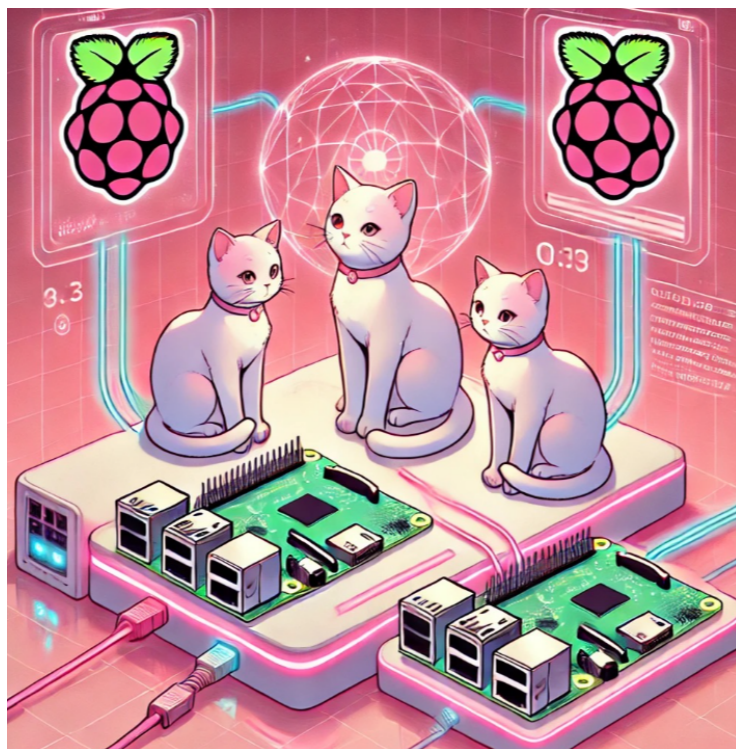


Arbeitsbericht

Modul RPi 2 (Display)



Abb¹: Ein Bild zum thema Raspberry PI

Name: **Lena-Marie Kaufleitner, Felix Neumayer**
Klasse: **4AHITS**
Fach: **ITP2I**
Datum: **24.01.2025**

¹Ai Genrated mit DALL·E

Inhaltsverzeichnis

1	Einleitung	3
2	Inbetriebnahme des Displays	3
3	Python-Programm zur Anzeige von Text	3
4	Automatischer Start des Skripts	4
4.1	Shell-Skript	4
4.2	Script ausführbar machen	4
4.3	rc.local anpassen	4
5	Font-Download	4
6	Fazit	5

1 Einleitung

In diesem Arbeitsbericht wird die Einrichtung eines 0,96 Zoll OLED SSD1306 Displays mit I2C-Schnittstelle am Raspberry Pi beschrieben. Die Aufgabe bestand darin, das Display in Betrieb zu nehmen, ein Python-Programm zu erstellen, das Netzwerkdetails ausgibt und das Programm anschließend so einzurichten, dass es beim Systemstart automatisch startet. Das Display kommuniziert über den sogenannten I2C-Bus (Inter-Integrated Circuit). Dieser Bus ermöglicht es, dass mehrere Geräte über nur zwei Datenleitungen (SDA, SCL) angeschlossen werden können. Für viele Sensoren und Displays beim Raspberry Pi ist I2C die Standard-Kommunikationsart.²

2 Inbetriebnahme des Displays

Die Verkabelung des Displays erfolgte gemäß der Spezifikationen des SSD1306:

Display	Raspberry Pi
VCC	3.3V
GND	GND
SCL	GPIO3 (Pin 5)
SDA	GPIO2 (Pin 3)

Zur Aktivierung der I2C-Schnittstelle:

```
sudo raspi-config
# Interfacing Options -> I2C aktivieren
sudo apt update
sudo apt install -y i2c-tools python3-pip
```

Hintergrund: Die I2C-Schnittstelle muss softwareseitig aktiviert werden, da sie standardmäßig deaktiviert ist, um ungewollte Störungen auf dem Bus zu vermeiden.

3 Python-Programm zur Anzeige von Text

Das Python-Programm liest dynamisch Netzwerkdaten vom Raspberry Pi aus und zeigt diese auf dem Display an:

```
import board
import busio
import adafruit_ssd1306
import netifaces as ni
import ipaddress
# I2C-Schnittstelle initialisieren
i2c = busio.I2C(board.SCL, board.SDA)
oled = adafruit_ssd1306.SSD1306_I2C(128, 64, i2c)
# Netzwerkdetails ermitteln
iface_name = "wlan0" # Netzwerk-Interface
ip = ni.ifaddresses(iface_name)[ni.AF_INET][0]['addr']
netmask = ni.ifaddresses(iface_name)[ni.AF_INET][0]['netmask']
iface = ipaddress.IPv4Interface(f"{ip}/{netmask}")
hostname = ni.gateways()['default'][ni.AF_INET][1]
# Text auf dem Display ausgeben
oled.fill(0)
```

²Quelle:<https://www.elektronik-kompodium.de/sites/raspberry-pi/1907101.html>

```
oled.text(f"Hostname:_{hostname}", 0, 0)
oled.text(f"IP:_{iface.ip}", 0, 10)
oled.text(f"Netmask:_{iface.network.prefixlen}", 0, 20)
oled.show()
```

Erklärung:

- `netifaces`: Liest IP-Adresse und Netzmaske aus dem Netzwerkinterface.
- `ipaddress`: Wandelt die Netzmaske in die Kurzschreibweise (/24) um.
- `adafruit_ssd1306`: Nutzt den Framebuffer des Displays zur Anzeige.³⁴⁵

4 Automatischer Start des Skripts

4.1 Shell-Skript

Erstellen eines Shell-Skripts zum Starten des Python-Programms:

```
nano /home/pi/dev/runrc.sh
```

Inhalt:

```
#!/bin/bash
cd /home/pi/dev
python3 display_text.py > log.txt 2>&1 &
```

Hinweis: Alle Ausgaben (inklusive Fehler) werden in die `log.txt` geschrieben.

4.2 Script ausführbar machen

```
chmod +x /home/pi/dev/runrc.sh
```

4.3 rc.local anpassen

```
sudo nano /etc/rc.local
```

Vor `exit 0` einfügen:

```
sudo -u pi /home/pi/dev/runrc.sh
```

Hintergrund: Durch `sudo -u pi` läuft das Script als Benutzer `pi`, da einige Bibliotheken als `root` nicht korrekt funktionieren.

5 Font-Download

Zur besseren Textdarstellung wurde ein Font heruntergeladen:

```
wget https://github.com/adafruit/Adafruit_CircuitPython_framebuf/raw/main/examples/font5x8.bin
```

³Quelle: <https://pypi.org/project/netifaces/>

⁴Quelle: <https://docs.python.org/3/library/ipaddress.html>

⁵Quelle: <https://circuitpython.readthedocs.io/projects/ssd1306/en/latest/>

Hinweis: Fonts sind nicht zwingend notwendig, verbessern aber die Lesbarkeit.



6 Fazit

Das Display wurde erfolgreich in Betrieb genommen. Das Python-Skript zeigt korrekt die Netzwerkdaten an und startet nach der Einrichtung automatisch beim Booten des Raspberry Pi. Die Einbindung des Displays sowie die Netzwerkabfrage über Python verlief ohne größere Probleme.