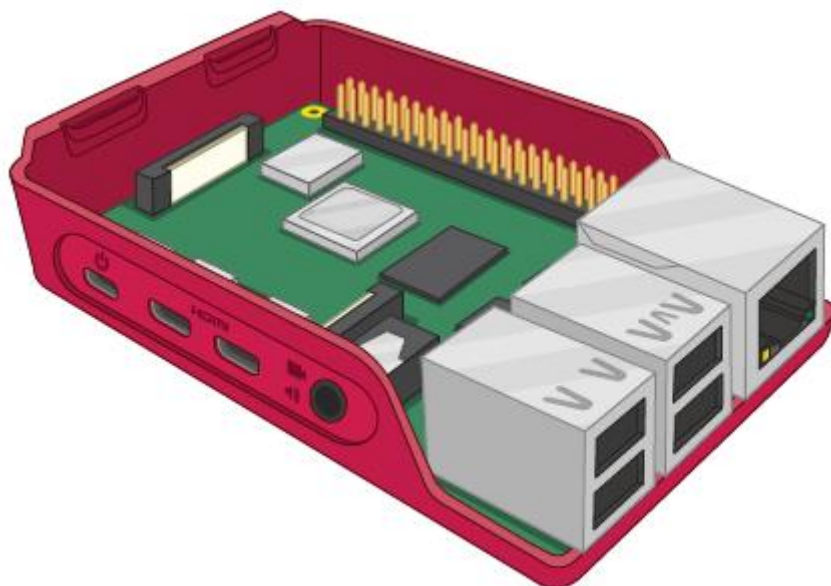


Raspberry Pi

Af: Michael Hansen, Coding Pirates Furesø, 2023, version 0.2

Dokumentet ligger her: <https://github.com/mhfalken/RaspberryPi>



Lidt forskelligt information om hvordan man kan bruge en Raspberry Pi.

Dokumentet er skrevet ud fra en: **Raspberry Pi 4B med 4 GB RAM.**

Der bliver også brugt et **Pi Camera module 3.**

Denne guide bruges også: **The Official Raspberry Pi Beginner's Guide 4th Edition.** Link:

<https://magpi.raspberrypi.com/books/beginners-guide-4th-ed/pdf/download>

(Det er vigtigt at bemærke at kamera delen i dette dokument er deprecated og derfor ikke kan bruges mere – mere her om senere i dokumentet).

1	Setup.....	3
1.1	GUI og Internet.....	3
1.2	Terminalvindue (kommandolinje)	3
1.3	Ping	3
1.4	SSH setup	4
1.5	VNC	4
1.6	Installer en editor (VSC).....	5
2	Hardware programmering i Python.....	5
3	PiCamera2.....	5
3.1	Python.....	6
4	Installer OpenCV.....	6
4.1	Setup virtual environment.....	6
4.2	Installer en masse moduler	7
4.3	Installer selve OpenCV	7
4.4	Test at det virker.....	7
5	CV2 og billede behandling	7
6	CV2 og Pi kamera	7
7	Løsninger	8

1 Setup

Lav et SD kort med et standard Raspberry Pi image på (se guide). Her er brugt en version fra 22/9-22. Sæt kortet i Raspberry Pi'en, tilslut en skærm, tastatur og mus og tænd for strømmen og konfigurer Raspberry Pi'en. Når den kommer til user setup så brug følgende:

User: `pi`

Password: Noget der er nemt at huske.

1.1 GUI og Internet

Kik rundt i GUI'en og se hvordan det virker.

Sørg for at Raspberry Pi'en kommer på internettet, enten med et Ethernet kable eller ved at tilslutte Wi-Fi.

Åben en browser og se at forbindelsen virker.

1.2 Terminalvindue (kommandolinje)

Prøv at åbne et terminal vindue og udfør nogle kommandoer.

```
ls
pwd
df -h
```

Her er en lille liste over kommandoer og deres mening:

Linux kommando	Betydning
ls	List filer
ls -l	List filer med størrelse
pwd	Viser fuld path
df -h	Viser oversigt over filsystem
sudo	Prefix, som angiver at kommandoen skal køre som superuser (root)
ifconfig	List netværksinterfaces, incl. MAC og IP adresse.

1.3 Ping

Når man skal undersøge om man kan 'se' en anden enhed på nettet, så kan man bruge *ping*. *Ping* sender en pakke til modtageren, som så svarer hvis den modtager pakken. (Mange systemer svarer dog ikke på *ping* mere pga. af sikkerhedsproblemer, da det kan udnyttes af hackere.)

Åben et terminal vindue på PC/MAC og prøv at pinge Raspberry Pi'en.

```
ping <ip-adresse>
```

Fx: `ping 192.168.0.152`

IP adressen på Raspberry Pi'en findes ved at bruge `ifconfig` kommandoen og kikke efter *inet* adressen under det aktive interface.

```

ifconfig

eth0: flags=4099<UP,BROADCAST,MULTICAST> mtu 1500
    ether dc:a6:32:1f:fb:59 txqueuelen 1000 (Ethernet)
    RX packets 0 bytes 0 (0.0 B)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 0 bytes 0 (0.0 B)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
    inet 127.0.0.1 netmask 255.0.0.0
    inet6 ::1 prefixlen 128 scopeid 0x10<host>
    loop txqueuelen 1000 (Local Loopback)
    RX packets 26 bytes 2675 (2.6 KiB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 26 bytes 2675 (2.6 KiB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

wlan0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 192.168.0.152 netmask 255.255.255.0 broadcast 192.168.0.255
    inet6 fe80::d44:b0c9:a7b3:7da2 prefixlen 64 scopeid 0x20<link>
    ether dc:a6:32:1f:fb:5a txqueuelen 1000 (Ethernet)
    RX packets 4347 bytes 319520 (312.0 KiB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 11780 bytes 14196714 (13.5 MiB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

```

1.4 SSH setup

Vi skal nu åbne et terminal vindue på PC/MAC, som kører på Raspberry Pi'en, dvs vi har et vindue som er synligt på PC/MAC men som kører på Raspberry Pi'en. På den måde kan vi udføre kommandoer direkte på Raspberry Pi'en fra vores PC/MAC.

Først skal vi have åbnet for adgangen på Raspberry Pi'en:

```
sudo raspi-config
```

Interface Options-> enable SSH

Åben en Windows Powershell (Windows) eller en Terminal (MAC):

```
ssh pi@<ip address>
```

Fx: ssh pi@192.168.0.152

Man kan nu udføre kommandoer direkte på Raspberry Pi'en fra terminalen. Prøv og se at det virker.

1.5 VNC

Vi skal nu et skridt videre med remote adgang og have adgang til hele Raspberry Pi skærmen på PC/MAC (virtual desktop), så man slipper for at have en ekstern skræm og tastatur.

Der skal først åbnes for adgang:

```
sudo raspi-config
```

Interface Options->VNC

Installer VNC viewer på PC/MAC: <https://www.realvnc.com/en/connect/download/viewer/>

Start VNC Viewer og connect til IP adressen på Raspberry Pi'en. Man skal nu gerne kunne se det samme billede som på Raspberry Pi'ens skærm.

Hvis man startet Raspberry Pi'en op uden skærm, så får den en default opløsning. Den kan ændres med:

```
sudo raspi-config
```

Display Options->VNC Resolution.

Efter man har sat en ny opløsning, så skal Raspberry Pi'en rebootes.

Hvis skærmopdateringen virker lidt langsomt, så kan det være en fordel at fjerne baggrundsbilledet.

1.6 Installer en editor (VSC)

Hvis man skal skrive kode, kan det være godt at have en god editor. Raspberry Pi'en har nogle mindre editorer installeret, men jeg vil anbefale at installere VSC også.

Brug GUI og vælg: **Recommended software**

Find **Visual Studio Code** og sæt kryds ud for den for at installere den.

Når programmet er installeret findes det i menuen. Start det op og se at det virker.

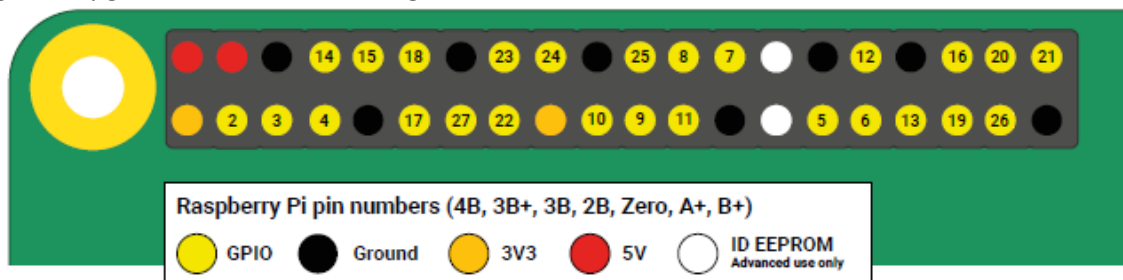
Prøv også at starte de fra et terminalvindue med følgende kommando:

```
code
```

2 Hardware programmering i Python

Brug den officielle guide kapitel 6: *Physical computation with Scratch and Python*. Vi er her kun interesseret i Python delen.

Løs følgende opgaver ved at kikke/læse i guiden.



1) Find en LED (lysdiode – lille pære) og tilslut den til Raspberry Pi'ens hardware GP25 pin.

- Man må gerne tilslutte en LED/kontakt mens der er strøm på Raspberry Pi'en.

2) Lav et Python program som får LED'en til at blinke. Kald det *hwledblink.py*.

3) Find en knap og forbind den til GP2

- Ground ledningen skal 'bare' forbindes til en Ground pin (Sort på billedet) på boardet. Det er vigtigt at det gøres korrekt, da man ellers kan kortslutte printet og potentielt ødelægge det!

4) Lav et nyt program og kald det *hwbutton.py*.

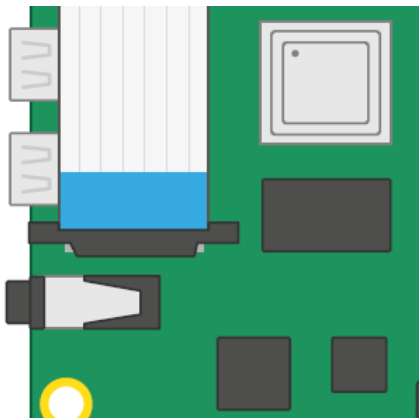
5) Få knappen til at styre LED'en, så den skiftevis tænder og slukker når man trykker på den

6) Prøv at holde knappen inden og se hvad der sker? Hvis LED lyser svagere, så overvej hvorfor og løs problemet med følgende linje:

```
button.wait_for_release()
```

3 PiCamera2

Sluk for strømmen og forbind et Raspberry Pi kamera til Raspberry Pi boardet som vist på nedenstående billede.



Den officielle guide omkring kameraet virker desværre ikke mere, da deres kode er blevet deprecated ...!

Tænd for boardet igen og test at kameraet virker med følgende kommando:

```
libcamera-still -o test.jpg
```

Der skal gerne komme et vindue frem som viser billedet fra kameraet i nogle sekunder, hvorefter vinduet lukker igen.

3.1 Python

Lav et Python program med følgende linjer:

```
from picamera2 import Picamera2, Preview
import time

picam2 = Picamera2()
camera_config = picam2.create_preview_configuration()
picam2.configure(camera_config)
picam2.start_preview(Preview.QTGL)
picam2.start()
```

Kør programmet og se at det virker.

Man kan starte og stoppe kameraet med følgende linjer:

```
picam2.stop()
picam2.stop_preview()
```

Den første stopper kameraet og den anden lukker vinduet.

1) Få programmet til at vise kameraets preview et par sekunder og luk derefter vinduet ned.

Man laver et delay med følgende linje:

```
time.sleep(2)
```

2) Få en fysisk knap til at tænde og slukke for kameraet.

Man kan få inspiration til koden ved at kikke i den officielle guide under kapitel 8: *Camera Module*. Husk at koden kan ikke bruges direkte!

4 Installer OpenCV

Vi skal nu have installeret Open CV. Open CV er et billede 'behandlingsprogram', som kan bruges til mange ting bla. at finde ansigter og øjne med meget mere i en video.

Følg denne video for at se hvordan man installerer det:

<https://www.youtube.com/watch?v=QzVYnG-WaM4>

Det er vigtigt at man kun udfører de kommandoer som står i dette dokument, da hans video desværre ikke helt passer til vores setup.

Åben en terminal på PC/MAC og udfør følgende kommandoer (som er forklaret i videoen). Kan man ikke få en terminal på PC/MAC til at fungere, kan man bruge en terminal direkte på Raspberry Pi'en, men så er det sværere at kopi/paste.)

4.1 Setup virtual environment

```
sudo apt-get install python3-pip python3-virtualenv
```

Vi kalder vores projekt *opencv*

```
mkdir opencv
cd opencv
```

Setup og aktiver det virtuelle miljø.

```
virtualenv env --system-site-packages
source env/bin/activate
```

4.2 Installer en masse moduler

Tager ca. 2 minutter.

```
sudo apt install -y build-essential cmake pkg-config libjpeg-dev
libtiff5-dev libpng-dev libavcodec-dev libavformat-dev libswscale-dev
libv4l-dev libxvidcore-dev libx264-dev libfontconfig1-dev libcairo2-
dev libgdk-pixbuf2.0-dev libpango1.0-dev libgtk2.0-dev libgtk-3-dev
libatlas-base-dev gfortran libhdf5-dev libhdf5-serial-dev libhdf5-103
libqt5gui5 libqt5webkit5 libqt5test5 python3-pyqt5 python3-dev
```

4.3 Installer selve OpenCV

Tager ca. 90 minutter, så start kun hvis der er tid nok tilbage, da man ikke må slukke for strømmen undervejs!

```
pip install opencv-python
```

4.4 Test at det virker

Start Python

```
python
```

Skriv følgende linjer i Python:

```
import cv2
cv2.__version__
```

Nu skal den gerne skrive '4.7.0', eller noget 'højere'.

Tast **Ctrl-D** for at komme ud af Python igen.

5 CV2 og billede behandling

Her er det lettest at arbejde direkte på RPi'en, dvs uden at bruge ssh, da vi har brug for flere vinduer som kan 'poppe' op.

Åben et terminal vindue og tast følgende for at komme ind i vores virtuelle OpenCV miljø.

```
cd project
source env/bin/activate
```

Jeg bruger generelt videoer fra ham her:

<https://www.youtube.com/@SamWestbyTech/videos>

6 CV2 og Pi kamera

```

# import the opencv library
import cv2
from picamera2 import Picamera2, Preview

# define a video capture object
picam2 = Picamera2()
camera_config = picam2.create_preview_configuration()
picam2.configure(camera_config)
picam2.start()

while(True):
    # Capture the video frame by frame
    frame = picam2.capture_array()

    # Display the resulting frame
    cv2.imshow('frame', frame)

    # the 'q' button is set as the quitting button
    if cv2.waitKey(1) & 0xFF == ord('q'):
        break

# After the loop release the cap object
picam2.stop()
# Destroy all the windows
cv2.destroyAllWindows()

```

7 Løsninger

hwbutton.py:

```

from gpiozero import LED, Button
from time import sleep

led = LED(25)
button = Button(2)

lastState = False

while True:
    button.wait_for_press()
    if lastState:
        led.off()
    else:
        led.on()
    lastState = not lastState
    button.wait_for_release()

```

hwbuttoncam.py:


```
from picamera2 import Picamera2, Preview
from gpiozero import LED, Button
import time

picam2 = Picamera2()
camera_config = picam2.create_preview_configuration()
picam2.configure(camera_config)

button = Button(2)
lastState = False

while True:
    button.wait_for_press()
    if lastState:
        picam2.stop()
        picam2.stop_preview()
    else:
        picam2.start_preview(Preview.QTGL)
        picam2.start()
        lastState = not lastState
    button.wait_for_release()
```