



Bild: Albert Hulm

Reaktionsmaschine

Einstieg in Heimautomation mit Node-Red

Vernetzte Geräte sollen ihren Besitzern das Leben leichter machen. Hersteller bezeichnen sie deshalb gern als „smart“. Doch damit das Licht angeht, wenn es draußen dämmt und jemand zu Hause ist, muss erst ein Mensch eine Regel erstellt haben. Mit Node-Red geht das in einer leicht verständlichen Oberfläche.

Von Jan Mahn

Für ein vernetztes Zuhause braucht es auf der einen Seite Sensoren, also Geräte, die auf ihre Umwelt reagieren. Sie messen die Temperatur oder reagieren auf einen Tastendruck. Auf der anderen Seite stehen Aktoren, also Geräte, die auf Befehl etwas schalten: eine Lampe, eine Heizung oder ein Rolll.

Damit die Aktoren sinnvolle Schaltbefehle bekommen, braucht es eine Zentrale, die der Heimautomatisierer mit Regeln füttert. Sie wertet Sensorwerte aus und reagiert auf Ereignisse. Viele Hausautomationsprodukte bringen eigene Schaltzentralen mit – meist als Dienst in

der Cloud, gesteuert per App und beschränkt auf ein Herstelleruniversum. Damit Geräte unterschiedlicher Hersteller und Eigenbauprojekte in einem Haushalt zusammenspielen, braucht es offene Schaltzentralen wie Node-Red.

Zusammengestöpselt

Node-Red wurde ursprünglich von IBM entwickelt und 2016 unter Apache-Lizenz als Open-Source-Projekt veröffentlicht. Die Software macht den Zusammenbau von Regeln sehr anschaulich und einfach. Der Nutzer zieht im Browser aus einem Inventar sogenannte Nodes auf die

Arbeitsfläche und verbindet sie mit der Maus zu einem Flussdiagramm. Die visuelle Schnittstelle ist dabei keinesfalls nur für Programmierlaien gedacht. Die Nodes geben Nachrichten untereinander als Objekte weiter. Wer JavaScript beherrscht, kann an jeder Stelle mit Code in den Verlauf eingreifen. Das macht Node-Red auch für komplexe Aufgaben interessant. Fertige Nodes als Schnittstellen gibt es nicht nur für viele Hausautomationsprodukte, sondern auch für Webdienste und Programme. Die Software ist damit nicht auf die Vernetzung des Zuhauses beschränkt und könnte auch im Unternehmen beispielsweise Cron-Jobs und Skripte ersetzen.

Vor dem Start

Node-Red ist eine Node.js-Anwendung und läuft unter Windows, Linux und macOS sowie auf dem Raspberry Pi oder BeagleBone. Da eine Zentrale für die Hausautomation rund um die Uhr laufen soll, eignet sich ein Raspberry als kostengünstiger und stromsparender Server, der nebenbei auch problemlos andere Heimserver-Aufgaben erledigen kann. Der kürzeste Weg zu einem Node-Red-Server führt auf allen Systemen über einen Docker-Container, der alle Abhängigkeiten enthält.

Voraussetzung ist die Container-Plattform Docker. Unter Windows, Linux und macOS finden Sie die Installations-

datei der kostenlosen Community-Version (CE) über docker.com. Unter Raspbian landet Docker mit folgendem Kommandozeilenbefehl auf dem System:

```
curl -sSL https://get.docker.com | sh
```

Anschließend muss der angemeldete Nutzer das Recht bekommen, Docker zu benutzen. Geben Sie dazu folgende Zeile ein und starten Sie den Raspberry neu:

```
sudo usermod -aG docker $user
```

Um mehrere Container in einer definierten Umgebung zu starten, empfiehlt sich Docker-Compose, das auf dem Raspi als eigenes Paket installiert werden muss:

```
sudo apt install docker-compose
```

Ab dafür

Für die ersten Versuche mit Node-Red soll der Server zusammen mit dem MQTT-Broker Mosquitto gestartet werden. MQTT ist ein Protokoll, das aus der Industrie stammt und mit dem Geräte Nachrichten austauschen können [1]. Die komplette Docker-Umgebung haben wir zu einem GitHub-Repository zusammengepackt. Laden Sie es per Git herunter und wechseln Sie anschließend in den Ordner:

```
git clone https://github.com/
jamct/node-red-pack.git
cd node-red-pack
```

Auf einem x64-Computer starten Sie das Container-Paket mit dem Befehl:

```
docker-compose up -d
```

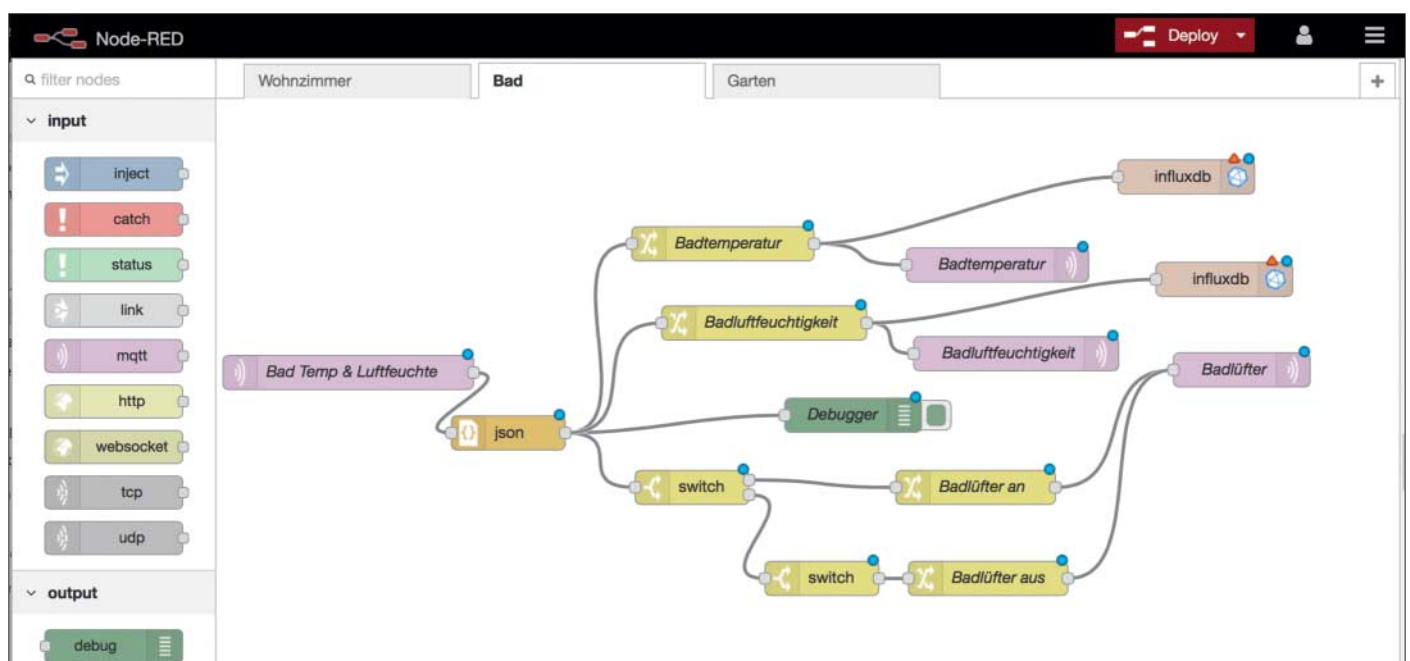
Auf dem Raspberry Pi müssen Sie zusätzlich die Konfiguration für die ARM-Images einbinden:

```
docker-compose -f docker-compose.yml
-f docker-compose.raspi.yml up -d
```

Wenn es Abend wird

Die Node-Red-Oberfläche ist jetzt bereit. Öffnen Sie im Browser die IP-Adresse des Servers, gefolgt von `:1880`. Der Regel-Editor besteht aus einer Node-Bibliothek auf der linken und einer großen Arbeitsfläche auf der rechten Seite. Der Arbeitsbereich kann in Tabs unterteilt werden. Wenn man Regeln für das ganze Haus zusammengebaut hat, wäre das auf einer Fläche schnell zu unübersichtlich. Sinnvoll ist eine Trennung nach Räumen oder Themengebieten wie Licht, Heizung, Sicherheit.

Das Beispielprogramm soll auf den Sonnenuntergang reagieren und per MQTT einen Schaltbefehl an eine Lampe erteilen. Scrollen Sie links in der Bibliothek ganz nach unten und ziehen Sie den Node `sunrise` auf die Arbeitsfläche. Der kann ein Ereignis auslösen, sobald die Sonne an Ihrem Standort rechnerisch auf- oder untergeht. Als Information



Auch komplexe Abläufe – wie die Steuerung eines Badlüfters abhängig von der Luftfeuchtigkeit – sind in Node-Red übersichtlich.

Der Node „sunrise“ berechnet den Sonnenstand am eingegebenen Standort. Mit der Information kann man zum Beispiel eine Lampe oder ein Rollo steuern.

benötigt sunrise nur den Längen- und Breitengrad.

Mit einem Doppelklick auf den Node öffnen Sie das Fenster mit den Einstellungen. Der Browser fragt Sie jetzt nach Ihrem Standort. Wenn Sie Standortdienste im Browser deaktiviert haben oder einen anderen Standort angeben möchten, beschaffen Sie sich die Koordinaten Ihres Zuhauses zum Beispiel über einen Kartendienst. In den beiden Dropdown-Menüs darunter können Sie definieren, ob der rechnerische Auf- und Untergang oder der erste und letzte Sonnenstrahl berücksichtigt werden sollen. In der Praxis empfiehlt es sich später, verschiedene Optionen auszuprobieren. Geben Sie dem Node im letzten Feld einen Namen wie „Sonne“ und speichern Sie mit „Done“ ab.

Scrollen Sie jetzt links im Inventar nach oben zu den „Output“-Nodes und ziehen Sie einen MQTT-Output-Node auf die Arbeitsfläche. Ausgehende Nodes erkennen Sie daran, dass das kleine graue Quadrat auf der linken Seite angebracht ist. Klicken Sie doppelt auf den Node, um ihn zu konfigurieren. Den MQTT-Server wählen Sie mit einem Klick auf das Stift-Symbol aus. Im Docker-Universum können sich die Container untereinander komfortabel per Hostname finden – wählen Sie als Serveradresse „mqtt“ und als Port den vorgegebenen Standard-Port „1883“. Haben Sie bereits einen anderen Broker zu Hause im Einsatz, können Sie auch dessen IP-Adresse angeben. Fügen Sie den Server mit „Add“ hinzu. Der Node benötigt jetzt ein Topic, auf dem er die MQTT-Nachricht rausgeben soll – zum Beispiel „garten/aussenlampe“.

Der Sonnen-Node hat auf der rechten Seite zwei graue Quadrate, die seine Ausgänge symbolisieren. Der obere Aus-

gang löst jede Minute aus und verschickt als Nachricht „1“ für Tag und „0“ für Nacht. Der untere Ausgang löst immer dann aus, wenn sich der Status verändert, also morgens und abends. Klicken Sie auf den unteren Ausgang und ziehen Sie eine Verbindung zum Eingang des MQTT-Nodes.

Bis jetzt passiert nichts. Nach jeder Änderung auf der Arbeitsfläche müssen Sie den roten „Deploy“-Button oben rechts betätigen. Sollte Ihr Programm einmal nicht wie geplant funktionieren, prüfen Sie, ob Sie vielleicht nur den Deploy vergessen haben. Node-Red meldet, dass es das Programm erfolgreich gestartet hat. Haben Sie ein MQTT-taugliches Gerät angeschlossen, das auf die Werte 1 und 0 reagiert, können Sie Ihre erste Automation bereits in Aktion erleben. Für den Test reicht ein Blick auf das Fenster „debug“, das Sie oben rechts öffnen. Hier protokolliert Node-Red die MQTT-Ereignisse. Wenn Sie andere Werte in den Debugger schreiben möchten, gibt es einen Output-Node mit dem Namen debug.

Verkehrte Welt

Bisher ist das Programm wenig sinnvoll. Gibt der Sonnenstand am Tag eine 1 zurück, wird dieser Wert der MQTT-Lampe übergeben und sie geht an. Anders herum wäre es sinnvoller. Um das zu erreichen, schreiben Sie eine kleine JavaScript-Funktion und lernen dabei gleichzeitig die Datenstruktur hinter den Nachrichten kennen:

Löschen Sie die Verbindung zwischen Sonnenstand und MQTT (Linie anklicken und Entf-Taste drücken), platzieren Sie einen Node mit der Bezeichnung „function“ in der Mitte und ziehen Sie neue Ver-

bindungen zu den anderen Nodes. Mit einem Doppelklick auf den Node gelangen Sie zu deren Funktionsdefinition. Von Node zu Node übergibt Node-Red das Objekt msg. Es enthält verpflichtend die Variable msg.payload mit dem Inhalt der Nachricht, der jetzt manipuliert werden soll:

```
if(msg.payload == 1){
    msg.payload = 0;
}else{
    msg.payload = 1;
}
return msg;
```

Mit diesen Zeilen vertauscht die Funktion 1 und 0 und gibt das veränderte Objekt msg über seinen Ausgang weiter.

Sonne für alle

Der Stand der Sonne ist nicht nur für diese eine Regel interessant. Er könnte bei vielen Automationsaufgaben eine Rolle spielen. Damit man einen Wert nicht immer neu abrufen muss, bietet Node-Red das Konzept globaler Variablen an. Ziehen Sie eine Funktion auf die Arbeitsfläche und verbinden Sie ihren Eingang mit dem oberen Ausgang des Sonnen-Nodes. Die Funktion bekommt jetzt zwei Zeilen Code:

```
global.set('sun', msg.payload);
return msg;
```

Mit global.set() weisen Sie hier der globalen Variable sun einen Wert zu. Diesen kann jetzt ein beliebiger anderer Node auslesen und daraus eine Entscheidung ableiten. Nutzen Sie zum Beispiel einen eingehenden MQTT-Node für einen Lichtschalter mit dem Topic „/living-room/switch“, verbinden Sie ihn mit einem Funktions-Node und diesen mit einem MQTT-Output. Die folgende Funktion sorgt dafür, dass das Licht nur dann angeht, wenn jemand den Schalter betätigt hat und es draußen dunkel ist:

```
if(msg.payload == "1" &&
global.get('sun') == "1"){
    msg.payload = "1";
}else{
    msg.payload = "0";
}
return msg;
```

Horizont erweitern

Ein besonderer Reiz von Node-Red liegt in der großen Community. In der Grundinstallation ist das Inventar bereits üppig

gefüllt. Mit weiteren Nodes schaffen Sie Verbindungen zu Hausautomationsprodukten anderer Hersteller oder zu Sprachassistenten wie Amazons Alexa. Die Oberfläche für neue Nodes ist etwas versteckt. Klicken Sie oben rechts auf das Hamburger-Menü, dann auf „Manage palette“ und im Dialog auf den Reiter „Install“. Suchen Sie beispielsweise nach „Hue“, zeigt Node-Red zahlreiche Nodes für die weitverbreiteten Lampen an. Die Bezeichnung `contrib` im Namen weist auf ein Paket aus der Community hin. Ein Klick auf den kleinen blauen Pfeil führt zu einer ausführlichen Beschreibung des Nodes.

Sehr nützlich ist der Node „node-red-dashboard“. Damit erstellen Sie schnell eine Bedienoberfläche mit Buttons, Farbwählern und Diagrammen. Ein ausgemustertes Tablet mit einem Browser genügt, um ein Bedienpanel für das Zuhause zu bauen.

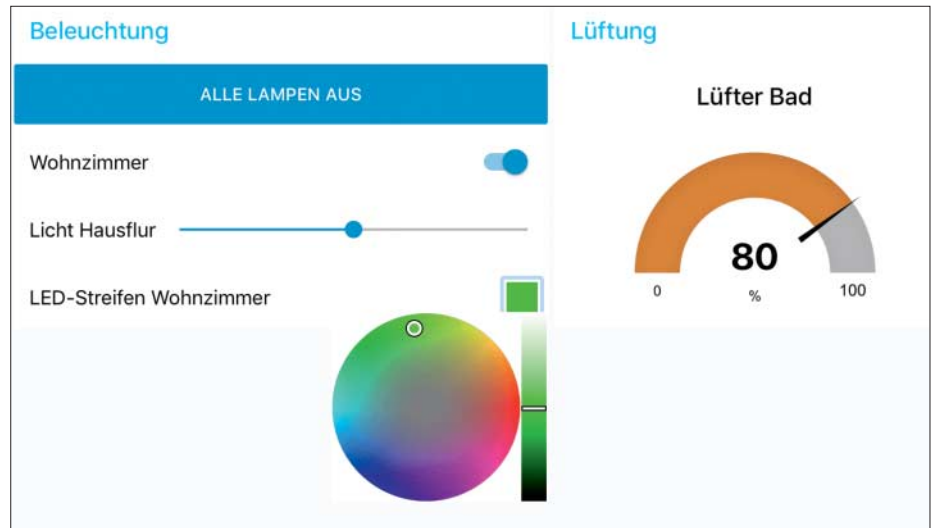
Nach der Installation des Nodes finden Sie in der Palette viele neue Nodes mit sprechenden Bezeichnungen. Fügen Sie einen Button hinzu, öffnen Sie seine Eigenschaften per Doppelklick und geben Sie ihm einen Namen. Damit er angezeigt wird, müssen Sie ihm eine Gruppe zuweisen und der Gruppe wiederum einen Tab. Klicken Sie dazu jeweils auf den Button mit dem Stift-Symbol. Schließen Sie anschließend alle Fenster. Nach einem Klick auf „Deploy“ ist die Oberfläche unter der Adresse „<Server-IP-Adresse>:1880/ui“ zu erreichen.

Aber sicher

Im Standard-Setup kann jeder im Netzwerk die Administrationsoberfläche von Node-Red erreichen. Das ist nicht schlimm, wenn Sie ein eigenes Netz für die Hausautomation betreiben, das nur Sie erreichen können. Um ein Kennwort festzulegen, ist etwas Handarbeit nötig, einen Assistenten gibt es dafür leider nicht.

Öffnen Sie eine Kommandozeile auf dem Server und zeigen Sie mit `docker ps` alle laufenden Container an (ziehen Sie das Fenster in der Breite auf, bis alle Spalten nebeneinander sind). Suchen Sie den Namen des Node-Red-Containers in der letzten Spalte. Wenn Sie den Ordner nicht umbenannt haben, ist das `node-red-pack_nodered_1`. Springen Sie jetzt in den Container:

```
docker exec -it node-red-pack_nodered_1 sh
```



Mit der Erweiterung „node-red-dashboard“ entsteht eine Weboberfläche, die auch auf mobilen Geräten gut bedienbar ist.

Sie befinden sich in der Kommandozeile im Container. Führen Sie dort folgende Zeile mit Ihrem Kennwort aus:

```
node -e "console.log(  
  require('bcryptjs').  
  hashSync(process.argv[1],  
  8));" KENNWORT
```

Sie bekommen eine Zeichenkette, die Sie in die Zwischenablage kopieren. Beenden Sie den Besuch im Container mit `exit` und navigieren Sie im Dateixplorer des Rechners zum Ordner „node-red-pack“. Einstellungen und angefallene Daten von Node-Red legt Docker im Ordner „data/nodered“ ab. Öffnen Sie hier die Datei „settings.js“ und scrollen Sie bis zur Zeile 118 (`adminAuth: {}`). Entfernen Sie die Kommentarzeichen // bis Zeile 125 (`()`),). In Zeile 122 ist hinter `password:` der Platz für die kryptische Kennwort-Zeichenkette, die Sie zuvor erzeugt haben. Speichern Sie die Datei, stoppen Sie die Docker-Container mit `docker-compose down` im Verzeichnis der Container-Umgebung und starten Sie das Setup mit `docker-compose up` erneut. Die Weboberfläche meldet sich jetzt mit einer Anmeldemaske.

Import/Export

Wenn Sie Ihre Abläufe mit anderen teilen oder auf einen anderen Server übertragen möchten, müssen Sie die Nodes nicht per Hand nachbauen oder einen Screenshot verschicken. Node-Red hat eine Im- und Exportschnittstelle über JSON. Den Dialog finden Sie über das Menü in der oberen rechten Ecke. Unter „Export/Clip-

board“ können Sie auswählen, welchen Tab Sie exportieren möchten und den angezeigten Text in die Zwischenablage kopieren. Einen Download gibt es nicht, zum Transport müssen Sie den Inhalt in eine Textdatei einfügen. Über „Import/Clipboard“ gelangen Abläufe ins System. Node-Red beschwert sich, wenn die importierten Regeln Nodes enthalten, die Sie nicht installiert haben, kann die Abhängigkeiten aber nicht automatisch auflösen – Sie müssen diese über „Manage palette“ selbst nachinstallieren.

Abgeschaut

Beim Erstellen Ihrer Regeln werden Sie immer wieder auf neue Fragestellungen und Probleme treffen. Die meisten davon hatten andere Nutzer schon vor Ihnen. Im Node-Red-Forum finden Sie meist eine brauchbare Antwort und in der Flow-Library teilen andere Anwender ihre Regeln (siehe ct.de/yavx). Das Vernetzen des ganzen Heims bleibt harte Arbeit und ist nicht an einem Wochenende erledigt – mit Node-Red bekommt man zwar kein „smarteres“ Zuhause, aber immerhin eine Grundlage für eigene Regeln.

(jam@ct.de) **ct**

Literatur

[1] Jan Mahn, Weltsprache, Das Protokoll MQTT für robusten Datenaustausch in Industrie und Hausautomation, c't 6/2018, S. 164

Dokumentation und Downloads:
ct.de/yavx