Software PWM

Oft macht es Sinn, dass man LEDs über softwareseitige PWM-Befehle dimmt. Dazu gibt es z.B. das Code-Beispiel "Knight-Rider" und "Rundumlicht".

Erklärung

Eine PWM bedeutet, dass sich das Puls-Pausen-Verhältnis ändert. Die Summe Pulszeit (high) + Pausenzeit (low) ist immer gleich. Sie können aber natürlich unterschiedlich lang sein. Ist der Puls lang, dann ist die Pause entsprechend kurz. Und umgekeht genauso. Langer Puls = lange Einschaltdauer, kurzer Puls = kleine Einschaltdauer. Einschalten = Led an. Länger die Led eingeschalten und kürzer aus bedeutet im Mittel eine heller leuchtende Led, im Vergleich mit kurz Led an und länger Led aus. Das 'im Mittel' bedeutet, dass das Aus- und Einschalten so schnell aufeinander folgt, dass das Auge nicht mehr mitkommt und ein Eindruck eines unterbrechungsfreies Leuchten entsteht.

Genau das passiert hier

```
'Soft PWM
'Vergleich zwischen frei laufendem Zähler Z und Helligkeits-Wert h
 If h > Z Then led=1 Else led=0
 If z<10 then incr z else z=0
```

Hier wird in einer durch interrupt schnell aufgerufener isr der Zähler von 0 bis 10 durchgezählt. Wird h=5 gesetzt, dann leuchtet die Led bei Z=0..4 und bei z=5..10 ist sie aus. 5 Zeiteinheiten an, 6 Zeiteinheiten aus. Im Mittel also bisschen weniger als halbe Helligkeit. Bei h=11 wäre die Led immer an, bei h=0 immer aus.

Software-PWM heißt, du musst die PWM von Hand machen. Also den Zähler nachbilden mittels Variable und eine 2. Variable als Vergleichs-Wert vorhalten.

Dann nimmst du einen Timer und lässt dir einen Interrupt generieren mit z.B. 1000 Hz. In der ISR inkrementierst du dann die Zähler-Variable. Ist die Variable 1 Byte groß würde die Überlaufen nach 255 und wieder 0 sein.

Dann vergleichst du die Zähler-Variable mit dem Vergleichswert. Ist das Gleich z.B. Pin setzen. Ist der Wert 0 dann Pin löschen. Oder auch umgedreht.

So hättest du eine Soft-PWM mit 256 Werten (8-Bit). PWM-Frequenz wäre in dem Fall ISR-Frequenz / 256 = 1000Hz / 256 = 3,9Hz

So funktioniert SoftPWM.

Wenn du den Kitt nachmachen willst brauchst du pro LED ein Variable (Match), und natürlich eine Zähler-Variable.

Übrigens muss man nicht bis 256 zählen will, man kann die Zählervariable auch bei >100 auf 0 setzen als Beispiel. Das wäre dann eine PWM, die Werte von 0 bis 100 als Werte zulässt.

Software-PWM heißt, du musst die PWM von Hand machen. Also den Zähler nachbilden mittels Variabler und eine 2. Variable als Vergleichs-Wert vorhalten.

Dann nimmst du einen Timer und lässt dir einen Interrupt generieren mit z.B. 1000 Hz. In der ISR inkrementierst du dann die Zähler-Variable. Ist die Variable 1 Byte groß würde die Überlaufen nach 255 und wieder 0 sein.

Dann vergleichst du die Zähler-Variable mit dem Vergleichswert. Ist das Gleich z.B. Pin setzen. Ist der Wert 0 dann Pin löschen. Oder auch umgedreht.

So hättest du eine Soft-PWM mit 256 Werten (8-Bit). PWM-Frequenz wäre in dem Fall ISR-Frequenz / 256 = 1000Hz / 256 = 3,9Hz

So funktioniert SoftPWM.

Wenn du den Kitt nachmachen willst brauchst du pro LED ein Variable (Match), und natürlich eine Zähler-Variable.

Übrigens muss man nicht bis 256 zählen will, man kann die Zählervariable auch bei >100 auf 0 setzen als Beispiel. Das wäre dann eine PWM, die Werte von 0 bis 100 als Werte zulässt.

From:
https://wiki.modellbahn-anlage.de/ - **Wiki der Modellbahn-Anlage.de**

Permanent link:
https://wiki.modellbahn-anlage.de/elektronik/software-pwm

Last update: **23.07.2026 18:09**