

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

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

```
define autocreate autocreate
```

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

■ 3.1 SmartHome über LAN und WLAN

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

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

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



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

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

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

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

3.1.1 Shelly-Schaltaktoren und Dimmer

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

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

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

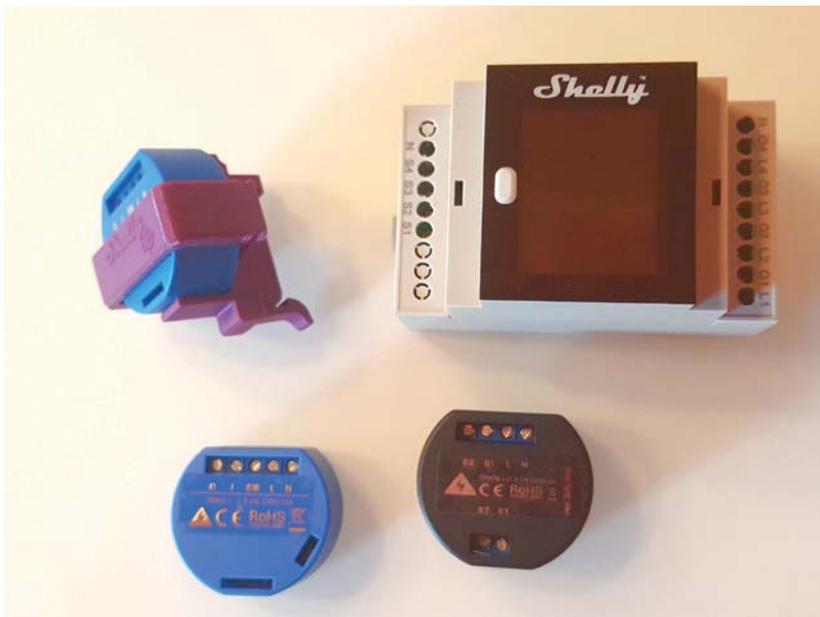


Bild 3.2 Schaltaktoren von Allterco Robotics (von links: Shelly 1 in einer Hutschienhalterung, Shelly 1, Shelly 4 Pro, Shelly 2)

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

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

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

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

3.1.1.1 Shelly 1 out of the box

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

Tabelle 3.1 Vergleich der verschiedenen Schaltaktoren von Allterco Robotics

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

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

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



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

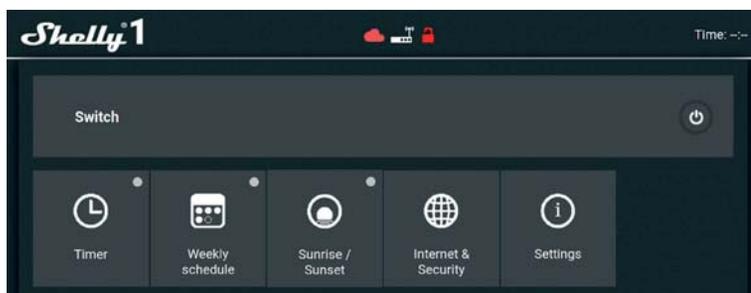


Bild 3.3 Access Point Mode eines Shelly 1: Das rote Cloud-Symbol zeigt, dass der Zugang zur Shelly Cloud nicht besteht, das rote Schloss-Symbol zeigt, dass der Zugang nicht durch ein Passwort gesichert ist.

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

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

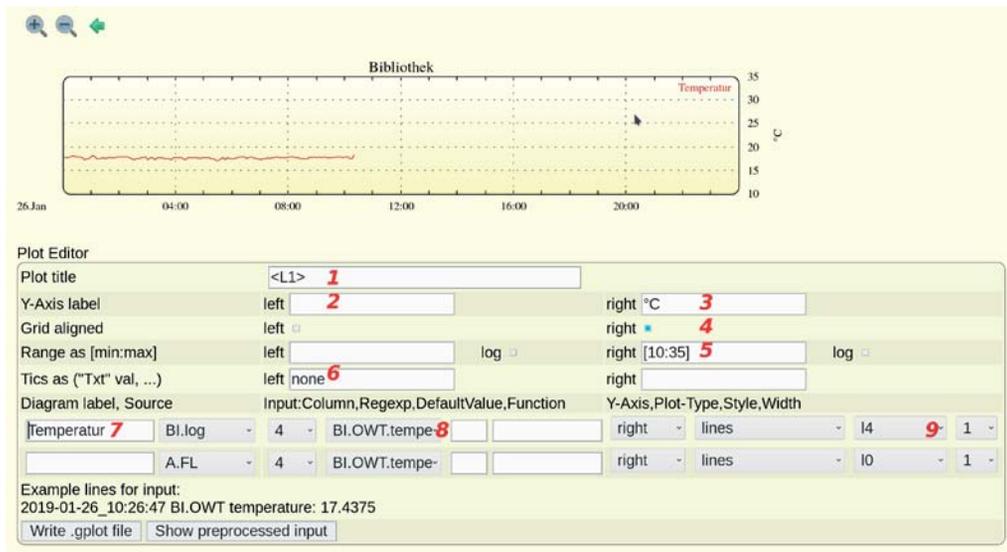


Bild 8.7 Beispielhafte Eingaben im Plot-Editor (Nummerierung siehe Text)

Klickt man anschließend auf WRITE .G PLOT FILE, werden diese Einstellungen in eine Datei SVG_BI.log_1.gplot im Ordner /opt/fhem/www/gplot gesichert, der Plot ist nicht mehr leer, sondern zeigt die Temperaturdaten aus der Logdatei. Darüber hinaus taucht die neue Gplot-Datei SVG_BI.log_1.gplot danach auch in der Liste editierbarer Dateien auf, die in Bild 4.16 dargestellt ist.

8.2.3 Bedienung und Optionen von Plots

In der Raumübersicht werden natürlich nicht die komplizierten Einstellungen des Plot-Editors angezeigt, sondern nur der Plot selbst mit einem Link darunter. Während die „Innereien“ des Diagramms mit dem Plot-Editor festgelegt wurden, kann man die Größe mit einem weiteren Attribut des SVG-Device bestimmen: attr SVG_BI.log_1 plotsize 600,300 würde das Diagramm mit einer Größe von 600 × 300 Pixeln zeichnen.

Links oben in Bild 8.7 sind ferner drei Icons sichtbar:

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

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

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

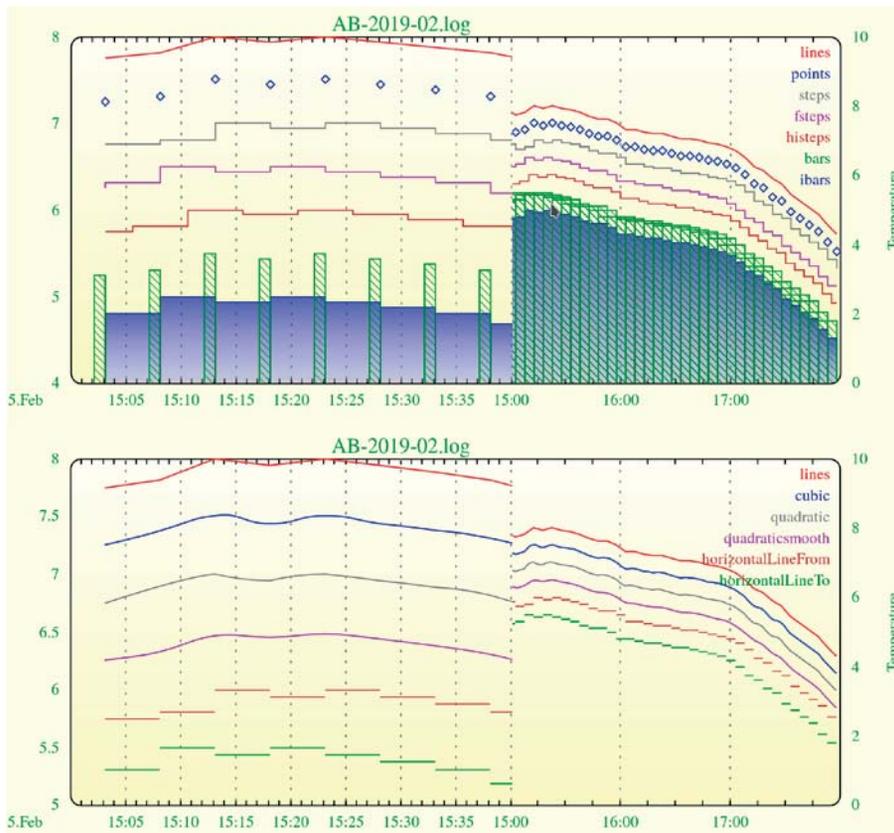


Bild 8.8 Verschiedene Linienstile in Plots

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

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

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

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

8.2.4 Gplot-Dateien

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

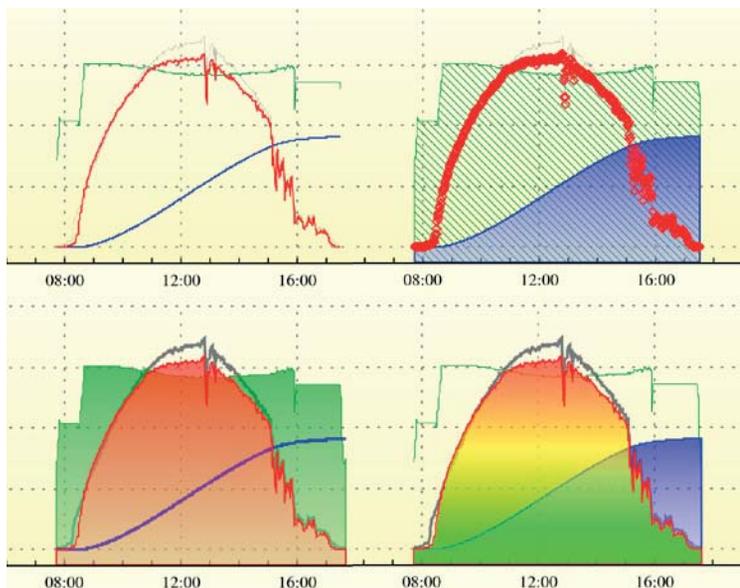


Bild 8.9 Verschiedene Linien- und Füllstile für Plots

■ 12.4 ChatBot mit RiveScript

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



Beispieldialog mit FHEM

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

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

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

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

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

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

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

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



Notfalldialog mit FHEM:

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

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

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

```
cpan RiveScript
```

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

12.4.1 RiveScript zur ChatBot-Programmierung

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

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

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

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

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

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

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

12.4.2 Notruf in FHEM

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

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

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

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

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

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