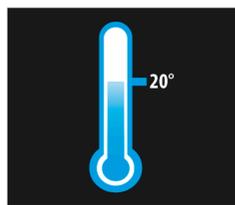
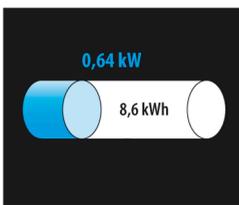


Peter A. Henning

SmartHome Hacks

Hausautomatisierung selber machen



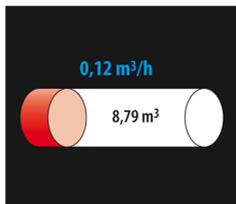
SMARTER
WOHNEN
MIT FHEM
RASPBERRY PI
HOMEMATIC
1-WIRE



Außen
12,3°
43% -2,2mm
997 hPa



Innen
23,8°
51%



O'REILLY®

edition **Make:**

Smarthome Hacks

Hausautomatisierung selber machen

Peter A. Henning

O'REILLY®

Peter A. Henning

Lektorat: Gabriel Neumann

Korrektorat: Friederike Daenecke

Satz: III-satz, www.drei-satz.de

Herstellung: Susanne Bröckelmann

Umschlaggestaltung: Michael Oreal

Fotonachweis Chili-Schote: istockfoto.com, Maksym Narodenko, stockfoto ID 17940210

Druck und Bindung: Stürtz GmbH, Würzburg

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.

ISBN:

Print 978-3-96009-012-0

PDF 978-3-96010-036-2

ePub 978-3-96010-037-9

mobi 978-3-96010-038-6

1. Auflage 2016

Dieses Buch erscheint in Kooperation mit O'Reilly Media, Inc. unter dem Imprint »O'REILLY«. O'REILLY ist ein
Markenzeichen und eine eingetragene Marke von O'Reilly Media, Inc. und wird mit Einwilligung des Eigentümers
verwendet.

Copyright © 2016 dpunkt.verlag GmbH
Wieblinger Weg 17
69123 Heidelberg

Die vorliegende Publikation ist urheberrechtlich geschützt. Alle Rechte vorbehalten. Die Verwendung der Texte und Ab-
bildungen, auch auszugsweise, ist ohne die schriftliche Zustimmung des Verlags urheberrechtswidrig und daher strafbar.
Dies gilt insbesondere für die Vervielfältigung, Übersetzung oder die Verwendung in elektronischen Systemen.

Es wird darauf hingewiesen, dass die im Buch verwendeten Soft- und Hardware-Bezeichnungen sowie Markennamen
und Produktbezeichnungen der jeweiligen Firmen im Allgemeinen warenzeichen-, marken- oder patentrechtlichem
Schutz unterliegen.

Die Informationen in diesem Buch wurden mit größter Sorgfalt erarbeitet. Dennoch können Fehler nicht vollständig
ausgeschlossen werden. Verlag, Autoren und Übersetzer übernehmen keine juristische Verantwortung oder irgendeine
Haftung für eventuell verbliebene Fehler und deren Folgen.

5 4 3 2 1 0

Inhalt

1	READ ME FIRST!	1
	Ein Beispiel.	2
	Welches System?	2
	Warnung vor Gefahren	5
	Anleitung für Anarchisten	7
	Vom WAF	7
	Wolkig mit Aussicht auf Verwirrung	8
	Technische Details	9
	Sensoren und Aktoren	11
	Einheiten und Maße	12
	Links und Ergänzungen	12
2	Hacks zur Lichtsteuerung	13
	Hack: Lichtschalter dorthin verlegen, wo man sie braucht	14
	Detaillierte Planung	14
	Einfache Realisierung	18
	Hack: Licht mit Bedingungen schalten	18
	Lichtsteuerung priorisieren	20
	Zwei Leuchten schalten	21
	Lichtsznarien am Morgen und am Abend	23
	Hack: Licht soll auf die Bewegung von Menschen reagieren	24
	Lichtsteuerung mit FHEM	29
	Hack: Farbtemperatur bei LEDs	30
	Farbtemperatur quantitativ	31
	LED in der Praxis	33
	Hack: Farbtemperatur einstellen	36
	Hack: Temperatur in Farbe umsetzen	38

3	Hacks zur Verbrauchsmessung	41
	Hack: Energieverbrauch aus Betriebszeiten bestimmen	43
	Stromfluss detektieren	44
	Umrechnung in den Energieverbrauch	46
	Hack: Stromverbrauch am Zähler messen	48
	Messung an der Ferraris-Drehscheibe	48
	Messung an der Leuchtdiode	50
	Messung am Zählwerk	51
	Hack: Gas- und Wasserverbrauch messen	52
	Hack: Verbrauch am Zähler mit S0-Ausgang messen	54
	S0-Zähler mit einem Raspberry Pi	55
	S0-Zähler mit dem 1-Wire-System	57
	Hack: Daten loggen und visualisieren	61
	Logging	62
	Visualisierung	63
	Schönere Bilder	66
4	Hacks zur Messung von Umweltdaten	69
	Hack: Funksensoren für Temperatur und Feuchte	70
	Hack: 1-Wire-Sensoren für die Temperatur	72
	Hack: 1-Wire-Feuchtemessung	74
	Hack: 1-Wire-Multisensor	77
	Luftdruckmessung	77
	Helligkeitsmessung	77
	Gaskonzentration messen	79
	Hack: Digitale Sensoren auswerten	80
	Temperatur- und Feuchtesensoren DHT11 und DHT22	80
	Sensoren und Module mit I ² C- und SPI-Interface	82
	Hack: Analoge und digitale Sensoren mit anderen Systemen auswerten	85
	Hack: Weitere Umweltdaten messen	86
	Hack: Datenlogger	89
	Hack: Daten archivieren	91
	Hack: Behaglichkeitsmessungen	94
	Behaglichkeitsmessung in FHEM	98
	Hack: Vermeidung von Schimmel	99
	Hack: Kalibrierung von Sensoren	102
	Kalibrierung von Thermometern	102
	Kalibrierung von Hygrometern	103
	Kalibrierung von Gassensoren	104
5	Hacks für die Heizung	107
	Hack: Wunschtemperatur in jedem Raum	107
	Hack: Heizbedarf ermitteln und nutzen	111
	Heizbedarf aus Ventilstellungen ermitteln	114
	Hack: Heizkurve optimieren und missbrauchen	116

Hack: Partybetrieb, Ferienprogramm und Spartasten nutzen	120
Hack: Tür-Fenster-Kontakte anbinden	122
Hack: Heizungsanlage ins SmartHome integrieren	123
eBus-Adapter für Heizungsanlagen von Vaillant, Weishaupt und Wolf.	124
Optolink-Adapter für Heizungsanlagen von Viessmann.	127
Hack: Zirkulationspumpe wie und wann einschalten?	128
Hack: Zirkulationspumpe durch ein Fertiggerät steuern.	131
Hack: Zirkulationspumpe mit einem selbst gebauten Interface steuern	132
Hack: Zirkulationspumpe durch einen Zustandsautomaten steuern	134
Hack: Solarthermischen Energieertrag messen	137
Hack: Visualisierung der Heizungsanlage	139
6 Hacks für ein sicheres Heim und Grundstück	143
Hack: Abschreckung durch Lichteffekte	145
Hack: Fenster- und Türenzustand erkennen	147
Fenster mit 1-Wire-Chips überwachen	148
Hack: Sichere Garage mit iButtons	149
Hack: Hoftür mit Panikschloss und iButtons	155
Hack: Bewegung melden und Video-Überwachung	161
Was hat sich verändert ?	164
Einbindung in FHEM	166
Hack: Feuchtigkeit melden	168
Hack: Gas melden	170
Hack: Rauch und Feuer melden.	172
Hack: Warnungs- und Alarmsignalisierung.	174
Hack: Alarmanlage mit FHEM	175
Sensoren der Alarmanlage.	178
Aktoren der Alarmanlage	180
7 Hacks zur Fernbedienung und Fernanzeige	183
Hack: Universalfernbedienung im SmartHome.	184
Hack: Infrarotfernbedienung mit dem Computer	187
Hack: Anwesenheitserkennung via Smartphone	191
Geozone rund um das SmartHome.	191
Anwesenheitserkennung im Haus – auch ohne Smartphone?	194
Hack: Textanzeige auf einem LCD	196
Hack: Digitaler Bilderrahmen als Anzeige	199
Hack: SmartHome-Daten in Bilder umwandeln	202
Daten – woher?	206
Hack: Tablet als Anzeige und Bediengerät.	207
Anzeige von Bildern.	209
Steuerung über das Tablet.	212
Weitere Tablet-Funktionen steuern	214
Hack: Eigene Widgets programmieren	215
Lineare Säule als Widget	216
Thermometer als Widget	219

8	Hacks für Musik und Medien	221
	Hack: Sprachausgabe im SmartHome	222
	Sprache auf einem Raspberry Pi ausgeben	225
	Hack: Spracherkennung im SmartHome	227
	Hack: Multiroom-Audiosystem	230
	Musik komprimieren – oder nicht?	231
	Hack: Smart-TV im SmartHome	234
	Eigene Apps für den Smart-TV	235
	Hack: Smart-TV selbst bauen	238
9	Hacks für Kalender und Zeiten	241
	Hack: An wiederkehrende Aufgaben erinnern	241
	Erzeugung von Kalenderdateien	242
	Auswertung von Kalenderdateien	244
	Hack: Feiertage automatisch erkennen	246
	Hack: Smarter Wecker mit FHEM	248
10	Hacks für das Wetter	255
	Hack: Daten der Wettervorhersage holen	256
	Hack: Stundengenaue Wetterprognose	263
	Hack: Eigene Wetterdaten im Netz zur Verfügung stellen	265
11	Hacks für Pflanzen und Tiere	267
	Hack: Bodenfeuchte messen	267
	Hack: Bewässerungsanlage steuern	270
	Sicherheit der Bewässerung	272
	Hack: Die Katze automatisieren	274
	Hack: Den Hund automatisieren	276
	Hack: Das Aquarium automatisieren	277
12	Systemkritik	281
	Hack: Funksysteme für das SmartHome	281
	InterTechno	283
	FS20	283
	Z-Wave	284
	EnOcean	286
	ZigBee	287
	Weitere Netze	288
	Hack: Ankopplung von Funksystemen	289
	Transceiver für Funksysteme	289
	Sende- und Empfangstechnik für Funksysteme	291
	Antennen für das SmartHome optimieren	293
	Hack: HomeMatic-System	295
	Peering und Pairing	296
	Protokoll	298
	HomeMatic-Komponenten	299

Hack: Kabelgebundene Systeme für das SmartHome	300
KNX	301
Bus-Systeme	302
Hack: 1-Wire-System	304
1-Wire-Busmaster	305
1-Wire-Verkabelung	309
1-Wire-Komponenten	313
13 Smarte Server für das SmartHome	317
Hack: FHEM auf FritzBox und NAS	317
FHEM auf der FritzBox	318
FHEM auf der NAS	318
Hack: Raspberry Pi und Co	319
Raspberry Pi	320
Weitere Mikrocomputer	323
Hack: Speicherkartenzugriffe optimieren	323
Hack: USB-Ports unter Linux fest zuordnen	324
Hack: Watchdog-Timer nutzen	326
Weitere Konfigurationsmöglichkeiten des Watchdog-Timers	327
Index	329

READ ME FIRST !

1

Derzeit kommt fast wöchentlich ein neues System für das SmartHome auf den Markt, und nicht einmal Experten können den Überblick über diesen Zoo behalten. Die Namen der verwendeten Protokolle sind ebenso klangvoll wie undurchsichtig: *HomeMatic IP*, *Z-Wave*, *EnOcean*, *Bluetooth Low Energy*. Dem Einsteiger stellen sich deshalb viele Fragen: Was ist zukunftssicher, was ist preiswert, was sieht gut aus?

Die Antworten sind natürlich in hohem Maße davon abhängig, was man erreichen *möchte*. Was man erreichen *kann*, habe ich in diesem Buch mit einer Vielzahl von Beispielen dokumentiert. Einsteiger werden sich anhand dieser Beispiele gut über die Möglichkeiten informieren und (vielleicht) für ein System entscheiden können.

Für die eigene Umsetzung der meisten Beispiele benötigt man dann schon ein gewisses Verständnis von Programmierung oder Elektronik – oder sogar beides. Auch diejenigen, die schon ein SmartHome-System nutzen, sollen Anregungen bekommen, etwa durch

- Ideen zur Programmierung,
- neue Sensordaten und Messgrößen und
- neue Steuermöglichkeiten und Aktoren für ihre Installation.

Eben, wie der Titel schon sagt, *Hacks*. Beispielsweise kann man einen hier im Buch vorgestellten Wasserdetektor an den Schalt- oder Tasteneingang eines beliebigen SmartHome-Systems anschließen; und ein Interface für die Steuerung einer Warmwasser-Zirkulationspumpe kann durch den Schaltausgang eines beliebigen SmartHome-Systems bedient werden.

Da kein SmartHome-System alle Möglichkeiten abdecken kann, bilden genau solche Ergänzungen den Unterschied zwischen einem zusammengekauften Satz von Fertiggeräten und dem echten »SmartHome«. Diese Anbindung verschiedenster Sensoren, Aktoren und Geräte an eine gemeinsame Logik macht das Home aber nicht nur smart – sondern kann ein Haus zu einem Bestandteil des *Internet of Things (IOT)*, des weltweiten Internet der Dinge machen. Das mag Geschmackssache sein – aber ich finde es cool, wenn ich von New York aus meine Gartenbewässerung anstellen kann, weil daheim gerade eine Hitzewelle herrscht.

Eine solche »digitale Ökologie« habe ich schon vor 15 Jahren in meinem »Taschenbuch Multimedia« vorhergesagt: Das IOT wird die fünfte Generation (5G) in der Digitalisierung der Welt sein. Netze der vierten Generation stellen uns heute schon größenordnungsmäßig *eine Milliarde Knoten* bereit: 2015 waren 1 Milliarde Computer im Domain Name Service (DNS) registriert, und 2014 gab es rund 2,6 Milliarden Nutzer fester Internetzugänge sowie 1,25 Milliarden verkaufte Smartphones. Ein 5G-Netz aber bedeutet, dass wir es vielleicht bald mit *500 Milliarden* digitalen Endgeräten zu tun bekommen, die miteinander kommunizieren können. Das ist die Zukunft.

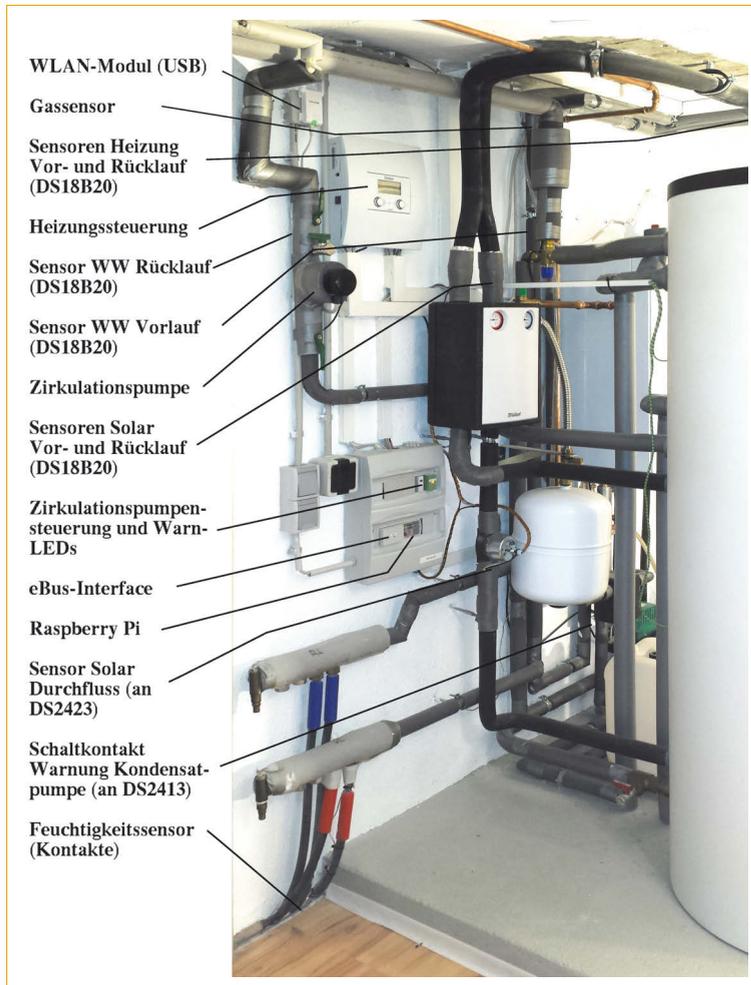
Ein Beispiel

Schauen wir uns doch einfach mal an, was man so alles machen kann. Abbildung 1–1 zeigt eine Heizungsanlage mit moderner Gas-therme und solarer Unterstützung für die Warmwasserbereitung.

Welches System?

Die oben gestellten Fragen nach Zukunftssicherheit, Preiswürdigkeit und gutem Design werde ich in diesem Buch (fast) nicht beantworten. Die Antworten wären erstens zu schnell veraltet, und zweitens mit Sicherheit nicht für jeden Anwender in gleichem Maße sinnvoll. In Kapitel 12 gebe ich deshalb eine Kurzübersicht über einige wichtige SmartHome-Systeme.

In den ersten elf Kapiteln ist es unser Ziel, solche Systeme zu erweitern, sie miteinander und mit der nichtdigitalen Umwelt zu verknüpfen. Eine solche Verknüpfung erfordert immer, die Systeme in eine gemeinsame Anwendungslogik einzubinden, denn es macht wenig Sinn, wenn der Bewegungsmelder der Firma X nicht mit Schaltbedingungen an die Leuchte des Herstellers Y angepasst werden kann.



◀ **Abbildung 1–1**

SmartHome im Keller – mit Raspberry Pi, diversen Temperatursensoren (1-Wire DS18B20), einer Steuerung der Zirkulationspumpe und einem eBus-Interface zur Heizung (siehe Kapitel 5) und Sicherheitssensoren für Gas, Wasser und den Zustand der Kondensatpumpe (siehe Kapitel 6). Abbildung 5–14 zeigt noch eine schematische Sicht auf diese Anlage.

Von besonderer Bedeutung ist deshalb Software, die sich solche übergreifende Funktionalität auf die Fahnen geschrieben hat. Diese Software werden wir im Folgenden als SmartHome-Server bezeichnen, oder als SmartHome-Zentrale, wenn damit auch der Computer gemeint ist, auf dem sie läuft. In Kapitel 13 stelle ich verschiedene Computer für diesen Zweck vor.

In Tabelle 1–1 nenne ich fünf SmartHome-Server, ohne dass diese Liste Anspruch auf Vollständigkeit erhebt: *FHEM*, *OpenHAB* und *Domoticz* als Open Source-Software, *IP Symcon* und *Loxone* als kommerzielle Systeme.

Tabelle 1–1 ▶
SmartHome-Systemsoftware
im Vergleich

System	Lizenz/Sprache	URL	Bemerkungen
FHEM	Open Source-Server; geschrieben in Perl, leicht erweiterbar; deutsches System	http://www.fhem.de	Universell, Interfaces sind für nahezu alle gängigen SmartHome-Systeme vorhanden. Mit 20.000 Teilnehmern sehr große deutschsprachige Nutzercommunity. Tiefe Eingriffsmöglichkeiten für Benutzer.
OpenHAB	Open Source-Server; geschrieben in Java, Erweiterung eher komplex	https://www.openhab.org	Universell, Interfaces sind für nahezu alle gängigen SmartHome-Systeme vorhanden. Kaum Konfigurationsmöglichkeit auf der Weboberfläche, aber schnelle Resultate.
Domoticz	Open Source-Server; geschrieben in C++, Erweiterung in Blockly, Lua, Python, Bash	http://www.domoticz.org	Universell, Interfaces sind für einige SmartHome-Systeme vorhanden. Hat Potenzial, ist aber noch nicht sehr weit entwickelt.
IP Symcon	Kommerziell, Basisversion ca. 100 €, Vollversion ca. 500 €. Mit dezidiertem Serverhardware 250–750 € Deutsches System	https://www.symcon.de/	Viele Interfaces für SmartHome-Systeme.
Loxone	Kommerziell, sog. Miniserver, ca. 400 €, Erweiterungen jeweils ca. 100–500 €	http://www.loxone.com	Hardware und Software = komplette SmartHome-Zentrale. Viele Erweiterungen für SmartHome-Systeme, jeweils spezielle Hardware

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.

Natürlich will ich nicht verschweigen, dass ich selbst eine SmartHome-Systemsoftware bevorzuge – die Open Source-Software FHEM. Diese Bevorzugung hat einen ganz einfachen Grund: FHEM kann durch eine Vielzahl von Modulen und Interfaces mit nahezu allen anderen SmartHome-Systemen interagieren. Es ist einfach programmierbar, und eine im Wesentlichen deutsche Nutzercommunity von etwa 20.000 Usern entwickelt das System immer weiter. Die herausragend leichte Erweiterbarkeit von FHEM erlaubt auch das schnelle Ausprobieren neuer Lösungen ohne langwierige

Entwicklungen. Wenn ich also in diesem Buch von der allgemeinen Abdeckung beliebiger SmartHome-Server abweiche – dann werden wir die Einbindung der vorgetragenen Ideen in FHEM diskutieren.

Aus der Beschreibung von FHEM:

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.

(...)

Das Programm läuft als Server, man kann es über Webbrowser, dedizierte Smartphone Apps oder telnet bedienen, TCP Schnittstellen für JSON und XML existieren ebenfalls.

(...)

Um es zu verwenden benötigt man einen 24/7 Rechner (NAS, Raspberry Pi, PC, MacMini, etc) mit einem Perl Interpreter und angeschlossene Hardware-Komponenten wie CUL, FHZ1300PC etc. für einen Zugang zu den Aktoren und Sensoren.

FHEM verdanken wir 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. Dieses Buch ist aber auch keine Anleitung für FHEM, denn dafür gibt es eine sehr gute Einsteigerdokumentation unter <http://fhem.de/Heimautomatisierung-mit-fhem.pdf>.

Warnung vor Gefahren

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«, mehr dazu in Kapitel 6), deren Beachtung unter Umständen nötig ist, um den Versicherungsschutz für Objekte zu erhalten.

Das bedeutet, 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!

Die Realität sieht natürlich ein wenig anders aus. Einerseits gibt es heute elektrische Anlagen auch im Niedervoltbereich, mit denen man Leib und Leben gefährden kann. Beispielsweise ist die Energiedichte eines normalen Laptop- oder Smartphone-Akkus so groß, dass eine 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 XY verkauft werden, von eingetragenen Elektroinstallateuren eingebaut und angeschlossen wird. Pikanterweise beteiligen sich die Energieversorgungsunternehmen selbst an diesem verlogenen Spiel. Kauft man beispielsweise einen Funkschalter aus der Serie *RWE SmartHome*, so wird eine ausführliche Einbauanleitung für Laien mitgeliefert – die natürlich aber die oben genannte Warnung enthält.

Mit anderen Worten: Gegen diese gesetzlichen Regelungen wird laufend verstoßen, wir haben also unter unseren Mitbürgern im Baumarkt offenbar eine große Zahl von nicht ganz gesetzestreuem 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 abgefackelt hat, hinterher dafür Geld von der Versicherung bekommt.

Darum folgt jetzt der zweite wichtige Hinweis zu diesem Buch: Es enthält neben nützlichen Tipps auch echte *Hacks*, also Anleitungen zur Veränderung technischer Anlagen ohne entsprechende Autorisierung. Führen Sie diese nur aus, wenn Sie bereit sind, den Pfad der Tugend zumindest ein klein wenig zu verlassen. Und bedenken Sie:

Was immer Sie tun: Sie tragen die Verantwortung dafür.

Ach ja: Da vielfach heute von politischen Kräften der Schutz vor allen Risiken gepredigt wird, könnte eine extrem wohlmeinende Bundesregierung (oder EU-Kommission) schon bald auf den Gedanken kommen, den Verkauf von Lichtschaltern in Baumärkten zu verbieten.

Dann werde ich auswandern.

Anleitung für Anarchisten

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 immer mit geeigneten Werkzeugen, beispielsweise mit Schraubenziehern, die eine isolierte Klinge haben.

Tipp

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

Vom WAF

Natürlich wendet sich dieses Buch an Frauen und Männer. Nur wenig würde mich mehr freuen, als zu hören, dass eine Frau nach

dem Lesen dieses Buches zum Schraubenzieher greift und ihre Lichtschalter gegen Funkschalter austauscht. Auch will ich keinesfalls einem altmodischen Rollenverständnis das Wort reden. Aber nach mehr als 40 Jahren Tätigkeit in der Erwachsenenbildung und 35 Jahren glücklicher Ehe kann ich eines ganz gewiss schreiben: Frauen und Männer sind unterschiedlich, und nur wenige Dinge lassen diese Unterschiede so zum Vorschein kommen wie das Thema SmartHome.

- ER wird sich freuen, wenn der neue SmartHome-Server in einem selbst gebauten Gehäuse mit blinkenden LEDs und vielen Kabeln im Bücherregal vor sich hin schnurrt. SIE wird das nur in Ausnahmefällen ertragen.
- SIE wird die Wände hoch gehen, wenn die selbst installierte Funkalarmanlage sich nicht auf einfachste Weise ausschalten lässt. ER wird hingegen das Gehäuse öffnen und die richtigen Kontakte überbrücken.

Die Bevorzugung von Technik, die erstens unsichtbar ist oder mindestens gut aussieht und zweitens ohne spezielle Eingriffe funktioniert, nennen wir aus historischen Gründen den *Woman Acceptance Factor* oder WAF. Deshalb eine dritte Warnung:

Was immer Sie tun: Behalten Sie dabei den Komfort und die Bedenken Ihrer Partnerin oder Ihres Partners im Auge.

- 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 Selbstbaulösung abgestürzt ist.
- Dokumentieren Sie, was Sie gebaut und geändert haben. Ihre Partner und Ihre Erben werden Sie dafür preisen.

Wolkig mit Aussicht auf Verwirrung

Die größte mediale Präsenz des Themas SmartHome ergibt sich derzeit dadurch, dass eine Vielzahl von Geräten auf den Markt geworfen werden, die durch eine »App« steuerbar sind. Entsprechende RGB-LED-Strips oder Steckdosen sind in fast jedem Baumarkt zu finden. Der technische Hintergrund ist immer, dass sich diese Geräte mit eingebauten WLAN-Modulen ins heimatische WLAN einbinden können. Diese Module sind inzwischen zu Preisen von ca. 2 € zu bekommen, ein wenig Elektronik dahinter für weitere 3 € – und fertig ist ein Bestandteil des *Internet of Things*. Das wäre prima, wenn es denn in irgendeiner Weise standardisiert wäre.

Ein solcher Standard existiert aber nicht, stattdessen lässt nahezu jeder Hersteller auf möglichst billige Art seine eigene »App« programmieren, die genau diese Komponente steuern kann. »Billig« ist dabei wörtlich zu nehmen – denn zur App-Programmierung braucht man fast keine Kenntnisse der Informatik. Schüler der zehnten Klasse können nach ein paar Stunden Einführung genau solche Steuerungssoftware zusammenklicken. Ich rate deshalb, sich vor dem Kauf dieser Art von Geräten genau zu überlegen, wie viele verschiedene Apps zur Steuerung seines SmartHome man auf dem Smartphone installieren möchte. Eine beliebte Frage in den Nutzerforen der SmartHome-Systeme ist deshalb:

»Ich habe da im Baumarkt so ein paar billige Funksteckdosen erstanden, wer schreibt mir bitte eine Anleitung, wie ich diese einbinden kann?«

Abgesehen von der Vielfalt dieser Einfalt gibt es aber noch ein viel gravierenderes Problem, das mit der Nutzung der *Cloud* durch diese Hersteller zusammenhängt. Denn viele von ihnen werben explizit mit der Möglichkeit, die Geräte auch von außerhalb des Hauses zu steuern. Das verlangt natürlich 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. Das ist gefährlich – und datenschutzrechtlich sehr bedenklich.

Ich bin beileibe kein Gegner des Cloud-Computing. Allerdings wird das im Moment weit übertrieben. Die Zukunft darf *nicht* so aussehen, dass kommerzielle Anbieter mit ihren Servern das *Internet of Things* kontrollieren – und auch nicht so, dass die Logik meines SmartHome davon abhängt, ob ein weit entfernter Dienst wie IFTTT (If-This-Then-That) meine Skripte ausführt. Bevor man sein SmartHome für die Cloud öffnet, sollte man sich deshalb sehr genau über diese Risiken informieren.

Technische Details

Was also benötigen Sie?

1. Manche der hier im Buch vorgestellten Hacks lassen sich bereits mit einigen wenigen Einzelkomponenten oder einer einfachen SmartHome-Zentrale eines beliebigen Systems verwirklichen. Gemeint sind etwa solche Aufgaben wie »wenn dies, dann jenes« – dafür benötigen Sie eben Sensoren, Aktoren, gegebenenfalls eine Zentrale und die Bedienungsanleitung.





2. Bei der nächsten Stufe handelt es sich um Hacks, die mithilfe eines Programms ablaufen. Dabei werden schon einmal Berechnungen ausgeführt – etwa Variablen miteinander multipliziert oder, ganz gewagt, ein Logarithmus berechnet.

Dafür benötigt man eine SmartHome-Zentrale, die in einer Programmiersprache programmierbar ist. Viele der modernen kommerziellen SmartHome-Systeme bieten dies, etwa durch eine eigene Skriptsprache für das HomeMatic-System (homeputer CL Studio) oder durch die Skriptsprache Lua in dem Fibaro Z-Wave System. Auch FHEM ist natürlich beliebig programmierbar, neben der echten Programmiersprache Perl verfügt es auch noch eine eigene Skriptsprache.

Codebeispiele

Die meisten Programmiersprachen sind heute sehr gut online erlernbar. Die besten Anleitungen für HTML, JavaScript und Perl findet man bei SelfHtml, z.B. <https://wiki.selfhtml.org/wiki/Perl>. Und wer mehr als diese lernen möchte, dem seien das *Handbuch Programmiersprachen* und das *Taschenbuch Programmiersprachen* empfohlen, die ich vor einigen Jahren zusammen mit Holger Vogelsang herausgegeben habe.

Bei Herunterladen von Programmen beachten Sie bitte die Lizenzbedingungen. Insbesondere die freien GNU-Lizenzen erfordern, dass die Namen der Autoren und die Lizenzinformationen in den Dateien erhalten bleiben – das ist nur fair, denn die Autoren haben da meist viel Arbeit hineingesteckt.

Um der Vielfalt der Programmiermöglichkeiten keinen Abbruch zu tun, werden deshalb die Codes in diesem Buch oft als so genannter »Pseudocode« geschrieben – mit einer ganz einfachen Syntax, die schnell in jede konkrete Programmiersprache übersetzt werden kann.



3. Des Weiteren gibt es natürlich Hardware-Hacks. Kenntnisse in Elektronik, Lötkenntnisse und Experimentierfreude sind für diese Hacks eine wichtige Voraussetzung – und natürlich das intensive Lesen der Warnhinweise.



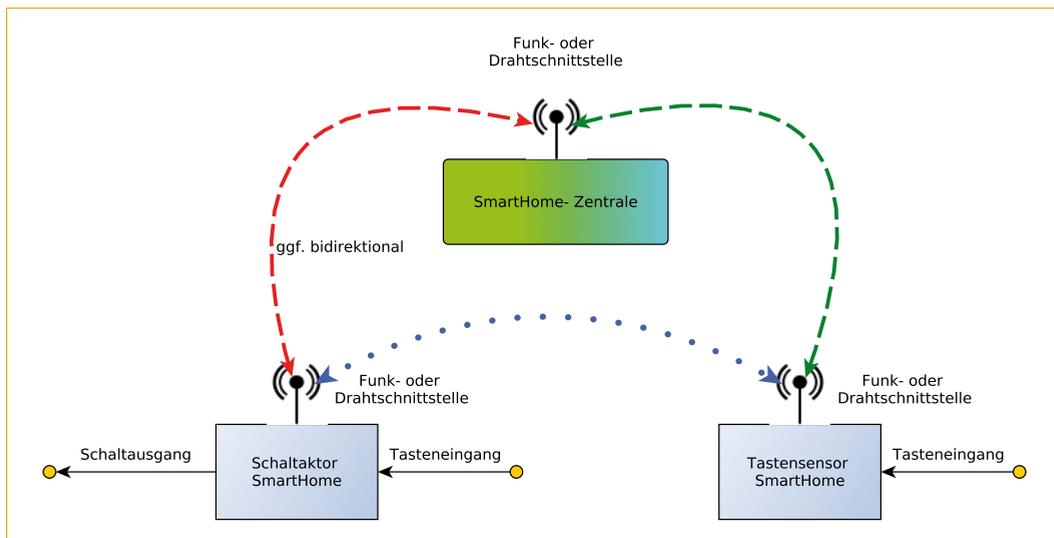
4. Als Krönung stelle ich auch ein paar Hacks vor, die auf einer separaten Plattform laufen – einem Arduino-Mikrocontroller. Das kombiniert dann Software und Hardware, aber verspricht natürlich erheblichen Gewinn, wenn es dann tatsächlich einmal »läuft«.

Der Einfachheit halber werden die Symbole, die ich den obigen Absätzen vorangestellt habe, auch im Buch verwendet. Sie können deshalb auf den ersten Blick erkennen, was der jeweils folgende Abschnitt Ihnen abverlangt.

Sensoren und Aktoren

In den folgenden Kapiteln werden wir häufig Lösungen sehen, die nicht an ein bestimmtes SmartHome-System gebunden sind, sondern an eine Vielzahl solcher Systeme angekoppelt werden können. In vielen Fällen geschieht diese Ankopplung über einen SmartHome-Schaltaktor oder einen SmartHome-Tastensensor – dafür kann man eine Vielzahl verschiedener Modelle und Systeme verwenden.

Dabei gehen wir davon aus, dass der Schaltausgang des Aktors nur eine niedrige Spannung schalten muss, also nicht etwa Netzspannung. Der Tasteneingang des Aktors soll – wenn er auf Massepotenzial = GND gesetzt wird, den Schaltausgang umschalten (»toggeln«). Der Tasteneingang des Sensors soll, wenn er auf Massepotenzial gesetzt wird, eine entsprechende Meldung an das SmartHome-System auslösen. Das kann entweder im Push-Modus geschehen, also ein aktives Signal auslösen, oder im Sensor selber eine Zwischenspeicherung erzeugen, die vom SmartHome-System aus abgefragt werden kann (Pull-Modus). Typische Realisierungen für solche Aktoren und Sensoren existieren in vielen SmartHome-Systemen. In Kapitel 12 werden deshalb die möglichen Realisierungen aufgeführt.



Ein weiterer Begriff muss an dieser Stelle eingeführt werden, denn wir müssen ja irgendwie benennen, was der Sensor nun eigentlich macht: Er löst in der SmartHome-Zentrale ein *Event* aus, ein Ereignis also. Für einen einfachen Tastensensor kann dies ein *toggle-Event* sein, also ein Umschaltereignis. Hat der Tastensensor zwei

▲ **Abbildung 1–2**
Ein generischer Schaltaktor und ein generischer Tastensensor interagieren mit der SmartHome-Zentrale (rot, grün) oder direkt miteinander (blau).

Stellungen, könnte es sich um ein *on-Event* oder ein *off-Event* handeln, je nachdem, in welche Richtung die Taste betätigt wird.

In der SmartHome-Zentrale besteht in der Regel die Möglichkeit, auf die von einem bestimmten Gerät (Device) ausgelösten Events zu lauschen und dann je nach Art des Events bestimmte Ereignisse auszulösen. Dafür gibt es dann einen Programmbestandteil, den wir als *Event-Handler* bezeichnen wollen.

Einheiten und Maße

Hexadezimale Größen werden im Text durch ein vorangestelltes 0x gekennzeichnet.

Temperaturen messen wir in Grad Celsius, abgekürzt °C. Temperaturdifferenzen werden allerdings in K = Kelvin (nicht »Grad Kelvin«) gemessen – derselbe Zahlenwert, aber eine andere Einheit. In Kelvin wird auch die absolute Temperatur gemessen, die wir aus dem Messwert in °C erhalten, indem 273,15 hinzugezählt werden. Anders ausgedrückt: Der absolute Nullpunkt der Temperatur (tiefer geht es nicht mehr...) liegt bei -273,15°C. Wir werden im Folgenden um einige Formeln nicht herumkommen. In ihnen verwenden wir für die absolute Temperatur in K die Abkürzung T und für die Temperatur in °C den griechischen Buchstaben θ (theta).

Links und Ergänzungen

Um die Nutzung des Buches zu vereinfachen, sind auf der Seite <http://lice-karlsruhe.de/smarthome-hacks-links-und-ergaenzungen/> alle Links zum direkten Anklicken aufgeführt. Dort finden Sie auch alle längeren Quelltexte aus dem Buch, soweit dies die jeweiligen Urheberrechte gestatten.

Hacks zur Lichtsteuerung

2

SmartHome bedeutete für unsere Großeltern, dass man durch Drehen oder Drücken eines kleinen Knopfes Licht in das Dunkel der Nacht bringen konnte. Für die meisten Menschen, die dem Thema SmartHome verfallen, ist dies tatsächlich auch heute noch der erste Anwendungsfall (oder »Use Case«, wie man in der Informatik sagt). Nur geht es nicht mehr darum, das *irgendwie* und *irgendwo* zu tun, sondern wir beherrschen das Licht inzwischen vollständig. Wir wollen bestimmen, von wo aus, mit welcher Helligkeit und Farbe, zu welchen Zeiten und unter welchen Bedingungen das Licht eingeschaltet wird.



◀ **Abbildung 2–1**

Hardware für die einfache Lichtsteuerung. Links oben ein HomeMatic-Systemschaltaktor (in verschiedene Schalterprogramme integrierbar), rechts oben eine HomeMatic-Fernbedienung. Links unten ein Unterputz-Schaltaktor für EnOcean, in der Mitte unten ein batterieloser Aufputz-Wandschalter von EnOcean und rechts unten eine schaltbare Zwischensteckdose des Z-Wave-Systems.

Hack: Lichtschalter dorthin verlegen, wo man sie braucht



Unsere Wohnung verändert sich mit uns:

- Ein neuer größerer Fernseher kommt ins Haus, und wir müssen umdekoriieren – und plötzlich können wir das Deckenlicht nicht mehr komfortabel vom Sessel aus dimmen.
- Ein Umbau – und plötzlich haben wir den Lichtschalter für die Gästetoilette auf der falschen Seite.
- Ein Pflegefall in der Familie – und wir möchten es der gepflegten Person ermöglichen, das Licht vom Bett aus zu schalten.

Kurz, wir brauchen einen Lichtschalter an einer anderen Stelle und möchten nicht, dass dafür Wände aufgestemmt werden – ein Funkschaltssystem muss her. Für diesen Anwendungsfall besteht das Funkschaltssystem mindestens aus zwei Komponenten:

- erstens aus einem Funksender. Dieser spielt die Rolle eines Sensors, denn er reagiert auf ein externes Ereignis – auf das Drücken einer Taste.
- zweitens aus einem Funkempfänger als Aktor, der die Leuchte letzten Endes anschaltet. In bidirektionalen Systemen hat der Aktor auch einen Sender, mit dem er dem Sensor Rückmeldung über die erfolgreiche Schaltaktion gibt, und dementsprechend hat der Sensor einen Empfänger, mit dem er diese Rückmeldung empfängt.

Ein solches Funkschaltssystem kann man auch einsetzen, wenn man gar keinen Eingriff in das Stromnetz vornehmen will: mit einem batteriebetriebenen Funksender und einem Aktor in Form eines schaltbaren Zwischensteckers. Diesen allereinfachsten Fall wollen wir aber im Folgenden nicht betrachten Stattdessen gehen wir davon aus, dass die Leuchte fest installiert ist und der bisherige Schalter in einer Anschlussdose in der Wand sitzt.

Im nächsten Abschnitt sollen alle relevanten Details eines solchen Schalteraustausches diskutiert werden. Das bedeutet: ganz viel Für und Wider – und eigentlich ein komplizierter Entscheidungsprozess, den man einmal genau durchlaufen sollte.

Für einen ersten Überblick kann man aber die nachfolgenden Ausführungen überspringen und direkt bei der Realisierung anfangen, also im nächsten Abschnitt.

Detaillierte Planung

Für den in der Wand befindlichen Schalter gibt es schon mal verschiedene Varianten:

- einfache Ein-/Ausschalter – diese schließen oder trennen einfach nur einen Kontakt.
- Wechselschaltungen, mit denen eine Leuchte von zwei Orten geschaltet werden kann. Dafür benötigt man einpolige Umschalter.
- Kreuzschaltungen, mit denen eine Leuchte von mehr als zwei Orten geschaltet werden kann. Dafür werden zweipolige Umschalter benötigt.

Die meisten Schaltaktoren für SmartHome-Systeme sind als einpolige Umschalter ausgeführt, können also in den beiden erstgenannten Fällen eingesetzt werden. Bei Kreuzschaltungen wird es deutlich schwieriger, einfach nur einen Schalter zu ersetzen. Hier ist dann zu empfehlen, alle Schalter durch Tastensensoren zu ersetzen und einen gemeinsamen Aktor anzusteuern.

Für die weitere Planung gehen wir davon aus, dass zunächst ein einzelner Schalter ausgetauscht werden soll. Dann sind verschiedene Entscheidungen zu treffen, die mit der Verfügbarkeit und Anordnung von Schaltdosen zu tun haben.

Zuerst muss man sich darüber klarwerden, ob der neue Lichtschalter (hier ist die Betätigungseinheit gemeint, also der Sensor) in eine vorhandene oder in eine neu zu schaffende *andere* Anschlussdose eingesetzt werden soll:

- Wenn nicht, muss der Sender entweder eine batteriebetriebene Funkfernbedienung sein, eine Smartphone-App oder ein batteriebetriebener Aufputz-(AP-)Funksender.
- Wenn ja, muss geklärt werden, ob die andere Anschlussdose einen Netzanschluss hat:
 - Wenn ja, kann er eine netzgebundene Stromversorgung haben.
 - Wenn nicht, muss der Funksender ein Unterputz-(UP-)Funksender mit Batteriebetrieb sein.

Die zweite wichtige Frage ist, ob der alte Schalter seine Funktion behalten soll:

- Wenn nicht, kann man ihn durch eine Blindplatte ersetzen und stattdessen dahinter den reinen UP-Schaltaktor einbauen. Allerdings muss noch geklärt werden, ob es sich um ein Zweileiter- oder um ein Dreileitersystem handelt (siehe unten).
- Wenn ja, muss geklärt werden, ob in dieser Schaltdose eventuell sogar neue Funktionen hinzukommen sollen, die andere Geräte steuern. Das ist deshalb möglicherweise sinnvoll, weil

Schalterdosen rar sind – und nicht jeder möchte dauernd zur Fernsteuerung sein Smartphone zücken müssen.

- Ist das nicht der Fall, kann der gesamte alte Schalter durch einen System-Schaltaktor ersetzt werden. Dieser kann dann sowohl vor Ort durch einen Tastendruck als auch aus der Entfernung durch den Funksender betätigt werden. Solche System-Schaltaktoren finden sich in fast allen SmartHome-Systemen, und zwar so, dass sie durch einfache Adapter auch in verschiedene Standard-Schalterprogramme von Markenherstellern integriert werden können. Voraussetzung ist meistens ein Dreileitersystem.
- Will man noch andere Geräte aus dieser Schalterdose steuern, kann man z.B. den alten Schalter durch einen Doppeltaster ersetzen und hinter diesen in der Dose, die man dazu möglicherweise vertiefen muss, einen UP-Funksender einbauen. Klar, dass für den Schaltaktor dann ein anderer Platz gefunden werden muss – etwa, indem er durch Umbau der anzusteuern Leuchte direkt in diese integriert wird. Auch hier ist in der Regel ein Dreileitersystem die Voraussetzung.

Die Vielzahl von Möglichkeiten für die Lichtschalterverlegung habe ich in Tabelle 2–1 zusammengefasst.

Was hat es jetzt mit dem Zweileitersystem auf sich? In älteren (oder sparsamen) Installationen sind Lichtschalter oft nur mit einem zweiadrigen Kabel als »Stichleitung« an die Anschlussdose angebunden: Phase hin, geschaltete Phase zurück, aber es fehlt der Nullleiter. Manchmal findet sich auch ein dreiadriges Kabel mit grünem Schutzleiter – aber ohne Nullleiter. Deshalb kann sich in diesen Fällen ein elektronisches Gerät (wie der einzubauende Schaltaktor) nicht direkt mit Netzspannung versorgen. Bei einem Dreileitersystem hingegen liegt in der Dose auch ein Nullleiter. Dann kann nahezu jeder Schaltaktor verwendet werden.

Es gibt zwar Funkaktoren, die ohne Nullleiter auskommen. Dabei handelt es sich aber in der Regel um Dimmaktoren, bei denen immer ein geringer Stromfluss durch die Leuchte stattfindet (irgendwoher muss die Energie für die Elektronik ja kommen). Entsprechende Modelle sind z. B. im Z-Wave- und im InterTechno-System erhältlich. Der Einbau eines solchen Aktors für ein Zweileitersystem kann allerdings nicht empfohlen werden, wenn man irgendwann auf LED-Beleuchtung umstellen möchte. Die in LEDs oft vorhandenen Schaltnetzteile reagieren auf einen kleinen Stromfluss bei manchen Ausführungen so, dass sie in regelmäßigen Abständen die LED aufblitzen lassen – und oft ist der vom Aktor »verlangte« minimale Stromfluss so groß, dass die LED gar nicht auszuschalten ist.

Alter Schalter	Neuer Schalter		
	Keine andere Dose	Andere Dose ohne Netz	Andere Dose mit Netz
Wird entfernt und durch Aktor ersetzt	Blindplatte mit Aktor + Aufputzfunksender (Batterie)	Blindplatte mit Aktor + Unterputzfunksender (Batterie)	Blindplatte mit Aktor + Unterputzfunksender (Netz)
Wird entfernt und überbrückt	Blindplatte + Aufputzfunksender (Batterie) + Aktor in Leuchte	Blindplatte + Unterputzfunksender (Batterie) + Aktor in Leuchte	Blindplatte + Unterputzfunksender (Netz) + Aktor in Leuchte
Bleibt, aber auch durch Funk steuerbar	Funksystemaktor + Aufputzfunksender (Batterie)	Funksystemaktor + Unterputzfunksender (Batterie)	Funksystemaktor + Unterputzfunksender (Netz)
Bleibt, zusätzlich andere Nutzung der Dose	Taster mit UP-Funksender + Aufputzfunksender (Batterie) + Aktor in Leuchte	Taster mit UP-Funksender + Unterputzfunksender (Batterie) + Aktor in Leuchte	Taster mit UP-Funksender + Unterputzfunksender (Netz) + Aktor in Leuchte

◀ Tabelle 2–1

Lichtschalter verlegen – aber wie?

Warnung

In jedem Fall gilt: Der Einbau von Aktoren in Schalterdosen und eventuell auch Leuchten erfordert bis auf den Fall des einschraubbaren Funkaktors die Mitwirkung des Elektroinstallateurs Ihres Vertrauens – das haben wir allerdings im ersten Kapitel ausführlich diskutiert. Seien Sie also auf der Hut, denn es besteht Lebensgefahr.

Die Alternative besteht für ein Zweileitersystem darin, den Schalter in der ursprünglichen Dose oder in der nächstgelegenen Wandanschlussdose komplett zu überbrücken. Damit liegen an der Leuchte alle drei Leiter an – und eventuell kann man den Funkaktor in die Leuchte selbst einbauen. Insbesondere im InterTechno-System gibt es preiswerte Funkaktoren, die einfach zwischen Leuchtmittel und Fassung eingeschraubt werden. Bei anderen Systemen wie FS20 oder HomeMatic benötigt man etwas mehr Platz in der Leuchte (Näheres zu diesen Funksystemen in Kapitel 12). Limitierender Faktor bei der Einbaulösung ist oft die Erwärmung der Leuchten, denn gerade durch Halogenleuchten wird der Sockel sehr warm.

Zwei weitere Systeme zum Schalten und Dimmen, aber auch zur Veränderung der Farbtemperatur, sind direkt in die Leuchtmittel integriert: *Philipps Hue* und *Osram Lightify* sind beides Varianten des ZigBee-Systems und somit eigentlich auch mit ganz vielen anderen SmartHome-Geräten vernetzbar. Faktisch ist das nach den Berichten, die man im Internet findet, nicht immer ganz einfach.

Normalerweise nutzt man deshalb die so genannten *Gateways* desselben Herstellers zur Verbindung mit dem WLAN und steuert die Leuchten über die mitgelieferte App. Das System von Philipps kann innerhalb des Hauses auch ohne Internetzugang betrieben werden, es benötigt nur bei Steuerung von außerhalb des Hauses einen Cloud-Zugang. Das System von Osram erfordert hingegen immer einen Internetzugang.

Warnung

Nicht nur auf der Hut sein, sondern insgesamt die Finger von der Installation in Schalterdosen sollten Sie lassen, wenn Ihnen die Begriffe Phase, Schutzleiter und Nullleiter nicht sehr genau klar sind. Das mit der Lebensgefahr ist kein Witz!

Tipp

Sie sollten hierzu unbedingt die allgemeinen Hinweise zur *Cloud* im ersten Kapitel sowie die Bemerkungen zum ZigBee-System in Kapitel 12 lesen.

Einfache Realisierung

Wichtig für die Realisierung sind die Warnungen im vorigen Abschnitt – die sollte man auf jeden Fall lesen (und beherzigen). Schauen wir uns noch einmal die drei ursprünglichen Use Cases an:

- Das Deckenlicht soll von einem anderen Ort aus gedimmt werden. Als einfachste Lösung bietet sich die in Tabelle 2–1 rot markierte Lösung an: Man ersetzt den bisherigen Lichtschalter durch einen Funksystemaktor zum Dimmen. Dieser wird dann per direkt angebrachter Taste ebenso gedimmt wie mit einer entsprechenden Funkverbindung.
- Der Lichtschalter für die Gästetoilette ist auf der falschen Seite. Als einfachste Lösung bietet sich die in Tabelle 2–1 grün markierte Lösung an: den alten Lichtschalter durch einen Unterputz-Funkaktor ersetzen, die Dose möglichst gut tarnen – und einen Aufputzfunksender auf die andere Türseite setzen, ohne die Wand aufzustemmen. Batterien der typischen Systeme halten 1–2 Jahre, und man kann den Batteriezustand im Funkinterface ablesen. Wer dies vermeiden möchte, verwendet das EnOcean-System, bei dem die Energie für das Funksignal durch den Tastendruck selbst gewonnen wird.
- Für die Lichtsteuerung durch eine pflegebedürftige Person hat es sich bewährt, einen einfachen Funk-Zwischenstecker (nahezu in jedem SmartHome-System verfügbar) durch einen in Griffweite des Patienten befindlichen Aufputz-Tastensensor zu steuern.

Tipp

Diesen Aufputz-Tastensensor sollte man gegen unbeabsichtigte Betätigung durch Pflegekräfte sichern – denn »leichte Bedienung« für die pflegebedürftige Person impliziert eine noch leichtere Bedienung durch die Pflegenden.

In diesem Hack habe ich mit Absicht weit ausgeholt, um den notwendigen Planungsprozess zu verdeutlichen. Nichts ist frustrierender als der Einstieg in die SmartHome-Welt durch eine nicht gut funktionierende Lösung. Die wichtigste Frage, die am Anfang gestellt werden muss, lautet also: Was möchte man erreichen? Erst danach kommt die Frage nach dem Wie, die Technik muss also immer der Funktion folgen, nicht umgekehrt.

Hack: Licht mit Bedingungen schalten



In diesem Hack soll nun die einfache Funktion »Licht ein- oder ausschalten« etwas aufgebohrt werden, indem wir das Licht nicht nur durch einen Schalter, sondern *auch* durch eine zentrale Steuerung schalten. Abbildung 1–1 in Kapitel 1 zeigt die verschiedenen Signalwege. Bisher hatten wir lediglich den blau markierten direkten Signalweg vom Tastensensor zum Schaltaktor realisiert. Jetzt soll also zusätzlich der rot markierte Signalweg von der Zentrale zum Schaltaktor verwendet werden. Aber wie geht das genau?

Soll die Zentrale das Licht auch dann ausschalten können, wenn es mit dem Tastensensor eingeschaltet wurde? Kann man Prioritäten setzen, wie etwa das Ausschalten durch den Tastensensor zu blockieren, wenn der gegenwärtige Zustand durch die Zentrale gesteuert wurde? Wie »weiß« die Zentrale darüber Bescheid, in welchem Zustand sich der Tastensensor und der Schaltaktor gerade befinden? Hierzu gibt es zwei Möglichkeiten.

Ein unidirektionales Funkprotokoll (z.B. InterTechno, FS20 oder verschiedene andere preiswerte Systeme) könnte verwendet werden, wenn eine visuelle Rückkoppelung zwischen Bediener und Leuchte besteht (mit anderen Worten: Man sieht ja wohl, ob das Licht auf der Kellertreppe an ist, oder nicht...). Dann kann die Zentrale natürlich über den Zustand des Schaltaktors nur informiert sein, wenn sie das Signal vom Tastensensor ebenfalls empfangen hat (grüner Signalweg in Abbildung 1–2).

Ein bidirektionales Funkprotokoll (z.B. bei HomeMatic, Z-Wave und vielen anderen modernen Systemen) ermöglicht dem Schaltaktor, die Zentrale auch dann über seinen Zustand zu informieren, wenn der eigentliche Schaltvorgang direkt vom Tastensensor ausgelöst wurde.

Nicht nur aus Gründen der Anwendungssicherheit kann es sinnvoll sein, alle Schaltaktionen durch die Zentrale ausführen zu lassen. Ein weiterer Grund dafür könnte, wie bereits oben erwähnt, eine Prioritätslogik der Schaltvorgänge sein. Diese soll durch einen so genannten Zustandsautomaten repräsentiert werden, also durch ein System mit mehreren inneren Zuständen und den Übergängen zwischen ihnen, die durch Events ausgelöst werden. Dabei gibt es zwei prinzipiell unterschiedliche Vorgehensweisen bei der Programmierung:

- Ansatz I: Die Zustände des Automaten können in so genannten globalen Variablen abgelegt werden, auf die andere Programme problemlos zugreifen können. Jedes Gerät, das diese Zustände ändert, bekommt ein eigenes kleines Programm, einen Event-Handler. Der Vorteil dieses Ansatzes ist die Übersichtlichkeit der Programme. Ein Nachteil könnte aber in entsprechend nebenläufigen Systemen sein, dass sich diese Programme gegenseitig unkalkulierbar beeinflussen.
- Ansatz II: Der Zustandsautomat wird von einem kompakten Programm gesteuert, das von jedem Event der beeinflussenden Geräte neu aufgerufen wird. Ein Nachteil dieses Ansatzes ist der längere Programmcode (oft sogar als Spaghetti-Code...), ein Vorteil aber der klare zeitliche Ablauf.

Tipp

Das gleichzeitige Empfangen eines unidirektionalen Funksignals durch zwei verschiedene Geräte ist sehr fehleranfällig. Sicherer in der Anwendung ist es, den Schaltaktor nur durch die Zentrale bedienen zu lassen. Allerdings kann dies durchaus zu merklichen Schaltverzögerungen führen.

Wir wollen den Ansatz I für Lichtschaltungen mit Bedingungen anhand von zwei Anwendungsfällen diskutieren: Zuerst soll eine einfache Prioritätslogik realisiert werden, bei der ein Tastensensor Vorrang vor einem anderen hat. Anschließend soll dies zur Lichtsteuerung an einer Kellertreppe erweitert werden. Der Ansatz II (kompaktes Programm) wird im *Hack: Zirkulationspumpe durch einen Zustandsautomaten steuern* in Kapitel 5 vorgestellt.

Lichtsteuerung priorisieren

Konkret könnte es sich bei unserer Prioritätslogik um die Beleuchtung eines Gartenhäuschens handeln – schaltet man die dortige Leuchte vom Hauptgebäude aus ein, soll sie nicht im Gartenhäuschen selbst ausschaltbar sein. *S1* und *S2* seien also Tastensensoren, aber selbstverständlich ließe sich jeder von ihnen auch durch ein auf der SmartHome-Zentrale ablaufendes Programm ersetzen. Interessant ist nur das Prinzip der Priorisierung: Wird die Leuchte durch den Tastensensor *S2* eingeschaltet, ist sie nicht durch *S1* auszuschalten, sondern nur wieder durch *S2*. Damit ist die direkte Ankopplung von *S1* an die Leuchte (der blaue Signalweg in Abbildung 1–2) ausgeschlossen. Denn natürlich wäre es unschön, erst auf die Veränderung des Leuchtenstatus durch einen Funkbefehl des Tastensensors zu warten, um diesen Funkbefehl dann vielleicht wieder rückgängig zu machen.

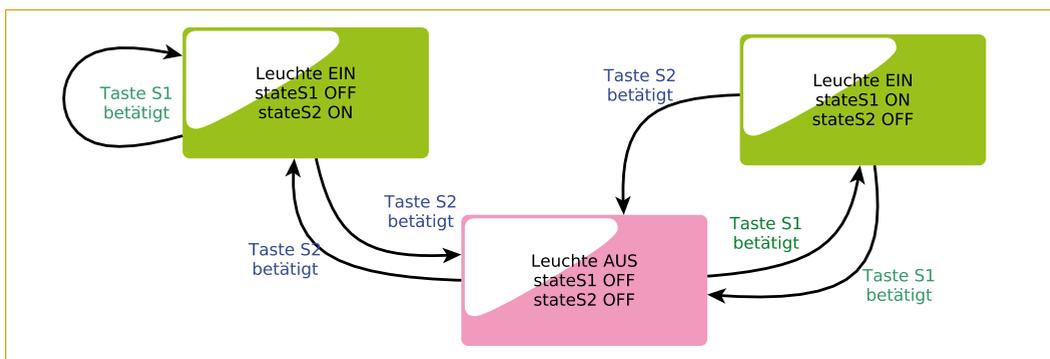


Abbildung 2-2 ▲ Zustandsdiagramm der einfachen Lichtsteuerung mit Priorität. Zustände mit grüner Grundfarbe stehen für eine eingeschaltete Leuchte, rot = aus. Deutlich sichtbar ist in diesem Diagramm die Asymmetrie: Die Schaltung durch den Tastensensor *S2* hat Vorrang vor Tastensensor *S1*.

Stattdessen müssen die Tastensensoren von der SmartHome-Zentrale überwacht werden. Wie im ersten Kapitel definiert, gehen wir davon aus, dass jeder Tastendruck dort *toggle-Events* auslöst, einfache »Wechsel«-Befehle also. Die SmartHome-Zentrale wertet diese mit einem so genannten Event-Handler aus. Das ist ein kleines Unterprogramm mit Zugriff auf den gegenwärtigen Zustand der Leuchte in den beiden globalen Variablen *stateS1* und *stateS2*.

Für solche und ähnliche Anwendungen empfiehlt es sich, ein Diagramm der möglichen Zustände und der Übergänge zwischen die-