

Handbuch



Raspberry Pi 3 Z-Wave Basis Kit

pi^{3g}
www.pi3g.com

Inhaltsverzeichnis

Wichtige Informationen	6
Erste Schritte mit dem RPG Smart-Home-Kit	7
Hinweise zu FHEM	15
Heimautomatisierung mit fhem	19
Vorwort – aller Anfang ist schwer	20
Einleitung und Grundlagen	21
Hardware-Systeme: FS20, HomeMatic und Andere	23
Weitere Hardware: Computer und Computer-Funk-Schnittstelle (CUL)	25
Adressierung, Hauscode und Tastencode	26
Planung Ihres Heimautomatisierungs-Systems	29
fhem-Grundlagen	30
fhem installieren	30
fhem Systemüberblick	30
Die fhem Benutzer-Oberfläche	31
Anlernen von Sensoren in fhem – autcreate	32
Schalten von Aktoren – set	34
Aufzeichnung in Logdateien	34
fhem-Konfiguration – Attribute	35
Attribute: Modell und Raum	37
fhem-Konfiguration: Das besondere Gerät „Global“	38
Geräte-Infos: Readings	39
Mehrere Geräte mit einem Klick schalten – structure	40
Timer	41
Dimmer	42
Schalten zu bestimmten Zeitpunkten – at	43
Schalten von Ereignissen abhängig machen – notify	44
Verwendung von notify als Makro	45
Starten eines Makros – trigger	45
Bearbeiten über das Webfrontend	46
Pairing: direkt oder indirekt	47
Anpassen der Darstellung im Webfrontend - FHEMWEB-Attribute	49
Bildschirmgröße und "Skin": stylesheetPrefix	49
Weniger Menüpunkte: hiddenroom	50
Zusätzliche Menüpunkte – menuEntries	51
Nur bestimmte Befehle für ein Gerät – webCmd	51
Befehle umbenennen – eventMap	51
Räume sortieren: sortRooms	52
Gruppen bilden und umbenennen: group	52
Mehrspaltige Darstellung: column	52
Iconaktualisierung ohne browser-refresh – longpoll	53

Welche Icons? StylesheetPrefix und IconPath	53
Eigene Icons	54
Icon vor dem Gerätenamen – icon	54
Icon vor dem Raumnamen – roomIcons	55
Eigene Icons für Geräte-Schaltzustände	55
Weblink	56
Beispiel zur Konfiguration	56
Gruppen frei auf dem Bildschirm platzieren: Dashboard	57
Einrichten eines Grundrisses mit eigenen fhem-Geräten	57
Einfache Programmierung: if-Bedingung	58
Gerätewerte in Bedingungen – Value, ReadingsVal, ReadingsTimestamp	60
Spezielle Variablen und Operatoren	62
Hinterlegen eigener Programme – 99_myUtils.pm	63
Nur am Wochenende? \$we	63
In den Ferien? holiday, holiday2we	64
Verschachteltes at	65
Heizungssteuerung	66
Messen der Ist-Temperatur	66
Berechnen der erforderlichen Heizleistung	67
Thermostate	67
Regeln der Heizleistung	67
Heizungssteuerung mit fhem	68
Temperaturverläufe als Diagramm – Logs & Plots	69
Jemand zuhause? Anwesenheitserkennung und HomeStatus	70
Anwesenheitssimulation	72
Schaltungen abhängig von Telefonanrufen – Callmonitor	72
Multimedia-Geräte	73
Infrarot-Fernbedienung	74
Prüfung und Aufwecken von Servern im Heimnetzwerk: WOL	74
Webservices, http-Verbindungen	74
Wetterbericht einbinden	74
Google-Kalender, Mails und Messaging	76
Kommunikation über HTTP	76
fhem neu starten – shutdown restart	77
fhem Programmaktualisierungen – update	77
Infoquellen	77
Kurz betrachtet: Weitere Geräte und Funktionen	78
Erweitert: Logs & Plots	78
Dummy	78
Watchdog	78
Grenzwertgesteuertes Schalten – threshold und dewpoint	79
shell-commands	79
Zusammenfassungen – readingsGroup	79

Schalten an bestimmten Wochentagen	79
Kleine Programmierbeispiele	80
Wakeup-Light	80
Sunset und Sunrise	80
Tag und Nacht – isday	80
Erstellen eigener fhem-Programme	80
Entwicklung eigener fhem-Module	80
Anhang: Ein Einblick in die Konfigurationsdatei fhem.cfg	81
fhem.cfg und Includes	82

Wichtige Informationen

pi3g Downloadportal

Im Downloadportal finden Sie das jeweils neueste zur Verfügung stehende Image der vorinstallierten Software

<https://download.pi3g.com/>

Support

In unserem Support-Portal stehen unter der Rubrik "Home Automation" Antworten zu häufigen Fragen, sowie weitere Anleitungen. Alternativ kontaktieren Sie uns bitte unter unserer Support e-Mailadresse:

<http://support.pi3g.com>

support@pi3g.com

FHEM-Community

Das vorinstallierte System FHEM wird von einer deutschsprachigen Community tatkräftig unterstützt. Bitte besuchen Sie die folgenden Links um am Erfahrungsaustausch mit teilzunehmen:

<http://www.fhemwiki.de>

<https://forum.fhem.de>

Zugriff auf die Weboberfläche

<http://smarthome.local>

Benutzer: pi

Passwort: raspberry

Falls <http://smarthome.local> nicht verfügbar ist, installieren Sie bitte die Software Bonjour Druckdienste der Firma Apple - damit funktionieren .local Adressen auch unter Windows. Diese kann unter folgender URL heruntergeladen werden:

<https://support.apple.com/kb/DL999>

Zugriff per SSH

Der Standardzugriff per SSH auf den Raspberry Pi ist möglich, mit den Zugangsdaten
pi / raspberry

wir empfehlen diese Daten aus Sicherheitsgründen zu ändern, beispielsweise mit Hilfe vom SSH Client PuTTY. Bitte beachten Sie unsere Onlinedokumentation für eine detailliertere Anleitung.

Erste Schritte mit dem RPG Smart-Home-Kit

AUSGEPACKT

*Christoph Langner
(Raspberry Pi Geek Magazin)*

Das betriebsfertige RPG Smart-Home-Kit enthält einen RasPi 3 mit der Heimautomatisierungssoftware FHEM sowie das Funkmodul RaZberry, mit dem Sie Z-Wave-Geräte vom RasPi aus fernsteuern und auslesen. So rüsten Sie Ihr Zuhause im Handumdrehen zum "Smart Home" auf.



README

Raspberry Pi, RaZberry, FHEM - diese Begriffe sagen Ihnen bisher wenig? Möchten Sie sofort durchstarten und Ihr Haus mit ein wenig Intelligenz ausstatten, dann leistet Ihnen unser RPG Smart-Home Kit dabei Starthilfe: Das Hardware-Set kommt mitsamt rundum vorkonfigurierter Heimautomatisierungssoftware.

Lieferumfang

- Raspberry Pi 3
- RaZberry-Modul für den GPIO des Raspberry Pi
- Micro-SD-Karte (SanDisk, 16 Byte, UHS-1/Class 10)
- Raspbian Jessie mit FHEM 5.7 vorinstalliert
- Gehäuse (schwarz)
- WLAN und Bluetooth in RasPi3 integriert
- USB-Netzteil (5 V, 2500 mAh)
- Netzkabel (2 m, Cat5e)

Auf der Speicherkarte befindet sich bereits ein Image der Raspberry-Pi-Distribution Raspbian "Jessie" inklusive vorinstalliertem FHEM 5.7, das wir mit einer Reihe von sinnvollen Paketen und Einstellungen erweitert haben. Bei der Bestellung des Kits bietet pi3g zur Ergänzung des Basiskits weitere Pakete für unterschiedliche Szenarien an.

Aktuell gibt es die folgenden Zubehörkits:

- *Outdoor-Paket für Garage und Garten*
- *Security-Paket für Heimsicherheit*
- *Power-Paket zum Schalten von Steckdosen und Lampen*

Auf diese Weise müssen Sie das Set nur noch auspacken, mit Strom versorgen und an Ihr Netzwerk anschließen – über den in den RasPi 3 integrierten WLAN-Adapter gelingt das auch ohne Kabel. Zum Steuern der Smart-Home-Zentrale genügt ein beliebiger PC mit einem Webbrowser. Da FHEM in der Regel seinen Dienst im Hintergrund verrichtet und sich das Programm komplett über eine Weboberfläche bedienen lässt, müssen Sie das Smart-Home-Kit daher auch nicht zwingend mit Maus und Tastatur ausstatten und das System an einen Monitor hängen. Diese Option steht Ihnen jedoch bei Bedarf ebenfalls offen, das System beinhaltet den für Raspbian typischen LXDE-Desktop.

Ausgesuchtes Zubehör mit hoher Qualität

Das Kit kommt komplett vormontiert zu Ihnen, Sie müssen es lediglich auspacken und an Strom sowie das Netzwerk anschließen.

Bei der Wahl von Hardware und Zubehör standen Qualität und Kompatibilität vor einem möglichst günstigen Preis. So liegt dem Kit bewusst eine MicroSD-Karte samt Adapter des Herstellers SanDisk bei, da günstigere Karten von Samsung oder Transcend aufgrund eines Firmware-Bugs des Raspberry Pi 2 auffällig oft Fehler bei Speicherzugriffen produzieren und nicht mehr booten. Bei den von uns ausgewählten Karten geschieht das nicht, obwohl sie zur schnellsten Klasse (UHS-1 / Class 10) zählen und sich mit bis zu 48 MByte/s auslesen lassen.

Das beigelegte Netzteil liefert mit 2500 mA auch bei voll ausgelastetem RasPi genügend Strom, um zusätzlich noch externe 2,5-Zoll-Festplatten über den USB-Anschluss des Mini-Rechners zu betreiben. So können Sie den Smart-Home-RasPi beispielsweise parallel auch als kleinen Fileserver nutzen. Das auf der Speicherkarte vorinstallierte Raspbian-System lässt Ihnen hierzu alle Freiheiten.



Der Chipsatz des in den Raspberry Pi integrierten WLAN-Adapters funktioniert unter Raspbian ohne die Installation weiterer Treiber. Die für das bereits vormontierte RaZberry-Modul notwendigen Einstellungen im Betriebssystem des Raspberry Pi 3 sind bereits erfolgt.

Das zur Kommunikation mit Z-Wave-Geräte benötigte RaZberry-Modul befindet sich beim Kit bereits auf seinem angestammten Platz. Das Gehäuse des Kits ist so gestaltet, dass das Modul das Schließen des Gehäuses nicht behindert.

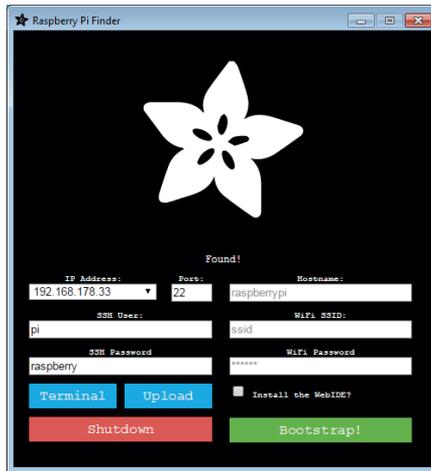
RPG Smart-Home-Kit aufbauen und einbinden

Für den Aufbau des RPG Smart-Home-Kits entnehmen Sie das Gerät mitsamt Zubehör der Verpackung. Der Raspberry Pi sitzt bereits im Gehäuse, ebenso stecken das RaZberry-Modul sowie die Speicherkarte schon im Gerät. Sie müssen daher den RasPi nur über das USB-Steckernetzteil mit Strom versorgen und den Mini-Rechner (zumindest bis zum Einrichten des WLAN-Zugangs) mit dem im Kit enthaltenen Netzkabel an Ihren Router anschließen.

Für den Netzwerkzugang schließen Sie entweder ein Netzkabel an die Ethernet-Schnittstelle des RasPi an oder nutzen den integrierten WLAN-Adapter. Nutzen Sie den FHEM-RasPi mitsamt Maus und Tastatur an einem Monitor oder Fernseher, dann konfigurieren Sie den drahtlosen Netzwerkzugang über die grafische Oberfläche.

Verzichten Sie hingegen auf eine grafische Oberfläche, dann hilft Ihnen der Adafruit Raspberry Pi Finder beim Auffinden des RasPi im Netz. Dieses Tool sowie weitere nützliche Anwendungen wie den SSH-Client PuTTY und oder den Win32 Disk Imager für Windows finden Sie auf der Heft-DVD. Alternativ richten Sie den Netzwerkzugang über das Terminal ein - der Artikel "RasPi im WLAN" (S. 44) nimmt Sie dabei an die Hand. Die Anmeldedaten für den SSH-Login haben wir unverändert bei *pi* als Benutzer und *raspberry* als Passwort belassen.

Mit dem Raspberry Pi Finder von Adafruit finden Sie den Raspberry Pi im Netz, auch wenn Sie keinen Zugriff auf die Weboberfläche ihres Routers besitzen und die IP-Adresse des RasPi-Systems nicht kennen.



TIPP

Für etwas mehr Komfort leitet der RasPi Anfragen von Port 80 auf den Port 8083 der FHEM-Oberfläche weiter. Sie können diese daher auch einfach über <http://smarthome.local> aufrufen. Möchten Sie auf dem Smart-Home-Kit nachträglich einen eigenen Webserver betreiben, kommt Ihnen diese Weiterleitung in die Quere. Sie lässt sich allerdings leicht durch eine kleine Änderung in der Konfigurationsdatei `/etc/default/pound` (`startup=1` in `startup=0`) deaktivieren.

Das auf der Speicherkarte vorinstallierte Raspbian spricht von Haus aus Deutsch. Die FHEM-Software gibt es jedoch nur auf Englisch, obwohl die Software bisher vornehmlich im deutschsprachigen Raum benutzt wird. Zahlreiche Dokumentationen, Foren und Blogs helfen Ihnen jedoch auf Deutsch weiter.



Zum Steuern des Systems greifen Sie zu einem Browser: Arbeiten Sie direkt auf der grafischen Oberfläche, dann finden Sie dort einen Link zur Weboberfläche von FHEM. Möchten Sie den RasPi aus dem Netzwerk von Ihrem PC aus ansprechen, dann erreichen Sie FHEM unter <http://smarthome:8083> oder (falls die Namensauflösung nicht funktioniert) über <http://IP-Adresse:8083>. Die Login-Daten entsprechen in der Grundeinstellungen denen des RasPi-Systems: *pi* mit *raspberry* als Passwort. Das Passwort gleicht sich nicht mit dem des RasPi-System ab, Sie müssten es in der FHEM-Konfigurationsdatei `/opt/fhem/fhem.cfg` von Hand ändern.

Auf dem Desktop des Raspbian-Systems finden Sie die wichtigsten Ressourcen zum Heimautomatisierungssystem FHEM. Benötigen Sie keine grafische Oberfläche, dann lässt sich diese zum Sparen von Ressourcen auch deaktivieren.

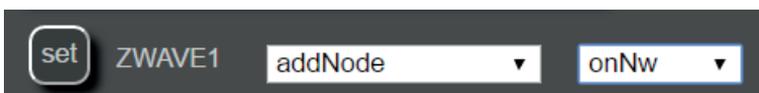
TIPP

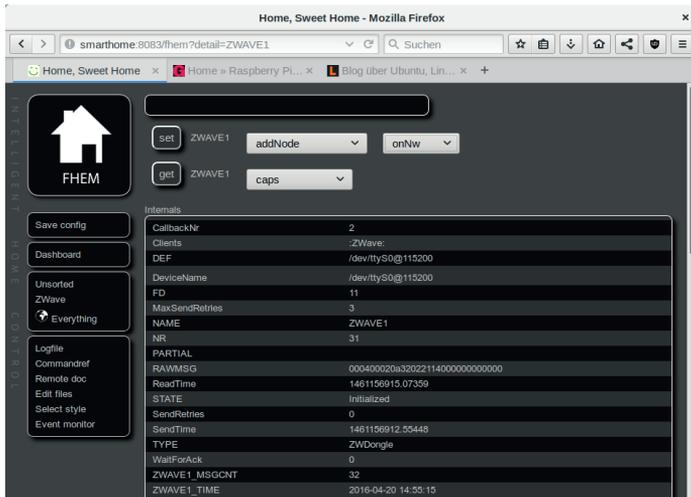
Nach dem Einrichten verrichtet der Smart-Home-RasPi in der Regel ohne Monitor, Maus und Tastatur seinen Betrieb. Eine grafische Desktop-Umgebung belastet dann das RasPi-System nur unnötig und verschwendet CPU-Zeit sowie Arbeitsspeicher. Es empfiehlt sich daher, den Start der GUI bei Bedarf abzustellen. Starten Sie dazu mit `sudo raspi-config` das Konfigurationswerkzeug des Raspbian-Systems. Dort stellen Sie unter **3 Boot Options** den Start des Systems auf **B1 Console** um. Falls Sie die Image-Datei des **RPG Smart-Home-Kits** auf einem eigenen RasPi installieren, sollen Sie außerdem über **Expand Filesystem** das Dateisystem auf die komplette Speicherkarte ausdehnen.

Verbindung zu den Sensoren und Aktuatoren aufnehmen

Zum Anlernen ("Inklusion") von Z-Wave-Geräten versetzen Sie das RaZberry-Modul über die Oberfläche von FHEM in den Verbindungsmodus. Dazu öffnen Sie über **Everything** aus der Seitenleiste die Komplettübersicht und wechseln dann zum Z-Wave-Modul des RasPi unter dem Namen **ZWAVE1**. Dort blendet FHEM im Kopf eine Befehlsleiste ein. Darunter finden Sie Felder zum Setzen von Zuständen, mitsamt der jeweiligen Parameter als Ausklappmenü (**set**).

In der nächsten Reihe blendet FHEM dann die für das Gerät spezifischen Parameter ein, die sich mit **get** auslesen lassen. Stellen Sie in der **set**-Zeile nun **addNode** sowie daneben **onNw** ein und tippen Sie auf die Schaltfläche **set**. Daraufhin schaltet FHEM das RaZberry-Modul in den Inklusionsmodus um. Alternativ geben Sie in der FHEM-Kommandozeile `set ZWAVE1 addNode onNw` direkt als Befehl ein.





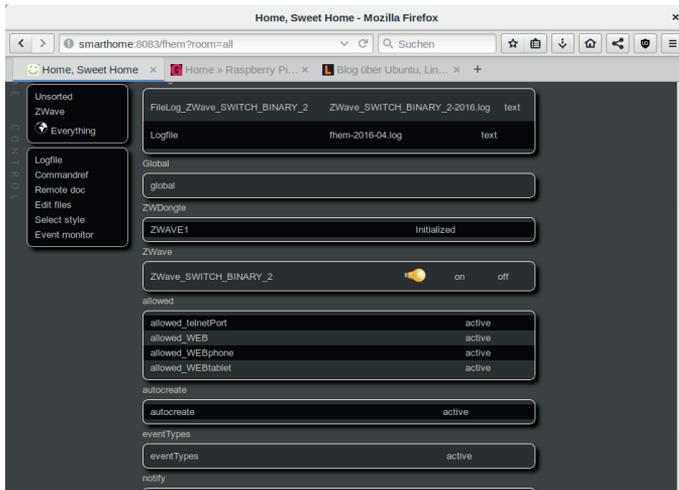
Das vorinstallierte Betriebssystem des RPG Smart-Home-Kits ist so konfiguriert, dass die Kommunikation zwischen FHEM und dem RaZberry-Modul ohne Probleme funktioniert. So lassen sich direkt erste Z-Wave-Geräte in das System einbinden.

Danach setzen Sie die Batterie des anzulernenden Z-Wave-Geräts ein oder stecken dieses in die Steckdose. Viele Geräte aktivieren dabei automatisch den Anlernmodus, bei anderen müssen Sie eventuell noch einen Knopf am Gerät drücken. Aktivieren Sie vor der Inklusion in FHEM den **Event monitor** aus der Seitenleiste und lassen sich zudem das **FHEM log** (Auswahlbox unter der FHEM-Befehlsleiste) anzeigen, dann erhalten Sie Informationen zum Erfolg der Aktion. Das genaue Vorgehen zum Einbinden (auch "Inklusion" genannt) von Z-Wave Geräten entnehmen Sie bitte dem jeweiligen Geräte-Handbuch.

Listing 1 zeigt das erfolgreiche Anlernen eines Z-Wave-kompatiblen Zwischensteckers zum Schalten von Geräten und Messen des Stromverbrauchs. FHEM listet das Gerät dann unter **ZWave** beziehungsweise **Everything** als Komponente des Smart-Home-Systems auf. Dort könnten Sie auch gleich den Zwischenstecker von der Ferne per Hand umstellen.

Listing 1

```
2016.04.20 14:54:06 2: autocreate: define ZWave_SWITCH_BINARY_2 ZWave
c3c3f7e2 2 25863272708575733127
2016.04.20 14:54:06 2: autocreate: define FileLog_ZWave_SWITCH_BINARY_2
FileLog ./log/ZWave_SWITCH_BINARY_2-%Y.log ZWave_SWITCH_BINARY_2
2016.04.20 14:54:06 2: ZWave set ZWave_SWITCH_BINARY_2 associationAdd 1 1
2016.04.20 14:54:06 2: ZWave get ZWave_SWITCH_BINARY_2 model
```



In diesem Beispiel stellt FHEM in der Weboberfläche einen in das System eingebuchten Z-Wave-Zwischenstecker mitsamt Statusanzeige und Buttons zum Betätigen der Schaltfunktion dar.



Für den Einstieg in die FHEM-Programmierung eignen sich am besten Zwischenstecker zum Schalten von Geräten. Diese bieten meist auch eine Funktion den Stromverbrauch der angeschlossenen Geräte zu messen.

Typische Szenarien für die Heimautomation

Damit Sie den Schalter leichter in der FHEM-Oberfläche finden, sollten Sie dessen Namen anpassen: Aus dem im Beispiel gezeigten **ZWave_SWITCH_BINARY_2** werden Sie nur solange schlau, wie Sie nur wenige Geräte in FHEM angelegt haben. Zum Umbenennen geben Sie `rename` als Befehl in der FHEM-Befehlszeile ein, gefolgt vom aktuellen und dem zukünftigen Namen (Listing 2), und drücken am Ende die Eingabetaste. Als Strategie für die Namensgebung hat es sich in der Praxis bewährt, dem sprechenden Namen ein Kürzel für das jeweilige Zimmer ("SZ" fürs Schlafzimmer, "WZ" fürs Wohnzimmer, etc.) anzugeben. Danach lässt sich der Zwischenstecker aus der FHEM-Befehlszeile oder aus

FHEM-Skripten beispielsweise per `set WZ.Zwischenstecker toggle` umschalten.

Listing 2

```
rename ZWave_SWITCH_BINARY_2 WZ.Zwischenstecker
```

Kombinieren Sie das Schalten mit weiteren Modulen wie `at`, dann lässt sich aus dem Zwischenstecker mit einer angeschlossenen Lampe ein Wecklicht bauen. Die Kommandos aus Listing 3 schalten um 8 Uhr das Licht ein und eine halbe Stunde später wieder aus. Mit einer kleinen `if`-Prüfung beschränken Sie den Wecker auf Wochentage, sodass Sie am Wochenende ungestört ausschlafen können (Listing 4).

Listing 3

```
### Wecklicht mit konkreten Zeiten zum An- und Ausschalten
define WecklichtAn at *08:00:00 set WZ.Zwischenstecker on
define WecklichtAus at *08:30:00 set WZ.Zwischenstecker off
### Aktiviere das Wecklicht für 1800 Sekunden (30 Minuten)
define Wecklicht at *08:00:00 set WZ.Zwischenstecker on-for-timer 1800
```

Listing 4

```
### Nur wochentags wecken

define WecklichtWochentag at *08:00:00 { if (!( $\$we$ )) { fhem("set
WZ.Zwischenstecker on-for-timer 1800");; } }
```

Das Schalten des Zwischensteckers lässt sich auch mit weiteren Sensoren und den von diesen ausgelösten Events kombinieren. Das Beispiel aus Listing 5 würde aktiviert den Zwischenstecker beim Öffnen einer Tür und sorgt damit etwa für Licht oder einen akustischen Alarm, der hoffentlich Einbrecher verschreckt.

Listing 5

```
### Licht beim Öffnen einer Tür schalten

define di_Schalten D0IF ( [WZ.Tuersensor:doorWindow] eq "ff" ) ( set
WZ.Zwischenstecker on )
```

Alternativen und Ausbaumöglichkeiten

Das RPG Smart-Home-Kit erleichtert Ihnen den Einstieg in die Welt der Heimautomatisierung mit dem RaspberryPi. Es enthält handverlesene, gut miteinander harmonisierende

Komponenten. Mit optional erhältlichen Zusatzpacks für unterschiedliche Szenarien (Outdoor, Security, Power-Paket) kombinieren Sie das Basiskit zu einem kompletten Smart-Home-Paket. Dank der vorinstallierten, bereits konfigurierten und handoptimierten Software müssen Sie das Kit lediglich auspacken und ans Netzwerk anschließen. Besitzen Sie bereits einen RaspberryPi, nutzen Sie das von pi3g vorbereitete Image auf der Heft-DVD.

Merken Sie nach eigenen Tests, dass Sie mit FHEM als Software nicht warm werden, dann können Sie das Set auch zusammen mit der vom RaZberry-Hersteller angebotenen Z-Way-Software verwenden. Die zusammen mit dem Modul erstandene Lizenz der Software gilt auch für das RPG Smart-Home-Kit.

Hinweis:

In der Sonderausgabe der Zeitschrift Raspberry Pi Geek zum Thema Home Automation finden Sie viele weitere wertvolle Hinweise die Ihnen die Einrichtung erleichtern. Beispielsweise bei der Installation der Alternativsoftware Z-Wave. Wir empfehlen den Kauf der Zeitschrift und den Besuch von raspberrypigeek.de.

Hinweise zu FHEM

Installationsort

FHEM ist im Ordner

`/opt/fhem`

installiert. Insbesondere die Datei

`/opt/fhem/fhem.cfg`

enthält Ihre spezifische Konfiguration und kann nach Ihren Vorstellungen bearbeitet werden. Anschließend sollte FHEM mit

`shutdown restart`

zum wieder-Einlesen neu gestartet werden.

Telnet

Telnet wurde auf Zugriffe von localhost beschränkt, zusätzlich wurde das Passwort "raspberry" für Zugriffe von außerhalb vergeben, falls man die erste Option (allowFrom) mittels # auskommentieren möchte.

`/opt/fhem/fhem.cfg`

```
define telnetPort telnet 7072 global
```

```
attr telnetPort allowfrom 127\.\0\.\0\.
```

```
attr telnetPort globalpassword raspberry
```

Webzugriff-Passwort ändern

Es wurden für die drei Webbackends (:8083, :8084, :8085) jeweils die folgenden Login-

Daten vergeben:

Nutzer: pi
Passwort: raspberry

Diese können in /opt/fhem/fhem.cfg angepasst werden (jeweils pi:raspberry durch eigene Kombination austauschen):

```
define WEB FHEMWEB 8083 global  
attr WEB basicAuth {"$user:$password" eq "pi:raspberry" }
```

```
define WEBphone FHEMWEB 8084 global  
attr WEBphone stylesheetPrefix smallscreen  
attr WEBphone basicAuth {"$user:$password" eq "pi:raspberry" }
```

```
define WEBtablet FHEMWEB 8085 global  
attr WEBtablet stylesheetPrefix touchpad  
attr WEBtablet basicAuth {"$user:$password" eq "pi:raspberry" }
```

Falls kein Passwortschutz gewünscht ist, können diese Zeilen auskommentiert werden.

Wichtige Hinweise zu den Z-Wave Geräten

Z-Wave Geräte können nur an einem Primärcontroller (beispielsweise der FHEM - Raspberry Pi Basisstation) gleichzeitig betrieben werden.

Der Z-Wave Controller (in unserem Fall der RaZberry) speichert die inkludierten NodeIDs intern im eigenen Speicher, und die Geräte selbst speichern ebenfalls die homeID des Controllers. Dieser Speicher ist vollkommen unabhängig von der microSD Karte des Raspberry Pi und dessen Inhalt.

Selbst wenn Sie FHEM neuinstallieren bleiben die Geräte daher in Ihrem Z-Wave Netzwerk inkludiert - bitte exkludieren Sie die Geräte vor einer Neuinstallation von FHEM daher gegebenenfalls, oder legen Sie die Geräte manuell nach der Neuinstallation erneut an.

Jedes RaZberry Modul hat darüber hinaus eine eigene homeID - sie sind also nicht einfach austauschbar. Bei einer Beschädigung der Basiseinheit müssen Sie Ihre Z-Wave Geräte daher manuell auf Werkseinstellungen zurück setzen, siehe entsprechende Dokumentation des jeweiligen Gerätes.

Ein Update auf eine neue Imageversion von uns überschreibt das gesamte Dateisystem und damit die Einstellungen - jedoch können Sie die Datei
/opt/fhem/fhem.cfg

aufheben, und auf das neue System übertragen. Dadurch stellen Sie Ihre speziellen Einstellungen und Gerätekonfigurationen wieder her. Wir empfehlen ein regelmäßiges Backup dieser Konfigurationsdatei.

Exklusion

Bitte lesen Sie das jeweilige Geräte-Handbuch für Hinweise zur Inklusion und zur Exklusion der Geräte. Falls das Gerät nicht auf Anrieb wie gewünscht erkannt wurde, versuchen Sie es zu exkludieren und erneut zu inkludieren. Sie können das Gerät exkludieren mittels

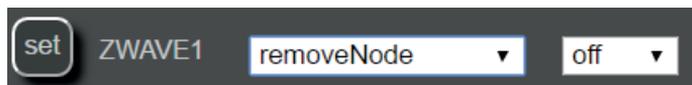
set ZWAVE1 removeNode onNw

und dann entsprechender Bestätigung am Gerät (siehe jeweilige Anleitung).



Das Gerät wird jetzt entfernt und lässt sich nicht mehr fernsteuern. Bitte setzen Sie jetzt:

set ZWAVE1 removeNode off



und entfernen Sie das Gerät auch aus der FHEM Oberfläche mittels

delete <device>

z.B. delete ZWave_SENSOR_NOTIFICATION_26



Falls Sie möchten können Sie auch die dazugehörige Log Datei löschen, bspw:

delete FileLog_ZWave_SENSOR_NOTIFICATION_26

Geräteklassen ergänzen

Falls nicht alle Funktionen wie erwartet zur Verfügung stehen, können Sie in FHEM fehlende Geräteklassen des jeweiligen Z-Wave Gerätes mittels

attr <Gerätename> classes <Klassen>

ergänzen. Beispielsweise:

attr ZWave_SWITCH_BINARY_20 classes ZWAVEPLUS_INFO SWITCH_BINARY CONFIGURATION ASSOCIATION ASSOCIATION_GRP_INFO MANUFACTURER_SPECIFIC SCENE_ACTIVATION SCENE_ACTUATOR_CONF VERSION FIRMWARE_UPDATE_MD POWERLEVEL SECURITY MARK DEVICE_RESET_LOCALLY HAIL BASIC

Geben Sie dabei jeweils alle Klassen die das Gerät hat mit an.

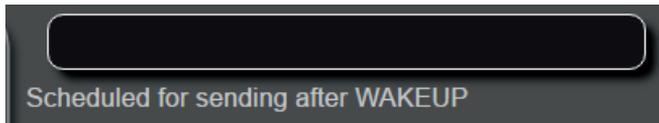


Hinweis:

POWERLEVEL ist zum Auslesen der Funkleistung des Gerätes, und nicht zum messen des Stromverbrauchs der angeschlossenen Geräte. Dazu dient die Klasse METER.

Aufwecken von batteriebetriebenen Geräten

Bei der Einstellung von Geräten mittels set, oder dem Auslesen von Gerätewerten mittels get wird die Weitergabe an das Gerät beim Aufwachen durchgeführt. Die Geräte wachen periodisch - oft in 24 Stunden Intervallen - auf. Um während der Einrichtung sofortige Resultate zu erzielen sollten Sie das Gerät wecken. Das erfolgt häufig durch drücken der Programmier Taste, bitte beachten Sie das jeweilige Handbuch.



Assoziation der Geräte untereinander

ZWave Geräte können untereinander assoziiert werden, beispielsweise kann ein Schalter oder ein Bewegungsmelder direkt eine Funksteckdose schalten. Es können gleichzeitig mehrere Geräte ausgelöst werden.

Die Assoziation kann in FHEM gesetzt werden:

```
set <device> associationAdd <group> <targetdevice>
```

Beispielsweise:

```
set ZWave_a1_23 associationAdd 2 8
```

Die benötigte Gruppe entnehmen Sie je nach Aktion dem Handbuch - oft gibt es nur die Gruppe 1. Mit

```
get <device> associationAll
```

können Sie die verfügbaren Gruppen, die momentan verbundenen Zielgeräte, und die maximale Zahl der möglichen Zielgeräte auslesen.

Die Target-Device ID entnehmen Sie dem von FHEM vergebenen Geräte-Namen (bspw für ZWave_SWITCH_BINARY_8 ist sie 8).

Um eine Assoziation zu löschen benutzen Sie das verwandte Kommando:

```
set <device> associationDel <group> <targetdevice>
```

Testspray für Rauchmelder

Bitte benutzen Sie zum Test von Rauchmeldern ein speziell zu erwerbendes Rauchmelder Test-Spray. Die Sensoren können durch Ablagerungen aus echtem Rauch beschädigt/ in der Funktion beeinträchtigt werden.

Heimautomatisierung mit them

– Für Einsteiger –



© 2012-2014 Ulrich Maaß, uli.maass@gmail.com, Letzte Aktualisierung: 02.02.2014
Abdruck in diesem Handbuch mit freundlicher Genehmigung des Autors Ulrich Maaß. Wir nahmen teilweise Kürzungen vor, und ergänzten in Kommentaren Z-Wave spezifische Teile.

Vorwort – aller Anfang ist schwer

Auch ich habe mal mit Heimautomatisierung und fhem angefangen. Ein super Produkt, zu dem man sich jedoch alle Infos schnipselweise zusammensuchen musste: Von wo herunterladen, wo die Hardware beziehen, was braucht man überhaupt, wie initialisiere ich die Hardware... Fragen über Fragen. Um anderen genau diese Anfangsschwierigkeiten zu ersparen, habe ich dieses Dokument geschrieben: Als Einstieg und Überblick zu den ersten Schritten. Als die erste Version entstand, war ich mit meinen ‚Grundlagenforschungen‘ gerade fertig – zwei Monate nach meinem Start mit fhem. Das war im Mai 2011. Seitdem wurde das Dokument bereits ca. 300.000 mal heruntergeladen – Ansporn für die nun vorliegende vierte Version.

In der zweiten Version wurde der Grundlagenteil überarbeitet und Kapitel zur fhem-Konfiguration durch das Webfrontend, zur Anpassung des Webfrontends sowie zu ersten kleinen fhem-Perl-Programmen sind hinzugekommen.

Die nun vorliegende vierte Version ist um Kapitel wie Heizungssteuerung, Anwesenheitserkennung, HomeStatus und Multimedia sowie viele kleine weitere Neuerungen erweitert worden.

Über ein kurzes Feedback von Lesern freue ich mich übrigens sehr! Hat's geholfen, ist eine bestimmte Passage nicht so gut verständlich, fehlt etwas? Schickt mir doch einfach mal ne kurze Mail an: uli.maass@gmail.com

Mein Dank gilt den Korrekturlesern Annette, Klaus, Andreas und Peter sowie allen Entwicklern von fhem, vor allem Rudolf König.

Viel Spaß beim Lesen und vor allem mit fhem,
Uli

Einleitung und Grundlagen

Wenn Sie das erste Mal einen Blick auf fhem („Freundliche Hausautomation und Energie-Messung“, sprich „feem“) werfen, möchten Sie sicher wissen, was Sie damit erreichen können.

Heimautomatisierung bedeutet: Wenn ich auf einen Schalter drücke, soll das Licht angehen; wenn es kalt ist, soll die Heizung eingeschaltet werden; bei Sonnenaufgang soll die Gartenwegbeleuchtung ausgehen; um 7 Uhr morgens soll der Rollladen hochfahren, sofern ich nicht in Urlaub bin. Geräte sollen auch vom PC, Tablet-PC oder Smartphone bedienbar sein. Alles das lässt sich mit fhem bewerkstelligen.

Damit lassen sich auch die Hauptziele der Heimautomatisierung erreichen:

- Energie sparen durch das zielgerichtete Regeln bzw. Ein- und Ausschalten von Geräten
- Mehr Komfort durch Automatisierungen – zeitgesteuert oder abhängig von Messdaten wie z.B. der Temperatur, dem Sonnenstand etc.
- Zugriff auch über (W)LAN und das Internet
- Spaß beim Basteln :-)

In der Welt der Heimautomatisierung gibt es dafür vier Gruppen von Geräten:

1. **Sensoren** (Thermometer, Lichtschalter, Fernbedienungen, Bewegungsmelder, Infrarot-Empfänger etc. - also Geräte, die Werte aus der Umwelt aufnehmen und an das System senden).
2. **Aktoren** (Steckdosenschalter, Unterputzschalter, Rollladen-Schalter etc. – also jedes Gerät, das Befehle vom System entgegennimmt und als Schaltvorgang umsetzt).
3. **Sender&Empfänger**, die als Funk-zu-PC-Schnittstelle fungieren und die (Funk-)kommunikation übernehmen – das kann ein Hardwaresystem-spezifisches PC-Interface sein oder auch ein „CUL“ genannter generischer Sender & Empfänger.
4. Außerdem benötigen Sie einen **Computer**, also einen PC oder einfachen Rechner wie z.B. ein NAS, einen RaspberryPi oder eine FritzBox, der ständig eingeschaltet ist, an der der CUL angeschlossen ist und auf dem fhem läuft.

Sensoren und Aktoren gibt es von mehreren Herstellern. Jeder Hersteller hat dabei ein jeweils in sich geschlossenes **Hardwaresystem**, das zunächst auch ohne fhem funktioniert. Beispiele dafür sind die Systeme FS20, FHT, HomeMatic, OneWire oder EnOcean (wenn Sie keine Ahnung haben, was das alles ist – später mehr). Jedes dieser Systeme umfasst eine Vielzahl von Komponenten, bestehend aus Sensoren wie z.B. Bewegungsmelder, Thermostate, Wandschalter (Unterputz und Aufputz) und Fernbedienungen, sowie Aktoren wie z.B. Steckdosenschalter (Funk-Zwischenstecker), funkgesteuerte Steckdosenleisten, Dimmer, Stellantriebe für Heizkörperventile oder

Rollladenschalter.

Die meisten dieser Hardwaresysteme funktionieren per Funk in den speziellen Frequenzbändern 433 MHz oder 868 MHz. Bei manchen Systemen erfolgt die Kommunikation aber auch über Kabel wie z.B. in den Hardwaresystemen One-Wire oder KNX/EIB.

Der überwiegende Teil dieser Hardwaresysteme bietet auch ein PC-Interface, um Schaltvorgänge vom PC, ggf. auch über Webfrontend und mit Automatisierungen, steuern zu können. Im FS20-System heißt dieser PC-Adapter z.B. FHZ1300PC, bei HomeMatic heißt er CCU oder LAN-Adapter.

fhem kann sich in diese Hardwaresysteme „einklinken“. Dazu wird entweder der systemeigene Adapter genutzt (es wird also die FHZ1300PC oder der LAN-Adapter von fhem aus gesteuert), oder man verwendet statt dieser relativ teuren Hardwaresystem-eigenen PC-Interfaces (=Sender & Empfänger) ein generisches Gerät, den sogenannten ‚CUL‘. Dieser ist deutlich günstiger und kann z.T. mehrere Hardwaresysteme gleichzeitig bedienen. Der CUL kann also z.B. eine FHZ1300PC ersetzen.

Z-Wave / Hinweis von pi3g

In unserem System ist der Z-Wave Adapter „RaZberry“ für die Ansteuerung von Z-Wave Home Automation Komponenten bereits innerhalb des Gehäuses verbaut. Sie müssen daher keine weiteren Funkmodule anschaffen.

Selbstverständlich können Sie das System aber um weitere USB Adapter, um auch andere Heimautomatisierungssysteme ansteuern zu können, erweitern. Das ist einer der großen Vorteile von FHEM.

Bitte beziehen Sie im übrigen die im Folgenden über „CUL“ genannten Informationen, soweit anwendbar, auf das Gerät „ZWAVE1“ (= der RaZberry Adapter).

Vermutlich besitzen Sie schon jetzt Geräte, die mit Hilfe von fhem gesteuert werden können: einen modernen SmartTV, eine FritzBox, vielleicht auch ihr AV-Receiver? Diese können Sie ohne Mehrkosten steuern und Überwachen.

Wenn Sie tiefer in die Heimautomatisierung mit fhem einsteigen möchten, sollten Sie sich zunächst für (mindestens) eines der oben genannten Hardwaresysteme entscheiden und die entsprechenden zueinander passenden Komponenten beschaffen.

Einfache Anwendungen, die sich dann gestalten lassen, sind z.B.:

- Ein Sensor soll einen Aktor steuern, z.B. soll mit einem Lichtschalter an der Wand eine Lampe ein- und ausgeschaltet werden.

- Derselbe Schalter soll evtl. auch mehrere Geräte (also Aktoren) bedienen, wenn z.B. mit nur einem Tastendruck mehrere Lampen oder beliebige Verbraucher ein- und ausgeschaltet werden sollen.
- Es sollen evtl. auch mehrere Schalter die gleiche Funktion auslösen, z.B. soll im Schlafzimmer das Licht sowohl vom Lichtschalter an der Wand als auch von der Fernbedienung am Bett bedienbar sein.
- Wenn Sie fhem nutzen, können Sie alle Schaltvorgänge sowohl von einem Webfrontend auf einem Computer als auch von einer App auf einem mobilen Gerät steuern. Außerdem können Sie zeit- und ereignisbasierte Schaltvorgänge auslösen.

Eine solche Konfiguration können Sie ganz ohne Programmierkenntnisse erreichen, es braucht nur einiges technisches Grundverständnis.

Wenn Sie Aktoren ausschließlich vom Webfrontend (also ohne physischen Schalter) bedienen wollen, müssen Sie diese durch wenige einfache Programmzeilen ‚hinzuprogrammieren‘.

Das Zusammenfassen von mehreren Aktoren in Räumen mit gemeinsamer Schaltung vom Webfrontend aus ist ebenfalls mit wenigen Programmzeilen recht einfach oder kann innerhalb des Hardwaresystems eingerichtet werden (also z.B. alle Lampen im Schlafzimmer sollen mit nur einem Tastendruck ausgehen).

Wenn Sie weitergehende Möglichkeiten nutzen möchten, z.B. Zeitsteuerungen wie Laufzeiten, Sonnen-auf-/untergang, Ferienkalender, thermostatgeführte Heizungssteuerung etc., müssen Sie zunehmend tief in die fhem- und ggf. Perl-Programmierung einsteigen.

fhem ist sehr umfangreich. Es bietet die Möglichkeit, einfache Konfigurationen ohne Programmierkenntnisse zu erstellen, wie auch eine komplexe Heimautomatisierung, die eben Programmierung benötigt.

Vor dem Einstieg in fhem ist es erforderlich, zunächst die Funktionsweise der Steuerung zu verstehen. Also geht es zunächst um

Hardware-Systeme: FS20, HomeMatic und Andere

Sicher haben sie einfache Hardware-Systeme schon einmal im Baumarkt gesehen: eine Funk-Fernbedienung, an die einige Zwischensteckdosen gekoppelt werden können. Mit der Fernbedienung kann man dann bequem Lampen oder sonstige Geräte ein- und ausschalten.

Grundsätzlich sind die Hardwaresysteme, die im Kontext zu fhem relevant sind, nichts anderes – sie bieten aber eine größere Anzahl unterschiedlicher Geräte, die miteinander

gekoppelt werden können, auch sind diese Hardwaresysteme erweiterbar. Und ja: auch manche (!) der vorgenannten Baumarkt-Steckdosen lassen sich aus fhem heraus schalten.

Das derzeit am weitesten verbreitete Hardwaresystem heißt FS20. Die Geräte dafür werden im Wesentlichen von den Firmen ELV und Conrad vertrieben. Wenn Sie auf deren Webseiten den Suchbegriff ‚FS20‘ eingeben, bekommen Sie eine bebilderte Übersicht der Komponenten: Sensoren wie Wand-Lichtschalter, Fernbedienungen, Bewegungsmelder, Thermostate oder Helligkeitssensoren, Aktoren wie Steckdosenschalter, Infrarotsender, Steckdosenleisten, Rollladenschalter etc. – schauen Sie doch gleich mal nach.

In FS20 können Sensoren ausschließlich senden und Aktoren ausschließlich empfangen (der Lichtschalter an der Wand kann Funktelegramme nur senden, die Steckdose kann sie nur empfangen). Das ist eigentlich ausreichend – bis ein Aktor z.B. durch eine Funkstörung oder wegen schlechter Laune mal einen Befehl verpasst. Das FS20-System geht einfach davon aus, dass der gesendete Befehl auch ausgeführt wurde. Im vergleichbaren Hardware-System HomeMatic hingegen können Aktoren auch senden, sie senden also eine Bestätigung, dass sie den Befehl auch wirklich ‚gehört‘ haben. Das ist zwar noch keine absolute Garantie, dass der Befehl auch wirklich umgesetzt wurde, aber immerhin gibt es ein Feedback. Bleibt dieses aus, kann z.B. der Befehl erneut gesendet oder eine Warnmeldung ausgegeben werden.

HomeMatic-Komponenten sind somit deutlich zuverlässiger, aber auch teurer (sie kosten häufig das Doppelte der vergleichbaren FS20-Komponente). Wenn Sie kritische Geräte steuern möchten, z.B. Ihre Rollläden auch als Einbruchsicherung dienen und daher definitiv geschlossen sein sollen, können Sie HomeMatic in Erwägung ziehen. Wenn es lediglich um das Ein- oder Ausschalten von Lampen u. ä. geht, oder Sie für gewöhnlich Zuhause sind und die Schaltergebnisse selbst sehen, ist FS20 wahrscheinlich ausreichend. Die Erfolgsquote beim Schalten von FS20-Geräten liegt bei mittleren Entfernungen, mittleren Eigenschaften der Wände und gutem Batterie-Ladestand der Sensoren (=Sender) bei gefühlten 95-98%.

Z-Wave / Hinweis von pi3g

Der Vorteil von Z-Wave liegt in der großen Auswahl von miteinander kompatiblen Modulen vieler unabhängiger Hersteller. Darüber hinaus bauen die Z-Wave-Geräte ein Mesh-Netzwerk auf. Zwischenstecker und andere Geräte mit fester Stromzufuhr erweitern so die Funkreichweite Ihres Z-Wave Netzwerks.

Z-Wave Geräte senden Bestätigungen über die empfangenen Kommandos zurück, ähnlich wie Homematic.

Die Firma Pollin Electronic führt neben unseren Z-Wave Zubehörkits übrigens auch Homematic-Komponenten.

Sowohl HomeMatic (HM) als auch FS20-Systeme lassen sich mit fhem steuern. Eine Mischung beider Systeme ist möglich – allerdings benötigen Sie dann einen CUL je System – also einen für FS20, einen für HM.

Bei der Definition eines Geräts in fhem gibt man den Typ des Hardwaresystems an. Wenn Sie in fhem ein Gerät definieren, müssen Sie dabei angeben, ob es sich um FS20, Homematic usw. handelt. Ist diese Gerätedefinition einmal vorgenommen, können dann alle fhem-Funktionen gleichartig genutzt werden, da das Gerät dann nur noch über seinen Namen angesprochen wird. Das Einschalten eines Geräts erfolgt z.B. durch den Befehl `set Lampe on`. Dabei ist es dann gleichgültig, ob das Gerät Lampe ein FS20, HomeMatic oder anderes Gerät ist.

Im Weiteren wird FS20 für alle Beispiele verwendet.

Die größte Besonderheit von fhem ist, dass mehrere Hardwaresysteme eingebunden und ggf. auch kombiniert werden können. Vermutlich werden Sie mit einem Hardwaresystem starten, haben aber die Möglichkeit, Ihr System später zu erweitern und sind bei dieser Erweiterung somit nicht auf nur ein Hardwaresystem festgelegt. Eine vollständige (ständig wachsende) Liste finden Sie in der <http://fhem.de/commandref.html> am Anfang unter „Devices“ sowie <http://fhem.de/fhem.html#Hardware>.

Weitere Hardware: Computer und Computer-Funk Schnittstelle (CUL)

Sie benötigen den **Sender&Empfänger**. Ein CUL sieht aus wie ein USB-Stick und kann an einen PC, ein NAS oder z.B. eine FritzBox angeschlossen werden. Zu beziehen ist der CUL von <http://busware.de/tiki-index.php?page=CUL&highlight=cul>. Für den RaspberryPi gibt es dort das Aufsteckmodul COC, das die gleiche Funktion wie der CUL übernimmt, dabei jedoch keinen USB-Steckplatz belegt.

Der CUL ist das am weitesten verbreitete und damit am besten getestete Gerät und wird daher empfohlen. Im weiteren Text wird nur CUL erwähnt, der dann synonym für „FHZ1300PC, HM LAN-Adapter, CCU, CUL, CUNO, Tuxradio oder Tuxrail sowie alle anderen Schnittstellen zwischen PC und dem Hardwaresystem“ steht.

Der CUL von busware kann leider nur ohne Firmware bestellt werden. Diese Firmware muss also zunächst auf den CUL „ge-flasht“ werden. In der neuesten fhem-Version wird ein CUL automatisch ge-flasht, der an einem Gerät angeschlossen wird, auf dem fhem läuft. Probieren Sie es also zunächst so. Sollte dies nicht funktionieren, können Sie Details in der fhem-Befehlsreferenz, nachlesen.

Sie müssen den **Sender & Empfänger betreiben**. Es gibt unterschiedliche Geräte, an denen der Sender betrieben werden kann: Ein PC mit Windows oder Linux, Linux-fähige

NAS-Geräte oder auch eine FritzBox. Da es in vielen Haushalten auch um das Sparen von Strom geht, nutzen Sie am besten ein Gerät, das ohnehin ständig eingeschaltet ist. fhem ist in der Programmiersprache Perl implementiert, d.h. grundsätzlich können Sie jedes Gerät verwenden, das über einen Netzwerkanschluss, einen USB-Anschluss und eine Perl-Installation verfügt.

Ebenfalls sehr gut zum Betrieb von fhem geeignet ist der Raspberry Pi. Es handelt sich um einen zigaretenschachtel-großen Kleincomputer mit nur 4W Leistungsaufnahme, der für den Betrieb von fhem aber völlig ausreichend ist. Gegenüber der FritzBox bietet er den Vorteil, dass weitere Perl- und Linux-Komponenten nachinstalliert werden können und dass er performanter ist. Eine Einrichtungsanleitung für den Raspberry Pi findet sich z.B. <http://www.meintechblog.de/2013/05/fhem-server-auf-dem-raspberry-pi-in-einer-stunde-einrichten/>, auch gibt es einen [Wiki-Artikel \(http://www.fhemwiki.de/wiki/Raspberry_Pi\)](http://www.fhemwiki.de/wiki/Raspberry_Pi).

Sie benötigen außerdem **Sensoren und Aktoren**. Im Weiteren werden als Beispiel eine funkgesteuerte Steckdose (FS20-st) und ein Lichtschalter (FS20-S4A) betrachtet. Andere Geräte funktionieren ähnlich: die Zwischen-Steckdose (FS20-st) arbeitet ebenso wie ein Unterputzschalter (FS20-SU). Ein Wand-Lichtschalter mit vier Knöpfen wirkt genauso wie eine Fernbedienung mit vier Knöpfen (FS20 S4).

Hinweis:

Von einigen Hardware-Komponenten finden Sie in den Webshops (leider) unterschiedliche Revisionsstände, z.B. FS20-st-2 oder FS20-st-3. Diese sind in ihrer Funktion identisch und können auch gemischt betrieben werden.

Wenn Sie starten möchten, bestellen Sie den Sender&Empfänger (CUL) zuerst, da er ein paar Tage Lieferzeit hat. Zum Testen können Sie sich außerdem z.B. zunächst 2 Funk-Schalt-Steckdosen (FS20-st) und einen Funk-Lichtschalter (FS20-S4A) besorgen.

Adressierung, Hauscode und Tastencode

Jedes Hardwaresystem verwendet eine ‚Adressierung‘, durch die alle Komponenten es Systems eindeutig identifiziert sind. So verwendet z.B. HomeMatic eine ab Werk im jeweiligen Gerät fest codierte ID als eindeutige Identifikation innerhalb dieses Hardwaresystems. Intertechno arbeitete mit einer Adressierung, die über DIP-Schalter an den Komponenten eingestellt wird. In FS20 wird am Sender der Hauscode eingestellt, der Empfänger wird auf eine Adresse „angelernt“. So hat jedes Hardwaresystem seinen eigenen Mechanismus, um Komponenten eindeutig ansprechen zu können.

Z-Wave / Hinweis von pi3g

Im Z-Wave System hat jeder Controller eine eigene ID, die er bei der Inklusion dem jeweiligen Z-Wave Gerät automatisch mitteilt.

homeld

c88e291f

nodeldHex

01

Die Home ID Ihres Systems - durch den RaZberry vergeben - können Sie bei dem Gerät ZWAVE1 in der Tabelle Internals ermitteln. Diese ID ordnet die Systeme eindeutig Ihrem Haushalt zu. Sie ist von Werk aus fest eingestellt und kann von Ihnen nicht geändert werden.

Jedes Gerät hat darüber hinaus eine nodeld, die hexadezimal (nodeldHex) von FHEM angegeben wird. Der Controller hat immer die nodeld 01. Sie benötigen die nodeld beispielsweise zum assoziieren der Geräte untereinander.

Hier eine kleine Tabelle zum Umrechnen:

Hex	1	2	3	4	5	6	7	8	9	0A	0B	0C	0D	0E	0F	10	11	12	13	14	15
Dez	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21

Jedes Z-Wave Gerät kann nur einmal inkludiert werden. Die Geräte verfügen über Assoziationsgruppen mit Hilfe derer sie zu Gruppen zusammengestellt werden können.

Die Anleitung wird im folgenden für das FS-20 System spezifisch, die Abschnitte "Einrichten von Sensoren", "Anlernen von Aktoren (Pairing)", "Funktionsgruppen, Lokale Master, Globale Master" haben wir daher aus diesem Nachdruck entfernt.

Das FS20-System arbeitet im 868 Megahertz Funkfrequenzband. Um die Geräte gegebenenfalls innerhalb Ihres Haushalts und vor allem von denen Ihrer Nachbarn zu unterscheiden, wird der sogenannte *Hauscode* verwendet. Der **Hauscode** ist die System-ID Ihres gesamten Hardwaresystems, der auf allen Komponenten, die gemeinsam funktionieren sollen, gleich ist und sie als ‚zusammengehörig‘ kennzeichnet. Normalerweise werden Sie in ihrem gesamten Hardwaresystem denselben Hauscode verwenden – so sind die Hardwaresysteme vom jeweiligen Hersteller ursprünglich konzipiert. Ein Hauscode in FS20 ist achtstellig, wobei jede Stelle Werte zwischen 1 und 4 annehmen kann. Ein Beispiel ist ‚12341234‘. Die Wahl Ihres eigenen Hauscodes ist beliebig, aber wichtig, falls z.B. auch Ihr Nachbar ein FS20-System betreibt – er soll ja nicht plötzlich Ihre Wohnzimmer-Beleuchtung einschalten können.

Am Sensor – also dem Wandschalter oder der Fernbedienung- wird jeder Taste bzw. jedem Tastenpaar außerdem ein **Tastencode** zugewiesen. Wird eine Taste gedrückt, sendet der Sensor ein Funktelegramm mit der Information Hauscode und Tastencode. Anderen Sensoren, wie z.B. einem Bewegungsmelder, wird ebenfalls ein Tastencode zugeordnet, damit im Funktelegramm eindeutig ist, von welcher Komponente das Funktelegramm stammt. An einigen wenigen Komponenten sind Hauscode und

Tastencode nicht frei wählbar, da diese keine Tasten zur Eingabe haben, z.B. ein Thermostat. Diese Geräte werden sich somit nicht in Ihren Hauscode einreihen lassen, fallen dadurch etwas aus der Reihe – funktionieren aber trotzdem.

Alternativ können Sie bei Verwendung eines CUL auch z.B. einen Hauscode je Etage Ihres Hauses verwenden, oder aber einen Hauscode je Raum oder sogar je Sensor. Der Vorteil liegt im Einrichten des Sensors, denn ab Werk sind zwar Hauscodes zufällig vergeben, Tastencodes jedoch immer gleich. Wenn Sie jedem Sensor seinen Werks-Hauscode lassen oder je Sensor einen anderen Hauscode vergeben, haben Sie mit geringem Aufwand wiederum die Eindeutigkeit der Funktelegramme erreicht – es gibt zwar mehrere Tasten mit demselben Tastencode, diese sind jedoch durch die unterschiedlichen Hauscodes unterscheidbar.

FS20 bietet die Möglichkeit, **Schaltgruppen** zu bilden, die sogenannten Funktionsgruppen, Lokale Master und Globale Master. Sie sind dazu gedacht, mehrere Aktoren, die sonst auch einzeln schaltbar sein sollen, mit nur einem Tastendruck gleichzeitig zu schalten (z.B. alle Lampen in einem Raum mit nur einem Tastendruck ausschalten).

Wichtig:

Diese im Hardwaresystem definierten Gruppen funktionieren nur innerhalb desselben Hauscodes! Die Gruppen werden durch vom Hardwaresystem vorgegebene, spezielle Tastencodes gebildet. Eine detaillierte Beschreibung finden Sie im Handbuch jedes FS20-Sensors. Auch ist zu beachten, dass ein Aktor gegebenenfalls mehrfach angelernt werden muss – je einmal für Funktionsgruppe, Local Master und Global Master.

Alternativ lassen sich solche Schaltgruppen ebenso in der fhem-Konfiguration bilden – ebenfalls über unterschiedliche Hauscodes hinweg. Das ist bequemer einzurichten als das Einstellen der Tastencodes an den Schaltern, führt jedoch beim Schalten von Gerätegruppen zu mehr Funkverkehr: es wird dann nicht ein Funktelegramm mit dem Gruppen-Tastencode gesendet, sondern ein Funktelegramm je Kombination Hauscode+Tastencode (=Adresse) – also ein Funktelegramm je Gerät. Da auf Grund funkrechtlicher Bestimmungen (1%-Regel, http://fhemwiki.de/wiki/1%25_Regel) nicht mehr als 160 Funktelegramme pro Stunde gesendet werden dürfen, kann dies in großen Systemen zu Problemen führen. Bitte berücksichtigen Sie dies, wenn Sie Ihre Systematik von Haus- und Tastencodes entwerfen.

Als Empfehlung für den Start: Je Hauscode können bis zu 225 Tastencodes definiert werden. Wenn dies für Sie ausreichend ist, nutzen Sie nur einen Hauscode, um ggf. später Schaltgruppen einführen zu können. Reicht Ihnen diese Anzahl nicht, nutzen Sie einen Hauscode je Etage. Damit können Sie Gruppen innerhalb der Etage durch das Hardwaresystem bilden, Gruppen über Etagen hinweg in fhem. Wenn Sie also 3 Etagen haben, und alle Lampen im Haus ausschalten wollen, müssen Sie nur 3 Funktelegramme senden (ein Funktelegramm für die entsprechende Gruppe je Etage).
Zu Gruppen und fhem später mehr.

Planung Ihres Heimautomatisierungs-Systems

Bevor Sie alle Ihre Sensoren und Aktoren bestellen, machen Sie sich eine Skizze, wie Ihre Heimsteuerung aussehen soll. Aspekte, die Sie dabei berücksichtigen sollten, sind:

- Welche Geräte möchten Sie schalten und steuern (z.B. Lampen, Rollläden, Heizkörperventile, Kaffeemaschinen, Radios, WLAN an/aus etc.)?
- Wo befinden sich diese Geräte, in welcher Struktur lassen sie sich zusammenfassen (z.B. Etage und/oder Zimmer)?
- Wo sollen welche Sensoren platziert werden?
- Welche Gruppen von Aktoren sollen gemeinsam geschaltet werden können?

Bauen Sie sich eine Liste auf, in der alle Paare von Sensoren und Aktoren gelistet sind. Geben Sie jedem dieser Paare einen Namen sowie einen Haus- und Tastencode, z.B.

Sensor	Hauscode	Tastencode	fhem-Name	Schaltet Gerät (Aktor)
FS20-S4A Wohnzimmer, Erdgeschoss, Tastenpaar 01	12341234	01	eg_wz_Stehlampe	Stehlampe, Wohnzimmer, Erdgeschoss
FS20-S4A Schlafzimmer, Erdgeschoss, Tastenpaar 00	12341234	00	og_sz_RolladenLinks	Rolladen links, Schlafzimmer, Obergeschoss

Sie können beliebige Namen wählen, die jedoch eindeutig sein müssen. Gerätenamen dürfen neben Punkt und Unterstrich keine Sonderzeichen enthalten. Ein weiteres Beispiel zur Benennung der Geräte finden Sie [hier \(https://www.fischer-net.de/hausautomation/fhem/22-fhem-devicenamen.html\)](https://www.fischer-net.de/hausautomation/fhem/22-fhem-devicenamen.html).

Definieren Sie auch, welche Geräte gemeinsam in **Gruppen** geschaltet werden sollen. Auslöser ist auch bei Gruppen ein bestimmter Tastencode. Führen Sie bei den Aktoren auf, welche Ihrer Aktoren bei Auslösen dieser Gruppe geschaltet werden sollen. Bedenken Sie, dass die Bildung von Gruppen nur innerhalb eines Hauscodes möglich ist.

Sensor	Hauscode	Gruppen- und Tastencode	fhem-Name (Gruppe)	Schaltet Gerät (Aktor)
FS20-S4A 2 Wohnzimmer, Erdgeschoss, Tastenpaar 00: Alle Lichter WZ	12341234	44 11*	eg_wz_LichtAlle	Funktionsgruppe fg 11: Stehlampe, Wandleuchte, Deckenlicht, Tischleuchte, Wohnzimmer, Erdgeschoss

FS20-S4A 2 Wohnzimmer, Erdgeschoss, Tastenpaar 01: Media	12341234	44 12*	haus_MediaAlle	Funktionsgruppe fg 12: Verstärker, TV, Videorecorder, Airplay Küche, Airplay Bad
--	----------	--------	----------------	---

*) Eine detaillierte Erklärung der Adress-Struktur finden Sie im Handbuch für den FS20 S4A ab Seite 12.

Mit den bisherigen Ausführungen sollte die Funktionsweise des FS20-Systems deutlich geworden sein. Weitere Details entnehmen Sie den FS20 Handbüchern, die jedem Sensor beiliegen und auch online einsehbar sind – schauen Sie z.B. auf der ELV-Homepage unter FS20-S4A nach, dort finden Sie das Handbuch als pdf.

fhem-Grundlagen

Wir können uns nun dem eigentlichen Spaß zuwenden: der Einbindung der gekoppelten Geräte. In diesem Abschnitt werden grundlegende FS20- und fhem-Befehle dargestellt. Es liegt im Auge des Betrachters, ob man dies als Scripting oder bereits als Programmierung bezeichnet.

Auch werden wir die unterschiedlichen Möglichkeiten kennenlernen, um diese Konfigurationen vorzunehmen.

fhem installieren

Die Installationsschritte sind abhängig vom verwendeten Computer. Grundsätzlich sind folgende Schritte erforderlich:

1. Installieren des Betriebssystems, z.B. Linux oder Windows
2. Installieren von Perl (dies ist Voraussetzung, da fhem in Perl programmiert ist)
3. Installieren von fhem

Im [fhem-Wiki \(http://www.fhemwiki.de/wiki/Hauptseite\)](http://www.fhemwiki.de/wiki/Hauptseite) finden Sie [Anleitungen \(http://www.fhemwiki.de/wiki/Kategorie:Server_Hardware\)](http://www.fhemwiki.de/wiki/Kategorie:Server_Hardware) zur Installation für unterschiedliche Betriebssysteme und teilweise auch für bestimmte Hardware, z.B. FritzBox oder QNAP NAS. Bitte schlagen Sie dort nach.

fhem Systemüberblick

fhem besteht aus dem Kernel (fhem.pl) sowie einer Vielzahl von Modulen. Module können die unterschiedlichsten Aufgaben übernehmen vom Anbinden eines

Hardwaresystems über die Bereitstellung eines Frontends bis zur Automatisierung von Aufgaben. Eine Übersicht der grundlegenden Systemarchitektur finden Sie [hier \(http://www.fhemwiki.de/wiki/Systemübersicht\)](http://www.fhemwiki.de/wiki/Systemübersicht).

Die fhem Benutzer-Oberfläche

Bevor die ersten Befehle über die fhem-Oberfläche getätigt werden, hier ein erster kurzer Blick auf das Standard-Frontend von fhem, genannt ‚pgm2‘.

Sie erreichen fhem über Ihren Browser unter der Adresse <http://<ip>:8083/fhem>

The screenshot displays the fhem web interface. On the left is a vertical navigation menu with a house icon and the text 'Fhem'. The menu items are: 'Floorplans', 'CUL WS', 'Esszimmer', 'Korridor', 'Kueche', 'Media', 'Plots', 'Schlafzimmer', 'Unsorted', 'Wohnung', 'Wohnzimmer', 'Everything', 'Howto', 'Wiki', 'Details', 'Edit files', 'Select style', and 'Event monitor'. On the right, there is a search bar with a 'save' button. Below the search bar are two control panels: 'FS20' and 'structure'. The 'FS20' panel contains three items: 'wz LampeKlein', 'wz Licht', and 'wz Wandbeleuchtung', each with a lightbulb icon and 'on off' buttons. The 'structure' panel contains one item: 'wz LichtAlle' with a lightbulb icon and 'on off' buttons. Numbered callouts (1-7) point to various elements: 1 to 'Kueche', 2 to 'Wiki', 3 to the search bar, 4 to the 'save' button, 5 to the lightbulb icons in the FS20 panel, 6 to the 'structure' panel, and 7 to 'Event monitor'.

Hinweis von pi3g: wir haben das dunkle Theme für das FHEM-Frontend für Sie voreingestellt. Die Funktion ist identisch.

- ❶ Liste der Räume – zu Beginn der Einrichtung erscheinen nur ‚Unsorted‘ und ‚Everything‘. Räume werden den Geräten in der Konfiguration mit `tels attr <Gerät> room <Raumname> zugeordnet`.
- ❷ Links auf Hilfetexte, die fhem-Doku und zur Konfigurationsseite ‚Edit files‘
- ❸ Eingabefeld zum direkten Ausführen von fhem-Befehlen
- ❹ Mit Klick auf den `save`-button speichern Sie Ihre Konfigurations-Änderungen in der Datei `fhem.cfg`. **Achtung!** Nach Eingabe eines fhem-Befehls NICHT auf den `SAVE`-button klicken, sondern lediglich `<ENTER>` betätigen!
In neueren fhem-Versionen gibt es statt des im screenshot gezeigten `save`-buttons im oberen Bereich des Menüs den Eintrag „save config“, dieser hat dieselbe Funktion.
- ❺ Nach der Einrichtung erscheinen hier Ihre Geräte mit der Darstellung des Schaltzustands und einem on/off-Schalter
- ❻ Um Schaltgruppen statt im Hardwaresystem alternativ in fhem aufzubauen, können Geräte auch Strukturen zugewiesen werden. Sie erlauben das gesammelte Schalten aller zugeordneten Geräte - auch über unterschiedliche Hauscodes hinweg.
- ❼ Mit dem Event-Monitor können Sie alle Funkbefehle verfolgen. Sehr hilfreich, um Ablauf und Inhalt der Funkkommunikation zu verstehen.
Hinweis: Der Funkverkehr ist auch über ein telnet-Terminalprogramm an Port 7072 zu sehen, wenn Sie in der telnet-Terminalsitzung `inform timer` eingeben.

Anlernen von Sensoren in fhem - autocreate

Hinweis von pi3g:

Bitte legen Sie Z-Wave Geräte gemäß unserer Anleitung mit `set ZWAVE1 addNode onNw an`.

Nachdem am Sensor der gewünschte Hauscode und die gewünschten Tastencodes eingestellt sind, werden die Tasten(paare) nacheinander in fhem angelernt.

Hinweis:

Sensoren können automatisch an fhem gekoppelt werden. Bevor Sie fortfahren, stellen Sie sicher, dass die dafür zuständige Funktion „autocreate“ aktiviert ist. Tippen Sie dafür im fhem webfrontend den Befehl `list autocreate` in das Kommandofeld ein und bestätigen mit `<ENTER>`. In der Ausgabe muss u.a. `STATE active` erscheinen.

```
Internals:
  NAME      autocreate
  NR        6
  STATE     active
  TYPE      autocreate
Attributes:
  autosave  1
  filelog   /var/Internals
```

Erscheint dies nicht, geben Sie in das Kommandofeld ein
define autocreate autocreate
um das automatische Anlegen von Geräten zu aktivieren.

Um den Sensor mit fhem zu koppeln, drücken Sie die erste zu ‚lernende‘ Taste auf dem Schalter (Sensor). Dadurch sendet dieser ein Funktelegramm, das vom CUL empfangen und an fhem weitergegeben wird. Da die Kombination von Hauscode und Tastencode bisher nicht bekannt ist, wird von fhem automatisch ein zugehöriges fhem-device angelegt. Auf dem fhem webfrontend (mit F5 den Web-Browser refreshen) erscheint dieses nun mit einem kryptischen Namen, sie heißt dort z.B. FS20_602b01. Es wurde von fhem also ein Funktelegramm mit dem vorläufigen Namen FS20_602b01, nämlich dem (hexadezimal dargestellten) Hauscode 602b und dem Tastencode 01 erkannt. Notieren Sie sich auch diese vierstellig hexadezimale Version des Hauscodes – ich habe mir dafür eine Excel-Tabelle angelegt. Auf Ihrem fhem-frontend finden Sie im oberen Bereich das weiße Eingabefeld , über das Sie direkt getippte fhem-Befehle absetzen können. Der erste Befehl soll dem Umbenennen des Schalters dienen.

Überlegen Sie sich die Bezeichnung Ihres Schalters gut, denn unter diesem Namen wird er fortan überall angezeigt. Ich habe mich dafür entschieden, die ersten zwei Stellen für das Zimmer zu verwenden (also wz=Wohnzimmer, ez=Esszimmer, sz=Schlafzimmer usw.) und dann eine Beschreibung anzuhängen. Meine Sensor-Tasten heißen so, wie die Geräte, die geschaltet werden sollen, also z.B. wz_LampeSofa, sz_LeseLampe, ez_LichtRegal. Man kann sich ein beliebiges Namenssystem überlegen, jedoch müssen die Namen eindeutig sein und dürfen keine Leerstellen enthalten.

Hinweis:

In der iPhone/iPad-App fhemobile wird der Teil vor dem Unterstrich nicht angezeigt, aus wz_LampeSofa wird in der Anzeige für das Wohnzimmer also LampeSofa.

Hinweis:

Die Darstellung der Werte nach ELV-Quad-Format und hexadezimaler Darstellung ist alternativ – beide Varianten funktionieren. Die Umrechnung erfolgt [so \(http://www.fhemwiki.de/wiki/FS20_Allgemein#FS20_Adressumrechnung\)](http://www.fhemwiki.de/wiki/FS20_Allgemein#FS20_Adressumrechnung).

Angenommen Sie möchten wz_LampeKlein schalten: Tippen Sie im weißen Eingabefeld rename <NameAlt> <NameNeu>, in unserem Beispiel also
rename FS20_602b01 wz_LampeKlein

Das ist nun also der neue fhem-Name des Funktelegramms, also dem Paar aus gedrücktem Taster bzw. Taster-Paars und dem entsprechend angelernten Aktor. Auf dem fhem-frontend erscheint nun wz_LampeKlein mit den klickbaren Links für ON und OFF.

FS20	State	Set to
wz_LampeKlein	 on	off
wz_Licht	 on	off
wz_Wandbeleuchtung	 on	off

Damit ist Ihr erstes fhem-device erstellt. Sie können diese Steckdose nun vom Schalter und vom Webfrontend steuern.

Um Ihre neue Konfiguration mit dem umbenannten FS20-device dauerhaft in der Konfigurationsdatei fhem.cfg zu speichern,

- Klicken Sie links im Menü auf „save config“, oder
- geben Sie im Eingabefeld den Befehl `save` ein und bestätigen mit `<ENTER>`.

Schalten von Aktoren - set

Versuchen Sie im Webfrontend das Klicken auf ON und OFF. Schalten Sie die Steckdose mit dem Schalter ein und aus – wenn sich dabei die Änderung im Web-Frontend nicht automatisch ändert, aktualisieren Sie die Darstellung in Ihrem Browser mit F5 und kümmern Sie sich später um **Iconaktualisierung ohne browser-refresh – longpoll (seite 34)**. Sie können das Schalten von Geräten auch durch getippte Befehle erreichen: in dem weißen Feld im oberen Bereich der fhem-Seite, dem Kommandofeld, können direkte Befehle eingegeben werden. Probieren Sie `set <Name> on` und `set <Name> off`, also z.B. `set wz_LampeKlein on`

Wenn derselbe Schalter mehrere Aktoren schalten soll, lernen Sie die weiteren Aktoren einfach genauso wie den ersten am selben Sensor (Schalter) an.

Um zu verstehen, welche Funktelegramme mit welchen Inhalten gesendet werden, verwenden Sie den Event Monitor oder Terminal 7072 `inform timer` (Anleitung siehe *Zugang über Terminalprogramm, Telnet*)

Mit dem `set`-Befehl können auch mehrere Geräte gemeinsam geschaltet werden. So können Sie z.B. mehrere Geräte aufzählen:

```
set lamp1,lamp2,lamp3 on
```

oder –falls Ihre Geräte passend benannt sind- einen Bereich angeben:

```
set lamp[1-3] on
```

oder wildcards verwenden (eine Perl-Besonderheit: `*` reicht nicht, es muss `*` angegeben werden):

```
set lamp.* on
```

Weitere Beispiele und Erklärungen finden Sie in der [commandref](http://fhem.de/commandref.html#devspec) unter [Device Specifications \(http://fhem.de/commandref.html#devspec\)](http://fhem.de/commandref.html#devspec). Mit den bisher beschriebenen Schritten lässt sich bereits eine recht umfangreiche Konfiguration erreichen.

Aufzeichnung in Logdateien

Alle Funktelegramme, egal ob von einem Sensor oder von fhem/CUL gesendet, werden in

Logdateien mitgeschrieben. Besonders wichtig ist die Haupt-Log-Datei von fhem. Diese ist normalerweise in Monatsscheiben aufgeteilt. In meinem System sieht das so aus:



Der Inhalt der Logdatei sind alle Funktelegramme, aber auch Fehlermeldungen von fhem. Insbesondere deshalb ist es wichtig, immer mal wieder, spätestens aber beim Auftreten von Fehlern in die Logdatei zu schauen.

Hier ein Auszug:

```
2012.03.12 08:40:00 2: FS20 set sz_Leselampe dim100% 1280
2012.03.12 09:01:01 2: FS20 set sz_Rollo off
2012.03.12 09:20:01 2: FS20 set sz_Leselampe off
2012.03.12 09:20:01 2: FS20 set sz_Stehlampe off
```

Die Funktion autocreate legt außerdem eine Logdatei je Gerät an. Insbesondere für Messgeräte wie Temperatursensoren ist diese wichtig, da der Inhalt die Grundlage für die Darstellung in Diagrammen ist. Dazu später mehr.

Bevor weitere fhem-Befehle erklärt werden, zunächst ein Blick auf die Konfiguration von fhem.

fhem-Konfiguration - Attribute

Die in fhem angelernten bzw. manuell angelegten Geräte können weiter konfiguriert werden, indem ihnen ‚Attribute‘ mit einem Wert zugewiesen werden.

Die Einstellungen z.B. zu ez_LichtRegal können über das webfrontend eingesehen werden, indem man nach dem Klick im linken Bereich auf ‚Everything‘ nicht auf ON oder OFF, sondern auf den Namen des Gerätes klickt, z.B. auf ez_LichtRegal. Dadurch erreicht man diesen sogenannten **device-Detail-Bildschirm** (*nächste Seite*):

The screenshot shows the Fhem web interface. On the left is a navigation menu with links like 'Esszimmer', 'Kueche', 'Media', etc. The main content area is titled 'Delete ez_LichtRegal Modify ez_LichtRegal'. It features a control panel with 'State' (off), 'Value' (toggle), and 'Measured' (2011-09-21 19:11:58). Below this is a table of 'Internal' variables:

Internal	Value
BTN	00
CUL_MSGCNT	1
CUL_RAWMSG	F6969001238
CUL_RSSI	-46
CUL_TIME	2011-09-21 19:11:58
DEF	6969 00
LASTIODev	CUL
MSGCNT	1
NAME	ez_LichtRegal
NR	11
STATE	on
TYPE	FS20
XMIT	6969

At the bottom is an 'Attribute' table:

Attribute	Value	Action
room		attr
Wohnung	Wohnung_Alle	deleteattr
fm_order	2	deleteattr
model	fs20st	deleteattr
room	Esszimmer	deleteattr

Dieser Block zeigt die „Internals“ eines Geräts. Hier ist der them-Name (NAME ❶) des Geräts zu sehen.

(DEF ❷) zeigt die Definition, wie sie zum Zeitpunkt des „define“-Befehls dieses Geräts angegeben wurde. Im screenshot sind dies Hauscode und Tastencode.

Der aktuelle Schaltzustand wird ebenfalls angezeigt (STATE ❸).

Im unteren Bereich werden die Attribute des Geräts angezeigt, die hier auch verändert werden können. Ein Beispiel ist die Angabe des FS20-Modells (model ❹).

Die Veränderung von Geräte-Attributen (Details später) können Sie auf drei unterschiedlichen Wegen erreichen:

1. Klicken Sie im oben gezeigten Detail-screen auf einen Attribut-Namen, um dessen Wert zu bearbeiten – oder wählen Sie aus der Dropdown-Liste das Attribut aus, das Sie hinzufügen möchten. Als Beispiel wählen Sie das Attribut `model` aus, dann in der rechten Liste `fs20-st` und klicken dann die Schaltfläche ‚attr‘. Schließen Sie Ihre Konfiguration ab durch Eingabe des Befehls `save` in das Eingabefeld oder klicken auf „save config“ im Menü, durch den Ihre Konfiguration in der Datei `fhem.cfg` gespeichert wird.
2. Geben Sie den Befehl `attr wz_MediaServer model fs20-st` in das Eingabefeld ein und bestätigen Sie mit <ENTER>. Schließen Sie Ihre Konfiguration ab durch Eingabe des Befehls `save` in das Eingabefeld, durch den Ihre Konfiguration in der Datei `fhem.cfg` gespeichert wird.
3. Klicken Sie im linken Bildschirmbereich auf `Edit Files`, dann auf `fhem.cfg` (oder verwenden Sie einen anderen Editor). Scrollen Sie zu der Definition Ihres devices. Fügen Sie darunter die Zeile `attr wz_MediaServer model fs20-st` ein. Klicken Sie im oberen Bildschirmbereich den button ‚save `fhem.cfg`‘.

Alle diese Bearbeitungsvarianten erzielen denselben Effekt: Sie ändern Ihre `fhem-`Konfiguration und speichern Ihre Änderungen in der Datei `fhem.cfg`. Am bequemsten ist der Weg 1, der auch Tippfehler bei der Eingabe weitgehend vermeidet. Zur besseren Lesbarkeit sind im weiteren Text alle Befehle so dargestellt, wie sie in das Kommandofeld

eingetippt werden können – alternativ können Sie alle Befehle aber auch durch Klicks im Detailbildschirm der Geräte erreichen.

Probieren Sie die drei Varianten der fhem-Konfiguration aus: Über über das webfrontend, über Kommandozeile oder direkt in der fhem.cfg. Es wird empfohlen, den Großteil der Konfigurationsschritte mit Variante 1 oder 2 vorzunehmen. Lediglich das Definieren neuer devices (also define...) muss über die Varianten 2 oder 3 erfolgen, da es das device im webfrontend ja noch nicht gibt und man daher auch nicht dorthin navigieren kann.

Besondere Beachtung verdient die Information STATE (siehe Screenshot ③). STATE zeigt den zuletzt gesendeten Zustand des Geräts. Meist ist dieser Zustand on oder off, kann aber auch toggle oder dim50% oder (bei einem Temperatursensor) T: 8.0 C sein – je nachdem, was zuletzt gesendet wurde und welches Gerät verwendet wird.

Attribute: Modell und Raum

Kennzeichnen Sie auch die bereits in Ihrer Konfiguration vorhandenen Schalter mit dem FS20 Steckdosenmodell (denn diese steuern Sie ja damit):

```
attr wz_Lampe model fs20-st
```

Die Angabe des Modells erlaubt eine passendere Darstellung in den fhem-frontends, z.B. werden für Schalter keine Dimm-Befehle angeboten etc.

Auch können Sie nun jedes device einem Raum in Ihrer Wohnung zuordnen:

```
attr wz_MediaServer room Wohnzimmer
```

```
attr wz_Lampe room Wohnzimmer
```

Sie sehen nun im Menü links einen neuen Raum ‚Wohnzimmer‘. Wenn Sie dort klicken, werden alle Geräte angezeigt, die diesem Raum zugeordnet wurden.

Hinweis:

Einen existierenden Raum können Sie im Detailscreen durch Auswahl aus der dropdown-Liste zuordnen. Wenn Sie einen neuen Raum anlegen möchten, müssen Sie diesen dem ersten relevanten Gerät durch Eingabe des Befehls in das Kommandofeld zuordnen. Allen weiteren Geräten kann dann auch dieser neue Raum wieder über Auswahl aus der Dropdown-Liste zugeordnet werden.

Wie bereits erklärt, gibt es drei Möglichkeiten zur Konfiguration von fhem:

- Über die Detail-Sicht der einzelnen Geräte
- Über die fhem Kommandozeile
- Über Einträge direkt in der Datei fhem.cfg

Wichtig:

Bei Nutzung der Varianten 1 und 2 werden Ihre Änderungen erst dann in die Datei `fhem.cfg` geschrieben, wenn Sie über die Kommandozeile den Befehl `save` ausführen.

Da der Weg 1 empfohlen wird, werden Sie für gewöhnlich nicht direkt in `fhem.cfg` arbeiten. Dennoch ist es wichtig, die Syntax und den Inhalt zu verstehen.

fhem-Konfiguration: Das besondere Gerät „Global“

Nachdem Sie gesehen haben, wie Attribute gehandhabt werden, können wir uns nun einem ganz besonderen Gerät zuwenden: dem Gerät „Global“. Es wird grundsätzlich ebenso behandelt wie alle anderen Geräte in `fhem`, sendet jedoch keine Funknachrichten aus, sondern dient als Speicherort für die grundlegende Konfiguration in `fhem`. Dabei werden alle Einstellungen über Attribute vorgenommen.

Hinweis:

Änderungen an Geräten werden erst dann in der Konfiguration gespeichert, wenn man im Kommandofeld den Befehl `save` eingibt und dann `<ENTER>` drückt. Verwendet man diesen Befehl nicht, sind alle Eingaben beim nächsten Neustart verloren.

Hinweis:

Für Details lesen Sie den [Anhang: Ein Einblick in die Konfigurationsdatei `fhem.cfg`](#). Ein direktes Bearbeiten dieser Datei ist für gewöhnlich nicht erforderlich – ebenso wenig wie das direkte Bearbeiten der Registry eines Windows-Systems.

The screenshot shows the Fhem web interface. On the left is a sidebar with navigation options like 'Floorplans', '01_Wohnung', '60_Wetter', etc. The main area displays the configuration for a device named 'global'. Under the 'Internals' section, the following attributes are visible:

- DEF: HMLAN_MSGCNT 5, HMLAN_RAWMSG RBA45A744.0001,02481A66,FF,FFC1,1D8002195CBE7853C3010100003E, HMLAN_RSSI -63, HMLAN_TIME 2014-01-13 07:24:17, LASTInputDev HMLAN, MSGCNT 5, NAME global, NR Global, STATE <no definition>, TYPE Global

Below the 'Internals' section, there are dropdown menus for 'attr' (set to 'global') and 'room' (set to '01_Wohnung'). The 'Attributes' section lists various system and user-defined attributes:

- autoload_undefined_devices 1
- backup_before_update 0
- configfile fhem.cfg
- holiday2we Bayern
- logfile /var/InternetSpeicher/Intenso-MicroLine-01/fhem/log/fhem-%Y-%m.log
- modpath /var/InternetSpeicher/fhem
- mold Util-fhem
- mseclog 1
- room 99_System_System
- sendStatistics onUpdate
- statefile /var/InternetSpeicher/Intenso-MicroLine-01/fhem/log/fhem.save
- uniqueID /var/InternetSpeicher/fhem/FHEM/FhemUtils/uniqueID
- userattr Wohnung Wohnung_map devStateIcon devStateStyle fm_fav fm_groups fm_name fm_order fm_type fm_view fp_Grundriss fp_Media fp_Peter fp_PlotsPage fp_dark icon room_map setList sortBy attrExclude webCmd
- verbose 3
- version \$Id: fhem.pl 4803 2014-01-10 12:28:58Z rudolfkoenig \$

Das Gerät „global“ zeigt u.a. den Namen des aktuellen logfile's an

Die Attribute des Geräts „global“ steuern das Systemverhalten.

Die wichtigsten Attribute im Überblick:

Attribut	Beschreibung
modpath	Gibt an, unter welchem Pfad des Betriebssystems sich der Ordner fhem befindet. Der Pfad hängt von dem Betriebssystem und häufig auch der Hardware ab, auf der fhem installiert wurde. attr global modpath /var/InternerSpeicher/fhem (hier muss natürlich der zu Ihrem Gerät passende Pfad stehen!) Wenn dieses Attribut in Ihrer fhem.cfg –Datei bereits eingestellt ist und fhem funktioniert, lassen Sie diese Angabe am besten unverändert.
logfile	Ebenso wichtig ist die Pfadangabe zur fhem Logdatei: attr global logfile /var/InternerSpeicher/fhem/log/fhem-%Y-%m.log (Hier muss natürlich der zu Ihrem Gerät passende Pfad stehen!). Durch die Angabe %Y-%m wird eine separate Datei pro Monat pro Jahr angelegt – wie weiter oben im Internal-Wert „currentlogfile“ zu sehen ist. Das erlaubt eine zielgerichtete Archivierung, da die Logdateien je nach Anzahl und Art der verwendeten Geräte recht schnell recht groß werden können. Im Beispiel ist zu sehen, dass die Logdateien auf einen USB-Stick ausgelagert wurden.
statefile	Neben der fhem.cfg spielt auch die Datei fhem.save eine zentrale Rolle. In dieser Datei werden die Schaltzustände aller Geräte gespeichert, wenn Sie save ausführen– so werden alle Geräte nach einem Neustart von fhem wieder mit ihrem letzten Schaltzustand angezeigt. Der Speicherort dieser Datei ist festgelegt durch attr global statefile /var/InternerSpeicher/fhem/log/fhem.save
verbose	Legt fest, wie viele Meldungen im Logfile protokolliert werden sollen: verbose 1 logt nur Katastrophen, verbose 5 jeden kleinsten Schritt. verbose 3 ist der default, mit dem ein „gesundes Mittelmaß“ mitgeloggt wird.

Eine vollständige (lange) Liste aller global-Attribute finden Sie in der commandref <http://fhem.de/commandref.html#global>.

Geräte-Infos: Readings

Einige Geräte stellen zusätzliche Informationen zur Verfügung, sogenannte Readings. So wird z.B. bei einem Bewegungsmelder der Zeitpunkt des letzten Auslösens angezeigt,

bei einem Temperatur- und Luftfeuchte-Sensor (S300TH) stehen eben die zusätzlichen Readings Temperatur, Luftfeuchtigkeit und Zeitpunkt der letzten Übermittlung zur Verfügung. Auch diese Readings werden im Detail-Bildschirm eines Geräts angezeigt.

Hier als Beispiel ein Temperatur- und Luftfeuchte-Sensor:

Fhem

[Floorplans](#)

[CUL_WS](#)

[Esszimmer](#)

[Korridor](#)

[Kueche](#)

[Media](#)

[Plots](#)

[Schlafzimmer](#)

[Unsorted](#)

[Wohnung](#)

[Wohnzimmer](#)

[Everything](#)

[Howto](#)

[Wiki](#)

[Details](#)

[Edit files](#)

[Select style](#)

[Event monitor](#)

CODE	1
CUL_MSGCNT	86
CUL_RAWMSG	K0104617406
CUL_RSSI	-71
CUL_TIME	2012-03-12 11:46:33
DEF	1
LASTIODev	CUL
MSGCNT	86
NAME	ez_Aussensensor
NR	203
STATE	T: 10.4 H: 74.6
TYPE	CUL_WS
corr1	0
corr2	0
corr3	0
corr4	0

Readings

DEVFAMILY	WS300	2012-03-12 11:46:33
DEVTYPE	S300TH	2012-03-12 11:46:33
humidity	74.6	2012-03-12 11:46:33
state	T: 10.4 H: 74.6	2012-03-12 11:46:33
temperature	10.4	2012-03-12 11:46:33

ez_Aussensensor

fm_order	15	deleteattr
model	S300	deleteattr
room	Wohnung,Esszimmer	deleteattr

[Select icon](#)

[Device specific help](#)

Mehrere Geräte mit einem Klick schalten - structure

Zum Schalten von Gerätegruppen gibt es grundsätzlich drei Möglichkeiten:

- Die Einrichtung von Gruppen innerhalb des Hardwaresystems. Diese Gruppen sind auch von fhem aus bedienbar. Die Vorgehensweise ist in den jeweiligen

Handbüchern beschrieben.

- Die Zuordnung der zu gruppierenden Geräte zu einem Makro (siehe notify später in diesem Dokument)
- Die Einrichtung von Gruppen innerhalb fhem. Dieser Weg ist in diesem Kapitel beschrieben.

Wie bereits erwähnt liegt der wesentliche Unterschied im erzeugten Funkverkehr: Während für das Schalten einer im Hardwaresystem gebildeten Gruppe das Senden nur eines Funktelegramms erfordert, wird beim Schalten einer in fhem gebildeten Gruppe ein Funktelegramm je Gerät gesendet.

Um alle Geräte einer Gruppe (also z.B. alle Geräte im Wohnzimmer) mit nur einem Klick gemeinsam schalten können zu können, legen Sie in der fhem.cfg die Gruppe fest:
define wz_LichtAlle structure room wz_Lampe wz_MediaServer.

Nach dem Speichern wird nun auch die Struktur wz_LichtAlle mit ihren ON und OFF-Schaltern angezeigt und ist bedienbar. Wenn Sie auch diese Struktur dem Raum Wohnzimmer zuweisen, wird die Struktur auch in diesem Raum angezeigt:
attr wz_LichtAlle room Wohnzimmer

Dabei wird diese Festlegung nun auf alle der Struktur zugehörigen Objekte vererbt (wz_Lampe und wz_MediaServer werden ebenfalls dem Raum Wohnzimmer zugeordnet). Details zur Einrichtung finden Sie [hier \(http://fhem.de/commandref.html#structure\)](http://fhem.de/commandref.html#structure).

Timer

Viele Aktoren beherrschen auch eine timer-Funktion, so z.B. die Funksteckdose FS20-st oder der Dimmer FS20-di.

In fhem stehen hierfür mehrere Befehle zur Verfügung. Ein Beispiel ist
set lamp1 on-for-timer 10
(lamp1 für 10 Sekunden einschalten, dann wieder ausschalten). Analog gibt es den Befehl off-for-timer.

Die Zeitangaben für timer-Befehle akzeptieren jeglichen ganzzahligen Wert in Sekunden. Der timer verfügt leider nur über [112 Werte \(http://www.fhemwiki.de/wiki/FS20_Allgemein\)](http://www.fhemwiki.de/wiki/FS20_Allgemein), mit denen sich in wachsender Abstufung Zeiträume von 0,25 Sekunden bis 4,25 Stunden abbilden lassen. (Die Auflösung beträgt 0,25s bei 0 bis 4s, 0,5s von 4 bis 8s, 1s von 8 bis 16s usw. Für bessere Genauigkeit bei großen Werten verwenden Sie

besser at) Ggf. wird Ihr Wert automatisch umgesetzt auf den nächsten Passenden Wert – in diesem Fall erscheint eine Meldung im Log.

Da der Timer innerhalb des Aktors abgearbeitet wird, und der letzte an ihn gesendete Befehl ein ‚on-for-timer‘ ist, würde nun das Glühbirnen-Symbol im Webfrontend dauerhaft auf ON stehenbleiben. Um dies zu verhindern gibt es das Attribut `follow-on-for-timer`.

Damit aktivieren sie diesen Modus für Ihr Gerät. Setzen Sie also z.B.

```
attr lampe1 follow-on-for-timer 1
```

(die 1 steht in diesem Fall nicht für eine Sekunde, sondern für "aktiv")

Details und weitere Beispiele sind in der [fhem-Referenz \(http://fhem.de/commandref.html#FS20\)](http://fhem.de/commandref.html#FS20) und [im Wiki \(http://www.fhemwiki.de/wiki/FS20_Allgemein\)](http://www.fhemwiki.de/wiki/FS20_Allgemein) beschrieben.

Dimmer

Hinweis von pi3g:

Bitte beachten Sie zum Dimmen die jeweilige Dokumentation Ihrer Z-Wave Aktoren und Sensoren, beispielsweise von Wandschaltern.

Als spezielle Aktoren gibt es auch Dimmer, z.B. FS20-di. Diese lassen sich wie Schalter mit on/off/toggle bedienen, beherrschen aber einige zusätzliche Befehle: `dimup`, `dimdown`, `dimupdown` sowie die Einstellung auf vorgegebene Dimmstufen, z.B. `dim50%`. Die vollständige Liste der 16 Dimmstufen finden Sie in der [fhem-Referenz](#).

Wenn Sie an einem Sensor (Schalter) den Taster länger als 0,4 Sekunden gedrückt halten, sendet dieser statt on/off/toggle den Befehl `dimup/dimdown/dimupdown`. So lassen sich wiederum ohne Programmierung auch Dimm-Vorgänge erreichen.

Ein Dimmen aus dem fhem-Webfrontend ist über eingetippte Befehle möglich, Sie können also z.B. `set lampe1 dimup` oder `set lampe1 dim50%` eingeben.

Eine besondere Variante via fhem ist die Einstellung eines gleichmäßigen Dimm-Vorgangs über einen Zeitraum. So kann z.B. die Nachtschlampe zum `wakeup-light` werden, wenn sie über ca. 20 Minuten (1280 Sekunden) langsam heller wird:

```
set lampe1 dim100% 1280
```

Die Funktionen des Wakeuplight lassen sich beliebig erweitern. So kann man den o.g.

Befehl z.B. durch einen at-Befehl (siehe nächstes Kapitel) auslösen lassen und über eines der Multimedia-Module (siehe ebenfalls weiter in diesem Handbuch) mit der Wiedergabe von Musik und dem Hochfahren der Rolläden kombinieren.

Schalten zu bestimmten Zeitpunkten – at

Ein häufiger Anwendungsfall ist das Schalten von Geräten zu festgelegten, sich ggf. wiederholenden Zeitpunkten. Dazu steht in fhem der Befehl `define...at` zur Verfügung:
`define <name> at <timespec> <command> also z. B.`

`define LampeAnUm1700 at 17:00:00 set lamp on`

(Um eine solche "Zeitschaltuhr" anzulegen, tippen Sie den o.g. Befehl einfach im fhem webfrontend in das Kommandozeilen-Feld ein.)

LampeAnUm1700 ist hierbei lediglich ein Name/Platzhalter, unter dem diese „Zeitschaltuhr“ in fhem gespeichert wird und später wiedergefunden werden kann. Sobald Sie den oben genannten Befehl eingeben, erscheint dieses geplante Ereignis unter ‚Everything‘ im Abschnitt ‚at‘. Ist die angegebene Uhrzeit erreicht, wird die Anweisung genau einmal ausgeführt. Damit ist das Ereignis erfolgreich abgearbeitet und wird gelöscht.

Auch Wiederholungen sind möglich, z.B. täglich um 17:00. Dazu wird der Uhrzeit ein * vorangestellt:

`define LampeTaeglichAn at *17:00:00 set lamp on`

Der **modify**-Befehl dient dazu, den Zeitpunkt zu ändern, ohne den Befehls-Teil erneut angeben zu müssen:

`modify LampeTaeglichAn *17:30:00`

Über die absolute Angabe von Schaltzeiten (um 17:00) hinaus erlaubt der at-Befehl auch relative Angaben. Dies wird durch ein vorangestelltes Plus-Zeichen erreicht. Also „in 10 Minuten“:

`define a5 at +00:10:00 set lamp on`

Die Kombination von + und * bewirkt dann z.B. „alle 10 Minuten“, hier für 30 Sekunden:

`define a6 at +*00:10:00 set lamp on-for-timer 30`

Für solche Durchläufe kann auch die gewünschte Anzahl der Wiederholungen in geschweiften Klammern angegeben werden:

`define a7 at +*{3}00:00:02 set lamp on-for-timer 1 # drei mal blinken`
(drei mal alle 2 Sekunden für 1 Sekunde einschalten).

In der fhem-Referenz sind hierzu viele Beispiele genannt, die auch Abhängigkeiten von Wochenenden, Sonnenauf- und -untergängen aufzeigen – dies erfordert einfache Programmierung und wird z.T. später in diesem Dokument aufgezeigt. Diese Infos finden Sie [hier \(http://fhem.de/commandref.html#at\)](http://fhem.de/commandref.html#at).

Schalten von Ereignissen abhängig machen - notify

Vor allem bei indirekten Schaltungen möchten Sie erreichen, dass ein Ereignis (z.B. Drücken einer Taste) das Schalten eines Aktors nach sich zieht. Die Kopplung erfolgt so:

```
define <name> notify <pattern> <command>  
also z.B.
```

```
define Schalter1Notify notify Schalter1 set wz_Media on
```

Der bei define...notify angegebene Name bezeichnet den ‚Event-Handler‘. Die besondere Eigenschaft dieser notify-Anweisungen ist, dass sie nicht zum Zeitpunkt der Eingabe oder zu einer festgelegten Uhrzeit ausgeführt werden, sondern im Hintergrund ‚mitlauschen‘. Sobald der nach notify angegebene Sensor (Schalter 1) einen Funkbefehl übermittelt, wird die im notify angegebene Anweisung ausgeführt. Die oben dargestellte Zeile schaltet also wz_Media an, sobald der Taster Schalter1 betätigt wird.

Häufig möchten Sie mit nur einem Klick mehrere Geräte schalten. In diesem Fall können Sie als <command> auch eine Liste von Geräten angeben (siehe devspec beim Befehl set) oder eine Struktur schalten (siehe structure).

Allerdings ist das o.g. Beispiel -so wie es da steht- ein Anwendungsfall vornehmlich für die Einstellung als 4-Kanal-Schalter, also Tastern: das Gerät soll beim Betätigen des Schalter 1 immer eingeschaltet werden. Für das Ausschalten würde man also analog Schalter 2 mit dem off-Befehl koppeln:

```
define Schalter2Notify notify Schalter2 set wz_Media off
```

Ist Ihr Schalter als 2-Kanal definiert, können Sie herausfiltern, ob der on- oder der off-button gedrückt wurde:

```
define Schalter1NotifyOn notify Schalter1:on set wz_Media on
```

oder Sie schalten das Licht aus, wenn Media eingeschaltet wird – und umgekehrt:

```
define Schalter1NotifyAn notify Schalter1:on set wz_Media on;;set wz_Licht off  
define Schalter1NotifyAus notify Schalter1:off set wz_Media off;;set wz_Licht on
```

Hinweis:

Bei Aufzählungen keine Leerstellen vor und nach den Semikola!

Als kürzere Alternative zu

```
define Schalter1NotifyOn notify Schalter1:on set wz_Media on  
define Schalter1NotifyOff notify Schalter1:off set wz_Media off
```

verwenden Sie die Variable \$EVENT:

```
define Schalter1Notify notify Schalter1 set wz_Media $EVENT  
define Schalter1Notify notify Schalter1 set wz_Media % (alte Schreibweise)
```

Dabei hat \$EVENT genau den Wert, der vom abgefragten Sensor (also Schalter 1) gesendet wurde. Sendet also Schalter 1 den Wert on, so wird der Befehl set wz_Media on ausgeführt. Sendet der Schalter 1 den Befehl off, so wird set wz_Media off ausgeführt. Analog funktioniert das für jeden Befehl des Sensors, also auch toggle, dimup, dimdown etc.

Eine Beschreibung mit vielen Beispielen gibt's im [fhemWiki \(http://www.fhemwiki.de/wiki/Funktionsbeschreibung_zu_notify\)](http://www.fhemwiki.de/wiki/Funktionsbeschreibung_zu_notify).

Verwendung von notify als Makro

Als Makro bezeichnet man das Ausführen mehrerer Befehle auf Grund nur eines auslösenden Ereignisses (also z.B. dem Drücken einer Taste, dem Klick auf einen Weblink o.ä.). In diesem Sinne lässt sich notify als Makro verstehen, wenn Sie wie oben als <command> mehrere Befehle – durch Semikola getrennt- angeben:

```
define Schalter1NotifyAn notify Schalter1:on set wz_Media on;;set wz_Licht off
```

Das Abarbeiten eines notify (Makros) wird gestartet, sobald das <pattern> eintritt.

Starten eines Makros – trigger

Ein solches Ereignis muss nicht durch ein Funktelegramm ausgelöst werden, sondern kann auch durch den Befehl trigger gestartet werden.

Wenn z.B. abends in jedem Raum eine Lampe eingeschaltet werden soll:

```
define Abends notify Abends set wz_LampeKlein on;;set sz_Stehlampe on;;set ku_Downlight on  
können Sie den Vorgang starten durch  
trigger Abends
```

Diese Möglichkeit ist besonders zum Testen neuer notify-Makros hilfreich. Eine umfangreiche Darstellung der Möglichkeiten von notify finden Sie in [diesem Wiki-Artikel \(http://www.fhemwiki.de/wiki/Funktionsbeschreibung_zu_notify\)](http://www.fhemwiki.de/wiki/Funktionsbeschreibung_zu_notify).

Bearbeiten über das Webfrontend

Wenn Ihre notify-Definition bzw. Ihr Makro einmal erstellt ist, können Sie es im Webfrontend bearbeiten. Klicken Sie dazu im Menü auf „Everything“ und scrollen dann zum Abschnitt notify. Durch Klicken auf den Namen des notify gelangen Sie in den

Detail-Bildschirm, den Sie ja von Geräten bereits kennen:

The screenshot shows the web frontend interface for editing a notify definition. The interface is divided into a sidebar on the left and a main content area on the right. The sidebar contains a navigation menu with options like 'Fhem', 'Floorplans', 'CUL_WS', 'Esszimmer', 'Korridor', 'Kueche', 'Media', 'Plots', 'Schlafzimmer', 'Unsorted', 'Wohnung', 'Wohnzimmer', 'Everything', 'Howto', 'Wiki', 'Details', 'Edit files', 'Select style', and 'Event monitor'. The main content area displays the configuration for a notify definition named 'Abends'. The definition text is 'Abends set wz_LampeKlein on;set sz_Stehlampe on;set ku_Downlight On'. Below the text is a 'modify Abends' button. A table shows the details of the notify definition: NAME: Abends, NR: 161751, NTFY_ORDER: 50-Abends, REGEXP: Abends, STATE: active, TYPE: notify. At the bottom, there is an 'attr' field with 'Abends Audio' selected and a 'save' button.

Wenn Sie auf DEF klicken, können Sie die Definition Ihres notify in einem Textfeld bearbeiten – z.B. wenn Sie ein weiteres Gerät zu Ihrem Makro hinzufügen möchten.

Hinweis:

Im DEF-Textfeld erscheinen nur <pattern> und <command>. Der Name des notify steht unterhalb des Textfensters.

Beenden Sie die Bearbeitung mit Klick auf die Schaltfläche modify <name> .

Hinweis:

Ihre Änderung wird erst dauerhaft in die Konfigurationsdatei fhem.cfg übernommen, wenn Sie nach dem Klick auf modify <name> über das Kommandofeld den Befehl save ausführen.

Pairing: direkt oder indirekt

Hinweis von pi3g:

ZWave verfügt statt über Hauscodes und Tastencodes über home-ID und nodeID, die von dem System automatisch verwaltet werden. Die Ausführungen über direkte Schaltung (assoziation von Z-Wave Geräten) oder indirekte Schaltung (Steuerung via FHEM) treffen jedoch zu.

In der bisher beschriebenen Konfiguration ist jeder Aktor direkt mit einem Sensor gekoppelt.

Nachdem Sie nun notify kennengelernt haben, können Sie auch eine „indirekte“ Kopplung vornehmen.

Eine direkte Kopplung (Pairing) hat den Vorteil, dass die Steuerung der Aktoren direkt vom Sensor vorgenommen wird und fhem zumeist nur ‚mitlauscht‘. Sobald also der Sensor – z.B. nach einem Tastendruck auf den Wandschalter oder dem Auslösen eines Bewegungssensors - ein Funktelegramm sendet, wird der Aktor direkt durch das vom Sensor gesendete Signal angesteuert und reagiert sofort. Eine Zeitverzögerung z.B. wegen Auslastung oder gar Nicht-Verfügbarkeit des fhem-Systems tritt nicht auf.

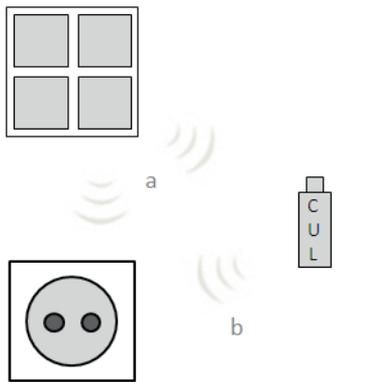
Das ist durchaus wichtig, denn schon eine Zeitverzögerung von einer halben Sekunde zwischen Drücken des Lichtschalters und Einschalten des Lichts kann sehr irritierend wirken, da man gewohnt ist, dass das Licht nach Betätigen des Lichtschalters eben sofort angeht. Geschieht dies nicht, vermuten viele einen Fehler und drücken den Schalter gleich noch mal...

Im Rahmen der Hausautomatisierung gibt es aber häufig den Fall, dass nach dem Drücken eines Schalters erst noch geprüft werden soll, welche Aktion nun auszuführen ist – ist es vielleicht Nacht und es soll statt des grellen Korridorlichts nur eine kleinere Lampe eingeschaltet werden? Daher gibt es auch die Alternative, den Aktor separat vom Sensor zu definieren - also nicht durch das Anlernen auf Basis eines Sensor-Funktelegramms. Sensor und Aktor verwenden in diesem Fall NICHT DIE SELBE Kombination von Hauscode und Tastencode; vielmehr wird in fhem ein device angelegt mit einer ANDEREN Adresse (Kombination Hauscode+Tastencode) als der des Sensors.

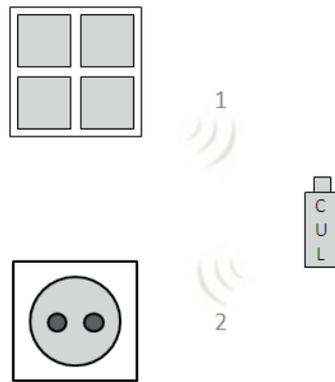
Auf diese separate fhem-Adresse wird der Aktor angelernt. Ein Funktelegramm eines Sensors (Lichtschalter) wird dann also von keinem Aktor umgesetzt, da kein Aktor auf dieses Funktelegramm angelernt wurde. CUL+fhem ‚hören‘ das Funktelegramm (und lernen –falls noch nicht geschehen- den Sensor über autcreate an). In fhem kann man nun mittels notify eine beliebige Reaktion auf dieses Funktelegramm,

also auf das eingetretene Ereignis (das Drücken einer Taste oder das Auslösen eines Bewegungsmelders) definieren. Je nach im notify definierter resultierender Aktion sendet fhem ggf. ein separates Funktelegramm mit einer anderen Adresse, auf die ein Aktor angelernt ist und seinerseits reagiert. Ähnliches gilt, wenn ein Aktor gar nicht von einem physischen Schalter, sondern ausschließlich aus dem Web-Frontend getriggert (engl. „ausgelöst“) werden soll: Eine bestimmte Kombination von Hauscode und Tastencode wird an keinem Sensor eingestellt, sondern nur in einem fhem-device definiert, um dann einen Aktor auf das fhem-Funktelegramm anzulernen, das nach Klick auf ‚on‘ oder ‚off‘ im webfrontend vom CUL gesendet wird.

Direkt vs. Indirekt



Direkt: Der Aktor ist direkt an den Sensor gekoppelt, Schaltbefehle werden sofort ausgeführt (a). Der CUL ‚lauscht dabei mit‘ (a) und wird nur bei Nutzung des Webinterfaces ein Schaltsignal senden (b). Die Adressen (Kombination von Hauscode und Tastencode) des Sensors und des fhem-device sind identisch.



Indirekt: Der Sensor sendet an den CUL (1), fhem reagiert und sendet daraufhin einen Schaltbefehl an den Aktor (2), ebenso bei Nutzung des Webinterfaces (2). Die Adresse (Kombination von Hauscode und Tastencode) des Sensors ist nicht dieselbe wie die des fhem-device.

Für indirektes Pairing muss ein Aktor ‚manuell‘ in fhem eingerichtet werden. Hierfür wird der Befehl `define...FS20` verwendet:

```
define <devicename> FS20 <Hauscode> <Tastencode> [fg][lm][gm] , also z.B.
define lampe1 FS20 12341234 01
```

Als Hauscode verwenden Sie hier

- Denselben Hauscode wie für Ihr Hardwaresystem gewählt, wenn Sie Schaltgruppen (Funktionsgruppe (fg), Lokaler Master (lm) oder Globaler Master (gm)) über die Adressierung des Hardwaresystems nutzen möchten
- Einen separaten Hauscode für alle ‚nur-fhem-devices‘, wenn eine Einbindung des devices in Gruppenschaltungen des Hardwaresystems nicht vorgesehen ist.

Gruppen können in fhem mittels `structure` oder `notify` über unterschiedliche Hauscodes hinweg definiert werden, das Schalten von Gruppen erzeugt dann aber mehr ‚Funklast‘.

Außerdem ist beim indirekten Pairing eine fhem-‚Kopplung‘ zwischen Sensor und Aktor erforderlich, da fhem zunächst nicht wissen kann, welcher Sensor welchen Aktor schalten soll. Für diese ‚Kopplung‘ wird der Befehl `define...notify` verwendet. Wenn also z.B. durch das Drücken von Schalter1 der Aktor `lampe1` eingeschaltet werden soll, so wird dies erreicht durch

```
define <name> notify <pattern> <command> , also z.B.  
define S1notify notify Schalter1 set lampe1 on
```

In diesem Beispiel waren Sensor und Aktor aus demselben Hardwaresystem FS20. Der Aktor `lampe1` kann aber ebenso gut aus einem anderen Hardwaresystem stammen – man kann also z.B. einen FS20-Wandschalter als Sensor verwenden, und dadurch über `notify` einen HomeMatic- oder 1-wire-Aktor schalten. Ebenso funktioniert es natürlich auch umgekehrt – ein EnOcean-Sensor kann eine FS20- oder Intertechno-Steckdose schalten.

Anpassen der Darstellung im Webfrontend - FHEMWEB-Attribute

Das fhem Webfrontend ist flexibel und funktional. Es gestattet, die Darstellung den eigenen Bedürfnissen anzupassen. Das Webfrontend ist in fhem über das (Pseudo-)Gerät FHEMWEB realisiert. Wie in der Liste der Geräte (Raum „Everything“) zu sehen ist, können Sie mehrere Geräte vom Typ FHEMWEB definieren und diesen Attribute zuordnen. Eine vollständige Liste aller Attribute steht in der comandref [hier \(http://fhem.de/commandref.html#FHEMWEB\)](http://fhem.de/commandref.html#FHEMWEB).

Ein Konfigurationsbeispiel mit screenshot und zugeordneten Attributen gibt's am Schluß dieses Kapitels.

Bildschirmgröße und „Skin“: stylesheetPrefix

Der Port **8083** wird dem Gerät WEB zugeordnet. Dieser ist erreichbar unter <http://<ip>:8083/fhem> `define WEB FHEMWEB 8083 global` erzeugt eine für PC-Bildschirme angepasste Darstellung. Den gewünschten „Skin“ können Sie unter dem Menüpunkt „Select Style“ auswählen. Jeder Style ist die

Kombination des Farbschemas (default, dark, iOS7) und der Optimierung auf die jeweilige Bildschirmgröße (default, smallscreen, touchpad).

Um eine Darstellung mit dunklem Hintergrund zu erzeugen, wählen Sie unter „Select style“ aus der Liste „dark“ aus – dadurch wird im Hintergrund gesetzt `attr WEB stylesheetPrefix dark` und dadurch zur Formatierung `darkstyle.css` verwendet.



Port 8084 (WEBphone) ist das Attribut `stylesheetPrefix smallscreen` zugeordnet. Dadurch wird die Darstellung für Geräte mit kleinem Bildschirm angepasst. Die Liste der Räume erscheint dafür nicht mehr am linken Bildschirmrand, sondern als Drop-Down-Liste oben.

`define WEBphone FHEMWEB 8084 global`
`attr WEBphone stylesheetPrefix smallscreen`
Im Hintergrund wird hier zur Formatierung die Datei `smallscreenstyle.css` verwendet. Analog existieren auch `darksmallscreen.css` und `iOS7smallscreen.css`.

Port 8085 (WEBtablet) trägt den prefix `touchpad`. Hierdurch werden Schriftgröße und Darstellungsart für Tablet-PCs optimiert.

`define WEBtablet FHEMWEB 8085 global`
`attr WEBtablet stylesheetPrefix touchpad`

Besonderheit

Für iPhone und iPad ist die Darstellung in `smallscreen` und `touchpad` so gestaltet, dass das Webfrontend wie eine Fullscreen-„App“ genutzt werden kann. Dazu öffnen Sie `fhem` in Safari mit dem entsprechenden port (also z.B. `http://<ip>:8085/fhem`). Klicken Sie auf den Optionen-button  und wählen Sie „Zum Homebildschirm“. Dadurch erhalten Sie ein neues Icon, mit dem Sie diese `fhem`-Seite mit nur einem Klick starten können. Die Leiste für URL und Websuche werden dann nicht angezeigt, so dass sich `fhem` verhält wie eine vollwertige App.

Weniger Menüpunkte: hiddenroom

Es besteht außerdem die Möglichkeit, aus dem Standard Webfrontend einige Menüpunkte aus dem unteren Menüblock auszublenden.

`attr WEB hiddenroom save,Howto,FAQ,Examples,Unsorted,„Select style“`

Ebenso kann man den Zugang zum Detailview der Geräte blockieren (detail), das Kommandofeld unterdrücken (input) und den save-Button ausblenden (save). Dies kann mit dem Attribut hiddenroom erreicht werden.

Hinweis:

Verwenden Sie eine ältere fhem-Version, ist insbesondere das Ausblenden des „save“-Buttons für Anfänger empfohlen, da er durch seine Position direkt neben dem Kommando-Eingabefeld immer wieder zu Verwirrung führt. Mit diesem Button wird nicht die Eingabe eines Befehls bestätigt (dazu reicht das drücken der <ENTER>-Taste), sondern es wird die aktuelle Konfiguration gespeichert, damit sie nach dem nächsten Neustart von fhem wieder verfügbar ist. Nach dem Ausblenden dieses Buttons kann das Speichern dadurch erfolgen, dass man den Befehl save in das Kommandofeld eingibt und <ENTER> drückt (oder bei neueren Versionen im fhem-Menü auf „save config“ klickt).

Zusätzliche Menüpunkte – menuEntries



Auch ist es möglich, dem fhem-Menü eigene Menüpunkte hinzuzufügen durch Verwendung des Attributs menuEntries. Hinterlegt wird eine komma-getrennte Liste von Paaren; zunächst die Bezeichnung wie sie auf dem Bildschirm erscheinen soll (z.B. restart) gefolgt von dem auszuführenden Befehl beginnend mit cmd= und dann dem fhem-Kommando. Leerstellen im Kommando werden dabei mit + aufgefüllt (z.B. cmd=shutdown+restart):

attr WEB menuEntries

restart,cmd=shutdown+restart,update,cmd=update,updatecheck,cmd=update+check,reloadMyUtils,cmd=reload+99_myUtils.pm

Nur bestimmte Befehle für ein Gerät - webCmd

Das Attribut webCmd beeinflusst die Anzahl der im webfrontend angezeigten Befehle. Ein FS20-Gerät hat z.B. ca. 25 mögliche Befehle (on, off, dimup, dimX%, toggle, on-for-timer, ...). Mit webCmd wählen Sie aus, welche davon im webfrontend angezeigt werden (wenn Sie webCmd nicht angeben, ist der default on,off).

attr Lampe webCmd on:off:on-for-timer 600

Befehle umbenennen - eventMap

Mit eventmap können Sie den für das webfrontend verwendeten Text eines Kommandos bzw. Schaltzustandes anpassen (dem Gerät sz_Rollo wurden eigene Icons zugeordnet,

deshalb sehen Sie hier nicht die gewohnte Glühlampe, sondern ein offenes Fenster – siehe nächster Abschnitt)

Vorher:



```
attr sz_Rollo eventMap on:Down off:Up
```

```
attr sz_Rollo webCmd Down:Up
```

Nachher:



Räume sortieren: sortRooms

Standardmäßig werden die Räume in alphabetischer Sortierung angezeigt. Dies kann mit dem Attribut `sortRooms` (<http://fhem.de/commandref.html#FHEMWEB>) geändert werden:

```
attr WEB sortRooms Wohnzimmer Schlafzimmer Media Wetter Heizung Systeme
```

Gruppen bilden und umbenennen: group

Standardmäßig werden in fhem Geräte nach ihrem TYPE sortiert, so gibt es Gruppen wie FS20, FHT usw. Möchten Sie eigene Gruppen bilden, funktioniert das mit dem Attribut `group` (<http://fhem.de/commandref.html#group>), das sich ähnlich wie `room` verwenden lässt. Eine Gruppe muss also nicht separat definiert werden, sondern entsteht dadurch, dass sie dem ersten Gerät zugeordnet wird.

```
attr ez_FHT group Heizung
```

```
attr ez_Aussensensor group Heizung
```

erzeugt also einen neuen Gruppe „Heizung“, die dann als separater Block in einem Raum dargestellt wird.

Mehrspaltige Darstellung: column

Standardmäßig erfolgt die Darstellung im Webfrontend einspaltig, es werden also alle Gruppen untereinander dargestellt. Um eine mehrspaltige Darstellung zu erreichen, legen Sie je Raum fest, welche Gruppen untereinander (getrennt durch Komma) bzw. nebeneinander (getrennt durch |) angeordnet werden. Das Attribut wird der Webinstanz zugeordnet:

```
attr WEB column Übersicht:Wohnung,Heizung,iTunes|weather
```

Durch Setzen dieses Attributs werden also im Raum „Übersicht“ die Gruppen Wohnung,

Heizung und iTunes untereinander dargestellt, in einer Spalte rechts daneben erscheint die Gruppe weather. Die Liste kann natürlich für weitere Räume erweitert werden.

Iconaktualisierung ohne browser-refresh – longpoll

Während man einen fhem-Raum im Browser anzeigt, kann ja eine Änderung des Schaltzustands eintreten, wenn z.B. ein Sensor auslöst oder ein Lichtschalter betätigt wird. In fhem wird dieser Schaltzustand in der Darstellung aktualisiert, ohne dass im Browser der „refresh“-button betätigt werden muss.

Möchte man dies nicht, kann man diesen sogenannten „longpoll“-Mechanismus für die Webinstanz deaktivieren:

```
attr WEB longpoll 0
```

Zur Anpassung der Darstellung im Webfrontend stehen weitere Attribute zur Verfügung. Eine vollständige Liste der Möglichkeiten findet sich in der [commandref \(http://fhem.de/commandref.html#FHEMWEB\)](http://fhem.de/commandref.html#FHEMWEB).

Welche Icons? stylesheetPrefix und iconPath

Zusammen mit fhem werden mehrere Icon-Bibliotheken ausgeliefert. Sie sind abgelegt unterhalb des Ordners fhem/www/images. Der Ordner default enthält die fhem Standard-Icons. In den Ordnern dark, smallscreen und tablet sind Icons abgelegt, die speziell zu der jeweiligen Darstellungsweise passen. Die Suchreihenfolge hängt dabei vom verwendeten stylesheetPrefix (siehe oben) ab: ist z.B. stylesheetPrefix dark gesetzt, werden Icons zuerst im Ordner dark gesucht. Werden sie dort nicht gefunden, wird die Suche im Ordner default fortgesetzt.

Der Ordner openautomation enthält eine Icon-Bibliothek des gleichnamigen Projekts. Diese Icons sind in SVG erstellt und lassen dadurch Größenskalierung und das Setzen von Farben zu. Die Liste ist sehr umfangreich, jedoch unterscheidet sich das Layout dieser icons stark von dem des fhem Standard.

Welche Icons verwendet werden sollen, kann mittels [iconPath \(http://fhem.de/commandref.html#iconPath\)](http://fhem.de/commandref.html#iconPath) eingestellt werden:

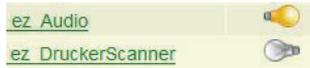
```
attr WEB iconPath default:openautomation
```

Ist die default-Einstellung, solange dieses Attribut nicht gesetzt ist. Mit

```
attr WEB iconPath openautomation:default
```

wird die Suchreihenfolge umgekehrt: wo vorhanden, werden die SVG-icons aus dem Ordner openautomation verwendet.

Vorher:



attr WEB iconPath openautomation:default

Nachher:



Probieren Sie es doch einfach mal aus.

Hinweis:

iOS 7 gestattet das Aktualisieren von jpg/png-icons mittels longpoll leider nicht. Möchten Sie also longpoll verwenden, stellen Sie auf die Darstellung mit openautomation-icons um.

Eigene Icons

fhem sieht außerdem die Möglichkeit vor, den Geräten andere bzw. eigene Icons zuzuweisen. Dabei kann grundsätzlich jedes Bild verwendet werden, es empfiehlt sich jedoch, sich an der Größe der bereits vorhandenen Icons zu orientieren. Legen Sie eigene Icons in den geeigneten Ordnern ab, siehe vorheriger Abschnitt.

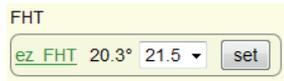
Icon vor dem Gerätenamen - icon

Klicken Sie in der Detail-Ansicht eines Geräts auf den Link ‚Select icon‘. Sobald Sie ein Icon auswählen, wird dem Gerät ein Attribut ‚icon‘ mit dem entsprechenden Dateinamen als Wert zugeordnet. Alternativ können Sie dieses Attribut natürlich auch manuell eingeben – oder es löschen, wenn kein Icon mehr verwendet werden soll.

Für die Liste der verfügbaren Icons werden von fhem alle Dateien mit dem Prefix ico angezeigt – benennen Sie Ihre eigenen Icon-Datei also ico*.jpg oder ico*.png. Icon-Bilddateien liegen im Icon-directory fhem/www/images/default.

Hier ein Beispiel mit und ohne Icon vor dem Gerätenamen:

Vorher:



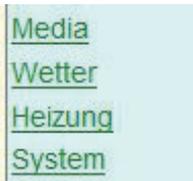
attr ez_FHT icon icoHEIZUNG.png

Nachher:



Icon vor dem Raumnamen - roomIcons

Auch vor einem Raumnamen im fhem-Menü kann ein icon angezeigt werden. Die Zuordnung erfolgt über das FHEMWEB-Attribut `roomIcons` (<http://fhem.de/commandref.html#roomIcons>): Angegeben werden durch: verbundene Paare von Raumname und icon, die Paare sind durch Leerstellen getrennt:



```
attr WEB roomIcons Media:audio_
eq Wetter:weather_cloudy
Heizung:sani_heating_temp
System:edit_settings
```

Hinweis:

Das ist eigentlich eine Zeile, zur korrekten Darstellung müsste die Schrift aber so klein eingestellt werden, dass man sie nicht mehr lesen könnte.



Eigene Icons für Geräte-Schaltzustände

Auch ist es möglich, an Stelle der grauen bzw. leuchtenden Glühlampe andere Icons anzuzeigen. Eine einfache Vorgehensweise ist, ein bestehendes Icon (z.B. FS20.ON.PNG) zu kopieren und die Kopie dann zu bearbeiten. Bitte beachten Sie, dass die Dateinamen 100%ig zu den Gerätenamen und Schaltzuständen passen müssen – auch Groß – und Kleinschreibung wird unterschieden.

Auch die hierfür verwendeten Dateien werden in den Ordnern unterhalb `fhem/www/images` abgelegt. Die Namenskonvention ist:

`<devicename>.<state>.[jpg|png|gif]`

Beispiel: `sz_Rollo.on.jpg` , `sz_Rollo.off.jpg`,

Werden diese Dateien nicht gefunden, verwendet fhem die Standard-Icons `<type>.<state>.[jpg|png|gif]` , also z.B. `FS20.on.png` , `FS20.off.png`

Die Suchreihenfolge ist dabei wie folgt:

- Suche nach einem Icon, das zum Gerätenamen passt
- Wird keines gefunden, wird ein Icon für den Geräte-TYPE gesucht (FS20, FHT, ...)

Jeder dieser Suchläufe erfolgt dabei auf Basis der Ordnerstruktur unterhalb des Dateipfads `fhem/www/images`.

- Eine Besonderheit der Suchreihenfolge ist die Berücksichtigung von `stylesheetPrefix`: ist hier z.B. „dark“ eingestellt, wird zunächst im Ordner `dark` gesucht, ob hier spezielle Icons zur Darstellung vor dunklem Hintergrund vorhanden sind. Wird dort kein Icon gefunden, wird die Suche gemäß `iconPath` fortgesetzt. Die Voreinstellung hierfür ist eine Suche zunächst im Ordner `default`, danach im Ordner `openautomation`.
- Wird kein icon gefunden, wird der aktuelle Schaltzustand nicht als Icon, sondern als Wort dargestellt.

Weblink



In das fhem-webfrontend können Hyperlinks und Web-Grafiken eingebunden werden. Hierzu ist das (Pseudo-)device weblink verfügbar. Details und Beispiele finden Sie in der commandref [hier](http://fhem.de/commandref.html#weblink) (<http://fhem.de/commandref.html#weblink>).

Ein Beispiel ist das Einblenden eines Wetterberichts (Achtung: EINE Zeile!):
define Wetter weblink iframe <http://www.wetteronline.de/cgi-bin/hpweather?PLZ=12345>

Zur Anzeige und Verwendung von Wetterdaten gibt es außerdem die fhem-Module [weather](http://fhem.de/commandref.html#Weather) (<http://fhem.de/commandref.html#Weather>) und [GDS](http://fhem.de/commandref.html#GDS) (<http://fhem.de/commandref.html#GDS>), siehe unter **Wetterbericht einbinden**.

Beispiel zur Konfiguration

The screenshot displays the fhem web interface with several key components:

- Left Sidebar (1):** Navigation menu including 'FHEM', 'Floorplans', 'Wohnung' (selected), and various room names like 'Esszimmer', 'Wohnzimmer', etc.
- Heizung (2):** Heating control panel showing a current temperature of 19.8°C and a desired temperature of 21.5°C. It includes controls for 'ez_FHT', 'hzg_CalorMaticSteuerung', 'hzg_HeizungFreigabeBrenner', and 'hzg_nachts'.
- Wohnung (6):** Apartment status panel showing 'HomeStatus' at 10:00, 'wakeUpChange', and 'whg_WLAN'.
- Music Player (7):** 'iTunes2' player showing 'Rivers (Sometimes) [Club Mix] (Peer Kusiv & Martin Jondo)'.
- weather (3):** Weather forecast panel showing a 7-day forecast with icons and temperature ranges.
- Bottom Graph:** A line graph showing 'Actuator (%)' and 'Temperature in C' over time, with 'Out: 3.2' and 'In: 19.8'.

①	Dunkler Hintergrund – Menüpunkt „Select style“ -> dark (FHEMWEB-Attribut stylesheetPrefix)
②	Bildung eigener Gerätegruppen – Geräte-Attribut group
③	Mehrspaltige Anordnung – FHEMWEB-Attribut column
④	Sortierung der Raumliste – FHEMWEB-Attribut sortRooms Menüpunkte ausgeblendet (z.B. Unsorted) – FHEMWEB-Attribut hiddenroom
⑤	Eigene Befehle – FHEMWEB-Attribut menuEntries
⑥	Geräte-Icons – Geräte-Attribut icon
⑦	Zustands-Icons – durch Erzeugen von icon-Dateien passend zum Gerätenamen und -Schaltzustand, hier z.B. whg_WLAN.on.png

Gruppen frei auf dem Bildschirm platzieren: Dashboard

Das Modul [Dashboard](http://fhem.de/commandref.html#Dashboard) (<http://fhem.de/commandref.html#Dashboard>) gestattet eine noch flexiblere Darstellung von Gruppen als das Attribut column. So können mittels Dashboard Gruppen in Tabs angeordnet und auf- und zugeklappt werden.

Einrichten eines Grundrisses mit eigenen fhem-Geräten

Einen Beispiel-Screenshot finden Sie auf dem Deckblatt dieses Dokuments. In fhem heißt so ein Grundriss FLOORPLAN, siehe dazu auch den [entsprechenden commandref-Eintrag](http://fhem.de/commandref.html#FLOORPLAN) (<http://fhem.de/commandref.html#FLOORPLAN>). In diesem commandref-Eintrag finden Sie auch den Link auf einen detaillierten [Einrichtungs-Leitfaden](http://sourceforge.net/p/fhem/code/HEAD/tree/trunk/fhem/docs/fhem-floorplan-installation-guide_de.pdf?format=raw) (http://sourceforge.net/p/fhem/code/HEAD/tree/trunk/fhem/docs/fhem-floorplan-installation-guide_de.pdf?format=raw).

Einfache Programmierung: if-Bedingung

Wir haben ja bereits gesehen, dass Schaltungen von bestimmten Kriterien abhängen können, also z.B. ob der on- oder off-Befehl gesendet wurde:

```
define Schalter1Notify notify Schalter1:off set wz_Media off
```

Auch die Variable \$EVENT haben wir bereits verwendet. Sie repräsentiert den aktuellen Schaltzustand:

```
define Schalter1Notify notify Schalter1 set wz_Media $EVENT
```

Da fhem in Perl programmiert ist, gibt es auch die Möglichkeit, Perl-code aus fhem heraus auszuführen. Oder andersherum: zusätzlich zu den fhem-eigenen Befehlen lassen sich auch Perl-Routinen zur Steuerung verwenden. Um unterscheiden zu können, ob fhem-Befehle oder Perl-Befehle verwendet werden, werden Perl-Kommandos immer in geschweifte Klammern gesetzt.

Auch aus Perl-Code heraus lassen sich fhem-Befehle absetzen. Daher bewirken folgende zwei Zeilen dasselbe:

```
set lamp off  
{fhem(„set lamp off“)}
```

In der ersten Zeile ist der bereits bekannte fhem set-Befehl verwendet.

In der zweiten Zeile ist Perl-Code dargestellt – er ist in geschweifte Klammern eingebunden. In Perl gibt es den Befehl fhem, mit dem man aus Perl heraus fhem-Befehle ausführen kann. Als Parameter des Befehls fhem ist der auszuführende Befehls-String angegeben – also das, was Sie sonst in das fhem Kommandofeld eingeben würden.

Die Perl-Version des o.g. Codes können Sie nun zusammen mit einem notify oder at verwenden, oder Sie können das Perl-Kommando (incl. geschweiften Klammern!) ebenso in das fhem Kommandofeld eingeben. Durch die geschweiften Klammern weiß fhem, dass es sich um Perl-Code handelt.

```
define LampeAusUm9 at 09:00 {fhem(„set lamp off“)}
```

Der vermutlich am häufigsten verwendete Befehl ist der if-Befehl. Seine Syntax lautet:
{ if (Bedingung) {Befehl1} else {Befehl2} }

Das eingangs gezeigte Beispiel

```
define Schalter1Notify notify Schalter1:off set wz_Media off
```

kann man also ebenfalls schreiben als

```
define Schalter1Notify notify Schalter1 { if ( „$EVENT“ eq „off“ ) {fhem(„set wz_Media off“)} }
```

Wenn also die Variable \$EVENT den Wert off hat, wird in fhem der Befehl set wz_Media off ausgeführt.

Hinweis:

In Codebeispielen im Wiki oder alten Forumsbeiträgen finden Sie eventuell noch die Variable % statt \$EVENT.

Achtung:

Wenn Sie im Code Ihres notify anstatt der Variable % wirklich das Zeichen % verwenden möchten, schreiben Sie dies als %% bzw. analog \ oder @. Das sieht im Programmcode seltsam aus, zum Ausführungszeitpunkt wird aber in *einfache* Zeichen umgewandelt. Details und weitere Sonderfälle finden Sie in der commandref [hier](http://fhem.de/commandref.html#notify) (<http://fhem.de/commandref.html#notify>) im Abschnitt „Notes“.

Tipp:

Um ein **notify** oder **at** mit **Perl-Code** zu **erstellen**, gehen Sie in zwei Schritten vor:

1. Definieren Sie das notify/at in der Kommandozeile mit einem leeren Perl-code-Block, also z.B. `define Schalter1Notify notify Schalter1 {}`
2. Wechseln Sie nun über das Menü ‚Everything‘ in die Detail-Ansicht Ihres at/notify. Klicken Sie dort auf ‚DEF‘ um die Details Ihres at/notify zu bearbeiten. Fügen Sie erst dort den Perl-Code ein. Klicken Sie schließlich die Schaltfläche ‚modify <name>‘ zum Speichern Ihres Codes. Auf diese Weise ist die Bearbeitung am einfachsten und entspricht der, die bereits bei notify gezeigt wurde (siehe dort **Bearbeiten über das Webfrontend**)

Hinweis:

Die **Schreibweise** von Perl-code ist in der Detail-Ansicht ‚ganz normal‘, d.h. Sie können ganz normalen Perl-code verwenden. Nachdem Sie Ihre Änderungen gesichert haben und außerdem den Befehl `save` ausgeführt haben, schauen Sie sich das Ergebnis in `fhem.cfg` an. Sie werden feststellen, dass dort a) alle Semikola, Prozentzeichen und @ verdoppelt wurden und b) jeder Zeilenumbruch mit einem backslash (\) ‚geschützt‘ wurde. Ersparen Sie sich diese Schreibweise, indem Sie nicht direkt Ihre `fhem.cfg` bearbeiten, sondern die Detail-Ansicht Ihres at/notify zur Bearbeitung von Perl-code verwenden.

Tipp:

Verwenden Sie einen Editor, bei dem Schlüsselwörter und vor allem zusammengehörende Klammerpaare ersichtlich sind. Der häufigste (Anfänger-)Fehler sind vergessene oder in falscher Reihenfolge gesetzte schließende Klammern! Verwenden Sie z.B. das kostenlose Notepad++ . Erst nachdem Sie in Notepad++ alle Klammern richtig gesetzt haben, kopieren Sie Ihr Miniprogramm nach `fhem` und hängen es z.B. wie oben beschrieben an ein at oder notify.

Tipp:

Für Notepad++ gibt es die Möglichkeit, `fhem`-Befehle erkennen zu lassen. Damit lassen sich Tippfehler leichter finden. Installieren Sie dafür die zusätzliche Programmiersprache `fhem` und wählen Sie diese beim Bearbeiten der `fhem.cfg` oder Ihrer include-Datei aus.

Tipp:

Um Ihren Perl-code zu **testen**, verwenden Sie das `trigger`-Kommando. Da dieses

Kommando nur auf notify, jedoch nicht auf at wirkt, entwickeln und testen Sie Ihren Perl-code immer mit einem notify, auch wenn Sie ihn letztendlich einem at zuordnen möchten – das hinüberkopieren ist ja einfach. Wenn Sie also z.B. ein define Testcode notify Schalter1:on {Perl-code} testen möchten, können Sie Ihren Code mit ‚trigger Schalter1 on‘ testen. trigger simuliert das Eintreten eines Ereignisses (also das Eintreffen eines Funktelegramms von Schalter1 mit dem Befehl on).

Gerätewerte in Bedingungen – Value, ReadingsVal, ReadingsTimestamp

Häufig sollen Schaltvorgänge abhängig gemacht werden. Für das Gerät, das ein notify auslöst, steht wie oben dargestellt die Variable \$EVENT zur Verfügung.

Sollen jedoch Werte anderer Geräte mit einbezogen werden, kommen die Funktionen Value() und ReadingsVal zum Einsatz.

Das o.g. Beispiel kann man auch so schreiben:

```
define Schalter1Notify notify Schalter1 { if ( Value("Schalter1") eq "off" ) {fhem("set wz_Media off")}}
```

Die Variable \$EVENT ist hier ersetzt durch die Funktion Value(), es wird der Status des Geräts Schalter1 ausgelesen. Man hat nun die Möglichkeit, den Status eines beliebigen Geräts als Schaltbedingung einzubauen, es muss nicht dasselbe sein, das den notify auslöst. Also z.B.

```
define Schalter1Notify notify Schalter1 { if ( Value("Schalter1") eq "off" && Value("Schalter2") eq "off" ) {fhem("set wz_Media off")}}
```

Hier wird wz_Media nur dann ausgeschaltet, wenn sowohl Schalter1 als auch Schalter2 den Zustand off haben.

Wie im Abschnitt **fhem-Konfiguration: Das besondere Gerät „Global“** gezeigt, verfügen viele Geräte nicht nur über den STATE im oberen Bereich (der bei den meisten Geräten on oder off ist, z.B. bei Thermometern aber die aktuellen Messwerte zeigt), sondern auch über Detailwerte, die sogenannten Readings im unteren Bildschirmbereich der Detailansicht.

Fhem

[Floorplans](#)

[CUL WS](#)

[Esszimmer](#)

[Korridor](#)

[Kueche](#)

[Media](#)

[Plots](#)

[Schlafzimmer](#)

[Unsorted](#)

[Wohnung](#)

[Wohnzimmer](#)

[Everything](#)

[Howto](#)

[Wiki](#)

[Details](#)

[Edit files](#)

[Select style](#)

[Event monitor](#)

CODE	1
CUL_MSGCNT	86
CUL_RAWMSG	K0104617406
CUL_RSSI	-71
CUL_TIME	2012-03-12 11:46:33
DEF	1
LASTIODev	CUL
MSGCNT	86
NAME	ez_Aussensensor
NR	203
STATE	T: 10.4 H: 74.6
TYPE	CUL_WS
corr1	0
corr2	0
corr3	0
corr4	0

Readings

DEVFAMILY	WS300	2012-03-12 11:46:33
DEVTYPE	S300TH	2012-03-12 11:46:33
humidity	74.6	2012-03-12 11:46:33
state	T: 10.4 H: 74.6	2012-03-12 11:46:33
temperature	10.4	2012-03-12 11:46:33

attr ez_Aussensensor Audio

fm_order	15	deleteattr
model	S300	deleteattr
room	Wohnung,Esszimmer	deleteattr

[Select icon](#)
[Device specific help](#)

Die Funktion **Value()** liest STATE des Geräts aus, oben blau markiert.

Um ein Reading (oben rot markiert) auszuwerten, wird die Funktion **ReadingsVal()** verwendet. Hier muss man zusätzlich zum Devicenamen auch den Namen des gewünschten Readings hinterlegen. Außerdem wird ein Default-Wert angegeben, der verwendet wird, falls das Auslesen des Readings fehlschlagen sollte. Die Syntax ist `ReadingsVal(<device>, <reading>, <default-Wert>)`

Möchte man also einen Schaltvorgang z.B. von der gemessenen Temperatur des im screenshot dargestellten device abhängig machen, könnte das so aussehen:

```
define Heizungssteuerung at +*01:00:00 { if (ReadingsVal("ez_Aussensensor","temperature",99) < 20) { fhem("set heizung on") } else { fhem("set heizung off") } }
```

Es wird also jede Stunde geprüft, ob die Außentemperatur unter 20 Grad Celsius liegt und abhängig davon die Heizung ein- oder ausgeschaltet. (o.k., man kann den Temperatursensor ja auch nach innen hängen.

Wenn statt des Reading-Werts der Zeitstempel ausgelesen werden soll, verwenden Sie die Funktion `ReadingsTimestamp(<device>, <reading>, <default-Wert>)`

Spezielle Variablen und Operatoren

Wie bereits dargestellt gibt es in fhem ein paar spezielle Variablen (siehe auch [hier: http://fhem.de/commandref.html#notify](http://fhem.de/commandref.html#notify)):

fhem-Schreibweise	Erklärung
\$EVENT	Enthält den Wert des zuletzt eingetretenen Ereignisses, also z.B. on oder off oder toggle oder 20°C. Die „alte“ Schreibweise dafür ist das Prozentzeichen, die neue und empfohlene Schreibweise ist \$EVENT.
\$EVTPART0, \$EVTPART1 usw.	Wenn der vom Gerät übermittelte event aus mehreren Worten besteht (getrennt durch Leerzeichen), so enthält \$EVTPARTx die einzelnen „Worte“. Bei einem event-Wert „actuator: 86 %“ enthalten die Variablen also folgende Werte: \$EVENT: actuator: 86 % \$EVTPART0: actuator: \$EVTPART1: 86 \$EVTPART2: %
\$NAME	Enthält den Namen des Geräts, von dem das Ereignis ausgelöst wurde. Die „alte“ Schreibweise dafür ist das at-Zeichen @, die neue und empfohlene Schreibweise ist \$NAME.
\$we	Wochenende – siehe nächste Seite

Man kann also schreiben:

```
define test notify *.* { if ($NAME eq "Taster" && $EVENT eq "on") {fhem("set lampe $EVENT")}}
```

Die Programmiersprache perl hat die Besonderheit, dass Variablen nicht mit einem Typ definiert werden. Deklariert man also z.B. MeineVariable (my \$MeineVariable;), so wird dabei nicht angegeben, ob sie vom Typ Character, Integer, Date o.ä. ist. Das vereinfacht die Deklaration und führt zu weniger Fehlermeldungen bei Wertzuweisungen im weiteren Programmverlauf. Allerdings muss bei späteren Auswertungen klargestellt werden, ob der Wert der Variable als Character (String) oder als Zahl betrachtet werden soll. Ein Beispiel:

```
my $var1 = ,10'; my $var2 = ,2';
# Ergibt true, da die Zahl 10 größer ist als die Zahl 2:
if ($var1 > $var2)...
# Ergibt false, da die Zeichenfolge 10 kleiner ist als die Zeichenfolge 2:
if ($var1 gt $var2)...
```

Die Unterscheidung wird also nicht zum Zeitpunkt der Variablendeklaration durchgeführt, sondern durch die Auswahl des Operators (hier > oder gt). Daher merke:

<, >, <=, >=, ==, != sind numerische Operatoren
lt, gt, le, ge, eq, ne sind die entsprechenden Stringvergleichsoperatoren

Auch sei darauf hingewiesen, dass der Operator eq auf identischen Inhalt prüft, zB
my \$val="on"; if (\$val eq "on") ... #ist true

Eine Prüfung auf ein Muster funktioniert NICHT:
my \$val="fhem"; if (\$val eq "*he*") ... #ist false

Hier kann der perl Mustervergleich =~ m verwendet werden:
my \$val="fhem"; if (\$val =~ m"he") ... #ist true

Dieser Mustervergleich (matching) ist äußerst mächtig und firmiert auch unter dem Namen „Regular Expression“ oder kurz „regexp“. Mehr Lesestoff findet sich z.B. [hier](http://www.troubleshooters.com/codecorn/littperl/perlreg.htm) (<http://www.troubleshooters.com/codecorn/littperl/perlreg.htm>).

Hinterlegen eigener Programme – 99_myUtils.pm

Kleine eigene Automatisierungen können wie bereits beschrieben mittels notify als „Makro“ hinterlegt werden. Werden diese eigenen Automatisierungen jedoch umfangreicher, lohnt sich das Anlegen einer separaten Programmdatei, in der diese eigenen Programmchen hinterlegt werden. Wie das geht und was dabei zu beachten ist, steht in [diesem Wiki-Artikel](http://www.fhemwiki.de/wiki/99_myUtils_anlegen) (http://www.fhemwiki.de/wiki/99_myUtils_anlegen).

Auch werden bereits einige Helfer-Routinen bereits mit ausgeliefert. Sie sind gesammelt in 99_Utils.pm (**Achtung** – 99_Utils.pm wird ggf. per update überschrieben - ist nicht dasselbe wie 99_myUtils.pm – in 99_Utils.pm daher keine eigenen Routinen hinterlegen!). Eine Liste der in 99_Utils.pm enthaltenen Funktionen findet sich [hier](http://fhem.de/commandref.html#Utils) (<http://fhem.de/commandref.html#Utils>).

Nur am Wochenende? \$we

Wenn Ihre Nachttischlampe als Wecker dienen und jeden Tag um 08:00 eingeschaltet

werden soll, erreichen Sie das mit

```
define WeckenMitLicht at *08:00 set sz_Lampe on
```

Wenn Sie am Wochenende länger schlafen möchten und daher auf diesen Weckdienst verzichten wollen, können Sie das folgendermaßen erreichen:

```
define WeckenMitLicht at *08:00 { if (!$we) { fhem(„set sz_Lampe on“) } }
```

Die Variable `$we` wird von `fhem` automatisch gefüllt: Sie ist `true` an Samstagen und Sonntagen, und `false` an Wochentagen.

Mit einem `!` können Sie außerdem die Auswertung umkehren: `!$we` steht für „nicht am Wochenende“ und ist gleichbedeutend mit „nur an Wochentagen“.

Wenn Sie also am Wochenende zu einer späteren Zeit geweckt werden möchten:

```
define WeckenMitLichtWE at *09:30 { if ($we) {fhem(„set sz_Lampe on“)}}
```

In den Ferien? `holiday`, `holiday2we`

Das vorgenannte Beispiel macht Schaltvorgänge von Wochenenden abhängig. Möglicherweise möchten Sie ja an Ferientagen ebenso wie an Wochenenden erst später geweckt werden. Hierfür gibt es in `fhem` zum einen die Möglichkeit, Kalender zu definieren, zum anderen die Möglichkeit, die Ergebnisse des Kalenders ebenfalls für die Variable `$we` zu berücksichtigen.

Holiday

In einer separaten Datei werden alle Feiertage eines Jahres gelistet. Hier ein Beispiel des Datei-Inhalts von **Bayern.holiday** (mit einer kleinen Erweiterung am Ende):

```
# Format für einzelne Tage: 1 MM-DD <Text>
1 01-01 Neujahr
1 01-06 Heilige Drei Koenige
1 05-01 Tag der Arbeit
1 08-15 Mariae Himmelfahrt
1 10-03 Tag der deutschen Einheit
1 11-01 Allerheiligen
1 12-25 1. Weihnachtstag
1 12-26 2. Weihnachtstag

# Osterbezogene Feiertage
# Format: 2 <relative Tage von Ostern> <Text>
2 -2 Karfreitag
2 1 Ostermontag
2 39 Christi Himmelfahrt
2 50 Pfingsten
2 60 Fronleichnam

5 -1 Wed 11 23 Buss und Betttag
```

```
# Urlaube
# Format: 4 MM-DD MM-DD <Text>
4 12-30 12-31 Urlaub1
```

Details zu den einzelnen Termintypen finden Sie in der commandref [hier \(http://fhem.de/commandref.html#holiday\)](http://fhem.de/commandref.html#holiday).

Der Inhalt der .holiday-Datei kann im Webfrontend bearbeitet werden. Wählen Sie dazu den Menüpunkt ‚Edit Files‘ und klicken Sie auf den Namen der gewünschten Holiday-Datei.

Diese Datei wird mit
define <holidayname> holiday also z.B.
define Bayern holiday
in fhem eingebunden.

Über die Kommandozeile können Sie dann prüfen, ob an einem bestimmten Datum ein Feiertag ist:

```
get <holidayname> mm-dd
get Bayern 12-26
```

Das Ergebnis ist „none“ oder der in der Liste hinterlegte Name des Feiertags.

holiday2we

Wenn eine Ihrer Feiertagsdateien zur Ermittlung von Schaltvorgängen einbezogen werden soll, wählen Sie diese zur Überleitung in die Variable \$we aus:

```
attr global holiday2we Bayern
```

Es wird nun täglich um Mitternacht geprüft, ob der anbrechende Tag gemäß der angegebenen holiday-Datei ein Feiertag ist und die Variable \$we ggf. auf true gesetzt.

Verschachteltes at

Ein häufiger Anwendungsfall ist das Schalten zu einer bestimmten Uhrzeit mit von der Startzeit abhängigen weiteren Schaltvorgängen. Beispielsweise soll Ihr morgendlicher Weckvorgang um 7:30 starten mit dem langsamen Hochdimmen der Nachtschlampe.

Außerdem soll 15 Minuten später das Rollo hochgefahren und schließlich 30 Minuten nach wakeup-Zeit die Nachtschlampe wieder ausgeschaltet werden. Wichtig: Den Startzeitpunkt können Sie leicht mit modify wakeup *08:00 ändern – die Angabe z.B. für das Rollo bezieht sich auf den Start dieser Prozedur, wird also durch das modify ebenfalls verschoben.

Als Code sieht das so aus:

```
CFGFN
DEF
*07:30 {
  if (!$we) {
    fhem("set sz_Leselampe dim100% 1280");
    fhem("define wakeup2 at +00:15 set sz_Rollo off");
    fhem("define wakeup3 at +00:30 set sz_Leselampe off");
  }
}

```

modify wakeup

NAME	wakeup
------	--------

Beachten Sie die Verwendung der * bei den Zeitangaben: Der wakeup-Zyklus soll täglich ausgeführt werden, also Angabe mit *. Nach diesem täglichen Start sollen aber die weiteren Steuerungen nur jeweils einmalig ausgeführt werden, daher ist hier bei den Zeitangaben **kein** * gesetzt.

Ein detailliert erklärtes Beispiel dazu finden Sie im [fhem-wiki \(http://fhemwiki.de/wiki/Wakeuplight\)](http://fhemwiki.de/wiki/Wakeuplight).

Heizungssteuerung

Die Steuerung von Heizungen kann zunächst in drei Bereiche gegliedert werden:

1. Messen der Ist-Temperatur
2. Berechnen der erforderlichen Heizleistung, um von der gemessenen Ist-Temperatur zur gewünschten Zieltemperatur zu gelangen. Dabei soll die angeforderte Heizleistung umso größer sein, je größer der Unterschied von Ist- und Solltemperatur ist, damit die Solltemperatur möglichst schnell erreicht wird.
3. Regeln der Heizleistung

Messen der Ist-Temperatur

Die Messung erfolgt durch ein Thermometer.

Berechnen der erforderlichen Heizleistung

Die Berechnung wird von „PID-Reglern“ (Proportional-Integral-Differential) genannten Algorithmen übernommen. PID ist also eine Software, die aus den Eingabeparametern Ist- und Solltemperatur als Output die erforderliche Heizleistung berechnet. Diese wird in % angegeben, sie kann also z.B. 20% oder 90% der maximal verfügbaren Heizleistung betragen.

Thermostate

Alle zur Heizungssteuerung relevanten Hardwaresysteme bieten als separate Komponente ein „Thermostat“, das die Einstellung einer Solltemperatur ermöglicht, über ein eingebautes Thermometer verfügt und auch einen PID-Rechner enthält, der die erforderliche Heizleistung berechnet. Meist kann dabei die Solltemperatur direkt manuell eingestellt werden oder auch als Tages- und Wochenprogramm hinterlegt werden, so dass z.B. nachts die Solltemperatur niedriger eingestellt wird als tagsüber. In festen Zeitzyklen (z.B. alle fünf Minuten) wird dann die gemäß Solltemperatur benötigte Heizleistung berechnet.

Viele Thermostate bieten optional auch die Möglichkeit, einen Fensterkontakt anzukoppeln. Meldet der Fensterkontakt „Fenster offen“, wird auf eine einstellbare niedrigere Solltemperatur (z.B. 17°C) gewechselt und die Heizleistung entsprechend reduziert. Wird das Fenster geschlossen, kehrt der Thermostat zur zuletzt eingestellten Solltemperatur zurück und passt darauf basierend die Heizleistung wieder an. So wird ein „für draußen heizen“ vermieden.

Regeln der Heizleistung

Das Regeln der Heizleistung erfolgt meist über sogenannte „Ventilstelltriebe“. Diese ersetzen die bisherigen (Thermostat-) Ventile an den Heizkörpern und sind per Funk regelbar. Der Austausch der Ventile ist dabei unproblematisch, da das eigentliche Ventil im Heizkörper verbleibt – es sind also keine Installateurs-Arbeiten erforderlich.

Auch gibt es Ventilstelltriebe, die den Thermostat selbst beinhalten. Der Unterschied zum (raum-) zentralen Thermostat ist lediglich der Standort der Messung der Ist-Temperatur.

Falls Sie selbst der Betreiber der Heizungsanlage sind (Etagenheizung oder eigene Heizanlage im Haus) besteht außerdem die Möglichkeit, die Heizungsanlage selbst zu steuern. Dazu bieten viele Heizanlagen die Möglichkeit, in die Brennersteuerung einzugreifen – insbesondere den Brenner „abzuwürgen“, auch wenn er laut heizsystem-eigener Steuerung weiter heizen würde,

Heizungssteuerung mit fhem

Hinweis von pi3g:

Auch ZWave-Thermostate lassen sich in das System einbinden, und verfügen zumeist über ein eingebautes Thermometer.

Wie beschrieben sind die verfügbaren Hardwaresysteme zur Heizungssteuerung in sich autarke Regelkreisläufe. Mit fhem kann man zur weiteren Steigerung der Energie-Effizienz und zur Erhöhung des Komforts in diese Regelungen eingreifen.

Häufigster Anwendungsfall ist dabei das Anpassen der Solltemperatur. Hierbei kann man das Wochenprogramm, das meist in den Thermostaten verfügbar ist, intelligent erweitern oder ersetzen.

So gibt es z.B. das spezielle fhem-Modul [Heating_Control \(http://fhem.de/commandref.html#Heating_Control\)](http://fhem.de/commandref.html#Heating_Control) zum Ersetzen bzw. sinnvollen Erweitern der thermostat-internen Wochenprogramme.

Hier einige Anwendungsbeispiele:

- Setzen der Solltemperatur abhängig davon, ob wirklich jemand zuhause ist
- Setzen der Tag/Nachttemperatur nicht nach festen Uhrzeiten, sondern Absenkung wenn die Bewohner wirklich schlafen gehen und Rückkehr zur Wohlfühltemperatur 20 Minuten vor der in fhem eingestellten Weckzeit
- Zeitpunkt der Umschaltung von Nachtabsenkung auf Tagestemperatur von der Nacht-Tiefsttemperatur abhängig machen – je kälter die Tiefsttemperatur laut yahoo-weather, desto früher soll morgens die Heizung wieder anspringen
- Steuern der Solltemperatur abhängig von Einträgen in einem Google-Kalender
- Manuelles Steuern der Solltemperatur via VPN-Verbindung von unterwegs

Auch kann man über die reine Beeinflussung der Solltemperatur hinausgehen. So kann man z.B. den Heizungsbrenner abschalten, wenn alle Thermostate als erforderliche Heizleistung 0% melden.

Hinweis:

Zur direkten Beeinflussung des Heizungsbrenners ist es in der Regel erforderlich, einen Funkschalter an die Heizungsanlage anzuschließen. Dies nicht von einem Elektriker oder Heizungsfachmann durchführen zu lassen, gefährdet nicht nur die Heizanlage, sondern auch Sie – es wäre „Sparen am falschen Ende“ !

In meinem eigenen Haushalt betreibe ich einen Funk-Raumthermostat in nur einem Raum, dessen Solltemperatur ich abhängig vom HomeStatus setze. Auf Ventilstelltriebe verzichte ich vollständig. Meldet der Thermostat eine erforderliche Heizleistung

von <50%, wird der Heizbrenner meiner Etagenheizung durch fhem abgeschaltet. Meldet der Thermostat eine erforderliche Heizleistung von >50%, wird der Brenner freigegeben und die Heizungsanlage arbeitet auf die dort fest eingestellte Solltemperatur (=Wohlfühltemperatur 21 °C) hin.

Da fhem zwei PID-Module bietet ([PID: http://www.fhem.de/commandref.html#PID](http://www.fhem.de/commandref.html#PID)) und die erweiterte Version [PID20: http://www.fhem.de/commandref.html#PID20](http://www.fhem.de/commandref.html#PID20)), ist es außerdem denkbar, reguläre Funkthermometer zu verwenden, die erforderliche Heizleistung durch den PID-Regler in fhem zu berechnen und gekoppelte Funk-Ventilstelltriebe direkt zu steuern und/oder den Heizbrenner direkt zu steuern. Dabei können die Komponenten durchaus aus unterschiedlichen Hardwaresystemen stammen, also z.B. Temperaturmessung mit 1-wire und Heizkörperreglung mit FHT, HomeMatic oder MAX.

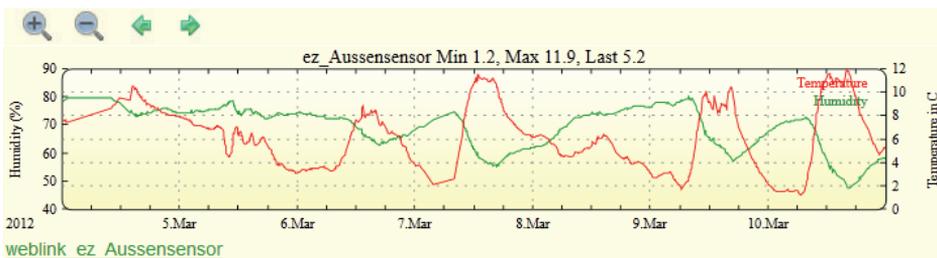
Für welchen Weg Sie sich entscheiden, hängt von den Gegebenheiten ihres Heims und Ihrer Heizungsanlage ab.

Die an fhem anbindbaren Hardwaresysteme sind:

Hardwaresystem	Raumthermostat	Ventilstelltrieb	Thermostat + Stelltrieb in einem
FHT	FHT80b	FHT8v	-/-
HomeMatic	HM-CC-TC (alt)	HM-CC-VD (alt)	HM-CC-RT-DN (neu)
MAX!	-/-	-/-	Ja
1-wire und andere	Nur Thermometer	-/-	-/-

Temperaturverläufe als Diagramm – Logs & Plots

Die autcreate-Funktion erzeugt beim Erstellen bestimmter Messgeräte wie z.B. FHT-80b (Heizungssteuerung) oder S300TH (Thermostat und Hygrometer) automatisch auch ein Verlaufsdiagramm. Hier als Beispiel die Wochendarstellung für ein S300TH, das als Außenthermometer dient. Die Temperatur wird mit einer roten, die Luftfeuchtigkeit mit einer grünen Linie dargestellt:



Mit den Schaltflächen am oberen Rand kann die Zoomstufe (Tag, Woche, Monat) verändert werden, auch kann man mit den grünen Pfeiltasten durch die Historie ‚wandern‘.

Die dargestellten Werte werden aus der zugeordneten Logdatei gelesen. In der commandref finden Sie die relevanten Infos unter [FileLog \(http://fhem.de/commandref.html#FileLog\)](http://fhem.de/commandref.html#FileLog) und [weblink fileplot \(http://fhem.de/commandref.html#weblink\)](http://fhem.de/commandref.html#weblink).

Details dazu folgen in der erweiterten Version dieses Dokuments.

Jemand zuhause? Anwesenheitserkennung und HomeStatus

Ein zentraler Parameter der Haussteuerung ist unterschiedliches Verhalten abhängig davon, ob derzeit jemand zuhause ist oder nicht (Solltemperatur der Heizung, Scharfschalten der Alarmanlage, Musik an/aus etc.) bzw. beim Übergang von einem Zustand zum Anderen.

Dazu besteht die Möglichkeit, einen „Status“ des Hauses zu definieren (ich nenne ihn hier einmal „HomeStatus“) der die Werte *Zuhause* oder *Nicht zuhause* annehmen kann oder – als differenziertere Variante – *Zuhause*, *Schlafen*, *Kurz weg* und *Lange weg*.

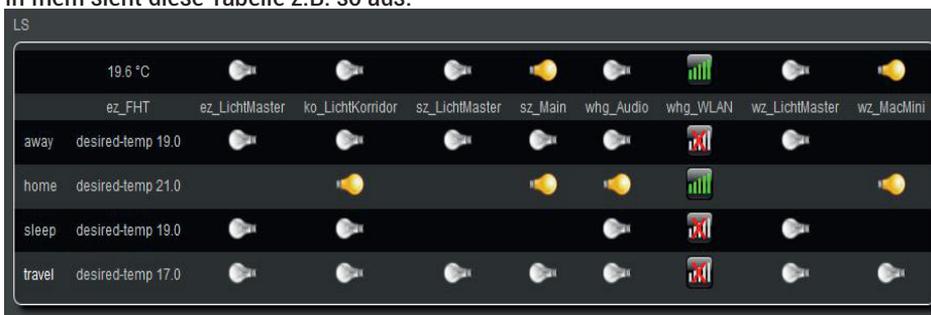
Beim Wechsel in einen neuen Status können dabei unterschiedliche Aktionen ausgelöst werden:

Wechsel zu Status	Beleuchtung	Heizung	Musik
Zuhause	Korridorbeleuchtung an für 5 Minuten	Solltemperatur 21°C	Starten

Schlafen	Alle Lichter aus außer im Schlafzimmer	Solltemperatur 18°C	Stoppen
Kurz weg	Alle aus	Solltemperatur 19°C	Stoppen
Lange weg	Alle aus	Solltemperatur 17°C	Server herunterfahren

Auch kann der HomeStatus zur Steuerung weiterer Geräte verwendet werden. So soll z.B. morgens zur Weckzeit das Wakeuplight nur dann aktiviert werden, wenn auch jemand zuhause ist (HomeStatus *Zuhause* oder *Schlafen*) oder nachts durch den Bewegungsmelder nicht das grelle Korridorlicht, sondern nur eine schwache Beleuchtung eingeschaltet werden (HomeStatus *Schlafen*), während die Alarmanlage nur dann aktiv ist, wenn niemand zuhause ist (HomeStatus *Kurz weg* oder *Lange weg*).

In fhem sieht diese Tabelle z.B. so aus:



(In der ersten Zeile sieht man den aktuelle Schaltzustand der Geräte, im unteren Block den Zielzustand je Status, hier home, sleep, away, travel)

Ein Beispiel mit dem zugehörigen code findet sich im [fhem-wiki \(http://fhemwiki.de/wiki/Zuhause-Status\)](http://fhemwiki.de/wiki/Zuhause-Status).

Der Komfort kann weiter erhöht werden, indem das Setzen des Status automatisch erfolgt. Zur Anwesenheitserkennung gibt es unterschiedliche Mechanismen:

- Prüfung, ob bestimmte Geräte/Handys im WLAN eingeloggt sind (fhem-Modul [PRESENCE: http://fhem.de/commandref.html#PRESENCE](http://fhem.de/commandref.html#PRESENCE)) – problematisch ist hierbei, dass häufig ein zeitlicher Verzug zwischen realer Ankunft/Abreise und dem Melden der An/Abmeldung des Geräts auftritt. Auch melden sich viele mobile Geräte bei Nicht-Nutzung aus dem WLAN ab (Stromsparmmodus).
- Verwendung von Bewegungsmeldern – funktioniert gut für den Übergang von „niemand da“ -> „bewohnt“, jedoch ist schwierig herauszubekommen, ob alle Bewohner das Haus verlassen haben oder nur gebannt vor dem Fernseher sitzen.

- Verwendung von Mikrotastern, um zu erkennen, ob die Haus-/Wohnungstür verschlossen ist – nur nutzbar falls nachts nicht abgeschlossen wird, erkennt auch nicht, ob niemand zuhause ist oder alle schlafen.
- Erkennung über Bluetooth – dies verbraucht bei mobilen Geräten jedoch noch zu viel Strom, um dauerhaft genutzt werden zu können.
- Erkennung des Nutzer-Aufenthaltsorts, z.B. über Geofancy (zu dem auch ein gleichnamiges fhem-Modul existiert) – auch hierzu erforderliche GPS-Empfänger verbrauchen in mobilen Geräten zu viel Strom, um dauerhaft eingeschaltet bleiben zu können.
- Manuelles Setzen des HomeStatus über einen Wandschalter neben der Eingangstür – funktioniert zuverlässig, sofern nicht vergessen wird, den Schalter beim Verlassen oder Betreten der Wohnung zu betätigen.

Der „Königsweg“ für dieses Thema ist also leider noch nicht gefunden. Welcher Weg für Ihre Umgebung am besten passt, müssen Sie selbst entscheiden.

In meinem eigenen Haushalt habe ich mich für das manuelle Setzen des HomeStatus durch einen Schalter neben der Eingangstür und eine Fernbedienung neben dem Bett entschieden.

Anwesenheitssimulation

Um während des eigenen Urlaubs potentiellen Einbrechern vorzugaukeln, dass jemand zuhause sei, lassen sich Geräte in definierbaren Zeiträumen zufällig ein- und ausschalten. Beispiele und die Einrichtung sind beschrieben im Modul [RandomTimer](http://fhem.de/commandref.html#RandomTimer) (<http://fhem.de/commandref.html#RandomTimer>).

Schaltungen abhängig von Telefonanrufen – Callmonitor

Wenn Sie eine FritzBox in Ihrem Heimnetz betreiben, können Sie eingehende Anrufe erkennen und daraufhin z.B. die Lautstärke Ihrer Audiogeräte reduzieren sowie nach dem Beenden des Gesprächs die Lautstärke wieder auf den vorherigen Stand zurücksetzen. Auch ist es möglich, eine bestimmte Rufnummer zur Heizungssteuerung zu reservieren und z.B. die Solltemperatur der Heizung zu erhöhen, wenn auf dieser Rufnummer ein Anruf von definierten Herkunfts-Telefonnummern erfolgt.

Dabei muss fhem nicht auf der FritzBox laufen, auch das Koppeln einer entfernten

FritzBox von Ihrem fhem-Server aus ist möglich. Details siehe [FB_Callmonitor \(http://fhem.de/commandref.html#FB_CALLMONITOR\)](http://fhem.de/commandref.html#FB_CALLMONITOR).

Multimedia-Geräte

Eine zunehmende Zahl von Multimedia-Geräten kann über LAN gesteuert werden. Ein einfaches Anwendungsbeispiel ist das Regeln der Lautstärke bei eingehendem Anruf. Die unten angegebenen Geräte können aus fhem heraus gesteuert werden. Die Anzahl verfügbarer Module wächst ständig, hier eine kleine Auswahl (Stand 01/2014):



Gerätetyp	Hersteller / Gerät	fhem-Modul
Audio	Logitech Squeezebox	SB_PLAYER
	MPD	MPD
	Mplayer (zur mp3- oder Audiostream-Wiedergabe)	StreamRadio
	Sonos	SONOS, SONOSPLAYER
	Yamaha Blueraey Player	YAMAHA_BD
TV / Video	Sony TV	Viera
	Samsung TV	STV
	LG TV	LGTV
	Philips TV	PhilipsTV
	AppleTV	iTunes
AVR (Audio/ Video-Receiver)	Yamaha	YAMAHA_AVR
	Onkyo	ONKYO_AVR
	Denon	DENON_AVR
Satellitenreceiver / Set-top-boxen	Dreambox, VUplus	ENIGMA2
Mediacenter	XBMC	XBMC
	MediaPortal	MediaPortal
	iTunes	iTunes
Radiowecker	Geräte mit Firmware von www.listenlive.nl	Listenlive

Details zu den genannten fhem-Modulen finden sich in der [commandref \(http://fhem.de/commandref.html\)](http://fhem.de/commandref.html).

Außerdem gibt es das Modul [remotecontrol \(http://fhem.de/commandref.html#remotecontrol\)](http://fhem.de/commandref.html#remotecontrol), das eine grafische Fernbedienung (wie oben links) anzeigt, die an jedes der o.g. Module angepasst und gekoppelt werden kann. So können Sie Ihr Tablet als Fernbedienung für Ihre an fhem gekoppelten Geräte verwenden.

Infrarot-Fernbedienung

Einige Geräte sind nur über Infrarot ansprechbar. Zur Einbindung in fhem gibt es zwei FS20 Infrarot-Geräte: FS20-iru (IR-Empfänger) und FS20-IRF (IR-Sender) sowie einen [iTach \(http://fhem.de/commandref.html#Itach_Relay\)](http://fhem.de/commandref.html#Itach_Relay).

Prüfung und Aufwecken von Servern im Heimnetzwerk: WOL

Zur Statusprüfung und dem Senden von WOL-packages dient das Modul [WOL \(http://fhem.de/commandref.html#WOL\)](http://fhem.de/commandref.html#WOL).

Webservices, http-Verbindungen

Wetterbericht einbinden

Hierfür gibt es die fhem-Module [weather \(http://fhem.de/commandref.html#Weather\)](http://fhem.de/commandref.html#Weather) basierend auf yahoo-Weather, sowie das Modul [GDS \(http://fhem.de/commandref.html#GDS\)](http://fhem.de/commandref.html#GDS) zur Einbindung von Daten des deutschen Wetterdienstes. Beide aktualisieren die Informationen in einstellbaren Intervallen (z.B. alle 15 Minuten) und stellen die gelesenen Daten in fhem als Readings zur Verfügung. In das Webfrontend können so Wetterberichtsdaten eingeblendet werden.

Die Vorhersagedaten können beispielsweise zur vorausschauenden Heizungssteuerung verwendet werden, um z.B. morgens die Heizung früher anspringen zu lassen, wenn es nachts besonders kalt war, oder um die Markise einzufahren und Rollläden zu schließen wenn starke Winde oder Hagel vorhergesagt sind.

condition	Schauer
current_date_time	17 Jan 2014 3:20 pm CET
day_of_week	Fr
fc1_code	11
fc1_condition	Schauer
fc1_day_of_week	Fr
fc1_high_c	6
fc1_icon	chance_of_rain
fc1_low_c	-1
fc2_code	26
fc2_condition	wolkig
fc2_day_of_week	Sa
fc2_high_c	8
fc2_icon	cloudy
fc2_low_c	3
fc3_code	26
fc3_condition	wolkig
fc3_day_of_week	So
fc3_high_c	10
fc3_icon	cloudy
fc3_low_c	1
fc4_code	11
fc4_condition	Schauer
fc4_day_of_week	Mo
fc4_high_c	6
fc4_icon	chance_of_rain
fc4_low_c	2
fc5_code	5
fc5_condition	Regen und Schnee
fc5_day_of_week	Di
fc5_high_c	4
fc5_icon	rainsnow
fc5_low_c	0
humidity	93
icon	chance_of_rain
pressure	982
pressure_trend	2
pressure_trend_sym	-
pressure_trend_txt	fallend
state	T: 4 H: 93 W: 3
temp_c	4
temp_f	39
temperature	4
visibility	8
wind	3
wind_chill	4
wind_condition	Wind: S 3 km/h
wind_direction	180
wind_speed	3

Schauer
4°C 93%
Wind: N 5 km/h

Fr: Schauer
min -1°C max 6°C

Sa: wolkig
min 3°C max 8°C

So: wolkig
min 1°C max 10°C

Mo: überwiegend wolkig
min 2°C max 6°C

Di: Regen und Schnee
min 0°C max 4°C

Zur Einbindung in das them-Webfrontend oder einen floorplan stehen besonders schöne Icons zur Verfügung.

Das Weather-Modul stellt viele Readings zur Verfügung, die für weitere Steuerungen verwendet werden können.

Google-Kalender, Mails und Messaging

Es können Einträge aus einem Google-Kalender verwendet werden, um Schaltvorgänge in fhem auszulösen. Details und umfangreiche Beispiele stehen [hier \(http://fhem.de/commandref.html#Calendar\)](http://fhem.de/commandref.html#Calendar) im commandref-Eintrag sowie [hier \(http://www.fhemwiki.de/wiki/Google-Kalender_zur_Steuerung_von_Dummies\)](http://www.fhemwiki.de/wiki/Google-Kalender_zur_Steuerung_von_Dummies).

Auch eingehende Mails können als Schalt-Trigger in fhem eingebunden werden. Das zugehörige fhem-Modul heißt [Mailcheck \(http://fhem.de/commandref.html#mailcheck\)](http://fhem.de/commandref.html#mailcheck).

Ebenso können Mails mit Statusinformationen versendet werden, z.B. beim Auftreten eines Alarms oder bei Eintreffen eines Anrufs. Das zu verwendende Modul ist abhängig vom Betriebssystem des Rechners, auf dem fhem betrieben wird. Ein generisches Modul ist [MSGMail \(http://fhem.de/commandref.html#MSGMail\)](http://fhem.de/commandref.html#MSGMail). Weitere Infos finden sich im [fhem-Wiki \(http://www.fhemwiki.de/wiki/Hauptseite\)](http://www.fhemwiki.de/wiki/Hauptseite) und im [fhem-Forum \(https://forum.fhem.de/\)](https://forum.fhem.de/).

Außerdem ist das Versenden von Push-Nachrichten an Smartphones möglich, siehe [Pushover \(http://fhem.de/commandref.html#Pushover\)](http://fhem.de/commandref.html#Pushover).

Kommunikation über HTTP

Immer mehr Geräte bieten eigene Web-Seiten, über die der Status ausgelesen werden kann bzw. eine Steuerung der Geräte möglich ist.

Um den Status eines Geräts auszulesen, muss häufig eine entsprechende URL aufgerufen werden, die dann eine html-Seite anzeigt, in der die gewünschten Statusinformationen enthalten sind.

Verwenden Sie das Modul HTTPMOD, das Sie durch Angabe der URL konfigurieren und dabei auch angeben, welche Information der zurückgegebenen Webseite in welches Reading extrahiert werden soll. Änderungen der Readingswerte lösen dabei fhem-Ereignisse aus, die dann mittels notify zur Steuerung beliebiger Aktionen verwendet werden können.

Um auf Basis beliebiger Ereignisse ein Gerät über dessen Webseite (also Angabe der URL erweitert um die entsprechenden CGI-Parameter) zu steuern, verwenden Sie die mit fhem ausgelieferte Funktion `GetFileFromURL()` (oder eine andere Funktion aus dem Funktionspool `HttpUtils.pm`) oder wiederum das o.g. Modul HTTPMOD.

Ist das parsing des Moduls HTTPMOD nicht hinreichend, schreiben Sie eine eigene Routine in myUtils, in der Sie mittels `GetFileFromURL()` die Geräte-Webpage „abholen“ und anschließend mittels `regexps` die gewünschten Werte extrahieren.

fhem neu starten – shutdown restart

Um –vor allem nach `update`- das Programm neu zu starten, führen Sie den Befehl `shutdown restart` aus.

Möchten Sie lediglich die Konfigurationsdatei `fhem.cfg` erneut einlesen, ohne `fhem` neu zu starten, nutzen Sie den Befehl `rereadcfg`.

fhem Programmaktualisierungen – update

Damit Sie Ihre `fhem`-Installation immer auf dem neuesten Stand halten, sichern Sie Ihren Stand regelmäßig in Backups und führen anschließend den Befehl `updatecheck` aus, so können Sie sich die neuesten bugfixes und Weiterentwicklungen anzeigen lassen. Werden diese Weiterentwicklungen von Ihnen benötigt, führen Sie nach einer Datensicherung den Befehl `update` aus zum herunterladen der neuesten Programmversionen auf Ihr Gerät.

Infoquellen

Mit den bisher dargestellten Möglichkeiten sollten Sie alle Grundlagen haben, um Ihre eigene Anwendung in `fhem` abzubilden.

Weitere hilfreiche Informationsquellen sind

- Die `fhem`-Doku unter <http://www.fhem.de>
- Die Befehlsreferenz unter <http://fhem.de/commandref.html>
- Das `fhem`-wiki unter <http://www.fhemwiki.de/wiki/FHEM>
- Das sehr aktive `fhem`-Forum unter <http://forum.fhem.de/>

Kurz betrachtet: Weitere Geräte und Funktionen

Die folgenden Kurzbeschreibungen sollen einen Ausblick auf Inhalte einer erweiterten Version von „Heimautomatisierung mit fhem“ bieten. Für ein Selbststudium sind z.T. bereits Links auf commandref und Wiki angegeben.

Erweitert: Logs & Plots

Erstellen Sie eigene Logdateien, die Daten unterschiedlicher Geräte zusammenfassen und damit die Ausgabe kombinierter Diagramme ermöglichen. Dieses Themengebiet ist sehr flexibel – und daher leider auch etwas kompliziert. In der commandref finden Sie die relevanten Infos unter [FileLog \(http://fhem.de/commandref.html#FileLog\)](http://fhem.de/commandref.html#FileLog) und [weblink fileplot \(http://fhem.de/commandref.html#weblink\)](http://fhem.de/commandref.html#weblink) sowie eine detaillierte Beschreibung unter [Creating plots \(http://fhemwiki.de/wiki/Creating_Plots\)](http://fhemwiki.de/wiki/Creating_Plots).

Dummy

In fhem können dummy-devices angelegt werden, die wie eine Variable genutzt werden können. So wird z.B. zur Abbildung des HomeStatus ein dummy benutzt: Der Wert des HomeStatus soll gespeichert werden, eine Funkkommunikation wie z.B. bei einem FS20-Gerät gibt es aber nicht. Diesem dummy können dann Attribute zugeordnet werden, die z.B. nur das Setzen bestimmter Werte zulassen, siehe **Nur bestimmte Befehle für ein Gerät - webCmd**.

Infos in der commandref finden Sie [hier \(http://fhem.de/commandref.html#dummy\)](http://fhem.de/commandref.html#dummy).

Watchdog

Mit dieser Funktion können Vorfälle überprüft und ggf. Aktionen eingeleitet werden, wenn z.B. ein Fenster länger als einen bestimmten Zeitraum offen steht oder sich ein Gerät seit über einen definierten Zeitraum nicht meldet. Watchdog ist also immer dann relevant, wenn nach Eintreten eines bestimmten Ereignisses nach einer bestimmten Zeitspanne geprüft werden soll, ob dasselbe oder ein anderes event (nicht) erneut eingetreten ist. Hier finden Sie die [commandref-Infos \(http://fhem.de/commandref.html#watchdog\)](http://fhem.de/commandref.html#watchdog).

Grenzwertgesteuertes Schalten – threshold und dewpoint

Als Auslöser zum Schalten können auch besondere Berechnungen vorgenommen oder Grenzwerte vorgegeben werden. [Dewpoint \(http://fhem.de/commandref.html#dewpoint\)](http://fhem.de/commandref.html#dewpoint) berechnet den Taupunkt und ermöglicht damit z.B. das zielgerichtete Schalten zur Entfeuchtung von Badezimmern oder Kellern.

Das Modul [THRESHOLD \(http://fhem.de/commandref.html#THRESHOLD\)](http://fhem.de/commandref.html#THRESHOLD) gestattet die Definition von Schwellwerten, bei deren Über- oder Unterschreiten eine Aktion ausgelöst werden kann.

shell-commands

Aus fhem heraus können nicht nur Perl-Programme, sondern auch Betriebssystem-Programme gestartet werden. Siehe hierzu den Aufruf über `system()` oder ``backticks`` - mehr Details stehen in der [commandref hier \(http://fhem.de/commandref.html#perl\)](http://fhem.de/commandref.html#perl).

Zusammenfassungen – readingsGroup

Mit dem Modul `readingsGroup` lässt sich ein Pseudo-Device anlegen, das dazu dient, `readings` unterschiedlicher Geräte in einer gemeinsamen Liste anzuzeigen. Diese Liste kann dann in einem beliebigen Raum oder in einem floorplan angezeigt werden.

Beispiele sind die Anzeige des `readings battery` von allen devices in Ihrer Konfiguration, oder das Anzeigen aller Temperature-Messwerte. Es kann auch eine beliebige Liste unterschiedlicher `readings` von unterschiedlichen Geräten zusammengestellt werden. Details stehen in der [Modulbeschreibung \(http://fhem.de/commandref.html#readingsGroup\)](http://fhem.de/commandref.html#readingsGroup), viele Beispiele finden Sie im entsprechenden [Wiki-Artikel \(http://www.fhemwiki.de/wiki/ReadingsGroup\)](http://www.fhemwiki.de/wiki/ReadingsGroup).

Schalten an bestimmten Wochentagen

Das Schalten von Geräten nach einem Zeitplan, der nicht jeden Tag gleich ist, lässt sich mit dem Modul [WeekdayTimer \(http://fhem.de/commandref.html#WeekdayTimer\)](http://fhem.de/commandref.html#WeekdayTimer) leicht einrichten.

Kleine Programmierbeispiele

Wakeup-Light

Ihre Nachtschlampe soll als Wakeuplight funktionieren, also morgens durch langsames ‚hochdimmen‘ den Sonnenaufgang simulieren, um das Aufwachen erträglicher zu machen.

Details in einem [Wiki-Artikel \(http://fhemwiki.de/wiki/Wakeuplight\)](http://fhemwiki.de/wiki/Wakeuplight).

Sunset und Sunrise

Steuerzeitpunkte können vom Sonnenauf- und -untergang abhängig gemacht werden. Infos stehen in der [commandref \(http://fhem.de/commandref.html#SUNRISE_EL\)](http://fhem.de/commandref.html#SUNRISE_EL). Auch eine dämmerungsabhängige Steuerung ist möglich auf Basis von [Twilight \(http://fhem.de/commandref.html#Twilight\)](http://fhem.de/commandref.html#Twilight).

Tag und Nacht - isday

Wenn Sie einen Bewegungsmelder haben, der aber für ein Nachtlicht nur nachts auslösen soll, verwenden Sie die Funktion `isday()`:

```
define Nachtlicht notify Bewegungsmelder {if (!isday() ) {fhem(„set lampe on“)}}
```

Da `isday()` von `SUNRISE_EL` gesetzt wird, muss dieses Modul aktiv sein.

Erstellen eigener fhem-Programme

Wenn Ihre eigenen Routinen an `notify` oder `at` zu lang und unübersichtlich werden, lohnt sich das Auslagern in eigene Programmdateien. Erste Schritte dazu finden Sie [hier \(http://www.fhemwiki.de/wiki/99_myUtils_anlegen\)](http://www.fhemwiki.de/wiki/99_myUtils_anlegen).

Beachten Sie die Übernahme von Code-Änderungen mit `reload`.

Beachten Sie die `fhem Perl-Specials` aus der `commandref` (siehe [hier: http://fhem.de/commandref.html#perl](http://fhem.de/commandref.html#perl)).

Entwicklung eigener fhem-Module

Der `fhem-core` ist bewusst flexibel gestaltet, um Entwicklern das Anbinden eigener Module zu erleichtern. Einen Einstieg zur Entwicklung eigener Module mit einer Erklärung der Interface-Datenstrukturen findet sich im Wiki [hier \(http://www.fhemwiki.de/wiki/DevelopmentModuleIntro\)](http://www.fhemwiki.de/wiki/DevelopmentModuleIntro).

Neue Module können auch in das [fhem-repository in sourceforge \(https://sourceforge.net/p/fhem/code/HEAD/tree/trunk/fhem/\)](https://sourceforge.net/p/fhem/code/HEAD/tree/trunk/fhem/) aufgenommen werden.

Dabei gibt es zwei Varianten:

- Einchecken in den Zweig „contrib“. Hier eingetragene Module werden nicht in die `fhem`-Distribution aufgenommen, werden nicht über „update“ verteilt und erscheinen auch nicht in der `commandref`. Möchte ein User dieses Modul verwenden, muss es manuell aus dem SVN heruntergeladen werden.
- Einchecken in den Zweig `fhem/FHEM`. Module, die hier eingetragene werden, werden automatisch zum Bestandteil der nächsten `fhem`-Version. Folgende Kriterien sind zu erfüllen:
 - Das Modul muss eine Dokumentation enthalten, die dann automatisch

in die `fhem` commandref eingebunden wird. Als Beispiel kann ein beliebiges anderes `fhem`-Modul dienen.

- Der Modulautor muss bereit sein, das Modul über das Forum mindestens drei Monate lang zu betreuen, also auf user-Fragen zu reagieren und ggf. bugfixes zu implementieren.

Für beide Wege muss sich der Modulautor einen logon zu Sourceforge besorgen und dann eine Mail an Rudolf König senden, damit der user Schreibrechte für den `fhem`-Branch erhält.

Anhang: Ein Einblick in die Konfigurationsdatei `fhem.cfg`

Hier eine kurze Erläuterung ‚am lebenden Objekt‘:

<pre>attr global autoload_undefined_devices 1 attr global holiday2we Bayern attr global logfile /var/InternerSpeicher/fhem/log/fhem-%Y-%m.log attr global modpath /var/InternerSpeicher/fhem attr global port 7072 global attr global statefile /var/InternerSpeicher/fhem/log/fhem.save attr global verbose 3 define CUL CUL /dev/ttyACM0@38400 2332 define WEB FHEMWEB 8083 global define WEBS FHEMWEB 8084 global attr WEBS smallscreen 1 define WEBP FHEMWEB 8085 global attr WEBP touchpad 1 define autocreate autocreate attr autocreate autosave 1 attr autocreate filelog /var/InternerSpeicher/fhem/log/%NAME-%Y.log attr autocreate weblink 1 attr autocreate weblink_room Plots define Bayern holiday define HomeStatus dummy attr HomeStatus room Wohnung define ez_Audio FS20 2332 51 attr ez_Audio model fs20st attr ez_Audio room Esszimmer define FileLog_ez_Audio FileLog /var/InternerSpeicher/fhem/log/ez_Audio-%Y.log ez_Audio attr FileLog_ez_Audio logtype text define ez_LichtRegal FS20 6969 00 attr ez_LichtRegal model fs20st attr ez_LichtRegal room Esszimmer define FileLog_ez_LichtRegal FileLog /var/InternerSpeicher/fhem/log/ez_LichtRegal-%Y.log ez_LichtRegal attr FileLog_ez_LichtRegal logtype text</pre>	<p>Feiertage Bayern Pfad und Name der Logdatei Pfad zum FHEM-Verzeichnis fhem soll über port 7072 per telnet zugänglich sein Pfad zu <code>fhem.save</code> – Speicherort der Schaltzustände Verbose Loglevel (5=<u>alles</u> loggen, 1=nur Katastrophen)</p> <p>Die Definition des CUL mit dessen FHT-ID</p> <p>Port 8083 – das ‚normale‘ fhem-frontend</p> <p>Port 8084 – für iPhone und andere Geräte mit kleinem Bildschirm</p> <p>Port 8085 – für das iPad und andere Tablet-PCs</p> <p>autocreate - Festlegung, dass für Funktelegramme bisher unbekannter Geräte automatisch ein Gerät in fhem erzeugt wird, incl. Logfile und weblink</p> <p>Feiertage Bayern – siehe holiday2we</p> <p>Definition einer globalen dummy-Variable, die dem Raum ‚Wohnung‘ zugeordnet wird</p> <p>Ein per autocreate erzeugter und dann umbenannter Aktor mit Namen <code>ez_Audio</code>, Typ FS20, Hauscode 2332, Tastencode 51 und Attributen zu Modell und Raum</p> <p>Ebenfalls von autocreate erzeugt – die Logdatei zum Gerät <code>ez_Audio</code></p> <p>Das nächste Gerät, selbe Abfolge wie zuvor</p>
---	--

Zu Anfang der Datei finden Sie alle Attribute des (Pseudo-)Geräts `global`, siehe auch **fhem-Konfiguration: Das besondere Gerät „Global“**.

Der Speicherort dieser Datei und damit der Betriebssystem-Pfad zum Ordner „fhem“ ist als Attribut `modpath` angegeben:

attr global **modpath** /var/InternerSpeicher/fhem

(hier muss natürlich der zu Ihrem Gerät passende Pfad stehen! Wenn dieses Attribut in Ihrer `fhem.cfg` –Datei bereits eingestellt ist und `fhem` funktioniert, lassen Sie diese Angabe am besten unverändert).

Ebenso wichtig ist die Pfadangabe zur `fhem` Logdatei:

attr global **logfile** /var/InternerSpeicher/fhem/log/fhem-%Y-%m.log

(Hier muss natürlich der zu Ihrem Gerät passende Pfad stehen!). Durch diese Angabe wird eine separate Datei pro Monat pro Jahr angelegt. Das erlaubt eine zielgerichtete Archivierung, da die Logdateien je nach Anzahl und Art der verwendeten Geräte recht schnell recht groß werden können.

Neben der `fhem.cfg` spielt auch die Datei **fhem.save** eine zentrale Rolle. In dieser Datei werden die Schaltzustände aller Geräte gespeichert, wenn Sie `save` ausführen – so werden alle Geräte nach einem Neustart von `fhem` wieder mit ihrem letzten Schaltzustand angezeigt.

Der Speicherort dieser Datei ist festgelegt durch

attr global **statefile** /var/InternerSpeicher/fhem/log/fhem.save

fhem.cfg und Includes

Da jede Geräte-Definition sowie alle Attribute, `notify`, `at`, `weblinks` usw. in der Datei `fhem.cfg` abgelegt werden, wird diese schnell lang und unübersichtlich. Mit `include` können Sie einzelne Konfig-Abschnitte in separaten Dateien speichern und beim Start von `fhem` zusammenführen.

Infos finden Sie in der [commandref \(http://fhem.de/commandref.html#include\)](http://fhem.de/commandref.html#include) und einem [Wiki-Artikel \(http://fhemwiki.de/wiki/Jalousie_und_Beleuchtung_in_mehreren_R%C3%A4umen\)](http://fhemwiki.de/wiki/Jalousie_und_Beleuchtung_in_mehreren_R%C3%A4umen).

Zubehör

ZWAVE Power Paket



ZWAVE Outdoor Paket



ZWAVE Security Paket

Support:
support.pi3g.com
support@pi3g.com

Kontakt:
kontakt@pi3g.com
www.pi3g.com