

# Inhaltsverzeichnis

<b>1</b>	<b>Node-RED - Allgemein</b>	<b>2</b>
1.1	Node-RED Vorteile . . . . .	2
1.2	Node-RED in der Projektarbeit . . . . .	2
1.3	Node-RED Installation/Inbetriebnahme . . . . .	2
1.4	Browserbasierte Entwicklungsoberfläche . . . . .	3
1.5	Visualisierung inNode-RED . . . . .	3
1.6	Node-RED Bedienoberfläche . . . . .	4
<b>2</b>	<b>Flow/Ausarbeit</b>	<b>4</b>
2.1	MainDashboard -Flow . . . . .	5
2.2	HMPresenceSensor-Flow . . . . .	5
2.3	HMDoorSensor-Flow . . . . .	6
2.4	HMPowermeter-Flow . . . . .	6
<b>3</b>	<b>Subflow</b>	<b>7</b>
3.1	PirSensor-Subflow . . . . .	7
3.2	UltraSchallSensor -Subflow . . . . .	9
<b>4</b>	<b>Vergleich zu anderen IoT Tools</b>	<b>11</b>
<b>5</b>	<b>Node-RED Fazit</b>	<b>11</b>

# 1 Node-RED - Allgemein

Node-RED ist ein Werkzeug für die datenflussbasierte visuelle Programmierung in einer browserbasierten Entwicklungsumgebung. Mit Node-RED ist es möglich verschiedene Geräte, APIs (Programmierschnittstellen) und Onlinedienste in einer grafischen Oberfläche auf einfache Weise virtuell zu verbinden. Dadurch ist es unter der Verwendung der bereits vorinstallierten „Packages“ (Pakete mit bestimmten Funktionsweisen) Datenflüsse logisch miteinander zu verknüpfen. Zusätzlich besteht die Möglichkeit eigene Nodes in JavaScript zu programmieren oder mithilfe des bereits integrierten Palletten Manager benötigte Nodes nachträglich zu installieren. Da Node-RED sehr leichtgewichtig ist, eignet es sich ideal für die Nutzung auf kostengünstiger Hardware wie dem Raspberry Pi.

## 1.1 Node-RED Vorteile

- Einfache Handhabung via Drag and Drop
- Ideal für die Implementierung von einfachen Abläufen
- Einfach zu erweitern
- Schnelle Gestaltung von interessanten Weboberflächen
- plattformübergreifend, da Node.js als Runtime

## 1.2 Node-RED in der Projektarbeit

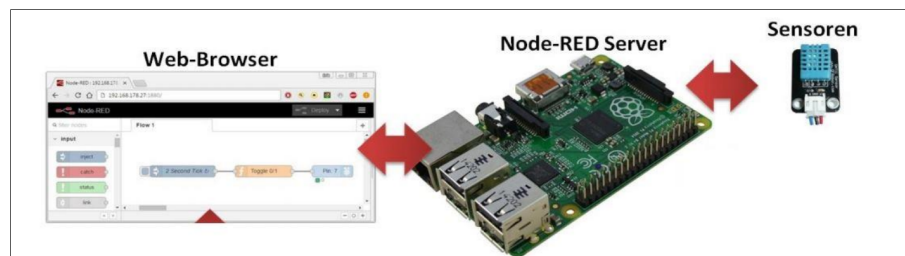


Abbildung 1: Anwendungsschema von Node-RED in der Projektarbeit

Das Tool Node-RED wird auf einem Raspberry Pi installiert, welcher als Webserver für die Erstellung des Dashboards dient. In der Entwicklungsoberfläche werden die logischen Programmabläufe via „Drag and Drop“ erstellt und Funktionsabläufe in der Programmiersprache JavaScript entwickelt. Die Sensordaten werden von den Modulen (HomeMatic und eigene Sensoren) an den Raspberry weitergeleitet und abschließend in Node-RED verarbeitet und visualisiert.

## 1.3 Node-RED Installation/Inbetriebnahme

Die Installation von Node-RED wird in Abschnitt ?? auf Seite ?? beschrieben.

## 1.4 Browserbasierte Entwicklungsoberfläche



Abbildung 2: Aufbau der Entwicklungsoberfläche in Node-RED

Nach der Installation ist die Node-RED Oberfläche im Browser unter folgender URL zu erreichen:

```
1 http://ip-des-raspberry:1880
```

Die schlicht gehaltene Node-RED-Entwicklungsoberfläche lässt sich in drei Bereiche einteilen.

Im 1. Bereich (gelbe Markierung) ist die Node-Palette welche die mitgelieferten Nodes enthält.

Der 2. Bereich beschreibt den eigentlichen Arbeitsbereich, dieser Bereich wird Flow, Arbeitsblatt oder Workspace genannt. Hier werden die logischen datenflussbasierten Abläufe und Funktionen mittels Drag & Drop der Nodes aus der Node-Palette erstellt. Durch das Plus-Zeichen in der oberen rechten Ecke im roten Bereich können mehrere Arbeitsblätter erstellt werden.

Im letzten 3. Bereich befindet sich das Informations-bzw. Debug-Panel. Dort werden Info-Texte zu den einzelnen Nodes und Anleitungen für die Konfiguration angezeigt. Der Debug-Tab zeigt die Ausgaben der Debug-Nodes und listet interne Fehler auf. Durch die rote Schaltfläche „Implementieren“ oben rechts, werden alle Flows in den Raspberry Pi geladen und ausgeführt.

## 1.5 Visualisierung inNode-RED

In der Grundausstattung sind bereits einige nützliche Nodes installiert. Werden für die Nutzung noch weitere Nodes benötigt, können diese direkt in der Web-Oberfläche nachinstalliert werden. Hier kann nach neuen Nodes gesucht und diese per Mausklick installiert werden.

Für die Anzeige von Visualisierungen in dem Dashboard wird das „node-red-dashboard“ Package benötigt. Für die Installation wird der Menüpunkt „Palette verwalten“ im Drop Down Menü oben rechts (3 horizontale Striche) ausgewählt. Durch die Eingabe „dashboard“ in die Suchleiste, wird das notwendige Package angezeigt. Mit der Bestätigung „installieren“ werden die benötigten Module in Node-RED hinzugefügt.

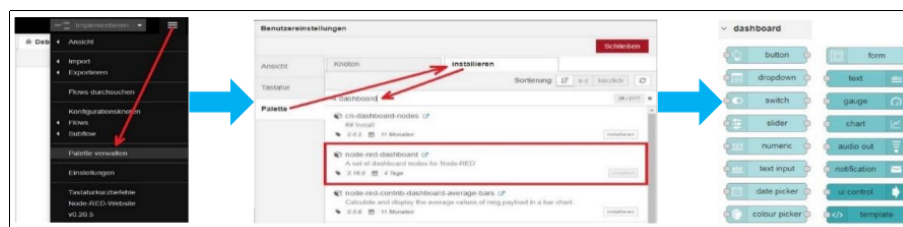


Abbildung 3: Palette verwalten Dienst für nachträgliche Package-Installationen

## 1.6 Node-RED Bedienoberfläche

Aufgerufen wird die Web-Oberfläche mit folgender URL:

```
1 http://ip-des-raspberry:1880/ui
```

Der Aufbau der Gruppen innerhalb des User Interfaces kann über das Dashboard-Panel, welches als weiterer Tab im Info-/Debug-Panel angezeigt wird, vorgenommen werden. In der Konfiguration können die Elemente auf mehrere Tabs und in Gruppen aufgeteilt werden. Gehören Elemente der gleichen Gruppe an, teilen sie sich eine Kachel in der Bedienoberfläche.

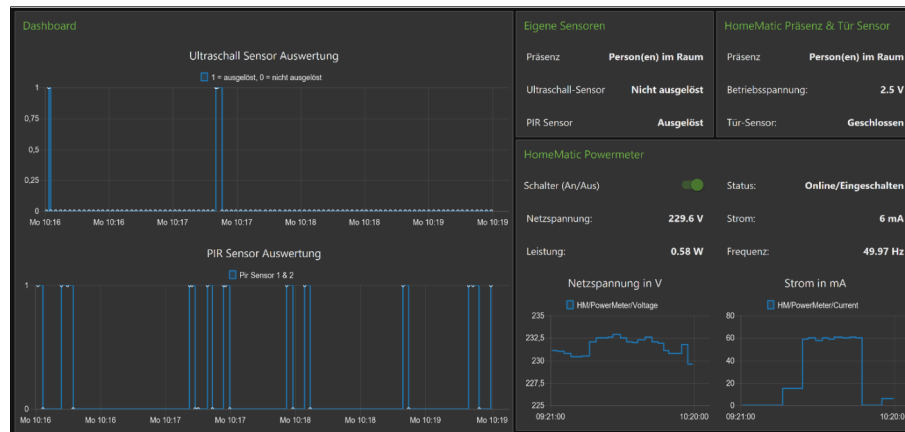


Abbildung 4: Bedienoberfläche im Webbrowser

Für die Projektarbeit wurde das Dashboard in vier logisch zusammenhängende Gruppen eingeteilt. Jede einzelne Gruppe repräsentiert eine Auswertung von Sensordaten mit visueller Darstellung. Dabei wurde zwischen kommerziellen und den selbstgebaute Sensoren differenziert.

Durch die beiden Gruppen *HomeMatic Präsenz & Tür Sensor* und *HomeMatic Powermeter* werden die kommerziellen Sensoren von HomeMatic vertreten, darunter ist der Präsenz Melder, die Steckdose und der Tür Sensor. Die verschiedenen Sensoren von HomeMatic wurden von dem Kommilitonen Sebastian Dohle angelernt und angebracht. Innerhalb der *HomeMatic Powermeter* Gruppe werden die Sensordaten der Steckdose angezeigt, darunter auch ein grafischer Verlauf der Netzspannung und des Stroms in einer Time-Chart. Bei *HomeMatic Präsenz & Tür Sensor* werden die Standard-Ausgabewerte der Sensoren angezeigt. Darunter der aktuelle Status des Tür-Sensors, also ob offen oder zu und die aktuelle Präsenzmeldung.

Hinsichtlich der Gruppe *Dashbaord* (links) sind die grafischen Auswertungen der eigenen Sensordaten dargestellt. Hierbei ist in der oberen Hälfte der Ultraschall Sensor visualisiert. Es werden nur die vereinfachten Werte 1 oder 0 angezeigt. Das gleiche gilt für den PIR Sensor in der unteren Hälfte. Die Sensordaten 1 oder 0 zeigen an ob der jeweilige Sensor ausgelöst hat oder nicht. Rechts dazu in der *Eigene Sensoren* Gruppe steht die textuelle Auswertung und eine logische Auswertung mit „Person(en) im Raum“ oder „Keine Person(en) im Raum“.

## 2 Flow/Ausarbeit

In Node-RED gibt es die Möglichkeit Arbeitsblätter anzulegen. Dadurch lassen sich die verschiedenen Sensoren in logische Gruppen verpacken. Das Projekt bekommt dadurch eine bessere Übersicht und Strukturierung. Jeder einzelner Flow beschreibt dabei eine eigene Logik, welche die Sensordaten auswertet und visualisiert.

Bei der datenflussbasierten Entwicklung in Node-RED wird der Datenaustausch durch sogenannte Message-Objekte umgesetzt.

## 2.1 MainDashboard -Flow

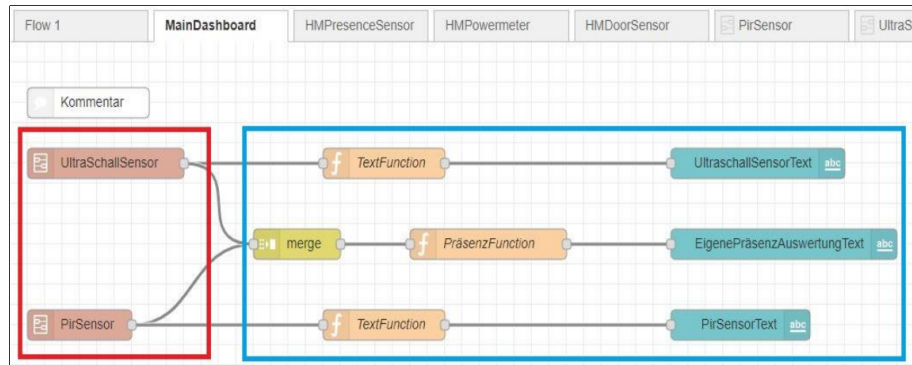


Abbildung 5: Auswertung und Visualisierung des PIR- und Ultraschallsensors

Für die visuelle Darstellung des PIR- und Ultraschallsensors ist das Flow „MainDashboard“ zuständig (s. Abb. 5). Zu Beginn sind im roten Bereich die beiden Subflows „UltraSchallSensor“ und „PirSensor“ erkennbar. Diese repräsentieren dabei eine selbst entwickelte Logik und geben ihre Sensordaten weiter an das gelbe „merge“ Node im blauen Kasten. Durch den „merge“ Baustein werden die beiden Datenströme miteinander kombiniert. Dadurch können diese Datenströme anschließend leichter durch den „PräsenzFunktion“ Node ausgewertet werden. Die beiden äußeren Zweige, haben eine einfache „TextFunction“ welche eine textuelle Darstellung der beiden Werte „Ausgelöst“ und „Nicht Ausgelöst“ in der UI anzeigen.

## 2.2 HMPresenceSensor-Flow

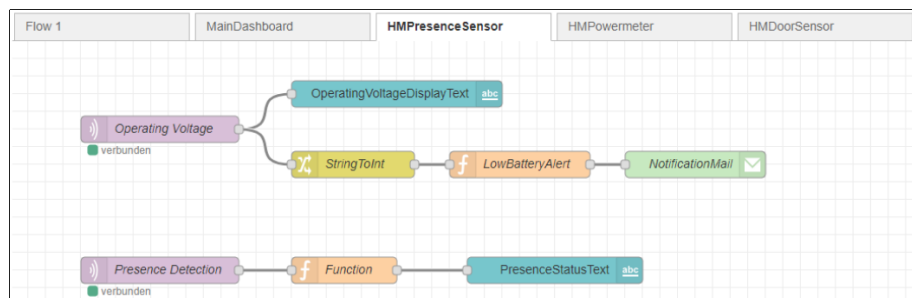


Abbildung 6: Flow für den HomeMatic IP Präsenzmelder

In diesem Flow (s. Abb. 6) wird die Logik von dem HomeMatic IP Präsenzmelder definiert. Zu Beginn wird eine Subscription durch die beiden violetten Nodes (linke Seite) erstellt, somit erhalten wir die Daten von vorgenanntem Sensor. Je nachdem, welchen Wert wir erhalten wird dementsprechend eine Weiterverarbeitung durch Function Nodes vorgenommen und anschließend durch einen Text auf der Bedienoberfläche dargestellt.

## 2.3 HMDoorSensor-Flow

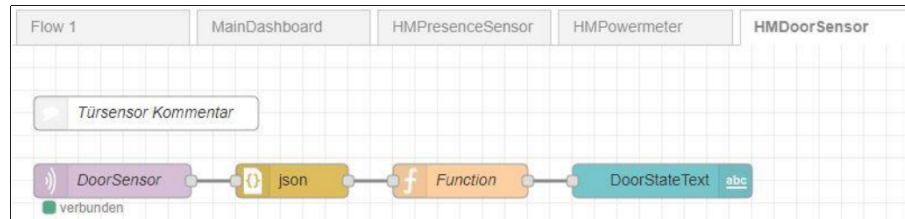


Abbildung 7: HomeMatic Türsensor Flow

Der HomeMatic Funk Tür-/Fensterkontakt liefert durch die Subscription „DoorSensor“ (violetterNode) einen Boolean-Wert, ob die Türe offen oder geschlossen ist. Der weitergeleitete Wert wird in der „Function“ Node durch den String „Geschlossen“ oder „Offen“ ersetzt und als Text in der Bedienoberfläche dargestellt.

## 2.4 HMPowermeter-Flow

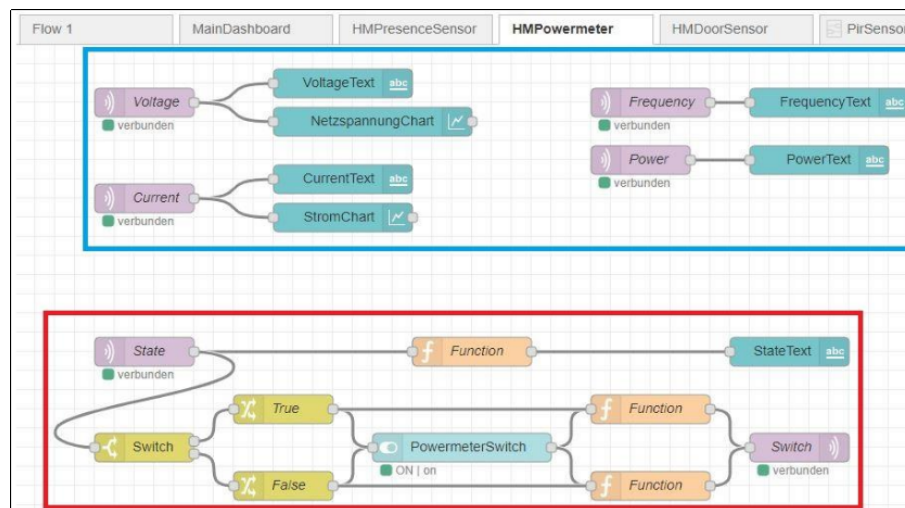


Abbildung 8: Sensorauswertung für den Homematic Funk-Schaltaktor

Der Homematic Funk-Schaltaktor bietet die Funktionalität Daten zu senden und zu empfangen. Die blaue Markierung beschreibt die Funktion Daten durch die Subscriptions (violetter Nodes) zu empfangen. Dabei werden die Werte Netzspannung, Strom, Leistung und Frequenz auf der Bedienoberfläche dargestellt. Die Netzspannung und der Strom haben zusätzlich eine grafische Auswertung erhalten, um dadurch den zeitlichen Verlauf der Werte besser verfolgen zu können. Für das Ein-/ Ausschalten des Funk-Schaltaktors wird ein Switch-Schalter innerhalb der Bedienoberfläche erstellt. Das Schalten erfolgt durch die Booleanwerte True/False und wird anschließend mit einem sogenannten Thema/Topic zu RedMatic weitergeleitet. Der „StateText“ Node zeigt in der Bedienoberfläche den aktuellen Verbindungsstatus des Homematic Moduls an.

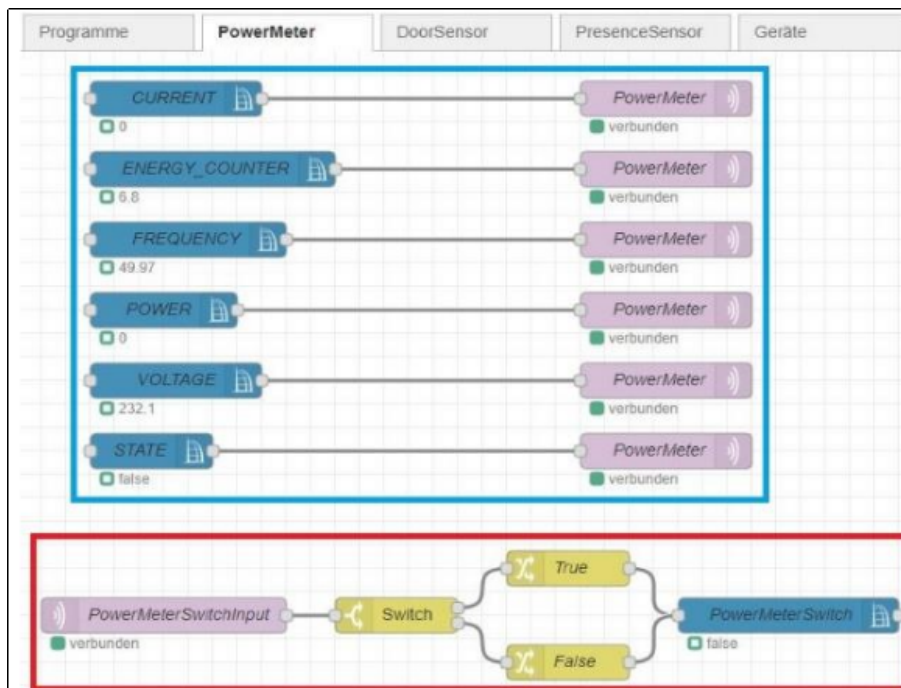


Abbildung 9: Weiterleitungen in RedMatic

Die Weiterleitung der Sensordaten an Node-RED erfolgt in RedMatic in Abschnitt ?? auf Seite ?. Wie bei der blauen Markierung erkennbar ist, werden in diesem Flow die Sensordaten des Homematic Moduls direkt an Node-RED weitergeleitet und dort empfangen. Die rote Markierung zeigt die Logik für das Umschalten des Funk-Schaltkontaktes. Zu Beginn wird eine Subscription auf Node-RED gemacht, welche die Daten des Switch-Schalters empfängt. Anschließend werden die empfangenen Boolean-Werte an das Modul gesendet und umgesetzt.

### 3 Subflow

Mit der Funktion Subflow bietet Node-RED die Möglichkeit, kleinere untergeordnete Flows zu erstellen die in einem Projekt auf verschiedenen Arbeitsblättern wiederverwendet werden können. Durch Subflows können komplexere Abläufe und Auswertungen in einem Node abstrahiert werden. Größere Projekte werden dadurch vereinfacht und klarer strukturiert.

#### 3.1 PirSensor-Subflow

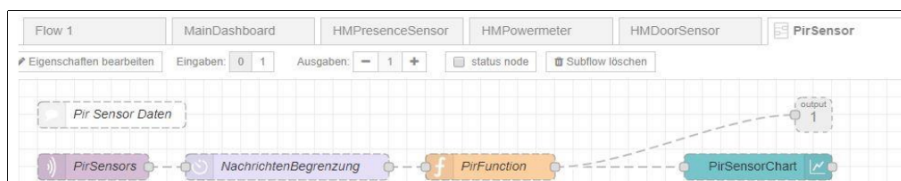


Abbildung 10: Subflow PIR-Sensor

Für die Anzeige der Sensordaten in einer Time-Chart, wird eine Subscription durchgeführt um die Daten zu empfangen. Durch den anschließenden „NachrichtenBegrenzung“ Node wird nur eine bestimmte Anzahl an Daten pro Zeiteinheit weitergeleitet, überschüssige Daten werden verworfen.

Die folgende „PirFunction“ filtert die Daten und leitet vereinfachte Werte weiter an das Time-Chart und an den Ausgang des Subflows.

```

1 var newMsg;
2
3 if(msg.payload === "10" || msg.payload === "00")
4 {
5     if(msg.payload === "10")
6     {
7         newMsg = {
8             topic: "Pir Sensor 1 & 2",
9             payload: 1 }
10    }
11    else
12    {
13        newMsg = {
14            topic: "Pir Sensor 1 & 2",
15            payload: 0 }
16    }
17 }
18 else if(msg.payload === "01" || msg.payload === "11")
19 {
20     if(msg.payload === "11")
21     {
22         newMsg = {
23             topic: "Pir Sensor 1 & 2",
24             payload: 1 }
25     }
26     else
27     {
28         newMsg = {
29             topic: "Pir Sensor 1 & 2",
30             payload: 0 }
31     }
32 }
33 return newMsg;

```

Das „PirFunction“ Node wertet die eintreffenden Daten bei der `if` Anweisung aus und erstellt ein abstrahiertes Messageobjekt für die Weiterverarbeitung. Die `If-Else` -Funktion in den Zeilen 3-17 und 18-32 wird unterschieden ob ein PIR-Sensor ausgelöst hat oder nicht, demnach wird ein neues Objekt mit dem Namen `newMsg` erstellt. Mit `return newMsg` werden die Daten weiter an den Ausgang und der Time-Chart geschickt.



## 3.2 UltraSchallSensor -Subflow

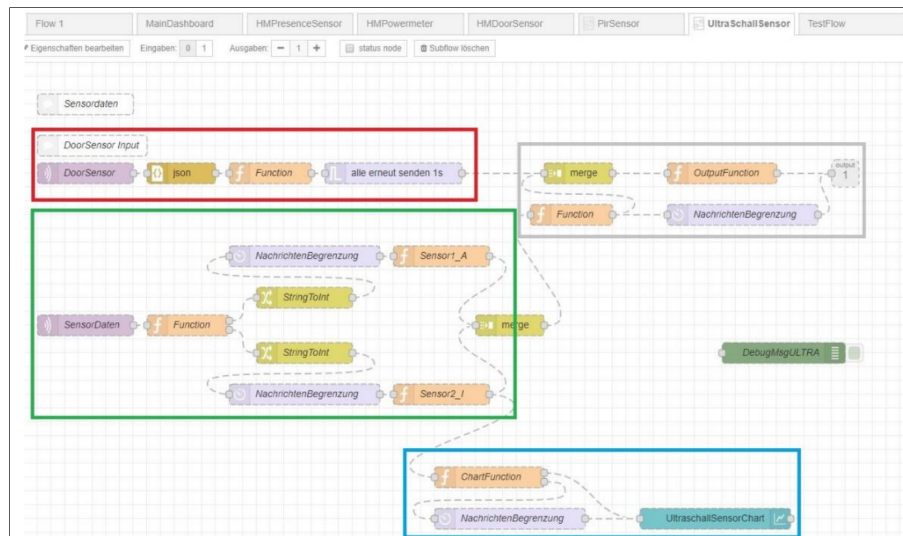


Abbildung 11: Ultraschallsensor Subflow

Das Arbeitsblatt für den „UltraSchallSensor“ wurde in vier logische Untergruppen eingeteilt (siehe farbige Markierungen in Abb. 11).

Innerhalb der grünen Markierung werden die Ultraschallsensordaten durch die Subscription empfangen. Durch das anschließende „Function“ Node wird der Datenstrom aufgeteilt und die Sensoren einzeln ausgewertet. Durch die `Sensor2_I` und `Sensor1_A` Funktion werden die einzelnen Datenströme ausgewertet. Da der eine Ultraschallsensor eine zu große Schwankung aufwies ist dieser nicht weiter mit in die Auswertung der Logik eingegangen, dadurch wurde in der `Sensor1_A` Funktion eine Weiterleitung mit `return 0` definiert.

Die türkise Markierung, unten rechts, definiert die Auswertungschart.

Die Graue Markierung verbindet die Ausgabedaten von dem Ultraschallsensor und Home-Matic Funk Tür-/Fensterkontakt. Durch das „merge“ Node werden die beiden Datenströme zusammengeführt und in der „OutputFunction“ verarbeitet. Nach der Verarbeitung werden die Ergebnisse an den Output geleitet.

```

1 // 'Innerer' Ultraschall-Sensor (Sensor Richtung Raum)
2 // Info: Neue Message wird weitergeleitet. Auswertung mit Bedingungsprüfung
3
4 const Distanz = 115; //eingestellte Entfernung (Programmierung in ArduinoStudio)
5 const Toleranz = 10; //Schwankungs-Toleranz
6
7 var newMsg; //Neue Message für neue Werte erstellen
8
9 if (msg.payload < (Distanz - Toleranz))//ausgelöst
10 {
11     newMsg = {
12         topic: "UltraschallSensor_Innen",
13         payload: true,
14         val: msg.payload } //msg.payload -> aktuellen Wert mitliefern
15 }
16 else
17 {
18     newMsg = {
19         topic: "UltraschallSensor_Innen",
20         payload: false,

```

```

21     val: msg.payload } //msg.payload -> aktuellen Wert mitliefern
22 }
23 return newMsg;

```

Die „Sensor2.I“ Function verarbeitet von einem der beiden Ultraschallsensoren die Daten. Innerhalb der Zeilen 4 und 5 werden zuerst die Konstanten festgelegt, welche die Distanz und eine Schwankungstoleranz definieren.

In den Zeilen 9 bis 22 wird durch eine Kontrollstruktur geprüft, ob jemand den Ultraschallsensor passiert hat. Sobald eine Person vorbeigelaufen ist, wird ein neues Message-Objekt erstellt und mit den dem payload-Wert true gefüllt.

```

1 var newMsg;
2 if(Array.isArray(msg.payload))
3 {
4     if(msg.payload[0].topic === "UltraschallSensor_Output" /* && msg.payload[1].
5     topic === "DoorSensor" */)
6     {
7         if(msg.payload[0].state === true /* && msg.payload[1].state === true */)
8         {
9             newMsg = { topic: "Output",
10             val: 1,
11             state: true }
12
13             return newMsg;
14         } else {
15             newMsg = { topic: "Output",
16             val: 0,
17             state: false }
18
19             return 0;
20         }
21     } else if(msg.payload[1].topic === "UltraschallSensor_Output" /* && msg.
22     payload[1].topic === "DoorSensor" */)
23     {
24         if(msg.payload[0].state === true /* && msg.payload[1].state === true */)
25         {
26             newMsg = { topic: "Output",
27             val: 1,
28             state: true }
29
30             return newMsg;
31         } else {
32             newMsg = { topic: "Output",
33             val: 0,
34             state: false }
35
36             return 0;
37         }
38     } else {
39         newMsg = { topic: "Output", val: 0, state: false }
40
41         return 0;
42     }
43 }

```

Die „OutputFunction“ beschreibt die Logikauswertung der Daten von dem HomeMatic Funk Tür-/ Fensterkontakt und des Ultraschallsensors. Durch die Überprüfung mit der Kontrollstruktur wird definiert, wenn die Türe geöffnet ist und jemand den Ultraschallsensor passiert.

## **4 Vergleich zu anderen IoT Tools**

## **5 Node-RED Fazit**

Mit dem Werkzeugbaukasten Node-RED ist es möglich innerhalb von kürzester Zeit optisch ansprechende Dashboards zu erstellen. Die große Auswahl an Bausteinen bietet eine gute Voraussetzung für das Entwerfen übergreifender Anwendungen. Das Spektrum reicht von der Ansteuerung von Hardware bis zu High-Level-APIs wie Social Media Plattformen. Node-RED kann durch eigene Scripte erweitert werden, diese werden in der Programmiersprache JavaScript geschrieben. Für die Projektarbeit hat sich Node-RED sehr gut bewährt, weil keine expliziten Programmierkenntnisse vorausgesetzt sind. Weiterhin stellt der Plattform-Anbieter ausführliche Anleitungen und Tutorials bereit.