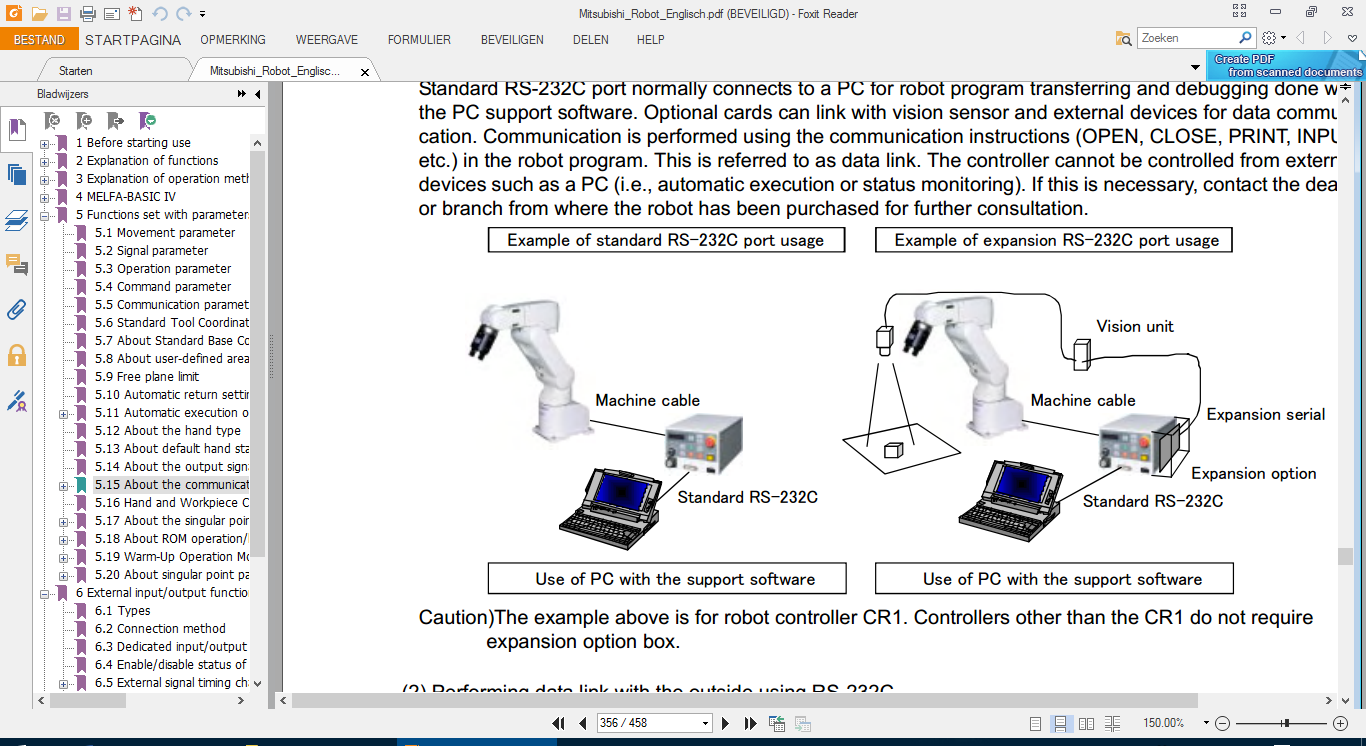
# Onderzoek communicatie

De robot bestaat uit twee onderdelen: de controller en de arm. De controller verzorgt de aansturing van de arm. Voor het aansturen van de controller bestaat de mogelijkheid om deze aan te sturen met de RS-232C aansluiting aan de voorzijde. Voor het onderzoek naar communicatie is uitgevoerd door informatie uit de handleiding in de praktijk te testen.



Figuur Schematische weergave aansluiting robot (Mitsubishi Electric, 2005)

## Communicatiemethoden

Volgens de uitgebreid specificaties van de controller (Mitsubishi Electric, 2005) kan er op twee manieren via de seriële RS-232C verbinding gecommuniceerd worden:

* Cosimir commando’s sturen
* Communiceren met de zelf ontwikkelde software op de robot arm

Het communiceren met zelf ontwikkelde software functioneert ook via Cosimir commando’s

De directe commando’s ondersteunen meerdere robots op één communicatie kanaal. Daarom moet bij het geven van een commando het robotnummer aangegeven worden. Voor multitasking bevat de robot verschillende sloten waar taken worden uitgevoerd. Deze sloten zullen dan parallel worden uitgevoerd. Hierdoor moet ook worden aangegeven met welk slot gecommuniceerd moet worden.

Het volledige commando komt er afhankelijk van of er één of meerdere parameters nodig zijn als volgt uit te zien:

* “Robotnr;slotnr;commando=(parameters)”
* “Robotnr;slotnr;commando parameter”

### Activeren van de RS-232C poort en configuratie van de software

Voor de communicatie via de RS-232C poort dient de poort te worden geconfigureerd. Dit kan door middel van de Teaching Pendant of via de Cosimir software. Voor de Cosimir software is het een vereiste dat de poort al is geactiveerd. Welke parameters dit zijn, waarop ze ingesteld moeten zijn en wat het betekent is te vinden in de onderstaande tabel. Om de communicatie te kunnen gebruiken om de robot te laten bewegen moet de sleutelschakelaar op de controller op auto(ext) staan.

|  |  |  |  |
| --- | --- | --- | --- |
| Parameter | Standaard waarde | Nieuwe waarde | Betekenis |
| COMDEV | RS232, , , , , , , | RS232, OPT11, , , , , , | Zet het communicatie kanaal open voor invoer van buiten. De lege ruimte is om extra invoer van de controller te activeren. OPT11 is om op slot 1 de COM poort uit te lezen. |
| CBAU232 | 9600 | 115200 | Baudrate van de COM poort |
| CPRTY232 | 2 | 2 | Pariteit van de verbinding (even) |
| CSTOP232 | 2 | 2 | Aantal stop bits |
| CTERM232 | 0(CR) | 1(CRLF) | Terminator voor commando’s |
| CPRC232 | 0 | 2 | Communicatie op datalink zetten zodat er in de software gebruik gemaakt kan worden van de COM poort |

Tabel Tabel met de in te stellen parameters (Mitsubishi Electric, 2005)

### Commando’s sturen over RS-232C

Het sturen van commando’s is geen officieel ondersteunde methode om te communiceren met de controller. Melfa beschrijft dan ook een beperkt aantal commando’s in hoofdstuk 6.3 van de uit gebreide specificaties van de controller (Mitsubishi Electric, 2005). Binnen deze commando’s is geen beschrijving te vinden om de robot arm anders aan te sturen dan in dan via jog commando’s (kleine bewegingen). In de Cosimir software is dit wel mogelijk. Door te luisteren met seriële poort monitor software op de communicatie tussen de robot en de Cosimir zijn extra commando’s ontdekt. De gevonden commando’s zijn in Tabel 2 te vinden. Om deze commando’s uit te kunnen voeren dient het Cosimir programma op één van de sloten van de controller geactiveerd zijn. Wanneer dit programma niet aanwezig is op de robot kan dit met de Cosimir software worden toegevoegd.

|  |  |
| --- | --- |
| **Commando** | **Effect** |
| OPEN=usertool | Verbinding openen |
| CTNLON | Control starten |
| SRVON | Servo aan |
| EXECJOVRD 100.0 | Snelheid instellen |
| EXECJCOSIROP=(-90.00,-60.00,-30.00,-30.00,-30.00,-30.00) | Joint Coördinaten instellen |
| EXECMOV JCOSIROP | Arm bewegen coördinaat |
| JPOSF | Joint posities |
| STATE | Huidige toestand van de arm |
| PPOSF | Huidige coördinaten |
| EXECPCOSIROP=(70.00,0.00,782.00,0.00,0.00,0.00)(6,0) | XYZ Coördinaten instellen |
| EXECMOV PCOSIROP | Arm bewegen naar coördinaat |
| CNTLOFF | Control uit |
| SRVOFF | Servo uit |
| RSTALRM | Alarm resetten |
| EXECSPD 200.0 | Uitvoersnelheid instellen |
| EXECMVS PCOSIROP | Jog arm beweging |
| PARRLNG | Onbekend |

Tabel Beschrijving van niet gedocumenteerde commando's

### Software op de robot

Naast het sturen van commando’s via RS-232C is het ook mogelijk om software op de controller aan te sturen. Voor het programmeren van de software op de controller zijn twee programmeertalen mogelijk: Melfa-basic 4 of Movemaster-command. Waar deze talen zich in onderscheiden is geen onderzoek voor te vinden met zoekwoorden “melfa basic vs movemaster command”, “movemaster command” en “Melfa programming” op Google en Bing. In de documentatie van de controller wordt wel Movemaster-command toegelicht maar wordt niet gesproken over voordelen of nadelen van specifieke talen. Hierdoor wordt aangenomen dat de talen voornamelijk verschillen in syntax. In de documentatie worden de voorbeelden veelal in Melfa Basic gegeven.

Om de keuze voor de programmeertaal in te stellen is er de parameter RLING nodig. Wanneer deze parameter op 1 staat kan Melfa-basic 4 gebruikt worden en wanneer deze op 0 staat kan Movemaster-command gebruikt worden.

Wanneer er gecommuniceerd gaat worden met behulp van software op de robot moet er in de software een COM poort worden geopend, te zien op regel 10 in het voorbeeld hieronder. Daarna wordt er data ontvangen via RS-232C en weer terug gestuurd.

**Command Comment**

10 OPEN "COM2:" AS #1 'Opens RS-232C communication.

20 INPUT #1,C1$ 'Inputs data.

30 PRINT #1,C1$ 'Outputs data.

40 GOTO 20 ‘Back to data input

Om via de seriële verbinden de data bij de software te krijgen is het volgende commando nodig:

“PRN data”.

### Vergelijking

Om het verschil tussen de twee communicatie methoden duidelijk te maken is in de tabel hieronder een vergelijking te zien tussen de twee methoden op twee manieren.

|  |  |  |
| --- | --- | --- |
|  | **Cosimir Commando’s** | **Communicatie met zelf ontwikkelde software** |
| Leercurve | Behalve een aantal gegeven commando’s die uitgevoerd moeten worden hoeft er niets extra’s geleerd worden | Er moet een extra programmeertaal, Movemaster-command of Melfa Basic geleerd worden. |
| Reactiesnelheid | Het is niet bekend hoe de robot het commando in het Cosimir programma uitvoert en daardoor kan er ook geen uitspraak gedaan worden over de reactiesnelheid. | Omdat de commando’s op de robot met zelf geschreven software wordt uitgevoerd kan er meer gezegd worden over de volgorde waarop opdrachten worden uitgevoerd. Echter zijn er nog steeds veel onbekenden, maar minder als bij gebruik van de Cosimir Commando’s |
| Parallel uitvoeren van taken | Alle taken worden hoogst waarschijnlijk Sequentieel uitgevoerd omdat het programma maar in één slot draait maar de bron is niet bekend. | Er is een mogelijkheid om taken parallel uit te voeren in meerdere sloten. Echter wordt er maar één regel code tegelijk uitgevoerd waardoor de uitvoertijd niet sneller zal zijn (Mitsubishi Electric, 2005) . |
| Communicatie | Altijd posities verzenden dus voor een slag beweging zullen meerdere positie commando’s gestuurd moeten worden. Daarnaast zijn moet er voor het veranderen van de positie twee commando’s gestuurd worden. | Er kunnen specifieke eigen commando’s verstuurd hierdoor hoeven niet alle locaties worden verstuurd en kan een slag beweging mogelijk met een enkel commando verstuurd worden. |
| Debugging/Error afhandeling | Alle fouten afhandelen op de pc en er kan geen data gepushed worden voor debugging alleen via polling is dit mogelijk. | In de software kunnen fouten worden afgehandeld en er kan ook data gepushed worden naar de server. |

## Conclusie

In de vergelijking uit paragraaf 1.1.4 is te zien dat behalve op het punt leercurve het realiseren van een eigen programma voor de controller de betere keuze. Doordat er minder commando’s moeten worden gestuurd kan er al performance winst geboekt worden ten opzichte van het gebruik van Cosimir commando’s. Voor de realisatie van de tafeltennis functionaliteit is het van belang dat de reactiesnelheid zo snel mogelijk is. Doordat het zelf geschreven programma op de robot kan data kan verzenden zonder dat hier specifiek op gevraagd wordt kan er voor zorgen dat er sneller gereageerd kan worden op fouten in de robot door de desktop software.

De keuze voor een programmeer taal om software voor de controller te programmeren gaat tussen Melfa Basic IV en Movemaster-command. De verschillen tussen deze twee talen worden nergens toegelicht en lijken zich dus enkel te onderscheiden in syntax. Echter wordt in de documentatie bij voorbeeld code overal Melfa Basic gebruikt, om die reden is de keuze voor Melfa Basic IV het meest logisch. Door de diverse voorbeelden is het ook makkelijker om de taal te leren.