

# HANSER



## Leseprobe

zu

## „Smart Home mit FHEM“

von Peter A. Henning

Print-ISBN: 978-3-446-45873-4  
E-Book-ISBN: 978-3-446-46098-0  
E-Pub-ISBN: 978-3-446-46247-2

Weitere Informationen und Bestellungen unter  
<http://www.hanser-fachbuch.de/978-3-446-45873-4>

sowie im Buchhandel

© Carl Hanser Verlag, München

# Inhalt

<b>1</b>	<b>Read me first!</b> .....	<b>1</b>
1.1	Über FHEM .....	2
1.1.1	Ein paar technische Details .....	3
1.1.2	Andere Systeme .....	3
1.1.3	Wie einsteigen? .....	4
1.1.4	Internet of Things .....	5
1.2	Hinweise für den häuslichen Frieden .....	6
1.2.1	Anleitung für nicht gesetzestreue Bürger .....	7
1.2.2	Immer noch vom WAF .....	8
1.3	Über dieses und andere Bücher .....	9
1.3.1	Weitere Quellen .....	9
1.3.2	Konventionen für dieses Buch .....	9
<b>2</b>	<b>Einstieg in FHEM</b> .....	<b>11</b>
2.1	Vorbereitende Arbeiten .....	12
2.2	Installation von FHEM .....	15
2.3	Devices .....	18
2.3.1	Anwesenheit feststellen mit <i>PRESENCE</i> .....	18
2.3.2	<i>dummy</i> -Device für eine Leuchte .....	21
2.3.3	Nützliche Befehle für Devices .....	25
2.4	Events .....	26
2.4.1	<i>notify</i> - Schalten mit Events .....	28
2.4.2	<i>at</i> - Schalten zu bestimmten Zeiten .....	29
2.4.3	<i>IF</i> und <i>DOIF</i> - Schalten mit Bedingungen .....	31
2.4.4	Nützliches für Events .....	33
2.5	Wie findet man Hilfe, wenn es nicht funktioniert? .....	35

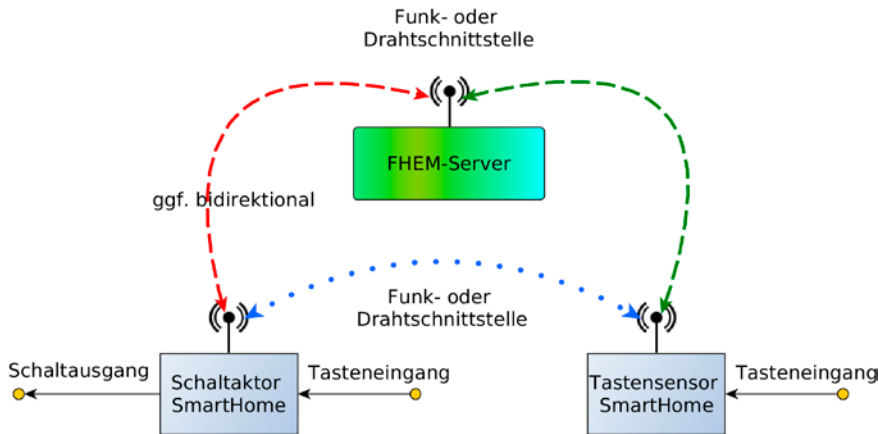
<b>3</b>	<b>Geräte hinzufügen</b>	<b>37</b>
3.1	SmartHome über LAN und WLAN	38
3.1.1	Shelly-Schaltaktoren und Dimmer	39
3.1.2	Anbindung einer FritzBox	48
3.1.3	Wenn das WLAN nicht reicht	50
3.2	Funksysteme für das SmartHome	51
3.2.1	HomeMatic	52
3.2.2	ZigBee	57
3.2.3	mySensors – ein Funksystem für Arduinos	62
3.3	Drahtgebundene Interfaces	63
3.3.1	1-Wire Bus und FHEM	64
3.3.2	Arduino mit ConfigurableFirmata	68
3.4	Beliebige Geräte steuern	70
<b>4</b>	<b>FHEM als Server</b>	<b>75</b>
4.1	Server-Pflege	77
4.1.1	Update von FHEM	77
4.1.2	Server-Update	78
4.1.3	Backup	79
4.1.4	Mitschrift im Logfile	80
4.2	FHEM über das Netz steuern	82
4.2.1	Die Standardoberfläche FHEMWEB	82
4.2.2	telnet-Zugriff	84
4.2.3	REST-Schnittstelle	85
4.3	SmartHome-Sicherheit	85
4.3.1	Cloud oder nicht Cloud – das ist hier die Frage	86
4.3.2	FHEM grundlegend absichern	87
4.3.3	HTTPS und SSL für FHEM	88
4.3.4	Schutz gegen Cross-Site-Request-Forgery	89
4.4	Kommunikationsserver	90
4.4.1	FHEM <sup>2</sup>	90
4.4.2	Server für das MQTT-Protokoll	92
4.4.3	Nachrichten als Mail versenden	93
4.4.4	Instant Messenger Telegram	95
4.5	Ordnung muss sein	98
4.5.1	Räume	98
4.5.2	Gruppen in Räumen	100
4.5.3	Gruppierung mit <i>readingsGroup</i>	101
4.5.4	Gruppierung mit <i>structure</i>	102
4.5.5	Auswahl mehrerer Devices	104

4.6	Dateien zum Ändern .....	105
4.6.1	Die Konfigurationsdatei .....	106
4.6.2	Das Modul <i>ConfigDB</i> .....	107
4.6.3	FHEM selbst erweitern .....	110
<b>5</b>	<b>Licht und Schatten .....</b>	<b>113</b>
5.1	Lichtsteuerung .....	113
5.1.1	Gruppierung von Leuchten mit <i>LightScene</i> .....	114
5.1.2	Licht als Zustandsautomat .....	116
5.1.3	Bewegtes Licht .....	118
5.2	Farbspiele .....	119
5.2.1	Farbtemperatur .....	119
5.2.2	Farbfunktionen .....	120
5.2.3	Farbauswahl .....	123
5.2.4	Farbschemata .....	127
5.3	Rollläden steuern .....	128
5.3.1	Zeitschaltung mit Wochenprogramm .....	129
5.3.2	Beschattungssteuerung .....	131
5.4	Noch smarter mit <i>ASC</i> .....	134
5.4.1	Attribute des ASC-Devices .....	136
5.4.2	Readings und Attribute für Rollläden .....	136
<b>6</b>	<b>Komfortabel bedienen .....</b>	<b>141</b>
6.1	Eingabe in <i>FHEMWEB</i> .....	142
6.1.1	Icons für Geräte und Räume .....	142
6.1.2	Stilangaben mit CSS .....	142
6.2	Widgets zur Eingabe .....	143
6.2.1	Überschreiben von Widgets .....	143
6.2.2	Einfache Widgets .....	145
6.2.3	Drehknopf und <i>DateTimePicker</i> .....	147
6.2.4	Icon-Widgets .....	148
6.2.5	LCARS-Panel mit SVG .....	149
6.3	Styles .....	151
6.3.1	Fertige Stilarten .....	152
6.3.2	Analyse der Stilarten .....	155
6.3.3	Umbau der Stilart <i>dark</i> .....	160

<b>7</b>	<b>Mit Daten steuern</b>	<b>163</b>
7.1	Einzeldaten aus Sensoren	163
7.1.1	Temperatur und Feuchte	165
7.1.2	Rate und Summe	166
7.1.3	Gleitender Mittelwert	168
7.1.4	Werte überwachen mit <i>THRESHOLD</i>	169
7.1.5	Smarte Zirkulationspumpensteuerung	170
7.2	Devices für Wetterdaten	172
7.2.1	Open Data des Deutschen Wetterdienstes	173
7.2.2	Wetterdaten von <i>PROPLANTA</i>	174
7.2.3	Wetterdaten von DarkSky	176
7.3	Daten aus Webseiten abgreifen	177
7.3.1	Rheinpegel anzeigen	177
7.3.2	Gamma-Ortsdosisleistung messen	182
7.3.3	Verkehrslage auf dem Weg zur Arbeit	184
7.4	Astronomiedaten mit Astro	185
7.4.1	Grundlegende Definition	186
7.4.2	Der Sonne Lauf	188
<b>8</b>	<b>Visualisierung mit FHEM</b>	<b>191</b>
8.1	Darstellung von Einzelwerten	191
8.1.1	Werte als Icons	192
8.1.2	SVG-Basics	193
8.1.3	SVG-Farben	196
8.1.4	SVG-Widgets in FHEM nutzen	198
8.1.5	Verbrauchswerte als Säule	200
8.2	Zeitreihen	202
8.2.1	Feinheiten von Logdateien	203
8.2.2	Einfache Plots	206
8.2.3	Bedienung und Optionen von Plots	208
8.2.4	Gplot-Dateien	210
8.2.5	Plot-Tricks für Fortgeschrittene	212
8.3	logProxy	215
8.3.1	Mittelwerte einblenden	216
8.3.2	Wellnessanzeige	217

<b>9</b>	<b>Zeit- und Erinnerungssteuerung</b>	<b>219</b>
9.1	Digitale Kalender mit FHEM nutzen	219
9.1.1	<i>Calendar</i> - und <i>CALVIEW</i> -Device	221
9.1.2	Kalenderdateien selbst erzeugen	223
9.2	Listenverwaltung	225
9.2.1	Konfiguration von <i>PostMe</i>	225
9.2.2	Anzeige von Listen	228
9.2.3	Verteilung von Listen	230
9.2.4	Listenverwaltung über Telegram	232
9.3	Automatisierung mit <i>YAAHM</i>	233
9.3.1	Modus, Sicherheitszustand und Tagestyp	234
9.3.2	Tagesprofile	236
9.3.3	Geräte-Aktionen	238
9.3.4	Wochenprofile	238
9.4	Weitere Ablaufsysteme	240
9.4.1	Automatisierung mit <i>HOMEMODE</i>	240
9.4.2	Wecker in FHEM	240
9.4.3	Alarmanlage in FHEM	242
<b>10</b>	<b>Andere Frontends</b>	<b>243</b>
10.1	Bedienung mit <i>Dashboard</i>	243
10.2	Bedienung mit <i>FLOORPLAN</i>	246
10.3	Steuerung mit Telegram	249
10.4	Tablet-UI	253
10.4.1	Installation und erste Schritte	254
10.4.2	Anwendungsbeispiel Sportplatzampel	256
10.4.3	Eigenes Widget executor	260
10.4.4	Eigenes Widget bar	263
10.4.5	Anordnungsschemata	267
<b>11</b>	<b>Tablets, TV und Audiosysteme</b>	<b>269</b>
11.1	Android-Tablet fernsteuern	269
11.1.1	Tablet mit <i>AMAD</i> einrichten	270
11.1.2	Automagic Flows	272
11.1.3	Fully Webbrowser	273
11.1.4	Altgeräte	274
11.2	Unterhaltungselektronik	274
11.2.1	Universelle Fernbedienungen	275
11.2.2	Audiowecker	276

11.3	Sprachausgabe mit FHEM .....	278
11.3.1	Android-Geräte .....	279
11.3.2	Ausgabe mit Audiosystemen .....	281
11.3.3	MP3-Datei zur Sprachausgabe erzeugen .....	283
11.3.4	Routine zur Sprachausgabe .....	285
<b>12</b>	<b>Sprich mit FHEM .....</b>	<b>289</b>
12.1	Spracherkennung STT .....	290
12.1.1	STT mit Android-Geräten .....	290
12.1.2	Weitergabe von Texten zur Steuerung .....	291
12.2	Sprachsteuerung mit <i>Talk2Fhem</i> .....	292
12.2.1	Konfiguration .....	293
12.2.2	Zeit- und Ereignisspezifikationen .....	296
12.3	Sprachsteuerung mit <i>Babble</i> .....	296
12.3.1	Geräte mit Babble steuern .....	298
12.3.2	Worte für das SmartHome .....	301
12.3.3	Temperatur ansagen lassen .....	303
12.3.4	Hauszustand und Hausmodus steuern .....	304
12.4	ChatBot mit RiveScript .....	306
12.4.1	RiveScript zur ChatBot-Programmierung .....	307
12.4.2	Notruf in FHEM .....	308
12.4.3	ChatBot und Babble .....	311
12.5	Alexa, Assistant und andere .....	312
<b>13</b>	<b>Perl und Module verstehen .....</b>	<b>315</b>
13.1	Perl-Grundlagen .....	315
13.1.1	Variablen, Arrays und Hashes .....	316
13.1.2	Anweisungen .....	318
13.1.3	Prozedurale Aspekte .....	319
13.1.4	Objektorientierte Aspekte .....	320
13.1.5	Eingebaute Funktionen .....	321
13.1.6	Comprehensive Perl ArchiveNetwork CPAN .....	322
13.2	Perl und FHEM .....	322
13.2.1	Funktionen und Variablen .....	323
13.2.2	Gliederung von Modulen .....	324
13.2.3	Modulfunktionen .....	325
	<b>Index .....</b>	<b>327</b>



**Bild 3.1** Sensor-Aktor-Prinzip mit FHEM als Zentrale: Auch ein Bewegungsmelder kann als „Tasten“-Sensor agieren.

Für viele Interfaces vereinfacht sich die Einbindung von Geräten in FHEM durch das Anlegen eines *autocreate*-Device:

```
define autocreate autocreate
```

Dieses Device (das nur einmal benötigt wird) versucht bei unbekanntem Events (die z. B. von einem Funkinterface kommen können), die zugehörigen Devices, *FileLog*- und *SVG*-Devices anzulegen. Letzte dienen der Datenaufzeichnung und Visualisierung und werden in Kapitel 8 behandelt. Als Namen der neu durch *autocreate* angelegten Devices werden typischerweise Bestandteile der Seriennummern der Geräte verwendet, was diese Namen etwas schwer handhabbar macht. Das Device *autocreate* stellt einen Befehl *createlog* zur Verfügung, mit dem auch für schon existierende Devices entsprechende *FileLog*- und *SVG*-Devices generiert werden können.

## ■ 3.1 SmartHome über LAN und WLAN

In mehr als 50 % aller Fälle wird in Privathaushalten der Kontakt zum Internet durch eine FritzBox von AVM hergestellt – und die ist in der Tat recht gut an FHEM angebunden. FHEM lässt sich auch direkt auf der FritzBox installieren, allerdings war das in früheren Versionen einfacher (genauer: bevor der Hersteller AVM beschloss, selbst in das Geschäft mit der Hausautomatisierung einzusteigen). Heute möchte ich eigentlich davon abraten, denn Kleinstcomputer wie der Raspberry Pi sind so sehr im Preis gefallen, dass sich die Umstände mit einer FHEM-Installation auf der FritzBox überhaupt nicht mehr lohnen. Also betrachte ich die FritzBox im Folgenden nur als „externes“ Gerät für FHEM.



Auch andere internetfähige Geräte im Haus können mit FHEM gesteuert werden (siehe dazu Abschnitt 7.3 und Kapitel 11 zu Geräten der Unterhaltungselektronik).

Fast jeder Haushalt verfügt heute über WLAN und viele nutzen dieses schon, um für sich einen Mehrwert zu erzielen. Auch im Außenbereich gibt es immer mehr Geräte, die sich einer WLAN-Anbindung bedienen. Beispielsweise sind bestimmte Gartengeräte und Mähroboter mit FHEM steuerbar. Mit gewissen Einschränkungen gilt dies auch für das System *HomeMatic IP* (nicht verwechseln mit HomeMatic, das in Abschnitt 3.2.1 behandelt wird, siehe [https://wiki.fhem.de/wiki/HomeMatic\\_IP](https://wiki.fhem.de/wiki/HomeMatic_IP)).



An dieser Stelle noch einmal der Tipp: Kaufen Sie nicht erst das Gerät und fragen dann andere nach der Einbindung in FHEM, sondern machen Sie die Einbindung zum Auswahlkriterium beim Kauf.

Seit einigen Jahren sind sehr preiswerte kleine Baugruppen auf dem Markt, die einen Mikrocontroller mit einem WLAN-Interface vereinen (und leider fast ausschließlich aus China stammen). Stellvertretend für diese Systeme sei hier der ESP8266 genannt, der für ca. 2 - 3 € erhältlich ist. Er verfügt über eine Vielzahl an Ein- und Ausgängen, ist leicht mit jedem Computer programmierbar und damit für eine Vielzahl von Mess- und Steuerungsaufgaben einsetzbar. Viele dieser Systeme sind darüber hinaus mit einer alternativen Firmware bestückbar und beherrschen danach das universelle MQTT-Protokoll (siehe Abschnitt 4.4.2). Recherchieren Sie dazu unter dem Begriff *Tasmota*.

Damit wurde eine regelrechte Revolution ausgelöst. Für Preise zwischen 10 und 20 € lassen sich heutzutage auch im Baumarkt WLAN-fähige Schaltsteckdosen oder Schalteinsätze für Schalterdosen kaufen, und im Internet finden sich tausende von Anleitungen, um mit einem ESP8266 schicke Sensorik und Aktorik aufzubauen.

Der große Vorteil WLAN-gesteuerter Geräte, Sensoren und Aktoren ist, dass sie in das existierende Netzwerk eingebunden werden und nicht separate Funkschnittstellen (eventuell sogar auf anderen Frequenzen) benötigen. Das ist sehr bequem, solange man nur wenige Sensoren und Aktoren in Betrieb hat. Werden es jedoch mehr, ist das nicht mehr so günstig, denn sie müssen sich die verfügbare Bandbreite mit vielen anderen Geräten teilen, und diese Bandbreite kann nicht ohne größere Investitionen aufgerüstet werden.

### 3.1.1 Shelly-Schaltaktoren und Dimmer

Eine sehr preisgünstige Serie von WLAN-fähigen Aktoren ist seit Sommer 2018 verfügbar – und so gut und preiswert, dass die Aktoren innerhalb kürzester Zeit ausverkauft waren. Es handelt sich dabei um die Modelle des bulgarischen Herstellers Allterco Robotics (<https://shelly.cloud>). Drei dieser Aktoren sind in Bild 3.2 dargestellt:

- Shelly 1 ist ein einkanaliger Schaltaktor, der mit einem Durchmesser von 41 mm eigentlich in jede Schalterdose passt und ca. 10 € kostet. In der Version Shelly 1 PM kann er auch die Leistung messen.

- ShellyPlug ist ein Zwischenstecker für ca. 30 € (inzwischen auch in einer kleineren Version als ShellyPlug S erhältlich).
- Shelly 2 bzw. 2.5 kann entweder als zweikanaliger Schaltaktor betrieben werden oder als Aktor für elektrisch betriebene Rollläden. Der Aktor enthält darüber hinaus eine Leistungsmessung (als Summe beider Schaltkanäle beim Shelly 2). Die Kosten liegen bei ca. 20 €.
- Shelly 4 Pro (im Hutschienengehäuse) ist ein vierkanaliger Schaltaktor mit Leistungsmessung in jedem Kanal (Kosten ca. 65 €).
- Shelly RGBW2 ist ein vierkanaliger Dimmaktor für 12/24 V-LED-Beleuchtung (Kosten ca. 25 €).



**Bild 3.2** Schaltaktoren von Allterco Robotics (von links: Shelly 1 in einer Hutschienhalterung, Shelly 1, Shelly 4 Pro, Shelly 2)

Das Interessante an diesem System ist die Vielfalt der Bedienmöglichkeiten (abgesehen von den am Aktor vorhandenen Schalteingängen):

- Durch einen eingebauten *Access Point Mode* können die Aktoren unabhängig von einem bereits existierenden WLAN betrieben werden und sind dann per Webbrowser steuerbar.
- Bei Einbindung in ein existierendes WLAN können sie
  - über einen Webbrowser, der auf den Shelly-internen Web-Server zugreift,
  - aus FHEM über ein dezidiertes FHEM-Modul *Shelly*
  - oder über das MQTT-Protokoll (auch aus FHEM heraus) gesteuert werden (siehe Abschnitt 4.4.2).

- Bei Einbindung in ein existierendes WLAN können die Aktoren zusätzlich mit der Cloud des Herstellers verbunden werden und sind ebenfalls durch eine proprietäre App zu schalten (das empfehle ich ausdrücklich **nicht**, siehe dazu Abschnitt 4.3.1).

Neben dem reinen Schaltvorgang kann man auch Schaltzeiten und die Nutzung von Dämmerungszeiten programmieren, doch das ist eher irrelevant für den FHEM-Nutzer, der so etwas in seiner SmartHome-Installation zentral verwalten will. Der Shelly 1 kann darüber hinaus an niedrigen Spannungen betrieben sowie mit alternativer Firmware geflasht werden. Das wollen wir uns ersparen und im Folgenden darstellen, wie man einen Shelly 1 out of the box mit FHEM verwenden kann.

### 3.1.1.1 Shelly 1 out of the box

Die Unterschiede der verschiedenen Modelle von Allterco Robotics sind in Tabelle 3.1 aufgeführt. Alle im Folgenden für Shelly 1 beschriebenen Schritte werden prinzipiell genauso für die anderen Aktoren ausgeführt.

**Tabelle 3.1** Vergleich der verschiedenen Schaltaktoren von Allterco Robotics

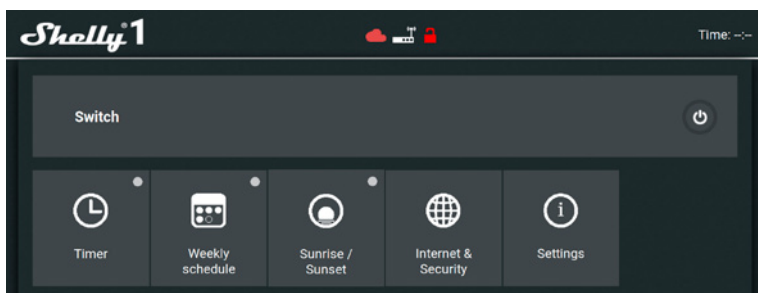
Modell	Anzahl Schaltkanäle	Anzahl Dimmkanäle	Anzahl Messkanäle	Besonderheit
ShellyPlug (S)	1	0	1	
Shelly 1 (PM)	1	0	0 (1)	Auch mit 24 – 60 V betreibbar
Shelly 2 (2.5)	2	0	1 (2)	Auch als Rollladenaktor nutzbar
Shelly 4	4	0	4	Hutschiene
Shelly RGBW2	0	4	1	Color+W oder 4 × Weiß

1. Schließen Sie das zu schaltende Gerät – sagen wir eine Leuchte – nach dem beiliegenden Schaltplan an. Verbinden Sie einen eventuell in der Dose vorhandenen Schalter oder Taster ebenfalls mit dem Shelly.
2. Verbinden Sie den Shelly nun mit der Netzspannung. Achtung: Dabei besteht immer Lebensgefahr! Beachten Sie unbedingt die Hinweise in Kapitel 1.
  - braunes oder schwarzes Kabel = Phase → Anschluss L
  - blaues Kabel = Neutralleiter → Anschluss N
3. Der Shelly befindet sich jetzt im Access Point Mode. Er baut also ein eigenes WLAN auf, unter einem Namen ähnlich wie *shelly1-5BA61C*. Suchen Sie dieses Netzwerk mit dem Smartphone oder einem anderen WLAN-fähigen Endgerät, und verbinden Sie sich mit diesem Netzwerk.
4. Öffnen Sie einen Browser, und geben Sie die Adresse 192.168.33.1 ein. Sie sind dann mit dem Shelly verbunden und können ihn durch Anklicken des Schaltersymbols ein- oder ausschalten.
5. Wählen Sie den Bereich INTERNET & SECURITY → WIFI MODE – CLIENT aus, und geben Sie dort den Namen sowie das Passwort des eigenen WLAN ein. Außerdem sollte eine

festen IP-Adresse vergeben werden, damit der Shelly auch von FHEM gefunden werden kann. Der Gateway-Eintrag sollte nur erfolgen, wenn Sie möchten, dass der Shelly seine interne Uhr (rechts oben in Bild 3.3) mit dem Internet synchronisiert. Bitte lesen Sie hierzu unbedingt den Text im nachfolgenden Kasten.



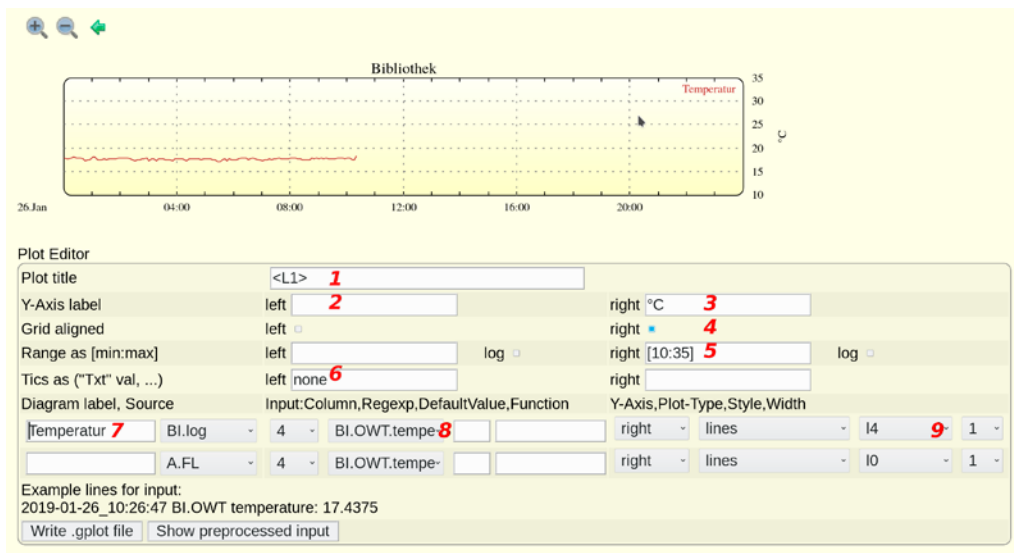
Sobald der Shelly via Gateway einen Zugang ins Internet hat, beschafft er sich **auch bei abgeschaltetem** Cloud-Zugang nicht nur die aktuelle Zeit, sondern mithilfe der externen IP-Adresse Ihres SmartHome auch Informationen in Form von Geodaten (allerdings nur die des Zugangspunktes des Providers). Es ist derzeit nicht davon auszugehen, dass ohne Einschalten des Cloud-Zugangs weitere Informationen ins Internet gelangen. Bei eingeschaltetem Cloud-Zugang sieht das ganz anders aus. Hier erfolgt ein erheblicher Datenaustausch zwischen Shelly und den Servern des Herstellers.



**Bild 3.3** Access Point Mode eines Shelly 1: Das rote Cloud-Symbol zeigt, dass der Zugang zur Shelly Cloud nicht besteht, das rote Schloss-Symbol zeigt, dass der Zugang nicht durch ein Passwort gesichert ist.

6. Der Shelly wird sich nun mit Ihrem WLAN verbinden. Sie können ihn dann durch Eingabe der vorher vergebenen festen IP-Adresse über den Webbrowser steuern (siehe Bild 3.4; wir verwenden im Folgenden 192.168.0.164).

Beim Einbau eines Shelly in eine Unterputzdose müssen natürlich die einschlägigen Regeln für gefährliche Spannungen befolgt werden (siehe Kapitel 1). Für die Einbindung des Shelly in FHEM können die anderen Einstellmöglichkeiten der Weboberfläche ignoriert werden. Lediglich bei einem notwendigen Firmware Update muss in dieser Oberfläche SETTINGS → FIRMWARE UPDATE angeklickt werden.



**Bild 8.7** Beispielhafte Eingaben im Plot-Editor (Nummerierung siehe Text)

Klickt man anschließend auf WRITE .G PLOT FILE, werden diese Einstellungen in eine Datei SVG\_BI.log\_1.gplot im Ordner /opt/fhem/www/gplot gesichert, der Plot ist nicht mehr leer, sondern zeigt die Temperaturdaten aus der Logdatei. Darüber hinaus taucht die neue Gplot-Datei SVG\_BI.log\_1.gplot danach auch in der Liste editierbarer Dateien auf, die in Bild 4.16 dargestellt ist.

### 8.2.3 Bedienung und Optionen von Plots

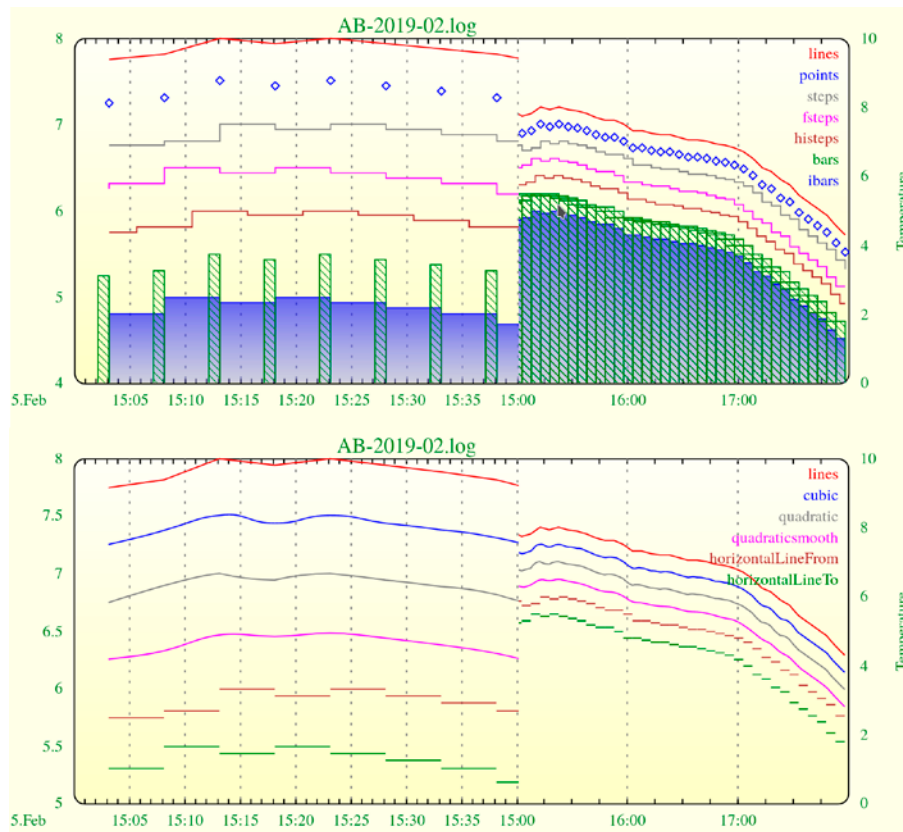
In der Raumübersicht werden natürlich nicht die komplizierten Einstellungen des Plot-Editors angezeigt, sondern nur der Plot selbst mit einem Link darunter. Während die „Innereien“ des Diagramms mit dem Plot-Editor festgelegt wurden, kann man die Größe mit einem weiteren Attribut des SVG-Device bestimmen: attr SVG\_BI.log\_1 plotsize 600,300 würde das Diagramm mit einer Größe von 600 × 300 Pixeln zeichnen.

Links oben in Bild 8.7 sind ferner drei Icons sichtbar:

- Das Plus-Zeichen sorgt für eine Vergrößerung des Plots. Der dargestellte Zeitausschnitt wird also verkleinert.
- Das Minus-Zeichen sorgt für eine Verkleinerung des Plots. Der dargestellte Zeitausschnitt wird also vergrößert.
- Der Pfeil nach links sorgt dafür, dass frühere Zeitausschnitte dargestellt werden.
- Ein Pfeil nach rechts taucht nur auf, wenn man einen früheren Zeitausschnitt betrachtet. Dann kann man mit diesem Pfeil zu späteren Zeitausschnitten blättern.

Nun kann es allerdings zu Problemen kommen, wenn die für den Plot angeforderten Zeitreihen weiter in die Vergangenheit reichen als die Daten in der Logdatei. Am Ende von Abschnitt 8.2.1 wurde dazu das Attribut `createGluedFile` vorgestellt.

Eine weitere Möglichkeit zur Bedienung von Plots ergibt sich, wenn man in Bild 8.8 rechts oben die Label in der Legende anklickt. Hierbei werden dann Möglichkeiten geboten, die anderen Linien im Plot auszublenden oder spezielle Datenwerte anzeigen zu lassen. Mit dem Attribut `captionPos` kann gewählt werden, auf welcher Seite die Legende angezeigt wird (bzw. ob diese auf die Achsen aufgespalten wird).



**Bild 8.8** Verschiedene Linienstile in Plots

Im Plot-Editor können auf der rechten Seite verschiedene Linienstile gewählt werden:

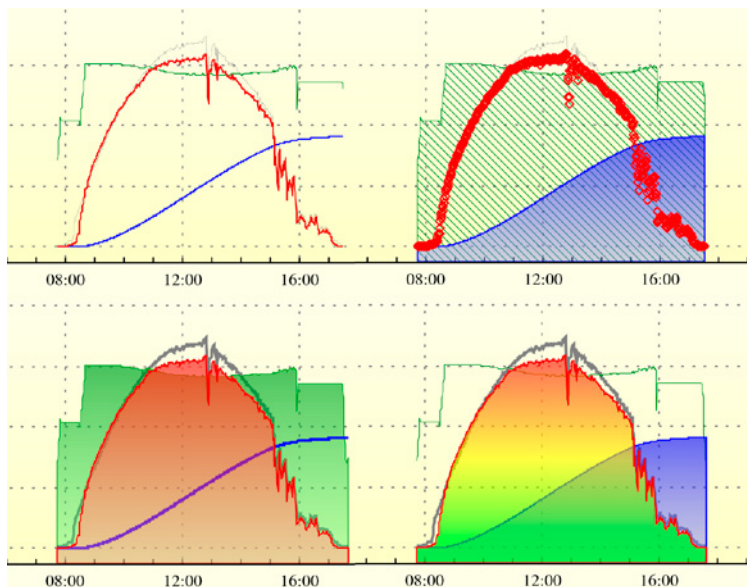
- `lines` stellt einfach eine Verbindung der Datenpunkte dar.
- `points` zeichnet Symbole an die Datenpunkte.
- `steps`, `fsteps` und `histeps` sind Stufenfunktionen, die sich darin unterscheiden, wie die Stufe in Bezug auf den Messzeitpunkt orientiert ist: rechts, links oder mittig.

- `bars` und `ibars` sind Balkendiagramme mit konstanter Balkenbreite bzw. Balkenbreite bis zum nächsten Datenwert.
- `horizontalLineFrom` und `horizontalLineTo` sind horizontale Liniendiagramme, die sich darin unterscheiden, in welche Richtung die Datenlinie in Bezug auf den Messzeitpunkt orientiert ist: rechts oder links.
- `cubic`, `quadratic` und `quadraticsmooth` sind Glättungen mit kubischen Splines oder quadratischen Bézierfunktionen.

Neben diesen Linienstilen sind natürlich auch die Linienfarben wählbar (Position 9 in Bild 8.7). Manche dieser Farben sind mit einer Füllung der entsprechenden Kurve verbunden (siehe dazu das Beispiel in Abschnitt 8.1.3). Bild 8.8 zeigt alle Linienstile für einen einfachen Plot, Bild 8.9 zeigt die gegenseitige visuelle Beeinflussung verschiedener Stile. Weitere Attribute sind in der `COMMANDREF` erläutert.

### 8.2.4 Gplot-Dateien

Bevor wir uns mit komplizierteren Darstellungen befassen, soll ein Blick auf die ominöse Gplot-Datei geworfen werden. Das Format dieser Datei basiert auf der seit 1986 bekannten und immer noch verwendeten Open Source-Anwendung *Gnuplot*. Dabei handelt es sich um eine kommandozeilenorientierte Anwendung zur grafischen Darstellung von Daten (mehr dazu unter <http://www.gnuplot.info>). Die von FHEM verwalteten Gplot-Dateien im Ordner `/opt/fhem/www/gplot` sind als dritter Block in Bild 4.16 zu sehen. Sie sind mit dem FHEM-Editor zu bearbeiten.



**Bild 8.9** Verschiedene Linien- und Füllstile für Plots

## ■ 12.4 ChatBot mit RiveScript

Ein ChatBot ist ein Programm, mit dem man Dialoge führen kann. Doch warum sollte man das mit seinem SmartHome machen wollen? In Abschnitt 12.1 habe ich erklärt, wo die Grenzen der meisten Spracherkennungssysteme liegen: Sie lassen nur sehr geringen Spielraum bei der Kommunikation. Mit der „semantischen Analyse“ in Abschnitt 12.3 haben wir das ein wenig umgehen können. Trotzdem kann auch Babble nicht beliebige Eingaben erkennen. Mit einem halbwegs intelligenten Dialog lässt sich ein Nutzer jedoch dahin führen, den richtigen Befehl für eine Steuerungsaufgabe zu finden.



### Beispieldialog mit FHEM

User: „Hallo“ → FHEM: „Hallo, ich bin Jeannie. Wer bist Du?“ (Hier antwortet der ChatBot, weil der unvollständige Eingangssatz nicht mit einem Kommando belegt ist.)

User: „Ich bin Peter.“ → FHEM: „Schön, Dich kennenzulernen, Peter!“ (Hier antwortet wieder der ChatBot.)

User: „Was kannst Du für mich tun?“ → FHEM: „Ich kann für Dich das Haus steuern.“ (Hier antwortet wieder der ChatBot.)

User: „Was kannst Du steuern?“ → FHEM: „Ich kenne die Geräte Licht, ...“ (Hier antwortet wieder der ChatBot, holt sich aber die Liste der Geräte aus den Babble-Daten.)

User: „Wie kannst Du Licht steuern?“ → FHEM: „Sprich: Licht, gefolgt von einer Ortsangabe und dem Wort an oder aus“ (Hier antwortet wieder der ChatBot, indem er den Hilfetext für das Gerät Licht von Babble holt.)

User: „Welche Orte kennst Du?“ → FHEM: „Ich kenne die Orte Wohnzimmer, ...“ (Hier antwortet wieder der ChatBot, holt sich aber die Liste der Orte aus den Babble-Daten.)

User: „Schalte das Licht in der Sitzgruppe an.“ → FHEM sagt nun gar nichts mehr, sondern schaltet das Licht in der Sitzgruppe an.

Eine weitere Anwendung für ChatBots sind Notsituationen, etwa in einer Seniorenwohnung, deren einziger Bewohner gestürzt ist. Zwar werden derzeit diverse Forschungsanstrengungen unternommen, Sensoren für diese Situation zu entwickeln, die Realität zeigt aber, dass die wirklichen Fallsituationen davon nicht erfasst werden.



### Notfalldialog mit FHEM:

User: „Hilfe!“ → FHEM: „Ich bin Jeannie. Wenn es sich um einen Notfall handelt, sprich bitte Notfall. Wenn nicht, sprich bitte Abbrechen“ (Hier antwortet der ChatBot.)

User: „Notfall!“ → FHEM: „Ich rufe die Notfallnummer an, bitte habe einen Augenblick Geduld, Hilfe wird so schnell wie möglich kommen.“ (FHEM führt die eingestellte Alarmierungsroutine durch.)



Zur Nutzung eines Chatbots in FHEM muss das Perl-Modul RiveScript installiert werden. Dazu gibt man Folgendes auf der Linux-Konsole ein:

```
cpan RiveScript
```

Mehr zur Verwendung von *CPAN* finden Sie in Abschnitt 13.1.6.

### 12.4.1 RiveScript zur ChatBot-Programmierung

RiveScript ist nicht nur der Name des ChatBot-Programms, sondern darüber hinaus auch die Bezeichnung einer einfachen Programmiersprache für ChatBots. Ein gutes Tutorial darüber findet man auf der Seite <https://www.rivescript.com/docs/tutorial>. RiveScript-Dateien (Endung *.rive*) kann man problemlos mit einem Texteditor erstellen, muss aber darauf achten, dass sie mit einem UTF8-Encoding gespeichert werden, wenn sie Umlaute enthalten. Die erste Datei mit dem Namen *.rive* wird im Verzeichnis */opt/fhem/rivescript* angelegt und hat den Inhalt

```
// Name des Hausgeistes
! var name = ... hier steht der Name des Hausgeistes ...
//# Begrüßungsfragen
+ hallo
- Hallo, ich bin <bot name>. Wer bist Du ?

+ wer bist du
- Ich bin <bot name>. Wer bist Du ?
- Ich bin <bot name>, eine mit RiveScript programmierte AI.

+ hallo *
* <formal> == <bot name> => Ebenfalls Hallo. Wer bist Du ?
- Ich heiße nicht <star>, sondern <bot name>. Wer bist Du ?
```

Damit man diesen ChatBot auch ohne FHEM aufrufen und testen kann, wird noch ein kleines Perl-Programm hinzugefügt. Es kommt in die Datei */opt/fhem/rivescript/testrive.pl* und hat den Code

```
use RiveScript;
#-- Create a new RiveScript interpreter.
my $rs = new RiveScript(utf8=>1);
#-- Load a directory of replies.
eval {$rs->loadDirectory (".")};
#-- Sort all the loaded replies.
$rs->sortReplies;
#-- Chat with the bot.
while (1) {
    print "Du> ";
    chomp (my $msg = <STDIN>);
    my $reply = $rs->reply ('localuser',$msg);
    print "Bot> $reply\n";
}
```

Wechseln Sie nun auf der Linux-Kommandozeilenebene in das Verzeichnis, in dem die RiveScript-Datei sowie das Perl-Programm stehen, und geben Sie `perl testrive.pl` ein. Sie sehen dann einen Prompt `Du>` in der Kommandozeile und können Ihre Fragen und Antworten eingeben. RiveScript ermöglicht das Setzen von Variablen, die Überprüfung von Bedingungen, Variationen der Antworten und den Aufruf beliebiger Perl-Programme. Es ist damit hervorragend geeignet, unser FHEM-SmartHome zu ergänzen. Das Programm wird durch `Ctrl-C` wieder verlassen.

### 12.4.2 Notruf in FHEM

In diesem Abschnitt soll der ChatBot in FHEM eingebunden werden, um den vorangehend beschriebenen Notfalldialog führen zu können. Dazu muss zuerst der Initialisierungscode der Datei `99_myUtils.pm` ergänzt werden:

```
my $rive = 0;
my $riveinterpreter;
my $rivedevice;

sub myUtils Initialize($$){
    my ($hash) = @_ ;
    if(eval {require RiveScript;1;} ne 1) {
        Log 1,"The RiveScript module is missing from your Perl ".
            "installation - chatbot functionality not available";
    } else {
        RiveScript->import();
        $rive = 1;
        Log 1,"The RiveScript module has been imported ".
            "successfully, chatbot functionality available";
    }
    ... Hier weiterer Code zur Initialisierung von 99_myUtils...
}
```

In dieselbe Datei kommen fünf kurze Unterprogramme. Das erste erzeugt den Chatbot, das zweite liefert ihm den vom Menschen gesprochenen Satz und erhält seine Antwort.

```
sub createRive(){
    $riveinterpreter = new RiveScript(utf8=>1);
    #--load a directory of replies
    eval{$riveinterpreter->loadDirectory ("./rivescript")};
    #-- sort all the loaded replies
    $riveinterpreter->sortReplies;
}

sub getRive($){
    my ($sentence) = @_ ;
    createRive()
        if( !defined($riveinterpreter) );
    my $reply = $riveinterpreter->reply ('localuser',$sentence);
    $reply = "Es tut mir leid, das habe ich nicht verstanden"
```

# Stichwortverzeichnis

## Symbole

1-Wire 64  
99\_myUtils.pm 168, 223, 169, 221, 308,  
49, 94, 111  
<fhem-ip> 16

## A

Alarm 242  
alarmclock 240, 277  
Alexa 313  
alias 25  
allowed\_WEB 87  
AMAD 270, 290, 279  
Anwesenheitserkennung 29  
AptToDate 78  
Arduino 56, 62, 64  
– Firmata 68  
Astro 185, 130  
at 29  
autocreate 38, 271  
Automagic Premium 271  
AutoShuttersControl 134  
average 168

## B

Babble 292, 296  
– Ansage 303  
– ChatBot 311  
– Geräte 298  
– Wortlisten 301  
– YAAHM 304

backup 79  
Beleuchtungsstärke 132  
Benutzername/Passwort 87  
Bewegungsmelder 116  
Bodenfeuchte 164  
BOSEST 275, 281  
BRAVIA 275

## C

Calendar 222  
CALVIEW 223  
Cascading Stylesheets CSS 196, 230, 142  
ChatBot 303, 306  
cmdlcon 24  
CMY 120  
CommandRef 20  
ConfigDB 107  
CPAN 307, 285, 322  
csrfToken 83  
csrfToken 85, 89  
CUL/CUN 51

## D

Dämmerung, zivil bis nautisch 188  
DarkSky 176  
Dashboard 243  
define 19  
defmod 25  
delete 25  
deleteattr 25  
deletereadng 26

Deutscher Wetterdienst 173  
 Device specific help 20  
 DevSpec 104  
 devStatelcon 23, 129  
 dewpoint 165  
 Dimmaktor 47  
 DOIF 170, 238, 32, 118, 49, 71, 130, 133,  
 277, 282  
 dummy 22, 49, 71, 116, 123, 282

## E

Edit files 105  
 Einkaufsliste 226  
 EnOcean 51  
 ESP8266 39  
 espeak 278  
 eventMap 33, 144  
 event-min-interval 34  
 Event monitor 26  
 event-on-change-reading 34  
 event-on-update-reading 34

## F

Farbschemata 127  
 Farbspezifikationen 193  
 Farbtemperatur 119  
 Farbumrechnung 121  
 Farbverlauf 197  
 FB\_CALLMONITOR 48  
 FHEM2FHEM 91  
 FHEM-Attribute 21  
 fhem.cfg 106  
 FHEM-Devices 18  
 FHEM-Events 26  
 FHEM-Module 20, 324  
 fhem.pl 75  
 FHEM-Räume 98  
 FHEMWEB 82, 142  
 - columns 100  
 - hiddenroom 99  
 - menuEntries 100  
 FHEM-Wiki 35  
 FileLog 202, 80

FLOORPLAN 246  
 Flows 272  
 FRITZBOX 48  
 FRM 69  
 FS20 51  
 FTUI 253  
 - executor 260  
 - Flex 268  
 - Gridster 267  
 - Installation 254  
 - Widget Bars 264  
 FULLY 273

## G

Gammastrahlung 182  
 Gasverbrauch 164  
 Geburtstagskalender 219, 223  
 getstate 26  
 global 17, 296, 298, 30, 77, 81, 121, 98,  
 324  
 Gnuplot 211  
 Google Assistant 313  
 Google TTS 278, 284  
 Gplot-Datei 210

## H

harmony 275  
 Heizung 172  
 holiday 220  
 HomeMatic 51, 279  
 - BidCoS 55  
 - Peering, Pairing 53  
 - Wired 64  
 HomeMatic IP 39  
 HOMEMODE 240  
 Hotword 312  
 HourCounter 167  
 HSV 121, 126  
 HTTPMOD 177, 184  
 Hue 58  
 HUEDevice 59

**I**

iCalendar 221  
icon 24  
iconPath 142  
Icons 142  
IF 31  
include 107  
init.d 109  
Installer 16  
Interface  
– CUL 52  
– CUN 65  
– HM-LGW-O-TW-W-EU 52  
– HM-MOD-RPI-PCB 52  
– MapleCUN 52  
– RaspBee/Conbee 58  
Internals 21  
ISM-Band 51

**J**

Jeelink 51  
JSON 184

**K**

knob 139  
KNX 63  
Künstliche Intelligenz 289

**L**

LaCrosse 51  
LCARS 195, 264, 150  
LGTV 274  
Lightify 58  
LightScene 114, 276  
Linienstil 197, 209  
list 25  
Logdateien archivieren 205  
Logfile 80  
Loglevel 80  
logProxy 215

**M**

Mail 93  
Matrix Voice 314  
miniDLNA 281  
Mittelwert 216  
– gleitend 168, 215  
MQTT 39, 58, 92  
MQTT2\_DEVICE 92  
MQTT2\_SERVER 92  
MQTT\_GENERIC\_BRIDGE 93  
Müllabfuhrkalender 224  
MYSENSORS 63

**N**

notify 291, 309, 28, 98, 123, 286

**O**

OWServer 65  
OWTHERM 66  
OWX 65

**P**

Perl  
– Array 317  
– Dokumentation 322  
– Funktion 319  
– Hash 317  
– Module 316  
PHTV 275  
Plot-Editor 208  
Plots 206  
Polly TTS 278  
PostMe 225  
PRESENCE 19  
PROPLANTA 174

**R**

Raspberry Pi 11, 52, 58, 71  
Readings 21  
readingsGroup 101, 138

Regenmenge 164  
 Regulärer Ausdruck 204, 300, 207, 27  
 remotecontrol 276  
 rename 25  
 rereadcfg 107  
 REST 85  
 restore 78  
 RFHEM 91  
 RFtrx433 51  
 RGB 121, 126  
 Rheinpegel 178  
 RiveScript 306  
 Rollladen 238, 46, 128  
 Rollladenaktor 102

## S

SamsungAV 275  
 Save config 17  
 Scalable Vector Graphics SVG 150  
 Select style 152  
 set 19  
 setreading 26  
 setstate 26  
 Shelly 39, 118  
 shutdown 77  
 SiSi 95  
 SNIPS 314  
 Snowboy 313  
 Solaranlagen 164  
 Sonnenauf- und -untergang 187  
 SONOS 275, 281  
 sox 284  
 Speech Synthesis Markup Language 231,  
 284, 286  
 Sportplatzampel 256, 70  
 Sprachausgabe 231  
 SSL 88  
 statistics 166  
 Stromverbrauch 164  
 structure 102  
 Style  
 - bright 154  
 - dark 153  
 - default 152

- f18 154  
 - own 162  
 Submenüs 99  
 SUNRISE\_EL 30, 130  
 SVGX\_widget 198, 200  
 systemd 109

## T

Tablet-UI 253  
 Talk2Fhem 292  
 Tasker 272  
 Tasmota 39  
 Taupunkt 165  
 Telegram 220, 271  
 - Keyboards 249  
 - Listenverwaltung 232  
 - Sprachausgabe 285  
 TelegramBot 95  
 telnet 84  
 Text2Speech 279  
 THRESHOLD 169  
 todoist 225  
 ToDo-Liste 227  
 TrådFri 58  
 trigger 34  
 TYPE 76, 100

## U

update 77  
 User Interface (UI) 141

## V

verbose 81  
 Verkehrslage 184  
 VIERA 275  
 VolumeLink 276

## W

Wasserverbrauch 164  
 weblink 175, 114  
 webViewControl 274, 280

Wellness 217  
Widget  
– colorpicker 123  
– DateTimePicker 147  
– Icon- 148  
– knob 147  
– slider 144  
– UZSU 146  
widgetOverride 143  
Widgets 143  
WLAN 39  
Wochenprogramm 238, 129  
wunderlist 225

**X**

XPath 180

**Y**

YAAHM 132, 276  
– Aktionen 238  
– Modus und Zustand 234  
– PostMe 228  
– Tagesprofil 236  
– Tagestyp 235  
yowsupp 95

**Z**

Zeitreihen 202  
ZigBee 51, 57  
Zirkulationspumpe 170  
Zustandsautomat 171, 116  
ZWave 51