Covid-19 Infostander

TVÆRFAGLIGT PROJEKT

Simon Locht Nørregaard, Anders Krog, Nikolai Winther Juhl, Nicolai Garro Throksø, Frederik Teddy Fly | 20-11-2020

Contents

Case

Opgaven går ud på at oprette en infoskærm med én skærm, der kan vise et slideshow automatisk. Vi skulle anvende Raspberry PI 3 og Philips-skærme, som skulle fungere som infostandere med grafisk info om tiltag ift. Covid-19. Den skal kunne bruges uden inputs fra keyboard eller netværksadgang. Den skal selv starte op, så snart den tilsluttes strøm. Projektet er tværfagligt og løses med hjælp fra andre faggrupper herunder automatikteknikerne, elektrikerne og smedene.

Fremgangsmåde

Vi har valgt at installere Raspberry PI OS uden Desktop for at spare kræfter og gøre maskinerne så effektive som muligt. Vi har oprettet en server (Apache2) direkte på vores maskine. Vi har derefter installeret Chromium, kører i 'Kiosk Mode', dvs. at Chromium kører i fuld skærm og ikke behøver bruger-inputs. Chromium er blevet sat til at åbne URL'en "http://localhost". På localhost-adressen viser Apache2-serveren en HTML-side, med billeder i et automatisk slideshow, som er programmeret med JavaScript. Billederne eller filerne har vi overført via et USB-medie. Til sidst har vi beskrevet hvad man kan gøre af variationer, og hvilke problemer man kan støde ind i.

Materiel

Standeren:

- Micro SD hukommelseskort. Vi har anvendt 16GB..
- USB-Kortlæser til Micro-SD eller eventuelt SD med adapter.
- Raspberry PI Model B+
- Micro USB 5V 2.5A strømforsyning
- En form for kabinet til at beskytte Raspberry'erne
- Full HD Skærm
- HDMI-kabel

Installation og opsætning.

- Internettilslutning
- Raspberry Pi Lite OS image fil
- Tastatur
- USB-medie til overførsel af websiden

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, så ville være sværere at opsætte.

Skærm

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

Hvis man skulle vælge en anden skærm, kunne man fx benytte en BenQ 22" skærm BL2283. IPS-panel er at foretrække fordi billedet generelt er pænere og læsevinklen god. Dens drawback, opdateringshastighed og Hz er irrelevant for vores projekt.

Videokabel

Med de anvendte skærme skal der anvendes HDMI til DVI, da Philips-skærmene ikke har HDMI-indgang. Hvis man anvender en skærm med HDMI-indgang kan man benytte et HDMI til HDMI. 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.

Ethernet-kabel og internet-adgang

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

Tastatur

Det er ikke nødvendigt med mus til installation eller opsætning.

USB-medie

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

Budget

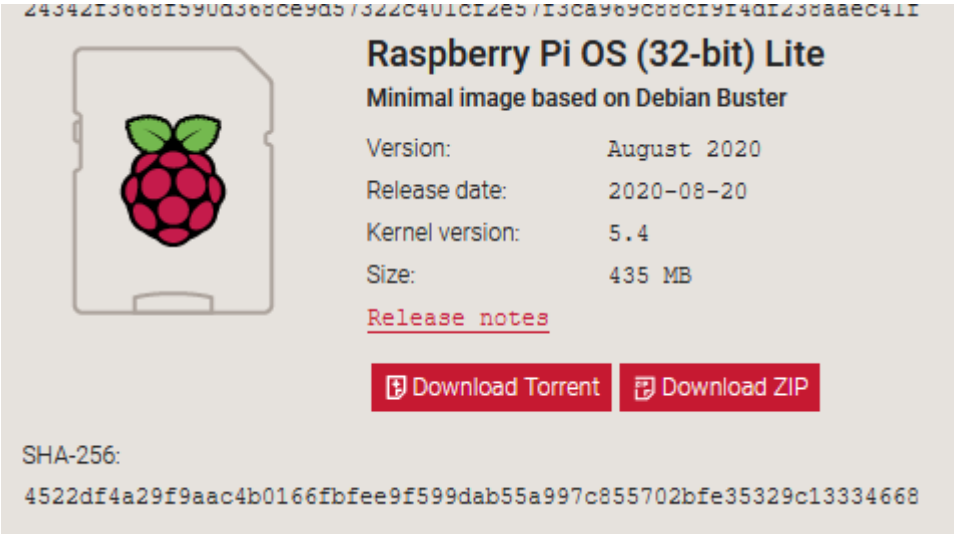
Produkt	Pris u. moms	Pris m. moms
BenQ 22"	633,6	792
Raspberry PI 3 Starter Kit	469,6	587
HDMI-kabel 1m	31,20	39
Stikdåse min. 2 stik	11,60	14,50

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 vælge versionen uden Desktop (Raspberry PI OS Lite)



Hent et program som kan brænde dit operativsystem image til dit SD-kort. Vi bruger Rufus i dette eksempel. Et alternativ til Linuxbrugere 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”. **Ret dette til XKBLAYOUT=”dk”.**

reboote eventuelt

Det er en god ide at updatere systemet inden du går i gang med installationen. Det køres med følgende kommandoer:

```
apt-get update
```

```
apt-get upgrade
```

Disse kommandoer opdaterer 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 et helt desktop.

Vi bruger 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 # er blot kommentarer, der ikke læses når scriptet køres, så disse linjer behøves ikke at indtastes. I filen skal du skrive følgende:

```
# 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://localhost'
```

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.

AUTOLOGIN

Får at få maskinen til automatisk at starte op skal vi anvende **autologin.**

Dette tilføjes i Raspberry PI's window manager menu

sudo nano raspi-config

I menuen der åbnes tryk system og autologin med bruger PI

For at starte selve programmet skal vi 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
```

INSTALLATION AF LOKAL SERVER

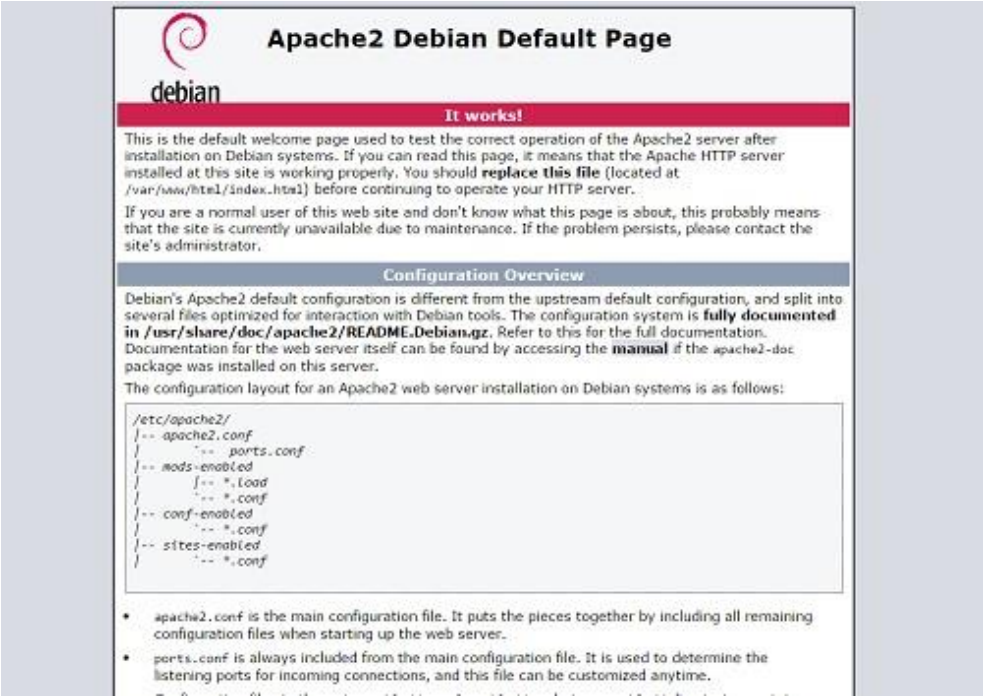
Vi bruger apache2 til at hoste vores server. Dette program installeres og startes med disse kommandoer:

```
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.

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.

Tilføj billeder til slideshowet

OVERFØR FILER VIA USB

Du kan tilføje nye billeder via USB-medie.

OBS.

Hvis man ikke ændrer i koden, men ønsker nye billeder, er det muligt at overskrive de gamle billeder med nye der hedder det samme.

Det er vigtigt at være opmærksom på store og små bogstaver i filnavnene, da Linux er 'case sensitive'

Hvis man har foretaget ændringer i CSS-filen skal man være opmærksom på at den er cached. Det vil sige, at den ikke automatisk genindlæses hvis den overskrives. For at løse dette tvang vi genindlæsning ved at give den et nyt navn og sørge for at henvise til det nye navn i html'en

OPRET MOUNT

For at kunne tilgå data'en på usb-lagermediet, skal man mounte det. Vi opretter en mappe specifikt til formålet:

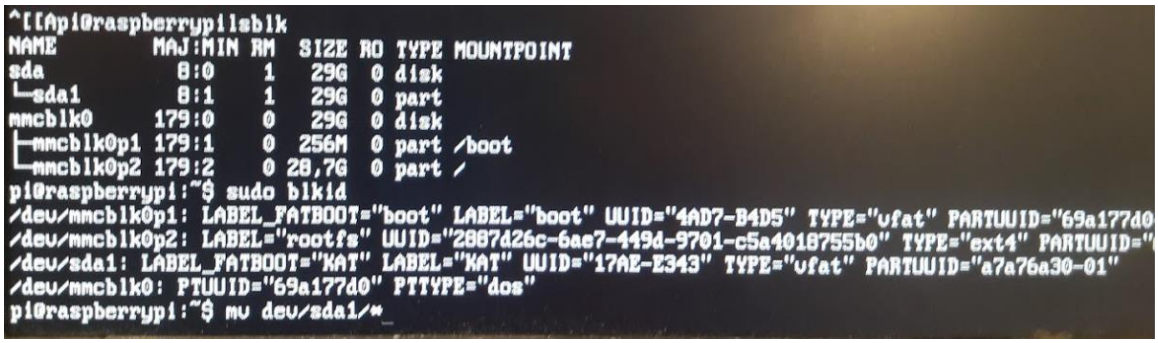
```
sudo mkdir /media/USB
```

for at finde navn og sti på USB-mediet:

```
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 stien, som vi skal bruge.

Kommandoerne er kun eksempler. Den egentlige sti kan være anderledes

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:

sudo mount /dev/sda1 /media/USB/

Nu hvor drevet er mountet, kan vi overføre filerne til den mappe hvor Apache2 lægger sine HTML-dokumenter som standard. Skriv kommandoen:

sudo cp /media/USB/* /var/www/html

Dokumentation af koden

Til at lave selve websiden, bruger vi en html-fil (index.html) der indeholder 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

```
<!DOCTYPE html>
<html>

<head>
<title>W3.CSS</title>
<meta name="viewport" content="width=device-width, initial-scale=1">
<meta content="text/html; charset=iso-8859-2" http-equiv="Content-Type">
<link rel="stylesheet" href="w3.css">
<style>
.mySlides {display:none;}
</style>
</head>

<body>
```

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.

```
<div class="w3-content" style="max-width:1920px; max-height:1080px">
  <img class="mySlides w3-image w3-animate-fading" src="img1.jpg" style="width:100%">
  <img class="mySlides w3-image w3-animate-zoom" src="img1.jpg" style="width:100%">

  <img class="mySlides w3-image w3-animate-fading" src="img2.jpg" style="width:100%">
  <img class="mySlides w3-image w3-animate-zoom" src="img2.jpg" style="width:100%">

  <img class="mySlides w3-image w3-animate-fading" src="img3.jpg" style="width:100%">
  <img class="mySlides w3-image w3-animate-zoom" src="img3.jpg" style="width:100%">

  <img class="mySlides w3-image w3-animate-fading" src="img4.jpg" style="width:100%">
  <img class="mySlides w3-image w3-animate-zoom" id="2" src="img4.jpg" style="width:100%">

  <img class="mySlides w3-image w3-animate-fading" src="img5.jpg" style="width:100%">
  <img class="mySlides w3-image w3-animate-zoom" src="img5.jpg" style="width:100%"

  <img class="mySlides w3-image w3-animate-fading" src="img6.jpg" style="width:100%">
  <img class="mySlides w3-image w3-animate-zoom" src="img6.jpg" style="width:100%"


</div>
```

Javascriptdelen ser således ud:

```
<script>
var myIndex = 0;
var myNextIndex = 1;
carousel();

function carousel() {
  var i;
  var x = document.getElementsByClassName("mySlides");

  // skjul alle bileder
  x[myNextIndex].style.display = "block";
  for (i = 0; i < x.length; i++) {
    if (i!=myNextIndex && i!=myIndex)
  {
    x[i].style.display = "none";
  }
  }

  //x[myIndex].style.display = "block";

  myIndex++;
  myNextIndex++;
  if (myIndex > x.length-1) {myIndex = 0}
  if (myNextIndex > x.length-1) {myNextIndex = 0}

// Tells the infoscreen to wait with changing images, depending, if the image has an id.
// To add more options for wait time, simply cope-paste the else if section, and put another id == x. And change the time in the setTimeout to the prefered waiting in miliseconds
// the "if" always needs to be on the top, and the "else" needs to be on the bottom.
  if(x[myIndex].id == 1) // This will run if the image has an id of 1
  {
    setTimeout(carousel, 15000);
  }
  // To add more wait-time options, copy from below this line.
  else if(x[myIndex].id == 2) // This will run if the image has an id of 2
  {
    setTimeout(carousel, 10000);
  }
```

```
    // Stop copy above this line
    else // This is the standard wait time, if the image has no id.
    {
      setTimeout(carousel, 5000);
    }
  }
  </script>

  </body>
  </html>
```

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

```
27  html,body{font-family:Verdana,sans-serif;font-size:15px;line-height:1.5}html{overflow:hidden}

.w3-animate-fading{animation: fading 10s 1}@keyframes fading{0%{opacity:0}50%{opacity:1}100%{opacity:1}}
```

```
123  .w3-animate-zoom {animation:animatezoom 40s}@keyframes animatezoom{from{transform:scale(1)} to{transform:scale(1.2)}}
```

Specifikationer for billeder

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. Det er muligt at anvende andre fil-formater, hvis man sørger for at angive filtypen. Men vi har holdt os til jpg. 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 setupet 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.

Kilder

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