

# IoT & RPI & Arduino & Node-RED

- Was will ich machen?
- Projektbausteine
- Raspberry Pi 32-Bit (bullseye)
- Node-RED
- Palette erweitern
- Arduino IDE
- Sensoren aus ALLNET 4 Kit
- Schaltung 1: Taster
- Arduino\_Schritt1.ino
- Node-RED Flow 1
- Node-RED Dashboard 1
- Schaltung 2: 2 Taster
- Arduino\_Schritt2.ino
- Node-RED Flow 2
- Node-RED Dashboard 2
- Schaltung 3: LDR
- Arduino\_Schritt3.ino
- Node-RED Flow 3
- Node-RED Dashboard 3
- Schaltung 4: LED einschalten
- Arduino\_Schritt4.ino
- Node-RED Flow 4
- Node-RED Dashboard 4
- Weitere Details

## Was will ich machen?

- *Raspberry Pi* mit Arduino über USB-Schnittstelle verbunden.
- Arduino-IDE auf dem *Raspberry Pi* installieren.
- Node-RED auf dem *Raspberry Pi* installieren.
- Schaltung mit Steckbrett und Arduino aufbauen.
- Sketch mit Arduino-IDE erstellen. Monitor-Test.
- Sketch vom *Raspberry Pi* auf den Arduino übertragen.
- Getesteten Sketch belassen. Node-RED Test.
- Node-RED für Verwendung mit Arduino vorbereiten.
- - Nodes um serielle Schnittstelle erweitern
- Serielle Verbindung von Node-RED zum Arduino einrichten.
- Node-RED-Flow zur Steuerung des Arduino schreiben.

---

# Projektbausteine

Hardware	<ul style="list-style-type: none"><li>• Raspberry Pi mit Betriebssystem „Debian Version: 11 “</li><li>• Arduino-UNO</li></ul>
	<ul style="list-style-type: none"><li>• Steckbrett</li><li>• 1 Taster</li><li>• ...</li><li>• Steckbrett-Kabel</li></ul>
Software	<ul style="list-style-type: none"><li>• Arduino-IDE</li><li>• Node-RED</li></ul>
Sketch	<ul style="list-style-type: none"><li>• Arduino-Sketch zur Programmierung des Arduino</li></ul>
Flow	<ul style="list-style-type: none"><li>• Flow (Node-RED) zur Steuerung mit Node-RED</li></ul>

---

# Raspberry Pi 32-Bit (bullseye)

Raspberry Pi OS

Raspbian Betriebssystem herunterladen.

Windows 10 PC

PC zum erstellen einer SD-Karte mit OS.

Imager herunterladen

[https://downloads.raspberrypi.org/imager/imager\\_latest.exe](https://downloads.raspberrypi.org/imager/imager_latest.exe)

Windows 10

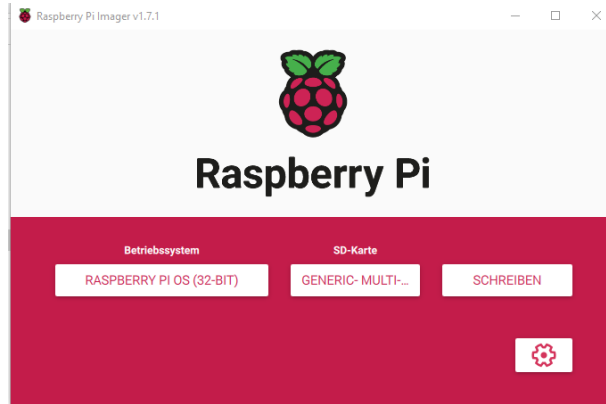
SD-Karte in SD-Card-Slot bereitstellen

January 28th 2022

System: 32-bit Kernel version: 5.10

Debian version: 11 (bullseye)

- Raspberry Pi OS(32-bit) wählen.
- SD-Karte auswählen.
- ...



RPI

Die SD-Karte in den RPI einstecken Schritte der Installation durchführen.

# Node-RED

Anleitung

<https://nodered.org/docs/getting-started/raspberrypi>

LXTerminal:

Prompt-Zeichen „\$“ nicht mit eintippen!

LXTerminal:

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

LXTerminal

Anzeige:

```
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.js 14 LTS                       ✓   v14.19.0   Npm 6.14.16
Clean npm cache                             ✓
Install Node-RED core                       ✓   2.2.2
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
```

# Node-RED

LXTerminal

Nach erfolgter erfolgreicher Installation:

```
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 : So 6. Mär 18:26:03 CET 2022  
Finished: So 6. Mär 18:34:38 CET 2022  
  
You may want to run node-red admin init  
to configure your initial options and settings.
```

Erster Start:

```
$ node-red-start
```

```
Start Node-RED  
  
Once Node-RED has started, point a browser at http://192.168.178.54:1880  
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
```

# Node-RED

LXTerminal

Weitere Anzeige:

```
Starting as a systemd service.
6 Mar 18:40:09 - [info]
Willkommen bei Node-RED!
=====
6 Mar 18:40:09 - [info] Node-RED Version: v2.2.2
6 Mar 18:40:09 - [info] Node.js Version: v14.19.0
6 Mar 18:40:09 - [info] Linux 5.10.92-v8+ arm64 LE
6 Mar 18:40:11 - [info] Paletten-Nodes werden geladen
6 Mar 18:40:15 - [info] Einstellungsdatei: /home/pi/.node-red/settings.js
6 Mar 18:40:15 - [info] Kontextspeicher: default [module=memory]
6 Mar 18:40:15 - [info] Benutzerverzeichnis: /home/pi/.node-red
6 Mar 18:40:15 - [warn] Projekte deaktiviert: editorTheme.projects.enabled=false
6 Mar 18:40:15 - [info] Flow-Datei: /home/pi/.node-red/flows.json
6 Mar 18:40:15 - [info] Neue 'flow'-Datei wird erstellt
6 Mar 18:40:15 - [warn]
-----
6 Mar 18:40:15 - [info] Server wird jetzt auf http://127.0.0.1:1880/ ausgeführt
6 Mar 18:40:15 - [info] Flows werden gestartet
6 Mar 18:40:15 - [info] Flows sind gestartet
```

# Node-RED

Raspberry Pi mit „bullseye“

Automatischen Start einrichten      LXTerminal: `$ sudo systemctl enable nodered.service`

Service beenden      LXTerminal: `$ sudo systemctl disable nodered.service`

Node-RED starten      LXTerminal: `$ node-red-start` (bei automatischem Start nicht erforderlich)

Node-RED beenden      LXTerminal: `$ node-red-stop`

Node-RED auf RPI (lokal)      Browser: `http://127.0.0.1:1880`

IP vom RPI?      LXTerminal: `$ hostname -I`      **ergibt bei mir „192.168.178.xxx“**

Node-RED von PC      Browser: `http://192.168.178.xxx:1880`



# Node-RED

Windows PC Browser

192.168.178.xx:1880 (Deine RPI-IP:1880)

The screenshot displays the Node-RED web interface accessed via a browser. The browser's address bar shows the URL `192.168.178.54:1880/#flow/4f4e6c2cdc7e7cb8`. The Node-RED interface features a left sidebar with a search bar and two main categories: 'Allgemein' (General) and 'Funktion' (Function). The 'Allgemein' category includes nodes like 'inject', 'debug', 'complete', 'catch', 'status', 'link in', 'link call', 'link out', and 'comment'. The 'Funktion' category includes a 'function' node. The main workspace, titled 'Flow 1', contains a flow diagram with four nodes: 'Enno Klatt' (a red node with a status indicator 'Nicht verbunden'), a 'function' node (orange), and two 'msg' nodes (green), labeled 'msg1' and 'msg2'. The 'function' node is connected to both 'msg1' and 'msg2'. The right sidebar shows the 'Info' panel for the selected 'Flow 1', displaying the flow ID '4f4e6c2cdc7e7cb8'.

# Palette erweitern

Serielle Schnittstelle (USB)  
nachrüsten

> Hamburger-Menü



[node-red-node-serialport \(node\) - Node-RED \(nodered.org\)](#)

> Manage Palette

> Tab Install

> Search modules > „node-red-node-serialport“

> install

[dashboard-evi \(node\) - Node-RED \(nodered.org\)](#)

Dashboard  
nachrüsten

> Search modules > „dashboard-evi“

> install

# Arduino IDE

Der Einstieg

Programmierung mit der an den Arduino angepassten Programmiersprache C++.

Arduino-IDE

Die Arduino-IDE ist auf das notwendigste reduziert. Sie bietet beim Erstellen eines Sketches (Programm) wenig Unterstützung.

Arduino UNO R3



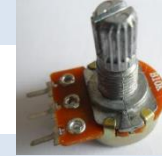
```
Arduino_Sep1
1 // https://www.youtube.com/watch?v=ryEFNylE\_LU
2
3 const byte sensorPin = 7;
4 byte lastState = LOW;
5
6 void setup() {
7     Serial.begin(9600);
8     pinMode(sensorPin, INPUT);
9 }
10
11 void loop() {
12     byte currentState = digitalRead(sensorPin);
13     if (currentState != lastState) {
14         lastState = currentState;
15         Serial.println(currentState);
16     }
17 }
```

# Sensoren aus ALLNET 4 Kit



Taster

Potentiometer 10 kΩ



LDR (Fotowiderstand)

Lageabhängiger Schalter mit Kugel



LM35, Temperatursensor im TO-92 Gehäuse,  
Messbereich: -55°C bis +150°C

Wasser-Sensor (Platine mit Schalttransistor)

Temperatur & Luftfeuchtigkeits-Sensor Typ: DHT11

Audio-Sensor KY-038

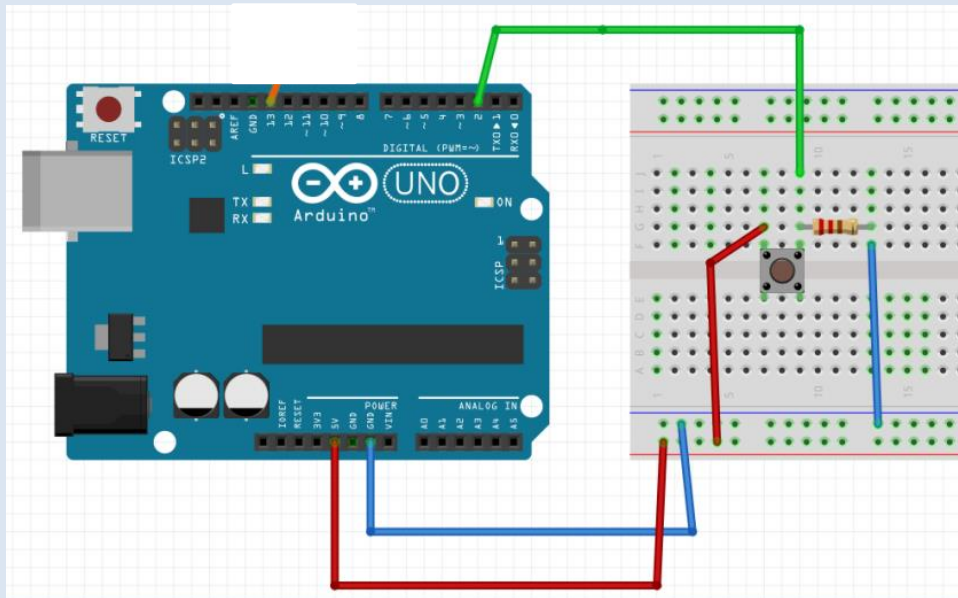


# Schaltung 1: Taster

Aufgabe

Taster gedrückt, dann Status „0“ oder „1“ ausgeben.

Fritzing Schaltung



Taster

Widerstand 10 k $\Omega$

(Im Bild Pin 2)

Aktuell im Sketch Pin 6

---

# Arduino\_Schritt1.ino

---

```
// Arduino_Schritt1.ino
// Aufgabe: Ein Arduino-Taster

const byte sensorPin = 6;
byte lastState = LOW;

void setup() {
  Serial.begin(9600);
  pinMode(sensorPin, INPUT);
}

void loop() {
  byte currentState = digitalRead(sensorPin);
  if (currentState != lastState){
    lastState = currentState;
    Serial.print(currentState);
    Serial.print('\n');
  }
}
```

---

# Node-RED Flow 1

Node-RED öffnen

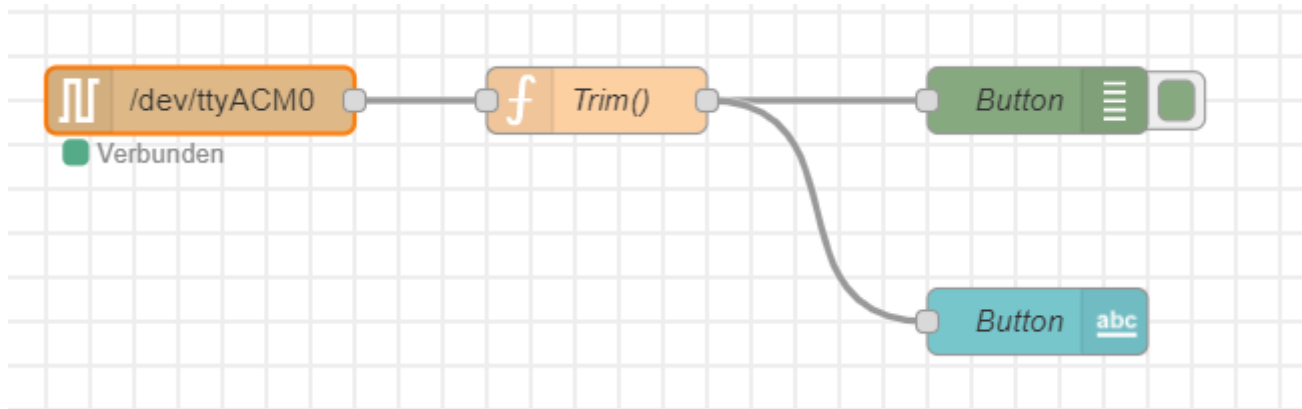
RPI-Browser:

<http://127.0.0.1:1880>

PC-Browser:

<http://192.168.178.xxx:1880>

Flow Arduino-Taster  
(Button)



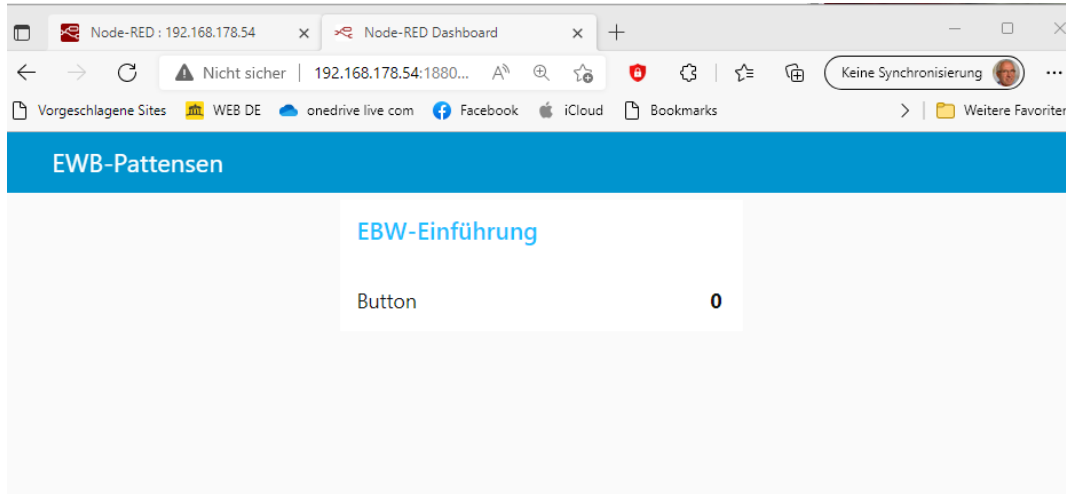
# Node-RED Dashboard 1

Node-RED öffnen

RPI-Browser: <http://127.0.0.1:1880/ui>

PC-Browser: <http://192.168.178.xxx:1880/ui>

Dashboard PC-Browser



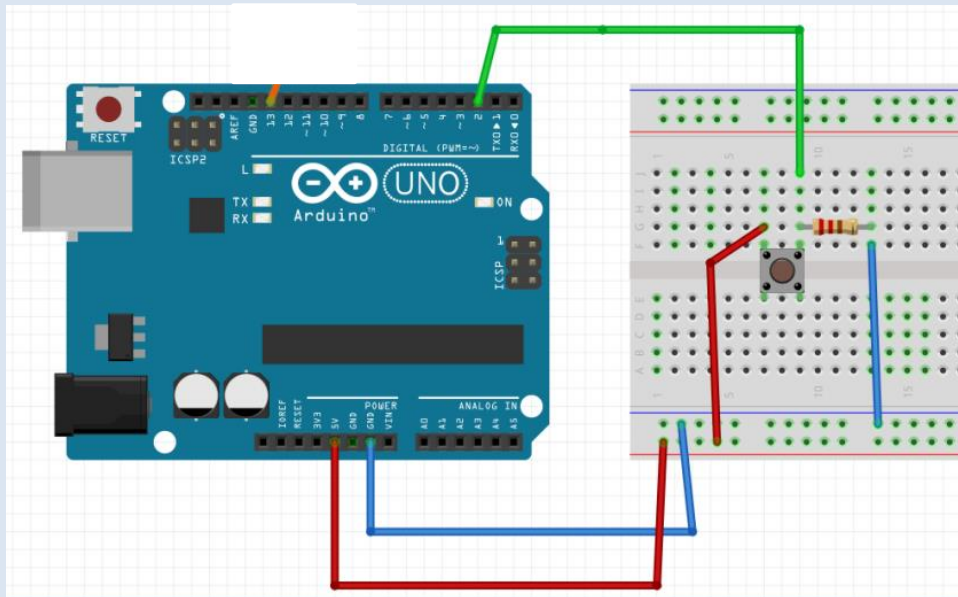


## Schaltung 2: 2 Taster

Aufgabe

Erster Taster gedrückt, dann Status „0“ oder „1“ ausgeben.  
Zweiter Taster gedrückt, dann Status „0“ oder „1“ ausgeben.

Fritzing Schaltung (hier nur 1 Taster)



Taster  
Widerstand 10 kΩ

Erster Taster  
(Im Bild Pin 2)  
Aktuell im Sketch Pin 6

Zweiter Taster  
Pin 7

# Arduino\_Schritt2.ino

```
// Arduino_Schritt2.ino
// Aufgabe: 2. Arduino-Taster

const byte sensorPinBt1 = 6;
const byte sensorPinBt2 = 7;
byte lastStateBt1 = LOW;
byte lastStateBt2 = LOW;

void setup() {
  Serial.begin(9600);
  pinMode(sensorPinBt1, INPUT);
  pinMode(sensorPinBt2, INPUT);
}

void loop() {
  // Button 1
  byte currentState = digitalRead(sensorPinBt1);
  if (currentState != lastStateBt1){
    lastStateBt1 = currentState;
    payload();
  }
  // Button 2
  currentState = digitalRead(sensorPinBt2);
  if (currentState != lastStateBt2){
    lastStateBt2 = currentState;
    payload();
  }
}

void payload() {
  Serial.print("Button_1:");
  Serial.print(lastStateBt1);
  Serial.print(";");
  Serial.print("Button_2:");
  Serial.print(lastStateBt2);
  Serial.print('\n');
}
```

# Node-RED Flow 2

Node-RED öffnen

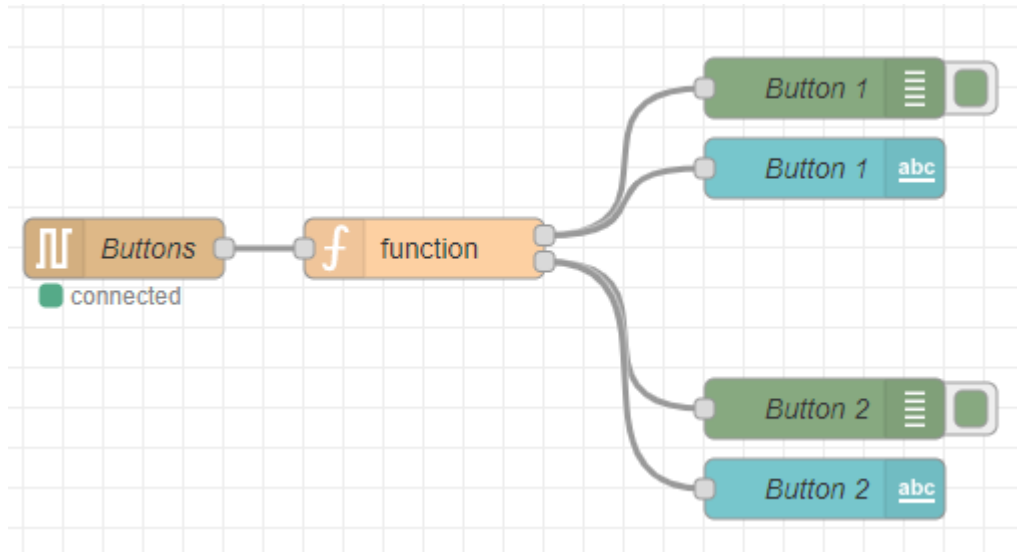
RPI-Browser:

<http://127.0.0.1:1880>

PC-Browser:

<http://192.168.178.xxx:1880>

Flow zwei Arduino-Taster



# Node-RED Dashboard 2

Node-RED öffnen

RPI-Browser: <http://127.0.0.1:1880/ui>

PC-Browser: <http://192.168.178.xxx:1880/ui>

Dashboard PC-Browser

Node-RED Dashboard

Nicht sicher | 192.168.178.54:1880/ui/#!/0?socketid=7...

PortableApps.com Gmail YouTube Maps Sale! - AZ-Delivery

### EWB-Pattensen

#### EBW-Einführung

Button_1	1
Button_2	0

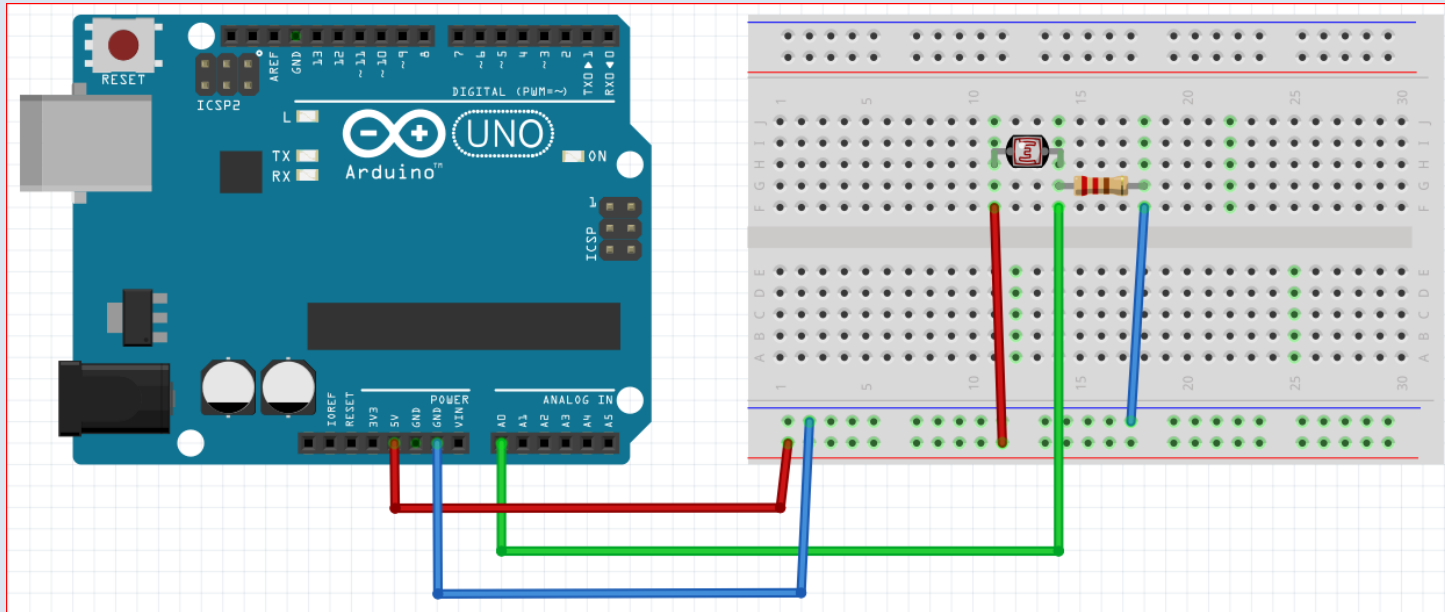
# Schaltung 3: LDR

Aufgabe

Analoge Messwerte des LDR im Web anzeigen

Fritzing Schaltung

LDR; Widerstand 10 kΩ



# Arduino\_Schritt3.ino

```
// Arduino_Schritt3.ino
// Aufgabe: Arduino-Analogeingang

const byte analogPin    = A0;
int lastAnalogLevel = 0;

void setup() {
  Serial.begin(9600);
}

void loop() {

  // Light
  byte currentState = analogRead(analogPin);
  if (currentState != lastAnalogLevel){
    lastAnalogLevel = currentState;
    payload();
  }
}

// creating message for Node-RED
void payload(){
  // LightSensor
  // Serial.print(";");           // delimiter ";"
  // which separates the topics
  Serial.print("Light:");
  Serial.print(lastAnalogLevel);

  Serial.print('\n'); // telling Node-Red end of mesg.
}
```

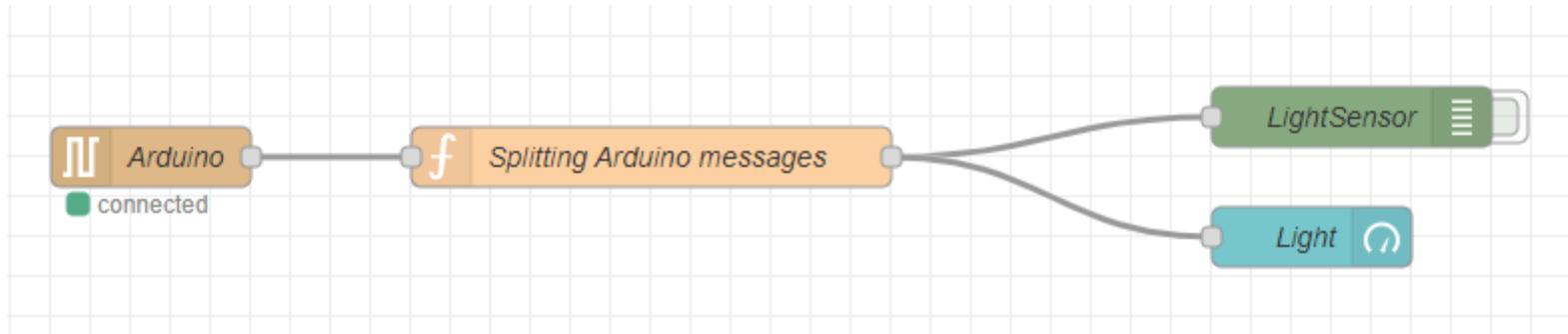
# Node-RED Flow 3

Node-RED öffnen

RPI-Browser: <http://127.0.0.1:1880>

PC-Browser: <http://192.168.178.xxx:1880>

Flow Arduino-LDR



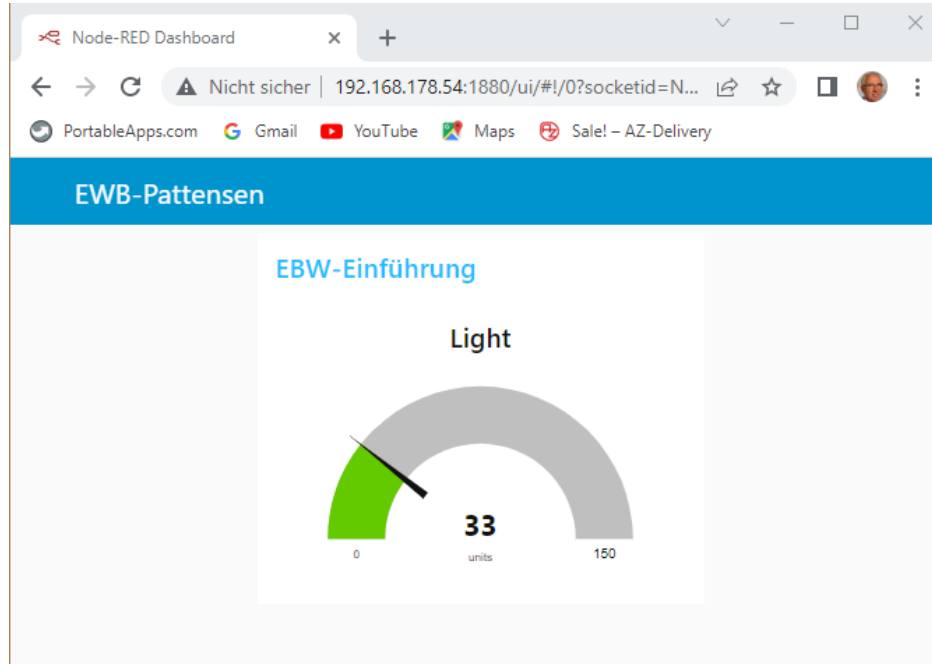
# Node-RED Dashboard 3

Node-RED öffnen

RPI-Browser: <http://127.0.0.1:1880/ui>

PC-Browser: <http://192.168.178.xxx:1880/ui>

Dashboard PC-Browser

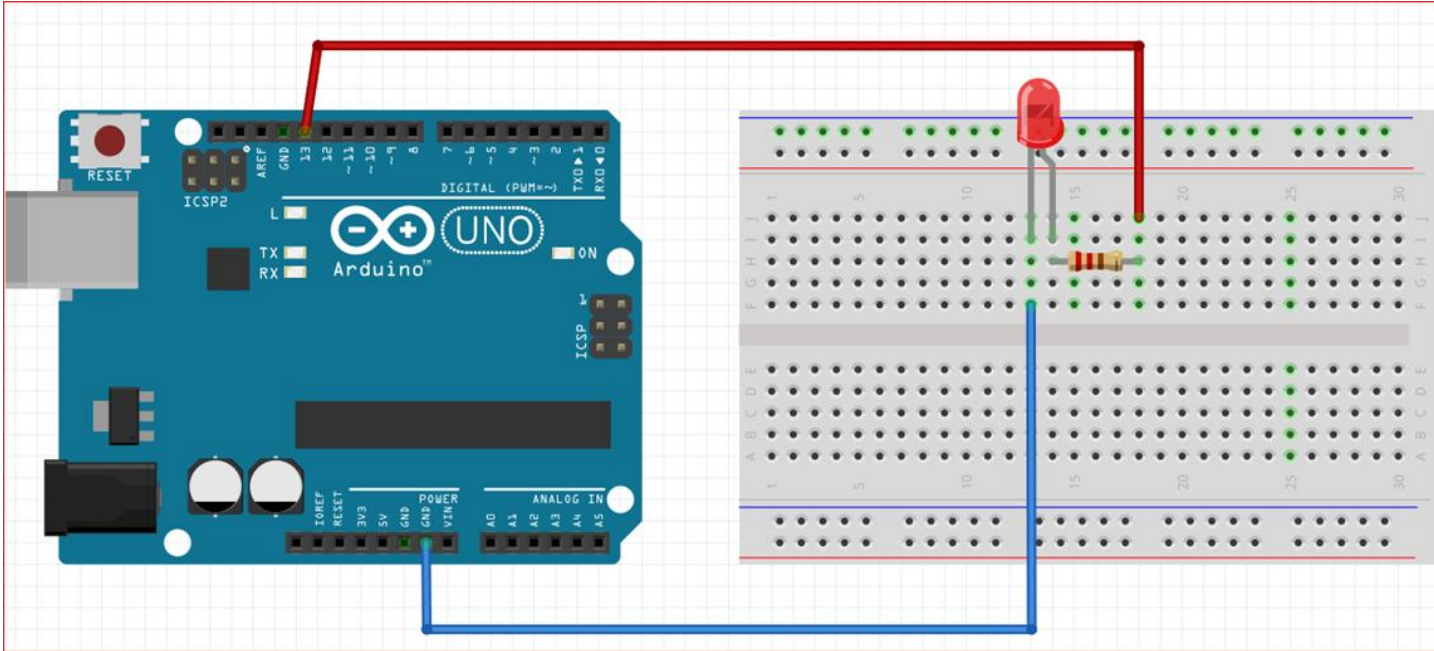




## Schaltung 4: LED einschalten

Aufgabe

Die LED über Node-RED ein-/ausschalten



Teile:

LED

Widerstand 220  $\Omega$

(Im Bild Pin 13)

Aktuell im Sketch

Pin 5

# Arduino\_Schritt4.ino

```
// Arduino_Schritt4.ino
// Von Node-RED Arduino-LED schalten
const byte ledPin = 5;
void setup() {
  Serial.begin(9600);
  while (!Serial) {
    ; // wait for serial port to connect. Needed for native USB port only
  }
  pinMode(ledPin, OUTPUT);
}

void loop() {
  // if we get a valid byte, read analog ins:
  if (Serial.available() > 0) {
    // get incoming string:
    String str = Serial.readStringUntil(':'); // getting the string, separator ":"
    int valueSerial = Serial.readStringUntil('\n').toInt(); // getting the value, separator "linefeed"

    if (str.equals("L")) {
      Serial.print(str + ": " + valueSerial);
      Serial.print('\n');

      if (valueSerial == 1) digitalWrite(ledPin, HIGH);
      if (valueSerial == 0) digitalWrite(ledPin, LOW);
    }
  }
}
```

# Node-RED Flow 4

Node-RED öffnen

RPI-Browser:

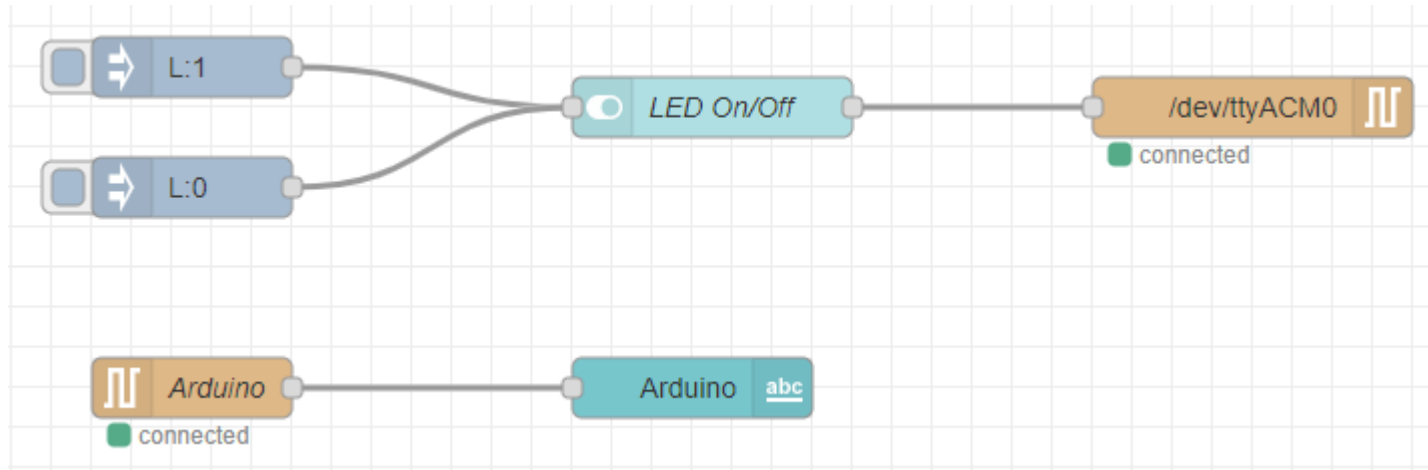
<http://127.0.0.1:1880>

PC-Browser:

<http://192.168.178.xxx:1880>

Flow

Arduino-LED einschalten



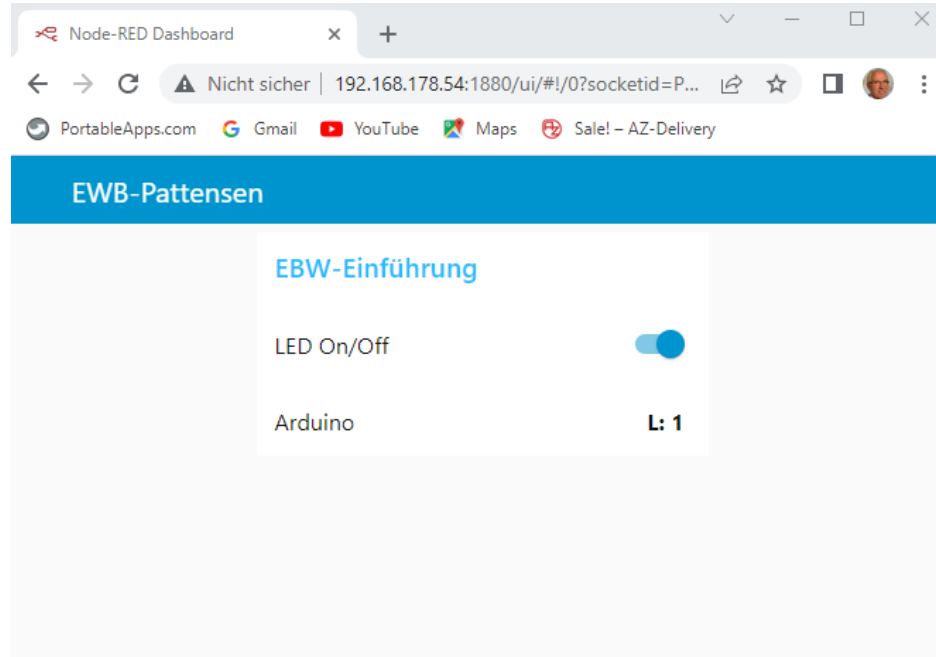
# Node-RED Dashboard 4

Node-RED öffnen

RPI-Browser: <http://127.0.0.1:1880/ui>

PC-Browser: <http://192.168.178.xxx:1880/ui>

Dashboard PC-Browser



---

## Weitere Details


---