

# Installation ShopBoard

## Vorbereiten Raspberry Pi

### Hardware:

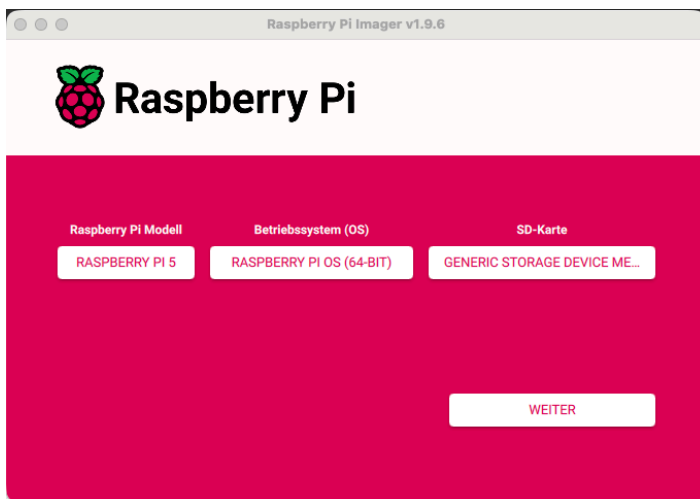
- Raspberry Pi 5B 4GB
- SD Card 64GB
- Kabel MicroHDMI auf HDMI

### Software:

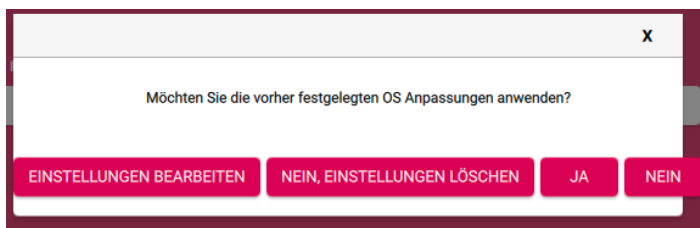
- Installieren RaspberryPi Imager
- Auswahl Image: Raspberry Pi OS 64bit

## OS Installieren

- Installieren Raspberry Pi Imager
- SD-Card einlegen in PC
- Auswahl des Images „Raspberry Pi OS 64 bit



Bearbeiten der Apps zur Installation (kann vieles entfallen)



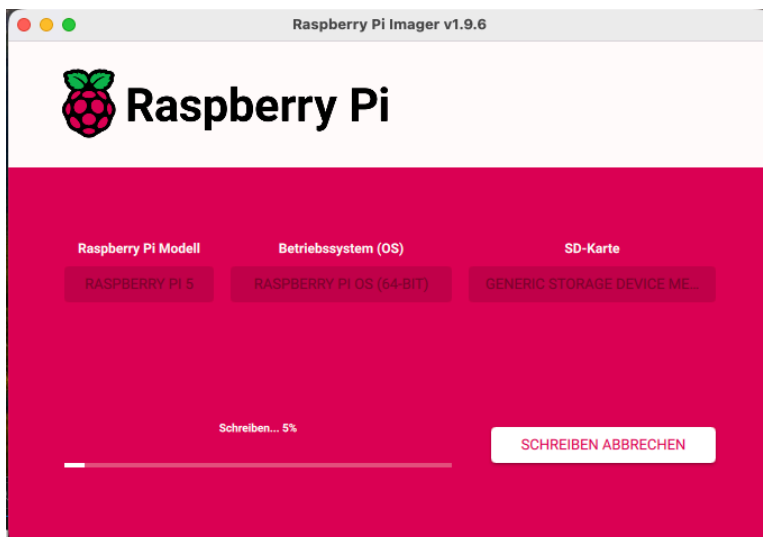
### Einstellen:

- Allgemein:
  - Hostname: „Shopboard-1“.local
  - Nutzernamen: „admin“
  - Passwort: „\_ShopBoard\_“
  - WIFI nicht einrichten (wir machen später einen AccessPoint daraus)
  - Spracheinstellungen: Tastaturlayout: de, Zeitzone: Berlin
- Dienste
  - SSH: aktivieren, Passwort zur Authentifizierung verwenden

- Optionen
  - Tonsignal nach Beenden abspielen: aus
  - Medien nach Beenden auswerfen: aus
  - Telemetrie aktivieren: aus

The screenshot shows the 'Allgemein' tab of the Raspberry Pi Imager. It includes a 'Hostname' field with 'ShopBoard-1.local', a 'Benutzername und Passwort festlegen' section with 'admin' as the username and a masked password, and a 'Wifi einrichten' section with empty fields for 'SSID' and 'Passwort'. At the bottom are 'CANCEL' and 'SPEICHERN' buttons.

Dann im Dialog „Möchten Sie die vorher festgelegten OS-Anpassungen anwenden? JA  
 Frage: Alle auf dem Medium gespeicherten Daten werden gelöscht: JA (prüfen, ob richtiges  
 Volume)  
 Dann startet der Schreibvorgang



## Raspberry Pi anschließen

Den RaspberryPi dann mit dem Adapterkabel (MicroHDMI auf HDMI) an einen Monitor anschließen. Netzwerkkabel anschließen. MicroSD Karte einstecken. Strom (USB-C) an den ganz linken Port anschließen.

### Bluetooth einrichten für Maus/Tastatur

Zum Einrichten ist eine USB-Maus und eine USB-Tastatur ganz gut. Zur Not geht es auch mit einer Tastatur alleine. Dann kann man z.B. auch eine Bluetooth-Maus/Tastatur einrichten. Geht dann so:

- Terminal öffnen mit Ctrl+Alt+T
- Eingeben
  - `sudo systemctl enable bluetooth`
  - `sudo systemctl start bluetooth`
- Das Bluetooth Tool starten
  - `bluetoothctl`
- Der Cursor sollte dann vorn blau sein und bluetooth# da stehen

- Eingeben
  - power on
  - agent on
  - default-agent
  - scan on
- Dann fängt das bluetooth an zu suchen, also Maus/Tastatur in Scan-Modus bringen
- Wenn ok, dann findet der das Gerät und so was steht dort:
  - [NEW] Device AA:BB:CC:DD:EE:FF BTMausxyz
- Dann eingeben
  - pair AA:BB:CC:DD:EE:FF
- Das Gerät sollte nun dran sein. Wenn es noch nicht geht, ggf. nachschieben
  - connect AA:BB:CC:DD:EE:FF
  - trust AA:BB:CC:DD:EE:FF
- Es sollte dann dort stehen: Connection successful

Zurück auf den PC und den Rest dann per SSH machen.

## **X11 Window Manager aktivieren**

Der neue Wayland Manager kann das mit dem Querformat nicht so richtig. Darum nehmen wir den alten X11 dafür wieder.

- sudo raspi-config
- Auf den Menüpunkt „6 Advanced Options“
- Zum Punkt „A7 Wayland“ und dann „W1 X11“
- Dann speichern (mit TAB navigieren) und Reboot

## **Power Management ausschalten**

Der Raspi hat per default ein Power Management. Das schaltet blöderweise den Bildschirm nach einer Weile aus und das wollen wir ja nicht. Also ausschalten:

- sudo nano /etc/rc.local

Im Editor dann diese Zeilen einfügen:

```
xset s off
xset -dpms
xset s noblank
```

Dann mit Ctrl-X beenden, speichern, reboot.

## **Software einrichten**

Terminal-Programm öffnen und „ssh admin@ShopBoard-1“ eingeben. Es folgt die Passwortabfrage und dann ist man am Prompt. Es sollte so was dastehen:

```

Last login: Wed Oct 15 20:43:27 on ttys000
/dev/fd/12:18: command not found: compdef
tommy@MacBook-Air-von-Thomas ~ % ssh admin@ShopBoard-1
The authenticity of host 'shopboard-1 (192.168.178.164)' can't be established.
ED25519 key fingerprint is SHA256:Zvq8pW4WLFUzkkTQEmG8f7wAEa17eT9wTAV3DY1jWTw.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added 'shopboard-1' (ED25519) to the list of known hosts.
admin@shopboard-1's password:
Linux ShopBoard-1 6.12.47+rpt-rpi-2712 #1 SMP PREEMPT Debian 1:6.12.47-1+rpt1 (2025-09-16) aarch64

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.

Wi-Fi is currently blocked by rfkill.
Use raspi-config to set the country before use.

```

```
admin@ShopBoard-1:~ $
```

## Einrichten NodeJS und NPM

Wir müssen nun noch nodeJS und npm installieren. Das geht so:

- hinzufügen des Repos für nodejs
  - `curl -fsSL https://deb.nodesource.com/setup_20.x | sudo -E bash -`
  - `sudo apt install nodejs -y`
- Wenn erfolgreich dann sollte mit `node --version` und `npm --version` etwas kommen

```

[admin@ShopBoard-1:~/preisliste $ sudo apt install nodejs -y
Installing:
  nodejs

Summary:
  Upgrading: 0, Installing: 1, Removing: 0, Not Upgrading: 91
  Download size: 31.0 MB
  Space needed: 197 MB / 51.2 GB available

Get:1 https://deb.nodesource.com/node_20.x nodistro/main arm64 nodejs arm64 20.19.5-1nodesource1 [31.0 MB]
Fetched 31.0 MB in 3s (9,584 kB/s)
Selecting previously unselected package nodejs.
(Reading database ... 119824 files and directories currently installed.)
Preparing to unpack .../nodejs_20.19.5-1nodesource1_arm64.deb ...
Unpacking nodejs (20.19.5-1nodesource1) ...
Setting up nodejs (20.19.5-1nodesource1) ...
Processing triggers for man-db (2.13.1-1) ...
[admin@ShopBoard-1:~/preisliste $ node --version
v20.19.5
[admin@ShopBoard-1:~/preisliste $ npm --version
10.8.2
admin@ShopBoard-1:~/preisliste $

```

## Einrichten der Applikation

Wir wechseln ins Home-Dir: `cd /home/admin`

Dann holen wir das Projekt von GitHub runter.

- `git clone https://github.com/dreyer-thomas/PriceList.git`

```

[admin@ShopBoard-1:~/preisliste $ git clone https://github.com/dreyer-thomas/PriceList.git
Cloning into 'PriceList'...
remote: Enumerating objects: 1325, done.
remote: Counting objects: 100% (79/79), done.
remote: Compressing objects: 100% (63/63), done.
remote: Total 1325 (delta 36), reused 43 (delta 16), pack-reused 1246 (from 1)
Receiving objects: 100% (1325/1325), 3.78 MiB | 3.26 MiB/s, done.
Resolving deltas: 100% (498/498), done.
admin@ShopBoard-1:~/preisliste $

```

Wir wechseln ins Verzeichnis

- `cd PriceList`

Und weiter in server

- cd server

Dort installieren wir die Abhängigkeiten:

- npm install

Das gleiche noch im Verzeichnis darüber

- cd ..
- npm install

```
[admin@ShopBoard-1:~ $ cd PriceList/
[admin@ShopBoard-1:~/PriceList $ cd server
[admin@ShopBoard-1:~/PriceList/server $ npm install

up to date, audited 70 packages in 869ms

14 packages are looking for funding
  run `npm fund` for details

found 0 vulnerabilities
npm notice
npm notice New major version of npm available! 10.8.2 -> 11.6.2
npm notice Changelog: https://github.com/npm/cli/releases/tag/v11.6.2
npm notice To update run: npm install -g npm@11.6.2
npm notice
[admin@ShopBoard-1:~/PriceList/server $ cd ..
[admin@ShopBoard-1:~/PriceList $ npm install
npm warn deprecated rimraf@3.0.2: Rimraf versions prior to v4 are no longer supported
npm warn deprecated lodash.isequal@4.5.0: This package is deprecated. Use require('node:util').isDeepStrictEqual instead.
npm warn deprecated glob@7.2.3: Glob versions prior to v9 are no longer supported
npm warn deprecated inflight@1.0.6: This module is not supported, and leaks memory. Do not use it. Check out lru-cache if you want a good and tested way to coalesce async requests by
a key value, which is much more comprehensive and powerful.

added 623 packages, and audited 624 packages in 33s

109 packages are looking for funding
  run `npm fund` for details

7 vulnerabilities (6 low, 1 high)

To address all issues, run:
  npm audit fix

Run `npm audit` for details.
admin@ShopBoard-1:~/PriceList $
```

Um das Ganze einfacher zu machen gibt es ein Skript unter /home/admin/PriceList/deploy, welches das alles zusammen macht. Das muss aber noch kopiert werden und ausführbar gemacht werden. Also:

- cp /home/admin/PriceList/deploy/startup.sh /home/admin/startup.sh
- Chmod +x /home/admin/startup.sh

Von nun an kann man immer wieder das Script unter /home/admin/startup.sh ausführen. Das macht dann folgendes:

- abrufen einer neuen Version inkl. Scripts von GitHub
- Installieren von neuen Abhängigkeiten im NodeJS
- Kompilieren der Anwendung (vorher backup der Settings und danach restore)
- Kopieren der ausführbaren Skripte in den /home/admin
- Starten des Webserver

Zum Beenden des Webserver einfach ctrl-x auf der Konsole und dann ist man wieder am Prompt.

## Set-Display Service einrichten

Dieser Dienst sorgt dafür, dass der Monitor auf eine feste Auflösung eingestellt wird und rotiert wird. Der Dienst ist schon im git repository mit drin unter /deploy. Also von dort den Dienst kopieren:

- sudo cp /home/admin/PriceList/deploy/display\_setup.service /etc/systemd/system/display\_setup.service

Der Dienst nutzt Umgebungsvariablen, die wir zunächst anlegen müssen:

- sudo nano /etc/default/display

Dann öffnet sich ein Text-Editor, der folgende Zeilen bekommt:

DISPLAY=:0

XAUTHORITY=/home/admin/.Xauthority

OUTPUT=HDMI-1

MODE=1920x1080

# leer = keine Drehung; oder: left|right|inverted|normal

ROTATE=

Der Wert hinter ROTATE ist entweder leer (dann haben wir den Bildschirm in quer) oder „left“, dann haben wir das Bild auf hochkant gestellt. Mit diesem Befehl kann das dann auch immer wieder geändert werden. Die anderen Einstellungen sollten so bleiben, wären aber ggf. auch hier einstellbar. Eine Beispieldatei befindet sich auch im deploy. Somit könnte das auch so gehen:

Sudo cp /home/admin/PriceList/deploy/display /etc/default/display

## WLAN-Accesspoint einrichten

Das ShopBoard soll per WLAN erreichbar sein. Dafür wird ein Access-Point eingerichtet. Dieser kann dann als admin genutzt werden für Einstellungen. Das besteht aus zwei Teilen: der Konfiguration und dem Skript.

Die Konfiguration befindet sich in der Datei /home/admin/PriceList/deploy/hotspot. Der Inhalt sieht so aus:

```
# Pflichtwerte
SSID="ShopBoard-1"
PSK="SB_270177"           # 8..63 Zeichen
WIFI_COUNTRY="DE"         # WLAN-Regdom
WLAN_IF="wlan0"           # WLAN-Interface
GATEWAY_IP="10.42.0.1"    # Hotspot-IP (NetworkManager Shared)

# Steuerung
HIDDEN="no"
BAND="bg"
CHANNEL=„6“
```

Die Werte können noch geändert werden. Um die Konfiguration im System anzulegen folgendes eingeben:

- sudo cp /home/admin/PriceList/deploy/hotspot /etc/default/hotspot

Bei Bedarf kann die Datei angepasst werden mit:

- sudo nano /etc/default/hotspot

Dann mit dem Editor ändern und mit Ctrl-X und yes beenden und speichern.

Dann das Installationsscript bereit legen:

- cp /home/admin/PriceList/deploy/setup-hotspot-nmcli.sh /home/admin/setup-hotspot-nmcli.sh

- chmod +x /home/admin/setup-hotspot-nmcli.sh

Und das script dann ausführen:

- sudo /home/admin/setup-hotspot-nmcli.sh

```
[admin@ShopBoard-1:~ $ sudo ./setup-hotspot-nmcli.sh
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
rfkill is already the newest version (2.41-5).
iw is already the newest version (6.9-1+b1).
0 upgraded, 0 newly installed, 0 to remove and 91 not upgraded.
==> Preflight für wlan0
==> Entferne ggf. alte Verbindung: hotspot-wlan0
Connection 'hotspot-wlan0' (25e5495b-7d44-43e3-8ac7-0a941636e134) successfully deleted.
==> Lege Hotspot an (WPA2-Only, hidden)
Connection 'hotspot-wlan0' (934da5e8-6233-4a12-8db7-c2eb10ee4f6b) successfully added.
==> Starte Hotspot...
Connection successfully activated (D-Bus active path: /org/freedesktop/NetworkManager/ActiveConnection/3)
==> Beacon/Security-Check (Scan nach eigener SSID)
```

Das WLAN sollte nun verfügbar sein unter dem Namen „ShopBoard-1“ mit dem Passwort „SB\_270177“. Wenn die Daten in der Konfiguration geändert wurden, dann gelten diese.

## Dienst für den WebServer anlegen

Das ShopBoard wird als WebServer ausgeführt und daher muss dieser Dienst ebenfalls gestartet werden. Dafür gibt es wieder eine Service-Datei, die wir anlegen müssen:

- `sudo cp /home/admin/PriceList/deploy/preisliste.service /etc/systemd/system/preisliste.service`
- `sudo systemctl daemon-reload`
- `sudo systemctl enable preisliste.service`
- `sudo systemctl start preisliste.service`

Damit sollte der Dienst nach reboot dann bereit sein und die Anwendung aktualisieren und starten.

## Browser im Kiosk-Modus starten

Zur Anzeige der Anwendung wird der Browser im Kiosk-Modus genutzt. Damit hat der keine Bedienelemente, nur den Inhalt und ist Bildschirmfüllend. Der wird mit dieser Datei angelegt:

`/config/lxsession/LXDE-pi/autostart`

Und die sollte diesen Inhalt haben:

```
@xset s off
@xset -dpms
@xset s noblank
@xrandr
@chromium-browser --kiosk --app=http://localhost:3000
```

Die Datei befindet sich oim /deploy Ordner und ist browser benannt. Zum Einrichten das ausführen:

- `sudo /home/admin/browser.sh`

Dann neu starten

- `sudo reboot`