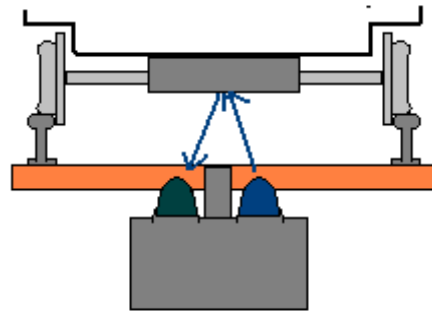


Zelf PIC's programmeren

Deel-13 IR-sensor



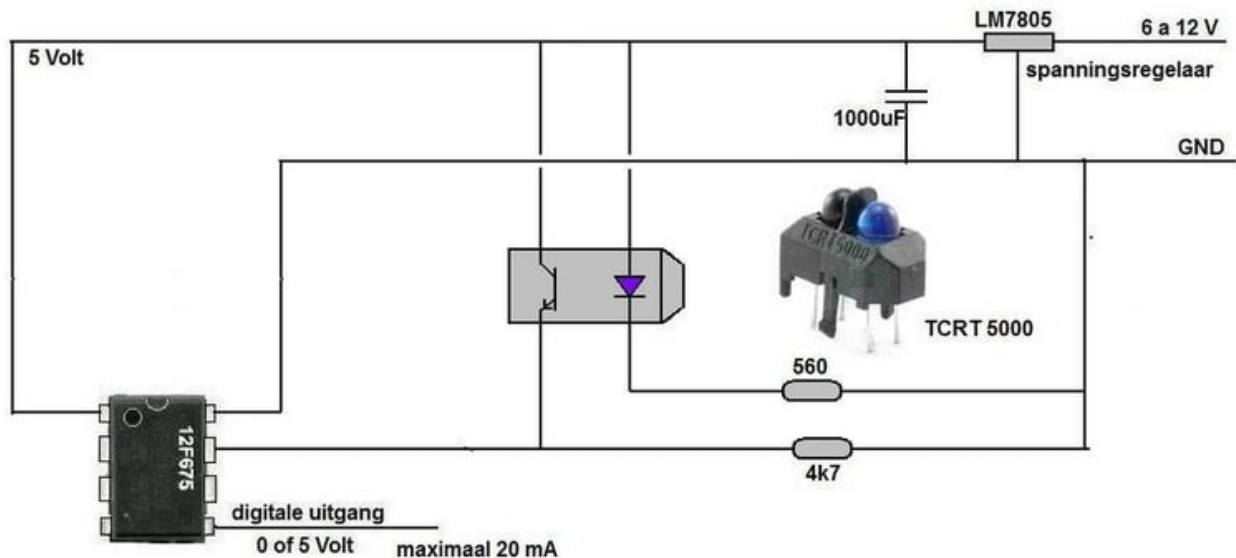
Met de Infrarood-sensor TCTR5000 kun je voorbijkomende treinen detecteren. Plaats de IR-sensor tussen de bielzen van de spoorrails. Door nu een reflectorstrip onder de wagon te plakken heb je een treindetectie.



De uitgang van de TCT5000 komt op de ingang van de 12F675 pin_A0. Als er een trein over IR-sensor rijdt, wordt pin_A0 even hoog.

De waarde op pin_A0 wordt analoog gemeten, hierdoor kun je de gevoeligheid van de sensor naar eigen wens instellen.

Het programma in de 12F675 houdt daardoor zijn uitgang op pin_A2 gedurende minimaal 1 seconde hoog.



De JALEdit code hiervoor is:

```

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)

include delay                                 -- het laden van de delay's

pin_A2_direction = Output    alias Uitgang is pin_A2

const byte ADC_NVREF = ADC_NO_EXT_VREF
const word ADC_RSOURCE = 5_000    -- maximum source resistance = 5 kOhm
include adc
adc_init()

Uitgang=low

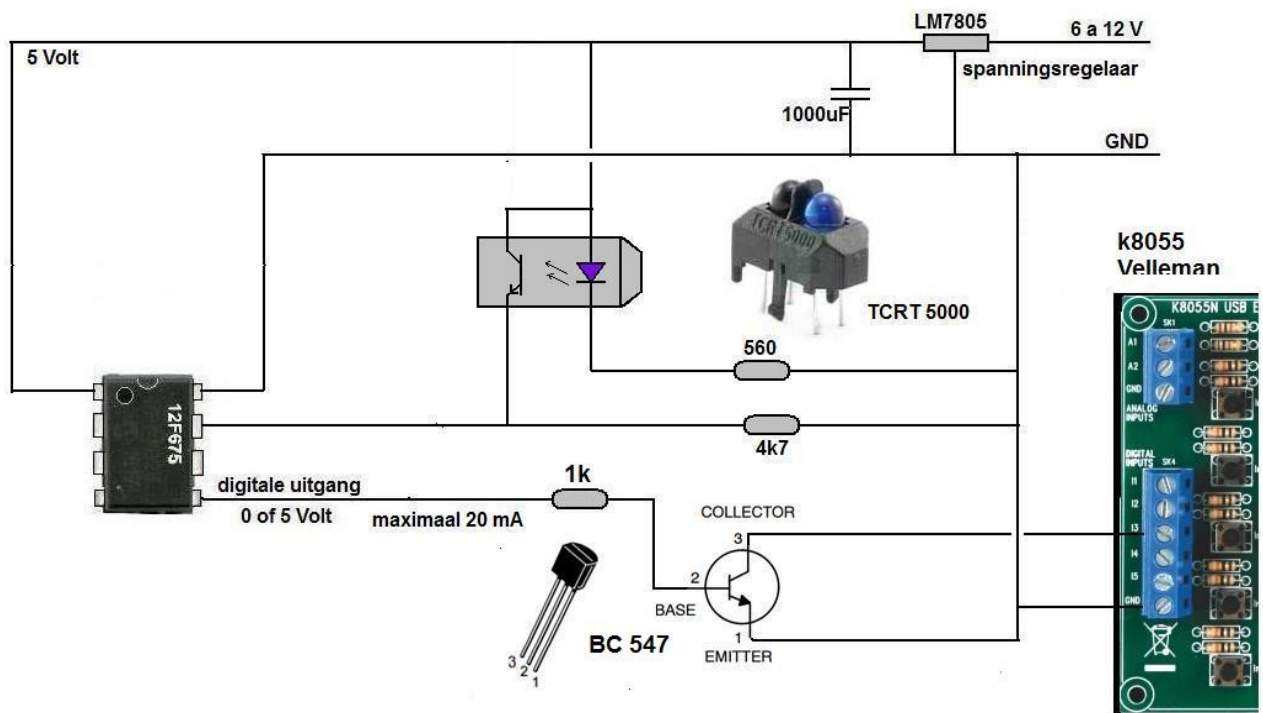
Forever loop
While adc_read_high_res(0)<200    Loop Delay_1ms(1) end loop

Uitgang=high
Delay_1s(2)
Uitgang=low

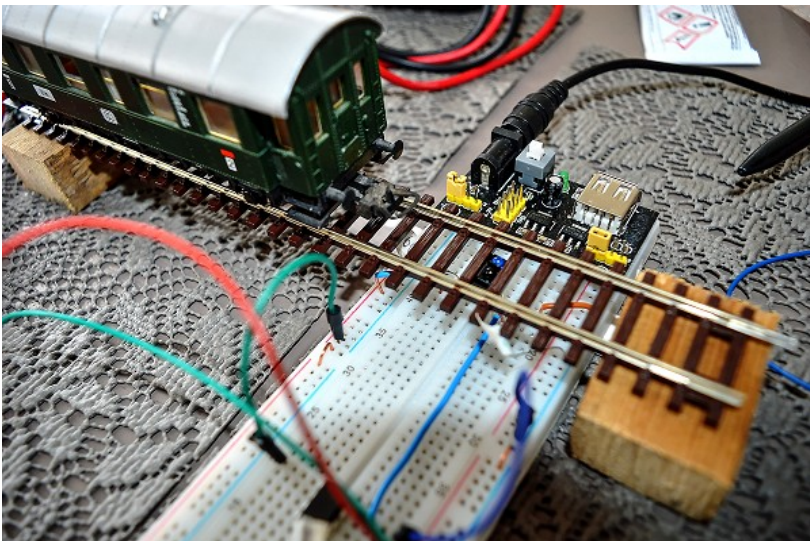
end loop

```

In deze opstelling wordt een I/Okaart (K8055) naar een PC aangestuurd.



De proefopstelling



Met de regel `“While adc_read_high_res(0)<200”` kun je gevoeligheid instellen.

Als `“adc_read_high_res(0)”` hoger is dan 200 verlaat het programma de kleine loop van 1 mS.

`“adc_read_high_res(0)”` kan een waarde tussen 0 en 1023 bevatten.

Zonlicht

Zonlicht kan de werking van de sensor storen, het IR-licht van de zon wordt ook door de sensor opgevangen.



De drempelwaarde (de schakelwaarde van 200) moet worden aangepast als de zon op de baan schijnt.

Je kan ook de code in JALedit aanpassen zodat het programma in de PIC dit nu zelf gaat doen. Je maakt op deze manier een zogenaamde smart-sensor. Een smart-sensor meet niet de hoeveelheid licht maar een snelle verandering van het licht. Als er een trein over de sensor rijdt heb je ineens een grote verandering van het IR-licht.

De PIC meet nu om iedere 10 mS de spanning op pin-0 en stopt die in waarde-W2
Als de spanning iets stijgt laat hij de waarde-W2 ook iets stijgen.
Is de spanning lager, dan laat hij de waarde-W2 weer iets zakken.

Op deze manier maakt hij een "zwevend nulpunt" (W2), dit is de waarde die de PIC op pin-0 normaal meet.

Wanneer er een trein overrijdt is er ineens een grote verandering op pin-0.
Als deze verandering b.v. nu meer dan 70 is, laat hij de uitgang op pin-2 opkomen.

Op deze manier schakelt de PIC nu niet de spanning (de oude 200) op pin-0 maar op hoge verandering van die waarde.

In het programma worden de volgende variabele gebruikt:

W3, is een word (0 t/m 65535)
In deze variabele wordt de oude meetwaarde bewaard.

W1,
in deze variabele wordt de nieuwe meetwaarde bewaard.

S1,
In deze variable wordt het verschil tussen W1 en W2 opgeslagen, maar wel zodanig dat dat altijd een positief getal is.

Het "smart"programma wordt nu:

```
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

include delay                                  -- het laden van de delay's

pin_A2_direction = Output    alias Uitgang is pin_A2

const byte ADC_NVREF = ADC_NO_EXT_VREF
const word ADC_RSOURCE = 5_000
include adc
adc_init()

var word W2, W1, S1

Uitgang=low  W2=50

Forever loop
W1=adc_read_high_res(0) Delay_1ms(10)         -- inlezen tussen 0 en 1023
If W1>W2 then W2=W2+1 end if                   -- stijgen ???
If W1<W2 then W2=W2-1 end if                   -- dalen ???
S1=0
If W1>W2 then S1=W1-W2 end if                  -- het berekenen van S1
If W1<W2 then S1=W2-W1 end if
If S1>70 then                                  -- is S1 meer dan 70, dan moet er een trein zijn
    Uitgang=high
    Delay_1s(2)
    Uitgang=low
end if
end loop
```