

Positiesturing van een stappenmotor met een microprocessor

door Thomas Langewouters
Thesisbegeleider: ing. Bart Vervecken
Richting Elektriciteit - Elektronica
Schooljaar 2003–2004

Positiesturing van een stappenmotor met een microprocessor

door
Thomas Langewouters

Richting Elektriciteit - Elektronica

Schooljaar 2003–2004

Thesisbegeleider: ing. Bart Verweken

Abstract

In vele toepassingen moeten belastingen of objecten in een nauwkeurige positie worden gebracht. De conventionele AC- en DC motoren hebben echter een belangrijk nadeel, het is namelijk moeilijk om de positie en de snelheid van de motor nauwkeurig te bepalen.

In deze GIP heb ik een studie uitgevoerd naar de werking en opbouw van stappenmotoren. Dit zijn elektromotoren waarmee gecontroleerde verplaatsingen kunnen worden gemaakt. Ook heb ik de werking van een microcontroller bestudeerd, en met de opgedane kennis een positiesturing gebouwd die de stappenmotor aanstuurt.

Als afsluiter heb ik ook een didactische opstelling gemaakt die de stappenmotor en een controllerprint bevat.

Dankwoord

Graag zou ik al mijn leerkrachten bedanken voor hun inzet tijdens mijn opleiding. De kennis en ervaring die ze me tijdens de voorbije jaren op het Technisch Instituut Scheppers op boeiende wijze hebben bijgebracht en de vele uren die we samen hebben doorgebracht zal ik niet vlug vergeten.

In het bijzonder zou ik meneer Vervecken willen bedanken voor de begeleiding van dit eindwerk, en de externe juryleden, die kostbare tijd hebben vrijgemaakt om de evaluaties bij te wonen.

Ook zou ik graag de vele mensen bedanken die informatie over stappenmotoren en microcontrollers ter beschikking hebben gesteld op het internet (zie literatuurlijst).

Thomas Langewouters

Lijst van afkortingen

μC : Microcontroller

μP : Microprocessor

PROM: Programmable read-only memory

EPROM: Erasable read-only memory

EEPROM: Electronically erasable read-only memory

I²C: Inter-IC bus

PIC: Programmable Intelligent controller

USB: Universal serial bus

HLL: High level language

BASIC: Beginner's All Purpose Symbolic Instruction Code

CISC: Complex Instruction Set Core

RISC: Reduced Instruction Set Core

UART: Universal asynchronous receiver/transmitter

MOSI: Serial Input

MISO: Serial Output

SCK: Serial Clock

Inhoudsopgave

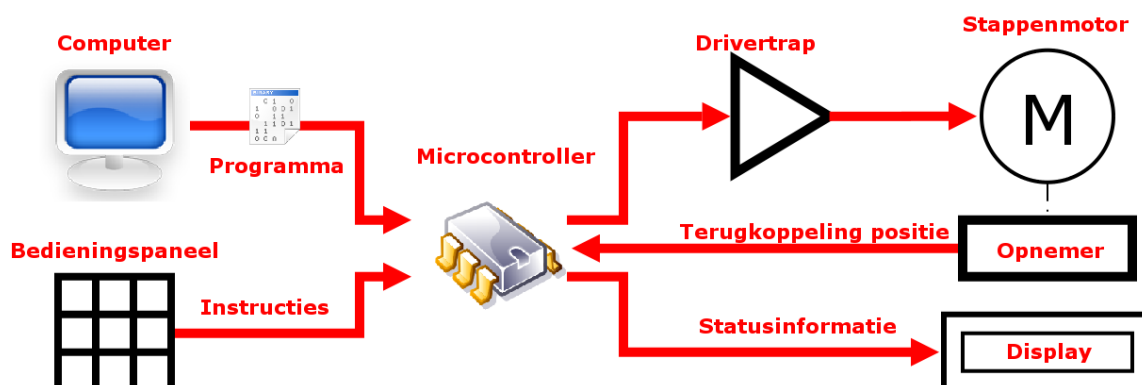
1	Blokschema	1
2	De stappenmotor	3
2.1	Inleiding	3
2.2	Principiële werking	5
2.3	Half/volstapbedrijf	7
2.4	Uitvoeringsvormen van de rotor	7
2.4.1	De permanente magneet stappenmotor	8
2.4.2	De variabele reluctantie stappenmotor	9
2.4.3	De hybride stappenmotor	10
2.5	Uitvoeringsvormen van de wikkelingen	11
2.6	De motorkarakteristieken	13
2.6.1	De koppel-hoek karakteristiek	13
2.6.2	De koppel frequentie karakteristieken	15
2.7	Het inschakelen van de fasen	17
2.8	Resonanties en instabiliteiten	19
2.9	Demping	19
2.9.1	Mechanische demping	20
2.9.2	Elektrische demping	21
2.10	Stapmodi	21
2.10.1	Volstapbedrijf	21
2.10.2	Halfstapbedrijf	22
2.10.3	Mini-stap bedrijf	22
2.11	Mijn motor	23

3	De Microcontroller	25
3.1	Inleiding	25
3.2	Soorten microcontrollers	26
3.3	Keuze van de μC	26
3.4	Talen	27
3.5	Features van de AT90S2313	28
3.6	Schakelingen met de μC	29
3.6.1	Basisschakeling	29
3.6.2	Aansluiten van schakelaars en led's	29
3.6.3	In System Programmer	30
3.6.4	Communicatie met de pc via RS-232	32
3.6.5	De RS-232 standaard	32
3.6.6	RS-232 – TTL Interface	33
4	De aansturing van de motor	35
4.1	De drivertrap	35
4.2	Drivertrap met een ULN2004	35
4.3	De sub stap	36
5	Positiesturing	41
5.1	Principe	41
5.2	Positieterugkoppeling	43
6	De praktische uitvoering	45
6.1	De opbouw van de print	45
6.1.1	AT90S2313 μC	45
6.1.2	RS-232 interface	45
6.1.3	ISP connector	46
6.1.4	Voeding	46
6.1.5	Motordriver	46
6.1.6	Connectoren	46
6.2	Het bereiden van de print	46
6.3	De didactische opstelling	48

Hoofdstuk 1

Blokschema

Om een overzicht te krijgen van de opbouw van mijn GIP heb ik een eenvoudig blokschema getekend dat de onderdelen weergeeft die zich uiteindelijk zullen bevinden in de didactische opstelling waarheen ik in de loop van dit schooljaar heb gewerkt.



Figuur 1.1: Het principieel blokschema van mijn GIP

Het brein van de opstelling is een microcontroller, die volgens zijn programma de opdrachten van de computer verwerkt, en de motor aanstuurt.

In het eerste deel van deze scriptie heb ik de motor bestudeerd, vervolgens de microcontroller en de positiesturing. Het vinden van een geschikte μC was niet echt een probleem omdat ik goed thuis ben in deze materie. Omdat we op school reeds over ontwikkelbordjes voor de AT90S2313 μC beschikken, lag het voor de hand dat die werden gebruikt.

Ook heb ik de verbinding tussen de microcontroller en de pc bestudeerd, deze is essentieel bij het laden van het programma in de microcontroller en eventueel het debuggen ervan. Naast de programmeerpoort heb ik ook een seriële verbinding

voorzien op de print. Hierdoor wordt een bedieningspaneel overbodig omdat er via het keyboard van de PC commando's kunnen worden gegeven aan de μC . De bediening van de microcontroller kan echter ook worden uitgevoerd in de vorm van een toetsenbordje met schakelaars, en de uitlezing met een LC-display of LEDS. Hiervoor heb ik ook aansluitingen voorzien op de print.

Hoofdstuk 2

De stappenmotor

2.1 Inleiding



Figuur 2.1: Een stappenmotor

In vele toepassingen moeten belastingen of objecten in een nauwkeurige positie worden gebracht. Als men de gewone AC- of DC-motoren toepast is het moeilijk om de positie van de belasting, de snelheid van de motor of de afgelegde weg nauwkeurig te bepalen. Daarom moet men gebruik maken van hulpmiddelen zoals sensoren, encoders of servosystemen en reminstallaties. Stappenmotoren zijn een speciaal soort elektromotoren waarmee gecontroleerde verplaatsingen gemakkelijk te verwezelijken zijn. De as van een stappenmotoren kan namelijk door middel van elektrische impulsen heel nauwkeurig geroteerd worden. Stappenmotoren kunnen doorgaans geen groot koppel leveren, en worden daarom meestal toegepast waar een nauwkeurige positionering nodig is, maar geen zwaar werk moet worden verricht. De werking van de stappenmotor omvat meerdere vakgebieden: de elektromechanica, elektronica, regeltechniek, digitale techniek en informatica. Door deze samenhang neemt hij dan ook een speciale plaats in tussen de andere elektromotoren, en is hij een uitstekende kandidaat om als onderwerp van een geïntegreerde proef te dienen.

Stappenmotoren hebben dankzij hun eigenschappen enkele voordelen t.o.v. gewone motoren:

- de motor en aansturing zijn goedkoper
- ze zijn gemakkelijk met de computer aan te sturen
- in onbekrachtigde toestand kan er een koppel beschikbaar zijn om de rotor op zijn plaats te houden
- de motor neemt niet veel plaats in
- een positieopnemer is niet noodzakelijk (per puls zal de rotor een bepaalde hoek verdraaien)

Stappenmotoren hebben echter ook enkele nadelen waar zeker rekening mee moet worden gehouden:

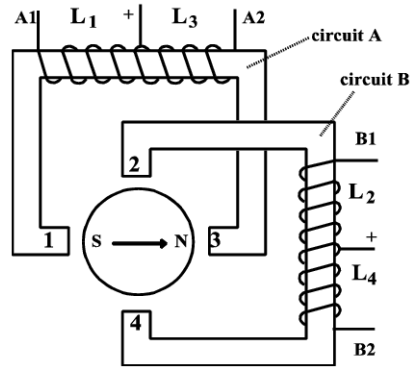
- de motor heeft een klein koppel per volume
- de snelheid is beperkt en afhankelijk van de mechanische belasting
- er kunnen resonanties en instabiliteiten optreden als hij niet goed aangestuurd wordt
- de nauwkeurigheid van de positie is afhankelijk van de stapgrootte en het lastkoppel.

Omwille van deze eigenschappen wordt de stappenmotor onder andere gebruikt in volgende toepassingen;

- kopbesturing van printers en kopieermachines
- kopbesturing van magnetische en optische schijfstations.
- aandrijving van robotarmen
- mespositionering in snijmachines
- het richten van camera's en telescopen

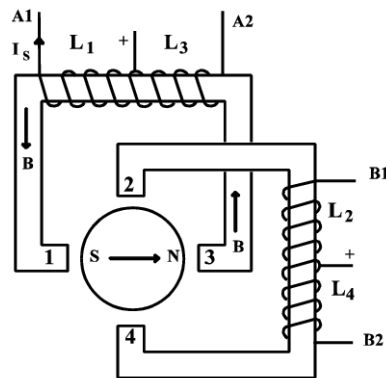
In dit hoofdstuk ga ik de verschillende soorten stappenmotoren, de werking en de aansturing bespreken.

2.2 Principiële werking



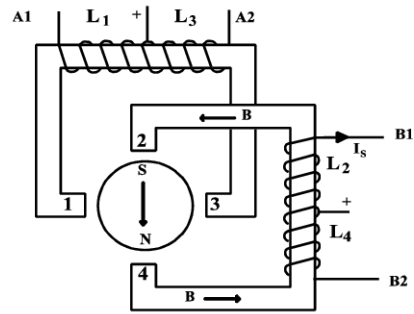
Figuur 2.2: PM stappenmotor met 2 polen

De werking van de stappenmotor ga ik uitleggen aan de hand van de PM-motor (een stappenmotor opgebouwd uit een tweepolige permanente magneet op de rotor en een stator met twee loodrecht op elkaar magnetische circuits) die getoond is in figuur 2.2. De wikkelingen A_1 - A_2 en B_1 - B_2 vormen elk een elektromagnetisch circuit. Elk



Figuur 2.3: L_1 is bekrachtigd

magnetisch circuit bestaat uit twee lichamelijke polen en twee deelwikkelingen (fasen genoemd). De rotor zal een bepaalde stand gaan innemen waarbij de magnetische weerstand voor de krachtlijnen zo klein mogelijk is. De magnetische polen van de rotor komen dus tussen polen (1, 3) of (2, 4) te staan. Deze vier standen noemt men voorkeurstanden en zullen ook worden ingenomen bij bekrachtiging van de fasen. De motor heeft dus vier stappen per omwenteling. Bekrachtigen we nu L_1 dan zal pool 1 een noordpool worden, en pool 3 een zuidpool, de rotor zal dan verdraaien en de stand innemen die in figuur 2.3 wordt getoond. Als we vervolgens fase L_2 bekrachtigen, zal de rotor verdraaien onder een hoek van 90° (zie figuur 2.4). Bij

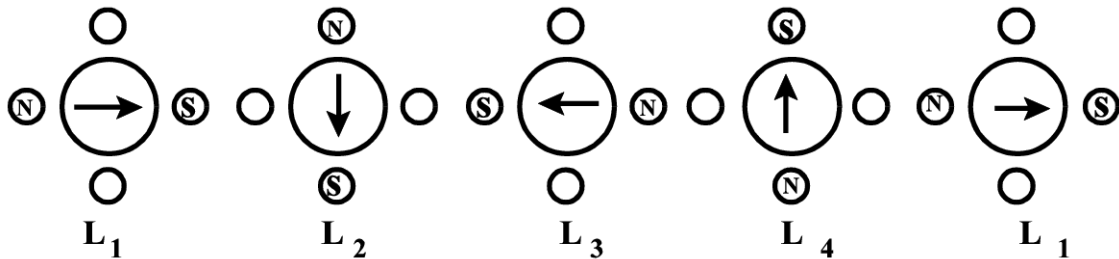


Figuur 2.4: L_2 is bekrachtigd

het afwisselend bekrachtigen van fasen L_1 , L_2 , L_3 , L_4 zal de rotor in plaats van te stappen overgaan in draaien. De draairichting kunnen we omkeren door de volgorde van bekrachtigen van de fasen te veranderen. Er zijn dus (minimaal) vier fasen nodig om de stappenmotor te laten draaien.

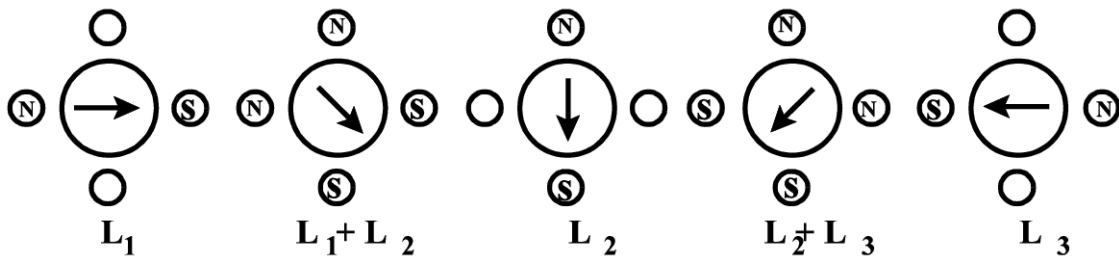
2.3 Half/volstapbedrijf

Het om beurten bekrachtigen van de fasen noemt men volstapbedrijf. (figuur 2.5)



Figuur 2.5: Volstapbedrijf

De staphoek in volstapbedrijf voor deze motor is 90° . Als we nu afwisselend 1 fase en 2 fasen tegelijk gaan bekrachtigen, krijgen we halfstapbedrijf, zoals te zien is in de figuur 2.6.

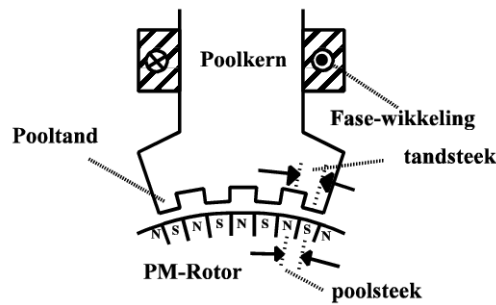


Figuur 2.6: Halfstapbedrijf

2.4 Uitvoeringsvormen van de rotor

Stappenmotoren bestaan uit een stator waarop fasewikkelingen zijn aangebracht en een rotor zonder wikkelingen. De constructie van de rotor zorgt echter ook voor onderscheiding tussen de stappenmotoren. Er zijn drie soorten rotoruitvoeringen:

- met een permanente magneet
- met een ijzeren rotor (principe van de variabele reluctantie)
- een combinatie van een PM en ijzer (hybride stappenmotoren)



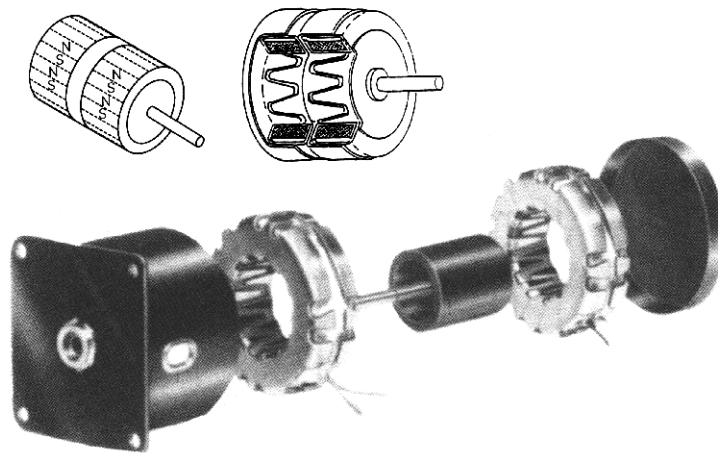
Figuur 2.7: Opbouw van een statorpool met pooltanden

2.4.1 De permanente magneet stappenmotor

De rotor van de PM-motor bestaat uit permanent magnetisch materiaal, en de stator is opgebouwd uit twee onder elkaar op de as aangebrachte permanente magneten. De grootte van de staphoek wordt bepaald door het aantal poolparen op de rotor en het aantal lichamelijke polen op de stator. De statorpolen kunnen nog van tanden worden voorzien waardoor het koppel groter wordt. (zie figuur 2.7) De poolkern met de fasewikkeling mondt uit in een viertal pooltanden van ijzer.

De tandsteek van de statorpool dient overeen te komen met de poolsteek van de rotor.

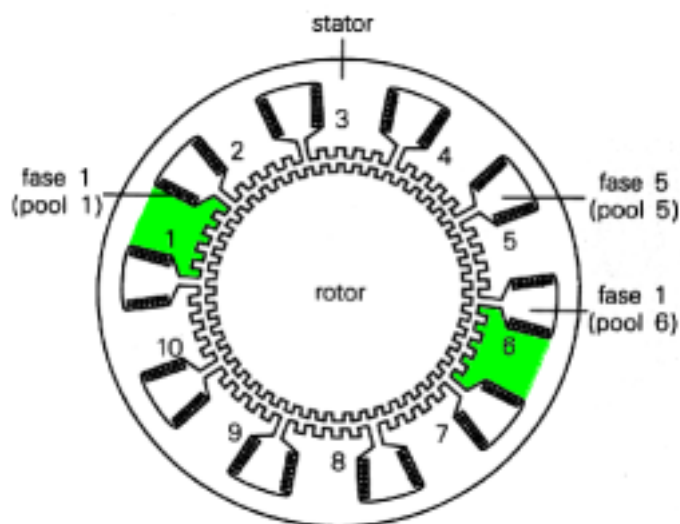
De pooltanden van 1 statorpool vormen te samen 1 voorkeursstand van de rotor. Het aantal stappen per omwenteling wordt bepaald door het product van het aantal polen van de rotor en het aantal statorfasen. In figuur 2.7 is de situatie getekend voor een bekrachtigde fasewikkeling. Voor een goede werking van een stappenmotor, dienen de tanden van verschillende statorfasen tov elkaar verschoven te zijn. Voor twee fasen moeten de tanden van de afzonderlijke statorfasen een halve tandsteek verschoven zijn, voor drie fasen een derde tandsteek etc. In figuur 2.8 is een open-gewerkte afbeelding van een veel voorkomende bipolaire PM motor getekend. Het statorijzer bestaat uit een boven- en onderdeksel en een middenstuk, waartussen zich 2 wikkelingen bevinden. Door de wikkelingen in 2 richtingen te bekrachtigen ontstaat er een vierfasige stappenmotor. De spoelen worden omgeven door deksels met tanden die elkaar niet raken maar wel in elkaar grijpen. Deze tanden komen overeen met de eerder genoemde pooltanden van de stator. Door de bovenste wikkeling te bekrachtigen kunnen alle tanden van het bovendeksel noordpolen en de bovenste tanden van het middenstuk zuidpolen worden. De rotor ziet dan afwisselend noord- en zuidpolen langs zijn optrek en zal dus een voorkeursstand innemen. Het onderste deksel werkt op dezelfde manier. De rotor van deze motor is opgebouwd uit twee onder elkaar op de as aangebrachte permanente magneten. De bovenste tanden van het middenstuk zijn t.o.v. die van de onderste tanden een halve poolsteek verschoven. Door de aanwezigheid van een permanente magneet is deze motor met een behoorlijk koppel zeer compact te bouwen. De prijs van deze motor is dan ook laag



Figuur 2.8: Opengewerkt model van een PM-motor

vanwege de eenvoudige constructie.

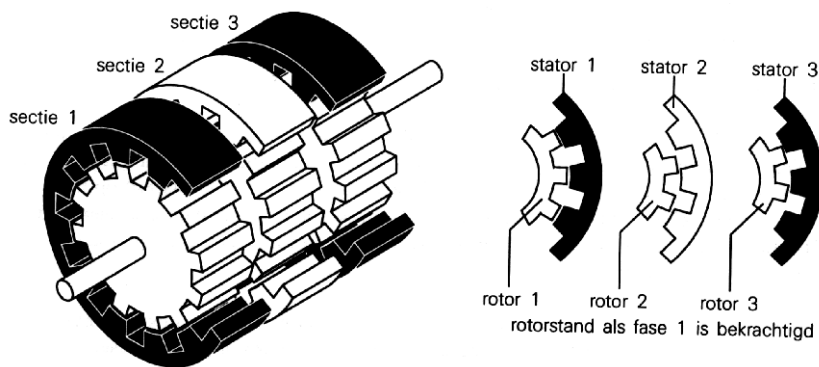
2.4.2 De variabele reluctantie stappenmotor



Figuur 2.9: Een vijffasige VR-motor

De rotor van de VR-motor bestaat uit vertand ijzer. De tanden zijn te vergelijken met de polen van de PM-rotor. De tandsteek van een statorpool dient overeen te komen met de tandsteek van de rotor. De koppelontwikkeling van deze motor is gebaseerd op het reluctantieprincipe. Volgens dit principe zal een tand van de rotor

zich richten naar een tand van een bekrachtigde statorpool. Door de vertanding van zowel stator als roter kan men een groot aantal stappen per omwenteling realiseren. De VR-stappenmotor heeft echter minimaal 3 fasen nodig om in twee richtingen te kunnen draaien. De reden hiervoor is dat een rotortand door een statorfase wordt gemagnetiseerd. De rotortand heeft geen vaste magnetische poolwaarde en daarom ook geen voorkeurstand t.o.v. een bekrachtigde statorpool. De draairichting wordt bepaald door de volgorde waarin de bekrachtiging van de fasen plaats vindt en niet door de polariteit van de bekrachtiging. Het aantal stappen per omwenteling wordt bepaald door het product van het aantal rotortanden en het aantal statorfasen. In figuur 2.9 is een doorsnede van een vijffasige VR stappenmotor getekend. De stator bestaat uit 10 lichamelijke polen, die per tegenoverliggend tweetal 1 fase vormen. De afstand tussen twee statortanden is zo gekozen, dat bij het na elkaar bekrachtigen van naast elkaar liggende fasen de rotor 1 stap uitvoert. Door combinatie van vijf fasen, de vertanding van de statorpolen en de vertanding van de rotor worden veel stappen per omwenteling gerealiseerd. Met dit type stappenmotor kan men tot duizenden stappen per omwenteling komen.

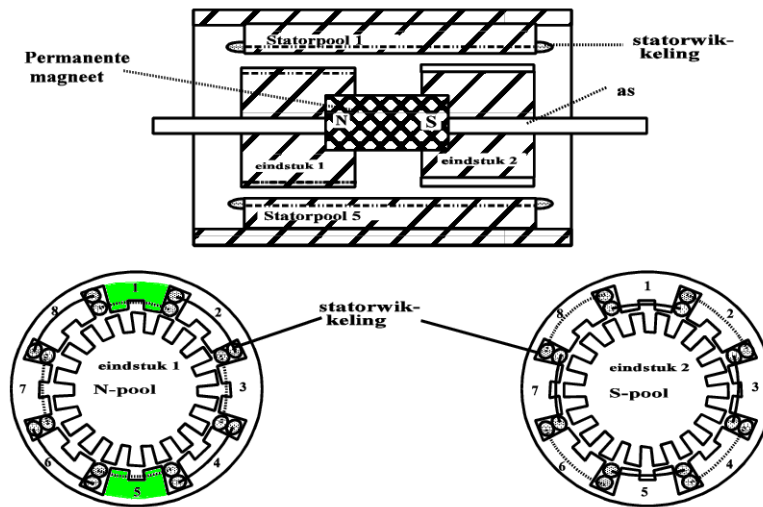


Figuur 2.10: Een VR-motor opgebouwd uit drie secties

Een andere uitvoering van een VR-stappenmotor bestaat uit een aantal secties. (zoals te zien in figuur 2.10) Elke sectie is opgebouwd uit een stator en een rotor. De stator van 1 sectie bevat meestal 1 fase. Het aantal secties ligt tussen 2 en 12. Voor een driefasige VR-motor dienen de rotor of statortanden uit een sectie t.o.v. elkaar $\frac{1}{3}$ tandsteek verschoven zijn.

2.4.3 De hybride stappenmotor

Dit type stappenmotor is een combinatie van de PM- en VR- stappenmotor. Hierdoor worden de goede eigenschappen van de twee typen verenigd. De rotor van een H-motor bestaat uit een in langsrichting gemagnetiseerde magneet met aan de polen vertande eindstukken van weekijzer. De tanden van de eindstukken zijn onderling

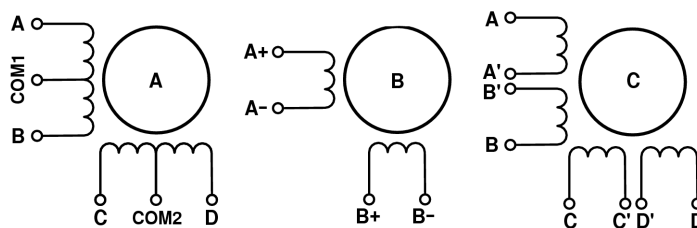


Figuur 2.11: De hybride stappenmotor

een halve tandsteek verschoven. De stator bestaat uit een aantal fasen met vertande polen. De statorpolen omvatten beide eindstukken. Het stappen in een bepaalde richting wordt gerealiseerd door in bepaalde volgorde de fasen te bekrachtigen. In figuur 2.11 is een doorsnede van een H-motor afgebeeld met 8 polen. De motor heeft verder 4 fasen en 2 tanden per statorpool. De poolnummers 1 en 5, 2 en 6, 3 en 7, 4 en 8 vormen samen de 4 fasen.

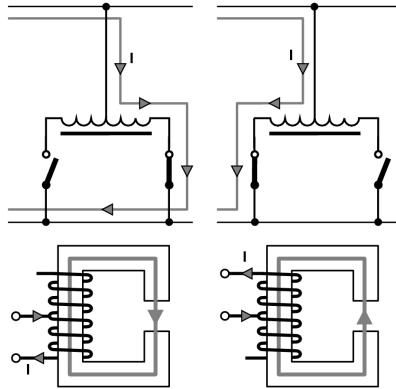
2.5 Uitvoeringsvormen van de wikkelingen

Stappenmotoren kunnen we in verschillende categorieën indelen. Zo kunnen we de stappenmotoren ook indelen volgens de constructiewijze van de statorwikkelingen. Er zijn drie manieren van wikkelen, namelijk *unipolair*, *bipolair* en *bifiliair* (zie figuur 2.12).



Figuur 2.12: De verschillende statorwikkelingen

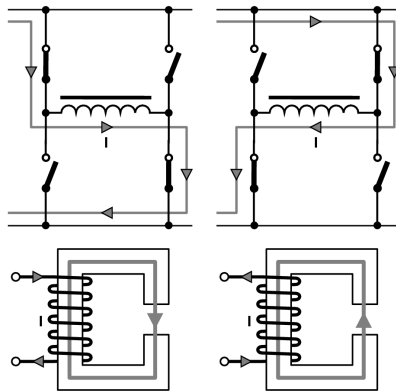
Unipolaire statorwikkelingen



Figuur 2.13: Principie"le aansturing van een fase van een unipolaire motor.

Bij een unipolaire motor zijn er twee wikkelingen per magnetisch circuit, waarvan 1 geleider gemeenschappelijk naar buiten wordt gebracht. (zie tekening A in fig. 2.12) Het bekrachtigen van de fasen van een unipolaire motor gebeurt zoals in fig. 2.13 is weergegeven.

Bipolaire statorwikkelingen



Figuur 2.14: Principie"le aansturing van een fase van een bipolaire motor.

Bij bipolaire motoren is er per magnetisch circuit slechts 1 wikkeling voorzien. Omdat elk circuit van polariteit moet kunnen omkeren, stuurt men de stroom in 2 richtingen door de wikkeling. Zo bekomt men de vier fasen die nodig zijn om de motor te laten draaien. (zie tekening B in fig 2.12) Om een wikkeling van een bipolaire motor te bekrachtigen maakt men gebruik van een H-brug (fig. 2.14). Door een H-brug toe te passen kan men de stroom in twee richtingen laten lopen.

Bifiliaire statorwikkelingen

Bifiliaire wikkelingen maken de stappenmotor universeel op het gebied van statorwikkelingen. Omdat voor ieder magnetisch circuit twee draden parallel zijn gewikkeld, kan men later, bij het ontwerpen van de toepassing, kiezen of men de motor als *bipolaire* motor of *unipolaire* motor gaat gebruiken, men hoeft enkel de motor op de gewenste manier aan te sluiten. (zie tekening C in fig 2.12)

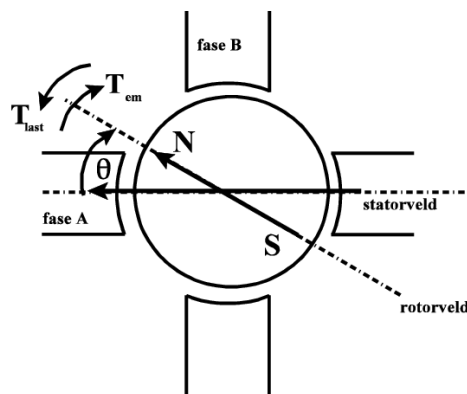
2.6 De motorkarakteristieken

Het gedrag van de meeste elektromotoren wordt hoofdzakelijk vastgelegd in de koppeltoerentekarakteristieken. Het elektromechanisch gedrag van de stappenmotor wordt echter gekenmerkt door twee anderssoortige motorkarakteristieken, de *koppelhoek karakteristiek* en de *twee koppelrequentie karakteristieken*.

2.6.1 De koppel-hoek karakteristiek

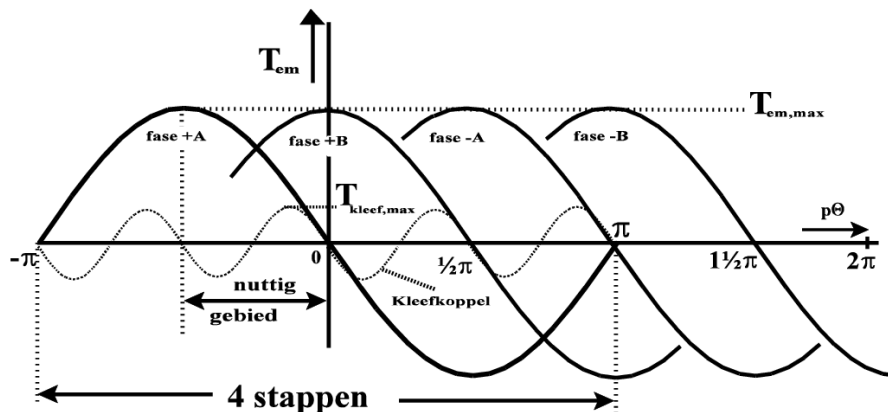
De koppel-hoek karakteristiek beschrijft het koppel T_{EM} als functie van de rotorhoekverdraaiing θ . De karakteristiek geeft het koppel gedurende n stap, als er n fase bekrachtigd is met een constante stroom. Deze karakteristiek wordt ook wel de statische motorkarakteristiek genoemd.

KH-karakteristiek bij PM-motoren



Figuur 2.15: een 2-fasige PM stappenmotor.

De koppel-hoekkarakteristiek wordt beschreven aan de hand van een bipolaire 2-fasige stappenmotor zoals die van figuur 2.15. Als we het koppel T_{EM} opmeten als

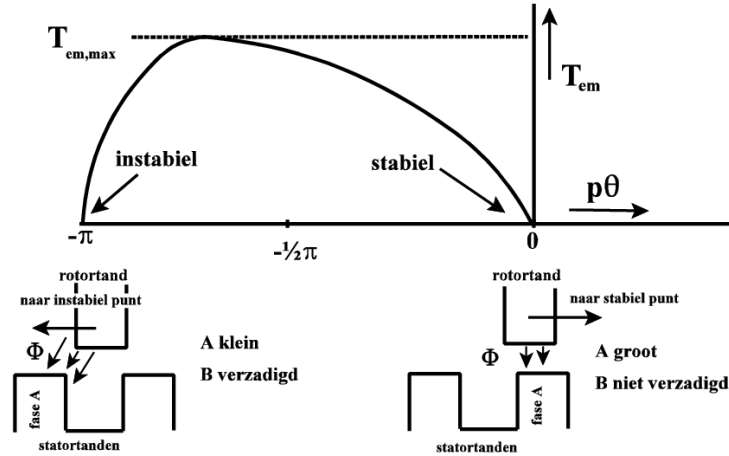


Figuur 2.16: De koppel-hoek karakteristiek van de PM stappenmotor.

functie van de hoekverdraaiing θ , dan krijgen we de statische koppelkarakteristiek van fase A. Het koppel T_{EM} is nagenoeg evenredig met $\sin(\theta)$. Als de stappenmotor meerdere poolparen (p) heeft, dan wordt T_{EM} logischerwijze evenredig met $\sin(p\theta)$. Het koppelverloop van een motor met p poolparen op de stator is voorgesteld in figuur 2.16.

Met fase +A wordt bedoeld dat fase A bekrachtigd is met een stroom in de richting van de wikkelzin van de fasespoel A. Wordt de fase bekrachtigd met dezelfde stroom, maar tegengesteld aan de wikkelzin, dan wordt deze aangeduid met fase -A. Als we een stap willen uitvoeren van $p\theta = \frac{1}{2}\pi$ naar $p\theta=0$, hebben we een positief koppel T_{EM} nodig tijdens de hoekverdraaiing. Hiervoor moeten we fase +A gedurende $-\frac{1}{2}\pi < p\theta < 0$ bekrachtigen. Als de rotor de positie $p\theta=0$ eenmaal bereikt heeft, dan is het koppel gelijk aan nul en blijft de rotor in deze positie staan. Fase +A heeft dus een 'stabiel' evenwichtspunt bij 0 elektrische graden. We kunnen hieruit besluiten dat alleen het gebied tussen $-\frac{1}{2}\pi < p\theta < 0$ te gebruiken is om een stap uit te voeren en leidt naar een stabiele eindpositie $p\theta=0$. Deze positie noemen we de nullast stand. Fase +A heeft een 'instabiel' evenwichtspunt bij 180 elektrische graden, omdat de rotor in deze positie op den duur altijd een kant op zal bewegen om in de positie $p\theta=0$ of $p\theta=2\pi$ te eindigen. De fasen -A, +B en -B hebben ook nullaststanden (bij resp. 180, 90 en 270 elektrische graden). Het houdkoppel T_{EM} is gelijk aan het maximale lastkoppel dat een bekrachtigde motor kan verplaatsen. Omdat de magneet van de rotor ook magnetisch is als de fasen niet zijn bekrachtigd, zorgt deze voor een kleefkoppel (het eigen houdkoppel T_{kleef}). Hierdoor heeft de rotor ook voorkeursposities in onbekrachtigde toestand. Dit kleefkoppel is periodiek met $2p\theta$ zoals te zien in figuur 2.16, waarin ook het verloop van het kleefkoppel van fase A getekend is. Het eigen houdkoppel T_{kleef} komt overeen met het maximale koppel dat op de as van een niet bekrachtigde motor mag worden uitgeoefend zonder dat er een stap uitgevoerd wordt. Omdat variabele reluctantiemotoren geen permanente magneet bevat, heeft deze logischerwijze geen kleefkoppel.

KH-karakteristiek bij VR-motoren



Figuur 2.17: De koppel-hoek karakteristiek van een VR stappenmotor.

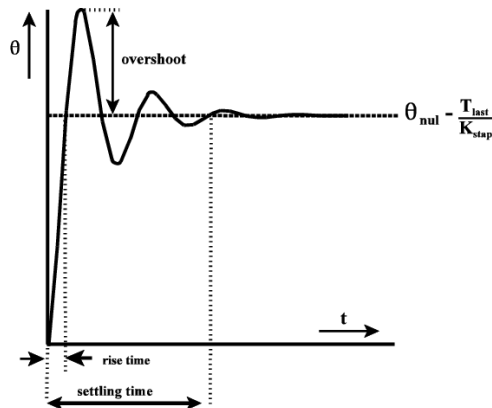
De koppelhoek karakteristiek van een VR-stappenmotor heeft een afwijkende vorm en heeft over het algemeen een hoger houdkoppel dan de PM-stappenmotor. De afwijking van de karakteristiek wordt veroorzaakt door een veranderende μ_r van het ijzer (door verzadiging) als de rotor zijn 'instabiel' evenwichtspunt nadert. (zie fig 2.17) Vergelijken we deze situatie met de nullaststand, dan is de waarde van de flux ongeveer gelijk, maar het bijbehorende oppervlak A is kleiner, waardoor de inductie B hoog is. Er treedt verzadiging in de tanden op, waardoor het koppel rond het instabiele punt sneller kleiner wordt. In figuur 2.17 is de koppel-hoek karakteristiek van een VR stappenmotor getekend.

2.6.2 De koppel frequentie karakteristieken

Twee koppelfrequentie karakteristieken beschrijven het koppel T_{EM} als functie van de stapfrequentie f_{stap} . De frequentie f_{stap} is de frequentie waarmee een stappenmotor de stappen uitvoert. Deze karakteristiek is vergelijkbaar met de koppeltoerental kromme van andere elektromotoren en worden de dynamische motorkarakteristieken genoemd. De aanlopfrequentie wordt weergegeven in de start-stop karakteristiek. Voor een reeds draaiende stappenmotor wordt de stapfrequentie weergegeven in de frequentiegrens karakteristiek.

De eerste stap

Het verloop van de eerste stap na stilstand wordt bepaald door de demping f_{visk} , de massa traagheid J , het lastkoppel T_{last} en het elektromagnetisch koppel T_{EM} van een

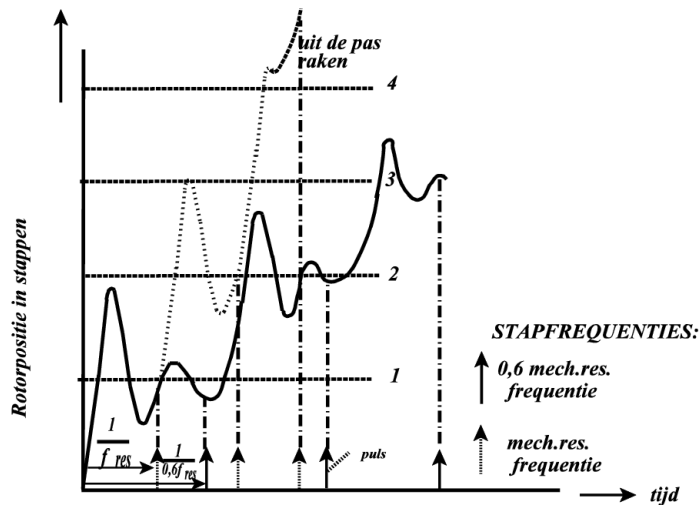


Figuur 2.18: De rotorpositie bij een enkele puls

stappenmotor. Als de stappenmotor een puls krijgt vanuit stilstand, zal de rotor slingerend de nullaststand bereiken zoals wordt getoond in figuur 2.18. Hierdoor mag men, als men meerdere pulsen na elkaar gaat uitvoeren, de pulsen niet te snel na elkaar laten volgen. Als de pulsen toch te snel na elkaar volgen, kan het zijn dat er een puls in de 'overshoot' wordt gegeven, en er stappen worden over geslagen. Dit fenomeen wordt getoond in figuur 2.19. Als de stapfrequentie ongeveer gelijk is aan de mechanische eigenfrequentie, raakt de motor uit de pas en kan er resonantie optreden. Daarom worden er grenzen opgelegd aan de stapfrequentie. In figuur 2.19 is te zien dat bij een stapfrequentie van $0,6f_{resp}$ de motor wel in de pas blijft.

De start-stop karakteristiek

Als we een stappenmotor starten, mogen we de stappen niet te snel op elkaar laten volgen. De start-stop karakteristiek geeft aan bij welke stapfrequentie een bepaalde last door de stappenmotor vanuit stilstand nog verplaatst kan worden. Omdat er geen beginsnelheid is, zal het verloop sterk door de massatraagheid J beïnvloed worden. Omdat T_{EM} tijdens een stap niet constant is, moeten we uitgaan van een gemiddelde waarde voor T_{EM} . Als we de demping f_{visk} verwaarlozen, dan is het versnellingskoppel alleen nodig om de massatraagheid te versnellen. Het versnellingskoppel is dan: $T_{acc} = T_{em,gem} - T_{last}$. De gemiddelde hoeksnelheid α is dan $\frac{T_{acc}}{J}$. Hieruit volgt dat bij toenemend lastkoppel T_{EM} de hoekversnelling α afneemt. Als het lastkoppel te groot wordt, zal de motor helemaal niet meer op gang komen. Als de motor niet belast is, is het hoogst mogelijke versnelkoppel beschikbaar. Deze frequentie noemen we $f_{stap,nul}$. In figuur 2.20 is de start-stop karakteristiek aangegeven. Bij een grotere massatraagheid schuift de start-stop karakteristiek naar links op. Ook zal bij toenemende demping de stapfrequentie afnemen omdat T_{acc} kleiner wordt en dus ook de hoekversnelling α .



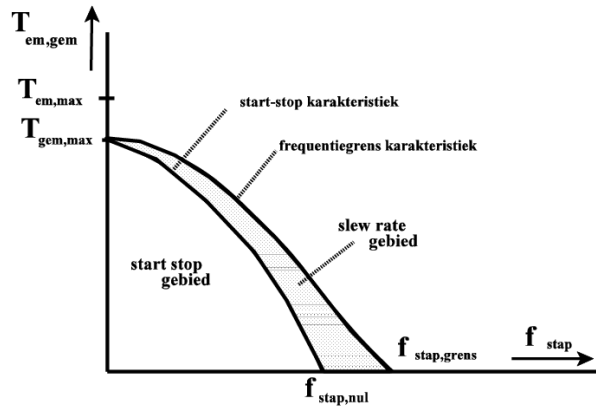
Figuur 2.19: Stapresponsies op f_{resp} en $0,6f_{resp}$

De frequentiegrens karakteristiek

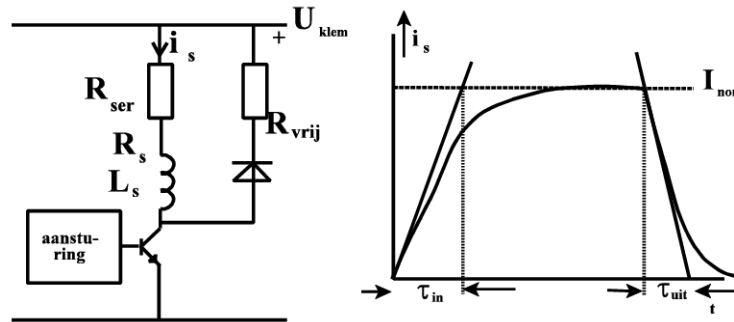
Als de stappenmotor eenmaal op gang is gekomen, dan is het versnellingskoppel niet meer nodig. De frequentiegrens karakteristiek wordt dan van belang. Deze geeft voor een draaiende motor de grens van de stapfrequentie voor een bepaald lastkoppel aan. De frequentiegrens karakteristiek valt dus te vergelijken met de koppel-toerental karakteristiek van een andere motor. In figuur 2.20 zijn de frequentiegrens karakteristiek en de startstop karakteristiek in 1 karakteristiek weergegeven. Het aanlooppkoppel is voor beide karakteristieken gelijk. Het gebied onder de start-stop karakteristiek heet het 'slew rate' gebied. Als men vanuit stilstand zo snel mogelijk naar een opgegeven positie wil, zal men de verschillende frequentiegebieden moeten doorlopen. De motor kan met een frequentie uit het start-stop gebied vanuit stilstand op gang komen. Vervolgens probeert men zo lang mogelijk in een werkpunt op de frequentiegrens karakteristiek te blijven. Een dergelijk werkpunt is echter alleen te bereiken via het 'slew rate' gebied. In dit gebied moet men behoedzaam de stapfrequentie opvoeren, omdat anders de kans bestaat dat de motor stil gaat staan. Er moet namelijk genoeg koppel aanwezig zijn om te kunnen versnellen.

2.7 Het inschakelen van de fasen

Om de stappenmotor zo goed mogelijk te laten werken, moeten we er voor zorgen dat de stap zo snel mogelijk uitgevoerd wordt. Dit houdt o.a. in dat de eindwaarde van de bekrachtigingsstroom I_s zo snel mogelijk bereikt moet worden. Het verloop van de stroom wordt bepaald door de weerstand R_s , de inductiviteit van de wikkeling



Figuur 2.20: Het werkgebied van een stappenmotor



Figuur 2.21: Het schema en verloop van i_s bij in- en uitschakelen van een fase

(L_s) en de voedingsspanning U_{klem} , wat in formule 2.1 wordt weergegeven.

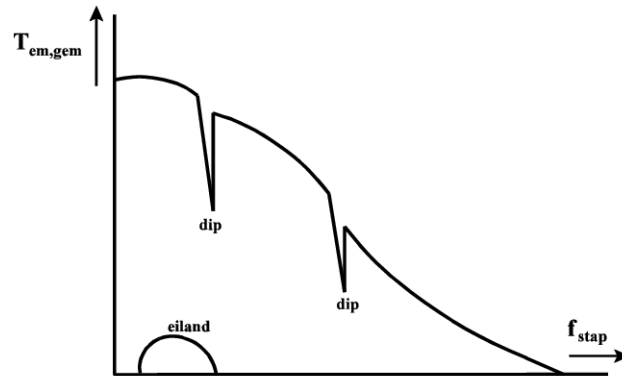
$$\tau_{in} = \frac{L_s}{R_{ser} + R_s} \quad (2.1)$$

Het verloop van i_s is in figuur 2.21 getekend. Is de waarde van R_{ser} groot, dan is τ_{in} klein en neemt de bekrachtigingsstroom i_s snel toe. Wil men bij een hoge R_{ser} toch eenzelfde eindwaarde voor i_s bereiken, dan zal men een hogere klemspanning moeten kiezen.

$$\tau_{uit} = \frac{L_s}{R_{ser} + R_s + R_{vrij}} \quad (2.2)$$

De constante τ_{uit} wordt de uitschakel tijdconstante genoemd. Met een vergroting van R_{vrij} zal de stroom i_s sneller afnemen. (formule 2.2) We moeten er wel rekening mee houden dat een te grote R_{vrij} het plaatsen van de vrijlooptiode teniet doet, wat de transistor kan vernielen.

2.8 Resonanties en instabiliteiten



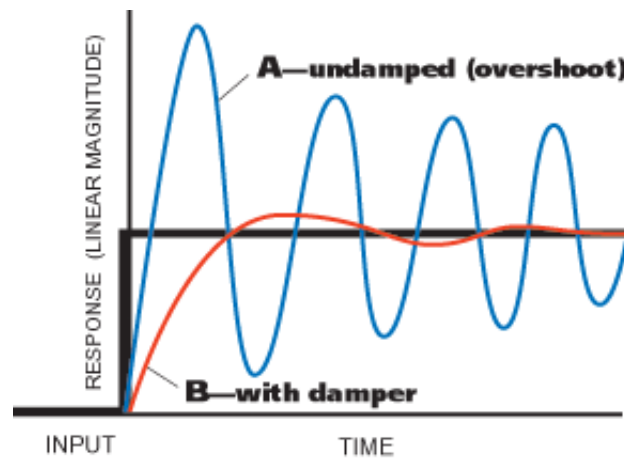
Figuur 2.22: Resonantie effecten op de frequentiegrens karakteristiek

Vanwege allerlei niet lineaire effecten (verzadiging, demping, in- en uitschakelen) vertonen de koppelfrequentie krommen dips en 'eilanden'. In deze gebieden is het gedrag van de stappenmotor niet voorspelbaar. In figuur 2.22 zijn deze gebieden geschetst voor een bepaalde stappenmotor. Binnen het gebied van een eiland kan de motor niet uit zichzelf starten. Door een hogere of een lagere startfrequentie te nemen kunnen deze gebieden vermeden worden. De resonanties verdeelt men in drie categorieën:

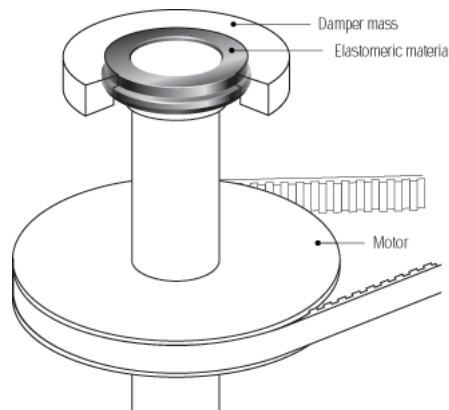
1. *Low frequency resonanties*. Deze zijn van mechanische aard en liggen rond de 100Hz. De resonanties worden veroorzaakt door eigenfrequenties van het systeem.
2. *Mid-range instabiliteiten* (500-1500Hz). De oorzaak van deze resonanties ligt bij de aanstuurcircuits (choppers).
3. *Higher-range oscillaties* (2500-4000Hz). De oorzaak van deze oscillaties is nog niet goed bekend. Ze worden verholpen door de stappenmotor te dempen.

2.9 Demping

Als de natuurlijke wrijving van ons systeem geen voldoende grote dempingsfactor oplevert, kunnen we de demping nog vergroten door het bijplaatsen van een mechanische en/of elektrische demper. (fig 2.23)



Figuur 2.23: Overshoot zonder en met demping



Figuur 2.24: droge demping

2.9.1 Mechanische demping

met droge wrijving

De motor drijft via een koppeling een traagheidsmassa aan (vliegwiel). Bij iedere snelheidsverandering zal de demping actief worden. Bij constant toerental gaat het vliegwiel geen energie meer opnemen, maar gewoon meedraaien. Als men mechanische demping toepast met droge wrijving, moet men wel rekening houden met het bijkomende traagheidsmoment bij het kiezen van de motor.

met visceuse wrijving

De traagheidsmassa draait via een aan de as gekoppelde schijf in een oliebad mee. De viscositeit van de olie, het oppervlak en de oppervlaktegesteldheid zal de demping bepalen.

2.9.2 Elektrische demping

Er zijn elektrische dempingsmodules die worden aangesloten op de elektronische vermogenschakelaars. De module zal bij iedere puls van de sturing, drie uitgangspulsen genereren. De eerste komt bij het begin van de oscillatorpuls, en stuurt de motor in voorwaartse richting, de tweede komt iets later, en is een rempuls die de overshoot gaat beperken. Tenslotte komt ook de derde puls, terug in voorwaartse richting, die belet dat de rotor vertraagt na de rempuls.

2.10 Stapmodi

Het bekrachtigen van de wikkelingen kan met verschillende patronen gebeuren. De gewone methodes, *Volstapbedrijf*, *Halfstapbedrijf* en ook *Mini-stap bedrijf* ga ik hier bespreken.

2.10.1 Volstapbedrijf

Volgorde	Fase A	Fase B	Fase \bar{A}	Fase \bar{B}
1	■			
2		■		
3			■	
4				■

Figuur 2.25: Volstapbedrijf (1 wikkeling per stap)

Bij deze eerste manier van volstapbedrijf wordt altijd maar 1 fase tegelijk bekrachtigd. De wikkelingen worden bekrachtigd zoals te zien is in tabel 2.25. Het nadeel van deze methode is dat in unipolaire motoren slechts 25% van de windingen wordt gebruikt, en in bipolaire motoren 50%. Hierdoor wordt niet het maximale koppel verkregen.

In het tweede geval worden twee fasen tegelijk bekrachtigd (tabel 2.26). De staphoek is hetzelfde als wanneer er slechts 1 wikkeling wordt bekrachtigd. Voor unipolaire motoren betekent dit dat 50% van de wikkelingen wordt benut en bij bipolaire

Volgorde	Fase A	Fase B	Fase \bar{A}	Fase \bar{B}
1				
2				
3				
4				

Figuur 2.26: Volstapbedrijf (2 wikkelingen per stap)

motoren 100% van de wikkelingen. Het koppel ligt dus hoger dan bij volstapbedrijf met 1 wikkeling per stap.

2.10.2 Halfstapbedrijf

Volgorde	Fase A	Fase B	Fase \bar{A}	Fase \bar{B}
1				
2				
3				
4				
5				
6				
7				
8				

Figuur 2.27: Half-step-drive

Als men de twee tabellen van volstapbedrijf samenbrengt, kan men de volgorde zodanig kiezen dat de rotor steeds een halve stap maakt. Op deze manier kunnen we dus een kleinere staphoek bekomen, waardoor de motor nauwkeuriger wordt.

2.10.3 Mini-stap bedrijf

In mini-stap bedrijf worden de stromen die door de wikkelingen lopen continu veranderd (meestal door een digitaal-analoog converter), op zodanige wijze dat een hele stap wordt onderverdeeld in meerdere kleine stapjes. Met ministappen kan men dus een grotere nauwkeurigheid in positie bereiken, omdat de fout in positie te compenseren is door de juiste fasestromen aan te bieden. Door het maken van kleine stapjes blijkt ook dat de motor veel rustiger beweegt. Men kan dus beter mini-stappen toepassen i.p.v. een volledige stap. Voor eenzelfde hoeksnelheid moeten de mini-stappen dan wel met een hoge frequentie aangeboden worden. Omdat de aansturing heel ingewikkeld is en er verschillende problemen optreden, ga ik niet dieper op ministapbedrijf in.

2.11 Mijn motor

De motor die ik dit schooljaar heb gebruikt om de drivertrap en software te testen, en die ik voor mijn positiesturing ga gebruiken, is een enkelpolige unipolaire stappenmotor van de fabrikant RS Electronics. Hij draagt het typenummer '440-284'. De motor is te zien in figuur 2.28. Hij is gemaakt voor een werkspanning van rond de 12 volt, en heeft een spoelstroom van enkele honderden milliampères als er 12 volt op wordt aangebracht. Zijn staphoek is 7.5 deg, wat vrij groot is. In figuur 2.29 is



Figuur 2.28: Mijn motor

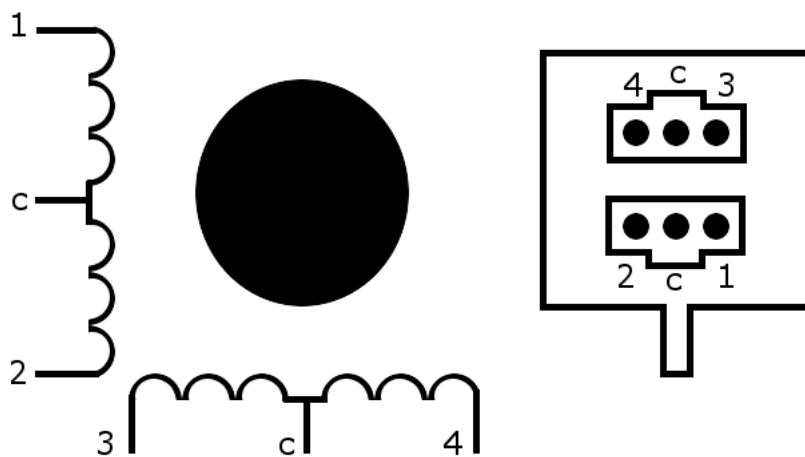
de tabel te zien met de stapsequentie waarmee de motor moet worden aangestuurd om volle stappen te maken. Ook de draaizin is aangeduid. In figuur 2.30 is de tabel te zien met de stapsequentie voor halve stappen. In de datasheet die RS Electronics bij de motor leverde stonden fouten in de Nederlandse, Spaanse en Engelse vertalingen. Ik heb alle tabellen vergeleken, en met de Zweedse tabel draaide de motor uitstekend. In de andere tabellen staan twee bits van stap 7 verwisseld, waardoor de motor op die stap raar doet. In figuur 2.31 zijn de aansluitingen van mijn motor getekend. De aansluitdraden hebben allemaal een kleur gekregen: fase 1 zwart, fase 2 groen, fase 3 rood, fase 4 blauw, en de twee middenaftakkingen (common) wit.

Stapnummer	Windingen				Wijzerszin ↓
	1	2	3	4	
1	1	0	1	0	
2	0	1	1	0	
3	0	1	0	1	
4	1	0	0	1	

Figuur 2.29: Tabel voor volstappen

Stapnummer	Windingen				Wijzerszin ↓
	1	2	3	4	
1	1	0	0	0	
2	1	0	0	1	
3	0	0	0	1	
4	0	1	0	1	
5	0	1	0	0	
6	0	1	1	0	
7	0	0	1	0	
8	1	0	1	0	

Figuur 2.30: Tabel voor halve stappen



Figuur 2.31: De aansluitingen van mijn motor.

Hoofdstuk 3

De Microcontroller

3.1 Inleiding

Een microcontroller (ook afgekort als μC) is een digitaal IC dat programmeerbaar is en taken kan verrichten.



Figuur 3.1: Assortiment microcontrollers van verschillende groottes

In tegenstelling tot een microprocessor (μP) (dat alleen een centrale verwerkings-eenheid bevat), is een de microcontroller ook uitgerust met enkele noodzakelijke onderdelen zoals programma- en RAM geheugen en I/O pinnen. Hierdoor heeft hij bijzonder weinig externe componenten nodig om een volledig microcomputer systeem te vormen. Om de schakeling rond de μC zo klein mogelijk te houden, kan men de μC ook nog uitrusten met andere zaken, die anders naast de μC op de print geplaatst moeten worden. Hierdoor kan men de schakeling simpeler houden en de productiekosten verlagen. Enkele voorbeelden van ingebouwde periferie:

Klokoscillatoren (waardoor er geen extern kristal meer nodig is!), *EEPROM geheugen* (niet vluchtig geheugen) dat kan dienen om meetwaarden en instellingen op te slaan, *Timers*, *PWM* (pulsbreedtemodulatie) *modules* en *A/D converters* die het werken met analoge signalen eenvoudiger maken, *synchrone seriële poorten* zoals de

SPI en I²C bus (die de bedrading vereenvoudigen) en *asynchrone seriële poorten* (om bvb via RS-232 de pc te communiceren). Ook zijn er μC 's die voor speciale toepassingen zijn ontwikkeld, die bevatten dan bvb een USB of CAN controller. Een concreet voorbeeld hiervan is de rfPIC serie van microchip, waarin een UHF modulator zit verwerkt.

3.2 Soorten microcontrollers

Er zijn veel soorten microcontrollers op de markt. Ze verschillen onderling (o.a.) in fysieke grootte, architectuur, geheugen, ingebouwde periferie, busgrootte en prijs. Door deze waaier in aanbod is het mogelijk een microcontroller te kiezen die zeer goed geschikt is voor een bepaalde toepassing. Meestal behoren μC 's tot een bepaalde processorfamilie. De leden van zo'n familie hebben gelijkaardige eigenschappen, maar verschillen in de soort van programmeergeheugen, de ingebouwde periferie, aantal I/O pinnen, ...

Microcontrollers zijn verkrijgbaar met verschillende soorten programmeergeheugen. Voor hobby projecten en (heel) kleinschalige productie worden meestal μC 's met flash-programmeergeheugen gebruikt. Flashgeheugen is overschrijfbaar en vele malen te programmeren. Indien de microcontroller in een product wordt ingebouwd dat een hoge oplage heeft, wordt het programmeergeheugen uitgevoerd in PROM of EPROM (zonder venster) waardoor hij maar 1 keer kan worden geprogrammeerd, maar de productiekosten laag zijn. Tijdens de ontwikkelingsfase wordt dan gebruik gemaakt van een ander lid van die processorfamilie, die een EPROM-wisvenstertje heeft of flash programmeergeheugen, waardoor hij gemakkelijk(er) gewist kan worden.

Microcontrollers verschillen onderling ook van architectuur. Zo zijn er microcontrollers gebaseerd op oudere, bewezen architecturen zoals de 6800 (vb: de 68HC05 van Motorola) of de 8051 (vb: de AT90S2051 van atmel). Dit zijn *Complex Instruction Set Core* μC 's. Ze hebben een heel uitgebreide instructieset. Toen aangetoond werd dat doorgaans slechts 20% van de instructies worden gebruikt, heeft men de *Reduced Instruction Set Core* μC 's bedacht. Deze μC 's hebben een kleinere, maar krachtigere instructieset (+- 118 instructies). Daardoor zijn ze goedkoper in productie (er moeten minder transistoren worden gebruikt), sneller en moet de assemblerprogrammeur minder instructies kennen.

3.3 Keuze van de μC

Voor mijn GIP zoek ik een μC die aan volgende voorwaarden voldoet:

- flashgeheugen
- programmeerbaar in de schakeling zelf (ISP)

- goede documentatie en voorbeeldtoepassingen
- beschikbare HLL compilers

Als we even op zoek gaan naar populaire μC 's die aan deze aan deze eisen voldoen, vinden we de volgende controllers:

- de PIC16f628 van Microchip (PIC RISC cpu)
- de 90S2313 van Atmel (AVR RISC cpu)
- de SX28AC van Uvicom (SX RISC cpu)
- de 89S2051 van Atmel (8051 cpu)
- de 68HC05 van Motorola (6800 cpu)

Aangezien ik zelf reeds ervaring heb met de PIC microcontroller (het dus geen uitdaging zou zijn om de PIC te gebruiken), en we in de school reeds ontwikkelpanelen hebben voor de ATMEL AVR μC 's, kiezen we voor de AT90S2313 .

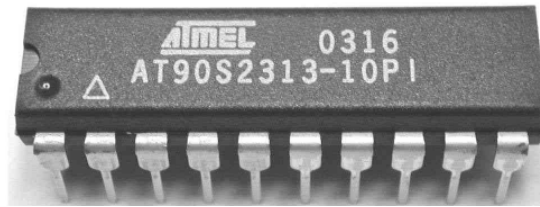
3.4 Talen

We kunnen de μC programmeren in verschillende talen. Iedere taal heeft zijn voor- en nadelen. We moeten dus afhankelijk van de toepassing een geschikte taal kiezen. We zouden een van de volgende *High Level Languages* kunnen kiezen:

- BASIC (bvb BASCOM AVR)
- C (bvb de GNU C Compiler)
- Pascal
- Java

Natuurlijk is er ook nog assembleertaal, waarvoor Atmel een gratis assembler aanbiedt, maar AVR-assembler leren neemt te veel tijd in beslag. De talen C en Java zijn dan weer qua syntax redelijk complex, en vereisen een grondige kennis van de programmeertaal zelf. Persoonlijk zou ik voor de taal C of assembleertaal kiezen, omdat ik reeds veel ervaring heb met BASIC, en ik graag eens zou experimenteren met C en assembler. Maar omdat deze talen te complex zijn, en mijn medeleerlingen BASIC gebruiken, ben ik genoodzaakt BASIC (wat staat voor *Beginner's All Purpose Symbolic Instruction Code*) te kiezen. De firma MCS Electronics verkoopt de BASIC ontwikkelomgeving BASCOM-AVR die, naast een basic compiler voor de Atmel AVR familie, ook een programmer en simulator bevat. Voor BASCOM-AVR is geen licentie vereist zolang het programma kleiner blijft dan 2Kb en niet voor commerciële doeleinden wordt gebruikt. Een uitstekende keuze dus, omdat deze beperkingen niet voor mij opgaan, de AT90S2313 bevat namelijk slechts 2Kb flashgeheugen.

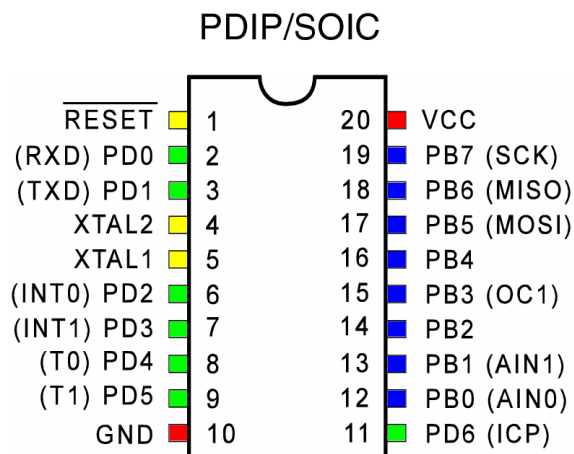
3.5 Features van de AT90S2313



Figuur 3.2: De AT90S2313 microcontroller

Hieronder is een lijst van de features van de AT90S2313 (figuur 3.2) te vinden:

- RISC architectuur, 118 instructies, de meeste duren 1 klokcyclus
- een maximale klokfrequentie van 10Mhz, waarbij hij 10MIPS verwerkingscapaciteit heeft
- 2Kb Flash-programmageheugen
- 128 bytes SRAMgeheugen
- 128 bytes EEPROMgeheugen
- 2 Timers (een 8-bits en 16-bits timer)
- CCP en PWM module
- analoog-digitaal converter
- full duplex UART
- in system programming



Figuur 3.3: De pinout van de AT90S2313

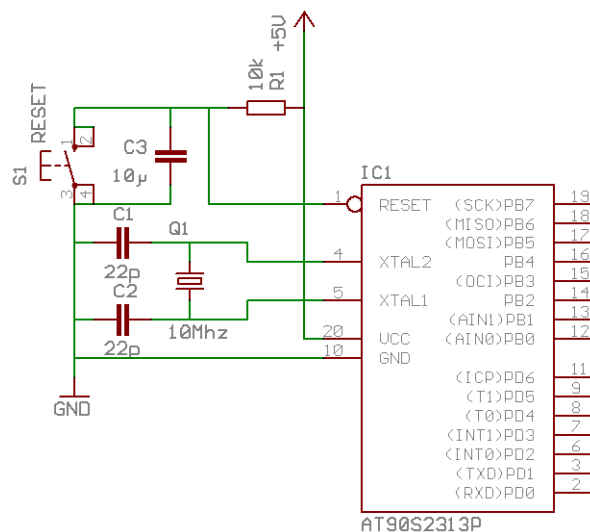
3.6 Schakelingen met de μC

3.6.1 Basisschakeling

Door de complexiteit van een microcontroller zijn er zeer weinig tot geen externe componenten nodig. De AT90S2313 heeft echter wel geen resonantiekring aan boord die de controller van een hartslag voorziet. Daarom dient er een kristal (met condensatoren) naast de μC op de print te worden geplaatst. De frequentie van het kristal bepaalt hoe snel de μC werkt. De AT90S2313 kan maximaal op een klokfrequentie van 10Mhz werken, maar ik gebruik een frequentie van 4Mhz, omdat mijn programma niet veel inspanning van de processor vraagt.

Om de AT90S2313 te laten werken, is naast een voedingsspanning, een programma en een kristal, ook een pullupweerstand nodig aan de actief lage resetingang, om de μC uit zijn resetstand te houden. In ontwikkelopstellingen wordt er ook een druktoets voorzien tussen de resetpin en de massa, om de μC gemakkelijk te kunnen resetten. Om de druktoets te ontzenderen is ook nog een condensator C3 van 10F voorzien.

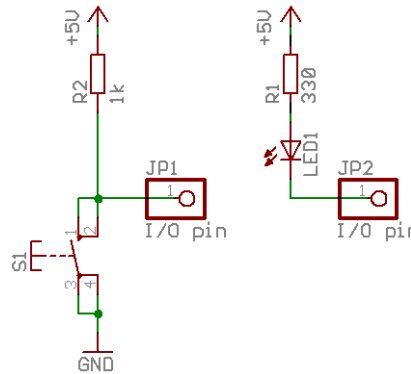
Figuur 3.4 toont de basisopstelling van de AT90S2313 .



Figuur 3.4: De basisschakeling van een AT90S2313

3.6.2 Aansluiten van schakelaars en led's

Om een visuele weergave te krijgen van het logisch niveau op een uitgangspin, gebruik ik een LED aangesloten zoals figuur 3.5. Indien er een logische 0 op de pin



Figuur 3.5: Led en schakelaar

staat, valt er 5V over de LED en licht deze op. De LED is gedoofd indien er op de pin 5V staat (logische 1). De LED is aangesloten tussen +5V en de I/O pin, omdat de stroom door de LED groter is dan de maximale sourcecurrent (stroom bij logische 1) van de μC . De sinkstroom (stroom bij logische 0) is wel groot genoeg om de LED te laten oplichten.

Figuur 3.5 toont ook een schakelaar met pullupweerstand aangesloten op een I/O pin. De reden waarom ik een pullup weerstand gebruik is dat de I/O pin intern ook al een (hoogohmige) pull-up weerstand bevat. Als er een draadbreek zou zijn, zal het logisch niveau niet veranderen, wat bij een pulldownweerstand naar de massa wel het geval zou zijn.

3.6.3 In System Programmer

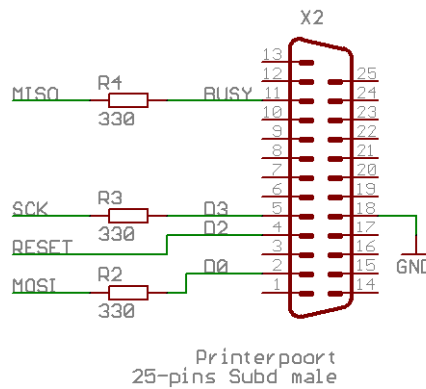
De AT90S2313 hoeft dankzij de ingebouwde ondersteuning voor "In System Programming" niet uit de schakeling worden gehaald om opnieuw geprogrammeerd te worden.

Zodra de \overline{RESET} pin laag wordt gemaakt, komt de AT90S2313 in programmeermodus terecht. Pinnen B5, B6 en B7 krijgen dan een nieuwe functie: ze dienen als de seriële interface die de μC met het programmeerapparaat verbindt (Pin B7 wordt dan de Serial Clock, pin B6 MISO (seriële uitgang) en B5 MOSI (seriële ingang)). Nadat het programmeerapparaat de Programming Enable instructie heeft uitgevoerd, kunnen er lees- en schrijfcommando's naar de μC worden gestuurd.

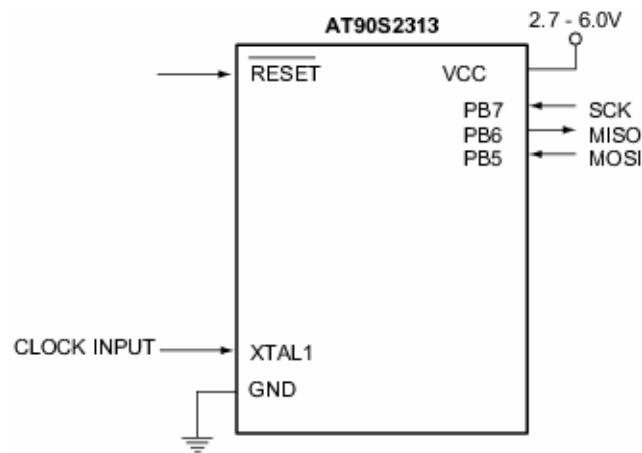
Op het gebruikte protocol moet ik niet dieper ingaan, omdat BASCOM dit voor ons afhandelt. Het is mogelijk om het programma- en datageheugen van de μC uit te lezen en te beschrijven via ISP. Ook kan men fuse en lockbits programmeren in de μC die het beschrijven en uitlezen van de μC beletten wanneer de μC in het eindproduct is geplaatst.

De interface tussen de programmer en de μC bestaat dus uit slechts vijf signalen: RESET, SCK, MISO, MOSI en GND. De soberste ISP programmer is getoond in figuur 3.6. Deze schakeling wordt via de parallele printerpoort met de pc verbonden. De drie weerstanden zijn toegevoegd om de stromen te beperken.

Naast serieel kan de μC ook parallel worden geprogrammeerd. Bij parallel programmeren wordt er gebruikgemaakt van een acht bits brede databus, en 7 stuursignalen. Dit vereist alle pinnen van de μC . Dit maakt parallel programmeren ongeschikt voor ISP.



Figuur 3.6: De hardware van het "programmeerapparaat"



Figuur 3.7: Voorstelling van seriële ISP uit de datasheet

3.6.4 Communicatie met de pc via RS-232

Naast de programmeerkabel zou het ook nog handig zijn een tweede verbinding met de PC te hebben, waarmee ik de μC tijdens het lopen van het programma bevelen zou kunnen geven met de PC. Zo'n verbinding is het eenvoudigst te realiseren door gebruik te maken van de interne *UART* van de μC . Met behulp van de *UART* is het mogelijk om over een seriële lijn asynchroon gegevens te versturen en te ontvangen. In de PC is ook een *UART* aanwezig, we hoeven dus enkel de seriële poort van microcontroller en PC op een correcte manier te verbinden. De *UART* van de μC wordt geconfigureerd en aangesproken door *BASCOM BASIC*, zodat ik in mijn programma de simpele *BASIC* instructies *Input* en *Print* kan gebruiken, die van de *UART* lezen en er naar schrijven. Voorbeelden van deze commando's zijn te zien in listing 3.1.

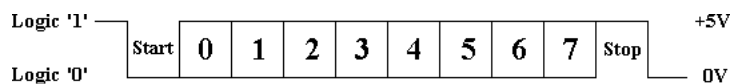
Listing 3.1: de *UART* wordt met 'print' en 'input' aangesproken

```
Input "Geef een getal in >" , Inp
Print "Het getal is " , Inp
```

Aan de PC-zijde maken HLL's het ook zeer eenvoudig om de *UART* te benaderen. Het is mogelijk een relatief eenvoudig PC programma te schrijven dat de invoer van het toetsenbord rechtstreeks naar de μC stuurt, en de karakters die de μC naar de pc stuurt, in een venster te tonen.

3.6.5 De RS-232 standaard

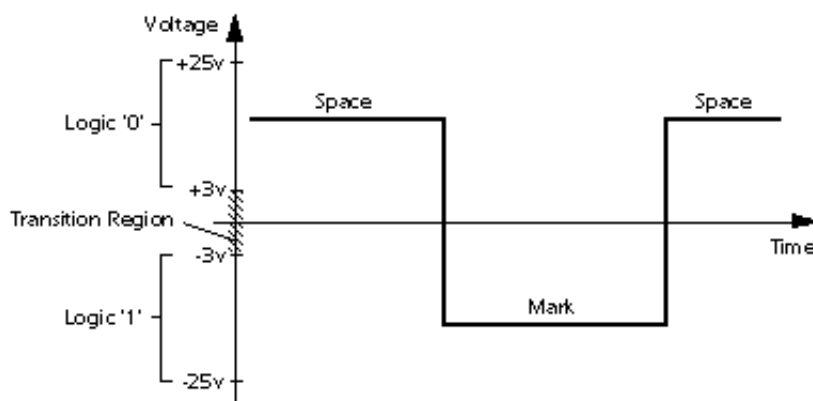
Zoals ik al heb vermeld is een RS-232 verbinding asynchroon. Dit betekent dat er geen kloksignaal wordt meegezonden met de gegevens. Elk woord (of byte) wordt intern gesynchroniseerd met behulp van een meegezonden startbit, en een intern kloksignaal dat wordt ingesteld afhankelijk van de baudrate.



Figuur 3.8: De opbouw van een frame.

Het dataformaat dat ik ga gebruiken is 8N1 bij 9600 baud. '8N1' is de afkorting voor 8 data bits, geen pariteitsbit, en 1 stop bit. Zoals getoond in figuur 3.8, wordt de communicatie gestart met een start bit (logisch 0). Vervolgens wordt elke bit achtereenvolgens over de lijn verstuurd, met de LSB (least significant bit) eerst. Nadat de MSB (most significant bit) is verstuurd, wordt er nog een stopbit (logische 1) verstuurd.

In het diagram is na de stopbit een logische 0 getoond. Dit is de startbit die een volgende byte voorafgaat. Indien er geen data meer moet verstuurd worden, wordt de lijn hoog gehouden, dit noemt men de idle state. Het is belangrijk dat de verzender en ontvanger met dezelfde snelheid werken, anders zal de communicatie in het honderd lopen.



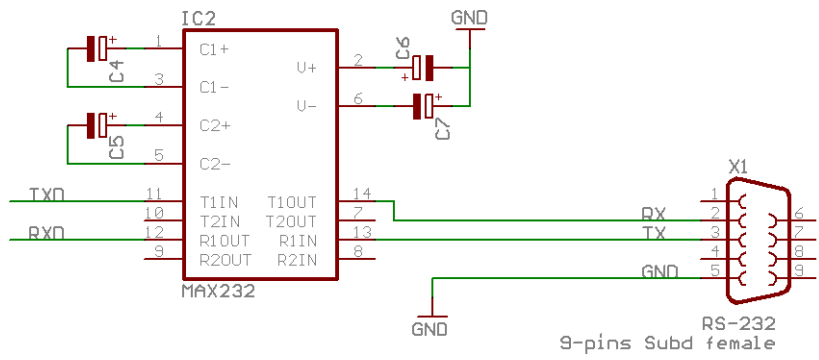
Figuur 3.9: De logische niveaus van RS-232

Het signaal komt zoals het getoond wordt in figuur 3.8 uit de zender. De RS-232 standaard legt echter op dat de gegevens met logische niveaus van +3 tot +25 volt als logische 0 (space) en -3 tot -25 volt als logische 1 (Mark) over de lijn worden verstuurd. Dit is getoond in figuur 3.9. Spanningen tussen +3V en -3V worden genegeerd.

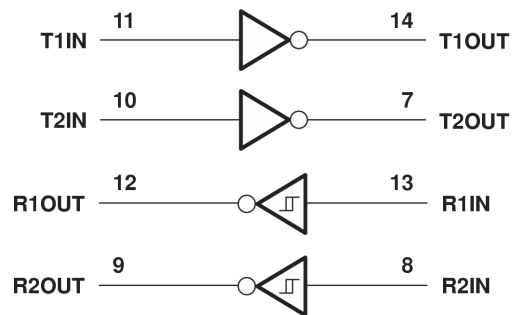
Om de signalen van TTL niveaus om te zetten naar RS-232 niveaus maakt men gebruik van een level-converter.

3.6.6 RS-232 – TTL Interface

Omdat de seriële poort van de pc volgens de RS-232 standaard werkt (of de EIA .V24 standaard die praktisch equivalent is), kunnen we deze niet direct met de μC verbinden. RS-232 werkt namelijk met logische niveaus van +15V en -15V, en de μC met TTL niveaus. We moeten dus een omzetter gebruiken als we de RS-232 signalen met de μC willen verbinden. De MAX232 is een heel bekend IC dat vaak voor dit doel wordt gebruikt. De toepassing van het IC is getoond in figuur 3.10. Het IC is intern opgebouwd uit twee receivers (die de signalen van de PC omzetten naar TTL niveau), en twee drivers (die de TTL niveaus omzet naar RS-232 niveaus) (figuur 3.11. Voor dit laatste is een symmetrische voedingsspanning nodig van 15V, wat een serieuze vereiste is, en waaraan een prijskaartje hangt dat het IC onaantrekkelijk zou maken. Hiervoor is een oplossing bedacht: men heeft namelijk in het IC een spanningspomp ingebouwd die zelf deze spanningen opwekt.



Figuur 3.10: Max232 Level-Converter



Figuur 3.11: Het logisch diagram van de MAX232

Hoofdstuk 4

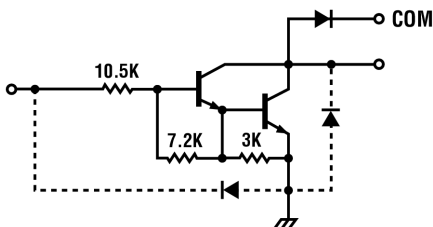
De aansturing van de motor

4.1 De drivertrap

De stappenmotor wordt via een drivertrap verbonden met de microcontroller. Als drivertrap volstaat eender welke elektronische schakelaar, zolang hij snel genoeg is en de stromen aan kan. Relais, vermogenmosfets, transistoren, kunnen allemaal worden gebruikt. De motor aansturen met transistoren lijkt de meest logische oplossing. Hoewel dit perfect mogelijk is en niet ingewikkeld, heb ik tijdens mijn zoektocht achter informatie een eenvoudigere oplossing gevonden. een geïntegreerd circuit.

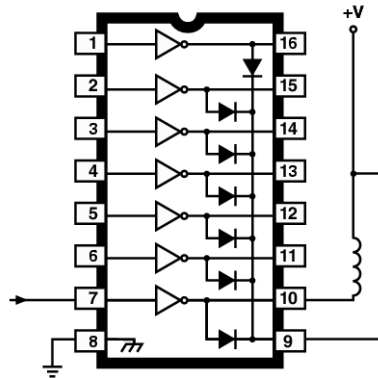
4.2 Drivertrap met een ULN2004

Als drivertrap neem ik een IC uit de ULN20xx reeks, deze zijn *High-Voltage, High-Current Darlington Arrays*. De IC bevat dus een array met drivertrapjes (zie figuur 4.1). Zoals de naam al vermeldt, kan de IC met hoge spanningen en hoge stromen werken (de IC kan vermogens tot 230W sturen). Met een maximale stroom van 500mA en maximale spanning van 95V is hij dan ook heel geschikt voor mijn toepassing.



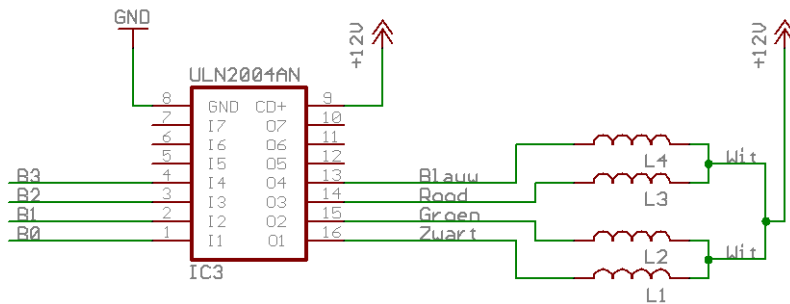
Figuur 4.1: 1 Drivertrap uit een ULN2004

TYPICAL APPLICATIONS



Figuur 4.2: Een voorbeeldschema uit de datasheet van de ULN2004

Als we de stappenmotor op de driver aansluiten zoals in het voorbeeld (figuur 4.2), krijgen we volgend schema: (4.3)



Figuur 4.3: De verbindingen tussen de driver en de motor

Als we nu pin 1-4 van het IC hoog maken, wordt spoel L_{1-4} bekrachtigd. Het IC heeft TTL compatible ingangen, die we rechtstreeks aan de μC kunnen hangen.

4.3 De sub stap

Om het hoofdprogramma te vereenvoudigen heb ik een subroutine geschreven die de 4 bits gaat bepalen waarmee de fasen van de motor gaan worden aangestuurd. Deze sub werkt met een tabel (4.4) met de halve stappen. De sub gaat telkens hij wordt opgeroepen de stapteller verhogen. Afhankelijk of er volstappen of halve stappen moeten worden gemaakt, gaat er al of niet een halve stap worden overgeslagen. Hierdoor is het mogelijk halve stappen en volledige stappen op elkaar te laten volgen.

De sub stap is samen met een voorbeeldlus die het gebruik van de sub duidelijk maakt, weergegeven in listing 4.1.

Stapnummer	Windingen				Tegen- Wijzerszin
	1	2	3	4	
1	1	0	0	0	↑ ↓
2	1	0	0	1	
3	0	0	0	1	
4	0	1	0	1	
5	0	1	0	0	
6	0	1	1	0	
7	0	0	1	0	
8	1	0	1	0	

Figuur 4.4: Tabel met stappen

Listing 4.1: Sub stap met programmavoorbeeld

```

'Subs declareren
Declare Sub Stap(byval Grootte As Byte ,
                  Byval Richting As Byte)

'pinrichting declareren
Config Portb = Output

'Synoniemen declareren
Const Wijzerszin = 1                'Richtingen
Const Tegenwijzerszin = 0
Const Half = 0
Const Volledig = 1

'voor globaal gebruik
Dim Stap As Byte                    'Huidige stap
Dim Tijd As Byte                    'delay tussen stappen

Tijd = 20                             'Variablen initialiseren
Stap = 1

' voorbeeldlus om het gebruik van de sub stap
' te demonstreren.

```

```

Do
  For N = 0 To 96                                '360 in halfstap
    Call Stap(half , Wijzerszin)
  Next N
  For N = 0 To 48                                '360 in volstap
    Call Stap(volledig , Tegenwijzerszin)
  Next N
Loop

End
'-----

Sub Stap(grootte As Byte , Richting As Byte)
'byte grootte bepaalt of er halve of er
'een volledige of halve stap wordt gemaakt.
'byte Richting bepaalt de draaizin.

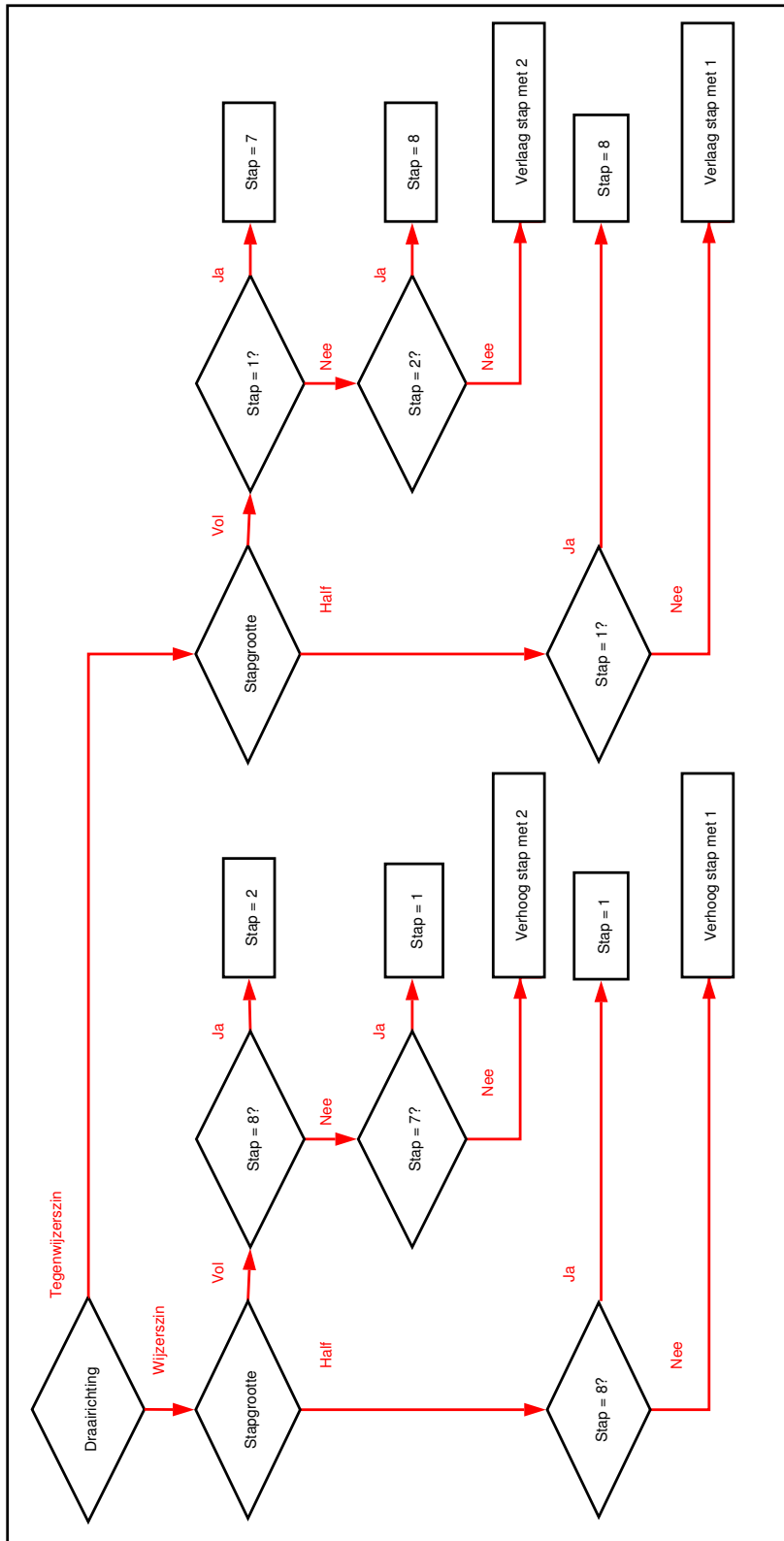
If Richting = Wijzerszin Then 'wijzerszin
  If Grootte = Volledig Then 'Volledige stap
  Select Case Stap
    Case 8 : Stap = 2
    Case 7 : Stap = 1
    Case Else : Stap = Stap + 2
  End Select
Else 'Halve stap
  If Stap = 8 Then
    Stap = 1
  Else
    Stap = Stap + 1
  End If
End If
Else 'Tegen wijzers
  If Grootte = Volledig Then 'Volledige stap
  Select Case Stap
    Case 1 : Stap = 7
    Case 2 : Stap = 8
    Case Else : Stap = Stap - 2
  End Select
Else 'Halve stap
  If Stap = 1 Then
    Stap = 8
  Else
    Stap = Stap - 1

```

```
    End If
  End If
End If

Select Case Stap                                'Fasen overeenkomstig'
  Case 1 : Portb = &B00001000                 'met stapnummer sturen'
  Case 2 : Portb = &B00001001
  Case 3 : Portb = &B00000001
  Case 4 : Portb = &B00000101
  Case 5 : Portb = &B00000100
  Case 6 : Portb = &B00000110
  Case 7 : Portb = &B00000010
  Case 8 : Portb = &B00001010
End Select

  Waitms Tijd                                  'Delay'
End Sub
```



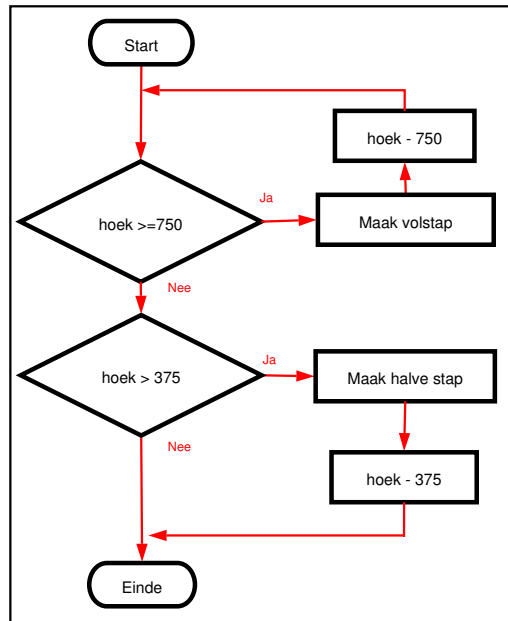
Figuur 4.5: Flowchart van de sub stap

Hoofdstuk 5

Positiesturing

5.1 Principe

Het subprogramma dat uit een opgegeven hoek het aantal stappen gaat berekenen, is weergegeven in listing 5.1. Het principe is eenvoudig. Omdat werken met komma-getallen op 8-bits microcontrollers niet mogelijk is zonder speciale programmatuur, maak ik geen gebruik van een deling, maar een lus. Deze lus zal worden doorlopen zolang de hoek die nog moet worden gedraaid groter is dan een volstap. In de lus zal telkens een volstap worden gemaakt, en vervolgens de grootte van een volstap van de hoek worden afgetrokken. Na de lus zal nog worden gecontroleerd of de resterende hoek groter is dan een halve stap, indien dit zo is wordt er nog een halve stap gemaakt, en een halve stap hoek afgetrokken van de resthoek. De resthoek is vrij groot bij motoren met een grote staphoek, zoals die van mij! Ik heb een flowchart getekend van dit programma, die getoond is in figuur 5.1. Het programma stuurt het aantal volstappen en halve stappen dat het maakt, en de resterende hoek naar de seriële poort voor debugging doeleinden.



Figuur 5.1: De flowchart voor sub draaihoek

Listing 5.1: Een hoek draaien

```

Sub Draaihoek(byval Grootte As Integer , Richting As Byte)
  ' hoekgrootte wordt 100x groter in parameters opgegeven.

While Grootte > 750
  Grootte = Grootte - 750
  N = N + 1
  Call Stap(volledig , Richting)
Wend

Print N ; "_Volstappen"

If Grootte >= 375 Then
  Call Stap(half , Richting)
  Grootte = Grootte - 375
  Print "1_Halve_stap"
End If

Print "De_resthoek_is_nu_0," ; Grootte

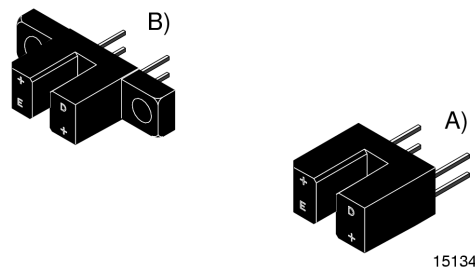
End Sub
  
```


5.2 Positieterugkoppeling

Om de stand van de as van de motor te kalibreren met een nulstand, maak ik gebruik van een positiedetectie met een CNY-37 lichtsluis. Zo'n lichtsluis bestaat uit een emittor en een detector, die samen in een zwart plastic behuizing zijn ondergebracht (zie fig. 5.2 en 5.3). De emittor bevat een LED die infrarood licht uitstraalt, en de detector bevat een fototransistor.

Het lichtsluis wordt volgens figuur 5.4 aangesloten. De emittor wordt via een serieweerstand van $\pm 100\Omega$ aangesloten op 5V, waardoor hij IR-licht gaat uitstralen. De detector wordt in serie met een weerstand van $1k\Omega$ aangesloten op 5V. Als het licht van de emittor op de detector valt, kan er via de fototransistor een stroom lopen, en daalt de spanning op het punt tussen de serieweerstand en de detector. Deze spanning wordt rechtstreeks op de microcontroller aangesloten, en is hoog resp. laag genoeg om als logisch niveau van een 1 en 0 gedetecteerd te worden.

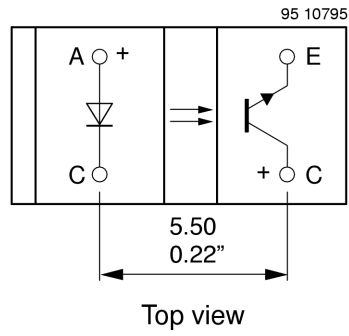
Op de as van de motor wordt een aluminium schijfje aangebracht waarvan de afmetingen te zien zijn in figuur 5.5. De lichtsluis wordt gemonteerd op de rand van de schijf zodat het gaatje in de schijf precies tussen de emittor en detector valt. Enkel als het gaatje zich tussen de emittor en detector bevindt, zal het IR-licht de detector bereiken, en een logische 0 aan de microcontroller geven. In listing 5.2 is een programma weergegeven om de rotor te verdraaien tot het gaatje zich in de lichtsluis bevindt.



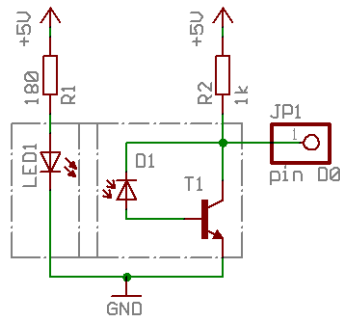
Figuur 5.2: De uitvoeringen van een CNY-37 lichtsluis

Listing 5.2: Draaien naar nulstand

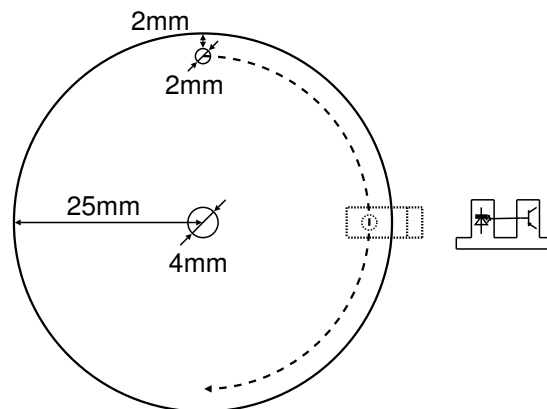
```
' dit programma draait de motor tot het gaatje
' in de lichtsluis is
While Pind.2 = 1
    Call Stap(volledig , Wijzerszin)
Wend
' fasen uitzetten na het draaien
Portb = &B00000000
```



Figuur 5.3: De inhoud van de lichtsluis



Figuur 5.4: Het aansluitschema van de lichtsluis



Figuur 5.5: De mechanische uitvoering van de positiedetectie

Hoofdstuk 6

De praktische uitvoering

Als praktische uitvoering van de positiesturing, heb ik met het tekenprogramma Eagle een print ontworpen die de noodzakelijke elektronica bevat om de microcontroller te laten werken en de stappenmotor aan te sturen. De print heb ik samen met mijn klasgenoten geëetst, en vervolgens bestukt en getest. Het schema (fig. 6.2) en de printlayout (fig 6.3, 6.4, 6.5) zijn te zien op de volgende pagina's.

6.1 De opbouw van de print

In figuur 6.3 heb ik de verschillende delen van de print genummerd. Deze nummering komt overeen met de nummering hieronder.

6.1.1 AT90S2313 μ C

Het hart van de print, de AT90S2313 μ C is samen met zijn reset- en oscillatieschakeling voorzien in het midden van de print. Ik heb gekozen voor de gewone versie van de μ C in DIL behuizing, omdat deze het gemakkelijkste is qua hanteerbaarheid en soldeerbaarheid.

6.1.2 RS-232 interface

De seriële poort van de microcontroller is verbonden met het MAX232 IC, dat voorzien is van de 4 $1\mu F$ condensatoren die het IC nodig heeft. Op de print zijn TX en RX verwisseld zodat een "straight serial extension cable" kan worden gebruikt om de print met de PC te verbinden.

6.1.3 ISP connector

Het laden van het programma in de microcontroller gebeurt via de ISP connector X1. Dit is een RJ-14 connector. Dit is de zespins versie van de veelgebruikte RJ (wat staat voor *Registered Jack*) connector (gebruikt in telefonie- en computer netwerken). Samen met de drie 330ω weerstanden vormt deze identiek dezelfde ISP voorziening als het BASCOM board.

6.1.4 Voeding

Op de print wordt op connector JP4 een gelijkspanning (evt met rimpel) aangelegd met een waarde die voor de motor past. Deze spanning wordt door C8 afgevlakt, en IC3 (een 7805 spanningsregelaar/stabilisator) verlaagt deze tot 5V. C10 en C11 zijn voorzien om storing op hoge frequenties (veroorzaakt door het schakelen van de 7805) kort te sluiten. Er is een stukje aluminium op IC3 gemonteerd om de warmte beter af te voeren. Ook is er nog een LED voorzien (LED5) die aangeeft of er voedingsspanning aanwezig is. Als beveiliging tegen het ompolen van de voedingsspanning (uit verstrooidheid) heb ik D1 (een gelijkrichter diode) toegevoegd.

6.1.5 Motordriver

Het ULN2003 IC heb ik voorzien van 4 LED's, die over de windingen staan en aangeven of de windingen van de motor bekrachtigd worden.

6.1.6 Connectoren

Om gemakkelijk uitbreidingen te kunnen aansluiten heb ik een aantal pinheaders voorzien. Hierop kunnen met gemak nog een lcd, schakelaars en de lichtsluis worden aangesloten.

6.2 Het bereiden van de print

Het vervaardigen van de print gebeurt in verschillende stappen:

1. *Het afdrukken van de layout:*

De eerste stap bestaat uit het afdrukken van de printlayout op ware grootte (figuur 6.5) op kalkpapier. Dit gebeurt best met een laserprinter omdat toner een betere lichtdichtheid heeft dan inkt die in inktjetprinters wordt gebruikt.

2. *Het belichten van de print:*

Het vel kalkpapier wordt op de lichtbak gelegd zodanig dat de inktlaag naar boven komt te liggen. Vervolgens wordt een printplaat genomen met fotolak (deze wordt zo verkocht), De lichtdichte afdeklaag wordt van de print getrokken en de print wordt op het kalkpapier gelegd. De lichtbak wordt gesloten en de lampen worden anderhalve minuut aangezet.

3. *Het ontwikkelen van de print:*

Tijdens het belichten wordt een oplossing van van 7 gram natrium hydroxide in 1 liter water voorbereid, en worden rubber handschoenen aangetrokken. Direct na het belichten wordt de print onmiddellijk in de oplossing ondergedompeld. Het NaOH zal er voor zorgen dat de fotolak waar licht is aangekomen oplost, en slechts de plaatsen waar koper op zal blijven nog bedekt zijn met een laklaag die ze tijdens het etsen zal beschermen tegen het zuur. Bij het onderdompelen moet er op gelet worden dat de laklaag niet aangeraakt wordt met de vingers. Om het ontwikkelproces vlugger te laten verlopen kan er met een fijne penseel over de laklaag worden gegaan.

4. *Het etsen:*

In de print wordt reeds een van de vier bevestigingsgaten geboord, waaraan een metaaldraad met isolatiemantel wordt bevestigd. Hierdoor is de print makkelijk hanteerbaar eens ze in het zuur zit. Een emmer wordt gevuld met water. In een plastic of glazen doos wordt 200 ml zoutzuur van 39% gegoten. De printplaat wordt ondergedompeld in het zuur zolang nog niet al het ongewenste koper weg is geëetst. De printplaat kan worden bewogen om het etsproces sneller te laten verlopen. Vervolgens wordt de printplaat in de emmer water ondergedompeld om het zuur er af te spoelen.

5. *Het testen:*

Met een ohmmeter wordt er onderzocht of er ohmse verbindingen zijn tussen naburige printbanen, indien dit zo is, kan het koper er met een mes tussen de banen worden weggekrabd.

6. *Het boren:*

Tenslotte dienen de gaten nog worden geboord in de print. Het boren van de kleine gaten wordt gedaan met een speciale printboormachine en een fijne boor.

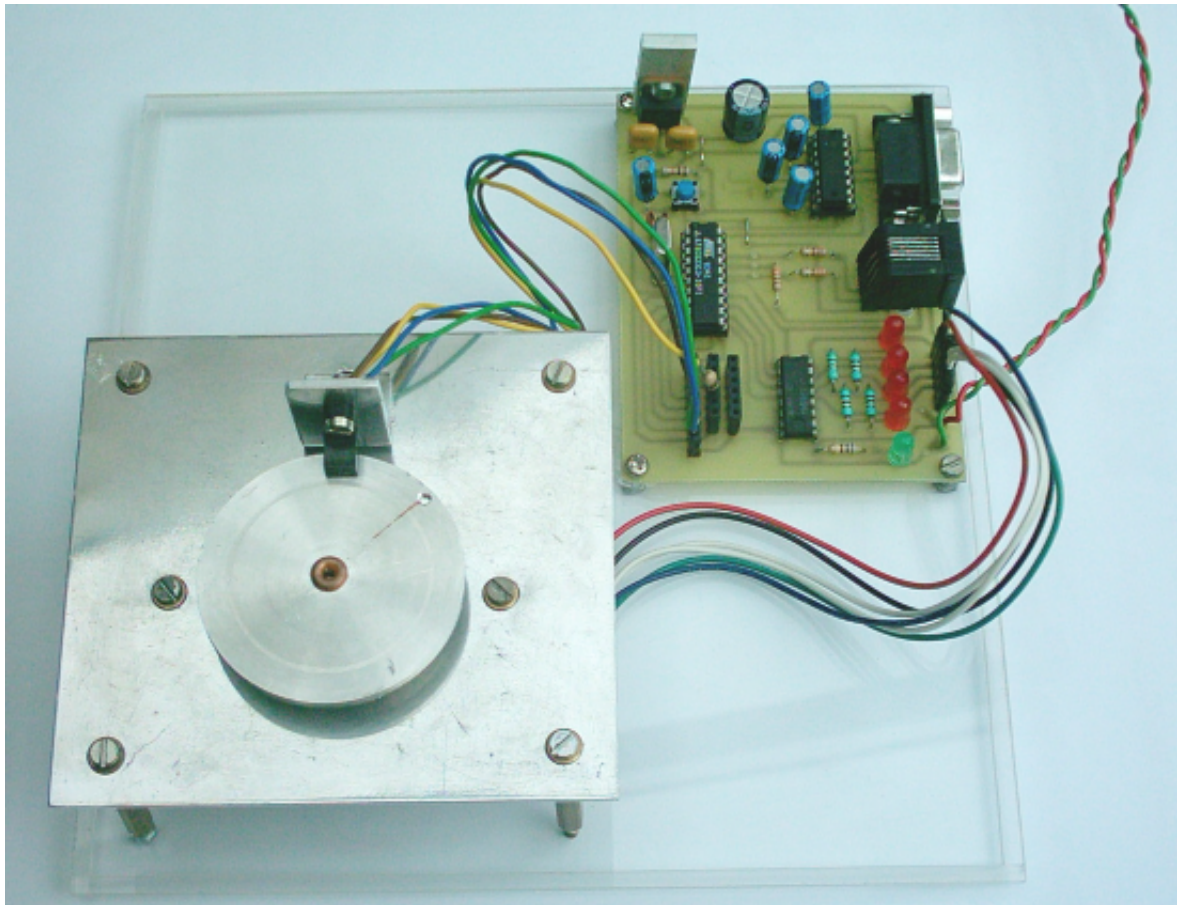
7. *Het solderen:*

Alle componenten kunnen nu op de print worden gesoldeerd.

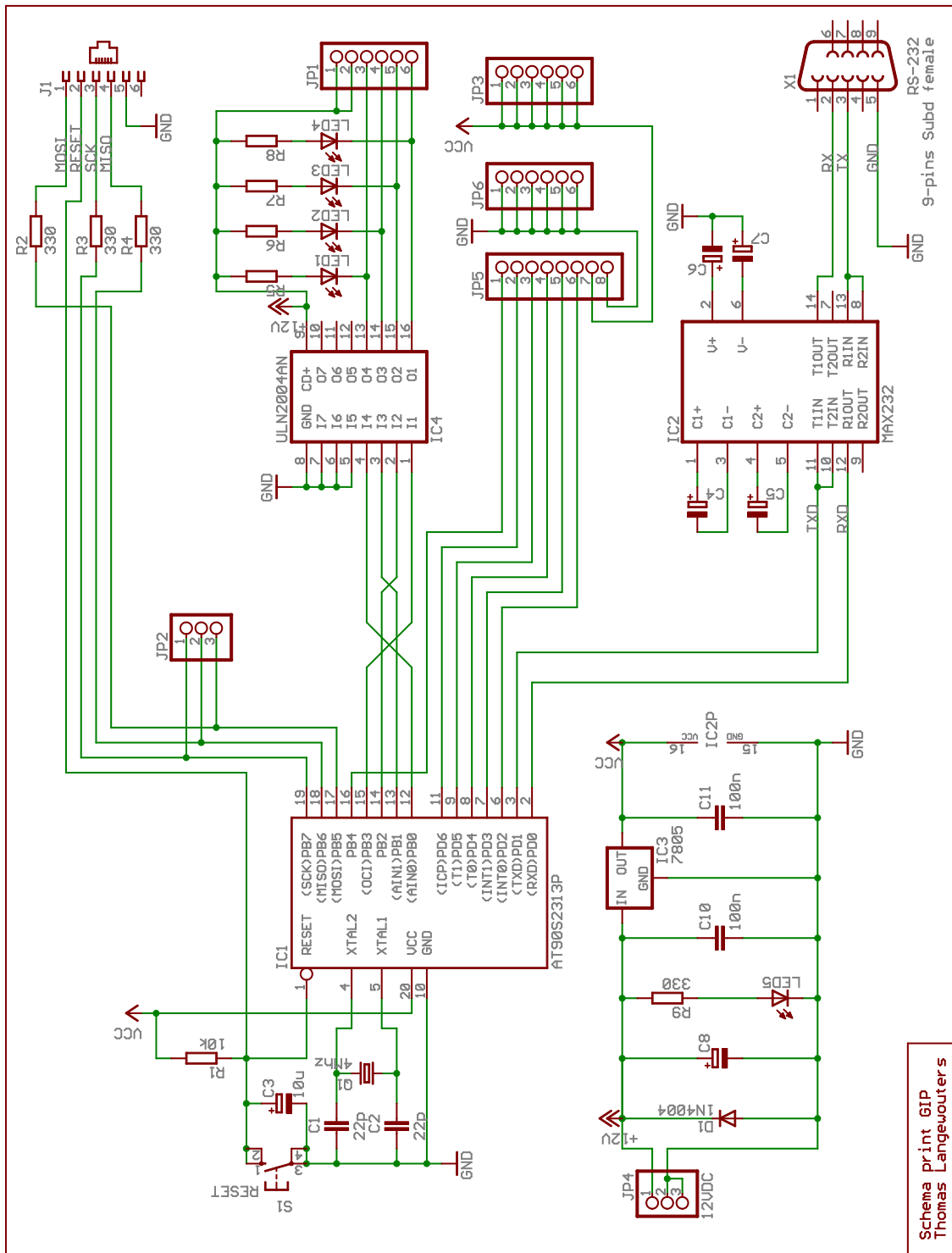
Het etsen ging vlot eens ik het onder de knie had. Ik heb twee keer moeten proberen. De eerste keer was de fotolak niet goed ontwikkeld, waardoor niet al het koper weg was geëetst. De tweede poging ging van een leien dakje. Er was wel een schoonheidsfoutje, het koper tussen twee naburige banen was niet goed weg waardoor de TX en RX van de μC aan elkaar hingen. Toen dit verholpen was werkte de print prima!

6.3 De didactische opstelling

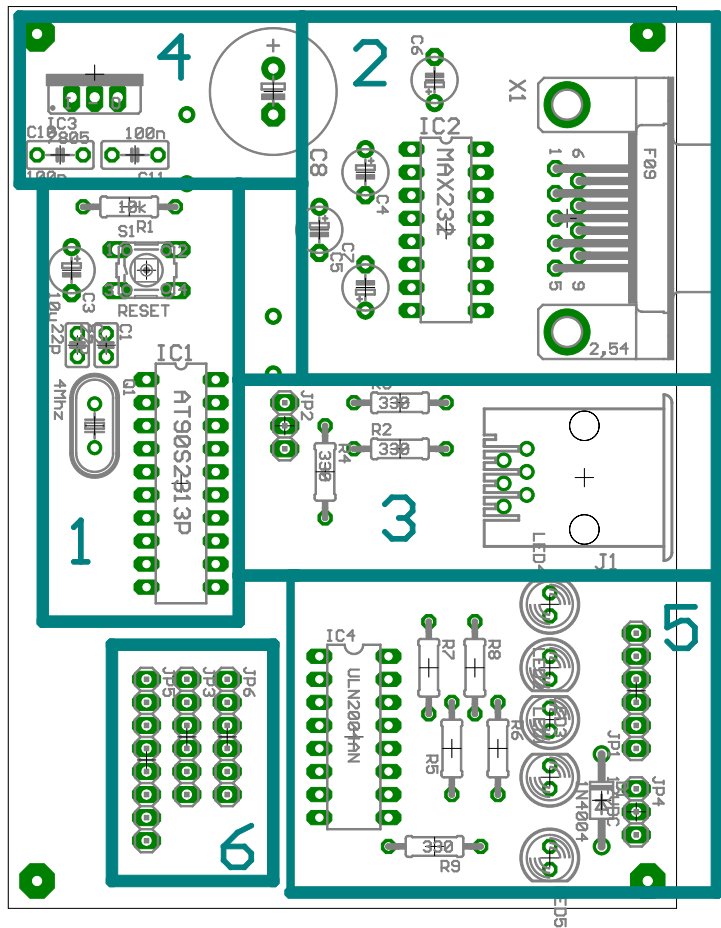
Ik heb een didactische opstelling gemaakt van de positiesturing die bestaat uit een plexiglazen plaat met daarop de print en de motor met de lichtsluis. Dit viel zeer goed mee dankzij het juiste gereedschap, en het resultaat mag er wezen.



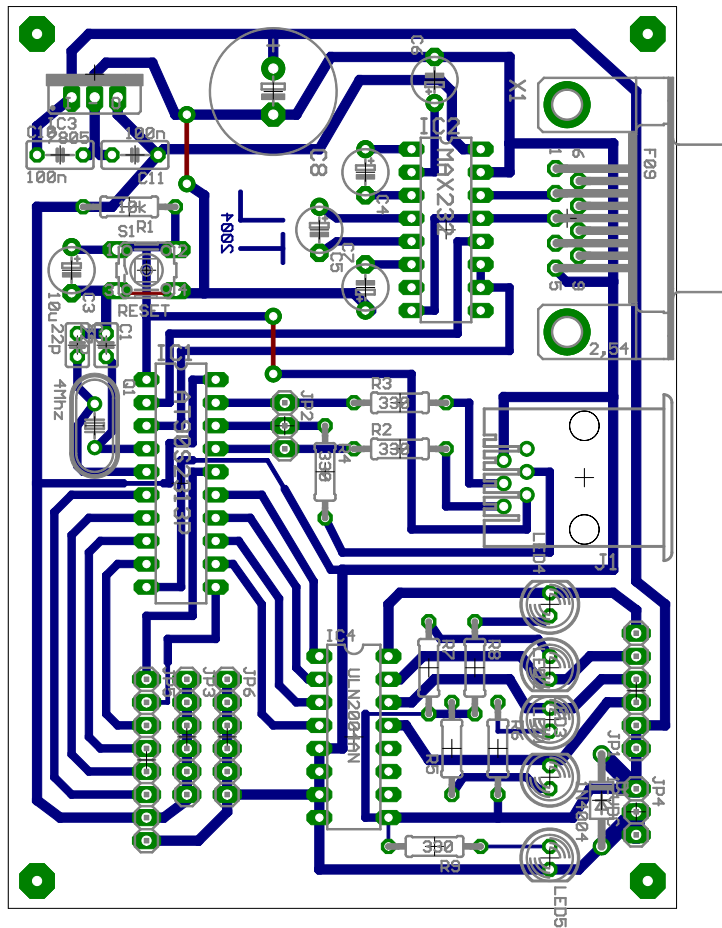
Figuur 6.1: De didactische opstelling



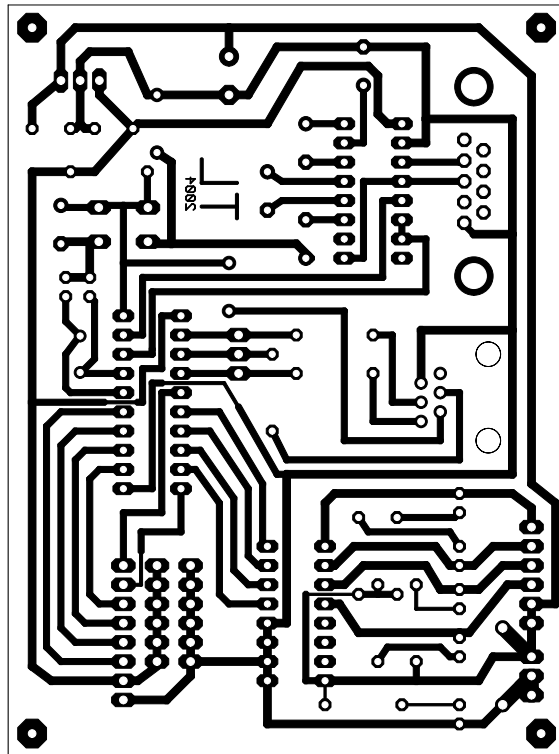
Figuur 6.2: Het elektrisch schema van de print



Figuur 6.3: De verschillende delen van de print aangeduid op de layout



Figuur 6.4: De print met banen en componenten



Figuur 6.5: De printlayout

Literatuurlijst

Stappenmotoren

Internetsites

- <http://www.ams2000.com/stepping101.html>
Step Motors Reference Guide
- <http://www.cs.uiowa.edu/~jones/step/#others>
Control of Stepping Motors, a tutorial. *door Douglas W. Jones*
- http://www.pages.drexel.edu/~twd25/stepper_appliance.html
Internet Appliance, stepper motor *door Todd Danko*
- <http://www.fontys.nl/elektro/medewerkers/tcramer/aandrijf/diktaat/EAPH9.pdf>
Stappenmotoren *door T.Cramer*
- <http://www.xs4all.nl/~sbolt/fotovoer.html>
De Fotovoer *door Steven Bolt*
- <http://users.pandora.be/vandewoestyne/eindwerk/node1.html>
Eindwerk Bart Vandewoestyne, *door Bart Vandewoestyne*
- <http://www.superior-solution.com/ProductsDB/Product.asp?ID=81>
Stepper motors *door SuperiorSolutions inc.*
- <http://www.geocities.com/zonnevolger/beginpagina.htm>
Zonnevolger *door Christophe alleine*
- <http://users.skynet.be/stevengeusens/stijngoffin.html>
Ontwerp van een autonoom voertuig *door Steven Geusens en Stijn Goffin*
- <http://www.earsc.com/HOME/engineering/TechnicalWhitePapers/TunedDamper/index.asp?SID=67>
Stepper Motor dampers *door E-A-R Specialty Composites inc.*

Microcontrollers

Websites

- http://www.dontronics.com/dt006_programming.html
DT006 Programming *door Dontronics*
- http://freespace.virgin.net/seant.ellis/avrrobot_hw.htm
AVR-based Robot Hardware *door Sean Ellis*
- <http://www.bsddhome.com/avrdude/>
AVRDUDE *door het AVRDUDE project*
- <http://www.ami.ac.uk/courseware/micros/restricted/ch7/micro07notes.html>
Microcontrollers *door AMI*
- <http://www.circuitsonline.net>
AVR Microcontroller tutorial *door Bastiaan Steenbergen*

Tijdschriftartikels

- AT90S2313 programmer
Elektuur, editie Juli-augustus 2003

Boeken

- Microprocessortechniek basistechniek
door S. Soenens en H. Vandenheede
- Digitale Technieken 3de graad
door H. Saeys en H. Vandenheede