

## StoomspoorExpress



**Stoomspoor Express**

**Bestuur je  
analoge  
treinbaan  
vanaf de  
laptop**

## Een Analoge treinbaan aansturen.

Met het programma StoomspoorExpress kun je een analoge treinbaan waarop analoge treinen rijden toch via je laptop aansturen en een dienstregeling uitvoeren.

Maakt diverse stopplaatsen door daar tijdelijk de stroom te onderbreken.

Zonder stroom staat de trein stil en wanneer je er weer spanning opzet gaat de trein weer verder.

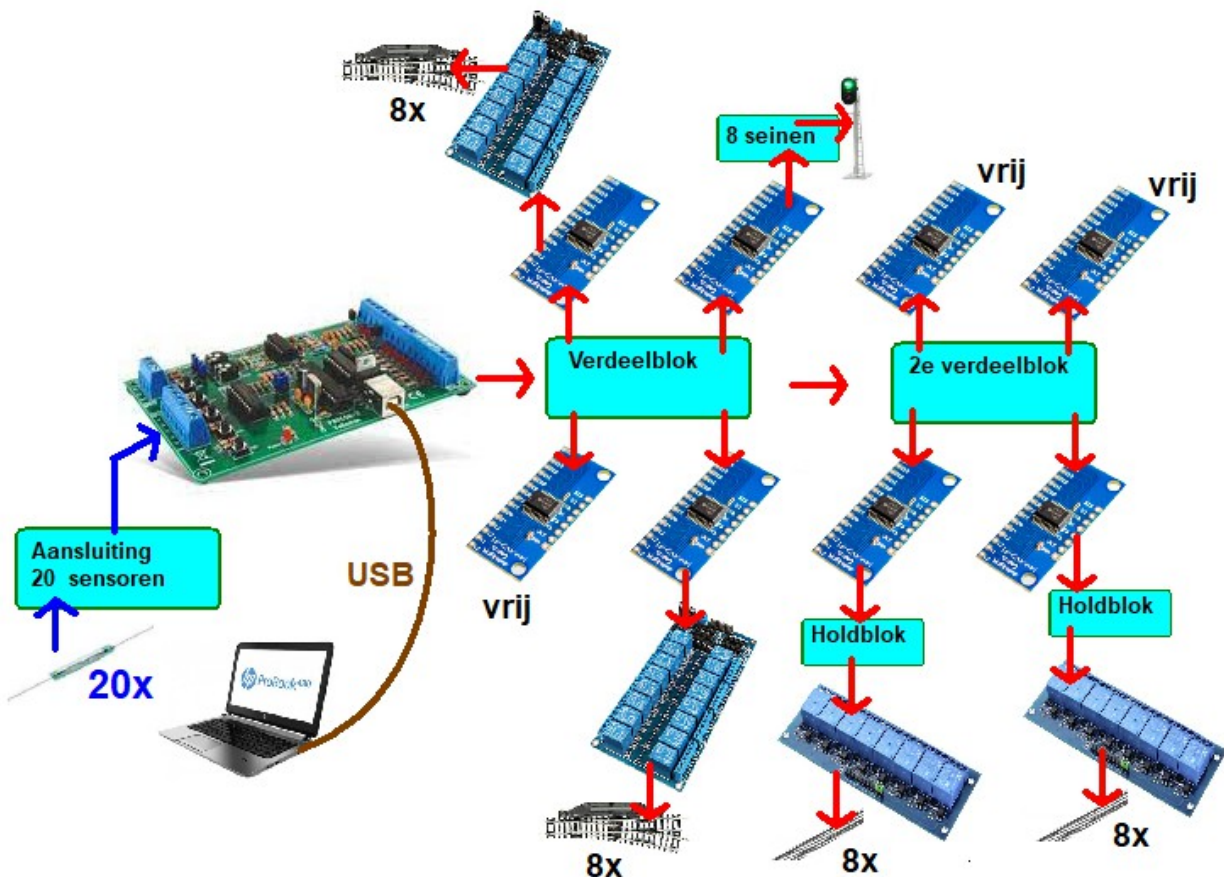
Deze stopplaatsen (blokken) plaats je natuurlijk bij seinen en stations.

Het digitale gebeuren wordt allemaal door je laptop afgehandeld, door het plaatsen van sensoren weet je laptop precies hoe je baan er voor staat.

Het dienstprogramma maak je helemaal zelf door het plaatsen van logische blokjes in het programma.

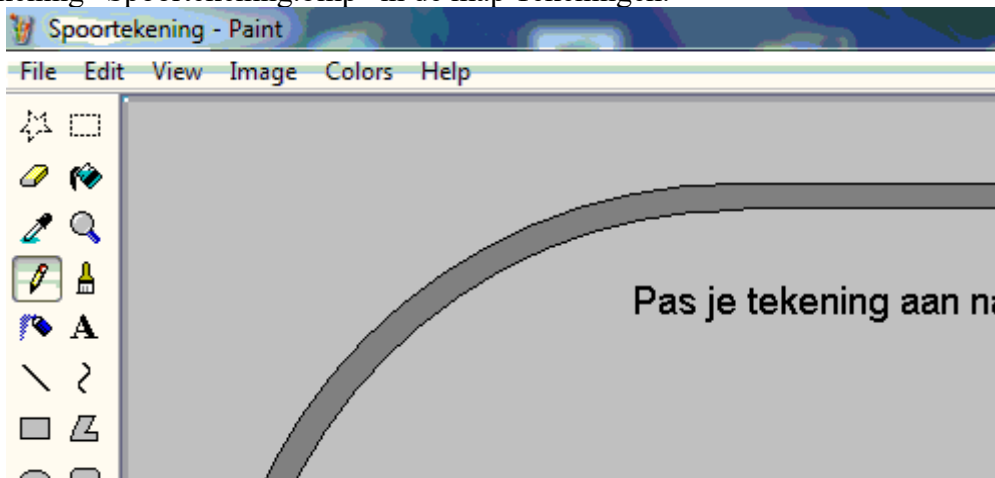


## Principe van de schakelingen

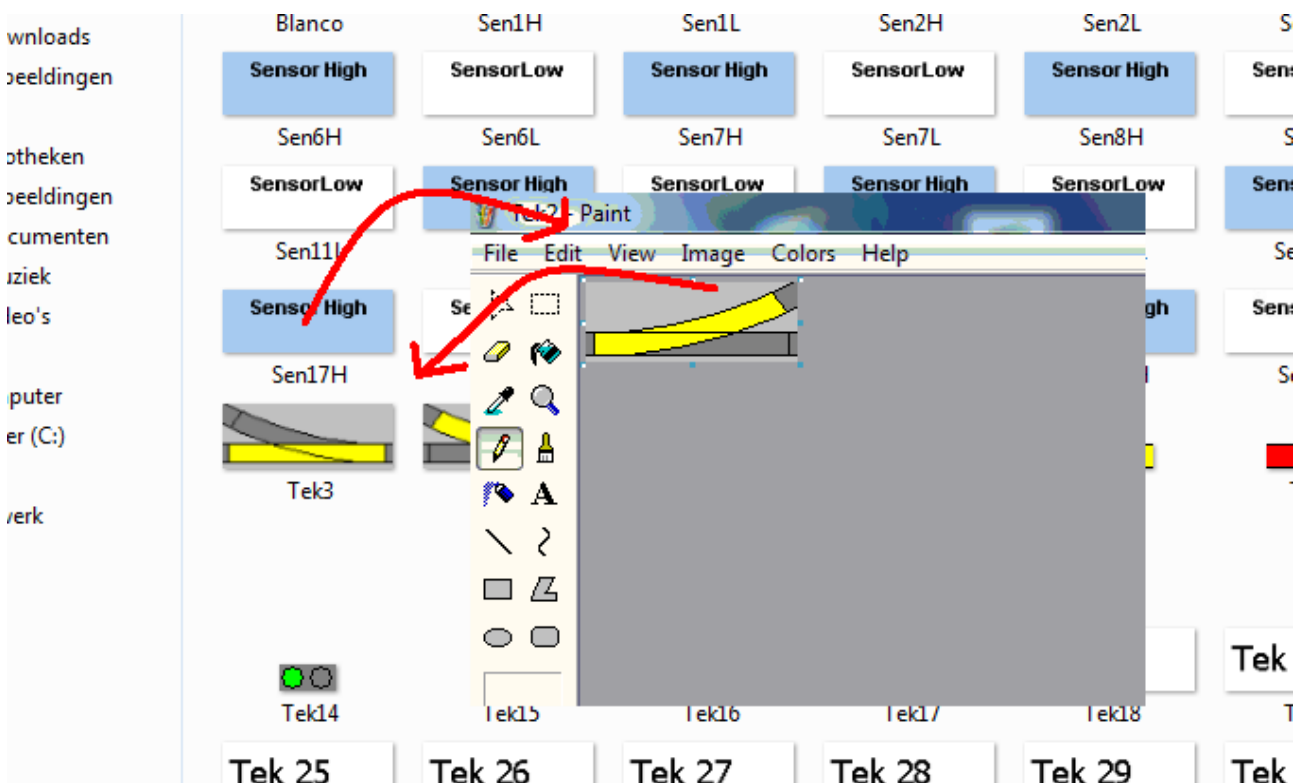


## Het tekenen van de baan

Gebruik een tekenprogramma zoals bv. paint om je eigen baan te tekenen.  
Open de tekening “Spoortekening.bmp” in de map Tekeningen.



Sla je tekening weer op als “Spoortekening.bmp”.  
Gebruik geen JPG, deze extensie wordt niet geaccepteerd.  
Het programma zal bij het opstarten automatisch deze tekening laden.

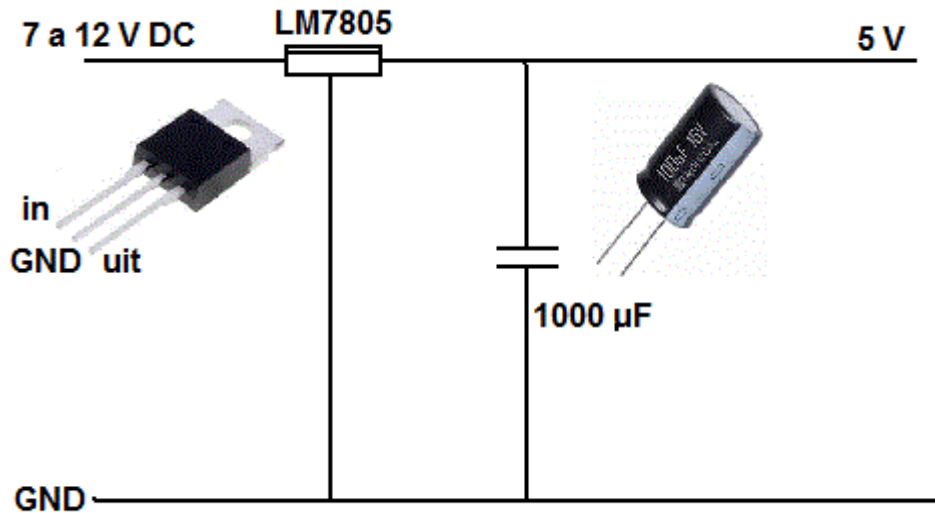


Wissels, blokken en seinen kunnen allemaal op dezelfde manier worden getekend en op de baan worden geplaatst. Ieder item heeft 2 tekeningen hiervoor, voor elke stand een. Als voorbeeld is Tek1.bmp en Tek2.bmp getekend als wissel.

Het programma stoomspooexpress.exe is te vinden via de website:

<http://www.ModelspoorPIC.nl>

## 5 Volt voeding

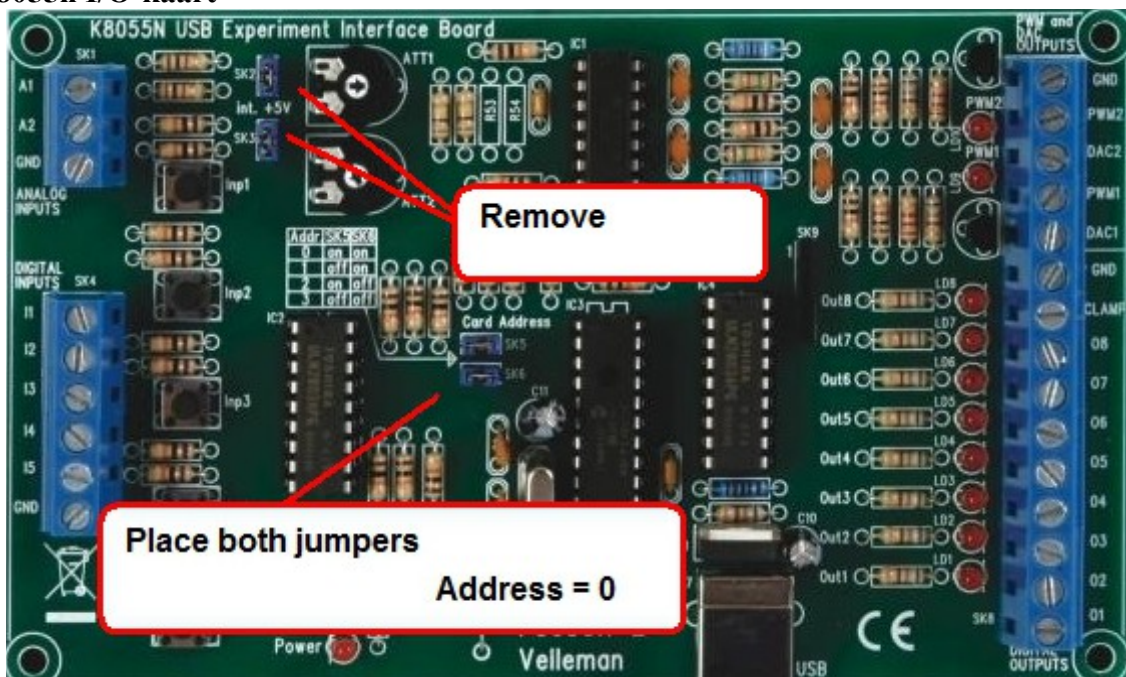


Gebruik een gestabiliseerde voeding voor de 5 Volt gelijkspanning, alle schakelingen werken op deze 5 Volt DC.



De zwarte draad is meestal de GND in de tafervoeding..

## De K8055n I/O-kaart



Met de K8055n is de verbinding vanuit je laptop, controleer de jumpers van de K8055n



## De aansturing van de multiplexers voor het maken van de uitgangen

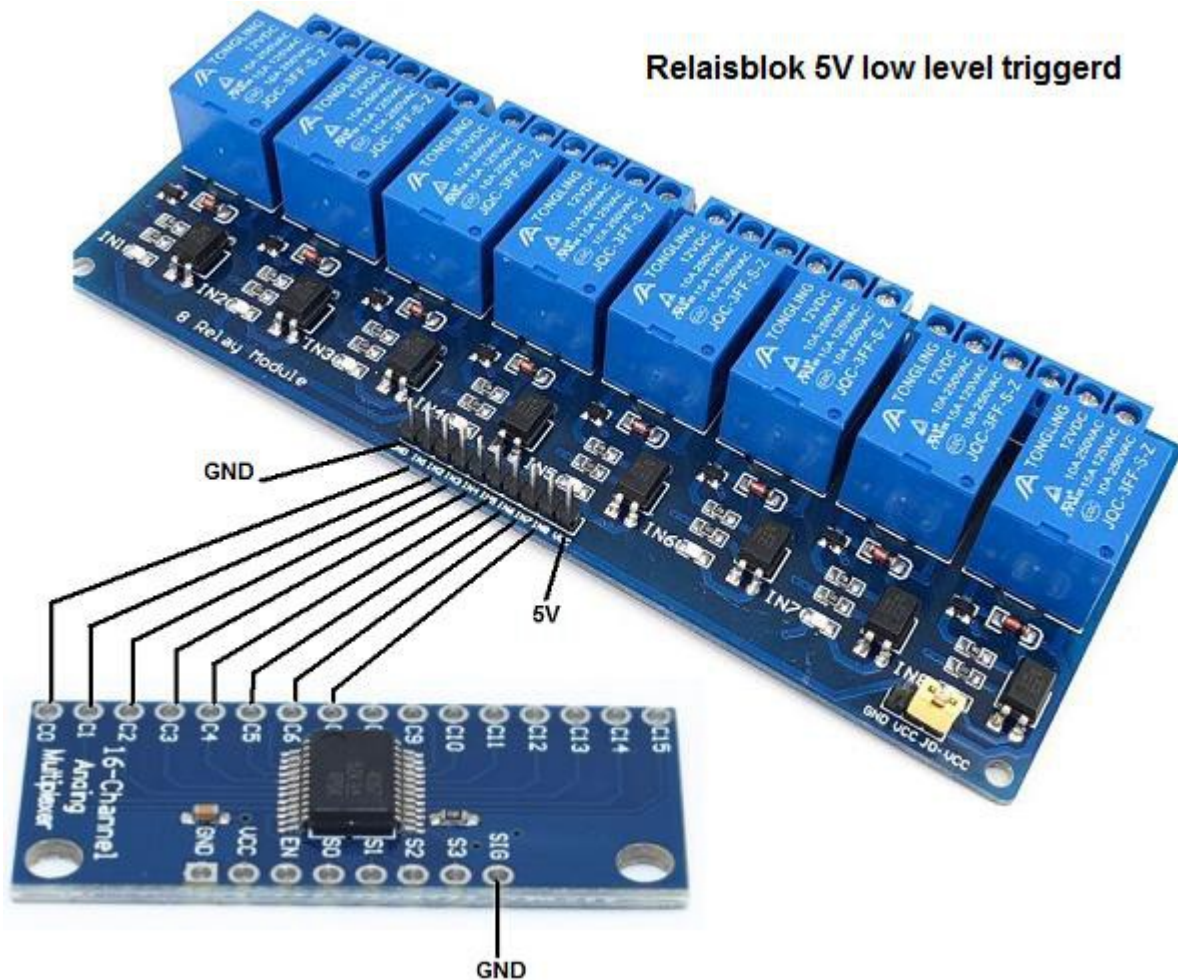
Gebruik maximaal 8 multiplexers zodat je in totaal  $8 \times 16 = 128$  uitgangen hebt.

Voor elke wissel, sein of baanblok heb je tekens 2 uitgangen nodig.

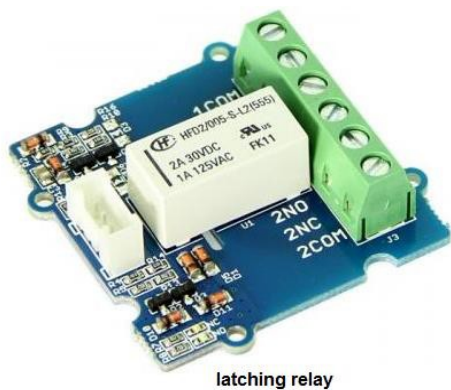
In totaal kun je hiermee 63 wissels, seinen of blokken aansturen (1 paar wordt niet gebruikt).

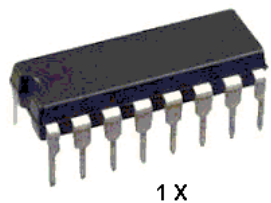
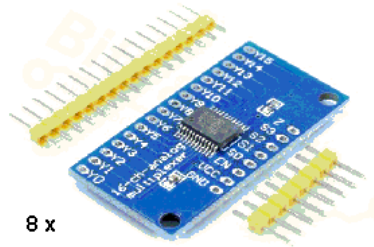
De CD4051be zorgt voor de verdeling naar de multiplexers.

Op elke multiplexer (low triggerd) kun je een relaisblok aansluiten. Gebruik **holdrelais (latching**

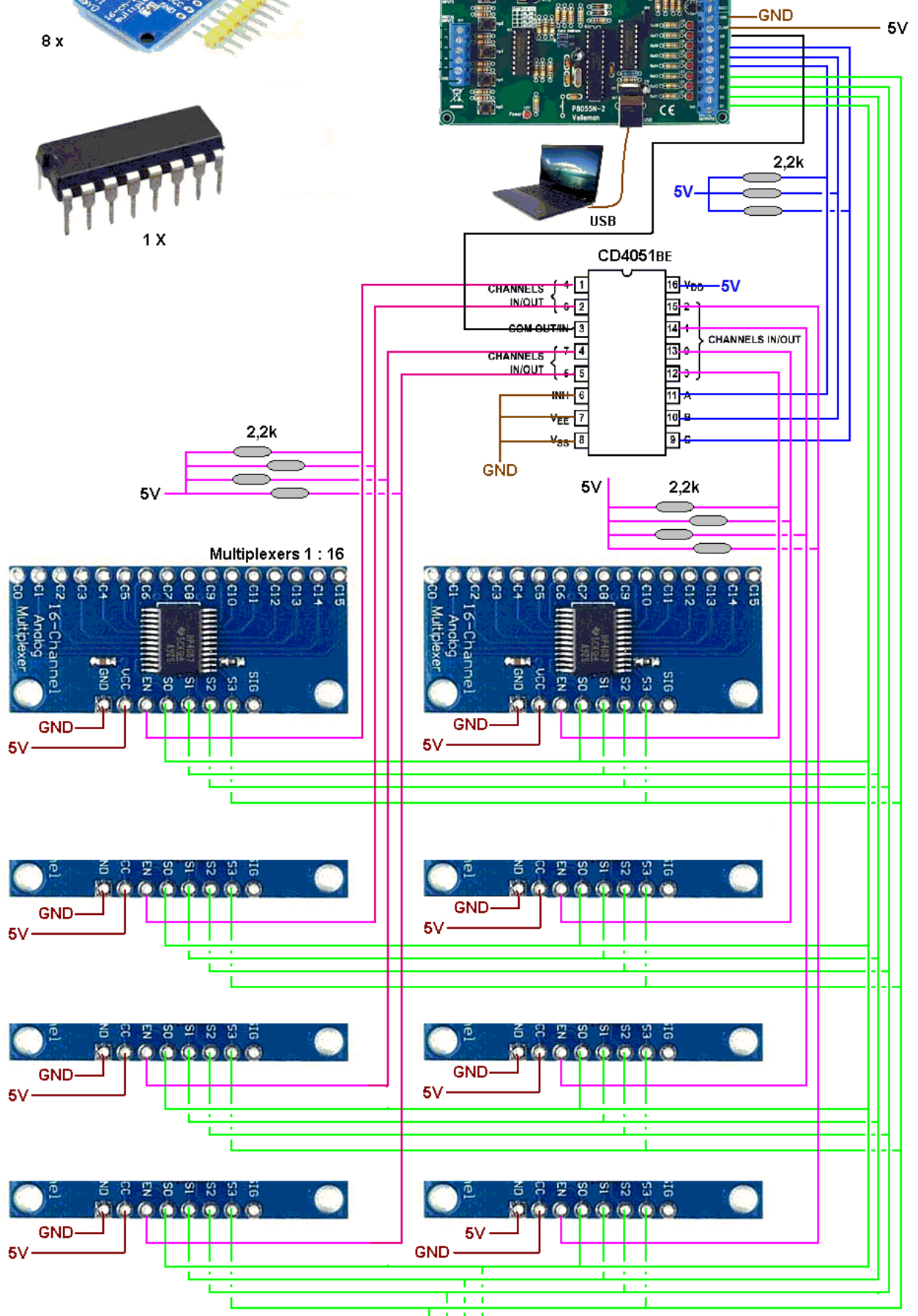
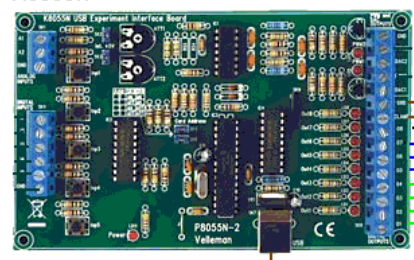


**relays**) om de blokken en seinen te maken, deze zijn in verschillende varianten te verkrijgen of te maken (er is ook een schakeling met 8-relaisblok mogelijk).

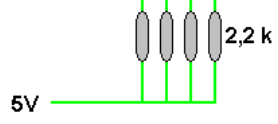




K8055n

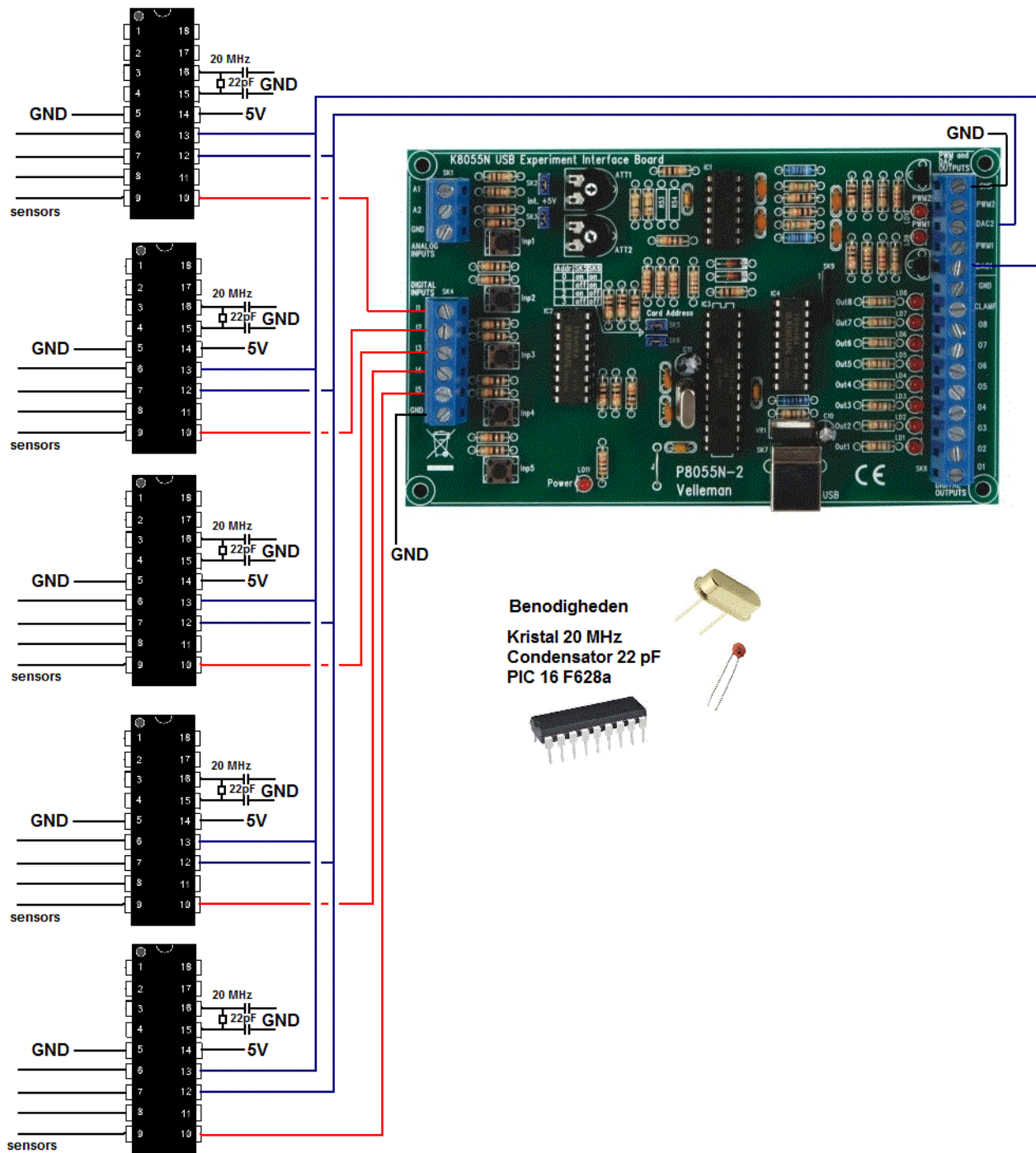


All resistors are 2,2 k  
Voltage should be 5V



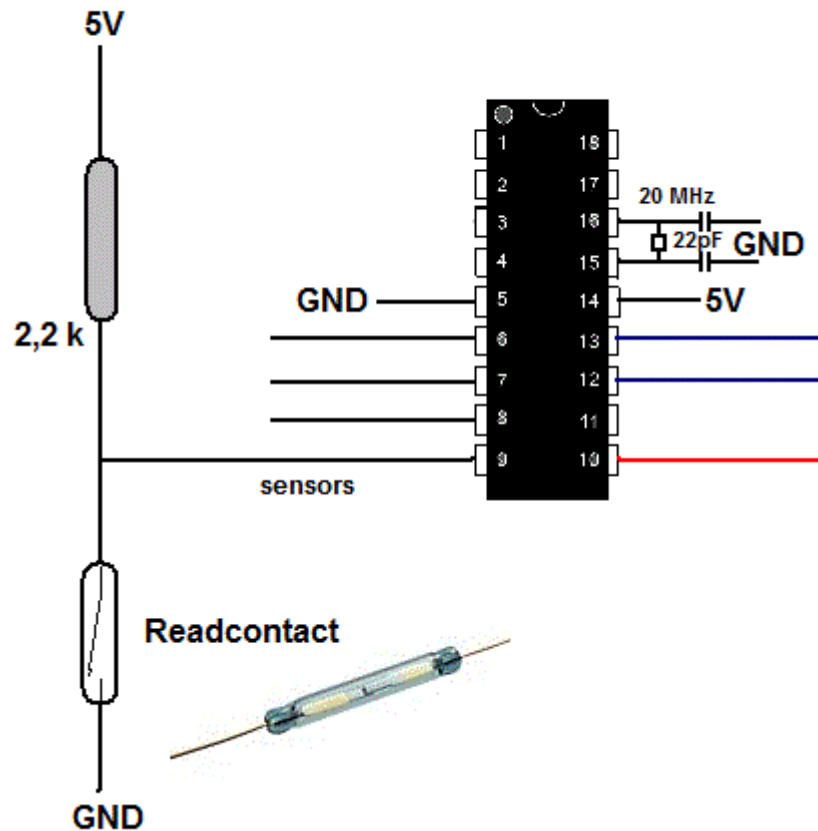
## De sensoringen

Gebruik de DAC1 en de DAC2 uitgang van de K8055n.



Zet de andere aansluiting op de ingangen van van je K8055n.  
De PIC 16628a heeft een 20 Mhz kristal nodig.

**De sensoren worden aangesloten op de 16F628a**



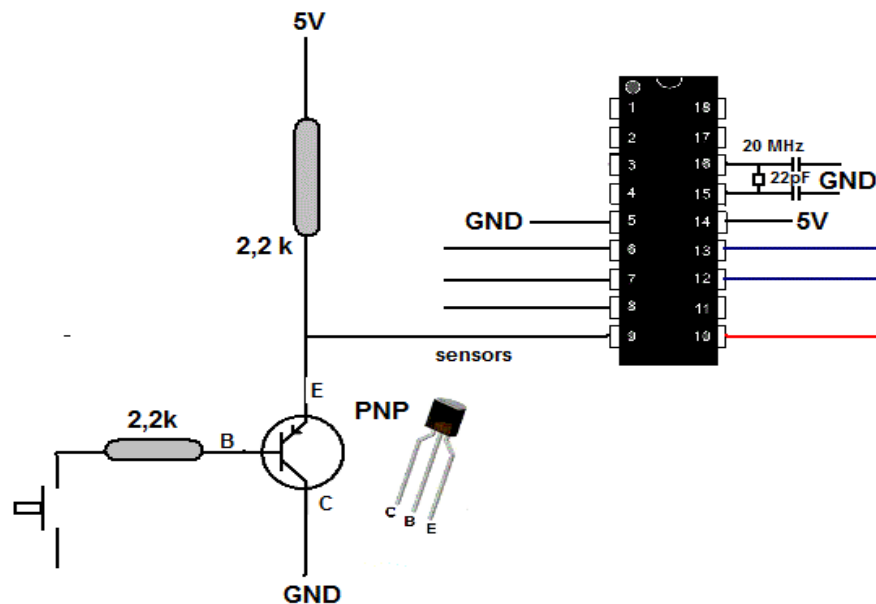
Op elke 16F628a kunnen 4 sensoren worden aangesloten.

Met 5 x 16F628a heb je in totaal 20 sensor ingangen.

Op pin15 en pin16 is een kristal aangesloten van 20.000 kHz en twee condensatoren van elk 22 pF.

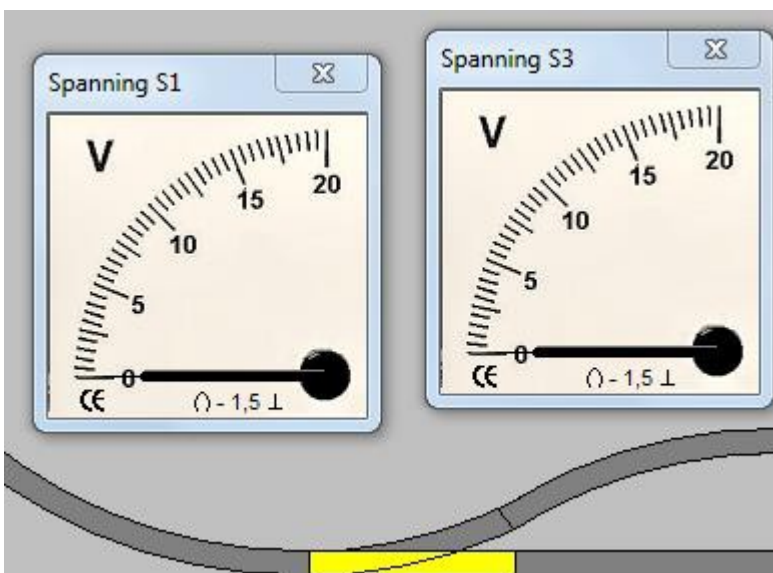
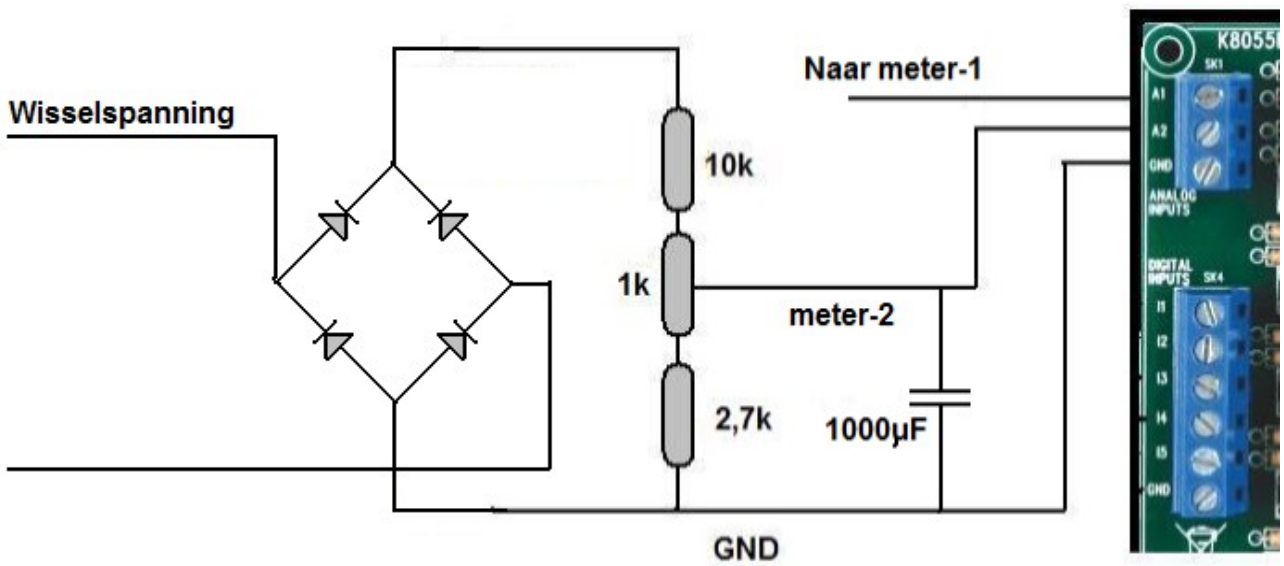
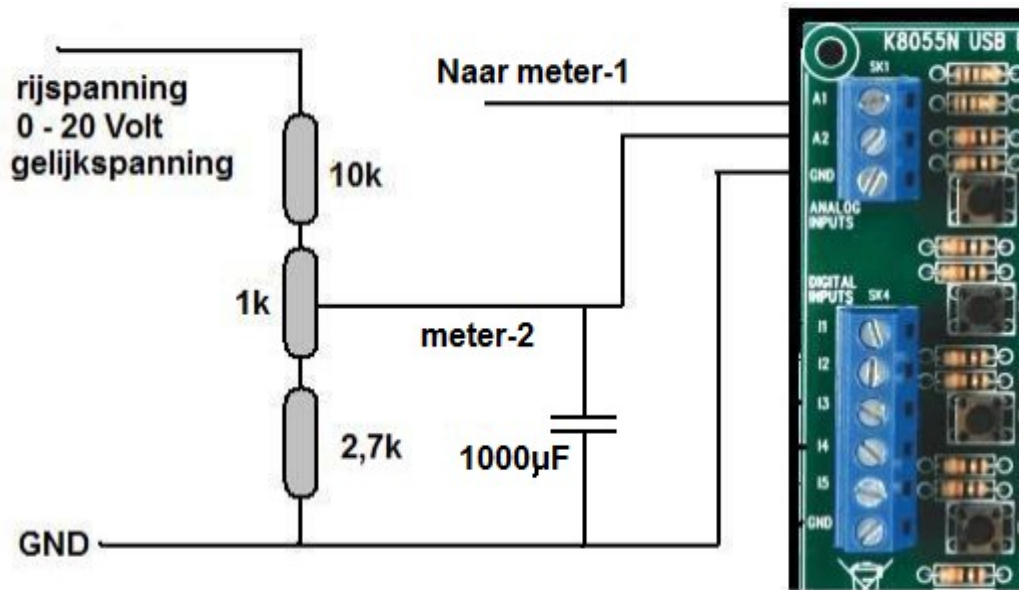
Zorg dat de sensor de pin leeg trekt van het ic, zoals dit is getekend.

In plaats van een reedcontact kun je ook een ander contact gebruiken zoals een bv. optocoupler of schakelaar met of zonder gebruik te maken van een PNP-transistor.





# Het inlezen van de voltmeters



## Het maken van een treinbaan

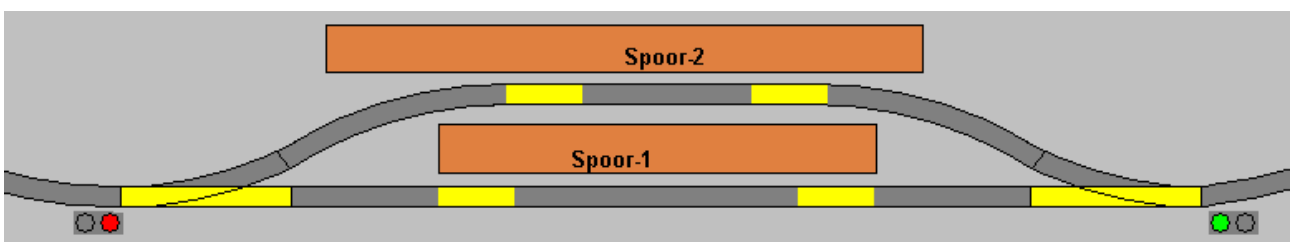
Het eerste ontwerp is hier een simpele baan met twee opstelsporen bij het station. De tekening "Spoortekening.bmp" moet daarvoor worden aangepast.



In dit ontwerp komen 2 wissels, 2 seinen, 4 blokken en eventueel 2 sensoren als voorbeeld.

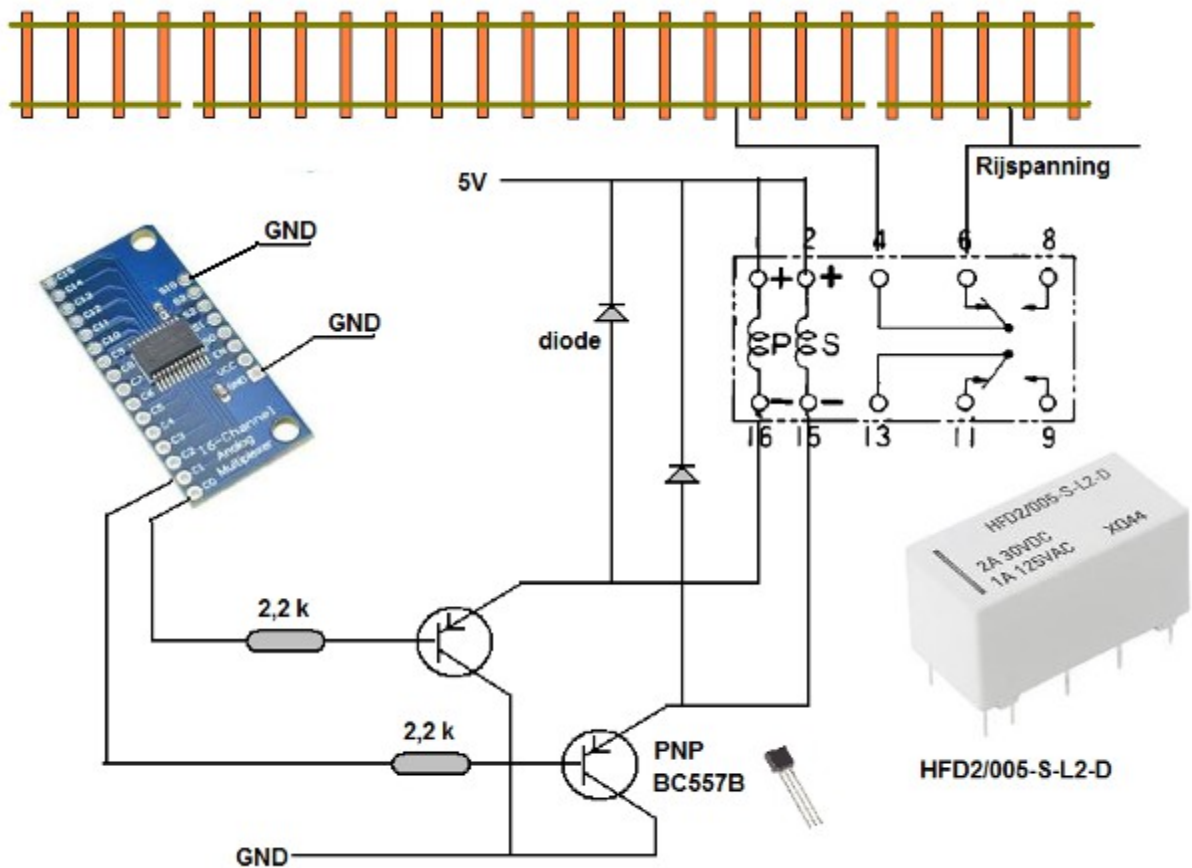
	aansluiting	tekening	naam
Wissel1 links	Pin 0 - 1	Tek1 – Tek2	W1-Re W1-Om
Wissel2 rechts	Pin 2 - 3	Tek3 – Tek4	W2-Re W2-Om
Blok1 links boven	Pin 16 - 17	Tek17 – Tek18	B1-on B1-off
Blok2 rechts boven	Pin 18 - 19	Tek19 – Tek20	B2-on B2-off
Blok3 links onder	Pin 20 - 21	Tek21 - Tek22	B3-on B3-off
Blok4 rechts onder	Pin 22 - 23	Tek23 – Tek24	B4-on B4-off
Sein1 links	Pin 32 - 33	Tek33 – Tek34	S1Red S1Grn
Sein2 rechts	Pin 34 - 35	Tek35 – Tek36	S2Red S2Grn
Sensor1 links		Sen1L – Sen1H	
Sensor2 rechts		Sen2L – Sen2H	

De volgende stap is het aanpassen en plaatsen van de (sein, blok of wissel-) tekeningen. Ook deze tekeningen staan in de map "tekeningen".



In de tekening zijn nu de 4 blokken, 2 wissels en 2 seinen geplaatst.

Gebruik voor het aansluiten van blokken Hold-relais, dan kan je dit schema gebruiken.

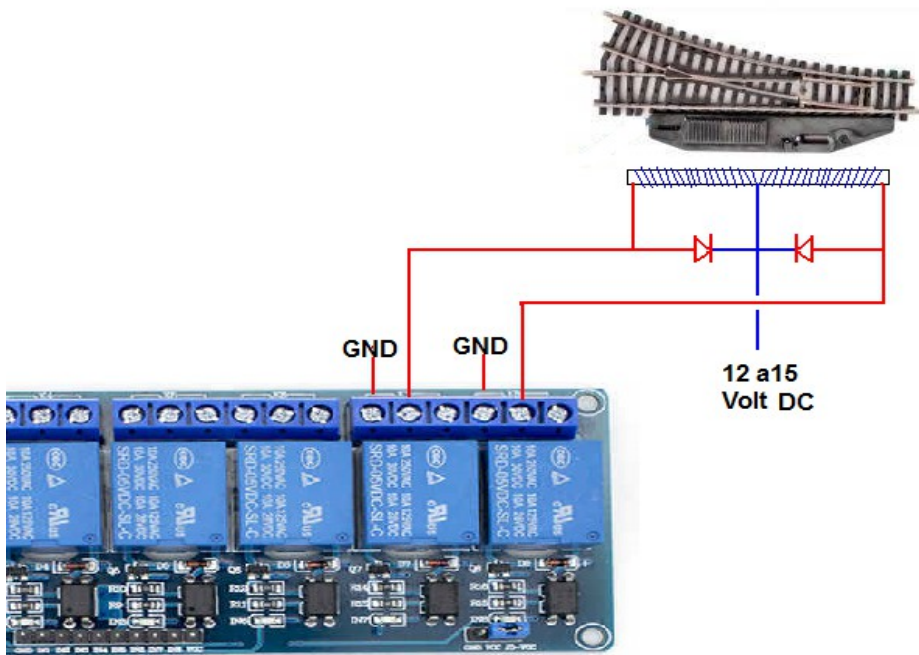


De aansluiting van de wissels, plaats voor ook de 2 blusdiodes.

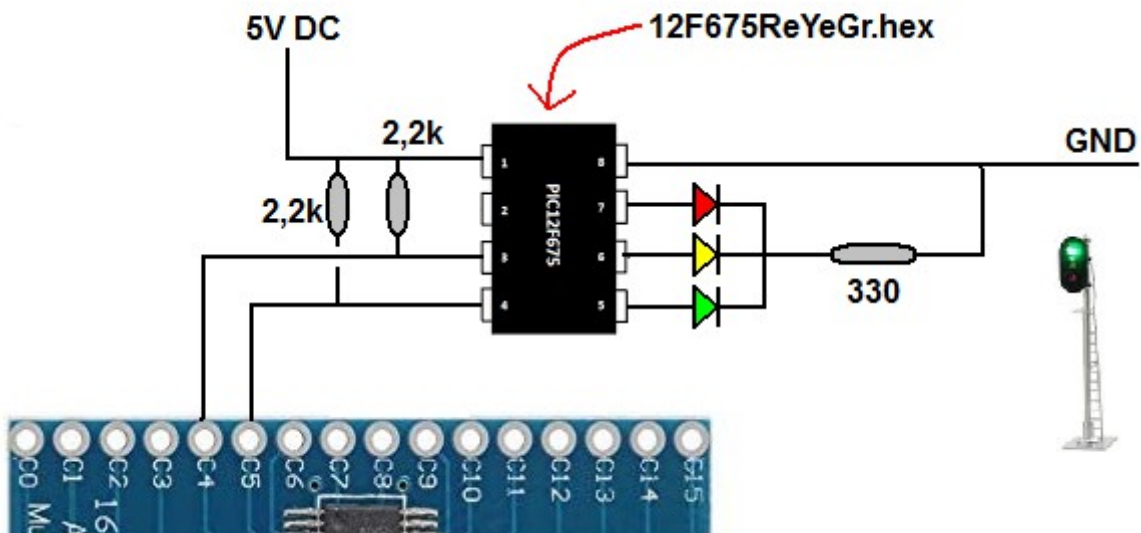
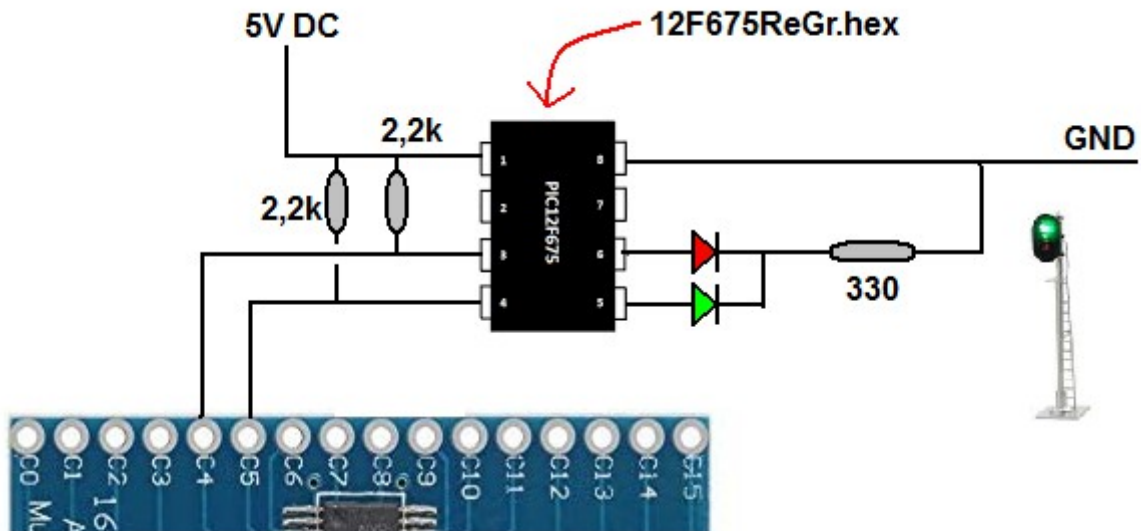
**De seinen kun je maken met een voorgeprogrammeerde pic 12F675.**

Er zijn twee soorten seinen te maken:

- seinen met alleen rood en groen



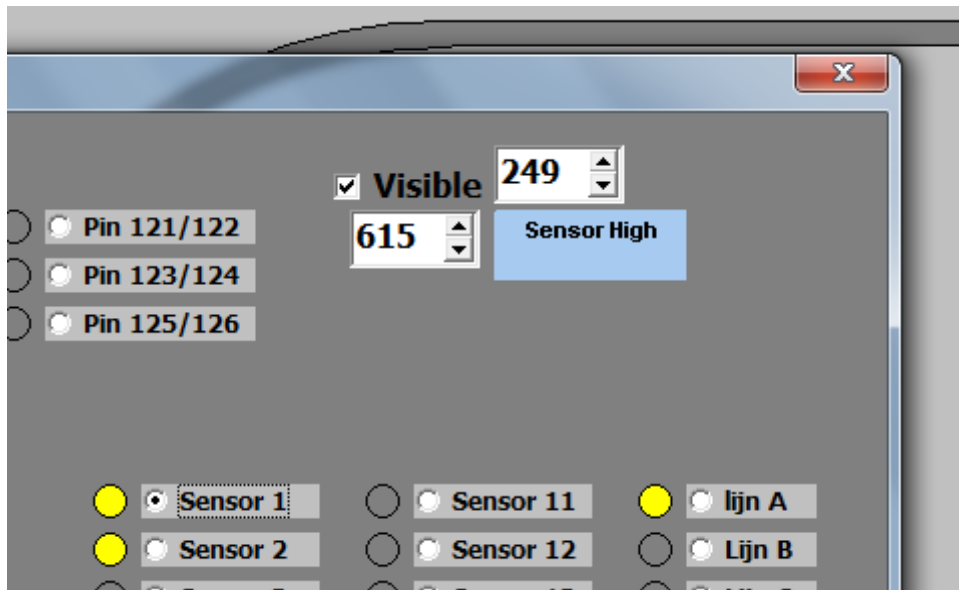
- seinen die vanuit rood via geel naar groen gaan.



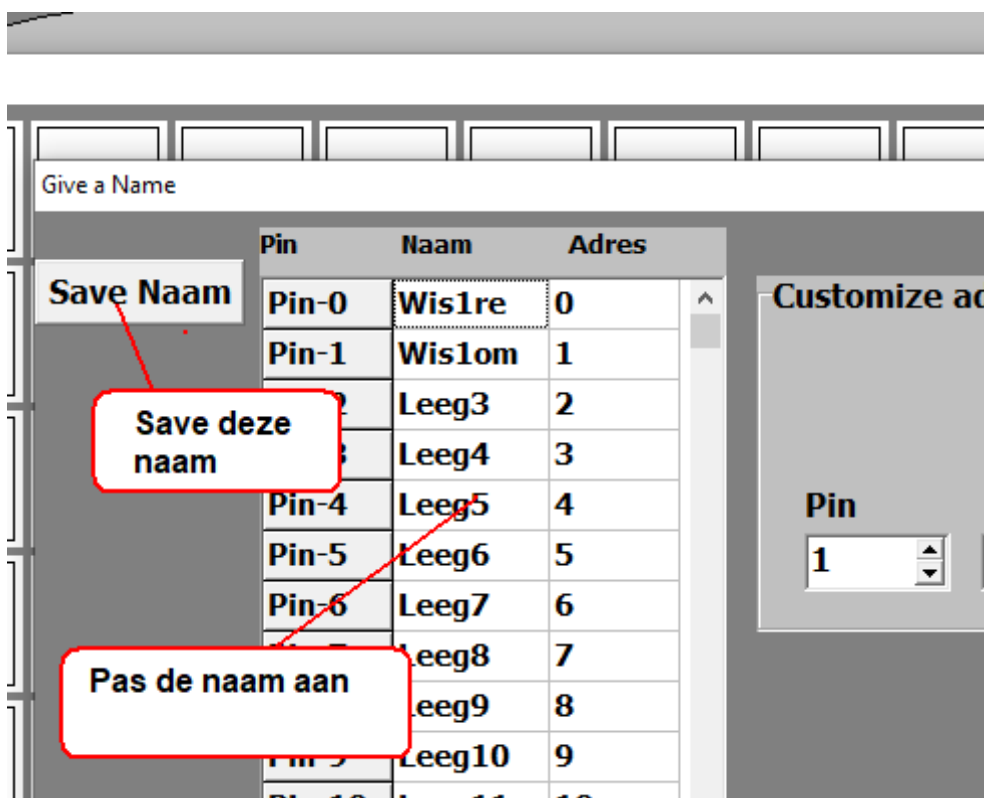
Zorg bij het aansluiten van LED's dat de stroom door de LED ongeveer 10 mA bedraagt.  
 De PIC zelf moet een stabiele voeding van 5 Volt hebben.  
 Zorg er altijd voor dat de stroom doorgang door een LED niet te hoog wordt, 10 mA is meestal voldoende.



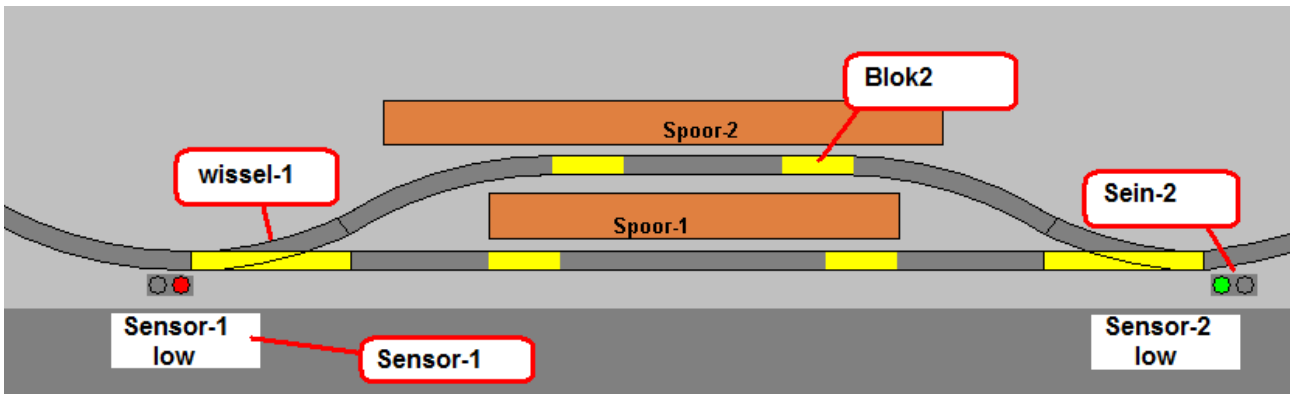
De sensoren **moeten** zichtbaar (Visible) geplaatst worden op je tekening alvorens je ze kunt gebruiken.



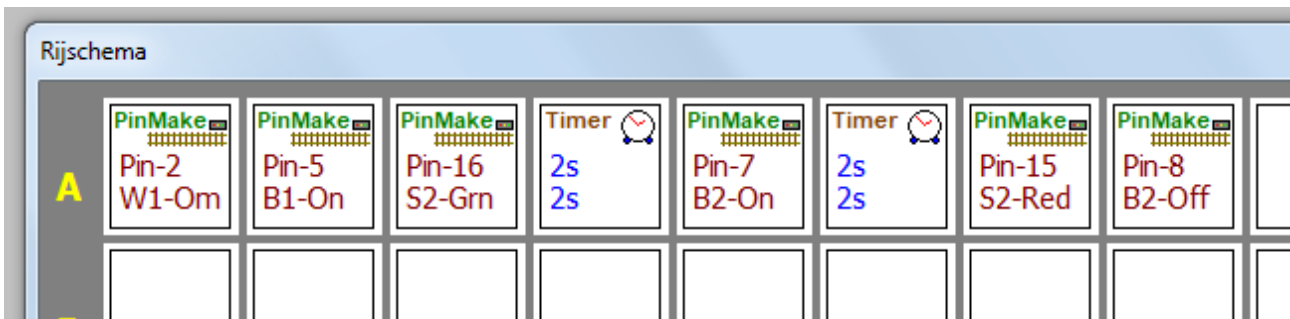
In de tabel hebben alle wissels, blokken en seinen een betere naam gekregen, ook deze nieuwe naam moet in programma worden ingevoerd.



De volgende stap is het programmeren van het Rijschema



Uitgaande dat er een trein klaar staat op blok-2.



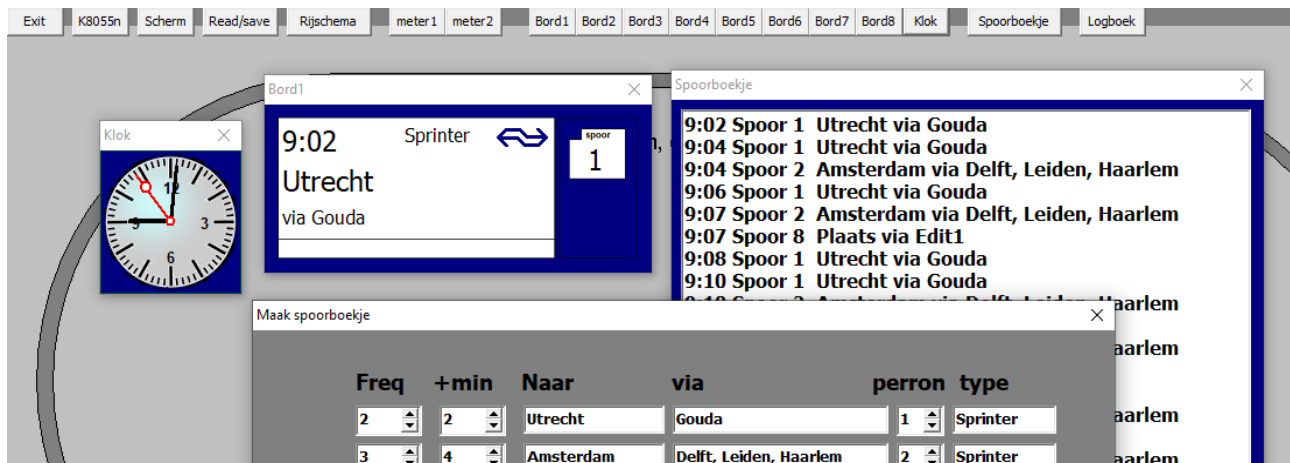
Het programma bestaat uit de volgende stappen:

1. wissel-1 wordt omgezet
2. blok-1 wordt onder spanning gezet zodat de trein dadelijk kan doorrijden naar blok-2
3. sein-2 komt op groen
4. er wordt nog 2 seconde gewacht
5. blok-2 komt op spanning, de trein vertrekt.
6. er wordt 2 seconde gewacht
7. het sein komt op rood
8. blok-2 wordt spannings loos.

Je kan deze rit telkens met de hand starten door op knop te drukken.

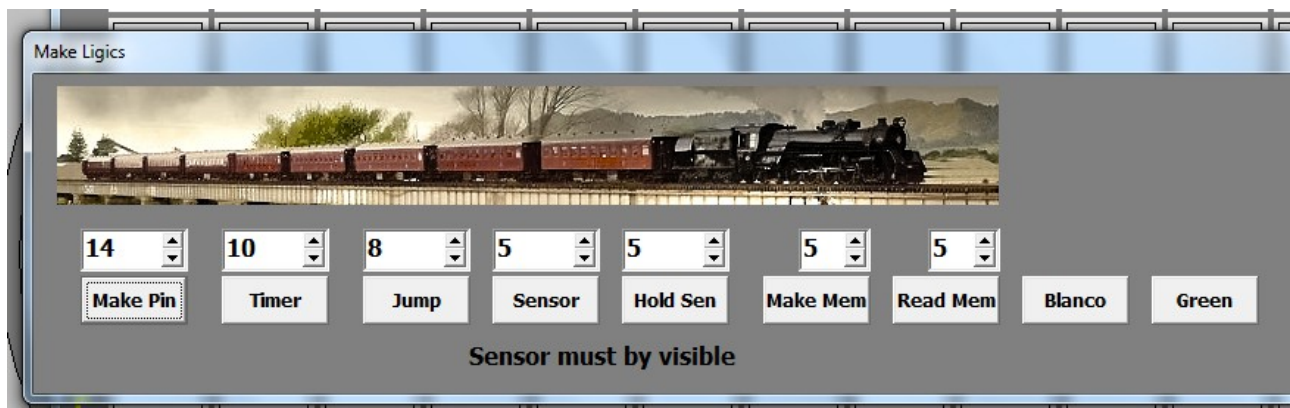


## Rijden volgens het spoorboekje



Wanneer je rijdt volgens het spoorboekje hoef je niet telkens een lijn te starten, dit gebeurt nu volgens het spoorboekje.

Je kunt de volgende blokken invoeren:



### Make Pin

Hiermee geef je een korte stroompje (0,5 sec) naar de geselecteerde pinuitgang.

### Timer

Wacht het aantal seconde om daarna weer verder te gaan. De opgegeven tijd is in seconde.

### Jump

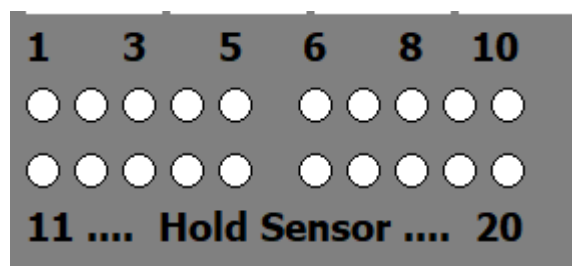
Deze functie geeft de mogelijk om zowel vooruit of achteruit te springen binnen de lijndienst. Met deze functie kun je oneindig durende lussen maken.

### Sensor

Dit blok wacht totdat de geselecteerde sensor hoog is om verder te gaan. De sensor moet wel zichtbaar (Visible) zijn.

### Hold Sen

Hold sen is een sensor die door de trein op de baan hoog gezet wordt en daarna hoog blijft. Pas wanneer het hold-sensor-blok actief wordt, wordt de hold-sen weer laag gezet. Hold-sen kan ook met de hand worden bediend.



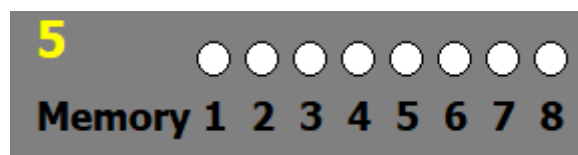
### Make Mem

Er zijn 8 geheugens (Memory) die hoog of laag gezet kunnen worden.

Met Make Mem maak je de memory hoog. Bij een hoge memory gaat de lijndienst daar niet verder door. De memory's werken overal de lijndiensten, hiermee kun je dus een andere lijndienst beïnvloeden.

### Read Mem

Hier wordt de geselecteerde Memory gelezen, als deze hoog staat, gaat de lijndienst niet verder. De memory's kun je ook met de hand bedienen.



### Blanco

Hiermee maak je een blokje weer leeg.

### Green

Deze wordt gebruikt als binnenkomst controle bij het gebruik van het spoorboekje.

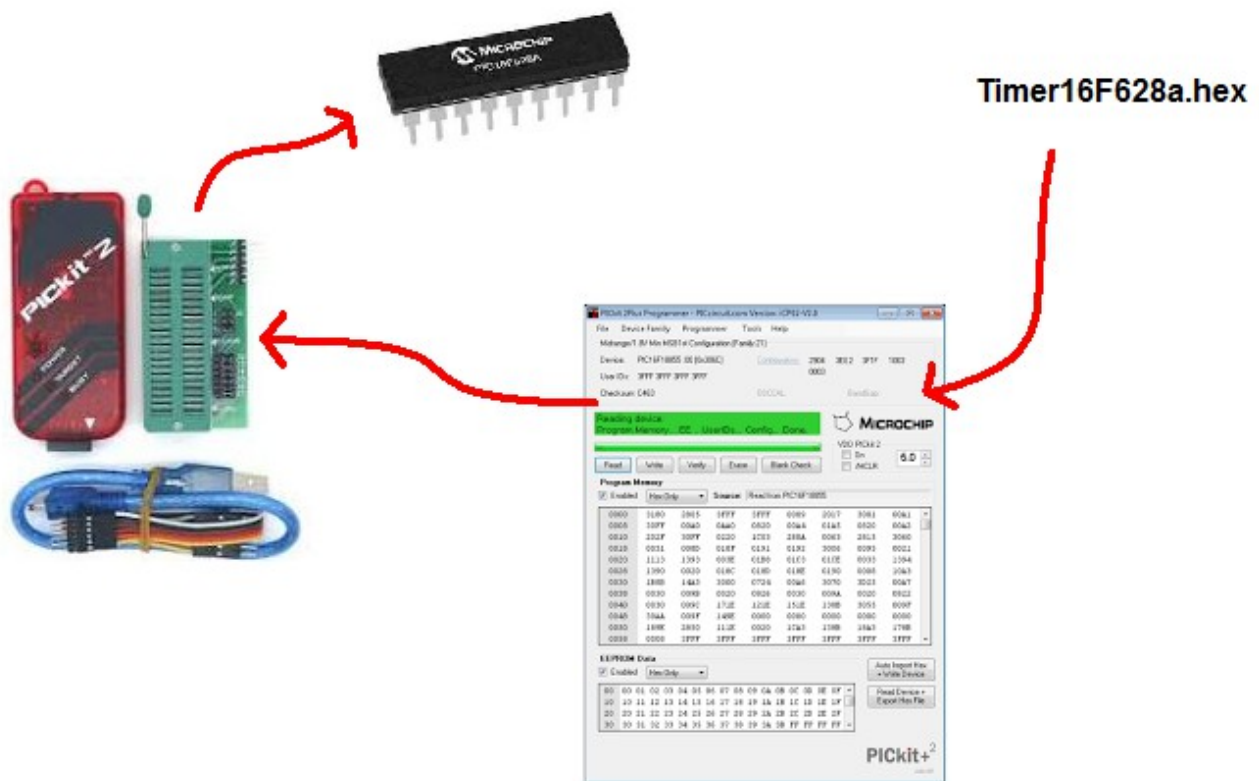
Wanneer de trein vertrekt volgens het spoorboekje wordt de indicator op rood gezet.

Wanneer nu de trein op tijd binnen komt, komt de indicator op groen en kan de volgende rit beginnen volgens het spoorboekje. Als de echter de trein niet op tijd is binnengekomen blijft de indicator rood en wordt deze rit overgeslagen. Op deze manier wordt voorkomen dat er een spooktrein vertrekt.





## Het laden van de 16F628a

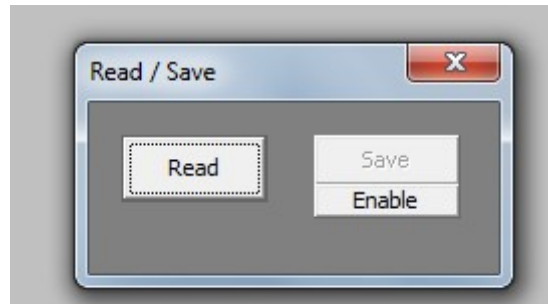


Dit geldt ook het inlezen van programma's voor de PIC 12F675  
 Gebruik voor het laden PICKIT-2 of PICKIT-3.

PIC	Programma	Toepassing
16F628a	Timer16F628a.hex	Multiplexing en vertrager sensoren
12F675	12F675RoGr.hex	Sein rood/groen
12F675	12F675RoYeGr.hex	Sein rood/geel/groen
12F675	12F675Hold.hex	Hold-relais
12F675	12F675DubbelHold	Hold-relais

Deze “.hex” programma's staan in de map VasteData.

## Opslaan en inlezen



Met Read/Save lees en sla je de baanontwerp op zoals je die gemaakt hebt..

Pas de adressen aan voor elke gebruikte multiplexer indien nodig.

Give a Name

Pin	Naam	Adres
Pin-1	Wis1re	0
Pin-2	Wis1om	1
Pin-3	Leeg3	2
Pin-4	Leeg4	3
Pin-5	Leeg5	4
Pin-6	Leeg6	5
Pin-7	Leeg7	6
Pin-8	Leeg8	7
Pin-9	Leeg9	8
Pin-10	Leeg10	9
Pin-11	Leeg11	10
Pin-12	Leeg12	11
Pin-13	Leeg13	12
Pin-14	Leeg14	13
Pin-15	Leeg15	14
Pin-16	Leeg16	15
Pin-17	Leeg17	16

Save Naam

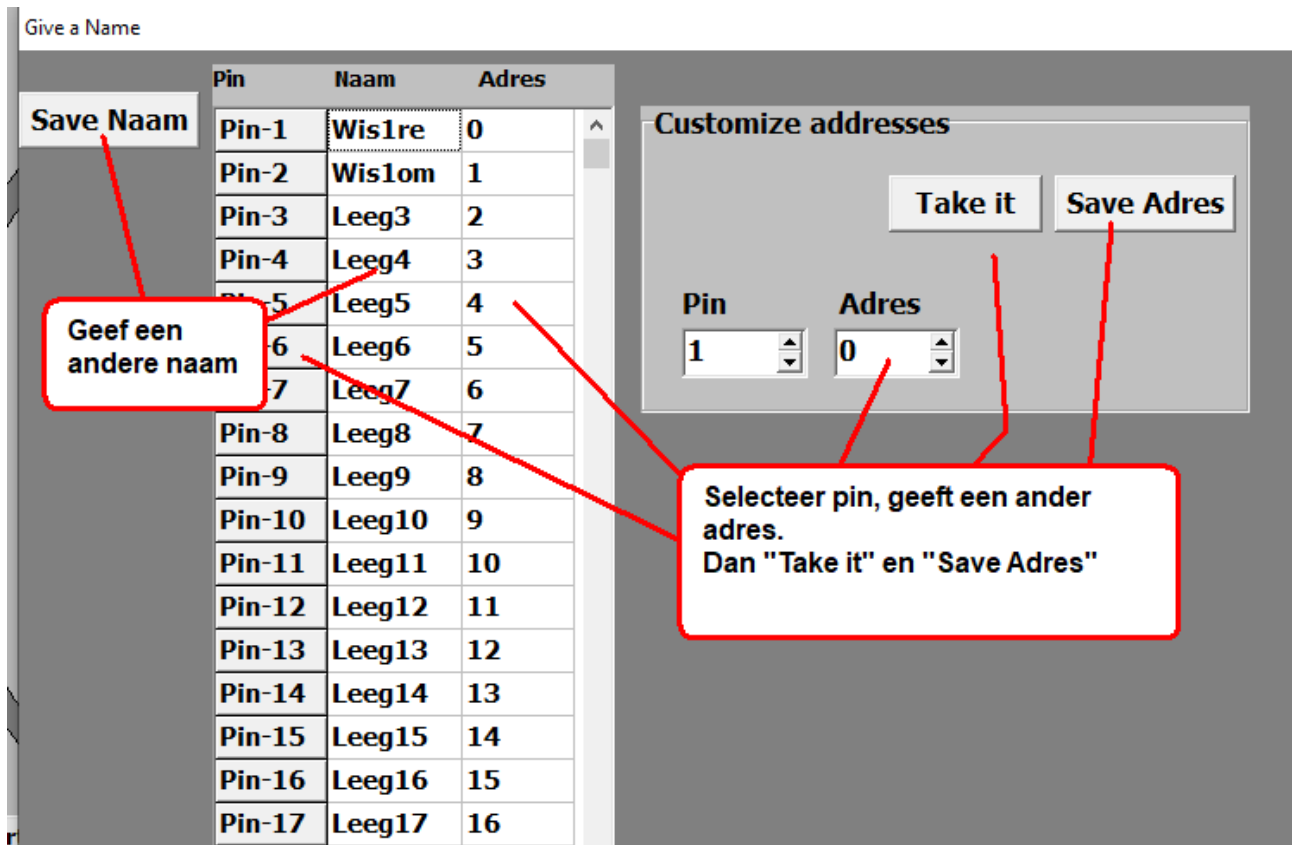
Geef een andere naam

Customize addresses

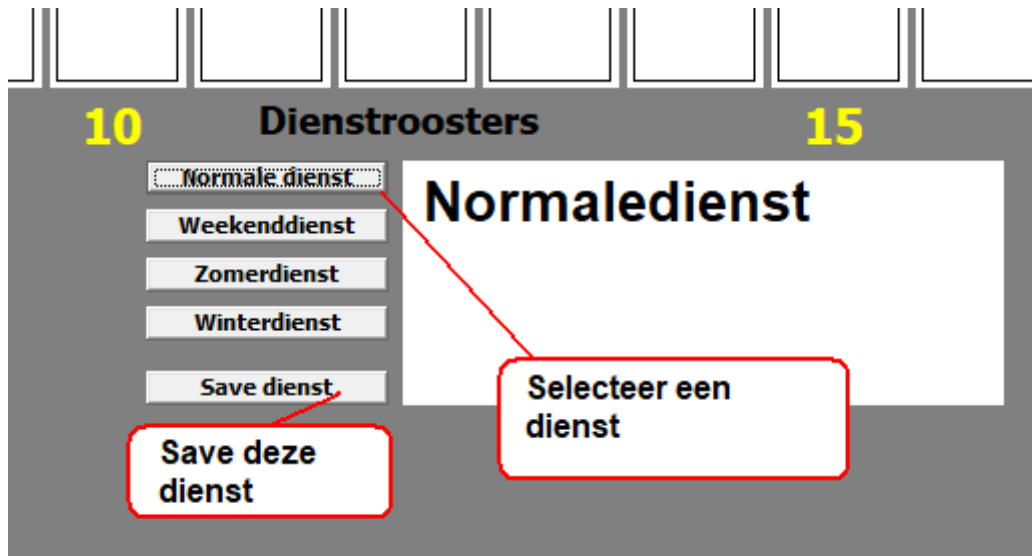
Take it Save Adres

Pin 1 Adres 0

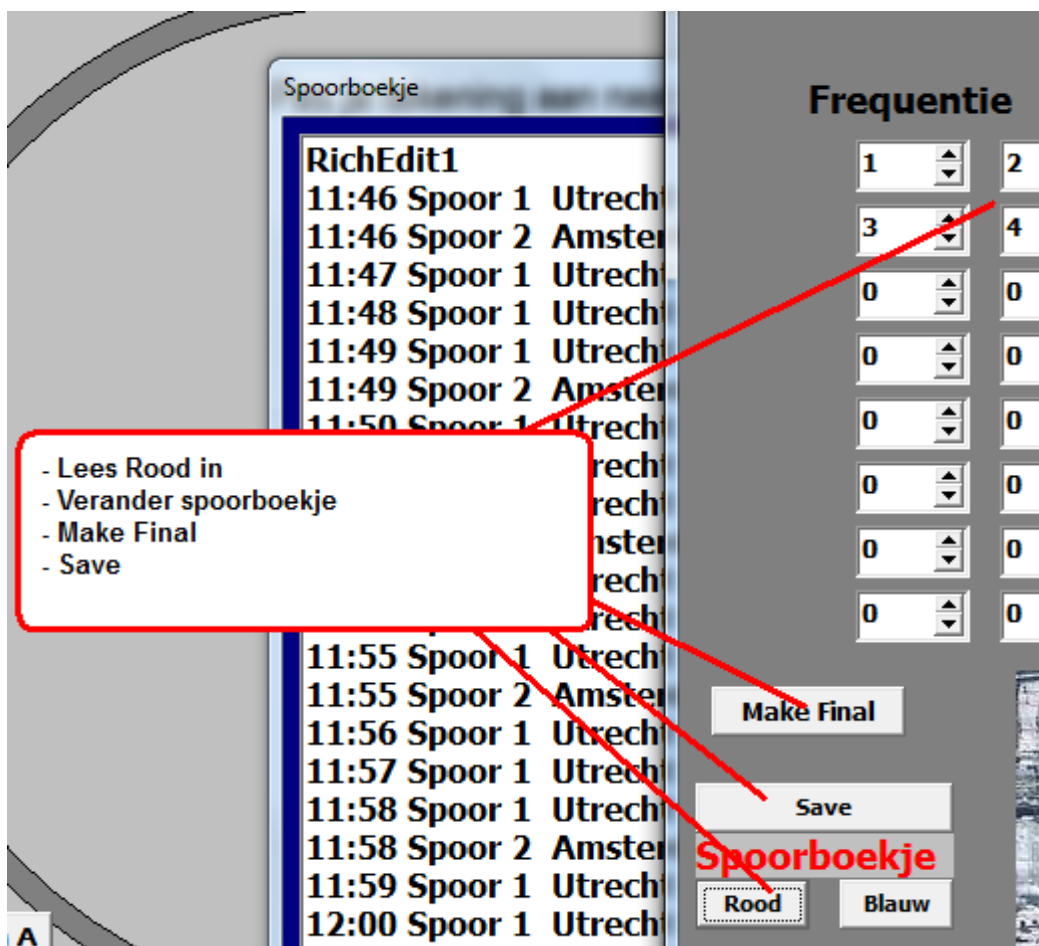
Selecteer pin, geeft een ander adres.  
Dan "Take it" en "Save Adres"



Maak een dienst



Maak het spoorboekje



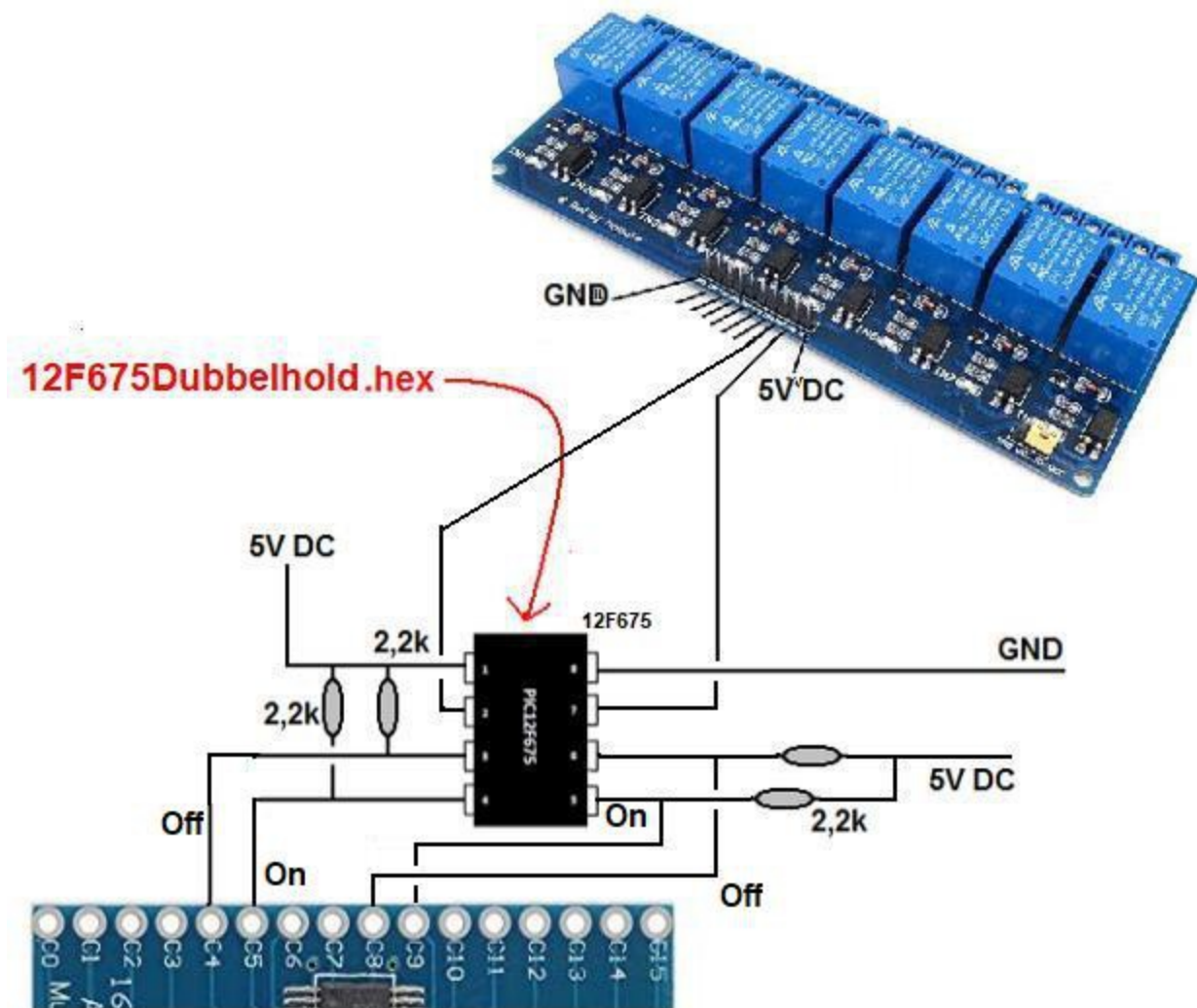
Er kunnen 2 spoorboekjes worden gemaakt, een rood en een blauw spoorboekje.





## De Dubbelhold-relais schakeling

Met de 12F675 pic kun je ook een dubbele hold-schakeling maken.

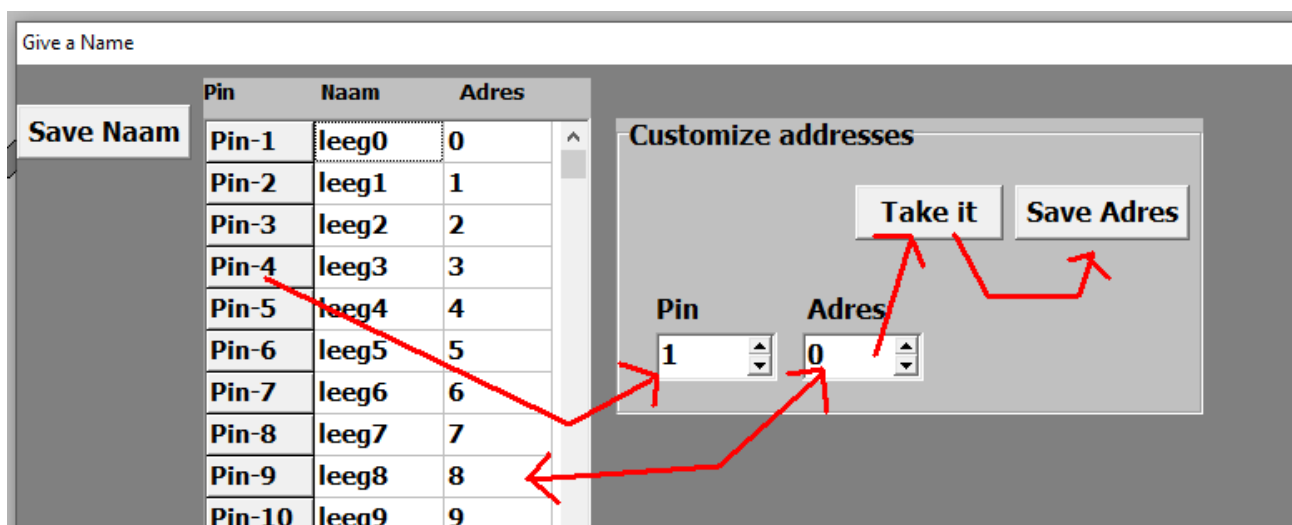


De stroom vanaf het relaisblok loopt naar de 12F675 die hem wel of niet laat afvloeien naar GND. Op pootje 1 en 2 van de 12F675 staat normaal 5 Volt spanning, via de multiplexer wordt deze op commando naar 0 Volt gebracht waardoor de 12F675 schakelt. Een pootje schakelt aan, de andere schakelt af.

## De standaard waarheidstabel

Als je gebruik maak van een 8-relaisblok kan het handig zijn om een waarheidstabel van elke multiplexer te maken om te bepalen welk relais is verbonden met welke knop. Je kan deze adressen aanpassen naar je eigen situatie

Uitgang K8055 naar multiplexer ...				Adres naar multiplexer en bijbehorende functie.	
1 =>S4	2 => S3	3 =>S2	4 =>S1		
0	0	0	0	15	Uit (rood)
0	0	0	1	14	Aan (groen)
0	0	1	0	13	.....
0	0	1	1	12	.....
0	1	0	0	11	
0	1	0	1	10	
0	1	1	0	9	
0	1	1	1	8	
1	0	0	0	7	
1	0	0	1	6	
1	0	1	0	5	
1	0	1	1	4	
1	1	0	0	3	
1	1	0	1	2	
1	1	1	0	1	
1	1	1	1	0	

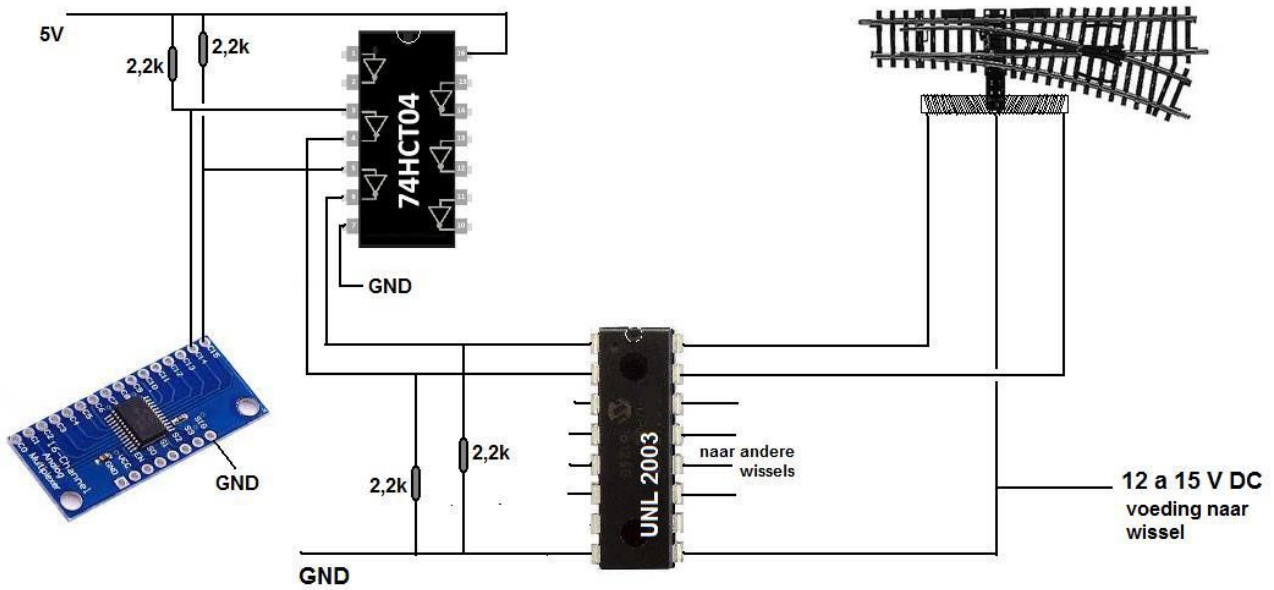


De adressen naar de multiplexers kun je naar wens aanpassen. Multiplexer-1 heeft de adressen van 0 t/m 15, multiplexer-2 heeft de adressen van 16 t/m 31 enz.

### Wissel aansturing met de UNL2003

Deze ic kun je als vervanging van het relais gebruiken.

De maximale stroom doorgang deze ic is 500 mA.



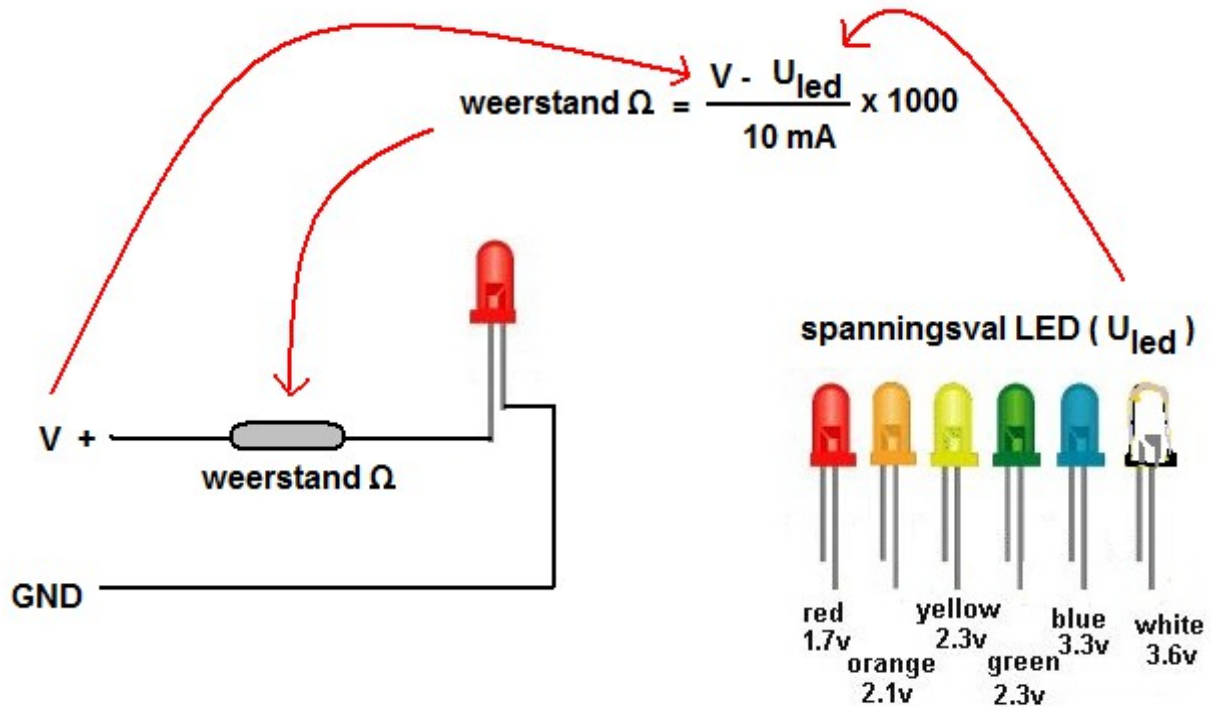
De 74HCT04 draait het signaal om, zodat de werking low triggerd wordt.

Er kunnen met deze twee ic's maximaal 3 wissels worden aangestuurd.

## LED's

Op de baan worden vaak led's gebruik. Je moet bij de montage vooral letten op de stroomdoorgang, 10 mA is meestal voldoende. De stroomdoorgang mag nooit meer zijn dan 20 mA, bij een hogere stroomdoorgang geven ze niet meer licht maar gaan ze wel eerder stuk.

De stroomdoorgang regel je met een voorschakelweerstand, hierbij speelt de aangelegde spanning en de kleur van de led een rol. Iedere kleur led heeft een eigen valspanning, deze is ook een klein beetje afhankelijk van de stroomdoorgang, maar voor gebruik kan deze formule worden gehanteerd.



Als je een rode led op een spanning van 5 Volt wilt zetten krijg je een weerstand van:

$$(5 - 1,7) / 10 \times 1000 = 330 \text{ Ohm}$$

Vaak moet je dan de weerstand kiezen die het meest in de buurt van je berekening komt. Een led heeft een lang en een kort aansluitdraadje, de lange draad sluit je aan op de plus.

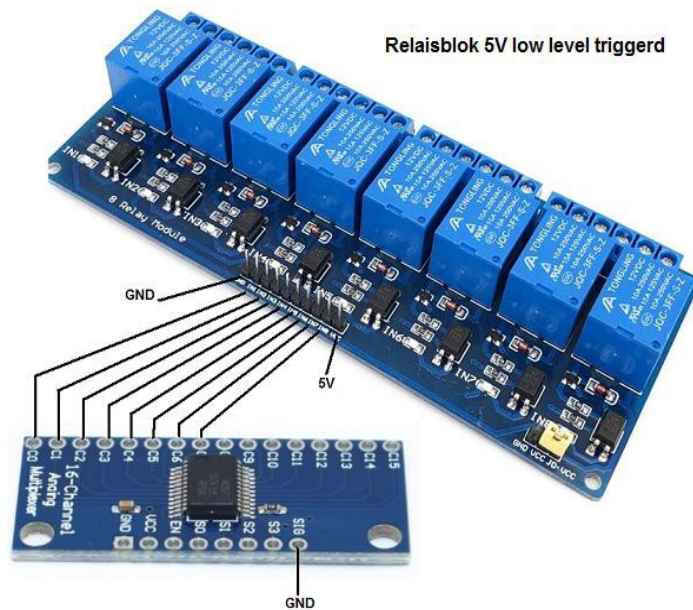
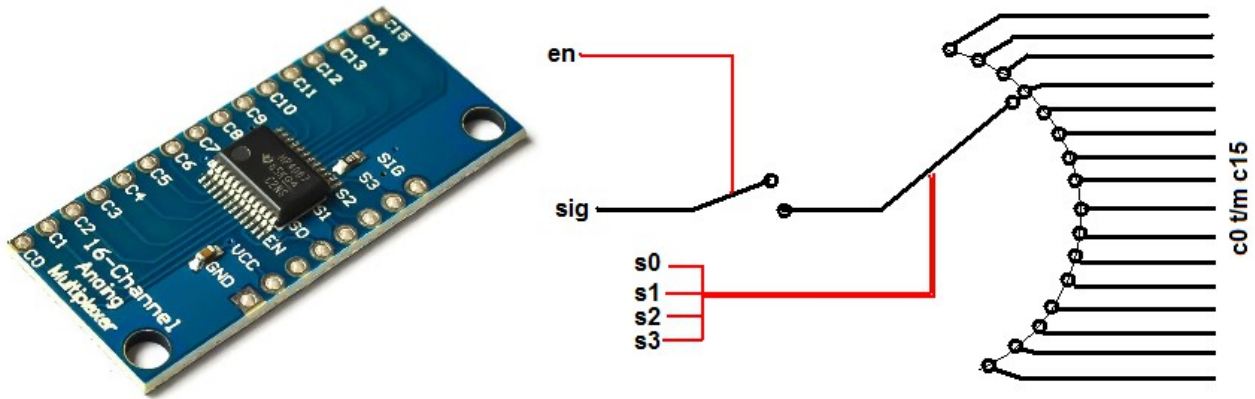


## De werking van de multiplexers

Een multiplexer kun je beschouwen als een ouderwetse draaiknop met 16 standen die je ook nog eens helemaal uit kan zetten. Het programma stuurt 8 multiplexers aan.

Elke multiplexer heeft 16 uitgangen zodat je totaal op  $8 \times 16 = 128$  uitgangen komt.

Maar voor elke wissel, baanvak of sein dat je aanstuurt, heb je 2 signalen nodig, een voor aanzetten en een voor het weer uit- of afzetten. Ook is er een ruststand nodig waardoor het totaal (aan wissels enz.) op 61 komt. De multiplexer werkt op een spanning van 5 Volt op pin-VCC (maximaal 9V).



De relais worden aangesloten op de uitgangen van de multiplexers, je kan daarvoor kant en klare relaisblokken gebruiken die voor arduino's zijn ontworpen.

Let er op dat deze low triggerd is. Als de uitgangen zijn gemaakt, even testen waar welke uitgang precies zit.

## Lijst van voornaamste benodigdheden

	Functie	Indicatie prijs (Euro)	Opmerkingen
1 x K8055n Velleman	USB I/O-kaart	40,-	Bouwpakket
8 x Multiplexer 74HC4067	Multiplexer-kaart	6,-	AliExpress
8 x Relaisblok lowtriggerd 5V	Relais	32,-	AliExpress
Latching relay HFD2/005-S-L-D	Houd-relais	10,-	Set 10 stuks
Weerstanden 2,2 k	Weerstanden	1,-	setje
Condensator 1000uF	Stabiele voeding	2,-	setje
Voeding 12 V dc	Stabiele voeding	8,-	voedingsstekker
LM7805	Spanningsregelaar 5V	1,-	setje
22 pF condensator	voor IC	2,-	setje
PIC-kit 2 of 3	programmeerset	10,-	AliExpress
CD4051be	multiplexing	1,-	setje
PIC 12F675	Geprogrammeerde IC	10,-	10 stuks
PIC 16F628a	Geprogrammeerde IC	10,-	10 stuks
Kristal 20 Mhz	IC	1,-	setje
PNP transistor BC557b	Transistor	1,-	setje
Diodes	Blusdiode	1,-	setje
Verbindingsdraadjes	Female-female	2,-	bundel
IC-voetjes	Behuizing IC	2,-	setje
PCB stripboard	Printplaatjes	6,-	
Soldeergereedschap			
Laptop met Windows XP of hoger			
	Totaal circa	150,-	

### Back-up

Het is verstandig om van je gehele map een back-up te maken, mocht er iets zoek raken dan kun je altijd een beroep doen op je back-up.

## De printplaten

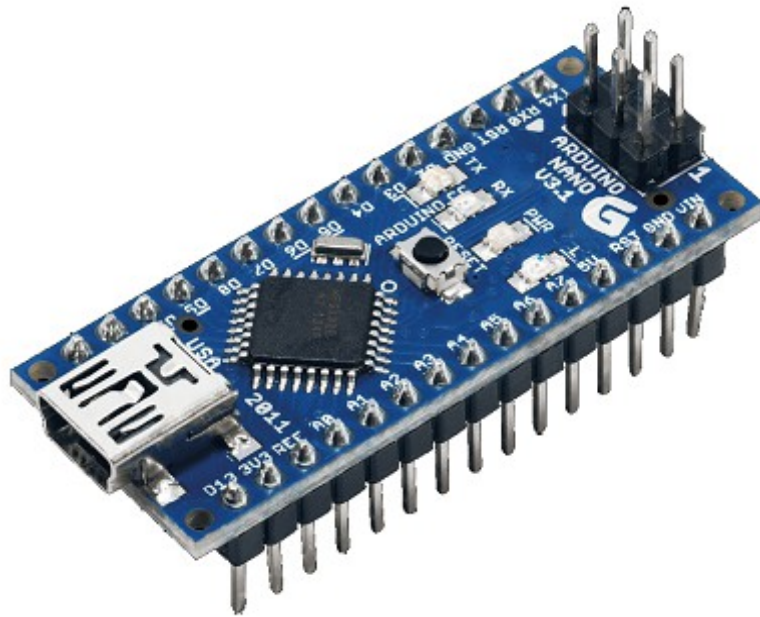
Het hele project bestaat uit het maken van de printplaten en deze met elkaar te verbinden.

De K8055n (van Velleman) is te koop als bouw pakket maar ook als een gesoldeerde versie (de VM110).

Als bouw pakket is hij wel een stuk voordeliger en hij makkelijk te solderen.

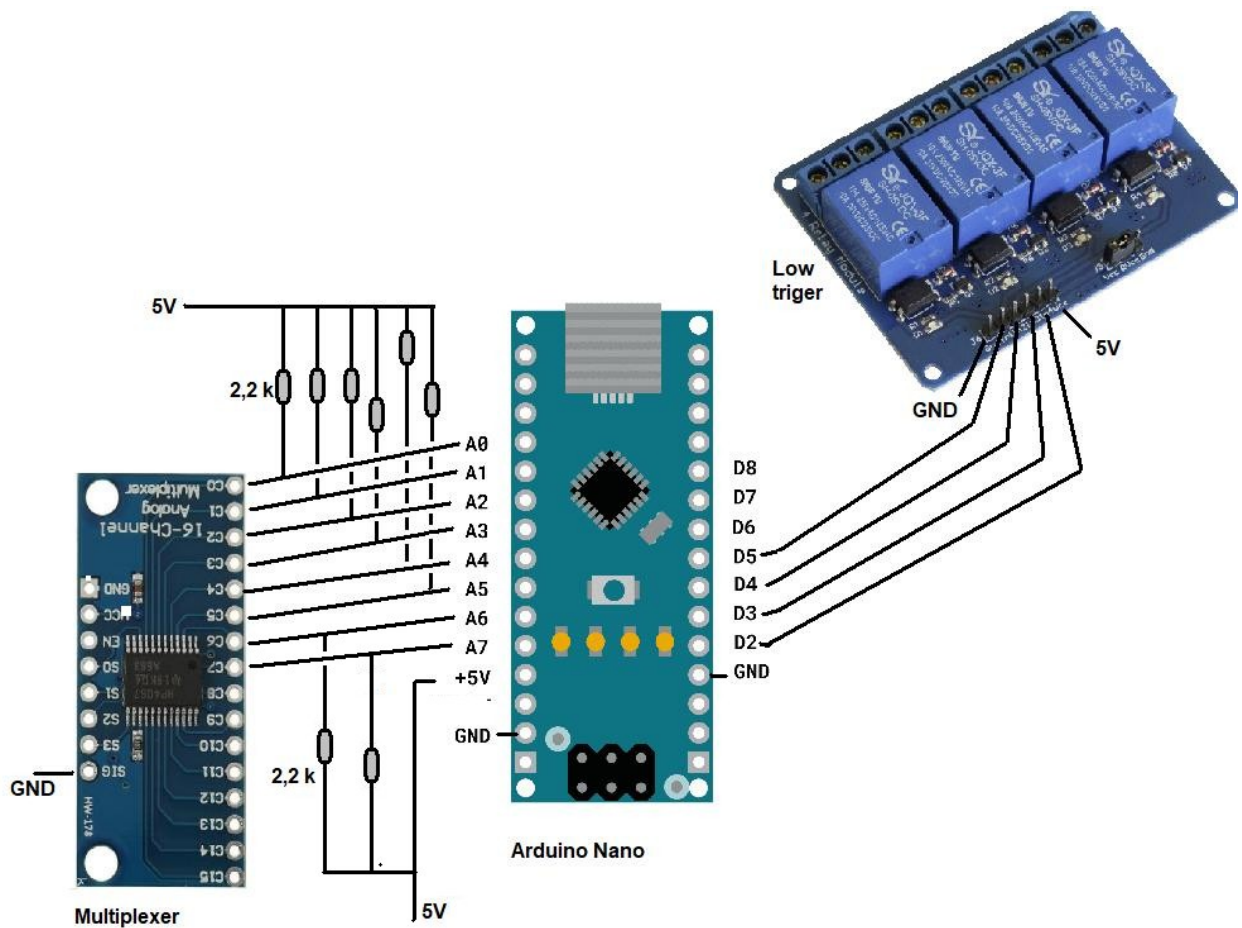


## Arduino Nano



De schakelingen zijn ook te maken met een Arduino's in plaats van met PICs. Het voordeel van een Arduino is, dat je deze direct kunt aansluiten tussen de multiplexer en het relaisblok, je hoeft nu niet te solderen.

### De holdschakelling met een Arduino



De Arduino code hiervoor is:

```
int Uit2 = 2;
int Ain0 = A0;
int Ain1 = A1;

int Uit3 = 3;
int Ain2 = A2;
int Ain3 = A3;

int Uit4 = 4;
int Ain4 = A4;
int Ain5 = A5;

int Uit5 = 5;
int Ain6 = A6;
int Ain7 = A7;

void setup()
{
  pinMode(Uit2, OUTPUT);
  pinMode(Uit3, OUTPUT);
  pinMode(Uit4, OUTPUT);
  pinMode(Uit5, OUTPUT);
}

void loop()
{
  if (analogRead(Ain0)<200) {digitalWrite (Uit2, LOW);}
  if (analogRead(Ain1)<200) {digitalWrite (Uit2,HIGH);}

  if (analogRead(Ain2)<200) {digitalWrite (Uit3, LOW);}
  if (analogRead(Ain3)<200) {digitalWrite (Uit3,HIGH);}

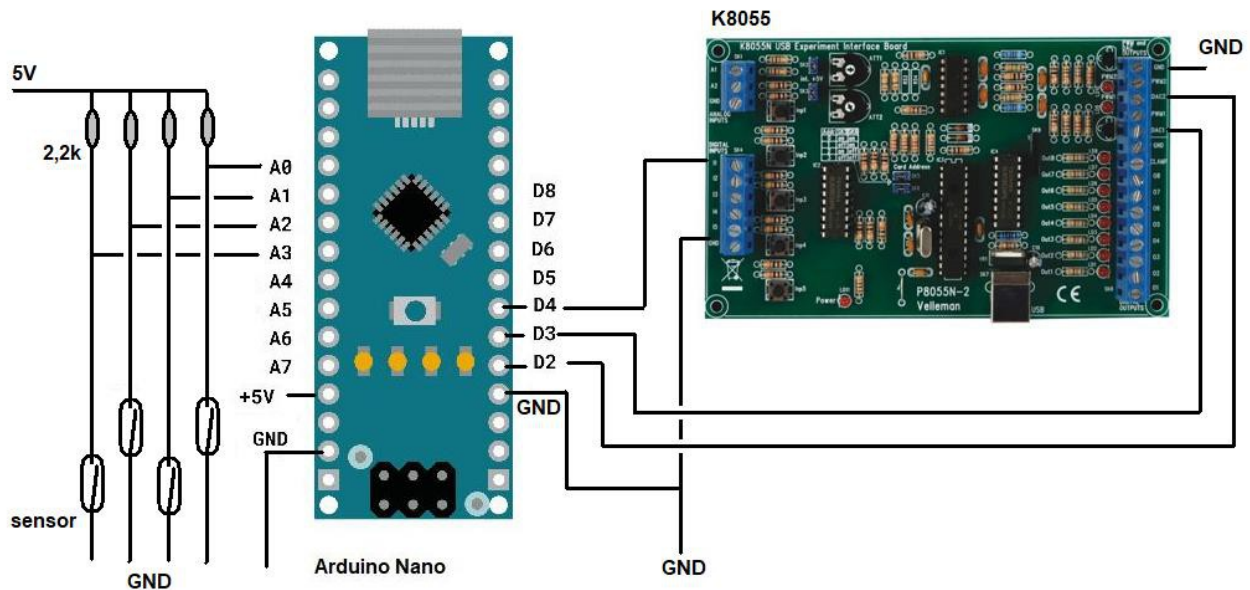
  if (analogRead(Ain4)<200) {digitalWrite (Uit4, LOW);}
  if (analogRead(Ain5)<200) {digitalWrite (Uit4,HIGH);}

  if (analogRead(Ain6)<200) {digitalWrite (Uit5, LOW);}
  if (analogRead(Ain7)<200) {digitalWrite (Uit5,HIGH);}

  delay(100);
}
```



## De sensoringen met de Arduino



Met de schakeling hier kunnen 4 sensors worden gelezen op 1 ingang van de de K8055.

Om de 20 sensors in te kunnen lezen heb je dus 5 Arduino's nodig.

(Het is ook mogelijk om 2 x 4 sensoren in te lezen op een Arduino.)

Er staat continu 5 Volt spanning op de ingang van de Arduino, als de sensor iets meet trekt hij dit signaal van 5 V omlaag naar 0 V. Als sensor zou je een readcontact kunnen gebruiken of een schakeling maken met een transistor die de spanning omlaag trekt.

De Arduino code :

```
// InlezenSensors
int Sen1 = A0;           // de analoge inputs wordt als input gebruikt welke de sensor afgeeft
int Sen2 = A1;
int Sen3 = A2;
int Sen4 = A3;

int Ain1 = 2;
int Ain2 = 3;
int Uit1 = 4;

int Tijd;
int T1;
int T2;
int T3;
int T4;

void setup() {
  pinMode(Ain1, INPUT);
  pinMode(Ain2, INPUT);
  pinMode(Uit1, OUTPUT);
  Tijd=2000;           // afteltijd komt overeen met circa 1 seconde
```

```

}

void loop() {
if (Sen1<200) {T1=Tijd;} // als signaal kleiner is dan 1 volt, dan sfteltijd op 2000 zetten
if (Sen2<200) {T2=Tijd;}
if (Sen3<200) {T3=Tijd;}
if (Sen4<200) {T4=Tijd;}

if (T1>0) {T1=T1-1;} // aftellen
if (T2>0) {T2=T2-1;}
if (T3>0) {T3=T3-1;}
if (T4>0) {T4=T4-1;}

if (Ain1==LOW && Ain2==LOW && T1>0) {Uit1=LOW; }
if (Ain1==LOW && Ain2==LOW && T1==0) {Uit1=HIGH;}

if (Ain1==LOW && Ain2==HIGH && T2>0) {Uit1=LOW; }
if (Ain1==LOW && Ain2==HIGH && T2==0) {Uit1=HIGH;}

if (Ain1==HIGH && Ain2==HIGH && T3>0) {Uit1=LOW; }
if (Ain1==HIGH && Ain2==HIGH && T3==0) {Uit1=HIGH;}

if (Ain1==HIGH && Ain2==LOW && T4>0) {Uit1=LOW; }
if (Ain1==HIGH && Ain2==LOW && T4==0) {Uit1=HIGH;}
}int Sen1 = A0; // de analoge inputs wordt als input gebruikt welke de sensor afgeeft
int Sen2 = A1;
int Sen3 = A2;
int Sen4 = A3;

int Ain1 = 2;
int Ain2 = 3;
int Uit1 = 4;

int Tijd;
int T1;
int T2;
int T3;
int T4;

void setup() {
pinMode(Ain1, INPUT);
pinMode(Ain2, INPUT);
pinMode(Uit1, OUTPUT);
Tijd=2000; // afteltijd komt overeen met circa 1 seconde
}

void loop() {
if (Sen1<200) {T1=Tijd;} // als signaal kleiner is dan 1 volt, dan sfteltijd op 2000 zetten
if (Sen2<200) {T2=Tijd;}
if (Sen3<200) {T3=Tijd;}
if (Sen4<200) {T4=Tijd;}

```

```
if (T1>0) {T1=T1-1;}           // aftellen
if (T2>0) {T2=T2-1;}
if (T3>0) {T3=T3-1;}
if (T4>0) {T4=T4-1;}

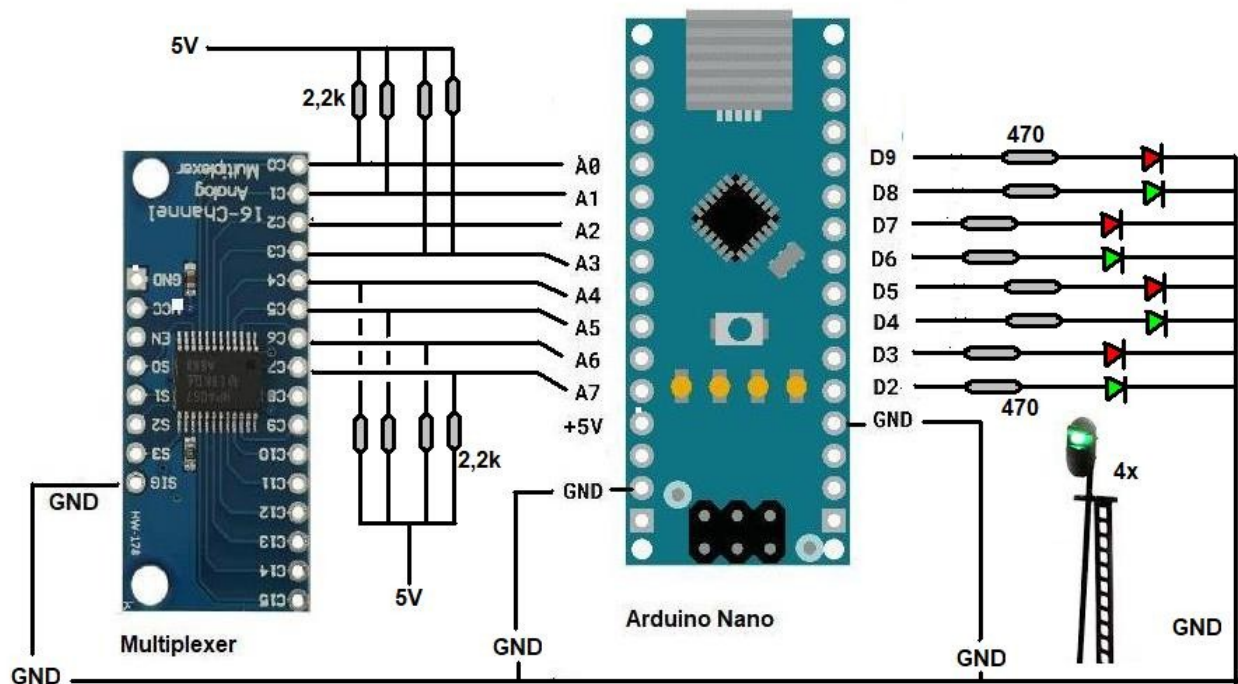
if (Ain1==LOW && Ain2==LOW && T1>0) {Uit1=LOW; }
if (Ain1==LOW && Ain2==LOW && T1==0) {Uit1=HIGH;}

if (Ain1==LOW && Ain2==HIGH && T2>0) {Uit1=LOW; }
if (Ain1==LOW && Ain2==HIGH && T2==0) {Uit1=HIGH;}

if (Ain1==HIGH && Ain2==HIGH && T3>0) {Uit1=LOW; }
if (Ain1==HIGH && Ain2==HIGH && T3==0) {Uit1=HIGH;}

if (Ain1==HIGH && Ain2==LOW && T4>0) {Uit1=LOW; }
if (Ain1==HIGH && Ain2==LOW && T4==0) {Uit1=HIGH;}
}
```

## Het RoodGroenSein



.Met de multiplexer en een Arduino kun je 4 seinen aansturen.  
 (De multiplexer heeft 16 kanalen, dus je kunt hierop 2 Arduino's op aansluiten)  
 In deze opstelling geven de seinen of rood of groen aan.  
 Zo zet A7 het sein op groen (D2) en A6 het sein weer op rood (D3).

De Arduino code is hiervoor:

```
int Ain0 = A0; int Ain1 = A1; int Ain2 = A2; int Ain3 = A3;
int Ain4 = A4; int Ain5 = A5; int Ain6 = A6; int Ain7 = A7;
```

```
int Groen1 = 2; int Rood1 = 3;
int Groen2 = 4; int Rood2 = 5;
int Groen3 = 6; int Rood3 = 7;
int Groen4 = 8; int Rood4 = 9;
```

```
void setup()
{
  pinMode(Groen1, OUTPUT);
  pinMode(Rood1, OUTPUT);
  pinMode(Groen2, OUTPUT);
  pinMode(Rood2, OUTPUT);
  pinMode(Groen3, OUTPUT);
  pinMode(Rood3, OUTPUT);
  pinMode(Groen4, OUTPUT);
  pinMode(Rood4, OUTPUT);
}
```

```
void loop()
```

```
{
if (analogRead(Ain7)<200) {digitalWrite (Groen1,HIGH); digitalWrite (Rood1,LOW);}
if (analogRead(Ain6)<200) {digitalWrite (Rood1,HIGH); digitalWrite (Groen1,LOW);}

if (analogRead(Ain5)<200) {digitalWrite (Groen2,HIGH); digitalWrite (Rood2,LOW);}
if (analogRead(Ain4)<200) {digitalWrite (Rood2,HIGH); digitalWrite (Groen2,LOW);}

if (analogRead(Ain3)<200) {digitalWrite (Groen3,HIGH); digitalWrite (Rood3,LOW);}
if (analogRead(Ain2)<200) {digitalWrite (Rood3,HIGH); digitalWrite (Groen3,LOW);}

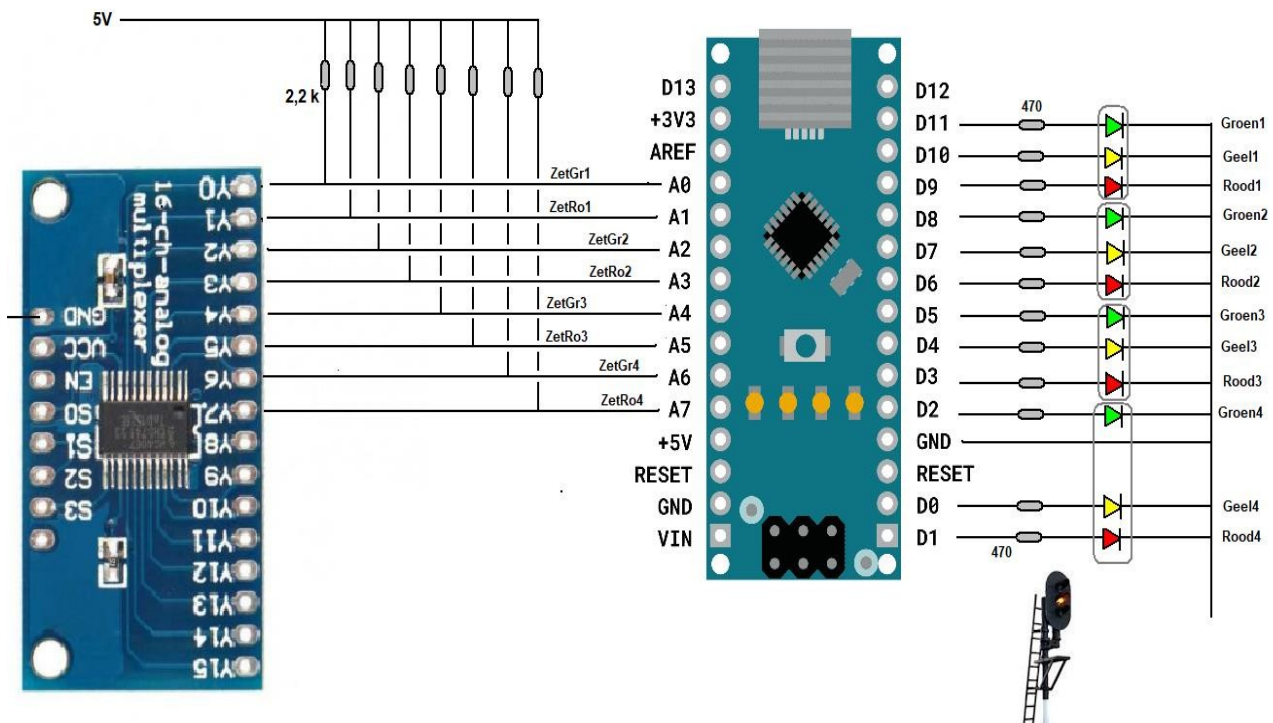
if (analogRead(Ain1)<200) {digitalWrite (Groen4,HIGH); digitalWrite (Rood4,LOW);}
if (analogRead(Ain0)<200) {digitalWrite (Rood4,HIGH); digitalWrite (Groen4,LOW);}

delay(100);
}
```



## Het RoodGeelGroenSein

Met dit sein gaat het sein niet direct op groen maar via eerst naar geel, alvorens naar groen te gaan.



Op een Arduino kunnen 4 van deze seinen worden aangesloten.

De Arduino code hiervoor is:

```
int Geel4 = 0; int Rood4 = 1; int Groen4 = 2; int Rood3 = 3; int Geel3 = 4; int Groen3 = 5;
int Rood2 = 6; int Geel2 = 7; int Groen2 = 8; int Rood1 = 9; int Geel1 = 10; int Groen1 = 11;
```

```
int ZetGr1 = A0; int ZetRo1 = A1; int ZetGr2 = A2; int ZetRo2 = A3;
int ZetGr3 = A4; int ZetRo3 = A5; int ZetGr4 = A6; int ZetRo4 = A7;
```

```
int Teller1 = 0; int X1 = 0;
int Teller2 = 0; int X2 = 0;
int Teller3 = 0; int X3 = 0;
int Teller4 = 0; int X4 = 0;
```

```
void setup() {
pinMode(Groen1,OUTPUT); pinMode(Geel1,OUTPUT); pinMode(Rood1,OUTPUT);
pinMode(Groen2,OUTPUT); pinMode(Geel2,OUTPUT); pinMode(Rood2,OUTPUT);
pinMode(Groen3,OUTPUT); pinMode(Geel3,OUTPUT); pinMode(Rood3,OUTPUT);
pinMode(Groen4,OUTPUT); pinMode(Geel4,OUTPUT); pinMode(Rood4,OUTPUT);
}
```

```
void loop()
{
if (analogRead(ZetRo1)<200) {digitalWrite(Rood1,HIGH); digitalWrite(Geel1,LOW);
digitalWrite(Groen1,LOW); Teller1=3000; X1=0;}
if (analogRead(ZetRo2)<200) {digitalWrite(Rood2,HIGH); digitalWrite(Geel2,LOW);
digitalWrite(Groen2,LOW); Teller2=3000; X2=0;}
if (analogRead(ZetRo3)<200) {digitalWrite(Rood3,HIGH); digitalWrite(Geel3,LOW);
```

```

digitalWrite(Groen3,LOW); Teller3=3000; X3=0;}
if (analogRead(ZetRo4)<200) {digitalWrite(Rood4,HIGH); digitalWrite(Geel4,LOW);
digitalWrite(Groen4,LOW); Teller4=3000; X4=0;}

if (analogRead(ZetGr1)<200) {X1=1;}
if (analogRead(ZetGr2)<200) {X2=1;}
if (analogRead(ZetGr3)<200) {X3=1;}
if (analogRead(ZetGr4)<200) {X4=1;}

if (Teller1>0) {Teller1=Teller1-X1;}
if (Teller2>0) {Teller2=Teller2-X2;}
if (Teller3>0) {Teller3=Teller3-X3;}
if (Teller4>0) {Teller4=Teller4-X4;}

if (Teller1==0) {X1=0;}
if (Teller2==0) {X2=0;}
if (Teller3==0) {X3=0;}
if (Teller4==0) {X4=0;}

if (Teller1>0 && Teller1<3000) {digitalWrite(Rood1,LOW); digitalWrite(Geel1,HIGH);
digitalWrite(Groen1,LOW);}
if (Teller2>0 && Teller2<3000) {digitalWrite(Rood2,LOW); digitalWrite(Geel2,HIGH);
digitalWrite(Groen2,LOW);}
if (Teller3>0 && Teller3<3000) {digitalWrite(Rood3,LOW); digitalWrite(Geel3,HIGH);
digitalWrite(Groen3,LOW);}
if (Teller4>0 && Teller4<3000) {digitalWrite(Rood4,LOW); digitalWrite(Geel4,HIGH);
digitalWrite(Groen4,LOW);}

if (Teller1==0) {digitalWrite(Rood1,LOW); digitalWrite(Geel1,LOW);
digitalWrite(Groen1,HIGH);}
if (Teller2==0) {digitalWrite(Rood2,LOW); digitalWrite(Geel2,LOW);
digitalWrite(Groen2,HIGH);}
if (Teller3==0) {digitalWrite(Rood3,LOW); digitalWrite(Geel3,LOW);
digitalWrite(Groen3,HIGH);}
if (Teller4==0) {digitalWrite(Rood4,LOW); digitalWrite(Geel4,LOW);
digitalWrite(Groen4,HIGH);}

delay(5);
}

```

T.L. Hengelmolen