

Rotfront!

Grafische Oberflächen für Node-Red entwickeln

Mit dem Dashboard-Modul für Node-Red bauen Sie in Windeseile ansprechende Oberflächen für das eigene Smart-Home.



Bild: Albert Hulm

Die Automation in "Hausautomation" ist das Ziel, aber im real existierenden Hausautomatismus muss man auch mal manuell eingreifen. Das macht mit einer aufgeräumten und auf jedem Gerät lauffähigen Oberfläche mehr Spaß als mit Kommandozeilenwerkzeugen und Konfigurationseditoren. Darüber hinaus ist es für weniger technikaffine Mitmenschen zugänglicher. Wer noch keine Erfahrungen mit Node-Red hat, sollte zunächst mit dem Artikel "Reaktionsmaschine" [1] aus c't 15/2018 einsteigen. Darin werden sowohl die Einrichtung als auch die ersten Schritte mit dem grafischen Entwicklungswerkzeug beschrieben.

Die Node-Red-Oberflächen können Sie vollständig innerhalb des Regeleditors ohne aufwendige Programmiererei bauen. Die Oberfläche läuft im Browser. Bei Smartphone und Tablets verhält sie sich sogar wie eine native App, denn sie ist als sogenannte Progressive-Web-App (PWA) umgesetzt. Die Gestaltungsmöglichkeiten der Oberflächen sind dabei - wie von Node-Red gewohnt - weitläufig: Von der einfachen Textanzeige über Verlaufsdiagramme bis hin zum Farbwahl-Dialog sind ausreichend Widgets dabei, um die Sensoren und Aktoren des Heims zu überwachen und zu steuern.

Im Rahmen der Erstellung komplexerer Dashboards gibt es zudem noch ein paar (Node-Red-)Fallstricke, denen man leicht ausweichen kann.

Einrichtung

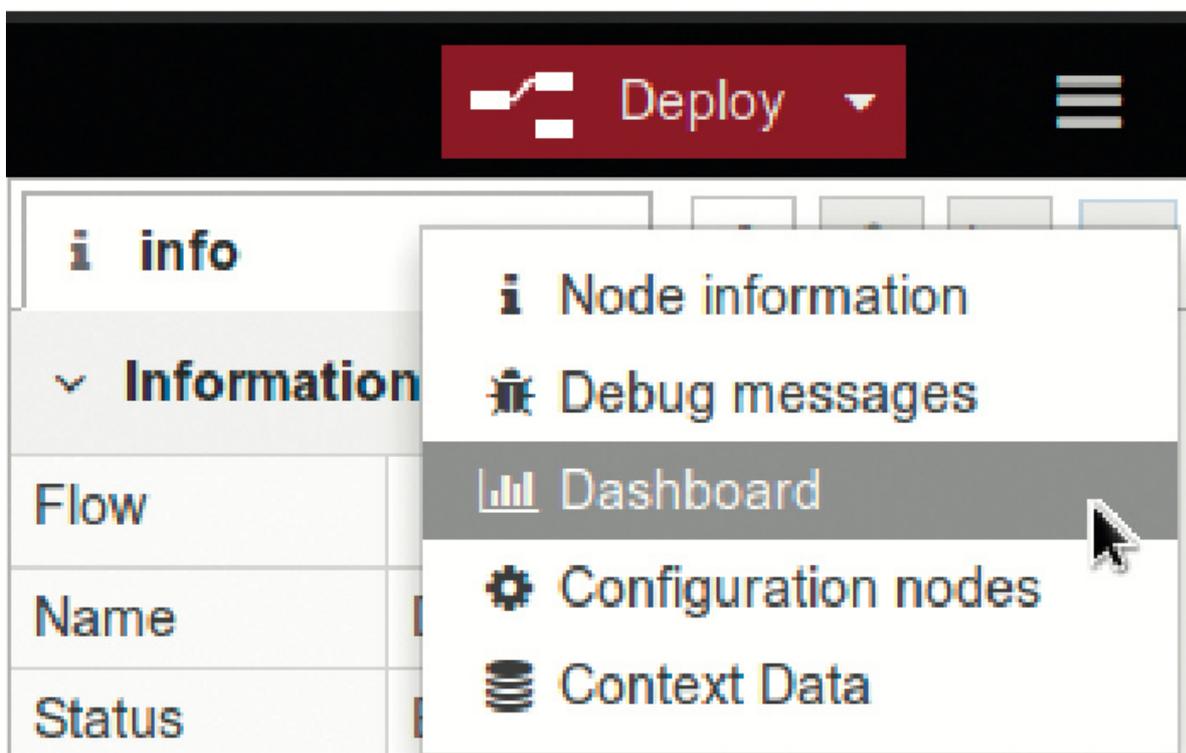
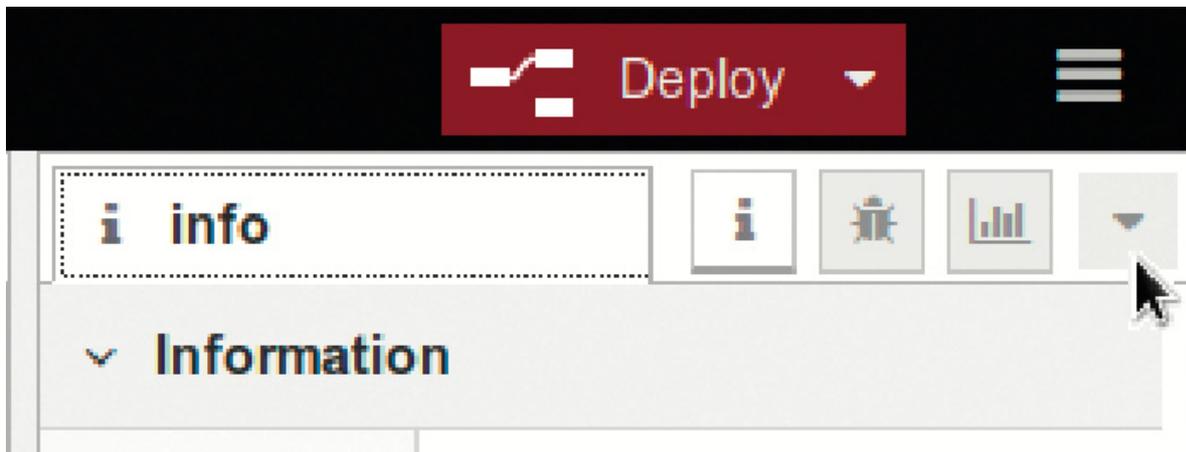
Das Dashboard-Modul installieren Sie wie alle anderen Erweiterungen über die Palette. Klicken Sie dazu im Regeleditor rechts oben auf das Hamburger-Menü (die drei horizontalen Balken) und anschließend auf "Manage palette". Im erschienenen Dialog wechseln Sie zum Reiter "Install". In das Suchfeld tippen Sie dann "node-red-

dashboard" ein und klicken Sie beim gleichnamigen Eintrag auf "Install". Nachdem die Installation abgeschlossen ist, schließen Sie den Dialog mit einem Klick auf "Close".

In der Liste der verfügbaren Nodes am linken Rand des Editors finden Sie nun eine Kategorie mit dem Titel "dashboard" und 16 neue Nodes. Auch neu ist der Reiter "Dashboard" in der rechten Seitenleiste. Er versteckt sich vermutlich hinter dem Overflow-Menü mit dem nach unten zeigenden Dreieck. Hier können Sie die Elemente, Gruppen und Tabs der Oberfläche sortieren und strukturieren. Nach dem ersten Klick auf den Reiter erzeugt Node-Red im Hintergrund einen Konfigurations-Node für das Dashboard, der dann im Reiter "Configuration nodes" erscheint. Lassen Sie ihn einfach in Ruhe, man kann an ihm nichts konfigurieren und er wird immer wieder neu erstellt, wenn man ihn löscht.

Die Seitenleiste für das Dashboard hat drei Unterreiter: "Layout", "Site" und "Theme". Klicken Sie zunächst auf "Site", um die Grundeinstellungen der Dashboard-Ansicht festzulegen. Die Einstellung "Title" legt den im Browser und auf dem Startbildschirm der PWA angezeigten Titel fest. Darunter kann man definieren, wie sich die Oberfläche verhält. Die erste Option "Show the titlebar" bestimmt, ob die Titelleiste und das Hamburger-Menü des Dashboards sichtbar sein sollen oder nicht. Für den Anfang lässt man sie besser sichtbar.

Grundsätzlich spart man durch das Verstecken aber Platz, was gerade bei Smartphones wünschenswert ist. Dafür verliert man die Möglichkeit, über das Menü zu anderen Tabs der Oberfläche zu wechseln. Das kann man durch das Einstellen von "Always show side menu" im folgenden Dropdown-Feld nur teilweise kompensieren. Dadurch wird nämlich eine Liste der Tabs immer eingeblendet, unabhängig von der Sichtbarkeit des Menüs. Auf Smartphones bleibt die Liste aber unsichtbar. Um das Wechseln auf dem Smartphone zwischen den Seiten zu vereinfachen, gibt es die Option Nummer drei ("Allow swipe between tabs"). Sie regelt die Möglichkeit, per Wischgeste zwischen den Seiten hin- und herzuwechseln. An sich ist das ein schönes Feature, aber die Software missversteht das Einstellen von Regeln gern als Wischgeste. Daher sollte man das Feature abstellen, falls es nervt.

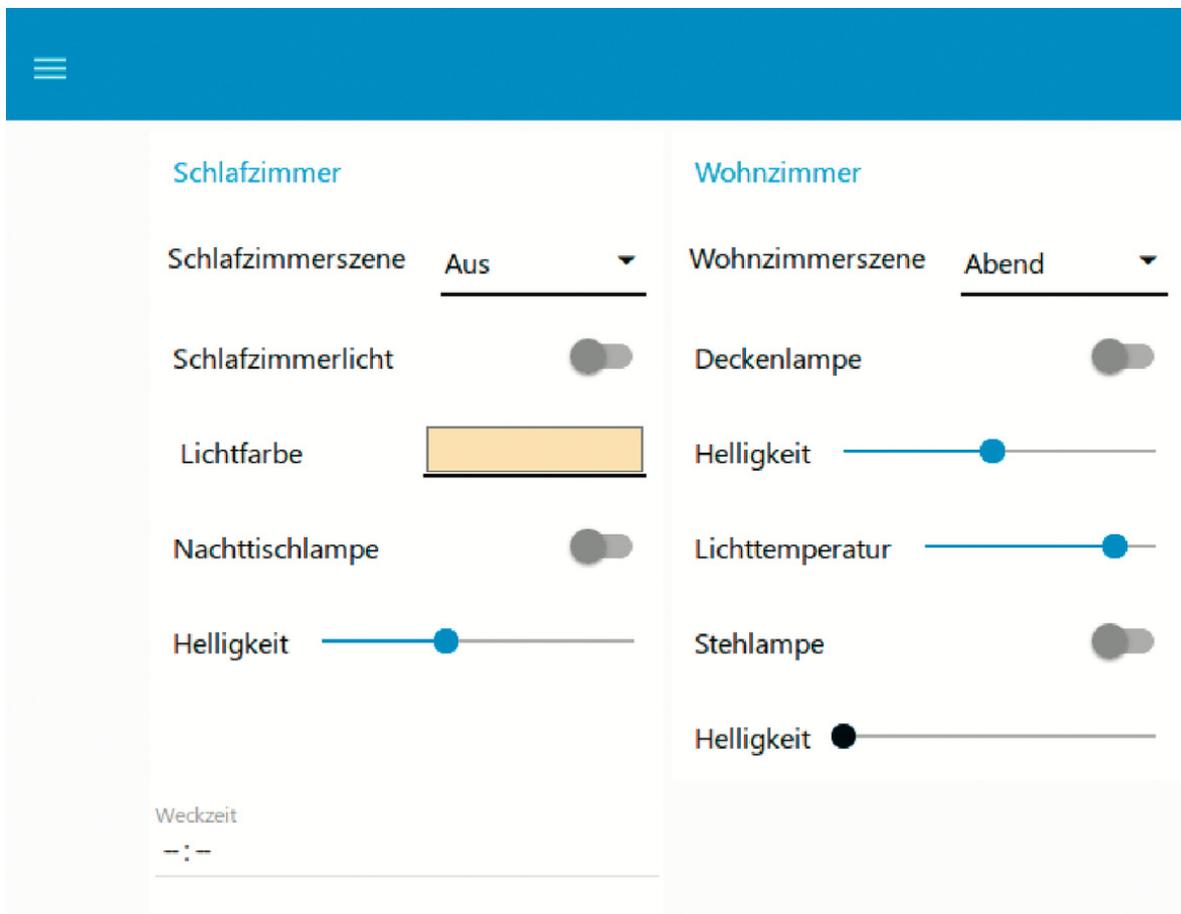


Die Einstellungen für das Node-Red-Dashboard verstecken sich hinter dem Dropdown-Menü am rechten Rand.

Die letzte Option des Blocks legt fest, welches Design das Dashboard verwendet. Die Standardeinstellung "Node-RED theme everywhere" sollte für den Anfang reichen und bietet im Reiter "Theme" gleich rechts daneben die meisten Einstellungsmöglichkeiten, um Farben und Akzente für das Dashboard zu definieren. Nachfolgend kommt noch das Datumsformat, das Sie nach Wunsch anpassen können. Welche Platzhalter Sie für welche Datumsfragmente benötigen, verrät ein Klick auf das kleine umkreiste *i* am rechten Rand. Die letzten Einstellungen zu den Widget-Größen sind von rein ästhetischer Natur. Sie legen die Rastergröße für die Elemente ("1x1 Widget Size"), Abstände ("Spacing") und Freiräume ("Padding") fest.

Struktur

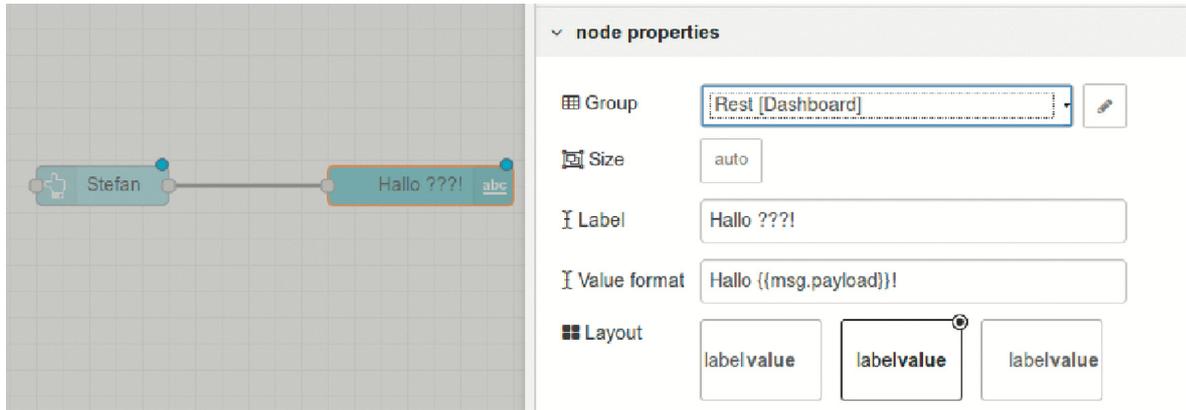
Wechseln Sie anschließend zum Reiter "Layout". An der rechten Seite gibt es zwei kleine Knöpfe. Der erste ist mit "+tab", der zweite mit "+link" beschriftet. Klicken Sie zunächst auf den "+tab"-Knopf, um einen neuen Tab zu erzeugen. Daraufhin erscheint ein neues Element mit dem Titel "Tab 1" in der Liste. Die Tabs sind Seiten innerhalb des Dashboards und enthalten wiederum Gruppen, in denen sich ihrerseits die Widgets befinden. Wenn Sie mit der Maus über den Tab in der Liste fahren, werden hier zwei Knöpfe sichtbar: "+group" und "edit". Klicken Sie auf "edit", um den Namen sowie das Icon des Tabs zu ändern. Legen Sie einen Namen wie etwa "Lampen" oder "Wohnzimmer" fest. Ein Icon können Sie aus drei Bibliotheken wählen: Googles "Material Design"-Icons, das populäre "Font Awesome"-Set und die "Weather Icon"-Sammlung von Paul Reed. Tragen Sie den Namen des Icons, den Sie der Bibliothekswebsite entnommen haben, zum Beispiel `fa-lamp` oder `signal_wifi_off`, ein. Im hellgelb hinterlegten Info-Kasten am Ende des Bearbeitungsdialogs sind Links zu den Symbol-Übersichten der drei Projekte. Klicken Sie auf "Update", um die Änderungen zu übernehmen.



Die Eingaben, die der Anwender auf der Oberfläche macht, verarbeiten die Flows von Node-Red.

Fügen Sie nun mittels des Knopfs "+group" eine neue Widget-Gruppe hinzu. Die erscheint als Unterelement des eben erzeugten Tabs. Auch hier erscheint beim Überfahren mit der Maus ein "edit"-Knopf. Klicken Sie ihn an, um der Gruppe einen Namen zu geben, wie etwa "Deckenlampe". Neben der Option, einen Namen festzulegen, können Sie hier den Tab auswählen, in dem sie erscheinen soll und falls nötig über "Add new ui_tab..." einen neuen Tab hinzufügen. Darunter gibt es die Option, die Breite ("Width") der Gruppe festzulegen. Standardwert ist hier 6, was in etwa der Bildschirmbreite eines Smartphones entspricht. Sind mehrere Gruppen mit der Breite 6 nebeneinander, werden Sie, wenn möglich, nebeneinander platziert. Geht das nicht, erscheinen Sie untereinander. Ist ein Widget zu groß für das Browserfenster, muss man hin- und herscrollen. Wollen Sie die Gruppe zur optischen, nicht aber inhaltlichen Organisation der Widgets verwenden, können Sie deren Namen und Titelzeile mit "Display group name" an- und abschalten. Soll man sie beim Klick auf den Titel einklappen können, setzen Sie den Haken bei "Allow group to be collapsed".

Elemente



Mit zwei Nodes und einem Mausklick begrüßt einen Node-Red mit Namen.

Ziehen Sie nun aus der Node-Liste am linken Rand einen "button" in den aktuellen Flow. Machen Sie dann einen Doppelklick auf den neuen "button", um ihn zu bearbeiten. Im Bearbeitungsdialog ist die eben angelegte Gruppe schon vorgewählt. Klicken Sie direkt wieder auf "Done", um die Einstellungen zu speichern. Klicken Sie dann rechts oben auf "Deploy", um die Änderungen anzuwenden. Wenn Sie nun hinter die URL der Node-Red-Adresse ein /ui anhängen, etwa <https://meinserver:1880/ui>, sollte sich das Dashboard mit dem so eben erzeugten Knopf öffnen. Der Knopf ist bereits anklickbar, auch wenn er nichts auslöst.

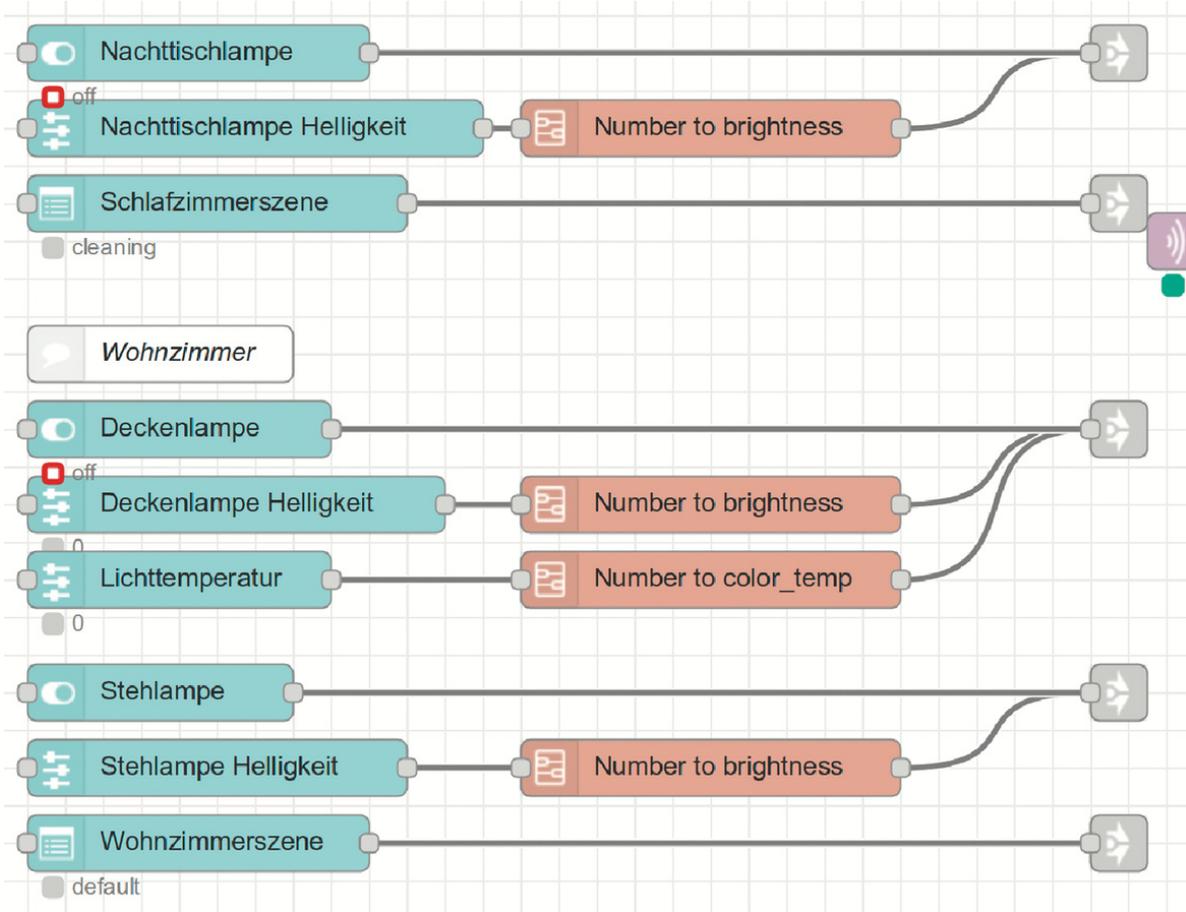
Die Dashboard-Nodes arbeiten wie alle anderen Elemente in Node-Red auch mit msg-Objekten und Payloads. Sie geben die Payloads in unterschiedlichen, meist konfigurierbaren Formen aus oder nehmen Sie entgegen. Hier ein kurzes Beispiel:

Wechseln Sie zurück zum Regeleditor, fügen Sie ein "text"-Node ein und machen Sie einen Doppelklick darauf. Tragen Sie beim Feld "Label" den Text Hallo ein. Bei "Value format" fügen Sie ein Ausrufungszeichen hinter `{{msg.payload}}` an. Sichern Sie die Änderungen dann mit einem Klick auf "Done". Nun bearbeiten Sie noch mal den Button und tragen im Feld Payload Ihren Namen (etwa "Stefan") ein und speichern sie die Änderungen. Zum Abschluss ziehen Sie eine Verbindung vom Ausgang des Knopfes zum Eingang des Textelements und klicken auf "Deploy".

Wenn Sie nun das Dashboard öffnen und den Knopf klicken, wird ein msg-Objekt mit der Payload abgeschickt. Der Text-Node übernimmt dann den Text in msg.payload und zeigt ihn wie in "Value format" definiert an. Sie sollten vom Dashboard mit Ihrem Namen begrüßt werden.

Öffnen Sie nun noch einmal die Einstellungen des Knopfes. Neben der Gruppe können Sie dort auch noch dessen Größe ("Size") festlegen. Im Gegensatz zu den Gruppen kann sie die Breite und Höhe eines Widgets festlegen. Die Angabe ist dimensionslos. In der Voreinstellung "auto" füllt das Widget die gesamte Breite der Gruppe aus und nimmt die minimal nötige Höhe ein - wie hoch das genau ist, kommt auf das jeweilige Widget und dessen Inhalte an. Im Feld Icon können Sie genauso einen Icon-Namen angeben wie bei den Gruppen. "Label" legt die Beschriftung des Widgets fest. "Color" und "Background" definieren Schrift- und Hintergrundfarbe. Die Felder nehmen hexadezimale HTML-Farbcodes in Form von #3FA201 oder HTML-Farbnamen wie red entgegen.

Durchschaltung

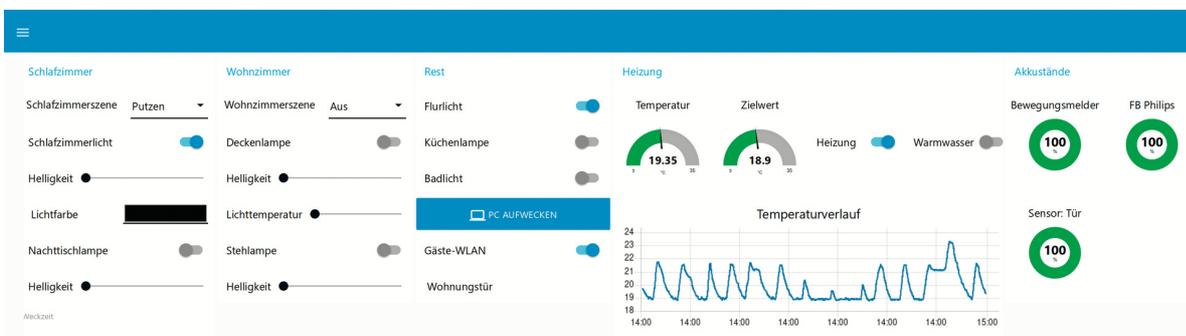


Die Dashboard-Nodes dienen zur Gestaltung der Oberfläche.

Viele der Dashboard-Nodes haben, wie auch der Button, neben dem Ausgang noch einen Eingang. Damit können Sie sowohl Daten einfach durchleiten, als auch den Zustand des Elements bestimmen. Das ist zum Beispiel dann nützlich, wenn der Schalter oder ein Dropdown-Feld eine bestimmte Aktion auslösen soll, deren Durchführung vom Aktor bestätigt werden soll. Beispielsweise, wenn man die Heizung einschaltet und diese den Vorgang bestätigt. Dazu sendet der Schalter-Node zunächst die Aufforderung "Heizen" an den Node für die Heizungssteuerung. Dieser Node sendet dann ein Feedback, das man etwa mit Hilfe eines "change"-, "switch"- oder wenn nötig "function"-Nodes auswertet. Das Ergebnis schickt man an den Schalter zurück.

Wichtig sind jedoch zwei Dinge: Erstens muss das Ergebnis den möglichen Payloads für den Schalter entsprechen. Entsprechen sie nicht den Vorgaben, zeigt der Schalter ein äußerst erratic Verhalten: Er springt auf Aus und sendet die letzte korrekte Payload erneut. Noch wichtiger ist, dass Sie den Haken bei "If msg arrives on input, pass through to output" in den Einstellungen des Schalters entfernen. Der Haken legt fest, ob der Node eingegangene msg-Objekte an die folgenden Nodes durchreichen soll. Im Beispiel führt das zu einer Schleife, bei der die Heizung immer wieder den Befehl "Heizen", den Sie dann bestätigt, was den Vorgang wieder von vorne auslöst.

Eingangsfragen



Von der Zimmertemperatur bis zum Gäste-WLAN lässt sich im Dashboard alles einsehen und regeln.

Einige Features der Eingabe-Nodes haben Ihre Tücken. So sollte man bei den Slider-Elementen beim Feld "Output" die Option "only on release" auswählen. Die Alternative "continuously while sliding" sendet bei jeder Veränderung den Wert des Sliders, was bei einigen Geräten zu Problemen führt. Konkret beginnen Zigbee-Lampen zu flackern, wenn man ihnen zu viele Steuersignale in kurzer Zeit liefert. Und die APIs Cloud-basierter Dienste machen schnell dicht,

wenn man sie mit Anfragen bombardiert. Mit der Einstellung "only on release" wird der Wert des Sliders erst dann abgeschickt, wenn der Nutzer den Finger von der Maustaste oder vom Touchscreen nimmt.

Edit dropdown node

Delete Cancel Done

node properties

Group

Size

Label

Placeholder

Options

<input type="checkbox"/>	default	<input type="text" value="Aus"/>	<input type="checkbox"/>
<input type="checkbox"/>	reading	<input type="text" value="Lesen"/>	<input type="checkbox"/>
<input type="checkbox"/>	evening	<input type="text" value="Abend"/>	<input type="checkbox"/>
<input type="checkbox"/>	cleaning	<input type="text" value="Putzen"/>	<input type="checkbox"/>

+ option

→ If arrives on input, pass through to output:

Topic

Name

Mit einem Dropdown-Menü lassen sich einfache Lichtszenen umsetzen.

Sehr hilfreich für Freunde bunter Beleuchtung ist der Color-Picker-Node. Mit seiner Hilfe kann man den ganzen Regenbogen von Farben für seine Lichter festlegen. Die Dashboard-Entwickler haben ihn vollständig in HTML umgesetzt, sodass er in jedem Browser läuft. Leider hat auch er mehr als einen Haken: So funktioniert in manchen

Android-Browsern die Farbwahl nicht korrekt und springt immer wieder zu der vorherigen Farbe zurück. Außerdem ist es aufgrund von Bugs nicht möglich, einfach von Schwarz zu Weiß zu wechseln. Das kann man aber durch einen zusätzlichen Helligkeits-Slider kompensieren.

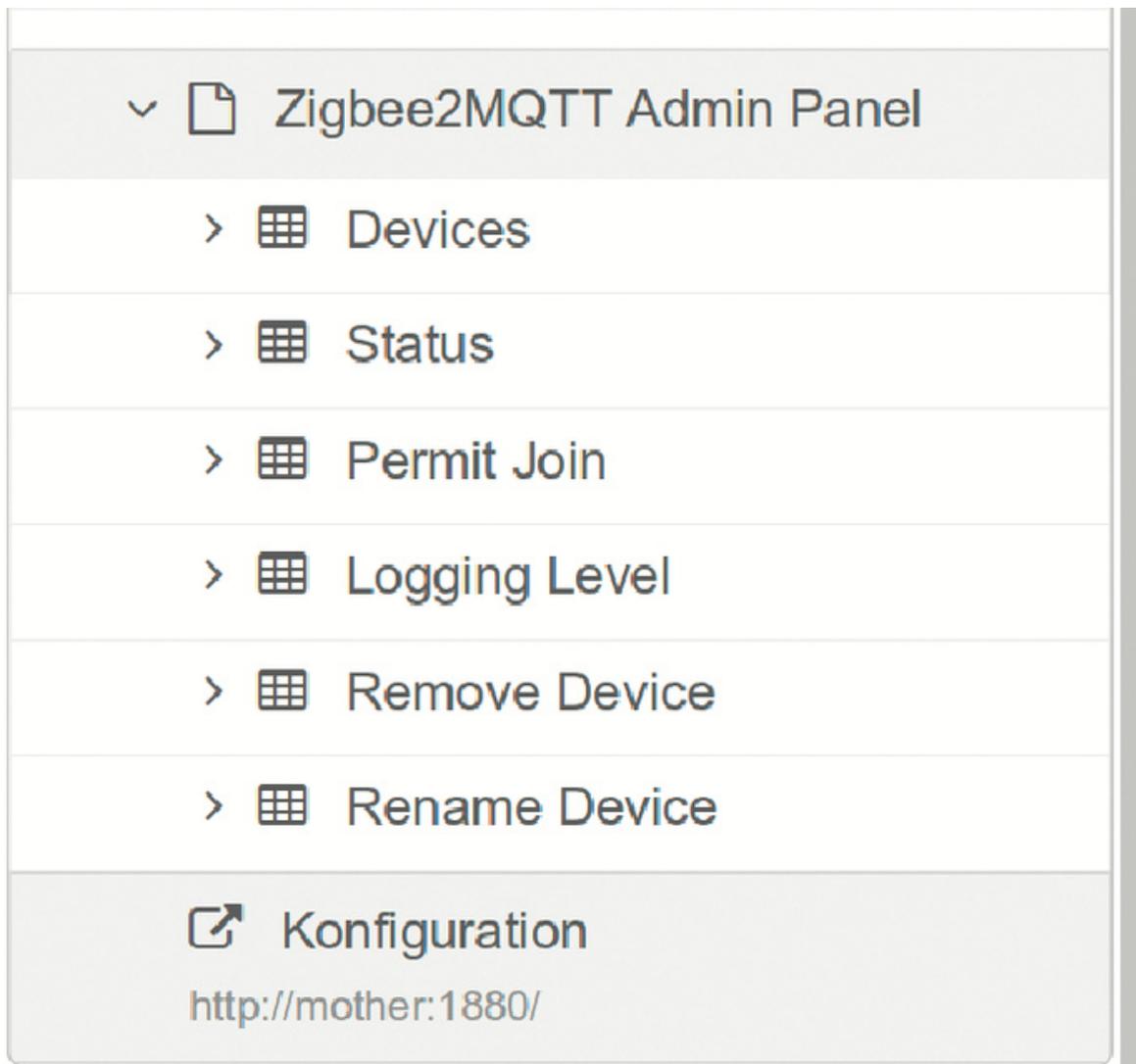
Eine gute Alternative findet sich mit dem Text-Input-Node. In dessen Optionen kann man mit "Mode" allerhand HTML5-Eingabetypen auswählen. Darunter auch einen HTML-Farbwähler, der vom Browser beziehungsweise Betriebssystem bereitgestellt wird und HTML-Farbcodes zurückgibt. Safari-Nutzer auf iOS und macOS sehen hier aber nur ein normales Textfeld, denn Apple hat den Farbwahl-dialog nicht implementiert. Wer in einem Haushalt mit heterogener Hardware-Ausstattung lebt, hinterlegt besser beide Varianten des Farbwählers.

Wer Dinge wie Daten, Zeiten oder Nummern in seinem Dashboard abfragt, sollte ebenfalls ein Auge auf den Text-Input-Node werfen. Die HTML-Eingabefelder sind besonders auf Mobilbrowsern hilfreich, denn anhand des Eingabetyps stellen diese auch die passende Tastatur bereit, etwa eine nur für Ziffern oder eine E-Mail-Tastatur mit einem @ und vielleicht schon dem passenden Vorschlag.

Für Formulardaten empfiehlt sich der "form"-Node. Wer bereits Erfahrung mit HTML-Formularen hat, wird sich sofort heimisch fühlen. Man kann verschiedene Eingabefelder von Text bis Checkboxen eintragen und bekommt im Dashboard ein Formular mit typischen "Absenden"- und "Abbrechen"-Knöpfen. Die vom Nutzer übermittelten Daten kann man dann in Node-Red als JSON verarbeiten.

Ordnung halten

The screenshot displays a dashboard interface with a top navigation bar. On the left, there is a search bar containing the text "dashboard" and a bar chart icon. To the right of the search bar are four icons: an information icon, a bug icon, a bar chart icon, and a dropdown arrow. Below the search bar, there are three main menu items: "Layout", "Site", and "Theme", each in a separate box. To the right of these items is a share icon. Below the main menu is a section titled "Tabs & Links" with four sub-headers: an up arrow, a down arrow, "+ tab", and "+ link". The main content area shows a list of tabs and links. The first tab is "Dashboard" with a document icon and a dropdown arrow. Below it are several links, each with a right-pointing arrow and a grid icon: "Schlafzimmer", "Wohnzimmer", "Rest", "Sensoren", "Heizung", and "Akkustände". At the bottom, there is another tab labeled "Akkustände" with a document icon and a dropdown arrow.



Mit Tabs, wie "Akkustände" und Gruppen, wie "Schlafzimmer", hält man die Widgets im Zaum und kann die Einstellungen strukturieren.

Bei vielen Dashboard-Widgets wird es in den Flows und auch den Dashboards schnell unübersichtlich. Daher sollte man lieber früher als später Vorgänge auftrennen. Dabei sind die Link-Nodes sehr hilfreich. Sie können über Flow-Grenzen hinweg Nodes und deren Nachrichten verbinden. So kann man in einem Flow alle Lampen im Haus sammeln und mit Links versehen, damit diese dann nur noch als kleine Link-Nodes auftauchen und nicht in jedem Flow dreimal ein großer Node namens "Wohnzimmerlicht" alles blockiert.

Wer für die Ausgabedaten des Dashboards repetitive Konvertierungen durchführen muss, zum Beispiel um Farbcodes in RGB-Werte zu wandeln, sollte Subflows erstellen. Diese erscheinen in den normalen Flows wie einzelne Nodes und können beliebig oft eingesetzt werden, um bestimmte Tätigkeiten durchzuführen. Auch dadurch wird die Verwaltung komplexerer Dashboards erheblich übersichtlicher.

Mit dem Node-Red-Dashboard lässt sich allerhand steuern und anzeigen. Grenzen setzt nur die eigene Fantasie. Damit es übersichtlich und nützlich bleibt, muss man probieren, wo und wie man eine Einstellung darstellt. Manchmal ist ein großer Knopf besser als ein kleiner Schalter und manchmal ist ein Dropdown-Menü nicht die klügste Wahl. Für den Anfang hilft es, einfach etwas herumzuspielen, um sich in das Dashboard und seine Möglichkeiten einzufinden und herauszufinden, welcher Vorgang welches Interface benötigt. (mls@ct.de)

Links und Downloads: ct.de/yxbn

Literatur

[1]Jan Mahn, Reaktionsmaschine, Einstieg in Heimautomation mit Node-Red, c't 15/2018, S. 142

<http://ct.de/yxbn>

c't Smart Home

Seit c't 02/2018 haben über mehrere Ausgaben hinweg gezeigt, wie Sie mit günstigen fertigen und selbstgebaute Komponenten ein cloudloses Smart Home aufbauen, das mit freier Software läuft. Weitere Artikel zum Thema finden Sie in Zukunft unter dem Motto "c't Smart Home".

An erster Stelle des c't Smart Home steht die Interoperabilität zwischen Komponenten vieler Hersteller - so sind Sie beim Einkauf der Gadgets nicht auf einen Anbieter festgelegt. Dafür setzen wir auf offene Protokolle wie beispielsweise MQTT. Das Telemetrieprotokoll ermöglicht sparsame, schnelle Kommunikation. Sensoren, zum Beispiel ein Fensterkontakt, schicken ihre Nachrichten unter einem Thema (Topic) an einen MQTT-Server (Broker) wie Mosquitto. Jede Smart-Home-Zentrale, die das Protokoll versteht, kann die Meldungen aus einem Topic abonnieren und verarbeiten - völlig unabhängig vom Hersteller des sendenden Geräts. Mit Zigbee2MQTT und HomeKit kommen Sensoren und Aktoren dazu.

Unser Fokus liegt auf Node-Red, eine Kombination aus Server und Framework, in der ein Smart-Home-Admin Regeln in einer grafischen Browser-Umgebung zusammenklickt. Das c't Smart Home läuft bei Ihnen zu Hause, beispielsweise auf einem Raspberry Pi und versagt auch beim Internetausfall nicht. Ein kostenloser Artikel, den Sie über ct.de/y2pj finden, erklärt die Grundlagen und die Basis-Installation, die Sie benötigen, um die Tipps in diesem und zukünftigen Artikeln umzusetzen.

Merlin Schumacher

Quelle:	c't Heft 2/2019 S. 160-165
ISSN:	0724-8679
Ressort:	Praxis
Rubrik:	Node-Red-Dashboard
Dokumentnummer:	1546757812903509

Dauerhafte Adresse des Dokuments: https://www-wiso-net-de.ezproxy.hs-augsburg.de/document/CT__5d3a13dc2aafa38c4d00603585faff3e28ff4f72
Alle Rechte vorbehalten: (c) Heise Zeitschriften Verlag GmbH & Co. KG

 © GBI-Genios Deutsche Wirtschaftsdatenbank GmbH