

Zelf PIC's programmeren

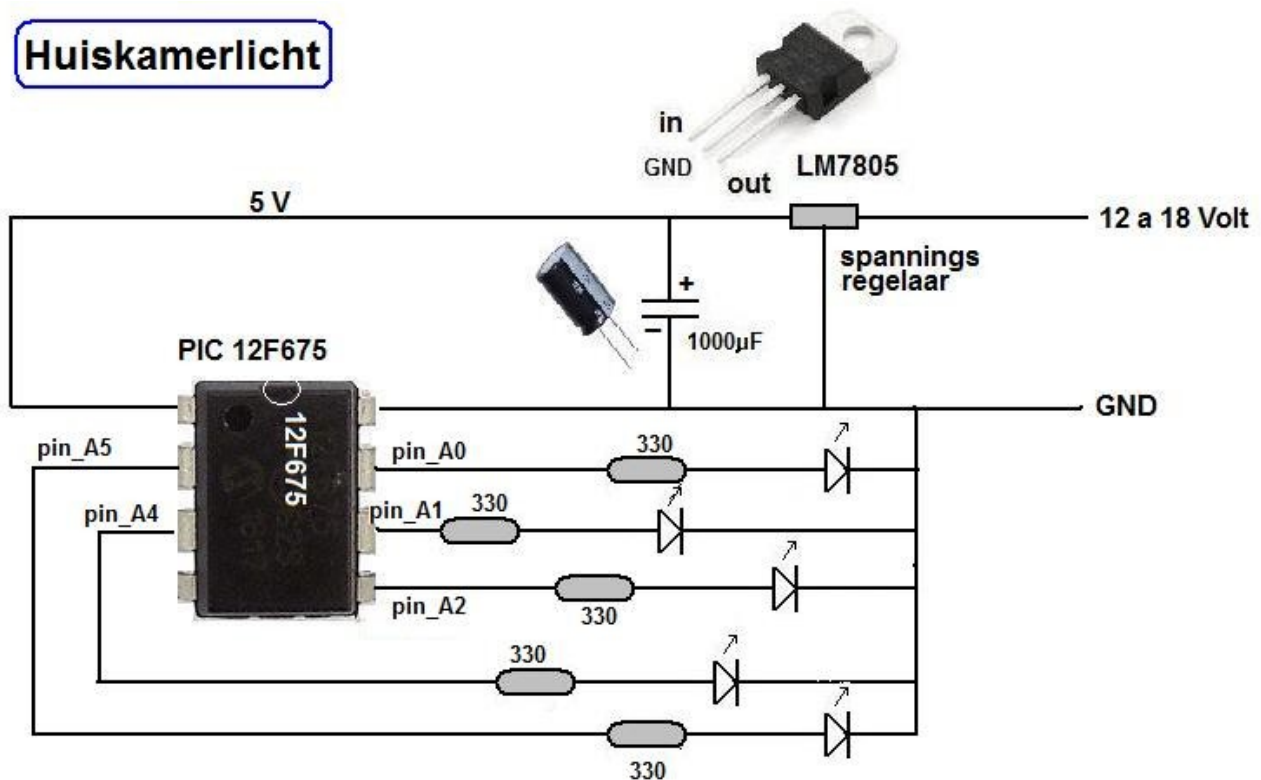
Deel-2 Huiskamerlicht

De verlichting in de huisjes op de modelbaan gaan avonds af en toe aan en uit.



Met de 12F675 kun je 5 van deze lichtjes (LEDs) aan/uit schakelen.

Huiskamerlicht



De 12F675 heeft 6 pinnen waarvan je slechts 5 kan gebruiken als uitgang. Pin_A3 is alleen te gebruiken als ingang.

De JALEdit code hiervoor is:

```
include 12f675
```

```
pragma target clock 4_000_000  
pragma target OSC INTOSC_NOCLKOUT  
pragma target WDT disabled  
pragma target MCLR internal  
pragma target BROWNOUT disabled  
enable_digital_io()
```

```
include delay  
include random
```

```
pin_A0_direction = Output  
alias LED1 is pin_A0
```

```
pin_A1_direction = Output  
alias LED2 is pin_A1
```

```
pin_A2_direction = Output  
alias LED3 is pin_A2
```

```
pin_A4_direction = Output  
alias LED4 is pin_A4
```

```
pin_A5_direction = Output  
alias LED5 is pin_A5
```

```
forever loop
```

```
If 10>Random_Byte then Led1=high end if -- aanzetten  
If 10>Random_Byte then Led2=high end if  
If 15>Random_Byte then Led3=high end if  
If 10>Random_Byte then Led4=high end if  
If 70>Random_Byte then Led5=high end if
```

```
If 10>Random_Byte then Led1=low end if -- uitzetten  
If 40>Random_Byte then Led2=low end if  
If 70>Random_Byte then Led3=low end if  
If 80>Random_Byte then Led4=low end if  
If 10>Random_Byte then Led5=low end if
```

```
Delay_1s(5) -- wacht 5 seconde  
end loop
```

Het programma begint met **include 12F675**, met deze code worden de algemene eigenschappen geladen voor deze PIC.
Dan volgen er nog een aantal extra (standaard) codes.

pragma target clock 4_000_000, hiermee geeft je de snelheid van de klok aan.

pragma target OSC INTOSC_NOCLKOUT, we gebruiken de interne klok van de PIC.
Deze loopt wat langzamer en is ook niet erg nauwkeurig, maar dat maakt in dit project niet uit.

include delay, de delay's worden geladen.

Je hebt de keuze uit een aantal mogelijkheden:

1. delay_10us(n)
2. delay_1ms(n)
3. delay_100ms(n)
4. delay_1s(n)

Include random, de random generator wordt geladen.

Met **Random_Byte** wordt een willekeurig getal gekozen tussen 0 en 255.

Met **If 10>Random_Byte then Led5=low end if**, wordt led5 uitgezet als het getrokken getal kleiner is dan 10.

Met **pin_A0_direction = Output**, wordt pin_A0 ingesteld als output, deze kan hierdoor twee waarden krijgen of hoog (= 5V) of laag (= 0V)

met **alias LED1 is pin_A0**, krijgt pin_A0 een ander naam in het programma, hij heet nu LED1.

In het programma wordt na de **forver loop** het geheel aan en uitgeschakeld.

LED1 wordt aangezet als het getrokken getal kleiner is dan 10.

Dan volgen de andere 4 LEDs.

Daarna wordt de LEDs eventueel weer uitgezet als het getrokken getal laag genoeg is.

Aan het einde wordt 5 seconde gewacht alvorens we we weer opnieuw beginnen.

Delay_1s(5), geeft een wachttijd van 5 seconde, als je deze waarde verhoogt schakeld het licht langzamer aan en/of uit.