Inhaltsverzeichnis

LED Panel with a real Sunrise

A RGB LED Panel can show all colors of a sunrise, but can it also simulate a real existing sunrise as such? Let's try..

First we need to find real sunrise data. YouTube is of course one of the best sources for that, so try to find a movie of a beautyful sunrise. Make sure that the sunrise shows many color transisions, not only a stupid „it gets brighter“..

One example would be https://www.youtube.com/watch?v=x3ikZ4gMnIk

The aqua scene was made with https://www.youtube.com/watch?v=qwz88S1P0os

When the right movie has been found, there are some online websites, which help to download YouTube videos as local movie files.

After done that, the magic of Imagemagick starts. In that package there's a image swiss knife tool simply named convert, which can do the whole data conversion in one single go:

- It tooks the pixel data of a defined rectangle for each single frame of the movie (`-crop`)
- then it calculates the RGB average of the whole rectangle into a single pixel rgb value (`-resize`)
- and it writes the result directly into a C array source code syntax (`-format`)

all this is done with the simple call

```
convert Timelapse\ of\ a\ Sunrise..mp4 -crop 520x390+100+80 +repage  -resize 1x1\!     -format "{%[fx:int(255*r+.5)],%[fx:int(255*g+.5)],%[fx:int(255*b+.5)]},\n" info:-
```

ImageMagick does not use top-left to bottom-right coordinates to crop. It uses WxH+Xoff+Yoff, where W=width, H=height and Xoff,Yoff are the top left coordinates.

Sometimes convert fails with an `convert-im6.q16: cache resources exhausted` error. In that case you can increase the buffer in `/etc/ImageMagick-6/policy.xml`:

```
  <policy domain="resource" name="memory" value="4GiB"/>
```

Take the source code from https://github.com/stko/esp-mqtt-rgb-led and insert the generated rgb map into the file esp-mqtt-rgb-led/mqtt_esp8266_light/sunrise.c

Install the Arduino IDE. Please note:

The Arduino Library Manager installs the ArduinoJson version 6 by default.

However, using version 5 is highly recommended because version 6 is still in beta stage. Open the Arduino Library Manager and make sure that ArduinoJson version 5 is installed.

To not need to connect the RGB controller straight to the LED Panel, I've placed a DSub connector in between with the following pinout (just for my own reference..):

Pin	female (controller side) wire color	male (Panel side) color	was originally connected on the MiFare controller as
1	OR-WH	Black	V+
2	OR	Yellow	WW
3	GN-WH	White	CW
4	GN		
5	-		
6	BL-WH		
7	BL	Green	G
8	BR-WH	Blue	B
9	BR	Red	R

In my setup I've used originally a hand wired combination of a Wemos SoC plugged on to of an N-Shield MOSFET driver PCB.

Pinbelegung RGB-Panel (PAN-RGBWW-30-120) an einem Mi-Light FUT039:

Mifare Panel	Wire	N-shield	Wemos	Pins	Drahtfarbe	ESP012
V+	Schwarz	V+		-	WH-OR	
WW:	Gelb	D3		D4	OR	02
CW:	Weiß	D5		D3	WH-GN	00
V+	Frei	-				
R	Rot	D6		D2	BR	04
B	Blau	D10		D5	WH-BR	14
G	Grün	D9		D1	BL	05

After that I've realized that exactly the same functionality is available with the ready-to-flash ESP012

P.S.: Note that you can also install the MAP service through the Paper UI → Extensions → Transformations → MAP

In der Arduino IDE das richtige Board einstellen, damit das Flashen funktioniert:

Board ist LOLIN (WEMOS) D1 R2 & mini

Ansteuerung mit einem H801

Der H801 ist die einfachste Lösung. Achtung; Wie das Sonoff-Tasmota- Wiki zu berichten weiß, ist die Beschriftung der RX/TX- Signale auf der Platine auf vom Terminal kommenden Signale ausgelegt, die beiden Signale müssen also im Gegensatz zum „Standard“ **nicht** gekreuzt werden!

In der Arduino IDE wählt man „Generic ESP8266 Module“, als Programmer den „ARVISP mkII“ (für den FTDI232)

Einrichten des Panels

Die Panel Firmware verwendet CoogleIOT als Wifi/MQTT Manager. Daraus ergibt sich nach dem

Flashen folgendes Setup:

- das Panel macht ein eigenes WLAN auf (CoogleIOT_xxxxx).
- Zuerst einmal verbindet man sich mit seinem Handy mit diesem WLAN (initiales Passwort `coogleiot`)
- Dann geht man auf die Einrichtungsseite http://192.168.0.1
- Dort gibt man ein:
  - neues AP Passwort
  - sein eigenen WLAN-AP und -Passwort
  - den eigenen MQTT- Server
  - den MQTT- Port (1883)
  - und die „MQTT Client Id“, also die, unter der die Topics des Devices dann laufen sollen (siehe unten). Weil das System selber nicht prüft, muss man selber auf eine zulässige Schreibweise des Client- Namens achten.

Danach bootet man mit `Save & Reboot` neu. Nach ca. 20 Sekunden sollte sich das Gerät dann im eigenen Netz eingewählt haben. Der eigene AP bleibt aber weiter geöffnet, sehr zur Experimentierfreude der Nachbarn..

MQTT Kommandos

Je nachdem, welche „MQTT Client Id“ (`MQTTClientId`) man dem Panel gegeben hat, ergibt sich daraus das Topic = `home/<MQTTClientId>/set`

Message	Effekt
{"state": "ON"}	Licht an
{"state": "OFF"}	Licht aus
{"effect":"sunrise","transition":600}	Movie „sunrise“ über 600 Sekunden
{"color": { "r": 0,"g": 0,"b": 255},"transition": 5,"white_value": 0,"cold_white_value": 0}	Blaue Transition
{"brightness": 255,"color": { "r": 255,"g": 255,"b": 255},"white_value": 255,"cold_white_value": 255}	Alles an


```
[{"id":"67a1adb0.e42284","type":"tab","label":"Sunrise","disabled":false,"in
fo":""},{"id":"3fec84a9.e74f6c","type":"inject","z":"67a1adb0.e42284","name"
:"","topic":"","payload":"{\"effect\":\"sunrise\",\"transition\":600}","payl
oadType":"json","repeat":"","crontab":"","once":false,"onceDelay":0.1,"x":28
0,"y":200,"wires":[["163d12b6.a7b6fd"]]},{"id":"163d12b6.a7b6fd","type":"mqt
t
out","z":"67a1adb0.e42284","name":"","topic":"home/SUNRISE_2/set","qos":"","
retain":"","broker":"570508d4.908688","x":720,"y":160,"wires":[]},{"id":"dee
f52c7.5f52e","type":"inject","z":"67a1adb0.e42284","name":"","topic":"","pay
load":"{\"state\":\"OFF\"}","payloadType":"json","repeat":"","crontab":"","o
nce":false,"onceDelay":0.1,"x":300,"y":240,"wires":[["163d12b6.a7b6fd"]]},{"
id":"5bf020f9.803f4","type":"inject","z":"67a1adb0.e42284","name":"","topic"
:"","payload":"{\"state\":\"ON\"}","payloadType":"json","repeat":"","crontab
":"","once":false,"onceDelay":0.1,"x":290,"y":280,"wires":[["163d12b6.a7b6fd
"]]},{"id":"3ebdb00c.5f7a3","type":"inject","z":"67a1adb0.e42284","name":"",
```

```
"topic":"","payload":"{\"effect\":\"aqua\"}","payloadType":"json","repeat":"
","crontab":"","once":false,"onceDelay":0.1,"x":300,"y":40,"wires":[["163d12
b6.a7b6fd"]]},{"id":"7a921fd3.17fb7","type":"inject","z":"67a1adb0.e42284","
name":"","topic":"","payload":"{\"effect\":\"sunrise\",\"loop\":\"false\"}",
"payloadType":"json","repeat":"","crontab":"","once":false,"onceDelay":0.1,"
x":280,"y":120,"wires":[["163d12b6.a7b6fd"]]},{"id":"3121006f.2d812","type":
"inject","z":"67a1adb0.e42284","name":"","topic":"","payload":"{\"effect\":\
"sunrOse\"}","payloadType":"json","repeat":"","crontab":"","once":false,"onc
eDelay":0.1,"x":310,"y":160,"wires":[["163d12b6.a7b6fd"]]},{"id":"62633276.2
5399c","type":"inject","z":"67a1adb0.e42284","name":"","topic":"","payload":
"{\"effect\":\"sunrise\"}","payloadType":"json","repeat":"","crontab":"","on
ce":false,"onceDelay":0.1,"x":310,"y":80,"wires":[["163d12b6.a7b6fd"]]},{"id
":"570508d4.908688","type":"mqtt-
broker","z":"","name":"OpenHAB","broker":"openhabianpi","port":"1883","clien
tid":"","usetls":false,"compatmode":false,"keepalive":"60","cleansession":tr
ue,"birthTopic":"","birthQos":"0","birthPayload":"","closeTopic":"","closeQo
s":"0","closePayload":"","willTopic":"","willQos":"0","willPayload":""}]
```