [] connection closed: disconnected
mgmt.0	2023-03-05 13:37:04.253	info	Client [] connected with secret 16780196024251_6725
mgmt.0	2023-03-05 12:36:48.211	info	Client [] connection closed: disconnected
mgmt.0	2023-03-05 12:36:48.205	info	Client [] connected with secret 1678019608204_6630
javascript.0	2023-03-05 12:36:35.058	info	script.js common: calculate_average_new.js: registere=0 subscriptions, 1 schedule, 0 messages, 0 logs and 0 file subscriptions
javascript.0	2023-03-05 12:36:34.950	info	Start javascript script.js common: calculate_average_new.js
mgmt.0	2023-03-05 12:36:32.171	info	Client [] connection closed: disconnected
mgmt.0	2023-03-05 12:36:32.166	info	Client [] connected with secret 1678019792164_4811
mgmt.0	2023-03-05 12:36:16.140	info	Client [] connection closed: disconnected
mgmt.0	2023-03-05 12:36:16.134	info	Client [] connected with secret 1678019778133_3550
javascript.0	2023-03-05 12:36:13.197	info	Stop script script.js common: calculate_average_new.js
mgmt.0	2023-03-05 12:36:08.113	info	Client [] connection closed: disconnected
mgmt.0	2023-03-05 12:36:08.108	info	Client [] connected with secret 1678019763197_3009
mgmt.0	2023-03-05 12:35:44.100	info	Client [] connection closed: disconnected
mgmt.0	2023-03-05 12:35:44.094	info	Client [] connected with secret 1678019744293_3343
mgmt.0	2023-03-05 12:35:28.000	info	Client [] connection closed: disconnected
mgmt.0	2023-03-05 12:35:28.006	info	Client [] connected with secret 1678019729285_4013
mgmt.0	2023-03-05 12:35:12.091	info	Client [] connection closed: disconnected
mgmt.0	2023-03-05 12:35:12.086	info	Client [] connected with secret 1678019712085_6177
mgmt.0	2023-03-05 12:34:56.100	info	Client [] connection closed: disconnected
mgmt.0	2023-03-05 12:34:56.094	info	Client [] connected with secret 1678019696993_3012
mgmt.0	2023-03-05 12:34:40.004	info	Client [] connection closed: disconnected

Vorher schauen die Variablen so aus:

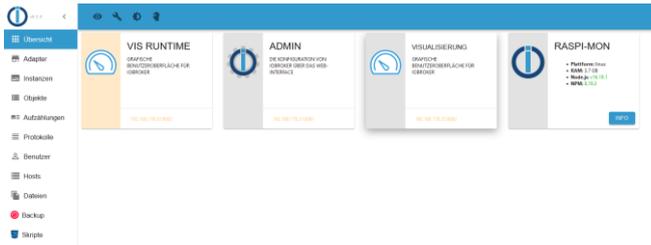
ID	Name	Typ	Rolle	Raum	Funktion	Wert	Einstellen...
0_userdata	admin						
0_userdata	alias						
0_userdata	backup						
0_userdata	discovery						
0_userdata	history						
0_userdata	javascript						
0_userdata	debug	Debug variables	channel				
0_userdata	scriptEnabled	Useful variables	channel				
0_userdata	scriptProblem	Useful variables	state				
0_userdata	variables	Useful variables	state				
0_userdata	Gesamtverbrauch	Useful variables	state			304.96 m3	

## Webseite aufbauen, um die Werte zu visualisieren:

Ziel ist es, dass die Anzeige so gestaltet ist:

Zählerstand Wasser		
Zähler-Seriennummer:		
Wasserstand absolut:	117,708	m3
Vormonatenswert:	112,309	m3
Durchschnittsverbrauch pro Tag:	0,1965	m3
Jahresanfangswert Januar:	46,405	m3
Zählerstand Dezember:	147,6775	m3
Jahresverbrauch:	101,272	m3
Wassertemperatur extern:	15	°C
Wassertemperatur intern:	12	°C
Status:	NaN	
Uhrzeit letzte Ablesung:	2021 07 18 18:24:04	

Man gelangt zur Konfiguration der Webseite über das Menü „Übersicht“ und dort auf die „Karte“ Visualisierung. Diese wird einmal angeklickt.



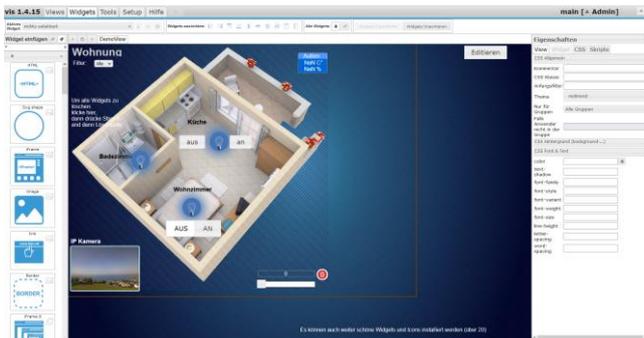
Danach erscheint ein Hinweis – der kann bestätigt werden:



Danach erscheint ein weiterer Hinweis, der mit OK bestätigt wird:

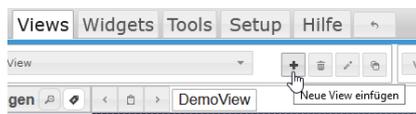


Nun erscheint die Default-Webseite zur Konfiguration.



Über den Menüpunkt Views legt man einen neuen View an:

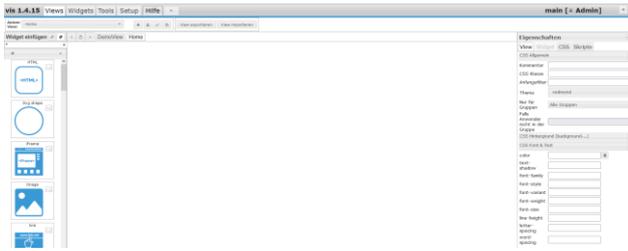
1. Views aktivieren, dann
2. Das Plus-Zeichen drücken



3. Im erscheinenden Feld einen neuen Namen vergeben: Multical21-Wasserszähler oder anderer Name (Home) – das ist wichtig für den späteren Seitenaufruf! Der Name der View wird an die URL angehängt – siehe unten

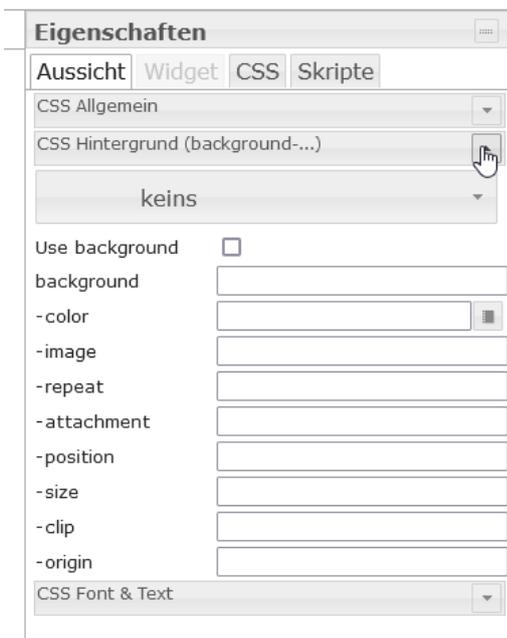


4. Widget einfügen
5. Und ENTER drücken oder den OK-Haken neben dem Feld betätigen
6. Nun öffnet sich ein neues leeres Fenster

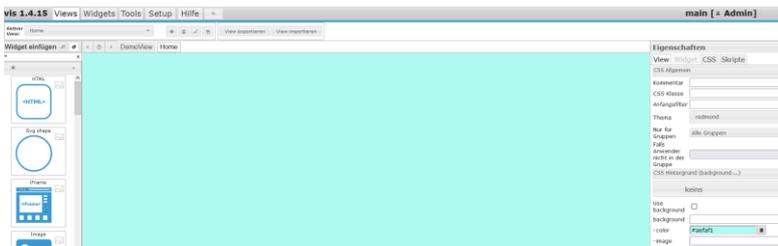


Setzen des Hintergrundes:

Am rechten Rand über Eigenschaften -> CSS Hintergrund (background...) ausklappen



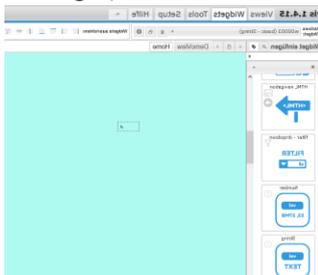
Danach die Farbe auswählen



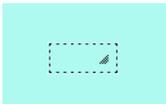
Und der Hintergrund ist hellblau – passend zum anzuzeigenden Zähler!

Nun werden die Steuerelemente positioniert.

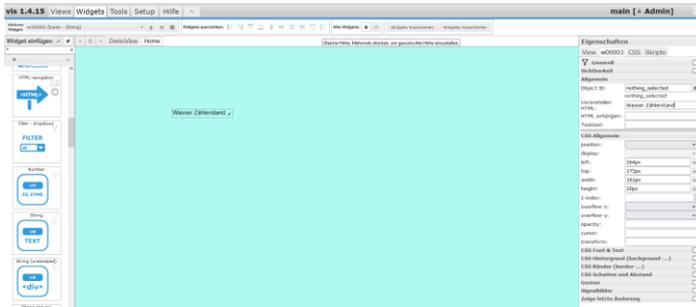
1. Überschrift einfügen über die linke Seite in der Leiste der Steuerelemente aussuchen: „String“ (mit dem linken Scrollbalken kann man in der Liste nach unten gehen)



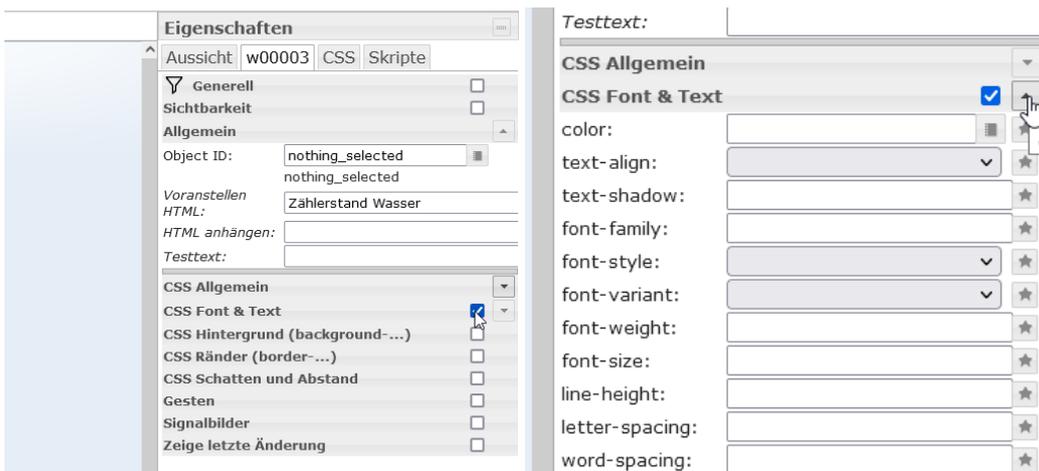
2. Dann per Drag and Drop das freie hellblaue Feld in den rechten Bereich ziehen, es bleibt ein kleiner mit Strichlinien umrandeter Kasten zurück.



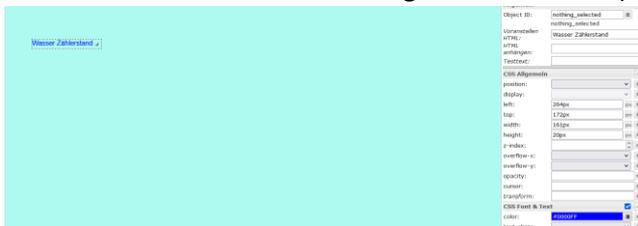
3. Dann im rechten Feld bei „Vorstellen HTML“ eingeben „Zählerstand Wasser“  
Das gestrichelte Feld kann man verändern, in dem man mit der Maus darüberfährt und wenn die Pfeile umschalten, diese per Linksklick halten und dann ziehen.



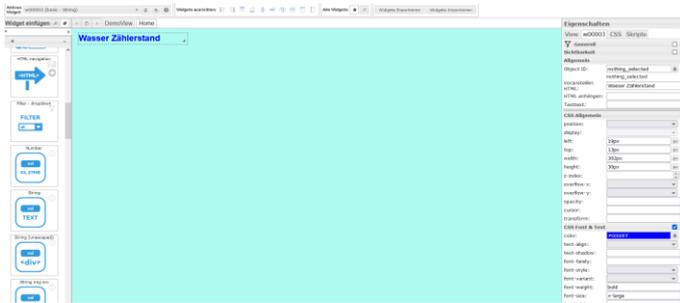
4. Nun kann die Farbe und Größe der Überschrift verändert werden über aktivieren des rechten Untermenüs CSS Font & Text (durch einen Haken im rechten Aktivierungsfeld, dann erscheint rechts ein Pfeil nach unten, in dem ein DropDownFeld aufklappt).



5. Nun kann bei color die Textfarbe geändert werden (blau)



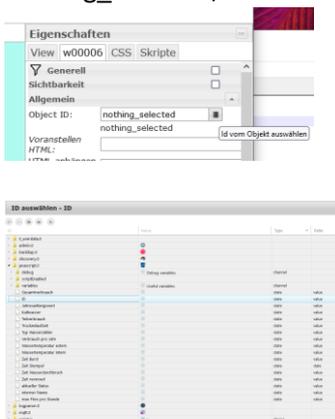
6. Und die Schriftgröße erhöht werden (font-size)(hier ggfs, wie oben beschrieben, die Feldgröße verändern!)



7. Jetzt wird das Feld positioniert, um die Zählernummer darzustellen  
Hier wird das Steuerelement „Number“ verwendet. Diese auch per Drag-and Drop auf die Fläche ziehen.



8. Dann das Feld anklicken und im rechten Bereich die Objekt-ID auswählen. Aktuell steht darin nothing\_selected, wenn man auf den rechten Knopf daneben klickt, geht ein Untermenü auf

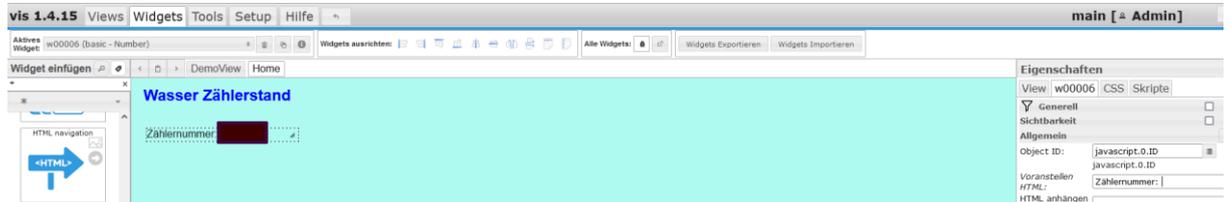


9. Es wird die Objektvariable aus dem Baum herausgesucht.  
Diese liegen unter javascript.0 -> ID (auf der rechten Seite müsste jetzt die Zähler-ID des Wasserzählers stehen)  
einfach die Auswahl (ID) doppelt! Anklicken, dann übernimmt er diese in die vorige Object-ID und springt in das vorige Fenster zurück.



10. Nun noch den Bezeichnungstext für den Wert angeben (im rechten Feld unter „Vorstellen HTML“):

Eingeben des Wertes: „Zählernummer :“ und mit <ENTER> bestätigen (hier kann man ein paar Leerzeichen noch mit einbringen, dann ist es etwas besser getrennt oder man teilt die Felder auf in ein Stringfeld mit der Bezeichnung und einem extra Number-Feld für den Wert – ganz nach gusto)

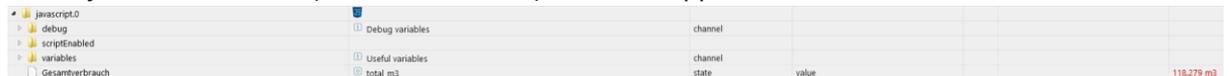


Das Fenster wieder etwas vergrößern.

11. Jetzt wird die Zählernummer angezeigt:

12. Nun wird der aktuelle Zählerstand hinzugefügt über Number per Drag-and-Drop in das Feld

13. Die Objekt-ID auswählen (Gesamtverbrauch) und mit Doppelklick übernehmen



14. Vorstellen HTML auswählen und Eingabe von: Wasserstand absolut: <LEERZEICHEN>

Ggfs. noch das Feld HTML anhängen (Plural) eingeben: <LEERZEICHEN> m3



15. Nunmehr das Feld für den Gesamtverbrauch hinzufügen (Number) mit dem Namen

„Verbrauch pro Jahr“

Danach wieder die HTML-Felder auf der rechten Seite füllen.

16. Nun können noch weitere Felder hinzugefügt werden.

17. Für die Anzeige mit dem Wasserzähler und der aktuellen Verbrauchswerte kann wie folgt vorgegangen werden:

- Heraussuchen eines Bildes des Wasserzähler
- Positionierung des Bildes auf der Webseite mit dem Steuerelement auf der linken Seite mit der ID „Image“
- Bei Quelle, dann die Datei zu dem Bild angeben (Upload der Datei und dann die Datei im Feld auswählen)
- Danach ein sog. Overlay, d.h. ein Numberfeld auf das Bild positionieren und hier die Object-ID „Gesamtverbrauch“ angeben.



Nun kann die Seite im Browser angezeigt werden.

Dann in einem neuen Tab kann die Webseite jetzt aufgerufen werden (Achtung auf die IP-Adresse achten, es ist die vom Raspberry Pi zu verwenden!)

Das Schema ist:

*<http://IP-Adresse-des-Raspi:8082/vis/index.html#Name-der-View>*

z.B.

<http://192.168.178.37:8082/vis/index.html#Multical21-Wasserzähler>

oder (wenn die Seite/View anders benannt wurde)

<http://192.168.178.37:8082/vis/index.html#Home>