

Software

Konfiguration im Docker-Container

version: '3'

services:

magicmirror:

container_name: mm_sven

image: karsten13/magicmirror:latest

ports:

- "8080:8080"

volumes:

- /mnt/synology_docker/mm_sven/config:/opt/magic_mirror/config

- /mnt/synology_docker/mm_sven/modules:/opt/magic_mirror/modules

- /mnt/synology_docker/mm_sven/css:/opt/magic_mirror/css

restart: unless-stopped

command:

- npm

- run

- server

networks:

default:

ipv4_address: IP der MagicMirror App

networks:

default:

name: macvlan

external: true

Konfiguration des Raspberry Pi Zero

Zuerst habe ich versucht bei dem Raspberry Pi Zero das Standard-Raspberry-Image zu installieren und ihn dazu zu bringen die MagicMirror-Webseite bei Systemstart zu öffnen. Totaler Fehlschlag - das Teil ist einfach zu schwach. Gottseidank habe ich ein anderes Betriebssystem gefunden, was genau das tut, was ich brauche. Chili-Pi. Nun - hab ich zuerst zumindest gedacht...

[Links zum Image und zur Chili-Pi-Anleitung](#)

<https://github.com/jareware/chilipie-kiosk/releases/download/v3.0.0-rc4/chilipie-kiosk-3.0.0-rc4.img.tar.gz>

<https://github.com/jareware/chilipie-kiosk/blob/master/docs/first-boot.md>

Erst mal mit ctrl+alt +F2 in den Raspi-Config-Modus und hier SSH aktivieren, Wifi aktivieren, Speicherplatz zuweisen und so weiter. Den Pi kann man dann mit sudo apt-get update / sudo apt-get upgrade auf den neuesten Stand bringen.

Da das Display hochkant eingebaut worden ist, muss die Anzeige noch gedreht werden. In der Anleitung steht zwar etwas anderes - aber bei mir hat das gereicht.

```
sudo nano /etc/X11/Xsession.d/45x11-rotate_screen  
DISPLAY=:0 xrandr --output HDMI-1 --rotate left
```

Was allerdings nicht funktioniert hat ist das umschalten auf andere Webseiten (z.B. einen zweiten MagicMirror Container oder eine andere Webseite.

Dann bin ich aber noch über eine andere Distribution gestolpert: FullPageOS - die sollte das gleiche machen. Also mal ausprobiert..

FullPageOS

Setzen der URL

```
sudo nano /boot/fullpageos.txt
```

Hier einfach z.B. <http://magicmirror.streb.home:8080> eintragen

Drehen des Bildschirms, da er hochkant eingebaut ist

```
sudo nano /etc/X11/Xsession.d/45x11-rotate_screen  
DISPLAY=:0 xrandr --output HDMI-1 --rotate left # das dreht das Display 3x mit dem Uhrzeigersinn  
DISPLAY=:0 xset s off && xset -dpms # damit power savings vom Controller ausgeschaltet wird
```

WLAN Power Mode abschalten

```
sudo iwconfig wlan0 power off
```

Bildschirm "dunkel" machen

xrandr --output HDMI-1 --off - das schaltet aber HDMI komplett ab und der Bildschirm wird halt blau - also besser:

```
sudo nano /home/pi/skripts/display-off.sh  
export DISPLAY=:0 # um das Display auszuwählen  
xrandr --output HDMI-1 --brightness 0.0 # um die Helligkeit auf 0 zu setzen
```

```
sudo nano /home/pi/skripts/display-on.sh # um das Display auszuwählen  
xrandr --output HDMI-1 --brightness 1.0 # um die Helligkeit auf 100% zu setzen
```

-> dreht also nur die Helligkeit auf 0, aber schaltet HDMI nicht ab.

weitere Skripte im Verzeichnis:

fullscreen (drückt F11)

reload_fullpageos_txt (schießt den Task ab und lädt neu)

refresh (läd die Seite neu - man braucht aber manchmal einen F11 um wieder in Fullscreen zu kommen)

... und noch ein paar andere, die ich noch nicht ausprobiert habe.

Konfiguration vom ioBroker

Auf IOBroker das Simple-SSH Modul installieren

-> npm install simple-ssh

der IOBroker bekommt 2 Dateien angelegt... eine Spiegel_aus.js und eine Spiegel_an.js (unter /opt/SpiegelScript)

Beispiel Spiegel_aus.js (analog dazu eine Spiegel_an.js erstellen)

```
var SSH = require('simple-ssh');
var ssh = new SSH({
  host: 'raspi-bad.streb.home',
  user: 'pi',
  pass: 'tollespasswort'
});

ssh.exec('/home/pi/skripts/display-off.sh',{
  out: function(stdout) {
    console.log(stdout);
  }
}).start();
```

Zur Erläuterung:

Das soll über einen Bewegungssensor im ioBroker geschaltet werden - das Display soll ja nicht den ganzen Tag an sein, sondern nur, wenn jemand im Bad rumstromert.

Das simple-SSH Modul was da installiert wurde plus die zwei .js Skripte befähigen den ioBroker dazu sich auf dem Pi anzumelden und das lokale display-off/on Skript auszuführen.

Testweise habe ich den Pi an nen Monitor gehangen und das Ausführen des Skripts an meinen Bewegungsmelder im Büro gebunden (die exec Bausteine). Klappt prima.

Blocky-Baustein:

Wenn sich das Objekt Motion ändert, dann schau mal

... wenn es sich auf true ändert, dann führe das eine Skript aus

... wenn es sich auf false ändert (Dauer konfigurierbar auf dem Bewegungsmelder), dann führe das andere Skript aus

[image.png](#)

Der Spiegel soll aber auch unsere Kameras anzeigen - doofer weise ist der Pi Zero etwas schwach, um das konstant zu schaffen - aber es reicht ja auch wenn er immer mal ein Foto anzeigt um zu sehen, wie das Wetter draußen ist.

Da der Browser aber auch nicht direkt auf die Snapshot-Funktion der Reolink-Kameras zugreifen kann habe ich ein kleines .sh-Skript geschrieben (get URL und crontab -e), um alle 5 Minuten die letzten Screenshot der Kameras herunterzuladen. Das Modul MMM-ImagesPhotos von MagicMirror zeigt die jetzt nacheinander an.

Um das aber noch aufm PI hinzubekommen braucht der auch einen "Refresh", damit die Webseite neu geladen wird (auch ein .sh-Skript - ./refresh.sh && sleep 10 && ./fullscreen).. der Refresh bringt den Browser ausm Fullscreen, also bekommt er noch den Tastendruck F11 verpasst, um wieder reinzugehen...

Revision #10

Created 2024-01-30 15:43:46 UTC by ahzurdan

Updated 2024-02-12 06:30:47 UTC by ahzurdan