

#Sonstiges | Raspberry Pi OLED Systemstatus

Was wollen wir?

Eine Statusanzeige der Betriebsdaten unseres Raspberrys

Warum wollen wir das?

Schnelle Übersicht der Systemdaten

Und wie geht das genau?

Welche Hardware benötigen wir für dieses Projekt?

- Einen Raspberry nach Wahl (bei mir laufen Pi 3 und 4 damit)
- Ein OLED Display 128 x 32 z.B. dieses: [Klick mich](#)
- Ein lauffähiges OS mit SSH wird vorausgesetzt

Punkt 1:

Der Anschluss des OLED Display am Raspberry erfolgt wie folgt:

VCC = PIN 1

SDA = PIN 3

SCK = PIN 5

GND = PIN 6

Punkt 2:

Als erstes installieren wir die folgenden Pakete.

Code

```
sudo apt-get install python3-pip
sudo pip3 install adafruit-circuitpython-ssd1306
sudo apt-get install python3-pil
sudo apt-get install -y python-smbus
sudo apt-get install -y i2c-tools
```

Punkt 3:

Nun aktivieren wir in der Raspberry-Config I2C und SPI und starten den Pi neu.

Code

```
sudo raspi-config

Interfacing Options > I2C > Enable (Ja)
Interfacing Options > SPI > Enable (Ja)

sudo reboot
```

Punkt 4:

Nachdem der Pi gestartet ist, verbinden wir uns mit SSH erneut und prüfen, ob das Display erkannt wurde.

Code

```
sudo i2cdetect -y 1
```

Dies sollte dann so aussehen:

Code

```

      0      1      2      3      4      5      6      7      8      9      a      b      c
00:                --      --      --      --      --      --      --      --      --      --      --
10:  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --
20:  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --
30:  --  --  --  --  --  --  --  --  --  --  --  --  3c  --  --  --
40:  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --
50:  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --
60:  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --
70:  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --
```

Punkt 5:

Nun erstellen wir das benötigte Skript:

Code

```
sudo nano /home/pi/stats.py
```

und füllen es mit folgendem Inhalt:

Python

```

# Copyright (c) 2017 Adafruit Industries
# Author: Tony DiCola & James DeVito
#
# Permission is hereby granted, free of charge, to any person obtaining a copy
# of this software and associated documentation files (the "Software"), to deal
# in the Software without restriction, including without limitation the rights
# to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
# copies of the Software, and to permit persons to whom the Software is
# furnished to do so, subject to the following conditions:
#
# The above copyright notice and this permission notice shall be included in
# all copies or substantial portions of the Software.
#
# THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
# IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
# FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
# AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
# LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
# OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN
# THE SOFTWARE.

# This example is for use on (Linux) computers that are using CPython with
# Adafruit Blinka to support CircuitPython libraries. CircuitPython does
# not support PIL/pillow (python imaging library)!

import time
import subprocess

from board import SCL, SDA
import busio
from PIL import Image, ImageDraw, ImageFont
import adafruit_ssdl306

# Create the I2C interface.
i2c = busio.I2C(SCL, SDA)

# Create the SSD1306 OLED class.
# The first two parameters are the pixel width and pixel height. Change these
# to the right size for your display!
disp = adafruit_ssdl306.SSD1306_I2C(128, 32, i2c)

# Clear display.
disp.fill(0)
disp.show()

# Create blank image for drawing.
# Make sure to create image with mode '1' for 1-bit color.
width = disp.width
height = disp.height
image = Image.new("1", (width, height))

# Get drawing object to draw on image.
draw = ImageDraw.Draw(image)

# Draw a black filled box to clear the image.
draw.rectangle((0, 0, width, height), outline=0, fill=0)

# Draw some shapes.
# First define some constants to allow easy resizing of shapes.
padding = -2
top = padding
bottom = height - padding
# Move left to right keeping track of the current x position for drawing shapes.
x = 0

https://ubiquiti-networks-forum.de/wiki/entry/67-sonstiges-raspberry-pi-oled-systemstatus/
4
# Load default font.
font = ImageFont.load_default()

```

Alles anzeigen

Punkt 6:

Als letztes aktivieren wir den Autostart.

Code

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

und tragen über exit 0 folgendes ein:

Code

```
sudo python3 /home/pi/stats.py &
exit 0
```

Nun den Pi neu starten und Ihr erhaltet folgende Anzeige:

7272-82478df6-d433-47e3-981d-c9d40e0d82a2-autoscaled.jpg

Disclaimer: Alle Anleitungen/Tutorials sind nach bestem Wissen und Gewissen verfasst, gehen immer von den definierten Software/Firmware-Versionen aus und sind auf das englische GUI ausgelegt.

Es gibt keine Garantien auf Erfolg. Im Falle eines Misserfolges hilft aber sicherlich die Community hier immer weiter.

Keiner der Autoren oder der Betreiber des Forums ist für die aus der Nutzung resultierenden Probleme/Herausforderungen verantwortlich.

Jegliche hier beschriebenen Schritte erfolgen ausnahmslos in eigener Verantwortung des Durchführenden. Eltern haften für ihre Kinder.

Auswählen: _____

Gültige Software-Version Keine Firmware-Relevanz!