

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 entschieden.

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 einzelnen 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

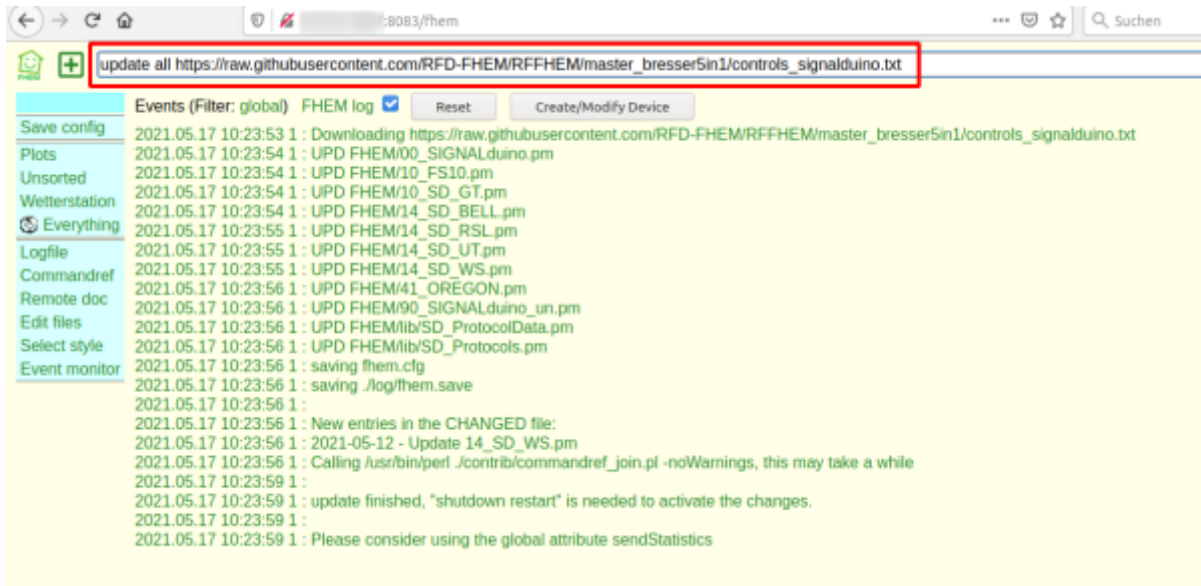
jetzt können wir die Web-Oberfläche vom FHEM erreichen

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

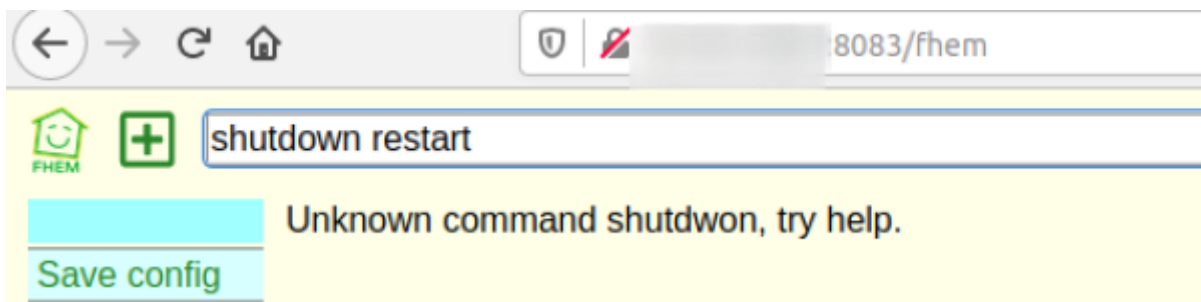
Installation vom SIGNALDuino Plugin im FHEM

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

- update all [https://raw.githubusercontent.com/RFD-FHEM/RFFHEM/master_bresser5in1/controls_signalduino.txt](https://raw.githubusercontent.com/RFD-FHEM/RFFHEM/master/bresser5in1/controls_signalduino.txt)



- FHEM neue Starten (shutdown restart)



Die USB-ID der Seriellen Schnittstelle ermitteln

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

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

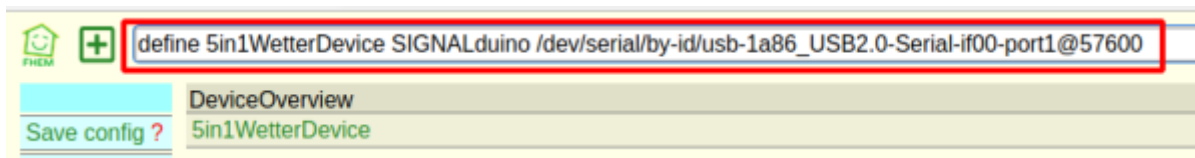
Ausgabe:

```
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**

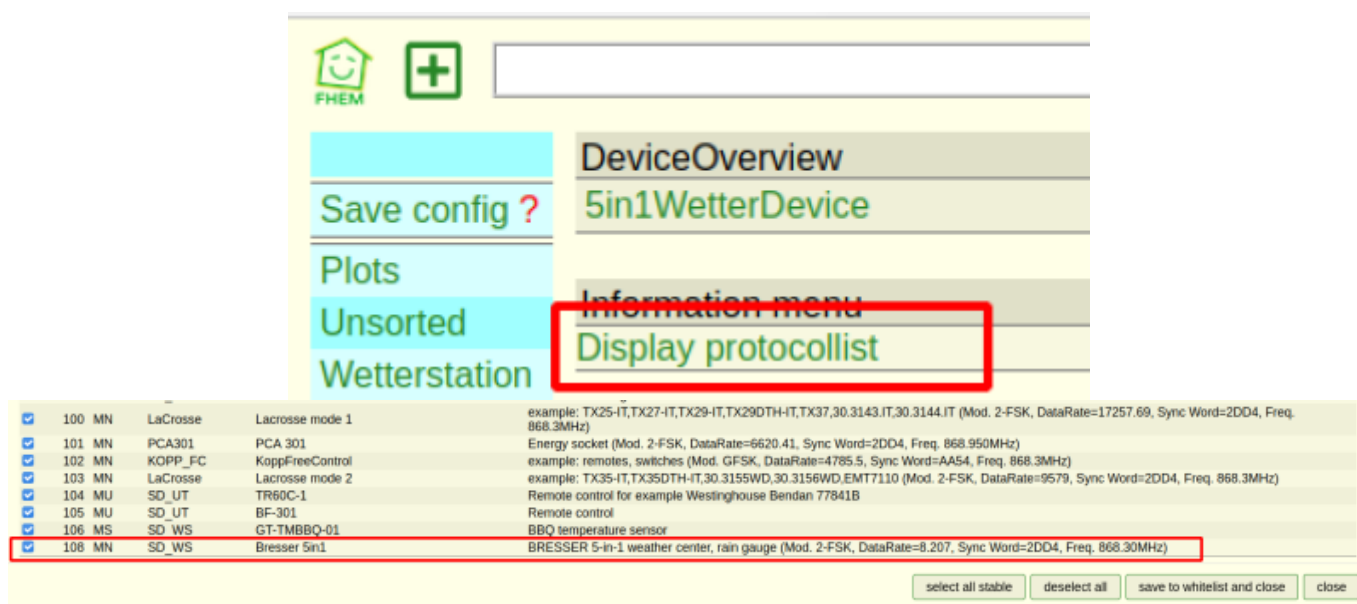
Anlegen des DEVICES im FHEM

```
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.



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

DeviceOverview	
5in1WetterDevice	
Information menu	
Display protocollist	
set 5in1WetterDevice	flash <input type="text"/>
get 5in1WetterDevice	availableFirmware <input type="text"/>
Internals	
CFGFN	
Clients	:CUL_EM:CUL_FHTTK:CUL_TCM97001:CUL_TX:CUL_WS:Dooya:FHT:F
DEF	/dev/serial/by-id/usb-1a86_USB2.0-Serial-if00-port1@57600
DMSG	nothing
DevState	disconnected
DeviceName	/dev/serial/by-id/usb-1a86_USB2.0-Serial-if00-port1@57600
FUUID	60a22d5e-f33f-9a2d-02bc-ddf01e689ba509c5
IDsNoDispatch	2,72.1,82
LASTDMSG	nothing
LASTDMSGID	nothing
NAME	5in1WetterDevice
NR	56
PARTIAL	
STATE	disconnected
TIME	1621241182
TYPE	SIGNALduino
versionProtocols	1.29
versionmodul	3.5.1+20210502
Readings	
state	disconnected
attr 5in1WetterDevice	rfmode <input type="text"/> Bresser_5in1 <input type="text"/>
ping	OK
state	opened
attr wetterstation	hardware <input type="text"/> nanoCC1101 <input type="text"/>

- Ergebniss

Attributes	
hardware	nanoCC1101
rfmode	Bresser_5in1

- Das DEVICE sollte jetzt so aussehen

Information menu
Display protocolist

set wetterstation [LaCrossePoiForSec]
get wetterstation [availableFirmware]

Intervall: []

Clients: CUL_EM:CUL_FHTK:CUL_TCM97001:CUL_TX:CUL_WS_Daoya:FHT:FLAMINGO:FS10:FS20:Femtron:Hioki:ITKOPP:FC:LaCrosse OREGON:PCA301:RFX10REC:Revolt:SD_AS:SD_BELL:SD_GT:SD_Kewloq:SD_RSL:SD_UT:SD_WS07:SD_WS09:SD_WS:SD_WS_Maverick:SCMPY:;Src:SIGNALduino_un;

DEF: id=serialby-id/usb-1a86_USB2.0-Serial-#00-port0@57600

DMSG: W1087D801ED01310450177280100

DevState: initialized

DeviceName: id=serialby-id/usb-1a86_USB2.0-Serial-#00-port0@57600

FD: 7

FUID: 60a0bb1c-f33f-9d12-bc3b-4d509a5ef083e0e

IDoNoDispatch: 2.72.1.82

LASTMSG: W1087D801ED01310450177280100

LASTMSGID: 108

MSGCNT: 62

NAME: []

NR: 15

PARTIAL: []

RAWMSG: MN:D=E0827FE12FCEFBFAFE8ED7FEFF1F7D801ED013104501772801000002:R=233;

RSSI: -85.5

STATE: opened

TIME: 1621244642

TYPE: SIGNALduino

cc1101_available: 1

sendworking: 0

version: V 3.5.0-dev+20201221 SIGNALduino cc1101 (chip CC1101) - compiled at May 8 2021 18:53:43

versionProtocol: 1.29

versionmodule: 3.5.1+20210502

Readings:

cc1101_config	Freq: 868.300 MHz, Bandwidth: 203 KHz, rAmp: 33 dB, sens: 8 dB, DataRate: 8207.32 Baud	2021-05-17 11:31:47
cc1101_config_ext	Modulation: 2-FSK, Syncmod: 16/16 sync word bits detected	2021-05-17 11:31:47
cc1101_passible	CSE = 00 84 00 00 00 00 00 => 5_dBm	2021-05-17 11:31:48
ping	OK	2021-05-17 07:49:45
state	opened	2021-05-17 11:31:47

id wetterstation [room] [Wetterstation]

Attributes:

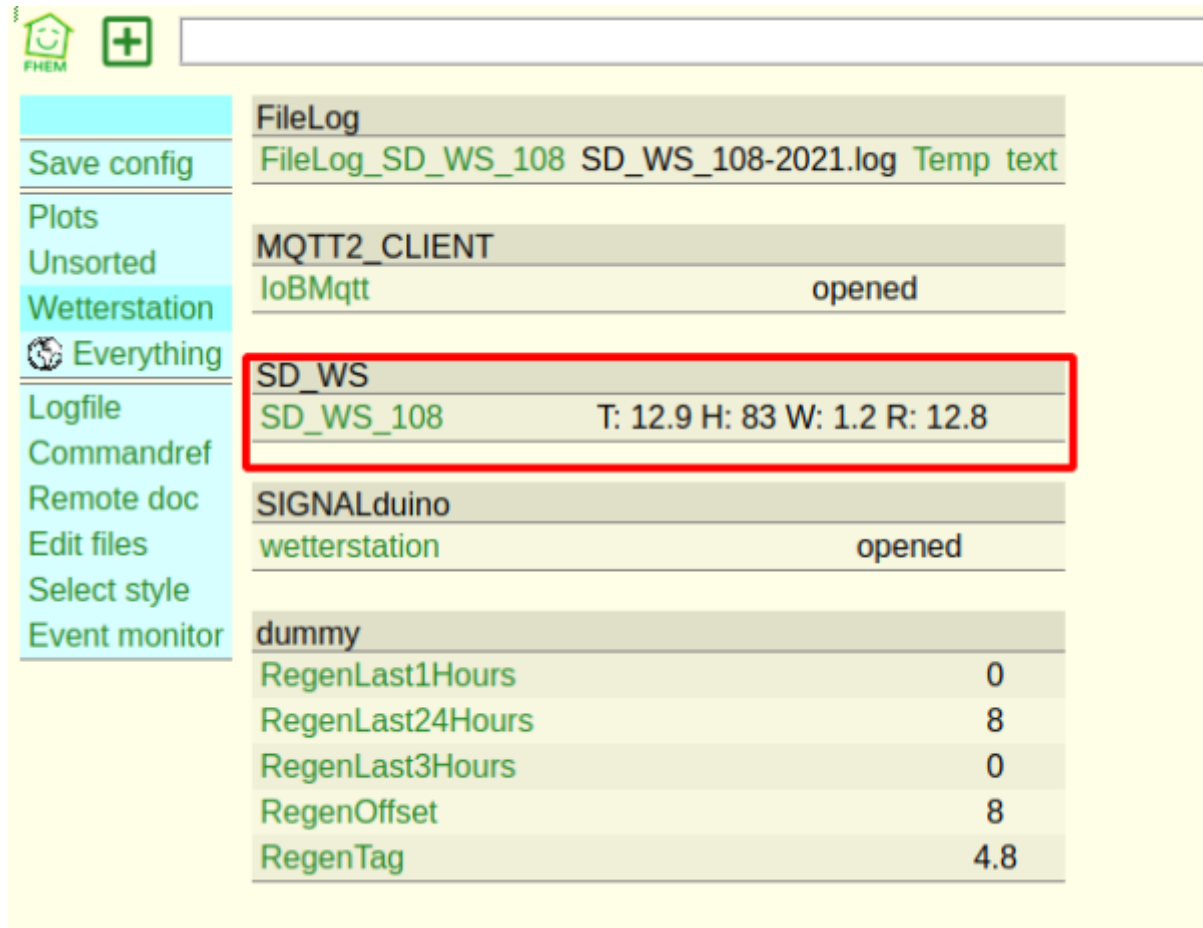
hardware	nanoCC1101	deleteattr
rmode	Bresser_Sn1	deleteattr

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, wird der Wettersensor automatisch als SD_WS_108 angelegt.



The screenshot shows the FHEM web interface. On the left is a sidebar with navigation links: Home, Add, Save config, Plots, Unsorted, Wetterstation, Everything, Logfile, Commandref, Remote doc, Edit files, Select style, and Event monitor. The main content area displays the configuration for the 'SD_WS' device. A red box highlights the 'SD_WS' section, which shows the device name 'SD_WS_108' and its current readings: 'T: 12.9 H: 83 W: 1.2 R: 12.8'. Below this, the 'SIGNALduino' section shows the device 'wetterstation' as 'opened'. At the bottom, the 'dummy' section shows various sensor readings: RegenLast1Hours (0), RegenLast24Hours (8), RegenLast3Hours (0), RegenOffset (8), and RegenTag (4.8).

FileLog	
FileLog_SD_WS_108	SD_WS_108-2021.log Temp text

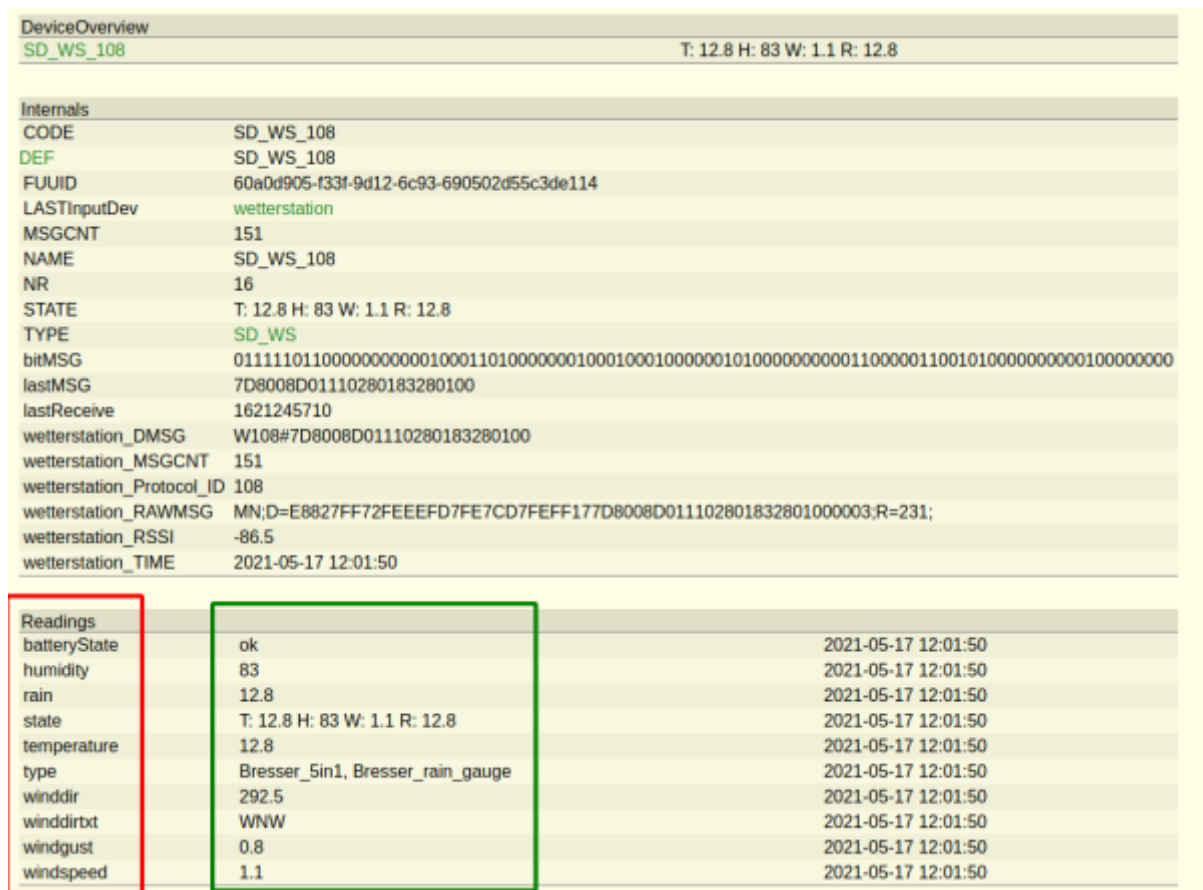
MQTT2_CLIENT	
IoBMqtt	opened

SD_WS	
SD_WS_108	T: 12.9 H: 83 W: 1.2 R: 12.8

SIGNALduino	
wetterstation	opened

dummy	
RegenLast1Hours	0
RegenLast24Hours	8
RegenLast3Hours	0
RegenOffset	8
RegenTag	4.8

und die Messwerte können abgelesen werden



The screenshot shows the 'DeviceOverview' for the 'SD_WS_108' device. The top section displays the device name and its current readings: 'T: 12.8 H: 83 W: 1.1 R: 12.8'. Below this is a table of internal device information, including CODE, DEF, FUUId, LASTInputDev, MSGCNT, NAME, NR, STATE, TYPE, bitMSG, lastMSG, lastReceive, wetterstation_DMSG, wetterstation_MSGCNT, wetterstation_Protocol_ID, wetterstation_RAWMSG, wetterstation_RSSI, and wetterstation_TIME. A red box highlights the 'Readings' section, which shows a list of sensor readings: batteryState (ok), humidity (83), rain (12.8), state (T: 12.8 H: 83 W: 1.1 R: 12.8), temperature (12.8), type (Bresser_5in1, Bresser_rain_gauge), winddir (292.5), winddirbt (WNW), windgust (0.8), and windspeed (1.1). A green box highlights the 'state' and 'type' readings.

DeviceOverview		
SD_WS_108	T: 12.8 H: 83 W: 1.1 R: 12.8	

Internals		
CODE	SD_WS_108	
DEF	SD_WS_108	
FUUId	60a0d905-f33f-9d12-6c93-690502d55c3de114	
LASTInputDev	wetterstation	
MSGCNT	151	
NAME	SD_WS_108	
NR	16	
STATE	T: 12.8 H: 83 W: 1.1 R: 12.8	
TYPE	SD_WS	
bitMSG	01111101100000000000100011010000000100010001000000101000000000110000011001010000000000100000000	
lastMSG	7D8008D01110280183280100	
lastReceive	1621245710	
wetterstation_DMSG	W108#7D8008D01110280183280100	
wetterstation_MSGCNT	151	
wetterstation_Protocol_ID	108	
wetterstation_RAWMSG	MN;D=E8827FF72FEEFD7FE7CD7FEFF177D8008D011102801832801000003;R=231;	
wetterstation_RSSI	-86.5	
wetterstation_TIME	2021-05-17 12:01:50	

Readings		
batteryState	ok	2021-05-17 12:01:50
humidity	83	2021-05-17 12:01:50
rain	12.8	2021-05-17 12:01:50
state	T: 12.8 H: 83 W: 1.1 R: 12.8	2021-05-17 12:01:50
temperature	12.8	2021-05-17 12:01:50
type	Bresser_5in1, Bresser_rain_gauge	2021-05-17 12:01:50
winddir	292.5	2021-05-17 12:01:50
winddirbt	WNW	2021-05-17 12:01:50
windgust	0.8	2021-05-17 12:01:50
windspeed	1.1	2021-05-17 12:01:50

Configuration

Meine funktionierende 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
setuid 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
setuid Logfile 60a14760-f33f-9a2d-3180-7f6c99e99b45801e

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

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

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

define wetterstation SIGNALduino /dev/serial/by-id/usb-1a86_USB2.0-
Serial-if00-port0@57600
setuid 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
setuid 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_winddir notify SD_WS_108:winddir:.* set IoBMqtt
publish Wetterstation/WindrichtungGrad $EVTPART1
setuuid SD_WS_108_notify_winddir 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-
cddb3fcdd5e7bf66
```

- /opt/fhem/FHEM/

```

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, $smmonth, $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;  
}
```

From:

<https://www.myworkroom.de/> - **Sodele**

Permanent link:

<https://www.myworkroom.de/p-lsd:bresser5in1wetterstation:fhemtomqtt>

Last update: **2021/05/17 10:04**

