

Covid-19 Infostander

tVÆRFAGLIGT PROJEKT

Simon Locht Nørregaard, Anders Krog, Nikolai Winther Juhl, Nicolai Garro Throksø, Frederik Teddy Fly | 18-11-2020

Contents

[Case 1](#_Toc56670450)

[Fremgangsmåde 2](#_Toc56670451)

[Materiel 2](#_Toc56670452)

[Værktøj 3](#_Toc56670453)

[Budget 3](#_Toc56670454)

[Opsætning af styresystem på Raspberry PI 4](#_Toc56670455)

[Frisk installation af Raspberry PI OS 4](#_Toc56670456)

[Opsætning af Raspberry PI OS 5](#_Toc56670457)

[Standardopsætning 5](#_Toc56670458)

[Installation af Chromium og Openbox 5](#_Toc56670459)

[Openbox opsætning 5](#_Toc56670460)

[Autologin 6](#_Toc56670461)

[Installation af lokal server 7](#_Toc56670462)

[Tilføje af slides til systemet 9](#_Toc56670463)

[Overføre filer via USB 9](#_Toc56670464)

[Opret mount 9](#_Toc56670465)

[Læse USB på Raspberry PI 10](#_Toc56670466)

[Mount 10](#_Toc56670467)

[Dokumentation af koden 10](#_Toc56670468)

[Projektforløb 11](#_Toc56670469)

[Kildelist 12](#_Toc56670470)

# Case

Opret en infoskærm med én skærm der kan vise slides automatisk. Vi fik at vide at vi skulle anvende Raspberry PI 3 og Philips-skærme, og at de skulle fungere som infostandere med grafisk info om tiltag ift. Covid-19. Den skal kunne bruges uden inputs fra keyboard. Den skal bare sættes til strøm hvorefter skærmen så skal vise slides.

# Fremgangsmåde

Vi har valgt at installere Raspberry PI OS uden Desktop for at spare kræfter og gøre maskinerne så effektive som muligt. Vi har oprette en server (Apache2) direkte på vores maskine. Vi har derefter installeret Chromium som skal køre i Kiosk Mode, dvs. at Chromium skal køre i fuld skærm og uden brugerinputs. Chromium er blevet sat til at åbne URL ”http://localhost”. På localhost-adressen viser Apache2-serveren en HTML-side, med billeder i slides, som er programmeret med JavaScript. Billederne eller filerne har vi overført til vores maskine via et USB-medie. Til sidst har vi med lidt kommentarer skrevet om hvad man kan gøre af variationer og hvilke problemer man kan støde ind i.

# Materiel

Maskinen

* Hukommelse
  + Micro SD. Minimum 4GB
* Kortlæser
  + Micro SD/SD til USB
* Raspberry PI
  + Raspberry PI Model B+
* Strømforsyning
  + Micro USB 5V 2.5A
* Kabinet
* Skærm
  + Full HD
* Kabel
  + HDMI

Værktøj

* Kablet internet
* Tastatur
* USB-medie

#### Hukommelse

Den samlede størrelse for hele installationen er under 3GB. Der følger et 16GB SD-kort med når man køber en Raspberry PI-startpakke. Dette burde være nok til alle kioskprojekter.

#### Kortlæser

Den skal bruges til at overføre styresystemet via en computer, så man skal bruge noget der kan overføre fra Micro SD til computeren.

#### Raspberry PI

Skal bruge Raspberry PI 3 model B+.

Model A+ har ikke Ethernet og kun én USB-port.

#### Skærm

Skærmen skal være 1920\*1080. Hvis anden opløsning ønskes skal CSS ændres.

Vi har valgt en [BenQ 22" skærm BL2283](https://www.komplett.dk/product/1140492/hardware/skaerme/skaerme/benq-22-skaerm-bl2283). IPS-panel er at foretrække fordi billedet er generelt pænere og læsevinklen er god. Dens backdraw er opdateringshastighed og Hz er irrelevant for vores projekt.

#### Kabel

HDMI til HDMI eller HDMI TIL DVI. Kablet skal helst ikke være for langt.

#### Strømforsyning

Man kan bruge en hvilken som helst strømforsyning så længe den har Micro USB og er på 5V 2.5A.

### Værktøj

#### Ethernet kabel og kablet internet

Vi bliver nødt til at have internet under installationen af maskinen. Vi skal installere Chromium og Apache.

#### Tastatur

Ja.

#### USB-medie

Vi skal kunne overføre vores ønskede slides til maskinen, så vi bliver nødt til at have en nøgle.

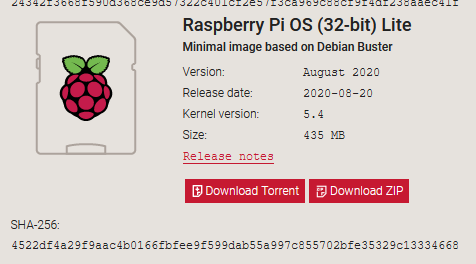
# Budget

|  |  |  |
| --- | --- | --- |
| Produkt | Pris u. moms | Pris m. moms |
| [BenQ 22"](https://www.komplett.dk/product/1140492/hardware/skaerme/skaerme/benq-22-skaerm-bl2283) | 633,6 | 792 |
| Raspberry PI 3 Starter Kit | 469,6 | 587 |
| HDMI-kabel 1m | 31,20 | 39 |
| Stikdåse min. 2 stik | 11,60 | 14,50 |
| I alt | 1.250,4kr | 1432,5kr |

# Opsætning af styresystem på Raspberry PI

## Frisk installation af Raspberry PI OS

Hent den nyeste version af Raspberry PI OS fra <https://www.raspberrypi.org/downloads/raspberry-pi-os/> Det er vigtigt at Man vælger versionen uden Desktop (Raspberry PI OS Lite)



Hent et program som kan installere dit operativsystem på dit SD-kort. Vi bruger et program kaldet Rufus i dette eksempel. Et alternativ til Linuxbruger kunne være Etcher.

<https://rufus.ie/>

Find en måde at læse dit Micro SD-kort på computeren. Dette kunne gøres med en kortlæser, eller indbygget kortlæser. Åben Rufus og vælg dit SD-kort og Raspberry PI OS Lite.  Klik start.

## Opsætning af Raspberry PI OS

Lad operativsystemet starte op. Når du bliver spurgt om brugernavn og password, så er standardbrugernavnet “**pi**” og kodeordet er “**raspberry**”. Hvis du er i tvivl om den fysiske opsætning kig på: <https://projects.raspberrypi.org/en/projects/raspberry-pi-setting-up>

## Standardopsætning

Systemet starter med **amerikansk keyboard** så det ville være smart at ændre dette til dansk.

Dette gøres ved at ændre på en config-fil. Kommandoen for at åbne filen er:

**sudo nano /etc/default/keyboard**

I filen står der XKBLAYOUT=”gb” et sted. **Ændre dette til XKBLAYOUT=”dk”.**

*Det kan være at man skal reboote systemet for at det virker*

Det er en god vane at skrive kommandoerne

**apt-get update**

**apt-get upgrade**

Disse kommandoer opdatere dit system med de nyeste softwarepakker.

# Installation af Chromium og Openbox

Vi skal kun køre én grafisk applikation, så der er ingen grund til at installere en hel desktop.

Vi skal kun bruge XServer og en window manager kaldet Openbox.  Disse installeres med denne kommando:

**sudo apt-get install --no-install-recommends xserver-xorg x11-xserver-utils xinit openbox**

Det næste som skal installeres, er Chromium. Indtast denne kommando i terminalen.

**sudo apt-get install --no-install-recommends chromium-browser**

## Openbox opsætning

Openbox skal konfigurere og starte Chromium. Vi sætter den til at starte i kiosk mode, sørge for at skærmen forbliver tændt, sætter en måde at afslutte Chromium på og henviser til en URL, der skal vises.

For at kunne konfigurere filen skal man skrive kommandoen:

**sudo nano /etc/xdg/openbox/autostart**

Linjer med # betyder at de er udkommenteret, så disse linjer behøves ikke at indtastes. I filen skal du skrive disse ting:

**# Disable any form of screen saver / screen blanking / power management**

**xset s off**

**xset s noblank**

**xset –dpms**

**# Allow quitting the X server with CTRL-ATL-Backspace**

**setxkbmap -option terminate:ctrl\_alt\_bksp**

**# Start Chromium in kiosk mode**

**sed -i 's/"exited\_cleanly":false/"exited\_cleanly":true/' ~/.config/chromium/'Local State'**

**sed -i 's/"exited\_cleanly":false/"exited\_cleanly":true/; s/"exit\_type":"[^"]\+"/"exit\_type":"Normal"/' ~/.config/chromium/Default/Preferences**

**chromium-browser --disable-infobars --kiosk '**[**http://dr.dk**](http://dr.dk)**'**

**Tryk nu** **CTRL+O** (Write Out)

For at tjekke om det virker skriv:

**startx**

Chromium burde nu start og vise den URL vi har indtastet. Dr.dk i dette eksempel. Tryk CTRL-ATL-Backspace for at afslutte. OBS. Dr.dk er selvølgelig bare en placeholder. Vi skifter det ud med <http://localhost> senere.

# Autologin

Det er vigtigt at man får maskinen til at køre med **autologin.** Vi skal ændre/oprette en fil kaldet **.bash\_profile**. Skriv kommandoen:

**sudo nano /home/pi/.bash\_profile**

I filen skal vi skrive denne kommando:

**[[ -z $DISPLAY && $XDG\_VTNR -eq 1 ]] && startx -- -nocursor**

**tryk CTRL+O**(Write out)

*OBS. Dette kan være en ny fil*

Som det næste skal vi logge ind på Raspberry PI’s window manager  menu

**sudo raspi-config**

I menuen der åbnes tryk system og autologin med bruger PI

I config-filen skal der skrives. Det kan være en ny fil der skal oprettes.

**Klik CTRL+O** (Write out)

# Installation af lokal server

Vi bruger apache2 til at hoste vores server. Dette program installeres og startes med disse kommandoer:

**sudo apt install apache2**

**sudo systemctl is-active apache2**

**sudo systemctl is-enabled apache2**

**sudo systemctl status apache2**

Enabled-funktionen skal bruges for at serveren også kører efter genstart. Og status for at se om den kører.

#### Openbox

For at få Chromium til at åbne en URL på Apache2-serveren skal vi ændre i Openbox autostart-filen igen. Skriv kommandoen:

**sudo nano /etc/xdg/openbox/autostart**

Ændre nu linjen:

**chromium-browser --disable-infobars --kiosk '**[**http://dr.dk**](http://dr.dk)**'**

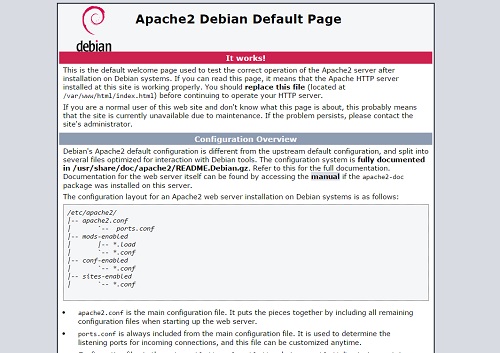
Til

**chromium-browser --disable-infobars --kiosk '**[**http://localhost/**](http://localhost/)**'**

**Tryk nu CTRL+O** (Write out) for at gemme ændringen

For at tjekke om det virker skriv kommandoen:

**startx**



Siden skal se sådan her ud. Dette er standardsiden som Apache2 viser når man kalder localhost.

Tryk **CTRL-ATL-Backspace** for at afslutte.

*OBS. Tryk på F5 for at refreshe chromebrowseren hvis apache-testsiden ikke vises.*

# Tilføje af slides til systemet

## Overføre filer via USB

For at lægge dine billeder på maskinen skal man overføre dem fra et USBmedie. De har en navnestandard, da scriptet er kodet sådan. Hvis der ønskes anden mængde billeder til at søge efter billeder med specifikke navne. Tallene bestemmer rækkefølgen. Hvis anden antal filer ønskes, læs html delen (s.8). Det kan være nemmere at finde dit medie, hvis du navngiver det til noget unikt. Vi har kaldet vores “KAT”. Vi kommer med eksemplet hvorfor, under *Læse USB på Raspberry PI*.

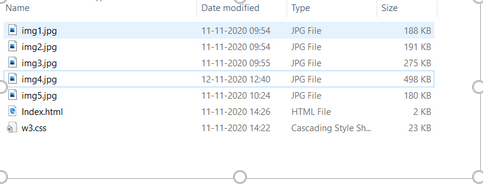
De skal hedde:

* **Img1.jpg**
* **Img2.jpg**
* **Img3.jpg**
* **Img4.jpg**
* **Img5.jpg**

*OBS. Hvis man ikke ændre i kode:*

*Man skal tilføje præcist 5 billeder og html filen skal hedde index.html med lille “i”. Dette er grundet at Linux er case sensitive og at der er en original-fil der skal overskrives når man overfører filerne til PI’en!*

*Ved opdatering skal man være opmærksom på at CSS-filen er cached dvs. At man skal kalde på den, på ny, ved at refreshe filen.*



Tips. Hvis man har færre billeder, som så kan man jo lave flere kopier af samme billede og kalde dem noget forskelligt.

## Opret mount

Man skal mounte sit USB-medie. For at kunne gøre dette skal man oprette en mappe til formålet. Skriv kommandoen:

**sudo mkdir /media/USB**

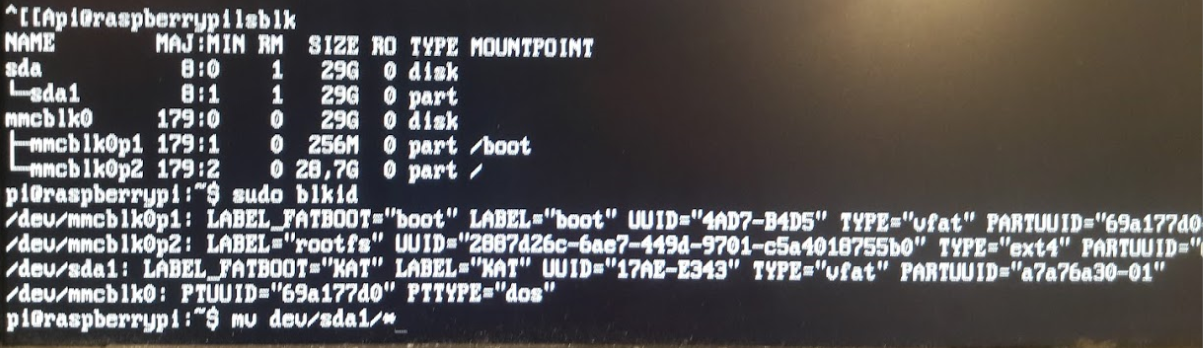
## Læse USB på Raspberry PI

Vi skal først finde stien og navnet på vores USB-medie. Skriv kommandoen:

**lsblk**

Nu får du en liste over drev og partitioner. Skriv nu kommandoen:

**sudo blkid**



Stien på dit USB-medie er noget lignende **/dev/sda1.** LABEL er navnet på dit drev. Vores medie hedder “KAT” i dette eksempel. Vi har kaldt den KAT, så det er nemt at finde stein, som vi skal bruge.

*Kommandoerne er kun vejledende. Den egentlige sti kan være anderledes*

## Mount

For at kunne bruge KAT drevet skal vi ankre det på vores PI. Dette gøres ved at henvise til en sti på PI’en. Skriv kommandoen:

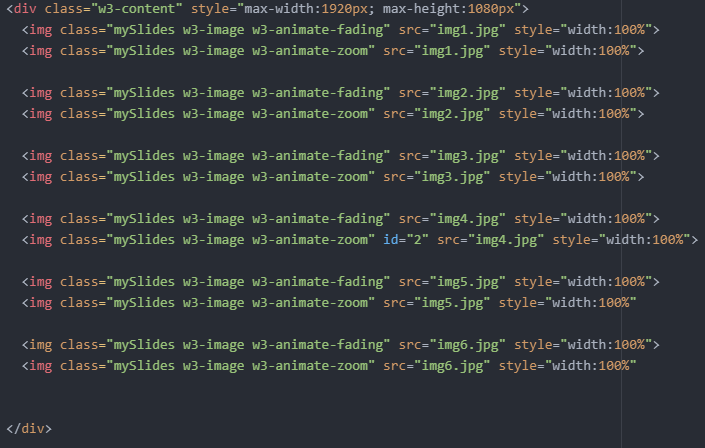
# Dokumentation af koden

Til at lave selve websiden, bruger vi en html-fil (index.html) der også indeholder et JavaScript og et stylesheet (w3.css). Stylesheetet kan hentes på w3schools.com og indeholder en række klasser med forskellige standarder og animationseffekter. Vi bruger de animerede effekter zoom og fade



Vi sætter max-størrelse til 1920x1080

Vi laver en div klasse med alle billederne. Vi anvender klasserne med effekterne. Vi sætter filerne ind to gange, da man kun kan anvende en effektklase af gangen. Resultatet er at der først fades og dernæst zoomes.



JavaScriptdelen ser således ud



Da vi ønsker at lave en dissolve mellem de to billeder, skal vi have vist to billeder af gangen. Vi opretter to variabler til at angive hvilket billede der vises og det næste billede der skal vises.

Vi opretter et array over de billeder i klassen mySlides

Vi opretter dernæst et loop som starter forfra hvert 5. minut

Inde i det kører vi et for-loop der skjuler alle andre billeder end det nuværende og det næste

Dernæst lægger vi en til variablerne der angiver det nuværende og næste billede, og tjekker efter om de variabler er større end arrayets længde -1 (pga. 0 index), og nulstiller dem hvis de er.

I CSS-filen har vi ændret standartstørrelsen på billederne, og justeret fade og zoom

C:\Users\niju\AppData\Local\Microsoft\Windows\INetCache\Content.Word\css3.png



# Krav

Alle billederne skal være i “landskab”(almindelig skærmforhold, vandret). Hvis portræt(højkant) ønskes, så skal både CSS og JavaScript ændres. Alle billeder SKAL være i samme opløsning. Standarden er 1920\*1080. For at ændre opløsningen, så skal CSS og JavaScript filerne også ændres. Det samme gælder antallet af billeder

# Projektforløb

Beskrivelse af forløb om Covid-infoskærme.

Vi meldte os frivilligt til at løse opgaven. Følgende personer på P2 datatekniker med programmering som speciale: Simon Locht Nørgaard, Anders Krog, Nikolai Winther Juhl, Nicolai Garro Throksø, Frederik Teddy Fly under instruktør Steve Jørgensen.

Vi fik at vide at vi skulle anvende raspberry pi 3 og Philips-skærme, og at de skulle fungere som infostandere med grafisk info om tiltag ift. Covid-19.

Men selve opsætningen og setup-et på maskinerne skulle vi selv finde en løsning på. Vi fandt hurtigt frem til at der er et koncept der hedder ’kioskmode’, som er lavet til netop at opsætte infoskærme. Vi fik anbefalet at anvende en grafisk version af Raspberry OS, men besluttede os for at anvende Raspberry Pi OS lite, da det er en nedskaleret udgave, som ikke optager så meget plads og vi mente at kunne løse problemet udelukkende fra et konsolmiljø, det ville også have den gavnlige sideeffekt, at det var sværere for folk udefra at pille ved maskinerne. Vi benyttede Apache til at køre en lokal server på maskinen, da enhederne ikke vil være tilsluttet netværk, var det en smart løsning at lave slideshowet som en lokal webside med HTML, CSS og javascript.

Vi fik noget materiale bestående af en tekst, og nogle informativ grafik fra sundhedsstyrelsen, som vi selv skulle tilpasse landskabsformat i den rigtige opløsning. I første omgang forsøgte vi at lave det så stilrent som muligt, med brug af få farver og holde stilen fra sundhedsstyrelsens materiale.

Beslagene blev lavet af smedene efter specifikke mål, så de passede til at montere skærmene og raspberry’erne på en væg. Automatikteknikere har drejet skruer, der passede til monteringen og elektrikerne har samlet stikdåser til projektet.

Da vi havde 4 fungerende enheder, fået beslagene, samt udarbejdet en grundlæggende dokumentation for opsætningen, præsenterede vi løsningen for uddannelseschef Niels og Instruktør Noree. De var tilfredse med løsningen og ønskede seks enheder til at sætte op på TEC Ballerup, de ønskede også at der var noget grafik der talte mere til målgruppen.

Undervejs har der været udfordringer med defekt hardware, manglende kabler til skærme. Ligesom det flere gange har bremset os, at vi ikke har rettigheder til at kunne slette og oprette data på lagringskort fra arbejdscomputerne, vi har derfor været afhængige af, at skulle have fat i instruktører eller support.

Vi kunne ligeledes godt have ønsket os, at det grafiske materiale til skærmene enten var udarbejdet centralt fra, eller at der havde været mere udførlige ønsker derfra.

Vi forsøgte dernæst at implementere video-materiale på siden, men det havde vi ikke held med. Dernæst udarbejdede vi grafik med humor med fokus på håndhygiejne, afstand og masker. Vi sendte grafikken til godkendelse og samlede enhederne, som så var klar til at blive sat op.

Grundlæggende har været en sjov opgave, det har været spændende at lave noget, der skulle anvendes til noget. Vi har lært en masse forskelligt, og det var en fed udfordring, at vi selv skulle finde løsningen og forløbet har været uden de store problemer eller frustrationer.

# Kildelist

https://www.komplett.dk/product/1140492/hardware/skaerme/skaerme/benq-22-skaerm-bl2283#

<https://raspberrypi.dk/produkt/raspberry-pi-3-model-b-plus-starter-kit/>

<https://www.harald-nyborg.dk/produkt/4-stikdaase>