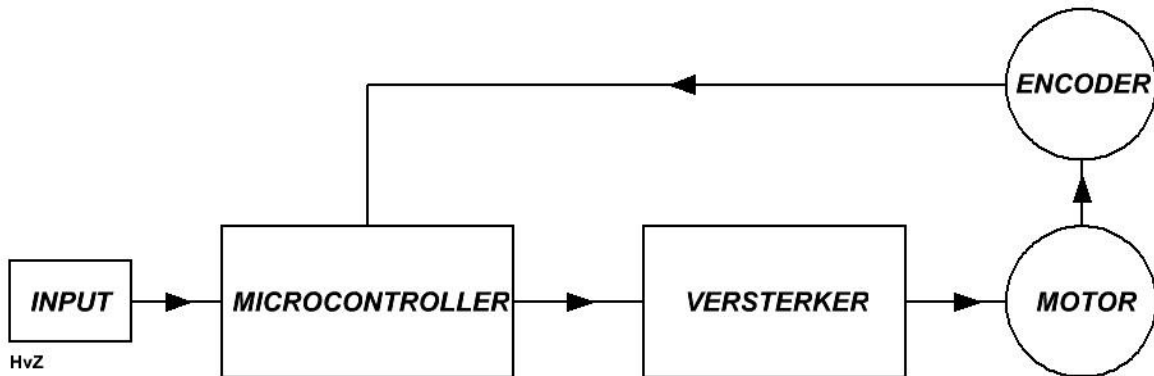
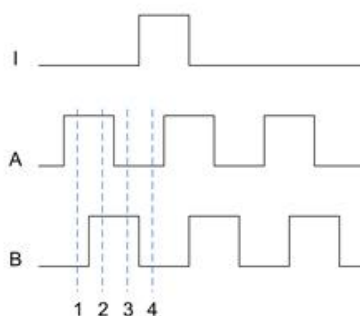
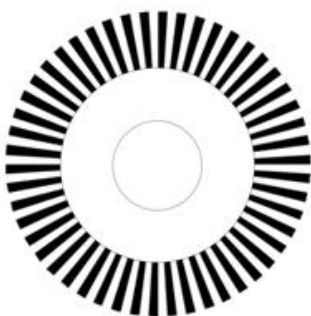


# WERKEN MET SERVO'S.

We gaan eens kijken wat een servo nu eigenlijk is. Er zijn verschillende servo systemen, je hebt ze bij hydrauliek, pneumatiek en bij elektrische systemen heb je DC, en AC systemen. Ze komen eigenlijk allemaal op hetzelfde neer; het zijn allemaal teruggekoppelde systemen. Ik ga proberen om in begrijpelijke taal uit te leggen hoe zo iets werkt. Hieronder een blokschema van zo'n systeem.



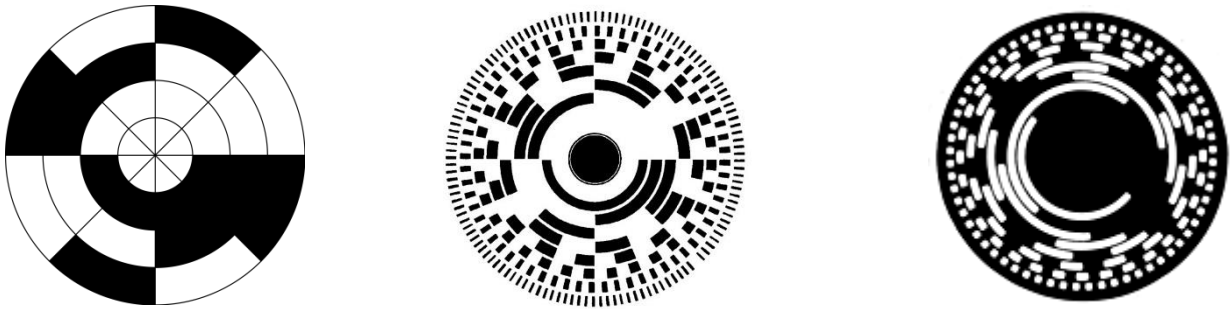
Zoals je in het blokschema kan zien bestaat zo'n systeem uit een controller, een versterker en een encoder. Bij de input wordt een waarde ingegeven waar de motor-as naar toeloopt. Dat signaal gaat van de controller naar de versterker, de versterker is een eindtrap die de motor aanstuurt. Aan de motor zit (in dit geval) een encoder gekoppeld, maar dat kan ook bijvoorbeeld een potmeter zijn, of een resolver, of een tachogenerator of een combinatie van de genoemde componenten. Een encoder geeft een X-aantal pulsen per omwenteling en kan naast het toerental ook de stand van de motor-as en de draairichting van de motor-as aangeven. Encoders heb je in een paar types. Er zijn optische encoders en magnetische encoders. De optische encoders bestaan uit een ronde doorzichtige schijf met streepjes erop en die draait tussen een lichtsluis. Je hebt ook schijfjes van metaal met gaatjes of met sleufjes erin.



Zo'n schijfje ziet er zo uit. Het signaal wat uit een encoder komt bestaat uit blokgolven. Vaak zijn dat drie signalen, maar soms ook twee. In het voorbeeld zijn het er drie, het A en B signaal en het zogenaamde index signaal. Het A en B signaal is 90 graden in fase verschoven ten opzichte van elkaar; op deze manier

kan er gekeken worden of de motor links- of rechtsom draait. Het index signaal komt eenmaal per omwenteling voorbij. Je hebt trouwens twee types. Het type hierboven is een incrementele encoder, maar je hebt ook absolute encoders. De absolute encoder behoudt te allen tijde zijn waarde, dus ook na stroomuitval. De incrementele encoder heeft dat niet, bij stroomuitval zijn alle gegevens weg.

Hieronder een paar voorbeelden van zulke encoderschijven.



Zoals je kan zien verschillen ze nogal van patroon, dat heeft ook met het aantal posities te maken die de schijf afgeeft. Ik had het daarnet ook over een magnetische encoder, een magnetische encoder bestaat uit een schijf van magnetisch materiaal, voorzien van bijvoorbeeld tanden en een opnemer die naast of tegenover de schijf is geplaatst. De opnemer zet, als gevolg van het verdraaien van de tanden, optredende variaties in het magnetisch veld om in elektrische pulsen. De puls geveer geeft een puls trein af waarbij de frequentie evenredig is aan het toerental van de motor-as, als deze tenminste direct op de motor-as gemonteerd is. Hij kan ook op een reductiekast gemonteerd worden. Het encodersignaal wordt terug gekoppeld naar de controller, de controller regelt dan het toerental en de positie van de motor-as. Op deze manier krijg je dus een gesloten regelkring. Het zijn dan ook complexe en dure regelingen.

Er zijn ook servomotoren waar de encoder, versterker en controller ingebouwd zijn in de servomotor.



Hiernaast zie je een voorbeeld van zo'n servo systeem. Je ziet dat alles in één unit is ingebouwd, zo'n systeem heeft alleen een voeding nodig en een signaal om de positie door te geven. De positie doorgeven wordt vaak door een bus-systeem gedaan. Dit kan gebeuren door bijvoorbeeld Profibus, maar er zijn ook andere veldbus-systemen.

Je hebt natuurlijk ook servo systemen met een losse servo driver en motor. In de servo driver zit de controller en de eindversterker die de motor bestuurt. Zo'n systeem wordt vaak door een PLC of een PC aangestuurd. De systemen worden vaak in de machinebouw gebruikt voor het snel positioneren van assen. Voor bijvoorbeeld CNC machines zijn het uitermate geschikte systemen.

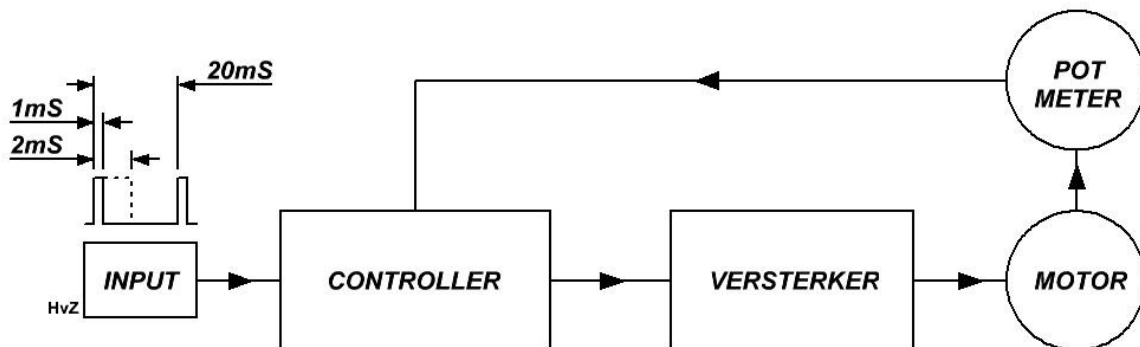
Hieronder zie je zo'n losse servo driver.



Dit is wel een drie fase AC servo driver, maar deze heb je ook voor DC. Vaak hebben dit soort drivers ook nog wat I/O aan boord voor diverse functies, de I/O poorten kunnen vaak naar eigen inzicht geprogrammeerd worden. Vaak zit er tegenwoordig ook een Ethernet of Profibus aansluiting op om ze op afstand uit te lezen of te bedienen.

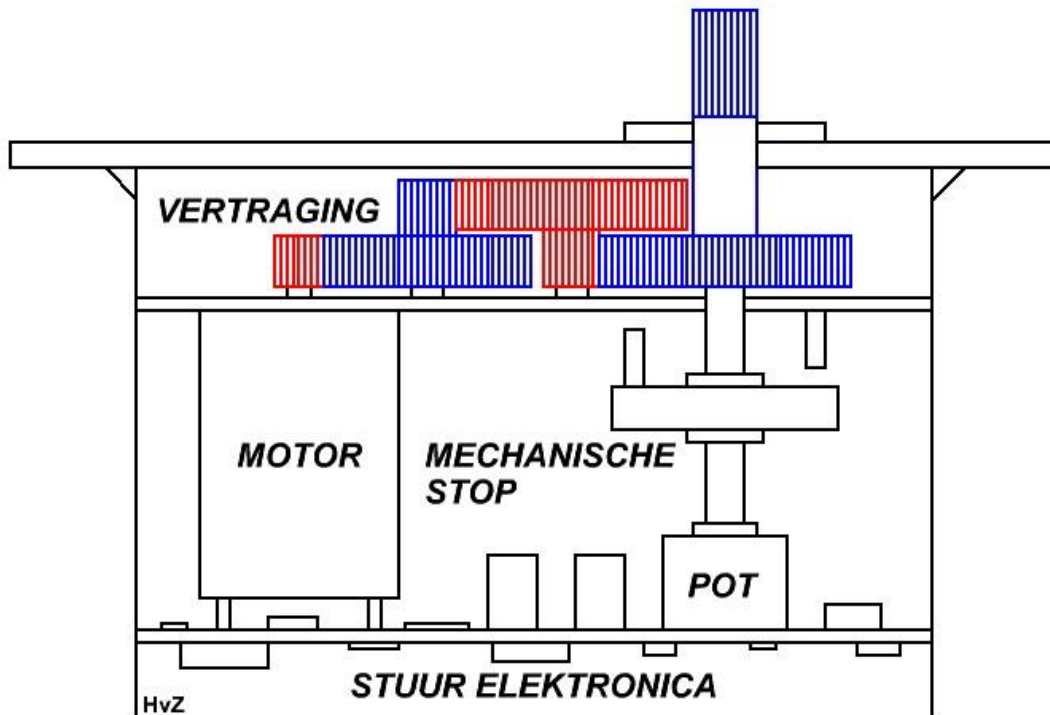
Dit zijn dure systemen en daar ga je niet zo snel mee experimenteren, maar gelukkig zijn er ook goedkope systemen. Die worden vaak in de modelbouw gebruikt en die noemen we dan ook modelbouw servo's. Dat zijn wel vaak systemen die maar een X-aantal graden versteld kunnen worden,

maar ze vallen toch onder de noemer servo. De werking is dan ook een beetje hetzelfde, het zijn ook gesloten regelsystemen, net als de grotere broers. Zie voorbeeld hieronder.



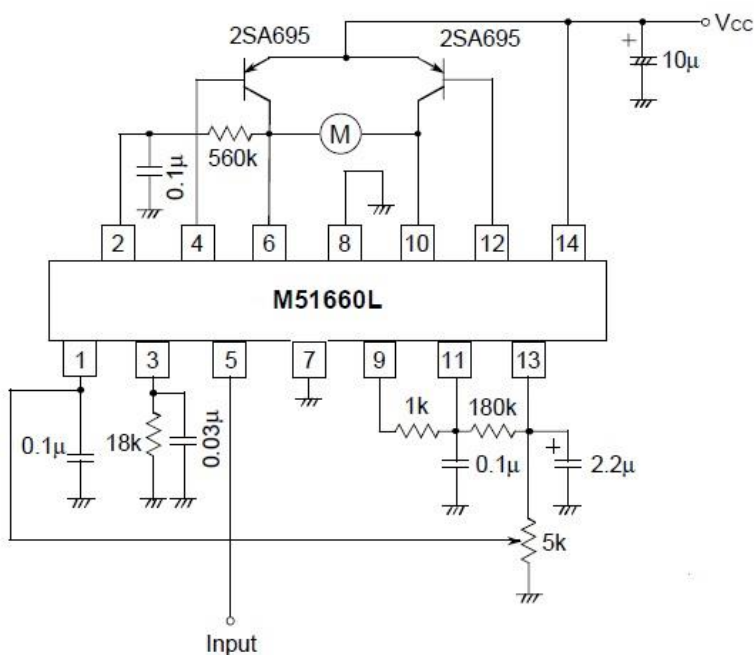
Zoals je kan zien verschilt het blokschema niet zoveel van zijn grote broer. Het inputsignaal is wel anders, hier is het een PWM signaal met een vaste frequentie van 50Hz en een variabele puls breedte die varieert tussen de 1 en 2 milliseconden. Met een signaal van 1 milliseconden staat de servo bijvoorbeeld helemaal links en met een signaal van 2 milliseconden staat de servo helemaal rechts. Als het signaal 1.5 milliseconden bedraagt staat de servo precies in de middenstand. De as van een modelbouw servo kan +/- zo'n 180 graden (verschilt per servo) verdraaien. Het inputsignaal is meestal afkomstig van een zender en ontvanger. Het zendersignaal wordt in de ontvanger omgezet naar een variabele puls breedte. De elektronica in de servo (tegenwoordig bij digitale servo's is dat een microcontroller) stuurt de eindversterker aan die op zijn beurt de motor bedient. Het teruggekoppelde potmetersignaal zorgt ervoor dat de servo naar zijn gewenste positie loopt. Dus wat de encoder doet bij de industriële systemen, doet de potmeter bij de modelbouw servo's. Niets anders dus dan de gewenste positie vergelijken met de positie van de motor-as.

Hieronder zie je hoe zo'n modelbouw servo is opgebouwd.



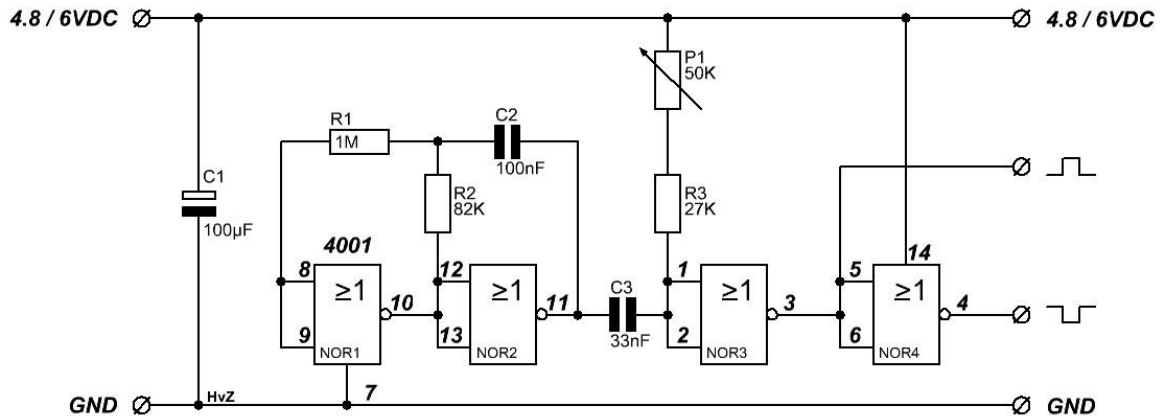
De servo bestaat uit een motor, een vertraging, de potmeter voor de positie bepaling, een mechanische stop (deze is belangrijk, want die beschermt de potmeter tegen te ver doordraaien) en niet te vergeten de besturingselektronica. De modelbouw servo heeft drie aansluitdraden, een plus en een min en de draad waar het PWM signaal op binnen komt.

Hieronder één van de mogelijke schema's van de besturingselektronica.



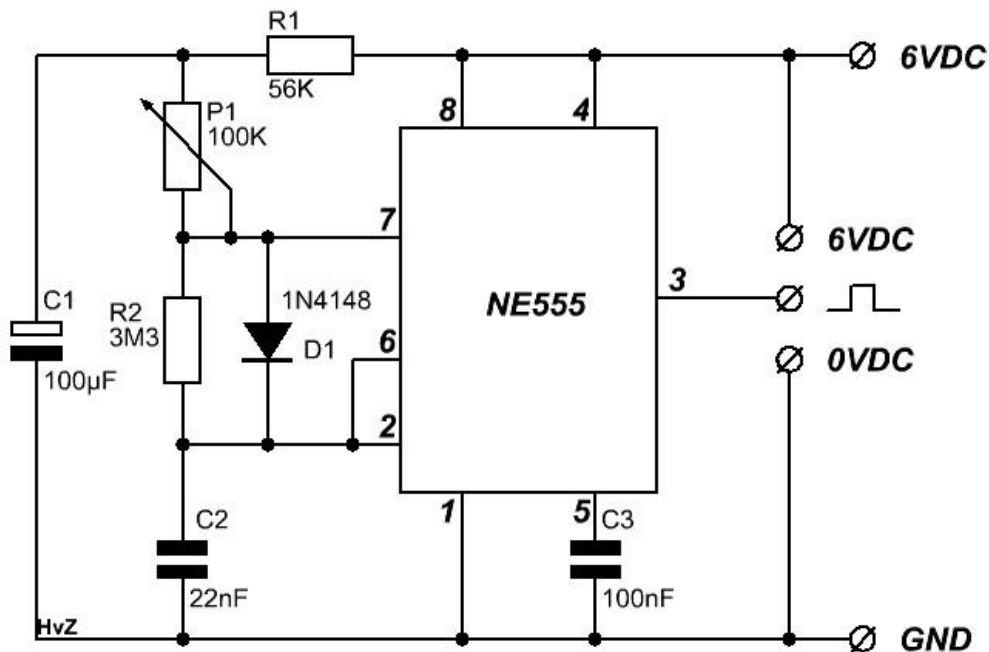
Op het schema kun je zien hoe zo'n besturing is opgebouwd. Het hart van de besturing is de M51660L, dit IC regelt alles. Het enige wat nodig is, zijn nog wat randcomponenten voor de juiste werking. Op pen 1 komt het potmetersignaal binnen, daar wordt de servo positie op ingelezen. Pen 2 en 3 zijn voor de timing. Pen 4 en 12 sturen de externe PNP transistoren aan. Pen 5 is het ingangssignaal dat bijvoorbeeld van de ontvanger komt. Op pen 6 en pen 10 zitten de interne transistoren, die samen met de externe transistoren de motor aansturen. Pin 9 is de error puls uitgang. Pen 11 de stretcher ingang.

Pen 13 verzorgt de spanning voor de potmeter. Pen 7 en pen 8 zijn de GND aansluitingen en pen 14 is de voeding voor het IC. Zo heb je een idee hoe een modelbouw servo is opgebouwd. Aan de besturing in de servo kan je weinig veranderen, maar het besturen van een modelbouw servo kan op verschillende manieren. We hadden het al over de ontvanger die voor dit signaal zorgt, maar dat kan ook anders. Je kan bijvoorbeeld met een 4001 (een nor poort) een prima puls generator maken waar je dan de servo mee kan bedienen. Zie schema hieronder.

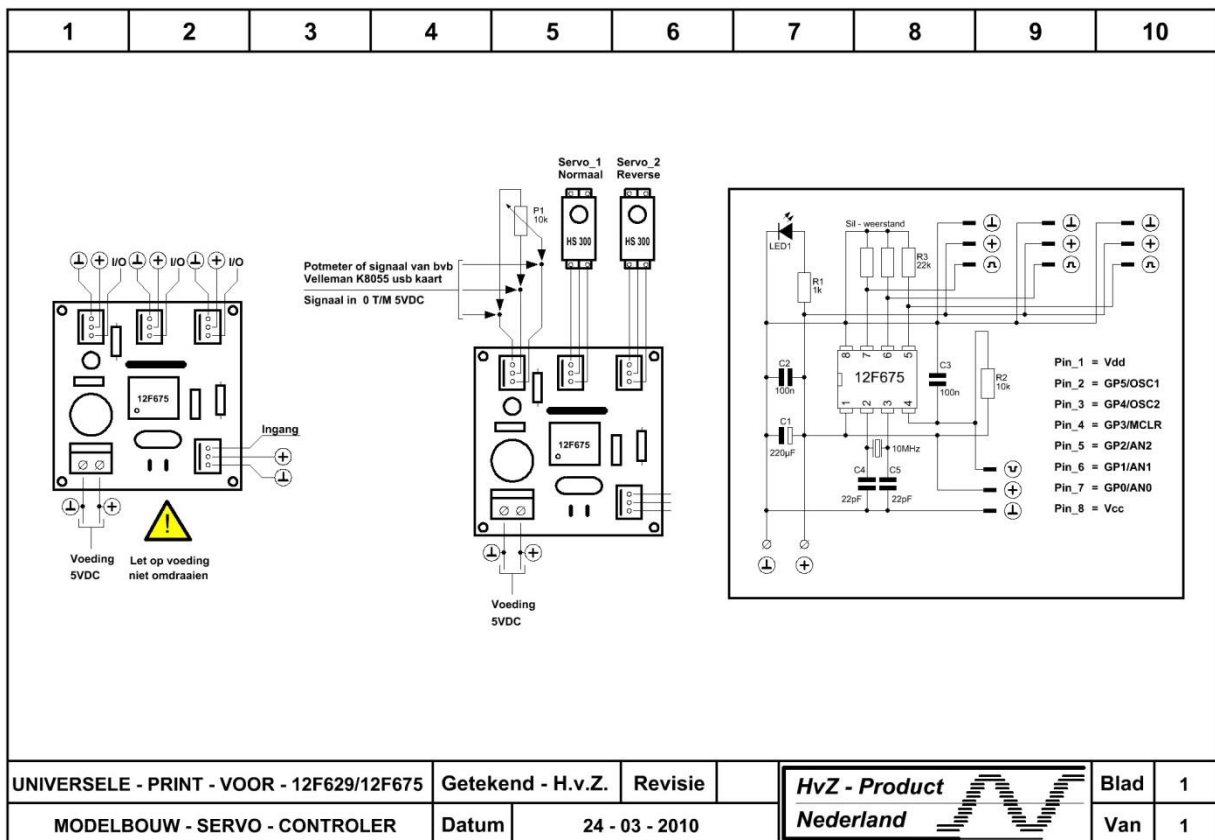


De schakeling kan gevoed worden met een spanning tussen de 4.8 en 6VDC. Er komen twee pulsen af, de ene is geïnverteerd ten opzichte van de ander. Dit kan makkelijk zijn omdat er servo's zijn die een geïnverteerd signaal nodig hebben. Zo kan je op een eenvoudige manier dus je servo bedienen.

En zo kunnen we met een NE555 ook een puls generator maken om een modelbouw servo te bedienen. Zie schema hieronder.



Het is ook mogelijk om het met een microcontroller te doen, met bijvoorbeeld een 12F629 of een 12F675 kan dat heel simpel gerealiseerd worden. Hieronder het schema.



Hier het programma wat in de controller staat voor het bedienen van de servo.

```

Device 12F675 ; Processor type

Xtal 10 ; Kristal 10 Mhz

Config WDT_OFF, _ ; WatchDog Timer uit
        PWRTE_ON, _ ; Power-up Timer Enable aan
        MCLRRE_OFF, _ ; Externe Master Reset Enable uit
        HS_OSC ; X-tal groter dan 4MHz

All_Digital true ; Alle poorten digitaal

Symbol servo_1 = GPIO.1 ; Servo uitgang_1
Symbol servo_2 = GPIO.2 ; Servo uitgang_2

Declare Adin_Res = 8 ; Resolutie 8 bits
Declare Adin_Tad = frc ; Set rc osc
Declare Adin_Stime = 50 ; Sample tijd 5

Dim potmeter As Byte ; Variabele waarde potmeter
Dim positie As Word ; Variabele waarde positie
Clear ; wis geheugen
    
```

```

        ;543210                ; Hulpregel
GPIO    = %000000            ; Maak poort laag
TRISIO  = %001001            ; In en uitgangen

        ;76543210            ; Hulpregel adcon register
ADCON0  = %00000001         ; A/D control register

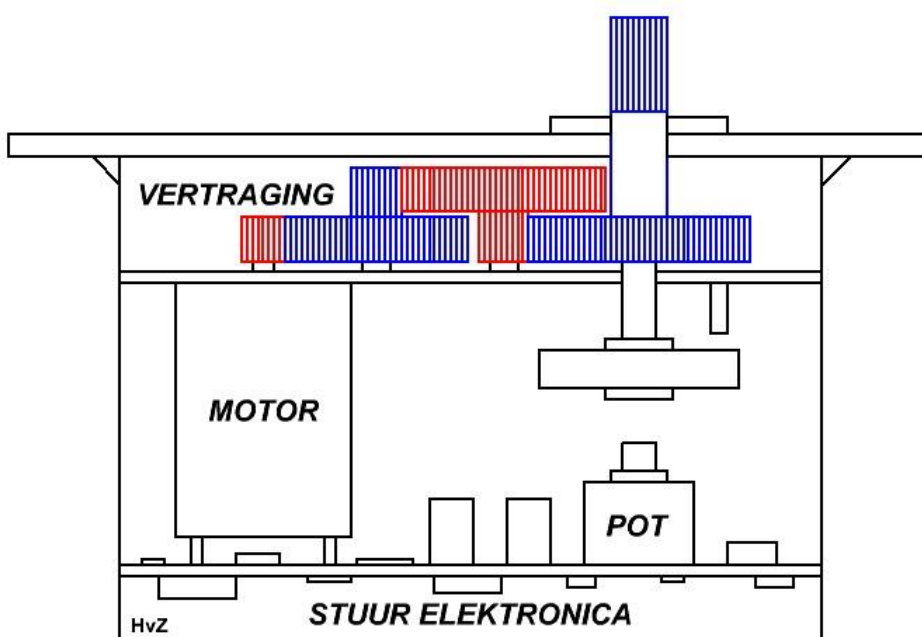
run:                                          ; Run programma
    potmeter = ADIn 0                ; Potmeter op analoog_0
    positie = 850 + (potmeter*7)      ; Berekening positie servo
    Servo servo_1,positie            ; Uitsturing positie servo_1
    Servo servo_2,positie            ; Uitsturing positie servo_2
    DelayMS 20                       ; Pauze 20 Ms
GoTo run                                  ; Ga naar run programma

End                                         ; Einde programma

```

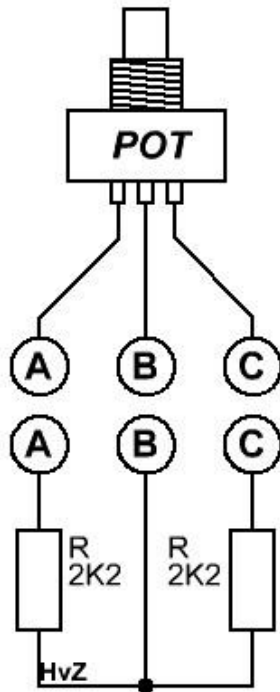
Zoals je kan zien stelt het programma niet veel voor, op de analoge ingang (ADIN\_0) komt de potmeter die de servo gaat bedienen. De minimale positie van de servo is 850, dat is de waarde als de potmeter op nul staat. De maximale waarde is  $850 + (255 \times 7) = 850 + 1785 = 2635$ . Die waardes vallen buiten de 1 en 2 milliseconden, maar de ene servo pikt dat wel en de andere servo niet. Dat ligt dus aan de constructie van de servo. Ik heb een HITEC HS300 gebruikt en die haalt het wel. Dus als je gaat experimenteren kan je de waarde achter de positie aanpassen, plus potmeter \* 7.

We hebben nu gezien hoe je een servo heen en weer kan laten lopen, daar is hij ook voor bedoeld. Maar met een kleine aanpassing kan je er ook een mooie aandrijving mee maken die gewoon rondjes kan draaien, bijvoorbeeld om een robot mee aan te drijven. Wat we dan doen is de mechanische stop eruit halen en de potmeter loshalen van de servo-as. De potmeter zet je nu in de middenstand en die lijm je bijvoorbeeld vast. Je kan de potmeter ook vervangen door twee weerstanden van gelijke waarde. In het schema van de standaard besturingselektronica staat dat de weerstand van de potmeter 5K is. Dus je zou twee weerstanden van bijvoorbeeld 2K2 kunnen nemen, maar je kan ook de potmeter laten zitten. Hieronder zie je het voorbeeld, kijk wat er veranderd is ten opzichte van de standaard servo.



Als je de potmeter wil vervangen kan je dat op deze manier doen.

Zie voorbeeld hieronder.



Als je de potmeter verwijdert en je zet de twee weerstanden zo neer als in het voorbeeld, dan krijg je dezelfde waarde als met de potmeter in de middenstand.

Ik heb nu 2K2 weerstanden genomen, maar dat kan ook een andere waarde zijn natuurlijk. Als je twijfelt kan je de waarde van de potmeter even opmeten, dan weet je het helemaal zeker. Meet de waarde tussen de twee buitenste pootjes en deel de waarde door twee. Als daar een waarde uitkomt waar geen weerstand voor is, dan kies je de waarde die er zo dicht mogelijk bij in de buurt komt. Voor de weerstanden kan je gewoon 0.6Watt types nemen. Ik zou wel 1% weerstanden nemen, dan heb je de kleinste afwijking.

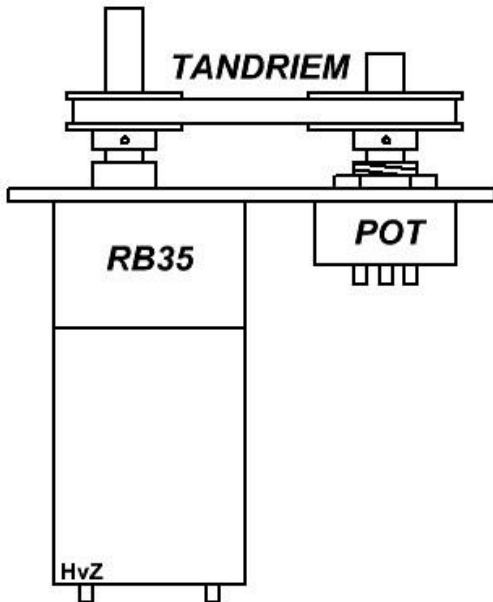
Als je de weerstanden gemonteerd hebt kan je de servo nu gebruiken als aandrijving om bijvoorbeeld een robot te laten rijden. Je hebt er dan wel twee nodig natuurlijk

Als je de servo nu aansluit op bijvoorbeeld één van de drie voorbeelden die ik gegeven heb, (de besturing met de NE555, de besturing met de controller, of de besturing met de window comparator) dan moet de servo stilstaan als de potmeter van de besturing in de middenstand staat. Als je nu de potmeter iets verdraait zal de servo langzaam gaan draaien, bijvoorbeeld rechtsom. Als je de potmeter wat verder draait gaat de servo sneller draaien. Draai je de potmeter nu terug naar de middenstand, dan zal de servo weer stil staan. Draai je de potmeter nu de andere kant op, dan loopt de servo linksom. Dus nu kan je de servo zowel rechtsom als linksom sturen en in toerental regelen.

En hoe komt het nu dat de servo zich anders gedraagt na de aanpassing? Dat komt omdat de interne potmeter zijn werkt niet meer doet. De servo is zijn positie terugkoppeling kwijt. De servo wordt wel in toeren geregeld en een bepaalde kant op gestuurd, maar hij weet niet meer waar hij moet stoppen.

Maar zoals ik al eerder zei, kun je er op deze manier toch leuke dingen mee doen. Je zou er bijvoorbeeld een lineaire actuator mee kunnen maken, met bijvoorbeeld een spindel, een tandriem, tandheugel, enzovoort. Je zou bijvoorbeeld een nep hydraulische cilinder kunnen maken met een stuk draadeind en een kleine aangepaste servo, leuk om een hydraulische kraan mee te maken. En zo is er genoeg te verzinnen.

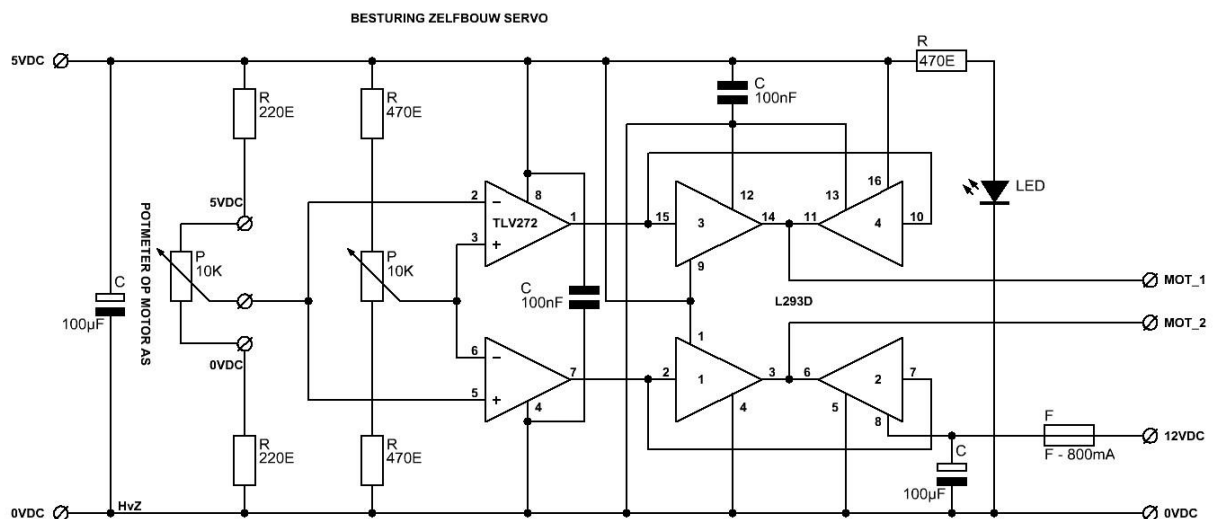
We hebben het over de in de handel zijnde servo's gehad, maar je kan er ook een zelf maken natuurlijk. Het voordeel is dan dat je een servo kan maken die enorm veel koppel kan leveren en het is tevens een leuk experimenteerproject. Zie hieronder een voorbeeld hoe je zo iets kan maken.



In het voorbeeld heb ik een tandriem gebruikt, maar het kan ook met bijvoorbeeld tandwielen. Of je zet de potmeter met een koppeling direct op de motor-as, alleen kan je dan slechter iets op de motor-as monteren. De motor kan je dan bijvoorbeeld aansturen met een L293D, dat is een H\_brug in IC vorm. De brug kan je dan weer aansturen met bijvoorbeeld een opamp die als window comparator staat geschakeld, de potmeter (die op de motor zit) wordt dan teruggekoppeld naar de opamp. Op deze manier kan je zelf een servo maken. Let er wel op dat de potmeter goed aangesloten moet worden, anders blijft de motor doorlopen en gaat de potmeter stuk.

**EVEN VOOR DE GOEDE ORDE, DEZE ZELF GEMAAKTE SERVO'S KAN JE NIET MET EEN MODELBOUWZENDER- / ONTVANGER BEDIENEN.**

Hieronder een schema van de besturing. Voor de hysteresis kan je nog wat experimenteren met de 220 Ohm weerstanden of je zet -in plaats van één weerstand van 220 Ohm- een potmeter van 5K.



Deze besturing kunnen we ook met een microcontroller maken, we gebruiken hier een 16F887 voor maar dat mag ook een ander type zijn natuurlijk. Hieronder het programma.

```

Device 16F887                ; Processor type

Xtal 10                       ; Cristal 10Mhz

Asm                            ; Config settings
CONFIG_REQ
__CONFIG __CONFIG1, HS_OSC & WDT_OFF & DEBUG_OFF & FCMEN_OFF & LVP_OFF &
IESO_OFF & BOR_OFF & CPD_OFF & CP_OFF & MCLRE_OFF & PWRTE_ON
__CONFIG __CONFIG2, WRT_OFF & BOR40V
EndAsm

All_Digital true              ; Alle poorten digitaal

Declare Adin_Res = 8           ; resolutie 8 bits
Declare Adin_Tad = frc         ; set RC osc
Declare Adin_Stime = 50        ; sample tijd 5

Declare LCD_RSPin PORTD.2      ; Reset display poort D.2
Declare LCD_ENPin PORTD.3      ; Enable display poort D.3
Declare LCD_DTPin PORTD.4      ; Data display poort D.4 t/m D.7

Symbol UIT1 = PORTB.0          ; Naar ingang H_brug
Symbol UIT2 = PORTB.1          ; Naar ingang H_brug

Dim WAARDE_1 As Byte           ; Variabele in programma
Dim WAARDE_2 As Byte           ; Variabele van potmeter

Cls                             ; Wis display

DelayMS 500                    ; Pauze 0.5 sec

Clear                          ; Wis geheugen

PORTA    ;543210                ; Hulpregel poort A
PORTA = %000000                ; Maak poort A laag
TRISA = %111111                ; Poort_A I/O

PORTB    ;543210                ; Hulpregel poort B
PORTB = %000000                ; Maak poort B laag
TRISB = %000000                ; Poort_B I/O

PORTC    ;76543210             ; Hulpregel poort C
PORTC = %00000000             ; Maak poort C laag
TRISC = %00000000             ; Poort_C I/O

PORTD    ;76543210             ; Hulpregel poort D
PORTD = %00000000             ; Maak poort D laag
TRISD = %00000000             ; Poort_D I/O

PORTE    ;210                  ; Hulpregel poort E
PORTE = %000                  ; Maak poort E laag
TRISE = %111                  ; Poort_E I/O

ADCON0    ;76543210           ; Hulpregel analoog
ADCON0 = %00000001           ; ADCON0 register analoog

ANSELH    ;543210             ; Hulpregel analoog poort_B
ANSELH = %000000             ; ANSEL register analoog poort_B

```

```

;-----
; PROGRAMMA ZELFBOUW SERVO.
;-----

positie_0:
    WAARDE_1 = 0

    WAARDE_2 = ADIn 0

    Print At 1,1,"GA NAAR POS 0"
    Print At 2,1,"POS"
    Print At 2,5,Dec ADIn 0, " "

    If WAARDE_1 > WAARDE_2 + 1 Then
        UIT1 = 1
        Else UIT1 = 0
        If WAARDE_1 < WAARDE_2 Then
            UIT2 = 1
            Else UIT2 = 0
        EndIf
    EndIf

    If WAARDE_2 = 0 Then
        GoTo wacht_0
    EndIf
GoTo positie_0

wacht_0:
    DelayMS 5000
    Cls
GoTo positie_1

positie_1:
    WAARDE_1 = 200

    WAARDE_2 = ADIn 0

    Print At 1,1,"GA NAAR POS 200"
    Print At 2,1,"POS"
    Print At 2,5,Dec ADIn 0, " "

    If WAARDE_1 > WAARDE_2 + 1 Then
        UIT1 = 1
        Else UIT1 = 0
        If WAARDE_1 < WAARDE_2 Then
            UIT2 = 1
            Else UIT2 = 0
        EndIf
    EndIf

    If WAARDE_2 = 200 Then
        GoTo wacht_1
    EndIf
GoTo positie_1

wacht_1:
    DelayMS 5000
    Cls
GoTo positie_2

```

```

positie_2:
    WAARDE_1 = 75

    WAARDE_2 = ADIn 0

    Print At 1,1,"GA NAAR POS 75"
    Print At 2,1,"POS"
    Print At 2,5,Dec ADIn 0, " "

    If WAARDE_1 > WAARDE_2 + 1 Then
        UIT1 = 1
    Else UIT1 = 0
        If WAARDE_1 < WAARDE_2 Then
            UIT2 = 1
        Else UIT2 = 0
        EndIf
    EndIf

    If WAARDE_2 = 75 Then
        GoTo wacht_2
    EndIf
GoTo positie_2

wacht_2:
    DelayMS 5000
    Cls
GoTo positie_3

positie_3:
    WAARDE_1 = 150

    WAARDE_2 = ADIn 0

    Print At 1,1,"GA NAAR POS 150"
    Print At 2,1,"POS"
    Print At 2,5,Dec ADIn 0, " "

    If WAARDE_1 > WAARDE_2 + 1 Then
        UIT1 = 1
    Else UIT1 = 0
        If WAARDE_1 < WAARDE_2 Then
            UIT2 = 1
        Else UIT2 = 0
        EndIf
    EndIf

    If WAARDE_2 = 150 Then
        GoTo wacht_3
    EndIf
GoTo positie_3

wacht_3:
    DelayMS 5000
    Cls
GoTo positie_4

```

```

positie_4:
    WAARDE_1 = 25

    WAARDE_2 = ADIn 0

    Print At 1,1,"GA NAAR POS 25"
    Print At 2,1,"POS"
    Print At 2,5,Dec ADIn 0, " "

    If WAARDE_1 > WAARDE_2 + 1 Then
        UIT1 = 1
    Else UIT1 = 0
        If WAARDE_1 < WAARDE_2 Then
            UIT2 = 1
        Else UIT2 = 0
        EndIf
    EndIf

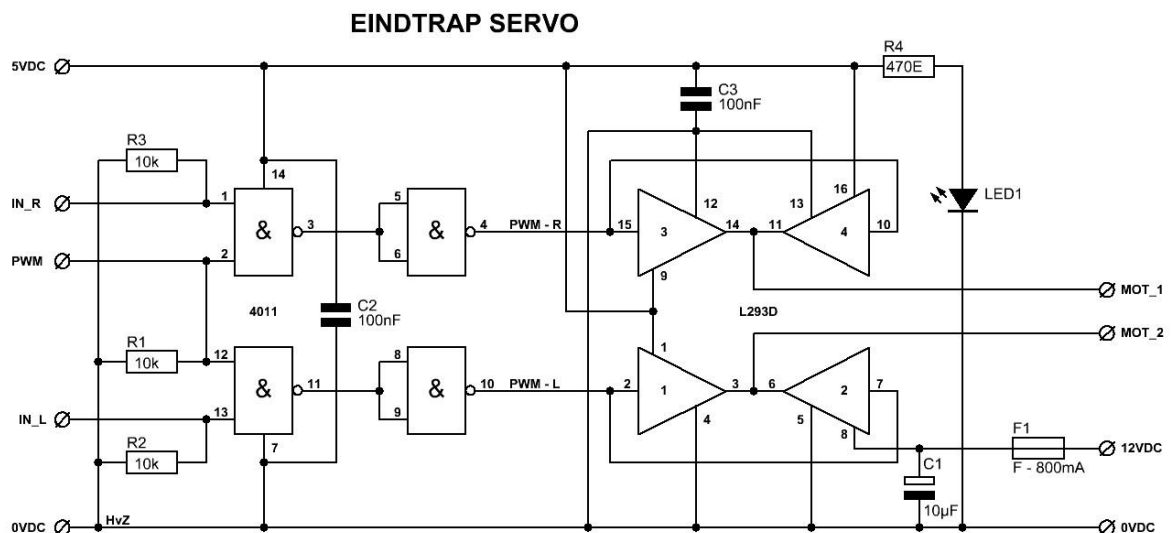
    If WAARDE_2 = 25 Then
        GoTo wacht_4
    EndIf
GoTo positie_4

wacht_4:
    DelayMS 5000
    Cls
GoTo positie_1

End

```

Hier hoort weer een ander schema bij, er wordt wel weer een L293D gebruikt.  
Hieronder het schema.

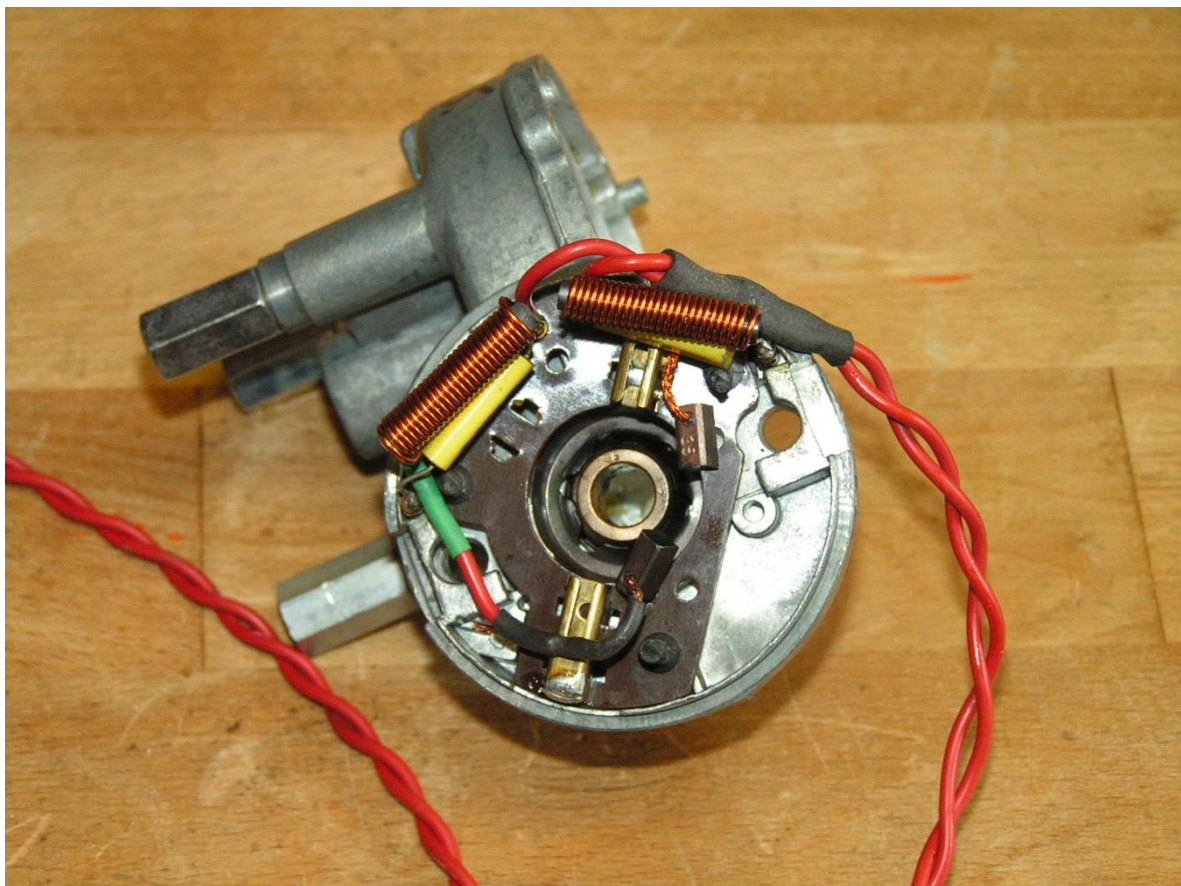


Port B.0 en port B.1 van de controller komen aan de klemmen IN\_R en IN\_L van de eindtrap. De PWM klem komt aan de 5VDC te zitten, omdat deze niet gebruikt wordt in het programma. Als de PWM functie in het programma erbij gezet wordt (dan moet het programma aangepast worden) dan kan de PWM klem van de eindtrap op de HPWM pin (poort C.1 of poort C.2) van de controller aangesloten worden. Op deze manier kan dan ook het toerental van de servo geregeld worden. Let er wel op dat de PWM frequentie niet groter dan 5kHz mag zijn voor de L293D.

Op de klemmen MOT\_1 en MOT\_2 wordt de motor aangesloten, op de klemmen 5VDC en 0VDC komt de 5VDC voeding voor de elektronica en op de klemmen 12VDC en 0VDC komt de 12VDC voeding voor de motor. De externe potmeter (de potmeter die op de motor-as zit) komt aan de analoge ingang A.0 (ADIN 0). Denk er ook hier weer aan dat de potmeter goed aangesloten moet worden.

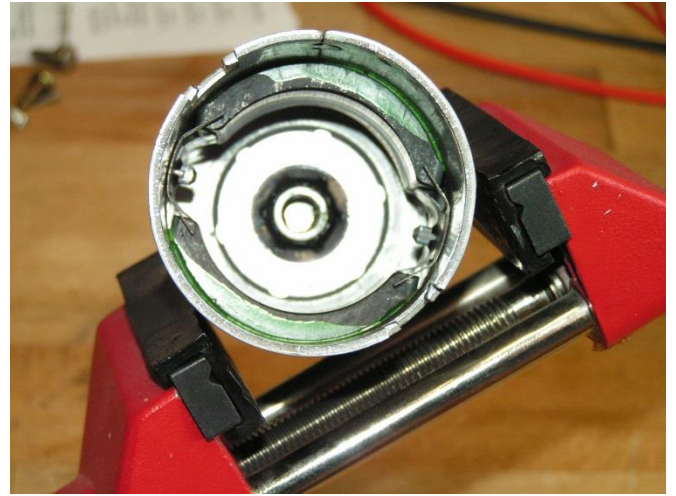
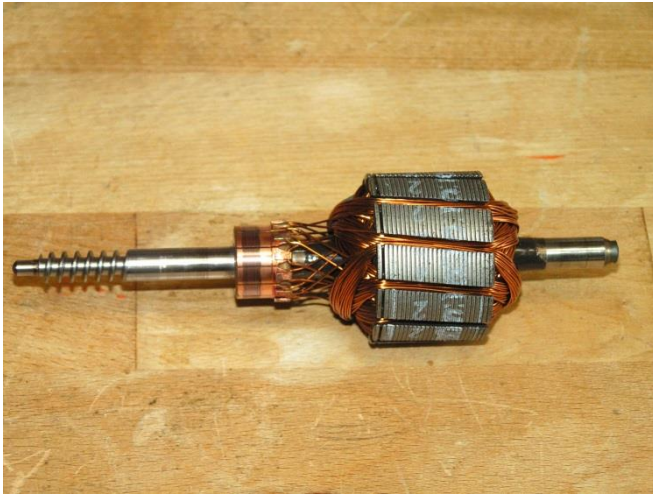
Het programma loopt automatisch vier posities af, maar dat kan naar eigen wens aangepast worden natuurlijk. Let er ook op dat de voorbeelden met de L293D alleen geschikt is voor kleine motoren, zoals de RB35, dus motoren die meer dan 1.5A trekken gaan niet met deze schakeling.

We gaan nog een servo maken van een ruitenwissermotor. Maar we moeten dan eerst de motor helemaal aanpassen. Ik ga met behulp van wat foto's uitleggen wat er moet gebeuren.



Ik had deze motor al een keer aangepast, dus heb hem voor wat foto's maar even uit elkaar gehaald. Op de foto kan je de borstelbrug zien, deze moet aangepast worden. Als je onder het linker spoeltje kijkt (één van de twee ontstoringfilters) dan zie je daar twee sleufjes in de borstelbrug zitten, daar heeft nog een koolborstelhouder gezeten. Die zit er standaard op bij de meeste ruitenwissermotoren en is voor het hoge toerental. Die heb je niet nodig, dus die haal ik dan ook altijd weg. Zoals je kan zien op de foto heb je nu twee borstels tegenover elkaar zitten. De spoeltjes staan in serie met de borstels en de nieuwe aansluitdraden. De condensatoren (de gele blokjes) zitten aan de ene kant aan de spoel en aan de andere kant aan de behuizing, maar daar is in principe niets aan veranderd. Dus als je zo'n motor aan wil passen wijst alles eigenlijk voor zich.

Hieronder de foto's van het anker en van het huis met de permanentmagneten.



Zoals je kan zien op de foto's is zowel het anker als het huis nog in prima staat.

Aan het anker hoeft dus niets te gebeuren, tenzij de collector slecht is. Die kan je eventueel iets opschuren (op bijvoorbeeld de draaibank) of iets afdraaien. Hierbij is wel de nodige voorzichtigheid geboden, want als je niet oppast maak je het alleen maar erger. Ook moet je de lamellen (dat zijn de stukjes tussen de koper vlakken van de collector) iets wegfrezen, dus als je de spullen daar niet voor hebt kan je er beter niet aan beginnen.

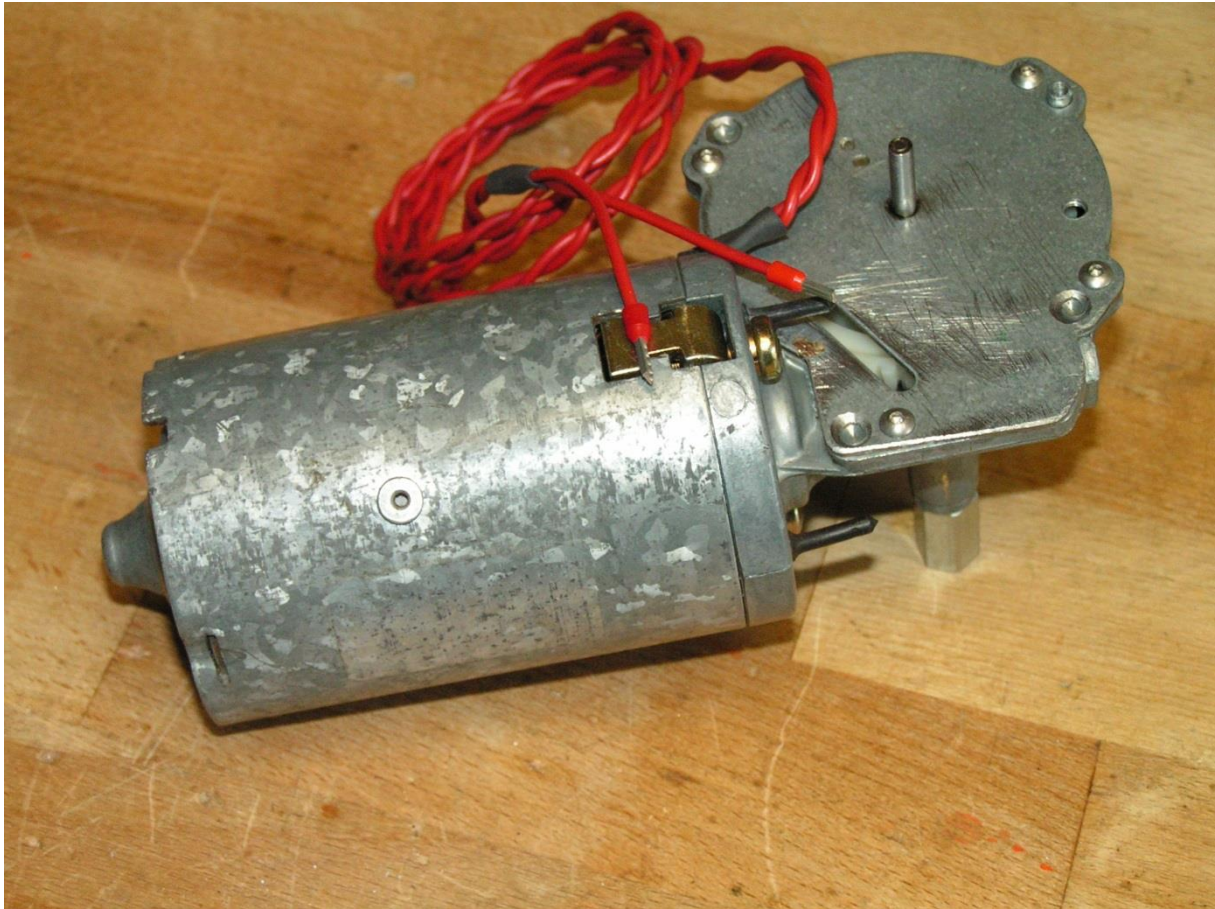
Dan hebben we het tandwiel waar de worm in loopt (het wormwiel) Zie foto's hieronder.



Daar heb ik een as van 4mm op gemaakt om bijvoorbeeld een encoder schijfje of een potmeter op te bevestigen. Ik heb op de draaibank een gat van 3,9mm in de as geboord en uitgeruimd naar 4mm. In het asje van 4mm heb ik met een centerpunt kleine putjes geslagen, zodat deze strak in het gat van 4mm past. Het asje is met Loctite 638 in het gat geperst en zit dus muurvast. Op deze manier kun je makkelijk een potmeter op de as monteren. Je kan de potmeter ook op dezelfde wijze monteren als in het voorbeeld met de RB35 motor natuurlijk, maar ik vind het op deze manier mooier.

Je kunt op deze manier ook makkelijker iets op de hoofdas monteren.

Hieronder heb je een foto van het geheel weer gemonteerd.



Je ziet op de foto de afdekplaat zitten (dat is de plaat die over de worm en het wormwiel zit). Daar zie je naast de schroefjes putjes zitten. Daar zit de afdekplaat standaard mee vastgeklonken om de motor uit elkaar te halen moet je die dus wegboren. Om de plaat weer vast te zetten heb ik nieuwe gaatjes geboord en schroefdraad (M3) getapt. Op deze manier kan je de afdekplaat weer goed vast zetten.

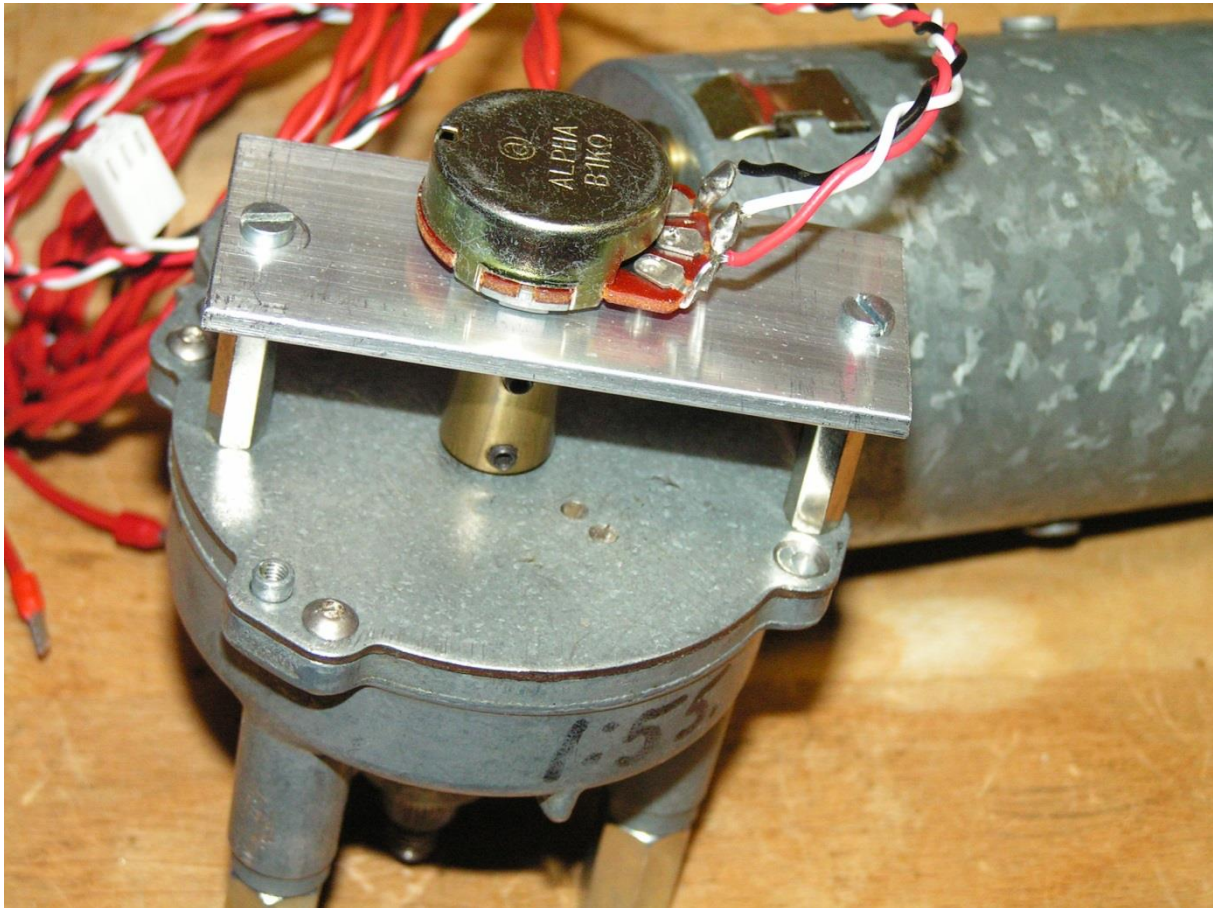
Je ziet ook een sleuf gat in de afdekplaat zitten; daar hebben de sleepcontacten aangezeten. Die zijn in kunststof ingegoten en in de afdekplaat geperst. Die moet je dus ook verwijderen, want die worden niet meer gebruikt. Op het wormwiel zaten ook koperplaatjes, ook die mag je verwijderen.

Om het anker in de borstelplaat te plaatsen, plak ik één borstel even met een stukje tape vast. De andere borstel houd ik zelf even op zijn plek. Op deze manier kan je het anker makkelijk in de plaat monteren. Niet vergeten om het plakband weg te halen.

Bij het anker in het magneethuis plaatsen moet je wel even goed opletten. Je moet het anker goed vasthouden anders vliegt hij je magneethuis in, met het gevolg dat je borstels weer uit je borstelhouders schieten en dan kan je weer opnieuw beginnen. Zelf houd ik hem altijd bij de worm vast, dan heb je lekker grip op het geheel en kan er weinig gebeuren. Op deze manier kan je een

ruitenwissermotor dus ombouwen. Niet alleen om er een servo mee te maken, maar ook voor andere toepassingen. Het zijn sterke motoren en ze kunnen tegen een stootje.

Hieronder een foto met de potmeter op de motor gemonteerd.



De potmeter is op een aluminium hoeklijntje gemonteerd, de verbinding tussen de potmeter en de as is met een messing koppeling gedaan. Deze koppelingen zijn zo te koop, dus die hoeft je niet zelf te maken. De koppeling heeft een gat van 4 en 6mm. Het geheel staat op twee afstandhouders en is daarmee op de motor gemonteerd. Let er wel op dat alles precies uitgelijnd is, anders komt er teveel zijdelingse kracht op de potmeter. Het geheel is nu klaar om gebruikt te worden.

Om een goed werkend geheel te krijgen moet er zo min mogelijk speling op de worm en wormwiel zitten. Vaak zit er op de kopse kant (in de behuizing boven de worm) van de motor een stelschroef, daar kan de axiale speling van het anker en worm mee afgesteld worden. Die speling moet minimaal zijn, anders gaat de motor staan oscilleren. Stel de schroef ook weer niet te strak af, anders loopt de motor te zwaar.

Voordat je de motor aansluit op de besturing kan je het beste één van de stelschroeven (waar de potmeter mee op de as zit) even losgezet worden. Dit is om te controleren of de potmeter goed aangesloten zit. Als je de bedienpotmeter verdraait moet de motor in een bepaalde richting gaan lopen. Als je nu de potmeter die op de as zit in dezelfde richting draait als de motor loopt, moet de motor op een bepaald punt stoppen. Doet de motor dat niet dan moet je de aansluitdraden van de

motor omdraaien. Als alles wel goed werkt, maar je wil dat de motor de andere kant op moet lopen, dan moet je de aansluitdraden van de motor omdraaien en de plus en min draden van de potmeter.

Hieronder het programma voor de servo regeling.

```
Device 16F887 ; Processor type

Xtal 10 ; Cristal 10Mhz

Asm ; Config settings
CONFIG_REQ
__CONFIG __CONFIG1, HS_OSC & WDT_OFF & DEBUG_OFF & FCMEN_OFF & LVP_OFF &
IESO_OFF & BOR_OFF & CPD_OFF & CP_OFF & MCLR_OFF & PWRTE_ON
__CONFIG __CONFIG2, WRT_OFF & BOR40V
EndAsm

All_Digital true ; Alle poorten digitaal

Declare Adin_Res = 8 ; resolutie 8 bits
Declare Adin_Tad = frc ; set RC osc
Declare Adin_Stime = 50 ; sample tijd 5

Declare LCD_RS_Pin PORTD.2 ; Reset display poort D.2
Declare LCD_EN_Pin PORTD.3 ; Enable display poort D.3
Declare LCD_DT_Pin PORTD.4 ; Data display poort D.4 t/m D.7

Declare CCP1_Pin PORTC.2 ; HPWM 1 op poort C.2

Symbol UIT1 = PORTC.0 ; Naar ingang H_brug
Symbol UIT2 = PORTC.1 ; Naar ingang H_brug

Symbol VERSTERKING = 140 ; Instelling reactie snelheid
Symbol FREQUENTIE = 8000 ; Instelling frequentie
Symbol HYSTERESIS = 2 ; Instelling hysteresis

Dim WAARDE_1 As Byte ; Variabele in programma
Dim WAARDE_2 As Byte ; Variabele van potmeter

Cls ; Wis display

DelayMS 500 ; Pauze 0.5 sec

Clear ; Wis geheugen

;543210 ; Hulpregel poort A
PORTA = %000000 ; Maak poort A laag
TRISA = %111111 ; Poort_A I/O

;543210 ; Hulpregel poort B
PORTB = %000000 ; Maak poort B laag
TRISB = %000000 ; Poort_B I/O

;76543210 ; Hulpregel poort C
PORTC = %00000000 ; Maak poort C laag
TRISC = %00000000 ; Poort_C I/O

;76543210 ; Hulpregel poort D
PORTD = %00000000 ; Maak poort D laag
TRISD = %00000000 ; Poort_D I/O
```

```

;210 ; Hulpregel poort E
PORTE = %000 ; Maak poort E laag
TRISE = %111 ; Poort_E I/O

;76543210 ; Hulpregel analoog
ADCON0 = %00000001 ; ADCON0 register analoog

;543210 ; Hulpregel analoog poort_B
ANSELH = %000000 ; ANSEL register analoog poort_B

;-----
; PROGRAMMA SERVOREGELING MET RUITENWISSERMOTOR.
;-----

```

```

RUN:
    WAARDE_1 = ADIn 0

    WAARDE_2 = ADIn 1

    HPWM 1, VERSTERKING, FREQUENTIE

    ;Print At 1,1,Dec3 WAARDE_1,
    ;Print At 2,1,Dec3 WAARDE_2,

    If WAARDE_1 > WAARDE_2 + HYSTERESIS Then
        UIT1 = 1
        UIT2 = 0
        Else UIT1 = 0
            If WAARDE_1 < WAARDE_2 Then
                UIT2 = 1
                UIT1 = 0
                Else UIT2 = 0
            EndIf
        EndIf
    EndIf
GoTo RUN

End

```

Zoals je ziet is het maar een klein programma.

De werking is als volgt: op de analoge ingang 0 wordt de bedienpotmeter aangesloten en op analoge ingang 1 de potmeter die op de motor-as zit. Deze twee waardes worden met elkaar vergeleken en als de waardes gelijk zijn zal de motor stilstaan. De HPWM uitgang zet een PWM signaal op de eindtrap en verzorgt de draaisnelheid van de as. Hoe hoger de PWM waarde is, hoe sneller de motor-as verdraait. Ik heb er versterking achter gezet, omdat ook de stroom toeneemt als de PWM waarde hoger ingesteld wordt.

```

Symbol VERSTERKING = 140
Symbol FREQUENTIE = 8000
Symbol HYSTERESIS = 2

```

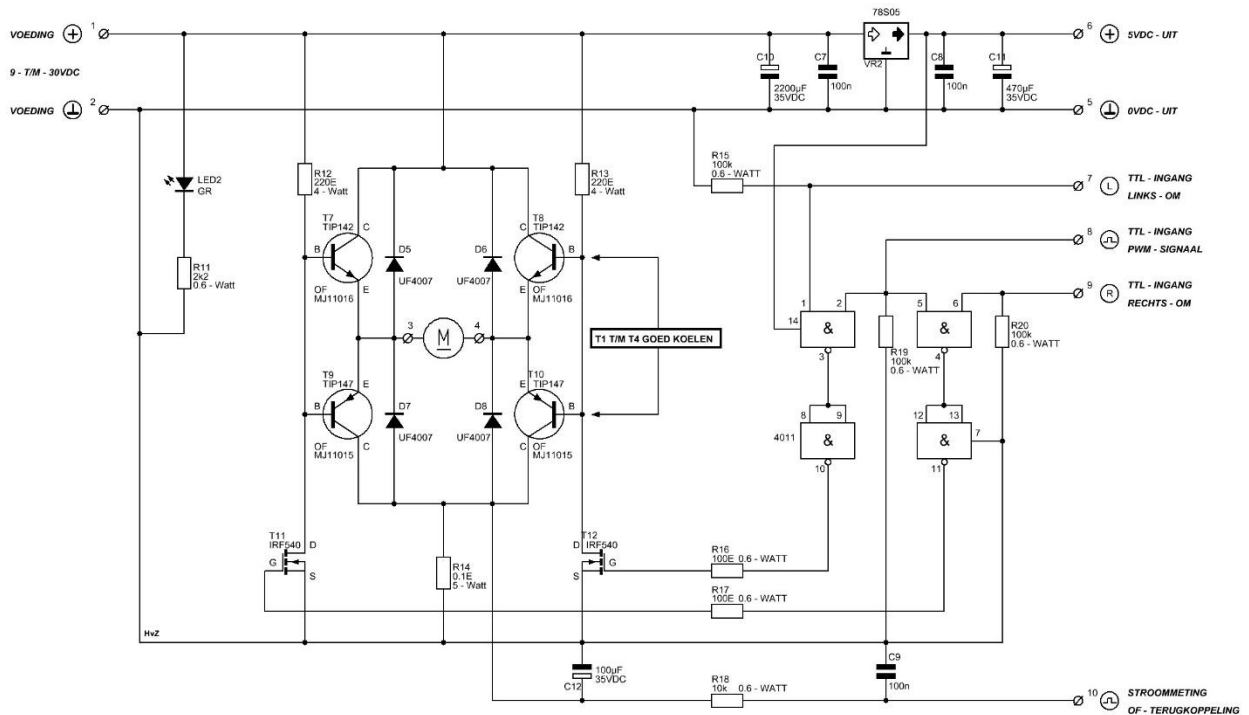
Achter versterking, frequentie en de hysteresis kan de waarde aangepast worden. De frequentie staat hier op 8KHz, kan je aanpassen maar is niet echt nodig. Met de versterking en de hysteresis kan je wat spelen, net zolang tot de motor netjes draait. Als alles goed reageert hoor je een knerpend geluid tijdens het verdraaien van de motor-as.

**Let erop als je een hevel op de motor as zet, je er niet met je vingers tussenkomt. Zo'n motor is behoorlijk sterk.**

Ik heb ook een LCD display aangesloten voor het testen van de servo. Als alles goed loopt kan de printfunctie uitgezet worden, zoals in het voorbeeld gebeurd is met de haakjes voor de printopdracht.

Dit programma kan ook gebruikt worden voor de servo met de RB35 motor.

Hieronder het schema van de eindtrap voor de servo regeling.



De eindtrap mag gebruikt worden met een voedingsspanning tussen de 9 en 30VDC, **let op de schakeling is niet tegen ompolen beveiligd**. De H\_brug kan opgebouwd worden met transistoren van het type TIP142 en TIP147, of met het type MJ11016 en MJ11015. Ik heb de TIP versie gebruikt, en die zijn prima voor ruitenwissermotoren. Voor motoren tot 200 Watt kan je de MJ11016 en MJ11015 gebruiken.

De aansluitingen van de microcontroller naar de eindtrap zijn; UIT1 (PORTC.0) komt op (TTL INGANG-LINKSOM), UIT2 (PORTC.1) komt op (TTL INGANG RECHTSOM), HPWM 1 (PORTC.2) komt op (TTL-INGANG PWM SIGNAAL), en de 0VDC (GND) van de controller komt op de 0VDC (GND) van de eindtrap.

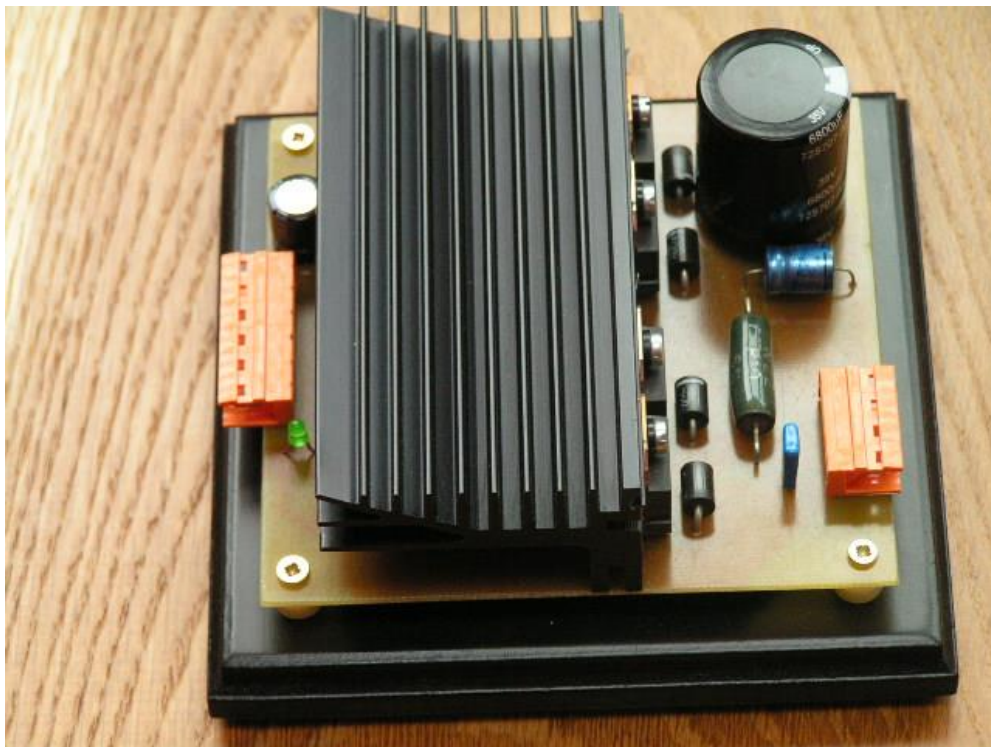
De stroommeet uitgang van de eindtrap wordt hier niet gebruikt, maar als je daar mee wilt experimenteren kan dat. De uitgang komt dan rechtstreeks op een analoge ingang van de controller te zitten. Op deze manier kan je dan nog stroomcompensatie toepassen of de eindtrap beveiligen tegen overstroom.



Je kan ook kiezen om de transistoren met veerklemmen op het koelblok vast te zetten, je hebt dan wel een koelblok nodig waar je de veren in kan drukken. Dat werkt een stuk makkelijker dan met M3 schroeven en met isolatieringen. Maar dat is een eigen keuze natuurlijk. Ik heb de transistoren met schroeven en de isolatieringen vastgezet.

Voor de rest wijst het voor zich, denk ik.

Hieronder nog een paar foto's van de eindtrap, onder het koelblok zit de rest van de componenten.



Ik ga ook nog een voorbeeld geven van een servo voor het bedienen van een klep of kraan. Dit type wordt veel in het bedrijfsleven gebruikt. Dit type servo (BELIMO SR24A-SR) wordt veel toegepast in zwembad installaties, klimaat regelingen en bij verwarmingsinstallaties.

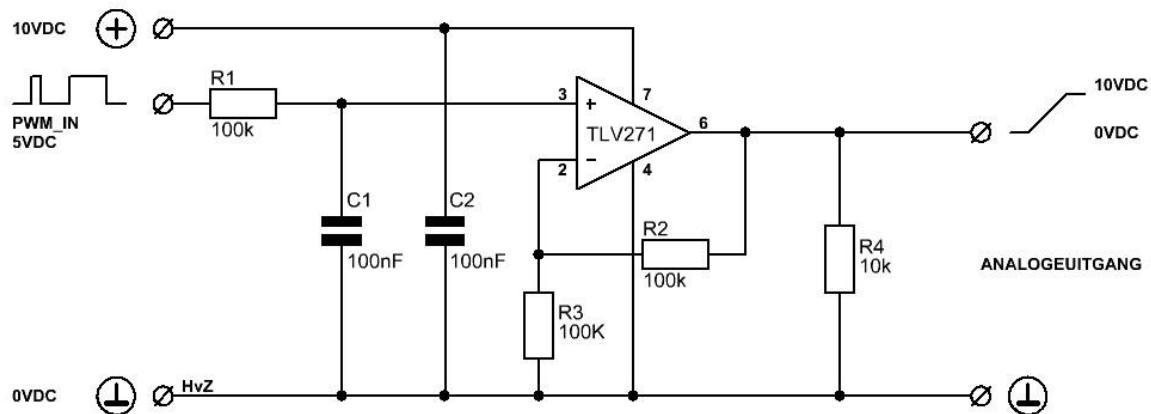
Hieronder een foto van de servo.



Hieronder zie je een PVC kogelkraan die vaak wordt toegepast in combinatie met de servo.



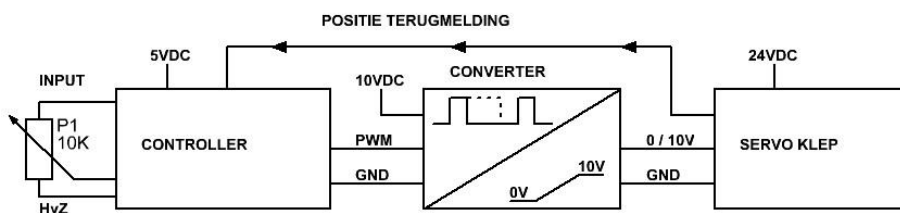
Ik ga een voorbeeldschema geven hoe je zo'n servo bijvoorbeeld aan kunt sturen.



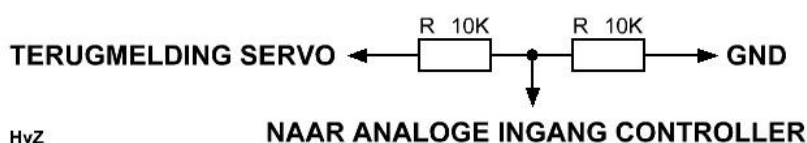
Op deze manier kan je van een PWM signaal een spanning maken (in dit geval 0 tot 10VDC), die spanning is nodig om de servo te besturen. In het schema zie je dat ik een TLV271 gebruikt heb, dit mag ook een ander type operationele (opamp) versterker zijn. Als het maar wel een common rail type opamp is. Anders wordt de 0 tot 10VDC spanning niet gehaald, en dat is voor een goede werking van de servo wel nodig.

Met bijvoorbeeld een microcontroller of met een NE555 kan je een PWM signaal maken (of met een andere schakeling die PWM opwekt) en daar zou je de servo mee kunnen bedienen. Op de servo zitten vier aansluitingen, twee voor de voeding, één draad waar het analoge signaal opkomt, en één draad waar een analogo signaal uitkomt. Dat is de terugmelding; op die manier kan je zien waar de klep staat.

Hieronder een blokschema.



Op deze manier wordt de servo bediend. Het inputsignaal kan natuurlijk van alles zijn, maar voor het voorbeeld wordt de servo via een potmeter bediend. Let er wel op dat het terugkoppelsignaal ook tot 10VDC oploopt, als je dat signaal in wil lezen op een analoge ingang van de controller, dan moet daar een spanningsdeler tussen gezet worden. Dat kunnen bijvoorbeeld twee weerstanden zijn van 10K. Zie voorbeeld hieronder.



HvZ

Voor de besturing heb ik een combinatie gemaakt met Basic en Profilab.

Hieronder het Basic programma.

```
Device 16F887 ; Processor type

Xtal 10 ; Cristal 10Mhz

Asm ; Config settings
CONFIG_REQ
__CONFIG __CONFIG1, HS_OSC & WDT_OFF & DEBUG_OFF & FCMEN_OFF & LVP_OFF &
IESO_OFF & BOR_OFF & CPD_OFF & CP_OFF & MCLR_OFF & PWRTE_ON
__CONFIG __CONFIG2, WRT_OFF & BOR40V
EndAsm

All_Digital true ; Alle poorten digitaal

Declare Adin_Res = 8 ; resolutie 8 bits
Declare Adin_Tad = frc ; set RC osc
Declare Adin_Stime = 50 ; sample tijd 5

Declare CCP1_Pin PORTC.2 ; PWM signaal uit

Declare Serial_Baud 9600 ; Baudrate 9600

Declare Rsin_Pin PORTB.0 ; Data in
Declare Rsout_Pin PORTB.1 ; Data uit

Symbol FREQUENTIE = 1000 ; Frequentie is 1KHz

Dim DATA_IN As Byte ; Variabele data in
Dim POSITIE_SERVO As Byte ; Variabele data

Clear ; Wis geheugen

;543210 ; Hulpregel poort A
PORTA = %000000 ; Maak poort A laag
TRISA = %111111 ; Poort_A I/O

;543210 ; Hulpregel poort B
PORTB = %000000 ; Maak poort B laag
TRISB = %000001 ; Poort_B I/O

;76543210 ; Hulpregel poort C
PORTC = %00000000 ; Maak poort C laag
TRISC = %00000000 ; Poort_C I/O

;76543210 ; Hulpregel poort D
PORTD = %00000000 ; Maak poort D laag
TRISD = %00000000 ; Poort_D I/O

;210 ; Hulpregel poort E
PORTE = %000 ; Maak poort E laag
TRISE = %111 ; Poort_E I/O

;76543210 ; Hulpregel analoog
ADCON0 = %00000001 ; ADCON0 register analoog

;543210 ; Hulpregel analoog poort_B
ANSELH = %000000 ; ANSEL register analoog poort_B
```

```

;-----
; PROGRAMMA SERIËLE BESTURING SERVO KLEP.
;-----

```

RUN:

```

    POSITIE_SERVO = ADIn 0

```

```

    RSOut POSITIE_SERVO

```

```

    DATA_IN = RSIn

```

```

    DelayMS 20

```

```

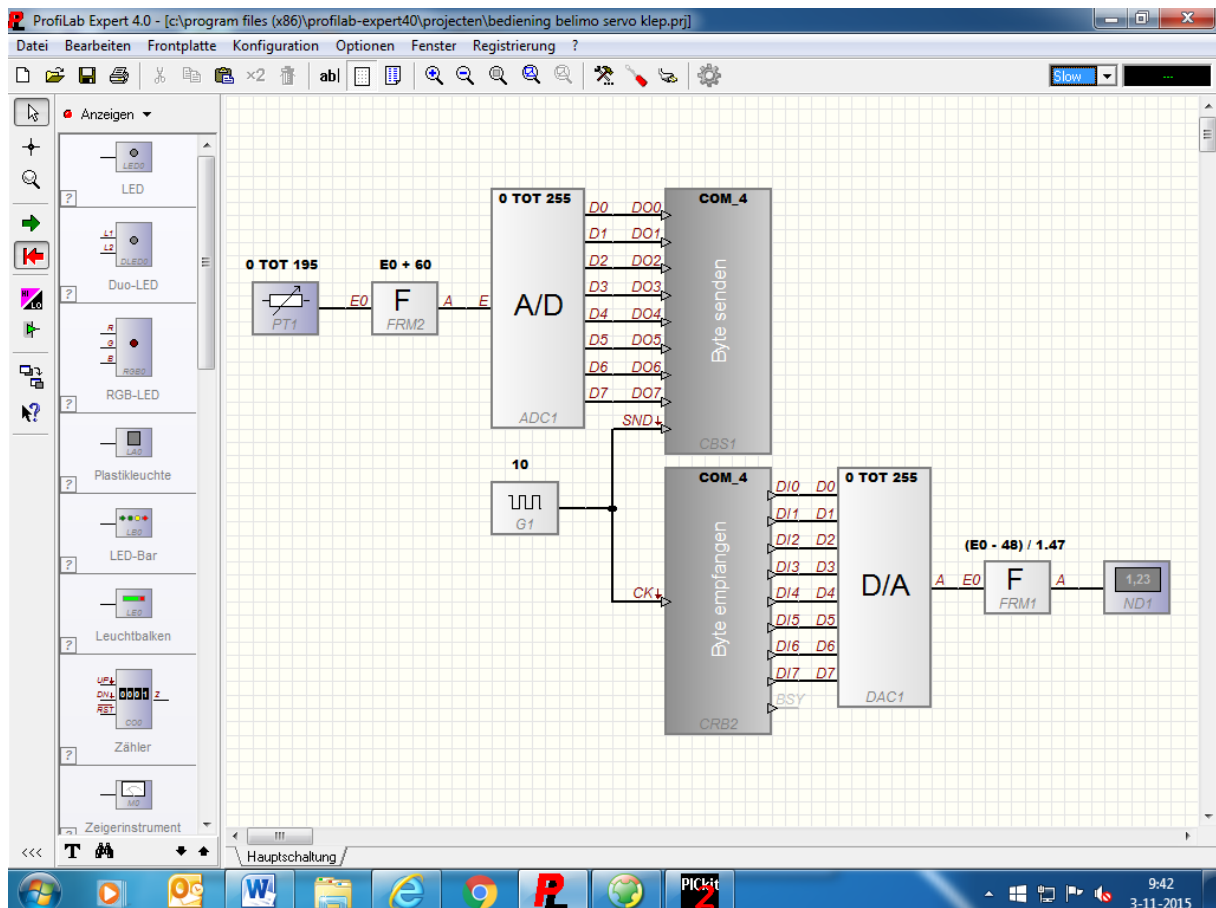
    HPWM 1, DATA_IN, FREQUENTIE

```

GoTo RUN

End

Zoals je kan zien stelt het programma niets voor. Het enige wat er gebeurt is, dat er een paar signalen serieel ontvangen en verstuurd worden. Het signaal dat op de analoge ingang (ADIN 0) komt, is het signaal dat afkomstig is van de servo en geeft de positie weer. Dit signaal wordt serieel verstuurd door (RSOUT) naar de PC en daar wordt in Profilab de waarde van de positie op het beeldscherm gezet. Dan hebben we nog (DATA\_IN) dit signaal is afkomstig van de potmeter in Profilab, dat signaal wordt serieel verstuurd en komt binnen via (RSIN). De waarde (DATA\_IN) wordt omgezet naar een PWM signaal met een vaste frequentie van 1KHz. Het PWM signaal wordt door de converter weer omgezet naar een analogoog signaal tussen de 0 en 10VDC. Hieronder het programma in Profilab.



Zoals je kan zien op het voorbeeld staan alle waardes erbij. Er is één potmeter gebruikt, tweemaal een formule module, eenmaal een puls geveer, eenmaal een byte versturen module, eenmaal een byte ontvangen module, eenmaal een A/D module, eenmaal een D/A module en eenmaal een display module. De data in het voorbeeld wordt serieel door de USB poort 4 verzonden, maar dat kan je zelf aanpassen. Hieronder de frontplaat voor de bediening van de servo klep.



Hier zie je de potmeter voor het bedienen van de servo en het display voor de positie van de servo. De waarde die getoond wordt is het percentage tussen de 0 en 100%, op het voorbeeld staat de servo klep dus voor 30% open.

Tot zover het artikel over servo's. Ik hoop dat het een beetje duidelijkheid schept over de werking van servo's.

Voor het versturen en ontvangen van seriële signalen tussen PC en controller heb je een seriële interface nodig, deze wordt in het boek werken met Profilab Expert 4.0 beschreven op bladzijde 52.

Hier kan je het boek gratis downloaden. <http://www.circuitsonline.net/artikelen/view/46>

**H. van Zwieten.**