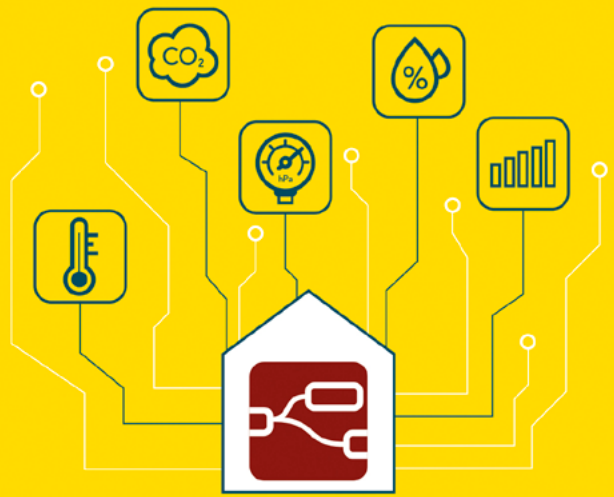


Udo Brandes

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



Messanzeigen

Temperatur BMP280



Luftdruck



Linendiagramme

Temperatur



Schalter

LED Pin 11



Schalter



Node-RED

Das umfassende Handbuch

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

4., aktualisierte Auflage



Beispiel-Flows und Projektdateien zum Download



Rheinwerk
Computing

Liebe Leserin, lieber Leser,

wenn Sie ganz neu in die Programmierung einsteigen, werden Sie sich zunächst auf das Wesentliche konzentrieren wollen und sind bestimmt froh, wenn Sie möglichst wenig Arbeit mit Syntax-Eigenheiten und kryptischen Codezeilen haben. Und falls Sie bereits über Erfahrung in der professionellen Softwareentwicklung verfügen, wissen Sie, wie wichtig eine gute grafische Planung beispielsweise mit UML ist, um im Verlauf eines anspruchsvollen Projekts nicht den Überblick zu verlieren. Sonst wird schnell Spaghetti-code produziert.

Da liegt es natürlich nahe, ganz auf ein visuelles Programmierkonzept zu setzen, das für klare Strukturen sorgt. Indem Sie Nodes intuitiv miteinander verbinden, sorgen Sie für übersichtliche Befehlsstränge, die Programmierlogiken und Schaltungen einfach abbilden. Und nicht nur das: Mit Node-RED haben Sie zudem ein Werkzeug, das alle wichtigen Module bereits mitbringt, die Sie für Maker-Projekte brauchen oder die Schaltungen in der Heimautomation einfacher machen.

Udo Brandes stellt Ihnen dazu die Idee hinter Flow-basierter Programmierung vor und gibt Ihnen einen Überblick über die zentralen Nodes. Dabei geht er aufgaben- und themenorientiert vor. Sie lernen strukturiert, wie Node-RED installiert und abgesichert wird und machen sich dann schrittweise mit den Nodes für unterschiedliche Aufgaben und Anforderungen vertraut. So dauert es nicht lange, bis Node-RED Mails mit Ihren Sensordaten versendet oder Sie von unterwegs auf Ihre Schaltungen zugreifen.

Abschließend noch ein Wort in eigener Sache: Dieses Werk wurde mit großer Sorgfalt geschrieben, geprüft und produziert. Aber wo Menschen arbeiten, passieren Fehler. Schreiben Sie mir, wenn Ihnen etwas auffällt. Ihre Kritik und konstruktiven Anregungen sind jederzeit willkommen.

Ihr Christoph Meister

Lektorat Rheinwerk Computing

christoph.meister@rheinwerk-verlag.de

www.rheinwerk-verlag.de

Rheinwerk Verlag • Rheinwerkallee 4 • 53227 Bonn

Auf einen Blick

1	Node-RED – das Setup: So starten Sie	23
2	Das zentrale Tool: der Node-RED-Editor	61
3	Das Fundament: die Basics von Node-RED	117
4	Das Node-RED-Dashboard	145
5	Funktionen programmieren	185
6	Daten über Netzwerkprotokolle abrufen	237
7	Daten mit Node-RED teilen	283
8	Daten speichern und archivieren	339
9	Node-RED-Hacks	387
10	Apps und externe Anbindung	427
11	Dashboards für Fortgeschrittene	459
12	Node-RED in andere Dienste integrieren	489
13	Eigene Nodes erstellen	503
14	Mit Node-RED Mikrocontroller programmieren	519

Impressum

Dieses E-Book ist ein Verlagsprodukt, an dem viele mitgewirkt haben, insbesondere:

Autor Udo Brandes

Lektorat Christoph Meister, Anne Scheibe

Typografie & Layout Vera Brauner

Satz E-Book SatzPro, Krefeld

Herstellung E-Book Norbert Englert

Covergestaltung Bastian Illerhaus

Coverbild iStock: 546022860 © golubovy

Bibliografische Information der Deutschen Nationalbibliothek:

Die Deutsche Nationalbibliothek verzeichnet diese Publikation in der Deutschen Nationalbibliografie; detaillierte bibliografische Daten sind im Internet über <http://dnb.dnb.de> abrufbar.

ISBN 978-3-367-10931-9 (PDF)

4., aktualisierte Auflage 2026

© Rheinwerk Verlag, Bonn 2026

Rheinwerk Verlag GmbH • Rheinwerkallee 4 • 53227 Bonn

service@rheinwerk-verlag.de

Informationen zu unserem Verlag und Kontaktmöglichkeiten finden Sie auf unserer Verlagswebsite www.rheinwerk-verlag.de. Dort können Sie sich auch umfassend über unser aktuelles Programm informieren und unsere Bücher und E-Books bestellen.

Inhalt

Materialien zum Buch	13
Einleitung	15
1 Node-RED – das Setup: So starten Sie	23
1.1 Node-RED – das zentrale Element	23
1.2 Node-RED aufsetzen	25
1.2.1 Node-RED auf einem Android-Gerät	25
1.2.2 Node-RED auf einem Windows-PC	25
1.2.3 Node-RED auf einem Raspberry Pi	29
1.2.4 Node-RED auf einem Linux-System	32
1.3 Node-RED überall: Docker-Container starten	34
1.3.1 Docker auf einem Ubuntu-System	34
1.3.2 Docker auf dem Raspberry Pi	35
1.3.3 Node-RED in Docker ausführen	36
1.4 Die Ausgaben von Node-RED beim Start	38
1.5 Node-RED administrieren	39
1.5.1 Dateien und Ordner	39
1.5.2 Dateien unter Docker	41
1.5.3 Einstellungen von Node-RED ändern	41
1.6 Node-RED absichern	43
1.6.1 Passwortschutz für den Node-RED-Editor	44
1.6.2 HTTPS aktivieren	45
1.6.3 Sicherheit für den http-in-Node	50
1.6.4 Die Konfigurationsdatei absichern	50
1.7 Node-RED Projekte	51
1.8 Fazit	60
2 Das zentrale Tool: der Node-RED-Editor	61
2.1 Den Node-RED-Editor in einem Browser öffnen	61
2.2 Die Kopfleiste	64
2.2.1 Die Schaltfläche deploy	64
2.2.2 Die Benutzerauthentifikation	66
2.2.3 Das Hauptmenü	66

2.3	Die linke Seitenleiste	74
2.3.1	Der Flow-Explorer	75
2.3.2	Die Node-Palette	75
2.4	Der Arbeitsbereich	83
2.4.1	Die Verwaltungsleiste für die Flows	84
2.4.2	Der Flow-Design-Bereich	85
2.4.3	Die Fußleiste des Arbeitsbereichs	91
2.4.4	Das Kontextmenü	92
2.5	Die rechte Seitenleiste	93
2.5.1	Das Register info	93
2.5.2	Das Register Hilfe	95
2.5.3	Das Register Debugging	95
2.5.4	Das Register Konfiguration	96
2.5.5	Das Register Kontext	97
2.5.6	Das Register Dashboard	97
2.6	Der erste Flow	97
2.7	Flows mit AI entwickeln	99
2.8	Gute Programmierung	101
2.8.1	Die Flow-Struktur	101
2.8.2	Das Message-Design	108
2.8.3	Die Dokumentation	108
2.9	Flows mit dem Flow-Debugger debuggen	109
2.9.1	Schrittweises Durchlaufen eines Flows	111
2.9.2	Messages löschen	111
2.9.3	Breakpoints nutzen	112
2.10	Probleme mit nrlinter aufspüren	112
2.10.1	Installation	113
2.10.2	Mit nrlinter arbeiten	113
2.11	Fazit	115
3	Das Fundament: die Basics von Node-RED	117
3.1	Das Message-Konzept von Node-RED	117
3.1.1	JSON – das Datenformat für den Datenaustausch	117
3.1.2	Messages in Node-RED	121
3.2	Die Geschwister inject-Node und debug-Node	125
3.2.1	Der inject-Node	125
3.2.2	Der debug-Node	127

3.3	Messages manipulieren: Der change-Node und seine Begleiter	129
3.3.1	Der switch-Node	130
3.3.2	Der change-Node	131
3.4	Der delay-Node	133
3.5	Dateiformate konvertieren	133
3.6	Auf Prozessereignisse reagieren	135
3.6.1	Der status-Node	135
3.6.2	Der complete-Node	136
3.6.3	Der catch-Node	137
3.7	Sequenzen (Folgen)	139
3.8	Fazit	144
4	Das Node-RED-Dashboard	145
4.1	Installation	145
4.2	Browseraufruf und Einstellungen	147
4.3	Der Schnelleinstieg: So erstellen Sie Ihre erste Dashboard-Ausgabe	147
4.4	Das Dashboard-Design bestimmen	150
4.4.1	Icons	151
4.4.2	Die rechte Seitenleiste	152
4.4.3	Custom CSS	156
4.4.4	Die Konfiguration von Tabs und Gruppen	156
4.5	Die Dashboard-Widgets in Aktion	158
4.5.1	Der button-Node	159
4.5.2	Der dropdown-Node	160
4.5.3	Der switch-Node	162
4.5.4	Der slider-Node	163
4.5.5	Der numeric-Node	164
4.5.6	Der text-input-Node	164
4.5.7	Der form-Node	165
4.5.8	Die beiden function-Nodes und der template-Node	166
4.6	Charts und Messanzeigen mit dem Raspberry Pi	168
4.6.1	Sensoren schalten und das benötigte Paket installieren	169
4.6.2	Die LEDs schalten	171
4.6.3	Wetterdaten erheben und ausgeben	171
4.7	Das Diagramm-Kaleidoskop	177

4.8	Das Dashboard 2.0	178
4.8.1	Installation	179
4.8.2	Der Schnelleinstieg	179
4.9	Fazit	183
5	Funktionen programmieren	185
5.1	Einsatz und Funktionsweise des function-Nodes	185
5.1.1	Eingangsnachrichten lesen	186
5.1.2	Nachrichten erstellen	188
5.1.3	Code zur Ausführung bei setup und close	191
5.1.4	Ereignisse loggen	192
5.1.5	Das Erscheinungsbild ändern	193
5.2	Programmierung mit JavaScript	193
5.2.1	Codeeditoren	193
5.2.2	Zeichen, Kommentare und Begriffe	194
5.2.3	Daten und Datentypen	196
5.2.4	Variablen und Konstanten	197
5.2.5	Ausdrücke und Operatoren	202
5.2.6	Das Array-Objekt (Tabellen)	209
5.2.7	Das Date-Objekt	211
5.2.8	Funktionen	216
5.2.9	Kontrollstrukturen	218
5.3	Programmbeispiele für den function-Node	227
5.3.1	Eine Zeichenfolge aufteilen	228
5.3.2	Eine Nachricht verzögern	228
5.3.3	Auf eine Umgebungsvariable zugreifen	229
5.3.4	Zusätzliche Module nachladen	229
5.3.5	Mit Kontextvariablen arbeiten	230
5.3.6	Nachrichten zusammenführen	231
5.3.7	Mit Puffern arbeiten	232
5.4	Externe Module laden	233
5.4.1	Den Hostnamen ausgeben	234
5.4.2	Den RGB-Farbwert prüfen	234
5.5	Der Monaco-Codeeditor	236
5.6	Fazit	236

6	Daten über Netzwerkprotokolle abrufen	237
6.1	Daten von einem Server im Netz abrufen	237
6.1.1	Grundlagen von HTTP-Verbindungen	238
6.1.2	Die Nodes nutzen	242
6.2	MQTT: das IoT-Protokoll	263
6.2.1	Installation und ein Flow für den Einstieg	265
6.2.2	Einen eigenen MQTT-Broker aufsetzen	269
6.2.3	Node-RED an den Mosquitto-Broker anbinden	272
6.2.4	Der Shelly 1-Schalter	272
6.3	Arduino & Co. über USB anbinden	275
6.4	Fazit	282
7	Daten mit Node-RED teilen	283
7.1	E-Mails versenden	283
7.1.1	Das E-Mail-Konto konfigurieren	284
7.1.2	Der E-Mail-Versand	285
7.1.3	Der E-Mail-Empfang	290
7.2	Instant-Messaging und Bots mit Telegram	293
7.2.1	Instant Messaging	294
7.2.2	Bots	294
7.2.3	Telegram	295
7.3	Geräte mit Pushbullet vernetzen	310
7.3.1	Pushbullet einrichten	311
7.3.2	Erste Schritte mit der Pushbullet-API	313
7.3.3	Pushbullet für Node-RED	316
7.4	Sprachsteuerung mit Alexa	320
7.5	Google-Dienste integrieren	326
7.6	Mit künstlicher Intelligenz experimentieren	333
7.6.1	OpenAI-Account und API-Key erstellen	334
7.6.2	Ein Bild erstellen	334
7.6.3	Eine Textnachricht generieren	336
7.7	Fazit	338
8	Daten speichern und archivieren	339
8.1	Kontextvariablen	339
8.1.1	Kontextvariablen vom Typ node	340

8.1.2	Kontextvariablen vom Typ flow	341
8.1.3	Kontextvariablen vom Typ global	343
8.1.4	Kontextvariablen im Dateisystem speichern	344
8.2	Daten in Dateien speichern	345
8.2.1	Messdaten speichern und wieder auslesen	346
8.2.2	Dateimanager	352
8.3	Node-RED und InfluxDB	356
8.3.1	InfluxDB, eine Time Series Database	357
8.3.2	InfluxDB installieren	359
8.3.3	Die ersten Schritte mit InfluxDB	361
8.3.4	Mit Node-RED Daten in InfluxDB speichern	366
8.3.5	Mit Node-RED Daten aus der InfluxDB auslesen	369
8.3.6	Die InfluxDB sauber halten	373
8.4	Node-RED und SQLite	376
8.4.1	Aufbau einer SQLite-Datenbank	376
8.4.2	SQLite installieren	379
8.4.3	Mit Node-RED Daten in der SQLite-Datenbank speichern	382
8.4.4	Mit Node-RED Daten aus der SQLite-Datenbank löschen	384
8.4.5	Mit Node-RED Daten aus der SQLite-Datenbank auslesen	385
8.5	Fazit	385
9	Node-RED-Hacks	387
9.1	Python-Skripte einbinden	387
9.2	Timer	389
9.2.1	Ausgaben	391
9.2.2	Timersteuerung	392
9.2.3	Erweiterte Möglichkeiten	392
9.3	»Himmelserscheinungen« auswerten	393
9.4	Wetterdaten mit OpenWeatherMap	395
9.4.1	Das openweathermap-Konto	396
9.4.2	Eine Wetteransage	397
9.4.3	Ein Frostwächter	399
9.5	Zeitangaben formatieren	399
9.5.1	Der Node Date/Time Formatter	400
9.5.2	Der humanizer-Node	402
9.6	Mit Bilddateien arbeiten	402
9.7	Einen QR-Code generieren	404

9.8	Geräte mit Ping orten	406
9.8.1	Anwesenheitsmitteilung senden	406
9.8.2	Alarmanlage aktivieren	408
9.8.3	Erreichbarkeit eines Servers überprüfen	408
9.9	Auf eine FRITZ!Box zugreifen	410
9.9.1	Anwesenheitsbenachrichtigung	411
9.9.2	Benachrichtigung bei Anrufen	413
9.9.3	Gastzugang schalten	414
9.10	FTP – Daten zwischen Rechnern übertragen	416
9.10.1	Einen FTP-Server aufsetzen	417
9.10.2	Das Verzeichnis mit Node-RED lesen	419
9.10.3	Node-RED-Konfigurationsdateien sichern	421
9.11	Benachrichtigungen mit Ntfy versenden	423
9.11.1	Nutzerkonto erstellen	423
9.12	Fazit	426
10	Apps und externe Anbindung	427
10.1	Apps aus den App-Stores	427
10.2	Blynk 2.0	428
10.2.1	Mit Blynk auf den Raspberry Pi zugreifen	428
10.2.2	Blynk und Node-RED kommunizieren miteinander	438
10.3	Die Termux-App	444
10.3.1	Node-RED auf dem Android-Gerät	444
10.3.2	Die Termux:API	446
10.3.3	Die Termux:API-App mit Node-RED nutzen	449
10.4	Der »Überall-Zugriff« mit ngrok	451
10.5	Fazit	457
11	Dashboards für Fortgeschrittene	459
11.1	Dynamische Dashboard-Steuerung	459
11.2	Der template-Node (Widget)	461
11.2.1	Einfache (statische) HTML-Ausgaben mit dem ui-template-Node ...	462
11.2.2	Den Eingabeport nutzen	467
11.2.3	ui-template-Node – den Ausgabeport nutzen	475
11.2.4	Statusinformationen im Dashboard-Header ausgeben	478
11.3	Ein aufwendigeres Beispiel	479

11.4	Ein alternatives Dashboard mit uibuilder	482
11.5	Fazit	488
12	Node-RED in andere Dienste integrieren	489
12.1	ioBroker	489
12.1.1	Installation und Inbetriebnahme	490
12.1.2	ioBroker-Objekte: So schalten Sie eine LED	493
12.1.3	ioBroker und Node-RED	495
12.2	Node-RED versus externe Dienste	500
12.3	Fazit	502
13	Eigene Nodes erstellen	503
13.1	Anforderungen definieren	503
13.2	Arbeitsverzeichnis erstellen und ausgestalten	504
13.3	Die Datei package.json generieren	505
13.4	Die Datei <node>.js programmieren	506
13.4.1	Der Rahmen	506
13.4.2	Den Rahmen ausfüllen	507
13.5	Ein Icon erstellen	511
13.6	Die Datei basic-math.html generieren	511
13.7	Den Node basic-math in Node-RED testen	514
13.8	Ausblick	518
13.9	Fazit	518
14	Mit Node-RED Mikrocontroller programmieren ...	519
14.1	Der Soft- und Hardwarerahmen	520
14.2	Installation von Moddable	520
14.3	Den Mikrocontroller an Moddable anbinden	523
14.4	Einrichtung in Node-RED und Test	525
14.5	Fazit	528
	Index	529

Materialien zum Buch

Auf der Webseite zu diesem Buch stehen folgende Materialien für Sie zum Download bereit:

- **Projektbeispiele aus dem Buch**
- **Zusätzliche Visualisierungen und Schaltungsbeispiele**

Sie finden für jedes Kapitel einen Node-RED Flow-Tab. Diese Tabs enthalten die Beispiel-flows. Jedes Tab können Sie im JSON-Format exportieren und entsprechend in Ihre eigenen Projekte importieren.

Bitte beachten Sie mögliche Probleme, wenn Sie Tabs importieren möchten, die Nodes enthalten, die Sie nicht installiert haben. Fügen Sie sie gegebenenfalls über den Palettenmanager hinzu.

Gehen Sie auf www.rheinwerk-verlag.de/6137. Klicken Sie auf den Reiter **Materialien**. Sie sehen die herunterladbaren Dateien samt einer Kurzbeschreibung des Dateiinhalts. Klicken Sie auf den Button **Herunterladen**, um den Download zu starten. Je nach Größe der Datei (und Ihrer Internetverbindung) kann es einige Zeit dauern, bis der Download abgeschlossen ist.

Einleitung

Mit individuellen Lösungen aus dem Smart-Home-Bereich machen Sie aus einem Haus Ihr ganz persönliches Zuhause. Es ist dort nämlich so wie in nahezu allen Lebensbereichen: Industrielle Angebote entsprechen oft nicht den persönlichen Bedürfnissen, oder die angebotenen Produkte treffen nicht den Geschmack. Daher wünscht man sich eine Lösung, die man selbst umgesetzt hat. Und zudem kann man so den Dingen auf den Grund zu gehen und weiß ganz genau, mit was man es zu tun hat.

Der Wunsch, dies auch auf Steuerungsmechanismen des häuslichen Umfelds (*Hausautomation*) oder kleine Helfer (*Personal Assistants*) zu beziehen, ist da keine Ausnahme. Ohne vertiefte Programmierkenntnisse gestaltete sich dieser Zugang jedoch lange Zeit schwierig und war vornehmlich Insidern oder speziell ausgebildeten und technisch affinen Personen vorbehalten.

Das hat sich allerdings in den letzten Jahren geändert: Mit Hausautomatisierungslösungen wie etwa *Home Assistant*, *openHAB* und Einplatinenrechnern wie dem *Raspberry Pi* können nun auch engagierte Maker ihre Vorstellungen in Eigenregie verwirklichen. Während der *Raspberry Pi* die zuverlässige und sparsame Hardware darstellt, schafft die angebotene Software einen Rahmen, innerhalb dessen Sie Ihre individuellen Vorstellungen entwickeln können. Auch wenn dieser Rahmen zwischenzeitlich recht weit gespannt ist, vermag er doch nicht alle denkbaren Varianten abzudecken.

Hier kommt *Node-RED* ins Spiel. Mit *Node-RED* können Sie professionelle Setups erstellen, die mit einer Vielzahl von Diensten kommunizieren, sich aber gleichzeitig einfach und überschaubar programmieren lassen, ohne dass langjährige Erfahrung in der Softwareentwicklung notwendig ist.

Die Geschichte von *Node-RED* ist noch relativ jung. Es wurde ursprünglich Ende 2013 von Nick O’Leary und Dave Conway-Jones, beide Wissenschaftler der IBM Engineering Technologies Group, als Open-Source-Projekt entwickelt. Mit ihm sollen auf einfache Weise unterschiedliche Geräte, Webdienste und Programmierschnittstellen verbunden werden.

Der Kern von *Node-RED* ist in der Programmiersprache *JavaScript* geschrieben und baut auf der Laufzeitumgebung *Node.js* auf. Das Wort »Node« steht daher auch Pate für einen Namensbestandteil. Der Zusatz »RED« hat wohl keine tiefere Bedeutung. Er geht wahrscheinlich auf einen Vorschlag von Dave Conway-Jones zurück, wohl weil sich der gesamte Name wie »Code-Red« anhört, vormals eine weitverbreitete Gruppe von Computervürmern. Und es klingt schlicht cool.

Node-RED ist eine rasante Erfolgsgeschichte. Innerhalb kurzer Zeit hat es eine große Popularität erreicht und ist heute ein weitverbreitetes Programmier- und *Prototyping*-Werkzeug. Das liegt an der überaus einfachen Handhabung, die hin und wieder etwas despektierlich als »Klick-dich-glücklich-Programmierung« bezeichnet wird. Dies ist inso-

fern zutreffend, als dass Node-RED auch Einsteigern recht schnell Erfolgserlebnisse verschafft. Node-RED bietet Ihnen aber auch ein nahezu unerschöpfliches Betätigungsspektrum, wenn Sie tiefer in die Programmierung einsteigen und professionell arbeiten möchten.

Um Node-RED ist eine lebhafte Benutzer- und Entwicklergemeinschaft entstanden. Eine Kerngruppe kümmert sich um den Node-RED-Code selbst; eine Vielzahl anderer Entwickler und Entwicklerinnen steuert *Nodes* oder *Flows* zur Node-Bibliothek bei. Das offizielle Forum (<https://discourse.nodered.org>) ist zwar englischsprachig, aber sehr rege. Mit etwas Glück antwortet sogar Nick O’Leary persönlich auf ein Problem. Wenn Sie Informationen auf Deutsch suchen, bietet <https://forum.heimnetz.de/forums/node-red.26/> eine gute Anlaufstelle.

Der Ansatz des Buchs

Dieses Buch ist keine schrittweise Einführung in ein spezielles und abgegrenztes Anwendungsgebiet von Node-RED wie z. B. die Robotik, die Hausautomation oder das *Internet of Things* (IoT). Sie werden also kein durchgehendes Projekt finden und auch keine breite Autobahn, auf der Sie durch alle Abschnitte dahinrauschen könnten.

Stattdessen möchte ich Ihnen möglichst umfassend verschiedene Aspekte näherbringen. Mit diesem Konzept möchte ich Sie zum Experimentieren anregen und Ihnen den Einstieg auch in komplexere Vorhaben ermöglichen, die Sie dann für Ihre individuellen Projekte nutzen können. Es ist deshalb nicht nötig (und wäre sogar eher ermüdend), das Buch sequenziell abzuarbeiten. Picken Sie sich einzelne Themen heraus und verknüpfen Sie sie mit anderen Blöcken. Nehmen Sie nur das mit, was Sie aktuell brauchen, und kehren Sie vielleicht später zurück, um dann in die Tiefe zu gehen.

Neben den Erklärungen spielen Abbildungen eine tragende Rolle. Getreu dem Motto, dass Bilder mehr als tausend Worte sagen, lässt sich mit ihnen trotz des Platzes, den Bilder nun einmal einnehmen, die zu vermittelnde Botschaft einfach besser darstellen.

Für wen ist dieses Buch gedacht?

Das Buch richtet sich an alle,

- die einen Einstieg in die Hausautomation, das IoT und die Anwendung persönlicher Assistenten suchen,
- die neugierig sind und gerne einmal »über den Tellerrand schauen«,
- die mit einfachen Mitteln schnell loslegen möchten,
- die nicht nur einen Einstieg in ein ausgewähltes Anwendungsgebiet suchen, sondern das gesamte Leistungsspektrum vorgestellt bekommen möchten,

- die es begrüßen, ausgewählte Techniken der Softwareentwicklung und Programmierung vorgestellt zu bekommen, und
- die gern eigene Projekte unabhängig von industriellen Produkten für den Massenmarkt entwickeln möchten.

Der Inhalt des Buchs

Das Buch soll eine umfassende Unterstützung bei der Programmierung mit Node-RED bieten. Das Universum von Node-RED ist zwischenzeitlich aber so groß geworden, dass nicht jeder Winkel und jede Option beleuchtet werden kann. Deshalb sprechen die Kapitel nicht alle Möglichkeiten an, sondern versuchen, eine breite und tragfähige Basis für eigenes Erkunden zu sein.

■ **Kapitel 1: Node-RED – das Setup: So starten Sie**

Zum Einstieg geht es um unterschiedliche Arten der Installation von Node-RED und um Fragen der Administration.

■ **Kapitel 2: Das zentrale Tool: der Node-RED-Editor**

Der Schwerpunkt dieses Kapitels ist der Einsatz des Node-RED-Editors. Im Editor werden Sie bei der Arbeit mit Node-RED die meiste Zeit verbringen, daher sollten Sie zum Einstieg auf jeden Fall einen Blick hineinwerfen.

■ **Kapitel 3: Das Fundament: die Basics von Node-RED**

Dieses Kapitel stellt die Core-Nodes vor und zeigt, wie Sie Nachrichten durch Flows schleusen.

■ **Kapitel 4: Das Node-RED-Dashboard**

Dieser Teil geht auf die wesentlichen Dashboard-Nodes ein. Sie lernen, wie Sie Informationen visualisieren und darstellen.

■ **Kapitel 5: Funktionen programmieren**

Sie können Node-RED sehr flexibel nutzen, wenn Sie eigene Funktionen programmieren. Dieses Kapitel zeigt, wie das geht.

■ **Kapitel 6: Daten über Netzwerkprotokolle abrufen**

Dieses Kapitel befasst sich mit der Anbindung von Node-RED an das Netz und unterschiedliche Netzwerkprotokolle, damit Sie Daten abrufen und nutzen können.

■ **Kapitel 7: Daten mit Node-RED teilen**

In diesem Kapitel geht es um das Zusammenspiel von Node-RED mit verschiedenen *Social-Media-Kanälen*. Wenn Sie Temperaturwerte Ihres Sensors mit der ganzen Welt teilen wollen, sind Sie hier richtig.

■ **Kapitel 8: Daten speichern und archivieren**

Dieser Teil widmet sich den Möglichkeiten und Erfordernissen, Daten in und mit Node-RED zu speichern.

■ **Kapitel 9: Node-RED-Hacks**

Dieses Kapitel geht auf Nodes ein, die nicht im Standardpaket enthalten sind. So können Sie Node-RED erweitern und anpassen.

■ **Kapitel 10: Apps und externe Anbindung**

Sie wollen Node-RED bestimmt nicht nur auf Ihrem Bürorechner nutzen. Wie Sie Node-RED auf und mit mobilen Geräten einsetzen, skizziere ich in diesem Kapitel.

■ **Kapitel 11: Dashboard für Fortgeschrittene**

Dashboards bieten viele individuelle Möglichkeiten. Ich möchte Ihnen einige Gestaltungstipps geben, die über die Grundfunktionen hinausgehen.

■ **Kapitel 12: Node-RED in andere Dienste integrieren**

Sie möchten Node-RED in andere Dienste zur Hausautomation integrieren? Dieses Kapitel zeigt Ihnen ein Beispiel mit ioBroker und liefert Ihnen Entscheidungshilfen bei der Auswahl.

■ **Kapitel 13: Eigene Nodes erstellen**

Wenn Ihnen die Nodes aus dem Standardpaket und die Zusatzangebote aus dem Internet nicht ausreichen, müssen Sie selbst die Ärmel hochkrempeln. Kurz vor Abschluss des Buchs gibt dieses Kapitel Ihnen einen Einstieg in die Eigenentwicklung von Nodes.

■ **Kapitel 14: Mit Node-RED Mikrocontroller programmieren**

Lernen Sie einen neuen Ansatz kennen, mit Node-RED zu arbeiten.

Hinweis für Schnelleinsteiger

Sie möchten schnell ein erstes Erfolgserlebnis haben? Kein Problem. Verschaffen Sie sich Zugang zu Node-RED – dazu gibt es unterschiedliche Möglichkeiten, die ich in [Abschnitt 1.2](#), »Node-RED aufsetzen«, beschreibe. Folgen Sie dann der Beschreibung in [Abschnitt 2.6](#), »Der erste Flow«, zur Einrichtung eines Hallo-Welt-Programms.

Besondere Formate

Um die Ausführungen in diesem Buch möglichst eindeutig und verständlich darzustellen, werden verschiedene Formate verwendet:

- `/Pfad/Zu/Einer/Datei` kennzeichnet einen Dateipfad.
- Neu verwendete Begriffe, z. B. *Entwicklungsumgebung*, werden kursiv dargestellt.
- Codeelemente, z. B. `delay()`, haben ebenfalls eine eigene Schriftart.
- **Bildschirmelemente** sind alle Objekte, die Sie auf Ihrem Bildschirm sehen sollten. Sie werden fett dargestellt.

Verwendete Software und Softwareversionen

Software und Bibliotheken unterliegen einem mitunter rasanten Wandel. Nicht immer sind sie abwärtskompatibel. [Tabelle 1](#) listet die Versionen der Software auf, mit denen die Beispiele für dieses Buch entstanden sind.

Programm/Bibliothek	Version
Windows	11-Pro (Build26200)
Linux (Ubuntu)	24.04 LTS
Raspberry Pi OS	Bookworm
npm	10.8.02
NodeJS	20.20.2 und 14.17.2
Node-RED	4.1.8
<i>node-red-dashboard</i>	3.6.5
<i>@flowfuse/node-red-dashboard</i>	1.21.0
<i>node-red-contrib-alexa-home-skill</i>	0.1.19
<i>node-red-contrib-bigtimer</i>	2.7.5
<i>node-red-contrib-blynk-ws</i>	1.0.5
<i>node-red-contrib-bme280</i>	1.0.0
<i>node-red-contrib-fritz</i>	1.4.5
<i>node-red-contrib-fs</i>	1.4.1
<i>node-red-contrib-ftp</i>	0.0.6
<i>node-red-contrib-image-tools</i>	1.4.0
<i>node-red-contrib-influxdb</i>	0.6.0
<i>node-red-contrib-moment</i>	4.0.0
<i>node-red-contrib-play-audio</i>	2.5.0
<i>node-red-contrib-qrcode-generator</i>	0.4.0
<i>node-red-contrib-telegrambot</i>	8.9.5
<i>node-red-node-arduino</i>	0.3.1

Tabelle 1: Versionen verwendeter Software und Bibliotheken

Programm/Bibliothek	Version
<i>node-red-node-base64</i>	0.3.0
<i>node-red-node-email</i>	1.12.1
<i>node-red-node-openweathermap</i>	0.4.0
<i>node-red-node-pi-gpio</i>	1.2.3
<i>node-red-node-ping</i>	0.3.1
<i>node-red-node-pushbullet</i>	0.0.17
<i>node-red-node-random</i>	0.4.1
<i>node-red-node-rbe</i>	0.5.0
<i>node-red-node-sentiment</i>	0.1.6
<i>node-red-node-serialport</i>	0.14.0
<i>node-red-node-smooth</i>	0.1.2
<i>node-red-node-sqlite</i>	0.6.0
<i>node-red-node-suncalc</i>	1.0.1
<i>node-red-node-tail</i>	0.3.1
<i>mosquitto</i>	2.0.11

Tabelle 1: Versionen verwendeter Software und Bibliotheken (Forts.)

Release 5 von Node-RED sollte im Januar 2026 produktiv sein; leider waren zum Redaktionsschluss Ende März 2026 nur Beta-Versionen verfügbar. Gravierende inhaltliche Änderungen sind nicht zu vermelden. Eine Ausnahme ist vielleicht ein geändertes Dashboard-Layout des Node-RED Editors. Diese greife ich in [Kapitel 2](#) mit dem zu diesem Zeitpunkt erreichten Entwicklungsstand auf. Im Übrigen basieren alle Screenshots auf dem offiziellen Release 4.

Noch eine dringende Bitte vorab

Node-RED selbst, aber auch die im Buch angesprochenen Komponenten, entwickeln sich rasant weiter. Dinge, die gestern noch nicht möglich waren, bereiten heute vielleicht keine Schwierigkeiten mehr oder funktionieren morgen wieder völlig anders.

Neben den technischen Veränderungen machen in letzter Zeit auch zunehmend Lizenzänderungen und Preisanpassungen das Arbeiten schwer. Viele Anbieter, die

bisher kostenfreie Leistungen angeboten haben, mussten diese in den letzten Monaten drastisch einschränken (z. B. Blynk) oder sogar gänzlich einstellen (wie beispielsweise das Hosting-Angebot zu Node-RED in der IBM-Cloud). Ich habe versucht, auf solche Dienste möglichst zu verzichten, aber nicht immer ist das möglich – viele Angebote sind einfach zu nützlich. Achten Sie aber darauf, ob die Dienste noch kostenfrei genutzt werden können und ob sich die Lizenzbedingungen verändert haben. Dort ist momentan noch mehr im Fluss als auf der technischen Seite.

Ein weiteres Symptom dieser schnellen Entwicklungsschritte sind die nur unzureichend übersetzten Menüs und Dialoge innerhalb der Software. Sie werden auf vielen Screenshots eine wilde Mischung aus Englisch und Deutsch sehen. Auch sind viele Begriffe nur sehr unzureichend und verwirrend übersetzt. Die Übersetzungen haben sich zudem mehrfach in den letzten Versionen geändert (bereits eingedeutschte Menüs erschienen plötzlich wieder auf Englisch), und wahrscheinlich wird es auch in zukünftigen Versionen noch Abweichungen geben. In den allermeisten Fällen lassen sich die Begriffe jedoch problemlos zuordnen.

Sollten Sie dennoch auf Probleme stoßen, ärgern Sie sich bitte nicht, sondern geben Sie entsprechende Hinweise an den Verlag.

Vielen herzlichen Dank

Udo Brandes

Kapitel 1

Node-RED – das Setup: So starten Sie

Erkunden Sie Node-RED und betreten Sie neues Terrain. Hier geht es um Grundlagen, um die Installationsoptionen und um Administrationsangelegenheiten. Es gibt viel zu entdecken!

In diesem Kapitel möchte ich Ihnen zeigen, was Node-RED ausmacht und welche Möglichkeiten bestehen, das Tool in Ihre bestehende Soft- und Hardwareumgebung einzuliefern und an Ihre Bedürfnisse anzupassen. Dies umfasst auch einen Blick auf die Administration des Systems. Dabei stelle ich Ihnen auch vor, welche Umgebung ich für die Beispiele aus diesem Buch nutze.

1.1 Node-RED – das zentrale Element

Node-RED ist ein Programmierwerkzeug, das auf denkbar einfache Weise Geräte, APIs (engl. *Application Programming Interface*, also Schnittstellen zu anderen Programmen) und Onlinedienste verknüpft. Als zentrales verbindendes Element gestaltet und steuert Node-RED das Zusammenspiel der einzelnen Komponenten (siehe [Abbildung 1.1](#)).

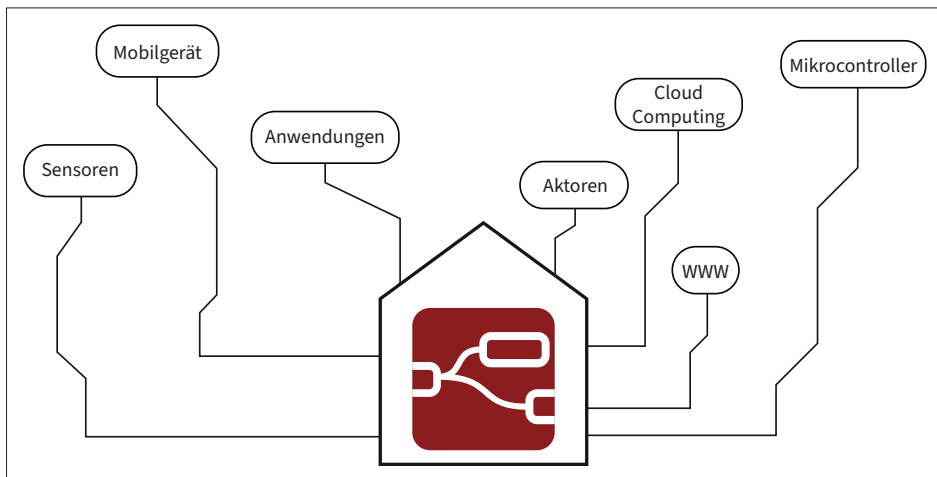


Abbildung 1.1: Node-RED als zentrales Element

Das Besondere dabei ist: Node-RED ist ein grafisches Entwicklungswerkzeug. Der Grundgedanke von Node-RED ist, dass vordefinierte Codeblöcke für die Ausführung von Aufga-

ben genutzt werden. Diese Codeblöcke heißen »Nodes« (dt. *Knoten*). Die jeweiligen Nodes lassen sich einfach miteinander vernetzen, sodass eine Linie von Eingabe-Nodes (hier startet der Datenverarbeitungsprozess), Verarbeitungs-Nodes (sie modellieren die Daten) und Ausgabe-Nodes (sie präsentieren das Ergebnis) entsteht. Ein solches Gebilde trägt die Bezeichnung *Flow* (dt. *(Daten-)Fluss*).

Auf diese Weise können Sie einzelne Nodes miteinander »verkabeln« und so Programmierlogik visuell darstellen. Beispielsweise kann ein Eingabe-Node den Wert eines Sensors aufnehmen, der anschließend an andere Nodes weitergegeben wird. Ein Verarbeitungs-Node prüft dann, ob eine vorher definierte Grenze überschritten wird. Falls dies zutrifft, wird ein Ausgabe-Node aktiviert, der eine Mail versendet oder eine LED leuchten lässt.

Flow-basierte Programmierung geht auf eine Idee von J. Paul Morrison aus den frühen 1970er-Jahren zurück. Sie betrachtet eine ganze Anwendung – und zwar nicht nur als einen einzelnen, sequenziellen Prozess, der zu einem bestimmten Zeitpunkt beginnt und dann jeweils eine Sache ausführt, bis sie abgeschlossen ist. Stattdessen geht es um ein Netzwerk von asynchronen Prozessen, die über Ströme strukturierter Datenblöcke (*Information Packets*, dt. *Informationspakete* (IPs)) kommunizieren.

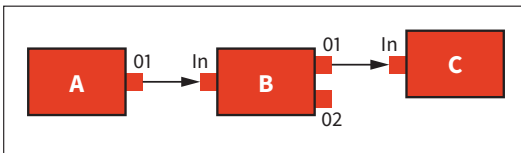


Abbildung 1.2: Flow-basierte Programmierung im Schema

Nehmen Sie für eine grobe Darstellung der Abläufe das Sensorbeispiel. A, B und C sind Prozesse, die Codekomponenten ausführen. 01, 02 und In sind Ports, die über Verbindungen die jeweiligen Prozesse verknüpfen. Komponente A erhebt die Sensordaten, bereitet sie auf und versendet das Ergebnis als Informationspaket über Port 01. Komponente B empfängt die Nachricht am Eingangsport (jede Komponente kann nur einen Eingangsport haben!) und führt die Grenzwertbetrachtung durch. Ist die Grenze überschritten, bedient die Komponente Port 01, andernfalls Port 02. Komponente C hängt am Ausgangsport 01 der Komponente B (Grenzwert überschritten) und tritt beim Ansteuern des Eingangsports in Aktion. Dies ist im Beispiel der Versand einer E-Mail oder das Schalten einer LED.

Nach dem Schaubild aus [Abbildung 1.2](#) läuft der Prozess rein sequenziell ab. In der Praxis jedoch sorgt das Flow-Management dafür, dass die Teilschritte asynchron abgearbeitet werden. Das System erkennt also, dass eine Komponente Arbeiten erledigen muss. Es weist dieser Komponente Ressourcen zu, wartet das Arbeitsergebnis ab (um es gegebenenfalls weiterzuleiten) und entzieht diese Ressourcen dann wieder. Darum müssen Sie sich also nicht kümmern, wenn Sie mit Node-RED arbeiten: Sie erstellen einfach die Flows, Node-RED kümmert sich um die Ausführung.

1.2 Node-RED aufsetzen

Node-RED ist ein *Dienstprogramm* (engl. *Service*), das sich auf verschiedensten Systemen und Systemumgebungen einsetzen lässt – in der IT spricht man vom *Hosten*. Als Basis stehen Ihnen verschiedene Optionen zur Verfügung:

- lokal auf einem PC unter Linux, Windows oder macOS,
- auf verschiedenen Geräten, beispielsweise auf einem Raspberry Pi, dem BeagleBone Black oder anderen Android-Devices,
- in der Cloud, indem Sie ein Angebot wie Amazon Web Services oder Microsoft Azure nutzen. Allerdings richten sich die Angebote an professionelle Anwender und sind auch nicht (mehr) kostenfrei.

Jede Option hat selbstverständlich objektiv und auch subjektiv ihre Vorteile, aber ebenso bestimmte Nachteile. In den folgenden Abschnitten werde ich Ihnen eine Auswahl an Einsatzmöglichkeiten näher vorstellen, sodass Sie entscheiden können, welches Szenario für Ihr Vorhaben am besten geeignet ist.

Ich möchte aber schon vorausschicken, dass ich für die meisten Projekte einen Raspberry Pi als Basis einsetzen würde. Er ist günstig, verbraucht nicht viel Strom und eignet sich gut als kleine Serverlösung für den Hausgebrauch. Die meisten Beispiele in diesem Buch habe ich daher auf einem Raspberry Pi ausgeführt.

1.2.1 Node-RED auf einem Android-Gerät

Sie können Node-RED auf Android-Geräten nutzen – es läuft auf vielen modernen Smartphones und Tablets und entfernt sich dadurch deutlich vom klassischen Einsatz als Dienst auf einem Server. Daher möchte ich dieses Szenario im Zusammenhang mit einer Anwendung erörtern und verweise Sie deshalb auf [Kapitel 10](#), »[Apps und externe Anbindung](#)«.

1.2.2 Node-RED auf einem Windows-PC

Es gibt unterschiedliche Wege, um Node-RED in einer Windows-Umgebung zu installieren. Eine Möglichkeit ist, eine *virtuelle Maschine* wie Oracle VirtualBox oder Hyper-V einzurichten und dann Node-RED mit einer Linux-Distribution zu betreiben – Linux-Umgebungen sind sehr komfortabel, wenn es um Serverdienste geht, und werden zumeist für solche Aufgaben eingesetzt.

Sie können Node-RED aber auch direkt unter Windows betreiben. Ihr Windows-Rechner ist dann der Wirt (der Host) für den Dienst (Node-RED). Genauso wie ein guter Wirt mehrere Gäste betreuen kann, kann ein Host auch mehrere Aufgaben gleichzeitig übernehmen. Node-RED läuft dabei im Hintergrund, führt Aufgaben aus und wartet, bis es etwas zu tun gibt. Es werden also ständig einige Ressourcen belegt, und Node-RED steht nicht mehr zur Verfügung, wenn Sie Ihren Rechner ausschalten. Abgesehen von dieser Einschränkung können Sie Ihren Rechner wie gewohnt nutzen.

Grundsätzlich ist es aber eine gute Idee, Serverdienste von der alltäglichen Arbeit zu trennen. Und Sie sollten auch gut überlegen, ob Sie Experimente mit Node-RED auf Ihrem Arbeitsrechner durchführen wollen. Daher bietet es sich zumindest für Testzwecke an, Node-RED nicht auf dem »Bürorechner« zu installieren, sondern eher eine andere Plattform wie den Raspberry Pi zu wählen.

Wenn Sie aber dennoch mit Ihrer gewohnten Windows-Umgebung arbeiten wollen, ist das auch kein Problem. Installieren Sie zunächst die Grundvoraussetzungen und konfigurieren Sie dann Node-RED.

Da Node-RED auf Node.js basiert, müssen Sie zunächst diese Komponente installieren. Sie sorgt dafür, dass Sie eine JavaScript-Anwendung wie Node-RED auf Ihrem Rechner ausführen können.

Node.js installieren

Besuchen Sie als Erstes die Download-Seite von *Node.js* (<https://nodejs.org/en/download>, siehe [Abbildung 1.3](#)).

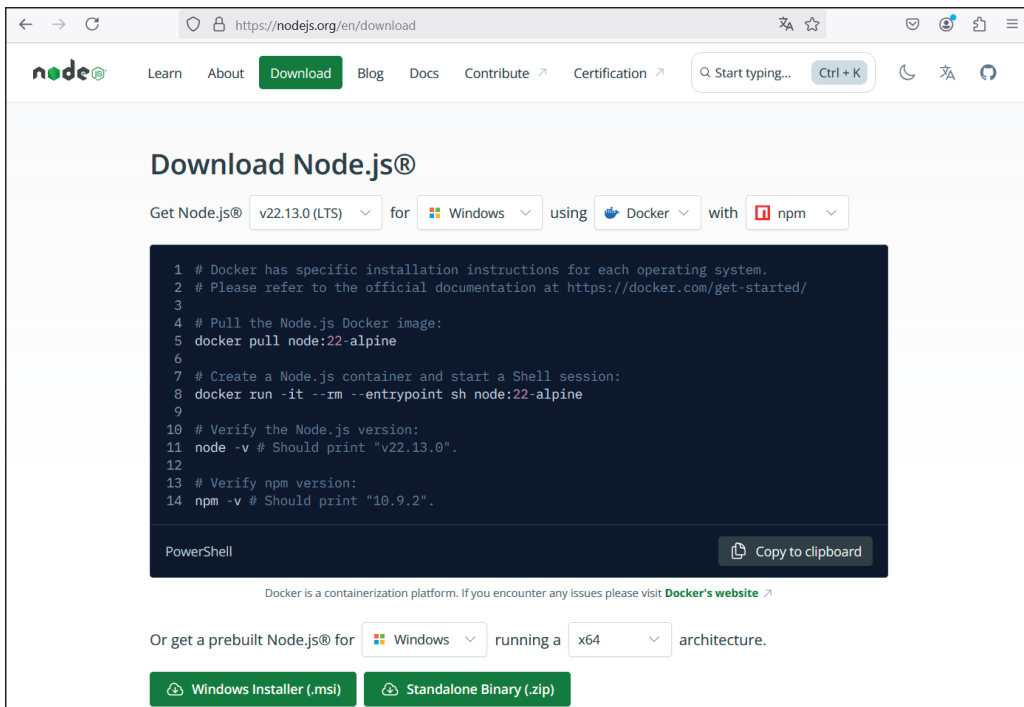


Abbildung 1.3: Download-Seite von Node.js

Laden Sie dort den Installer durch einen Klick auf **Windows Installer** herunter. Das System sucht die zutreffende Version aus. Führen Sie nun den Installer aus, der Setup-Wizard führt Sie dabei durch die Installation. Folgen Sie den Vorschlägen des Programms. Die In-

Installation benötigt lokale Administratorrechte. Eine Eingabeaufforderung erfragt das Administrationspasswort in den Fällen, in denen die Rechte nicht vorliegen. Der Installer speichert Node.js sowie npm, den *Node Package Manager*, per Voreinstellung in dem Ordner `C:\Program Files\nodejs\`.

npm ist der Paketmanager für die JavaScript-Laufzeitumgebung Node.js, mit dem Sie neue Module herunterladen. Er kümmert sich auch darum, dass alle Abhängigkeiten automatisch aufgelöst werden.

Überprüfen Sie den Erfolg der Installation:

- in der PowerShell: `node --version; npm --version`
- in der Eingabeaufforderung: `node --version && npm --version`

Als Ausgabe sollten Sie dann die entsprechenden Versionsangaben erhalten.

Node-RED installieren

Installieren Sie nun Node-RED über npm in der Windows PowerShell:

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

Der Parameter `-g` veranlasst, dass Node-RED als globales Modul eingerichtet wird. Das Flag `unsafe-perm` führt npm mit Administratorrechten aus – das ist notwendig, denn Node-RED soll ja als Dienst unabhängig von Ihrem Windows-Nutzer laufen.

Der Installationsprozess zeigt einige Warnungen an, die Sie aber ignorieren können, da sie nur optionale Abhängigkeiten betreffen. Er endet mit einer Mitteilung über die eingerichtete Version und die Anzahl der Pakete (siehe [Abbildung 1.4](#)).

```
npm WARN SKIPPING OPTIONAL DEPENDENCY: Exit status 1
+ node-red@1.2.9
added 303 packages from 282 contributors in 33.842s
PS C:\Users\ > █
```

Abbildung 1.4: Node-RED-Installation unter Windows

Node-RED starten

Starten Sie nun Node-RED auf Kommandozeilenebene mit dem Befehl `node-red`. Die Terminalmeldungen aus [Abbildung 1.5](#) geben eine Vielzahl wertvoller Hinweise. Einige Erläuterungen hierzu finden Sie in [Abschnitt 1.4](#). Lassen Sie das Fenster geöffnet, solange Sie mit dem Programm arbeiten wollen. Wenn Sie es schließen, beenden Sie auch Node-RED.

PowerShell versus Eingabeaufforderung

Der Start von Node-RED über die Eingabeaufforderung kann auf Ihrem System unter Umständen nicht möglich sein, weil Pfade nicht richtig gesetzt sind. Daher ist es bes-



ser, wenn Sie die moderne PowerShell nutzen. Aber auch hier kann es zu einer Fehlermeldung kommen, falls die Ausführung von Skripten auf dem System deaktiviert ist. Erlauben Sie dann die Skriptausführung:

1. Klicken Sie mit der rechten Maustaste auf die PowerShell-Verknüpfung und wählen Sie **Als Administrator ausführen**.
2. Führen Sie hier den Befehl `Set-ExecutionPolicy RemoteSigned` aus.

Bedenken Sie, dass Sie damit das Sicherheitslevel Ihres Systems mindern. Wenn Sie regelmäßig PowerShell-Skripte nutzen, sollten Sie die Einstellung individuell für Node-RED anpassen.

```
C:\Users\>node-red
7 Mar 13:57:06 - [info]
Welcome to Node-RED
=====
7 Mar 13:57:06 - [info] Node-RED version: v1.2.9
7 Mar 13:57:06 - [info] Node.js version: v14.16.0
7 Mar 13:57:06 - [info] Windows_NT 10.0.19041 x64 LE
7 Mar 13:57:07 - [info] Loading palette nodes
7 Mar 13:57:08 - [info] Settings file : C:\Users\...\node-red\settings.js
7 Mar 13:57:08 - [info] Context store : 'default' [module=memory]
7 Mar 13:57:08 - [info] User directory : C:\Users\...\node-red
7 Mar 13:57:08 - [warn] Projects disabled : editorTheme.projects.enabled=false
7 Mar 13:57:08 - [info] Flows file : C:\Users\...\node-red\flows_LAPTOP-M559PT0A.json
7 Mar 13:57:08 - [info] Creating new flow file
7 Mar 13:57:08 - [warn]
-----
Your flow credentials file is encrypted using a system-generated key.
If the system-generated key is lost for any reason, your credentials
file will not be recoverable, you will have to delete it and re-enter
your credentials.
You should set your own key using the 'credentialSecret' option in
your settings file. Node-RED will then re-encrypt your credentials
file using your chosen key the next time you deploy a change.
-----
7 Mar 13:57:08 - [info] Server now running at http://127.0.0.1:1880/
7 Mar 13:57:08 - [info] Starting flows
7 Mar 13:57:08 - [info] Started flows
C:\Users\>
```

Abbildung 1.5: Node-RED wurde auf der Windows-Kommandozeile gestartet.

Falls Sie Node-RED als Dienst betreiben möchten, der ständig zur Verfügung steht und sich um die Belange der Hausautomation kümmert oder stets Ihr persönlicher Assistent ist, müssen Sie dafür sorgen, dass er bereits mit dem Hochfahren des Systems gestartet wird. Verwenden Sie dafür den Windows-Task-Scheduler:

- Öffnen Sie das Windows-Startmenü und suchen Sie nach »Aufgabenplanung«.
- Folgen Sie den weiteren Schritten über **Aktion • Aufgabe erstellen**.
- Legen Sie unter **Trigger** und **Aktionen** fest, dass Node-RED beim Systemstart gestartet wird.

Eine Installation von Node-RED auf einem Windows-PC (wie auch auf einem Desktoprechner mit Linux oder macOS) hat den Vorteil, dass Ihnen dort sehr wahrscheinlich große Systemressourcen zur Verfügung stehen. Moderne PCs haben mehr als genug Power, um Node-RED als Dienst zu betreiben. Ein weiterer Vorteil ist die vergleichsweise hohe Zuverlässigkeit der Systeme.

Ein großer Nachteil ist der Stromverbrauch bei einem ständigen Betrieb, also wenn Node-RED rund um die Uhr (24/7) laufen soll. Dafür sind Desktop-PCs weniger gut geeignet; moderne Minirechner wie der Raspberry Pi sind aber für diese Aufgabe wie gemacht.

Überlegen Sie sich also, wie Sie Node-RED nutzen wollen. Für erste Gehversuche bietet sich der Einsatz auf Ihrem Desktop-PC durchaus an; für den Einsatz im Alltag ist Ihr Rechner wahrscheinlich vollkommen überdimensioniert, was sich dann auf der Stromrechnung bemerkbar macht. So liegen die Kosten bei einem durchschnittlichen PC (Stromverbrauch ca. 150 W/h) überschlägig um den Faktor 30 höher als bei einem Raspberry Pi (Stromverbrauch ca. 5 W/h).

1.2.3 Node-RED auf einem Raspberry Pi

Der *Raspberry Pi* hat seine Karriere als kleiner Bastelrechner gestartet, inzwischen gilt er aber mit einer gewissen Berechtigung als zuverlässige und ressourcensparende Plattform für alle Anwendungen, die rund um die Uhr erreichbar sein sollen. Daher ist er in der Maker-Szene fest verankert und zum Fundament zahlreicher Bastelprojekte geworden. In den vergangenen Jahren ist der Raspberry Pi deutlich leistungsfähiger geworden. Das aktuelle Modell Raspberry Pi 5 kann problemlos mit einfachen PCs mithalten, die ein bisschen in die Jahre gekommen sind.

Die Macher von Node-RED unterstützen Sie bei der Installation mit einem Skript, das auf Ihre Umgebung ausgerichtet ist. Sie sollten auf dieses Skript zurückgreifen, obwohl Node-RED auch als Paket in den vordefinierten *Repository*s von Raspberry Pi OS vorhanden ist. (Ein Repository [dt. Lager, Depot] ist ein verwaltetes Verzeichnis zur Speicherung und Beschreibung digitaler Objekte wie Dateien.) Der Grund für meine Empfehlung ist, dass der Paketmanager `npm` von Haus aus nicht in diesen Paketen enthalten ist.

Öffnen Sie ein Terminal auf dem Raspberry Pi und gehen Sie wie folgt vor:

```
sudo apt update && sudo apt upgrade
bash <(curl -sL https://raw.githubusercontent.com/node-red/
  linux-installers/master/deb/update-nodejs-and-nodered)
```

Der erste Befehl aktualisiert Ihr System, der zweite holt das Installationskript und führt es aus. Das ist sehr komfortabel, stellt aber auch eine kleine Gefahr für die Sicherheit dar – prinzipiell ist es nichts anderes, als eine `.exe`-Datei aus dem Internet herunterzuladen und auszuführen. Sie können aber den Inhalt des Skripts vorher genau prüfen, indem Sie es sich zunächst auf der Webseite anschauen.

Zu Beginn bittet das Skript Sie um eine Bestätigung des Installationsprozesses und fragt, ob Sie die Raspberry-Pi-spezifischen Nodes installieren möchten. Beantworten Sie die Fragen mit **Y**. Danach lädt das Skript die notwendigen Pakete herunter und installiert sie. Der gesamte Prozess dauert eine Weile. Die einzelnen Schritte lassen sich im Terminal gut verfolgen (siehe [Abbildung 1.6](#)).

```

Running Node-RED install for user pi at /home/pi on debian
This can take 20-30 minutes on the slower Pi versions - please wait.
  Stop Node-RED                               ✓
  Remove old version of Node-RED               ✓
  Remove old version of Node.js                ✓
  Install Node 20.18.1-inodesource1           ✓ v20.18.1  Npm 10.8.2
  Clean npm cache                              ✓
  Install Node-RED core                        ✓ 4.0.8
  Move global nodes to local                   -
  Npm rebuild existing nodes                  ✓
  Install extra Pi nodes                      ✓
  Add shortcut commands                       ✓
  Update systemd script                       ✓

Any errors will be logged to /var/log/nodered-install.log
All done.
You can now start Node-RED with the command node-red-start
or using the icon under Menu / Programming / Node-RED
Then point your browser to localhost:1880 or http://{your_pi_ip-address}:1880

Started : Tue 7 Jan 15:09:12 CET 2025
Finished: Tue 7 Jan 15:10:30 CET 2025

*****
### WARNING ###
DO NOT EXPOSE NODE-RED TO THE OPEN INTERNET WITHOUT SECURING IT FIRST

```

Abbildung 1.6: Installation von Node-RED auf einem Raspberry Pi

Die Ausgaben in der Konsole können bei Ihnen etwas abweichen; sie verdienen jedoch durchaus ein kleines Augenmerk. Die erfolgreich abgearbeiteten Schritte weisen einen grünen Haken auf.

Beachtenswert sind die Angaben zu den Versionen der installierten Softwarepakete. Diese Information brauchen Sie, denn wenn Sie später andere Node-Pakete installieren, können diese andere Mindestanforderungen stellen. Zudem sollten Sie diese Versionsangaben kennen, wenn Sie Hilfe in Foren suchen, denn ohne dieses Wissen kann ein fremder Dritter kaum zielführend helfen.

Darauf folgen in der Konsolen-Ausgabe ein erster Hinweis zum Start von Node-RED (`node-red-start`) und zum Zugang zum Webinterface (`localhost:1880`). Anschließend informiert das Skript über ein paar Sicherheitsaspekte (siehe auch [Abbildung 1.6](#)). Den Abschluss bildet die Ersteinrichtung der Konfigurationsdatei `settings.js` (siehe [Abbildung 1.7](#)).

Für den Anfang reicht es, die vorgeschlagenen Werte mit **Y** zu übernehmen bzw. auf das Setup der *user security* mit **no** zu verzichten. Sie können die Ergänzungen zu einem späteren Zeitpunkt problemlos nachholen. Die Einstellungen greife ich auch im Verlaufe des Buches näher auf.

```

Node-RED Settings File Initialisation
=====
This tool will help you create a Node-RED settings file.
✓ Settings file · /home/pi/.node-red/settings.js
User Security
=====
✓ Do you want to setup user security? · No
Projects
=====
The Projects feature allows you to version control your flow using a local git repository.
✓ Do you want to enable the Projects feature? · No
Flow File settings
=====
✓ Enter a name for your flows file · flows.json
✓ Provide a passphrase to encrypt your credentials file ·
Editor settings
=====
✓ Select a theme for the editor. To use any theme other than "default", you will need to install @node-red-contrib-themes/theme-collection in your Node-RED user directory. · default
✓ Select the text editor component to use in the Node-RED Editor · monaco (default)
Node settings
=====
✓ Allow Function nodes to load external modules? (functionExternalModules) · Yes
Settings file written to /home/pi/.node-red/settings.js
pi@pifenster:~$ █

```

Abbildung 1.7: Installation von Node-RED – Konfigurationsdatei

Node-RED startet danach nicht automatisch. Öffnen Sie daher nach der Installation ein Terminal und rufen Sie Node-RED mit `node-red-start` auf. Alternativ geht das auch über die grafische Benutzeroberfläche und das Icon unter **Menu • Entwicklung • Node-RED**.

Sie beenden Node-RED durch Eingabe von `node-red-stop` in einem weiteren Terminal. Zudem können Sie Node-RED im Hintergrund als Systemprozess ausführen.

- Als Systemprozess starten Sie Node-RED mit:

```
sudo systemctl start nodered.service
```

- Den aktuellen Status des Diensts erfahren Sie mit:

```
sudo systemctl status nodered.service
```

- Den Systemprozess beenden Sie mit:

```
sudo systemctl stop nodered.service
```

- Wenn Node-RED als Dienst ständig auf Ihrem Raspberry Pi verfügbar sein soll, richten Sie den Start beim Bootvorgang ein:

```
sudo systemctl enable nodered.service
```

Der Raspberry Pi ist die ideale Plattform für Node-RED und auch die häufigste Installationsvariante. Daher ist dies die Basis für meine weiteren Ausführungen, die selbstverständlich aber auch für Node-RED in allen anderen Systemumgebungen gelten.



Versionswechsel von Raspberry Pi OS

Auch Raspberry Pi OS erfährt Versionswechsel. Die Portierung von alt nach neu ist nahezu immer ein zumindest zeitaufwendiges Vorhaben. Besser ist es, wenn Sie bei einer neuen Version eine komplette Neuinstallation durchführen. Richten Sie Node-RED dann auf der neuen Version mit dem genannten Installationskript ein.

Für die Neuinstallation der verwendeten Nodes erstellen Sie eine Datei mit folgenden Anweisungen:

```
cd ~/.node-red/  
for addonnodes in moment node-red-contrib-bigtimer \  
node-red-contrib-telegrambot ; do  
  npm $NQUIET install --save ${addonnodes}  
done
```

Ergänzen Sie die Paketliste Ihren Bedürfnissen entsprechend um andere Pakete, z. B. um `node-red-contrib-fritz`. Machen Sie die Datei ausführbar und lassen Sie die Pakete hinzufügen. Die Übertragung der Flows erfolgt aus dem Dateisystem (siehe [Abschnitt 1.5, »Node-RED administrieren«](#)).

1.2.4 Node-RED auf einem Linux-System

Die Installation von Node-RED auf einem »normalen« Linux-System hängt von dessen Distribution ab. Für *Debian*-basierte Distributionen wie etwa *Ubuntu* entspricht der Installationsweg dem beim Raspberry Pi. Sie brauchen vorher aber noch einige Pakete, die Sie mit einem vorgeschalteten Kommando installieren:

```
sudo apt install build-essential git
```

Dies sorgt dafür, dass `npm` zu installierende Binärmodule bauen kann. Danach können Sie so vorgehen, wie in [Abschnitt 1.2.3](#) beschrieben.

Ähnlich läuft die Installation auf andere Distributionen ab. Konsultieren Sie die Dokumentation, falls es Probleme bei der Installation von `Node.js` und `npm` gibt.



Bestehende Node.js-Installationen

Wenn Sie `Node.js` bereits installiert haben, müssen Sie aufpassen, denn redundante Installationen sorgen stets für Probleme und können Fehler beim Ausführen des Skripts zur Folge haben.

Grundsätzlich sollten Sie die aktuelle Version von `Node.js` installieren. Wenn Sie explizit eine Vorgängerversion verwenden müssen, bietet das Installationskript entsprechende Optionen an (siehe [Abbildung 1.8](#)). Sie können dann eine konkrete

Node.js-Version vorgeben, indem Sie diese als Option beim Bash-Aufruf mitgeben. Das sollte aber die Ausnahme bleiben.

```

Datei Bearbeiten Reiter Hilfe
pi@raspberrypi:~ $ bash <(curl -sL https://raw.githubusercontent.com/node-red/linux-installers/master/deb/update-nodejs-and-nodered) --help
Usage: /dev/fd/63 [options]

Options:
--help                display this help and exit
--confirm-root        install as root without asking confirmation
--confirm-install     confirm installation without asking confirmation
--confirm-pi          confirm installation of PI specific nodes without asking confirmation
--skip-pi             skip installing PI specific nodes without asking confirmation
--restart             restart service if install succeeds
--allow-low-ports     add capability to bind to ports below 1024 (default is disallow)
--update-nodes        run npm update on existing installed nodes (within scope of package.json)
--no-init             don't ask to initialise settings if they don't exist
--nodered-user        specify the user to run as, useful for installing as sudo - e.g. --nodered-user=pi
--nodered-version     if not set, the latest version is used - e.g. --nodered-version="3.0.1"
--node12              if set, forces install of major version of nodejs 12 LTS
--node14              if set, forces install of major version of nodejs 14 LTS
--node16              if set, forces install of major version of nodejs 16 LTS
--node18              if set, forces install of major version of nodejs 18 LTS
                     if none set, install nodejs 16 LTS if nodejs version is less than 14,
                     otherwise leave current install

Note: if you use allow-low-ports it may affect the node modules paths - see https://stackoverflow.com/a/65560687
pi@raspberrypi:~ $ █

```

Abbildung 1.8: Bash-Optionen des Installationskripts

Exkurs: Beta-Versionen ausprobieren

Vielleicht verspüren Sie ja einmal das Bedürfnis, eine Beta-Version von Node-RED auszuprobieren. Gehen Sie dann wie folgt vor:

1. Installieren der korrekten Node.js-Version

Installieren Sie die korrekte Node.js-Version. Das ist wichtig. Sie ist in der Regel der Release-Ankündigung zu entnehmen. Im Beispiel laden wir die Dateien für die Version 22 herunter:

```
curl -fsSL https://deb.nodesource.com/setup_22.x | sudo -E bash
- sudo apt install -y nodejs
```

Sie können sich die installierten Versionen anzeigen lassen:

```
node -v
npm -v
```

2. Entfernen der alten Node-RED Installation

Entfernen Sie eine eventuell installierte alte Node-RED-Version:

```
sudo npm remove -g node-red
```

3. Installieren der Beta-Version

Installieren Sie die Beta-Version. Entnehmen Sie auch hier die korrekte Bezeichnung der Release-Ankündigung (Beispiel):

```
sudo npm install -g --unsafe-perm node-red@5.0.0-beta.4
```

4. Node-RED starten

Starten Sie zunächst Node-RED mit:

```
node-red
```

Bei Erfolg haben Sie den bekannten Zugriff über den Browser: <http://localhost:1880>

1.3 Node-RED überall: Docker-Container starten

Docker ist eine Software, die die Bereitstellung von Anwendungen vereinfacht, indem sie die nötigen Pakete in Container packt. Ihr Ziel ist es, Dateien auf einfache Weise zu transportieren und zu installieren – Sie müssen nur den Container starten und sind fertig. Aufwendige Installationen, bei denen Sie zunächst andere Vorbereitungen treffen müssen, lassen sich so deutlich vereinfachen. Gleichzeitig trennen und verwalten Container die Ressourcen eines Rechners. In gewisser Weise verhalten sich die Container wie extrem leichtgewichtige, modulare virtuelle Maschinen. Die Basis von Docker sind Linux-Techniken wie *Cgroups* und *Namespaces*. Mehr Informationen erhalten Sie auf <https://docs.docker.com/>.

Docker läuft auf Windows-, macOS- und Linux-Maschinen und sogar auf dem Raspberry Pi. Zwei Linux-Installationsvarianten möchte ich Ihnen vorstellen. Falls Sie Docker unter Windows nutzen wollen, steht Ihnen unter <https://docs.docker.com/docker-for-windows/install/> der *Docker Desktop* zur Verfügung. Da Docker jedoch eine Linux-Software ist, müssen Sie unter Windows zunächst noch das *Windows Subsystem for Linux* (WSL) aktivieren.

1.3.1 Docker auf einem Ubuntu-System

Docker ist auf einem Ubuntu-System innerhalb weniger Minuten eingerichtet. Aktualisieren Sie die Paketliste und installieren Sie benötigte Pakete nach:

```
sudo apt update
sudo apt install apt-transport-https ca-certificates curl gnupg
```

Fügen Sie den Docker-GPG-Key hinzu:

```
curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo gpg
--dearmor
-o /usr/share/keyrings/docker-archive-keyring.gpg
```

Anschließend fügen Sie das Docker-Repository hinzu. Das folgende Kommando unterstellt, dass Ihr System auf einer x84_64- bzw. amd64-Architektur läuft. Dies ist bei aktuellen PCs überwiegend der Fall.

```
echo "deb [arch=amd64 signed-by=/usr/share/keyrings/docker-archive-
keyring.gpg] https://download.docker.com/linux/ubuntu $(lsb_release
-cs) stable" | sudo tee /etc/apt/sources.list.d/docker.list > /dev/
null
```

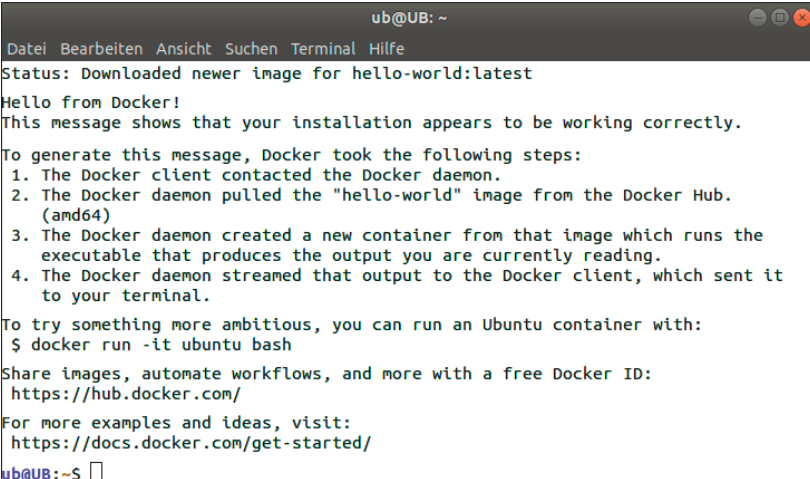
Jetzt können Sie die Docker-Pakete installieren:

```
sudo apt update
sudo apt install docker-ce docker-ce-cli containerd.io
```

Testen Sie die erfolgreiche Installation von Docker:

```
sudo docker run hello-world
```

Sie sollten eine Bildschirmausgabe gemäß [Abbildung 1.9](#) erhalten, in der ein kleiner Con-
tainer heruntergeladen und ausgeführt wird, der die Zeile »Hello from Docker!« ausgibt.



```
ub@UB: ~
Datei Bearbeiten Ansicht Suchen Terminal Hilfe
Status: Downloaded newer image for hello-world:latest
Hello from Docker!
This message shows that your installation appears to be working correctly.

To generate this message, Docker took the following steps:
 1. The Docker client contacted the Docker daemon.
 2. The Docker daemon pulled the "hello-world" image from the Docker Hub.
    (amd64)
 3. The Docker daemon created a new container from that image which runs the
    executable that produces the output you are currently reading.
 4. The Docker daemon streamed that output to the Docker client, which sent it
    to your terminal.

To try something more ambitious, you can run an Ubuntu container with:
$ docker run -it ubuntu bash

Share images, automate workflows, and more with a free Docker ID:
https://hub.docker.com/

For more examples and ideas, visit:
https://docs.docker.com/get-started/
ub@UB:~$
```

Abbildung 1.9: Die Hello-World-Ausgabe von Docker

Überdies erhalten Sie mit den Befehlen `sudo docker version` sowie `sudo docker info` Informationen zur Docker-Installation.

1.3.2 Docker auf dem Raspberry Pi

Das Docker-Team bietet für die Installation auf einem Raspberry Pi ein eigenes Installationskript an. Mit dem folgenden Befehl laden Sie es herunter und führen es aus:

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

Der gesamte Vorgang nimmt einige Zeit in Anspruch. Am Ende sollten Sie eine Bildschirmausgabe gemäß [Abbildung 1.10](#) erhalten.

```

Engine:
  Version:      20.10.5
  API version:  1.41 (minimum version 1.12)
  Go version:   go1.13.15
  Git commit:   363e9a8
  Built:        Tue Mar 2 20:16:18 2021
  OS/Arch:      linux/arm
  Experimental: false
containerd:
  Version:      1.4.4
  GitCommit:    05f951a3781f4f2c1911b05e61c160e9c30eaa8e
runc:
  Version:      1.0.0-rc93
  GitCommit:    12644e614e25b05da6fd08a38ffa0cfe1903fdec
docker-init:
  Version:      0.19.0
  GitCommit:    de40ad0
If you would like to use Docker as a non-root user, you should now consider
adding your user to the "docker" group with something like:
    sudo usermod -aG docker pi
Remember that you will have to log out and back in for this to take effect!
WARNING: Adding a user to the "docker" group will grant the ability to run
containers which can be used to obtain root privileges on the
docker host.
Refer to https://docs.docker.com/engine/security/security/#docker-daemon-attack-surface
for more information.
pi@raspberrypi:~$ █

```

Abbildung 1.10: Docker auf dem Raspberry Pi installieren – Abschlussmeldungen

Die erfolgreiche Installation können Sie wiederum mit folgender Anweisung testen:

```
sudo docker run armhf/hello-world
```

Die Bildschirmausgabe sollte im Wesentlichen der aus [Abbildung 1.9](#) entsprechen.

1.3.3 Node-RED in Docker ausführen

Das Kommando, um Node-RED in einem Docker-Container auszuführen, hat mehrere Bestandteile. Dies sind im Einzelnen:

- `docker run` – führt den Container aus.
- `-it` – öffnet eine Terminal-Session, über die Sie das Geschehen verfolgen können.
- `-p 1880:1880` – stellt eine Verbindung von Port 1880 zu Port 1880 im Container her, auf dem Node-RED läuft.
- `-v node_red_data:/data` – bindet das Node-RED-Datenverzeichnis in das Verzeichnis `/data` des Docker-Containers ein. Sie können dadurch aus dem Container auf die Dateien zugreifen, die sich dort befinden. Gleichzeitig erreichen Sie so von Ihrem Host aus die Daten, die Node-RED innerhalb des Containers schreibt (siehe dazu auch [Abschnitt 1.5.2](#)).
- `--name mynodered` – legt einen benutzerdefinierten Namen für Ihren Container fest, hier `mynodered`. Wenn Sie diese Option auslassen, vergibt Docker automatisch einen Fantasienamen.

- `nodered/node-red` – legt schließlich das Docker-Image fest, das benutzt werden soll. In unserem Fall ist es das offizielle Node-RED-Image. Sie finden Informationen dazu im DockerHub unter <https://hub.docker.com/r/nodered/node-red>.

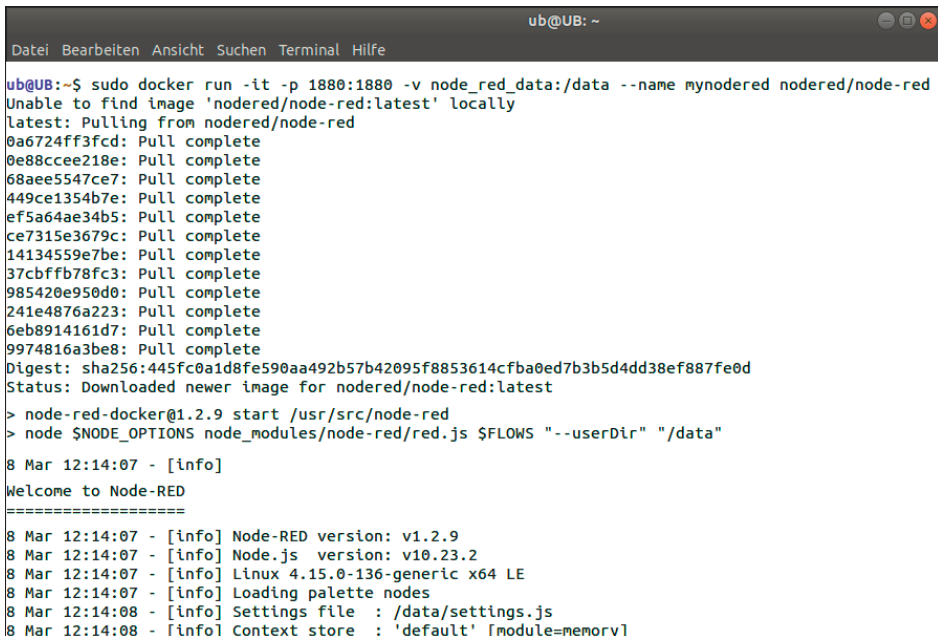
Geben Sie folgenden Befehl ein, um Node-RED in einem Docker-Container auszuführen:

```
sudo docker run -it -p 1880:1880 -v node_red_data:/data \
  --name mynodered \nodered/node-red
```

Docker bezieht das Node-RED-Image aus der offiziellen Docker-Registry und startet die Anwendung (siehe [Abbildung 1.11](#)). Mit einem Befehl sind die komplette Installation und das Starten des Diensts erledigt.

Auf dem Raspberry Pi nutzen Sie den gleichen Befehl. Da Ihr Raspberry Pi wahrscheinlich sehr viel weniger Rechenpower als Ihr Desktoprechner hat, dauert es eine Weile, bis alle nötigen Schritte abgeschlossen sind.

Sie kehren mit `[Strg]+[Q]` bzw. `[Strg]+[P]` zur Kommandoebene zurück; Docker läuft aber im Hintergrund weiter.



```

ub@UB: ~
Datei Bearbeiten Ansicht Suchen Terminal Hilfe
ub@UB:~$ sudo docker run -it -p 1880:1880 -v node_red_data:/data --name mynodered nodered/node-red
Unable to find image 'nodered/node-red:latest' locally
latest: Pulling from nodered/node-red
0a6724ff3fcd: Pull complete
0e88ccee218e: Pull complete
68aee5547ce7: Pull complete
449ce1354b7e: Pull complete
ef5a64ae34b5: Pull complete
ce7315e3679c: Pull complete
14134559e7be: Pull complete
37cbffb78fc3: Pull complete
985420e950d0: Pull complete
241e4876a223: Pull complete
6eb8914161d7: Pull complete
9974816a3be8: Pull complete
Digest: sha256:445fc0a1d8fe590aa492b57b42095f8853614cfba0ed7b3b5d4d38ef887fe0d
Status: Downloaded newer image for nodered/node-red:latest
> node-red-docker@1.2.9 start /usr/src/node-red
> node $NODE_OPTIONS node_modules/node-red/red.js $FLOWS "--userDir" "/"data"
8 Mar 12:14:07 - [info]
Welcome to Node-RED
=====
8 Mar 12:14:07 - [info] Node-RED version: v1.2.9
8 Mar 12:14:07 - [info] Node.js version: v10.23.2
8 Mar 12:14:07 - [info] Linux 4.15.0-136-generic x64 LE
8 Mar 12:14:07 - [info] Loading palette nodes
8 Mar 12:14:08 - [info] Settings file : /data/settings.js
8 Mar 12:14:08 - [info] Context store : 'default' [module=memorv]

```

Abbildung 1.11: Node-RED in einem Docker-Container ausführen

Nutzen Sie

- `sudo docker attach mynodered`, um wieder in die Ausgabe zurückzukehren,
- `sudo docker restart mynodered`, um den Container neu zu starten,

- `sudo docker stop mynodered`, um den Container zu beenden, und
- `sudo docker rm mynodered`, um den Container zu löschen.

Docker bietet sehr viele Optionen und macht das Ausprobieren von neuer Software sehr einfach – wenn man einmal das Prinzip hinter Softwarecontainern verstanden hat. Schlagen Sie dazu am besten in der sehr ausführlichen Dokumentation unter <https://docs.docker.com/> nach. Deutschsprachige Informationen finden Sie beispielsweise in Öggl/Kofler: Docker. Das Praxisbuch für Entwickler und DevOps-Teams. Rheinwerk Verlag 2026.

1.4 Die Ausgaben von Node-RED beim Start

Node-RED verfügt über ein ausgezeichnetes und aussagekräftiges Logging, das bereits während des Startvorgangs auf der Konsole wertvolle Hinweise über den Zustand der Software gibt. Unabhängig von der Systemumgebung erhalten Sie beim Start in einem Terminal mehrere Meldungen zum Status der Anwendung, wie [Abbildung 1.12](#) zeigt.

```

pi@rpfenster:~ $ node-red-start
Start Node-RED
Once Node-RED has started, point a browser at http://192.168.178.105:1880
On Pi Node-RED works better with the Firefox or Chrome browser

Use  node-red-stop           to stop Node-RED
Use  node-red-start         to start Node-RED again
Use  node-red-log           to view the recent log output
Use  sudo systemctl enable nodered.service to autostart Node-RED at every boot
Use  sudo systemctl disable nodered.service to disable autostart on boot

To find more nodes and example flows - go to http://flows.nodered.org

Starting as a systemd service.
8 Jan 09:51:18 - [info]
Welcome to Node-RED
=====
8 Jan 09:51:18 - [info] Node-RED version: v4.0.8
8 Jan 09:51:18 - [info] Node.js version: v20.18.1
8 Jan 09:51:18 - [info] Linux 6.6.62+rpt-rpi-2712 arm64 LE
8 Jan 09:51:18 - [info] Loading palette nodes
8 Jan 09:51:20 - [info] Settings file   : /home/pi/.node-red/settings.js
8 Jan 09:51:20 - [info] Context store  : 'default' [module=memory]
8 Jan 09:51:20 - [info] User directory : /home/pi/.node-red
8 Jan 09:51:20 - [warn] Projects disabled : editorTheme.projects.enabled=false
8 Jan 09:51:20 - [info] Flows file     : /home/pi/.node-red/flows.json
8 Jan 09:51:20 - [info] Creating new flow file
8 Jan 09:51:20 - [warn] Using unencrypted credentials
8 Jan 09:51:20 - [info] Starting flows
8 Jan 09:51:20 - [info] Started flows
8 Jan 09:51:20 - [info] Server now running at http://127.0.0.1:1880/

```

Abbildung 1.12: Startmeldungen von Node-RED

Fast ausnahmslos versieht Node-RED die Meldungen mit einem Zeitstempel. Wertvoller als die Kenntnis über den Zeitpunkt des Programmstarts ist diese Information im laufenden Betrieb bei der Ausgabe von Fehlermeldungen. Sie hilft Ihnen dabei, Fehlerquellen einzugrenzen.

Nach der Willkommensmeldung werden die Versionsnummern der Kernsoftware und des Betriebssystems protokolliert. Diese Angaben sind unerlässlich, wenn Sie in Foren Hilfestellung bei Problemen suchen.

Es folgen mehrere Angaben zum Node-RED-Dateisystem (siehe [Abschnitt 1.5](#)) und zu den Systemeinstellungen von Node-RED. Die Warnmeldung hinsichtlich der Flow-Berechtigungen können Sie ignorieren; Näheres finden Sie in [Abschnitt 1.5.1](#).

Darunter meldet der Node-RED-Server, dass der Editor und das Dashboard gestartet wurden, und er gibt an, unter welcher Adresse Sie den Server in einem Browser erreichen, der auf dem gleichen Gerät läuft. Starten Sie beispielsweise den Chromium auf Ihrem Raspberry Pi und geben Sie `127.0.0.1:1880` ein. Mehr dazu finden Sie in [Kapitel 2](#), »[Das zentrale Tool: der Node-RED-Editor](#)«.

Logging während des Starts des Diensts beim Bootprozess

Node-RED unterstützt aktuell nur das Logging an eine Konsole. Die Anweisung `node-red-log` dupliziert die Log-Meldungen, wenn Node-RED während des Bootvorgangs als Systemdienst gestartet wurde und Meldungen zusätzlich in einer Konsole ausgegeben werden sollen. Diese Ausgaben lassen sich dann wiederum mit dem `>`-Operator in eine Datei umleiten:

```
node-red-log > meinlog.txt
```



1.5 Node-RED administrieren

In diesem Abschnitt werfen wir einen Blick hinter die Kulissen. Für den Einstieg ist dieser eigentlich nicht erforderlich, doch das kann sich im Laufe der Zeit ändern, wenn spezifische Fragestellungen auftreten, z. B. wenn es um den Speicherort von Flows und Nodes innerhalb von Node-RED geht.

1.5.1 Dateien und Ordner

Für Node-RED ist der Ordner `~/.node-red` von zentraler Bedeutung. Bei Linux bzw. auf dem Raspberry Pi ist dies ein versteckter Ordner in Ihrem Home-Verzeichnis (z. B. `~/home/pi`); unter Windows befindet sich der Pfad in `C:\Benutzer\<Ihr Benutzername>`. Er enthält einige Dateien mit wesentlichen Funktionen:

- `~/.node-red/node_modules`

Der Ordner `node_modules` enthält die installierten Nodes. Sie sollten die Nodes jedoch mit dem *Node-RED-Editor* verwalten, wie ich es in [Kapitel 2](#) zeige. Dies ist einfacher und sicherer, denn Informationen zu den installierten Nodes befinden sich auch in der Datei `config-nodes.json` sowie in der Datei `package.json`. Von Hand sollten Sie diese Dateien nicht ändern.

■ `~/.node-red/flows.json`

Die Datei `flows.json` enthält alle Flows im JSON-Format (siehe ebenfalls [Kapitel 2](#)). Sichern Sie diese Datei vor jeder Änderung an Node-RED, da einige Updates diese Datei in den Urzustand zurückversetzen. Denken Sie auch an eine Sicherung, falls Sie einmal Ihren Rechnernamen ändern. (Den Hostnamen eines Raspberry Pi legen Sie beispielsweise unter **Raspberry-Pi Konfiguration • System** fest.)

■ `~/.node-red/flows_raspberrypi_cred.js`

Die Datei `flows_raspberrypi_cred.js` enthält Zugangsdaten für die Flows, falls diese benötigt werden. Entsprechend den Einstellungen zu `credentialSecret` sind die Angaben entweder verschlüsselt oder im Klartext abgelegt (siehe [Abschnitt 1.5.3](#)).

■ `~/.node-red/settings.js`

Der Inhalt der Datei `settings.js` beeinflusst die Konfiguration von Node-RED. Um sie geht es in [Abschnitt 1.5.3](#) ausführlicher.

■ `~/.node-red/lib/flows`

Der Ordner `flows` enthält Bibliotheken von Flows, die Sie exportieren können (siehe [Kapitel 2](#), »Das zentrale Tool: der Node-RED-Editor«).

■ `~/.npm`

`.npm` ist ein versteckter Ordner in Ihrem Home-Verzeichnis. Er nimmt lokal installierte npm-Pakete auf.



Node-RED von Ihrem Gerät entfernen

Vielleicht tritt der – unwahrscheinliche – Fall ein, dass Sie Node-RED wieder komplett von Ihrem System entfernen möchten. Beenden Sie dann zunächst Node-RED:

```
node-red-stop
```

bzw.:

```
sudo systemctl stop Node-RED.service
```

```
sudo systemctl disable Node-RED.service
```

Entfernen Sie nun die Node-RED-Pakete:

```
sudo npm -g remove node-red
```

```
sudo npm -g remove node-red-admin
```

Entfernen Sie dann den Konfigurationsordner:

```
rm -R ~/.node-red
```

Falls Node-RED die einzige Anwendung war, die auf Node.js basierte, können Sie natürlich auch noch diese Abhängigkeit entfernen, die nun nicht mehr benötigt wird:

```
sudo apt -y remove --purge nodejs
```

```
sudo apt -y autoremove
```

Entfernen Sie außerdem folgende Dateien:

```
sudo rm -rf .npm .node-gyp
sudo rm /usr/share/applications/Node-RED.desktop
sudo rm /usr/share/icons/hicolor/scalable/apps/node-red-icon.svg
```

1.5.2 Dateien unter Docker

Standardmäßig liegen die Anwendungsdaten bei Docker in einem Container. Über den Parameter `-v` führen Sie diese Dateien und Ordner nach »außen«, damit Sie von Ihrem Hostsystem darauf zugreifen können:

```
sudo docker run -it -p 1880:1880 -v /home/pi/.node-red:/data \
  --name mynodered nodered/node-red
```

Durch die Angabe von `/home/pi/.node-red` finden Sie in diesem Ordner die gleiche Datenstruktur, wie sie bereits im vorherigen Abschnitt beschrieben wurde. Sie können dort dann Dateien ablegen und bearbeiten, die Sie im Docker-Container brauchen.

1.5.3 Einstellungen von Node-RED ändern

Node-RED ist nach der Installation direkt betriebsbereit, denn es bringt eine vorgefertigte Konfigurationsdatei namens *settings.js* mit. Ab Version 2.0 besteht aber auch die Möglichkeit, diese Datei skriptgesteuert durch ein Frage-und-Antwort-Spiel zu erstellen. Rufen Sie dazu folgenden Befehl auf:

```
node-red admin init
```

Das Problem ist jedoch, dass Sie sich im Voraus recht gut Ihre Eingaben überlegen müssen. Für einen Überblick über die diversen Einstellungen ist das Skript aber allemal zu empfehlen. Sie können die Einstellungen dann anschließend anpassen, egal ob es sich um die Standardwerte handelt oder ob Sie dialoggeführt eigene Werte festgelegt haben.

Die Einstellungen von Node-RED ändern Sie in der Datei *settings.js*. An dieser Stelle möchte ich nur auf die wichtigsten Grundeinstellungen eingehen. Die zahlreichen weiteren Optionen werden an den passenden Stellen dieses Buchs im Kontext beschrieben.

■ uiPort

Das User-Interface von Node-RED, mit dem Sie sich über den Browser verbinden (siehe [Kapitel 2](#)), lauscht per Default auf Port 1880. Ändern Sie diesen Port nur bei Konflikten mit anderen Anwendungen.

■ debugMaxLength

Die Angabe beeinflusst die maximale Länge der Ausgabe des *debug*-Nodes. Erhöhen Sie den Standardwert von 1000 Ihren jeweiligen Erfordernissen entsprechend.

■ `credentialSecret`

Unter *Credentials* versteht man die Angaben von Zugangsdaten zu anderen Anwendungen, also zumeist Benutzername und Passwort. Einige Nodes benötigen in ihren Eigenschaften derartige Angaben. Darauf bezieht sich `credentialSecret`. Der Eintrag ist zunächst auskommentiert, sodass ein vom System generierter Schlüssel zum Einsatz kommt.

Entfernen Sie die Kommentarzeichen und setzen Sie den Wert auf `false`, wenn Sie auf eine Verschlüsselung grundsätzlich verzichten möchten.

Möchten Sie den Schlüssel festlegen, entfernen Sie die Kommentarzeichen und setzen einen Wert Ihrer Wahl in die Anführungszeichen. Vermeiden Sie jedoch eine nochmalige Änderung. Dies hätte zur Folge, dass Node-RED nicht mehr in der Lage wäre, die Zugangsdaten zu entschlüsseln, was dazu führt, dass diese verloren sind.

Die Änderungen werden erst nach einem Neustart von Node-RED wirksam. Die Warnmeldung aus [Abbildung 1.12](#) ist dann verschwunden.

■ `logging`

Hier legen Sie die Art und den Umfang der Konsolenausgaben fest.

Als Log-Levels können Sie vorgeben:

- `fatal` – gibt nur Fehler aus, die die Anwendung unausführbar machen.
- `error` – gibt die Fehler aus, die für eine bestimmte Anfrage als fatal angesehen werden.
- `warn` – gibt Warnmeldungen aus.
- `info` – gibt generelle Informationen über die Anwendung aus. Dies ist die Voreinstellung.
- `debug` – gibt ausführlichere Informationen als die Option `info` aus und hilft bei der Fehlersuche.
- `trace` – gibt sehr detaillierte Log-Informationen aus. Bei dieser Einstellung werden Sie wahrscheinlich mit Infos überflutet.
- `off` – gibt keine Log-Ausgaben aus, schaltet die Funktion also ab.

Abgesehen vom Level `off` beinhalten die Ausgaben eines Levels alle Ausgaben des höheren Levels, z. B. beinhaltet `info` auch alle Meldungen der Stufen `warn`, `error` und `fatal`.

■ `metrics`

Die Option `metrics` bezieht sich auf die Ausgabe von Informationen zur Ausführung von Flows und zur Speicherplatzbelegung. (Die Ausgabe bei `metrics: true` sehen Sie in [Abbildung 1.13](#).)

```

10 Apr 07:14:53 - [metric] {"level":99,"nodeid":"2824e11e.f2d94e","event":"node.change.done","msgid":"8a900013.5367d","timestamp":1618031693713}
10 Apr 07:14:53 - [metric] {"level":99,"nodeid":"43a59514.9737dc","event":"node.debug.receive","msgid":"8a900013.5367d","timestamp":1618031693724}
10 Apr 07:14:53 - [metric] {"level":99,"nodeid":"43a59514.9737dc","event":"node.debug.done","msgid":"8a900013.5367d","timestamp":1618031693726}
10 Apr 07:14:59 - [metric] {"level":99,"event":"runtime.memory.rss","value":89423872,"timestamp":1618031699417}
10 Apr 07:14:59 - [metric] {"level":99,"event":"runtime.memory.heapTotal","value":52801536,"timestamp":1618031699418}
10 Apr 07:14:59 - [metric] {"level":99,"event":"runtime.memory.heapUsed","value":47656388,"timestamp":1618031699418}

```

Abbildung 1.13: Logging mit der Einstellung »metrics : true«

■ audit

Die Einstellung `audit : true` bewirkt, dass bei Ereignissen eine Ausgabe an die HTTP-API erfolgt (siehe [Abbildung 1.14](#)).

```

10 Apr 07:28:39 - [audit] {"event":"library.get","library":"local","type":"flows","path":"","level":98,"timestamp":1618032519700}
10 Apr 07:28:39 - [audit] {"event":"library.get","library":"_examples_","type":"flows","path":"","level":98,"timestamp":1618032519713}
10 Apr 07:28:50 - [audit] {"event":"flows.set","type":"flows","level":98,"path":"/flows","ip":"192.168.178.21","timestamp":1618032530667}

```

Abbildung 1.14: Logging mit der Einstellung »audit : true«

Logging in eine Datei umleiten und Log-Datei bereinigen



Manchmal ist es wertvoll, die Log-Ausgaben in einer Datei zu sammeln. Verwenden Sie hierzu das Kommando `<node-red-start-kommando> > /home/.pi/nr-log.log`. Ersetzen Sie dabei `<node-red-start-kommando>` durch das Startkommando, also `node-red`.

Die Log-Datei halten Sie mit dem Kommando

```
sudo tail -n 20 /home/.pi/nrlog.log | sudo tee /home/.pi/nrlog.log
```

im Zaum. Automatisieren Sie den Bereinigungsverfahren bei Bedarf mit einem *inject*- und einem *exec*-Node (siehe [Kapitel 3](#), »Das Fundament: die Basics von Node-RED«, und [Kapitel 9](#), »Node-RED-Hacks«).

1.6 Node-RED absichern

Nach der Installation ist Node-RED nicht gesichert und steht für Zugriffsversuche völlig offen. Das heißt: Alle, die Zugang zu Ihrem Netzwerk haben, können auf den Node-RED-Editor zugreifen und so Flows verändern. Dies ist eine nicht zu unterschätzende Gefahr für Ihre Systeme, denn spezielle Nodes (z. B. der *function*-Node) sind so mächtig, dass über sie ein Zugriff auf den gesamten Rechner und darüber hinaus möglich ist.

Das Gleiche gilt für den Datentransfer zwischen dem Browser und dem Node-RED-Server. Er ist unverschlüsselt und mit entsprechenden Kenntnissen und Werkzeugen wie Wire-shark oder Ettercap einfach lesbar.

Sichern Sie also Ihre Installation ab! Spätestens sobald Sie einen Zugang von außerhalb einrichten, ist das sehr wichtig.

1.6.1 Passwortschutz für den Node-RED-Editor

Node-RED unterstützt zwei Möglichkeiten zur Authentifikation:

- Zugangsschutz über Benutzernamen und Passwort
- Zugang über einen OAuth-/OpenID-Provider wie Google oder GitHub

An dieser Stelle konzentriere ich mich auf den Zugangsschutz über Benutzernamen und Passwort, denn er ist einfacher einzurichten und nicht abhängig von anderen Diensten. Dabei wird das zu verwendende Passwort als Hash in der Konfigurationsdatei *settings.js* abgelegt. Richten Sie den Passwortschutz wie folgt ein:

Schritt 1: Admin-CLI-Tool herunterladen

Für das Generieren des Passwort-Hashs ist das Kommandozeilenwerkzeug `node-red-admin` notwendig. In der Regel müssen Sie es nachinstallieren:

```
sudo npm install -g node-red-admin
```

Bei der Installation kann es zu einigen Warn- und Fehlerhinweisen kommen, die Sie aber ignorieren können.

Schritt 2: Passwort-Hash erzeugen

Erzeugen Sie jetzt den Passwort-Hash:

```
sudo node-red-admin hash-pw
```

Das Tool erfragt ein Passwort (z. B. »raspberrry«) und gibt den entsprechenden Hash aus (siehe [Abbildung 1.15](#)).

```
pi@pi4:~$ sudo node-red-admin hash-pw
Password:
$2a$08$1G0ift3/vyTCnbXBQPD0uuGhKNJ0DzLUJsX3uQanCy4C2/mgV5utu
pi@pi4:~$ █
```

Abbildung 1.15: Passwort-Hash für Node-RED

Kopieren Sie den Hash zunächst einmal in eine leere Textdatei.

Schritt 3: Konfigurationsdatei anpassen

Öffnen Sie die Datei *settings.js* mit einem Editor (z. B. `nano /home/pi/.node-red/settings.js`). Suchen Sie den bereits vorgegebenen, aber auskommentierten Bereich `//SecuringNode-Red`. Entfernen Sie die Kommentarzeichen und ergänzen Sie den Wert des Schlüssels `password` wie folgt durch den generierten Passwort-Hash:

```
adminAuth: {
  type: "credentials",
```

```

users: [{
  username: "admin",
  password: "<der-generierte-Hash>",
  permissions: "*"
}]
},

```

Der * bei permissions besagt, dass der Benutzer admin alle Berechtigungen haben soll. Wenn Sie einen weiteren Nutzer anlegen und ihm hier den Eintrag `read` zuweisen, erlauben Sie ihm nur lesenden Zugriff.

Schritt 4: Node-RED neu starten

Nach einem Neustart erscheint nach dem Browseraufruf des Node-RED-Editors eine Login-Maske, in der Sie sich mit dem erstellten Benutzer authentifizieren müssen (siehe [Abbildung 1.16](#)). Dazu brauchen Sie das entsprechende Passwort, nicht den Hash!

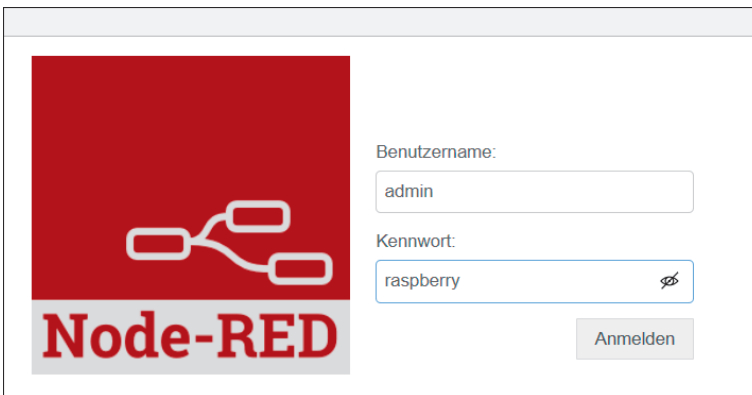


Abbildung 1.16: Browser-Log-in bei Node-RED

Legen Sie bei Bedarf weitere Benutzer mit eigenem Passwort an und schränken Sie den Vollzugriff ein.

1.6.2 HTTPS aktivieren

In der Grundkonfiguration findet der Datenaustausch zwischen dem Node-RED-Server und dem Browser via HTTP (*Hypertext Transfer Protocol*) statt. Der Datenverkehr erfolgt dabei unverschlüsselt. Es liegt auf der Hand, dass dies massive Sicherheitsprobleme bedeutet, sobald die Verbindung Geräte außerhalb des heimischen Netzwerks betrifft.

Abhilfe schafft HTTPS (*Hypertext Transfer Protocol Secure*). Es ist im technischen Sinne kein eigenständiges Protokoll, sondern steht für die Verwendung von HTTP über TLS/SSL. Damit werden die Daten während des Transports verschlüsselt.

Der Einsatz von TLS/SSL (HTTPS) verfolgt drei Hauptziele:

- Vertraulichkeit (durch Verschlüsselung)
- Datenintegrität (Schutz vor Datenmanipulation)
- Authentizität (Überprüfung, ob die Identität des Seitenbetreibers korrekt ist)

Sie erkennen im Browser eine verschlüsselte Verbindung daran, dass der besuchten Internetadresse *https* statt *http* vorangestellt ist. Außerdem wird wie in [Abbildung 1.17](#) ein kleines Schlosssymbol eingeblendet.

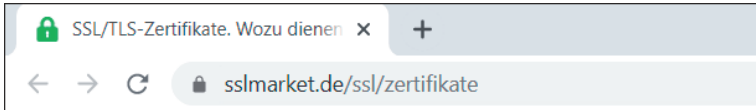


Abbildung 1.17: Mit dem Schloss kennzeichnet der Browser eine HTTPS-Verbindung.

Eine HTTP-Authentifizierung findet in mehreren Schritten statt:

1. Verbindungsaufbau – Der Client kontaktiert den Server.
2. Serverauthentifizierung – Verbindungsannahme durch den Server; der Server schickt sein Zertifikat mit dem öffentlichen Schlüssel seines Schlüsselpaars zur Authentifizierung an den Client.
3. Validierung – Der Client führt eine Gültigkeitsprüfung des Serverzertifikats durch und bricht bei Ungültigkeit des Zertifikats die Verbindung ab.
4. Schlüsselerzeugung – Der Client erzeugt bei Gültigkeit des Zertifikats den symmetrischen Sessionschlüssel.
5. Client Key Exchange – Mit dem öffentlichen Schlüssel des Servers verschlüsselt der Client den Sitzungsschlüssel (Sessionschlüssel) und schickt ihn an den Server.
6. Mit seinem privaten Schlüssel entschlüsselt der Server den verschlüsselten Sitzungsschlüssel.
7. Der Server bestätigt den geheimen Sitzungsschlüssel.
8. Austausch verschlüsselter HTTP-Requests und -Responses bis zum Verbindungsende (siehe [Abbildung 1.18](#))

Sie benötigen also für den Einsatz von HTTPS ein *Verschlüsselungszertifikat*.

Verschlüsselungszertifikate nutzen als Verschlüsselungstechnik ein asymmetrisches Verfahren (*Public-Key-Verfahren*). Dabei erzeugt ein Benutzer ein Schlüsselpaar, das aus einem geheimen (*private*) und einem öffentlichen (*public*) Schlüssel (*key*) besteht. Der öffentliche Schlüssel kann auch öffentlich ausgetauscht werden. Er ermöglicht jedem, Daten für den Inhaber des privaten Schlüssels zu verschlüsseln. Nur der Besitzer des geheimen (privaten) Schlüssels ist in der Lage, diese Daten wieder zu entschlüsseln.

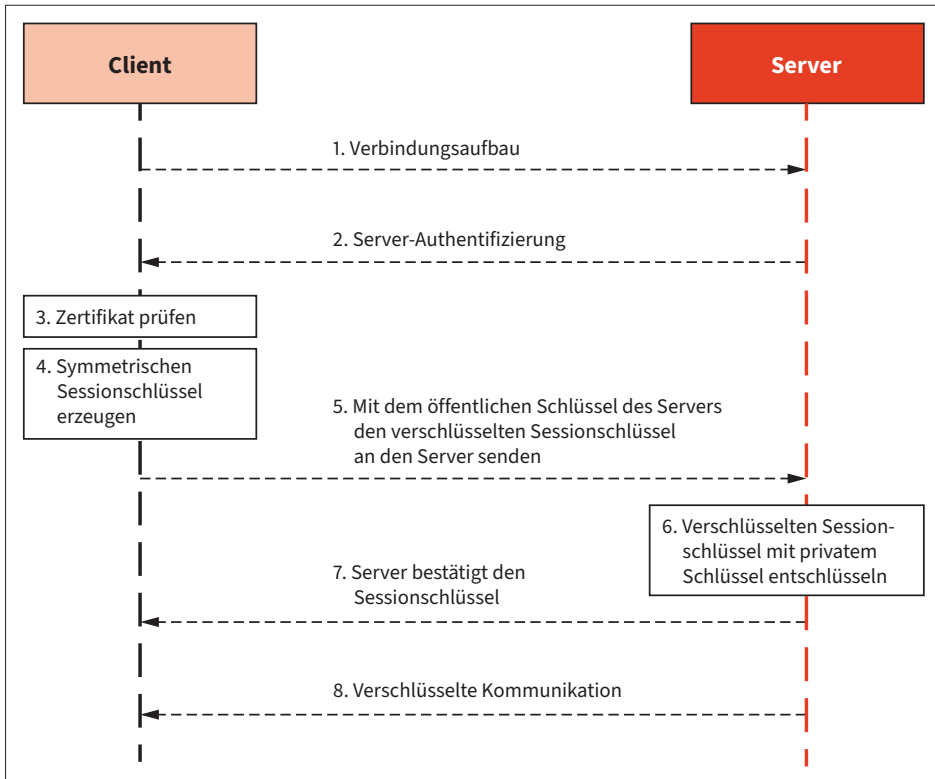


Abbildung 1.18: Die Schritte bei der HTTP-Authentifizierung

Das Erstellen vollzieht sich in drei Schritten, die ich kurz skizziere, bevor es dann an die Umsetzung geht:

Schritt 1: Einen RSA-Schlüssel erzeugen

Erzeugen Sie einen *RSA-Schlüssel*. (RSA steht für die Anfangsbuchstaben der Familiennamen der Entwickler dieses Verschlüsselungsverfahrens: Rivest, Dhamir, Adleman.)

Schritt 2: CSR erstellen

Erstellen Sie eine Zertifikatsregistrierungsanforderung (engl. *Certificate Signing Request*, CSR). Der CSR enthält Firmeninformationen, den Domännennamen (*Common Name*, CN), Ihre E-Mail-Adresse und weitere Informationen. Die Angabe des CN ist wichtig, wenn das Zertifikat für eine bestimmte Domain erstellt wird.

Schritt 3: Zertifikat signieren

Signieren Sie das Zertifikat. So wird »bewiesen«, dass es sich um ein echtes und vertrauenswürdiges Zertifikat handelt.

Es gibt drei unterschiedliche Arten von signierten Zertifikaten:

- **Öffentlich signierte Zertifikate** – Das sind offiziell gültige Zertifikate, die über eine öffentliche Zertifizierungsstelle (engl. *Certification Authority*, CA) signiert wurden. Um ein solches Zertifikat zu erhalten, müssen Sie eine Gebühr an eine entsprechende Stelle zahlen. Für den Privatgebrauch ist das unnötig.
- **Privat signierte Zertifikate** – Das Zertifikat wird über eine private (nicht öffentliche) CA signiert. Solche Dienste werden von großen Firmen oder Organisationen gepflegt. Für Sie wird dies sehr wahrscheinlich ebenfalls keine Option darstellen.
- **Selbst signierte Zertifikate** – Diese werden durch den eigenen privaten Schlüssel signiert, nicht über eine CA. Diese Zertifikate können Sie selbst ausstellen.

Für die Stärke bzw. Sicherheit der Verschlüsselung ist das gewählte Verfahren absolut irrelevant. Da jedoch bei selbst signierten Zertifikaten die Authentizität des Ausstellers nicht geprüft werden kann, sehen Besucher einer derart abgesicherten Seite eine Browserwarnung, die das Zertifikat als nicht vertrauenswürdig einstuft – Sie bezeugen ja quasi nur selbst, dass das von Ihnen erstellte Zertifikat in Ordnung ist. Deshalb eignet sich ein selbst signiertes SSL-Zertifikat nur für private Webseiten bzw. zur Absicherung von Tools, auf die nur eine begrenzte Anzahl von Benutzern zugreift.

Ein Zertifikat, das von einer offiziellen Zertifizierungsstelle signiert wurde, genießt den vollen Vertrauensschutz und wird in der Regel von allen gängigen Programmen (z. B. Browsern) ohne Probleme akzeptiert, da diese ein übergeordnetes Zertifikat der Zertifizierungsstelle enthalten.

Privat signierte Zertifikate werden dagegen als vertrauenswürdig akzeptiert, wenn das Zertifikat der verwendeten Zertifizierungsstelle auf dem Client-Rechner (also im Browser) installiert ist.

Nach diesem Theorieeinschub sollte klar sein, dass für die meisten Einsatzszenarien ein selbst signiertes Zertifikat ausreicht. Damit werden die Daten sicher verschlüsselt. Dass das nur durch Sie selbst garantiert wird, sollte kein Problem darstellen.

Ein selbst signiertes Zertifikat lässt sich mit dem Kommandozeilenprogramm `openssl` leicht erstellen. Das Paket ist bei den meisten Linux-Distributionen bereits installiert, so auch auf dem Raspberry Pi. Erstellen Sie gegebenenfalls einen eigenen Ordner für Ihre generierten Zertifikate (z. B. mit `mkdir ssl-zertifikate`) und führen Sie folgende Anweisung aus:

```
openssl req -new -days 3650 -newkey rsa:4096bits -sha512 -x509
  -nodes -out nodeRED.crt -keyout nodeRED.key
  -subj "/CN=localhost"
```

Die Anweisung (siehe [Abbildung 1.19](#)) erstellt nach dem Standard x509 ein Schlüsselpaar, das 3.650 Tage Gültigkeit hat und mit 4.096 Bits verschlüsselt ist. Die Ausgabe erfolgt in

die Dateien `nodeRED.crt` (Zertifikat) und `nodeRED.key` (Schlüssel). `CN=localhost` bestimmt als Domänenname den lokalen Rechner.

```
pi@raspberrypi:~ $ mkdir ssl-zertifikate
pi@raspberrypi:~ $ cd ssl-zertifikate
pi@raspberrypi:~/ssl-zertifikate $ openssl req -new -days 3650 -newkey rsa:4096bits -sha512
-x509 -nodes -out nodeRED.crt -keyout nodeRED.key -subj "/CN=localhost"
Generating a RSA private key
.....+++++
.....+++++
writing new private key to 'nodeRED.key'
-----
pi@raspberrypi:~/ssl-zertifikate $ █
```

Abbildung 1.19: Zertifikate und Schlüssel mit openssl generieren

SSL-Zertifikate unter Windows

Unter Windows ist es leider deutlich schwieriger, SSL-Zertifikate zu erstellen und selbst zu signieren. Am besten installieren Sie OpenSSL für Windows. Eine entsprechende Installationsdatei finden Sie unter:

<https://slproweb.com/products/Win32OpenSSL.html>

Starten Sie dann den Win64-OpenSSL-Command-Prompt, den Sie im Startmenü finden. Dort können Sie dann den Befehl ausführen, die Fragen beantworten und ein Zertifikat erstellen.

Eine Alternative – auch für Linux und andere Systeme – ist der Dienst *Let's Encrypt*:

<https://letsencrypt.org>

Er richtet sich jedoch eher an Betreiber von Webseiten und erfordert ein wenig Konfigurationsaufwand.

Haben Sie Ihr Zertifikat erstellt und signiert, öffnen Sie die Node-RED-Konfigurationsdatei, z. B. über `nano /home/pi/.node-red/settings.js`. Suchen Sie den Bereich `///https object`, aktivieren Sie das JSON-Objekt `https` und passen Sie die Pfade Ihren Dateien an:

```
https: {
  key: require("fs").readFileSync('/home/pi/ssl-zertifikate/nodeRED.key'),
  cert: require("fs").readFileSync('/home/pi/ssl-zertifikate/nodeRED.crt')
},
```

Starten Sie Node-RED dann neu und überprüfen Sie mit `node-red-log`, ob Node-RED die Änderungen akzeptiert hat. Eine Meldung wie die in [Abbildung 1.20](#) signalisiert, dass nunmehr HTTPS aktiviert ist.



```
-----
22 Dec 10:17:44 - [info] Flows starten
22 Dec 10:17:44 - [info] Flows gestartet
22 Dec 10:17:44 - [info] Server wird jetzt auf https://127.0.0.1:1880/ ausgeführt.
█
```

Abbildung 1.20: Dieser Node-RED-Server nutzt HTTPS.

Node-RED ist ein Profi-Tool. Deswegen bietet es noch einige weitere Funktionalitäten, den Zugang für Dritte zu gestalten (z. B. über so genannte *Custom authentication token*). Weiterführende Informationen finden Sie unter <https://nodered.org/docs/user-guide/runtime/securing-node-red>.

1.6.3 Sicherheit für den http-in-Node

Mit dem *http-in*-Node kann auf Ihr System zugegriffen werden. Ein Beispiel dazu finden Sie in [Kapitel 3, »Das Fundament: die Basics von Node-RED«](#). Diesen Zugang sollten Sie absichern. Suchen Sie in der Datei *settings.js* den Eintrag `httpNodeAuth`.

Entfernen Sie die Kommentarzeichen und ergänzen Sie diesen Eintrag:

```
httpNodeAuth: {user:"admin",pass:"<der generierte Hash>"},
```

Der Wert von `pass` ist der Passwort-Hash, wie Sie ihn mit Schritt 2 in [Abschnitt 1.6.1](#) generiert haben. Ein Seitenaufruf ist dann nur mit gültigen Anmeldedaten möglich (siehe [Abbildung 1.21](#)).

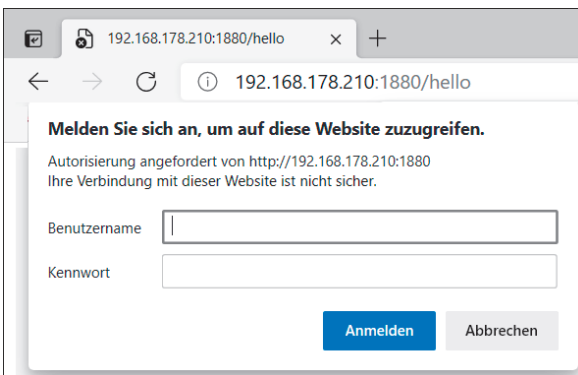


Abbildung 1.21: Sichern Sie den http-in-Node durch eine Anmeldung ab.

1.6.4 Die Konfigurationsdatei absichern

Das Installationsskript weist darauf hin, dass der Eigentümer der Konfigurationsdatei `~/.node-red/settings.js` auf den User *root* geändert werden sollte, um sie vor unberechtigtem Zugriff zu schützen. Dankenswerterweise erledigt das Skript diesen Schritt gleich mit:

```
sudo chown root:root ~/.node-red/settings.js
```

1.7 Node-RED Projekte

Node-RED Projekte ist ein professioneller Ansatz für die Entwicklung von Flows innerhalb von Node-RED. Es bietet Ihnen einen vernetzten Dienst zur Versionsverwaltung für Softwareentwicklungsprojekte und hilft Ihnen beim Erstellen einer Node-RED-Anwendung, die Sie weiterverteilen können und an der Sie auch mit anderen Makern arbeiten können.

Die Basis ist ein Git-Repository, d. h., alle Dateien sind vollständig versioniert. Sie teilen Ihre Arbeit in einzelne Schritte ein, die sie nach der Fertigstellung *committen*, d. h. im Repository abspeichern. So können Sie stets zu diesen Meilensteinen zurückkehren und Änderungen rückgängig machen. Zudem können Sie von unterschiedlichen Rechnern oder mit mehreren Leuten gleichzeitig an einem Projekt arbeiten und den Fortschritt in einzelne Commits gliedern. Entwicklerteams können ihre eingespielten Workflows für die gemeinsame Arbeit an einem Projekt weiterbetreiben.

Node-RED Projekte richtet sich an erfahrene Maker und an Entwicklerteams, die bereits einige Kenntnisse im professionellen Software-Engineering mitbringen. Ich skizziere die notwendigen Schritte daher nur grob – passen Sie sie an Ihre eigenen Anforderungen an.

Schritt 1: Vorbereitungen

Aktivieren Sie zunächst das Projekte-Feature in der Konfigurationsdatei *settings.js*. Navigieren Sie darin zum Bereich `//Customising the editor` und setzen Sie `projects: {enabled: true}`:

```
// Customising the editor
editorTheme: {
  projects: {
    // To enable the Projects feature, set this value to true
    enabled: true
  }
}
```

Starten Sie dann Node-RED neu.

Node-RED Projekte verlangt, dass `git` und `ssh-keygen` als Kommandozeilentool verfügbar sind. Beim Raspberry Pi und vielen Linux-Distributionen ist das bereits direkt nach der Installation der Fall. Sind diese Tools nicht vorhanden, erhalten Sie von Node-RED eine entsprechende Fehlermeldung.

Schritt 2: Ein erstes Projekt im Node-RED-Editor erstellen

Beim ersten Browserzugriff nach dem Neustart begrüßt der Node-RED-Editor Sie wie in [Abbildung 1.22](#) mit der Einladung, ein neues Projekt zu erstellen bzw. ein Repository zu klonen.

Klicken Sie für Ihr erstes Projekt auf **Projekt erstellen**. Im nächsten Schritt erfolgt die Konfiguration des Projekts (siehe [Abbildung 1.23](#)).

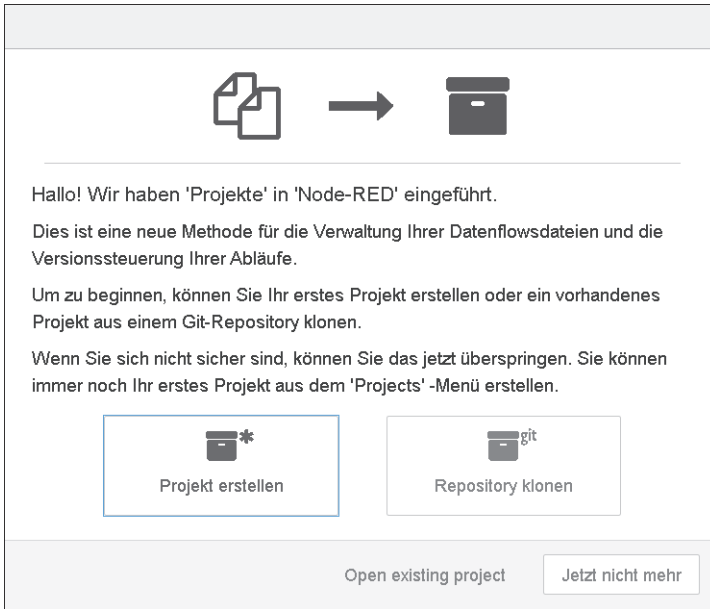


Abbildung 1.22: »Node-RED Projekte« – Projekt erstellen

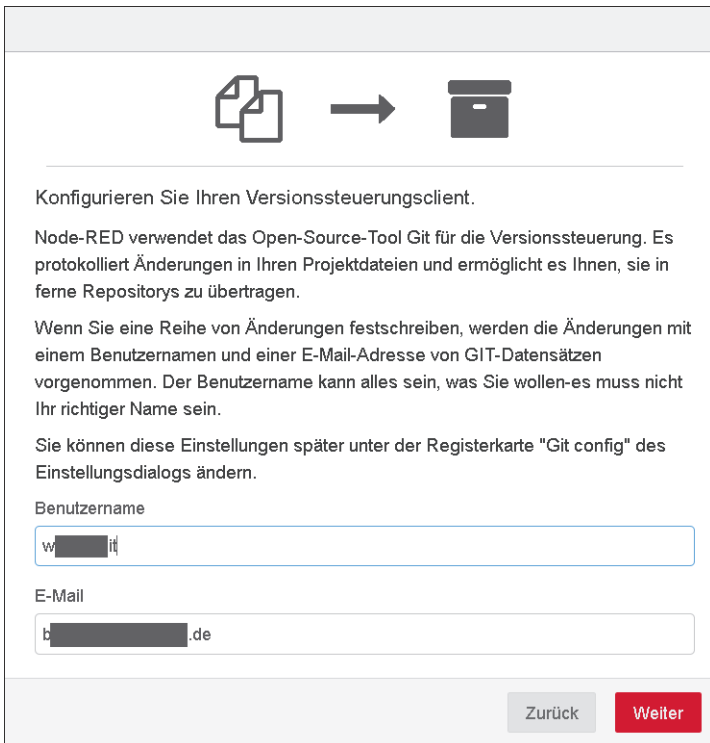


Abbildung 1.23: »Node-RED Projekte« – Projekt konfigurieren

Mit **Weiter** geht es zu dem Schritt, in dem Sie das Projekt beschreiben können. Danach gelangen Sie zur Option, Ihre Projektdateien festzulegen. Ändern Sie gegebenenfalls die Dateinamen und gehen Sie mit **Weiter** zum nächsten Schritt (siehe [Abbildung 1.24](#)).

Setup der Verschlüsselung Ihrer Berechtigungsnachweisdatei

Die Datei mit den Datenflowberechtigungen wird derzeit mit einem vom System generierten Schlüssel verschlüsselt. Sie sollten einen neuen geheimen Schlüssel für dieses Projekt angeben.

Der Schlüssel wird separat von den Projektdateien gespeichert. Sie müssen den Schlüssel angeben, damit dieses Projekt in einer anderen Instanz von Node-RED verwendet werden kann.

Berechtigungsnachweis

Verschlüsselung aktivieren

Verschlüsselung inaktivieren

Vorhandenen Schlüssel kopieren

Angepasster Schlüssel verwenden

Zurück **Projekt erstellen**

Abbildung 1.24: »Node-RED Projekte« – Verschlüsselung der Berechtigungsdatei

Wählen Sie die für Sie zutreffende Konfiguration. Näheres finden Sie in [Abschnitt 1.5.1](#). **Projekt erstellen** schließt den Vorgang mit einer Vollzugsmeldung ab. Wählen Sie dort **Fertig** aus, und Sie gelangen wieder zur regulären Editor-Ansicht. Über das Info-Icon gelangen Sie zu den Projekteinstellungen.

In der rechten Seitenleiste ist nun ein neues Icon namens **Verlauf** vorhanden. Sobald Sie Änderungen bereitstellen, erhält die Rubrik **Lokale Änderungen** einen Eintrag wie in [Abbildung 1.25](#).

Jede Änderung, die Sie nun vornehmen, schlägt sich in Ihrem lokalen Repository nieder. Mit einem Klick auf + bereiten Sie die Änderungen zum Festschreiben (*Committen*) vor (siehe [Abbildung 1.26](#)). Damit werden Ihre Änderungen in das Git-Repository übermittelt.

Ein Klick auf **Festschreiben** leitet den Überförungsprozess (Commit) in Ihr lokales Git-Repository ein (siehe [Abbildung 1.27](#)).

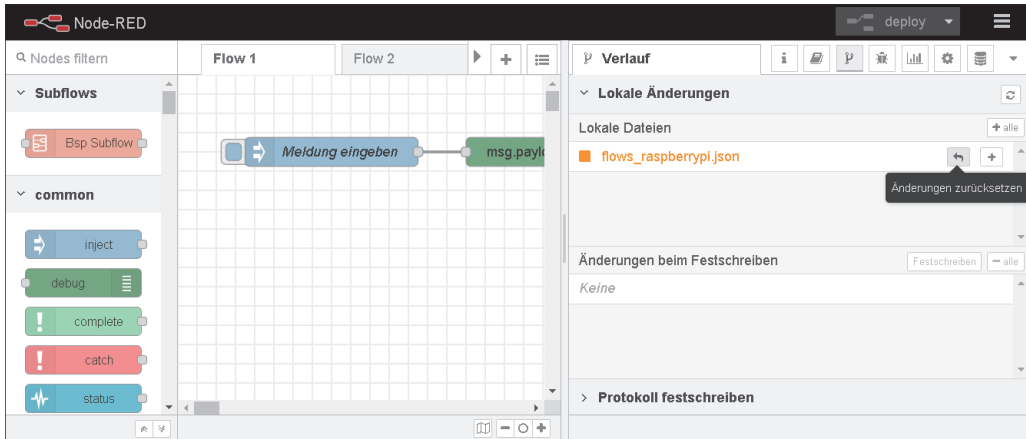


Abbildung 1.25: »Node-RED Projekte« – das Verlauf-Icon und seine Registerkarte

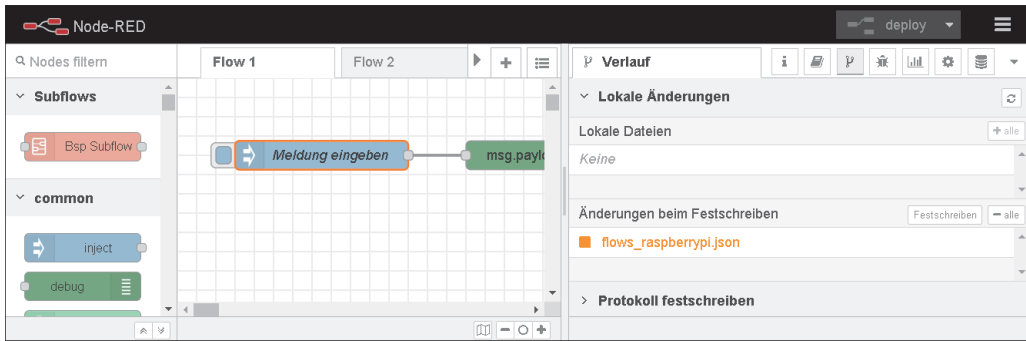


Abbildung 1.26: »Node-RED Projekte« – Änderungen zum Festschreiben

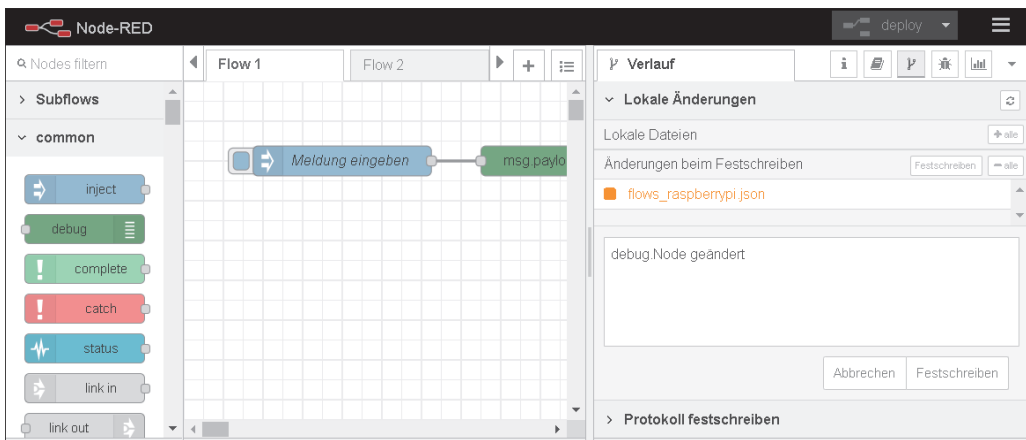


Abbildung 1.27: »Node-RED Projekte« – Angaben überführen mit »Festschreiben«