

Het bouwen van een robotarm.

Het bouwen:

Ik liep al een tijd met de gedachte rond om een keer een robotarm te gaan maken, maar het kwam er maar niet van. Heb er toch de tijd voor gevonden en ik moet zeggen dat het een leuk project geworden is. Het heeft bij elkaar wel +/- een jaar geduurd, maar heb er niet constant aan gewerkt. Ik ben met wat ruwe schetsen begonnen en heb wat berekeningen gemaakt voor de motoren, dat wilde ik wel eerst voor elkaar hebben zodat ik later niet voor verrassingen zou komen te staan.

Voor de motoren heb ik stappenmotoren genomen met een vertraging van 20:1, deze motoren leveren een koppel van 20Nm. Je ziet vaak dat er voor modelbouw servo's gekozen wordt, maar die waren voor mij sowieso te klein. En ik vind ze veel te soppig, vooral als ze wat zwaarder belast gaan worden.

Het is de bedoeling dat het een vijfassige arm gaat worden, met een basis, bovenarm, onderarm, pols en een hand.

Als materiaal heb ik voor verschillende materialen gekozen, de basis en bovenarm worden gemaakt van MDF. De onderarm wordt een combinatie van MDF en PVC. Het draaigedeelte voor de basis wordt van staal en wordt met kogellagers gelagerd. De as voor de pols wordt van RVS gemaakt en dan nog wat aluminium onderdelen voor de bediening van de boven en onderarm.

Ik wilde ook dat de arm op verschillende manieren bediend kon worden, ik heb gekozen voor een bediening via een IR afstandsbediening en een bediening via de PC. Via de PC wordt er weer gebruik gemaakt van Profilab Expert 4.0 en voor de IR afstandsbediening gebruik ik een Marmitek CableLink 75. De afstandsbediening werkt met RC5 code en die is weer goed uit te lezen via de Microchip controllers die ik gebruik.

Voor de besturing ga ik gebruik maken van twee verschillende controllers de 16F873 en de 18F4520. De 16F873 wordt als interface ingezet voor de communicatie tussen de afstandsbediening en de bediening via de PC. Ik heb hier voor gekozen omdat het uitlezen van de afstandsbediening nogal wat instructietijd vraagt en dat ga je terugzien in de snelheid van de stappenmotoren. Het uitsturen van de signalen van de PC gebeurt via de USB poort met een USB naar serieel converter. Dit gebruik ik vaker als ik signalen van de PC naar een controller stuur en van de controller naar de PC. Dit werkt prima en zonder problemen.

De 18F4520 regelt het aansturen van de stappenmotoren en de DC motor voor de hand. In deze controller staat het nullen van de assen, de handbediening van de arm en het auto programma voor het automatisch aflopen van de geprogrammeerde bewegingen van de arm.

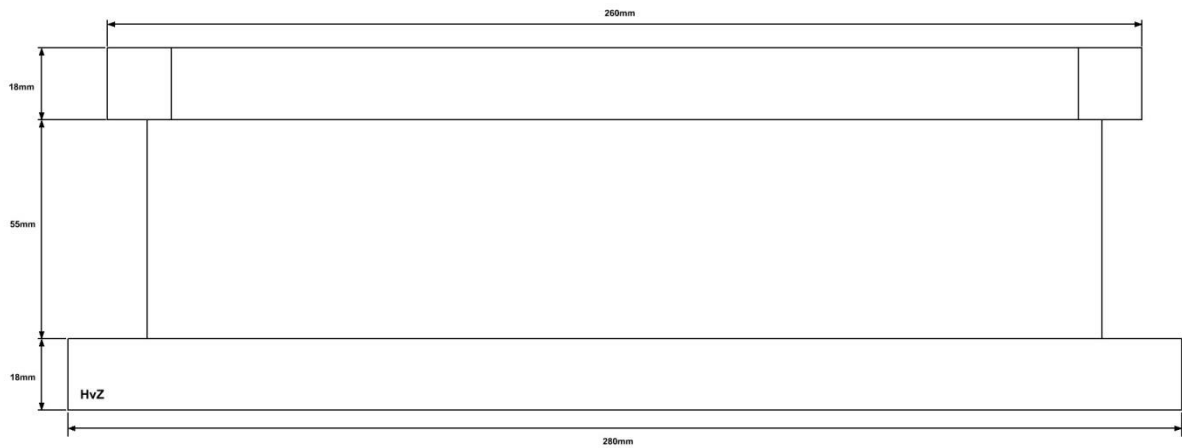
De communicatie tussen beide controllers gebeurt parallel, maar dat wordt later wel duidelijk als de software aan de orde komt.

Voor de bediening van de hand ga ik een L293D gebruiken in combinatie met een 4011. Voor de stappenmotoren ga ik vier M325 micro step drivers gebruiken, deze zijn zwaar genoeg voor mijn toepassing.

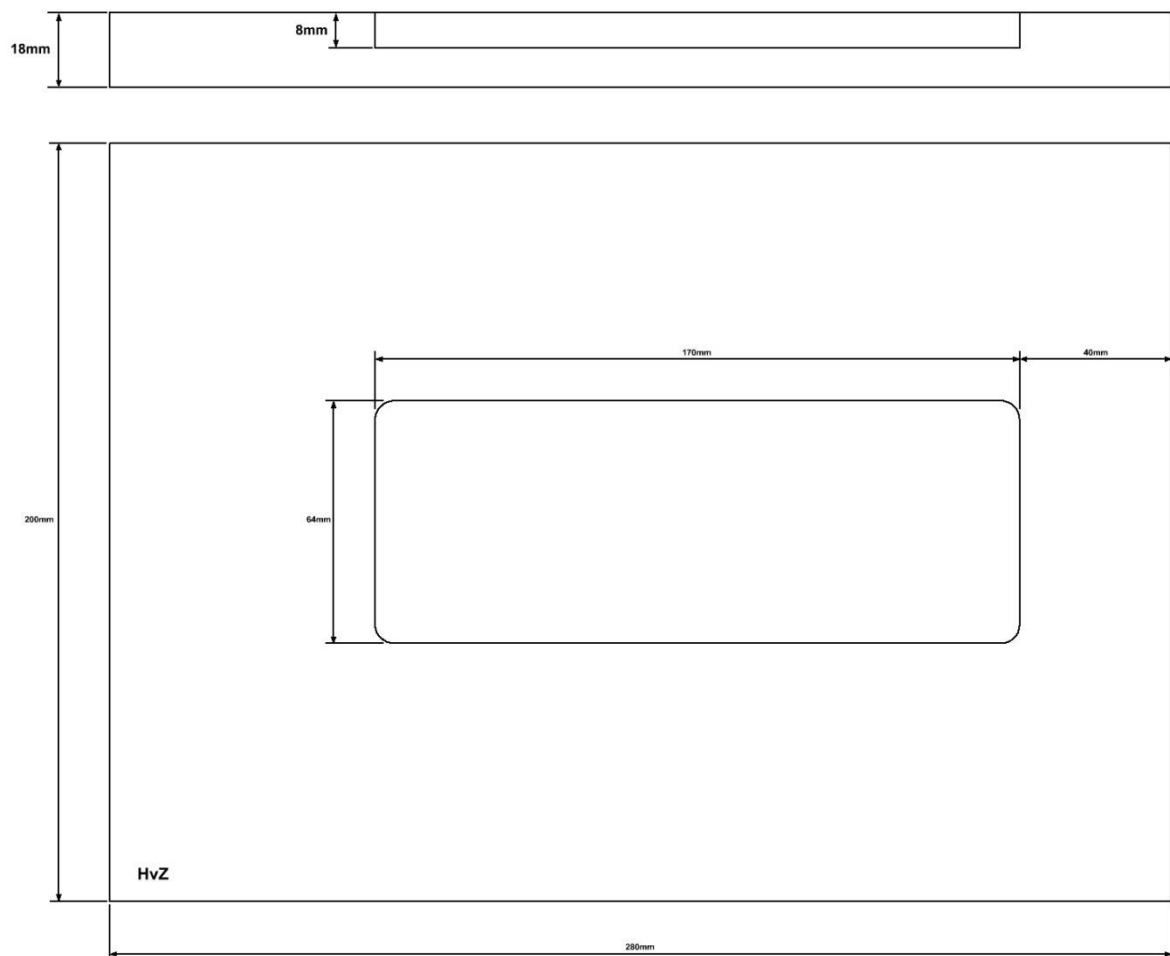
Als voeding voor het geheel ga ik een geschakelde voeding gebruiken van 12VDC bij 6A, dat is meer dan voldoende voor de motoren en de controllers.

Nu we het één en ander uitgezocht hebben kunnen we de onderdelen gaan maken.

Hieronder zie je de samengestelde basis met de bovenplaat, de voetplaat en het middenstuk. Alles is van MDF 18mm gemaakt, dit heb ik gewoon op maat laten zagen, hier hoefde niet veel aan te gebeuren verder.

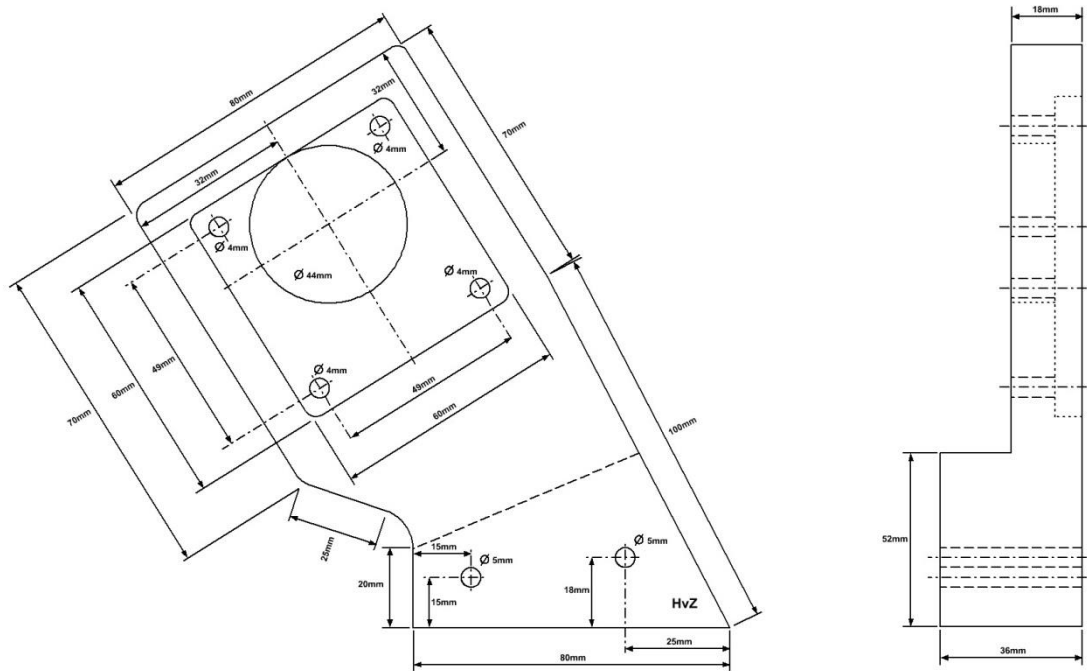


Hieronder de voetplaat.

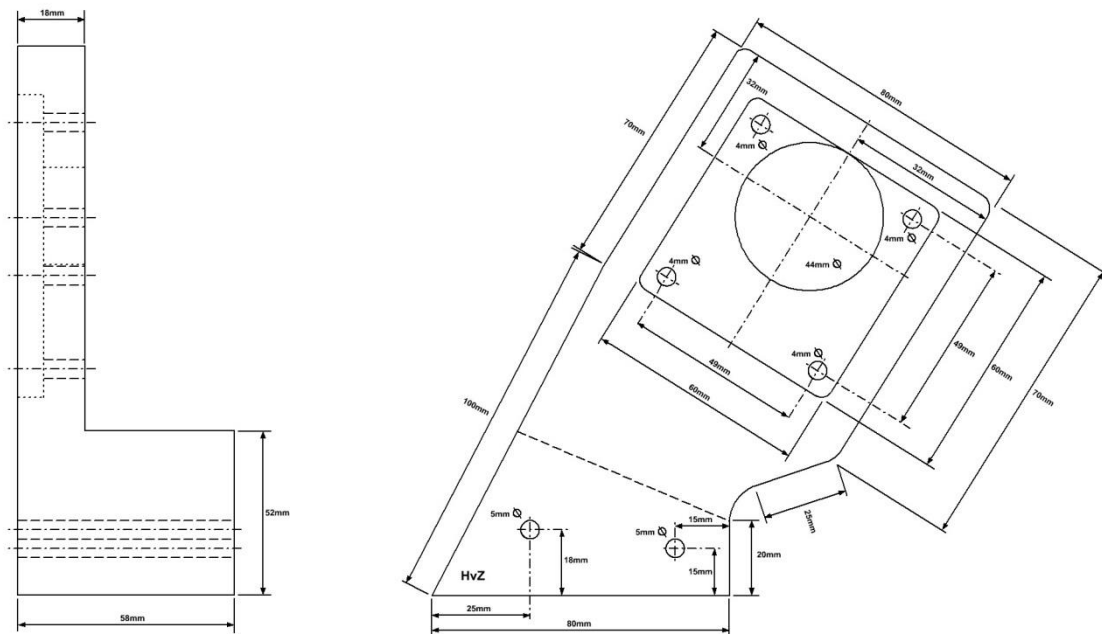


Alleen in de voetplaat moest een uitsparing gefreesd worden voor de aandrijving. Hier komt later de tandriem overbrenging in voor de basis.

Hieronder de linker motordrager, daar komt de motor in voor de onderarm.

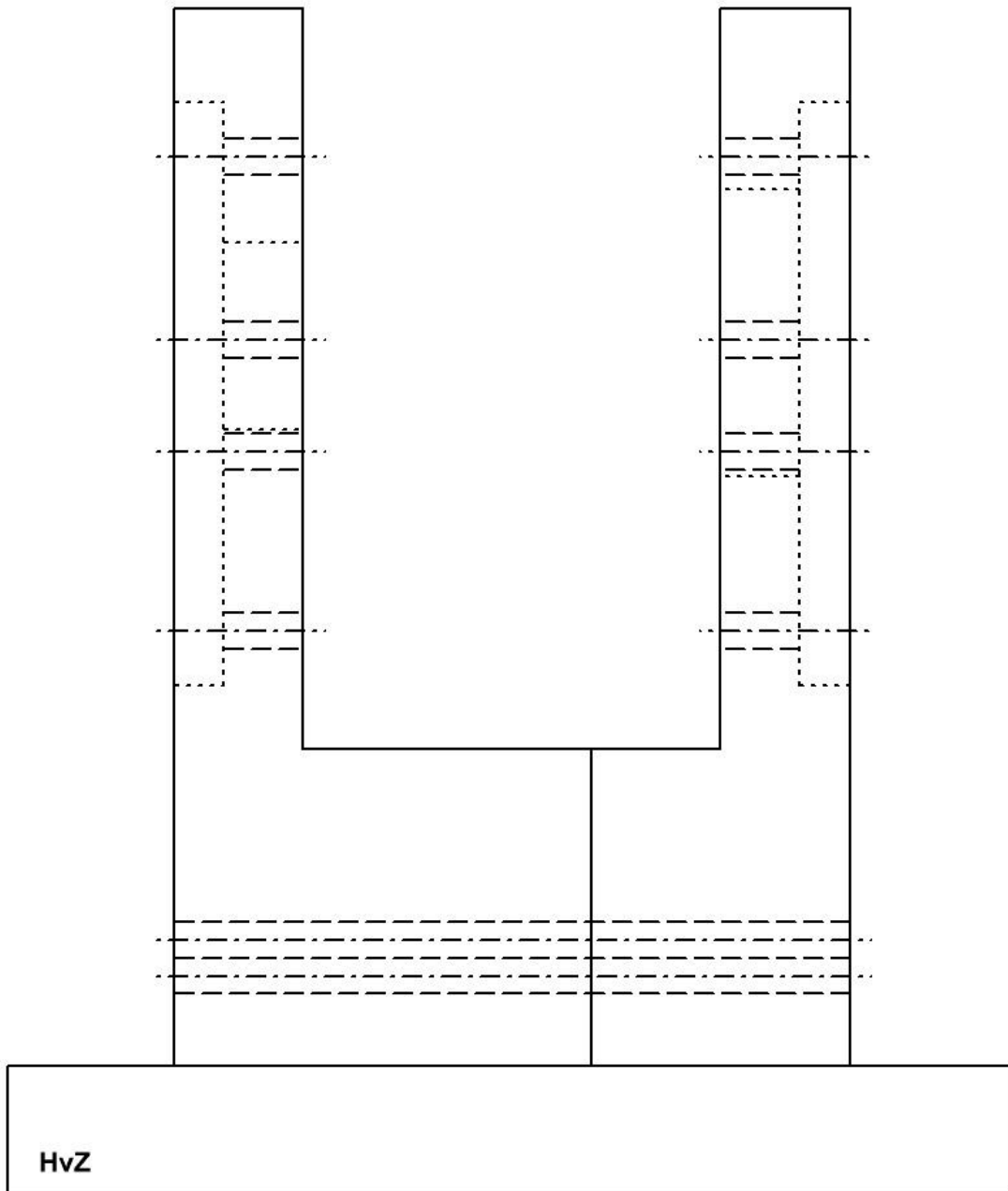


Hieronder de rechter motordrager, daar komt de motor in voor de bovenarm.



De dragers worden op de draaischijf gemonteerd en aangedreven door de basis motor.

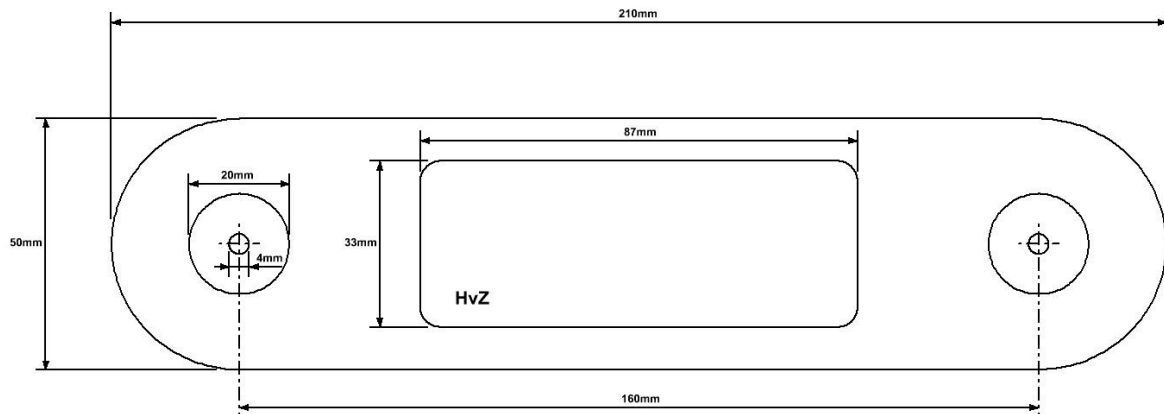
Hieronder de samenstelling “vooraanzicht” van de rechter en linker drager.



De dragers worden door middel van twee 5mm draadeinden aan elkaar gekoppeld. Het geheel is demonteerbaar en dat is nodig voor de montage van de motoren, bovenarm en onderarm. De rechter drager is aan de draaischijf gelijkmd. De dragers zijn van 18mm MDF gemaakt. De dragers zijn in zijn geheel gefreesd en in de zijkanten is een uitsparing gefreesd voor de aandrijving, hier passen de motoren precies in. De motoren worden met vier 4mm inbusbouten op de dragers gemonteerd. De dragers en draaischijf komen later op een stalen draaiplateau wat weer gelagerd is in de basis.

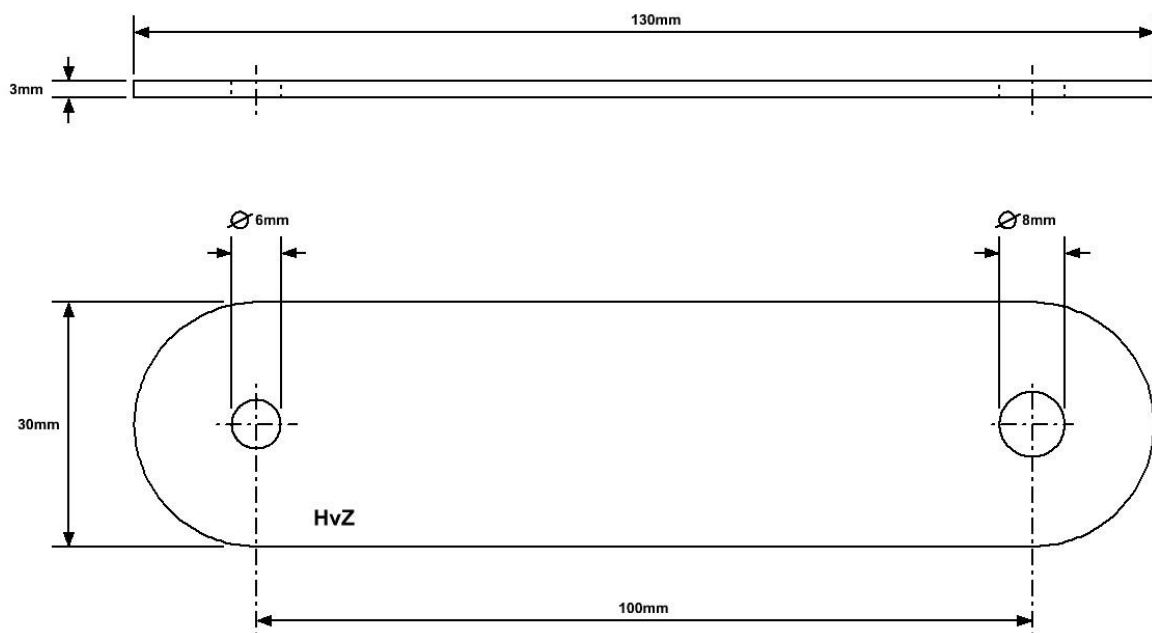
Hieronder zie je de bovenarm, deze is samen gesteld uit drie stukken MDF van 18mm. Eerst zijn de twee buitenkanten uitgefreesd, "de stukken met de rondingen eraan"

Het middenstuk is een rechthoek die later tussen de twee delen met de rondingen is gelijmd.



In de twee buitenste delen is nog een uitsparing gefreesd voor een sticker, hier komt later de tekst "ROBOTIC ARM" in te staan.

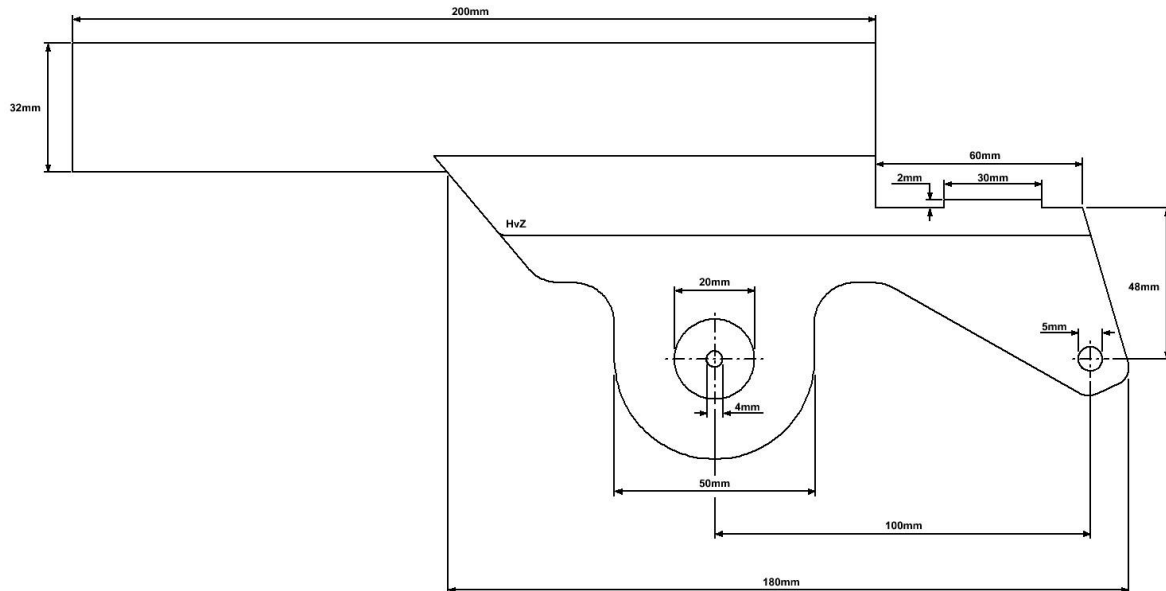
Hieronder de hevel die de onderarm gaat bedienen.



De hevel is gemaakt van aluminium en gaat de onderarm bedienen doormiddel van twee stangen, de stangen komen tussen de hevel en de onderarm te zitten.

Hieronder de onderarm.

De onderarm is samengesteld uit MDF en PVC, het geheel is verlijmt met een twee componentenlijm van Bison.



Aan de voorkant van de onderarm komt een kogellager te zitten voor de as van de pols. Daar komt later de hand aan te zitten. Aan de achterkant komt de as met een koppeling aan de aandrijving vast te zitten.

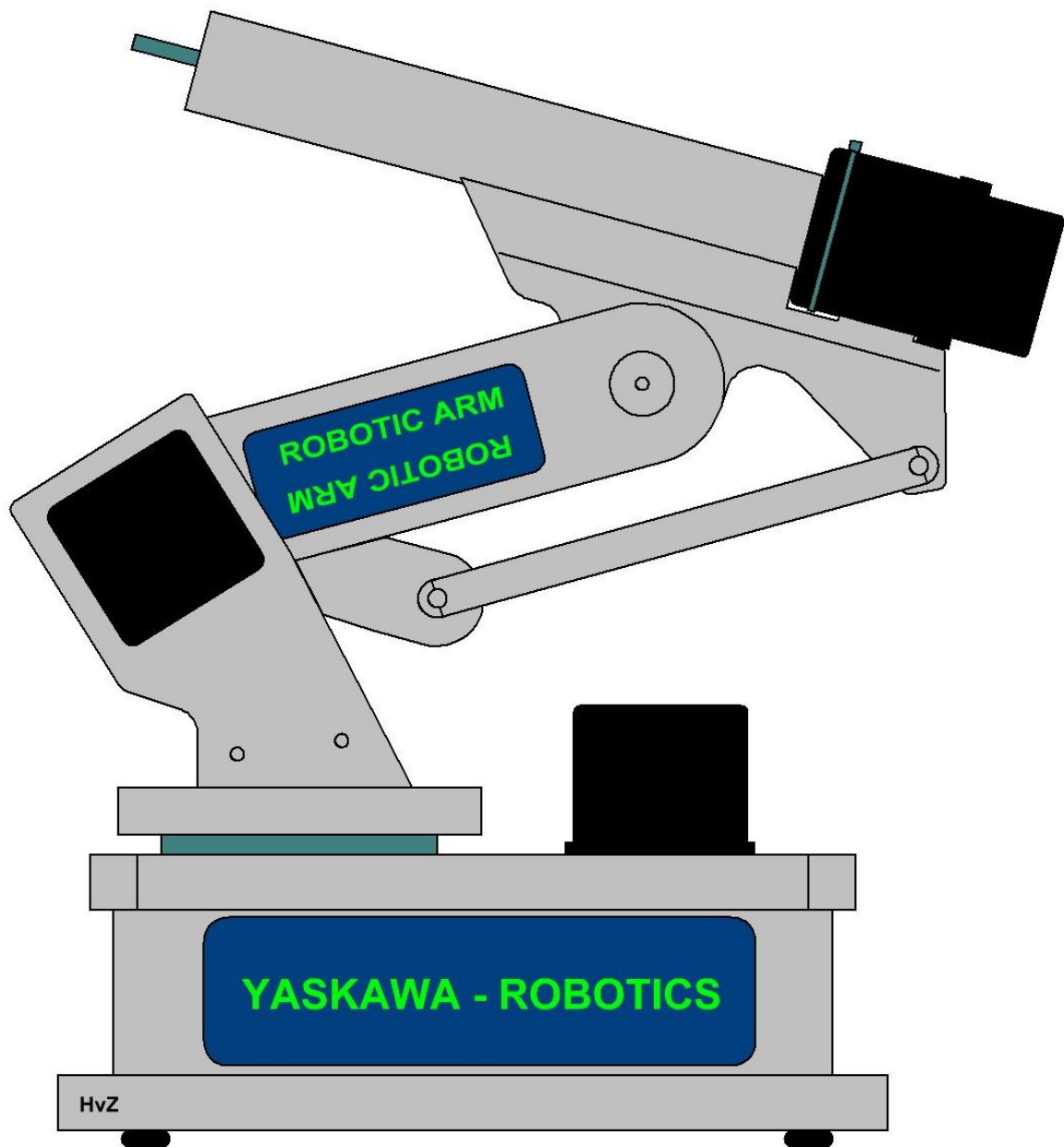
De motor is een kleiner type stappenmotor met een vertraging van 115.9:1. Ik had deze vertraging nog liggen en heb hem helemaal aangepast zodat het geheel gebruikt kon worden met de kleine stappenmotor. In het MDF gedeelte is een aluminium prop gemaakt van rond 20mm met een gat van 4mm. De aluminium prop is weer met twee componentenlijm in het MDF gedeelte gelijmd. Deze proppen zitten ook aan één kant in het ronde gedeelte van de bovenarm. Dit is het draaipunt tussen de boven en onderarm, het geheel wordt verbonden door een stalen as. Tussen de draaipunten van de onder en bovenarm zitten kunststof slijtplaten van 1.5mm dik.

In het 5mm gat komt een bout die verbonden is met de stangen die naar de hevel gaan. De stangen zijn van aluminium met daaraan gemonteerde kogelgewrichten, zo kan het geheel spelingsvrij draaien en toch een beetje bewegen. Al is dat niet echt nodig.

De kracht wordt door de stangen overgebracht van de hevel naar de onderarm. Ook kunnen de boven en onderarm apart van elkaar bewegen, de bovenarm kan bijvoorbeeld omhoog bewegen en de onderarm omlaag. Of je beweegt beide armen omhoog of omlaag.

Ale assen kunnen apart of in combinatie bewogen worden en dat allemaal tegelijk. Het is wel zo hoe meer assen er bewegen hoe trager het proces gaat, misschien dat er in de software nog iets veranderd of aangepast kan worden om dit tegen te gaan.

Hieronder de gehele samenstelling van de robotarm.



Hier is de gehele constructie goed te zien. In de behuizing van de basis zit de tandriem overbrenging van 1.5:1 plus de vertraging van 20:1 van de stappenmotor. Ik heb voor de tandriem gekozen zodat de stappenmotor boven op de basis kwam te zitten, dit is gedaan om de basis wat lager te kunnen houden en om wat tegengewicht te creëren.

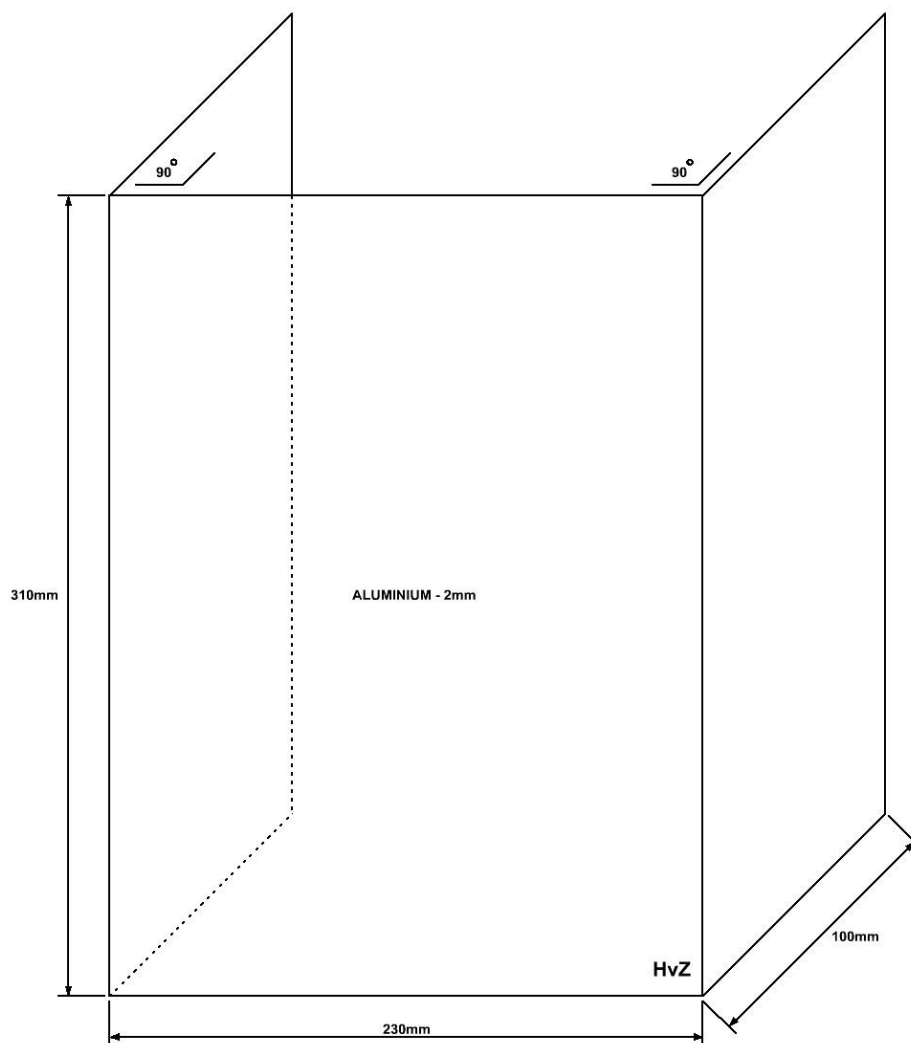
In het draaigedeelte van de basis zijn twee kogellagers opgenomen om het geheel soepel en spelingsvrij te laten draaien. De uitsparingen voor de lagers zijn uit gefreesd in het MDF en verlijmt met twee componentenlijm.

Het freeswerk van het geheel is op mijn omgebouwde CNC freesbank gedaan, het tekenwerk is gedaan met Frontdesigner 3.0. Hier kun je HPGL mee genereren voor de CNC freesbank. Het tekenwerk en freeswerk heeft toch behoorlijk wat tijd in beslag genomen, maar ik ben wel blij dat ik doorgezet heb. Maar ik heb wel eens gedacht, waar ben ik aan begonnen. Het heeft dan ook zeker meer dan een jaar stil gelegen voordat ik er weer tijd voor had en niet te vergeten zin in had. Maar het resultaat mag er zijn al zeg ik het zelf.

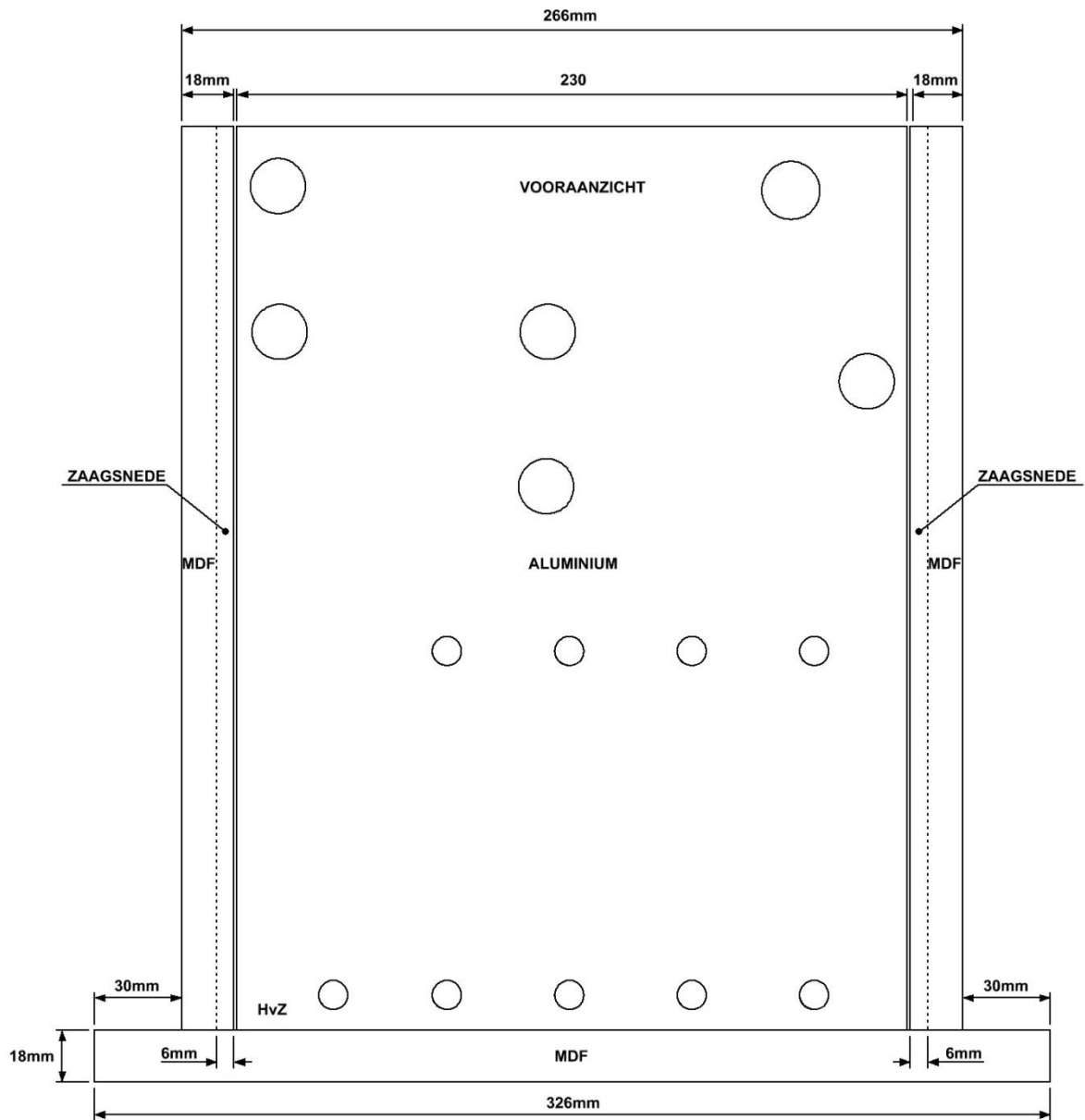
Behuizing voor de besturing:

Het was wel even uitzoeken om een goede behuizing te verzinnen, er zijn natuurlijk behuizingen te kust en te keur, maar ik wilde iets hebben zodat de besturing te zien is. Dit met het oog op eventuele demonstraties op scholen, ik vind dat er veel te weinig aandacht besteed wordt aan techniek en de mogelijkheden er van. Er zou dus meer aandacht aan besteed moeten worden in bijvoorbeeld groep 7 en groep 8, maar dat terzijde.

Hieronder het aluminium gezette U profiel, wat de basis is voor de behuizing. Hier worden de componenten op gemonteerd, zoals de controllers, de voeding, de microstepdrivers en de motorregelaar voor het bedienen van de hand.



Hieronder het vooraanzicht van de samenstelling van de behuizing.

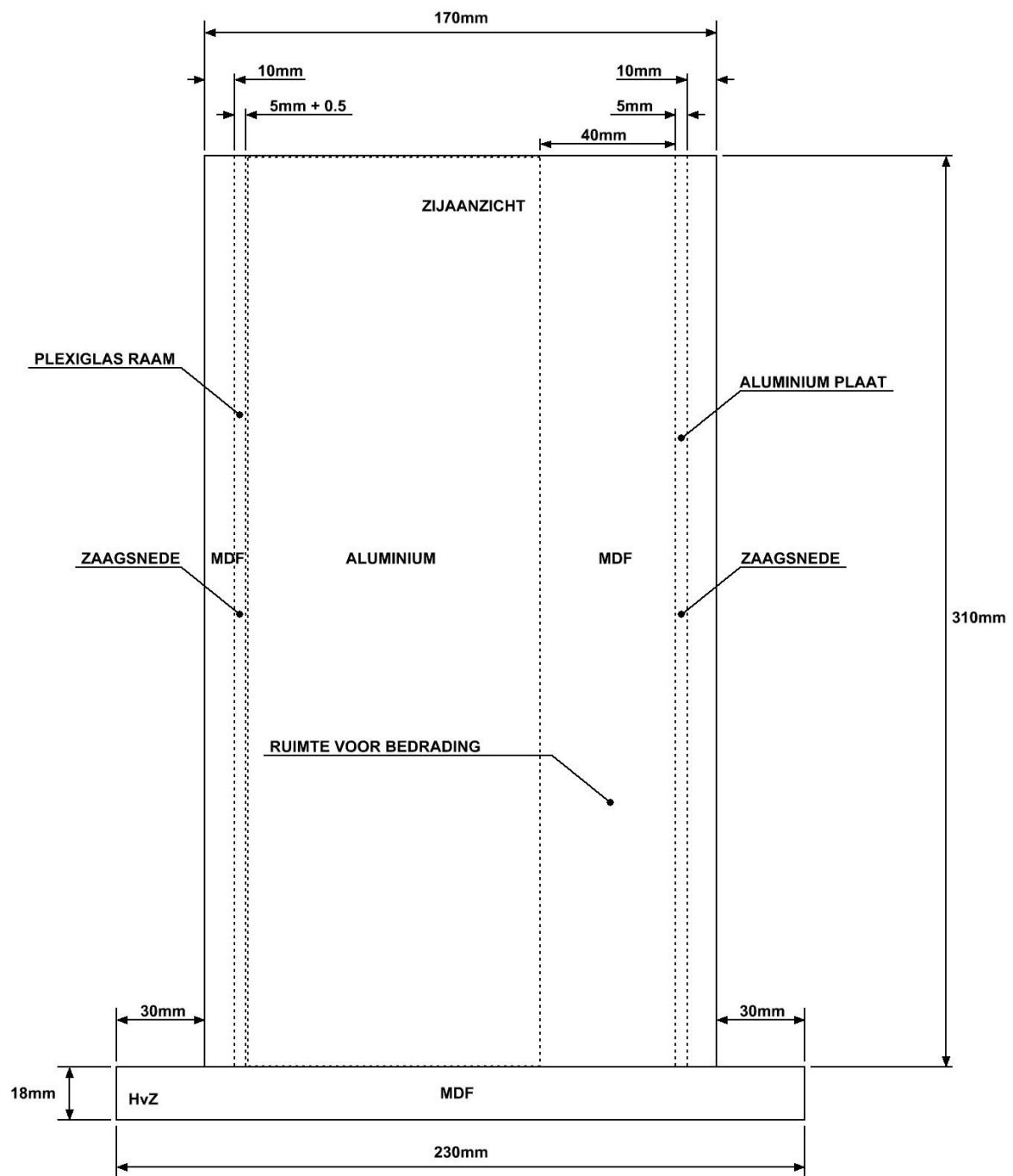


Het aluminium u profiel komt tussen twee MDF delen, deze delen worden op het aluminium gelijmd en geschroefd. In de MDF zijkanten komen twee zaagsneden te zitten, zowel aan de voorkant als aan de achterkant. Aan de voorkant komt er een plexiglas ruit in te zitten, zodat je de besturing kunt zien. Aan de achterkant komt er een paneel met de aansluitingen voor de motoren en de eindschakelaars. Aan de zijkant komt er nog een aansluiting voor de USB en een aansluiting voor de netspanning. De gaten in het aluminium profiel zijn voor het doorvoeren van de bedrading, aan de achterkant wordt er dan overgestoken naar een ander gat. Op deze manier ziet het er dan netjes uit, ik wilde eerst gebruik maken van bedradingskoker, maar dat nam iets te veel ruimte in beslag naar mijn zin. Na dat het MDF aan het aluminium profiel is bevestigd, worden de MDF zijkanten op de voetplaat gelijmd en geschroefd. Zo wordt het een stevig geheel.

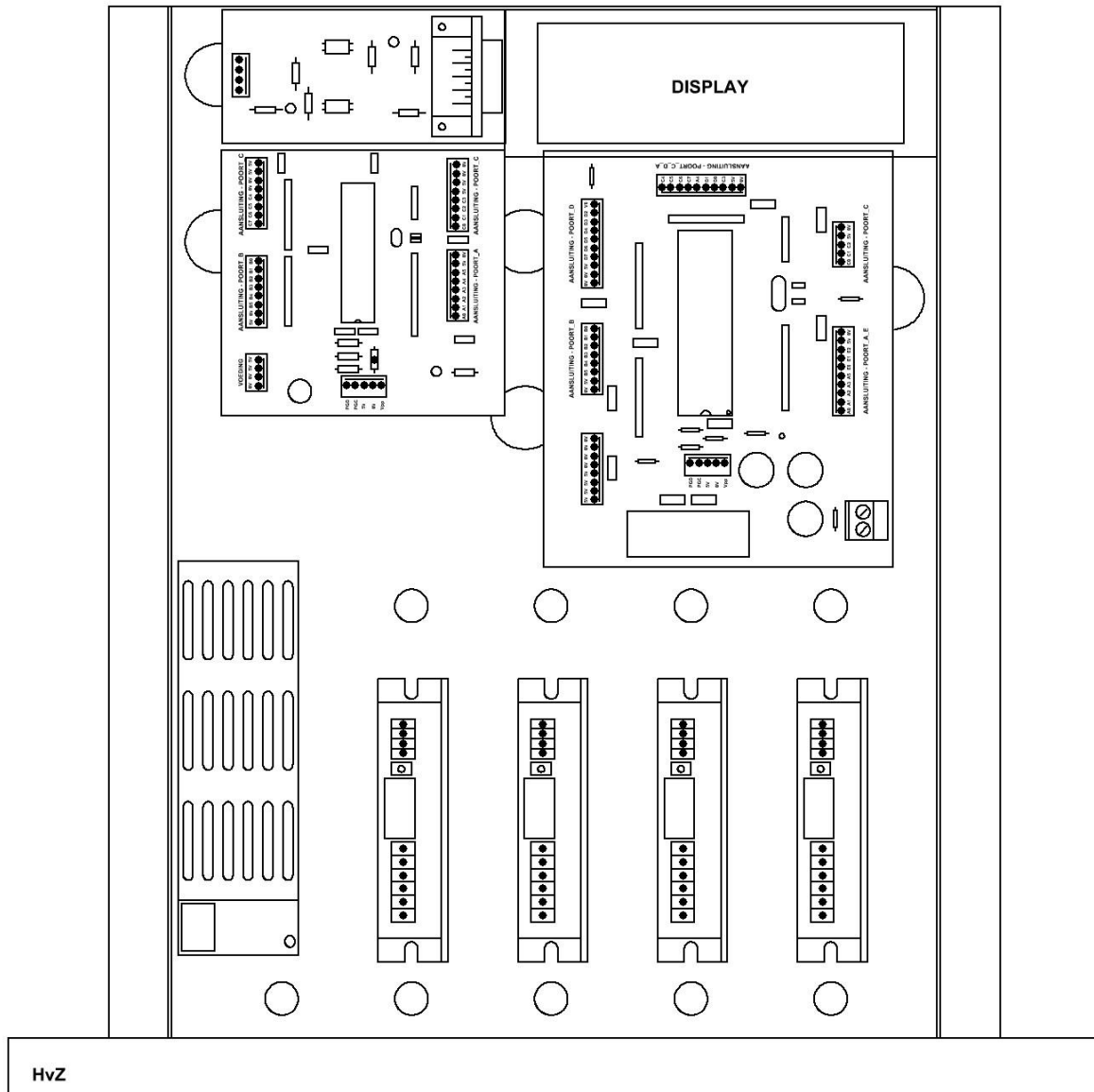
Hieronder het zijaanzicht van de behuizing.

Hier kun je goed zien wat ik hierboven beschreven heb. Zoals je kunt zien zit er een ruimte van 40mm tussen het aluminium profiel en de aluminium achter plaat. Dat is nodig voor de chassis delen van de stekkers, ik heb hiervoor vijf polige audio stekkers gebruikt. Voor de eindschakelaars heb ik een negen polige sub-D stekker gebruikt.

Aan de voorzijde komen de controllers, voeding en de microstepdrivers.

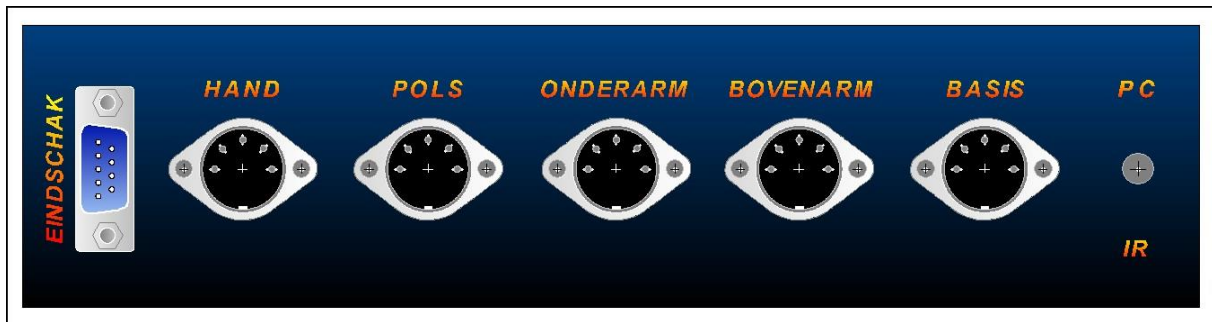


Hieronder het vooraanzicht met de componenten.



Linksboven zit de print voor de communicatie tussen de PC en de interfacecontroller, de interfacecontroller zit daar vlak onder. Op de interfacecontroller komen de commando's binnen van de PC en van de IR afstandsbediening. Ook de keuze schakelaar voor PC of IR zit daarop. Als IR sensor wordt een T-SOP 1738 gebruikt. Rechtsboven komt een display met daarop de tellers, de tellers geven de positie van de basis weer, de positie van de bovenarm, de positie van de onderarm en de positie van de pols. Bij het aanzetten van de besturing wordt er gevraagd om het homen van de assen te starten. Onder het display zit de hoofdcontroller, deze stuurt de microstepdrivers aan en het display. Ook komen hier de signalen op binnen van de eindschakelaars en de commando's van de interfacecontroller. Links onderaan zie je de voeding en rechts daarvan zie je de M325 microstepdrivers zitten.

Hieronder de sticker met de stekker aansluitingen.



Deze stekker aansluitingen komen helemaal onderin de behuizing te zitten, onder de aluminium achter plaat.

Helemaal rechts komt de keuzeschakelaar voor de keuze PC of IR.

De stickers maak ik altijd met Frontdesigner, "een programma van ABACOM" dat print ik dan uit op foto papier en lamineer het. Het geheel plak ik op de achterkant met dun dubbelzijdig tape of ik lijm het. De stickers voor de bovenarm en de basis zijn op dezelfde wijze gemaakt.

Bediening via IR RC5:

We gaan nu naar de besturing kijken, zoals ik al zei kunnen we de arm via de PC of IR bedienen. Via IR gaan we dat doen met een Marmitek RC5 afstandsbediening. Deze afstandsbediening had ik nog liggen en voldoet goed. Om te weten te komen wat de knopjes doen, moeten we eerst de afstandsbediening uitlezen. Dit heb ik via een microcontroller gedaan met een display, later zal ik de code daarvoor plaatsen.

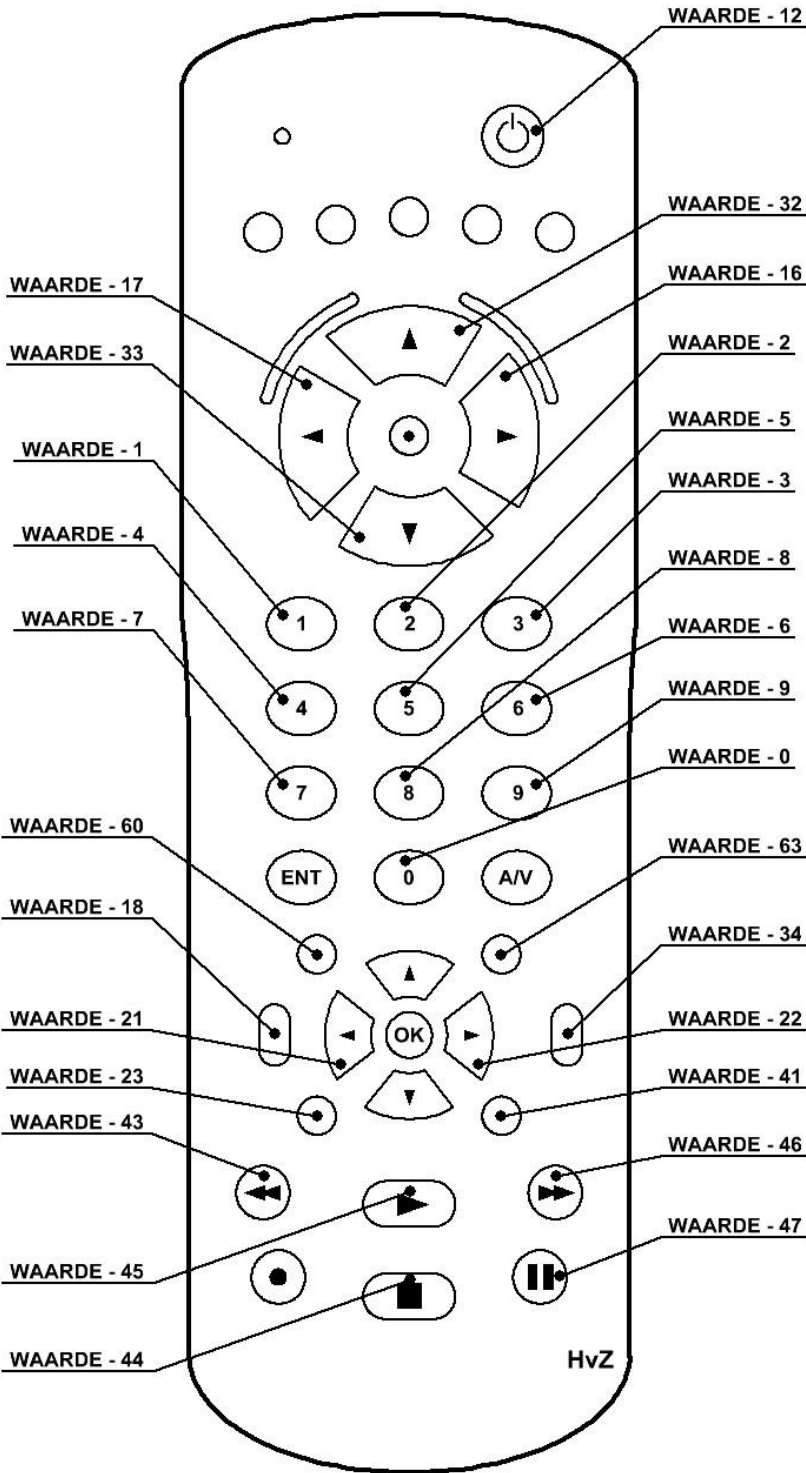
In totaal heb ik 28 knoppen tot mijn beschikking en dat is ruim voldoende. In totaal zitten er 34 knoppen op die je zou kunnen gebruiken, maar er zitten een aantal knoppen op die een dubbele waarde geven. Daar heb ik dus niets aan, die knoppen worden dan ook niet gebruikt.

Wat ik nodig heb is basis links en rechtsom, bovenarm op en neer, onderarm op en neer, pols links en rechtsom, hand open en dicht, keuze homen van de assen, keuze automatisch afwerken van het programma en een soort van een noodstop zodat alle assen stoppen als dat nodig is. Een echte noodstop is het niet, maar het is wel makkelijk is al gebleken.

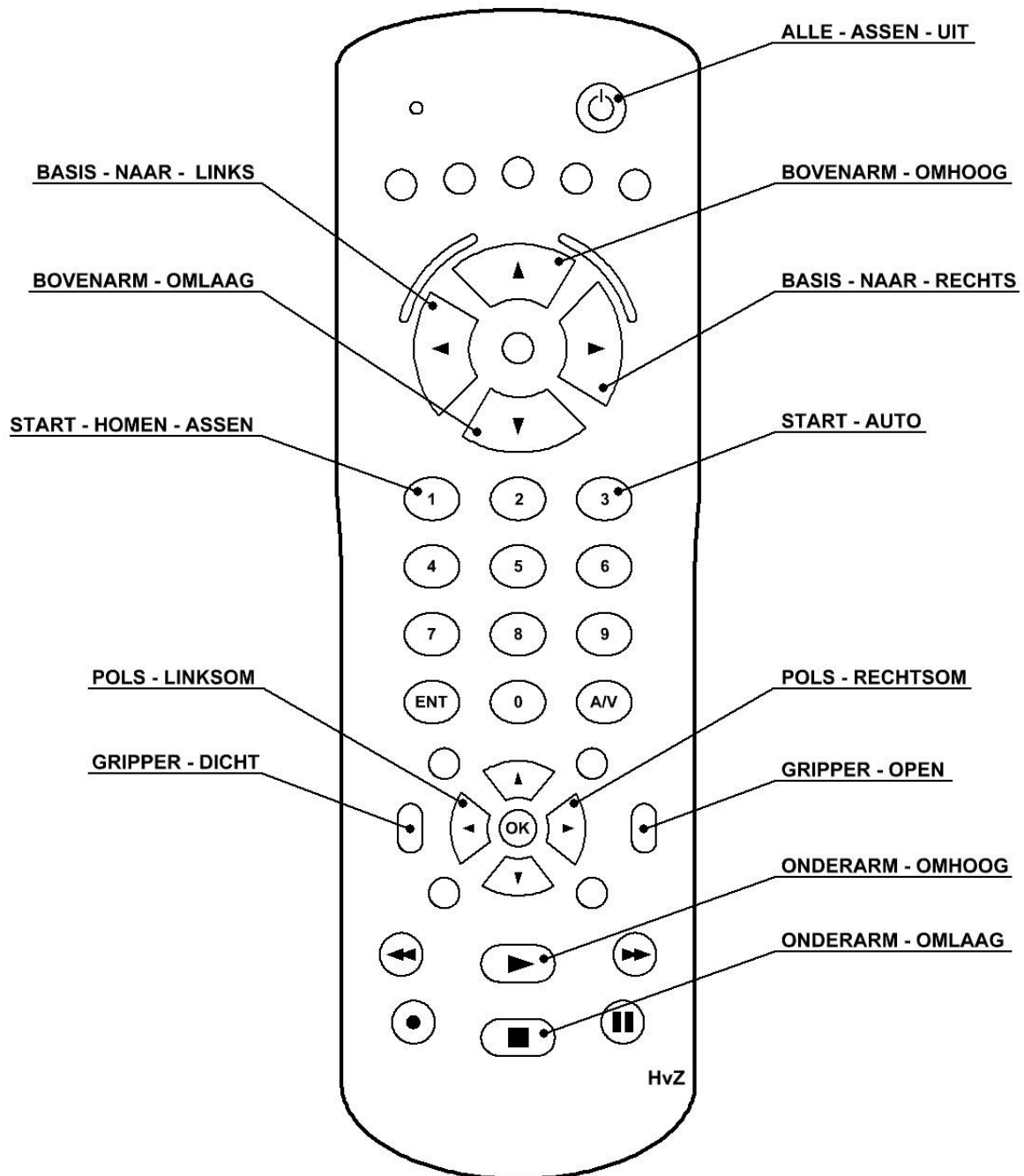
Ik heb dus in totaal 13 knoppen nodig van de 28 die ik tot mijn beschikking heb.

Maar wie weet komt er nog iets bij in de toekomst. Al zal dat niet mee gaan vallen, want de hoofdcontroller zit echt bom vol. Echt alle pinnen zijn gebruikt.

Hieronder de uitgelezen waardes die onder de knoppen zitten.



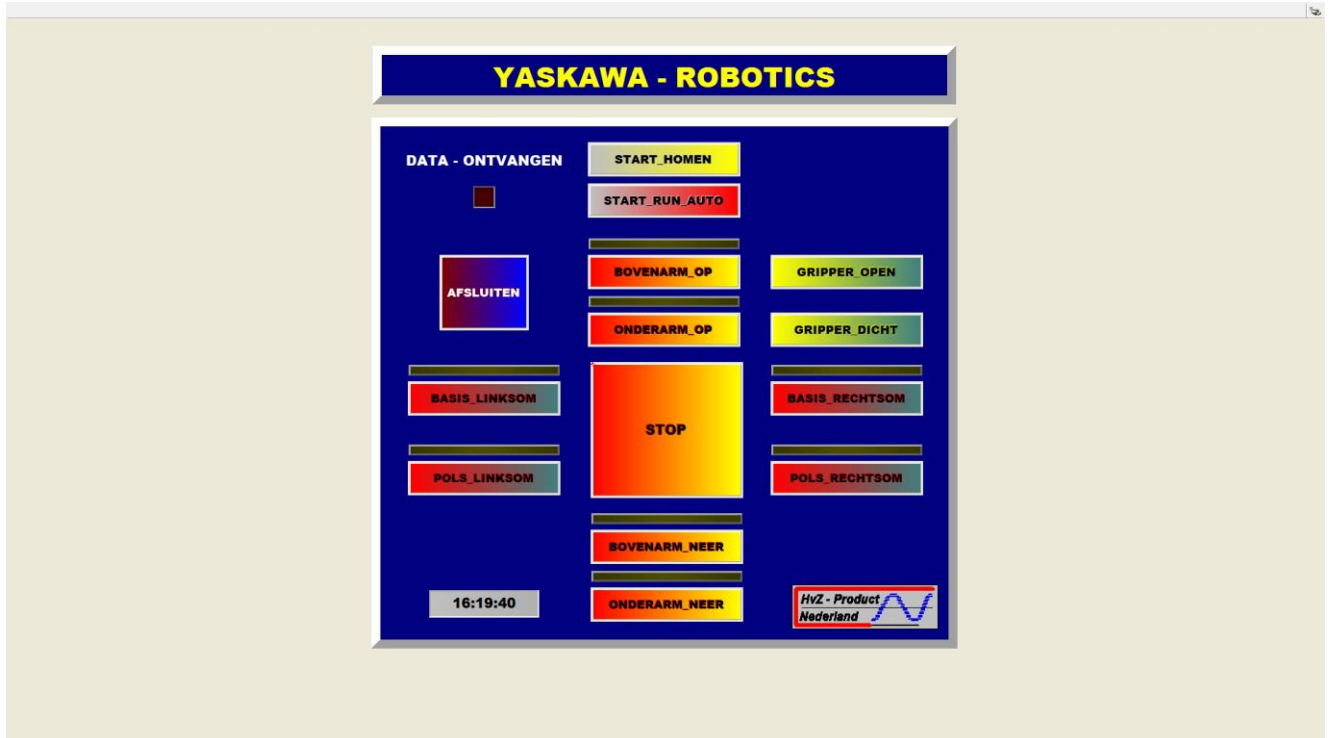
En zo heb ik de knoppen ingedeeld.



Ik heb geprobeerd om het zo logisch mogelijk in te delen, zodat de knoppen ook te herkennen zijn voor een bepaalde functie. Dat gaat natuurlijk niet op voor alle knoppen, maar voor sommige knoppen wel. Basis links rechts, bovenarm op neer, pols links rechts en de rest van de knoppen is zo logisch mogelijk gekozen. Tenminste voor mij dan. De aan uit knop zorgt ervoor dat alle assen stoppen. Knop 1 is home van de assen, knop 3 is start auto, knop menu is hand dicht, knop exit is hand open, knop afspelen is onderarm op en knop set is onderarm neer.

Bediening via de PC:

Voor de bediening via de PC heb ik een programma gemaakt in Profilab Expert 4.0. Hieronder de frontplaat.



De naam “YASKAWA – ROBOTICS” heb ik gekozen omdat dit een bestaand merk is en omdat ik daar vroeger wel eens aan gewerkt heb. Zoals je kunt zien staan alle functies op de frontplaat. Het lampje onder “DATA – ONTVANGEN” licht op zodra er een commando verstuurd is naar de interfacecontroller, de interfacecontroller geeft dus een signaal terug als de data ontvangen is. Het verzenden en ontvangen van het signaal gebeurt 300 maal per seconden, het kan nog sneller maar dat is niet nodig. Helemaal rechtsboven in zie je een USB stekker staan, als je daar op klikt kun je de desbetreffende USB poort instellen. Linksonder in loopt er een systeem klokje mee.

Voor de rest wijst het voor zich denk. De stopknop bedient hier alle assen weer, dus als er bijvoorbeeld vier assen draaien en er wordt op de stopknop gedrukt, dan stoppen alle assen. Dit is alleen voor de basis, bovenarm, onderarm en de pols. De hand wordt door een DC motor bediend op tijd. De hand gaat open en dicht in +/- 1 seconden, als je een commando aan de hand geeft duurt dat 1.2 seconden. Dus dat is tijd genoeg om de hand geheel open te sturen of dicht te sturen. Op de hand zit ook een beveiliging zodat de motor niet kan verbranden, als de hand snel achter elkaar open en dicht gestuurd wordt stop deze er even mee, dan spreekt de beveiliging dus aan. Maar bij normaal gebruik heb je daar geen last van.

Ik verzend hier alleen bepaalde waardes mee, eigenlijk het zelfde als met de IR afstandsbediening.

Voor home is dat waarde 5.

Voor start run auto is dat waarde 10.

Voor basis rechtsom is dat waarde 15.

Voor basis linksom is dat waarde 20.

Voor bovenarm omhoog is dat waarde 25.

Voor bovenarm omlaag is dat waarde 30.

Voor onderarm omhoog is dat waarde 35.

Voor onderarm omlaag is dat waarde 40.

Voor pols rechtsom is dat waarde 45.

Voor pols linksom is dat waarde 50.

Voor hand open is dat waarde 55.

Voor hand dicht is dat waarde 60.

Voor stop is dat waarde 200.

Voor de terugmelding of het signaal aangekomen is, is dat waarde 100.

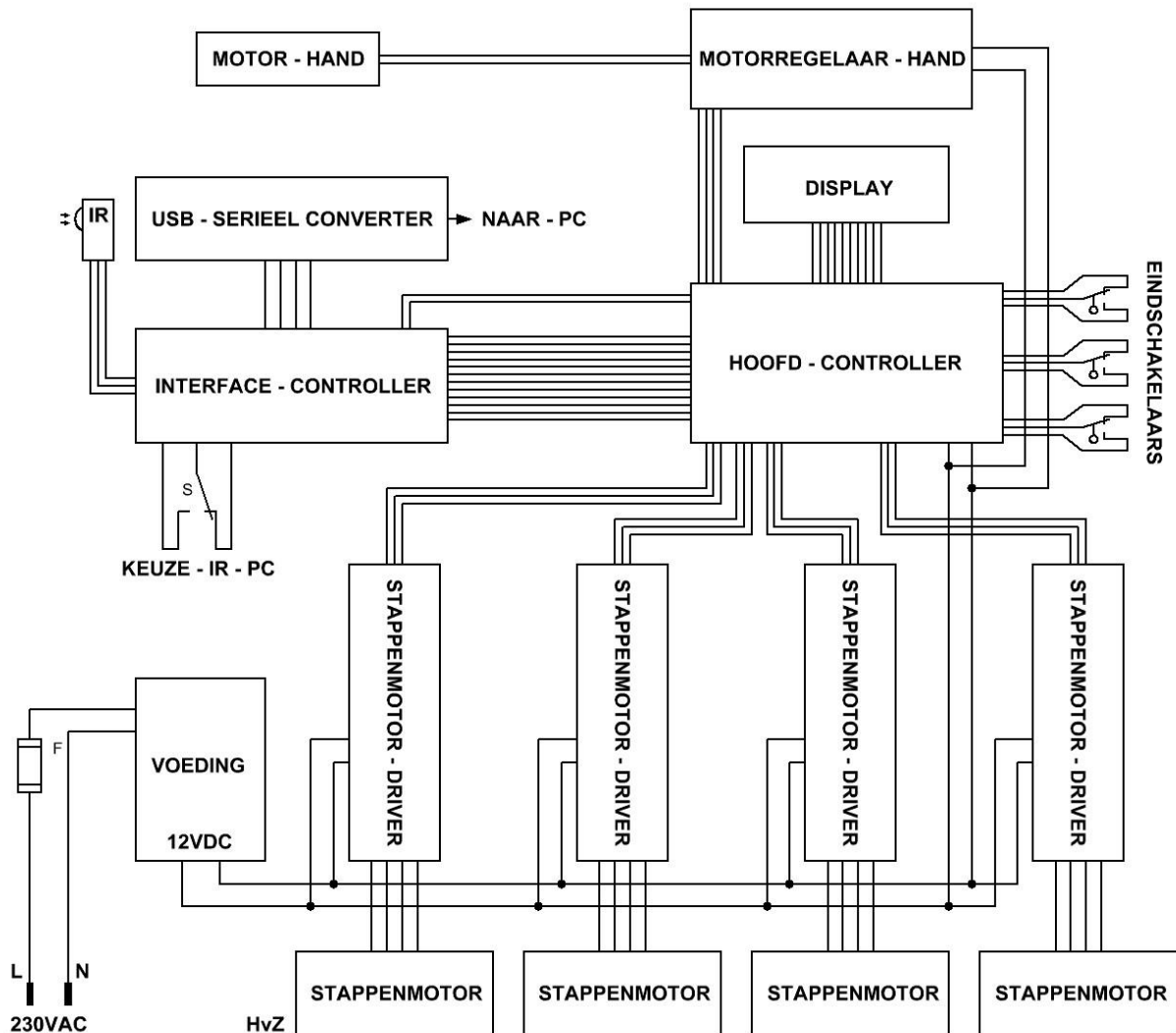
Als er dus op een desbetreffende drukknop wordt gedrukt en de drukknop wordt weer losgelaten wordt de bij behorende waarde serieel verzonden naar de interfacecontroller. De interfacecontroller stuurt een commando aan de hoofdcontroller en deze werkt de opdracht verder af. De lampjes boven sommige drukknoppen worden bediend door een teller, door de teller op een bepaalde manier te schakelen krijg je een toggle functie. Eenmaal drukken is aan, nog eenmaal drukken is uit. Ik had de lampjes ook kunnen bedienen op terug melding van de interfacecontroller, maar dat zou een stuk ingewikkelder zijn geworden en het geeft weinig meerwaarde aan het geheel. Ik kan nu ook zien of er data ontvangen is of niet.

Zoals ik al eerder aanhaalde lijkt het overbodig om hiervoor twee controllers te gebruiken, maar dat is het niet. De tijd om bijvoorbeeld het IR signaal in te lezen komt nogal precies en kost nogal wat instructie tijd. Ook het inlezen van het seriële signaal kost tijd en de controller wacht op binnen komende data voordat deze er iets mee doet. Nu is er nog wel iets mogelijk met bijvoorbeeld een interrupt, maar ik vond dat het niet lekker werkte. Dus vandaar dat ik voor deze oplossing gekozen heb en daar heb ik geen spijt van, want het werkt vlekken loos en lekker snel.

Ik gebruik nu een USB naar serieel converter om de waardes te verzenden en te ontvangen, maar dat zou ook zonder kunnen als ik de seriële poort van de PC zou gebruiken. Dus die mogelijkheid is er ook. Maar aangezien er steeds meer Pc's komen zonder seriële poort, heb ik voor deze oplossing gekozen.

We gaan nu het blokschema bekijken.

Hieronder het blokschema van de besturing.



De interfacecontroller gebruikt drie ingangen, keuze PC of IR, de data van de IR ontvanger en de data-in van de USB serieel converter. De interfacecontroller stuurt 13 signalen uit naar de hoofdcontroller, de data-uit van de USB serieel converter, basis rechtson, basis linksom, bovenarm op, bovenarm neer, onderarm op, onderarm neer, pols rechtson, pols linksom, hand open en hand dicht.

De hoofdcontroller krijgt 12 signalen van de interfacecontroller, degene die hier boven beschreven staan. De 3 ingangen voor de eindschakelaars komen direct van de eindschakelaars af. Dit zijn de eindschakelaars van de basis, bovenarm en onderarm. De assen van de pols en hand hebben geen eindschakelaar, dat is ook niet nodig omdat deze geen gevaar lopen op eventueel vastlopen. De eindschakelaars dienen eigenlijk alleen voor het homen van de assen, na het homen zijn de eindschakelaars niet meer bediend.

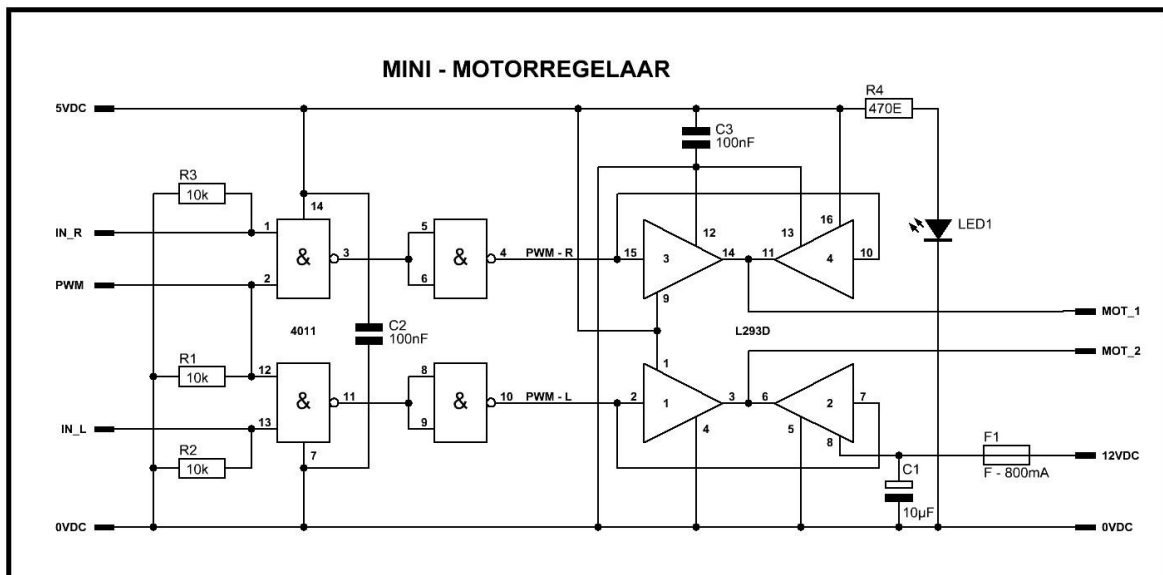
De uitgangen van de hoofdcontroller bedienen de microstepdrivers, dat is de puls en richting voor de basis, bovenarm, onderarm, pols en de hand. Dit zijn in het totaal 10 uitgangen. En dan hebben we nog 6 uitgangen voor het display.

Hier nog een plaatje van de gebruikte gripper.

De gripper is van Make-Block, de motor heeft een vertraging van 50:1. De motor drijft een kleine spindel aan die op zijn beurt de gripper bediend. De voedingspanning mag tussen de 5 en 12VDC zijn. De opgenomen stroom is maximaal 1700mA.



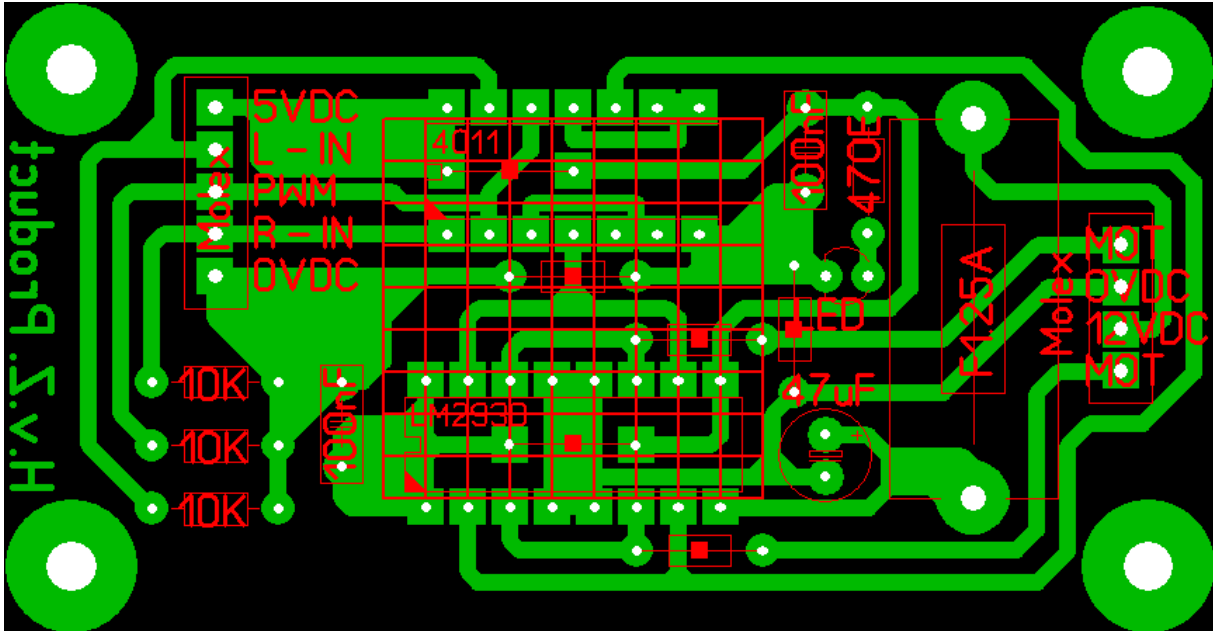
De gripper wordt bediend door een kleine motorregelaar, zie hieronder.



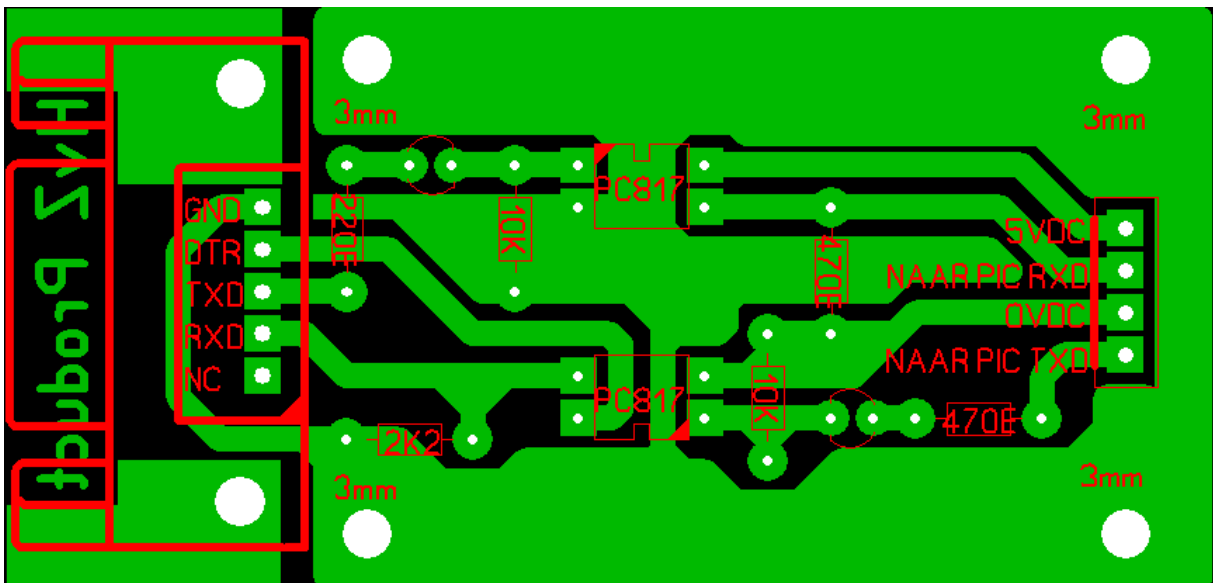
Deze regelaar heb ik al vaker laten zien en gebruikt in andere artikelen. De brug is een L293D in combinatie met een 4011. Het PWM signaal wordt hier niet toegepast, het is nu continue hoog. De ingangen IN_R en IN_L komen van de hoofdcontroller af. De regelaar mag met +/- 1.2A belast worden, maar zo hoog loopt de stroom niet op bij bediening van de gripper.

Hier nog wat printen van de regelaar, interfacecontroller, hoofdcontroller en de seriële converter.

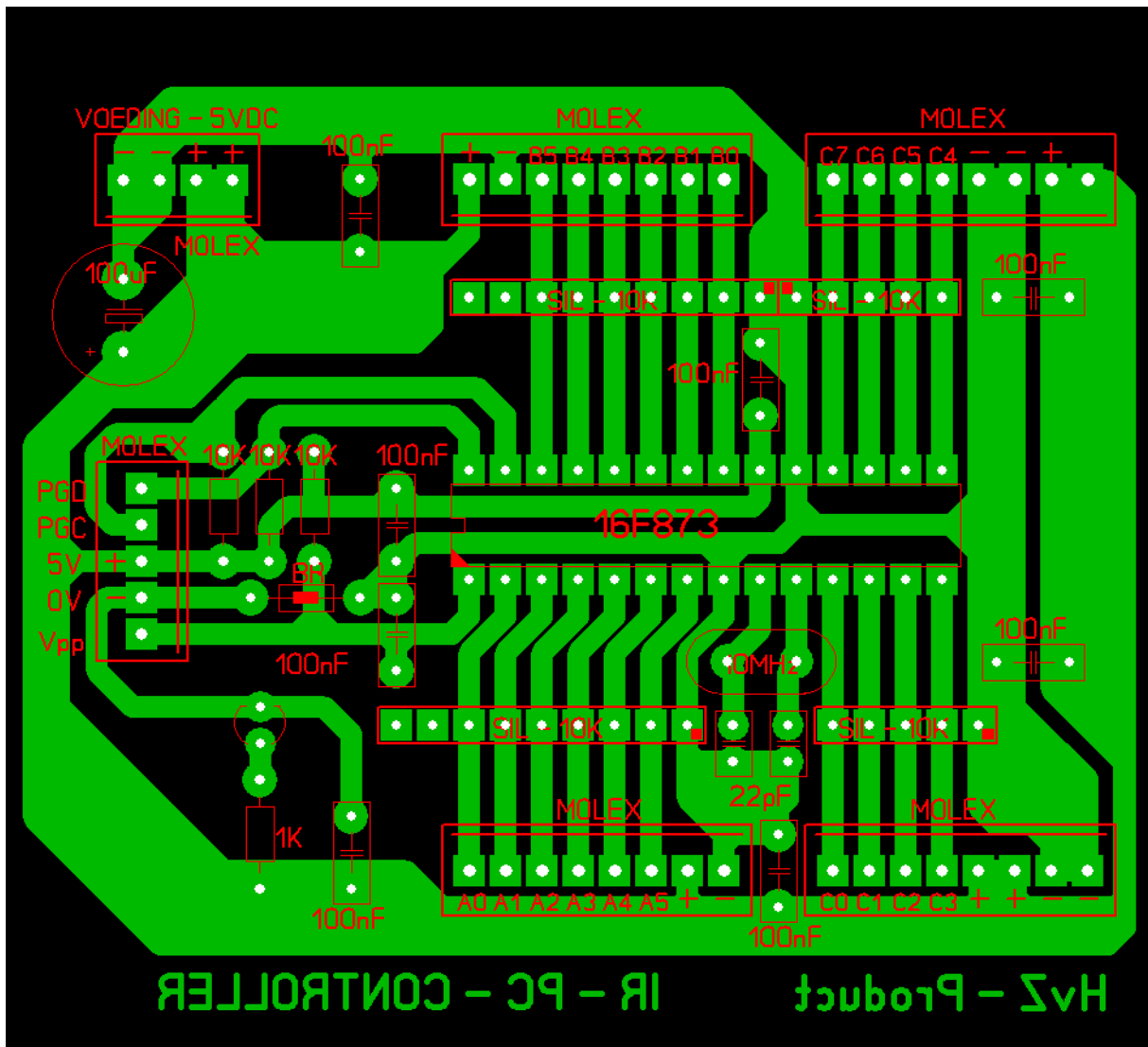
Hier het printje van de motorregelaar voor de gripper. De L293D heeft twee H_bruggen, deze heb ik parallel gezet zodat het IC meer stroom kan leveren aan de motor. Normaal is dat 600mA continue per brug, nu is dat dus 1.2A in totaal. De regelaar kan een piek aan van 2.4A, het geheel is afgezekerd met een snelle zekering van 1.25A.



Hier het printje van de USB converter, het PC signaal is galvanisch gescheiden van de microcontroller doormiddel van twee optocouplers. Hier heb ik tweemaal de PC817 toegepast, op deze manier kan ik veilig data versturen van de PC naar de controller en van de controller naar de PC.



Hier de print van de interfacecontroller, alle I/O pinnen zijn uitgevoerd op Molex connectoren. Zoals je kunt zien is hier een 16F873 gebruikt, de controller kan geprogrammeerd worden via ICSP, "in Circuit Serial Programming" op deze manier hoeft de controller niet uit de print genomen te worden. De controller draait op een 10MHz kristal. De voeding voor de interfacecontroller komt van de hoofdcontroller, hier staat een 5VDC spanningsregelaar op, een 78S05. Deze regelaar zorgt voor de voeding van alles wat op 5VDC werkt, dus dat zijn alle printen, de eindschakelaars, de IR ontvanger en het display. De regelaar is een 2A type en is beveiligd tegen over temperatuur en is kortsluit vast. De regelaar wordt met 12VDC gevoed uit de voeding die ook voor de microstepdrivers gebruikt wordt.




```
'* Name      : UITLEZEN RC5 CODE MET 16F887.
'* Author    : H.van Zwieten.
'* Notice    : Copyright (c) 2009 H.v.Z.
'*          : All Rights Reserved
'* Date      : 30-12-2009
'* Version   : 1.0
'* Notes     : 468 programma woorden en 21 variabele gebruikt.
```

```
Device 16F887 ; Processor type

Xtal 10 ; Kristal 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 & MCLRRE_OFF & PWRTE_ON
__CONFIG __CONFIG2, WRT_OFF & BOR40V
EndAsm

All_Digital true ; Alle poorten digitaal

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

Declare RC5In_Pin PORTB.0 ; ir ontvanger op poort B.0

Dim rc5_commando As Byte ; variabele RC5 commando

Cls ; Wis display

Clear ; Wis geheugen

DelayMS 2000 ; Wacht 2 seconden

;543210 ; Hulpregel poort A
PORTA = %000000 ; Maak poort A laag
TRISA = %000000 ; 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 = %000 ; Poort_E I/O

;76543210 ; Hulpregel analoog
ADCON0 = %00000001 ; ADCON0 register analoog 8 bit
```

```

        ;76543210                ; Hupregel analoog
;ADCON1 = %10000000            ; ADCON1 register analoog 10 bit

        ;76543210                ; Hulpregel analoog poort_B
;ANSELH = %00000000            ; ANSEL register analoog poort_B

;-----
; PROGRAMMA UITLEZEN IR COMMANDO.
;-----

start:
Print At 1,1,"COMMANDO"        ; Zet commando op het display
Repeat                               ; herhaal
rc5_commando = RC5In           ; Lees ir ingang
Until rc5_commando <> 255       ; Waarde 0 t/m 255
Print At 1,10,Dec rc5_commando , " " ; Zet waarde op display
GoTo start

End

```

Zoals je kunt zien zijn er niet veel regels nodig om de afstandsbediening uit te lezen, het programma wacht net zo lang tot dat er een knop op de afstandsbediening wordt ingedrukt. De waarde die onder een bepaalde knop zit wordt nu op het display getoond, op deze manier kun je dus de hele afstandsbediening uitlezen en de waarden die daar uitkomen verwerken in een programma. Dat is dus precies wat ik ook gedaan heb met het programma voor de interfacecontroller. Hieronder het programma voor de interfacecontroller.

```

'* Name      : INTERFACE IR - PC BEDIENING ROBOTARM.
'* Author    : H van Zwieten.
'* Notice    : Copyright (c) 2019 H.v.Z.
'*          : All Rights Reserved
'* Date      : 20-11-2019
'* Version   : 1.0

```

Device = 16F873

```

Config FOSC_HS, WDTE_OFF, PWRTE_OFF, CP_OFF, BOREN_OFF, LVP_OFF, CPD_OFF,
WRT_OFF, DEBUG_OFF

```

```

Xtal 10                ; Cristal 10Mhz

All_Digital true       ; Alle poorten digitaal

Declare RC5In_Pin PORTC.4 ; Ingang ir ontvanger

Declare Rsin_Pin PORTC.5  ; Ingang data in
Declare Rsout_Pin PORTC.6 ; Uitgang data uit

Declare Serial_Baud 9600 ; Baudrate 9600

Declare Serial_Data 8    ; Acht bit data

Symbol SELECT_IN = PORTC.7 ; Ingang keuze ir of pc

Symbol BASIS_RECHTS_OM = PORTB.0 ; Uitgang hand basis rechtsom
Symbol BASIS_LINKS_OM = PORTB.1  ; Uitgang hand basis linksom
Symbol BOVENARM_OMHOOG = PORTB.2 ; Uitgang hand bovenarm op

```

```

Symbol BOVENARM_OMLAAG = PORTB.3           ; Uitgang hand bovenarm neer
Symbol ONDERARM_OMHOOG = PORTB.4           ; Uitgang hand onderarm op
Symbol ONDERARM_OMLAAG = PORTB.5           ; Uitgang hand onderarm neer
Symbol POLS_RECHTS_OM = PORTA.0            ; Uitgang hand pols rechtsom
Symbol POLS_LINKS_OM = PORTA.1             ; Uitgang hand pols linksom
Symbol HOMEN_ASSEN = PORTA.2               ; Uitgang homen assen
Symbol RUN_AUTO = PORTA.3                  ; Uitgang run auto
Symbol HAND_OPEN = PORTC.0                 ; Uitgang hand open
Symbol HAND_DICHT = PORTC.1                ; Uitgang hand dicht

```

```

Dim rc5_code As Byte                     ; variabele rc5 code
Dim rc5_command As rc5_code.LowByte      ; variabele rc5 commando

```

```

Dim DATA_IN As Byte                     ; Variabele seriele data-in

```

```

DelayMS 3000                               ; Pauze 3 sec

```

```

Clear                                       ; Wis geheugen

```

```

           ;543210                            ; Hulpregel poort A
PORTA = %000000                             ; Maak poort A laag
TRISA = %000000                             ; 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 = %10110000                          ; Poort_C I/O

```

```

;-----
; PROGRAMMA IR AFSTANDBEDIENING ROBOT ARM.
; PLUS PC BEDIENING ROBOT ARM.
;-----

```

```

;-----
; SELECTEER AFSTANDBEDIENING OF BEDIENING PC.
;-----

```

```

START:

```

```

If SELECT_IN = 0 Then
  GoTo RUN_IR
EndIf

```

```

If SELECT_IN = 1 Then
  GoTo RUN_PC
EndIf

```

```

GoTo START

```

```

;-----
; UITLEZEN IR AFSTANDBEDIENING.
;-----

```

```

RUN_IR:

```

```

Repeat
  rc5_code = RC5In
Until rc5_command <> 255

```

```

If rc5_command = 1 Then GoSub START_HOMEN_ASSEN_IR
If rc5_command = 3 Then GoSub START_RUN_AUTO_IR
If rc5_command = 16 Then GoSub BASIS_RECHTS_IR
If rc5_command = 17 Then GoSub BASIS_LINKS_IR
If rc5_command = 32 Then GoSub BOVENARM_OP_IR
If rc5_command = 33 Then GoSub BOVENARM_NEER_IR
If rc5_command = 45 Then GoSub ONDERARM_OP_IR
If rc5_command = 44 Then GoSub ONDERARM_NEER_IR
If rc5_command = 22 Then GoSub POLS_RECHTS_IR
If rc5_command = 21 Then GoSub POLS_LINKS_IR
If rc5_command = 34 Then GoSub GRIPPER_OPEN_IR
If rc5_command = 18 Then GoSub GRIPPER_DICHT_IR
If rc5_command = 12 Then GoSub NOODSTOP_IR

```

GoTo RUN_IR

```

;-----
; SERIELE COMMANDO'S ONTVANGEN VAN DE PC.
;-----

```

RUN_PC:

```

DATA_IN = RSIn
DelayMS 5

If DATA_IN = 5 Then GoSub START_HOMEN_ASSEN_PC
If DATA_IN = 10 Then GoSub START_RUN_AUTO_PC
If DATA_IN = 15 Then GoSub BASIS_RECHTS_PC
If DATA_IN = 20 Then GoSub BASIS_LINKS_PC
If DATA_IN = 25 Then GoSub BOVENARM_OP_PC
If DATA_IN = 30 Then GoSub BOVENARM_NEER_PC
If DATA_IN = 35 Then GoSub ONDERARM_OP_PC
If DATA_IN = 40 Then GoSub ONDERARM_NEER_PC
If DATA_IN = 45 Then GoSub POLS_RECHTS_PC
If DATA_IN = 50 Then GoSub POLS_LINKS_PC
If DATA_IN = 55 Then GoSub GRIPPER_OPEN_PC
If DATA_IN = 60 Then GoSub GRIPPER_DICHT_PC
If DATA_IN = 200 Then GoSub NOODSTOP_PC

```

GoTo RUN_PC

```

;-----
; BEDIENING VIA IR AFSTANDBEDIENING.
;-----

```

START_HOMEN_ASSEN_IR:

```

HOMEN_ASSEN = 1
DelayMS 100
HOMEN_ASSEN = 0
DelayMS 100
Return

```

GoTo START_HOMEN_ASSEN_IR

START_RUN_AUTO_IR:

```

RUN_AUTO = 1
DelayMS 100
RUN_AUTO = 0

```

DelayMS 100
Return

GoTo START_RUN_AUTO_IR

BASIS_RECHTS_IR:

Toggle BASIS_RECHTS_OM
DelayMS 200
Return

GoTo BASIS_RECHTS_IR

BASIS_LINKS_IR:

Toggle BASIS_LINKS_OM
DelayMS 200
Return

GoTo BASIS_LINKS_IR

BOVENARM_OP_IR:

Toggle BOVENARM_OMHOOG
DelayMS 200
Return

GoTo BOVENARM_OP_IR

BOVENARM_NEER_IR:

Toggle BOVENARM_OMLAAG
DelayMS 200
Return

GoTo BOVENARM_NEER_IR

ONDERARM_OP_IR:

Toggle ONDERARM_OMHOOG
DelayMS 200
Return

GoTo ONDERARM_OP_IR

ONDERARM_NEER_IR:

Toggle ONDERARM_OMLAAG
DelayMS 200
Return

GoTo ONDERARM_NEER_IR

POLS_RECHTS_IR:

Toggle POLS_RECHTS_OM
DelayMS 200
Return

GoTo POLS_RECHTS_IR

```
POLS_LINKS_IR:
    Toggle POLS_LINKS_OM
    DelayMS 200
    Return
```

```
GoTo POLS_LINKS_IR
```

```
GRIPPER_OPEN_IR:
```

```
    HAND_OPEN = 1
    DelayMS 1200
    HAND_OPEN = 0
    DelayMS 100
    Return
```

```
GoTo GRIPPER_OPEN_IR
```

```
GRIPPER_DICHT_IR:
```

```
    HAND_DICHT = 1
    DelayMS 1200
    HAND_DICHT = 0
    DelayMS 100
    Return
```

```
GoTo GRIPPER_DICHT_IR
```

```
NOODSTOP_IR:
```

```
    BASIS_RECHTS_OM = 0
    BASIS_LINKS_OM = 0
    BOVENARM_OMHOOG = 0
    BOVENARM_OMLAAG = 0
    ONDERARM_OMHOOG = 0
    ONDERARM_OMLAAG = 0
    POLS_RECHTS_OM = 0
    POLS_LINKS_OM = 0
    DelayMS 500
```

```
    If SELECT_IN = 1 Then
        GoTo RUN_PC
    Else
        Return
    EndIf
```

```
GoTo NOODSTOP_IR
```

```
-----
; BEDIENING VIA DE PC.
; RSOUT 100 EN RSOUT 0 DIENT TER CONTROLE-
; OF HET SIGNAAL ONTVANGEN IS DOOR DE PC.
-----
```

```
START_HOMEN_ASSEN_PC:
```

```
    HOMEN_ASSEN = 1
    RSOut 100
    DelayMS 100
    HOMEN_ASSEN = 0
    RSOut 0
```

DelayMS 100
Return

GoTo START_HOMEN_ASSEN_PC

START_RUN_AUTO_PC:

RUN_AUTO = 1
RSOut 100
DelayMS 100
RUN_AUTO = 0
RSOut 0
DelayMS 100
Return

GoTo START_RUN_AUTO_PC

BASIS_RECHTS_PC:

Toggle BASIS_RECHTS_OM
RSOut 100
DelayMS 100
RSOut 0
DelayMS 100
Return

GoTo BASIS_RECHTS_PC

BASIS_LINKS_PC:

Toggle BASIS_LINKS_OM
RSOut 100
DelayMS 100
RSOut 0
DelayMS 100
Return

GoTo BASIS_LINKS_PC

BOVENARM_OP_PC:

Toggle BOVENARM_OMHOOG
RSOut 100
DelayMS 100
RSOut 0
DelayMS 100
Return

GoTo BOVENARM_OP_PC

BOVENARM_NEER_PC:

Toggle BOVENARM_OMLAAG
RSOut 100
DelayMS 100
RSOut 0
DelayMS 100
Return

GoTo BOVENARM_NEER_PC

ONDERARM_OP_PC:

```
Toggle ONDERARM_OMHOOG  
RSOut 100  
DelayMS 100  
RSOut 0  
DelayMS 100  
Return
```

GoTo ONDERARM_OP_PC

ONDERARM_NEER_PC:

```
Toggle ONDERARM_OMLAAG  
RSOut 100  
DelayMS 100  
RSOut 0  
DelayMS 100  
Return
```

GoTo ONDERARM_NEER_PC

POLS_RECHTS_PC:

```
Toggle POLS_RECHTS_OM  
RSOut 100  
DelayMS 100  
RSOut 0  
DelayMS 100  
Return
```

GoTo POLS_RECHTS_PC

POLS_LINKS_PC:

```
Toggle POLS_LINKS_OM  
RSOut 100  
DelayMS 100  
RSOut 0  
DelayMS 100  
Return
```

GoTo POLS_LINKS_PC

GRIPPER_OPEN_PC:

```
HAND_OPEN = 1  
RSOut 100  
DelayMS 1200  
HAND_OPEN = 0  
RSOut 0  
DelayMS 100  
Return
```

GoTo GRIPPER_OPEN_PC

GRIPPER_DICHT_PC:

```
HAND_DICHT = 1  
RSOut 100  
DelayMS 1200
```

```

    HAND_DICHT = 0
    RSOut 0
    DelayMS 100
    Return

GoTo GRIPPER_DICHT_PC

NOODSTOP_PC:

    BASIS_RECHTS_OM = 0
    BASIS_LINKS_OM = 0
    BOVENARM_OMHOOG = 0
    BOVENARM_OMLAAG = 0
    ONDERARM_OMHOOG = 0
    ONDERARM_OMLAAG = 0
    POLS_RECHTS_OM = 0
    POLS_LINKS_OM = 0
    RSOut 100
    DelayMS 100
    RSOut 0
    DelayMS 100

    If SELECT_IN = 0 Then
        GoTo RUN_IR
    Else
        Return
    EndIf

GoTo NOODSTOP_PC

End

```

Het programma vergt wel wat uitleg denk, onder START wordt er gekeken of de SELECT_IN schakelaar op IR staat of op PC. Als SELECT_IN 0 is dan kan de afstandsbediening gebruikt worden en als SELECT_IN 1 is dan kan de robotarm met de PC bediend worden.

Als je onder RUN_IR kijkt kun je zien welke waardes er uitgelezen worden. Om de robotarm te starten moeten eerst de assen naar de home positie gestuurd worden, dit commando staat onder knop 1 van de afstandsbediening. Als er nu op knop 1 gedrukt wordt, wordt er naar START_HOMEN_ASSEN_IR gegaan. Daar wordt een uitgang 1 en 0 gemaakt en wordt er terug gegaan naar RUN_IR. Daar staat het programma weer te wachten op de volgende opdracht. De 1 wordt ingelezen door de hoofdcontroller en voert de opdracht uit, de assen worden nu naar de home positie gestuurd.

Als de SELECT_IN schakelaar nu op PC staat, wordt er naar RUN_PC gegaan. Daar kun je zien dat de waarde DATA_IN 5 het zelfde doet, maar nu wordt er naar START_HOMEN_ASSEN_PC gegaan. Daar wordt de zelfde uitgang weer 1 en 0 gemaakt en wordt er terug gegaan naar RUN_PC, daar blijft het programma weer wachten op het volgende commando. Ook nu wordt de 1 weer ingelezen door de hoofdcontroller en wordt de opdracht uitgevoerd. En zo werkt het eigenlijk met alle commando's.

Nu kunnen we naar het programma van de hoofd controller gaan kijken.

```

'* Name      : HOOFD PROGRAMMA ROBOT ARM.
'* Author    : H van Zwieten.
'* Notice    : Copyright (c) 2019 H.v.Z.
'*          : All Rights Reserved
'* Date      : 4-11-2019
'* Version   : 1.0

```

```

;-----
; BEREKENING POSITIE BASIS:
; AANTAL GRADEN VERDRAAIING IS AANTALSTAPPEN X 0.0075.
; VOORBEELD 90 GRADEN IS 90 : 0.0075 = 12000 STAPPEN.
; BEREKENING POSITIE BOVENARM:
; AANTAL GRADEN VERDRAAIING IS AANTAL STAPPEN X 0.01125.
; VOORBEELD 90 GRADEN IS 90 : 0.01125 = 8000 STAPPEN.
; BEREKENING POSITIE ONDERARM:
; AANTAL GRADEN VERDRAAIING IS AANTAL STAPPEN X 0.01125.
; VOORBEELD 90 GRADEN IS 90 : 0.01125 = 8000 STAPPEN.
; BEREKENING POSITIE POLS:
; AANTAL GRADEN VERDRAAIING IS AANTAL STAPPEN X 0.0155306298533218.
; VOORBEELD 90 GRADEN IS 90 : 0.0155306298533218 = 5795.
;-----

```

Device = 18F4520

; Processor type

Xtal 10

; Cristal 10MHz

Config_Start

```

OSC = HS ; HS oscillator
FCMEN = OFF ; Clock Monitor disabled
IESO = OFF ; Oscillator Switcho disabled
PWRT = OFF ; PWRT disabled
BOREN = OFF ; Brown-out Reset disabled
BORV = 3 ; Minimum setting
WDT = OFF ; WDT disabled
WDTPS = 32768 ; 1:32768
CCP2MX = PORTC ; CCP2 in/out is multiplexed
PBADEN = OFF ; PORTB<4:0> as digital I/O
LPT1OSC = OFF ; Timer1 configured power
MCLR = OFF ; MCLR disabled
STVREN = On ; Stack full/underflow Reset
LVP = OFF ; Single-Supply ICSP disabled
XINST = OFF ; Instruction set extension
Debug = OFF ; Background debugger disabled
Cp0 = OFF ; Block 0 (000800-001FFFh)
CP1 = OFF ; Block 1 (002000-003FFFh)
CP2 = OFF ; Block 2 (004000-005FFFh)
CP3 = OFF ; Block 3 (006000-007FFFh)
CPB = OFF ; Boot block (000000-0007FFFh)
CPD = OFF ; Data EEPROM not code-
protected
WRT0 = OFF ; Block 0 (000800-001FFFh)
WRT1 = OFF ; Block 1 (002000-003FFFh)
WRT2 = OFF ; Block 2 (004000-005FFFh)
WRT3 = OFF ; Block 3 (006000-007FFFh)
WRTC = OFF ; Registers (300000-3000FFFh)
WRTB = OFF ; Boot block (000000-0007FFFh)
WRD = OFF ; Data EEPROM
EBTR0 = OFF ; Block 0 (000800-001FFFh)
EBTR1 = OFF ; Block 1 (002000-003FFFh)
EBTR2 = OFF ; Block 2 (004000-005FFFh)

```

```

EBTR3 = OFF ; Block 3 (006000-007FFFh)
EBTRB = OFF ; Boot block (000000-0007FFFh)
Config_End

All_Digital true ; Alle poorten digitaal

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 FREQUENTIE_BEGIN_POSITIE = 600 ; Stap frequentie positie
Symbol FREQUENTIE_HOMEN = 400 ; Stap frequentie homen
Symbol FREQUENTIE_HAND = 100 ; Stap frequentie hand 150
Symbol FREQUENTIE_AUTO = 10 ; Stap frequentie auto 20

Symbol EINDSCHAK_BASIS = PORTE.0 ; Ingang eindschak basis
Symbol EINDSCHAK_BOVENARM = PORTE.1 ; Ingang eindschak bovenarm
Symbol EINDSCHAK_ONDERARM = PORTE.2 ; Ingang eindschak onderarm

Symbol START_HOME = PORTC.0 ; Ingang start homen assen
Symbol START_AUTO = PORTC.1 ; Ingang start run automatisch

Symbol HAND_BASIS_RECHTS_OM = PORTC.3 ; Ingang hand basis rechtsom
Symbol HAND_BASIS_LINKS_OM = PORTD.0 ; Ingang hand basis linksom
Symbol HAND_BOVENARM_OMHOOG = PORTD.1 ; Ingang hand bovenarm op
Symbol HAND_BOVENARM_OMLAAG = PORTA.4 ; Ingang hand bovenarm neer
Symbol HAND_ONDERARM_OMHOOG = PORTC.7 ; Ingang hand onderarm op
Symbol HAND_ONDERARM_OMLAAG = PORTC.6 ; Ingang hand onderarm neer
Symbol HAND_POLS_RECHTS_OM = PORTC.5 ; Ingang hand pols rechtsom
Symbol HAND_POLS_LINKS_OM = PORTC.4 ; Ingang hand pols linksom
Symbol HAND_GRIPPER_OPEN = PORTA.5 ; Ingang hand gripper open
Symbol HAND_GRIPPER_DICHT = PORTC.2 ; Ingang hand gripper dicht

Symbol PULS_BASIS = PORTB.0 ; Uitgang puls basis
Symbol DIR_BASIS = PORTB.1 ; Uitgang richting basis
Symbol PULS_BOVENARM = PORTB.2 ; Uitgang puls bovenarm
Symbol DIR_BOVENARM = PORTB.3 ; Uitgang richting bovenarm
Symbol PULS_ONDERARM = PORTB.4 ; Uitgang puls onderarm
Symbol DIR_ONDERARM = PORTB.5 ; Uitgang richting onderarm
Symbol PULS_POLS = PORTA.0 ; Uitgang puls pols
Symbol DIR_POLS = PORTA.1 ; Uitgang richting pols
Symbol GRIPPER_OPEN = PORTA.2 ; Uitgang gripper open
Symbol GRIPPER_DICHT = PORTA.3 ; Uitgang gripper dicht

Dim TELLER_BASIS As Word ; Variabele teller basis
Dim TELLER_BOVENARM As Word ; Variabele teller bovenarm
Dim TELLER_ONDERARM As Word ; Variabele teller onderarm
Dim TELLER_POLS As Word ; Variabele teller pols

Dim BASIS As Word ; Variabele basis
Dim BOVENARM As Word ; Variabele bovenarm
Dim ONDERARM As Word ; Variabele onderarm
Dim POLS As Word ; Variabele pols
Dim COMBINATIE_ASSEN As Word ; Variabele combinatie assen

Cls ; Wis display

DelayMS 3000 ; Pauze 3 sec

Clear ; Wis geheugen

```

```

;543210 ; Hulpregel poort A
PORTA = %000000 ; Maak poort A laag
TRISA = %110000 ; 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 = %11111111 ; Poort_C I/O

;76543210 ; Hulpregel poort D
PORTD = %00000000 ; Maak poort D laag
TRISD = %00000011 ; 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 8 bit

;76543210 ; Hupregel analoog
;ADCON1 = %10000000 ; ADCON1 register analoog 10
bit

;76543210 ; Hulpregel analoog poort_B
;ANSELH = %00000000 ; ANSEL register analoog
poort_B

;-----
; PROGRAMMA ZELFBOUW ROBOT ARM.
;-----

;-----
; ZET TELLERS OP NUL.
;-----

TELLER_BOVENARM = 0
TELLER_ONDERARM = 0
TELLER_BASIS = 0
TELLER_POLS = 0

;-----
; START HOME PROCEDURE.
;-----

START:

Print At 1,1,"HOMEN - ASSEN"
Print At 2,1,"DRUK OP START"

If START_HOME = 1 Then
Cls
DelayMS 100
GoTo HOME_POSITIE
EndIf

GoTo START

```

```
;-----  
; HOME PROCEDURE IN WERKING.  
;-----
```

HOME_POSITIE:

```
Print At 1,1,"HOMEN - ACTIEF"
```

```
DIR_BOVENARM = 0
```

```
If EINDSCHAK_BOVENARM = 1 Then
```

```
    PULS_BOVENARM = 0
```

```
Else
```

```
    PULS_BOVENARM = 1
```

```
    DelayUS FREQUENTIE_HOMEN
```

```
    PULS_BOVENARM = 0
```

```
    DelayUS FREQUENTIE_HOMEN
```

```
EndIf
```

```
DIR_ONDERARM = 0
```

```
If EINDSCHAK_ONDERARM = 1 Then
```

```
    PULS_ONDERARM = 0
```

```
Else
```

```
    PULS_ONDERARM = 1
```

```
    DelayUS FREQUENTIE_HOMEN
```

```
    PULS_ONDERARM = 0
```

```
    DelayUS FREQUENTIE_HOMEN
```

```
EndIf
```

```
DIR_BASIS = 1
```

```
If EINDSCHAK_BOVENARM = 1 Then
```

```
    If EINDSCHAK_ONDERARM = 1 Then
```

```
        If EINDSCHAK_BASIS = 0 Then
```

```
            PULS_BASIS = 0
```

```
        Else
```

```
            PULS_BASIS = 1
```

```
            DelayUS FREQUENTIE_HOMEN
```

```
            PULS_BASIS = 0
```

```
            DelayUS FREQUENTIE_HOMEN
```

```
        EndIf
```

```
    EndIf
```

```
EndIf
```

```
If EINDSCHAK_BOVENARM = 1 Then
```

```
    If EINDSCHAK_ONDERARM = 1 Then
```

```
        If EINDSCHAK_BASIS = 0 Then
```

```
            Cls
```

```
            DelayMS 100
```

```
            GoTo BEGIN_POSITIE
```

```
        EndIf
```

```
    EndIf
```

```
EndIf
```

```
GoTo HOME_POSITIE
```

```
;-----  
; NAAR BEGIN POSITIE.  
; VANAF DEZE POSITIE START DE ARM.
```

```
;-----  
BEGIN_POSITIE:
```

```
Print At 1,1,"NAAR BEGIN"
```

```
Print At 2,1,"POSITIE"
```

```
For BOVENARM = 0 To 800 Step 1
```

```
DIR_BOVENARM = 1
```

```
PULS_BOVENARM = 1
```

```
DelayUS FREQUENTIE_BEGIN_POSITIE
```

```
PULS_BOVENARM = 0
```

```
DelayUS FREQUENTIE_BEGIN_POSITIE
```

```
Next
```

```
For ONDERARM = 0 To 800 Step 1
```

```
DIR_ONDERARM = 1
```

```
PULS_ONDERARM = 1
```

```
DelayUS FREQUENTIE_BEGIN_POSITIE
```

```
PULS_ONDERARM = 0
```

```
DelayUS FREQUENTIE_BEGIN_POSITIE
```

```
Next
```

```
For BASIS = 0 To 2800 Step 1
```

```
DIR_BASIS = 0
```

```
PULS_BASIS = 1
```

```
DelayUS FREQUENTIE_BEGIN_POSITIE
```

```
PULS_BASIS = 0
```

```
DelayUS FREQUENTIE_BEGIN_POSITIE
```

```
Next
```

```
Cls
```

```
DelayMS 100
```

```
If BOVENARM >= 800 Then
```

```
TELLER_BOVENARM = 0
```

```
Print At 1,1,"BA",Dec5 TELLER_BOVENARM
```

```
EndIf
```

```
If ONDERARM >= 800 Then
```

```
TELLER_ONDERARM = 0
```

```
Print At 2,1,"OA",Dec5 TELLER_ONDERARM
```

```
EndIf
```

```
If BASIS >= 2800 Then
```

```
TELLER_BASIS = 0
```

```
TELLER_POLS = 0
```

```
Print At 1,9,"BS",Dec5 TELLER_BASIS
```

```
Print At 2,9,"PO",Dec5 TELLER_POLS
```

```
EndIf
```

```
GoTo RUN_HAND
```

```
GoTo BEGIN_POSITIE
```

```
;-----  
; HANDBEDIENING VAN DE ARM.  
; MET DEZE BEDIENING KUN JE DE ARM INLEREN.  
;-----
```

RUN_HAND:

```
If HAND_BASIS_RECHTS_OM = 1 Then
  DIR_BASIS = 0
  If TELLER_BASIS = 12000 Then
    PULS_BASIS = 0
  Else
    PULS_BASIS = 1
    DelayUS FREQUENTIE_HAND
    If PULS_BASIS = 1 Then
      Inc TELLER_BASIS
      Print At 1,9,"BS",Dec5 TELLER_BASIS
      PULS_BASIS = 0
      DelayUS FREQUENTIE_HAND
    EndIf
  EndIf
EndIf

If HAND_BASIS_LINKS_OM = 1 Then
  DIR_BASIS = 1
  If TELLER_BASIS = 53535 Then
    PULS_BASIS = 0
  Else
    PULS_BASIS = 1
    DelayUS FREQUENTIE_HAND
    If PULS_BASIS = 1 Then
      Dec TELLER_BASIS
      Print At 1,9,"BS",Dec5 TELLER_BASIS
      PULS_BASIS = 0
      DelayUS FREQUENTIE_HAND
    EndIf
  EndIf
EndIf

If HAND_POLS_RECHTS_OM = 1 Then
  DIR_POLS = 0
  If TELLER_POLS = 5795 Then
    PULS_POLS = 0
  Else
    PULS_POLS = 1
    DelayUS FREQUENTIE_HAND
    If PULS_POLS = 1 Then
      Inc TELLER_POLS
      Print At 2,9,"PO",Dec5 TELLER_POLS
      PULS_POLS = 0
      DelayUS FREQUENTIE_HAND
    EndIf
  EndIf
EndIf

If HAND_POLS_LINKS_OM = 1 Then
  DIR_POLS = 1
  If TELLER_POLS = 59740 Then
    PULS_POLS = 0
  Else
    PULS_POLS = 1
    DelayUS FREQUENTIE_HAND
    If PULS_POLS = 1 Then
      Dec TELLER_POLS
      Print At 2,9,"PO",Dec5 TELLER_POLS
      PULS_POLS = 0
    EndIf
  EndIf
EndIf
```

```

        DelayUS FREQUENTIE_HAND
    EndIf
EndIf
EndIf

If HAND_GRIPPER_OPEN = 1 Then
    GRIPPER_OPEN = 1
Else
    GRIPPER_OPEN = 0
EndIf

If HAND_GRIPPER_DICHT = 1 Then
    GRIPPER_DICHT = 1
Else
    GRIPPER_DICHT = 0
EndIf

If HAND_BOVENARM_OMHOOG = 1 Then
    DIR_BOVENARM = 0
    If EINDSCHAK_BOVENARM = 1 Or TELLER_BOVENARM = 0 Then
        PULS_BOVENARM = 0
    Else
        PULS_BOVENARM = 1
        DelayUS FREQUENTIE_HAND
        If PULS_BOVENARM = 1 Then
            Dec TELLER_BOVENARM
            Print At 1,1,"BA",Dec5 TELLER_BOVENARM
            PULS_BOVENARM = 0
            DelayUS FREQUENTIE_HAND
        EndIf
    EndIf
EndIf

If HAND_BOVENARM_OMLAAG = 1 Then
    DIR_BOVENARM = 1
    If TELLER_BOVENARM = 15000 Then
        PULS_BOVENARM = 0
    Else
        PULS_BOVENARM = 1
        DelayUS FREQUENTIE_HAND
        If PULS_BOVENARM = 1 Then
            Inc TELLER_BOVENARM
            Print At 1,1,"BA",Dec5 TELLER_BOVENARM
            PULS_BOVENARM = 0
            DelayUS FREQUENTIE_HAND
        EndIf
    EndIf
EndIf

If HAND_ONDERARM_OMHOOG = 1 Then
    DIR_ONDERARM = 0
    If EINDSCHAK_ONDERARM = 1 Or TELLER_ONDERARM = 0 Then
        PULS_ONDERARM = 0
    Else
        PULS_ONDERARM = 1
        DelayUS FREQUENTIE_HAND
        If PULS_ONDERARM = 1 Then
            Dec TELLER_ONDERARM
            Print At 2,1,"OA",Dec5 TELLER_ONDERARM
            PULS_ONDERARM = 0
            DelayUS FREQUENTIE_HAND
        EndIf
    EndIf
EndIf

```

```

    EndIf
  EndIf
EndIf

If HAND_ONDERARM_OMLAAG = 1 Then
  DIR_ONDERARM = 1
  If TELLER_ONDERARM = 12000 Then
    PULS_ONDERARM = 0
  Else
    PULS_ONDERARM = 1
    DelayUS FREQUENTIE_HAND
    If PULS_ONDERARM = 1 Then
      Inc TELLER_ONDERARM
      Print At 2,1,"OA",Dec5 TELLER_ONDERARM
      PULS_ONDERARM = 0
      DelayUS FREQUENTIE_HAND
    EndIf
  EndIf
EndIf

If START_HOME = 1 Then
  Cls
  DelayMS 100
  GoTo HOME_POSITIE
EndIf

If START_AUTO = 1 Then
  GoTo RUN_AUTO
EndIf

```

GoTo RUN_HAND

```

;-----
; RUN ARM AUTOMATISCH.
; PROGRAMMA WORDT AUTOMATISCH AFGEWERKT.
;-----

```

```

;#####
; PLAATS HIER TUSSEN HET ROBOT ARM PROGRAMMA VOOR AUTO RUN.
;#####

```

```

;-----
; TEST ROBOT ARM PRGRAMMA.
;-----
; VERDRAAIING BASIS:
; GEEN VERDRAAIING VAN DE BASIS.
; VERDRAAIING BOVENARM:
; 14809 X 0.01125 = 166.60125 GRADEN.
; VERDRAAIING ONDERARM:
; 6947 X 0.01125 = 78.15375 GRADEN.
; VERDRAAIING POLS:
; GEEN VERDRAAIING VAN DE POLS.
;-----

```

RUN_AUTO:

```

  DelayMS 1000

  For COMBINATIE_ASSEN = 1 To 6947 Step 1
    DIR_BOVENARM = 1

```

```

PULS_BOVENARM = 1

DelayUS FREQUENTIE_AUTO

If PULS_BOVENARM = 1 Then
  Inc TELLER_BOVENARM
  Print At 1,1,"BA",Dec5 TELLER_BOVENARM
  PULS_BOVENARM = 0
  DelayUS FREQUENTIE_AUTO
EndIf

DIR_ONDERARM = 1

PULS_ONDERARM = 1

DelayUS FREQUENTIE_AUTO

If PULS_ONDERARM = 1 Then
  Inc TELLER_ONDERARM
  Print At 2,1,"OA",Dec5 TELLER_ONDERARM
  PULS_ONDERARM = 0
  DelayUS FREQUENTIE_AUTO
EndIf
Next

DelayMS 500

For BOVENARM = 6947 To 14809 Step 1
  DIR_BOVENARM = 1

  PULS_BOVENARM = 1

  DelayUS FREQUENTIE_AUTO

  If PULS_BOVENARM = 1 Then
    Inc TELLER_BOVENARM
    Print At 1,1,"BA",Dec5 TELLER_BOVENARM
    PULS_BOVENARM = 0
    DelayUS FREQUENTIE_AUTO
  EndIf
Next

DelayMS 5000

For BOVENARM = 14809 To 6947 Step - 1
  If EINDSCHAK_BOVENARM = 1 Then
    PULS_BOVENARM = 0
  EndIf

  DIR_BOVENARM = 0

  PULS_BOVENARM = 1

  DelayUS FREQUENTIE_AUTO

  If PULS_BOVENARM = 1 Then
    Dec TELLER_BOVENARM
    Print At 1,1,"BA",Dec5 TELLER_BOVENARM
    PULS_BOVENARM = 0
    DelayUS FREQUENTIE_AUTO
  EndIf

```

Next

DelayMS 500

For COMBINATIE_ASSEN = 6947 **To** 1 **Step** - 1

If EINDSCHAK_BOVENARM = 1 **Then**

PULS_BOVENARM = 0

EndIf

DIR_BOVENARM = 0

PULS_BOVENARM = 1

DelayUS FREQUENTIE_AUTO

If PULS_BOVENARM = 1 **Then**

Dec TELLER_BOVENARM

Print At 1,1,"BA",**Dec5** TELLER_BOVENARM

PULS_BOVENARM = 0

DelayUS FREQUENTIE_AUTO

EndIf

If EINDSCHAK_ONDERARM = 1 **Then**

PULS_ONDERARM = 0

EndIf

DIR_ONDERARM = 0

PULS_ONDERARM = 1

DelayUS FREQUENTIE_AUTO

If PULS_ONDERARM = 1 **Then**

Dec TELLER_ONDERARM

Print At 2,1,"OA",**Dec5** TELLER_ONDERARM

PULS_ONDERARM = 0

DelayUS FREQUENTIE_AUTO

EndIf

Next

```
-----  
; VERDRAAIING BASIS:  
; 10000 X 0.0075 = 75 GRADEN NAAR LINKS.  
; VERDRAAIING BOVENARM:  
; 14000 X 0.01125 = 157.5 GRADEN.  
; VERDRAAIING ONDERARM:  
; 10000 X 0.01125 = 112.5 GRADEN.  
; VERDRAAIING POLS:  
; GEEN VERDRAAIING VAN DE POLS.  
-----
```

DelayMS 1000

For COMBINATIE_ASSEN = 1 **To** 10000 **Step** 1

DIR_BOVENARM = 1

PULS_BOVENARM = 1

DelayUS FREQUENTIE_AUTO

If PULS_BOVENARM = 1 **Then**

```

    Inc TELLER_BOVENARM
    Print At 1,1,"BA",Dec5 TELLER_BOVENARM
    PULS_BOVENARM = 0
    DelayUS FREQUENTIE_AUTO
  EndIf

  DIR_ONDERARM = 1

  PULS_ONDERARM = 1

  DelayUS FREQUENTIE_AUTO

  If PULS_ONDERARM = 1 Then
    Inc TELLER_ONDERARM
    Print At 2,1,"OA",Dec5 TELLER_ONDERARM
    PULS_ONDERARM = 0
    DelayUS FREQUENTIE_AUTO
  EndIf

  DIR_BASIS = 1

  PULS_BASIS = 1

  If PULS_BASIS = 1 Then
    Inc TELLER_BASIS
    Print At 1,9,"BS",Dec5 TELLER_BASIS
    PULS_BASIS = 0
    DelayUS FREQUENTIE_AUTO
  EndIf
Next

DelayMS 500

For BOVENARM = 10000 To 14000 Step 1
  DIR_BOVENARM = 1

  PULS_BOVENARM = 1

  DelayUS FREQUENTIE_AUTO

  If PULS_BOVENARM = 1 Then
    Inc TELLER_BOVENARM
    Print At 1,1,"BA",Dec5 TELLER_BOVENARM
    PULS_BOVENARM = 0
    DelayUS FREQUENTIE_AUTO
  EndIf
Next

DelayMS 5000

For COMBINATIE_ASSEN = 10000 To 1 Step - 1
  If EINDSCHAK_BOVENARM = 1 Then
    PULS_BOVENARM = 0
  EndIf

  DIR_BOVENARM = 0

  PULS_BOVENARM = 1

  DelayUS FREQUENTIE_AUTO

```

```

If PULS_BOVENARM = 1 Then
  Dec TELLER_BOVENARM
  Print At 1,1,"BA",Dec5 TELLER_BOVENARM
  PULS_BOVENARM = 0
  DelayUS FREQUENTIE_AUTO
EndIf

If EINDSCHAK_ONDERARM = 1 Then
  PULS_ONDERARM = 0
EndIf

  DIR_ONDERARM = 0

  PULS_ONDERARM = 1

  DelayUS FREQUENTIE_AUTO

If PULS_ONDERARM = 1 Then
  Dec TELLER_ONDERARM
  Print At 2,1,"OA",Dec5 TELLER_ONDERARM
  PULS_ONDERARM = 0
  DelayUS FREQUENTIE_AUTO
EndIf

  DIR_BASIS = 0

  PULS_BASIS = 1

If PULS_BASIS = 1 Then
  Dec TELLER_BASIS
  Print At 1,9,"BS",Dec5 TELLER_BASIS
  PULS_BASIS = 0
  DelayUS FREQUENTIE_AUTO
EndIf
Next

DelayMS 500

For BOVENARM = 4000 - 1 To 1 Step - 1
  If EINDSCHAK_BOVENARM = 1 Then
    PULS_BOVENARM = 0
  EndIf

  DIR_BOVENARM = 0

  PULS_BOVENARM = 1

  DelayUS FREQUENTIE_AUTO

  If PULS_BOVENARM = 1 Then
    Dec TELLER_BOVENARM
    Print At 1,1,"BA",Dec5 TELLER_BOVENARM
    PULS_BOVENARM = 0
    DelayUS FREQUENTIE_AUTO
  EndIf
Next

If START_AUTO = 0 Then
  GoTo RUN_HAND
EndIf

```

```

If START_AUTO = 1 Then
  GoTo RUN_AUTO
EndIf

```

```

GoTo RUN_AUTO

```

```

;#####
; PLAATS HIER TUSSEN HET ROBOT ARM PROGRAMMA VOOR AUTO RUN.
;#####

```

```

End

```

Voordat we naar het programma van de hoofd controller gaan kijken, wil ik eerst wat uitleggen over de vertragingen van de stappenmotoren.

De vier stappenmotoren zijn alle vier 200 stappen per omwenteling. We hebben drie grote motoren met een vertraging van 20:1 en één daarvan heeft nog een extra vertraging van 1.5:1. De kleine stappenmotor heeft een vertraging van 115.9:1.

De motor voor de basis heeft een extra vertraging in de vorm van een tandriem overbrenging. Waarom dit gedaan is, is in het begin al beschreven. Als we nu willen weten hoeveel stappen er gemaakt moeten worden voor een X aantal graden verdraaiing, moeten we een kleine berekening maken.

Voor de basis wordt dat:

De microstep driver stellen we in op 1/8 stap, dan krijgen we dus 200 stappen x 8 = 1600 x de 20:1 vertraging = 32000 x de tandriem vertraging van 1.5:1 = 48000. We moeten dus 48000 stappen maken om 360 graden te verdraaien. Dus 360 graden : 48000 = 0.0075 graden per stap. Willen we nu bijvoorbeeld 90 graden verdraaien dan moeten we 90 : 0.0075 = 12000 stappen doen.

Voor de boven en onderarm wordt dat:

De microstep driver stellen we hier ook in op 1/8 stap, dan krijgen we dus 200 stappen x 8 = 1600 x de vertraging van 20:1 = 32000. We hebben voor deze assen geen extra vertraging meer, nu wordt dat 360 graden : 32000 = 0.01125. Willen we weer 90 graden verdraaien dan krijgen we dus 90 : 0.01125 = 8000 stappen doen.

Voor de pols wordt dat:

De microstep driver stellen we hier in op hele stappen, dus 200 stappen is één omwenteling van de stappenmotor. Op de motor zit een vertraging van 115.9:1, dan krijgen we dus 200 x 115.9 = 23180. Nu wordt het 360 graden : 23180 = 0.0155306298533218. Willen we nu 90 graden verdraaien dan moeten we 90 : 0.0155306298533218 = 5795 stappen doen.

Dit moeten we weten omdat je anders nooit een programma kunt maken dat automatisch de arm naar een bepaalde positie kan sturen. Voor de hand bediening maakt het verder niet uit, omdat je het hier zelf kunt bepalen. Je laat gewoon de assen bewegen totdat je het goed vindt.

Hier ook weer wat uitleg over het programma zelf.

Zoals je kunt zien gebruik ik vier verschillende frequenties, FREQUENTIE_BEGIN_POSITIE = 600, FREQUENTIE_HOMEN = 400, FREQUENTIE_HAND = 100 en FREQUENTIE_AUTO = 10. De waardes zijn in microseconden. Daaronder staan de in en uitgangen, dat wijst voor zich denk. Daaronder staan de tellers, er zijn vier tellers aangemaakt om de positie van de assen bij te houden. De waardes van de tellers lopen van 0 tot 65535. De tellers worden op 0 gezet voordat er naar START gegaan wordt.

Als we bij START kijken dan zien we dat daar gevraagd wordt om het home te starten. Als we nu op knop 1 van de afstandsbediening drukken zal er naar de HOME_POSITIE gegaan worden, daar lopen de bovenarm, onderarm en de basis naar de eindschakelaars toe. Als de assen de eindschakelaars bereikt hebben wordt er naar BEGIN_POSITIE gegaan. Daar lopen de bovenarm en de onderarm 800 stappen uit de eindschakelaars en de basis 2800 stappen. Als de waardes bereikt zijn worden de tellers weer op 0 gezet en er wordt naar RUN_HAND gegaan, de drie assen staan nu in de 0 positie en vanuit die positie wordt er een as aangestuurd. De pols zit hier niet bij omdat de hand meestal recht staat, staat de hand niet recht dan kan die met de hand recht gezet worden via de afstandsbediening of via de PC. Daarna moet er wel nog een keer op home gedrukt worden om de teller van de pols ook weer op 0 te zetten, maar dat duurt maar een paar seconden omdat de assen al op 0 staan.

In RUN_HAND kan nu ieder as afzonderlijk of allemaal tegelijk aangestuurd worden, de tellers houden nu de positie bij waar de assen zich bevinden. De assen worden ook door de tellers beveiligd, de basis kan niet verder draaien dan 90 graden naar links en naar rechts. Je ziet daar een maximale waarde van 12000 en 53535 staan. Voor de pols is dat 5795 en 59740, die is dus ook op 90 graden beveiligd. De bovenarm kan bewegen tussen 0 en 15000 en de onderarm tussen 0 en 12000. Er kan op deze manier nooit iets stuk draaien.

Ik gebruik de handbediening ook om een automatisch programma te maken, ik stuur de arm naar een bepaalde positie en noteer dan de teller standen. Deze standen neem ik dan op in het automatische programma, zo kan ik een enkele as sturen of een combinatie van assen.

Onder knop 3 van de afstandsbediening wordt er naar RUN_AUTO gegaan, daar wordt het automatische programma afgewerkt. Dat gebeurt maar eenmaal, als je het automatische programma weer wil starten moet er weer op knop 3 van de afstandsbediening gedrukt worden. Dit kan natuurlijk ook via de PC als de PC geselecteerd is en het Profilab programma draait.

Zo dit was het weer, ik hoop dat dit meer mensen aanspoort tot het bouwen van een robotarm. Het is echt leuk om te doen, en ook het artikel schrijven is leuk en dat is dan weer leerzaam voor anderen hoop ik. Al moet ik wel zeggen dat dit geen beginners project is, maar je kunt het ook eenvoudiger maken met bijvoorbeeld modelbouw servo's. Maar zoals ik al eerder zei, daar is deze arm iets te groot voor.

H van Zwieten.