

CE

CONRAD

Alle Versuche im Überblick

Bevor es losgeht	3
Tag 1: Das NodeESP-Board	4
Möglichkeit 1: WPS	4
Möglichkeit 2: Konfiguration über die Webseite	4
Tag 2: OnBoard-LED und Updates	6
Tag 3: UDP und Packet Sender	8
Tag 4: UDP und JSON	10
Tag 5: Node-RED	12
Tag 6: Touch to Speach	14
Tag 7: Treppenhauslicht	15
Tag 8: Ein Regler	16
Tag 9: Ein Stimmungslicht.....	18
Tag 10: Deep Sleep.....	19
Tag 11: MQTT	21
Tag 12: MQTT-Button.....	23
Tag 13: Smarter Schalter für ein Soundboard.....	25
Tag 14: Smarte Lichter	27
Tag 15: Fernbedienungsempfänger	29
Tag 16: Universalfernbedienung.....	30
Tag 17: Zeitgesteuerte Lichter	33
Tag 18: Kerzenalarm	34
Tag 19: Cheerlights	35
Tag 20: Pflanzenwächter.....	36
Tag 21: Piano-Player	38
Tag 22: Klimawächter	39
Tag 23: Alarmanlage	41
Tag 24: Tannenbaumbeleuchtung	42

Bevor es losgeht ...

... noch ein paar Worte zu dem diesjährigen Adventskalender. Als ich gefragt wurde, ob ich Zeit und Lust hätte, den IoT-Kalender 2018 zu gestalten, habe ich begeistert zugesagt. Es hat mir sehr viel Spaß gemacht, die beiden Adventskalender von 2015 und 2016 zu gestalten. Schnell war für mich auch klar, dass es an der Zeit ist, ein neues Entwicklungsboard ins Spiel zu bringen, und zwar eines auf Basis des ESP32-Mikrocontrollers. Das war die Geburt des **NodeESP**-Boards.

Als ich anfing, die Arduino-Programme für das Board zu schreiben, ist mir eines schnell klar geworden: dass die Integration des ESP32 in Arduino noch im vollen Gange ist. Beinahe wöchentlich gab es Änderungen, und mehrere Male funktionierten Programme plötzlich nicht mehr wie gewünscht. Gleichzeitig wurden die Quelltexte immer länger und komplexer, da ich der Leistungsfähigkeit und dem Funktionsumfang des ESP32-Controllers gerecht werden wollte. Solche Komplikationen stellen natürlich ein sehr großes Risiko für eine gelungene Weihnachtszeit dar, sowohl für Sie als auch für mich. Deswegen musste eine bessere Lösung her.

Daher habe ich mich in diesem Adventskalender für einen etwas anderen Weg entschieden. In diesem Kalender wird der NodeESP als eine Art Blackbox betrachtet, die nur einige Schnittstellen bereitstellt. Die Funktionen werden einfach als Programm des Tages auf der Boardwebseite ausgewählt. Das Board öffnet dann die Kommunikationskanäle und reagiert auf vordefinierte Befehle. Damit es aber nicht zu langweilig wird, haben Sie dennoch die Möglichkeit, über das Programm **Node-RED** selbst kleine Programme zu erstellen.

Da es auch dabei zu Schwierigkeiten kommen kann und die Schnittstellen und Befehle selbst ebenfalls einiges an Dokumentation erfordern, gibt es auch in diesem Jahr wieder eine Webseite zu dem Adventskalenderprojekt. Auf www.iot.fkainka.de wird nicht nur täglich das Projekt des Tages vorgestellt, sondern Sie können mithilfe der Kommentarfunktion auch über das Projekt diskutieren oder Unterstützung bei Problemen bekommen. Außerdem finden Sie auf der Webseite den Flow des Tages, also den Programmablauf, den Sie einfach in Ihre Node-RED-Umgebung importieren können. Zusätzlich werden auf der Webseite einige weitere Informationen und Zusatzprojekte veröffentlicht, sodass es sich lohnt, auch nach der Weihnachtszeit mal vorbeizuschauen.

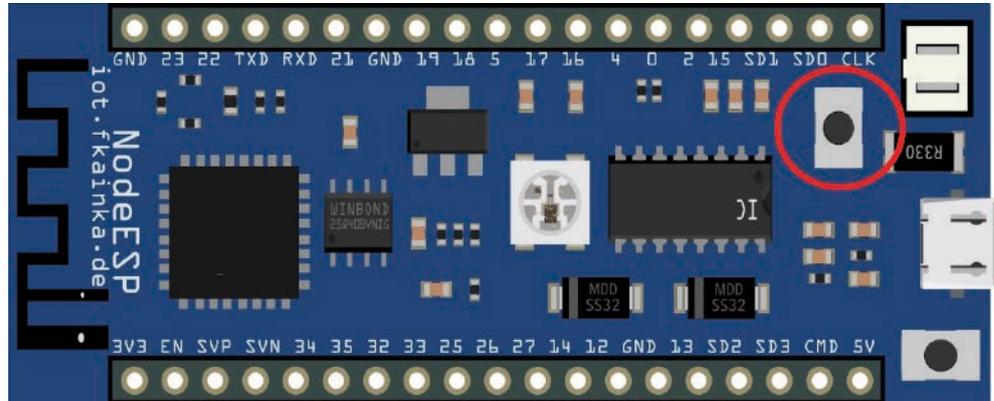
Ich hoffe, dass sich auf der Webseite auch in diesem Jahr viele technikbegeisterte Leser tummeln werden, die zusammen das Adventskalenderprojekt noch interessanter, lehrreicher und sozialer gestalten.

In diesem Sinne wünsche ich Ihnen eine frohe und spannende Weihnachtszeit!

Ihr Fabian Kainka

Tag 1: Das NodeESP-Board

Es ist der erste Tag des Adventskalenderprojekts 2018 und somit Zeit, das erste Türchen zu öffnen. Hinter dem Türchen finden Sie das Mikrocontrollerboard dieses Kalenders. Heute soll es vor allem darum gehen, eine Verbindung zum Heimnetzwerk herzustellen und das Board etwas besser kennenzulernen.



Das NodeESP-Board mit markierter WPS-Taste.

Der erste Schritt ist, das Board mit Strom zu versorgen. Sie können das Board einfach per MicroUSB-Kabel mit dem PC oder einem gängigen Smartphone-Ladegerät verbinden. Sie brauchen mindestens 300 mA, um das Board zu starten, ältere Laptops und USB-Hubs ohne externe Stromversorgung könnten problematisch sein. Die RGB-LED auf dem Board beginnt nach kurzer Zeit, gelb zu leuchten. Damit ist die Initialisierung abgeschlossen. Sie müssen nur noch Ihre WLAN-Daten eingeben. Dazu gibt es zwei Möglichkeiten:

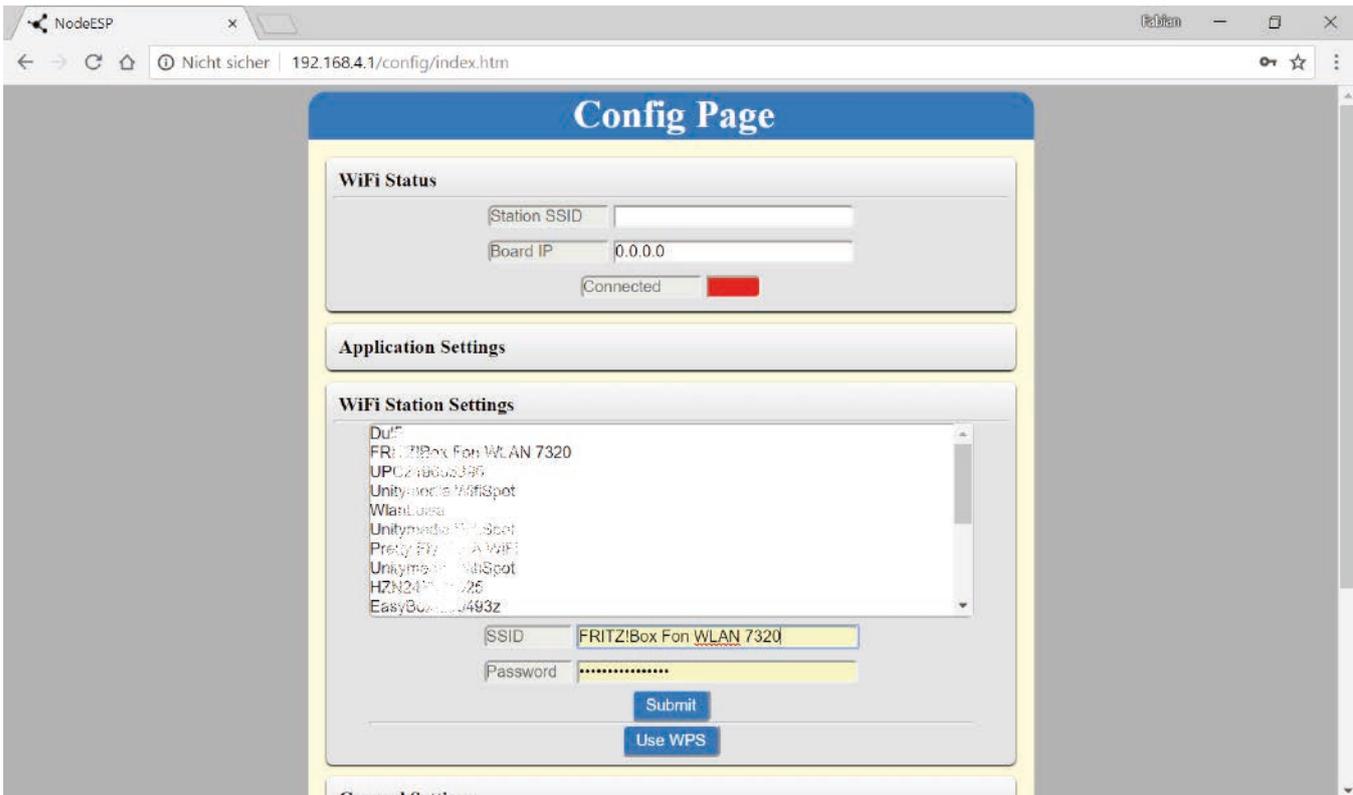
Möglichkeit 1: WPS

Vielleicht kennen Sie die WPS-Funktion bereits von anderen Geräten. Durch Drücken eines Buttons an Ihrem Router und an dem neu zu verbindenden Gerät tauschen beide die WLAN-Daten selbstständig aus. Auch das NodeESP-Board hat diese Funktion. Drücken Sie den Boot-Button (der Button nahe am Batterieanschluss, im Bild markiert) für ein paar Sekunden, und die LED blinkt gelb. Drücken Sie dann die WPS-Taste auf Ihrem Router. Nach einer kurzen Wartezeit startet der NodeESP neu. War der Verbindungsaufbau erfolgreich, leuchtet die LED nach kurzem Blinken dauerhaft grün. Ist das nicht der Fall, können Sie den Vorgang wiederholen oder die zweite Möglichkeit ausprobieren.

Möglichkeit 2: Konfiguration über die Webseite

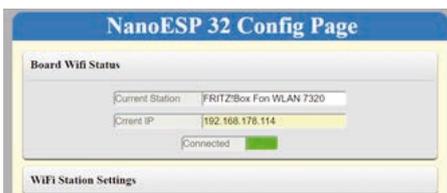
Die zweite Möglichkeit, eine Verbindung mit Ihrem Heimnetz aufzubauen, besteht darin, die vorprogrammierte Konfigurationswebseite zu benutzen. Dazu müssen Sie sich mit dem Hotspot des Boards verbinden. Es erzeugt nämlich in Ihrer Umgebung ein offenes WLAN-Netzwerk mit dem Namen **NodeESP**. Verbinden Sie sich nun mit einem PC oder Smartphone mit diesem Hotspot. Anschließend rufen Sie einen Browser auf und tippen die Adresse des Boards im Access-Point-Modus **http://192.168.4.1** ein. Es baut sich nun die im Board integrierte Webseite auf.

Auf dem kleinen Board läuft ein kompletter Webserver, und es gibt auch ein paar Seiten, die Sie besuchen können. Unter anderem finden Sie dort die Konfigurationsseite **Board Configuration**. Hier erhalten Sie unter anderem die Möglichkeit, Ihre WLAN-Daten einzugeben. Klicken Sie auf den Punkt **WiFi Station Settings**. Das Board scannt nach WLAN-Netzwerken in Reichweite. Nach kurzer Wartezeit können Sie Ihr Netzwerk in der Liste auswählen und in das untere Textfeld das Passwort eingeben.



Die Konfigurationsseite des Boards.

Die Verbindung zum Heimnetzwerk steht jetzt also. Nun können Sie die Boardwebseite ebenfalls über das Heimnetzwerk aufrufen. Die IP des Boards erfahren Sie auch über die Konfigurationsseite des Boards. Eventuell müssen Sie sich dazu erneut mit dem NodeESP-Netzwerk verbinden. Im Bereich **Board WiFi Status** finden Sie die IP-Adresse, die Ihr Board von Ihrem Router zugewiesen bekommen hat. Verbinden Sie sich nun wieder mit Ihrem Heimnetzwerk und rufen Sie diese IP im Browser auf. Im abgebildeten Beispiel ist das **http://192.168.178.114/**.



Die Konfigurationsseite des Boards.

Ein etwas praktischerer Weg führt über den Bonjour-Service von Apple. Dazu müssen Sie ein kleines Tool installieren, das Sie von der Webseite

<https://support.apple.com/de-de/bonjour>

herunterladen können.

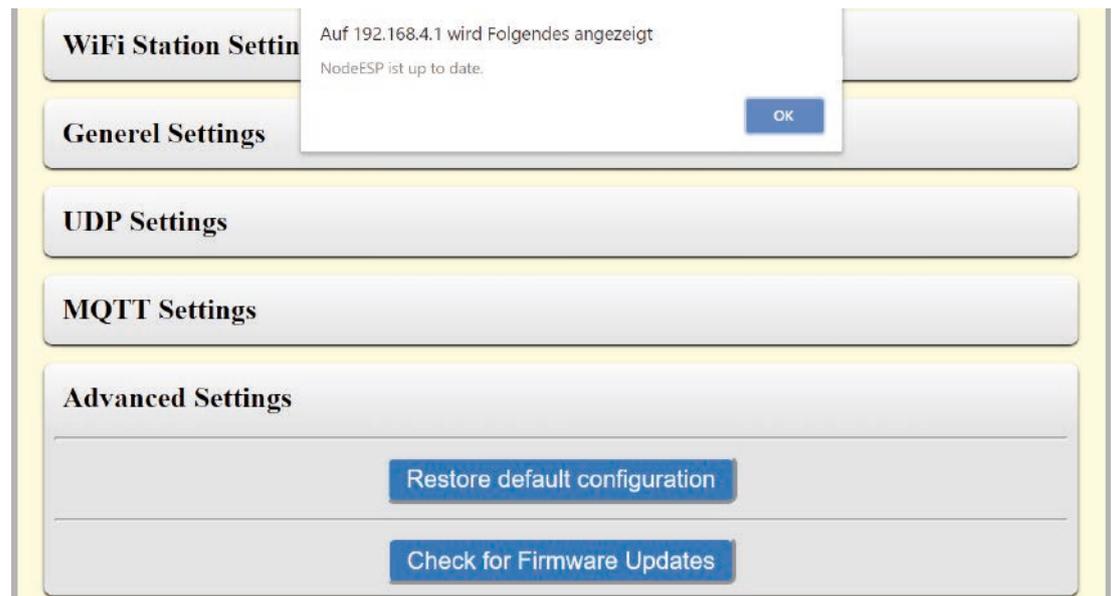
Hinweis: Diese Schritte gelten nur für Windows-Anwender. Apple- und Linux-Nutzer haben den Dienst bereits vorinstalliert. Es funktioniert nicht mit Android-Smartphones.

Nach der Installation tippen Sie im Browser einfach **http://nodeesp.local/** ein, um zu der Webseite des Boards zu gelangen. Sich die IP-Adresse zu merken ist dann überflüssig. Der dahinterstehende Service nennt sich übrigens **mDNS (multicast Domain Name System)** und funktioniert, weil das Board ständig seine IP und den Domainnamen an das Netzwerk sendet. Der mDNS-Name kann übrigens auch auf der Konfigurationsseite angepasst werden. Damit wäre eine Verbindung zum Heimnetzwerk hergestellt. Sollten Sie auf Probleme gestoßen sein, finden Sie auf der Projektseite www.iot.fkainka.de Hilfe.

Tag 2: OnBoard-LED und Updates

Hinter dem Türchen des zweiten Tages finden Sie ein Steckboard für die kommenden Versuchsaufbauten. Ich empfehle aber, Ihren NodeESP noch nicht auf das Steckboard zu stecken. Stattdessen schauen wir uns heute an, wie Sie die vorinstallierten Programme auf dem Board starten können, und bringen das Board außerdem auf den neuesten Stand.

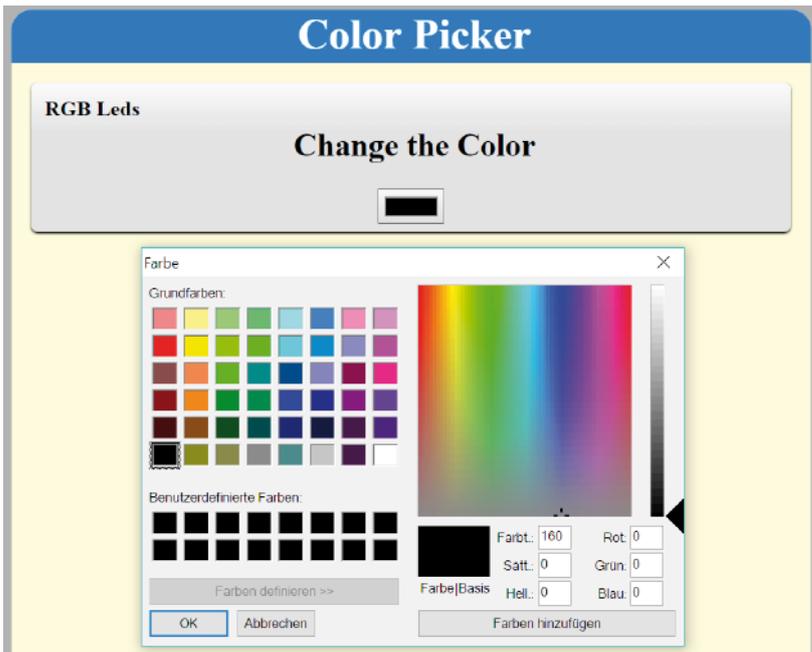
Verbinden Sie Ihr Board zunächst wieder mit einer Stromquelle. Die WLAN-Daten, die Sie gestern eingegeben haben, sind weiterhin gespeichert, und somit sollte sich das Board automatisch mit Ihrem WLAN verbinden. Rufen Sie dann die Boardseite über die IP oder den mDNS-Namen (**http://nodeesp.local/**) auf. Als Erstes soll nach möglichen Software-Updates gesucht werden. Dazu öffnen Sie die Konfigurationsseite, klicken ganz unten auf den Punkt **Advanced Settings** und dort auf **Check for Firmware Updates**.



Das Board ist auf dem neuesten Stand.

Wenn ein neues Update gefunden wurde, bekommen Sie auf der Konfigurationswebseite eine Meldung, die Sie bestätigen müssen. Danach können Sie sich erst mal einen Kaffee holen, denn der Updatevorgang kann eine ganze Weile dauern (bis zu 20 Minuten). Sobald das Board wieder grün leuchtet, können Sie weitermachen. Überprüfen Sie erneut, ob es Updates gibt. Im Normalfall meldet das Board nun, dass alles auf dem neuesten Stand ist. Sollte aber etwas grundsätzlich falsch gelaufen sein, finden Sie auf der Projektwebseite eine Anleitung dazu, wie Sie es reparieren können.

Doch nun zu den vorinstallierten Programmen: Auf der Konfigurationsseite sehen Sie den Punkt **Application Settings**. Dort finden Sie alle Programme und eine Infobox, die Ihnen sagt, welches gerade aktiv ist. Wählen Sie zunächst in der oberen Box **IoT Adventskalender 2018** aus und schließlich in der unteren das Programm des heutigen Tages (beginnt mit **IoT Calendar 2018 - Day 2**). Bestätigen Sie anschließend mit **Submit**. Das Board startet jetzt mit den ausgewählten Programm neu.

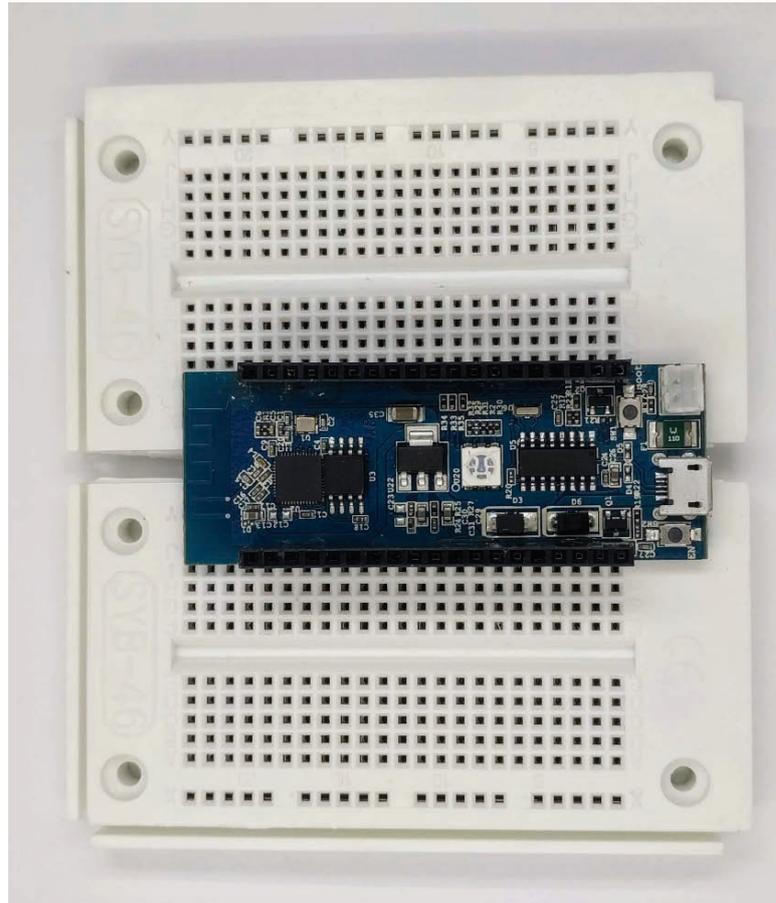


Das Color-Picker-Element.

Nach Neustart und Neuverbinden können Sie nun zur Hauptseite des Boards wechseln. Dort finden Sie einen weiteren Link mit der Bezeichnung **Applikation Page**. Wenn Sie diesem Link folgen, erreichen Sie eine Webseite mit einem sogenannten Color-Picker-Element, mit dem Sie Farben auswählen können. Die von Ihnen ausgewählte Farbe wird an das Board übertragen und von der OnBoard-LED dargestellt. Sie haben also eine kleine, praktische, bunte Lampe - und das ganz ohne dass Sie etwas zusammenstecken mussten.

Tag 3: UDP und Packet Sender

Hinter dem heutigen Türchen befindet sich ein weiteres Steckboard. Es lässt sich mit dem gestrigen Steckboard zusammenstecken. Das NodeESP-Board kann dann auf beiden Steckboards gleichzeitig platziert werden. Allerdings hat sich herausgestellt, dass diese Konstellation nicht bei jedem funktioniert. Die einfachste Lösung ist, die Seitenteile der Steckboards abzubrechen. Dazu habe ich die Boards ineinander gesteckt und einfach solange gebogen, bis ein Seitenteil abgebrochen ist. Mit dem anderen Board bin ich genauso verfahren. Mit einer Zange funktioniert das aber genauso gut. Ohne Seitenteile lassen sich die Steckboards näher beieinander platzieren, und es gibt keine Probleme mit dem Raster. Das NodeESP-Board stecken Sie dann einfach auf beide Boards, wie auf dem Bild zu sehen. So bleibt genügend Platz für weitere Bauteile. Im heutigen Projekt werden wir sehen, wie Sie vom PC aus eine Verbindung zum Board aufbauen können, und zwar ohne den Webbrowser zu verwenden.



Wählen Sie auf der Konfigurationsseite das Programm des heutigen Tages aus. Das Board startet daraufhin neu. Sie sind mit den Einstellungen allerdings noch nicht ganz durch. Auf der Konfigurationsseite gibt es einen weiteren Bereich, **UDP Settings**. Dort finden Sie den Punkt **IP (send)**. Unter Umständen ist diese IP bereits richtig, aber wahrscheinlich müssen Sie hier Ihre Broadcast-Adresse manuell eingeben. In den meisten Fällen brauchen Sie einfach Ihre lokale IP, die oben im Bereich **Board Wifi Status** steht, in das Feld zu kopieren und die letzte Zahl nach dem Punkt durch 255 zu ersetzen. Es gibt aber auch einige Heimnetze, die eine andere Struktur verwenden. Wenn es kompliziert wird, können Sie ein Onlinetool wie

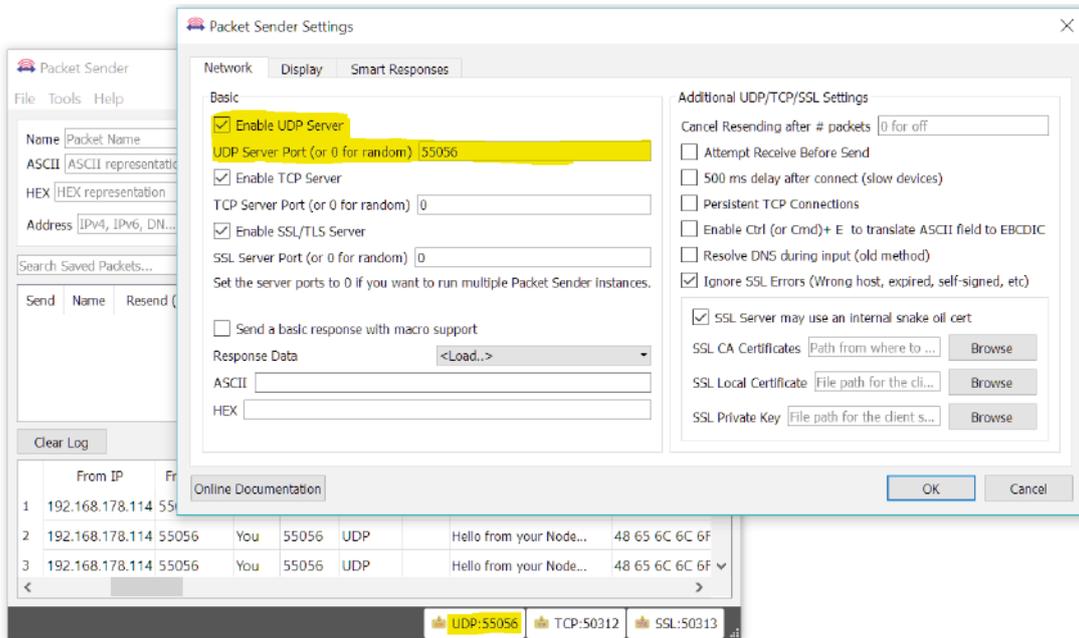
<https://remotemonitoringsystems.ca/broadcast.php>

einsetzen oder im Kommentarbereich zum heutigen Programm Hilfe suchen. Nach dem Bestätigen mit **Submit** startet das Board wie gewohnt neu und beginnt bereits, Nachrichten zu senden. Doch nun benötigen wir noch ein Programm auf dem PC, mit dem wir die Nachrichten empfangen können. Dafür empfiehlt sich das Tool **Packet Sender**, das für alle gängigen Betriebssysteme von der Seite

<https://packetsender.com/download>

heruntergeladen werden kann. Je nach Vorliebe verwenden Sie die Version zum Installieren oder die Portable-Version, die Sie direkt starten können. Nach dem Start fragt Windows eventuell nach den Firewall-Einstellungen. Hier müssen Sie **Heimnetz** und **Zugriff zulassen** auswählen.

Unten im Packet-Sender-Programm sollte nun **UDP** gefolgt von der Zahl **55056** zu sehen sein. Wenn diese Zahl nicht 55056 ist, müssen Sie sie unter **File/Settings** im Register **Network** bei **UDP Server Port** eingeben, denn derselbe Port ist auch auf der Konfigurationsseite unseres Boards eingestellt.



Das Packet-Sender-Programm mit geöffneten Einstellungen.

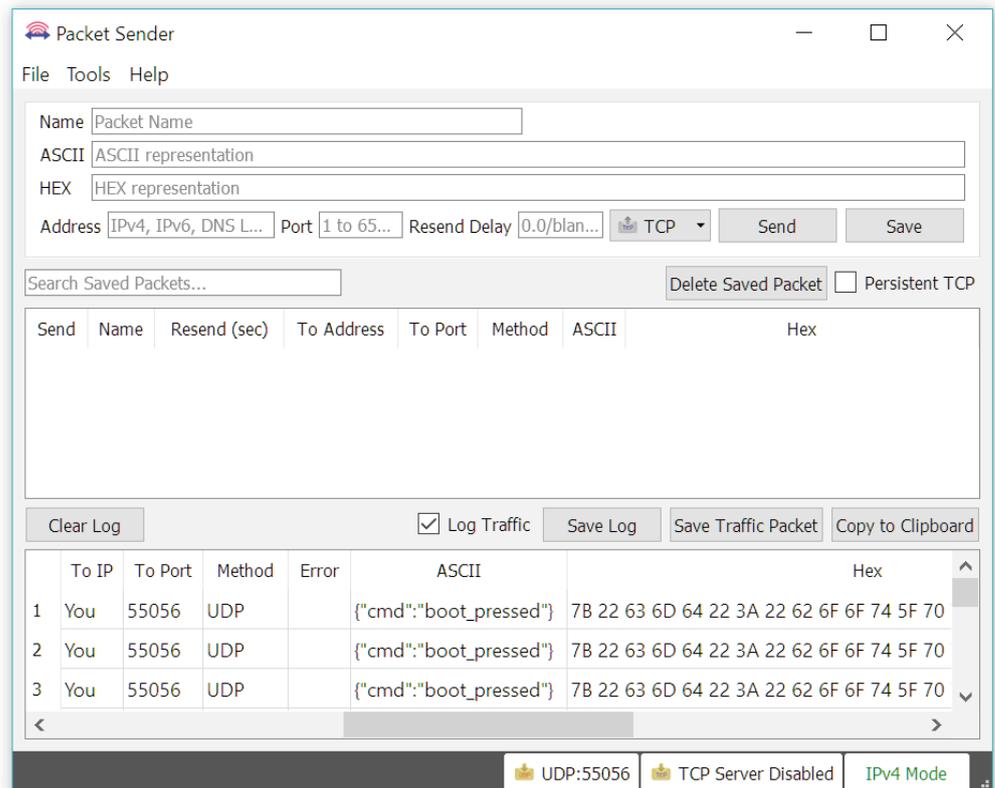
Und tatsächlich, nach kurzer Zeit erscheint im unteren Fenster eine Nachricht, die sich alle fünf Sekunden wiederholt. Sie haben also bereits erfolgreich eine Nachricht von Ihrem Board empfangen.

Tag 4: UDP und JSON

Hinter dem vierten Türchen des Adventskalenders finden Sie einen Satz Kabel, die für die kommenden Hardwareaufbauten nützlich sind. Im heutigen Projekt wollen wir uns aber anschauen, was das JSON-Format ist und wie es uns hilft, mit dem Board zu kommunizieren.

Zuerst aktivieren Sie wie gewohnt das Programm des heutigen Tages. Außerdem werden Sie das Packet-Sender-Programm mit den gleichen Einstellungen wie gestern benötigen. Heute erscheint erst dann eine Nachricht, wenn Sie kurz auf den Boot-Button auf dem Board gedrückt haben. Diese Nachricht sieht so aus:

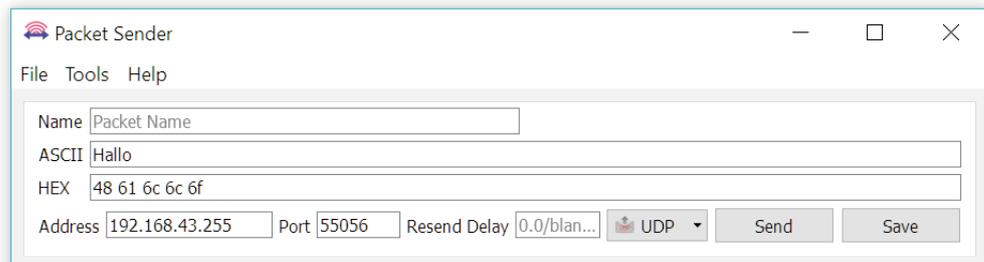
```
{"cmd":"boot_pressed"}
```



Die Nachricht im Packet-Sender-Programm.

Das liegt daran, dass diese Nachricht der JSON (**JavaScript Object Notation**) folgt. Dabei handelt es sich um ein schlankes Datenaustauschformat für Maschinen, das aber auch für Menschen sehr gut lesbar ist. Im Grunde besteht im JSON-Format alles aus Namen und Werten. Zuerst kommt der Name des Werts (auch Key genannt), hier **cmd** (kurz für Command, Kommando), und anschließend der Wert, hier **boot_pressed**. Beide sind durch einen Doppelpunkt voneinander getrennt und, da es sich hier um Text handelt, von Anführungszeichen umschlossen. Ein komplettes JSON-Objekt ist zusätzlich von geschweiften Klammern umgeben.

Das Board kann nicht nur Nachrichten senden, sondern auch Nachrichten empfangen. Dazu müssen Sie im Packet-Sender-Programm ein paar Einstellungen vornehmen. Im oberen Bereich finden Sie ein Textfeld, das mit **Address** beschriftet ist. Hier sollten Sie wieder die Broadcast-Adresse eingeben, die Sie auch auf dem Board verwendet haben. Alternativ können Sie die Board-IP oder sogar nur den mDNS-Namen (einfach **nodeesp**) nutzen. In das Feld **Port** rechts daneben fügen Sie wieder den Port **55056** ein. Als letzte Einstellung wählen Sie im Drop-down-Menü weiter rechts **UDP** aus. Nun können Sie etwas weiter oben im Feld **ASCII** eine Nachricht eingeben und mit dem **Send**-Button übertragen. Senden Sie zunächst einen willkürlichen Text, etwa **Hallo**. Da das Packet-Sender-Programm unter dem gleichen Port auch lauscht und Sie die Broadcast-IP-Adresse verwenden, werden Sie einige der gesendeten Nachrichten auch selbst empfangen. Zusätzlich empfangen Sie vom Board die Nachricht **{"cmd":"ack"}**. **ack** ist die Kurzform von **acknowledged** (bestätigt). Damit bestätigt das Board also den Empfang einer Nachricht, auch wenn es wahrscheinlich mit der Nachricht selbst nichts anfangen konnte.



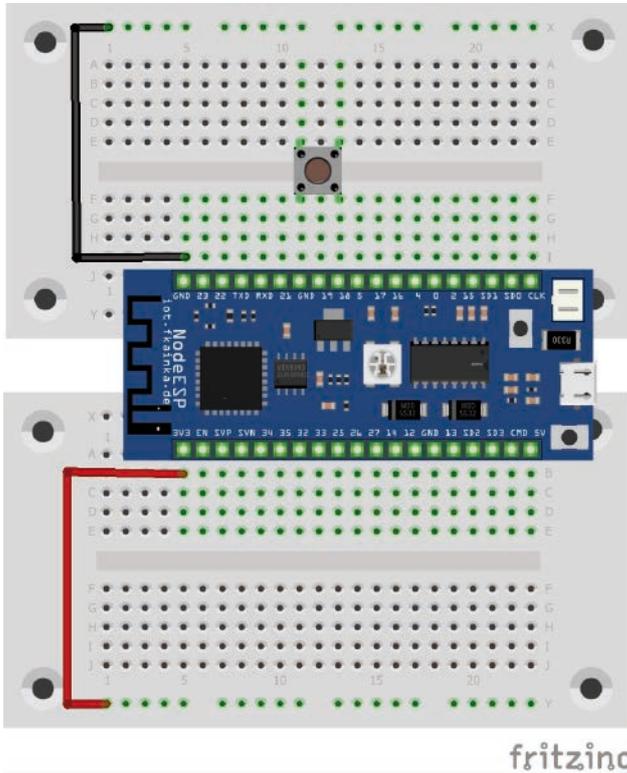
Wenn Sie nun aber einen vordefinierten Befehl im JSON-Format abschicken, sieht die Sache schon anders aus. Senden Sie zum Beispiel den Text **{"cmd":"ping"}**, kommt nach kurzer Zeit ein **{"cmd":"pong"}** als Antwort vom Board zurück. Es gibt noch weitere Kommandos, die Sie ausprobieren können. Mit **{"cmd":"get_wifi_status"}** bekommen Sie beispielsweise den Verbindungsstatus des Boards zurück. Mit **{"cmd":"get_wifi_scan"}** werden alle WLAN-Netzwerke in Reichweite erfasst und in einer Liste übermittelt.

Es gibt eine ganze Reihe von allgemeingültigen Kommandos, die Sie auf der Boardseite oder auf der Projektseite einsehen können. Aber Vorsicht: Ein paar Kommandos aus der Kategorie **Settings** können die Einstellungen des Boards so verändern, dass nichts mehr geht. Wenn das passiert, können Sie zu den Standardeinstellungen zurückkehren, indem Sie den Boot-Button über zehn Sekunden lang gedrückt halten. Dies ist in allen Programmen möglich, bei denen WPS in Klammern hinter dem Programmnamen steht. Die gleiche Funktion ist auch auf der Konfigurationsseite unter **Advanced Settings** zu finden (**Restore Default Configuration**). Als dritte Möglichkeit können Sie Pin 13 beim Boardstart auf LOW setzen. Vergessen Sie aber nicht, die Verbindung an Pin 13 nach dem Reset wieder zu trennen.

5. Tag

Tag 5: Node-RED

Hinter dem fünften Türchen des Adventskalenders finden Sie einen Taster, der in dem heutigen Versuch anstelle des Boot-Buttons eingesetzt werden soll. Der Button wird so an Pin 18 und GND angeschlossen, dass er beim Drücken den Pin auf Masse, also LOW, zieht. Wenn Sie nun das Programm des Tages auswählen und den Button drücken, wird wieder eine Nachricht über UDP übermittelt.



Heute werden Sie das Programm **Node-RED** kennenlernen, mit dem Sie Ihre eigenen kleinen Programmabläufe erstellen und Vorgänge automatisieren können. Node-RED ist eine kostenlose Software, die es uns ermöglicht, verschiedene Geräte, Onlinedienste und APIs (Programmierschnittstellen) über eine einfache grafische Oberfläche zu verbinden. Die Verbindungen, Flows genannt, werden dabei einfach per Mausklick gezogen.

Node-RED gibt es für Linux, Windows und macOS. Sinnvoll ist die Verwendung eines Raspberry Pi, da dieser ohne großen Energiebedarf immer aktiv sein kann, ganz im Sinne des Smarthome. Außerdem ist Node-RED dort unter Raspbian bereits vorinstalliert.

Ich stelle die Installation beispielhaft für Windows 10 vor. Die meisten dieser Befehle sind aber auf andere Systeme übertragbar.

Im ersten Schritt muss **Node.js** installiert werden. Diese Software bekommen Sie von der Webseite <https://nodejs.org/>, auf der auch eine genaue Installationsanweisung zu finden ist. Installieren Sie die ca. 16 MByte große MSI-Datei per Doppelklick. Um sicherzugehen, dass alles korrekt installiert wurde, sollten Sie nun das Windows-Terminal, auch Kommandozeile oder Shell genannt (Windows-Taste + [R] und dann **cmd** eingeben), öffnen und

node --version && npm --version

eingeben. Das Ergebnis lautet z. B. **v8.10.0** und **5.6.0**, Sie könnten aber bereits eine neuere Version haben. In der Kommandozeile müssen Sie dann den Befehl zur Installation von Node-RED eingeben:

npm install -g --unsafe-perm node-red

Das kann eine Weile dauern. Anschließend starten Sie Node-RED mit diesem Kommandozeilenbefehl:

node-red

```
C:\Users\done1>node --version && npm --version
v8.10.0
5.6.0

C:\Users\done1>npm install -g --unsafe-perm node-red
npm WARN deprecated i18next-client@1.10.3: you can use npm install i18next from version 2.0.0
npm WARN deprecated nodemailer@1.11.0: All versions below 4.0.1 of Nodemailer are deprecated. See https://nodemailer.com/status/
npm WARN deprecated node-uuid@1.4.8: Use uuid module instead
C:\Users\done1\AppData\Roaming\npm\node-red-pi -> C:\Users\done1\AppData\Roaming\npm\node_modules\node-red\bin\node-red-pi
C:\Users\done1\AppData\Roaming\npm\node-red -> C:\U

> bcrypt@1.0.3 install C:\Users\done1\AppData\Roami
> node-pre-gyp install --fallback-to-build

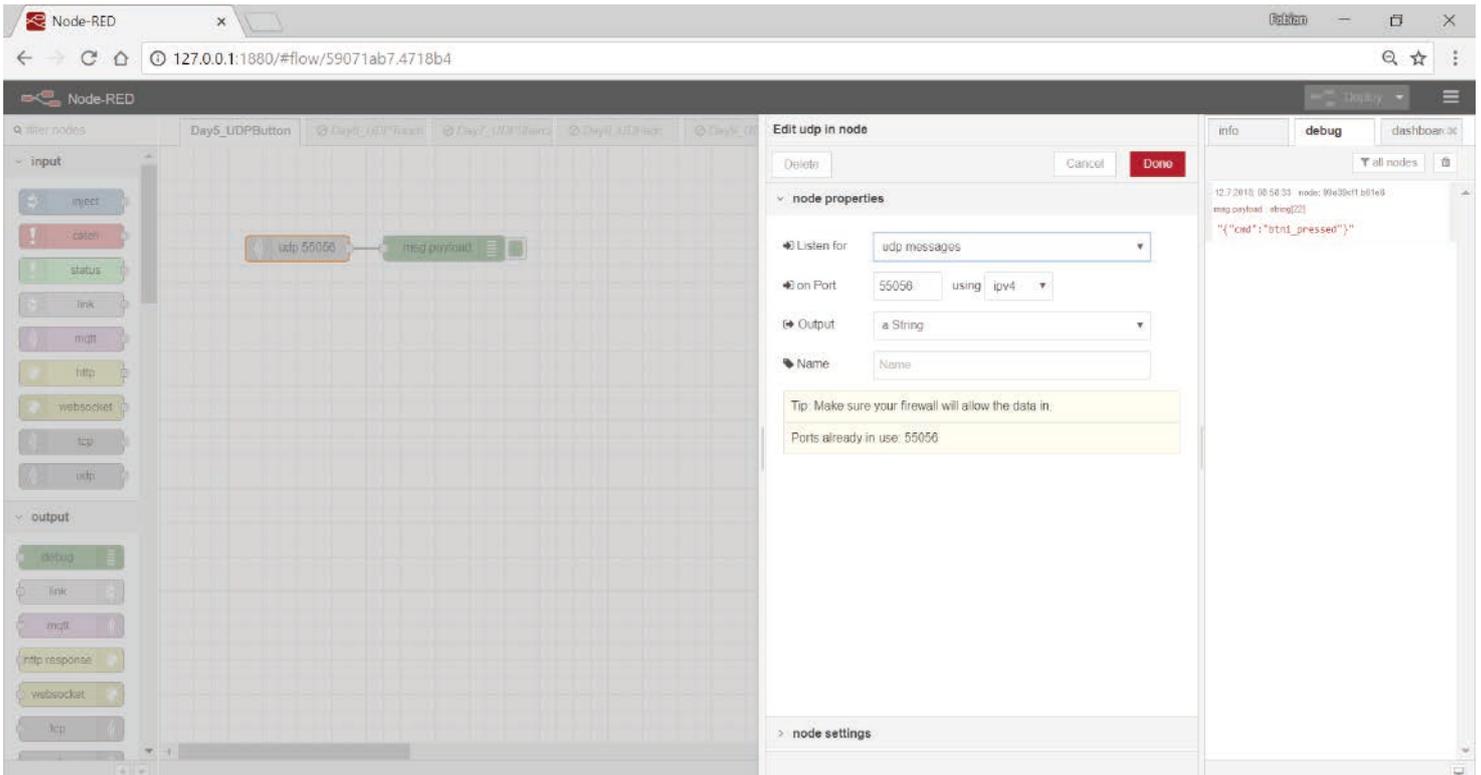
[bcrypt] Success: "C:\Users\done1\AppData\Roaming\n
+ node-red@0.18.4
added 367 packages in 161.208s

C:\Users\done1>node-red
7 Mar 12:30:58 - [info]
Welcome to Node-RED
=====
7 Mar 12:30:58 - [info] Node-RED version: v0.18.4
7 Mar 12:30:58 - [info] Node.js version: v8.10.0
7 Mar 12:30:58 - [info] windows_NT 10.0.16299 x64 L
7 Mar 12:30:59 - [info] Loading palette nodes
7 Mar 12:31:00 - [warn] -----
7 Mar 12:31:00 - [warn] [node-red/rpi-gpio] Info :
7 Mar 12:31:00 - [warn] [node-red/tail] Not current
7 Mar 12:31:00 - [warn] -----
7 Mar 12:31:00 - [info] Settings file : C:\Users\done1\AppData\Roaming\SPB_Data\node-red\settings.js
7 Mar 12:31:00 - [info] User directory : C:\Users\done1\AppData\Roaming\SPB_Data\node-red
7 Mar 12:31:00 - [warn] Projects disabled: set editorTheme.projects.enabled=true to enable
7 Mar 12:31:00 - [info] Flows file : C:\Users\done1\AppData\Roaming\SPB_Data\node-red\flows_LAPTOP-1UTMOOMI.json
7 Mar 12:31:00 - [info] Creating new flow file
7 Mar 12:31:00 - [info] Server now running at http://127.0.0.1:1880/
7 Mar 12:31:00 - [info] Starting flows
7 Mar 12:31:00 - [info] Started flows
```

Die Firewall meldet sich. Im Hintergrund startet Node-RED in der Konsole.

Sofort meldet sich die Firewall, die wissen möchte, wie mit der neuen Verbindung vorgegangen werden soll. Wichtig ist, dass Sie **Private Netzwerke** und **Zugriff zulassen** wählen. Nur dann kann Node-RED richtig arbeiten. Wichtig ist auch, dass Sie das Terminal jetzt nicht schließen, denn damit beenden Sie auch Node-RED wieder.

Nun haben Sie Node-RED installiert und gestartet, aber außer ein paar Zeilen im Terminal noch nicht viel Nützliches gesehen. Das liegt daran, dass die Bedienoberfläche der Software über den Browser geöffnet wird. Dazu öffnen Sie die Adresse **http://127.0.0.1:1880/**. Dabei handelt es sich um die localhost-Adresse, mit der Sie Ihren Rechner adressieren. Die Zahl 1880 steht für den Port, über den Sie Node-RED erreichen. Wenn Sie von einem anderen Rechner aus auf Node-RED zugreifen möchten, müssen Sie natürlich die richtige IP des Rechners verwenden. Nun sehen Sie die Oberfläche von Node-RED mit der Flow-Fläche in der Mitte, den Nodes auf der linken und Informationen auf der rechten Seite.



Einstellungen und eine empfangene Nachricht

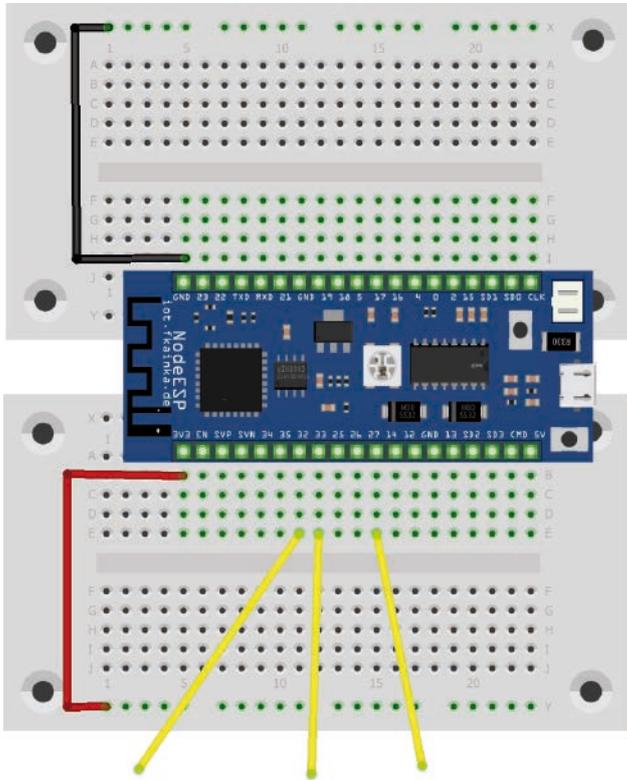
Auf der linken Seite finden Sie im Bereich **Input** einen Node mit der Bezeichnung **udp**. Ziehen Sie ihn in die freie Fläche in der Mitte. Etwas weiter unten unter dem Punkt **output** finden Sie **debug**. Ziehen Sie diesen Node ebenfalls auf die Fläche. Verbinden Sie die beiden Nodes durch Klicken und Ziehen mit der Maus. Anschließend klicken Sie doppelt auf den UDP-Node, um die Einstellungen zu öffnen. Geben Sie alles genau so ein, wie Sie es auf dem Bild oben sehen. Dann bestätigen Sie mit **Done**. Zum Start müssen Sie auf den **Deploy**-Button oben rechts in der Ecke klicken. Es erscheint kurz die Meldung **Successfully deployed**. Der neue Flow ist jetzt aktiv. Als letzten Schritt wählen Sie am rechten Rand das Register **debug** aus und drücken dann auf den Button auf dem Steckboard. Im **debug**-Fenster erscheint eine Nachricht im bekannten JSON-Format: **{"cmd": "btn1_pressed"}**. Die Verbindung zu Node-RED steht also.

Hinweis: Das Abzählen der Board-Pins kann mühsam sein. Deswegen gibt es auf der Boardseite im Bereich **Pinout** eine Schablone, die Sie ausdrucken, ausschneiden und auf das Board legen können.

6. Tag

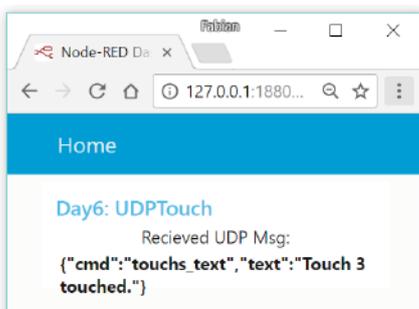
Tag 6: Touch to Speech

Hinter dem sechsten Türchen des Adventskalenders finden Sie einen Widerstand mit 1 kOhm, der erst morgen zum Einsatz kommt. Heute wollen wir uns weiter mit Node-RED beschäftigen und die Touch-Funktion des Boards ausprobieren. Dazu müssen Sie wie gewohnt das Programm des heutigen Tages auswählen und außerdem die folgende Schaltung aufbauen. Die gelben Drähte sind die Touch-Sensoren mit einem freien Ende.



fritzing

Drei Drähte hängen an einem Ende in der Luft.



Das einfache Dashboard.

Nach dem Aufbau können Sie Node-RED wieder über das Terminal und den Befehl `node-red` starten. Öffnen Sie dann über die Adresse <http://127.0.0.1:1880/> den Node-RED-Editor. Als Erstes werden wir heute ein paar Nodes nachinstallieren. Dazu klicken Sie am rechten oberen Rand auf die drei Striche. Sie finden dann den Punkt **manage palette**. Hier sind alle installierten Nodes aufgelistet. Wenn Sie nun das **Install**-Register auswählen, sehen Sie eine riesige Anzahl unterschiedlicher Module, die Sie installieren können. Für heute suchen wir nach dem Modul **node-red-dashboard**. Klicken Sie auf den **Install**-Button. Eine Nachricht erscheint, die Sie vor den teilweise schwierigen Installationen warnt. Klicken Sie einfach auf **Ok**, denn mit diesem Modul verhält es sich einfach. Nach kurzer Wartezeit erscheint eine Erfolgsmeldung, und Sie haben die neuen Nodes in Ihrer Node-Zeile.

Im nächsten Schritt können Sie den Flow des heutigen Tages einrichten. Damit Sie nicht alles selbst machen müssen, gibt es die Möglichkeit, ganze Flows in Node-RED zu importieren. Klicken Sie dazu wieder auf die drei Striche und anschließend auf **Import**, gefolgt von **Clipboard**. In das erscheinende Textfeld können Sie die Flows kopieren. Sie finden den Flow des heutigen Tages, sowie die aller folgenden Tage, auf der Webseite iot.fkainka.de und dort auf der Unterseite zum aktuellen Versuch.

Nach dem Importieren sollte oben in der Leiste ein neues Register zu sehen sein. Zunächst müssen Sie aber den gestrigen Flow deaktivieren, weil er am UDP-Port lauschen würde. Klicken Sie doppelt auf den Tab des Flows, wechseln Sie dann im Flow-Menü den Status von **Enabled** auf **Disabled** und bestätigen Sie mit **Done**. Nun können Sie die neuen Abläufe mit **Deploy** bestätigen.

Öffnen Sie im Browser diese Seite:

<http://127.0.0.1:1880/ui>

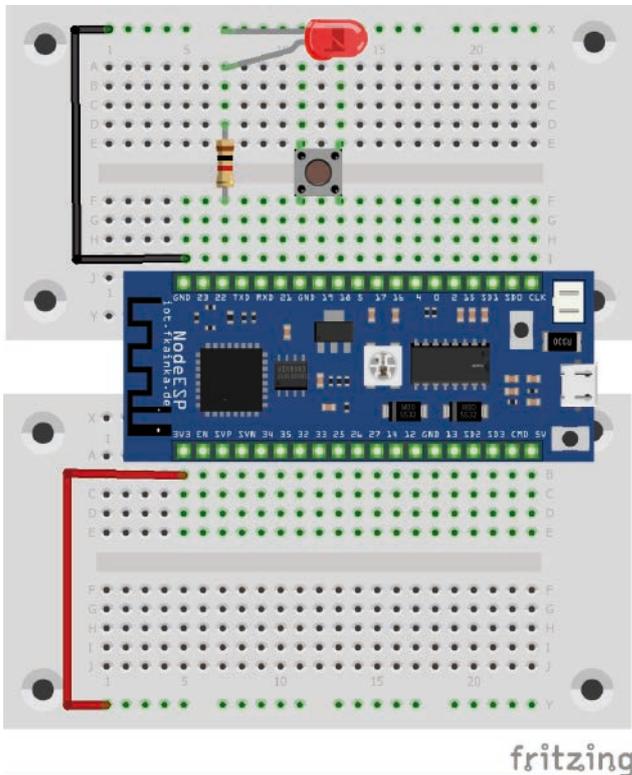
Was Sie nun sehen, ist ein Userinterface (UI), oft auch Dashboard genannt. Dank der Installation des neuen Node-Moduls ist es Ihnen jetzt möglich, eine eigene Oberfläche zu erstellen, über die Sie Vorgänge steuern können. Die Oberfläche von heute haben Sie mit dem Flow des heutigen Tages importiert. Es handelt sich um eine Textbox, die die UDP-Nachrichten anzeigt. Es gibt aber noch einen kleinen Zusatz: Ich hoffe, Sie haben die Lautstärke Ihres PCs nicht zu hoch eingestellt, wenn Sie jetzt einen der frei hängenden Drähte am Board berühren. Das Board erkennt automatisch die Berührung und sendet eine UDP-Nachricht, die im Textfeld erscheint. Zusätzlich gibt es auf dem Dashboard einen Text-to-Speech-Node, der den Textwert im JSON-Objekt vorliest. Sie haben also erfolgreich eine erste Sprachausgabe realisiert – und das alles mit nur einem einzigen Node.

Hinweis: Falls eine Berührung keine Nachricht auslöst, passen Sie die Sensibilität der Touch-Erkennung auf der Konfigurationsseite an. Je höher der Wert, umso empfindlicher reagieren Touch-Buttons auf Berührungen.

Tag 7: Treppenhauslicht

Hinter dem heutigen Türchen befindet sich eine rote LED. Damit wollen wir nun ein einfaches Treppenhauslicht mit Webinterface realisieren. Dazu müssen Sie die folgende Schaltung aufbauen und das heutige Programm aktivieren.

7. Tag



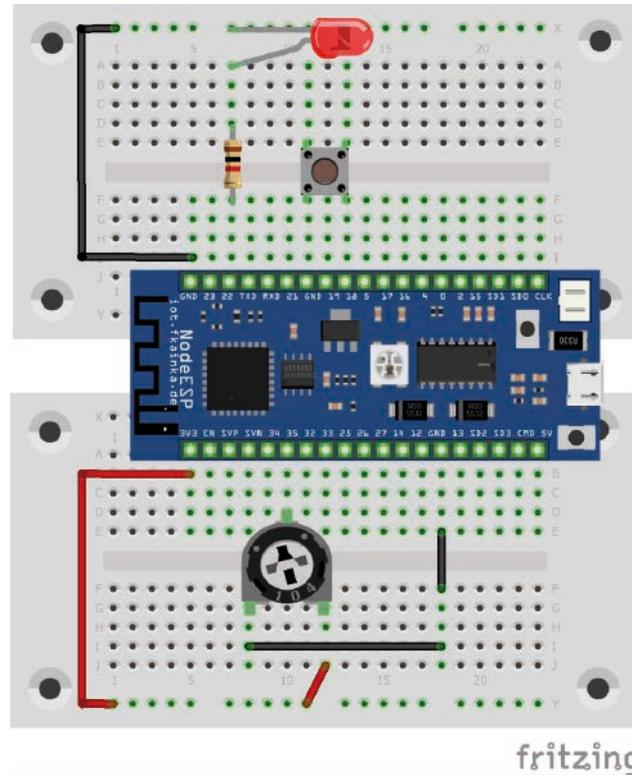
Den Node-RED-Flow zum Importieren finden Sie auf der Webseite. Deaktivieren Sie nach dem Importieren den Flow von gestern, damit sich die UDP-Empfangs-Nodes nicht in die Quere kommen. Im Flow selbst finden Sie diesmal neben dem Empfangs- auch einen Sende-Node. Diesen müssen Sie wahrscheinlich an Ihre Broadcast-IP anpassen. Wenn am UDP-Node eine Nachricht ankommt, wird sie wieder von einem JSON-String, also reinem Text, in ein JSON-Objekt konvertiert. Der Unterschied ist, dass man nun direkt auf einzelne Attribute zugreifen kann.

Der folgende Switch-Node ist vergleichbar mit einer IF-Abfrage. Es wird überprüft, ob das Kommando **trigger** gesendet wurde. Dieses Kommando wird durch Drücken des Buttons ausgelöst. Ein Trigger-Node übernimmt daraufhin die Arbeit. Wird der Trigger-Node aktiviert, sendet er den Befehl **light_on** für „Licht an“. Diese Nachricht, ebenfalls im JSON-Format, wird vom NodeESP-Board interpretiert, und entsprechend wird das Licht angeschaltet. Nach weiteren vier Sekunden sendet der Trigger-Node die Nachricht: **light_off** für „Licht aus“. Auch hierauf reagiert das Board entsprechend. Die Off-Nachricht kann verzögert werden, indem der Trigger-Node erneut ausgelöst wird. Dafür wurde in dem Modul eine entsprechende Option aktiviert. Zusätzlich kann der Trigger-Node auch über die Web-UI aktiviert werden. Also können Sie das Treppenlicht auch über das Smartphone aktivieren, ganz im Sinne von Smarthome.

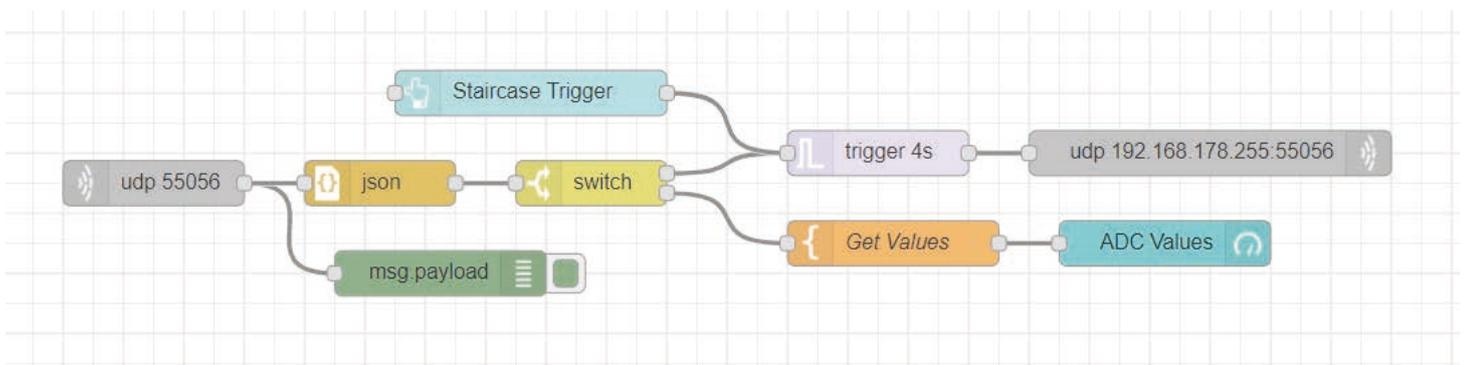
Hinweis: In der Windows-Version von Node-RED funktionierte der UDP-Node zum Senden nicht mit der Einstellung **broadcast message** (in den Node-Properties unter **send a**). In der Node-RED-Version, die ich auf einem Raspberry getestet habe, funktionierte allerdings das Senden nur mit dieser Einstellung. Das bedeutet, Raspberry-Nutzer müssen an dieser Stelle wahrscheinlich den UDP-Node zum Senden anpassen.

Tag 8: Ein Regler

Hinter dem achten Türchen des Adventskalenders befindet sich ein Potenziometer, das hier als Spannungsteiler verwendet wird. Sie können eine Spannung zwischen 0 V und 3,3 V einstellen, die vom Board gemessen und an Node-RED übertragen wird. Zusätzlich zu dem Treppenhauslicht von gestern können Sie so auch Messwerte übertragen.



Den Flow von heute bekommen Sie wie gewohnt von der Webseite. Vergessen Sie nicht, den gestrigen zu deaktivieren und eventuell die Sende-IP anzupassen. In dem heutigen Flow wird außerdem ein neuer Node für das Dashboard verwendet, nämlich ein Gauge (Messgerät), über das der gemessene Wert visuell dargestellt wird.

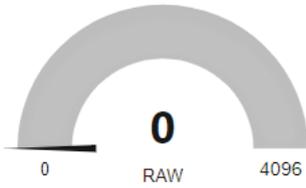


Wenn Sie das Programm des heutigen Tages aktivieren, werden Sie sehen, dass sich das Gauge-Element (Zeigerinstrument) auf der Dashboard-Seite etwa alle zwei Sekunden aktualisiert und die neuen Messwerte anzeigt. Wenn Sie nun an dem Rädchen des Potis drehen, verändern sich diese Werte. Dieser Messwert könnte für verschiedene Anwendungsfälle genutzt werden, beispielsweise um die Helligkeit der LED zu variieren oder die Leuchtdauer des Treppenlichts einzustellen, also typische Anwendungen in Ihrem Smarthome.

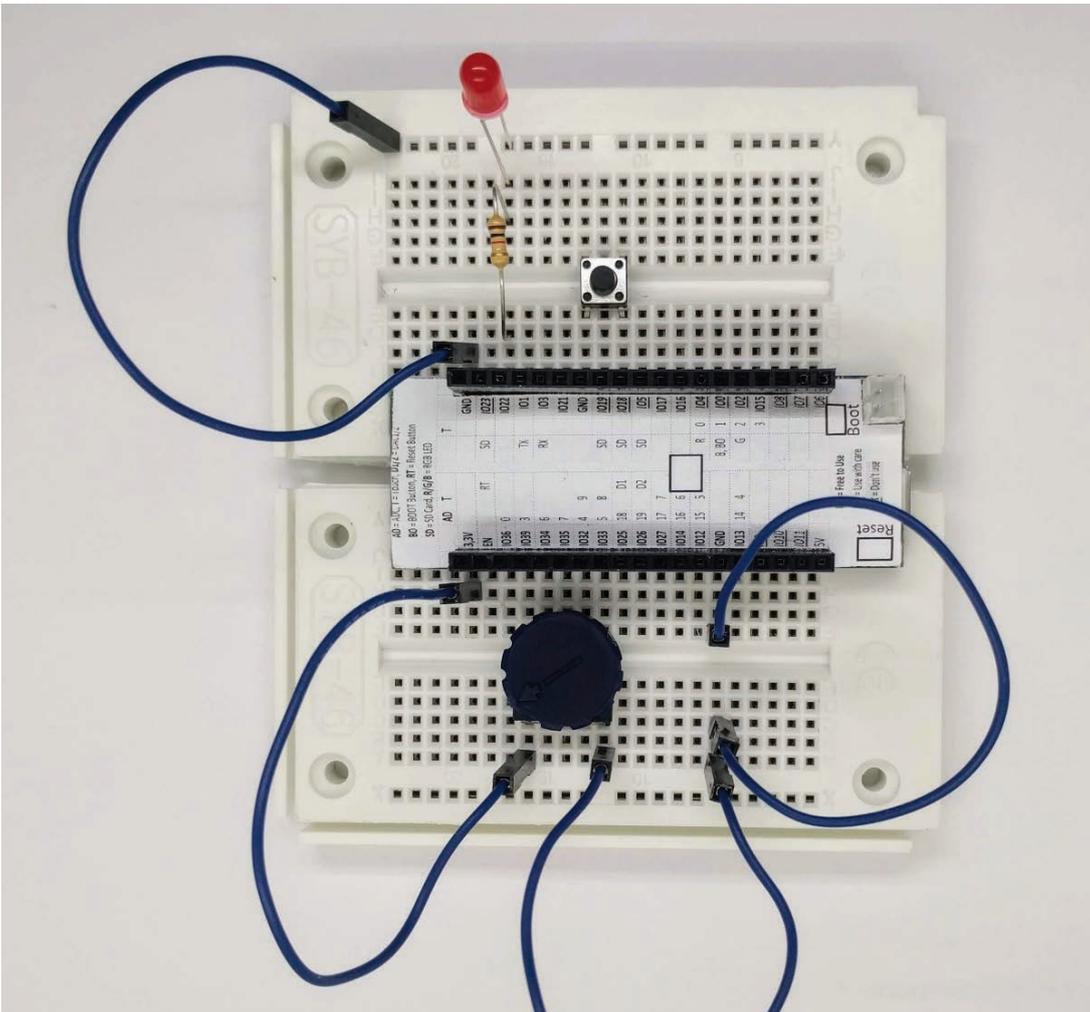
[Day 8] Staircase and ADC

STAIRCASE TRIGGER

ADC Values



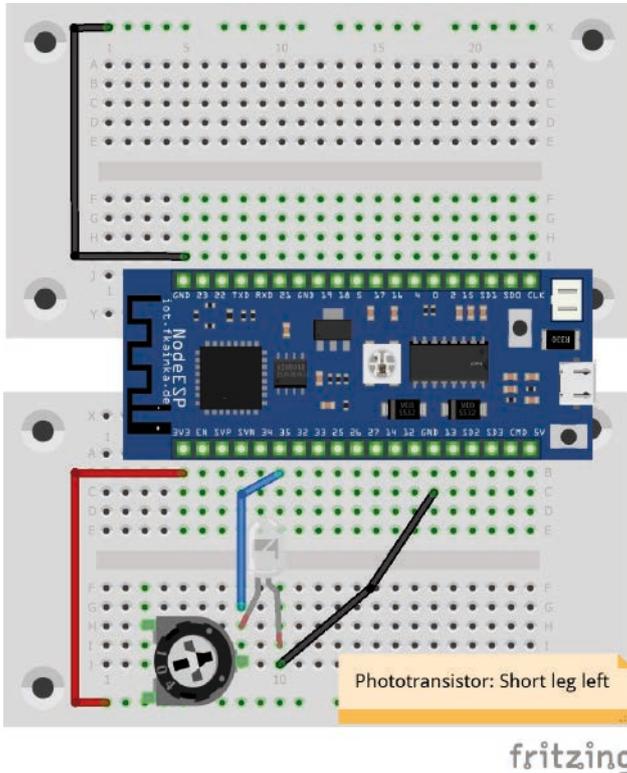
Das Userinterface zum heutigen Versuch.



9. Tag

Tag 9: Ein Stimmungslicht

Hinter dem heutigen Türchen finden Sie einen Fototransistor. Er wird verwendet, um die Umgebungshelligkeit zu messen. In dieser Schaltung liegt das Poti in Reihe zum Fototransistor. So kann man mit dem Poti die Sensibilität des Sensors verändern. Die Schaltung bewirkt, dass das Abdunkeln des Sensors den Messwert erhöht.



fritzing

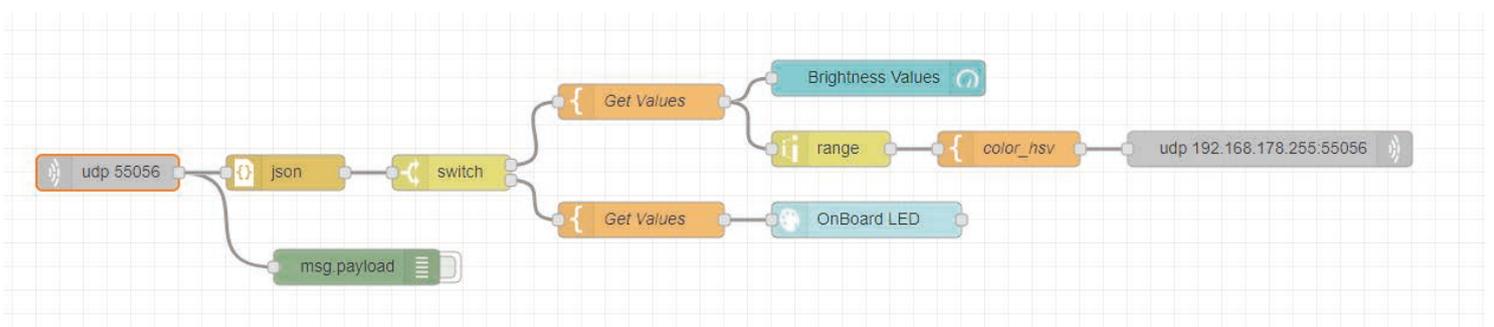
Dies können Sie im Gauge-Element auf dem Nutzerinterface zum heutigen Tag sehen. Dazu müssen Sie aber zunächst den Flow des heutigen Tages in Ihre Node-RED-Umgebung importieren und starten, natürlich nicht, ohne den gestrigen zu deaktivieren. Vergessen Sie auch nicht, das heutige Programm auf der Konfigurationsseite zu aktivieren und die IP des UDP-Sende-Nodes anzupassen.

Zusätzlich zur Anzeige der Messwerte kommt auch wieder die OnBoard-LED zum Einsatz. Diese soll, abhängig vom gemessenen Wert, die Farbe automatisch den Lichtverhältnissen anpassen. Wenn es absolut dunkel ist, soll die LED rot leuchten. Je heller es wird, desto mehr verändert sich die Farbe über Grün nach Blau. So können Sie Ihren Arbeitsplatz abends zum Beispiel mit einem entspannenden Rot beleuchten, während tagsüber ein kräftiges Blau leuchtet.

Der Range-Node Node rechnet die gemessenen Werte von 0 bis 4095 auf einen Bereich von 240 bis 0 um. Damit wird die gewünschte LED-Farbe im sogenannten HSV-Format gesendet. HSV steht für **Hue**, **Saturation** und **Value**, also Farbton, Sättigung und (hier) Helligkeitswert. Es handelt sich um eine alternative Darstellung zum RGB-System. Im HSV-Format wird der Farbton durch einen Winkel von 0° bis 360° dargestellt. Rot entspricht 0°, Grün liegt bei 120° und Blau bei 240°. Deswegen wird an dieser Stelle der Messwert in einen Wert bis 240 umgerechnet, denn damit ist die ganze Farbpalette von Rot bis Blau abgedeckt. Im folgenden Template-Node wird das Kommando zusammengestellt, das anschließend an das Board gesendet wird. Bei einem mittleren Wert wäre das Kommando:

```
{"cmd":"set_rgb_hsv","value":"hsv(120, 100%, 100%)"} 
```

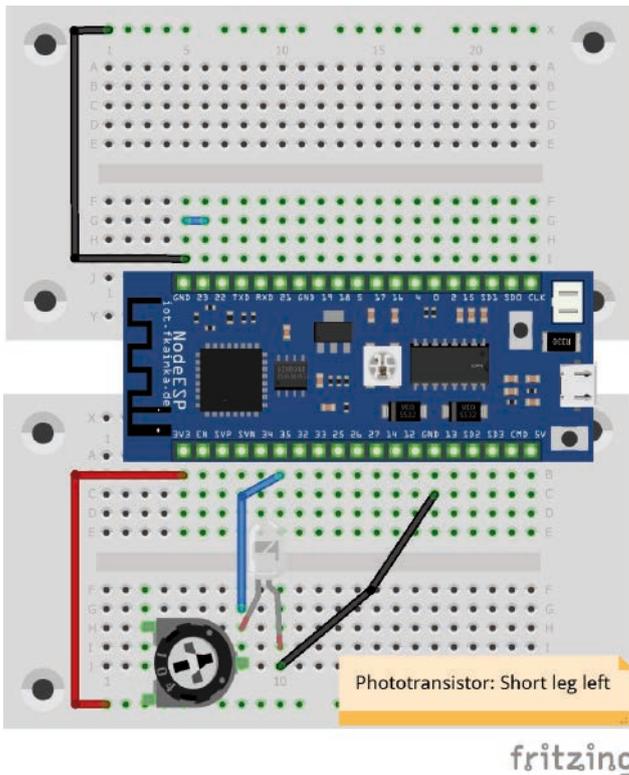
Die Prozentzeichen nach der Sättigung und der Helligkeit können auch weggelassen werden. Das Board ändert nun die Farbe und sendet als Bestätigung die gleichen Werte mit dem Kommando **get_rgb_hsv** zurück. Dies wird ebenfalls von Node-RED empfangen und anhand eines Color-Picker-Elements auf der Interface-Webseite dargestellt. So können Sie immer verfolgen, welche Farbe das Board gerade anzeigt.



Der Flow des heutigen Tages.

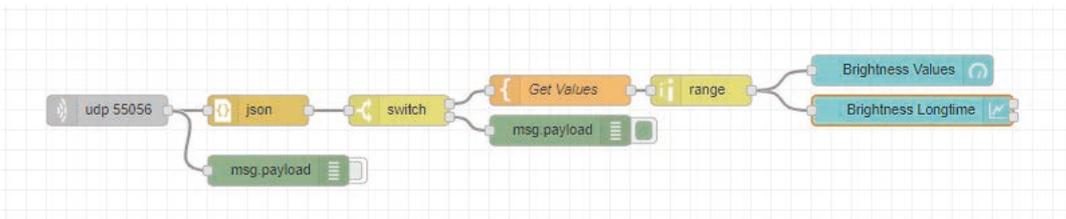
Tag 10: Deep Sleep

Hinter dem heutigen Türchen finden Sie ein Batteriefach für drei A-Batterien (4,5 V), um das Board unabhängig von einer Steckdose betreiben zu können. Für einen sparsamen Batteriebetrieb gibt es den Deep-Sleep-Modus. Dieser erlaubt es dem Board, in einen Tiefschlafmodus zu wechseln und so gut wie gar keinen Strom zu verbrauchen. Nach einer gewissen Zeit weckt sich das Board selbst auf und erfüllt eine kleine Aufgabe, um kurz danach wieder schlafen zu gehen. Durch die langen Schlaf- und kurzen Wachzeiten spart dieser Modus viel Energie und kann das Board so über Monate über die Batterien versorgen. Der Aufbau von heute verwendet wieder den Lichtsensor und bekommt zusätzlich eine Drahtbrücke von Pin 23 nach GND.

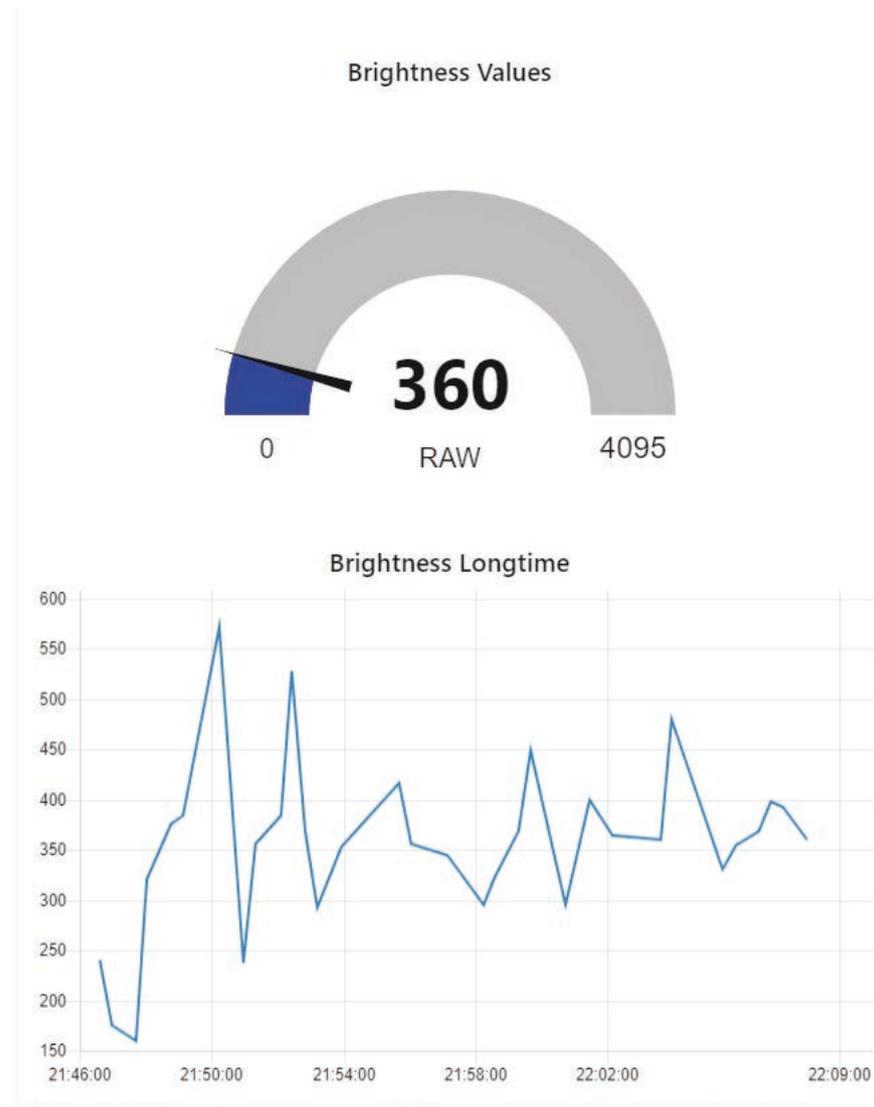


Die Drahtbrücke an Pin 23 beachten.

Stellen Sie zunächst das Programm des heutigen Tages ein. Wenn sich nun das Board resettet, blinkt die Status-LED zwar kurz grün, erlischt dann aber wieder. Nach ca. 20 Sekunden geht das Spiel von vorne los, der Deep-Sleep-Modus ist also aktiv.



Der Flow des heutigen Tages besteht aus einem Gauge und einem Chart. Ein Chart dient dazu, mehrere Messwerte über eine Zeitperiode grafisch aufbereitet darzustellen. Heute wurden die Messwerte übrigens mit einem Range-Node umgedreht, sodass eine starke Helligkeit auch einen hohen Messwert auf dem Gauge und dem Chart liefert. Sie können nun also die Node-RED-Instanz auf einem Rechner oder Raspberry laufen lassen, während Sie das Board mit den Batterien irgendwo platzieren und dort die Helligkeit über einen ganzen Tag verteilt messen.



Das Gauge- und das Chart-Element auf der Dashboard-Seite.

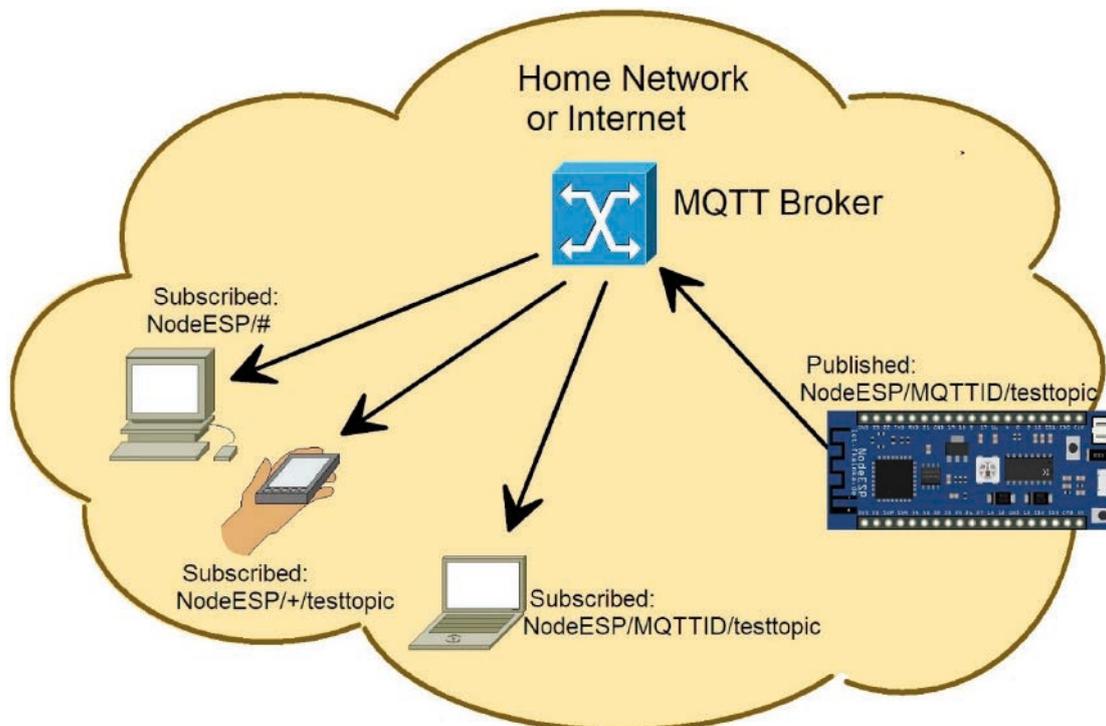
Hinweis: Wenn das Blinken der Status-LED alle 20 Sekunden Sie stört, beispielsweise weil Sie die Lichtverteilung in Ihrem Schlafzimmer über Nacht messen wollen, können Sie diese auch einfach auf der Konfigurationsseite deaktivieren. Um den Deep-Sleep-Modus zu verlassen, müssen Sie lediglich die Drahtbrücke an Pin 23 wieder herausnehmen und das Board resetten oder einen Deep-Sleep-Zyklus abwarten.

Tag 11: MQTT

Hinter dem heutigen Türchen finden Sie wieder einen Satz Kabel, die wir für die nächsten Tage und die komplexer werdenden Aufbauten benötigen werden.

In den letzten Tagen haben wir die meisten unserer Daten mittels UDP übertragen. Dieses Protokoll ist sehr praktisch, da es einerseits einen kleinen Overhead (Daten, die neben den eigentlichen Nutzdaten übermittelt werden) hat und andererseits Broadcast-fähig ist und somit gleich an mehrere Geräte gleichzeitig Daten senden kann. Heute wollen wir aber noch ein zweites Protokoll näher kennenlernen, das im Bereich IoT eine große Rolle spielt. Es geht um das Nachrichtenprotokoll MQTT (**Message Queue Telemetry Transport**), das speziell für die Kommunikation von Maschine zu Maschine (M2M) entwickelt wurde. Der Grund dafür, dass dieses Protokoll so genial ist, liegt in der Kompaktheit und Einfachheit, mit der Daten wie Steuerdaten, Messdaten oder Texte zwischen Geräten ausgetauscht werden können. Im Vergleich zu HTTP ist der Overhead sehr klein. Deswegen können auch Embedded Devices, also Mikrocontroller wie unser NodeESP, die Daten gut verarbeiten.

Grundlage für die Kommunikation bildet ein sogenannter Broker. Ein Broker ist eine Serveranwendung, die alle Nachrichten verarbeitet. Ein Gerät wie unser NanoESP kann sich mit dem Broker verbinden und Nachrichten veröffentlichen (publishen), also an den Broker senden. Ein anderes Gerät kann dann bestimmte Themen (Topics genannt) abonnieren (subscribe) und bekommt sofort alle neu zu diesem Thema veröffentlichten Nachrichten vom Broker zugestellt.



Der MQTT-Broker stellt die Nachrichten zu.

Die Topics sind hierarchisch strukturiert, wobei Unterthemen jeweils durch ein / getrennt sind. Zusätzlich gibt es sogenannte Wildcards. Durch + oder # können mehrere Themen gleichzeitig abonniert werden. Die Topics **MyHome/Kitchen/Temperature** und **MyHome/Bathroom/Temperature** können so z. B. durch das Abonnieren des Themas **MyHome/+Temperature** gleichzeitig empfangen werden. Das Plus ersetzt dabei immer nur eine Topic-Ebene, die Raute ersetzt gleich alle. Mit dem Topic **MyHome/#** empfangen Sie also gleich alle Nachrichten, die in allen Untertopics von **MyHome** gepublished werden. Eine Nachricht kann so ziemlich alles sein: vom reinen Text über Messwerte bis zu einfachen I/O-Befehlen zum Schalten von Licht.

Für heute brauchen Sie keinen Hardwareaufbau, denn wir beginnen mit ein paar grundlegenden Tests. Schauen Sie sich zunächst die Einstellungen auf der Konfigurationsseite an. Bestimmt ist Ihnen schon der Bereich **MQTT Settings** aufgefallen. Hier finden Sie bereits die richtig vorkonfigurierten Einstellungen. Als Broker-Adresse ist ein speziell für das Adventskalenderprojekt eingerichteter Server angegeben, nämlich **broker.fkainka.de**. Sie können aber auch selbst einen Broker aufsetzen, z. B. auf Ihrem Raspberry Pi. Ich empfehle dazu die Open-Source-Software **Mosquitto**. Die Client-ID wurde anhand der MAC-Adresse des Boards generiert. Der Grund dafür ist, dass diese ID serverweit nur von einem einzigen Gerät genutzt

werden darf. Da eine MAC-Adresse weltweit eindeutig sein sollte, bietet sie sich hier als ID an. Einen Usernamen und ein Passwort brauchen Sie für den Testbroker nicht.

Sie können die Verbindung testen, indem Sie weiter unten auf den Button **Test Connection** klicken. Nun sollte die Farbe der Fläche daneben auf Grün wechseln, wenn die Verbindung zum Broker erfolgreich hergestellt wurde. Falls es mal nicht funktioniert, kann es sein, dass der Server down ist und Sie den Broker wechseln müssen. Ein alternativer frei verfügbarer Broker ist z. B. **broker.mqttdashboard.com**.

Nachdem Sie die prinzipielle Verbindung getestet haben, können Sie nun das Programm des heutigen Tages einstellen. Gehen Sie anschließend auf die Seite

<http://www.hivemq.com/try-out/>

und klicken Sie dort auf **Try the websocket client**. Es erscheint eine Oberfläche, mit der Sie sich auf einem MQTT-Broker mit WebSocket-Unterstützung anmelden können. Unter **Host** müssen Sie einen Broker, also hier **broker.fkainka.de**, eintragen und anschließend mit **connect** eine Verbindung herstellen. Es öffnet sich die Oberfläche aus der Abbildung. Sie können nun mit Klick auf **Add New Topic Subscription** Ihr eigenes, persönliches Topic abonnieren. Dieses setzt sich aus dem Begriff NodeESP und der Client-ID zusammen. Bei meinem Board war das zum Beispiel:

NodeESP/240AC48178E4/

Sämtliche MQTT-Kommunikation vom und zum Board beginnt mit diesem Topic. Ich werde den Client-ID-Teil in Zukunft mit **[PID]** für „persönliche ID“ abkürzen. Sie müssen diesen Teil durch Ihre Client-ID ersetzen. Im ersten Versuch können Sie alle Kommunikation vom Board mit dem Topic

NodeESP/[PID]/#

empfangen. Wenn Sie das Board resetten, sehen Sie nach kurzer Zeit eine Meldung, die Ihnen sagt, dass sich das Board mit dem Broker verbunden hat. Diese Meldung wurde in das Topic **NodeESP/[PID]/status** geschrieben und durch das Subscriben mit **#** ebenfalls empfangen. Nun können Sie eine erste Nachricht an das Board übermitteln. Dazu geben Sie im **Publish**-Bereich im Feld **Topic** Ihr persönliches Topic plus das Subtopic **testtopic** ein, also in meinem Fall:

NodeESP/[PID]/testtopic

In dem Feld **Message** können Sie jetzt eine beliebige Nachricht eingeben. Das Feld kann sogar leer bleiben. Klicken Sie anschließend auf **Publish**, und nur wenige Augenblicke später haben Sie zwei weitere Nachrichten im **Messages**-Bereich. Die eine Nachricht ist ein Begrüßungstext und wurde im Topic **NodeESP/[PID]/testtopic/out** veröffentlicht. Die andere Nachricht enthält ebenfalls den Begrüßungstext, allerdings als JSON-String inklusive Kommando-Key, und wurde im Topic **NodeESP/[PID]/out** veröffentlicht.

Diese beiden Nachrichten kommen dadurch zustande, dass das Board die Topics in das bekannte Kommandoformat übersetzt und dann auswertet. Die Antworten erscheinen in einem Topic, das sich aus dem Kommando und **out** zusammensetzt, hier also **.../testtopic/out**, und gleichzeitig im Kanal **NodeESP/[PID]/out**.

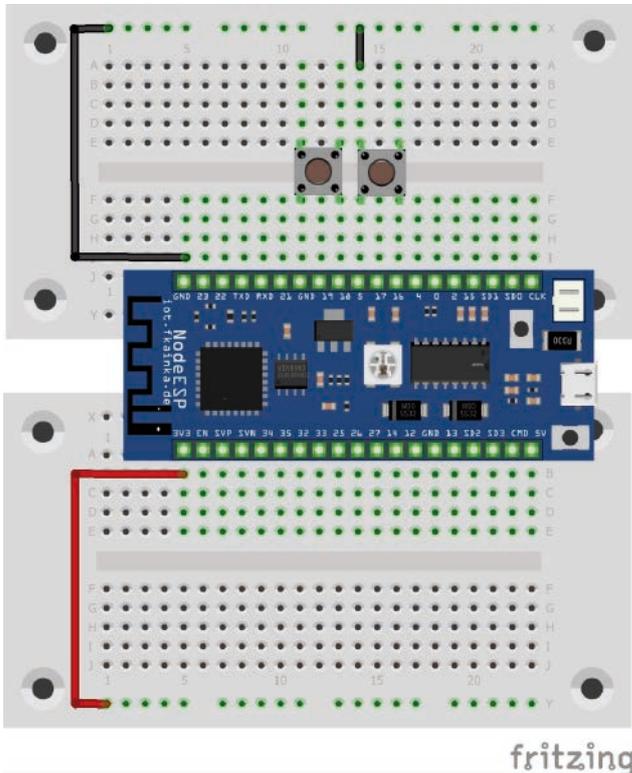
Äquivalent zum **out**-Topic gibt es ein **in**-Topic, in das Sie direkt Kommandos im JSON-Format publishen können. Wenn Sie zum Beispiel in das Topic **NodeESP/[PID]/in** den JSON-String **{"cmd":"testtopic"}** schreiben, käme es zu den gleichen Antworten wie beim ersten Versuch. Sie können also ein Kommando entweder über ein entsprechendes Topic oder über den **in**-Channel als JSON-Objekt schicken. Wundern Sie sich nicht darüber, dass einige bekannte Kommandos wie z. B. **get_wifi_status** etc. nicht funktionieren. Das liegt daran, dass diese Kommandos durchaus sensible Daten enthalten können und deswegen nicht über einen öffentlichen Broker gesendet werden sollten.

Die Verbindung zum Board über MQTT steht also. Morgen schauen wir uns genauer an, was wir damit machen können.

Tag 12: MQTT-Button

Hinter dem heutigen Türchen des Adventskalenders finden Sie einen weiteren Button. In dem Versuch des heutigen Tages werden wir uns weiter mit dem MQTT-Protokoll beschäftigen, diesmal in Verbindung mit Node-RED. Außerdem sehen wir, wie zwei Buttons gleichzeitig ausgewertet werden können. Der Versuch des heutigen Tages hat folgenden Aufbau:

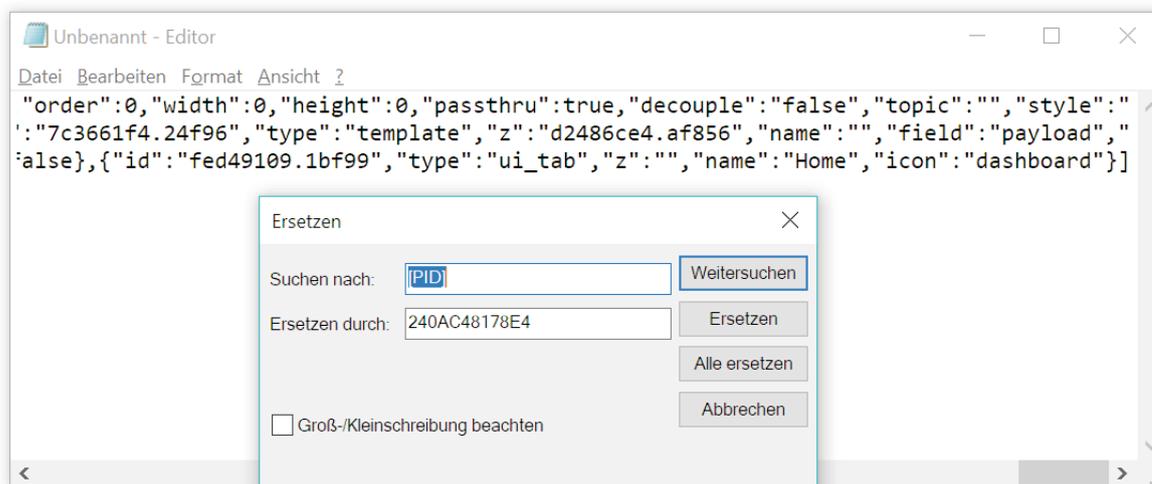
12. Tag



Der Aufbau mit zwei Buttons.

Es gibt also zwei Buttons, die wir nutzen können. Aktivieren Sie nun das Programm des heutigen Tages. Anschließend können Sie den Node-RED-Flow importieren. Diesen müssen Sie zunächst aber leicht anpassen. Sie müssen nämlich **[PID]** im Topic durch Ihre persönliche MQTT-ID ersetzen. Das geht aber ganz einfach mit dem folgenden Trick:

Kopieren Sie den Node-RED-Flow in einen Texteditor Ihrer Wahl, in Windows zum Beispiel in den Editor. Anschließend wählen Sie die Option **Ersetzen** (im Editor unter **Bearbeiten/Ersetzen** ([Strg]+[H])). Dort geben Sie im Feld **Suchen nach [PID]** ein und im Feld **Ersetzen durch** Ihre persönliche ID. Anschließend können Sie einfach auf **Alle ersetzen** klicken. Schon haben Sie den Flow angepasst und können ihn in Node-RED importieren und starten, natürlich nicht, ohne die alten Flows zu deaktivieren.



So können Sie den Flow schnell auf Ihre persönliche ID anpassen.

Wenn Sie nun in Node-RED auf das **Debug**-Register gehen und einen Button auf dem Steckboard drücken, sehen Sie dort zum Beispiel folgende Nachricht:

```
{
  "cmd": "get_button_state",
  "value": [
    {
      "pin_name": "Button1",
      "pin_id": 1,
      "pin_nr": 18,
      "pin_state": true,
      "pin_value": 0
    },
    {
      "pin_name": "Button2",
      "pin_id": 2,
      "pin_nr": 16,
      "pin_state": false,
      "pin_value": 0
    }
  ]
}
```

Unter dem Key **cmd** wird das Kommando **get_button_state** gesendet. Unter **value** folgt ein Array mit allen Buttons und ihren Werten. Über **pin_name**, **pin_id** oder **pin_nr** können Sie einen Button identifizieren. Die ersten beiden Werte sind selbst definiert, der dritte gibt den verwendeten IO-Pin am Board an. **Pin_state** gibt an, wie der boolesche Wert des Buttons ist, also TRUE oder FALSE bzw. HIGH oder LOW. Ist der Button nicht gedrückt, ist der Pin-Value übrigens TRUE. Das liegt daran, dass der Pin mit einem internen Pull-up-Widerstand auf HIGH und per Tastendruck auf LOW gezogen wird. Der Key **pin_value** gibt einen analogen Wert an, der bei Buttons keine Rolle spielt.

In diesem Fall habe ich Button2 gedrückt, und das Board hat sofort eine Meldung mit allen aktivierten Buttons und deren Werten sowohl an das **out**-Topic, das am Debug-Node hängt, als auch an das Topic

NodeESP/[PID]/button/state/get/out

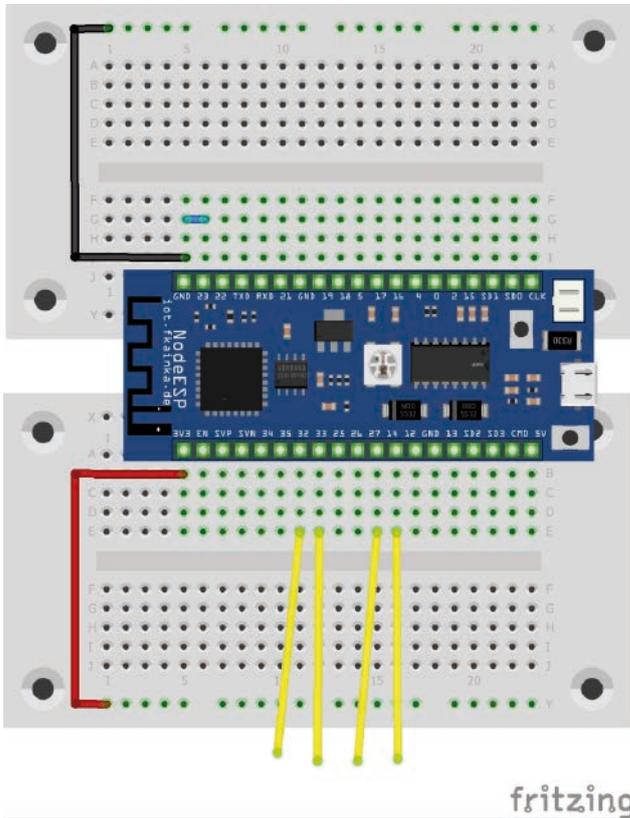
gesendet. Das letzte Topic wird in Node-RED ausgewertet. Dazu kommt ein JSON-Node, der aus dem reinen Text ein JSON-Objekt macht. Der Split-Node trennt das Button-Array in einzelne Elemente auf, die nacheinander an den Switch-Node weitergereicht werden. Dieser überprüft den Pinnamen und entscheidet, um welchen der beiden Buttons es sich handelt. Es folgt ein Template-Node, der einfach nur den Wert von **pin_state** filtert, also TRUE oder FALSE an den nächsten Node weitergibt. Hierbei handelt es sich um einen Dashboard-Button, der einen An-Zustand anzeigt, wenn der Wert FALSE war, und umgekehrt. Er invertiert an dieser Stelle also sozusagen den Wert des Buttons.

Wenn Sie auf das Dashboard wechseln, indem Sie beispielsweise auf das Dashboard-Register auf der rechten Seite und anschließend auf das Rechteck mit dem Pfeil nach oben klicken, sehen Sie auch die beiden Buttons. Da das Board beim Loslassen des Buttons direkt die Statusveränderung mitteilt, können Sie also sofort sehen, welcher Button gedrückt und welcher nicht gedrückt ist.

Der große Vorteil bei MQTT und dem Verwenden eines Online-Brokers ist, dass man die Buttons von überall in der Welt auswerten kann. Sie können also das Board mit zur Arbeit nehmen, während der Node-RED-Server zu Hause auf einem Rechner läuft. Und dann könnten Sie beispielsweise das Licht und die Heizung einstellen, wenn Sie sich auf den Nachhauseweg machen, oder per Sprachausgabe den Daheimgebliebenen mitteilen, dass Sie auf dem Weg sind. Und das alles mit einem simplen Knopfdruck von irgendwo auf der Welt.

Tag 13: Smarter Schalter für ein Soundboard

Hinter dem Türchen des heutigen Tages befindet sich ein weiterer Widerstand mit 1 kOhm, der heute noch keine Rolle spielt. Stattdessen möchte ich Ihnen zeigen, wie man einen smarten Touch-Schalter baut. Dazu ist folgender Aufbau nötig, der nur aus vier lose hängenden Drähten als Touch-Buttons und einer Drahtbrücke an Pin 23 besteht.



Fünf gelbe Drähte als Touch-Buttons.

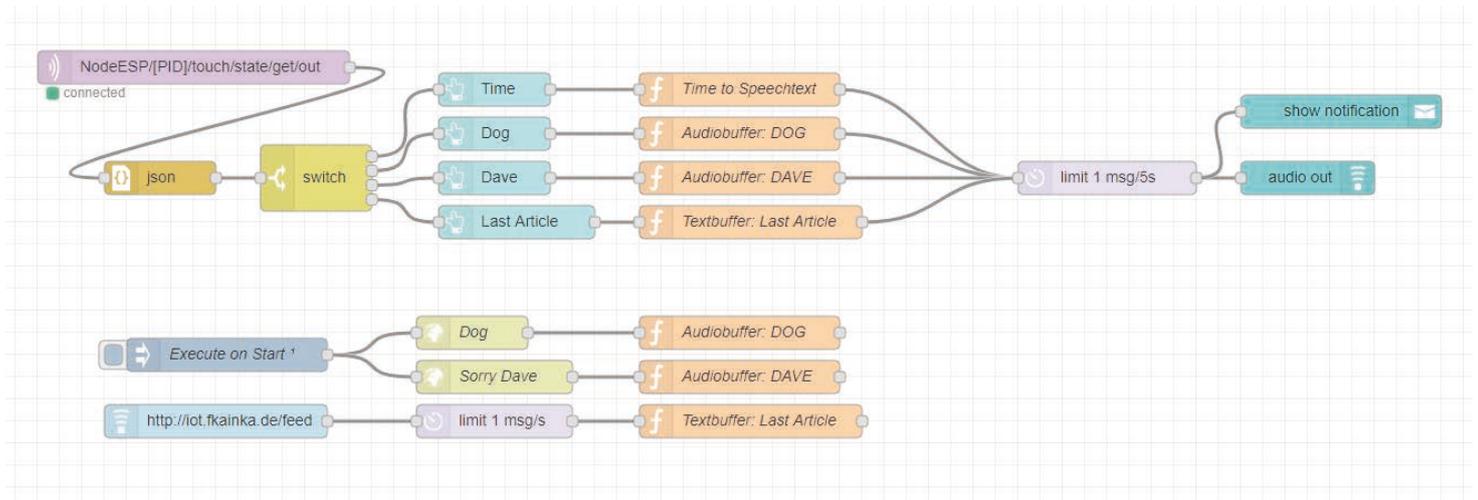
Die Besonderheit des heutigen Versuchs besteht vor allem im Programm selbst, das Sie an dieser Stelle über die Konfigurationsseite aktivieren können. Nach dem Aktivieren werden Sie feststellen, dass die Status-LED kurz blinkt und dann erlischt. Das liegt daran, dass das Board wieder im stromsparenden Deep-Sleep-Modus betrieben wird. Aufgeweckt wird es aber nicht nach einer gewissen Zeit, sondern erst, wenn Sie einen der Touch-Buttons berühren. Das Board sendet dann Infos über den Touch-Button zum Broker, der das Aufwecken ausgelöst hat. Das sieht dann z. B. so aus:

```
{"cmd":"get_touch_state","pin_name":"Touch4","pin_id":6,"pin_nr":14,"pin_state":true}
```

Während des Verbindungsaufbaus können Sie sehen, dass die Status-LED grün blinkt. Nach dem Senden leuchtet sie grün und zeigt so das erfolgreiche Übertragen an. Sollte die LED rot aufleuchten, hat die Übertragung nicht funktioniert. Direkt danach geht das Board wieder in den Deep-Sleep-Modus. Das System ist so stromsparend, dass man das Board wieder über die Batterien versorgen kann, um den smarten Schalter irgendwo in der Wohnung zu platzieren.

Hinweis: Um vom Deep-Sleep-Modus wieder in den normalen Modus zu wechseln, müssen Sie einfach nur die Drahtbrücke an Pin 23 herausziehen und das Board resettet.

Der smarte Schalter selbst ist schon eine Besonderheit, aber wir wollen den Schalter nun auch mit Node-RED und einer Anwendung verbinden. Wir werden in dem Node-RED-Flow von heute noch einmal den Audio-out-Node bemühen, diesmal aber auch, um Audiodateien abzuspielen. Dazu müssen Sie zunächst einmal den Flow des heutigen Tages importieren, natürlich nicht, ohne die alten Flows zu deaktivieren, und Ihre persönliche ID statt des [PID]-Platzhalters einfügen.



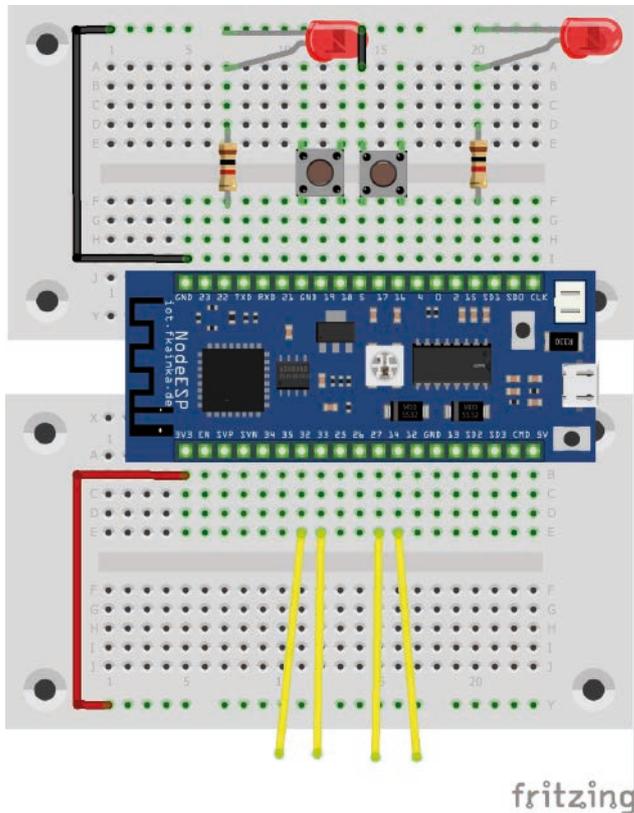
Der Flow des heutigen Tages.

Nun können Sie einfach mal ausprobieren, was die einzelnen Buttons eigentlich für Funktionen versteckt halten. Aber nicht vergessen, die Lautsprecher leiser zu stellen.

Tag 14: Smarte Lichter

Hinter dem Türchen des 14. Tages befindet sich eine weitere LED. Nun haben wir zwei LEDs mit Vorwiderständen und zwei Taster sowie genug Kabel, um ein komplexes Lampensystem zu bauen. Dazu benötigen Sie den folgenden Aufbau:

14. Tag



Der Aufbau mit vier losen Kabeln (gelb) als Touch-Buttons.

Nun können Sie das Programm des heutigen Tages aktivieren und den heutigen Flow einfügen. Vergessen Sie nicht, die **[PID]** durch Ihre persönliche ID zu ersetzen, am einfachsten mit dem Editortrick.

In dem von Ihnen aktivierten Programm sind die LEDs als PWM-LEDs eingestellt. Das bedeutet, dass Sie die Helligkeit der LEDs verändern können. Der Wertebereich reicht dabei von 0 bis 255. Zusätzlich können Sie die LEDs aber immer noch wie digitale Ausgänge ein- oder ausschalten. Folgende Kommandos können Sie verwenden:

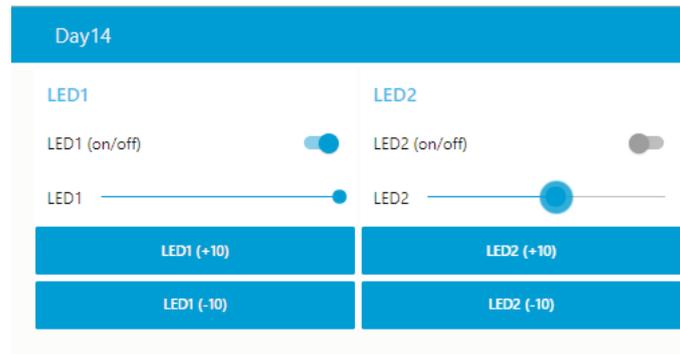
Kommando im in-Topic	Einstellungen
{„cmd“:„set_pwm_state“, „LED1“:255}	Die LED wird auf die maximale Helligkeit eingestellt. Nur sichtbar, wenn die LED an ist.
{„cmd“:„set_pwm_state“, „LED1“:0}	Die LED wird auf die minimale Helligkeit eingestellt (die LED leuchtet nicht).
{„cmd“:„set_pwm_state“, „LED1“:true}	Die LED wird aktiviert und die letzte Helligkeit eingestellt. Änderungen der Helligkeit werden sofort sichtbar.
{„cmd“:„set_pwm_state“, „LED1“:„ON“}	
{„cmd“:„set_pwm_state“, „LED1“:false}	Die LED wird deaktiviert. Eine Veränderung der Helligkeit wird erst nach einem erneuten Aktivieren sichtbar.
{„cmd“:„set_pwm_state“, „LED1“:„off“}	

Diese Statusveränderungen können über das **out-Topic** oder über **NodeESP/[PID]/pwm/state/set** gesendet werden. Bei einer Veränderung eines Status wird der aktuelle Zustand auch über das Topic

NodeESP/[PID]/pwm/state/get/out

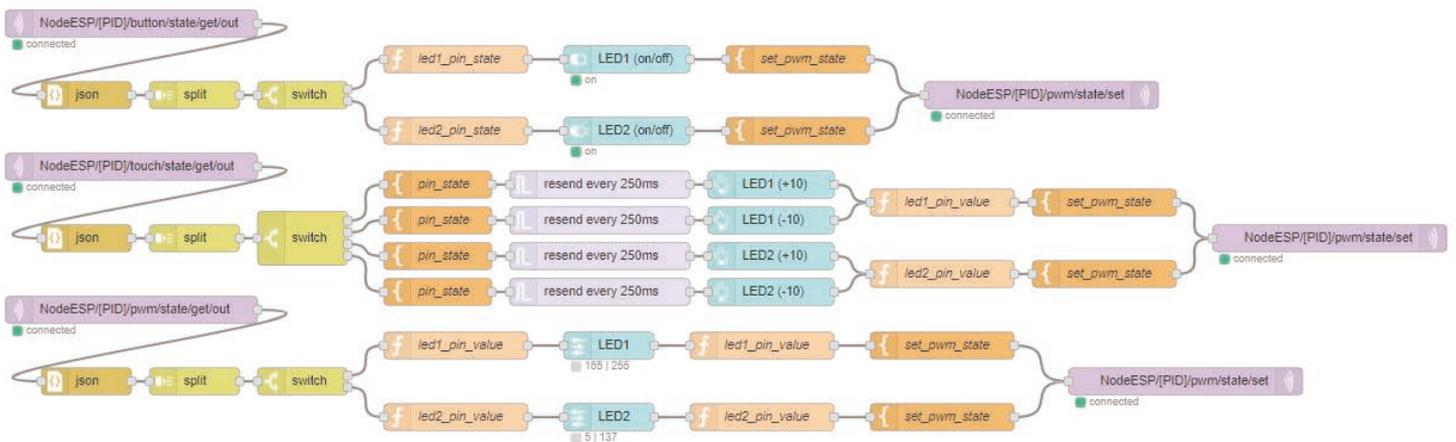
gepublizt.

Zusätzlich sind in dem heutigen Programm zwei Buttons und vier Touch-Buttons aktiviert, die nach dem bekannten System funktionieren. Der Node-RED-Flow ist heute etwas größer, denn für jeden Button und jede Einstellung gibt es eine UI-Schaltfläche.



Das Userinterface des heutigen Programms.

Das Userinterface gibt schon einige Hinweise darauf, wie der heutige Versuch funktioniert. Die beiden LEDs simulieren zwei dimmbare Lampen im Haushalt. Mit den beiden Tastern auf dem Steckboard oder über das Webinterface können sie an- oder ausgeschaltet werden. Mit den Reglern in der UI kann die Helligkeit eingestellt werden. Über die Schaltflächen darunter wird die Helligkeit jeweils um +10 oder -10 verändert. Die gleiche Funktion erfüllen auch die Touch-Buttons. Wenn Sie den auf dem Aufbau bild linken Draht berühren, verringern Sie über die Zeit die Helligkeit der linken LED. Der Draht direkt rechts daneben erhöht dabei die Helligkeit. Die beiden verbliebenen Drähte funktionieren genauso für die rechte LED. Sie können die Helligkeit auch verändern, wenn die LED gerade nicht an ist. Allerdings wird dann die Veränderung erst sichtbar, nachdem Sie die LED wieder angeschaltet haben. Die Funktion kann dennoch sehr praktisch sein, wenn Sie beispielsweise die Helligkeit der Lampe in Abhängigkeit von der Uhrzeit variieren wollen.



Der Flow des Tages.

Tag 15: Fernbedienungsempfänger

Hinter dem heutigen Türchen befindet sich ein ganz besonderes Bauteil, nämlich ein Infrarotempfängermodul. Damit ist es möglich, handelsübliche Infrarotfernbedienungssignale zu empfangen. Dazu benötigen wir nebenstehenden Aufbau.

15. Tag

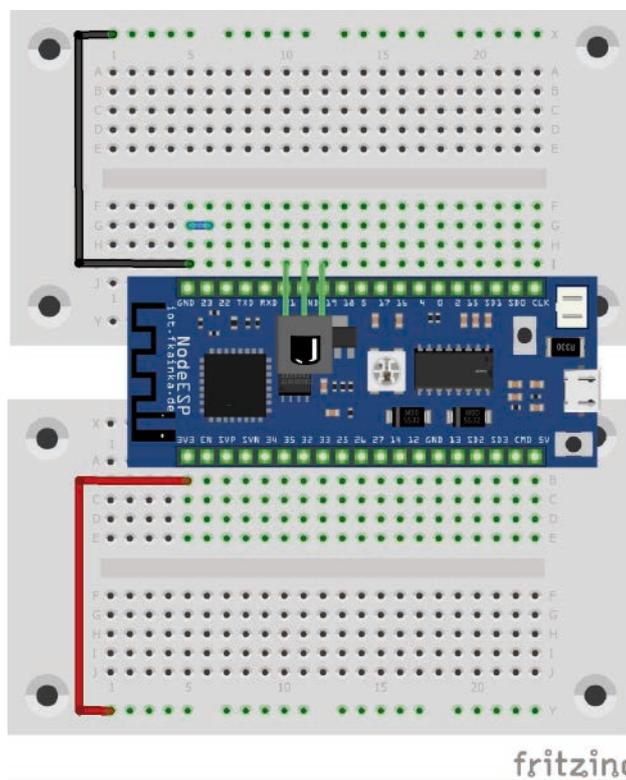
Nachdem Sie die Schaltung aufgebaut haben, können Sie das Programm des heutigen Tages auswählen. Die Boardwebseite können Sie dann nicht mehr erreichen, denn sie musste für den Versuch deaktiviert werden. Das Board arbeitet nun im Headless-Modus, also ohne sichtbare Oberfläche. Um wieder die Webseite starten und zum Beispiel andere Programme aktivieren zu können, müssen Sie lediglich die Drahtbrücke an Pin 23 herausziehen und das Board neu starten. Dann läuft wieder alles wie gewohnt.

Um zu sehen, ob das Programm des heutigen Tages richtig läuft, müssen Sie zuerst den Flow des heutigen Tages in Ihre Node-RED-Umgebung einfügen. Vergessen Sie nicht, die vorangegangenen Flows zu deaktivieren und das MQTT-Topic an Ihr persönliches Topic anzupassen.

Nachdem Sie den Flow aktiviert haben, können Sie eine Fernbedienung auf den Empfänger richten. Sie sehen dann auf der GUI-Seite den vom Board übermittelten Text, z. B. folgenden:

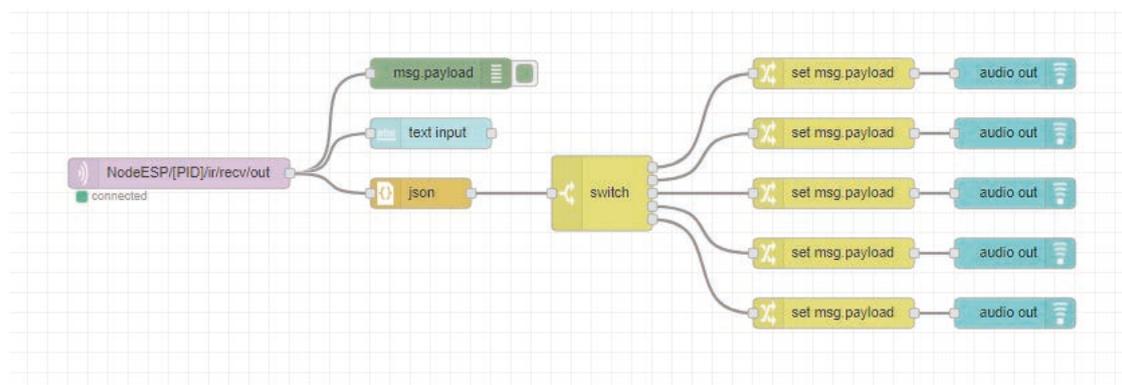
```
{"cmd":"ir_rcv","type":3,"value":551489775,"length":32}
```

Wichtig ist vor allem der **value**-Wert. Wählen Sie zum Beispiel die Power-Taste auf Ihrer Fernbedienung und kopieren Sie den empfangenen Wert. Dieser muss in den Switch-Node eingetragen werden. Sie können derzeit fünf verschiedene Werte, also fünf verschiedene Fernbedienungstasten, in den Switch-Node eintragen. Jede dieser Tasten führt dann zu einem Node, der einen Text als Payload setzt und diesen an einen Audio-out-Node weitergibt. Der erste Wert im Switch-Node entspricht der Power-Taste. Also wird auch „Power“ vom Browser vorgelesen. Dies dient allerdings allein der Demonstration. Im Sinne des Smarthomes können Sie genauso einfach die Wohnzimmerbeleuchtung dimmen, wenn der Fernseher angeschaltet wird. Node-RED macht die Anpassung an eigene Wünsche sehr einfach.



fritzing

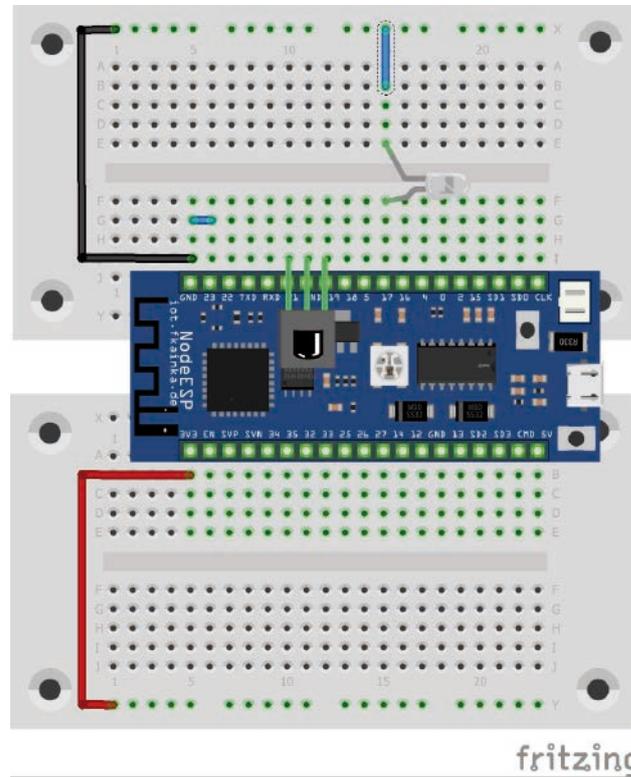
So schließen Sie das IR-Modul richtig am Board an.



Der Flow des heutigen Tages.

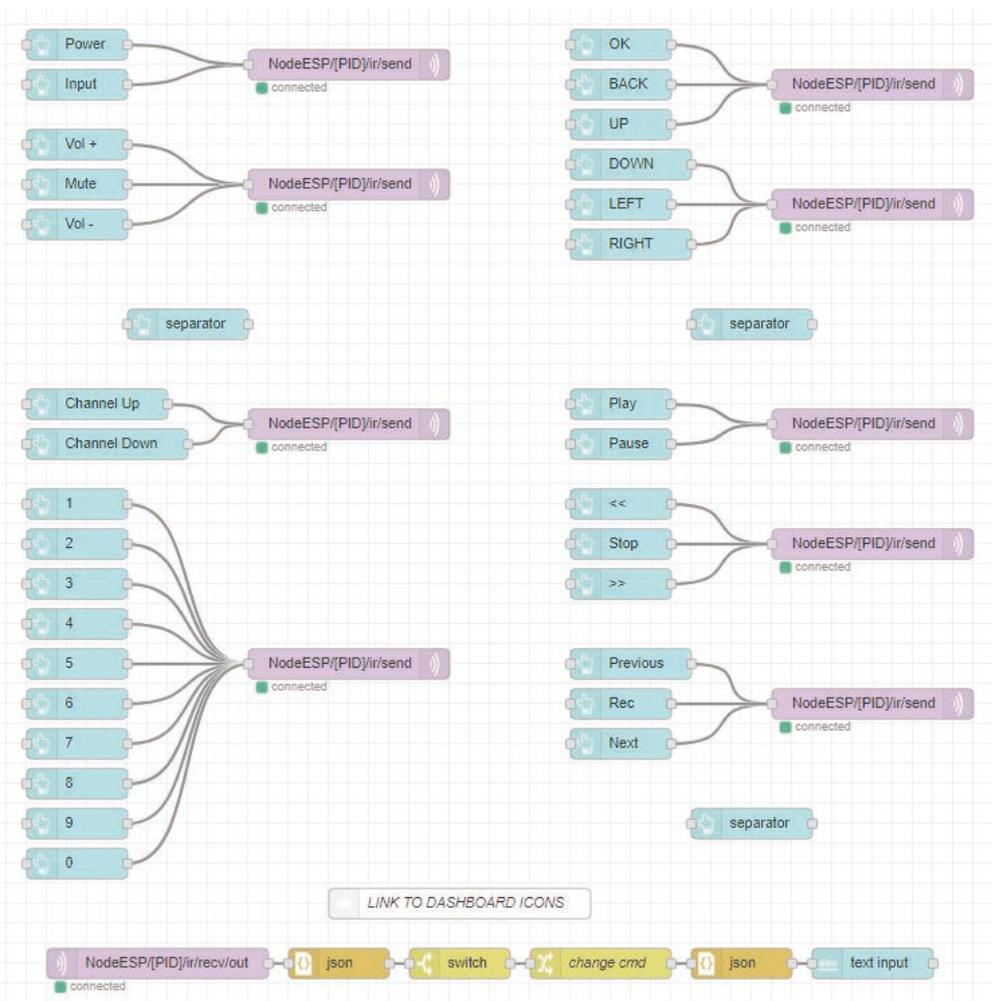
Tag 16: Universalfernbedienung

Hinter dem Türchen des heutigen Tages befindet sich das Gegenstück zum gestrigen Bauteil, nämlich eine IR-Diode. Jetzt können Sie eine komplette Fernbedienung für den Fernseher nachbauen, die Sie per Smartphone und Webseite bequem steuern. Dazu benötigen Sie den folgenden Aufbau.



Der Aufbau der Fernbedienungen mit IR-Empfänger und Sendediode.

Wählen Sie das Programm des heutigen Tages aus. Das Board startet anschließend wieder im Headless-Modus, in dem die Webseite des Boards nicht erreichbar ist. Stattdessen ist lediglich die MQTT-Funktion des Boards verfügbar. Um in den Standardmodus zu gelangen, müssen Sie die Drahtbrücke an Pin 23 entfernen und das Board neu starten.



Der Flow des heutigen Tages.

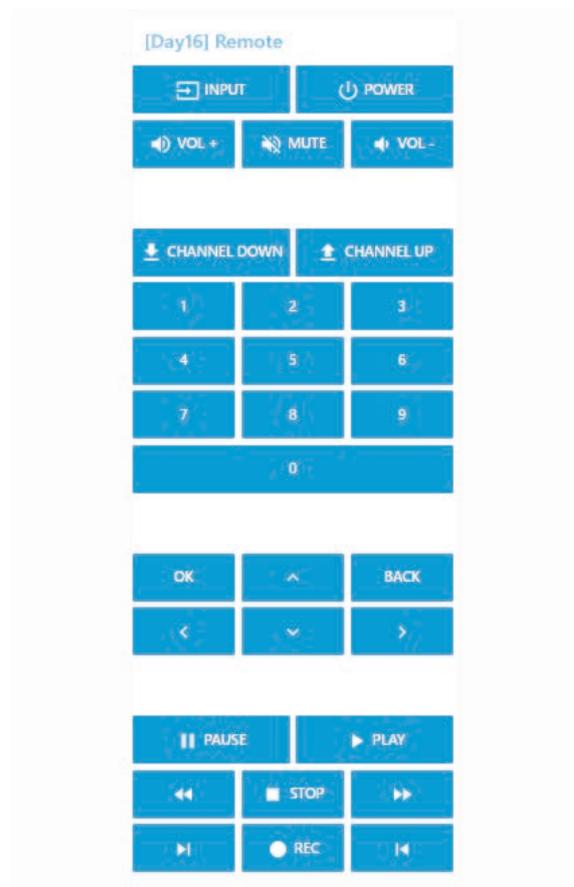
Der Flow des Tages sieht kompliziert aus, besteht aber im Grunde nur aus einer großen Anzahl von UI-Buttons, die alle in dasselbe Topic schreiben. Bevor Sie den Flow importieren und starten, sollten Sie alle Topics mit dem „Alles ersetzen“-Trick auf Ihr persönliches Topic umschreiben. Außerdem sollten Sie alle vorherigen Flows deaktivieren. Anschließend können Sie den Flow starten und auf die UI-Seite wechseln. Sie sehen eine ganze Reihe von Buttons, die allerdings noch nicht auf Ihren Fernseher kalibriert sind. Deswegen gehen Sie zunächst auf den zweiten Tab mit dem Namen **Day16 Recieve IR Codes**. Diese Seite sieht zwar viel langweiliger aus, denn sie enthält nur ein Textfeld, sie ist aber sehr nützlich. Wenn Sie nämlich nun eine Fernbedienung auf den Empfänger richten und eine Taste drücken, erscheint ein Text im Fenster, zum Beispiel folgender:

```
{"cmd":"ir_send","type":3,"value":551494365,"length":32}
```

Dieser Text wurde über das Topic **NodeESP/[PID]/ir/recv/out** empfangen und enthält die Daten des empfangenen Codes. Nur ein paar Kleinigkeiten wurden vom Flow geändert. Zum einen werden sogenannte Wiederholungs-codes herausgefiltert, zum anderen wurde das Kommando von **ir_recv** auf **ir_send** geändert, und zwar aus diesem Grund: Sie brauchen nun den Text praktischerweise nur zu kopieren und können ihn anschließend in den Payload-Bereich eines Buttons einfügen. Schon sendet der Button beim Betätigen auf der UI-Seite das Kommando mit den Daten des Codes an das Board, das wiederum den Code per IR-Diode an das TV überträgt. Auf diese Weise können Sie Code für Code einprogrammieren und sich Ihre eigene Fernbedienung erstellen.

The screenshot shows the Node-RED web interface. On the left is a palette of nodes. The main workspace contains a flow with several 'NodeESP/(PID)/ir/send' nodes connected to various input nodes. The 'Edit button node' panel is open on the right, showing the configuration for a button node. The 'node properties' section includes fields for Group, Size, Icon, Label, Colour, and Background. The 'When clicked, send:' section has a Payload field containing a JSON object: `{ "cmd": "ir_send", "type": 3, "value": 551494365, "length": 32 }`. There is also a checkbox for 'If msg arrives on input, pass through to output' and a Name field.

So werden die Kommandos als Payload des Buttons gespeichert.



Hinweis: Das Layout der GUI ist einer durchschnittlichen modernen Fernbedienung nachempfunden. Icons auf den Buttons sollen die Funktion verdeutlichen. Eine Liste nutzbarer Icons finden Sie auf der Webseite <https://klarsys.github.io/angular-material-icons/>.

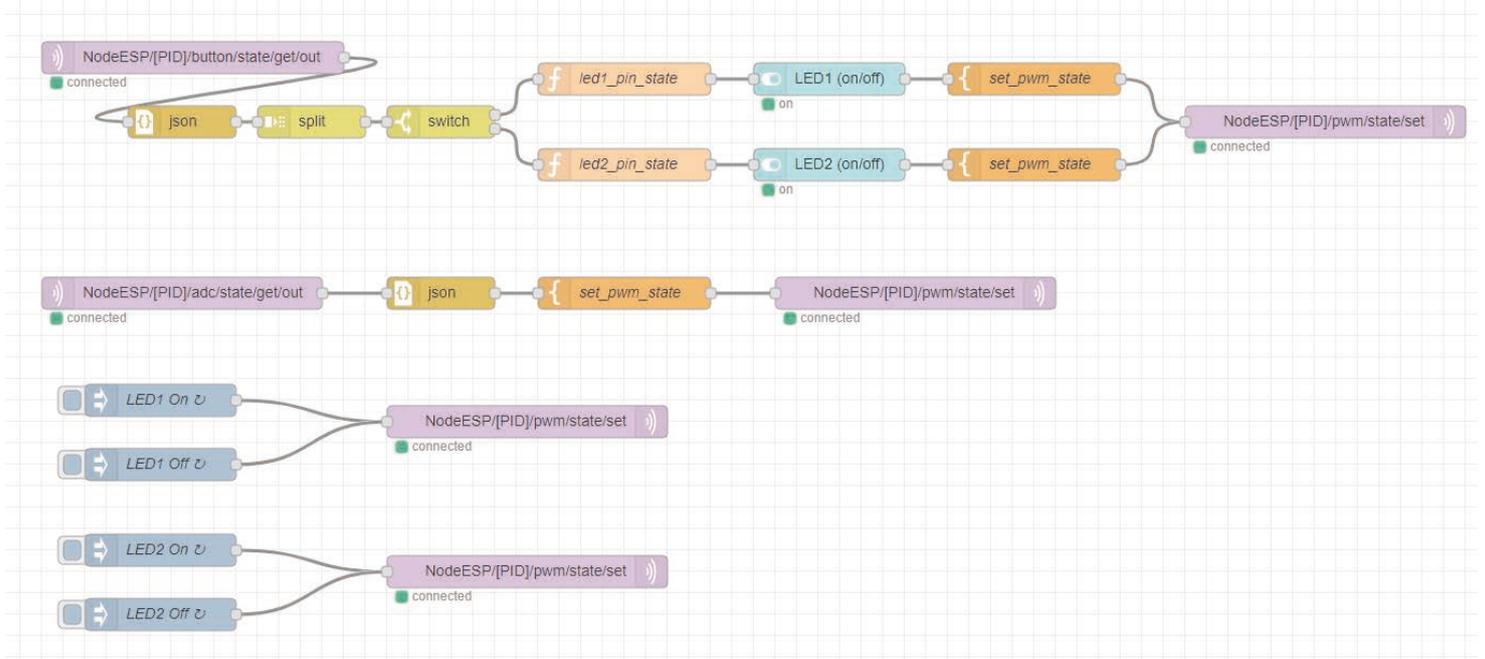
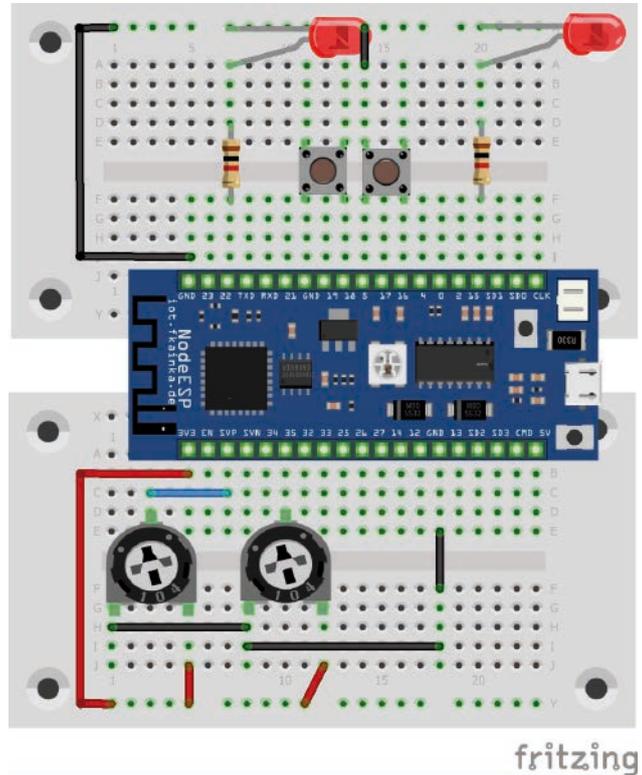
Tag 17: Zeitgesteuerte Lichter

17. Tag

Hinter dem Türchen des 17. Tages befindet sich ein weiteres Poti. Das bedeutet, dass nun für jede LED genau ein Poti zur Verfügung steht. Damit können wir das Programm von Tag 13, die smarten Lichter, um eine etwas praktischere und intuitivere Dimm-Funktion erweitern. Außerdem werden wir heute sehen, wie man mit Node-RED zeitgesteuerte Ereignisse startet.

Sie müssen das Programm des heutigen Tages in der Weboberfläche aktivieren. Anschließend können Sie den heutigen Flow in die Node-RED-Umgebung einfügen, natürlich nicht, ohne die vorherigen Flows zu deaktivieren und das MQTT-Topic an Ihre persönliche ID anzupassen.

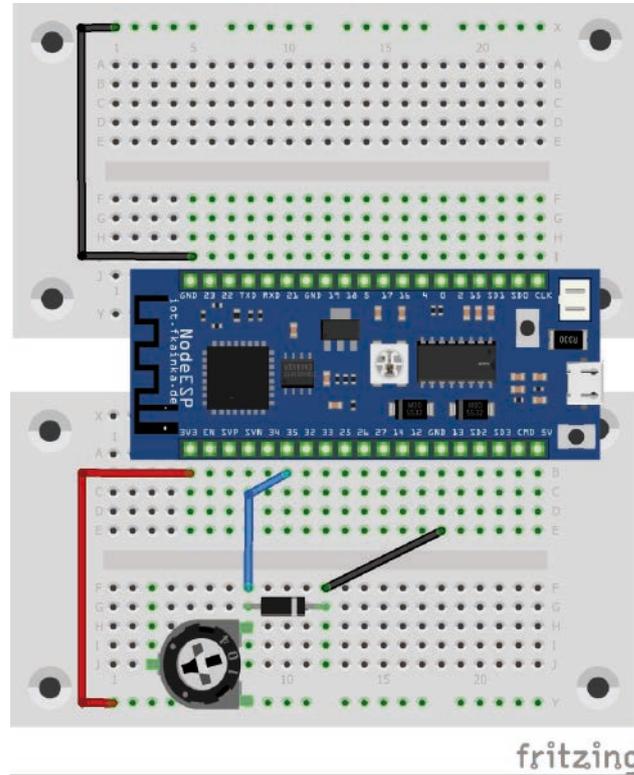
Die LEDs können nun, wie schon bekannt, über die Hardware-Buttons oder über das Userinterface an- und ausgeschaltet werden. Die Helligkeit der einzelnen LEDs wird über die Potis gesteuert. Als Alternative zu den Buttons gibt es die Inject-Nodes unten links. Über diese können Zeiten angegeben werden, zu denen ein Befehl an das Board gesendet wird. Dazu müssen Sie wieder einfach nur doppelt auf den entsprechenden Node klicken, und schon können Sie einen Zeitpunkt und auch Tage auswählen, an denen die Nachricht geschickt werden soll. Somit können Sie beispielsweise eine automatische Lampe für Räume oder ein Aquarium einstellen oder an Arbeitstagen eine spezielle Beleuchtung aktivieren - ganz automatisch und ganz im Sinne von Smarthome.



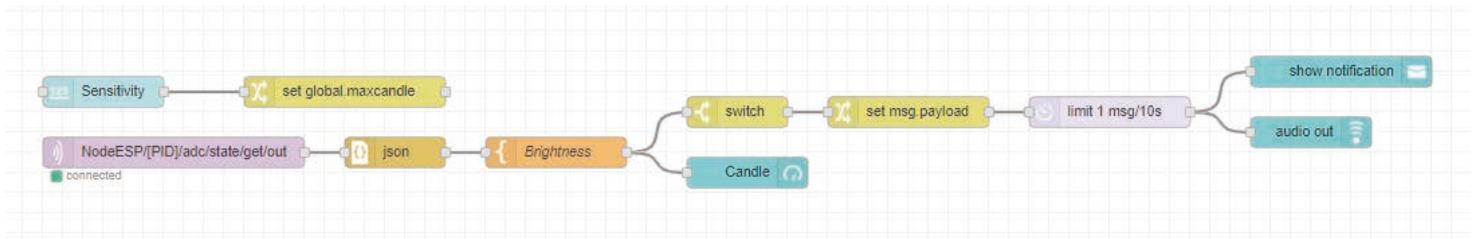
Der Flow von Tag 17.

Tag 18: Kerzenalarm

Hinter dem Türchen des heutigen Tages befindet sich eine kleine Diode. Wir werden diese Diode als Temperatursensor für besonders hohe Temperaturen nutzen. Damit ist es uns möglich, einen kleinen Kerzenwächter zu bauen.



Der Aufbau des heutigen Tages besteht, neben der Diode, aus einem Poti, das einfach als 10-kOhm-Widerstand herhält. Dieser Widerstand ist fix und kann auch nicht eingestellt werden. Das Board misst nun die Spannung an der Diode und überträgt die Werte über MQTT an Node-RED.



Der Flow des Kerzenmelders.

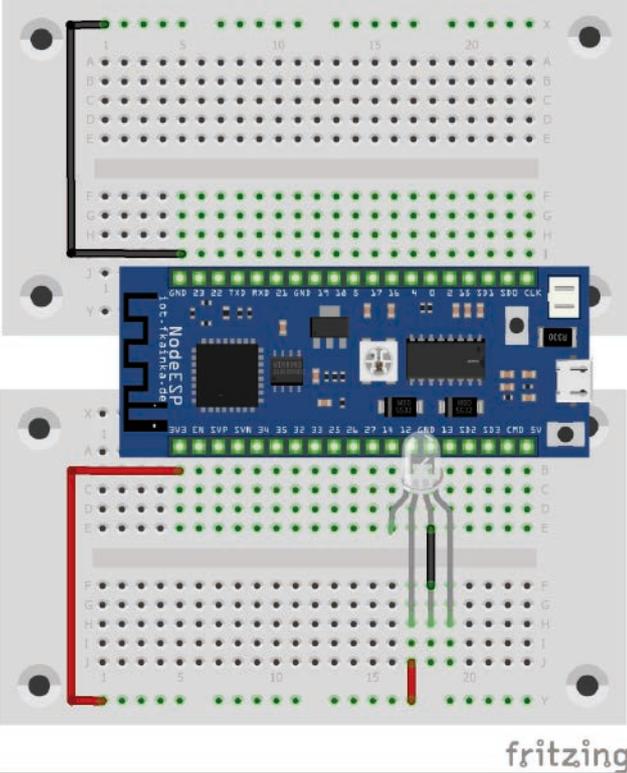
Der Flow des heutigen Tages wertet nun ständig den gemessenen Wert aus. Sollte ein bestimmter Wert unterschritten werden (je höher die Temperatur, desto kleiner die gemessene Spannung), wird ein Alarm in Form einer vorgelesenen Textnachricht und einer Infobox ausgegeben. Sie können im Userinterface einen Schwellenwert für den Alarm einstellen. Wenn Sie nun die Diode so an den unteren Teil einer Kerze anlegen, dass die Flamme beim Abrennen in die Nähe der Diode kommt, haben Sie einen praktischen Melder für eine abgebrannte Kerze.

Tag 19: Cheerlights

19. Tag

Hinter dem Türchen des heutigen Tages befindet sich eine ganz besondere LED. Es handelt sich um eine RGB-LED mit integriertem Controller (Typenbezeichnung PL9823, vergleichbar mit WS2812). Sie ermöglicht es, mehrere LEDs hintereinanderschalten und über ein Protokoll einzeln anzusteuern. Heute werden wir allerdings zunächst eine LED ansteuern. Dafür kommt ein Versuch gerade recht, der schon eine gewisse Tradition in meinen Adventskalendern hat: das Cheerlights-Projekt von Hans Schaaler. Für alle diejenigen, die das Projekt nicht kennen: Es geht um weltweit vernetzte bunte Lichter, die sich simultan über Twitter-Kommandos steuern lassen. Und dazu passt Node-RED sehr gut. Für den Versuch des heutigen Tages benötigen Sie den folgenden Aufbau und das Programm des heutigen Tages.

Die LED kann nun über eine Reihe von Kommandos gesteuert werden. Sie wird über den Begriff **Pixel** angesprochen. Am heutigen Tag werden wir vor allem sehen, wie man alle LEDs mit der gleichen Farbe ansteuern kann, denn wir haben heute ja nur eine LED. In einem späteren Versuch wird aber deutlich, mit welchen Kommandos man den LEDs einzelne Farben zuweisen kann.



So wird die PL9823 richtig angeschlossen (lange Beinchen rechts).

Kommando im in-Topic	Einstellungen
{„cmd“:„set_pixel_hex“, „value“:„#FFFFFF“}	Setzt alle LEDs auf den Farbwert im HEX-Format.
{„cmd“:„set_pixel_rgb“, „value“:„rgb(255,255,255)“}	Setzt alle LEDs auf den Farbwert im RGB-Format.
{„cmd“:„set_pixel_hsv“, „value“:„hsv(120, 100%, 100%)“}	Setzt alle LEDs auf den Farbwert im HSV-Format.
{„cmd“:„get_pixel_hex“}	Gibt die aktuelle Farbe im entsprechenden Format wieder, wird automatisch beim Setzen einer Farbe zurückgegeben.
{„cmd“:„get_pixel_rgb“}	
{„cmd“:„get_pixel_hsv“}	

Das Cheerlights Cheerlights-Projekt hat eine Webseite (<https://cheerlights.com/>), auf der Sie einige Informationen zu dem Projekt bekommen. Außerdem finden Sie dort eine Unterseite, bezeichnet mit **Cheerlights API**, auf der Sie eine Vielzahl von Möglichkeiten dazu erhalten, wie Sie den aktuellen Farbwert bekommen. Wir benutzen hier die Schnittstelle

<http://api.thingspeak.com/channels/1417/field/2/last.json>

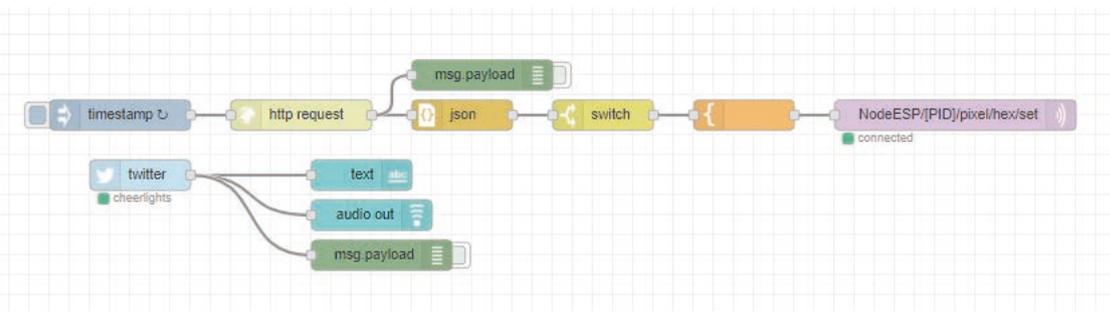
über die der aktuelle Wert als JSON-Value und im HEX-Format angeboten wird.

In dem Flow des heutigen Tages müssen wir einfach nur alle paar Sekunden diese Seite aufrufen und den aktuellen Wert extrahieren. Es wird überprüft, ob sich der Wert geändert hat. Ist das der Fall, wird die neue Farbe an das Board übertragen. Zusätzlich können Sie auch den Twitter-Node mit Ihren Twitter-Daten konfigurieren, falls Sie bei Twitter angemeldet sind. Er sucht dann ebenfalls nach Tweets mit dem Cheerlights-Text und liest die Nachricht vor, während sie auf der GUI angezeigt wird.

Doch wie können Sie die Farbe verändern? Das geht, indem Sie eine Twitter-Nachricht verfassen und damit Ihre und auch die Farbe aller anderen Cheerlights-Nutzer verändern. Die Nachricht muss zum einen #Cheerlights, @Cheerlights oder einfach nur das Stichwort Cheerlights enthalten, und zum anderen muss eine der vordefinierten Farben hinter dem Stichwort stehen. Zu den vordefinierten Farben gehören:

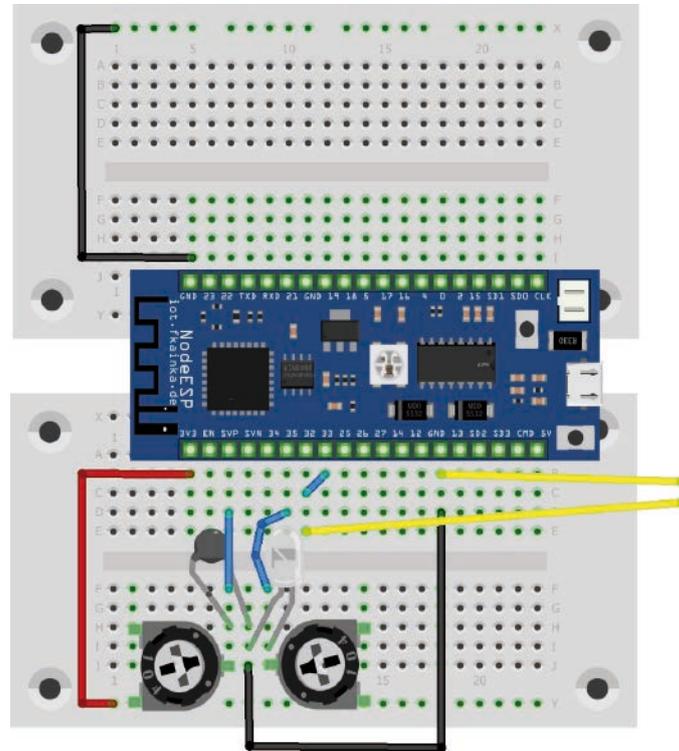
red, green, blue, cyan, white, warmwhite, purple, magenta, yellow, orange, pink

Ein möglicher Tweet lautet also: Testing my #cheerlights project on my #NodeESP with the color blue. Damit geben Sie sich und der ganzen Welt eine neue Farbe.



Tag 20: Pflanzenwächter

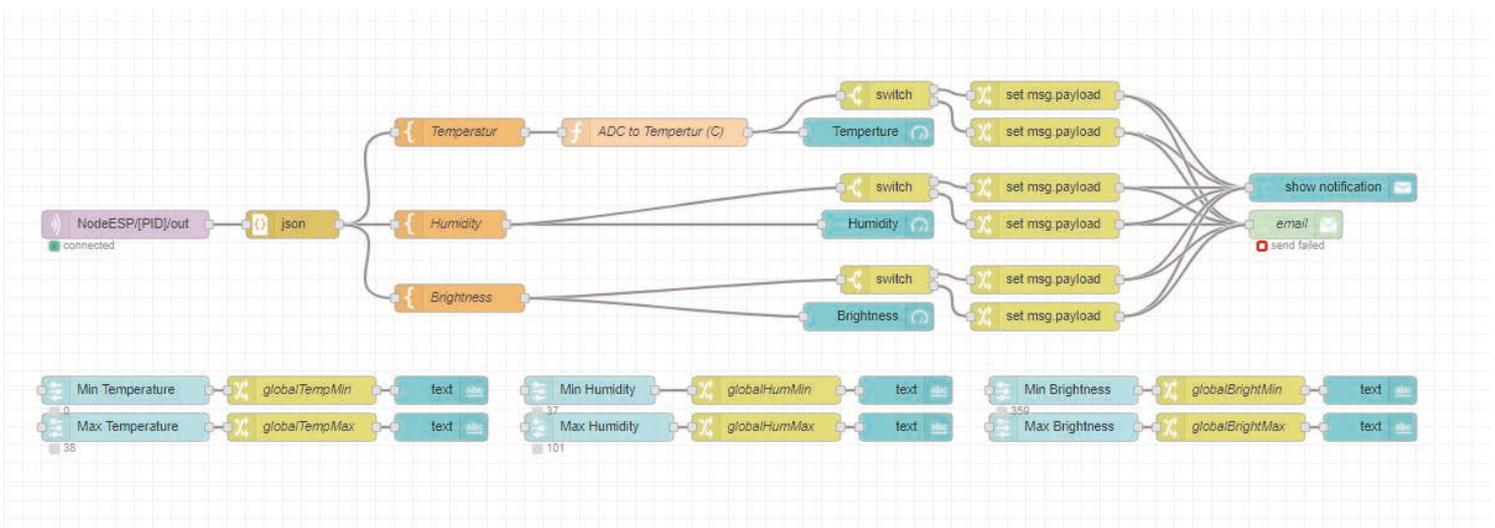
Hinter dem Türchen des heutigen Tages befindet sich ein NTC-Widerstand (**Negative Temperature Coefficient**) zur Messung der Temperatur. Zusammen mit dem Fototransistor und ein paar Drähten bauen wir damit einen Pflanzenwächter, der uns vor schlechten Bedingungen für eine zu überwachende Pflanze warnt. Dazu benötigen wir den folgenden Aufbau und das Programm des aktuellen Tages.



fritzing

Der Pflanzenwächter.

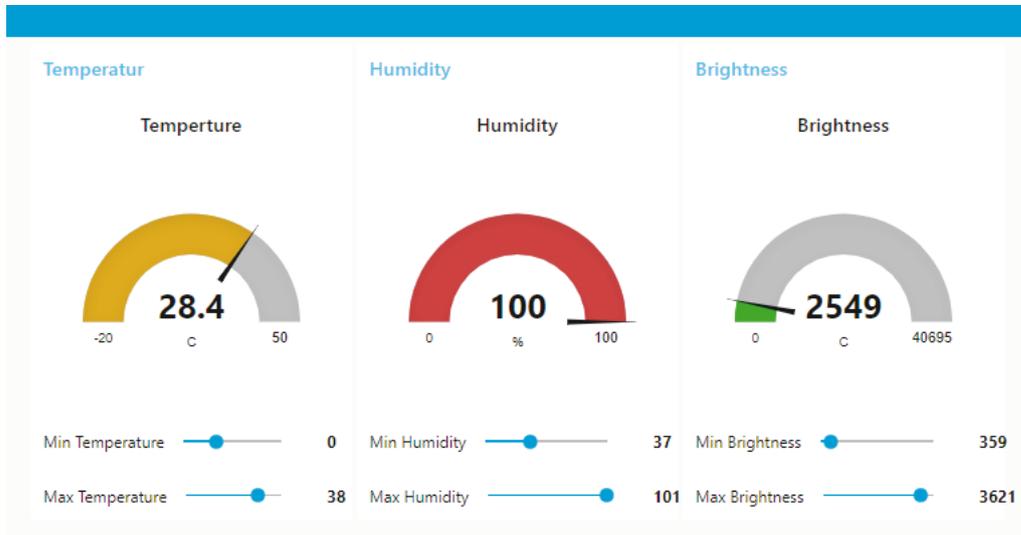
Die beiden gelben Drähte sind dabei einfache Drähte, die in die Pflanzenerde gesteckt werden können. Sie sollen überwachen, wie feucht die Erde ist. Für erste Tests reicht es aber aus, beide Drähte mit einem Finger zu berühren.



Der Flow des Tages kann wieder einfach in die Node-RED-Umgebung importiert werden, natürlich nicht, ohne die vorherigen zu deaktivieren und das Topic anzupassen. Über das **out**-Topic wird nun ein JSON-String empfangen, der in etwa so aussieht:

```
{ "cmd": "get_hum_state", "Light": 3903, "Temp": 368, "HUMIDITY": 45 }
```

Es werden also die Daten Licht, Temperatur und Feuchtigkeit übertragen. Licht ist hierbei wieder ein relativer Wert mit einem Wertebereich von 0 bis 4095, genau wie die gemessene Feuchtigkeit. Die Temperatur wird ebenfalls in diesem Wertebereich übermittelt. Da damit aber niemand etwas anfangen kann, gibt es einen Function-Node, der die Werte in Celsius umrechnet. Dargestellt werden die Werte praktisch auf der Nutzeroberfläche. Neben den Anzeigen finden sich dazu noch ein paar Regler.



Das Userinterface des Pflanzenwächters.

Diese Regler sind dazu da, Grenzbereiche für die Messwerte festzulegen. Man kann jeweils einen Minimal- und einen Maximalwert bestimmen. Wenn man auf einen Wert oder eine Grenze nicht testen möchte, kann man den Regler auch auf eine Extremposition setzen, die niemals vom Board gemessen werden kann. Wenn man also beispielsweise nicht möchte, dass nachts der Alarm für eine zu geringe Helligkeit angeht, kann man den Minimalregler auf -1 stellen.

Sollte einer der Grenzwerte überschritten werden, triggert der entsprechende Switch-Node, und eine Nachricht wird eingestellt, die dann über die UI ausgegeben wird. Da man diese häufig nicht geöffnet hat, kann man zusätzlich seine E-Mail-Zugangsdaten in den E-Mail-Node eintragen. Dann wird eine E-Mail gesendet, die daran erinnert, dass man mal wieder die Pflanze gießen sollte oder für bessere Lichtverhältnisse zu sorgen hat. So wird selbst die einfache Zimmerpflanze smart.

21. Tag

Tag 21: Piano-Player

Hinter dem Türchen des 21. Tages finden Sie einen Lautsprecher. In dem heutigen Versuch werden wir damit etwas Musik abspielen. Dazu benötigen wir den folgenden einfachen Aufbau.

In diesem Versuch ist der Lautsprecher an einem der beiden DAC-Ausgänge des Boards angeschlossen. DAC bedeutet **Digital/Analog-Converter** und ermöglicht es, durch digitale Befehle ein analoges Signal zu erzeugen. In dem heutigen Versuch werden wir damit Töne erzeugen, genauer gesagt Töne wie von einem Piano. Die Noten, Pausen und Tonarten werden dabei per UDP übertragen, und zwar als Array von Zahlen.

Kopieren Sie zunächst den Node-RED-Flow des heutigen Tages in Ihre Umgebung, natürlich nicht, ohne die vorherigen zu deaktivieren. Außerdem müssen Sie noch die IP des UDP-Nodes auf die Broadcast-IP Ihres Boards anpassen. Wählen Sie zudem das Programm des heutigen Tages. Dann können Sie den Flow starten.

Es gibt bereits vordefinierte Songs, die über das Userinterface gestartet werden können. Sie dürfen allerdings auch selbst Songs übersetzen oder komponieren. Dazu gibt es ein Textfeld, in das Sie Ihre Komposition kopieren können.

Die Noten entsprechen dabei dem einfachen Notenschema und werden in der folgenden Form angegeben:

NOTE_C1, NOTE_CS1, NOTE_D1, NOTE_DS1, NOTE_E1, NOTE_F1, NOTE_FS1,
NOTE_G1, NOTE_GS1, NOTE_A1, NOTE_AS1, NOTE_B1

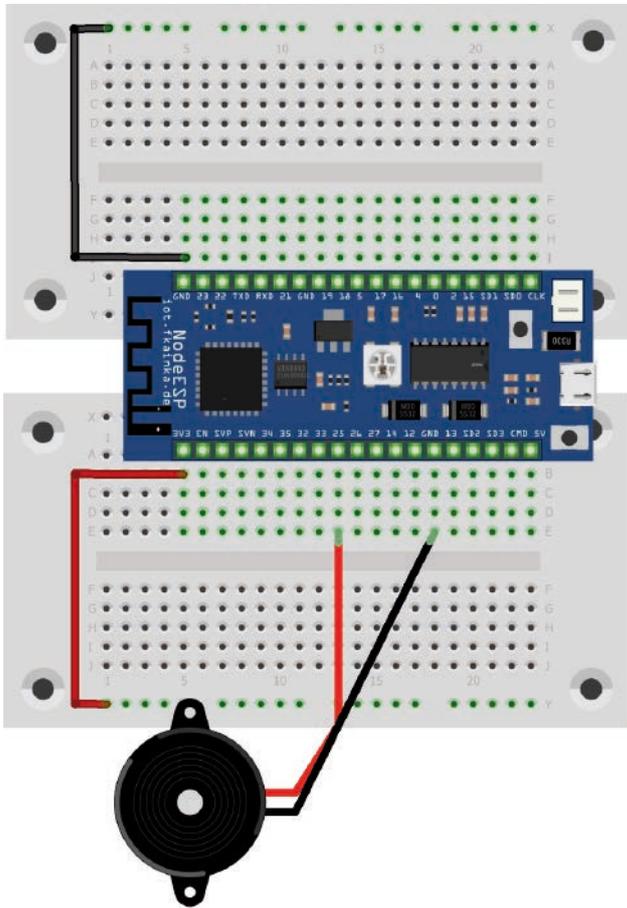
Die Zahl hinter der Note entspricht der Oktave. Sie können die 1 durch eine Zahl von 1 bis 7 ersetzen. Außerdem können Sie die Notenlänge mit dem Präfix BEAT angeben, um so halbe, Viertel- oder Achtelnoten daraus zu machen. Das BEAT-Attribut bezieht sich immer auf die davorstehende Note. Standard ist BEAT_1, daher muss es nicht extra vor jeder Note angegeben werden. Es gelten folgende Notenlängen:

BEAT_025, BEAT_05, BEAT_075, BEAT_1, BEAT_125, BEAT_15, BEAT_175, BEAT_2,
BEAT_25, BEAT_3, BEAT_4, BEAT_5

Um zum Beispiel den Anfang von Beethovens berühmter 5. Symphonie zu spielen, kann folgender Text in das Fenster eingegeben werden:

NOTE_G5, BEAT_075, NOTE_G5, BEAT_075, NOTE_G5, BEAT_075, NOTE_E5, BEAT_5,
NOTE_F5, BEAT_075, NOTE_F5, BEAT_075, NOTE_F5, BEAT_075, NOTE_D5, BEAT_5

Sie müssen allerdings kein Musikgenie sein. Probieren Sie einfach ein paar Noten und Notenlängen aus und komponieren Sie so Ihr eigenes Lied. Oder setzen Sie ein bekanntes Lied in dieses System um. Ihrer Fantasie sind dabei keine Grenzen gesetzt.



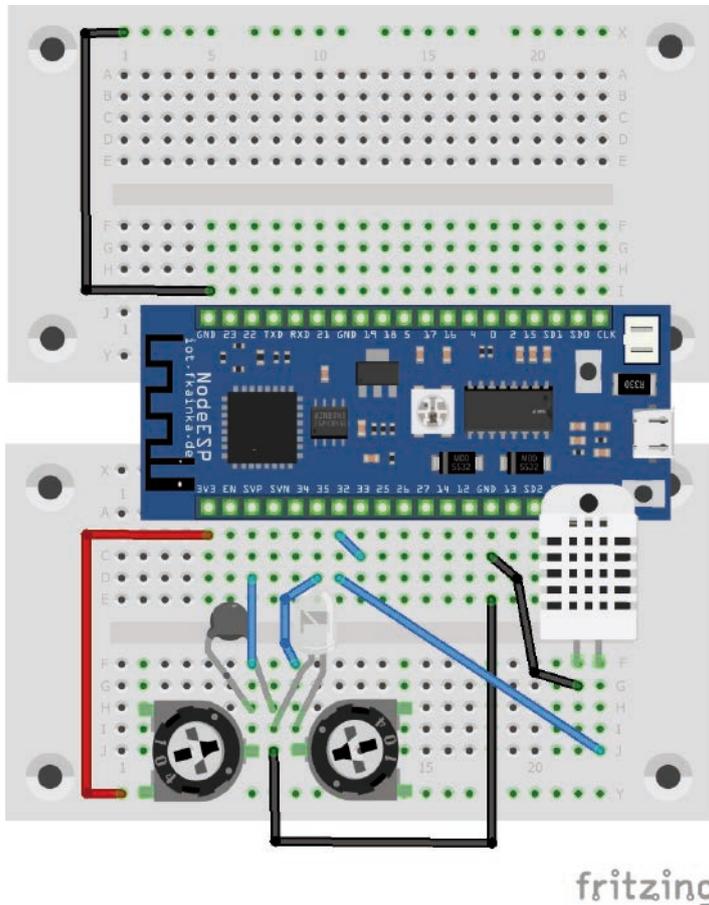
fritzing

Der Flow des heutigen Tages.

Tag 22: Klimawächter

Hinter dem Türchen des 22. Tages befindet sich ein neuer Sensor, der die aktuelle Luftfeuchtigkeit messen kann. Wir benutzen ihn, um einen vollständigen Raumklimawächter zu realisieren. Dazu benötigen wir die folgende Schaltung.

22. Tag



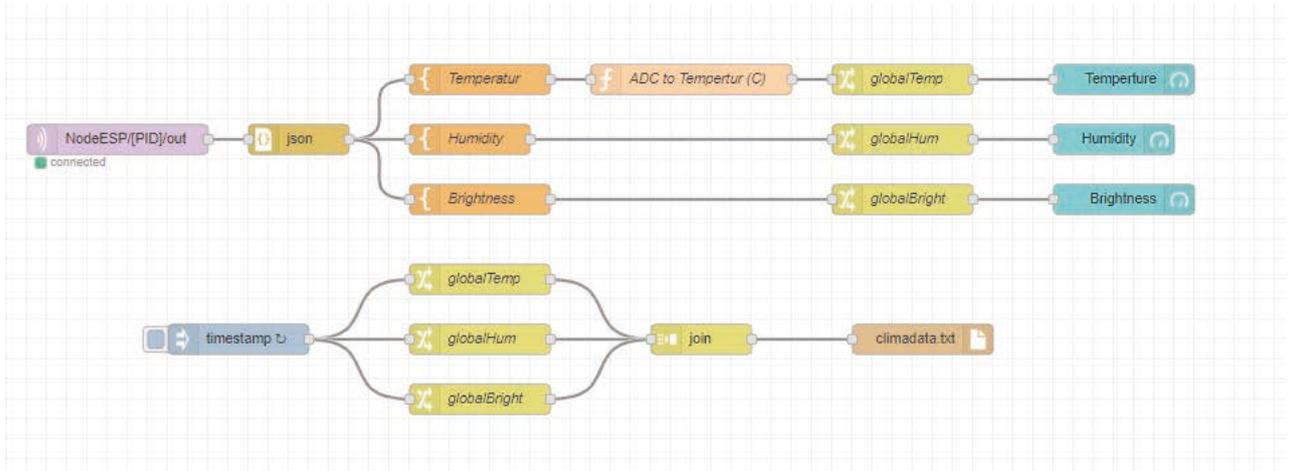
Die Schaltung des Raumklimawächters.

Der Versuch des heutigen Tages vereint drei Sensoren: den Temperatursensor (NTC), den Lichtsensor (Fototransistor) und den Luftfeuchtesensor. Zusammen können sie einen recht guten Eindruck der aktuellen Klimalage in einem Zimmer abgeben. Dazu müssen Sie aber zunächst das Programm des heutigen Tages starten.

Das Programm misst alle 30 Sekunden alle Werte und überträgt sie über MQTT, was beispielsweise so aussieht:

```
{"cmd": "get_hum_state", "Light": 3903, "Temp": 368, "HUMIDITY": 45}
```

Wenn Sie den Flow des heutigen Tages in Ihre Node-RED-Umgebung kopieren, natürlich nicht, ohne das persönliche Topic anzupassen und die vorherigen Flows zu deaktivieren, können Sie auf der GUI-Seite die gemessenen Werte einsehen. Ein Function-Node hat auch heute die Temperatur auf einen passenden Wert umgerechnet. Zur genaueren Kalibrierung sind die beiden Potis da, mit denen Sie die Werte Temperatur und Umgebungshelligkeit anpassen können.



Der Flow zur Langzeitmessung.

Es gibt aber noch ein Zusatz-Feature. Alle fünf Minuten speichert das Programm nämlich die aktuell gemessenen Werte in eine Datei. Diese trägt den Namen **climadata.txt** und befindet sich in der Regel in dem Verzeichnis, in dem Node-RED arbeitet. Bei mir war dies:

C:\Users\[USERNAME]

[USERNAME] steht für den eigenen Windows-Benutzernamen. Sie können den Dateipfad aber auch in dem Node anpassen. Gespeichert wird übrigens in einem CSV-Format (**Comma Separated Values**), bei dem alle Werte durch ein Komma getrennt sind. Somit können Sie die Daten auch einfach in eine Excel-Tabelle einfügen und dort auswerten. Die Auswertung der Langzeitmessung wird dadurch kinderleicht.

Tag 23: Alarmanlage

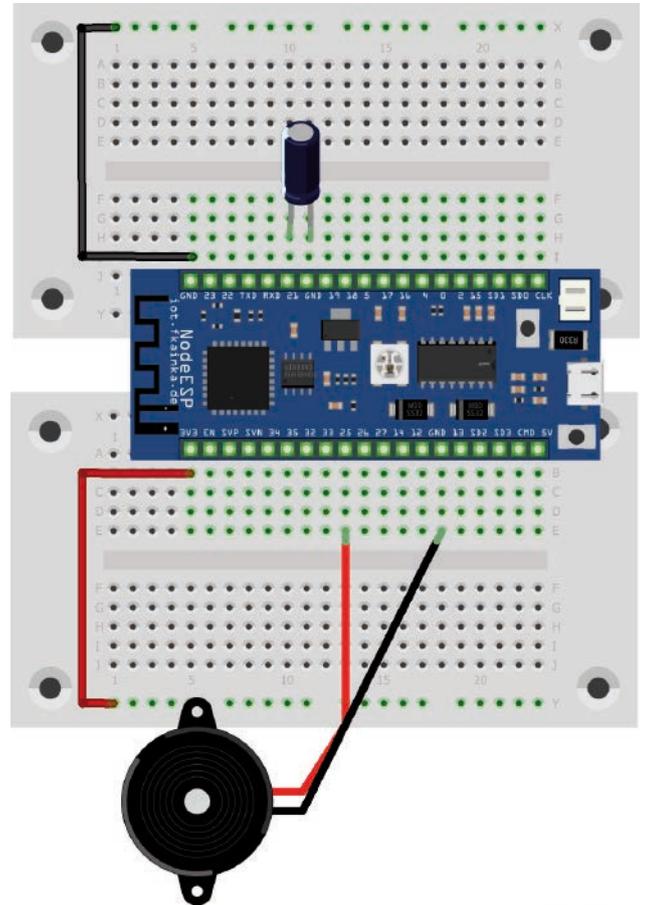
23. Tag

Hinter dem vorletzten Türchen des Adventskalenders befindet sich ein sogenannter Tilt-Sensor (Neigungssensor). Der Sensor erkennt, wenn er gekippt wird. Wir setzen ihn heute als Sensor für eine Alarmanlage ein. Der Lautsprecher von vor zwei Tagen dient uns dabei als Signalgeber. Dazu wird der folgende Aufbau benötigt.

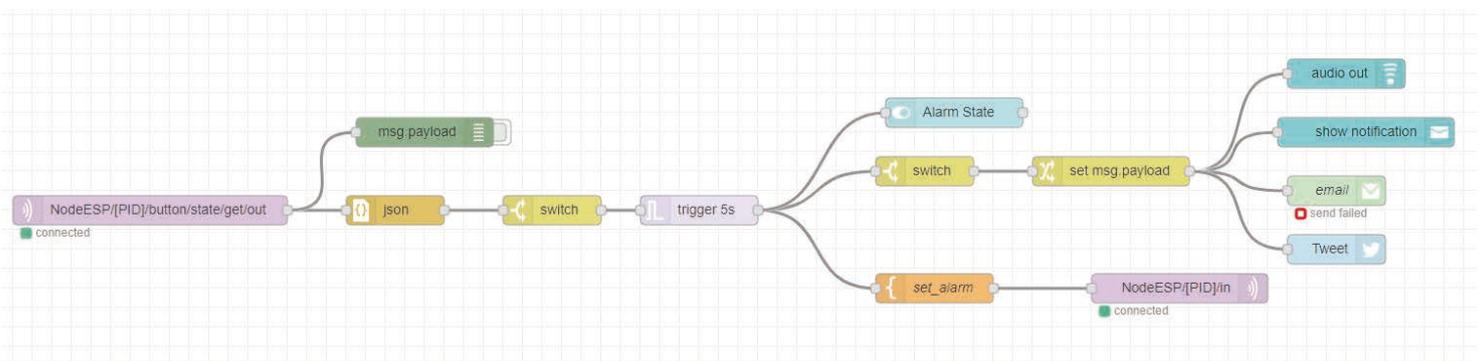
Aktivieren Sie zunächst wie gewohnt das Programm des heutigen Tages und fügen Sie anschließend den Node-RED-Flow in die Umgebung ein. Passen Sie außerdem das MQTT-Topic an und deaktivieren Sie alle vorangegangenen Flows.

Der Flow wartet nun darauf, dass der Tilt-Sensor ausgelöst und eine Nachricht in das entsprechende Topic gepublikt wird. Dies löst einen Trigger-Node aus, der das Payload auf TRUE setzt. Dieser wird zum einen über ein Topic wieder an das Board gesendet, das einen Alarm auslöst. Der Trigger-Node sendet dann nach fünf Sekunden ein FALSE, um den Alarm wieder zu deaktivieren.

Gleichzeitig mit dem Alarmsignal über das Board gibt es in der oberen rechten Ecke noch eine Reihe von weiteren Nodes, die einen Alarm signalisieren. Zum einen gibt es heute einen Audio-out-Node, der über den Alarm informiert. Eine Textnachricht erscheint ebenfalls auf dem Userinterface. Außerhalb des Userinterface können Sie sich per Mail oder Twitter über einen Alarm informieren lassen. Ebenfalls möglich ist es, Nodes nachzuinstallieren, die Push-Nachrichten an das Smartphone senden können. **Pushbullet** ist beispielsweise einer dieser Services mit einer praktischen App. Somit ist Ihr smartes Heim auch vor Einbruch geschützt.



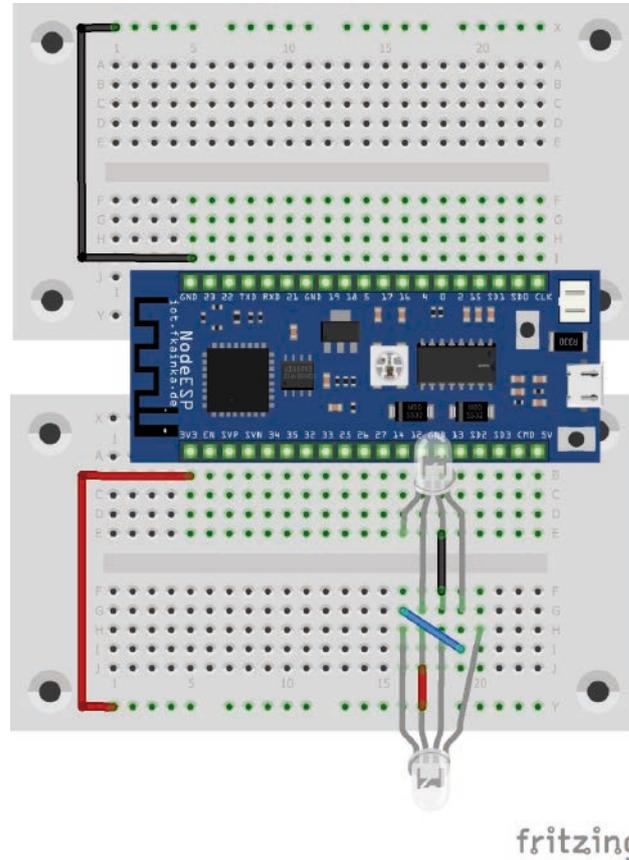
fritzing



Der Flow der Alarmanlage.

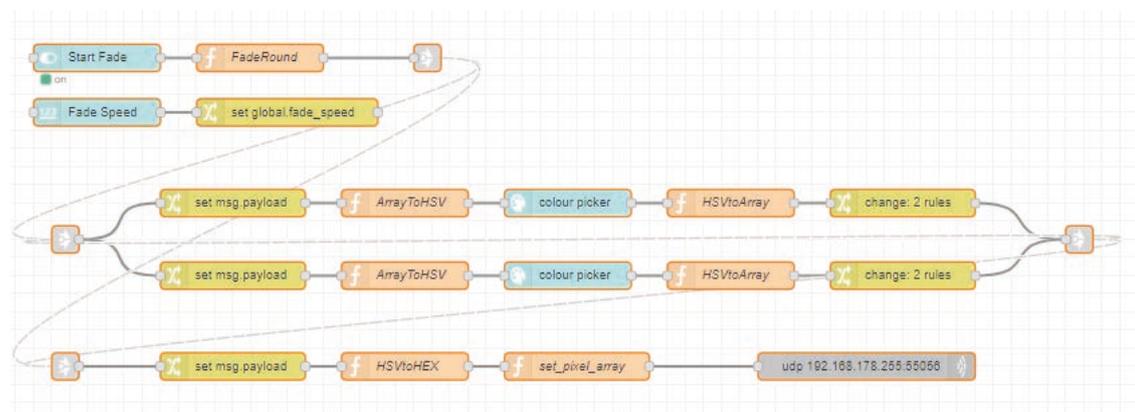
Tag 24: Tannenbaumbeleuchtung

Hinter dem letzten Türchen des Adventskalenders befindet sich eine weitere programmierbare RGB-LED. Damit haben Sie jetzt zwei LEDs, die Sie zu einer kleinen Weihnachtsbaumbeleuchtung verbinden können. Dazu benötigen Sie den folgenden Aufbau.



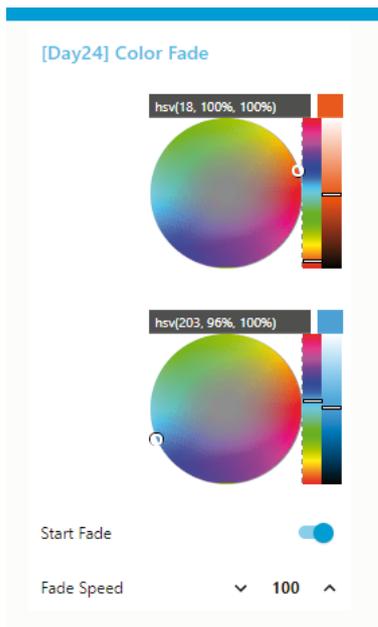
Der Aufbau des Weihnachtsbaumlichts. Die langen Beinchen der LEDs befinden sich rechts

Aktivieren Sie dann das Programm und importieren Sie den Flow des heutigen Tages. Deaktivieren Sie alle vorangegangenen Flows und ändern Sie gegebenenfalls die Broadcast-IP des UDP-Nodes.



Der letzte Flow des Adventskalenders.

Der Flow enthält wieder ein paar Userinterface-Elemente, allen voran zwei Color-Picker-Elemente, über die Sie die beiden LEDs steuern können. Es gibt aber noch zwei Elemente, und zwar ein Switch und ein Element zum Einstellen einer Zahl. Wenn Sie den Button drücken, sehen Sie, wie sich die beiden Color-Picker-Elemente verändern. Der Kreis, der den aktuellen Farbwert anzeigt, wandert laufend im Kreis herum. Gleichzeitig ändern sich die Farben der beiden LEDs am NodeESP-Board. Sie können auch die Fade-Speed-Zeit verändern und so einen langsameren oder schnelleren Wechsel einstellen. Außerdem können Sie während des Fade-Vorgangs die Farbe einer LED verändern. Der Verlauf geht dann ab dem neuen Punkt weiter. Das Gesamtergebnis ist eine kleine schöne Beleuchtung, die Sie sich in den Weihnachtsbaum hängen können. In diesem Sinne wünsche ich Ihnen ein schönes Weihnachtsfest.



Das Userinterface mit den beiden Color-Pickern.

Vorsichtsmaßnahmen

Auf keinen Fall irgendwelche GPIO-Pins miteinander verbinden und abwarten, was passiert.

Nicht alle GPIO-Pins lassen sich frei programmieren. Einige sind für die Stromversorgung und andere Zwecke fest eingerichtet.

Warnung! Augenschutz und LEDs:

Blicken Sie nicht aus geringer Entfernung direkt in eine LED, denn ein direkter Blick kann Netzhautschäden verursachen! Dies gilt besonders für helle LEDs im klaren Gehäuse sowie in besonderem Maße für Power-LEDs. Bei weißen, blauen, violetten und ultravioletten LEDs gibt die scheinbare Helligkeit einen falschen Eindruck von der tatsächlichen Gefahr für Ihre Augen. Besondere Vorsicht ist bei der Verwendung von Sammellinsen geboten. Betreiben Sie die LEDs so, wie in der Anleitung vorgesehen, nicht aber mit größeren Strömen.

Liebe Kunden!



Dieses Produkt wurde in Übereinstimmung mit den geltenden europäischen Richtlinien hergestellt und trägt daher das CE-Zeichen. Der bestimmungsgemäße Gebrauch ist in der beiliegenden Anleitung beschrieben.

Bei jeder anderen Nutzung oder Veränderung des Produktes sind allein Sie für die Einhaltung der geltenden Regeln verantwortlich. Bauen Sie die Schaltungen deshalb genau so auf, wie es in der Anleitung beschrieben wird. Das Produkt darf nur zusammen mit dieser Anleitung weitergegeben werden.



Das Symbol mit der durchgekreuzten Mülltonne bedeutet, dass dieses Produkt getrennt vom Hausmüll als Elektroschrott dem Recycling zugeführt werden muss. Wo Sie die nächstgelegene kostenlose Annahmestelle finden, sagt Ihnen Ihre kommunale Verwaltung.

© 2018 Franzis Verlag GmbH, Richard-Reitzner-Allee 2, 85540 Haar

Alle Rechte vorbehalten, auch die der fotomechanischen Wiedergabe und der Speicherung in elektronischen Medien. Das Erstellen und Verbreiten von Kopien auf Papier, auf Datenträgern oder im Internet, insbesondere als PDF, ist nur mit ausdrücklicher Genehmigung des Verlags gestattet und wird widrigenfalls strafrechtlich verfolgt.

Die meisten Produktbezeichnungen von Hard- und Software sowie Firmennamen und Firmenlogos, die in diesem Werk genannt werden, sind in der Regel gleichzeitig auch eingetragene Warenzeichen und sollten als solche betrachtet werden. Der Verlag folgt bei den Produktbezeichnungen im Wesentlichen den Schreibweisen der Hersteller.