

Inhoudstabel

1	VOORWOORD	1
2	BESPREKING VAN DE COMPUTERPOORTEN	2
2.1	INLEIDING.....	2
2.2	SERIËLE POORT	2
2.3	PARALLELE POORT	5
2.3.1	<i>Het verschil tussen een seriële poort en een parallelle poort.....</i>	<i>5</i>
3	BESPREKING VAN DE SCHUIFREGISTERS	7
3.1	D-FLIPFLOP	7
3.2	ALGEMENE WERKING VAN EEN SCHUIFREGISTER.....	10
3.2.1	<i>Serieel naar parallel.....</i>	<i>10</i>
3.2.2	<i>Serieel naar parallel toegepast (HEF4094B).....</i>	<i>11</i>
3.2.3	<i>Parallel naar serieel.....</i>	<i>13</i>
3.2.4	<i>Parallel naar serieel toegepast (HEF4021B).....</i>	<i>14</i>
4	OVERZICHT VAN DE HARDWARE.....	16
5	OPBOUW ZENDMODULE.....	17
5.1	BLOKSCHEMA.....	17
5.2	ELEKTRONISCH SCHEMA.....	19
5.3	DATADISTRIBUTIE	20
5.3.1	<i>Doel:.....</i>	<i>20</i>
5.3.2	<i>Schema:.....</i>	<i>20</i>
5.3.3	<i>Opbouw 1ste deel:.....</i>	<i>21</i>
5.3.4	<i>Programma:.....</i>	<i>21</i>
5.3.5	<i>Omschrijving code met schuifregister:.....</i>	<i>22</i>
5.3.6	<i>Opbouw 2de deel:.....</i>	<i>22</i>
5.3.7	<i>Code in samen werking met multiplexer/decoder:.....</i>	<i>22</i>
5.3.8	<i>Opbouw 3de deel:.....</i>	<i>23</i>
5.3.9	<i>Gebruik van datadistributie:.....</i>	<i>24</i>
5.4	VORMING VAN DATA EN ADRES	25
5.5	UITBREIDING: DISPLAY	27
5.5.1	<i>Schema.....</i>	<i>28</i>
5.5.2	<i>Programma</i>	<i>29</i>
5.5.3	<i>Display Instructies.....</i>	<i>29</i>
5.6	HOLTEK ENCODER.....	34
5.6.1	<i>Doel.....</i>	<i>34</i>
5.6.2	<i>Schemavoorstelling</i>	<i>34</i>
5.6.3	<i>Signaaluitgang van de Holtek Encoder.....</i>	<i>34</i>
5.6.4	<i>Oscillator.....</i>	<i>35</i>
5.6.5	<i>Gemeten signaal:.....</i>	<i>36</i>
5.7	DE JOYSTICK	37
5.7.1	<i>Klem aansluitingen.....</i>	<i>37</i>
5.7.2	<i>Knop overzicht.....</i>	<i>38</i>
5.7.3	<i>Wat is een OPAMP.....</i>	<i>38</i>
5.7.4	<i>Bepalen van de joystickweerstand.....</i>	<i>39</i>
5.7.5	<i>Bepalen van de grenswaarden.....</i>	<i>40</i>
5.7.6	<i>Gegevens serieel maken.....</i>	<i>42</i>
6	OPBOUW RACEWAGEN	43
6.1	BLOKSCHEMA.....	43
6.2	ELEKTRONISCH SCHEMA.....	44
6.3	HOLTEK DECODER.....	45
6.3.1	<i>Doel.....</i>	<i>45</i>
6.3.2	<i>Schema van bewerkingen.....</i>	<i>46</i>
6.3.3	<i>Het gemeten signaal.....</i>	<i>47</i>

6.3.4	<i>Oscillator</i>	48
6.4	AANSTUREN MOTOREN MET L298N.....	49
6.4.1	<i>Werking L298N motor driver</i>	49
6.4.2	<i>Toestand 1</i>	51
6.4.3	<i>Toestand 2</i>	52
6.4.4	<i>Toestand 3</i>	53
6.4.5	<i>Toestand 4</i>	54
7	WERKING VAN DE SOFTWARE	55
7.1	WAAR IS SOFTWARE VOOR NODIG?.....	55
7.2	DE PROGRAMMEERTAAL VISUAL BASIC.....	56
7.3	BESPREKING VAN HET BESTAND PORT .DLL.....	57
7.4	SAMENSTELLING VAN HET VOLLEDIGE PROGRAMMA.....	58
7.4.1	<i>CGRPrintplaat.dll</i>	58
7.4.2	<i>frmMain</i>	63
7.4.3	<i>frmJoy</i>	75
7.4.4	<i>Module</i>	76
8	BRONNEN	77
8.1	WEBSITES.....	77
8.2	BOEKEN.....	78
9	BIJLAGE 1: MAKEN VAN PRINTPLATEN	79
9.1	WAT IS EEN PRINTPLAAT.....	79
9.2	HOE WORDT EEN PRINTPLAAT GEMAAKT.....	79
9.3	PRINTPLATEN ONTWERPEN.....	81
9.3.1	<i>Ontwerp procedure</i>	81
9.3.2	<i>Het resultaat</i>	82
9.4	HET ETSEN.....	83
9.5	SPOELEN.....	84
10	BIJLAGE 2: DE TRANSSTOR	85
10.1	WERKING.....	85
10.2	DE TRANSISTOR ALS SCHAKELAAR.....	86
11	BIJLAGE 3: ILLUSTRATIES	88
12	DANKWOORD	91
13	LOGBOEK	92

1 Voorwoord

Naar aanleiding van de geïntegreerde proef hebben we besloten om een computer gestuurde racewagens te ontwerpen. We kwamen op dit idee vanuit onze gemeenschappelijke interesse in computers en vanuit onze gedrevenheid om onze kennis van programmeren uit te breiden. Aangezien deze zaken onontbeerlijk zijn bij het sturen van een elektronische toepassing, was onze keuze snel gemaakt.

We zijn van start gegaan met het aanleren van enkele toepasselijke basiscodes in Visual Basic voor het aansturen van poorten. Door het maken van een proefopstelling controleerden we of deze codes correct toekwamen op de seriële poort.

Vervolgens begonnen we met de geleidelijke opbouw van de hardware. Omdat we de instructies via radiogolven naar de racewagens willen verzenden, bestaat deze schakeling uit twee grote delen: een zender en een ontvanger. De zender wordt gestuurd met de computer via de seriële poort. Via deze zender kunnen we 256 verschillende ontvangers aansturen, maar voor ons eindwerk houden we het bij één. De printplaat voor de zender en de ontvanger werd ontworpen in het professionele programma Protel 99. Het leren werken met dit tekenpakket was dan ook een deel van ons eindwerk.

Daarna werd het stuurprogramma herschreven. Dit programma bestond enkel uit zuivere programmacode, en was niet echt gebruiksvriendelijk. Om dit probleem op te lossen werd een gebruikersprogramma gebouwd. Zo kan de gebruiker een parcours vormen. Door een simpele muisklik kan de racewagen dan bediend worden. Het gebruikersprogramma stuurt dan instructies door naar het stuurprogramma.

Voor de wat meer ervaren piloten werd er ook een joysticksturing voorzien.

2 Bespreking van de computerpoorten

2.1 Inleiding

Elk commando, teken ... die door de computer worden gebruikt bestaan uit een byte of 8 bits. Deze bits kunnen nu serieel of parallel verzonden worden.

Bij een seriële poort worden de bits één voor één door de datakabel verzonden. Aan de andere kant bevindt zich een toestel die deze bits terug kan omzetten tot een byte. In ons geval gebeurt dit met een shiftregister of schuifregister (bespreking volgt). Om de communicatie soepel te laten verlopen worden nog controlebits en andere informatie aan de eigenlijke data toegevoegd, met gevolg dat de seriële overdracht geen hoge overdracht snelheid heeft.

Bij de parallelle overdracht worden de gegevens (8 bits) tegelijkertijd doorgezonden. Dit komt overeen met een achttvoudige overdrachtssnelheid t.o.v. de seriële poort.

2.2 Seriële poort

De seriële poort is gemakkelijk te herkennen aan de 9-polige contactbus, of in mindere mate 25-polige contactbus aan de achterzijde van de computer. Seriële poorten worden ook communicatiepoorten genoemd en worden aangeduid door COM1 en COM2. Met behulp van het bestand port.dll kan men deze poort vanuit Visual Basic bedienen.



Figuur 1: Uitzicht van een seriële poort

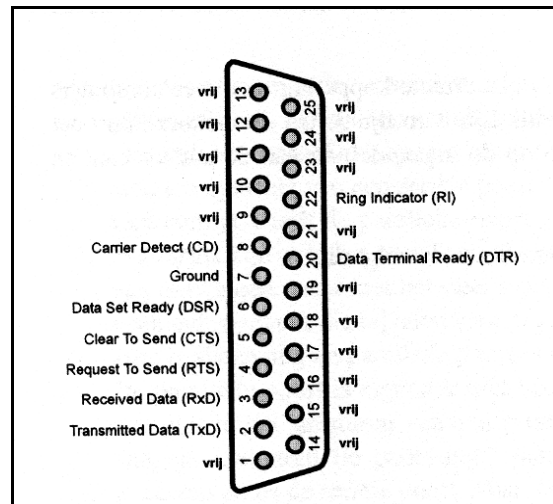
Seriële poorten mogen dan wel langzaam zijn, ze hebben wel doorslaggevende voordelen:

- Bijna iedere computer heeft een seriële poort.
- De basisdefinitie voor gegevensverwerking en penconfiguratie heeft een lange geschiedenis achter zich. Al in 1969 is door de EIA (Electronic Industries Associations) de RS-232-standaard vastgelegd, die tot vandaag niet is gewijzigd. Dus iets die zo lang in gebruikt is, kan nog zo slecht niet zijn.
- Zolang het niet op hoge snelheid aankomt, is de seriële poort met een volwassen chiptechnologie en zijn eenvoudig concept de meest voor de hand liggende basis. Geen

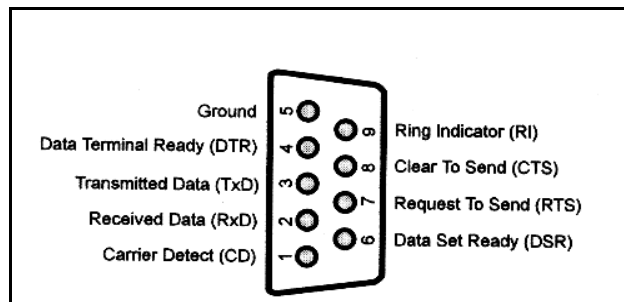
wonder dat in het begin de muis en klavier van de pc op de COM-poort aangesloten werd (nu wordt de USB-poort daarvoor gebruikt). De hoeveelheid te versturen gegevens zijn hierbij niet groot.

- Veel machines hebben ook seriële poorten. De eerste computers werden voor de besturing van die machines ingezet om de gegevens van een sensor te verwerken en indien nodig wijzigingen aan te brengen. Deze informatie werd ontvangen via de eerste seriële poort en de besturing van de machine gebeurde m.b.v. de tweede poort.
- De basisbedrading van de RS-232-standaard volstaat met een drie-aderige kabel voor de verbinding van 2 computers. Eén ader is hierbij de massaleiding, de andere twee worden voor de gegevensoverdracht gebruikt: zender en ontvanger van gegevens. U vraagt zich af waarom er 9 of 25 polen worden gebruikt voor een drie-aderige verbinding. Waar dient de rest van de pennen dan voor? Eigenlijk is het antwoord simpel. De EIA (Electronic Industries Association) en de CCITT (Consultative Committee for International Telegraph and Telephone), een comité dat zich bezig houdt met standaardisatie op gebied van telefonie en telegrafie, hebben nog 6 (voor de 9 pennen) of 17 (voor de 25 pennen) andere lijnen van signalen voorzien. Deze signalen maken het sturen van de informatiestroom flexibeler en betrouwbaarder. Voor een verbinding met een modem worden nog 6 extra signalen toegepast. Met behulp van deze signalen kan de modem en de pc de gegevensstroom in goede banen leiden. Dit is noodzakelijk wanneer de modem bijvoorbeeld de gegevens sneller ontvangt dan de computer kan verwerken. Dan zijn er signaallijnen nodig om de modem af te remmen. Zou dat niet gebeuren, gaan er gegevens verloren doordat deze niet snel genoeg opgenomen kunnen worden. Gelukkig is het niet nodig om al die signalen te gebruiken. In feite zijn slechts drie aders noodzakelijk om een seriële overdracht mogelijk te maken.

Een 25- en 9-polige bus ziet er als volgt uit:



Figuur 2: Penzetting bij 25-polige bus



Figuur 3: Penzetting bij 9-polige bus

Bij een directe koppeling van twee computers moeten de lijnen 2 en 3 verwisseld worden. Zowel bij figuur 2 als figuur 3. Op die manier komt het uitgaande signaal van de ene computer aan op de ingaande pen van de andere computer en omgekeerd.

We zien op beide figuren de lijnen CTS, RTS, DTR enz. Dit zijn de lijnen die we met VB kunnen aanspreken via de port.dll. Dit volgt in de bespreking van het programmeren.

2.3 Parallele poort

2.3.1 Het verschil tussen een seriële poort en een parallele poort

Om te beginnen is er een verschil in uiterlijk. De parallele poort bestaat net als de seriële poort uit een 25-polige aansluiting, maar hierbij wordt, in plaats van een male (pennen) uitvoering, een de female (gaten) uitvoering gebruikt.



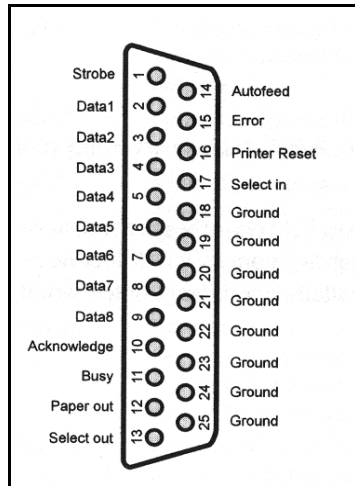
Figuur 4: Uitzicht van een parallele poort (female)

Gewoonlijk wordt de parallele poort gebruikt om de printer aan te sturen. Natuurlijk zijn er in het verleden ook printers geweest die hun gegevens serieel konden ontvangen. Maar met het steeds sneller worden van de printers en de grote hoeveelheid gegevens, die bij talen voor paginabeschrijving zoals PostScript gebruikelijk zijn, is alleen nog de parallele poort in staat om aan de eisen te voldoen.

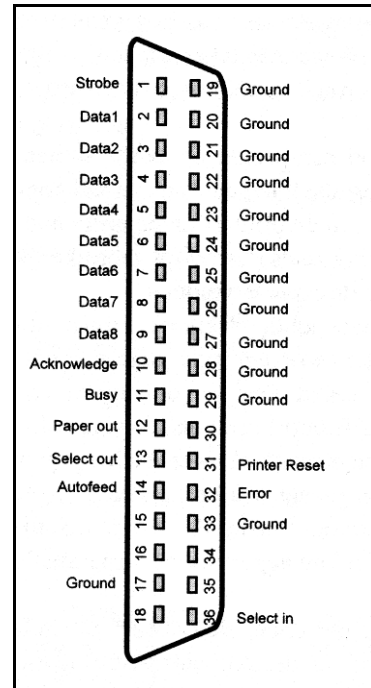
De parallele poort is eigenlijk niet bedoeld voor tweerichtingsverkeer (zenden en ontvangen), omdat zijn taak ligt in het aansturen van een printer. De printer moet tenslotte gegevens ontvangen. De parallele poort heeft dan ook geen buffer voor het ontvangen van gegevens zoals de seriële poort. Maar toch bezit de parallele poort enkele signaallijnen naast de acht datalijnen. Zo zijn er signalen die de pc waarschuwen indien het papier op is of dat de printer offline is. Moeten er nu gegevens via de parallele poort verstuurd of ontvangen worden, dan kunnen deze signaallijnen daarvoor “misbruikt” worden.

Wanneer u een printerkabel wat nader bekijkt, zult u zien dat er aan de ene kant een 25-polige connector te vinden is en aan de andere kant een 36-polige connector, met name de Centronics-connector.

Op volgende figuren zie je welke signalen op de bussen te vinden zijn en ook het verband van deze signalen op beide connectoren:



Figuur 5: Parrelle poort (female)



Figuur 6: De Centronics-connector

3 Bespreking van de schuifregisters

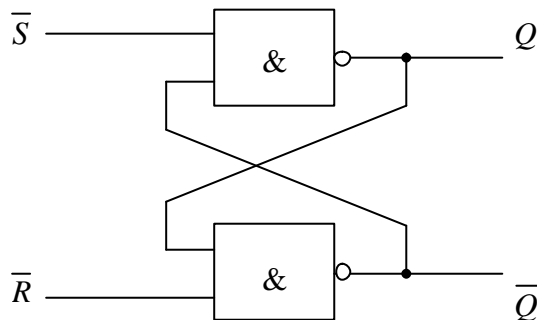
Het doel van een schuifregister (HEF4094B) is om de seriële data in te lezen, om te zetten naar parallel en de data tijdelijk op te slaan in een geheugen. Dit parallelle signaal kan nadien worden uitgelezen uit dit geheugen.

Omgekeerd kan het natuurlijk ook. Het parallelle signaal wordt ingelezen en tijdelijk opgeslagen in een geheugenbuffer. Dit signaal kan dan serieel (bit per bit) terug worden uitgelezen (HEF4021B).

Een schuifregister bestaat uit een reeks D-flipflops. Vooraleer we de werking van het schuifregister zelf verklaren moeten we weten hoe een D-FF is opgebouwd.

3.1 D-flipflop

Een D-FF is opgebouwd uit een \overline{RS} -FF en bezit de mogelijkheid om één bit op te slaan. In volgend schema zie je hoe de D-FF tot stand is gekomen.

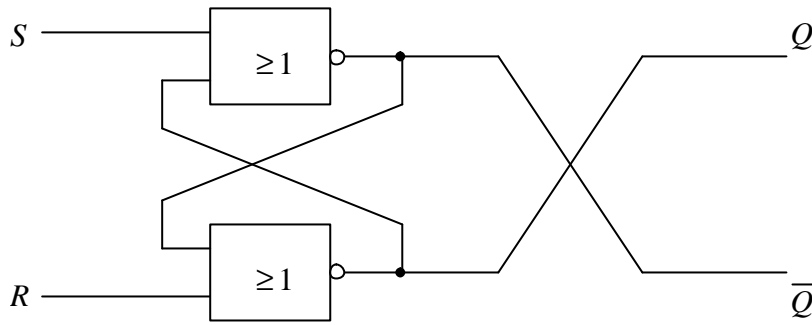


\overline{RS} -FF met NAND-poorten

Ingangen		Uitgangen		Beschrijving
Set	Reset	Q	\overline{Q}	
0	0	Qn	\overline{Qn}	Geheugen toestand
0	1	1	0	Set
1	0	0	1	Reset
1	1	0	0	Verboden toestand

Qn is de vorige toestand van Q

\overline{Qn} is de vorige toestand van \overline{Q}



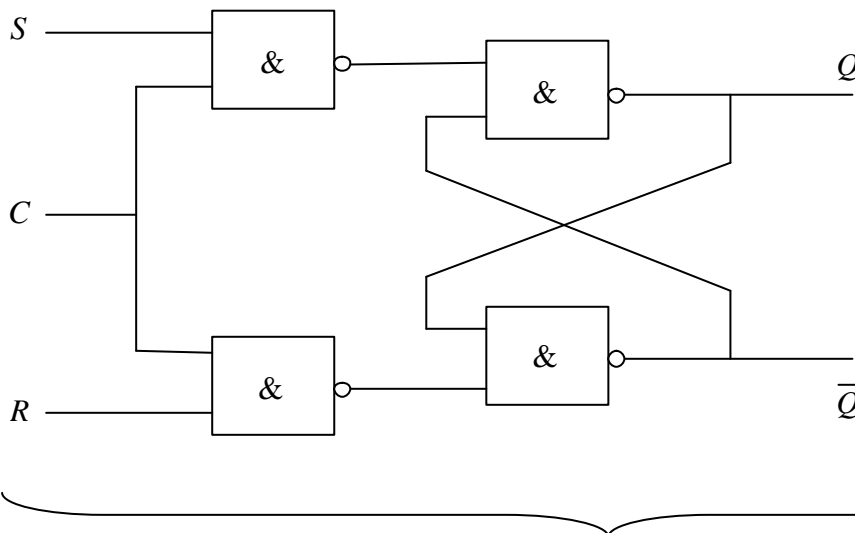
RS-FF met NOR-poorten

Ingangen		Uitgangen		Beschrijving
Set	Reset	Q	\bar{Q}	
0	0	Q _n	\bar{Q}_n	Geheugen toestand
0	1	0	1	Reset
1	0	1	0	Set
1	1	0	0	Verboden toestand

Q_n is de vorige toestand van Q

\bar{Q}_n is de vorige toestand van \bar{Q}

Deze flipflop wordt nu voorzien van een klokingang zodat elke verandering aan de ingangen (\bar{R} en \bar{S}) niet zichtbaar gemaakt wordt op de uitgangen (Q en \bar{Q}).

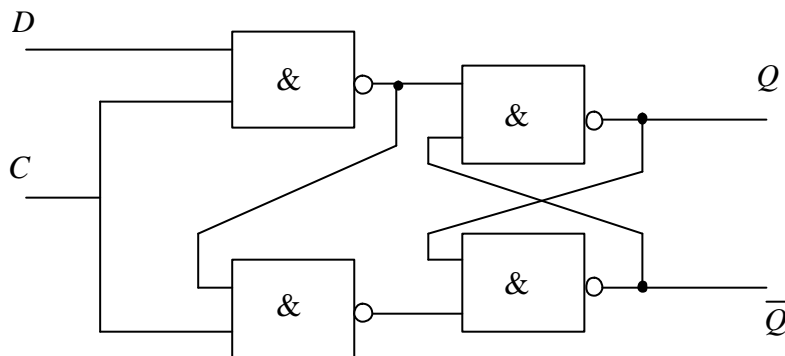


Gepoorte RS-FF

Ingangen			Uitgangen		Beschrijving
Set	Reset	Clock	Q	\bar{Q}	
X	X	0	Q _n	\bar{Q}_n	Geheugen toestand
0	0	1	Q _n	Q _n	Geheugen toestand
0	1	1	0	1	Reset
1	0	1	1	0	Set
1	1	1	1	1	Verboden toestand

\bar{Q}_n is de vorige toestand van \bar{Q}
 Q_n is de vorige toestand van Q
 X is gelijk welke toestand

Om de verboden toestand weg te werken worden de Set en de Reset ingangen vervangen door één data-ingang.



D-flipflop

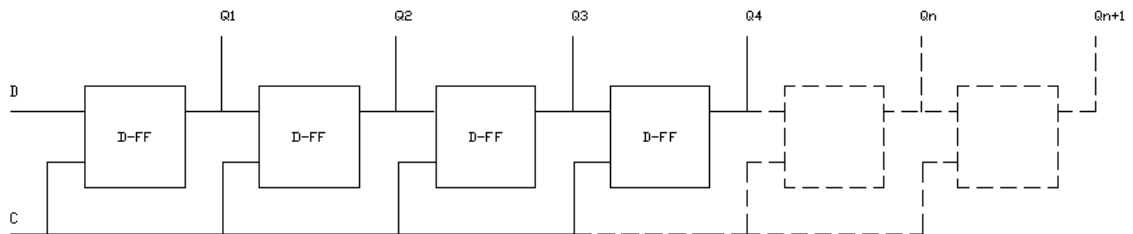
Ingangen		Uitgangen		Beschrijving
Data	Clock	Q	\bar{Q}	
X	0	Q _n	\bar{Q}_n	Geheugen toestand
0	1	0	1	Reset
1	1	1	0	Set

Q_n is de vorige toestand van Q
 \bar{Q}_n is de vorige toestand van \bar{Q}
 X is gelijk welke toestand

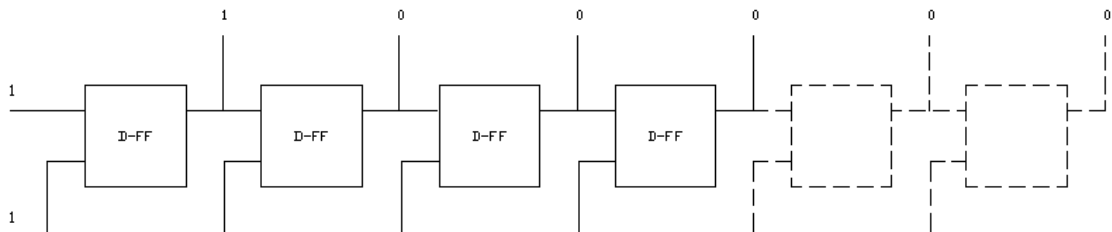
3.2 Algemene werking van een schuifregister

3.2.1 Serieel naar parallel

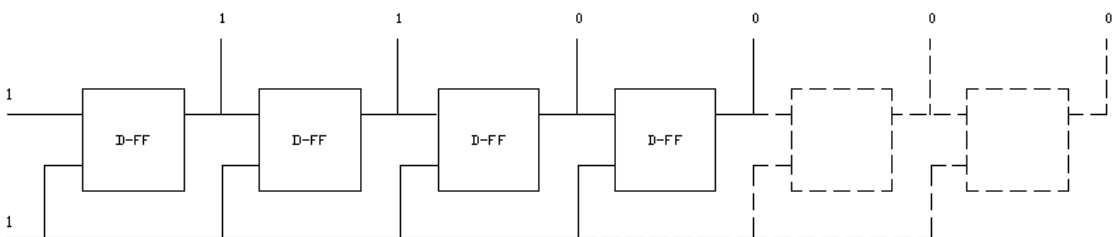
De opbouw van een schuifregister ziet er nu als volgt uit:



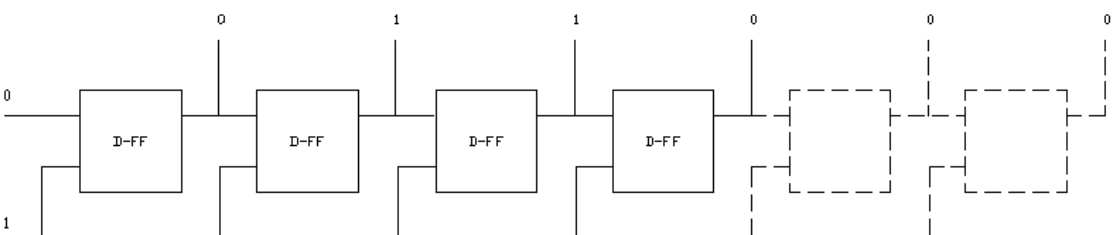
Stel nu dat we een 1-sigitaal op de eerste ingang plaatsen en we geven een klokpuls. Dan kunnen we uit de voorgaande bespreking van de D-FF afleiden dat Q1 een 1-sigitaal zal bezitten. De rest blijft laag (0-sigitaal) aangezien hun ingangen 0 waren.



We laten nu ingang D hoog (m.a.w. een 1-sigitaal) en geven nog een klokpuls. Aangezien Q1 opnieuw een ingang (D) is van een volgende D-FF kunnen we besluiten dat uitgang Q2 eveneens hoog zal worden (1-sigitaal) en Q1 wordt net zoals bij vorige klokpuls hoog. De rest blijft laag.



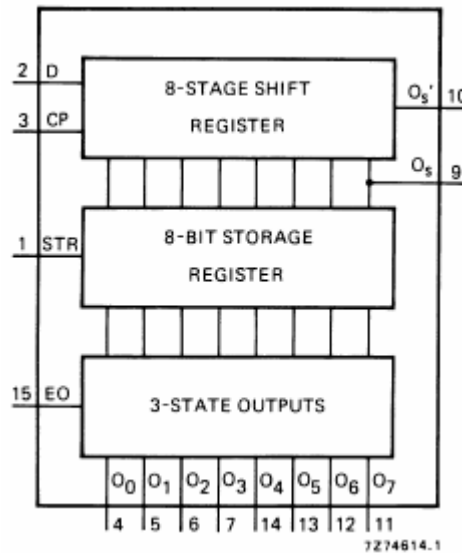
Voor de verandering houden we ingang D laag (0-sigitaal) en geven opnieuw een klokpuls. Uitgang Q1 wordt dus laag, Q2 en Q3 hoog. De rest blijft nog altijd laag.



3.2.2 Serieel naar parallel toegepast (HEF4094B)

Het IC HEF4094B bestaat uit 3 delen: een 8-bit schuifregister, een 8-bit storage-register en de 3-state-outputs (parallele uitgangen O0 tot O7).

Opbouw:



PINNING

D	data input	EO	output enable input
CP	clock input	O ₅ , O ₅ '	serial outputs
STR	strobe input	O ₀ to O ₇	parallel outputs

De data-ingang D wordt nu gebruikt voor het lezen van de seriële informatie.

Bij iedere positief flank op de CLOCK (CP) wordt de data bit per bit ingelezen in de flipflop (zie de bespreking van de schuifregisters). Het parallele uitgangssignaal, gevormd door de seriële data, kunnen we opslaan in het storage-register door de ingang STROBE (STR) hoog te maken. Tot slot kunnen we dit signaal vormen op de "3-STATE OUTPUT" (uitgangen O0 tot O7) door op de ingang "OUTPUT ENABLE" (EO) een 1 signaal te plaatsen.

De uitgangen O₅ en O₅' worden gebruikt om nog een IC aan te sluiten (D aansluiten op O₅). Naar gelang het aantal IC's die in serie worden geplaatst kunnen we een 16-bit signaal, 24-bit signaal enz ... verwerken.

Om duidelijk te laten zien hoe de IC opgebouwd is vind je hieronder een schematische voorstelling.

Functietabel:

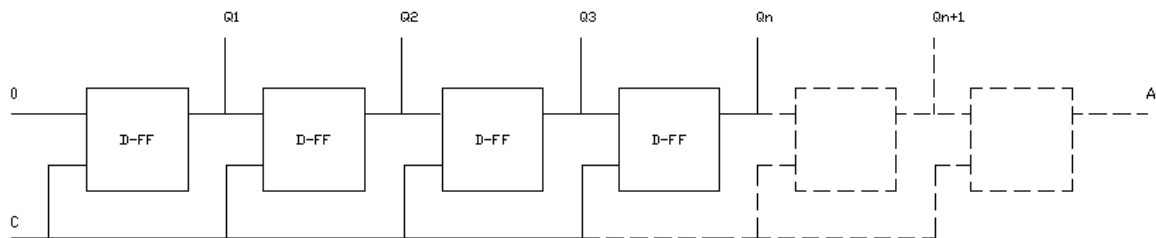
INPUTS				PARALLEL OUTPUTS		SERIAL OUTPUTS	
CP	EO	STR	D	O ₀	O _n	O _s	O' _s
↗	L	X	X	Z	Z	O' ₈	nc
↘	L	X	X	Z	Z	nc	O ₇
↗	H	L	X	nc	nc	O' ₈	nc
↗	H	H	L	L	O _{n-1}	O' ₈	nc
↗	H	H	H	H	O _{n-1}	O' ₈	nc
↘	H	H	H	nc	nc	nc	O ₇

Notes

1. H = HIGH state (the more positive voltage)
2. L = LOW state (the less positive voltage)
3. X = state is immaterial
4. ↗ = positive-going transition
5. ↘ = negative-going transition
6. Z = high impedance off state
7. nc = no change
8. O'₈ = the information in the seventh shift register stage

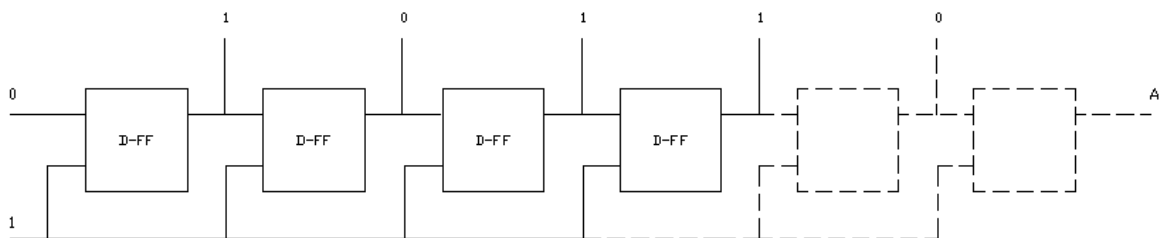
3.2.3 Parallel naar serieel

Dit gebeurt eigenlijk op dezelfde manier als het omzetten van serieel naar parallel. Het enige verschil is dat de uitgangen nu zogezegd als ingangen worden beschouwd.

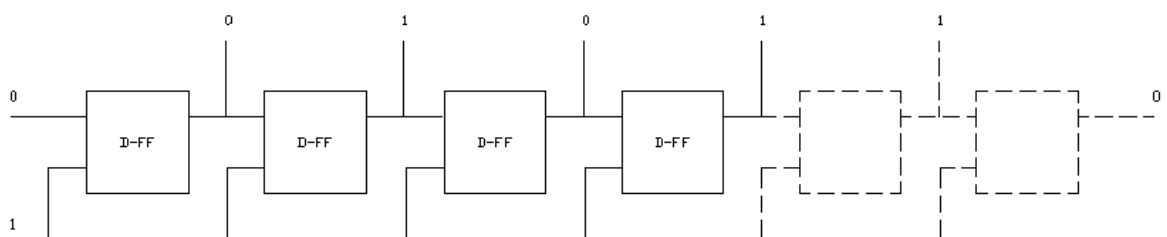


We lezen het parallelle signaal in met de ingangen Q1 tot Qn+1 (n is afhankelijk van het aantal karakters van het signaal), dus m.a.w. heeft elke DFF een ingang. Geven we een klokpuls dan wordt het signaal ingelezen en tijdelijk opgeslagen in de flipflops. Bij elke volgende klokpuls wordt het parallelle signaal serieel uitgelezen van rechts naar links. Tijdens het uitlezen worden de ingangen van links naar rechts terug laag (op 0) gebracht.

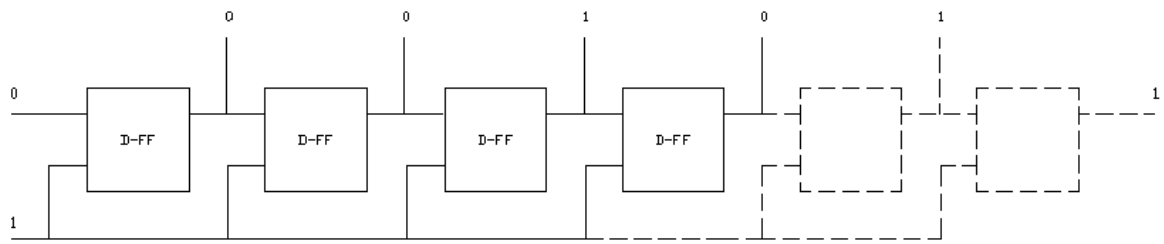
Voorbeeld: het signaal 10110 serieel uitlezen



We geven de eerste klokpuls



We geven een tweede klokpuls

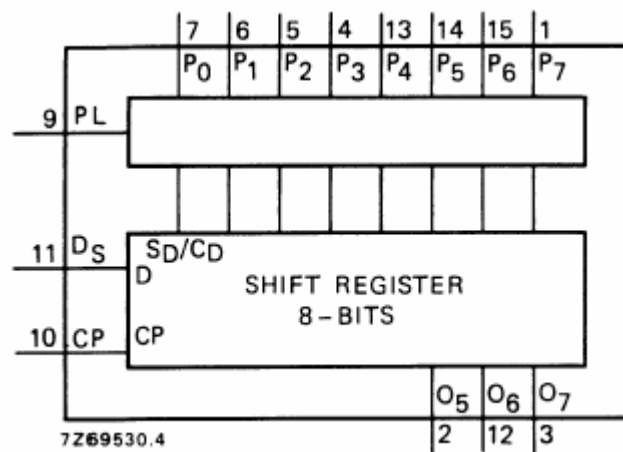


We geven zoveel klokpulsen als er karakters zijn in het parallelle signaal, en op deze manier hebben we het volledige parallelle signaal, serieel gemaakt.

3.2.4 Parallel naar serieel toegepast (HEF4021B)

Het IC HEF4021B is de tegenhanger van de HEF4094B. In plaats van het omzetten van een serieel naar een parallel signaal doet die IC net het omgekeerde. Hij is een 8-bit schuifregister met een synchrone seriële data-ingang (DS), een asynchrone parallelle laadingang (PL), acht asynchrone parallel data-ingangen (P0 tot P7) en 3 seriële uitgangen verbonden met de laatste 3 parallelle ingangen (zie schema).

Opbouw:



PINNING

PL	parallel load input
P ₀ to P ₇	parallel data inputs
D _S	serial data input
CP	clock input (LOW to HIGH edge-triggered)
O ₅ to O ₇	buffered parallel outputs from the last three stages

Het schuifregister bestaat uit verscheidene D-flipflops met een 'set direct' / 'clear direct' ingang. De informatie op ingangen P0 tot P7 wordt asynchroon geladen in het register terwijl ingang PL hoog is, onafhankelijk van CP en DS. Als PL laag is wordt de data op DS op de eerste positie (aan de linker kant) van het schuifregister geplaatst en alle andere data schuift één positie naar rechts op bij een positieve flank van de Clock (CP). Door deze opschuiving van data is het mogelijk het parallelle signaal als het ware serieel naar buiten te sturen over de uitgangen O5-O7. Welke seriële uitgang je gebruikt hangt af van het aantal bits die je parallel signaal bevat. In ons geval bevat het signaal 8 bits, dus gebruiken we O7 als seriële uitgang. In de onderstaande figuur zie je de werking van het IC.

Functietabellen

Serial operation

n	INPUTS			OUTPUTS		
	CP	D _s	PL	O ₅	O ₆	O ₇
1	↗	D ₁	L	X	X	X
2	↗	D ₂	L	X	X	X
3	↗	D ₃	L	X	X	X
6	↗	X	L	D ₁	X	X
7	↗	X	L	D ₂	D ₁	X
8	↗	X	L	D ₃	D ₂	D ₁
	↘	X	L	no change		

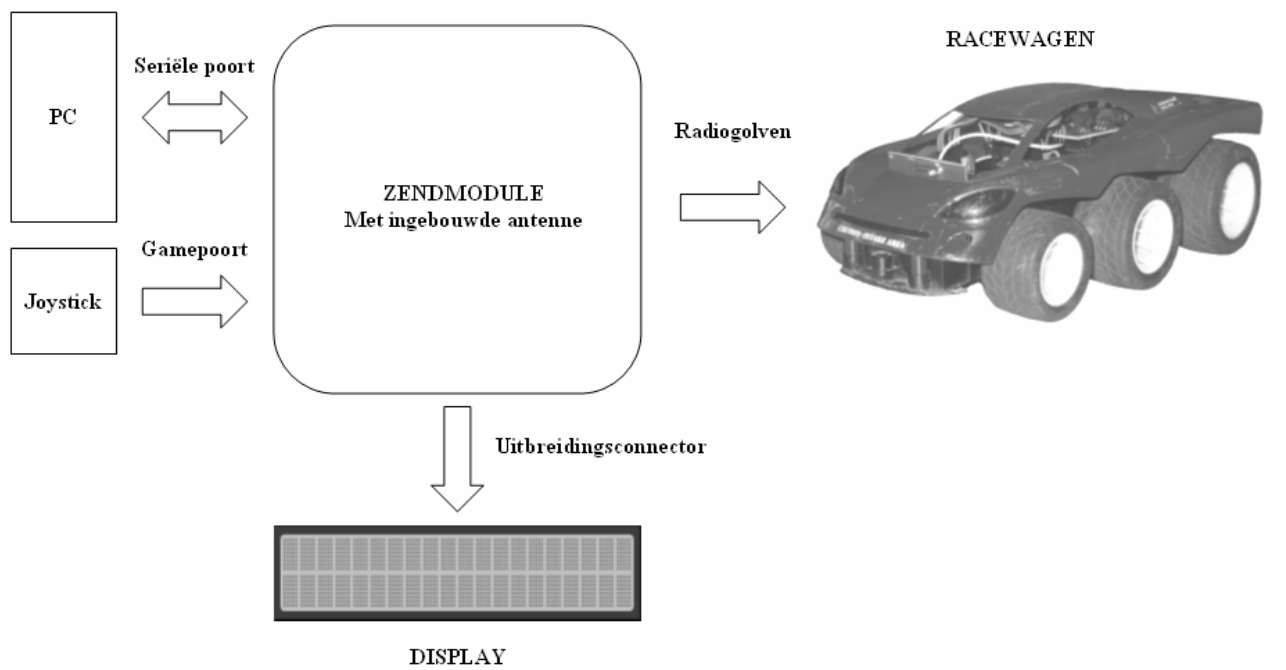
Parallel operation

n	INPUTS			OUTPUTS		
	CP	D _s	PL	O ₅	O ₆	O ₇
	X	X	H	P ₅	P ₆	P ₇

Notes

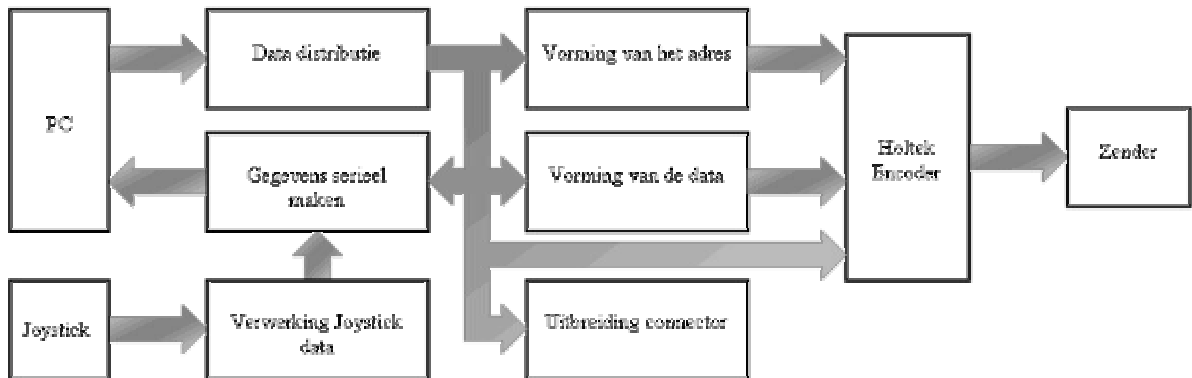
- H = HIGH state (the more positive voltage)
 L = LOW state (the less positive voltage)
 X = state is immaterial
 ↗ = positive-going transition
 ↘ = negative-going transition
 D_n = either HIGH or LOW
 n = number of clock pulse transitions

4 Overzicht van de hardware



5 Opbouw Zendmodule

5.1 Blokschema



Figuur 7: Blokschema zendmodule

Het doel van de zendmodule is het verwerken van de computergegevens tot een datasignaal die bruikbaar is voor de racewagens. Dit signaal bevat het adres en de stuurcommando's voor de motoren. Dit adres is nodig om de racewagen te identificeren, aangezien 256 racewagens gestuurd kunnen worden. De racewagen vergelijkt het ontvangen adres met zijn vast adres. Indien deze adressen overeenstemmen met elkaar worden de stuurcommando's uitgevoerd.

Het samenvoegen van het adres en de commando's zelf gebeurt m.b.v. de 'Holtek Encoder' en wordt daarna draadloos verstuurd.

Omdat we ook data wilden inlezen via de seriële poort hebben we een extra optie voorzien, namelijk een joystick. De spanning over de potentiometers in de joystick wordt vergeleken met een referentiespanning. Aan de hand van deze spanningen kunnen we de positie van de joystick bepalen. De opgemeten gegevens worden serieel naar de computer gestuurd.

Er is eveneens een uitbreidingsconnector voorzien. Hierop kunnen we een display aansluiten om bepaalde berichten te tonen.

Wat gebeurt er nu zoal in de zendmodule?

De gegevens die door de pc worden uitgezonden worden door de datadistributie verder verspreid over de andere delen van de zendmodule. Zo wordt door de datadistributie de twee blokken “vorming van de data” en “vorming van het adres” aangesproken om het eigenlijke signaal te vormen die door de Holtek encoder in 1 signaal wordt omgezet. Tijdens het gehele proces kunnen ook teksten, info of instructies worden weergegeven op de display. Ook dit wordt aangesproken door de datadistributie.

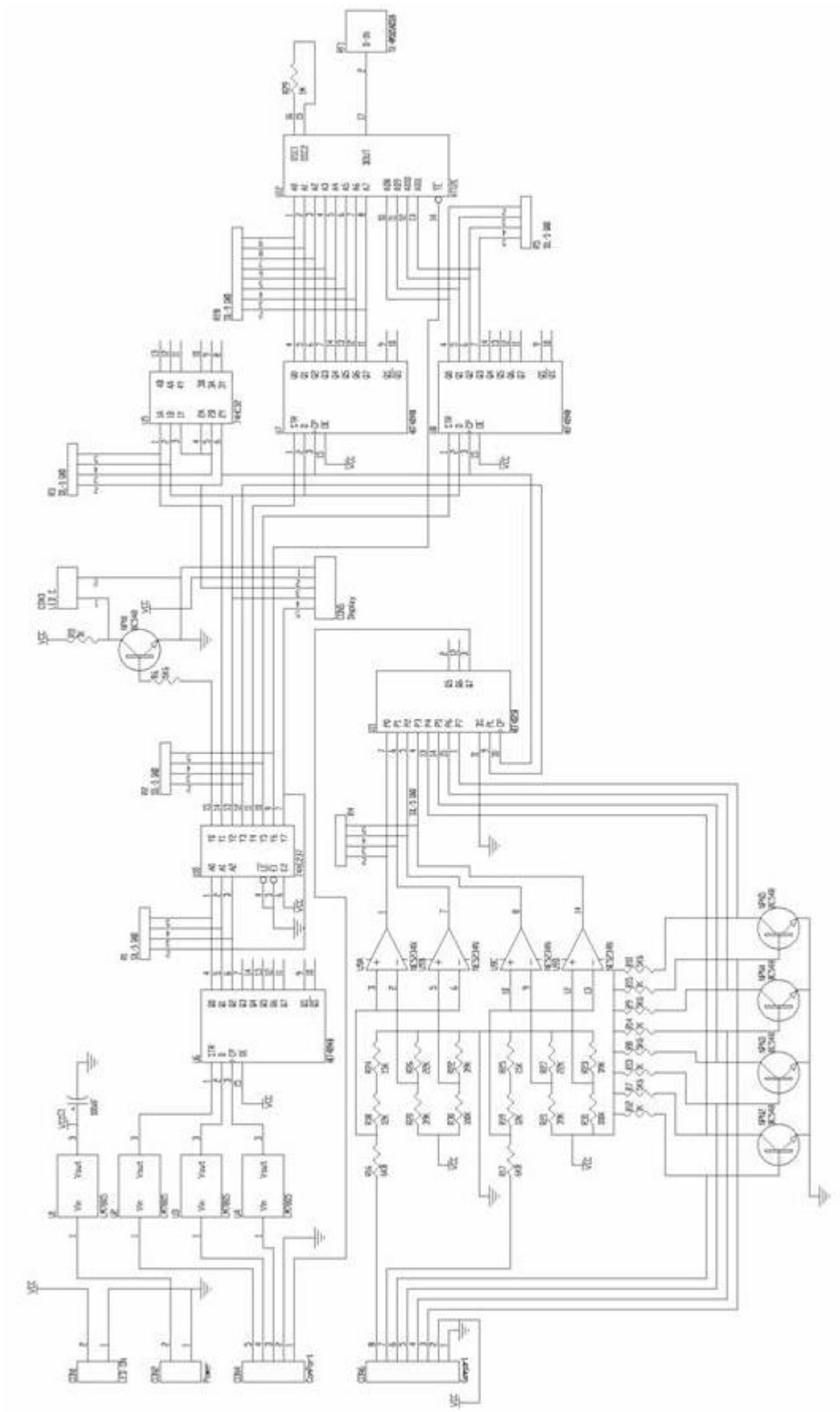
Voor de joystick nemen we dan wel een andere route. Eerst verwerken we de gegevens van de joystick (potentiometers) zodat die bruikbaar zijn voor de pc. Daarna worden de gegevens door de computer ingelezen.

Er moet vooraf wel een keuze gemaakt worden:

- ofwel bedien je de wagen met de joystick;
- ofwel met de pc.

Het tegelijkertijd bedienen van de racewagen met joystick en met de computer is niet mogelijk. Dit is om complicaties te voorkomen. Na dit proces is het signaal klaar om verzonden te worden naar de wagen.

5.2 Elektronisch schema

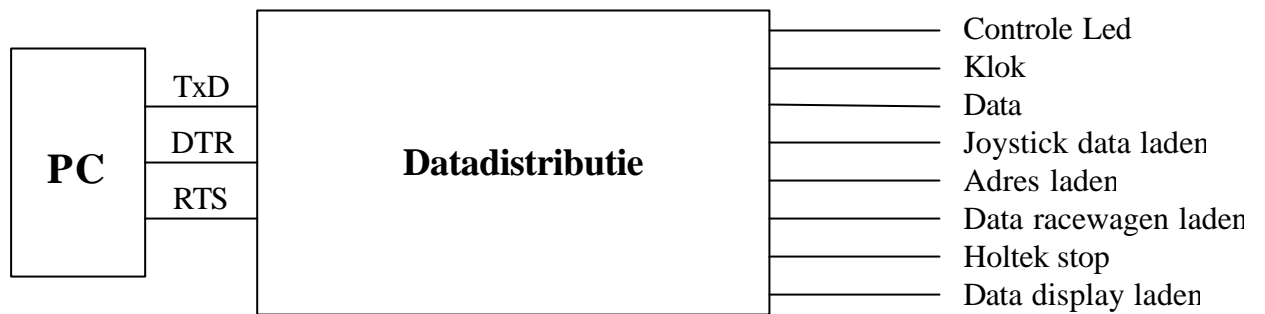


5.3 Datadistributie

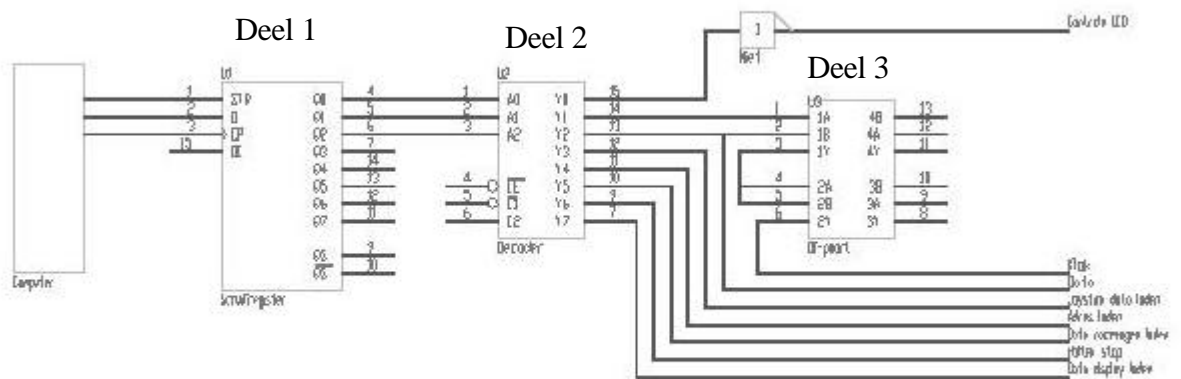
5.3.1 Doel:

De data distributie bevat drie delen: een schuifregister, een multiplexer/decoder en een OF poort.

Het heeft als doel alle componenten van de volledige schakeling te besturen.

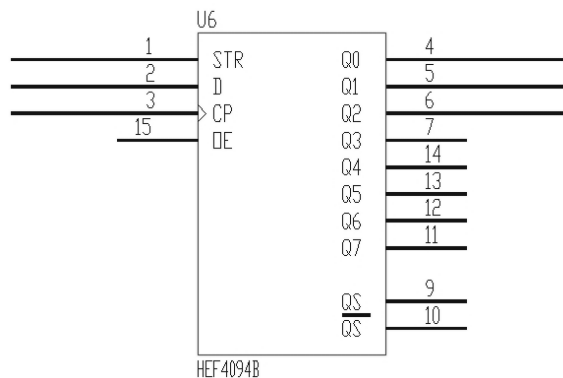


5.3.2 Schema:



5.3.3 Opbouw 1ste deel:

De datadistributie is aangesloten op de seriële poort van de computer. Voor de computer gestuurde racewagen worden er slechts drie van de vijf uitgangen gebruikt, namelijk Txd, DTR, RTS. (Zie seriële poort) Deze uitgangen dienen om het eerste deel van de datadistributie aan te sturen, namelijk een schuifregister met acht uitgangen. Deze zet de seriële gegevens om naar parallelle gegevens. Die seriële gegevens worden gevormd door middel van een zelf geschreven programma.



5.3.4 Programma:

```
Private Sub Distribution(Channel As Byte)
If Check = True Then
  Dim TellerI As Byte
  RTS 0
  For TellerI = 0 To 2
    If (4 / (2 ^ TellerI) And Channel) > 0 Then
      TXD 1
    Else
      TXD 0
    End If
    DTR 1
    DTR 0
  Next TellerI
  RTS 1
  RTS 0
End If
End Sub
```

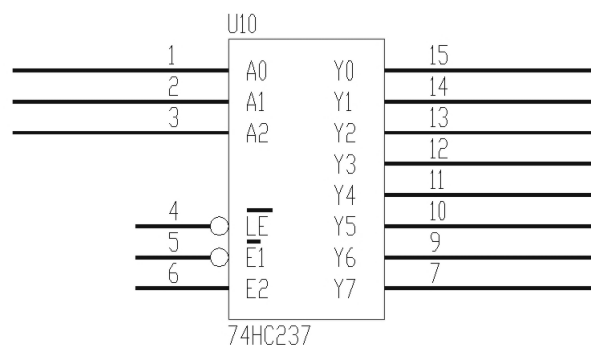
Verklaring van dit stukje code bevindt onder het hoofdstuk software CGRPrintplaat.dll.

5.3.5 Omschrijving code met schuifregister:

Het decimaal getal 'Channel' wordt via de code omgezet naar een binair getal dat serieel wordt verzonden over het kanaal 'TXD'. In de datadistributie worden er maar drie van de acht uitgangen van het schuifregister gebruikt. Het kanaal 'DTR' dient als klokpuls om het schuifregister synchroon met de computer te laten werken. De klokpuls wordt door de code gegenereerd nadat er een 1 of een 0 op het kanaal 'TXD' is geplaatst. Nadat de teller zijn cyclus drie maal heeft doorlopen, wordt er nog een korte puls over het 'RTS' kanaal verzonden. Die puls wordt gebruikt om de ingelezen data van het schuifregister op de uitgangen te plaatsen.

5.3.6 Opbouw 2de deel:

Het 2de deel van de datadistributie bestaat uit een multiplexer/decoder en heeft als functie het onafhankelijk aansturen van de te besturen componenten (adresvorming, datavorming, display, joystick). Deze multiplexer/decoder heeft acht uitgangen (Y0...Y7) en drie adres ingangen (A0... A2). Die drie ingangen zijn verbonden met de uitgangen van het schuifregister. De multiplexer/decoder werkt als volgt, als het getal 5 (binair =101) op de adresingangen gevormd wordt, zal er op de uitgang Y5 een 1 signaal tevoorschijn komen en de andere uitgangen blijven 0.



5.3.7 Code in samen werking met multiplexer/decoder:

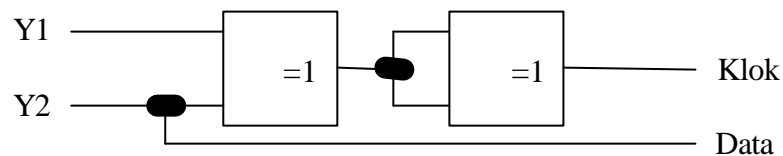
Doordat de ingangen van de multiplexer/decoder verbonden zijn met de uitgangen van het schuifregister, kan het programma de acht uitgangen bedienen. Dus als de vijfde uitgang 1 moet zijn dan wordt gewoon 'Channel' gelijk aan vijf gesteld en de code wordt afgespeeld.

5.3.8 Opbouw 3de deel:

De bedoeling van de datadistributie is het aansturen van andere schuifregisters. Om een 1 signaal in te lezen bij een schuifregister moet de datalijn **en** de klok op een gegeven moment beiden gelijk aan 1 zijn (zie schuifregisters). Maar met de multiplexer/decoder kan maar slechts één uitgang gelijk aan 1 stellen. Om dit probleem op te lossen is er nog een derde deel in de datadistributie aanwezig. Dit derde deel bestaat uit OF-poorten en zal de andere schuifregisters voorzien van een kloksignaal en data.

Als de uitgang Y1 van de multiplexer/decoder gelijk is aan 1 moet de data gelijk zijn aan 0 en het kloksignaal gelijk zijn aan 1. Bij de uitgang Y2 gelijk aan 1, moet de data en de klok gelijk zijn aan 1.

Dus de schakeling ziet er als volgt uit.



Bij de schuifregisters die gekoppeld zijn aan onze datadistributie is het nodig dat het data signaal eerder wordt ingegeven dan het kloksignaal. De oplossing hiervoor is het kloksignaal door twee OF poorten te laten vloeien. De vertraging is wel miniem maar het is wel nodig voor een goede werking.

5.3.9 Gebruik van datadistributie:

De verkregen klok en datalijn zijn verbonden met alle andere schuifregisters in de volledige schakeling, zodat iedere schuifregister dezelfde gegevens krijgt.

Let op: Iedere STR ingang (zie hoofdstuk schuifregisters) van de schuifregisters is verbonden met een andere uitgang van de multiplexer/decoder.

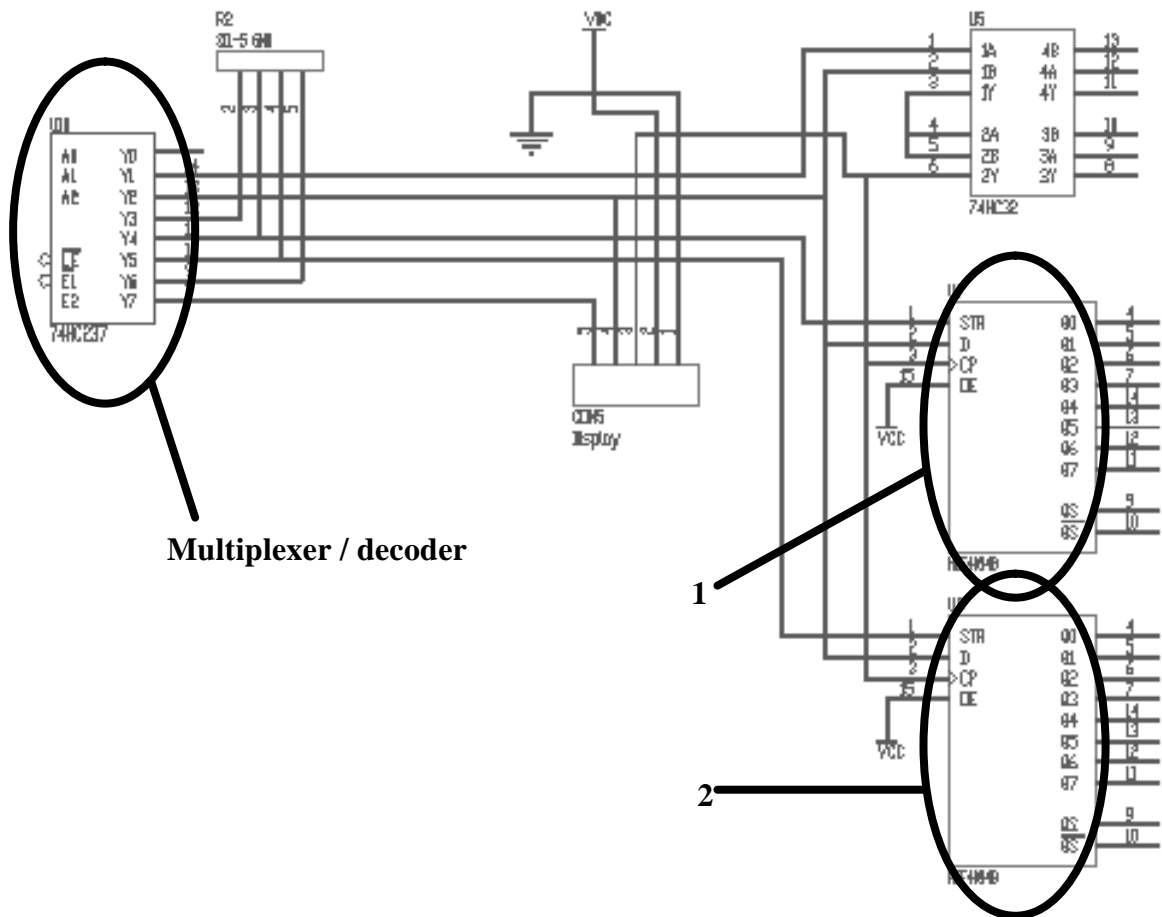
Hieronder volgt een klein overzicht.

Y0:	Rust toestand, is verbonden met controle Led via een Niet-poort
Klok:	Ingang schuifregisters (CP)
Data:	Ingang schuifregisters (D)
Y3:	Joystick data laden in schuifregister
Y4:	Adres laden in schuifregister
Y5:	Data racewagen laden in schuifregister
Y6:	Stoppen met zenden
Y7:	Data display laden in schuifregister

De uitgang Y0 is aan de ingang van een NIET-poort aangesloten. De uitgang ervan is verbonden met een controle LED dat vooraan op het paneel is gemonteerd. Dus de Led zal branden wanneer Y0 gelijk is aan 0.

5.4 Vorming van data en adres

Het vormen van het adres en de data begint bij de datadistributie (zie hoofdstuk datadistributie). De multiplexer/decoder (74HC237) zorgt ervoor dat de schuifregister 1 (datavorming) en de schuifregister 2 (adresvorming) (zie bijhorende figuur) hun informatie krijgen die ze nodig hebben om data en adres te vormen.



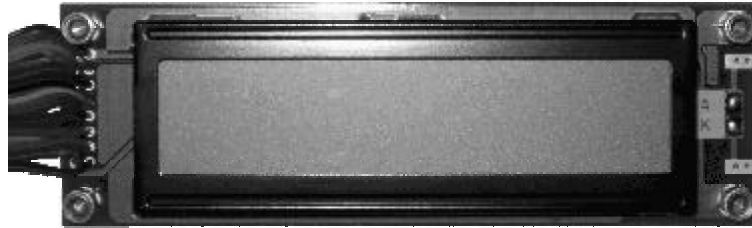
Met de uitgang Y2 (data uitgang) van de datadistributie spreekt men de data-ingang van schuifregister 1 en 2 aan. Hiermee kunnen we het eigenlijke signaal van de seriële informatie omzetten naar parallele gegevens (zie schuifregister). Wanneer voor de eerste bit van het adres een 1 signaal nodig is, komt dit 1-signaal op:

- schuifregister 1 voor de adresvorming
- schuifregister 2 voor de datavorming
- schuifregister 3 voor de display
- de OF-poorten

M.a.w. staat het 1-signaal dus op elk schuifregister. Uit het hoofdstuk over de schuifregisters volgt dat het signaal, opgeslagen in het storage register van het schuifregister, uitgelezen wordt bij een 1-signaal op de STR-klem van het schuifregister. Wanneer er dus een 8 bit signaal gevormd moet worden die als adres moet dienen, geven we enkel een STR-puls, met klem Y4 van de multiplexer/decoder, op schuifregister 1. Indien dit signaal bestemd is als data dan geven we een STR-puls, met klem Y5 van de multiplexer/decoder, op schuifregister 2.

Misschien is het niet helemaal duidelijk waarom de OF-poorten nodig zijn. Wanneer we de D-ingang van het schuifregister willen inlezen geven we een klokpuls. Deze klokpuls mag niet tegelijk of later aankomen dan het signaal op de D-ingang. Aangezien we wegens tekort aan uitgangen van de seriële poort niet in staat waren elke schuifregister van een afzonderlijke klokpuls te voorzien, moest alles gebeuren in één stap. Dit gebeurt namelijk op het moment dat het signaal naar de D-ingang wordt gestuurd. De OF-poorten zorgen voor een kleine vertraging en sturen dan een klokpuls naar de klok van ieder schuifregister. Dit betekent nog niet dat de data uitgelezen is. Alleen het opslaan in het storage-register gebeurt op dat moment. Op deze manier kunnen we alle componenten vlot aanspreken door een goed stukje programma te schrijven.

5.5 Uitbreiding: Display



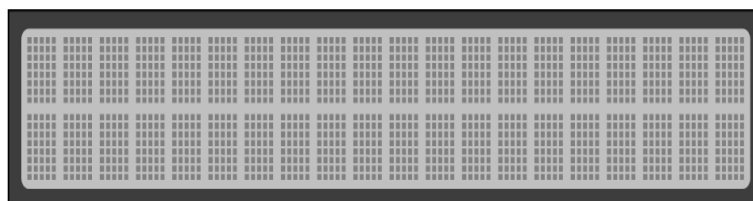
De displaymodule bestaat uit een LCD-scherm en een HD44780 controller. De HD44780 controller heeft twee 8-bit registers, namelijk een instructieregister en een dataregister.

Het instructieregister slaat de instructiecodes op, bijvoorbeeld de display wissen, cursor verplaatsen, adresinformatie voor het displaygeheugen en de karaktergenerator, enz.

Het dataregister slaat de data voor het displaygeheugen en de karaktergenerator tijdelijk op. Om te switchen tussen die 2 registers, is er een R_s lijn voorzien. Indien $R_s = 0$ dan worden de instructies ingelezen, als $R_s = 1$ dan worden de geheugens bediend.

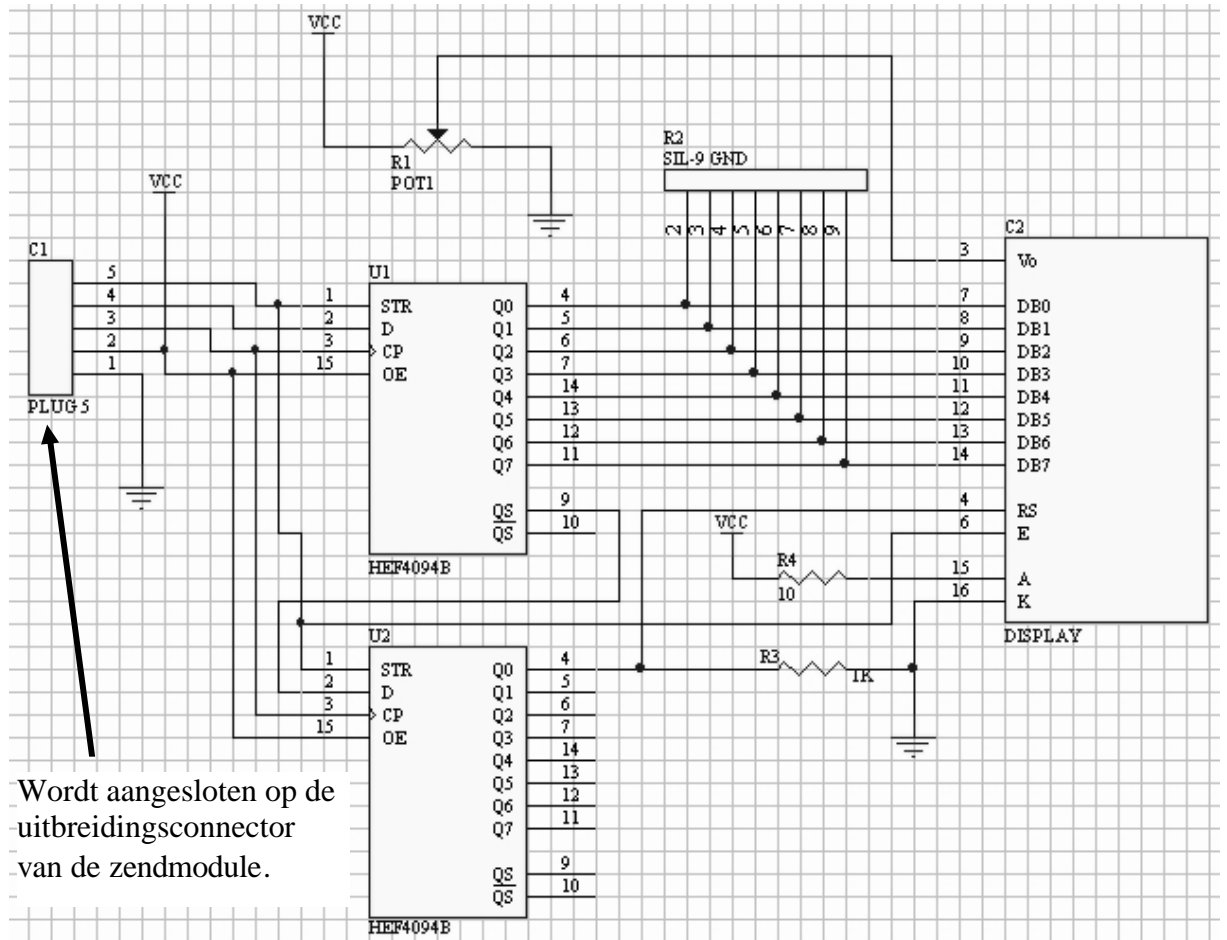
De karaktergenerator bezit 192 verschillende tekens en bezit nog een klein karaktergeheugen om acht zelfgemaakte tekens op te slaan.

De display bezit ook nog een displaygeheugen om de plaats van de tekens op het scherm te onthouden. Deze display heeft een capaciteit van 80 karakters. Verder kan de display verschuiven over de 80 opgeslagen karakters.



5.5.1 Schema

De onderstaande schakeling is aan de uitbreidingsconnector van de zendmodule aangesloten.



De display wordt bediend met twee schuifregisters die in serie staan. Het eerste schuifregister wordt volledig benut en bij de tweede schuifregisters enkel de eerste uitgang. De 8 data ingangen van de display zijn gekoppeld aan het eerste schuifregister. De Rs ingang is aan het tweede schuifregister aangesloten.

De D ingang van het schuifregister is verbonden aan de Data uitgang van de datadistributie.

De CP ingang van het schuifregister is verbonden aan de Klok uitgang van de datadistributie.

De STR ingang van het schuifregister is verbonden aan de uitgang Y7 van de datadistributie. (zie hoofdstuk datadistributie)

De E (enable) ingang van de display is eveneens verbonden aan de uitgang Y7 van de datadistributie. Op het moment dat de uitgangen van het schuifregister worden aangepast door een puls van Y7, wordt de informatie van het schuifregister in de display gelezen.

Met behulp van de een potentiometer POT1 wordt het contrast van de display ingesteld.

5.5.2 Programma

```
Private Sub DisplayValue(Waarde As Integer)
Dim Teller As Byte
For Teller = 0 To 9
    If (512 / (2 ^ Teller) And Waarde) > 0 Then
        Distribution 2
    Else
        Distribution 1
    End If
    Distribution 0
Next Teller
Distribution 7
DELAY 2
Distribution 0
End Sub
```

Met dit programma kunnen we de twee schuifregisters voor de displaydata bedienen. De waarde, dat in de code wordt ingegeven, wordt serieel naar de schuifregisters verzonden.

Zij zetten dan die seriële data om naar parallelle informatie en wordt dan in de display gelezen.

5.5.3 Display Instructies

Om de display te kunnen bedienen zij er enkele vastgelegde instructies.

Instructie	Rs	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
Clear display	0	0	0	0	0	0	0	0	1

Deze instructie plaatst in iedere DDRAM adres een spatie.

DDRAM adres wordt op 0 geplaatst.

I/D van de 'Entry mode set' wordt op 1 geplaatst.

S van de 'Entry mode set' verandert niet.

```
Public Sub Display_Clear()
    DisplayValue 1
End Sub
```

Instructie	Rs	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
Return home	0	0	0	0	0	0	0	1	-

DDRAM adres wordt op 0 geplaatst.

De display en de cursor worden naar hun start positie verschoven.

```
Public Sub Display_ReturnHome()
    DisplayValue 2
End Sub
```

Instructie	Rs	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
Entry mode set	0	0	0	0	0	0	1	I/D	S

Bij het ingeven van een nieuw karakter in DDRAM gebeurt er het volgende:

S = 0 & I/D = 0 : de cursor verplaatst zich naar links, display blijft op de zelfde plaats

S = 0 & I/D = 1 : de cursor verplaatst zich naar rechts, display blijft op de zelfde plaats

S = 1 & I/D = 0 : de display verschuift zich naar rechts, cursor blijft op de zelfde plaats

S = 1 & I/D = 1 : de display verschuift zich naar links, cursor blijft op de zelfde plaats

```
Public Sub Display_EntryModeSet(I_D As Boolean, SH As Boolean)
    Dim EntryModeSet As Integer
    EntryModeSet = 4 + Abs(I_D) * 2 + Abs(SH) * 1
    DisplayValue EntryModeSet
End Sub
```

Instructie	Rs	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
Display on/off control	0	0	0	0	0	1	D	C	B

D = 0 : Display is uit

D = 1 : Display is aan

C = 0 : Cursor is onzichtbaar

C = 1 : Cursor is zichtbaar

B = 0 : Knipperend vlak is onzichtbaar

B = 1 : Knipperend vlak is zichtbaar

```
Public Sub Display_ON_OFF(D As Boolean, C As Boolean, B As Boolean)
    Dim ON_OFF As Integer
    ON_OFF = 8 + Abs(D) * 4 + Abs(C) * 2 + Abs(B) * 1
    DisplayValue ON_OFF
End Sub
```

Instructie	Rs	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
Cursor or display shift	0	0	0	0	1	S/C	R/L	-	-

S/C = 0 & R/L = 0 : Cursor verplaatst zich 1 stap naar links

S/C = 0 & R/L = 1 : Cursor verplaatst zich 1 stap naar rechts

S/C = 1 & R/L = 0 : Display verschuift zich 1 stap naar links

S/C = 1 & R/L = 1 : Display verschuift zich 1 stap naar rechts

```

Public Sub Display_Shift(SC As Boolean, R_L As Boolean)
    Dim Shift As Integer
    Shift = 16 + Abs(SC) * 8 + Abs(R_L) * 4
    DisplayValue Shift
End Sub

```

Instructie	Rs	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
Function set	0	0	0	1	DL	N	F	-	-

Met DL stellen we de lengte van de interface in.

DL = 1 : Gegevens worden over 8-bit verzonden

DL = 0 : Gegevens worden over 4-bit verzonden

N = 0 : Er wordt maar 1 regel gebruikt van de display, 1 rij van 80 karakters

N = 1 : Er worden 2 regels gebruiken, 2 rijen van 40 karakters

F = 0 : karaktergrootte 5 x 8

F = 1 : karaktergrootte 5 x 10

```

Public Sub Display_SetFunction(DL As Boolean, N As Boolean, F As Boolean)
    Dim SetFunction As Integer
    SetFunction = 32 + Abs(DL) * 16 + Abs(N) * 8 + Abs(F) * 4
    DisplayValue SetFunction
End Sub

```

Instructie	Rs	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
Set CGRAM address	0	0	1	CG	CG	CG	CG	CG	CG

Ingeven van het CGRAM adres.

```
Public Sub Display_SetCgRamAddress(CgRamAddress As Byte)
    Dim SetCgRamAddress As Integer
    If CgRamAddress > 63 Then Exit Sub
    SetCgRamAddress = 64 + Val(CgRamAddress)
    DisplayValue SetCgRamAddress
End Sub
```

Instructie	Rs	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
Set DDRAM address	0	1	DD	DD	DD	DD	DD	DD	DD

Ingeven van het DDRAM adres

```
Public Sub Display_SetDdRamAddress(DdRamAddress As Byte)
    Dim SetDdRamAddress As Integer
    If DdRamAddress > 127 Then Exit Sub
    SetDdRamAddress = 128 + Val(DdRamAddress)
    DisplayValue SetDdRamAddress
End Sub
```

Instructie	Rs	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
Write data to CGRAM or DDRAM	1	D	D	D	D	D	D	D	D

Gegevens wegschrijven ofwel in het DDRAM ofwel in het CGRAM.

Om gegevens weg te schrijven in het CGRAM moet men met de functie 'Set CGRAM address' het juiste adres ingeven. En daarna wordt de functie 'Write data' toegepast.

Om gegevens weg te schrijven in het DDRAM is het zelfde principe geldig. Eerste de functie 'Set DDRAM address' aanspreken en daarna 'Write data'.

```
Public Sub Display_SetWriteData(WriteData As Byte)
    Dim SetWriteData As Integer
    If WriteData > 255 Then Exit Sub
    SetWriteData = 256 + Val(WriteData)
    DisplayValue SetWriteData
End Sub
```

Bij het starten van de display wordt er automatisch een reset functie geactiveerd. Hierbij wordt het volledige karaktergeheugen gewist en worden de instellingen terug naar de oorspronkelijke staat geplaatst.

- Display clear
- Function set

DL = 1 : 8-bit interface data

N = 0 : 1-line display

F = 0 : 5 x 8 puntkarakters

- Display on/off control

D = 0 : Display is uit

C = 0 : Cursor is onzichtbaar

B = 0 : Knipperend vlak is onzichtbaar

- Entry mode set

S = 0 & I/D = 1: de cursor verplaatst zich naar rechts, display blijft op de zelfde plaats

De overeenkomst tussen de het karakteradres en het karakterfiguur.

Low 4 bits	High 4 bits	0000	0001	0010	0011	0100	0101	0110	0111	1000	1001	1010	1011	1100	1101	1110	1111
xxxx0000	CG RAM (1)			0	@	P	`	P				-	9	3	α	p	
xxxx0001	(2)			!	1	A	Q	a	q			.	7	ç	ä	q	
xxxx0010	(3)			"	2	B	R	b	r			「	イ	ツ	×	β	θ
xxxx0011	(4)			#	3	C	S	c	s			」	ウ	テ	ε	ω	
xxxx0100	(5)			\$	4	D	T	d	t			、	エ	ト	†	μ	Ω
xxxx0101	(6)			%	5	E	U	e	u			・	オ	ナ	1	σ	Ü
xxxx0110	(7)			&	6	F	V	f	v			ヲ	カ	ニ	ヨ	ρ	Σ
xxxx0111	(8)			'	7	G	W	g	w			ヲ	キ	ヌ	ラ	g	π
xxxx1000	(1)			<	8	H	X	h	x			イ	ク	ネ	リ	フ	×
xxxx1001	(2)			>	9	I	Y	i	y			ウ	ケ	ル	ル	"	y
xxxx1010	(3)			*	:	J	Z	j	z			エ	コ	ン	レ	j	†
xxxx1011	(4)			+	;	K	[k	[オ	サ	ヒ	ロ	*	†
xxxx1100	(5)			,	<	L	¥	l	¥			オ	シ	フ	ワ	†	†
xxxx1101	(6)			-	=	M]	m]			ユ	ズ	ン	ト	†	†
xxxx1110	(7)			.	>	N	^	n	^			ヨ	セ	ホ	ト	†	†
xxxx1111	(8)			/	?	O	_	o	_			ウ	ツ	マ	ト	†	†

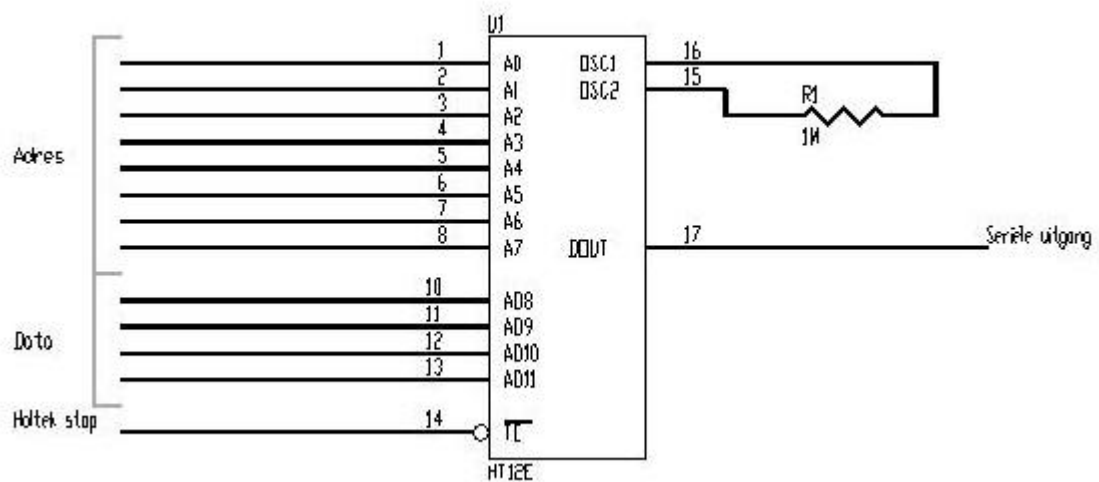
5.6 Holtek Encoder

5.6.1 Doel

De Holtek Encoder zet twaalf parallelle ingangen om naar één seriële uitgang. Deze uitgang wordt dan verder verbonden met een zendmodule die zet dan het digitale signaal om naar een draadloos hoogfrequent signaal (433.99MHz) .

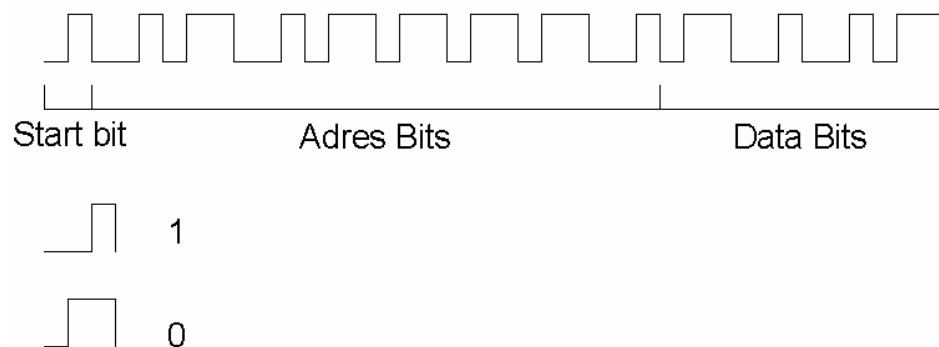
5.6.2 Schemavoorstelling

De Holtek bestaat uit acht adres ingangen A0...A7 (256 combinaties) en vier data-ingangen AD8...AD11.



5.6.3 Signaaluitgang van de Holtek Encoder

Deze Holtek zendt het adres en de data serieel naar zijn uitgang. Op de volgende figuur volgt het signaal dat op de uitgang tevoorschijn komt.

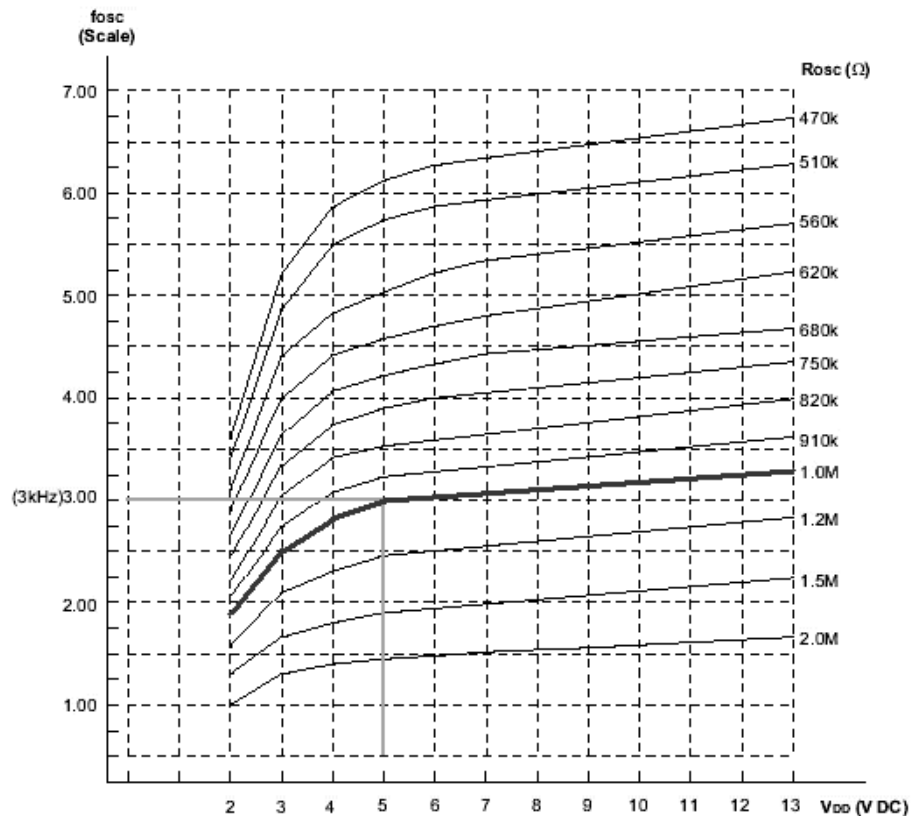


$$\text{Adres Bits} = (10000101)_L \rightarrow (133)_{10}$$

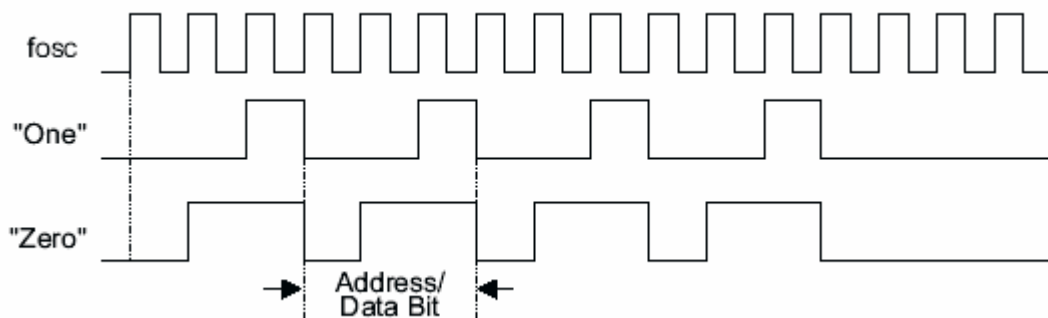
$$\text{Data Bits} = (0110)_L \rightarrow (6)_{10}$$

Dit signaal herhaalt zich ononderbroken, en kan enkel gestopt worden als de uitgang Y6 van de datadistributie 1 wordt. Deze uitgang is verbonden met de \overline{TE} ingang van de Holtek Encoder. Dus als de \overline{TE} ingang 1 wordt zal de transmissie stoppen.

5.6.4 Oscillator



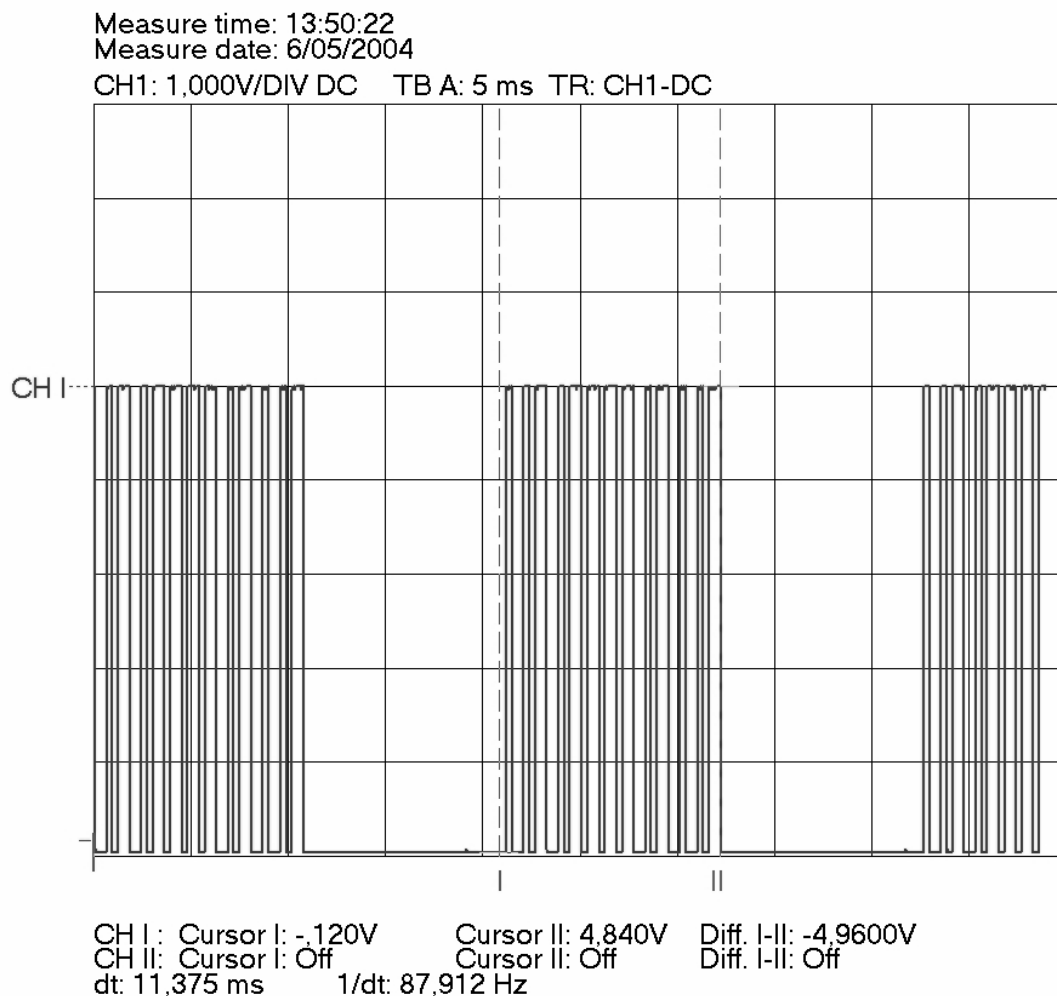
Het weerstandje dat aangesloten is op de Holtek Encoder, is om de zendsnelheid van de uitgang in te stellen. Hier bedraagt de weerstandswaarde 1M om een klokfrequentie te maken van 3 kHz (zie bovenstaande figuur).



Eén periode van het 1 oscillatie signaal duurt $T = \frac{1}{f} = \frac{1}{3kHz} = 333.33 \mu s$, maar men heeft

drie perioden nodig om 1 Bit te verzenden. Men zal dus 1 ms nodig hebben om 1 bit te verzenden. In het uitgangssignaal bevinden zich 12 bits (acht adresbits + vier databits) samen met startbit. Die 12 bits duren in totaal 12 ms, en de startbit duurt twee perioden van het oscillatie signaal dus 666.67 μs . Het volledige signaal duurt dus 12,67 ms.

5.6.5 Gemeten signaal:



We zien in de praktijk dat de totale tijd wat verschilt van de berekende tijd. Dit komt doordat de berekende tijd grafisch is opgelost. De voedingsspanning is niet precies 5V en de weerstand van 1M wijkt ook een beetje af. Het is tevens ook niet gemakkelijk om heel nauwkeurig te bepalen waar het signaal begint. Maar gelukkig wijkt het signaal niet overdreven veel af van het berekende. De Holtek Decoder ontvangt de signalen zonder problemen.

5.7 De Joystick

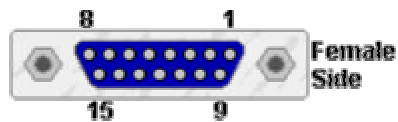
Enkele specificaties:

- 4 knoppen
- 2 potentiometers
- 4 zuignapjes voor stabiele bediening
- 15 pin's gamepoort connector



5.7.1 Klem aansluitingen

Om deze joystick te kunnen gebruiken moeten we eerst weten waarvoor elke pin van de connector dient. We meten de joystick uit terwijl we de knoppen bedienen en de assen laten roteren. We kwamen tot volgende tabel (in vet aangeduid = gebruikte klem):



Pinnummer	Functie	Pinnummer	Functie
1	+5V V DC	9	+5V DC
2	Joystick knop 1	10	Joystick knop 3
3	Joystick X-as	11	
4	Gnd (Ground)	12	Gnd
5	Gnd	13	
6	Joystick Y-as	14	Joystick knop 4
7	Joystick knop 2	15	+5V DC
8	+5V DC		

5.7.2 Knop overzicht

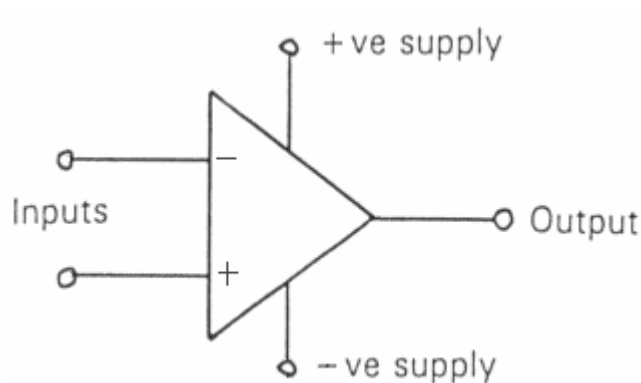


Nadat we nu weten hoe de joystick werkt, kunnen we verdergaan met de schakeling te ontwerpen. Eerst en vooral moeten we kunnen afleiden uit de waarde van de potentiometer van de joystick (X-as en Y-as) in welke positie de joystick zich bevindt. Dit kunnen we doen m.b.v. OPAMP's.

5.7.3 Wat is een OPAMP

Het woord OPAMP komt uit het Engels en gaat als volgt: “OPerational AMPlifier”. Dit betekend in het Nederlands “Operationele versterker”. Het deel operationeel wijst erop dat een OPAMP bewerkingen kan uitvoeren met de signalen die aangesloten worden. Hij kan bvb. twee signalen op - of aftellen.

Symbol:

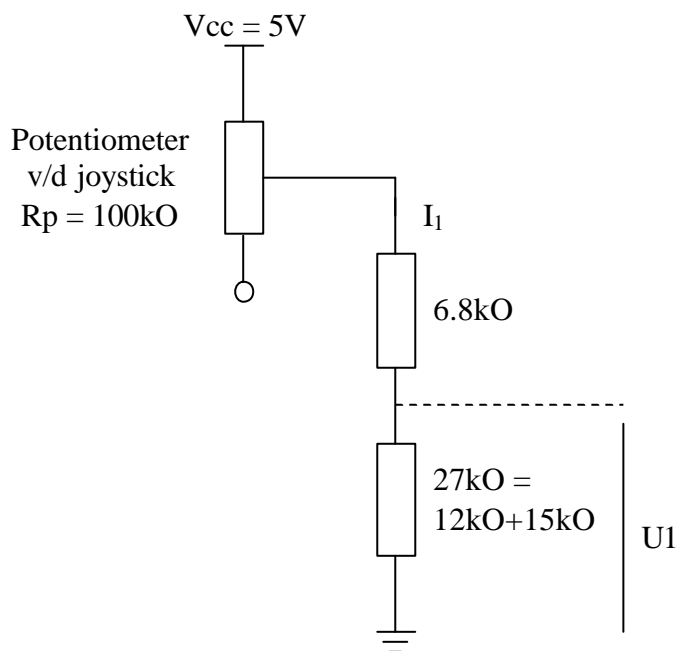


Als we nu 2 ingangen (spanningen) op de OPAMP aansluiten kunnen we deze waarden vergelijken.

De ene spanning is afkomstig van de spanning over de potentiometer (100k?) van de joystick. De andere spanning is een constante, op voorhand bepaalde grenswaarde (grensspanning). De OPAMP vergelijkt deze waarden. Is de spanning op de min-klem kleiner dan die op de plus-klem dan komt er op de uitgang een 1-sigitaal. Indien dit niet het geval is komt er een 0-sigitaal op de uitgang.

5.7.4 Bepalen van de joystickweerstand

Uit de datasheets weten we dat deingangsspanning van de OPAMP moet variëren tussen een spanning die iets groter is dan 0V en iets kleiner dan 5V. Als we de joystick rechtstreeks op de OPAMP zouden aansluiten zou de spanning variëren van 0V tot 5V. M.b.v. een spanningsdeler kunnen we deze spanningswaarden aanpassen zodat we de datasheet gegevens volgen (zie onderstaande uitwerking).



Als R potentiometer = 100k? dan

$$R_v = 133.8k\Omega$$

$$I_1 = \frac{V_{cc}}{R_v} = 37.37\mu A$$

$$U_1 = I_1 \cdot 27k\Omega = 1.009V$$

Als R potentiometer = 0k? dan

$$R_v = 33.8k\Omega$$

$$I_1 = \frac{V_{cc}}{R_v} = 147.93\mu A$$

$$U_1 = I_1 \cdot 27k\Omega = 3.994V$$

Als R potentiometer = 50k? dan

$$R_v = 83.8k\Omega$$

$$I_1 = \frac{V_{cc}}{R_v} = 59.67\mu A$$

$$U_1 = I_1 \cdot 27k\Omega = 1.611V$$

We doen dit zowel voor de X-as als voor de Y-as van de joystick.

5.7.5 Bepalen van de grenswaarden

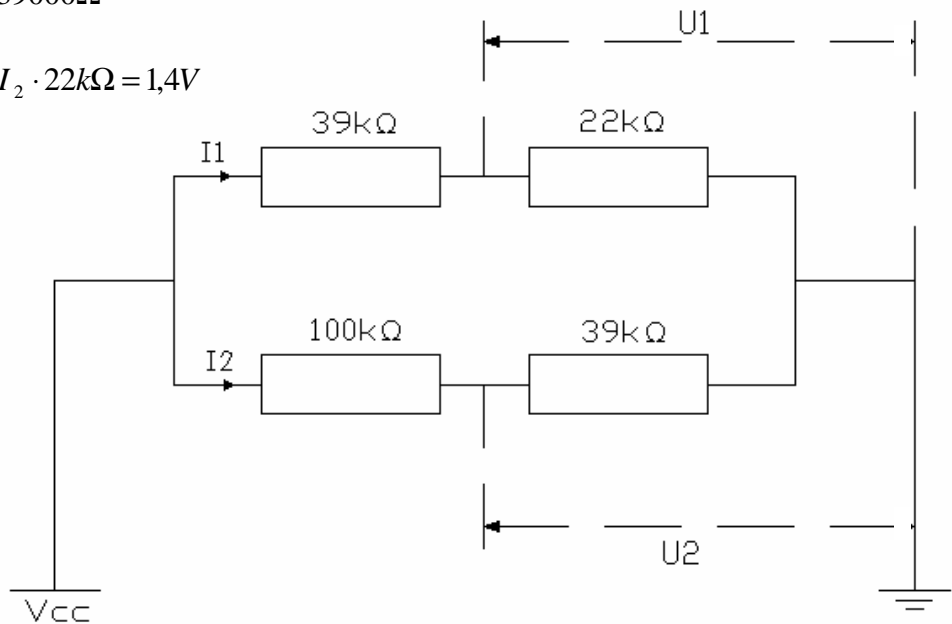
De waarde van de potentiometers bedraagt 100k?, maar is slechts een uitwijking mogelijk van $\pm 20k?$ tot 80k?. De grenzen worden dus op 40k? en 60k? genomen. Net zoals bij de potentiometers van de joystick moeten de ingangsspanningen een aantal volts boven 0V en onder 5V blijven. Dit gebeurt m.b.v. vier “spanningsdelers” (2 weerstanden). Twee voor de vergelijkingen met de X-as en twee voor de Y-as van de joystick. Hieruit volgen vier grensspanningen. Naargelang de spanning over de potentiometer van de joystick dan groter of kleiner is dan één van de 4 grensspanningen, weten we of de joystick links, recht, naar voor, naar achter, rechts voor, links achter ... gekanteld is.

$$I_1 = \frac{V_{cc}}{61k\Omega} = \frac{5V}{61000\Omega} = 81,97 \text{ mA}$$

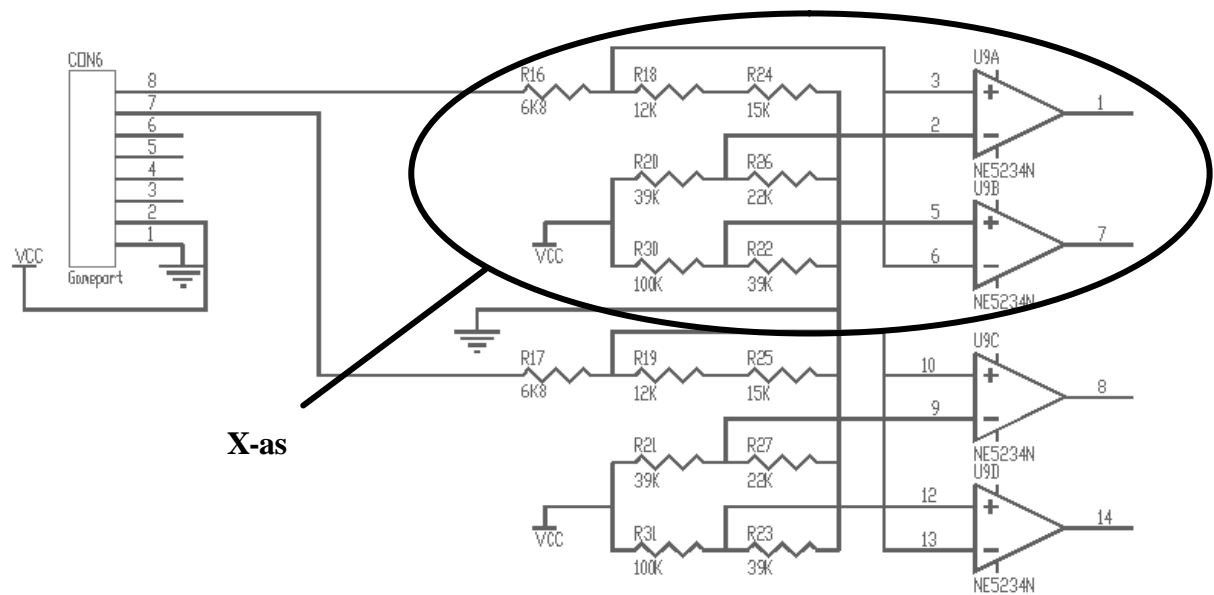
$$\Rightarrow U_1 = I_1 \cdot 22k\Omega = 1,8V$$

$$I_2 = \frac{V_{cc}}{139k\Omega} = \frac{5V}{139000\Omega} = 35,97 \text{ mA}$$

$$\Rightarrow U_2 = I_2 \cdot 22k\Omega = 1,4V$$



Alles samen wordt dit dan:



Aangezien beide situaties precies hetzelfde inhouden, bespreken we alleen de X-as (zie bepalen van de joystickweerstand en bovenstaande figuur).

Als we de joystick volledig doen uitwijken naar rechts ($R_p=0k\Omega$) dan krijgen we als ingangsspanning van de 2 OPAMP's (klem 3 en 6) een spanning van 3,994V. Op klem 2 komt een grensspanning van 1,8V en op klem 3 een spanning van 1,4V.

Bij de bovenste OPAMP:

Klem 3 (3,994V) is groter dan klem 2 (1,8V) dus uitgang (1) = 1

Bij de tweede OPAMP:

Klem 5 (1,4V) is kleiner dan klem 6 (3,994V) dus uitgang (7) = 0

Wijken we uit naar links ($R_p=100k\Omega$) dan komen op de klemmen 3 en 6 een spanning van 1,009V. Op de klem 2 en 5 nog steeds respectievelijk 1,8V en 1,4V.

Bij de bovenste OPAMP:

Klem 3 (1,009V) is kleiner dan klem 2 (1,8V) dus uitgang (1) = 0

Bij de tweede OPAMP:

Klem 5 (1,4V) is groter dan klem 6 (1,009V) dus uitgang (7) = 1

Blijven we in het midden (neutraal) dan is de potentiometer (X-as) van de joystick 50k?. De spanning op de klemmen 3 en 6 is dus 1,611V en op de klemmen 2 en 5 respectievelijk 1,8V en 1,4V.

Bij de bovenste OPAMP:

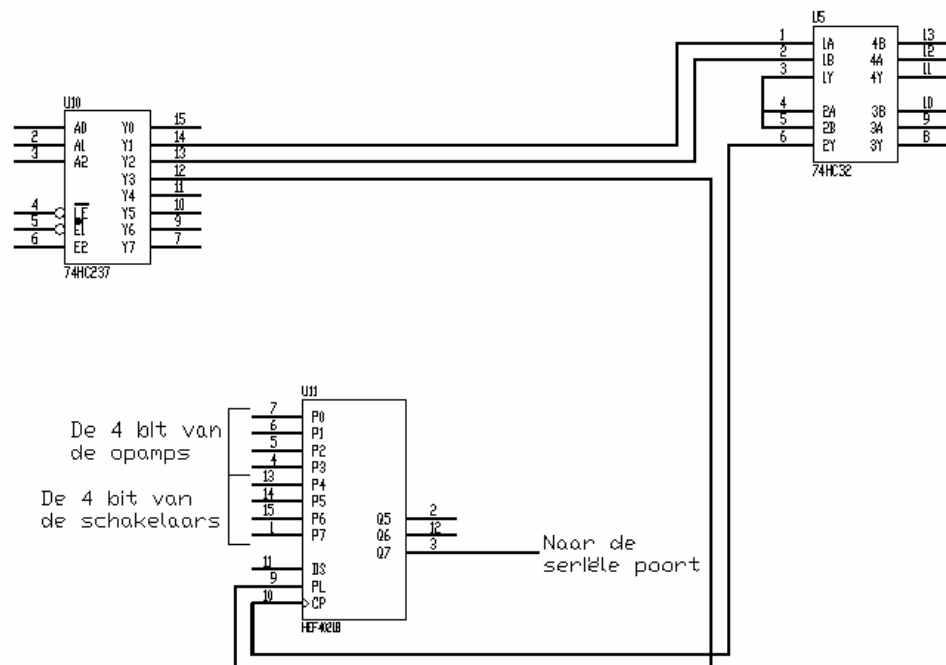
Klem 3 (1,611V) is kleiner dan klem 2 (1,8V) dus uitgang (1) = 0

Bij de tweede OPAMP:

Klem 5 (1,4V) is kleiner dan klem 6 (1,611V) dus uitgang (7) = 0

5.7.6 Gegevens serieel maken

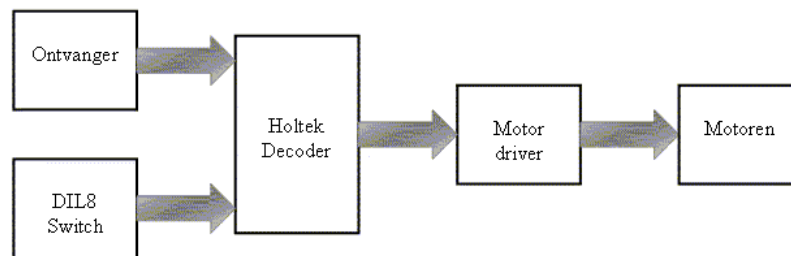
Met de vier uitgangen van de Opamps en de vier schakelaars van de joystick bekomen we een parallel signaal van acht bit. Dit zetten we met een schuifregister om in een serieel signaal. M.b.v. de datadistributie kunnen we het schuifregister aansturen. Door een puls met uitgang Y3 van de multiplexer/decoder op ingang PL van het schuifregister te geven wordt het parallelle signaal opgeslagen. Vervolgens kan dit signaal terug uitgelezen worden, via de uitgang Q7, door klokpulsen te geven op de ingang CP. Na acht pulsen is het volledige signaal serieel in de computer ingelezen.



Via de software die speciaal geschreven werd voor de joystick wordt het signaal ontleden (waar bevinden zich de 1- of 0-signalen) en wordt de positie van de joystick bepaald.

6 Opbouw Racewagen

6.1 Blokschema



Figuur 8: Blokschema Racewagen

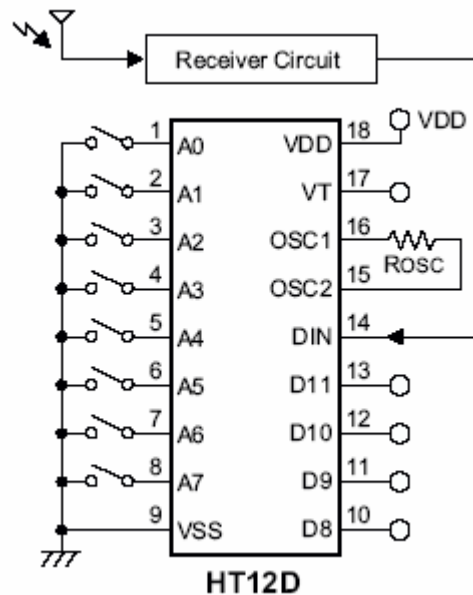
De opbouw van de racewagen bezit verschillende delen, namelijk een ontvanger, DIL switch, de Holtek Decoder, de motordriver en natuurlijk de motoren.

Het doel van de ontvangmodule is het ontvangen van het datasignaal om de motoren aan te sturen. Met de DIL switch is het mogelijk een vast adres te geven aan de racewagen, omdat men met de computer 256 verschillende toepassingen zal kunnen sturen. Daarna vergelijkt de Holtek Decoder het adres uit het ontvangen datasignaal, zodat enkel de wagen met het correcte adres zal gestuurd worden. Als de beide adressen overeenstemmen met elkaar worden de motoren aangestuurd.

Om de draaizin van de motoren te wijzigen maken we gebruik van een motor-driver.

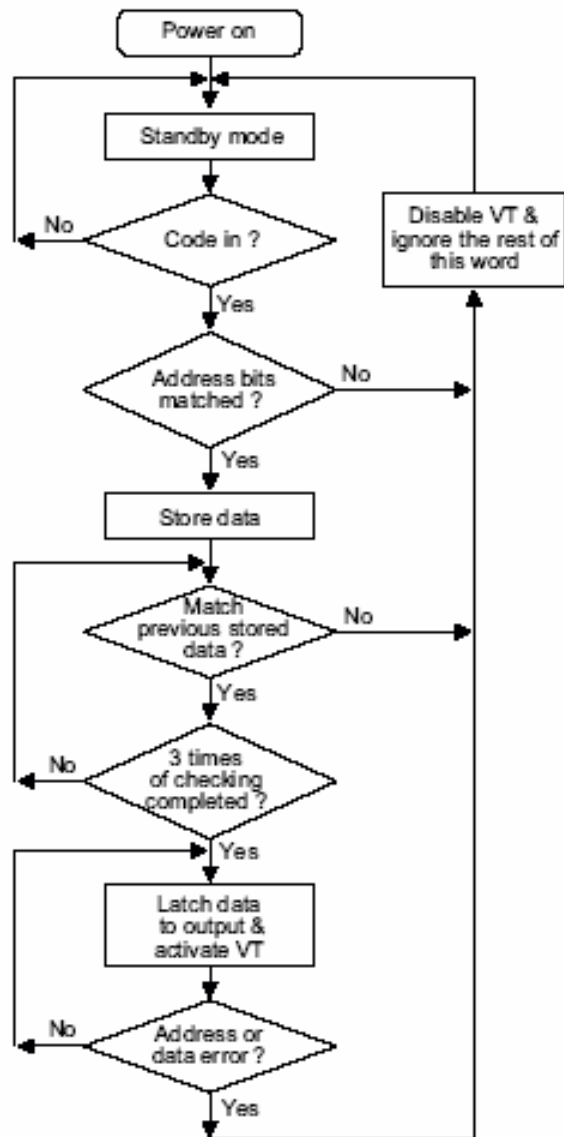
6.3 Holtek Decoder

6.3.1 Doel



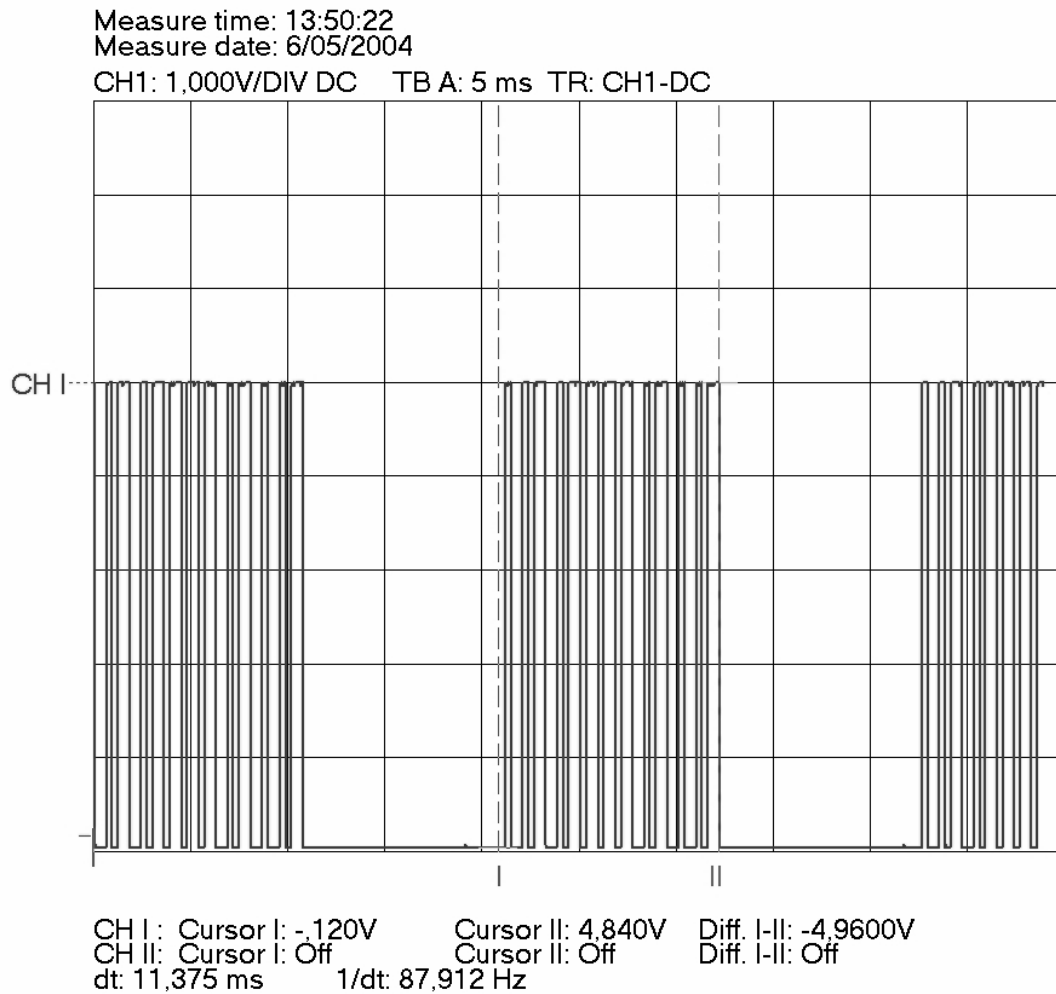
Het doel van de Holtek HT12D is het uitgezonden signaal van de HT12E (Encoder) terug om te zetten in bruikbare informatie, namelijk de data. Dit signaal komt binnen op klem 14. Vooraleer de data op de uitgangen D8-D11 komt moeten een aantal stappen doorlopen worden. Het adres, dat in het ontvangen signaal verwerkt zat, wordt vergeleken met het adres dat ingesteld is op de ingangen A0-A7. Dit adres stellen we in m.b.v. acht schakelaars (DIL-switch). Indien deze niet overeenkomen wordt de rest van het signaal genegeerd en wordt een nieuw signaal ingelezen. Dit signaal wordt dan opnieuw vergeleken enz.... Komen deze adressen overeen dan wordt de data opgeslagen. Deze data wordt dan vervolgens ook vergeleken met de data die ook in het signaal verwerkt zat. Komt de data overeen dan wordt er voor de zekerheid nog drie maal vergeleken vooraleer de data op de uitgangen D8-D11 komt. Komen deze ook niet overeen dan gebeurt net hetzelfde als bij het adres. Een nieuw signaal wordt ingelezen, opnieuw vergeleken... (zie volgend schema).

6.3.2 Schema van bewerkingen



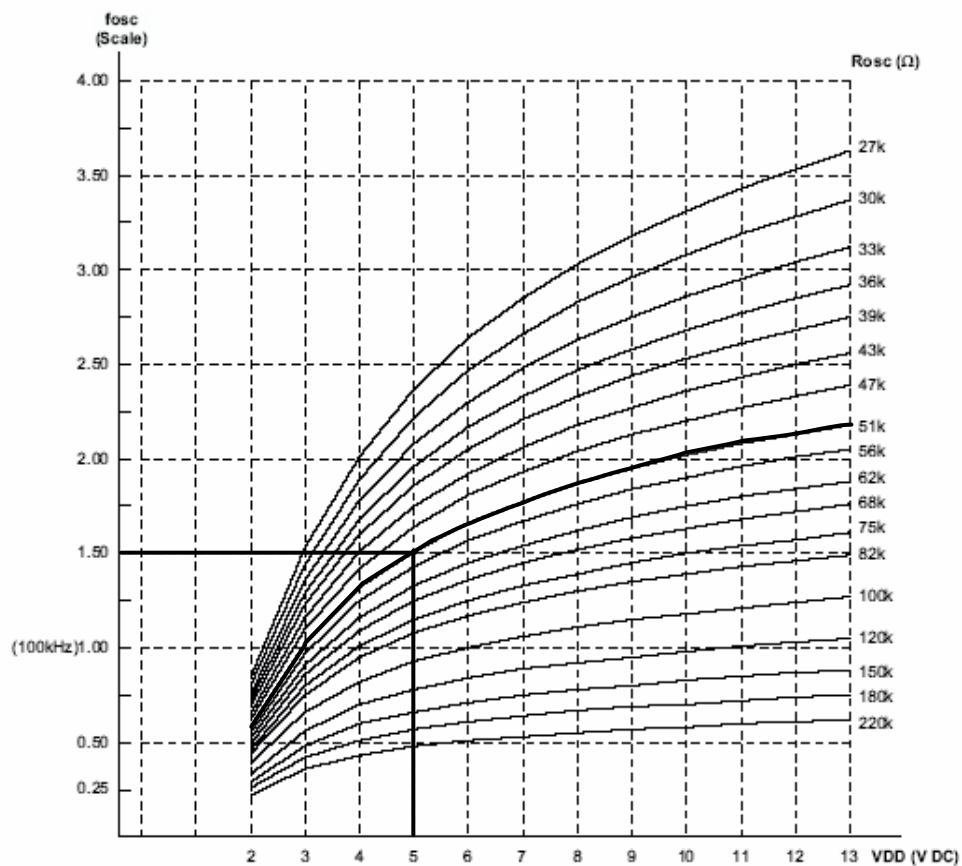
6.3.3 Het gemeten signaal

Op onderstaande figuur kun je het ingelezen signaal zien. Het is hetzelfde signaal dat de HT12E (Encoder) heeft gevormd. Voor meer info over dit signaal is het dan ook aangeraden om even terug te kijken naar het hoofdstuk over de HT12E

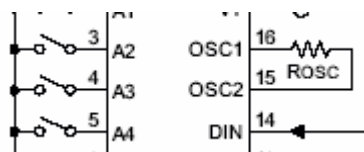


6.3.4 Oscillator

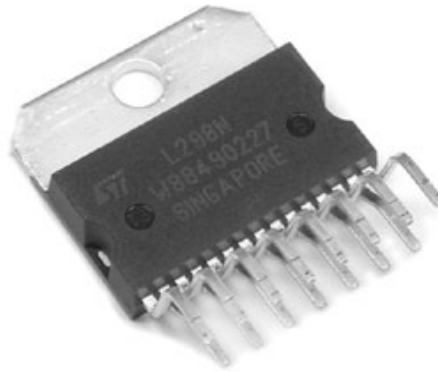
Uit de datasheet van de HT12D volgt dan de aangeraden frequentie van de oscillator ($f_{OSC D}$) 50 keer groter moet zijn dan de oscillatorfrequentie van de HT12E (f_{OSCE}). Aanzien deze frequentie ingesteld is op 3 kHz (Zie Holtek Encoder) moet $f_{OSC D}$ gelijk zijn aan: $50 \times 3 \text{ kHz} = 150 \text{ kHz}$. Deze frequentie kunnen we instellen door de oscillatorweerstand (R_{OSC}) te kiezen a.d.h.v. onderstaande grafiek. Meer uitleg vindt men in het hoofdstuk van de Holtek Encoder.



We weten dat de voedingsspanning 5V bedraagt. De aangeraden f_{SCD} is 150 kHz dus volgt uit de grafiek een R_{OSC} van 51 k? .

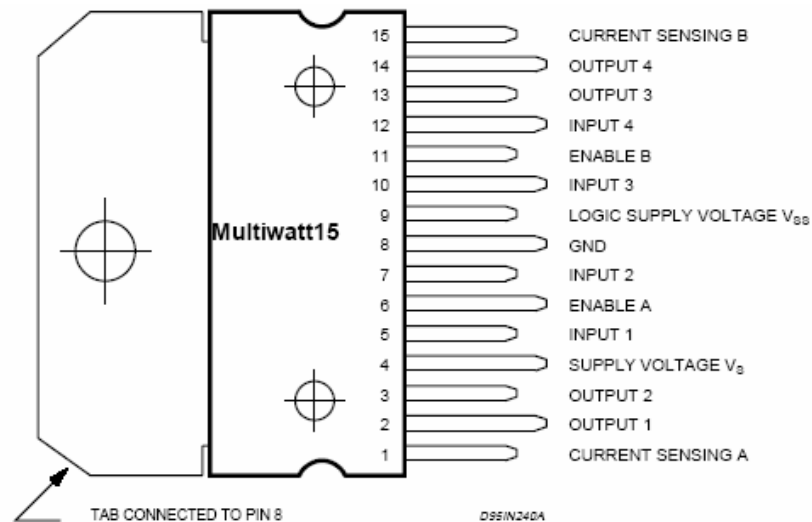


6.4 Aansturen motoren met L298N



De L298N is een IC in een 15 pin verticale Multiwatt verpakking. Het is een hoge spanning, hoge stroom tweevoudige volle brug driver die ontwikkeld is om standaard TTL logische niveaus (0 of 1) te ontvangen en inductieve belastingen zoals relais, spoelen, DC-motoren en stappenmotoren te sturen. Hij is voorzien van een inwendige hitte beschermingscircuit die de chip uitschakelt als die te warm wordt. De L298N kan eveneens op een koelplaat bevestigd worden.

6.4.1 Werking L298N motor driver



Zoals uit de naam van het component volgt zijn er twee “Bruggen aanwezig in de L298N. Indien het woord brug je niets zegt dan wordt dit in onderstaande toestanden wel duidelijk. Elke “Brug” van de motordriver bestaat uit 2 delen. Elk deel bestaat namelijk uit twee ingangen, twee uitgangen, één enable ingang en één current sensing ingang.

We hebben bijgevolg volgende klemmen op het IC:

Klemmen	Brug 1	Brug 2
2 ingangen	Input 1 en 2	Input 3 en 4
2 uitgangen	Output 1 en 2	Output 3 en 4
Enable	Enable A	Enable B
Current sensing	CS A	CS B
Logic supply voltage Vss	Gemeenschappelijke '+ klem' voor interne schakeling	
Supply voltage Vs	Gemeenschappelijke '+ klem' voor motoren	
GND	Gemeenschappelijke '- klem' voor interne schakeling	

De "Logic supply voltage"-klem en Gnd-klem voorziet de inwendige schakeling van voeding.

De "Supply voltage"-klem en Gnd-klem zal de motoren van spanning voorzien.

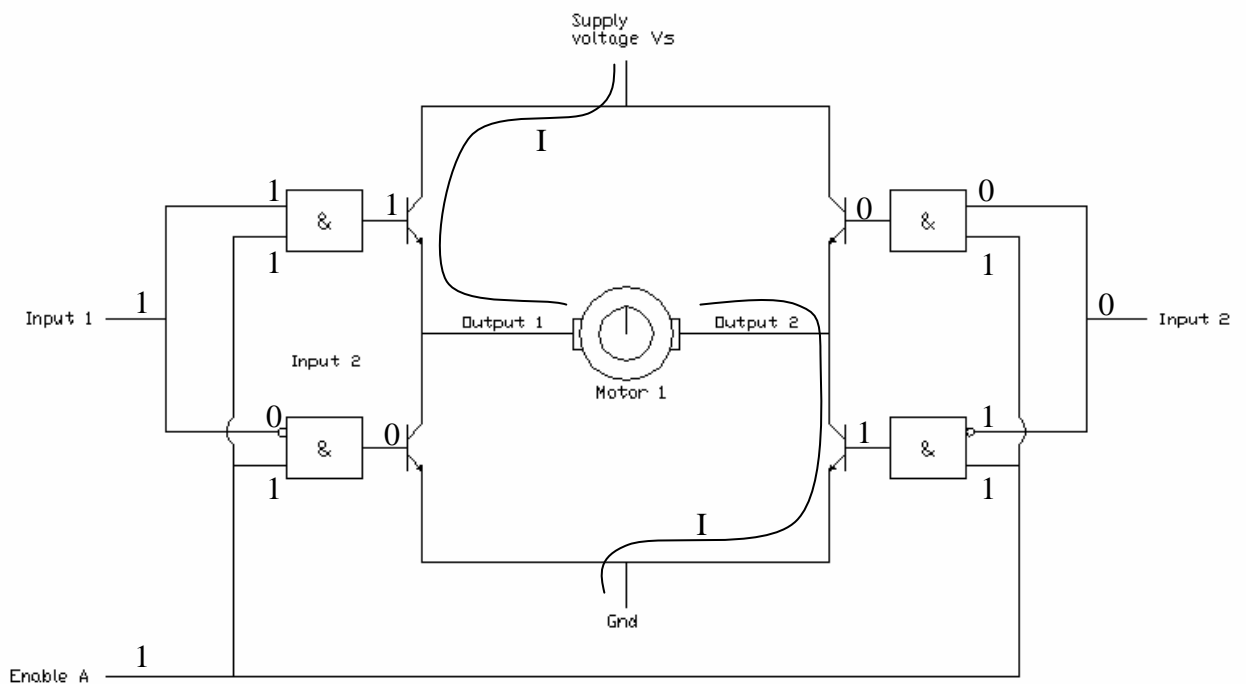
Wanneer de Enable-ingang van de brug hoog is kan d.m.v. een combinatie van de ingangen een draaizin voor de motoren gevormd worden. Hieruit volgen vier verschillen toestanden die we met een schema in beeld kunnen brengen.

Hierbij zullen de transistoren werken als schakelaar (zie bijlage). Wanneer op de basis van de transistor een hoog komt zal deze zich gedragen als een gesloten schakelaar.

Anderzijds, bij een laag op de basis, zal de transistor werken als open schakelaar.

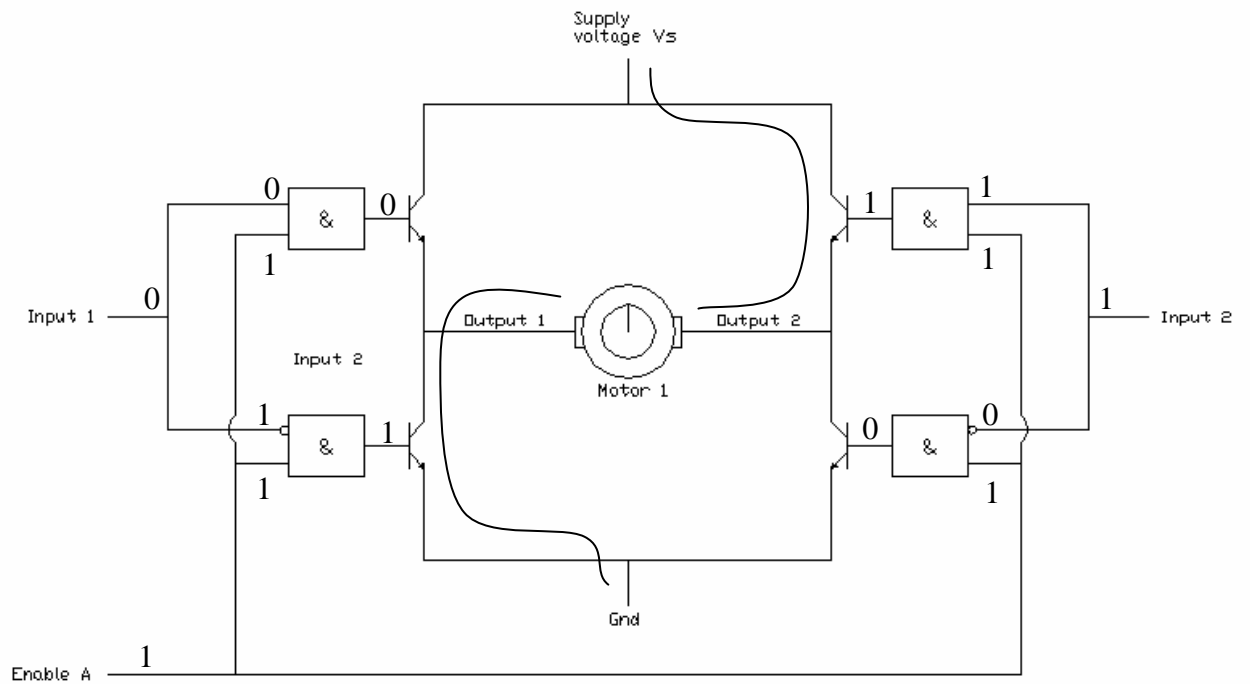
6.4.2 Toestand 1

Als we op Input 1 hoog maken en input 2 laag dan kunnen we met behulp van onderstaande figuur de stroomzin door de transistors volgen. Afhankelijk van de aansluitklemmen op de motor zal de motor dan links of rechts draaien.



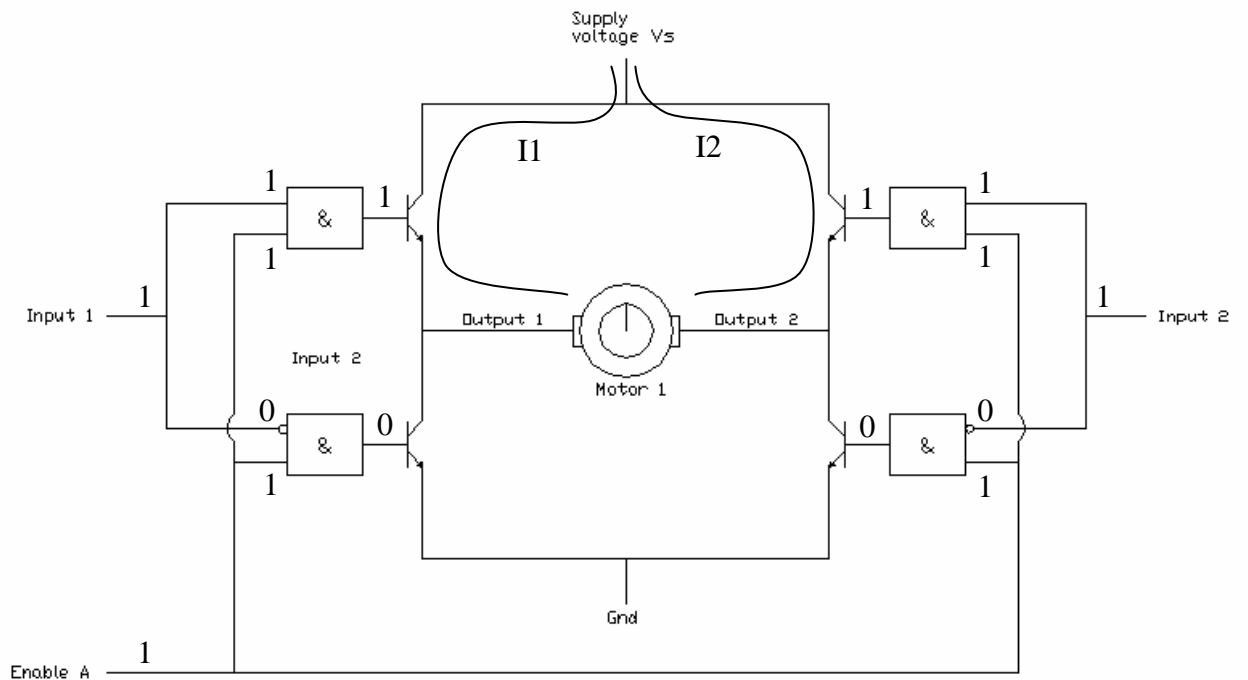
6.4.3 Toestand 2

Maken we nu Input 2 hoog en input 1 laag dan verandert de stroomzin door de motor en draait hij dan ook in de tegenovergestelde richting.



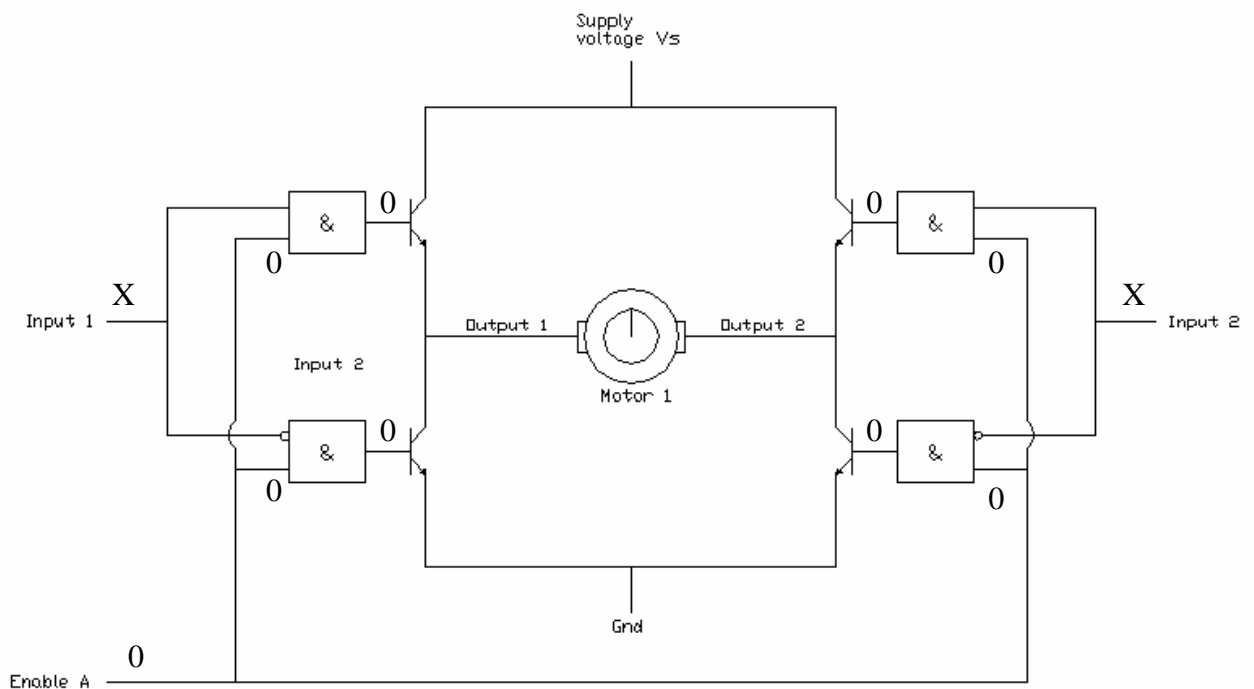
6.4.4 Toestand 3

Zijn beide Input's hoog of beide laag dan krijgen we een combinatie van toestand 1 en toestand 2 zodat de motor als het ware geblokkeerd wordt. Dit systeem kunnen we gebruiken om te remmen. Op onderstaand schema zijn beide ingangen 1.



6.4.5 Toestand 4

Om de wagen te laten uitbollen bekommt men dit door geen stroom te laten vloeien door de motor. We zorgen dan dat de ingang Enable laag is. Zo hebben de Input's geen invloed op de Output. (zie onderstaande figuur)



X = don't care

Onze pc-gestuurde wagen heeft twee motoren, dus kunnen we beide bruggen van de L298N motor driver gebruiken. Met de verschillende draaizinnen van deze twee motoren kunnen we de wagen alle mogelijke bewegingen laten uitvoeren. Draaien om zijn as (links en rechts), Vooruit (beide motoren draaien voorwaarts) en achteruit (beiden motoren achterwaarts).

7 Werking van de Software

7.1 Waar is software voor nodig?

Taken van de software:

- Een eerste taak bestaat erin gegevens in te voeren, dat wil zeggen dat de software u in staat moet stellen om invloed uit te oefenen op het gedrag van de schakeling. Men geeft met de muis of het toetsenbord informatie aan de software door. Deze informatie wordt vervolgens weer via de poorten aan de schakeling doorgegeven.
- Een tweede taak bestaat uit het zodanig voorbereiden van de over te brengen informatie, dat deze informatie over de poorten verstuurd kan worden en door de schakeling verwerkt kan worden. De software voert daartoe omrekeningen, conversies en andere handelingen uit.
- Een derde taak bestaat erin de poorten zo aan te spreken, dat er in ieder geval gegevensuitwisseling kan plaatsvinden. Dit is absoluut noodzakelijk, ongeacht of de gegevensstroom vanaf, of naar de computer loopt. De software handelt overigens nog een andere taak af: wordt de schakeling ervoor gebruikt om gegevens te verzamelen die vervolgens aan de computer doorgegeven worden, dan moet de software deze gegevens inlezen en zodanig bewerken, dat het resultaat bijvoorbeeld op het scherm zichtbaar wordt.
- Zo nodig verwerkt de software de gegevens om vervolgens passend te reageren. Dit kan bijvoorbeeld er zo uitzien, dat er gegevens door de schakeling verzameld en via de poorten aan de software doorgegeven worden. Deze gegevens worden vervolgens verwerkt en aan de hand van het resultaat kan dan bijvoorbeeld weer informatie teruggestuurd worden naar de schakeling. Op deze manier ontstaat een zogenaamde regelkring.

Zoals u reeds kunt zien kan de schakeling niet functioneren zonder programma. De software is met diverse belangrijke taken belast. Zonder deze taken kunnen de schakelingen niet gestuurd worden.

7.2 De programmeertaal Visual Basic.

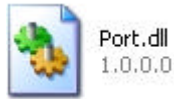
De software wordt met behulp van een zogenaamde programmeertaal vervaardigd. Met zo'n programmeertaal kan een programmeur de computer meedelen welke handelingen hij moet verrichten of welke berekeningen hij moet uitvoeren. Zijn nu alle gewenste functies geprogrammeerd, dan genereert de programmeertaal een zelfstandige toepassing, die precies die functies bevat die zojuist zijn geprogrammeerd.

Bij het programmeren van software komt het erop neer de zogenaamde broncode bestanden te vervaardigen. In deze bestanden bevinden zich de opdrachten die de software uit moet voeren. Voor iedere schakeling is een aparte toepassing geprogrammeerd, wat betekend dat voor elke toepassing een bijhorend broncode bestand moet gecreëerd worden.

Voor het gebruik van de poorten heb je in de bronbestanden een speciale DLL-file nodig. (Zie volgend hoofdstuk) DLL staat voor Dynamic Link Library. Wij gebruiken de port.dll varianten, de 16-bits portio.dll en de 32-bit versie portio32.dll. Om deze DLL's te gebruiken moet je ze eerst kopiëren naar de subdirectory system32 van Windows plaatsen.

7.3 Bespreking van het bestand Port.dll

Onder Windows vormt de toegang tot de poorten van de PC het grote probleem. Daarom maken we hier gebruik van een universeel bruikbare DLL (Dynamic Link Library), een bibliotheek met Windows-functies. Deze worden geschreven in Delphi.



Port.dll vervult onderstaande taken:

- openen van interfaces;
- seriële dataoverdracht;
- toegang tot poortlijnen;
- algemeen: in - en uitvoer van data via poorten;
- nauwkeurige milliseconde-pulsgever;
- pulsgever in microseconde-raster;
- toegang tot de geluidskaart en
- invoer via joystick.

De DLL kan onder verschillende programmeersystemen worden gebruikt.

De ontwikkeling van een algemene DLL voor de toegang tot willekeurige hardware staat in zekere zin haaks op de Windows-filosofie om alle toegang tot de hardware via drivers te laten verlopen. Een driver hoort altijd bij een specifiek apparaat. Voor kleine hardware-experimenten is daar eigenlijk geen plaats meer voor. De “correcte” ontwikkeling van een driver is echter een ingewikkeld proces dat eigenlijk uitsluitend aan grote bedrijven is voorbehouden.

Onder DOS waren er in elke programmeertaal poortinstructies voorhanden. Via deze instructies hadden we rechtstreeks toegang tot alle hardwarecomponenten. Onder Windows 3.1 werd dat al minder maar de beperkingen bleven miniem. Onder Windows 95 is dit alles nog een stuk moeilijker geworden. Er bestaat geen rechtstreekse mogelijkheid meer om toegang tot poorten te krijgen. Daarvoor werd de DLL gebruikt om vanuit Visual Basic weer toegang tot de poortadressen te krijgen.

7.4 Samenstelling van het volledige programma

7.4.1 CGRPrintplaat.dll



Dit programma is gemaakt om de volledige schakeling te bedienen via de onderstaande woorden.

- COM_Start
- Com_Restart
- PositionX
- PositionY
- Button 1
- Button 2
- Button 3
- Button 4
- Restart
- JoystickRead
- Holtek12Encoder
- MotorStop
- TransmitterSTOP
- Display_Clear
- Display_ReturnHome
- Display_EntryModeSet
- Display_ON/OFF
- Display_Shift
- Display_SetFunction
- Display_SetCgRamAddress
- Display_SetDdRamAddress
- Display_SetWriteData
- Display_Tekst
- Translatie
- Rotatie

```

Option Explicit
Private Declare Function OPENCOM Lib "port.dll" (ByVal A$) As Integer
Private Declare Sub DTR Lib "port.dll" (ByVal B%)
Private Declare Sub RTS Lib "port.dll" (ByVal B%)
Private Declare Sub TXD Lib "port.dll" (ByVal B%)
Private Declare Sub DELAY Lib "port.dll" (ByVal ms As Integer)
Private Declare Function CTS Lib "port.dll" () As Integer
Private Declare Sub CLOSECOM Lib "port.dll" ()
Private Declare Sub DELAYUS Lib "port.dll" (ByVal L As Long)
Private Check As Boolean
Private ComNumber As Byte
Dim JoystickValue As Joystick
Private Type Joystick
    PosX As Integer
    PosY As Integer
    B1 As Boolean
    B2 As Boolean
    B3 As Boolean
    B4 As Boolean
End Type

Public Sub COM_Start(Number As Byte)
    ComNumber = Number
    OPENCOM ("COM" & ComNumber & ",9600,N,8,1")
    TXD 0
    DTR 0
    RTS 0
End Sub

Public Sub Com_Restart()
    OPENCOM ("COM" & ComNumber & ",9600,N,8,1")
End Sub

Public Function PositionX()
    PositionX = JoystickValue.PosX
End Function

Public Function PositionY()
    PositionY = JoystickValue.PosY
End Function

Public Function Button1()
    Button1 = JoystickValue.B1
End Function

Public Function Button2()
    Button2 = JoystickValue.B2
End Function

Public Function Button3()
    Button3 = JoystickValue.B3
End Function

Public Function Button4()
    Button4 = JoystickValue.B4
End Function

Private Sub Distribution(Channel As Byte)
    Dim TellerI As Byte
    RTS 0
    For TellerI = 0 To 2
        If (4 / (2 ^ TellerI) And Channel) > 0 Then
            TXD 1
        Else
            TXD 0
        End If
        DTR 1
        DTR 0
    Next TellerI
    RTS 1
    RTS 0
End Sub

```

```

Public Sub Restart()
Dim Teller As Byte
For Teller = 0 To 8
    Distribution 1
    Distribution 0
Next Teller
    Distribution 4
    Distribution 5
    Distribution 0
End Sub

```

```

Public Sub JoystickRead()
Dim Teller As Byte
    JoystickValue.PosX = 0
    JoystickValue.PosY = 0
    JoystickValue.B1 = False
    JoystickValue.B2 = False
    JoystickValue.B3 = False
    JoystickValue.B4 = False
Distribution 3
inlezen
Distribution 0
For Teller = 0 To 7
    If CTS = 1 Then
        Select Case Teller
            Case 7
                JoystickValue.PosX = -1
            Case 6
                JoystickValue.PosX = 1
            Case 5
                JoystickValue.PosY = -1
            Case 4
                JoystickValue.PosY = 1
            Case 3
                JoystickValue.B1 = True
            Case 2
                JoystickValue.B2 = True
            Case 1
                JoystickValue.B3 = True
            Case 0
                JoystickValue.B4 = True
        End Select
        End If
        Distribution 1
        Distribution 0
    Next Teller
End Sub

```

```

Public Sub Holtek12Encoder(ByVal Waarde As Byte)
Dim Teller As Byte
For Teller = 0 To 7
    If (128 / (2 ^ Teller) And Waarde) > 0 Then
        Distribution 2
    Else
        Distribution 1
    End If
    Distribution 0
Next Teller
Distribution 4
Distribution 0
End Sub

```

```

Private Sub CarData(Waarde As Byte)
Dim Teller As Byte
For Teller = 0 To 3
    If (8 / (2 ^ Teller) And Waarde) > 0 Then
        Distribution 2
    Else
        Distribution 1
    End If
    Distribution 0
Next Teller
Distribution 5
Distribution 0
End Sub

```

```

Public Sub MotorStop()
    CarData 0
End Sub
' de code MotorStop
' verwijzen naar code

Public Sub TransmitterStop()
    Distribution 6
End Sub
' de code TransmitterStop
' TE ingang van de Holtek encoder op 1 plaatsen

Private Sub DisplayValue(Waarde As Integer)
    Dim Teller As Byte
    For Teller = 0 To 8
        If (256 / (2 ^ Teller) And Waarde) > 0 Then
            Distribution 2
        Else
            Distribution 1
        End If
        Distribution 0
    Next Teller
    Distribution 7
    plaatsen
    DELAY 2
    Distribution 0
End Sub
' display datavorming
' Teller telt van 0 tot en met 8
' kijken of 256 .. 1 in Waarde voorkomen (binair maken)
' een 1 in alle schuifregisters plaatsen
' zo niet
' een 0 in alle schuifregisters plaatsen
' alle uitgangen van datadistributie op nul plaatsen
' Teller verhoogt met 1
' STR ingang van de display schuifregister op 1
' pauze van 2 ms
' alle uitgangen van datadistributie op nul plaatsen

Public Sub Display_Clear()
    DisplayValue 1
End Sub

Public Sub Display_ReturnHome()
    DisplayValue 2
End Sub

Public Sub Display_EntryModeSet(I_D As Boolean, SH As Boolean)
    Dim EntryModeSet As Integer
    EntryModeSet = 4 + Abs(I_D) * 2 + Abs(SH) * 1
    DisplayValue EntryModeSet
End Sub

Public Sub Display_ON_OFF(D As Boolean, C As Boolean, B As Boolean)
    Dim ON_OFF As Integer
    ON_OFF = 8 + Abs(D) * 4 + Abs(C) * 2 + Abs(B) * 1
    DisplayValue ON_OFF
End Sub

Public Sub Display_Shift(SC As Boolean, R_L As Boolean)
    Dim Shift As Integer
    Shift = 16 + Abs(SC) * 8 + Abs(R_L) * 4
    DisplayValue Shift
End Sub

Public Sub Display_SetFunction(DL As Boolean, N As Boolean, F As Boolean)
    Dim SetFunction As Integer
    SetFunction = 32 + Abs(DL) * 16 + Abs(N) * 8 + Abs(F) * 4
    DisplayValue SetFunction
End Sub

Public Sub Display_SetCgRamAddress(CgRamAddress As Byte)
    Dim SetCgRamAddress As Integer
    If CgRamAddress > 63 Then Exit Sub
    SetCgRamAddress = 64 + Val(CgRamAddress)
    DisplayValue SetCgRamAddress
End Sub

Public Sub Display_SetDdRamAddress(DdRamAddress As Byte)
    Dim SetDdRamAddress As Integer
    If DdRamAddress > 127 Then Exit Sub
    SetDdRamAddress = 128 + Val(DdRamAddress)
    DisplayValue SetDdRamAddress
End Sub

```

```
Public Sub Display_SetWriteData(WriteData As Byte)
    Dim SetWriteData As Integer
    If WriteData > 255 Then Exit Sub
    SetWriteData = 256 + Val(WriteData)
    DisplayValue SetWriteData
End Sub
```

```
Public Sub Display_Tekst(Letter As String)
    Display_SetWriteData Asc(Letter)
End Sub
```

```
Private Sub Class_Terminate()
    CLOSECOM
End Sub
```

'bij het afsluiten van de DLL
'Com poort sluiten

```
Public Sub Translatie(Richting As Boolean)
    If Richting = True Then
        CarData 5
    Else
        CarData 10
    End If
End Sub
```

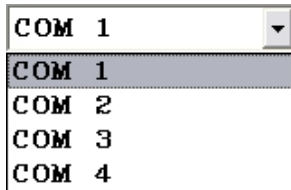
'translatie richting uitvoeren
'als richting = true dan
'vooruit rijden
'als richting = false dan
'achteruit rijden

```
Public Sub Rotatie(Richting As Boolean)
    If Richting = True Then
        CarData 6
    Else
        CarData 9
    End If
End Sub
```

'draai richting uitvoeren
'als richting = true dan
'links draaien
'als richting = false dan
'rechts draaien

7.4.2 frmMain

```
Option Explicit
Private DisplayList As Boolean
Private Punt As Coördinaat
Private Opslaan As Boolean
Private RecordLengte As Integer
Private AantalRecords As Single
Private Richting As Single, Tijd As Single
Private BestandsPad As String
Private BestandsNaam As String
Private TellerP As Integer
Private MaxP As Integer
Private HT As Byte
Private tT As Single
```



```
Private Sub Combol_Change()
CGR.COM_Start (Combol.ListIndex + 1)
Timer2.Interval = COMTIME
Timer2.Enabled = True
CGR.Display_Clear
CGR.Display_ReturnHome
CGR.Display_SetFunction True, True, False
CGR.Display_EntryModeSet True, False
CGR.Display_ON_OFF True, False, False
Tekst1 " Ruben D'Hulst 7"
Tekst2 " Yves Yde 17"
End Sub
```

'kiezen van COM poort
'gekozen poort wordt geopend
'interval tijd van timer2 instellen
'timer2 starten
'Display wissen
'cursor op begin positie plaatsen
'8BitInterface,2Lines,5x7dots
'the cursor moves to the right
'Display ON,Cursor OFF,Blink OFF
'verwijzen naar code Tekst1
'verwijzen naar code Tekst2



```
Private Sub cmdUp_Click()
Dim Inhoud As String
Dim IndexC As Single
DisplayList = False
IndexC = ProgList.ListIndex
If Not IndexC = -1 And Not IndexC = 0 Then
    Inhoud = ProgList.Text
    ProgList.RemoveItem IndexC
    ProgList.AddItem Inhoud, IndexC - 1
    ProgList.ListIndex = IndexC - 1
End If
DisplayList = True
Opslaan = True
End Sub
```

'bij het klikken op de knop omhoog
'displayverandering afleggen
'de plaats van het geselecteerde item wordt opgeslagen
'er moet een item geselecteerd zijn en
'de eerste plaats mag niet geselecteerd zijn
'de inhoud van de geselecteerde item wordt opgeslagen
'het geselecteerde item verwijderen
'het terug invoegen maar 1 plaatsje hoger
'het ingevoegde selecteren
'displayverandering aanleggen
'op het einde vragen om op te slaan



```
Private Sub cmdDown_Click() 'bij het klikken op de knop omlaag
Dim Inhoud As String
Dim IndexC As Single
DisplayList = False 'displayverandering afleggen
IndexC = ProgList.ListIndex 'de plaats van het geselecteerde item wordt opgeslagen
If Not IndexC = -1 And Not ProgList.ListCount - 1 = IndexC Then
    'er moet een item geselecteerd zijn en
    'de laatste plaats mag niet geselecteerd zijn
    Inhoud = ProgList.Text 'de inhoud van de geselecteerde item wordt opgeslagen
    ProgList.RemoveItem IndexC 'het geselecteerde item verwijderen
    ProgList.AddItem Inhoud, IndexC + 1 'het terug invoegen maar 1 plaatsje lager
    ProgList.ListIndex = IndexC + 1 'het ingevoegde selecteren
End If
DisplayList = True 'displayverandering aanleggen
Opslaan = True 'op het einde vragen om op te slaan
End Sub
```



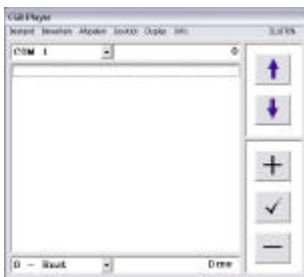
```
Private Sub cmdAdd_Click() 'bij het klikken op de knop toevoegen
Dim Spaties As String
Dim IndexC As Integer
IndexC = ProgList.ListIndex 'de plaats van het geselecteerde item wordt opgeslagen
If Not Len(txtTime.Text) = 0 Then 'als de waarde txtTime is ingevuld
    Spaties = String((13 - Len(Str(txtTime.Text))), " ")
    'aantal spaties berekenen om rechts te kunnen uitlijnen
    ProgList.AddItem " > " & Lijst.Text & Spaties & txtTime.Text & " ms <", IndexC + 1
    'een regel onder het geselecteerde toevoegen
    ProgList.ListIndex = IndexC + 1 'het ingevoegde selecteren
End If
Opslaan = True 'op het einde vragen om op te slaan
End Sub
```



```
Private Sub cmdDelete_Click()
Dim IndexC As Integer
IndexC = ProgList.ListIndex 'de plaats van het geselecteerde item wordt opgeslagen
If Not IndexC = -1 Then 'er moet een item geselecteerd zijn
    DisplayList = False
    ProgList.RemoveItem IndexC 'het geselecteerde item verwijderen
    DisplayList = True
    If ProgList.ListCount = IndexC Then 'als de laatste plaats verwijderd is
        ProgList.ListIndex = IndexC - 1 'het plaats boven verwijderde selecteren
    Else 'als de laatste plaats niet verwijderd is
        ProgList.ListIndex = IndexC 'het plaats van verwijderde selecteren
    End If
End If
Opslaan = True 'op het einde vragen om op te slaan
End Sub
```



```
Private Sub cmdChange_Click()
Dim Spaties As String
Dim IndexC As Integer
IndexC = ProgList.ListIndex 'de plaats van het geselecteerde item wordt opgeslagen
If Not IndexC = -1 Then 'er moet een item geselecteerd zijn
    ProgList.RemoveItem IndexC 'het geselecteerde item verwijderen
    If Not Len(txtTime.Text) = 0 Then 'er moet een item geselecteerd zijn
        Spaties = String((13 - Len(Str(txtTime.Text))), " ")
        ProgList.AddItem " > " & Lijst.Text & Spaties & txtTime.Text & " ms <", IndexC 'aantal spaties berekenen om rechts te kunnen uitlijnen
        'een regel onder het geselecteerde toevoegen
    End If
    ProgList.ListIndex = IndexC 'het plaats van verwijderde selecteren
End If
End If
Opslaan = True 'op het einde vragen om op te slaan
End Sub
```



```
Private Sub Form_Load() 'bij het laden van frmMain
frmSplash.Timer1.Enabled = False 'Timer1 stoppen
DisplayList = True 'displayverandering aanleggen
Combol.Clear 'Lijst voor COM wissen
Combol.AddItem "COM 1", 0 'COM 1 in lijst toevoegen
Combol.AddItem "COM 2", 1 'COM 2 in lijst toevoegen
Combol.AddItem "COM 3", 2 'COM 3 in lijst toevoegen
Combol.AddItem "COM 4", 3 'COM 4 in lijst toevoegen
Lijst.Clear 'Lijst voor richting wissen
Lijst.AddItem "0 - Rust " '0 - Rust in lijst toevoegen
Lijst.AddItem "1 - Vooruit " '1 - Vooruit in lijst toevoegen
Lijst.AddItem "2 - Achteruit " '2 - Achteruit in lijst toevoegen
Lijst.AddItem "3 - Links " '3 - Links in lijst toevoegen
Lijst.AddItem "4 - Rechts " '4 - Rechts in lijst toevoegen
Lijst.ListIndex = 0 '0 - Rust selecteren
Combol.ListIndex = 0 'COM 1 selecteren
CGR.COM_Start 1 'COM 1 openen
Timer2.Interval = COMTIME 'interval tijd van timer2 instellen
Timer2.Enabled = True 'timer2 starten
CGR.Display_Clear 'Display wissen
CGR.Display_ReturnHome 'cursor op begin positie plaatsen
CGR.Display_SetFunction True, True, False '8BitInterface,2Lines,5x7dots
CGR.Display_EntryModeSet True, False 'the cursor moves to the right
CGR.Display_ON_OFF True, False, False 'Display ON,Cursor OFF,Blink OFF
Tekst1 " Ruben D'Hulst 7" 'verwijzen naar code Tekst1
Tekst2 " Yves Yde 17" 'verwijzen naar code Tekst2
CGR.Holtek12Encoder 0 'Adres op 0 plaatsen
If Not Len(Command) = 0 Then 'als een File is aan geklikt
    BestandsPad = Command 'bestandspad opslaan
    BestandsNaam = StripPath(BestandsPad)
    'bestandsnaam uit bestandspad filteren
    If Len(BestandsPad) = 0 Then 'als er geen bestandspad aanwezig is
        Me.Caption = "CGR Player" 'naam van frmMain wijzigen
    Else 'als er een bestandspad aanwezig is
        Me.Caption = BestandsNaam & " - CGR Player" 'naam van frmMain wijzigen
        LoadFile (BestandsPad) 'verwijzen naar code Loadfile
    End If
End If
Opslaan = False 'op het einde vragen om op te slaan
End Sub
```

```

Private Function StripPath(T$) As String      'deze code dient om de bestandsnaam
                                             'uit het bestandspad te filteren
Dim X%, ct%
StripPath$ = T$
X% = InStr(T$, "\")
Do While X%
    ct% = X%
    X% = InStr(ct% + 1, T$, "\")
Loop
If ct% > 0 Then StripPath$ = Mid$(T$, ct% + 1)
End Function

```

0

```

Private Sub HTE_Change()                    'wijzigen van het adres
On Error GoTo Fout                          'bij problemen ga naar fout
If (HTE.Text >= 0) And (HTE.Text <= 255) Then
    HT = Val(HTE.Text)                      'als het adres tussen 0 en 255 ligt
    CGR.Holtek12Encoder Val(HTE.Text)      'adres wijzigen op printplaat
Else                                         'als het adres niet tussen 0 en 255 ligt
    HTE.Text = HT                          'vorige waarde van het adres weergeven
End If
Exit Sub                                    'Sub beëindigen
Fout:                                       'bij fouten of problemen hier verder lezen
    HTE.Text = HT                          'vorige waarde van het adres weergeven
End Sub

```

STOP

SLUITEN

```

Private Sub mnuClose_Click()                'klikken op de blak Sluiten/Stop
If mnuClose.Caption = String(39, " ") & "SLUITEN" Then
    mnuSluiten_Click                       'als de balk Sluiten noemt dan
ElseIf mnuClose.Caption = String(44, " ") & "STOP" Then
    mnuStop_Click                          'als de balk Stop noemt dan
End If                                     'verwijzen naar code mnuSluiten_Click
End Sub                                    'verwijzen naar code mnuStop_Click

```

Display

```

Private Sub mnuDisplay_Click()              'klikken op het menu Display
CGR.Display_Clear                          'display wissen
CGR.Display_ReturnHome                    'cursor op begin positie plaatsen
frmDisplay.Show                           'frmDisplay weergeven
frmMain.Hide                              'frmMain verbergen
End Sub

```

Joystick

```

Private Sub mnuJoystick_Click()            'klikken op het menu Joystick
CGR.Display_Clear                          'display wissen
CGR.Display_ReturnHome                    'cursor op begin positie plaatsen
Tekst1 " Eindwerk 2003-2004 "             'verwijzen naar code Tekst1
Tekst2 " Joystick besturing "             'verwijzen naar code Tekst2
frmJoy.Show                               'frmJoy weergeven
frmMain.Hide                              'frmMain verbergen
End Sub

```



```

Private Sub mnuOpslaan_Click() 'klikken op het menu Opslaan
Dim IndexC As Integer
IndexC = ProgList.ListIndex 'de plaats van het geselecteerde item wordt opgeslagen
If Not Dir$(BestandsPad) = "" And Not BestandsPad = "" Then
Kill BestandsPad 'als het bestandspad bestaat dan
WriteFile BestandsPad 'bestand verwijderen
Else 'verwijzen naar code WriteFile
mnuOpslaanAls_Click 'als het bestandspad niet bestaat dan
'verwijzen naar code mnuOpslaanAls_Click
End If
ProgList.ListIndex = IndexC 'het aangeklikte item wordt terug geselecteerd
End Sub

```



```

Private Sub mnuOpslaanAls_Click() 'klikken op het menu Opslaan Als
Dim IndexC As Integer
IndexC = ProgList.ListIndex 'de plaats van het geselecteerde item wordt opgeslagen

With dlgCD 'verwijzen naar een commondialog
If Not Len(BestandsPad) = 0 Then 'als er een bestandspad aanwezig is
.FileName = BestandsPad 'de directory van bestandspad weergeven
End If
.DialogTitle = "CGR Opslaan" 'de titel van het kader opslaan veranderen
.CancelError = False 'foutmelding bij annuleren afleggen
.Filter = "CGR Files (*.cgr)|*.cgr" 'enkel .cgr files weergeven
.ShowSave 'het kader opslaan als weergeven
If Right(.FileName, 4) = LCase(".cgr") Then
BestandsPad = .FileName 'als de extensie aanwezig is
'bestandsnaam op vragen
Else 'als de extensie niet aanwezig is
BestandsPad = .FileName & ".cgr" 'bestandsnaam op vragen en extensie toevoegen
End If
End With
If Not Dir$(BestandsPad) = "" Then 'als het bestandspad bestaat dan
Kill BestandsPad 'bestand verwijderen
End If

BestandsNaam = StripPath(BestandsPad) 'bestandsnaam uit bestandspad filteren
If Len(BestandsNaam) = 0 Then 'als er geen bestandspad aanwezig is
Me.Caption = "CGR Player" 'naam van frmMain wijzigen
Else 'als er een bestandspad aanwezig is
Me.Caption = BestandsNaam & " - CGR Player" 'naam van frmMain wijzigen
End If
WriteFile BestandsPad 'verwijzen naar code WriteFile
ProgList.ListIndex = IndexC 'het aangeklikte item wordt terug geselecteerd
End Sub

```

```

Private Sub Form_Unload(Cancel As Integer) 'bij het klikken op het kruisje
mnuSluiten_Click 'verwijzen naar code mnuSluiten_Click
End Sub

```

Info

```
Private Sub mnuInfo_Click()           'bij het klikken op het menu Info
CGR.Display_Clear                    'display wissen
CGR.Display_ReturnHome              'cursor op begin positie plaatsen
Tekst1 " Eindwerk 2003-2004 "       'verwijzen naar code Tekst1
Tekst2 "      CGR  INFO      "      'verwijzen naar code Tekst2
frmAbout.Show                       'frmAbout weergeven
frmMain.Hide                        'frmMain verbergen
End Sub
```

Bewerken



Invoegen
Wijzigen
Verwijderen

```
Private Sub mnuInvoegen_Click()      'bij het klikken op het menu Invoegen
cmdAdd_Click                         'verwijzen naar code cmdAdd_Click
End Sub
```

Bestand



Nieuw
Openen ...
Opslaan
Opslaan Als ...
Sluiten

```
Private Sub mnuNieuw_Click()         'bij het klikken op het menu Nieuw
Dim i
If Opslaan = True Then              'als Opslaan = True dan
i = MsgBox("Wilt u dit bestand opslaan?" & vbNewLine & BestandsPad, _ 'msgbox weergeven, vragen om op te slaan
    vbExclamation + vbYesNo, "CGR Player")
    If i = vbYes Then                'als men ja heeft geklikt
        mnuOpslaan_Click            'verwijzen naar code mnuOpslaan_Click
    End If
End If
BestandsPad = ""                    'bestandspad wissen
BestandsNaam = ""                  'bestandsnaam wissen
ProgList.Clear                     'proglis t wissen
Me.Caption = "CGR Player"          'naam van frmMain wijzigen
Lijst.ListIndex = 0                '0 - Rust selecteren
txtTime.Text = 0                   'invoervak tijd op 0 plaatsen
CGR.Display_Clear                  'Display wissen
CGR.Display_ReturnHome              'cursor op begin positie plaatsen
Tekst1 " Ruben D'Hulst      7"     'verwijzen naar code Tekst1
Tekst2 " Yves  Yde         17"     'verwijzen naar code Tekst2
End Sub
```



```

Private Sub mnuOpen_Click() 'bij het klikken op het menu Openen
Dim i
If Opslaan = True Then 'als Opslaan = True dan
i = MsgBox("Wilt u dit bestand opslaan?" & vbCrLf & BestandsPad, _
vbExclamation + vbYesNo, "CGR Player") 'msgbox weergeven, vragen om op te slaan
If i = vbYes Then 'als men ja heeft geklikt
mnuOpslaan_Click 'verwijzen naar code mnuOpslaan_Click
End If
End If

With dlgCD 'verwijzen naar een commondialog
.DialogTitle = "CGR Open" 'de titel van het kader opslaan veranderen
.CancelError = False 'foutmelding bij annuleren afleggen
.Filter = "CGR Files (*.cgr)|*.cgr" 'enkel .cgr files weergeven
.ShowOpen 'het kader openen als weergeven
BestandsPad = .FileName 'bestandspad opvragen
End With
BestandsNaam = StripPath(BestandsPad) 'bestandsnaam uit bestandspad filteren
ProgList.Clear 'prolist wissen
If Len(BestandsPad) = 0 Then 'als er geen bestandspad aanwezig is
Me.Caption = "CGR Player" 'naam van frmMain wijzigen
Else 'als er een bestandspad aanwezig is
Me.Caption = BestandsNaam & " - CGR Player" 'naam van frmMain wijzigen
LoadFile (BestandsPad) 'verwijzen naar code LoadFile
End If
End Sub

```

```

Private Sub LoadFile(Bestand As String) 'de code LoadFile
Dim Teller As Integer
Dim Spaties As String
DisplayList = False 'displayverandering afleggen
OpenFile Bestand 'verwijzen naar code OpenFile
For Teller = 1 To AantalRecords 'tellen van 1 tot aantalrecords in een file
ShowRecord Teller 'verwijzen naar code ShowRecord
Spaties = String((13 - Len(Str(Tijd))), " ") 'aantal spaties berekenen om rechts te kunnen uitlijnen
Lijst.ListIndex = Richting 'plaats van het item geven dat moet toegevoegd worden
ProgList.AddItem " > " & Lijst.Text & Spaties & Tijd & " ms <" 'een regel toevoegen
Next Teller 'Teller + 1
CloseFile 'verwijzen naar code CloseFile
If Not ProgList.ListCount = 0 Then 'als er items aanwezig zijn
ProgList.ListIndex = 0 'het eerste item selecteren
End If
DisplayList = True 'displayverandering aanleggen
Opslaan = False 'op het einde niet meer vragen om op te slaan
End Sub

```

```

Private Sub WriteFile(Bestand As String) 'de code WriteFile
Dim Teller As Integer
DisplayList = False 'displayverandering afleggen
If Not ProgList.ListCount = 0 Then 'als er items aanwezig zijn
OpenFile Bestand 'verwijzen naar code OpenFile
For Teller = 0 To ProgList.ListCount - 1 'tellen van 0 tot laatste item van de proglis
ProgList.ListIndex = Teller 'item selecteren met plaats = Teller
Richting = Val(Mid(ProgList.Text, 4, 1)) 'Richting uit het geselecteerde item filteren
Tijd = Val(Mid(ProgList.Text, 18, 11)) 'Tijd uit het geselecteerde item filteren
NewRecord Richting, Tijd 'verwijzen naar code Newrecord
Next Teller 'Teller + 1
CloseFile 'verwijzen naar code CloseFile
End If
DisplayList = True 'displayverandering aanleggen
Opslaan = False 'op het einde niet meer vragen om op te slaan
End Sub

```

```

Private Sub ShowRecord(Number As Integer) 'de code Showrecord
    Get 1, Number, Punt 'record uit file halen
    Richting = Punt.R 'richting opslaan
    Tijd = Punt.T 'tijd opslaan
End Sub

```

```

Private Sub NewRecord(R As Single, T As Single) 'de code Newrecord
    Punt.R = R 'richting in type Coördinaat opslaan
    Punt.T = T 'tijd in type Coördinaat opslaan
    RecordLengte = Len(Punt) 'record lengte berekenen
    AantalRecords = LOF(1) / RecordLengte 'aantal records berekenen
    Put 1, AantalRecords + 1, Punt 'record wegschrijven
End Sub

```

```

Private Sub OpenFile(FilePath As String) 'de code OpenFile
    RecordLengte = Len(Punt) 'record lengte berekenen
    Open FilePath For Random As 1 Len = RecordLengte 'bestand als naam 1 openen
    AantalRecords = LOF(1) / RecordLengte 'aantal records berekenen
End Sub

```

```

Private Sub CloseFile() 'de code closefile
    Close 1 'bestand met naam 1 sluiten
End Sub

```



```

Private Sub mnuPlay_Click() 'bij klikken op het menu mnuPlay_Click
If Not ProgList.ListCount = 0 Then 'er moeten items aanwezig zijn
    mnuPlay.Enabled = False 'het menu Play uitschakelen
    mnuStop.Enabled = True 'het menu Stop inschakelen
    Lijst.Enabled = False 'lijst richtingen uitschakelen
    ProgList.Enabled = False 'lijst instructies uitschakelen
    txtTime.Enabled = False 'invoervak tijd uitschakelen
    cmdUp.Enabled = False 'knop naar boven verplaatsen uitschakelen
    cmdDown.Enabled = False 'knop naar beneden verplaatsen uitschakelen
    cmdAdd.Enabled = False 'knop toevoegen uitschakelen
    cmdChange.Enabled = False 'knop wijzigen uitschakelen
    cmdDelete.Enabled = False 'knop wissen uitschakelen
    Combol.Enabled = False 'lijst compoorten uitschakelen
    HTE.Enabled = False 'invoervak adres uitschakelen
    lblms.Enabled = False 'de tekst "ms" uitschakelen
    mnuBestand.Enabled = False 'het menu Bestand uitschakelen
    mnuBewerken.Enabled = False 'het menu Bewerken uitschakelen
    mnuJoystick.Enabled = False 'het menu Joystick uitschakelen
    mnuDisplay.Enabled = False 'het menu Display uitschakelen
    mnuInfo.Enabled = False 'het menu Info uitschakelen
    mnuClose.Caption = String(44, " ") & "STOP" 'balk Sluiten veranderen in Stop
    MaxP = ProgList.ListCount - 1 'plaats van laatste item opslaan
    TellerP = 0 'tellerp op nul plaatsen
    Afspelen 'verwijzen naar code Afspelen
End If
End Sub

```

```

Private Sub Afspelen()
On Error GoTo Fout
If TellerP > MaxP Then
    mnuStop_Click
Else
    ProgList.Refresh
    ProgList.ListIndex = TellerP
    Richting = Val(Mid(ProgList.Text, 4, 1))

    Tijd = Val(Mid(ProgList.Text, 18, 11))
    Select Case Richting
    Case 0
        CGR.MotorStop
    Case 1
        CGR.Translatie True
    Case 2
        CGR.Translatie False
    Case 3
        CGR.Rotatie False
    Case 4
        CGR.Rotatie True
    End Select
    If Tijd = 0 Then
        TellerP = TellerP + 1
        Afspelen
    Else
        Timer.Interval = Tijd
        Timer.Enabled = True
    End If
End If
Exit Sub
Fout:
TellerP = TellerP + 1
Afspelen
End Sub

```



```

Private Sub Timer_Timer()
Timer.Enabled = False
TellerP = TellerP + 1
Afspelen
End Sub

```

Afspelen

Play

Stop

```
Private Sub mnuStop_Click()
CGR.Com_Restart
Timer.Enabled = False
CGR.MotorStop
ProgList.ListIndex = 0
ProgList.Enabled = True
mnuPlay.Enabled = True
mnuStop.Enabled = False
Lijst.Enabled = True
txtTime.Enabled = True
cmdUp.Enabled = True
cmdDown.Enabled = True
cmdAdd.Enabled = True
cmdChange.Enabled = True
cmdDelete.Enabled = True
Combo1.Enabled = True
HTE.Enabled = True
lblms.Enabled = True
mnuBestand.Enabled = True
mnuBewerken.Enabled = True
mnuJoystick.Enabled = True
mnuDisplay.Enabled = True
mnuInfo.Enabled = True
mnuClose.Caption = String(39, " ") & "SLUITEN"

Tekst1 " Ruben D'Hulst      7"
Tekst2 " Yves Yde          17"
End Sub
```

'bij het klikken op Stop
'compoort opnieuw starten
'timer stoppen
'motoren stoppen
'eerste item selecteren
'lijst instructies inschakelen
'het menu Play inschakelen
'het menu Stop uitschakelen
'lijst richtingen inschakelen
'invoervak tijd inschakelen
'knop naar boven verplaatsen inschakelen
'knop naar beneden verplaatsen inschakelen
'knop toevoegen inschakelen
'knop wijzigen inschakelen
'knop wissen inschakelen
'lijst compoorten inschakelen
'invoervak adres inschakelen
'de tekst "ms" inschakelen
'het menu Bestand inschakelen
'het menu Bewerken inschakelen
'het menu Joystick inschakelen
'het menu Display inschakelen
'het menu Info inschakelen
'balk Stop veranderen in Sluiten
'verwijzen naar code Tekst1
'verwijzen naar code Tekst2

Bestand

Nieuw

Openen ...

Opslaan

Opslaan Als ...

Sluiten

```
Private Sub mnuSluiten_Click()
Dim i
If Opslaan = True Then
i = MsgBox("Wilt u dit bestand opslaan?" & vbNewLine & BestandsPad, _
vbExclamation + vbYesNoCancel, "CGR Player")
If i = vbYes Then
mnuOpslaan_Click
Afsluiten
ElseIf i = vbNo Then
Afsluiten
End If
Else
Afsluiten
End If
End Sub
```

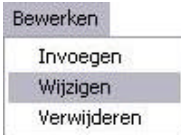
'bij het klikken op het menu Sluiten
'als Opslaan = True dan
'als men op Ja klikt
'verwijzen naar code mnuOpslaan_Click
'verwijzen naar code Afsluiten
'als men op Neen klikt
'verwijzen naar code Afsluiten
'verwijzen naar code Afsluiten

```
Private Sub Afsluiten()
CGR.Display_Clear
CGR.Display_ReturnHome
CGR.Display_SetFunction True, True, False
CGR.Display_EntryModeSet True, False
CGR.Display_ON_OFF True, False, False
Tekst1 " Ruben D'Hulst      7"
Tekst2 " Yves Yde          17"
CGR.MotorStop
CGR.TransmitterStop
Timer2.Enabled = False
End Sub
```

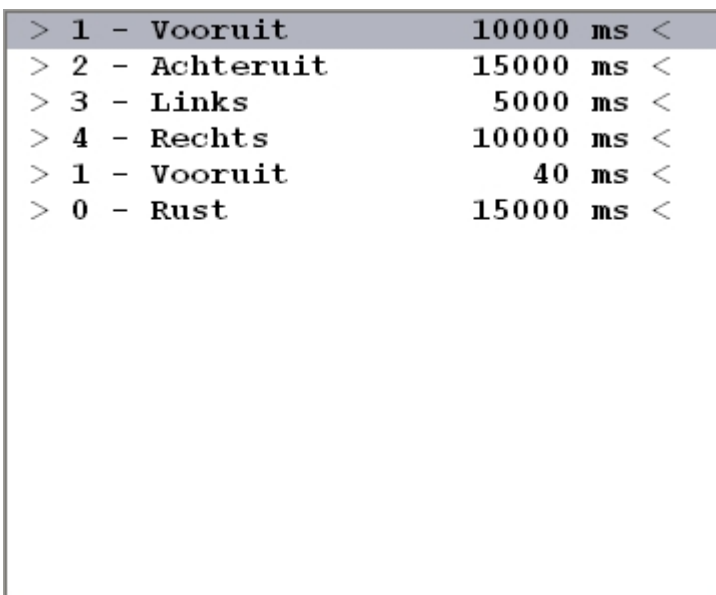
'de code Afsluiten
'Display wissen
'cursor op begin positie plaatsen
'8BitInterface,2Lines,5x7dots
'the cursor moves to the right
'Display ON,Cursor OFF,Blink OFF
'verwijzen naar code Tekst1
'verwijzen naar code Tekst2
'motoren stoppen
'holtek encoder stoppen
'timer2 stoppen
'programma beëindigen



```
Private Sub mnuVerwijderen_Click()           'bij het klikken op het menu Verwijderen
cmdDelete_Click                             'verwijzen naar code cmdDelete_Click
End Sub
```



```
Private Sub mnuWijzigen_Click()             'bij het klikken op het menu Wijzigen
cmdChange_Click                             'verwijzen naar code cmdChange_Click
End Sub
```



```
Private Sub ProgList_Click()               'bij het klikken in de proglis
Dim TextR As String, TextT As String
Lijst.ListIndex = Mid(ProgList.Text, 4, 1) 'richting weergeven in lijst
txtTime.Text = Val(Mid(ProgList.Text, 18, 11)) 'tijd weergeven in tekstvak

If DisplayList = True Then                'als DisplayList = True dan
    Select Case Lijst.ListIndex
        Case 0                             'als richting = 0
            TextR = "Rust"
        Case 1                             'als richting = 1
            TextR = "Vooruit"
        Case 2                             'als richting = 2
            TextR = "Achteruit"
        Case 3                             'als richting = 3
            TextR = "Links"
        Case 4                             'als richting = 4
            TextR = "Rechts"
    End Select
    TextT = String(5 - Len(txtTime.Text), " ") & txtTime.Text & " ms"
                                           'tijd rechts uitlijnen
    CGR.Display_Clear                       'display wissen
    CGR.Display_ReturnHome                 'cursor op begin positie plaatsen
    Tekst1.Text = TextR & TextT            'richting en tijd weergeven op display
End If
End Sub
```

```
Private Sub ProgList_KeyDown(KeyCode As Integer, Shift As Integer)
If KeyCode = 46 Then      'bij het indrukken van de DEL toets
cmdDelete_Click         'verwijzen naar code cmdDelete_Click
End If
End Sub
```



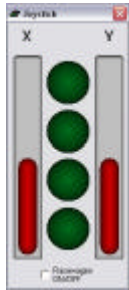
```
Private Sub Timer2_Timer()      'telkens na het ingestelde interval van timer2
CGR.Com_Restart                'compoort opnieuw openen
End Sub
```

0 ms

```
Private Sub txtTime_Change()    'tijd wijzigen in tekstvak
On Error GoTo Fout             'bij problemen ga naar fout
If (txtTime.Text >= 0) And (txtTime.Text <= 65535) Then
'als de tijd tussen 0 en 65535 ligt dan
'tijd opslaan
Else
'als de tijd niet tussen 0 en 65535 ligt dan
'tijd veranderen in het vorige opgeslagen getal
End If
Exit Sub                       'Sub beëindigen
Fout:                          'bij fouten of problemen hier verder lezen
txtTime.Text = tT              'tijd veranderen in het vorige opgeslagen getal
End Sub
```

7.4.3 frmJoy

Option Explicit
Dim X As Integer, Y As Integer



```
Private Sub Form_Load() 'bij het laden van frmJoy
Timer1.Enabled = True 'timer1 starten met interval 80 ms
Timer2.Interval = COMTIME 'interval van timer2 instellen
Timer2.Enabled = True 'timer2 starten
End Sub
```

```
Private Sub Form_Unload(Cancel As Integer) 'bij het sluiten van frmJoy
Timer1.Enabled = False 'timer1 stoppen
CGR.MotorStop 'motoren stoppen
CGR.Display_Clear 'display wissen
CGR.Display_ReturnHome 'cursor op begin positie plaatsen
Tekst1 " Ruben D'Hulst 7" 'verwijzen naar code Tekst1
Tekst2 " Yves Yde 17" 'verwijzen naar code Tekst2
Timer2.Enabled = False 'timer2 stoppen
frmJoy.Hide 'frmJoy verbergen
frmMain.Show 'frmMain weergeven
End Sub
```



```
Private Sub Timer1_Timer() 'om de 80ms wordt deze code herhaald
Dim Value As Integer
CGR.JoystickRead 'joystick binnenlezen
X = CGR.PositionX 'X-as beweging opvragen
Y = CGR.PositionY 'Y-as beweging opvragen
Joy1.Value X
Joy2.Value Y
Led1.Value CGR.Button1 'groene bol wordt zichtbaar bij button = true
Led2.Value CGR.Button2 'groene bol wordt onzichtbaar bij button = false
Led3.Value CGR.Button3 'dit geldt voor de 4 bollen
Led4.Value CGR.Button4
If ONOFF.Value = 1 Then 'als de checkbox is aangevinkt
Value = (X + 1) + (Y + 1) * 3 + 1 'decimale waarde berekenen van de assen
Select Case Value
Case 2 'als de waarde = 2
CGR.Translatie True 'vooruit rijden
Case 8 'als de waarde = 8
CGR.Translatie False 'achteruit rijden
Case 4 'als de waarde = 4
CGR.Rotatie False 'naar recht draaien
Case 6 'als de waarde = 6
CGR.Rotatie True 'naar links draaien
Case Else 'als de waarde gelijk is aan iets anders
CGR.MotorStop 'motoren stoppen
End Select
Else
CGR.MotorStop 'motoren stoppen
End If
End Sub
```



```
Private Sub Timer2_Timer() 'telkens na het ingestelde interval van Timer2
CGR.Com_Restart 'compoort opnieuw openen
End Sub
```

7.4.4 Module

```
Type Coördinaat
  R As Single
  T As Single
End Type
```

```
Type Instructie
  Lijn As String * 4
End Type
```

```
Global CGR As New CGRPrintplaat.Racewagen
Global Const COMTIME = 1000
```

```
Sub Tekst1(Tekst As String)          'de code Tekst1
Dim Teller As Integer
CGR.Display_SetDdRamAddress 0        'cursor op plaats 0 zetten
For Teller = 1 To Len(Tekst)         'teller starten van eerste teken tot laatste
CGR.Display_Tekst Mid(Tekst, Teller, 1) 'ieder teken apart wegschrijven naar display
Next Teller                          'teller +1
End Sub
```

```
Sub Tekst2(Tekst As String)          'de code Tekst2
Dim Teller As Integer
CGR.Display_SetDdRamAddress 40       'cursor op plaats 40 zetten
For Teller = 1 To Len(Tekst)         'teller starten van eerste teken tot laatste
CGR.Display_Tekst Mid(Tekst, Teller, 1) 'ieder teken apart wegschrijven naar display
Next Teller                          'teller +1
End Sub
```

8 Bronnen

8.1 Websites

DANIELS, C., 'Werken met bestanden', internet,

(<http://www.khlim.be/~cdaniels/VisualBasic8.PDF>)

VDDER, 'Lichteffecten met de PC', internet,

(<http://users.pandora.be/laboe/Leerkrachten/Eric/Lichteffecten%20met%20de%20computer.pdf>)

DE REEDER, J., 'Experiment 7. Bi-directional Control Of Motors And The H-Bridge', internet,

(<http://www.learn-c.com/experiment7.htm>)

HOLTEK, '2¹² Series of Encoders', internet, 24/01/2003,

(http://www.holtek.com/pdf/consumer/2_12ev110.pdf)

HOLTEK, '2¹² Series of Decoders', internet, 18/11/2002,

(http://www.holtek.com/pdf/consumer/2_12dv110.pdf)

NATIONAL SEMICONDUCTOR, 'LM78XX Series Voltage Regulators', internet, mei 2000,

(<http://cache.national.com/ds/LM/LM7512C.pdf>)

ÖGREN, J., 'PC Gameport', internet, 21/01/2004,

(<http://www.hardwarebook.net/connector/userinput/gameportpc.html>)

'Visual Basic Categories', internet,

(<http://pscode.com/vb/default.asp?lngWId=1#categories>)

'Tips & Tricks', internet,

(<http://cuinl.tripod.com/tips-and-tricks-all.htm>)

'Tips & Tricks', internet,

(<http://www.vbexplorer.com/VBExplorer/vbtips.asp>)

'IR Remote Control', internet,

(http://www.divelectronics.com/Projects/IR_Remote.html)

'Type conversie en Typefuncties.', internet, "21/03/2002",

(http://home2.pi.be/rudiea/les_10.htm)

'Tips & Co – Overzicht', internet,

(<http://www.vbgroup.nl/techniek/TipsCo.asp>)

STMICROELECTRONICS, 'L298', internet,

(www.learn-c.com/l298.pdf)

8.2 Boeken

VANDENBORN, T., Telecommunicatie 1, Die Keure, Brugge, 160 pagina's.

CLAEYS, J., Datacommunicatie, Die Keure, Brugge, 151 pagina's

ZOLLER, B., Pc en electronica, Easy Computing, Brussel, 1997, 279 pagina's

9 Bijlage 1: Maken van printplaten

9.1 Wat is een printplaat

Een printplaat bestaat uit een elektrisch niet geleidend basismateriaal dat is voorzien van een elektrisch geleidende laag of lagen, meestal koper. De koperlaag wordt dan gedeeltelijk weggeëtsd (opgelost in een zuur), waardoor er printbanen ontstaan, die de verschillende onderdelen volgens schema moeten verbinden.

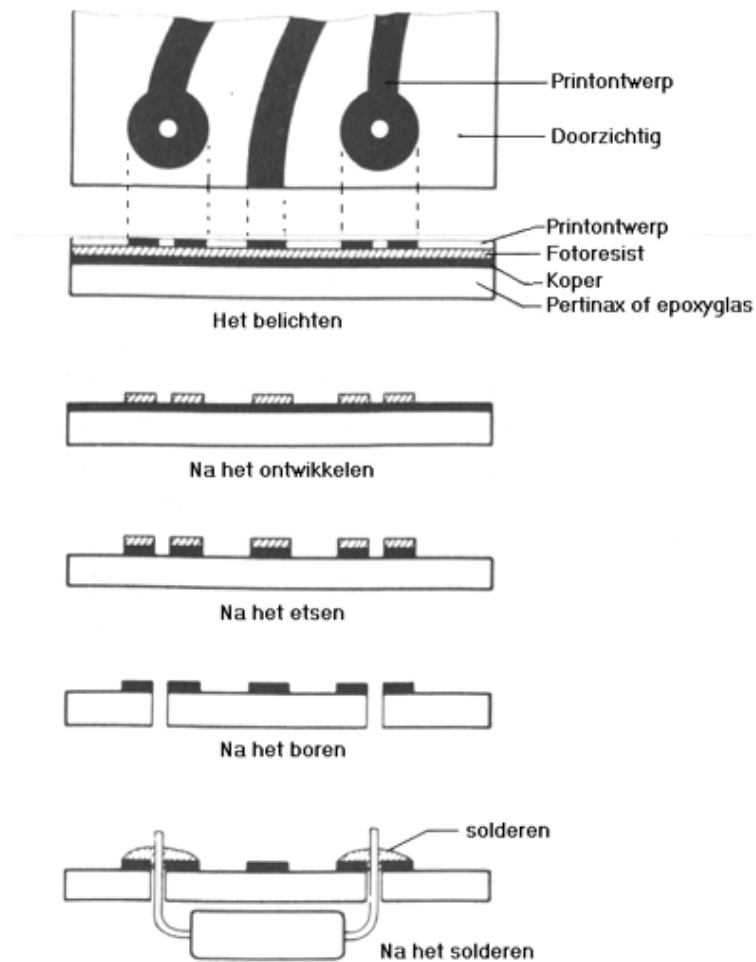
De printen kunnen enkelzijdig, dubbelzijdig en met meer lagen zijn uitgevoerd. Bij de enkelzijdige print is aan een kant van het basismateriaal een koper sporenpatroon aanwezig. Bij de dubbelzijdige print is dit sporenpatroon aan beide zijden aanwezig. Beide zijden worden via doorgemetalliseerde gaten doorverbonden (gaten waar ook geleidend materiaal is aangebracht). Deze doormetallisering maakt een dubbelzijdige print moeilijker te produceren en daardoor aanzienlijk duurder. Een meerlagen- of ook wel multilayerprint bestaat uit meerdere lagen basismateriaal en koper sporenpatroon.

9.2 Hoe wordt een printplaat gemaakt

- Eerst wordt van het elektronische schema een lay-out of printontwerp gemaakt. Dit is dus een tekening van het bedradingspatroon.
- Vervolgens wordt dit printontwerp op de koperlaag van de printplaat overgebracht. Hiervoor zijn verschillende technieken:
 - Je kunt het printontwerp op een transparant afprinten en daarna de lichtgevoelige printplaat belichten met de transparant erbovenop. De plaatsen die niet werden beschermd door de inkt op de transparant worden zacht (week) en kunnen zo weggeëtsd worden.
 - De transparant kan ook worden vervangen door een alcoholstift of iets dergelijks. Je tekent dan het schema met stift op de plaat en zo bescherm je de plaatsen waar het koper moet blijven. Daarna wordt er opnieuw geëtsd.
- Na het etsen worden alle resten van het etsmiddel en eventueel ook het ets-resist (alcoholstift) verwijderd met thinner of aceton.
- Nu wordt de print bespoten met een printlak om oxidatie te voorkomen.



- Als de lak droog is moeten nog enkel de gaatjes geboord worden en dan kunnen de componenten op hun juiste plaats worden vastgesoldeerd.



9.3 Printplaten ontwerpen

9.3.1 Ontwerp procedure

Eerst verzamelen we alle onderdelen die op de print moeten komen bij elkaar. Je moet immers de afmetingen kennen van de componenten om de boorpunten te kunnen bepalen en om te weten hoeveel plaats voor elk onderdeel voorzien moet worden.

Ook belangrijk zijn de aansluitgegevens van de componenten. Deze haalt men meestal uit datasheets.

Als men de print in een behuizing wil monteren is men gebonden aan enkele voorwaarden. De afmetingen en de vorm van de print speelt daar een belangrijke rol. Daarmee moet je dan ook rekening houden.

Wanneer je een draadverbinding met een toepassing buiten de schakeling wil maken gebruik je daar best de daarvoor geschikte connectoren. Zoals in ons geval een seriële poort, een gamepoort, een voeding ... aanwezig is. Deze plaats je het best aan de buitenkant van de print, zo zitten deze achteraf niet in de weg.

Voor het definitieve ontwerp van onze print maakten we gebruik van het computer programma Protel 99 SE.

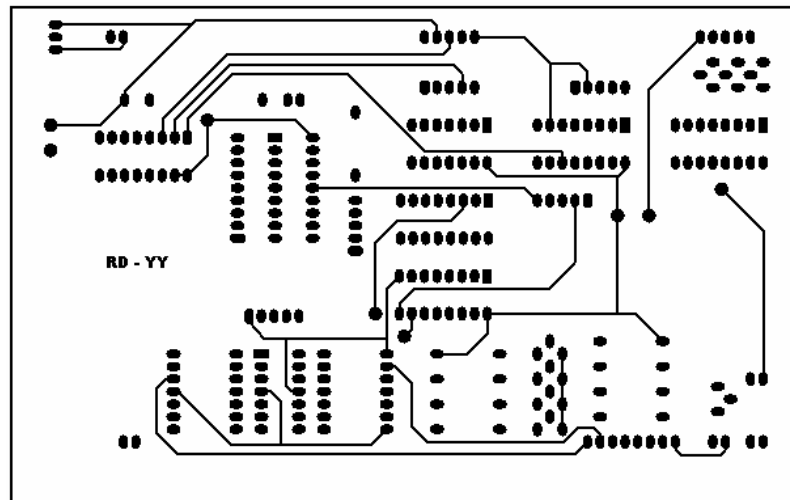
Bij de meeste PC-ontwerpprogramma's vertrekt men van het schema van de schakling die op de print moet komen. Dezelfde aandachtspunten als hierboven blijven in principe behouden.

Een print ontwerpen met de pc geeft meestal het beste eindresultaat. Zo is er een vrij uitgebreide componentenbibliotheek of meerdere bibliotheken voorhanden met aansluitgegevens en afmetingen van de verschillende componenten. Uitzonderlijke componenten kunnen zelf gedefinieerd of gedownload worden. Men kan eveneens opgeven wat de afstand en/of de dikte van de printsporen moet zijn,

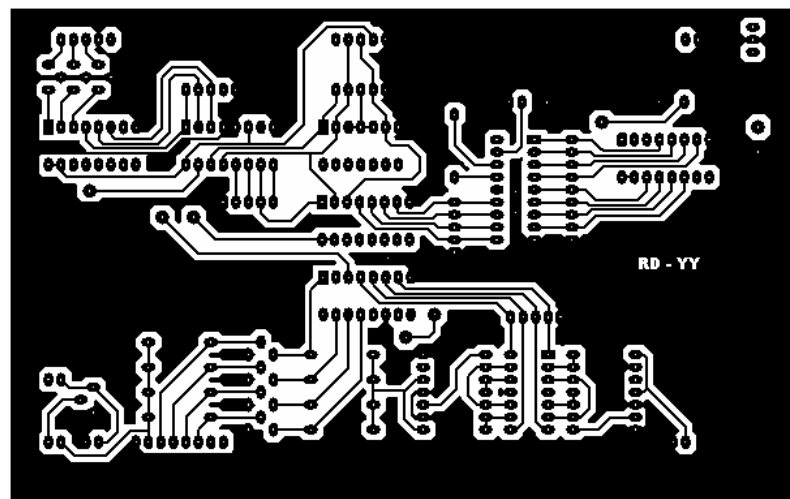
Wat echter niet automatisch gebeurt, is het plaatsen van de componenten. Deze plaatst men dan ook zelf. Het programma zal echter wel aangeven wat theoretisch de beste plaats is. Als er een compromis tussen de theoretische beste plaats en de gebruikerswensen is gevonden, kan men het geheel laten uittekenen (printsporen laten berekenen).

We laten onze print niet autorouten omdat de resultaten met de pc voor teveel problemen zorgden. We bepaalden manueel de ligging van de componenten en de printbanen.

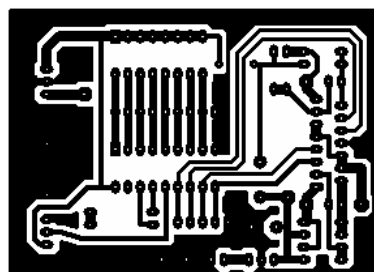
9.3.2 Het resultaat



Figuur 9: Bovenzijde zendmodule



Figuur 10: Onderzijde zendmodule



Figuur 11: Onderzijde ontvangmodule

9.4 Het etsen

Wanneer men een print gaat etsen moet men de nodige veiligheidsmaatregelen treffen:

- een goed geventileerde ruimte;
- handbescherming;
- oogbescherming en
- bescherming voor uw lichaam en kledij (een labo jas).

Bij eventueel contact van het zuur met uw huid moet je zo snel mogelijk overvloedig spoelen met lauw water.

Voor het te gebruiken zuur zijn meerdere mogelijkheden:

- IJzertrichloride (FeCl_3): 500g oplossen in 1l water, etstemperatuur ca. 40-50°C
- Ammoniumpersulfaat ($(\text{NH}_4)_2\text{S}_2\text{O}_8$): 100g oplossen in 1l water, etstemperatuur ca. 40-50°C
- Salpeterzuur (HNO_3): verhouding 1:1 in water, etstemperatuur ca. 40-50°C
- Zoutzuur (HCl) & waterstofperoxyde (H_2O_2): 200ml zoutzuur (35%) + 30ml waterstofperoxyde (30%) + 770ml water, etstemperatuur: 20-40°C

Volgens enkele ondervindingen blijkt dat de onderstaande zuren de beste resultaten geven.



IJzertrichloride

Ammoniumpersulfaat

Nu de nodige maatregelen zijn getroffen en het printontwerp om de print is aangebracht kunnen we van start gaan. Om te beginnen wordt de oplossing om te etsen klaargemaakt. Het is uitermate belangrijk dat het etsen gebeurt in een glazen of kunststof bad/pot.

Gebruik geen metalen potten (ook geen roestvast staal of "INOX"), want de zuren bijten sterk! Het beste is een "pyrex" pot, dit is warmtevast laboglas.

Nu moet de oplossing nog op temperatuur worden gebracht. In geval van een pyrex pot kan men deze rechtstreek op een vuur plaatsen. Een kunststof pot kan men "au bain marie" verwarmen. Zorg hierbij voor dat de doeltemperatuur niet al te veel overschreven wordt. Dit heeft namelijk weinig nut en kan zelfs gevaarlijk zijn. Laat de oplossing zeker NIET koken.

Eenmaal het bad op temperatuur is mag men de warmtebron uitschakelen en de print in het etsbad brengen. Zorg hierbij voor beweging van het etsbad d.m.v. te roeren met een plastic lepeltje.

Na ongeveer 10 min. moet alle koper (buiten de banen) weg zijn en moet de print uit het bad te gehaald worden. Laat de print niet langer dan nodig in het bad omdat dan de stift of foto-resist mee zal wegetsen!

9.5 Spoelen

Na het etsproces is het van groot belang de printplaat goed te spoelen in water. Dit is om ervoor te zorgen dat alle ets materiaal verwijderd is. Zoniet, dan worden de koper baantjes beschadigd.

10 Bijlage 2: De transistor

10.1 Werking

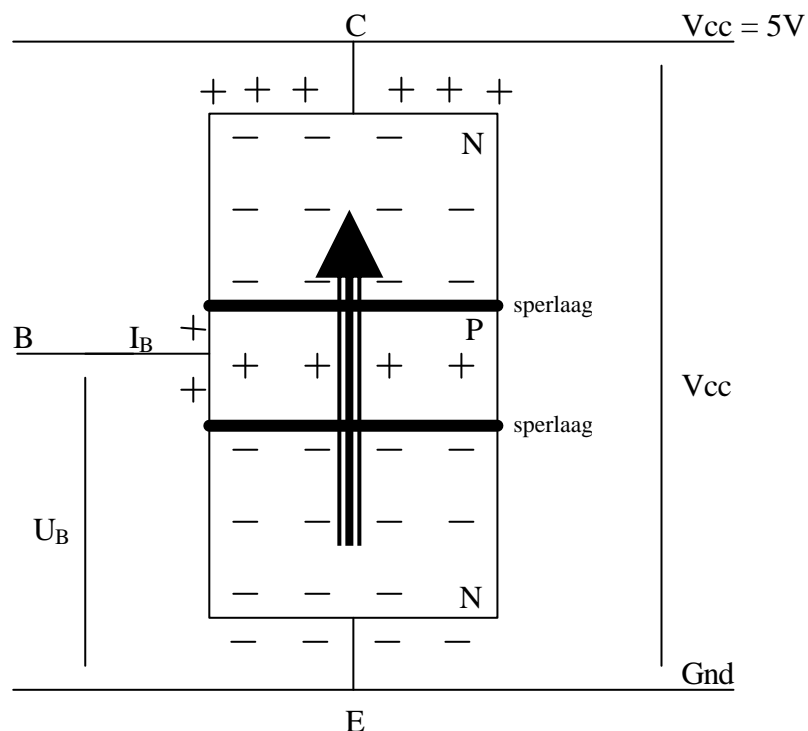
De transistor is een component die opgebouwd is uit 2 soorten kristallen:

- P-kristal
- N-kristal

Het kenmerk van het P-kristal is dat die meer positieve vrije ladingsdragers bezit dan negatieve. Bij het N-kristal is dit net andersom.

Als we nu ook nog weten dat een positieve en een negatieve lading elkaar aantrekken en twee dezelfde ladingen elkaar afstoten dan kunnen we kort de werking van de component verklaren.

E: Emitter
B: Basis
C: Collector



Wanneer we op de basis een potentiaal verschil aansluiten, worden de elektronen uit de emitter zo sterk aangetrokken door de positieve deeltjes aan de basis en collector dat ze door de sperlaag breken. Het positieve deeltje aan de collector zorgen ervoor dat de elektronen ook door de 2de sperlaag breken en naar de sterk positieve collector vloeien. Slechts een klein gedeelte van de elektronen vloeit naar de basis (1 à 5%). Het merendeel gaat naar de collector. De basisstroom (basis-emitterspanning) regelt de veel grotere

collectorstroom: stroomgestuurde stroombron. Wanneer de basisstroom gelijk is aan nul dan is de collectorstroom gelijk aan nul en de collector-emitterspanning gelijk aan de voedingsspanning. Indien de basisstroom niet gelijk is aan nul dan is de collectorstroom niet nul en de collector-emitterspanning kleiner dan de voedingsspanning.

10.2 De transistor als schakelaar

Wanneer we de transistor als schakelaar bespreken moeten we rekening houden met de uiterste werkpunten (A en B) van de karakteristiek. De karakteristiek duidt eveneens de werkgebieden van de transistor aan.

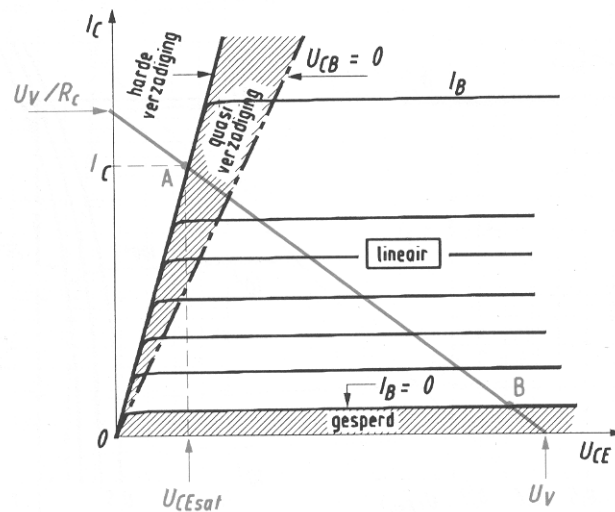


Fig.8-5: Werkgebieden van een transistor.

Je kunt op deze karakteristiek aflezen welke Collector-Emitterspanning en Collectorstroom hoort bij bepaalde basisstromen. Wanneer we in het verzadigingsgebied (m.a.w. sperlaag afgebouwd) werken, werkt de transistor als een open schakelaar. Werken we in het spergebied dan gedraagt de transistor zich als een gesloten schakelaar.

Op onderstaande figuur is de werking van de transistor als ideale schakelaar uitgelegd a.d.h.v. een schema.

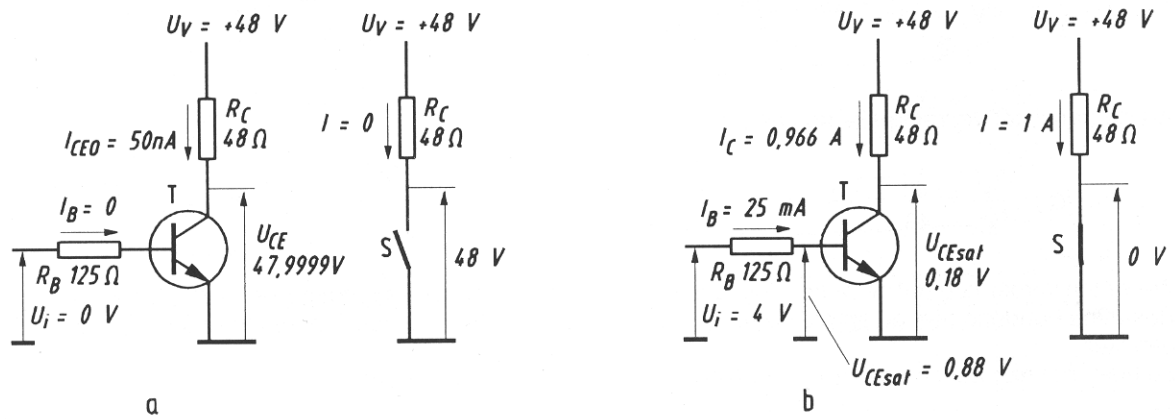
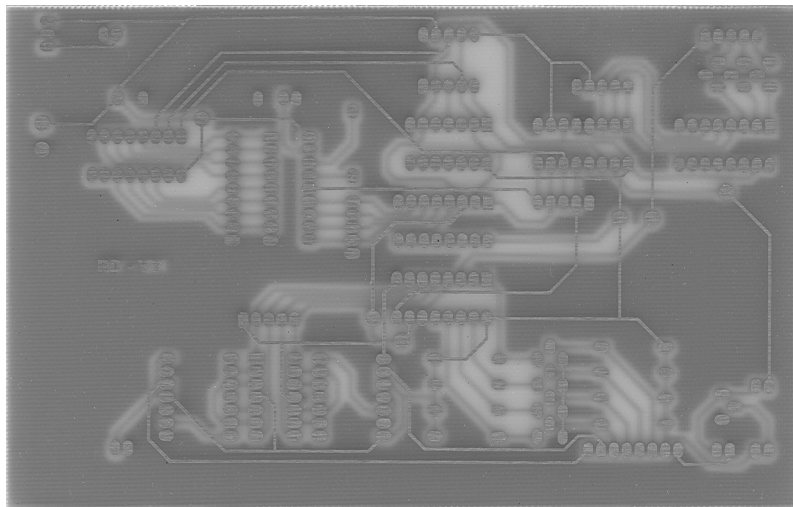


Fig.8-6: Niet-ideaal gedrag van de transistor als schakelaar.

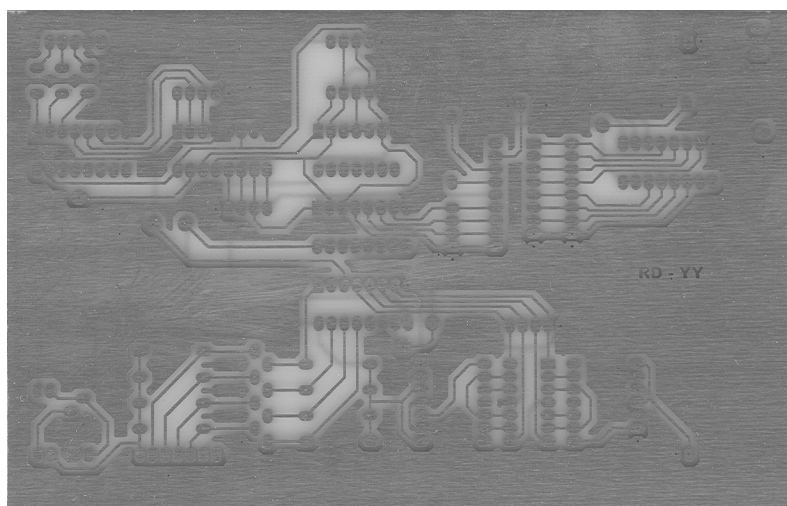
De linker figuur stelt een open schakelaar voor. Wanneer I_B (basisstroom) = 0 dan is I_C (collectorstroom) = 0 en U_{CE} (collector-emitterspanning) = U_{CC} (voedingsspanning).

De rechter figuur stelt een gesloten schakelaar voor. Wanneer I_B (basisstroom) = 25mA dan is I_C (collectorstroom) = 1A en U_{CE} (collector-emitterspanning) = 0V.

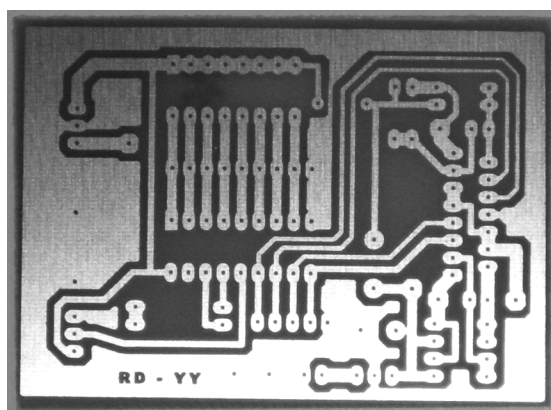
11 Bijlage 3: Illustraties



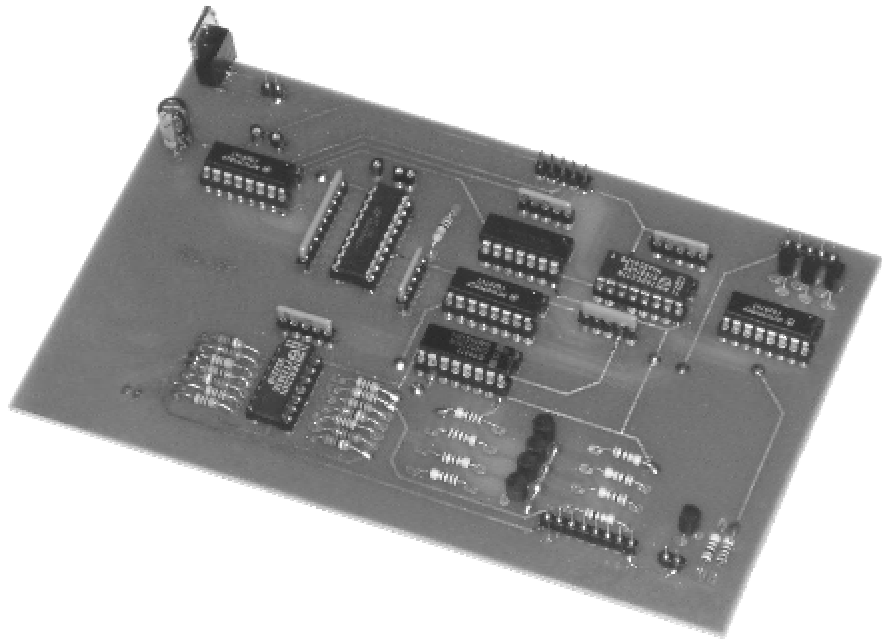
Figuur 12: Bovenzijde zendmodule



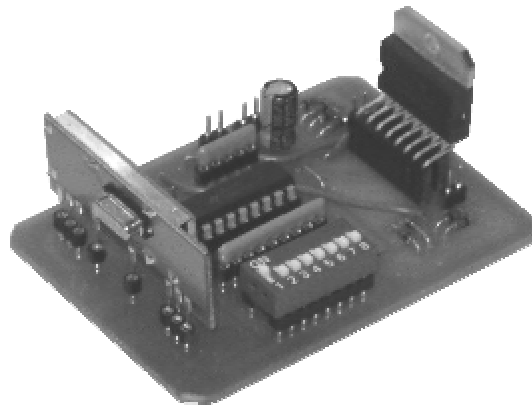
Figuur 13: Onderzijde zendmodule



Figuur 14: Ontvangmodule



Figuur 15: Afgewerkte zendmodule



Figuur 16: Afgewerkte ontvangmodule



Figuur 17: Racewagen



Figuur 18: Zendmodule

12 Dankwoord

Voor deze verhandeling zouden we graag enkele mensen willen bedanken. Vooral door hun steun en hulp is dit eindwerk tot een goed einde gebracht.

Dank u:

- dhr. Boterberge Jurgen en dhr. Deschepper Geert, voor de begeleiding tijdens het tot stand brengen van zowel het praktische gedeelte als het theoretische gedeelte. Zij leenden ons ook informatieve boeken en stelden een computer ter beschikking;
- Yde Johan, voor het leveren van componenten, het vervaardigen van de printplaten en ook de begeleiding en steun die we kregen;
- Yde Kevin, voor het solderen van de componenten op de printplaten;
- D'Hulst Jan, voor het vervaardigen van de metalen onderdelen van de zendmodule en de racewagen;
- Gouwy C., voor de uitleg betreffende de OPAMP;
- En aan onze ouders voor hun geduld en de tijd die ze voor ons hadden.

13 Logboek

30 augustus '03	<p>Opzoeken van informatie in verband met de seriële poort. Componenten gaan kopen in het winkeltje Telehome te Torhout. Schakeling bouwen om 8 LED's (uitgangen) aan te sturen via de seriële poort. Voorbeeld programma downloaden van Internet en laten uitvoeren op de gebouwde schakeling.</p>
31 augustus '03	<p>Het programma begrijpen en aanpassing doen om te testen. Zelf programma's schrijven voor de schakeling.</p>
6 september '03	<p>Schakeling om de 8 LED's (uitgangen) aan te sturen wat verbouwen + programma wijzigen. Schakeling bouwen om 8 drukknoppen (ingangen) in te lezen via de seriële poort. Met de kennis van de seriële poort, zelf een programma schrijven voor de drukknoppen (ingangen). Er doken enkele problemen op.</p>
7 september '03	<p>Oplossing gevonden voor het programma van de ingangen. De software van de ingangen combineren met de uitgangen. Definitieve beslissing van het eindwerk. Namelijk het bouwen van een PC gestuurde racewagen. Uittypen van het GIP voorstel.</p>
13 september '03	<p>Probleem: het racewagentje hadden we graag draadloos gebouwd. Na heel wat onderzoekwerk van de voorbije week hebben we besloten te werken met infrarood. De reden hiervan is dat de schakeling van infrarood een heel stuk eenvoudiger is dan de schakeling met radiogolven. Op Internet hebben we enkele verschillende infrarood schakelingen gevonden. Wij hebben gekozen voor de schakeling die opgebouwd is met Holtek's. De chips zijn niet echt duur en moeten niet meer geprogrammeerd worden. Naar Telehome geweest maar daar verkochten ze niet onze gewenste componenten. Naar Elektro 8000 in Brugge geweest daar kregen we hetzelfde antwoord.</p>
14 september '03	<p>Het programma voor de seriële poort hebben we wat efficiënter gemaakt. Informatie gezocht over het bedienen van DC motoren. Niet echt iets interessants gevonden.</p>
20 september '03	<p>Dan maar gaan opzoeken in de Gouden Gids naar andere elektronica winkels. We zijn terecht gekomen in Oostende in het winkeltje Micro World. Die mevrouw liet ons weten dat ze die Holtek's kon bestellen. Een hele opluchting dus.</p>
27 september '03	<p>Naar Micro World geweest. Helaas de bestelde componenten waren nog niet binnen.</p>

4 oktober '03	Terug naar Micro World om onze bestelling te gaan ophalen. Alles was binnen behalve de infrarood module. We zijn dan verder gereden naar het winkeltje Gobin om daar dit componentje te kopen. We hebben de schakeling gebouwd maar het werkte niet. Na enkele testen zijn we tot de conclusie gekomen dat het ontvang gedeelte de verkeerde signalen opvangt.
5 oktober '03	Enkele programma's als oefening geschreven in Visual Basic 6.0. Voor het gebruik van listboxen. Verschillende subcode's schrijven zoals het toevoegen, wijzingen en verwijderen van gegevens in een listbox.
7 oktober '03	Afspraak maken met dhr. Boterberge om een oscilloscoop meting te mogen uitvoeren op een woensdag namiddag. We hadden graag eens die verkeerde signalen op beeld gezien en bestudeerd.
8 oktober '03	Samen met dhr. Boterberge naar de bibliotheek in de Hogeschool van Oostende. Daar hebben we informatie gevonden over systemen op radiofrequentie en op infrarood. We hebben enkele teksten verzameld over de seriële poort. We hebben ook een interessante motorschakeling gevonden die we kunnen gebruiken voor ons wagentje.
9 oktober '03	Opzoek gegaan naar informatie i.v.m radiofrequentie modules. We hebben gebeld naar Gobin voor informatie maar niemand nam op.
10 oktober '03	Opnieuw gebeld naar Gobin Maar ze begrepen niet echt wat we precies nodig hadden i.v.m radiofrequentie modules.
15 oktober '03	Metingen op de Holtek's uitgevoerd. Het ontvangen signaal gemeten, we zagen duidelijk dat er een stoor signaal aanwezig was. Uiteindelijk geen echte oplossing gevonden. Misschien was het component stuk. Dus hebben we gesloten om over te schakelen naar radiofrequentie.
18 oktober '03	We zijn naar Gobin gegaan en daar hebben we nog eens ons probleem uitgelegd. Voor we vertrokken hadden we op de website van Velleman bezocht om te weten welke RF-modules te verkrijgen zijn in België. Hiervan hebben we een lijst opgesteld van alle RF-modules die van toepassing kunnen zijn op ons eindwerk. Gelukkig was er één module die ze herkenden. Die hebben we dan maar besteld.
25 oktober '03 tot 2 november '03	Gans de vakantie hebben we ons toegespitst op de samenstelling van de schakeling. We hebben een schakeling ontworpen met veelzijdige functies. Bijvoorbeeld meerdere auto's bedienen via de pc. De mogelijkheid om het adres van een wagentje te wijzigen.
8 november '03	Naar Gobin geweest om de bestelde RF-modules te gaan ophalen. Helaas waren ze nog niet binnen.
10 november '03	Weer naar Gobin gegaan om de RF-modules op te halen. Gelukkig met succes. De Holtek's aansluiten op de RF-modules en uitproberen.

11 november '03	Schuifregisters aansluiten op de Holtek's en via de RF-modules de nodige data verzenden.
15 november '03	Beginnen aan de opbouw van de (definitieve) schakeling.
16 november '03	Testen van de gebouwde schakeling. Bij de eerste test reageerde er niets. Dit komt omdat we iets kleins over het hoofd hadden gezien. Namelijk het data signaal moet eerder in het schuifregister toekomen dan de klokpuls. Dit hebben we verholpen door een vertraging op te bouwen voor de klokpuls.
22 november '03	Er wordt een grote aanpassing van de schakeling verricht door één centrale schuifregister te koppelen aan de seriële poort en die ervoor zorgt dat alle componenten rondom bestuurd worden.
23 november '03	Het begin van het programma schrijven voor de gebouwde schakeling.
29 november '03	Wat oefenprogramma's, die van toepassing kunnen zijn in ons eindwerk, schrijven. Zoals het opslaan en openen van verwerkte gegevens in een zelf geschreven programma.
20 december '03 4 januari '04	Leren werken met het professionele tekenprogramma Protel 99. Het bouwen van bibliotheken in Protel. Schakelingen uittekenen in Protel en omzetten naar een printplaat. Het opzoeken van datagegevens over de joystick. Een schakeling ontwerpen voor de joystick aan te sluiten aan de seriële poort.
10 januari '04	Een totale ombouw van de schakeling. Verbeteringen en nog wat veranderen aan de schakeling.
11 januari '04	Alles opnieuw uitwerken in Protel.
20 januari '04	Bespreking van het eindwerk met dhr. Deschepper. Bestelde componenten gaan ophalen in Oudenaarde. Joystick schakeling met de gekochte Opamps bouwen. Zender module schakelen op een breadboard. (niet volledig af)
21 januari '04	Zender module afwerken. Connector voor de joystick voorzien van draden en schakelen aan de zender module. Ontvanger module schakelen op een breadboard. (enkel de shift register met de motor driver) Code schrijven voor de gebouwde schakeling. (niet volledig af)
22 januari '04	Code voor de modules afwerken. De code op de schakeling uittesten. Alles werkt perfect maar te traag. Er is 1,5 seconden nodig om joystick in te lezen en 1 seconde om data naar de racewagen te zenden. (Geen oplossing gevonden voor het probleem)

23 januari '04	Misschien een oplossing gevonden. We dachten aan de racewagen rechtstreeks te koppelen aan de joystick. Dus de computer uit de schakeling te halen. Want de computer is de oorzaak van de vertraging. Er wordt verder gezocht naar een gepaste schakeling. We komen tot de vaststelling dat het te gecompliceerd wordt.
24 januari '04	We dachten aan de schakeling van de joystick weg te laten om de vertraging te beperken. En dan de gamepoort van de computer gebruiken om de joystick in te lezen. Info zoeken omtrent de gamepoort. (Niet veel opgeleverd) Voorbeeld programma's op internet zoeken om de gamepoort te bedienen. (Enkele geprobeerd, is niet echt gelukt, telkens errors)
26 januari '04	Onze nieuwe oplossing is het leren werken met Delphi om een '.dll' te maken. We zijn te weten gekomen dat een '.dll' sneller kan werken dan een gewone '.exe'.
27 januari '04	Naar de bibliotheek om cursussen te vinden over Delphi. Tot te conclusie gekomen dat het iets te lang gaat duren vooraleer we kunnen werken met Delphi. De overgang van VB naar Delphi is te groot.
28 januari '04	Bespreking van het eindwerk met dhr. Boterberge Vandaag ontdekt dat het mogelijk is om met VB ActiveX DLL's te maken. Eens iets geprobeerd op het gevoel. De dll bouwen leverde niet veel problemen op, maar het aanspreken van de dll lukte niet.
29 januari '04	Het component 7805 een nieuwe vorm gegeven in de bibliotheek van het tekenpakket Protel 99. De schakeling wat verbouwen in Protel 99.
30 januari '04	Ontvanger module beginnen uittekenen voor de printplaat. Printplaten tekenen gaat vrij traag omdat de componenten zo moeten geplaatst worden dat de baanafstand zo kort mogelijk is. De bedoeling is ook zo weinig mogelijk overbruggingen te maken.
31 januari '04	Een 2 ^{de} breadboard gebruiken om de ontvanger-module volledig uit te bouwen.
1 februari '04	Printplaat van de ontvanger definitief afwerken in Protell.
2 februari '04	Denken aan een gebruiksvriendelijk programma. Misschien een soort tekenprogramma.
3 februari '04	Naar Davotronics in Oudenaarde gaan om de bestelde componenten op te halen. Jammer ze zijn nog niet binnen
4 februari '04	De Opamp 5234 wordt helaas niet meer gemaakt. Opzoek gaan naar een equivalent.
6 februari '04	Naar Davotronics in Oudenaarde bellen. Nog niet binnen.
7 februari '04	Naar Elektro in Brugge gaan voor 7432. Te weten gekomen hoe men een ActiveX DLL moet aanspreken. De printplaat voor de racewagen (ontvanger) etsen.

8 februari '04	ActiveX DLL gemaakt van alle code. Tot de conclusie gekomen dat de snelheid niet gestegen is. We gaan toch de '.dll' blijven gebruiken omdat nu alle code verpakt is in 1 bestand. De totale code zit er overzichtelijker uit omdat de code van de '.dll' niet meer zichtbaar is in de '.exe'.
9 februari '04	De motor driver eens uit getest met Led's. Tot de conclusie gekomen dat het component stuk is.
10 februari '04	Opzoek gaan naar de grote vertraging in de code. Heel de code hebben we uit elkaar gehaald en de vertragingstijden gemeten. Probleem was dat door de combinatie van alle codes de opdracht regel OPENCOM zich 283 keer liet afspelen per gevraagde in de '.dll'. De opdracht regel vraagt tamelijk veel tijd. Het is slechts een paar milliseconden maar maal 283, loopt de duurtijd snel op. We hebben de code heel wat moeten verbouwen om het aantal maal te beperken. Nu loopt alles perfect behalve de data komt niet aan in de racewagen. Naar Davotronics in Oudenaarde gaan voor de bestelde componenten.
11 februari '04	Met de nieuwe motor driver draaien de motoren perfect. Doordat de Holtek's trager werken dan het programma is de reden dat de data van de racewagen niet verwerkt wordt. De oplossing is een vertraging in de code te plaatsen om de Holtek's de kans te geven te verzenden en ontvangen. Door middel van proefneming zijn we tot een vertragingstijd van 300 ms gekomen.
12 februari '04	De '.dll' nog wat verder optimaliseren met de nieuwe veranderingen. De volledige schakeling testen. De racewagen werkt, maar veel te traag. Na de Holtek's sturen we nog een schuif register aan. Die heeft 9 pakketten data nodig om aangestuurd te kunnen worden. Dus dit wordt dan 9 x 300 ms dat is bijna 3 seconden. We hebben besloten om het schuifregister uit de schakeling te halen. We sluiten de motor driver rechtstreeks aan de Holtek. Zo kunnen we met 1 pakket data de motoren bedienen.
13 februari '04	Doordat de schakeling is gewijzigd, moet de code ook aangepast worden. De printplaat voor racewagen moeten we opnieuw maken. Dus de print layout moet nu volledig opnieuw getekend worden.
14 februari '04	Ontvanger module opnieuw teken in Protel met de nieuwe schakeling
15 februari '04	Teken programma beginnen bouwen om de code grafisch te kunnen maken.
17 februari '04	Een Zoom functie maken voor het tekenprogramma
18 februari '04	De schakeling in school getest. Het heeft eventjes gewerkt en dan was het gedaan. Vermoedelijk een geleider die losgekomen is. Een raster functie maken voor het tekenprogramma
19 februari '04	Een Move functie en een Autosnap functie maken.
20 februari '04	Schakeling herstellen Selecteer functie voor het tekenprogramma maken.

21 februari '04	Bibliotheek van Protel veranderen. Alle soldeervlakjes van de printplaat waren te klein getekend. Het zou onmogelijk zijn om te solderen. Enkele correcties maken aan het schema van de racewagen en de print layout veranderen. Schema van de zendmodule verbouwen + het bijhorende code veranderen
22 februari '04	Beginnen aan de bespreking van de schakeling
23 februari '04	De print layout van de zendmodule is vrij gecompliceerd. we proberen een structuur op de printplaat te krijgen. We besluiten om op een dubbelzijdige printplaat te werken omdat er anders te veel overbrugging nodig zijn.
24 februari '04	Componenten op de printplaat van de zendmodule goed plaatsen
25 februari '04	Opzoek gaan naar informatie over een display. Die hadden we graag gekoppeld aan de zendmodule.
26 februari '04	De componenten nog wat verdraaien van printplaat 1. Nog een connector voorzien om de display aan te sluiten.
27 februari '04	Programma code maken om de functies van de display te kunnen bedienen. Print layout van de zendmodule beginnen bouwen. Weerstand nauwkeurig plaatsen om gemakkelijker te kunnen tekenen.
28 februari '04	De nieuwe printplaat van de racewagen boren. Racewagen van een controle Led en een aan uit schakelaar. Het overtollige plastic van de racewagen uit frezen om plaats te maken voor de printplaat.
29 februari '04	Print layout van de zendmodule afwerken. Heel de dag goed doorgewerkt.
1 maart '04	Bestellijstje faxen naar Davotronics in Oudenaarde. Printplaat van de zendmodule etsen.
2 maart '04	Splash screen maken voor onze applicatie. Problemen met de timer functie.
3 maart '04	Dil switch bouwen in VB en er een ocx file van maken.
6 maart '04	Fout in de Splash screen gevonden.
7 maart '04	Een led in Autocad tekenen en verwerken in VB en er dan nadien weer een ocx van maken.
9 maart '04	Bestelling in Oudenaarde ophalen. Helaas nog niet alles binnen.
11 maart '04	Racewagen.dll ombouwen naar een Racewagen.ocx. Omdat hij dan gemakkelijker aanspreekbaar is.
12 maart '04	Racewagen.ocx testen. De property functies werken niet zoals het hoort te werken. Vermoedelijk is het de code die de motoren aanstuurt die telkens blijft hangen.

13 maart '04	Printplaat van de zendmodule boren. Verder werken aan de uitleg van de schakeling. We stellen de bespreking van de code nog wat uit omdat die nog kan gewijzigd worden.
14 maart '04	Componenten voor de printplaten verzamelen.
16 maart '04	Module 1 en 2 fluxen
17 maart '04	Connector voor flatkabel kopen in Oudenaarde
18 maart '04	Hoek en Afstand programmeren → enkele fouten gevonden
19 maart '04	Afstand code werkt perfect Hoek bevat nog altijd fouten
22 maart '04	Componenten op de printplaten plaatsen en solderen
23 maart '04	Kleine fout ontdekt in module 2 → 2 draden gewisseld Connectoren aanbrengen op de draden van de COM poort en gamepoort Module 1 testen op fouten → geen gevonden
10 april '04	Beginnen aan een eigen install file
11 april '04	Verder werken aan die file
12 april '04	Beginnen aan een nieuw teken programma Het uitzicht beginnen bouwen
13 april '04	Rastercode en Zoomcode invoegen
14 april '04	Resize functie van de form optimaliseren Er ontstaat een probleem met de ingevoegde toolbars
15 april '04	Resize functie weglaten omdat de fout niet echt te herstellen valt
16 april '04	Maken van de code om een lijn met de muis te tekenen
17 april '04	Eens een grote test uitgevoerd → veel fouten ontdekt → we hebben ingezien dat het maken van een tekenprogramma in Visual Basic iets te ver gaat. Het is wel mogelijk, maar we hebben gebrek aan tijd.
21 april '04	Een nieuw ontwerp van een gebruikersprogramma. Namelijk de gebruiker de mogelijkheid geven om lijst te maken van instructies dat de racewagen moet uitvoeren. Enkele schetsen op papier maken. Het globale werk proces uitschrijven.
22 april '04	Het programma vorm geven op de computer. En krijgt de naam CGRPlayer. De code voor de knoppen Up en Down schrijven.
23 april '04	Het schrijven van de code om een bestand te lezen en te schrijven. Het bestand krijgt als extensie .cgr
24 april '04	Module 3 en 4 boren en fluxen
25 april '04	De code invoegen, wijzigen, wissen schrijven
26 april '04	Play functie schrijven
27 april '04	Afwerken van de code Play

28 april '04	Fout van de code Kill bestand verbeteren
29 april '04	Een vervang code zoeken voor Command → niets gevonden
30 april '04	Het tabblad Instellingen uit het programma verwijderen De instellingen in het frmMain tabblad verwerken
1 mei '04	Browse kader in setup verbeteren Snelkoppeling programmeren naar Alle Programma's Uninstaller bouwen
2 mei '04	Module 3 en 4 solderen Display monteren + Module 3 testen + code wat optimaliseren
3 mei '04	Extensie van Uninstaller.exe en CGRPlayer.exe wijzigen naar .dat
5 mei '04	Module 4 testen
6 mei '04	Racewagen testen met de computer van school Oscilloscoop meting gedaan op de HolteKE
8 mei '04	Plaatje ontwerpen om op de barst van de racewagen te kleven Scharnier voor batterijhouder kleven aan de racewagen
9 mei '04	Ontwerp batterij deksel
11 mei '04	frmMain uitleggen
12 mei '04	frmMain uitleg afwerken
13 mei '04	frmAbout uitleggen
14 mei '04	frmJoy uitleggen
18 mei '04	Foto's van eindwerk maken
21 mei '04	Plaatjes plakken aan racewagen Randen van voorwielhouders afslijpen
22 mei '04	Gat boren voor batterij deksel Racewagen volledig in elkaar monteren
23 mei '04	Antenne bevestigen