

# INHOUDSOPGAVE

|          |   |           |          |   |           |
|----------|---|-----------|----------|---|-----------|
| <b>1</b> | <b>Inleiding – fischertechniek modellen besturen met ROBO Pro.....</b>      | <b>3</b>  | <b>5</b> | <b>Niveau 3: Variabelen, bedieningsvelden &amp; Co.....</b>                 | <b>36</b> |
| 1.1      | Installatie van ROBO Pro.....   | 3         | 5.1      | Variabelen en bevelen.....  | 36        |
| 1.2      | Installatie van de USD-driver voor de ROBO-Interface.....                   | 4         | 5.2      | Variabelen en meerdere processen.....                                       | 38        |
| 1.3      | Eerste stappen.....   | 7         | 5.3      | Bedieningsvelden.....   | 38        |
| <b>2</b> | <b>Voor de programmering een korte test van de hardware.....</b>            | <b>11</b> | 5.4      | Timer.....  | 42        |
| 2.1      | De interface op de PC aansluiten.....                                       | 11        | 5.5      | Bevelingen voor subprogramma's.....   | 43        |
| 2.2      | Opdat de verbinding goed zou werken – de instellingen van de interface..... | 11        | 5.6      | Lijsten (Arrays).....   | 46        |
| 2.3      | Als de verbinding niet goed werkt - geen verbinding met de Interface?.....  | 13        | 5.7      | Operatoren.....   | 47        |
| 2.4      | Werkt alles – de Interfacetest.....   | 13        | <b>6</b> | <b>Uitbreidingsmodule en meerdere interfaces aansturen.....</b>             | <b>51</b> |
| <b>3</b> | <b>Niveau 1: Je eerste besturingsprogramma.....</b>                         | <b>15</b> | 6.1      | Uitbreidingsmodule.....   | 51        |
| 3.1      | Een nieuw programma maken.....  | 15        | 6.2      | Meerdere interfaces.....  | 51        |
| 3.2      | De elementen van het besturingsprogramma.....                               | 16        | 6.3      | Interface-toewijzingen in subprogramma's.....                               | 53        |
| 3.3      | Programma-elementen invoegen, verschuiven en veranderen.....                | 16        | 6.4      | Tips en tricks.....   | 54        |
| 3.4      | Verbinden van de programma-elementen.....                                   | 19        | 6.5      | Wijzigen van het serienummer van de Interface of van de firmwareversie..... | 54        |
| 3.5      | Testen van het eerste besturingsprogramma.....                              | 20        | <b>7</b> | <b>Overzicht van programma-elementen.....</b>                               | <b>56</b> |
| 3.6      | Verdere programma-elementen.....  | 22        | 7.1      | Basiselementen (Niveau 1).....  | 56        |
| 3.6.1    | Wachttijd.....  | 22        | 7.1.1    | Start.....  | 56        |
| 3.6.2    | Wachten op ingang.....  | 23        | 7.1.2    | Einde.....  | 56        |
| 3.6.3    | Impulsteller.....   | 23        | 7.1.3    | Digitale vertakking.....  | 57        |
| 3.6.4    | Tellus.....   | 24        | 7.1.4    | Analoge vertakking.....   | 57        |
| 3.7      | Online- of Download-bedrijfsmodus – Wat is precies het verschil?.....       | 24        | 7.1.5    | Wachttijd.....  | 58        |
| 3.8      | Tips en tricks.....   | 27        | 7.1.6    | Motoruitgang.....   | 58        |
| <b>4</b> | <b>Niveau 2: Werken met subprogramma's.....</b>                             | <b>28</b> | 7.1.7    | Lampuitgang (Niveau2).....  | 59        |
| 4.1      | Je eerste subprogramma.....   | 29        | 7.1.8    | Wachten op Ingang.....  | 61        |
| 4.2      | De bibliotheek van de subprogramma's.....                                   | 33        | 7.1.9    | Impulsteller.....   | 62        |
| 4.2.1    | Gebruiken van de bibliotheek.....   | 33        | 7.1.10   | Tellus.....   | 62        |
| 4.2.2    | Het gebruik van de eigen bibliotheek.....                                   | 34        | 7.2      | Subprogramma I/O (Niveau2-3).....   | 63        |
| 4.3      | Bewerken van subprogrammasymbolen.....                                      | 34        | 7.2.1    | Subprogramma-ingang (Niveau 2).....   | 63        |
|          |   |           |          | Subprogramma-uitgang (Niveau 2).....  | 63        |
|          |   |           | 7.2.2    | Subprogramma-bevelingang (Niveau 3).....                                    | 63        |
|          |   |           | 7.2.3    | Subprogramma-beveluitgang (Niveau 3).....                                   | 64        |
|          |   |           | 7.2.4    | Subprogramma-beveluitgang (Niveau 3).....                                   | 64        |
|          |   |           | 7.3      | Variabele, Lijst, ... (Niveau 3).....                                       | 64        |
|          |   |           | 7.3.1    | Variabele (globaal).....  | 64        |
|          |   |           | 7.3.2    | Lokale variabele.....   | 65        |
|          |   |           | 7.3.3    | Constanten.....   | 66        |
|          |   |           | 7.3.4    | Timer-variabele.....  | 66        |

|        |  |    |
|--------|--|----|
| 7.3.5  | Lijst.....   | 67 |
| 7.4    | Bevelen (Niveau 3).....                                    | 69 |
| 7.4.1  | = ( Toewijzen ) .....                                      | 70 |
| 7.4.2  | + ( Plus ) .....   | 70 |
| 7.4.3  | - ( Min ) .....  | 71 |
| 7.4.4  | - Rechts .....   | 71 |
| 7.4.5  | - Links .....  | 71 |
| 7.4.6  | - Stop.....  | 71 |
| 7.4.7  | - Aan .....  | 71 |
| 7.4.8  | - Uit.....   | 71 |
| 7.4.9  | - Tekst.....   | 72 |
| 7.4.10 | - Waarde toevoegen .....                                   | 72 |
| 7.4.11 | - Waarde(n) verwijderen .....                              | 72 |
| 7.4.12 | - Waarde vervangen .....                                   | 72 |
| 7.5    | Vergelijkingen, wachten op, ... (Niveau 3).....            | 72 |
| 7.5.1  | Vertakking ( met data-ingang) ....                         | 73 |
| 7.5.2  | Vergelijking met vaste waarde ...                          | 73 |
| 7.5.3  | Vergelijking .....   | 74 |
| 7.5.4  | Wachttijd.....   | 74 |
| 7.5.5  | Wachten op.....  | 74 |
| 7.5.6  | Impulsteller .....   | 75 |
| 7.6    | Interface-Ingangen / -Uitgangen, ..... 75                  | 75 |
| 7.6.1  | Digitale ingang .....                                      | 75 |
| 7.6.2  | Analoge ingang .....                                       | 76 |
| 7.6.3  | Infrarood-ingang.....                                      | 77 |
| 7.6.4  | Motoruitgang .....   | 78 |
| 7.6.5  | Lampuitgang .....  | 79 |
| 7.6.6  | Bedieningsveld-ingang .....                                | 80 |
| 7.6.7  | Bedieningsveld-uitgang .....                               | 81 |
| 7.7    | Operatoren .....   | 81 |
| 7.7.1  | Aritmetische operatoren.....                               | 82 |
| 7.7.2  | Logische operatoren.....                                   | 82 |
| 8      | Overzicht van bedieningselementen en bedieningsvelden..... | 83 |
| 8.1    | Weergaven .....  | 83 |
| 8.1.1  | Meetapparaat.....  | 83 |
| 8.1.2  | Tekstweergave.....   | 84 |
| 8.1.3  | Weergavelamp .....   | 86 |
| 8.2    | Besturingselementen.....                                   | 86 |
| 8.2.1  | Knop.....  | 86 |
| 8.2.2  | Regelaars .....  | 87 |
| 9      | Tekenfuncties.....   | 89 |

# 1 Inleiding – fischertechnik modellen besturen met ROBO Pro

Je hebt je waarschijnlijk ook wel al eens afgevraagd hoe dat nu precies gaat als robots sommige taken uitvoeren alsof ze door de hand van een geest bewogen worden. Maar de besturings- en de automatiseringstechniek is niet alleen terug te vinden bij echte robots, maar ook in tal van andere domeinen. Bijvoorbeeld ook bij fischertechnik. Zo zullen we reeds in het derde hoofdstuk samen een klein besturingsprogramma voor een automatische garagepoort ontwerpen en daarbij leren hoe men met behulp van de software ROBO Pro voor Windows dergelijke besturingstaken kan oplossen en testen. De bediening van ROBO Pro is daarbij erg gemakkelijk. Op het grafisch bedieningsoppervlak kunnen de besturingsprogramma's, of beter gezegd de procedureplannen en later de datastrooplannen, zoals we nog zullen leren, bijna uitsluitend met behulp van de muis opgemaakt worden.

Om je fischertechnik modellen via de PC te kunnen besturen, heb je naast de besturingssoftware ROBO Pro nog een interface nodig als verbindingsschakel tussen de computer en het model. Deze vormt de bevelen van de software zodanig om dat bijvoorbeeld de motoren aangestuurd worden en dat de signalen door de sensoren verwerkt kunnen worden. Fischertechnik heeft een ROBO Interface art. nr. 93293 en de oudere Intelligent Interface art. nr. 30402. Beide Interfaces kan je samen met ROBO Pro gebruiken. ROBO Pro steunt evenwel alleen de Online-bedrijfsmodus van de Intelligent Interface. De oude parallelle interface art. nr. 30520 wordt door ROBO Pro niet meer ondersteund.

Nog een paar woorden over de opbouw van dit handboek. Het is onderverdeeld in twee delen. Het eerste deel van hoofdstuk 1 tot hoofdstuk 4 beschrijft de basiswerkwijze bij het programmeren met ROBO Pro. Daarbij krijg je veel informatie en achtergrond over het programmeren in het algemeen en over de bedieningswijze van de software ROBO Pro.

Het tweede deel omvat het hoofdstuk 5 tot 7 en geeft informatie over de functies voor gevorderde programma's.

De hoofdstukken vanaf hoofdstuk 8 zijn eerder bedoeld als naslagwerk. Als je dus na het eerste deel vertrouwd bent met de bediening van ROBO Pro en specifieke informatie zoekt, vind je daar een uitvoerige uitleg over de diverse programma-elementen.

Maar nu gaan we beginnen! Je bent vast al razend benieuwd welke mogelijkheden je met de software ROBO Pro hebt om je fischertechnik modellen te programmeren. Veel plezier!

## 1.1 Installatie van ROBO Pro

De voorwaarden voor de installatie van ROBO Pro zijn:

- Een IBM-compatibele PC met Pentium-processor met minstens 600 MHz kloksnelheid, 32 MB RAM en ongeveer 20 MB vrije opslagruimte op de harde schijf.
- Een monitor en een grafische kaart met een resolutie van minstens 1024x768 beeldpunten. Bij monitoren met een beeldbuis moet de opfrisfrequentie minstens 85 Hertz bedragen om een flikkervrij beeld te verkrijgen. Platte TFT-beeldschermen leveren bij elke opfrisfrequentie een flikkervrij beeld, zodat de opfrisfrequentie bij TFT-beeldschermen onbelangrijk is.
- Microsoft, Windows, Versie Windows 95, 98, ME, NT4.0, 2000 of XP
- Een vrije USB-interface of een vrije RS232-interface COM1 tot COM4 voor de aansluiting van de ROBO Interface — art. Nr. 93293 — of een vrije RS232-interface COM1 tot COM4 voor de aansluiting van de oudere Intelligent Interface — art. nr. 30402

Eerst moet je de computer natuurlijk starten en wachten tot het besturingssysteem (Windows) volledig geladen is. De ROBO-interface mag pas na een succesvolle installatie op de computer aangesloten worden. Steek nu de installatie-CD in de CD-ROM eenheid. Het installatieprogramma op de CD wordt dan automatisch gestart.

- In het eerste welkomvenster van het installatieprogramma druk je op de knop **Volgende**.
- Het tweede venster **Belangrijke tips** bevat belangrijke actuele tips over de installatie van het programma of over het programma zelf. Ook hier druk je op de knop **Volgende**.
- In het derde venster **Licentieovereenkomsten** wordt de licentieovereenkomst voor ROBO Pro weergegeven. Je moet de licentieovereenkomst met **Ja** aanvaarden voor je met **Volgende** naar het volgende venster kan gaan.
- In het volgende venster **Gebruikersinformatie** geef je je naam in.
- In het venster **Installatietype** kan je tussen een **Standaardinstallatie** en een **Aangepaste installatie** kiezen. Bij de Aangepaste installatie kan je sommige delen van de installatie weglaten. Als je een nieuwe versie van ROBO Pro via een oudere versie installeert en enkele van de voorbeeldprogramma's van de oudere versie veranderd hebt, kan je de in de Aangepaste Installatie die door de gebruiker gedefinieerd wordt, de voorbeelden van de installatie weglaten. In het andere geval worden voorbeeldprogramma's bij de installatie **zonder waarschuwing overschreven**. Als je de Aangepaste installatie kiest en op **Volgende** drukt, verschijnt een bijkomend venster waarin je de te installeren delen kan kiezen.
- In het venster **doeldirectory** kan je de gewenste map resp. het directorypad selecteren waar het programma ROBO Pro geïnstalleerd moet worden. Normaal is dit het pad C:\Programma's\ROBO Pro. Maar je kan ook een andere directory (of bestandenlijst) ingeven.
- Als je in het laatste venster op **Voltooiën** drukt, wordt de installatie uitgevoerd. Zodra de installatie voltooid is — dat duurt normalerwijze slechts enkele seconden — meldt het programma de succesvolle installatie. Als er problemen zijn, wordt een foutmelding weergegeven die je zou moeten helpen om het probleem op te lossen.

## 1.2 Installatie van de USD-driver voor de ROBO-Interface

Deze stap is enkel noodzakelijk als de nieuwe ROBO-Interface op de USB-interface aangesloten moet worden. De ROBO-interface kan ook op de seriële interface COM1-COM4 aangesloten worden. Voor de oudere Intelligent Interface is deze stap niet noodzakelijk, aangezien de Intelligent Interface uitsluitend serieel aangesloten kan worden. De oudere Windows-versies Windows 95 en Windows NT 4.0 ondersteunen de USB-interface niet. Bij gebruik van Windows 95 of NT 4.0 kan de ROBO-interface enkel via de seriële interface aangesloten worden. Er hoeft dan ook geen driver geïnstalleerd te worden.

### ***Belangrijke tip voor de installatie onder Windows 2000 en Windows XP:***

De USB-driver kan enkel geïnstalleerd worden door een gebruiker die op de PC beheersrechten heeft. Als het installatieprogramma zou melden dat je de USB-driver niet mag installeren, moet je ofwel aan een systeembeheerder vragen om de driver te installeren of ROBO Pro zonder die driver installeren. Dan kan je je interface alleen via de iets tragere seriële interface aansluiten.

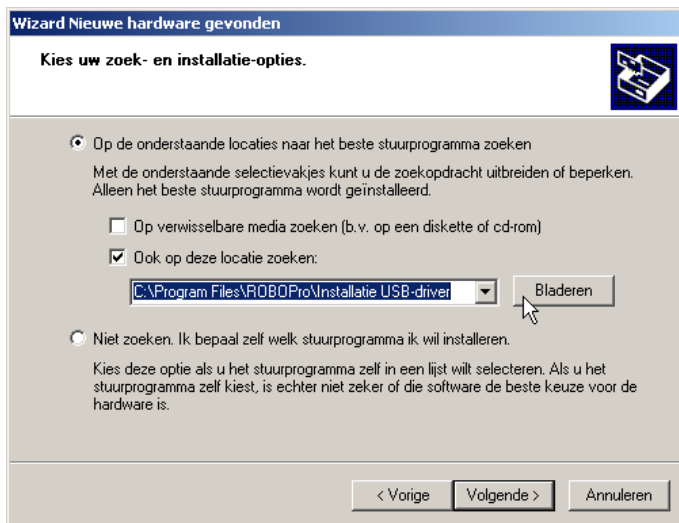
Om de USB-driver te installeren moet je eerst de ROBO-interface met een USB-kabel aan je computer aansluiten en van stroom voorzien. Windows herkent automatisch dat de interface aangesloten is en geeft het volgende venster aan:



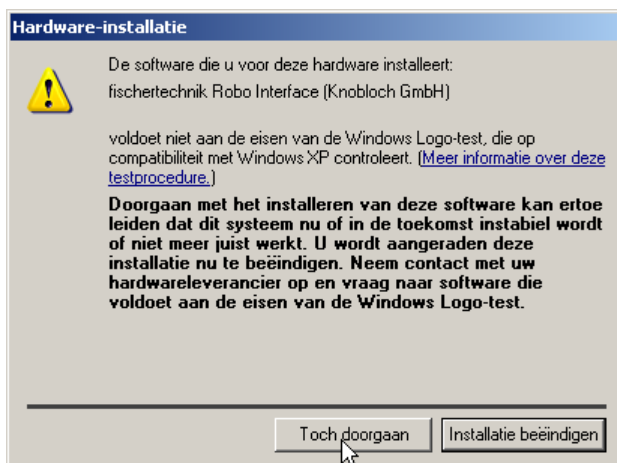
Het venster kan er volgens het besturingssysteem lichtjes anders uitzien dan hierboven afgebeeld wordt!

Hier moet je **Software** van een lijst of van een bepaalde bron installeren kiezen en op **Volgende** drukken.

In het volgende venster schakel je **Mobiele schijfenheden doorzoeken** en activeer je **Volgende bronnen eveneens doorzoeken**. Dan druk je op **Doorzoeken** en kies je de subdirectory **Installatie USB-driver** in de Directory waarin ROBO Pro geïnstalleerd werd (de standaarddirectory is C:\ROBOPro\):

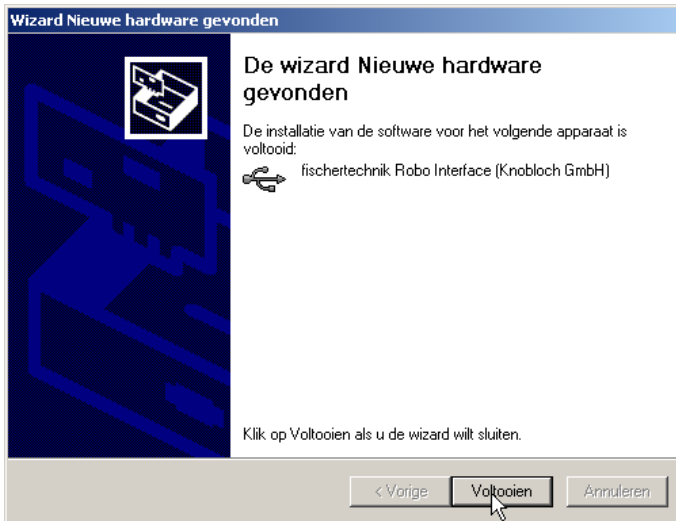


Nadat je op **Volgende** gedrukt hebt, verschijnt onder Windows XP mogelijk de volgende melding:



De USB-driver wordt nog door Microsoft gecontroleerd. Zodra de controle afgesloten is, wordt de driver door Microsoft gesigneerd, zodat deze melding niet meer verschijnt. Om de driver te installeren, op **Installatie voortzetten** klikken.

Tot slot verschijnt de volgende melding:



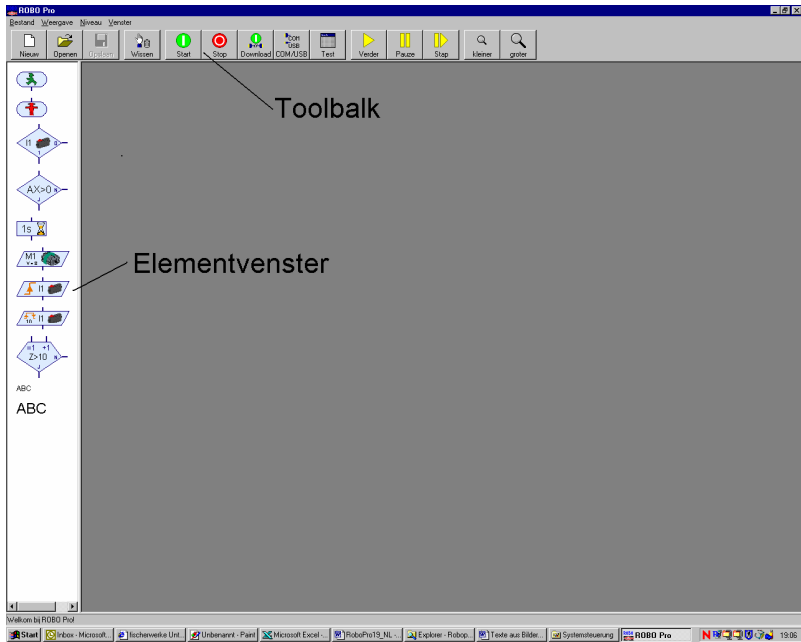
Druk op **Voltooien** om de installatie van de USB-driver af te sluiten.

### 1.3 Eerste stappen

Nieuwsgierig? Start dan toch even het programma ROBO Pro. Hiervoor klik je op de startknop in de takenbalk en kies je aansluitend **Programma** of **Alle programma's** en **ROBO Pro**. In deze map van het startmenu vind je de volgende opties:



Met de optie Verwijderen kan je ROBO Pro verwijderen. De optie Help opent het helpbestand van ROBO Pro en de optie ROBO Pro opent het ROBO Pro programma. Kies nu de optie ROBO Pro om de software te starten.

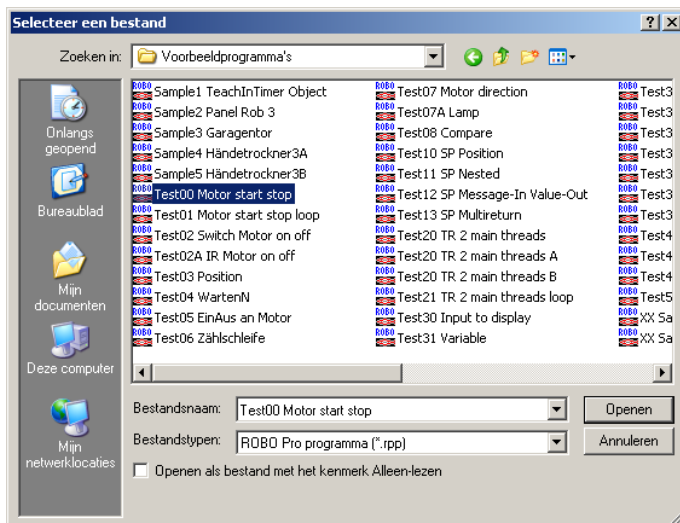


Het venster heeft boven een menubalk en een toolbalk met verschillende bedieningsknoppen en aan de linkerkant een venster met programma-elementen. Als je in de linker randkolom twee vensters boven elkaar ziet, is ROBO Pro niet op **Niveau 1** ingesteld. Om de functies van ROBO Pro aan je groeiende kennis aan te passen, kan je ROBO Pro op Niveau 1 voor beginners tot Niveau 5 voor Experts instellen. Controleer nu in het menu **Niveau**, of **Niveau 1: Beginners** aangevinkt is. Zo niet, schakel dan over naar Niveau 1.

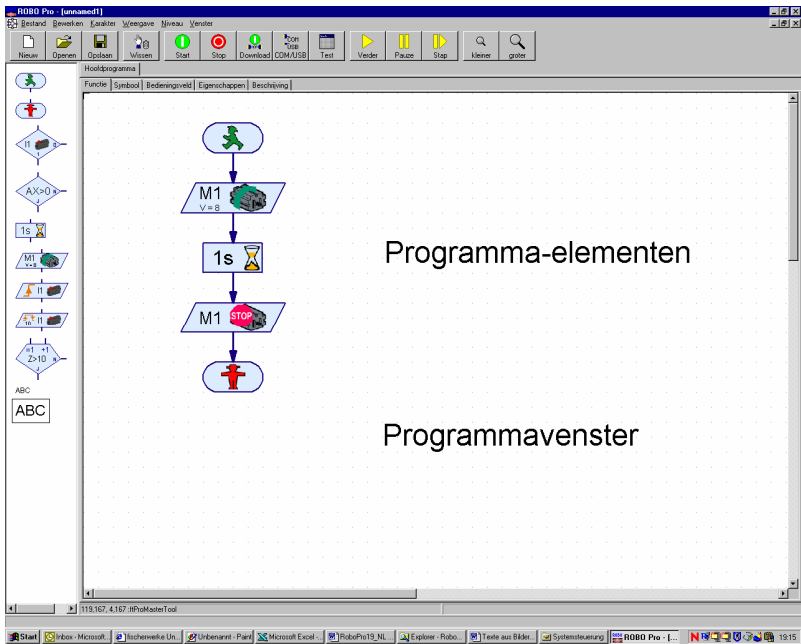




Je hebt nu de mogelijkheid om ofwel een nieuw programmabestand op te stellen, of om een reeds bestaand programmabestand te openen. We zullen pas in hoofdstuk 3 een nieuw programmabestand aanmaken, als we ons eerste besturingsprogramma gaan maken. Om het bedieningsoppervlak te leren kennen, openen we een reeds bestaand voorbeeldprogramma. Daarvoor klik je in het menu **Bestand** op de opdracht **Openen** of gebruik je de knop **Openen** in de toolbalk. De voorbeeldbestanden bevinden zich in de directory (bestandenlijst) **C:\Programma's\ROBO Pro\Voorbeeldprogramma's**.



Open het bestand **Test00 Motor start Stop.rpp**:



Hier kan je zien hoe een eenvoudig ROBO Pro programma eruit ziet. Met de programma-elementen uit het elementvenster worden bij het programmeren in het programmavenvster de verloopscema's van de besturingsprogramma's opgesteld. De afgewerkte verloopscema's kunnen dan gecontroleerd en met een aangesloten fischertechnik Interface getest worden. Maar kalm aan, we zullen in de volgende hoofdstukken stapsgewijs leren programmeren! Nadat je zo een eerste indruk van het bedieningsoppervlak gekregen hebt, sluit je het programmabestand via het bevel **Beëindigen** in het menu **Bestand** terug. De vraag of je het bestand wilt opslaan, kan je met **Nee** beantwoorden.

## 2 Voor de programmering een korte test van de hardware

Om de besturingsprogramma's die we later zullen opstellen, ook te kunnen testen, moet de interface op de PC aangesloten worden, dat is duidelijk. Maar volgens de gebruikte interface (ROBO-interface art. 93293 of de oudere Intelligent Interface art. nr. 30402) moet ook de software overeenkomstig ingesteld worden en moet de verbinding getest worden. Dit gaan we in het volgende hoofdstuk doen.

### 2.1 De interface op de PC aansluiten

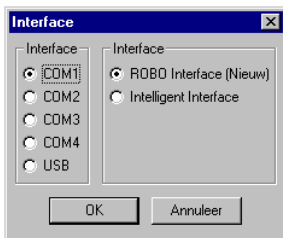
Dit zou geen groot probleem mogen zijn. De met de interface meegeleverde kabel wordt aan de interface en op een interface van de PC aangesloten:

- Bij de ROBO-Interface (art. nr. 93293) kan een USB-interface of een seriële interface COM1 tot COM4 gebruikt worden.
- Bij de Intelligent Interface (art. nr. 30402) moet een seriële interface COM1 tot COM4 gebruikt worden.

De aansluitingen van deze interfaces bevinden zich in de regel aan de achterzijde van je computer. De juiste positie van de diverse aansluitingen staat nauwkeurig beschreven in de gebruiksaanwijzing van je PC. Raadpleeg dus je gebruiksaanwijzing. USB-aansluitingen bevinden zich vaak aan de voorkant van de PC. Vergeet niet om de interface van stroom te voorzien (adapter of accu). De afzonderlijke aansluitingen van de interface worden nauwkeurig beschreven in de gebruiksaanwijzing van de betrokken interface.

### 2.2 Opdat de verbinding goed zou werken – de instellingen van de interface

Opdat de verbinding tussen de PC en de interface correct zou werken, moet de in ROBO Pro gebruikte interface ingesteld worden. Start daarvoor ROBO Pro via de optie **ROBO Pro** in het startmenu onder **Programma's** of **Alle Programma's** en **ROBO Pro**. Druk dan in de toolbalk op de knop **COM/USB**. Het volgende venster verschijnt:



Hier kan je zowel de interface als het interfacetype kiezen. De Intelligent Interface ondersteunt zoals reeds gezegd enkel de seriële interfaces COM1 – COM4.



Test

Nadat je de juiste instellingen ingebracht hebt, sluit je het venster af met OK. Open nu het venster voor de interfacetest met de Testknop op de toolbalk.

De op de interface aanwezige ingangen en uitgangen worden weergegeven. De groene balk onderaan rechts geeft de verbindingstatus tussen de PC en de interface aan:

- **Verbinding met de Interface OK** bevestigt een correcte verbinding met de interface.
- **Geen verbinding met de Interface** wijst erop dat de verbinding niet correct ingesteld werd en dat de PC geen verbinding met de interface kon opbouwen. De balk verschijnt dan in het rood.

Om de instellingen van de interface of de verbinding te kunnen wijzigen, moet je het testvenster sluiten (met de X boven rechts) en moet je zoals hiervoor beschreven via de knop COM/USB in de toolbalk een andere interface of een ander interfacetype kiezen.

Als je de verbinding tussen de PC en de interface zo hebt kunnen instellen en in het Testvenster de groene balk verschijnt, kan je het volgende hoofdstuk gerust overslaan.

Zo niet, dan kunnen de tips in het volgende hoofdstuk je wellicht voorthelpen.

## 2.3 Als de verbinding niet goed werkt - geen verbinding met de Interface?

Als bij de ROBO Interface of de Intelligent Interface ondanks het feit dat de seriële COM-interface (zie hierboven) juist ingesteld is, de boodschap **Geen verbinding met de interface** verschijnt, moet je de volgende punten controleren. Je moet hiervoor ook eventueel een "Computerspecialist" raadplegen:

- **Stroomvoorziening:**  
Wordt de interface goed van stroom voorzien? Gebruik je voor de stroomvoorziening batterijen of accu's, dan bestaat de mogelijkheid dat lege accu's niet genoeg spanning meer leveren. Daalt de spanning van de batterij onder 6 V, dan werkt de processor van de Robo Interface niet meer. In dit geval blijft de rode led ofwel branden, of anders brandt er helemaal geen diode. Als de spanning in orde is, knipperen enkele van de groene lichtdioden.. Bij de oudere Intelligent Interface kan men aan de hand van de lichtdioden niet gemakkelijk vaststellen of de spanning voor de processor volstaat. Als de spanning te laag is, moet de accu opnieuw geladen worden of moeten nieuwe batterijen gebruikt worden, of anders moet de interface met een adapter getest worden.
- **Werkt de interface wel?**  
Dit kan je achterhalen door een ander serieel apparaat zoals een externe modem op de interface te testen.
- Is er een conflict met de driver van een ander apparaat op dezelfde interface -bv. modem)? Dan moet deze driver eventueel uitgeschakeld worden (zie handboek Windows of van het apparaat).
- Enkel voor Windows NT/2000/XP en de oudere Intelligent Interface: Is bij het starten van de PC reeds een oudere Intelligent Interface met de computer verbonden en van stroom voorzien, dan wordt door Windows NT jammer genoeg naar de Download-bedrijfsmodus omgeschakeld. Om de verbinding met de PC terug tot stand te brengen, moet je gewoon de stroomvoorziening aan de interface kort onderbreken. Bij de nieuwe Robo Interface kan dat niet gebeuren.
- Als je nog steeds geen verbinding met de interface tot stand kan brengen, is de interface of de verbindingkabel waarschijnlijk defect. In dat geval moet je je tot de service van fischertechnik wenden (adres: zie menu: „?“ / Info over).

## 2.4 Werkt alles – de Interfacetest

Nadat de verbinding correct ingesteld is, kunnen we met behulp van de interfacetest de interface zelf en de daarop aangesloten modellen testen. Zoals reeds vermeld, toont het testvenster de verschillende in- en uitgangen van de interface:



Test

- **Digitale ingangen I1–I8**  
I1-I8 zijn digitale ingangen van de interface. Hier worden zogenaamde sensoren aangesloten. Digitale ingangen kunnen enkel de toestanden 0 en 1 of Ja en Neen aannemen. Aan de digitale ingangen kunnen als sensoren schakelaars (minischakelaars), maar ook fototransistoren (lichtsensoren) of reedcontacten (magneetsensoren) aangesloten worden.
- Je kan de functie van deze ingangen controleren door aan de interface bv. een minischakelaar (art. nr. 37783) aan te sluiten (gebruik op de schakelaar de contacten 1 en 3). Zodra je op de schakelaar drukt, verschijnt op het scherm van I1 een haakje. Heb je de

schakelaar andersom aangesloten (contacten 1 en 2), dan verschijnt het haakje meteen en verdwijnt het door de schakelaar in te drukken.

- **Motoruitgangen M1–M4**

M1 – M4 zijn de uitgangen van de interface. Hier worden de zogenaamde actoren aangesloten. Dit kunnen bv. motoren, elektromagneten of lampen zijn. De 4 motoruitgangen kunnen zowel in snelheid in 8 trappen als in richting gestuurd worden. De schuifregelaar dient om de snelheid te regelen. Bovendien wordt de snelheid naast de schuifregelaar als getal aangeduid. Als je een uitgang wilt testen, sluit je aan deze uitgang, bv. M1, een motor aan.

- **Lampuitgangen O1–O8**

De motoruitgangen kunnen alternatief ook als een paar van afzonderlijke uitgangen gebruikt worden. Daarmee kan men lampen, maar ook motoren die maar in één richting moeten bewegen (bv. een transportband) aansturen. Als je deze uitgangen wilt testen, sluit je een aansluiting van de lamp op de uitgang aan, bv. B. O1. De andere aansluiting van de lamp verbind je met de massabus van de interface (⊥).

- **Analoge ingangen AX–AY**

De analoge ingangen AX en AY meten de weerstand van de aangesloten sensor. Hier kunnen NTC–weerstanden voor de temperatuurmeting, potentiometers, fotoweerstanden of fototransistoren aangesloten worden.

- **Analoge ingangen A1–A2**

Deze beide ingangen meten een spanning van 0–10V.

- **Afstandssensoren D1–D2**

Aan de ingangen van de afstandssensoren D1 en D2 kunnen enkele speciale afstandssensoren aangesloten worden. Voor de afstandssensoren D1 en D2 zijn er zowel digitale ingangen als analoge ingangen voorhanden.

- **Uitbreidingsmodule EM1–EM3**

Voor zover op de interface uitbreidingsmodules aangesloten zijn (tot drie I/O-uitbreidingen art. nr. 93294 bij de ROBO Interface, maar hoogstens een uitbreidingsmodule art. nr. 16554 bij de Intelligent Interface), kunnen deze gestuurd worden door via het register op de bovenste rand op de uitbreidingsmodule om te schakelen.

### 3 Niveau 1: Je eerste besturingsprogramma

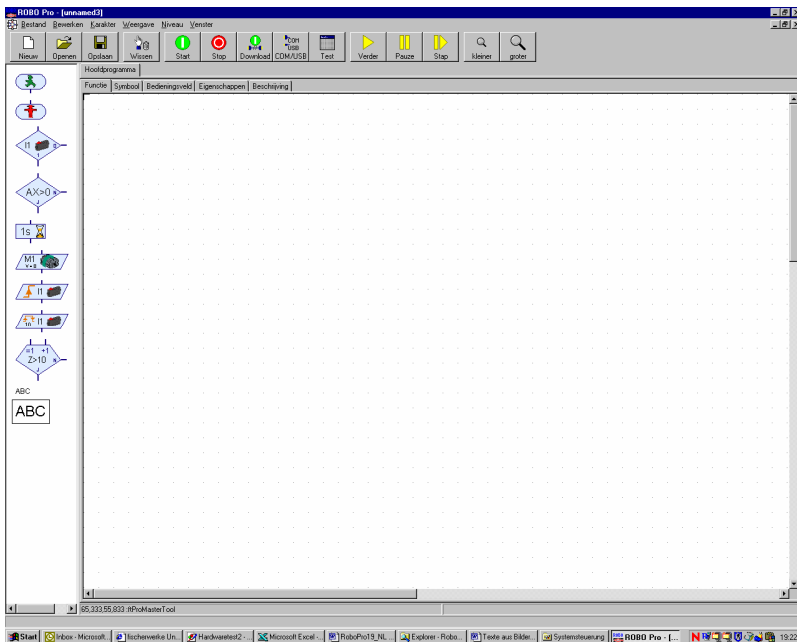
Nadat je in het laatste hoofdstuk de hardware, dus de interface en de daarop aangesloten schakelaars en motoren, getest hebt, zullen we ons nu bezighouden met het programmeren. Maar wat betekent "programmeren" eigenlijk? Stel je nu eens voor dat aan onze interface bv. een robot aangesloten is. Maar deze robot is zo dom dat hij alleen niet kan werken. Gelukkig zijn wij een beetje slimmer. We kunnen de robot precies zeggen wat hij moet doen. Hoe? Wel, wat gebeurde er toen we in het laatste hoofdstuk met de muistoets de motoruitgang M1 op „Links“ ingesteld hebben? Juist, we hebben de motor uitgeschakeld. Als deze motor bv. de grijptang van onze robot zou bewegen, zouden we niets anders gedaan hebben dan tegen de robot gezegd hebben: "Grijp het voorwerp!" Maar we willen niet elke stap manueel regelen. We willen dat de robot dit "automatisch" doet. Daarvoor moeten we de afzonderlijk uit te voeren stappen zodanig opslaan dat de robot ze na elkaar kan afwerken, d.w.z. we moeten een programma opstellen dat de robot in onze plaats stuurt. In de vaktaal noemt men dat dan logischerwijze een besturingsprogramma.

#### 3.1 Een nieuw programma maken

Met de software ROBO Pro hebben we nu een fantastisch werktuig in de hand om dergelijke besturingsprogramma's te ontwerpen en met behulp van een aangesloten interface te testen. Geen angst, we gaan niet meteen een robot programmeren. We nemen voorlopig genoegen met eenvoudige stuurtaken. Hiervoor moeten we een nieuw programma maken. In de toolbalk vind je de optie "Nieuw". Als je daarop klikt met de linker muisknop, wordt een nieuw leeg programma opge maakt:



Nieuw



Je ziet nu een groot wit tekenvlak waarin je zo meteen je eerste programma gaat ingeven. Als je aan de linkerrand twee vensters boven elkaar ziet, stel dan het menu **Niveau** in op **Niveau 1: Beginners**.

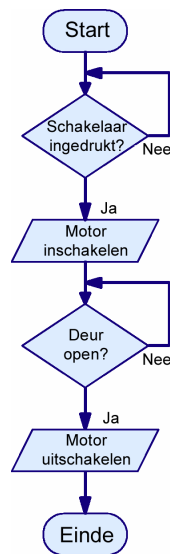
## 3.2 De elementen van het besturingsprogramma

Nu kunnen we aan het werk gaan en ons eerste besturingsprogramma maken. Dit gaan we doen aan de hand van een concreet voorbeeld:

### Beschrijving van de functie:

Stel je een garagepoort voor die automatisch geopend kan worden. Misschien hebben jullie er thuis wel zo een! Als men met de auto aan de garage komt, volstaat het om op de zender op een knop te drukken en de garagepoort wordt, aangedreven door een motor, geopend. De motor moet blijven lopen tot de garagepoort volledig open is.

Nu is het erg omslachtig en ook niet erg overzichtelijk om een besturing met woorden te beschrijven. Daarom gebruikt men voor de voorstelling van na elkaar uit te voeren acties en voorwaarden die voor deze acties vervuld moeten zijn zogenaamde **stroomschema's**. De voorwaarde voor de actie "Inschakelen van de motor" is in het geval van onze besturing dat de schakelaar ingedrukt wordt. Het lezen van een dergelijk stroomschema is heel eenvoudig. Steeds stapsgewijs de pijlen volgen! Deze geven dan de juiste werkwijze van de besturing - de afzonderlijke stappen kunnen enkel in de door de pijlen aangegeven volgorde uitgevoerd worden, nooit anders. Anders hoefden we al dat werk niet te doen - of wel?



Met behulp van onze software ROBO Pro kunnen we dit stroomschema precies tekenen en daarmee het **besturingsprogramma** voor de aangesloten hardware (interface, motoren, schakelaars enz.) opstellen. De rest doet de software, wat trouwens bij grote, industriële toepassingen ook niet anders is! Daarmee kunnen we ons volledig op de opstelling van het stroomschema concentreren.

Het stroomschema vorm je op basis van programma-elementen. Weer een nieuw begrip? Dat is niet zo erg! In ROBO Pro worden de afzonderlijke elementen waarmee het stroomschema opgesteld wordt, programma-elementen genoemd. De actie "Motor inschakelen" houdt immers niets anders in dan dat de interface de motor, die op de interface aangesloten is, echt moet inschakelen. De beschikbare programma-elementen vind je in het elementvenster aan de linkerkant.



## 3.3 Programma-elementen invoegen, verschuiven en veranderen

Nu gaat het erom met de programma-elementen uit het elementvenster het stroomschema voor onze besturing van de garagepoort op te stellen. Alle beschikbare programma-elementen kunnen uit het elementvenster gehaald worden en kunnen in het programmavenster ingevoegd worden.

### Invoegen van programma-elementen

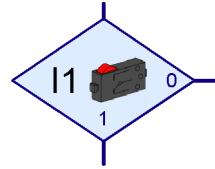
Daartoe beweeg je de muis over het symbool van het gewenste programma-element en klik je er met de linker muistoets een keer op. Dan beweeg je de muis in het programmavenster (dat is het grote witte vlak) op de gewenste plaats en klik je nogmaals. Je kan het programma-element ook met een ingedrukte





muistoets uit het elementvenster in het programmavenster trekken. Een programma begint steeds met een startelement. Het startelement is het afgeronde element met het groene verkeersmannetje. Je probeert het best meteen eens uit met dit programma-element: klik een keer met de linker muistoets op het startelement in het elementvenster, beweeg de muis bovenaan in het programmavenster en klik daar nogmaals op de linker muistoets.

Als volgende komt in het stroomschema een element dat een ingang ondervraagt en volgens de toestand van de ingang in een of ander programmapad vertakt. Klik in het elementvenster op het rechts afgebeelde element en beweeg dan de muis onder het startelement dat je daarnet ingevoegd hebt. Als de bovenste ingang van het vertakkingselement een of twee rasterpunten onder de uitgang van het startelement staat, verschijnt in het programmavenster een verbindinglijn. Als je nogmaals op de linker muisknop klikt, wordt het vertakkingselement ingevoegd en wordt het automatisch met het startelement verbonden.



### ***Verschuiven van programma-elementen en van groepen***

Een programma-element kan ook na het invoegen bij een ingedrukte linker muistoets naar de gewenste positie verschoven worden. Als je meerdere elementen samen wilt verschuiven, kan je eerst met een ingedrukte linker muistoets een kader rond de elementen optrekken. Je moet daarvoor met de linker muistoets in een **lege** zone klikken, de toets ingedrukt houden en met de muis een rechthoek optrekken die de gewenste elementen bevat. De elementen in de rechthoeken worden nu met een rode rand weergegeven. Als je nu een van de rode elementen met de linker muistoets verschuift, worden alle rode elementen mee verschoven. Je kan ook afzonderlijke elementen rood markeren door met de linker muistoets bij een ingedrukte shifttoets (dat is de hoofdlettertoets) op de elementen te klikken. Als je met de linker muistoets in een lege zone klikt, worden alle rood gemarkeerde elementen terug normaal weergegeven.

### ***Kopiëren van programma-elementen en van groepen***

Voor het kopiëren van programma-elementen zijn er twee mogelijkheden. Je kan hetzelfde doen als voor het verschuiven, maar voor je de elementen verschuift, druk je de Control-toets op het toetsenbord in. Daardoor worden de elementen niet verschoven, maar gekopieerd. Met deze functie kan je de elementen echter alleen binnen een programma kopiëren. Als je elementen van het ene programma naar een ander wilt kopiëren, kan je het **klembord** van Windows gebruiken. Kies eerst enkele elementen uit, zoals dat in het vorige hoofdstuk beschreven werd bij het verschuiven van elementen. Als je nu **Ctrl+C** op het toetsenbord indrukt, of de menupunten **Bewerken / Kopiëren** oproept, worden alle geselecteerde elementen naar het klembord van Windows gekopieerd. Je kan nu overgaan naar een ander programma en de elementen daar met **CTRL+V** of **Bewerken / Invoegen** terug invoegen. Je kan eenmaal gekopieerde elementen ook meerdere keren invoegen. Als je elementen van het ene programma in een ander programma wilt verschuiven, kan je in het begin in plaats van **CTRL+C** of **Bewerken / Kopiëren** de functie **CTRL+X** of **Bewerken / Knippen** gebruiken.

### ***Wissen van elementen en Functie Ongedaan maken***

Het wissen van elementen is eveneens heel eenvoudig. Je kan alle rood gemarkeerde elementen (zie vorig hoofdstuk) wissen door de wistoets (**Delete**) op het toetsenbord in te drukken. Je kan ook afzonderlijke elementen met de wistoets wissen. Klik daarvoor eerst op de afgebeelde knop in de toolbalk en dan op het element dat je wilt wissen. Probeer het meteen maar eens uit. Je kan het gewiste element daarna opnieuw tekenen. Om het gewiste element terug te halen, kan je ook de



wissen

functie **Ongedaan maken** in het menu **Bewerken** gebruiken. Via dit menupunt kan je alle wijzigingen aan het programma ongedaan maken.

## Eigenschappen van programma-elementen bewerken

Als je met de **rechter** muistoets op een programma-element in het programmavenster klikt, verschijnt een dialoogvenster waarin je de eigenschappen van het element kan veranderen. Het venster Eigenschappen voor een vertakkingselement is rechts afgebeeld.

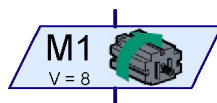
- Met de knoppen I1 tot I8 kan je ingeven welke ingang van de interface opgevraagd moet worden.
- De selectie **Interface / Uitbreiding** wordt pas in hoofdstuk 6 *Uitbreidingsmodule en meerdere interfaces aansturen* op pagina 51 uitgelegd.
- Onder **Beeld** kan je een beeld voor de aan de ingang aangesloten sensor selecteren. Digitale ingangen worden meestal met schakelaars gebruikt, maar ook vaak met fototransistoren of reedcontacten.
- Onder **1/0 Aansluitingen verwisselen** kan je de positie van de 1 en 0 uitgangen van de vertakking verwisselen. Normalerweise staat de 1 uitgang onderaan en de 0 uitgang rechts. Maar vaak is het praktischer als de 1 uitgang rechts staat. Druk op **1/0 aansluitingen verwisselen**, dan worden de 1 en 0 aansluitingen verwisseld zodra je het venster met OK sluit.



**Tip:** als je een minischakelaar als sluiters op aansluiting 1 en 3 van de schakelaar aansluit, gaat de programmavertakking naar tak 1 als de schakelaar ingedrukt is en anders naar tak 0.

Als je een minischakelaar als openaar op de aansluiting 1 en 2 van de schakelaar aansluit, gaat de programmavertakking naar tak 1 als de schakelaar niet ingedrukt is en anders naar tak 0.

Het volgende programma-element in onze besturing van de garagepoort is een motorelement. Voeg het zoals de twee vorige elementen in het programma in en weliswaar onder het vertakkingselement. Je kan het element best terug zo plaatsen dat het automatisch met het element daarboven verbonden wordt.



Met het motorelement kan je zowel een motor als een lamp of een elektromagneet aan- of uitschakelen. Het venster Eigenschappen voor het motorelement open je terug met een rechter muisklik op het element.

- Via de knoppen **M1** tot **M4** kan je kiezen welke uitgang van de interface aangestuurd moet worden.
- Onder **Beeld** kan je een beeld kiezen dat de op de uitgang aangesloten fischertechnik bouwsteen voorstelt.
- De selectie **Interface / Uitbreiding** wordt pas in hoofdstuk 6 *Uitbreidingsmodule en meerdere interfaces aansturen* op pagina 51 uitgelegd
- Onder **Actie** kan je kiezen hoe de uitgang beïnvloed moet worden. Een motor kan je met draairichting links of rechts starten of stoppen. Een lamp kan je aandoen of uitschakelen.
- Onder **Snelheid/Intensiteit** kan je instellen met welke snelheid de motor moet draaien, bv. hoe licht de lamp moet zijn. Je kan de waarden instellen van 1 tot 8.

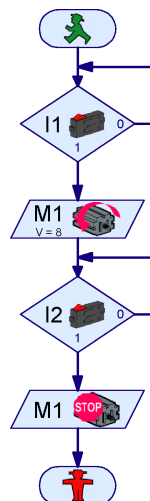
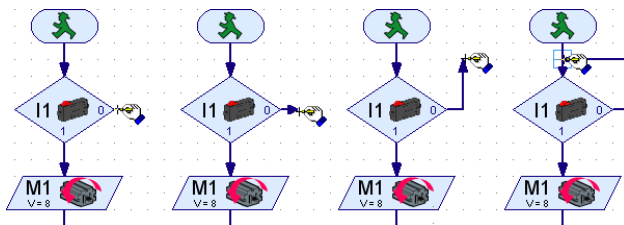


Voor ons stroomschema hebben we het bevel **Motor M1 links met snelheid 8** nodig.

### 3.4 Verbinden van de programma-elementen

Nu je weet hoe men elementen in een besturingsprogramma invoegt, kunnen we ons besturingsprogramma afwerken. Denk eens terug aan de functiebeschrijving van de besturing voor de garagepoort - ontbreekt daar niets meer? Juist, we hebben weliswaar de motor met een toetsdruk ingeschakeld, maar als de poort open is, moet die ook automatisch terug uitgeschakeld worden! In de praktijk gebeurt dit met een zogenaamde eindschakelaar. Dat is een schakelaar die zodanig op de garagepoort aangebracht is dat hij bediend wordt op het moment dat de motor de poort volledig geopend heeft. En zoals bij het uitschakelen van de motor kan dit signaal gebruikt worden om de motor terug uit te schakelen. Voor de ondervraging van de eindschakelaar kunnen we terug het vertakkingselement gebruiken.

Voeg dus in je besturingsprogramma nog een vertakkingselement in dat de eindschakelaar aan de Ingang I2 ondervraagt. Vergeet niet met de rechter muistoets op het element de klikken en de ingang naar I2 om te schakelen. Zodra de garagepoort open is en de eindschakelaar ingedrukt is, moet de motor terug stoppen. Dit wordt via een motorelement bereikt. Gebruik eerst hetzelfde element als voor het inschakelen van de motor. Als je met de rechter muistoets op het element klikt, kan je de functie van het element veranderen in **Motor stoppen**. Het programma wordt afgesloten door een eindelement. Het programma moet er nu bijna uitzien zoals rechts afgebeeld is. Als je de elementen steeds met een afstand van een of twee rasterpunten vlak onder elkaar geplaatst hebt, zijn de meeste in- en uitgangen reeds door programmapijlpunten met elkaar verbonden. De Neen-uitgang (N) van de beide vertakkingen is echter nog niet aangesloten. Zolang de schakelaar aan de ingang I1 niet ingedrukt is, moet het programma teruggaan en de schakelaar nogmaals ondervragen. Om deze lijn te tekenen, klik je met de muis na elkaar op de in de afbeelding aangeduide plaatsen.



**Tip:** Als een lijn eens niet correct met een aansluiting of een andere lijn verbonden zou zijn, wordt dit door een groene rechthoek aan de pijlpunt aangeduid. In dat geval moet je de verbinding herstellen door de lijn te verschuiven of door ze te wissen en opnieuw te tekenen. Anders werkt het stroomschema op die plaats niet.

## Wissen van programmastroomlijnen

Het wissen van lijnen verloopt precies zoals het wissen van programma-elementen. Klik gewoon met de linker muistoets op de lijn, zodat ze rood gemarkeerd wordt. Druk nu op de wistoets (**Wis**) op het toetsenbord om de lijn te wissen. Je kan ook meerdere lijnen uitkiezen als je de shift-toets (dat is de toets om heen en weer te schakelen tussen hoofdletters en kleine letters) ingedrukt houdt en dan na elkaar met de linker muistoets op de lijn klikt. Bovendien kan je voor het markeren van meerdere lijnen ook een kader rond deze lijnen trekken. Nu kan je alle rood gemarkeerde lijnen ineens wissen door op de **Wis**-toets te drukken.

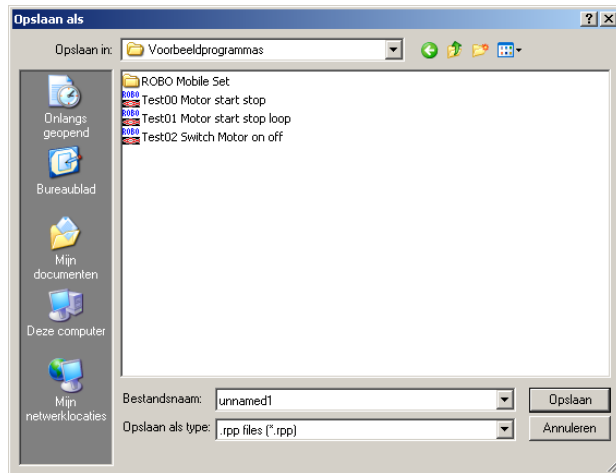
## 3.5 Testen van het eerste besturingsprogramma

Om ons eerste besturingsprogramma te kunnen testen, moet je een klein model opbouwen. Daarvoor volstaat het om op de interface op I1 en I2 een schakelaar en op M1 een motor aan te sluiten.

**Tip:** De aansluiting van de interface op de PC en de instelling van de interface is reeds in het vorig hoofdstuk gebeurd en kan daar geraadpleegd worden.

Voor je het besturingsprogramma test, moet je het programmabestand op de harde schijf van je computer opslaan. Klik met de muis op het bevel **Opslaan onder** in het menu **Bestand**. Daarna verschijnt het volgende dialoogvenster:

Kies bij "Opslaan als" een bestandenlijst waarin je het programma wilt opslaan. Geef bij "Bestandsnaam" een nog niet bestaande naam in, bv. GARAGEPOORT en bevestig met een linker muisklik op "Opslaan".



Om het programma te testen, druk je op de links getoonde startknop in de toolbalk. Eerst test ROBOPro of alle programma-elementen juist verbonden zijn. Als een element niet juist verbonden is, of als iets anders niet in orde zou zijn, wordt het rood gemarkeerd en wordt een foutmelding gegeven die beschrijft wat er niet in orde is. Als je bv. vergeten bent om de Neen-uitgang (N) van de programmavertakking aan te sluiten, ziet het er zo uit:

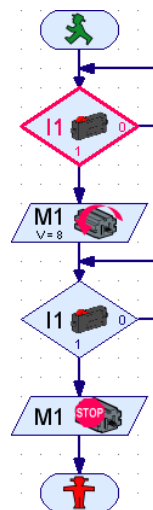


Als je een foutmelding gekregen hebt, moet je eerst de gemelde fout verbeteren. In het andere geval wordt het programma niet gestart.

**Tip:** Een uitvoerige uitleg over deze bedrijfsmodus en over de bedrijfsmodus „Download-bedrijfsmodus“ vind je in hoofdstuk 3.7 op pagina 24.

Het eerste vertakkingselement wordt rood gemarkeerd. Er wordt aangegeven dat het stroomschema in dit programma-element op een gebeurtenis wacht, namelijk dat de schakelaar aan I1, die de garagepoort moet openen, ingedrukt wordt. Zolang de schakelaar niet ingedrukt wordt, vertakt het programma naar de Neen-uitgang (N) van de programmavertakking en gaat van daar terug naar het begin van de vertakking. Druk nu de schakelaar in die aangesloten is op de I1-ingang van de interface. Daarmee is de voorwaarde voor de doorschakeling vervuld en wordt de motor ingeschakeld. Het stroomschema wacht er in de volgende stap op dat de eindschakelaar aan de ingang I2 ingedrukt wordt. Zodra je de eindschakelaar op I2 activeert, vertakt het programma naar de tweede motoruitgang en schakelt de motor terug uit. Tot slot bereikt het programma het programma-einde. Er verschijnt een melding dat het programma beëindigd werd.

Is het allemaal gelukt? Hartelijk gefeliciteerd! Daarmee heb je je eerste besturingsprogramma opgesteld en getest. Als het niet werkt - de moed niet verliezen. Gewoon alles nog eens nakijken. Er zit waarschijnlijk ergens maar een klein foutje verstopt. Elke programmeur maakt wel eens een fout en uit fouten leert men meestal. Goede moed dus!!

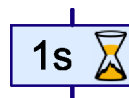


### 3.6 Verdere programma-elementen

Als je je eerste besturingsprogramma op een echt model van een garagepoort uitgeprobeerd hebt, staat de poort nu open. Maar hoe wordt die nu terug gesloten? Natuurlijk kunnen we de motor terug starten door het indrukken van een schakelaar! Maar we gaan eerst een andere oplossing testen en gaan daarbij kennismaken met een nieuw programma-element. Hiervoor bewaar je het programma eerst onder een nieuwe naam (het huidige stroomschema hebben we later nog eens nodig). Gebruik daarvoor het menupunt **Opslaan onder ...** in het menu **Bestand** en geef daar een nog niet gebruikte bestandsnaam in.

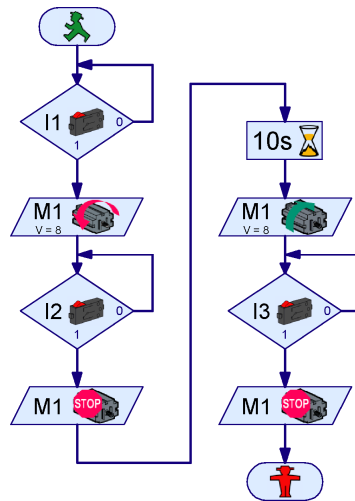
#### 3.6.1 Wachtijd

Voor we het stroomschema verder kunnen uitwerken, moet je de verbinding tussen „Motor uitschakelen“ en „Programma-einde“ wissen en het indelement naar beneden verschuiven. De nieuwe programma-elementen kan je nu tussen deze beide elementen invoegen. De garagepoort moet na een tijd van 10 seconden automatisch gesloten worden. Hiervoor kan je het rechts afgebeelde programma-element **Wachtijd** gebruiken. De wachtijd kan je grotendeels naar wens instellen door zoals gebruikelijk met de rechter muistoets op het element te klikken. Geef de gewenste wachtijd van 10 seconden in. Voor het sluiten van de garagepoort moet de motor natuurlijk in de andere richting, dus naar rechts, lopen. De motor wordt door een verdere eindschakelaar op I3 uitgeschakeld.

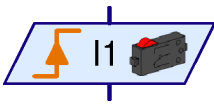




Je stroomschema is klaar en moet er uitzien zoals rechts afgebeeld wordt. De nieuwe programma-elementen zijn voor een betere weergave naar rechts verschoven. Als er in het stroomschema geen fouten staan, kan je de uitgebreide besturing van de garagepoort zoals gewoonlijk testen met de **Start** - knop. Bij de activering van de schakelaar aan I1 wordt de motor ingeschakeld. Bij activering van I2 wordt hij terug uitgeschakeld. Daarmee is de garagepoort geopend. Nu wordt het programma-element wachttijd voor 10 seconden - dat is onze ingestelde wachttijd - met een rode rand omgeven. Dan wordt de motor met een andere draairichting ingeschakeld tot de toets aan I3 geactiveerd wordt. Probeer ook eens om de wachttijd te veranderen.



### 3.6.2 Wachten op ingang



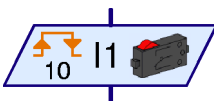
Naast het element Wachttijd zijn er nog twee andere elementen die op iets wachten tot ze het programma verder uitvoeren. Het links afgebeeld element **Wachten op ingang** wacht tot een ingang van de interface een bepaalde toestand heeft, of op een bepaalde manier verandert. Van dit element zijn er 5 varianten:

| Symbol                                      |                        |                    |                                       |                                       |  |
|---|------------------------|--------------------|---------------------------------------|---------------------------------------|--|
| Wachten op                                  | Ingang=1<br>(gesloten) | Ingang=0<br>(open) | wissel 0-1<br>(open naar<br>gesloten) | wissel 1-0<br>(gesloten naar<br>open) | Willekeurige<br>wissel<br>(1-0 of 0-1) |
| Zelfde<br>functie maar<br>met<br>vertakking |                        |                    |                                       |                                       |  |



Men kan daarvoor ook een combinatie van vertakkingselementen gebruiken, maar met het element **Wachten op ingang** gaat het gemakkelijker en is het overzichtelijker.

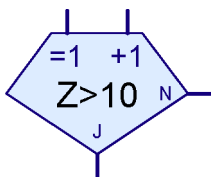
### 3.6.3 Impulsteller



Veel Fischertechnik robotmodellen gebruiken ook impulstandwielen. Deze tandwielen activeren een schakelaar bij elke omwenteling vier keer. Met dergelijke impulstandwielen kan men een motor in plaats van gedurende een bepaalde tijd, voor een precies vastgelegd aantal omwentelingen inschakelen. Daarvoor moet men het aantal impulsen aan een ingang van de interface tellen. Daarvoor kan je het links afgebeelde **Impulstellerelement** gebruiken dat op een instelbaar aantal impulsen wacht. Ook bij dit element kan je instellen of een

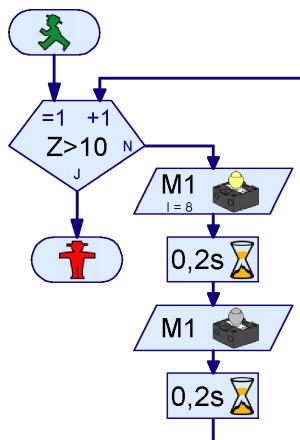
bepaald aantal wijzigingen of enkel de overgang 0-1 of 1-0 als impuls beschouwd moet worden. Bij impulstandwielen wacht men in de regel op willekeurige wijzigingen, zodat men bij 4 tanden een oplossing van 8 stappen per omwenteling bereikt.

### 3.6.4 Tellus



Met het telluselement kan je heel eenvoudig een bepaald programmadeel meerdere keren laten uitvoeren. Het afgebeelde programma schakelt bijvoorbeeld een lamp aan M1 10x aan en terug uit. Het telluselement heeft een ingebouwde teller. Als de

tellus via de =1 ingang betreden wordt, wordt de teller op 1 gezet. Als de tellus daartegen via de +1 ingang betreden wordt, wordt de teller 1 erbij geteld. Naargelang de teller al dan niet groter is dan de door jou ingegeven waarde, vertakt de tellus naar de uitgang Ja ( J ) of Neen ( N ). De Ja-uitgang wordt zo gebruikt dat de lus zo vaak doorlopen werd als je in de getalwaarde aangegeven hebt. Als nog andere lusdoorgangen noodzakelijk zijn, vertakt de tellus daarentegen naar de Neen-uitgang. Zoals bij het vertakkingselement kan je de Ja- en Neen-uitgang via het venster Eigenschappen ook verwisselen.



## 3.7 Online- of Download-bedrijfsmodus – Wat is precies het verschil?



Start

Tot nu hebben we onze besturingsprogramma's in de zogenaamde **Online-bedrijfsmodus** getest. Je kon daarbij het verloop van de programma's op het beeldscherm mee volgen omdat het actieve element op het beeldscherm telkens in het rood gemarkeerd werd. De online-bedrijfsmodus gebruik je om programma's te verstaan of om fouten in het programma op te zoeken.



Pauze

In de Online-bedrijfsmodus kan je het programma ook stoppen en terug voortzetten door op de knop **Pauze** te drukken. Dat is zeer praktisch als je bij je model iets wilt onderzoeken zonder het programma volledig te stoppen. Ook als je probeert het verloop van een programma te begrijpen, kan de pauzefunctie erg nuttig zijn.



Stap

Met de **Stap**-knop kan je het programma in afzonderlijke stappen element per element uitvoeren. Telkens als de Stap-knop ingedrukt wordt, gaat het programma naar het volgend programma-element. Als je een **Wachtijd** element of een **Wachten op** element uitvoert, kan het natuurlijk een tijdje duren tot het programma bij het volgende element aankomt.



Download

Als je een Robo Interface (geen Intelligent Interface) hebt, kan je in de plaats van de online-bedrijfsmodus ook de **Download-bedrijfsmodus** gebruiken. In de Online-bedrijfsmodus worden de programma's door je computer uitgevoerd. Je computer stuurt daarbij stuurbevelen zoals "Motor inschakelen" naar de interface. Daarvoor is het noodzakelijk dat de interface met de computer verbonden is, zolang het programma loopt. In de Download-bedrijfsmodus wordt het programma daarentegen door de interface zelf uitgevoerd. Je computer slaat het programma in de interface op. Zodra dat gebeurd is, kan de verbinding tussen computer en interface verbroken worden. Nu kan de interface het besturingsprogramma onafhankelijk van de computer uitvoeren. Belangrijk is de Download-bedrijfsmodus bv. bij de programmering van mobiele robots waarbij



een verbinding kabel tussen de PC en de robot zeer hinderlijk zou zijn. Maar toch moeten besturingsprogramma's eerst in de Online-bedrijfsmodus getest worden, aangezien mogelijke fouten zo beter opgespoord kunnen worden. Het uitgeteste programma kan dan per download op de Robo Interface overgedragen worden. Bij de Robo Interface kan de storende kabel door de radioverbinding **ROBO RF datakoppeling**, art. nr. 93295 vervangen worden. Daarmee is het model ook in de Online-bedrijfsmodus onbeperkt mobiel.

De Online-bedrijfsmodus heeft tegenover de Download-bedrijfsmodus evenwel ook voordelen. Een computer heeft in vergelijking met de interface veel meer werkgeheugen en kan veel sneller rekenen. Bij grote programma's is dit een voordeel. Bovendien kunnen in de online-bedrijfsmodus meerdere interfaces parallel aangestuurd worden, ook Robo Interfaces en Intelligent Interfaces gemengd.

De twee bedrijfsmodi in een overzicht

| Bedrijfsmodus | Voordeel  | Nadeel  |
|---------------|---|---|
| Online        | <ul style="list-style-type: none"> <li>• De uitvoering van het programma kan op het beeldscherm gevolgd worden</li> <li>• De uitvoering van grotere programma's is ook erg snel</li> <li>• Er kunnen meerdere interfaces parallel aangestuurd worden</li> <li>• De oudere Intelligent Interface wordt ondersteund</li> <li>• Er kunnen bedieningsvelden gebruikt worden</li> <li>• Het programma kan gestopt en voortgezet worden.</li> </ul> | <ul style="list-style-type: none"> <li>• Computer en Interface moeten met elkaar verbonden blijven</li> </ul>   |
| Download      | <ul style="list-style-type: none"> <li>• Computer en Interface kunnen na de Download van elkaar gescheiden worden</li> </ul>  | <ul style="list-style-type: none"> <li>• De oudere Intelligent Interface wordt niet ondersteund</li> <li>• De programma-uitvoering kan niet op het beeldscherm gevolgd worden.</li> <li>• Het programma kan max 3 uitbreidingsmodules aansturen.</li> </ul> |



## De Download-bedrijfsmodus gebruiken

Als je dus de nieuwe ROBO Interface hebt, kan je je besturing van je garagepoort met de **Download**-knop in je toolbalk naar de interface overdragen. Eerst wordt het rechts getoonde dialoogvenster getoond. De Robo Interface beschikt over meerdere programmeergeheugenplaatsen, een **RAM** (Random Access Memory) zone en twee **Flash** zones. Een programma in het RAM gaat verloren zodra je de interface van de stroomvoorziening scheidt of wanneer de accu leeg is. Een in Flash opgeslagen programma blijft daarentegen ook zonder stroom jarenlang op de interface opgeslagen. Natuurlijk kan je programma's in Flash op gelijk welk moment overschrijven. De Download in het RAM gaat echter duidelijk sneller en wordt daarvoor voor testdoeleinden aangeraden.



In de twee Flashzones kan je twee verschillende programma's, bijvoorbeeld twee verschillende gedragswijzen voor een mobiele robot, opslaan. De twee programma's kan je dan met de **Prog**-toets op de interface selecteren, starten en stoppen. Als de optie **Programma na download starten** geactiveerd is, zal het programma na de download onmiddellijk gestart worden. Terwijl het programma loopt, knippert op de Robo Interface de groene LED **Prog 1** (Programma in Flash 1 geladen) of **Prog 2** (Programma in Flash 2 geladen) naast de **Prog**-toets. Wordt een programma in het RAM geladen, dan knipperen de beide LEDs. Om het programma te stoppen, druk je op de **Prog**-toets. Dan brandt de LED continu. Om van programma 1 naar programma 2 over te schakelen, houd je de prog-toets ingedrukt tot de LED voor het gewenste programma 1 of 2 oplicht. Om het programma te starten, druk je de toets opnieuw in.

Bij mobiele robotten is de optie **Programma's via schakelaar op de interface starten** zinvoller. Als je geen RF-datakoppeling hebt, moet je namelijk nog de kabel losmaken voor je programma de robot in beweging zet. In dat geval moet je eerst via de prog-toets op de interface het gewenste programma kiezen en dit starten door nogmaals op de toets te drukken.

Als de laatste optie **Bij het inschakelen automatisch starten** geactiveerd is, wordt het programma 1 in Flash automatisch gestart, zodra de interface van stroom voorzien wordt. Daarmee kan je bv. je interface via een adapter met schakelklok van stroom voorzien en het programma elke dag op hetzelfde uur starten. Dan moet de interface niet voortdurend ingeschakeld blijven en moet ook het programma niet iedere keer na het inschakelen via de prog-toets gestart worden.

### Aanwijzingen:

Als een programma in het Flash geladen wordt of vanuit het Flash-geheugen uitgevoerd wordt, gaan in de RAM geladen programma's verloren, omdat Flash-programma's eveneens gebruik maken van het RAM-geheugen.

Je vindt ook een uitvoerige beschrijving van de functies van de ROBO-interface in de gebruiksaanwijzing van de interface.

### 3.8 Tips en tricks

#### Verbindingslijnen veranderen

Als je elementen verschuift, doet ROBO Pro zijn best om de verbindingenlijnen verstandig aan te passen. Als een aangepaste lijn je eens niet bevalt, kan je de verbindingenlijnen gemakkelijk veranderen door met de linker muistoets op de lijn te klikken en kan je de lijn met een ingedrukte muistoets verschuiven. Volgens de plaats waar de muis zich op de lijn bevindt, wordt een hoekpunt of een kant van de lijn verschoven. Dat wordt door verschillende muiscursoren aangeduid:



Als de muis zich boven een verticale verbindingenlijn bevindt, kan je met een ingedrukte linker muistoets de hele verticale lijn verschuiven.



Als de muis zich boven een horizontale verbindingenlijn bevindt, kan je met een ingedrukte linker muistoets de hele horizontale lijn verschuiven.



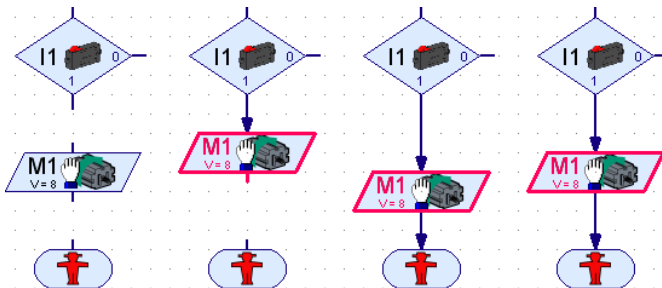
Als de muis zich boven een schuine verbindingenlijn bevindt, wordt een nieuw punt in de verbindingenlijn ingevoegd als je op de linker muistoets drukt. Je moet de linker muistoets ingedrukt houden en de muistoets pas loslaten als de muis zich daar bevindt waar het nieuw punt moet komen.



Als de muis zich in de buurt van een hoekpunt of een eindpunt van een verbindingenlijn bevindt, kan je het punt met een ingedrukte linker muistoets verschuiven. Een verbonden lijneindpunt kan je enkel op een andere passende aansluiting van een programma-element trekken. In dit geval wordt het eindpunt van de verbindingenlijn met dit aansluitpunt verbonden. In het andere geval wordt het punt niet verschoven.

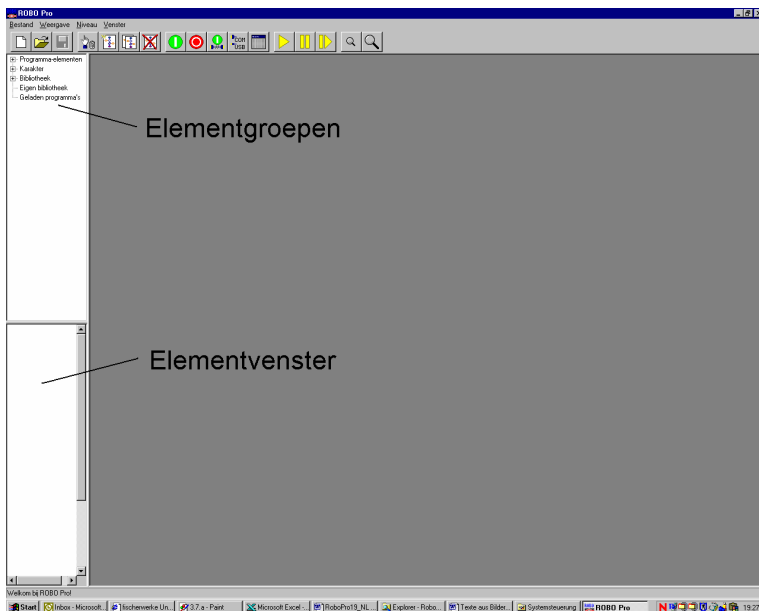
#### Verbindingslijnen eens anders

Verbindingenlijnen kunnen ook door het verschuiven van programma-elementen getrokken worden. Als je een programmalijn zodanig verschuift dat zijn ingang een of twee rasterpunten onder de uitgang van een ander ligt, wordt tussen de beide elementen een verbindingenlijn tot stand gebracht. Dat geldt ook bij een uitgang die over een ingang geschoven wordt. Daarna kan je het programma-element in zijn eindpositie schuiven of verdere verbindingen voor de resterende in- en uitgangen tekenen:



## 4 Niveau 2: Werken met subprogramma's

Nadat je succesvol je eerste besturingsprogramma opgesteld en getest hebt, ben je klaar voor ROBO Pro Niveau 2. Kies nu in het menu **Niveau** het **Niveau 2: subprogramma's**. Je zal het verschil beseffen meteen werken. Het elementvenster is verdwenen en in de plaats daarvan heb je nu aan de linker rand twee vensters boven elkaar:



Maar wees niet bang! Het elementvenster is er nog. Alleen is het nu leeg. In niveau 2 zijn er meer programma-elementen, zodat het onoverzichtelijk zou zijn om alles in een venster te pakken. Vandaar dat de elementen vanaf niveau 2 in elementgroepen samengebracht zijn. De elementen zijn in groepen georganiseerd op een gelijkaardige manier als bestanden in mappen op de harde schijf van je computer. Als je in het bovenste venster aan de linkerrand een groep selecteert, verschijnen alle elementen van deze groep in het onderste venster. De elementen uit het niveau 1 vind je in de groep **Programma-elementen / Basiselementen**. Aangezien het elementvenster maar half zo groot meer is, moet je de rolbalk rechts aan het elementvenster gebruiken om de onderste elementen weer te geven.

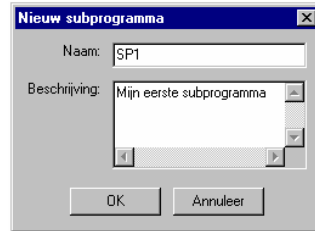
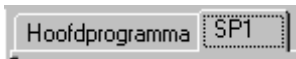
Zo, maar nu gaan we over tot het eigenlijke thema: de subprogramma's! De stroomschema's die we tot nu ontworpen hebben, zijn weliswaar nog niet zo omvangrijk dat we nu al het overzicht verliezen, maar je kan je waarschijnlijk wel voorstellen dat dit bij grotere projecten met omvangrijke stroomschema's gemakkelijk het geval kan zijn. Plots staat je werkblad vol bouwstenen, overal staan verbindinglijnen en op het beeldscherm moet men met de scrollbalk voortdurend heen en weer gaan. "Waar was deze of die uitgang nu weer?" Kortom - er dreigt een kleine chaos te ontstaan! Is er dan geen mogelijkheid om hier voor wat meer orde te zorgen? Toch wel - en dat noemt men een **Subprogramma**!

## 4.1 Je eerste subprogramma

Een subprogramma lijkt goed op de programma's die je tot nu hebt leren kennen. Om dat wat beter te kunnen onderzoeken, moet je eerste een nieuw programma en in het programma een nieuw leeg subprogramma maken. Druk daarvoor op Programma **Nieuw** en dan op de knop **SP Nieuw** in de toolbalk. Er verschijnt een venster waarin je de naam en een beschrijving voor het subprogramma kan ingeven.

De naam mag niet te lang zijn (circa 8-10 letters), aangezien het symbool voor het subprogramma anders erg groot wordt. Je kan alle gegevens die je hier ingeeft natuurlijk later op gelijk welk moment veranderen.

Zodra je het venster **Nieuw subprogramma** met OK sluit, verschijnt het nieuwe subprogramma in de lijst van de subprogramma's.



SP Nieuw

Je kan op gelijk welk moment heen en weer schakelen tussen het hoofdprogramma en een subprogramma door in de lijst van de subprogramma's op de programma-naam te klikken. Maar omdat beide programma's nog leeg zijn, ziet men nog geen verschil.

We gaan nu de besturing van de garagepoort uit het vorige hoofdstuk (zie hoofdstuk 3.6 *Verdere programma-elementen* op pagina 22) in subprogramma's indelen. Het programma bestaat uit vier functie-eenheden:

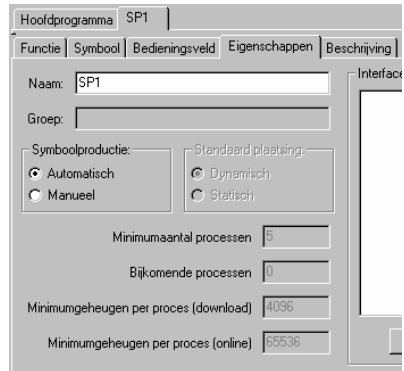
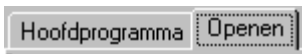
- Wachten tot Schakelaar I1 ingedrukt wordt
- Poort openen
- 10 seconden wachten
- Poort sluiten

Het openen en sluiten gaan we nu in twee subprogramma's verdelen. De beide subprogramma's kan je dan uit het hoofdprogramma slechts met een symbool oproepen. Het wachten op schakelaar I1 en de wachttijd van 10 seconden blijven in het hoofdprogramma, aangezien beide slechts uit een element bestaan. Je hebt nu een nieuw programma namens **Subprogramma 1** aangelegd. **Openen** en **Sluiten** zouden echter betere namen zijn voor de twee subprogramma's. Je kan het reeds aangemaakte subprogramma hernoemen door via de lijst van de subprogramma's eerst subprogramma 1 te selecteren voor zover dat nog niet gebeurd is.

Schakel dan via de functiebalk over naar het venster Eigenschappen voor het subprogramma, door op **Eigenschappen** te klikken. Hier kan je de naam **SP 1** in **Openen** veranderen. De meeste andere velden kunnen enkel in het expertniveau gewijzigd worden. Het punt van de **Symbolcreatie** zal straks uitgelegd worden.

Als je in de functiebalk op beschrijving klikt, kan je de eerder gegeven beschrijving wijzigen, hoewel "Mijn eerste subprogramma" nog steeds volledig toepasselijk is.

Klik nu in de functiebalk op **Functie** zodat je de functie van het subprogramma kan programmeren. Nu zie je terug het programmavenster waar je reeds in het vorig hoofdstuk de programma-elementen voor je eerste ROBO Pro programma ingevoegd hebt. Let erop dat in de lijst van de subprogramma's het subprogramma **Openen** geselecteerd is:

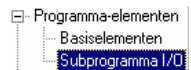


Ben je klaar om je eerste subprogramma te programmeren? Vooruit dan maar! Maar waarmee begint een subprogramma eigenlijk? Een goede vraag! Een hoofdprogramma ben je altijd met het startelement begonnen. Een subprogramma begint met een gelijkaardig element, de ingang van het subprogramma. Dat element heet zo omdat door dit element de programmeiading van het hoofdprogramma naar het subprogramma gaat. Je kan hier geen startelement gebruiken, omdat hier geen nieuw proces gestart moet worden.



|  |                     |  |
|--|---------------------|--|
|  | Startelement        | Start een nieuw, autonoom proces   |
|  | Ingang subprogramma | Hier wordt de programmeiading van het hoofdprogramma aan het subprogramma doorgegeven. |

De ingang van het subprogramma vind je in het venster voor elementgroepen onder **Subprogramma I/O**. Plaats de ingang van het subprogramma boven in het programmavenster voor het subprogramma **Openen**. Een ingangselement van het subprogramma kan je ook een andere naam geven dan **Ingang**, maar dat is enkel nodig als je later een subprogramma met meerdere ingangen schrijft.



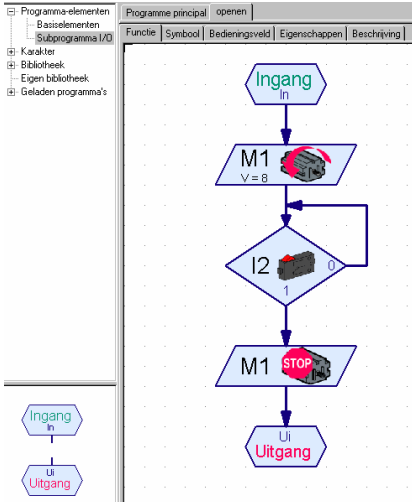
Het verdere programmaverloop in het subprogramma is identiek met het deel Openen van het vorig hoofdprogramma. Je schakelt de motor M1 met draairichting links aan, wacht tot de schakelaar aan de ingang I2 gesloten is en schakelt dan de motor terug uit.

Om het subprogramma af te sluiten, gebruik je een uitgang voor het subprogramma. Het verschil tussen de uitgang van een subprogramma en het stopelement is hetzelfde als datgene tussen de ingang van het subprogramma en het begin van de procedure.



|  |                              |   |
|--|------------------------------|---|
|  | Stopelement                  | Beëindigt de uitvoering van het programma van een onafhankelijk proces                    |
|  | Uitgang van het subprogramma | Hier wordt de programmaleiding van het subprogramma terug aan het hoofdprogramma gegeven. |

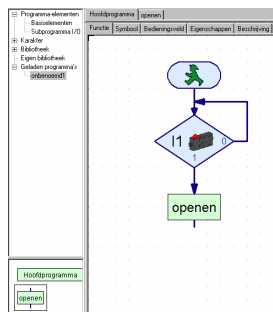
Je afgewerkt subprogramma moet er nu ongeveer zo uitzien:



Let erop dat je het subprogramma echt op **Openen** en niet onder het **Hoofdprogramma** opgeslagen hebt. Schakel nu in de lijst van de subprogramma's terug over van **Openen** naar **Hoofdprogramma**. Je ziet nu het nog steeds lege programmavenster van het hoofdprogramma. Voer in het hoofdprogramma zoals gewoonlijk een startelement (geen ingang van een subprogramma!) in. Ook de ondervraging van de schakelaar aan I1, die de garagepoort moet openen, voer je zoals gewoonlijk in het hoofdprogramma uit.

Je nieuwe subprogramma kan je nu zoals een gewoon programma-element in je hoofdprogramma (of een ander subprogramma) invoegen. Je vindt dat in het venster elementgroepen onder **Geladen programma's** en de bestandsnaam van je programma. Als je het bestand nog niet opgeslagen hebt, is de naam **onbenoemd1**. Als je nog meer programmaprogramma's geladen hebt, kan je in het selectieprogramma ook de subprogramma's die tot andere bestanden behoren, selecteren. Op deze manier kan je zeer gemakkelijk subprogramma's uit een ander bestand gebruiken.

In de elementgroep **Geladen programma's / onbenoemd1** vind je twee groene subprogramma-symbolen. Het eerste met de naam **Hoofdprogramma** is het symbool voor het hoofdprogramma. Dat wordt eerder zelden als subprogramma gebruikt, maar het is wel mogelijk, bijvoorbeeld als je een heel machinepark stuurt en de besturingen voor de afzonderlijke machines vooraf afzonderlijk



als hoofdprogramma ontwikkeld hebt. Het tweede symbool met de naam **Openen** is het symbool van je nieuw subprogramma. **Openen** is de naam die je onder eigenschappen ingeschreven hebt. Voeg nu het symbool van het subprogramma zoals je dat van gewone programma-elementen gewend bent, in je hoofdprogramma in. Zo eenvoudig is dat!

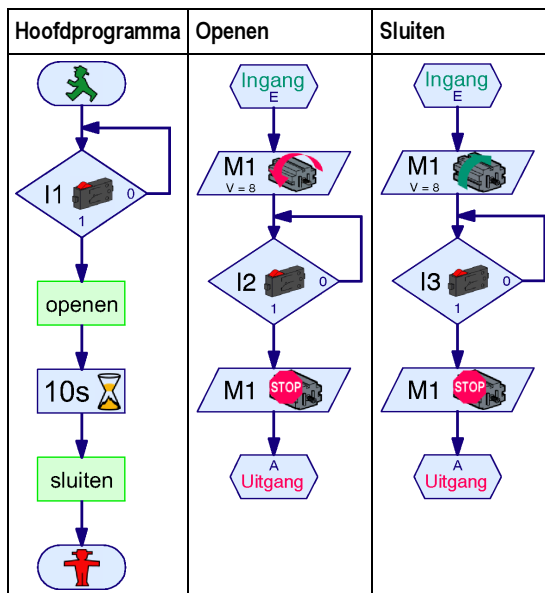


SP Nieuw

Je kan je hoofdprogramma reeds met een stopelement afsluiten en het uitproberen. De poort kan geopend worden door op de schakelaar I1 te drukken, maar het deel sluiten hebben we nog niet geprogrammeerd. Daarvoor maak je een ander subprogramma aan. Druk op de knop **SP nieuw** in de toolbalk en geef in het Venster **Nieuw subprogramma** als naam **Sluiten** in. Een beschrijving hoeft je niet in te geven, maar het kan ook geen kwaad, zodat je later nog weet waarvoor het subprogramma bedoeld is.

Geef nu in het programmavenster voor het subprogramma **Sluiten** het programma in voor het sluiten van de garagepoort. Je begint terug met een ingang voor het subprogramma. De motor **M1** moet eerst naar rechts draaien. Zodra de eindschakelaar aan I3 gesloten wordt, moet de motor **M1** stoppen. Het subprogramma wordt opnieuw met een subprogramma-uitgang afgesloten.

Schakel nu terug over van de lijst van de subprogramma's naar het hoofdprogramma. Als je vooraf je hoofdprogramma met een stopelement aangesloten hebt om het uit te proberen, moet je het stopelement nu terug wissen. Nadat de garagepoort geopend is, moet ze 10 seconden open blijven voor ze terug gesloten wordt. Na een wachttijd van 10 seconden voeg je uit de elementgroep **Geladen programma's / onbenoemd1** het symbool **Sluiten** voor het subprogramma in. Het hoofdprogramma en de twee subprogramma's moeten er ongeveer zo uitzien:



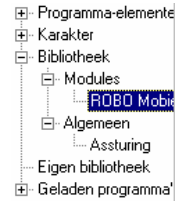
Het programma start aan het startelement in het **hoofdprogramma**. Dan wacht het programma tot de schakelaar I1 ingedrukt is. Daarvoor zou je overigens ook het element **Wachten op ingang** kunnen gebruiken (zie hoofdstuk 7.1.8 *Wachten op Ingang* op pagina 61). Nadat de schakelaar I1 ingedrukt is, stoot het hoofdprogramma op de oproep van het subprogramma **Openen**. De



programma-uitvoering wisselt daardoor naar de subprogramma-ingang van het subprogramma **Openen**. Het subprogramma **Openen** opent de garagepoort en komt dan tot een subprogramma-uitgang. Op deze plaatst vertakt het programma terug naar het hoofdprogramma. Na het einde van het subprogramma **Openen** wordt in het hoofdprogramma 10 seconden gewacht. Aansluitend schakelt de programma-uitvoering over naar het subprogramma **Sluiten** dat de garagepoort terug sluit. Na de terugkeer uit het subprogramma **Sluiten** stoot het hoofdprogramma op een stopelement, waardoor het programma beëindigd wordt.

## 4.2 De bibliotheek van de subprogramma's

Je kan zeer gemakkelijk subprogramma's van het ene bestand naar een ander bestand kopiëren door beide bestanden te laden en dan een subprogramma uit een bestand via de elementgroep **Geladen programma's** in het andere bestand in te voegen. Voor vaak gebruikte subprogramma's gaat het echter nog eenvoudiger en weliswaar met de **bibliotheek**. ROBO Pro bevat een bibliotheek van klaargemaakte subprogramma's die je gemakkelijk terug kan gebruiken. Bovendien kan je een eigen bibliotheek aanleggen waarin je je vaak gebruikte subprogramma's kan bewaren. .

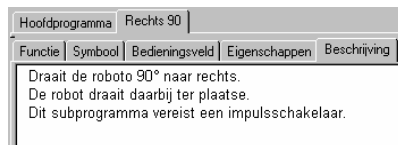


### 4.2.1 Gebruiken van de bibliotheek

De **Bibliotheek** is eerst in twee hoofdgroepen onderverdeeld. Onder de groep **Modules** vind je subprogramma's die je voor modellen uit sommige modules kan gebruiken. Onder de groep **Algemeen** vind je subprogramma's die je voor alle mogelijke modellen kan gebruiken. De meeste van deze subprogramma's uit de groep **Algemeen** vereisen evenwel technieken uit het niveau 3 die in het volgend hoofdstuk uitgelegd worden.

Voor elke Computing-module, zoals bijvoorbeeld de ROBO Mobiele Set, bestaat er in de groep **Modules** een eigen subgroep. Deze is soms verder in modellen onderverdeeld die je in de montagehandleiding van de module vindt. Als je de module of een van de modellen selecteert, worden in het elementvenster de afgewerkte subprogramma's voor dit model getoond.

Als je met de muis op een van de subprogrammasymbolen wijst, wordt een korte beschrijving getoond. Als je een subprogramma in je programma invoegt, kan je een precieze beschrijving weergeven door het subprogramma in de subprogrammalijst te selecteren en dan in de functielijst op **Beschrijving** te klikken:



**Opgelet:** Als je een subprogramma uit de bibliotheek invoegt, worden gedeeltelijk verdere subprogramma's ingevoegd die door dit subprogramma gebruikt worden. Je kan alle subprogramma's terug verwijderen door in het menu **Bewerken** de functie **Undo** (ongedaan maken) te selecteren.

## 4.2.2 Het gebruik van de eigen bibliotheek

Nadat je een tijdje met ROBO Pro bezig geweest bent, zal je waarschijnlijk eigen subprogramma's hebben die je vaker gebruikt. Opdat je niet elke keer het overeenkomstig bestand zou moeten zoeken en laden, kan je ook een eigen bibliotheek van subprogramma's aanleggen die precies hetzelfde werkt als de voorgedefinieerde bibliotheek. De eigen bibliotheek bestaat uit een of meerdere ROBO Pro bestanden die allemaal in een map opgeslagen zijn. Voor elk bestand in deze map wordt in de groepselectie een eigen groep aangegeven.

In welke map je je eigen bibliotheek wilt opslaan, kan je in het menu **Bestand** onder **Eigen Bibliotheekdirectory** aangeven. De standaard bestandenlijst (directory) voor de eigen bibliotheek is C:\Programma's\ROBOPro\Eigenbibliotheek. Als je een eigen gebruikersdirectory op je computer hebt, is het raadzaam om er daar een eigen map voor aan te leggen en die te gebruiken.

**Tip:** In het begin kan je de map waar je ook je ROBO Pro programma's opslaat, bij **Eigen bibliotheekdirectory** aangeven. Dan heb je snel toegang tot alle bestanden in je werkmapij.

## Organiseren van je eigen bibliotheek

Er zijn in ROBO Pro geen speciale functies om een bibliotheek te wijzigen. Maar toch gaat dat heel eenvoudig, als je aan een bibliotheekgroep subprogramma's wilt toevoegen of eruit wilt verwijderen, moet je eerst het overeenkomstig bestand laden. Je vindt dit bestand in de directory die je onder **Eigen bibliotheekdirectory** ingesteld hebt. Nu kan je bijvoorbeeld een tweede bestand laden en hieruit een subprogramma uit de groep **Geladen programma's** in het hoofdprogramma van de bibliotheek trekken. Bij een bibliotheek is het hoofdprogramma geen echt programma, maar enkel een verzameling van alle subprogramma's van de bibliotheek. Het hoofdprogramma zelf wordt bij bibliotheken in het elementvenster niet aangegeven. Je kan natuurlijk ook subprogramma's uit een bibliotheek wissen of subprogramma's wijzigen.

Als je een bibliotheekbestand gewijzigd en opgeslagen hebt, moet je in het menu **Bestand** het menupunt **Eigen bibliotheek actualiseren** kiezen. Daardoor wordt de bestandenlijst in het groepsvenster geactualiseerd.

## 4.3 Bewerken van subprogrammasymbolen

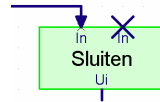
Zoals je in het vorig hoofdstuk gezien hebt, creëert ROBO Pro voor je subprogramma's automatisch groene subprogrammasymbolen. Je kan echter ook eigen symbolen tekenen die beter verduidelijken wat je subprogramma's doen. Daarvoor moet je in het venster **Eigenschappen** het subprogramma van automatisch naar manueel omschakelen. Aansluitend kan je in de functiebalk van **Eigenschappen** naar **Symbool** omschakelen en daar het subprogrammasymbool bewerken. Tekenfuncties vind je in het venster voor elementgroepen onder **Tekenen**.



Onder **tekenen / vormen** vind je alle gebruikelijke grafische basiselementen zoals een rechthoek, cirkel, ovaal, veelhoek en dergelijke. Onder **Tekenen / Tekst** vind je tekstobjecten in verschillende lettertypes. In de andere groepen onder Tekenen vind je functies voor het wijzigen van de kleur en gelijksoortige eigenschappen van geselecteerde elementen. De juiste toepassing van de tekenfuncties wordt uitgelegd in het hoofdstuk 9 *Tekenfuncties* op pagina 89. Let ook op de functies in het hoofdmenu onder **Tekenen**.

Je kan ook de aansluitingen van het subprogramma verschuiven, maar je kan de aansluitingen niet wissen of nieuwe toevoegen. In het subprogrammasymbool is er steeds voor elke programma-ingang of subprogramma-uitgang van de subprogrammafunctie een aansluiting. De aansluitelementen worden ook automatisch aangemaakt als je omgeschakeld bent naar manuele symboolcreatie.

Zodra je het venster voor de symboolbewerking verlaat, zullen alle oproepen van het subprogramma in het hoofdprogramma of in andere subprogramma's overeenkomstig aangepast worden. Gelieve erop te letten dat wanneer je aansluitingen van een subprogramma verschoven hebt, bij het oproepen van een subprogramma een kleine verwarring kan ontstaan als de aansluitingen reeds aangesloten waren. De eindpunten van de verbindingslijnen eindigen dan soms niet meer op de juiste aansluiting, wat door een kruis op het lijneindpunt en aan de aansluiting aangeduid wordt (zie afbeelding). In de regel is het voldoende dat je een keer met de linker muistoets ergens op de verbindingslijn klikt. De lijn wordt dan automatisch terug verlegd. Bij subprogramma's met veel verbindingen kan het echter zijn dat je de lijn nog moet bewerken.

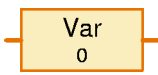


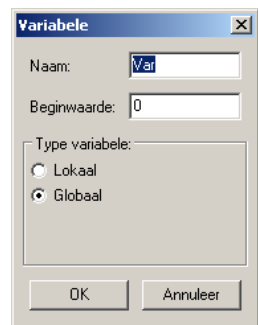
## 5 Niveau 3: Variabelen, bedieningsvelden & Co

Denk eraan om ROBO Pro in het menu **Niveau op Niveau 3** (of hoger) in te stellen!

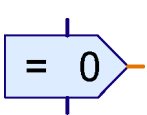
Stel je eens voor dat je in een museum in een tot nu onbekende zijgang een fascinerende machine ontdekt die je absoluut met fischertechniek wilt nabouwen. Bij het onderzoek van de machine vergeet je gewoon de tijd en je merkt niet dat alle andere bezoekers het museum verlaten. Pas als het museum al gesloten is, heb je de machine grondig genoeg onderzocht om ze te kunnen nabouwen. Maar jammer genoeg moet je eerst een onaangename nacht alleen in het museum doorbrengen voor je kan beginnen. Opdat dit niet meer zou gebeuren, biedt de museumdirecteur je aan om een bezoeker teller te programmeren die alle bezoekers bij het binnengaan en het buitengaan telt en die een rode controlelamp inschakelt zodra er nog bezoekers in het museum zijn. Maar hoe doe je dat? Hoe kan je met ROBO Pro iets tellen? Het antwoord luidt: met **variabelen**.

### 5.1 Variabelen en bevelen

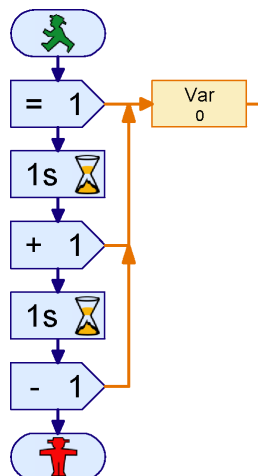
 Een variabele is een element dat een getal kan onthouden. In het venster Eigenschappen van de variabelen geef je een **Naam** in die een aanwijzing moet geven wat voor een getal in de variabele opgeslagen is. Onder **beginwaarde** kan je aangeven welk getal aan het begin van het programma in de variabelen opgeslagen moet worden. De instelling **Type variabele** wordt in het hoofdstuk 7.3.2 *Lokale variabele* op pagina 65 uitgelegd.



Het opgeslagen getal kan je veranderen door bevelen naar de variabele te besturen. Een variabele verstaat 3 verschillende bevelen: =, + en -. Het = bevel vervangt het opgeslagen getal door een nieuw getal. Het + en - bevel telt bij het opgeslagen getal iets bij of trekt er iets van af.

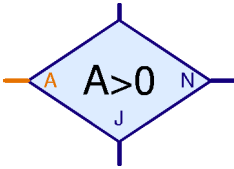
 De bevelen stuur je met een **Bevelement** naar de variabele. Het bevelement heeft zoals de meeste andere programma-elementen bovenaan een blauwe programma-ingang en onderaan een blauwe programma-uitgang. Rechts heeft het echter iets totaal nieuw, een **oranje aansluiting**. Dat is een beveluitgang. Steeds wanneer het bevelement uitgevoerd wordt, stuurt het via deze uitgang naar alle aangesloten elementen een bevel. De variabele heeft aan de linkerkant een passende bevelingang. Als je de beveluitgang met de bevelingang verbindt, tekent ROBO Pro in plaats van de gebruikelijke blauwe verbindingen een oranje lijn. Via de oranje lijnen kunnen programma-elementen bevelen of berichten besturen en zo informatie uitwisselen.

Het programma rechts stuurt de variabelen **Var** eerst een = 1 bevel. Een bevel bestaat in de regel uit een eigenlijk bevel zoals = en een waarde zoals 1. Het =1 bevel zet de variabele op 1. Na



een seconde stuurt het programma van de variabele een **+1** bevel. De variabele telt daarop de tot nu actuele waarde 1 bij en heeft dan de waarde 2. Na nog een seconde stuurt het programma een **-1** bevel. Daarop heeft de variabele terug de waarde 1.

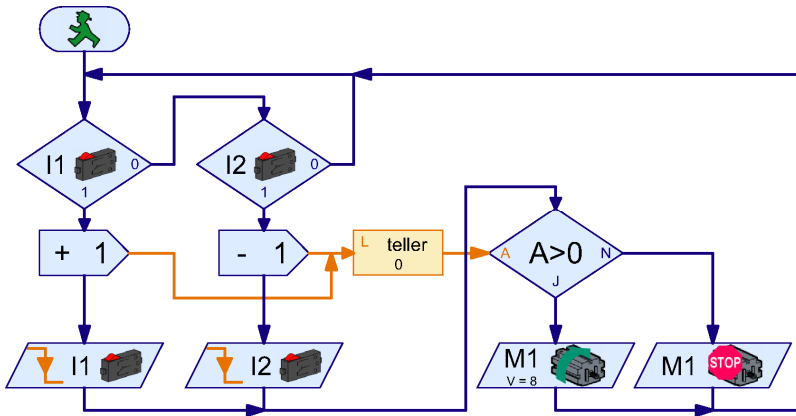
Probeer eens om dit eenvoudig programma in ROBO Pro te tekenen. De bevelementen vind je in de groep **Bevelen**, de variabele in de Groep **Variabele**, **Timer**, .... Als je het programma in de Online-bedrijfsmodus uitvoert, zie je hoe de waarde van de variabele verandert.



Dat is goed en wel zal je misschien zeggen. Ik kan de waarde van de variabele bekijken, maar wat doe ik daarmee? Dat is heel eenvoudig: de variabele heeft rechts een oranje aansluiting en via deze aansluiting stuurt ze berichten met haar actuele waarde naar alle aangesloten elementen. Er zijn enkele elementen in ROBO Pro, die links een oranje ingang hebben die je met de uitgang van de variabelen kan verbinden. Zo vind je bijvoorbeeld in de groep **Vertakking**, **wachten**, ... een Ja / Neen vertakkingselement dat niet

direct een ingang ondervraagt, maar een willekeurige waarde kan opvragen, onder meer de waarde van een variabele.

Daarmee kan het aantal bezoekers voor het museum als volgt geprogrammeerd worden:



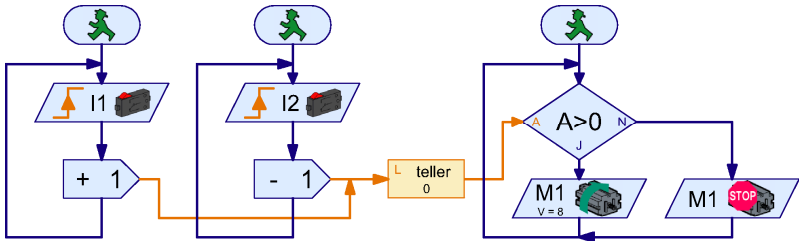
Het draaikruis aan de ingang bedient de schakelaar aan I1, het draaikruis aan de uitgang de schakelaar aan I2. Zodra I1 ingedrukt is, stuurt het programma de variabele **Teller** een **+ 1** bevel. Aansluitend wacht het programma tot de schakelaar aan I1 losgelaten wordt. Met de schakelaar voor de uitgang aan I2 gedraagt het zich precies hetzelfde. Alleen dat hier naar de variabele **Teller** een **- 1** bevel gestuurd wordt. Telkens als de teller veranderd is, wordt de tellerstand gecontroleerd. Als de variabele **Teller** een waarde **> 0** heeft, wordt de rode controlelamp aan M1 ingeschakeld en anders uitgeschakeld.



Teken nu het bovenstaand programma na en probeer het uit. Zodra je de schakelaar aan I1 indrukt en terug loslaat, licht de controlelamp aan M1 op. Als je de schakelaar aan I2 bedient, gaat ze terug uit. Als je I1 meermaals moet bedienen, moet je I2 even vaak bedienen, zodat de controlelamp terug dooft. Probeer ook eens wat er gebeurt als er eerst 5 bezoekers komen, er dan 2 weggaan en er dan nog eens 3 komen. Hoe vaak moet je nu de schakelaar aan I2 bedienen opdat de controlelamp terug zou doven?

## 5.2 Variabelen en meerdere processen

Misschien is het je bij het testen van het aantal bezoekers opgevallen dat het problemen geeft als de schakelaars aan I1 en I2 gelijktijdig ingedrukt worden. Zolang een van de schakelaars ingedrukt is, kan het programma niet meer op de andere schakelaar reageren. Maar vermits de bezoekers aan de ingang en aan de uitgang gelijktijdig door het betrokken draaikruis kunnen gaan, leidt dit tot telfouten. Je kan deze fout verhelpen door meerdere parallelle processen te gebruiken. Tot nu hadden alle programma's maar één startelement. Maar je kan ook meerdere startelementen gebruiken. Alle schema's met een eigen startelement worden dan naast elkaar afgewerkt. De vakman spreekt hier van **parallelle processen**. Met deze techniek kan je het programma voor het tellen van de bezoekers als volgt wijzigen:



Voor I1 en I2 worden nu onafhankelijke processen gebruikt. Als de schakelaar aan I1 ingedrukt is, blijft het proces voor I2 onafhankelijk en kan verder de schakelaar aan I2 controleren. Voor het opvragen van de getalwaarde en voor het aan- en uitschakelen van de controlelamp wordt eveneens een eigen proces gebruikt.

Zoals je ziet, zijn er geen problemen om vanuit meerdere processen een bepaalde variabele te gebruiken. Je kan een variabele vanuit meerdere processen bevelen sturen en je kan de waarde van een variabele in meerdere processen gebruiken. Daardoor zijn variabelen ook zeer goed om informatie tussen processen uit te wisselen.



De museumdirecteur is zo enthousiast over je geniale bezoekersteller dat hij je meteen vraagt om een ander probleem op te lossen. Het museum heeft een nieuwe tentoonstelling ingericht. Maar aangezien alle bezoekers de nieuwe tentoonstelling willen zien, heerst daar zo'n gedrag dat niemand nog iets kan zien. De directeur wil daarom het aantal bezoekers in de tentoonstelling tot 10 beperken. Aan de ingang en aan de uitgang van de tentoonstelling heeft de directeur telkens een draaikruis opgesteld. Het draaikruis aan de ingang kan elektronisch vergrendeld worden. Nu hoeft hij enkel nog een goede programmeur nodig, namelijk jou!

Probeer het beschreven programma met ROBO Pro te ontwikkelen. Het werkt in feite zoals de bezoekersteller. De elektronische vergrendeling van de ingang simuleer je door een rode lamp aan M1, die ingeschakeld moet zijn als er 10 bezoekers in de tentoonstelling zijn.

## 5.3 Bedieningsvelden

Nadat je het probleem met de tentoonstelling opgelost hebt, heeft de museumdirecteur alweer een nieuwe taak. Hij wil weten hoeveel bezoekers er op een dag in zijn museum geweest zijn. Een programma dat kan tellen, is nu geen probleem meer voor jou, maar hoe kan je een waarde aanduiden? Natuurlijk zou je het programma in de online-bedrijfsmodus kunnen uitvoeren en aan de museumdirecteur kunnen tonen bij welke variabelen hij de waarde kan nakijken. Maar voor een computerleek zoals de museumdirecteur is dat nog erg ingewikkeld. Het moet iets zijn dat gemakkelijker is!

Voor dergelijke gevallen dienen de bedieningsvelden van ROBO Pro. Een bedieningsveld is een eigen pagina waarop je aanduidingen en bedieningsknoppen kan tekenen. Laad je programma voor het tellen van de bezoekers en schakel in de functiebalk over naar **Bedieningsveld**.

Funcie | Symbool | Bedieningsveld | Eigenschappen | Beschrijving

Het bedieningsveld is voorlopig een leeg, grijs vlak. Op dit vlak plaats je weergaven en besturingselementen die je in het venster met elementgroepen onder **Bedieningselementen** vindt. Onder de besturingselementen vind je drukknoppen, schuifregelaars en gelijksoortige. Onder Weergaven vind je Tekstweergaven, controlelampen en weergaven met een draaiwijzer.



**Opgelet:** Een bedieningsveld is een deel van een subprogramma. Als je subprogramma's hebt, let er dan op dat je het bedieningsveld onder **Hoofdprogramma** en niet onder een subprogramma bewaart! Later kan je als prof meerdere bedieningsvelden aanleggen.

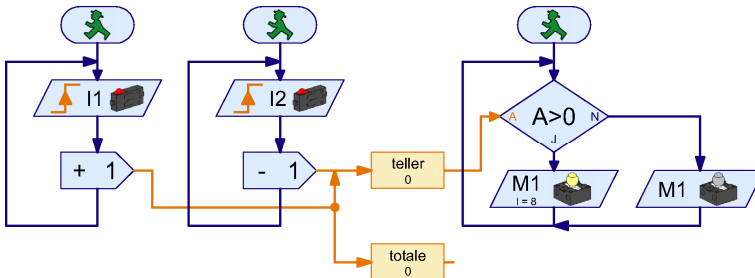
Als je een bedieningsveld getekend hebt en het bedieningsveld later plots verdwenen is, heb je vermoedelijk in de subprogrammalijs een subprogramma geselecteerd. Schakel terug naar **Hoofdprogramma** en je bedieningsveld is er terug.

Var: 0

Voor de bezoeker teller neem je een **Tekstweergave** (de kleur speelt geen rol) uit het elementvenster **Bedieningselementen / Weergaven** en plaats je deze in het bedieningsveld. In deze

weergave moet nu het aantal museumbezoekers aangeduid worden.

Maar nu moet je eerst aan je programma een tweede variabele toevoegen die het aantal bezoekers aan de ingang telt zonder de bezoekers aan de uitgang terug af te trekken. Daarvoor schakel je in de functiebalk terug over naar **Funcie** en voeg je de variabele **Totaal** als volgt in:



Zoals je ziet, kan een bevelement ook gebruikt worden om aan twee variabelen gelijktijdig een bevel te besturen. De -1 bevelen bevatten geen variabele **Totaal** omdat de bevelen langs de oranje lijn enkel in de pijlrichting worden overgedragen. De +1 bevelen worden daarentegen aan beide variabelen doorgegeven. Dit is slechts als voorbeeld bedoeld. In de regel is het eenvoudiger en overzichtelijker om een tweede bevelement te gebruiken.



**Tip:** Als oranje lijnen vertakken, is het vaak praktischer om de lijnen vanaf het doel aan het begin te tekenen. Als je in het bovenstaande voorbeeld de lijn naar de variabele **Totaal** wilt tekenen, klik dan eerst op de ingang van de variabele **Totaal** en verleg de lijn dan achterwaarts tot aan het vertakkingspunt. Als je daarentegen een oranje lijn op een bestaande oranje lijn wilt beginnen, moet je met de linker muistoets dubbel klikken op de plaats waar de nieuwe lijn moet beginnen.



Zo, nu heb je een tekstweergave in het bedieningsveld en een variabele die je in de weergave wilt aanduiden. Hoe combineert men die twee nu? Aangezien de tekstweergave en de variabele op verschillende pagina's staan, kan je die twee immers moeilijk met een lijn verbinden.

Daarom is er een speciaal element dat de waarde die in het bedieningsveld weergegeven moet worden, aan de overeenkomstige weergave doorgeeft. Het hierboven afgebeeld element **Bedieningsvelduitgang** vind je aan het einde van de groep **Ingangen, Uitgangen**. Voeg zo'n bedieningsvelduitgang in je programma naast de variabele **Totaal** een verbind de rechter aansluiting van de variabele met de aansluiting van de **Bedieningsvelduitgang**.

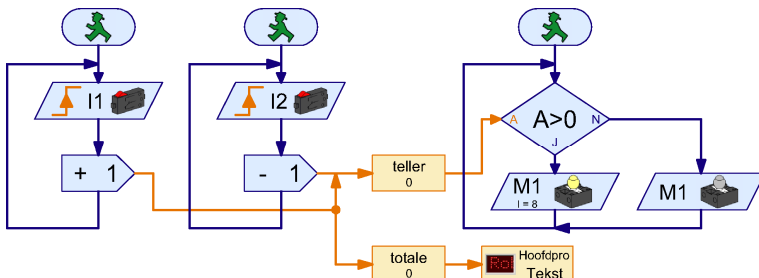
Aangezien je in de regel meer dan een weergave in een bedieningsveld zal hebben, moet je aan de uitgang van het bedieningsveld ook nog meedelen naar welke weergave deze de waarde van de variabele moet sturen. Dat gaat gewoon via het venster Eigenschappen van het element. Als je met de rechter muistoets op de uitgang van het bedieningsveld klikt, zie je een keuzelijst waarin alle weergaven vermeld staan die reeds in het bedieningsveld ingevoegd werden. Aangezien elk subprogramma een eigen bedieningsveld kan hebben, zijn de bedieningsvelden volgens subprogramma's geschikt. In ons voorbeeld is er geen subprogramma, alleen het hoofdprogramma. Daaronder staat een weergave met de naam **Tekst**. Kies deze weergave en klik op OK.



Zodra je de uitgang van het bedieningsveld met een weergave verbonden hebt, verandert het symbool en het opschrift overeenkomstig. De door ons gebruikte uitgang van het bedieningsveld brengt een koppeling tot stand met de tekstweergave met de naam **Tekst** in het (sub-)programma **HOOFD**.



Nadat je de uitgang van het bedieningsveld ingevoegd hebt en met de tekstweergave verbonden hebt, ziet je programma er zo uit:



Probeer het meteen eens uit. Zodra je het programma in de online-bedrijfsmodus gestart hebt, toont de weergave in het bedieningsveld het aantal bezoekers die het draaikruis gepasseerd zijn.

**Tip:** Als je meer dan een weergave in een bedieningsveld wilt gebruiken, is het belangrijk dat je elke weergave een andere naam geeft, zodat je ze bij de koppeling met het programma



kan onderscheiden. Daarvoor druk je met de rechter muistoets op de weergave in het bedieningsveld. Daar kan je onder **ID / Naam** een naam ingeven. Als je dan een uitgang van het bedieningsveld met de weergave verbindt, verschijnt deze naam in het selectievenster van de uitgang van het bedieningsveld. Aangezien we op het moment maar een weergave hebben, is de naam onbelangrijk en behouden we de naam **Tekst**.

Het programma is nog niet helemaal perfect. Wat nog ontbreekt, is een schakelaar om de teller te resetten. Daarvoor willen we echter geen normale schakelaar, maar een knop gebruiken waarop we in het bedieningsveld kunnen drukken.

### Knop

De bedieningsknop vind je in het elementvenster onder de Groep **Bedieningselementen/besturingsselementen**. Schakel in de functie balk over naar **Bedieningsveld** en voeg in het bedieningsveld naast de tekstweergave een knop in. Het opschrift **Knop** is natuurlijk niet echt passend, maar kan via het venster Eigenschappen voor de knop gemakkelijk veranderd worden. Klik met de rechter muistoets op de knop en geef bij **Tekst opschrift** bijvoorbeeld 0000 in en bevestig met OK.

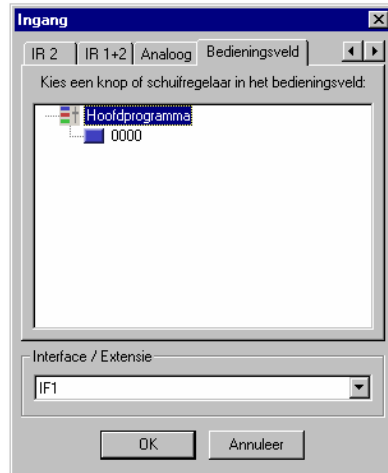


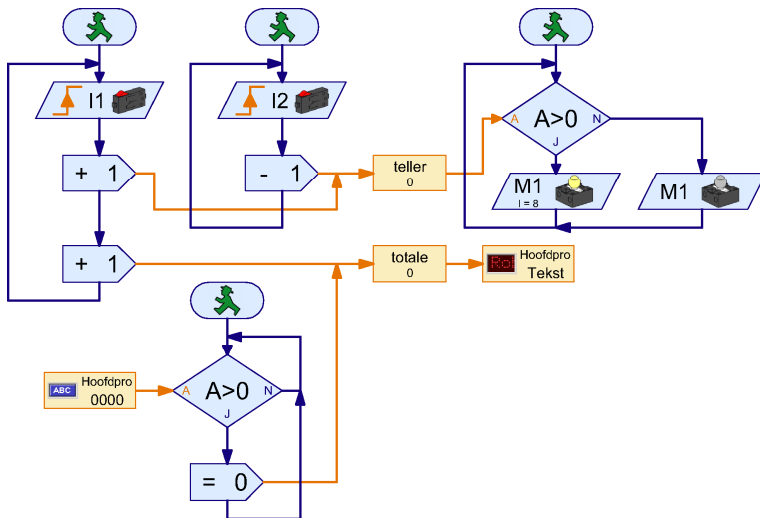
Net zoals bij de tekstweergave hebben we ook een programma-element nodig dat de knop met het stroomschema verbindt. Schakel daarom in de functie balk eerst terug over naar **Functie**. Je vindt in het elementvenster in de groep **Ingangen, Uitgangen** het afgebeelde element **Bedieningsveld-ingang**. Plaats het in het stroomschema onder het vorige schema.

Nu moet je de ingang van het bedieningsveld nog verbinden met de knop in het bedieningsveld. Daarvoor klik je met de rechter muistoets op het element **Bedieningsveld-ingang**. De besturingsselementen zijn zoals bij de weergaven volgens subprogramma's gerangschikt, aangezien elk subprogramma een eigen bedieningsveld kan hebben. Kies nu de knop **0000** en bevestig met OK.

Misschien is het je opgevallen dat men dit element via de ruiterbalk van het venster eigenschappen op alle mogelijke soorten van ingangen kan instellen. Maar dat wordt twee hoofdstukken verder bij *Data-ingangen voor subprogramma's* uitgelegd.

De waarde die de ingang van het bedieningsveld levert, vraag je op met een vertakkingselement. Het afgewerkte programma met reset ziet er nu zo uit:





Zolang de knop **0000** ingedrukt is, wordt naar de totaalteller een **= 0** bevel gestuurd die de teller op 0 zet.

## 5.4 Timer

De museumdirecteur weet na je succes niet meer wat hij zonder jou moet beginnen en benoemt je daarom tot computeradviseur van het museum. Zo'n post brengt natuurlijk veel roem en eer met zich mee, maar ook veel werk en weliswaar het volgende: in het museum zijn er veel modellen die op een knopdruk bewegen. Maar nu drukken sommige bezoekers tamelijk lang op de knoppen, zodat de modellen warm lopen en zo voortdurend hersteld moeten worden. De directeur wil nu dat de modellen lopen zolang de knop ingedrukt wordt, maar hoogstens 30 seconden aan een stuk. Als het model eenmaal gelopen heeft, moet er een pauze van 15 seconden ingelast worden tot het model terug ingeschakeld kan worden.

Hmm, dat is helemaal geen probleem, denk je nu misschien. Een paar wachttijden, een paar programmavertakkingen en klaar. Probeer het gerust eens! Na een tijdje zal je vaststellen dat het helemaal niet zo eenvoudig is en dit om twee redenen:

- Tijdens die tijd van 30 seconden moet het programma de knop ondervragen om vast te stellen of de knop reeds voor het verstrijken van 30 seconden losgelaten wordt. Nu goed, toegegeven, dat kan men door twee processen oplossen die gelijktijdig verlopen, zie hoofdstuk 5.2 *Variabelen en meerdere processen* op pagina 38.
- Als een bezoeker de knop na 5 seconden loslaat en dan na 15 seconden terug indrukt, moet de wachttijd van 30 seconden opnieuw gestart worden. De wachttijd is echter pas  $5 + 15 = 20$  seconden verstreken en daarmee nog actief. Ook met parallel verlopende processen kan men een wachttijd niet opnieuw starten. Eventueel gaat het met twee wachttijden in drie processen die je afwisselend start, maar je krijgt al stilaan hoofdpijn van het nadenken.

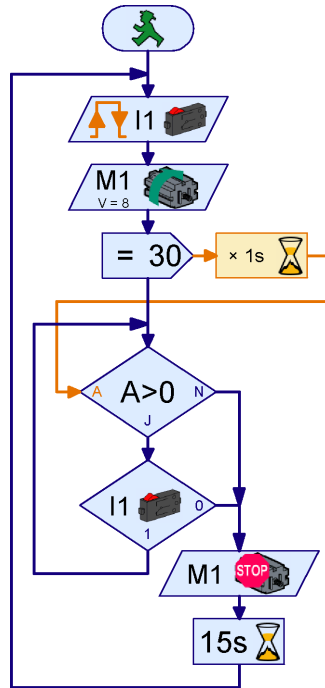


Gaat dat niet wat eenvoudiger? Toch wel, namelijk met **Timervariabelen**, of kortweg **Timer**.

Een timer werkt eerst zoals een doodgewone variabele. De timer onthoudt een getal en je kan dat getal met  $=$ ,  $+$  en  $-$  bevelen wijzigen. Het bijzondere aan een timer is nu dat hij het getal vanzelf regelmatig tot 0 aftelt. De tijd voor een getalstap kan je immers in stappen van een duizendste seconde en van een minuut instellen. Met timers kan men veel tijdbesturingstaken veel eleganter oplossen dan met wachttijden. Zie je al hoe je de taak met een timer kan oplossen?

Juist: Zodra de bezoeker de schakelaar aan I1 indrukt, start je het model en dan stel je de timer in met een  $=$  bevel op  $30 \times 1$  seconde = 30 seconden. Dan vraag je in een lus op of de tijd van 30 seconden verstreken is, en of de schakelaar aan I1 losgelaten werd. Als een van beide afbreekcriteria vervuld is, stop je het model en wacht je 15 seconden. Daarna begint het opnieuw.

Toegegeven, stilaan worden de programma's veeleisender. Maar probeer deze taak eens op te lossen. Ontwikkel een programma met dezelfde functie met wachttijden in plaats van met timers! **Opgelet: Dit is een zeer moeilijke taak en is alleen bedoeld voor diegenen die graag wat langer aan een raadsel puzzelen! Alle anderen kunnen gewoon overgaan naar het volgende hoofdstuk.** Er zijn voor deze taak twee oplossingen: je kan twee wachttijden gebruiken die je afwisselend in eigen processen start. Aangezien er een uitschakeltijd van 15 seconden is, is één van de beide wachttijden uiterlijk na de tweede doorgang afgelopen, zodat die dan opnieuw gestart kan worden. Een ander alternatief zou zijn een timer met een normale variabele en een element **Wachttijd** met een korte wachttijd van bijvoorbeeld een seconde na te bouwen.

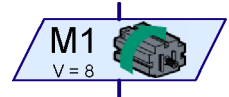


## 5.5 Bevelingen voor subprogramma's

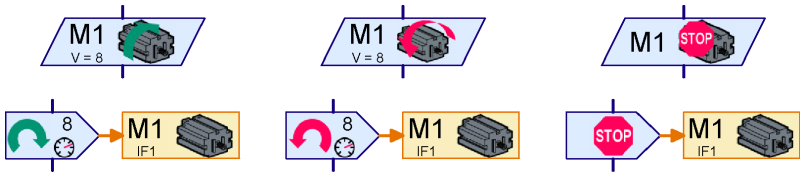
Zoals altijd werkt je programma uitstekend en fischertechnik is blij omdat in het museum alle modellen met de ROBO Interface uitgerust worden. Maar jammer genoeg is zoals overal in openbare inrichtingen in het museum de kassa leeg. Daarom wil de directeur met zo weinig mogelijk interfaces rondkomen. Een ROBO Interface met vier motoruitgangen en voldoende ingangen volstond immers om vier modellen aan te besturen. Aangezien de meeste modellen maar in een richting kunnen draaien, kan je zelfs 8 modellen via de eenpolige uitgangen O1 tot O8 aansturen.

Dat bespaart de museumdirecteur natuurlijk een heleboel geld. Daarvoor moet je nu het programma 7 keer kopiëren en overal de in- en uitgangen aanpassen. Of toch niet? Zou men dat ook niet met subprogramma's kunnen doen?

Dat zou men wel kunnen, maar daarbij duikt een probleem op. Als je in een subprogramma de gebruikte schakelaarondervragingen en motorelementen uit de groep **Basiselementen** gebruikt, wordt bij elke oproeping van het subprogramma dezelfde schakelaar ondervraagd en worden dezelfde motoren aangestuurd. Dat komt doordat bijvoorbeeld in een motoruitgangelement het stuurbevel voor de motor (rechts, links of stop) en het motoruitgangnummer (M1...M8) een eenheid vormen. Aangezien een subprogramma maar één keer bestaat, staat daar ook steeds dezelfde motor in. Wijzig je in de oproeping van een subprogramma het nummer van de motoruitgang, dan wordt die in alle voorhanden zijnde oproepen van het subprogramma eveneens gewijzigd. Je moet dus terug het subprogramma 7 keer kopiëren, elk subprogramma een andere naam geven en overal de in- en uitgangen manueel aanpassen.

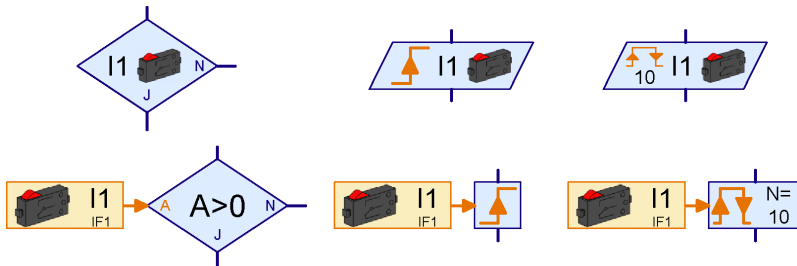


Maar men kan dit probleem veel eleganter oplossen. De truck bestaat erin om de stuurbevelen van de motorsymbolen te scheiden. Dan kan men de stuurbevelen (links, rechts, stop) in het subprogramma en de motorelementen in het hoofdprogramma plaatsen. In het schema van de subprogramma's stuur je dan een bevelement dat je al bij de variabelen hebt leren kennen, de bevelen links, rechts of stop naar het hoofdprogramma, waar je ze dan naar de verschillende motoren kan doorsturen. Voor de motor is er een motorelement dat slechts een motor vertegenwoordigt zonder vast te leggen wat de motor moet doen. Dat element heeft een bevelingang waarnaar je bevelen kan besturen. De elementen uit de groep basiselementen kan je als volgt door een bevelement en een motorelement vervangen:



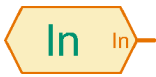
In de bovenste regel zie je telkens een motorelement uit de groep **Basiselementen**. Op de tweede regel staat een combinatie van een bevelement uit de groep **Bevelen** en een motorelement uit de groep **Ingangen, Uitgangen** afgebeeld die precies hetzelfde effect heeft. De bovenste elementen zijn immers maar afkortingen of vereenvoudigingen voor de combinaties op de onderste regel. Beide zenden een bevel rechts, links of stop naar de motor **M1**.

Hetzelfde geldt ook voor het ondervragen van schakelaars:



In de bovenste regel zie je terug elementen uit de groep **Basiselementen**. Op de onderste regel vind je telkens een combinatie van een digitale ingang en een element uit de groep **Vertakking, Wachten, ...** Het oranje element van de **Digitale ingang** vind je zoals het motorelement in de groep **Ingangen, Uitgangen**.

Met deze truck kan je de logica van een stroomschema van de in- en uitgangen scheiden. Maar er ontbreekt nog iets. Als de motor- en schakelaarelementen in het hoofdprogramma moeten staan en de bevelen in een subprogramma, moet er een manier bestaan om de schakelaars en de motorelementen met het subprogramma te verbinden. De daarvoor benodigde aansluitelementen vind je in de groep **Subprogramma I/O**.



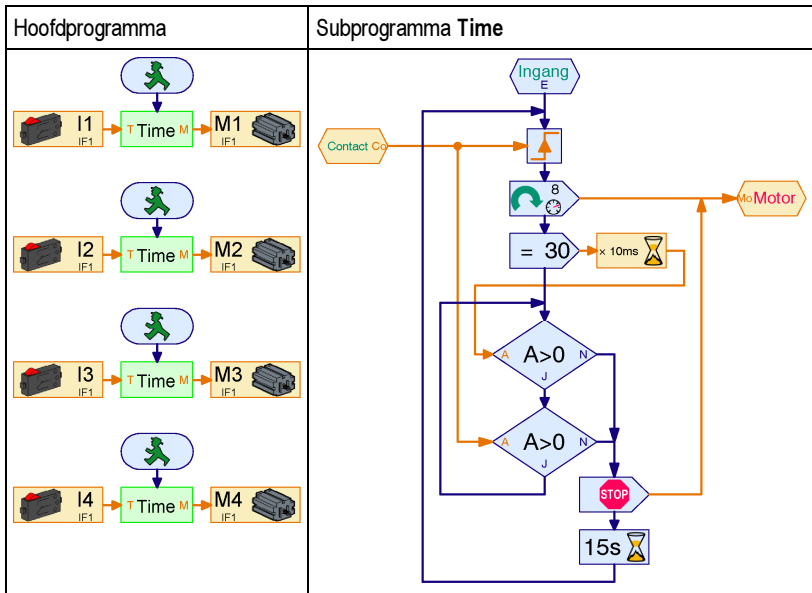
Via een bevelingang in een subprogramma kan je bevelen van buitenuit in een subprogramma besturen. Het digitaal ingangselement (schakelaar) stuurt via de oranje lijn een nieuwe waarde als de toestand aan de ingang verandert (met een zogenaamd "= bevel"). In het dialoogveld van het element

kan je aan de ingang een naam geven.



Via een beveluitgang in een subprogramma kan je bevelen uit een subprogramma naar buiten besturen. Zo kan je bijvoorbeeld de bevelen links, rechts of stop vanuit een subprogramma naar een motor besturen. Ook bij dit element kan je in het dialoogveld een naam ingeven.

Nu heb je alles bij elkaar voor een **meervoudige modeltimer met subprogramma's**:



Het **subprogramma Time** is bijna hetzelfde als het programma uit het vorige hoofdstuk. De elementen **Wachten op Digitale ingang I1** aan het begin en in de lus zijn echter vervangen door **Wachten-op**-elementen met data-aansluitingen voor oranje lijnen uit de groep **Vertakking, Wachten, ...**. Beide zijn met de bevelingang **Schakelaar** van het subprogramma verbonden. De twee besturingselementen voor de motor aan het begin en aan het einde van het programma werden door bevelementen vervangen. Beide zenden hun bevelen naar de Beveluitgang **Motor** van het subprogramma.



In het **Hoofdprogramma** wordt het subprogramma **Time** vier keer opgeroepen. De Bevelingang **Schakelaar** van het subprogramma heeft aan het groene subprogrammasymbool aan de linkerkant automatisch de oranje aansluiting **T** aangemaakt. Door de beveluitgang **Motor** van het subprogramma is aan de rechterkant de aansluiting **M** ontstaan. De aansluiting **T** van het

subprogrammasymbool wordt telkens met een van de schakelaars **I1** tot **I4** verbonden. Aan de aansluiting **M** wordt telkens een van de motoren **M1** tot **M4** aangesloten. Op deze manier ondervraagt elke oproeping van het subprogramma **Time** een andere schakelaar en stuurt een andere motor!

Probeer het bovenstaande subprogramma en hoofdprogramma na te tekenen en probeer het uit. Je moet eerst het subprogramma tekenen omdat je het subprogramma anders niet in het hoofdprogramma kan invoegen. Als je moeilijkheden met het subprogramma hebt, lees dan hoofdstuk 4 Niveau 2: *Werken met subprogramma's* op pagina 28 nog eens door.

## 5.6 Lijsten (Arrays)

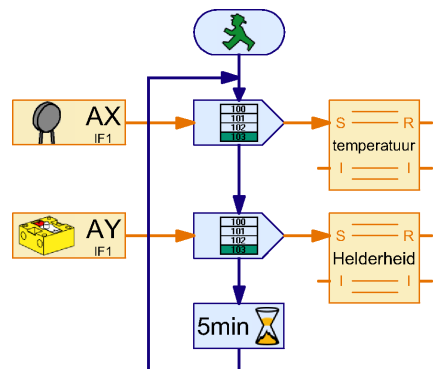
Nadat alle experimenteerstanden in het museum met je kostenbesparende besturing uitgerust zijn, laat het volgende probleem van de museumdirecteur niet lang op zich wachten. In een zaal met zeer waardevolle antieke tentoonstellingsstukken is het de laatste tijd tot schadelijke temperatuurschommelingen gekomen. Je vermoedt dat dit met de binnenvallende zonnestralen te maken heeft. Om dat verband te bewijzen, moet je een apparaat bouwen dat de helderheid en de temperatuur optekent. De ROBO Interface heeft immers meerdere analoge ingangen en je weet ook al hoe men met behulp van variabelen waarde kan opslaan. Dat mag dus allemaal geen probleem zijn, of toch? Om gedurende 12 uur om de 5 minuten twee waarden op te nemen, heeft men 288 variabelen nodig! Dat kan dus een reusachtig en onoverzichtelijk programma worden. Kan men dat misschien terug met een subprogramma vereenvoudigen? Dat wel, maar er is een veel betere weg: het element **Lijst** (Programmeurs noemen dat „Array“).

In een lijst kan men niet slechts een waarde, maar een hele lijst van waarden opslaan. Aan het begin is de lijst in de regel leeg. Als je aan de data-ingang links bovenaan met de benaming **S** een bevel **Toevoegen** stuurt, wordt de waarde die in dit bevelement aangegeven is, aan het einde van de lijst toegevoegd. De maximum lengte van de lijst kan je tussen 1 en 32767 via het venster Eigenschappen van het element **Lijst** instellen. Daarmee wordt het programma voor de optekening van de temperatuur en de helderheid heel eenvoudig:



Aan de analoge ingang **AX** is de temperatuursensor en aan de analoge ingang **AY** de lichtsensor aangesloten. Het programma leest in een lus om de 5 minuten beide waarden en voegt ze via het bevel **Toevoegen** telkens aan een lijst toe.

**Tip:** Bij het invoegen van een bevelement moet je in het venster Eigenschappen de optie **Data-ingang voor bevelwaarde** activeren. Dan verschijnt links aan het bevelement een data-ingang waaraan je de analoge ingang kan aansluiten.



Voor het testen van het programma is het nuttig om de lustijd van 5 minuten tot enkele seconden te verkorten.

Nu vraag je je zeker af hoe je de opgeslagen waarden terug uit de lijst kan uitlezen. Daarvoor zijn er twee mogelijkheden: je kan de waarden zoals bij een gewone variabele uitlezen en verder in je programma verwerken. Aangezien de lijst meerdere elementen bevat, kies je eerst aan de linker

data-ingang met de benaming I het nummer van het element dat je wilt uitlezen. Dan wordt de waarde die dit element bezit, aan de data-uitgang R aan de rechterkant uitgegeven.

ROBO Pro kan echter ook alle waarden van de lijst in een bestand op je computer opslaan dat je dan bv. in Excel verder kan verwerken. Aangezien je in dit geval de opgetekende waarden voor de helderheid en de temperatuur enkel wilt bekijken en vergelijken, is dat beslist praktischer. ROBO Pro slaat de waarden op in een zogenaamd **CSV-bestand** (comma separated values = met een komma gescheiden waarden). CSV-bestanden zijn tekstbestanden die een of meerdere kolommen bevatten met telkens een reeks gegevens. Je kan dus meerdere meetreeksen zoals temperatuur en licht in verschillende kolommen in een CSV-bestand opslaan. De kolommen zijn door een komma gescheiden. In landen waar men 0,5 met een komma en niet 0.5 met een punt schrijft (bv. in Duitsland), wordt als scheidingsteken een **puntkomma ( ; )** gebruikt. Als je problemen hebt bij de uitwisseling van CSV-bestanden tussen ROBO Pro en bijvoorbeeld Microsoft Excel, kan je in het venster Eigenschappen van de Lijst het **Scheidingsteken voor de kolommen** wijzigen.

De naam van het CSV-bestand en de kolommen waarin de inhoud van een lijst opgeslagen moet worden, kan je in het venster Eigenschappen van de lijst onder **CSV-bestand opslaan** instellen. De bestanden worden opgeslagen als het programma in de Online-bedrijfsmodus beëindigd wordt, of als je in het menu **Bestand** het punt **CSV-bestanden opslaan** kiest, zolang het programma nog loopt (Online of Download). In de Download-bedrijfsmodus kan je de ROBO Interface voor het optekenen van de gegevens van de PC scheiden en later voor het opslaan terug aansluiten.

Nadat je het bovenstaand programma in de online-bedrijfsmodus uitgevoerd hebt, kan je het door ROBO Pro aangemaakte .CSV-bestand met de gegevens in Microsoft Excel of een ander rekenbladprogramma openen. Als je geen rekenbladprogramma hebt, kan je ook de Windows Editor (Notepad.exe) gebruiken die je meestal in het startmenu van Windows onder Accessoires vindt.

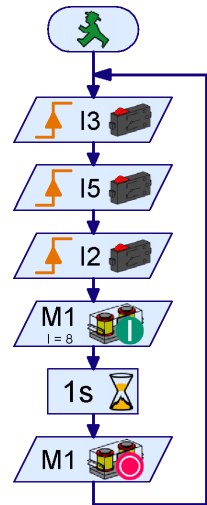
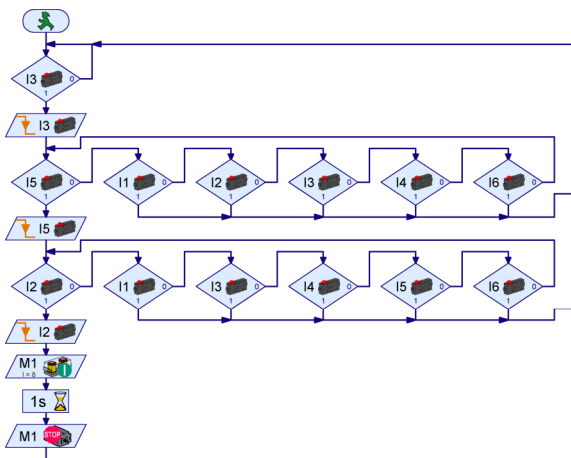
Zolang het programma in de Online-bedrijfsmodus loopt, kan je de bestanden in een lijst ook bekijken door met de rechter muistoets op het lijstelement te klikken.

## 5.7 Operatoren

Je programma voor het optekenen van de helderheid en de temperatuur heeft goed gewerkt, maar uit de optekeningen blijkt dat de temperatuur in de tentoonstellingsruimte van het museum niets te maken heeft met de zon. Het blijkt dat enkele bezoekers de klimaatregeling in de tentoonstellingszaal verwisseld hebben met een modelsturing en daar vlijtig aan gedraaid hebben. Geen wonder dat de temperatuur in de tentoonstellingszaal ontregeld is!

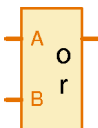
Maar dat probleem is gemakkelijk te verhelpen en weliswaar met een elektronisch cijferslot. Het cijferslot moet een toetsveld met toetsen van 1 tot 6 hebben. Als 3 cijfers na elkaar juist ingegeven zijn, moet het toetsslot via een magneet de afdekking van de klimaatregeling vrijgeven.

Op het eerste zicht is zo'n toetsslot heel eenvoudig: het programma wacht gewoon af tot de juiste toetsen in de juiste volgorde ingedrukt worden. Rechts is zo'n programma te zien voor de combinatie 3-5-2. Op het tweede zicht stelt zich bij dit programma echter een probleem: men kan het slot gemakkelijk kraken door 3 keer na elkaar op alle toetsen van 1 tot 6 te drukken. Daarmee heeft men in ieder geval altijd ook de juiste toets ingedrukt. Hoe zei Albert Einstein dat zo mooi: „Men moet de dingen zo eenvoudig mogelijk maken - maar niet gemakkelijker“. Het programma moet dus niet alleen controleren of de juiste toetsen komen, maar ook of een verkeerde toets ingedrukt werd. Het programma ziet er dan zo uit:



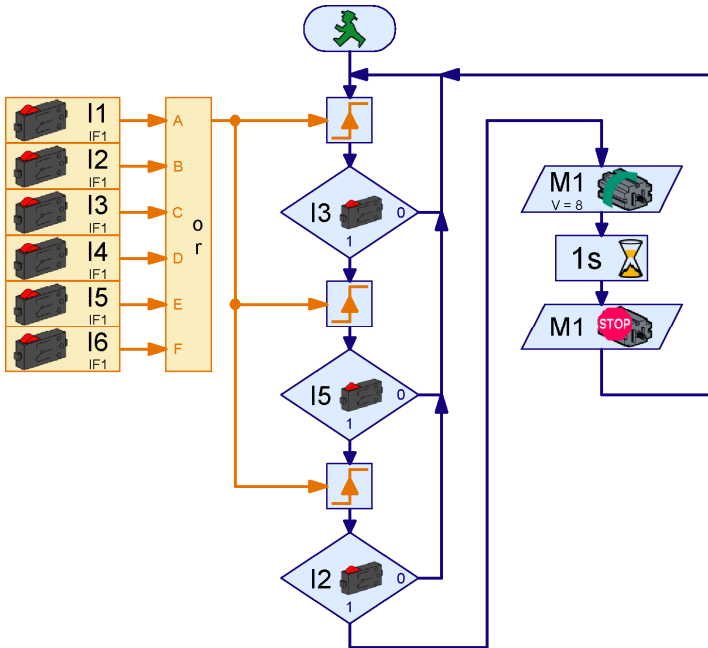
Dit programma opent het slot alleen als de toetsen 3 5 2 ingedrukt worden zonder dat tussendoor enige andere toets ingedrukt wordt. Als bijvoorbeeld drie keer de toets 3 ingedrukt wordt, wacht het programma eerst tot de toets terug losgelaten wordt. Wordt aansluitend

een andere toets dan de toets 5 ingedrukt, dan begint het programma terug van bij het begin. Het programma werkt dus goed, maar eenvoudig en overzichtelijk is het niet. Bovendien is het erg moeilijk om de code te veranderen. Maar wees niet bang, het gaat ook eenvoudig en goed en weliswaar met **Operatoren**. Er zijn verschillende soorten Operatoren. Je vindt ze onder **Programma-elementen** in de groep **Operatoren**. Voor het cijferslot hebben we eerst een **Of Operator** nodig.



Aan de ingangen van de **Of Operators** (In het Engels "or") kan men meerdere signalen aansluiten. Aan de uitgang levert de Operator steeds 1 als minstens één van de ingangen 1 (of groter dan 0) is. Als men aan de ingangen van de **Of Operator** meerdere schakelaars aansluit, is de uitgang van de operator altijd 1 als minstens één van de schakelaars ingedrukt is. Het getal van de ingangen kan via het venster van de eigenschappen van Operator 26 ingesteld worden. Men kan dus ook alle 6 de schakelaars aan een operator aansluiten. Je vraagt je nu misschien af hoe men het cijferslot daarmee kan vereenvoudigen? Dat is heel eenvoudig: met de operator kan je in elke stap eerst wachten tot een of andere toets ingedrukt wordt. Daarna kan je controleren of dat de juiste toets is. Per cijfer heb je dan in plaats van 7 maar 2 programma-elementen meer nodig.





De schakelaars aan de ingangen I1 tot I6 worden via een Of Operator met 6 ingangen samengevat. Als minstens één van de schakelaars ingedrukt is, levert de Of Operator een uitgangswaarde van 1, anders 0. met een **Wachten op Element** wacht het programma tot één van de schakelaars ingedrukt wordt. Aansluitend wordt dan meteen getest of het de juiste schakelaar was. Indien ja, wordt terug gewacht tot de volgende schakelaar ingedrukt wordt. Als een verkeerde schakelaar ingedrukt werd, begint het programma opnieuw.

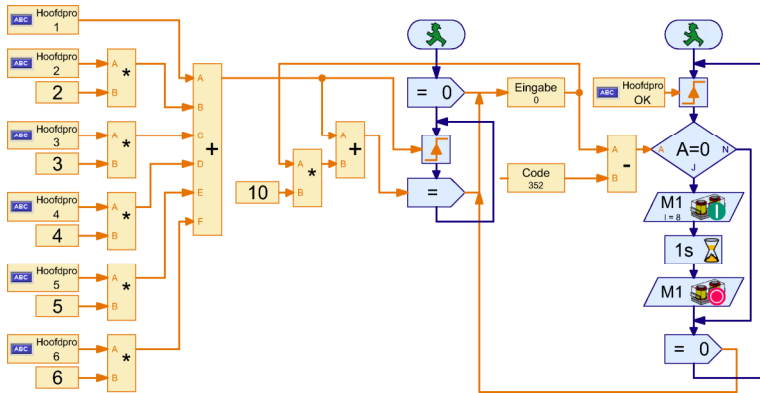


Wijzig het bovenstaande programma zodanig dat het in plaats van schakelaars bedieningselementen in een bedieningsveld gebruikt. Teken daarvoor eerst een bedieningsveld met 6 knoppen met als opschrift 1 tot 6. Wijzig dan de digitale ingangen via het venster van de eigenschappen. De vertakkingen moet je door vertakkingen met de data-ingang en de ingangen van het bedieningsveld vervangen.



Het cijferslot werkt nu onberispelijk, maar het is nog steeds niet zo eenvoudig om de code (3 5 2) te wijzigen. Men moet daarvoor de ingangen in drie vertakkingselementen wijzigen. Voor de klimaatregeling van het museum is niet noodzakelijk om de code regelmatig te veranderen, maar als je het codeslot bijvoorbeeld voor een alarminstallatie gebruikt, wil je de code vermoedelijk regelmatig veranderen. Het zou natuurlijk eenvoudiger zijn als men de code in een variabele zou kunnen opslaan. Dan zou men de code zelfs automatisch kunnen veranderen. Als bijvoorbeeld in de alarminstallatie een stil alarm uitgelokt wordt, zou men de normale code door een bijzondere alarmcode kunnen vervangen.

Opdat je de ingevoerde gegevens met de codevariabelen zou kunnen vergelijken, moet je de ingevoerde gegevens zelf eveneens in een variabele opslaan. Aan het begin moet de variabele met de invoerwaarde de waarde 0 hebben. Als je nu de toets 3 indrukt, moet de variabele de waarde 3 hebben. Na de volgende toetsdruk op de toets 5 de waarde 35 en tot slot na een druk op de toets 2 de waarde 352.



Het cijferslot met codevariabele heeft twee processen. In het linker proces wordt met enkele vermenigvuldigungsoperatoren en een plus-operator aan elke toets een getal toegewezen. Aan toets 1 het getal 1, aan toets 2 het getal 2 en zo verder. De toetsen leveren een waarde van 1 of 0 en als men de waarde met een vast getal  $\times$  vermenigvuldigt, ontstaat een waarde van 0 of  $\times$ . Aangezien de waarden 0 zijn als de toetsen niet ingedrukt zijn, kan men alle waarden optellen en verkrijgt men zo de toetswaarde als getal. Zodra een toets ingedrukt wordt, wordt aan de invoervariabelen 10 keer de vorige waarde plus de waarde van de ingedrukte toets toegewezen. De vermenigvuldiging met 10 schuift de vorige waarde van de ingave-variabele een decimale plaats verder (35 wordt bijvoorbeeld 350).

Het rechter proces wacht zo lang tot na het ingeven van de code de Ok-toets in het bedieningsveld ingedrukt wordt. Dan wordt de codevariabele Code, die bij een juist ingegeven code de waarde 352 heeft, met de invoervariabelen vergeleken. Als beide dezelfde waarde hebben, wordt de openingsmagneet gelost. Anders niet. Tot slot wordt de invoervariabele terug op 0 gezet. De variabele Invoer en Code worden met elkaar vergeleken doordat hun verschil met 0 vergeleken wordt. Je zou ook een vergelijkingselement kunnen gebruiken.



Als je twee toetsen tegelijk indrukt, worden de waarden van de toetsen opgeteld. Als je bijvoorbeeld 2 en 6 gelijktijdig indrukt, ontstaat de waarde 9. Daarmee kan je een super geheim slot bouwen waarbij men soms meerdere toetsen tegelijk moet indrukken. Denk erover na welke toetsen je in welke volgorde moet indrukken, zodat het slot bij een code van 495 opent. Let er daarbij op dat het element **Wachten op...** het programma voortzet als de waarde verhoogt en niet alleen wanneer deze van 0 in 1 verandert.



Werkt het cijferslot ook voor codes met 2 of 4 posities? Zo ja, tot een getal met hoeveel posities werkt het en waarom? Hoe zit het met de andere cijferslotprogramma's?

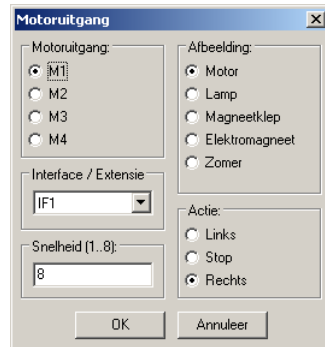
## 6 Uitbreidingsmodule en meerdere interfaces aansturen

Met een ROBO interface of een Intelligent Interface kan men reeds omslachtige modellen besturen. Toch zouden sommigen onder jullie misschien nog iets verder willen gaan. Als je met 8 schakelaars, 4 motoren en 4 analoge ingangen niet toekomt, kan je je ROBO Interface met max 3 **ROBO I/O-uitbreidingen** of je Intelligent Interface met een **Uitbreidingsmodule** uitbreiden. En voor wie zelfs dat niet voldoende is, is er nog de mogelijkheid om in de Online-bedrijfsmodus meerdere interfaces (elk mogelijkwerwijze met uitbreidingsmodules), vanuit een programma aan te besturen. Het aantal in- en uitgangen is minstens in de online-bedrijfsmodus enkel door de capaciteit van je PC begrensd.

### 6.1 Uitbreidingsmodule

Misschien is je de keuzelijst onder **Interface / Uitbreiding** in het venster Eigenschappen voor de in- en uitgangselementen al opgevallen. Daar kan je kiezen aan welke interface of uitbreidingsmodule zich de in- of uitgang bevindt. Voor zover je niets anders ingesteld hebt (zie volgende hoofdstuk) bevat de lijst de volgende elementen:

- **IF1:** Das is de eigenlijke interface (dus geen uitbreidingsmodule)
- **EM1..EM3:** Dat zijn de uitbreidingsmodules 1 tot 3. Als je een Intelligent Interface gebruikt, kan je maar een uitbreidingsmodule gebruiken (**EM1**). De uitbreidingsmodules **EM2** en **EM3** worden echter ook aangeduid als je met een Intelligent Interface werkt, omdat de programma's niet afhankelijk mogen zijn van het feit of je een Robo Interface of een Intelligent Interface gebruikt.



Het is dus heel gemakkelijk om uitbreidingsmodules zoals **ROBO I/O Uitbreidingen** te gebruiken. Je hoeft daarvoor bij de in- en uitgangen enkel de gewenste module (Interface of Uitbreiding) te kiezen.

### 6.2 Meerdere interfaces

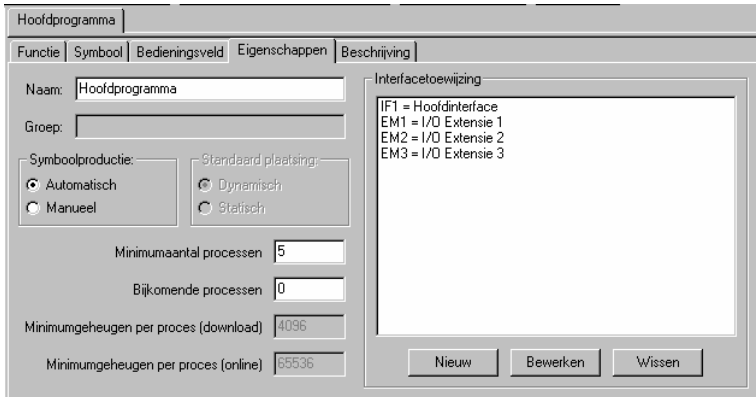
Als je meerdere interfaces vanuit een programma wilt aansturen, werkt dat enkel in de Online-bedrijfsmodus. Je kan bv. een Robo Interface aan COM1 aansluiten, een tweede aan USB. Beide kunnen max. met 3 ROBO I/O-uitbreidingen uitgerust worden. Een andere combinatiemogelijkheid zou bv. bestaan uit 2 ROBO Interfaces aan USB (beide met I/O-uitbreiding) en een Intelligent Interface aan COM1 (graag ook met uitbreidingsmodule). Opdat je in het venster Eigenschappen voor de in- en uitgangen zou kunnen bepalen welke interface aangesproken moet worden, moet je de interface-toewijzing aanpassen.

Zolang je niets anders instelt, vind je in de **Interface / Uitbreiding** selectielijsten voor de in- en uitgangen de invoerwaarden **IF1**, **EM1**, **EM2** en **EM3**. Je kan de lijst echter ook wijzigen of uitbreiden. Daarvoor kunnen er meerdere redenen zijn:

- Om een beter overzicht te hebben, wil je de modules in plaats van IF1 of EM1 een naam geven die aangeeft welk deel van je machine of van je robot de module stuurt.

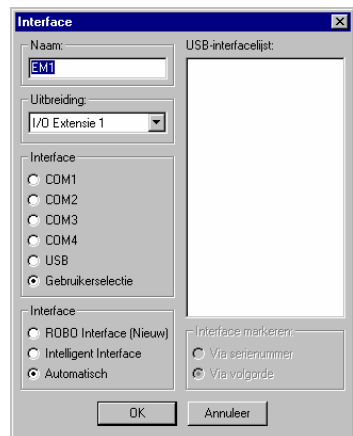
- Je wilt twee uitbreidingsmodules (bijvoorbeeld EM1 en EM2) verwisselen, omdat het met de kabels beter gaat, maar je wilt je programma niet veranderen.
- Je wilt een programma dat- voor een Robo Interface met meer dan een uitbreidingsmodule geschreven werd, met meerdere Intelligent Interfaces laten lopen.
- Je wilt in je programma meer dan een Robo Interface of meer dan een Intelligent Interface gebruiken.

Je kan dat allemaal gemakkelijk doen door de **Interface-toewijzing** in het venster Eigenschappen van het **Hoofdprogramma** te veranderen:



Hier kan je zien welke module (Interface of Uitbreiding) aan de namen IF1 tot EM3 toegewezen zijn. Met de knop **Nieuw** kan je een nieuwe interface toevoegen. Wil je een invoer in de lijst wijzigen, kies je die uit en klik je op **Bewerken**. In beide gevallen wordt het volgende venster weergegeven:

- Onder **Naam** kan je de naam voor de module wijzigen. De naam mag niet te lang zijn omdat de plaats voor de interfacenaam in de grafische symbolen zeer klein is. Als je de naam wijzigt, moet je meestal ook de modulenaam wijzigen in alle in- en uitgangselement die deze naam gebruiken.
- Onder **Uitbreiding** kan je instellen of de naam betrekking heeft op een interface of op een uitbreidingsmodule 1 tot 3.



- Onder **Interface** kan je kiezen op welke plaats de interface aangesloten is. Als je hier **Gebruikersinstelling** aangeeft, wordt de interface gebruikt die je in de toolbalk onder **COM/USB** geselecteerd hebt. Zolang je maar een interface met meerdere uitbreidingsmodules, maar niet meerdere interfaces wilt gebruiken, is dat het gemakkelijkste, omdat op die manier ook iemand anders je programma kan gebruiken zonder iets te wijzigen. Gebruik je meerdere interfaces, dan stel je hier de interface in die aan de betrokken interface aangesloten is.



- Onder **Interface** kan je aangeven of je een Robo Interface of een Intelligent Interface wilt gebruiken. Is de interface met de seriële interface verbonden, dan kan het programma automatisch herkennen om welke interface het gaat (Selectie **Automatisch**).
- Het rechter gedeelte van het venster is alleen belangrijk als je meerdere ROBO Interfaces op de USB-bus aangesloten hebt. Als je onder **Interface** op **USB** klikt, kan je onder **USB Interfacelijst** een van de Interfaces kiezen.

**Opgelet:** Als je meerdere Interfaces op de USB-bus wilt gebruiken, moet je aan elke interface een eigen serienummer toewijzen. Standaard worden alle interfaces met hetzelfde serienummer geleverd om problemen bij het omruilen van interfaces te vermijden. Het Windows besturingssysteem herkent evenwel alleen interfaces met verschillende serienummers. Hierover verneem je meer in het hoofdstuk 6.5 *Wijzigen van het serienummer van de Interface of van de firmwareversie* op pagina 54.

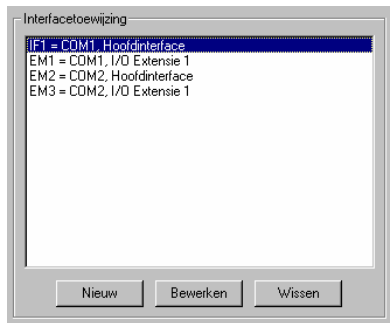
- Onder **Interface herkennen** kan je instellen hoe het programma de geselecteerde interface herkent. Hiervoor zijn er twee mogelijkheden. Als je **Via serienummer** kiest, slaat het programma het serienummer van de interface op. Ook wanneer je verdere interfaces aan de USB-bus aansluit en verwijdert, kan het programma de geselecteerde interface steeds aan de hand van het serienummer vinden. Dat heeft evenwel het nadeel dat het programma enkel nog met een interface met hetzelfde serienummer werkt. Als je een interface met een ander serienummer wilt gebruiken, moet je de interface-toewijzing of het serienummer van de interface veranderen. Om problemen met de serienummers te omzeilen, bestaat de tweede mogelijkheid: **Via volgorde**. Als je dit punt selecteert, slaat het programma in plaats van het serienummer de volgorde van de lijst op. Dat kan weliswaar tot verwarring leiden als je interfaces aan de USB-bus toevoegt of verwijdert, maar het programma werkt ongewijzigd met gelijk welke interface.

### 6.3 Interface-toewijzingen in subprogramma's

Normalerwijze doe je alle interface-toewijzingen voor je programma in het venster Eigenschappen van het hoofdprogramma. Je kan echter ook in subprogramma's interface-toewijzingen ingeven. In het subprogramma kan je dan de interface-toewijzingen van het subprogramma en van het hoofdprogramma gebruiken. Als twee toewijzingen dezelfde naam hebben, heeft de toewijzing in het subprogramma voorrang. Zo kan je bijvoorbeeld vastleggen dat in het hoofdprogramma IF1 naar de hoofdinterface verwijst, maar dat IF1 in een bepaald subprogramma voor een uitbreidingsmodule staat. Dat is zeer praktisch als je een heel machinepark wilt besturen, waarbij elke machine door een eigen interface gestuurd wordt. Je kan dan de besturingen voor de afzonderlijke machines eerst als onafhankelijke programma's ontwikkelen, waarbij elk hoofdprogramma gebruik maakt van IF1. Later kan je alle hoofdprogramma's van de machines in een totaalprogramma als subprogramma invoegen. In het totaalprogramma hoeft je dan enkel nog de interfacetoewijzingen te wijzigen, maar niet de naam in elke individuele in- en uitgang.

## 6.4 Tips en tricks

Als je een programma dat voor een ROBO Interface met 3 uitbreidingsmodules ontwikkeld werd, op 2 Intelligent Interfaces met telkens een uitbreidingsmodule wilt laten werken, kan je de afgebeelde interface-toewijzing gebruiken. De uitbreidingsmodules 2 en 3 worden daarbij door een verdere Intelligent Interface met uitbreidingsmodule aan COM2 vervangen.

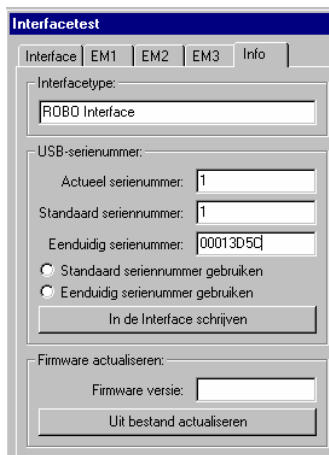


## 6.5 Wijzigen van het serienummer van de Interface of van de firmwareversie

Standaard worden alle ROBO Interfaces en ROBO I/O-uitbreidingen met hetzelfde serienummer geleverd. Zolang je slechts een interface op een computer wilt gebruiken, is dat praktischer, omdat op die manier alle interfaces voor de computer er hetzelfde uitzien en er geen problemen zijn bij het vervangen van interfaces. Als je echter meer dan een interface op een computer via USB wilt gebruiken, moet je vooraf het serienummer van de interface wijzigen, zodat de computer de interfaces kan onderscheiden en kan aanspreken. Als je de interfaces via meerdere seriële interfaces aanspreekt, is dat echter niet nodig.

Om het serienummer van een interface te wijzigen, ga je als volgt tewerk:

- Sluit de interface **afzonderlijk** aan de USB-bus van de computer aan.
- Druk in de toolbalk op de **COM/USB**-knop en selecteer de USB-interface.
- Open het venster voor de Interfacetest via de **Test**-knop in de toolbalk en verander naar de **Info**-ruijer:
- Onder **Interface-type** wordt het type van de interface, bijvoorbeeld **ROBO Interface** of **ROBO I/O Uitbreiding** aangeduid.
- Onder **USB-serienummer** kan je het serienummer instellen die de interface bij het starten gebruikt. Elke interface heeft twee ingebouwde serienummers, een **Standaard serienummer** dat 1 is zolang je niets anders instelt en een **eenduidig serienummer** dat je niet kan wijzigen en dat bij elke interface anders is. De gemakkelijkste weg om meerdere interfaces aan de USB-bus te gebruiken, bestaat erin om bij elke interface de selectieknop op **Eenduidig serienummer gebruiken** om te schakelen. Dan is gegarandeerd dat elke interface een eigen, onverwisselbaar serienummer heeft. Als je veel interfaces voor een model gebruikt, kan dat echter zeer onpraktisch zijn om al die serienummers te onthouden. In dat geval is het eenvoudiger als je de standaard serienummers van je interfaces bijvoorbeeld op 1,2, 3 enz



instelt en deze gebruikt. Nadat je het serienummer gewijzigd of geselecteerd hebt, moet je nog op de knop **In Interface schrijven** indrukken. Nadat je het serienummer gewijzigd hebt, moet je de interface van de stroomvoorziening scheiden en terug aansluiten.

**Opgelet:** Als het serienummer veranderd wordt, moet de driver eventueel opnieuw geïnstalleerd worden. Hiervoor heb je onder Windows NT, 2000 en XP beheersrechten nodig. Als je het serienummer wijzigt, maar de driver niet opnieuw kan installeren omdat je geen beheersrechten hebt, kan je de interface niet meer via USB gebruiken. In dit geval kan je de interface van de stroomvoorziening scheiden en bij het herstarten de **Poort-toets** ingedrukt houden. De interface start dan met het serienummer 1 en wordt terug door de reeds geïnstalleerde driver herkend. Daarbij wordt het serienummer evenwel niet blijvend veranderd. D.w.z. bij de volgende start zonder dat de Poort-toets ingedrukt wordt, zal het vorige serienummer terug ingesteld zijn. Om het serienummer blijvend te veranderen, moet je tewerk gaan zoals hierboven beschreven werd.

- Onder **Firmware actualiseren** kan je tot slot het interne besturingsprogramma van je ROBO Interface actualiseren als fischertechnik eens een nieuwe versie van de Interface-firmware zou aanbieden.

## 7 Overzicht van programma-elementen

Alle programma-elementen die in ROBO Pro ter beschikking staan, worden hierna, volgens elementgroepen geordend, beschreven in de volgorde waarin ze in het elementvenster afgebeeld zijn.

### 7.1 Basiselementen (Niveau 1)

#### 7.1.1 Start



Een proces in een programma begint steeds met een startelement. Als dit programma-element aan het begin ontbreekt, wordt het proces niet afgewerkt. Als een programma meerdere processen bevat, moet elk van deze processen met een startbouwsteen beginnen. De verschillende processen worden dan

gelijktijdig gestart.

Een startelement heeft eigenschappen die je kan veranderen. In tegenstelling tot de meeste andere elementen wordt daarvoor **geen** venster Eigenschappen geopend als je met de rechter muisklik op het element klikt.

#### 7.1.2 Einde



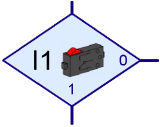
Als een proces beëindigd moet worden, verbindt men de uitgang van het laatste element met het Eindelement. Een proces kan ook op verschillende plaatsen met dit element beëindigd worden. Het is ook mogelijk om uitgangen van verschillende elementen met een enkele eindbouwsteen te verbinden. Maar het

kan ook gebeuren dat een proces als eindeloze lus uitgevoerd wordt en geen eindelement bevat.

Een eindelement heeft geen eigenschappen die je kan veranderen. Daarom wordt in tegenstelling tot de meeste andere elementen **geen** venster Eigenschappen geopend als je met de rechter muisklik op de bouwsteen klikt.



### 7.1.3 Digitale vertakking



Met deze vertakking kan je het stroomschema afhankelijk van de toestand van een van de digitale ingangen I1 tot I8 in twee richtingen besturen. Als bv. een schakelaar aan de digitale ingang gesloten is (=1), vertakt het programma naar de 1-uitgang. Als de ingang daarentegen

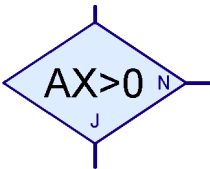
open is (=0), vertakt het programma naar de 0-uitgang.

Als je met de rechter muistoets op het element klikt, wordt het venster Eigenschappen weergegeven:

- Met de knoppen I1 tot I8 kan je ingeven welke ingang van de interface ondervraagd moet worden.
- Onder **Interface / Uitbreiding** kan je selecteren of je een ingang van de interface of een ingang van een uitbreidingsmodule of een twee interface wilt gebruiken. Je verneemt hier meer over in het hoofdstuk 6 *Uitbreidingsmodule en meerdere interfaces aansturen* op pagina 51.
- Onder **Beeld** kan je een beeld selecteren voor de aan de ingang aangesloten sensor. Aan de digitale ingangen worden meestal schakelaars aangesloten, maar ook vaak fototransistoren of reedcontacten.

Onder **1/0 aansluitingen verwisselen** kan je de positie van de 1- en 0-uitgangen van de vertakking verwisselen. Normalerweise staat de 1-uitgang onderaan en de 0-uitgang rechts. Maar het is vaak praktischer dat de 1 uitgang rechts staat. Druk op **1/0 aansluitingen verwisselen**, dan worden de twee aansluitingen verwisseld zodra je het venster met OK sluit.

### 7.1.4 Analoge vertakking



Naast de digitale ingangen heeft de ROBO Interface 6 analoge ingangen, weerstandsingenen AX en AY, 2 spanningsingenen A1 en A2 alsook 2 ingangen voor afstandssensoren D1 en D2. Met deze vertakking kan je de waarde van een analoge ingang met een vast

getal vergelijken en, naargelang de vergelijking opgaat of niet, naar de uitgang Ja (J) of Neen (N) vertakken.

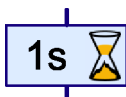
Als je met de rechter muistoets op het element klikt, wordt het venster Eigenschappen weergegeven:

- Onder **Analoge ingang** kan je selecteren welke ingang van de interface ondervraagd moet worden. Alle analoge ingangen leveren een waarde tussen 0 en 1023. Verdere informatie over de verschillende analoge ingangen vind je in het hoofdstuk 7.6.2 *Analoge ingang* op pagina 76.
- Onder **Interface / Uitbreiding** kan je selecteren of je een ingang van de interface of een ingang van een uitbreidingsmodule wilt gebruiken. Je verneemt hierover meer in het hoofdstuk 6 *Uitbreidingsmodule en meerdere interfaces aansturen* op pagina 51.

- Onder **Voorwaarde** kan je een vergelijkingsoperator zoals kleiner dan (<) of groter dan (>) uitkiezen en de vergelijkingswaarde ingeven. De vergelijkingswaarde moet in het bereik tussen 0 en 1023 liggen. Als je een programma met een vertakking voor analoge ingangen in de Online-bedrijfsmodus start, wordt de actuele analoge waarde aangegeven.

Onder **J/N Aansluitingen verwisselen** kan je de positie van de 1- en 0-uitgangen van de vertakking verwisselen. Normalerweise staat de Ja-uitgang (J) onderaan en de Neen-uitgang (N) rechts. Maar het is vaak praktischer dat de Ja-uitgang rechts staat. Druk op **J/N Aansluitingen verwisselen**. Dan worden de Ja- en Neen-aansluitingen verwisseld zodra je het venster met OK sluit.

## 7.1.5 Wachttijd



Met het element **Wachttijd** kan je de verdere uitvoering van een proces met een instelbare tijd vertragen.

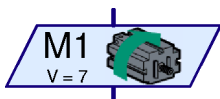
Als je met de rechter muistoets op het element klikt, wordt het venster Eigenschappen weergegeven. Hier kan je de wachttijd in seconden, minuten of uren ingeven. Het instelbereik voor de wachttijd gaat van een milliseconde (dat is een duizendste van een seconde) tot 500 uren (een kleine 3 weken). Maar de tijdmeting wordt onnauwkeuriger naarmate de wachttijd langer is.



De volgende lijst geeft de nauwkeurigheid aan voor de diverse wachttijden:

| Wachttijd       | Nauwkeurigheid |
|-----------------|----------------|
| Tot 30 seconden | 1/1000 seconde |
| Tot 5 minuten   | 1/100 seconde  |
| Tot 50 minuten  | 1/10 seconde   |
| Tot 8,3 uren    | 1 seconde      |
| Tot 83 uren     | 10 seconden    |
| Tot 500 uren    | 1 minuut       |

## 7.1.6 Motoruitgang



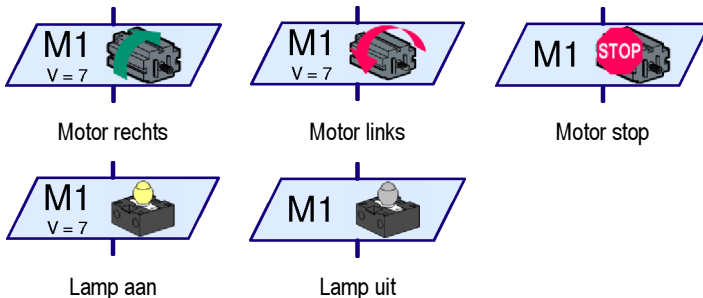
Met het programma-element **Motoruitgang** schakelt men een van de tweepolige uitgangen M1-M4 van de interface. De uitgangen van de interface kunnen zowel voor motoren als voor lampen of elektromagneten gebruikt worden. Bij een motor zou men naast de snelheid ook de draairichting kunnen instellen.

Als je met de rechter muistoets op het element klikt, wordt het venster Eigenschappen weergegeven:

- Onder **Motoruitgang** kan je instellen welke van de vier motoruitgangen **M1** tot **M4** gebruikt moeten worden.
- Onder **Interface / Uitbreiding** kan je selecteren of je een uitgang van de interface of een uitgang van een uitbreidingsmodule of van een tweede interface wilt gebruiken. Meer daarover verneem je in hoofdstuk 6 *Uitbreidingsmodule en meerdere interfaces aansturen* op pagina 51.
- Onder **Beeld** kan je een beeld selecteren dat de aan de uitgang aangesloten module van fischertechnik voorstelt.
- Onder **Actie** stel je in hoe de uitgang beïnvloed moet worden. Een motor kan je rechts of links laten lopen, of laten stoppen. Als je een lamp aan een motoruitgang aansluit (zie tip onder lampuitgang) kan je deze aan- of uitschakelen.
- Tot slot kan je nog een **Snelheid** of **Intensiteit** tussen 1 en 8 aangeven. 8 is de hoogste snelheid, helderheid of magneetkracht, 1 de kleinste. Bij het stoppen of uitschakelen hoeft je natuurlijk geen snelheid aan te geven.

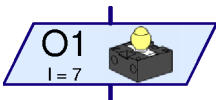


Hierna volgt een overzicht van de symbolen voor enkele acties en beelden:



**Tip:** Soms wordt een motor maar in een richting gedreven, bv. bij een transportband. In dit geval kan je voor de motor een lampuitgang gebruiken, zodat een aansluiting minder gebruikt wordt.

### 7.1.7 Lampuitgang (Niveau2)



Met het programma-element **Lampuitgang** schakelt men een van de eenpolige uitgangen O1-O8 van de interface. De uitgangen van de interface kunnen ofwel paarsgewijs als motoruitgangen (zie boven) of afzonderlijk als lampuitgangen O1-O8 gebruikt worden. De lampuitgangen bezetten in tegenstelling tot de motoruitgang slechts

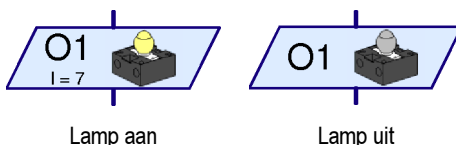
een aansluiten. Daardoor kan je 8 lampen of magneetventielen gescheiden aansturen. De andere aansluiting van de lamp verbind je met de massabus ( $\perp$ ) van de interface.

**Tip:** Als je maar vier lampen of motoren wilt aansluiten, kan je ook voor lampen een motoruitgang gebruiken. Dat is praktischer omdat je zo beide aansluitingen van de lamp aan de interface-uitgang direct kan aansluiten en niet alle minpolen met de massabus moet verbinden.

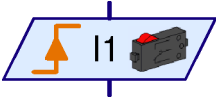
Als je met de rechter muistoets op het element klikt, wordt het venster Eigenschappen weergegeven:

- Onder **Lampuitgang** kan je instellen welke van de acht uitgangen O1 tot O8 gebruikt moet worden.
- Onder **Interface / Uitbreiding** kan je kiezen of je een uitgang van de interface of een uitgang van een uitbreidingsmodule of een tweede interface wilt gebruiken. Je verneemt hierover meer in Hoofdstuk 6 *Uitbreidingsmodule en meerdere interfaces aansturen* op pagina 51.
- Onder **Beeld** kan je een beeld selecteren dat de aan de uitgang aangesloten fischertechnik module voorstelt.
- Onder **Actie** stel je in hoe de uitgang beïnvloed moet worden. Een lamp kan je aan- of uitschakelen.
- Tot slot kan je nog een **Intensiteit** tussen 1 en 8 aangeven. 8 is de hoogste helderheid of magneetkracht, 1 de kleinste. Bij het uitschakelen hoeft je natuurlijk geen intensiteit aan te geven.

Hier zijn de symbolen voor de verschillende acties voor het beeld **Lamp** weergegeven:



## 7.1.8 Wachten op Ingang

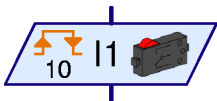


Het element **Wachten op Ingang** wacht tot een ingang een bepaalde toestand heeft of op een bepaalde manier verandert.

Als je met de rechter muistoets op het element klikt, wordt het venster Eigenschappen weergegeven:

- Onder **Wachten op** kan je selecteren op welke manier van verandering of toestand gewacht moet worden. Als je **1** of **0** kiest, wacht het element zo lang tot de ingang gesloten ( **1** ) of open ( **0** ) is. Als je **0 -> 1** of **1 -> 0** selecteert, wacht het element tot de ingangstoestand van open in gesloten (0->1) of van gesloten in open (1->0) **verandert**. Bij de laatste mogelijkheid wacht het element tot de ingang verandert, om het even of dat van open naar gesloten is, of omgekeerd. In het hoofdstuk 3.6 *Verdere programma-elementen* op pagina 22 wordt voor uitgelegd hoe je dit element met het element vertakking kan nabouwen.
- Onder **Digitale ingang** kan je instellen welke van de 8 digitale ingangen van I1 tot I8 ondervraagd moet worden.
- Onder **Interface / Uitbreiding** kan je selecteren of je een ingang van de interface of een ingang van de uitbreidingsmodule of van een tweede interface wilt gebruiken. Je verneemt hierover meer in Hoofdstuk 6 *Uitbreidingsmodule en meerdere interfaces aansturen* op pagina 51.
- Onder **Beeld** kan je een beeld voor de aan de ingang aangesloten sensor selecteren. Aan digitale ingangen worden meestal schakelaars aangesloten, maar ook vaak fototransistoren of reed-contacten.

## 7.1.9 Impulsteller



Veel Fischertechnik robotmodellen gebruiken ook zogenaamde impulstandwielen. Deze tandwielen bedienen een schakelaar bij elke omwenteling 4 keer. Met dergelijke

impulstandwielen kan men een motor in plaats van gedurende een bepaalde tijd ook voor een bepaald aantal omwentelingen inschakelen. Daarvoor moet men het aantal impulsen aan de ingang van de interface tellen. Hiervoor dient element **Impulsteller** dat op een instelbaar aantal impulsen wacht.

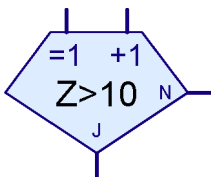


Als je met de rechter muistoets op het element klikt, wordt het venster Eigenschappen weergegeven:

- Onder **Impulstype** kan je selecteren op welke aard van impuls geteld moet worden. Als je 0 -> 1 (toenemend) kiest, wacht het element tot de ingangstoestand zo vaak van open in gesloten (0->1) veranderd is als je aangegeven hebt onder **Aantal impulsen**. Bij 1 -> 0 (dalend) wacht het element tot de ingangstoestand zo vaak van gesloten in open veranderd is als je aangegeven hebt. Met impulstandwielen wordt echter vaker de derde mogelijkheid gebruikt. Hier telt het element zowel de veranderingen 0 -> 1 als de veranderingen 1 -> 0, zodat per omwenteling van een impulstandwiel 8 impulsen geteld worden.
- Onder **Digitale ingang** kan je instellen welke van de 8 digitale ingangen van I1 tot I8 opgevraagd moet worden.
- Onder **Interface / Uitbreiding** kan je kiezen of je een ingang van de interface of een ingang van een uitbreidingsmodule of van een tweede interface wilt gebruiken. Je verneemt hierover meer in hoofdstuk 6 *Uitbreidingsmodule en meerdere interfaces aansturen* op pagina 51.

Onder **Beeld** kan je een beeld voor de aan de ingang aangesloten sensor selecteren. Voor digitale ingangen worden meestal schakelaars gebruikt, maar ook vaak fototransistoren of reedcontacten.

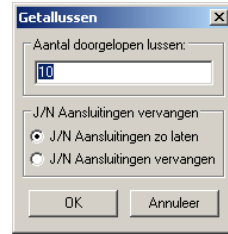
### 7.1.10 Tellus



Met het element **Tellus** kan je heel gemakkelijk een bepaald programmeel meerdere malen laten uitvoeren. Het telluselement heeft een ingebouwde teller. Als de tellus via de =1 ingang betreden wordt, wordt de teller op 1 gezet. Als de tellus daarentegen via de +1 ingang betreden wordt, wordt de teller 1 erbij geteld. Naargelang de teller groter dan een door jou ingegeven waarde is of niet, vertakt de tellus naar de uitgang Ja ( J ) of naar de uitgang Neen ( N ). Een voorbeeld daarvoor vind je in het hoofdstuk 3.6.4 *Tellus* op pagina 24.

Als je met de rechter muistoets op het element klikt, wordt het venster eigenschappen weergegeven:

- Onder **Aantal doorgelopen lussen** geef je in hoe vaak de tellus via de Neen-uitgang ( **N** ) verlaten moet worden, voordat de Ja-uitgang ( **J** ) geactiveerd wordt. De ingegeven waarde moet positief zijn.
- Als je **J/N-aansluitingen verwisselen** aanklikt, worden de J- en de N-aansluiting verwisseld zodra je het venster met OK sluit. Volgens de plaats waar de J- en de N-aansluiting zich bevindt, staat het programmeel dat herhaald wordt, rechts of onder de tellus.



## 7.2 Subprogramma I/O (Niveau 2-3)

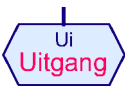
In deze elementengroep vind je programma-elementen die je enkel in subprogramma's nodig hebt.

### 7.2.1 Subprogramma-ingang (Niveau 2)



Een subprogramma kan een of meerdere subprogramma-ingangen hebben. Via deze ingangen geeft het hoofdprogramma of het hoger geplaatste subprogramma de uitvoering door aan het subprogramma. In het groene symbool van het subprogramma dat in het hoger geplaatste programma ingevoegd wordt, wordt voor elke subprogramma-ingang aan de bovenkant een aansluitpin ingevoegd. De aansluitingen in het symbool hebben dezelfde volgorde (van links naar rechts) als de subprogramma-ingangen in het functieplan van het subprogramma. Als je met de rechter muistoets op het element klikt, wordt het venster eigenschappen weergegeven. Daar kan je aan de ingang een naam geven die dan in het symbool weergegeven wordt. Je vindt meer informatie over het thema subprogramma's in het hoofdstuk 4: *Niveau 2: Werken met subprogramma's* op pagina 28.

### 7.2.2 Subprogramma-uitgang (Niveau 2)



Een subprogramma kan een of meerdere subprogramma-uitgangen hebben. Via deze uitgangen geeft het subprogramma de programmeleiding terug aan het hoofdprogramma of het hoger geplaatste subprogramma. In het groene symbool van het subprogramma, dat in het hoger geplaatste programma ingevoegd wordt, wordt voor elke subprogramma-uitgang aan de onderkant een aansluitpin ingevoegd. De aansluitingen in het symbool hebben dezelfde volgorde (van links naar rechts) als de subprogramma-uitgangen in het stroomschema van het subprogramma. Als je met de rechter muistoets op het element klikt, wordt het venster Eigenschappen weergegeven. Daar kan je aan de uitgang een naam geven die dan in het element weergegeven wordt. Je vindt meer over het thema subprogramma's in hoofdstuk 4: *Niveau 2: Werken met subprogramma's* op pagina 28.

### 7.2.3 Subprogramma-bevelingang (Niveau 3)



Via dit element kunnen subprogramma's met ingangselementen zoals schakelaars in het hoofdprogramma of in het hoger geplaatste subprogramma verbonden worden, of kunnen van daar met waarden uit variabele-elementen zoals coördinaten, voorzien worden. In het groene symbool van het subprogramma, dat in het hoger geplaatste programma ingevoegd wordt, wordt voor elke

bevelingang van het subprogramma aan de linkerkant een aansluitpin ingevoegd. De aansluitingen in het symbool hebben dezelfde volgorde (van boven naar beneden) als de bevelingen in het subprogramma. Als je met de rechter muistoets op het element klikt, wordt het venster eigenschappen weergegeven. Daar kan je aan de uitgang een naam geven die dan in het element weergegeven wordt. Het gebruik van dit element wordt uitvoerig uitgelegd in het hoofdstuk 5.5 *Bevelingen voor subprogramma's* op pagina 43.

## 7.2.4 Subprogramma-beveluitgang (Niveau 3)

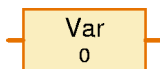


Via dit programma-element kunnen bevelen zoals links, rechts, stop, naar motoren of andere uitgangselementen gestuurd worden die zich in het hoofdprogramma of in het hoger geplaatste subprogramma bevinden. In het groene symbool van het subprogramma, dat in het hoger geplaatste programma ingevoegd wordt, wordt voor elke beveluitgang van het subprogramma aan de rechter kant een aansluitpin ingevoegd. De aansluitingen in het symbool hebben dezelfde volgorde (van boven naar beneden) als de beveluitgangen in het subprogramma. Als je met de rechter muistoets op het element klikt, wordt het venster Eigenschappen weergegeven. Daar kan je aan de uitgang een naam geven die dan in het element weergegeven wordt. Het gebruik van dit element wordt uitvoerig uitgelegd in hoofdstuk 5.5 *Bevelingen voor subprogramma's* op pagina 43.

## 7.3 Variabele, Lijst, ... (Niveau 3)

Die programma-elementen in deze groep kunnen een of meerdere getalwaarden opslaan. Daarmee kan men programma's ontwikkelen die een geheugen hebben.

### 7.3.1 Variabele (globaal)



Een variabele kan een individuele getalwaarde tussen -32767 en 32767 opslaan. De waarde van de variabelen wordt ingesteld doordat men aan de bevelingang aan de linkerkant een = bevelsegment aansluit (zie hoofdstuk 7.4.1 = (Toewijzen) op pagina 70). Men kan aan een variabele via het venster Eigenschappen ook een beginwaarde geven die de variabele behoudt tot ze het eerste bevel voor de verandering van de waarde ontvangt.

ROBO Pro legt voor alle variabele-elementen met **dezelfde naam** en het zelfde **Variabele-type = Globaal** slechts een variabele aan. Alle globale variabelen met dezelfde naam zijn identiek en hebben steeds dezelfde waarde, ook al komen ze in verschillende subprogramma's voor. Als een van deze variabele-elementen via een bevel gewijzigd wordt, veranderen ook alle andere globale variabele-elementen met dezelfde naam. Er zijn ook lokale variabelen (zie volgende hoofdstuk) waarbij dat niet zo is.

Naast het = bevel verstaat een variabele ook het + en het – bevel. Ontvangt een variabele bv. Het bevel + 5, dan telt ze de waarde 5 bij haar actuele waarde. Bij het – bevel wordt de doorgegeven waarde van de actuele waarde van de variabele afgetrokken.

#### Opgelet:

Als de waarde van de variabelen bij een + of – het waardebereik van de variabelen overschrijdt, wordt bij de waarde van de variabelen 65536 opgeteld of afgetrokken, zodat de waarde terug geldig is. Aangezien dit gedrag in de regel niet gewenst is, moet je erop letten dat dit niet gebeurt.

Telkens als de waarde van de variabelen verandert, stuurt ze een = bevel met de nieuwe waarde naar alle elementen die aan de beveluitgang van de variabelen aangesloten zijn. Als je de waarde



van een variabele wilt bekijken, kan je aan de uitgang van de variabelen een bedieningsveld-weergave aansluiten ( zie hoofdstuk 7.6.6 *Bedieningsveld-ingang* op pagina 80 ).

Hier nog eens ter overzicht van alle bevelen die het element variabelen kan verwerken:

| Bevel | Waarde           | Actie  |
|-------|------------------|--|
| =     | -32767 tot 32767 | Zet de waarde van de variabelen op de via het bevel doorgegeven waarde.                        |
| +     | -32767 tot 32767 | Telt bij de actuele waarde van de variabele de waarde die door het bevel werd doorgegeven.     |
| -     | -32767 tot 32767 | Trekt van de actuele waarde van de variabele de waarde af die door het bevel werd doorgegeven. |

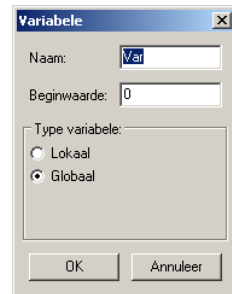
De curve van het waardebereik van -32767 tot 32767 wordt trouwens veroorzaakt doordat de computer in het binair systeem rekt en niet in een tientallig systeem zoals wij. In het binair systeem is 32767 een getal zoals 9999 in het tientallig systeem. Maar daar hoeven wij niet voor te zorgen, aangezien de computer alle getallen van het binair in het tientallig systeem omrekent. Alleen bij maximale waarde van variabelen merkt men daar toch iets van en wanneer het bij het rekenen tot een overloop komt.

### Het venster Eigenschappen voor Variabelen

- Onder **Naam** kan je een Naam voor de Variabele ingeven. De naam heeft geen enkel technisch belang. In het bijzonder zijn variabelen met dezelfde naam niet identiek. Voor elk variabelelement in je programma zal een eigen variabele aangelegd worden, ook wanneer twee variabelen dezelfde naam hebben.

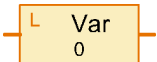
Toch zou je geen variabelen met dezelfde naam in een programma mogen gebruiken, aangezien dit gedrag in een later versie zou kunnen veranderen.

- Onder **beginwaarde** kan je de beginwaarde voor de variabele ingeven. De variabele behoudt deze waarde tot ze via een =, + of – bevel een nieuwe waarde ontvangt.



Het punt **Variabele-type** is enkel interessant voor variabelen in subprogramma's en wordt in het volgend hoofdstuk „Lokale variabelen“ nader uitgelegd. Bij variabelen in het hoofdprogramma hebben beide instellingen hetzelfde effect.

### 7.3.2 Lokale variabele




Alle **globale** variabelen met dezelfde naam gebruiken een en dezelfde variabele en hebben steeds dezelfde waarde. Dit is waarschijnlijk wat je verwacht en wat in de regel ook praktisch is. Als je echter variabelen in subprogramma's gebruikt, kan dat tot grote problemen leiden. Als je

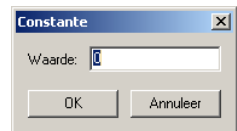
programma meerdere parallele processen heeft, kan een subprogramma op een bepaald tijdstip meerdere keren uitgevoerd worden. In een dergelijke situatie leidt het in de regel tot chaos als het subprogramma in alle processen dezelfde variabele gebruikt. Om die reden zijn er **lokale variabelen**. Een lokale variabele gedraagt zich bijna zoals een globale variabele, met een verschil: de lokale variabele geldt enkel in het subprogramma waarin ze gedefinieerd is. Zelfs als twee

lokale variabelen in verschillende subprogramma's dezelfde naam hebben, zijn het verschillende, onafhankelijke variabelen. Ook wanneer een subprogramma door meerdere processen meervoudig parallel uitgevoerd wordt, heeft het subprogramma in elk proces een onafhankelijke set van lokale variabelen. Lokale variabelen bestaan enkel zo lang tot het subprogramma waarin ze gedefinieerd zijn, uitgevoerd wordt. Daarom wordt de beginwaarde van de lokale variabelen niet bij de start van het programma, maar bij elke start van het betrokken subprogramma toegewezen. Aangezien een subprogramma bij meerdere oproepen in de regel steeds hetzelfde moet doen, is het veel praktischer als de variabelen bij elke oproeping op de beginwaarde gezet worden. Lokale variabelen hebben zo te zeggen geen herinnering aan vorige oproepingen van hetzelfde subprogramma.


In het hoofdprogramma gedragen lokale variabelen en globale variabelen zich hetzelfde, aangezien het Globaal programma en het hoofdprogramma gelijktijdig gestart worden. Lokale variabelen zijn echter wat efficiënter in de programma-uitvoering. Lijstelementen moet men daarentegen eerder globaal definiëren, omdat het geheugenbereik voor globale variabelen groter is dan voor lokale variabelen.

### 7.3.3 Constanten

 Een constante heeft zoals een variabele een waarde, maar de waarde kan niet door het programma gewijzigd worden. Een constante kan je met een data-ingang van een subprogramma-symbool verbinden als het subprogramma steeds dezelfde waarde moet gebruiken. Ook bij berekeningen met operatoren zijn constanten zeer praktisch. Een voorbeeld daarvoor vind je aan het einde van het hoofdstuk 5.7 *Operatoren* op pagina 47.

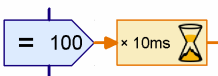


### 7.3.4 Timer-variabele

 Een timer-variabele gedraagt zich in feite precies hetzelfde als een variabele. Ook het verschil tussen een normale en statische variabele bestaat bij timer-variabelen. Het enige verschil is dat een timer-variabele de opgeslagen waarde in een vast tijdsinterval tot 0 aftelt. Zodra de timerwaarde op 0 staat, blijft de timer-variabele staan. Als de timerwaarde bv. door een min-bevel negatief wordt, wordt de waarde bij de volgende tijdstap terug 0.

De snelheid waarmee de timer-variabele aftelt, kan je tussen 1/1000 seconde per stap en 1 minuut per stap in het venster Eigenschappen instellen. Je moet er daarbij opletten dat de nauwkeurigheid van de timer van de ingestelde tijdstappen afhangt. Als je bijvoorbeeld een timer op 1 x 10 seconden instelt, kan de volgende tijdstap van 10 seconden korte tijd (bv. reeds na een seconde) of pas na 10 seconden gebeuren. Timers zijn altijd maar even nauwkeurig als de ingestelde tijdstappen. Daarom moet je eerder kleine tijdstappen kiezen, bijvoorbeeld 10 x 1 seconde of 100 x 0,1 seconde in plaats van 1 x 10 seconden. Een tijdstap van een minuut moet je alleen kiezen als het programma minstens een uur moet wachten. Dan zal het op een minuut meer of minder niet meer aankomen.

Het aantal stappen dat afgeteld moet worden, wordt aan de timervariabele in de regel via een =-bevel door een bevelement toegewezen. In het afgebeeld voorbeeld worden 100 stappen van telkens 10ms afgeteld. Dat komt overeen met een tijdduur van 1000 ms=1 seconde. De nauwkeurigheid bedraagt daarbij 10ms.



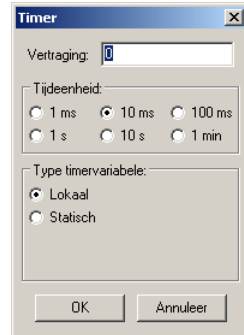
Met Timer-variabelen kan men zeer eenvoudig ook moeilijke taken voor tijdmetingen en vertragingen oplossen. als een robot bijvoorbeeld een zoektocht na 20 seconden moet afbreken, kan je aan het begin van de zoektocht een timervariabele op 20 x 1 seconde (of 200 x 0,1 seconde) instellen en dan in het zoekprogramma regelmatig opvragen of de timerwaarde nog groter is dan 0. Je kan ook bij gedeeltelijke successen bij het zoeken de timer terug op de beginwaarde plaatsen.

Als je een tijd wilt meten, moet je de timer-variabele aan het begin op een zo groot mogelijke positieve waarde ( 30000 of 32767 ) instellen, zodat er veel tijd resteert tot de timerwaarde 0 is. Als je wilt weten, hoeveel tijd er sindsdien verstreken is, trek je de actuele timerwaarde van de beginwaarde af.

### Het venster Eigenschappen voor timer-variabelen

- Onder **Vertraging** kan je de beginwaarde voor de timervariabele vastleggen. In de regel geeft men hier 0 in zet men de waarde voor de timervariabele met een = bevel in op het passend tijdstip. Als de timer echter ook bij de start van het programma of van een subprogramma moet beginnen lopen, kan hier de overeenkomstige waarde ingegeven worden.
- Onder **tijdeenheid** kan je de omvang van de tijdstappen kiezen waarin de timer-variabele afgeteld wordt.

Onder **Type timervariabele** kan je instellen of de timer een globale of lokale variabele is (zie hoofdstuk 7.3.2 *Lokale variabele* op pagina 65).






### 7.3.5 Lijst



Het element **Lijst** komt overeen met een variabele waarmee men niet alleen een waarde, maar meerdere waarden kan opslaan. Het maximaal aantal waarden die in een lijst opgeslagen kunnen worden, wordt in het venster eigenschappen van het element vastgelegd.

Je kan aan de lijst achteraan waarden toevoegen en waarden aan het einde van de lijst verwijderen. Bovendien kan je een willekeurige waarde in de lijst veranderen of uitlezen en een willekeurige waarde in de lijst door de eerste waarde in de lijst vervangen. Een waarde in het midden of aan het begin van de lijst invoegen of wissen kan men niet direct. Men kan echter een overeenkomstig subprogramma schrijven dat deze functies uitvoert

De volgende functies van een lijst worden gebruikt door naar het element aan de ingang **S** (voor Schrijven ) bevelen stuurt. Naar de **S**-ingang kan men de volgende bevelen besturen:

| Bevel  | Waarde              | Actie  |
|--|---------------------|--|
| Toevoegen<br>   | -32767<br>tot 32767 | Voegt de waarde die door het bevel doorgegeven wordt, toe aan het einde van de lijst. De lijst wordt met een element verhoogd. Als de lijst reeds de maximale grootte heeft, wordt het bevel genegeerd.  |
| Verwijderen<br> | 0 tot<br>32767      | Wist het aangegeven aantal elementen aan het einde van de lijst. De waarde die door het bevel doorgegeven wordt, is het aantal te wissen elementen. Als het aantal groter is dan het aantal elementen in de lijst, worden alle elementen in de lijst gewist. Is het aantal 0 of negatief, dan wordt het bevel genegeerd. |
| Vervangen<br>   | 0 tot<br>32767      | Vervangt het aangegeven element door het eerste element in de lijst. De waarde die door het bevel doorgegeven wordt, is de nummer van het element dat vervangen moet worden.   |

Via de ingang I ( voor Index ) kan een bepaald element uit de lijst geselecteerd worden. Daarvoor stuurt men naar de I-ingang een =bevel met het gewenst elementnummer. Het eerste element heeft daarbij nummer 1. Aan het element dat via de I-ingang geselecteerd werd, kan men een nieuwe waarde toewijzen door naar de S-ingang een =bevel te besturen met de gewenste nieuwe waarde.

Het via de I-ingang geselecteerde element kan via de R-uitgang (uitgang voor het teruglezen) ondervraagd worden. Wordt de I-ingang of de waarde van de invoer die via de I-ingang geselecteerd werd, gewijzigd, dan stuurt de lijst de actuele waarde van de geselecteerde invoer naar de elementen die aan de R-uitgang aangesloten zijn.

Via de I-uitgang kan men opvragen of de aan de I-ingang aangelegde index geldig is. Als N het aantal elementen is, moet aan de I-ingang een waarde tussen 0 en N-1 aangelegd worden. is dat het geval, dan stuurt de I-uitgang een =bevel met waarde 1, in het andere geval met waarde 0 naar alle aangesloten elementen.

## Het venster Eigenschappen voor lijsten

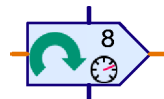
- Onder **Maximumaantal** kan je het maximaantal elementen van de lijst ingeven. Dit aantal kan door **Toevoegbevelen** niet overschreden worden.
- Onder **beginnaantal** geeft je het aantal elementen in waarvan de lijst bij de start voorzien moet zijn.
- Onder **Lijst van de beginwaarden** kan je de beginwaarden aangeven waarvan de lijst voorzien moet zijn. Met de knoppen rechts naast de lijst kan je de lijst bewerken.
- Onder **Uit .CSV bestand laden** kan je een Excel-compatibel .CSV bestand kiezen waaruit de lijst zijn waarden overneemt. In het selectieveld in het midden kan je de kolom van het .CSV bestand kiezen die daarvoor gebruikt moet worden.

Het bestand wordt onmiddellijk geladen en onder **Lijst van de beginwaarden** aangegeven. Als je het programma start of een download uitvoert, probeert ROBO Pro nogmaals de actuele waarde uit het bestand te laden. Lukt dat niet, dan worden de onder de Lijst van de beginwaarden opgeslagen waarden gebruikt.

- Onder **In .CSV -bestand opslaan** kan je een bestand aangeven waarin de inhoud van de lijst na afloop van het programma opgeslagen moet worden. Dat werkt alleen in de Online-bedrijfsmodus en enkel voor statische lijsten (zie volgende punt). De inhoud van de lijst wordt in de geselecteerde kolom van het bestand geschreven. **Scheidingsteken voor kolom** kan je kiezen of je de afzonderlijke kolommen in de lijst met een komma of met kommapunten gescheiden moeten worden. In landen waar men 0.5 met een punt schrijft, wordt in de regel een komma als scheidingsteken voor de kolommen gebruikt. Aangezien men in Duitsland 0,5 met een komma schrijft, wordt in Duitsland ook vaak een kommapunt als scheidingsteken voor de kolommen gebruikt. Als je bij de import van een .CSV-bestand van ROBOT Pro bijvoorbeeld in Microsoft Excel problemen hebt, probeer dan een ander scheidingsteken voor de kolommen te gebruiken.
- Onder **Type van lijstvariabelen** kan je instellen of de lijst Timer een globale of een lokale variabele is (zie hoofdstuk 7.3.2 *Lokale variabele* op pagina 65). Voor grote lijsten (maximumaantal meer dan 100 elementen) is het type **Globaal** aan te raden omdat voor globale variabelen meer geheugen beschikbaar is dan voor lokale variabelen.

## 7.4 Bevelen (Niveau 3)

Alle programma-elementen in deze groep zijn bevelementen. Volgens de toepassing worden ze ook als nieuwselementen bestempeld. Als een bevelement uitgevoerd wordt (d.w.z. wanneer het stroomschema bovenaan in de blauwe ingang van het element binnengaat), stuurt het bevelement een bevel of een nieuws naar het element dat rechts aan zijn uitgang



aangesloten is. Er zijn verschillende bevelen zoals rechts, links of stop, die verschillende effecten op het aangesloten element hebben. In de regel verstaan de aangesloten elementen maar weinig bevelen. Bij de verschillende programma-elementen wordt opgesomd welke bevelen ze verstaan en welk effect de bevelen hebben. De meeste bevelen worden nog door een waarde begeleid. Bij een **Rechts**-bevel geeft men bijvoorbeeld nog een snelheid tussen 1 en 8 aan. Een **stop**-bevel heeft daarentegen geen bijkomende waarde.

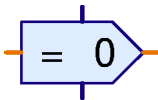
### Het venster Eigenschappen voor bevelementen

- Onder **Bevel** kan je uit een lijst van alle mogelijke bevelen de gewenste bevelen uitkiezen.
- Onder **Waarde** geef je de getalwaarde aan die met een bevel doorgegeven moet worden. Als geen waarde doorgegeven moet worden, blijft dit veld leeg.
- Onder **Waardebenaming** kan je een korte aanwijzingstext ingeven die samen met de waarde in het bevelement aangegeven wordt (bv. X= of T=). De aanwijzing moet verduidelijken om welke soort waarde het gaat. Dit dient enkel als commentaar en heeft geen enkele functie.



Onder **Data-ingang voor bevelwaarde** kan je aangeven of het bevelement aan zijn linkerkant een oranje data-ingang voor de door te geven waarde moet hebben. De waarde kan bij alle bevelementen ofwel direct ingegeven worden in het bevelement, of kan via een data-ingang aan de linkerkant van het bevel direct ingelezen worden. Daardoor kan men bijvoorbeeld een motor in een regelkring met een veranderlijke snelheid aansturen.

### 7.4.1 = ( Toewijzen )

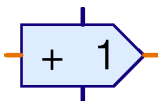


Het =-bevel wijst aan de ontvanger een waarde toe. In de regel wordt dit gebruikt om aan variabelen, timer-variabelen, lijstelementen of bedieningsveld-uitgangen een waarde toe te wijzen.

Het =-bevel wordt echter niet door bevelementen gestuurd, maar door alle programma-elementen met data-uitgangen. Alle elementen versturen =-bevelen, als de waarde van een uitgang verandert. Een digitaal ingangselement verstuurt bijvoorbeeld een 1-bevel wanneer een schakelaar aan de ingang gesloten wordt en een = 0-bevel wanneer de schakelaar geopend wordt. Daarbij wordt evenwel geen bevelement gebruikt. In programma-elementen met data-uitgangen zijn zo te zeggen =-bevelementen ingebouwd.

Alle data-ingangen van ROBO Pro programma-elementen kunnen minstens het =-bevel verwerken. Het =-bevel is daarmee het meest gebruikt bevel in ROBO Pro.

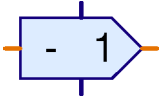
### 7.4.2 + ( Plus )



Het +-bevel wordt naar variabelen of timer-variabelen gestuurd om de waarde van de variabelen te verhogen. Met het +-bevel kan een willekeurige waarde doorgegeven worden die bij de variabelen opgeteld wordt. Aangezien de door het bevel doorgegeven waarde ook negatief kan zijn, kan de waarde van de variabele daardoor ook verminderen. Zie hoofdstuk **Fehler!**

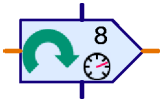
**Verweisquelle konnte nicht gefunden werden.** *Variabele (globaal)* op pagina 64 en Hoofdstuk 7.3.4 *Timer-variabele* op pagina 66.

### 7.4.3 – ( Min )



Het **- bevel** wordt op een gelijkaardige manier gebruikt als het hiervoor beschreven **+ bevel**. Het enige verschil is dat de waarde die door het bevel doorgegeven wordt, van de variabele afgetrokken wordt.

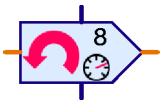
### 7.4.4 – Rechts



Het bevel **Rechts** wordt naar motoruitgangelementen gestuurd om de motor met rechtse draairichting in te schakelen. Zie hoofdstuk 7.6.4 *Motoruitgang* op pagina 78.

De waarde is een snelheid van 1 tot 8.

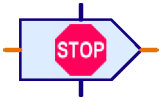
### 7.4.5 – Links



Het bevel **Links** wordt naar motoruitgangelementen gestuurd om de motor met linkse draairichting in te schakelen. Zie hoofdstuk 7.6.4 *Motoruitgang* op pagina 78.

De waarde is een snelheid van 1 tot 8. .

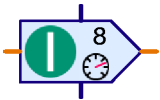
### 7.4.6 – Stop



Het bevel **Stop** wordt naar motoruitgangelementen gestuurd om de motor te stoppen. Zie hoofdstuk 7.6.4 *Motoruitgang* op pagina 78.

Met het **Stop**-bevel wordt geen waarde doorgegeven.

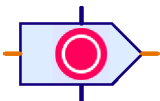
### 7.4.7 – Aan



Het bevel **Aan** wordt naar lampuitgangelementen gestuurd om de lamp aan te doen. Zie hoofdstuk 7.6.5 *Lampuitgang* op pagina 79. Een **Aan**-bevel kan ook naar motoruitgangelementen gestuurd worden. Het komt overeen met het bevel **Rechts**. Voor Motoren is het evenwel beter het bevel **Rechts** te gebruiken omdat de draairichting direct herkenbaar is.

De waarde is de helderheid of de intensiteit van 1 tot 8.

### 7.4.8 – Uit



Het bevel **Uit** wordt naar lampuitgangelementen gestuurd om de lamp uit te schakelen. Zie hoofdstuk 7.6.5 *Lampuitgang* op pagina 79. Een **Uit**-bevel kan ook naar motoruitgangelementen gestuurd worden en komt overeen met het bevel **Stop**.

Met het **Uit**-bevel wordt geen waarde doorgegeven.

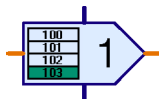
### 7.4.9 – Tekst



Het bevel **Tekst** is een bijzonder bevel, aangezien het geen bevel met een getal, maar een willekeurige tekst naar het aangesloten element stuurt. Er is evenwel maar een programma-element dat het bevel Tekst kan verwerken, namelijk een tekstweergave in een bedieningsveld. Je vindt hierover verdere informatie in het hoofdstuk 8.1.2 *Tekstweergave*

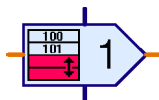
op pagina 84.

### 7.4.10– Waarde toevoegen



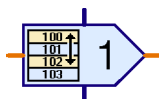
Het bevel **Toevoegen** is een speciaal bevel van lijstelementen. Zie hoofdstuk 7.3.5 *Lijst op pagina 67*. Met het bevel wordt een waarde doorgegeven die aan het einde van de lijst toegevoegd wordt. Als de lijst reeds vol is, wordt het bevel genegeerd.

### 7.4.11– Waarde(n) verwijderen



Het bevel **Verwijderen** is een speciaal bevel van lijstelementen. Zie hoofdstuk 7.3.5 *Lijst op pagina 67*. Met het bevel kan een willekeurig aantal elementen aan het einde van een lijst gewist worden. Het gewenste aantal wordt met het bevel als waarde doorgegeven. Is de doorgegeven waarde groter dan het aantal elementen in de lijst, dan worden alle elementen in de lijst gewist. Om een lijst volledig te wissen, kan men een bevel **Verwijderen** met de maximum mogelijke waarde van 32767 naar een lijstelement besturen.

### 7.4.12 – Waarde vervangen



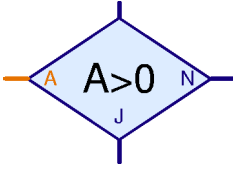
Het bevel **Vervangen** is een speciaal bevel voor lijstelementen. Zie hoofdstuk 7.3.5 *Lijst op pagina 67*. Met het bevel kan een willekeurig element van een lijst door het eerste element van de lijst vervangen worden. Het nummer van het element waardoor het eerste element vervangen wordt, wordt door het bevel als waarde doorgegeven. **Belangrijk:** het eerste element van een lijst heeft het nummer 1. Als de doorgegeven waarde geen geldig elementnummer is, wordt het bevel door het lijstelement genegeerd.

## 7.5 Vergelijkingen, wachten op, ... (Niveau 3)

De programma-elementen in deze groep dienen allemaal voor de programmavertakking of voor de vertraging van het programmaverloop.



### 7.5.1 Vertakking ( met data-ingang)



Deze programmavertakking heeft een oranje data-ingang **A** links aan het element. Daardoor wordt een waarde ingelezen die vaak van een ingangselement (zie hoofdstuk 7.6.1 tot 7.6.6

vanaf pagina 75 ) komt. De data-ingang **A** kan echter ook met de data-uitgangen van variabelen, timer-variabelen of operatoren (zie hoofdstuk 7.7 **Operatoren op pagina Fehler!**

**Textmarke nicht definiert.**) verbonden worden. De waarde aan de data-ingang **A** wordt door het element met een vaste, maar vrij definieerbare waarde vergeleken. Naargelang de vergelijking al dan niet opgaat, vertakt het element naar de **J**- of de **N**-uitgang.

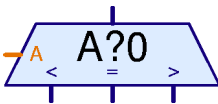


#### Het venster Eigenschappen voor die vertakking

- Onder **Voorwaarde** vul je in het rechterveld de waarde in waarmee de ingangswaarde **A** vergeleken moet worden. Voor de vergelijking staan alle gebruikelijke vergelijkingsoperatoren ter beschikking.
- Als je selecteert om J/N-aansluitingen te verwisselen, worden de **J**- en **N**-uitgang met elkaar verwisseld zodra je het venster Eigenschappen met Ok sluit. Om de J/N-aansluitingen terug op hun uitgangspositie te brengen, kan je ze nogmaals verwisselen.

De meest gebruikte vergelijking is **A > 0**. Dat betekent dat het stroomschema aan de **J**-uitgang vertakt als de aan data-ingang **A** aangelegde waarde groter is dan 0. Daarmee kunnen bijvoorbeeld digitale ingangen afgelezen worden die een 1 of 0 leveren. Maar ook timer-variabelen en vele andere waarden kunnen met de vergelijking **A > 0** zinvol geïnterpreteerd worden.

### 7.5.2 Vergelijking met vaste waarde

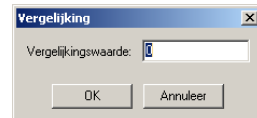


Met het programma-element **Vergelijking met vaste waarde** kan de waarde aan de data-ingang **A** met een vaste, maar vrij definieerbare waarde vergeleken worden. Naargelang de aangelegde waarde aan de Data-ingang **A** gelijk is aan of groter of kleiner is dan de vaste waarde, vertakt dit vergelijkingselement naar de rechtse, de linkse of

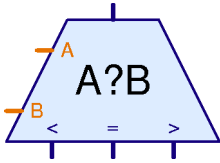
de middelste uitgang. In de regel wordt aan de Data-ingang **A** de uitgang van een variabele of een lijst aangesloten. Het vergelijkingselement kan door twee vertakkingselementen vervangen worden. Het is echter in vele gevallen overzichtelijker wanneer men slechts één element nodig heeft.

#### Het venster Eigenschappen voor Vergelijking

- Onder Vergelijkswaarde kan je de constante waarde ingeven waarmee de ingang **A** vergeleken moet worden.



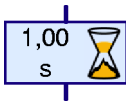
### 7.5.3 Vergelijking



Met het programma-element **Vergelijking** kunnen de aan beide data-ingangen A en B aangelegde waarden met elkaar vergeleken worden. Naargelang A kleiner is dan B, A groter dan B of A gelijk aan B, vertakt het element dan naar de linkse, de rechtse of de middelste uitgang. De meest gebruikte toepassing hiervan is de vergelijking tussen een streefwaarde en een reële waarde. Naargelang de verhouding van de streefwaarde tegenover de reële waarde, kan een motor bijvoorbeeld links of rechts draaien of gestopt worden.

Het programma-element **Vergelijking** heeft geen instelmogelijkheden en heeft bijgevolg geen venster met Eigenschappen.

### 7.5.4 Wachtijd



Met dit element kan men een **Wachtijd** in een stroomschema programmeren. De wachtijd begint als het element in het schema aan de beurt is. Zodra de ingegeven wachtijd voorbij is, wordt het schema voortgezet. Zie ook hoofdstuk **Fehler! Verweisquelle konnte nicht gefunden werden.** *Wachtijd* op pagina **Fehler! Textmarke nicht definiert..**

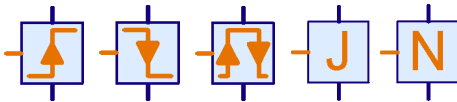
Het venster Eigenschappen voor Wachtijd:

- Onder **Tijd** kan je de wachtijd ingeven. Je kan ook kommagetallen zoals 1,23 gebruiken.
- Onder **Tijdeenheid** kan je als tijdeenheid seconde, minuut of uur kiezen. De tijdeenheid heeft, anders dan bij een timervariabele, geen invloed op de nauwkeurigheid van de wachtijd. Een wachtijd van 60 seconden en een wachtijd van 1 minuut gedragen zich precies hetzelfde.



In de expertmodus (Niveau 5) wordt een uitgebreid venster Eigenschappen weergegeven dat meer lijkt op het venster Eigenschappen voor Timervariabelen.

### 7.5.5 Wachten op...



Het programma-element **Wachten op...** vertraagt de uitvoering van het programma tot aan de data-ingang van het element een verandering plaatsgevonden heeft, of een bepaalde toestand ingetreden is. Het

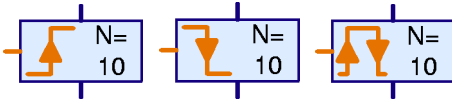
element bestaat in vijf uitvoeringen: het element links wacht zo lang tot de waarde aan de ingang groter geworden is. Daarbij tellen niet alleen de veranderingen van 0 naar 1, maar elke vergroting, bijvoorbeeld ook van 2 naar 3. Het tweede element wacht zo lang tot de waarde aan de ingang kleiner geworden is en het element in het midden wacht op één of andere verandering, om het even in welke richting. Het derde element wordt vaak voor impulstandwielen gebruikt. Het vierde en vijfde element wachten niet op een verandering, maar tot zich aan de ingang een toestand Ja (>0) of Neen (<=0) voordoet. Als de betrokken toestand reeds voorhanden is, wacht het element niet. De eerste drie elementen wachten daarentegen altijd tot een verandering aan de ingang herkend wordt.

## Het venster Eigenschappen voor Wachten op verandering

- Onder **Veranderingswijze** kan je tussen de vijf hierboven beschreven functiewijzen schakelen.
- Als de knop **Wijzigingen herkennen indien niet actief** ingedrukt is, herkent het element ook veranderingen die zich voorgedaan hebben als het element in het stroomschema niet eens aan de beurt was. In dit geval slaat het element de laatste bekende waarde op. Als het element dan opnieuw uitgevoerd wordt, zet het de uitvoering van het programma voort als de waarde intussen op de juiste manier gewijzigd werd. Op die manier is de waarschijnlijkheid kleiner dat men een wijziging mist omdat het programma net iets anders gedaan heeft.



## 7.5.6 Impulsteller



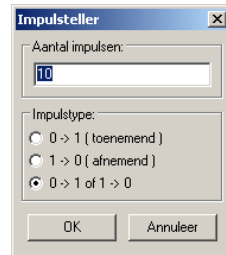
Dit programma-element wacht op een instelbaar aantal impulsen aan de data-ingang aan de linkerkant voor de uitvoering van het programma voortgezet wordt. Dat is voor eenvoudige positioneertaken met

impulstandwielen erg praktisch. Voor omslachtigere positioneringen, bv. met een veranderlijke waarde, moeten dan subprogramma's met variabelen gebruikt worden.

## Het venster Eigenschappen voor impulstellers

- Onder **Aantal impulsen** geef je het aantal impulsen in waarop gewacht moet worden tot de programma-uitvoering voortgezet wordt.
- Onder **Impulstype** kan je kiezen tussen de drie impulstypes 0-1, 1-0 of willekeurig.

De mogelijkheid om wijzigingen te herkennen als het element niet actief is, zoals bij het gewoon **Wachten op...** element, bestaat bij dit element niet



## 7.6 Interface-Ingangen / -Uitgangen, ...

Deze groep van programma-elementen bevat alle in- en uitgangselementen. Hoe deze elementen gebruikt worden, is uitgelegd in hoofdstuk 5 *Niveau 3: Variabelen, bedieningsvelden & Co* op pagina 36.

### 7.6.1 Digitale ingang

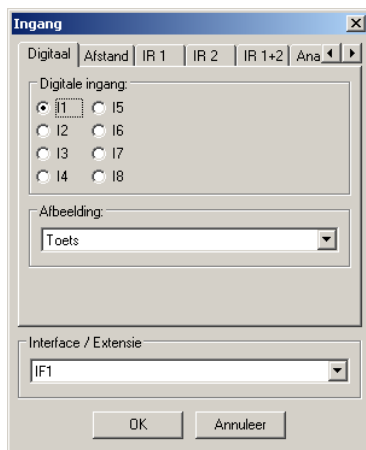


Via het element **Digitale ingang** kan men de waarde van de digitale ingangen I1 tot I8 van de interface ondervragen. Zijn de beide bij de ingang horende bussen aan de interface met elkaar verbonden, dan levert het digitale ingangselement aan zijn oranje aansluiting een waarde van 1, in

het andere geval een waarde van 0.

### Het venster Eigenschappen voor Digitale ingangen:

- Onder **Digitale ingang** kan je kiezen welke ingang van de interface gebruikt moet worden. Ingangen van uitbreidingsmodules selecteer je onder **Interface / Uitbreiding**.
- Onder **Beeld** kan je een beeld van de sensor selecteren die aan de ingang aangesloten is. In de meeste gevallen zal dat een **Schakelaar** zijn. Een **reedcontact** is een schakelaar die op magneetvelden reageert. Er kan ook een **fototransistor** aan een digitale ingang aangesloten worden, hoewel hij eigenlijk een analoge sensor is. Aan een digitale ingang aangesloten kan je de fototransistor samen met een lenslamp als lichtbarrière gebruiken die ofwel onderbroken (= 0) of gesloten (= 1) is. Als je de fototransistor daarentegen aan een *Analoge ingang* aansluit, kan je veel trappen van licht en donker onderscheiden.
- Onder **Interface / Uitbreiding** kan je kiezen of je een ingang van de interface of een ingang van een uitbreidingsmodule of een tweede interface wilt gebruiken. Je verneemt hierover meer in hoofdstuk 6 *Uitbreidingsmodule en meerdere interfaces aansturen* op pagina 51.



Nader bekeken, is er voor alle soorten van ingangen maar één programma-element. Via de ruiters boven in het venster Eigenschappen kan je het ingangstype steeds omschakelen. Dat is vooral praktisch voor het omschakelen tussen schakelaars, INFRAROOD-ingangen en bedieningsveld-ingangen.

### 7.6.2 Analoge ingang



Via het element **Analoge ingang** kan men de waarde van een van de analoge ingangen opvragen. In tegenstelling tot digitale ingangen die alleen de waarde 0 of 1 kunnen leveren, kunnen analoge ingangen fijne schakeringen onderscheiden. Alle analoge ingangen leveren een uitgangswaarde tussen 0 en 1023. Bij de ROBO Interface zijn er nog verschillende soorten van analoge ingangen die verschillende fysische formaten hebben. Er zijn analoge ingangen voor weerstandsmetingen, voor spanningsmeten en voor een speciale sensor voor de afstandsmeting:

| Ingang | Ingangstype             | Meetbereik |
|--------|-------------------------|------------|
| A1, A2 | Spanningsingangen       | 0-10,23V   |
| AX, AY | Weerstandsignangen      | 0-5,5kΩ    |
| D1, D2 | Ingangen afstandssensor | ca. 0-50cm |
| AV     | Stroomvoorziening       | 0-10V      |

De gebruikelijke fischertechnik-sensoren NTC-weerstand, Fototransistor en Fotoweerstand vormen de meetwaarde (temperatuur resp. lichtsterkte) om in een weerstand. Vandaar dat je deze sensoren moet aansluiten op de **AX** - of **AY**-ingangen. De spanningsingangen **A1** en **A2** zijn voor alle sensoren bedoeld die een spanning tussen 0 en 10V opleveren.

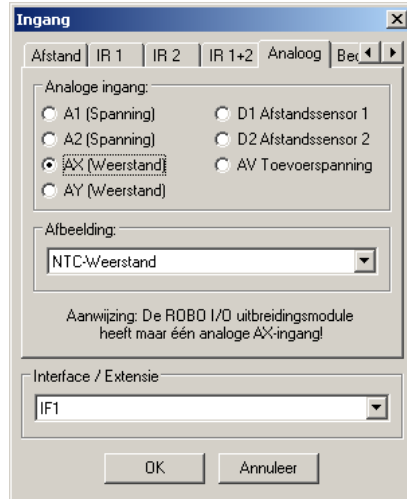
Voor de AV-ingang zijn er kleine bussen op de ROBO Interface. Hij is steeds verbonden met de stroomvoorziening van de interface. Op die manier kan je bv. de accuspanning controleren en je model in de uitgangspositie brengen voor de accu leeg is.

Aan de ingangen van de afstandssensoren D1 en D2 kunnen speciale sensoren van fischertechnik aangesloten worden die de afstand bv. tot een hindernis kunnen meten.

De Intelligent Interface heeft slechts twee analoge ingangen, EX en EY. Deze komen overeen met de ingangen AX en AY van de ROBO-interface. De andere analoge ingangen kunnen met de Intelligent Interface niet gebruikt worden!

### Het venster Eigenschappen voor Analoge ingangen:

- Onder **Analoge ingangen** kan je in overeenstemming met de bovenstaande tabel de gewenste analoge ingangen selecteren.
- Onder **Beeld** kan je een beeld van de sensor uitkiezen dat op de ingang aangesloten is.
- Onder **Interface / uitbreiding** kan je kiezen of je een ingang van de interface, een uitbreidingsmodule of een tweede interface wilt gebruiken. Je verneemt hierover meer in hoofdstuk 6 *Uitbreidingsmodule en meerdere interfaces aansturen* op pagina 51.



Aan de hand van het venster Eigenschappen voor analoge ingangen wordt terug duidelijk dat ROBO Pro voor alle ingangen maar een enkel element gebruikt dat via de ruiters op alle ingangstypes omgeschakeld kan worden. Ter vereenvoudiging staan in het elementvenster reeds verschillende ingangselementen ter selectie klaar.

### 7.6.3 Infrarood-ingang



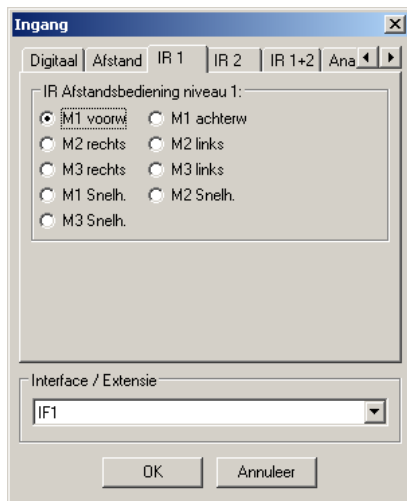
In de ROBO Interface is een infrarood-ontvanger voor de handzender uit de fischertechnik **Infrarood-besturingsset** art. nr. 30344 ingebouwd. De infrarood-zender is niet alleen voor de besturing van op afstand, maar ook algemeen als bedieningsbord voor je modellen erg praktisch. Voor de

**Infrarood-besturingsset** zijn er twee ontvangers die men met de toetsen 1 en 2 aan de handzender kan omschakelen. In de ROBO Interface kan je daarom elke toets van de handzender van een dubbele functie voorzien. Tussen de twee functies kan je dan heen en weer schakelen met de omschakeltoetsen 1 en 2. Alternatief kunnen de toetsen 1 en 2 als volledig normale toetsen gebruikt worden.

In het venster Eigenschappen van een Infrarood-ingang kan je in de ruitelijst bovenaan tussen IR 1, IR 2 en IR 1+2 omschakelen. Als je IR 1 geselecteerd hebt, levert het INFRAROOD-ingangselement alleen dan een 1 als de overeenkomstige schakelaar op de zender ingedrukt is en de zender daarvoor via de 1 toets ingesteld werd op functie 1. Als je IR 2 kiest, moet de zender daarentegen voor Toets 2 op functie 1 ingesteld zijn.

Als je echter IR 1+2 kiest, is het gelijk hoe de handzender ingesteld is. Dan kan je de toetsen 1))) en 2))) eveneens als ingangen gebruiken.

In het programma-element wordt de selectie door een witte 1 of 2 beneden rechts in het symbool van de handzender getoond. Bij IR 1+2 wordt in het programma-element geen getal getoond.



## 7.6.4 Motoruitgang



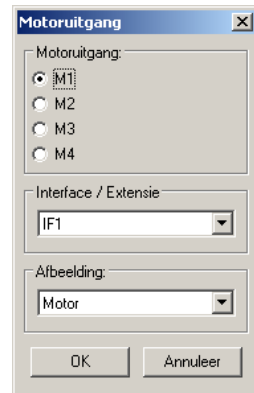
Via het element **Motoruitgang** kan één van de 4 tweepolige motoruitgangen van een ROBO Interface of Intelligent Interface aangestuurd worden. Een motoruitgang gebruikt steeds twee aansluitingen van de interface, terwijl een lampuitgang maar een aansluiting gebruikt. Je kan meer informatie vinden over het verschil tussen een motor- en een lampuitgang in de hoofdstukken 7.1.6 *Motoruitgang* op pagina 58 en 7.1.7 *Lampuitgang*.

Aan een motoruitgang moet via een bevelsegment een bevel gestuurd worden, zodat de uitgang geschakeld wordt. Een motorelement kan de volgende bevelen verwerken:

| Bevel  | Waarde   | Actie   |
|--------|----------|---|
| Rechts | 1 tot 8  | De motor draait naar rechts met snelheid 1 tot 8  |
| Links  | 1 tot 8  | De motor draait naar links met snelheid 1 tot 8   |
| Stop   | geen     | De motor stopt  |
| Aan    | 1 tot 8  | Zoals rechts  |
| Uit    | geen     | Zoals stop  |
| =      | -8 tot 8 | Waarde -1 tot -8: de motor draait naar links<br>Waarde 1 tot 8: de motor draait naar rechts<br>Waarde 0: de motor stopt |

Het venster Eigenschappen voor motorelementen:

- Onder **Motoruitgang** kan je selecteren welke uitgangsaansluitingen van de interface gebruikt moeten worden. Uitgangen van uitbreidingsmodules kies je onder **Interface / Uitbreiding**.
- Onder **Interface / Uitbreiding** kan je kiezen of je een uitgang van de interface, een uitbreidingsmodule of een tweede interface wilt gebruiken. Meer daarover verneem je in Hoofdstuk 6 *Uitbreidingsmodule en meerdere interfaces aansturen* op pagina 51.
- Onder **Beeld** kan je een beeld selecteren van de verbruiker die aan de uitgang aangesloten is. In de meeste gevallen zal dat een **motor** zijn. Je kan echter ook een **elektromagneet**, een **magneetventiel** of een **Lamp** aan een motoruitgang aansluiten.



### 7.6.5 Lampuitgang



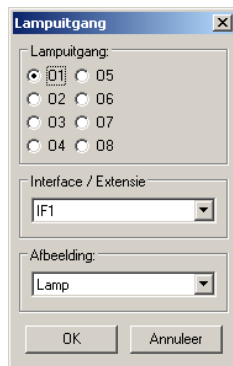
Via het element **Lampuitgang** kan je een van de 8 eenpolige lampuitgangen O1-O8 van een ROBO Interface of Intelligent Interface aansturen. Een lampuitgang gebruikt steeds maar een Uitgang-aansluiting van de Interface. De andere aansluiting van de verbruiker wordt met de massabus verbonden. Een zo aangesloten verbruiker kan je alleen aan- of uitschakelen, maar niet ompolen. Je verneemt meer over het verschil tussen een motor- en een lampuitgang in de hoofdstukken 7.1.6 *Motoruitgang* op pagina 58 en 7.1.7 *Lampuitgang (Niveau2)*.

Aan een lampelement moet via een bevelement een bevel gestuurd worden dat de uitgang schakelt. Een lampelement kan de volgende bevelen verwerken:

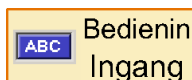
| Bevel | Waarde  | Actie   |
|-------|---------|---|
| Aan   | 1 tot 8 | De lamp wordt met een helderheid van 1 tot 8 ingeschakeld                           |
| Uit   | geen    | De lamp wordt uitgeschakeld   |
| =     | 0 tot 8 | Waarde 1 tot 8: de lamp wordt ingeschakeld<br>Waarde 0: de lamp wordt uitgeschakeld |

### Het venster Eigenschappen voor lampuitgangselementen:

- Onder **Lampuitgang** kan je selecteren welke uitgangsaansluiting van de interface gebruikt moet worden. Uitgangen van uitbreidingsmodules kies is je onder **Interface / Uitbreiding**.
- Onder **Interface / Uitbreiding** kan je kiezen of je een uitgang van de interface, een uitbreidingsmodule of een tweede interface moet gebruiken. Je verneemt hierover meer in het hoofdstuk 6 *Uitbreidingsmodule en meerdere interfaces aansturen* op pagina 51.
- Onder **Beeld** kan je een beeld van de verbruiker kiezen die aan de uitgang aangesloten is. In de meeste gevallen zal dat een **lamp** zijn. Je kan echter ook een **elektromagneet**, een **magneetventiel** en zelfs een **motor** aan een lampuitgang aansluiten. Maar een motor die aan een lampuitgang aangesloten is, kan slechts in een richting draaien.



## 7.6.6 Bedieningsveld-ingang



ROBO Pro biedt de mogelijkheid om eigen bedieningsvelden voor je modellen te tekenen. Je verneemt daar meer over in hoofdstuk 8 *Referentie Bedieningselementen en Bedieningsvelden* op pagina 83. Op die manier kan je je modellen comfortabel vanuit je computer

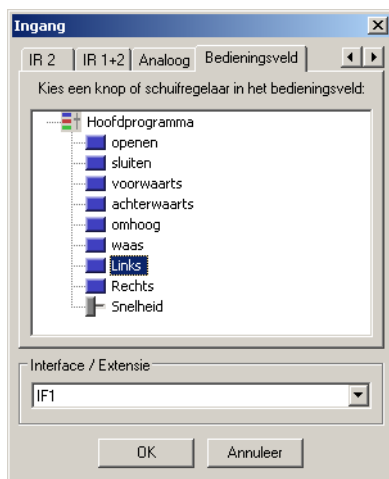
besturen. In het bedieningsveld beschik je over drukknoppen, schuifregelaars en invoerelementen. De toestand van deze elementen kan in het programma opgevraagd worden via het element **Bedieningsveld-ingang**. De drukknoppen leveren een waarde van 0 of 1. Schuifregelaars leveren een waarde in een instelbaar bereik (standaard 0 tot 100).

Bedieningsvelden kunnen enkel in de Online-bedrijfsmodus gebruikt worden. Je vindt hierover meer in het hoofdstuk 3.7 *Online- of Download-bedrijfsmodus – Wat is precies het verschil?* op pagina 24.

### Het venster Eigenschappen voor Bedieningsveld-ingangen:

Bij elk hoofd- of subprogramma hoort een bedieningsveld. Onder de naam van de programma's staat een lijst van de bedieningselementen. Als je nog geen bedieningselementen aangelegd hebt, verschijnen er ook geen elementen in de lijst. Je moet dus eerst het bedieningsveld tekenen voor je een ingang van een bedieningsveld met een bedieningselement kan verbinden.

De selectie onder **Interface / Uitbreiding** wordt bij de bedieningsveld-ingangen genegeerd, aangezien het hierbij niet om echte ingangen aan een interface-module gaat.





## 7.6.7 Bedieningsveld-uitgang



ROBO Pro biedt de mogelijkheid om eigen bedieningsvelden voor je modellen te tekenen. Je verneemt hierover meer in het hoofdstuk 8 *Referentie Bedieningselementen en Bedieningsvelden* op pagina 83. Naast drukknoppen en andere invoerelementen voor de besturing van je model,

kan je in een bedieningsveld ook weergave-elementen invoegen. in de weergave-elementen kan je bijvoorbeeld de asgegevens van een robot, of de toestand van een eindschakelaar laten weergeven. De aangeduide waarde kan je wijzigen door in je programma een element **Bedieningsveld-uitgang** in te voegen en het element =-bevel te besturen, bv. door een variabele, een analoge ingang, of een bevelement aan te sluiten.

Bedieningsvelden kunnen enkel in de Online-bedrijfsmodus gebruikt worden. Je verneemt hierover meer in het hoofdstuk 3.7 *Online- of Download-bedrijfsmodus – Wat is precies het verschil?* op pagina 24.

### Het venster Eigenschappen voor weergaven van bedieningsvelden:

Bij elk hoofd- of subprogramma hoort een bedieningsveld. Onder de naam van de programma's worden de weergaven van de bedieningsvelden in een lijst weergegeven. Als je nog geen bedieningsvelden aangelegd hebt, verschijnen er in de lijst ook geen elementen. Je moet dus eerst een bedieningsveld tekenen voor je een ingang van een bedieningsveld met een bedieningselement kan verbinden.



## 7.7 Operatoren

De programma-elementen van deze groep zijn zogenaamde operatoren. De operatoren hebben een of meerdere oranje data-ingangen. De waarde aan de data-ingangen worden door de operator tot een waarde gekoppeld en aan de uitgang van de operator per =-bevel uitgegeven.

### Het venster Eigenschappen voor Operatoren

Alle Operatoren gebruiken hetzelfde venster Eigenschappen. Je kan via het venster Eigenschappen ook de ene operator in een andere omzetten.

- Onder **Bewerking** kan je instellen hoe de operator zijn ingangen moet verbinden. De afzonderlijke functies worden in de volgende twee hoofdstukken uitgelegd.

Onder **Aantal ingangen** kan je instellen hoeveel ingangen de operator moet hebben voor Operatoren.



### 7.7.1 Aritmetische operatoren

ROBO Pro stelt je de vier belangrijkste rekenbewerkingen als operatoren ter beschikking. De symbolen bij twee ingangen zien er als volgt uit:

|         |         |         |         |       |
|---------|---------|---------|---------|-------|
|         |         |         |         |       |
| Plus    | Min     | Maal    | Gedeeld | Min   |
| $A + B$ | $A - B$ | $A * B$ | $A / B$ | $- A$ |

Als de **Min-operator** meer dan twee ingangen heeft, worden alle verdere ingangswaarden van de waarde aan de ingang 1 afgetrokken. Als de Min-operator slechts een ingang heeft, keert de operator het voorteken van de ingangswaarde om.

Als de **Deel-operator** meer dan twee ingangen heeft, wordt de waarde aan de ingang 1 door alle verdere ingangswaarden gedeeld.

### 7.7.2 Logische operatoren

Voor de koppeling van digitale ingangen zijn er bijvoorbeeld in ROBO Pro drie logische operatoren:

|                           |                           |            |
|---------------------------|---------------------------|------------|
|                           |                           |            |
| En                        | Of                        | Niet       |
| $A > 0 \text{ en } B > 0$ | $A > 0 \text{ of } B > 0$ | $A \leq 0$ |

De logische operatoren interpreteren een waarde groter dan nul aan de ingang als **ja** of **waar** en een waarde kleiner dan of gelijk aan nul als **neen** of **fout**. Digitale ingangen leveren een waarde van 0 of 1, zodat 0 als **fout** en 1 als **waar** geïnterpreteerd wordt.

De **En-operator** stuurt de aan de uitgang aangesloten elementen een = bevel met waarde 1, als aan alle ingangen de waarde waar is, dus als de waarde  $> 0$  is. In het andere geval stuurt het element een = bevel met waarde 0.

De **Of-operator** stuurt naar de elementen die aan de uitgang aangesloten zijn een = bevel met waarde 1, als aan minstens een van de ingangen de waarde waar is, dus een waarde heeft  $> 0$ . In het andere geval stuurt het element een = bevel met waarde 0.

De **Niet-operator** stuurt naar de elementen die aan de uitgang aangesloten zijn een = bevel met waarde 1, als aan de ingang de waarde fout is, dus een waarde heeft  $\leq 0$ . In het andere geval stuurt het element een = bevel met waarde 0.

De functie van logische operatoren kan men ook met meerdere vertakkings-elementen nabouwen. Maar het is vaak veel overzichtelijker om meerdere ingangen met operatoren te verbinden.

## 8 Overzicht van bedieningselementen en bedieningsvelden

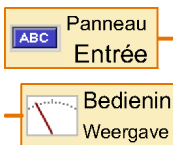
In ROBO Pro kunnen enkele bedieningsvelden gedefinieerd worden. Via bedieningsvelden kunnen complexe modellen comfortabel gestuurd worden. Aangezien het bedieningsveld op het PC-beeldscherm aangeduid wordt, werken de bedieningsvelden alleen in de Online-bedrijfsmodus. Zie daarvoor hoofdstuk 3.7 *Online- of Download-bedrijfsmodus – Wat is precies het verschil?* op pagina 24

Om een bedieningsveld op te stellen, selecteer je in de functiebalk **Bedieningsveld**:



In het lege grijze veld eronder kan je dan bedieningselementen invoegen. Een bedieningsveld hoort steeds bij het hoofd- of subprogramma waarin je je bij het opstellen van het bedieningsveld bevond. Daarom is het belangrijk dat je in de lijst van subprogramma's het juiste subprogramma selecteert voor je een bedieningsveld aanlegt. Meestal legt men bedieningsvelden aan onder het **Hoofdprogramma**.

In bedieningsvelden zijn er Weergaven en Besturingselementen. met Weergaven kan je bijvoorbeeld waarden van variabelen of tekstboodschappen weergeven. Besturingselementen werken daarentegen zoals bijkomende schakelaars of analoge ingangen.



Bij elk van de bedieningselementen die je in het bedieningsveld invoegt, hoort in het programma een element **Ingang van het bedieningsveld** (voor besturingselementen) of **Uitgang van het bedieningsveld** (voor Weergaven). Via deze programma-elementen maak je de koppeling tussen je programma en je bedieningsveld. Je vindt ze in de elementgroep **Ingangen, Uitgangen**. Volgens de manier waarop je het bedieningselement met deze programma-elementen verbindt, wordt een ander symbool weergegeven. In de elementenlijst staan er echter maar twee elementen: een voor weergaven en een voor besturingselementen.

### 8.1 Weergaven

Weergaven worden ongeveer hetzelfde gebruikt als interface-uitgangen. De waarde van een weergave kan je instellen met een =-bevel.

#### 8.1.1 Meetapparaat



Het **meetapparaat** is gemaakt naar het voorbeeld van een analogo meetinstrument met wijzer. Het wordt meestal gebruikt om de waarde van analoge ingangen aan te duiden, maar kan ook voor variabelen of andere programma-elementen gebruikt worden.

Het meetapparaat wordt vanuit een programma via een uitgang van een bedieningsveld gestuurd. De **uitgang van het bedieningsveld** vind je in de elementengroep **Ingangen, Uitgangen**.



De waarde van het meetapparaat stel je in door naar de bijhorende uitgang van het bedieningsveld in het programma een =-bevel te besturen. Bijna alle programma-elementen met data-uitgangen besturen een =-bevel, als



hun waarde verandert. Analoge ingangen of variabelen kan je bijvoorbeeld rechtstreeks met de uitgang van het bedieningsveld verbinden.

### Het venster Eigenschappen voor meetapparaten

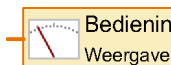
- Onder **ID / Naam** moet je eerst een naam voor het meetapparaat ingeven. De naam is belangrijk, zodat je meerdere meetapparaten in je programma van elkaar kan onderscheiden.
- Onder **Achtergrondkleur** kan je een andere kleur dan wit instellen.
- Onder **Minimumwaarde** en **Maximumwaarde** geeft je de waarde in die overeenkomt met de wijzerpositie aan de linker- en de rechterraand van de schaal. Als een van de waarden kleiner is dan 0 en de andere groter dan 0, wordt een bijzonder lange 0-streep getekend.
- De schaal bestaat uit lange en korte strepen. De afstand tussen de lange en de korte strepen geef je in onder **Lengte korte / lange markeringen**. Als beide dezelfde waarden hebben, zijn enkel lange markeringen zichtbaar.

### 8.1.2 Tekstweergave

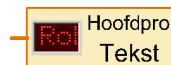


In een tekstweergave kan je getalwaarden, tekst of beide gemengd weergeven.

De tekstweergave wordt vanuit een programma via een uitgang van een bedieningsveld gestuurd. De **Uitgang van het bedieningsveld** vind je in de elementengroep **Ingangen, Uitgangen**.

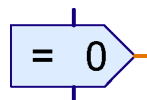


Zodra je de uitgang van het bedieningsveld via zijn venster Eigenschappen met een tekstweergave verbonden hebt, verandert het symbool en verschijnt de naam van het bedieningsveld (bv. hoofd) en de weergave (bv. tekst).

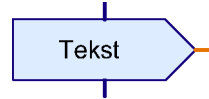


Je kan de tekst in de weergave op twee manieren instellen:

- Je kan de inhoud van de weergave instellen door naar de bijhorende uitgang van het bedieningsveld een **=bevel** te besturen. Dat is zeer praktisch als je de weergave wilt gebruiken om de waarde van een variabele of een ander programma-element weer te geven, omdat de meeste programma-elementen via hun data-uitgangen automatisch **=bevelen** besturen als de waarde verandert. Het **=bevel** overschrijft enkel de laatste 6 tekens van de weergave. De rest van de weergave kan je met een vooraf bepaalde tekst vullen. Op die manier kan je voor de waarde de tekst van een tip of een aanwijzing in de weergave plaatsen. Als de weergave meerdere regels heeft, kan je de tekst van de tip of de aanwijzing ook in een eigen cel zetten. Bij weergaven van meerdere regels worden door een **=bevel** enkel de laatste 6 tekens van de laatste regel overschreven.



- Met het tekstbevel kan je de inhoud van de weergave willekeurig instellen. Het tekstbevel is een speciaal bevelement dat niet alleen een getal, maar ook een hele tekst via zijn uitgang kan besturen. Zoals een gewoon bevelement kan het bevelsegment **Tekst** ook een data-ingang hebben. Je kan dan de bij de data-ingang passende getalwaarde in de tekst inbouwen. Als je een weergave-element met meerdere **Tekst**-bevelen stuurt, worden de teksten aan elkaar gehangen. Op deze manier kan je getallen en tekst willekeurig mengen.



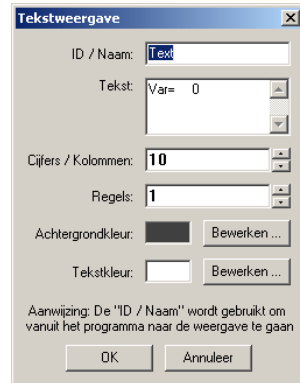
### Stuurtekens in tekstbevelen

In het bevelsegment **Tekst** kan je de volgende tekens gebruiken om speciale effecten te bereiken:

| Besturings-<br>teken | Effect  |
|----------------------|---|
| #####                | Geeft de waarde aan de data-ingang als getal met 5 posities + voorteken   |
| ##.##                | Geeft de waarde aan de data-ingang als getal met 2 cijfers na de komma die door een <i>punt</i> gescheiden.       |
| ##,##                | Geeft de waarde aan de data-ingang als getal met 2 cijfers na de komma die door een <i>komma</i> gescheiden zijn. |
| \c                   | Weergave wissen en invoegpunt aan het begin van de weergave plaatsen.   |

### Het venster Eigenschappen voor Tekstweergaven

- Onder **ID / Naam** moet je eerst een naam voor de weergave ingeven. De naam is belangrijk zodat je meerdere weergaven in je programma van elkaar kunt onderscheiden.
- Onder **Tekst** geef je de inhoud van de weergave in. Deze inhoud blijft behouden tot je vanuit het programma een bevel naar de weergave stuurt. Als je een **=**-bevel naar de weergave stuurt, worden enkel de laatste 6 tekens van de weergaveinhoud overschreven. Het begin van de tekst blijft behouden, zodat je voor een getal nog een aanwijzing kan opnemen om welk soort getal het gaat. In het afgebeelde voorbeeld blijft de tekst **Var=** behouden. De weergave heeft 10 karakters, er blijven dus  $10-6=4$  karakters behouden.
- Onder **Cijfers/kolommen** en onder **Regels** kan je ingeven voor hoeveel karakters de weergave plaats moet bieden. In een weergave met meerdere regels kan je een aanwijzing zoals **Var=** of **Bezoekers** in een eigen regel weergeven.
- Onder **Achtergrondkleur** en **tekstkleur** kan je het kleurendesign van de weergaven wijzigen. Klik op **Bewerken ...** om een kleur uit te kiezen of om een eigen kleur te definiëren.



### 8.1.3 Weergavelamp



De **weergavelamp** is de eenvoudige manier van weergave. Ze werkt ongeveer hetzelfde als een fischertechnik lampmodule die aan een uitgang van de interface aangesloten is.

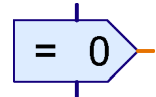
De weergavelamp wordt vanuit het programma via een uitgang van een bedieningsveld gestuurd. De **uitgang van het bedieningsveld** vind je in de elementengroep **Ingangen, Uitgangen**



Zodra je de uitgang van het bedieningsveld via het venster Eigenschappen met een weergavelamp verbonden hebt, verandert het symbool en verschijnt de naam van het bedieningsveld (bv. hoofd) en de lamp.



Je kan de lamp aan- of uitschakelen door aan de bijhorende uitgang van het bedieningsveld in je programma een bevel **Aan** of **Uit** te besturen, zoals je die ook voor echte lampuitgangen gebruikt. Je kan de weergavelamp ook via een **=**-bevel aan- en uitschakelen. Is de waarde groter dan 0, dan wordt de lamp ingeschakeld. Is de waarde kleiner dan of gelijk aan 0, dan wordt de lamp uitgeschakeld.



#### Het venster Eigenschappen voor weergavelampen

- Onder **ID / Naam** moet je eerst een naam voor de weergavelamp ingeven. De naam is belangrijk zodat je meerdere weergavelampen in je programma van elkaar kan onderscheiden.
- Onder **Kleur** kan je de kleur van de weergavelamp wijzigen. Klik daarvoor op de knop **Bewerken...**
- Als **aan het begin aan** aangevinkt is, brandt de weergavelamp tot het bijhorende programma-element het eerste bevel krijgt. In het andere geval is de weergavelamp aan het begin gedoofd.



## 8.2 Besturingselementen

Besturingselementen worden ongeveer op dezelfde manier gebruikt als interface-ingangen.

### 8.2.1 Knop

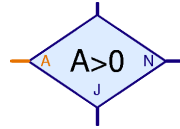


De **bedieningsknop** kan je gebruiken zoals een schakelaar van fischertechnik die aan de ingangen van de interface aangesloten is.

De knop wordt door het programma via een **Ingang van een bedieningsveld** ondervraagd. De **Ingang van het bedieningsveld** vind je in de elementengroep **Ingangen, Uitgangen**.

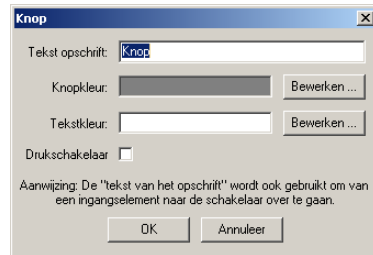


De knop die bij de ingang van het bedieningsveld hoort, kan je zoals een digitale ingang van de interface aan alle programma-elementen met een data-ingang, zoals bijvoorbeeld aan de **Vertakking**, aansluiten. Als de knop ingedrukt wordt, levert hij als waarde 1, in het andere geval de waarde 0.



### Het venster Eigenschappen voor knoppen

- Onder **Tekst opschrift** kan je het opschrift voor de knop ingeven. Dat is tegelijk de naam waarmee je vanuit het programma gebruik zal maken van de knop. Een bijkomende naam /ID-veld zoals bij de andere bedieningselementen is er bij deze knop niet.
- Onder **Kleur van de knop en Tekstkleur** kan je het kleurdesign van de knop veranderen. Klik daarvoor op **Bewerken...**
- Als bij **Drukschakelaar** een haakje verschijnt, werkt de knop niet als taster, maar als een schakelaar. Bij de eerste klik op de knop wordt de knop ingedrukt en blijft dan ingedrukt tot hij een tweede keer ingedrukt wordt. In het andere geval werkt de knop als taster en springt meteen terug als hij losgelaten wordt.



## 8.2.2 Regelaars

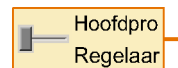


De regelaar kan je zoals een potentiometer gebruiken die aan een analoge ingang van de interface aangesloten is. Anders dan de knop kan de regelaar niet alleen de waarde 0 en 1 leveren, maar zoals een analoge ingang vele waarden. Het waardebereik kan via het venster Eigenschappen ingesteld worden. De regelaar kan bijvoorbeeld gebruikt worden om het motortoerental tussen 1 en 8 in te stellen.

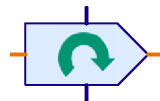
De regelaar wordt vanuit het programma via een ingang van het bedieningsveld ondervraagd. De **Ingang van het bedieningsveld** vind je in de elementengroep **Ingangen, Uitgangen**.



Zodra je de ingang van het bedieningsveld via zijn venster Eigenschappen met een regelaar verbonden hebt, verandert het symbool en verschijnt de naam van het bedieningsveld (bv. Hoofd) en van de regelaar.

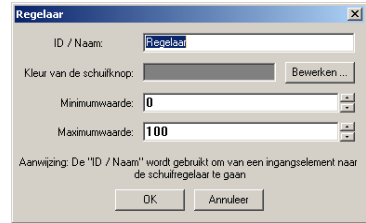


De bij de regelaar behorende ingang van het bedieningsveld kan je zoals een analoge ingang van de interface aan alle programma-elementen met een data-ingang aansluiten. Zeer vaak wordt de regelaar aan een bevelement met data-ingang aangesloten, zodat de regelaar de snelheid van een motor stuurt.



## Het venster Eigenschappen voor regelaars

- Onder **ID / Naam** moet je eerst een naam voor de regelaar ingeven. De naam is belangrijk, zodat je meerdere regelaars in het programma van elkaar kan onderscheiden.
- Onder **kleur schuifknoppen** kan je de kleur van de schuifknop veranderen. Klik daarvoor op **Bewerken**.
- Onder **Minimumwaarde** en **Maximumwaarde** geef je het waardebereik voor de regelaar aan. Als je de regelaar voor de besturing van een motorsnelheid wilt gebruiken, moet het waardebereik van 1 tot 8 gaan.

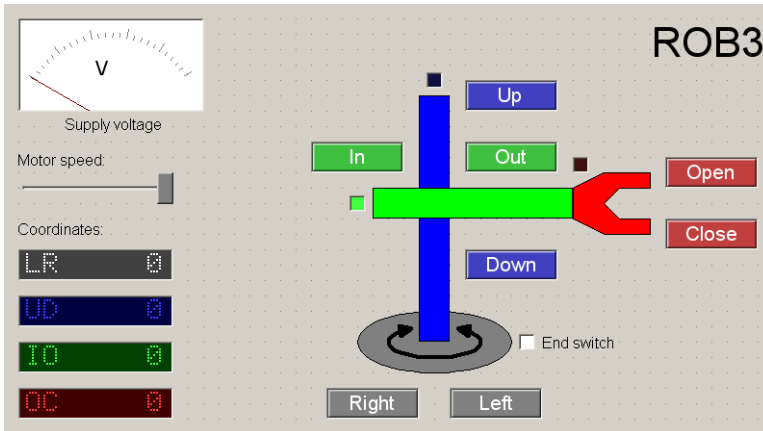




## 9 Tekenfuntcies

ROBO Pro beschikt over de gebruikelijke tekenfuncties. Je vindt ze in het venster voor elementgroepen onder **Tekenen**. In der Subgroep **Vormen** zijn tekentools voor verschillende geometrische basisvormen opgenomen. In de subgroep **Tekst** vind je tekst-tools voor verschillende werktuigen voor verschillende lettertypes. De andere subgroepen bevatten functies voor het veranderen van de kleur en de lijndikte.

Met de tekenfuncties kan je de bedieningsvelden en programma's illustreren om de functie te verduidelijken. Hier is bijvoorbeeld een zelf getekend bedieningsveld voor een robot afgebeeld:



De knoppen, de coördinaatweergaven en de lampen van de eindschakelaars zijn telkens in dezelfde kleur gehouden, zoals de diverse assen in de schematische tekening van de robot. Daardoor ontstaat een zeer overzichtelijk bedieningsveld.

De toepassing van de tekenfuncties zou geen grote moeilijkheden mogen veroorzaken. Hierna worden alleen nog enkele punten uitgelegd die misschien niet meteen duidelijk zijn:

- Grafische objecten zoals rechthoeken en cirkels worden **niet** zoals in vele andere programma's met een ingedrukte muistoets getrokken, maar door twee muisklikken, een in de linker bovenhoek en een in de rechter benedenhoek.
- Tekst wordt niet in een dialoogvenster bewerkt, maar direct in de werkzone. Als je een nieuw tekstobject invoegt, verschijnt eerst een lichtblauwe kader. Je kan nu gewoon via het toetsenbord schrijven en de geschreven tekst verschijnt dan in het werkvlak. Je kan ook tekst van het klembord met CTRL+V invoegen.
- Nadat je een object getekend hebt, kan je het object bewerken door de kleine blauwe grijppunten te verschuiven. Er zijn ook grijppunten voor het draaien en vervormen van objecten. Een rechthoek heeft boven links twee grijppunten. Als je het tweede grijppunt verschuift, kan je de hoek van de rechthoek afronden. Je kan de bewerkingsmodus beëindigen door de rechter muistoets of de **ESC**-toets in te drukken.
- Als je een object achteraf wilt bewerken, kies in het menu **Tekenen** de functie **Bewerken**. Als je dan op een object klikt, verschijnen terug de lichtblauw grijppunten.
- Vele objecten hebben twee of meer bewerkings- en tekenmodi. Je kan tussen de diverse modi heen en weer gaan met de tabulatortoets op het toetsenbord, terwijl je een object tekent

of bewerkt. Bij een kring kan je bijvoorbeeld kiezen of je twee hoekpunten of het middelpunt en een eindpunt wilt aangeven. Bij veelhoeken kan je tussen puntbewerking en functies zoals "Draaien" wisselen. Bij tekstvoorwerpen kan omschakelen tussen Bewerking van de tekst en Verandering van het tekstformaat en de draaihoek.

- In het menu **Tekenen** zijn er functies **Object op de voorgrond / achtergrond**. Met deze functies kan je alle geselecteerd ( rood nagetekende) objecten naar voren of naar achteren verschuiven, zodat andere objecten ze bedekken of erdoor bedekt worden.
- Met de functie **Rastersnap** in het menu **Tekenen** kan je het tekenraster in- of uitschakelen. Je moet er daarbij wel op letten dat het raster ingeschakeld is, als je je programma bewerkt, aangezien alle programma-elementen aan het raster aangepast zijn.
- Bij tekstobjecten kan je de tekstrichting veranderen door de CTRL + een toets van 1-9 op het cijferblok in te drukken. Dit kan echter alleen als de Num-lock lamp op het toetsenbord brandt. Als deze niet brandt, moet je eerst de Num-toets indrukken.