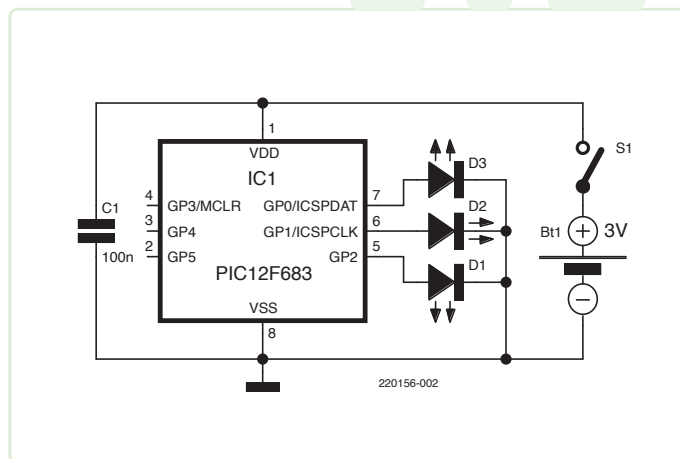




Kaars-simulator

Stefano Purchiaroni (Italië)

Wil je een kaars simuleren? Dat kan met dit eenvoudige project, dat slechts één actieve component en drie LED's gebruikt. Het effect is uiterst realistisch, en de bouw neemt niet meer dan een uurtje in beslag. Zonder print!



Deze kaarssimulator heeft twee gele LED's (5 mm) en een grote rode LED (10 mm) zoals je op de foto's kunt zien. Een papieren (closet-)rol rond de schakeling maakt de elektronische kaars compleet.

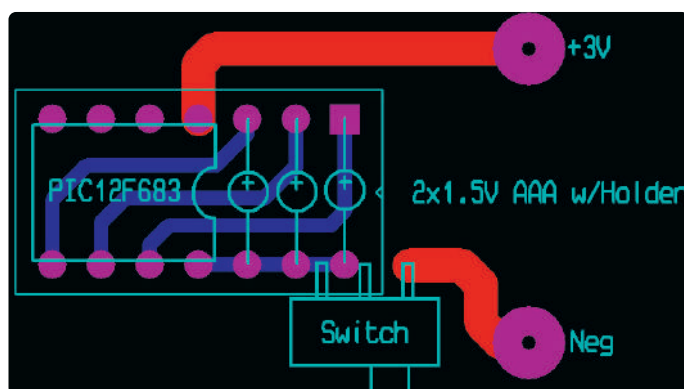
Het schema kan eigenlijk niet eenvoudiger. Je hoeft slechts een Microchip-PIC te programmeren met het HEX-bestand uit de download. Het spreekt vanzelf dat de broncode ook in de download is opgenomen, voor wie daarin geïnteresseerd is. Delen van de code zijn te zien in **listing 1**. De helderheid van elke LED wordt in hoog tempo gevarieerd; dit wordt met een timer getriggerd, met een herhaaldelijk willekeurig ingestelde aflooptijd. De nieuwe helderheidswaarden (geregeld door PWM-signalen op de uitgangen) worden ook willekeurig gekozen – binnen bepaalde hard-gecodeerde grenswaarden. De PWM-signalen worden softwarematig gegenereerd.

Op de foto's zie je mijn constructie, opgebouwd op een 14-polig IC-voetje: de PIC12F683 neemt 8 pinnen in beslag, terwijl de andere zes gebruikt kunnen worden om te experimenteren met verschillende lichteffecten (je kunt bijvoorbeeld de drie LED's vervangen door exemplaren met andere kleuren).

De batterijhouder (2 stuks 1,5-V AAA-batterijen) ligt ondersteboven en vormt zo een stevige steun; de schakeling wordt dan op de vlakke onderkant van de houder bevestigd met een stuk dubbelzijdige tape. De closetrol kan met wat doorzichtige tape worden vastgezet.

Alkaline AAA-batterijen hebben voldoende capaciteit om de kaars een paar dagen ononderbroken te laten werken. Je kunt ook oplaadbare NiMH-accu's gebruiken, aangezien de schakeling ook bij 2 V nog functioneert. ◀

220156-03



↓ Project-download



www.elektormagazine.nl/summer-circuits-22

WEBLINKS

[1] S. Purchiaroni, projectvideo: https://www.youtube.com/watch?v=wmvdNMYtW_s

[2] Hex-code op de website van de auteur: <http://www.purchiaroni.com/progetti/Candela/Candela.hex>



Listing 1. Gedeelten van de broncode.

```
#define PWM_MIN      110
#define PWM_VAR      145
#define TIM_MIN      200
#define TIM_VAR      400

unsigned short p[6];           // Current PWM duty per each LED 0..5
unsigned int  t=0;             // Remaining time (tics) before duty change
unsigned short pulse=0;        // Current pulse in PWM block [0..255]

...

void OutPwm() {
// Turn on/off all LEDs according to current PWM setting per each LED

unsigned short i;
for (i=0;i<NUMLED;i++) {
    if (pulse<=p[i]) setbit(GPIO,i); else clearbit(GPIO,i);
}
}

...

void SetPwm() {
// Set a new random pwm duty cycle value per each LED
// This function is called at duration expiration
unsigned short i;
for (i=0;i<NUMLED;i++) {
    p[i] = PWM_MIN+random(PWM_VAR);
}
}

void interrupt() {

...

    if (t==0) {                // Duration expired
        SetPwm();              // Change all duty cycles on a random base
        t = TIM_MIN+random(TIM_VAR); // Set a new duration
    }

    t--;                       // Decrease remaining duration

...
}

void main() {

...

    while (True) {
        pulse++;               // Count 256 pulses in a PWM cycle
        OutPwm();              // Apply current PWM duty(s)
    }
}
```

