

Zelf PIC's programmeren

Deel-11 AHOPoverweg

Bij de AHOP overweg gaan eerst de rode lichten om beurten aan en uit, waarna de slagbomen dalen.

Is de trein gerepasseerd dan gaan eerst de bomen omhoog waarna de lichten pas doven.



De schakeling voor de lichten met een 12F675 is beschreven in “deel-3 Spoorwegovergang”.

In “deel-10 servomotor” staat hoe je een servomotor aanstuurt met een 12F675.

Door deze twee delen aan elkaar te koppelen kun je een complete AHOP-overweg maken.

De 12F675 links wordt de master genoemd en de andere 12F675 de slave.

De master wacht totdat de sensor een melding geeft van een trein.

Hij stuurt dan gelijk de twee lichten om beurten aan en zet de teller op 10.

De teller telt nu langzaam af, bij “teller=7” geeft hij het signaal door naar de slave dat de bomen dicht kunnen.

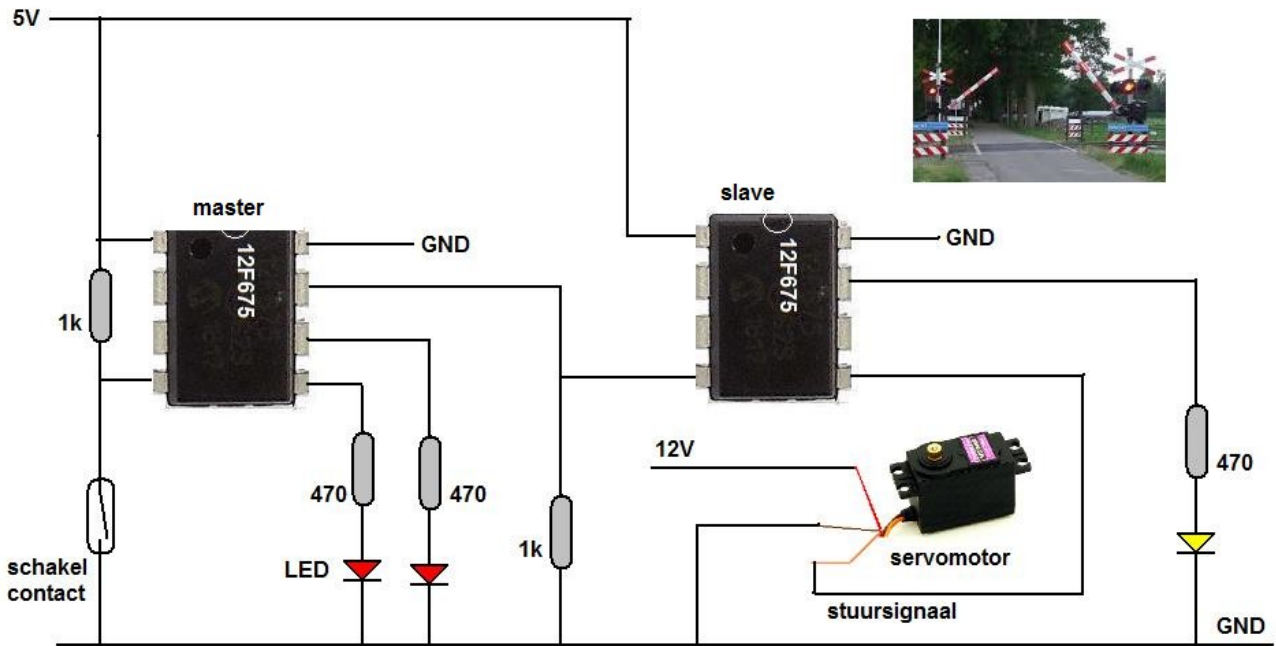
De teller telt vervolgens weer verder af.

Bij “teller=5” wordt gecontroleerd of de trein er nog is, als deze er nog is, wordt de teller telkens even op 6 gezet.

Is er geen trein meer dan telt de teller wel verder af.

Bij “teller=3” mogen de bomen weer open, dit wordt doorgegeven aan de slave.

Na “teller=0” doven de lichten.



De andere 12F675 is de slave, deze werkt zoals beschreven is in deel-10. Deze sluit de bomen als zijn ingang op pin_A3 hoog komt. Hij sluit de bomen door de servomotor aan te sturen via pin_A2. De gele LED is weer ter controle van de 50 Hz. Je hebt te maken met twee 12F675's, en dus ook met twee programma's.

De JALEdit voor de master

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 -- het laden van de delay's

pin_A0_direction = output alias boom is pin_A0

pin_A1_direction = output alias Rood1 is pin_A1

pin_A2_direction = output alias Rood2 is pin_A2

pin_A3_direction = input alias sensor is pin_A3

Var Byte teller

Rood1=low

Rood2=low

forever loop

While sensor==high loop end loop

Teller= 10 -- zet de teller op 10

Repeat

Rood1=high Rood2=low Delay_1s(1) -- het knipperen

Rood1=low Rood2=high Delay_1s(1)

Teller=Teller-1

If Teller==7 then boom=high end if -- boom naar beneden

if Teller==5 & sensor==low then Teller=6 end if -- wacht totdat de trein weg is

If Teller==3 then boom=low end if -- boom omhoog

Until Teller==0

Rood1=low Rood2=low -- doof de lichten

end loop

De code voor de slave.

```
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_A0_direction = Output    alias LED is pin_A0
pin_A2_direction = Output    alias Uit is pin_A2
pin_A3_direction = Input     alias In1 is pin_A3

var word x1, PL, Tel

forever loop

If In1==low then PL=20 else PL=50 end if      -- welke PulsLengte?

If x1>450 then x1=0 Tel=Tel+1 else x1=x1+1 end if -- 450 zorgt voor de 50 Hz

If Tel>49 then LED=!LED Tel=0 end if         -- LED aan/uit

If pl>x1 then Uit=high else Uit=low end if   -- voor de pulzen

end loop
```