
KONZEPTION UND IMPLEMENTIERUNG VON SMARTHOME

Schritt-für-Schritt-Anleitung

Inhaltsverzeichnis

1.	Setup Raspberry Pi	1
1.1.	Raspberry Pi Imager und Setup	1
1.2.	FHEM Setup	3
1.2.1.	FHEM Setup via Skript	4
1.2.1.1.	Das Hauptinstallationskript	4
1.2.1.2.	Die ergänzenden Skripte	5
1.2.2.	FHEM Setup manuell	6
1.2.3.	Alternatives FHEM Setup	7
1.3.	FTUI Setup	8
1.4.	Sicherungskonzept	9
1.5.	Konfiguration einer CA und HTTPS	11
2.	Implementation in FHEM	18
2.1.	Funktion Weihnachtsbeleuchtung	18
2.1.1.	Flashen und Einrichten der Funksteckdose	18
2.1.2.	Manueller Betrieb	19
2.1.3.	Betriebsmodus Feste Betriebszeit	19
2.1.4.	Betriebsmodus Sonnenaufgang und Sonnenuntergang	20
2.1.5.	Betriebsmodus Sonnenaufgang und Sonnenuntergang mit Uhrzeiten	20
2.1.6.	Betriebsmodus Anwesenheitserkennung	21
2.1.7.	Betriebsmodusauswahl	23
2.2.	Funktion Sturmwarnung	24
2.2.1.	Einrichten eines Bots zum Senden von Telegram Nachrichten über FHEM	24
2.2.2.	Sensordaten auslesen	27
2.2.3.	Anlegen der Dummy-Devices für die Eingabe der Schwellenwerte und für den Schalter der Sturmwarnung	29
2.2.4.	Hinzufügen von Do-If-Abfragen	30
2.3.	Funktion Photovoltaik/Balkonkraftwerk	32
2.3.1.	Einrichtung Shelly	32
2.3.2.	Einrichten einer DB zum Speichern von Daten	33
2.3.3.	Einfaches Heizen	35
2.3.4.	Optimiertes Heizen	37
2.3.5.	Warmwasserboiler	43
2.3.6.	Kombination zwischen Warmwasserboiler und Infrarotheizung	46

3.	FTUI erstellen	47
4.	HTML-Seiten Konfiguration.....	48
5.	Optionale Aufgabe: Müllplan.....	49

1. Setup Raspberry Pi

Um das Setup für den Raspberry Pi einzurichten benötigst du:

- Raspberry Pi
- SD-Karte (mind. 16 GB)
- SD-Karten-Slot am PC

Hast du alles, um das Setup einzurichten, so kannst du der nachfolgenden Anleitung folgen. Du hast aber auch die Möglichkeit einer Videoanleitung zu folgen. Diese findest du auf der Website www.raspberrypi.com/software/ mit dem Schrift-Link „Watch our 45-second video“ im Text unter der Überschrift „Install Raspberry Pi OS using Raspberry Pi Imager“. Bei dem Kopieren des Quellcodes sollte darauf geachtet werden, dass die Seitenzahl des Dokuments nicht mitkopiert wird. Dies könnte zu Fehlern führen.

1.1. Raspberry Pi Imager und Setup

1. Rufe die Website www.raspberrypi.com/software/ auf
2. Lade den Raspberry Pi Imager für dein Betriebssystem (Windows/macOS/Ubuntu for x86) die passende Version herunter.
3. Stecke die SD-Karte in den SD-Kartenleser deines Computers.
4. Installiere den Raspberry Pi Imager, indem du die heruntergeladene .exe-Datei durch einen Doppelklick startest.
 - 4.1. Wähle das Betriebssystem „Raspberry PI OS (32 Bit)“ aus.
 - 4.2. Drücke anschließend die Tasten Strg+shift+x, sodass sich ein weiteres Fenster mit „Erweiterten Optionen“ öffnet.
 - 4.2.1. Setze den Hacken bei „SSH aktivieren“. Wähle ein Passwort für die Authentifizierung aus und lege ein Passwort für den Raspberry Pi fest.

☒ SSH aktivieren

☒ Password zur Authentifizierung verwenden

Passwort für 'pi':

☐ Authentifizierung via Public-Key

authorized_keys für 'pi':

☒ Wifi einrichten

SSID:

Passwort:

☐ Passwort anzeigen

Wifi-Land:

4.2.3. Lege die Spracheinstellungen fest

☒ Spracheinstellungen festlegen

Zeitzone: Europe/Berlin ▼

Tastaturlayout: DE

☒ Einrichtungsassistent überspringen

4.3. Wähle die entsprechende SD-Karte, die du nutzt, aus.

4.4. Führe den Button „Schreiben“ durch Anklicken aus und klicke dich mit der weiter Taste durch die Installation bis du am Ende angekommen bist und der Installationsteil abgeschlossen ist.

4.4.1. Sollten noch Daten auf der SD-Karte vorhanden sein, drücken sie in der Auswahl, dass alle vorherigen Daten gelöscht werden dürfen. Bitte beachten sie, dass nach einer Überschreibung keine Daten wiederhergestellt werden können.

5. Entferne die SD-Karte aus dem SD-Kartenleser deines Computers und stecke sie in den Raspberry Pi. Den Raspberry Pi musst du folgend an das Stromnetz, an einen Bildschirm mit HDMI-Eingang (Computer), sowie vorteilhaft an eine Maus und eine Tastatur anschließen.

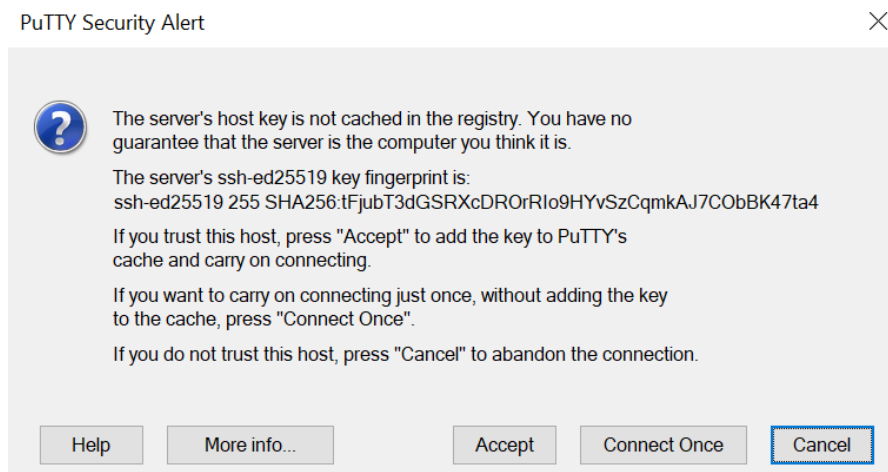
5.1. Warte bis der Raspberry Pi sich mit deinem WLAN verbunden hat.

6. Lade dir auf dem PC/Laptop PuTTY von der Website www.putty.org/ herunter und installiere dies.

7. Starte den PuTTY und verbinde ihn mit dem Raspberry Pi.

8. Trage unter Host Name (or IP address) entweder den Gerätenamen (siehe 8.1) des Raspberry Pi oder die IP-Adresse (siehe 8.2) ein.

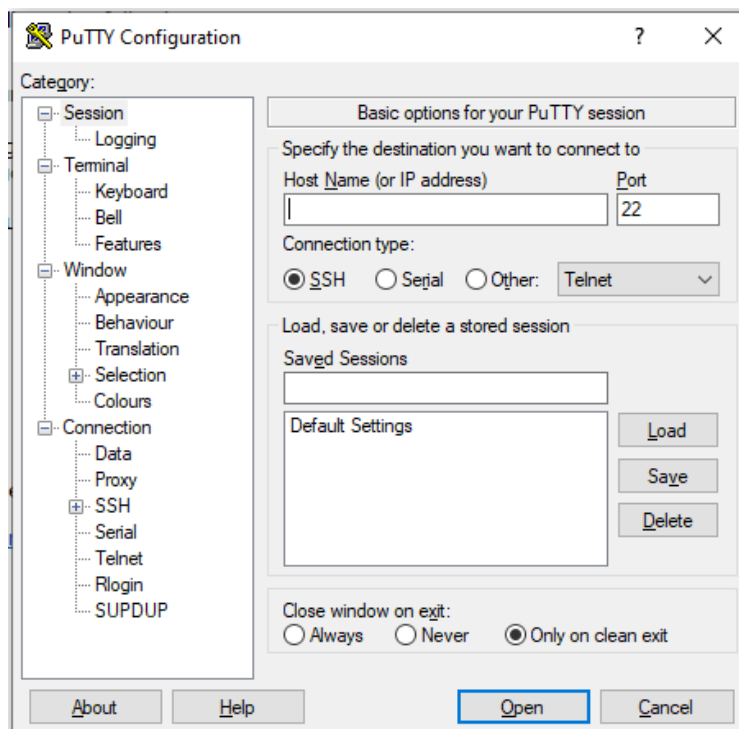
8.1. Gib unter Host Name (or IP address) „raspberrypi“ ein. Folgende Meldung erscheint, welche du mit „Accept“ bestätigen musst.



8.2. Alternativ: Öffne die Kommandozeile auf dem RaspberryPi und führe den Befehl: `ifconfig` aus. Dort kannst du nun die IP-Adresse des Raspberry Pi's ablesen.

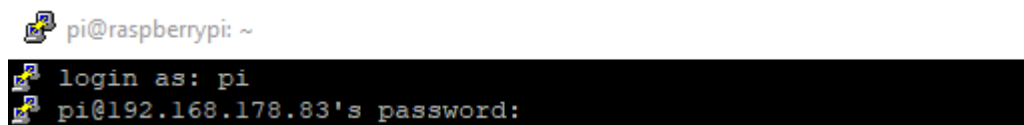
8.3. Trage den Wert 22 unter Port ein.

8.4. Wähle SSH bei Connection type aus.



9. PuTTY-Eingaben:

9.2. Password: Passwort aus Schritt 4.2.1. → Enter



→ Enter

9.4. Eingabe 2: `sudo apt upgrade` → Enter (ggfs. zusätzliche Bestätigung mit „Y“ → Enter)

1.2. FHEM Setup

Wir bieten im Rahmen der Schritt-für-Schritt Anleitung zwei Möglichkeiten an die von der Projektgruppe entwickelte FHEM Konfiguration auf einem Raspberry PI zu installieren. Es besteht die Wahl zwischen einem Setup via Skript, detailliert in 1.2.1 FHEM Setup via Skript beschrieben, oder manuell, wie in 1.2.2 FHEM Setup manuell beschrieben. Das Setup via Skript führt die Installation von FHEM, der erarbeiteten Konfiguration, sowie der Einrichtung von Datenbank und Geräten automatisch aus, wobei nur wenige Eingaben des Benutzers nötig sind. Das Skript arbeitet die Schritte der Schritt-für-Schritt-Anleitung ab. Wird die Installation via Skript gewählt, so muss zumindest das `install.sh` Skript ausgeführt werden. Vorteil der Installation via Skript ist die immense Zeit- und Aufwandsersparnis.

1.2.1. FHEM Setup via Skript

Die Installation via Skript bedingt das Ausführen des Hauptinstallationsskriptes. Ergänzende kleinere Skripte, die zusätzliche Funktionalität installieren, sind bereitgestellt. Diese können nur genutzt werden, wenn das **Hauptinstallationsskript zuvor ausgeführt** worden ist. Zudem ist es notwendig vor der Ausführung der ergänzenden Skripte einige Vorarbeiten zu leisten. Teilweise sind auch interaktive Skripte vorhanden, welche bedient werden müssen. Die Skripte sind darauf ausgelegt genau einmal auf einem Pi ausgeführt zu werden, welcher frisch installiert wurde.

Nach dem Aufsetzen des Pis und dem Herstellen einer SSH Verbindung (siehe Raspberry Pi Imager und Setup) können die Installationsskripte bei AWS wie folgt heruntergeladen und entpackt werden.

Ein neues Verzeichnis an der aktuellen Stelle erstellen und in dieses Wechseln:

```
mkdir Skripte
cd Skripte
```

Das gezippte Archiv herunterladen und entpacken:

```
wget https://fhemchumbucket.s3.eu-central-1.amazonaws.com/install_fhem.tar.gz
tar -xf install_fhem.tar.gz
```

Nun sind 5 Skripte, ein Ordner mit Dateien und das Archiv vorzufinden.

Zu beachten ist, dass das Sicherungskonzept (siehe 1.4) von Hand auszuführen ist!

1.2.1.1. Das Hauptinstallationsskript

Für das Ausführen des Hauptinstallationsskriptes muss der Shelly im Netzwerk angemeldet und seine IP bekannt sein. Dies ist in dem Abschnitt Einrichtung Shelly (siehe Einrichtung Shelly), unter den Punkten 1-8 der ersten Auflistung, beschrieben. Die IP des Shellys ist dem angeschlossenen Router zu entnehmen und beim Start des Skripts **install.sh** als Argument anzugeben:

```
sudo sh install.sh 192.168.178.56
```

Die IP 192.168.178.56 ist hier als Beispiel notiert, diese bitte entsprechend ihrer IP abändern. Nun werden die benötigten Abhängigkeiten, FHEM, die Konfiguration, FTUI mit Dateien, die Datenbank und Shelly heruntergeladen, bzw. installiert. Dies bildet die Grundfunktionalität ab.

1.2.1.2. Die ergänzenden Skripte

Die ergänzenden Skripte benötigen weitere händische Eingaben oder sind komplett interaktiv auszuführen. Die ergänzenden Skripte sind nur ausführbar, wenn das Hauptinstallationskript zuvor ausgeführt wurde.

Installation Skript `install_cul.sh`

Für das Ausführen des Skriptes `install_cul.sh` müssen zwei Argumente angegeben werden, welche in Erfahrung zu bringen sind. Diese sind der Devicename, unter dem der CUL in `/dev/` zu finden ist und der achstellige Hexcode, unter dem das Gerät ansprechbar ist. Der Devicename ist in Flashen und Einrichten der Funksteckdose (siehe Flashen und Einrichten der Funksteckdose) aus der ersten Auflistung bis Punkt 4 zu entnehmen. Der Hexcode ist zu generieren, hier ist vorzugehen wie in der zweiten Auflistung in Flashen und Einrichten der Funksteckdose in Punkt 4 beschrieben ist. Die Befehle der zweiten Auflistung sind nicht einzugeben. Nur der generierte achstellige Hexcode ist als Argument anzureichen. Die Zeichen `0F` am Ende werden im Skript hinzugefügt.

Auszuführen ist somit:

```
sudo sh install_cul.sh TTY000 FF000000
```

TTY000 und FF000000 sind auch hier als Beispiel notiert, diese bitte ersetzen.

Installation Skript `install_anwesenheitserkennung.sh`

In diesem Skript wird die Anwesenheitserkennung von iPhones via FritzBox installiert. Hierfür wird die IP der FritzBox als erstes Argument und die MAC Adresse des iPhones benötigt. Die MAC-Adresse heißt bei Apple WLAN-Adresse. Im Laufe der Installation werden FritzBox Username und Password abgefragt. Näheres hierzu in Betriebsmodus Anwesenheitserkennung (siehe 2.1.6) unter Punkt 3.1.

```
sudo sh install_anwesenheitserkennung.sh 192.168.0.1 XX:YY:AA:SS:DD:FF
```

192.168.0.1 und XX:YY:AA:SS:DD:FF sind auch hier als Beispiel notiert, diese bitte ersetzen.

Installation Skript `install_telegram_bot.sh`

Die Einrichtung des Telegrambots benötigt 2 Argumente. Anzureichen sind der API-Token und der Nutzernamen. Das Kapitel „Einrichten eines Bots zum Senden von Telegram Nachrichten über FHEM“ ist bis zur Nummer vier der Auflistung zu befolgen (siehe Einrichten eines Bots zum Senden von Telegram Nachrichten über FHEM).

```
sudo sh install_telegram_bot.sh  
5086529595:AAGoVMH8AIwA_RCBONvtUI6hft_lYySYmRY Mightyrey
```

Der API-Token und der Name sind auch hier Beispiele.

Installation Skript Install_SSL.sh

Die Installation von SSL/HTTPS ist das kritischste der Skripte. Hier ist sonderlich auf die richtige Eingabe der Daten zu achten. Am besten die Daten parallel notieren, da sie wiederholt eingegeben werden müssen, ohne sich voneinander zu unterscheiden. Ggf. kann die Konfiguration einer CA und HTTPS (siehe 1.5) zur Hilfe und Orientierung herangezogen werden. In diesem Fall sind nur die **roten Textstellen** als Orientierung zu lesen.

```
sudo sh install_SSL.sh
```

Nach der Installation der ergänzenden Skripte ist der gleiche Stand erreicht wie bei einer rein manuellen Installation. Nur das Sicherungskonzept ist noch händisch zu ergänzen.

1.2.2. FHEM Setup manuell

Um das FHEM Setup herzustellen, musst du auf Basis der vorherigen Ausgangssituation weitermachen. Solltest du den PuTTY bereits geschlossen haben, so beginnst du bei Kapitel 1.1 mit dem Schritt 7. Solltest du so weit sein kannst du mit den folgenden Schritten weitermachen.

1. Befehl:


```
apt install -y gpg
```
2. Befehl:


```
sudo wget -O- https://debian.fhem.de/archive.key | gpg --  
dearmor | sudo tee /usr/share/keyrings/debianfhemde-archive-  
keyring.gpg
```
3. Wechsel ins Verzeichnis `"/etc/apt/"`. Hierhin gelangst du mit dem `cd` Befehl (`„cd /etc/apt/“ --<` Enter) ins Verzeichnis.
4. Öffne die Datei `„sources.list“`. Hierfür kannst du den Befehl `„sudo nano sources.list“` nutzen.
 - 4.1. Füge in die geöffnete Datei


```
deb [signed-by=/usr/share/keyrings/debianfhemde-archive-  
keyring.gpg] https://debian.fhem.de/nightly/ / ein.
```
 - 4.2. Speicher die Datei mit der Taste `F3` → Enter
 - 4.3. Schließe die Datei mit `Strg + x`
5. Befehl:

```
sudo apt update
```
6. Befehl:

```
sudo apt install fhem
```

 (ggfs. zusätzliche Bestätigung mit `„Y“` → Enter)

1.2.3. Alternatives FHEM Setup

Bei der Installation wie in 1.2. beschrieben stellte sich beim Testen heraus, dass diese zeitweise unzuverlässig funktionierte. In diesem Fall ist ein alternativer Installationsweg möglich, der folglich beschrieben wird:

Auf dem RaspberryPi mit sudo-Rechten ausführen: (`sudo -i`)

```
apt -y install perl-base libdevice-serialport-perl libwww-perl libio-
socket-ssl-perl libcgi-pm-perl libjson-perl sqlite3 libdbd-sqlite3-perl
libtext-diff-perl libtimedate-perl libmail-imapclient-perl libgd-graph-
perl libtext-csv-perl libxml-simple-perl liblist-moreutils-perl fonts-
liberation libimage-librsvg-perl libgd-text-perl libsocket6-perl libio-
socket-inet6-perl libmime-base64-perl libimage-info-perl libusb-1.0-0-
dev libnet-server-perl libdate-manip-perl libhtml-treebuilder-xpath-
perl libmojolicious-perl libxml-bare-perl libauthen-oath-perl
libconvert-base32-perl libmodule-pluggable-perl libnet-bonjour-perl
libcrypt-urandom-perl nodejs npm libnet-dbus-perl wget libfuse2 libffi-
dev make gcc-avr avrdude avr-libc
```

Download des Pakets mit `wget`, danach Installation mit `dpkg`. Im Anschluss wird ein User namens `fhem` erstellt. Der `Copy`-Befehl kopiert eine Service Datei in die Systemumgebung um die Steuerung mit `Systemctl` zu ermöglichen. Der Daemon lädt anschließend die Datei ein via dem `reload`.

```
wget http://fhem.de/fhem-6.1.deb
```

```
dpkg -i fhem-6.1.deb
```

```
useradd --system --home /opt/fhem --gid dialout --shell /bin/false fhem
```

```
cp /opt/fhem/contrib/init-scripts/fhem.service /etc/systemd/system/
```

```
systemctl daemon-reload
```

1.3. FTUI Setup

Um das FTUI Setup herzustellen kannst du der Anleitung auf der Website https://wiki.fhem.de/wiki/FHEM_Tablet_UI folgen oder du folgst den unten beschriebenen Schritten.

1. Öffne die FHEM-Befehlszeile. Kopiere nacheinander in der vorgegebenen Reihenfolge die Befehle, füge sie in die FHEM-Befehlszeile ein und führe sie aus.
 - 1.1. Befehl: „update all https://raw.githubusercontent.com/knowthelist/fhem-tablet-ui/master/controls_fhemtabletui.txt“
 - 1.2. Befehl: `define TABLETUI HTTPSrv ftui/ ./www/tablet/ Tablet-UI`
 - 1.3. Befehl: `attr WEB longpoll websocket` (ggf. Beispieldatei mit dem Befehl: „`sudo cp -a /opt/fhem/www/tablet/index-example.html /opt/fhem/www/tablet/index.html` kopieren und auf dem Pi über PuTTY ausführen)
 - 1.4. Änderungen speichern mit „Save config“
2. Starte FHEM neu und füge den Befehl `shutdown restart` zur Ausführung in die FHEM-Befehlszeile ein.
3. Nach Neustart ist FTUI über den Link in der FHEM-Oberfläche im Browser oder über **Fehler! Linkreferenz ungültig.** erreichbar

1.4. Sicherungskonzept

Sicherung des gesamten Pi in die Dropbox einrichten

Alle Befehle werden als root ausgeführt (zu Beginn einmal `sudo -i`)!!

1. `apt-get install libfuse2`
2. `pip3 install dbxfs`
3. Verzeichnis anlegen
 - 3.1. `cd /mnt/`
 - 3.2. `mkdir mydropbox`
4. Verzeichnis mounten
 - 4.1. `dbxfs /mnt/mydropbox`
 - 4.2. Link nach „Go to“ kopieren und im Browser öffnen
 - 4.3. Mit Dropbox anmelden
 - 4.4. Authorization code kopieren und im Terminal-Fenster einfügen
 - 4.5. Mit Enter bestätigen
5. Backup Script anlegen
 - 5.1. Verzeichnis anlegen
 - 5.1.1. `mkdir /home/fhem/scripts/`
 - 5.1.2. `nano`
 - 5.2. Script anlegen
 - 5.2.1. Einfügen:

```
#!/bin/bash
```

```
# Define dates for today and 28 days (4 weeks ago)
```

```
today=$(date +%Y-%m-%d)
```

```
date2delete=$(date --date="$today -28 day" +%Y-%m-%d)
```

```
# Echo the defined dates
```

```
echo "$today"
```

```
echo "$date2delete"
```

```
# Create dd backup
```

```
sudo dd if=/dev/mmcblk0p2 of=/mnt/mydropbox/DHBW-  
Pi/backup-$today
```

```
# Compress backup file
```

```
sudo tar -cpzf /mnt/mydropbox/DHBW-Pi/backup-  
$today.tar.gz --one-file-system /mnt/mydropbox/DHBW-  
Pi/backup-$today
```

```
# Remove uncompressed backup
```

```
sudo rm /mnt/mydropbox/DHBW-Pi/backup-$today
```

```
# Remove 28 day old backup
```

```
sudo rm /mnt/mydropbox/DHBW-Pi/backup-$date2delete.tar.gz
```

5.2.2. Speichern mit „Strg + O“

5.2.3. „backup.sh“ als Name eingeben und mit Enter bestätigen

5.2.4. Mit Strg + X den Editor verlassen

5.3. Script ausführbar machen `chmod +x backup.sh`

5.4. Cronjob anlegen

5.4.1. `crontab -e`

5.4.2. Am Ende der Datei einfügen:

```
0 0 * * * bash /home/fhem/scripts/backup.sh
```

5.4.3. Mit Strg + O speichern

5.4.4. Mit Strg + X verlassen

Nun wird jeden Tag um 0:00 Uhr ein Backup in der Dropbox erstellt, welches nach vier Wochen (28 Tagen) beim nächsten Backup wieder gelöscht wird.

1.5. Konfiguration einer CA und HTTPS

Das Einrichten von Zertifikaten und einer Certification Authority (CA) ist durch den Ablauf der folgenden Zeilen zu erreichen. Die genutzten Programme agieren hierbei interaktiv, die getätigten Eingaben sind beispielhaft und können abgeändert werden. **Eingaben des Benutzers** sind farblich markiert. Gerade bei der Angabe von Daten beim Erstellen von Zertifikaten ist darauf zu achten valide Angaben zu machen, welche sich im Optimalfall nicht unterscheiden. Die Angaben wiederholen sich teilweise. **Beispielweise Angaben sind ebenfalls markiert.** Beim Generieren des Serverkeys ist die Angabe der IP des Pis wichtig! Diese kann identifiziert werden via "sudo hostname -I", die erste IP ist einzutragen. In der openssl.cnf sind Angaben abzuändern. Die Datei ist im Zielzustand abgebildet und so abzuändern. Anleitungen des FHEM-Wiki sind hier ausführlicher und aktueller als der Dargestellte Ablauf, die Links sind an entsprechender Stelle eingefügt. Das Password redbull ist als Beispiel wichtig and letzterer Stelle erneut genau so einzugeben. Das passwordEgal ist egal und muss nur kurz nach dem setzen nochmals eingegeben werden um sich selbst zu deaktivieren.

https://wiki.fhem.de/wiki/FHEM_mit_HTTPS_SSL-Zertifikat_und_eine_eigene_Zertifizierungsstelle

Verbinden mit dem PI via SSH.

```
pi@raspberrypi:~# sudo -i
```

```
root@raspberrypi:~/ca# apt install libio-socket-ssl-perl -y
```

```
root@raspberrypi:~/ca# apt install libwww-perl -y
```

```
root@raspberrypi:~# mkdir /root/ca
```

```
root@raspberrypi:~# cd /root/ca
```

```
root@raspberrypi:~/ca# openssl req -new -x509 -newkey rsa:2048 -keyout  
fhemSSL.pem -out cacert.pem -days 3650
```

Generating a RSA private key

```
.....+++++
```

```
.....
```

```
.....+++++
```

writing new private key to 'fhemSSL.pem'

Enter PEM pass phrase: #redbull

Verifying - Enter PEM pass phrase: #redbull

```
-----
```

You are about to be asked to enter information that will be incorporated

into your certificate request.

What you are about to enter is what is called a Distinguished Name or a DN.

There are quite a few fields but you can leave some blank

For some fields there will be a default value,

If you enter '.', the field will be left blank.

Country Name (2 letter code) [AU]:DE

State or Province Name (full name) [Some-State]:Baden-Württemberg

Locality Name (eg, city) []:Stuttgart

Organization Name (eg, company) [Internet Widgits Pty Ltd]:DHBW

Organizational Unit Name (eg, section) []:WWI2019D

Common Name (e.g. server FQDN or YOUR name) []:Gruppe2

Email Address []:Gruppe2.WWI2019D@DHBW.de

```
root@raspberrypi:~/ca# ls -la
```

```
total 16
```

```
drwx----- 7 root root 4096 Jan  5 11:54 ..
```

```
-rw----- 1 root root 1854 Jan  5 11:56 fhemSSL.pem
```

```
-rw-r--r-- 1 root root 1489 Jan  5 11:57 cacert.pem
```

```
drwxr-xr-x 2 root root 4096 Jan  5 11:57 .
```

```
root@raspberrypi:~/ca# chmod 600 fhemSSL.pem
```

```
root@raspberrypi:~/ca# openssl genrsa -out serverkey.pem -aes128 2048
```

Generating RSA private key, 2048 bit long modulus (2 primes)

```
.....
```

```
.....+++++
```

```
.....+++++
```

e is 65537 (0x010001)

Enter pass phrase for serverkey.pem: #passwordEgal

Verifying - Enter pass phrase for serverkey.pem: #passwordEgal

```
root@raspberrypi:~/ca# openssl rsa -in serverkey.pem -out serverkey.pem
```

Enter pass phrase for serverkey.pem: #passwordEgal

```
root@raspberrypi:~/ca# openssl req -new -key serverkey.pem -out req.pem
-nodes
```

You are about to be asked to enter information that will be incorporated

into your certificate request.

What you are about to enter is what is called a Distinguished Name or a DN.

There are quite a few fields but you can leave some blank

For some fields there will be a default value,

If you enter '.', the field will be left blank.

Country Name (2 letter code) [AU]:DE

State or Province Name (full name) [Some-State]:Baden-Württemberg

Locality Name (eg, city) []:Stuttgart

Organization Name (eg, company) [Internet Widgits Pty Ltd]:DHBW

Organizational Unit Name (eg, section) []:WWI2019D

Common Name (e.g. server FQDN or YOUR name) []:192.168.178.81

Email Address []:admin@192.168.178.81

Please enter the following 'extra' attributes

to be sent with your certificate request

A challenge password []:

An optional company name []:

```
root@raspberrypi:~/ca# ls -la
```

```
total 24
```

```
drwx----- 7 root root 4096 Jan  5 11:54 ..
```

```
-rw----- 1 root root 1854 Jan  5 11:56 fhemSSL.pem
```

```
-rw-r--r-- 1 root root 1489 Jan  5 11:57 cacert.pem
```

```
-rw----- 1 root root 1679 Jan  5 12:05 serverkey.pem
```

```
-rw-r--r-- 1 root root 1082 Jan  5 12:11 req.pem
```



```
drwxr-xr-x 2 root root 4096 Jan  5 12:11 .
```

```
root@raspberrypi:~/ca# nano /etc/ssl/openssl.cnf
```

```
[ CA_default ]
```

```
dir                = /root/ca                # Where everything is kept
certs              = $dir/certs              # Where the issued certs are kept
crl_dir            = $dir/crl                # Where the issued crl are kept
database           = $dir/index.txt         # database index file.
#unique_subject    = no                     # Set to 'no' to allow creation of
                                                # several certs with same
subject.
new_certs_dir      = $dir                    # default place for new certs.

certificate        = $dir/cacert.pem         # The CA certificate
serial             = $dir/serial             # The current serial number
crlnumber          = $dir/crlnumber          # the current crl number
                                                # must be commented out to
leave a V1 CRL
crl                = $dir/crl.pem            # The current CRL
private_key        = $dir/fhemSSL.pem        # The private key

RANDFILE           = $dir/.rand

x509_extensions    = usr_cert               # The extensions to add to the
cert

# Comment out the following two lines for the "traditional"
# (and highly broken) format.

name_opt           = ca_default             # Subject Name options
```

```
cert_opt          = ca_default          # Certificate field options
```

```
# Extension copying option: use with caution.
```

```
# copy_extensions = copy
```

```
# Extensions to add to a CRL. Note: Netscape communicator chokes on V2 CRLs
```

```
# so this is commented out by default to leave a V1 CRL.
```

```
# crlnumber must also be commented out to leave a V1 CRL.
```

```
# crl_extensions      = crl_ext
```

```
default_days      = 3650                # how long to certify for
```

```
default_crl_days= 30                    # how long before next CRL
```

```
default_md        = default             # use public key default MD
```

```
preserve          = no                  # keep passed DN ordering
```

```
# A few difference way of specifying how similar the request should look
```

```
# For type CA, the listed attributes must be the same, and the optional
```

```
# and supplied fields are just that :-)
```

```
policy            = policy_match
```

```
Hier Speichern via Strg + O und schließen mit Strg + X.
```

```
root@raspberrypi:~/ca# echo 01 > serial
```

```
root@raspberrypi:~/ca# touch index.txt
```

```
root@raspberrypi:~/ca# openssl ca -in req.pem -notext -out  
servercert.pem
```

```
Enter pass phrase for ./cakey.pem: #redbull
```

```
...
```

```
Certificate is to be certified until Jan 05 10:45:36 2032 GMT (3650
days)
```

```
Sign the certificate? [y/n]: y
```

```
1 out of 1 certificate requests certified, commit? [y/n] y
```

```
Write out database with 1 new entries
```

```
Data Base Updated
```

```
-----
-----
```

Hier kopieren wir eine Auswahl der generierten Dateien in ein Verzeichnis von FHEM unter /opt/fhem/ und passen die Zugriffsrechte an.

```
https://wiki.fhem.de/wiki/Raspberry_Pi_%26_HTTPS
```

```
root@raspberrypi:~/ca# mkdir /opt/fhem/certs
```

```
root@raspberrypi:~/ca# cd /opt/fhem/certs
```

```
root@raspberrypi:~/ca# cp serverkey.pem /opt/fhem/certs
```

```
root@raspberrypi:~/ca# cp servercert.pem /opt/fhem/certs
```

```
root@raspberrypi:~/ca# sudo chown -R fhem:dialout /opt/fhem/certs/
```

```
-----
-----
```

Anschließend sind zwei Zeilen auf der Oberfläche von FHEM einzugeben, um HTTPS zu aktivieren:

In FHEMWEB:

```
attr WEB sslVersion TLSv12:!SSLv3
```

```
attr WEB HTTPS 1
```

Ab nun ist die Oberfläche mit https://... aufrufen. Manche Browser bereiten Probleme beim Aufruf von selbstsignierten Zertifikaten. In diesem Fall ist hier eine Handlungsempfehlung aufgeführt: [https://wiki.fhem.de/wiki/FHEM mit HTTPS SSL-Zertifikat und eine eigene Zertifizierungsstelle#Einbinden der root CA in die Zertifizierungsstelle de s Browsers](https://wiki.fhem.de/wiki/FHEM_mit_HTTPS_SSL-Zertifikat_und_eine_eigene_Zertifizierungsstelle#Einbinden_der_root_CA_in_die_Zertifizierungsstelle_de_s_Browsers) Die entsprechende Datei cacert.pm lässt sich via scp auf einen Windows-PC kopieren.

Hierfür: Auf dem PI:

```
Sudo cp /root/ca/cacert.pm /home/pi/
```

```
Sudo chown pi:pi /home/pi/cacert.pm
```

Auf einem Windows PC in der PowerShell oder CMD mit der IP des PIs(`sudo hostname -I`):

```
Scp pi@IP://home/pi/cacert.pm .
```

Die Datei wird nach Eingabe des Passworts des Users PI an die Stelle kopiert, an welcher sich der User in PowerShell/CMD momentan befindet. Wenn nicht verändert bei Start: `C:/Users/<user>/`

2. Implementation in FHEM

2.1. Funktion Weihnachtsbeleuchtung

2.1.1. Flashen und Einrichten der Funksteckdose

Auf dem Raspberry Pi:

1. Arduino in USB des Pi einstecken
2. Befehl `sudo tail -f /var/log/syslog` → STRG + C
3. Befehl `sudo tail -f /var/log/message` → STRG + C
4. Nach rot markierten Werten (sollte derselbe sein) suchen und merken

```
pi@raspberrypi:~$ sudo tail -f /var/log/syslog
Dec 10 11:13:56 raspberrypi mtp-probe: bus: 1, device: 4 was not an MTP device
Dec 10 11:13:56 raspberrypi kernel: [589540.813696] usbcore: registered new interface driver usbserial_generic
Dec 10 11:13:56 raspberrypi kernel: [589540.815838] usbserial: USB Serial support registered for generic
Dec 10 11:13:56 raspberrypi kernel: [589540.821840] usbcore: registered new interface driver ch341
Dec 10 11:13:56 raspberrypi kernel: [589540.823660] usbserial: USB Serial support registered for ch341-uart
Dec 10 11:13:56 raspberrypi kernel: [589540.823801] ch341 1-1.4:1.0: ch341-uart converter detected
Dec 10 11:13:56 raspberrypi kernel: [589540.830384] usb 1-1.4: ch341-uart converter now attached to ttyUSB0
Dec 10 11:13:56 raspberrypi mtp-probe: checking bus 1, device 4: "/sys/devices/platform/soc/3f980000.usb/usb1/1-1/1-1.4"
Dec 10 11:13:56 raspberrypi mtp-probe: bus: 1, device: 4 was not an MTP device
Dec 10 11:14:14 raspberrypi systemd[1]: Started Session 190 of user pi.
^C
pi@raspberrypi:~$ sudo tail -f /var/log/messages
Dec 10 11:16:21 raspberrypi kernel: [589686.483219] usb 1-1.2: new full-speed USB device number 5 using dwc_otg
Dec 10 11:16:21 raspberrypi kernel: [589686.616384] usb 1-1.2: New USB device found, idVendor=1a86, idProduct=7523, bcdDevice= 2.54
Dec 10 11:16:21 raspberrypi kernel: [589686.616411] usb 1-1.2: New USB device strings: Mfr=0, Product=2, SerialNumber=0
Dec 10 11:16:21 raspberrypi kernel: [589686.616426] usb 1-1.2: Product: USB2.0-Serial
Dec 10 11:16:21 raspberrypi kernel: [589686.621343] ch341 1-1.2:1.0: ch341-uart converter detected
Dec 10 11:16:21 raspberrypi kernel: [589686.631819] usb 1-1.2: ch341-uart converter now attached to ttyUSB0
Dec 10 11:16:21 raspberrypi mtp-probe: checking bus 1, device 5: "/sys/devices/platform/soc/3f980000.usb/usb1/1-1/1-1.2"
Dec 10 11:16:21 raspberrypi mtp-probe: bus: 1, device: 5 was not an MTP device
Dec 10 11:16:22 raspberrypi mtp-probe: checking bus 1, device 5: "/sys/devices/platform/soc/3f980000.usb/usb1/1-1/1-1.2"
Dec 10 11:16:22 raspberrypi mtp-probe: bus: 1, device: 5 was not an MTP device
```

5. Befehl `wget http://culfw.de/culfw-1.67.tar.gz`
6. Befehl `tar xfv culfw-1.67.tar.gz`
7. Befehl `sudo apt-get install make gcc-avr avrdude avr-libc`
8. Befehl `cd culfw-1.67/Devices/nanoCUL/`
9. Befehl `make`
10. Befehl `make program`

In FHEM:

1. Befehl `define nanoCUL CUL /dev/ttyUSB0@38400 1234` (den roten Teil durch das Gemerkte aus Schritt 4 ersetzen)
2. Befehl `attr nanoCUL rfmode HomeMatic`
3. Befehl `attr nanoCUL room 241_Weihnachtsbeleuchtung`
4. Befehl `define funkLamp IT FF000000FF FF F0` (den roten Teil entsprechend des Drehschalters auf der Rückseite der Funksteckdose und https://wiki.fhem.de/wiki/Intertechno_Code_Berechnung#Original_Intertechno_System ersetzen. Auf der Funksteckdose ist hinten ein Drehschalter, mit dem die Frequenz ausgewählt werden kann. Diese kann frei gewählt werden. Diese wird durch eine Kombination aus Buchstaben + Zahl dargestellt und haben einen entsprechenden vierstelligen Code in der Tabelle haben. Die ersten vier Stellen des roten Teils werden durch die Code-Entsprechung des Buchstabens und die zweiten vier Stellen durch die Code-Entsprechung der Zahl ersetzt.)
5. Befehl `attr funkLamp IODev nanoCUL`
6. Befehl `attr funkLamp model itswitch`

7. Befehl `attr funkLamp room 241_Weihnachtsbeleuchtung`

Funksteckdose in eine Steckdose stecken und innerhalb von 5 Sekunden in FHEM beim Gerät „funkLamp“ auf „on“ klicken (siehe Screenshot)

2.1.2. Manueller Betrieb

Mit der Einrichtung der Funksteckdose und des CUL ist die manuelle Bedienung, bis auf die Verbindung zum Frontend, bereits gegeben.

2.1.3. Betriebsmodus Feste Betriebszeit

1. Erstellen eines Erweiterungsmoduls „at“, um die Funklampe um 7 Uhr auszuschalten:

```
define a_funkLamp_morning_off at *07:00:00 set funkLamp off
```

2. Raum zuweisen:

```
attr a_funkLamp_morning_off room 241_Weihnachtsbeleuchtung
```

3. Erstellen eines weiteren Erweiterungsmoduls „at“, um die Funklampe um 17 Uhr einzuschalten:

```
define a_funkLamp_evening_on at *17:00:00 set funkLamp on
```

4. Raum zuweisen:

```
attr a_funkLamp_evening_on room 241_Weihnachtsbeleuchtung
```

2.1.4. Betriebsmodus Sonnenaufgang und Sonnenuntergang

1. Koordinaten in FHEM hinterlegen (für die Daten von Sonnenauf- und Sonnenuntergang wird der Standort benötigt)

Unter `Unsorted` das Attribut `global` auswählen

Dort die Attribute `latitude` und `longitude` aus den gewählten Koordinaten hinzufügen

```
attr global latitude 48.773486
```

```
attr global longitude 9.170658
```

latitude	48.773486
logfile	./log/fhem-%Y-%m.log
longitude	9.170658

```
Define doif_funk_sonne doif ([{sunset("REAL")}]) (set funkLamp on)DOELSEIF ([{sunrise("REAL")}]) (set funkLamp off)
```

```
attr doif_funk_sonne room 241_Weihnachtsbeleuchtung
```

```
attr doif_funk_sonne disable 1
```

2.1.5. Betriebsmodus Sonnenaufgang und Sonnenuntergang mit Uhrzeiten

1. Einen DOIF-Typen erstellen ("doif_funk_szeit"), damit die Lampe bei Sonnenaufgang aus- und bei Sonnenuntergang eingeschaltet wird. Die Lampe geht jedoch spätestens um 7:30 Uhr aus.

```
define doif_funk_szeit DOIF ([{sunset("REAL")}]) (set funkLamp on)DOELSEIF ([{sunrise_abs("REAL",0,"00:00","07:30")}]) (set funkLamp off)"
```

```
attr doif_funk_szeit room 241_Weihnachtsbeleuchtung
```

```
attr doif_funk_szeit disable 1
```

2.1.6. Betriebsmodus Anwesenheitserkennung

Einrichtung der FritzBox in FHEM:

1. IP-Adresse der FritzBox herausfinden (am besten über die Weboberfläche der FritzBox, häufig 192.168.178.1)
2. Skript einfügen
 - 2.1. In FHEM links auf "Edit files"
 - 2.2. Öffnen von "myUtilsTemplate.pm"
 - 2.3. Inhalt ersetzen durch den Inhalt von 99_myUtils.txt, welches unter <https://github.com/doenisf/HomeAutomationProjektGruppe2> zu finden ist in dem UtilFiles Ordner
 - 2.4. Im Textfeld neben "Save as" "99_myUtils.pm" eintragen und mit Klick auf "Save as" speichern
3. In FHEM:

```
define FritzBox FRITZBOX <IP>
```

- 3.1 Bei gewissen FritzBoxen sind zusätzliche Schritte notwendig. Tests lassen die Vermutung zu, dass Provider wie Unitymedia Funktionalität blockieren. Sollte das Modul eine Meldung aufzeigen wie etwa "Keine Session ID" sind folgende Eingaben zu tätigen. In der FritzBox muss ein User angelegt werden mit Nutzernamen und Password.

```
attr FritzBox boxUser <username>
```

```
set FritzBox password <password>
```

Ggf. Kann es auch notwendig sein das Attribut allowTr064Command auf 1 zu setzen:

```
attr FritzBox allowTR064Command 1
```

4. Raum zuweisen:


```
attr FritzBox room 99_Fritz
```
5. Herausfinden der MAC-Adresse des iPhones (am iPhone)
 - 4.1. Einstellungen -> "Allgemein" -> "Info" -> "WLAN-Adresse"
 - 4.2. Unter IOS "WLAN-Adresse" = "MAC-Adresse"
 - 4.3. MAC-Adresse merken (Format xx:xx:xx:xx:xx:xx)

6. In FHEM:

```
define iPhone PRESENCE function {checkAllFritzMACpresent("<MAC-Adresse>")} 01 01
```

<MAC-Adresse> ersetzen durch die Adresse aus Schritt 4b

7. Die Schritte 4. und 5. können für beliebig viele weitere iPhones wiederholt werden, es muss nur in Schritt 5. der Name des Geräts (hier "iPhone") geändert werden.
8. Skript zur Anwesenheitserkennung (und Subroutine für 2.1.7 Betriebsmodusauswahl) einfügen
 - 7.1. In FHEM links auf "Edit files"
 - 7.1.1. "myUtilsTemplate.pm" öffnen
 - 7.1.2. Inhalt ersetzen durch den Inhalt von 99_myUtils_241.txt, welches unter <https://github.com/doenisf/HomeAutomationProjektGruppe2> zu finden ist in dem UtilFiles Ordner

7.1.3. Im Textfeld neben "Save as" "99_myUtils_241.pm" eintragen und mit Klick auf "Save as" speichern

9. In FHEM:

```
define a_funkLamp_anwesenheitsCheck at +*00:00:01  
{modusAnwesenheit("funkLamp", 17, 7)}
```

Raum zuweisen:

```
attr a_funkLamp_anwesenheitsCheck room 241_Weihnachtsbeleuchtung
```

2.1.7. Betriebsmodusauswahl

Diese Funktion verwendet eine Subroutine aus 99_myUtils_241.pm. Das Einfügen dieser Funktion wird unter 2.1.6 beschrieben.

Allgemeines:

1. Einen DUMMY-Typen erstellen, der für die Übermittlung des Betriebsmodus vom Frontend ins Backend fungiert.

```
define funkLampModus dummy

attr funkLampModus room 241_Weihnachtsbeleuchtung

attr funkLampModus          setList
switch:festeZeit,Sonne,SZeit,Anwesenheit,Manuell
```

2. Einen DOIF-Typen ("doif_funk_general") erstellen, um die verschiedenen Betriebsmodi anzutriggern.

```
define doif_funk_general DOIF ([funkLampModus:"festeZeit"])(attr
doif_funk_sonne disable 1)(attr doif_funk_szeit disable 1)(attr
a_funkLamp_anwesenheitsCheck disable 1)(attr a_funkLamp_evening_on
disable 0)(attr a_funkLamp_morning_off disable
0)({checkTime("funkLamp", 17, 7)})DOELSEIF
([funkLampModus:"Sonne"])(attr doif_funk_sonne disable 0)(attr
doif_funk_szeit disable 1)(attr a_funkLamp_anwesenheitsCheck
disable 1)(attr a_funkLamp_evening_on disable 1)(attr
a_funkLamp_morning_off disable 1)({checkTime("funkLamp",
sunset_abs(), sunrise_abs())})DOELSEIF
([funkLampModus:"SZeit"])(attr doif_funk_sonne disable 1)(attr
doif_funk_szeit disable 0)(attr a_funkLamp_anwesenheitsCheck
disable 1)(attr a_funkLamp_evening_on disable 1)(attr
a_funkLamp_morning_off disable 1)({checkTime("funkLamp",
sunset_abs(), sunrise_abs("REAL",0,"00:00","07:30"))})DOELSEIF
([funkLampModus:"Anwesenheit"])(attr doif_funk_sonne disable
1)(attr doif_funk_szeit disable 1)(attr
a_funkLamp_anwesenheitsCheck disable 0)(attr a_funkLamp_evening_on
disable 1)(attr a_funkLamp_morning_off disable
1)DOELSEIF([funkLampModus:"Manuell"])(attr doif_funk_sonne disable
1)(attr doif_funk_szeit disable 1)(attr
a_funkLamp_anwesenheitsCheck disable 1)(attr a_funkLamp_evening_on
disable 1)(attr a_funkLamp_morning_off disable 1)

attr doif_funk_general room 241_Weihnachtsbeleuchtung
```

2.2. Funktion Sturmwarnung

2.2.1. Einrichten eines Bots zum Senden von Telegram Nachrichten über FHEM

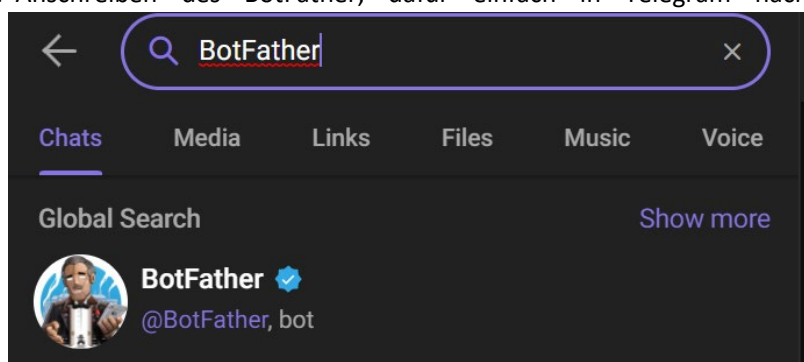
Installation von Telegram auf seinem Smartphone. Einrichten eines Telegram Bots, Telegrambot in FHEM einrichten, um darüber an bestimmte Personennachrichten zu senden.

1. Installation von Telegram auf deinem Smartphone, dafür kann die Telegram App einfach über den Apple Store oder Google Play Store heruntergeladen werden.



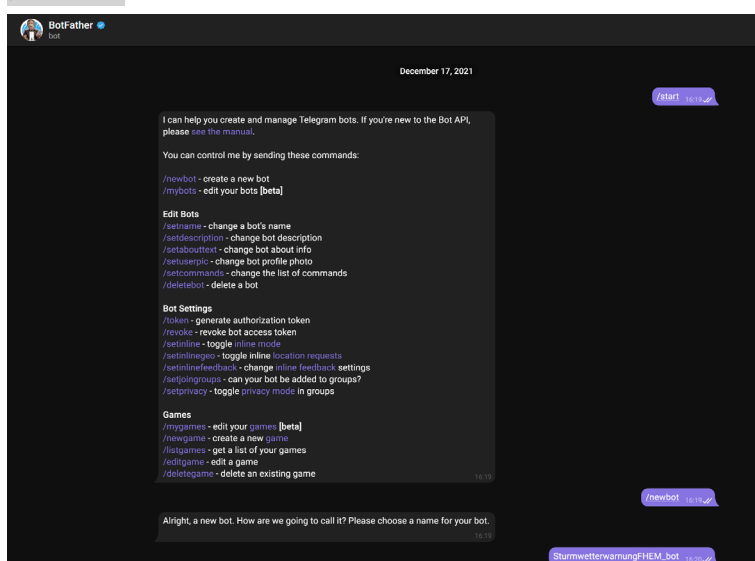
3. Telegramm auf dem Handy öffnen oder Telegram Web verwenden.

- 3.1. Anschreiben des BotFather, dafür einfach in Telegram nach "BotFather" suchen

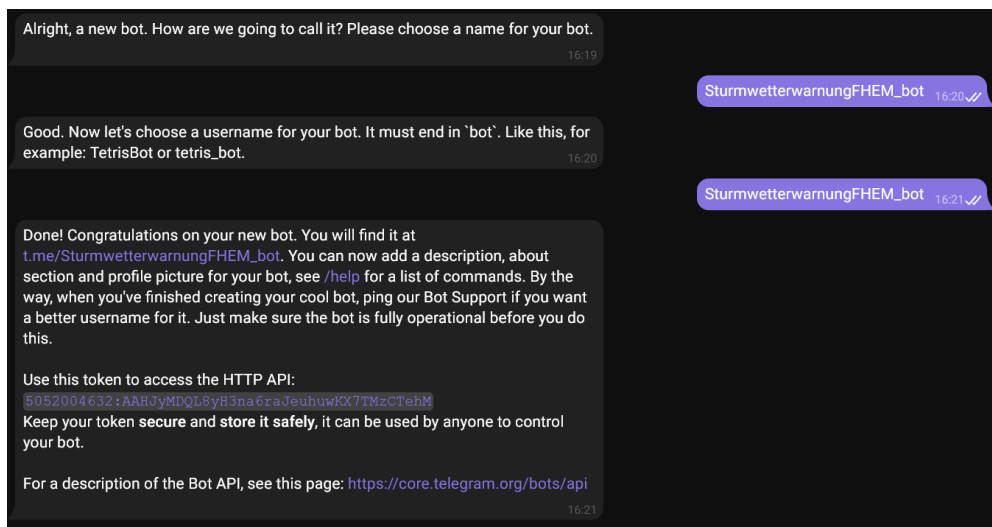


/start

/newbot



- 3.3. Nun vergeben sie den Namen ihres Bots, achten sie dabei darauf, dieser Name darf noch nicht vergeben sein und sollte auf "bot" enden. Sie bekommen dann einen HTTP API Token, dieser ist für die weitere Einrichtung in FHEM sehr wichtig. Bitte merken sie sich den Token und geben sie diesen nicht weiter!



4. Einrichtung Telegram Bot in FHEM

Botname: `SturmweatherwarnungFHEM_bot`

Token HTTP API: `5052004632:AAHJyMDQL8yH3na6raJeuhwKX7TMzCTehM`

Dazu geben sie den Befehl in FHEM ein:

```
define <name> TelegramBot <token>
```

Mit unserem Bot wäre das:

```
define SturmweatherwarnungFHEM_bot TelegramBot
5086529595:AAGoVMH8AIwA_RCBONvtUI6hft_lYySYmRY
```

5. Einstellung des Attributes pollingTimeout und dem Raum "Sturmweatherwarnung" zuordnen.:

Polling Attribut:

```
attr <name> pollingTimeout 120
```

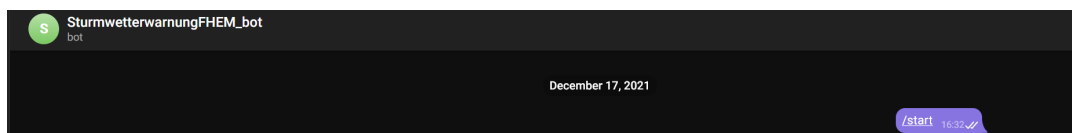
Raum Attribut:

```
attr <name> room 242_Sturmweatherwarnung
```

Mit unserem Bot wäre das:

```
attr SturmweatherwarnungFHEM_bot pollingTimeout 120
attr SturmweatherwarnungFHEM_bot room 242_Sturmweatherwarnung
```

6. Um den Bot zu starten und damit dieser ihnen eine Nachricht schreiben kann, müssen sie diesen über Telegram eine Nachricht senden.



DeviceOverview

SturmweatherwarnungFHEM_bot

Polling

set SturmweatherwarnungFHEM_bot

_msg

get SturmweatherwarnungFHEM_bot

peerId

Internals

FAILS

0

FUUIID

61bcab8e-f33f-c687-aefb-7ad864ca8e01b573

NAME

SturmweatherwarnungFHEM_bot

NR

42

OLD_POLLING

9

POLLING

9

SNAME

SturmweatherwarnungFHEM_bot

STATE

Polling

TYPE

TelegramBot

UPDATER

0

WAIT

0

me

5052004632:SturmweatherwarnungFHEM_bot:@SturmweatherwarnungFHEM_bot

sourceVersion

\$Id: 50_TelegramBot.pm 24867 2021-08-23 10:23:15Z viegener \$

Readings

Contacts

1279895775:Melissa_Negele:@XXXXXXXXXX

2021-12-17 16:32:41

PollingErrCount

0

2022-01-17 00:00:40

PollingLastError

NonBlockingGet timed out on read from <hidden> after 245s

2022-01-11 21:44:10

msgChat

Melissa_Negele

2021-12-17 16:32:41

msgChatId

1279895775

2021-12-17 16:32:41

msgFileId

2021-12-17 16:32:41

msgId

1

2021-12-17 16:32:41

msgPeer

Melissa_Negele

2021-12-17 16:32:41

msgPeerId

1279895775

2021-12-17 16:32:41

msgReplyMsgId

2021-12-17 16:32:41

msgText

/start

2021-12-17 16:32:41

sentMsgId

22

2021-12-25 22:47:44

sentMsgPeerId

1279895775

2021-12-25 22:47:44

sentMsgResult

SUCCESS

2021-12-25 22:47:44

```
attr <Botname> defaultPeer @<Ihr Benutzername>
```

Bei uns wäre dies:

```
attr SturmweatherwarnungFHEM_bot defaultPeer @XXXXXX
```

- Um neue Benutzer über die FHEM UI Oberfläche hinzuzufügen, müssen die neuen Benutzer zuerst den TelegramBot anschreiben, danach können sie über die Oberfläche hinzugefügt werden. Sollten sie einen anderen Bot Namen verwenden als den vorgegebenen, müssen sie gegeben falls für die Anzeige der Benutzernamen auch die Frontend Datei ändern und dort den vergebenen Bot Namen durch den von ihnen neu angelegten ersetzen.



2.2.2. Sensordaten auslesen

Verbindung zwischen FHEM und Sensor Hochdorf herstellen

Weitere Beschreibung:

Anlegen eines neuen Devices und auslesen der Windgeschwindigkeit sowie der Böen-Stärke von der Seite:
<https://measurements.mobile-alerts.eu/Home/SensorsOverview?phoneid=285142992122>

1. Neues Device "SensorDaten" erzeugen und Verbindung zwischen FHEM und Sensor Hochdorf über das HTTPMOD Modul herstellen. Eine Prüfung der Verbindung findet im Intervall 60 sek. statt.

```
define SensorDaten HTTPMOD https://measurements.mobile-
alerts.eu/Home/SensorsOverview?phoneid=285142992122 60
```

2. HTTPMOD ruft den Sensor Hochdorf über einen eigenen, leider stark veralteten Browser auf, dieser kann moderne Webseiten nicht mehr anzeigen. Deswegen ist es nötig zu sagen, HTTPMOD soll die Webseite über den neueren Mozilla Browser aufrufen.

```
attr SensorDaten reading01Regex requestHeader1
attr SensorDaten requestHeader1 User-Agent: Mozilla/5.0 (Windows
NT 6.0)
```

3. Auslesen der Windgeschwindigkeit vom Sensor Hochdorf über das HTTPMOD Modul

```
attr SensorDaten reading02Name Windgeschwindigkeit
attr SensorDaten reading02Regex
Windgeschwindigkeit[\S\s\r\n]*?>([\-\d\d\d,\d]+)
attr SensorDaten reading02OExpr join ".", (split /,/, $val)
```

4. Auslesen der Böen-Stärke vom Sensor Hochdorf über das HTTPMOD Modul

```
attr SensorDaten reading03Name Boe
attr SensorDaten reading03Regex B&#246;[\S\s\r\n]*?>([\-\d\d\d,\d]+)
attr SensorDaten reading03OExpr join ".", (split /,/, $val)
```

5. Auslesen der Windrichtung vom Sensor Hochdorf über das HTTPMOD Modul

```
attr SensorDaten reading04Name Windrichtung
attr SensorDaten reading04Regex
Windrichtung[\S\s\r\n]*?>(?|(Norde)|(Nordnordost)|(Nordost)|(Ostno
rdost)|(Ost)|(Ostsüdost)|(Südost)|(Südsüdost)|(Süd)|(Südsüdwest)|
(Südwest)|(Westwärts)|(West)|(Westnordwest)|(Nordwest)|(Nordnordw
est))
```

6. Dem SensorDaten einem Raum zuweisen

```
attr SensorDaten room 242_Sturmweatherwarnung
```

7. Optional: Icon in FHEM hinzufügen

```
attr SensorDaten icon weather_wind_speed@SkyBlue
```

DeviceOverview
SensorDaten
???

set
SensorDaten
attr:Template
?

Internals

BUSY	0
DEF	https://measurements.mobile-alerts.eu/Home/SensorsOverview?phoneid=285142992122 60
FUUIID	61b77018-f33f-e209-1359-c496b865e120b175
Interval	60
MainURL	https://measurements.mobile-alerts.eu/Home/SensorsOverview?phoneid=285142992122
ModuleVersion	4.1.10 - 6.7.2021
NAME	SensorDaten
NOTIFYDEV	global
NR	22
NTFY_ORDER	50-SensorDaten
STATE	???
TYPE	HTTPMOD
value	

Readings

Böe	3,2	2021-12-15 19:32:23
Windgeschwindigkeit	1,1	2021-12-15 19:32:23
Windrichtung	West Südwest	2021-12-15 19:32:23

attr
SensorDaten
room

Attributes

reading01Regex requestHeader1	deleteattr
reading02Name Windgeschwindigkeit	deleteattr
reading02Regex Windgeschwindigkeit[S\s/r/n]*?>[^\s\d\, d +)	deleteattr
reading03Name Böe	deleteattr
reading03Regex Bö[S\s/r/n]*?>[^\s\d\, d +)	deleteattr
reading04Name Windrichtung	deleteattr
reading04Regex Windrichtung[S\s/r/n]*?>(?(Norde) (Nordnordost) (Nordost) (Ostnordost) (Ost) (Ostsüdost) (Südost) (Südsüdost) (Süd) (Südsüdwest) (Südwest) (West Südwest) (West) (Westnordwest) (Nordwest) (Nordnordwest))	deleteattr
requestHeader1 User-Agent: Mozilla/5.0 (Windows NT 6.0)	deleteattr

[Select icon](#)
[Extend devStateIcon](#)
[Raw definition](#)
[Delete this device \(SensorDaten\)](#)
[Device specific help](#)

2.2.3. Anlegen der Dummy-Devices für die Eingabe der Schwellenwerte und für den Schalter der Sturmweatherwarnung

Um die Schwellenwerte eingeben und die Sturmwarnung ein- und ausschalten zu können, werden Dummy-Devices angelegt. Bis auf ein Attribut beim Schalter für die Sturmwarnung müssen keine weiteren Attribute oder Readings angelegt werden.

Anlegen des Dummy-Devices für den Schwellwert der Windböe; Dazu muss folgender Befehl in FHEM ausgeführt werden:

Anlegen des Dummy-Devices für den Schwellwert der Windgeschwindigkeit und der Böen-Stärke; Dazu muss folgender Befehl in FHEM ausgeführt werden:

```
define Schwellwert_Windstaerke dummy
attr Schwellwert_Windstaerke room 242_Sturmweatherwarnung
set Schwellwert_Windstaerke 200
```

```
define Schwellwert_Windboe dummy
attr Schwellwert_Windboe room 242_Sturmweatherwarnung
set Schwellwert_Windboe 200
```

Anlegen des Dummy-Devices für den Schalter der Sturmwarnung. Dazu müssen folgende Befehle nacheinander in FHEM ausgeführt werden:

```
define SturmwarnungSchalter dummy
attr SturmwarnungSchalter room 242_Sturmweatherwarnung
attr SturmwarnungSchalter webCmd on:off
define MerkerSchalterBoe dummy
attr MerkerSchalterBoe room 242_Sturmweatherwarnung
attr MerkerSchalterBoe webCmd on:off
setreading MerkerSchalterBoe state off
define MerkerSchalterWindstaerke dummy
attr MerkerSchalterWindstaerke room 242_Sturmweatherwarnung
attr MerkerSchalterWindstaerke webCmd on:off
setreading MerkerSchalterWindstaerke state off
```


2.2.4. Hinzufügen von Do-If-Abfragen

Aufpassen, hier muss der bei einem neu angelegten Bot dieser in den DO-If-Abfragen ersetzt werden durch den Namen, unter dem sie ihren Bot in FHEM angelegt haben. (Rot markiert)

1. Anlegen eines Devices vom Typ DOIF für das Überschreiten des Windstärke Schwellenwerts:

```
define UeberschreitenSchwellenwertWindstaerke DOIF
([Schwellwert_Windstaerke:state] < [SensorDaten:Windgeschwindigkeit]
and [SturmwarnungSchalter:state] eq "on" and
[MerkerSchalterWindstaerke:state] eq "off") (set
SturmweatherwarnungFHEM_bot message Achtung! Windgeschwindigkeit wurde
überschritten. Aktuelle Windgeschwindigkeit:
[SensorDaten:Windgeschwindigkeit] km/h) (set MerkerSchalterWindstaerke
on)
```

```
attr UeberschreitenSchwellenwertWindstaerke room
242_Sturmweatherwarnung
```

```
attr UeberschreitenSchwellenwertWindstaerke do always
```

2. Anlegen eines Devices vom Typ DOIF für das Überschreiten des Böen Schwellenwerts:

```
define UeberschreitenSchwellenwertBoe DOIF
([Schwellwert_Windboe:state] < [SensorDaten:Boe] and
[SturmwarnungSchalter:state] eq "on" and [MerkerSchalterBoe:state] eq
"off") (set SturmweatherwarnungFHEM_bot message Achtung!
Böengeschwindigkeit wurde überschritten. Aktuelle
Böengeschwindigkeit: [SensorDaten:Boe] km/h) (set MerkerSchalterBoe
on)
```

```
attr UeberschreitenSchwellenwertBoe room 242_Sturmweatherwarnung
```

```
attr UeberschreitenSchwellenwertBoe do always
```

3. Anlegen eines Devices vom Typ DOIF für das Entwarnen (Böe):

```
define UnterschreitenSchwellenwertBoe DOIF
([Schwellwert_Windboe:state] > ([SensorDaten:Boe]*1.3) and
[SturmwarnungSchalter:state] eq "on" and [MerkerSchalterBoe:state] eq
"on") (set SturmweatherwarnungFHEM_bot message Die Wetterlage hat sich
beruhigt. Aktuelle Böengeschwindigkeit: [SensorDaten:Boe] km/h) (set
MerkerSchalterBoe off)
```

```
attr UnterschreitenSchwellenwertBoe room 242_Sturmweatherwarnung
```

```
attr UnterschreitenSchwellenwertBoe do always
```

4. Anlegen eines Devices vom Typ DOIF für das Entwarnen (Windstärke):

```
define UnterschreitenSchwellenwertWindstaerke DOIF
([Schwellwert_Windstaerke:state] >
([SensorDaten:Windgeschwindigkeit]*1.3) and
[SturmwarnungSchalter:state] eq "on" and
[MerkerSchalterWindstaerke:state] eq "on") (set
```

```
SturmweatherwarnungFHEM_bot message Die Wetterlage hat sich beruhigt.
Aktuelle Windgeschwindigkeit: [SensorDaten:Windgeschwindigkeit]
km/h) (set MerkerSchalterWindstaerke off)
```

```
attr UnterschreitenSchwellenwertWindstaerke room
242_Sturmweatherwarnung
```



```
attr UnterschreitenSchwellenwertWindstaerke do always
```

5. Anlegen eines Devices vom Typ DOIF für die Logik hinter Windstärke/Böenstärke

```
define doif_kombinationWindBoe DOIF ([SturmwarnungSchalter:state] eq
"off") (set MerkerSchalterBoe off) (set MerkerSchalterWindstaerke
off) (attr UeberschreitenSchwellenwertBoe disable 1) (attr
UeberschreitenSchwellenwertWindstaerke disable 1) (attr
UnterschreitenSchwellenwertBoe disable 1) (attr
UnterschreitenSchwellenwertWindstaerke disable 1) DOELSEIF
([SturmwarnungSchalter:state] eq "on") (attr
UeberschreitenSchwellenwertBoe disable 0) (attr
UeberschreitenSchwellenwertWindstaerke disable 0) (attr
UnterschreitenSchwellenwertBoe disable 0) (attr
UnterschreitenSchwellenwertWindstaerke disable 0)
```

```
attr doif_kombinationWindBoe room 242_Sturmweatherwarnung
```

Anbei ein Screenshot aller angelegten Devices im Raum 242_Sturmweatherwarnung;

DOIF			
UeberschreitenSchwellenwertBoe	cmd_2		
UeberschreitenSchwellenwertWindstaerke	cmd_2		
UnterschreitenSchwellenwertBoe	cmd_2		
UnterschreitenSchwellenwertWindstaerke	cmd_2		
HTTPMOD			
 SensorDaten	???		
TelegramBot			
SturmweatherwarnungFHEM_bot	Polling		
dummy			
MerkerSchalterBoe	???	on	off
MerkerSchalterWindstaerke	???	on	off
Schwellwert_Windboe	999		
Schwellwert_Windstaerke	55		
SturmwarnungSchalter		on	off

2.3. Funktion Photovoltaik/Balkonkraftwerk

2.3.1. Einrichtung Shelly

Shelly mit WiFi verbinden

1. Shelly Cloud App runterladen
2. Shelly Cloud Account erstellen und/oder anmelden
3. Raum in Shelly Cloud App erstellen
4. Smartphone mit WiFi-Access Point des Shelly verbinden
 - 4.1. Access Point startet entweder beim Verbinden des Shelly mit dem Strom oder alternativ durch 10 Sekunden gedrückt halten des Buttons auf der Rückseite des Shelly (Shelly ist in der Unterputzdose)
5. In Shelly Cloud App neues Device hinzufügen
6. Im Schritt `Include new devices in your wi-fi network` Name (SSID) sowie Passwort des Netzwerks eingeben, in das der Shelly eingebunden werden soll
7. In Dropdown-Liste „Shelly25“ auswählen
8. Name für Shelly vergeben und mit Shelly Cloud verbinden

Shelly in FHEM einbinden:

INFO: der zweite Kanal des Shelly ist auf Grund begrenzter Hardware doppelt belegt. Zum Testen des kombinierten Betriebes von Wasserboiler und Infrarotheizung kann eines der beiden Geräte auf den anderen Kanal umgestellt werden (`attr <gerätename> defchannel 0`). Im tatsächlichen Gebrauch würden mehrere Shellys zum Einsatz kommen. Diese werden auf die gleiche Art eingerichtet, bis auf die entsprechend andere IP-Adresse im Befehl.

1. Erster Channel (Balkonkraftwerk) einbinden

- 1.1. Befehl in FHEM-Kommandozeile `define balkonkraftwerk Shelly <Shelly-IP>`
(IP des Shelly kann über den Router des Netzwerks rausgefunden werden)
- 1.2. Befehl `attr balkonkraftwerk model shelly2.5`
- 1.3. Befehl `attr balkonkraftwerk mode relay`
- 1.4. Befehl `attr balkonkraftwerk defchannel 0`
- 1.5. Befehl `attr balkonkraftwerk room 243_Balkonkraftwerk`

2. Zweiten Channel (Infrarotheizung) einbinden

- 2.1. Befehl in FHEM-Kommandozeile `define infrarotheizung Shelly <Shelly-IP>` IP des Shelly kann über den Router des Netzwerks rausgefunden werden)
 Befehl `attr infrarotheizung model shelly2.5`
 Befehl `attr infrarotheizung mode relay`
 Befehl `attr infrarotheizung defchannel 1`
 Befehl `attr infrarotheizung room 243_Balkonkraftwerk`

3. Zweiten Channel erneut (Wasserboiler) einbinden

- 3.1. Befehl in FHEM-Kommandozeile `define wasserBoiler Shelly <Shelly-IP>` (IP des Shelly kann über den Router des Netzwerks rausgefunden werden)

- 3.2. Befehl `attr wasserBoiler model shelly2.5`
- 3.3. Befehl `attr wasserBoiler mode relay`
- 3.4. Befehl `attr wasserBoiler defchannel 1`
- 3.5. Befehl `attr wasserBoiler room 243_Wasserboiler`

2.3.2. Einrichten einer DB zum Speichern von Daten

Die Daten werden in einer SQLite DB gespeichert, welche noch installiert werden muss. Dafür mit dem PI verbinden und auf der Shell folgende Abhängigkeiten installieren:

```
sudo apt install -y sqlite3 libdbi-perl libdbd-sqlite3-perl
```

Es lässt sich nun eine DB erstellen und aufrufen via:

```
sudo sqlite3 /opt/fhem/fhem.db
```

Anschließend findet man sich in der sqlite Console wieder, in diese folgende Zeilen einfügen und mit Enter abschicken, jeweils bis zum Semikolon:

```
CREATE TABLE history (TIMESTAMP TIMESTAMP NOT NULL DEFAULT
CURRENT_TIMESTAMP, DEVICE varchar(64), TYPE varchar(64), EVENT
varchar(512), READING varchar(64), VALUE varchar(128), UNIT
varchar(32));
```

```
CREATE TABLE current (TIMESTAMP TIMESTAMP, DEVICE varchar(64),
TYPE varchar(64), EVENT varchar(512), READING varchar(64), VALUE
varchar(128), UNIT varchar(32));
```

```
CREATE INDEX Search_Idx ON `history` (DEVICE, READING, TIMESTAMP);
```

Die sqlite Konsole schließt sich mit dem Befehl:

```
.exit
```

Ändern der Rechte und Zugriffsmodifikationen:

```
sudo chown fhem /opt/fhem/fhem.db
```

```
sudo chmod 600 /opt/fhem/fhem.db
```

Anlegen einer Datei, über die die DB Konfiguriert wird:

```
sudo nano /opt/fhem/db.conf
```

In nano den folgenden Textblock einfügen, anschließend mit Strg + O speichern und mit Strg + X schließen:

```
%dbconfig=(
connection => "SQLite:dbname=/opt/fhem/fhem.db",
user => "",
password => ""
);
```

Zum Abschluss der Arbeiten auf dem PI wird FHEM neu gestartet:

```
sudo systemctl restart fhem
```

In FHEM Werden nun noch Readings angelegt und die DB vorstellig gemacht. Dazu wie gewohnt in die Befehlszeile von FHEM eingeben:

```
setreading balkonkraftwerk sendData [balkonkraftwerk:power_0]
define logdb DbLog /opt/fhem/db.conf balkonkraftwerk:sendData.*
define timerSetReading at +*00:01:00 setreading balkonkraftwerk
sendData [balkonkraftwerk:power_0]
```

Speichern via Save.

Funktionalität, welche Daten akkumuliert und in die DB speichert wird durch DBrep bereitgestellt. Das Modul wird definiert für Wochen und Monate:

```
define repBalkonkraftwerkDaily DbRep logdb
attr repBalkonkraftwerkDaily reading sendData
attr repBalkonkraftwerkDaily aggregation day
attr repBalkonkraftwerkDaily device balkonkraftwerk
attr repBalkonkraftwerkDaily timeDiffToNow d:1 FullDay
define repDaily at *23:59:59 set repBalkonkraftwerkDaily sumValue
writeToDBSingle
```

```
define repBalkonkraftwerkMonthly DbRep logdb
attr repBalkonkraftwerkMonthly reading sendData
attr repBalkonkraftwerkMonthly aggregation month
attr repBalkonkraftwerkMonthly device balkonkraftwerk
attr repBalkonkraftwerkMonthly timeDiffToNow d:31 FullDay
define repMonthly at *23:59:59 {if ((strftime "%d",localtime
time+86400) eq "01") {set repBalkonkraftwerkMonthly sumValue
writeToDBSingle}}
```

Anschließend Speichern.

2.3.3. Einfaches Heizen

1. In das Device "balkonkraftwerk" Readings hinzufügen.

```
setreading balkonkraftwerk energieSchwellwert 300
```

2. Anlegen eines DOIFs, das in einem regelmäßigen Zeitabstand prüft, ob die erzeugte Energie den Schwellenwert überschreitet bzw. unterschreitet und die Infrarotheizung demnach ein- oder ausschaltet

Dazu folgende Befehle ausführen:

```
define doif_UeberschreitenEnergieSchwellenwert DOIF
([+([doif_UeberschreitenEnergieSchwellenwert:intervall]*60)] and
[balkonkraftwerk:power_0] >
[balkonkraftwerk:energieSchwellwert] and [infrarotheizung:relay_1]
eq "off") (set infrarotheizung on) DOELSEIF
([+([doif_UeberschreitenEnergieSchwellenwert:intervall]*60)] and
[balkonkraftwerk:power_0] < [balkonkraftwerk:energieSchwellwert]
and [infrarotheizung:relay_1] eq "on") (set infrarotheizung off)

attr doif_UeberschreitenEnergieSchwellenwert room
243_Balkonkraftwerk

attr doif_UeberschreitenEnergieSchwellenwert disable 1

"etreading doif_UeberschreitenEnergieSchwellenwert intervall 5

attr doif_UeberschreitenEnergieSchwellenwert disable 0

attr doif_UeberschreitenEnergieSchwellenwert do always

attr doif_UeberschreitenEnergieSchwellenwert disable 1
```

3. Um den Überhitzungsschutz umzusetzen, werden zwei DOIFs benötigt.

Der erste schaltet die Infrarotheizung aus, wenn diese eine bestimmte Zeit ununterbrochen läuft.

Dazu folgende Befehle ausführen:

```
define doif_HeizungEinfach DOIF ([infrarotheizung:relay_1] eq
"on") (set infrarotheizung off) (attr
doif_UeberschreitenEnergieSchwellenwert disable 1) (setreading
doif_HeizungEinfach disable 1)

attr doif_HeizungEinfach room 243_Balkonkraftwerk

setreading doif_HeizungEinfach disable 1

setreading doif_HeizungEinfach on_time 120

attr doif_HeizungEinfach do resetwait

attr doif_HeizungEinfach wait [doif_HeizungEinfach:on_time]*60

attr doif_HeizungEinfach disable 1
```

Das zweite DOIF stellt sicher, dass die Heizung für eine bestimmte Zeit ausgeschaltet bleibt.

Dazu folgende Befehle ausführen:

```
define doif_disableTimerEinfach DOIF
([doif_HeizungEinfach:disable] eq "1") (attr
doif_UeberschreitenEnergieSchwellenwert disable 0) (setreading
doif_HeizungEinfach disable 0)

attr doif_disableTimerEinfach room 243_Balkonkraftwerk"

attr doif_disableTimerEinfach disable 1

setreading doif_disableTimerEinfach disabled
{AttrVal("doif_UeberschreitenEnergieSchwellenWert","disable",2)}
```

-> wenn 2 zurückgegeben wird, existiert das Attribut nicht

```
setreading doif_disableTimerEinfach off_time 15

attr doif_disableTimerEinfach do resetwait

attr doif_disableTimerEinfach wait
[doif_disableTimerEinfach:off_time]*60
```

2.3.4. Optimiertes Heizen

2.3.4.1. Sensordaten auslesen

Verbindung zwischen FHEM und Sensor Hobbyraum herstellen

Weitere Beschreibung:

Anlegen eines neuen Device's und auslesen der Temperatur sowie des Zeitstempels von der Seite:
<https://measurements.mobile-alerts.eu/Home/MeasurementDetails?deviceid=035DD29D94C5&vendorid=244DD836-16DE-465E-B265-B3F1596A26D4&appbundle=de.synertronixx.remotemonitor>

1. Neues Device "Temperatursensor" erzeugen und Verbindung zwischen FHEM und Sensor Hobbyraum Messung über das HTTPMOD Modul herstellen. Eine Prüfung der Verbindung findet im Intervall 420sec. Statt (Der Sensor aktualisiert seine Zeit nur alle 7min, deswegen ist eine mehrfache Prüfung nicht notwendig).

```
define Temperatursensor HTTPMOD https://measurements.mobile-alerts.eu/Home/MeasurementDetails?deviceid=035DD29D94C5&vendorid=244DD836-16DE-465E-B265-B3F1596A26D4&appbundle=de.synertronixx.remotemonitor 420
```

2. HTTPMOD ruft den Sensor Hochdorf über einen eigenen, leider stark veralteten Browser auf, dieser kann moderne Webseiten nicht mehr anzeigen. Deswegen ist es nötig zu sagen, HTTPMOD soll die Webseite über den neueren Mozilla Browser aufrufen.

```
attr Temperatursensor reading01Regex requestHeader1
attr Temperatursensor requestHeader1 User-Agent: Mozilla/5.0 (Windows NT 6.0)
```

3. Cookies auf der Webseite aktivieren

```
attr Temperatursensor enableCookies 1
```

4. Auslesen der Temperatur vom Sensor Hobbyraum Messung über das HTTPMOD Modul

```
attr Temperatursensor reading02Name temperature
attr Temperatursensor reading02Expr join ".", (split /,/, $val)
attr Temperatursensor reading02Regex ([^>]\d*[,\]\d)
```

5. Auslesen des Zeitstempels vom Sensor Hobbyraum Messung über das HTTPMOD Modul

```
attr Temperatursensor reading03Name timestamp
attr Temperatursensor reading03Regex ([^>]\d+[.]\d+[.]\d+[s]\d+[:]\d+[:]\d+)
```

6. Auslesen der Luftfeuchtigkeit vom Sensor Hobbyraum Messung über das HTTPMOD Modul

2.3.4.2. Logik Optimierte Heizen

Anlegen von Dummy-Devices für die Eingabe der Schwellwerte und für den Schalter für das Optimierte Heizen

Die Schwellwerte für das Aus- und Einschalten kommen dabei einmal vom Temperatursensor und von der FHEM UI Übersicht.

Anlegen der Dummy-Devices für das Ein und Ausschalten des Optimierten Heizen; Dazu müssen folgende Befehle nacheinander in FHEM ausgeführt werden:

1. `define OptimiertesHeizenSchalter dummy`
`attr OptimiertesHeizenSchalter room 243_Balkonkraftwerk`
`attr OptimiertesHeizenSchalter webCmd on:off`
2. `define Schwellwert_Temperatur_Ein dummy`
`attr Schwellwert_Temperatur_Ein room 243_Balkonkraftwerk`
`set Schwellwert_Temperatur_Ein 19`
3. `define Schwellwert_Temperatur_Aus dummy`
`attr Schwellwert_Temperatur_Aus room 243_Balkonkraftwerk`
`set Schwellwert_Temperatur_Aus 21`
4. `define Schwellwert_Luftfeuchtigkeit_Ein dummy`
`attr Schwellwert_Luftfeuchtigkeit_Ein room 243_Balkonkraftwerk`
`set Schwellwert_Luftfeuchtigkeit_Ein 75`
5. `define Schwellwert_Luftfeuchtigkeit_Aus dummy`
`attr Schwellwert_Luftfeuchtigkeit_Aus room 243_Balkonkraftwerk`
`set Schwellwert_Luftfeuchtigkeit_Aus 70`

2.3.4.3. Hinzufügen von Do-If-Abfragen

Anlegen eines Devices vom Typ DOIF für das Einschalten der Infrarotheizung nach unterschreiten der Schwellwert Temperatur oder überschreiten des Schwellwertes Luftfeuchtigkeit

```
define EinschaltenOptimiertesHeizen DOIF
([Schwellwert_Temperatur_Ein:state] > [Temperatursensor:temperature]
or [Schwellwert_Luftfeuchtigkeit_Ein:state] <
[Temperatursensor:humidity]) and [OptimiertesHeizenSchalter:state] eq
"on" and [+([doif_UeberschreitenEnergieSchwellenwert:intervall]*60)]
and [balkonkraftwerk:power_0] > [balkonkraftwerk:energieSchwellwert]
and [infrarotheizung:relay_1] eq "off") (set infrarotheizung on)
```

```
attr EinschaltenOptimiertesHeizen room 243_Balkonkraftwerk
attr EinschaltenOptimiertesHeizen do always
setreading EinschaltenOptimiertesHeizen EnergyTimeRange 60
attr EinschaltenOptimiertesHeizen disable 1
```

Anlegen eines Devices vom Typ DOIF für das Ausschalten der Infrarotheizung nach überschreiten des Schwellwert Temperatur oder unterschreiten des Schwellwertes Luftfeuchtigkeit

```
define AusschaltenOptimiertesHeizen DOIF
([Schwellwert_Temperatur_Aus:state] < [Temperatursensor:temperature]
and [Schwellwert_Luftfeuchtigkeit_Aus:state] >
[Temperatursensor:humidity] and [OptimiertesHeizenSchalter:state] eq
"on" and [+([doif_UeberschreitenEnergieSchwellenwert:intervall]*60)])
(set infrarotheizung off)
```

```
attr AusschaltenOptimiertesHeizen room 243_Balkonkraftwerk
attr AusschaltenOptimiertesHeizen do always
setreading AusschaltenOptimiertesHeizen EnergyTimeRange 60
attr AusschaltenOptimiertesHeizen disable 1
```

Um den Überhitzungsschutz umzusetzen, werden zwei DOIFs benötigt.

Der erste schaltet die Infrarotheizung aus, wenn diese eine bestimmte Zeit ununterbrochen läuft.

Dazu folgende Befehle ausführen:

```
define doif_HeizungOptimierend DOIF ([infrarotheizung:relay_1] eq
"on") (set infrarotheizung off) (attr EinschaltenOptimiertesHeizen
disable 1) (setreading doif_HeizungOptimierend disable 1)
attr doif_HeizungOptimierend room 243_Balkonkraftwerk
setreading doif_HeizungOptimierend disable 1
attr doif_HeizungOptimierend do resetwait
attr doif_HeizungOptimierend wait [doif_HeizungEinfach:on_time]*60
```

```
attr doif_HeizungOptimierend disable 1
```

Das zweite DOIF stellt sicher, dass die Heizung für eine bestimmte Zeit ausgeschaltet bleibt.

Dazu folgende Befehle ausführen:

```
define doif_disableTimerOptimierend DOIF
([doif_HeizungOptimierend:disable] eq "1") (attr
EinschaltenOptimiertesHeizen disable 0) (setreading
doif_HeizungOptimierend disable 0)
```

```
attr doif_disableTimerOptimierend room 243_Balkonkraftwerk"
```

```
attr doif_disableTimerOptimierend disable 1"
```

```
setreading doif_disableTimerOptimierend disabled
{AttrVal("doif_UeberschreitenEnergieSchwellenWert","disable",2)}
```

-> wenn 2 zurückgegeben wird, existiert das Attribut nicht

```
attr doif_disableTimerOptimierend do resetwait
```

```
attr doif_disableTimerOptimierend wait
[doif_disableTimerEinfach:off_time]*60
```

2.3.4.4. Betriebsmodullogik Heizen

1. Erstellen eines dummy zur Kommunikation des ausgewählten Betriebsmodus vom Frontend ins Backend.

```
define heizenModus dummy

attr heizenModus room 243_Balkonkraftwerk

attr heizenModus setList switch:einfach,optimierend,aus
```

2. Erstellen eines DOIF zur Steuerung der Betriebsmodi.

```
define doif_heizenModus DOIF ([heizenModus:"einfach"]) (attr
doif_UeberschreitenEnergieSchwellenwert disable 0) (attr
doif_HeizungEinfach disable 0) (attr doif_disableTimerEinfach
disable 0) (attr doif_HeizungOptimierend disable 1) (attr
doif_disableTimerOptimierend disable 1) (attr
EinschaltenOptimiertesHeizen disable 1) (attr
AusschaltenOptimiertesHeizen disable 1) (set
OptimiertesHeizenSchalter off) DOELSEIF
([heizenModus:"optimierend"]) (attr EinschaltenOptimiertesHeizen
disable 0) (attr AusschaltenOptimiertesHeizen disable 0) (attr
doif_HeizungEinfach disable 1) (attr doif_disableTimerEinfach
disable 1) (attr doif_HeizungOptimierend disable 0) (attr
doif_disableTimerOptimierend disable 0) (set
OptimiertesHeizenSchalter on) (attr
doif_UeberschreitenEnergieSchwellenwert disable 1) DOELSEIF
([heizenModus:"aus"]) (attr
doif_UeberschreitenEnergieSchwellenwert disable 1) (attr
EinschaltenOptimiertesHeizen disable 1) (attr
AusschaltenOptimiertesHeizen disable 1) (attr doif_HeizungEinfach
disable 1) (attr doif_disableTimerEinfach disable 1) (attr
doif_HeizungOptimierend disable 1) (attr
doif_disableTimerOptimierend disable 1) (set infrarotheizung off)
(set OptimiertesHeizenSchalter off)

attr doif_heizenModus room 243_Balkonkraftwerk
```

2.3.5. Warmwasserboiler

2.3.5.1. Manueller Betrieb

Mit der Einrichtung des Shelly-Device wasserBoiler ist das manuelle Steuern bis auf die Anbindung ans Frontend bereits gewährleistet.

2.3.5.2. Vorgegebener Zeitraum

1. Skript einfügen

1.2 In FHEM links auf "Edit files"

1.3 Öffnen von "myUtilsTemplate.pm"

1.4 Inhalt ersetzen durch den Inhalt von 99_myUtils_243.txt, welches unter <https://github.com/doenisf/HomeAutomationProjektGruppe2> zu finden ist in dem UtilFiles Ordner

1.5 Öffne Putty auf deinem Rechner (falls du es nicht mehr offen hast, siehe 1.1)

1.6 In dem Terminal den Befehl: „sudo cpan Time::Piece“ ausführen und anschließend mit yes bestätigen

1.7 Im Textfeld neben "Save as" "99_myUtils_243.pm" eintragen und mit Klick auf "Save as" speichern

```
Define a_boiler_zeit_on at *00:00:10 set wasserBoiler on
```

```
Define a_boiler_zeit_off at *00:00:10 set wasserBoiler off
```

```
Setreading a_boiler_zeit_on datum <dd-mm-yyyy>
```

```
à Datum am Ende muss in der Zukunft liegen
```

```
Setreading a_boiler_zeit_off datum <dd-mm-yyyy>
```

```
à Datum am Ende muss in der Zukunft liegen
```

```
Modify a_boiler_zeit_on *{parseDate("a_boiler_zeit_on")} set  
wasserBoiler on
```

```
attr a_boiler_zeit_on room 243_Wasserboiler
```

```
Modify a_boiler_zeit_off *{parseDate("a_boiler_zeit_off")} set  
wasserBoiler off
```

```
attr a_boiler_zeit_off room 243_Wasserboiler
```

```
define doif_update_a_boiler_zeit_on_and_off DOIF ([+60])
```

```
(modify a_boiler_zeit_off *{parseDate("a_boiler_zeit_off")} set  
wasserBoiler off)(modify a_boiler_zeit_on  
*{parseDate("a_boiler_zeit_on")} set wasserBoiler on)
```

```
attr doif_update_a_boiler_zeit_on_and_off do always
```

```
attr doif_update_a_boiler_zeit_on_and_off room 243_Wasserboiler
```

2.3.5.3. Stromwerttabelle

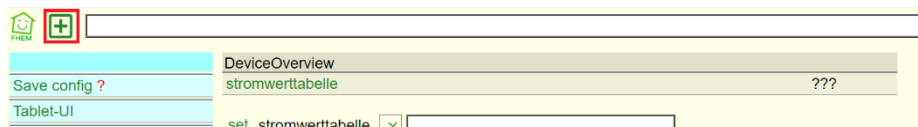
1. Dummy einrichten zum Speichern der Werte

1.1. In FHEM:

```
define stromwerttabelle dummy
attr stromwerttabelle room 243_Wasserboiler
```

1.2. Readings einrichten

1.2.1. In FHEM oben links auf das "+"-Icon



1.2.2. Folgendes einfügen

```
setreading stromwerttabelle t1 0
setreading stromwerttabelle t1_on 00:00
setreading stromwerttabelle t1_off 00:00
setreading stromwerttabelle t2 1
setreading stromwerttabelle t2_on 00:00
setreading stromwerttabelle t2_off 00:00
setreading stromwerttabelle t3 2
setreading stromwerttabelle t3_on 00:00
setreading stromwerttabelle t3_off 00:00
setreading stromwerttabelle t4 1
setreading stromwerttabelle t4_on 00:00
setreading stromwerttabelle t4_off 00:00
setreading stromwerttabelle t5 2
setreading stromwerttabelle t5_on 00:00
setreading stromwerttabelle t5_off 00:00
setreading stromwerttabelle t6 1
setreading stromwerttabelle t6_on 00:00
setreading stromwerttabelle t6_off 00:00
setreading stromwerttabelle we 0
setreading stromwerttabelle we_on 00:00
setreading stromwerttabelle we_off 00:00
```

1.2.3. Auf "Execute" drücken



2. In FHEM:

```
define a_boiler_stromwert at +*00:00:10 {modusStromwertttabelle()}
attr a_boiler_stromwert room 243_Wasserboiler
```

2.3.5.4. Schwellwert

```
Define doif_boiler_schwellwert DOIF
([+([doif_boiler_schwellwert:intervall]*60)]
and([balkonkraftwerk:power_0] > [wasserBoiler:schwellwert]))(set
wasserBoiler on)DOELSEIF ([+([doif_boiler_schwellwert:intervall]*60)]
and([balkonkraftwerk:power_0] < [wasserBoiler:schwellwert]))(set
wasserBoiler off)

Setreading wasserBoiler schwellwert 400

attr doif_boiler_schwellwert room 243_Wasserboiler
```


2.3.5.5. Unterstützende Funktion

```
Define doif_wasserboiler_general DOIF
([wasserBoilerModus:"manuell"])(attr a_boiler_zeit_on disable 1)(attr
a_boiler_zeit_off disable 1)(attr doif_boiler_schwellwert disable
1)(attr a_boiler_stromwert disable 1)DOELSEIF
([wasserBoilerModus:"festeZeit"])(attr a_boiler_zeit_on disable 0)(attr
a_boiler_zeit_off disable 0)(attr doif_boiler_schwellwert disable
1)(attr a_boiler_stromwert disable 1)DOELSEIF
([wasserBoilerModus:"stromwert"])(attr a_boiler_zeit_on disable 1)(attr
a_boiler_zeit_off disable 1)(attr doif_boiler_schwellwert disable
1)(attr a_boiler_stromwert disable 0)DOELSEIF
([wasserBoilerModus:"schwellwert"])(attr a_boiler_zeit_on disable
1)(attr a_boiler_zeit_off disable 1)(attr doif_boiler_schwellwert
disable 0)(attr a_boiler_stromwert disable 1)
```

```
Define wasserBoilerModus dummy

attr wasserBoilerModus room 243_Wasserboiler

Attr wasserBoilerModus setList switch: manuell festeZeit stromwert
schwell

attr doif_wasserboiler_general room 243_Wasserboiler
```

2.3.6. Kombination zwischen Warmwasserboiler und Infrarotheizung

Logik zur Kombination zwischen Infrarotheizung und Warmwasserboiler.

```
define doif_kombinationHeizungBoiler01 DOIF ([balkonkraftwerk:power_0]
< [infrarotheizung:power_1] and [wasserBoilerModus] eq "stromwert")
(set infrarotheizung off) ( set wasserBoiler off) DOELSEIF
([balkonkraftwerk:power_0] < [infrarotheizung:power_1]) (set
infrarotheizung off)
```

```
attr doif_kombinationHeizungBoiler01 room 243_Balkonkraftwerk
```

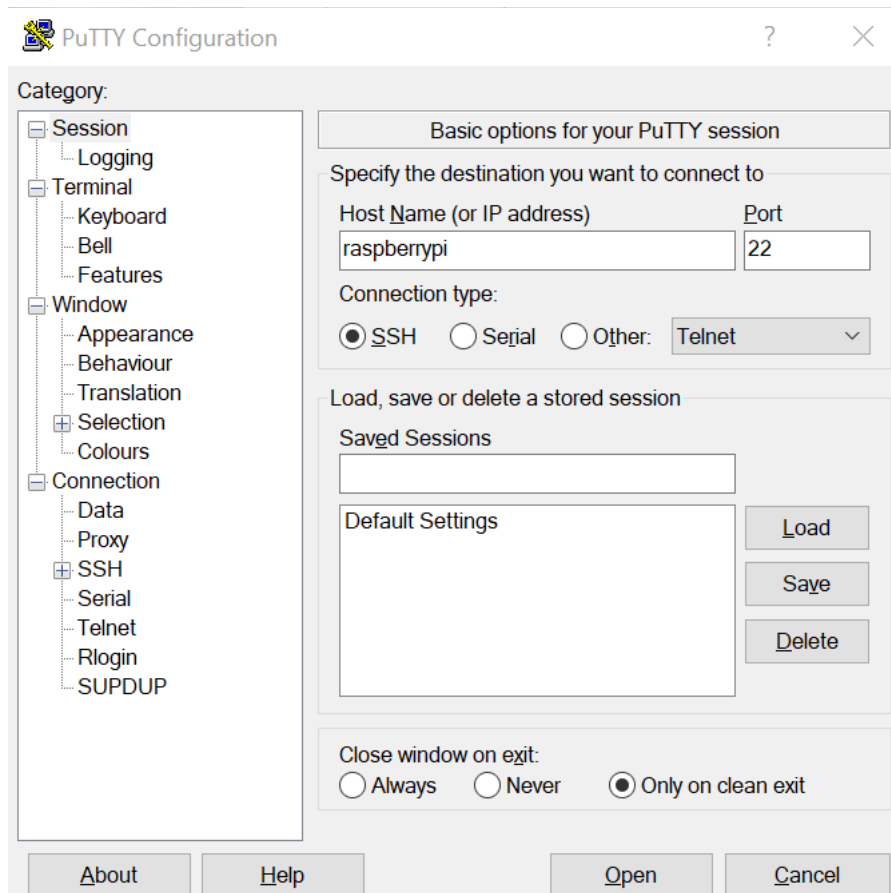
```
define doif_kombinationHeizungBoiler02 DOIF ([balkonkraftwerk:power_0]
> [infrarotheizung:power_1] and [balkonkraftwerk:power_0] <
([infrarotheizung:power_1] + [wasserBoiler:schwellwert]) and
[wasserBoilerModus] eq "stromwert") (set wasserBoiler off)
```

```
attr doif_kombinationHeizungBoiler02 room 243_Balkonkraftwerk
```

3. FTUI erstellen

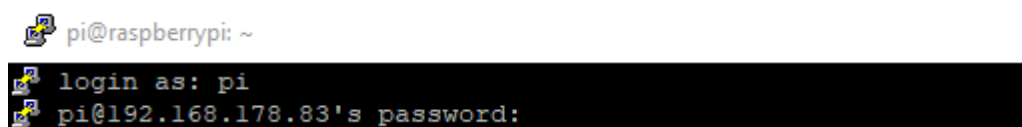
Um die FTUI Oberflächen einrichten zu können ist es maßgeblich, dass die ein Kapitel 1. genannten Schritte vollständig durchgeführt wurden. (Siehe Setup Raspberry Pi).

1. Öffne Putty auf deinem Rechner.
2. Stelle eine Verbindung zu deinem Raspberrypi her, indem du in dem Feld Host Name den Host Namen „raspberrypi“ oder die IP-Adresse des Raspberrypi's eingibst.



Drücke auf „Open“ und die Verbindung wird hergestellt.

3. Gib den Benutzernamen „pi“ und das von dir vergebene Passwort bei der Passwortabfrage ein.



4. Navigiere zu dem Ordner, in dem die Oberflächen gespeichert sind. Gebe dafür den folgenden Befehl ein:

```
cd /opt/fhem/www/
```

5. Lösche den angelegten Ordner namens „tablet“ mit folgendem Befehl:

```
sudo rm -r tablet
```

6. Kclone nun das von uns für dich erstellte Git-Repository indem du folgenden Befehl ausführst:

```
sudo git clone https://github.com/tabascoel/tablet.git
```

Nachdem du alle diese Schritte durchgeführt hast, kannst du Putty beenden und die Oberfläche wie nachfolgend beschrieben testen.

7. Nun öffne deine FHEM Oberfläche und starte FHEM neu, indem du folgenden Befehl in die FHEM Kommandozeile eingibst.

```
Shutdown restart
```

8. Nach Neustart sind alle FTUI-Oberflächen über den Link in der FHEM-Oberfläche namens „Tablet UI“ oder über **Fehler! Linkreferenz ungültig.** erreichbar.

Alle HTML, CSS und JavaScript Dokumente können auch unter dem nachfolgenden Link gefunden werden:

<https://github.com/tabascoel/tablet>

4. HTML-Seiten Konfiguration

Durch das Klonen des Git Repositories sind die HTML Seiten eingerichtet. Achtung, wenn die Devices anders benannt wurden als in der Schritt-für-Schritt Anleitung vorgegeben, müssen die HTML-Dateien angepasst werden. Hierfür muss in der entsprechenden Datei das „data-device“ des entsprechenden Devices angepasst werden.

```
<li data-row="1" data-col="2" data-size="5" data-size="5">
  <!--Widget Überschrift-->
  <Header>Sturmwarnung einschalten</header>
  <!--Switchdevice zur Einstellung der Sturmwarnung-->
  <div class="top-space-2" data-type="switch" data-icon="fa-toggle-on" data-device="SturmwarnungSchalter"
    style="font-size: xx-large;"></div>
</li>
```

Abbildung 1: Frontend Beispiel Sturmweatherwarnung

Am Beispiel des „SturmwarnungSchalter“ zusehen, wenn dies im Backend geändert wird, muss dies auch im Frontend angepasst werden. Es muss der Name des Device im Frontend eingetragen werden.


DeviceOverview			
SturmwarnungSchalter		on	off

Abbildung 2: Backend Beispiel der Sturmweatherwarnung

5. Optionale Aufgabe: Müllplan

Beim Müll Plan ist zu beachten, dass das Frontend auf die Readings vom Backend abgestimmt sind. Das heißt, wenn ein anderer Müll Plan verwendet wird, als der Müll Plan, der in der Schritt-für-Schritt Anleitung enthalten ist, müssen Anpassungen erfolgen.

```
<div class="cell">
  <!-- Reading Text aus dem Backend anzeigen-->
  <div data-type="label" data-device="myABFALLKalender" data-get="Biomuell_text">
  </div>
</div>
<div class="cell left-align">
  <!-- Symbol Widget welches sich dreht, wenn der Abholtag heute ist und blinkt wenn der Abholtag morgen ist-->
  <div data-type="symbol" data-device="myABFALLKalender" data-get="Biomuell_days"
    data-off-color="#5b3a29" data-get-on='["off","0","1","[2-9]{1}|[0-9]{2}"]'
    data-icons='["fa-trash warn", "fa-trash warn fa-spin","fa-trash warn blink", "fa-trash warn"]'
    data-colors='["#5b3a29","#5b3a29","#5b3a29","#5b3a29"]' class="normal"
    data-background-colors='["#808080", "#808080", "#808080","#808080"]'
    data-background-icon="fa-circle">
  </div>
  <div data-type="label" class="left-space">Tag(e)
  </div>
</div>
```

Abbildung 3: Frontend Beispiel Müll Plan

Am Beispiel des oben gezeigten Codes müssen folgende Befehle angepasst werden.

data-get = muss angepasst werden im Beispiel hier heißt das Feld, welches wir auslesen möchten „Biomuell_text“ welches, aus dem Backend aus den Readings entnommen werden kann.

Readings	
Biomuell_date	27.01.2022
Biomuell_days	13
Biomuell_description	Leerung: Biomüll \nBezirk: Bietigheim-Stadt-und-Metterzimmern
Biomuell_location	
Biomuell_text	Biomüll
Biomuell_uid	eae11f08931147e7b688a160dbccb13b
Biomuell_weekday	Donnerstag

Abbildung 4: Backend Beispiel Müll Plan

Beispiel: https://www.avl-ludwigsburg.de/fileadmin/Files/Abfallkalender/ICS/Privat/Privat_2022_Bietigheim-Stadt-und-Metterzimmern.ics

Ggf. muss auf dem PI das Perl Modul zum Nutzen von HTTPS installiert werden. Hierfür auf der Shell eingeben:

```
sudo cpan install IO::Socket::SSL
```

Das Modul ABFALL in FHEM installieren via:

```
update all https://raw.githubusercontent.com/uniqueck/fhem-abfall/master/controls\_fhemabfall.txt
```

Danach FHEM einmal neu starten via:

```
shutdown restart
```

Danach einen Kalender anlegen mit dem Link zum Abfallkalender via:

```
define abfallKalender Calendar ical url https://www.avl-ludwigsburg.de/fileadmin/Files/Abfallkalender/ICS/Privat/Privat\_2022\_Bietigheim-Stadt-und-Metterzimmern.ics
```

```
attr abfallKalender room 97_Abfall
```

Hier für den Beispielkalender. Bei Bedarf den Link individuell abändern.

Anschließend das Modul ABFALL installieren mit der Referenz zum abfallKalender.

```
define myABFALLKalender ABFALL abfallKalender
```

```
attr myABFALLKalender room 97_Abfall
```

Speichern mit Save config.

Mit dem schon angelegten Frontend-Part sind die Leerungstermine nun einsehbar. Die Korrektheit der Daten hängt davon ab, wie diese vom Bereitsteller zur Verfügung gestellt werden. Da es keinen Standard gibt kann sich dies unterscheiden und bedarf manuellen Aufwand zur Anpassung.