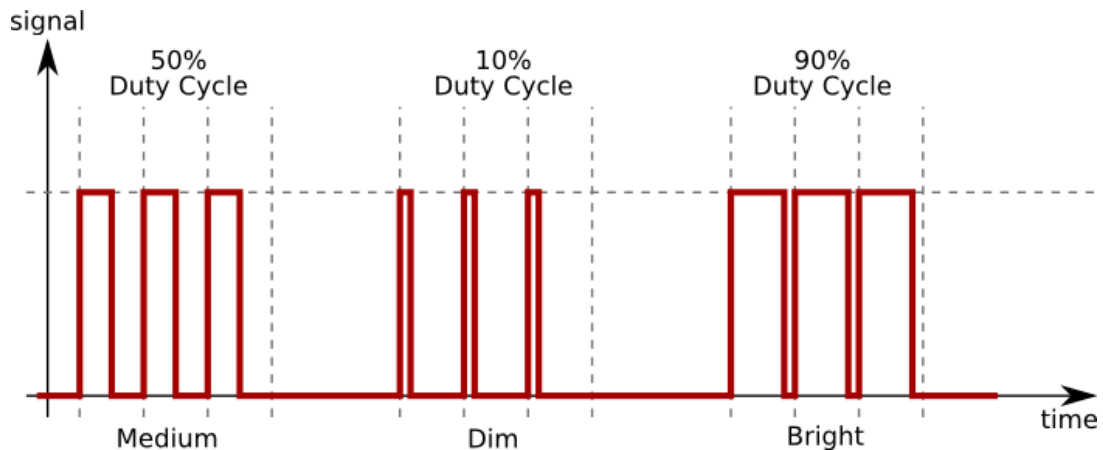


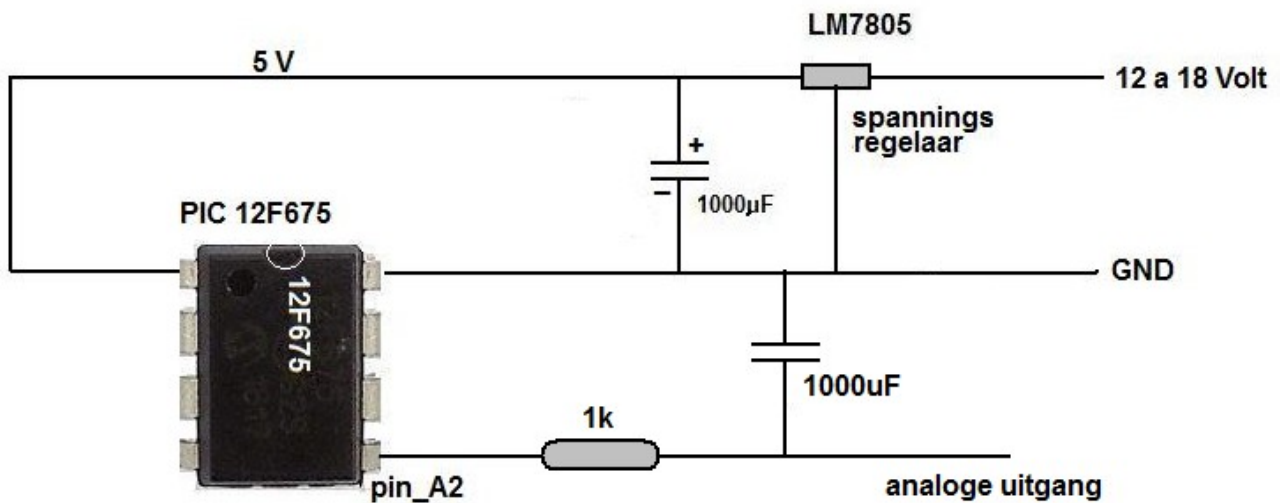
Zelf PIC's programmeren

Deel-7 PWMsignaal

Een PWMsignaal is een signaal dat uit korte pulsen bestaat.
PWM = Pulse Width Modulation



Door deze pulsen met een condensator af te vlakken verkrijg je een redelijke gelijkspanning.



Sommige PICs hebben een ingebouwde PWM-functie.
De 12F675 mist deze functie, wil je uit de 12F675 een PWM-signaal krijgen dan moet je deze zelf programmeren.

De JALEdit code

```
include 12f675                -- target PICmicro

pragma target clock  4_000_000  -- oscillator frequency

pragma target OSC    INTOSC_NOCLKOUT  -- Internal oscillator
pragma target WDT    disabled         -- no watchdog
pragma target MCLR   internal         -- make MCLR pin available I/O
pragma target BROWNOUT disabled      -- no brownout

enable_digital_io()          -- disable analog I/O (if any)
```

```
pin_A2_direction = Output
alias PWM is pin_A2
```

Var Byte Teller, span

```
span= 200
```

```
forever loop
```

```
Teller=Teller+31
```

```
If teller<span then PWM=high else PWM=low end if
```

```
end loop
```

In deze code laten we teller niet met 1 maar telkens met 31 oplopen.
De teller krijgt hierdoor

0 – 31 – 62 – 93 – 125 – 155 – 186 – 217 – 248 – 24 – 55 – 86 – enz

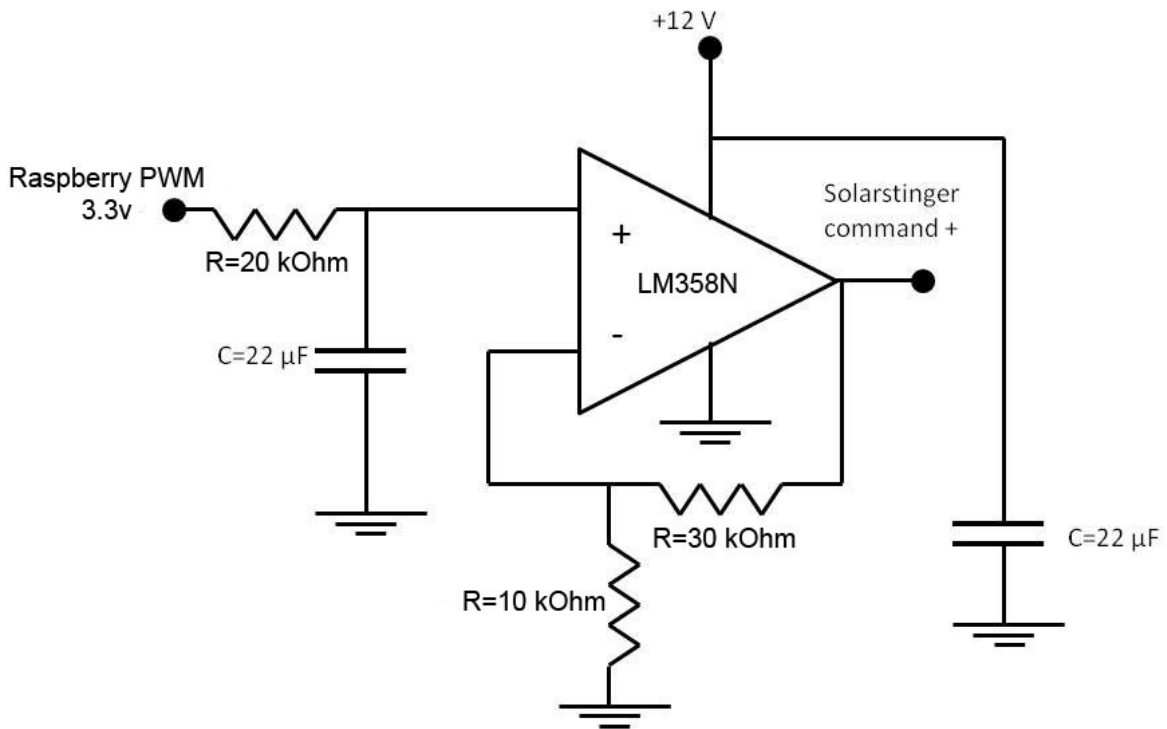
Na 248 springt de teller terug naar 24, dit komt omdat “teller” niet verder kan tellen dan 255.

Als teller een byte (0-255) is, wordt $255 + 1 = 0$.

Op deze manier krijg je een betere verdeling van de pulsen en wordt je analoge signaal nog beter.

Het analoge signaal is nu tussen de 0 en 5 volt, met de waarde van de **span** kun je instellen. Met een $\text{span} = 200$ dan krijg je een uitgang van $200/255 \times 5 = 3,9$ Volt.

Je kunt deze spanning versterken met een “opamp” schakeling.



Met de verhouding van de weerstanden ($R=30\text{ k}\Omega$) en ($R=10\text{ k}\Omega$) bepaal je de versterkingsfactor.

