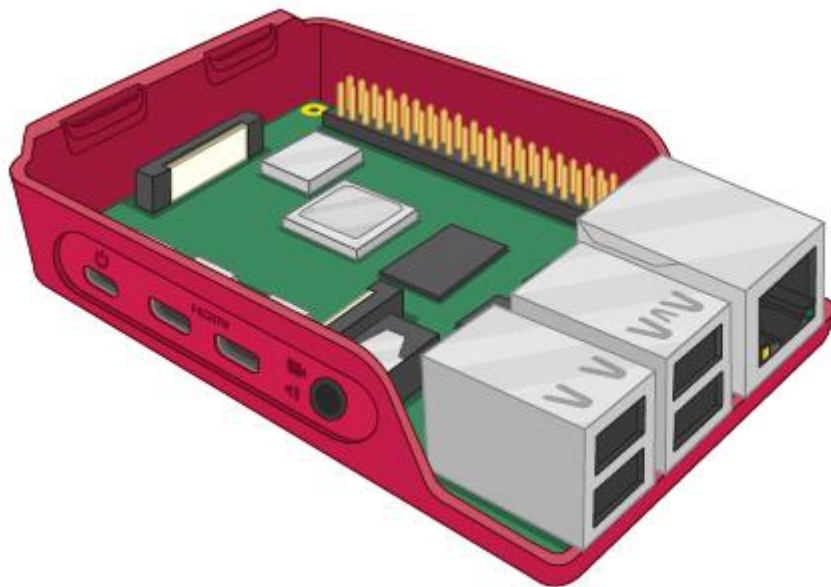


# Raspberry Pi

Af: Michael Hansen, Coding Pirates Furesø, 2023, version 0.8

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 kameradelen 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	Ifconfig.....	3
1.4	Ping .....	4
1.5	SSH setup.....	5
1.6	VNC .....	5
1.7	Samba .....	5
1.8	Installer en editor (VSC).....	6
2	Hardware programmering i Python.....	6
3	PiCamera2.....	7
3.1	Python.....	7
4	Installer OpenCV.....	8
4.1	Setup virtual environment.....	8
4.2	Installer en masse moduler .....	8
4.3	Installer selve OpenCV.....	8
4.4	Test at det virker.....	8
5	CV2 og Pi kamera.....	9
5.1	Vis kamera billede .....	9
5.2	Tegn en firkant.....	9
5.3	Find ansigter .....	10
5.4	Find øjne .....	10
6	Løsninger .....	10

# 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
df -h	Viser oversigt over filsystem
pwd	Viser fuld path
ifconfig	List netværksinterfaces, incl. MAC og IP adresse.
sudo	Prefix, som angiver at kommandoen skal køres som superuser (root)
shutdown now	Lukker systemet ned, så man kan fjerne power.

## 1.3 Ifconfig

Vi skal nu til at kommunikere med Raspberry Pi'en over internettet og vi har derfor brug for at finde dens IP adresse. IP adressen er dens adressen ude på internettet.

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

```

I overstående tilfælde er Raspberry Pi'en forbundet via Wi-Fi interfacet og har fået IP adresse 192.168.0.152. Man kan også se hvor mange pakker den har modtaget (RX) og sendt (TX). Det kan bruges til at debugge med, hvis man ikke kan få forbindelsen til at virke.

## 1.4 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.)

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

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

Man skal gerne få et resultat ala følgende:

```

ping 192.168.0.152

Pinging 192.168.0.152 with 32 bytes of data:
Reply from 192.168.0.152: bytes=32 time=85ms TTL=64
Reply from 192.168.0.152: bytes=32 time=4ms TTL=64
Reply from 192.168.0.152: bytes=32 time=3ms TTL=64
Reply from 192.168.0.152: bytes=32 time=3ms TTL=64

Ping statistics for 192.168.0.152:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 3ms, Maximum = 85ms, Average = 23ms

```

I overstående tilfælde sender den 4 pakker som alle er modtaget. Hvis der er forbindelse er det normalt at man ikke taber nogle pakker. Man kan også se hvor langt tid det har taget at få pakkerne frem og tilbage. Disse tal afhænger typisk af om man er forbundet via på Wi-Fi eller Ethernet kabel. Prøv for sjov at pinge noget på internettet fx DR.

```
ping www.dr.dk
```

PC/MAC kan selv finde ud af at oversætte [www.dr.dk](http://www.dr.dk) til en IP adresse. Det er ikke sikkert at Raspberry Pi'en kan det, men så kan man bare indtaste IP adresse direkte.

## 1.5 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.

## 1.6 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.

## 1.7 Samba

Samba er en mulighed for at kunne tilgå filerne på Raspberry Pi'en fra en anden PC. På den måde kan man bruge en editor på sin PC til at rette i filerne direkte på Raspberry Pi'en.

Der er brugt denne vejledning: <https://pimylifeup.com/raspberry-pi-samba/>

Åben et terminal vindue på Raspberry Pi'en og skriv følgende for at installere Samba softwaren.

```
sudo apt-get install samba samba-common-bin
```

Man skal nu tilføje det sted som man ønsker adgang til. Det gør man ved at ændre i følgende fil.

```
sudo nano /etc/samba/smb.conf
```

Tilføj disse linjer i *bunden* af filen:

```
[pi_share]
path = /home/pi
writeable=Yes
browseable=Yes
create mask=0777
directory mask=0777
public=no
```

For at gemme filen og komme tilbage tryk: **ctrl-x -> y -> <enter>**

Nu skal man oprette en Samba bruger og password.

```
sudo smbpasswd -a pi
```

Indtast password (2 gange). Brug gerne samme password som til Raspberry Pi login.

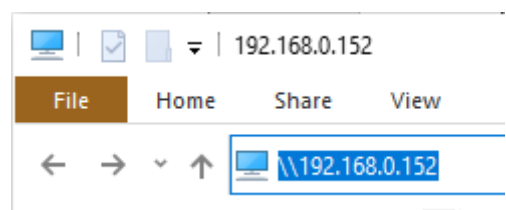
Reset Samba serveren for at aktivere de nye indstillinger:

```
sudo systemctl restart smbd
```

På en Windows PC åbner man fil explorer og taster følgende ind i adressefeltet i toppen (se billede)

```
Fejl! Linkreferencen er ugyldig. adresse>
```

Hvor <IP adresse> er IP adressen på Raspberry Pi'en.



Indtast **pi** og <password>.

Du skulle nu kunne se filerne på Raspberry Pi'en. Det er vigtigt at man vælger den folder der hedder **pi\_share** når man tilgår filerne.

## 1.8 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.

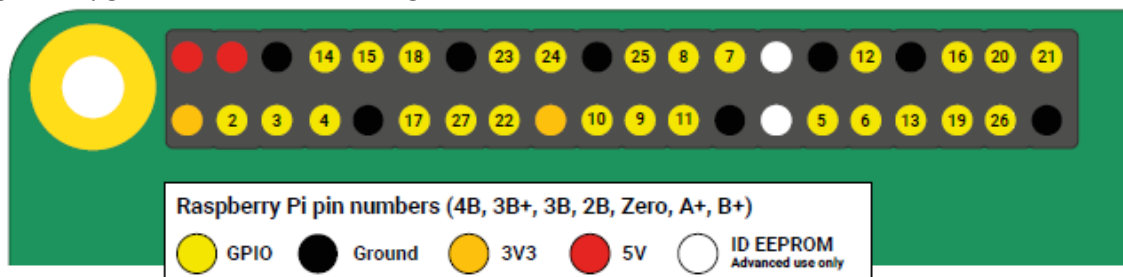
Prøv også at starte de fra et terminalvindue med følgende kommando:

```
code
```

## 2 Hardware programmering i Python

Brug den officielle guide kapitel 6: *Physical computation with Scratch and Python* (link i toppen af dokumentet). 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.

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

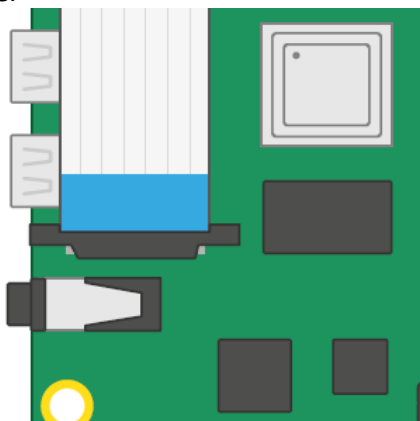
5) Få knappen til at styre LED'en, så den skiftevis tænder og slukker når man trykker på den

6) Prøv at holde knappen inde og se hvad der sker? Hvis LED'en 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 OpenCV. OpenCV er et billede 'behandlingsprogram', som kan bruges til mange ting bla. at finde ansigter og øjne med meget mere i en video.

Se følgende video (den varer kun 6 minutter) for at se hvordan man installerer OpenCV:

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

Installer nu OpenCV ved at lave følgende steps, som følger videoen, men med få ændringer så det virker på vores system.

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.

Husk at hver gang man åbner et nyt terminal vindue på Raspberry Pi'en som skal køre Open CV, så skal man køre følgende kommandoer (ellers virker det ikke):



```
cd opencv
source env/bin/activate
```

## 5 CV2 og Pi kamera

Vi skal nu bruge OpenCv og et kamera. Følgende er lavet med et Raspberry Pi kamera modul 3. Andre kameraet vil virke, men så skal koden ændres lidt.

For at bruge OpenCv skal man først åbne et terminal vindue og så indtaste følgende kommandoer:

```
cd opencv
source env/bin/activate
```

### 5.1 Vis kamera billede

Indtast følgende kode i en Python fil og kald den *cvtest.py*.

```
# 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
    key = cv2.waitKey(1)
    if key == ord('q'):
        break

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

For at køre koden skal man skrive:

```
python cvtest.py
```

Man skulle nu gerne kunne se kamerabilledet på skærmen. For at lukke billedet skal man trykke på 'q'.

### 5.2 Tegn en firkant

Vi skal nu bruge en af featuren i CV til at tegne en firkant oven på billedet.

Tilføj følgende linjer efter `'frame = picam2 ...'`:

```
faces = [(10, 10, 30, 30)]
for (x, y, w, h) in faces:
    cv2.rectangle(frame, (x, y), (x + w, y + h), (255, 0, 0), 2)
```

Kør programmet og nu skulle der gerne komme en firkant frem på skærmen.

- 1) Ændre i værdierne i `faces` listen. (De 2 første er x, y og de 2 næste er bredde, højde)
- 2) Farven på firkanten er styret af de **røde** tal. Prøv at ændre farven af firkanten. Farven er (B, G, R)!
- 3) Tilføj en tuple mere i `faces` listen med nogle andre værdier i og se hvad der sker.

## 5.3 Find ansigter

Vi skal nu få CV til at finde ansigter i billedet.

Tilføj følgende linjer:

Denne linje skal stå efter import linjerne i toppen af filen:

```
face_cascade = cv2.CascadeClassifier('haarcascade_frontalface_default.xml')
```

(Det kræver at man har en fil som hedder *haarcascade\_frontalface\_default.xml* liggende lokalt).

Indsæt disse 2 linjer i stedet for `faces = [...]` – dvs `faces = []` skal slettes.

```
gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)
faces = face_cascade.detectMultiScale(gray, 1.3, 5)
```

Kør programmet og kik ind i kameraet. Nu skulle den gerne tegne en firkant rundt om ens ansigt. Prøv at få flere til at kikke ind i kameraet og se at den kan finde flere ansigter på en gang.

## 5.4 Find øjne

Vi skal nu have CV til at finde øjne i ansigterne.

Tilføj følgende linje efter `face_cascade = ...`:

```
eye_cascade = cv2.CascadeClassifier('haarcascade_eye.xml')
```

(Det kræver at man har en fil som hedder *haarcascade\_eye.xml* liggende lokalt).

Ændre koden til følgende (linjer med **fed** er de nye):

```
for (x, y, w, h) in faces:
    cv2.rectangle(frame, (x, y), (x + w, y + h), (255, 0, 0), 2)
    roi_gray = gray[y:y + h, x:x + w]
    roi_color = frame[y:y + h, x:x + w]

    eyes = eye_cascade.detectMultiScale(roi_gray)
    for (ex, ey, ew, eh) in eyes:
        cv2.rectangle(roi_color, (ex, ey), (ex + ew, ey + eh), (0, 255, 0), 2)
```

Kør programmet og se at den kan finde både ansigter og øjne i billedet.

# 6 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()
```

*cvtest.py:*

```

import cv2, time
from picamera2 import Picamera2, Preview

face_cascade = cv2.CascadeClassifier('haarcascade_frontalface_default.xml')
eye_cascade = cv2.CascadeClassifier('haarcascade_eye.xml')

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

while True:
    frame = picam2.capture_array()

    gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)
    faces = face_cascade.detectMultiScale(gray, 1.3, 5)
    #faces = [(10, 10, 30, 30)]
    for (x, y, w, h) in faces:
        cv2.rectangle(frame, (x, y), (x + w, y + h), (255, 0, 0), 2)
        roi_gray = gray[y:y + h, x:x + w]
        roi_color = frame[y:y + h, x:x + w]

        eyes = eye_cascade.detectMultiScale(roi_gray)
        for (ex, ey, ew, eh) in eyes:
            cv2.rectangle(roi_color, (ex, ey), (ex + ew, ey + eh), (0, 255, 0), 2)

    cv2.imshow("Frame", frame)

    key = cv2.waitKey(1)
    if key == ord('q'):
        break

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

```