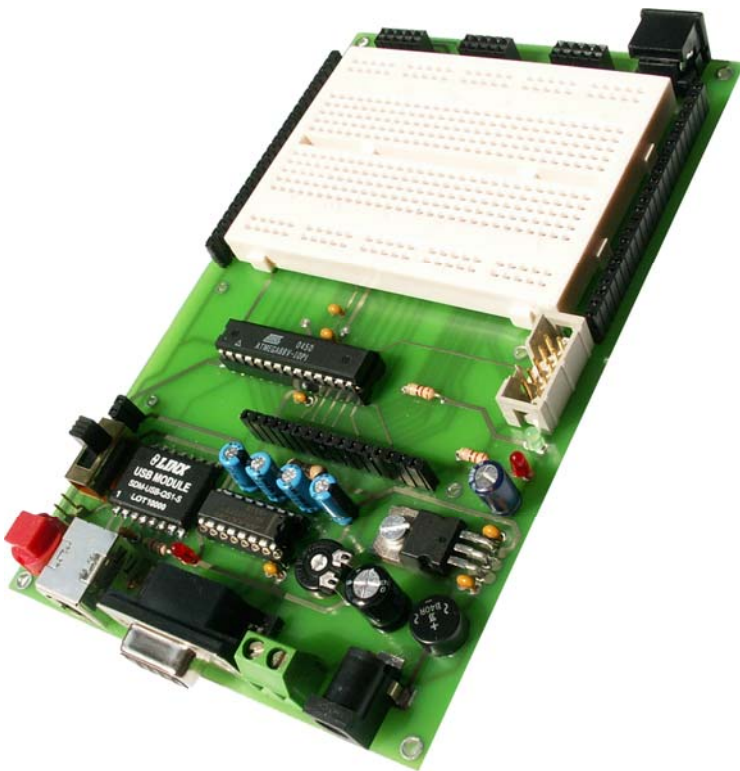


Versie

0.3

MCS ELECTRONICS

Making Things Easy



Educatief Ontwikkel Bord
kenmerken:

- LCD
- PS/2 toetsenbord/muis connector
- Breadbord
- USB interface
- RS-232 interface
- ISP aansluiting
- Wordt geleverd met uitgebreide handleiding

Educatief Ontwikkel Bord Handleiding

UPDATES BESCHIKBAAR OP

www.mcselec.com/edb

MCS ELECTRONICS

Educatief Ontwikkel Bord Handleiding

Barry de Graaff
© 2005 MCS Electronics
www.mcselec.com

Inhoud

Introductie	5
1. Voorkennis	7
1.1 Hardware benodigdheden	7
1.2 Opgelet! Niet-geprogrammeerde chips!	7
2. Beginnen met het EDB	8
2.1 Bestuken van de print	8
2.2 Bascom AVR installeren	13
2.3 De controller programmeren met de seriële poort (Boot loader procedure)	19
3. Experimenten	22
3.1 Basic I/O Experimenten	24
3.1.2 Looplicht LED's	26
3.1.3a Output met een transistor of FET	28
3.1.3b Output met een FET	31
3.1.3c Output met een relais	32
3.1.4 Output met een ULN2803	33
3.1.5 I/O met Opto coupler	35
3.2 UART Data communicatie	38
3.2.1 De UART	38
3.2.2 De Software UART	43
3.3 Display weergave	45
3.3.1 HD44780 LCD	45
3.3.2 7 segment display	47
3.4 Toetsenbord invoer	49
3.4.1 PS2 / AT Toetsenbord	49
3.4.2 Matrix toetsenbord	51
3.4.3 RC5 afstandsbediening	53

Inhoud (*vervolg*)

3.5 Geavanceerde I/O	56
3.5.1 Teller met anti dender ingang	56
3.5.2 Het GetRC commando	58
3.5.3 Sirene met het SOUND commando	59
3.6 AD en DA conversie	60
3.6.1 PWM uitgang	60
3.6.2 AD conversie met een LDR	62
3.6.3 Low cost voltmeter	65
3.7 Andere mogelijkheden van de controller	69
3.7.1 Geheugen	69
3.7.2 Watchdog	71
3.7.3 Een timer gebruiken	72
3.8 Andere interfaces	73
3.8.1 De I ² C bus	73
3.8.2 USB Interface	75
3.8.3 Installatie procedure	77
3.8.3.1 Installatie procedure Virtuele COM-poort Windows	77
3.8.3.2 Installatie procedure USB Device Windows	80
3.8.3.3 Driver de-installatie	83
3.8.4 USB interface, met Virtuele COM-poort	84
3.8.5 USB interface, device standaard PID&VID	86
3.8.6 USB interface, device met eigen PID&VID	87
3.9 Motoren	88
3.9.1 Stappen motor	88
3.9.2 De DC motor	89
4. Andere programmeer methoden	90
4.1 Parallel programmeren (STK200/300)	91
4.2 USB programmeren (Wiazania USP ISP)	92
4.3 De bootloader (opnieuw)programmeren	93
Bijlage 1 STK200/300 ISP dongle	96
Bijlage 2 ASCII Tabel	97
Bijlage 3 7 segment display pin bezetting	100
Bijlage 4 EDB schema's	101
Copyright and Disclaimer	104

Introductie

In de late jaren 80 van de vorige eeuw waren microcontrollers relatief duur en werden alleen toegepast in computers en consumenten elektronica (TV's en Hifi). Door de grote vraag naar eenvoudige en flexibele elektronica werden microcontrollers steeds betaalbaarder. Tegenwoordig zijn microcontrollers voor iedereen met interesse in elektronica ontwikkeling beschikbaar.

Het Educatief Ontwikkel Bord (Educational Development Board EDB) biedt een introductie tot de wereld van elektronica en microcontrollers. Met behulp van het bord en deze handleiding kunt u zich de elementaire zaken in de wereld van elektronica en microcontrollers eigen maken. U maakt kennis met deze elementaire zaken door het uitvoeren van experimenten. Deze experimenten volgen elkaar op in een logische en opbouwende volgorde, zodat u aangemoedigd blijft om verder te gaan met het volgende experiment.

In de experimenten is de nadruk gelegd op het mixen van conventionele componenten, zoals weerstanden en led's, met een microcontroller. Waar nodig zal een simpele uitleg van de componenten en de theorie achter de microcontroller worden gegeven.

Raak niet in paniek als u de microcontroller niet direct begrijpt. In de eerste experimenten zullen we de microcontroller steeds als een "zwarte doos" beschouwen. Later zullen we de theorie verder toelichten en verwijzingen naar het Internet geven.

Bij deze handleiding heeft u ook een CD ontvangen. Op deze EDB-CD vindt u de broncode voor de experimenten, de oefeningen, maar ook Visual Basic voorbeelden, datasheets en application notes. Een overzicht van de CD kunt u vinden in het bestand "Inhoud van de CD.pdf" in de root van de CD.

MCS Electronics wenst u veel succes met het experimenteren en we verwelkomen uw opmerkingen over dit en onze andere producten. Voor meer informatie of als u contact met ons wilt opnemen kunt u onze website bezoeken op www.mcselec.com.

1. Voorkennis

Dit hoofdstuk licht toe welke voorkennis gewenst is als u wilt beginnen met het Educatief Ontwikkel Bord (EDB).

Het Educatief Ontwikkel Bord is bedoeld voor mensen met weinig of geen ervaring met het gebruik van microcontrollers. Het is wel belangrijk dat u wat ervaring heeft met het bestuken en solderen van (Velleman-)kits, zodat u uw eigen Educatief Ontwikkel Bord kunt assembleren. Basiskennis in computergebruik is ook nodig.

1.1 Hardware benodigdheden

Om Bascom AVR te kunnen installeren heeft u een PC nodig met Windows 95, 98, NT, 2000 of XP. In deze handleiding zullen we steeds de seriële poort gebruiken om de chip te programmeren. Als u geen seriële poort heeft kunt u de STK200/300 dongle of de Wiazania USB ISP programmer gebruiken. U kunt in hoofdstuk 4 lezen hoe u de STK200/300 dongle en de Wiazania USB ISP programmer moet gebruiken.



1.2 Opgelet! Niet-geprogrammeerde chips!

De ATmega88 microcontroller aangeleverd door MCS Electronics is al geprogrammeerd om het serieel programmeren mogelijk te maken. Als u (per ongeluk) de chip wist of als u een nieuwe chip heeft gekocht, moet u deze (eenmalig) programmeren met de STK200/300 dongle *of* de Wiazania USB ISP programmer. Deze procedure staat beschreven in hoofdstuk 4.

2. Beginnen met het EDB

Dit hoofdstuk legt uit hoe u kunt beginnen met het EDB.

Het beste kunt u de alinea's in dit hoofdstuk stap voor stap uitvoeren. Op die manier kunt u soepel doorstromen naar de experimenten in hoofdstuk 3 zonder onaangename verrassingen.

2.1 Bestuken van de print

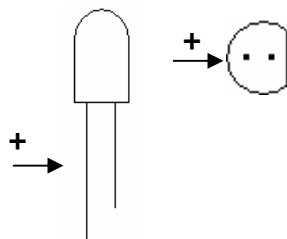
Paragraaf 2.1 legt uit hoe u uw EDB kunt bouwen. Het solderen van de componenten kan het beste in de volgorde zoals in de onderdelenlijst op pagina 8.

Op de print vindt u eilanden zoals deze:

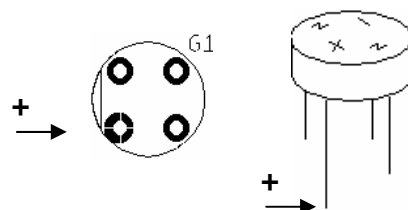


Bij het solderen van IC's geeft het vierkante eiland pin 1 aan. Bij led's en condensatoren geeft het vierkante eiland de (+) positieve pool aan. Hieronder staan wat afbeeldingen van de componenten die verduidelijken hoe de componenten geplaatst moeten worden.

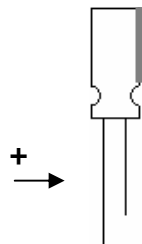
LED



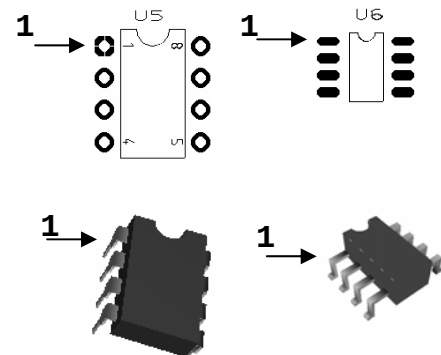
GELIJKRICHTER



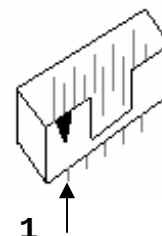
CONDENSATOR



IC



BOXHEADER

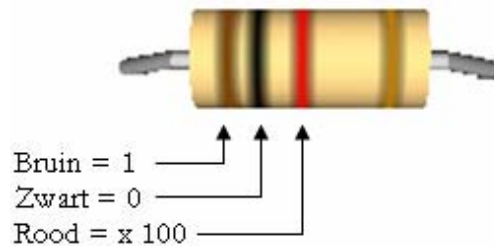


WEERSTANDEN

Weerstanden zijn niet gepolariseerd, het maakt dus niet uit welke pin in welk eiland wordt gesoldeerd. Het is wel belangrijk dat de juiste weerstandswaarde op de juiste plaats op de print wordt gesoldeerd. U kunt de waarde van een weerstand snel achterhalen met een OHM-meter maar u kunt ook de kleurcode gebruiken.

Kleurcode weerstand	Waarde Ring een en twee	Waarde Ring drie (x)
Zwart	0	x1
Bruin	1	x10
Rood	2	x100
Oranje	3	x1000
Geel	4	
Groen	5	
Blauw	6	
Violet	7	
Grijs	8	
Wit	9	

Een voorbeeld hoe u de kleurcode kunt lezen:



Dus 1000 OHM = 1kOHM

De vierde ring is de tolerantie-ring. Goud staat voor een tolerantie van 5%.

ONDERDELENLIJST

Component	Omschrijving	Waarde
R1,R5	WEERSTAND 1/4 W	330E
R4	WEERSTAND 1/4 W	220E
C3,C4,C9,C10, C11,C12,C13	CERAMISCHE CONDENSATOR	100N
R2	WEERSTAND 2W	47E
U1	IC VOET DIP28	for U1
U3	IC VOET DIP16	voor U3
U4	VOLT REGELAAR	7805
R3	POTMETER	10K
G1	GELIJKRICHTER	
D1,D2,D3	LED	3mm
X15	S. UART FEM. HEADER	1x2
X5,X6, X7	FEMALE HEADER	HDR2X5
X3,X4	FEMALE HEADER	HDR1X24
X12	FEMALE HEADER LCD	HDR1X16
X13	USB CONNECTOR	B TYPE FEMALE
X9	ISP CONNECTOR	BOXHDR 10
X1	POWER	POWER CONNECTOR
C5,C6,C7,C8	ELCO RADIAAL	1U
C2	ELCO RADIAAL	100U
C1	ELCO RADIAAL	220U
RESET	RESET SCHAKELAAR	SW_SPST
X16	DB9 CONNECTOR FEMALE	DB9FL
X2	POWER	SCHROEFKLEM
X10	PS2 TOETSENBORD	MINIDIN6
S1	USB/UART SCHAKELAAR	SWITCH
-	BREADBORD	7x7 CM

Soldeer deze componenten nog niet:

1x ATmega88 (U1) + 1x USB MODULE SDM-USB-QS1-S (U2)

ADVIES

Wij adviseren u IC-voeten te gebruiken in plaats van de IC's direct op de print te solderen. Als u de ATmega meerdere malen uit de print wenst te halen verdienen "low quality" IC voeten de voorkeur boven gedraaide IC voeten. (waarin chips minder makkelijk zijn te verwisselen)

De USB module kan alleen direct op de print worden gesoldeerd. **Soldeer deze nog niet. Plaats ook de IC's nog niet.**

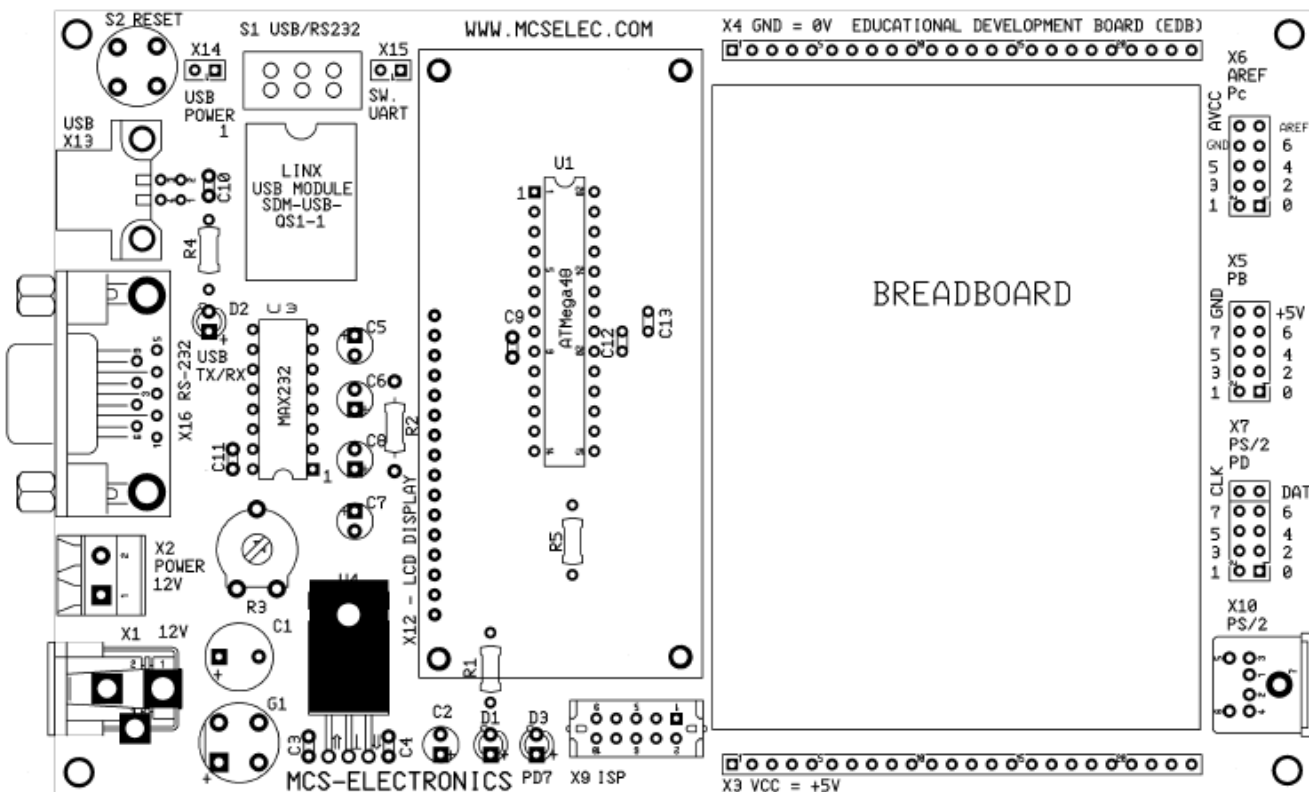
VOEDING OPTIES

Er zijn twee opties om het EDB van stroom te voorzien, u kunt een externe voeding aansluiten op X2 of een adapter op X1. U kunt de EDB ook USB-BUS powered gebruiken. In dat geval moet u een draadbrug solderen in X14 "USB power". LET OP! Als u ervoor kiest USB-BUS-power te gebruiken mag u U4 LM7805 niet plaatsen en mag u geen voeding aansluiten op X1 en/of X2.

Denk eraan niet te veel stroom te trekken van de USB-BUS. (MCS Electronics is niet verantwoordelijk voor schade aan uw computer na USB-BUS-power problemen.)

PRINT LAYOUT

U kunt de juiste plek van de componenten vinden met behulp van dit schema. Het is ook afgedrukt op de print zelf en vergroot beschikbaar in bijlage 4.



ALVORENS DE VOEDING AAN TE BRENGEN

Controleer de print op soldeerdruppels tussen de eilanden nadat u alle componenten heeft gesoldeerd. Verwijder de soldeerdruppels waar nodig.

Meet de weerstand tussen de volgende pinnen:
X2.1 en 2.2 zou ongeveer 18M moeten zijn en
X3 en X4 zou ongeveer 2k6 moeten zijn.

Er is kortsluiting als u een weerstand meet van rond de 0 OHM. Sluit in dat geval de voeding niet aan! Controleer de print nogmaals op soldeerdruppels en controleer of u U4 en G1 juist heeft geplaatst.

Als u een oneindig hoge weerstand meet heeft u waarschijnlijk connector X2, X3 of X4 onvoldoende gesoldeerd. Soldeer deze connectors opnieuw.

De bovengenoemde waarden kunnen afwijken op uw print, het gaat erom dat uw print niet meer dan $\pm 10\%$ afwijkt van de genoemde waarden.

Als de weerstandswaarden juist zijn kunt u een voeding (of USB-power) aansluiten. De aan te sluiten adapter moet een spanning hebben van 9-15 Volt gelijkspanning. De polariteit maakt niet uit omdat er op het EDB een brug-gelijkrichter is aangebracht.

De power LED zou nu moeten branden. Meet de spanning tussen pin 2 en 3 van U4. Het gemeten voltage zou circa 5V moeten zijn.

Als de spanning niet juist is, verwijder de voeding en controleer de print.

Als de spanning juist is, verwijder de voeding en plaats de MAX232 seriële buffer chip. Plaats ook de ATmega88 en soldeer de USB module

Soldeer een enkele male header aan de LCD module... u kunt de LCD aansluiten op connector X12 door de LCD met de male header aan te brengen in X12.

2.2 Bascom AVR installeren

Om de ATMega88 microcontroller op het EDB te programmeren heeft u Bascom AVR nodig. U kunt kiezen voor een gratis demo versie van Bascom AVR. De demo kan 4kB compileren. U kunt ook kiezen voor een volledige versie. Beide versies zijn beschikbaar via www.mcselec.com.

Installatie procedure voor de volledige versie

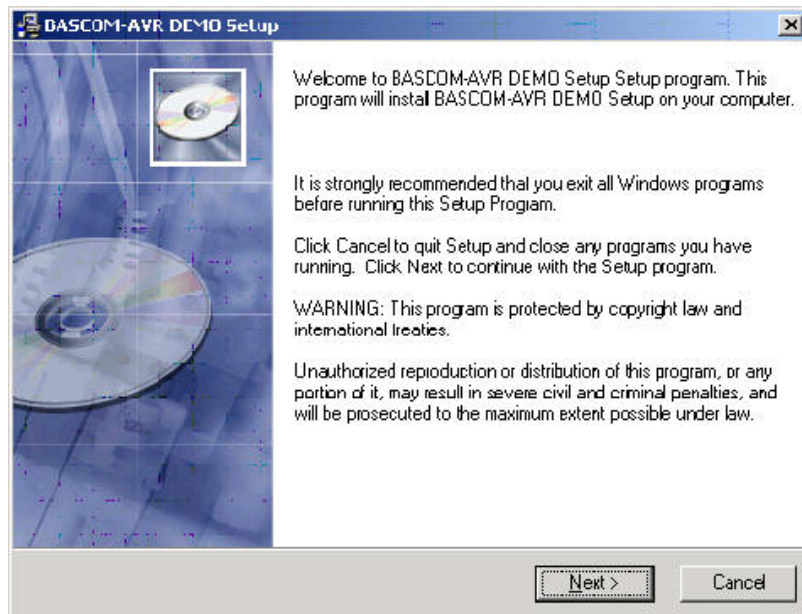
De volledige versie wordt aangeleverd op CD. Deze CD start vanzelf als u deze in de CD-ROM drive doet, indien dit niet het geval is kunt met uw verkenner naar de CD-ROM navigeren en daar dubbelklikken op setup.exe. U kunt nu verder lezen op de volgende pagina.

Installatie procedure voor de demo versie

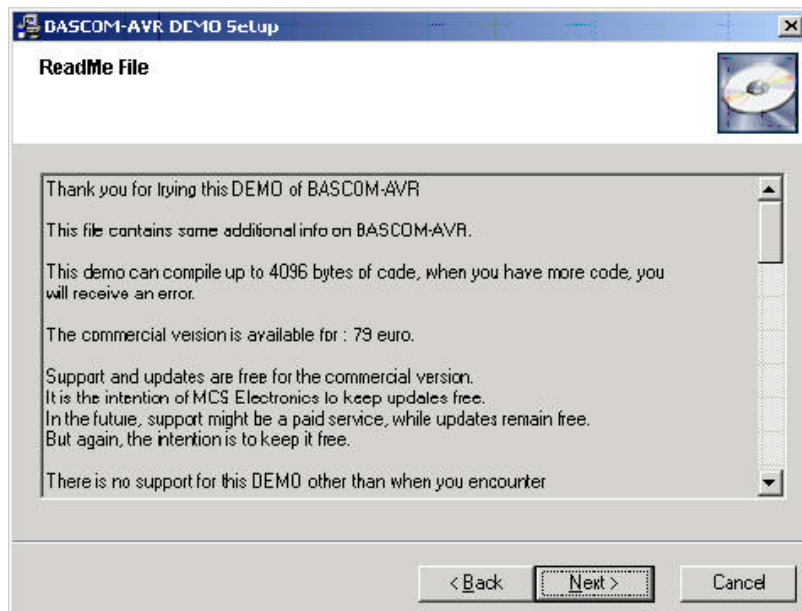
- Download de installatie bestanden (bcavrd.zip) van www.mcselec.com.
- In Windows XP kunt u ZIP bestanden standaard al openen. Zorg ervoor dat u alle bestanden uitpakt in dezelfde map. Als u het ZIP bestand niet kunt openen, dient u eerst WinZip te installeren. (Download op www.winzip.com)
Setup.exe niet vanuit het ZIP bestand uitvoeren!
- Nadat u het ZIP bestand heeft uitgepakt, heeft u een map die het bestand setup.exe bevat.
- Open setup.exe om de installatie te beginnen.

Lees verder op de volgende pagina.

U ziet nu dit scherm:

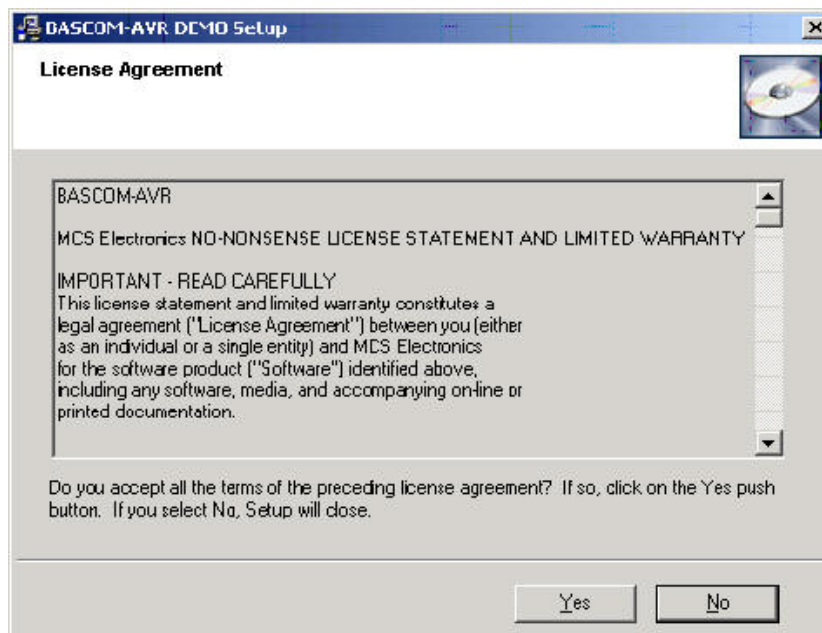


Klik op “Next >”
U ziet nu dit scherm:



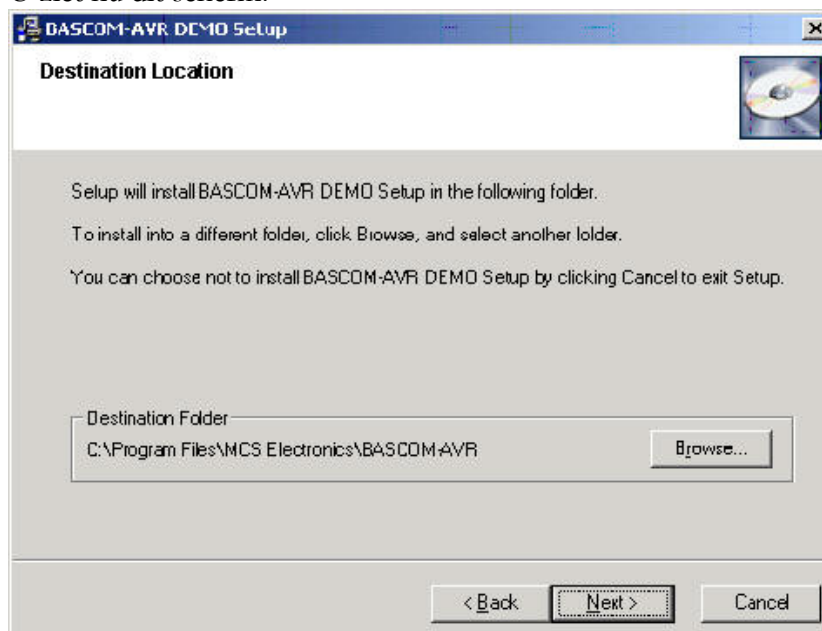
Klik op “Next >”.

U ziet nu dit scherm:



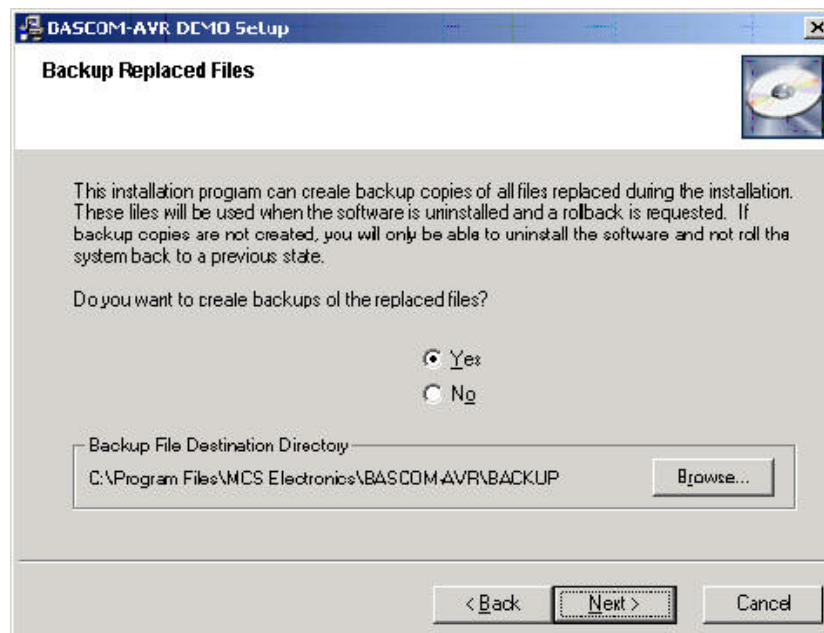
Klik op "Yes"

U ziet nu dit scherm:



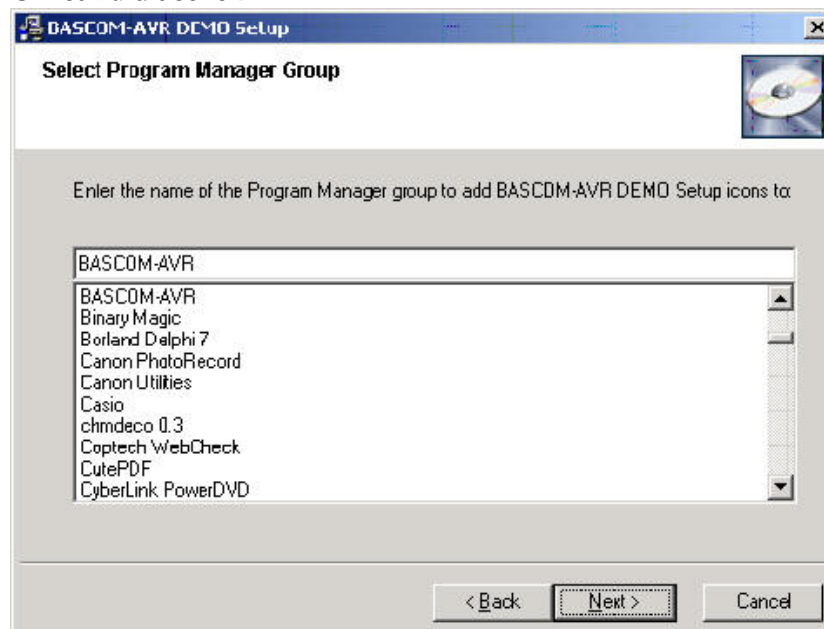
U kunt nu een schijf en map selecteren voor de installatie. U kunt ook meteen op "Next" drukken om de standaard schijf en map te gebruiken. Druk op Next

U ziet nu dit scherm:



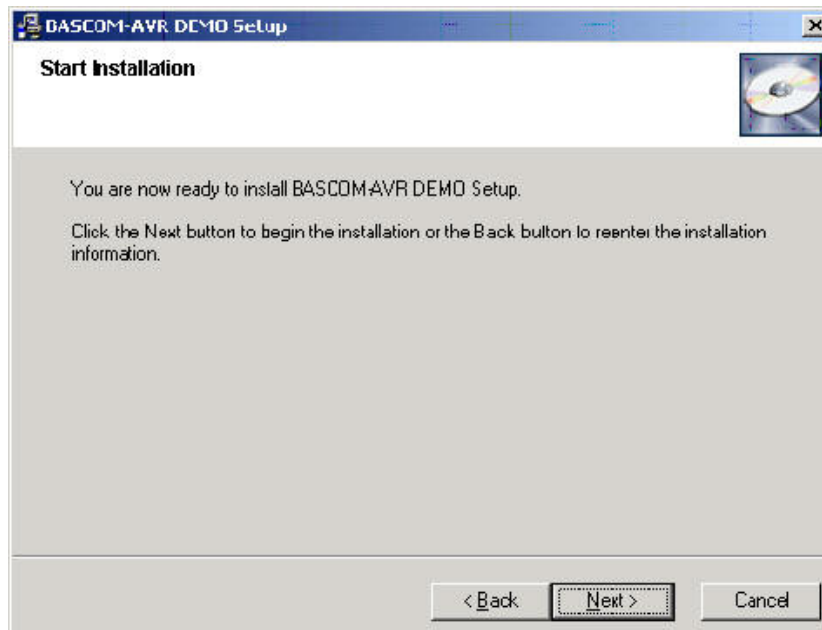
In dit scherm krijgt u de mogelijkheid een backup te maken van bestanden die worden overschreven. Normaal gezien worden er geen bestanden vervangen, toch raden wij aan dat u "Yes" kiest. Klik vervolgens op "Next".

U ziet nu dit scherm:

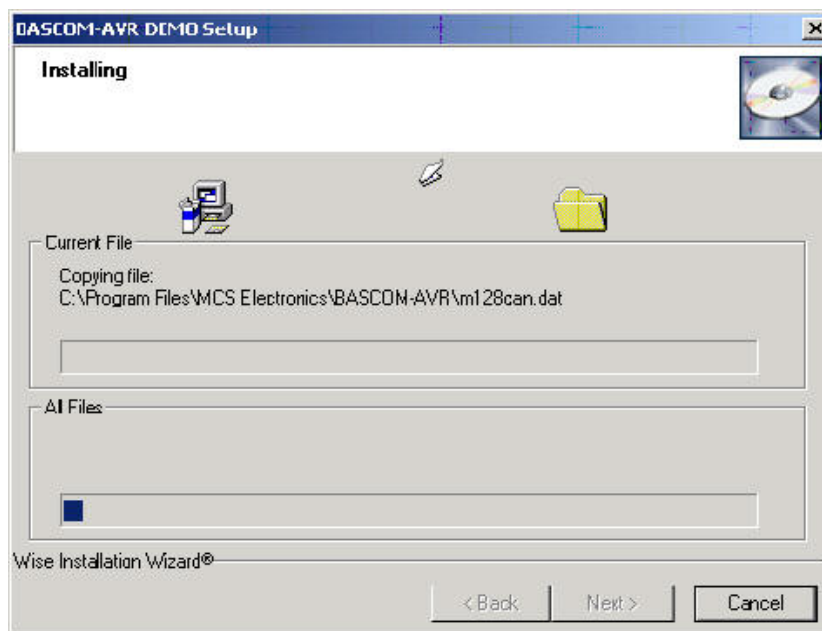


U kunt nu selecteren waar Bascom in uw startmenu komt te staan. Druk op "Next" als u een locatie heeft aangegeven.

U ziet nu dit scherm:



Het setup programma heeft nu alle informatie die nodig is voor installatie. Druk op "Next" om te beginnen.



Tijdens de installatie ziet u het bovenstaande scherm.

BASCOM-AVR is nu gereed voor gebruik.

Om Bascom te installeren onder Windows NT, Windows 2000 of XP, heeft u Administrator rechten nodig. U heeft ook Administrator rechten nodig als u Bascom voor het eerst start. Daarna kunnen alle gebruikers Bascom gebruiken.

2.3 De controller programmeren

met de seriële poort (Boot loader procedure)

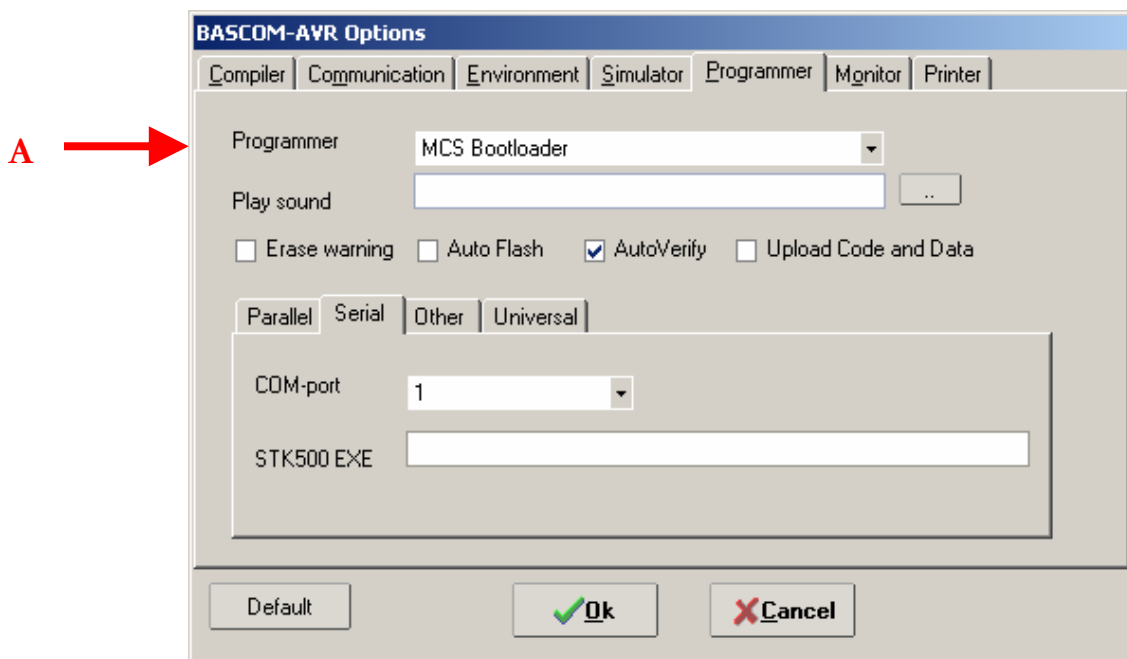
Deze paragraaf legt uit hoe u Bascom kunt instellen om de microcontroller te programmeren via de seriële poort.

Deze methode om de controller te programmeren via de seriële poort wordt “Bootloader” genoemd. Een bootloader is een stuk software om uw programma code in het geheugen van de controller te laden. (De bootloader zelf zit ook in het geheugen van de controller, maar daarvoor heeft MCS Electronics al gezorgd.)

U kunt het Educatief Ontwikkel Bord programmeren met de seriële kabel die is meegeleverd bij de KIT. Maar u kunt elke “straight” kabel gebruiken die pin 2->2, pin 3->3, pin4->4 en pin 5->5 met elkaar verbind.

Sluit de seriële kabel aan op de seriële poort van de computer en sluit het andere eind van de kabel aan op connector X16 (DB-9) er staat “RS232” naast gedrukt.

Klik in Bascom op “Options” en “Programmer” u zou nu dit scherm moeten zien:



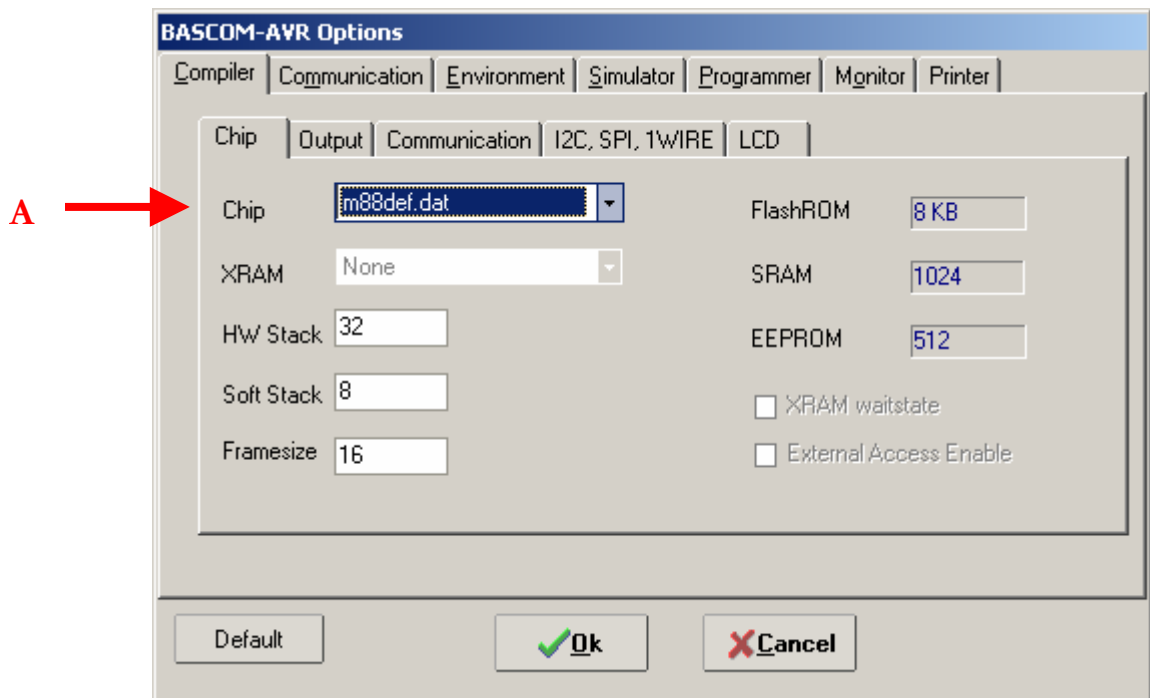
- Selecteer “MCS Bootloader” in de drop down box (A).
- Selecteer de juiste “COM-port” en klik op “OK”.

“De flash programmeren”

Voordat u de chip kunt programmeren, dient u eerst het test programma te openen en te compileren. U kunt het test programma vinden op de EDB-CD. Open het EDBtest.bas bestand in Bascom AVR. (Klik “File” > “Open”)

Open het menu met chip opties (“Options” > “Compiler” > “Chip”)

U zou nu dit scherm moeten zien:



- Selecteer m88def.dat bij het “Chip” drop down menu(A).
- Wijzig niets aan de andere opties en klik “OK”
- Druk op F7 om het test programma te compileren
- Druk op F4 om de chip te programmeren met “auto program”.

Als alles juist is gegaan zou u nu een knipperende led moeten zien op het EDB.

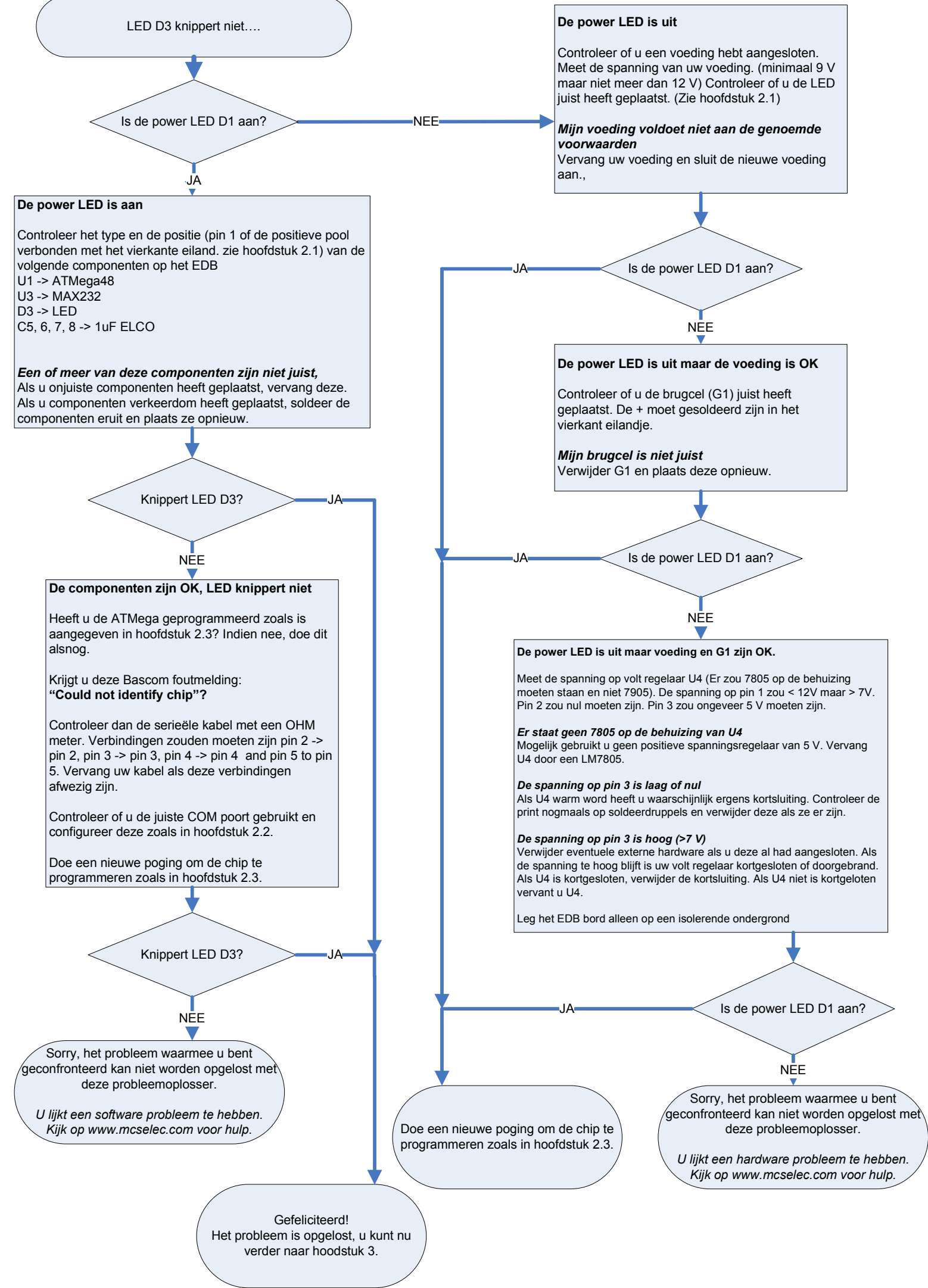
Als u een ATMega88 gebruikt die niet is aangeleverd door MCS Electronics of als u wijzigingen heeft aangebracht in de fuse bits, lees dan eerst hoofdstuk 4.3.

- Ziet u led D3 (PD7) knipperen?

Als het antwoord JA is, gefeliciteerd! Ga verder naar hoofdstuk 3

Als het antwoord NEE is, geen zorgen... ga verder op de volgende bladzijde.

2.4 Probleemoplossing



3. Experimenten

Dit hoofdstuk leert u de basics van elektronica en microcontrollers.

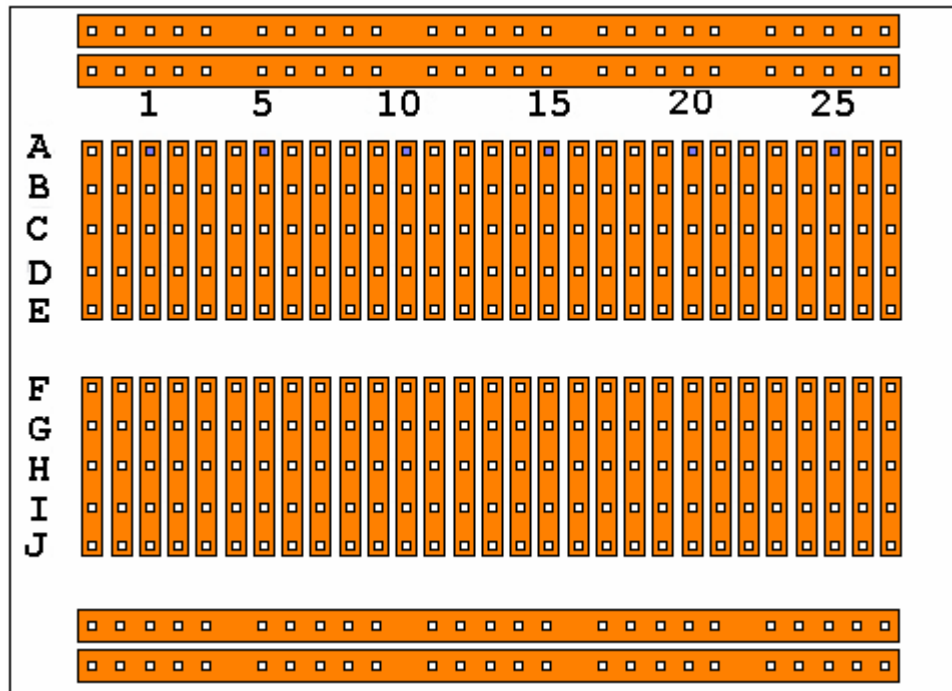
Nu dat uw Educatief Ontwikkel Bord is bestukt en getest, kunt u beginnen met experimenten. De experimenten volgen elkaar op in een logisch opbouwende volgorde, zodat u aangemoedigd blijft om verder te gaan met het volgende experiment.

De experimenten kunnen worden onderverdeeld in deze categorieën,

- Basic I/O
- Seriële communicatie met de UART
- Display weergave (LCD, 7 segment)
- Toetsenbord invoer (PS/2, Matrix, RC5 afstandsbediening)
- Geavanceerde I/O
- A/D conversie
- Andere controller mogelijkheden
- Andere interfaces (I²C, USB)
- Motoren

BREADBORD

Een breadbord wordt gebruikt voor het maken van soldeerloze verbindingen tussen componenten in experimentele fase. U kunt het gebruiken voor het uitvoeren van de experimenten. Als dit de eerste keer is dat u een breadbord gebruikt, kijk dan even naar deze afbeelding die de interne verbindingen van het breadbord laat zien.



De **koper kleurige** gebieden geven de interne verbindingen weer. Componenten met een 2,54 mm raster tussen de pinnen kunt u op het breadbord plaatsen. Gebruik draden met een diameter van 0,3 tot 0,8 mm. Gebruik niet teveel kracht bij het plaatsen van componenten en draden op het breadbord.



**GEVAAR VOOR ERNSTIGE
VERWONDINGEN OF DOOD!**

**Breadborden zijn niet geschikt voor
aansluiting op het lichtnet.**

Sluit het breadbord NOOIT aan op het lichtnet.

3.1 Basic I/O Experimenten

I/O betekend “input en output” wat weer invoer en uitvoer betekend. Ingaande signalen worden aangesloten op input pinnen van een microcontroller terwijl uitvoer wordt aangesloten op output pinnen. (Een bidirectionele pin van een microcontroller kan worden geconfigureerd als input of output.) Er bestaan vele soorten I/O het simpelst is een LED output en een schakelaar input.

Experiment

1

Onderdelen

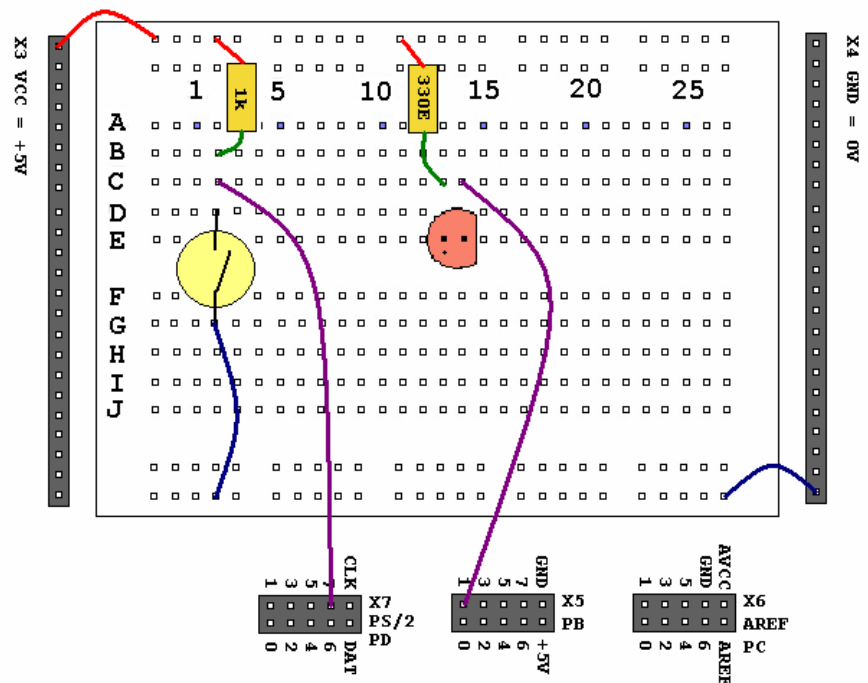
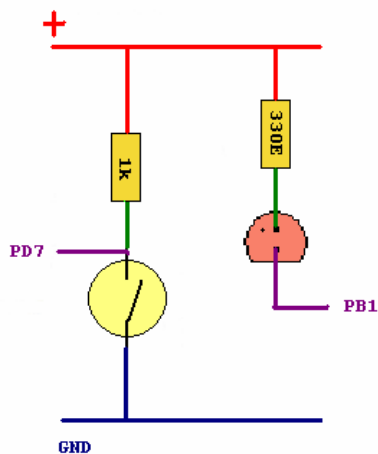
- 1x Weerstand 1k
- 1x Weerstand 330E
- 1x LED
- 1x Switch of een draad
- 6x Breadbord draad

Doel

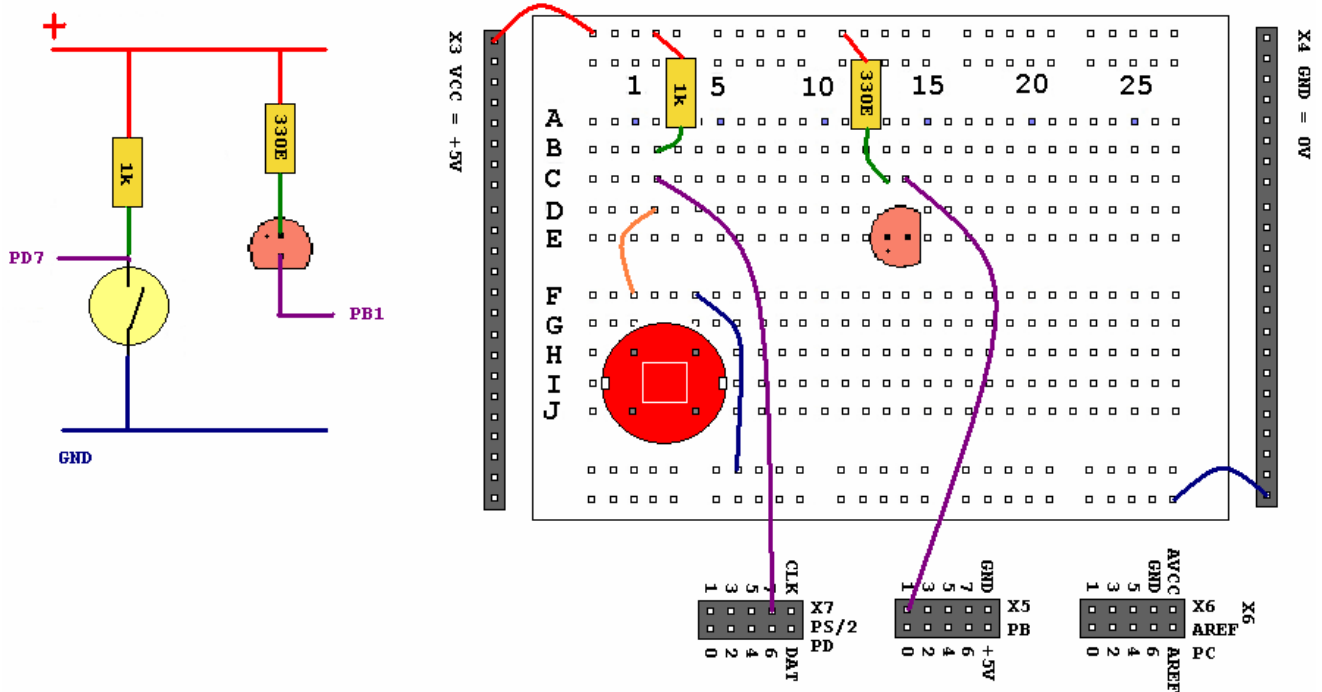
- Basic input en output (I/O) begrijpen

3.1.1 LED output, schakelaar input

Bouw onderstaand schema. Het laat zien hoe u een schakelaar en een LED kunt aansluiten op een microcontroller. Als u geen schakelaar heeft kunt u ook gewoon een draad gebruiken. Naast het schema staat ook een voorbeeld breadbord opstelling.



U kunt ook een door MCS Electronics aangeleverde schakelaar gebruiken, het schema zou er dan zo uitzien:



Open het EDBexperiment1.bas bestand van de EDB-CD.
Het ziet er zo uit:

```
EDBexperiment1.bas
Experiment 1 for the Educational Development Board
(c) 1995-2005, MCS Electronics
Fileversion 1.0
```

```
'Purpose:
'This program assigns the value of Pind.7 to Pinb.1.
'Conclusions:
'You should be able to switch the led with a key press.
```

```
$regfile = "m88def.dat"
```

```
Config Pind.7 = Input
Config Pinb.1 = Output
```

```
Do
  Portb.1 = Pind.7
Loop
```

```
End
```

```
'To tell Bascom which chip we use
```

```
'Configure Pin D7 as input
'Configure Pin B1 as output
```

```
'Do...loop will loop forever
'Assign the output with the input value
```

Programmeer deze code in uw controller net zoals u dat deed met het test programma van hoofdstuk 2.3 (*Onder "De flash programmeren"*). Test de functionaliteit van dit experiment door op de schakelaar te drukken. (De LED zou aan moeten zijn als u drukt.)

3.1.2 Looplicht LED's

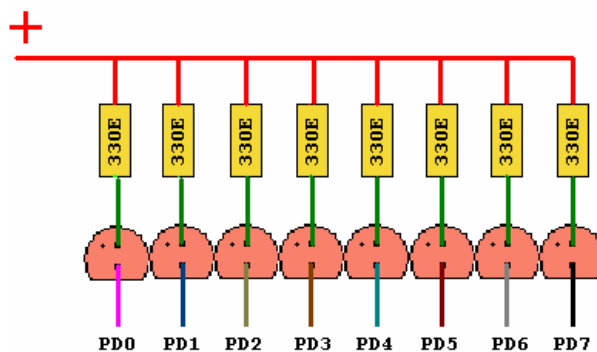
Onderdelen

8x Weerstand 330E
 8x LED
 9x Breadbord draad

Doel

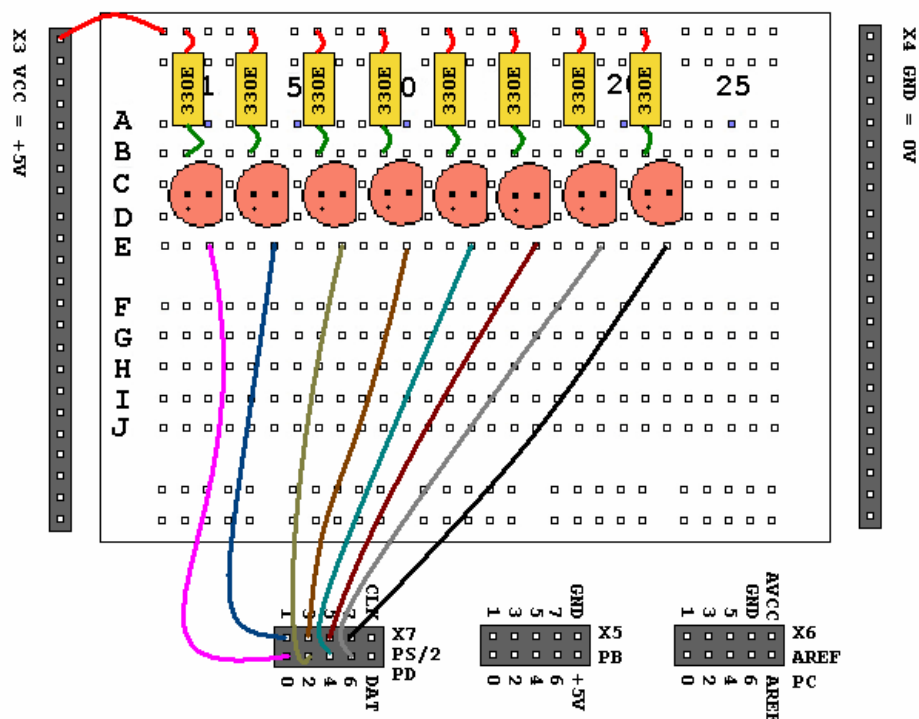
- Het rotate commando begrijpen,
- Basic (I/O) begrijpen

Nog een voorbeeld van basic I/O, een rotate operatie.



U hoeft de ground niet aan te sluiten.
 De ground wordt door de controller geleverd.

Een voorbeeld breadbord opstelling:



Open het EDBexperiment2.bas bestand van de EDB-CD.
Het ziet er zo uit:

```
-----
                        EDBexperiment2.bas
Experiment 2 for the Educational Development Board
                        (c) 1995-2005, MCS Electronics
                        Fileversion 1.0
-----

'Purpose:
'This program 'rotates' the LED's on port D.
'
'Conclusions:
'You should be able to see a moving LED effect.

$regfile = "m88def.dat"                                'To tell Bascom which chip we use
Config Portd = Output                                  'Configure Pin D7 as output
Portd = &B1111110                                       'We start by making pd0 low (led on)

Do
  Rotate Portd , Right , 1                             'Move the 'bits' one place to the right
  Waitms 100                                             'Now PD1 will light and PD0 goes out.
Loop                                                    'Wait otherwise we cannot see te rotate

End
```

Programmeer deze code in uw controller net zoals u dat deed met het test programma van hoofdstuk 2.3 (*Onder "De flash programmeren"*). Als het goed is ziet u nu een led looplicht op het breadbord.

Verander de regel:

```
Rotate Portd , Right , 1
Naar:
Rotate Portd , Right , 2
```

Kunt u het verschil beschrijven?

Zoek zelf uit hoe u de richting van de rotate operatie kunt veranderen van rechts naar links. U kunt help opvragen door de cursor naar het rotate statement te verplaatsen en op F1 te drukken.

Experiment 3a

3.1.3a Output met een transistor of FET

Onderdelen

- 1x BC547 NPN transistor
- 1x LED
- 5x Breadbord draad
- 1x Weerstand 330, 47k ohm

Doel

- Kennismaking met de elektrische eigenschappen van de ATmega
- Basic transistor en FET gebruik

Theorie

Experimenten 1 en 2 gebruiken “directe I/O” d.w.z. dat we de schakelaar en de LED direct verbonden met de controller. Dat is eenvoudig maar er kleven wat nadelen aan. Ten eerste kan de controller niet veel stroom leveren. Het tweede nadeel wordt gevormd door het feit dat directe I/O met zich meebrengt dat de controller kan beschadigen door elektro statische ontlading.

De stroom die wordt geleverd door een controller moet binnen bepaalde grenzen blijven. Deze waarden staan in het datasheet van de ATmega88. (www.atmel.com) De stromen staan onder “**Electrical Characteristics**”. Een voorbeeld van de “Electrical Characteristics” kunt u vinden op de volgende pagina.

Kijk naar de tabel “**Absolute Maximum Ratings**” de waarden in deze tabel zijn het Absolute Maximum, dat betekent dat als deze waarden niet worden overschreden de controller niet zal beschadigen. Het zegt *niets* over de functionaliteit van uw applicatie. (Dus als er 40 mA uit 1 port pin komt, krijgt u van Atmel de garantie dat de controller heel blijft. Atmel geeft *geen* garantie dat de controller functioneert terwijl er 40 mA wordt getrokken)

Welke stromen zijn toegelaten?

Stroom die vloeit van een pin op 0V niveau:

1. De stroom van de pinnen C0 - C5, ADC7, ADC6 niet hoger dan 100 mA.
2. De stroom van de pinnen B0 - B5, D5 - D7, XTAL1, XTAL2 niet hoger dan 100 mA.
3. De stroom van de pinnen D0 - D4, RESET niet hoger dan 100 mA.

Stroom die vloeit van een pin op 5V niveau:

1. De stroom van de pinnen C0 - C5, D0 - D4, ADC7, RESET niet hoger dan 150 mA.
- 2 De stroom van de pinnen B0 - B5, D5 - D7, ADC6, XTAL1, XTAL2 niet hoger dan 150 mA.

En dit met een maximum van 200mA voor de hele controller (Absolute Max.)

Download altijd de laatste versie van de dataheet voor de volledige eigenschappen en ook omdat de waarden hierboven uit een “preliminary datasheet” komen.

27. Electrical Characteristics

27.1 Absolute Maximum Ratings*

Operating Temperature.....	-55°C to +125°C
Storage Temperature	-65°C to +150°C
Voltage on any Pin except $\overline{\text{RESET}}$ with respect to Ground	-0.5V to $V_{CC}+0.5V$
Voltage on $\overline{\text{RESET}}$ with respect to Ground.....	-0.5V to +13.0V
Maximum Operating Voltage	6.0V
DC Current per I/O Pin	40.0 mA
DC Current V_{CC} and GND Pins	200.0 mA

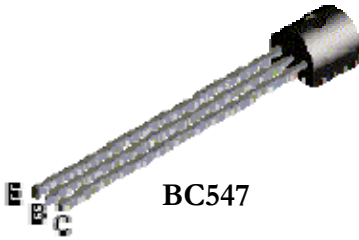
*NOTICE: Stresses beyond those listed under "Absolute Maximum Ratings" may cause permanent damage to the device. This is a stress rating only and functional operation of the device at these or other conditions beyond those indicated in the operational sections of this specification is not implied. Exposure to absolute maximum rating conditions for extended periods may affect device reliability.

27.2 DC Characteristics ATmega48/88/168*

* Note: The following table contains preliminary data for the ATmega88 and ATmega168.

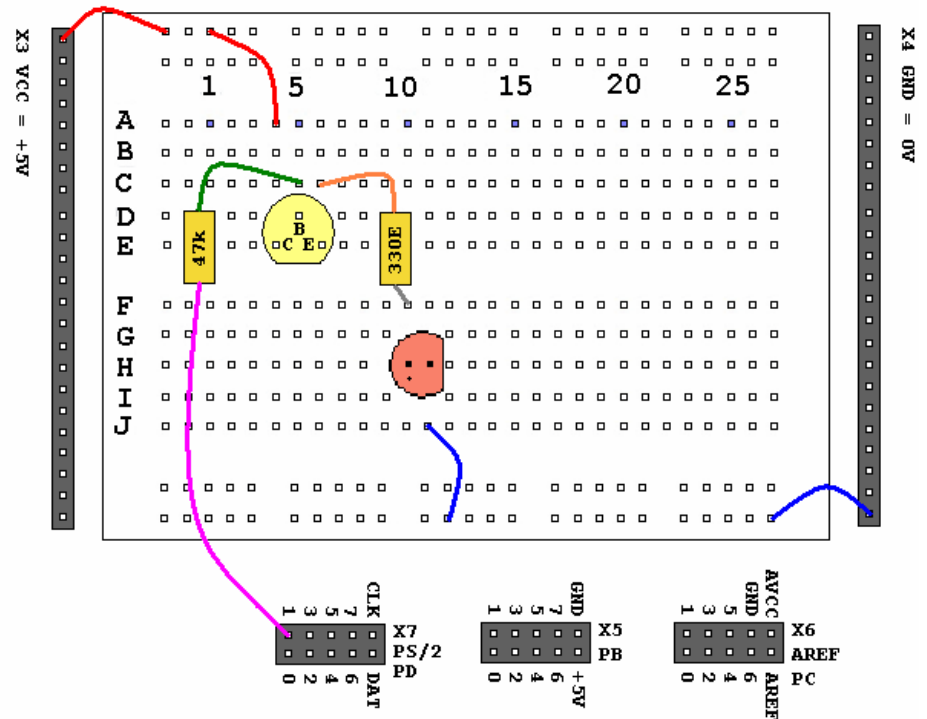
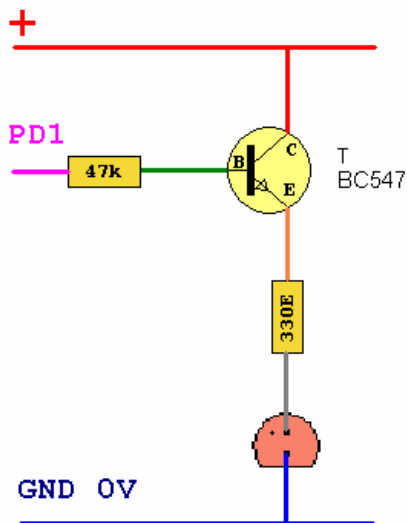
$T_A = -40^\circ\text{C}$ to 85°C , $V_{CC} = 1.8V$ to $5.5V$ (unless otherwise noted)

Symbol	Parameter	Condition	Min. ⁽⁵⁾	Typ.	Max. ⁽⁵⁾	Units
V_{IL}	Input Low Voltage, except XTAL1 and $\overline{\text{RESET}}$ pin	$V_{CC} = 1.8V - 2.4V$ $V_{CC} = 2.4V - 5.5V$	-0.5 -0.5		$0.2V_{CC}^{(1)}$ $0.3V_{CC}^{(1)}$	V
V_{IH}	Input High Voltage, except XTAL1 and $\overline{\text{RESET}}$ pins	$V_{CC} = 1.8V - 2.4V$ $V_{CC} = 2.4V - 5.5V$	$0.7V_{CC}^{(2)}$ $0.6V_{CC}^{(2)}$		$V_{CC} + 0.5$ $V_{CC} + 0.5$	V
V_{IL1}	Input Low Voltage, XTAL1 pin	$V_{CC} = 1.8V - 5.5V$	-0.5		$0.1V_{CC}^{(1)}$	V
V_{IH1}	Input High Voltage, XTAL1 pin	$V_{CC} = 1.8V - 2.4V$ $V_{CC} = 2.4V - 5.5V$	$0.8V_{CC}^{(2)}$ $0.7V_{CC}^{(2)}$		$V_{CC} + 0.5$ $V_{CC} + 0.5$	V
V_{IL2}	Input Low Voltage, $\overline{\text{RESET}}$ pin	$V_{CC} = 1.8V - 5.5V$	-0.5		$0.2V_{CC}^{(1)}$	V
V_{IH2}	Input High Voltage, $\overline{\text{RESET}}$ pin	$V_{CC} = 1.8V - 5.5V$	$0.9V_{CC}^{(2)}$		$V_{CC} + 0.5$	V
V_{IL3}	Input Low Voltage, $\overline{\text{RESET}}$ pin as I/O	$V_{CC} = 1.8V - 2.4V$ $V_{CC} = 2.4V - 5.5V$	-0.5 -0.5		$0.2V_{CC}^{(1)}$ $0.3V_{CC}^{(1)}$	V
V_{IH3}	Input High Voltage, $\overline{\text{RESET}}$ pin as I/O	$V_{CC} = 1.8V - 2.4V$ $V_{CC} = 2.4V - 5.5V$	$0.7V_{CC}^{(2)}$ $0.6V_{CC}^{(2)}$		$V_{CC} + 0.5$ $V_{CC} + 0.5$	V
V_{OL}	Output Low Voltage ⁽³⁾ , $\overline{\text{RESET}}$ pin as I/O	$I_{OL} = 20mA$, $V_{CC} = 5V$ $I_{OL} = 6mA$, $V_{CC} = 3V$			0.7 0.5	V
V_{OH1}	Output High Voltage ⁽⁴⁾ , $\overline{\text{RESET}}$ pin as I/O	$I_{OH} = -20mA$, $V_{CC} = 5V$ $I_{OH} = -10mA$, $V_{CC} = 3V$	4.2 2.3			V
V_{OL3}	Output Low Voltage ⁽³⁾ , $\overline{\text{RESET}}$ pin as I/O	TBD			TBD	V
V_{OH3}	Output High Voltage ⁽⁴⁾ , $\overline{\text{RESET}}$ pin as I/O	TBD	TBD			V



BC547

Als u toch meer stroom wilt kunnen leveren dan met de controller mogelijk is, dan kunt u een transistor als schakelaar gebruiken. Bouw onderstaand schema.



Open het EDBexperiment3.bas bestand van de EDB-CD en programmeer de ATmega88.

Online bronnen:

BC547: <http://www.fairchildsemi.com/pf/BC/BC547.html>

ATmega88: http://www.atmel.com/dyn/products/product_card.asp?part_id=3302

Experiment 3b

3.1.3b Output met een FET

Onderdelen

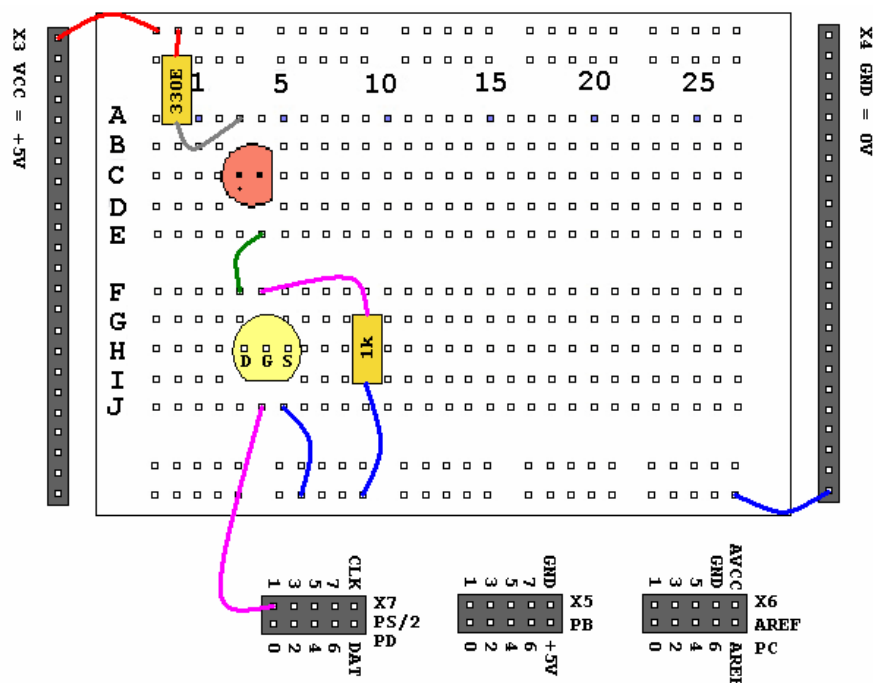
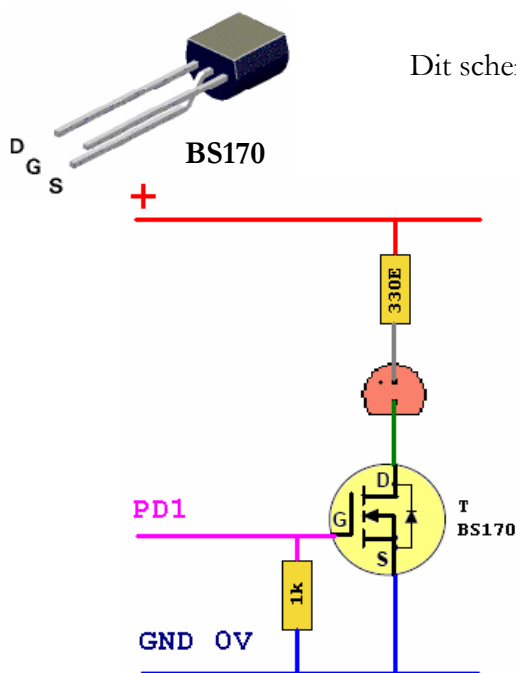
- 1x BS170 N-channel enh. FET
- 1x LED
- 5x Breadbord draad
- 1x Weerstand 330, 47k ohm

Doel

- Gebruiken van een FET

Het gebruik van een MOSFET in plaats van een “klassieke” BC547 transistor heeft wat voordelen. Een MOSFET heeft een hoger rendement en heeft een gate stroom van rond de 0 mA.

Dit schema laat zien hoe u een FET op de controller kunt aansluiten.



Als u de ATMega88 geprogrammeerd heeft in experiment 3a, dan hoeft u dat nu niet te doen. Indien niet, open EDBexperiment3.bas van de EDB-CD en programmeer deze in de ATMega88. Laat de componenten op het breadbord, we gebruiken die ook in het volgende experiment.

Online bronnen:

BS170: <http://www.fairchildsemi.com/pf/BS/BS170.html>

Experiment 3c

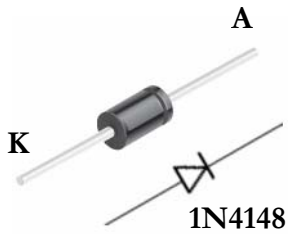
3.1.3c Output met een relais

Onderdelen

- 1x BS170 N-channel enh. FET
- 1x Relais (Schrack PE014-005)
- 7x Breadbord draad
- 1x Weerstand 1k ohm
- 1x Diode 1N4148

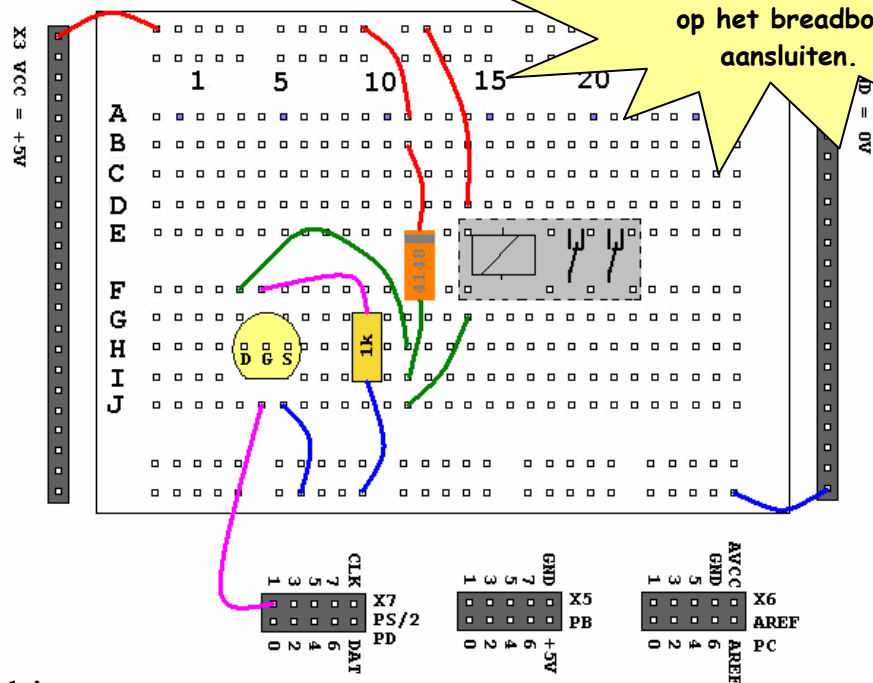
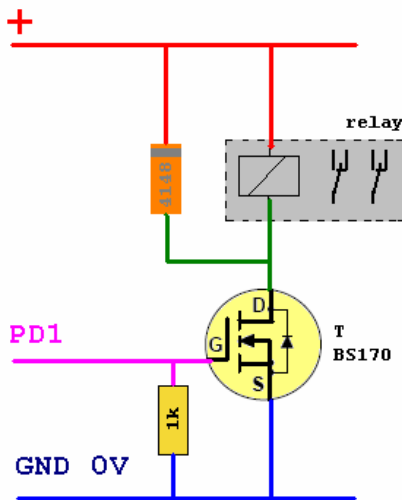
Doel

- Waarom en hoe een relais te gebruiken



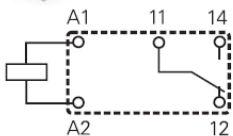
MOSFET's en transistors zijn beide halfgeleiders. De halfgeleiders uit de voorgaande experimenten kunnen alleen gelijkspanning schakelen. U kunt een relais gebruiken als u een potentiaal vrij contact wilt. Dat geeft u de mogelijkheid wisselspanning van het lichtnet te schakelen.

U heeft wel een FET nodig om het relais te schakelen...



Voorbeeld van een relais

Maar let op! Er zijn veel verschillende types.



Als u de ATmega88 geprogrammeerd heeft in experiment 3a/3b, dan hoeft u dat nu niet te doen. Indien niet, open EDBexperiment3.bas van de EDB-CD en programmeer deze in de ATmega88.

Online bronnen:

Relais: <http://relays.tycoelectronics.com/>

3.1.4 Output met een ULN2803

Onderdelen

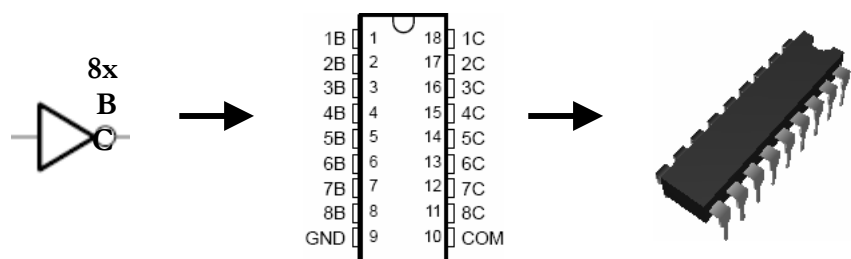
1x ULN2803 of gelijkwaardige
 2x LED
 7x Breadbord draad
 2x Weerstand 330E

Doel

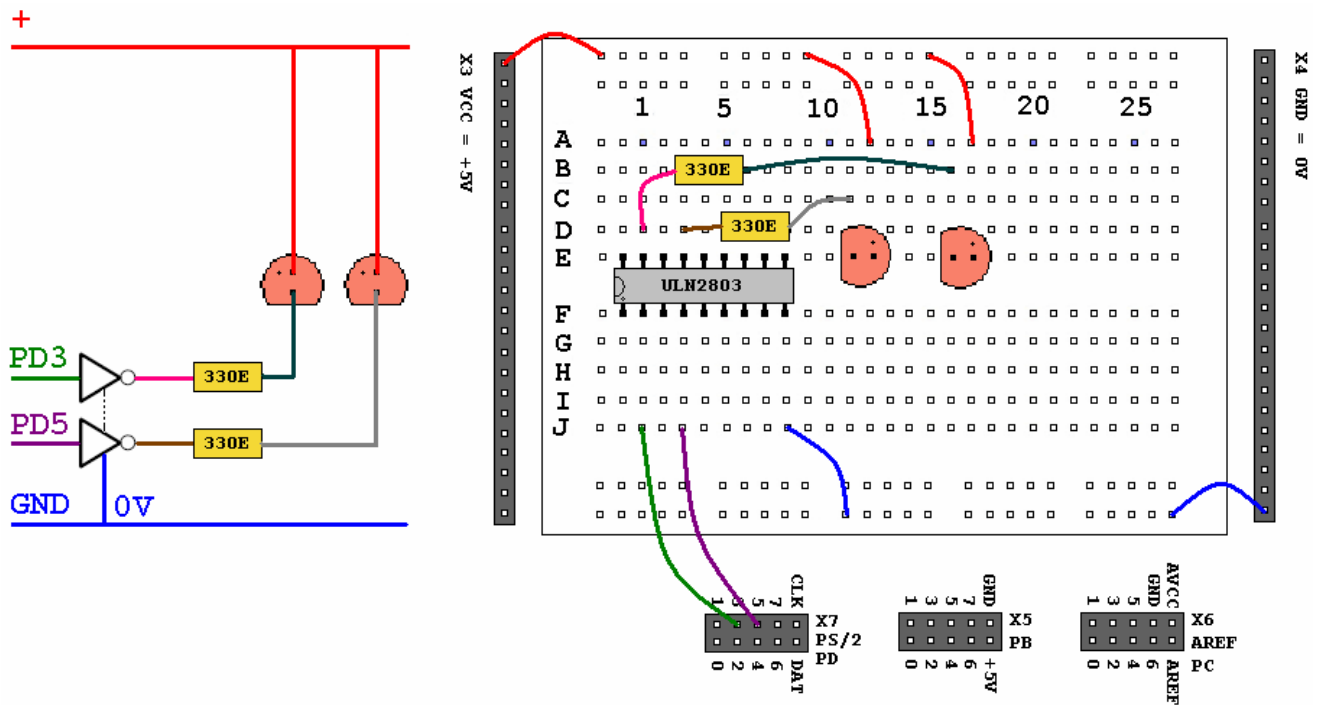
- Kennis maken met de ULN2803

In veel toepassingen, volstaat het niet om maar 1 uitgang te hebben. Stelt u zich eens voor hoeveel componenten u nodig zou hebben met 7 transistor uitgangen! (Zie hoofdstuk 3.1.3a) Dat zou duur en onbetrouwbaar zijn.

U kunt dit voorkomen door gebruik te maken van transistor arrays in DIL behuizing. De ULN2803 heeft 8 ingebouwde “poorten” die stromen kunnen leveren tot 30 mA per pin.



Een voorbeeld breadbord opstelling:



Open het EDBexperiment4.bas bestand van de EDB-CD en programmeer de ATmega88.

In dit experiment hebben we slechts 2 poorten van de ULN2803 gebruikt. Natuurlijk kunt u ook 8 LED's of een aantal relais aansluiten. Let er wel op dat u niet te veel stroom trekt. Kijk in het datasheet voor meer informatie.

Online bonnen:

ULN2803: <http://focus.ti.com/docs/prod/folders/print/uln2803a.htm>

3.1.5 I/O met Opto coupler

Onderdelen

1x PC817 of SFH601 of gelijkwaardig
 1x LED
 7x Breadbord draad
 1x Weerstand 330E, 820E

Doel

- Leren hoe/waarom een Opto-Coupler te gebruiken
- Het bestaan van ESD onderkennen

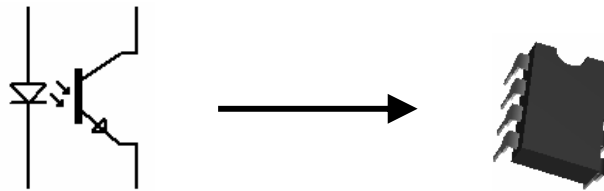
Theorie
ESD

Als u ooit wel eens een vonk heeft gevoeld bij het aanraken van iets of iemand, dan heeft u Electro Statische Ontlading ervaren. (ESD Electro Static Discharge). Halfgeleiders zijn gevoelig voor schade die ontstaat door ESD. ESD kan niet altijd worden gevoeld, of gemerkt.

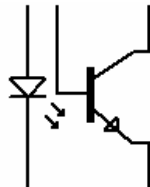
Het ontwerpen van ESD-veilige elektronica is niet eenvoudig en het ligt buiten het doel van deze handleiding er heel diep op in te gaan. Wel zullen we een voorbeeld geven hoe u **galvanische isolatie** kunt creëren met een opto-coupler.

Een opto-coupler is een speciale transistor met een ingebouwde LED. De transistor in een opto-coupler heeft geen basis pin. De basisstroom die nodig is om de transistor in geleiding te sturen wordt verzorgd door het licht dat wordt uitgestraald door de ingebouwde LED als er stroom door vloeit.

Dit plaatje verduidelijkt te situatie:

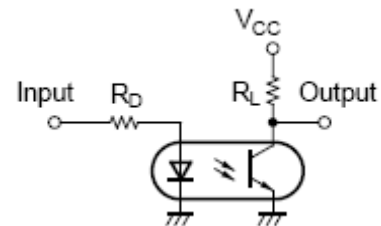
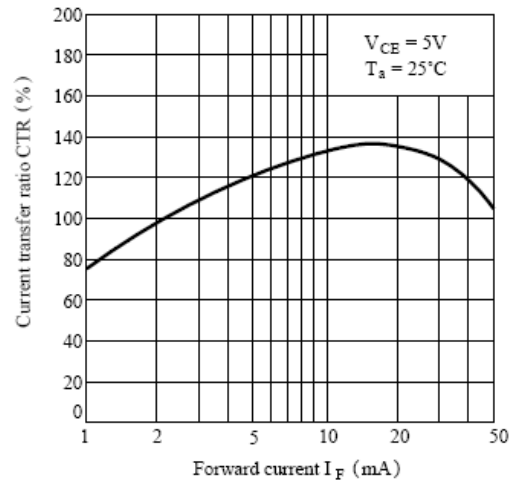


Er zijn ook types die wel een basis pin hebben, in dat geval kunt u de transistor in geleiding brengen met de LED en/of de basis pin, zoals hier:



Voor dit experiment kunt u de PC817 of de SFH601 opto-coupler gebruiken. Laten we er even van uitgaan dan u de PC817 gebruikt. Als u in het datasheet (EDB-CD) kijkt kunt u deze figuur vinden. Figuur 4 geeft weer welke stroom nodig is om de transistor in geleiding te sturen.

Fig. 4 Current Transfer Ratio vs. Forward Current

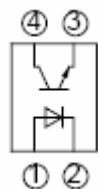
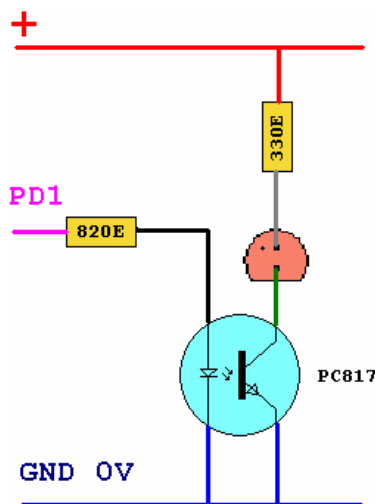


Om een veiligheidsmarge te hebben kunt u het beste ongeveer 5mA aanbrenge. Stel dat de spanning die over de led valt 1,3V is (Forward Voltage) en VCC is 5V. De spanning over R_d zou dan $5 - 1,3 = 3,7V$ moeten zijn. Dit geeft voor R_d
 $R_d = 3,7V / 5mA = 740 \text{ OHM}$.

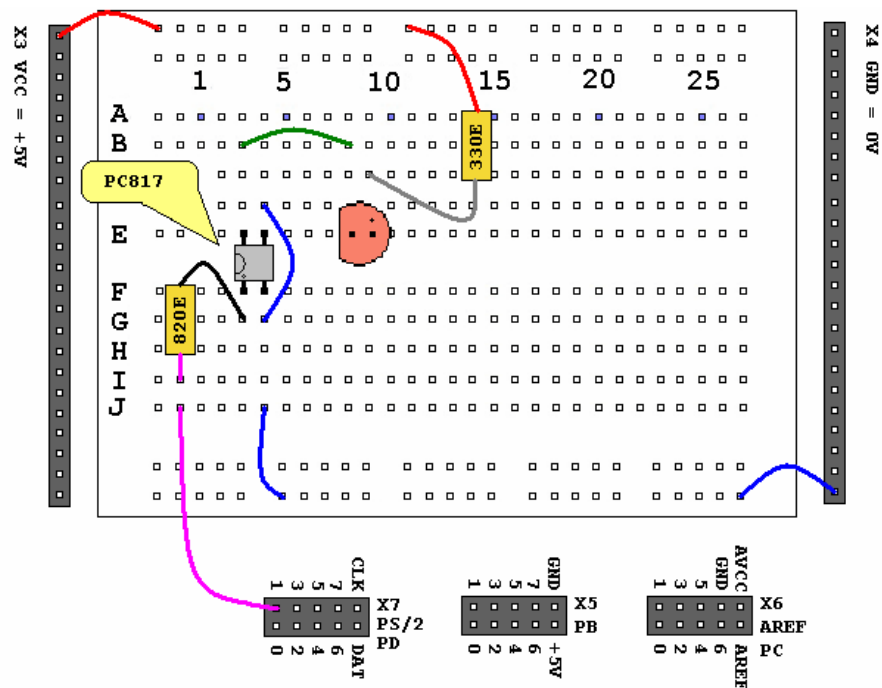
740 OHM is geen waarde in de E24 weerstandsreeks. De dichtstbijzijnde waarde is 820. Dit geeft een stroom van 4,5 mA wat overeenkomt met 120% dus OK.

Online bronnen:

SFH601: <http://www.vishay.com/optocouplers/list/product-83663/>

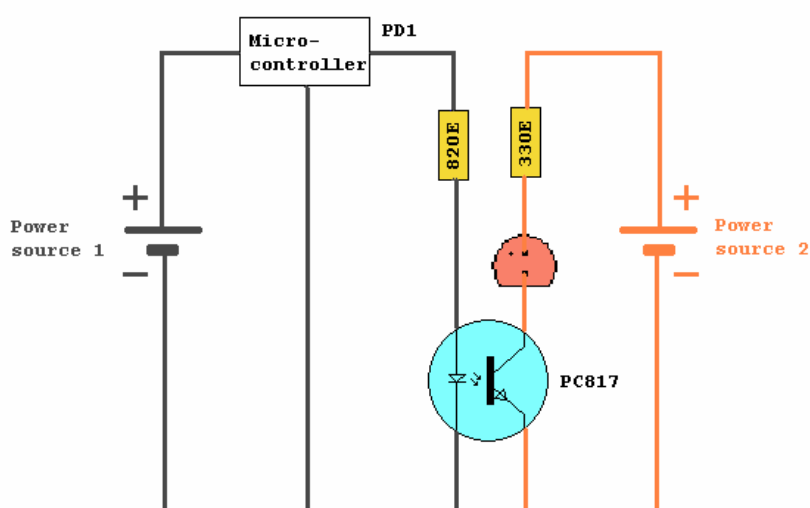


PC817



Open het EDBexperiment3.bas bestand van de EDB-CD en programmeer de ATmega88.

Zoals u kunt zien knippert de LED. ESD kan plaatsvinden door alle *geleiders*. Dus in het bovenstaande schema is PD1 pin wel beveiligd. Maar ESD zou net zo goed over de GND en + kunnen komen en de controller alsnog beschadigen. Om dit te voorkomen heeft u volledige galvanische isolatie nodig:



Hetzelfde principe kan worden gebruikt om ingangen te beveiligen

U hoeft dit schema niet te bouwen, maar kijk of u het principe begrijpt.

3.2 UART Data communicatie

Met de Basic I/O van de vorige experimenten kunnen we veel eenvoudige problemen oplossen. We kunnen een gebruiker een knop later indrukken en de ontvangst bevestigen met een LED. Maar wat als je de gebruiker zijn/haar naam wilt laten intypen en deze naam wilt weergeven op een display.

Dit hoofdstuk gaat over de meest populaire manier van data communicatie in de wereld van microcontrollers.

3.2.1 De UART

Experiment

6a

Onderdelen

1x RS232 kabel
1x Computer met COM poort

Doel

- De seriële verbinding leren gebruiken

Theorie

UART staat voor Universal Asynchronous Receiver en Transmitter. U kunt een UART gebruiken voor het zenden en ontvangen van data tussen uw PC en uw microcontroller (EDB).

Een UART lijkt veel op de COM-poort op de achterkant van uw computer. Het grote verschil is dat de COM-poort RS232/V24 signaal niveaus gebruikt (+15 en -15V) terwijl de UART TTL niveaus gebruikt (5V) of LVTTTL (3V of minder). Om de UART te kunnen laten werken met de PC gebruiken we een *MAX232 level shifter (U3)*. De MAX232 is een IC die de spanningen genereert om te kunnen communiceren met de COM-poort.

Verbind voor dit experiment een 1 op 1 seriële kabel tussen X16 (de DB-9 connector “RS-232”) op het EDB en de COM-poort van de PC.

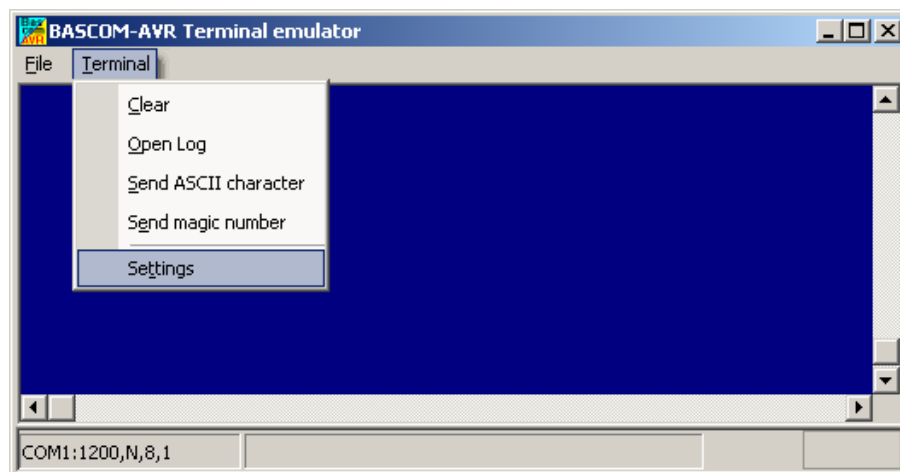
De kabel zou tenminste deze verbindingen moeten hebben:

pin2 naar pin2,
pin3 naar pin3
pin4 naar pin4 en
pin5 naar pin5.

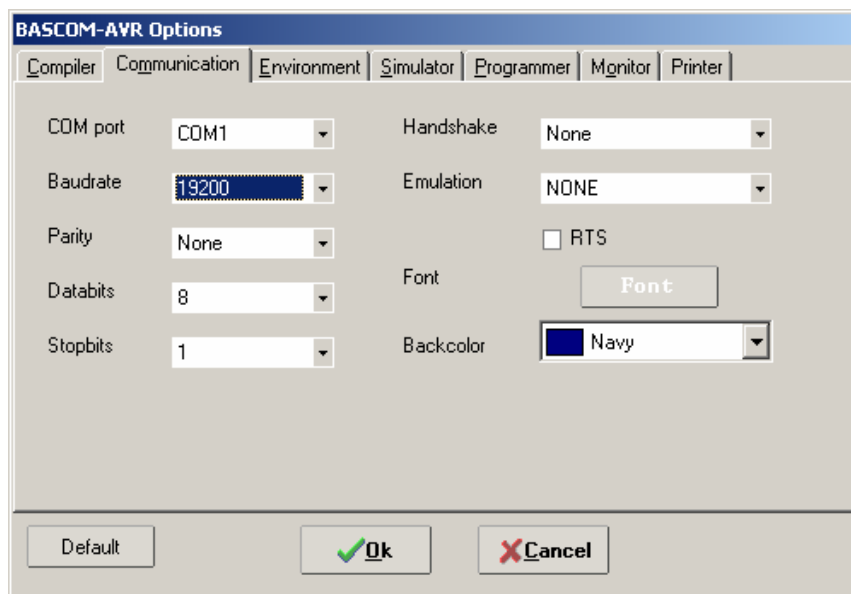
Let erop dat schakelaar “S1 USB/RS232” op het EDB op RS232 staat.

Nadat u de kabel heeft aangesloten, kunt u Bascom AVR starten en de ingebouwde Terminal emulator openen door op  te klikken.

U zou dit moeten zien:



Klik op “Terminal” en “Settings”



Selecteer de COM poort waarop u de kabel heeft aangesloten en kies 19200 als baudrate.

Indien uw PC geen COM poort heeft kunt u een USB to Serial adaptor kopen en die als virtuele COM-poort gebruiken.

Sluit het terminal emulator scherm.

Zoals gezegd gebruiken we een UART om data te zenden en ontvangen tussen de microcontroller op het EDB en de PC. Laten we de data richting als volgt vaststellen:

Gezien vanaf het EDB:

Zenden -> data van het EDB naar de PC en

Ontvangen -> data van de PC naar het EDB.

Eigenlijk ziet de data er hetzelfde uit, maar het is makkelijker praten als we de richting vastleggen. De data zelf kan variabelen, register waarden, constanten (Tekst of getallen) of het resultaat van berekeningen bevatten. U kunt de controller ook een boodschap laten weergeven om aan te geven in welke subroutine het programma zich bevindt in debug-fase.

Data kan ook worden verzonden vanaf een toetsendbord, dit geeft de gebruiker de mogelijkheid boodschappen in te voeren of een menu te gebruiken. We komen hier later op terug.

Experiment 6a

Open het EDBexperiment6a.bas bestand van de EDB-CD en programmeer de ATmega88. Lees de opmerkingen die u kunt vinden in het EDBexperiment6a.bas bestand. Op die manier kunt u begrijpen hoe u de UART kunt initialiseren en gebruiken.

Nadat u de controller heeft geprogrammeerd en de seriële kabel heeft verbonden kunt u de terminal emulator openen door op  te klikken in Bascom.

U zou het EDBexperiment6a.bas programma “Hello World” moeten zien weergeven op het scherm van de PC, zoals hier:

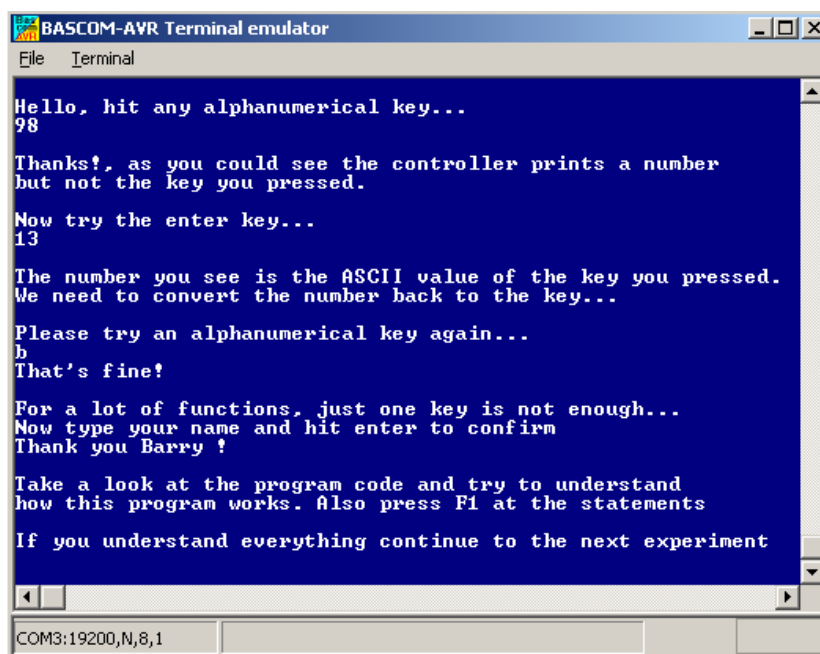


Experiment 6b

Zorg ervoor dat schakelaar “S1 USB/RS232” op het EDB op RS232 staat.

Open het EDBexperiment6b.bas bestand van de EDB-CD en programmeer de ATmega88. Nadat u de controller heeft geprogrammeerd en de seriële kabel heeft verbonden, kunt u de terminal emulator openen door te klikken op  in Bascom.

Volg de instructies die worden weergegeven op scherm door het EDBexperiment6b.bas programma:



```
BASCOM-AVR Terminal emulator
File Terminal

Hello, hit any alphanumerical key...
98

Thanks!, as you could see the controller prints a number
but not the key you pressed.

Now try the enter key...
13

The number you see is the ASCII value of the key you pressed.
We need to convert the number back to the key...

Please try an alphanumerical key again...
h
That's fine!

For a lot of functions, just one key is not enough...
Now type your name and hit enter to confirm
Thank you Barry !

Take a look at the program code and try to understand
how this program works. Also press F1 at the statements

If you understand everything continue to the next experiment

COM3:19200,N,8,1
```

Merk op dat u ook een RS232 verbinding kunt gebruiken om PC's met PC's te verbinden of microcontrollers met microcontrollers.

Op de volgende pagina kunt u nog wat ASCII theorie lezen.

Online bronnen:

UART Hardware

MAX232: http://www.maxim-ic.com/quick_view2.cfm/qv_pk/1798

ASCII

Zoals u kon zien in EDBexperiment6b.bas kunnen we het PRINT commando gebruiken om tekst te verzenden naar de UART. Eigenlijk sturen we niet zomaar tekst, we sturen ASCII tekens/karakters. ASCII staat voor American Standard Code for Information Interchange. ASCII is een lijst van 127 karakters met een bijbehorende binaire waarde.

ASCII Tabel (niet compleet)

Decimaal	Hex	Binair	Karakter
000	000	00000000	NUL (Null char.)
008	008	00001000	BS (Backspace)
009	009	00001001	HT (Horizontal Tab)
010	00A	00001010	LF (Line Feed)
012	00C	00001100	FF (Form Feed)
013	00D	00001101	CR (Carriage Return)
048	030	00110000	0
049	031	00110001	1
052	034	00110100	4
065	041	01000001	A
066	042	01000010	B
067	043	01000011	C

U kunt een complete ASCII tabel vinden in bijlage 2.

CARRIAGE RETURN (CR) EN LINE FEED (LF)

Zoals u kon zien in EDBexperiment6b.bas gebruiken we PRINT om data naar de UART te zenden. U kunt ook zien dat een tweede PRINT commando altijd tekst weergeeft op de volgende regel. Dit wordt veroorzaakt door het feit dat het print commando altijd CR en LF toevoegd aan de te zenden tekst.

Als we dus dit programmeren:

```
Print "ABC"
```

Zenden we 65 66 67 13 10 naar de UART. (In binair formaat)

Het carriage return karakter (13) verplaats de cursor naar positie 0 van de huidige regel. De line feed (10) verplaatst de cursor naar de volgende regel.

```
Print "ABC" ;
```

Als u een puntkomma (;) toevoegd stuurt Bascom geen carriage return/line feed, op die manier kunt u meerdere teksten op een regel printen.

```
Print "ABC" ; Chr(13) ;
```

Stuurt alleen ABC CR. Het volgende print commando overschrijft ABC.

Experiment 6c

3.2.2 De Software UART

Onderdelen

- 1x RS232 kabel
- 1x Computer met COM poort

Doel

- Een software UART leren gebruiken

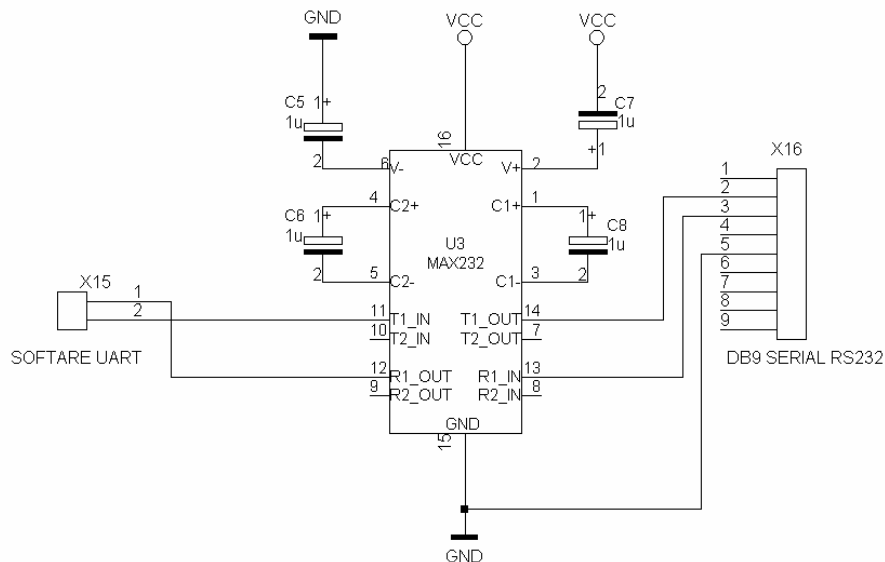
De vorige experimenten maakte gebruik van de hardware UART. Dat betekent dat de compiler de interne UART registers en interne hardware (portd.0 en portd.1) van de ATmega88 gebruikt. Soms willen we meer dan 1 UART gebruiken. Of we willen zowel USB als RS232 gebruiken. (USB wordt uitgelegd in hoofdstuk 3.8.2 en verder.)

Met de Bascom compiler is het maken van extra UART's eenvoudig en mogelijk voor alle poortpinnen die als I/O kunnen worden ingesteld.

Zorg ervoor dat schakelaar "S1 USB/RS232" op het EDB op USB staat.

We gebruiken geen USB interface, maar met de schakelaar op USB zijn de hardware UART pinnen portd.0 en portd.1 losgekoppeld van de MAX232 level shifter.

Met S1 op USB ziet de verbinding met de level shifter er zo uit:



Voordat we kunnen communiceren, moeten we I/O pinnen aansluiten op X15

Verbind de X15 pin die het dichtst bij S1 is met portc.1.

Verbind de andere pin van X15 met portc.2.

Opmerking: als u de bootloader gebruikt om het EDB te programmeren, moet u S1 (USB/RS232) op RS232 zetten om de controller te programmeren. U moet de schakelaar weer op USB zetten om de software UART te testen. (Druk op de reset knop na het overschakelen op USB, op die manier mist u zeker geen boodschappen in de terminal emulator)

Open het bestand EDBexperiment6c.bas van de EDB-CD en programmeer de ATMega88. Als u de opmerkingen leest in het EDBexperiment6c.bas bestand, dan kunt u begrijpen hoe u een software UART kunt initialiseren en gebruiken.

Nadat u de controller heeft geprogrammeerd en de seriële kabel heeft verbonden, kunt u de terminal emulator openen door te klikken op  in Bascom.

Als het goed is ziet u het EDBexperiment6c.bas programma om een alfanumerieke invoer vragen, waarna diezelfde input weer wordt weergegeven.

3.3 Display weergave

Het is niet altijd gewenst om het beeldscherm van een PC te gebruiken. Stel u voor dat u slechts een karakter of een of een paar regels wilt weergeven. Voor dit doel is er veel betere oplossing, een LCD display.

3.3.1 HD44780 LCD

Experiment

7

Onderdelen

- 1x HD44780 LCD
- 1x Computer met COM poort

Doel

- Gebruik van een LCD display

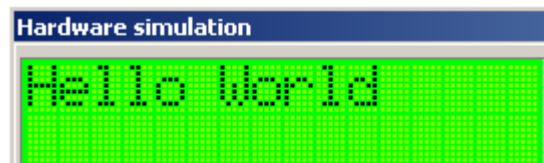
Zorg ervoor dat u het LCD heeft geplaatst op connector X12 op het EDB.

Open het bestand EDBexperiment7.bas van de EDB-CD en programmeer de ATmega88. U kunt vinden hoe u een LCD kunt initialiseren en gebruiken in de opmerkingen van EDBexperiment7.bas.

Als u de LCD heeft geplaatst en de controller is geprogrammeerd, dan kunt u de inmiddels bekende “Hello World” boodschap zien op het LCD display.

Kijk ook eens naar Bascom’s “real hardware simulator” met dit programma. Zorg ervoor dat EDBexperiment7.bas u nog steeds geopend is. Klik op  in Bascom.

Klik dan  in het “AVR simulator” scherm. Klik vervolgens op de “play” en “step-into-code” knoppen. (een paar keer) Als het goed is ziet u dan:



Online bronnen:

Hitachi is tegenwoordig Renesas

LCD fabrikant: <http://www.renesas.com/>

LCD achtergronden: <http://home.iae.nl/users/pouwaha/lcd/lcd.shtml>

Oefening

1

Oefening

De beste manier om controllers te leren kennen is – door het zelf te doen. Maak een programma dat vraagt om “Uw naam” via de UART maar dat het vervolgens weergeeft op het LCD in plaats van op het scherm van de PC. (De oplossing kan worden gevonden op de EDB-CD Exercise1.bas)

3.3.2 7 segment display

Onderdelen

1x 7 Segment display
(Common Anode SL119-OS516HWA)
Conrad 146536 of gelijkwaardig
8x Breadbord draad

Doel

- Een 7-segment display
leren gebruiken

Er zijn 2 nadelen aan het gebruik van een LCD, in de eerste plaats zijn ze relatief duur en ten tweede zijn ze niet zo robuust als LED's. Als u slechts 1 karakter wilt weergeven, kan een 7 segment display uitkomst bieden.

Een 7 segment display bestaat uit 7 LED's. Zodanig in een plastic behuizing gegoten dat er karakters mee gevormd kunnen worden als je de LED's aan of uit schakeld. Er zijn twee soorten gemeenschappelijk anode en gemeenschappelijk kathode. (Common Anode en Common Kathode)

Displays met een gemeenschappelijke anode hebben de POSITIEVE pool gemeenschappelijk, Dat betekend dat u VCC moet aanbrengen op de common pin. En dat uw controller een 0V/GND signaal moet aanbrengen om een LED aan te zetten.

Displays met een gemeenschappelijke kathode hebben de NEGATIEVE pool gemeenschappelijk, Dat betekend dat u 0V/GND moet aanbrengen op de common pin. En dat uw controller een +5V signaal moet aanbrengen om een LED aan te zetten.

Common anode is het meest gebruikt omdat controllers meer stroom kunnen leveren in de richting van de ground dan in de richting van VCC.

Online bronnen:

Application note voor meerdere 7 segment displays:

http://www.maxim-ic.com/appnotes.cfm/appnote_number/3210

Leest u alstublieft de opmerkingen in EDBexperiment8.bas file.
Controleer of u de Data en Read commando's begrijpt.

3.4 Toetsenbord invoer

Het zou leuk zijn als we een PS2/AT toetsenbord direct op onze microcontroller konden aansluiten in plaats van dat via de PC te doen. (dus zonder UART terminal interface.) In dit hoofdstuk kunt u zien dat het aansluiten van een toetsenbord op uw controller eenvoudig is met Bascom in combinatie met het EDB.

Experiment

9

3.4.1 PS2 / AT Toetsenbord

Onderdelen

- 1x HD44780 LCD
- 1x PS2 of AT keyboard
- 1x Optioneel een PC met COM poort

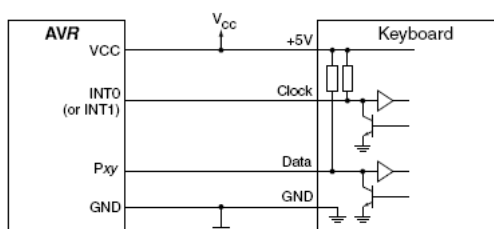
Doel

- Een PS2 of AT Toetsenboard aansluiten op de controller

Het EDB is al voorzien met een standaard PS2 interface. Dat betekent dat u direct een standaard PS2 toetsenbord kunt aansluiten op X10 (PS2) op het EDB. U kunt ook een ouder AT toetsenbord gebruiken, maar als u dat wilt, heeft u wel een DIN42524 naar mini DIN adapter nodig.

De tekening hieronder laat de toetsenbord interface zien. Ook staat er een tabel met de pin bezetting van de toetsenbord connector.

Toetsenbord interface



Pin bezetting

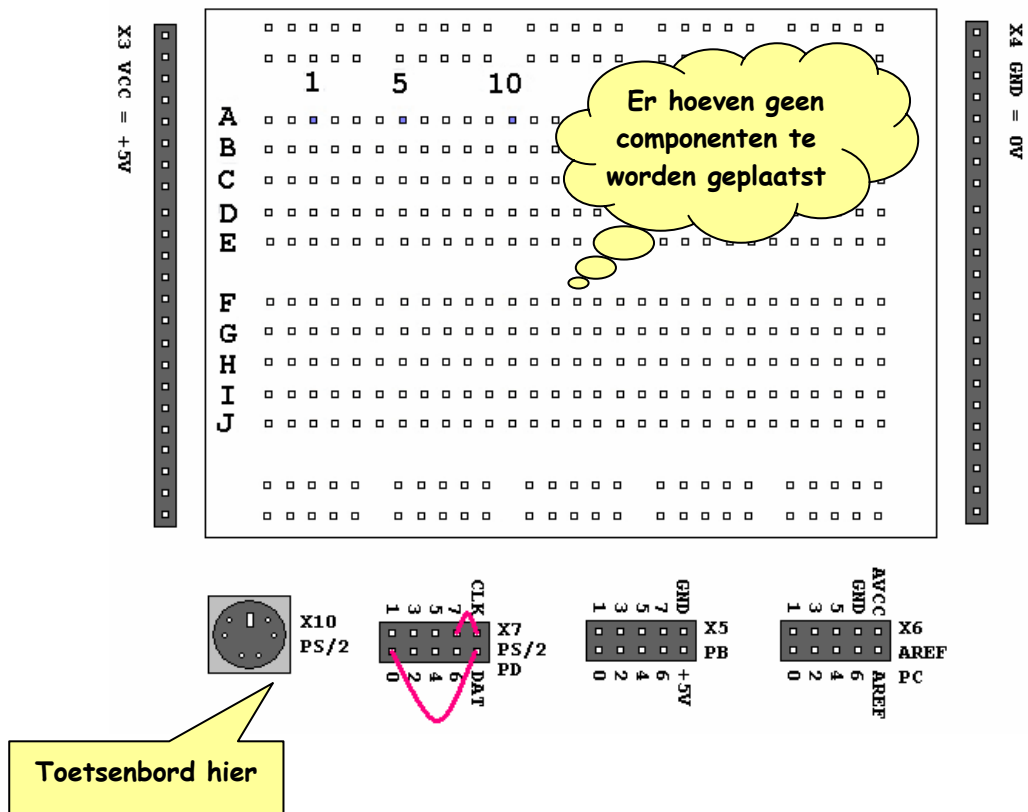
AT Computer		
Signals	DIN41524, Female at Computer, 5-pin DIN 180°	6-pin Mini DIN PS2 Style Female at Computer
Clock	1	5
Data	2	1
nc	3	2,6
GND	4	3
+5V	5	4
Shield	Shell	Shell

Bron: Atmel Application Note "AVR313: Interfacing the PC AT Keyboard"

Online bronnen:

Atmel Application notes:

http://www.atmel.com/dyn/products/app_notes.asp?family_id=607



Sluit uw toetsenbord aan en verbind CLK met PD.7 en DAT met PD.0 (de paarse draden in de figuur). Open vervolgens EDBexperiment9.bas en programmeer de controller.

Als u nu wat op het toetsenbord intypt gevolgd door ENTER, zal de ingetypte tekst op de LCD verschijnen.

Experiment 10

3.4.2 Matrix toetsenbord

Onderdelen

- 1x Matrix toetsenbord
Conrad 709840
- 1x Computer met COM poort
- 8x Weerstand 470E

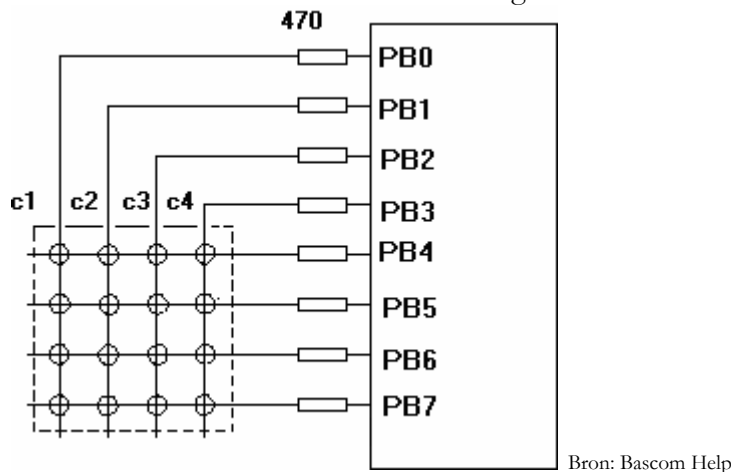
Doel

- Een matrix toetsenbord leren gebruiken

Theory

De ATmega88 AVR is (tijdens het schrijven van deze handleiding) een van de recentste chips in de AVR familie. Bovenop de standaard features heeft de ATmega88 meer interrupt bronnen dan zijn voorgangers. Die interrupten kunnen zeer goed worden gebruikt om een matrix uit te lezen. Er zijn matrix toetsenborden met 16 of 9 knoppen.

Een matrix kan het beste zo worden aangesloten:



Zoals u kunt zien zijn er slechts 8 I/O lijnen nodig om 16 knoppen te lezen. Dit uitlezen kan op 2 manieren. De eerste manier is “polling”, dat betekent dat we de software continu laten controleren of er gedrukt is. (Een do...loop structuur) Het probleem met polling is dat de controller niet veel additionele taken kan doen tijdens het pollen.

De tweede manier is het gebruiken van interrupts. Zoals gezegd kunnen de meeste pinnen op de ATmega88 als interrupt bron worden gebruikt. Met interrupts is het mogelijk de “matrix read” routine alleen uit te voeren nadat er op een knop is gedrukt. (De “matrix read” routine in de interrupt routine is hetzelfde als met polling, alleen wordt deze nu slechts 1 maal uitgevoerd per interrupt/toetsaanslag.)

Een kleine uitleg over interrupts is te vinden op de volgende pagina.

INTERRUPTS

Een interrupt is een gebeurtenis die de uitvoering van het programma stopt en springt naar een bepaalde interrupt routine. Interrupts kunnen worden geactiveerd door I/O pinnen maar ook de UART RxD pin en de Timers/Counters kunnen interrupts genereren.

Nadat een interrupt is afgehandeld, gaat het programma verder waar het programma was gebleven voordat de interrupt plaats vond.

U kunt interrupts gebruiken als u in uw systeem aan bepaalde taken prioriteit wilt geven.

We geven geen breadbord layout voor dit experiment, omdat de matrix niet op het breadbord past. De bedoeling is dat u de verbindingen maakt zoals aangegeven op de vorige bladzijde. Dus:

Verbind Pin1 van de matrix met een weerstand van 470 Ohms met portb.0,
Verbind Pin2 van de matrix met een weerstand van 470 Ohms met portb.1, etc,etc

Programmeer de ATMega88 met EDBexperiment10.bas en verbind het EDB met de PC door middel van de seriële kabel.

U zou nu de waarde 16 in uw terminal emulator moeten zien, als u op een knop drukt krijgt u een waarde < 16.

Om de interrupts en het GetKBD commando te begrijpen, kunt u het beste de opmerkingen in EDBexperiment10.bas lezen.

Dit experiment kan alleen worden uitgevoerd met Bascom versie 1.11.8.0 of hoger.

Als u een oude versie van Bascom gebruikt, kunt u een recentere versie downloaden van www.mcselec.com of u kunt gebruik maken van EDBexperiment10polling.bas een Matrix read programma zonder interrupts.

Kijk ook naar EDBexperiment10polling.bas dat ook het matrix toetsenbord uitleest, maar dan zonder interrupts. Controleer of u het verschil begrijpt tussen de beide manieren.

Experiment 11

3.4.3 RC5 afstandsbediening

Onderdelen

- 1x TSOP1736 of gelijkwaardige
- 1x Computer met COM poort
- 1x RC5 Afstandsbediening

Doel

- Een RC5 afstandsbediening leren gebruiken

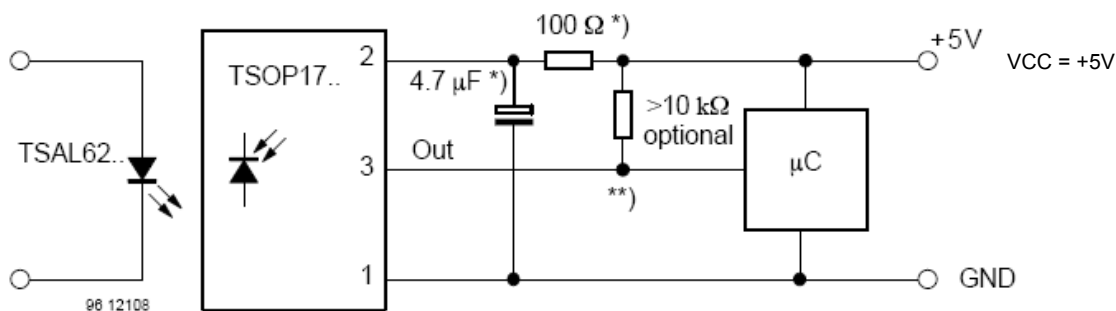
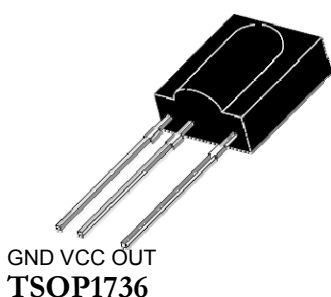
Er kan over gediscussieerd worden of een “afstandsbediening” in het hoofdstuk toetsenbord invoer valt. Hoedanook we kunnen een afstandsbediening als invoer-apparaat gebruiken.

Tegenwoordig heeft ieder huishouden verschillende afstandsbedieningen. Sommige van deze afstandsbedieningen gebruiken het RC5 protocol. RC5 is geïntegreerd in Bascom AVR. Daardoor kan afstandsbedieningsfunctionaliteit makkelijk worden toegevoegd aan het EDB.

Als u meer wilt weten over het RC5 protocol, kunt u deze application note van Philips eens lezen:

http://www.semiconductors.philips.com/acrobat_download/applicationnotes/AN10210_2.pdf

De TSOP1736 is een Infra Rood ontvangst module, deze tekening laat een standaard interface zien voor de TSOP1736.



*) recommended to suppress power supply disturbances

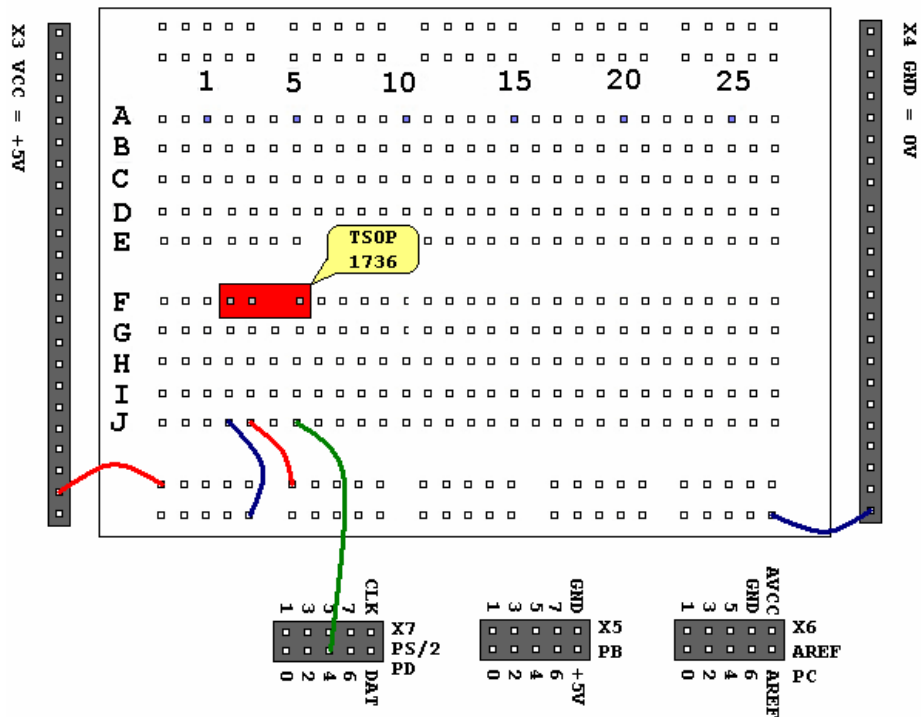
**) The output voltage should not be hold continuously at a voltage below 3.3V by the external circuit.

Bron: Vishay Semiconductors

(De TSAL diode bevindt zich in de afstandsbediening)

U kunt ook een andere Infra Rood ontvangst module gebruiken (zoals de SFH506 of de SFH 5110-36). Als u een andere module gebruikt moet u wel de pinbezetting controleren voordat u de module aansluit op het EDB.

Een voorbeeld breadbord layout voor dit experiment:



De meeste Philips afstandsbedieningen gebruiken RC5 maar ook de meeste universele afstandsbedieningen ondersteunen dit protocol. Als u geen van beide heeft kunt u het beste gewoon wat afstandsbedieningen uitproberen.

Probeer uw afstandsbediening door eerst EDBexperiment11.bas te openen en in de controller te programmeren. Verbind de seriële kabel tussen uw EDB en uw PC. Richt met de afstandsbediening op de bolvormige kant van de Infra Rood ontvangst module. Als u op een knop van de afstandsbediening drukt zou u andere waarden in de terminal emulator moeten zien verschijnen.

Dit experiment kan alleen worden uitgevoerd met Bascom versie 1.11.8.0 of hoger.

Lees de opmerkingen in EDBexperiment11.bas en controleer of u begrijpt wat er gebeurt.

Online bronnen:

RC5 achtergronden: <http://www.clearwater.com.au/rc5/>
<http://www.epanorama.net/links/irremote.html>

3.5 Geavanceerde I/O

In dit hoofdstuk bespreken we geavanceerde I/O. Om deze geavanceerde I/O te begrijpen speelt de UART een cruciale rol. (Vandaar deze volgorde in de hoofdstukken.) In hoofdstuk 3.5.1 zullen we niet ideale schakelaars bespreken en in de daarop volgende hoofdstukken zullen we zien hoe we de stand van een potmeter kunnen bepalen, hoe we een licht/donker schakeling kunnen maken met een LDR en tenslotte zullen we tonen genereren met een speaker.

3.5.1 Teller met anti dender ingang

Experiment

12

Onderdelen

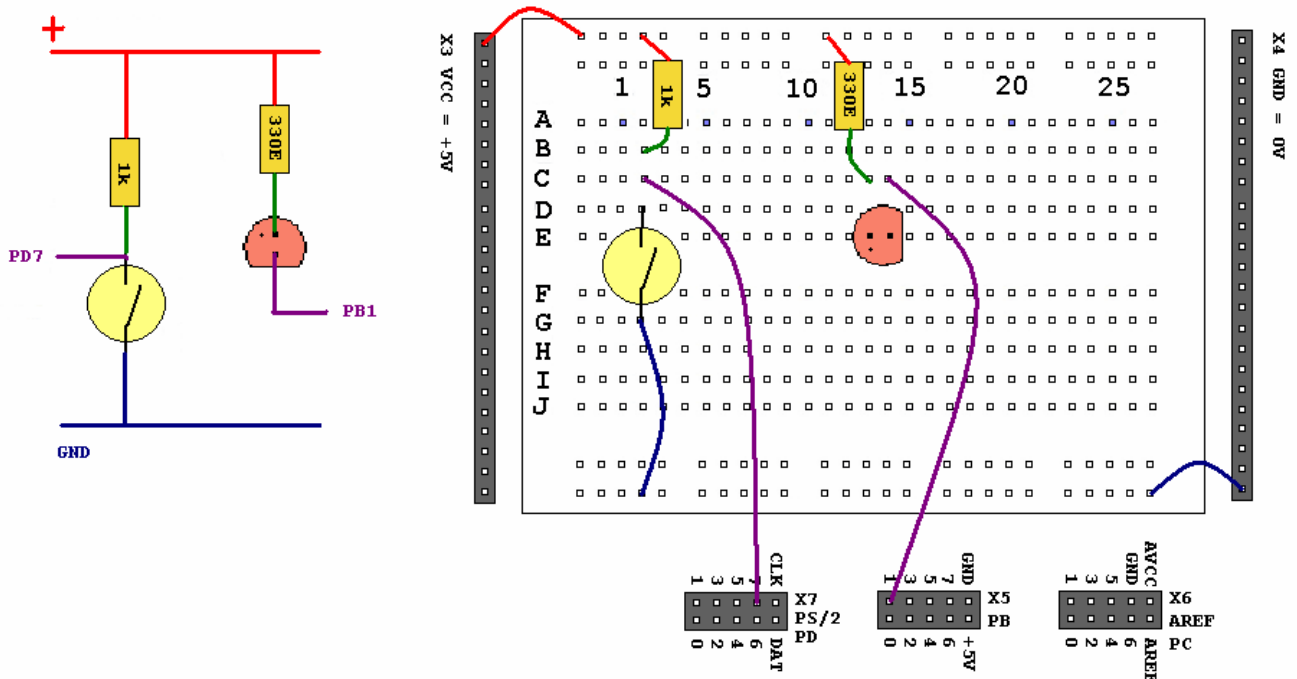
- 1x LED
- 1x Schakelaar of draad
- 6x Breadbord draad
- 1x Weerstand 330E, 1k

Doel

- Een niet-ideale schakelaar begrijpen
- Software anti dender

Schakelaars zijn niet ideaal. Dit is niet hinderlijk als u alleen een led wilt aansturen. Stel dat u een pulsteller wilt maken, en dat u de schakelaar verbindt met uw microcontroller als in experiment1. De telstand zou dan niet accuraat zijn door niet-ideaal gedrag van de schakelaar.

Bouw nog eens de opstelling van experiment 1:

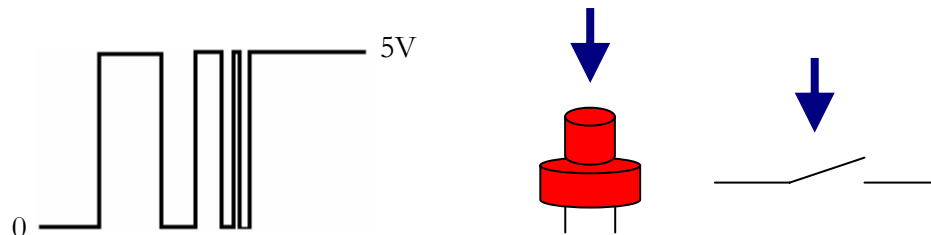


Programmeer de ATmega88 met EDBexperiment12a.bas en verbind de seriële kabel tussen uw EDB en uw computer.

Het EDBexperiment12a.bas programma “telt” het aantal keren dat u contact maakt met de schakelaar. Zoals u kunt zien in het terminal emulator scherm. Als u gelukt heeft telt het programma precies 1 keer iedere keer dat u contact maakt.

Vervang nu de schakelaar met een normale breadbord draad. En maak en verbreek het contact een aantal keer en bekijk de telstand. Zoals u kunt zien geeft dit een onnauwkeurige telstand. Dat wil zeggen de telstand is wel juist alleen soms telt het programma meerdere tikken voor iedere keer dat u contact maakt.

Dit is wat er gebeurt; iedere keer als u contact maakt zal de schakelaar niet meteen stabiel zijn. In andere woorden hij *denderd* zoals hier:



Dit verschijnsel wordt contact dender genoemd. Het verschijnsel duurt slechts enkele milliseconden, maar omdat de microcontroller op 8 MHz draait kan deze dalende en stijgende flanken detecteren in tijden van microseconden. Dus als u de schakelaar indrukt zoals in de situatie hierboven, zal de controller 4 tellen in plaats van 1.

De oplossing is het toevoegen van een *dode zone* waarin de controller geen veranderingen ziet in het spanningsniveau. Zo’n dode zone kan eenvoudig worden gemaakt door het toevoegen van een “Waitms 100” commando. Dit kan gedaan worden op de regel na het commando waar de controller de schakelaar heeft gecontroleerd.

Programmeer de microcontroller nu met EDBexperiment12b.bas, controleer de opmerkingen en kijk even naar het “Waitms 100” commando. Als uw controller nog steeds niet juist telt kunt u de waarde van Waitms veranderen naar 250.

Het toevoegen van het “Waitms” commando om een nauwkeurige telling te verkrijgen wordt “software anti dender beveiliging” genoemd.

Experiment 13



Keramisch 100 nF
"104"

3.5.2 Het GetRC commando

Onderdelen

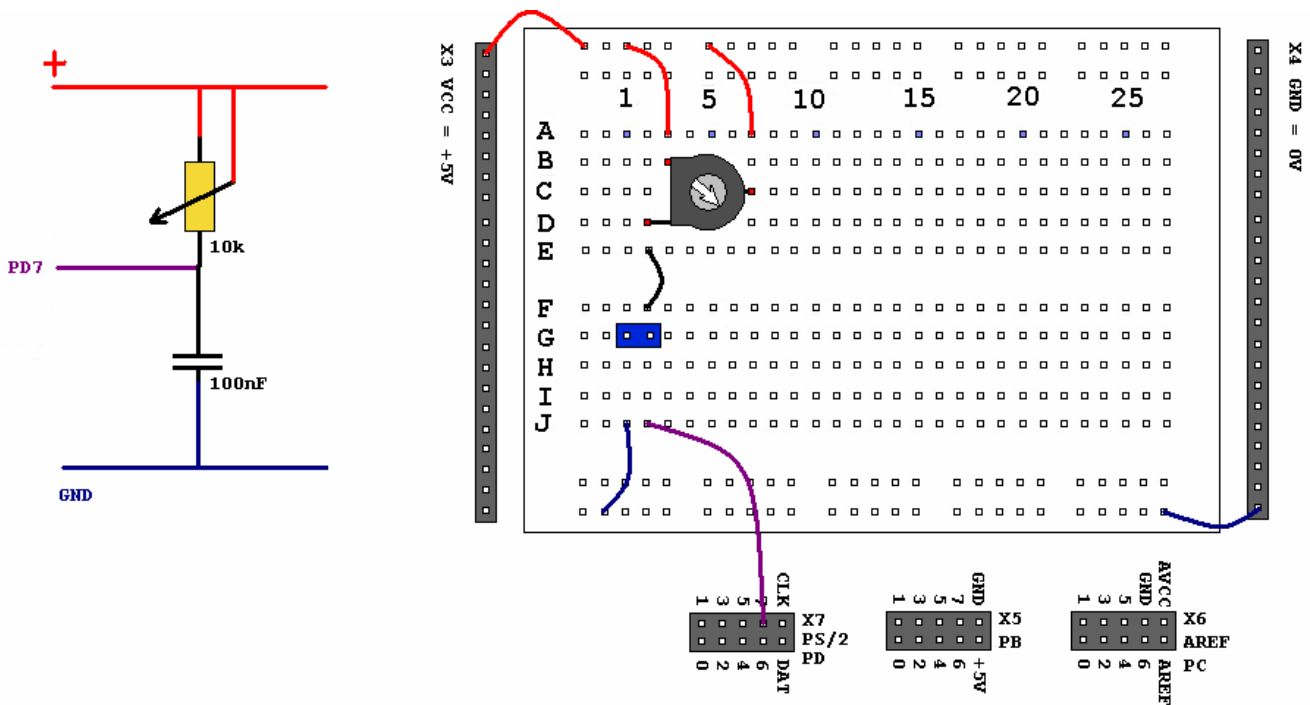
- 1x Keramische condensator 100nF
- 1x Potmeter 10k
- 7x Breadbord draad

Doel

- Kennis maken met GetRC

Met het GetRC commando kunt u de stand van de loper van een potmeter bepalen. GetRC kan alleen worden gebruikt voor niet nauwkeurige toepassingen.

Bouw onderstaande layout:



Op de keramische condensator staat een waarde in pico Farad.
Dus $100 \text{ nF} = 100000 \text{ pF}$. Het laatste getal op de condensator is het aantal nullen. (Dus 104 op de condensator = $100000 \text{ pF} = 100 \text{ nF}$)

De potmeter past niet op alle breadborden, als hij niet past kunt u het beste draden van 0,5 mm aan de potmeter solderen voordat u deze op het breadbord plaatst.

Open EDBexperiment13.bas en programmeer de ATmega88. Verbind de seriële kabel tussen uw EDB en uw computer. Als het goed is ziet u nu waarden in uw terminal emulator, u kunt deze waarden veranderen door aan de potmeter te draaien. Lees de opmerkingen in EDBexperiment13.bas en controleer of u begrijpt wat er gebeurt.

Experiment 14

3.5.3 Sirene met het SOUND commando

Onderdelen

- 1x Speaker RS267-6968 of gelijkw.
- 1x Elco $\approx 100\mu\text{F}$
- 1x Weerstand 120E
- 7x Breadbord draad

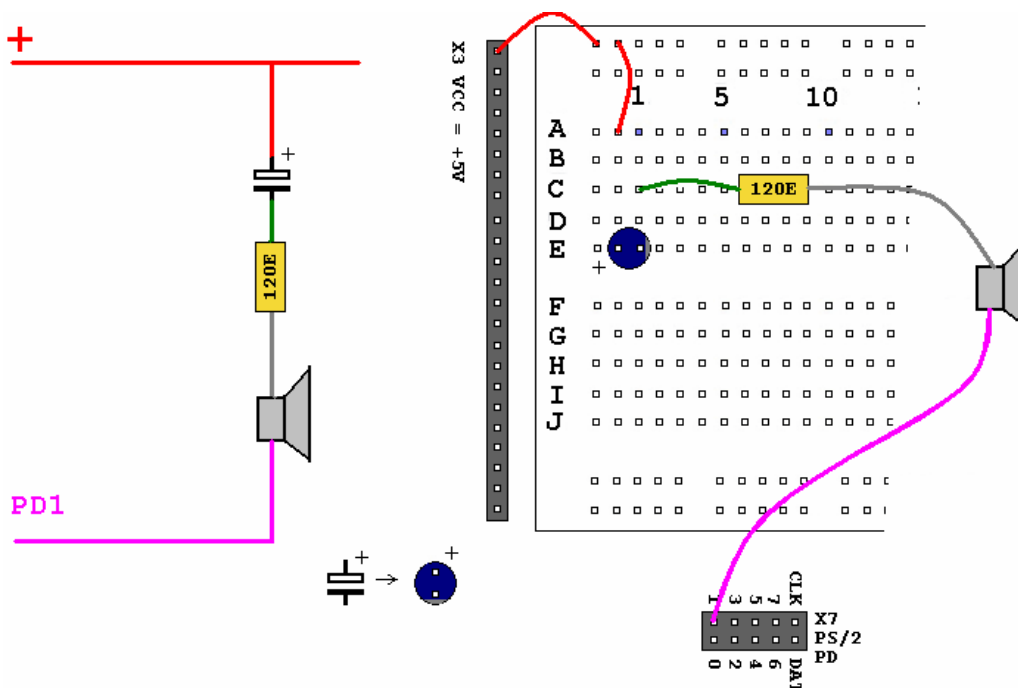
Doel

- Kennis maken met SOUND

Met het SOUND commando kunt u diverse tonen maken met een speaker. Dit kan van pas komen als u een toepassing heeft waarin u de gebruiker wilt inlichten dat er een bepaalde situatie optreedt. (Bijvoorbeeld een OK of fout piep enz.) Het is niet mogelijk met SOUND precieze frequenties te genereren. Als u precieze frequenties wilt kunt u een timer gebruiken. (Hoofdstuk 3.7.3)

Voor dit experiment kun u een ELCO gebruiken van ongeveer $100\mu\text{F}$. (maar $47\mu\text{F}$, of $220\mu\text{F}$ werkt ook) Verder kunt u een oude PC speaker gebruiken of u kunt RS 267-6968 bestellen. In de meeste gevallen past de speaker niet op het breadbord, in dat geval kunt u een draad solderen aan de speaker. Bij voorkeur een massieve koperdraad met een diameter van ongeveer $0,5\text{ mm}$.

Bouw onderstaande layout:



Programmeer de ATmega88 met EDBexperiment14.bas en controleer of u de programma code begrijpt. U zou een sirene moeten horen.

3.6 AD en DA conversie

Microcontrollers zijn digitaal, maar ze kunnen ook analoge naar digitale en digitale naar analoge conversie uitvoeren. Dit geeft de mogelijkheid analoge signalen te meten, maar ook kunt u “analoge” uitgangen maken. Dit hoofdstuk leert u hoe u analoge signalen kunt converteren of creëren.

Experiment 15a

3.6.1 PWM uitgang

Onderdelen

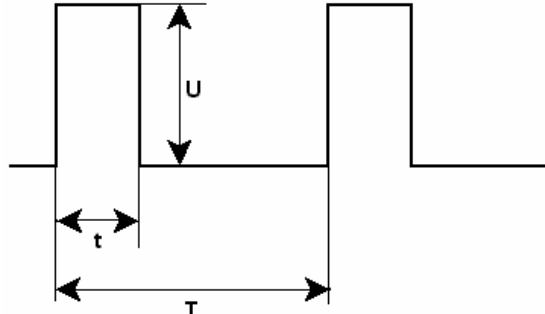
1x MOSFET IRF520 of gelijkw.
1x Lamp 6V
6x Breadbord draad

Doel

- Het hoe en wat over PWM leren

PWM is een vorm van digitale uitvoer, maar we kunnen het als analoge uitgang gebruiken.

PWM signalen zijn digitale DC signalen die er zo uit zien:



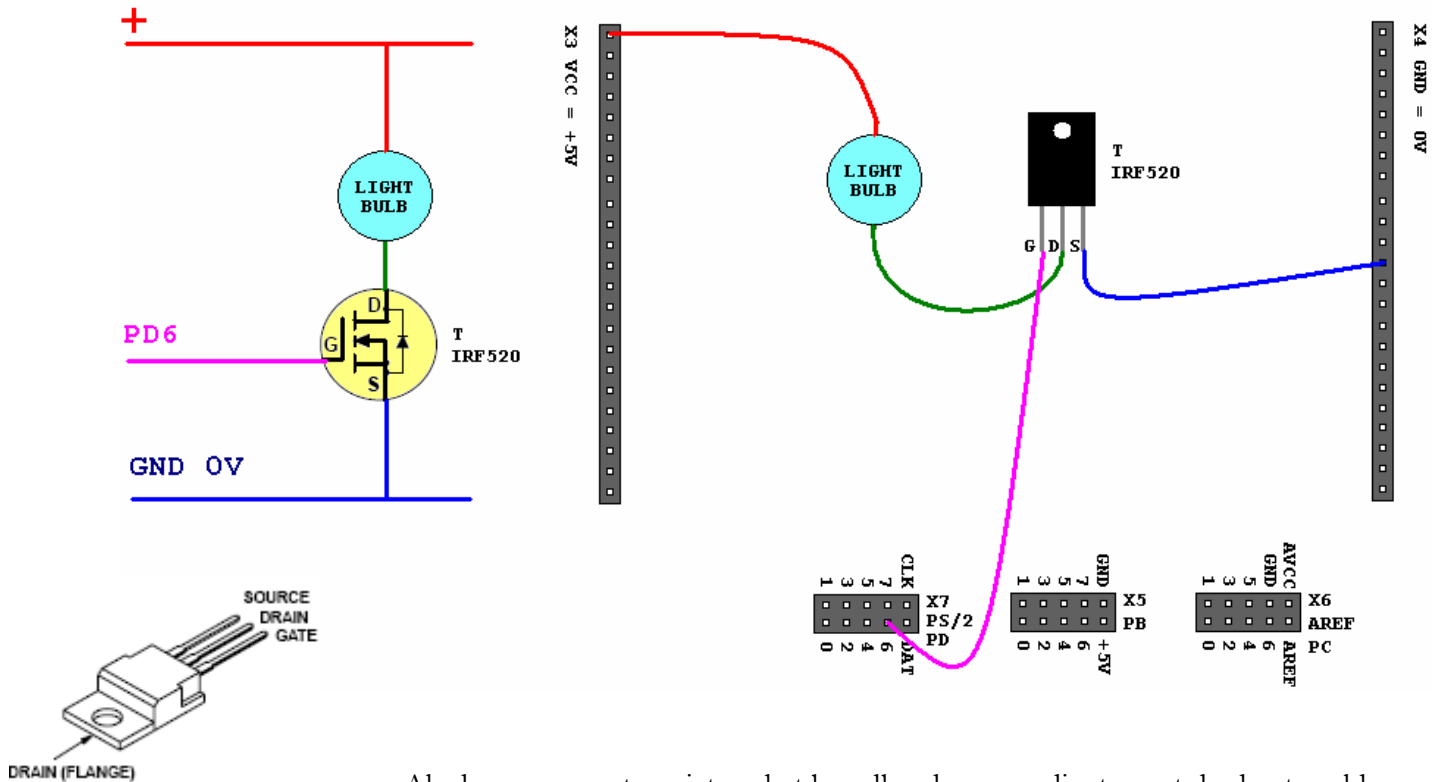
Stel u voor dat we het bovenstaande signaal aanbrengen op een lamp. Zouden we nu tijd t vergroten dan zou de gemiddelde spanning over de lamp ook groter worden. We kiezen tijd T zodanig dat we de lamp niet zien knipperen. Voor het oog lijkt het dus alsof de lamp analoog wordt geregeld.

Als u iets anders dan een lamp aansluit, dan is het toevoegen van een condensator meestal genoeg om het analoge signaal te creëren.

Het zojuist beschreven systeem wordt pulse breedte modulatie genoemd. (In het Engels PWM Pulse Width Modulation). PWM is geïntegreerd in de meeste AVR chips en het wordt ondersteund door Bascom.

Dit experiment toont een voorbeeld van PWM.

Bouw onderstaande layout:



Als de componenten niet op het breadbord passen, dient u wat draden te solderen.

Programmeer de ATmega88 met EDBexperiment15a.bas en verbind de seriële kabel tussen uw EDB en uw computer.

Het licht dat door de lamp wordt uitgestraald zou moeten variëren en u kunt de PWM waarden lezen in uw terminal emulator scherm.

Lees de opmerkingen in EDBexperiment15a.bas en kijk of u begrijpt wat er gebeurt. Het EDBexperiment15a.bas programma gebruikt een timer, we zullen meer over timers vertellen in hoofdstuk 3.7.3.

Online bronnen:

PWM achtergrond: <http://www.netrino.com/Publications/Glossary/PWM.html>
<http://www.4qdttec.com/pwm-01.html>

IRF520: <http://ec.irf.com/v6/en/US/adirect/ir?cmd=catProductDetailFrame&productID=IRF520>

Experiment 15b

3.6.2 AD conversie met een LDR

Onderdelen

- 1x LDR NSL19-MS51
RS596-141
- 1x Computer met COM poort
- 2x Weerstand 2k2

Doel

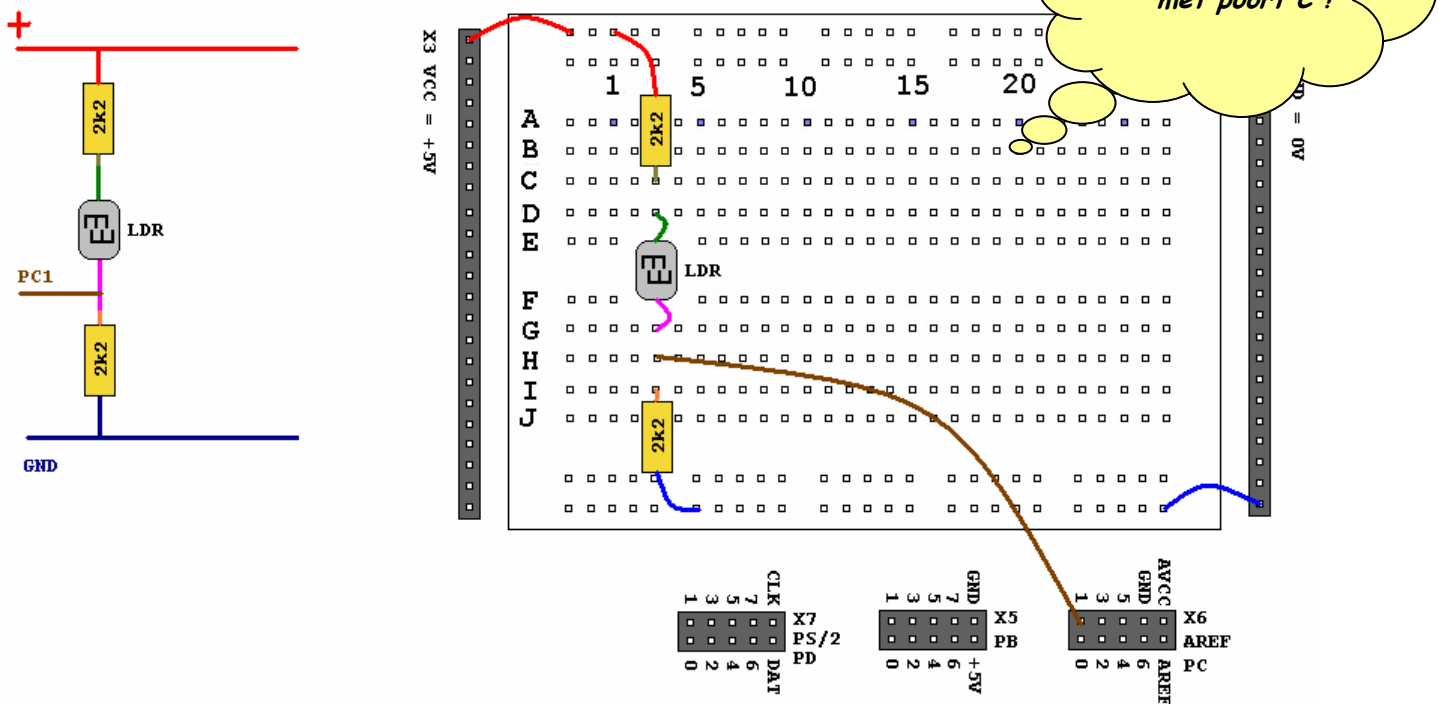
- Een AD converter leren gebruiken
- Kennis maken met LDR's

Dit experiment laat een standaard applicatie met AD conversie zien en bereid u voor op oefening 2 “de LDR licht schakelaar”.

LDR's zijn lichtgevoelige weerstanden (Light Dependant Resistors), dat betekend dat de weerstandswaarde veranderd als er meer of minder licht op de weerstand valt. Voor dit experiment kunnen de meeste LDR's worden gebruikt.

De LDR NSL19-MS51 heeft een weerstand van 5k in daglicht (zonder zonnenschijn). Een weerstand van 1M2 in het duister en een weerstand van >35M in complete duisternis. Voor dit experiment zijn de meeste LDR's geschikt. Als u een andere LDR gebruikt zult u andere waarden uitlezen in uw terminal.

Bouw onderstaande layout:



Programmeer de ATmega88 met EDBexperiment15b.bas er verbind de seriële kabel tussen uw EDB en uw computer.

Als het goed is verschijnen er nu waarden in uw terminal emulator. U kunt de waarden veranderen door uw hand boven de LDR te houden of met een lamp op de LDR te schijnen.

Lees de opmerkingen in EDBexperiment15b.bas en controleer of u de werking begrijpt.

> Verbind PC1 ook eens met ground en controleer de terminal waarden, kijk ook welke waarden er zijn indien u PC1 met VCC = +5V verbind.

Online bronnen:

LDR achtergrond: http://www.radio-electronics.com/info/data/weerstand/ldr/light_dependent_weerstand.php

<http://www.technologystudent.com/elec1/ldr1.htm>

Oefening

2

Oefening 2, LDR licht schakelaar

Test uw kennis uit de voorgaande hoofdstukken door deze oefening te maken.

Maak een LDR licht schakelaar,

- Verbind een LED of een lampje (zoals u kunt vinden in uw fietslicht) met uw microcontroller. Let erop dat u de lamp **niet direct aansluit**, gebruik een BS170 om de relatief grote stroom te schakelen.
- Controleer of u de lamp kunt aan- en uitschakelen met een simpel programma zoals u heeft gezien in experiment 1.
- Gebruik een LDR zoals u heeft gezien in het voorgaande hoofdstuk en schakel de lamp aan als het donker is en uit als het niet donker is. Als het werkt, gefeliciteerd!! Indien niet, u kunt de oplossing van deze opdracht vinden op de EDB-CD bestandsnaam: **Exercise2.bas**.

Experiment 16

3.6.3 Low cost voltmeter

Onderdelen

- 1x Computer met COM poort
- 1x Weerstand 1k2, 3k9
- 1x Potmeter 10k
- 1x Voltmeter

Doel

- Een AD converter leren gebruiken
- Een AD toepassing begrijpen

Theorie

Als u de AD waardes in experiment 15 heeft gecontroleerd dan heeft u kunnen concluderen dat een spanning van 0V overeenkomt met een AD waarde van ongeveer 0 en een spanning van 5V ongeveer overeenkomt met 1024. Eigenlijk wordt AD waarde 1024 al bereikt bij 1,1V.

We kunnen nu een voltmeter maken voor spanningen tussen 0 en ongeveer 5,0V met een resolutie van 0,1V.

Omdat we alleen spanningen kunnen meten tot 1,1V gebruiken we een paar weerstanden als spanningsdeler. Op die manier kunnen we een AD spanning van 1V een signaal van 5 volt laten voorstellen.

De tekening aan de linkerkant laat zien welke weerstanden u kunt gebruiken voor de spanningsdeler.

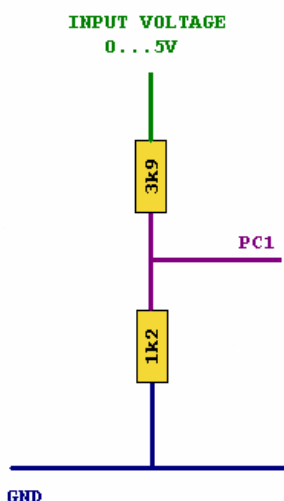
De conclusie is dat we 1024 moeten delen door 11 dit geeft 93,09. Het heeft de voorkeur te werken met gehele getallen omdat microcontrollers niet zo best kunnen delen. (Ze kunnen wel delen, maar met rest, $1024:11 = 93 \text{ rest } 1$)

Daarom ronden we 93,09 af naar 93. Als we nu de AD waarden *berekenen* voor iedere volt, dan vinden we de tabel aan de rechterkant van deze pagina.

Stelt dat we een spanning aanbrengen op port C.1 en dat de AD waarde 190 is. 190 is geen waarde in de tabel... dus moeten we ergens afronden.

Laten we zeggen dat alle waarden onder 93 gelijk zijn aan 0,0V. Waardes van 93 tot 185 zijn 0,1V, waardes van 186 tot 278 zijn 0,2V, etc, etc...

Het enige dat we dan hoeven te doen is een programma te schrijven dat controleert of de AD waarde in een bepaald bereikt ligt... zoals hier:



V	AD waarde
1,1	1024
1	931
0,9	838
0,8	745
0,7	652
0,6	559
0,5	465
0,4	372
0,3	279
0,2	186
0,1	93
0	0

```

Test_value = 93      '(The initial value "11" because everything below 11 is 0,0V)
AD_value = 190
Voltage = 00

```

```

Not_ready:
  If AD_value ≤ Test_value then goto ready
  Test_value = Test_value + 93
  Voltage = Voltage + 1
goto not_ready

```

```

Ready:
Print Voltage

```

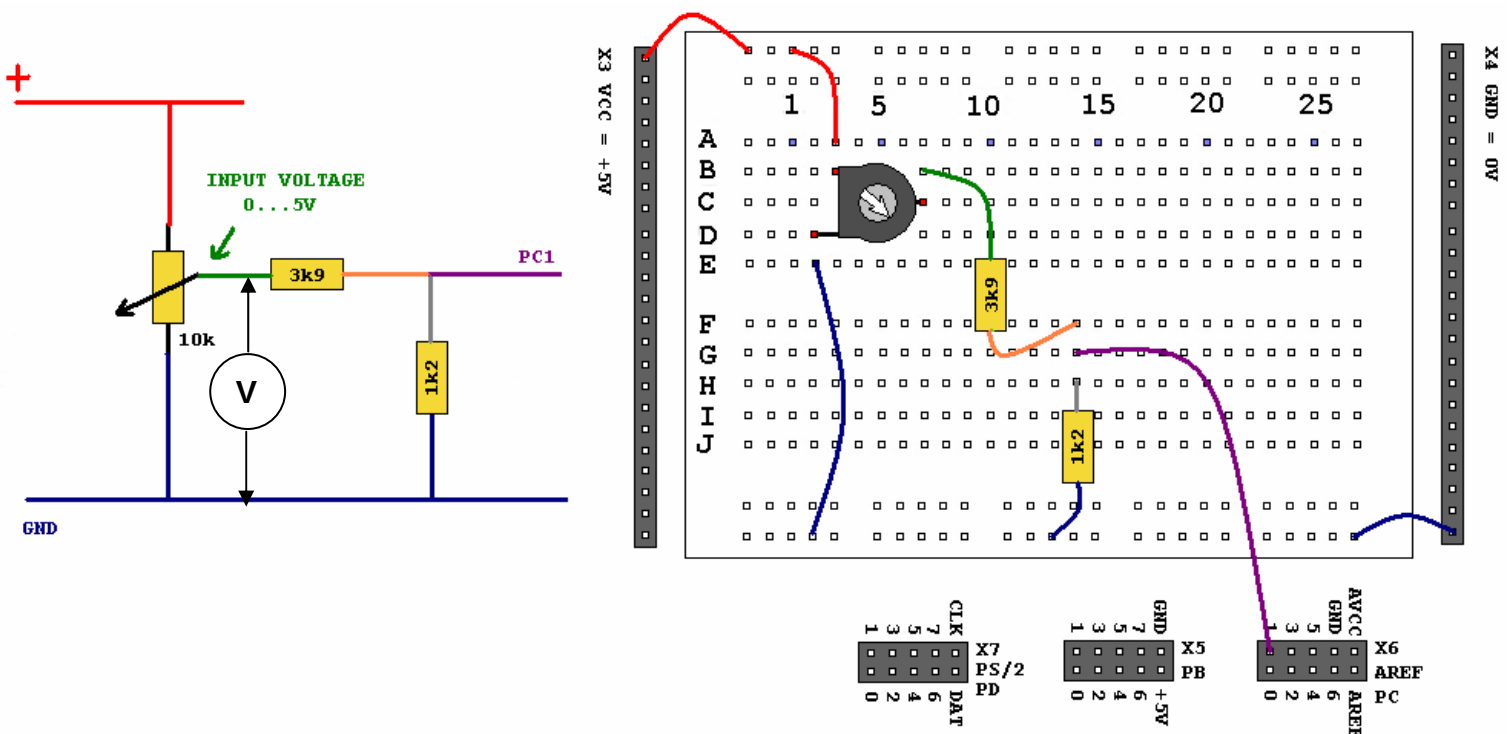
In ons voorbeeld was AD_value gelijk aan 190, de initiële waarde van Test_value is 93. Dus de IF-vergelijking is FALSE... nu tellen we 93 op bij test value en 1 bij voltage. (Daarmee hebben we geconstateerd dat de gemeten spanning niet gelijk is aan 0,0V.)

Test_value is nu 186 dus de IF-vergelijking is nog steeds FALSE, dus de spanning is niet 0,1V. Opnieuw tellen we 93 op bij test value en 1 bij voltage.

Test_value is nu 279 en de IF-vergelijking is TRUE! Nu springen we naar de ready tag. Als we nu de Voltage variabele zouden weergaven zien we 0,2V. We hebben nu de AD waarde geïnterpreteerd zonder gebruik te maken van een deling.

Het programma hierboven geeft een weergave van 0 tot 1,1 V. Delen we nu alle test waarden door 4, dan krijgen we een volledige schaal.

→ Programmeer uw EDB met EDBexperiment16.bas, verbind de seriële kabel en bouw de layout hieronder. **Controleer** of de controller werkt zoals aangegeven in de theorie door aan de potmeter te draaien en de spanning tussen GND en PC1 te meten met een voltmeter.



Oefening

3

Oefening 3, Voltmeter met bargraph uitlezing

Test u kennis uit de voorgaande experimenten door deze oefening te maken.

Maak een bargraph uitlezing voor de voltmeter uit het voorgaande experiment.

- Verbind 8 LED's met Port B. Als de spanning 5V is zouden alle LED's aan moeten zijn, bij een spanning van 2,5V zouden LED's PD0...4 aan moeten zijn, enz, enz.

Hint:

```
Select Case AD_value
  Case Is < 128 : Portb = &B11111110
  Case 128 To 256 : Portb = &B111111100
  Case 256 To 384 : Portb = &B111111000
  Case 384 To 512 : Portb = &B111110000
  Case 512 To 640 : Portb = &B111100000
  Case 640 To 768 : Portb = &B110000000
  Case 768 To 896 : Portb = &B100000000
  Case Is > 896 : Portb = &B000000000
End Select
```

- Geef de gemeten spanning ook weer op een LCD.

Als dit gelukt is, gefeliciteerd!! Indien niet, de oplossing van deze oefening is te vinden op de EDB-CD bestandsnaam: **Exercise3.bas**.

Oefening

4

Oefening 4, Een variabele lichtbron

Test uw kennis van de voorgaande hoofdstukken door het maken van deze oefening.

Maak een variabele lichtbron. De bedoeling is dat u de lichtsterkte kunt wijzigen door aan een potmeter te draaien.

- Verbind een lampje (zoals u kunt vinden in uw fietslicht) met uw microcontroller. Let erop dat u de lamp **niet direct aansluit**, gebruik een IRF520 om de relatief grote stroom te schakelen.
- Controleer of u de lamp kunt aan- en uitschakelen met een simpel programma zoals u heeft gezien in experiment 1.
- Maak gebruik van de PWM functionaliteit om de lichtsterkte van de lamp te regelen.
- Gebruikt een potmeter in combinatie met A/D conversie, zoals in het vorige experiment om de lichtsterkte van de lamp te regelen/variëren.

Als dat gelukt is, gefeliciteerd!! Indien niet, de oplossing van deze oefening is te vinden op de EDB-CD bestandsnaam: **Exercise4.bas**.

3.7 Andere mogelijkheden van de controller

In dit hoofdstuk gaan we verder in op veelgebruikte mogelijkheden van de controller, die we niet hebben gezien in de voorgaande experimenten. In dit hoofdstuk leert u hoe het geheugen, de watchdog en de interne timers te gebruiken.

3.7.1 Geheugen

Theorie

Onderdelen

- *geen*-

Doel

- Het geheugen leren gebruiken

De ATmega88 microcontroller heeft 3 soorten geheugen.

- FLASH -> Bootloader FLASH
Application FLASH (Toepassingsflash)
- RAM
- EEPROM

Het FLASH geheugen (8 kilobytes) kan **alleen** worden geprogrammeerd met een externe programmer zoals de STK200/300 dongle of de Wiazania USB STK programmer (zie hoofdstuk 4).

Er is echter 1 uitzondering, de bootloader. Wanneer u de bootloader optie gebruikt, wordt het FLASH in 2 delen verdeeld. De bootloader zelf bevindt zich in het “Bootloader FLASH” geheugen en wordt geprogrammeerd met een STK programmer. Hierna kan de Application FLASH worden geprogrammeerd met de seriële kabel (hoofdstuk 2.3).

In de Application FLASH slaan we de programmacode op van het programma dat we willen uitvoeren. Als u de bootloader niet gebruikt kunt u het gehele FLASH geheugen gebruiken om uw programmacode op te slaan. FLASH is **non volatile** geheugen dat betekent dat het geheugen behouden blijft als er geen voeding op de chip aanwezig is.

Terwijl FLASH wordt gebruikt om uw programma op te slaan, wordt **RAM geheugen** gebruikt voor het opslaan van de invoer en het resultaat van rekenkundige en logische operaties. Ook bytes die worden verzonden/ontvangen van de UART worden eerst in RAM opgeslagen. Ook het terugkeer adres van jump instructies wordt opgeslagen in RAM.

RAM (Random Access Memory) is een vorm van tijdelijke geheugenopslag. Na een operatie moet u de ram data opslaan of overschrijven. RAM is **volatile** dat betekend dat het geheugen verloren gaat als de voeding niet aanwezig is.

Een speciale vorm van geheugen is **EEPROM geheugen** (Electrically Erasable/Programmable Read Only Memory) Laat deze term u niet misleiden, in vroegere tijden kon een EEPROM alleen worden gelezen... toen heette het ook ROM geheugen en het kon NOOIT worden gewist door de controller zelf. De ROM kon alleen worden geprogrammeerd met een ROM programmer en de ROM kon worden gewist met een UV licht box. (Deze chips hadden een glazen venster).

Tegenwoordig kan de controller de ROM *elektrisch* wissen en programmeren, daardoor is het mogelijk data te bewaren, zelfs wanneer de voeding niet aanwezig is. Dit is dus **non volatile** geheugen.

De EEPROM kan van pas komen als uw toepassing instellingen heeft die kunnen worden ingesteld door de gebruiker. Stel u voor dat u iedere keer als u, uw TV uitzette alle zenders opnieuw moest programmeren... dat hoeft gelukkig niet, omdat er een EEPROM in uw TV aanwezig is.

Experiment 17

Verbind de seriële kabel tussen het EDB en uw computer, programmeer EDBexperiment17.bas en start de Bascom Terminal Emulator.

Het programma zal vragen om een alfanumerieke invoer. Waarna het programma deze data zal opslaan. Volg de instructies op het scherm en kijk wat er gebeurt als u de voeding uitschakeld en weer inschakeld.

Heeft u geconstateerd dat de EEPROM data bewaard bleef?

Online bronnen:

Atmel AVR familie:

http://www.atmel.com/dyn/products/param_table.asp?family_id=607

3.7.2 Watchdog

Onderdelen

- geen -

Doel

- De Watchdog leren gebruiken

Een watchdog timer is een beveiliging die voorkomt dat de controller vastloopt. Als u de watchdog gebruikt, moet uw software de timer periodiek resetten.

Theorie

Dus als de controller vastloopt (eventueel na een elektrisch probleem) of als er externe hardware problemen zijn die ervoor zorgen dat de controller niet meer werkt, zal de watchdog niet meer gereset worden. Daardoor zal er een watchdog timer overflow voorkomen, hierdoor zal de controller worden gereset. Daarna zal de controller opnieuw starten en zal het programma weer van begin af aan worden uitgevoerd.

Een typische watchdog implementatie ziet er zo uit:

Config Watchdog = 2048
Start Watchdog

'reset after 2048 mSec
'start the watchdog timer

Do
'Print "Hello"
Reset Watchdog
'Your program code goes here, repeat the Reset Watchdog statement
Loop
End

Als u wilt kunt u wat experimenteren met de watchdog, u kunt het Watchdog.bas programma vinden op de EDB-CD. Haal de ' -tekens voor "Print "Hello" weg en plaats ' -tekens voor "reset watchdog". U kunt nu de watchdog reset zien in het Terminal scherm.

De maximum tijd voor de watchdog van de ATmega88 = 8 seconden. Maar voor de meeste toepassingen is een dergelijke tijd veel te lang. Het verdient de voorkeur 2048 of 1024 te gebruiken.

Experiment 18

3.7.3 Een timer gebruiken

Onderdelen

-geen-

Doel

- Een timer leren gebruiken

De ATmega88 heeft ingebouwde timers. U kunt timers gebruiken voor het meten van de tijd tussen signaalveranderingen of u kunt uw eigen wacht routine schrijven. Dit experiment laat een voorbeeld van een seconden timer zien.

Verbind de seriële kabel tussen uw computer en uw EDB, programmeer de controller met EDBexperiment18.bas en open de Bascom Terminal.

U zult nu een seconden teller zien in het Terminal venster. Lees de opmerkingen in EDBexperiment18.bas en controleer of u begrijpt wat er gebeurt.

Opmerking:

Het EDBexperiment18.bas programma is accuraat voor een minuut of 10, het programma maakt gebruik van de interne RC Oscillator van de ATmega88. Als u een precieze tijd wilt kunt u beter een real time clock IC gebruiken, zoals de Maxim Dallas DS1307 Real Time Clock. De DS1307 gebruikt een I²C interface.

Meer over I²C in het volgende hoofdstuk.

OVERZICHT VAN DE BESCHIKBARE TIMERS

De ATmega88 heeft de beschikking over deze timers:

8-bit Timer/Counter0 met PWM

16-bit Timer/Counter1 met PWM

8-bit Timer/Counter2 met PWM en Asynchrone Werking

U kunt meer vinden over de initialisatie en de interrupt bronnen van deze timers in de Bascom help. Druk op F1 in de Bascom AVR IDE en kijk naar "CONFIG TIMER" in de index.

Experiment 19

3.8 Andere interfaces

Nast de UART zijn er nog andere interfaces die vaak worden gebruikt. In dit hoofdstuk bespreken we het I²C protocol en zullen we zien hoe we de USB module op het EDB kunnen gebruiken.

3.8.1 De I²C bus

Theorie

Onderdelen

- 1x DS1624
- 2x Weerstand 4k7
- 1x Computer met COM poort

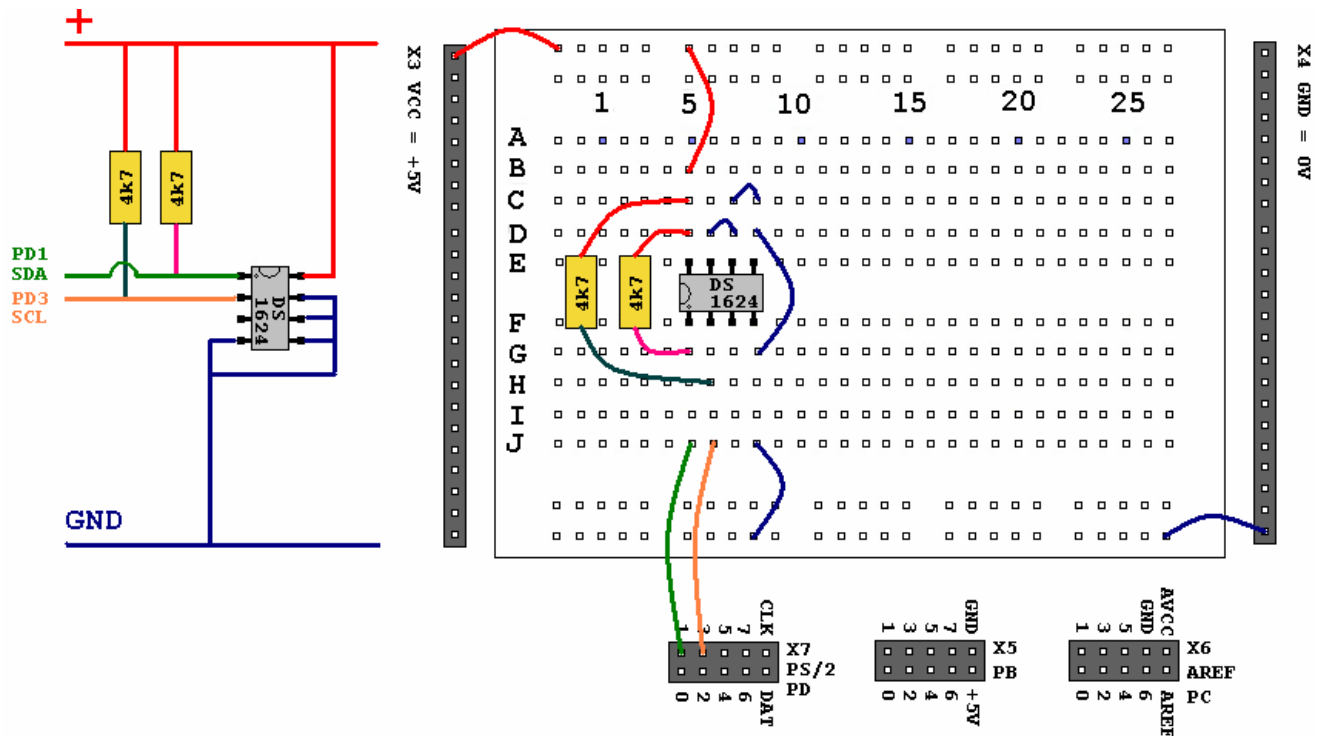
Doel

- De I²C bus leren gebruiken

De DS1624 is een digitale thermometer met een meetbereik van -55°C tot +125°C. Het uitlezen van deze temperatuur vindt plaats via de I²C bus. I²C is een serieel protocol dat gebruikt maakt van 2 I/O lijnen. Deze 2 lijnen zijn Seriële DATA (SDA) en Seriële CLock (SCL).

Bij I²C wordt de microcontroller MASTER genoemd en de DS1624 slave. De master is ook de chip die de SCL klok genereert. U kunt vele slave chips aansluiten maar slechts 1 master.

Bouw onderstaand schema:



Als u meer wilt weten over de I²C bus, kunt u het beste de DS1307 datasheet van Maxim/Dallas semiconductor lezen. We gebruiken in dit experiment een andere chip dan de DS1307 maar het protocol is hetzelfde. En het datasheet van de DS1307 is wat beter te begrijpen.

Het datasheet van de DS1307 kan worden gevonden op de EDB-CD of kan worden gedownload van www.maxim-ic.com. Begin met lezen vanaf hoofdstuk 5 onder “2-WIRE SERIAL DATA BUS” en stop met lezen na pagina 8. (Denk eraan dat we dus een andere chip gebruiken, het principe is hetzelfde maar de address- and command bytes niet.)

Programmeer de ATmega88 met EDBexperiment19.bas en verbind de seriële kabel tussen uw EDB en uw computer.

Als het goed is kunt u nu de temperatuur lezen in uw terminal emulator.

Lees de opmerkingen in EDBexperiment19.bas en controleer of u begrijpt wat er gebeurt.

Online bronnen:

I2C Achtergrond:

<http://www.semiconductors.philips.com/markets/mms/protocols/i2c/>

I2C Interface Ondersteuning voor Windows:

<http://www.microsoft.com/whdc/archive/i2c.msp>

3.8.2 USB Interface

De meeste computers zijn tegenwoordig voorzien van USB, toch nemen ontwikkelaars niet vaak de stap om USB toe te passen in embedded systemen. Dit hoofdstuk leert u USB toe te passen.

Eerst zijn er 3 opties waaruit u moet kiezen voordat u uw hardware ontwikkeld.

1. USB apparaat als virtuele COM poort

Dit is de meest eenvoudige mogelijkheid om USB toe te passen, u sluit uw *apparaat/device* aan op de USB poort van de computer en het besturingsysteem zal de device behandelen alsof het een COM poort is.

Voordelen:

- Eenvoudig
- Compatibel met terminal emulator programma's en bestaande software

Nadelen:

- Gebruiker ziet de COM poort in het configuratie scherm, onprofessioneel
- COM poort kan onjuist worden ingesteld
- Gebruiker moet 2 drivers installeren

2. USB apparaat met standaard VID&PID en driver

In het algemeen: Toepassingen hebben geen toegang tot USB poorten, maar ze hebben toegang tot bepaalde devices/apparaten met een bepaalde VID&PID (Vendor & part ID) op de bus.

Met deze tweede optie kunt u de standaard drivers gebruiken die worden geleverd door Linx Technologies (www.linxtechnologies.com) om verbinding te maken met uw embedded systeem.

Voordelen:

- Relatief eenvoudig in gebruik

Nadelen:

- U moet de VID&PID van Linx gebruiken.

3. USB apparaat met uw eigen VID&PID en driver

Met deze optie kunt u uw eigen USB device maken. Als u uw VID koopt via MCS Electronics krijgt u er automatisch een tool bij dat het *.inf setup-bestand voor u maakt. Als u uw VID niet via MCS heeft gekocht moet u handmatig een *.inf bestand schrijven.

Voordelen:

- U kunt uw eigen VID&PID gebruiken

Nadelen:

- U moet uw eigen VID kopen bij MCS of het USB consortium
- U moet uw eigen *.inf bestand schrijven.

USB devices/apparaten worden hoofdzakelijk gebruikt voor een verbinding met een computer, daarom moet u vrijwel altijd ook een PC applicatie schrijven *bovenop* op het programma van de microcontroller. Dat en de prijs van een USB module (en een optioneel VID) zijn obstakels voor de implementatie van USB. In de toekomst zijn COM-poorten niet langer beschikbaar op computer systemen en zullen USB modules voordeliger worden. Dat is de reden dat we dit USB hoofdstuk aan de EDB handleiding hebben toegevoegd.

Een uitgebreide verhandeling over USB valt buiten het doel van deze handleiding. (www.usb.org/developers) Wel zullen we beschrijven hoe u het EDB kunt gebruiken zoals beschreven in optie 1 en 2. We zullen ook een Visual Basic 6 voorbeeld programma laten zien.

De derde optie zal niet worden beschreven, we zullen alleen een tool introduceren om de EEPROM van de USB module te veranderen. Dit heeft u nodig om het VID&PID van de module te veranderen.

Online bronnen:

USB module: <http://www.linxtechnologies.com/>

USB chip in de Linx module: <http://www.ftdichip.com/Products/FT232BM.htm>

USB algemeen: www.usb.org

Koop uw eigen vendor ID: www.mcselec.com

3.8.3 Installatie procedure



Voor het gebruik van optie 1 (virtuele COM-poort) dient u de installatie procedure te volgens zoals beschreven in hoofdstuk 3.8.3.1. Voor de overige opties dient u de installatie procedure te volgens zoals in 3.8.3.2.

Als u wilt omschakelen van optie 1 (virtuele COM-poort) naar optie 2, 3. Of als u wilt omschakelen van optie 2, 3 naar optie 1 (virtuele COM-poort), dan moet u eerst de oude driver verwijderen. De de-installatie procedure is beschreven in hoofdstuk 3.8.3.3

Zorg ervoor dat uw computer een werkende USB poort heeft. USB poorten werken niet onder Windows NT en sommige versies van Windows 95. Wij adviseren u iedere keer dezelfde USB poort te gebruiken om uw EDB aan te sluiten. Op die manier hoeft u niet steeds de drivers opnieuw te installeren.

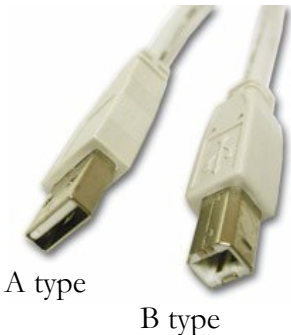
3.8.3.1 Installatie procedure Virtuele COM-poort Windows

Zorg ervoor dat de voeding is aangesloten op het EDB en verbind een USB-A naar USB-B kabel tussen uw EDB en uw computer.

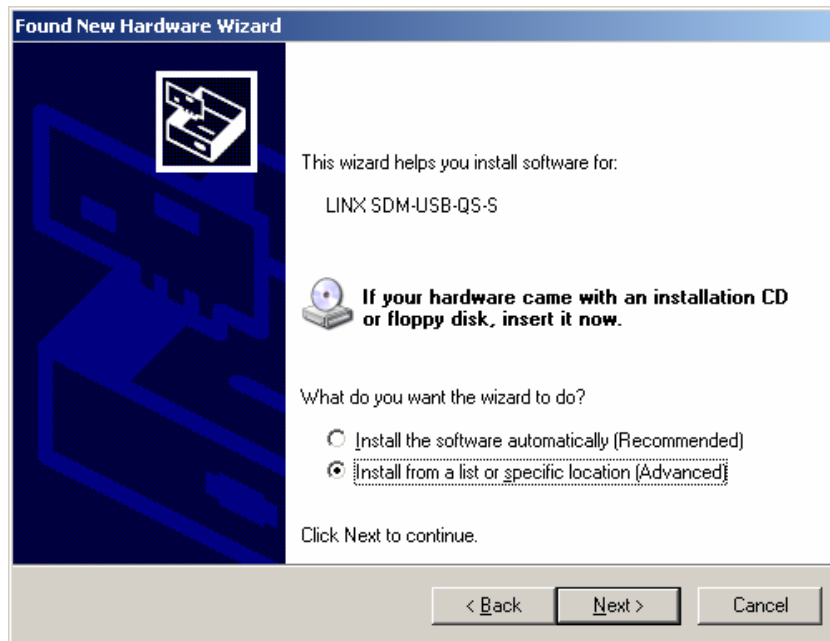
U zou nu dit venster moeten zien:



Selecteer “No, not this time” en klik “Next”

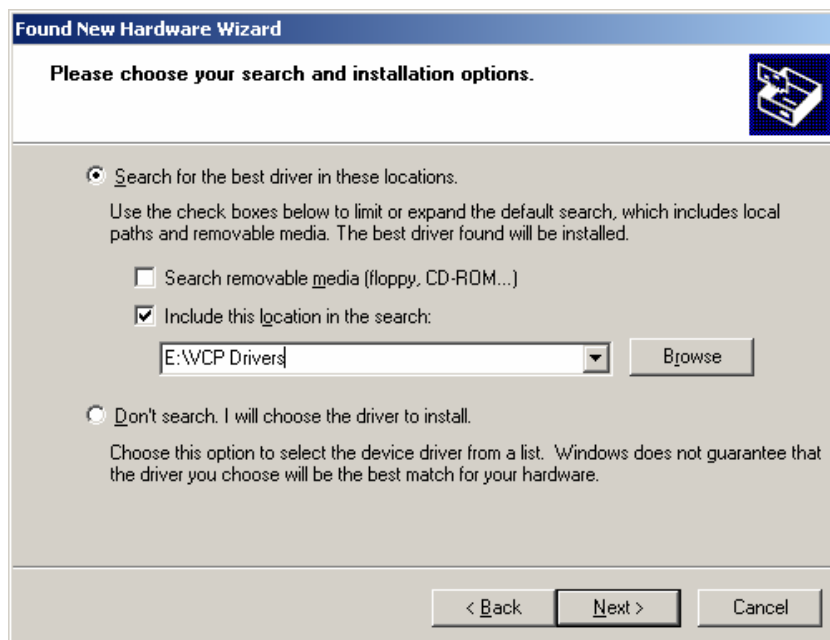


Na een ogenblik zou u dit venster moeten zien:



Selecteer “Install from a list or specific location (Advanced)” en klik op “Volgende”

Na een ogenblik zou u dit venster moeten zien:



Plaats de EDB-CD in uw CD/DVD drive en druk “Browse” navigeer naar:
“Uw Cd-rom drive”:\ USB_Module\Virtual_COM_Drivers\ en klik “Next”.

Na een ogenblik zou u dit venster moeten zien:



Dit venster is alleen een waarschuwing dat de driver niet is getest door “Microsoft certification”. De drivers geleverd voor de USB module zijn onafhankelijk getest en zouden geen problemen moeten vormen voor uw systeem, mits u ze zelf wijzigt. Klik dus op “Continue Anyway”.

Na een ogenblik zal Windows een boodschap weergaven met “The wizard has finished installing the software for Linx SDM-USB-QS-S” druk op finish.

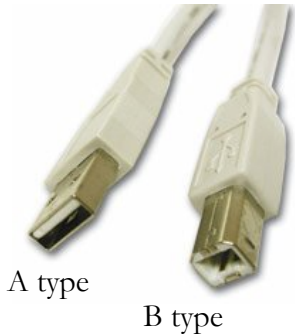
Laat de USB connector en voeding aangesloten, de installatie is nog niet klaar.

Vervolgens zal de computer opnieuw nieuwe hardware detecteren. De hardware die nu wordt gedetecteerd is de Virtuele COM-poort. Om de Virtuele COM-poort te installeren dient u alle voorgaande stappen te herhalen. (begin met lezen vanaf 3.8.3.1 vanaf “Found new hardware wizard”)

De Virtuele COM-poort drivers zijn nu geïnstalleerd.

-> Lees verder in hoofdstuk 3.8.4

3.8.3.2 Installatie procedure USB Device Windows



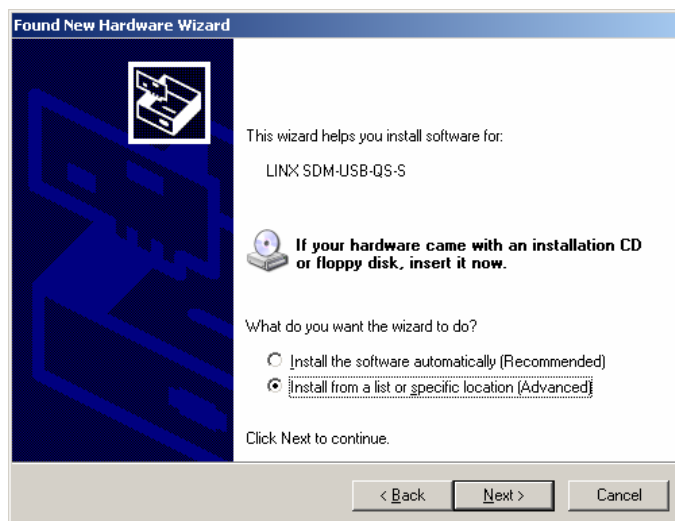
Zorg ervoor dat de voeding is aangesloten op het EDB en verbind een USB-A naar USB-B kabel tussen uw EDB en uw computer.

Na een ogenblik zou u dit venster moeten zien:



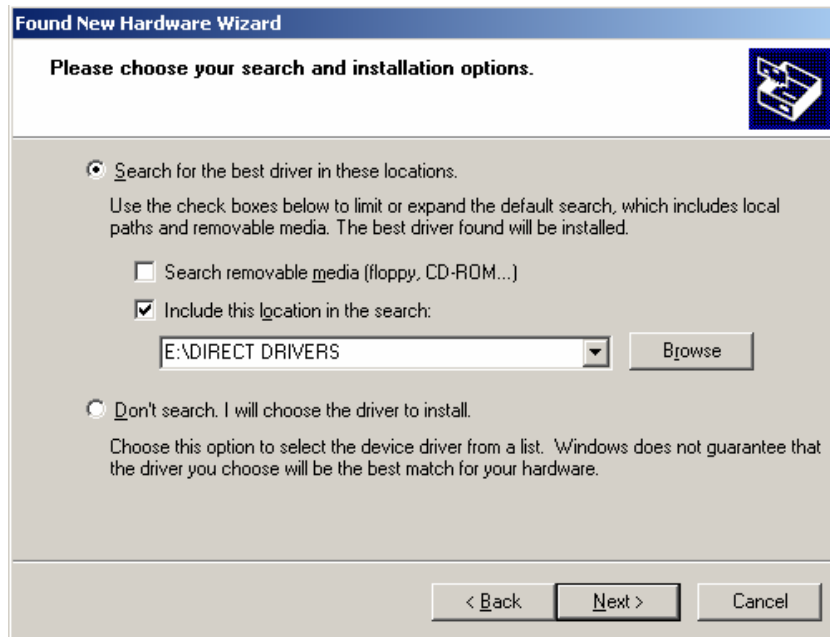
Selecteer “No, not this time” en klik “Next”

Na een ogenblik zou u dit venster moeten zien:



Selecteer “Install from a list or specific location (Advanced)” en klik “Next”

Na een ogenblik zou u dit venster moeten zien:



Plaats de EDB-CD en klik “Browse” om deze map te selecteren: “Uw Cd-rom drive”:\USB_Module\Virtual_COM_Drivers\ en klik “Next”.

***Als u gebruik maakt van uw eigen VID&PID,
dan moet u hier uw eigen driver selecteren.***

Na een ogenblik zou u dit venster moeten zien:



Dit venster is alleen een waarschuwing dat de driver niet is getest door “Microsoft certification”. De drivers geleverd voor de USB module zijn onafhankelijk getest en zouden geen problemen moeten vormen voor uw systeem, mits u ze zelf wijzigt. Klik dus op “Continue Anyway”.

Na een ogenblik zal Windows een boodschap weergaven met “The wizard has finished installing the software for Linx SDM-USB-QS-S” druk op finish.

Het USB apparaat is nu geïnstalleerd.

-> Lees verder in hoofdstuk 3.8.5

3.8.3.3 Driver de-installatie

Als u wilt omschakelen van optie 1 (virtuele COM-poort) naar optie 2, 3. Of als u wilt omschakelen van optie 2, 3 naar optie 1 (virtuele COM-poort), dan moet u eerst de oude driver verwijderen. In dit hoofdstuk wordt de deinstallatie procedure beschreven.

Plaats de EDB-CD in uw computer.

Als u de Virtuele COM-poort driver heeft geïnstalleerd,

Gebruik de bestandsverkenner en navigeer naar uw Cd-rom drive.

Open USB_Module\Virtual_COM_Drivers\FTDIUNIN.EXE

U zou dit venster moeten zien:



Druk op “Continue” en na een ogenblik op “Finish”. Als u wilt kunt u nu de direct driver installeren zie hoofdstuk 3.8.3.2.

Als u de DIRECT driver heeft geïnstalleerd.

Gebruik de bestandsverkenner en navigeer naar uw Cd-rom drive.

Open deze map: “Direct Drivers” en start “FTD2XXUN.EXE”

USB_Module\Direct_Drivers\FTD2XXUN.EXE U zou dit venster moeten zien:



Druk op “Continue” en na een ogenblik op “Finish”. Als u wilt kunt u nu de direct driver installeren zie hoofdstuk 3.8.3.1.

Experiment 20

3.8.4 USB interface, met Virtuele COM-poort

Onderdelen

- 1x USB-A naar USB-B kabel
- 1x Computer met USB poort

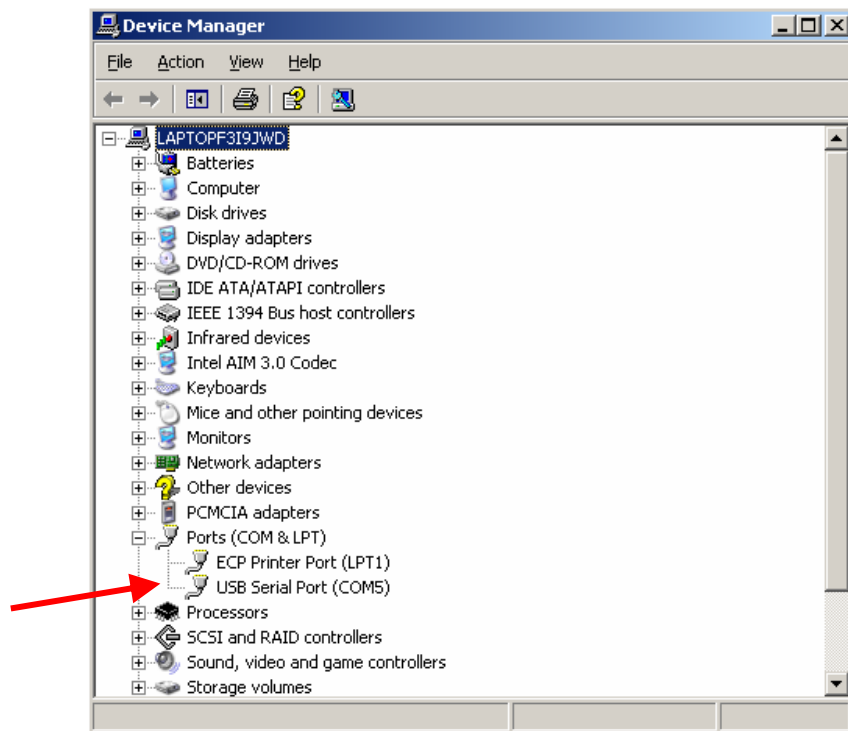
Doel

- Leren hoe u een Virtuele COM-poort kunt gebruiken voor communicatie.

Dit experiment kan alleen worden uitgevoerd als de drivers zijn geïnstalleerd zoals staat beschreven in hoofdstuk 3.8.3.1. **Zorg ervoor dat schakelaar “SI USB/RS232” in de USB stand staat.**

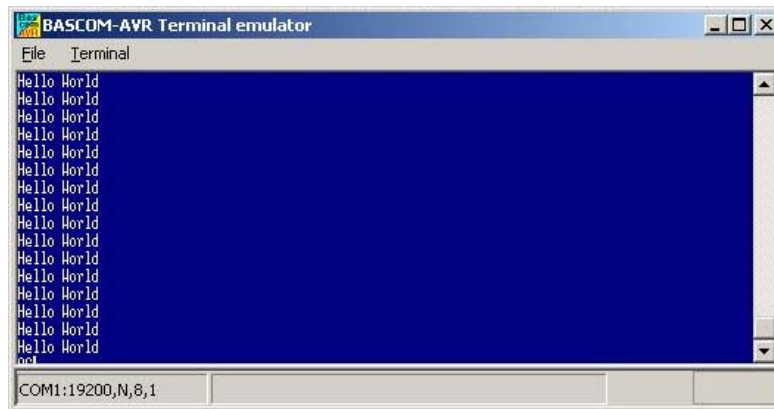
Open het bestand EDBexperiment6a.bas van de EDB-CD en programmeer de ATmega88. We gebruiken hetzelfde programma als in Experiment 6a om de Virtuele COM poort te testen. Nadat u de controller heeft geprogrammeerd en de USB kabel heeft aangesloten, opent u de terminal emulator door op  te klikken in Bascom.

De instellingen zijn onveranderd 19200 baud, no parity, 8 data bits, 1 stop bit. Zorg ervoor dat u de juiste (Virtuele) COM-poort selecteert. U kunt deze vinden in het configuratie scherm. (System, Hardware Tab, en Device Manager)



In dit geval zou u COM5 gebruiken, maar dit kan afwijken van uw systeem

U zou nu het EDBexperiment6a.bas programma “Hello World” moeten zien weergeven op het scherm van de PC.



Het is mogelijk uw eigen applicaties te schrijven voor Windows/Linux enz. Als u Windows gebruikt kunt u hiervoor Visual Basic gebruiken. (Dat is een Basic taal zoals in Bascom AVR). U kunt ook andere talen gebruiken die meerdere operating systems ondersteunen bijvoorbeeld C++ Borland Builder, Pascal of een andere taal die het gebruik van ActiveX componenten toelaat.

Door gebruik te maken van een ActiveX (software-)component kunnen we data zenden en ontvangen via de COM-poort in een Windows/Linux applicatie. U kunt de gratis OCX component (mcscomm.ocx) van MCS Electronics gebruiken. (Er is ook een OCX van Microsoft maar die ondersteund geen events.)

U kunt een voorbeeld Visual Basic programma vinden in de “Visual Basic map op de EDB-CD. In deze map kunt u ook mcscomm.ocx vinden.

Experiment 21

3.8.5 USB interface, device standaard PID&VID

Onderdelen

- 1x USB-A naar USB-B kabel
- 1x Computer met USB poort

Doel

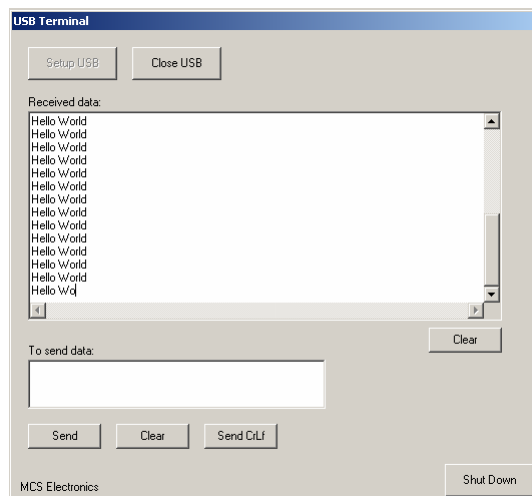
- Een USB interface met standaard PID&VID leren gebruiken

Dit experiment kan alleen worden uitgevoerd als de drivers zijn geïnstalleerd zoals staat beschreven in hoofdstuk 3.8.3.2. **Zorg ervoor dat schakelaar “S1 USB/RS232” in de USB stand staat.**

Open EDBexperiment6a.bas van de EDB-CD en programmeer de ATmega88. We gebruiken hetzelfde programma als in Experiment 6a om de USB interface met standaard PID&VID te testen.

Open het USB testprogramma “USB Terminal” het kan gevonden worden op de EDB-CD map: “Visual_Basic\USB_Terminal”.

Druk op “Setup USB”, als het goed is ziet u nu “Hello World” afgedrukt worden op het scherm van de computer. Probeer ook EDBexperiment6b.bas.



Het is mogelijk u eigen USB applicaties te schrijven voor Windows/Linux enz. Als u Windows gebruikt kunt u daarvoor Visual Basic gebruiken. U kunt ook een andere taal kiezen die meerdere besturingssystemen ondersteund. Bijvoorbeeld C++ Borland Builder, Pascal of een andere taal, zolang DLL componenten worden ondersteund.

We gebruiken een DLL (software-) component om het zenden en ontvangen over de USB-poort door een applicatie in Windows/Linux enz. mogelijk te maken.

De broncode van het bovenstaande programma kunt u vinden in de Visual Basic map op de EDB-CD. Ook de DLL is in die map te vinden. Meer voorbeeld programma's in diverse talen zijn te vinden op de FTDI website:

<http://www.ftdichip.com/Projects/CodeExamples.htm>.

3.8.6 USB interface, device met eigen PID&VID

Dit hoofdstuk is alleen bedoeld voor gevorderde gebruikers, in dit hoofdstuk staan alleen de geavanceerde mogelijkheden beschreven. Het is geen stap-voor-stap handleiding die u precies vertelt welke stappen u moet doen om uw eigen PID USB device te maken.

WAARSCHUWING: Het wijzigen van de EEPROM (PID&VID) van de USB module kan ertoe leiden dat de module niet meer kan worden gebruikt. Het wijzigen van de EEPROM en de drivers is voor eigen risico. MCS Electronics geeft geen support voor problemen veroorzaakt door het wijzigen van de EEPROM of door schade ontstaan door gewijzigde drivers.

U kunt alleen de EEPROM programmeren als u de direct driver heeft geïnstalleerd zoals beschreven in hoofdstuk 3.8.3.2.

Op het EDB wordt de USB gerealiseerd door de SDM-QS-S1-S USB module van Linx technologies, in de module zelf bevindt zich een FT232xx chip van FTDI.

Linx is te vinden op: www.linxtechnologies.com

FTDI is te vinden op: www.ftdichip.com

U kunt lezen en schrijven naar de EEPROM van de USB SDM-QS-S1-S USB module. Dit geeft u de mogelijkheid het VID&PID van de module te veranderen.

Het is sterk aan te raden dat u de bovengenoemde websites bezoekt en dat u de “designers manuals” leest voordat u wijzigingen aanbrengt in de EEPROM. Voor het programmeren van de EEPROM kunt u een tool vinden in de “USB Module” map op de EDB-CD. Er zijn ook tools beschikbaar op de websites van de fabrikanten.

Voordat u, uw USB device met eigen VID&PID kunt gebruiken in Windows/Linux, enz. dient u eerst de *.inf setup informatie aan te passen. U kunt de *.inf bestanden van de EDB-CD als voorbeeld gebruiken.

Online bronnen:

Bestel uw eigen Vendor ID: www.mcselec.com

3.9 Motoren

Afgezien van datacommunicatie en gebruikersinterfaces kunnen microcontrollers ook actuatoren aansturen. De meest gebruikte actuatoren zijn DC en stappenmotoren. Dit hoofdstuk laat zien hoe u gebruik kunt maken van DC en stappenmotoren.

3.9.1 Stappen motor

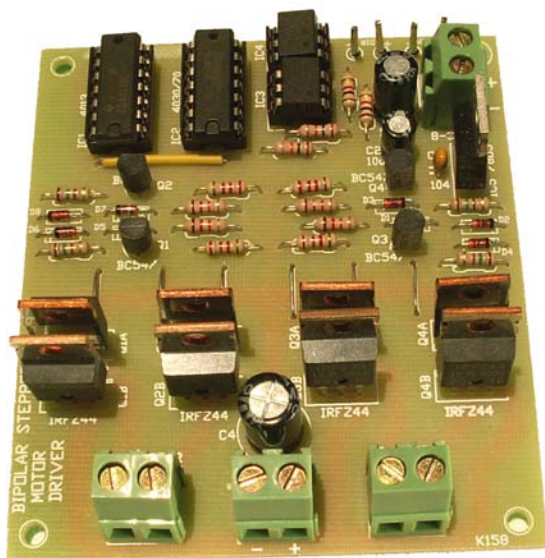
Onderdelen

- 1x Bi-polaire Stappenmotor
- 1x Quasar Stappenmotor Driver

Doel

- Een stappenmotor leren gebruiken

Voor dit experiment gebruiken we Quasar Bipolaire Stappen Motor Driver 3158, meer informatie kan worden gevonden op de Quasar website www.quasarelectronics.com/kit-files/3000/3158.pdf.



Als u geen stappenmotor heeft, kunt u de stappenmotor uit een oude floppy drive gebruiken. Meestal zijn de stappenmotoren uit een diskdrive niet bipolair dus als we deze driver gebruiken kunnen we ze wel gebruiken om mee te experimenteren maar niet om te gebruiken in een echte toepassing.

Verbind de stappenmotor met de “TO MOTOR” connectoren op de PCB van de driver.

Verbind Portd.0 met de “STEP + ” pin,
Verbind Portd.1 met de “DIR +” pin,
Verbind GND van de EDB met “STEP -“ en “DIR-“

Sluit 8...30 V aan op beide power connectoren van de driver. Begin met een lage spanning als u geen specificaties heeft van uw motor. (galvanische isolatie is mogelijk, zie de 3158 pdf voor details.)

Als u een stappenmotor heeft gebruikt uit een floppy drive, moet u mogelijk 3 draden verbinden om de motor aan te sluiten. U moet in dat geval 1 pin van de “TO MOTOR” connector vrij laten.

Programmeer de ATmega88 met EDBexperiment22. Als het goed is ziet u nu de motor van de ene naar de andere kant stappen. Lees de opmerkingen in EDBexperiment22.bas en controleer of u de werking begrijpt.

Experiment 23

3.9.2 De DC motor

Onderdelen

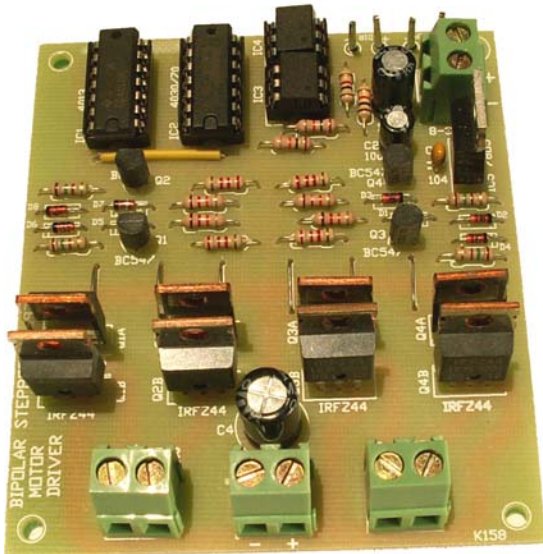
- 1x DC motor
- 1x Quasar Stappenmotor Driver

Doel

- Een DC motor leren gebruiken

Voor dit experiment gebruiken we Quasar Bipolaire Stappen Motor Driver 3158, meer informatie kan worden gevonden op de Quasar website

www.quasarelectronics.com/kit-files/3000/3158.pdf.



Als u geen DC motor heeft kunt u er 1 vinden in een oude cassette speler.

Verbind de DC motor met een van de “TO MOTOR” connectoren op de PCB van de driver.

Verbind Portd.0 met de “STEP +” pin,
Verbind Portd.1 met de “DIR +” pin,
Verbind GND van de EDB met “STEP -” en “DIR-”

Sluit 8...30 V aan op beide power connectoren van de driver. Begin met een lage spanning als u geen specificaties van uw motor heeft. (galvanische isolatie is mogelijk, zie de 3158 pdf voor details.)

Programmeer de ATmega88 met EDBexperiment22. Als het goed is ziet u nu de motor eerst een tijd de ene en dan een tijd de andere kant op draaien.

Lees de opmerkingen in EDBexperiment22.bas en controleer of u de werking begrijpt.

Online bronnen:

Motor driver: http://www.quasarelectronics.com/motor_controllers_drivers.htm

4. Andere programmeer methoden

Dit hoofdstuk legt uit hoe u de ATMega88 kunt programmeren met de parallelle of de USB poort.

U kunt de ATMega88 ook programmeren met de parallelle of de USB poort. Er zijn 3 redenen om dit te doen:

1. U heeft geen seriële poort,
2. U heeft een ongeprogrammeerde of gewiste microcontroller,
3. U wilt het volledige FLASH geheugen programmeren.
Dit laatste kan niet met een boot loader omdat deze zelf ook geheugen gebruikt.

Hoofdstuk 4.3 beschrijft hoe u de bootloader opnieuw kunt te programmeren.

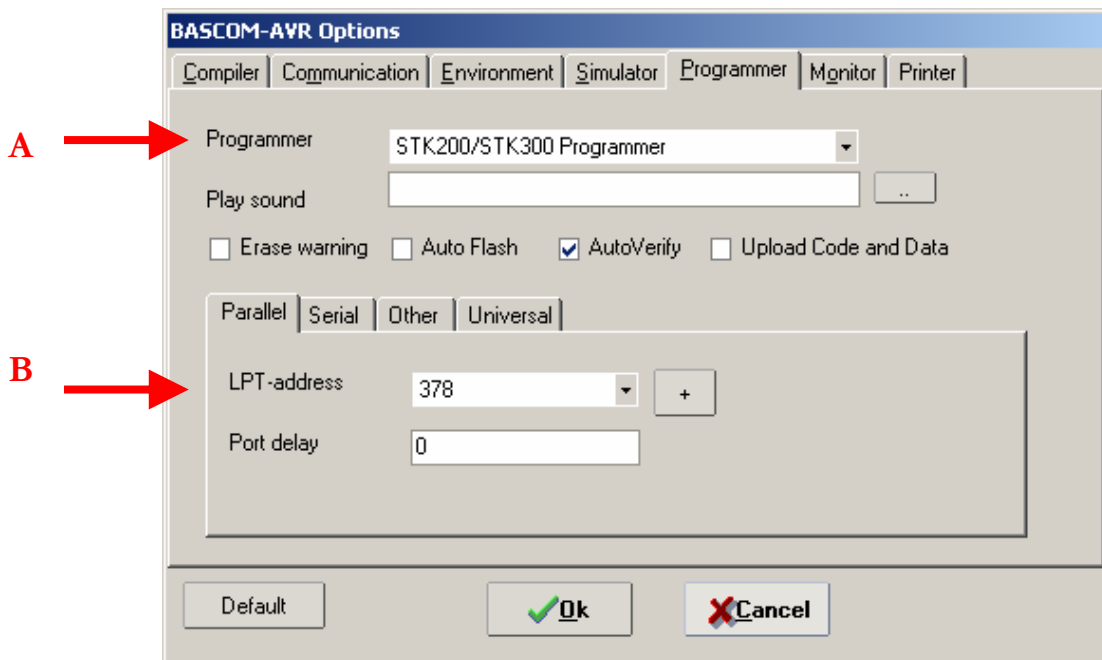
4.1 Parallel programmeren (STK200/300)

U kunt een STK200/300 dongle bestellen bij MCS Electronics, maar u kunt deze ook zelf bouwen. Het schema van de STK dongle is te vinden in bijlage 1.

- Voordat u verdergaat, controleer in de BIOS of uw parallelle poort is ingesteld op ECP. De dongle werkt niet in sommige andere modes zoals de bi-directional mode.

Verbind de STK200/300 dongle met de parallelle poort van de computer en verbind de flat cable van de STK200/300 dongle met connector X9 “ISP” op het Educatief Ontwikkel Bord. (EDB)

Druk in Bascom op “Options” en “Programmer” u ziet nu dit venster:



- Selecteer “STK200/STK300 Programmer” in de “drop down box” (A).
- Selecteer vervolgens uw LPT-adress. Voor LPT1 is dit standaard “378” en voor LPT2 kiest u “278”.
- Druk op de OK-knop.

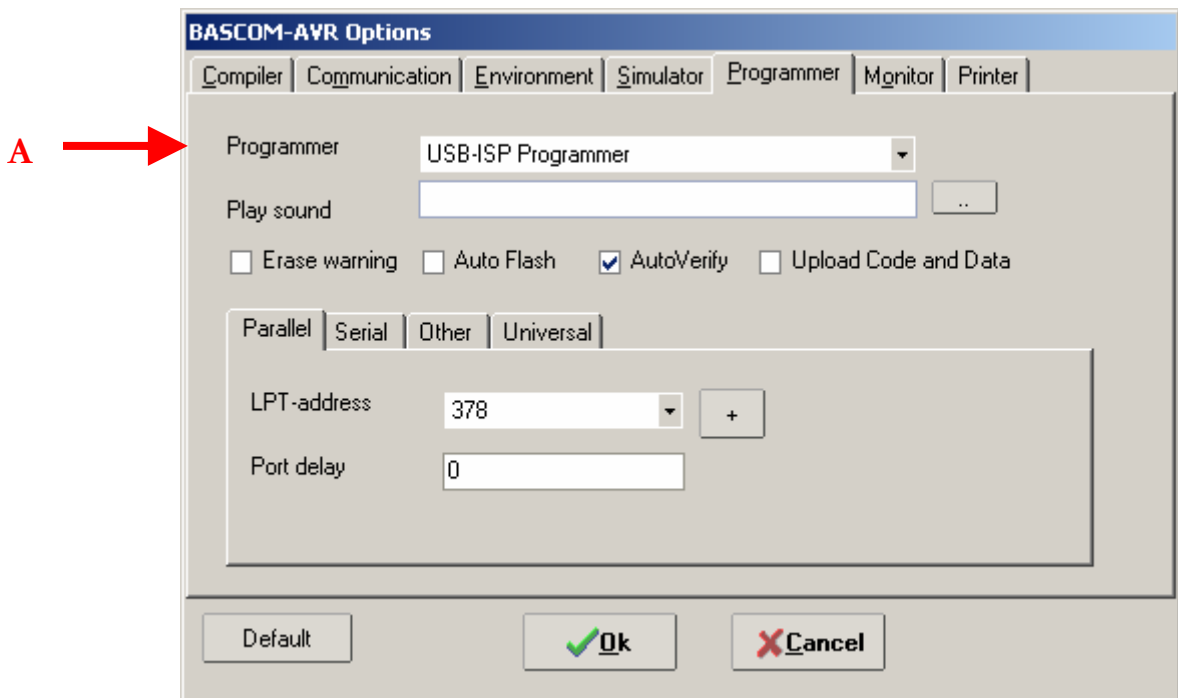
Als u de bootloader wilt gebruiken moet u deze nu programmeren, zie hoofdstuk 4.3. Anders kunt u nu uw programma programmeren zoals beschreven in hoofdstuk 2.3 vanaf de regel “**De flash programmeren**”.

4.2 USB programmeren (Wiazania USP ISP)

U kunt de Wiazania USB ISP programmeer dongle bestellen bij MCS Electronics. De USB ISP programmer heeft dezelfde mogelijkheden als de STK200/300 dongle alleen gebruikt deze de USB poort in plaats van de parallelle poort.

Verbind de USP-ISP programmer met de USB poort van de PC en verbind de flat cable van de STK200/300 dongle met connector X9 “ISP” op het Educatief Ontwikkel Bord. (EDB)

Druk in Bascom op “Options” en “Programmer” u ziet nu dit venster:



- Selecteer “USB-ISP Programmer” ” in de “drop down box” (A).
- Druk nu op de OK-knop.

Als u de bootloader wilt gebruiken moet u deze nu programmeren, zie hoofdstuk 4.3. Anders kunt u nu uw programma programmeren zoals beschreven in hoofdstuk 2.3 vanaf de regel “**De flash programmeren**”.

4.3 De bootloader (opnieuw)programmeren

Gebruik makende van de programmeer methodes van hoofdstuk 4.1 of 4.2.

- Open bootloader.bas in Bascom AVR.
- Compileer de bootloader (door op F7 te drukken)
- Zorg ervoor dat u een van de STK dongles gebruikt die zijn beschreven in hoofdstuk 4.1 en 4.2 en programmeer de chip (F4)
- Programmeer de fuse bits van de ATMega88 zoals hieronder beschreven.
- Nadat u de fusebits heeft geprogrammeerd selecteert u “MCS Bootloader” bij “programmers” zoals beschreven in hoofdstuk 2.3

Programmering van de fuse/lock bits

De ATMega AVR familie maakt gebruik van fuse/lock bits om de geavanceerde instellingen van de chip te configureren. Bij normaal bedrijf worden ze slechts een keer geprogrammeerd, maar ze kunnen meerdere keren worden geprogrammeerd. U kunt de fuse/lock bits programmeren met de ingebouwde programmer van Bascom AVR.

U hoeft de stappen op de volgende pagina alleen uit te voeren als een van deze stellingen voor uw situatie gelden.

1. U gebruikt een ATMega88 die niet is geleverd door MCS Electronics of u heeft de fuse/lock bits van een door MCS geleverde chip gewijzigd.
2. Het lukte niet de chip te programmeren zoals beschreven op de vorige pagina.

U heeft een STK200/300 dongle of een Wiazania USB programmer nodig voor het programmeren van de fuse/lock bits.

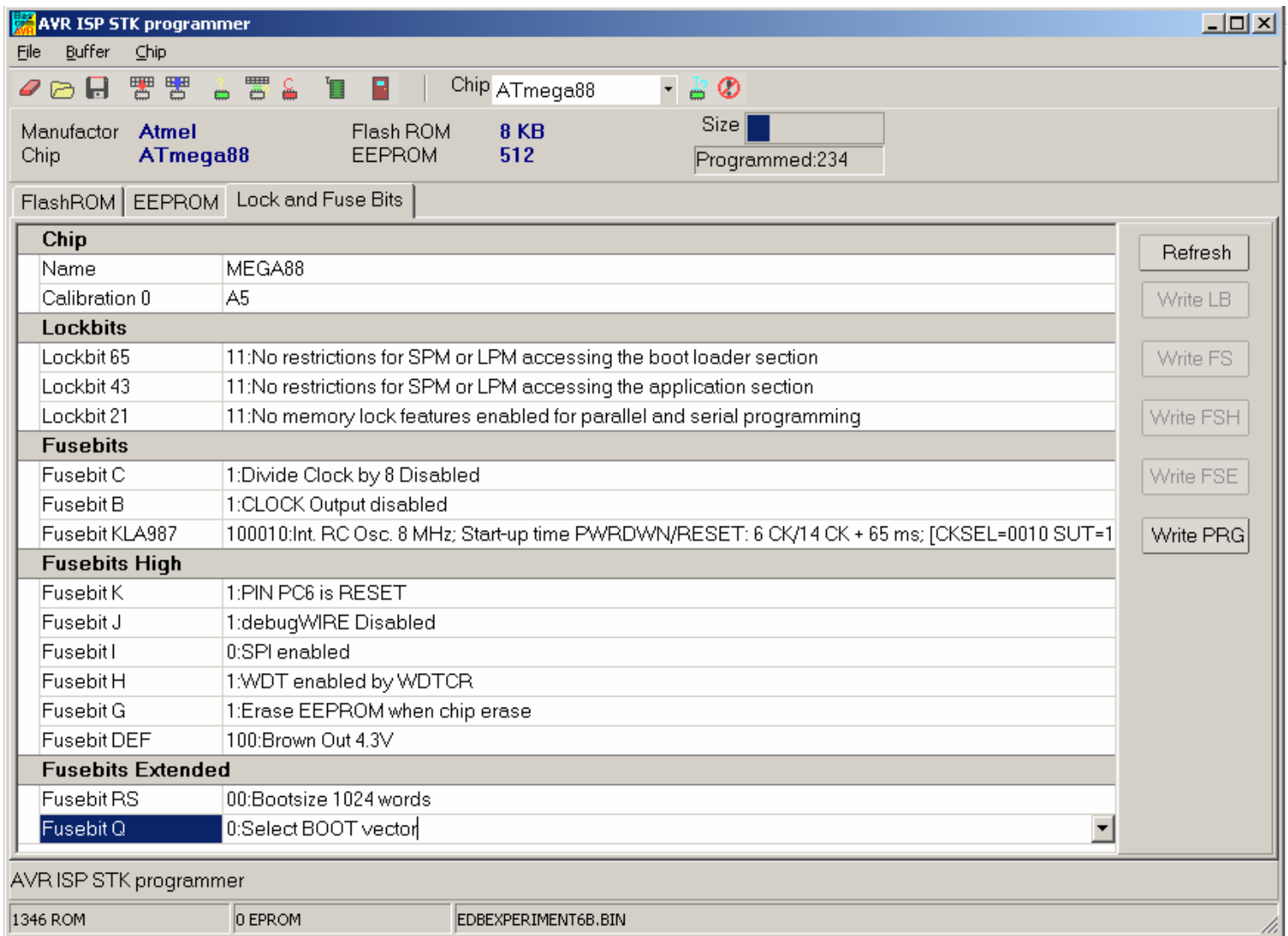
Door het uitvoeren van de volgende stappen, kunt u de ATMega88 juist configureren voor gebruik met het EDB.

Start Bascom AVR, open een *.bas bestand of een nieuw bestand.

Open de programmer op deze manier :



Selecteer het tabblad “Lock and Fuse Bits” en maximaliseer het venster.



Let erop dat de fuse/lock bits zijn geprogrammeerd zoals hierboven is te zien, druk op de “Refresh” knop om de werkelijke instellingen te zien. Fusebit Q moet BOOT zijn.

Waarschuwing Fusebit KLA987 stelt de oscillator opties in, het zou moeten zijn 100010: Int. RC Osc.... SUT = 1. Bij wijziging van deze optie zal de controller niet meer oscilleren zonder externe frequentie oscillator. (frequentie generator). Zorg ervoor dat deze dus op 100010 blijft staan.

U kunt de bits in de controller programmeren door op de “Write XXX” knoppen te drukken die beschikbaar komen als u een optie heeft gewijzigd. (drop down box). Klik hiervoor eerst op een andere fuse/lock bits sectie.

Atmel AVR In System Programming Dongle

Atmel offers a software package called the Atmel AVR ISP which allows the programming of AVR Microcontrollers in circuit with a simple dongle which is attached to the Parallel Port. This dongle is detailed below. It can be built cheaply, making it an ideal starting point for developing with ATMEL AVR micros.

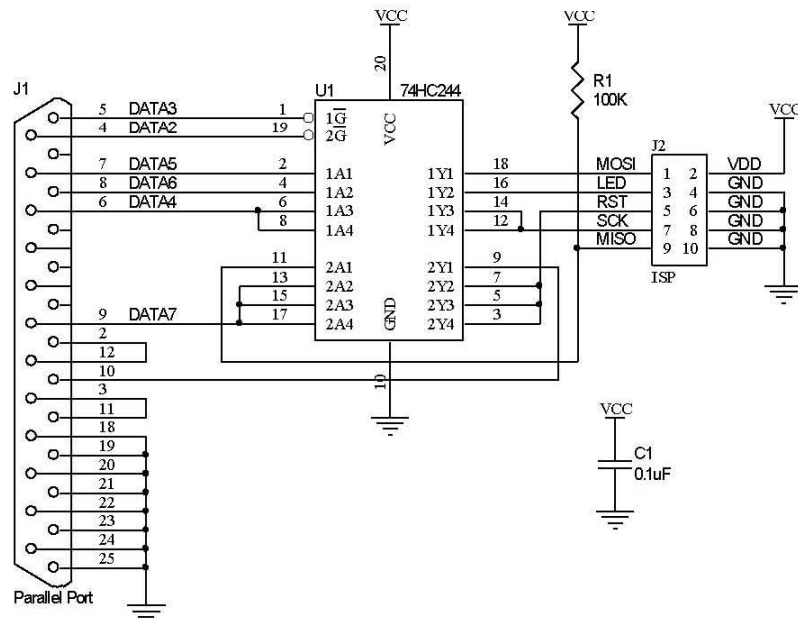


Figure 1 : Schematic for the ATMELE ISP Value Added Dongle

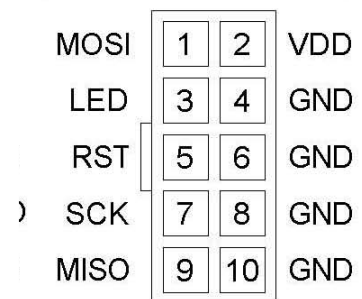
Using a 74HC244 Tri-State buffer as the main component, operation is extremely simple. The two loopback connections, pin 2 to 12 and 3 and 11 is used to identify the dongle. With both links in place the dongle is identified as a Value Added Pack Dongle. With only pins 2 and 12 links, it is reported as a STK300 or AVR ISP Dongle. With only 3 and 11 the dongle is reported as an STK200 or old Kanda ISP Dongle.

DATA2 and DATA3 of the Parallel Port Drive the TriState Outputs. A low will allow the passing of the serial clock and data during programming. MOSI, LED, SCK and Reset being outputs are buffered from the Parallel Port's DATA5, DATA6, DATA4 and DATA7 Respectively. The only input, MISO is fed into nACK, a status input of the Parallel Port.

There are two standard ISP Connectors for Atmel AVR Microcontroller ISP Programming. One standard is the 10 pin version using a DIL 5x5 header of 0.1" Pitch, shown in the above schematic. This is used on the ATMEL STK Kits. The other is a more compact 6 pin version, once again using a DIL 3x3 header of 0.1" pitch. This 6 pin version is the standard connector for ATMEL ISP Programmers.

The main advantage of the 10 pin header is the clean and easy use of 10 pin IDC crimp headers.

Name	Function	Description
MOSI	Master Out - Slave In	Data being transmitted to the part being programmed is sent on this pin
LED	Program LED	Optional Programming LED
RST	Target MCU Reset	Connects to Target AVR. Target AVR is programmed while in Reset State.
SCK	Shift Clock	Serial Clock Generated by the Programmer
MISO	Master In - Slave Out	Data received from the part being programmed is sent on this pin
VCC	ISP Power	Power Supply for the ISP. ISP Header must supply power to the dongle.
GND	Ground	Common Ground



10 Pin Header

Bijlage 2 ASCII Tabel

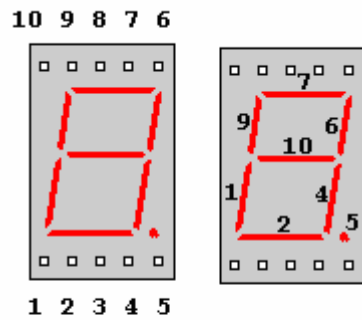
Decimal	Hex	Binary	Value
-----	----	-----	-----
000	000	00000000	NUL (Null char.)
001	001	00000001	SOH (Start of Header)
002	002	00000010	STX (Start of Text)
003	003	00000011	ETX (End of Text)
004	004	00000100	EOT (End of Transmission)
005	005	00000101	ENQ (Enquiry)
006	006	00000110	ACK (Acknowledgment)
007	007	00000111	BEL (Bell)
008	008	00001000	BS (Backspace)
009	009	00001001	HT (Horizontal Tab)
010	00A	00001010	LF (Line Feed)
011	00B	00001011	VT (Vertical Tab)
012	00C	00001100	FF (Form Feed)
013	00D	00001101	CR (Carriage Return)
014	00E	00001110	SO (Shift Out)
015	00F	00001111	SI (Shift In)
016	010	00010000	DLE (Data Link Escape)
017	011	00010001	DC1 (XON) (Device Control 1)
018	012	00010010	DC2 (Device Control 2)
019	013	00010011	DC3 (XOFF) (Device Control 3)
020	014	00010100	DC4 (Device Control 4)
021	015	00010101	NAK (Negative Acknowledgement)
022	016	00010110	SYN (Synchronous Idle)
023	017	00010111	ETB (End of Trans. Block)
024	018	00011000	CAN (Cancel)
025	019	00011001	EM (End of Medium)
026	01A	00011010	SUB (Substitute)
027	01B	00011011	ESC (Escape)
028	01C	00011100	FS (File Separator)
029	01D	00011101	GS (Group Separator)
030	01E	00011110	RS (Request to Send) (Record Separator)
031	01F	00011111	US (Unit Separator)
032	020	00100000	SP (Space)
033	021	00100001	! (exclamation mark)
034	022	00100010	" (double quote)
035	023	00100011	# (number sign)
036	024	00100100	\$ (dollar sign)
037	025	00100101	% (percent)
038	026	00100110	& (ampersand)
039	027	00100111	' (single quote)
040	028	00101000	((left/opening parenthesis)
041	029	00101001	) (right/closing parenthesis)
042	02A	00101010	* (asterisk)
043	02B	00101011	+ (plus)
044	02C	00101100	, (comma)
045	02D	00101101	- (minus or dash)
046	02E	00101110	. (dot)
047	02F	00101111	/ (forward slash)
048	030	00110000	0
049	031	00110001	1
050	032	00110010	2

Decimal -----	Hex ---	Binary -----	Value -----
051	033	00110011	3
052	034	00110100	4
053	035	00110101	5
054	036	00110110	6
055	037	00110111	7
056	038	00111000	8
057	039	00111001	9
058	03A	00111010	:
059	03B	00111011	;
060	03C	00111100	<
061	03D	00111101	=
062	03E	00111110	>
063	03F	00111111	?
064	040	01000000	@
065	041	01000001	A
066	042	01000010	B
067	043	01000011	C
068	044	01000100	D
069	045	01000101	E
070	046	01000110	F
071	047	01000111	G
072	048	01001000	H
073	049	01001001	I
074	04A	01001010	J
075	04B	01001011	K
076	04C	01001100	L
077	04D	01001101	M
078	04E	01001110	N
079	04F	01001111	O
080	050	01010000	P
081	051	01010001	Q
082	052	01010010	R
083	053	01010011	S
084	054	01010100	T
085	055	01010101	U
086	056	01010110	V
087	057	01010111	W
088	058	01011000	X
089	059	01011001	Y
090	05A	01011010	Z
091	05B	01011011	[
092	05C	01011100	\
093	05D	01011101	]
094	05E	01011110	^
095	05F	01011111	_
096	060	01100000	`
097	061	01100001	a
098	062	01100010	b
099	063	01100011	c
100	064	01100100	d
101	065	01100101	e
102	066	01100110	f
103	067	01100111	g
104	068	01101000	h

Decimal -----	Hex ----	Binary -----	Value -----
105	069	01101001	i
106	06A	01101010	j
107	06B	01101011	k
108	06C	01101100	l
109	06D	01101101	m
110	06E	01101110	n
111	06F	01101111	o
112	070	01110000	p
113	071	01110001	q
114	072	01110010	r
115	073	01110011	s
116	074	01110100	t
117	075	01110101	u
118	076	01110110	v
119	077	01110111	w
120	078	01111000	x
121	079	01111001	y
122	07A	01111010	z
123	07B	01111011	{ (left/opening brace)
124	07C	01111100	(vertical bar)
125	07D	01111101	} (right/closing brace)
126	07E	01111110	~ (tilde)
127	07F	01111111	DEL (delete)

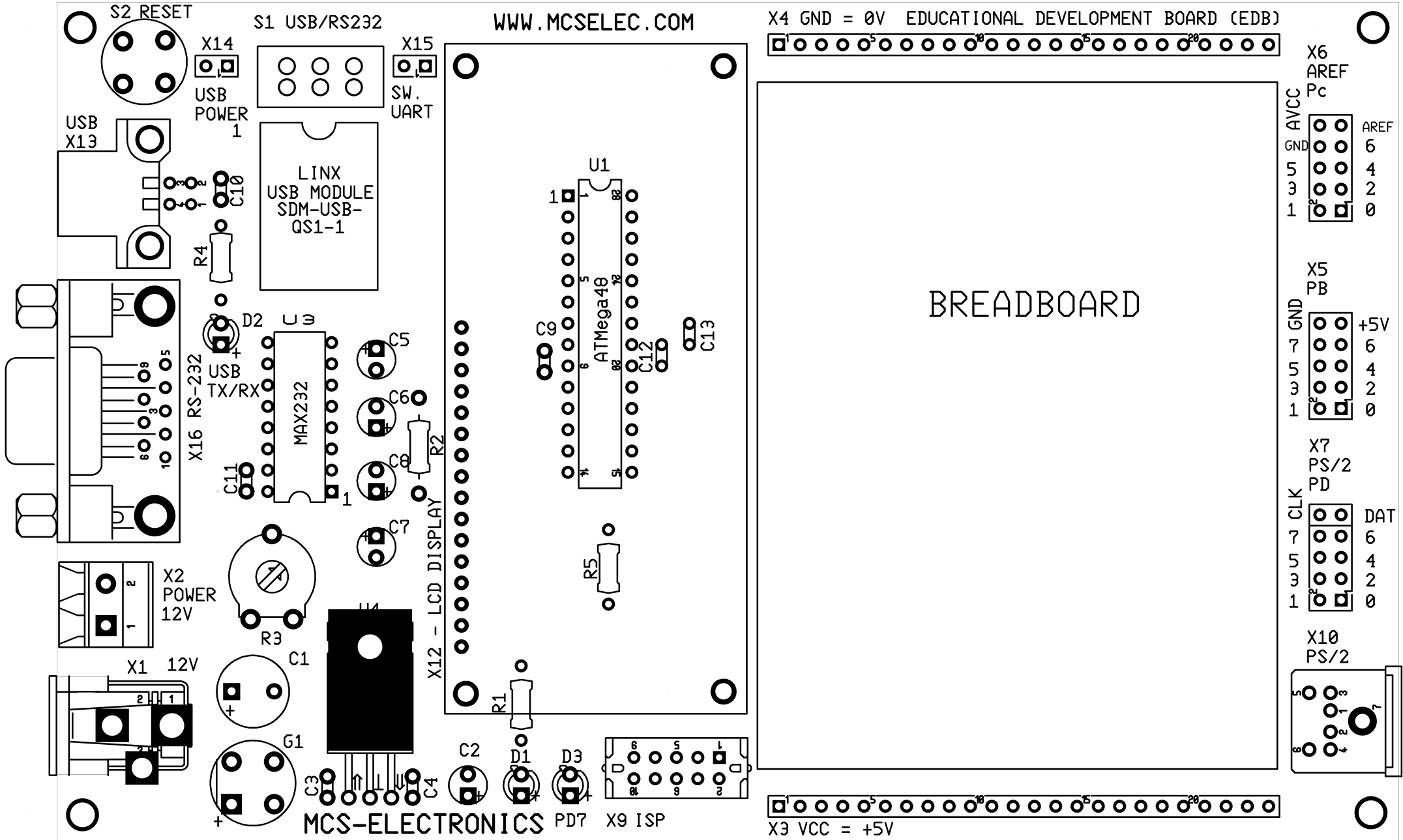
Bijlage 3 7 segment display pin bezetting

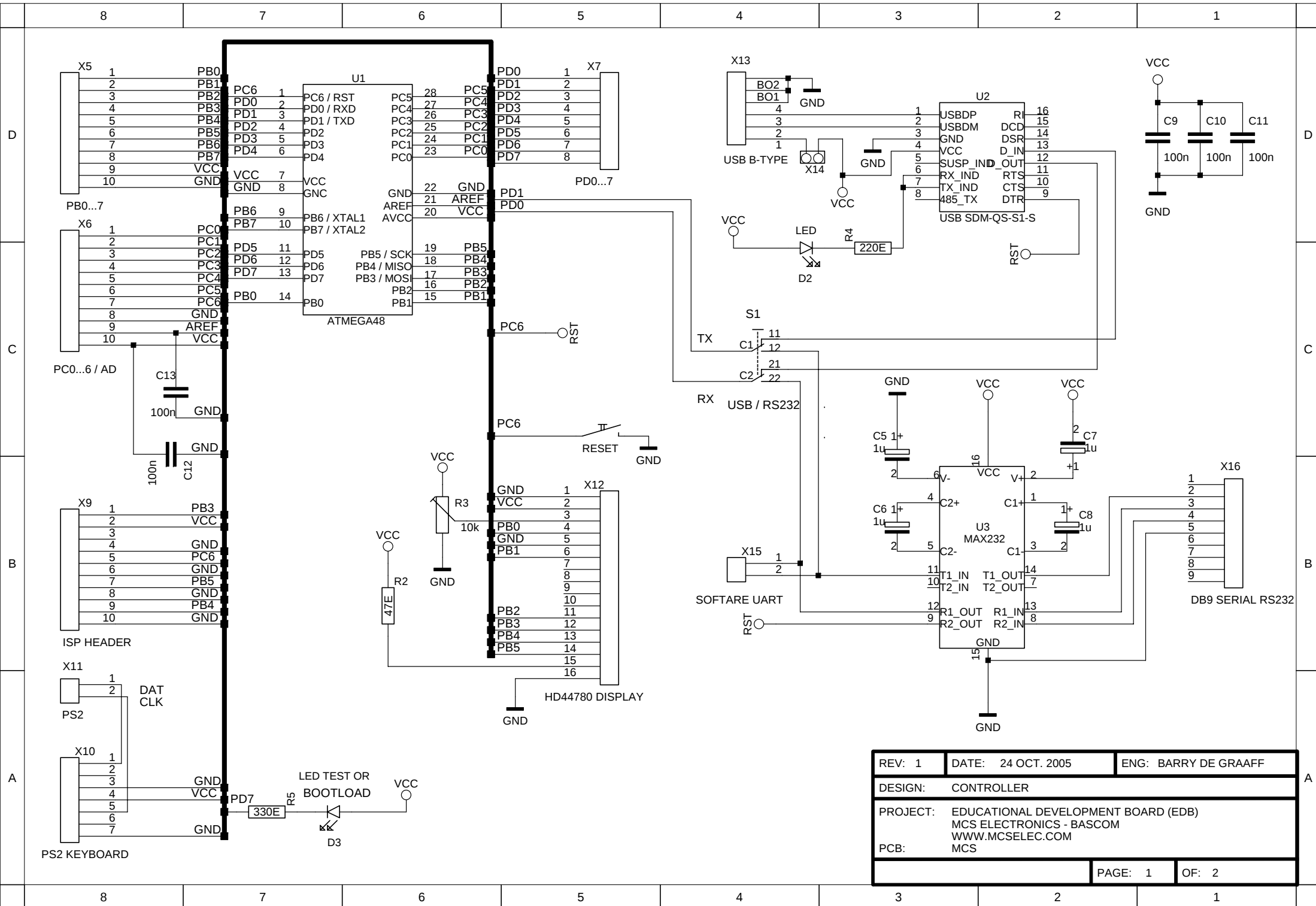
Pin bezetting voor het SL119 / OS516HWA 7 Segment display

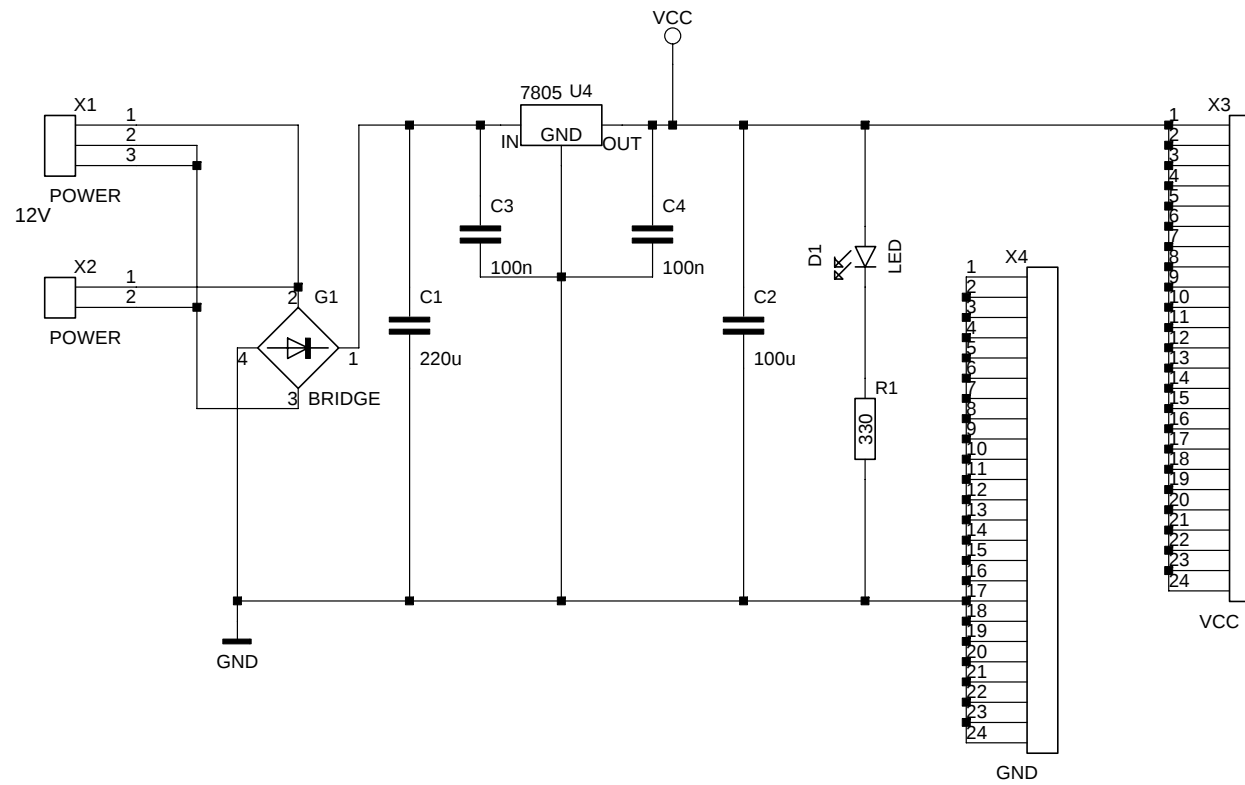


VCC to pin 3 or 8

Bijlage 4 EDB schema's







REV: 1	DATE: 24 OCT. 2005	ENG: BARRY DE GRAAFF
DESIGN:	POWER	
PROJECT:	EDUCATIONAL DEVELOPMENT BOARD (EDB) MCS ELECTRONICS - BASCOM WWW.MCSELEC.COM	
PCB:	MCS	
		PAGE: 2 OF: 2

Copyright Notice

The contents of this document are subject to national and international copyright laws and are the property of MCS Electronics © 2005 except for:

Paragraph 3.1.3a "ATMega88 Datasheet" which is property of Atmel corporation,
Paragraph 3.1.5 "Current Transfer Ratio" which is property of Sharp Semiconductor,
Paragraph 3.4.1 "AN AVR313" which is property of Atmel corporation,
Paragraph 3.4.3 "TSOP Interface" which is property of Vishay Semiconductor,
Annex 1 "AVR ISP Dongle" which is property of Beyondlogic.org.

MCS Electronics gives you permission to reproduce this document for Personal/Educational or Non Commercial use only. In any other case this document may not be reproduced, copied, stored, manipulated in any way, or used whole or in part of a derivative work, without the written permission of MCS Electronics.

Terms and product names used in this document may be trademarks of others.

Disclaimer of warranties and liability

The information in this document is provided "as is" and "where is" without any express or implied warranty of any kind, including, without limitation, warranties of merchantability, title, non infringement of intellectual property rights, or fitness for any particular purpose.

You agree that in no event will MCS Electronics be liable for any damages whatsoever (including, without limitation, damages for loss of profits or business interruption and/or consequential, punitive, exemplary, special or incidental damages) arising out of the use of the information in this document, even if MCS Electronics has been advised of the possibility of such damages.

MCS Electronics is not liable for any damage caused by the use of the information in this document nor is MCS Electronics liable for damage caused by EDB hardware. (including, PCB and components) Use of materials (hardware and software) from MCS Electronics is at your own risk.

MCS Electronics does not warrant the accuracy or completeness of the information, text, graphics, links or other items in this document. MCS Electronics may make changes to such information, text, graphics, links or other items in this document at any time without notice. MCS Electronics makes no commitment to update such information, text, graphics, links or other items.

No rights can be claimed out of the contents of this document.

MCS Electronics © 2005 All rights reserved.