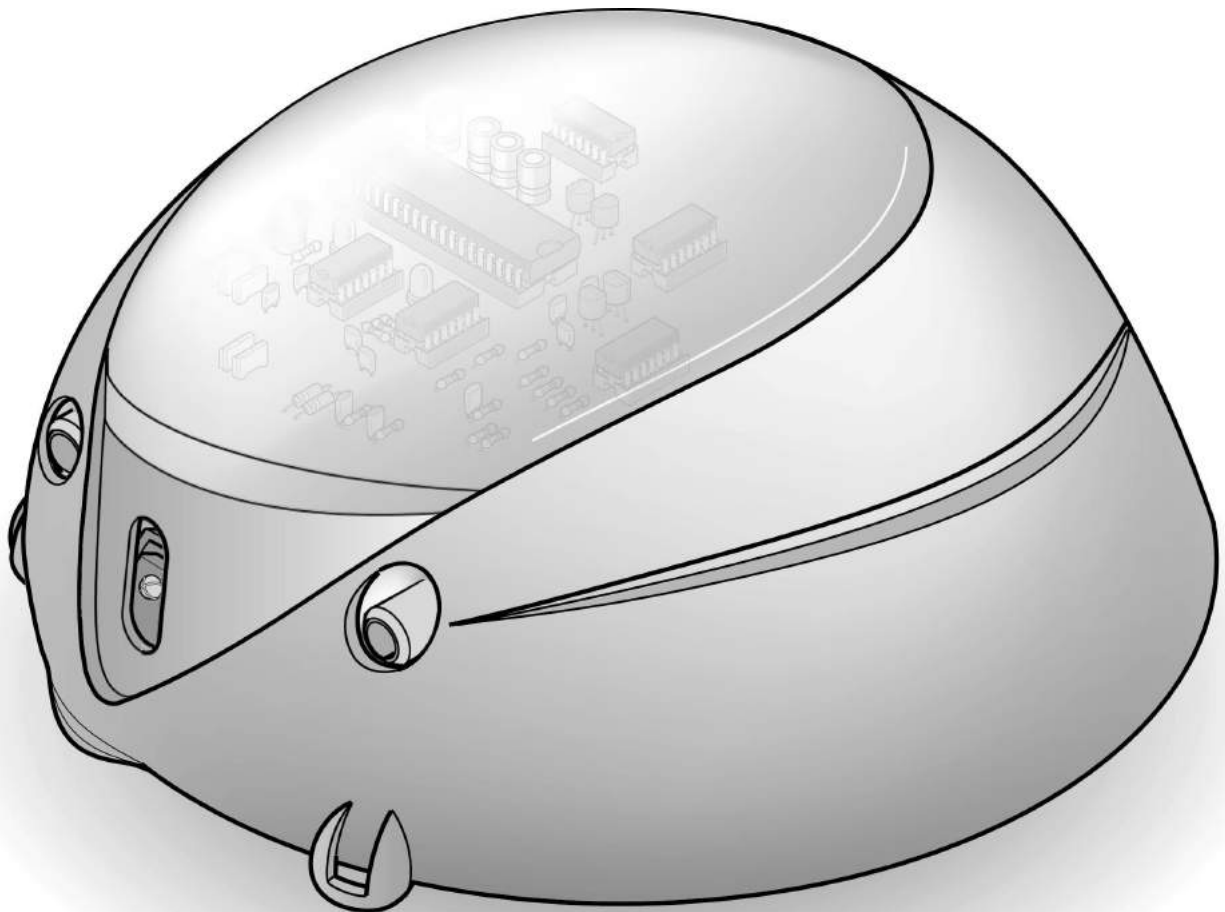

ELEKIT®

MOVIT®

AREXX
AREXX Engineering

AREXX EDUCATIEF ROBOTSISTEEM MR-9762E



Bouwbeschrijving: model MR-9762E

© EK JAPAN CO., LTD. 2001
© Nederlandse handleiding: AREXX Engineering

INHOUD

1. Productbeschrijving P!MOT	3
2. Geschiedenis van de Robot	4
3. Belangrijke informatie	10
3.1 Benodigde gereedschappen	10
3.2 Tips en trucs	10
3.3 Monteren van onderdelen	11
4. Onderdelenlijst P!MOT	12
5. Bouwen van het chassis	14
6. Uitleg van de P!MOT delen	25
6.1. Namen en functies	25
6.2. Beschrijving van de toetsen	23
6.3. Bedrijfstest	26
6.3.3 Foutzoeken	27
6.4. Definitie van de Modi	28
6.5 Informatie voor het programmeren	30
7. Programmeren	33
8. Technische details	40
8.B. Wat is een computer?	41
8.C Motoraaandrijving	45
8.D Sensoren	48
8.E Schema	50
8.F Notities	55

MOVIT en ELEKIT zijn geregistreerde handelsmerken van EK Japan Co., Ltd.

(c) Nederlandse vertaling (March 2016): AREXX Engineering (NL).

Niets uit deze handleiding mag op welke wijze en voor welke doeleinde dan ook worden overgenomen zonder schriftelijke toestemming van de Europees importeur:

AREXX Engineering te Zwolle (NL).

De fabrikant en importeur stellen zich niet verantwoordelijk en aanvaarden geen enkele aansprakelijkheid voor de gevolgen van ondeskundige handelingen en/of eventuele fouten bij het bouwen en bij het gebruik van dit product door het niet opvolgen van deze handleiding.

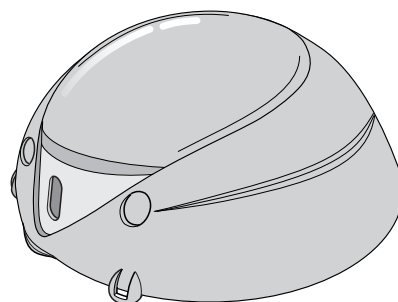
De inhoud van deze handleiding kan zonder kennisgeving vooraf door ons worden gewijzigd.



Fabrikant:
EK Japan Co., Ltd.
FUKUOKA Japan



Europees importeur:
AREXX Engineering
ZWOLLE The Netherlands



Technische ondersteuning:
www.arexx.com

© EK JAPAN CO., LTD. 2000

© Nedelands AREXX Engineering, Zwolle

1. PRODUCTBESCHRIJVING P!MOT

P!MOT is een tweewielige robot, voorzien van een microcomputer "AT90S8515" met een Flash-geheugen. De tandwieloverbrenging in de P!MOT beschikt over een encoder, die de robot in staat stelt relatief nauwkeurig rechtuit te rijden. Bovendien omvat het systeem een infraroodsensor, die de reflecties van een lichtbundel uit de infrarood-LED's aan beide zijden detecteert, en vier optische sensoren, die de bodemoppervlakte naar helderheidsvariaties afzoeken.

Aan de hand van de informatie, die deze sensoren opleveren neemt de P!MOT beslissingen en regelt daarbij in een besturingsprogramma de robotsnelheid op drie niveaus.

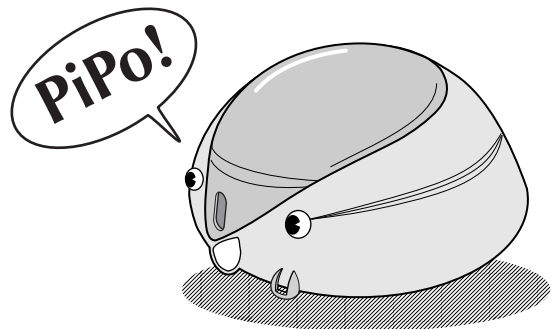
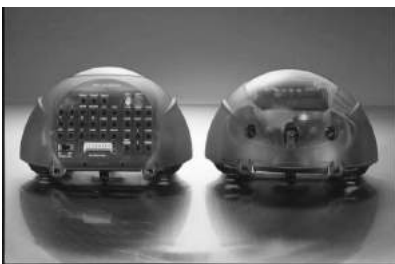
* De sensor analyseert de toestand van de bodemoppervlakte en moet daarom eerst aan de belichting aangepast worden.

Specificaties:

Bedrijfsspanning (motorenmodule):	3 Volt DC (2 Stuks AA - batterijen)
Bedrijfsspanning (elektronicamodule):	9 Volt DC (1 Stuk "rechthoekig" batterijtype)
Vermogensopname (motorenmodule):	maximaal 780 mA
Vermogensopname (elektronicamodule):	in bedrijf ca. 100 mA, in rust ca. 1,8 mA
Batterijenverbruik (motorenmodule):	gemiddeld circa 2 uur (in bedrijf)
Batterijenverbruik (elektronica module):	gemiddeld circa 25 uur (in bedrijf)
Afmetingen (hoogte x lengte x breedte)	85 x 140 x 145 (waardes in mm)
Gewicht (zonder batterijen)	420 gram
Maximaal aantal stappen programmering	60
FOR-NEXT- multiplex loops	maximaal 30 stappen

Wat betekend P!MOT ?

De naam "P!MOT is gebaseerd op Programma en MOTion (Engels: "beweging"). We hebben de robot zo genoemd om iedereen op eenvoudige manier te leren programmeren.



Waarschuwing

- ▶ Zodra de plastic zakjes met onderdelen geopend worden, vervalt het retourrecht.
- ▶ Lees voor het bouwen eerst deze gebruiksaanwijzing aandachtig door.
- ▶ Wees voorzichtig met het gebruik van het gereedschap.
- ▶ Bouw niet in het bijzijn van kleine kinderen. Zij kunnen zich bezeren aan het gereedschap of kleine onderdelen in hun mond stoppen.
- ▶ Gebruik alleen volle batterijen en batterijen van hetzelfde type.
- ▶ Plaats de batterijen op de correcte wijze + en -.
- ▶ Zorg dat de batterijen en de batterijhouders niet nat worden. Als dit toch gebeurt, verwijder dan de batterijen en maak alles goed droog.
- ▶ Verwijder na gebruik de batterijen uit de robot.

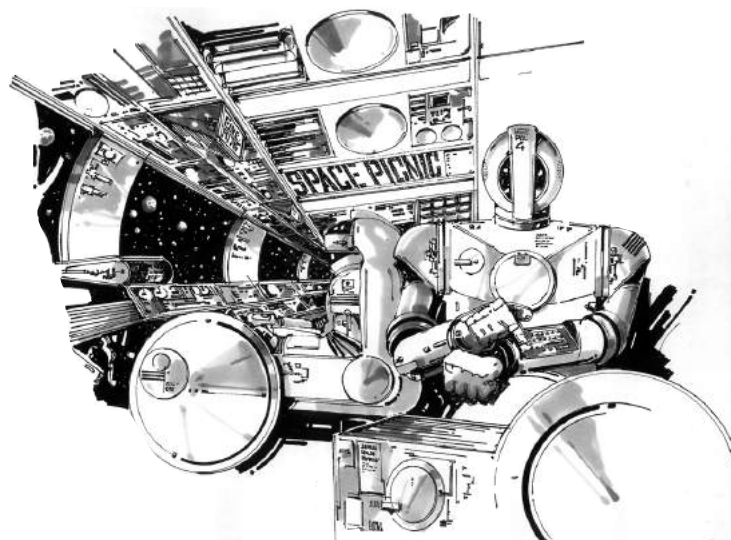
2. ROBOTS, GESCHIEDENIS EN TOEKOMST

2.1 Wat is een robot ?

Robots zijn alom bekend in de hedendaagse wereld. Ze spelen een rol op diverse gebieden, maar we kennen ze vooral uit de industriële sector. Maar op de vraag wat robots eigenlijk zijn, is het antwoord niet zo eenvoudig te geven. Het woordenboek zal de volgende omschrijvingen geven:

- 1) Een door de mens gemaakte pop werkend via allerlei gecompliceerde technieken, een kunstmatige mens.
- 2) Machines die d.m.v. de bovengenoemde technieken zelfstandig kunnen werken zonder tussenkomst van de mens.

Deze beschrijvingen bieden nog geen afdoende antwoord op onze vraag. Een robot als geheel is wetenschappelijk gezien nog niet gedefinieerd, m.u.v. de robots in industriële toepassingen misschien. Wanneer je het begrip robot niet strict neemt en de volgende tekst eens doorleest, zul je interessante dingen te weten komen over robots.

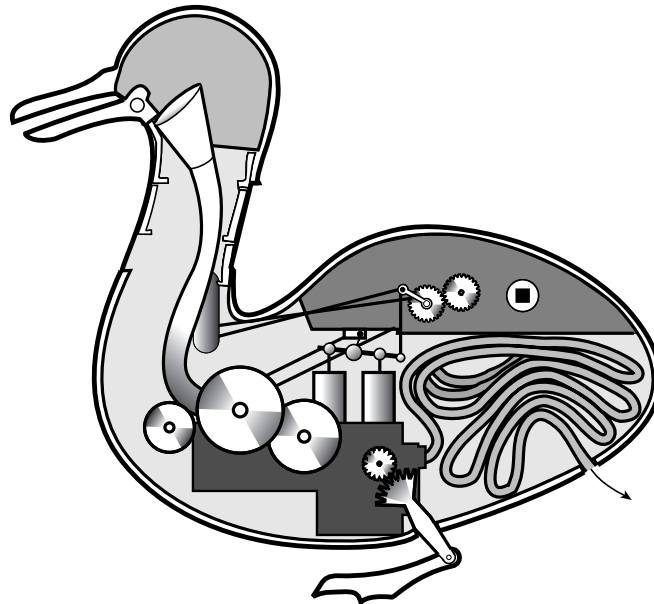


De definitie van een robot:

Nog maar sinds kort wordt de robot gezien als een automatisch systeem zoals je vaak ziet in industriële toepassingen en bij diepzee- en ruimteproeven. Langere tijd geleden had een robot een menselijk gedaante. Ze werden gezien als een automaat die dezelfde handelingen als de mens kon verrichten. Dit soort robots zie je vaak terug in stripverhalen en in Science Fiction series. Ze worden kunstmatige mensen genoemd, omdat ze op onnatuurlijke wijze ontstaan. Omdat ze eruit zien als mensen, worden ze daarnaast vaak "androids" of "humanoids" genoemd. Ook een "cyborg", een combinatie van organische en machinale bestanddelen, wordt in de breedste zin des woords een robot genoemd. Robots als "Automata" kunnen worden beschouwd als robots uit de allereerste beginfase.

Het woord robot is van oorsprong afkomstig uit het toneelstuk "Rur" uit 1920, dat werd geschreven door de Tsjechische toneelschrijver Karel Capek. In dit toneelstuk worden een aantal kunstmatige mensen gefabriceerd, die echte mensen in een werkplaats vervangen. Dit was de eerste keer dat het woord robot werd gebruikt in de betekenis van kunstmatige mens.

Het woord robot is hier afgeleid van het woord "robota" uit de Tsjechische taal. Het betekent zoiets als "gedwongen arbeid". Vanuit dit gezichtspunt is het woord nauw verbonden met de hedentendaagse robot in de industriële toepassingen. De robot in het toneelstuk is echter niet mechanisch, maar een combinatie van organische onderdelen.



2.2 De geschiedenis van de robots

a) Robots in onze literatuur, als speelgoed en als ideologie.

Het was dus in 1920 dat het woord robot is ontstaan. Maar het enthousiasme voor robots was al aanwezig in mythen en literatuur uit de oude tijd voor Christus. De mens had de wens tot het produceren van een gereedschap, dat het werk van de mens kon overnemen. Mensen hebben altijd zelf al veel dingen gecreëerd. Zo is er bijvoorbeeld het gouden meisje uit een Griekse mythe uit de 8e eeuw voor Christus en de bronzen duivel "Talos" uit de 3e eeuw voor Christus. Heron uit de oude Griekse tijd ontwierp o.a. een soort automatische deur, zoals wij die vandaag kennen. In de 18e eeuw was er een Fransman genaamd Beaukerson die een kunstmatige eend creëerde. De eend kon zich baden, eten, huilen en ontlasting produceren.

Er waren veel andere ingenieurs en uitvinders die voor de lol diverse soorten poppen en gereedschappen ontwierpen die konden schrijven, tekenen, etc. Deze poppen en gereedschappen waren succesvol, omdat ze het doel bereikten waarvoor ze werden gemaakt, namelijk het dienen als decoratie en het amuseren van mensen. Ze waren echter niet geschikt voor industriële toepassingen. Het vervaardigen van automatische poppen en gereedschappen was ook niet direct de hoofdactiviteit van deze mensen. Beaukerson bijvoorbeeld fabriceerde weefgetouwen om stoffen te weven. Het is echter niet bekend wat de invloed is geweest van de technieken die hij gebruikte bij het maken van de automatische poppen op het maken van de weefgetouwen.

Uit de 19e eeuw zijn er werken over robots bekend, die het begin vormen van de orthodoxe robotliteratuur. Zo zien wij bijvoorbeeld een quasi-wetenschappelijke manier van schrijven en afbeeldingen van kunstmatige wezens die de strijd met de mens aangaan.

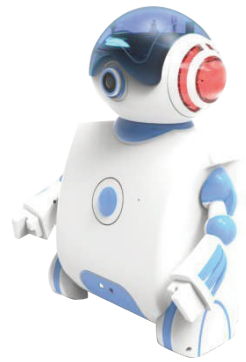
b) De ontwikkeling van wetenschap, technologie en robots

Door de vooruitgang van wetenschap en techniek in de 20e eeuw werden de automatische poppen ontwikkeld tot meer geavanceerde mechanismen. Alhoewel ze in kwaliteit nog duidelijk onderdoen voor de hedendaagse robot, waren de robots van toen de eerste opzet tot een combinatie van mechanische en elektrische technieken (niet vergelijkbaar met de mechatronics van nu). Op de wereldtentoonstelling die in 1927 in New York werd gehouden, was er een Amerikaans bedrijf genaamd Westinghouse dat de robot "Willy" showde. Deze robot kon lopen, praten, knipogen en kleuren identificeren. De robot was gemaakt om menselijke handelingen over te nemen, maar was beperkt in zijn beweging en het ontbrak hem aan een bepaald niveau dat noodzakelijk is voor praktische toepassingen. In de beginjaren 40, toen de wetenschap en techniek inmiddels weer wat verder waren, presenteerde Isaac Asimov "Drie principes voor het ontwikkelen van robots". Door dit werk verdween alle abstractie en tegenstrijdigheid m.b.t. robots, vanwege het zeer duidelijke en logische verhaal.

De drie principes waren:

- 1) Robots mogen mensen geen pijn doen of mensen negeren die pijn wordt aangedaan.
- 2) Robots moeten aan de orders van mensen gehoorgeven, zolang deze niet in strijd zijn met #1
- 3) Robots moeten zichzelf beschermen, zolang dit niet in strijd is met #1 en #2

Deze principes geven een ideaal beeld van dit technologische product.



c) Een update van de huidige stand van zaken

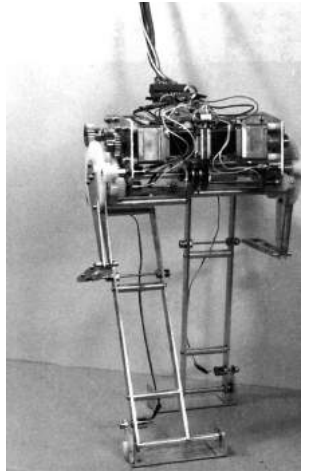
Vandaag de dag kunnen robots niet functioneren zonder de ontwikkeling van computersystemen en besturingstechnologie, waarmee gedurende de Tweede Wereldoorlog werd begonnen. Samen met de ontwikkeling van draadloze communicatie in diezelfde periode, werden ook de prestaties van de vacuümbuizen sterk verbeterd. Deze ontwikkelingen hebben de basis gelegd voor de hedendaagse robottechnologie. Hoewel, in het stadium dat vacuümbuizen geïntegreerd werden in computers, was het nog niet mogelijk om de robots te produceren zoals wij die nu kennen. Momenteel worden er bepaalde eisen gesteld voor wat betreft het functioneren van robots. Een voorbeeld daarvan is dat robots niet slechts één bepaalde handeling kunnen verrichten, maar dat ze d.m.v verschillende computerprogramma's meerdere acties kunnen verrichten.



De computer vormt hierbij het brein van de robot en daarom heeft het technologische niveau van computersystemen een grote invloed op de robot-technologie. Het is dus duidelijk dat hedendaagse robots nooit konden worden geproduceerd in combinatie met de computers die destijds met vacuümbuizen werkten.

Met de komst van de nog verder ontwikkelde IC's (LSI en VLSI), werd de werkings-snelheid van de computers aanzienlijk verhoogd en werd het formaat van de computers nog kleiner.

Deze ontwikkelingen hadden op hun beurt ook weer een positieve invloed op de verdere ontwikkeling van de robots.



Korte tijd later werd begonnen met de ontwikkeling van apparatuur (sensoren) die beschikte over de vijf zintuigen waarover ook de mens beschikt: zien, horen, voelen, ruiken en proeven.

Zoals gezegd werd het woord robot bedacht door Capek in zijn toneelstuk "Rur". Je kunt echter iets interessants zien met betrekking tot de denkbeeldige robots uit vroegere tijden en de hedendaagse robots gebaseerd op technologie en wetenschap. De hoogstaande horlogetechniek van nu vindt zijn oorsprong in een geautomatiseerde pop werkend op een vernuftig raderwerk die voor het eerst werd getoond tijdens een ballet in 1870. Dit is een duidelijk voorbeeld van een zich steeds verder ontwikkelende techniek.

De robots reflecteren dus steeds het technologische niveau in een bepaalde periode uit het verleden.



2.3 De huidige en toekomstige Robots

Robots van nu:

Technologische ontwikkelingen maakten het mogelijk dat het idee van de robot werkelijkheid kon worden. Sommige robots hebben menselijke trekjes, maar over het algemeen lijken de robots die vandaag de dag worden gebruikt niet op de mens.

a) Industriële robots

Industriële robots zijn over het algemeen machinale systemen die bepaalde functies van de mens in een industrieel proces overnemen. Vanaf de jaren 50 en 60 werden voor dit doel op grote schaal robots geproduceerd. Robotarmen zijn nu beschikbaar in simpele tot technisch uiterst gecompliceerde uitvoeringen en in vele soorten en maten.

De grotere types die hele zware gewichten kunnen tillen tot de kleinere modellen met dezelfde afmetingen als de menselijke arm.

Ze worden standaard bestuurd door microprocessors. Allerlei ontwikkelingen zijn nog gaande voor wat betreft de robottaal en besturingssoftware.

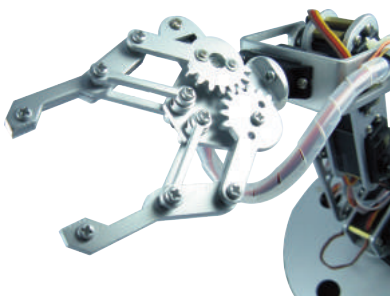


b) Andere industriële robots

Behalve de robots die in het productieproces worden ingezet, zijn er ook robots die van belang zijn in de wetenschappelijke omgeving, bijv. in de ruimte, bij diepzeeproeven en in kernreactors. Verder zijn er robots die in de agrarische sector en in de visserij hun diensten bewijzen. Ook zijn er robots die ramen en vloeren schoonmaken in (kantoor)gebouwen.

c) Intelligente robots

Intelligente robots kunnen geheel zelfstandig taken overnemen van de mens en ze succesvol uitvoeren. Het onderzoek in hoeverre intelligente robots technisch gezien gerealiseerd kunnen worden, werd opgestart tijdens het eerste internationale congres over intelligente robots dat werd gehouden in de Verenigde Staten in 1969. Maar het feit blijft dat e.e.a. in de praktijk niet zo eenvoudig te verwezenlijken is.



d) Overige robottypes

Ook in de medische wereld kunnen we diverse voorbeelden geven, waarbij robots een belangrijke rol spelen. Zo zijn er robots in de vorm van een menselijk lichaam ontwikkeld voor medische studenten. Deze robots vormen een trainingsobject voor het oefenen van medische technieken en handelingen zoals anesthesie, hartmassage, kunstmatige beademing en het werken met een infuus. De robot heeft interne sensors die bijvoorbeeld pols- en bloeddruk aangeven, zodat gezien kan worden of een handeling succesvol is geweest.



Er worden al sinds lange tijd speelgoedrobots gemaakt, die werken met mechanische basisonderdelen zoals een versnelling en tandwielen. In de jaren 80 verschenen er robots met ingebouwde en zelf te programmeren computers op de markt. Dit waren robots die bijvoorbeeld konden lopen of hun handen en vingers konden bewegen.

In 1996 introduceerde Honda de P2 robot. Deze robot heeft net als de mens twee armen en twee benen en kan niet alleen op een vlakke ondergrond lopen, maar ook op een hobbelige of steile ondergrond. De robot kan zelfs een trap op- en aflopen.

De P2 is één van de meest mensachtige robots die ooit is geproduceerd en biedt diverse nieuwe mogelijkheden voor de relaties tussen mens en robot..



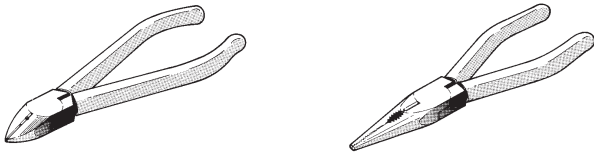
3. Mechanica

Lees dit eerst voordat je begint met de montage!

3.1 Benodigde gereedschappen:

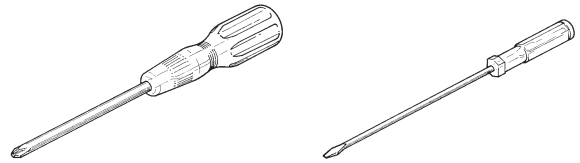
Het juiste gereedschap is het halve werk!

Kniptang & Punttang

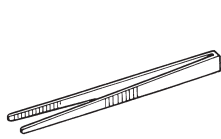


Miniatuur (150 mm).

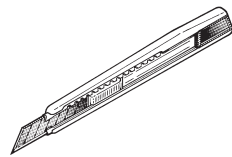
Schroevendraaierset



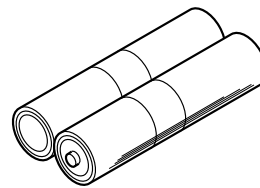
Gebruik altijd de juiste afmeting elektronica-schroevendraaier



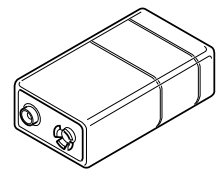
Pincet



Hobbymes



2 St. AA-Batterijen



9V Blokbatterij

3.2 Tips en Trucs:

Uitnijden van plastic onderdelen

Ga niet vooruit werken, volg de Stap-voor-Stap handleiding om een foute montagevolgorde te voorkomen. Wanneer je de bouwbeschrijving nauwkeurig volgt en zo nu en dan de foto op de verpakking bekijkt, krijg je in één keer een perfect werkende robot.

Snij de onderdelen pas uit op het moment dat je ze nodig hebt. Ze zijn namelijk genummerd in de sets. De nummers staan NIET op de onderdelen zelf. Alle onderdelen passen perfect in elkaar, je hoeft dus niets met geweld in elkaar te duwen. Werk rustig en lees eerst de HELE instructie door voordat je begint.

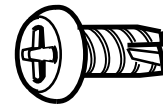


Gebruik voor het uitsnijden een scherp hobbymes of knip ze uit met een scherpe tang. Snij of knip voorzichtig en zo glad mogelijk langs het onderdeel. Werk met behulp van een mes of vijl de eventuele scherpe randen bij.

LET OP !

Snij geen onderdelen uit voordat je ze nodig hebt.

3.3 Montage van de onderdelen:



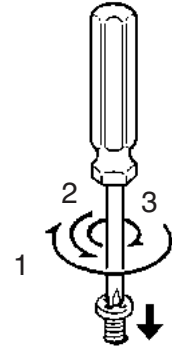
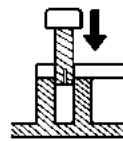
Zelfsnijdende (tappende) schroeven (Parkers)

Een zelftappende schroef (ook wel parker genoemd) heeft dezelfde eigenschappen als een hout-schroef namelijk, schroefdraad snijden en vastschroeven in één handeling. Vergeleken met een gewone schroef is de schroefdraad grover en het eind puntiger.

Deze zelftapschroeven hebben een inkeping, wat het snijden van de schroefdraad vergemakkelijkt.

De juiste manier om zelftappende schroeven vast te schroeven is;

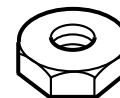
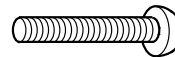
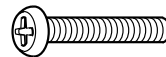
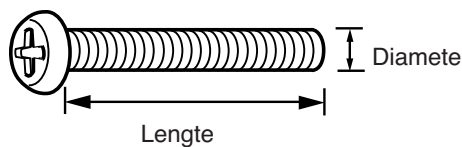
- 1 Schroef indraaien
- 2 Een beetje losschroeven
- 3 Dan weer vastdraaien



LET OP !

Als je deze schroeven te vaak los -en vastschroeft, wordt het gat groter en passen ze niet goed meer

Bouten en moeren



Moer



Borgmoer

Bouten en moeren kun je het beste goed vastdraaien, zodat ze op den duur niet lostrillen. Wil je ze nog extra beschermen tegen het lostrillen, doe er dan na het vastdraaien een beetje nagellak op.

Op deze manier kun je ze later ook weer eenvoudig losdraaien als je dat zou willen. *Gebruik geen loctite of andere lijmen.* Anders kun je de schroeven nooit meer losdraaien als dat nodig mocht zijn. Er zijn ook speciale borgmoeren. Deze blijven uit zichzelf al stevig vastzitten.

Het type bout wordt aangegeven d.m.v. de dikte en de lengte. Bijv.: Een bout met de aanduiding **M3 x 20 is 3 mm dik en 20 mm lang.**

Bij moeren geven we alleen de diameter weer. Bijv.: **M3 is een moer voor een bout van 3 mm.**

LET OP !

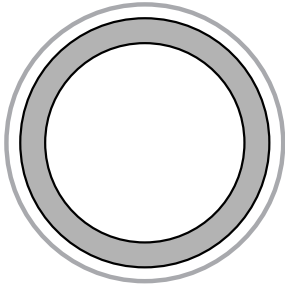
Lees altijd de waarschuwingen in de zwarte tekstblokken!

Enkele onderdelen (Bijv. de aandrijfmodule) zijn al met smeermiddelen behandeld. Bouw de robot op een veilige ondergrond zonder deze te beschadigen.

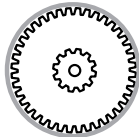
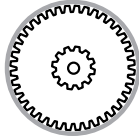
4. Onderdelenlijst P!MOT

De onderdelenlijst

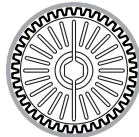
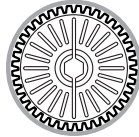
Banden 2 St.



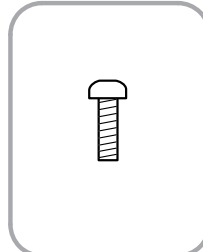
Koppeltandwiel vlak 4 St.



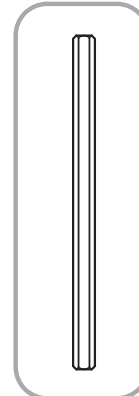
Vlak tandwiel met ashouder 2 St.



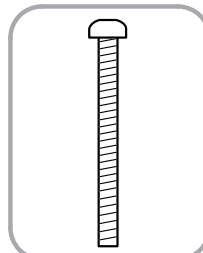
Schroef klein 14 St.



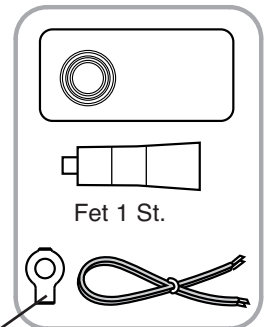
Zeskant-as 2 St.



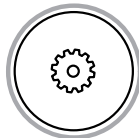
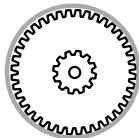
Schroef lang 3 St.



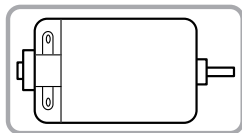
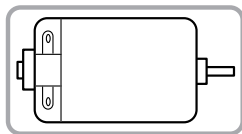
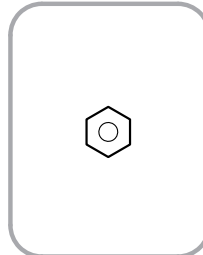
Batterijcontact 1 St.



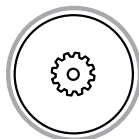
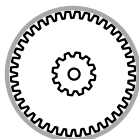
Kroontandwiel 2 St.



Moer 17 St.



Motoren 2 St.

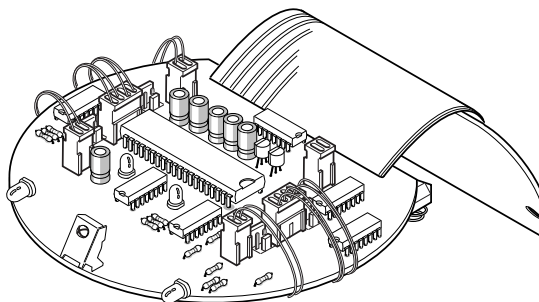


Aandrijftandwiel 2 St.

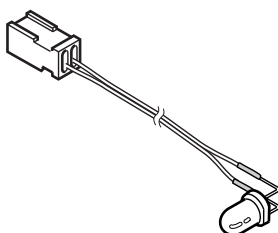
Motorenklem 4 St.

Kabel 2 St.

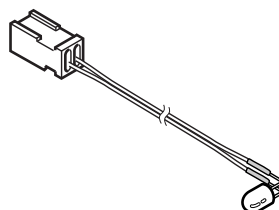
Printplaat 1 St.



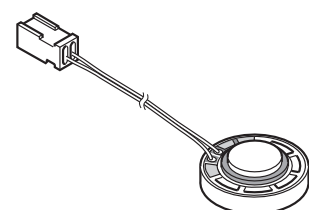
LED (Rood/ Wit) 2 St.



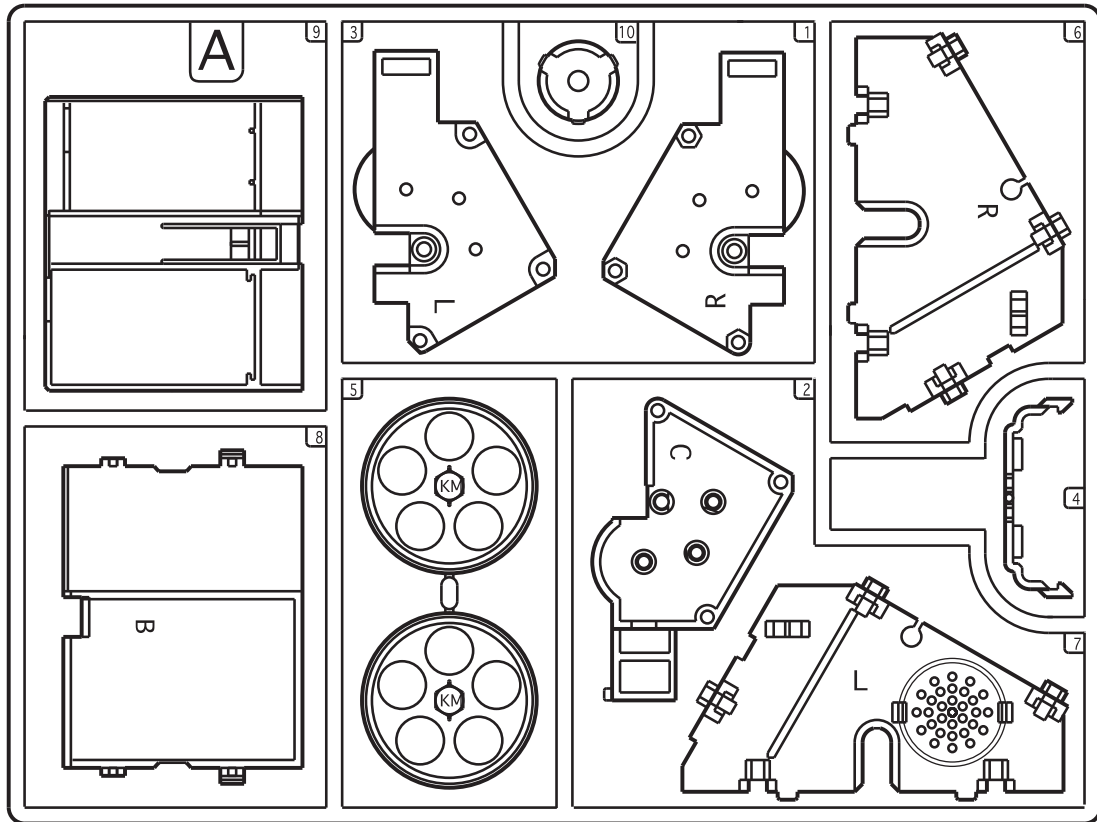
Fototransistor 2 St.



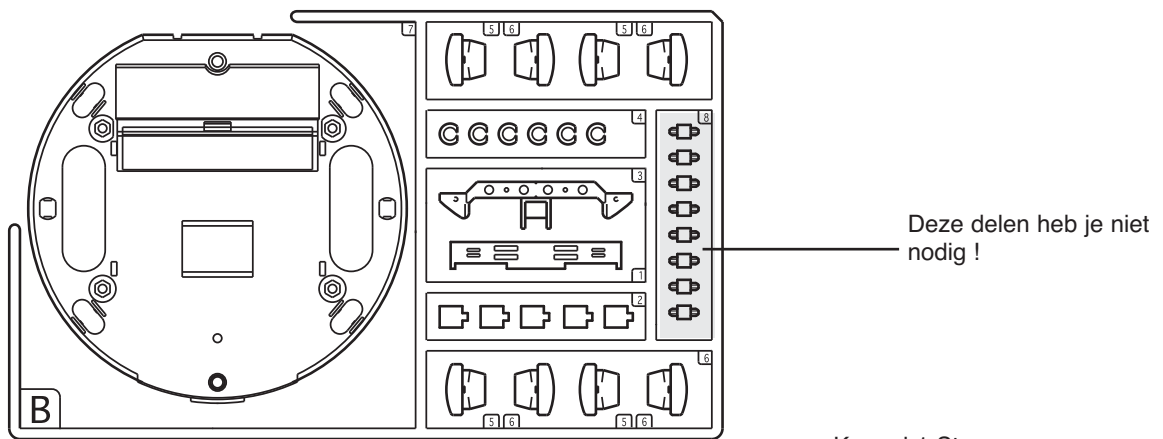
Luidspreker 1 St.



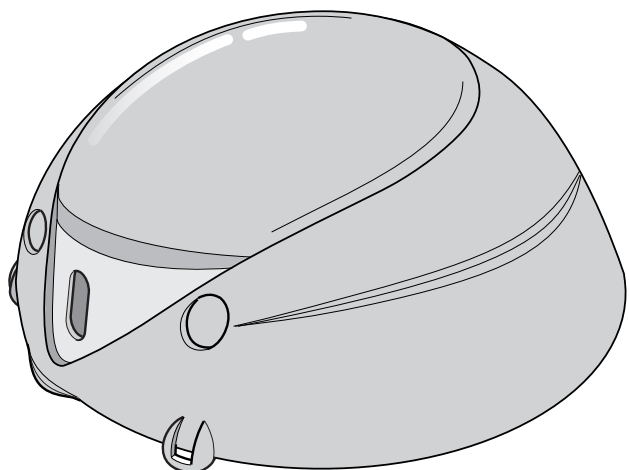
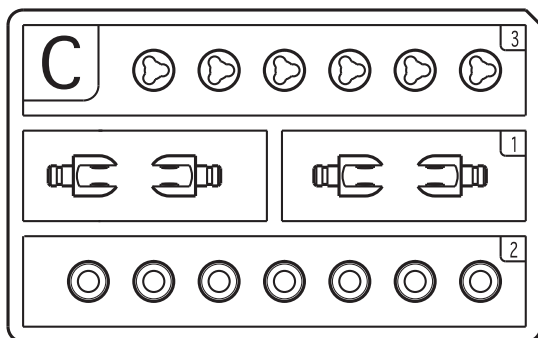
Set A 1 St.



Set B 1 St.



Set C 1 St.



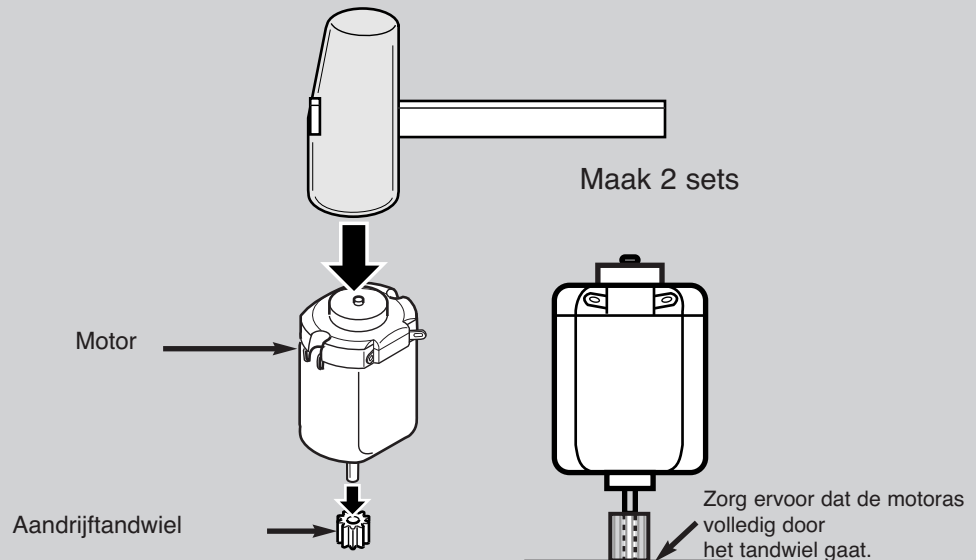
5. Assemblage van chassis

De nummer van de onderdelen bevinden zich meestal niet op het onderdeel zelf! Knip of snijd ze daarom pas uit als je ze echt nodig hebt!

Benodigde onderdelen:

Motoren (2 st.)
Aandrijftandwiel (2 st.)

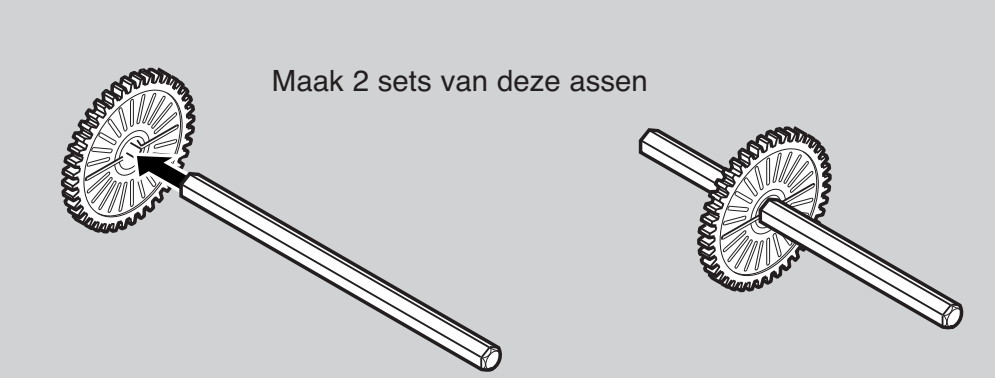
5.1 Montage van het aandrijftandwiel



Benodigde onderdelen:

Zeshoekige as (2 st.)
Vlak tandwiel met een sleuf (2 st.)

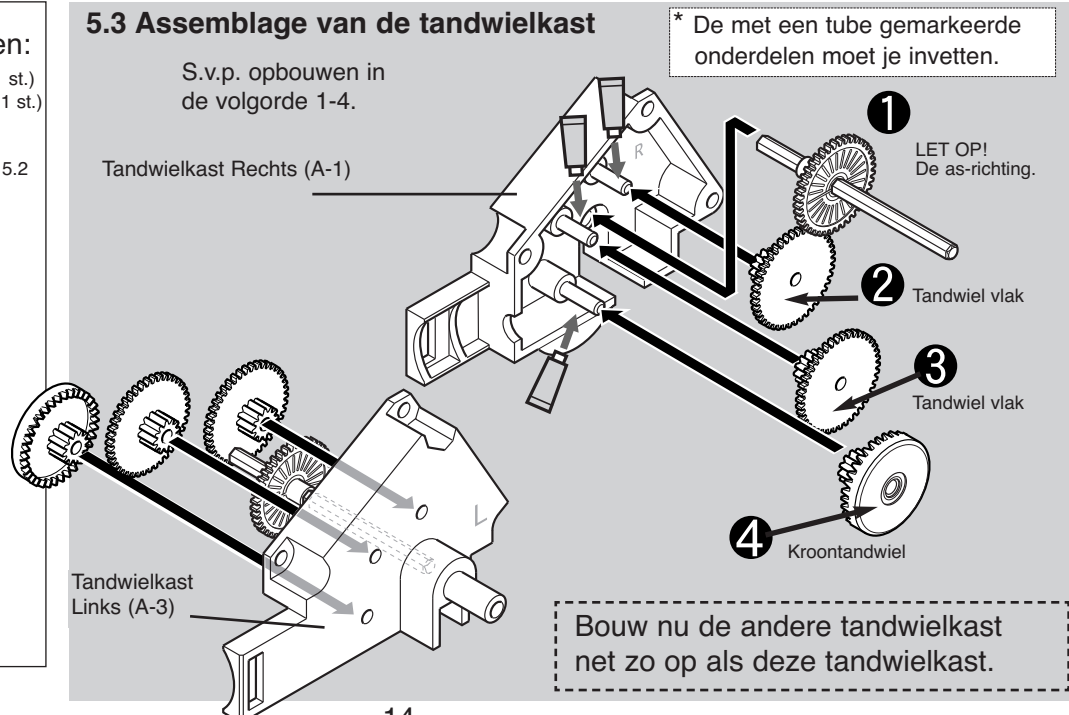
5.2 Assen assembleren



Benodigde onderdelen:

Tandwielkast – linkerhelft (A-1, 1 st.)
Tandwielkast – rechterhelft (A-3, 1 st.)
Tandwiel (vlak, 4 st.)
Kroontandwiel (2 st.)
Geassembleerde assen uit stap 5.2

5.3 Assemblage van de tandwielkast

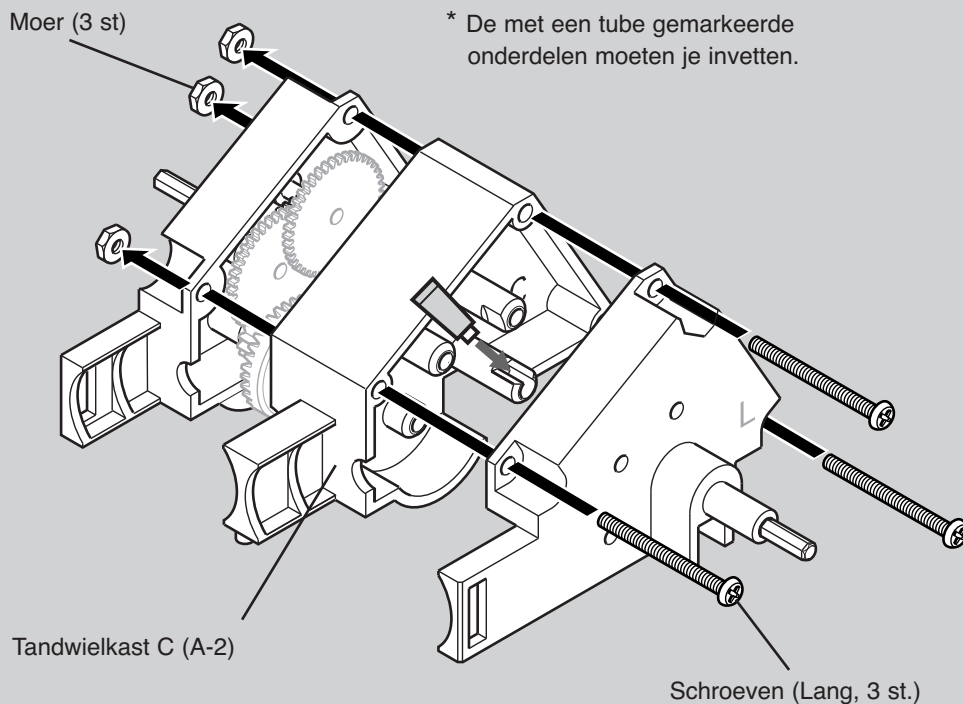


Benodigde onderdelen:

Benodigde onderdelen:

- Tandwielkast C (A-2, 1 st.)
- Schroef (lang, 3 st.)
- Moer (3 st.)
- Geassembleerde deel uit stap 3

5.4 Assemblage van de motoren

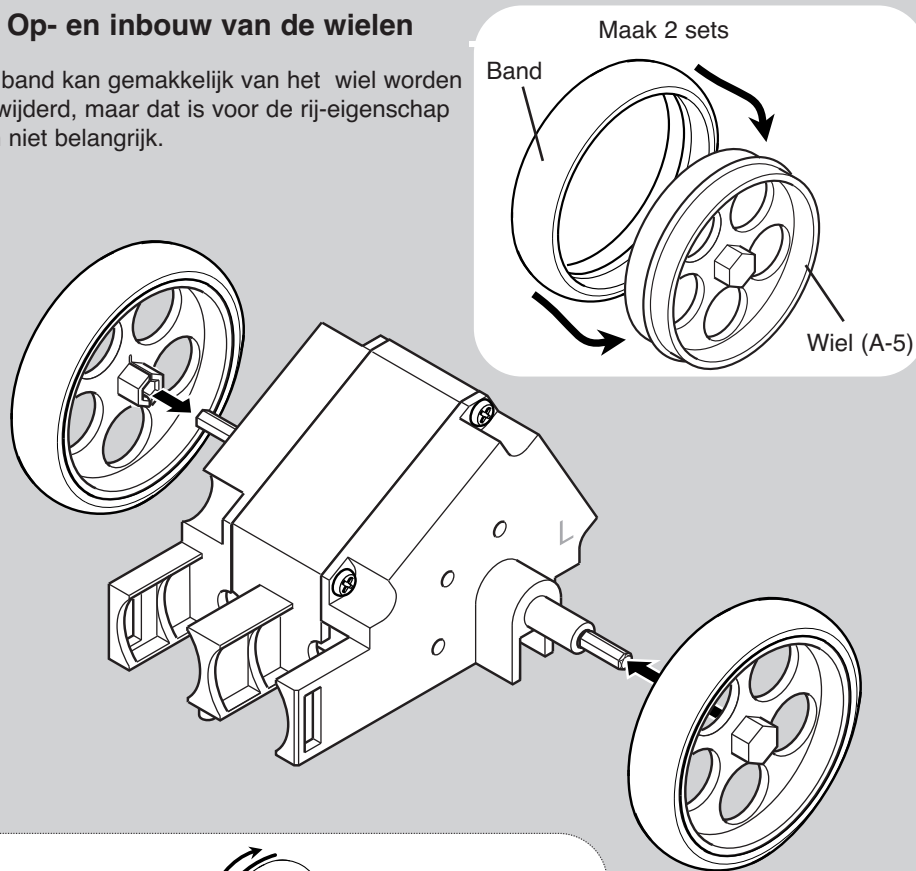


Benodigde onderdelen:

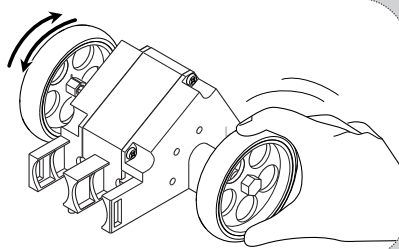
- Wiel (A-5, 2 st.)
- Band (2 st.)
- Geassembleerde deel uit stap 5.4

5.5 Op- en inbouw van de wielen

* De band kan gemakkelijk van het wiel worden verwijderd, maar dat is voor de rij-eigenschappen niet belangrijk.



* Laat de banden draaien en controleer ook of ze echt soepel draaien.

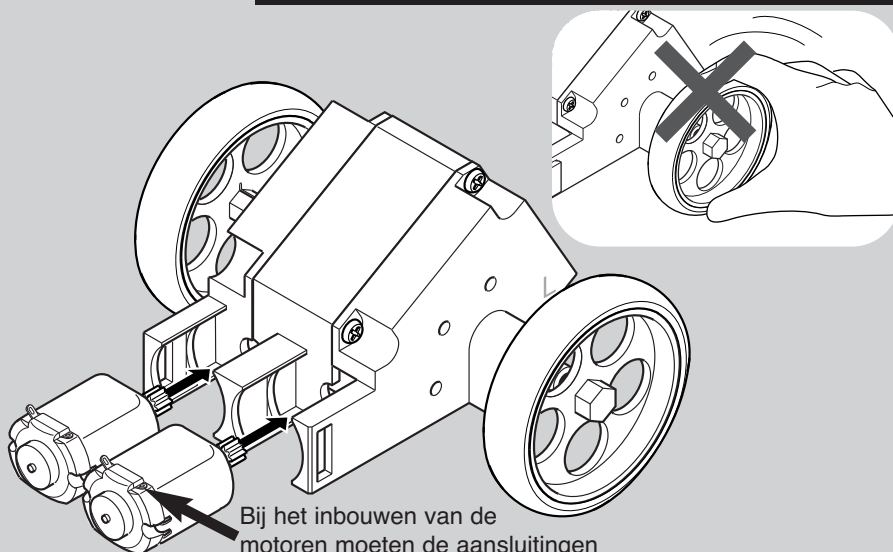


Benodigde onderdelen:

Motorklem (A-4, 1 st)
Geassembleerde deel uit stap 5.1
Geassembleerde deel uit stap 5.5

5.6 Inbouwen van de motoren

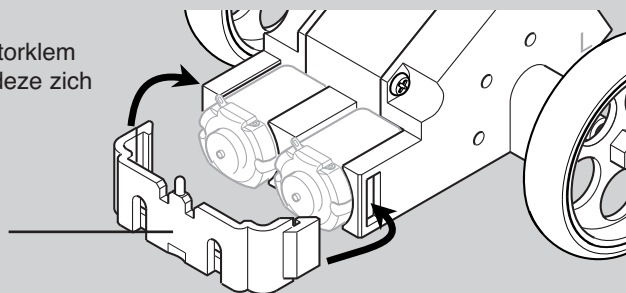
Indien je na het inbouwen van de motor de wielen met de hand verdraait wordt het tandwielsysteem beschadigd. De omzettingfactor van het PIMOT-tandwielsysteem is zeer hoog. Draai de wielen na het inbouwen van de motoren niet meer met de hand.



Bij het inbouwen van de motoren moeten de aansluitingen zich aan de bovenzijde bevinden.

Druk de motorklem vast totdat deze zich vastklikt.

Motorklem (A-4)



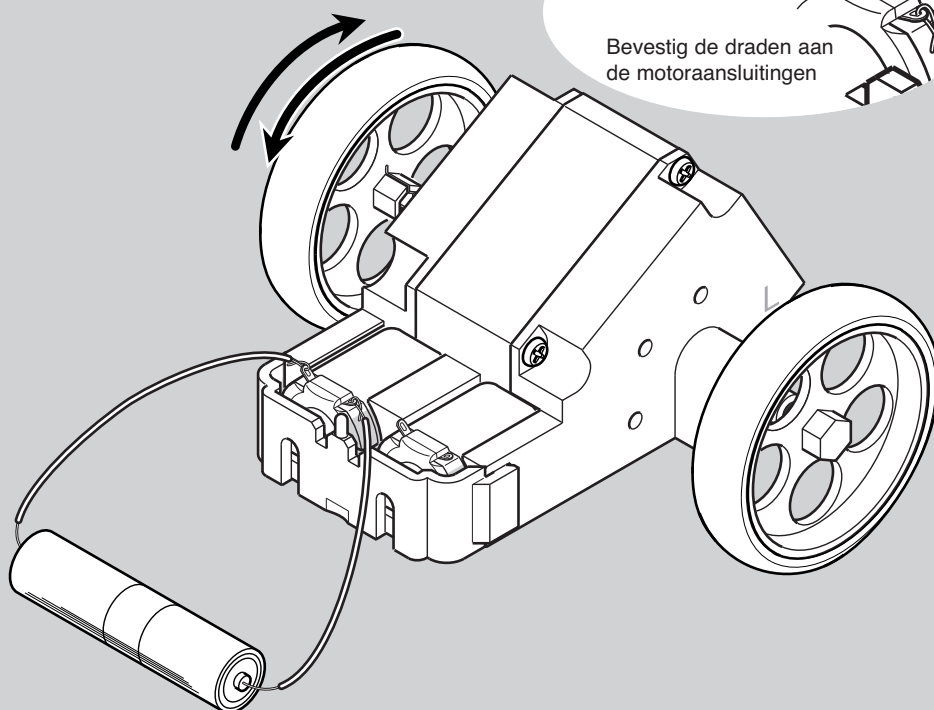
Benodigde onderdelen:

Alkaline Batterij AA, 1 st
Draad voor het proefdraaien
Geassembleerde deel uit stap 5.6

5.7 Controle van het tandwielsysteem



Bevestig de draden aan de motoraansluitingen



Indien het wiel niet draait, moet je de volgende punten controleren:

Is de inbouwpositie van de vlakke tandwielen en het kroontandwiel correct?

Heb je de betreffende motor in stap 5.6 correct ingebouwd?

Heb je alle aangegeven onderdelen goed ingevet?

Zijn de schroeven goed vastgeschroefd?

Is de batterij misschien leeg?

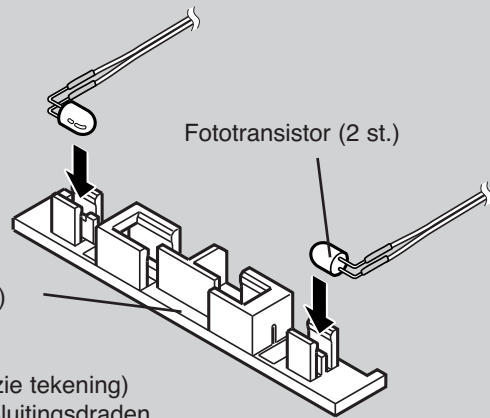
- 1) Bevestig de draad zoals aangeduid in de tekening aan de aansluitklem voor de motor. Sluit nu de AA Batterij aan en controleer of het wiel draait. Als het wiel goed draait, moet je de aansluitingen omwisselen en vervolgens de omgekeerde draairichting controleren
- 2) Controleer op dezelfde wijze de werking van de tweede motor.

Benodigde onderdelen:

Fototransistor (2 st.)
LED (rood, 2 stuks)
Behuizing impulsensor (B-1, 1 st)
Afdekking impulsensor (B-2, 2 st.)

5.8 Assemblage van de impulsensoren

- 1) Bevestig de foto-transistoren in de behuizing voor de impulsensor.

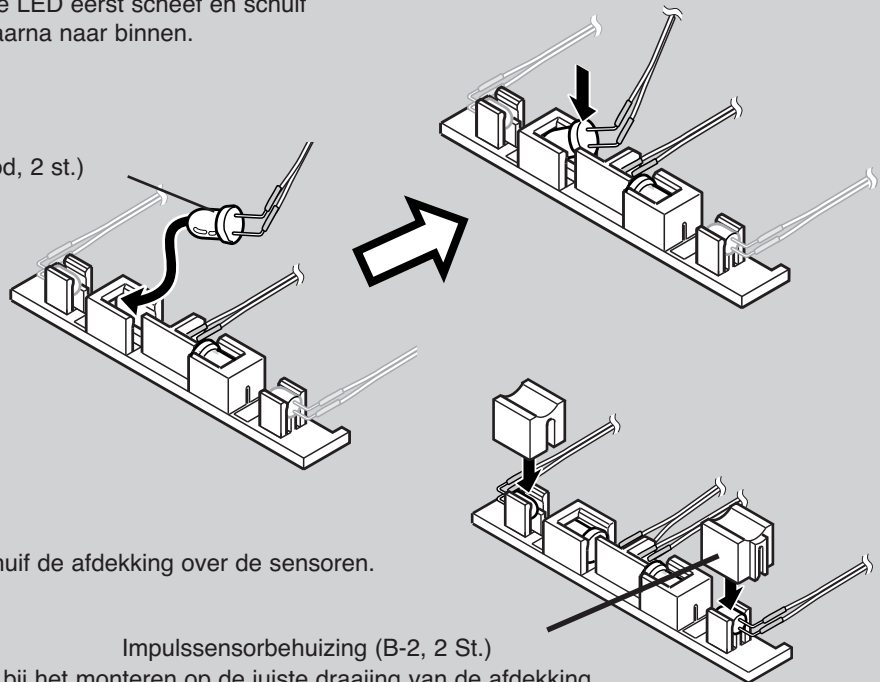


Impulsensorbehuizing (B-1)

- 2) Montage van de LED's (zie tekening)
Bevestig daarna de aansluitingsdraden.

* De inbouw is niet zo eenvoudig.
Houd de LED eerst scheef en schuif deze daarna naar binnen.

LED (rood, 2 st.)



- 3) Schuif de afdekking over de sensoren.

Impulsensorbehuizing (B-2, 2 St.)

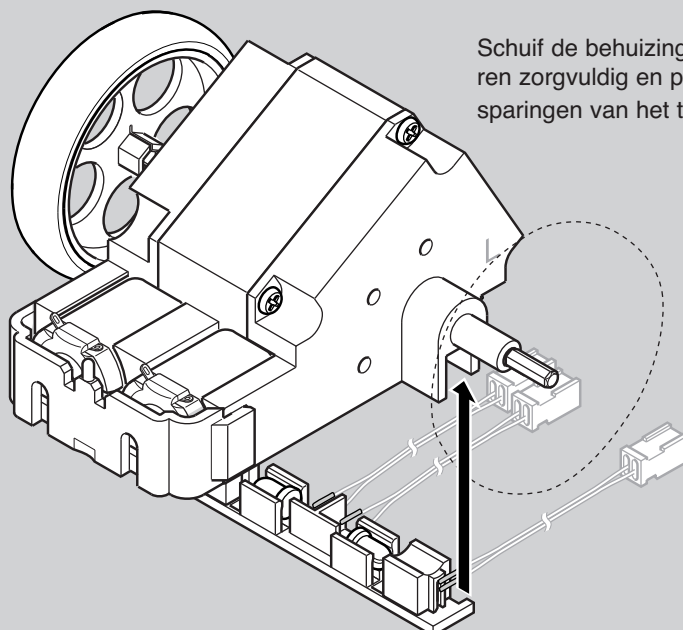
Let bij het monteren op de juiste draaiing van de afdekking.

Benodigde onderdelen:

Set uit stap 5.7
Set uit stap 5.8

5.9 Inbouw van de impulsensoren

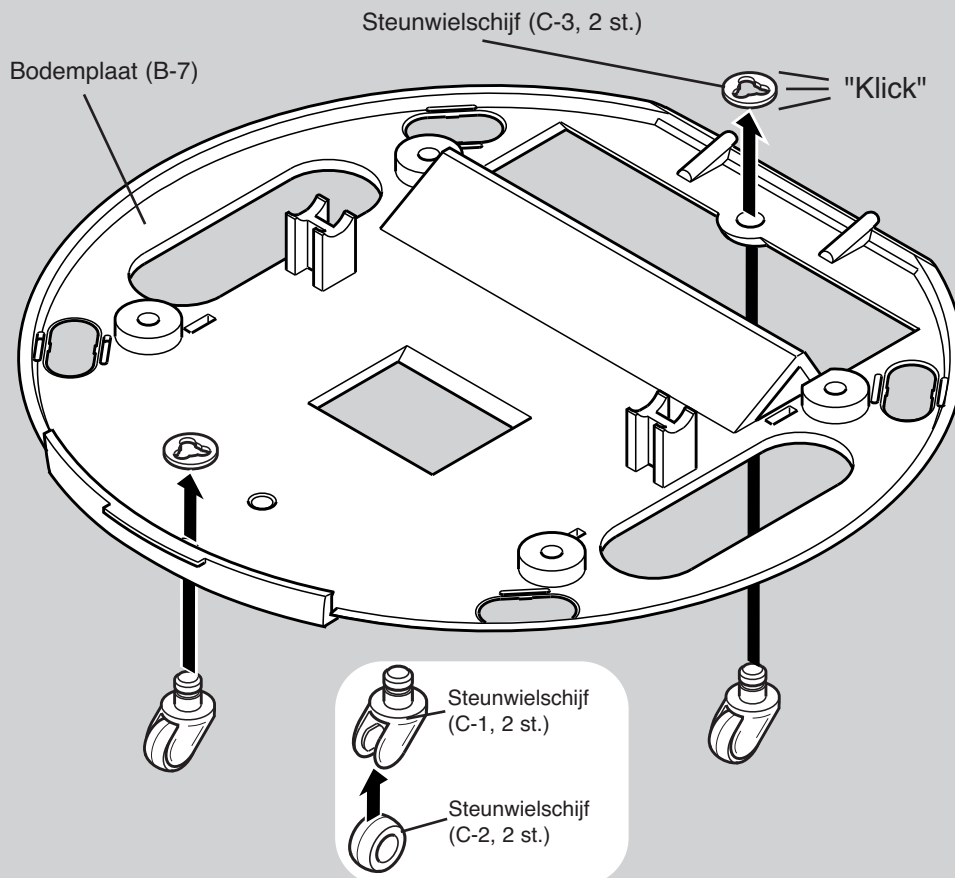
Schuif de behuizing van de sensoren zorgvuldig en precies in de uitsparingen van het tandwielsysteem.



Benodigde onderdelen:

- Houder voor steunwiel (C-1, 2 st.)
- Steunwiel (C-2, 2 st.)
- Steunwiel (C-3, 2 st.)
- Bodemplaat (B-7, 1 st.)

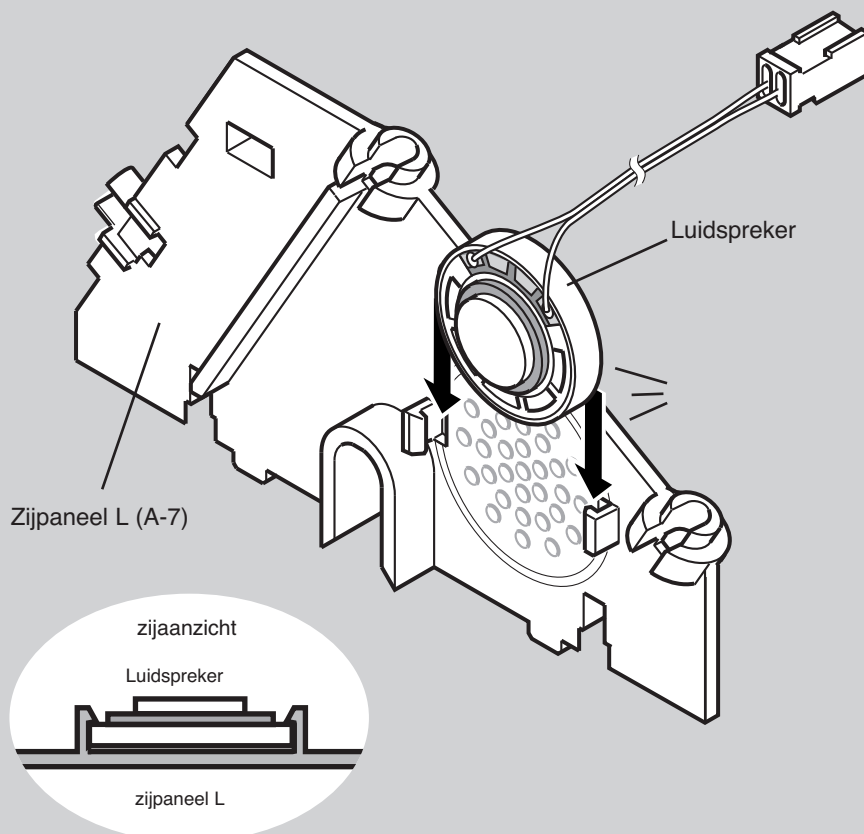
5.10 Inbouw van het voorwiel



Benodigde onderdelen:

- Luidspreker (1 st.)
- Zijpaneel L (A-7, 1 st.)

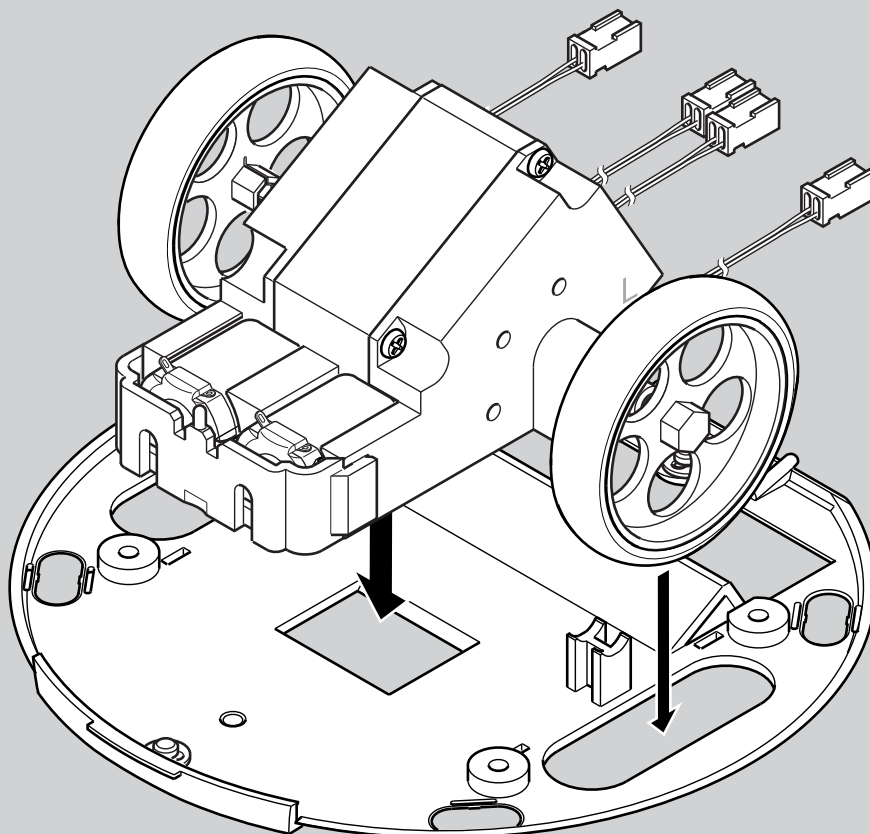
5.11 Inbouw van de luidspreker



Benodigde onderdelen:

Set uit stap 5.9
Set uit stap 5.10

5.12 Montage van het tandwielsysteem

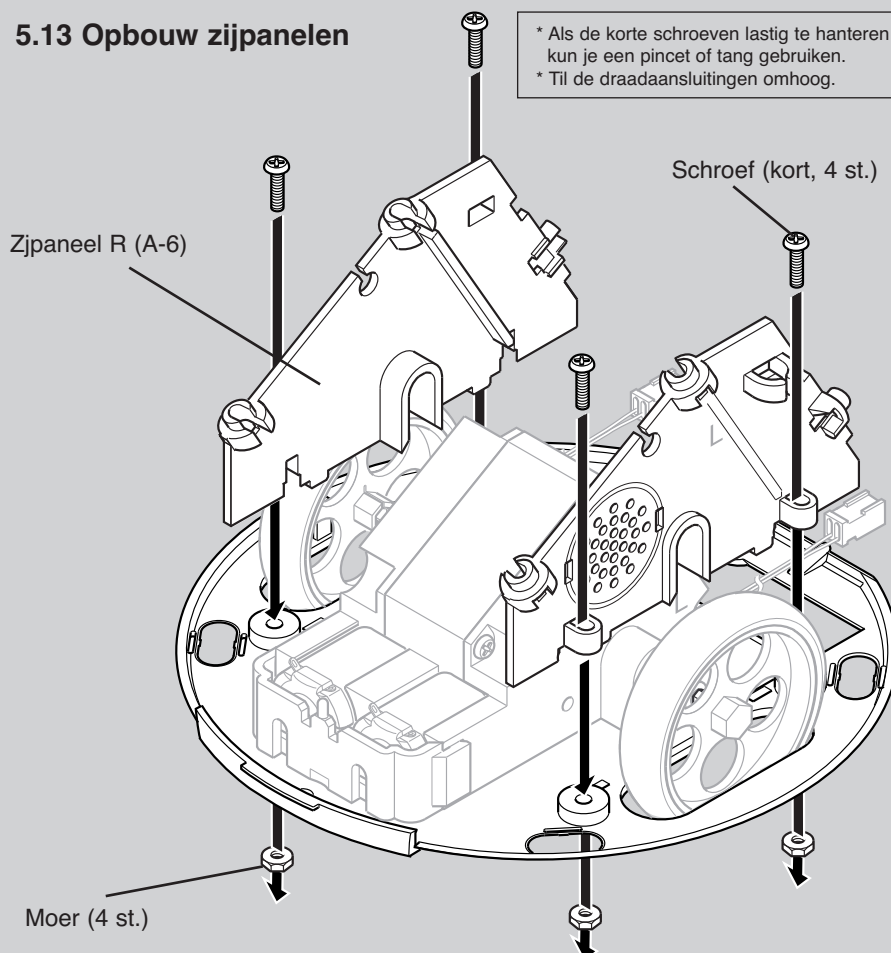


Benodigde onderdelen:

Zijpaneel R (A-6, 1 st.)
Schroef (kort, 4 st.)
Moer (4 st.)
Set uit stap 5.11
Set uit stap 5.12

5.13 Opbouw zijpanelen

* Als de korte schroeven lastig te hanteren zijn kun je een pincet of tang gebruiken.
* Til de draadaansluitingen omhoog.



Benodigde onderdelen:

Printplaat (1 st.)
Infraroodafscherming (B-3, 1 st.)
Motoraansluitingen (4 st.)
Schroef (kort, 4 st.)
Moer (4 st.)

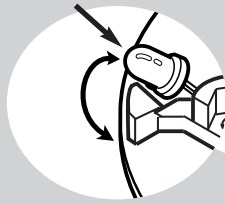
5.14 Montage van de infraroodafscherming

Bouw via de volgorde 1..2 op.

* Elektronica-componenten op de printplaat kunnen door elektrische ontladingen beschadigd worden.

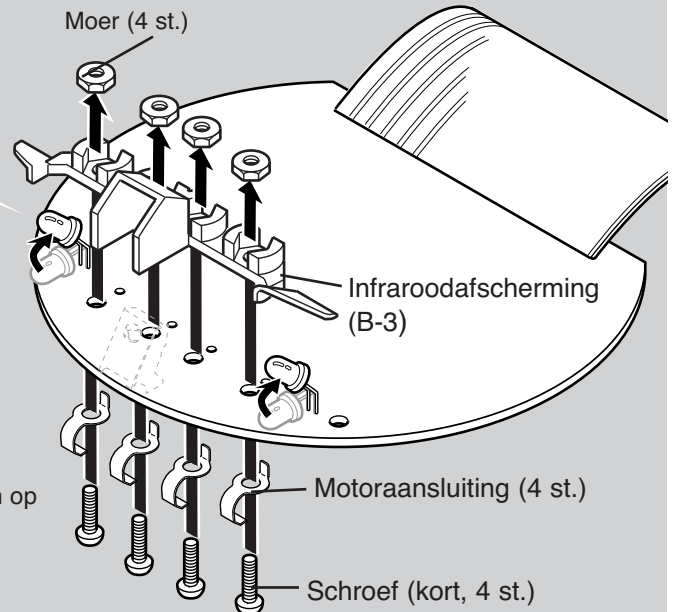
Daarom moet je zoveel mogelijk het aanraken van soldeerpunten vermijden.

Buig ongeveer 45 graden

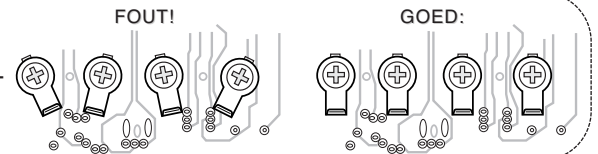


1
Buig de infrarood LED's wat naar boven. Plaats daarna de Infraroodafscherming op de printplaat.

2
Monteer de motoraansluitingen op de printplaat.



* Let op! dat de motoraansluitingen geen andere onderdelen raken. Dit kan eventueel beschadiging en storingen in de elektronische schakelingen veroorzaken.

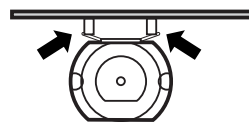


Benodigde onderdelen:

Schroef (kort, 4 st.)
Moer (4 st.)
Set uit stap 5.13
Set uit stap 5.14

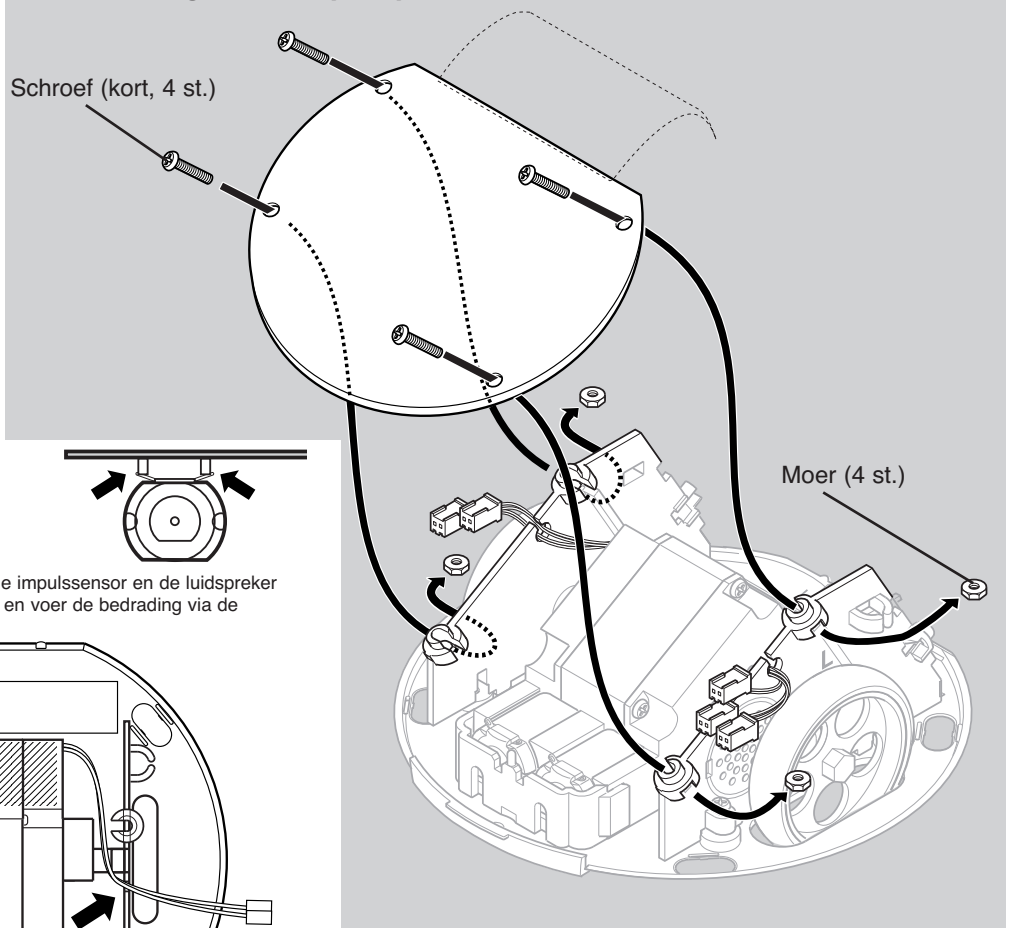
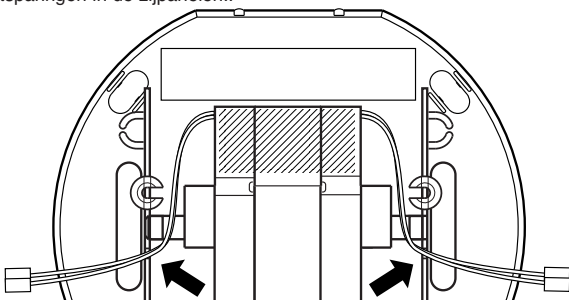
5.15 Montage van de printplaat

Schroef (kort, 4 st.)



* Let op, dat de motorcontacten uit stap 14 goed aansluiten op de contactaansluitingen voor de motor.

* Let erop, dat de aansluitdraden van de impulsensor en de luidspreker de gearceerde oppervlakte niet raken en voer de bedrading via de uitsparingen in de zijpanelen..



Benodigde onderdelen:

Behuizing voor de bodemsensor A (B-5, 4 stuks)

Behuizing voor de bodemsensor B (B-6, 4 stuks)

CdS (4 stuks, bevestigd aan de print met elektronica)

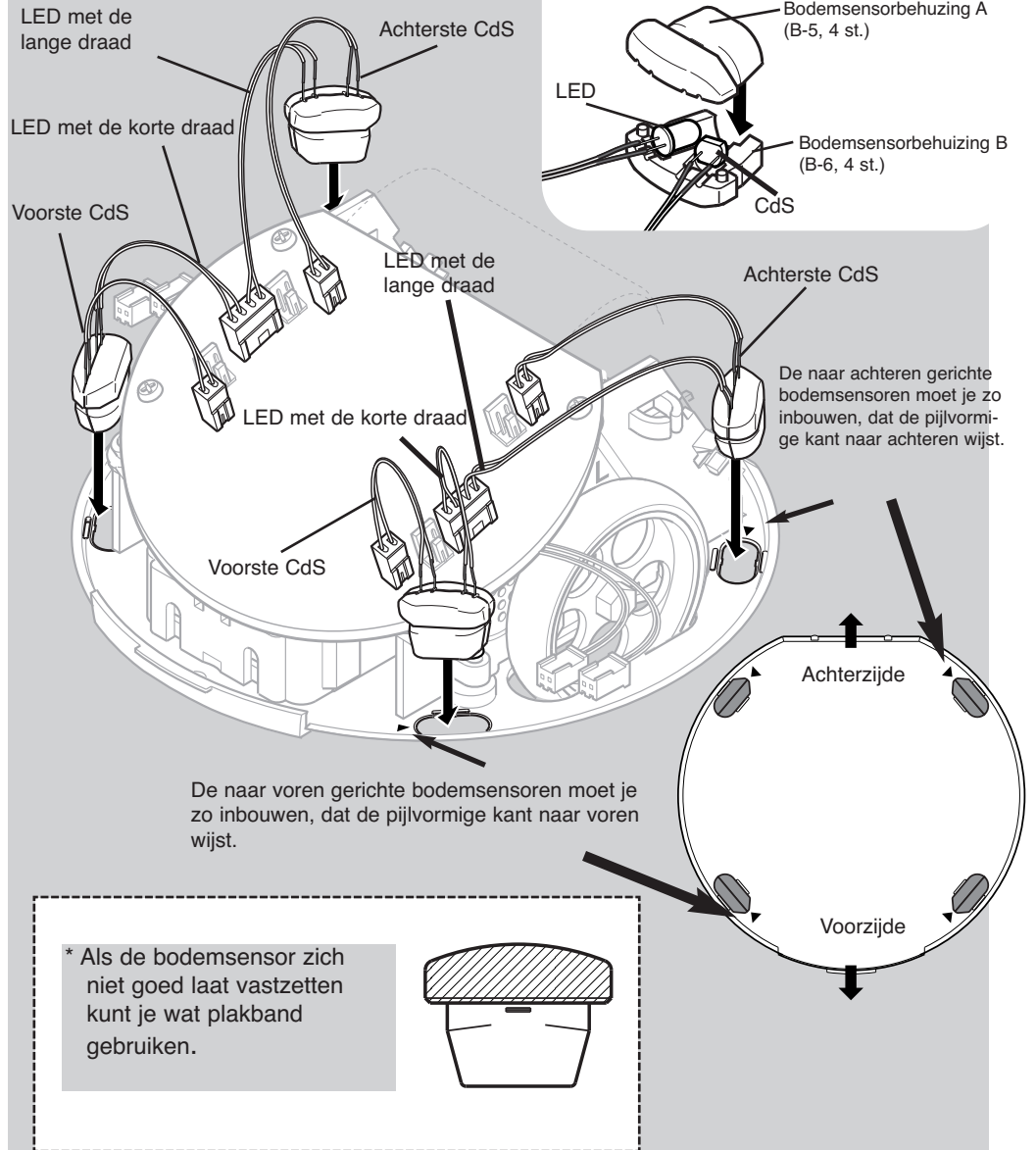
LED (4 stuks, bevestigd aan de print met electronica)

Geassembleerde set uit stap 5.15

CdS = lichtgevoelige weerstand

5.16 Inbouw bodemsensoren

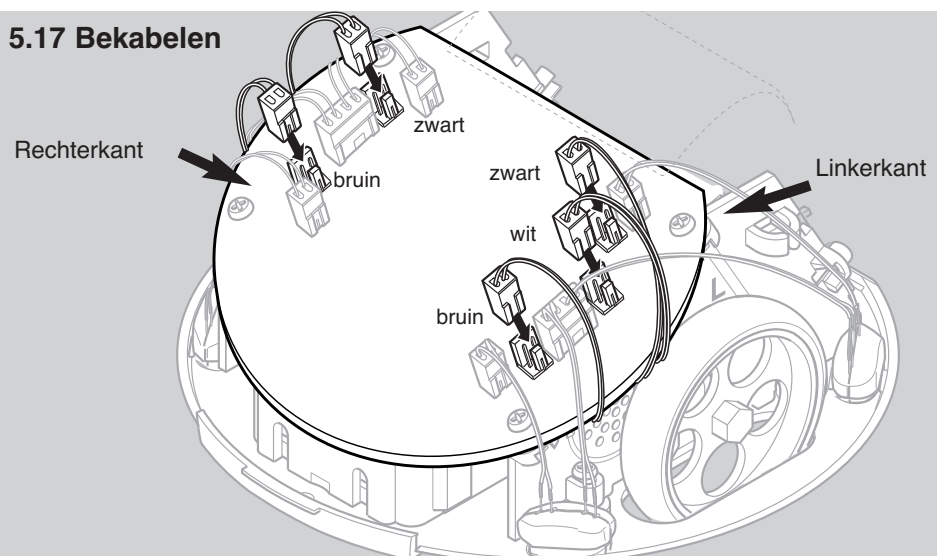
Maak 4 sets



Benodigde onderdelen:

Set uit stap 5.16

5.17 Bekabelen



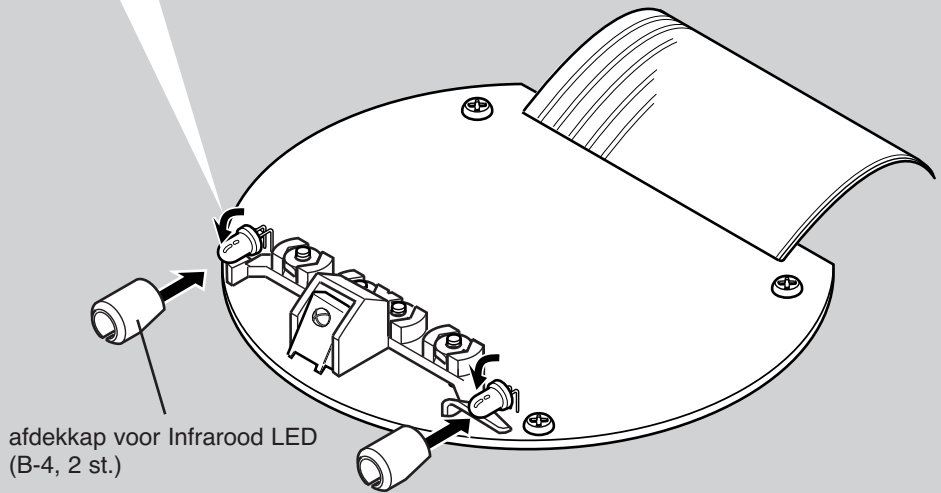
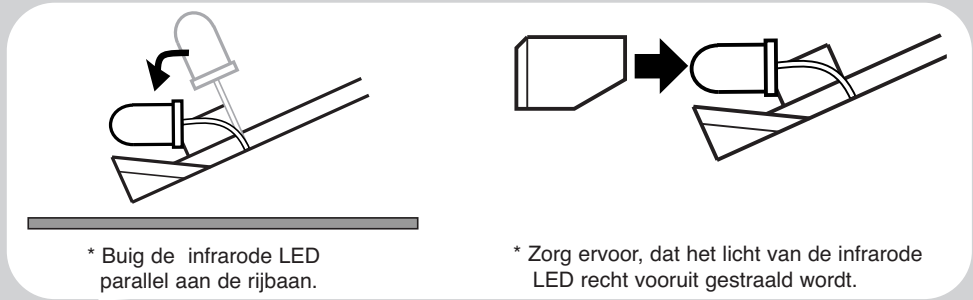
LED voor de Impulssensor (bruine aansluiting)	RL	Naar bruine aansluiting rechts
	LL	Naar brune aansluiting links
Fototransistor voor de Impulssensor (zwarte aansluiting)	MPR	naar zwarte aansluiting rechts
	MPL	Naar zwarte aansluiting links
Luidspreker	SP	Naar witte aansluiting

Benodigde onderdelen:

Afdekkap voor Infrarood LED
(B-4, 2 st.)

set uit stap 5.17

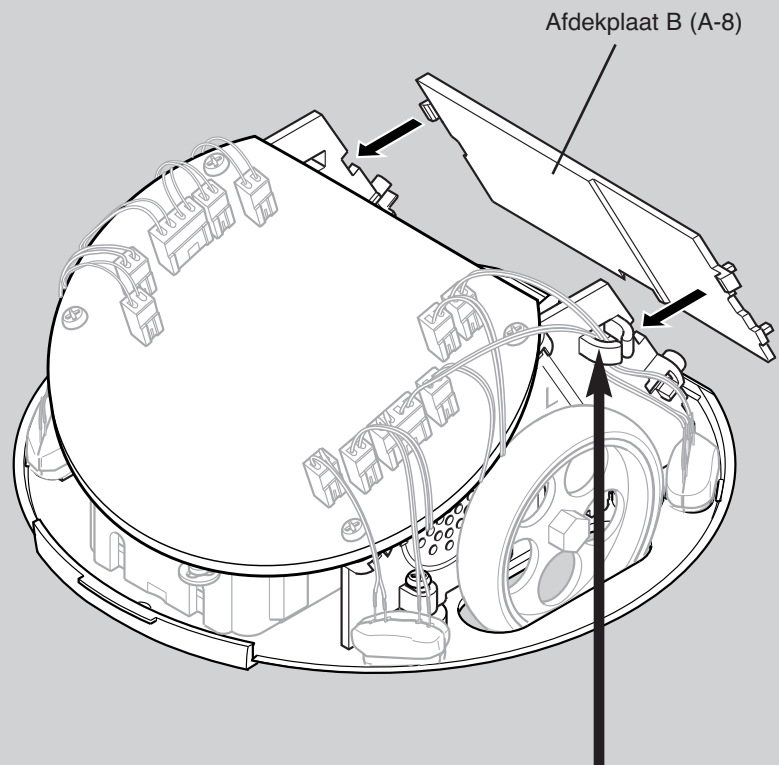
5.18 Montage van het afdekkapje voor de infrarode LED



Benodigde onderdelen:

Afdekplaat B (A-8, 1 St.)
Set uit stap 5.18

5.19 Montage van de afdekplaat B

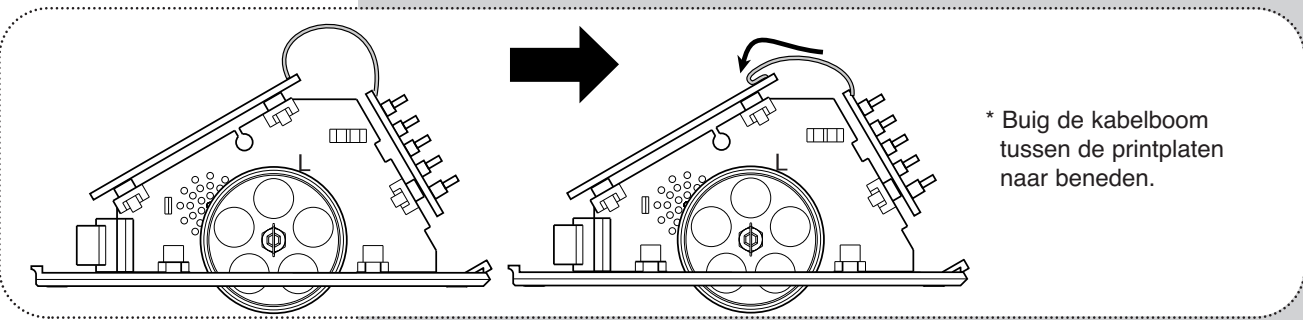
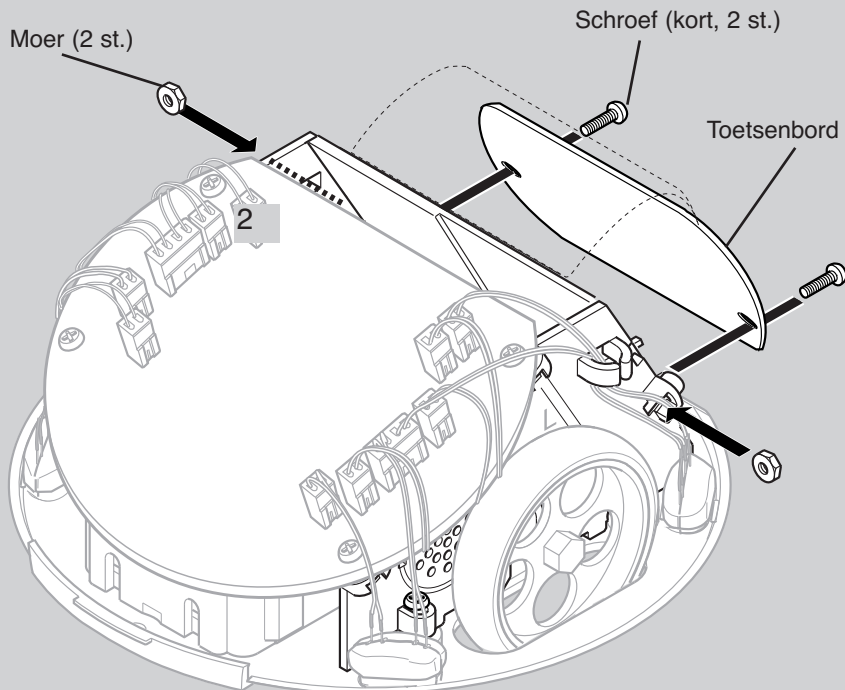


* Zorg ervoor, dat de bedrading van de achterste bodemsensoren door de uitsparingen in de zijwanden wordt gelegd.

Benodigde onderdelen:

Schroef (kort, 2 st.)
 Moer (2 st.)
 Set uit stap 5.19

5.20 Montage van de toetsenbord

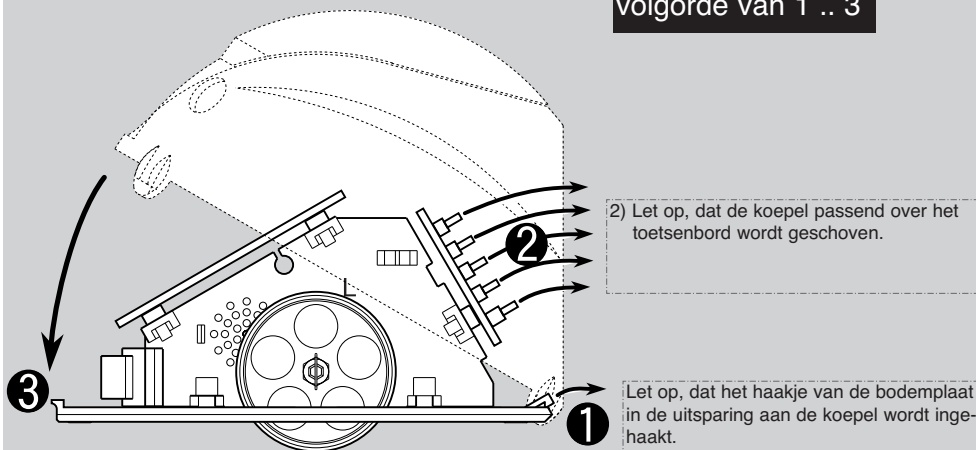


Benodigde onderdelen:

Koepel (1 st.)
 set uit stap 5.20

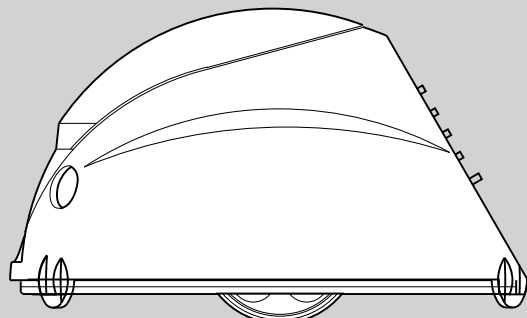
5.21 Montage van de koepel

Bouw op in de
 volgorde van 1 .. 3



3) Druk de koepel naar beneden tot deze zich met een klik sluit.

* De methode om de koepel te openen staat op bladzijde 27.

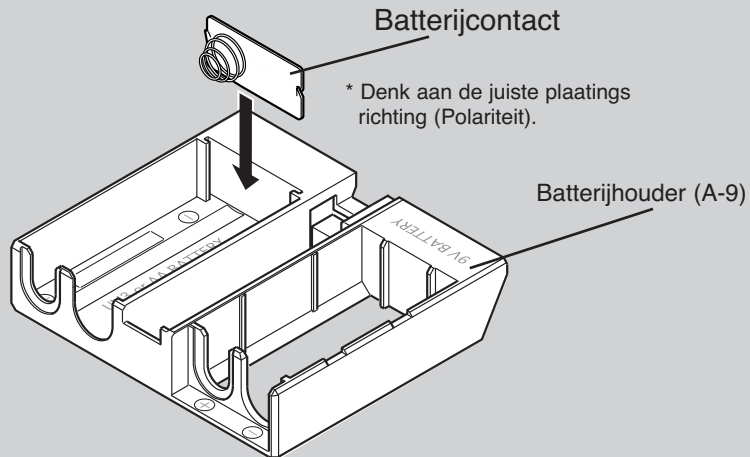


5.21 Montage van de batterijhouder

Benodigde onderdelen:

Batterijhouder (A-9, 1 St.)
Batterijcontact (1 St.)

5.21.1 Assemblage van de batterijhouder



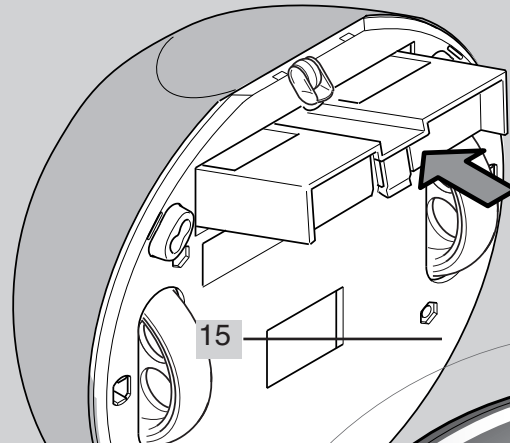
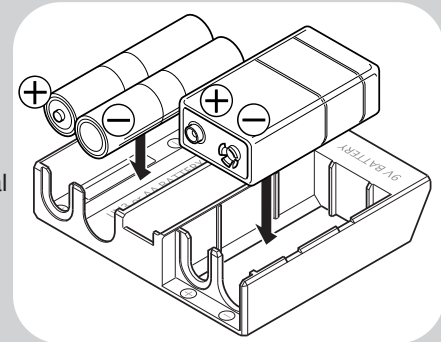
Benodigde onderdelen:

Alkaline batterij 1,5V (AA, geen onderdeel van de bouwdoos, 2st.)

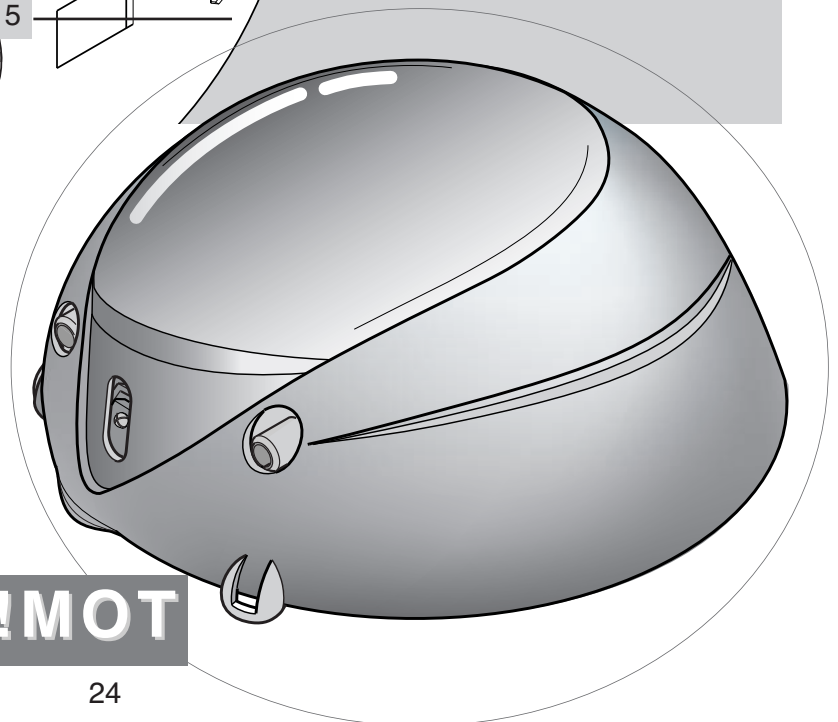
Alkaline batterij 9V (blok, geen onderdeel van de bouwdoos, 1st.)

5.21.2 Montage op het chassis

- * Denk aan de juiste polariteit + en - van de batterijen.
- * Zorg ervoor dat de hoofdschakelaar uit staat al je de batterijhouder in de robot plaatst.



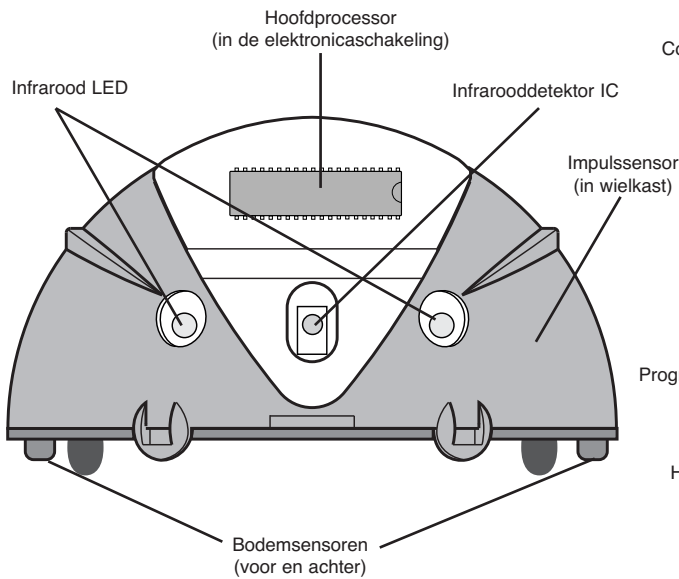
Schuif het compartiment in de behuizing van de robot totdat de vergrendeling klikt. Voor het uitschuiven van het batterijencompartiment moet men de ontgrendeling indrukken.



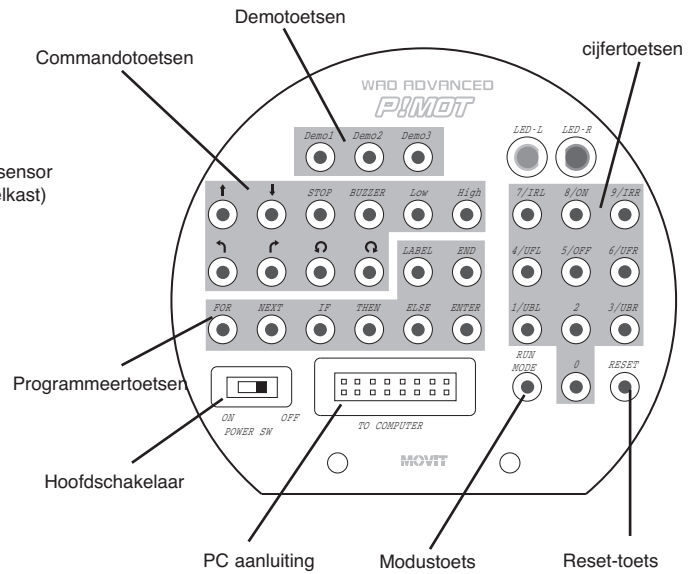
De complete P!MOT

6. Beschrijving van de P!MOT

6.1 Namen en functies van de onderdelen



[Toetsenbord op de achterkant]



6.2 Beschrijving van de toetsenfuncties

Het toetsenbord omvat 33 toetsenfuncties, 2 LED's, een PC-aansluiting en een hoofdschakelaar. Enkele toetsen bedienen een tweetal functies. De toets 7 betekent bijvoorbeeld "Licht infraroodsensor".

Demotoets

P!MOT is uitgerust met drie voorgeprogrammeerde demoprogramma's. De demotoets stuurt de demoprogramma's.

Commandotoets

Deze toetsen bedienen de functies, die op het toetsenbord staan. In de onderstaande regels worden ze individueel beschreven:

【↑】

Vooruit rijden. Normaal rijdt de robot dan met de gemiddelde snelheid. Daarna moet je nog de tijdsduur met de cijfertoetsen aangeven.

【↓】

Achteruit rijden. Normaal rijdt de robot dan met de gemiddelde snelheid. Daarna moet je nog de tijdsduur met de cijfertoetsen aangeven.

【↶】

Bocht naar links (Alleen het rechterwiel draait voorwaarts). Daarna moet je nog de tijdsduur met de cijfertoetsen aangeven.

【↷】

Bocht naar rechts (Alleen het linkerwiel draait voorwaarts). Daarna moet je nog de tijdsduur met de cijfertoetsen aangeven.

【Low】

Langzaam rijden voor alle navolgende commando's

“↑” “↓” “↶” “↷”

【High】

Snel rijden voor alle navolgende commando's

“↑” “↓” “↶” “↷”

【↻】

Op de plaats naar links draaien (Linkerwiel achteruit, rechterwiel vooruit). Daarna moet je nog de tijdsduur met de cijfertoetsen aangeven.

【↻】

Op de plaats naar rechts draaien (Linkerwiel vooruit, rechterwiel achteruit). Daarna moet je nog de tijdsduur met de cijfertoetsen aangeven.

【STOP】

Stoppen. Als P!MOT zich beweegt, zal de robot nu direct stoppen.

【BUZZER】

Toeteren. P!MOT kan vier soorten toetersignalen genereren. Na het commando “toeteren” kiest u een cijfertoets 1, 2, 3 of 4.

* De standaard tijdseenheid voor de commando's bedraagt 0,5 seconde als eenheid

“↑” “↓” “↶” “↷”

Kies daarna hoe vaak de P!MOT de toetersignalen herhalen moet. De toetersignalen worden in de navolgende tabel beschreven:

Cijfertoets	Toon
1	Pipi
2	Pipo
3	Popi
4	Pi--

Cijfertoets	Deze toets wordt toegepast om de tijdsduur en/of het aantal herhalingen vast te leggen.
Reset-toets	Met de resettoets stoppen we het programma van de robot. Bovendien schakelen we met deze toets de robot om in de programmeermodus, dat wil zeggen in de modus, waarin je programma's via het toetsenbord kunt invoeren.
Modustoets	Met de modustoets en de daarop volgende cijfertoets schakelen wij de robot in een andere modus. Naast de programmeermodus en de demomodus zijn nog andere modi gedefinieerd: zie bladzijde 28 voor details. Bovendien wordt de modustoets nog toegepast bij het programmeren in de programmeermodus.
Programmeertoets	Deze toets wordt gebruikt als we bijvoorbeeld een herhaling van een commando, een "IF"-conditie of de status van een "sensor"-waarde in een programma willen toepassen.
LED	Tijdens het actieve gebruik van het toetsenbord knippert de LED. Het knipperen signaleert een normaal programmeercommando. Als de bediening foutloos verloopt, licht de groene LED kort op en geeft het luidsprekertje een korte BEEP-toon weer. Bij fouten in de commando's, licht een rode LED kort op en geeft het luidsprekertje meerdere BEEP-tonen. Ook signaleren de LEDs, dat een van de sensoren wat vastgesteld heeft. Als de sensor aan de rechterkant iets ontdekt, licht de rode LED op. Als de sensor aan de linkerkant iets ontdekt, licht de groene LED op.
PC aansluiting	Via deze aansluiting kun je een optionele interfacekaart (zie pagina 54 voor details) en een PC als externe programmeerbron aansluiten.
Hoofdschakelaar	Met de hoofdschakelaar wordt de stroomvoorziening van de robot in respectievelijk uitgeschakeld.

* De toeter wordt ingeschakeld, als een van de batterijen voor het motorsysteem leeg is en als de hoofdschakelaar ingeschakeld wordt of de resettoets of de MODE(RUN)-toetsen ingedrukt worden.

6.3 We gaan de robot testen

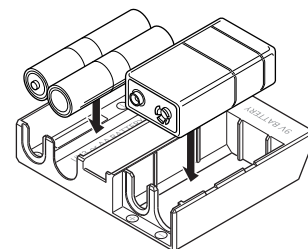
6.3.1 Het plaatsen van de batterijen

Leg twee AA "alkaline" batterijen en een alkaline blokbatterij (9V) klaar. (Deze batterijen behoren niet tot inhoud de bouwdoos).

* Gebruik s.v.p. nieuwe batterijen en neem geen verschillende batterijtypes.

- (1) Let op de juiste polariteit. Controleer de polariteit aan de hand van het plus- en min-teken.
- (2) Let vóór het plaatsen van de batterijen op, dat de hoofdschakelaar op "OFF" ("uit") staat en plaats daarna pas de batterijen. Zie ook pagina 24.

* Schuif het batterijcompartiment zover in de houder, tot het compartiment in de houder klikt.



6.3.2 Bedrijfstest

Nu kunnen we de robot met het voorgeprogrammeerde demonstratieprogramma testen.

Dat doen we met het eerste testprogramma DEMO 2: de "uitwijk-modus"

* Let er bij een testfase op een tafelblad op, dat de robot niet naar beneden kan vallen.

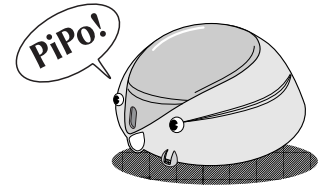
- (1) Zet de hoofdschakelaar op "**ON**" ("aan").
- (2) Druk nu de **RESET**-toets in. Controleer, dat zowel de groene als rode LED oplichten.
- (3) Druk vervolgens op de **MODUS**-toets. De robot antwoordt nu met "Pipo".
- (4) Druk daarna de "**Demo2**"-toets in. De robot antwoordt nu met "Pipo". Een seconde later zal de P!MOT beginnen te rijden.
- (5) De P!MOT rijdt voorwaarts, maar bij het contact met een obstakel zal de robot stoppen en de "Pipo"-tooncombinatie laten horen. De LED's zullen oplichten als het systeem een voorwerp waarneemt. De LED-L reageert bij een voorwerp aan de linkerkant. De LED-R reageert bij een voorwerp aan de rechterkant.
- (6) Als het obstakel wordt verwijderd zal de robot zich weer gaan rijden. Als er een voorwerp aan de linkerkant wordt ontdekt zal de LED-L oplichten, de "Pipo"-klank weerklinkt en de robot zal naar rechts draaien. Als er een voorwerp aan de rechterkant wordt ontdekt, zal de LED-R oplichten, de "Pipo"-klank weerklinken en de robot zal naar links draaien. Dit proces wordt herhaald, tot het voorwerp uit de het bereik van de robotsensoren verdwijnt.

Deze procedure volgt bij dezelfde werkwijze steeds hetzelfde patroon.

Schakel de P!MOT nu met een druk op de resettoets uit.

In de volgende test gaan we de programmering en de impulsensor controleren.

- (7) Druk op de **RESET**-toets. Controleer, dat zowel de groene als rode LED oplichten.
- (8) Druk op de **LOW**-toets. De robot antwoordt nu met "Pipo".
- (9) Druk de **<naar boven wijzende pijl>**-toets . De robot antwoordt nu met "Pipo".
- (10) Druk de **<1>**-toets. De robot antwoordt nu met "Pipo".
- (11) Druk de **<0>**-toets . De robot antwoordt nu met "Pipo".
- (12) Druk de **<ENTER>**-toets. De robot antwoordt nu met "Pipo".
- (13) Druk Sie **"MODE"**-toets, terwijl u de P!MOT in de hand vasthoudt.
- (14) De LED gaat nu uit, maar een seconde later zal het wiel beginnen te draaien.
- (15) Controleer nu of het wiel aan de rechterkant ook stopt, als je het linkerwiel in punt (14) met de hand tegenhoudt.



* In het tandwielsysteem bevindt zich geen sleepkoppeling die het rechter wiel aandrijft maar het wiel stopt wel met draaien als je het tegenhoud. Druk voorzichtig op het wiel en het draaien zal stoppen.

- (16) Controleer nu of het wiel aan de linkerkant ook stopt, als je het rechter wiel in punt (14) met de hand tegenhoudt.
- (17) Als een wiel tijdens deze testprocedure stopt, moet je nogmaals op de **"MODE"**-toets drukken en de procedure herhalen.

Met de afloop van deze procedure wordt de bedrijfstest afgesloten.
Schakel nu met behulp van de hoofdschakelaar de P!MOT uit.

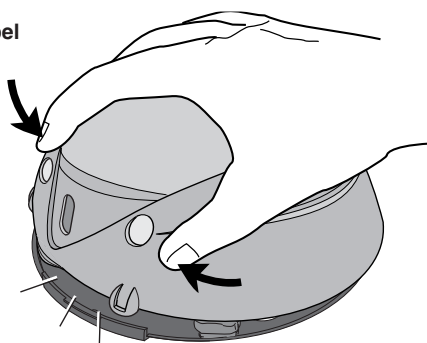
6.3.3. Foutzoeker

Probleem	Oplossing
LED-L en LED-R lichten niet op, ook dan niet als de RESET-toets wordt gedrukt.	<ul style="list-style-type: none"> • Is de hoofdschakelaar ingeschakeld? • Zijn de batterijen correct geplaatst? (Controleer de polariteit + en -) • Is de blokbatterij (9V) volgeladen?
P!MOT beweegt helemaal niet P!MOT beweegt zich af en toe wel en soms niet.	<ul style="list-style-type: none"> • Is de hoofdschakelaar ingeschakeld? • Zijn de batterijen correct geplaatst? (Controleer de polariteit + en -) • Zijn de batterijen volgeladen? • Maken de batterijcontacten contact met de robot • Veroorzaken de motoraansluitingen geen kortsluitingen? • Is het tandwielsysteem correct geassembleerd?
P!MOT ontwijkt geen obstakel.	<ul style="list-style-type: none"> • Is de blokbatterij (9V) volgeladen?
P!MOT ontwijkt een niet bestaand obstakel.	<ul style="list-style-type: none"> • Is de infrarode LED te veel op de bodem gericht? Als de LED wellicht de bodem detecteert, kunt je de infrarode LED's wat naar boven buigen. • Is de afschermkap op de infrarode LED's geplaatst?
P!MOT genereert geen "Pipo"-tonen.	<ul style="list-style-type: none"> • Is de Blokbatterij (9V) volgeladen? • Is de luidspreker correct aangesloten? • Is de witte stekker correct aangesloten?
Bij het testen van de impulsensoren stopt een wiel niet, als het andere wiel met de hand tegengehouden wordt.	<ul style="list-style-type: none"> • Is de blokbatterij (9V) volgeladen? • Is de impulsensor correct aan het tandwielsysteem aangesloten? • Heb je de MPL, MPR, LL en LR correct aangesloten ? • Heb je de bruine en zwarte stekkers correct aangesloten?

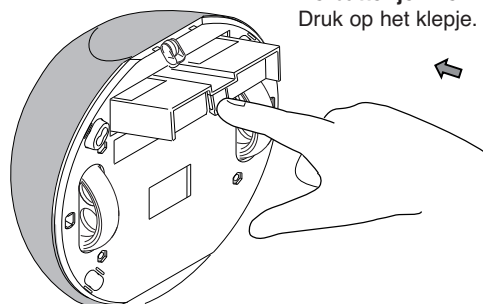
* Er zijn veel oorzaken mogelijk, die een normaal gebruik storing kunnen geven.
 Controleer s.v.p. de assemblage van het tandwielsysteem, de printplaat, het toetsenbord, enzovoorts.
Controleer alle stappen van de opbouw nogmaals.

Openen van de koepel

Druk op de koepel op de plek van de pijlen en til hem omhoog.



De batterijen verwijderen
Druk op het klepje.



6.4 Definitie van de modi.

P!MOT beschikt over zes modi:

1	Programmeermodus	In deze modus kun je programma's ingeven en starten.
2	Demo-modus	In deze modus worden voorbeeldprogramma's gestart.
3	Programma-otvangstmodus	In deze modus kun je programma's vanuit de PC naar de P!MOT sturen.
4	Slaapmodus	In deze modus kun je stroom besparen zodat de batterijen langer meegaan
5	Programma herladen-modus	Als een programma na het uitschakelen van de hoofdschelaar of na een reset verwijderd is kun je het met deze modus weer terughalen.
6	Instellen van de bewegingstijd	Met deze modus kun je de bewegingstijd (en daarmee de beweging) nauwkeurig instellen.

Zie bladzijde 25 voor de beschrijving van de toetsen.

De modus introductie

(1) Programmeermodus

(2) Demo-modus

Klik op "**Enter**" om in de programmeermodus te komen, nu zullen op de hoofdprintplaat vier rode en groene LED's oplichten. Dit betekent dat wij met het programmeren kunnen beginnen. De details van het programmeren staan beschreven vanaf pagina 33 in het hoofdstuk "Programmeren".

1. Druk de **Reset**-toets. Druk op Mode. Nu bevindt de P!MOT zich in de demo-modus. Klik vervolgens op "**Demo1**", "**Demo2**" of "**Demo3**". Daarop voert de robot het bijbehorende programma uit. De onderdelen van de demoprogramma's worden in de onderstaande tekstdelen beschreven.

"Demo1" (toevalspatroonprogramma)

Programmabeschrijving: na een tevoren gekozen tijdsinterval zal de robot zich plotseling met een BEEP-toon melden en een willekeurig patroon gaan rijden.

Programmeringscommando's: **Reset -> Mode -> Demo1 -> Tijdsduur** (hiervoor s.v.p. een getalwaarde ingeven)

Handleiding: De getalwaarde van de cijfers "0" tot "9" geven een intervaltijd van nul tot negen minuten aan.

Om te stoppen moet je op de **RESET**-toets drukken.

"Demo2" (Ontwijkprogramma)

Programmabeschrijving: Zodra de infraroodsensor een obstakel waarneemt, zal de robot het voorwerp uitwijken. Als er geen obstakel gevonden wordt rijdt de robot rechtdoor.

Programmeringscommando's: **Reset -> Mode -> Demo2**

Handleiding: Om te stoppen moet je op de **RESET**-toets drukken.

* P!MOT vindt soms geen obstakel, als het voorwerp te klein of donker is.

"Demo3" (Volgmodus)

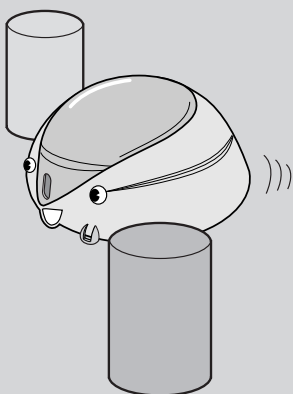
Programmabeschrijving: Houd de hand voor de P!MOT. De infraroodsensor zal nu het voorwerp waarnemen en volgen.

Programmeringscommando's **Reset -> Mode -> Demo3**

Handleiding: Om te stoppen moet je op de **RESET**-toets drukken.

Als de rechter- en linkersensor een voorwerp waarnemen, moet de P!MOT wachten. Als P!MOT bij het volgen van een hand het spoor kwijtraakt, zal de robot, door een pirouette te maken en de hand gaan zoeken.

* Als er andere voorwerpen in de zijn zal de P!MOT daarop reageren. Deze modus kunt je het best in een omgeving starten waarin zich niet te voorwerpen bevinden.

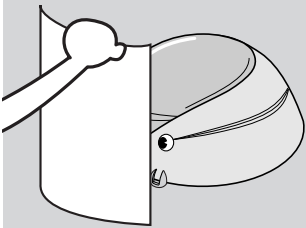


(3) Programma- ontvangstmodus	<p>Om de programma-ontvangstmodus te activeren moet je de toetsen Reset -> Mode -> 0 aanklikken. Vier rode en groene LED's op de hoofdprintplaat geven aan dat de robot klaar is om een programma van de PC naar de robot te uploaden. Voor de robotsoftware upload van de PC naar de robot heb je een speciale interfacekaart voor de P!MOT nodig en een PC met Windows inclusief een oud type centronics naar RS232 printerkabel.</p> <p>* De interfacesoftware voor de upload en voor het opslaan van P!MOT-programma's staat op onze webpagina ter beschikking. Zie pagina 54.</p> <p>* LET OP! moderne PC's hebben meestal geen standaard RS232 poort.</p>
(4) Slaapmodus	<p>Druk de toetsen Reset -> Mode -> 1 om de slaapmodus te activeren, waardoor P!MOT het stroomverbruik reduceert en de levensduur van de batterijen verlengt. Indien je de robot voor langere tijd niet gebruikt, moet je deze met de hoofdschakelaar uitschakelen.</p>
(5) Programma- Reloadmodus	<p>Druk de toetsen Reset -> Mode -> 8 in om de programma-reloadmodus te activeren. Het allerlaatste programma kan opnieuw worden opgeroepen en gestart. Omdat het allerlaatst toegepaste programma bewaard blijft, kun je dit zelfs na het uitwissen door middel van de RESET-toets of na het uitschakelen van de stroomvoorzorging weer herinstalleren. Als de modustoets direct voor het herinstalleren van het laatste programma wordt aangeklikt, start de robot het laatstgebruikte programma.</p>
(6) Vastleggen van de draaibewegingstijd	<p>Druk op de toetsen Reset -> Mode -> 9 om de draaibewegingstijdsduur vast te leggen. In deze modus registreert de robot de draaibewegingstijdsduur voor een complete cirkel ofwel een 360°-rotatie en slaat deze waarde in het geheugen op. Met deze waarde kan de robot de draaibewegingen nauwkeuriger uitvoeren.</p> <p>* Zie pagina 30 voor de details bij het kalibreren.</p> <p>* Als er andere voorwerpen in de buurt zijn zal de P!MOT daarop reageren en kan de kalibratie (ijking) niet nauwkeurig gedaan worden. Voor de ijking van de robot kun je het best een omgeving zonder stoorbronnen kiezen.</p> <p>* De meetfout zal bij een relatief groot ijkingsobject relatief groot zijn. Een ijkingsobject ter grootte van een A4-formaat blad papier is als doel goed geschikt. Een te klein ijkingsobject is niet voor beide sensoren tegelijkertijd zichtbaar.</p> <p>* Alhoewel deze ijkingprocedure de draaibewegingen nauwkeuriger moet maken zul je wel begrijpen, dat door de grootte van het ijkingsobject en de verschillen in de P!MOT-modules en motoren, enz. een exacte 360°-rotatie nauwelijks mogelijk is.</p>

6.5 Voorbereidingen voor het programmeren

<p>Voor het programmeren</p>	<p>Kun je niet wachten om de P!MOT direct met een eigen programma laten rondtoeren? Dan moet je nog even geduld hebben!</p> <p>Want alhoewel de P!MOT met de componenten rondom de microcomputer en software bestuurd wordt, is het gedrag van de robot ook nog gedeeltelijk afhankelijk van de toleranties van de onderdelen, zoals de motoren, tandwielsystemen, elektronica-componenten, enzovoort).</p> <p>Het kan dus gebeuren, dat een geprogrammeerde draaicirkel rechtsom 360° bestrijkt, maar de draaicirkel linksom slechts 300° omvat. Daarom moeten we eerst een ijkingprocedure doorlopen om dit soort fouten zo klein mogelijk te maken.</p>
-------------------------------------	--

Het kalibreren van de draaibewegingstijd



Vorbereidende werkzaamheden: neem een wit A4-blad papier

Om de draaibewegingsperiode te kalibreren maken we gebruik van een infraroodsensor. We doen dit op een schoon werkvlak dat ongeveer 50cm rodrom vrij van obstakels moet zijn. Als er zich voorwerpen in dat bereik bevinden kan de robot de ijkingsprocedure misschien niet correct doorvoeren.

De kalibratie procedure gaat als volgt:

- (1) Plaats de P!MOT op een vrije werkvlakte met een radius van ca. 50 cm.
- (2) Schakel de hoofdschakelaar aan.
- (3) Druk de "RESET"-toets.
- (4) Druk nu "MODE". De robot reageert met een "Pipo"-toon.
- (5) Druk nu de toets "9". De robot reageert met een "Pipo"-toon. LED-L en LED-R worden uitgeschakeld.
- (6) Het tevoren uitgezochte papier wordt op op 10 cm afstand van de P!MOT gehouden (zie de tekening aan de linkerkant). Dit blad papier moet tot het einde van de ijkingsprocedure in die positie worden gehouden.
- (7) LED-L en LED-R lichten kort op. Dan draait P!MOT naar rechts en reageert met een "Pipo"-klank.
- (8) Nadat de robot een complete 360° cirkel afgesloten heeft, lichten LED-L en LED-R kort op en de P!MOT blijft dan stil staan.
- (9) LED-L en LED-R lichten kort op en dan draait de P!MOT naar links.
- (10) De bewegingsprofielen voor de draaiingen links- en rechtsom zijn vrijwel gelijk. De draaiing wordt beëindigd met een oplichten van de LED's. Daarmee is de kalibratieprocedure afgesloten. U kunt nu het blad papier wegnemen en de robot testen.
- (11) Druk "RESET" dan --> --> -->
Nu druk je vervolgens --> -->
- (12) Druk nu de "MODE"-toets. P!MOT maakt nu een cirkel naar links. Let erop, dat de robot de cirkel slechts eenmaal mag doorlopen.
- (13)) Druk "RESET" dan --> --> -->
- (14) Druk nu de "MODE"-toets. P!MOT maakt nu een cirkel naar rechts. Let erop, dat de robot de cirkel slechts eenmaal mag doorlopen.

De kalibratieprocedure en de bijbehorende tests zijn nu afgesloten.

Schakel de robot met de hoofdschakelaar uit.

Misschien heb je al gemerkt, dat je voor een 90°-bocht in het programma een getal "4" moet invoeren. En bij de invoering van een getal "1" zal de robot een draaiing van ongeveer 22,5 graden maken.

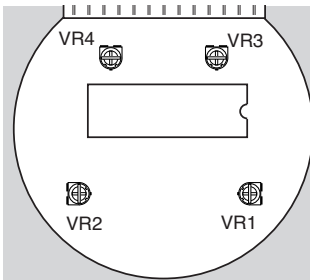
* Hoewel we de kalibratieprocedure met de sensoren en een blad papier geautomatiseerd doorvoeren, blijft een exacte draaibeweging onmogelijk.

Een kalibratieprocedure voor de bodemsensoren is overbodig, omdat deze al tijdens de fabricage worden uitgevoerd. De zogenaamde CdS-componenten zijn echter wel voor zijlicht gevoelig en soms is achteraf een correctie zinvol.

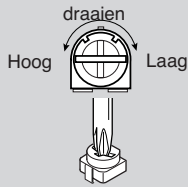
Deze correctie wordt in de volgende regels beschreven:

- (1) Schakel de hoofdschakelaar aan.
- (2) Druk de "RESET"-toets.
- (3) Druk achtereenvolgens:
- (4) Druk achtereenvolgens:
- (5) Druk achtereenvolgens:
- (6) Druk achtereenvolgens:
(
- (7) Druk achtereenvolgens:
- (8) Druk de "MODE"-toets en lees verder op badzijde 31

Kalibreren van de bodemsensoren.



Gevoeligheid



* Bij de volumeregeling voor de luidspreker moet u de schroeven draaier heel voorzichtig hanteren, totdat er een geringe weerstand optreedt. Bij teveel geweld wordt de potentiometer beschadigd.

Nu knippert of brand de onderste sensor.

Controleer de sensor aan de linkerkant (UFL) zodra dez sensor de bodem (of hand) registreert moet LED-L oplichten en zal de robot de "Pipo"-toon afspelen. Omgekeerd blijft LED-L uitgeschakeld als de robot geen bodem registreert. Om de kaibratie beter te volgen kun je het geluid harder zetten met potentiometer VR1.

Diezelfde procedure herhalen wij voor de overige sensoren UFR, UBL en UBR. Vervang daartoe op regel 5 in de kalibratieprocedure de toets definitie 4/UFL door de naam van de te ijken sensor.

* Lichtdetectoren zijn soms niet in staat een zwarte bodem te registreren, omdat de zwarte kleur het licht absorbeert. Omdat de P!MOT over extreem gevoelige bodemsensoren beschikt, kan de robot af en toe niet helemaal goed afgeregeld worden omdat een zwarte bodem als rand voor de vaak niet goed werkt. Ook een hele witte werkvakke als rand wordt soms moeilijk herkend.

<i>Bodemsensor</i>	<i>potentiometer</i>	<i>Bijbehorende LED</i>
UFL	VR1	LED-L
UBL	VR2	LED-L
UFR	VR3	LED-R
UBR	VR4	LED-R

6.6 FLOWCHARTS

Nu zou je kunnen starten met het programmeren van de P!MOT, maar..... wacht nog even.....

Bij het programmeren is haastwerk vaak een groot probleem. We beginnen daarom bij voorkeur met het opschrijven van onze programmeerdoelen in een flowchart (stroomdiagram).

Programma's en flowcharts.

Als we een programma schrijven, doen de vakkundige programmeurs dat meestal in een flowchart.

Waarom kiezen ze voor een dergelijke methode?



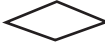


Het kost veel tijd en je kunt ook zonder flowchart programmeren.

Inderdaad we zijn sneller als we direct beginnen met het invoeren van de programmeercommando's.

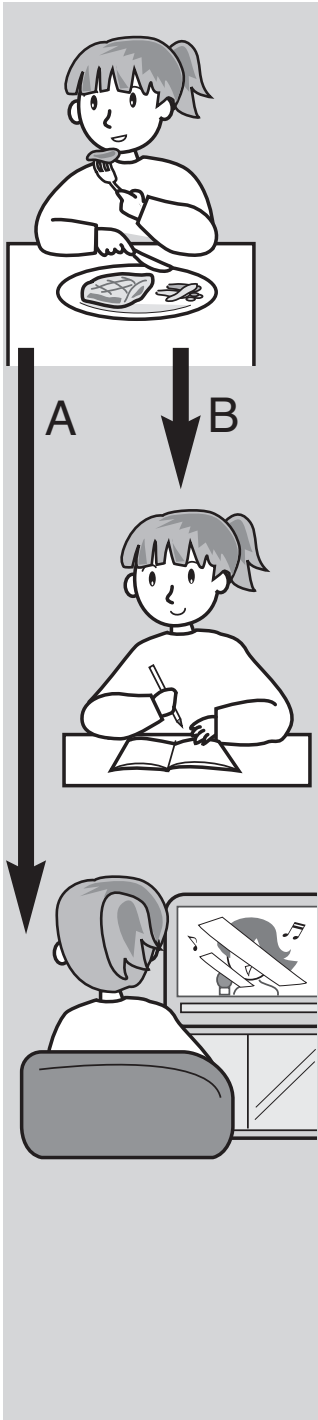
toch is een flowchart belangrijk. Zelfs de professionele programmeurs maken eerst een flowchart en beginnen dan pas met het programmeren. In het volgende hoofdstuk zullen we zien waarom.

Wat is een flowchart?

Een flowchart is een landkaart of schema voor de stappen die het programma moet volgen. Met behulp van dit schema kunnen we fouten lokaliseren en vinden. In de flowchart noteren wij onze gedachten en organiseren wij de deelopdrachten. In een flowchart gebruiken wij de onderstaande symbolen die we verderop in de documentatie ook weer terug zullen vinden:

-  ... Label: begin-/eindpunt van een programma
-  ... Proces: een werkvolgorde van opdrachten
-  ... Keuzeblok: kiest een pad uit diverse paden
-  ... Knooppunt: hier komen diverse paden bijeen
-  ... Verlooppijl: dit soort pijlen beschrijven de volgorde van een programma

Bij het maken van een flowchart plaatsen we deze symbolen in de correcte volgorde die het programma moet doorlopen.

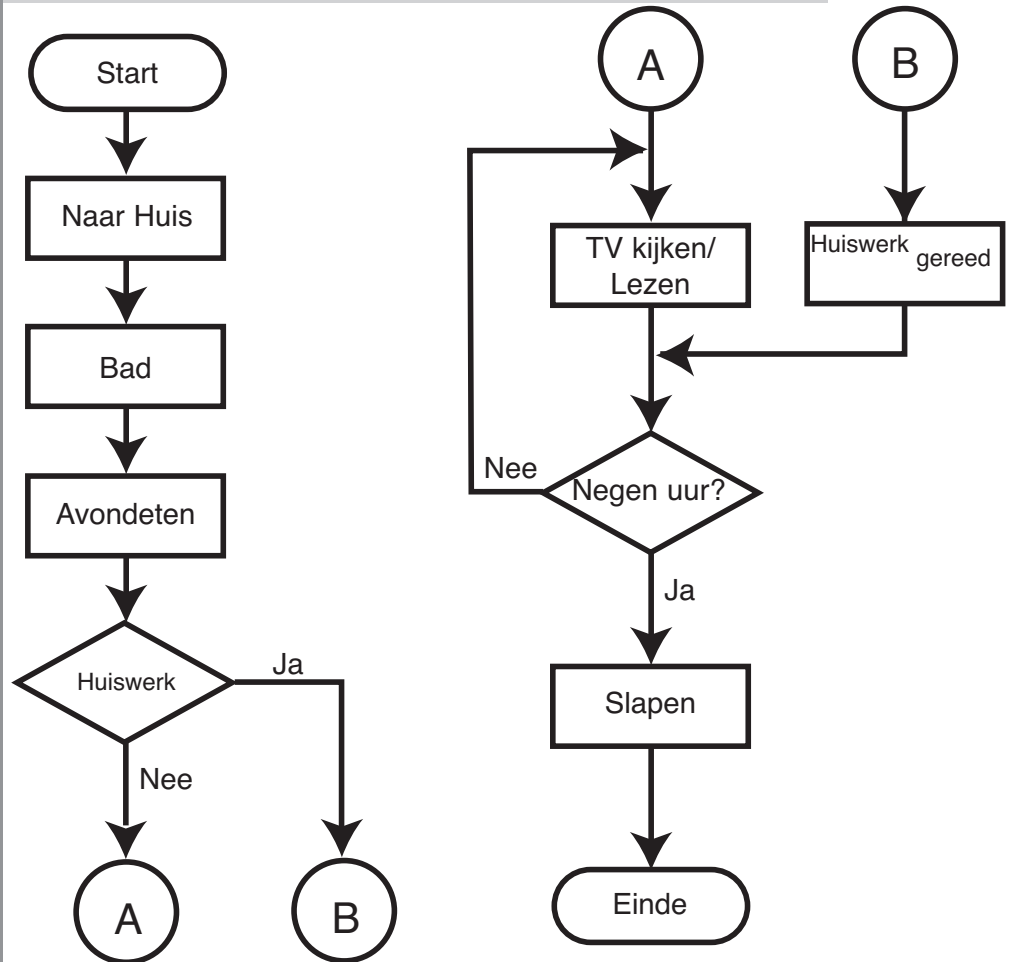


Als voorbeeld gaan we in een flowchart maken over Nancy en waar Nancy na de school haar tijd aan besteedt. Eerst beschrijven wij haar activiteiten in tekstvorm en daarna in een flowchart.

(1) Beschrijving van de dagindeling in tekstvorm:

Nancy gaat naar huis zodra zij thuis aankomt, gaat zij in bad en daarna aan tafel voor het avondeten. Na het avondeten moet ze nog vaak het huiswerk maken. Als ze geen huiswerk hoeft te maken, of deze al heel snel af heeft, volgt meestal was televisie kijken of lezen. Rond 9 uur gaat zij naar bed.

(2) Flowchart van dezelfde dagindeling in grafische vorm



Begrijpt je nu, waarom een flowchart zo belangrijk is?

Zoals je in het voorbeeld kunt aflezen, is een flowchart veel overzichtelijker dan de tekstvorm. Met de flowchart vermijdt je programmeerfouten en kun je het programma van begin af aan zonder ontwerpfouten opbouwen.

Een programmeerfout heet in de vaktaal "Bug#" en het verwijderen van programmeerfouten noemen we "Debuggen". In het ideale geval ontwerpt men foutloze software, maar als mensen maken we altijd wel eens een foutje. Met behulp van een flowchart zijn fouten eenvoudiger op te sporen en soms zelfs vermijdbaar.

Voordelen van een flowchart:

- * Je kunt de te stap voor stap processen in een programma goed identificeren
- * De flowchart geeft een goed totaal overzicht van het programma
- * Je kunt de afloop van het programma uitstekend controleren of de eventuele fouten in het programma gemakkelijker vinden en corrigeren.

Maak daarom voor het programmeren altijd eerst een flowchart.

Engelse woord voor "Kever" of "plaagdier"

7. Programmeren

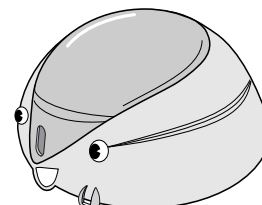
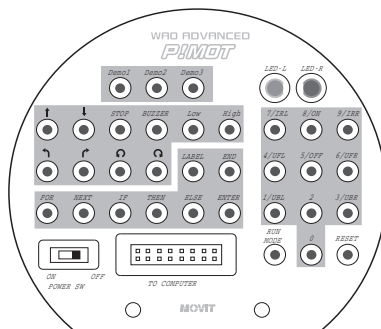
In dit hoofdstuk beschrijven we de programmering van de P!MOT-robot.

In de programmeermodus kunt je programma's schrijven en starten die uit reeksen commando's bestaan. Informatie wordt verkregen vanuit de sensoren en verder worden er motoren en LED's aangestuurd. Een programma bestaat uit commando's, herhalingen en variabelen.

Commando-toetsen

[↑] [↓] [↶]
 [↷] [Low] [High]
 [↻] [↺] [STOP]
 [BUZZER]

Deze toetsen zijn voor het invoeren van de commando's die boven de toetsen staan. De toetsen worden op bladzijde 25 uitvoerig beschreven:



Programmeertoets

FOR

Deze programmeertoets heb je bin de FOR – NEXT – herhalingen nodig.

Een FOR wordt steeds in combinatie met een NEXT toegepast. De herhaling wordt net zo vaak uitgevoerd zoals je de na FOR hebt ingesteld. Als je na FOR een nul specificeert wordt het proces onbegrensd lang herhaald (oneindig dus).

NEXT

Deze programmeertoets geldt als afsluiting voor de processen in de FOR – NEXT – herhalingen. Het programma herhaalt de opdracht tussen FOR en NEXT net zo vaak als je na FOR hebt vastgelegd. Daarna wordt het programma voortgezet met het eerstvolgende commando na NEXT.

IF

Deze functie is nodig voor de beslissing, die wij na het opvragen van een sensorenwaarde moeten nemen. Het is allemaal afhankelijk van de opgevraagde waarde, daarna moeten we besluiten, hoe het programma verdergaat. De IF-beslissing wordt als volgt gelezen: als de sensor bijvoorbeeld de waarde A bereikt, dan wordt het commando X. Als dat niet het geval is, dan moet het commando Y worden, of je gaat door tot het eerstvolgende commando. Dat wil zeggen: afhankelijk van de sensorenwaarde wordt commando X, Y of eventueel het eerstvolgende commando uitgevoerd. Op die manier kunnen we in een programma een aftakking definiëren.

IRL

Met de toets "7/IRL" na een "IF" wordt de linker infraroodsensor geactiveerd en bevestigd.

IRR

Met de toets "9/IRR" na een "IF" wordt de rechter infraroodsensor geactiveerd en bevestigd.

UFL

Met de toets "4/UFL" na een "IF" wordt de linker, voorste bodemsensor geactiveerd en bevestigd.

UFR

Met de toets "6/UFR" na een "IF" wordt de rechter, voorste bodemsensor geactiveerd en bevestigd.

UBL

Met de toets "1/UBL" na een "IF" wordt de linker, achterste bodemsensor geactiveerd en bevestigd.

UBR

Met de toets "3/UBR" na een "IF" wordt de rechter, achterste bodemsensor geactiveerd en bevestigd.

ON

De toets "8/ON" na "IF" en na de specificatie van een sensor schakelt de sensor bij een bevestiging op "INGESCHAKELD".

OFF

De toets "5/OFF" na "IF" en na de specificatie van een sensor schakelt de sensor bij een bevestiging op "UITGESCHAKELD".

THEN

Als de IF-voorwaarde waar is, springt het programma naar de regel met het labelnummer na het commando "THEN" en werkt van daaruit de commando's verder af. Bij de specificatie "THEN" -> "3" springt het programma naar de regel met labelnummer 3.

ELSE

Als de IF-voorwaarde niet waar is, springt het programma naar de regel met het labelnummer na "ELSE" en werkt van daaruit de commando's verder af. We kunnen de ELSE-specificatie ook weglaten, in dat geval werkt het programma van daaruit de eerstvolgende commando's verder af.

LABEL

Deze specificatie markeert het sprongadres van een programmaregel. Na het woord LABEL volgt een nummer, dat de programmaregel eenduidig markeert.

END

ENTER

Met dit commando wordt het programma beëindigd. Wij passen deze functie toe om een regel via het toetsenbord in te voeren. Daarna wordt de regel gecontroleerd. Als de regel in orde is, licht de groene LED op. Er wordt dan een Beep-toon weergegeven en we kunnen daarna de volgende regel invoeren. Bij een fout licht de rode LED op. Er wordt dan een "Pipo"-toon weergegeven en de foutieve regel wordt gewist. Daarna kunt je een correcte regel invoeren.

MODE
(RUN)

Met een klik op de toets MODE(RUN) kun je, na beëindiging van het invoeren van het programma, het programma starten. De P!MOT reageert dan direct. Bij problemen reageert de P!MOT met de "Pipo" toon en schakelt vervolgens over in de slaapmodus. In dat geval moet je de programmeertekst controleren en eventueel nieuw invoeren.

De formulering van een IF- structuur

Na de IF-toets leg je de naam van de sensor en de gewenste waarde vast.

Voorbeeld 1:

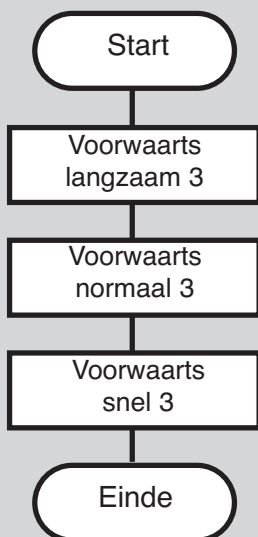
Uitgevoerde programmacode na label 1 als de rechter infraroodsensor ingeschakeld is
IF --> 9/IRR --> 8/ON --> THEN --> 1 --> ELSE --> 2

Voorbeeld 2:

"Als de linker achterste sensor uitgeschakeld is, moet het programma vanaf label 3 uitgevoerd worden, anders moet het programma op de volgende regel voortgezet worden" ziet er in programmacode als volgt uit: **IF --> 1/UBL --> 5/OFF --> THEN --> 3**

Oefenen door te programmeren

Oefening 1



Is de functie van alle toetsen duidelijk geworden? Wie het nog niet goed begrepen heeft, moet het, door te oefenen en uitproberen, goed leren beheersen. t volgende voorbeeld zullen we een P!MOT-programma aan een nader onderzoek onderwerpen.

Versnelling van LANGZAAM via NORMAAL naar SNEL.

Eerst beschrijven wij het programma in het hiernaast afgebeelde flowdiagram. Het programma ziet er zo uit:

[Low] [↑] [3]

[↑] [3]

[High] [3]

Het programma wordt als volgt in de P!MOT ingevoerd:

*** Druk op "RESET" om de programmeermodus te activeren.**

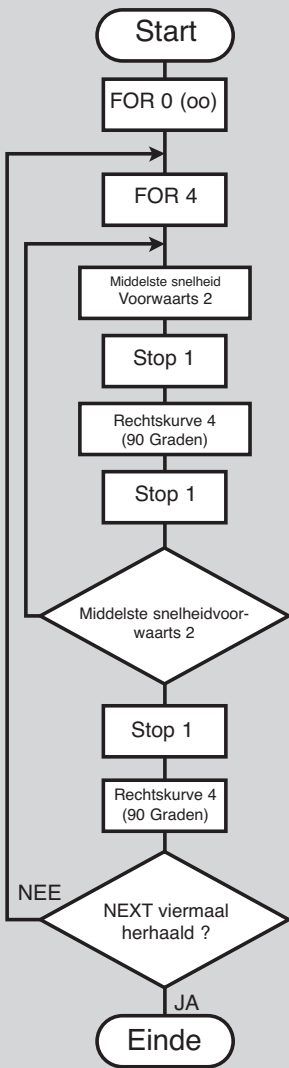
LOW -> ↑ -> 3 -> ENTER

↑ -> 3 -> ENTER

HIGH -> ↑ -> 3 -> ENTER

Daarmee is de programmering voor voorbeeld 1 afgesloten. Om dit programma te starten drukken we op de toets "**MODE(RUN)**", de robot start dan even later met de uitvoering van het programma.

Oefening 4



We kunnen dit programma met de FOR-NEXT-structuur ook binnen een andere FOR-NEXT-structuur oproepen. Deze programmastructuur noemen ze een meervoudige herhaling. In het nu volgende voorbeeld gaan we een eeuwigdurende herhaling programmeren die een meervoudige herhaling met kleine modificaties bevat:

* Druk eerst op "RESET" om de programmeermodus te activeren.

- | | | | |
|-----|------|---------|-------|
| 1. | ↑ | -> 2 -> | ENTER |
| 2. | STOP | -> 1 -> | ENTER |
| 3. | ↻ | -> 4 -> | ENTER |
| 4. | STOP | -> 1 -> | ENTER |
| 5. | ↑ | -> 2 -> | ENTER |
| 6. | STOP | -> 1 -> | ENTER |
| 7. | ↻ | -> 4 -> | ENTER |
| 8. | STOP | -> 1 -> | ENTER |
| 9. | ↑ | -> 2 -> | ENTER |
| 10. | STOP | -> 1 -> | ENTER |
| 11. | ↻ | -> 4 -> | ENTER |
| 12. | STOP | -> 1 -> | ENTER |
| 13. | ↑ | -> 2 -> | ENTER |
| 14. | STOP | -> 1 -> | ENTER |
| 15. | ↻ | -> 4 -> | ENTER |
| 16. | STOP | -> 1 -> | ENTER |

Nu is het kwadraatbochten-programma klaar.

Om het programma te starten druk je op de toets "MODE(RUN)".

Zoals je ziet wordt de FOR-NEXT-herhaling tweemaal toegepast. P!MOT kan dergelijke meervoudige herhalingen tot maximaal 30 maal in elkaar versleutelen. Nu begrijp je hoe de FOR-NEXT-structuur voor herhalingen wordt toegepast.

Als je deze methode nogal voorzichtig en argwanend benadert, kun je het herhalingsaantal eens variëren en een eigen voorbeeld ontwerpen. Na enige oefeningen zul je de FOR-NEXT-herhaling leren waarderen.

Toch kan het programma de robot, ook bij zorgvuldige programmering, niet helpen obstakels te ontwijken. Het programma maakt namelijk van geen enkele sensor gebruik.

Als we de robot als strategie kunnen leren: "Vermijd obstakels aan de linkerkant en rijd recht-door, als de weg vrij is.", zal de P!MOT niet meer tegen muren gaan botsen.

Robots worden uitgerust met sensoren, die het menselijke oog, onze handen, de huid, enz. vervangen. De robot neemt via zijn sensoren de omgeving waar. In het nu volgende hoofdstuk maken wij gebruik van de IF- conditie om met behulp van de sensorengegevens een beslissing te nemen.

We ontwerpen nu een programma, dat na een melding vanuit de linker infraroodsensor, een Pipo-geluid genereert. In de tekening hiernaast het bijhorende flowdiagram afgebeeld. De programmatekst luidt:

* Druk eerst op "RESET" om de programmeermodus te activeren.

- | | | | |
|----|--------|---|-------|
| 1. | IF | -> IRL -> ON -> THEN -> 1 -> ELSE -> 2 -> | ENTER |
| 2. | LABEL | -> 1 -> | ENTER |
| 3. | Buzzer | -> 2 -> 1 -> | ENTER |
| 4. | LABEL | -> 2 -> | ENTER |

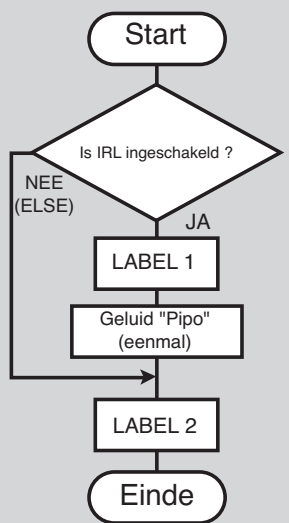
Om het programma te starten druk je op de toets "MODE(RUN)".

We starten nu het programma.

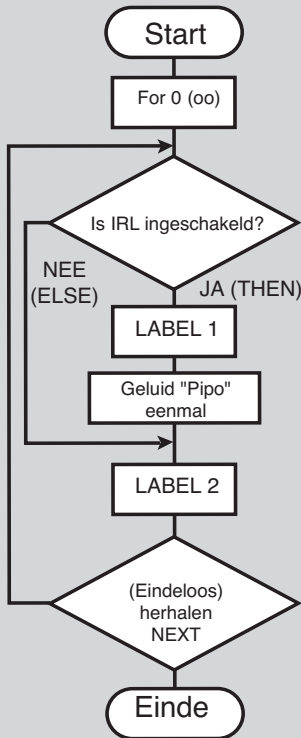
Reageert P!MOT nu inderdaad met "Pipo" ? Misschien is dat niet gebeurd. Een ontbrekend geluidspatroon wil nog niet zeggen, dat je een fout hebt gemaakt. De processor in de robot werkt buitengewoon snel en het programma was misschien al klaar.

Dat wil zeggen, dat de robot blijft zwijgen, ook als je na het drukken van de MODE(RUN)-toets je hand voor de betreffende sensoren hebt gehouden.

Oefening 5



Oefening 6



Als we met een FOR-NEXT-structuur een eindeloze herhaling opbouwen zal het programma permanent de door de sensor aangeboden gegevens in de gaten houden. Daardoor kunnen we uitstekend controleren of de sensor goed werkt. Deze methode is in de hiernaast afgebeelde flowchart geïllustreerd.

De programmatekst luidt als volgt:

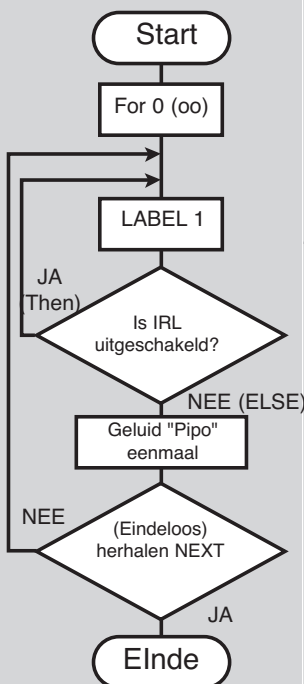
* Druk eerst op "RESET" om de programmeermodus te activeren.

1. FOR -> 0 -> ENTER
2. IF -> IRL -> ON -> THEN -> 1 -> ELSE -> 2 -> ENTER
3. LABEL -> 1 -> ENTER
4. **Buzzer** -> 2 -> 1 -> ENTER
5. LABEL -> 2 -> ENTER
6. NEXT -> ENTER

Om het programma te starten druk je op de toets "MODE(RUN)".

Dit programma wordt nu gestart om de werking van de sensor te controleren. De P!MOT moet een "Pipo"-geluid produceren als er een obstakel in de weg staat en moet zwijgen als er geen obstakel in de weg staat. Deze reacties bewijzen, dat P!MOT het sensorsignaal goed verwerkt.

Oefening 7

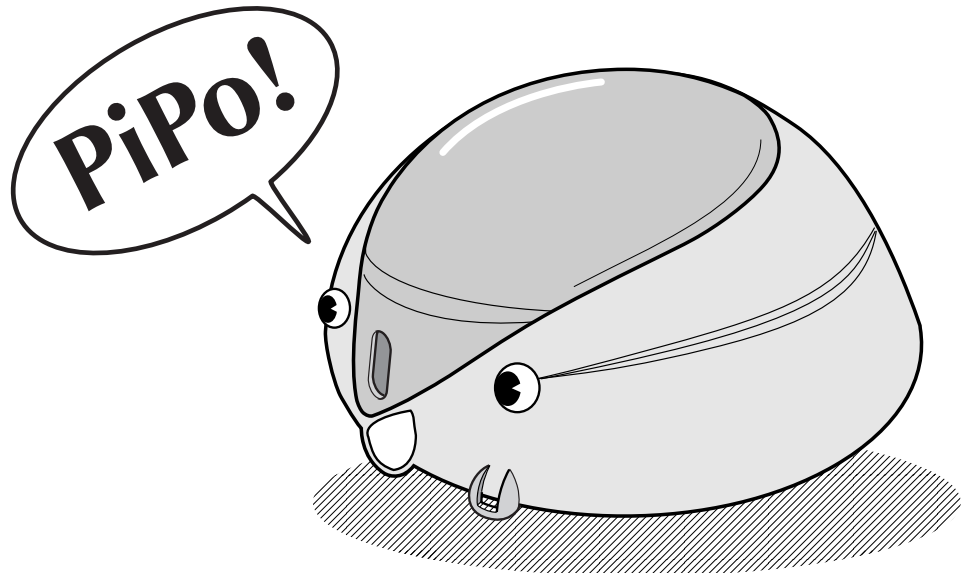


IF-condities vormen conditionele (ze moeten aan een bepaalde voorwaarde voldoen) zijpaden, die afhankelijk van de voorwaarde(n), de daarop volgende stap in het programma bepalen. Omdat we de sprongadressen ook mogen weglaten en daarmee automatisch de eerstvolgende stap in het programma uitgevoerd wordt, kunnen we het bovenstaande voorbeeld ook als volgt formuleren:

* Druk eerst op "RESET" om de programmeermodus te activeren.

1. FOR -> 0 -> ENTER
3. LABEL -> 1 -> ENTER
2. IF -> IRL -> OFF -> THEN -> 1 -> ENTER
4. **Buzzer** -> 2 -> 1 -> ENTER
6. NEXT -> ENTER

Om het programma te starten druk je op de toets "MODE(RUN)".



Afsluiting oefening 7

Heb je nu het programmeersysteem van de P!MOT begrepen?

Alhoewel wij tot nu toe eenvoudige programmaatjes als voorbeelden hebben toegepast, kan de P!MOT ook veel complexere functies uitvoeren.

Wij zullen nu met dezelfde methoden een complexe opgave beschrijven:

* Ontwerp een programma, waarmee de P!MOT met de infraroodsensoren verhindert dat de robot over de tafelrand naar beneden kan vallen. Bovendien moet de robot als er niets gebeurt, na een korte tijd ophouden met bewegen. Als P!MOT in die periode niets ontdekt, moet deze een "Pipo"-geluid produceren.

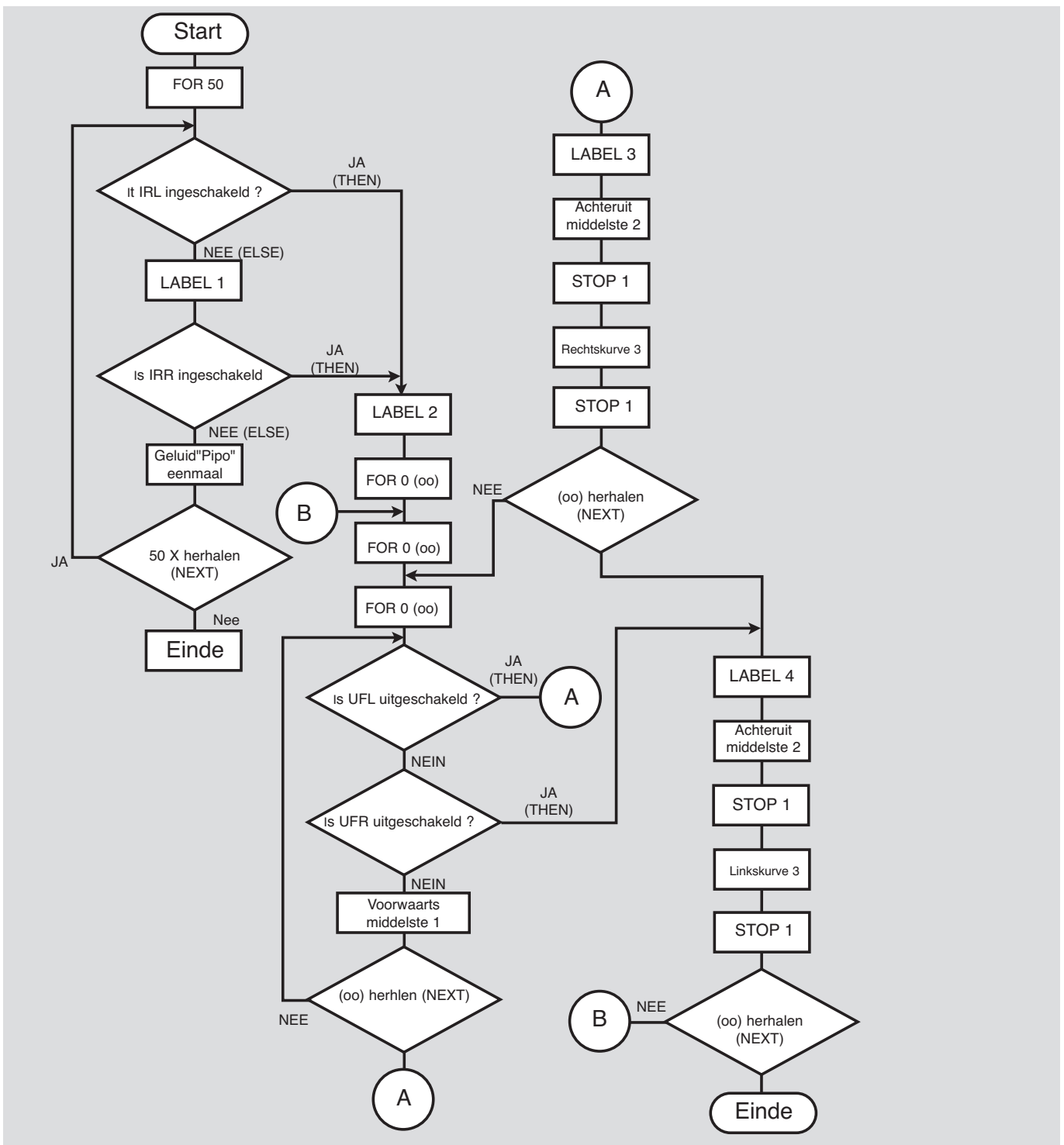
* De herhalingstijd voor het bemonsteren van de infraroodsensor wordt op 50 gezet.

* Als de linker, voorste bodemsensor geactiveerd wordt, moeten 2 bochten achteruit en vervolgens 3 bochten naar rechts uitgevoerd worden.

* Als de rechter, voorste bodemsensor geactiveerd wordt, moeten 2 bochten achteruit en vervolgens 3 bochten naar links uitgevoerd worden.

* Let op, dat de robot permanent het risico loopt van de tafel te vallen. Dit risico is onder meer afhankelijk van de toleranties in de tijdsperioden en lichtomstandigheden.

De flowchart ziet er als volgt uit:



Oefening 8

De programmatekst luidt als volgt :

*** Druk eerst op "RESET" om de programmeermodus te activeren.**

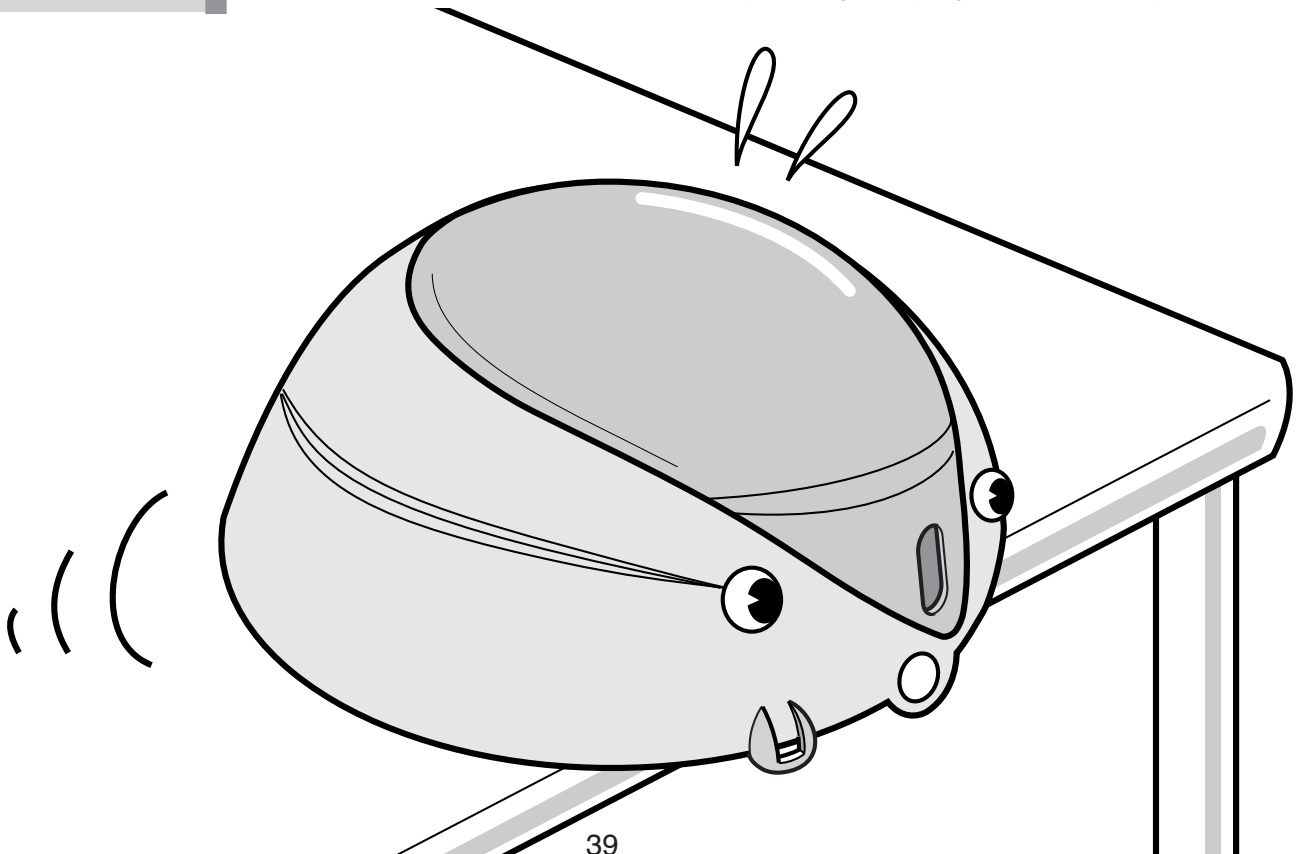
```
1.   FOR   -> 5  -> 0  -> ENTER
2.   IF    -> IRL -> ON  -> THEN -> 2 -> ELSE -> 1 -> ENTER
3.   LABEL -> 1  -> ENTER
4.   IF    -> IRR -> ON  -> THEN -> 2 -> ENTER
5.   Buzzer -> 2 -> 1 -> ENTER
6.   NEXT  -> ENTER
7.   END   -> ENTER
8.   LABEL -> 2  -> ENTER
9.   FOR   -> 0  -> ENTER
10.  FOR   -> 0  -> ENTER
11.  FOR   -> 0  -> ENTER
12.  IF    -> UFL -> OFF -> THEN -> 3 -> ENTER
13.  IF    -> UFR -> OFF -> THEN -> 4 -> ENTER
14.  ↑     -> 1  -> ENTER
15.  NEXT  -> ENTER
16.  LABEL -> 3  -> ENTER
17.  ↓     -> 2  -> ENTER
18.  STOP  -> 1  -> ENTER
19.  [⏸]   -> 1  -> ENTER
22.  LABEL -> 4  -> ENTER
23.  ↓     -> 2  -> ENTER
24.  STOP  -> 1  -> ENTER
25.  [⏸]   -> 1  -> ENTER
26.  STOP  -> 1  -> ENTER
27.  NEXT  -> ENTER
```

Om het programma te starten druk je op de toets "MODE(RUN)".

* Let op, dat de robot permanent het risico loopt van een tafel te vallen als je daarop werkt. Dit risico is onder meer afhankelijk van de toleranties in de tijdsperiodes en lichtomstandigheden.

Door de combinatie van FOR, NEXT en IF-regels kun je zeer complexe opgaven programmeren. Het ziet er misschien moeilijk uit, maar het is in de P!MOT eenvoudig programmeerbaar. Als je de benodigde commando's stap voor stap analyseert, wordt het programmeren steeds eenvoudiger.

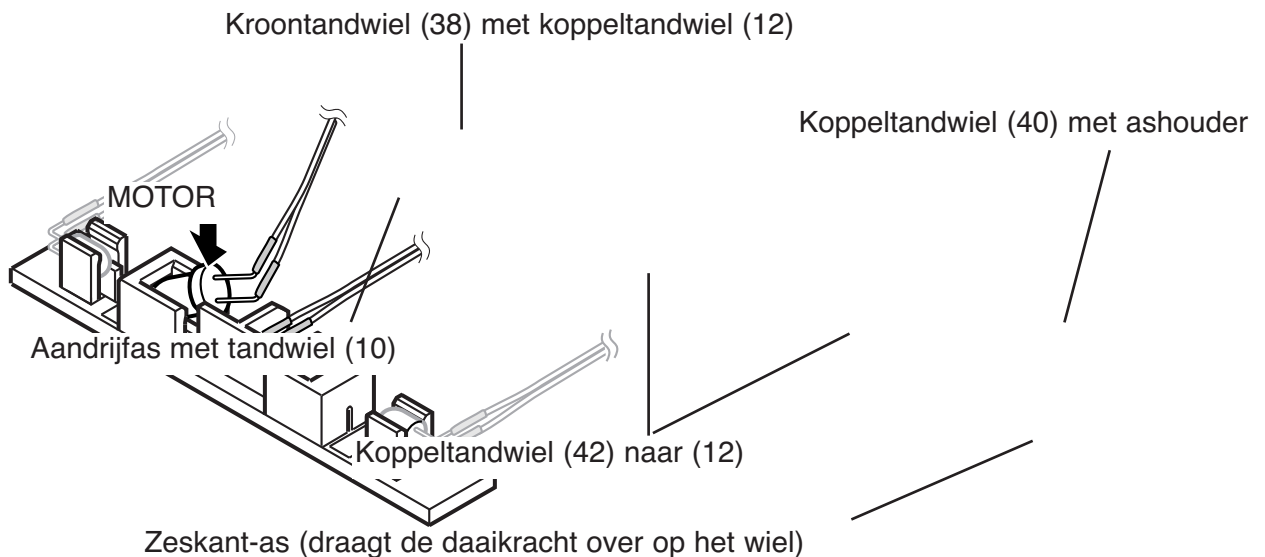
OK! Laten we nu voor de lol eens zelf een paar originele programma's ontwerpen.



8. Technische Details

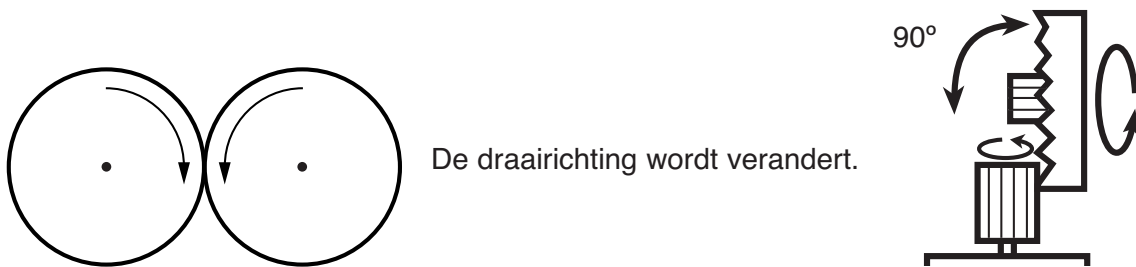
A. Overzicht van de mechanische structuur

De P!MOT is uitgerust met een tandwielsysteem dat het draaivermogen van de motor door middel van tandwieloverbrenging omzet in een bandenaandrijving. We zijn ervan overtuigd, dat je al eerder met tandwielmechanismen in contact bent gekomen. Tandwielen vind je in allerlei apparatuur, een tandwieloverbrenging wordt echter niet alleen voor het omzetten van en voor draaikracht toegepast, zoals je hieronder zult zien.



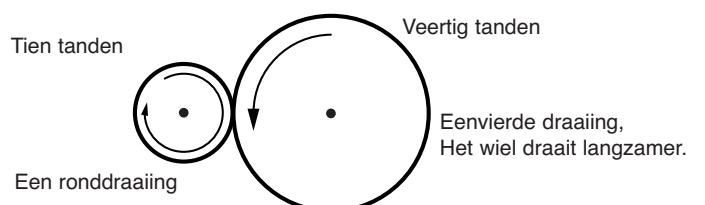
* Veranderen van de draaisnelheid

Het veranderen van de draaisnelheid is een gevolg van het aantal tanden dat het tandwiel heeft. Als voorbeeld nemen we een tandwiel met 10 tanden en een tandwiel met 40 tanden. Als het eerste tandwiel een volledige draaiing heeft gemaakt, dan heeft het tweede tandwiel nog maar 1/4 draaiing gemaakt. Dus om het tweede tandwiel een volledige draaiing te laten maken, moet het eerste tandwiel



* Veranderen van de draairichting

Met het veranderen van de draairichting wordt bedoeld, dat als het eerste tandwiel met de wijzers van de klok meedraait, het daaropvolgende tandwiel tegen de wijzers van de klok indraait. Een tandwiel verandert dus de draairichting.



*** HET OVERBRENGEN VAN DRAAIKRACHT**

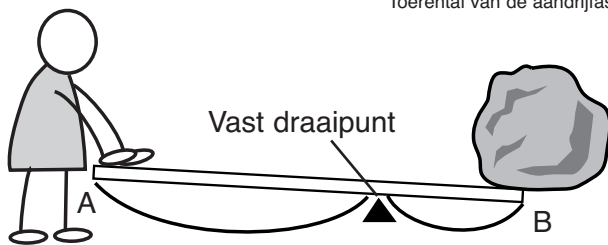
De draaikracht is ook van belang voor de draaisnelheid. Hoe lager de draaisnelheid wordt, hoe groter de draaikracht wordt en omgekeerd. Door deze tandwieleigenschappen wordt de kracht overgebracht naar de aandrijfas.

Het veranderen van de draaikracht kunnen we vergelijken met het draaien van een slaghout. Stel je voor dat twee mensen die elk een uiteinde van het slaghout vasthouden, proberen om het slaghout ten opzichte van zichzelf één bepaalde kant uit te draaien (bijv. naar rechts). De persoon die het handvat vasthoudt moet veel meer kracht gebruiken dan de persoon die het brede deel vasthoudt.

Hetzelfde geldt voor tandwielen. De kracht aan de rand van de tandwielen wordt groter als het dichterbij het centrum van het tandwiel komt. Aangezien de motor van de P!MOT niet de kracht heeft om de benen van de robot te laten bewegen, worden hiervoor tandwielen ingezet.

Met een hefboom

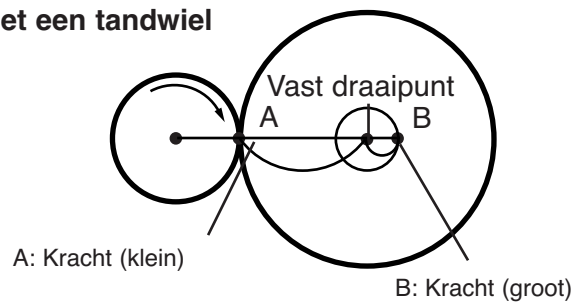
$$\text{Omzetting} = \frac{\text{Motortoerental}}{\text{Toerental van de aandrijfas}}$$



A: Als in een punt op (A), grote afstand van het vaste draaipunt, een kleine kracht wordt uitgeoefend.

B: Krijg je op punt (B) een veel groter hefvermogen en kun je een enorm groot gewicht eenvoudig optillen.

Met een tandwiel



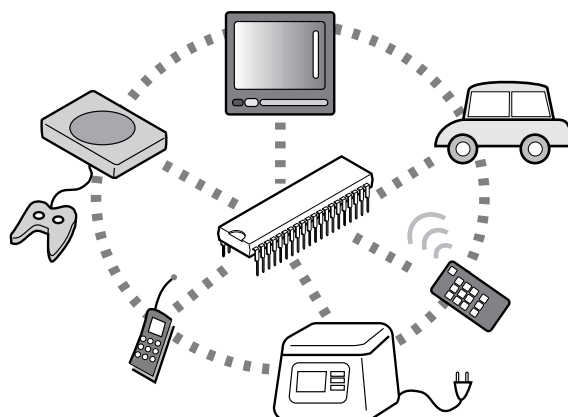
	Tandwieloverbrenging (Klein)	Tandwieloverbrenging (groot)
Draaisnelheid	Snel	Langzaam
Koppel (kracht)	Klein	Groot

B. Wat is een computer?

De P!MOT beschikt dus over een processor en wordt daarmee ook bestuurd. Tegenwoordig wordt steeds moeilijker een apparaat te vinden dat zonder computer werkt.

Zo bevatten bijvoorbeeld een telefoon, magnetron, elektrische oven, camera, televisie, tablet, game-console, PC, electrisch speelgoed en ook de auto allemaal computers.

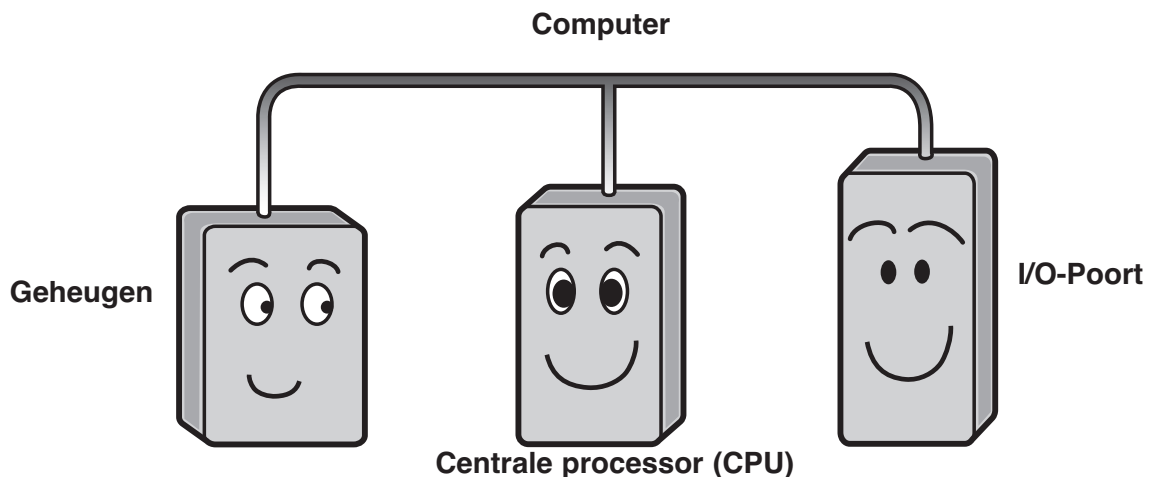
Weet je echter ook, hoe die computers werken? In het volgende hoofdstuk zullen we de rol van de computers in een groot aantal elektrische producten aan de hand van de P!MOT-computer beschrijven.



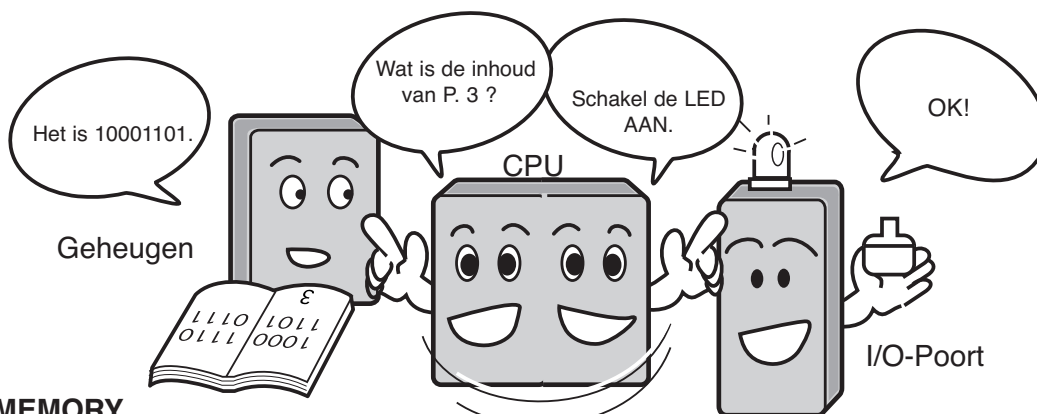
* De opbouw van een computer

De drie fundamentele onderdelen van een computer zijn:

- CPU De centrale processor of CPU (Central Processing Unit), voor het uitvoeren van opdrachten en de controle.
- Geheugen De opslagplaats voor programma's en gegevens
- I/O-Poort De communicatiewegen (O=Output) voor resultaten en het invoerinterface (I=Input) voor de buitenwereld



Hoewel computers sterk kunnen verschillen, beschikken zij allemaal over alle bovenstaande onderdelen. De centrale processor (CPU) bestuurt, met behulp van de gegevens uit het geheugen, de I/O-communicatie. Daarnaast voert het berekeningen uit, met behulp van een in het geheugen opgeslagen programma, en beslist aan de hand van resultaten wat er nu moet gebeuren.



MEMORY

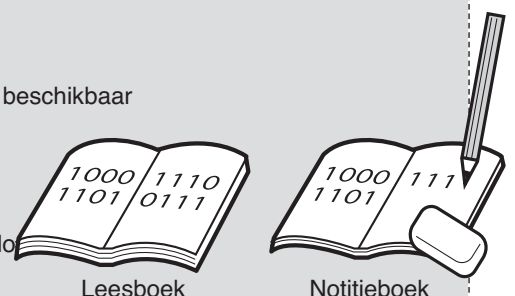
Voor de geheugenfuncties (Memory) zijn diverse opslagmethodes in gebruik. Wij zullen die methodes later nog leren kennen. Geheugens worden in twee soorten gecategoriseerd. De ene methode heet ROM, d.w.z. Read Only Memory en de andere RAM, d.w.z. Random Access Memory. De verschillen illustreren wij met als voorbeeld een leesboek en een notitieboekje.

Leesboek (vergelijkbaar met een ROM-geheugen)

- * wordt na het drukken alleen maar voor het lezen gebruikt
- * de opgeslagen informatie blijft ook na het afschakelen van de stroomvoorzorging beschikbaar
- * wordt hoofdzakelijk voor het besturingssysteem van de CPU gebruikt.

Notitieboek (vergelijkbaar met een RAM-geheugen)

- * kan beschreven en gelezen
- * de opgeslagen informatie gaat bij het uitschakelen van de stroomvoorzorging verloren
- * wordt voor het verwerken van gegevens in de CPU als kladpapier gebruikt.



* Computer en programma

Alhoewel vrijwel alle computers op een gelijk basisprincipe berusten kunnen ze in apparatuur toch een grote diversiteit aan opgaven afwerken. Hoe is dat mogelijk?

Hoewel een computer de meest uiteenlopende opgaven kan uitvoeren doen ze dat niet uit zichzelf. Daarvoor hebben ze een programma nodig.

De processor in een afstandsbediening kan bijvoorbeeld zeggen: "Goed, je bent nu eenmaal een afstandsbediening. Als iemand de kanaaltoetsen drukt, verander je het ontvangstkanaal en als iemand de hoofdschakelaar drukt, schakel je het toestel aan of uit."

De computer in de afstandsbediening reageert precies volgens het voorgeschreven programma.

* Programma en programmeertaal

Je ziet dat computers een voorgeschreven programma afwerken? Wat is een programma? In het woordenboek staat onder het begrip "programma": een plan, een voorschrift, enz.

Iedereen kent het programma van een sportfestijn of concert, dat we aan het begin van de activiteiten ontvangen. In de computertechnologie vervult het programma een vergelijkbare rol, dat wil zeggen als een activiteitenlijst. Wij zullen nu een programma in onze eigen taal schrijven, waarmee een televisieprogramma omgeschakeld wordt op een ander kanaal.

Als wij het televisieprogramma willen omschakelen, zeggen we het televisietoestel niet gewoon "Verander het programma!" je weet al dat de televisie dat commando niet begrijpt.

De taal van de televisietoestellen verschilt nogal van de mensentaal.

Om een computer te programmeren moet je de taal beheersen. Wat voor taal wordt er in een computer eigenlijk toegepast? Het klinkt wel merkwaardig, maar een computer werkt alleen met twee symbolen, en wel een "1" en een "0". De "een" signaleert, dat er een stroom vloeit en een "nul", dat er geen stroom vloeit. Het is deze taal, bestaande uit nullen en enen, die je moet beheersen om met een computer te communiceren. We noemen deze taal de computertaal en demonstreren een voorbeeld van een programma, waarmee het televisieprogramma op het kanaal Nederland 2 wordt omgeschakeld.

Mensentaal:	Schakel naar Nederland twee
Computertaal:	1110001101000101000101000100100100100100010011100....

Voor computers is dat zo klaar als een klontje. Wij mensen kunnen met deze schrijfwijze niet zo goed overweg en maken daarbij fouten. Ook als het slechts een enkel klein foutje betreft, kan het bijvoorbeeld betekenen, dat de televisie uitgeschakeld of de geluidsterkte plotseling enorm hard wordt. Altijd geldt echter, dat de computer aan deze programmeerfout onschuldig is. De computer blijft een trouwe knecht, die de hem opgedragen bevelen netjes en precies uitvoert.

De computer kan niet beoordelen, of iets fout of juist is. De programmeur is verantwoordelijk voor de foutieve respectievelijk correcte toepassing van de computer.

Om nu het grootste deel van de standaardfouten te voorkomen, passen wij een derde, goed leesbare tussentaal toe, die de menselijke taal met behulp van een "omzetter" in de computertaal omzet.

Mensentaal:	Schakel naar Nederland 2
Tussentaal:	IF 2ch BUTTON = ON THEN CHANGE 2ch OUT
Computertaal:	1110001101000101000101000100100100100100010011100....

Waarom is die tussentaal nodig? Er zijn talloze methodes om menselijke commando's te beschrijven en het is moeilijk die allemaal in de computertaal te vertalen. Daarom gebruiken wij een goed leesbare taal, die toch goed in de computertaal kan worden omgezet.

Deze talen noemen wij hoge programmeertalen. Voorbeelden van hoge programmeertalen zijn "Arduino" en "C".

Hoe is dat nu in de P!MOT opgelost?

In de P!MOT passen wij precies dezelfde methode toe.

Als wij nu eerst eens de interne P!MOT computertaal nemen, dan kunnen wij de computercode van de robot aan een nader onderzoek onderwerpen, bijvoorbeeld:

Als wij de "Langzame versnelling recht vooruit" kiezen, lezen wij de code: 11110000.

Als wij de "Bocht naar links" kiezen,lezen wij de code: 11110110.

Als wij het commando "Stop" kiezen, lezen wij de code: 00000000.

Wij hoeven echter niet in de symbolen "1" en "0" te programmeren. Voor het P!MOT- programma kiezen wij een gemakkelijkere methode, die in een voorbeeld wordt gedemonstreerd:

Mensentaal:	Langzame snelheid 5 rechuit, dan stoppen en daarna Beep 2.						
Tussentaal:	^	5	STOP	1	BUZZ	2	1
Computertaal:	11110100	10100101	00000000	10100001	01100010		

Zoals je ziet, vindt de vertaling in de processor plaats.

Is het niet vreemd, dat computers en programma's met nullen en enen zoveel kunnen presteren?

Nu zullen we ons eerst eens met de P!MOT bezighouden en een eerste programma invoeren.

Programmeertaal

Er bestaat een groot aantal programmeertalen en elke taal beschikt over bijzondere eigenschappen, zoals:

Assembler	Machinetaal, die in een voor mensen goed leesbare vorm wordt gecodeerd.
Arduino	Computertaal voor beginners
COBOL	Computertaal voor de boekhouding
FORTRAN	Computertaal voor de technici
C	Computertaal voor snelle processen

P!MOT Microcomputer

P!MOT is uitgerust met een microprocessor met een ingebouwd Flash-geheugen van het type "AT90S8515" uit de AVR- processorfamilie van de leverancier ATMEL.

In de navolgende tabel geven wij een overzicht van de eigenschappen van de AVR Microprocessor "AT90S8515":

Geheugencapaciteit

Flash ROM	8	kByte
EEPROM	512	Bytes
RAM	512	Bytes
Uitvoeringstijd	120	
I/O	32	
UART	1	
8 Bit Timer	1	
16 Bit Timer	1	
PWM	2	
Analoge comparator	geïntegreerd	
Flash ROM schrijfcyclussen	1000 (gegarandeerd)	

De actuele specificatie van de "AT90S8515" staat beschikbaar in de hieronder aangegeven web-site van ATMEL. De bijbehorende ontwikkelingsomgeving "AVR STUDIO" met de Assembler voor de AVR microprocessoren, simulatoren, enz. staan op deze wegpagina eveneens ter beschikking. Met deze werktuigen kunt u eigen software voor deze processorfamilie ontwikkelen. U kunt met die werktuigen een P!MOT bouwen, die er precies hetzelfde uitziet, maar over totaal andere geprogrammeerde functies beschikt, door de "AT90S8515" processor door een andere processor met geheel nieuwe software uit te wisselen.

ATMEL Website: <http://www.atmel.com>

*1. Als je de processor "AT90S8515" in de P!MOT uitwisselt, moet je er wel op letten, dat je de pinnen niet ombuigt.

*2. Bij het werken aan elektronische onderdelen (zoals de microprocessoren) moet je erop letten, dat deze zeer gevoelig zijn voor elektrostatische ontladingen. Wij kunnen helaas geen garantie geven voor schade, die bij het uitbouwen van de processor door elektrostatische ontladingen veroorzaakt wordt.

*3. Wij kunnen geen informatie leveren of vragen beantwoorden, die op de voorbeeldsoftware, de AVR-processor "AT90S8515", programmeertechnieken, enz. van de P!MOT-robot betrekking hebben.

* ATMEL en AVR zijn geregistreerde handelsnamen van de Atmel

C. Het motorsysteem

P!MOT beweegt zich heen en weer met behulp van een elektromotor. Daartoe is de processor voorzien van de volgende commando's:

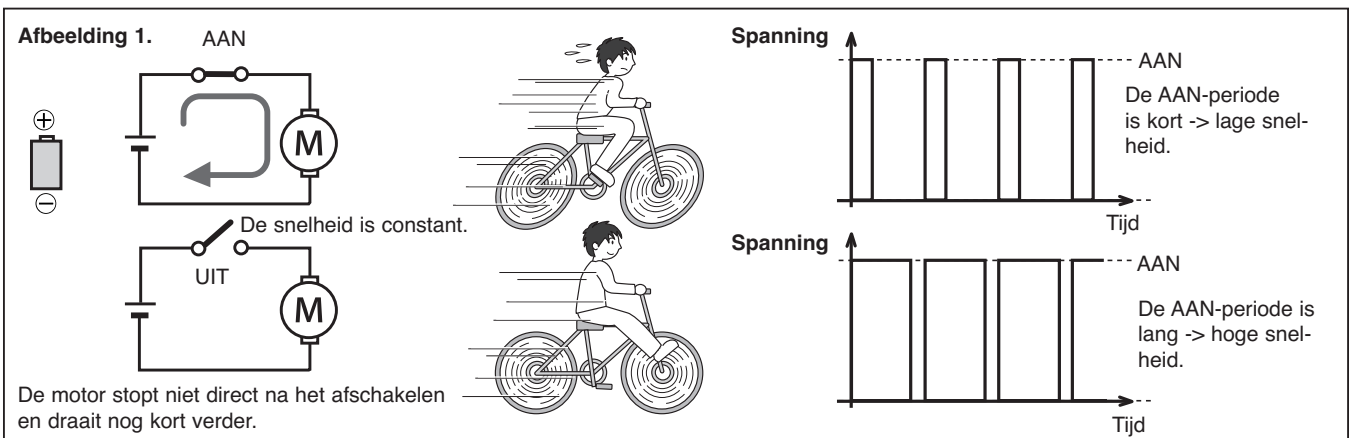
De snelheid is instelbaar: "langzaam", "gemiddelde snelheid", "snel".

We zullen nu beschrijven hoe deze commando's in P!MOT worden toegepast

1. De snelheidsregeling voor de motor

Het standaardprincipe voor de motor voor een motorsturing wordt in figuur 1 geschetst. In deze schakeling is het toerental voor de motor echter constant. Wat zal er gebeuren, als we de schakelaar uitschakelen bij een motor, die met een constant toerental draait?

Je weet, dat de motoraandrijving uitgeschakeld wordt. Denk je dat de motor direct tot stilstand komt? Dat is niet het geval, hoewel de miniatuurmotor in de P!MOT schijnbaar direct stilstaat, loopt het toerental langzaam terug en duurt het een tijdje tot de as werkelijk stilstaat. Dat ligt aan de massa-traagheid die de motor, ook na het uitschakelen van de stroomtoevoer, door laat lopen. Ook bij het fietsen kunnen we stoppen met trappen. De fiets rijdt dan door maar gaat dan wel steeds langzamer. Ook als wij een ventilator uitschakelen draait die nog een hele tijd door. De benodigde bewegings-energie wordt daarbij geleverd door de massa-traagheid. Als we die massa-traagheid eens nuttig gaan toepassen dan kunnen we de motorschakelaar steeds weer opnieuw aan- en uitschakelen en dan toch een constante snelheid aanhouden. Als we de uitgeschakelde periode verlengen neemt het toerental af. In de P!MOT wordt de snelheid gestuurd door een nauwkeurig ingestelde verschuiving van de impulsbreedten voor de AAN- respectievelijk UIT-fasen. Daarom heet deze regelingsmethode ook Impuls Breedte Modulatie (Engels: PWM = Pulse Width Modulation).



P!MOT werkt dus met een impulsbreedtemodulatie, maar wie schakelt nu eigenlijk de motor aan en uit? Misschien heb je al gezien, dat de processor de motor aan- en uitschakelt. Het daartoe ontworpen commando luidt: "Lever de impulsvorm voor lage snelheid". De processor genereert dan de passende impulsvorm voor de AAN-/UIT-schakelaar van de motor voor geringe, gemiddelde, respectievelijk hoge snelheid.

De P!MOT maakt als schakelaar gebruik van een FET. Deze schakelaar geleidt de stroom zodra de processor de FET een "H"-signaal ("H" = "hoge" spanning) aanbiedt. De impulsbreedten voor de AAN-, respectievelijk UIT-perioden luiden:

Snelheid	AAN	:	UIT
Lage snelheid	5	:	5
Gemiddelde snelheid	7	:	3
Hoge snelheid	9	:	2

Omdat de schakelaar alleen wordt ingeschakeld, als de motor energie nodig heeft, blijft het stroomverbruik gering. Bij gelijkstroommotoren kunnen we zo de snelheid goed regelen.

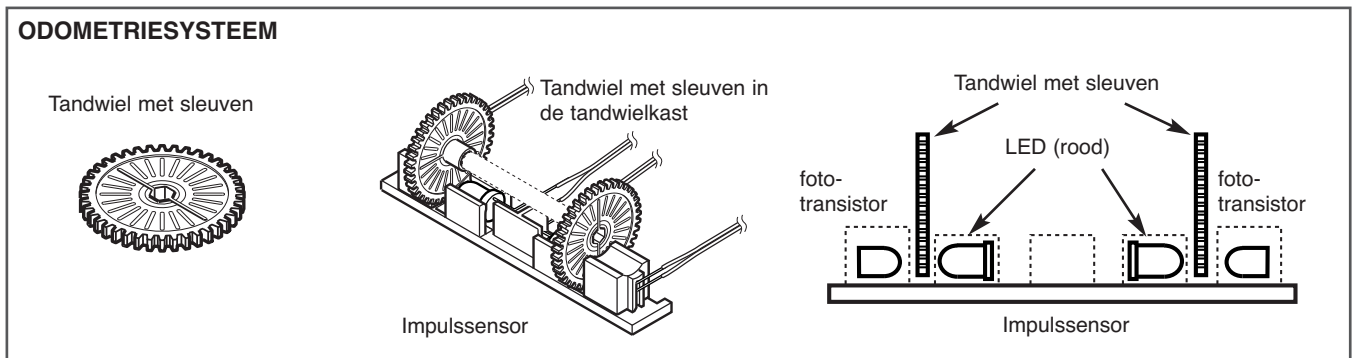
In een groot aantal apparaten van ons huishouden passen wij geregelde gelijkstroommotoren toe. Als je meer wilt weten over de snelheidsregeling met behulp van impulsbreedtemodulatie kunt je een handboek voor motorsturing uit de technische vakliteratuur bestuderen.

2. De meten van de (wiel)snelheid (Odometrie)

P!MOT rijdt naar voren, terug, maakt bochten naar links en rechts, en maakt een volledige draaicirkel op de plaats met beide motoren, maar soms kan de robot door de toleranties van het productieproces en de assemblage niet op een recht spoor blijven. Wij zullen nu analyseren hoe wij dit rijgedrag verbeteren kunnen.

Bij de assemblage van het tandwielsysteem heb je een tandwiel met een sleuven ingebouwd. Herinner je, je dat tandwiel nog? Bij die montage stap heb je ook een fototransistor en een rode LED ingebouwd en zo is er een impulsgever in de versnellingsbak geïntegreerd. Dit zijn de onderdelen die voor het optimaliseren van de rijnaauwkeurigheid verantwoordelijk zijn.

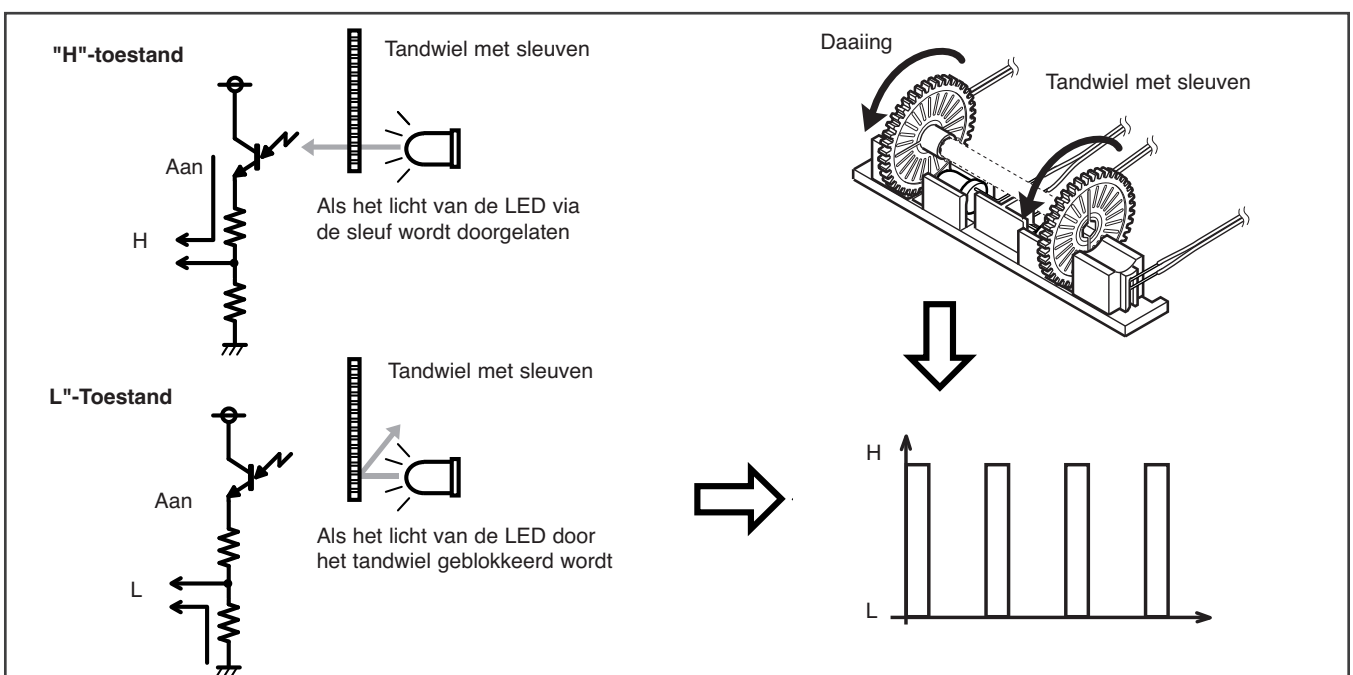
De volgende afbeelding laat zien hoe het tandwiel met de sleuven, de fototransistor en de rode LED in het tandwielsysteem gerangschikt zijn.



De rode LED wordt alleen geactiveerd, als de motor loopt. Als de processor de motor een rijden-commando geeft, schakelt de LED aan en begint de motor te draaien.

Omdat het tandwiel met de sleuf een bestanddeel van het tandwielsysteem vormt, begint ook dit tandwiel te draaien. Het licht van de LED wordt via de sleuf doorgelaten of door het tandwiel geblokkeerd. In het eerste geval wordt de fototransistor belicht en geactiveerd. Dat wil zeggen, dat de fototransistor een "H"-signaal ("H" = "hoge" spanning) genereert. Bij een blokkering van de lichtstraal genereert de fototransistor een "L"-signaal ("L" = "lage" spanning). (Zie de bijgevoegde tekening).

De motoraandrijving zet dus het tandwiel met de sleuf in beweging en levert aan de uitgang van de fototransistor een impulsvormsignaal af. Dit signaal wordt aan de processor doorgegeven. Omdat de processor weet welke wielen met welke snelheid moeten draaien, kun je met deze middelen een nauwkeurige snelheidsregeling opbouwen. De componenten in het odometriesensorsysteem genereren de "L" en "H" impuls niveaus.



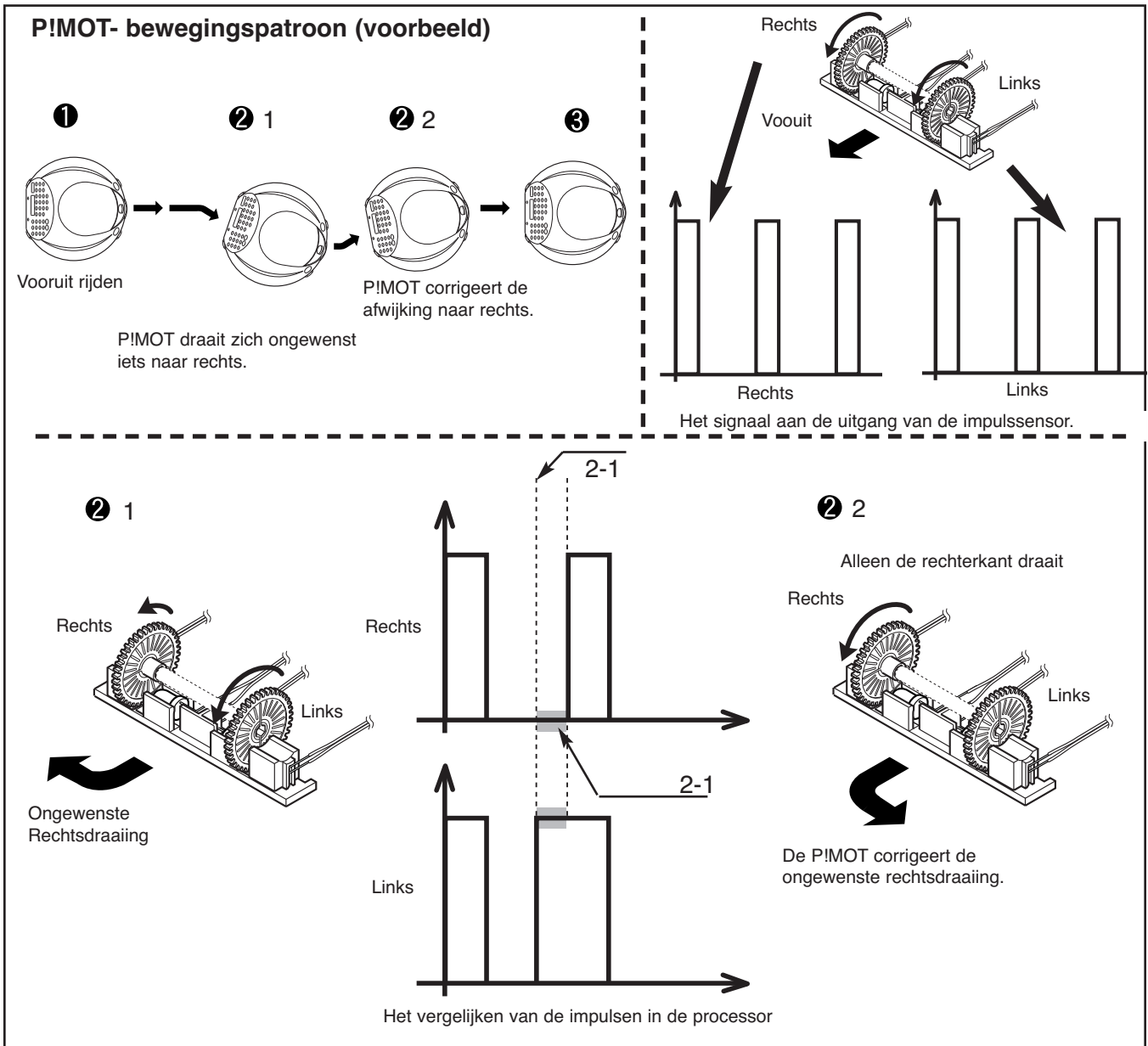
P!MOT maakt bij beide motoren gebruik van impulsensoren. Zowel bij een vooruit- of achteruitbeweging worden de impulsen aan de processor doorgegeven. Door vergelijking van de impulsen van de linker en rechter sensoren kunnen wij het rechtuit rijden verbeteren. Wat de processor precies moet doen, wordt later beschreven.

1). De processor genereert een rijden-commando.

De rode LED begint te branden en het wiel begint te draaien.

2). Als de rechter impulsensor een "H"-signaal en de linker impulsensor een "L"-signaal genereert, wordt de motor aan de rechterkant kort stopgezet, totdat de linker impulsensor een "H"-signaal afgeeft.

3). Als de linker impulsensor een "H"-signaal genereert, draaien de beide motoren weer.



We kunnen de P!MOT met deze correctiemethode tot een rechtlijnige beweging dwingen door de impulsgenerator in de robot te integreren. De processor zorgt voor de uitvoering van de correcties en bestuurt de motoren. Een processor met de passende software beheerst deze besturingsmethode. Een computer zonder software of een processor met een andere software kan deze correcties niet uitvoeren. De opbouw van het tandwiel met een sleuf, fototransistor en LED vormen samen een zogenaamde encoder (pulsgever). Een encoder is een systeem dat uit deze drie onderdelen is opgebouwd.

Een encoder is een sensor, die ons informatie bezorgt over de positie en snelheid van een draaiend object. Deze sensor is een buitengewoon praktische motorbesturing die voor een plaatsbepaling en voor de snelheidsmeting de impulsen telt.

D. Sensoren

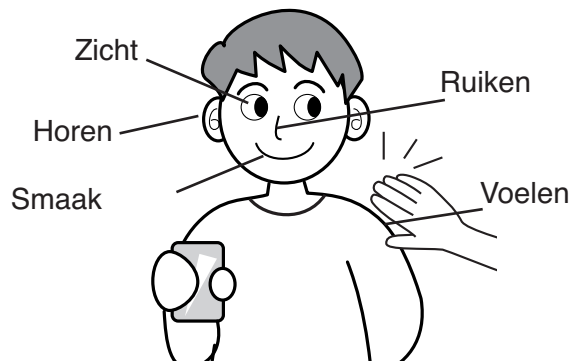
• Wat zijn sensoren?

Heb je al eens eerder een robot gezien? Misschien heb je op televisie wel eens iets over moderne robots gezien en gehoord. Hoe worden dergelijke robots gebouwd? Natuurlijk bevatten dergelijke robots processoren (het elektronische brein) en mechanische ledematen maar met name de sensoren hebben een belangrijke rol. Wat gebeurt er als zo'n systeem geen sensoren ter beschikking heeft? Welnu, zonder sensoren kan een robot zijn omgeving niet waarnemen en zich dus niet zelfstandig bewegen. Daarom was het nodig om diverse sensoren voor de robots te ontwerpen.

• Sensortypen

Ja, er bestaan diverse sensorentypes, maar welke typen leveren welke informatie? Je kunt de robot wel een beetje met de zintuigen van het menselijk lichaam vergelijken. Met zijn sensoren, dat wil zeggen de zintuigen, kan de mens zijn omgeving waarnemen. We onderscheiden vijf zintuigcategorieën: zien, horen, voelen, ruiken en smaak.

En zoals de mens over zintuigen beschikt, worden de robots met sensoren uitgerust.



* Zichtsensoren (lichtsensor, camera, enz.)

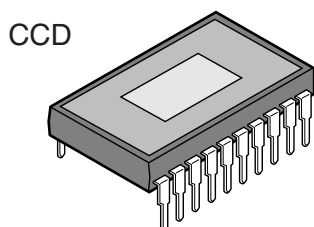
Er bestaan talloze soorten zichtsensoren, bijvoorbeeld sensoren, die alleen helder en donker kunnen onderscheiden en andere types zoals de camera die de omgeving als afbeelding waarnemen. P!MOT past drie lichtsensorentypes toe.

Zo is er een CdS-sensor, die we als bodemsensor toepassen. De CdS-sensor registreert met behulp van een variabele weerstandswaarde de verandering van de lichtintensiteit en onderscheidt een heldere van een donkere omgeving.

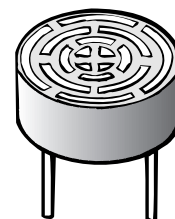
Op de tweede plaats is er de infraroodsensor, die in een IC zit en het voor mensen onzichtbare infraroodlicht waarneemt. Met het door de IR-LED uitgestraalde infraroodlicht kan de botsingssensor vaststellen of er een obstakel in de weg staat of ligt.

Op de derde en laatste plaats beschrijven wij de fototransistor die als impulssensor wordt toegepast. De fototransistor transformeert zichtbaar licht in stroom en onderscheidt alleen licht en donker. De impulssensor zelf is een lichtsensor en geen zichtsensoren.

De CCD in een camera is een beeldsensor. Deze CCD-sensor neemt de omgeving als afbeelding waar. Lichtsensoren worden toegepast in straatverlichtingssystemen en schakelen de straatlantaarns aan zodra het donker wordt. Ook vinden we dit soort sensoren in kopieermachines, waarin deze controleren of zich in de machine een origineel bevindt. CCD-sensoren past men daarentegen toe in camera-systemen of in de foutencontrole (bijvoorbeeld bij de controle van afmetingen).



Ultrasoonsensor



* **Gehoorsensoren (microfoon, ultrasoon, enz.)**

Een microfoon is de standaard sensor voor geluid. Een geluid is een luchtrilling. Een microfoon moet dus luchtrillingen waarnemen en in een elektrisch signaal kunnen omzetten.

Het menselijke hoorspectrum reikt van 20Hz tot 20.000 Hz. Hertz is de standaard eenheid voor de frequentie. 1Hz beschrijft één trilling per seconde en hoe hoger de Hertzwaarde reikt, des te hoger klinkt ook de toon. Sommige sensoren registreren ook de tonen, die buiten het hoorbereik van de mensen liggen, bijvoorbeeld ultrasone geluidstrillingen.

Een geluidsniveaumeter bevat een geluidssensor om het geluidsniveau te meten. En een ultrasone geluidssensor wordt in de visserij als onderwaterradar toegepast om een school vissen te lokaliseren.

* **Tastsensoren (druksensor, temperatuursensor, enz.)**

Er zijn talloze voelsensoren, zoals bijvoorbeeld het "drukgevoel", een "pijngevoel", een "kougevoel" of een "warmtesensatie". In dit soort categorieën staan talloze sensoren ter beschikking.

Zoals een sensor, die alleen het aanraken/niet aanraken onderscheidt of een rekstrookje, dat een druk in een evenredig elektrisch signaal kan omzetten.

P!MOT neemt beslissingen op basis van sensorengegevens, zoals "bodem gevonden" of "geen bodem gevonden".

Een bekende temperatuursensor is de thermistor, met als basis een temperatuurafhankelijke weerstand. Een dergelijke temperatuursensor wordt in een airconditioning toegepast om de omgevings-temperatuur op de gewenste waarde af te regelen.

Een voelsensor wordt in de tablet en telefoon toegepast om de apparaten via het scherm te besturen. Druksensoren worden ook voor gewichtscontrole gebruikt, bijvoorbeeld in een weegschaal, of bij het hanteren van lasten door robots. Daarbij zijn de robots zo gevoelig ingesteld, dat zij een ei kunnen pakken zonder de schaal te laten barsten.

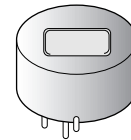
* **Reuksensoren (gasdetector, CO2 sensor, enz.)**

De reuksensor kan net als onze neus geuren waarnemen. De gassensor is een reuksensor en wordt in de keuken als gasalarmsensor toegepast.

Thermistor



Gassensor



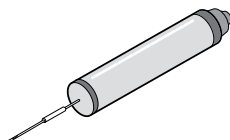
* **Smaaksensor (zoutsensor, enz.)**

Het waarnemen van smaak behoort bij de elektronische sensoren tot de moeilijkste opgaven. Een sensor kan goede of slechte smaken niet onderscheiden. De individuele smaak verschilt sterk van persoon tot persoon en hangt voor een groot deel van de gewoonten af. Daarom komen in de praktijk vrijwel alleen sensoren voor suiker- en zoutconcentraties voor.

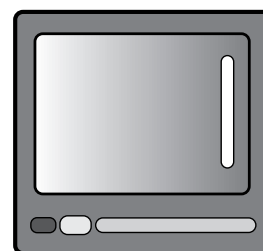
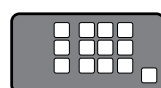
* **Overige sensoren**

Alhoewel de mens slechts vijf zintuigen kent, kunnen enkele sensoren meer waarnemen dan de menselijke organen. De Hallsensor kan bijvoorbeeld een magneetveld, de Geiger-Müller-teller die radioactiviteit en een infraroodsensor het voor ons onzichtbare infraroodlicht waarnemen.

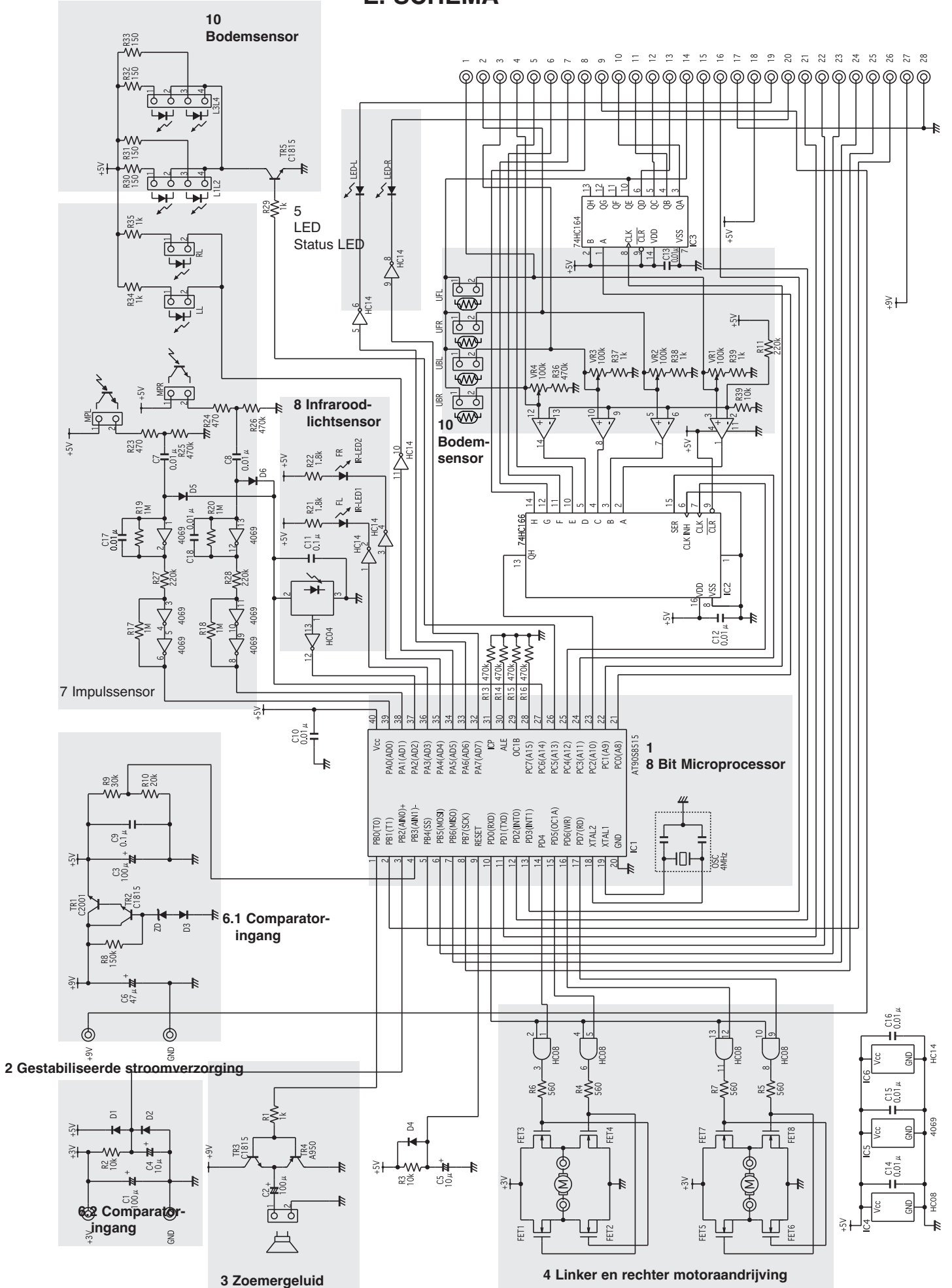
Geiger-Müller-Teller



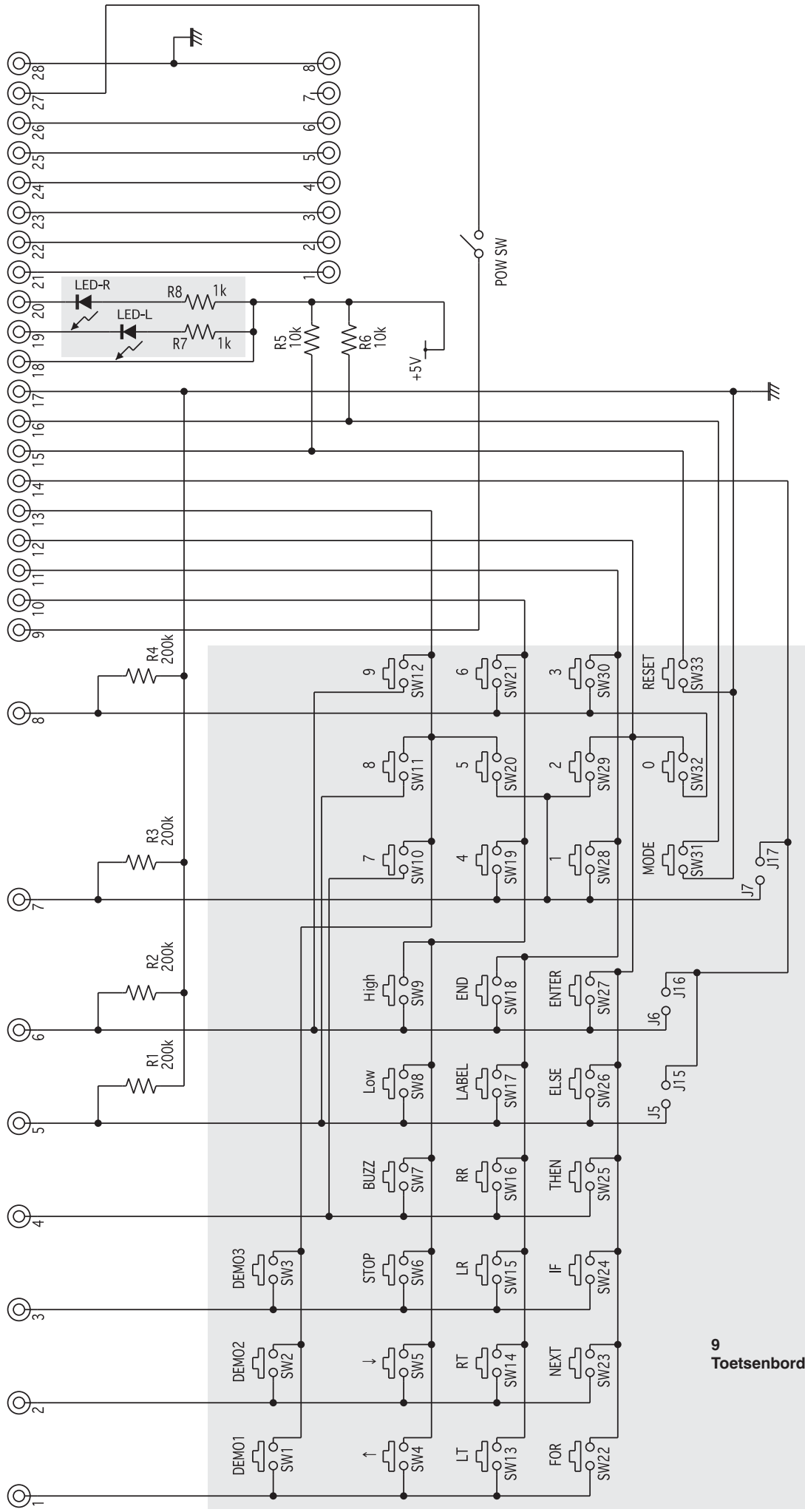
Infraroodsensor



E. SCHEMA



9 Display LED's



9 Toetsenbord

De P!MOT is een programmeerbare robot, die gebaseerd is op een 8 bit microprocessor "AT90S8515" van ATMEL. De P!MOT wordt volledig bestuurd door deze microprocessor. We zullen elke module individueel beschrijven.

(1) 8 Bit microprocessor

Het commando voor de motoraandrijving en voor de gewenste tooncombinaties wordt gemaakt op basis van de programmeercommando's die via het toetsenbord en de overige programmatechnische reacties aan de hand van de sensorensignalen.

De Software voor de P!MOT blijft permanent opgeslagen in een ROM, maar wordt gedurende het operationele bedrijf in het RAM van de microprocessor. Daar worden ook de processor, (en tijdelijk) de toetsenbordcommando's, statusgegevens van de motoren, timers, enz. opgeslagen.

(2) Gestabiliseerde stroomvoorzorging

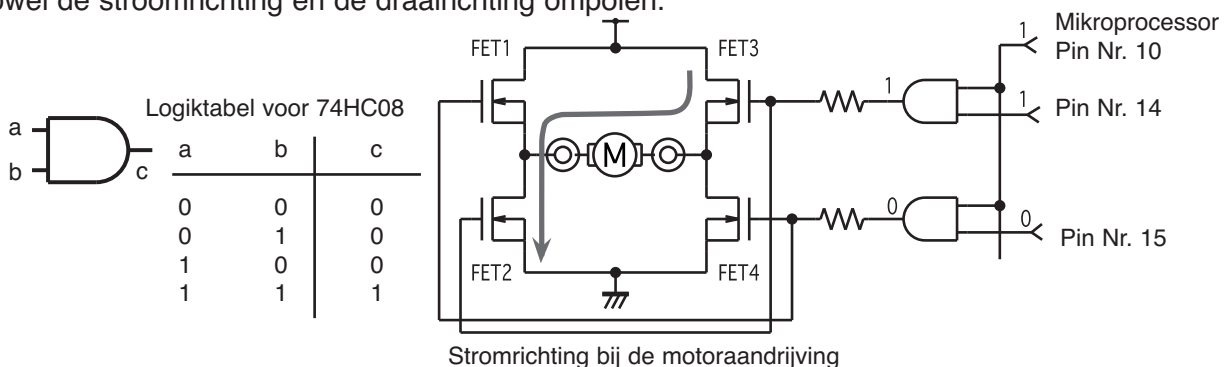
Deze module zorgt ervoor dat de microprocessor de stabiele 5V stroomvoorzorging krijgt. Zodra P!MOT een tijdlang geen commando's meer heeft verwerkt of per commando in een diepe sluimerfase is geplaatst, bevindt de robot in een diepe slaapfase. Als de stroomvoorzorging in de slaapmodus gereduceerd wordt, moet ook de ruststroom in de stroomvoorzorging verminderd worden. De stroomvoorzorging is daarom uitgerust met een 2-fasige eindtrap, die bestaat uit een transistor als versterker en een Darlington-uitgang. Deze twee fasen stellen ons in staat de schakeling in een energie sparende modus om te schakelen.

(3) Beeper

De microprocessor in de P!MOT kan vier verschillende tooncombinaties genereren. Omdat de luidspreker niet direct op de processor kan worden aangesloten, werken we met een extra versterkertrap.

(4) De linker en rechter motoraandrijving

Dit impulsmodulatiesysteem stuurt met behulp van de processorsignalen de motoraandrijving. Daartoe hebben wij de schakeling uitgerust met vier FET's, die om de motor gegroepeerd worden. In dit geval worden de FET's als schakelaars gebruikt, die op een "1"-commando van de microprocessor sluiten. Alhoewel de schakeling in totaal 8 FET's elk met eigen signaalaansluitingen bevat, heeft de P!MOT toch slechts vijf besturingscommando's nodig. Ter codering van de signalen en ter beoordeling welke FET moet worden aangestuurd, wordt IC4 toegepast. Je kunt in de coderingstabel voor de logische schakeling 74HC(s)08 aflezen hoe deze methode in elkaar steekt. Wij onderzoeken nu, wat er met de commando's voor de besturing van de linkermotor gebeurt. Voor een stopsignaal wordt een "0" op pin 10 van de microprocessor gegenereerd. De FET wordt uitgeschakeld en er vloeit geen motorstroom. Voor een activeringssignaal (een "rijdsignaal") wordt een "1" op pin 10 van de microprocessor gegenereerd. Pin 14 is "1" en Pin 15 is "0", zodat FET2 en FET3 gaan geleiden en de stroom via FET3 naar FET2 kan vloeien. Bij een ompolen van de draairichting moet Pin 10 weer naar "1" schakelen en worden Pin 14 en Pin 15 geïnverteerd, zodat zowel de stroomrichting en de draairichting ompolen.



5) LED Statusindicatie

LED-L (groen) en LED-R (rood) beschrijven de toestand van de P!MOT. Tijdens een programmeerfase zijn beide LED's geactiveerd. Daarbij signaleert LED-L (groen) een correct programmeercommando en LED-R (rood) een foutief of verkeerd begrepen programmeercommando. Tijdens de uitvoering van een programma, als de stroomvoorzorging wegvalt of de processor in de slaapmodus geraakt, worden beide LED's gedoofd. Tijdens de uitvoering van een programma worden de LED's bij een IF-conditie ingeschakeld. Bij het activeren van de linker infraroodsensor, de voorste sensor of de linker lichtsensor, knippert de groene LED-L en voor de rechterkant knippert de rode LED-R.

(6) Comparator-ingang

Ter controle van de laadtoestand van de batterijen worden de bedrijfsspanningen via een spanningsdeeler R9/R10 op de processingingang voor de ingebouwde comparator aangesloten.

Elke spanning wordt individueel gecontroleerd. Als de stroomvoorzorging niet voldoende vermogen voor de motoren kan leveren, waarschuwt de beeper alle aanwezigen. De dioden D1 en D2 onderdrukken de stoorsignalen die een lopende motor veroorzaakt en beschermen de microprocessor tegen spanningspieken.

(7) Impulssensor

De impulssensor bestaat uit een tandwiel met een sleuf, een fototransistor en een LED, die tezamen een encoder vormen. Deze sensor wordt op de uitgang van het tandwielsysteem aangesloten en levert de microprocessor een impulsreeks met het toerental aan deze uitgang, zodat de P!MOT nauwkeuriger rechtuit in het spoor kan blijven rijden. In hoofdstuk 3 wordt de motorsturing gedetailleerd beschreven (zie de afbeelding in dat hoofdstuk).

Wij onderzoeken nu, hoe de robot op een impuls uit de linkersensor reageert;

Zodra de motor na een "0"-commando op pin 27 van de microprocessor stopt, biedt de linker impulssensor een "0" aan op pin 40. Vervolgens wordt pin 27 op "1" gezet, zodat de motor weer verder loopt. De amplitude van het signaal is minimaal en wordt daarom versterkt in IC5, waarbij de signaalvorm door R17 wordt afgevlakt.

Zodra het licht in de fototransistor weggevallen is, genereert de sensor weer een "0" op pin 40 van de microprocessor. De processor controleert de spanningscurven van de sensoren op de pins 40 en 39 en stuurt de motor recht vooruit.

(8) Infraroodlichtsensor

Bij het verwerken van een IF-conditie in een programma controleert de P!MOT de toestand van de infraroodsensor. Deze bestaat uit een infrarooddetector en een infrarode LED voor de linker- en rechterkant. De detector registreert de gereflecteerde signalen van muren en andere obstakels. Op die manier kan P!MOT obstakels waarnemen. Zodra de robot gereflecteerd licht ontdekt, levert de sensor een signaal af aan pin 37 van de microprocessor. De P!MOT reageert op de betreffende sensorensignalen door de in het programma vastgelegde opdrachten.

(9) Toetsenbord

Via deze module kunt je op de 33 toetsen je programma's invoeren. Het toetsenbord representeert een 4 x 8 matrix. Om het aantal signaalaansluitingen naar de microprocessor te reduceren worden de toetsensignalen in IC2 en IC3 gecodeerd. Op dezelfde wijze worden ook de signalen van de vier bodemsensoren gecodeerd.

- 🕒 74HC166 is een 8 Bit schuifregister, dat parallelsignalen in seriesignalen transformeert.
- 🕒 74HC164 is een 8 Bit schuifregister, dat seriesignalen in parallelsignalen transformeert.

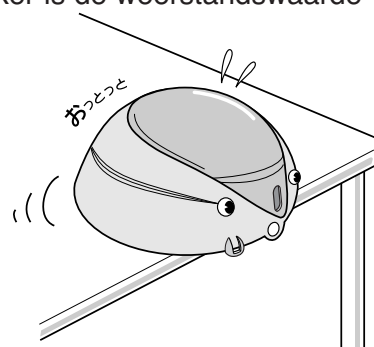
(10) Bodemsensor

Bij het verwerken van een IF-conditie in een programma controleert de P!MOT de toestand van de bodemsensoren. Deze bestaan uit een CdS-detector en een LED voor de linker- en rechterzijde, elk ook nog eens voor de voor- en achterkant.

Omdat P!MOT slechts een beperkt aantal pins (aansluitingen) aan de microprocessor ter beschikking heeft worden de signaalleidingen in een matrix gecodeerd.

De CdS beschikt over een lichtgevoelige weerstand. In het donker is de weerstandswaarde vrijwel "0" en in het licht "1". Daartussen kan het signaal onbetrouwbaar zijn en variëren.

Daarom wordt het signaal vóór een verdere verwerking in de microcomputer gefilterd in IC8 en VR. Aan de hand van het gereflecteerde signaal kan P!MOT beoordelen of er zich onder de sensor een bodem bevindt of niet.



F. Softwareprogramma

Wie thuis of op school een PC ter beschikking heeft kan de P!MOT ook op een PC programmeren en het programma dan naar de robot uploaden.

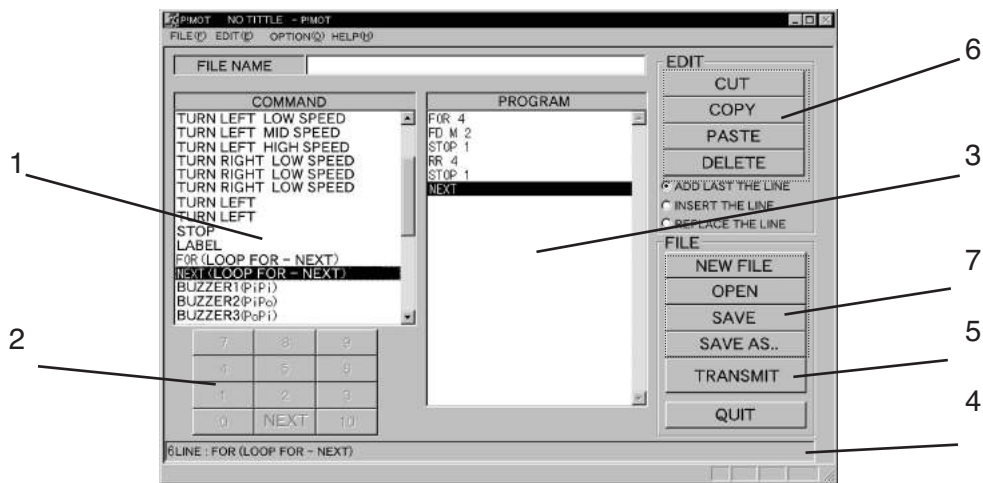
De programma's worden in de PC geschreven en kunnen daar dan ook worden opgeslagen.

De communicatie met de PC vindt via een oud type Centronics printerkabel plaats die wordt aangesloten op een interface (IF-96, niet bij het bouwpakket inbegrepen).

Het programma behoort is freeware je eenvoudig van de website van ELEKKIT Japan kunt downloaden.

Het softwareprogramma voor de P!MOT

- ⌘ Je kunt het toetsenbord van de P!MOT via de PC programmeren.
- ⌘ Op de PC kunt je eenvoudig meerdere programma's schrijven en opslaan.
- ⌘ De programma's worden regel voor regel afgebeeld en met de muis woden aangeklikt.
- ⌘ De afbeelding van de softwarecode is voor beginners een prima hulpmiddel.



(1) Lijst met commando's

Het geselecteerde commando wordt in het programma voor de P!MOT opgenomen.

(2) Toetsenbord voor de cijfers

Deze toetsen worden toegepast voor het invoeren van getallen voor tijdsperioden, herhalingen, labels, enzovoort.

(3) De afbeelding van de softwarecode

Deze module documenteert het programma, dat je momenteel voor de P!MOT schrijft. Je kunt nu bijvoorbeeld een nieuw commando kiezen.

(4) Programmaregel

beschrijft de inhoud van de momenteel geselecteerde programmaregel.

(5) Programma upload

transfereert het geschreven programma naar de P!MOT. Daartoe moet de P!MOT eerst op de ontvangstmodus overgeschakeld worden. Druk daartoe (in de hier aangegeven volgorde): "**RESET**", "**MODE**", en "**0**". Controleer daarbij, dat beide LED's (LED-L en LED-R) knipperen en start vervolgens de transfer.

(6) Bewerkingstoets

Met behulp van de bewerkingstoets schakelt je de software over in de bewerkingsmodus, waarin je het programma verder kunt bewerken.

(7) Bestandstoets

Met behulp van de bestandstoets wordt een programmabestand opgeslagen, respectievelijk geopend. Voor de transfer van programma's tussen PC en P!MOT heb je een optionele interfacekaart (IF-96E) en een optionele centronic printerkabel nodig. **De downloadsite is <http://www.elekit.co.jp>**

G. Notities



Er is een hele serie robots en electronica bouwpakketen te koop voor scholen en leerlingen die er meer over willen weten. Ze zijn onder andere te koop bij Conrad, zie de website: www.conrad.nl en zoek daar op AREXX.

