

Anleitung zum Nachbau

Im Folgenden wird eine detaillierte Anleitung zum schrittweisen Nachbau des Messgeräts beschrieben. Dabei ist zu berücksichtigen, dass einige Hard- und Software-Komponenten ausgetauscht werden können und sich die Ausführungen dementsprechend verändern können.

1.1. Installation des Betriebssystems mit dem Raspberry Pi Imager



Abbildung 1: Hauptmenü des Raspberry Pi Imager

Die microSD-Karte muss in ein geeignetes Laufwerk des PCs gesteckt und der Raspberry Pi Imager gestartet werden. Als Betriebssystem wird „RASPBERRY PI OS (32-BIT)“ und als SD-Karte wird das Laufwerk mit der zuvor eingesetzten SD-Karte ausgewählt (Abbildung 1).

Für die Aktivierung der Verbindung über SSH müssen die erweiterten Optionen (Zahnrad-Icon) aufgerufen werden.

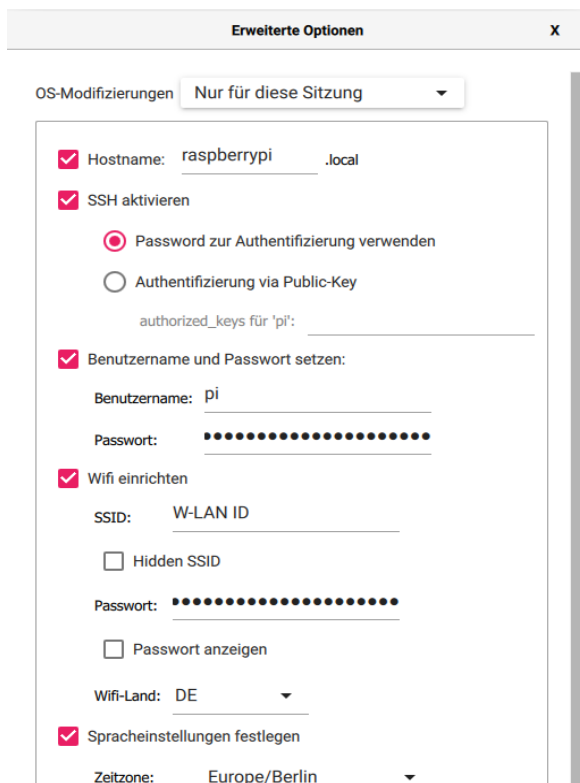


Abbildung 2: Erweiterte Einstellungen im RPI Imager

Hostname, Benutzername und Kennwort, können beliebig gewählt werden (Abbildung 2). Das Häkchen bei „SSH“ und „Wifi einrichten“ muss gesetzt werden. Unter Wifi einrichten wird das vom PC verwendete Drahtlos-Netzwerk samt Passwort und Land eingetragen. Zeitzone und Tastaturlayout können ebenfalls festgelegt werden. Nachdem alle Einstellungen getroffen sind, kann die SD-Karte beschrieben werden.

Die mit dem Betriebssystem beschriebene SD-Karte wird in den dafür vorgesehenen Slot des Raspberry Pis eingesteckt. Der Raspberry Pi Zero W verfügt über zwei Mikro-USB Ports „PWR“ und „USB“. Mit einem Mikro-USB-Datenkabel können PC und Raspberry Pi über den „USB“-Slot verbunden werden. Der Raspberry Pi wird vom

PC mit Strom versorgt. Wenn die grüne LED auf der Platine aufhört zu blinken, ist der RPI hochgefahren.

1.2. RNDIS-Treiber und Bonjour Print Services

Da der Raspberry Pi Zero W nicht über einen Ethernet-Port verfügt, aber zur Datenübertragung eine Netzwerkverbindung erforderlich ist, ist es notwendig USB-Networking zu ermöglichen. Dafür müssen ein RNDIS-Treiber (Remote Network Driver Interface Specification, Anhang 1.2) und der Apple Service „Bonjour“ (nur bei Windows Betriebssystemen, Anhang 1.3) heruntergeladen beziehungsweise installiert werden.

Für die Installation des RNDIS-Treibers wird der Geräte-Manager geöffnet und unter „Anschlüsse (COM & LPT)“ mit einem Rechtsklick der Treiber für das serielle USB-Gerät aktualisiert (Abbildung 3).

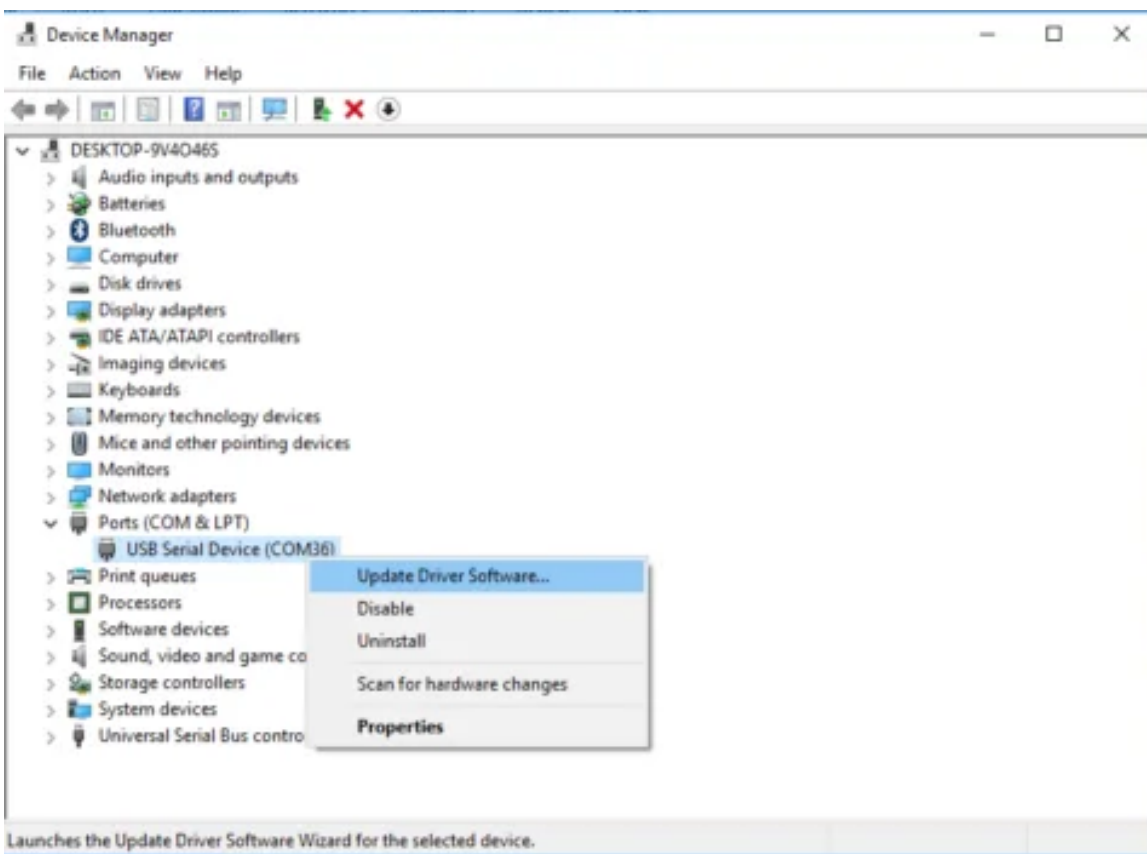


Abbildung 3: Installation des RNDIS-Treibers (Sarath Kumar 2018)

Im Fenster, das sich daraufhin öffnet, wird anschließend „Auf meinem Computer nach Treibern suchen“ ausgewählt. Im nächsten Fenster wird der Pfad angegeben, in den die ZIP-Datei aus Anhang 1.2 entpackt wurde.

Zur Installation von Bonjour muss den Anweisungen des Installationsprogrammes aus Anhang 1.3 gefolgt werden.

1.3. Booten und Verbindungsaufbau mittels SSH

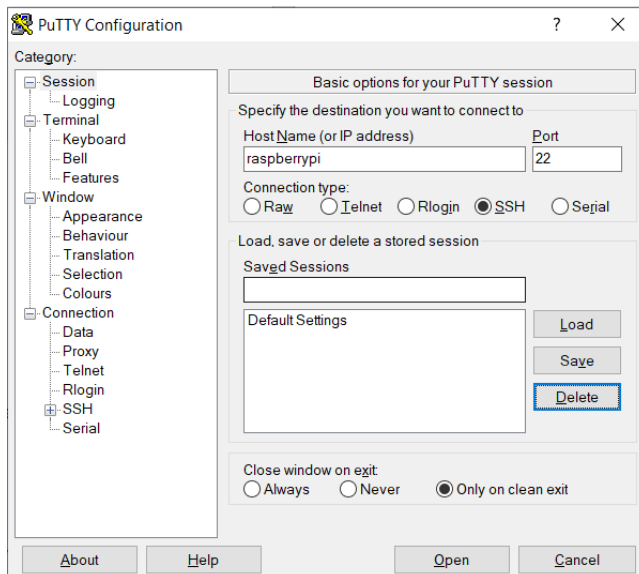


Abbildung 4: PuTTY-Konfiguration

Als nächstes muss die Verbindung über SSH mittels Drahtlosnetzwerk aufgebaut werden. Dafür wird das Programm „PuTTY“ (siehe Anhang 1.4) installiert und geöffnet. Mit der Eingabe des in 1.1. verwendeten Hostnamen und Verwendung von Port 22 kann mit „Open“ die SSH-Verbindung aufgebaut werden (Abbildung 4).

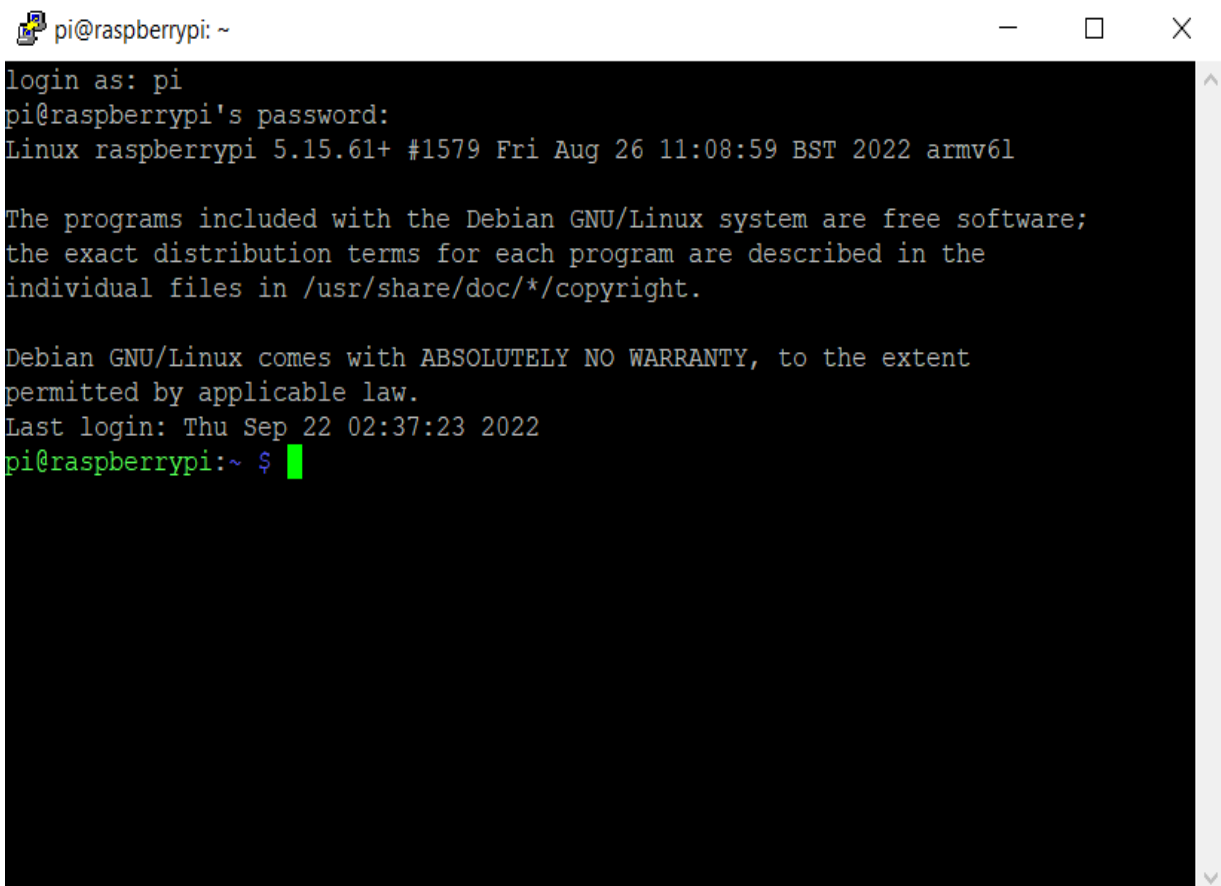


Abbildung 5: PuTTY-Login mit Nutzernamen und Passwort

Bei erfolgreicher Verbindung kann sich mit dem in 1.1. gewählten Nutzernamen und Passwort eingeloggt werden (Abbildung 5). Um Komptabilitätsprobleme zu vermeiden, wird das Betriebssystem mit den Befehlen

```
sudo apt-get update
```

und

```
sudo apt-get upgrade
```

auf den neuesten Stand gebracht.

1.4. Einstellen des USB-Networking

Damit die Verbindung über USB auch ohne Internetzugang erfolgen kann, muss das USB-Networking über PuTTY aktiviert werden. Dafür werden folgende Befehle im Terminal ausgeführt:

```
echo dtoverlay=dwc2 | sudo tee -a /boot/config.txt
```

```
echo dwc2 | sudo tee -a /etc/modules
```

```
echo g_ether | sudo tee -a /etc/modules
```

Nach einem Neustart des Raspberry Pis kann nun auch ohne aktive Internetverbindung eine SSH-Verbindung aufgebaut werden.

1.5. Herunterladen und Installation der Skripte

In diesem Schritt wird das Python-Skript aus Anhang 3 heruntergeladen bzw. kopiert und auf dem Raspberry Pi gespeichert. Dazu muss WinSCP (Anhang 1.5) installiert werden.

Unter „Neue Sitzung“ müssen die Zugangsdaten für den Benutzernamen und Passwort aus 1.1 gewählt werden (Abbildung 6).

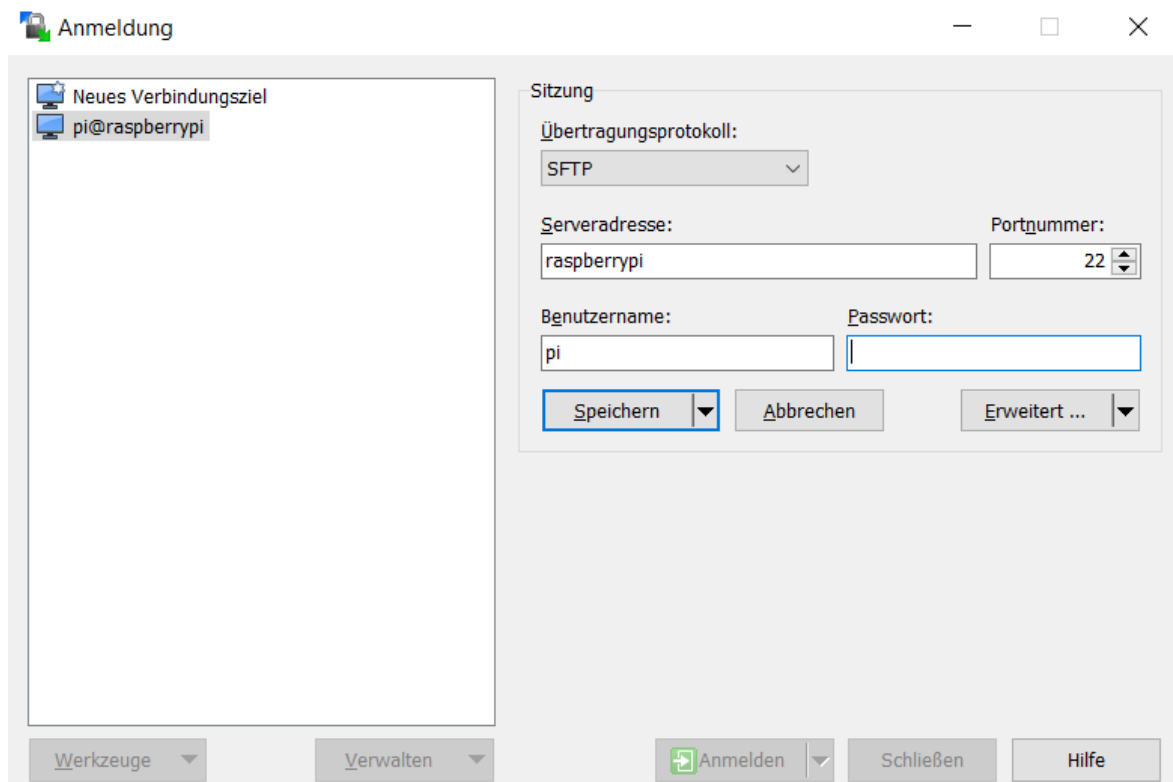


Abbildung 6: WinSCP-Konfiguration

Für einen vollen Zugriff auf alle Ordner müssen administrative Rechte freigeschaltet sein. Dazu wird unter „Erweitert ...“ – „Umgebung“ – „Shell“ der Befehl `sudo -s` eingetragen (Abbildung 7).

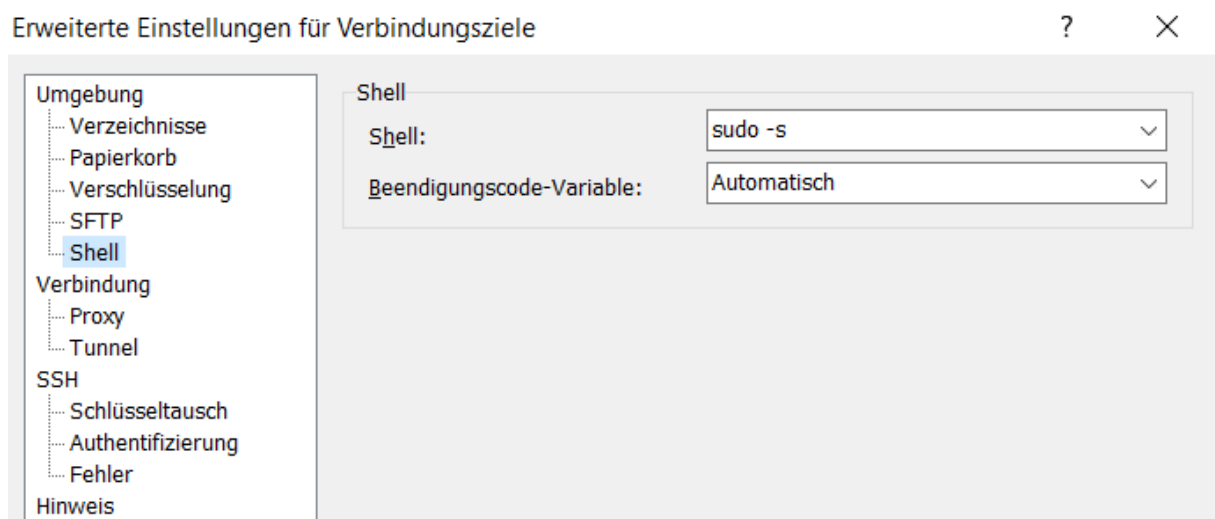


Abbildung 7: Freischaltung administrativer Rechte für WinSCP

Nun kann mit „Anmelden“ eine Verbindung aufgebaut werden. Das Skript aus Anhang 3 wird anschließend mit WinSCP im Ordner `/home/pi/` platziert.

1.6. Automatisches Starten der Skripte

Damit das Skript bei jedem Hochfahren des Raspberry Pi automatisch startet wird ein sogenannter „Systemd service“ im Betriebssystem eingerichtet. Dieses Verfahren stellt sicher, dass das Skript auch bei unerwartetem Beenden des Skriptes immer wieder neu gestartet wird, solange der Raspberry Pi mit Strom versorgt und hochgefahren ist. Als erstes wird dafür mit Notepad++ (Anhang 1.6) eine Datei mit folgendem Inhalt erstellt:

```
[Unit]

Description=Run Rain Gauge and Button on boot

After=multi-user.target

[Service]

WorkingDirectory=/home/pi

Type=idle

User=pi

ExecStart=/usr/bin/python3 /home/pi/raingauge.py

Restart=always

[Install]

WantedBy=multi-user.target
```

Die Datei wird als „raingaugeonreboot.service“ abgespeichert und mit WinSCP nach `/etc/systemd/system/` kopiert:

Im nächsten Schritt muss der Zugriff erlaubt und der Service mit folgenden Befehlen im PuTTY-Terminal freigegeben werden:

```
sudo chmod 644 /etc/systemd/system/raingaugeonreboot.service

sudo systemctl daemon-reload

sudo systemctl enable raingaugeonreboot
```

Nach einem Neustart (`sudo reboot`) des Systems ist der Service eingerichtet.

1.7. Installation der Echtzeit-Uhr

Der Raspberry Pi Zero W verfügt nicht über eine Echtzeit-Uhr, weshalb sich die interne Uhr des Betriebssystems über einen längeren Zeitraum ohne Internetanschluss verstellen wird. Damit die Zeitstempel der aufgezeichneten Daten dauerhaft korrekt sind, wird ein Echtzeituhr-Modul (Anhang 2.5) angeschlossen. Das Modul kann wie in Abbildung 8 dargestellt auf die GPIO-Pins (v.l.n.r. Pin Nummer 1,3,5,7,9) gesteckt werden.

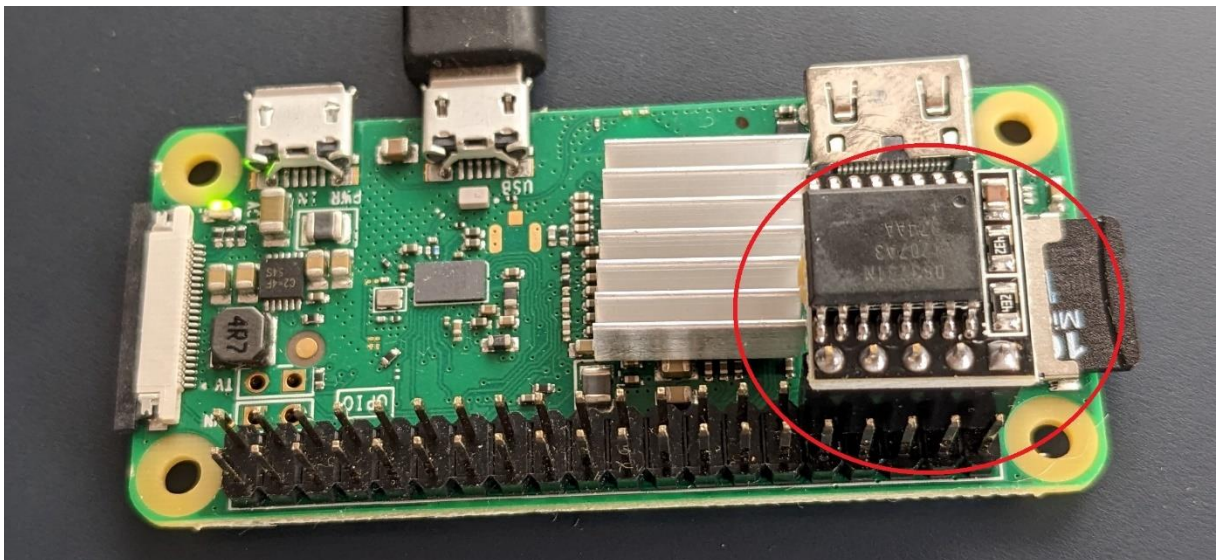


Abbildung 8: Echtzeituhr DS3231 Steckmodul auf der GPIO-Pinleiste

Nun muss noch der I²C Datenbus aktiviert werden, damit die Echtzeit-Uhr korrekt ausgelesen werden kann. Mit dem Befehl „sudo raspi-config“ im PuTTY Terminal öffnet sich das Konfigurationswerkzeug des Raspberry Pi. Hier wird unter „Interfacing Options“ die Rubrik „I2C“ ausgewählt und aktiviert. Damit die Echtzeit-Uhr bei jedem Neustart des Geräts wiederverwendet wird, muss ähnlich wie in 1.4. die boot.txt Datei im PuTTY Terminal mit dem Befehl

```
echo dtoverlay=i2c-rtc,ds3231 | sudo tee -a /boot/config.txt
```

ergänzt werden. Außerdem muss eine neue Regel unter Verwendung von Notepad++ mit dem Dateinamen „rtc-start.rules“ erstellt werden. In die erste Zeile der Datei wird der Befehl

```
ACTION=="add", SUBSYSTEM=="rtc", ATTRS{hctosys}=="0",  
RUN+="/sbin/hwclock -s --utc"
```

geschrieben und diese anschließend mit WinSCP in das Verzeichnis /etc/udev/rules.d/ kopiert. Bei einer aktiven Internetverbindung des Rechners, der den Raspberry Pi headless betreibt, kann nun die aktuelle Uhrzeit mit dem Befehl `sudo hwclock -w` in das Modul

geschrieben und mit dem Befehl `sudo hwclock -r` überprüft werden. Mit dem Befehl `sudo apt-get purge fake-hwclock` wird abschließend die interne Uhr deaktiviert, da es sonst zu Problemen bei der Uhrzeitbestimmung des Systems kommen kann.

1.8. Drucktaster zum Herunterfahren und Deaktivieren des W-LAN

Für das kontrollierte Herunterfahren des Raspberry Pis und der manuellen Deaktivierung des W-LAN zur Stromeinsparung wird der Drucktaster (Anhang 2.6) mit den Jumper-Kabeln wie in Abbildung 9 angeschlossen. Pin-Nummer 38 (GPIO20) wird mit der Signalschnittstelle, Pin-Nummer 17 (3V3) mit der Stromversorgung und Pin-Nummer 34 (Ground) mit der Masse des Drucktasters verbunden.

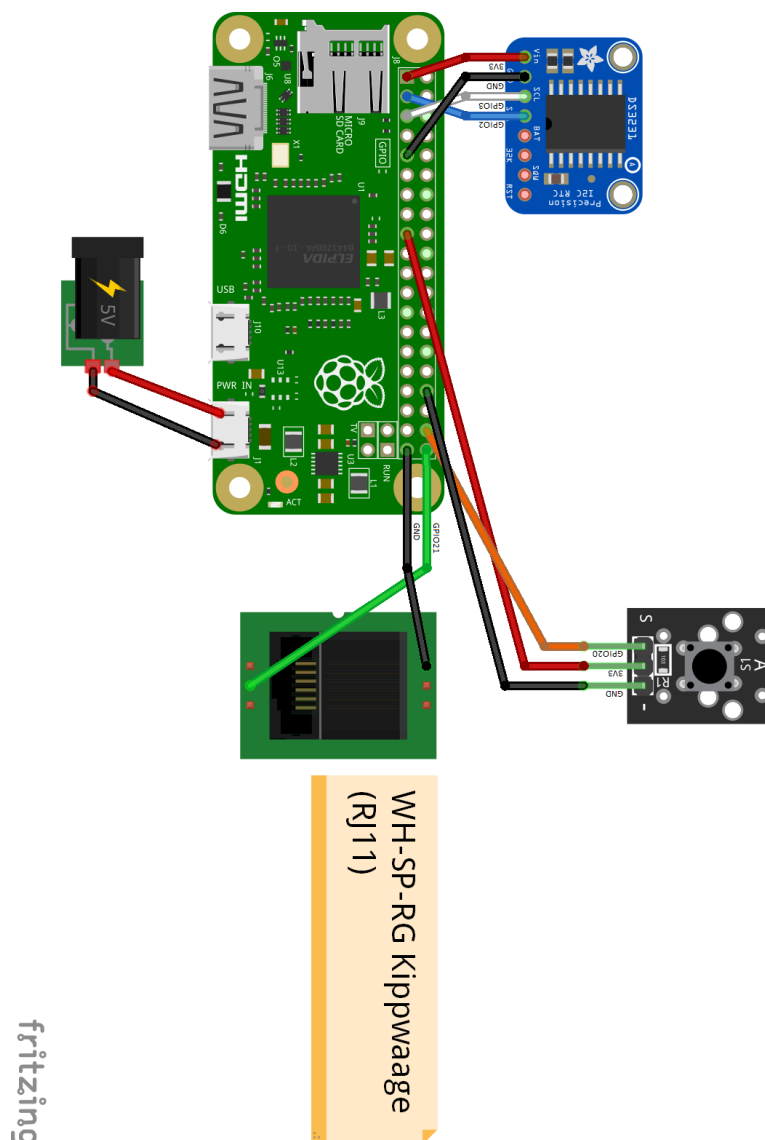


Abbildung 9: Pin-Belegung des Drucktasters

1.9. Stromeinsparung

Zusätzlich zur Deaktivierung des W-LAN können weitere Maßnahmen zur Stromeinsparungen ergriffen werden. Der HDMI-Port findet während des Betriebs keine Verwendung und kann deshalb folgendermaßen deaktiviert werden:

Im Verzeichnis `/etc/` befindet sich die Datei `rc.local`, welche über WinSCP mit Notepad++ geöffnet wird. Hier werden die Beschreibung und der Befehl:

```
#Disable HDMI
```

```
/usr/bin/tvservice -o
```

vor dem Abschlusseintrag `exit 0` eingefügt.

Dazu kann auch die Bluetooth-Funktion deaktiviert werden:

Im Verzeichnis `/boot/` wird die Datei `config.txt` mit Notepad++ geöffnet. Unter dem Textabschnitt `[all]` wird am Ende die neue Zeile `dtoverlay=disable-bt` eingefügt. Nach einem Neustart sind beide Funktionen abgeschaltet.

1.10. Gehäusezusammenbau und Inbetriebnahme

Für den Zusammenbau wird zunächst der RJ-11-Adapter (Anhang 2.3) auf das Breakout-Board (Anhang 2.4) gesteckt und mit Jumper-Kabeln (Anhang 2.11) verbunden. Die männlichen Enden der Kabel werden mit den Positionen 3 & 4 des Breakout-Boards verlötet. Die weiblichen Enden des Kabels werden auf GPIO-Pin 21 und einen Ground-Pin gesteckt. Nun kann der RJ-11 Stecker der Kippwaage mit dem Adapter verbunden werden.

Damit das gesammelte Wasser abfließen kann, werden einige Löcher in die Platte gebohrt. Als nächstes wird die Kippwaage auf der Platte (Anhang 2.12) mit Schrauben fixiert. Anschließend wird das Rohr (Anhang 2.7) auf eine Länge von ca. 26 cm zugesägt und anschließend mit aufgesetztem Trichter über der Kippwaage zentriert und mit Winkeln fixiert.

Der Raspberry Pi wird in der wasserfesten Verbindungs-Box (Anhang 2.9) montiert und die Kabel für die Kippwaage und den Akku durch die Öffnungen geführt und angeschlossen. Abbildung 10 zeigt das betriebsbereite Messgerät. Für die Inbetriebnahme müssen lediglich der Akku und der Raspberry Pi über den USB-Anschluss verbunden werden. Nach dem Hochfahren wird der Drucktaster einmal betätigt, um das WLAN zu deaktivieren. Die

aufgezeichneten Messdaten können in der Datei `/home/pi/rainfall_data.csv` ausgelesen werden.

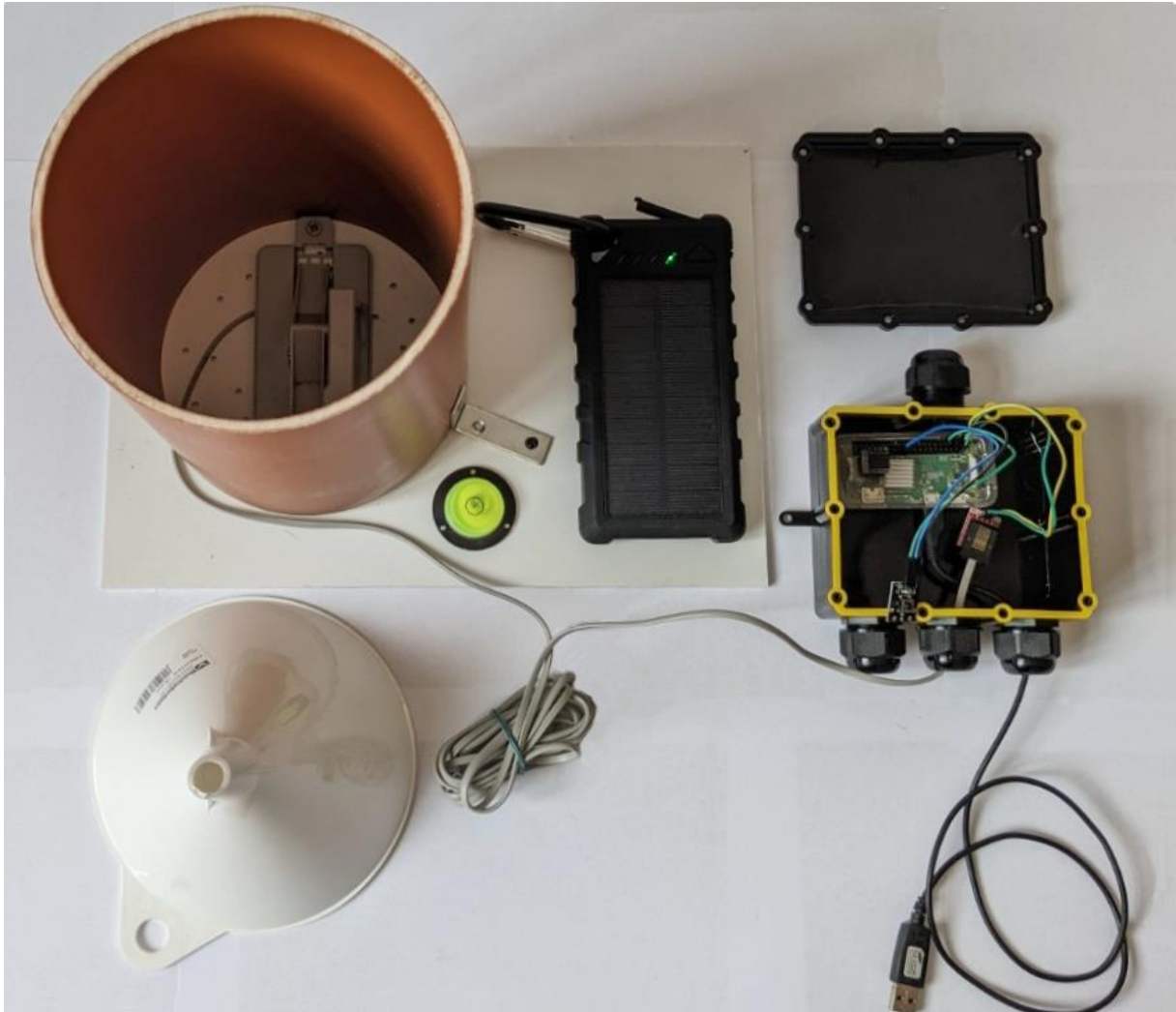


Abbildung 10: Zusammengebautes Niederschlagsmessgerät

Anhang 1: Links und Software

Anhang 1.1: Raspberry Pi Imager

<https://www.raspberrypi.com/software/>

Anhang 1.2: Hyperlink zum RNDIS-Treiber

<https://www.factoryforward.com/wp-content/uploads/2018/01/mod-duo-rndis.zip>

Anhang 1.3: Bonjour Print Services

https://support.apple.com/kb/DL999?locale=en_US

Anhang 1.4: PuTTY

<https://www.putty.org/>

Anhang 1.5: WinSCP

<https://winscp.net/eng/download.php>

Anhang 1.6: Notepad++

<https://notepad-plus-plus.org/downloads/>

Anhang 2: Hardware

Anhang 2.1: Raspberry Pi Zero W

<https://www.raspberrypi.com/products/raspberry-pi-zero-w/>

Anhang 2.2: Kippwaage WH-SP-RG

<https://www.ideal.de/preisvergleich/Liste/118104246/wh-sp-rg.html>

Anhang 2.3: RJ11-Adapter

<https://www.berrybase.de/rj11-6-pin-steckverbinder>

Anhang 2.4: RJ11-Breakout-Board

<https://www.berrybase.de/sparkfun-rj11-breakout>

Anhang 2.5: Real Time Clock DS3231

<https://www.berrybase.de/en/ds3231-real-time-clock-modul-fuer-raspberry-pi>

Anhang 2.6: KY-004 Drucktaster

<https://www.berrybase.de/ky-004-drucktaster-modul>

Anhang 2.7: Abflussrohr

<https://www.hornbach.de/p/kg-rohr-dn-160-laenge-0-50m/267710/>

Anhang 2.8: Trichter

<https://www.shopping4u-24.com/kategorien/plastikwaren/trinkflasche/trichter/a-13032342>

Anhang 2.9: Wasserdichte Verbindungsbox

<https://www.longlife-led.de/Verbindungsbox-XXL-gross-4-fach-5-polig-IP68-wasserdicht-24A-450V/4503>

Anhang 2.10: Dupont Kabel Female

<https://www.berrybase.de/40pin-jumper/dupont-kabel-female-female-trennbar>

Anhang 2.11: Dupont Kabel Male Female

<https://www.berrybase.de/40pin-jumper/dupont-kabel-male-female-trennbar>

Anhang 2.12: Polystyrol Platte

<https://www.conrad.de/de/p/polystyrol-platte-reely-l-x-b-330-mm-x-230-mm-5-mm-229798.html>

Anhang 2.13: Toolcraft Libelle

<https://www.conrad.de/de/p/toolcraft-wwn-136-to-6699408-mini-wasserwaage-2233136.html>

Anhang 3: Python Skript

```
import gpiozero
import time
import csv
import sys
import shutil
import threading
import os
from datetime import datetime

# define the GPIO input pins
poweroff = gpiozero.Button(20, pull_up=False)
tippingbucket = gpiozero.Button(21, pull_up=True)

# set the directory where the file is supposed to be saved
directory = '/home/pi/rainfall_data.csv'

# check if the directory exist first
if not os.path.exists(os.path.dirname(directory)):
    # if the directory does not exist, create it
    os.makedirs(os.path.dirname(directory))

# check if the file exists first
if not os.path.exists(directory):
    # if the file doesn't exist, create it
    open(directory, 'a').close()

# create lock for controlled access of the data file
file_lock = threading.Lock()

# create a function that saves a backup file every "TIME_TO_BACKUP" seconds
def backup_file():
    TIME_TO_BACKUP = 900
    while True:
        time.sleep(TIME_TO_BACKUP)
        with file_lock:
            shutil.copy(directory, '/home/pi/rainfall_data_backup.csv')

backup_thread = threading.Thread(target=backup_file)
backup_thread.start()

# define the button press count and create a function to shutdown wifi after 1 and shutdown after
# 2 counts
button_press_count = 0

def button_press():
    global button_press_count
    button_press_count += 1
    if button_press_count == 1:
        os.system("sudo ip link set wlan0 down")
```

```

elif button_press_count >= 2:
    os.system("sudo shutdown -h now")

# call the button press function if the poweroff button is pressed
poweroff.when_pressed = button_press

# define a function that writes a timestamp to a file
def tipping_impulse():
    with file_lock:
        writer.writerow([datetime.now().strftime("%Y-%m-%d %H:%M:%S")])
        # flush the memory buffer
        csvfile.flush()
    time.sleep(0.4)

# open the file and write a timestamp through the tipping_impulse function
with open(directory , 'a', newline='') as csvfile:
    # create a csv writer object
    writer = csv.writer(csvfile)
    start_time = time.time()
    # test for errors
    try:

        while True:
            tippingbucket.when_pressed = tipping_impulse

except KeyboardInterrupt:
    # save and close the file when the script is interrupted
    csvfile.close()
    sys.exit()

```