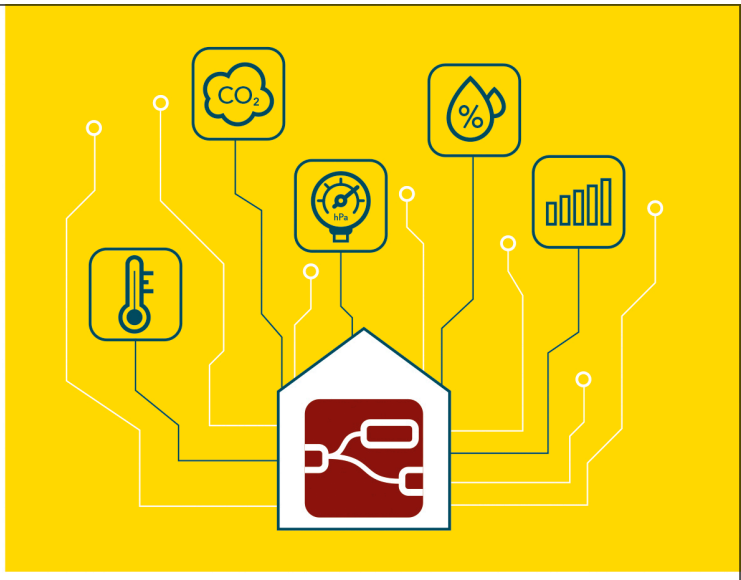
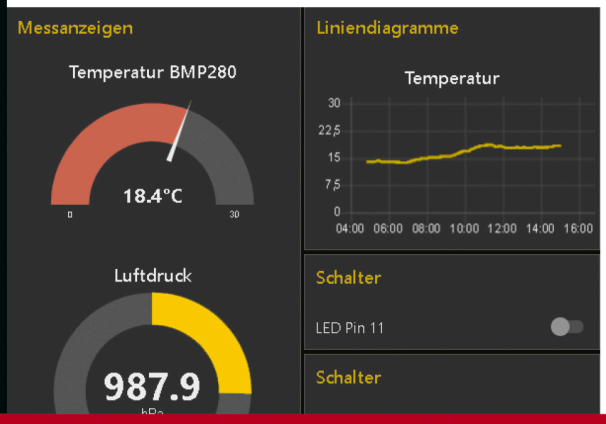
Udo Brandes

Flow, JSONat
ngrok, funct
Context, MSQ
Raspberry Pi
inject, Node





Node-RED

Das umfassende Handbuch

- Von den Grundlagen bis zum fertigen Dashboard
- Steuerungen und Logik visuell programmieren
- Daten verarbeiten in Maker- und IoT-Projekten und in der Hausautomation

4., aktualisierte Auflage

Beispiel-Flows und Projektdateien zum Download

Rheinwerk Computing

Kapitel 4

Das Node-RED-Dashboard

Eine Datenverarbeitung ohne visuell ansprechende Darstellung der Ergebnisse ist heutzutage kaum mehr vorstellbar. Node-RED unterstützt Sie hier in gewohnter Weise mit einfachen Nodes, die aber auch komplexe Anforderungen abdecken können. Möglich ist (fast) alles.

Das Dashboard von Node-RED ist eine mächtige grafische Benutzeroberfläche (*User Interface*, UI). Sie fügt sich lückenlos in das Node-RED-Konzept ein, da die einzelnen Elemente durch Nodes repräsentiert werden.

Dieses Kapitel führt auf Basis der Grundlagen aus Kapitel 3 und kleiner Anwendungen mit dem Raspberry Pi in die Anwendung einfacher Dashboard-Darstellungen ein. An geeigneter Stelle, insbesondere in Kapitel 11, erfolgt dann noch einmal eine detailliertere Behandlung anhand ausgewählter weiterreichender Beispiele.

Die Sache hat aber einen nicht unerheblichen Pferdefuß: Das originale Dashboard von Node-RED basiert auf Angular v1.0, das offiziell veraltet ist – es funktioniert, aber keiner gibt eine Garantie, dass es noch lange seinen Dienst tun wird. Bislang hat das Node-RED-Entwicklerteam noch keinen Nachfolger vorgestellt.

Es gibt allerdings mit Node-RED Dashboard 2.0 eine Alternative: Es ist als Open-Source-Projekt angelegt. Dahinter steht jedoch die Firma *FlowFuse* (*https://flowfuse.com*). FlowFuse wurde 2021 von Nick O'Leary, dem Co-Gründer von Node-RED ins Leben gerufen. Damit bestehen recht gute Aussichten, dass der kostenlose Zugang zu den neuen Dashboard-Nodes erhalten bleibt, auch wenn Sie von einer Firma bereitgestellt werden, deren Geschäftszweck natürlich darin besteht, solche Dienste gegen ein Entgelt anzubieten.

Für dieses Buch möchte ich Ihnen zunächst das originale Node-RED Dashboard vorstellen, das noch immer weit verbreitet ist. Abschnitt 4.8 widmet sich dann dem Dashboard 2.0. Sie können beide parallel betreiben.

4.1 Installation

Die Installation des Dashboards erledigen Sie mit dem Palettenmanager (**Hauptmenü • Palette verwalten • installieren**). Suchen Sie nach »dashboard«. Die Liste enthält mehr als 40 Treffer. Als Grundlage für die Beispiele in diesem Buch verwende ich aus den oben genannten Gründen das Paket *node-red-dashboard*.

Nach der Installation verfügt die Palette über zwei neue Gruppen (*dashboard* und *Dashboard*) mit insgesamt 16 Nodes sowie drei weiteren Konfigurations-Nodes. Tabelle 4.1 liefert Ihnen einen Überblick.

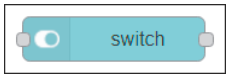
Node	Funktion
	Eine Schaltfläche. `msg.payload` enthält den Inhalt des Felds *Payload*.
	Ein Drop-down-Auswahlfeld. Mehrere Wert-Label-Paare können nach Bedarf hinzugefügt werden.
	Ein Schalter. Statusänderungen erzeugen eine msg.payload mit den angegebenen On- und Off-Werten.
	Ein Schieberegler. Die Werte ändern sich zwischen den Grenzen `min` und `max`.
	Ein Feld für die Eingabe eines Zahlenwerts.
	Ein Feld für die Eingabe einer Zeichenkette.
	Ein Dialog für die Datumsauswahl.
	Ein Dialog für die Farbauswahl. Er verbindet Eingaben mit gleichen Farbwerten zu Gruppen.
	Ein Feld für die Formulareingabe.
	Ein Feld für die Textausgabe.
	Eine analoge Messgeräteausgabe. Der Node stellt eine Ausgabe wie bei einem Messinstrument dar.
	Eine Diagrammausgabe.
	Eine Audioausgabe. Der Node spielt Audiodateien oder Text-to-Speech (TTS) ab.

Tabelle 4.1: Die Dashboard-Nodes

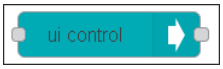
Node	Funktion
	Ein allgemeine Benachrichtigung. Der Node zeigt Dialogmeldungen an.
	Der Node konfiguriert eine dynamische, d. h. zur Laufzeit veränderbare Dashboard-Steuerung.
	Eine Vorlage für dynamische Benutzeroberflächenelemente.

Tabelle 4.1: Die Dashboard-Nodes (Forts.)

4.2 Browseraufruf und Einstellungen

Das Dashboard erreichen Sie im Browser über die IP-Adresse des Node-RED-Editors. Ergänzen Sie zusätzlich noch die Pfadbezeichnung *ui*.

Der vollständige Aufruf lautet daher *http://<IP-Adresse-Node-RED>/ui*, also z. B.:

```
http://192.168.1.210:1880/ui
```

Die Pfadbezeichnung können Sie in der Datei *settings.js* ändern:

```
ui: { path: "ihr_neuer_Pfad" },
```

Das Dashboard können Sie entsprechend dem *http-in*-Node in der Konfigurationsdatei *settings.js* mit `httpNodeAuth` absichern, damit der Zugriff nur verschlüsselt möglich ist. Wie das geht, erkläre ich in Abschnitt 1.6, »Node-RED absichern«.

4.3 Der Schnelleinstieg: So erstellen Sie Ihre erste Dashboard-Ausgabe

Das Dashboard weist eine bestimmte Hierarchie auf:

- **Basis**: Definiert die Basis-URL (z. B. */ui*) für Ihr Dashboard.
- **Seite**: Eine bestimmte Seite, zu der ein Besucher navigieren kann.
- **Gruppe**: Eine Sammlung von Widgets, die auf eine Seite gerendert ist.
- **Widget**: Ein einzelnes Widget (z. B. Diagramm, Schaltfläche, Formular), das im Dashboard dargestellt wird.

Die Konfiguration eines Dashboards kann recht umfassend und komplex sein. Beginnen Sie deshalb mit einem kleinen Beispiel, um ein Gefühl für die Darstellung zu bekommen.

Unser Ziel ist, über ein Dashboard per Knopfdruck einen Gruß abzusetzen. Die Anzeige des Grußes soll nach zwei Sekunden wieder erlöschen.

Erstellen Sie dazu einen Flow mit drei Nodes gemäß Abbildung 4.1.

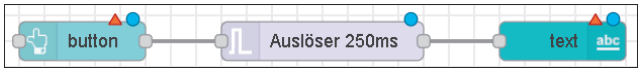


Abbildung 4.1: Der Flow, der einen Gruß absenden soll

Rote Dreiecke oberhalb der Nodes zeigen an, dass Sie noch weitere Einstellungen vornehmen müssen.

Schritt 1: Den button-Node konfigurieren und eine Dashboard-Gruppe erstellen

Öffnen Sie die Maske mit den Node-Eigenschaften (siehe Abbildung 4.2).

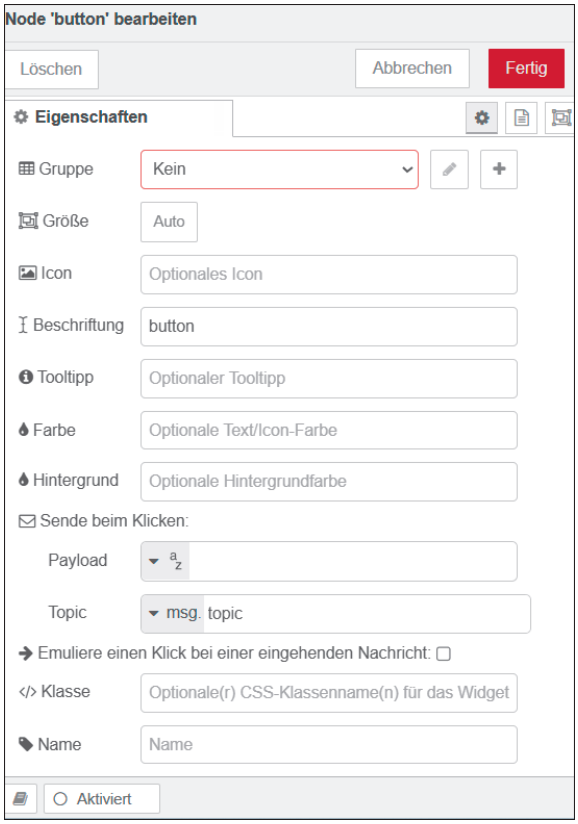


Abbildung 4.2: Die Eigenschaften des button-Nodes

Sofort sticht das rot umrandete Feld **Gruppe** ins Auge. Hier legen Sie die Zugehörigkeit des Nodes zu einer Dashboard-Gruppe fest. Eine Dashboard-Seite enthält eine oder mehrere Gruppen (weitere Informationen dazu finden Sie in Abschnitt 4.4).

Üblicherweise enthält die Drop-down-Liste alle bereits definierten Gruppen zur Auswahl. Die Liste ist zu Beginn selbstverständlich leer. Erstellen Sie daher eine neue Gruppe durch Klick auf das Stift-Icon. Sie gelangen zu einem weiteren Fenster, in dem Sie die Dashboard-Gruppe konfigurieren können (siehe Abbildung 4.18).

Schritt 2: Den Dashboard-Tab festlegen

Eine wesentliche Eigenschaft einer Dashboard-Gruppe ist die Zugehörigkeit zu einem Dashboard-Tab (siehe Abschnitt 4.4). Auch hier haben Sie die Möglichkeit, einen vorhandenen Dashboard-Tab auszuwählen oder zu einem Fenster für die Neuerstellung zu wechseln. Die Liste ist zu Beginn ebenfalls leer, sodass Sie durch Klick auf das Stift-Icon ein weiteres Fenster (siehe Abbildung 4.17) zur Konfiguration eines neuen Tabs aufrufen müssen.

Lassen Sie die Einstellungen unverändert und gehen Sie mit **Hinzufügen** zurück. Der Dashboard-Tab steht nun fest (siehe Abbildung 4.3, **Tab: Home**).

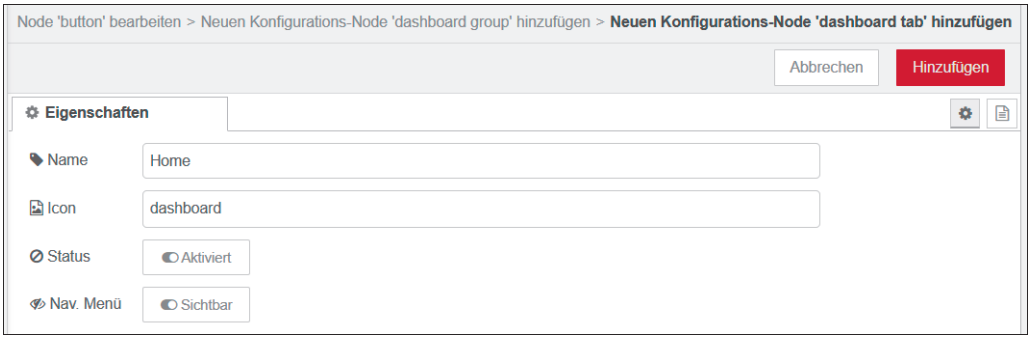


Abbildung 4.3: Konfigurieren Sie den groupconfig-Node.

Lassen Sie erneut die Einstellungen unverändert und gehen Sie mit **Hinzufügen** zurück. Nun haben Sie auch die Dashboard-Gruppe definiert. In Abbildung 4.4 sehen Sie das an **Gruppe: [Home] Standard**.

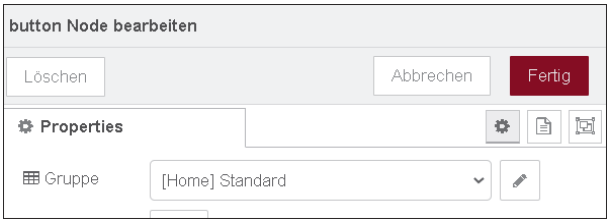


Abbildung 4.4: Der button-Node ist konfiguriert.

Fügen Sie vor dem Klick auf **Fertig** bei **Payload** noch »Hallo Welt« ein.

Die roten Dreiecke sind jetzt verschwunden, weil Node-RED die getroffenen Einstellungen zur Dashboard-Ausgabe auch auf den *text*-Node übertragen hat.

Schritt 3: Den trigger-Node einstellen

Der *trigger*-Node hat in diesem Beispiel die Aufgabe, die eingehende Nachricht weiterzuleiten und nach zwei Sekunden eine Löschnachricht hinterherzuschicken. Abbildung 4.5 zeigt die Eigenschaften des *trigger*-Nodes, die Sie für unser Beispiel brauchen.

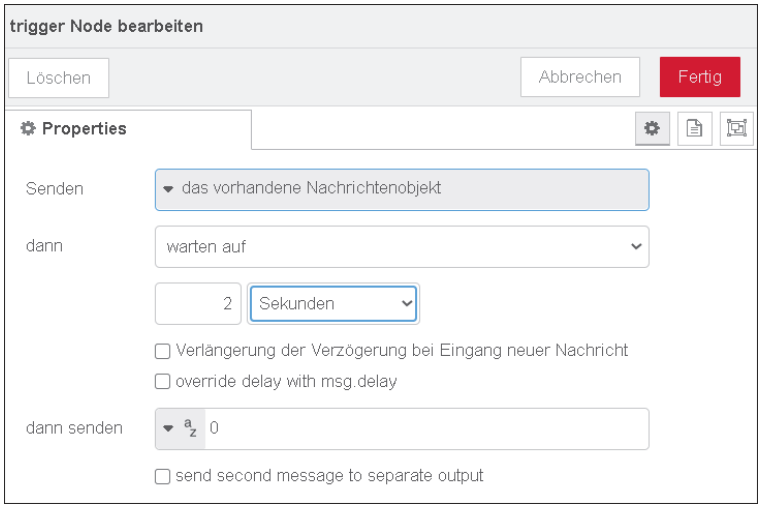


Abbildung 4.5: Die Einstellungen des trigger-Nodes

Schritt 4: Die Dashboard-Ausgabe starten

Klicken Sie nun auf **Deploy** und öffnen Sie das Dashboard in einem Browser, um den Flow zu testen. Wie Abbildung 4.6 zeigt, sehen Sie einen Button, mit dem Sie einen kurzen Text produzieren können.

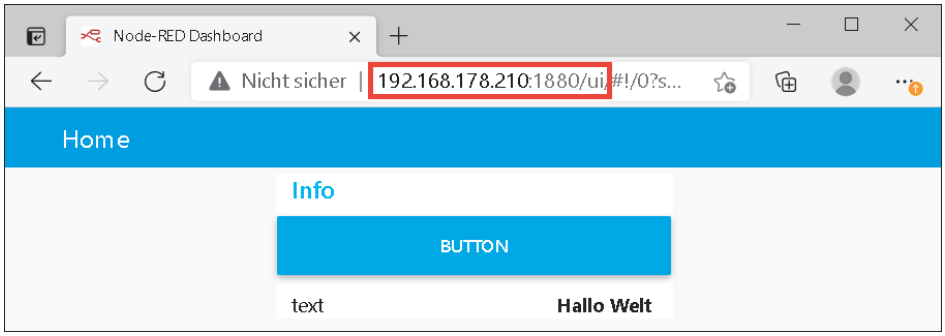


Abbildung 4.6: Flow-Gruß, Dashboard-Ausgabe

4.4 Das Dashboard-Design bestimmen

Die erste Dashboard-Ausgabe ist nur ein erster Blick auf die weite Spielwiese des Dashboard-Designs. Die folgenden Abschnitte sollen wesentliche Aspekte aufgreifen und Ih-

ren Werkzeugkasten für eigene Entwicklungen anreichern. Aber was ist das Dashboard eigentlich?

Mit dem Browseraufruf (siehe Abschnitt 4.2) gelangen Sie gewissermaßen zur Startseite Ihrer Webanwendung. Ihre Website kann aber mehrere Seiten haben. In Node-RED heißen diese Seiten *Tabs* (siehe auch Abschnitt 4.4.2). Das Layout einer Seite orientiert sich an einem Gitter bzw. Zeichenraster.

Eine Seite gliedert sich in Spalten (sogenannte *Gruppen*, siehe auch Abschnitt 4.4.2). Die Spaltenbreite beträgt per Voreinstellung sechs Rasterkästchen (siehe Abbildung 4.7), und ein Rasterkästchen hat per Voreinstellung eine Größe von 48 × 48 Pixeln.

Abbildung 4.7: Die Aufteilung des Node-RED-Dashboards

Die grafischen Erscheinungsbilder der Nodes werden *Widgets* genannt. In einer solchen Gruppe befinden sich die Widgets dann wie in einer Liste.

Gruppen und Widgets orientieren sich in Größe und Platzierung an diesem Raster.

Darüber hinaus kennt das Node-RED-Dashboard noch ein weiteres Element, das *Link* heißt. Schließlich können auch Icons Verwendung finden.

Demnach versteht man in Node-RED unter dem Begriff *Dashboard* also alle einzelnen Tabs und Links Ihrer Webanwendung.

Das Design Ihres Dashboards legen Sie auf zwei Wegen fest:

- über die rechte Seitenleiste (siehe Abschnitt 4.4.2)
- über Konfigurations-Nodes für Tabs und Gruppen (siehe Abschnitt 4.4.4)

4.4.1 Icons

Einige Widgets (z. B. das Button-Widget, siehe Abbildung 4.2) oder Tabs (siehe Abbildung 4.17) lassen sich mit Icons aufwerten. Sie können vier Icon-Sets einsetzen:

- **Font Awesome 4.7** (*https://fontawesome.com/v4.7.0/icons*)
  Ergänzen Sie beispielsweise die Eigenschaft **Icon** des *button*-Nodes aus dem vorherigen Beispiel um das Icon *User-Circle*, um eine ansprechendere Gestaltung der Schaltfläche zu erreichen (siehe Abbildung 4.8). Der passende Eintrag lautet `fa-user-circle`.

Abbildung 4.8: Button-Widget mit Icon

- **Angular Material Icons** (*https://klarsys.github.io/angular-material-icons*)
  Setzen Sie für eine ansprechendere Gestaltung des Dashboard-Menüeintrags aus Abbildung 4.9 beispielsweise die Eigenschaft **Icon** des *tabconfig*-Nodes auf das Icon *Home*. Der Eintrag lautet `home`.

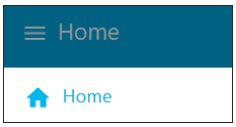


Abbildung 4.9: Dashboard-Menü-Icon

- **Weather Icons Lite** (*https://github.com/Paul-Reed/weather-icons-lite/blob/master/css_mappings.md*)
  Hier finden Sie spezielle Wetter-Icons, z. B. `wi-darksky-cloudy`.
- **Material Design Iconfont** (*https://jossef.github.io/material-design-icons-iconfont*)
  Für dieses Icon-Pack benötigen Sie eine Internetverbindung, damit das Icon auch angezeigt wird. Wenn Sie beispielsweise ein Wecker-Icon suchen, nutzen Sie den Eintrag `mi-alarm_on`.

4.4.2 Die rechte Seitenleiste

Die rechte Seitenleiste enthält ein weiteres Icon namens **Dashboard**. Der Eintrag ist in drei Registerkarten unterteilt (siehe Abbildung 4.10):

- **Layout**
- **Site**
- **Theme**

Abbildung 4.10: Das Icon »Dashboard« und seine Registerkarten

Das -Icon auf der rechten Seite blättert unmittelbar das Dashboard auf.

Layout

Die Registerkarte **Layout** beginnt mit der Zeile **Tabs & Links** (siehe Abbildung 4.11). Die Icons in dieser Zeile expandieren die Listeneinträge bzw. klappen sie wieder zusammen oder erstellen einen neuen Tab (**+ Tab**) bzw. einen neuen Link (**+ Link**).

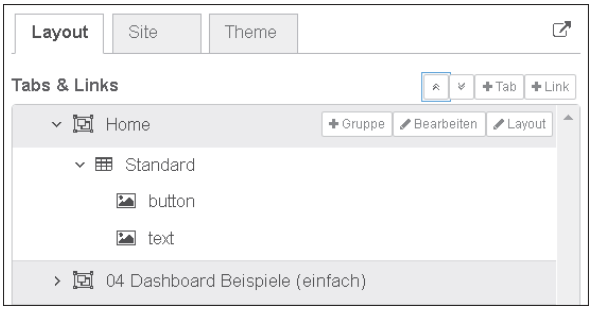


Abbildung 4.11: Dashboard-Einstellungen mit dem Tab »Layout«

Die Reihenfolge der Listeneinträge bestimmt die Darstellung im Dashboard. Der erste Tab der Liste ist der erste Menüeintrag des Dashboard-Menüs und gleichzeitig der Startbildschirm Ihrer Website. Die Gruppen unterhalb des Tabs sind von links nach rechts geordnet, die Widgets von oben nach unten. Die Reihenfolge ändern Sie per Drag-and-drop.

Abweichende Reihenfolge zwischen Editor und Dashboard

Es kann passieren, dass die Reihenfolge im Dashboard eine andere ist als die im Node-RED-Editor vorgegebene. Dies ist ein kleiner Bug in Node-RED. Verschieben Sie die Tabs im Node-RED-Editor erneut und deployen Sie die Änderungen. Der Fehler sollte dann behoben sein.

Sobald ein Listeneintrag mit dem Mauszeiger den Fokus erhält, erscheinen weitere Schaltflächen:

- bei Tabs – **+ Gruppe**, **Bearbeiten**, **Layout**
- bei Gruppen – **+ Abstand**, **Bearbeiten**
- bei Widgets – **Bearbeiten**

Bearbeiten führt zur Maske für die Einstellungen des jeweiligen Nodes.

Mit **+ Gruppe** (ebenfalls mit **+ Tab**) erstellen Sie eine neue Gruppe und gelangen in die Konfiguration der Gruppe bzw. des Tabs.

Ein Klick auf **Layout** öffnet den Dashboard-Layout-Editor (siehe Abbildung 4.12). Anpassungen sind hier übersichtlicher und einfacher zu bewerkstelligen als über die einzelnen Konfigurations-Nodes.

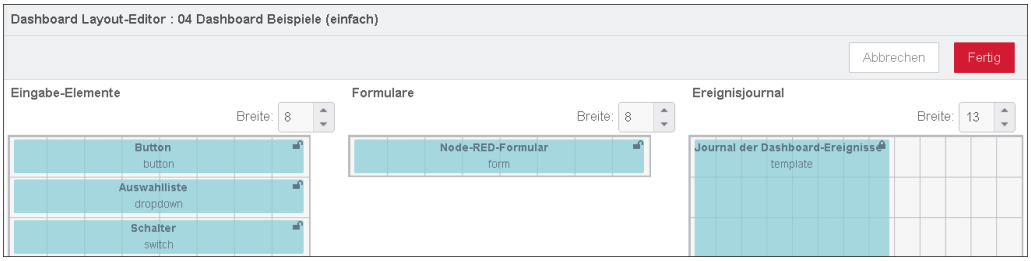


Abbildung 4.12: Der Dashboard-Layout-Editor

Das Layout des Dashboards

Der Layoutalgorithmus des Dashboards versucht immer, Elemente so hoch und so weit links wie möglich in ihrem Container zu verorten. Dies gilt sowohl für die Positionierung von Gruppen auf der Seite als auch für die Positionierung von Widgets in einer Gruppe.

Das führt bei einer Gruppe mit der Breite 6 mit sechs Widgets (jedes hat eine Breite von 2) zu einer Anordnung in zwei Zeilen mit jeweils drei Widgets.

Dagegen stehen alle sechs Widgets nebeneinander, wenn zwei Gruppen (Breite 6) zu je drei Widgets angelegt sind – zumindest solange Ihr Browserfenster breit genug ist. Wenn Sie den Browser verkleinern, verschiebt sich die zweite Gruppe irgendwann in eine Spalte unter die erste.

Im Hinblick auf eine dynamische Änderung der Seitengröße auf kleineren Bildschirmen sind mehrere Gruppen anstelle einer größeren sinnvoller.

+ Abstand fügt ein Abstands-Widget (*Spacer*) ein. Spacer helfen bei der Gliederung des Layouts. Die Größe ist per Default ein Rasterkästchen, lässt sich aber hinsichtlich der Anzahl horizontaler und vertikaler Elemente einstellen bzw. auf **AUTO** setzen.

+ Link fügt einen Link ein, den Sie über das Dashboard-Menü auswählen können. Abbildung 4.13 zeigt als Beispiel eine Verlinkung zur Node-RED-Homepage.

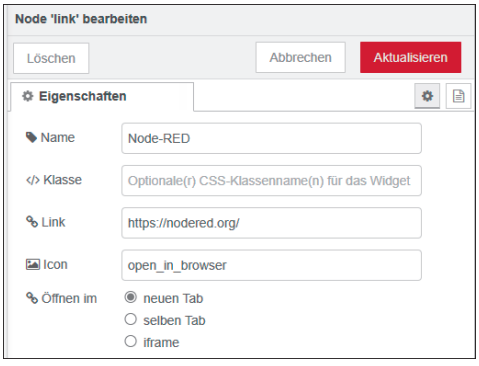


Abbildung 4.13: Einen link-Node definieren

Site

Die Registerkarte **Site** beinhaltet globale Layouteinstellungen Ihrer »Node-RED-Website«.

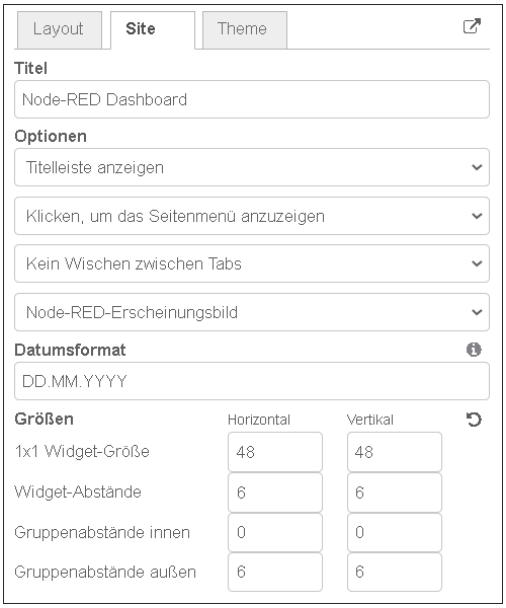


Abbildung 4.14: Dashboard-Einstellungen mit dem Tab »Site«

- **Titel** – Voreingestellt ist »Node-RED Dashboard«.
- Die **Optionen** betreffen die Anzeige der Titelleiste und die Anzeige der Seitenleiste, sofern Ihr Dashboard mehrere Tabs hat. Eine weitere Option gilt dem Wischen zwischen Tabs sowie möglichen Erscheinungsbildern (hierzu folgen weitere Informationen in Kapitel 11, »Dashboards für Fortgeschrittene«).
- **Datumsformat** – Voreingestellt ist das Format **DD.MM.JJJJ** (also z. B. 12.12.2026). Klicken Sie auf das ℹ-Icon, um sich weitere Varianten anzusehen.
- **Größen** – Mit den Vorgaben beeinflussen Sie die Rastergröße und die Platzierung von Widgets und Gruppen (siehe Abbildung 4.15).

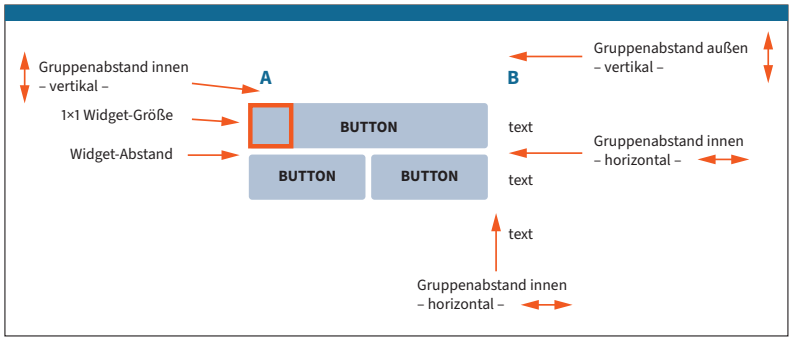


Abbildung 4.15: So definieren Sie die Größen im Dashboard.

- **1×1 Widget-Größe** – Bestimmt die Größe eines Rasterkästchens in Pixeln. Der kleinste Wert ist 24.
- **Widget-Abstände** – Legt den horizontalen und vertikalen Abstand zwischen zwei Widgets fest. Abstand 0 bedeutet, dass alle Widgets unmittelbar aneinanderstoßen.
- **Gruppenabstände** – Gruppenabstände betreffen die Platzierung der Gruppen.

Theme

Das Register **Theme** bietet Ihnen die Möglichkeiten aus Abbildung 4.16, um das Erscheinungsbild Ihres Dashboards einzustellen.

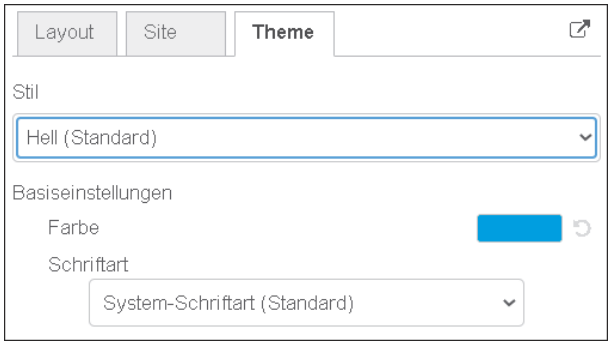


Abbildung 4.16: Gestaltungsoptionen mit dem Tab »Theme«

Wählen Sie bei **Stil** die Option **Dunkel**, wenn Sie eine augenfreundliche Erscheinung bevorzugen, oder definieren Sie sich Ihren eigenen Stil (**Benutzerdefiniert**).

Die Option **Farbe** bezieht sich auf die Farbe der Widgets. Sie können den Farbeditor nutzen und im RGB-Farbspektrum eine Farbe wählen. Mit Hilfe des Color-Picker können Sie direkt aus dem Bildschirm eine Farbe übernehmen.

4.4.3 Custom CSS

Bei einigen Nodes können Sie das Erscheinungsbild gezielt festlegen, z. B. bei dem *template*-Node, dem *chart*-Node, dem *notification*_node, dem *gauge*-Node, dem *dashboard-link*-Node und dem *groupconfig*-Node. Beispiele zum *template*-Node finden Sie in Kapitel 7 und in Kapitel 12.

4.4.4 Die Konfiguration von Tabs und Gruppen

Das Node-RED-Dashboard kennt für die Konfiguration von Tabs und Gruppen zwei Konfigurations-Nodes (Beiden sind Sie schon begegnet):

- den *tabconfig*-Node für Dashboard-Tabs
- den *groupconfig*-Node für Dashboard-Gruppen

Der tabconfig-Node

Abbildung 4.17 zeigt den Eingabebildschirm des *tabconfig*-Nodes.

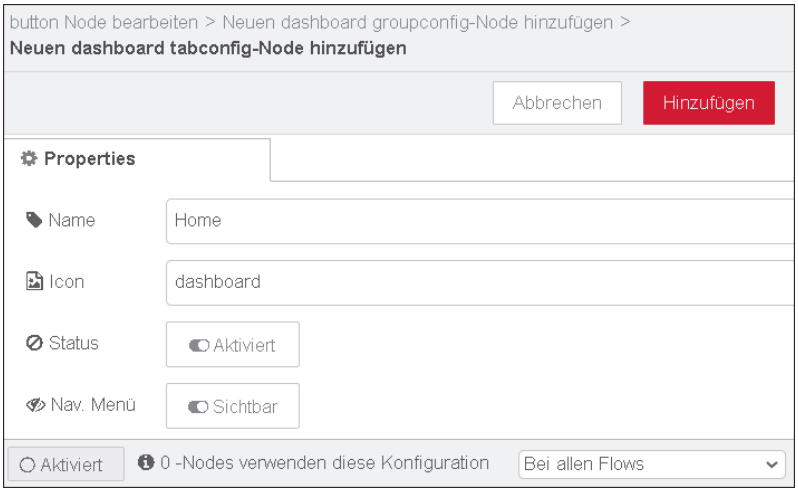


Abbildung 4.17: Die Eigenschaften von »dashboard-tabconfig«

Mit dem *tabconfig*-Node können Sie folgende Parameter für Ihr Dashboard setzen:

- **Name** ist die Bezeichnung des Tabs; sie findet sich in der Tab-Überschrift des Dashboards und in der Menüleiste.
- Das **Icon** (siehe Abschnitt 4.4.1) erscheint in der Menüleiste des Dashboards.
- Die Option **Status** bezieht sich auf die Anzeige des Tabs im Dashboard. Ist der Status **Deaktiviert**, erscheint der Tab in der Menüliste des Dashboards, lässt sich aber nicht mehr auswählen.
- **Nav. Menü** – Üblicherweise führt die Menüliste des Dashboards jeden Tab auf (**Sichtbar**). Die Option **Versteckt** blendet die Anzeige des Tabs im Dashboard aus.

Die Fußzeile erlaubt ein Aktivieren bzw. Deaktivieren des Nodes. Wenn Sie ihn deaktivieren, führt das dazu, dass kein Menüeintrag auf dem Dashboard vorhanden ist. Außerdem gibt es Informationen darüber, wie häufig Widgets diesen Konfigurations-Node nutzen.

Der groupconfig-Node

Die Konfiguration des *groupconfig*-Nodes sehen Sie in Abbildung 4.18.

Der *groupconfig*-Node erlaubt folgende Einstellungen:

- **Name** gibt die Bezeichnung der Gruppe an; sie ist die Spaltenüberschrift der Dashboard-Seite.
- **Tab** ist der Dashboard-Tab, dem die Gruppe zugeordnet ist.

- **Klasse** erlaubt mit Hilfe von CSS die Darstellung anzupassen
- **Breite** bezieht sich auf die zugebilligte Rasterbreite
- **Gruppenname anzeigen** sorgt dafür, dass die Spaltenüberschrift angezeigt wird.
- **Gruppenreduzierung zulassen** blendet die Widgets der Gruppe aus, die Spalte als solche bleibt jedoch in voller Breite sichtbar.

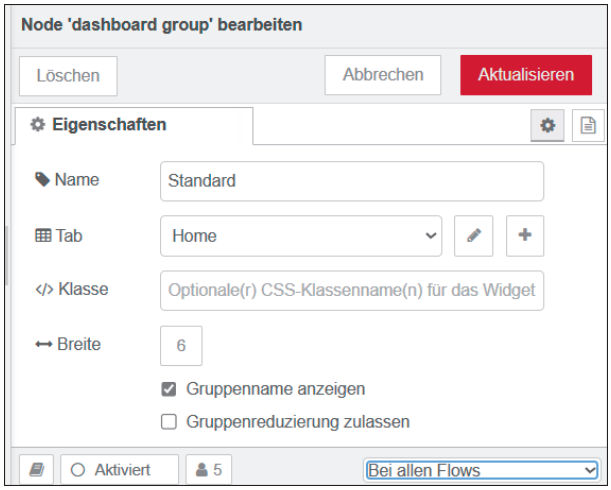


Abbildung 4.18: Die Eigenschaften von »dashboard-groupconfig«

Die Fußzeile erlaubt ein Aktivieren bzw. Deaktivieren des Nodes. Ein Deaktivieren hat zur Folge, dass die Gruppe nicht mehr im Dashboard angezeigt wird. Außerdem gibt es dort Informationen darüber, wie häufig Widgets diesen Konfigurations-Node nutzen. In einer Dropdown-Liste können Sie den Anwendungsbereich des Nodes auf einen speziellen Flow einschränken.

4.5 Die Dashboard-Widgets in Aktion

In diesem Abschnitt möchte ich Ihnen anhand eines kleinen Beispiels Dashboard-Eingabe-Nodes bzw. -Widgets vorstellen. Den Beispiel-Flow sehen Sie in Abbildung 4.19. Er verwendet bereits den *function*-Node (siehe Abschnitt 5.1) und den *template*-Node (siehe Abschnitt 11.2), sodass an dieser Stelle lediglich die notwendigen Codezeilen stehen.

In diesem Flow sind alle Dashboard-Eingabe-Nodes zumindest einmal vertreten. Verschaffen Sie sich also eine Übersicht.

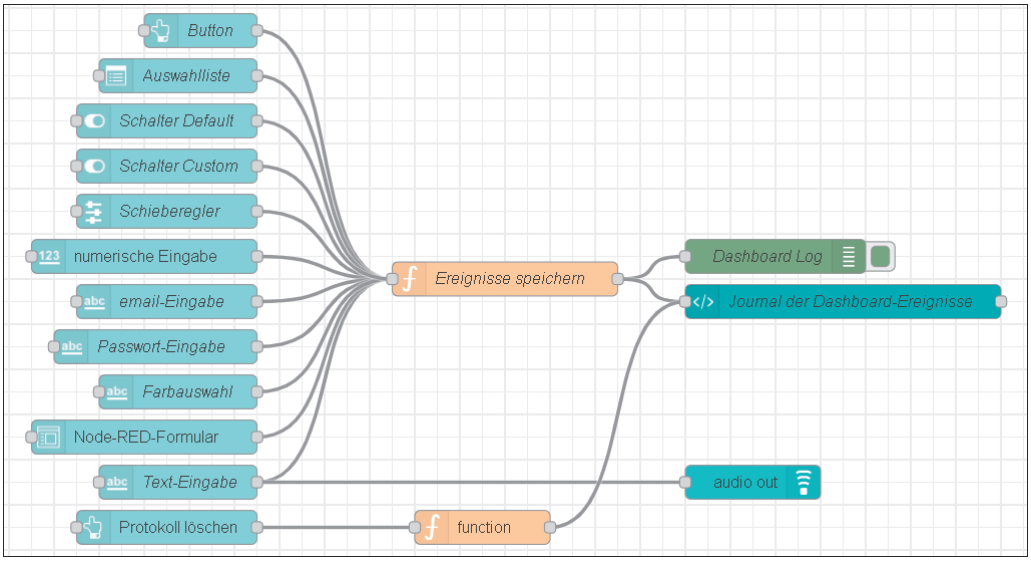


Abbildung 4.19: Ein Beispiel-Flow mit Dashboard-Eingabe-Nodes

4.5.1 Der button-Node

Der *button*-Node repräsentiert einen Taster. Bemerkenswert ist, dass er neben dem Ausgangsport auch über einen Eingangsport verfügt. Abbildung 4.20 zeigt die Konfigurationsmöglichkeiten.

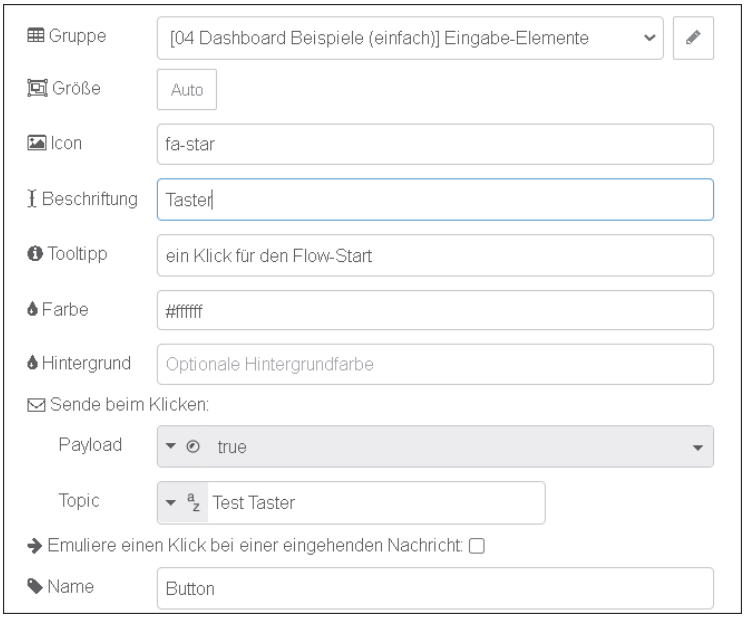


Abbildung 4.20: Die Eigenschaften des button-Nodes

Die Parametrisierungsoptionen sind folgende:

- **Gruppe** – bestimmt den Tab und die Gruppe, in der das Widget erscheinen soll.
- **Größe** – legt die Größe des Widgets in Rasterkästchen fest. **Auto** bedeutet eine Größe von Gruppenbreite × *n* Rasterkästchen. Innerhalb der Grenzen der Gruppengröße sind andere Maße möglich (z. B. 2 × 2).
- **Icon** – Einem Button lässt sich ein Icon zuordnen (siehe Abschnitt 4.4.1). In diesem Beispiel ist es das Icon *fa-star* aus der Sammlung *Font Awesome*.
- **Beschriftung** – Hiermit können Sie das Widget beschriften. Die Beschriftung erfolgt in Großbuchstaben.
- **Tooltipp** – Legen Sie hier fest, welche Meldung im Dashboard erscheint, wenn das Widget mit dem Mauszeiger den Fokus erhält.
- **Farbe** – Das ist die hexadezimal codierte RGB-Schriftfarbe des Widgets (z. B. entspricht #ffffff der Farbe Weiß).
- **Hintergrund** – Hier definieren Sie die hexadezimal codierte RGB-Hintergrundfarbe des Widgets.
- **Sende beim Klicken** – Hier legen Sie die Payload und das Topic fest, die gesendet werden, wenn der Button im Dashboard angeklickt wird.
  - **Payload** – Der Nachrichteninhalt kann mit Ausnahme von Umgebungsvariablen alle sonstigen Formate haben (Zeichenkette, Zeitstempel etc.).
  - **Topic** – das Nachrichtenthema.
- **Emuliere einen Klick …** – Üblicherweise löst ein Klick auf das Widget den Flow aus. Den gleichen Effekt erzielt der Eingang einer Nachricht am Eingangsport des *button*-Nodes, wenn dieses Kontrollkästchen aktiviert ist. Es handelt sich um sogenannte *programmgesteuerte Klicks*.
- **Name** – die Bezeichnung des Nodes im Node-RED-Editor.

4.5.2 Der dropdown-Node

Der *dropdown*-Node erzeugt ein Auswahllisten-Widget. Abbildung 4.21 zeigt die Einstellungsoptionen.

Folgende Optionen stehen Ihnen zur Verfügung:

- **Group** – Hier legen Sie den Tab und die Gruppe fest, in der das Widget erscheinen soll.
- **Size** – die Größe des Widgets in Rasterkästchen. **Auto** bedeutet eine Größe von Gruppenbreite × 1 Rasterkästchen. Innerhalb der Grenzen der Gruppengröße sind andere Maße möglich (auch z. B. ein Schrumpfen auf 2 × 1).
- **Label** – Das ist die Beschriftung des Widgets.
- **Tooltip** – Diese Meldung erscheint im Dashboard, wenn das Widget mit dem Mauszeiger den Fokus erhält.

- **Placeholder** – Der Platzhalter steht im Zusammenhang mit dem Kontrollkästchen **If msg arrives on input …**, mit dem Sie festlegen, ob die Message weitergeleitet werden soll. Bei aktiviertem Kontrollkästchen erscheint der Platzhalter im Dashboard. Eine Ausgabe am Ausgangsport unterbleibt jedoch.
- **Options** – enthält die Auswahlliste. Einzelne Einträge fügen Sie mit **+ option** hinzu. Jeder Listeneintrag besteht aus der zu sendenden Nachricht – dies kann eine Zeichenkette, eine Zahl oder ein boolescher Wert sein (z. B. »Hallo«) – und der Bezeichnung in der Auswahlliste (z. B. »Option 1: Gruß«).
- **Allow multiple selection** – erlaubt eine Mehrfachauswahl. `msg`-Payload ist dann eine Objekt-Tabelle.
- **If msg arrives on input …** – Diese Option müssen Sie aktivieren, wenn Sie einen Platzhalter nutzen wollen.
- **Name** – Bezeichnung des Nodes im Node-RED-Editor.

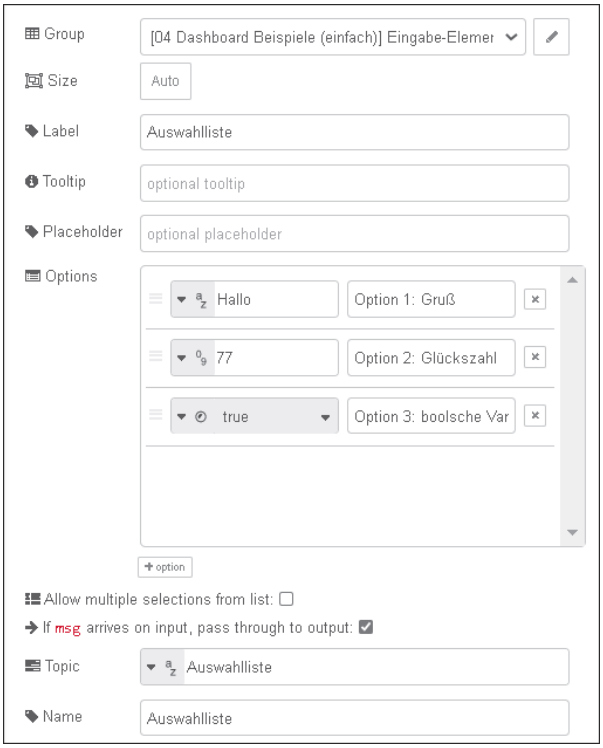


Abbildung 4.21: Die Eigenschaften des dropdown-Nodes

Im Flow-Design-Bereich des Node-RED-Editors erscheint die Nachricht der jeweils getätigten Auswahl (z. B. »Hallo«), sofern Sie die Ausgabe einer Statusmitteilung über **Hauptmenü • Einstellungen • Ansicht** aktiviert haben (siehe Kapitel 2, »Das zentrale Tool: der Node-RED-Editor«).

4.5.3 Der switch-Node

Der *switch*-Node erfüllt die Funktionalität eines Schalters. Abbildung 4.22 zeigt die Konfigurationsmöglichkeiten.

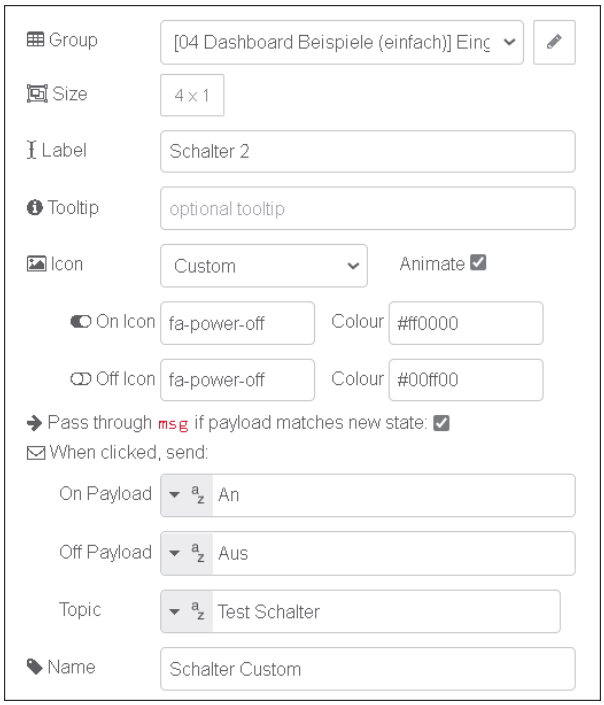


Abbildung 4.22: Die Eigenschaften des switch-Nodes

Die Parametrisierungsoptionen entsprechen in weiten Teilen denen beim *button*-Node:

- **Group** – der Tab und die Gruppe, in der das Widget erscheinen soll.
- **Size** – die Größe des Widgets in Rasterkästchen. **Auto** bedeutet eine Größe von Gruppenbreite × 1 Rasterkästchen. Innerhalb der Grenzen der Gruppengröße sind andere Maße möglich (z. B. 2 × 2).
- **Label** – Das ist der Text auf dem Widget. Er erscheint in Großbuchstaben.
- **Tooltip** – Diese Meldung erscheint im Dashboard, wenn das Widget mit dem Mauszeiger den Fokus erhält.
- **Icon** – Sie haben die Auswahl zwischen **Default** oder **Custom**, um das Icon (siehe Abschnitt 4.4.1) und die Farbe weiter anzupassen.
- **Pass through msg if payload …** – leitet die Nachricht vom Eingangs-Node weiter.
- **When clicked, send** – Bei dieser Option bestehen folgende Einstellungsmöglichkeiten:
  - **On Payload** – Der Nachrichteninhalt kann mit Ausnahme von Umgebungsvariablen alle sonstigen Formate haben (Zeichenkette, Zeitstempel etc.)

  - **Off Payload** – wie **On Payload**.
  - **Topic** – das Nachrichtenthema.
- **Name** – die Bezeichnung des Nodes im Node-RED-Editor.

4.5.4 Der slider-Node

Der *slider*-Node ist mit einem Schieberegler vergleichbar, der numerische Werte innerhalb eines definierten Bereichs ausgibt. Abbildung 4.23 zeigt die entsprechenden Node-Eigenschaften.

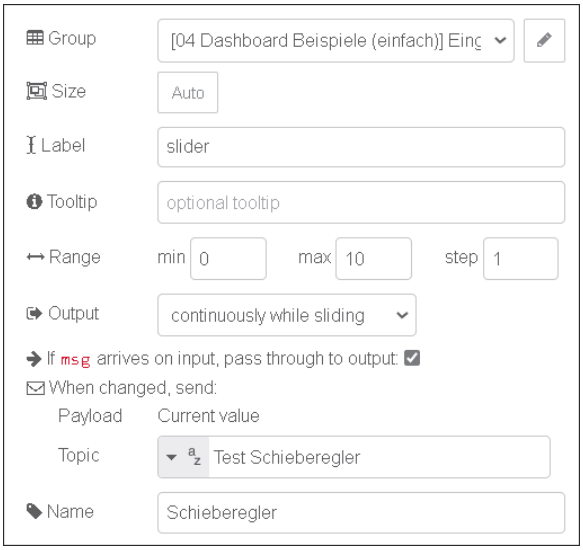


Abbildung 4.23: Die Eigenschaften des slider-Nodes

Als Node-Eigenschaften können Sie festlegen:

- **Group** – der Tab und die Gruppe für das Dashboard.
- **Size** – die Größe des Widgets in Rasterkästchen. **Auto** bedeutet eine Größe von Gruppenbreite × 1 Rasterkästchen.
- **Label** – der Text für das Widget. Er erscheint in Großbuchstaben.
- **Tooltip** – Diese Meldung erscheint im Dashboard, wenn das Widget mit dem Mauszeiger den Fokus erhält.
- **Range** – Das ist der Nummernbereich in einer Spanne von **min** bis **max** und einer Schrittweite von **step**.
- **Output** – bestimmt die Art der Ausgabe:
  - **continuously while sliding** – gibt die Nachricht bei jedem Schritt aus.
  - **only on release** – wartet auf das Ende des Schiebevorgangs und gibt erst dann die Nachricht aus.

- **If msg arrives on input …** – leitet die Nachricht vom Eingangs-Node weiter.
- **When changed, send:**
  - **Payload** – Der Nachrichteninhalt ergibt sich als numerischer Wert aus der Schieberegelposition und der Output-Einstellung.
  - **Topic** – das Nachrichtenthema.
- **Name** – die Bezeichnung des Nodes im Node-RED-Editor.

4.5.5 Der numeric-Node

Der *numeric*-Node weist viele Übereinstimmungen mit dem *slider*-Node auf, allerdings erfolgt die Auswahl der Zahl ähnlich einer Drop-down-Liste. Insofern sind auch die Parameter, die Sie in Abbildung 4.24 sehen, weitgehend identisch.

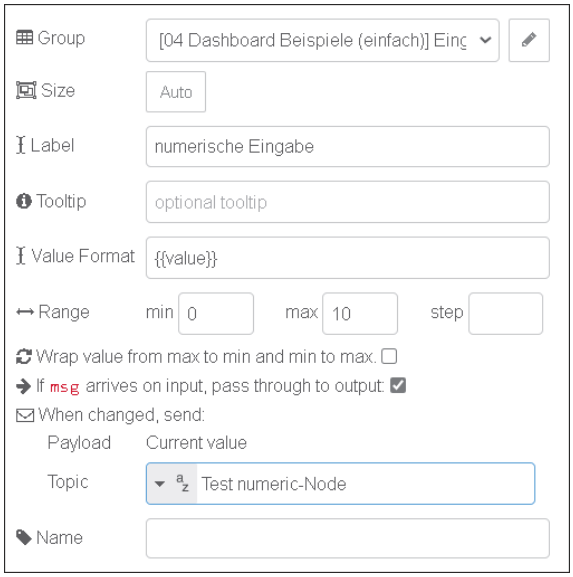


Abbildung 4.24: Die Eigenschaften des numeric-Nodes

Der wesentliche Unterschied zum *slider*-Node ist das zusätzliche Kontrollkästchen **Wrap value …** Wenn Sie es aktivieren, stoppt die Auswahl nicht an den Bereichsendpunkten, sondern folgt einem nicht enden wollenden Nummernkreis (z. B. folgt nach 10 wieder 0).

4.5.6 Der text-input-Node

Im Dashboard sind Texteingaben über den *text-input*-Node möglich. Abbildung 4.25 zeigt die Konfigurationsoptionen.

Der *text-input*-Node weist gegenüber den anderen Dashboard-Eingabe-Nodes eine spezifische Besonderheit auf, nämlich die Option **Mode** (siehe Abbildung 4.25). Der Beispiel-

Flow setzt den *text-input*-Node an vier Stellen ein, und zwar bei der E-Mail-Eingabe, der Passworteingabe, der Farbauswahl und der Texteingabe. Testen Sie die Wirkung der Modi im Dashboard.

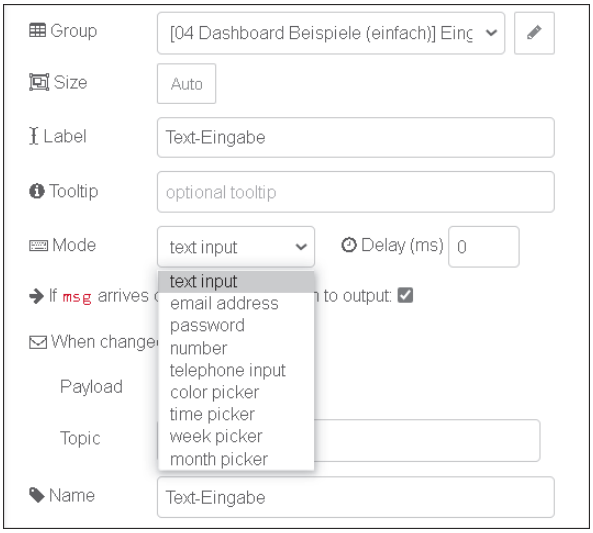


Abbildung 4.25: Die Eigenschaften des text-input-Nodes

Die Angabe einer Verzögerung bei **Delay** bewirkt, dass der eingegebene Text nicht mehr in seiner Gesamtheit ausgegeben wird, sondern in Zeichenfolgen, die jeweils alle bislang erfassten plus das aktuelle Zeichen umfassen (z. B. h, he, heu, heut, heute).

Im Beispiel-Flow habe ich einen *text-input*-Node mit einem *audio-out*-Node verbunden. Die Eigenschaften sehen Sie in Abbildung 4.26.

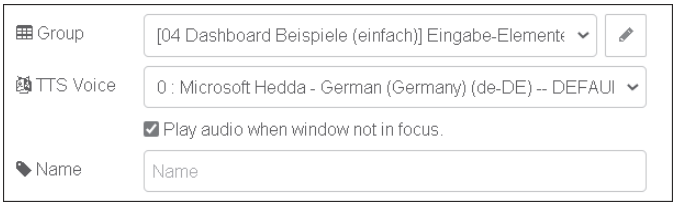


Abbildung 4.26: Die Eigenschaften des audio-out-Nodes

4.5.7 Der form-Node

Eingabeformulare gestalten Sie mit dem *form*-Node, dessen Optionen Sie in Abbildung 4.27 sehen.

Die Wirkungsweise der Optionen **Gruppe**, **Größe** und **Beschriftung** sowie **Topic** und **Name** ist identisch mit denen der bisher erörterten Nodes.

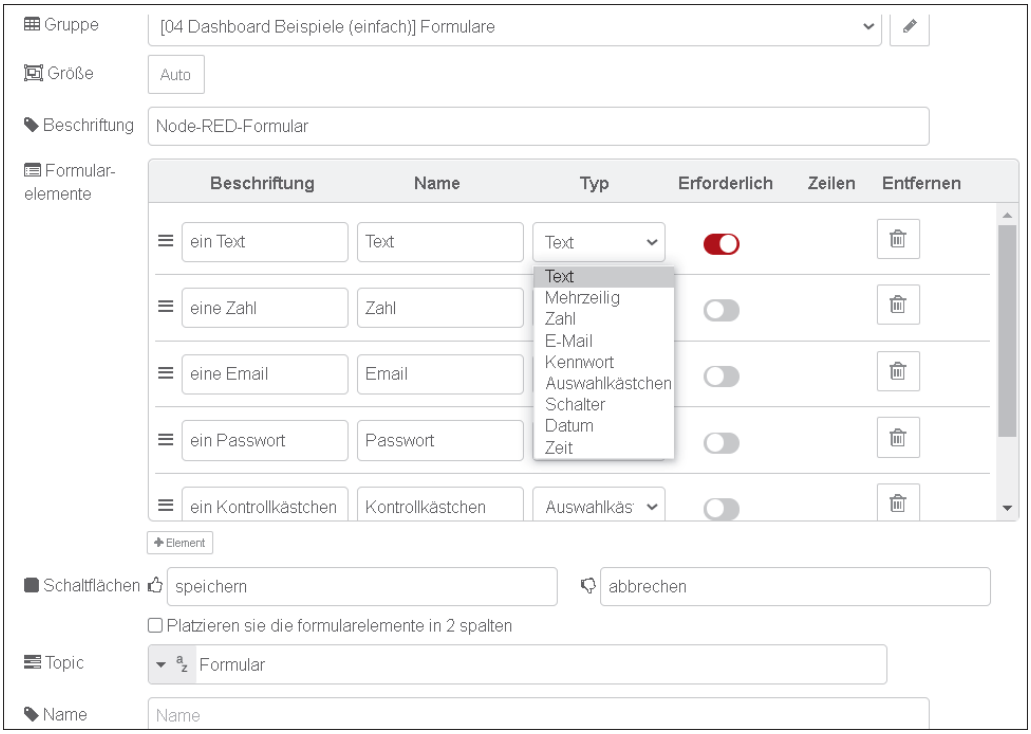


Abbildung 4.27: Die Eigenschaften des form-Nodes

Die Formularelemente sind in einer Liste organisiert. **+ Element** fügt ein Element hinzu; ein Klick auf das Papierkorb-Icon löscht den Eintrag.

Die Spaltenüberschriften sind weitgehend selbsterklärend. Das Aktivieren von **Erforderlich** hat zur Folge, dass ein Speichern nur bei gefülltem Dashboard-Listenelement möglich ist. Ein Stern neben der Dashboard-Beschriftung weist auf die Eigenschaft als Pflichtfeld hin (siehe Abbildung 4.28).

Ist der Elementtyp **Mehrzeilig**, können Sie unter **Zeilen** die Anzahl festlegen.

`msg.payload` ist hier ein Objekt, das die Listeneinträge enthält.

4.5.8 Die beiden function-Nodes und der template-Node

Der *function*-Node wird ausführlich in Kapitel 5, »Funktionen programmieren«, beschrieben, daher soll an dieser Stelle ein knapper Hinweis auf die Aufgaben der Nodes und die Angabe des Codes genügen.

Der erste *function*-Node (siehe Listing 4.1) fügt die `msg.payload` der eingegangenen Nachricht in einer Tabelle an, speichert die Tabelle und gibt sie zur Ausgabe weiter:

```
var dashboardLog = flow.get('dashboardLog')|| [];
dashboardLog.push(msg);
if (dashboardLog.length > 20){
    // Löschen der ältesten Nachricht bei mehr als 20 Einträgen
    dashboardLog.shift();
    dashboardLog.length = 20;
}
// Liste speichern
flow.set('dashboardLog',dashboardLog);
// Liste weitergeben
msg = {};
msg.payload = dashboardLog;
return msg;
```

Listing 4.1: Der erste function-Node

Der zweite *function*-Node löscht die gespeicherten Nachrichten und die Dashboard-Ausgabe:

```
flow.set('dashboardLog',[]);
msg = {};
msg.payload = [];
return msg;
```

Listing 4.2: Der zweite function-Node

Der *template*-Node sorgt für die aufbereitete Darstellung der erhaltenen Nachricht:

```
<ul>
 <li ng-repeat="x in msg.payload">
 <font color="red">{{x.topic}}</font>
    <ul>
        <li>{{x.payload}}</li>
    </ul>
 </li>
</ul>
```

Listing 4.3: Der template-Node

Abbildung 4.28 zeigt den Flow in Aktion und führt die gezeigten Nodes zusammen.

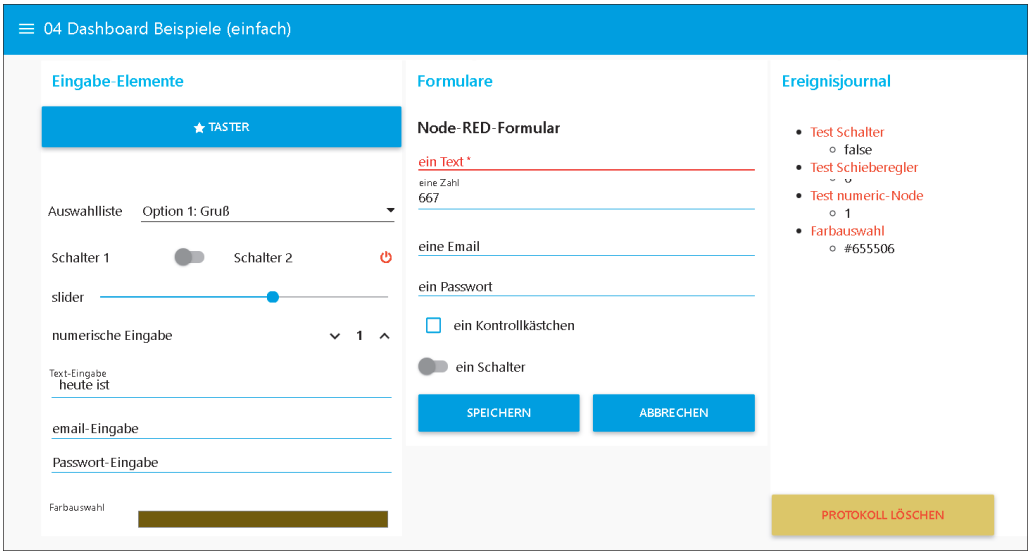


Abbildung 4.28: Der Beispiel-Flow im Dashboard

4.6 Charts und Messanzeigen mit dem Raspberry Pi

Häufig läuft Node-RED auf einem Raspberry Pi. Was liegt also näher, als mithilfe des Raspberry Pi eine kleine Indoor-Wetterstation zu realisieren? Das Beispiel umfasst auch, wie Sie die GPIOs des Raspberry Pi mit Node-RED steuern können.

Sie brauchen:

- einen Sensor BMP280 für die Temperatur- und Luftdruckmessung
- eine LED
- einen Widerstand (z. B. 470 Ω)
- ein Steckbrett und Jumper-Kabel

Abbildung 4.29 zeigt das Schaltbild, nach dem Sie den Sensor, die LED und den Raspberry Pi verkabeln müssen.

Da das Pin-out des Sensors abweichen kann, zeigt Tabelle 4.2 die Verdrahtung in einer Übersicht.

	RPi	BMP280	Farbe
GND	Pin 9 – GND	GND	Schwarz
VCC	Pin 1 – VCC	VCC	Rot

Tabelle 4.2: Die Verdrahtung des Pi mit dem BMP280

	RPi	BMP280	Farbe
SDA	Pin 3 – SDA1	SDA	Grün
SCL	Pin 5 – SCL1	SCL	Gelb

Tabelle 4.2: Die Verdrahtung des Pi mit dem BMP280 (Forts.)

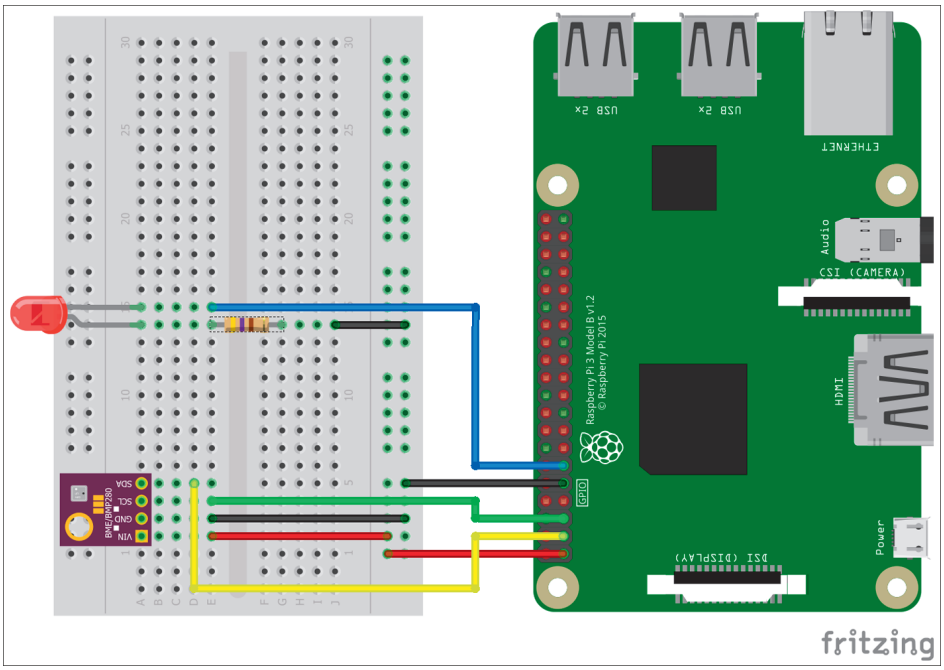


Abbildung 4.29: Schaltung für den Sensor BMP280

4.6.1 Sensoren schalten und das benötigte Paket installieren

Sie müssen zunächst den Sensor an den Raspberry Pi anschließen, dann das entsprechende Paket installieren und schließlich den Flow bauen. Gehen Sie das Projekt schrittweise an.

Schritt 1: Schaltung erstellen

Verbinden Sie die einzelnen Bauteile. Aber Achtung: Die Pin-Belegung des BMP280 kann vom Schaltplanbeispiel abweichen: SCL (BMP280) gehört an RPi-GPIO3 (SCL1), SDA (BMP280) an RPi-GPIO2 (SDA1).

Inhalt

Index

D

E

F

G

H

I

J

K

L

M

N

O

P

Q

R

S

T

U

V

W

X

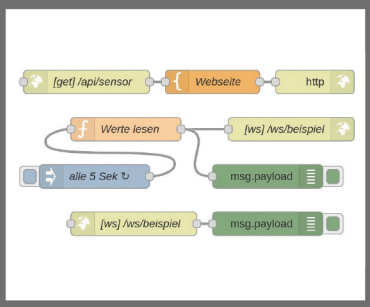
Z

Node-RED

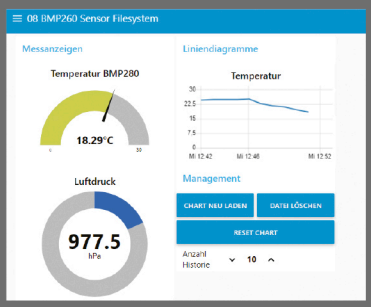
»Ohne Programmier-kenntnisse einsteigen«

Einfach visuell programmieren

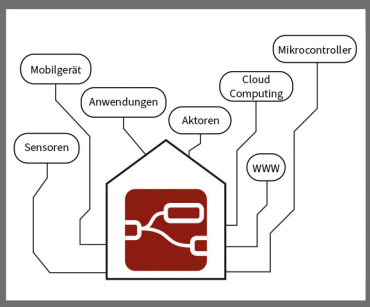
Mit dem grafischen Programmierkonzept von Node-RED bauen Sie Steuerungen einfach und ohne viel Code auf. Udo Brandes zeigt Ihnen, wie Sie Ihre Anforderungen übersichtlich modellieren und in strukturierte Abläufe überführen. So entstehen im Handumdrehen passende Dashboards und Automatisierungen für Ihre IoT- und Smart-Home-Projekte.



Flows aufbauen



Dashboards & Visualisierungen



Steuerung für Maker-Projekte

Unkompliziert loslegen

Nichts geht über KISS: Keep it Simple, Stupid. Und was könnte einfacher sein, als Programmierlogik direkt im Browser visuell zu erstellen? Vermeiden Sie Code-Chaos und bauen Sie schnell erste Tests und Prototypen.

Für Maker-Projekte und das Smart Home

Node-RED bietet Werkzeuge für die Heimautomation, für IoT-Prototyping oder Maker-Projekte. Speichern Sie Messwerte performant in InfluxDB ab, integrieren Sie Ihre FRITZ!Box und verarbeiten Sie Sensordaten mit dem Raspberry Pi.

Node-RED für Ihre Projekte

Dieser umfassende Einstieg begleitet Sie auch bei anspruchsvollen Szenarien: Programmieren Sie eigene Nodes, steuern Sie Ihr Setup mobil per App, gestalten Sie professionelle Dashboards, greifen Sie auf die ChatGPT-API zu oder programmieren Sie Mikrocontroller mit Node-RED.

Beispiel-Flows und alle Projektdateien zum Download

Udo Brandes beschäftigt sich mit der Heimautomation, Mikrocontrollern und dem großen Maker-Bereich. In diesem Handbuch zeigt er Ihnen, wie Sie mit Node-RED eigene Projekte umsetzen.

Aus dem Inhalt

- Installation und Grundlagen
- Administration und sichere Anbindung
- Der Editor von Node-RED
- Die Basics: Nodes und Flows
- Daten auf Dashboards anzeigen
- Eigene Dashboards erstellen
- Funktionen programmieren
- Grundlagenwissen zu JavaScript, Node.js und gutem Programmieren
- Daten abrufen und speichern
- Datenaustausch über MQTT und TCP/IP
- Hacks: fortgeschrittene Nodes nutzen
- App-Steuerung und Zugriff von unterwegs
- Eigene Nodes entwickeln
- Microcontroller steuern

Rheinwerk Computing

Gedruckt in Deutschland
Mineralölfreie Druckfarben
Zertifiziertes Papier

€ 39,90 [D] € 41,10 [A]

Smart Home
ISBN 978-3-367-10929-6