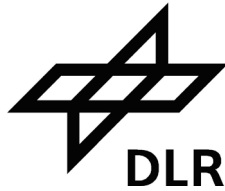
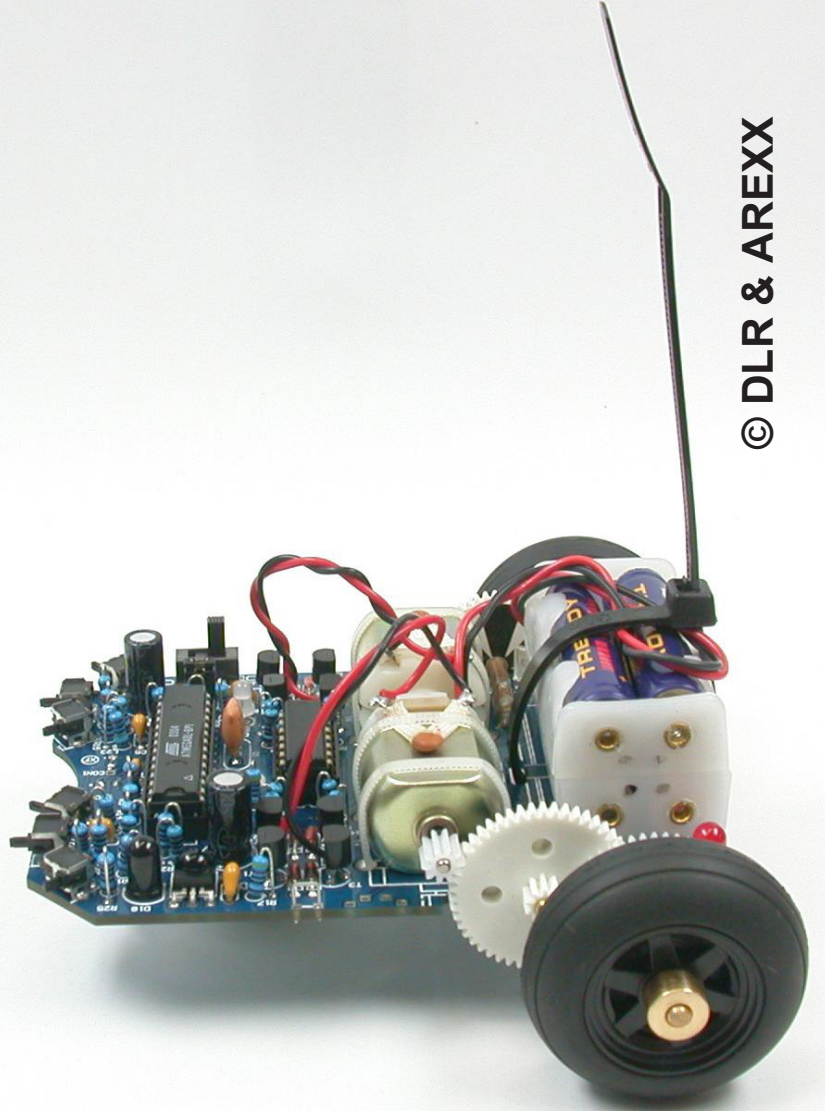
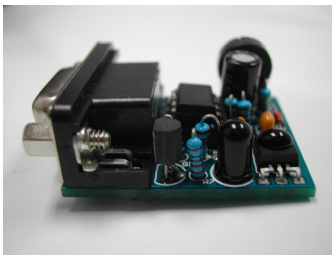
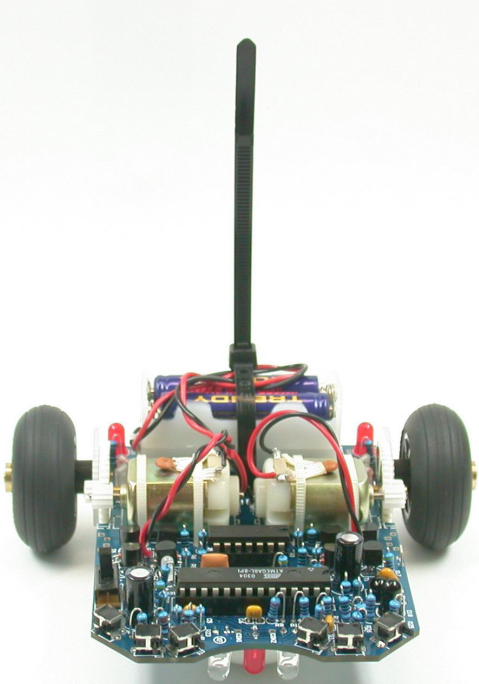


Licence by DLR



ASURO ROBOT KIT



© DLR & AREXX

BOUW- EN BEDIENINGS HANDLEIDING

Model ARX-03

Fabrikant

AREXX, Zwolle NEDERLAND
JAMA, Taichung - TAIWAN


www.arexx.com



Inleiding

De ASURO is een kleine, vrij in C programmeerbare, mobiele robot, die in het Duitse Centrum voor Lucht- en Ruimtevaart (DLR = Zentrum für Luft- und Raumfahrt) in het Instituut voor Robotics en Mechatronica voor het onderwijs is ontwikkeld. De montage van de robot is voor de ervaren elektronicus een eenvoudige klus en voor de beginners op het gebied van de elektronica een uitstekend startobject. Met uitzondering van de printen bevat het bouw pakket gewone, gemakkelijk in de handel verkrijgbare onderdelen, die men met gewone handgereedschappen kan monteren. Voor de programmering passen wij uitsluitend vrije software (freeware) toe. De ASURO is dus uitstekend geschikt voor de hobbyist, die ervaring op het gebied van processorgestuurde schakelingen wil opbouwen, voor scholieren en studieprojecten, opbouwstudies en cursussen aan volkshogescholen. Doordat de elektronica geheel met behulp van (althans voor privégebruik kostenloze) freeware-software heeft plaatsgevonden, levert het ASURO-project het bewijs, dat men ook zonder omvangrijke en dure software, techniek en werktuigen een goed werkende robot kan ontwerpen.

De ASURO beschikt over een RISC-Processor, twee individueel bestuurbare motoren, een optische lijnvolger, zes botsingdetectoren, twee snelheidsensoren op de wielen, drie optische indicatoren en een infrarood-communicatiemodule, die voor de programmering en ook voor een afstandsbediening via de PC kan worden toegepast (zie figuur .0.1).

Het waarschuwingsymbool  wijst op de risico's in bepaalde hoofdstukken, die de lezer zorgvuldig moet bestuderen, omdat fouten tot schade aan de apparatuur en aan de gezondheid kunnen leiden.

Wij wijzen er in dit kader ook op, dat de ASURO natuurlijk geen speelgoed en voor kinderen onder de drie jaar ongeschikt is, omdat het bouw pakket talloze miniatuuronderdelen bevat, die kinderen kunnen inslikken.

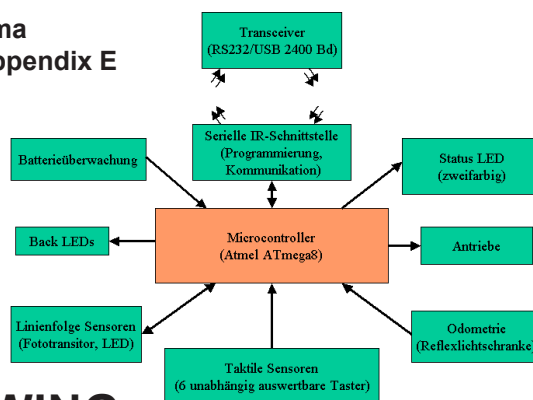
Ach ja, ASURO betekent overigens "Another Small and Unique Robot from Oberpfaffenhofen"!

© DLR & AREXX

Oberpfaffenhofen 2004

Jan Grewe
Robin Gruber

Afb.0.1 Blokschema
zie ook appendix E



WAARSCHUWING

- * Zodra de plasticverpakkingen met de componenten en onderdelen worden geopend, vervalt het recht op teruggave.
- * Lees voor het bouwen eerst de handleiding zorgvuldig door.
- * Wees steeds voorzichtig bij het werken met gereedschappen
- * Bouw niet aan dit project in gezelschap van kleine kinderen. De kinderen kunnen zich aan de gereedschappen verwonden of kleine onderdelen in de mond steken.
- * Let op de correcte inbouw van batterijen en accu's
- * Zorg ervoor, dat de batterijen/accu's en batterijcompartimenten steeds droog blijven.
- * Als de ASURO eens nat wordt, moet je de batterijen verwijderen.
- * Verwijder de batterijen, als de robot langer dan een week niet wordt gebruikt.

Inhoudsopgave

I. Mechanica	6
1. Benodigde gereedschappen	6
2. Mechanische voorbereidingen	7
2.1. Motortandwiel	7
2.2. Pingpong balletje	7
2.3. Wielsensoren	8
II. Elektronica	9
3. Inwijding in de kunst van het solderen	9
3.1. Soldeerpunt, tin en temperatuur	9
3.2. Voorbereidende werkzaamheden	10
3.3. Insolderen van de componenten	11
3.4. Uitsolderen componenten	12
4. Montagefase	13
4.1. Montage van de RS232-Infrarood-transceiver	13
4.2. info USB-Infrarood-transceiver	15
4.3. montage van de ASURO-hoofdprint	16
4.4. Motormontage	20
4.5. Stroomvoorziening	20
5. Inbedrijfname en test	21
5.1. RS232-Infrarood-transceiver	21
5.2. USB-Infrarood-transceiver	22
5.2.1. Windows	22
5.2.2. Linux	23
5.3. ASURO	24
5.3.1. Indicatoren	25
5.3.2. Fototransistoren (T9, T10)	25
5.3.3. Schakelaars	26
5.3.4. Lichtgevoelige sensoren (Hodometrie)	26
5.3.5. Andrijving	26
5.3.6. IR Transceiver	26
5.3.7. Klaar?	27

6. Foutzoeken	28
6.1. RS232-IR-transceiver werkt niet !	28
6.1.1. Ingetoetste letters en de weergegeven letters verschillen	28
6.1.2. Het terminalprogramma geeft geen tekens weer	28
6.1.3. Het werkt nog steeds niet	28
6.2. USB-Infrarot-tranceiver funktioneert niet	28
6.3. Back-LEDs (D15,D16) lichten niet op!	28
6.3.1. Geen van de Back-LEDs doet het	28
6.3.2. Slechts een van de beide LED's licht op	29
6.3.3. Status-LED (D12) licht na de Start niet tweekleurig op	29
6.4. Een Indicator doet het niet	29
6.4.1. Status-LED D12 doet het niet	29
6.4.2. Voorste-LED D11 doet het niet	29
6.4.3. Linker Back-LED D15 doet het niet	30
6.4.4. Rechter Back-LED D16 doet het niet	30
6.5. Lijnvolgesensor (T9, T10) reageert niet	30
6.6. Een schakelaar funktioneert niet goed	30
6.6.1. Het lijkt erop dat op een combinatie schakelaars gedrukt wordt	30
6.6.2. Het lijkt erop dat er schakelaars omgedraaid zijn	31
6.6.3. Opde een of andere manier werkt het nog steeds niet goed	31
6.7. Een lichtsensor werkt niet goed	31
6.7.1. Geen van de licht sensoren doet het	31
6.7.2. De linker lichtsensor doet het niet	31
6.7.3. De rechter lichtsensor doet het niet	31
6.8. Een motor doet het niet	32
6.8.1. beide motoren doen het niet	32
6.8.2. De linker motor draait alleen in één richting	32
6.8.3. De rechter motor draait alleen in één richting	32
6.8.4. Een motor draait in de verkeerde richting	32
6.9. IR-transceiver	32
6.9.1. ASURO zend geen tekens	32
6.9.2. ASURO ontvangt geen tekens	32
6.9.3. het gaat nog steeds niet zo goed	33
7. De afsluitende karweitjes	34
III. Informatica	35
8. Installatie van de software en de eerste programma's	35
8.1. Windows	35
8.1.1. Flash-programma	35
8.1.2. Installatie van de programma editors en de compilers	35
8.1.3. Voorbeeldprogramma's	39
8.2. Linux	51
8.2.1. Flash-programma	51
8.2.2. Compiler	52
8.3. Flash - het ASURO-programmeer programma	53
8.3.1. Hoe funktioneert het Flashen?	53
8.4. Flash fouten	54
8.5. Je eerste eigen programma	54

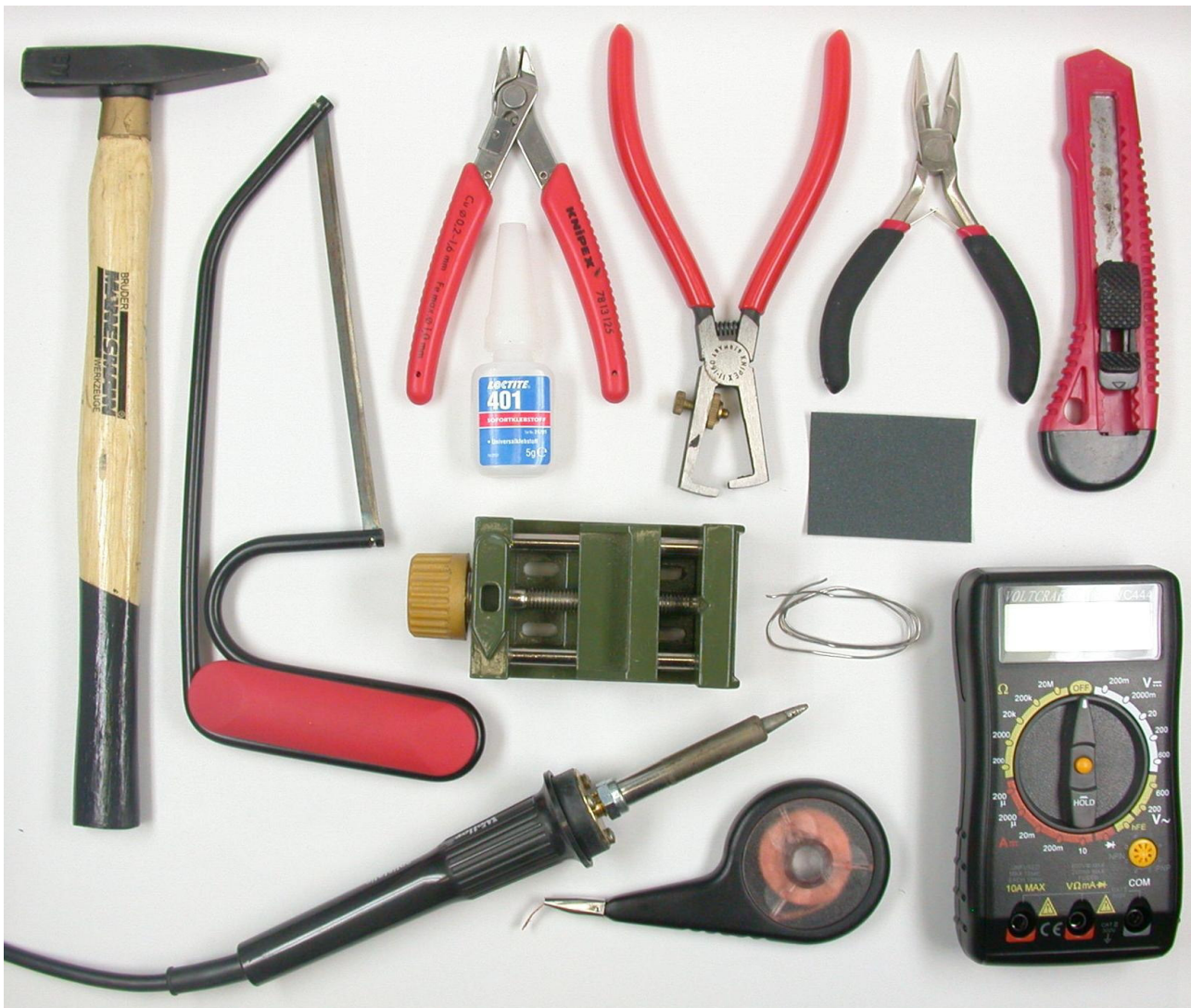
9. C voor de ASURO	56
9.1. Grondslagen van de programmering in C	56
9.1.1. Algemeen	56
9.1.2. Variabelen en datatypen	57
9.1.3. Compilerdirektieven (richtlijnen)	59
9.1.4. Voorwaardelijke stukturen	59
9.1.5. Herhalingsstructuren	61
9.1.6. Funkties	62
9.1.7. Pointers en Vectoren	64
9.2. Beschrijving van de ASURO-funkties	65
9.2.1. void Init(void)	66
9.2.2. void StatusLED(unsigned char color)	66
9.2.3. void FrontLED(unsigned char status)	67
9.2.4. void BackLED(unsigned char left, unsigned char right)	67
9.2.5. void Sleep(unsigned char time72kHz)	67
9.2.6. void MotorDir(unsigned char left_dir, unsigned char right_dir)	67
9.2.7. void MotorSpeed(unsigned char left_speed, unsigned char right_speed)	68
9.2.8. void SerWrite(unsigned char *data, unsigned char length)	68
9.2.9. void SerRead(unsigned char *data, unsigned char length, unsigned int timeout)	68
9.2.10. void LineData(unsigned int *data)	69
9.2.11. void OdometrieData(unsigned int *data)	70
9.2.12. unsigned char PollSwitch(void)	71
IV. Appendix	72
A. Onderdelenlijst	72
B. Schema ASURO	74
C. RS-232 IR Transceiver	75
D. USB IR-Transceiver	76
E. Blokschema ASURO	77
F. Blokschema PIC Processor	77
G. Leveromvang ASURO KIT	78
H. Elektronica-componenten	79

Deel I. Mechanica

1. Benodigde gereedschappen

Om de ASURO goed te kunnen monteren heeft men - naast de onderdelen uit de bouwdoos - de volgende gereedschappen en hulpmaterialen nodig:

- een kleine spanvoet of “derde hand”: twee handen zijn immers niet altijd voldoende
- een Stanley mes of een kleine zaag
- een fijne montagetang
- een zijknijptangetje (geschikt voor elektronica)
- eventueel een striptang
- soldeerbout: voor elektronicawerk (ca. 20W tot 40W) of een soldeerstation (minimaal 50W)
- soldeer: 1 mm dik soldeerdraad voor elektronica, dat eventueel ook loodvrij mag zijn.
- Desoldeer-litze: ca. 2-3mm breed, om soldeer te verwijderen (voor het geval er wat op een ongewenste plaats belandt)
- fijnkorrelig schuurpapier
- secondenlijm, tweecomponentenlijm of een kleefpistool
- eventueel een hamertje
- eventueel een multimeter
- computer: Laptop of een PC met Windows of Linux



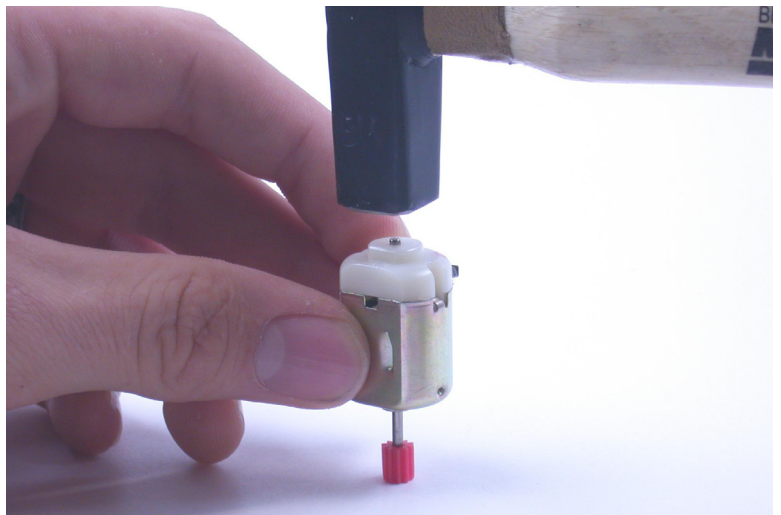
2. Mechanische voorbereidingen

Voordat je met het eerste knutselwerk kunt beginnen, is het raadzaam om eerst even te controleren of alle benodigde onderdelen ter beschikking staan. Dat kun je het eenvoudigst met de onderdelenlijst uit appendix A doen. Voordat wij overgaan tot het elektronicawerk, moeten wij daarna ook nog enkele mechanische karweitjes doen.

2.1. Het motortandwiel

Om ervoor te zorgen, dat de motoren het vermogen goed op de assen kunnen overbrengen, moeten de motortandwielen (dit zijn de tandwielen met de 1,9mm boring en de tien tandjes) op de assen worden gedrukt. Als de door ons in de verpakking geleverde motoren niet reeds met deze tandwielletjes op de assen worden geleverd, moeten wij ze zelf nog even bevestigen. Daartoe steek je zonder al te veel kracht op de as van beide motoren een tandwielletje. Eerst moet het wielletje gewoon blijven zitten. Dan plaats je de motor met het tandwiel naar beneden op een niet al te harde ondergrond (plastic, karton, of iets dergelijks) en klopt je voorzichtig met een hamertje op het uiteinde van de motoras, dat aan de achterkant van de motor naar buiten steekt. Herhaal het kloppen totdat de as van de motor helemaal in het tandwielletje steekt (zie figuur 2.1).

Je kunt het monteren ook iets anders aanpakken en het met de hand geplaatste tandwielletje met behulp van een bankschroef op de motoras aandrukken. Daarbij mag je echter alleen druk op de as van de motor en in geen geval druk op de behuizing of de lagers uitoefenen.



Figuur 2.1.: Aandrukken van het motortandwiel

2.2. Het pingpongballetje

De afgebouwde ASURO glijdt op een half pingpongballetje. Dit "voorwiel" moet je eerst zelf nog even maken. Daarvoor neem je een heel pingpongballetje en zaagt of snijdt dit gewoon (d.w.z. het balletje en niet je eigen vingers!) met een Stanley mes doormidden. De snijkant moet je nog even met een kleine vijl of met schuurpapier bijwerken.



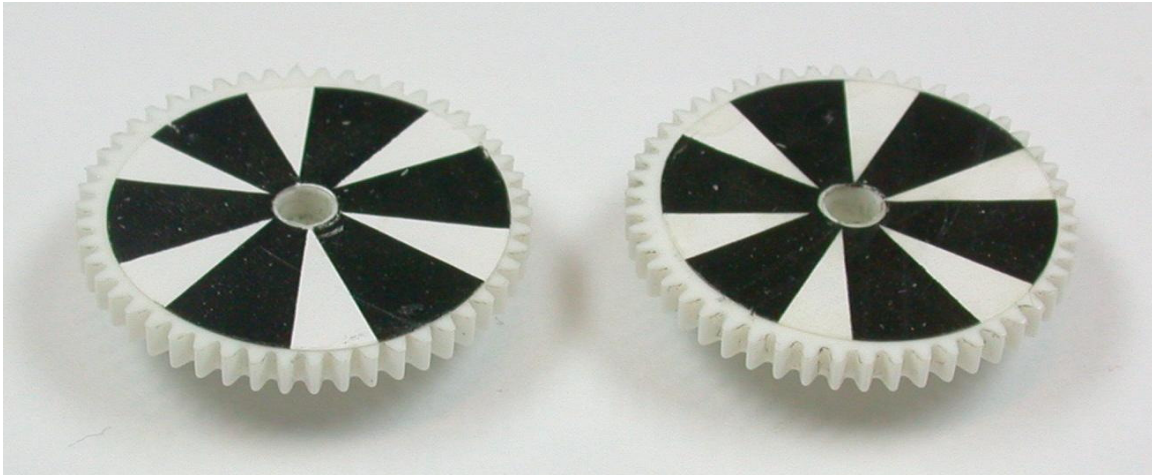
Figuur 2.2.: Door midden gezaagd pingpongballetje



Omdat pingpongballetjes zeer brandbaar zijn, mag je bij dit karweitje geen elektrische gereedschappen gebruiken.

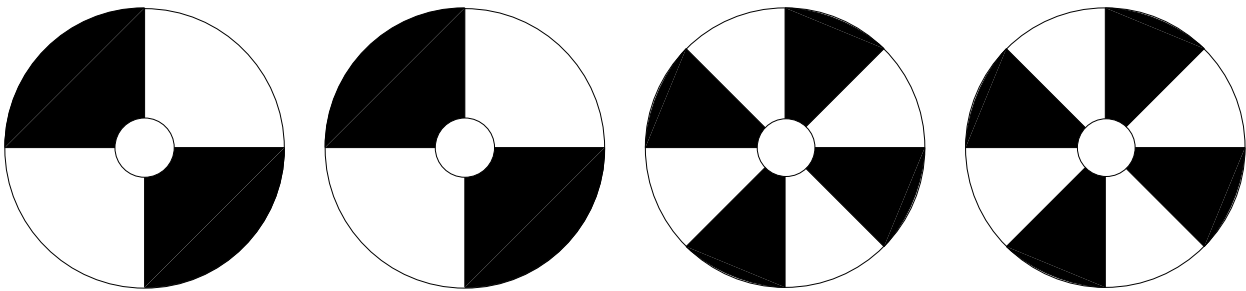
2.3. De wielsensoren

De LED en de fototransistor (deze vormen samen de wielsensor voor de (h)odometer**), die zich vol vertrouwen tot het eerste aandrijftandwiel wenden, mogen wij niet teleurstellen. Daarom moet je op de eerste aandrijftandwielen (dit is het tandwiel met de 50, respectievelijk 10 tandjes) aan de vlakke zijkant zonder tandkrans nog even de zelfklevende patroonschijven aanbrengen (zie figuur 2.3.)..



Figuur 2.3.: De opgeplakte (zelfklevende) patroonschijven

Hoe meer segmenten het patroon bevat, des te nauwkeuriger kunnen wij het toerental van het tandwiel en daarmee ook de snelheid van de ASURO meten. Een te groot aantal segmenten in het patroon veroorzaakt echter een afname van de gevoeligheid in de meting tussen licht en donker.



Figuur 2.4.: Voorbeelden van de wielsensoren voor de hodometer

Nu ben je al klaar met het eerste deel. De mechanische onderdelen zijn nu allemaal afgewerkt.

Korte pauze...

Nu gaan we verder met de elektronica.

** Hodometer (Hodos = Grieks voor weg) dit is een meter die de afstand meet!

Deel II. Elektronica

3. Inwijding in de kunst van het solderen

Alhoewel alle onderdelen voor de ASURO met draden worden bevestigd en dus - in tegenstelling tot de **Surface Mounted Device-componenten (SMD) (figuur 3.1 demonstreert het verschil tussen de kleinste verkrijgbare SMD-behuizing en de door ons gebruikte versie van de ASURO Processor. De daarin toegepaste siliciumchip is in beide gevallen volkomen gelijk!) - bijzonder goed met de hand kunnen worden gemonteerd, moeten wij toch vooral de beginnelingen der soldeerkunst met een paar tips op weg helpen.



Het spreekt eigenlijk vanzelf, dat de te bewerken printplaat in elk geval nergens onder spanning mag staan.

Uitschakelen alleen is niet voldoende! Je moet ook de batterijen verwijderen!

*** SMD betekent oppervlaktemontage. Deze onderdelen hebben geen lange aansluitdraden.*

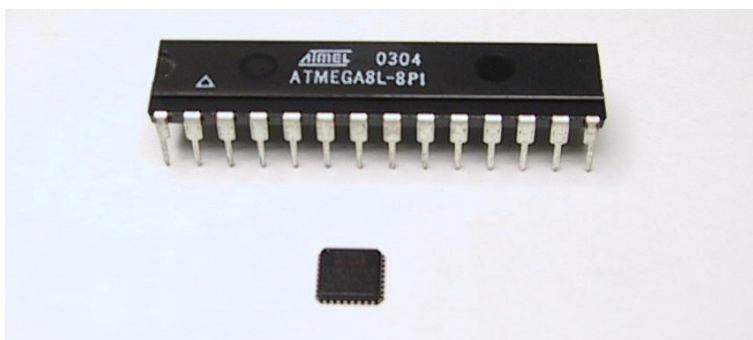
3.1. Soldeerpunt, soldeertin en soldeertemperatuur

Figuur 3.2 beschrijft het belangrijkste werktuig voor goed soldeerwerk! De hete soldeerpunt hoort bij het solderen van loodhoudend soldeer een temperatuur van ca. 360°C en bij het solderen van loodvrij soldeer een temperatuur van ca. 390°C te bereiken. Voor het solderen van de assen mag de temperatuur iets worden verhoogd (420°C). Bij het solderen van de in de ASURO toegepaste elektronische onderdelen gebruiken wij een potloodvormige soldeerpunt. Voor het solderen van de assen mag je eventueel een bredere punt toepassen.

Verder moet je de soldeerspons nog even onder de waterkraan houden (de spons mag niet drijfnat zijn) en dan kort voor het solderen de hete soldeerpunt van een oppervlakkig laagje soldeer voorzien.

Kort voordat je de soldeerbout na een pauze of aan het begin van een soldeerkarweitje tegen een te solderen onderdeel houdt of telkens wanneer de slakken van tinresten aan de soldeerpunt zichtbaar opvallen, kun je de soldeerpunt eenvoudig even aan de vochtige soldeerspons afvegen.

Als soldeertin gebruik je soldeerdraad met 0,8 of 1mm diameter.



Figuur 3.1 De kleinste verkrijgbare behuizing en de door ons gebruikte -grootste- versie van de ATmega8L-processor.



Figuur 3.2.: Het belangrijkste werktuig voor goed soldeerwerk

De dampen, die bij het solderen ontstaan, zijn bij langdurige blootstelling ongezond en het inademen moet je dus zoveel mogelijk vermijden. Nog beter is het werken onder een afzuiginstallatie! Gebruik alleen soldeertin voor elektronicaonderdelen. Gebruik geen andere soort soldeertin of "soldeerwater", want deze kunnen de schakeling of de onderdelen onherstelbaar beschadigen!



3.2. Voorbereidende werkzaamheden

Iedereen, die al eens heeft gesoldeerd, kent het probleem: eigenlijk heb je steeds een hand te weinig ter beschikking. Daarom is het geen wonder, dat er enkele trucs bestaan om de diverse onderdelen onder controle te houden totdat men deze met soldeerbout en tin goed te lijf kan gaan. ASURO's transistors, de LED's, de fototransistors, de IC's, druktoetsen, schakelaars, condensatoren en draadbruggen zijn reeds goed gevormd met draadaansluitingen in de gewenste richting. Bij de diodes en weerstanden moeten wij deze aansluitingen nog voorbereiden.

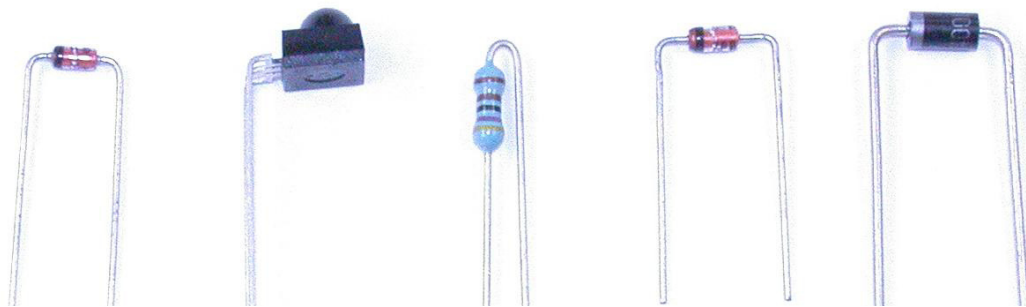
In de compacte bouwwijze van de ASURO worden alle weerstanden rechtopstaand ingebouwd, d.w.z. een aansluiting blijft onveranderd recht en de tweede draad wordt 180° gebogen. Voor deze ombuiging behoor je een diameter van 2,5mm te gebruiken en daarbij enkele millimeters afstand tot de behuizing over te laten. Deze maatregelen verhinderen een mechanische belasting van het onderdeel, die tot schade zou kunnen leiden.

Bij het later nog volgende soldeerwerk geeft een cirkel in de opdruk van de print aan boven welk boorgat de weerstand moet worden geplaatst en een kleine markering wijst naar het bijbehorende gaatje voor de tweede draadaansluiting.

De diodes worden liggend ingebouwd, dat wil zeggen, dat beide aansluitingen (liefst met een fijn montagetangetje) haaks moet worden omgebogen, zodat deze gemakkelijk in de bijpassende boorgaten kunnen worden gemonteerd.

De processor IC1 ATmega8, de poortschakeling IC3 CD4081 en de IR-ontvanger IC2 SFH5110-36 zijn alle drie gevoelig voor elektrostatische ontladingen. Dit betekent, dat de onderdelen al kunnen worden beschadigd of vernield door deze gewoon aan te pakken, als men tenminste tevoren elektrostatisch opgeladen was. Het opladen gebeurt bijvoorbeeld al door het lopen over een vaste vloerbedekking. Bij het werken met deze gevoelige onderdelen dient men zich met een aardleidingband te aarden of tenminste de metalen behuizing van een geaard apparaat of bijvoorbeeld een waterleiding, respectievelijk een verwarming aan te raken.





Figuur 3.3.: Onderdelen met correct gebogen aansluitdraden

A. Onderdeel

B. Draden iets buigen zodat het onderdeel niet uit de printplaat kan vallen

C. De tin moet goed in het gat vloeien

D. Soldeertin

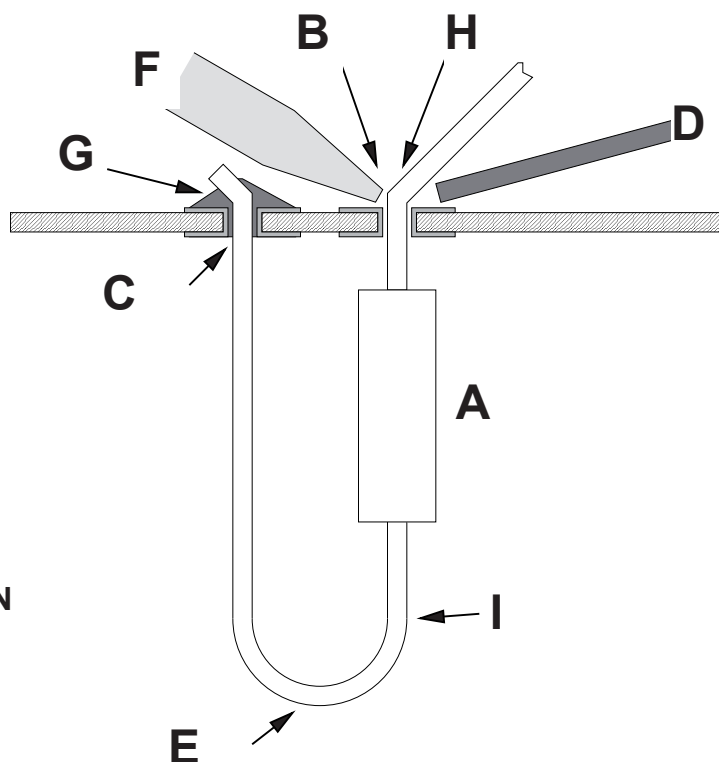
E. Rond buigen geen knik maken

F. Soldeerpunt

G. Glanzende en schone soldering die er niet korrelig mag uitzien TIN goed laten VLOEIEN

H. De draden en de koperbaan worden eerst met de soldeerpunt goed verhit. Dan een klein beetje tin op de koperbaan toevoegen, niet op de soldeerpunt!

I. De buiging niet te dicht bij het onderdeel beginnen



Figuur 3.4.: Het aanbrengen van een goede soldeerverbinding

3.3. Montage en solderen van de onderdelen

Zodra de onderdelen voorbereid zijn, kunnen wij deze door de gemetalliseerde boorgaten van de printplaat steken en - bij onderdelen met slechts twee of drie aansluitingen - deze aan de onderkant van de print direct aan het boorgat iets uiteenbuigen (een hoek tussen ongeveer 30° tot 40° is ruim voldoende), zodat de onderdelen er niet meer uit kunnen vallen.

Bij onderdelen met meer aansluitingen - zoals bijvoorbeeld de voeten van de IC's - is het voldoende om twee diagonaal geplaatste aansluitingen naar buiten te buigen. Het is onpraktisch de pootjes meer dan 45° om te buigen, omdat je een dergelijk onderdeel - als je dit toevallig per ongeluk verkeerd hebt geplaatst - slechts met grote moeite weer van de print kunt halen.

Als het onderdeel eenmaal goed vastzit, verhit je de aansluitingen van dit onderdeel en het metaal rond het boorgat stuk voor stuk met de soldeerboutpunt en voert tegelijkertijd op deze plaats wat soldeermateriaal toe. Dit soldeer smelt daarbij en vloeit in het boorgat. Je moet nu net zolang soldeer laten toevloeien totdat het boorgat helemaal gevuld is (zie daartoe figuur 3.4). Vervolgens neem je eerst de soldeerdraad en daarna de soldeerbout weg, zodat deze soldeerplaats kan afkoelen. Tijdens het afkoelen mag je het onderdeel onder geen voorwaarde bewegen, want elke beweging leidt tot koude soldeerverbindingen en veroorzaakt slechte verbindingen. Als je een onderdeel beweegt, moet je de betreffende aansluiting opnieuw solderen.

Bij aansluitingen, die aan de boven- en onderkant van de print aan kopervlakten zijn aangesloten, is eventueel een iets grotere warmtetoevoer nodig, totdat het soldeertin goed in het boorgat is gevloeid.

Slecht gesoldeerde aansluitingen kun je herkennen aan de kogelvormige soldeerklodders op of naast de aansluiting, respectievelijk aan een mat (en bij loodvrij solderen een zeer mat) oppervlak. Deze slecht gesoldeerde aansluitingen moet je repareren. Bij de voeten en andere onderdelen, die vlak op de printplaat worden gemonteerd, kan men de volgende truc toepassen: het onderdeel wordt eerst met een aansluiting vastgesoldeerd. Dan drukt men van boven voorzichtig op het onderdeel en verwarmt de soldeerverbinding nogmaals (Voorzichtig: Het onderdeel kan daarbij erg heet worden), zodat het onderdeel zichzelf op de print kan laten zakken.

Vervolgens soldeer je alle andere aansluitingen. Ook de allereerst gesoldeerde aansluiting wordt nogmaals nagesoldeerd met wat soldeertin. Nadat een onderdeel is ingebouwd, kun je de uitstekende, overtollige draadstukjes - zonder daarbij aan de draden te trekken - met een kniptangetje kort boven de printplaat afknippen.



Let er bij het afknippen op, dat niemand door de eventueel rondvliegende stukjes draad kan worden gewond. Het is wel duidelijk, dat de aansluitingen van de onderdelen die je op de print plaatst, elkaar nergens mogen aanraken. Daartoe moet je deze aansluitingen eventueel wat opzij buigen.

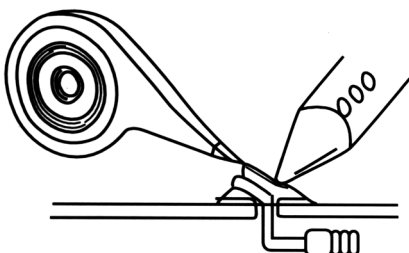
3.4. Het lossolderen van verkeerd ingebouwde onderdelen

Als je toch een onderdeel verkeerd hebt ingebouwd, dan moet je dit weer uitbouwen. Omdat de ASURO - zoals gezegd - een dubbelzijdige printplaat met doorgemetalliseerde boorgaten bevat, is dat geen sinecure.

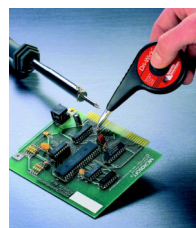
De volgende werkwijze is daarbij in de praktijk bruikbaar gebleken:

Ten eerste brengt men binnen korte tijd het soldeer op alle aansluitingen van dit onderdeel tot vloeien (soms is het nodig daartoe wat vloeibaar soldeertin toe te voegen). Als alle soldeerpunten heet genoeg zijn, trekt men het onderdeel voorzichtig loodrecht met een combinatietang uit de printplaat. Vervolgens bevrijdt men de boorgaten met "desoldeer litze" van het overtollige soldeertin.

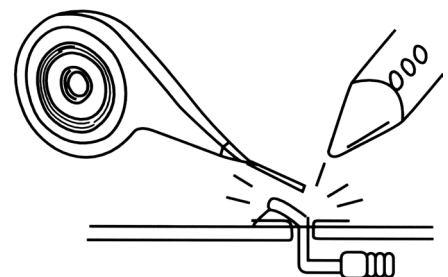
Daartoe leg je de "desoldeer litze" op de te bewerken aansluiting en verhit de litze en het soldeer tegelijkertijd, totdat de litze het soldeertin heeft opgezogen. Daarna neem je de soldeerbout en het litze weer weg. Het kan nodig zijn ook aan de bovenkant van de print wat soldeertin weg te zuigen



1. Het desoldeer-litze op de soldeerverbinding leggen. Dan de litze en het soldeer tegelijkertijd verhitten.



Professioneel desoldeer-litze



2. Daarna neem je de soldeerbout en het desoldeer-litze weer weg, zodra de litze het soldeertin heeft opgezogen.

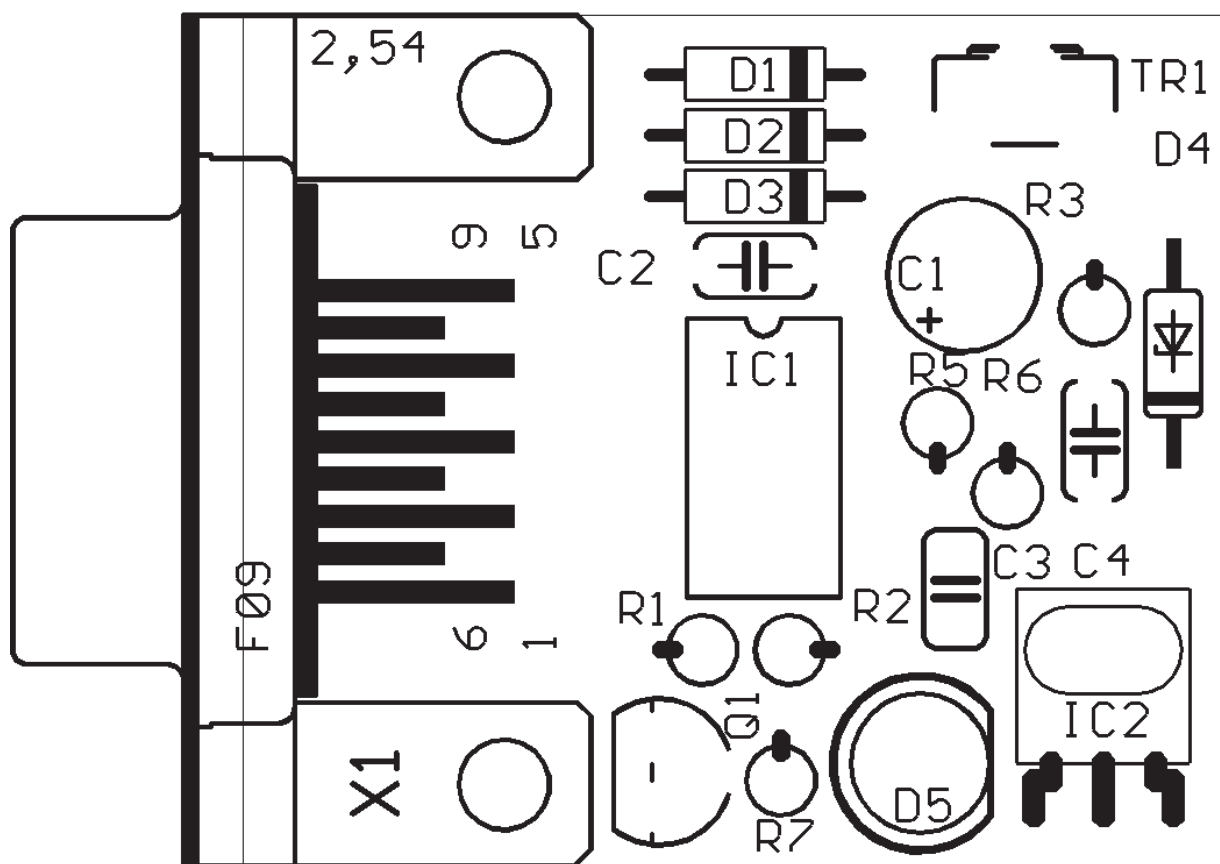
4. De montagefase

Heb je de “Inwijding in de kunst van het solderen” helemaal gelezen? Echt waar? OK, vooruit dan maar!

4.1. Montage van de RS232-Infrarood-Transceiver (Zender/Ontvanger)

LET OP! Zie ook Appendix H (elektronica componenten), voor meer informatie

- IC1: Hier monteren en solderen wij voorlopig alleen de achtpolige IC-voet. Deze is voorzien van een markering ter identificatie van de polariteit en moet met de markering op de print overeenkomen.
- D1, D2, D3: 1N4148, let op de correcte polariteit! Niet verwisselen met de ZPD5.1 of met de BZX55-C5V1 (let op de opdrukinformatie)!
- D4: ZPD5.1 of BZX55-C5V1, let op de correcte polariteit!
Niet verwisselen met de 1N4148 (let op de opdrukinformatie)!
- C2, C4: 100nF keramisch, opdrukinformatie: 104
- C3: 680pF keramisch, opdrukinformatie: 681
- Q1: BC547 (A,B of C) of BC548 (A,B of C)
- R1, R5: 20k Ohm, 5% (rood, zwart, oranje, goud)
- R2: 4.7k Ohm, 5% (geel, violet, rood, goud)
- R3: 470 Ohm, 5% (geel, violet, bruin, goud)
- R6: 10k Ohm, 5% (bruin, zwart, oranje, goud)
- R7: 220 Ohm (rood, rood, bruin, goud)
- C1: 100µF/ minimaal 16V, let op de correcte polariteit! ZIE OOK APPENDIX H!
- TR1: 10k Ohm potmeter
- D5: SFH 415-U IR-LED (zwarte behuizing), let op de correcte polariteit!
Behuizing moet vlak op de printplaat liggen!
- IC2: SFH5110-36 Infrarood-Ontvanger-IC, aansluitingen met tangetje ombuigen!
Let op de correcte polariteit (De kant met het koepeltje moet naar boven wijzen), Opgelet: elektrostatisch gevoelig en - als waarschuwing voor de hobbylassers - ook hittegevoelig!
- X1: 9pol. SUB-D connector. Behuizing moet vlak op de printplaat liggen, ook de montageaansluitingen moeten vastgesoldeerd worden!
- IC1: NE555P plaatsen, let op de polariteitmarkering (pijlje of cirkel)!



Figuur 4.1.: Montagetekening van de RS232- Infrarood-Transceiver

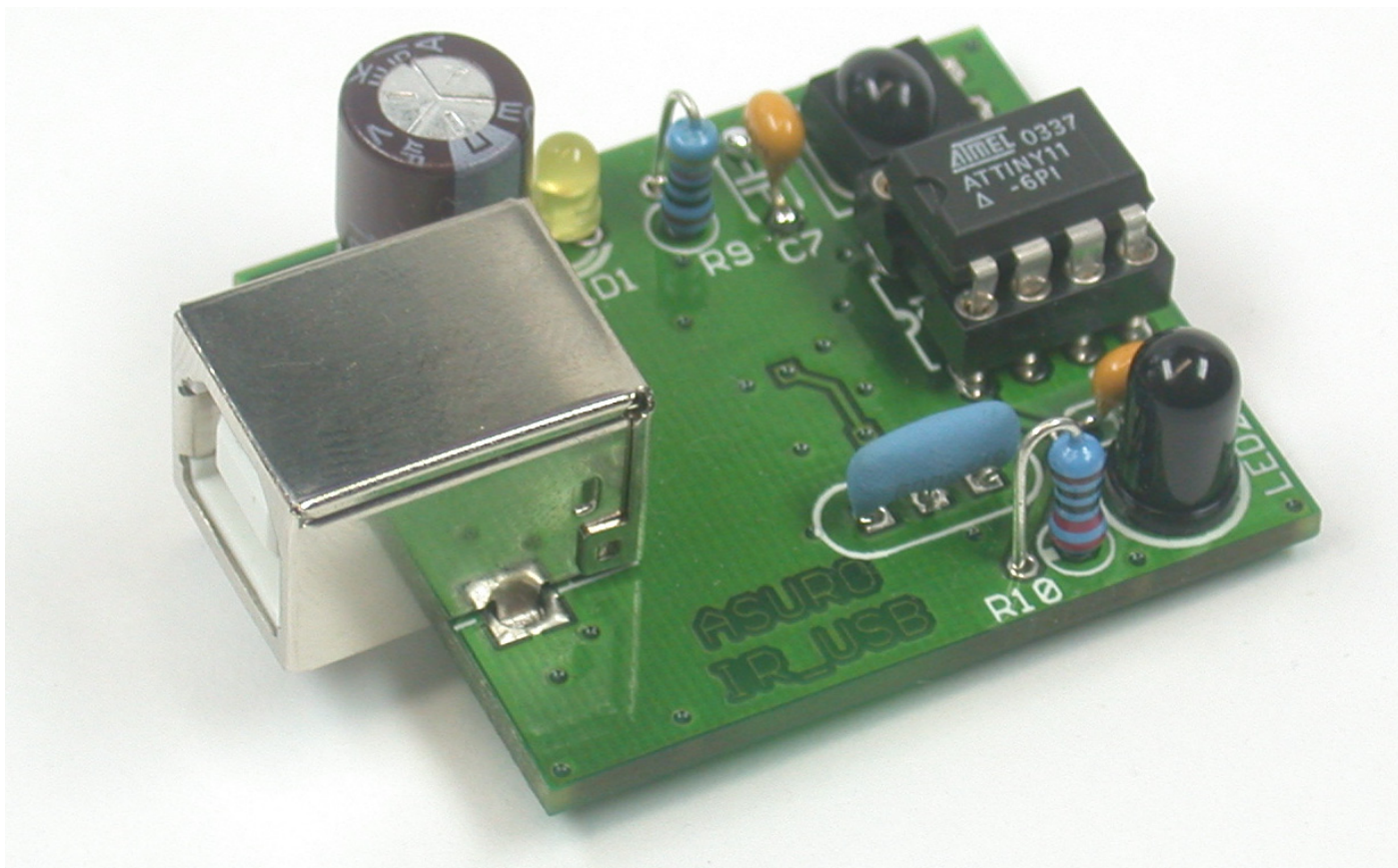
Nu nog even een kritische blik op de soldeerverbindingen, waarbij wij op de glanzende, respectievelijk matte kleur en mogelijke kortsluitingen letten.

Daarbij worden twijfelgevallen of fouten direct gerepareerd.

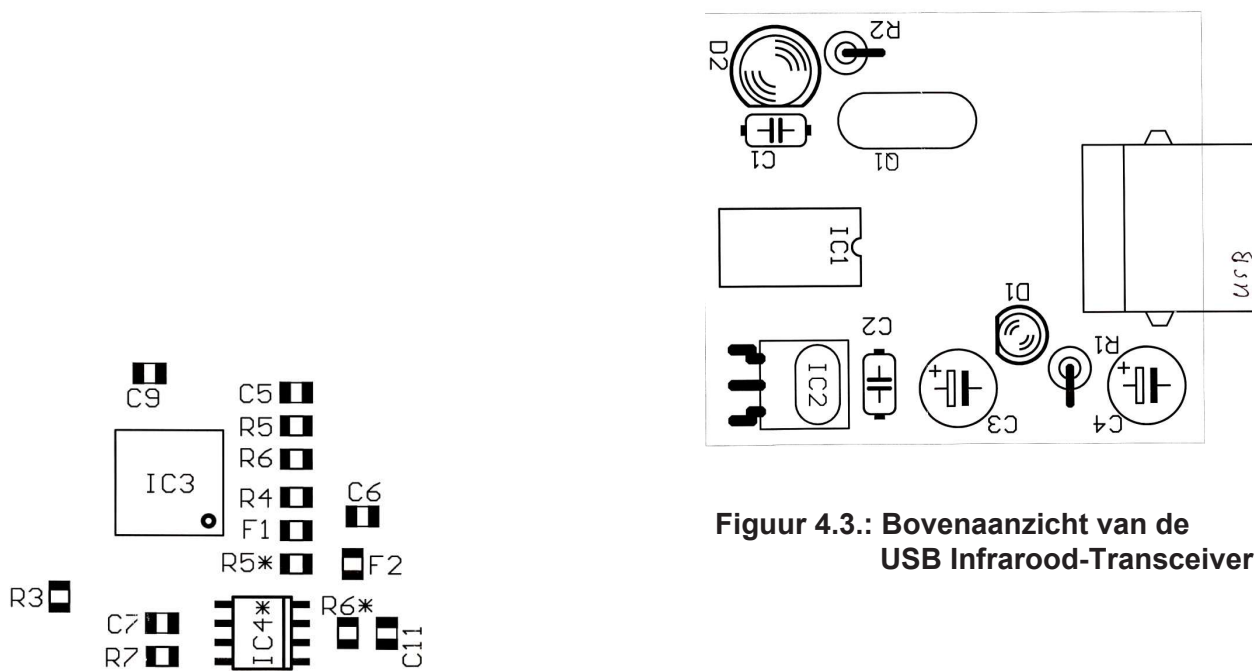
Klaar is Kees!

4.2. Info: kant en klare USB-Infrarood-Transceiver

Optioneel is een USB-IR-Transceiver kant en klaar leverbaar.



Figuur 4.2.: USB Infrarood-Transceiver



Figuur 4.3.: Bovenaanzicht van de USB Infrarood-Transceiver

Figuur 4.4.: Onderaanzicht van de USB Infrarood-Transceiver

4.3. Montage van de ASURO-Printplaat

De twee langste assen, die wij als tweede trap in de transmissie toepassen, worden eenvoudig aan de onderkant van de print vastgesoldeerd of gelijmd.

Solderen is eenvoudiger, omdat correcties dan eenvoudiger zijn en omdat het afkoelen minder tijd kost dan het wachten tot de lijm uitgehard is.

De twee kortere assen liggen aan de bovenkant van de print en bevinden zich dicht bij het centrum van de print. Voor het monteren kun je de assen over de te solderen of te lijmen lengte (maar niet op de plaats waar zich de wielen zullen bevinden) nog even met zeer fijn schuurpapier (korrelgrootte 240 of beter) schoonmaken. Dan vloeit het soldeertin beter en wordt de lijm beter opgenomen. Als je voor het solderen kiest, kun je het beste als volgt werken:

Allereerst monteer je de langste assen. Daartoe draai je de print met de bovenkant naar beneden en leg je de te bewerken as tot op het einde in de uitgespaarde gleuf. De as mag niet scheef liggen en moet vlak op de print rusten. Dan wordt de soldeerbout vertind en druk je met de soldeerbout de as op de printplaat. Zodra de as goed heet is geworden, laat je weer wat meer soldeertin toevloeien over de lengte van de contactplaatsen, zodat de as met de print wordt verbonden.

Als de as helemaal gesoldeerd is, druk je de as met een schroevendraaier nog een tijdje vast op de printplaat en neem je de soldeerbout weg.

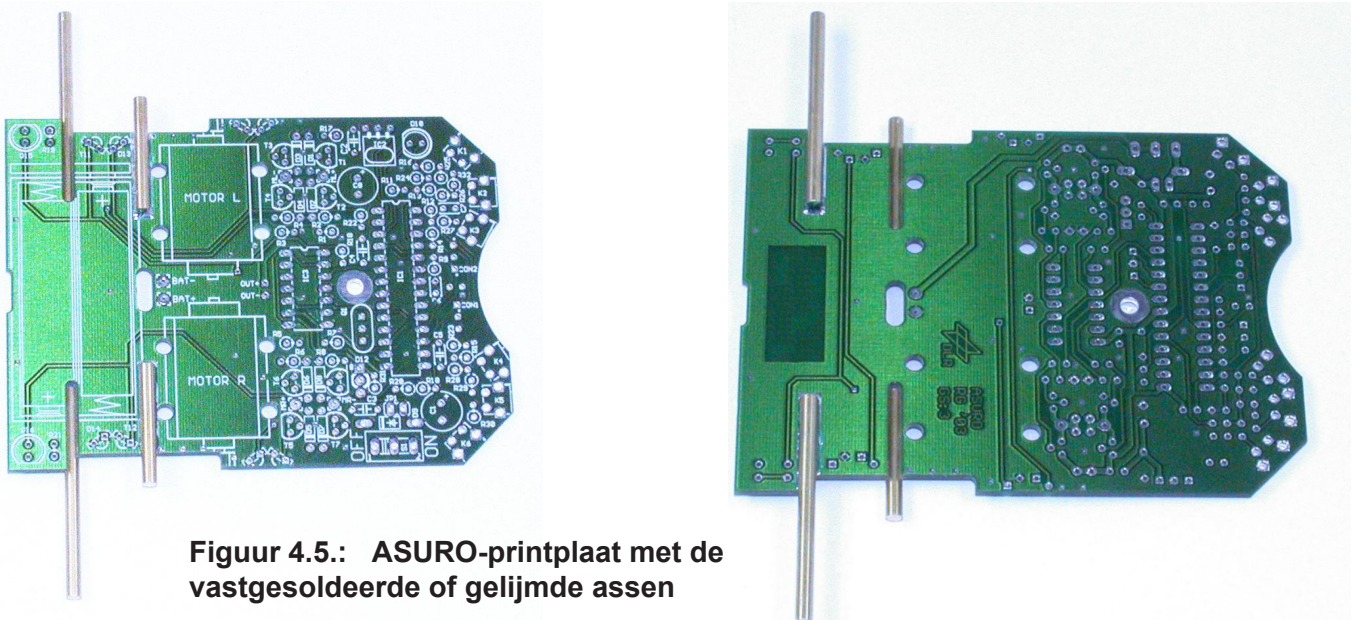
Voor het solderen van de assen kun je de soldeertemperatuur het beste iets verhogen (tot ca. 420°C) en daarbij een (ca. 3mm) brede soldeerpoint kiezen. Voor het solderen van elektronicaonderdelen moet je de temperatuur natuurlijk daarna weer terugdraaien tot ca. 360°C.

Nadat alles weer afgekoeld is, soldeer je de tweede as op de print vast. Vervolgens komen de assen aan de bovenkant van de print aan de beurt. Volg daarbij dezelfde procedure. Figuur 4.5 laat zien hoe de printplaat met de vastgesoldeerde assen eruit ziet.

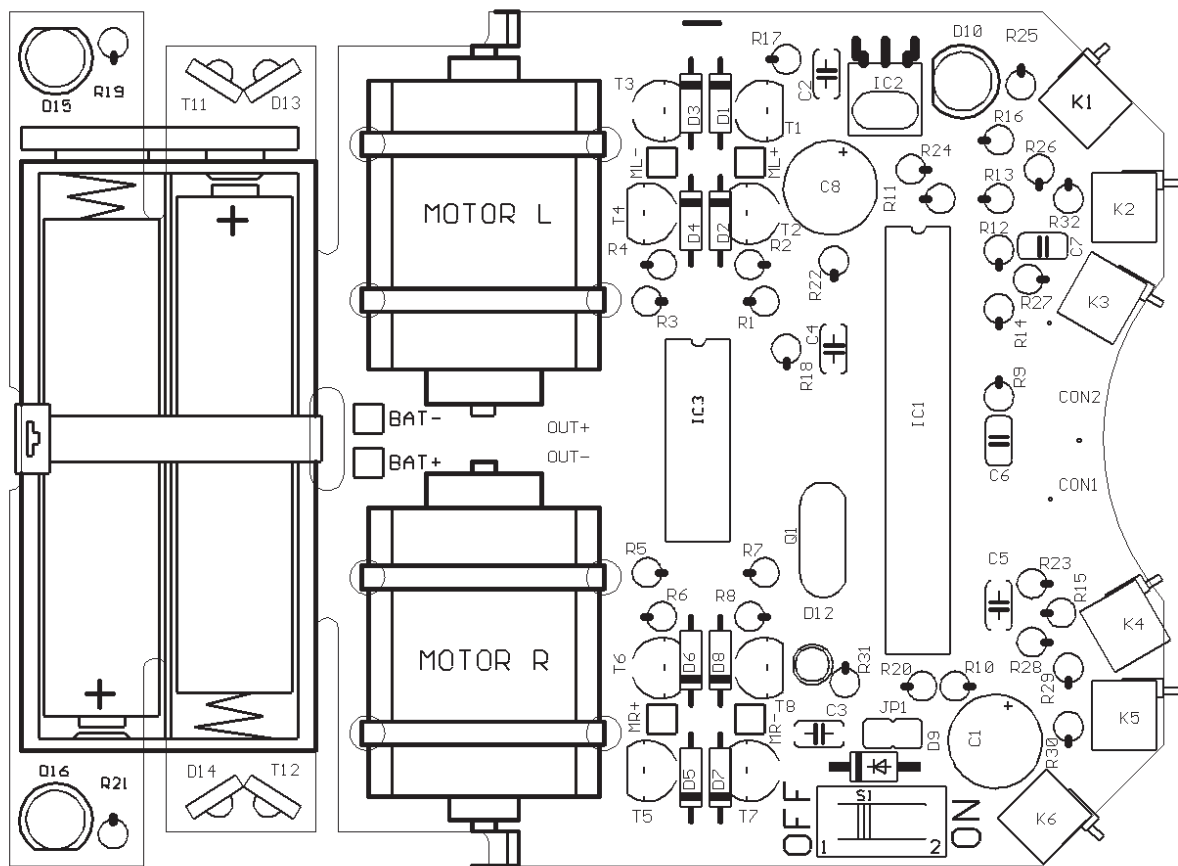
Na het solderen of lijmen van de assen steek je de tandwielen op de assen. De tandjes moeten uiteraard goed in elkaar grijpen en de tandwielen moeten zich gemakkelijk laten draaien. Als dat niet het geval is, zijn de assen wellicht scheef vastgesoldeerd en moeten deze weer gecorrigeerd worden (oefening baart kunst), ofwel er bevinden zich nog enkele soldeerspetters in het draaibereik van de assen. Deze soldeerspetters moeten uiteraard worden verwijderd.

Voor deze correcties gebruik je het beste een fijne vijl of schuurpapier.

Als alles goed past, leg je de tandwielen opzij en monteren wij eerst de overige onderdelen op de print.



Figuur 4.5.: ASURO-printplaat met de vastgesoldeerde of gelijmde assen



Figuur 4.6.: Montagetekening van de ASURO hoofdprint (bovenzijde)

De onderdelen in de aangegeven volgorde monteren:

- IC1: eerst wordt de voet geplaatst; dat is een 28-polige voet (als deze wordt geleverd) of twee 14-polige voeten naast elkaar. Let op de correcte polariteit (de markering van de voet moet met de markering op de print overeenkomen)!
- IC3: in dit geval ook alleen de 14polige voet plaatsen; let op de correcte polariteit (de markering van de voet moet met de markering op de print overeenkomen)!
- K1, K2, K3, K4, K5, K6: sensors; deze moeten vlak liggend op de printplaat worden gemonteerd!
- Q1; Kristal 8MHz
- D1, D2, D3, D4, D5, D6, D7, D8: 1N4148; let op de correcte polariteit!
- D9: 1N4001; let op de correcte polariteit!
- JP1: tweepolige jumper; de korte stekertjes worden gemonteerd, maar je mag de bijbehorende jumper nog niet plaatsen!
- D12: tweekleurige LED, 3mm diameter, drie aansluitingen. Let op de correcte polariteit! (De markering kan verschillend zijn. De kortste aansluitdraad moet in het vierkante contact worden aangesloten)!
- C2, C3, C4, C5: 100nF keramisch; opdruk: 104
- C6, C7: 4,7nF keramisch; opdruk: 472

Elektronica

- T1, T3, T5, T7: BC327-40 of BC328-40
- T2, T4, T6, T8: BC337-40 of BC338-40
- R1, R2, R3, R4, R5, R6, R7, R8, R19, R21: 1k Ohm, 5% (bruin, zwart, rood, goud)
- R9, R16: 220 Ohm, 5% (rood, rood, bruin, goud)
- R10, R17, R22, R31: 470 Ohm, 5% (geel, violet, bruin, goud)
- R11: 100 Ohm, 5% (bruin, zwart, bruin, goud)
- R12: 12k Ohm, 1% (bruin, rood, zwart, rood, bruin)
- R13: 10k Ohm, 1% (bruin, zwart, zwart, rood, bruin)
- R14, R15: 20k Ohm, 5% (rood, zwart, oranje, goud)
- R18, R20: 4,7k Ohm, 5% (geel, violet, rood, goud)
- R23: 1M Ohm, 5% (bruin, zwart, groen, goud)
- R24: 1k Ohm, 1% (bruin, zwart, zwart, bruin, bruin)
- R25, R26, R32: 2k Ohm, 1% (rood, zwart, zwart, bruin, bruin)
- R27: 8,2k Ohm, 1% (grijs, rood, zwart, bruin, bruin)
- R28: 16k Ohm, 1% (bruin, blauw, zwart, rood, bruin)
- R29: 33k Ohm, 1% (oranje, oranje, zwart, rood, bruin)
- R30: 68k Ohm, 1% (blauw, grijs, zwart, rood, bruin)
- C1, C8: Elko 220 μ F 10V of hogere spanning, let op de correcte polariteit!
- IC2: SFH5110-36 Infrarood-Ontvanger-IC. Aansluitingen met tangetje ombuigen!
Let op de correcte polariteit (De kant met het koepeltje moet naar boven wijzen), Opgelet: elektrostatisch gevoelig en - hier weer als waarschuwing voor de hobbylassers - ook hittegevoelig!
- D10: SFH 415-U IR-LED 5mm; zwarte behuizing; Let op de correcte polariteit!
De behuizing moet vlak liggend op de print worden gemonteerd!
- T11, T12: LPT80A, Fototransistor, kleurloze behuizing moet vlak op de print worden gemonteerd.
Let op de juiste polariteit!
- D13, D14: IRL80A, IR-LED, rozekleurige behuizing moet vlak op de print worden gemonteerd.
Let op de juiste polariteit!
- D15, D16: LED 5mm rood, rode behuizing. Let op de juiste polariteit (korte draadaansluiting op de plaats van de markering)
- S1: Aan/Uitschakelaar

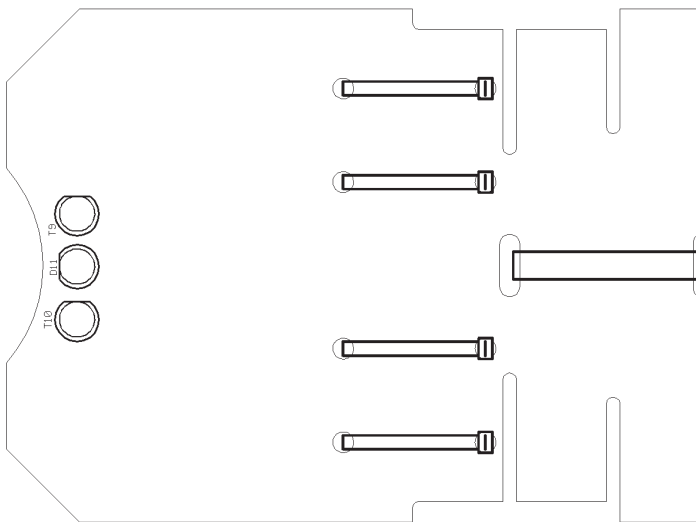
Elektronica

Drie andere onderdelen staan nog op het programma (deze zijn nodig om een lijn te volgen), maar deze worden aan de onderkant van de printplaat gemonteerd en vanaf de bovenzijde vastgesoldeerd (zie figuur 4.7.):

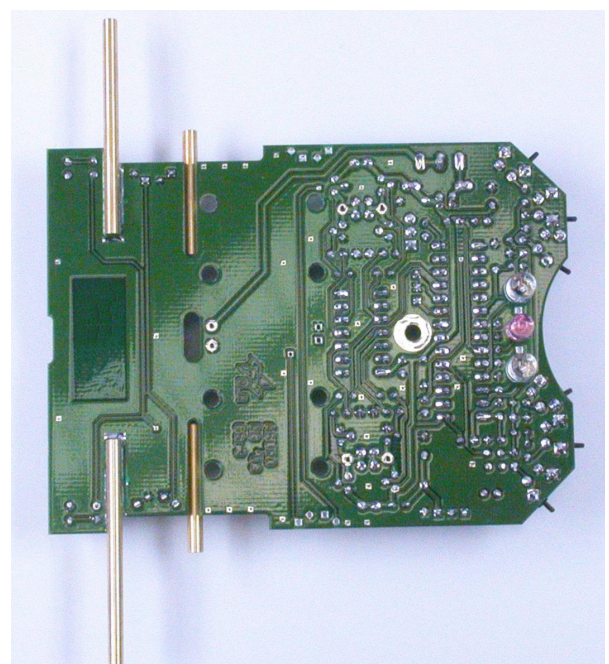
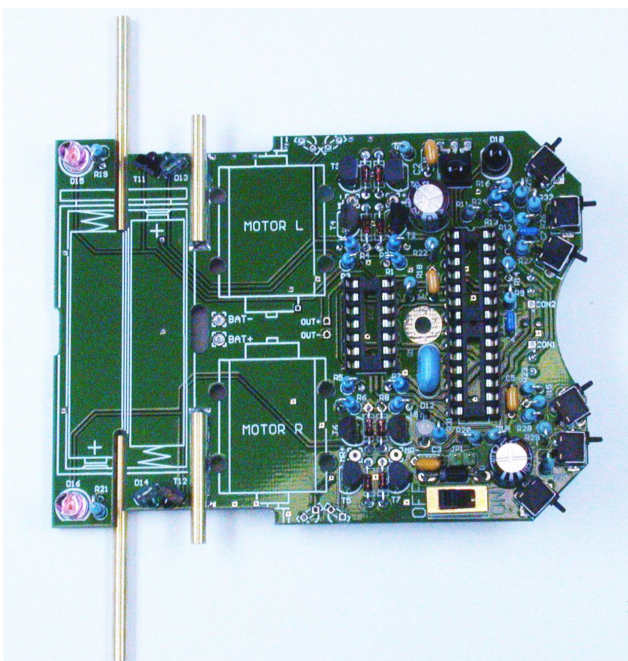
- T9, T10: SFH300, fototransistor 5mm, let op de juiste polariteit! Deze worden op korte afstand van de print ingebouwd.
- D11: LED 5mm rood, rode behuizing. Let op de juiste polariteit (het korte been wordt bij de markering aangesloten)!

Figuur 4.8. laat de gemonteerde printplaat zowel van de boven- als van de onderkant zien.

***Dat was het! Meer elektronica hebben wij niet nodig.
Nu moeten wij nog de elektromechanische en de mechanische onderdelen aanbrengen.***



Figuur 4.7 Montagetekening voor de ASURO-printplaat aan de onderzijde



Figuur 4.8. Gemonteerde ASURO-printplaat zowel van de boven- als van de onderkant.

4.4. De montage van de motoren

Als wij de montagewerkzaamheden aan de ASURO-print hebben afgesloten, moeten wij alleen nog maar de motoren van kabels voorzien en provisorisch bevestigen.

Voor de aansluiting van de motoren heb je een ongeveer 70mm lange, zwarte en rode kabel met gestripte en vertinde uiteinden nodig. Als deze kabels nog niet in deze vorm ter beschikking staan, moet je deze over een lengte van ca. 4mm lang strippen, dan in elkaar draaien en vervolgens met soldeer vertinnen. Daartoe moet je de einden met wat soldeertin tegen de hete soldeerbout houden. Eventueel moet je met een kniptangetje overtollige soldeerresten van de einden verwijderen. De rode kabel wordt nu aan de motoraansluiting met de rode punt of met een plusmarkering aangesloten. De zwarte kabel aan de andere aansluiting van de motor. De aansluitaders van de motoren worden elk afzonderlijk in elkaar gedraaid (dit twisten is niet echt nodig, maar biedt wat voordelen bij de bestrijding van elektromagnetische storingen en een netjes verstrengelde kabel ziet er toch ook beduidend beter uit...).

De rode aansluitkabel van de linker motor wordt op "ML+" en de zwarte op "ML-" aangesloten en gesoldeerd.

De rode aansluitkabel van de rechter motor wordt op "MR+" en de zwarte op "MR-" aangesloten en gesoldeerd.

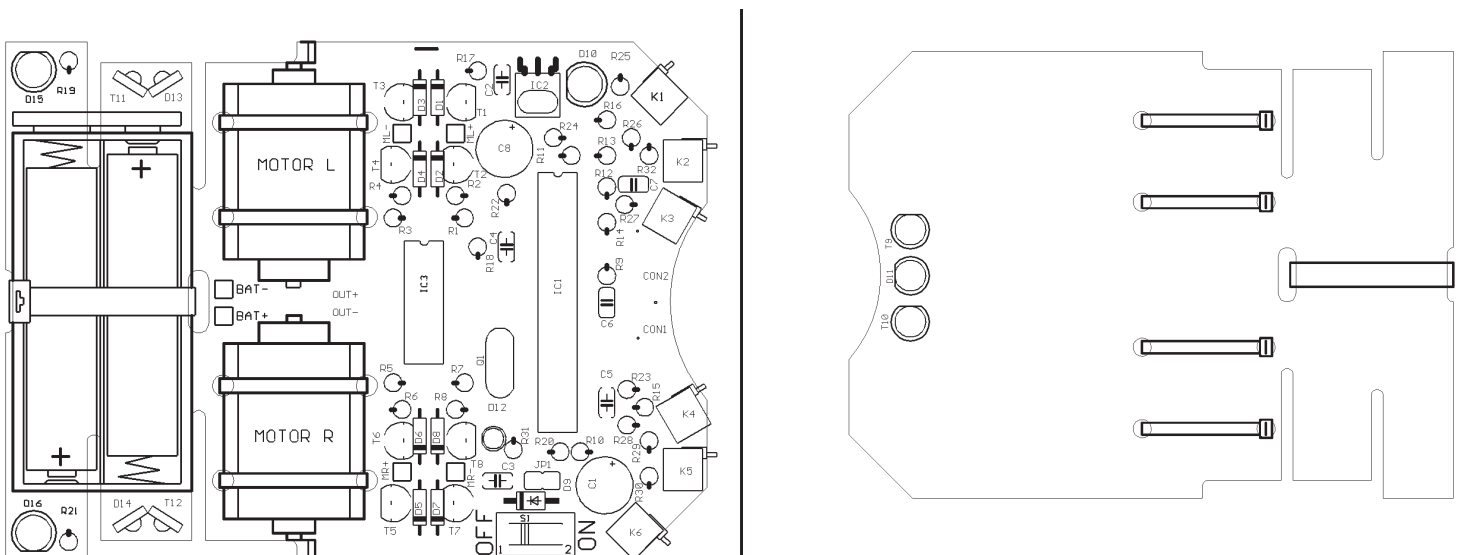
Nu moeten wij de motoren nog provisorisch op de print bevestigen. Daartoe schuif je kabelbinders door de gaten, die zich op de print naast de motoren bevinden, en wel zo, dat de kopstukken van de kabelbinders zich steeds aan de onderkant van de print bevinden en de motoren omsluiten.

4.5. De stroomvoorziening



Als de ASURO met batterijen moet worden gevoed, moet jumper JP1 beslist geopend worden! Bij accuvoeding moet je jumper JP1 sluiten. Een verkeerd aangesloten accuvoeding met gesloten jumper JP1 veroorzaakt onherstelbare schade aan de elektronische schakeling!

Het batterijcompartiment wordt (nog zonder batterijen) met een rode kabel op BAT+ en met de zwarte kabel op BAT- aangesloten. Dan controleer je of de hoofdschakelaar op OFF staat en je legt de vier batterijen respectievelijk vier accu's met de juiste polariteit in het batterijcompartiment. Controleer nogmaals de correcte polariteit van de batterijen en kabels! Het batterijcompartiment wordt nu meteen of na de inbedrijfname met de grote kabelbinder (die weer kan worden losgemaakt) door de twee gaten in de print vastgezet.



5. Inbedrijfname en test

Nadat je eindelijk alles hebt opgebouwd, begint nu je eerste echte ASURO-rit. Maar voordat je de motoren mag starten, moet je nog wel even de ingebouwde foutjes zoeken, vinden en repareren zonder daarbij al te veel schade te veroorzaken.

5.1. RS232-Infrarood-Transceiver

Deze inbedrijfname geldt alleen voor de RS232-IR-Transceiver.

Je moet deze module eerst volledig testen, omdat de perfect werkende transceiver daarna voor de zelftest van de ASURO nodig is.

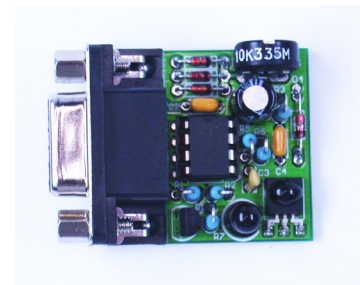
Voor de test sluit je de RS232-IR-Transceiver met het meegeleverde seriële interfacekabel op een vrije seriële PC-interface aan.

Vervolgens start je het terminalprogramma van Windows: "Hyperterminal" (in Linux heet dit programma bijvoorbeeld "Minicom"). In Windows vind je "Hyperterminal" onder Start --> Programma's --> Bureau-accessoires --> Communicatie --> Hyperterminal

Terminalprogramma's stammen in feite nog uit de tijd van de modems of zelfs nog voor die tijd. Destijds gebruikte men de seriële interfaces intensief voor de onderlinge communicatie tussen computers. Tegenwoordig is dit gebruik vaak beperkt tot het eenvoudige aanmelden op een andere computer in het Internet.

Nadat je hyperterminal hebt gestart vraagt het programma je naar een naam voor de nieuwe verbinding. Op deze vraag kun je met ASURO antwoorden en een willekeurig symbool kiezen. In het nu volgende venster kies je bij "Verbinden via:" de COM-interface, waarop je de transceiver van tevoren hebt aangesloten. Na het aanklikken van "OK" kies je:

- Bits per seconde: 2400
- Databits: 8
- Pariteit: geen
- Stopbits: 1
- Protocolsturing: geen



Vervolgens weer afsluiten met "OK".

Nu houd je de IR-Transceiver ca. 10 cm boven een wit blad papier. De onderdelen moeten daarbij in de richting van het papier gehouden worden. Druk nu een aantal lettertoetsen in op het toetsenbord van je PC en controleer of deze letters tegelijkertijd op het terminalprogramma worden weergegeven.

De IR-Transceiver zendt voor elke toetsbeweging een infraroodsignaal uit via de IR-Diode (D5). Het signaal wordt op het blad papier gereflecteerd, ontvangen en gedetecteerd in de IR-ontvanger-IC (IC2) en aan de PC doorgegeven. Als er aan de ontvanger helemaal geen of foutieve signalen aankomen, kun je met een kleine schroevendraaier voorzichtig de trimmer tussen de linkse en rechtse aanslag bijstellen en steeds weer enkele toetsen indrukken, totdat het terminalprogramma de juiste lettersymbolen weergeeft.

Werkt het niet, zoals het hierboven beschreven staat? Pech, want dan is er ergens een foutje ingeslopen, dat wij eerst moeten repareren (zie hoofdstuk 6.1).

Voor alle zekerheid kun je de IR-Transceiver eerst weer wegnemen en nogmaals enkele lettertoetsen indrukken. Dan mogen er geen lettertekens meer op het terminalprogramma verschijnen.

5.2. USB-Infrarood-Tranceiver

Deze inbedrijfname geldt alleen voor de USB-Infrarood-Transceiver. Opgelet! Zonder afschermbehuizing is de USB-Infrarood-Transceiver gevoelig voor elektrostatische ontladingen. Voor het werken aan deze module moet je je om schade te voorkomen even aan een metalen object (de verwarming, waterleiding of een PC-behuizing) ontladen. Eventueel kun je de transceiver natuurlijk ook in een voor infrarood licht doorzichtige behuizing inbouwen.

5.2.1 Windows

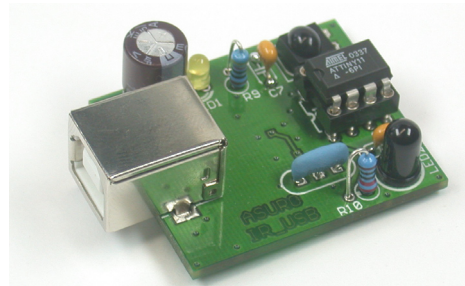
Je sluit de USB-transceiver met behulp van de USB-kabel op een vrije USB-aansluiting van je PC aan, waarop je computer meldt:

“Nieuwe Hardware gevonden: **AREXX ASURO USB-IR-Transceiver**”

Vervolgens moet je het USB-stuurprogramma van de ASURO-CD installeren. Als het stuurprogramma niet automatisch gevonden wordt, dan kun je de map (D: betekent in dit geval de CD-ROM-drive) D:\Windows\USB_Driver” kiezen. Het is mogelijk, dat je daartoe administratorrechten op je PC nodig hebt. In dat geval moet je je afmelden en je als administrator opnieuw op de PC aanmelden. Daarna kun je een stuurprogramma installeren, waarmee je onder Windows de USB-Transceiver als een gewone seriële interface kunt aanspreken.

Nadat deze installatie zonder problemen is verlopen, kun je ook in dit geval voor een systeemtest het terminalprogramma “Hyperterminal” starten. Je antwoordt daartoe op de vraag naar een verbindingnaam met ASUROUSB en kiest een willekeurig symbool. In het nu volgende venster kies je bij “Verbinden via:” de laatste beschikbare COM-interface. Na het aanklikken van “OK” kies je:

- Bits per seconde: 2400
- Databits: 8
- Pariteit: geen
- Stopbits: 1
- Protocolsturing: geen



Vervolgens weer afsluiten met “OK”.

Nu houd je de transceiver ca. 10 cm boven een wit blad papier. De printzijde met de LED moet daarbij in de richting van het papier worden gehouden. Als je de transceiver zonder behuizing gebruikt, mag je dit printplaatje alleen aan de stekker of aan de rand van de print vasthouden, zodat je de werking van de schakeling niet stoort. Druk nu een aantal lettertoetsen op het toetsenbord van je PC. Daarbij moet de gele LED op de print oplichten en de ingetoetste letters tegelijkertijd op het terminalprogramma worden weergegeven.

Werkt het niet? Dan moet je in hoofdstuk 6.2 verder lezen.

Als alles goed afloopt, kun je de inbedrijfname van de ASURO-print voortzetten.

5.2.2 Linux

De USB-transceiver wordt met de USB-Kabel op een vrije USB-stekker aangesloten. De PC reageert nu met een kort “Piep”-signaal, zodra Linux de transceiver heeft geregistreerd. Om te testen of de PC de transceiver correct heeft herkend, kun je de bijbehorende registratie in het proc-bereik controleren:

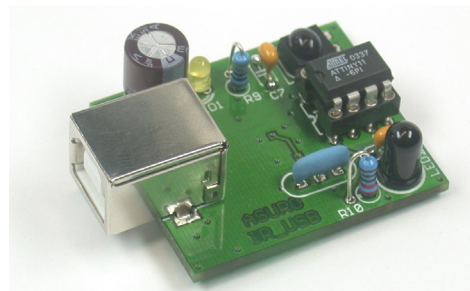
```
foo@bar: /> cat /proc/tty/driver/usb-serial
```

Een randapparaat, dat een output moet kunnen leveren, behoort minimaal over de volgende parameters te beschikken (waarbij in plaats van de “0:” eventueel ook “1:”, “2:”, enzovoorts kan staan):

```
usbserinfo:1.0 driver:v1.4  
0: module:ftdi_sio name:”FTDI 8U232AM Compatible” vendor:0403 product:6001  
num_ports:1 port:1 path:usb-00:11.2-1
```

In een test configureer je Minicom op het interface /dev/ttyUSB0 (respectievelijk 1, 2 enzovoorts...) en met behulp van de volgende parameters:

- Bits per seconde: 2400
- Databits: 8
- Pariteit: geen
- Stopbits: 1
- Protocolsturing: geen



Vervolgens weer afsluiten met “OK”.

Daartoe moet je eventueel over root rechten beschikken.

Ook moet je misschien aan de User (gebruiker) of aan de gewenste gebruikersgroep nog lees- en schrijfrechten voor het randapparaat toekennen. Dat doe je met de commando’s:

```
chmod u+rw /dev/ttyUSB0 (respectievelijk 1, 2...) of  
chmod g+rw /dev/ttyUSB0 (eveneens met root rechten)
```

Nu houd je de transceiver ca. 10 cm boven een wit blad papier. De transceiver moet daarbij met de LED-zijde naar beneden in de richting van het papier wijzen. Als je de transceiver zonder behuizing gebruikt, mag je dit printplaatje alleen aan de stekker of aan de rand van de print vasthouden, zodat je de werking van de schakeling niet stoort. Druk nu een aantal lettertoetsen in op het toetsenbord van je PC. Daarbij moet de gele LED op de print oplichten en de ingetoetste letters tegelijkertijd op het terminalprogramma worden weergegeven.

Werkt het niet? Dan moet je in hoofdstuk 6.2 verder lezen.

Als alles goed afloopt, kun je de inbedrijfname van de ASURO-print voortzetten.

5.3. Inbedrijfname van de ASURO-print



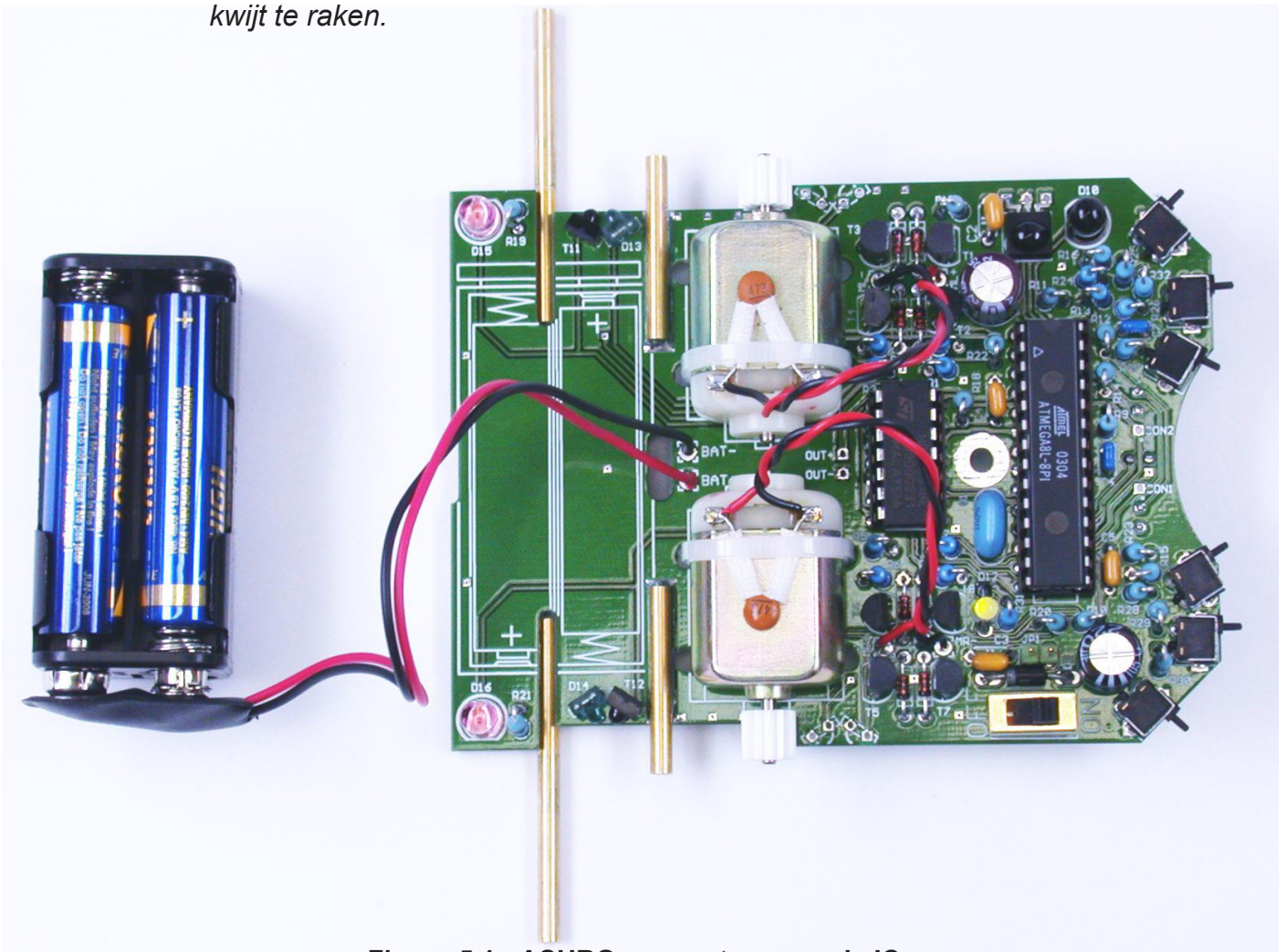
De processor (IC1) is tot op dit moment nog steeds niet ingebouwd!

Nu zet je je schrap en zet de hoofdschakelaar moedig op ON. Beide Back-LED's (D15, D16) moeten nu zwak oplichten. Als dat niet gebeurt, moet je de hoofdschakelaar onmiddellijk op OFF terugschakelen en hoofdstuk 6.3 doorlezen.

Is dit experiment goed gegaan? Dan de schakelaar op OFF terugzetten en IC1 (de processor) en IC3 (de AND poorten) insteken (zie figuur 5.1). Wellicht moet je de aansluitingen van de IC's nog voorzichtig wat ombuigen om alle pootjes op de bijpassende gaten in de voetjes te kunnen voeren. Dat ombuigen lukt meestal het beste, als je het IC aan de zijkant vastpakt en de rij pootjes allemaal tegelijk voorzichtig tegen een tafelblad drukt.



De processor IC1 (ATmega8) en het poortenarray IC3 (CD4081) zijn allebei gevoelig voor elektrostatiche ontladingen. Dit betekent, dat een gewone aanraking deze schakelingen al onherstelbaar kan beschadigen, als de gebruiker zich van tevoren bijvoorbeeld door het rondlopen met rubber zolen op een vaste vloerbedekking elektrostatich had opgeladen. Wij geven je de raad, dat je vóór elke manipulatie met deze onderdelen een aardingarmband omdoet of dat je tenminste vóóraf de metalen behuizing van een geaard apparaat of bijvoorbeeld een verwarming even vastpakt om een eventueel aanwezige elektrostatiche lading kwijt te raken.



Figuur 5.1.: ASURO na montage van de ICs

Als je de ASURO met accu's wilt voeden, moet je de jumper (J1) plaatsen. De IC-markeringen moeten met de markeringen op de (hopelijk ook correct ingebouwde) voetjes overeenstemmen. De processor is al tijdens het productieproces van een zelftestprogramma voorzien en begint na het inschakelen automatisch met een test van alle onderdelen. Om daarbij storingen te vermijden, is het raadzaam om vóór het inschakelen de navolgende tekstpassage volledig door te lezen en dan weer naar dit punt van de inbedrijfname terug te keren.

Vanaf nu lossen wij het startschot, zetten de hoofdschakelaar op ON en verliezen de ASURO niet meer uit het oog.



Als je hyperterminal (Windows) of minicom (Linux) hebt gestart, de IR-Transceiver hebt geplaatst en je de ASURO in de gaten houdt, kun je de afloop van de zelftest op de PC-monitor meelesen.

5.3.1. LED Indicatoren

De Status-LED (D12) licht kort oranjegeel op en de "Back-LED's" (D15, D16) gloeien eveneens op, maar niet zo fel. Als je deze verschijnselen niet waarneemt, moet je de hoofdschakelaar direct op OFF schakelen en de fouten aan de hand van de informatie in hoofdstuk 6.3.3. repareren. De nu doorlopen startfase is de zogenaamde "bootphase" van de ASURO. Daarin test het systeem achtereenvolgens telkens drie seconden lang alle displayelementen, en wel in de onderstaande volgorde:

- Status-LED (D12) groen
- Status-LED (D12) rood
- Front-LED (D11) aan de onderkant van de ASURO
- Back-LED (D15) links
- Back-LED (D16) rechts
- Alle displayelementen tegelijkertijd

Als er onverhoopt een fout optreedt, moet je de ASURO onmiddellijk uitschakelen en de fout repareren (zie hoofdstuk 6.4), omdat alle zojuist gecontroleerde displayelementen voor de nu volgende tests gebruikt worden en dus foutloos moeten werken.

Fototransistoren (T9, T10)

Na de LED displaytest moet Status-LED (D12) groen oplichten. Dit is een duidelijk teken, dat de fototransistors, die wij voor het volgen van een lijn nodig hebben en die zich daarom aan de onderkant van de ASURO bevinden, nu getest worden. De test neemt ongeveer 10 seconden in beslag. Zodra er licht valt op een van de fototransistors (T9, T10) moet de bijbehorende Back-LED (D15, respectievelijk D16) oplichten en bij het uitschakelen van de lichtbron ook weer uitgaan (fototransistor rechts (T10) --> Back-LED rechts (D16); Fototransistor links (T9) --> Back-LED (D15) links).

Het is mogelijk, dat de Back-LED in uitgeschakelde toestand niet helemaal uitdooft, maar nog wat gloeit. Dit is normaal. Als er een fout optreedt, kun je gewoon met de zelftest doorgaan en de fout later repareren.

Schakelaar (front)

De ASURO staat stil en alle indicatoren zijn uit. Dat is een goed teken! Het systeem test nu (gedurende ca. 15 seconden) de schakelaars (drukknoppen). Je moet nu wat op de schakelaars intoetsen. Hopelijk gebeurt er nu ook wat. Als alles normaal reageert, moet de ASURO zich als volgt gedragen:

- K1 --> Status-LED (D12) licht op in de kleur groen
- K2 --> Status-LED (D12) licht op in de kleur rood
- K3 --> Front-LED licht op (D11) aan de onderkant van de robot
- K4 --> Back-LED links (D15)
- K5 --> Back-LED rechts (D16)
- K6 --> De Motor aan de linkerkant begint te draaien

(Als de motor niet draait, kun je toch met de zelftest doorgaan. In een aparte test zullen wij nog elke motor afzonderlijk onder de loep nemen en ook in staat zijn om fouten in de motorsturing (zie hoofdstuk 6.8) te ontdekken. Bij het activeren van meerdere toetsen tegelijkertijd zullen ook combinaties van de bijbehorende signalen worden opgewekt. Bij fouten mag je de zelftest voortzetten en de fouten later repareren.

De lichtgevoelige sensor (Hodometer)

De lijnvolger-LED (D11) aan de onderkant van de ASURO licht op en geeft aan, dat de volgende test met een lengte van ongeveer 15 seconden begint. De lichtgevoelige sensoren voor de hodometer worden gecontroleerd. Door een wit papier voor een van de lichtgevoelige sensoren te houden kun je de schakeling laten reageren door het oplichten van de bijbehorende Status-LED. Zodra je het papier voor de sensor (T11) aan de linkerkant houdt, licht de Status-LED (D12) groen op en als je het bij de sensor (T12) aan de linkerkant houdt, moet daarentegen de Status-LED (D12) rood oplichten. Als je het papier wegneemt, dooft de LED. Een overgang van licht naar donker kan op deze wijze geregistreerd worden en de sensorschakeling voor de hodometrie is zover in orde. Als er een fout optreedt, kun je met de zelftest doorgaan. Fouten kun je ook later repareren.

De Motoren

Beide Back-LEDs (D15, D16) lichten op als signaal, dat nu de op een na laatste test op het programma staat. Ook deze test duurt ca. 15 seconden. De motoren worden grondig gecontroleerd. Het systeem brengt eerst de linkse motor van stilstand tot op het hoogste toerental voorwaarts in beweging en vervolgens weer tot stilstand. Dan wordt de richting omgekeerd en weer van stilstand op hoogtoeren en daarna terug naar stilstand geschakeld. In een tweede cyclus wordt de rechtse motor aan dezelfde test onderworpen. Ter afsluiting volgt nog een test met beide motoren tegelijkertijd. Tot slot volgt ook hier de inmiddels bekende regel, dat je bij fouten met de zelftest kunt doorgaan en dat fouten ook achteraf kunnen worden gerepareerd.

IR-Transceiver

Zodra de Status-LED geel blinkt, is de laatste test (eveneens ca. 15sec) al in volle gang. De IR-Transceiver verzendt, respectievelijk ontvangt data. Om deze signalen te kunnen detecteren, moet je de afgebouwde IR-Tranceiver op je PC aansluiten en een terminalprogramma zoals bijvoorbeeld het Windows terminalprogramma "Hyperterminal" opstarten. Ter configuratie van dit programma kun je dezelfde parameters nemen als voor de test van de IR-Tranceiver.

Op de ontvangst van de symbolen antwoordt ASURO met een lettersymbool, dat in het alfabet op het ontvangen symbool volgt. Als er in een vooringestelde periode geen gegevens binnenkomen, zendt ASURO een 'T'-symbool. Bij ieder uitgezonden symbool schakelt het systeem na de groene Status-LED een rode Status-LED aan. Deze combinatie zorgt ervoor, dat onze ogen deze afwisseling als een geel blinksignaal interpreteren.

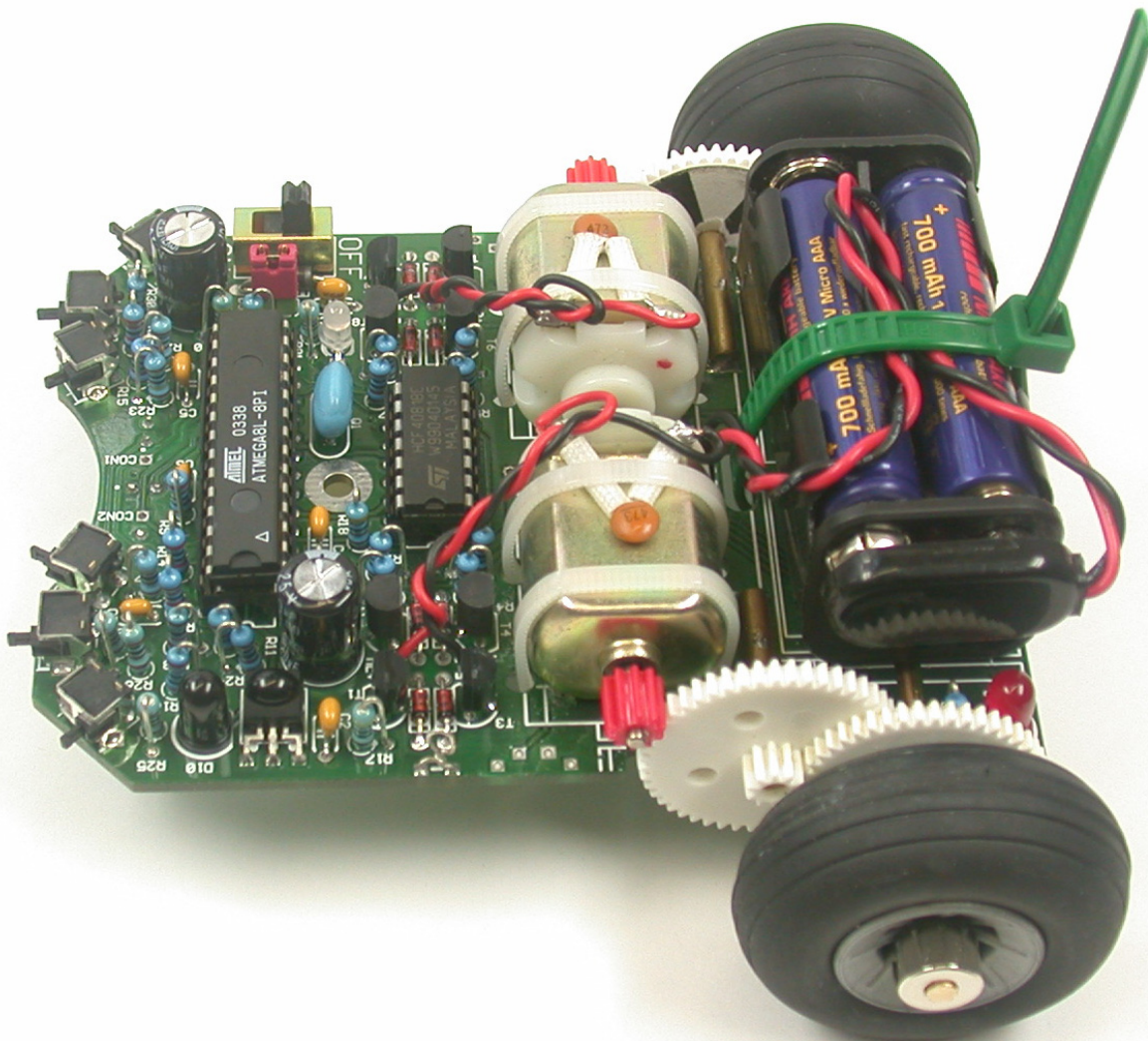
Als de ASURO zich binnen het zichtcontact (tot op maximaal ca. 50 cm afstand) van de Infrarood-Transceiver bevindt, zal er op het beeldscherm van het terminalprogramma regelmatig een 'T' verschijnen, respectievelijk de op het PC-toetsenbord aangeslagen letter verschijnt éénmaal (als het door de transceiver uitgezonden en gereflecteerd signaal) op het scherm, gevolgd door de in het alfabet daarop volgende (en door de ASURO uitgezonden) letter, bijvoorbeeld:

Je drukt de toets "e" => het terminalprogramma schrijft "ef"
Je drukt de toets "j" => het terminalprogramma schrijft "jh"
Je drukt de toets "3" => het terminalprogramma schrijft "34"

In twijfelgevallen s.v.p. hoofdstuk 6.9 bestuderen.

5.3.3. Klaar?

Als er een fout wordt geconstateerd, moet je de batterijen verwijderen en de fout met behulp van hoofdstuk 6 repareren. Vervolgens moet je de zelftest herhalen, als er tenminste geen onderdeel is beschadigd door de fout. Dan pas is de ASURO volledig bedrijfsklaar en kun je er ook met zekerheid vanuit gaan, dat alle daarna gevonden fouten allereerst in de software en niet in de hardware mogen worden gezocht. Als later in bedrijf ooit weer eens een zelftest nodig mocht zijn, nadat er al andere programma's op de ASURO zijn gestart, dan kun je het zelftestprogramma nogmaals laden met behulp van het bestand "SelfTest.hex" van de CD.



6. Het zoeken en repareren van fouten

Meer info vindt je op www.arexx.com en www.roboternetz.de

6.1. De RS232-IR-Transceiver werkt niet!

6.1.1. De ingetoetste letters en de weergegeven letters verschillen

Draai net zolang aan trimmer TR1, totdat de ingetoetste letters en de weergegeven letters overeenkomen.

6.1.2. Het terminalprogramma geeft helemaal geen lettersymbolen weer

Is de Timer-IC (IC1) met de juiste polariteit geplaatst (d.w.z. de markering wijst in de richting van de drie diodes)? Neem nu de infraroodafstandsbediening van een willekeurig apparaat (videorecorder, TV, tuner, enzovoort...). Houd dan de afstandsbediening in de richting van de IR-transceiver en druk een paar toetsen in. Als er dan niets gebeurt, moet je nogmaals de COM-Port-parameters controleren. Als het terminalprogramma gekke symbolen weergeeft, werkt de ontvangersectie (IC2, R3, C4, D4, T1). Dan moet je alle andere onderdelen controleren.

6.1.3. Het werkt nog steeds niet

Controleer nu alle ingebouwde onderdelen eerst op de juiste polariteit en waarde (zie figuur 4.1). Daarna ook de soldeerpunten op slechte contacten en kortsluitingen nog even controleren. Is er misschien ergens een koperbaan of een aansluiting losgeraakt? Als je bij deze speurtochten niets vindt, moet je met het schema (zie appendix B) en een geschikt meetinstrument (een multimeter of oscilloscoop) het defecte onderdeel zoeken. Mogelijke kandidaten voor defecten zijn vaak de onderdelen IC1, IC2, Q1, D4.

6.2. De USB-Infrarood-Transceiver werkt niet

6.2.1 Windows

Is het stuurprogramma (de driver) goed geïnstalleerd? Soms worden wel eens andere COM-Portnummers gebruikt in plaats van de laatste. In dat geval kun je in hyperterminal eenvoudig enkele andere ports activeren en de test dan herhalen. In de systeemsturing kun je controleren, welke port werkelijk ingeschakeld is.

2.2 Linux

In Linux is het raadzaam de USB-Transceiver even te verwijderen, een tijdje te wachten en dan weer in te steken. Als de melding dan nog steeds niet in de proc-map verschijnt, moet je misschien een nieuwere Linux-kernel installeren.

6.3. Back-LED's (D15,D16) lichten na het inschakelen niet op!

6.3.1. Geen van beide Back-LED's licht op

Omdat deze LED's maar zwak oplichten, moet je goed nakijken en de omgeving eventueel tegen vals licht afschermen. Als er dan nog steeds niets te zien is, moet je de volgende punten controleren:

- Zijn alle 4 batterijen met de juiste polariteit geplaatst. Zijn de batterijen/accu's vol?
- Is de kabelaansluiting voor de batterijen met de juiste polariteit ingebouwd? (rood Bat+ / zwart Bat-)
- Is de diode D9 (1N4001) met de juiste polariteit ingebouwd?
- Heeft R22 de juiste waarde? 470 (geel,violet,bruin,goud)
- Misschien moet je ook nog even de volgende weerstanden controleren: R18 4,7K Ω (ge,vio,ro,gd), R19 1K Ω (br,zw,ro,gd), R20 4,7K Ω (ge,vio,ro,gd), R21 1K Ω (br,zw,ro,gd)

6.3.2. Slechts een van beide LED's licht op

Zijn beide diodes (met de rozekleurige behuizing) D13 (links), D14 (rechts) en de fototransistors (doorzichtige behuizing) op de juiste plaats (zie figuur 4.3) en met de juiste polariteit ingebouwd?

Hebben de volgende weerstanden de juiste waarde: R18, R19 (links) en R20, R21 (rechts)?

4,7K (geel, violet, rood, goud)
1K (bruin, zwart, rood, goud)

Zijn alle onderdelen op de ASURO-print correct ingebouwd? Controleer bij de weerstanden de informatie, die op de print is gedrukt!

6.3.3. Na de start licht Status-LED (D12) niet in twee kleuren op

Status-LED licht helemaal niet op => zie hoofdstuk 6.4!

Status-LED brandt onregelmatig => de batterijspanning is te laag => batterijen verwisselen. Als de batterijen nog vol zijn, moet je de weerstanden R12 en R13 controleren:

12K (bruin, rood, zwart, rood, bruin)
10K (bruin, zwart, zwart, rood, bruin)

6.4. Een LED indicator werkt niet

Is de processor correct geplaatst? (controleer de juiste polariteit !)

6.4.1. De status-LED D12 werkt niet

Controleer de polariteit van LED D12.

Weerstand R10, R31 controleren.
470 (geel, violet, bruin, goud)

Een eenvoudige test: je verwijdert voorzichtig de processor (IC1) en maakt een verbinding tussen Pin7* (VCC) en Pin14 (status-LED brandt groen), respectievelijk Pin4 (status-LED brandt rood). Als de status-LED in deze test werkt, is er een fout in de processor, in het kristal of in een van de verbindingselementen (printsporen, soldeerpunten, enzovoorts).

*** Pin1 ligt links bovenaan, daarna wordt er links naar beneden en rechts naar boven geteld.**

6.4.2. De front-LED D11 werkt niet

De polariteit van D11 controleren.

Weerstand R9 controleren.
220 (rood, rood, bruin, goud)

Een eenvoudige test: je verwijdert voorzichtig de processor (IC1) en maakt een verbinding tussen Pin7* (VCC) en Pin12 (front-LED brandt rood). Als de front-LED in deze test werkt, is er een fout in de processor, in het kristal of in een van de verbindingselementen (printsporen, soldeerpunten, enzovoorts).

*** Pin1 ligt links bovenaan, daarna wordt er links naar beneden en rechts naar boven verder geteld.**

6. Foutzoeken

6.4.3. De Back-LED D15 aan de linkerkant werkt niet

Polariteit D15 controleren.

Weerstand R19, R18 controleren.

1K (bruin,zwart,oranje,goud)

4,7K (geel,violet,rood,goud)

Een eenvoudige test: je verwijdert de processor (IC1) en maakt een verbinding tussen Pin7 (VCC) en Pin24 (linkse Back-LED D15 brandt rood). Als de linkse Back-LED D15 in deze test werkt, is er een fout in de processor, in het kristal of in een van de verbindings-elementen (printsporen, soldeerpunten, enzovoorts).

6.4.4. De Back-LED D16 aan de rechterkant werkt niet

Polariteit D16 controleren.

Weerstand R21, R20 controleren.

1K (bruin,zwart,oranje,goud)

4,7K (geel,violet,rood,goud)

Een eenvoudige test: je verwijdert voorzichtig de processor (IC1) en maakt een verbinding tussen Pin7 (VCC) en Pin23 (rechtse Back-LED D16 brandt rood). Als de rechtse Back-LED D16 in deze test werkt, is er een fout in de processor, in het kristal of in een van de verbindings-elementen (printsporen, soldeerpunten, enzovoorts).

6.5. De sensor voor de lijnvolger (T9, T10) reageert niet

Polariteit T9, T10 controleren.

Controleer de weerstanden R14 20K (rood,zwart,oranje,goud), R15 20K (rood,zwart,oranje,goud).

Pas ook op, dat de aansluitdraden voor R15 niet met de daarnaast liggende aansluitingen van R23 of R28 worden verwisseld!

Met een multimeter kun je bij de uitgebouwde processor aan Pin 25 respectievelijk Pin 26 het sensorsignaal nameten: donker = 0V, licht = VCC).

6.6. Een schakelaar, resp. drukknop werkt niet goed

6.6.1. Het lijkt erop, dat een combinatie van meerdere schakelaars gedrukt wordt

Controleer R12 12K (bruin,rood,zwart,rood,bruin) en R13 10K (bruin,zwart,zwart,rood,bruin).

Verder moet je controleren: R25, R26, R27, R28, R29, R30, R32!

R24	1K	(bruin,zwart,zwart,bruin,bruin)	R28	16K	(bruin,blauw,zwart,rood,bruin)
R25	2K	(rood,zwart,zwart,bruin,bruin)	R29	33K	(oranje,oranje,zwart,rood,bruin)
R26	2K	(rood,zwart,zwart,bruin,bruin)	R30	68K	(blauw,grijs,zwart,rood,bruin)
R27	8,2K	(grijs,rood,zwart,bruin,bruin)	R32	2K	(rood,zwart,zwart,bruin,bruin)

Bij wijze van uitzondering kan het gebeuren, dat de toleranties van de onderdelen het systeem een verkeerd patroon laten aflezen: Deze fout wordt later in de software gecorrigeerd.

6.6.2. Het lijkt erop, dat enkele schakelaars verwisseld zijn

De weerstanden voor de schakelaars zijn misschien verwisseld. Je controleert de waarden voor:

R24	1K	(bruin,zwart,zwart,bruin,bruin)
R25	2K	(rood,zwart,zwart,bruin,bruin)
R26	2K	(rood,zwart,zwart,bruin,bruin)
R27	8,2K	(grijs,rood,zwart,bruin,bruin)
R28	16K	(bruin,blauw,zwart,rood,bruin)
R29	33K	(oranje,oranje,zwart,rood,bruin)
R30	68K	(blauw,grijs,zwart,rood,bruin)
R32	2K	(rood,zwart,zwart,bruin,bruin)

6.6.3. Op de een of andere manier werkt het nog steeds niet goed

Controleer de weerstanden, respectievelijk de condensator

R23 1M (bruin,zwart, groen,goud)

R24 1K (bruin,zwart,rood,goud)

R12 12K (bruin,rood,zwart,rood, bruin)

R13 10K (bruin,zwart, zwart, rood, goud)

de condensator: C7 220F/10V

6.7. De lichtsensoren werken niet goed

6.7.1. Geen enkele lichtsensor werkt

Weerstand R22 controleren!

470 (geel,violet,bruin,goud)

Controleer s.v.p. D13 en D14.

D13 en D14 zijn de rozegekleurde tweepolige behuizingen met een kleine markering aan een kant. Deze markering moet naar de buitenkant van de print wijzen

6.7.2. De lichtsensor aan de linkerzijde werkt niet

Weerstand R18 controleren!

4,7 K (geel,violet,rood,goud)

Controleer de positie en de aansluitingen van T11.

T11 is de transparante tweepolige behuizing met een kleine markering aan een kant. Deze markering moet naar de buitenkant van de print wijzen.

6.7.3. De lichtsensor aan de rechterzijde werkt niet

Weerstand R20 controleren!

4,7 K (geel,violet,rood,goud)

Controleer de positie en de aansluitingen van T12.

T12 is de transparante of zwarte tweepolige behuizing met een kleine markering aan een kant. Deze markering moet naar de buitenkant van de print wijzen.

Met een multimeter kan men aan Pin24 respectievelijk aan Pin23 het sensorsignaal bij de uitgebouwde processor nameten: donker = VCC, licht = 0V.

6.8. De motoren werken niet goed

6.8.1. Geen enkele motor werkt

Controleer de polariteit, de aansluitingen en positie van IC3.

6.8.2. De motor aan de linkerzijde werkt niet of draait slechts in één richting

In dit geval moeten wij de complete motorbrug, bestaande uit de transistors T1, T2, T3, T4 (heb je wel de goede transistors op de juiste plaats ingebouwd?), de diodes D1, D2, D3, D4 (polariteit!) en de weerstanden R1, R2, R3, R4 controleren.

T1, T3 (BC327-40 of BC328-40), T2, T4 (BC337-40 of BC338-40)

R1, R2, R3, R4 1K (bruin, zwart, rood, goud)

6.8.3. De motor aan de rechterzijde werkt niet of draait slechts in één richting

In dit geval moeten wij de complete motorbrug, bestaande uit de transistors T5, T6, T7, T8 (heb je wel de goede transistors op de juiste plaats ingebouwd?), de diodes D5, D6, D7, D8 (polariteit!) en de weerstanden R5, R6, R7, R8 controleren.

T5, T7 (BC327-40 of BC328-40), T6, T8 (BC337-40 of BC338-40)

R5, R6, R7, R8 1K (bruin, zwart, rood, goud)

6.8.4. Een motor draait in de verkeerde richting

De twee aansluitpunten van de kabels voor de motor, die in de verkeerde richting draait, moeten onderling verwisseld worden.

6.9. De IR-interface

6.9.1. ASURO verzendt geen lettertekens

Controleer de polariteit van de IR-Diode D10.

Is er iets mis met weerstand R16?

220 (rood, rood, bruin, goud)

6.9.2. ASURO ontvangt geen enkel symbool

Tussen de IR-Transceiver en ASURO moet zichtcontact over een afstand van maximaal 50cm mogelijk zijn. Ook moet de IR-Transceiver goed werken (zie hoofdstuk 6.1).

Is IC2 correct ingebouwd?

Controleer vervolgens R17 en C2 !

470 (bruin, zwart, bruin, goud)

100nF (opschrift 104)

Als je nog steeds geen fout gevonden hebt, moet je nu overleggen of IC2 misschien "gelast" in plaats van "gesoldeerd" is. IC2 is nogal hittegevoelig en kan bij de soldeerprocedure kapot zijn gegaan. In dat geval moet je een nieuw IC (SFH 5110-36) aanschaffen en inbouwen.

6.9.3. Het werkt nog steeds niet zoals het moet...

Is de polariteit van C8 correct?
220F/ minimaal 10V

Als de informatie uitwisseling tussen PC en ASURO steeds weer moeilijkheden oplevert, moet je de trimmer TR1 van de transceiver een beetje bijstellen. Erg belangrijk is ook de RS-232 spanning van de computer, die moet hoog genoeg zijn! Je kunt een buffer elko over deze spanning zetten of er zelfs aan kunnen denken om een externe spanningsbron te gebruiken voor de IR-transceiver, in plaats van de RS-232 computerspanning.



Tweecomponentenlijm is gemeen en gevaarlijk. De twee chemische stoffen en het eindproduct kunnen allergische reacties ontwikkelen. Raak het materiaal dus niet aan. Vinylhandschoenen zijn aan te bevelen. Als je een van deze stoffen toch op de huid krijgt, dan deze onmiddellijk en grondig met zeep afwassen!

Secondenlijm is oorspronkelijk voor de chirurgie ontwikkeld. Dat wordt vooral duidelijk als je opeens merkt, dat de duim en vinger binnen seconden aan elkaar vastplakken. Zolang het alleen maar je vingers zijn, kun je met warm water, zeep en wat geduld de verbinding weer losmaken. Dit mag echter in geen geval met je lippen of ogen gebeuren! Het is dus belangrijk om tijdens het werken met secondenlijm elke krabneiging aan het hoofd, aan het gezicht en in de buurt van de ogen te onderdrukken.

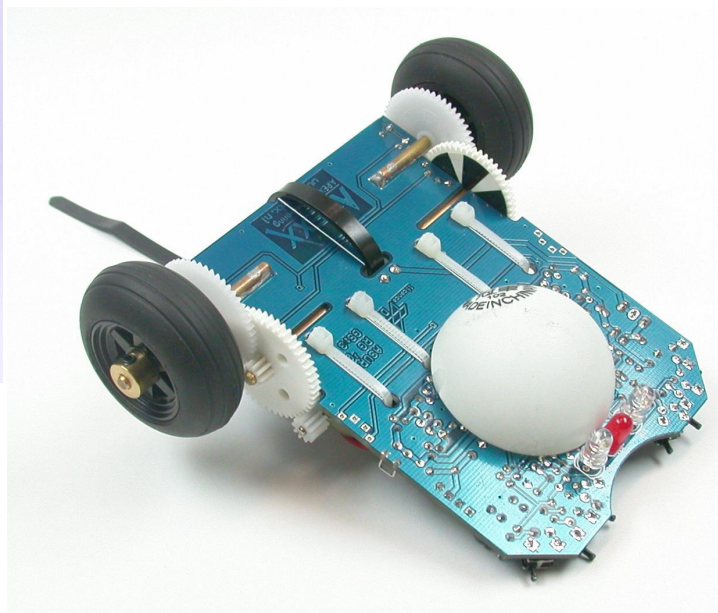
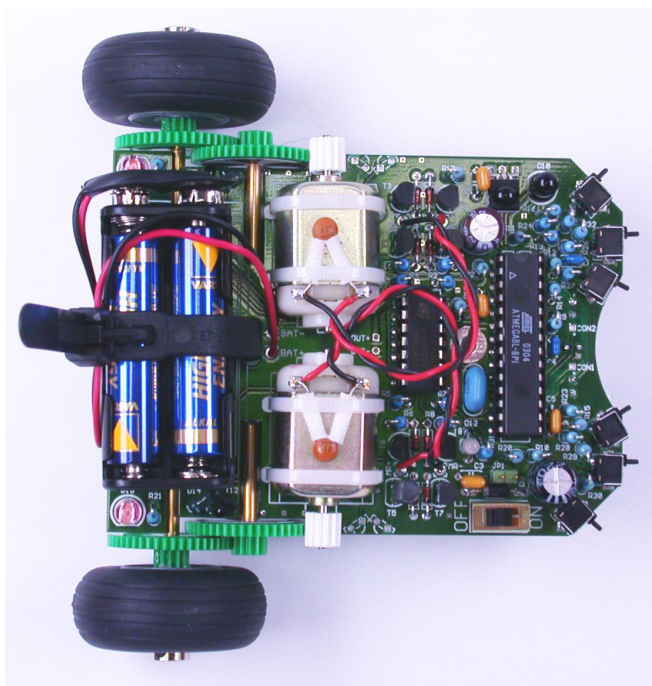
7. Afsluitende karweitjes

Tot slot worden de assen licht ingevet. Steek nu de tandwielen met de fraaie zwartwitte patronen op de korte assen. Plaats dan het wiel op de tandwielen met de 50 en 12 tandjes en schuif deze samen op de achterste as. Fixeer de positie losjes met een stelring, zodat de wielen gemakkelijk kunnen draaien.

De provisorisch bevestigde motor wordt voorzichtig zodanig heen en weer verschoven, dat het rondsel (motoraandrijfwieltje) over de hele breedte optimaal in het tandwiel grijpt. Het rondsel en het tandwiel moeten gemakkelijk draaien (Ter controle van de goede werking kun je eventueel nog eens de complete zelftest laten lopen).

Als de positie van de motor goed gekozen is, houd je de motor en printplaat goed vast en zet met een druppel secondenlijm in de spleet tussen de motor en de printplaat de motor muurvast op de print. Daarbij moet je er echter rekening mee houden, dat de secondenlijm soms ook enkele minuten nodig heeft om te drogen.

Plak nu nog de halve pingpongbal met twee tegenover elkaar geplaatste druppels secondenlijm op de onderkant van de printplaat. De correcte positie van de halve pingpongbal ligt pal achter de onderdelen, die de lijnvolger (zie figuur 7.1) besturen. Ook deze lijm moet je nog even laten drogen.



Figuur 7.1.: Een kant en klare ASURO

Deel III. Informatica

8. Installatie van de Software en eerste stappen op weg naar de programma's

Je kunt de ASURO-CD in je CDROM-drive plaatsen, dan start de CD automatisch. Als de Autostart-optie afgeschakeld is, kun je de CD ook in de Explorer openen. Na de keuze van de taal vind je in de map "Software" alles, wat er voor het werken met de ASURO nodig is. Deze programma's moeten eerst op je PC worden geïnstalleerd. Voor deze installatie moet je over administratierechten beschikken. Als je als PC-gebruiker daarover niet beschikt, meld je je even af en meld je je als administrator weer aan.

De installatie van de software verloopt in de volgende vier fasen:

1. Allereerst installeer je het Flashprogramma om eigen programma's naar de ASURO over te brengen en daarin op te slaan.
2. Vervolgens installeer je een programma editor (Programmers Notepad 2, PN2) en een Compiler (WinAVR).
3. Dan wordt er een programmavoorbeeld van de CDROM op de harde schijf gekopieerd.
4. In de programma editor (PN2) wordt ook nog een menu MAKE en een menu CLEAN gegenereerd.

8.1 Windows



8.1.1 Flash-programma



Het Flashprogramma kun je naar keuze in een map op de harde schijf (bijvoorbeeld C:\Programma's\Flash) kopiëren of laten starten direct van de CD. In beide gevallen is het aanleggen van een link naar de desktop handig en aan te bevelen, omdat je daarmee het Flashprogramma gemakkelijk kunt starten.

Klik op [Save]

8.1.2 Installatie van de programma-editor en de compiler

Voor de installatie van de compiler moet je over administratierechten beschikken, omdat je daarbij een regel in het PC-register aanpast. Als je als PC-gebruiker niet over deze rechten beschikt, meld je je even af en meld je je als administrator weer aan!

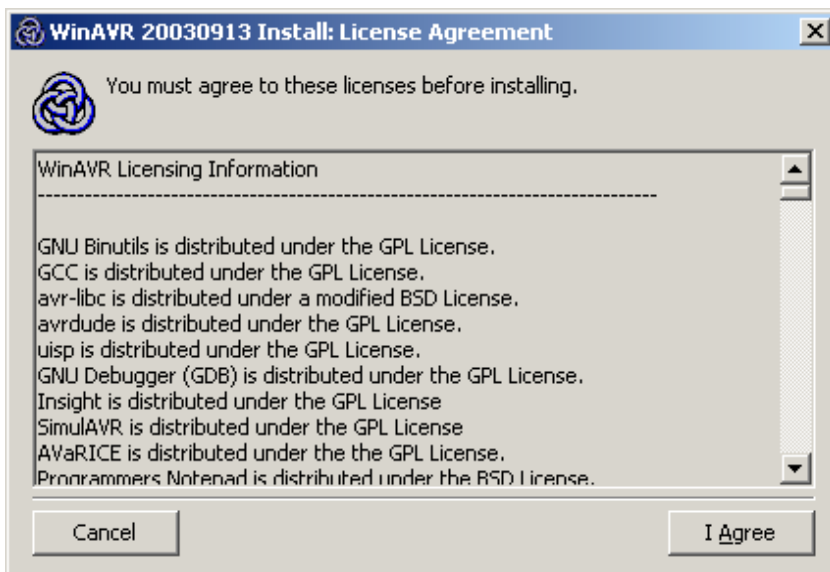
klik op [Install]



COMPILER WinAVR (200xxxxx)

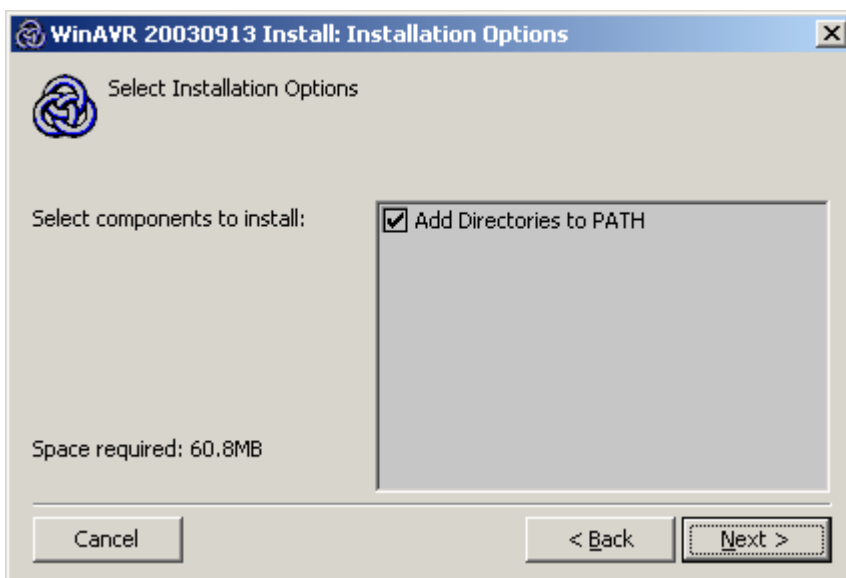
8. Informatica

Nu verschijnt het volgende venster:



Klik op [I Agree]

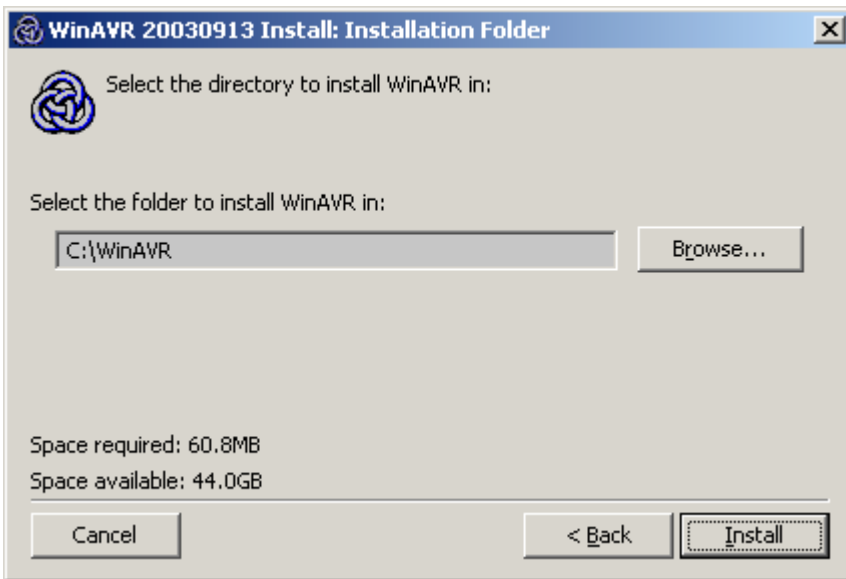
Daarna verschijnt het volgende venster:



Klik op [Next]

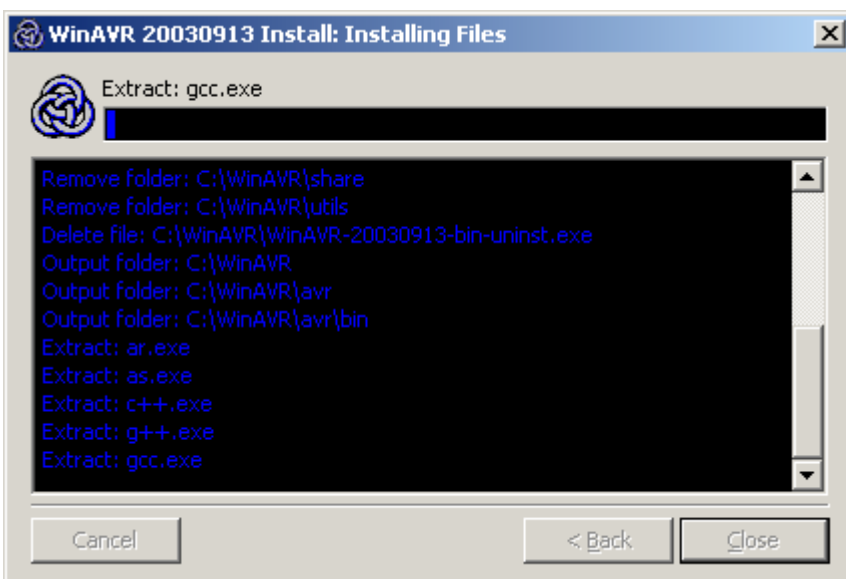
8. Informatica

Het volgende venster verschijnt:



Klik op [Install]

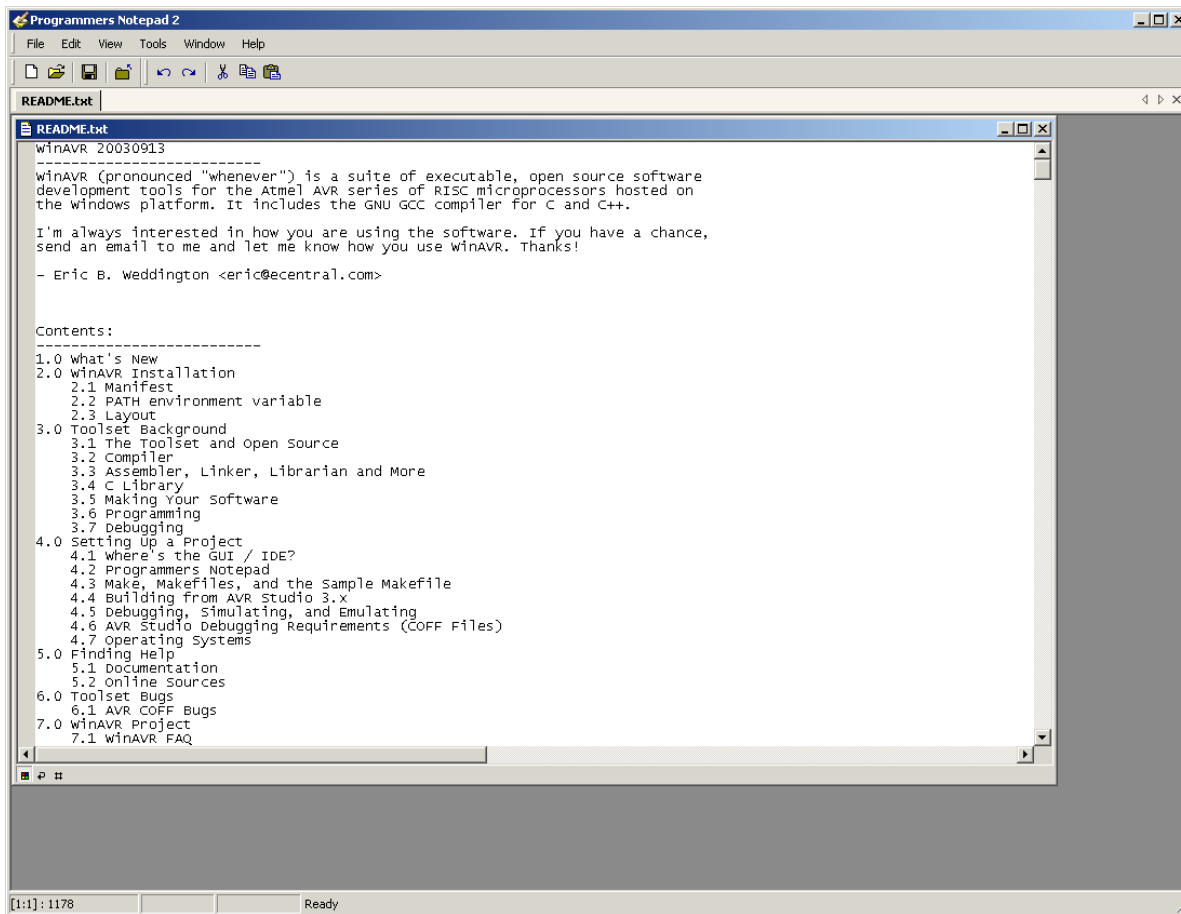
Daarna verschijnt het volgende venster, waarin programma's worden uitgepakt:



Wachten...

8. Informatica

....totdat de Programmers Notepad 2 (PN2) Editor met het bestand README.txt verschijnt.



The screenshot shows a window titled "Programmers Notepad 2" with a menu bar (File, Edit, View, Tools, Window, Help) and a toolbar. The main text area displays the content of a file named "README.txt". The text includes a title "winAVR 20030913", a description of winAVR as open-source software for AVR microprocessors, contact information for Eric B. Weddington, and a detailed table of contents with sections numbered 1.0 through 7.1.

```
winAVR 20030913
-----
winAVR (pronounced "whenever") is a suite of executable, open source software
development tools for the Atmel AVR series of RISC microprocessors hosted on
the windows platform. It includes the GNU GCC compiler for C and C++.

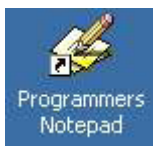
I'm always interested in how you are using the software. If you have a chance,
send an email to me and let me know how you use winAVR. Thanks!

- Eric B. Weddington <eric@central.com>

Contents:
-----
1.0 What's New
2.0 winAVR Installation
  2.1 Manifest
  2.2 PATH environment variable
  2.3 Layout
3.0 Toolset Background
  3.1 The Toolset and Open Source
  3.2 Compiler
  3.3 Assembler, Linker, Librarian and More
  3.4 C Library
  3.5 Making Your Software
  3.6 Programming
  3.7 Debugging
4.0 Setting Up a Project
  4.1 where's the GUI / IDE?
  4.2 Programmers Notepad
  4.3 Make, Makefiles, and the Sample Makefile
  4.4 Building from AVR Studio 3.X
  4.5 Debugging, Simulating, and Emulating
  4.6 AVR Studio Debugging Requirements (COFF Files)
  4.7 Operating Systems
5.0 Finding Help
  5.1 Documentation
  5.2 online sources
6.0 Toolset Bugs
  6.1 AVR COFF Bugs
7.0 winAVR Project
  7.1 winAVR FAQ
```

Nu moet je het venster `Programmers Notepad 2' sluiten.

Op de Desktop van je PC verschijnt nu een symbool met de titel `Programmers Notepad 2':



De Programma-editor en de Compiler zijn geïnstalleerd en staan nu op je PC ter beschikking.

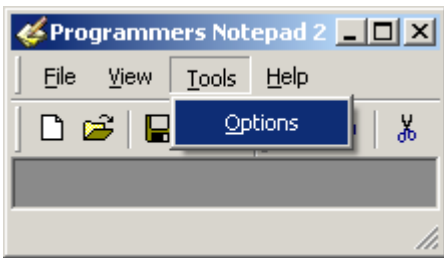
8.1.3. Voorbeeldprogramma's van de CDROM naar je harde schijf kopiëren.

Van de CDROM kun je de map `ASURO_src` naar een willekeurige map (bijvoorbeeld `C:\ASURO_src`) op je harde schijf kopiëren.

Nu markeer je de gekopieerde bestanden in deze nieuwe map, activeert vervolgens de rechtse muistoets en schakelt dan in „Eigenschappen“ alle bestanden op „veranderbaar“.

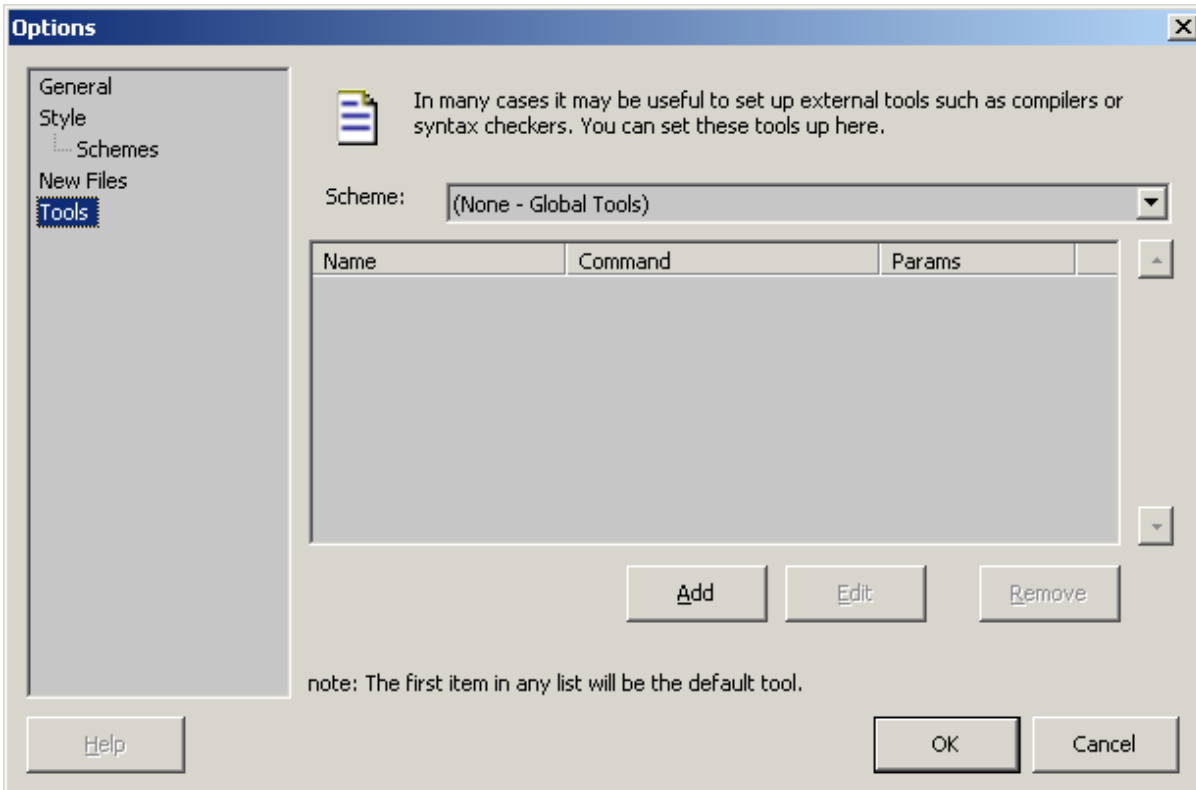
In de programma editor genereer je nu een menuveld om de compiler te starten

Open de `Programmers Notepad 2` met een dubbele klik op het desktopsymbool `Programmers Notepad`:



In het menu tools moet je nu „Opties“ kiezen.

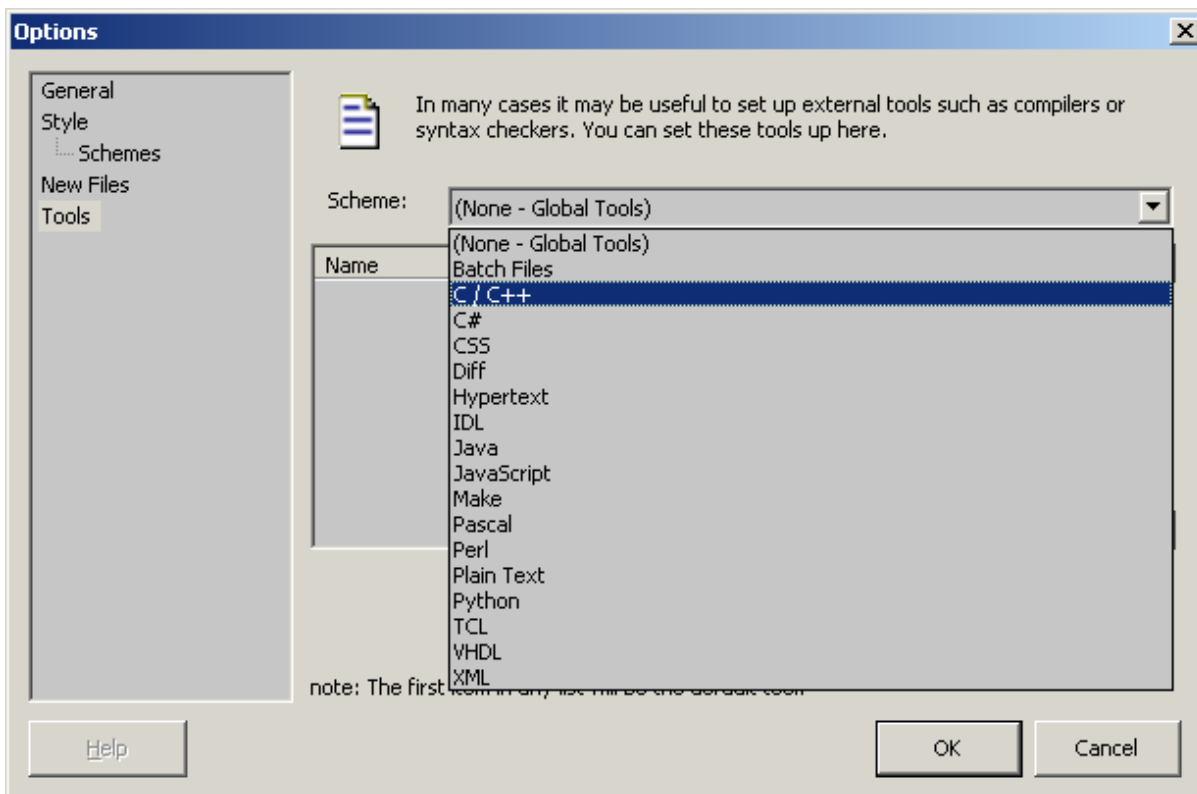
Daarop verschijnt het Opties venster.



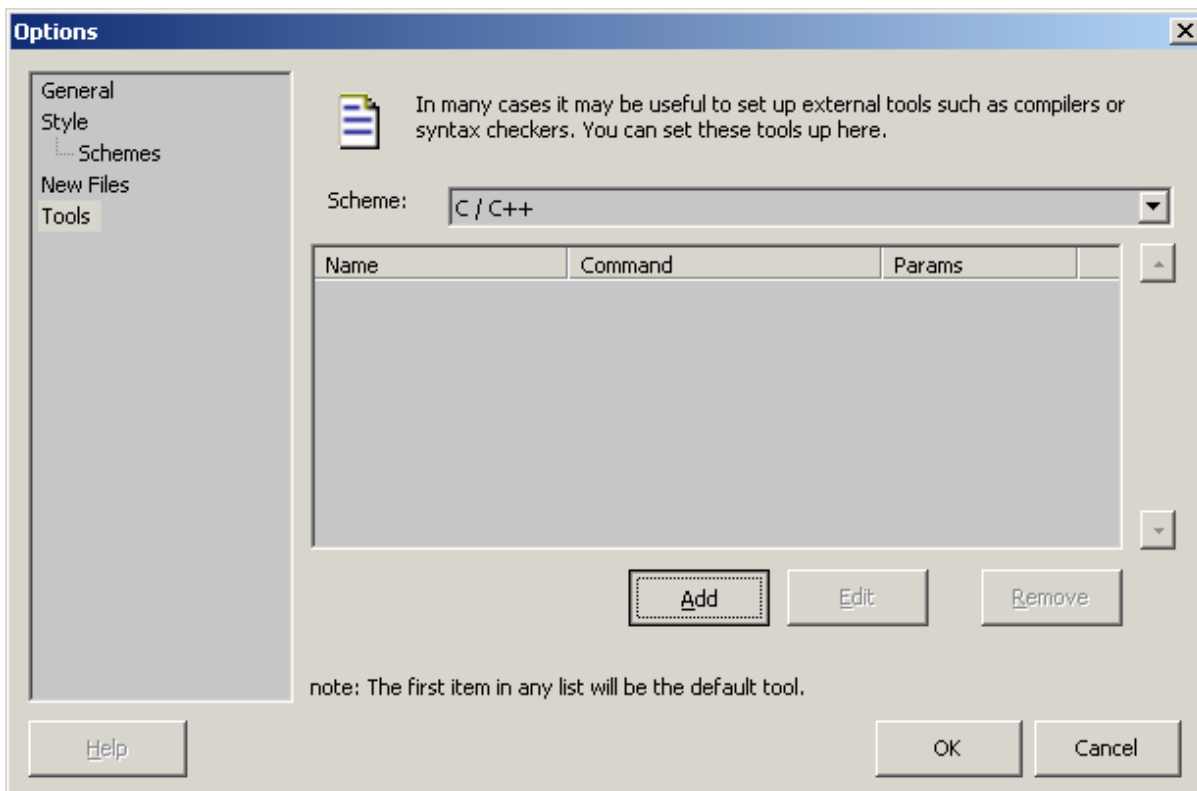
Nu nog tools kiezen.

8. Informatica

In het venster aan de rechterkant het schema `C/C++` kiezen.



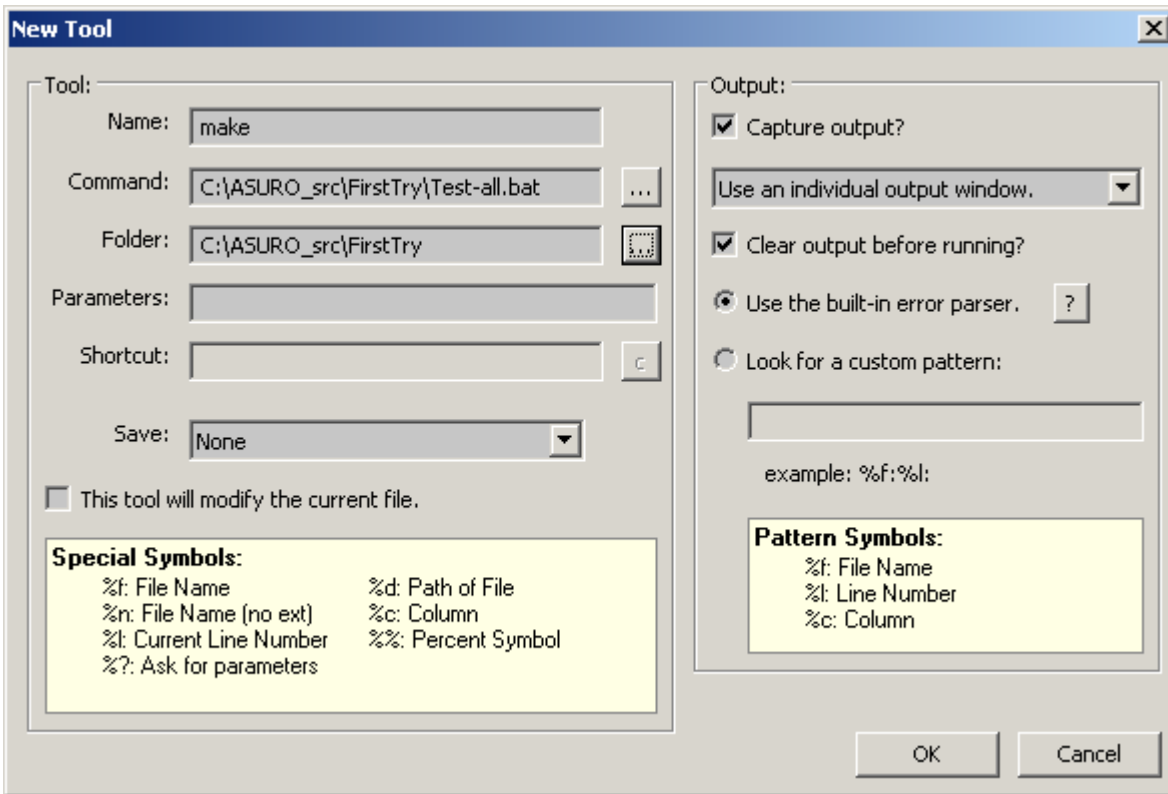
Je hebt nu het schema `C/C++` geselecteerd.



Op [Add] klikken (...om een nieuwe tool toe te voegen)

8. Informatica

Het venster 'New Tool' verschijnt nu op het beeldscherm.



Typ nu de volgende gegevens in of kies de juiste parameters met de Browse-toets:

Name: make
Command: C:\ASURO_src\FirstTry\Test-all.bat
Folder: C:\ASURO_src\FirstTry

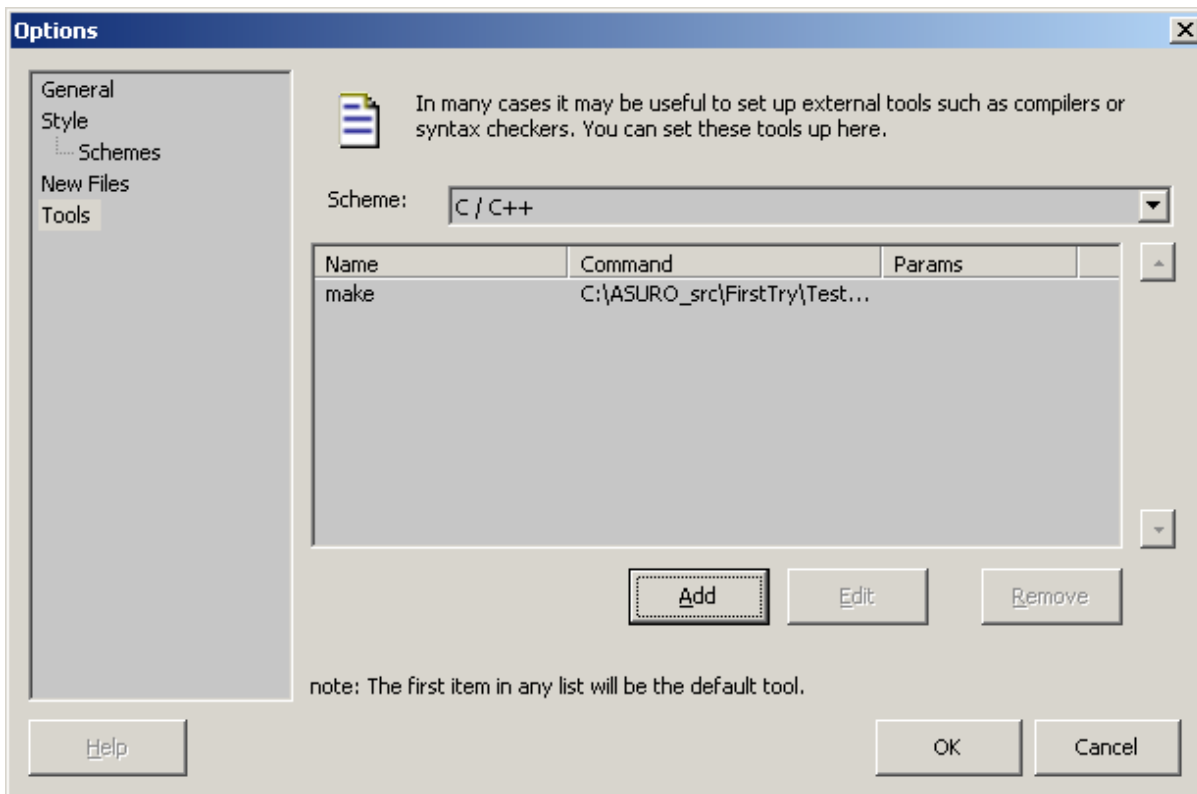
Deze keuzes moet je nog even bevestigen met [OK]

Met behulp van deze commando's stelt het systeem je direct een nieuwe PN-Tool met de naam make in het Tools-hoofdmenu ter beschikking. Een klik op de bijpassende toets activeert een Batchbestand met de naam Test-clean.bat, dat het programma test.c – tezamen met asuro.c - compileert en een bestand text.hex genereert.

8. Informatica

Toevoeging van een commandotoets met de opdracht „Opruimen“ in de Programmeereditor.

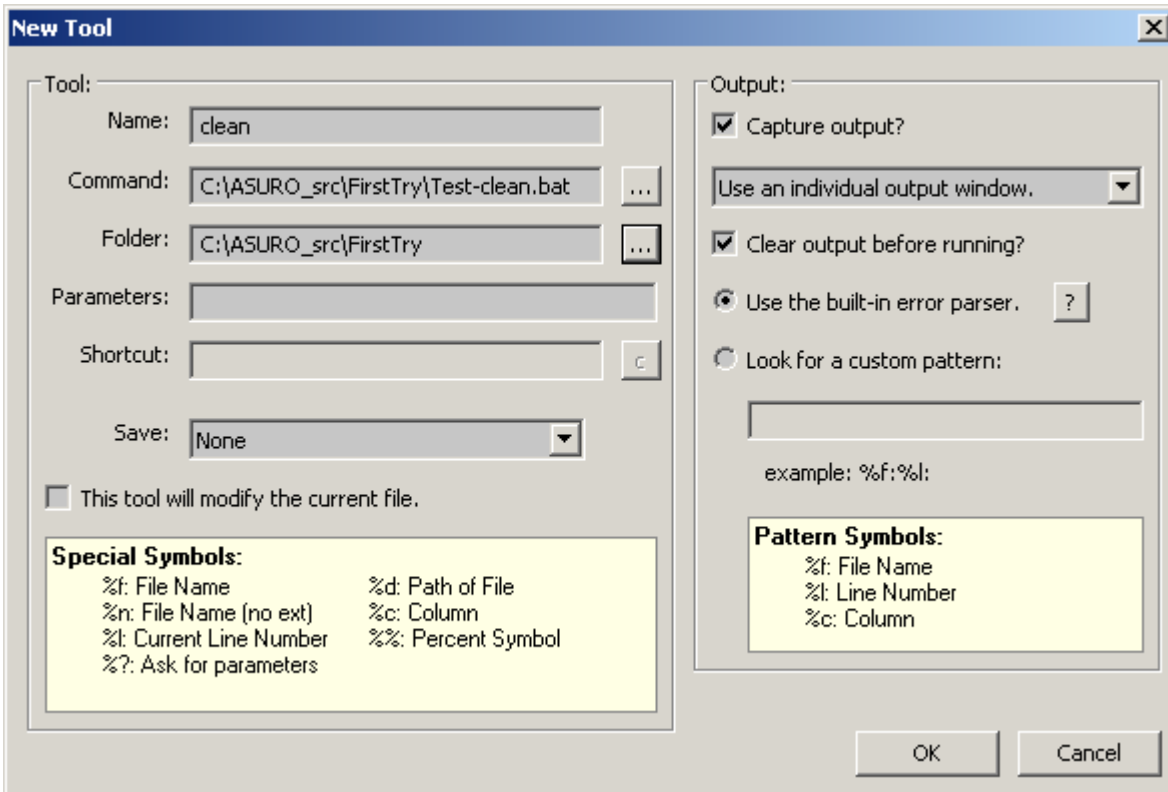
Kies in het hoofdmenu “Tools” nogmaals “Options” en daarna weer “C/C++-Scheme”:



Vervolgens [Add] aanklikken om weer een nieuw werktuig te genereren:

8. Informatica

Het systeem biedt een venster 'New Tool' aan.



Typ daar de volgende parameters in, respectievelijk selecteer de gewenste instellingen met de Browse-toets: 

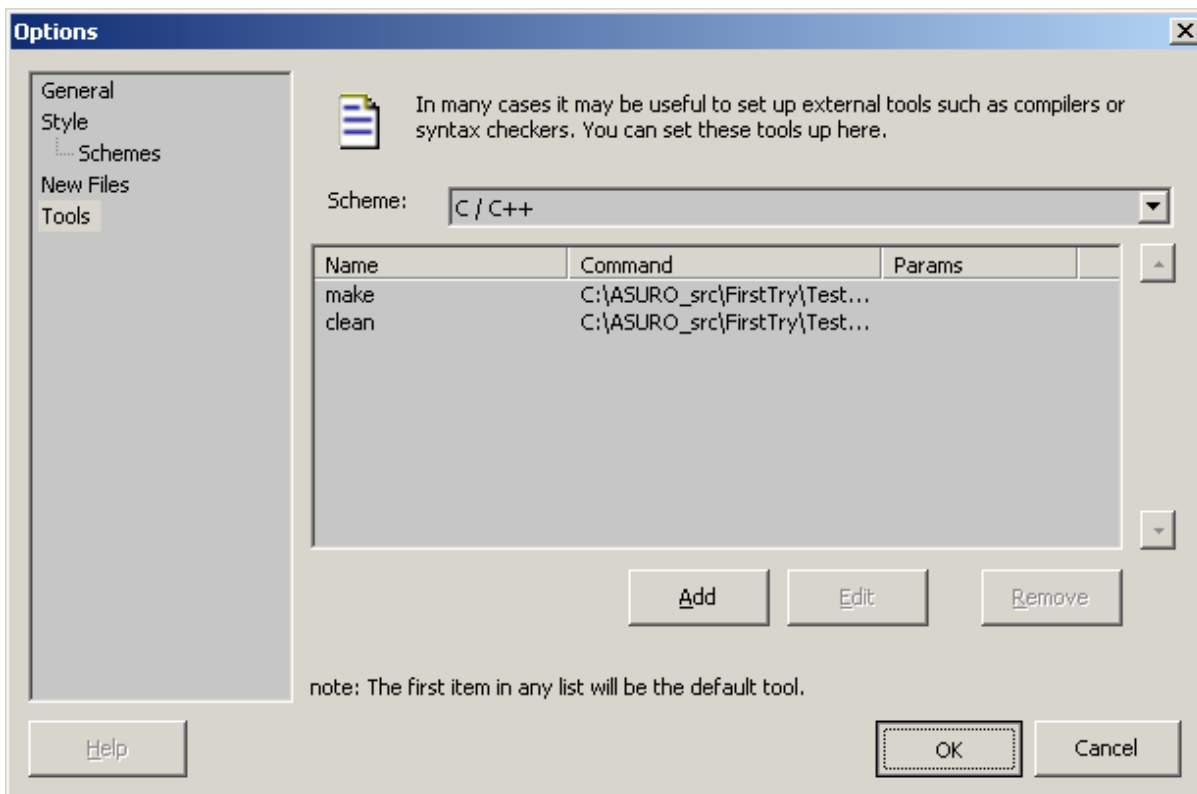
Name: clean
Command: C:\ASURO_src\FirstTry\Test-clean.bat
Folder: C:\ASURO_src\FirstTry

Bevestig deze keuzes met [OK]

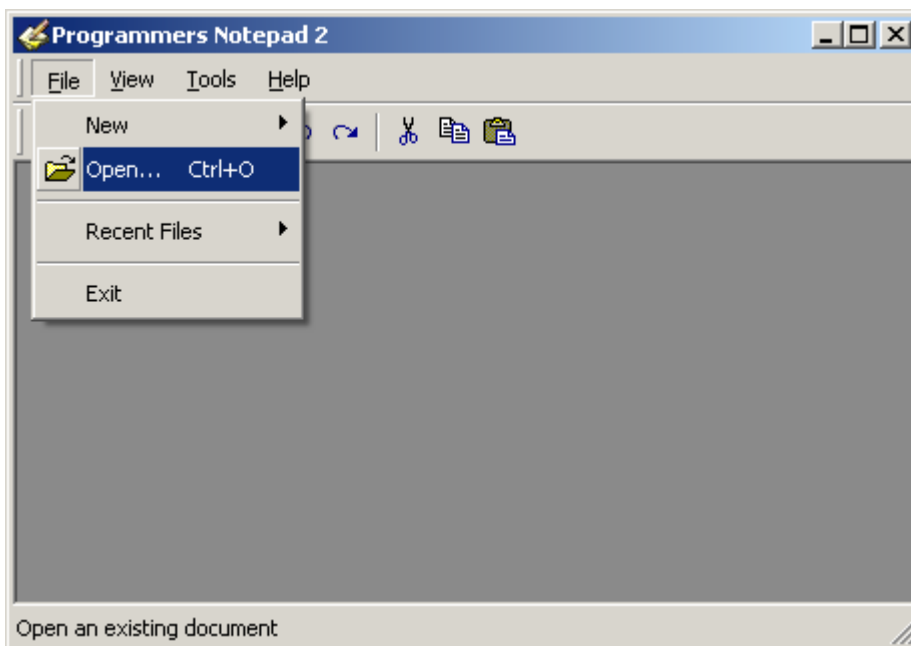
Met behulp van deze commando's stelt het systeem je direct een nieuwe PN-Tool met de naam clean in het Tools-hoofdmenu ter beschikking. Een klik op de bijpassende toets activeert een Batch-bestand met de naam Test-clean.bat, dat de tijdelijke bestanden in de map C:\ASURO_src\FirstTry opruimt (uitwist).

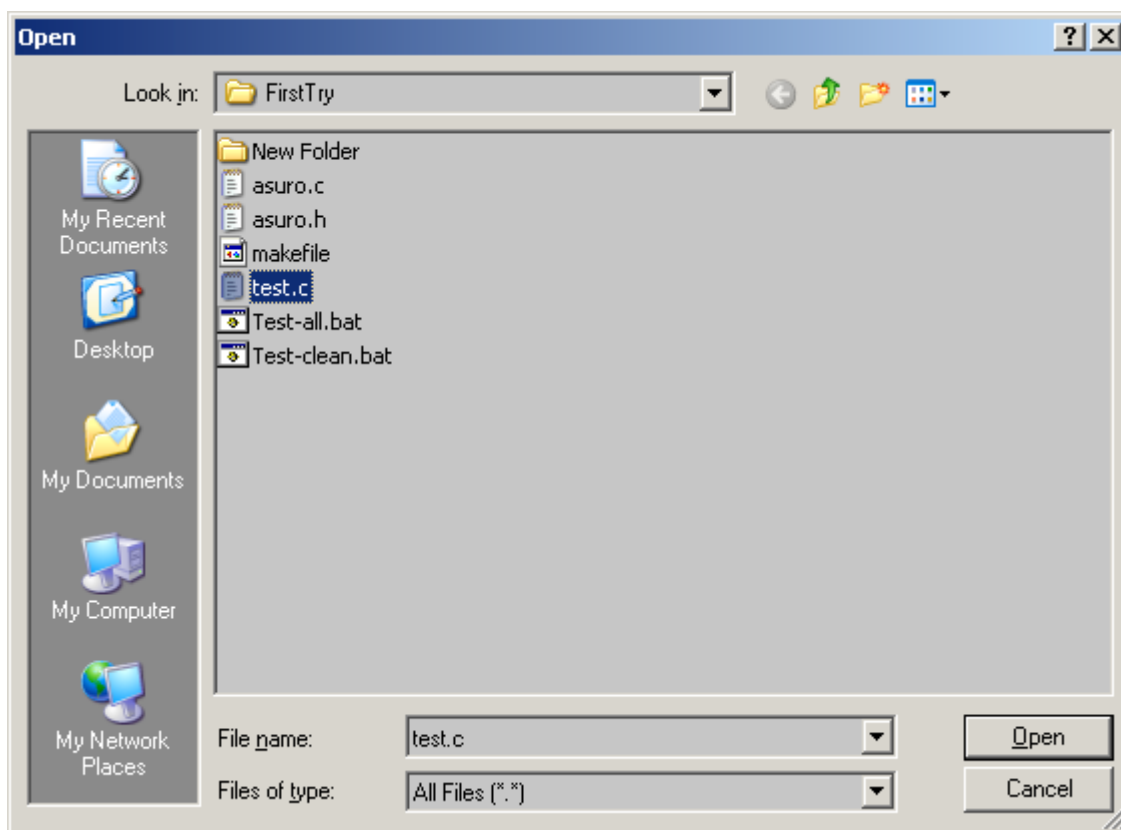
8. Informatica

In het venster met de titel Options moet je nu beide titels `make` en `clean` terugvinden.



Bevestig dit nogmaals met [OK].



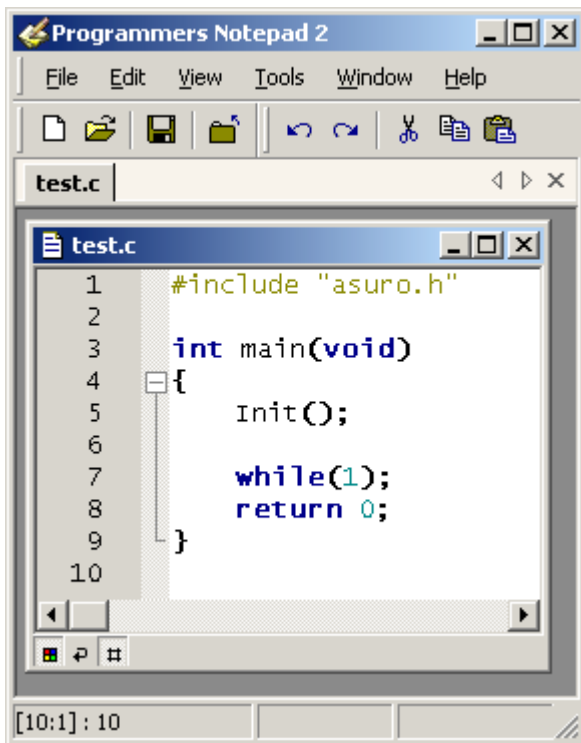


Klik nu op [Open].

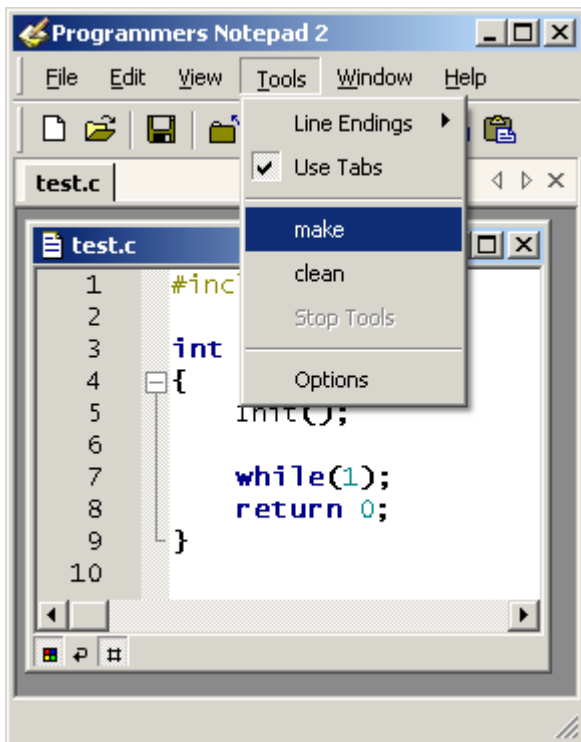
Open het bestand 'C:\ASURO_src\FirstTry\test.c' voor een eerste systeemtest:

8. Informatica

Het bestand test.c wordt geopend.



Als je nu Tools kiest...

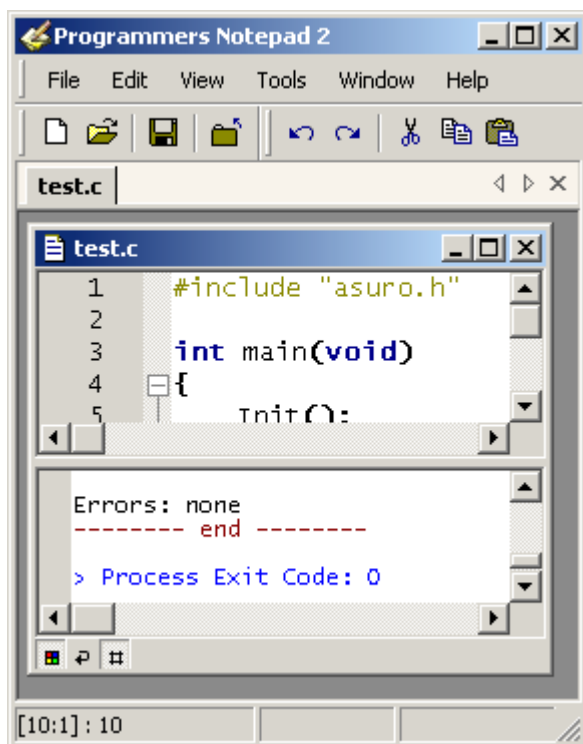


...zie je de nieuwe Tools make en clean, die je zojuist aan het menu hebt toegevoegd.

Klik nu op make.

8. Informatica

Het programma compileert nu de bestanden voor test.c (samen met asuro.c)...



...en als het programma geen fouten bevat (wat wij mogen verwachten, omdat je zojuist een voorbeeldprogramma hebt geladen), meldt de compiler je onderaan: Errors: none.

Wat is er nu gebeurd?

Uit het bestand test.c (en asuro.c) heeft de compiler een bestand test.hex gegenereerd. Dit bestand vormt het programma in machinecode, dat nu naar het geheugen van de ASURO kan worden overgebracht (geflashed). Dit programma doet zelf nog niets, maar wij willen het later nog gebruiken om de Flash-Tool te testen.

Wat gebeurt er op de achtergrond?

Het menucommando "make" voert het Batchbestand "Test-all.bat" uit (een Batchbestand bevat een lijst met commando's op afzonderlijke regels, die regel voor regel worden afgewerkt). In Test-all.bat voeren wij een commando `make all' uit. `make' werkt altijd een makefile af, dat zich (in het ASURO-programmeersysteem) steeds in dezelfde map moet bevinden als Test-all.bat. Een makefile is een tekstbestand met aanwijzingen hoe een of meerdere programma's moeten worden gecompileerd.

Dat blijft allemaal nog overzichtelijk, zolang programma's nog uit een bestand worden verwerkt. Voor zeer complexe in C geschreven programma's wordt de programmeertekst (code) echter over een groot aantal bestanden opgedeeld, die allemaal in een bepaalde volgorde moeten worden gecompileerd (vertaald) en verbonden (gelinked). Deze werkwijze maakt dan ook een "makefile" zeer complex.

Het commando `all' verwerkt een makefile met de naam `all', waarmee een compleet project (in tegenstelling tot het vertalen van losse projectdelen) kan worden gecompileerd.

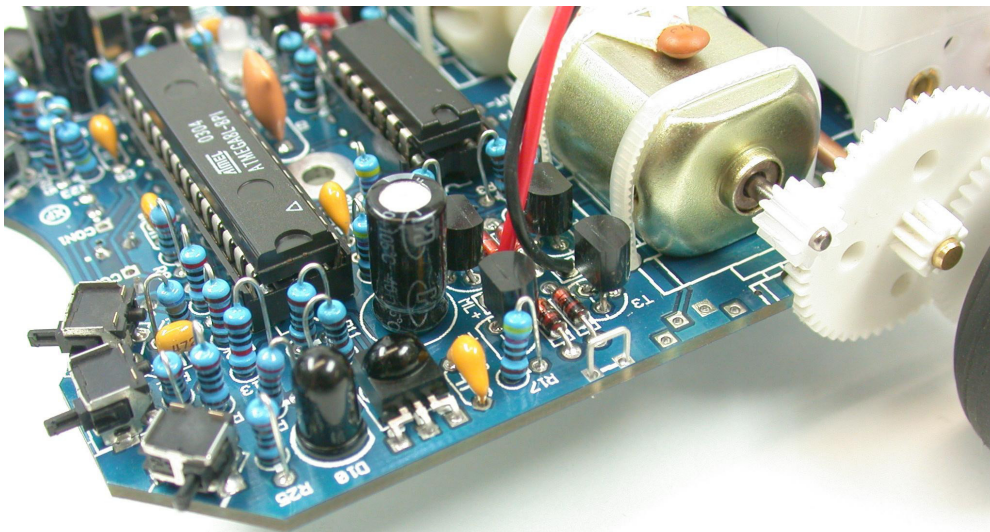
Het makefile voor ons voorbeeldprogramma is zodanig geschreven, dat het een bestand test.c tezamen met asuro.c (dit bestand bevat enkele vooraf vastgelegde definities) compileert en een .hex-bestand genereert, dat direct naar de ASURO kan worden overgebracht.

Let op! Dat betekent, dat wij – zolang wij de makefile onveranderd gebruiken en dus steeds alleen maar kopiëren – ons eigen programma altijd test.c moeten noemen.

Wie makefiles echt goed wil begrijpen (wat echter in onze eerste tests niet nodig is) kan de documentatie in <http://www.gnu.org/directory/make.html> eens bestuderen.

Met de basiskennis voor de de programmering van de ASURO maken we in hoofdstuk 9 kennis.

Bij het compileren van een programma genereert het systeem enkele „hulpbestanden“, die alleen tijdens de vertaling worden gebruikt en daarna overbodig zijn. Voor het opruimen van deze overtollige bestanden hebben wij zojuist de `clean'-Tool aan het menu toegevoegd.

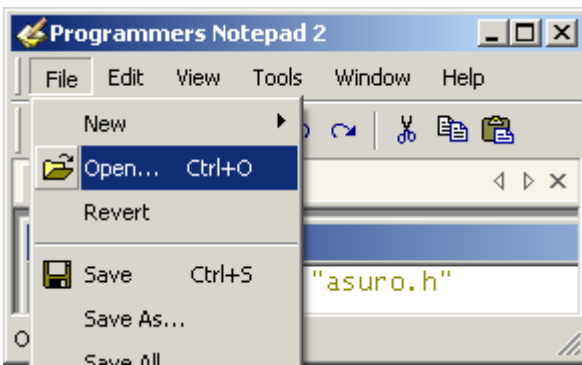


VRAGEN?

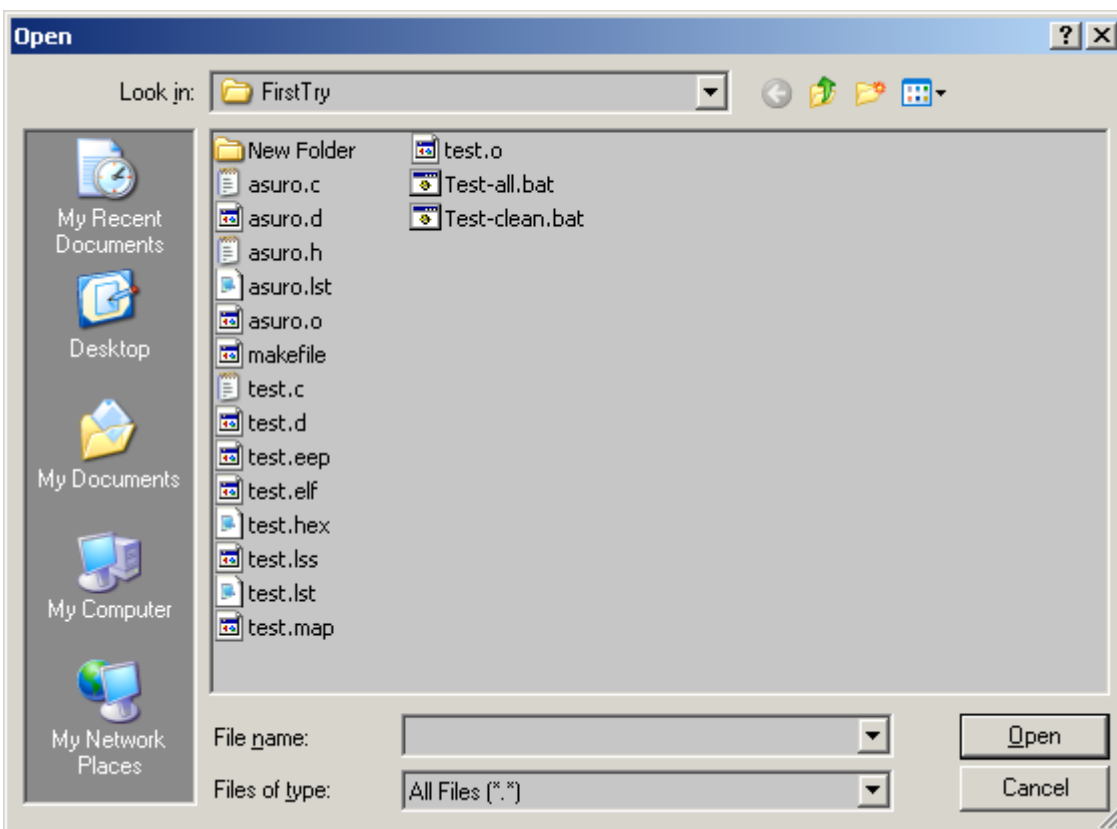
Zie www.arexx.com en www.roboternetz.de

8. Informatica

Bij het openen...

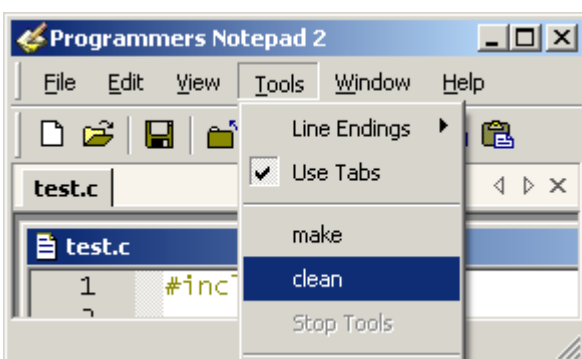


...zie je een lijst van de gegenereerde bestanden...



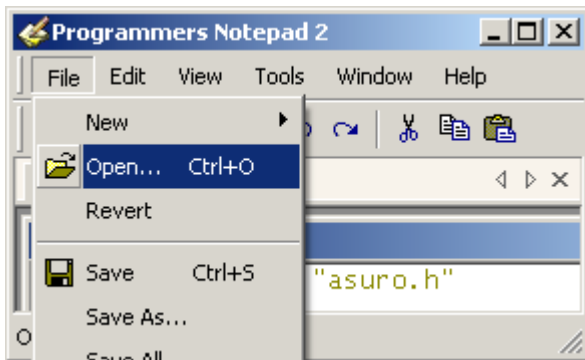
(klik nu op [Cancel])

...en na het opruimen met `clean`...

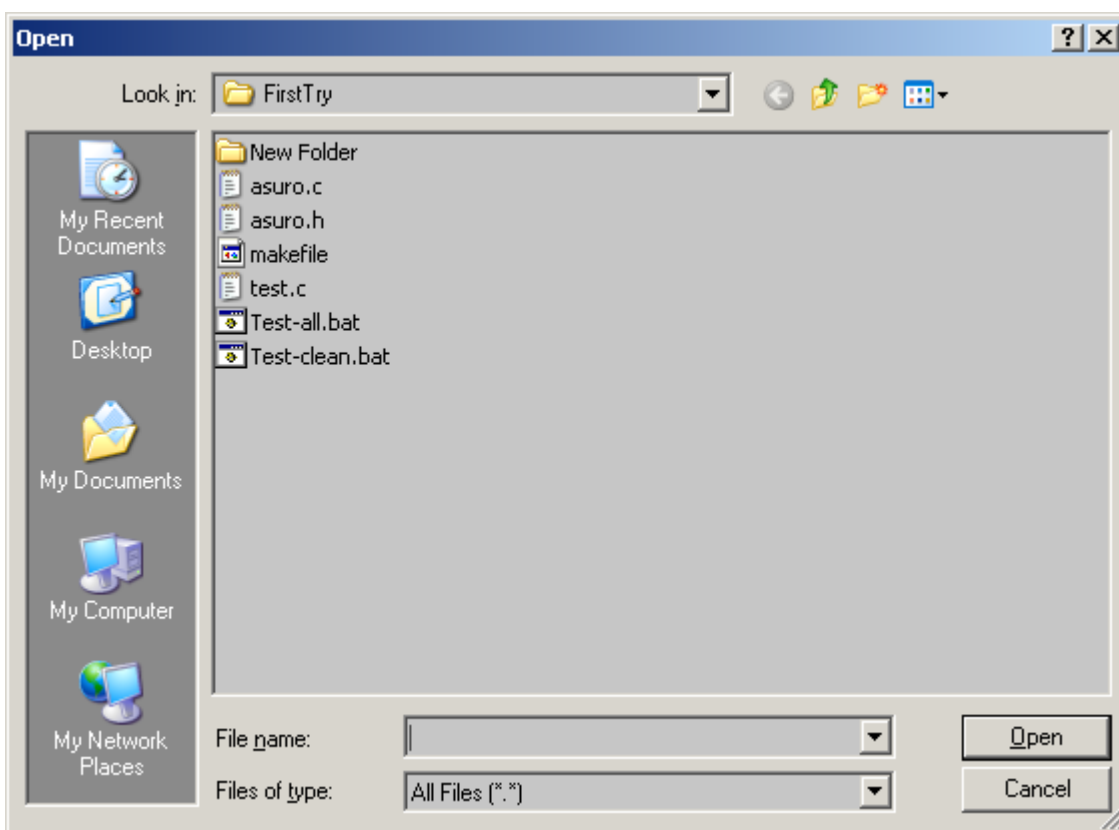


8. Informatica

...kun je zien...



...dat alle gegenereerde bestanden weer zijn verdwenen.



Hoe werkt dat?

Het menucommando `clean` heeft het Batchbestand Test-clean.bat gestart, waarin make met de parameter `clean` wordt opgeroepen. Dit commando voert een regelcommando clean in het makebestand clean uit, dat alle intussen overbodig geworden bestanden weer verwijdert.

8.2. LINUX

Voor de installatie van de software moet je onder Linux over rootrechten beschikken. Je moet je daartoe:

- ofwel afmelden (uitloggen) en als root weer aanmelden (inloggen)
- of -als alternatief- een shell openen en met “su” de rootrechten inschakelen.

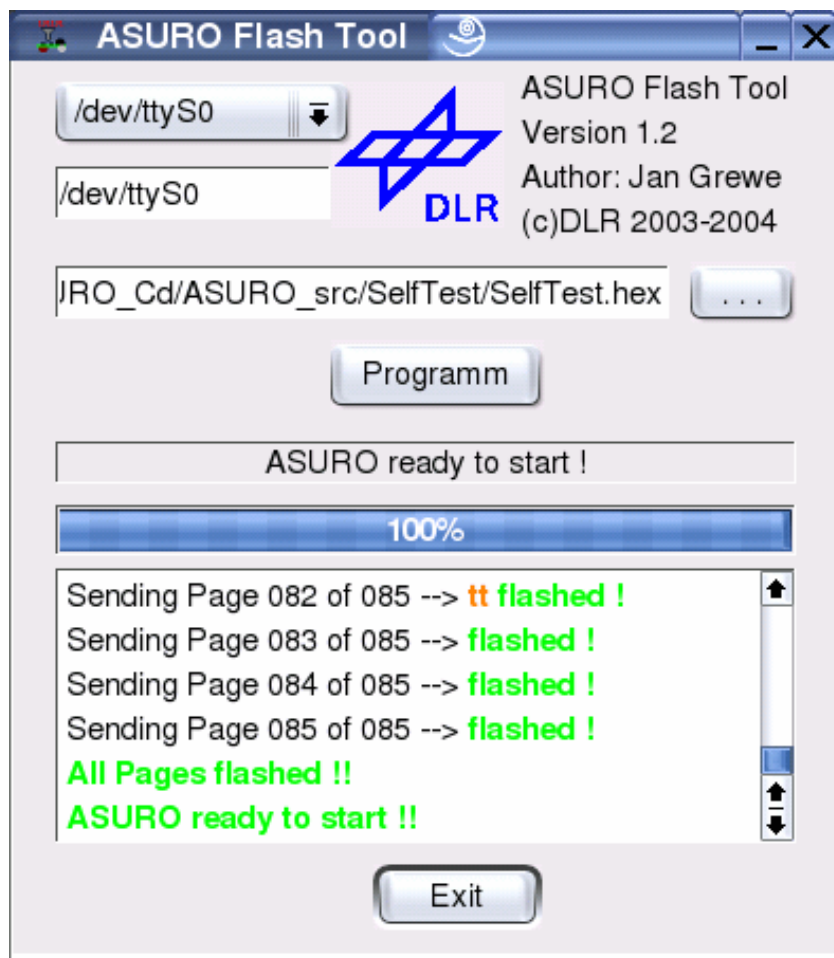
8.2.1 Flash-Tool

Plaats de ASURO-CDROM in je PC en zo nodig moet je de CD-drive nog mounten. Vervolgens moet je de twee Flash-Tools “asuroflash” en “asurocon” uit de map “/Linux/Tools/” naar de map “/usr/local/bin” kopiëren. Vervolgens is het nog nodig je toestemming om deze programma’s te kunnen starten met het commando:

```
“chmod a+x /usr/local/bin asurocon asuroflash”
```

te activeren.

Als je in een shell het bevel “asuroflash” intypt en de PC het bijbehorende programma niet kan vinden, moet je nog even de parameter “/usr/local/bin” aan de %PATH-variabele toevoegen of – als alternatieve methode – het programma met de volledige bestandsnaam starten.



Figuur 8.1.: Flash tool

Het eerste programma is foutloos verzonden.

8.2.2 Compiler

Ter installatie van de Gnu-Compiler voor de AVR-Processoren moet je de ASURO-CDROM in de drive plaatsen en uit de map “/Linux/Compiler/” in elk geval minimaal de volgende pakketten in de aangegeven volgorde installeren:

1. **avr-binutils-... .rpm**
2. **avr-gcc-... .rpm**
3. **avr-libc-... .rpm**

De installatie is heel eenvoudig!

Je schrijft in de console met rootrechten het commando: `rpm -i <pak>.rpm`

Klaar is Kees!

Als editors zijn bijvoorbeeld Exmacs, Kate of Kedit geschikt. Om deze uit te proberen kun je als gewoon gebruiker (d.w.z. geen rootrechten) de voorbeeldbestanden uit de CD-map “/ASURO_src/FirstTry/” in een map in je home-bereik kopiëren, bijvoorbeeld in: “~/ASURO/”.

Daarna open je een shell, ga naar de boven genoemde map in je homebereik en typ daarin het commando “make”. Als je alles goed hebt geïnstalleerd, zie je op het scherm het volgende protocol verschijnen (zie figuur 8.2):

```

Befehlsfenster - Konsole
Sitzung Bearbeiten Ansicht Einstellungen Hilfe

grewe@linux:~/FirstTry> make all
set -e; avr-gcc -MM -mmcu=atmega8 -I. -g -O3 -funsigned-char -funsigned-bitfield
s -fpack-struct -fshort-enums -Wall -Wstrict-prototypes -Wa,-ahlms=asuro.lst asu
ro.c \
| sed 's,\(.*\)\.o[ :]*,\1.o \1.d : ,g' > asuro.d; \
[ -s asuro.d ] || rm -f asuro.d
set -e; avr-gcc -MM -mmcu=atmega8 -I. -g -O3 -funsigned-char -funsigned-bitfield
s -fpack-struct -fshort-enums -Wall -Wstrict-prototypes -Wa,-ahlms=test.lst test
.c \
| sed 's,\(.*\)\.o[ :]*,\1.o \1.d : ,g' > test.d; \
[ -s test.d ] || rm -f test.d
----- begin -----
avr-gcc --version
avr-gcc (GCC) 3.3 20030512 (prerelease)
Copyright (C) 2003 Free Software Foundation, Inc.
This is free software; see the source for copying conditions. There is NO
warranty; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

avr-gcc -c -mmcu=atmega8 -I. -g -O3 -funsigned-char -funsigned-bitfields -fpack-
struct -fshort-enums -Wall -Wstrict-prototypes -Wa,-ahlms=test.lst test.c -o tes
t.o
avr-gcc -c -mmcu=atmega8 -I. -g -O3 -funsigned-char -funsigned-bitfields -fpack-
struct -fshort-enums -Wall -Wstrict-prototypes -Wa,-ahlms=asuro.lst asuro.c -o a
suro.o
avr-gcc -mmcu=atmega8 -I. -g -O3 -funsigned-char -funsigned-bitfields -fpack-str
uct -fshort-enums -Wall -Wstrict-prototypes -Wa,-ahlms=test.o test.o asuro.o -
-output test.elf -Wl,-Map=test.map,--cref -lm
avr-objcopy -O ihex -R .eeprom test.elf test.hex
avr-objcopy -j .eeprom --set-section-flags=.eeprom="alloc,load" \
--change-section-lma .eeprom=0 -O ihex test.elf test.eep
avr-objdump -h -S test.elf > test.lss
Size after:
test.elf :
section      size      addr
.text        1422      0
.data         0      8388704
.bss          0      8388704
.noinit       0      8388704
.eeprom       0      8454144
.stab        8364      0
.stabstr     2311      0
Total       12097

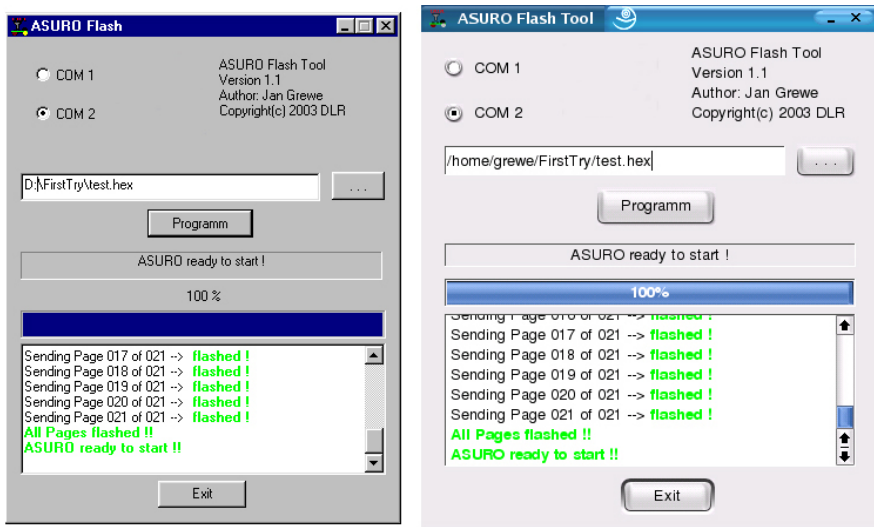
Errors: none
----- end -----
grewe@linux:~/FirstTry> █

```

Figuur 8.2.: Make all

8.3. Flash - het ASURO-programmeergereedschap

Daartoe gebruiken wij het Programma Flash werktuig (zie Fig. 8.3).



Figuur 8.3.: Flashtools voor Windows en LINUX

Om ervoor te zorgen, dat het flashen goed werkt, moet je natuurlijk de RS232- of de USB-IR-transceivers in bedrijf hebben. Dan start je het programma en kiest het interface, dat ook al bij de inbedrijfname heeft gefunctioneerd.

Je kiest het bestand "Test.hex" uit de map "C:\ASURO_src\FirstTry" (bzw. ~/ASURO/).

Dan neem je de afgebouwde en al geteste ASURO in je handen en je drukt in het Flash-Tool op de knop "Program". Bij deze test moet er tussen de ASURO en de IR-Transceiver over maximaal 50 cm afstand "zichtcontact" zijn, d.w.z. voor een optische communicatie moeten beide prints met de onderdelenkant naar elkaar gehouden worden en er mag zich geen obstakel binnen de afstand van maximaal 50cm bevinden). Je moet de ASURO nu inschakelen (d.w.z. S1 op ON zeten) voordat de statusbalk helemaal rechts is aangekomen.

Als je niet snel genoeg gereageerd hebt of als de communicatie niet werkt, moet je gewoon even de ASURO uitzetten, opnieuw op "Program" klikken en de ASURO weer inschakelen.

Zodra de communicatie werkt, zie je aan de statusindicator en aan het protocolvenster, dat de PC nu het bestand Test.hex naar de ASURO overbrengt. In de ASURO wordt het programma dan in het flashgeheugen van de processor opgeslagen, zodat het programma ook na een uitschakelen van de stroomvoorziening permanent beschikbaar blijft.

Als deze overdrachtprocedure goed is verlopen, moet je de ASURO nog even uit- en dan weer aanzetten om het programma te starten. Het zojuist opgeslagen programma wordt nu uitgevoerd en de groene Status-LED brandt helder.

8.3.1. Hoe werkt het Flashen?

Zodra je het programma Flash start, probeert de PC 10 seconden lang een verbinding met de ASURO tot stand te brengen. Als je de ASURO inschakelt, brandt de Status-LED ongeveer 1 seconde lang tweekleurig. Dit is de zogenaamde "bootfase", waarin de ASURO controleert of de PC nieuwe programma's aanlevert. Zo ja, dan worden deze geladen. Na het uit- en aanschakelen start de ASURO dan dit programma.

8.4. Problemen bij het flashen

Bij het flashen kunnen de volgende fouten optreden:

- * “c” Checksum Error. De gegevens, die de PC heeft verstuurd, wijken af van de data, die de ASURO heeft ontvangen. Dat kan door storende lichtbronnen (bijvoorbeeld TL-buizen en spaarlampen), door korte onderbrekingen in het zichtcontact, enzovoorts gebeuren.
- * “t” Timeout. Dan is het zichtcontact en/of verbinding met de ASURO afgebroken.
- * “v” Verify Error. ASURO heeft foutieve data in het Flashgeheugen geschreven. Dat kan normaal gesproken niet gebeuren en is in feite een teken aan de wand, dat het permanente geheugen (Flash-EPROM) de bovengrens van het elektronische leven nadert. Dergelijke onderdelen kunnen ongeveer 10.000 maal opnieuw geprogrammeerd worden. Het systeem probeert maximaal tienmaal het programma op te slaan. Als dat niet lukt, wordt het flashen beëindigd.



Als er bij het flashen vaker checksum errors optreden, helpt het uitschakelen of afschermen van lichtbronnen soms. Dit geldt met name voor TL-buizen.



Je moet altijd eerst de Program-toets activeren en pas daarna de ASURO inschakelen, anders werkt de transfer van de programma's niet!

8.5. . Je eerste eigen programma

Voordat wij een kort overzicht van de C-programmering behandelen, willen wij nog even een kort voorbeeld van een eigen programma presenteren. Daartoe laad je in de Programmers Notepad (in Linux s.v.p. een geschikt alternatieve editor kiezen) het bestand test.c uit de map C:\eigen programma\ASURO_src\FirstTry:

```
#include "asuro.h"
int main(void) {
  Init();
  while(1);
  return 0;
}
```

In deze eerste tests moet je steeds voor je programma de bestandsnaam test.c gebruiken, omdat het kant en klare makefile-voorbeeld (dat is een bestand, dat de PC voorschrijft, hoe een programma moet worden gecompileerd) daarop is ingericht. Het is meestal ook handig om eerst een bestaand en bekend voorbeeld als basis voor de dan volgende eigen producten te gebruiken. Als je eenmaal weet hoe het moet, kun je zelf eigen programma's en makefiles schrijven.

Nu verander je het opgeslagen programma als volgt (**attentie:** je moet erop letten, dat je de tekst precies zo aanpast als in het voorbeeld staat en ook op het verschil tussen kleine letters en hoofdletters letten):

```
#include "asuro.h"
int main(void) {
  Init();
  StatusLED(RED);
  while(1);
  return 0;
}
```

Daarna kies je in het menu weer Tools -> make (onder Linux moet je in een shell in de map "~/ASURO/" "make" intypen of de editor aanpassen, resp. configureren: Dan wacht je het resultaat van het compilerproces af , d.w.z. tot er geen nieuwe meldingen meer in het statusveld verschijnen).

Controleer nu, of de melding "Process Exit Code: 0" onderaan in het statusveld verschijnt. Die geeft aan dat de compiler het programma zonder problemen heeft kunnen verwerken. Als er een andere code verschijnt, moet je de fout met behulp van de meldingen proberen te vinden. Meestal is het handig, om met de speurtocht te beginnen op de tekstregel, waarin – volgens het statusveld – de eerste fout is opgetreden. In de editor staat helemaal linksonder het regelnummer van de cursorpositie, zodat je daarmee de regel voor de fout kunt opzoeken.

Zodra het compilerproces goed werkt, kun je het nieuwe programma flashen. Daartoe moet je weer de IR-Transceiver aansluiten, het Flashprogramma starten, het bestand test.hex en het juiste COM-interface selecteren, de ASURO weer in zichtcontact met de IR-Transceiver brengen, het programma aanklikken, de ASURO inschakelen en de overdracht van het programma afwachten.

Als ook de transfer – volgens het statusveld – goed werkt, moet je de ASURO uitschakelen en weer inschakelen. Dan volgt een spannende seconde, totdat (*Plop*) de status-LED rood oplicht. Om nu te vermijden, dat wij nog meer programmeercode zonder programmeerbasis behandelen, moeten wij voor verdere experimenten nog het volgende hoofdstuk even doorwerken.

9. C voor de ASURO

In dit hoofdstuk behandelen wij de programmeertaal C. De lezer wordt daarbij alleen met de voor de programmering van de ASURO nodige C-elementen vertrouwd gemaakt. Het wordt dus geen volledige C-cursus: voor dat doel zijn andere boeken veel beter geschikt¹.

Wij hebben C als programmeertaal gekozen, omdat deze standaard veel wordt toegepast en C-compilers voor vrijwel elke processor ter beschikking staan. Voor de ASURO gebruiken wij de Gnu-C-Compiler, omdat dit programma vrij verkrijgbaar als Freeware-Programma op de markt is en desondanks een efficiënte code genereert voor de ATmega8, ASURO's processor.

Wie al ervaring heeft met de programmering in C kan gewoon doorgaan naar hoofdstuk 9.2. Voor deze ervaren lezersgroep is de rest van dit hoofdstuk oninteressant. Wij behandelen alleen het absolute minimum aan taalelementen, om op zo eenvoudig mogelijke wijze alleen het nodige voor het werken met de ASURO te leren.

En maak je geen zorgen: als je braaf op de juiste plaatsen je haakjes en puntkomma's plaatst, is C helemaal niet zo moeilijk als het lijkt. Bovendien is de ASURO geen speeltuigrobotje voor de kleuterschool!

9.1. De grondslagen van de programmering in C

9.1.1. De basis van C

Allereerst geldt het principe, dat de processor de commando's, regel voor regel en van boven naar beneden afwerkt². Onze processor kan niet meerdere commando's tegelijkertijd verwerken, d.w.z. deze uitspraak geldt voor de ASURO-processor. Stel je dus maar voor: het ene karweitje komt na het andere...

De spaties aan het begin van een regel in de voorbeeldprogramma's zijn niet echt nodig. De structurering van de tekst met spaties en tabulators is echter zinvol, als je lange teksten overzichtelijk wilt houden.

Elk commando wordt in C met een ";" afgesloten. Met deze afspraak kan de compiler de opeenvolgende commando's onderscheiden.

Als je meerdere commando's wilt samenvatten, bijvoorbeeld voor functies, herhalingen (loops) of in voorwaardelijke paden (daarop zullen wij nog terugkomen), dan worden deze blokken in accolades ("{" , "}") samengevat.

Voorbeeld:

```
#include "asuro.h"
int main (void) {
/* Alles, was wat hier tussen de accolades staat, vormt samen een blok */
}
```

¹ Op internet staan veel boeken en cursussen over programmeren in C

² Methoden die op de sequentiële cyclus van opdrachten ingrijpen worden verderop in het hoofdstuk nog verder beschreven.

Als je enkele regels in de programmatekst wilt uitschakelen, kun je deze tekst eenvoudig in een commentaar veranderen door deze met “/*” te beginnen en met “*/” af te sluiten. Het omzetten van een enkele, losse regel in commentaar is nog eenvoudiger: daartoe kun je in deze regel voor het commentaar een “//” plaatsen³. Deze commentaartekens zijn voor de compiler een signaal, dat de betreffende tekstpassages niet moeten worden gecompileerd. Je kunt dus commentaren in het programma invoegen, zonder het compileren te storen.

9.1.2. Variabelen en datatypen

Variabelen zijn in feite “opslagplaatsen” voor gegevens. Gedurende de afloop van een programma kunnen deze locaties worden beschreven, gelezen of veranderd worden. Om een variabele te kunnen gebruiken, moet je deze eerst declareren. Daarbij moet je vastleggen, welk type variabele je wilt gaan gebruiken en eventueel ook de beginwaarde, waarmee de variabele moet worden aangelegd. Met het type leg je vast, welke soort getallen je in de variabele kunt opslaan (gehele getallen, positieve gehele getallen, decimale breuken...).

De naam voor een variabele moet daarbij met een letter beginnen (waarbij “_” eveneens tot de letters wordt gerekend) en mag ook cijfers, maar geen andere exotische tekens, bevatten. Kleine letters en hoofdletters worden in C als verschillende tekens beschouwd: dus zijn x en X in een C-programma twee verschillende variabelen. Volgens de traditie worden de kleine letters gereserveerd voor de variabele namen. De navolgende namen zijn gereserveerd en deze mag je niet meer als variabele gebruiken:

<i>auto</i>	<i>default</i>	<i>float</i>	<i>long</i>	<i>sizeof</i>	<i>union</i>
<i>break</i>	<i>do</i>	<i>for</i>	<i>register</i>	<i>static</i>	<i>unsigned</i>
<i>case</i>	<i>double</i>	<i>goto</i>	<i>return</i>	<i>struct</i>	<i>void</i>
<i>char</i>	<i>else</i>	<i>if</i>	<i>short</i>	<i>switch</i>	<i>volatile</i>
<i>const</i>	<i>enum</i>	<i>int</i>	<i>signed</i>	<i>typedef</i>	<i>while</i>
<i>continue</i>	<i>extern</i>				

De volgende datatypen zijn voor het programmeren van ASURO-software van betekenis:

Type	Symboolbereik	Opmerking
char	-128 ... +127	een bytesymbool; een byte kan een symbool uit de symboolreeks beschrijven
unsigned char	0 ... 255	char, die alleen in het positieve bereik tussen nul en 255 definieerbaar is.
int	-32768 .. +32767	symbool, dat ook twee bytes kan representeren
unsigned int	0 ... 65535	Int, d.w.z. een geheel, positief getal tussen nul en 65535
float		Een breukgetal met enkelvoudige precisie

³ “//” is een commentaarregel conform de C++ Standaard. Omdat de door ons gebruikte compiler eigenlijk een C++-Compiler is, functioneert dat hier, maar het kan bij andere compilers foutmeldingen geven.

De declaratie van variabelen kun je vastleggen: zowel als zogenaamde globale variabele buiten de main()-functie (dit betekent, dat de variabele in het hele programma ter beschikking staat) als ook binnen de main()-functie (dan geldt deze variabele alleen voor dat deel van het programma, dat zich binnen de main()-functie bevindt), of binnen een eigen functie (dan geldt de variabele alleen voor deze functie).

Maar wat kunnen wij met deze variabelen eigenlijk doen? Op de eerste plaats is het toewijzen van een waarde nogal eenvoudig:

```
a=17; // a bevat daarna een waarde 17
```

maar een variabele kan ook een waarde aannemen door een berekening:

```
a=17+23; // a is daarna 40
b=a+3; // b wordt nu 43
b=b*2; // b wordt nu 86
```

En nu in een compleet programma als voorbeeld:

```
#include "asuro.h"
int main(void) {
    int i; // i kan getallen tussen -32768 en 32767 verwerken en opnemen
    char symbool; // de variabele "symbool" kan ASCII-symbolen
                  // of getallen tussen -128 en 127 verwerken
    i=3;
    symbool =17+i; // symbool is nu 20
    i=i/2; // deling door 2, er wordt afgerond, dus i wordt nu 1!
    return 0;
}
```

C kent ook nog een paar handige afkortingen. In plaats van

```
i=i+1;
```

kunnen wij ook schrijven:

```
i++;
```

En

```
i=i-1;
```

is dan natuurlijk:

```
i--;
```

9.1.3. Compilerdirectieven

Ongetwijfeld heeft het `#include "asuro.h"` al enige vragen opgeworpen. De `#include`-directieve betekent, dat de tekst van het hier aangegeven bestand in het programma ingevoegd en eveneens gecompileerd moet worden. Onze `#include`-directieve betreft enkele functies, die voor het werken met de robot nodig zijn en die aldus in het programma ter beschikking worden gesteld. Een andere belangrijke directieve (er zijn er nog meer, maar een behandeling daarvan wordt voor dit boek te omvangrijk) is de zogenaamde tekstdefinitie. Deze tekstdefinitie heeft de vorm:

```
#define NAME tekstdefinitie
```

en men gebruikt deze vooral ter definitie van constanten, zoals bijvoorbeeld het getal Pi.

Zodra de compiler het symbool NAME nu in de broncode vindt, wordt dit symbool automatisch door de tekstdefinitie vervangen. Voor de vastlegging van NAME in `#define` gelden dezelfde regels als voor de variabelennamen. Het is inmiddels bij het programmeren in C gebruikelijk om voor de schrijfwijze van NAME in `#define` alleen maar hoofdletters te gebruiken.

Voorbeeld:

```
#include "asuro.h"
#define STARTWAARDE 33
int main(void) {
    int i;
    i= STARTWAARDE; // i is nu 33
    return 0;
}
```

Op compilerdirectieven volgt overigens geen puntkomma!

9.1.4. Voorwaardelijke structuren

Vaak is het nodig om commando's alleen dan uit te laten voeren, als er aan bepaalde voorwaarden is voldaan. Daartoe heb je beslissingstructuren (controlestructuren) nodig. De eenvoudigste controlestructuur, waarmee je beslissingen vastlegt, is de "if-else"-structuur. Formeel geldt daarvoor de volgende syntaxis:

```
if (voorwaarde)
    programmablok 1
else
    programmablok 2
```

De voorwaarde (conditie) wordt eerst onderzocht. Als deze waar (d.w.z. niet de waarde nul aanneemt), dan wordt programmeerblok 1 uitgevoerd, anders programmeerblok 2.

9. C voor de ASURO

Als je bij het nemen van een beslissing uit meerdere alternatieven wilt kiezen, kun je meerdere “else if”-structuren toepassen.

```
if (voorwaarde 1)
    programmablok 1
else if (voorwaarde 2)
    programmablok 2
else if (voorwaarde 3)
    programmablok 3
else if (voorwaarde 4)
    programmablok 4
else
    programmablok 5
```

Je kunt de volgende voorwaarden formuleren:

Operator	Betekenis
==	logisch onderzoek op gelijkheid (identiteit)
!=	logisch onderzoek op ongelijkheid
<	logisch onderzoek op kleiner dan ..
>	logisch onderzoek op groter dan ..
<=	logisch onderzoek op kleiner of gelijk
>=	logisch onderzoek op groter of gelijk

Voorbeeld:

```
#include "asuro.h"
int main(void) {
    Init ()
    while (1) {
        if (PollSwitch()>0) {StatusLED (RED);}
        else {StatusLED (GREEN);}
    }
}
```

Als een van de aanrijdingschakelaars wordt ingedrukt, schakelt de status-LED op rood en anders op groen. De overige delen van het programma worden later uitvoeriger besproken.

In C betekent “1” “waar” en “0” betekent “niet waar”. De if-structuur:

```
if (0) {StatusLED(RED);}
```

leidt ertoe, dat het commando StatusLED (RED) dus nooit zal worden uitgevoerd.

9.1.5. Herhalingstructuren

Met herhalingstructuren (loops) kun je programmablokken meermaals uitvoeren. In een while-herhalingstructuur wordt eerst een voorwaarde onderzocht. Als deze conditie waar is, voert het programma het programmablok uit en onderzoekt dan de conditie nogmaals. Deze procedure wordt dan herhaald, totdat de conditie niet meer waar is. Daarna zet het programma de afwerking na de programmablok voort.

```
while( voorwaarde)
  programmablok
```

Voorbeeld:

```
#include "asuro.h"
int main(void) {
    Init ()
    MotorDir(FWD,FWD);           // beide motoren op vooruit schakelen
    MotorSpeed(120,120);         // beide motoren op halve kracht vooruit
    StatusLED(GREEN);           // status-LED op groen schakelen
    while (PollSwitch()==0) {    // zolang er geen aanrijding plaatsvindt...
        SerWrite("Alles OK!\n",10); // ... euforie: alles loopt prima!
    }
    MotorSpeed(0,0);             // Aanrijding! Meteen stoppen!
    StatusLED(RED);              // status-LED op rood schakelen
    while (1) {
        SerWrite("Aua!\n",5);    // ....en met huilen beginnen!
    }
}
```

De "for (expr1, epr2, expr3)"-structuur is vergelijkbaar met en gelijkwaardig aan :

```
expr1;
while (expr2) {
    programmablok
    expr3;
}
```

De "for"-herhalingstructuur wordt ook vaak als aftelstructuur toegepast.

```
for (i = 0; i < n; i++)
...
...
...
```

Voorbeeld:

```
#include "asuro.h"
int main(void) {
    Init ()
    int teller;                    // variabele voor het aftellen declareren
    for (teller=0; teller<10; teller++) { // tienmaal herhalen:
        SerWrite("Klaar AF!\n",12);    // "Klaar AF!" melden
    }
    MotorDir(FWD,FWD);             // Beide motoren op vooruit
    MotorSpeed(120,120);           // Beide motoren op halve kracht vooruit
```

De expressie “while(1)”, die hetzelfde resultaat oplevert als “for (;;)” is een oneindige herhalingsloop. De herhalingsfunctie kan niet worden beëindigd, omdat de conditie voor een beëindiging nooit onwaar (d.w.z. 0) wordt.

Een andere herhalingsfunctie is de “do”-herhaling

```
do
  programmablok
while( voorwaarde);
```

In tegenstelling tot de “while”-herhalingsfunctie wordt de voorwaarde pas aan het einde van het programmablok op waarheidsgehalte getoetst. De “while”-herhalingfunctie wordt dus minstens eenmaal afgewerkt.

9.1.6. Functies

In C worden functies altijd gedefinieerd volgens een vast patroon:

```
Functietype FunctieNaam (ParameterType 1 ParameterNaam 1,
ParameterType 2 ParameterNaam 2, ...)
```

Geweldig, functiedefinities! Maar wat heb ik daaraan???

Functies zijn heel handig, maar maken alles wel wat gecompliceerder. Je kunt de nu volgende achtergronden voor het gebruik van functies trouwens ook wel later lezen...

Het komt vaak voor, dat delen van het programma op meerdere plaatsen worden toegepast. Dan kun je die gedeeltes steeds weer opnieuw schrijven (dat is natuurlijk zeer lastig, inefficiënt en maakt het programma onoverzichtelijk), maar je kunt zo'n gedeelte ook als functie declareren.

In een aantal gevallen wil je een functie ook een of meerdere gegevens meegeven. Zo kun je je voorstellen, dat het handiger is direct in de functie RijdEenStukVooruit() de gegevens zoals de snelheid, de tijdsduur of de afstand te specificeren. Dat gebeurt met behulp van zogenaamde parameters.

Ook is het gebruikelijk, dat een functie een waarde teruglevert. Dat kun je je voorstellen voor de functie HoeveelToetsenZijnErIngedrukt(). De teruggegeven waarde wordt overgedragen met behulp van de zogenaamde returnwaarde van de functie, die op een gegeven moment ergens in de functie wordt gevuld. Om deze redenen eindigt elke functiedefinitie ook met een return; of met return GETAL;

Een bijzondere functie is de main ()-functie. Deze vormt een startpunt voor een programma. In de ASURO begint het programma na het inschakelen met de main ()-functie. De main ()-functie moet in elk C-programma worden gedefinieerd!

9. C voor de ASURO

Nadat wij de datatypes hebben leren kennen en wat over functies hebben ervaren, testen we nu een kleine voorbeeldfunctie, die twee 8-bitgetallen vermenigvuldigt en het resultaat moet teruggeven.

```
int Mult(char a, char b)
/* De functie levert een intwaarde terug, heet Mult en werkt met een input van twee char als parameter
*/
{
    // begin van de functie
    int c;      // variabele c wordt als int gedeclareerd
    c = a * b;  // bereken c
    return c;   // schrijf c terug
}              // einde van de functie Mult
```

Nu volgt nog een klein programma, dat de zojuist gedefinieerde functie uitvoert:

```
int main (void)      // de functie main levert altijd een int terug,
                    // en werkt zonder parameter
{
    // begin van de functie
    char mult1,mult2; // definitie van twee charvariabelen
    int erg;          // definitie van een intvariabele, die het resultaat
                    // van de vermenigvuldiging der variabelen mult1 en mult2
                    // moet bevatten
    mult1 = 2;       // toewijzing
    mult2 = 10;      // toewijzing
    erg = Mult(mult1,mult2); // Oproep van de tevoren gedefinieerde functie Mult
    return 0;
}                    // einde van de functie main
```

9.1.7. Pointers en vectoren

Wij zullen pointers en vectoren hier slechts behandelen voorzover dat voor het werken met de ASURO nodig is.

Als wij de lijnvolgersensoren (dat zijn de sensoren voor de hodometrie) willen aflezen, moeten wij vectoren toepassen. De declaratie van deze elementen is nogal eenvoudig:

```
int IData[2];
int oData[2];
```

Zoals je ziet, definiëren wij voor de lijnvolgersensoren, d.w.z. voor de hodometrie, twee vectoren (IData, oData) met twee elementen. Na de oproep van de bijbehorende ASURO-functie Functie (LineData(), OdometrieData ()) bevat Element [0] de waarde van de linkse sensor en Element [1] de waarde van de rechtse sensor.

Daartoe een voorbeeldje:

Als een der lijnvolgersensoren, bijvoorbeeld de rechtse meer licht ontvangt dan de linkse, moet commando 1 worden uitgevoerd en anders commando 2.

```
int IData[2];           // geheugenplaats voor de meetgegevens reserveren
LineData(IData);       // meetgegevens inlezen
if (IData[1] > IData[0])
    commando1;
else
    commando2;
```

Om de bestaande functies voor de seriële interfaces (SerWrite(), SerRead()) te kunnen gebruiken, moeten wij strings toepassen. Deze strings worden als volgt gedeclareerd:

```
char message [] = "Dit is een tekst";
```

In de ASURO kun je een string wegsturen door het uitvoeren van de functie SerWrite() met de juiste parameterwaarden. De eerste parameter bevat de tekst, respectievelijk de stringinhoud en de tweede parameter geeft aan, hoeveel stringsymbolen moeten worden geschreven.

```
SerWrite(message,20);
```

respectievelijk:

```
SerWrite("Dit is een tekst ",20);
```

zendt via de IR-interface uit: " Dit is een tekst ".

Als je strings (d.w.z. reeksen symbolen) wilt ontvangen, kun je in het ASURO-systeem de functie `SerRead ()` gebruiken. De eerste parameter bevat de stringvariabele, waarin de te ontvangen symbolen worden opgeslagen. De tweede parameter beschrijft, hoeveel symbolen je wilt lezen. De derde parameter is een zogenaamde time-outparameter. Als er binnen de hierin aangegeven tijdsperiode (in clockimpulsen van de processor) geeningangsymbolen worden ontvangen, sluit de functie de ontvangst af. Indien wij als timeout 0 aangeven, wacht de functie tot alle symbolen zijn ontvangen.

Ook in dit geval even een voorbeeld:

ASURO moet bijvoorbeeld "Hallo hier ben ik!" via de IR-interface ontvangen:

```
char message [] = "01234567890123456789";
```

Met deze definitie hebben wij voldoende plaats voor de te ontvangen tekst gereserveerd. De hier aangegeven string moet groot genoeg zijn om de te ontvangen tekst op te nemen.

```
SerRead(message,18,0);
```

Lees nu 18 symbolen en wacht daarbij zolang, totdat alle 18 symbolen zijn aangekomen. Wij gaan er nu vanuit, dat de string "Hallo hier ben ik!" intussen uitgezonden is. Dan ziet de tevoren gedefinieerde string "message" er nu als volgt uit:

```
Hallo hier ben ik!89
```

De ontvangen symbolen hebben dus de eerste 18 symbolen van message vervangen.

9.2. Beschrijving van de ASURO-functies

Om de programmering van de ASURO zo eenvoudig mogelijk te maken, staan er enkele kant en klare functies ter beschikking. Deze vormen nu niet direct een optimale bibliotheek en voor een aantal toepassingen is het vast beter, eigen functies te schrijven.

De functies worden volgens het boekje officieel in de vorm van de declaraties gespecificeerd. Wie daarmee niets kan beginnen, moet gewoon even de voorbeelden inspecteren.

Ter verduidelijking: functies, die iets besturen, zoals motorfuncties of ook de displayfuncties, leggen waarden vast, die zo lang geldig blijven totdat zij weer veranderd worden. Dat wil zeggen, dat een groene Status-LED zo lang groen blijft, totdat deze op een andere kleur omgeschakeld of uitgeschakeld wordt.

9.2.1. void Init(void)

Deze functie brengt de microcontroller in zijn initiële uitgangspositie en moet steeds aan het begin van een programma worden opgeroepen. Als deze initialisatie niet plaatsvindt, weet de processor niet eens, wat er met zijn pootjes gebeurt.

Een ASURO-programma moet op zijn minst aan de volgende structuur voldoen:

```
#include "asuro.h"
int main(void) {
    // hier worden de benodigde variabelen gedeclareerd
    Init();
    // op deze plaats staan je eigen ideeën geprogrammeerd
    while(1); // eindeloze herhalingsloop
    return 0; // deze regel wordt in het programma niet bereikt
}
```

Waartoe dient eigenlijk de eindeloze herhalingsloop op het einde van de main () –functie?

Normaal gesproken wordt een C-programma bij het bereiken van de regel “return 0;” beëindigd. Bij de ASURO is het echter denkbaar, dat het programma dan verder gaat in gedeelten van vroeger geflashte programma’s of dat de ASURO het programma opnieuw start, wat tot allerlei vreemde effecten kan leiden. Om dat te vermijden “vangen” wij het programma na het afwerken van de bewust gegeven opdrachten in een eindeloze herhalingsloop, die een goed gedefinieerd einde voor het programma vormt.

9.2.2. void StatusLED(unsigned char color)

Met behulp van deze functie kun je de Status-LED (D12) laten oplichten. De toegelaten parameters zijn OFF, GREEN, RED of YELLOW

Voorbeeld:

De Status-LED moet rood oplichten:

```
StatusLED(RED);
```

En nu nogmaals een voorbeeld in een compleet programma:

```
#include "asuro.h"
int main(void) {
    Init();
    StatusLED(YELLOW);
    while(1); // eindeloze herhalingsloop
    return 0;
}
```

9.2.3. void FrontLED(unsigned char status)

Hiermee kun je de Front-LED (D11) in- en uitschakelen. De toegelaten parameters zijn ON respectievelijk OFF.

Voorbeeld:

De Front-LED moet oplichten:

```
FrontLED(ON);
```

9.2.4. void BackLED(unsigned char left, unsigned char right)

Hiermee kun je de Back-LEDs (D15 en D16) in- en uitschakelen. De eerste parameter beschrijft de toestand van de linkse Back-LED (D15), de tweede parameter die van de rechtse Back-LED (D16). De toegelaten parameters zijn ON respectievelijk OFF.

Voorbeeld:

Je wilt rechtse BackLED(D16) aan- en de linkse (D15) uitschakelen:

```
BackLED(OFF,ON);
```

9.2.5. void Sleep(unsigned char time72kHz)

De sleep-functie laat de processor een bepaalde, instelbare tijd wachten. Daarmee kun je prima een pauze programmeren. De functie is gebaseerd op een 72kHz-Timer en je kunt als parameter een getal met een maximale waarde van 255 kiezen (unsigned char)⁴.

Voorbeeld:

De processor moet ca. 3ms wachten $\Rightarrow \frac{0,003s}{\frac{1}{72KHz}} = 216$. Je moet dus, om 3ms lang te wachten, de sleep-functie als volgt oproepen:

```
Sleep (216) ;
```

9.2.6. void MotorDir(unsigned char left_dir, unsigned char right_dir)

Deze functie bestuurt de draairichting van beide motoren. Je moet deze functie vóór het instellen van de snelheid oproepen. De toegelaten parameters zijn FWD (voorwaarts), RWD (achteruit), BREAK (remmen, respectievelijk stilstand, waartoe de motoren met behulp van de transistorbruggen kortgesloten worden) en FREE (vrijloop).

Voorbeeld:

Je wilt, dat de linkse motor voorwaarts draait en de rechtse motor blijft staan.

```
MotorDir(FWD,BREAK);
```

⁴ Dat is een boosaardig doel van de auteurs om je tot nadenken te dwingen ;-)

9.2.7. void MotorSpeed(unsigned char left_speed, unsigned char right_speed)

Met behulp van deze functie kun je de snelheid van de motoren regelen. De maximale snelheid is daarbij 255 (unsigned char). De motor begint pas bij een waarde van ca. 60 te draaien. Deze minimale waarde hangt er sterk vanaf hoe je de robot mechanisch hebt gebouwd. De hier gekozen parameter beschrijft eigenlijk alleen maar, hoeveel elektrische energie aan de motoren moet worden toegevoerd. Dan hangt het nog van andere factoren af, zoals de wrijving of de helling, hoe snel de motoren in zo'n geval zullen draaien.



Zodra je deze functie oproept, kan ASURO beginnen met rijden. Soms is het resultaat van je programma echter onverwacht en ongewenst, zodat je steeds moet opletten, dat de ASURO zichzelf en anderen door onverwachte bewegingen geen schade kan toebrengen.

Voorbeeld:

Je wilt de linkse motor met volle kracht en de rechtse motor helemaal niet laten draaien. De draairichting is met de functie MotorDir() al van te voren vastgelegd.

```
MotorSpeed (255,0) ;
```

9.2.8. void SerWrite(unsigned char *data, unsigned char length)

In dit geval zendt de ASURO via het seriële IR-interface met 2400Bit/s, No-Parity, 1 StopBit, NoFlowControl gegevens uit. Met deze instelling heb je ook tevoren de tests voor de IR-Transceiver doorgevoerd (is dat geen toeval?). In de eerste parameter van de functie leg je het adres vast, waarnaar je de data wilt versturen. De tweede parameter legt vast, hoeveel bytes je wilt versturen.

Voorbeeld:

Je wilt de string „Hé, jij daar!” via het seriële IR-interface uitzenden.

```
SerWrite(“Hé, jij daar!” ,12);
```

9.2.9. void SerRead(unsigned char *data, unsigned char length, unsigned int timeout)

Als je al gegevens via het seriële IR-interface kunt uitzenden, wil je misschien ook wel wat kunnen ontvangen. Daartoe hebben wij de hier beschreven functie geschreven. De eerste parameter is de pointer, die op de geheugenplaats wijst, waar je de ontvangen data wilt opslaan. De tweede parameter legt vast, hoeveel bytes je verwacht.

De derde en laatste parameter vormt de time-outinstelling. Daarmee kun je ervoor zorgen, dat de functie niet oneindig lang op data wacht. Als je na verloop van een instelbare tijd geen gegevens meer ontvangt, sluit de functie de ontvangst af. In dat geval overschrijft de functie het allereerste byte van de ontvangen data met de letter 'T' (Timeout). Als je voor de derde parameter de waarde '0' kiest, wacht de functie zolang, totdat het in de tweede parameter aangegeven aantal bytes ook werkelijk is ontvangen.

Voorbeeld:

Je wilt de string „Vertrek!” ontvangen en je wilt controleren, dat ook werkelijk alle symbolen bij de ASURO zijn aangekomen voordat het programma verder loopt. Opmerking: Er worden alleen maar 8 symbolen ontvangen. Er vindt geen controle plaats, of inderdaad „Vertrek!” is aangekomen.

```
#include "asuro.h"
int main(void) {
    char daten[8]; // geheugenplaatsen reserveren
    Init();
    SerRead(daten,8,0); // gegevens lezen
    MotorDir(FWD,FWD);
    MotorSpeed(120,120);
    while(1); // eindeloze herhalingsloop
    return 0;
}
.
```

9.2.10. void LineData(unsigned int *data)

Met behulp van deze functie lees je de fototransistors aan de onderkant van de ASURO af. Je moet daartoe het gewenste geheugenadres aangeven, dat twee integerwaarden kan opnemen. De functie schrijft dan de A/D-converteerwaardes van beide fototransistors in deze geheugenplaats. Het eerste integergetal beschrijft de converteerwaarde van de linkse (T9), het tweede integergetal de converteerwaarde van de rechtse fototransistor (T10). Een waarde '1023' beschrijft een maximale lichtintensiteit, en bij '0' behoort absolute duisternis. Normaal gesproken zul je deze extreme randwaarden nooit tegenkomen en bevindt zich een meetwaarde steeds daartussen.

Voorbeeld:

Aflesen der fototransistors (T9, T10)

```
unsigned int data[2]; // geheugenplaatsen reserveren
.
.
LineData(data);
```

data[0] bevat de waarde van de linkse fototransistor (T9).

data[1] bevat de waarde van de rechtse fototransistor (T10).

Ja ja, ook nu weer een voorbeeldje in een compleet programma:

```
#include "asuro.h"           // de lijnvolging in een eenvoudig voorbeeld
int main(void) {
  unsigned int data[2];      // geheugenplaatsen reserveren
  Init();
  FrontLED(ON);             // schakel de lijnverlichting aan
  MotorDir(FWD,FWD);        // Beide motoren vooruit schakelen
  while(1){                 // eindeloze herhaling, ASURO blijft

    // oneindig lang een lijn volgen
    LineData(data);         // actuele lichtwaarden van de
                            // fototransistors aflezen
    if (data [0] > data [1] ) // indien het links helderder dan rechts is...
      {MotorSpeed(200,150);} // ... dan links meer gas geven...
    else
      {MotorSpeed(150,200);} // ... anders rechts meer gas geven!
  }
  return 0;
}
```

9.2.11. void (H)odometrieData(unsigned int *data)

Wij lezen nu de reflexlichtsensor af. De functie schakelt de LED's (D13, D14) in en leest de A/D-converteerwaardes van de fototransistors (T11, T12) af. Net zoals in de functie LineData () geven wij een geheugenplaats voor twee integergetallen aan, die vervolgens door de functie wordt gevuld. Het eerste integergetal beschrijft de converteerwaarde van de linkse (T11), het tweede integergetal de converteerwaarde van de rechtse fototransistor (T12). Een waarde '1023'⁵ beschrijft een absolute duisternis en bij '0' behoort een maximale lichtintensiteit. Normaal gesproken zul je deze extreme randwaarden nooit tegenkomen en bevindt zich een meetwaarde steeds daartussen.

Voorbeeld:

Het aflezen van de reflexlichtsensor

```
unsigned int data[2];        // geheugenplaatsen reserveren
.
.
HodometrieData(data);
```

data[0] bevat de waarde van de linkse fototransistor (T11)

data[1] bevat de waarde van de rechtse fototransistor (T12)

Ter voorkoming van misverstanden: HodometrieData(data) leest geen toerentallen af, maar alleen de actuele lichtintensiteit op de decodeerschijf van de lichtsensor. De omrekening van de intensiteitwaarden, het tellen van de licht/donkerwisselingen en de berekening van het toerental van het bijbehorende wiel is de taak van de programmeur!

⁵ Dat de betekenis precies andersom is, dan bij de fototransistoren van de lijnvolger, is schakeltechnisch een voorwaarde en noodzakelijk om de schakeling eenvoudig te kunnen houden.

9.2.12. unsigned char PollSwitch(void)

De functie leest de drukknoppen (K1-K6) af. Deze functie levert een bytewaarde terug, die de informatie bevat, welke schakelaars op dat moment zijn gedrukt. Daarin activeert drukknop 1 bitnummer 5 en drukknop 6 bitnummer 0.

Bit.

Bit0 (1) -> K6

Bit1 (2) -> K5

Bit2 (4) -> K4

Bit3 (8) -> K3

Bit4 (16) -> K2

Bit5 (32) -> K1

Als dus de drukknoppen 1, 3 en 5 ingedrukt zijn, antwoordt de functie met de waarde $32 + 8 + 2 = 42$.

Eventueel moet je de functie opeenvolgend meermaals oproepen om het “juiste” resultaat af te kunnen lezen. De opgeladen condensator C7 moet zich eerst kunnen ontladen en daarvoor is wat tijd nodig.

Als je de A/D-converter te vroeg afleest, kunnen allerlei spanningwaarden spontaan worden weergegeven.

Voorbeeld:

```
unsigned char drukknop;  
.  
.  
drukknop = PollSwitch();  
if (drukknop >0) {MotorSpeed(0,0);}
```

Zo, dat is dat. Nu mag je er zelf op los programmeren!

MEER INFO

WWW.AREXX.COM

WWW.ROBOTERNETZ.DE

***en de Duitse boeken “MEHR SPAß MIT ASURO” BAND I und II
of het Engelse boek “MORE FUN WITH ASURO” volume I***

Deel IV. Bijlagen

A. Onderdelenlijst:

Naast een pingpongballetje heb je voor de bouw van de ASURO de volgende onderdelen nodig:

- 1x Printplaat ASURO
- 2x Motoren van het type Igarashi 2025-02
- 1x Diode 1N4001
- 8x Diodes 1N4148
- 4x Transistors BC 327/40 of BC 328/40
- 4x Transistors BC 337/40 of BC 338/40
- 1x IC (Integrated Circuit) CD 4081BE
- 1x Processor ATmega 8L-8PC (in de fabriek al voorgeprogrammeerd)
- 1x IR-ontvanger SFH 5110-36
- 1x IR-LED SFH415-U
- 2x Fototransistors SFH300
- 3x LEDs 5mm rood met helder diffuus licht
- 1x Duo-LED 3mm rood/groen
- 2x Side-Fototransistors LPT80A
- 2x Side-LEDs IRL80A
- 1x kristaloscillator 8MHz
- 2x Elco 220 μ F minimaal 10V RM 3,5/10
- 4x ceramische condensatoren 100nF RM 5,08
- 2x ceramische condensatoren 4,7nF RM 2,54
- 1x 100 1/4 W 5%
- 2x 220 1/4 W 5%
- 4x 470 1/4 W 5%
- 10x 1K 1/4 W 5%
- 1x 1K 1/4 W 1%
- 3x 2K 1/4 W 1%
- 2x 4,7K 1/4 W 5%
- 1x 8,2K 1/4 W 1%
- 1x 10K 1/4 W 1%
- 1x 12K 1/4 W 1%
- 1x 16K 1/4 W 1%
- 1x 20K 1/4 W 5%
- 1x 33K 1/4 W 1%
- 1x 68K 1/4 W 1%
- 1x 1M 1/4 W 5%
- 3x Voetjes 14 pol.
- 6x Detectordrukknoppen
- 1x Schakelaar
- 1x Batterijhouder
- 1x Jumper
- 1x Insteekstrip 2pol RM 2.5
- 2x Tandwielen 10/50 tandjes; 3,1mm boring (modulewaarde 0,5)
- 2x Tandwielen 12/50 tandjes; 3,1mm boring (modulewaarde 0,5)
- 2x Motorronddsel met 10 tandjes
- 2x Stelingen voor de 3mm As

A. Onderdelenlijst

4x Kabelbinders
1x Kabelbinder (geschikt voor herhaald openen en sluiten)
2x Gummibanden 38mm
2x Messing as 42mm lang, 3mm diameter
2x Messing as 24,5mm lang, 3mm diameter
ca. 15cm litzedraad rood 0,14mm
ca. 15cm litzedraad zwart 0,14mm
2x Decodeerschijven (zie 2.4)

Voor de bijbehorende RS-232-IR-Transceiver heb je nog de volgende onderdelen nodig:

1x Printplaat IR-RS232-Transceiver
3x Diodes 1N4148
1x Zenerdiode ZPD5.1
1x Transistor BC547 A,B of C of BC548 A,B of C
1x IC (Integrated Circuit) NE555N
1x IR-ontvanger SFH 5110-36
1x IR-LED SFH415-U
1x Elco 100 μ F minimaal 16V RM 2,5/6
2x ceramische condensatoren 100nF RM 5,08
1x ceramische condensator 680pF RM 2,54
1x 220 1/4 W 5% of beter
1x 470 1/4 W 5% of beter
1x 4,7K 1/4 W 5% of beter
1x 10K 1/4 W 5%
2x 20K 1/4 W 5% of beter
1x Trimmer 10K rechtopstaand RM 2,5/5
1x Voet 8 polig
1x 9-polige SUB-D-aansluiting

**Als alternatief kun je ook kiezen voor een USB-Tranceiver:
De USB-IR-Transceiver wordt compleet afgebouwd geleverd!**

Meer ASURO PRODUCTEN ZIJN:

- ASURO Boek "*Mehr spaß mit ASURO*"
- ASURO *experimenteer set*
- ASURO *ultra sound, snake vision en display KITS*

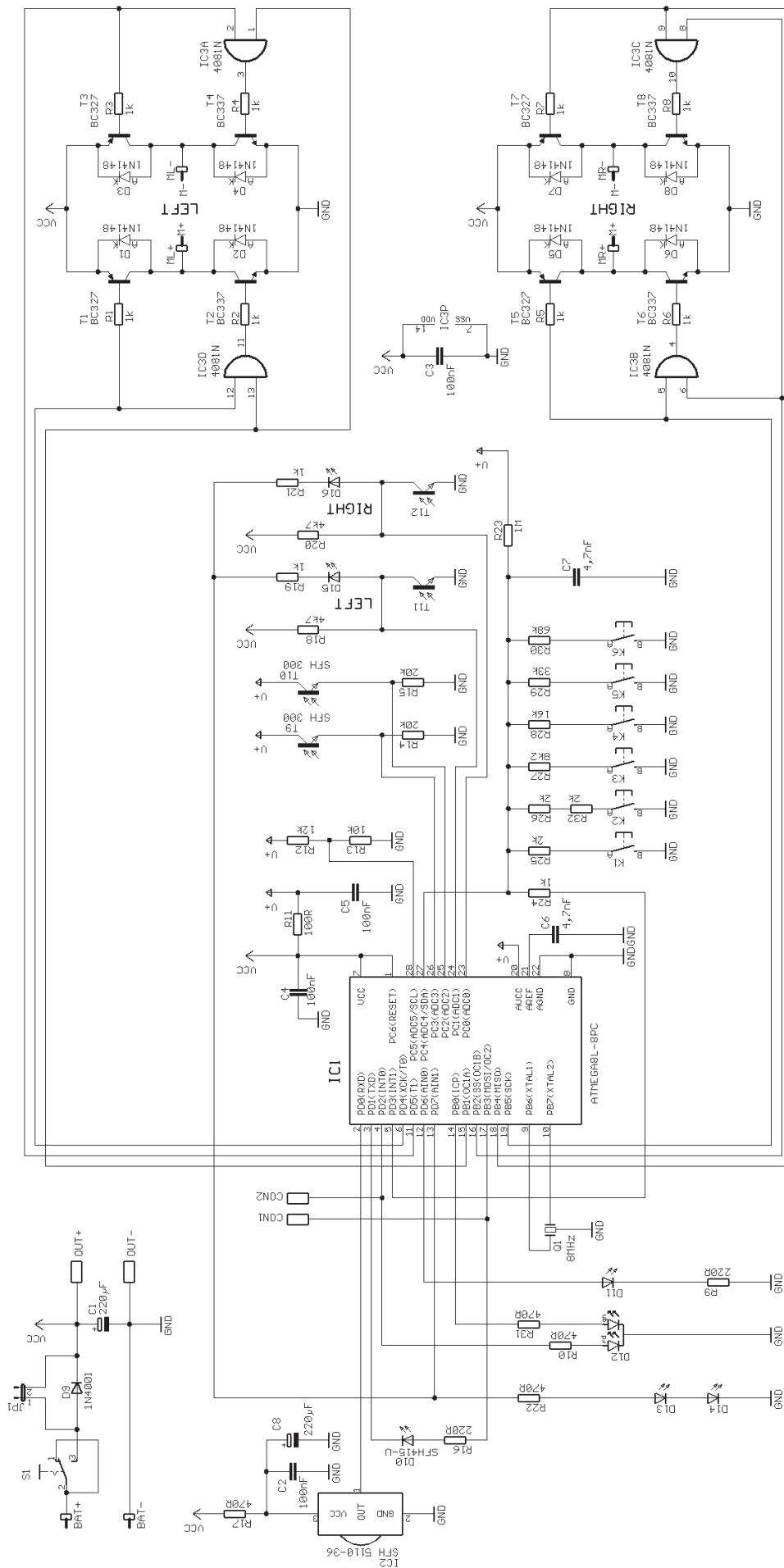
Voor meer informatie over de ASURO, "GOOGLE" op ASURO ROBOT

Technische vragen? zie;

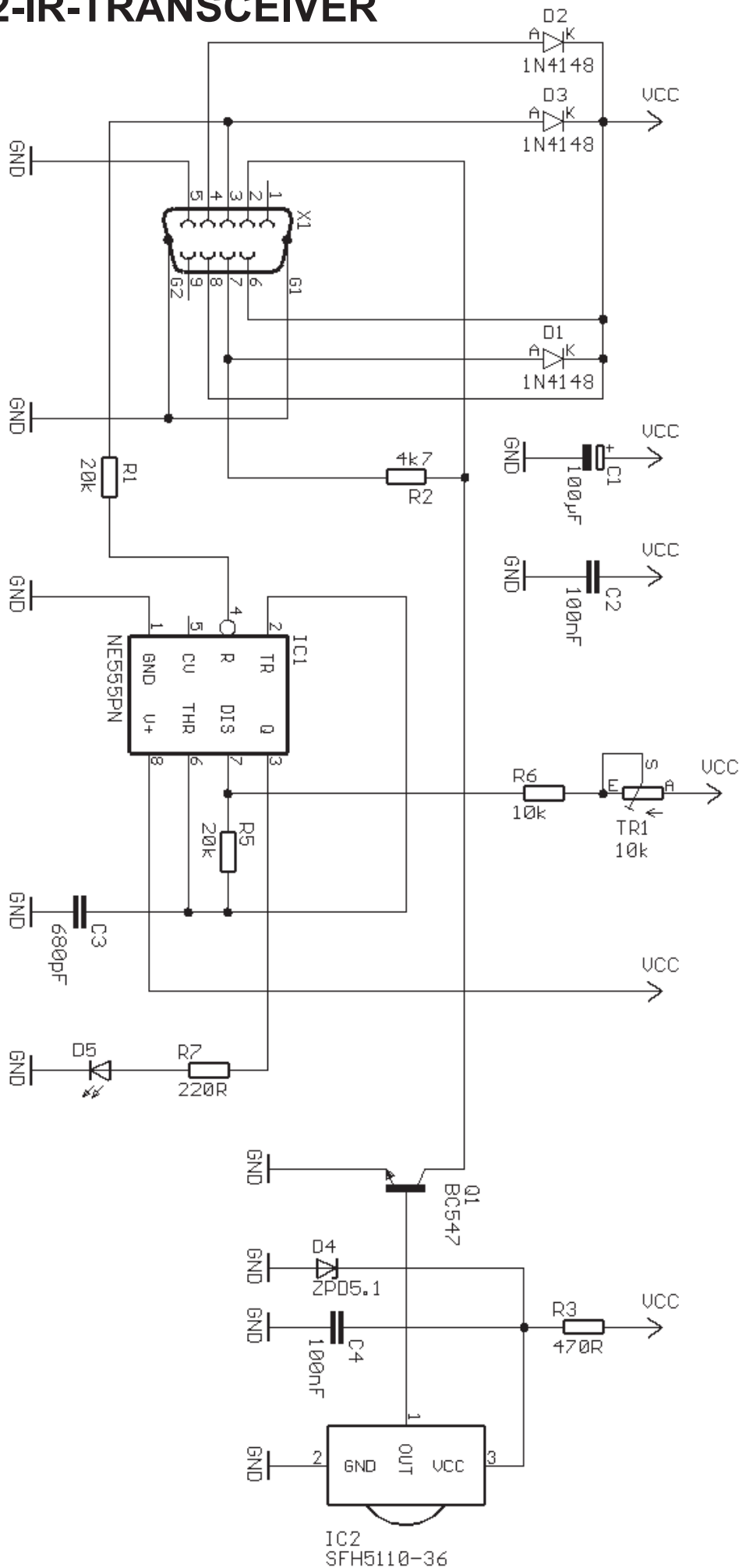
www.arexx.com --> Forum

www.Roboternetz.de --> ASURO FORUM

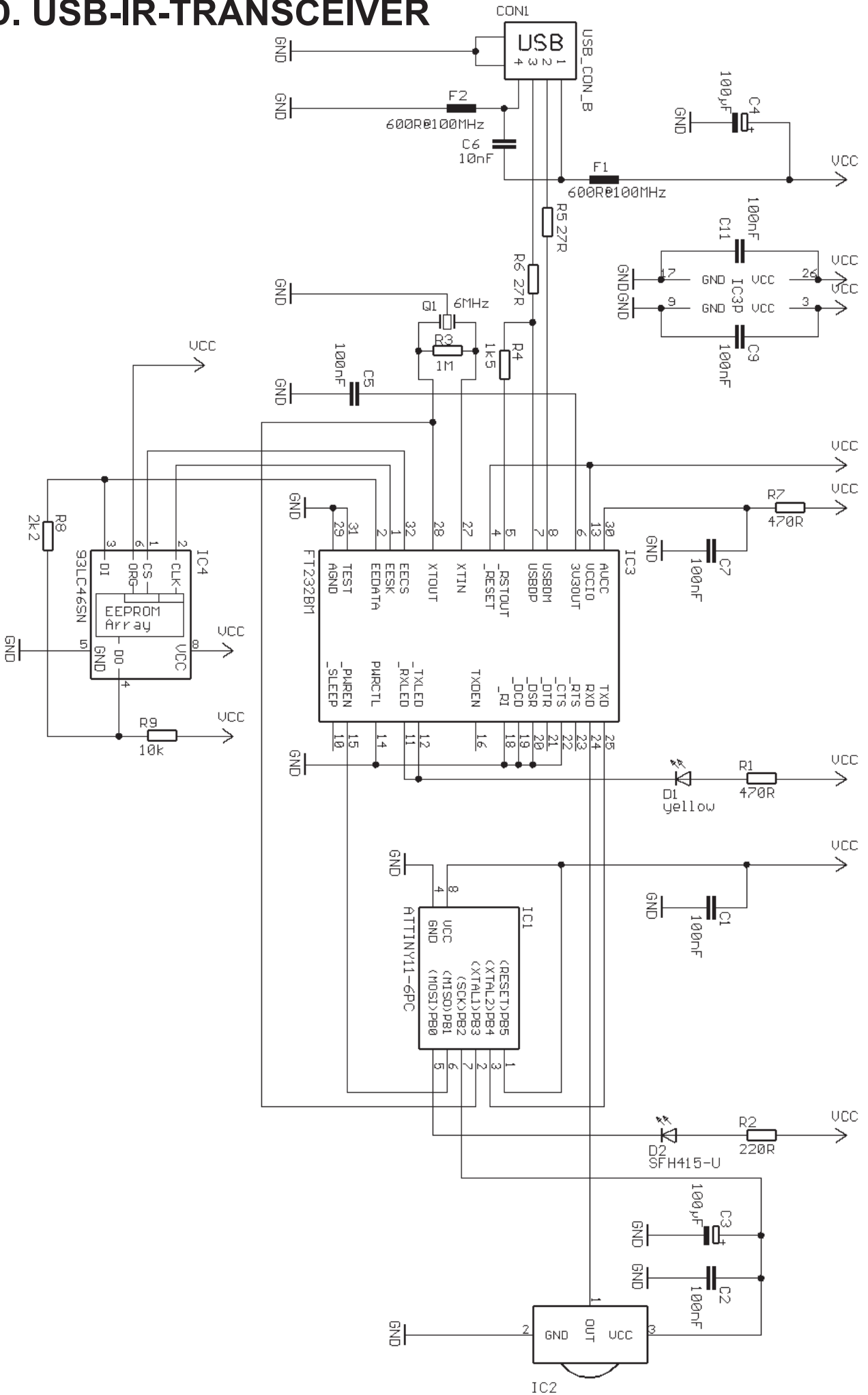
B. Schema ASURO



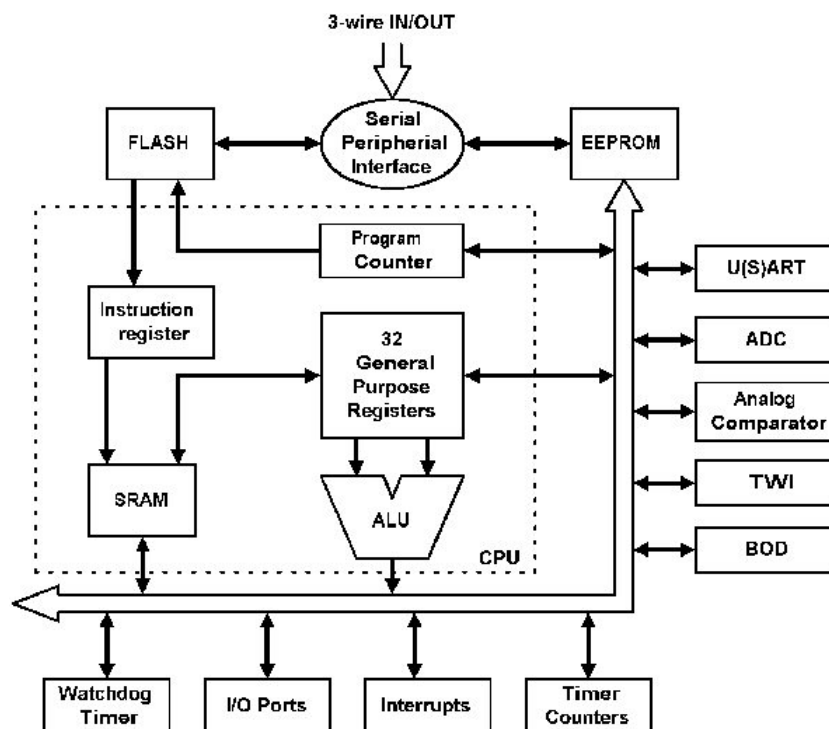
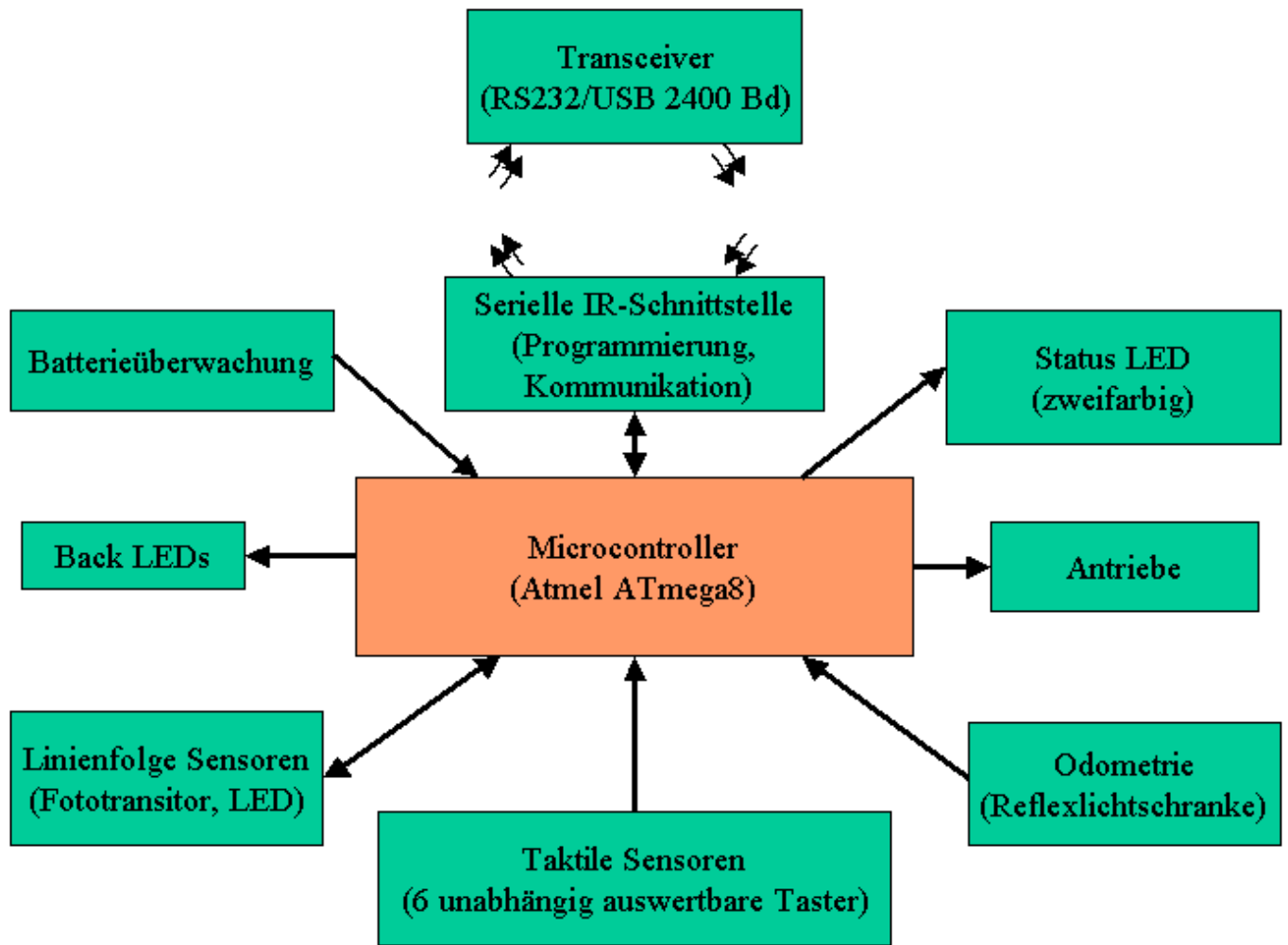
C. RS232-IR-TRANSCIEVER



D. USB-IR-TRANSCIVER

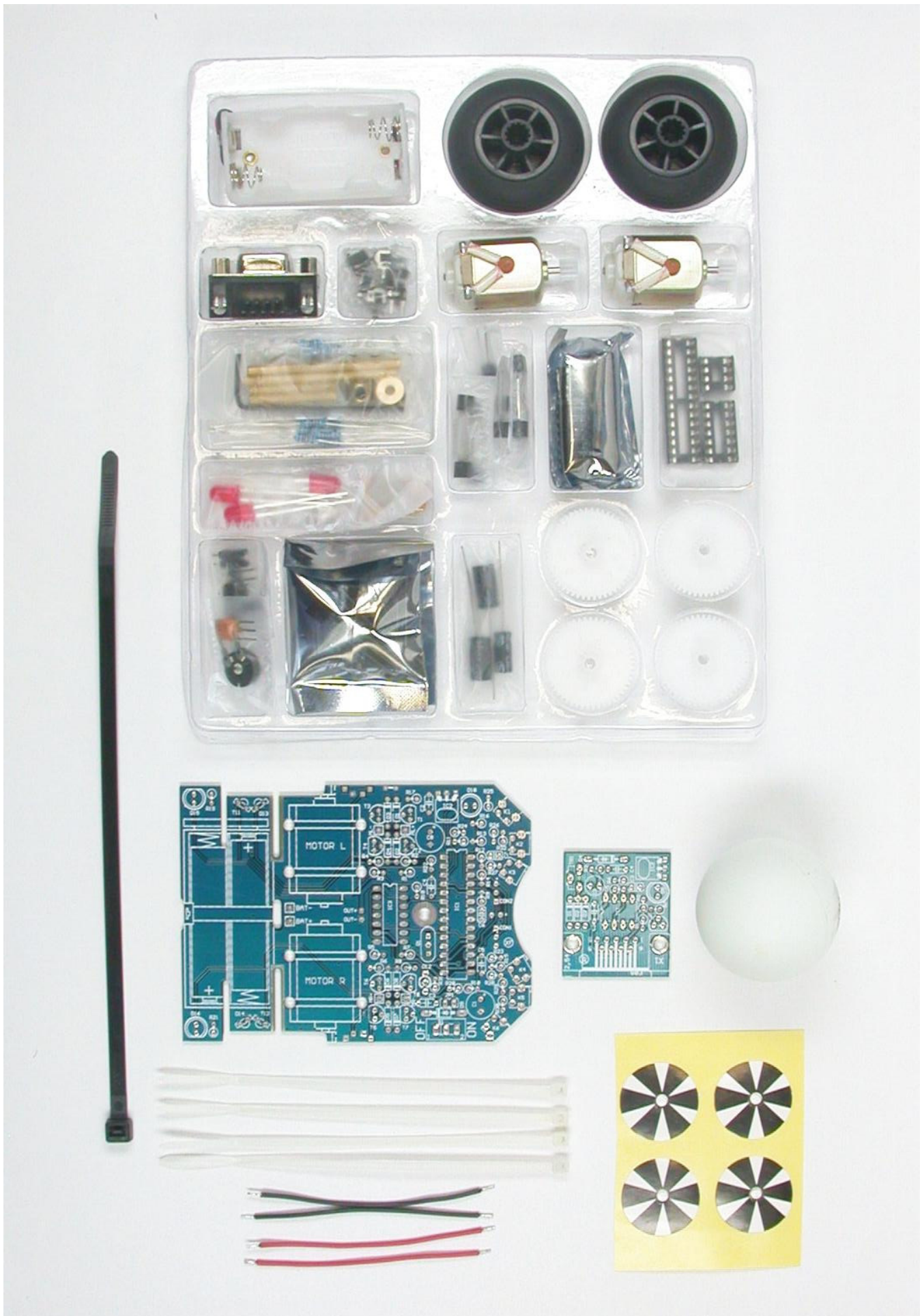


E. BLOKSCHHEMA ASURO (Duits)




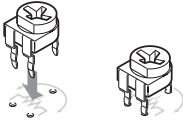



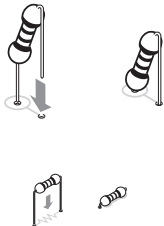


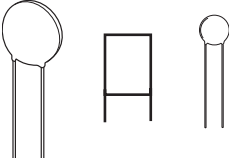



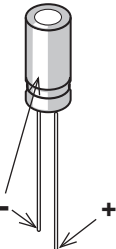
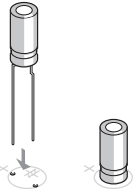

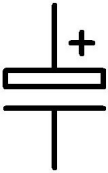
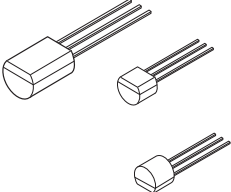
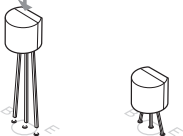

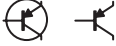


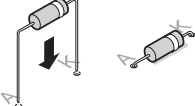


F. BLOKSCHHEMA AVR PROCESSOR

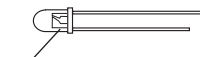
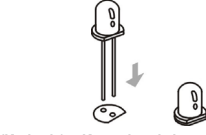

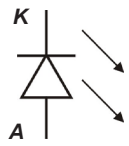



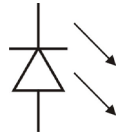
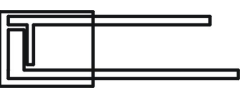
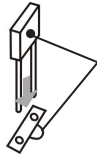

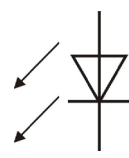
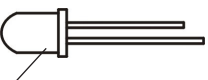
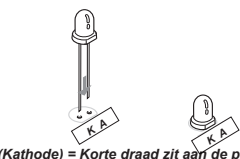

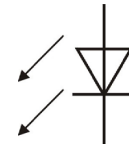
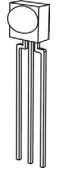
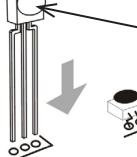
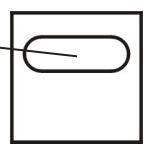


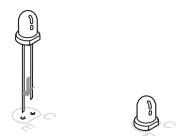

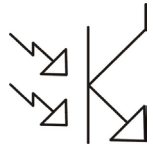
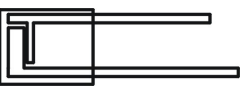
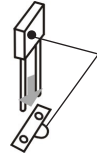

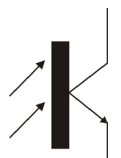

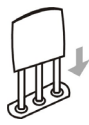

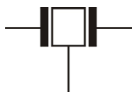
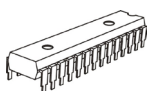
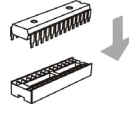


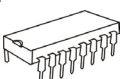
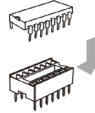
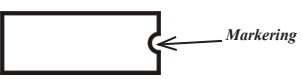
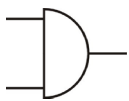


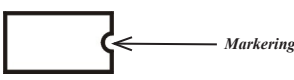

G. LEVEROMVANG ASURO

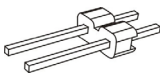
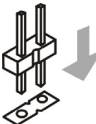

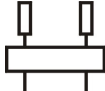
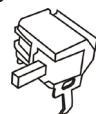
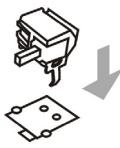
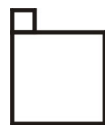
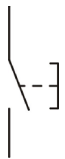

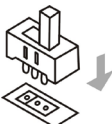
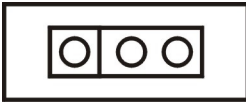

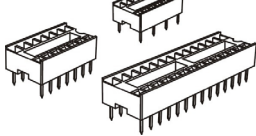
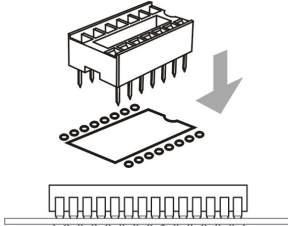

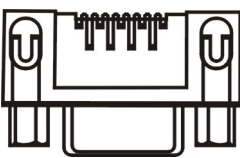
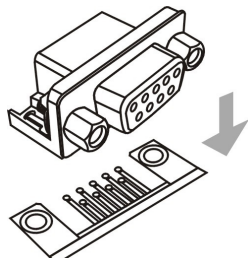

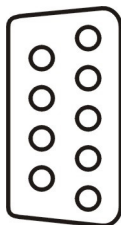
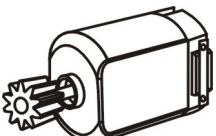
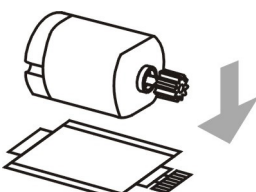
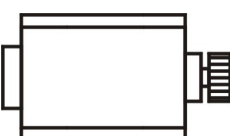


H. ELEKTRONICA-COMPONENTEN

Bij een componentonderdeel met een markering is de inbouwrichting op de print en de polariteit van het onderdeel extreem belangrijk! Een foute montage kan het onderdeel of de gehele schakeling beschadigen!

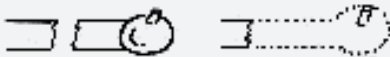
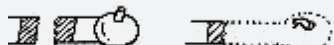
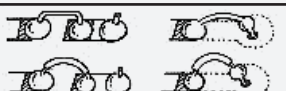
Komponent	Montage	Printsymbool	Schema-symbool
<p>Regelweerstand (Potentiometer)</p> 			
<p>Weerstand</p> 			
<p>Kondensator</p> 			
<p>Electrolytische Condensator (ELCO)</p> 			
<p>Transistor</p> 		<p><i>B en E zijn alleen als voorbeeld, de werkelijke positie kan per type verschillen</i></p> 	<p>PNP type </p> <p>NPN type </p>
<p>(Zener)Diode</p> 			

Komponent	Montage	Printsymbool	Schema-symbool
LED rood D15-D16  Rood	 <i>K (Kathode) = Korte draad zit aan de platte kant!</i>	 <i>Platte kant is K = -</i>	 K A
DUO LED D12 	 <i>korte draad komt in het kwadratische gat !</i>		
IR-LED IRL80A Kleurloze type 	 <i>Let op de markerings punt!</i>		
IR-LED SFH415-U  Zwart	 <i>K (Kathode) = Korte draad zit aan de platte kant!</i>		
IR-Receiver SFH5110 	 <i>Let op! kant met markerings punt moet naar boven wijzen</i>		
FOTO-TRANSISTOR SFH-300  Farblos	 <i>Op de polariteit letten!</i>		
FOTO-TRANSISTOR LPT80A Roze type 	 <i>Let op de markerings punt!</i>		
Keramische resonator Q1 	 <i>Geen polariteit</i>		
IC Atmega 8L 	 <i>Op de polariteit letten!</i>	 <i>Markering</i>	
IC CD4081 	 <i>Op de polariteit letten!</i>	 <i>Markering</i>	
IC NE555 	 <i>Op de polariteit letten!</i>	 <i>Markering</i>	

Komponent	Montage	Printsymbool	Schema-symbool
Stekker 2 pin JP1 			
Sensorschakelaar K1-K6 			
Schakelaar (aan/uit) S1 			
IC (voet) 		<p>de halve cirkel is de markering en komt op dezelfde plaats als bij het printsymbool</p> 	GEEN SYMBOOL
SUB-D connector 			
Motor 			GEEN SYMBOOL (M+ Rechts/Links M-)

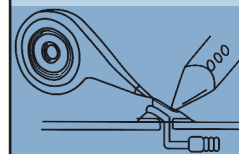
Print reparatie

Printplaten bestaan uit koperbanen, die aansluitpunten van de componenten elektrisch verbinden. Het koper op de aansluitpunten wordt met tin bedekt. Een groene isolerende lak beschermt de koperbaan tegen kortsluiting en roestvorming. Als we bij het solderen het koper te lang verhitten, kunnen het aansluitpunt en de koperbaan van de printplaat loslaten. Ter reparatie moeten we dan een stukje laklaag verwijderen zonder de koperbaan te bekrassen. Hiervoor is een mesje met een gebogen snijvlak (bijv. een scalpel) het meest geschikt.

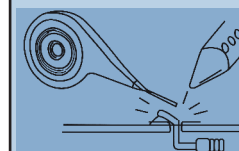
	Als de baan is onderbroken, of het soldeereilandje is weg,
	haal dan de laklaag van de baan met een mesje weg en buig de componentdraad.
	Soldeer een stukje draad van baan tot baan, of van baan tot component..

DESOLDEREN

Als we een fout gemaakt hebben met het solderen, kunne we dat weer repareren met de hulp van desoldeer litze.



Leg de desoldeerlitze over de soldeer-verbinding. Verhit de de litze en de soldeer-verbinding.



De litze zuigt nu het tin op, haal de soldeerbout en de litze pas weg als alle tin door de litze is opgezogen.