

#makers  
DO IT.

Inklusive Tablet-Interface  
und Sprachsteuerung

Peter A. Henning

# SMART HOME MIT FHEM

Individuelle und flexible  
Open-Source-Hausautomatisierung

HANSER

Henning  
**Smart Home mit FHEM**



**BLEIBEN SIE AUF DEM LAUFENDEN!**

Hanser Newsletter informieren Sie regelmäßig über neue Bücher und Termine aus den verschiedenen Bereichen der Technik. Profitieren Sie auch von Gewinnspielen und exklusiven Leseproben. Gleich anmelden unter

**[WWW.HANSER-FACHBUCH.DE/NEWSLETTER](http://WWW.HANSER-FACHBUCH.DE/NEWSLETTER)**

# Alles für Maker



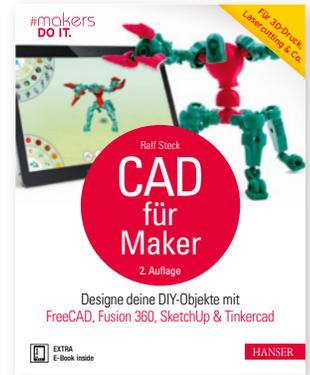
Hüwe  
**IoT at Home**  
 ISBN 978-3-446-45661-7



Regele  
**Mach was mit 3D-Druck!**  
 ISBN 978-3-446-44781-3



Rother  
**3D-Drucken ... und dann?**  
 ISBN 978-3-446-45062-2



Steck  
**CAD für Maker**  
 ISBN 978-3-446-45681-5



Pomaska  
**3D-Fotos und -Videos**  
 ISBN 978-3-446-45630-3



Kehrer, Philipp, Rens  
**Lasercutting**  
 ISBN 978-3-446-45039-4



Steck  
**CNC-Fräsen für Maker**  
 ISBN 978-3-446-45491-0



Bartmann, Donges  
**Open Robots für Maker**  
 978-3-446-45489-7



Jänisch, Donges  
**Mach was mit Arduino!**  
 ISBN 978-3-446-45128-5



Bertko, Weber  
**Home, Smart Home**  
 ISBN 978-3-446-45061-5



Schmidt  
**Raspberry Pi programmieren mit C/C++ und Bash**  
 ISBN 978-3-446-45342-5



Hüwe  
**Raspberry Pi für Windows 10 IoT Core**  
 ISBN 978-3-446-44719-6

Peter A. Henning

# Smart Home mit FHEM

Individuelle und flexible Open-Source-Hausautomatisierung

HANSER

Der Autor:

*Peter A. Henning*, Weingarten



Alle in diesem Buch enthaltenen Informationen wurden nach bestem Wissen zusammengestellt und mit Sorgfalt geprüft und getestet. Dennoch sind Fehler nicht ganz auszuschließen. Aus diesem Grund sind die im vorliegenden Buch enthaltenen Informationen mit keiner Verpflichtung oder Garantie irgendeiner Art verbunden. Autor und Verlag übernehmen infolgedessen keine Verantwortung und werden keine daraus folgende oder sonstige Haftung übernehmen, die auf irgendeine Art aus der Benutzung dieser Informationen – oder Teilen davon – entsteht. Ebenso wenig übernehmen Autor und Verlag die Gewähr dafür, dass beschriebene Verfahren usw. frei von Schutzrechten Dritter sind. Die Wiedergabe von Gebrauchsnamen, Handelsnamen, Warenbezeichnungen usw. in diesem Werk berechtigt auch ohne besondere Kennzeichnung nicht zu der Annahme, dass solche Namen im Sinne der Warenzeichen- und Markenschutz-Gesetzgebung als frei zu betrachten wären und daher von jedermann benutzt werden dürften.

Bibliografische Information der Deutschen Nationalbibliothek:

Die Deutsche Nationalbibliothek verzeichnet diese Publikation in der Deutschen Nationalbibliografie; detaillierte bibliografische Daten sind im Internet über <http://dnb.d-nb.de> abrufbar.

Dieses Werk ist urheberrechtlich geschützt.

Alle Rechte, auch die der Übersetzung, des Nachdruckes und der Vervielfältigung des Buches, oder Teilen daraus, vorbehalten. Kein Teil des Werkes darf ohne schriftliche Genehmigung des Verlages in irgendeiner Form (Fotokopie, Mikrofilm oder ein anderes Verfahren) – auch nicht für Zwecke der Unterrichtsgestaltung – reproduziert oder unter Verwendung elektronischer Systeme verarbeitet, vervielfältigt oder verbreitet werden.

© 2019 Carl Hanser Verlag München

[www.hanser-fachbuch.de](http://www.hanser-fachbuch.de)

Lektorat: Julia Stepp

Herstellung: Björn Gallinge

Titelmotiv: © Sebastian Völkel, unter Verwendung von Grafiken von © istockphoto.com/Andrey Suslov

Coverrealisation: Max Kostopoulos

Satz: Kösel Media GmbH, Krugzell

Druck und Bindung: NEOGRAFIA, a.s., Martin-Priekopa (Slowakei)

Printed in Slovakia

Print-ISBN: 978-3-446-45873-4

E-Book-ISBN: 978-3-446-46098-0

E-Pub-ISBN: 978-3-446-46247-2

# 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>	

# 1

## Read me first!

Egal, ob man die Zeitung liest, im Internet surft oder fernsieht: Überall begegnet einem der Begriff SmartHome. Das mit digitalen Geräten vollgepflasterte *Intelligente Heim* (oder wörtlich übersetzt das *Schlaue Heim*) soll irgendwie komfortabler sein als das normale Heim. Meine klare Aussage dazu: Das stimmt, wenn man es richtig macht. Wenn nicht, dann nervt es nur.

Doch wie geht „richtig“? Darauf hat nun leider jeder Hersteller eine andere Antwort, die natürlich immer das eigene Geräteprogramm in den Mittelpunkt stellt. Der eine Hersteller bietet nur Leuchten und seltsam aussehende batteriebetriebene Schalter an, der zweite setzt auf hässliche Zwischenstecker und fast jeder liefert dazu noch eine eigene „App“, die angeblich ganz intuitiv zu bedienen ist, aber mit den Geräten der Konkurrenz nicht umgehen kann.

Nur sehr wenige Hersteller wagen sich über Systemgrenzen hinaus. In der Regel bleibt dies Fachbetrieben überlassen, die oft vertraglich gebunden sind. Will man also die gesamte Bandbreite der modernen Technik für sein SmartHome nutzen, bleibt – von Ausnahmen abgesehen – nur ein Weg: selbst Hand anzulegen. Damit ist nicht gemeint, elektronische Geräte selbst zu bauen. Das kann man zwar, muss es aber nicht tun. Ebenso wenig heißt dies, dass man programmieren lernen muss, um sein SmartHome zu realisieren. Man muss auch nicht Tausende von Euro an Kosten einplanen, um durch ein wenig Digitalisierung das eigene Heim komfortabler zu machen.

Die ersten Schritte lassen sich vielmehr mit einem normalen Computer (Desktop oder Laptop) vollziehen, den fast jede Frau und jeder Mann sowieso besitzen. Etwas logisches Denken hinzugenommen – und schon kann es losgehen, etwa mit der täglichen Verkehrsmeldung entlang der Fahrtroute zum Arbeitsplatz oder der Erinnerung, dass übermorgen die Restmülltonne herausgestellt werden muss. Die Einrichtung eines SmartHome ist also eines ganz sicher: ein Prozess, bei dem wir neue Werkzeuge ausprobieren, um unser Heim zu verbessern.

Die Menschheitsgeschichte ist voll mit Beispielen dafür, wie der Gebrauch neuer Werkzeuge unsere Lebensumstände und letztlich auch uns selbst verändert hat. Das ist beim Thema SmartHome nicht anders. Dieses Buch führt deshalb im Laufe der Kapitel immer weiter von der einfachen Lichtfernbedienung weg, bis das SmartHome in Kapitel 12

schließlich mit Technologien ausgestattet wird, die wir heute unter dem Begriff *Künstliche Intelligenz (KI)* zusammenfassen.



Mein erster Tipp: Planen Sie nicht Ihr komplettes SmartHome auf einmal, sondern gehen Sie schrittweise vor. Die Ideen für Anwendungsfälle (Use Cases) kommen erst mit der Nutzung.

Wichtig ist deshalb, von Anfang an ein System zu verwenden, das als Universalwerkzeug gelten und immer wieder neue Anwendungsfälle integrieren kann. Das SmartHome-System FHEM, das Schweizer Taschenmesser unter den Hausautomatisierungslösungen, bietet genau diese Vorzüge.

## ■ 1.1 Über FHEM

Ich erspare mir, eine eigene Kurzbeschreibung von FHEM zu erfinden, sondern zitiere die offizielle FHEM-Website (<http://www.fhem.de>):

*„FHEM ist ein in Perl geschriebener, GPL lizenzierter Server für die Heimautomatisierung. Man kann mit FHEM häufig auftretende Aufgaben automatisieren, wie z.Bsp. Leuchten / Rollläden / Heizung / usw. schalten, oder Ereignisse wie Temperatur / Feuchtigkeit / Stromverbrauch protokollieren und visualisieren. (...) Ausgesprochen wird es ohne h, wie bei feminin.“*

Die Besonderheit von FHEM ist, dass es buchstäblich hunderte verschiedener Gerätetypen in eine gemeinsame Anwendung einbinden und diese mit einer gemeinsamen Oberfläche steuern kann. Mit FHEM als SmartHome-Zentrale kann der Bewegungsmelder des Herstellers X die Leuchten der Herstellers Y steuern und gleichzeitig auch noch eine Sprachnachricht an das Smartphone der Marke Z senden.

FHEM ist also nicht deshalb so gut, weil es das schnellste oder modernste System zur Hausautomatisierung ist, sondern weil es durch eine Vielzahl von Modulen und Interfaces mit nahezu allen anderen SmartHome-Systemen interagieren kann. Eine im Wesentlichen deutsche Nutzergemeinschaft von etwa 20 000 Usern entwickelt FHEM stetig weiter.

FHEM ist ursprünglich einmal die Abkürzung für „Freundliche Hausautomatisierung und Energiemessung“ gewesen, davon hat sich das System jedoch weit entfernt. Wir danken FHEM den Ideen und dem Engagement von Rudolf König, der auch heute noch eine wesentliche Rolle bei der Weiterentwicklung spielt und dem ich hiermit für seine Arbeit danken möchte.

Zum Abschluss dieser einleitenden Bemerkungen möchte ich noch auf die Gralshüter von FHEM hinweisen: Seit Mitte 2016 gibt es den eingetragenen Verein FHEM e.V., der nur

sehr wenige Mitglieder hat. Das sind im Wesentlichen die Betreiber der Webseiten, Foren und Software-Repositories. Hinzu kommt noch eine etwas größere Zahl von Fördermitgliedern, die dem Verein Mittel zur Verfügung stellen, sich aber aus Richtlinien etc. heraushalten. Wer sich länger mit der Open Source-Szene beschäftigt, wird schnell verstehen, dass diese eher wenig demokratische Struktur ein Garant für langfristige Verfügbarkeit, Pflege und Weiterentwicklung von FHEM ist.

### 1.1.1 Ein paar technische Details

FHEM wird auf einem Computer installiert, der unter Ihrer eigenen Kontrolle steht. Es ist in der Regel Cloud-frei, kommt ohne spezialisierte „App“ daher und bedarf keiner besonderen Benutzungsoberfläche (was aber nicht heißt, dass es diese nicht als kostenlosen Zusatz gibt). Perl (nicht: Pearl), die Sprache, in der FHEM geschrieben wurde, ist eine schon etwas ältere Programmiersprache, die 1987 von Larry Wall entworfen wurde (siehe dazu Kapitel 13). *GPL* ist die Abkürzung für die *Gnu Public License*, unter der die Bestandteile von FHEM veröffentlicht und frei genutzt werden können.

Zu Bedienung von FHEM benötigt man lediglich eine so genannte Laufzeitumgebung für Perl und einen Webbrowser. Die meisten Definitionen und Befehle in FHEM lassen sich in einer einfachen Skriptsprache verfassen, die weitgehend selbsterklärend ist. Erst wenn man tiefer einsteigen möchte, ergibt sich das Erlernen einiger Elemente der Sprache Perl sozusagen nebenbei und fast wie von selbst.

Die Einfachheit des zentralen Konzeptes von FHEM und seine leicht erlernbare Bedienung erlauben es, mal schnell zwischen Abendessen und Tagesschau etwas an der Steuerung zu ändern, um etwas Neues auszuprobieren (es sei zur Wahrung des häuslichen Friedens aber von dieser Vorgehensweise abgeraten).



Kinderleicht? Warum lernen dann nicht schon Kinder das algorithmische Denken mit FHEM? Das ist in der Tat eine gute Frage. Seit etwa zwei Jahren verfolge ich als nebenläufiges Projekt den Bau eines Demonstrationsmodells für ein FHEM-SmartHome, das in Schulklassen zur Verfügung gestellt werden kann. Voraussichtlich im Sommer 2019 soll es fertiggestellt und auf der LEARNTEC, an deren Organisation ich mitwirke, ausgestellt werden.

### 1.1.2 Andere Systeme

Bevor ich mehr über FHEM schreibe, will ich es in Relation zu anderen SmartHome-Servern setzen, die ich persönlich für die besten Systeme halte (siehe Tabelle 1.1). Diese SmartHome-Server unterscheiden sich stark in der Optik und Usability. Auch können sie mit unterschiedlichen Frontends zur Bedienung und Visualisierung ausgestattet werden. Diese Vielfalt und die Unterschiede zwischen den Systemen sollen in diesem Buch nicht

beschrieben oder bewertet werden. Zu stark spielen dabei persönliche Vorlieben eine Rolle. Von der Bedienbarkeit der jeweiligen Frontends sollte man sich besser selbst einen Eindruck verschaffen. Eine interessante Möglichkeit ist ferner, ein solches anderes System als Frontend zu verwenden und FHEM als universelles Backend einzusetzen. Eine Kommunikation ist beispielsweise über das MQTT-Protokoll möglich (siehe Abschnitt 4.4.2).

**Tabelle 1.1** SmartHome-Systemsoftware im Vergleich

System/URL	Lizenz/Sprache	Bemerkungen
FHEM <i>http://www.fhem.de</i>	Open Source-Server; geschrieben in Perl; leicht erweiterbar; deutsches System	Universell; Interfaces für nahezu alle gängigen SmartHome-Systeme vorhanden; mit 20 000 Teilnehmern sehr große deutschsprachige Community; tiefe Eingriffsmöglichkeiten für Benutzer
OpenHAB <i>https://www.openhab.org</i>	Open Source-Server; geschrieben in Java; Erweiterung eher komplex	Universell; Interfaces für nahezu alle gängigen SmartHome-Systeme vorhanden; kaum Konfigurationsmöglichkeit auf der Weboberfläche, aber schnelle Resultate
Domoticz <i>http://www.domoticz.com</i>	Open Source-Server; geschrieben in C++; Erweiterung in Blockly, Lua, Python oder Bash	Universell; Interfaces für einige SmartHome-Systeme vorhanden
IP Symcon <i>https://www.symcon.de</i>	Kommerziell; deutsches System; Basisversion ca. 100 €, Vollversion ca. 500 €; mit dezidiert Serverhardware: 250 – 750 €	Viele Interfaces für SmartHome-Systeme
Loxone <i>http://www.loxone.com</i>	Kommerziell, sog. Miniserver ca. 400 €, Erweiterungen jeweils ca. 100 – 500 €	Hardware und Software = komplette SmartHome-Zentrale Viele Erweiterungen für SmartHome-Systeme, jeweils spezielle Hardware

### 1.1.3 Wie einsteigen?

Mein eigener Einstieg in das Thema SmartHome ergab sich, als wir bei der Modernisierung unseres Wohnzimmers den Schalter einer Deckenleuchte an anderer Stelle haben wollten. Meine Vorstellungen, wo welche Wände für das Kabel aufzustemmen waren, trafen bei meiner Frau auf schieres Entsetzen. Also sah ich mich nach einem einfachen Funkschaltssystem um und stieß auf das inzwischen vollkommen veraltete System FS20 und seine Einbindung in FHEM.

Das ist nur wenige Jahre her. Inzwischen braucht meine Frau nur zu sagen: „Schalte die Bewässerung an“, und ein Bewässerungszyklus des Gartens wird gestartet. Oder: „Stelle den Rollladen an der Terrasse auf 70%“, und das Ding fährt herunter. Kurz gesagt: Inzwischen haben wir ein SmartHome, in dem eine Vielzahl von hochmodernen intelligenten Geräten, etwa elektronische Heizkörperthermostate, mit ferngesteuerten unintelligenten Geräten, wie etwa einer motorisierten Wandhalterung für den Fernseher, oder gar alter Technologie, wie dem Teewasserkocher in der Küche, interagieren und ein (meist) harmonisches Ganzes bilden.



Heutzutage ist der Markt so dynamisch, dass fast jede Woche ein neues System auf den Markt kommt und auch schnell wieder verschwinden kann. Deshalb folgt nun ein zweiter wichtiger Tipp zum Einstieg: Kaufen Sie nicht erst Hardware, die Ihnen im Baumarkt oder im Fernsehen gut gefallen hat, und fragen hinterher: „Wie binde ich das in FHEM ein?“ Oder: „Hat nicht jemand Lust, mir ein Modul dafür zu schreiben?“ Besser probieren Sie FHEM erst einmal ohne zusätzliche Hardware aus, so wie dies in Kapitel 2 schrittweise erklärt wird. In Abschnitt 2.5 werden die verfügbaren Hilfsangebote und Informationsquellen aufgeführt. Informieren Sie sich dort, ob das von Ihnen gewünschte Gerät bereits von FHEM unterstützt wird.

### 1.1.4 Internet of Things

Die Integration verschiedenster Sensoren, Aktoren und Geräte in eine gemeinsame Anwendungslogik macht das SmartHome zu einem Bestandteil des Internet of Things (IoT), des weltweiten Internet der Dinge. Diese „digitale Ökologie“ habe ich schon 2000 in der ersten Auflage des *Taschenbuch Multimedia* vorhergesagt (siehe Abschnitt 1.3): Das IoT wird die fünfte Generation (5G) in der Digitalisierung der Welt sein. Netze der vierten Generation stellen uns heute größenordnungsmäßig schon eine Milliarde Knoten bereit: Im Januar 2019 waren 1 Milliarde Computer im Domain Name Service (DNS) registriert, und 2018 gab es rund 4 Milliarden Nutzer fester Internetzugänge sowie 3 Milliarden Nutzer von Smartphones. Ein 5G-Netz aber hat zur Folge, dass wir es bald mit 500 Milliarden digitalen Endgeräten zu tun bekommen, die miteinander kommunizieren können. Das ist die Zukunft, mit Sicherheit.

Allerdings darf die Zukunft nicht darin bestehen, dass jeder Hersteller eines smarten Gerätes dieses auch mit einer „App“ ausstattet, die mit den Servern des Herstellers irgendwo in der Cloud kommuniziert und Daten sammelt. Sie darf auch nicht darin bestehen, ihre SmartHome-Zentrale permanent mit dem Internet kommunizieren zu lassen, denn das verlangt einen Durchgriff durch die eigene Firewall ins Internet, entweder indem die App von außen Daten durch die Firewall senden darf (sehr gefährlich!) oder indem eine interne Komponente (Gateway) von innen Kontakt mit einem Server des Herstellers hält und mit diesem Daten austauscht. Auch das ist gefährlich und datenschutzrechtlich sehr bedenklich. Vor allem wegen der oft grottenschlechten Qualität der

mitgelieferten, billig produzierten Apps ist das ein Risiko, dem ich mein Heim nicht aussetzen will.

Auch aus prinzipiellen Gründen bin ich dagegen, dass kommerzielle Anbieter mit ihren Servern das Internet of Things kontrollieren oder dass die Logik meines SmartHome davon abhängt, ob ein weit entfernter Dienst wie IFTTT (If-This-Then-That) meine Skripte ausführt. Bevor Sie Ihr SmartHome für die Cloud öffnen, sollten Sie Abschnitt 4.3 lesen.

## ■ 1.2 Hinweise für den häuslichen Frieden

Aus gutem Grund gibt es in Europa (und natürlich ganz besonders in Deutschland) eine Vielzahl von gesetzlichen Regelungen für das Errichten und Betreiben elektrischer Anlagen. Wer diese Anlagen nämlich ohne erforderliche Fachkenntnisse errichtet oder verändert, kann nicht nur sein eigenes Leben, sondern auch das aller anderen Nutzer der Anlage gefährden.

Die „Zweite Durchführungsverordnung zum Energiewirtschaftsgesetz“ aus dem Jahr 1987 legt fest, dass bei der Errichtung und Unterhaltung von Anlagen zur Erzeugung, Fortleitung und Abgabe von Elektrizität die „allgemein anerkannten Regeln der Technik“ zu beachten sind. Nach herrschender Rechtsauffassung ist der Errichter und Betreiber für die Einhaltung dieser Regeln selbst verantwortlich. Wer sich dabei nach den DIN-VDE-Bestimmungen richtet, liegt auf der sicheren Seite. Die für die Elektroinstallation wichtigsten Regeln sind in der DIN VDE 0100 „Bestimmungen für das Errichten von Starkstromanlagen mit Nennspannungen bis 1000 V“ niedergelegt.

In der „Verordnung über Allgemeine Bedingungen für den Netzanschluss und dessen Nutzung für die Elektrizitätsversorgung in Niederspannung“ (NAV) aus dem Jahr 2006 ist ferner festgelegt, dass elektrische Anlagen nur durch einen eingetragenen Elektroinstallateur errichtet, erweitert, verändert und unterhalten werden dürfen.

Schließlich gibt es auch noch die Vorschriften des VdS (früher „Verband deutscher Sachversicherer“, heute „VdS Schadenverhütungs GmbH“), deren Beachtung unter Umständen nötig ist, um den Versicherungsschutz für Objekte zu erhalten.



Die gesetzliche Lage hat zur Folge, dass eigentlich auch zum Austausch eines defekten Lichtschalters ein Eintrag in die Handwerksrolle nötig ist. Dementsprechend rate ich zu Beginn dieses Buches ganz eindringlich: Lassen Sie Arbeiten an elektrischen Anlagen, die an das Stromnetz angeschlossen sind und mit Netzspannung arbeiten, nur von eingetragenen Elektroinstallateuren ausführen.

Allerdings hinken diese gesetzlichen Regelungen der Realität hinterher. Denn einerseits gibt es heute auch im Niederspannungsbereich elektrische Anlagen, mit denen man Leib und

Leben gefährden kann. Die Energiedichte eines normalen Smartphone-Akkus ist beispielsweise so groß, dass eine kleine mechanische Beschädigung zu einem Brand oder gar zu einer Explosion führen kann (**nicht** ausprobieren!). Andererseits kann mir niemand erzählen, dass die Vielzahl von Lichtschaltern und anderen elektrischen Bauteilen, die an einem normalen Samstag im Baumarkt verkauft werden, von eingetragenen Elektroinstallateuren eingebaut und angeschlossen wird.

Gegen diese gesetzlichen Regelungen wird also laufend verstoßen. Wir haben unter unseren Mitbürgern im Baumarkt offenbar eine große Zahl von nicht ganz gesetzestreu Menschen. Neben dem vollkommen berechtigten Zweck des Schutzes von Leib und Leben anderer Nutzer haben diese gesetzlichen Regelungen daher vor allem einen Sinn: Verantwortung zu verschieben, beispielsweise zu verhindern, dass irgendjemand, der sein Haus durch einen selbst eingebauten Schalter abgepackelt hat, hinterher dafür Geld von der Versicherung bekommt.



Darum folgt jetzt der zweite wichtige Hinweis zu diesem Buch: Falls Sie in Ihrem SmartHome wirklich selbst Hand anlegen, tun Sie dies bitte nur, wenn Sie bereit sind, den Pfad der Tugend zumindest ein klein wenig zu verlassen. Und bedenken Sie: Was immer Sie tun, Sie selbst tragen die Verantwortung dafür.

Ach ja: Wenn eine extrem wohlmeinende Bundesregierung (oder EU-Kommission) auf den Gedanken käme, den Verkauf von Lichtschaltern in Baumärkten zu verbieten, dann werde ich auswandern (vorzugsweise auf den Mars).

### 1.2.1 Anleitung für nicht gesetzestreue Bürger

Wenn Sie also – mit einem leichten und durchaus angenehmen Gruseln, weil Sie das eigentlich nicht dürfen – den Schraubenzieher ansetzen, um einen Lichtschalter auszutauschen oder ähnliche Arbeiten durchzuführen, beachten Sie unbedingt die folgenden fünf Sicherheitsregeln:

1. **Freischalten:** Schalten Sie immer die Sicherung ab, bevor Sie an spannungsführenden Teilen arbeiten.
2. **Gegen Wiedereinschalten sichern:** Kennzeichnen Sie die abgeschaltete Sicherung eindeutig, und sorgen Sie dafür, dass niemand sie „aus Versehen“ wieder einschaltet, während Sie gerade die Hand am Draht haben.
3. **Spannungsfreiheit feststellen:** Prüfen Sie unbedingt (z. B. mit einem Phasenprüfer, von dem Sie genau wissen, dass er funktioniert), ob Ihr unmittelbarer Arbeitsbereich wirklich spannungsfrei ist.
4. **Erden und kurzschließen:** Sorgen Sie als zusätzlichen Schutz dafür, dass Ihr unmittelbarer Arbeitsbereich mit einem Schutzleiter (Erder) verbunden ist.
5. **Benachbarte spannungsführende Bereiche abdecken**

Arbeiten Sie darüber hinaus immer mit geeigneten Werkzeugen, beispielsweise mit Schraubenziehern, die eine isolierte Klinge haben.



Das Anziehen – auch dünnster – Kunststoffhandschuhe ist ein nützlicher Schutz vor Stromschlägen.

### 1.2.2 Immer noch vom WAF

Mein Buch *SmartHome Hacks* enthält einen Abschnitt, der diesem ähnelt und in dem ich vorsichtig und zur erheblichen Belustigung vieler Frauen in meiner Bekanntschaft darauf hingewiesen habe, dass Frauen und Männer beim Thema SmartHome unterschiedliche Herangehensweisen haben. Nach einiger Zeit erhielten der Verlag und ich ein paar vor Wut geifernde Mails von Menschen, die offensichtlich das Buch zwar nicht gelesen hatten, aber dennoch geschlechterneutrale Darstellung verlangten. Auch wenn das nur ein lauer Wind war und kein regelrechter Shitstorm, werde ich dem nicht nachgeben. Ganz politisch unkorrekt beharre ich aus jahrzehntelanger Erfahrung mit der Technologieakzeptanz von Menschen darauf, dass Frauen und Männer in der überwiegenden Mehrheit unterschiedlich sind.

Die meisten Männer werden sich freuen, wenn der neue SmartHome-Server in einem selbst gebauten Gehäuse mit blinkenden LEDs und vielen Kabeln im Bücherregal vor sich hin schnurrt. Die meisten Frauen werden das nur in Ausnahmefällen ertragen.

Die meisten Frauen werden die Wände hochgehen, wenn die selbst installierte Funkalarmanlage sich nicht auf einfachste Weise ausschalten lässt. Die meisten Männer hingegen werden hingegen das Gehäuse öffnen und die richtigen Kontakte überbrücken.

Das ist keine Wertung, sondern das Ergebnis von 200 000 Jahren menschlicher Geschichte, wie auch andere für das SmartHome wichtige Aspekte (siehe Abschnitt 5.2.1).



Die Bevorzugung von Technik, die erstens unsichtbar ist oder zumindest gut aussieht und zweitens ohne spezielle Eingriffe funktioniert, nenne ich aus historischen Gründen den Woman Acceptance Factor (WAF). Behalten Sie angesichts des WAF bei der Realisierung von SmartHome-Projekten stets den Komfort und die Bedenken Ihrer Partnerin oder Ihres Partners im Auge.

Von nun an kann es mit geschlechterneutralen Hinweisen weitergehen:

- Sorgen Sie dafür, dass man auch in Ihrer Abwesenheit die Systeme wieder in einen definierten Zustand zurückführen kann.
- Sorgen Sie dafür, dass kritische Systeme auch dann autonom laufen, wenn Ihre selbst entworfene Lösung abgestürzt ist.
- Dokumentieren Sie, was Sie gebaut und geändert haben. Ihre Partner und Ihre Erben werden Sie dafür preisen.

## ■ 1.3 Über dieses und andere Bücher

Dieses Buch ist eine praxisorientierte FHEM-Anleitung, die Anfängern den leichten Einstieg ermöglicht und schnelle Resultate erbringt. Dem fortgeschrittenen FHEM-Anwender sollen mit dem Buch neue Ideen vermittelt werden, von denen viele nicht von mir stammen, sondern aus der Experten-Community. Ich versuche, möglichst häufig zwischen praktischen Beispielen und generellen Erklärungen zu wechseln.

### 1.3.1 Weitere Quellen

Zwei weitere Bücher werde ich von Zeit zu Zeit zitieren, weil ich sie selbst geschrieben habe und am besten kenne. Zum einen ist dies das *Taschenbuch Multimedia*, das 2007 in vierter Auflage bei Hanser erschienen ist (nur noch als E-Book lieferbar, ISBN 978-3-446-41156-2). Olle Kamellen? Nein, denn mein damaliges Ziel, eine Vielzahl von Anwendungen ganz knapp formuliert und nahezu nebeneinander in einem Buch zu vereinen, überzeugt mich auch heute noch. Und tatsächlich dienen mir manche Kapitel aus dem Buch – sei es über HTML oder über SVG – auch heute noch als rasches Nachschlagewerk. Das zweite Buch, auf das ich von Zeit zu Zeit verweisen werde, ist *SmartHome Hacks* (ISBN 978-3-96009-012-0), das 2016 bei O'Reilly erschienen ist. Im Gegensatz zu diesem Buch, das sich auf Softwareprojekte mit FHEM beschränkt, enthält es auch viele Ideen für neue und spannende Hardware.

### 1.3.2 Konventionen für dieses Buch

Bestimmte Bestandteile aus FHEM werden immer *kursiv* gesetzt, nämlich die *Devicenames* und die dazugehörigen *Module*.

Codezeilen haben eine andere Schriftart, ein solches Zeichen \ am Ende einer Zeile soll bedeuten, dass diese Zeile \ ohne die Eingabe eines Zeilenumbruches fortgesetzt \ werden muss. Dabei tauchen <Parameter>, die durch \ einen tatsächlichen Wert ersetzt werden müssen, \ in spitzen Klammern auf.

Dieselbe Schriftart wird für Code im Text verwendet, hingegen sind anklickbare Befehle als KAPITÄLCHEN formatiert.



Alle Programmbeispiele sind getestet, allerdings übernehmen weder der Verlag noch ich irgendwelche Garantien und haften auch nicht für Schäden, die sich mittelbar oder unmittelbar aus der Befolgung der Tipps und Beispiele ergeben.



Für eine Probeinstallation von FHEM benötigt man einen internetfähigen Computer unter eigener Kontrolle. Die meisten Menschen verwenden Windows als Betriebssystem – und ja, auch darauf kann man FHEM installieren, ebenso wie auf Apple-Computern. Empfehlen will ich es allerdings nicht, weil dabei eine Vielzahl unterschiedlicher Programme aus unterschiedlichen Quellen benötigt wird. Wie man weiß, kann eine solche Installation den eigenen Computer manchmal in einen nicht mehr kontrollierbaren Zustand versetzen.

Stattdessen rate ich dazu, zum Ausprobieren von FHEM auf einem Windows- oder Apple-Computer eine so genannte virtuelle Maschine (Virtual Machine, VM) einzurichten. Diese VM stellt einen komplett vom Hauptsystem getrennten virtuellen Computer dar, auf dem man eine echte Linux-Umgebung mit FHEM einrichten kann. Das kann man übrigens auch dann machen, wenn man bereits ein Linux-System benutzt, dieses aber nicht durch FHEM und seine Bestandteile verändern möchte. Zwei wichtige Vorteile dieser Herangehensweise sind:

- Das FHEM-System und alle seine Komponenten können wirklich restlos vom „echten“ Computer entfernt werden.
- Die Schritte zur Installation und zum Betrieb von FHEM auf einer VM sind dieselben, die man (vielleicht später) auf einem dedizierten Linux-Computer oder einem Raspberry Pi vollziehen muss.

Als Alternative bietet sich an, für wenig Geld einen Raspberry Pi zu kaufen und darauf Linux nach einer der vielen Anleitungen zu installieren. Wenn Sie diesen Weg gehen wollen, springen Sie bitte gleich zu Abschnitt 2.2.



Eine gute Anleitung zur Erstkonfiguration eines neuen Raspberry Pi finden Sie z. B. unter <https://www.joerg-lohrer.de/2018/05/24/einrichtung-eines-neuen-raspberry-pi-headless>. Wollen Sie dabeibleiben, können Sie auch noch allerhand Optimierungen vornehmen, etwa nach dieser Anleitung: <https://www.elektronik-kompodium.de/sites/raspberry-pi/2002251.htm>.

Nach der Erstinstallation werden die grundlegenden Konzepte von FHEM eingeführt: Devices, Events und die logische Verknüpfung derselben über verschiedene Bedingungen. Spezielle Hardware benötigen Sie dafür noch nicht, lediglich etwas Fantasie.

## ■ 2.1 Vorbereitende Arbeiten

Für die Installation von FHEM auf Ihrem „normalen“ Computer, der damit zum Hostcomputer wird, laden Sie sich zuerst die kostenlose Software VirtualBox in der zu Ihrem Betriebssystem passenden Version unter <https://www.virtualbox.org/wiki/Downloads> herunter und installieren diese. Das dauert gar nicht lange, denn es handelt sich „nur“ um ca. 70–110 Mbyte. Im nächsten Schritt sollten Sie gleich das Oracle VM VirtualBox Extension Pack installieren, das wie VirtualBox selbst von dem Branchenriesen Oracle gepflegt und auf derselben Webseite zur Verfügung gestellt wird. Im dritten Schritt besorgen Sie sich die VBox Guest Additions, z. B. unter [https://download.virtualbox.org/virtualbox/5.2.16/VBoxGuestAdditions\\_5.2.16.iso](https://download.virtualbox.org/virtualbox/5.2.16/VBoxGuestAdditions_5.2.16.iso) (natürlich in der zur VirtualBox passenden Version). Dabei handelt es sich um das Abbild einer DVD, die spezielle Programme für das Gastbetriebssystem enthält.

Nach dem Start des Programms VirtualBox präsentiert sich eine zunächst leere Liste von virtuellen Maschinen. Hier fügen Sie eine solche VM durch das Anklicken von NEU hinzu. Dabei müssen Sie wenige Parameter eingeben. Wir wählen als Namen der VM natürlich *FHEM*, als Betriebssystem *Linux* und als Version ein *Ubuntu (32-bit)* (siehe Bild 2.1).



**Bild 2.1** Anlegen einer neuen virtuellen Maschine in VirtualBox

Das Betriebssystem Linux müssen Sie natürlich erst beschaffen – und wählen dazu eine so genannte Mini-Version von Ubuntu 18.04 Bionic Beaver, die mit 57 Mbyte ebenfalls recht klein ist. Das zugehörige DVD-Abbild finden Sie z. B. unter <http://de.archive.ubuntu.com/ubuntu/dists/bionic/main/installer-i386/current/images/netboot/mini.iso>.

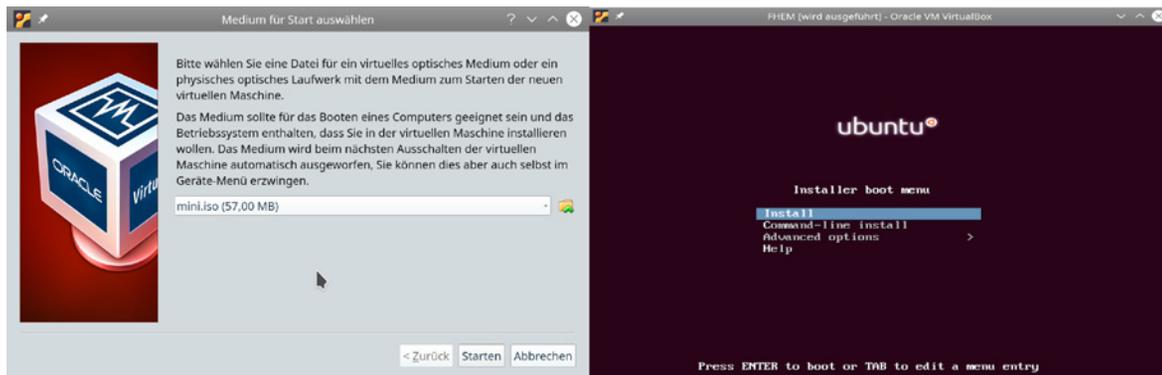
Diese Datei wird heruntergeladen und idealerweise an der gleichen Stelle wie die .iso-Datei mit den *Guest Additions* gespeichert. Dann erst klicken Sie im VirtualBox-Fenster auf WEITER. VirtualBox wird nun Vorschläge machen – etwa zur Größe des Arbeitsspeichers (1024 Mbyte) und zur virtuellen Festplatte (10 GByte). All diese Vorschläge können problemlos und ohne Änderung übernommen werden. Nach dem letzten Schritt wird die neue VM mit dem Namen *FHEM* im VirtualBox-Fenster angezeigt. Sie ist noch ausgeschaltet. Entweder durch den Button ÄNDERN in der Hauptmenüleiste oder durch Aufruf des Kontextmenüs mit der rechten Maustaste ändern Sie jetzt noch den Netzwerktyp auf Netzwerkbrücke, bestätigen mit OK und starten diese VM (siehe Bild 2.2).



**Bild 2.2** Konfigurieren der neuen virtuellen Maschine

Damit greift die VM über den Hostcomputer auf das Netzwerk zu und wird als eigenständiger Computer erkannt. Das bedeutet: Im Netzwerk muss ein so genannter DHCP-Server laufen, der dieser neuen VM eine lokale Internet-Adresse zuteilt. Bei den meisten LAN- und WLAN-Systemen ist das der Fall. Man möchte sich schließlich mit einem neuen Computer im Netzwerk anmelden können. WLAN-Passwörter werden dabei nicht benötigt, denn der Hostcomputer ist ja schon im WLAN angemeldet.

Das System wird danach zuerst nach einer DVD verlangen. Deren Abbild ist genau die bereits heruntergeladene *mini.iso*, die man in das entsprechende Textfeld eintragen muss. Ein Klick auf STARTEN führt zum Startbildschirm des Ubuntu-Installers, den man mit einem einfachen ENTER in Gang setzt (siehe Bild 2.3).



**Bild 2.3** Start-DVD festlegen und Installation starten

Bei den nun folgenden Installationsschritten sind nur wenige Angaben zu machen. Alle Abfragen sind mit ENTER zu bestätigen. Folgende Punkte werden abgefragt:

- die Systemsprache (German)
- der Rechnername im Netzwerk, sagen wir FHEM
- die Größe der virtuellen Festplatte (10 Gbyte), welche einfach bestätigt werden kann (bei der Frage nach der Formatierung muss man manuell JA auswählen)
- der Klarname, der Benutzername und das Passwort des menschlichen Benutzers: Hier sollte man **nicht** *fhem* als Benutzernamen wählen, denn unter diesem Benutzernamen wird FHEM später laufen,
- das zu installierende Softwaresystem: Damit ist die Auswahl der zu installierenden Pakete gemeint. Hier wählt man den untersten Menüpunkt *Minimal Ubuntu Server*.

Das neue Ubuntu-Linux wird all diese Pakete aus dem Internet holen und installieren. Am Ende wird gefragt, ob das neue System gestartet werden soll. Das sollte man **nicht** tun, weil man sonst ruckzuck wieder von der virtuellen Installations-DVD bootet. Natürlich ließe sich auch ganz schnell *F12* auf dem Boot-Screen (siehe Bild 2.4) drücken und als Boot-Device die Festplatte auswählen, doch wer möchte das schon jedes Mal tun ...

Schalten Sie also bitte erst die VM aus: Betätigen Sie die rechte Maustaste im VirtualBox-Fenster, wählen Sie im Kontextmenü SCHLIESSEN → AUSSCHALTEN aus und warten Sie, bis das Konsolenfenster sich geschlossen hat. Entfernen Sie danach mit ÄNDERN → MASSEN-SPEICHER den Anschluss *mini.iso* aus dem virtuellen DVD-Laufwerk. Starten Sie dann die VM neu.

Ergebnis ist zuerst der Boot-Screen und dann eine Konsolenabfrage von Benutzername und Passwort. Gibt man diese ein (für das Beispiel habe ich, naheliegenderweise, den Benutzernamen *phennig* gewählt), werden ein paar Systeminformationen und danach der Kommandozeilenprompt `phennig@FHEM:~$` angezeigt (Bild 2.4).



**Bild 2.4** Boot-Screen und Konsolenansicht des neuen Ubuntu-Servers



Um sich später eventuell Tipparbeit zu ersparen, kann man die VM noch so konfigurieren, dass sie auch Copy-and-Paste-Befehle annimmt. Dafür wird die vorangehend heruntergeladene Datei mit den *GuestAdditions* benötigt. Eine Anleitung dafür findet man unter <https://www.techrepublic.com/article/how-to-enable-copy-and-paste-in-virtual-box>. Insgesamt ist es für eine Probeinstallation einfacher, die Befehle von Hand einzutippen, als den Aufwand zur Installation von Copy-and-Paste auf sich zu nehmen.

## ■ 2.2 Installation von FHEM

Unabhängig davon, ob es sich um eine VM oder um einen Raspberry Pi handelt, ist das System jetzt so weit, dass Sie FHEM installieren können. Melden Sie sich also auf der Linux-Konsole an. Nutzer von Windows oder Apple-Systemen oder durch Tablet und Smartphone verwöhnte Mitbürger werden sich bei einer solchen Bedienung auf der Konsole möglicherweise unwohl fühlen. Sie sollten jedoch bedenken, dass Sie hier ein Server-System installieren, dessen kostbare Systemressourcen nicht durch eine grafische Benutzungsoberfläche verschwendet werden sollen. Sie erschaffen sozusagen einen dienstbaren Geist, der unsichtbar irgendwo im Hintergrund werkeln soll.

Zunächst müssen ein paar Pakete zur Erweiterung geholt werden, indem Sie folgende Zeilen in die Linux-Kommandozeile eintippen und mit ENTER abschließen:

```
sudo apt-get update
sudo apt-get install perl libdevice-serialport-perl \
  libio-socket-multicast-perl
sudo apt-get install libio-socket-ssl-perl libwww-perl \
  libjson-perl
sudo apt-get install libdbd-sqlite3-perl libdbi-perl \
  libtext-diff-perl sqlite3
sudo apt-get install libxml-parser-perl libxml-xpath-perl
sudo apt-get install libxml-treebuilder-perl \
```

```
libhtml-treebuilder-xpath-perl
sudo apt-get install libxml-simple-perl libsoap-lite-perl
sudo apt-get install libcgi-pm-perl openssl-server
```



Für diejenigen, die sich noch nie mit Linux befasst haben, sei noch kurz auf die verwendeten Befehle eingegangen. `sudo` weist das Linux-System an, den nachfolgenden Befehl als „Superuser“ (*root* genannt) auszuführen, also als Systemadministrator. Dafür wird beim ersten `sudo` in einer Sitzung jeweils das Passwort des Benutzers abgefragt. Der ausgeführte Befehl lautet `apt-get install`. Er holt die nachfolgend benannten Softwarepakete aus den Linux-Respositorien und installiert sie. `apt-get update` aktualisiert die im System vorhandenen Softwarepakete. In Zukunft wird die Installation übrigens deutlich einfacher sein, denn ein im Frühjahr 2019 in Arbeit befindliches FHEM-Modul *Installer* erlaubt die Verwaltung der Zusatzpakete per Mausclick.

Im nächsten Schritt holen Sie das FHEM-Softwarepaket und installieren es wie folgt:

```
wget fhem.de/fhem-5.9.deb
sudo dpkg --install fhem-5.9.deb
```

Fertig! Damit haben Sie ein lauffähiges FHEM-System. Doch wie finden Sie dieses System? Geben Sie dazu Folgendes in der Kommandozeile ein:

```
sudo ifconfig
```

In der Antwort auf diesen Befehl taucht eine Zeile ähnlich dieser auf:

```
inet:192.168.0.57 netmask:255.255.255.0 broadcast:192.168.0.254
```

Die erste Adresse 192.168.0.57 ist die lokale Internet-Adresse, die bei Ihnen vielleicht etwas anders aussehen wird. Wenn im Folgenden die Abkürzung `<fhem-ip>` verwendet wird, ist damit diese Zahlenkombination gemeint, die Sie anstelle der Abkürzung verwenden müssen.

Öffnen Sie jetzt auf dem Hostcomputer einen Webbrowser, und tippen Sie Folgendes in die Adresszeile ein:

```
http://<fhem-ip>:8083/fhem
```

Dann bekommen Sie die Hauptseite der frischen FHEM-Installation zu sehen (siehe Bild 2.5).



**Bild 2.5** Die Hauptseite einer frischen FHEM-Installation

Über die Farbwahl und das Icon will ich an dieser Stelle noch kein Wort verlieren. Das folgt später. Auch die Meldung *Security Check* braucht im Moment nicht beachtet zu werden. Sie bedeutet nur, dass noch kein Passwort für FHEM festgelegt wurde. Vier Bereiche müssen aber erläutert werden (siehe Bild 2.5):

- Das leere Textfeld oben rechts neben dem FHEM-Icon („das grinsende Haus“) ist die FHEM-Kommandozeile. Hier werden FHEM-Befehle direkt eingetippt und verarbeitet. Sie können die Kommandozeile z. B. verwenden, um die Meldung *Security Check* loszuwerden, indem Sie darin folgenden Befehl eintippen und mit ENTER bestätigen:

```
attr global motd none
```

- Der oberste Menüblock unter dem FHEM-Icon enthält den Link SAVE CONFIG. Klickt man ihn an, wird die gegenwärtige Konfiguration in die FHEM-Konfigurationsdatei übernommen und steht dann beim nächsten Systemstart automatisch zur Verfügung. Ändert man an der vorhandenen Konfiguration etwas, wird der Text mit einem zusätzlichen roten Fragezeichen verziert, um auf die Änderung der Konfiguration hinzuweisen.
- Der mittlere Menüblock enthält die *Räume* von FHEM. Das sind entweder wirkliche Räume des SmartHome oder Funktionsgruppen, die gemeinsam auf einer Webseite angezeigt werden sollen. Voreingestellt sind nur die Räume *Unsorted* (für allerhand nützliche Geräte und Funktionen) und *Everything*, in dem alle Definitionen zu sehen sind (sehr unübersichtlich). Mehr zu den Räumen von FHEM finden Sie in Abschnitt 4.5.1.
- Der unterste Menüblock enthält Links zu nützlichen Systemwerkzeugen. Beispielsweise schreibt FHEM (mit verschiedenen wählbaren Detailtiefen) alle Systemmeldungen in eine Log-Datei, die mit dem Link LOGFILE angezeigt wird. Klicken Sie in diesem Menüblock auf SELECT STYLE und wählen Sie auf der nachfolgend angezeigten Seite (siehe Bild 2.6) *default* aus. Das ist der (alte) Standardstil für die Anzeige in FHEM, der zur Klarheit der Darstellung für die meisten Screenshots in diesem Buch verwendet wurde. Selbstverständlich können Sie hier auch schon mit den anderen Stilarten experimentieren. Systematisch werden sie aber erst in Abschnitt 6.3 behandelt.



**Bild 2.6** Stilauswahl *default* (der Übersichtlichkeit halber): Man kann aber auch schon mit den anderen Stilarten experimentieren.

Detailinformationen zum untersten Menüblock findet man in folgenden Abschnitten:

- Abschnitt 4.14: LOGFILE
- Abschnitt 2.3.1: COMMANDREF und REMOTE DOC
- Abschnitt 4.5: EDIT FILES
- Abschnitt 6.3: SELECT STYLE
- Abschnitt 2.4: EVENT MONITOR

In den folgenden Screenshots wird das Hauptmenü der Übersichtlichkeit halber meist weggelassen.

## ■ 2.3 Devices

Logischerweise wollen Sie mit FHEM Geräte steuern. FHEM arbeitet jedoch mit einem allgemeineren Konzept, das mit dem englischen Wort *Devices* bezeichnet wird. Devices sind logische Geräte, die nicht unbedingt als echte Hardware vorhanden sein müssen. Zur Einstimmung auf dieses Konzept soll zuerst ein Device erstellt werden, das überprüft, ob ein Bewohner anwesend ist oder nicht. Im zweiten Beispiel wird dann ein Device für eine Leuchte erstellt, wobei man für Testzwecke auch dabei ohne die entsprechende Hardware auskommen kann.

### 2.3.1 Anwesenheit feststellen mit *PRESENCE*

Wir nehmen an, dass der Bewohner über ein Smartphone verfügt, dessen WLAN-Schnittstelle eingeschaltet ist. Ich habe dieses Beispiel seiner Einfachheit wegen ausgewählt.

Falls Sie kein Smartphone oder kein WLAN haben, sollten Sie das etwas kompliziertere Beispiel aus Abschnitt 2.3.2 in Betracht ziehen.

Betritt der Eigentümer des Smartphones das SmartHome, wird sich das Smartphone mit dem eigenen WLAN verbinden und in der Regel eine immer gleiche Internet-Adresse (IP-Adresse) zugeteilt bekommen. Diese kann man im Einstellungsbereich des Smartphones unter den Geräteinformationen finden, sagen wir, diese Adresse sei 192.168.0.55.

Die Anwesenheit des Eigentümers kann festgestellt werden, indem man ein Device mit dem Namen *Owner* erschafft. Devices werden in FHEM durch den Befehl `define` angelegt. Also wird in die FHEM-Kommandozeile folgender Befehl eingegeben:

```
define Owner PRESENCE lan-ping 192.168.0.55 60
```

Der Devicename ist frei wählbar. Sie sollten ihn an Ihre persönlichen Vorlieben anpassen. Der erste Parameter `PRESENCE` ist der Typ des Device, vereinfacht gesagt handelt es sich also bei *Owner* um ein `PRESENCE`-Device. Der Parameter `lan-ping` legt fest, mit welchem Verfahren die Anwesenheit geprüft wird. Die IP-Adresse muss natürlich durch Ihre eigene Adresse ersetzt werden. Der Parameter `60` (Sekunden) steht für das Intervall, in dem das Device die Erreichbarkeit des Smartphones prüft, indem ein „Ping“ im WLAN abgesendet wird.

Ein Druck auf die `ENTER`-Taste schickt den `define`-Befehl ab und FHEM wechselt zur Detailansicht des Device, das nun fleißig mit dem Datensammeln beginnt. FHEM registriert, dass die Konfiguration geändert wurde, und zeigt in dieser Ansicht ein Fragezeichen im `SAVE CONFIG`-Menüpunkt (so wie in Bild 2.6 zu sehen). Klickt man darauf, wird das momentan nur flüchtig definierte Device *Owner* fest in die Konfiguration übernommen.

In der Detailansicht (siehe Bild 2.7) ist oben im ersten Block noch einmal der Name des Device zu sehen, gefolgt vom so genannten *state*. Hierbei handelt es sich um eine kurze Zusammenfassung der wichtigsten Device-Daten, im Folgenden auch als *Status* bezeichnet. In unserem Fall nimmt der Status die Werte `absent` oder `present` an.

Unterhalb dieses Blocks wird in der Detailansicht eine Zeile mit dem Button `SET`, dem Devicenamen, einem Auswahlfeld und einem Textfeld gezeigt (siehe Bild 2.7). Damit stellt FHEM verschiedene Befehle an das Device zur Verfügung, die durch einfache Bedienung mit der Maus abgesetzt werden können. Für manche dieser Befehle ist die Funktion offensichtlich. `set Owner overrideInterval <Wert>` überschreibt z.B. die Intervallfestlegung aus der Definition des Device. Wichtig ist, dass diese Befehle auch direkt als Text in die ganz oben stehende FHEM-Kommandozeile eingegeben werden können. Die Eingabe des Befehls

```
set Owner overrideInterval 10
```

in der FHEM-Kommandozeile löst dasselbe aus wie die Mausbedienung des `SET`-Buttons. Das Textfeld hinter dem Button kann natürlich beliebige Eingaben annehmen, auch solche, die sinnlos wären. Im Modul wird das (hoffentlich) abgefangen. Mit etwas Geschick