

## **Gedankensammlung vom 15.6.2016**

Dies ist meine Gedankensammlung von gestern zur Zechenuhr. Kurz zum Projekt: Die Uhr an der Zeche soll zum einen wieder ans rennen gebracht (also drehende Zeiger und Hintergrundbeleuchtung) und zum anderen ein bisschen lebendiger gemacht werden (Farbe, LEDs). Der classicmode soll dabei immer einschaltbar bleiben (Denkmalschutz u know.). Hier kurz die Dinge die Stephan und mir wichtig waren

### **Wünsche Stephan**

Wenn ich mich richtig entsinne ist Stephan MQTT für die Datenübertragung auf einer der Strecken zwischen Welt und Uhr wichtig damit das Setup erweiterbar für weitere Sensoren in der Zeche bleibt. Als "Gehirn" ist Stephan der STM32 wichtig ebenfalls für spätere Erweiterbarkeit.

### **Wünsche Malte**

Mir war ganz wichtig das man am Ende ein leicht verständliches Interface für Menschen hat. Also am besten eine Website mit Textfeld in welches man Menschenlesbare Strings zur LED-Konfiguration einträgt. Das zweite was mir wichtig ist, ist Netzwerkanbindung, also man soll es am Ende über LAN/WLAN konfigurieren können und nicht mit einem USB-Kabel in die Zeche laufen müssen.

### **Konzept Uhrbeleuchtung**

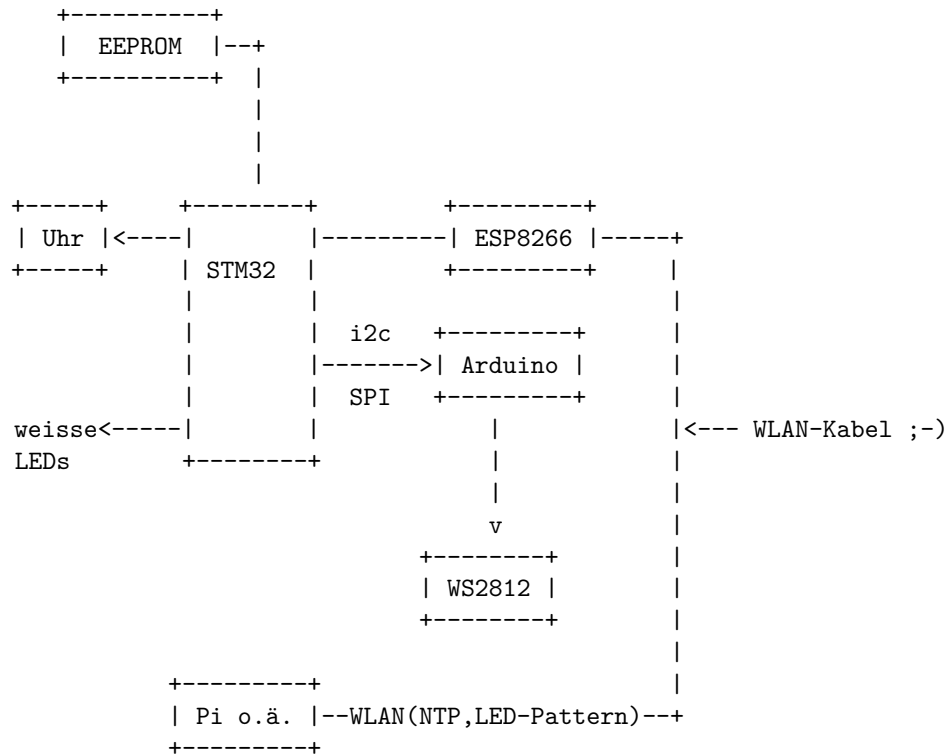
In die Uhr wird ein Aluring eingesetzt auf dem 30 weiße 1W LEDs sitzen 15 strahlen auf das Ziffernblatt das auf den Hof zeigt und 15 auf das andere. Auf der Seite die auf den Hof zeigt sitzen zusätzlich 60 WS2812 LEDs. Zusätzlich ist auf dem Ziffernblatt das auf den Hof zeigt an jeder "Sekunde" eine WS2812 LED befestigt. Auf ein Display werden wir wohl verzichten da jedes in Ziffernblattnähe angeklebte Teil einen Schatten auf das Blatt wirft und damit den "classic Mode" verhindert. Die Sekunden-LEDs werden auch in Blattnähe angeklebt allerdings in nähe der Sekundenmarker sodass kein zusätzlicher Schwarzer Fleck entsteht.

### **Konzept Uhransteuerung**

Das Herz der Anlage wird der STM32. Er übernimmt die Ansteuerung der Uhr (aussenden der Impulse). Ansteuerung der weissen LEDs für die Hintergrundbeleuchtung. Ansteuerung des Arduino. Der Arduino steuert die WS2812 LED-Stripes an. Parsen der LED-Pattern. Der ESP8266 dient dem entpacken von MQTT und senden der Pattern und der Zeit zum STM32. Zusätzlich ist der

ESP8266 natürlich das WLAN-Gateway der Anlage. Auf dem Pi läuft ein MQTT Server und der Pi sendet kontinuierlich die Uhrzeit und neue LED-Pattern.

## Schema



## Idee LED-Pattern-Sprache

Idee zur Pattern Sprache ist wie folgt. Es gibt Blöcke die mit einem Startzeichen beginnen, gefolgt von Anzeige und Überblendzeit, gefolgt von LED-Adressen und deren Helligkeit. Die Uhr stellt immer nur die Änderungen dar also alles seit dem letzten "Paket(Block)". Beispiel: Wir gehen von einer ausgeschalteten Uhr aus und schalten jede Sekunde einen weiteren LED-Punkt an.

---

```
Start|0|1|1->255|Start|0|1|2->255|Start|0|1|3->255|Start|0|1|4->255
```

---

Übersetzt: Block: 0 Sekunden Überblendzeit, 1 Sekunde warten vorm nächsten Block. LED1 auf volle Helligkeit. Block: 0 Sekunden Überblendzeit, 1 Sekunde warten vorm nächsten Block. LED2 auf volle Helligkeit. Block: 0 Sekunden

Überblendzeit, 1 Sekunde warten vorm nächsten Block. LED3 auf volle Helligkeit.  
Block: 0 Sekunden Überblendzeit, 1 Sekunde warten vorm nächsten Block. LED4  
auf volle Helligkeit. Nach vier Sekunden sind also die ersten vier LEDs an.

Stand des ganzen 16.06.2016

*Stephan, Jens, Rene, Thorsten und Malte*