FHEM als MQTT Publisher für eine Bresser 5in1 Wetter Center

ENTWURF nicht vollständig

Ich habe günstig eine 5in1 Bresser Wetterstation mit SW-Display bekommen.

Der 5in1 Sensor sendet seine Daten regelmäßig per 868Mhz zu seinem Empfänger. Da ich die Daten gerne in meinem ioBroker nutzen wollte, habe ich mich auf die Suche nach einer Integration gemacht.

Ich habe ich zwei Ansätze gefunden.

- SDR-RTL
- SIGNALDuino + CC1101

Da ich schon aus andere Projekte den SDR-RTL kenne und mir die Idee vom SIGNALDuino sehr gut gefällt habe ich mich für diesen Ansatz entschied.

Schnell ist mir aufgefallen, das es gar nicht so einfach ist dem CC1101 Werte zu entlocken. Zum Glück gibt es eine FHEM Implementierung die das meiste schon für einen erledigt. Noch schöner wäre sicherlich eine Lösung die direkt per ESP –> MQTT arbeiten würde, aber dies gibt es aktuell nicht. An dieser Stelle einen Herzlichen Dank an „elektron-bbs“ (github) für seine Geduld.
Damit ich wenig Arbeit habe, nutze ich eine DEBIAN Installation vom FHEM in einer KVM Umgebung. Auf das einrichten der KVM Umgebung gehe ich hier nicht ein, wichtig ist nur, das der USB-Stick an die Virtuelle Hardware weitergeleitet wird.

FHEM unter Debian installieren

es gibt eine sehr gute Anleitung direkt von den FHEM Machern.

Ich fasse die einzelne Schritte kurz zusammen:

Alle Schritte werden als User „root“ durchgeführt

```
# wget -qO - http://debian.fhem.de/archive.key | apt-key add -
# echo "deb http://debian.fhem.de/nightly/ /" >> /etc/apt/sources.list
# apt update
# apt install fhem
# apt install libdigest-crc-perl
```

- Danach ist die Version Version: 6.0.24458 (oder neuer) installier
- libdigest-crc-perl wird für den SIGNALDuino benötigt

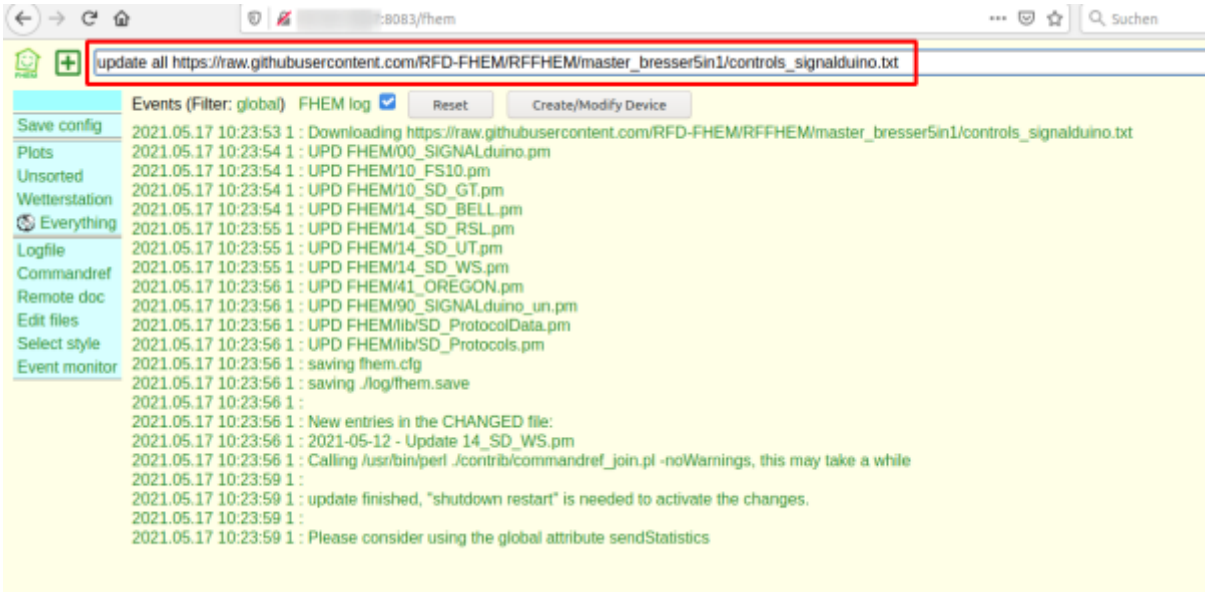
jetzt können wir die Wenoberfläche vom FHEM erreichen

```
http://<IP-Adresse_FHEM-Server>:8083
```

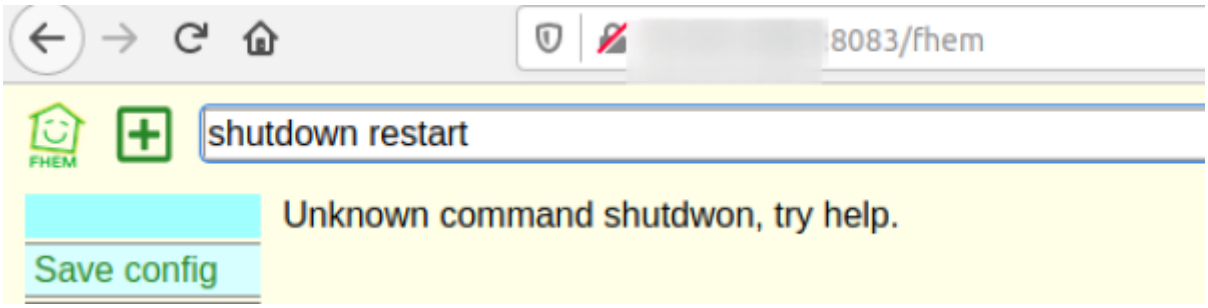
Installation vom SIGNALDuino Plugin im FHEM

Die folgende Zeile muss in der FHEM WEB-Comandline eingetragen werden

- update all https://raw.githubusercontent.com/RFD-FHEM/RFFHEM/master_bresser5in/controls_signalduino.txt



- FHEM neue Starten (shutdown restart)



Die USB-ID der Serielle Schnittstelle ermitteln

Damit wir das DEVICE anlegen können, müssen wir den Pfad auf dem Linux System ermitteln.

```
ls /dev/serial/by-id/ -l
```

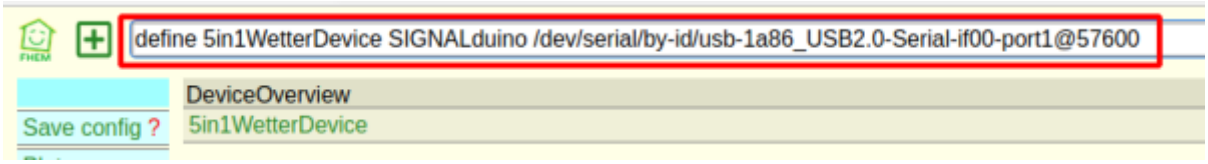
Ausgabe:

```
insgesamt 0
lrwxrwxrwx 1 root root 13 Mai 17 07:30 usb-1a86_USB2.0-Serial-if00-port1 ->
../../ttyUSB0
```

Ich habe somit den Pfad: **/dev/serial/by-id/usb-1a86_USB2.0-Serial-if00-port1**

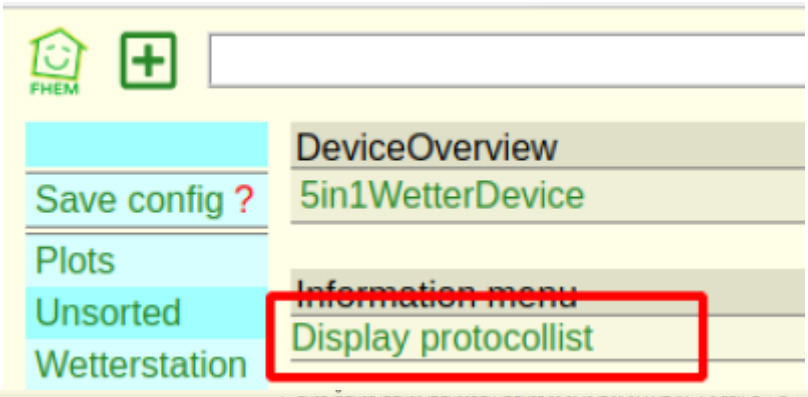
Anlegen des DEVICES im FHEM

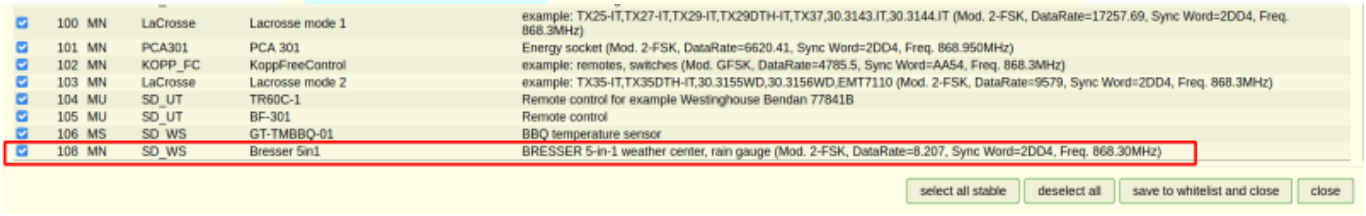
```
define 5in1WetterDevice SIGNALduino /dev/serial/by-id/usb-1a86_USB2.0-
Serial-if00-port1@57600
```



Das Device für den 5in1 Sensor konfigurieren

- Kontrolle ob das Protokoll 108 aktiviert ist.

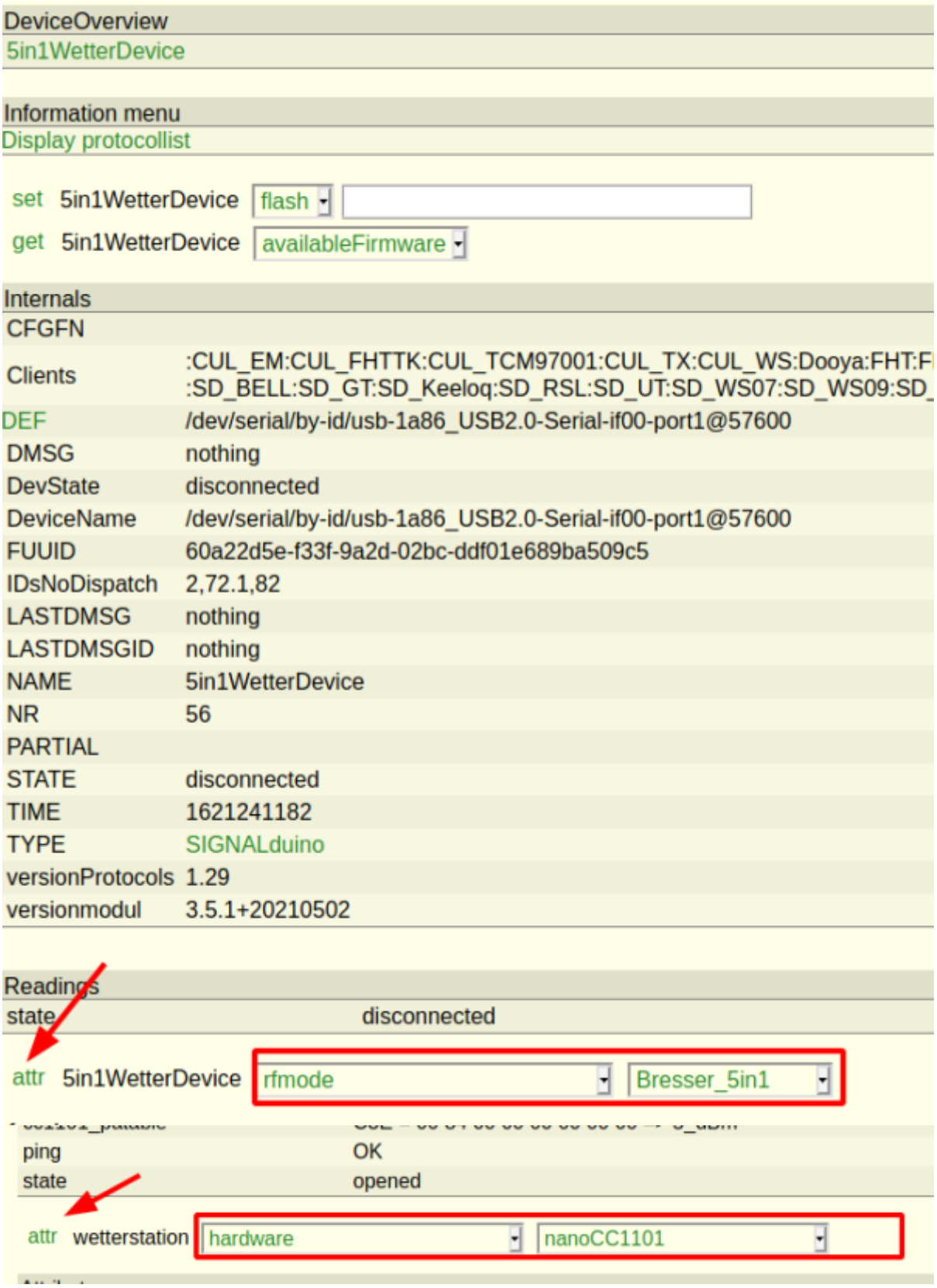




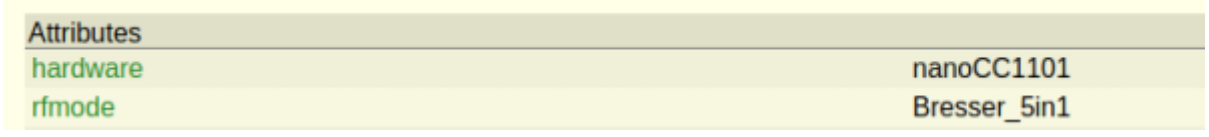
100	MN	LaCrosse	Lacrosse mode 1	example: TX25-IT,TX27-IT,TX29-IT,TX29DTH-IT,TX37,30.3143.IT,30.3144.IT (Mod. 2-FSK, DataRate=17257.69, Sync Word=2DD4, Freq. 868.3MHz)
101	MN	PCA301	PCA 301	Energy socket (Mod. 2-FSK, DataRate=6620.41, Sync Word=2DD4, Freq. 868.950MHz)
102	MN	KOPP_FC	KoppFreeControl	example: remotes, switches (Mod. GFSK, DataRate=4785.5, Sync Word=AA54, Freq. 868.3MHz)
103	MN	LaCrosse	Lacrosse mode 2	example: TX35-IT,TX35DTH-IT,30.3155WD,30.3156WD,EMT7110 (Mod. 2-FSK, DataRate=9579, Sync Word=2DD4, Freq. 868.3MHz)
104	MU	SD_UT	TR60C-1	Remote control for example Westinghouse Bendan 77841B
105	MU	SD_UT	BF-301	Remote control
106	MS	SD_WS	GT-TMBBQ-01	BBQ temperature sensor
108	MN	SD_WS	Bresser 5in1	BRESSER 5-in-1 weather center, rain gauge (Mod. 2-FSK, DataRate=8.207, Sync Word=2DD4, Freq. 868.30MHz)

select all stable | deselect all | save to whitelist and close | close

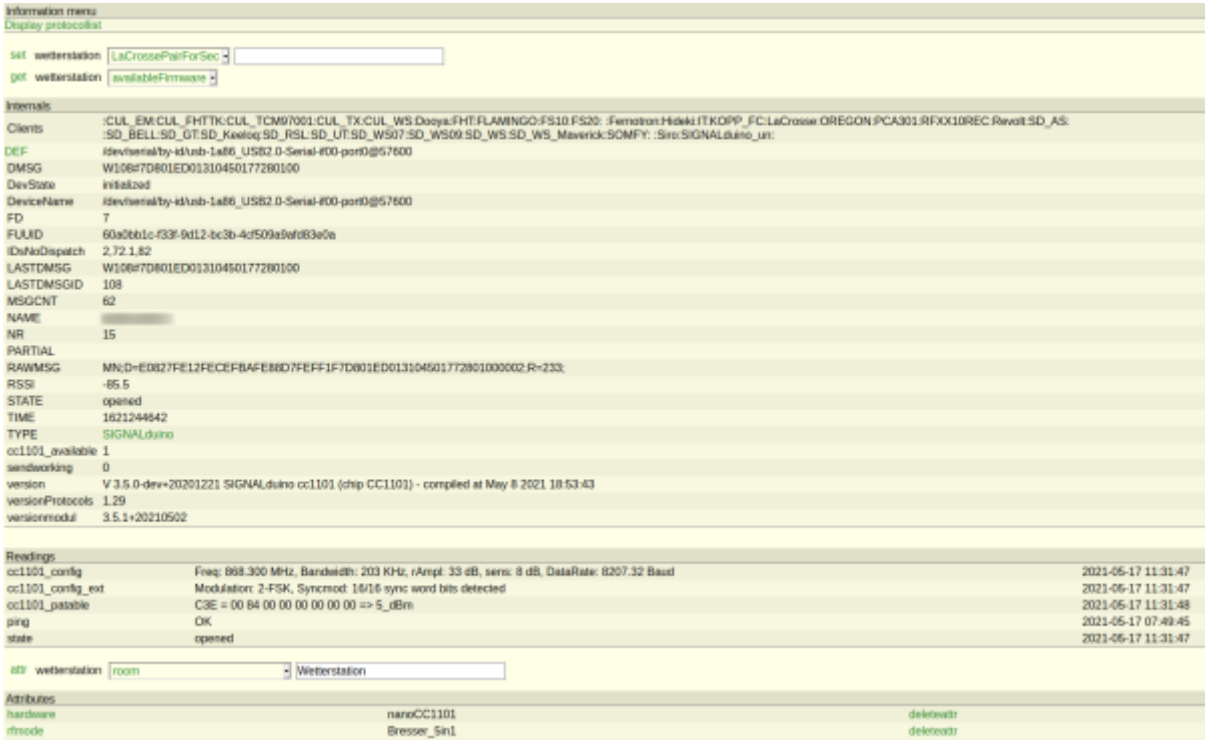
- „rfmode“ auf den Wert „Bresser_5in1“ setzen
- „hardware“ auf „CC1101“ seten



- Ergebniss

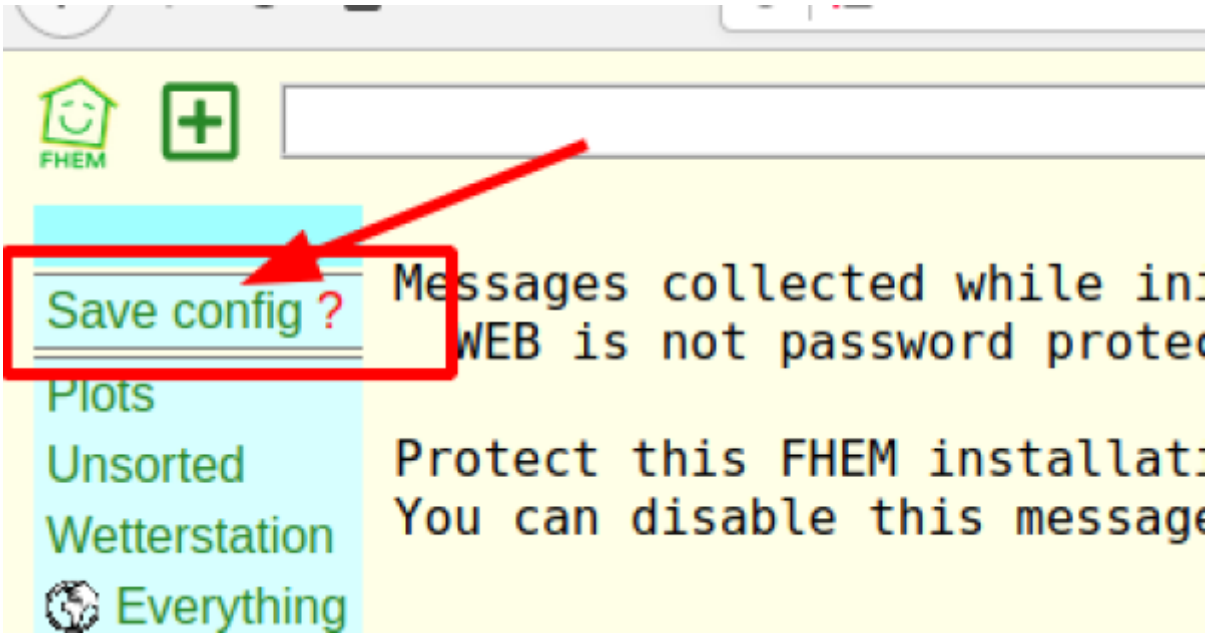


- Das DEVICE sollte jetzt so aussehen



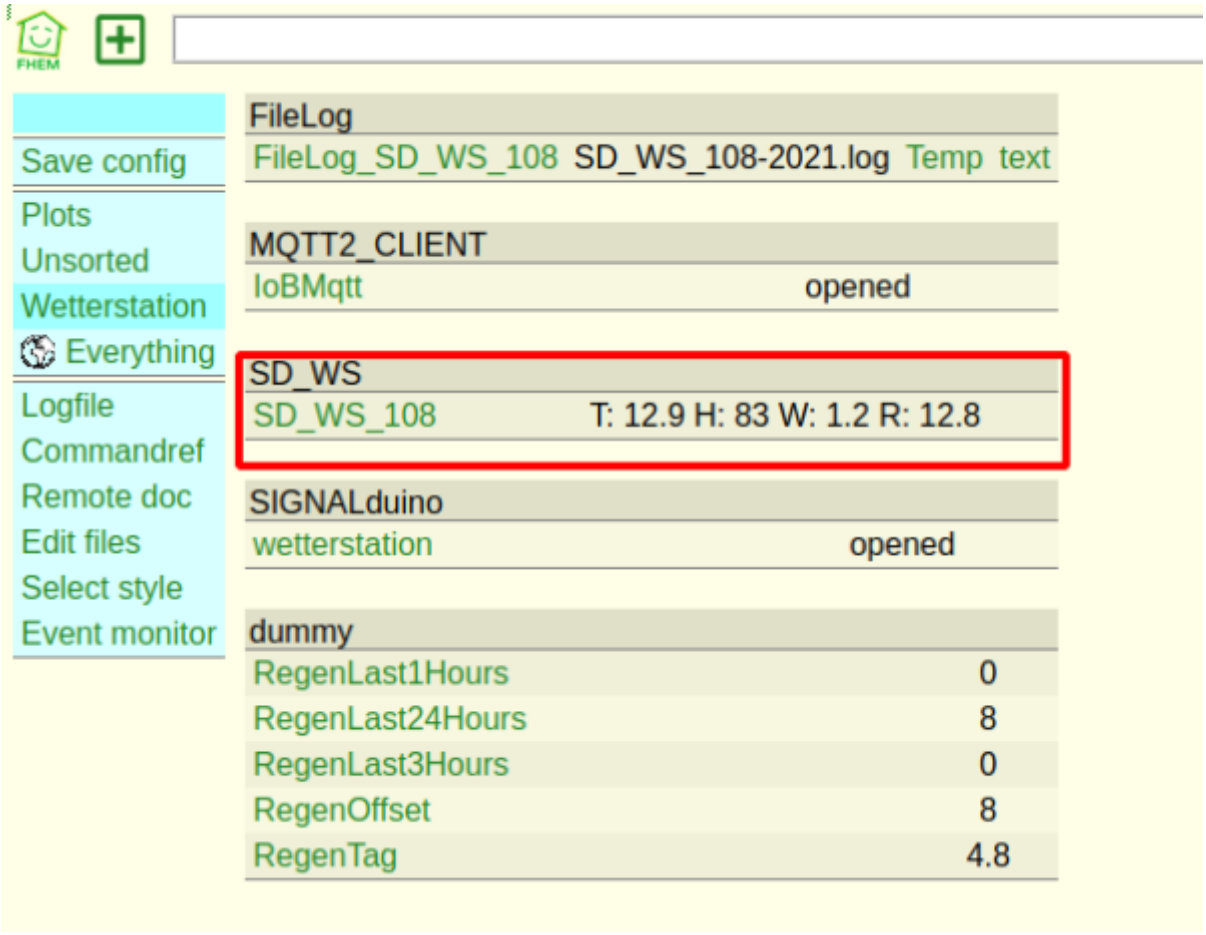
Sichern nicht vergessen

damit die ganze Anpassungen dauerhaft in die fhem.cfg eingetragen werden, muss man den SAVE Knopf drücken.

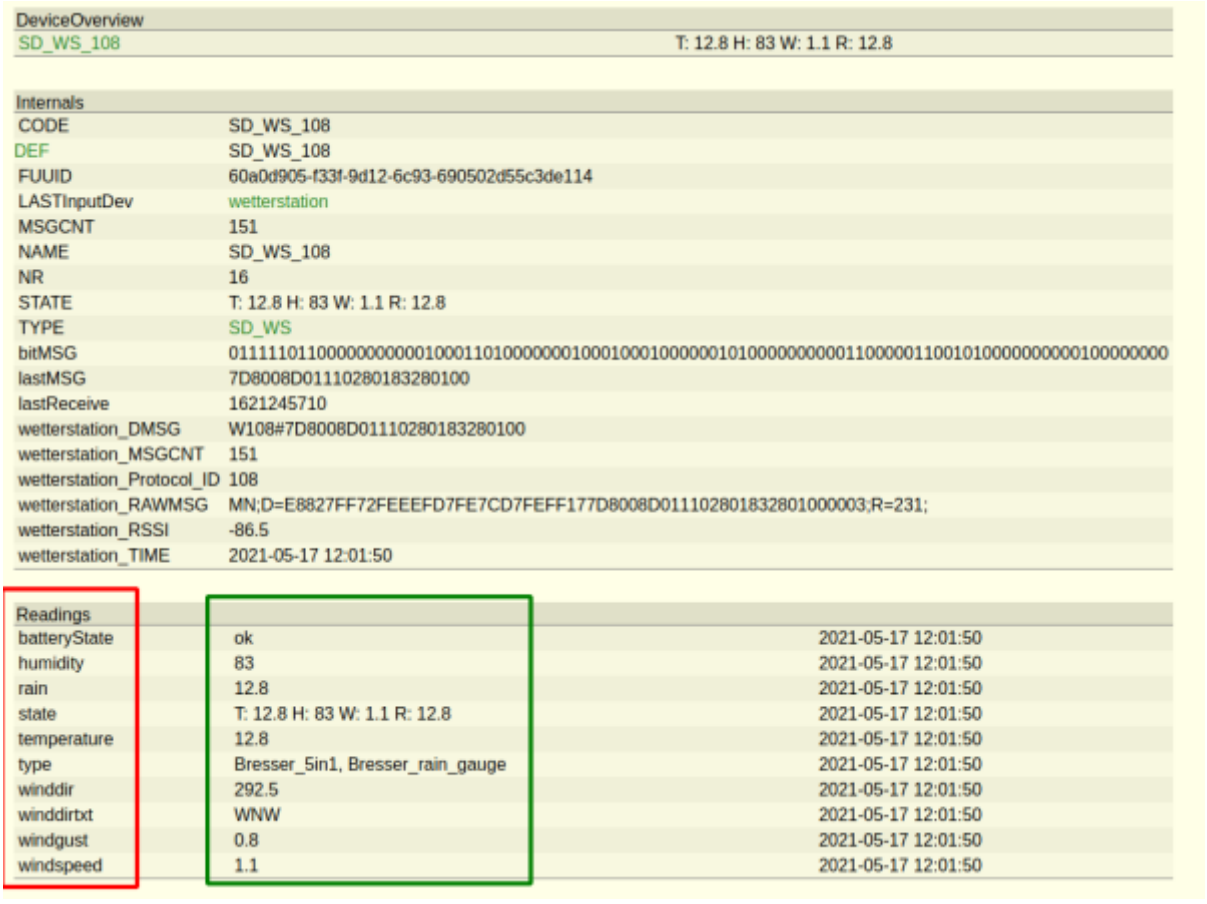


Der Wettersensor

Sobald das DEVICE 5 komplette Datensätze Empfangen hat, wir der Wettersensor automatisch als SD_WS_108 angelegt.



und die Messwerte können abgelesen werden



Configuration

Meine funktionierene Config inklusive MQTT richtung IOBroker:

LSD: fhem.cfg und 99_MyUtils.pm

- /opt/fhem/

fhem.cfg

```
attr global userattr cmdIcon devStateIcon:textField-long devStateStyle
icon sortby webCmd webCmdLabel:textField-long widgetOverride
attr global autoload_undefined_devices 1
attr global logfile ./log/fhem-%Y-%m.log
attr global modpath .
attr global statefile ./log/fhem.save
attr global verbose 3

define WEB FHEMWEB 8083 global
setuuid WEB 60a14760-f33f-9a2d-43f2-8f077cf58cefaaef

# Fake FileLog entry, to access the fhem log from FHEMWEB
define Logfile FileLog ./log/fhem-%Y-%m.log fakelog
setuuid Logfile 60a14760-f33f-9a2d-3180-7f6c99e99b45801e

define autocreate autocreate
setuuid autocreate 60a14760-f33f-9a2d-f8b4-25f9d8b81d3fbcbe
attr autocreate filelog ./log/%NAME-%Y.log

define eventTypes eventTypes ./log/eventTypes.txt
setuuid eventTypes 60a14760-f33f-9a2d-858e-e63292934e09ab90

# Disable this to avoid looking for new USB devices on startup
define initialUsbCheck notify global:INITIALIZED usb create
setuuid initialUsbCheck 60a0204d-f33f-9d12-3c39-bede2fc02b0566c4

define wetterstation SIGNALduino /dev/serial/by-id/usb-1a86_USB2.0-
Serial-if00-port0@57600
setuuid wetterstation 60a0bb1c-f33f-9d12-bc3b-4cf509a9afd83e0a
attr wetterstation hardware nanoCC1101
attr wetterstation rfmode Bresser_5in1
attr wetterstation room Wetterstation
define SD_WS_108 SD_WS SD_WS_108
setuuid SD_WS_108 60a0d905-f33f-9d12-6c93-690502d55c3de114
attr SD_WS_108 event-min-interval .*:300
attr SD_WS_108 event-on-change-reading .*
attr SD_WS_108 room Wetterstation
attr SD_WS_108 webCmdLabel humidity
define FileLog_SD_WS_108 FileLog ./log/SD_WS_108-%Y.log SD_WS_108
```

```
setuuid FileLog_SD_WS_108 60a0d905-f33f-9d12-d508-f5badf5126a55e29
attr FileLog_SD_WS_108 logtype temp4:Temp,text
attr FileLog_SD_WS_108 room Wetterstation
define SVG_SD_WS_108 SVG FileLog_SD_WS_108:SVG_SD_WS_108:CURRENT
setuuid SVG_SD_WS_108 60a0d905-f33f-9d12-1605-43c4eb71eb2aed15
attr SVG_SD_WS_108 label "SD_WS_108 Min $data{min1}, Max $data{max1},
Last $data{currval1}"
attr SVG_SD_WS_108 room Plots

define IoBMqtt MQTT2_CLIENT 10.101.3.102:1883
setuuid IoBMqtt 60a15917-f33f-9a2d-7252-bb75d91337d44746
attr IoBMqtt autocreate no
attr IoBMqtt clientId fhem
attr IoBMqtt lwt offline
attr IoBMqtt room Wetterstation
attr IoBMqtt username iobroker
define SD_WS_108_notify_batteryState notify SD_WS_108:batteryState:.*
set IoBMqtt publish Wetterstation/Batterie $EVTPART1
setuuid SD_WS_108_notify_batteryState 60a18961-f33f-9a2d-
f536-8cb0a48b0f7f9a00

define SD_WS_108_notify_temp notify SD_WS_108:temperature:.* set
IoBMqtt publish Wetterstation/Temp $EVTPART1
setuuid SD_WS_108_notify_temp 60a16d6b-f33f-9a2d-20ae-da5fc1e0104f540f
define SD_WS_108_notify_windspeed notify SD_WS_108:windspeed:.* set
IoBMqtt publish Wetterstation/WindSpeed $EVTPART1
setuuid SD_WS_108_notify_windspeed 60a16e1b-
f33f-9a2d-92ae-2b7c2e675db3e0ec
define SD_WS_108_notify_windir notify SD_WS_108:winddir:.* set IoBMqtt
publish Wetterstation/WindrichtungGrad $EVTPART1
setuuid SD_WS_108_notify_windir 60a16e7d-
f33f-9a2d-0d48-0c0db4d2430a1ea2
define SD_WS_108_notify_humidity notify SD_WS_108:humidity:.* set
IoBMqtt publish Wetterstation/Luftfeuchtigkeit $EVTPART1
setuuid SD_WS_108_notify_humidity 60a16ed2-f33f-9a2d-a5ea-
e36f23723a2ecb27
define SD_WS_108_notify_winddirtxt notify SD_WS_108:winddirtxt:.* set
IoBMqtt publish  Wetterstation/WindrichtungTEXT $EVTPART1
setuuid SD_WS_108_notify_winddirtxt 60a16f6b-f33f-9a2d-2f01-
df89df1e3cb6e317
define SD_WS_108_notify_windgust notify SD_WS_108:windgust:.* set
IoBMqtt publish Wetterstation/WindBoeen $EVTPART1
setuuid SD_WS_108_notify_windgust 60a174a8-f33f-9a2d-d6a1-
a5cfe34b4f354f5f
define SD_WS_108_notify_rain notify SD_WS_108:rain:.* set IoBMqtt
publish Wetterstation/Regen $EVTPART1
setuuid SD_WS_108_notify_rain 60a17683-f33f-9a2d-afd0-d61a4f0f9323b7d2

define RegenOffset dummy
setuuid RegenOffset 60a17e2f-f33f-9a2d-3c2f-35d51a991d89a4f6
attr RegenOffset room Wetterstation
```

```
define RegenTag dummy
setuuid RegenTag 60a17e42-f33f-9a2d-9dbe-f651d87d2c584877
attr RegenTag room Wetterstation
define RegenLast1Hours dummy
setuuid RegenLast1Hours 60a17e55-f33f-9a2d-fbfb-92ff6b05a4baf7a1
attr RegenLast1Hours room Wetterstation
define RegenLast3Hours dummy
setuuid RegenLast3Hours 60a17e5e-f33f-9a2d-df69-a999729323ccef50
attr RegenLast3Hours room Wetterstation
define RegenLast24Hours dummy
setuuid RegenLast24Hours 60a17e63-f33f-9a2d-353b-2ae73660303c9905
attr RegenLast24Hours room Wetterstation

define RegenNotify notify SD_WS_108:rain.* {\
  my $menge = (ReadingsVal("SD_WS_108", "rain", 0) -
ReadingsVal("RegenOffset", "state", 0));;\
  my $last1hours = myDiff("3600", "FileLog_SD_WS_108", "10:::");;\
  my $last3hours = myDiff("10800", "FileLog_SD_WS_108", "10:::");;\
  my $last24hours = myDiff("86400", "FileLog_SD_WS_108", "10:::");;\
  fhem("set RegenTag $menge");;\
  fhem("set RegenLast1Hours $last1hours");;\
  fhem("set RegenLast3Hours $last3hours");;\
  fhem("set RegenLast24Hours $last24hours");;\
}
setuuid RegenNotify 60a18250-f33f-9a2d-91d9-df2f92a22a9507cd

define RegenOffsetReset at *00:00:00 {\
  my $offset = ReadingsVal("SD_WS_108", "rain", 0);;\
  $offset    = sprintf("%.1f",$offset );;\
  fhem("set RegenOffset $offset");; \
}
setuuid RegenOffsetReset 60a18250-f33f-9a2d-0f4a-d3cd3d917fed6282

define RegenTag_notify notify RegenTag:.* set IoBMqtt publish
Wetterstation/RegenHeuteTag $EVENT
define RegenLast3Hours_notify notify RegenLast3Hours:.* set IoBMqtt
publish  Wetterstation/RegenLast3h $EVENT
setuuid RegenLast3Hours_notify 60a18c27-f33f-9a2d-4325-f906bdd71cf2ff63
define RegenLast1Hours_notify notify RegenLast1Hours:.* set IoBMqtt
publish Wetterstation/RegenLast1h $EVENT
setuuid RegenLast1Hours_notify 60a18db5-f33f-9a2d-375d-d0ab533a85dfc6ea
define RegenLast24Hours_notify notify RegenLast24Hours:.* set IoBMqtt
publish  Wetterstation/RegenLast24h $EVENT
setuuid RegenLast24Hours_notify 60a18db5-f33f-9a2d-fb9c-
cdbb3fcdd5e7bf66
```

- /opt/fhem/FHEM/

99_MyUtils.pm

```
package main;
 use strict;
 use warnings;
 use POSIX;
 sub
 MyUtils_Initialize($$)
 {
  my ($hash) = @_;
 }
#####################################################################
########
 #
 #  Moving average
 #
 #  Aufruf: movingAverage(devicename,readingname,zeitspanne in s)
 #
#####################################################################
########

 sub movingAverage($$$){
    my ($name,$reading,$avtime) = @_;
    my $hash = $defs{$name};
    my @new = my ($val,$time) =
($hash->{READINGS}{$reading}{VAL},$hash->{READINGS}{$reading}{TIME});
    my ($cyear, $cmonth, $cday, $chour, $cmin, $csec) = $time =~
/(\d+)-(\d+)-(\d+)\s(\d+):(\d+):(\d+)/;
    my $ctime = $csec+60*$cmin+3600*$chour;
    my $num;
    my $arr;
    #-- initialize if requested
    if( ($avtime eq "-1") ){
      $hash->{READINGS}{$reading}{"history"}=undef;
    }
    #-- test for existence
    if( !$hash->{READINGS}{$reading}{"history"}){
       #Log 1,"ARRAY CREATED";
       push(@{$hash->{READINGS}{$reading}{"history"}},\@new);
       $num = 1;
       $arr=\@{$hash->{READINGS}{$reading}{"history"}};
    } else {
       $num = int(@{$hash->{READINGS}{$reading}{"history"}});
       $arr=\@{$hash->{READINGS}{$reading}{"history"}};
       my $starttime = $arr->[0][1];
       my ($syear, $smonth, $sday, $shour, $smin, $ssec) = $starttime
=~ /(\d+)-(\d+)-(\d+)\s(\d+):(\d+):(\d+)/;
       my $stime = $ssec+60*$smin+3600*$shour;
       #-- correct for daybreak
       $stime-=86400
         if( $stime > $ctime);
       if( ($num < 25)&&( ($ctime-$stime)<$avtime) ){
         #Log 1,"ARRAY has $num elements, adding another one";
```

```
          push(@{$hash->{READINGS}{$reading}{"history"}},\@new);
        }else{
          shift(@{$hash->{READINGS}{$reading}{"history"}});
          push(@{$hash->{READINGS}{$reading}{"history"}},\@new);
        }
      }
      #-- output and average
      my $average = 0;
      for(my $i=0;$i<$num;$i++){
        $average+=$arr->[$i][0];
        Log 4,"[$name moving average] Value = ".$arr->[$i][0]." Time =
".$arr->[$i][1];
      }
      $average=sprintf( "%5.3f", $average/$num);
      #--average
      Log 4,"[$name moving average] calculated over $num values is
$average";
      return $average;
  }


 1;

# myDiff
# berechnet die Differenz aus der ersten Zeile eines LogFiles und der
letzten Zeile eines LogFiles über einen Zeitraum zwischen einem
Zeitpunkt in der Vergangenheit und dem Zeitpunkt des Aufrufs
sub
myDiff($$$)
{
 my ($offset,$logfile,$cspec) = @_;
 my $period_s = strftime "%Y-%m-%d\x5f%H:%M:%S", localtime(time-
$offset);
 my $period_e = strftime "%Y-%m-%d\x5f%H:%M:%S", localtime;
 my $oll = $attr{global}{verbose};
 $attr{global}{verbose} = 0;
 my @logdata = split("\n", fhem("get $logfile - - $period_s $period_e
$cspec"));
 $attr{global}{verbose} = $oll;
 my ($cnt, $first, $last, $diff) = (0)x4;
 foreach (@logdata){
  my @line = split(" ", $_);
  if(defined $line[1] && "$line[1]" ne ""){
   $cnt += 1;
    if ($cnt == 1) {
     $first = $line[1];
    }
   $last = $line[1];
  }
 }
```

```
$diff = $last - $first;
Log 4, ("myDiff: File: $logfile, Field: $cspec, Period: $period_s bis
$period_e, First: $first, Last: $last, Diff: $diff");
return $diff;
}
```