

FISCHERTECHNIKROBOP
ROBASCOMAVRFISCHERT



ECHN
AVRI
BOPE
ERTE
COMA
KROBOPROBASCOMAVRFI
SCHERTECHNIKROBOPRO
BASCOMAVRFISCHERTEC
HNIKROBOPROBASCOMAV
RFISCHERTECHNIKROBO
PROBASCOMAVRFISCHER
TECHNIKROBOPROBASCO
MAVRFISCHERTECHNIKR

Voorbeelden Besturingstechniek met ROBO Pro en Bascom-AVR

1^{ste} druk , november 2011

De voorbeelden zijn eigendom van fischertechnikclub Nederland. Het is niet toegestaan om deze voorbeelden geheel of gedeeltelijk te kopiëren, te verkopen of te verhuren zonder toestemming van de fischertechnikclub Nederland

“fischertechnik” is een geregistreerd handelsmerk van fischertechnik GmbH.
Weinhalde 14-18, D-72178, Waldachtal.

“Bascom-AVR” is ontwikkeld door MCS Electronics, De Paal 112, 1351JJ,
Almere-Haven.

1. Inleiding.

Tijdens het eerste lustrum van de fischertechnikclub Nederland (1996) is door de heren Stef Dijkstra, Cees Nobel en Herman Ettema besloten tot het maken en uitgeven van een “Cursus Besturingstechniek met Luckylogic, Basic en PLC”. De bedoeling van deze cursus was om m.b.v. een computer of m.b.v. een PLC een fischertechnikmodel aan te sturen. In de cursus werd een lasrobot als model gebruikt. De lasrobot kon toen worden gebouwd met onderdelen uit de doos Computing.

Inmiddels zijn wij 15 jaar verder en is er op het gebied van fischertechnik-besturing het een en ander veranderd. Luckylogic is in 2000 opgevolgd door LLWin. In 2004 is ROBO Pro geïntroduceerd als vervanging voor LLWin.

Om de leden vertrouwd te maken met ROBO Pro heeft de fischertechnikclub Nederland een interactieve cursus ROBO Pro georganiseerd. De cursus (voorbereid door Paul Kaakman, Peter Damen en Andries Tieleman) is in mei 2010 gegeven door Andries Tieleman. Ook voor deze cursus werd gebruik gemaakt van een lasrobot.

Gezien het succes van deze ROBO Pro cursus kwam Cees Nobel op het idee om bij gelegenheid van het 4^{de} lustrum van de fischertechnikclub Nederland de voorbeelden uit de cursus besturingstechniek uit 1996 te herschrijven voor ROBO Pro.

Naast ROBO Pro worden de voorbeelden uit de Luckylogic-cursus ook gepresenteerd in de programmeertaal Bascom-AVR.¹⁾ Bascom AVR heeft een groot voordeel boven andere programmeertalen omdat het, na een korte inwerktijd, al mogelijk is een eigen programma te schrijven.

Hoewel qua programmering ROBO Pro veel sneller tot een resultaat leidt dan Bascom-AVR blijft een nadeel dat de ROBO Pro interface (TX controller) duur is in aanschaf. Voor een kleine 50 euro kun je al uit de voeten met een Bascom_AVR controller (inclusief een motorstuurkaart), terwijl je voor de ROBO TX controller al gauw 200 euro kwijt bent.

Wederom is voor de voorbeelden gebruik gemaakt van een lasrobot. Voor het schrijven van dit voorbeeldenboek is/mocht dan ook dankbaar gebruik gemaakt worden van het materiaal uit de interactieve ROBO Pro cursus uit 2010.

Voor de toekomst bestaan er plannen om de voorbeelden ook uit te werken in andere programmeertalen (b.v. Arduino, C++, Propellor, MS Robotics Studio-VPL).

Indien u vragen hebt over de voorbeelden kunt u altijd terecht bij de fischertechnikclub Nederland of bij een van de auteurs van deze cursus:

Voorbeelden ROBO Pro: Cees Nobel.

Voorbeelden Bascom-AVR: Rien van Harmelen.

¹⁾ Windows Bascom-AVR is een Basic Compiler voor de AVR familie. Voor informatie over Bascom-AVR wordt verwezen naar de website <http://www.mcselec.com>

2. **Bouwbeschrijving van de lasrobot**

De lasrobot bestaat uit een draaischijf en een laskolom met puntlasser. Het model is identiek aan het model uit de interactieve ROBO Pro-cursus gegeven in mei 2010 (en is vergelijkbaar met het model uit de Luckylogic cursus). Het model kan in 2 fasen gebouwd worden , omdat alleen de laatste programma's het volledige model nodig hebben:

- Fase 1 bestaat uit de basisplaat, het “bedieningspaneel” met 2 schakelaars I3 en I4 en een draaischijf;
- In fase 2 wordt het model gecompleteerd met de laskolom met puntlasser.

In Bijlage 1 is aan de hand van foto's de opbouw van de lasrobot gegeven.

3. Besturing van de lasrobot

3.1 Besturing via ROBO Pro

Voor ROBO Pro kunnen de volgende interfaces worden gebruikt:

- ROBO interface (art. nr 93393);
- ROBO I/O Extension (art nr. 93294);
- ROBO TX Controller (art. Nr. 500995).

3.1.1 Aansluiten bekabeling

In de lasrobot zijn ingebouwd:

- 2 motoren;
- 2 lampen;
- en 6 schakelaars.

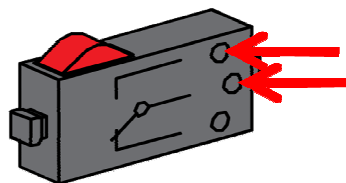
Voor de aansluitgegevens van de motoren en lampen op de interface: zie onderstaande tabel:

Motor	Locatie	Aansluiten op	Opmerking
Motor M1	draaischijf	M1	Links=counterclockwise Rechts= clockwise
Motor M3	laskolom	M3	Links=omhoog Rechts=naar beneden
Lampje M2	laskolom	M2	
Lampje M4		M4	Initiatielamp

Voor aansluitgegevens van de schakelaars op de interface: zie onderstaande tabel:

Schakelaar	Locatie	Aansluiten op	Opmerking
Schakelaar I1	draaischijf	I1	startpositie-schakelaar
Schakelaar I2	draaischijf	I2	positieteller draaitafel
Schakelaar I3		I3	startknop
Schakelaar I4		I4	noodknop
Schakelaar I5	laskolom	I5	eindschakelaar onder
Schakelaar I6	laskolom	I6	eindschakelaar boven

Alle schakelaars zijn nul in rust toestand (normally open).

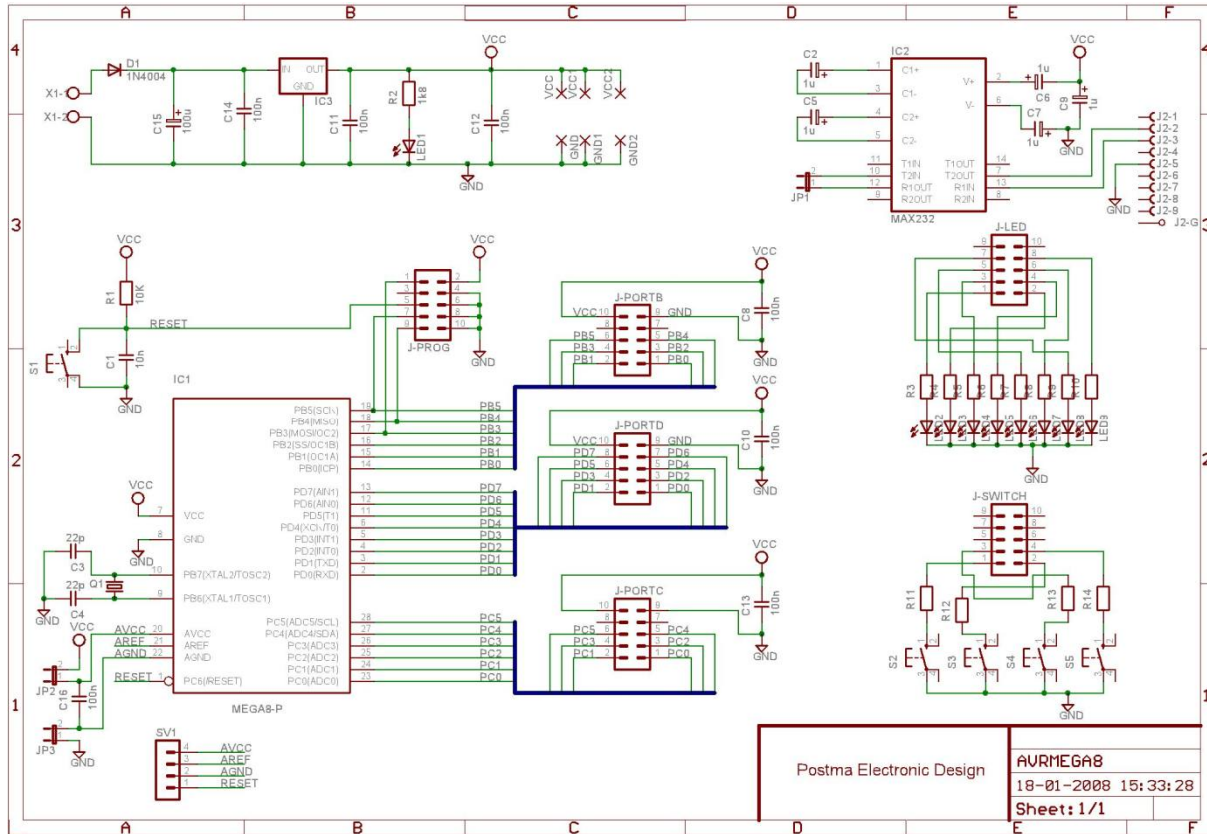


Figuur 1: Te gebruiken ingangen van de fischertechnik schakelaar (37783)

Zie verder de documentatie van fischertechnik.

3.2 Besturing via Bascom-AVR

In principe kan de lasrobot via Bascom-AVR met alle Atmega-AVR microcontrollers aangestuurd worden. In deze cursus is gebruik gemaakt van een (instap) bordje voorzien van een Atmega8 processor met een 16 Mhz kristal.



Figuur 1 : Schema van het gebruikte Atmega8 bordje (www.postmaelectronics.nl)

3.2.1 Aansluitingen bekabeling

In de lasrobot zijn ingebouwd:

- 2 motoren (M1: draaitafel en M3: lasarm);
- 2 lampen (M2: lampje op lasarm en M4: initialisatielamp);
- 1 startpositie-schakelaar(E1);
- 2 eindschakelaars (E5 en E6);
- 1 pulsschakelaar (positieteller draaitafel E2);
- 1 start schakelaar (E3);
- 1 noodstop-schakelaar (E4).

Hieronder zal worden aangegeven hoe de verschillende onderdelen aangesloten worden op het microcontrollerbordje.

Motoren

De motoren kunnen niet direct door de microcontroller aangestuurd worden. De twee motoren (M1 en M3) worden aangestuurd door de microcontroller via een L298 motorstuurkaart (waar in ons geval, twee motoren op aangesloten kunnen worden).

Per motor zijn er op de motorstuurkaart twee inputpoorten (In1 en In2) en een enable poort.

De inputpoorten bepalen de draairichting van de motor. De enable poort bepaalt de snelheid van de motoren via het Bascom-AVR commando:

- Config Timer1 = Pwm , Pwm = 10 , Prescale = 1 , Compare A Pwm = Clear Down , Compare B Pwm = Clear Down ²⁾

Omdat in de gegeven voorbeelden variatie in snelheid niet relevant is, blijft in alle voorbeelden de enable poort onveranderd hoog (speed = 1000 \approx 1024).



Figuur 2: L298 motorstuurkaart van VOTI (www.voti.nl)

Input			Output
Enable	In1	In2	Motor1/Motor3
H	H	H	Motor remt
H	L	L	Motor remt
H	H	L	Draait richting 1
H	L	H	Draait richting 2
L	X	X	Motor juttert (zie voorbeeld 14)

De aansturing (input) van de motorstuurkaart gebeurt via onderstaande poorten op de microcontroller:

	Enable	In1	In2
Motor draaischijf (M1)	Portb.1	Portb.0	Portb.5
Motor puntlasser (M3)	Portb.2	Portd.6	Portd.7

Opmerking:

- Motor M1 "links" betekent: motor draait counterclockwise (ccw);
- Motor M1 "rechts" betekent: motor draait clockwise (cw);
- Motor M3 "links" betekent: puntlaslamp gaat omhoog;
- Motor M3 "rechts" betekent: puntlaslamp gaat naar beneden.

²⁾ Informatie hierover is te vinden onder http://www.rn-wissen.de/index.php/Bascom_und_PWM

Lampen

De lampen M2 (lamp puntlasser) en M4 (initialisatielamp) kunnen op verschillende manieren aangesloten worden:

- Bij gebruik van leds kunnen de lampen M2 en M4 direct (via eventueel een weerstand) op de microcontroller aangesloten worden;
- Bij gebruik van de fischertechnieklampjes zijn er twee mogelijkheden:
 - Aansluiten op de microcontroller via een ULN 2803;
 - Of ook zoals de motorjes (M1 en M3) via een motorstuurkaart.

In het (Bascom-AVR) model zijn de fischertechnieklampjes aangesloten op een motorstuurkaart. Omdat er bij de lampen geen sprake is van een draairichting is er slechts één poort nodig voor de aansturing:

	Enable	In1	In2
Lamp puntlasser (M2)	5V	GND	Portb.3
Initialisatielamp (M4)	5V	GND	Portb.4

Opmerking:

De frequentie waarmee M4 wordt aangestuurd wordt bepaald door Timer0.

De Prescale van Timer0 = 1024. De timerfrequentie is kloksignaal van het crystal/prescale waarde (=16000000/1024 = 15625 Hz). De Timer0 telt van 0 tot 256. Bij iedere 256^{ste} klokpuls verandert het signaal van 1 naar 0 of van 0 naar 1. Frequentie van het signaal is dus $15625/(2 \times 256) = 30.51$ Hz. Door het introduceren van een wachttijd = 30 wordt de frequentie : $30,51/30 = 1$ Hz (dus “trillings” tijd van de initialisatielamp= 1 sec).

Drukknoppen

In het model worden de drukknoppen gebruikt als:

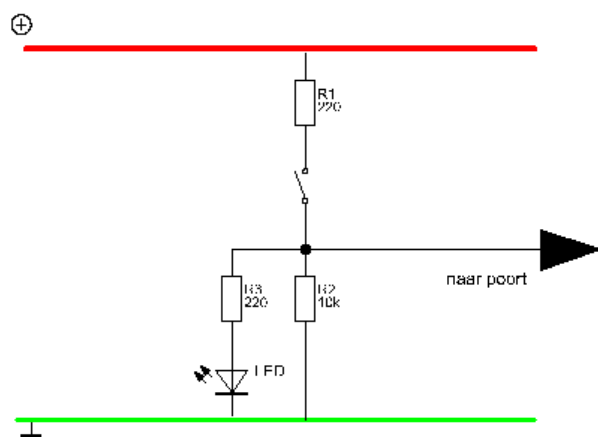
- Startknop (E3);
- Noodstop (E4);
- Startpositieschakelaar (E1);
- Eindschakelaar (E5 en E6);
- Puls/positieteller (E2).

M.u.v. de pulsteller (E2) zijn alle drukknoppen aangesloten op microcontroller-poort C.

Schakelaar	Poort
Startpositie schakelaar E1	PortC.5
Puls/positie-teller E2	PortD.3.
Startknop E3	PortC.4
Noodstop E4	PortC.3
Eindschakelaar E5	PortC.2
Eindschakelaar E6	PortC.0

PortD.3 is de interrupt poort (INT1).

Voor de drukknoppen op poort C geldt : ingedrukt = 1 !!, dus “nul” in rusttoestand. Om dit te bereiken zijn de ingangen van poort C via een weerstand van 10k verbonden met GND.



Figuur 3: Schema van de aansluiting van de drukknoppen op de microcontrollerkaart

Probleem met een puls/positieteller door een schakelaar is de dender (button debounce) . De invloed van de dender kan op twee manieren onderdrukt worden :

- Hardware matig;
- Software matig.

Hardwarematig kan de dender onderdrukt worden door gebruik te maken van een flipflip (b.v. het IC7474 P). Zie b.v. “Electronica echt niet moeilijk. Deel 1, 2 en 3” blz. 474.

Softwarematig kan in Bascom–AVR gebruik gemaakt worden van de Config Debounce statement. Voor de debounce waarde is proefondervindelijk 20 gekozen.

3.2.2 Extra

Om in de uitvoering van de programma’s de verschillende fasen te kunnen volgen is een LCD scherm aangebracht. I.v.m. een tekort aan vrije poorten van de Atmega8 processor is er gebruik gemaakt van een I2C-LCD scherm dat is aangesloten op Portd.0 (SCL) en Portd.1 (SDA).³⁾

³⁾ Voor informatie over het aansturen van het gebruikte LCD scherm zie: [http://www.mcselec.com/Bascom-AVR, Application Notes, AN #118 \(I2C LCD & Keyboard Library\)](http://www.mcselec.com/Bascom-AVR, Application Notes, AN #118 (I2C LCD & Keyboard Library))

4. Programma's

Er zijn in totaal 39 programma's:

- Programma 1 t/m programma 32 hebben betrekking op de draaischijf ;
- Programma 33 t/m programma 36 hebben betrekking op de laskolom met punt lasser;
- Programma 37 t/m programma 39 gelden voor het complete model.

Programma	Schakeling	Programma	Schakeling
1	Yes schakeling	20	Tijdschakeling
2	No schakeling	21	Tijdschakeling
3	And schakeling	22	Tijdschakeling
4	Or schakeling	23	Veiligheidsschakeling
5	Nand schakeling	24	Opkomstvertraging
6	Nor schakeling	25	Afvalvertraging
7	Exor schakeling	26	Op/afvalvertraging
8	Exnor schakeling	27	Op/afvalvertraging
9	veiligheidsschakeling	28	Positionering
10	Joystickschakeling	29	Positionering
11	Joystickschakeling	30	Positionering
12	Geheugenschakeling	31	Positionering
13	Geheugenschakeling	32	Positionering
14	Geheugenschakeling	33	Omhoog/omlaag
15	Geheugenschakeling	34	Omhoog/omlaag
16	Geheugenschakeling	35	Omhoog/omlaag
17	Geheugenschakeling	36	Lamp aan/uit
18	Geheugenschakeling	37	Combinatie 35-36
19	Tijdschakeling	38	Compleet model
		39	Compleet model

4.1 Opmerkingen over de programma's

Het is duidelijk dat je op tig manieren zowel in ROBO Pro als in Bascom-AVR een programma kunt schrijven.

ROBO Pro

In ROBO Pro is er voor gekozen om uit te gaan van elementaire functies en zo min mogelijk gebruik te maken van subprogramma's. In de grafische omgeving is zo het verloop van het programma goed te volgen.

Bascom AVR

Voor de duidelijkheid is er voor gekozen om, indien het programma bestaat uit meerdere stappen, gebruik te maken van SELECT CASE structuur. Iedere stap in het programma, inclusief de pauzes, krijgen een apart CASE nummer. Voordeel: het programma-verloop is goed te volgen. Nadeel: de programma's hebben de neiging wat te lang te worden.

Het is aan de lezer om te kijken of er mogelijkheden zijn de betreffende programma's wat in te korten.

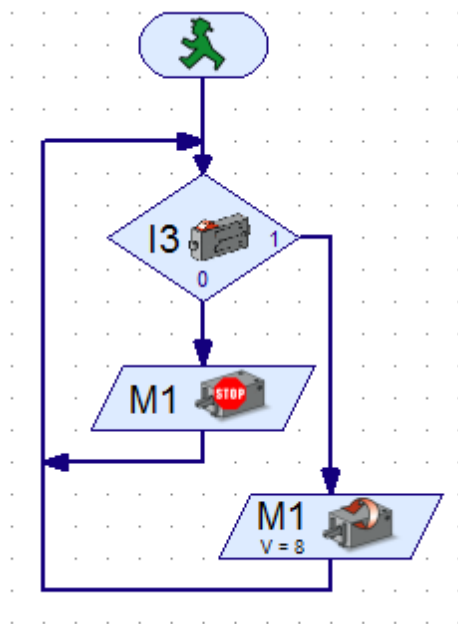
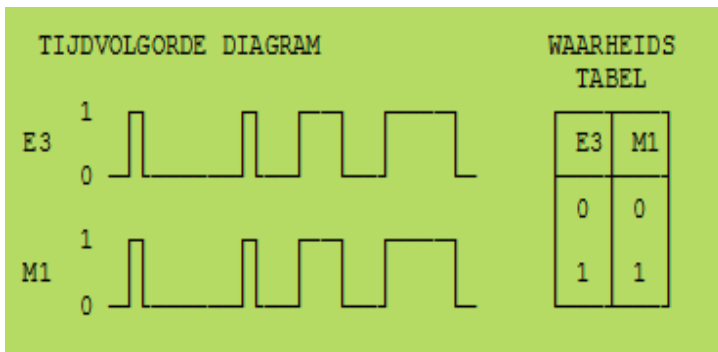
5 Invoer van de programma's 1 t/m 39 in ROBO Pro en Bascom AVR

Opmerking:

- De invoer van de Bascom-AVR programma's is beperkt tot het gedeelte tussen de DO-LOOP;
- Voor de Bascom-AVR compiler gegevens en instellingen: zie Bijlage 2;
- Voor de Bascom-AVR subroutines: zie Bijlage 3.

VOORBEELD-01: YES-SCHAKELING.

Als drukknop (E3) ingedrukt wordt, dan gaat de draaischijf (M1) linksom draaien.

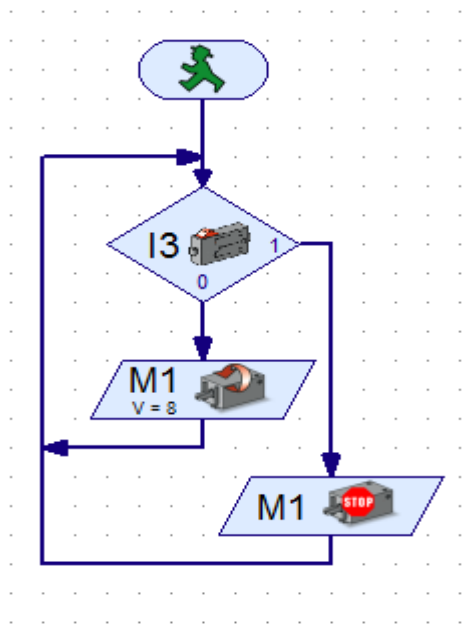
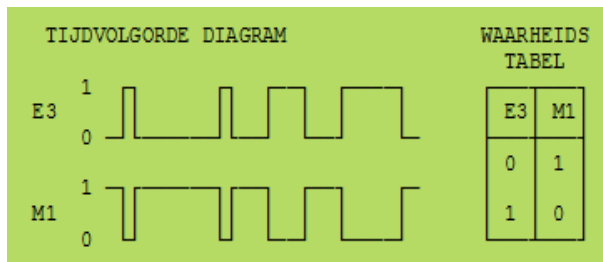


```

BASC0M-AVR
Compiler gegevens en instellingen.
(zie bijlage:2)
Do
  If E3 = 1 Then
    Call Motor1_links
  Elseif E3 = 0 Then
    Call Motor1_stop
  End If
Loop
Subroutines
(zie bijlage:3)
    
```

VOORBEELD-02: NOT-SCHAKELING.

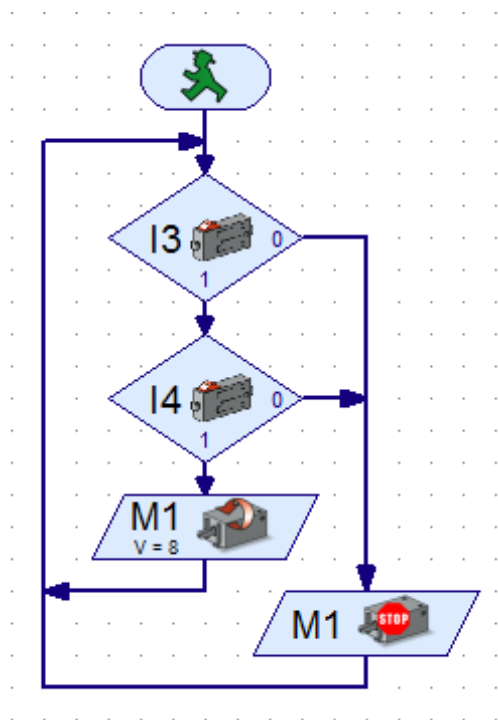
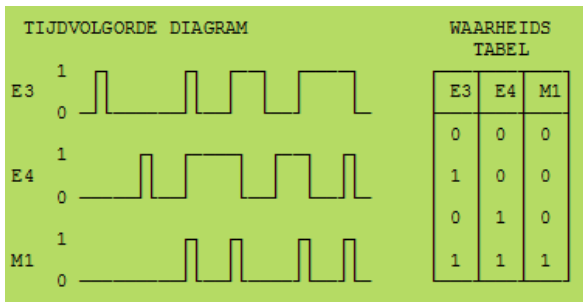
Als het programma start gaat de draaischijf (M1) linksom draaien.
De draaischijf stop als drukknop (E3)ingedrukt wordt.



```
BASCOM-AVR
Compiler gegevens en instellingen.
(zie bijlage:2)
Do
  If E3 = 1 Then
    Call Motor1_stop
  Elseif E3 = 0 Then
    Call Motor1_links
  End If
Loop
Subroutines
(zie bijlage:3)
```

VOORBEELD-03: AND-SCHAKELING.

Alleen als BEIDE drukknoppen (E3 en E4) ingedrukt worden, dan gaat de draaischijf (M1) linksom draaien.

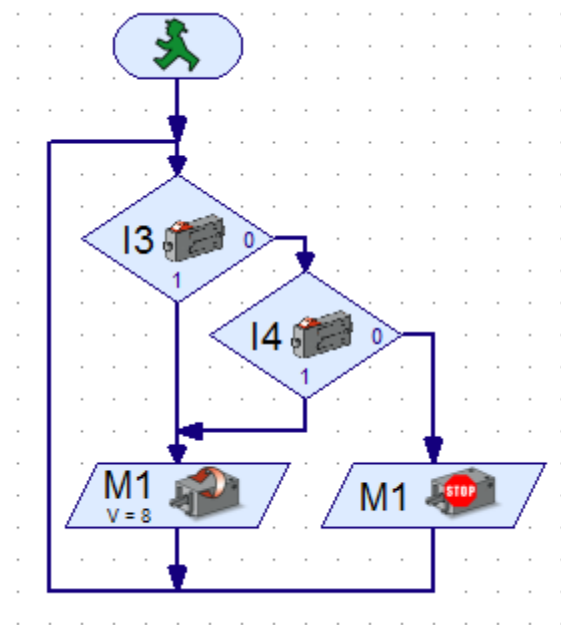
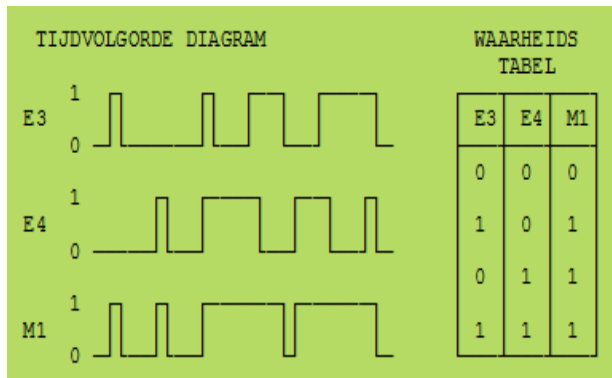


```

BASCOS-AVR
Compiler gegevens en instellingen.
(zie bijlage:2)
Do
  If E3 = 1 And E4 = 1 Then
    Call Motor1_links
  Else
    Call Motor1_stop
  End If
Loop
Subroutines
(zie bijlage:3)
    
```

VOORBEELD-04: OR-SCHAKELING.

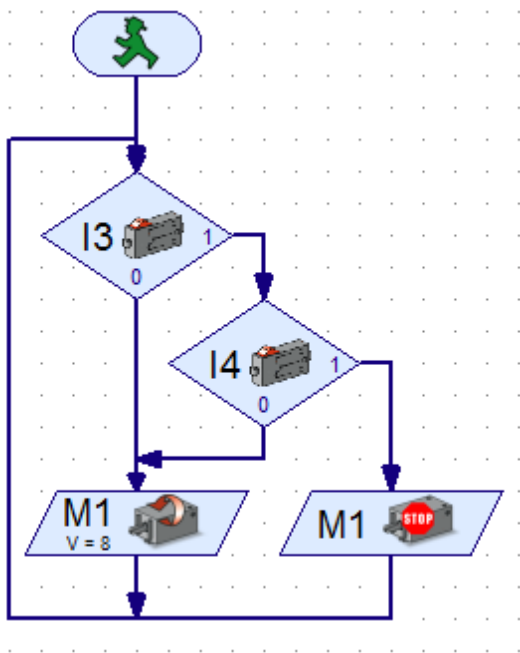
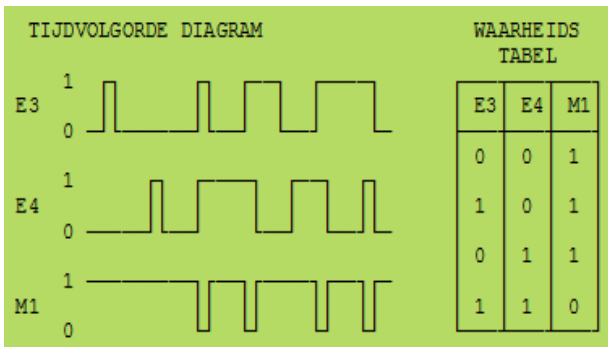
Als beide of één van beide drukknoppen (E3 en/of E4) ingedrukt worden, dan gaat de draaischijf (M1) linksom draaien.



```
BASCOM-AVR
Compiler gegevens en instellingen.
(zie bijlage:2)
Do
  If E3 = 1 Or E4 = 1 Then
    Call Motor1_links
  Else
    Call Motor1_stop
  End If
Loop
Subroutines
(zie bijlage:3)
```

VOORBEELD-05: NAND-SCHAKELING.

Als het programma start dan gaat de draaischijf (M1) linksom draaien.
De draaischijf stopt als BEIDE drukknoppen (E3 en E4) ingedrukt worden.



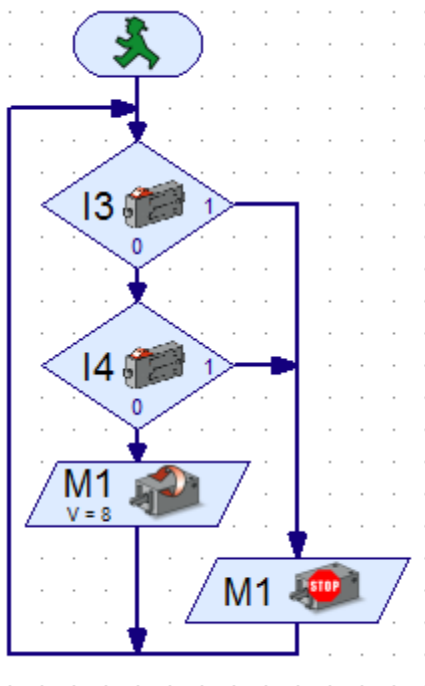
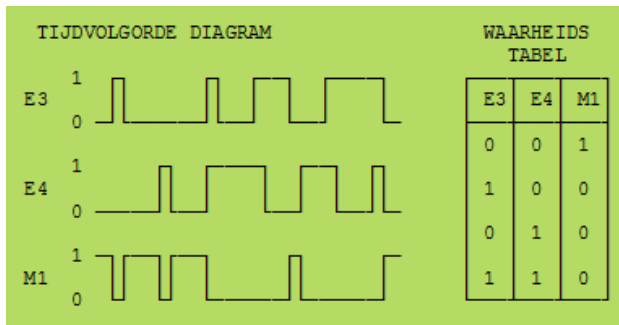
```

BASCOM-AVR
Compiler gegevens en instellingen.
(zie bijlage:2)
Do
  If E3 = 1 And E4 = 1 Then
    Call Motor1_stop
  Else
    Call Motor1_links
  End If
Loop
Subroutines
(zie bijlage:3)
  
```


VOORBEELD-06: NOR-SCHAKELING.

Als het programma start dan gaat draaischijf (M1) linksom draaien.

De draaischijf stopt als beide of één van beide drukknoppen (E3 en/of E4) ingedrukt worden.

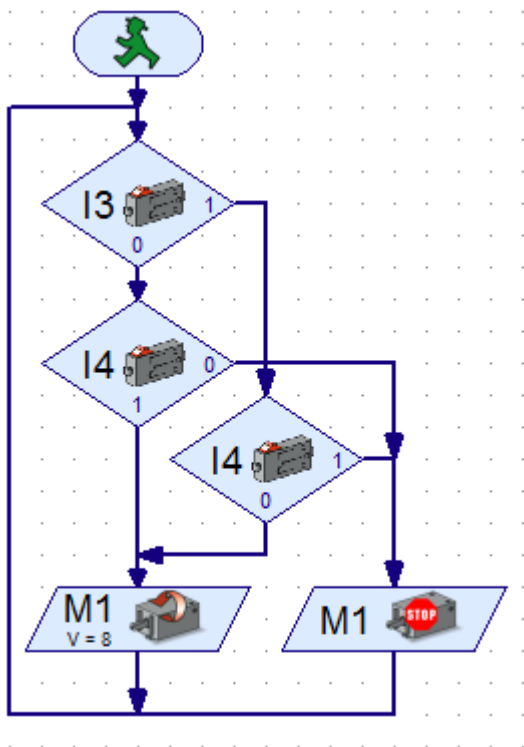
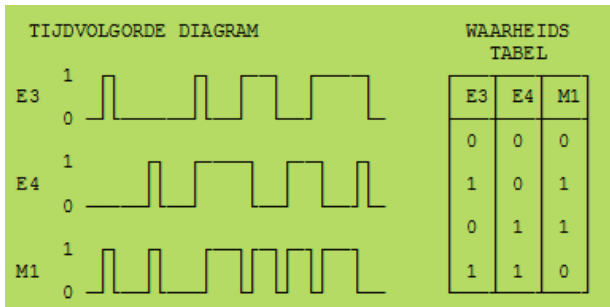


```

BASC0M-AVR
Compiler gegevens en instellingen.
(zie bijlage:2)
Do
  If E3 = 1 Or E4 = 1 Then
    Call Motor1_stop
  Else
    Call Motor1_links
  End If
Loop
Subroutines
(zie bijlage:3)
    
```

VOORBEELD-07: EXCLUSIVE-OR SCHAKELING.

Alleen als één van beide drukknoppen (E3 of E4) ingedrukt wordt, dan gaat de draaischijf (M1) linksom draaien



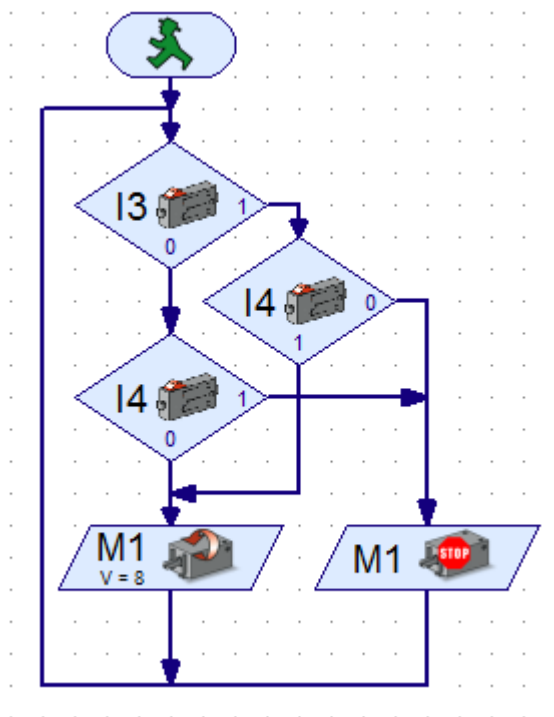
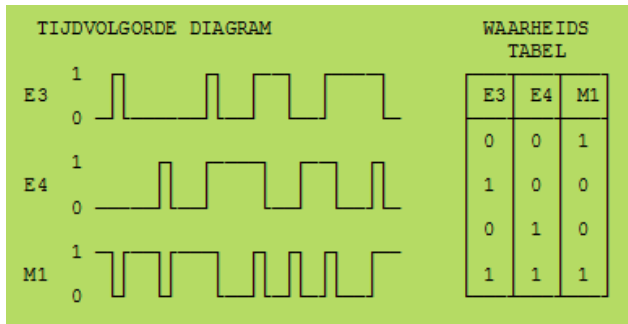
```

BASCOS-AVR
Compiler gegevens en instellingen.
(zie bijlage:2)
Do
  If E3 = 1 And E4 = 0 Then
    Call Motor1_links
  Else
    If E3 = 0 And E4 = 1 Then
      Call Motor1_links
    Else
      Call Motor1_stop
    End If
  End If
Loop
Subroutines
(zie bijlage:3)
    
```

VOORBEELD-08: EXCLUSIVE-NOR SCHAKELING.

Als programma start dan gaat de draaischijf (M1) linksom draaien.

De draaischijf stopt als één van beide drukknoppen (E3 of E4) ingedrukt wordt.



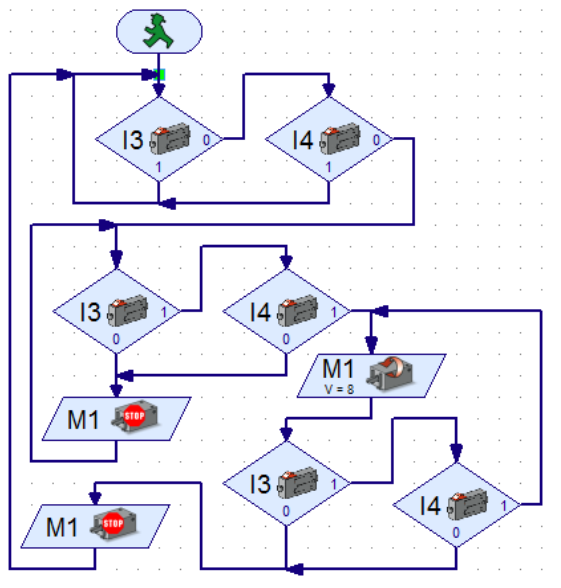
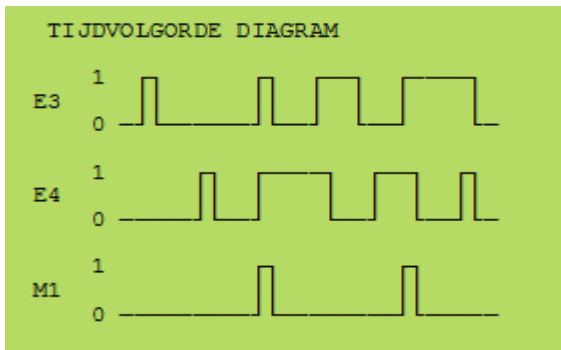
```

BASC0M-AVR
Compiler gegevens en instellingen.
(zie bijlage:2)
Do
  If E3 = 1 And E4 = 0 Then
    Call Motor1_stop
  Else
    If E3 = 0 And E4 = 1 Then
      Call Motor1_stop
    Else
      Call Motor1_links
    End If
  End If
Loop
Subroutines
(zie bijlage:3)
    
```

VOORBEELD-09: VEILIGHEIDS-SCHAKELING mbv AND.

Alleen als BEIDE drukknoppen (E3 en E4) ingedrukt worden, dan gaat de draaischijf (M1) linksom draaien. De draaischijf stopt zodra één van de drukknoppen wordt losgelaten. Herstart is pas weer mogelijk nadat beide drukknoppen zijn los gelaten.

Deze schakeling kan bijvoorbeeld worden toegepast als twee-handen-bediening voor een persschaar.



```
BASCOM-AVR
Compiler gegevens en instellingen.
(zie bijlage:2)
Do
Select Case Task

  Locate 2 , 1
  Lcd "task = " ; Task

  Case 0:
    If E3 = 0 And E4 = 0 Then
      Var = 0
      Task = 1
    End If

  Case 1:
    Locate 1 , 1
    Lcd "var = " ; Var

    Var = Var + 1
    If Var > 100 Then Task = 2

  Case 2:
    If E3 = 1 And E4 = 1 Then Task = 3

  Case 3:
    Do
      Call Motor1_links
      Task = 4
    Loop Until E3 = 0 Or E4 = 0

  Case 4 :
    Call Motor1_stop
    Task = 0

End Select
Loop
Subroutines
(zie bijlage:3)
```

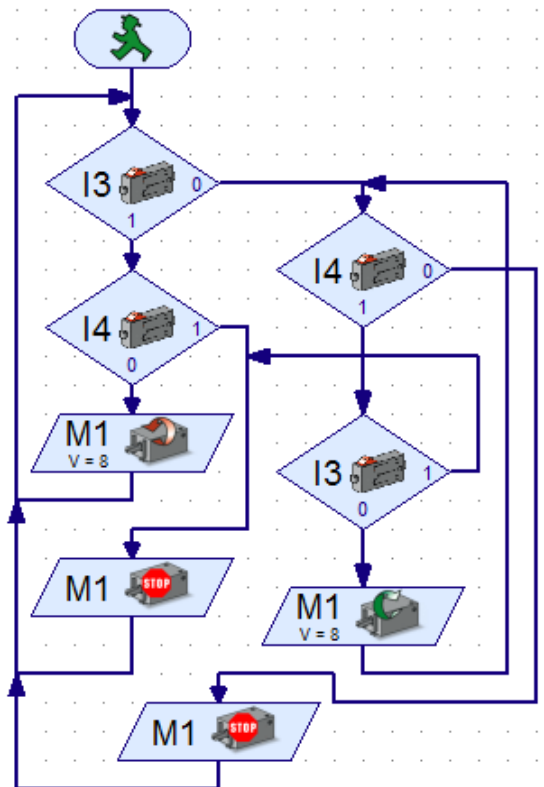
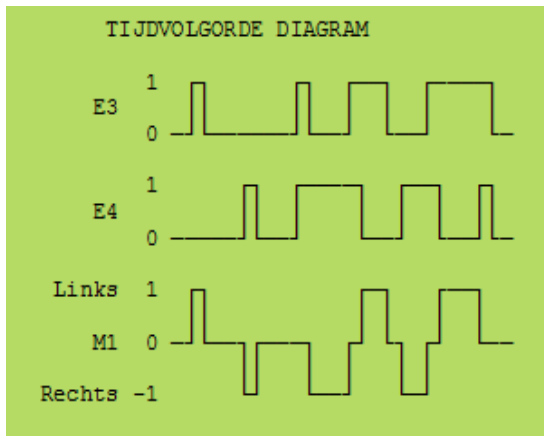
VOORBEELD-10: JOYSTICK SCHAKELING (1), Mechanische beveiliging.

Als drukknop (E3) is ingedrukt, dan gaat de draaischijf (M1) linksom draaien.

Als drukknop (E4) is ingedrukt, dan gaat de draaischijf (M1) rechtsom draaien.

De draaischijf staat stil als beide drukknoppen (E3 en E4) worden ingedrukt.

Deze schakeling kan worden toegepast indien gebruik wordt gemaakt van een z.g. joystick-knuppel. De drukknoppen E3 en E4 liggen dan tegenover elkaar.

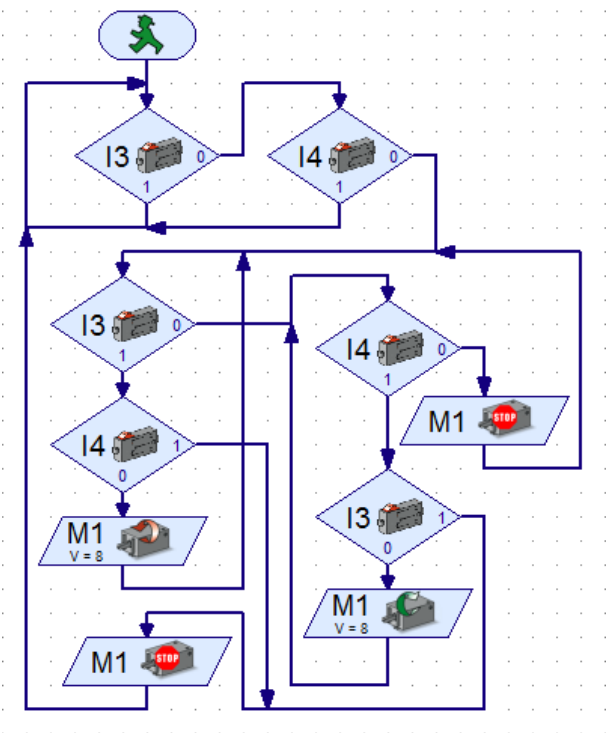
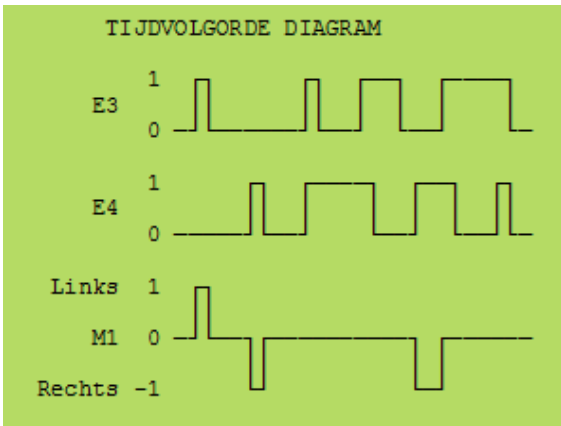


```

BASCOM-AVR
Compiler gegevens en instellingen.
(zie bijlage:2)
Do
  If E3 = 0 And E4 = 0 Then Call Motor1_stop
  If E3 = 1 And E4 = 1 Then Call Motor1_stop
  If E3 = 1 And E4 = 0 Then Call Motor1_links
  If E3 = 0 And E4 = 1 Then Call Motor1_rechts
Loop
Subroutines
(zie bijlage:3)
    
```

VOORBEELD-11: JOYSTICK SCHAKELING (2), Softwarematige beveiliging.

Als drukknop (E3) is ingedrukt, dan gaat de draaischijf (M1) linksom draaien.
 Als drukknop (E4) is ingedrukt, dan gaat de draaischijf (M1) rechtsom draaien.
 De draaischijf staat stil als beide drukknoppen (E3 en E4) worden ingedrukt.
 Herstart is pas weer mogelijk als beide drukknoppen zijn losgelaten
Deze schakeling kan worden toegepast indien géén gebruik wordt gemaakt van een z.g. joystick-knuppel, maar van 2 losse drukknoppen, zoals dat in dit model is toegepast.



```

BASCOM-AVR
Compiler gegevens en instellingen.
(zie bijlage:2)
Do
Select Case Task

If Task <> 0 Then
    Locate 2 , 1
    Lcd "task = " ; Task
End If

Case 0:
    If E3 = 1 And E4 = 1 Then Task = 3
    If E3 = 1 And E4 = 0 Then Task = 1
    If E3 = 0 And E4 = 1 Then Task = 2
    If E3 = 0 And E4 = 0 Then Task = 4

Case 1:
    Call Motor1_links
    Task = 0

Case 2:
    Call Motor1_rechts
    Task = 0

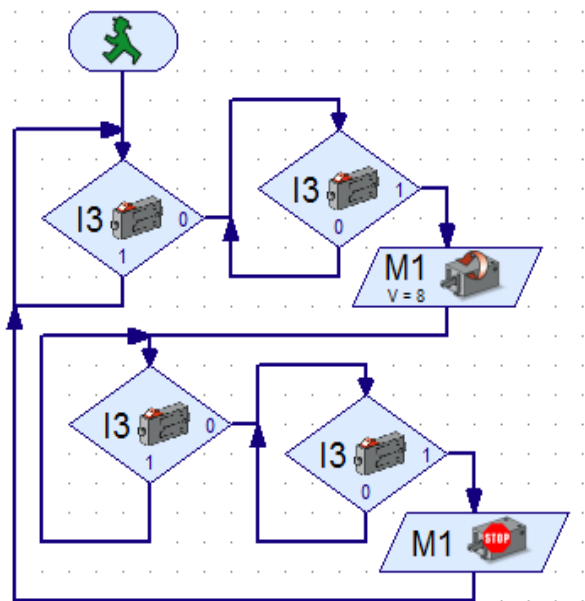
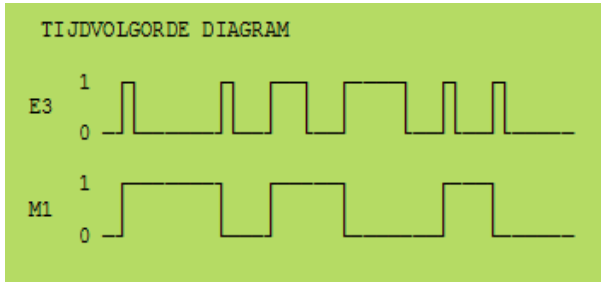
Case 3:
    Do
        Call Motor1_stop
    Loop Until E3 = 0 And E4 = 0
    Task = 0

Case 4:
    Call Motor1_stop
    Task = 0
End Select
Loop
Subroutines
(zie bijlage:3)
    
```

VOORBEELD-12: GEHEUGENSCHAKELING (1).

Als drukknop (E3) ingedrukt wordt, dan gaat de draaischijf (M1) linksom draaien. Zodra de linker drukknop (E3) nogmaals wordt ingedrukt, dan stopt de draaischijf (M1) gaat dus aan c.q. uit op het moment dat de drukknop ingedrukt wordt.

Dit houdt in dat de schakeling werkt op de "opgaande flank" van de drukknop (E3= 0 -> 1).



```

BASCOS-AVR
Compiler gegevens en instellingen.
(zie bijlage:2)
Do
Select Case Task

If Task <> 0 Then
  Locate 2 , 1
  Lcd "task = " ; Task
End If

Case 0:
  If E3 = 1 And E4 = 1 Then Task = 3
  If E3 = 1 And E4 = 0 Then Task = 1
  If E3 = 0 And E4 = 1 Then Task = 2
  If E3 = 0 And E4 = 0 Then Task = 4

Case 1:
  Call Motor1_links
  Task = 0

Case 2:
  Call Motor1_rechts
  Task = 0

Case 3:
  Do
    Call Motor1_stop
  Loop Until E3 = 0 And E4 = 0
  Task = 0

Case 4:
  Call Motor1_stop
  Task = 0

End Select
Loop

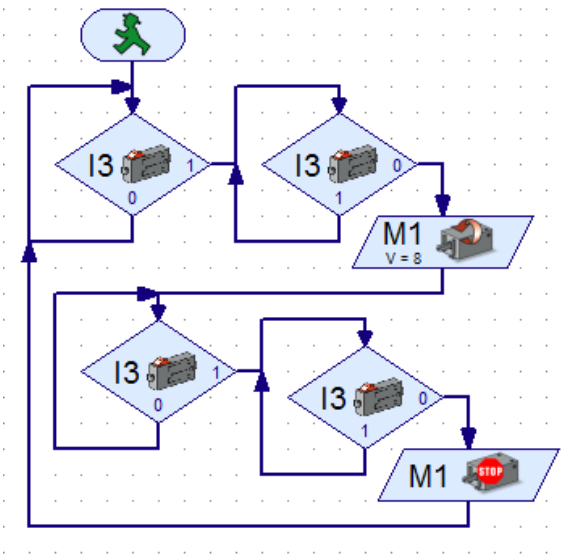
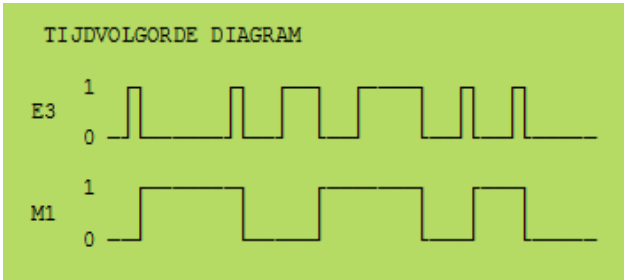
Subroutines
(zie bijlage:3)

```

VOORBEELD-13: GEHEUGENSCHAKELING (2).

Als drukknop (E3) na het indrukken wordt losgelaten, dan gaat de draaischijf (M1) linksom draaien. Zodra drukknop (E3) nogmaals na indrukken wordt losgelaten, dan stopt de draaischijf. M1 gaat dus aan c.q. uit op het moment dat de drukknop losgelaten wordt.

Dit houdt in dat de schakeling werkt op de "neergaande flank" van de drukknop (E3= 1 -> 0).



```

BASCOS-AVR
Compiler gegevens en instellingen.
(zie bijlage:2)
Do
Select Case Task

If Task <> 0 Then
Locate 2 , 1
Lcd "task = " ; Task
End If

Case 0:
If E3 = 1 Then Task = 1

Case 1:
If E3 = 0 Then
Task = 2
Call Motor1_links
End If

Case 2:
If E3 = 1 Then Task = 3

Case 3:
If E3 = 0 Then
Task = 0
Call Motor1_stop
End If

End Select
Loop
Subroutines
(zie bijlage:3)
    
```

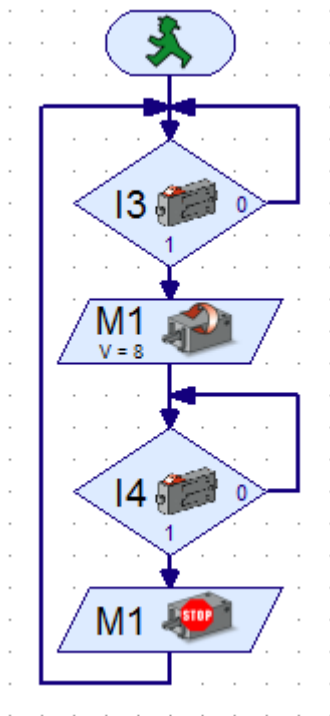
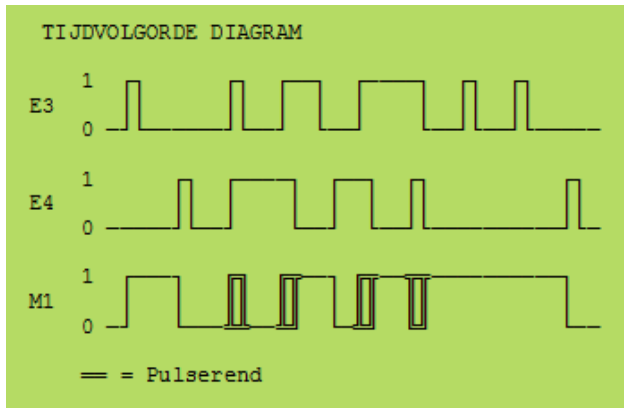

VOORBEELD-14: GEHEUGENSCHAKELING (3).

Als drukknop (E3) wordt ingedrukt, dan gaat de draaischijf (M1) linksom.

Als drukknop (E4) wordt ingedrukt, dan stopt de draaischijf.

Als beide drukknoppen (E3 en E4) worden ingedrukt, dan gaat de draaischijf (M1) pulserend aan/uit.

Dit is een ongewenste situatie. Zie voor een verbeterde schakeling voorbeeld-15.



```

BASCOM-AVR
Compiler gegevens en instellingen.
(zie bijlage:2)
Do
  Select Case Task
    'If Task <> 0 Then
      Locate 2 , 1
      Lcd "task = " ; Task
    'End If

    Case 0:
      If E3 = 1 Then
        Task = 1
        Call Motor1_links
      End If

    Case 1:
      If E4 = 1 Then
        Task = 0
        Call Motor1_stop
      End If

  End Select

Loop
  Subroutines
  (zie bijlage:3)
  
```

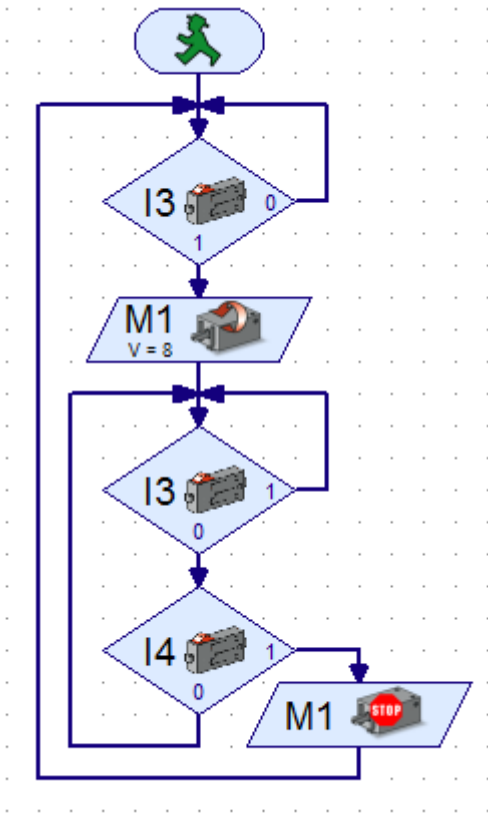
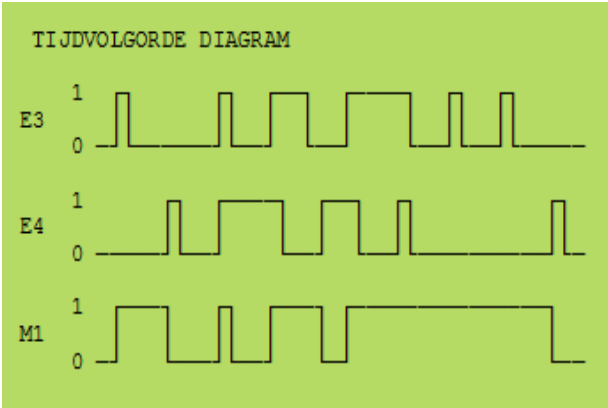
VOORBEELD-15: GEHEUGENSCHAKELING (4).

Als drukknop (E3) wordt ingedrukt, dan gaat de draaischijf (M1) linksom draaien.

Als drukknop (E4) wordt ingedrukt, dan stopt de draaischijf.

Indien beide drukknoppen (E3 en E4) worden ingedrukt, dan blijft de draaischijf (M1) linksom draaien.

Dit houdt in dat hier sprake is van een dominantie (AAN=DOMINANT).



```

BASCOM-AVR
Compiler gegevens en instellingen.
(zie bijlage:2)
Do
  Select Case Task

    Locate 2 , 1
    Lcd "task = " ; Task

    Case 0:
      If E3 = 1 Then
        Task = 1
        Call Motor1_links
      End If

    Case 1:
      If E3 = 0 And E4 = 1 Then
        Task = 0
        Call Motor1_stop
      End If

  End Select
Loop
Subroutines
(zie bijlage:3)
    
```

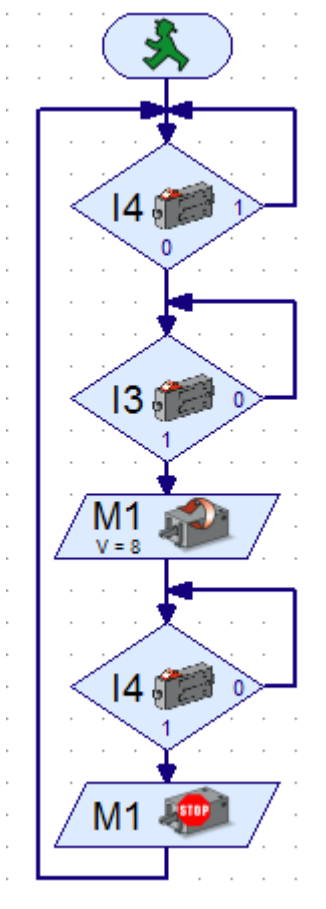
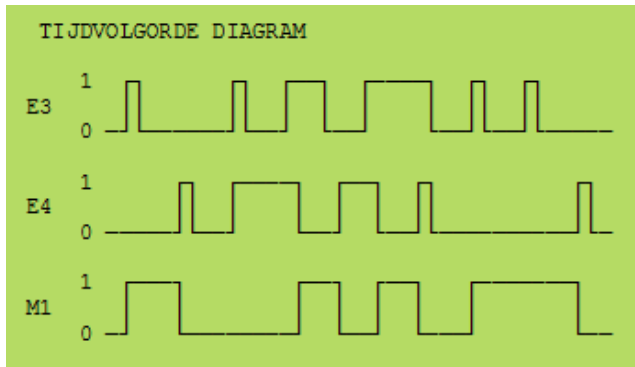
VOORBEELD-16: GEHEUGENSCHAKELING (5).

Als drukknop (E3) wordt ingedrukt, dan gaat de draaischijf (M1) linksom draaien.

Als drukknop (E4) wordt ingedrukt, dan stopt de draaischijf.

Indien beide drukknoppen (E3 en E4) worden ingedrukt, dan stopt de draaischijf.

Dit houdt in dat hier sprake is van een dominantie (UIT=DOMINANT).



```

BASCOM-AVR
Compiler gegevens en instellingen.
(zie bijlage:2)
Do
Select Case Task

  Locate 2 , 1
  Lcd "task = " ; Task

  Case 0:
    If E4 = 0 Then Task = 1

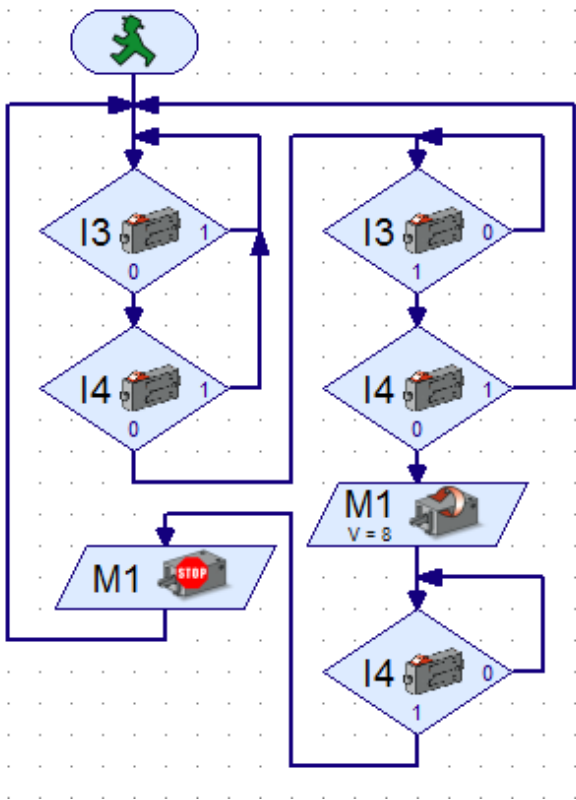
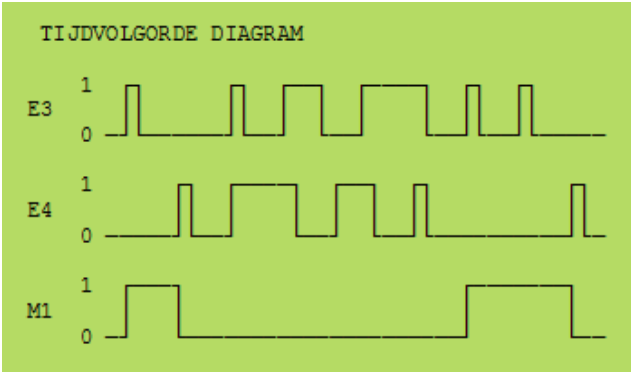
  Case 1:
    If E3 = 1 Then
      Task = 2
      Call Motor1_links
    End If

  Case 2:
    If E4 = 1 Then
      Task = 0
      Call Motor1_stop
    End If

End Select
Loop
Subroutines
(zie bijlage:3)
    
```

VOORBEELD-17: GEHEUGENSCHAKELING (6).

Als drukknop (E3) wordt ingedrukt, dan gaat de draaischijf (M1) linksom draaien.
 Als drukknop (E4) wordt ingedrukt, dan stopt de draaischijf.
 De draaischijf stopt ook indien beide drukknoppen (E3 en E4) worden ingedrukt(UIT=DOMINANT).
 Herstart is pas weer mogelijk nadat beide drukknoppen zijn los gelaten.



```

BASCOM-AVR
Compiler gegevens en instellingen.
(zie bijlage:2)
Do
Select Case Task

Locate 2 , 1
Lcd "task = " ; Task

Case 0:
If E3 = 0 And E4 = 0 Then Task = 1

Case 1:
If E3 = 1 And E4 = 0 Then
Task = 2
Call Motor1_links
End If

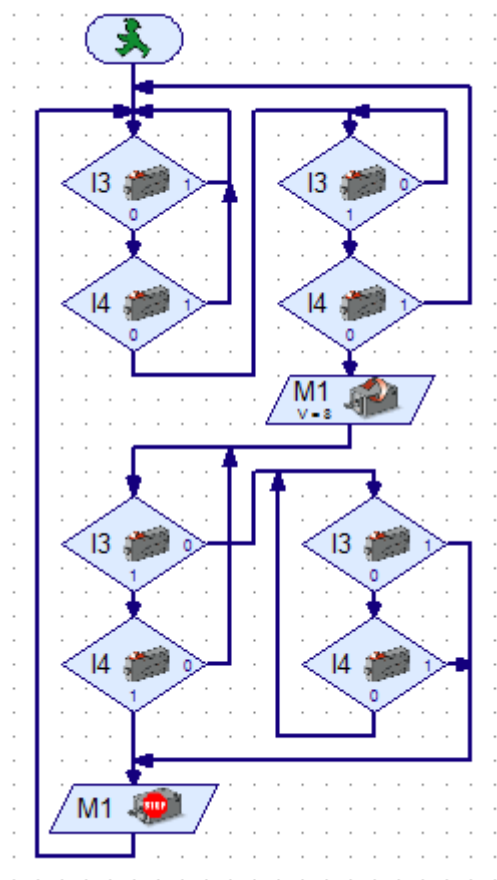
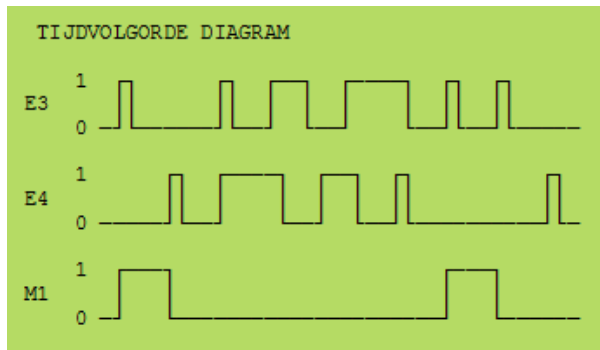
If E3 = 1 And E4 = 1 Then
Do
Call Motor1_stop
Task = 0
Loop Until E3 = 0 And E4 = 0
End If

Case 2:
If E4 = 1 Then
Task = 0
Call Motor1_stop
End If

End Select
Loop
Subroutines
(zie bijlage:3)
    
```

VOORBEELD-18: GEHEUGENSCHAKELING (7).

Als drukknop (E3) wordt ingedrukt, dan gaat de draaischijf (M1) linksom draaien.
 Als drukknop E3 en/of E4 wordt ingedrukt, dan stopt de draaischijf. (UIT=DOMINANT)
 Herstart is pas weer mogelijk nadat beide drukknoppen zijn losgelaten.



```

BASCOM-AVR
Compiler gegevens en instellingen.
(zie bijlage:2)
Do
Select Case Task
'If Task <> 0 Then
  Locate 2 , 1
  Lcd "task = " ; Task
'End If

Case 0:
  If E3 = 0 And E4 = 0 Then Task = 1

Case 1:
  If E3 = 1 And E4 = 0 Then
    Task = 2
    Call Motor1_links
  End If

  If E3 = 1 And E4 = 1 Then
    Task = 0
  End If

Case 2:
  If E3 = 0 Then Task = 3

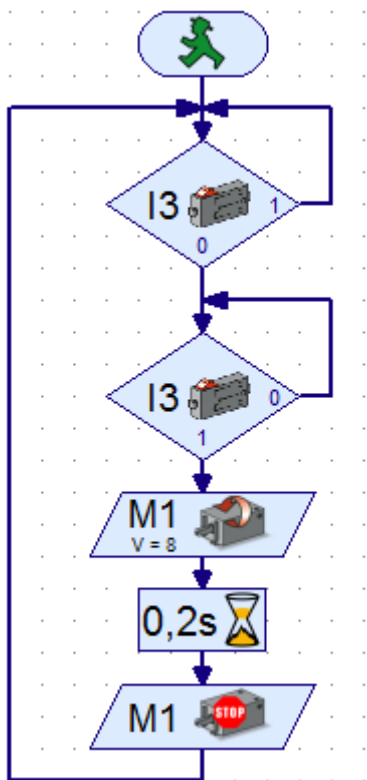
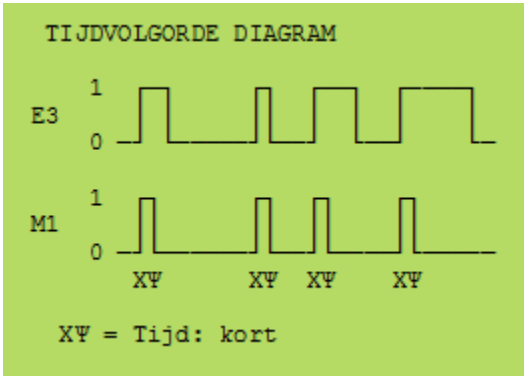
  If E3 = 1 And E4 = 1 Then
    Task = 0
    Call Motor1_stop
  End If

Case 3:
  If E3 = 1 Or E4 = 1 Then
    Task = 0
    Call Motor1_stop
  End If

End Select
Loop
Subroutines
(zie bijlage:3)
    
```

VOORBEELD-19: TIJDSCHAKELING KORT (ONE-SHOT).

Als drukknop (E3) wordt ingedrukt, dan gaat de draaischijf (M1) linksom draaien. Na korte tijd stopt de draaischijf automatisch. De tijdsduur van het AAN zijn van motor M1 is onafhankelijk van de tijdsduur van het intoetsen van drukknop E3 maar is afhankelijk van de waarde van een tijdvariabele (var).



```

BASCOS-AVR
Compiler gegevens en instellingen.
(zie bijlage:2)
Do
Select Case Task

Locate 2 , 1
Lcd "task = " ; Task

Case 0:
If E3 = 0 Then Task = 1

Case 1:
If E3 = 1 Then
Task = 2
Var = 0
Call Motor1_links
End If

Case 2:
Var = Var + 1
If Var >= 100 Then
Task = 0
Call Motor1_stop
End If

End Select
Loop
Subroutines
(zie bijlage:3)
    
```

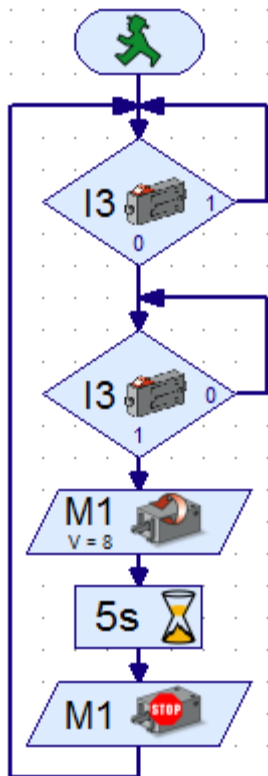
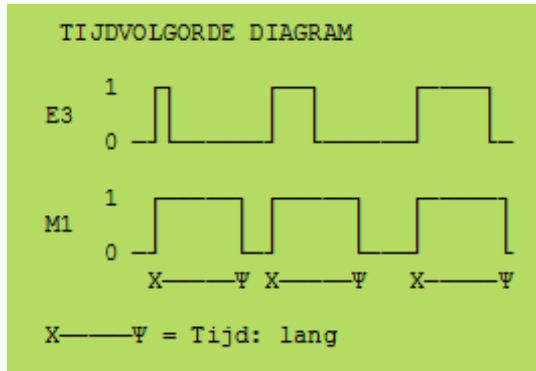
VOORBEELD-20: TIJDSCHAKELING LANG (1).

Als drukknop (E3) wordt ingedrukt, dan gaat de draaischijf (M1) linksom draaien.

Na verloop van tijd stopt de draaischijf automatisch.

De tijdsduur van het AAN zijn van motor M1 is onafhankelijk van de tijdsduur van het intoetsen van drukknop E3 maar is afhankelijk van de waarde van een tijdvariabele (var).

Bij de korte tijdschakeling is de motor reeds gestopt voordat de drukknop wordt losgelaten.



```

BASCOM-AVR
Compiler gegevens en instellingen.
(zie bijlage:2)
Do
Select Case Task
    Locate 2 , 1
    Lcd "task = " ; Task

    Case 0:
        If E3 = 0 Then Task = 1

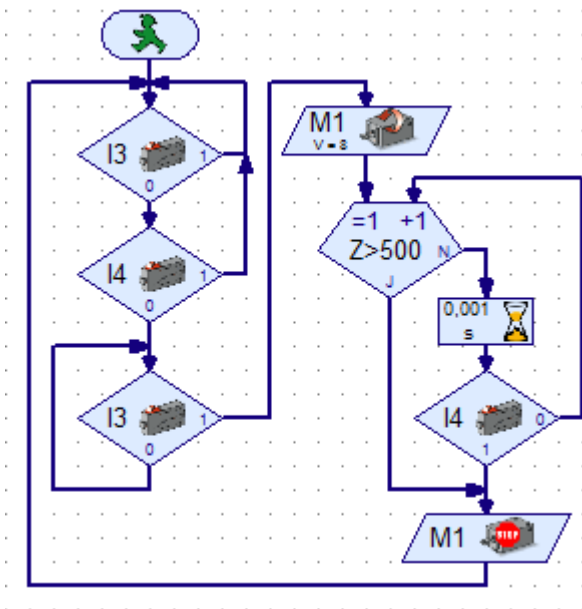
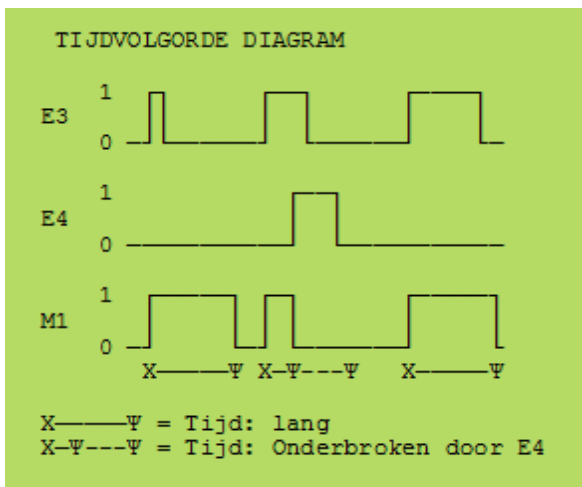
    Case 1:
        If E3 = 1 Then
            Task = 2
            Var = 0
            Call Motor1_links
        End If

    Case 2:
        Var = Var + 1
        If Var >= 500 Then
            Task = 0
            Call Motor1_stop
        End If

End Select
Loop
Subroutines
(zie bijlage:3)
    
```

VOORBEELD-21: TIJDSCHAKELING LANG (2).

Zodra drukknop (E3) wordt ingedrukt, dan gaat de draaischijf (M1) linksom draaien.
 Na verloop van tijd stopt de draaischijf automatisch.
 De tijdsduur van het AAN zijn van motor M1 is onafhankelijk van de tijdsduur van het intoetsen van drukknop E3 maar is afhankelijk van de waarde van een tijdvariabele (var).
 Indien drukknop (E4) wordt ingedrukt, dan stopt de draaischijf (vroegtijdig).
 Herstart is pas weer mogelijk nadat beide drukknoppen zijn losgelaten.



```

BASCOS-AVR
Compiler gegevens en instellingen.
(zie bijlage:2)
Do
If E4 = 1 Then
    Task = 0
    Call Motor1_stop 'draaitafel'
End If

Select Case Task

    Locate 2 , 1
    Lcd "task = " ; Task

    Case 0:
        If E3 = 0 And E4 = 0 Then Task = 1

    Case 1:
        If E3 = 1 Then
            Task = 2
            Var = 0
            Call Motor1_links
        End If

    Case 2:
        Var = Var + 1
        If Var >= 500 Then
            Task = 0
            Call Motor1_stop
        End If

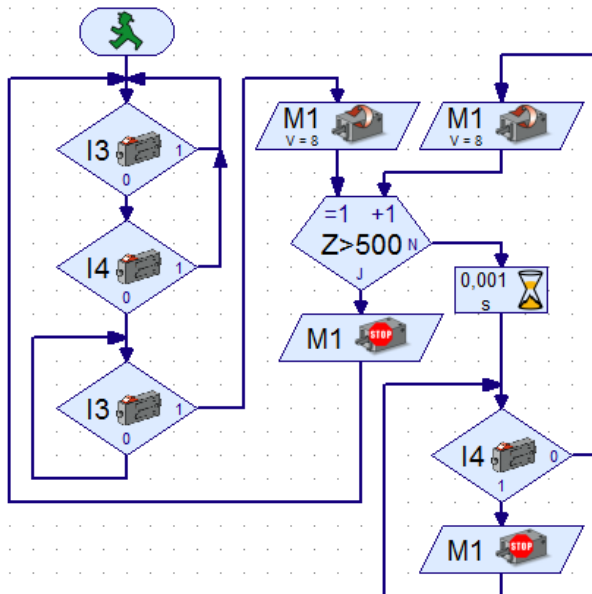
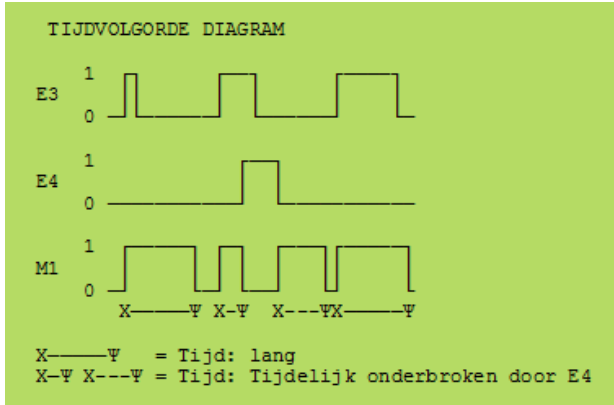
        If E4 = 1 Then
            Task = 1
            Call Motor1_stop
        End If

    End Select

Loop
Subroutines
(zie bijlage:3)
    
```


VOORBEELD-22: TIJDSCHAKELING LANG (3).

Zodra drukknop (E3) wordt ingedrukt, dan gaat de draaischijf (M1) linksom draaien. Na verloop van tijd stopt de draaischijf automatisch. De tijdsduur van het AAN zijn van motor M1 is onafhankelijk van de tijdsduur van het intoetsen van drukknop E3 maar wordt bepaald door de tijdvariable (var). Met drukknop (E4) kan de motor tijdelijk worden uitgeschakeld. Indien deze drukknop wordt ingedrukt, dan stopt de draaischijf. Zodra deze wordt losgelaten, dan gaat de draaischijf weer linksom draaien. De totale tijdsduur waarbij de motor aan blijft verandert hierdoor niet!. Dit komt omdat de tijd (var) NIET op 0 wordt gezet, zodra E4=1. Bij de vorige schakeling wordt de tijd (var) wél op 0 gezet, zodra E4=1. Herstart is pas weer mogelijk nadat beide drukknoppen zijn losgelaten.



```

BASCOS-AVR
Compiler gegevens en instellingen.
(zie bijlage:2)
Do
Select Case Task
    Locate 2 , 1
    Lcd "task = " ; Task

    Case 0:
        If E3 = 0 And E4 = 0 Then
            Task = 1
            Var = 0
        End If

    Case 1:
        If E3 = 1 Then
            Task = 2
            Call Motor1_links
        End If

    Case 2:
        Var = Var + 1
        If Var >= 1000 Then
            Task = 0
            Call Motor1_stop
        End If

        If E4 = 1 Then
            Task = 3
            Call Motor1_stop
        End If

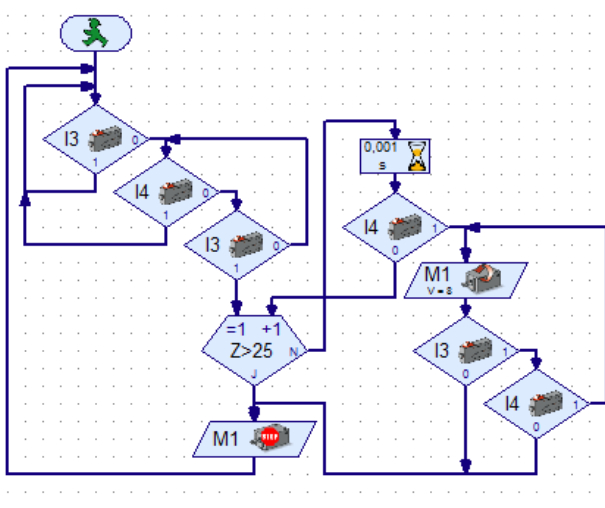
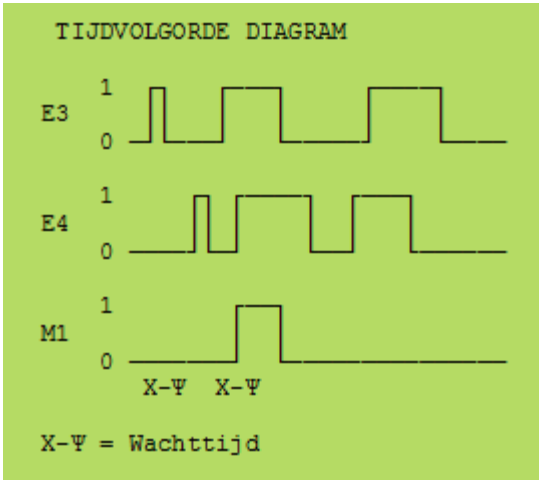
    Case 3:
        If E4 = 0 Then
            Task = 2
            Call Motor1_links
        End If

End Select
Loop
Subroutines
(zie bijlage:3)
    
```

VOORBEELD-23: VEILIGHEIDS-SCHAKELING (mbv AND en geheugenschakeling).

Deze schakeling is een verbeterde versie van de veiligheidsschakeling van voorbeeld 09. Bij deze schakeling moet eerst drukknop (E3) worden ingedrukt (en ingedrukt blijven) en daarna binnen bepaalde tijd ook drukknop (E4). Pas dán zal de draaischijf (M1) linksom draaien. De motor gaat uit zodra één van beide drukknoppen (E3 en E4) wordt losgelaten. Herstart is pas weer mogelijk nadat beide drukknoppen zijn losgelaten.

Deze schakeling kan worden toegepast als tweehanden-bediening voor bijvoorbeeld een persschaar.



```

BASCOM-AVR
Compiler gegevens en instellingen.
(zie bijlage:2)
Do
Select Case Task

  Locate 2 , 1
  Lcd "task = " ; Task

  Case 0:
    If E3 = 0 And E4 = 0 Then Task = 1

  Case 1:
    If E3 = 1 Then
      Task = 2
      Var = 0
    End If

    If E4 = 1 Then Task = 0

  Case 2:
    Var = Var + 1
    If Var < 100 Then
      If E4 = 1 Then
        Task = 3
        Call Motor1_links
      End If
    Else
      Task = 0
    End If

  Case 3:
    If E3 = 0 Or E4 = 0 Then
      Task = 0
      Call Motor1_stop
    End If

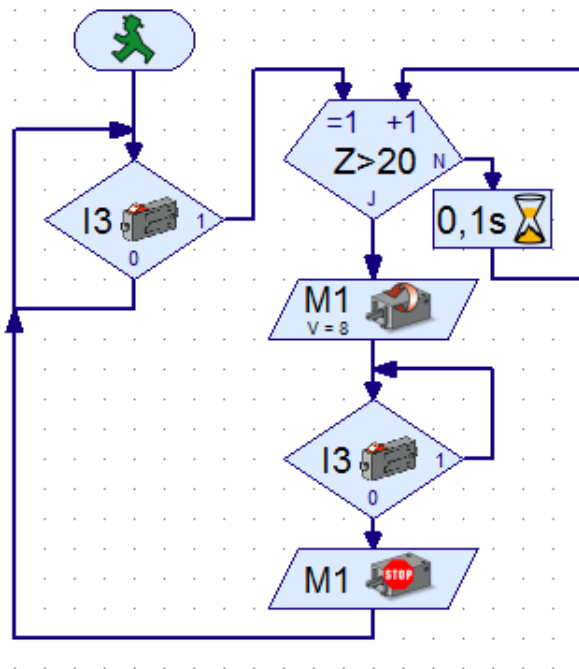
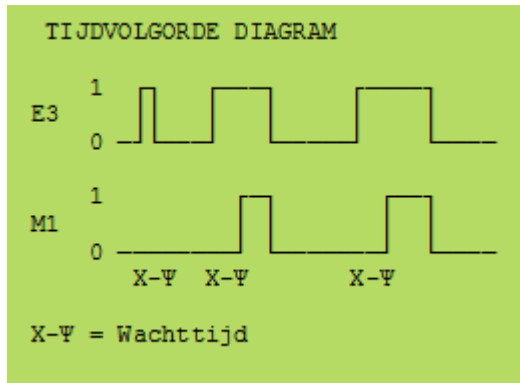
End Select
Loop
Subroutines
(zie bijlage:3)

```

VOORBEELD-24: OPKOMVERTRAGING.

Als drukknop (E3) ingedrukt wordt, (en ingedrukt blijft) dan gaat de draaischijf (M1) pas na verloop van een ingestelde tijd linksom draaien. De draaischijf stopt onmiddellijk zodra de drukknop wordt losgelaten.

De tijd-teller gaat lopen op de "opgaande flank" van E3 (0 -> 1).



```

BASCOM-AVR
Compiler gegevens en instellingen.
(zie bijlage:2)
Do
Select Case Task
  Locate 2 , 1
  Lcd "task = " ; Task

  Case 0:
    If E3 = 1 Then
      Task = 1
      Var = 0
    End If

  Case 1:
    Var = Var + 1
    If Var >= 100 Then
      Task = 2
      Call Motor1_links
    End If

    If E3 = 0 Then Task = 0

  Case 2:
    If E3 = 0 Then
      Task = 0
      Call Motor1_stop
    End If

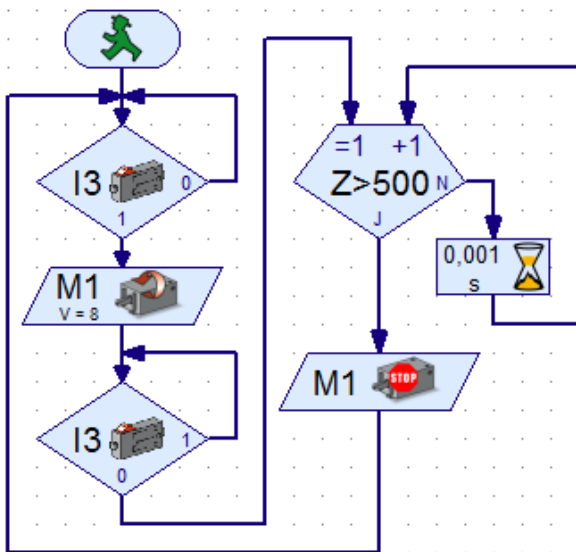
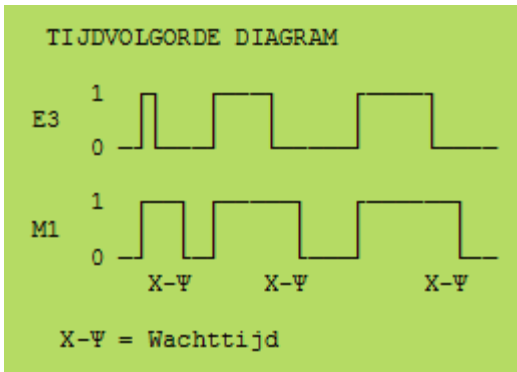
End Select
Loop
Subroutines
(zie bijlage:3)

```

VOORBEELD-25: AFVALVERTRAGING.

Als drukknop E3 ingedrukt wordt, (en ingedrukt blijft) dan gaat de draaischijf (M1) onmiddellijk linksom draaien. Zodra drukknop (E3) wordt losgelaten, stopt de draaischijf pas na verloop van de ingestelde tijd.

De tijd-teller gaat pas lopen op de neergaande flank van E3 (1 -> 0).



```

BASCOS-AVR
Compiler gegevens en instellingen.
(zie bijlage:2)
Do
Select Case Task
  Locate 2 , 1
  Lcd "task = " ; Task

  Case 0:
    If E3 = 1 Then
      Task = 1
      Call Motor1_links
    End If

  Case 1:
    If E3 = 0 Then
      Task = 2
      Var = 0
    End If

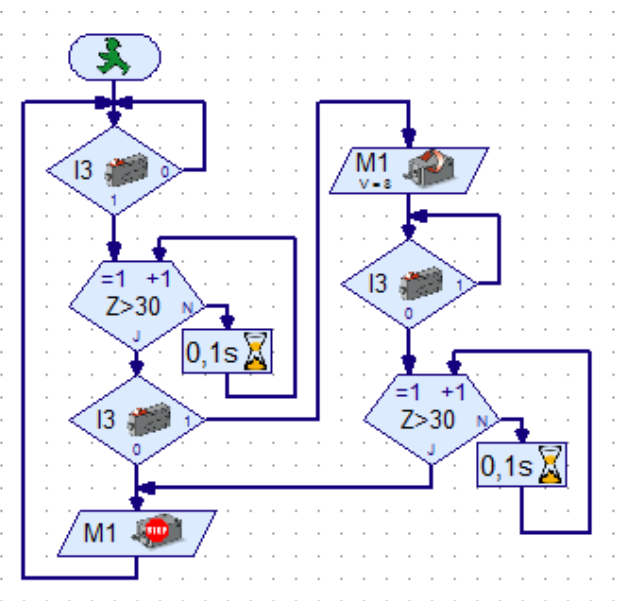
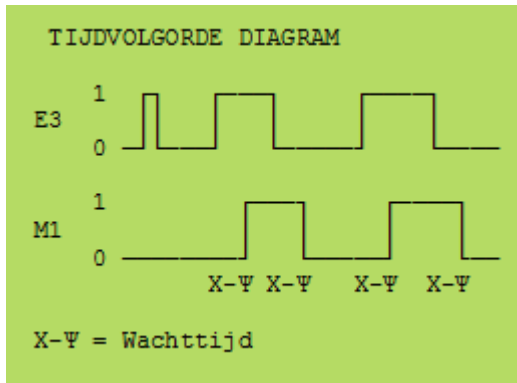
  Case 2:
    Var = Var + 1
    If Var >= 100 Then
      Task = 0
      Call Motor1_stop
    End If

End Select
Loop
Subroutines
(zie bijlage:3)
    
```

VOORBEELD-26: OPKOM- en AFVALVERTRAGING (1).

Als drukknop (E3) ingedrukt wordt, (en ingedrukt blijft) dan gaat de draaischijf pas na verloop van de ingestelde tijd (T1) linksom draaien. Zodra de drukknop (E3) wordt losgelaten, stopt de draaischijf pas na verloop van een ingestelde tijd (T2).

De tijdteller gaat lopen op zowel de opgaande flank van E3 alsook op de neergaande flank van E3.



```

BASCOM-AVR
Compiler gegevens en instellingen.
(zie bijlage:2)
Do
  Select Case Task

    Locate 2 , 1
    Lcd "task = " ; Task

    Case 0:
      If E3 = 1 Then
        Task = 1
        Var = 0
      End If

    Case 1:
      Var = Var + 1
      If Var >= 100 Then
        Task = 2
        Call Motor1_links
      End If

      If E3 = 0 Then Task = 0

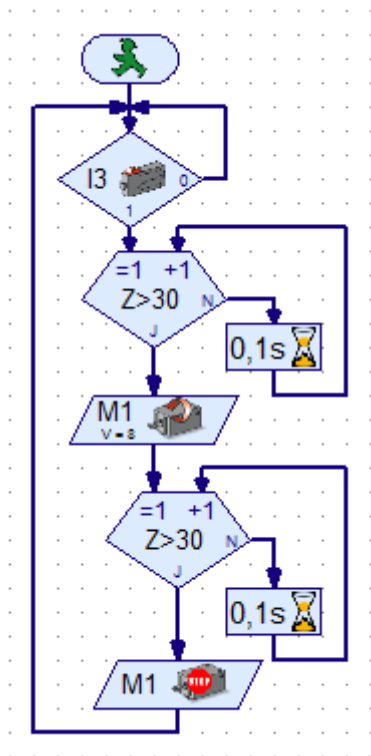
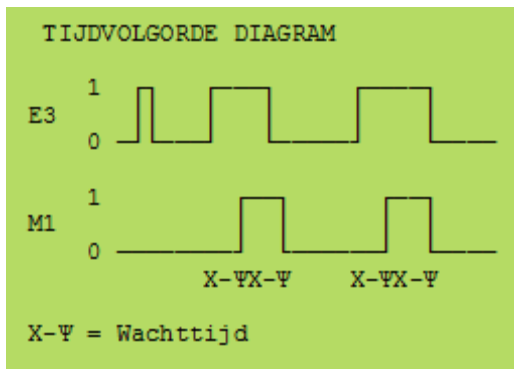
    Case 2:
      If E3 = 0 Then
        Task = 3
        Var = 0
      End If

    Case 3:
      Var = Var + 1
      If Var >= 100 Then
        Task = 0
        Call Motor1_stop
      End If

  End Select
Loop
  Subroutines
  (zie bijlage:3)
  
```

VOORBEELD-27: OPKOM- en AFVALVERTRAGING (2).

Als drukknop (E3) ingedrukt wordt, dan gaat de draaischijf (M1) pas na verloop van de ingestelde tijd (T1) linksom draaien. De motor gaat na verloop van de ingestelde tijd (T2) automatisch uit.



```

BASCOS-AVR
Compiler gegevens en instellingen.
(zie bijlage:2)
Do
Select Case Task

Locate 2 , 1
Lcd "task = " ; Task

Case 0:
If E3 = 1 Then
Task = 1
Var = 0
End If

Case 1:
Var = Var + 1
If Var >= 100 Then
Task = 2
Call Motor1_links
End If

Case 2:
If E3 = 0 Then
Task = 3
Var = 0
End If

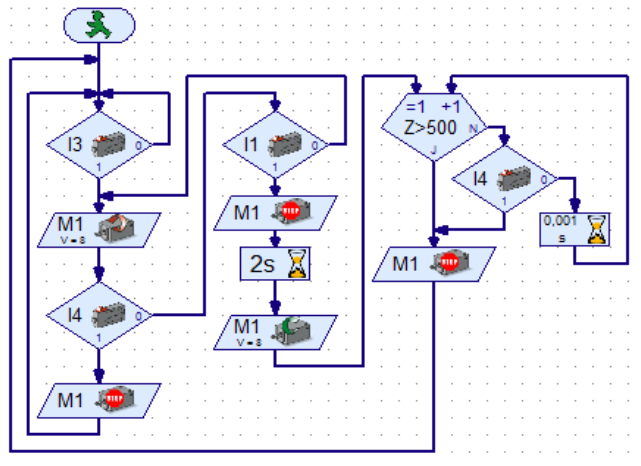
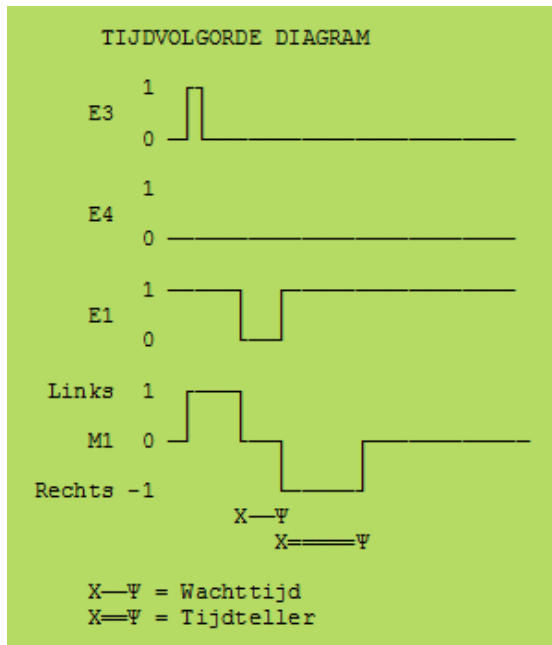
Case 3:
Var = Var + 1
If Var >= 100 Then
Task = 0
Call Motor1_stop
End If

End Select
Loop
Subroutines
(zie bijlage:3)
    
```

VOORBEELD-28: POSITIONERING (1).

Als drukknop (E3) ingedrukt wordt, dan gaat de draaischijf (M1) eerst linksom draaien zolang drukknop E1 (startpositie) niet is ingedrukt. Hierna volgt een korte pauze. Dan gaat de draaischijf een bepaalde tijdseenheid (mbv tijdteller) rechtsom draaien. Zodra drukknop (E4) wordt ingedrukt, dan stopt de draaischijf onmiddellijk.

E4 dient dus als veiligheids- of noodstop-drukknop.



```

BASCOM-AVR
Compiler gegevens en instellingen.
(zie bijlage:2)
Do
  If E4 = 1 Then
    Task = 0
    Call Motor1_stop
  End If

  Select Case Task

    Locate 2 , 1
    Lcd "task = " ; Task

    Case 0:
      If E3 = 0 Then Task = 1

    Case 1:
      If E3 = 1 Then Task = 2

    Case 2:
      Call Motor1_links

      If E1 = 1 Then
        Task = 3
        Var = 0
        Call Motor1_stop
      End If

    Case 3:
      Var = Var + 1
      If Var >= 200 Then
        Task = 4
        Var = 0
        Call Motor1_rechts
      End If

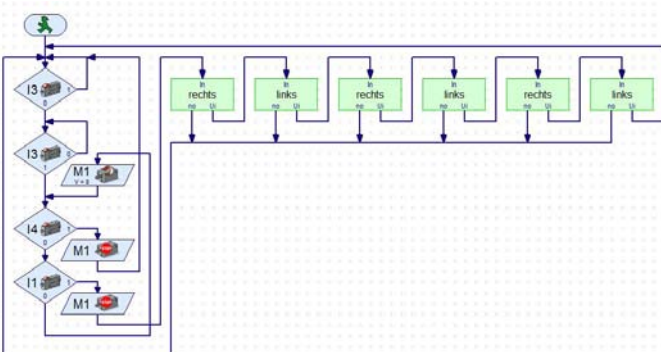
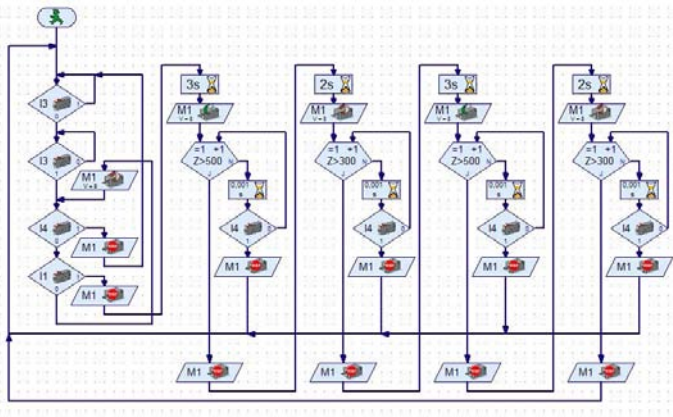
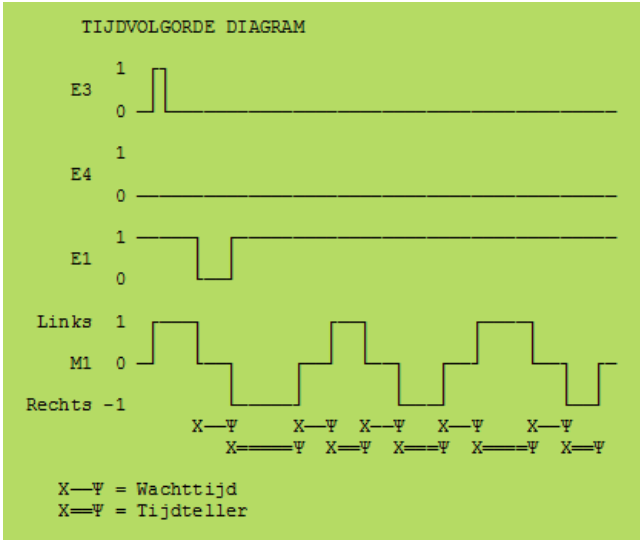
    Case 4:
      Var = Var + 1
      If Var >= 500 Then
        Task = 0
        Call Motor1_stop
      End If

  End Select
Loop
Subroutines
(zie bijlage:3)
  
```

VOORBEELD-29: POSITIONERING (2).

Als drukknop (E3) ingedrukt wordt, dan gaat de draaischijf (M1) eerst linksom draaien zolang drukknop E1 (startpositie) niet is ingedrukt. Hierna volgt een korte pauze. Dan gaat de draaischijf een bepaalde tijdseenheid (mbv tijdteller) rechtsom draaien. Daarna volgt weer een korte pauze. Daarna gaat de draaischijf een bepaalde tijdseenheid (mbv tijdteller) linksom, enz. (in totaal 3x). Zodra drukknop (E4) wordt ingedrukt, dan stopt de draaischijf onmiddellijk. E4 dient dus als veiligheids- of noodstop-drukknop.

De tijdteller (var) is cumulatief, daar deze maar eenmalig op 0 wordt gesteld.



```

BASCOS-AVR
Compiler gegevens en instellingen.
(zie bijlage:2)
Do
If E4 = 1 Then
    Task = 0
    Stop Timer0
    Call Motor1_stop      'nood
Else
    Task = Oldtask
End If

Select Case Task

    Locate 2 , 1
    Lcd "task = " ; Task

    Case 0
        If E3 = 0 Then Task = 1

    Case 1
        If E3 = 1 Then Task = 2

    Case 2:
        Call Motor1_links

        If E1 = 1 Then
            Call Motor1_stop
            Var = 0
            Task = 3
            Start Timer0
        End If
    
```



```

Case 3:
  Var = Var + 1
  If Var > 200 Then
    Var = 0
    Task = 4
  End If

Case 4:
  Call Motor1_rechts

  Var = Var + 1
  If Var > 200 Then
    Var = 0
    Task = 5
    Call Motor1_stop
  End If

Case 5:
  Var = Var + 1
  If Var > 200 Then
    Var = 0
    Task = 6
  End If

Case 6:
  Call Motor1_links

  Var = Var + 1
  If Var > 200 Then
    Var = 0
    Task = 7
    Call Motor1_stop
  End If

Case 7:
  Var = Var + 1
  If Var > 200 Then
    Var = 0
    Task = 8
  End If

Case 8:
  Call Motor1_rechts

  Var = Var + 1
  If Var > 200 Then
    Var = 0
    Task = 9
    Call Motor1_stop
  End If

Case 9
  Var = Var + 1
  If Var > 200 Then
    Var = 0
    Task = 10
  End If

```

```

Case 10
  Call Motor1_links

  Var = Var + 1
  If Var > 200 Then
    Var = 0
    Task = 11
    Call Motor1_stop
  End If

Case 11:
  Var = Var + 1
  If Var > 200 Then
    Var = 0
    Task = 12
  End If

Case 12:
  Call Motor1_rechts

  Var = Var + 1
  If Var > 200 Then
    Var = 0
    Task = 13
    Call Motor1_stop
  End If

Case 13:
  Var = Var + 1
  If Var > 200 Then
    Var = 0
    Task = 14
  End If

Case 14:
  Call Motor1_links

  Var = Var + 1
  If Var > 200 Then
    Var = 0
    Task = 0
    Call Motor1_stop
    Stop Timer0
    M4 = 0

End Select

Oldtask = Task

Loop
  Subroutines
  (zie bijlage:3)

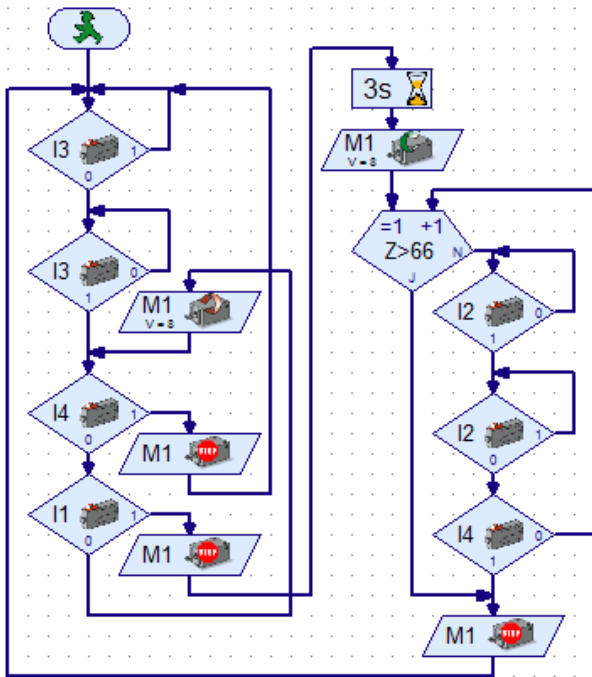
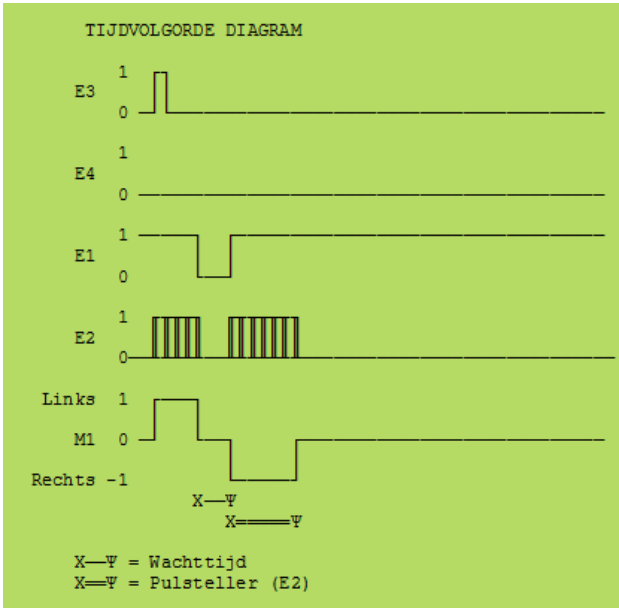
```

VOORBEELD-30: POSITIONERING (3).

Als drukknop (E3) ingedrukt wordt, dan gaat de draaischijf (M1) eerst linksom draaien zolang drukknop E1 (startpositie) niet is ingedrukt. Hierna volgt een korte pauze. Dan gaat de draaischijf een bepaald aantal posities (mbv positieteller E2) rechtsom draaien.

Als drukknop (E4) wordt ingedrukt, dan stopt de draaischijf onmiddellijk.

E4 dient dus als veiligheids- of noodstop-drukknop.



```

BASCOM-AVR
Compiler gegevens en instellingen.
(zie bijlage:2)
Do
If E4 = 1 Then
    Task = 0
    Stop Timer0
    Call Motor1_stop      'nood'
Else
    Task = Oldtask
End If

Select Case Task

    Locate 2 , 1
    Lcd "task = " ; Task

    Case 0
        If E3 = 0 Then Task = 1

    Case 1
        If E3 = 1 Then Task = 2

    Case 2:
        Call Motor1_links

        If E1 = 1 Then
            Call Motor1_stop
            E2_count = 0
            Var = 0
            Task = 3
            Start Timer0
        End If

    Case 3:
        Var = Var + 1
        If Var > 200 Then
            Var = 0
            Task = 4
        End If

```

```
Case 4:
  Call Motor1_rechts

  If E2_count > 35 Then
    Locate 2 , 13
    Lcd ; E2_count
    Var = 0
    Task = 5
    Call Motor1_stop
  End If
```

```
Case 5:
  Var = Var + 1
  If Var > 200 Then
    Var = 0
    Task = 0
    Stop Timer0
    M4=0
```

```
End Select
```

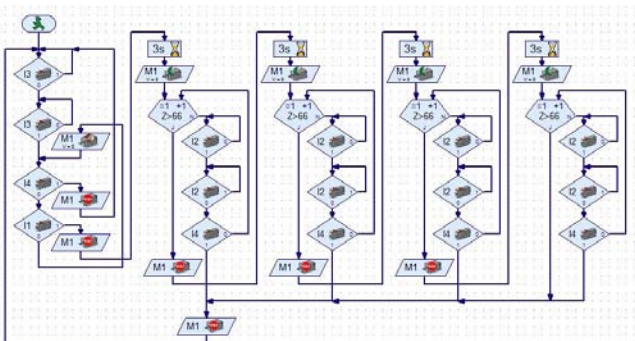
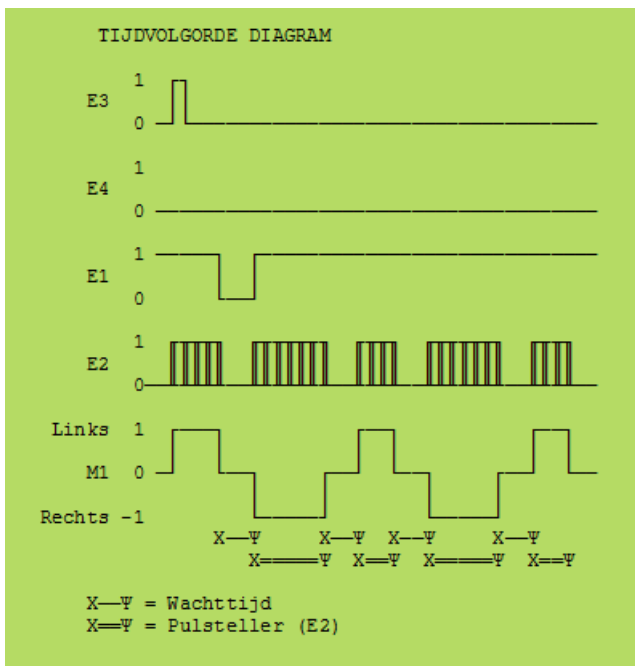
```
Oldtask = Task
```

```
Loop
```

```
Subroutines
(zie bijlage:3)
```

VOORBEELD-31: POSITIONERING (4).

Als drukknop (E3) ingedrukt wordt, dan gaat de draaischijf (M1) eerst linksom draaien zolang drukknop E1 (startpositie) niet is ingedrukt. Hierna volgt een korte pauze. Dan gaat de draaischijf een bepaald aantal posities (mbv positieteller E2) rechtsom draaien. Hierna volgt weer een korte pauze. Daarna gaan de draaischijf weer een bepaald aantal posities (mbv E2) linksom draaien, enz. (in totaal 4x). Zodra drukknop (E4) wordt ingedrukt, dan stopt de draaischijf onmiddellijk. E4 dient dus als veiligheids- of noodstop-drukknop.



```

BASCOM-AVR
Compiler gegevens en instellingen.
(zie bijlage:2)
Do
If E4 = 1 Then
    Task = 0
    Stop Timer0
    Call Motor1_stop      'nood
Else
    Task = Oldtask
End If

Select Case Task

    Locate 2 , 1
    Lcd "task = " ; Task

    Case 0
        If E3 = 0 Then Task = 1

    Case 1
        If E3 = 1 Then Task = 2

    Case 2:
        Call Motor1_links

        If E1 = 1 Then
            Call Motor1_stop
            E2_count = 0
            Var = 0
            Task = 3
            Start Timer0
        End If
    
```

```

Case 3:
  Var = Var + 1
  If Var > 300 Then
    E2_count = 0
    Task = 4
  End If

Case 4:
  Call Motor1_rechts

  If E2_count > 50 Then
    Locate 2 , 13
    Lcd ; E2_count
    Task = 5
    Call Motor1_stop
  End If

Case 5:
  Var = Var + 1
  If Var > 300 Then Task = 6

Case 6:
  Call Motor1_links

  If E2_count > 75 Then
    Locate 2 , 13
    Lcd ; E2_count
    Task = 7
    Var = 0
    Call Motor1_stop
  End If

Case 7:
  Var = Var + 1
  If Var > 300 Then Task = 8

Case 8:
  Call Motor1_rechts

  If E2_count > 100 Then
    Locate 2 , 13
    Lcd ; E2_count
    Task = 9
    Call Motor1_stop
  End If

Case 9:
  Var = Var + 1
  If Var > 300 Then Task = 10

Case 10:
  Call Motor1_links

  If E2_count > 125 Then
    Locate 2 , 13
    Lcd ; E2_count
    Task = 11
    Var = 0
    Call Motor1_stop
  End If

```

```

Case 11:
  Var = Var + 1
  If Var > 300 Then Task = 12

Case 12:
  Call Motor1_rechts

  If E2_count > 150 Then
    Locate 2 , 13
    Lcd ; E2_count
    Task = 13
    Call Motor1_stop
  End If

Case 13:
  Var = Var + 1
  If Var > 300 Then Task = 14

Case 14:
  Call Motor1_links

  If E2_count > 175 Then
    Locate 2 , 13
    Lcd ; E2_count
    Task = 15
    Var = 0
    Call Motor1_stop
  End If

Case 15:
  Var = Var + 1
  If Var > 300 Then Task = 16

Case 16:
  Call Motor1_rechts

  If E2_count > 200 Then
    Locate 2 , 13
    Lcd ; E2_count
    Task = 17
    Call Motor1_stop
  End If

Case 17:
  Var = Var + 1
  If Var > 300 Then Task = 18

Case 18:
  Call Motor1_links

  If E2_count > 225 Then
    Locate 2 , 13
    Lcd ; E2_count
    Call Motor1_stop
    Stop Timer0
    M4 = 0
    Task = 0
  End If

End Select

Oldtask = Task

```

Loop

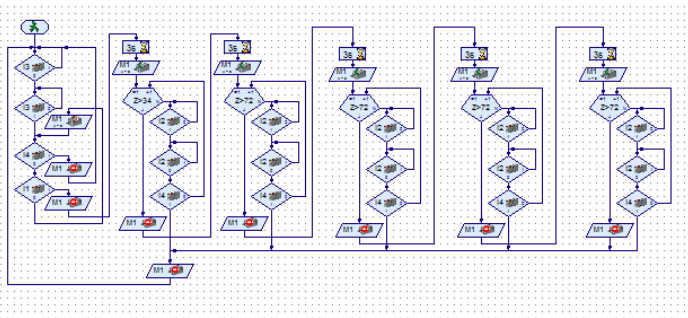
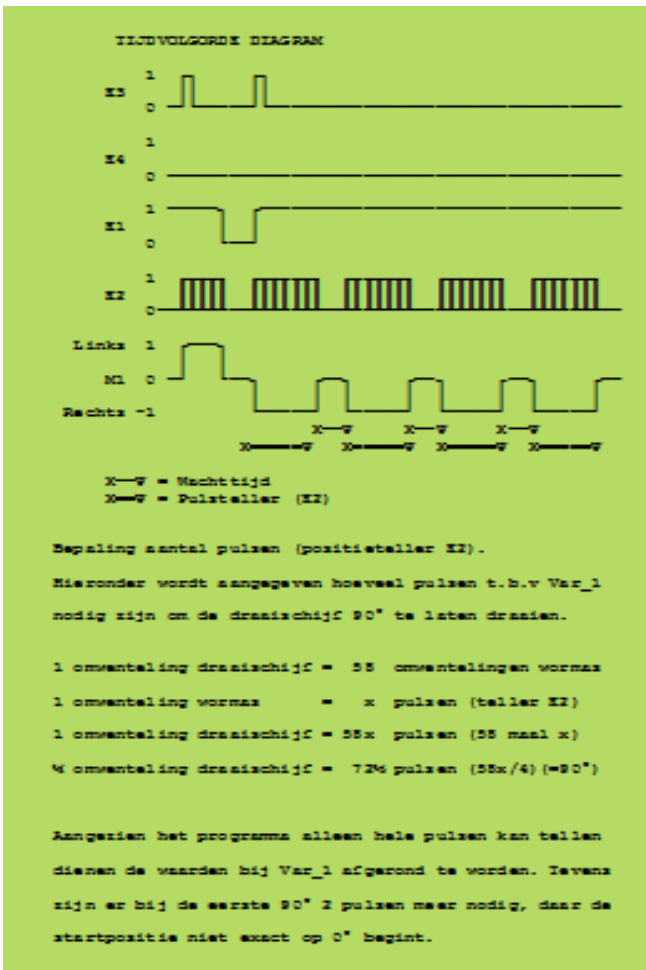
Subroutines
(zie bijlage:3)

VOORBEELD-32: POSITIONERING (5).

Als drukknop (E3) ingedrukt wordt, dan gaat de draaischijf (M1) eerst linksom draaien zolang drukknop E1 (startpositie) niet is ingedrukt. Zodra opnieuw de linker drukknop (E3) wordt ingedrukt, dan wordt het proces vervolgd. Dan gaat de draaischijf een bepaald aantal posities (mbv positieteller E2) rechtsom draaien (90°). Daarna volgt een korte pauze.

Daarna gaat de draaischijf weer 90° rechtsom, enz. (in totaal 4x).

Deze schakeling zal later worden gebruikt ten behoeve van de lasrobot.



```

BASCOS-AVR
Compiler gegevens en instellingen.
(zie bijlage:2)
Do
If E4 = 1 Then
    Task = 0
    Stop Timer0
    Call Motor1_stop 'nood'
Else
    Task = Oldtask
End If

Select Case Task

    Locate 2 , 1
    Lcd "task = " ; Task

    Case 0
        If E3 = 0 Then Task = 1
    
```

```

Case 1
  If E3 = 1 Then
    E2_count = 0
    Task = 2
  End If

Case 2:
  Call Motor1_rechts

  If E2_count > 10 Then
    Call Motor1_stop

    Locate 2 , 13
    Lcd ; E2_count

    Task = 3
    Var = 0
  End If

Case 3:
  Call Motor1_links

  If E1 = 1 Then
    Call Motor1_stop
    E2_count = 0
    Var = 0
    Task = 4
    Start Timer0
  End If

Case 4:
  Var = Var + 1
  If Var > 200 Then
    E2_count = 0
    Task = 5
  End If

Case 5:
  Call Motor1_rechts

  If E2_count > 35 Then
    Call Motor1_stop

    Locate 2 , 13
    Lcd ; E2_count

    Task = 6
    Var = 0
  End If

Case 6:
  Var = Var + 1
  If Var > 200 Then Task = 7

Case 7:
  Call Motor1_rechts

  If E2_count > 93 Then
    Call Motor1_stop

    Locate 2 , 13
    Lcd ; E2_count

    Task = 8
    Var = 0
  End If

```

```

Case 8:
  Var = Var + 1
  If Var > 200 Then Task = 9

Case 9:
  Call Motor1_rechts

  If E2_count > 151 Then
    Call Motor1_stop

    Locate 2 , 13
    Lcd ; E2_count

    Task = 10
    Var = 0
  End If

Case 10
  Var = Var + 1
  If Var > 200 Then Task = 11
Case 11:
  Call Motor1_rechts

  If E2_count > 209 Then
    Call Motor1_stop

    Locate 2 , 13
    Lcd ; E2_count

    Task = 12
    Var = 0
  End If

Case 12:
  Var = Var + 1
  If Var > 200 Then Task = 13

Case 13:
  Call Motor1_rechts

  If E2_count > 266 Then
    Call Motor1_stop

    Locate 2 , 13
    Lcd ; E2_count

    Task = 0
    Stop Timer0
    M4 = 0
  End If

End Select

Oldtask = Task

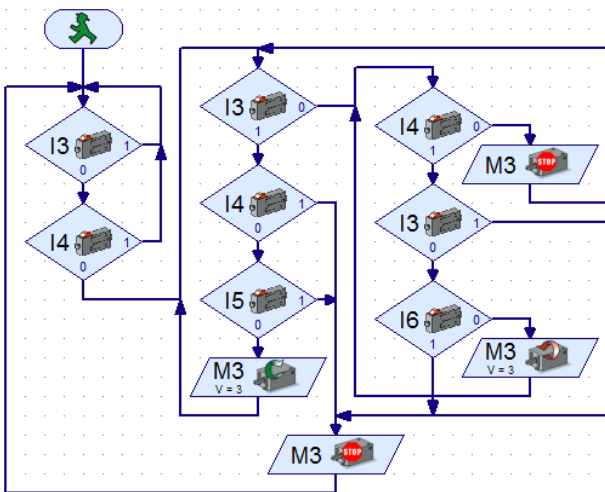
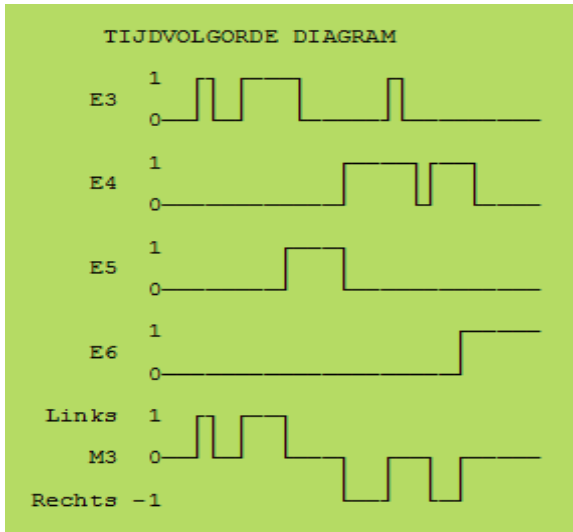
```

Loop

Subroutines
(zie bijlage:3)

VOORBEELD-33: PUNTLASROBOT OMHOOG/OMLAAG (1).

Als drukknop (E3) ingedrukt is dan gaat de puntlasrobot (M3) omlaag. De motor stopt zodra drukknop (E3) wordt losgelaten óf zodra eind-drukknop E5 is bereikt. Als drukknop (E4) ingedrukt is dan gaat de puntlasrobot (M3) omhoog. De motor stopt zodra knop (E3) wordt losgelaten óf zodra eind-drukknop E6 is bereikt. De motor gaat ook uit indien beide drukknoppen (E3 en E4) worden ingedrukt. Herstart is pas weer mogelijk nadat beide drukknoppen zijn losgelaten.



```

BASCOM-AVR
Compiler gegevens en instellingen.
(zie bijlage:2)
Do
  Select Case Task

    Locate 2 , 1
    Lcd "task = " ; Task

    Case 0
    If E3 = 0 And E4 = 0 Then Task = 1

    Case 1
    If E3 = 1 Then
      Task = 2
    Else
      Task = 3
    End If

    Case 2:
    If E4 = 1 Then
      Task = 0
      Call Motor3_stop
    ElseIf E5 = 1 Then
      Task = 0
      Call Motor3_stop
    Else
      Task = 1
      Call Motor3_down
    End If

    Case 3:
    If E4 = 1 Then
      Task = 4
    Else
      Task = 1
      Call Motor3_stop
    End If
  
```



```
Case 4:  
  If E3 = 1 Then  
    Task = 0  
    Call Motor3_stop  
  ElseIf E6 = 1 Then  
    Task = 0  
    Call Motor3_stop  
  Else  
    Task = 3  
    Call Motor3_up  
  End If
```

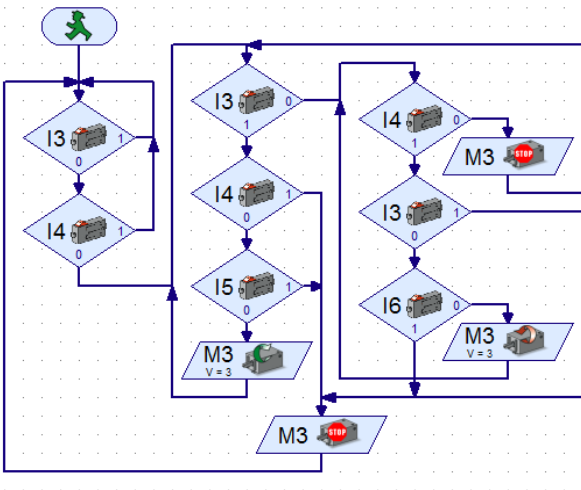
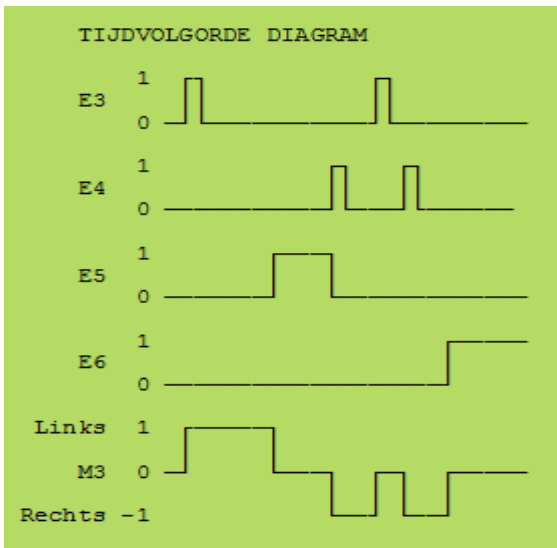
```
End Select
```

Loop

Subroutines
(zie bijlage:3)

VOORBEELD-34: PUNTLASROBOT OMHOOG/OMLAAG (2).

Als drukknop (E3) ingedrukt wordt, dan gaat de puntlasrobot (M3) omlaag. De motor stopt zodra eind-drukknop E5 is bereikt. Zodra de rechter drukknop (E4) ingedrukt wordt, dan gaat de puntlasrobot (M3) omhoog. De motor stopt zodra eind-drukknop E6 is bereikt. De motor gaat ook uit indien beide drukknoppen (E3 en E4) worden ingedrukt. Herstart is pas weer mogelijk nadt beide drukknoppen zijn losgelaten.



```

BASCOM-AVR
Compiler gegevens en instellingen.
(zie bijlage:2)
Do
Select Case Task

Locate 2 , 1
Lcd "task = " ; Task

Case 0
    If E3 = 0 And E4 = 0 Then Task = 1

Case 1
    If E3 = 1 Then
        Task = 2
    Else
        Task = 3
    End If

Case 2:
    If E4 = 1 Then
        Task = 0
        Call Motor3_stop
    ElseIf E5 = 1 Then
        Task = 0
        Call Motor3_stop
    Else
        Task = 2
        Call Motor3_down
    End If

```

```
Case 3:
  If E4 = 1 Then
    Task = 4
  Else
    Task = 1
    Call Motor3_stop
  End If

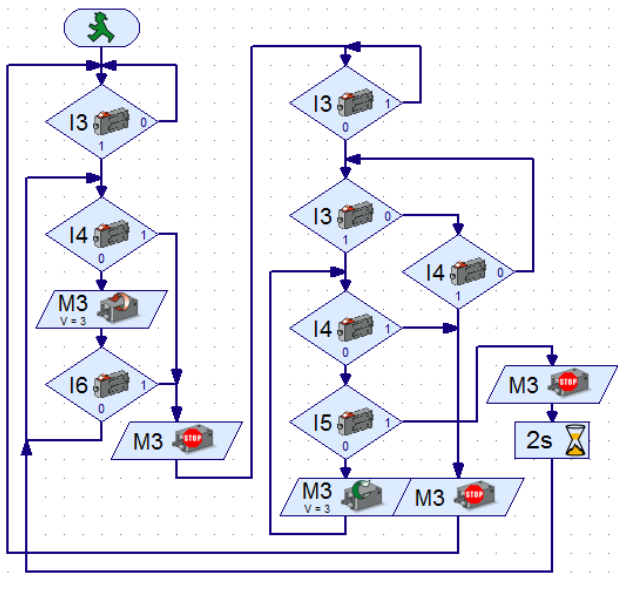
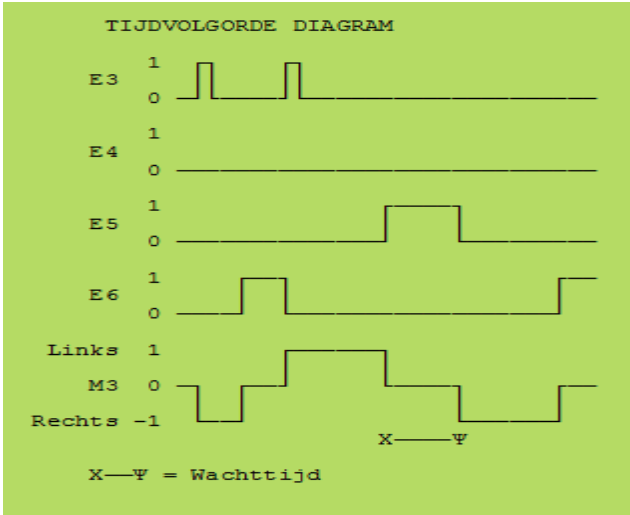
Case 4:
  If E3 = 1 Then
    Task = 0
    Call Motor3_stop
  ElseIf E6 = 1 Then
    Task = 0
    Call Motor3_stop
  Else
    Task = 4
    Call Motor3_up
  End If
End Select
```

Loop

Subroutines
(zie bijlage:3)

VOORBEELD-35: PUNTLASROBOT OMHOOG/OMLAAG (3).

Als drukknop (E3) ingedrukt wordt, dan gaat de puntlasrobot (M3) omhoog. De motor stopt zodra de eind-drukknop (E6) is bereikt (*Dit gebeurt alléén indien de puntlasrobot zich NIET in de startpositie bevindt*). De puntlasrobot bevindt zich nu in de startpositie. Als E3 nogmaals ingedrukt wordt, dan gaat de puntlasrobot omlaag. De motor stopt zodra de eind-drukknop (E5) is bereikt. Hierna volgt een korte pauze. Dan gaat de puntlasrobot weer omhoog. De motor stopt zodra de eind-drukknop (E6) is bereikt. De motor gaat ook uit zodra de rechter drukknop (E4) wordt ingedrukt. Herstart is pas weer mogelijk nadat beide drukknoppen zijn losgelaten.



```

BASCOM-AVR
Compiler gegevens en instellingen.
(zie bijlage:2)
Do
If E4 = 1 Then
    Task = 0
    Call Motor3_stop
Else
    Task = Oldtask
End If

Select Case Task

    Locate 2 , 1
    Lcd "task = " ; Task

    Case 0
        If E3 = 1 Then Task = 1

    Case 1:
        If E6 = 1 Then
            Task = 2
            Call Motor3_stop
        Else
            Call Motor3_up
        End If
    End Case
End Do
    
```

```
Case 2:
  If E3 = 0 Then Task = 3

Case 3:
  If E3 = 1 Then Task = 4

Case 4:
  If E5 = 1 Then
    Task = 5
    Var = 0
    Call Motor3_stop
  Else
    Call Motor3_down
  End If

Case 5:
  Var = Var + 1
  If Var > 150 Then Task = 1

End Select

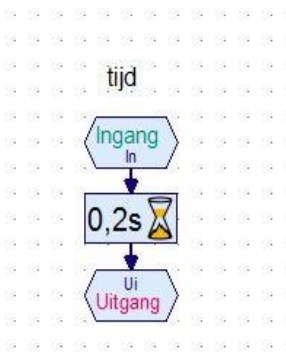
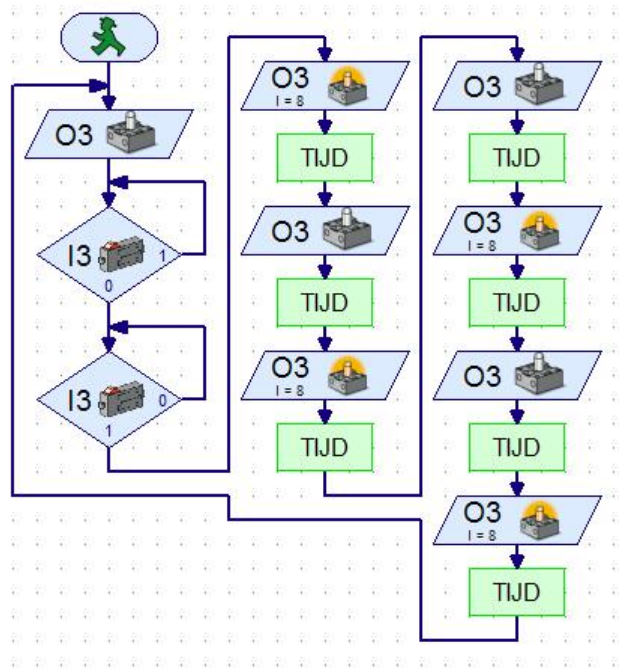
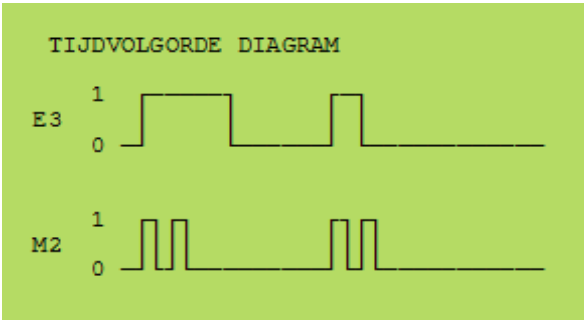
Oldtask = Task
```

Loop

Subroutines
(zie bijlage:3)

VOORBEELD-36: PUNTLASSER AAN/UIT (2x).

Als drukknop (E3) ingedrukt wordt, gaat de puntlasser-lamp (M2) 4x aan en uit.



```

BASC0M-AVR
Compiler gegevens en instellingen.
(zie bijlage:2)
Do

Select Case Task

  Locate 2 , 1
  Lcd "task = " ; Task

  Case 0
    If E3 = 0 Then Task = 1

  Case 1:
    If E3 = 1 Then
      Var = 0
      Task = 2
      M2 = 1
    End If

  Case 2:
    Var = Var + 1
    If Var > 10 Then
      Var = 0
      Task = 3
      M2 = 0
    End If

  Case 3:
    Var = Var + 1
    If Var > 10 Then
      Var = 0
      Task = 4
      M2 = 1
    End If

```

```
Case 4:  
  Var = Var + 1  
  If Var > 10 Then  
    Var = 0  
    Task = 5  
    M2 = 0  
  End If
```

```
Case 5:  
  Var = Var + 1  
  If Var > 10 Then  
    Var = 0  
    Task = 6  
    M2 = 1  
  End If
```

```
Case 6:  
  Var = Var + 1  
  If Var > 10 Then  
    Var = 0  
    Task = 7  
    M2 = 0  
  End If
```

```
Case 7:  
  Var = Var + 1  
  If Var > 10 Then  
    Var = 0  
    Task = 8  
    M2 = 1  
  End If
```

```
Case 8:  
  Var = Var + 1  
  If Var > 10 Then  
    Var = 0  
    Task = 0  
    M2 = 0  
  End If
```

```
End Select
```

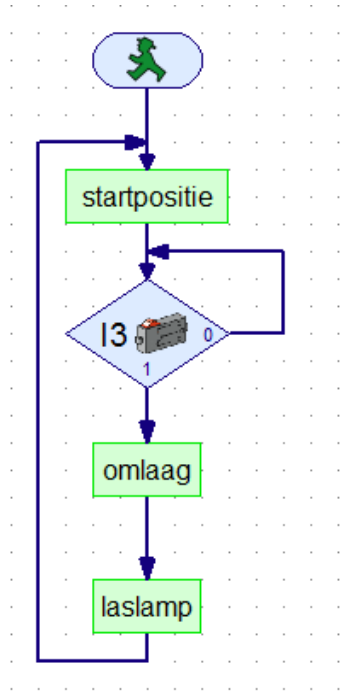
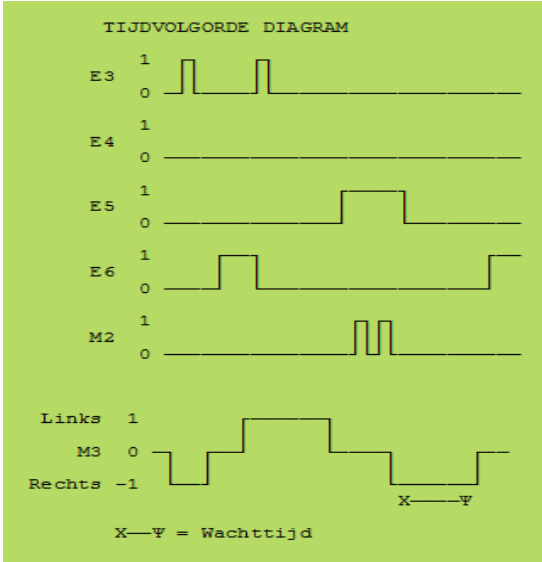
Loop

Subroutines
(zie bijlage:3)

VOORBEELD-37: PUNTLASROBOT met PUNTLASSER (1).

Als drukknop (E3) ingedrukt wordt, dan gaat de puntlasrobot (M3) omhoog. De motor stopt zodra de eind-drukknop (E6) is bereikt. **(Dit gebeurt alléén indien de puntlasrobot zich NIET in de startpositie bevindt).** De puntlasrobot bevindt zich nu in de startpositie.

Als drukknop (E3) nogmaals ingedrukt wordt, dan gaat de puntlasrobot omlaag. De motor stopt zodra de eind-drukknop (E5) is bereikt. Hierna volgt een korte pauze, waarbij de puntlasser (M2) 2x aan en uit gaat. Daarna gaat de puntlasrobot weer omhoog. De motor stopt zodra de eind-drukknop (E6) is bereikt. De motor en de lamp gaan uit zodra de rechter drukknop (E4) wordt ingedrukt. Herstart is pas weer mogelijk nadat beide drukknoppen zijn losgelaten.



```

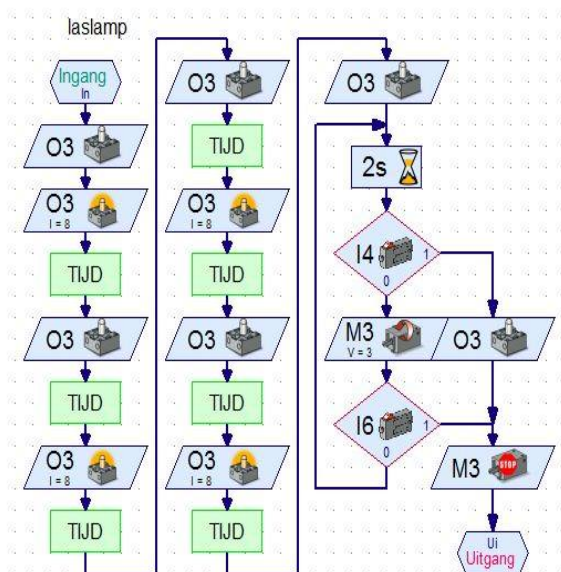
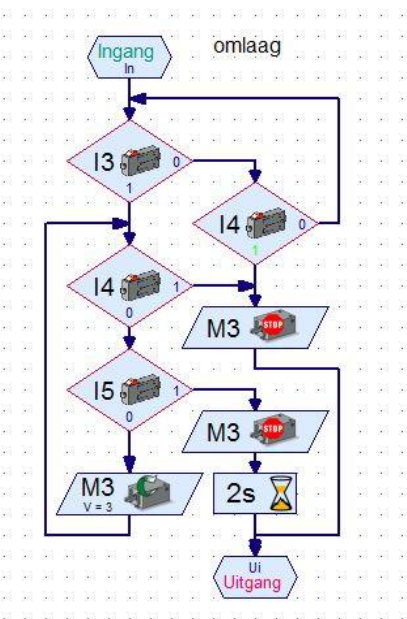
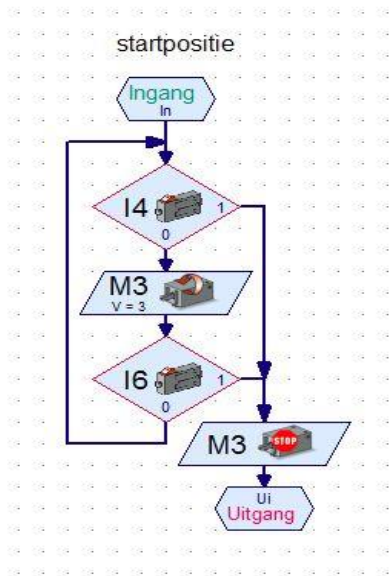
BASCOS-AVR
Compiler gegevens en instellingen.
(zie bijlage: 2)
Do
  If E4 = 1 Then
    Task = 0
    Call Motor3_stop
    Stop Timer0
    M4 = 0
  Else
    Task = Oldtask
  End If

  Select Case Task

  Locate 2 , 1
  Lcd "task = " ; Task

  Case 0
    If E3 = 1 Then Task = 1

  Case 1
    If E6 = 1 Then
      Task = 2
      Call Motor3_stop
      M4 = 0
    Else
      Call Motor3_up
    End If
  
```

Case 2:
If E3 = 0 Then Task = 3

Case 3:
If E3 = 1 Then Task = 4

Case 4:
If E5 = 1 Then
Task = 5
Var = 0
Call Motor3_stop
Else
Call Motor3_down
End If

Case 5:
Var = Var + 1
If Var > 200 Then
Var = 0
M2 = 1
Task = 6
End If

Case 6:
Var = Var + 1
If Var > 10 Then
Var = 0
M2 = 0
Task = 7
End If

Case 7:
Var = Var + 1
If Var > 10 Then
Var = 0
M2 = 1
Task = 8
End If

Case 8:
Var = Var + 1
If Var > 10 Then
Var = 0
M2 = 0
Task = 9
End If

Case 9:
Var = Var + 1
If Var > 10 Then
Var = 0
M2 = 1
Task = 10
End If

Case 10:
Var = Var + 1
If Var > 10 Then
Var = 0
M2 = 0
Task = 11
End If

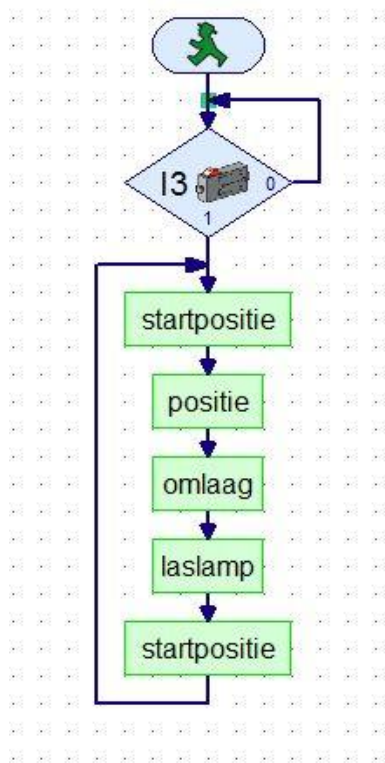
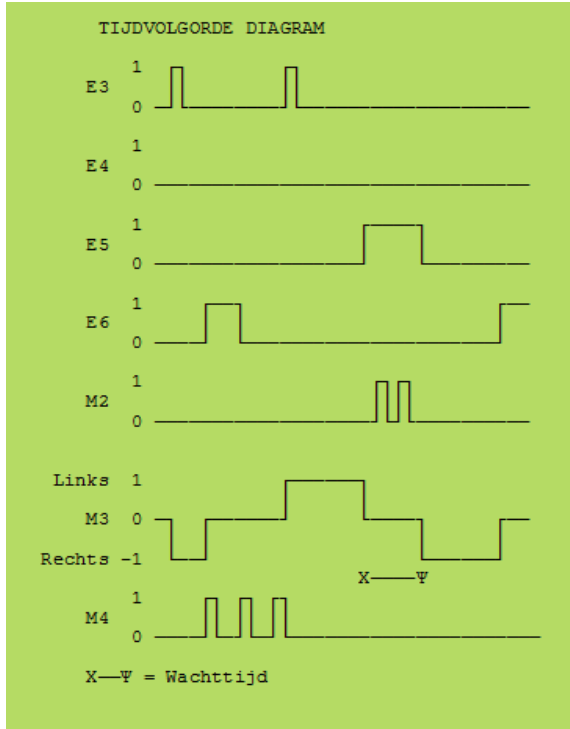
```
Case 11:  
  Var = Var + 1  
  If Var > 200 Then Task = 1  
  
End Select  
  
Oldtask = Task
```

Loop

Subroutines
(zie bijlage:3)

VOORBEELD-38: PUNTLASROBOT met PUNTLASSEER (2).

Zodra de linker drukknop (E3) ingedrukt wordt, dan gaat de puntlasrobot (M3) omhoog. De motor stopt zodra de eind-drukknop (E6) is bereikt. (Dit gebeurt alléén indien de puntlasrobot zich NIET in de startpositie bevindt). De puntlasrobot bevindt zich nu in de startpositie. De initialisatielamp gaat knipperen. Zodra de linker drukknop (E3) nogmaals ingedrukt wordt, dan gaat de puntlasrobot omlaag. De motor stopt zodra de einddrukknop (E5) is bereikt. Hierna volgt een korte pauze, waarbij de puntlasser (M2) 2x aan en uit gaat. Daarna gaat de puntlasrobot weer omhoog. De motor stopt zodra de eind-drukknop (E6) is bereikt. Beveiliging: De motor en lampen gaan uit zodra de rechter drukknop (E4) wordt ingedrukt. Herstart is pas weer mogelijk als beide drukknoppen zijn losgelaten. Deze schakeling is een uitgebreide versie van cursus37.



```

BASCOS-AVR
Compiler gegevens en instellingen.
(zie bijlage: 2)
Do
  If E4 = 1 Then
    Call Motor1_noodstop
    Call Motor3_noodstop
    Task = 0
    Stop Timer0
    M2 = 0
    M4 = 0
  Else
    Task = Oldtask
  End If

  Select Case Task

  Locate 2 , 1
  Lcd "task = " ; Task

  Case 0:
    If E3 = 0 Then Task = 1
    
```

```

Case 1:
  Stop Timer0
  M2 = 0
  M4 = 0
  Aantal_keren = 0
  If E3 = 1 Then Task = 2

Case 2:
  Call Motor3_up
  If E6 = 1 Then
    Call Motor3_stop
    Task = 3
  End If

Case 3:
  If E3 = 0 Then Task = 4

Case 4:
  If E3 = 1 Then Task = 5

Case 5:
  Call Motor1_rechts
  If E2_count > 10 Then
    Call Motor1_stop

    Locate 2 , 13
    Lcd ; E2_count

    Task = 6
  End If

Case 6:
  Call Motor1_links
  If E1 = 1 Then
    Call Motor1_stop
    Var = 0
    Task = 7
  End If

Case 7:
  Var = Var + 1
  If Var > 200 Then
    E2_count = 0
    Task = 8
  End If

Case 8:
  Call Motor1_rechts
  If E2_count > 35 Then
    Call Motor1_stop

    Locate 2 , 13
    Lcd ; E2_count

    Var = 0
    Task = 9
  End If

Case 9:
  Var = Var + 1
  If Var > 200 Then
    Start Timer0
    E2_count = 0
    Var = 0
    Task = 10
  End If

```

```

Case 10:
  Var = Var + 1
  If Var > 200 Then Task = 11

Case 11:
  Call Motor1_stop
  Call Motor3_down
  If E5 = 1 Then
    Call Motor3_stop
    Var = 0
    Task = 12
  End If

Case 12:
  Var = Var + 1
  If Var > 200 Then
    M2 = 1
    Var = 0
    Task = 13
  End If

Case 13:
  Var = Var + 1
  Select Case Var
    Case 0 To 10
      M2 = 0
    Case 11 To 20
      M2 = 1
    Case 21 To 30
      M2 = 0
    Case 31 To 40
      M2 = 1
    Case 41 To 50
      M2 = 0
    Case 51 To 60
      M2 = 1
    Case 61 To 70
      M2 = 0
      Task = 14
      Var = 0
  End Select

Case 14:
  Var = Var + 1
  If Var > 200 Then
    Var = 0
    Task = 15
  End If

Case 15
  Call Motor3_up
  If E6 = 1 Then
    Call Motor3_stop
    Task = 0
  End If
End Select

Oldtask = Task

```

Loop

Subroutines
(zie bijlage: 3)

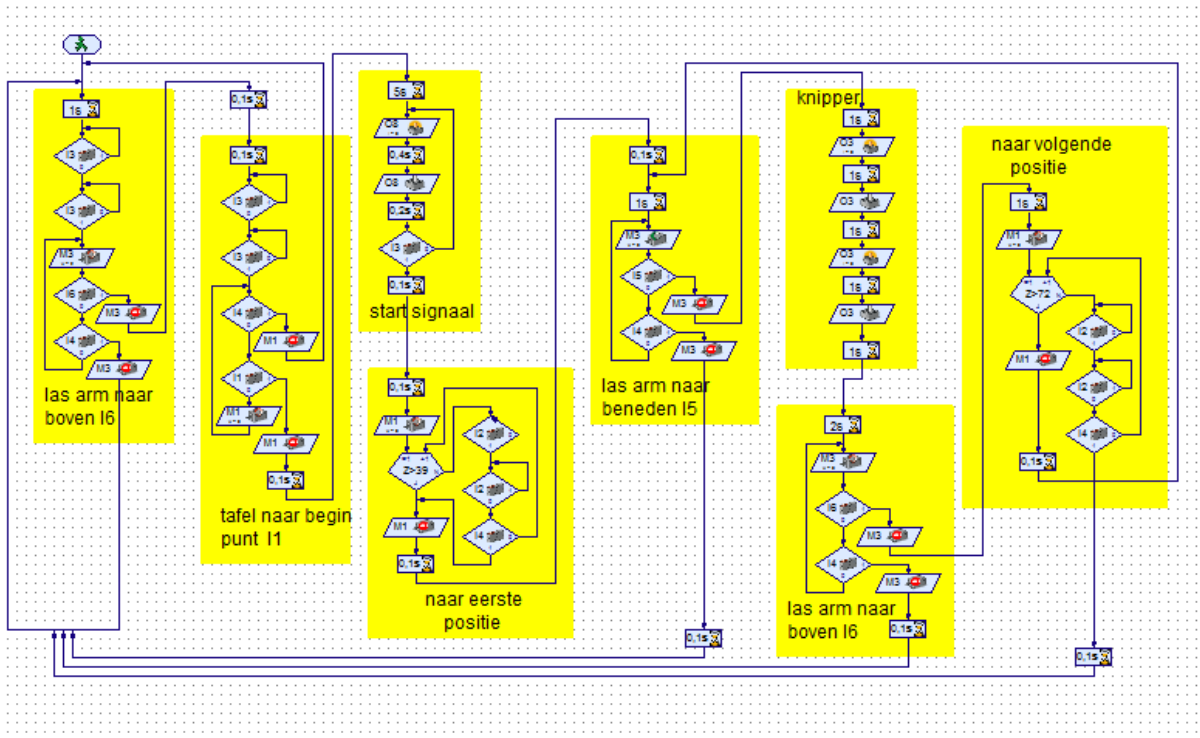
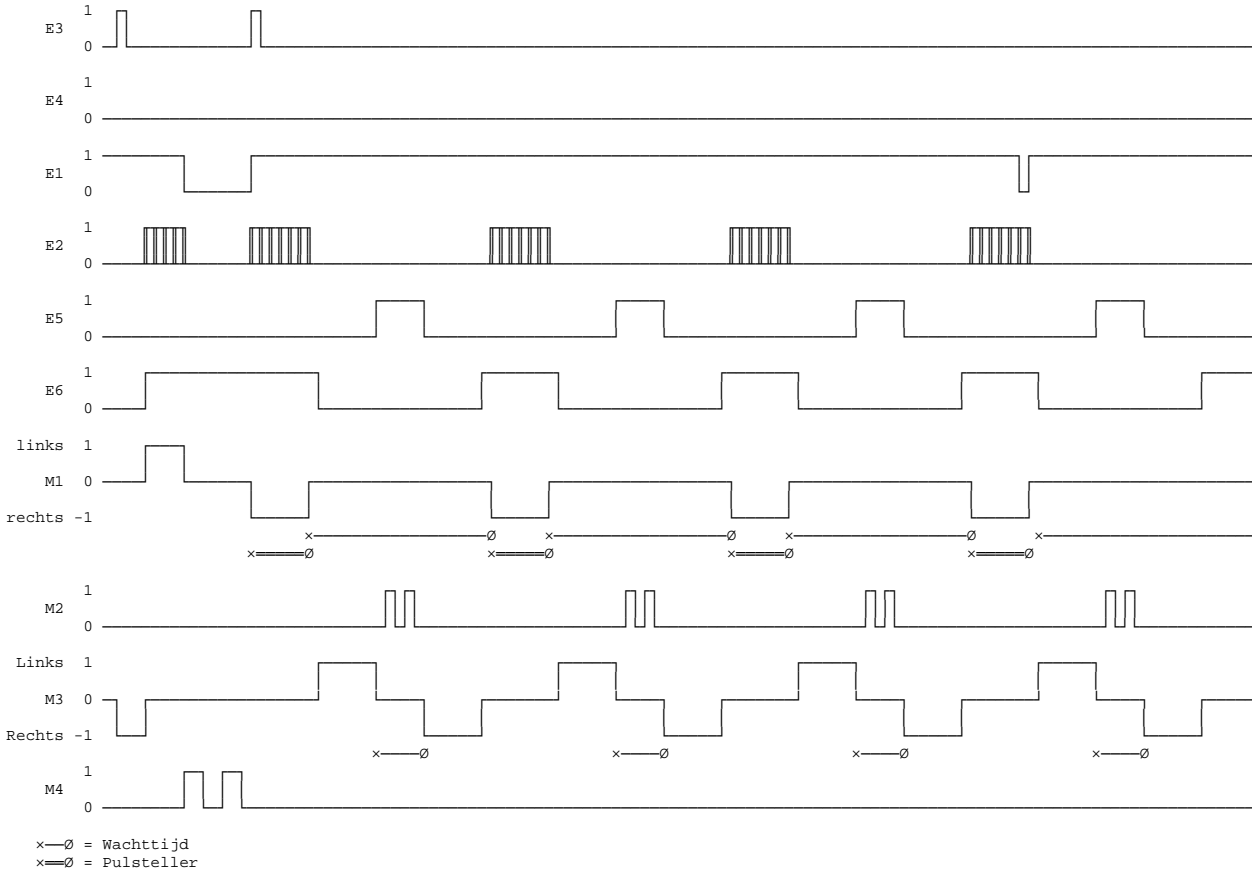
VOORBEELD-39: PUNTLASROBOT, COMPLEET MODEL (3)

Dit programma is nagenoeg gelijk aan die van cursus38.

Nadat de initialisatielamp gaat knipperen, dient men binnen een bepaalde tijd de linker drukknop in te drukken, om het proces te starten.

Indien men dit niet doet, dan wordt het programma beëindigd.

TIJDVOLGORDE DIAGRAM TEN BEHOEVE VAN VOORBEELD-39:



Compiler gegevens en instellingen.
(zie bijlage: 2)

Do

'Noodstop

```
If E4 = 1 Then
  Task = 0
  Call Motor1_noodstop
  Call Motor3_noodstop
  Stop Timer0
  M2 = 0
  M4 = 0
Else
  Task = Oldtask
End If
```

Select Case Task

```
  Locate 2 , 1
  Lcd "task = " ; Task
```

'Initialisatie puntlasser

```
Case 0:
  If E3 = 0 Then Task = 1
```

```
Case 1:
  Stop Timer0
  M2 = 0
  M4 = 0
  Aantal_keren = 0
  If E3 = 1 Then Task = 2
```

```
Case 2:
  Call Motor3_up
  If E6 = 1 Then
    Call Motor3_stop
    Task = 3
  End If
```

'Initialisatie draaitafel

```
Case 3:
  If E3 = 0 Then Task = 4
```

```
Case 4:
  If E3 = 1 Then Task = 5
```

```
Case 5:
  Call Motor1_rechts
  If E2_count > 10 Then
    Call Motor1_stop
```

```
  Locate 2 , 13
  Lcd ; E2_count
```

```
  Task = 6
  End If
```

Case 6:

```
  Call Motor1_links
  If E1 = 1 Then
    Call Motor1_stop
    Var = 0
    Task = 7
  End If
```

Case 7:

```
  Var = Var + 1
  If Var > 200 Then
    E2_count = 0
    Task = 8
  End If
```

Case 8:

```
  Call Motor1_rechts
  If E2_count > 35 Then
    Call Motor1_stop
```

```
  Locate 2 , 13
  Lcd ; E2_count
```

```
  Var = 0
  Task = 9
  End If
```

Case 9:

```
  Var = Var + 1
  If Var > 200 Then
    E2_count = 0
    Task = 11
```

'Werk-in-proces lamp aan

```
  Start Timer0
  End If
```

```
*****
Case 10 t/m case 16 wordt 4 maal
uitgevoerd
```

'Innemen laspositie draaitafel

Case 10:

```
  Call Motor1_rechts
  If E2_count => Puls Then
    Call Motor1_stop
```

```
  Locate 2 , 13
  Lcd ; E2_count
```

```
  Var = 0
  Task = 11
  End If
```

Case 11:

```
  Var = Var + 1
  If Var > 200 Then Task = 12
```

```

'Puntlasser down
*****
Case 12:
  Call Motor1_stop
  Call Motor3_stop
  Var = 0
  Task = 13
End If

Case 13:
  Var = Var + 1
  If Var > 200 Then
    M2 = 1
    Var = 0
    Task = 14
  End If

'Lassen
*****
Case 14:
  Var = Var + 1
  Select Case Var
    Case 0 To 10
      M2 = 0
    Case 11 To 20
      M2 = 1
    Case 21 To 30
      M2 = 0
    Case 31 To 40
      M2 = 1
    Case 41 To 50
      M2 = 0
    Case 51 To 60
      M2 = 1
    Case 61 To 70
      M2 = 0
      Task = 15
  End Select

Case 15:
  Var = Var + 1
  If Var > 200 Then
    Var = 0
    Task = 16
  End If

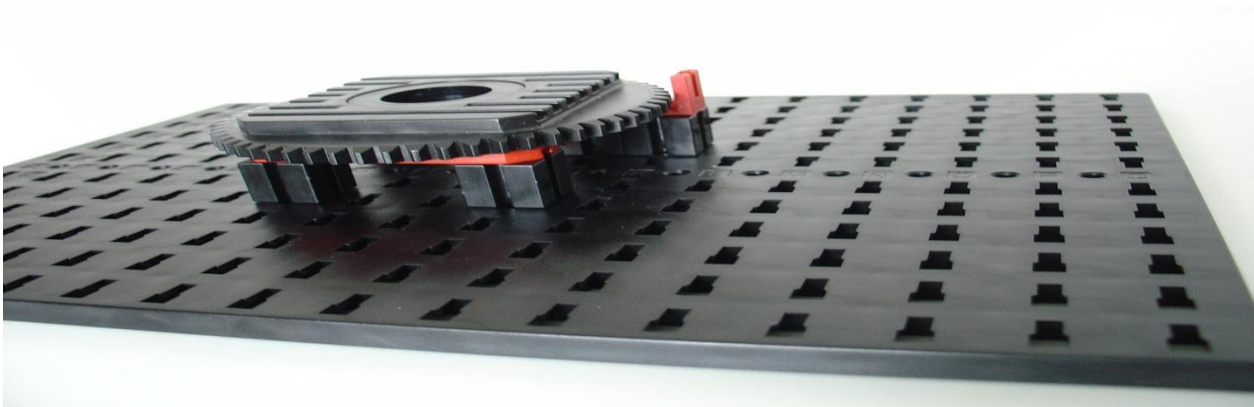
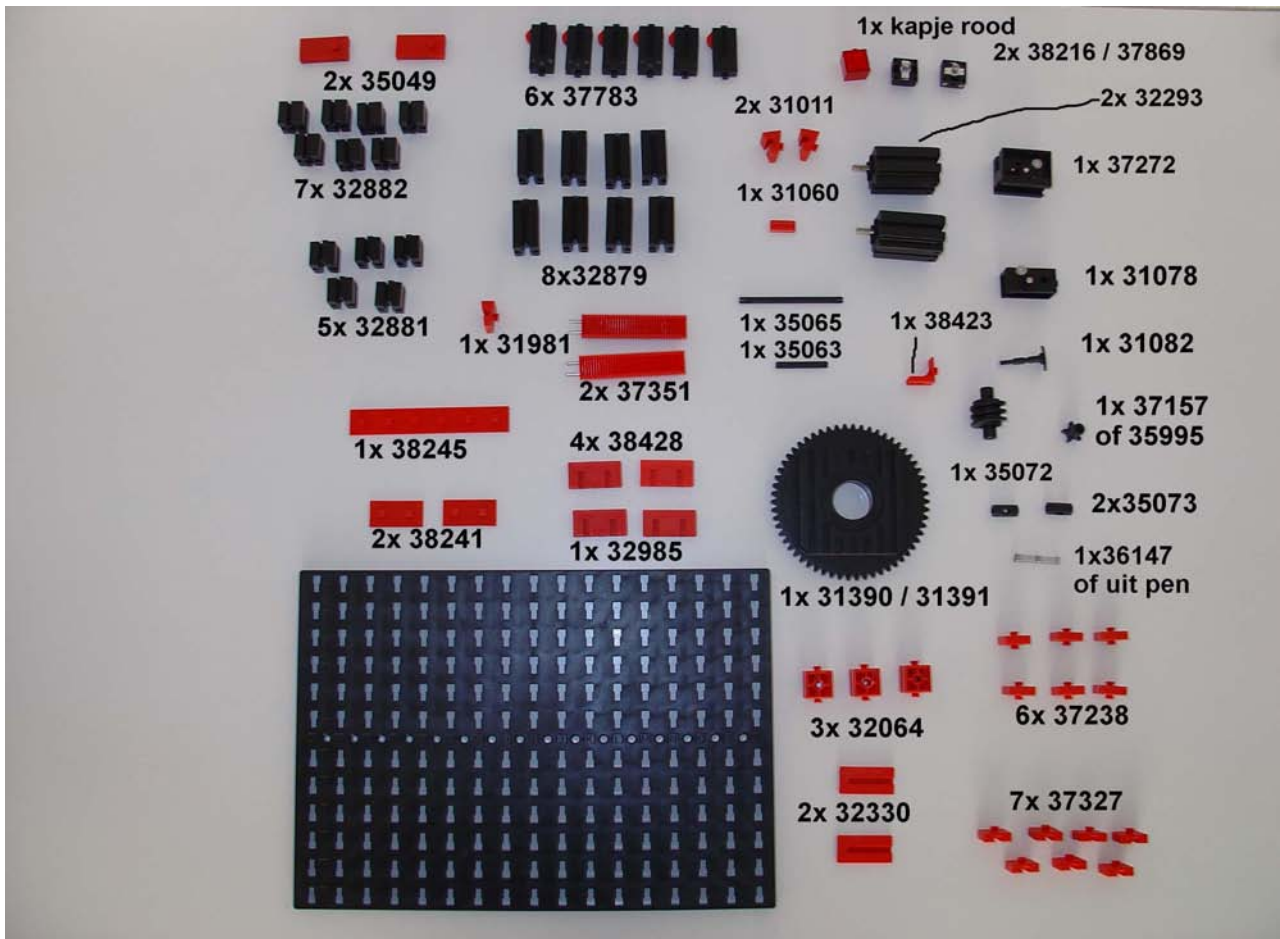
'Puntlasser up
*****
Case 16
  Call Motor3_up
  If E6 = 1 Then
    Call Motor3_stop
    Task = 10
    Aantal = Aantal + 1
    Select Case Aantal
      Case 1 : Puls = 56
      Case 2 : Puls = 114
      Case 3 : Puls = 172
      Case 4 : Puls = 230
      Case 5 : Task = 0
    End Select
  End If
End Select

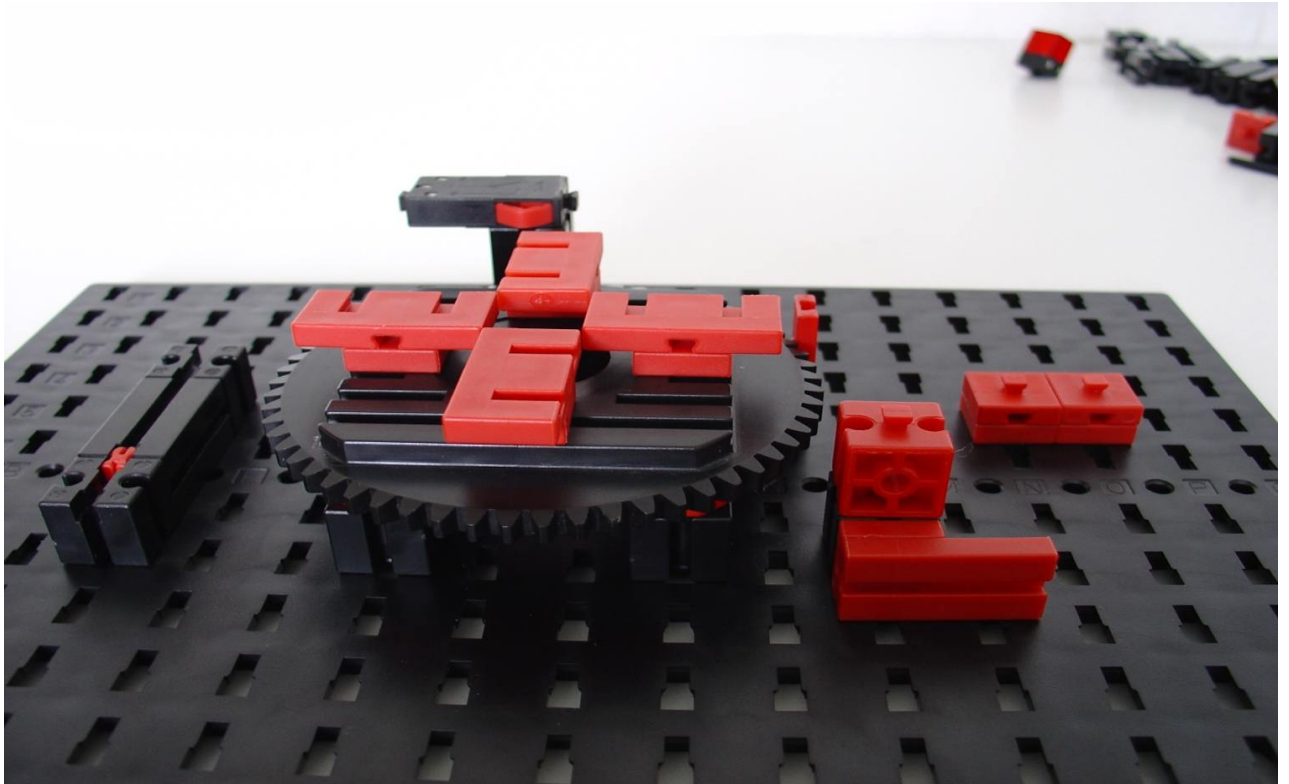
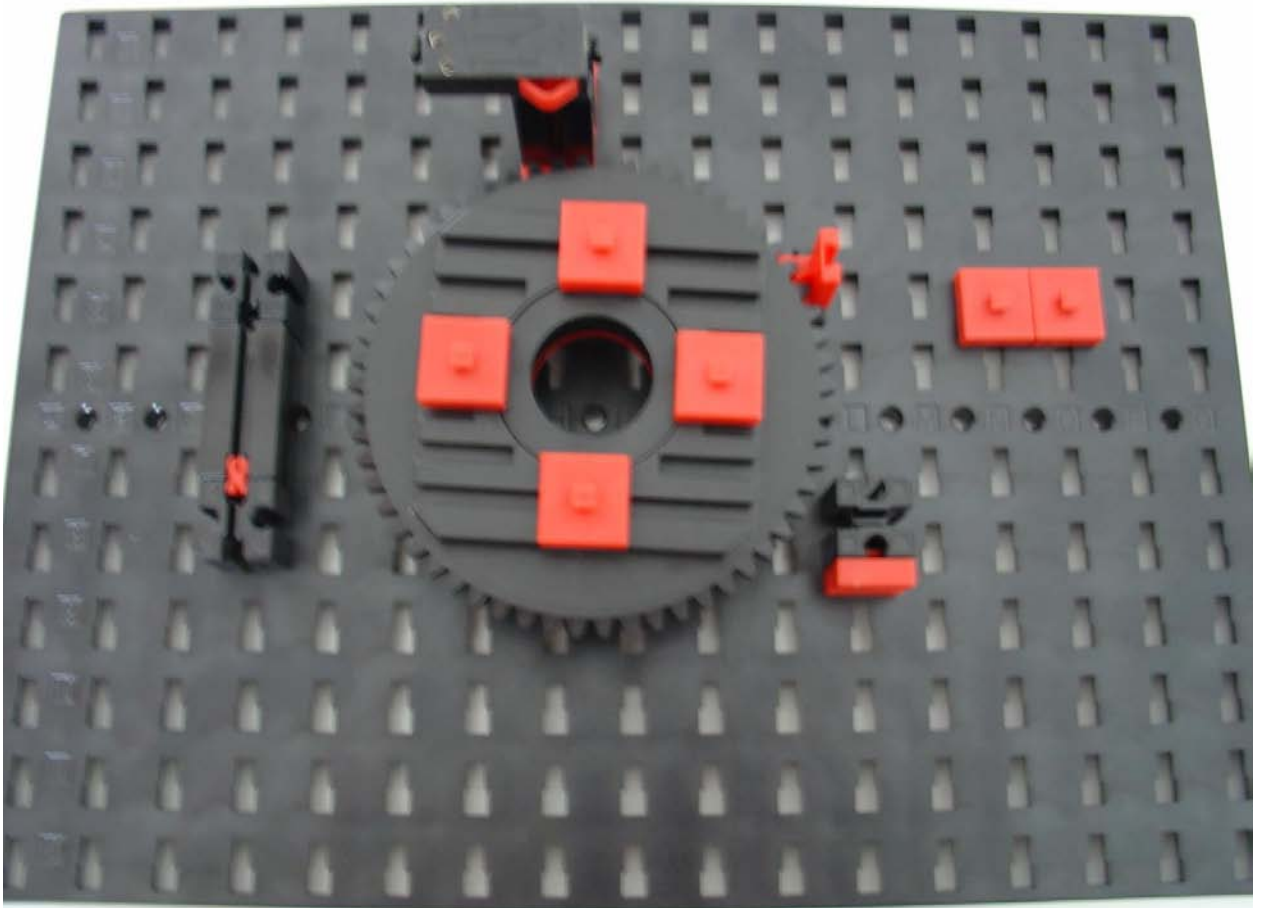
Oldtask = Task
Loop

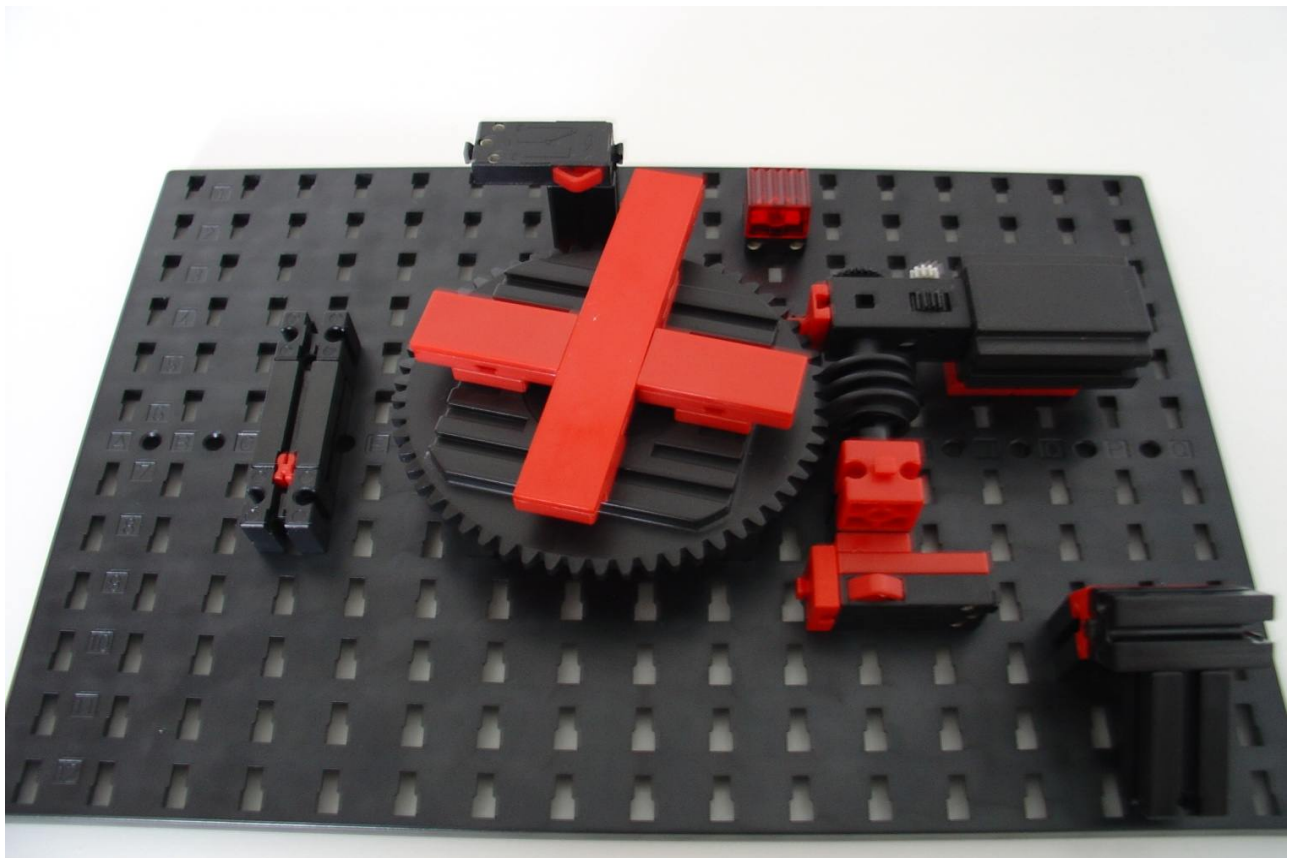
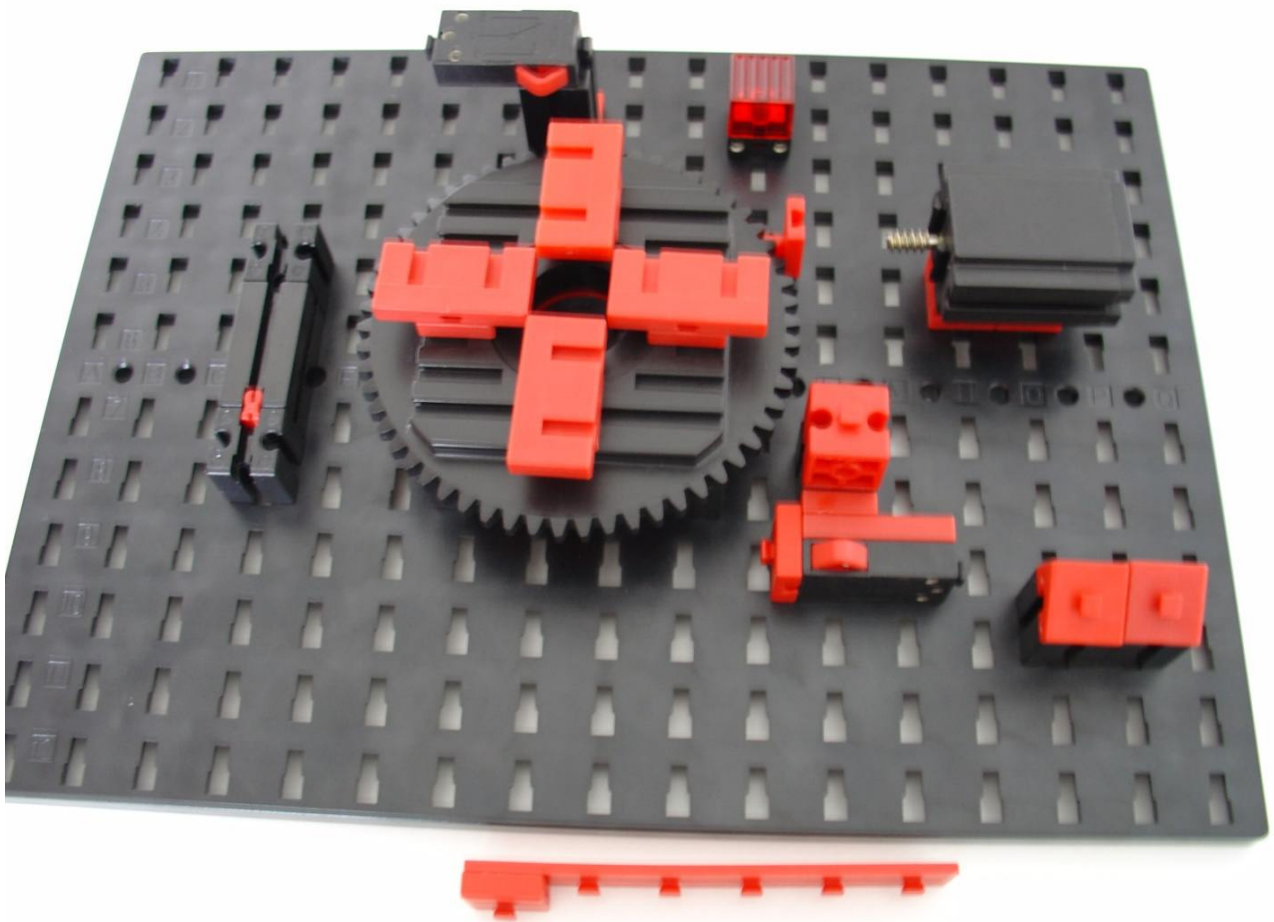
```

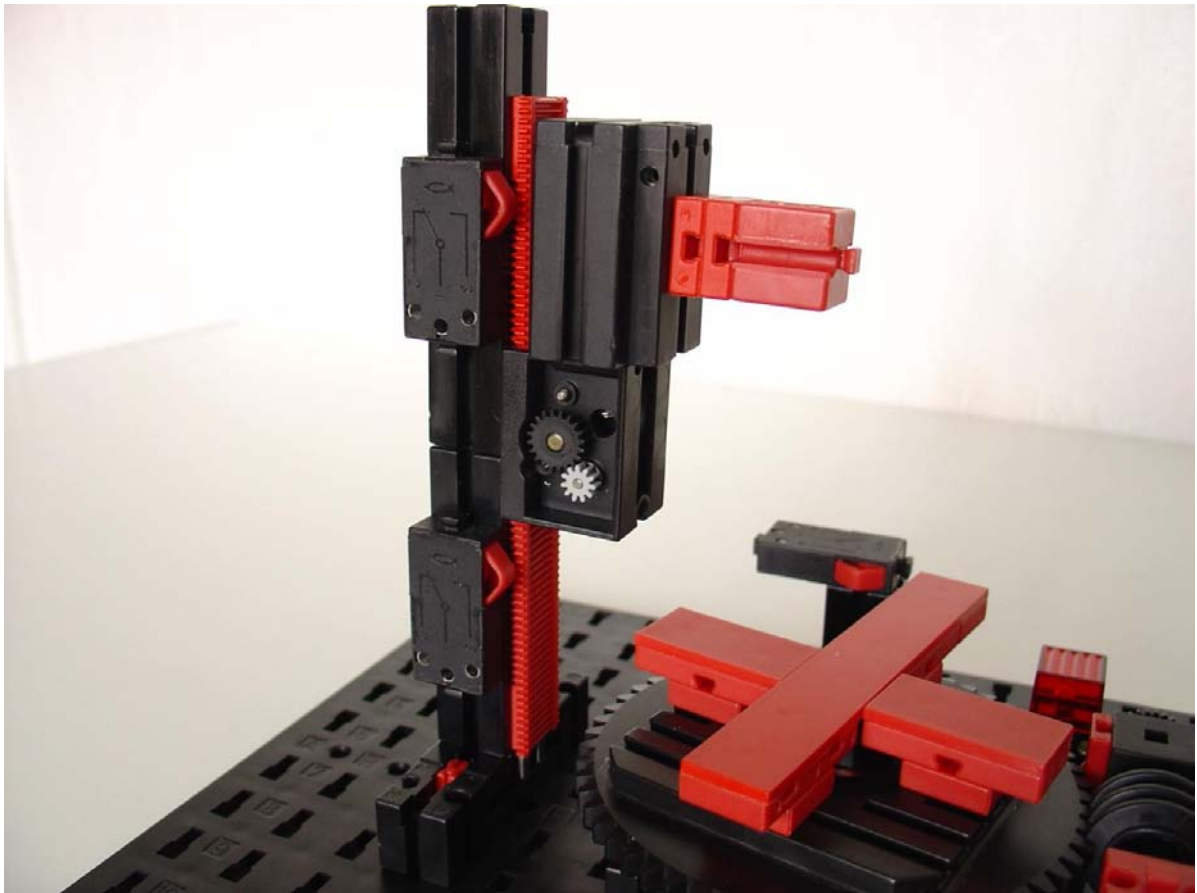
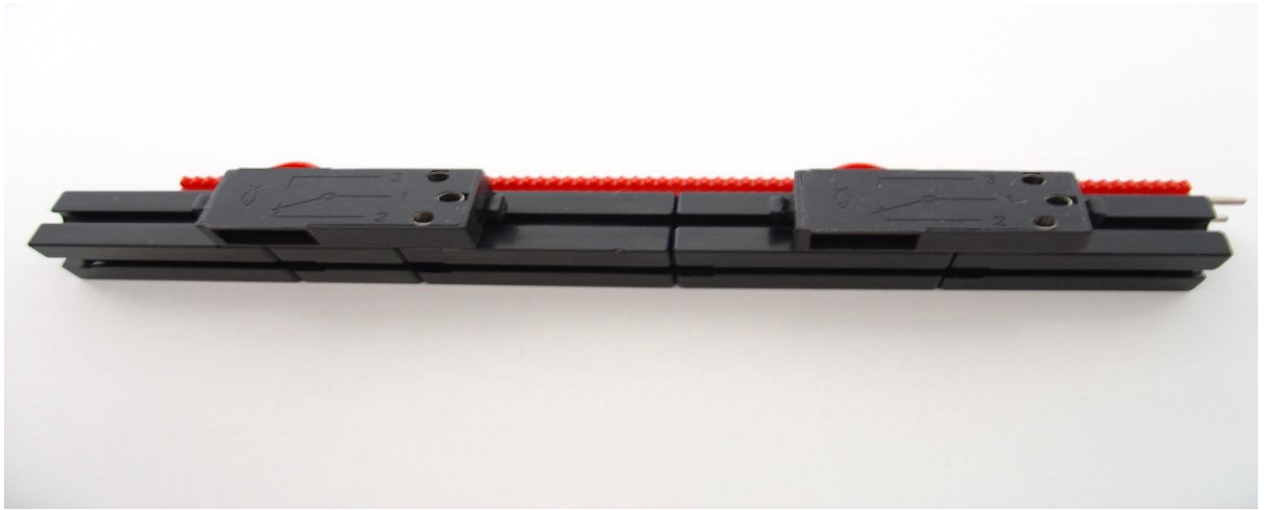
Subroutines
(zie bijlage: 3)

Bijlage 1: Bouwbeschrijving van de lasrobot

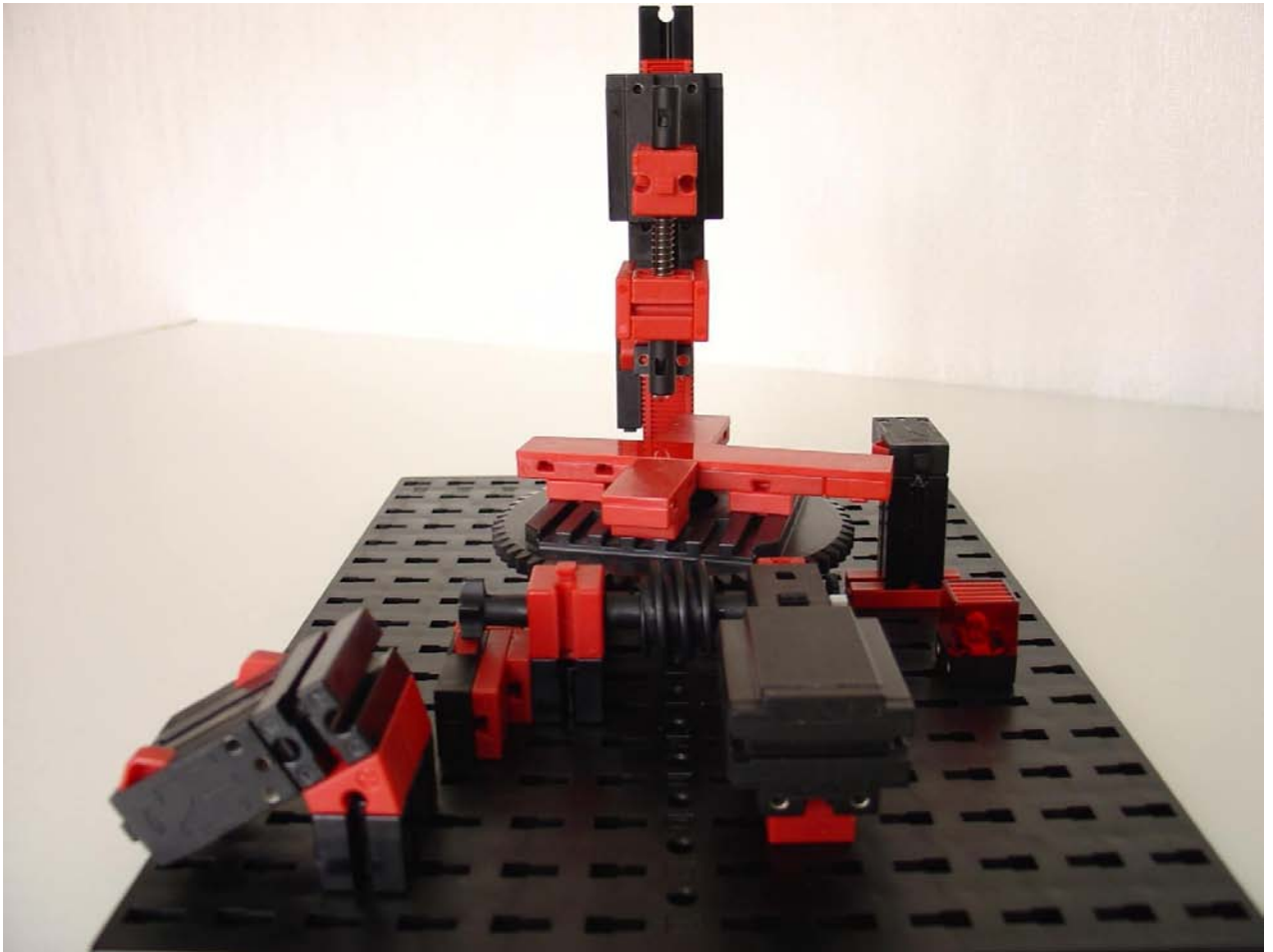












Bijlage 2: Bascom-AVR: Compiler gegevens en instellingen

'1. Titel blok

'Auteur Rien van Harmelen
'Datum juni 2011
'File naam Basisprogramma

'2. Programma beschrijving.

'Het basisprogramma is het vaste deel
'van de Bascom-AVR programma's 1 t/m 39
'beschreven in het fischertechnik lustrumboek 2011

'3. Compiler gegevens

\$regfile = "m8def.dat"
\$crystal = 16000000
\$baud = 9600

'4. Declaraties

Declare Sub Motor1dir(byval M1dir As Byte)
Declare Sub Motor3dir(byval M3dir As Byte)
Declare Sub Motor1speed(byval M1speed As Integer)
Declare Sub Motor3speed(byval M3speed As Integer)

Declare Sub Motor1_links
Declare Sub Motor1_rechts
Declare Sub Motor1_stop
Declare Sub Motor1_noodstop

Declare Sub Motor3_down
Declare Sub Motor3_up
Declare Sub Motor3_stop
Declare Sub Motor3_noodstop

Declare Sub Keypressed
Config Debounce = 20

'5 Timer0 instellen tbv initialisatielamp

Config Timer0 = Timer , Prescale = 1024
On Timer0 Ontimer0
Enable Interrupts
Enable Timer0
Start Timer0
Dim Wachtijd As Byte
Wachtijd = 30

'6 Congfigureren I2C-LCD poorten

```
$lib "Lcd_i2c.lib"  
Config I2cdelay = 1  
Const Pcf8574_lcd = &H40  
Config Scl = Portd.0  
Config Sda = Portd.1  
Dim _lcd_e As Byte  
_lcd_e = 192  
Config Lcd = 16 * 2
```

```
Cls  
Cursor Off
```

```
Locate 1 , 1  
Lcd "Basisprogramma"
```

'7 Configureren Motorpoorten (M1 en M3)

'Motor M1

'-----

```
Config Portb.1 = Output  
Config Portb.0 = Output  
Config Portb.5 = Output
```

'Motor M3

'-----

```
Config Portb.2 = Output  
Config Portd.6 = Output  
Config Portd.7 = Output
```

```
Const Fwd = 1  
Const Rwd = 2  
Const Free = 4  
Const Brake = 3  
Dim Speed As Integer  
Speed = 1023
```

'8 Timer1 instellen tbv PWM

```
Config Timer1 = Pwm , Pwm = 10 , Prescale = 1 , Compare A Pwm = Clear Up ,  
Compare B Pwm = Clear Up
```

'9. Configureren Schakelaarpoorten (E1,E3,E4,E5 en E6)

Config Pinc.0 = Input

Portc.0 = 0

E6 Alias Pinc.0

Config Pinc.2 = Input

Portc.2 = 0

E5 Alias Pinc.2

Config Pinc.3 = Input

Portc.3 = 0

E4 Alias Pinc.3

Config Pinc.4 = Input

Portc.4 = 0

E3 Alias Pinc.4

Config Pinc.5 = Input

Portc.5 = 0

E1 Alias Pinc.5

'10 Configureren Endcoderpoort(schakelaar E2)

Config Pind.3 = Input

Portd.3 = 0

On Int1 Int1_isr

Enable Int1

Enable Interrupts

Config Int1 = Falling

Sreg.7 = 1

Dim E2_count As Integer

E2_count = 0

'11 Configureren Initialisatielamppoort (M4) (work-in-proces lamp)

Config Portb.4 = Output

M4 Alias Portb.4

'12 Configureren Puntlasserlamppoort (M2)

Config Portb.3 = Output

M2 Alias Portb.3

'13 Initialieren variabelen

'Select case

'-----

Dim Task As Byte

Task = 0

Dim Oldtask As Byte

Oldtask = 0

'Softwarematige "wait"

'-----

Dim Var As Integer

Var = 0

'Aantal lasbeurten

'-----

Dim Aantal_keren As Byte

Dim Aantal As Byte

Aantal_keren = 4

Aantal = 0

'Aantal pulsen (schakelaar E2)

'-----

Dim Puls As Integer

Puls = 0

'14. Timer0 uitschakelen (work-in-proces lamp (=M4)uit)

Stop Timer0

'15 Programma start hier

Do

.

.

.

.

Loop

Bijlage 3: Bascom-AVR: Subroutines

'16 Subroutines motoren

'Draairichting motor M1

Sub Motor1dir(byval M1dir As Byte)

 If M1dir = 1 Then

 Portb.0 = 0

 Portb.5 = 1

 Elseif M1dir = 2 Then

 Portb.0 = 1

 Portb.5 = 0

 Elseif M1dir = 3 Then

 Portb.0 = 1

 Portb.5 = 1

 Elseif M1dir = 4 Then

 Portb.0 = 0

 Portb.5 = 0

 End If

End Sub

'Draairichting motor M3

Sub Motor3dir(byval M3dir As Byte)

 If M3dir = 1 Then

 Portd.6 = 0

 Portd.7 = 1

 Elseif M3dir = 2 Then

 Portd.6 = 1

 Portd.7 = 0

 Elseif M3dir = 3 Then

 Portd.6 = 1

 Portd.7 = 1

 Elseif M3dir = 4 Then

 Portd.6 = 0

 Portd.7 = 0

 End If

End Sub

'Snelheid motor M1

Sub Motor1speed(byval M1speed As Integer)

 Pwm1b = M1speed

End Sub

'Snelheid motor M3

'-----

Sub Motor3speed(byval M3speed As Integer)

 Pwm1a = M3speed

End Sub

'Richting motor M1

'-----

Sub Motor1_links

 Call Motor1dir(rwd)

 Call Motor1speed(speed)

End Sub

Sub Motor1_rechts

 Call Motor1dir(fwd)

 Call Motor1speed(speed)

End Sub

'Stop en noodstop motor M1

'-----

Sub Motor1_stop

 Call Motor1dir(brake)

 Call Motor1speed(0)

End Sub

Sub Motor1_noodstop

 Call Motor1dir(brake)

 Call Motor1speed(0)

 'End

End Sub

'Richting motor M3

'-----

Sub Motor3_down

 Call Motor3dir(fwd)

 Call Motor3speed(speed)

End Sub

Sub Motor3_up

 Call Motor3dir(rwd)

 Call Motor3speed(speed)

End Sub

'Stop en noodstop motor M3

Sub Motor3_stop
 Call Motor3dir(brake)
 Call Motor3speed(0)
End Sub

Sub Motor3_noodstop
 Call Motor3dir(brake)
 Call Motor3speed(0)
 'End
End Sub

'17 Subroutine Timer0 tbv M4

Ontimer0:
 Wachttijd = Wachttijd - 1
 If Wachttijd < 1 Then
 Toggle M4
 Wachttijd = 30
 End If
Return

'18 Interrupt subroutine tbv E2_count

Int1_isr:
 Debounce Pind.3 , 0 , Keypressed , Sub
Return

Sub Keypressed
 Incr E2_count
End Sub