

Phosphorescent Persistence of Vision



In diesem Projekt soll ein Display gebaut werden, bei dem mit UV-Leds auf phosphoreszierende Folie ein Text oder Grafiken geschrieben werden kann. Die Folie wird dafür auf eine rotierende Trommel geklebt. Der Aufbau soll aber auch zum Modden einladen, um Displays in beliebigen Formen (Scheibe, Band, ...) zu realisieren.

Bestellung für die Workshop Vorbereitung

Ausführlichere Liste: [pov_stueckliste.xlsx](#)

Software



Als Controller wird ein ESP8266 in Form eines WeMos D1 mini verwendet.

Das WEMOS D1 mini ist ein Modul aus dem Hause Wemos.

https://wiki.wemos.cc/products:d1:d1_mini

Es basiert auf einem ArduinoIDE kompatiblen ESP8266 (ESP) Controller mit 4Mbit Flash und WiFi Antenne zur Verbindung mit einem WLAN Netz und somit mit der Möglichkeit Internet oder Browser Zugriff zu bieten. Zum Programmieren und zur Stromversorgung steht eine Micro-USB Buchse zur Verfügung.

Die Grundidee ist ein ständig laufender Webserver auf dem ESP, der sowohl für die Kommunikation mit der Applikation (hier der UV LED Steuerung) als auch für die Konfiguration des WLAN zuständig ist. Das Ganze wurde in der Arduino IDE entwickelt und kann selbst angepasst werden oder einfach als .HEX Firmwarefile auf den ESP geflasht werden.



Zur Programmierung wird eine Arduino-Entwicklungsumgebung (Version 1.0, die man z.B. auf ubuntu 16.04 findet, ist allerdings zu alt) und die ESP8266-Anpassung benötigt. Eine Anleitung findet man unter https://wiki.wemos.cc/tutorials:get_started:get_started_in_arduino

Danach kann man unsere aktuelle Software von <https://github.com/fablab-ka/PHABLABS-POV> herunterladen und mit der Arduino-Entwicklungsumgebung auf das Board überspielen.

Firmware Flashen

ESPPoV auf den ESP flashen **ACHTUNG Dieser Absatz ist nicht aktuell. Siehe vorherigen Absatz zu Software**

Die benötigte Firmware für den ESP lässt sich auf http://wiki.fablab-karlsruhe.de/doku.php?id=projekte:2017:phablabs4_0:persistence_of_vision herunterladen. Am unteren Ende der Seite findet man den Link zur benötigten Firmware.

Im entpackten ZIP-Archive (zum Zeitpunkt der Erstellung: ESPPoV_1.0x) erhält man nun die benötigten Dateien. Das Programm zum Flaschen ist mit dabei. Das Flaschen wird durch Öffnen der Datei „flash.cmd“ gestartet. Es öffnet sich ein Konsolenfenster. Bevor das Flaschen beginnt, müssen noch drei kleine Fragen beantwortet werden:

- Abgefragt wird der COM-Port unter dem das Board zu finden ist.
- Des Weiteren wird nach der Größe des Flash-Speichers gefragt. Diese bezieht sich auf den zu programmierenden ESP8266-Chip. Bei den WeMos-Boards ist dies zum Beispiel 4096 Bytes.
- Zum Schluss wird noch nach der gewünschten Build-Nummer gefragt.

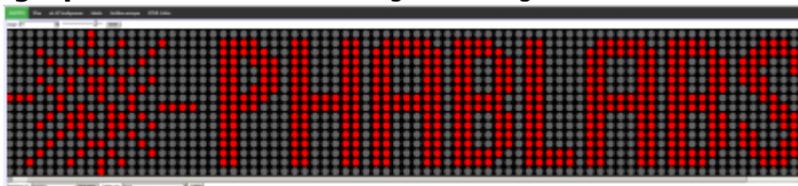
EasyESP flashtool

```
C:\Windows\system32\cmd.exe
Comport (example 3, 4, ..) :3
Flash Size (example 512, 1024, 4096) :4096
Build (example 71, 72, ..) :78
```

Bedienung

Derzeit startet der ESP in den AP-Mode und ist über den im Sourcecode hinterlegten SSID-Namen PHABLABS und das Passwort ABCdef123456 erreichbar. Sollte der ESP bereits die Zugangsinformationen zu einem aktuell erreichbaren WLAN gespeichert haben (z.B. weil vorher ein anderer Sketch geladen war), so verbindet er sich dorthin! Das POV Display zeigt in allen Fällen dann die IP-Adresse an, unter welcher der ESP zu erreichen ist.

- **/content** Erlaubt Inhalt, Helligkeit und Geschwindigkeit des Textes einzustellen. Uhrzeit und Datum sind derzeit nur dann aktuell, wenn der WeMos sich als Client in einem WLAN befindet und einen NTP-Server erreicht (de.pool.ntp.org).
- **/graphicsEditor.html** Ermöglicht es, grafische Inhalte zu editieren:



- **/wificonnectAP** Ermöglicht es dem ESP, sich als Client an ein vorhandenes WLAN zu verbinden.
- **/wificonfigAP** Erlaubt es, SSID und Passwort für den AP Mode, sowie das Passwort für den Administrationsaccount (voreingestellt admin/admin) zu ändern.
- **/wifiRestartAP** Löscht die gespeicherten WLAN Client-Zugangsdaten und startet im AP-Mode
- **/edit** Startet den eingebauten ACE Editor, um die auf dem SPIFFS abgelegten Dateien zu editieren. Für den Editor ist es nun unabdingbar, dass der Inhalt des data-Unterverzeichnisses mittels des oben beschriebenen ESP-Uploadmanagers auf des SPIFFS geladen wird. Ansonsten werden die Javascriptdateien nicht gefunden!

Links

- <https://hackaday.com/2014/01/11/i-am-the-midnight-message-board-what-messages-at-midnight/>
- <http://hackaday.com/2011/07/13/faux-led-scroller-using-phosphorescence/>
- <http://hackaday.com/2011/09/09/spinning-uv-light-writer/>

Material

Phosphorband: 25mm 50mm: 1m ca 6 EUR

Motor:

Schrittmotor, reicht 5,70 EUR

[Motor bei Reichelt](#)

hier im Wiki: [Schrittmotor 28BY-48](#)

LED:



Platine

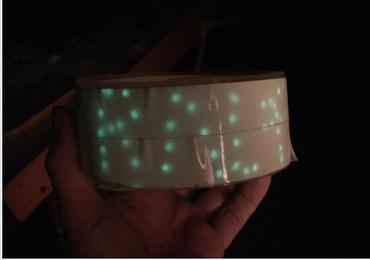


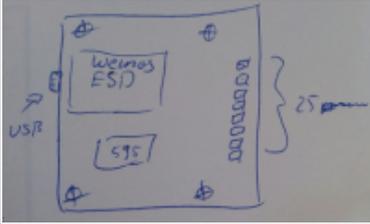
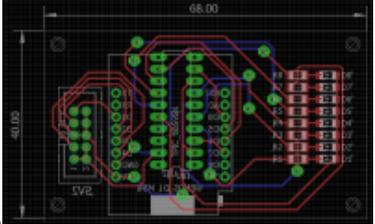
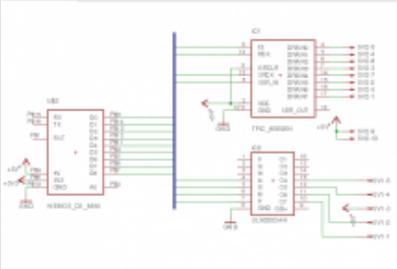
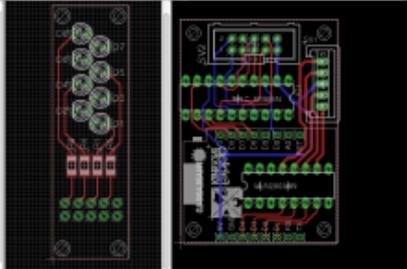
Team

- Stefan S.
- Wolfgang K.
- Ralf
- Philip C.
- (Sebastian H.)

LOG

Datum	Name	Task	Beschreibung
24.02.2017	Sebastian	KickOff	Erstes Treffen der Gruppe (noch unter dem „Laser Cutter“ Projekt) 
28.02.2017	Sebastian	Brainstorming	Treffen zum Ideen sammeln und Verteilen von Aufgaben. Erstes grobes Design.
...			
06.03.2017	Gerd	Erstentwurf	Erstes Design der Mechanik fertig
...			
18.04.2017	Ralf / S2i	Full Stop	Abbruch des Projekts „Lasercutter“, da Lasersicherheit während des Aufbaus nicht garantiert werden kann 

Datum	Name	Task	Beschreibung
25.04.2017		KickOff	Projektstart für „UV PoV Projekt“
28.04.2017	Hans	Experimente	Erste Versuche und Ideen zum Aufbau 
02.05.2017	Hans/ Wolfgang	Start	Projekttreffen
23.05.2017	Seb./ Hans/ Ryad	Prototyp	Aufbau erster Prototyp mit gefräster Platine von Philip 
25.-28.05.2017	Wolfgang	Software	GPN - Software Hack Session https://github.com/fablab-ka/ESP-Matrix
01.06.2017	Philip	Experimente	Versuche zum Zusammenspiel von LEDs, Leuchtfolie und Stromstärken/Timing
05.06.2017	Ralf	Dokumentation	Kurzbeschreibung und Dokumentation auf dem Wiki aktualisiert
08.06.2017	Ralf	Dokumentation	Beschreibung zu Software, Firmware und Bedienung hinzugefügt
13.06.2017	Ralf	Partnermeeting	Ein Einwand zur Sicherheit - UV Licht ist für die Augen schädlich, darum: - In der Anleitung darauf hinweisen - evtl. weiße LEDs verwenden (Schließreflex) - LEDs nur einschalten wenn SyncCode (schwarz-weißes Muster) erkannt
18.06.2017	Sebastian	Zeit	für die nächsten 4-6 Wochen keine Zeit am Projekt zu arbeiten

Datum	Name	Task	Beschreibung
23.06.2017	Philip C. / Stefan S.	Prototypenbau	- Platine mit einschaltbaren LEDs gefräst und bestückt. - Motor- und Rollenhalterung gebaut
24.06.2017	Ralf	Eckdaten	-PCB mit 5x5cm od. 5x10cm -Phosphorband mit 25mm oder 50mm Höhe (teuer!) Bandlänge 50cm - 100cm (gibt weniger als die Hälfte sichtbaren Bereich) -Auf ein 50mm Band passen acht 5mm LEDs od. 16 SMD LEDs. -Ein Platinen Nutzen hat 100x100mm → vier 50x50 Varianten möglich. -LowCost: 25mm Band mit 8 SMD LEDs Motor: aktuell Schrittmotor -LowCost (+ Sicherheit): Weisse od blaue SMD LED -LowCost: 50cm 25mm Band auf Trommel - z.B. Dose/ Honigeimer/ Zylinder/ ...
25.06.2017	Ralf	Bestellung	Schrittmotor bei reichelt bestellt
26.06.2017	Ralf	Platine	Skizze 
07.07.2017	Philip	PCB-Design	fräsbare Platine designed und gefertigt. Bestückung wegen  fehlenden Komponenten vertagt.
08.07.2017	Philip	PCB_Bestückung, Prototypenbau	 Platine bestückt und getestet. Ergebnis: SMD-LEDs eignen sich wegen großem Abstrahlwinkel nicht, Umstellung auf 3mm LEDs. Platine sieht außen hässlich und unförmig aus. Wieder LEDs und Restplatine trennen?
14.07.2017	Philip	PCB gefertigt	Board für 3mm LEDs gefertigt und bestückt. Funktioniert viel besser als mit SMD-LEDs. Layout für PCB-Bestellung wird als nächstes designed.
15.07.2017	Philip	PCB Layout abgeschlossen	  LED und Controller-Platinen gelayouted.
21.07.2017	Philip	neues PCB Layout fertig	Neues, preis-optimiertes Layout fertig.
24.07.2017	Philip	Platinen bestellt	Zeit bis Lieferung: Bis zu drei Wochen.

Datum	Name	Task	Beschreibung
09.08.2017	Philip	Platinen angekommen	<p data-bbox="671 197 1476 232">Mit Bestückung angefangen, fehlende Teile werden noch besorgt.</p> 
27.11.2017	Philip	Erster verwendbarer Kurs-Aufbau in Aufbau	<p data-bbox="671 645 1476 712">Hardware und Platine sind (vermutlich) fertig, Software muss noch fertig geschrieben werden.</p>