# 'Egel werkboek



<sup>3e druk</sup> © 1998-2004

# HCC Forth gebruikersgroep

# Voorwoord

Dit werkboek is samengesteld door de werkgroep 'Egel ("tick-Egel" of kortweg Egel) van de HCC Forth gebruikersgroep. De HCC Forth gg is een groep binnen de Hobby Computer Club die zich bezig houdt met het programmeren in en het verspreiden van de programmeertaal Forth.

Tijdens de HCC-dagen in 1996 en 1997 was er veel belangstelling voor de toepassingen die door Willem Ouwerkerk gedemonstreerd werden. In 1997 is de 'Egel werkgroep gestart. Gedurende de HCC-dagen in 1997 is een eerste, nog bescheiden, opzet van het werkboek getoond. Op de HCCdagen in 1998 is de eerste druk van het eerste Werkboek uitgebracht.

Het systeem dat hier wordt voorgesteld is geïnspireerd door de Basic Stamp van de Amerikaanse fabrikant Parallax. Een dergelijk project is bij uitstek geschikt voor toepassing van de programmeertaal Forth.

De basis van het systeem wordt gevormd door:

- Het B+-bord, later opgewaardeerd tot het AF-bord, een ontwerp van Bas Boetekees;
- ANS-Forth in EPROM voor het B+/AF-bord, geschreven door Willem Ouwerkerk;
- Server- en Crosscompilersoftware van Willem Ouwerkerk;
- FLAP, FLAsh-Programmer van Willem Ouwerkerk;
- AT51, experimenteerprint van Willem Ouwerkerk;
- AT89Cx051, de kern van het project, een microcontroller van Atmel.

Tijdens de looptijd van het werkboekproject zijn tevens tot stand gekomen:

- F+-bord, vergelijkbaar met B+- en AF-bord, van Paul Wiegmans
- FAT-15V-voeding voor B+- en AF-bord
- RS-FLAP, een flashprogrammer voor de PC-versie van Willem Ouwerkerk
- Pcross, een crosscompiler die zonder het AF-bord werkt, van Willem Ouwerkerk.

De projecten, zowel de hardware als de software, zijn ontwikkeld door Willem Ouwerkerk. De leden van de werkgroep hebben de projecten gebouwd, getest, de documentatie samengesteld, wijzigingen voorgesteld en doorgevoerd en vooral onderling veel commentaar geleverd.

De 'Egel werkgroep is voor informatie en bestellingen bereikbaar via:

Willem Ouwerkerk	e-mail: voorz@forth-gg.hobby.nl
Boulevard Heuvelink 126	
6828 KW Arnhem	
telefoon: 026 44 31 305	Forth BBS: 026 44 22 164

1<sup>e</sup> druk najaar 1998 2<sup>e</sup> druk najaar 1999 3e druk najaar 2004 © HCC Forth gg.

Zie binnenzijde achterflap

# 'Egel werkboek



# © HCC Forth gebruikersgroep

# Werkgroep 'Egel

Bas Boetekees Leendert van den Heuvel Willem Ouwerkerk Paul Wiegmans At van Wijk Dick Willemse

#### Auteursrecht:

De teksten, figuren, schema's en programma's in dit werkboek, m.u.v. de technische publicaties betreffende de producten van verschillende fabrikanten genoemd in de appendices, zijn beschermd eigendom van de HCC Forth Gebruikersgroep. Het geheel of gedeeltelijk overnemen van in dit boek gepubliceerde onderwerpen is slechts toegestaan na uitdrukkelijke schriftelijke toestemming van de HCC Forth Gebruikersgroep.

Voor het verkrijgen van die toestemming kunt u zich wenden tot:

Voorzitter HCC Forth gg W. Ouwerkerk Boulevard Heuvelink 126 6828 KW Arnhem Tel. 026-4431305 e-mail voorz@forth-gg.hobby.nl

De HCC Forth Gebruikersgroep stelt zich niet aansprakelijk voor de gevolgen van eventuele fouten.

# Inhoudsopgave

Voorwoord	2
Inhoudsopgave	5
Lijst met software / hardware	6
Inleiding	7
Algemene aanwijzingen.	9
Testen van je zelfgebouwde systeem	10
Binaire teller	13
Looplicht	16
Looplicht-2	
Analoog / Digitaal-omzetting	21
Lichtmeter	23
Seriële I/O (RS232)	26
Analoge Datalogger	29
Pulsbreedtemodulatie (PBM)	34
Relais-aansturing	37
Servo-aansturing	40
Unipolaire stappenmotor	44
Bipolaire stappenmotor	
Frequentiemeting	
Ontvanger Afstandsbediening	58
Zender Afstandsbediening	63
7-Segmentdisplays	68
I CD-aansturing	72
I <sup>2</sup> C-inleiding	76
	78
1°C 011	
I2C-FEPROM	84
I <sup>2</sup> C-Klok	
Appendix 'Input/Output'	90
Appendix 'Timers/Counters'	94
Annendix 'Serieel / R\$232'	97
Appendix 'Externe Interrunts'	90
Annendix 'Stannenmotoren'	100
Appendix 'BC5'	103
Appendix (LCD)	106
Appendix 'I2C'	108
Appendix 1 C	116
Appendix 'Solution	110
Appendix 'Drogrammeren'	110
Appendix 'Het leden von bestenden'	110
Appendix 'Internet adressen'	119
Appendix 'III C 540'	120
Appendix TLC 549	121
Appendix (DIVI00 50CL)	125
Appendix DUK100-500L	123
Appendix bC517	127
Appendix ULN2005	129
Appendix (SEU504)	131
Appendix SFDJU0	133
Appendix 51015101	133
Appendix L1554/AK	13/
Appendix Schema van EHW 10-D	139
Appendix 1 C-print	140
Appendix Bouwtips	142
Achterilap / Colophon	143

# Lijst met software / hardware

Hardware EHW-01 Software ESW-01	13
Hardware EHW-01 Software ESW-02	16
Hardware EHW-03 Software ESW-03	
Hardware EHW-04	21
Software ESW-04	21
Hardware EHW-05 Software ESW-05	23
Hardware EHW-06 Software ESW-06	
Hardware EHW-07 Software ESW-07	
Hardware EHW-08 Software ESW-08	
Hardware EHW-09 Software ESW-09	
Hardware EHW-10 Software ESW-10	40
Hardware EHW-11 Software ESW-11	44
Hardware EHW-12 Software ESW-12	
Hardware EHW-13 Software ESW-13	
Hardware EHW-14 Software ESW-14	58
Hardware EHW-15 Software ESW-15	63
Hardware EHW-16A EHW-16D Software ESW-16	68
Hardware EHW-17 Software ESW-17	72
Hardware EHW-18A EHW-18I	76
Hardware EHW-18A EHW-18I Software ESW-18	78
Hardware EHW-18A EHW-18I Software ESW-19	81
Hardware EHW-18A EHW-18I Software ESW-20	
Hardware EHW-18I EHW-18A Software ESW-21	

# Inleiding

Het 'Egel-werkboek is opgezet als cursus- en ideeënboek voor het bouwen van hard- en softwaretoepassingen. Het bevat:

- Schema's, PCB-layouts en onderdelenlijsten voor de hardware
- Listings voor de software (ook op flop)
- Appendices met aanvullende informatie

De geteste en gedocumenteerde electronicabouwprojecten zijn bedoeld als een startpunt voor het zelf ontwerpen en bouwen van meet- en regelsystemen, besturingen....enzovoorts.

Het werkboek bevat een aantal elementaire toepassingen, zoals poorten, RS232, I<sup>2</sup>C, ADC, enzovoort. Het is de bedoeling dat je deze als bouwstenen voor je eigen, nog te ontwikkelen ontwerpen gaat gebruiken. De projecten in dit werkboek worden gemaakt met algemeen verkrijgbare onderdelen. De onderwerpen zijn zodanig gekozen en zo in volgorde gezet dat ook beginners, zowel in het ontwikkelen van hardware als in de taal Forth, ervaring kunnen opdoen en hun kennis kunnen uitbreiden.

De eerste 7 projecten vormen de inleiding en worden alle zeven op een enkel experimenteerprintje AT51 gebouwd. De overige projecten kunnen meestal op eenzelfde experimenteerprint worden gebouwd. Voor twee projecten, het 7-segmentdisplay en I<sup>2</sup>C, zijn aparte printplaten ontworpen en leverbaar.

Wat moet je aanschaffen voor de 'Egelprojecten?

Daar gaan we:

- AF-bord met ANS-Forth-EPROM en Crosscompiler.
- FLAP voor het programmeren van een 89Cx051 via het AF-bord.
- AT51-experimenteerprint.
- of
- RS-FLAP voor het programmeren van een 89Cx051 via de COM-poort van uw PC.
- AT51-experimenteerprint.

We kunnen de volgende varianten leveren:

- Het beginnerspakket: AF-bord met ANS-Forth EPROM
- Het pakket zonder controllerbord: als je reeds een AF- of B+-bord bezit
- De PC-versie (met simulator)

Wat heb je nog meer nodig?

Bij elk van deze systemen is een homecomputer met een RS232-aansluiting nodig. De server is geschreven voor de PC maar elk ander systeem met een eigen communicatieprogramma is, weliswaar met enige beperkingen, geschikt. De PC-versie werkt alleen op de 'compatibles'.

Ga je verder met eigen ontwikkelingen dan vind je alle info die daarvoor nodig is in dit werkboek en in het ByteForth Handboek. De gebruikte programmeertaal is Forth. Een spoedcursus Forth is in het handboek opgenomen.

AT89Cx051 staat voor de AT89C2051-microcontroller van de chipproducent Atmel die in dit werkboek gebruikt wordt; ook wel de "2051". Maar ook de AT89C1051 en AT89C4051 zijn bruikbaar. Alle hebben dezelfde 8051-Intel-instructieset en een gelijksoortige geheugenindeling.

De AT89C2051-24PC wordt hier gebruikt met een kristal van 6 MHz. Dit is voor alle toepassingen in dit werkboek snel genoeg: 500.000 instrukties per seconde. Al onze egelsoftware, waarbij interne timers en softwarematige tijdlussen worden gebruikt, houdt rekening met deze kristalfrequentie van 6 MHz. Voor je eigen projecten kan je ook voor een 12-MHz-kristal of zelfs voor een 24-MHz-exemplaar kiezen.

Wat de hardware betreft zijn er twee benaderingen mogelijk:

1. Versie met AF-bord en FLAP

2. PC-versie met RS-FLAP

Het AF-bord is een herziene uitvoering van het B+-bord. Het wordt kant en klaar geleverd, zonder voeding. Het AF-bord wordt gebruikt in combinatie met de FLAP-programmer.

De PC-versie is een programma voor de PC waarmee de projecten ook uitgevoerd kunnen worden. Met de bijbehorende RS-FLAP-programmer worden dezelfde resultaten verkregen als bij de AF-versie: een geprogrammeerde microcontroller die op het AT51-printje zijn taak gaat uitvoeren. De PC-versie heeft tevens een simulator waarmee programma's getest kunnen worden voordat de microcontroller geprogrammeerd wordt.

Voor alle systemen is hetzelfde werkboek van toepassing. Alleen van het ByteForth Handboek bestaan aangepaste versies.

De listings die bij de projecten geleverd worden (ESW-xx), worden op schijf bewaard in de subdirectory 'EGEL. De werkdirectory is de subdirectory WORK. Zelfgemaakte bestanden, zoals de zelf in te typen projecten ESW-1 t/m 4, maak je aan in WORK of je kopieert ze daar naar toe. Geteste bestanden kun je later verplaatsen naar een andere subdirectory met bijvoorbeeld de naam FILES.

# Algemene aanwijzingen.

#### **Beginners**

In het werkboek zijn de eerste zeven instruktieprojecten bedoeld voor de echte beginners: beginners in solderen en/of beginners in Forth.

Diegenen die geen soldeerervaring hebben lezen voordat zij gaan bouwen de appendix 'Solderen'. Daarin zijn enkele practische tips en wat algemene aanwijzingen gegeven.

De eerste zeven projecten moeten in de gegeven volgorde gebouwd worden op een en hetzelfde AT51-printje. Stap voor stap wordt zo de nodige basiskennis en -vaardigheid opgedaan om de volgende projecten zonder al te grote moeilijkheden te kunnen realiseren.

#### Gevorderden

Het is op zijn minst nuttig de eerste zeven projecten toch door te werken om bekend te raken met de gekozen aanpak en met de basisroutines uit de listings. Dezelfde primitieven worden later steeds weer toegepast. Ook met de wijze waarop de programma's zijn opgebouwd raakt men zo gemakkelijk vertrouwd. Bouwen van deze projecten blijft aanbevolen, ook al omdat de gemaakte print later in nog een project gebruikt wordt.

#### Allen

Vanaf project 8 zijn de toepassingen zelfstandig werkend. De meeste passen op een enkele AT51-print. Voor het project Frequentiemeting ESW-13 is het hardware-eindresultaat EHW-06 van de eerste 7 projecten bruikbaar. Voor de projecten 7-Segmentdisplay ESW-16 en I<sup>2</sup>C ESW-18 t/m 22 zijn aparte printplaten ontworpen.

Bovendien wordt sterk aanbevolen om tenminste de eerste vier projecten niet van schijf in te lezen maar om zelf een tekstbestand te maken met een eenvoudige ASCII-editor en dit weg te schrijven naar de directory WORK. Zo lang zijn ze nou ook weer niet. De listings staan ook in de subdirectory EGEL. Op deze manier maak je het best kennis met de programma-opbouw.

#### **Projectnummers**

De projecten gaan vergezeld van twee aanduidingen: EHW-xx : de 'Egel-HardWare: schema en bedradingsplan ESW-xx : de 'Egel-SoftWare: programmalisting

De ESW-nummers worden doorgenummerd; bij elk project hoort andere software en een andere listing. De hardware (schema en bedradingsplan) kan voor een gegeven project hetzelfde zijn als voor een eerder project. Het EHW-nummer is dan hetzelfde als het eerdere project en het is dan dus lager dan het ESW-nummer. In de schema's en bedradingsplannen worden alleen de componenten getekend die nieuw zijn in dat project en die toegevoegd moeten worden aan de minimaal aanwezige componenten van een kant en klaar gebouwde AT51-printplaat. Dit betekent dat de microcontroller en de onderdelen direkt daaraan verbonden, zoals kristal, resetknop e.d., niet in de schema's getekend worden. Zie het ByteForth Handboek.

Blader nu eerst door de appendices en ga dan weer verder vanaf dit punt.

Het ontwikkelsysteem op basis van AF



Testen van je zelfgebouwde systeem

Voor de AF-versie zijn nu beschikbaar:

pcross

· AF-bord met de 8052-ANS-Forth-EPROM met je PC als terminal. Deze wordt gebouwd geleverd.

· Zelf te bouwen FLAP met de bestanden Server/Cross v1.70. Voor de installatie zie ByteForth Handboek v1.70

AT-voet ZIF

AT89C2051

Ο

Voor de PC-versie zijn beschikbaar:

 $\cdot$  Zelf te bouwen RS-FLAP, rechtstreeks via een COM-poort te besturen vanuit je PC met het bestand Cross v1.70pc. Voor de installatie zie ByteForth Handboek v1.70pc of hoger.

Bij het aansluiten van de voedingsspanning op het AF-bord of op de FLAP: <u>GEEN</u> chip in de ZIF-voet!

Bij het laden van CROSS.OVL en compileren: GEEN chip in de ZIF-voet!

De nieuwe samenstelling wordt eerst op interpreter-niveau getest:

1. De PC, het AF-bord, de FLAP (de flashprogrammer, nog zonder de te programmeren microcontroller) en de voeding zijn gereed en op elkaar aangesloten. Het AT51-printje leggen we even opzij. Voor de PC is er de ByteForthsoftware SERVER.EXE- en CROSS.OVL. Het 8052-ANS-Forth zit in de EPROM op het AF-bord.

2. De SERVER wordt opgestart. Een aantal maal <CR> geeft aan of er leven in het systeem zit. Niet? Druk dan op de resetknop van het AF-bord, als die tenminste aanwezig is. Deze knop heeft hetzelfde effect als het intikken van het woord **COLD** en brengt je weer in de opstarttoestand.

Nuttige toetsen zijn <F10>, <F1> en <ESC>. Geldige toetsen: zie de statusbalk onder aan het scherm.

3. <F10>-transmit-DnLdOVL, vraagt om [path]CROSS[.OVL]. <ESC> <ESC> [<CR>] De sneltoets <Shift-F8> doet hetzelfde. Zie het menu onder F1. Cross is nu naar het geheugen van AF gestuurd.

4. We controleren nu de werking en gebruiken daarvoor het woord TEST dat de code op de host executeert.

COLD	\	reset het systeem (of: de resetknop)
ATOM DUP	١	DUP als subroutine (Cross is nog leeg!)
1 TEST DUP		
.5	١	laat zien wat er op de stack staat
atom 2drop	١	MAG DUS OOK IN KLEINE LETTERS

Zolang je de voedingsspanning op AF laat staan kun je de PC uit- en weer aanzetten. De crosscompiler blijft daarbij in het RAM van het AF-bord behouden. Een Nicadbatterij op je AF is ook heel practisch; die levert voldoende energie om de software enige weken tot zelfs maanden in RAM te bewaren, ook als de voeding uitgeschakeld is.



#### **De AT51-experimenteerprint**

De meeste projecten in dit werkboek worden gebouwd op dit experimenteerprintje. Het hieronder afgebeelde exemplaar is voorzien van de draadbruggen die voor de eerste 7 projecten aangebracht moeten worden.



Voor de eerste 7 projecten is het nodig om eerst deze draadbruggen van geïsoleerd draad aan te brengen. Zie ook de appendix 'Solderen'.

Het volgende bedradingsplan (lay-out) bevat het resultaat van de eerste 7 projecten die achtereenvolgens allemaal op een-en-hetzelfde printje gebouwd zullen worden.



# **Binaire teller**

#### in: ( - ) uit: poort-1 (8 LED's)

Acht LED's (Light Emitting Diodes) lichten op in 8-bit binaire code. De waarde neemt steeds met 1 toe. De microcontroller stuurt de LED's op poort-1 geïnverteerd aan (hoog = uit, laag = aan).





Opbouw van 1-bit uitgang





Schema

Snelstartinstructie (1)

De voorgaande test is goed verlopen. Nu is het eerste project (EHW-01) gebouwd op het AT51-printje en dat ligt onder handbereik. We gaan de listing ESW-01 van project-1 invoeren.

De procedure daartoe is als volgt. Programma's moeten eerst met een ASCII-editor geschreven worden en in de subdirectory WORK opgeslagen. Daarna worden ze van schijf teruggelezen. De microcontroller wordt vervolgens in de FLAP geprogrammeerd en wordt tenslotte overgeplaatst naar het AT51-printje. Tijdens het overbrengen van het programma van de computer naar de FLAP mag de microcontroller NIET in de ZIF-voet zitten! Het AT51-printje met de LED's zal apart gevoed worden door een 9V batterij.

Voor dit eerste project volgt hier een samenvatting van de handleidingen. Later wordt een kortere methode gegeven. Het systeem is opgestart volgens de beschrijving die eerder gegeven is:

1. **COLD** \ reset naar de (koude, lege) begintoestand \ of gebruik, indien aanwezig, de resetknop op de voeding

2. <F10>-DOScmd-DIR-..-editor-bestandsnaam.FRT

Schrijf het programma (een bestand met de extensie .FRT) met de editor. Schrijf het programma weg naar schijf in de subdirectory WORK.

- 3. <F10>-transmit-DnLdASCII-[bestandsnaam] \ Compileer naar AF-bord. Vastloper? <Ctrl-Brk>
- 4. <ESC> <ESC> \ om zonder menu naar het hoofdscherm te gaan <CR> \ om te controleren of alles goed is gegaan.

5. Plaats nu de microcontroller in de ZIF-voet. Let op de plaatsingsrichting.

6.	<b>e</b> <cr></cr>	\ empty: flashgeheugen leegmaken
	p < CR >	\ program: programma overbrengen
	$\mathbf{v} < \!\! \mathbf{CR} \!\! >$	\ verify: overdracht controleren

7. Haal de chip nu uit de programmer, controleer de pennetjes en plaats hem op de AT51-print. Let weer op de richting.

8. Sluit de batterij aan op de AT51 en beoordeel de werking. Bedien ook eens de resetknop op dit printje.

We gaan nu weer terug naar de PC:

9. **COLD** \ Reset het systeem met de resetknop op het AF-bord.

Als het systeem nu vast zit (check <CR>) dan is er een eenvoudige ingreep die vaak werkt:

<F10>-poort-verander COM-poort <CR> en zet de COM-poort weer op de eerste waarde.

# Bedradingsplan



# Onderdelenlijst

Referentie	Omschrijving
D1 t/m D8	Rode LED 2 mm of 2,5 mm
RA1	SIL-9 weerstandarray 8 * 330 Ω

# Bouwaanwijzing

Soldeer eerst de SIL-9 weerstand op de aangegeven plaats op de experimenteerprint. Breng de LED's aan en gebruik de aansluitdraad van de LED als draadbrug naar de SIL.

# Software

De LED's zijn aangesloten op poort-1 (1.0...1.7) van de microcontroller. Bij het opstarten gaan alle acht LED's eerst aan en daarna weer uit. Dit is de "koude start" van het programma. Vervolgens komt het programma in een eindeloze lus. De LED's worden daarin met een cyclustijd van 25 ms aangestuurd. De top van de datastack wordt gebruikt als teller.

```
\ Binaire teller met AT89C2051 op poort-1, codelengte 92 bytes.
```

89 \$9	9C2051 TARGET 90 SFR LEDS		\ Compileer voor AT89C2051 \ Adres van poort-1 voor het aansturen \ van 8 LED's
:	TELLER () SETUP		\ Een binaire teller
	0 TO LEDS 250 MS		\ Alle LED's aan en wacht
	-1 TO LEDS 250 MS		\ Alle LED's uit en wacht
0 BEGIN	0		\ Zet teller op de stack
	BEGIN		
	1 +		\ Verhoog teller op de stack
	DUP		\ Maak kopie
	INVERT TO LEDS		\ Keer om en stuur LED's aan
	25 MS		\ Wacht even
	AGAIN		\ Begin weer opnieuw
	;		
	TELLER RESET-VEC	SET-VECTOR	\ TELLER start op na een reset

# Looplicht

#### in: ( - ) uit: poort-1 (8 LED's)

Van acht LED's op poort-1 wordt steeds een volgende LED aangestuurd. Na de achtste LED (poort-1.7) komt weer de eerste (1.0). Hierdoor ontstaat het effect van een rondlopend licht.

# Schema



# Bedradingsplan



#### Onderdelenlijst

Referentie	Omschrijving
D1 t/m D8	Rode LED 2 mm of 2,5 mm
RA1	SIL-9 weerstandarray 8 * 330 Ω

# Bouwaanwijzing

Soldeer JP1 en JP2 en breng de draadbruggen aan. Soldeer de SIL-9 weerstand op de aangegeven plaats op de experimenteerprint. Breng de LED's aan. Gebruik de aansluitdraden van de LEDs als draadbrug om de weerstandarray aan te sluiten.

# Software

De acht LED's zijn aangesloten op poort-1 van de microcontroller. De 'koude start' is nu in een aparte colondefinitie ondergebracht. Het woord **SETUP** moet altijd aan het begin van de hoofdroutine worden geplaatst om de Forthstack correct te installeren. Binnen de DO-LOOP wordt de waarde van de index I gebruikt om een schuifregister na te bootsen.

\ Looplicht met AT89C2051 op poort-1, codelengte 128 bytes.

```
89C2051 TARGET
                                         \ Compileer voor AT89C2051
$90 SFR LEDS
                                         \ Adres van poort-1 voor het aansturen
                                         \ van 8 LED's
: KNIPPER
                ( -- )
                                         \ Visualiseer opstarten
   0 TO LEDS
                250 MS
                                         \ Alle LED's aan
    -1 TO LEDS 250 MS
                                         \ Alle LED's uit
    ;
: LOOPLICHT
                ( -- )
                                         \ Een looplicht
   SETTIP
                                         \ Installeer Forth stacks
   KNIPPER
   BEGIN
        8 0 DO
                                         \ Doe lus acht maal
            1 I LSHIFT
                                         \ Maak bitpatroon
            INVERT TO LEDS
                                         \ Keer om en stuur de LED's aan
            100 MS
                                         \ Wacht even
        LOOP
   AGATN
                                         \ Begin weer opnieuw
    ;
' LOOPLICHT RESET-VEC SET-VECTOR
                                         \ LOOPLICHT start op na een reset
```

\ Einde

#### Snelstartinstructie (2)

Voor het tweede project wordt hetzelfde AT51-printje als bij het eerste project (EHW-01) gebruikt. Het systeem is opgestart volgens de voorgaande beschrijving. In de FLAP zit weer GEEN microcontroller! We gaan de listing ESW-02 van het tweede project invoeren.

We gebruiken nu de funktietoetsen:

- 1. <CR> COLD \ als eerste altijd nuttig
- 2.  $\langle F3 \rangle \setminus DIR en/of$
- <F4> \ editor
- <F5> \ download programmabestand (of <Shift-F5>)
   plaats de chip (nu pas!)
- 5. **e**  $\mathbf{p} \cdot \mathbf{v} < CR > \ zie \dots \mathbf{e}$  chip eruit, chip erin,  $\mathbf{p} \cdot \mathbf{v} < CR >$
- chip overplaatsen naar de AT51-print en testen.

Looplicht-2	Hard
	Sof

In: poort-3.2...-3.3 (2 schakelaars) Uit: poort-1 (8 LED's)

Met de stand van twee schakelaars is de rondloopsnelheid van het looplicht in te stellen op 4 waarden in stappen van 10 ms. De stand van de twee schakelaars wordt via de laagste twee bits van poort-3 door de microcontroller ingelezen. Als een schakelaar open is dan wordt de ingang van de microcontroller door de interne pullupweerstand naar plus (logisch hoog niveau) getrokken. Zie ook de appendix Input/output.





Ingang met N.O. schakelaar





# Bedradingsplan



# Onderdelenlijst

Referentie	Omschrijving
D1 t/m D8	Rode LED 2 mm of 2,5 mm
RA1	SIL-9 weerstandarray 8 * 330 $\Omega$
JP1, JP2	2-polige contactstrip male
-	Twee jumpers

# Bouwaanwijzing

Soldeer JP1 en JP2 en breng de draadbruggen aan. Soldeer de SIL-9 weerstand op de aangegeven plaats op de experimenteerprint. Breng de LED's aan. Gebruik de aansluitdraden van de LEDs als draadbrug om de weerstandarray aan te sluiten.

#### Software

Alle 8 bits van poort-3 worden tegelijk ingelezen. Bit-2 en -3 worden hieruit afgezonderd en verder in het programma verwerkt.

\ Looplicht met regelbare snelheid op poort-1, codelengte 178 bytes.

```
89C2051 TARGET
                                          \ Compileer voor AT89C2051
$90 SFR LEDS
                                          \ Adres van poort-1 voor het aansturen
                                          \ van 8 LED's
$B0 SFR INGANG
                                          \ Ingang met 2 schakelaars
                                          \  \  \text{op bit-2 en 3}
: 10MS
               (u--)
                                          \ Wacht u x10 millisec.
    ?DUP IF 0 DO 10 MS LOOP THEN
    ;
: WACHT
               ( -- )
                                          \ Wacht 0 tot 120 millisec
                                          \ Schakelaars zitten aan massa!
    INGANG INVERT
    $0C AND
                                          \ Alleen bit-2 en 3 zijn geldig
    10MS
                                          \ Wacht INGANG x10 millisec.
    ;
: NAAR-LEDS
               (x--)
                                          \  \  \mathbf{Zet} \  \mathbf{x} \  \mathbf{op} \  \mathbf{de} \  \mathbf{LED's}
    INVERT TO LEDS
                                          \ Inverteer x en stuur LED's aan
    ;
: KNIPPER
               ( -- )
                                          \ Visualiseer opstarten
    -1 NAAR-LEDS 250 MS
                                          \ Alle LED's aan
    0 NAAR-LEDS 250 MS
                                          \ Alle LED's uit
    ;
: LOOPLICHT ( -- )
                                          \ Een eenvoudig looplicht
    SETUP
                                          \ Installeer Forth stacks
    KNIPPER
    BEGIN
        8 0 DO
                                          \ Doe lus acht maal
            1 I LSHIFT
                                          \ Maak bitpatroon
                                          \ Zet patroon op de LED's
            NAAR-LEDS
            WACHT
                                          \ Wacht even
        LOOP
    AGAIN
                                          \ Begin weer opnieuw
    ;
' LOOPLICHT RESET-VEC SET-VECTOR
                                          \ LOOPLICHT start op na een reset
```

```
\ Einde
```

```
Algemene programma-struktuur

89C2051 TARGET \ 89C2051 is default en mag dus worden weggelaten

$90 SFR pnaam \ poort-1 voor de LED's

.... \ variabelen e.d.

: naaml ...; \ de colondefinities

.... \ enz.

: naam

SETUP \ in hoofdroutine

....;

' naam RESET-VEC SET-VECTOR \ opstarten programma
```

# Analoog / Digitaal-omzetting

in: P3.5 (ADC data) uit: poort-1 (8 LED's); P3.4 (ADC-klok); P3.7 (ADC-select)

De spanning van een potentiometer wordt door een analoog/digitaalomzetter in een 8-bitwaarde omgezet. Deze waarde wordt serieel door de microcontroller ingelezen en in binaire code op de 8 LED's weergegeven.

# Schema



Bedradingsplan



#### Onderdelenlijst

Referentie	Omschrijving
D1	BAT85 diode
R1	100 kΩ
P1	1 k $\Omega$ potmeter
J1	3-polige contactstrip male
J2	3-polige contaststrip female
U1	TLC549IP
-	DIL-8 IC-voet

#### Bouwaanwijzing

Soldeer eerst de weerstand en de diode, daarna de DIL-8 IC-voet op de plaats van U1 en soldeer de draadbruggen en aansluitdraden. Breng de potmeter van 1 k $\Omega$  aan volgens het bedradingsplan.

## Software

De 8-bit uitvoer van de ADC wordt bit-voor-bit ingelezen in de microcontroller. De communicatie tussen de microcontroller en de ADC verloopt via bits 4, 5 en 7 van poort-3. Deze worden individueel als bit-SFR aangesproken voor in- en uitvoer van de ADC.

\ Analoog naar digitaal omzetting met de TLC549IP. Lengte 201 bytes.

```
89C2051 TARGET
                                       \ Compileer voor AT89C2051
$90 SFR LEDS
                                       \ Adres van poort-1 voor het aansturen
                                       \ van 8 LED's
$B4 BIT-SFR ADC-KLOK
                                       \ Klokpuls-invoer P3.4
$B5 BIT-SFR ADC-DATA
                                       \ Data-uitvoer P3.5
$B7 BIT-SFR ADC-SELECTEER
                                       \ Activeer invoer P3.7
: ADC
                   ( -- u )
                                       \ Doe een AD-omzetting
   FALSE TO ADC-KLOK
                                       \ Stop lezen van analoge invoer
   TRUE TO ADC-SELECTEER
                                       \ Start AD-omzetting
    4 0 DO LOOP
                                       \ Conversie duurt min. 17 us
   FALSE TO ADC-SELECTEER
                                       \ Start uitlezen
   ADC-DATA 1 AND
                                       \ Lees databit-7
   7 0 DO
                                       \ Lees resterende 7 databits
       TRUE TO ADC-KLOK
       FALSE TO ADC-KLOK
                                       \ Schuif volgende bit naar buiten
       2 * ADC-DATA 1 AND OR
                                       \ Schaal data in en lees volgende bit
   LOOP
   TRUE TO ADC-KLOK
                                       \ Blijf analoge invoer lezen
    ;
              (u--)
: NAAR-LEDS
                                       \ Toon data binair
   INVERT TO LEDS
    ;
: KNIPPER
               ( -- )
                                       \ Visualiseer opstarten
   -1 NAAR-LEDS 250 MS
                                       \ Alle LED's aan
   0 NAAR-LEDS 250 MS
                                       \ Alle LED's uit
    ;
: TOON-ADC ( -- )
                                       \ Toon omzetting op LED's
   SETUP KNIPPER
   BEGIN
       ADC NAAR-LEDS
                                        \ Lees ADC en toon uitkomst
   AGAIN
 ;
' TOON-ADC
            RESET-VEC SET-VECTOR
                                       \ Installeer ADC omzetter
\ Einde
```

# Lichtmeter

#### in: P3.5 (ADC-data)

uit: Poort-1 (LED's); P3.4 (ADC-klok); P3.7 (ADC-select)

De potmeter uit project EHW-04 vormt een spanningsdeler die kan worden vervangen door een serieschakeling van een weerstand en een sensor (LDR, NTC e.d.). Dan moet wel rekening worden gehouden met het weerstandsbereik. Bedenk dat de sensor waarschijnlijk niet lineair is.

De 8-bit waarden van de ADC kunnen door de LED's op twee verschillende manieren worden weergegeven: als 'VU-schaal' (enkele punt) of als 'thermometerschaal' (lichtstrook). De keuze moet vóór het invoeren of compileren van de broncode gemaakt worden: de keuze wordt gemaakt door de backslash in **BEELD-AF** al of niet te verwijderen.

#### Schema





# Bedradingsplan



# Onderdelenlijst

Dit project is bijna hetzelfde als EHW-04 uit het vorige hoofdstuk. De potmeter is hier vervangen door R2 en R3.

Referentie	Omschrijving
D1	BAT85 diode
R1	100 kΩ
R2	LDR003 of vergelijkbaar type
R3	4,7 kΩ
J1	3-polige contactstrip male
J2	3-polige contaststrip female
J3	3-polige contactstrip female
U1	TLC549IP
-	DIL-8 IC-voet

# Bouwaanwijzing

Soldeer eerst de weerstand en de diode, daarna de DIL-8 IC-voet op de plaats van U1 en soldeer de draadbruggen en aansluitdraden. Breng de serieschakeling van R2 en de LDR aan en soldeer deze aan connector J2.

# Software

De manier waarop de LED's de uitvoer weergeven kan in het programma veranderd worden. In de listing staat de regel voor de 'thermometerschaal' achter een backslash. Tekst na een backslash wordt tot het regeleinde als commentaar gezien. Dergelijke regels worden dus noch gecompileerd, noch geëxecuteerd. Om de 'thermometerschaal' te kunnen gebruiken wordt in de routine **BEELD-AF** de backslash voor de regel **DUP 1- OR** verwijderd. Als je liever de 'VU-schaal' gebruikt dan kun je de regel met de backslash ervoor ook helemaal weghalen, maar dan wordt het lastiger om de zaak later weer terug te draaien.

Let op het gebruik van CLEAR en SET !!!

\ Analoog naar digitaal omzetting met de TLC549IP. Codelengte 261 bytes. \ Compileer voor AT89C2051 89C2051 TARGET \$90 SFR LEDS \ Adres van poort-1 voor het aansturen \ van 8 LED's \$B3 BIT-SFR ADC-KLOK \ Klokpuls-invoer op P3.4 \$B4 BIT-SFR ADC-DATA \ Data-uitvoer P3.5 \$B5 BIT-SFR ADC-SELECTEER \ Activeer invoer P3.7 : ADC ( -- u ) \ Doe een AD-omzetting CLEAR ADC-KLOK \ Stop lezen van analoge invoer SET ADC-SELECTEER \ Start AD-omzetting 4 0 DO LOOP \ Conversie duurt min. 17 us CLEAR ADC-SELECTEER \ Start uitlezen ADC-DATA 1 AND \ Lees databit-7 7 0 DO \ Lees resterende 7 databits SET ADC-KLOK CLEAR ADC-KLOK \ Schuif volgende bit naar buiten 2\* ADC-DATA 1 AND OR \ Schaal data in, lees volgende bit LOOP SET ADC-KLOK \ Blijf analoge invoer lezen ; : NAAR-LEDS (u--) \ Toon data binair INVERT TO LEDS ; : KNIPPER ( -- ) \ Visualiseer opstarten -1 NAAR-LEDS 250 MS 0 NAAR-LEDS 250 MS ; : BEELD-AF (u--) \ Toon met VU- of thermometerschaal 0 32 UM/MOD \ Schaal waarde in voor 8 LED's SWAP 15 > IF 1+ THEN \ Rond af naar hogere waarde DUP IF \ Uitkomst groter dan nul? 1 SWAP 1- LSHIFT \ Ja, bepaal positie van LED \ Vul LED's daaronder in \ DUP 1- OR THEN NAAR-LEDS \ Stuur LED's aan ; : TOON-ADC ( -- ) \ Toon AD-omzetting op LED's SETUP \ Installeer Forthstacks KNIPPER ADC DROP \ Eerste AD-omzetting is ongeldig BEGIN ADC BEELD-AF \ Lees ADC en toon uitkomst AGAIN ; ' TOON-ADC RESET-VEC SET-VECTOR \ TOON-ADC start op na een reset

\ Einde

# Seriële I/O (RS232)

in: P3.0 (RS232-in RX) uit: Poort-1 (8 LED's); P3.1 (RS232-uit TX)

De microcontroller wordt via een RS232-verbinding met J5 op de seriële poort van een computer aangesloten. Op die computer draait een communicatieprogramma naar keuze van de gebruiker. Er is nu communicatie tussen het AT51-printje en de computer mogelijk. De communicatiesnelheid is 2400 baud wanneer een kristal van 6 MHz is geplaatst.

#### Schema



## Bedradingsplan



# Onderdelenlijst

Referentie	Omschrijving
C1, C2, C3, C4	Elco 10 µF - 16 Volt
J1	3-polige contactstrip female
J2	9-polige D-connector female
J3	3-polige contactstrip male
U1	MAX232 buffer/omzetter
-	DIL-16 IC-voet

# Bouwaanwijzing

Soldeer de draadbruggen en de connector J3 op de print, breng daarna de IC-voet aan op de plaats van U1. Tenslotte C1 t/m C4 monteren; let op de + en –. Sluit J1 en J2 aan zoals in het bedradingsplan is aangegeven.

#### Software

De toetsenbord-invoer op de computer wordt door de microcontroller verwerkt. Elke ingetypte letter wordt op de LED's binair afgebeeld en een bericht (een string) wordt via de RS232-verbinding naar de computer terug-gestuurd en als tekst op het scherm afgebeeld.

Het programma is geheel in high-level ByteForth geschreven. De baudrate wordt ingesteld op 2400 baud.

\ RS232-in- en uitvoer, met als extra uitvoer naar LED's. Codelengte 329 bytes.

89C2051 TARGET	\ Compileer voor AT89C2051
\$90 SFR LEDS	\ Adres van poort-1 voor het aansturen
	\ van 8 LED's
\$99 BIT-SFR EMIT?	\ RS232-zendbuffervlag; true indien leeg
\$98 BIT-SFR KEY?	\ RS232-ontvangbuffervlag; true indien vol
\$8E BIT-SFR TIMER1	\ Timer-1 start/stop
\$AC BIT-SFR RS232-INTERRUPT	\ RS232-interrupt aan/uit
\ Tussen haakjes zijn de MCS51-benaminge	en van de nu volgende adressen gegeven
\$89 ( TMOD ) SFR TIMER-MODE	\ Timer/Counter Mode Control Register
\$87 ( PCON ) SFR POWER-MODE	\ RS232-snelheidsverdubbeling
\$98 ( SCON ) SFR RS232-MODE	\ RS232-instelling
\$8D ( TH1 ) SFR TIMER1-HIGH	\ Hoge byte van timer-1
\$99 ( SBUF ) SFR RS232-BUFFER	\ RS232-in/uitvoerbuffer
: RS232-SETUP ( )	\ Installeer RS232
TIMER-MODE \$20 OR TO TIMER-MODE	\ Timer-1 in mode-2 = 8-bit autoreload
243 TO TIMER1-HIGH	\ Selecteer 1200/2400 baud (decimaal)
TRUE TO TIMER1	\ Start timer-1
POWER-MODE \$80 OR TO POWER-MODE	Verdubbel RS232 (2400 baud)
\$52 TO RS232-MODE	\ Seriële mode-1, ontvangst aan
	\ Zet zendinterruptylag
FALSE TO RS232-INTERRUPT	\ Zet RS232-interrupt uit
;	
: EMIT ( char )	\ Stuur karakter via RS232
BEGIN EMIT? UNTIL	\ Wacht tot de zendbuffer is geleegd
TO RS232-BUFFER	\ Zet nieuw karakter in de zendbuffer
FALSE TO EMIT?	\ De zendbuffer is gevuld
•	, to tonewarter ip govard
/	

```
: TYPE
                  ( $addr u -- )
                                       \ Stuur ROM-string via RS232
    ?DUP IF
                                       \ Druk de string alleen af als u>0
      0 DO
                                       \ Doorloop deze lus u maal
           COUNT EMIT
                                       \ Druk karakter af en verhoog adrespointer
       LOOP
                                       \ Volgende karakter
    THEN
    2DROP
                                       \ Verwijder adres stringeinde van de stack
    ;
: KEY
                  ( -- char )
                                       \ Ontvang karakter via RS232
   BEGIN KEY? UNTIL
                                       \ Wacht tot karakter ontvangen is
   RS232-BUFFER
                                       \ Lees het karakter uit de buffer en zet
                                       \ het op de stack
   FALSE TO KEY?
                                       \ Geef aan dat de buffer gelezen is
    ;
\ Een klein RS232-tekstin- en uitvoervoorbeeld
ATOM INLINE$
                  ( -- $addr u )
                                       \ Importeer extra macro om ." te kunnen
                                       \ gebruiken
           ^M EMIT ^J EMIT ;
: CR
                                       \ Nieuwe regel
          ^L EMIT ;
                                       \ Maak beeldscherm schoon
: PAGE
: NAAR-LEDS
                  ( x -- )
                                       \ Verstuur de data naar de LED's
   INVERT TO LEDS
                                       \ Inverteer de bits, stuur de LED's aan
    ;
: BEHANDEL-INVOER ( -- )
                                       \ Lees keyboard en verstuur een tekst
   BEGIN
       CR ." > " KEY DUP NAAR-LEDS
                                       \ Copieer karakter op LED's
       DUP &A = IF ." Aap " THEN
DUP &B = IF ." Boef " THEN
                                       \ Verzend 'Aap' via RS232 als key='A'
                                       \ Etc.
       DUP &C = IF ." Chip " THEN
    &Q = UNTIL
                                       \ Verlaat de lus als key='Q'
    ;
: TEKST-I/O ( -- )
                                       \ In- en uitvoerproef voor RS232-tekst
   SETUP
                                       \ Installeer Forthstacks
   RS232-SETUP
                                       \ Installeer RS232
   BEGIN
       PAGE ." RS232 I/O-proef voor AT51-printje"
       BEHANDEL-INVOER
                                       \ Lees keyboard, verstuur tekst
   AGAIN
                                       \ Eindloze lus vanaf 'BEGIN'
    ;
' TEKST-I/O RESET-VEC SET-VECTOR \ TEKST-I/O start op na een reset
```

```
\ Einde
```

# Analoge Datalogger

in: P3.0 (RX); P3.5 (ADC-data) uit: Poort-1 (8 LED's); P3.1 (TX); P3.4 (ADC-klok); P3.7 (ADC-select)

De waarden aan de uitgang van een ADC worden door de microcontroller gelezen en verzameld in een tabel. Vervolgens worden deze gegevens in blokken via een RS232-verbinding aan een computer doorgegeven. De waarden worden na iedere meting ook door de 8 LED's weergegeven.

## Schema



# Bedradingsplan



# Onderdelenlijst

In all project worden EHw-03, EHw-04 en EHw-06 gecombineer			
Referentie	Omschrijving		
C1, C2, C3, C4	Elco 10 µF - 16 Volt		
D1 t/m D8	Rode LED 2 mm of 2,5 mm		
D9	BAT85 diode		
J1,J4	3-polige contactstrip male		
J2,J3	3-polige contactstrip female		
J5	9-polige D-connector female		
R1	100 kΩ		
RA1	SIL-9 weerstandarray 8 * 330 Ω		
P1	1 kΩ		
U1	TLC549IP		
U2	MAX232 buffer/omzetter		
-	DIL-8 IC-voet		
-	DIL-16 IC-voet		

#### In dit project worden EHW-03, EHW-04 en EHW-06 gecombineerd

# Bouwaanwijzing

Soldeer J1 en J4 en breng de draadbruggen aan, breng daarna de IC-voeten aan op de plaats van U1 en U2. Soldeer weerstand R1, de weerstandarray RA1 en de diode D9. Soldeer de LEDs D1 t/m D8. Monteer dan C1 t/m C4; let goed op de + en - aansluitingen. Breng de potmeter van 1 k $\Omega$  aan en soldeer aan de connector J2 volgens het bedradingsplan. Plaats de ICs U1 en U2 in de respectievelijke IC-voeten. Sluit J3 en J5 aan zoals in het bedradingsplan is aangegeven.

#### Software

De baudrate is ingesteld op 2400 baud. Ook dit programma is voor een groot deel opgebouwd uit bibliotheekfunkties.

Vergelijk het programma met dat van de projecten ESW-05 en ESW-06. Zoek in de bibliotheek de gebruikte functies op. Het is mogelijk voor een eigen programma de routines uit de bibliotheek over te nemen en zonodig te veranderen. Laat echter de routines die in LIB staan ongewijzigd!!

```
\ Analoge datalogger met uitvoer naar RS232 en LED's. Lengte 698 bytes.
89C2051 TARGET
                                        \ Compileer voor AT89C2051
\ ByteForth RS232-woorden
$99 BIT-SFR RS232-EMIT?
                                         \ RS232-zendbuffer leeg ?
$8E BIT-SFR TIMER1
                                         \ Timer-1 start/stop
$AC BIT-SFR RS232-INTERRUPT
                                         \ Timer-1-interrupt aan/uit
$89 ( TMOD ) SFR TIMER-MODE
                                         \ Timer/Counter Mode Control Register
                                   \ Timer/Counter Mode Contr
\ RS232-snelheidsverdubbel:
\ RS232-instelling
\ De hoge byte van timer-1
\ RS232-in/uitvoerbuffer
$87 ( PCON ) SFR POWER-MODE
                                         \ RS232-snelheidsverdubbeling
$98 ( SCON ) SFR RS232-MODE
$70 ( SCON ) SFR RS232-MODE
$8D ( TH1 ) SFR TIMER1-HIGH
$99 ( SBUF ) SFR RS232-BUFFER
                                        \ RS232-in/uitvoerbuffer
: RS232-SETUP
    TIMER-MODE $20 OR TO TIMER-MODE
                                        \ Timer-1 mode-2 = 8-bit autoreload
    243 TO TIMER1-HIGH
                                         \ Selecteer 1200/2400 baud
                                         \ Start timer-1
    SET TIMER1
    POWER-MODE $80 OR TO POWER-MODE
                                         \ Verdubbel RS232 ( 2400 baud )
    $52 TO RS232-MODE
                                         \ Seriële mode-1, ontvangst aan
                                         \ Zet zend-interruptvlag
    CLEAR RS232-INTERRUPT
                                         \ Zet RS232-interrupt uit
    ;
: RS232-EMIT
                   ( char -- )
                                  \ Stuur karakter via RS232
    BEGIN RS232-EMIT? UNTIL
    TO RS232-BUFFER
    CLEAR RS232-EMIT?
    ;
: RS232-TYPE
                   ( $addr u -- ) \ Stuur ROMstring via RS232
    ?DUP IF
       FOR
         COUNT RS232-EMIT
       NEXT
    THEN
    2DROP
    ;
: RS232-RTYPE
                   ( addr u -- ) \ Stuur RAMstring via RS232
    ?DUP IF
      FOR
          @+ RS232-EMIT
       NEXT
    THEN
    DROP
    ;
: TYPE
          RS232-TYPE ;
                                                  ( $addr u -- )
         RS232-RTYPE ;
                                                  ( addr u -- )
: RTYPE
            ^M RS232-EMIT ^J RS232-EMIT ;
: CR
                                                 ( -- )
        BL RS232-EMIT ;
: SPACE
           BL RS232-EMIT ;
                                                  ( -- )
                                                  ( -- )
: PAGE
\ Einde RS232
\ Getal uitvoer
ATOM UM/MOD
                                         \ Importeer enkele macro's als woord
ATOM INLINES
                                         \ Zodat ." aangeroepen kan worden
VARIABLE BASE
                                         \ Grondtal voor conversie
VARIABLE HLD
                                         \ Pointer in getalconversiebuffer
$80 CONSTANT NUM
                                         \ Startadres van getalconversiebuffer
                                         \ Alleen geldig voor de AT89C2051
```

```
: M/MOD (d1 u1 -- u2 d2) \ Deel dubbel met teken door single
   >R 0 R@ UM/MOD R> SWAP >R UM/MOD R> \ zonder teken.
   ;
\ Primitieven voor getaluitvoer
\ Zet karakter char in de uitvoer buffer
       DECR HLD FROM HLD ! ;
                                                     ( char -- )
: HOLD
\ Initialiseer de uitvoerbuffer
          NUM TO HLD ;
                                                     ( -- )
: <#
\ Beëindig getalconversie, geef addr en lengte u van de string in de buffer
          2DROP FROM HLD NUM FROM HLD - ;
                                                    ( d -- addr u )
: #>
\ Zet een cijfer van d1 in de uitvoerbuffer, d2 blijft over
: #
          FROM BASE M/MOD ROT 9 OVER <
                                                    ( d1 -- d2 )
           IF 7 + THEN [CHAR] 0 + HOLD ;
\ Uitvoerwoorden voor getallen
\ Druk getal u af met 3 cijfers gevolgd door 2 spaties
: .DATA 0 <# # # # # > RTYPE SPACE SPACE ;
                                                     (u -- )
\ Decimaal grondtal
: DECIMAL #10 TO BASE ;
                                                     ( -- )
\ Einde getaluitvoer
\ De datalogger bewaart elke seconde een meting. Na 20 metingen
\ wordt de data in twee blokken d.m.v. RS232 verstuurd.
$90 SFR LEDS
                                     \ \ Adres van poort-1 voor het aansturen
                                     \ van 8 LED's
$B4 BIT-SFR ADC-KLOK
                                     \ Klokpulsinvoer P3.4
$B5 BIT-SFR ADC-DATA
                                     \ Data-uitvoer P3.5
$B7 BIT-SFR ADC-SELECTEER
                                     \ Activeer invoer P3.7
: ADC
                  ( -- u )
                                     \ Analoog naar digitaalomzetting
   CLEAR ADC-KLOK
                                     \ Stop lezen van analoge invoer
   SET ADC-SELECTEER
                                     \ Start AD-omzetting
   4 FOR NEXT
                                     \ Conversie duurt min. 17 us
   CLEAR ADC-SELECTEER
                                     \ Start uitlezen
   ADC-DATA 1 AND
                                     \ Lees databit-7
   7 FOR
                                     \ Lees resterende 7 bits
      SET ADC-KLOK
       CLEAR ADC-KLOK
                                     \ Schuif volgende bit naar buiten
       2* ADC-DATA 1 AND OR
                                     \ Schaal data in, lees volgende bit
   NEXT
   SET ADC-KLOK
                                     \ Blijf analoge invoer lezen
   ;
: 100MS
             (u--)
                                     \ Wacht u x100 milliseconden
   FOR 100 MS NEXT
: NAAR-LEDS (x -- )
                                     \ Zet x op LED's
   INVERT TO LEDS
                                     \ Inverteer x en stuur LED's aan
   ;
: KNIPPER ( -- )
                                    \ Visualiseer opstarten
   NIPPER ( -- )
-1 NAAR-LEDS 250 MS
                                    \ Alle LED's aan
   0 NAAR-LEDS 250 MS
                                    \ Alle LED's uit
   ;
```

```
: BEELD-AF (u--)
                                      \ Toon met VU- of thermometer-methode
   0 32 UM/MOD
                                      \ Schaal waarde in voor 8 LED's
                                      \ Rond af naar hogere waarde
   SWAP 15 > IF 1+ THEN
   DUP IF
                                      \ Uitkomst groter dan nul ?
       1 SWAP 1- LSHIFT
                                      \ Ja, bepaal positie van LED
١
      DUP 1- OR
                                      \ Vul LED's daaronder in
   THEN
   NAAR-LEDS
                                      \ Stuur LED's aan
   ;
: SAMPLE-ADC ( -- u )
                                      \ Elke tiende AD-omzetting op de stack
   0
                                      \ Dummy
   10 0 DO
                                      \ Toon tien metingen op LED's
                                      \ Verwijder oude AD-omzetting
       DROP
       ADC DUP BEELD-AF
                                      \ Lees ADC en toon resultaat op LED's
       1 100MS
                                      \ Wacht 1/10 sec.
   LOOP
   ;
20 VARIABLES DATA
                                      \ Ruimte voor 20 metingen
: ZEND-TIEN-METINGEN ( offset -- ) \ Stuur 10 metingen over
   10 0 DO
      DUP I + FROM DATA .DATA
   LOOP
   DROP
    ;
: DATALOGGER
             ( -- )
                                      \ Eénkanaals analoge datalogger
   SETUP RS232-SETUP
   DECIMAL PAGE
                                      \ Scherm schoon
    ." ADC naar RS232 data logger"
                                      \ Opstartmelding
   KNIPPER
   ADC DROP
                                      \ Eerste AD-omzetting is ongeldig
   BEGIN
       CR 20 0 DO
                                      \ Bewaar 20 samples
           SAMPLE-ADC
                                      \ Sample ADC en toon resultaat op LED's
          I TO DATA
                                      \ Sla meting I op
       LOOP
       CR 00 ZEND-TIEN-METINGEN
                                      \ Eerste tien
       CR 10 ZEND-TIEN-METINGEN
                                      \ Tweede tien
   AGAIN
    ;
' DATALOGGER RESET-VEC SET-VECTOR
                                      \ DATALOGGER start op na reset
```

\ Einde

# Pulsbreedtemodulatie (PBM)

Met pulsbreedtemodulatie (PBM of Pulse Width Modulation, PWM) kan het vermogen van een gelijkstroombelasting geregeld worden. Als voeding wordt gelijkspanning gebruikt. Aan de uitgang kan een ohmse (lamp) of induktieve (gelijkstroommotor) belasting worden aangesloten.

De voedingsspanning wordt aan- en uitgeschakeld binnen een cyclus van 15 milliseconden. De verhouding tussen de aantijd en de cyclustijd bepaalt het vermogen dat aan de belasting geleverd wordt. De aantijd is maatgevend voor de breedte van de puls en wordt ingesteld met de vier jumpers S1 t/m S4. De 16 mogelijke standen daarvan geven aantijden van 0 t/m 15 ms in stappen van 1 ms; van geheel uit tot vol vermogen.



Schema



### Bedradingsplan



### Onderdelenlijst

Referentie	Omschrijving
S1,S2,S3,S4,J1	2-polige male headerconnector
R1	weerstand 2,2 k $\Omega$
D1	diode 1N4001
T1	FET BUK100-50GL
M1	1 gelijkstroommotor voor 5 V

Headerconnectoren worden in strippen van 20 of 36 pinnen geleverd. Je kunt met een enkele strip volstaan en voor elke connector de gewenste maat afbreken.

## Bouwaanwijzing

Soldeer de volgende componenten op de bovenzijde van de print in de aangegeven gaatjes:

- J1 in K22 en J22
- R1 in O15 en K15
- D1 in O21 en J21
- T1 in H17, H18 en H19
- S1 in N1 en N2, S2 in M1 en M2, S3 in L1 en L2, S4 in K1 en K2.
- Soldeer aan de onderzijde een draad tussen de volgende gaatjes:
- N2 naar N8, M2 naar M8, L2 naar J8, K2 naar I8
- O15 naar O21 en K22
- J22 naar J21 en H18
- K15 naar H17
- Tenslotte H19 naar A19.

Gebruik indien mogelijk de lange aansluitdraden van de componenten hiervoor. Sluit motor M1 aan op connector J1.

#### Software

De timer van de controller loopt zelfstandig. Elke 15 ms wordt een interrupt gegenereerd. De uitgang wordt laag gemaakt zodat de FET en de motor worden ingeschakeld en de aantijd gaat in. De FET blijft gedurende de aantijd ingeschakeld.

Bij een 6-MHz-kristal maakt de processor 500.000 stappen per seconde. Timer-0 wordt gebruikt om de PBMcyclus te genereren. Met TH0 en TL0 is de cyclustijd van 15 ms vastgelegd. Het PBM-signaal komt op poort-1, bit-4 (P1.4). De pulsbreedte wordt ingesteld via poort-3, bits-0 t/m 3.

```
\ Vermogensregeling m.b.v. PulsBreedteModulatie. Lengte 135 bytes
```

89C2051 TARGET	\ Compileer voor AT89C2051
15 CONSTANT CYCLUS	\ PBM-cycluslengte
VARIABLE PERIODE	\ Tel PBM-cyclus af
VARIABLE AANPULS	\ PBM-aantijd
VARIABLE AANTIJD	\ PBM-aan-werkregister
\$94 BIT-SFR UITGANG	\ PBM-uitgang P1.4
\ De PBM-interrupt gebruikt 44x1000+8x50	)=44400 cycles van
$\smallsetminus$ de 500.000, dat is minder dan 10% van	de processortijd.
\ We gebruiken hier de Forth-assembler.	
CODE PBM ( )	\ Timer-0: PBM-opwekking
ACC: PUSH,	\ Bewaar het gebruikte register
TH0: -500. NIP # MOV,	\ Stel herhalingstijd in
TL0: $-500$ . DROP # MOV,	
A: ADR AANTIJD MOV,	\ PBM-aantijd verstreken ?
ZER IF,	
ADR UITGANG CLR,	\ Ja, dan uitgang laag
ELSE,	
ADR UITGANG SETB,	\ Nee, uitgang hoog en
ADR AANTIJD DEC,	\ verlaag uitgang hoogteller
-	

```
THEN,
   ADR PERIODE DEC,
                                       \ Verlaag cyclusteller
                                       \ PBM-cyclustijd verstreken?
   A: ADR PERIODE MOV,
    ZER IF,
       ADR AANTIJD ADR AANPULS MOV,
                                       \ Ja, herstel uitgang tikken
       ADR PERIODE CYCLUS # MOV,
                                       \ en stel cyclusduur in
    THEN,
   ACC: POP,
                                       \ Herstel het gebruikte register
   RETI,
END-CODE
CODE SETUP-PBM
                   ( -- )
   TH0: -500. NIP # MOV,
                                       \ Stel de cyclustijd van teller-0 in
   TLO: -500. DROP # MOV,
   ADR PERIODE CYCLUS # MOV,
                                       \ Stel periodetijd van een PBM-cyclus in
   ADR AANTIJD 0 # MOV,
                                       \ PBM-puls is nul
   ADR AANPULS 0 # MOV,
   ADR UITGANG CLR,
                                       \ PBM-uitgang staat uit
   TMOD: $01 # MOV,
                                       \ Zet Timer-0 in mode-1
   IE: .1 SETB,
                                       \ Sta timer-0 interrupts toe
    TCON: .4 SETB,
                                       \ Start timer-0
   IE: .7 SETB,
                                       \ Zet interruptmechanisme aan
   RET,
END-CODE
: ZET-PULSBREEDTE ( -- )
                                       \ Stel nieuwe pulsbreedte in
   CYCLUS UMIN TO AANPULS
    ;
\ Motor regeling demonstratie
                                       \ Ingang met de schakelaars P3.0-P3.3
$B0 SFR INGANG
: ZET-SNELHEID
                  ( -- u )
                                       \ Lees de schakelaarstand en inverteer
   INGANG INVERT $F AND
                                       \ het resultaat, de laagste 4 bits zijn
    ZET-PULSBREEDTE
                                       \ geldig. Motorsnelheid instellen.
    ;
                  ( -- )
: REGEL-MOTOR
                                      \ Regel een gelijkstroommotor
   SETUP SETUP-PBM
   BEGIN ZET-SNELHEID AGAIN
    ;
' PBM
               TF0-VEC
                           SET-VECTOR \ Installeer de PBM-interrupt
' REGELMOTOR RESET-VEC SET-VECTOR \ REGELMOTOR start op na een reset
```
Relais-aansturing	Hardware EHW-09
	Software ESW-09

Om grote stromen te schakelen wordt vaak een relais toegepast. Met een elektromechanisch relais is er een galvanische scheiding tussen de stuurschakeling en de te schakelen belasting.

Het relais wordt bekrachtigd als de darlingtontransistor gaat geleiden. De transistor geleidt wanneer er een stroom door de basis vloeit, dus wanneer P1.3 hoog is. Deze stroom wordt geleverd door de microcontroller via een interne pull-upweerstand. Daarom is er gekozen voor een transistor met een grote stroomversterking.

De uitgangspoorten van de processor zijn bij het opstarten altijd hoog. Dit heeft tot gevolg dat bij een reset het relais altijd bekrachtigd wordt. In de code wordt daarom eerst de uitgang laag gemaakt, zodat het relais afvalt.

### Schema



### Bedradingsplan



### Onderdelenlijst

Referentie	Omschrijving
S4	2-polige male headerconnector
R2	weerstand 22 k $\Omega$
R3	weerstand 1 k $\Omega$
T2	transistor BC517
D2	diode SB150
J2	2-polige male headerconnector
RL1	relais met 1 * maakcontact
	of met 2 * maakcontact

Headerconnectoren worden in strippen van 20 of 36 pinnen geleverd. U kunt met een enkele strip volstaan en voor elke connector de gewenste maat afbreken.

## Bouwaanwijzing

Soldeer de volgende componenten aan de bovenzijde van de print in de gaten, voor zover dat nog niet eerder gedaan is:

- S4 in L1 en L2
- R2 in O14 en J14
- R3 in J14 of in J15
- J2 in D22 en C22
- diode D2 in G21 en C21

• transistor T2 in A17, B18 en C17. Let op de juiste plaatsing.

Verbind de volgende kopereilandjes via een draad aan de onderzijde van de print met elkaar:

- L2 aan J8
- B18 aan B15
- C17 aan C21 en C22
- Q19 aan G21 en D22

Gebruik als het even kan de overtollige lengte van de aansluitdraden van de componenten.

## Software

Van de microcontroller wordt poort-3, pin-5 (P3.5) gebruikt als ingang voor het schakelen van het relais. Dit schakelen gebeurt met de jumper die gebruikt wordt als S1. De uitgang is P1.4.

```
\ Relaisaansturing met afvalvertraging. Lengte 152 bytes.
```

89C2051 TARGET	\ Compileer voor AT89C2051
\$93 BIT-SFR RELAIS	\ Relaisuitgang P1.3
: RELAIS-AAN SET RELA : RELAIS-UIT CLEAR RE	IS ; LAIS ;
: SETUP-RELAIS ( RELAIS-UIT ;	) \ Zet relais uit

```
\ Relais besturing demonstratie
```

```
$B3 BIT-SFR INGANG
                                     \ Ingang met schakelaar P3.3
: LEES-INGANG
              ( -- flag )
                                      \ Bemonster de schakelaar
   0
                                      \ Beginwaarde
   20 FOR
                                      \ Neem 20 monsters
                                      \ Wacht 5 millisec.
       5 MS
       INGANG INVERT
                                      \ Lees ingang ($00 of $FF)
       1 AND
                                      \ We gebruiken alleen het laagste bit
                                      \ Tel bij de beginwaarde op
       +
   NEXT
   20 =
                                      \ Alle 20 monsters waar ?
   ;
: 10MS
                  (u--)
                                      \ Wacht u x10 millisec.
   FOR 10 MS NEXT
   ;
: BEDIEN-RELAIS
                 ( -- u )
                                      \ Relais snel aan, vertraagd uit
   LEES-INGANG IF
                                      \ Schakelaar gesloten?
     RELAIS-AAN
                                      ∖ Ja, ...
   ELSE
      100 10MS RELAIS-UIT
                                      \ Nee, na 1 sec. afvalvertraging het
   THEN
                                      \ relais uitzetten
   ;
: RELAIS-STURING ( -- )
                                     \ Bedien relais via schakelaar
   SETUP SETUP-RELAIS
                                     \ Zet relais uit
   100 10MS RELAIS-AAN
                                     \ Toon het opstarten door 1x aan/uit-
   200 10MS RELAIS-UIT
                                     \ schakelen van het relais
   BEGIN
      BEDIEN-RELAIS
   AGAIN
   ;
```

' RELAIS-STURING RESET-VEC SET-VECTOR \ RELAIS-STURING start op na een reset

Servo-aansturing	Hardware EHW-10
een re danota mg	Software ESW-10

Servomotoren voor de modelbouw zijn goed verkrijgbaar. In dit project worden twee van dergelijke motoren aangestuurd. De uitgaande as van de toegepaste servo's kan ongeveer 100 graden versteld worden.

Elke servo wordt aangestuurd door een puls met een tijdsduur die ligt tussen 1 en 2 ms. De pulsduur is bepalend voor de stand van de motoras. Een servo reageert alleen op de puls: als er geen puls komt dan verandert de stand van de as niet.

De software hanteert een periode waarbinnen voor elk van de twee servo's een puls gegenereerd wordt, gevolgd door een pauze. Aan het begin van elke periode stuurt de software eerst servo-1 op P3.4 aan met een puls, direkt gevolgd door de puls voor servo-2 op P3.5. Daarna volgt een pauze van 20 ms. De totale tijdsduur van de periode is nu de som van de pulstijden plus de pauzetijd.

De pulstrein:



Het programma maakt positieve pulsen:



Schema



## Bedradingsplan



### Onderdelenlijst

Referentie	Omschrijving
S4	2-polige male headerconnector
R4	weerstand 1 $\Omega$
J3, J4	3-polige male headerconnector
M1, M2	modelbouwservo S21

Headerconnectors worden geleverd in strippen van 36 pinnen. Je kunt met een enkele strip volstaan en voor elke connector het gewenste aantal pinnen afbreken. Servo's worden verkocht bij diverse modelbouwzaken. Quartel met vestigingen in o.a. Amsterdam, Arnhem en Pijnakker levert Robbe servo's voor Fl. 29,00 p.s. Conrad Electronics levert de S21 voor ongeveer Fl. 22,95.

## Bouwaanwijzing

Soldeer de volgende componenten aan de bovenzijde van de print in de aangegeven gaten voor zover dat nog niet eerder is gedaan:

- S4 in K1 en K2
- J3 in H5, H6 en H7
- J4 in F5, F6 en F7

• Steek R4 in Q6 en I6. Solderen op Q6 en H6. Deze belangrijke weerstand dient als stroombegrenzing. Verbind de volgende kopereilandjes aan de onderzijde van de print met een draad aan elkaar:

- K2 aan I8, H7 aan H8, F1 aan F5 en H5
- F7 aan G8. Let op dat er geen contact gemaakt wordt met eilandje F8.

Benut indien mogelijk de aansluitdraden van de componenten.

De servomotoren worden aangesloten op J3 en J4. Let op de wijze van aansluiting; bij Conradtypen moet de zwarte draad altijd links op de connectoren worden aangesloten. De kleuren van de Robbeservo: zwart (-), rood (+) en wit (puls). Van Conrad S21: bruin (-), rood (+) en geel (puls). Er zijn in de software soms enige aanpassingen nodig om de maximale uitslag van een servo te bereiken. Informeer bij Willem Ouwerkerk of kijk op de Forth g.g. website of er voor jouw servo een aanpassing is.

## Software

Het programma stuurt twee servo's aan. Een derde kan eenvoudig worden toegevoegd:

toegepast: wordt voor drie servo's:

IF IF (servo-0) (servo-0) RETI RETI HEN THEN THEN IF IF (servo-1) (servo-1) RETI RETI THEN THEN IF (servo-2) RETI THEN

De uitwerking:

\ Besturing van twee servo's. Lengte 245 bytes.

89C2051 TARGET	\ Compileer voor AT89C2051
\$89 SFR TMOD	\ Timer-0 werkingsmode
\$8C SFR THO	\ Timer-0 registers
\$8A SFR TLO	
\$A9 BIT-SFR ETO	\ Timer-0 interrupt aan-/uitbit
\$8C BIT-SFR TRO	\ Timer-0 start-/stopbit
ŞAF BIT-SFR EAL	\ Interruptmechanisme aan/uit
VARIABLE STAND	\ Toestand van de servo-interrupt
2 2VARIABLES SERVOS	\ Pulsduur van beide servo's
: SERVO (us) >R 1 D2* DNEGATE R> TO SERVOS ;	\ Zet nieuwe stand u van servo s \ Converteer u naar timerwaarde \ voor servo's

```
\ Een servoroutine op interrupts
CODE SERVO-INTERRUPT ( -- )
                                         \ Aansturing voor 2 servo's
    ACC: PUSH,
    A: ADR STAND MOV,
                                          \ Geef interrupttoestand
    ZER IF,
                                          \ Is die nul?
        P3: .4 SETB,
                                          \ Puls op uitgang P3.4
        ADR THO 0 SERVOS MOV,
                                          \ Stel pulsbreedte voor servo-0 in
        ADR TLO 0 SERVOS LOW MOV,
        ADR STAND INC,
                                          \ Naar volgende toestand
        ACC: POP,
       RETI,
    THEN,
    A: DEC,
                                          \ De interrupttoestand was 1
    ZER IF,
                                          \ Is die nul?
        P3: .4 CLR,
                                          \ Stop puls op uitgang P3.4
        P3: .5 SETB,
                                          \ Puls op uitgang P3.5
        ADR TH0 1 SERVOS MOV,
                                          \ Zet pulsbreedte voor servo-1
        ADR TLO 1 SERVOS LOW MOV,
        ADR STAND INC,
                                          \ Naar volgende toestand
        ACC: POP,
        RETI,
    THEN,
                                          \setminus De interrupttoestand is 2
    P3: .5 CLR,
                                          \ Stop puls op uitgang P3.5
    ADR THO -10000. NIP # MOV,
                                          \ Zet 20 millisec. pauzepuls
    ADR TLO -10000. DROP # MOV,
    ADR STAND 0 \# MOV,
                                          \ Begin weer in toestand-0
    ACC: POP,
    RETI,
END-CODE
: SETUP-SERVO
                   ( -- )
                                          \ Stel hardware in voor servo-aansturing
    CLEAR STAND
                                          \ Begin in toestand-0
    $01 TO TMOD
                                          \ Zet timer-0 in mode-1
    SET THO SET TLO
                                          \ Stel timerregister in
    SET ETO
                                          \ Sta timerinterrupt toe
    SET TRO
                                          \ Activeer timer-0
    SET EAL
                                          \ Zet interruptmechaniek aan
    ;
\ Demonstratieprogramma met twee servo's
                                         \ Wacht u x100 millisec.
: 100MS
               (u--)
    FOR 100 MS NEXT
    :
: BEWEEG-SERVOS ( -- )
    SETUP SETUP-SERVO
                                          \ Initialiseer stacks en servohardware
    BEGIN
                                          \ Start servodemo
        25 0 DO
            I 10 * 0 SERVO
                                         \ Zet servo-0, vervolgens
            25 I - 10 * 1 SERVO 3 100MS \ servo-1 en wacht even
        LOOP
        20 100MS
    AGAIN
    ;

    SERVO-INTERRUPT TF0-VEC SET-VECTOR \ Installeer timer-0 interrupt
    BEWEEG-SERVOS RESET-VEC SET-VECTOR \ BEWEEG-SERVOS start op na een reset
```

# **Unipolaire stappenmotor**

De toegepaste unipolaire stappenmotor heeft twee stators met ieder twee wikkelingen. In totaal zijn er dan vier wikkelingen: de vier fasen. Steeds wordt per stator 1 wikkeling bekrachtigd. De motor heeft zes aansluitingen.





De motor kan worden aangestuurd in: hele-stapbedrijf: eenfase-aansturing: telkens 1 wikkeling bekrachtigd tweefasen-aansturing: 2 wikkelingen bekrachtigd halve-stapbedrijf: een combinatie van deze twee.

Microstepping wordt hier niet toegepast.

Elk aansluitpunt wordt gecodeerd met een waarde zodat de motor met een getal aangestuurd kan worden:

poort	P1.4	P1.5	P1.6	P1.7	masker
waarde	\$10	\$20	\$40	\$80	
tweefasen aansturing:	•			•	\$90
			•	•	\$C0
		•	•		\$60
	•	•			\$30

De wikkelingen van de stator vragen meer stroom dan de uitgangen van de controller kunnen leveren. Daarom wordt driver-IC ULN2803 toegepast om het benodigde vermogen te kunnen leveren. Dit IC bevat acht laagvermogen-transistoren waarvan er vier gebruikt worden voor het aansturen van de wikkelingen. Op elk moment wordt voor eenfasebedrijf slechts een wikkeling van een stator met massa verbonden. Voor tweefasenbedrijf wordt van beide stators elk een wikkeling aan massa gelegd.

Omdat de gemeenschappelijke aansluiting met de plus is verbonden krijgen de wikkelingen de volle voedingsspanning. In beide gevallen is er sprake van hele-stapbedrijf. In halve-stapbedrijf worden afwisselend een of twee fasen bekrachtigd, waardoor het aantal stappen per omwenteling tweemaal zo groot is als bij het hele-stapbedrijf.

# Schema



Bedradingsplan



pin 1

## Onderdelenlijst

Referentie	Omschrijving
S1,S2,S3,S4, J2	2-polige male headerconnector
RA1	weerstandnetwerk 8-voudig 4,7 k $\Omega$ met 1 common
U1	ULN2803A
J1	6-polige male headerconnector
-	18-pins IC-voetje
M1	unipolaire stappenmotor, bijvoorbeeld Kemo PM35S

Headerconnectors worden geleverd in strippen van 36 pinnen. Je kunt met een enkele strip volstaan en voor elke connector het gewenste aantal pinnen afbreken. De KEMO stappenmotor is verkrijgbaar bij alle elektronicawinkels die het KEMO-assortiment voeren, b.v. Display Elektronika.

### Bouwaanwijzing

Soldeer de volgende componenten aan de bovenzijde van de print in de aangegeven gaten, voor zover dat nog niet eerder gedaan is:

- de connectors S1 in N1 en N2, S2 in M1 en M2, S3 in L1 en L2, S4 in K1 en K2
- weerstandsnetwerk RA1 in G14 tot en met O14 ;
- het IC-voetje voor U1 met pin 1 in N16 zoals aangegeven
- connector J1 in J21 tot en met O21.

Verbind de volgende kopereilandjes met een draad aan de onderzijde van de print aan elkaar:

- N2 aan N8, M2 aan M8, L2 aan J8, K2 aan I8
- K15 aan K16, L15 aan L16, M15 aan M16, N15 aan N16
- N19 aan O21, M19 aan M21, L19 aan L21, K19 en J21
- A6 aan F6, F19 aan K21, N21 en Q19

Een aparte voedingsspanning voor de motor kan worden aangesloten op J2. Wanneer u een 5V-stappenmotor gebruikt en deze via het breadboard wilt voeden, dan dient u een spanningsstabilisator van 1 A te gebruiken. In andere gevallen dient een aparte voeding met voldoende stroomcapaciteit gebruikt te worden. De ULN2803 kan per uitgang maximaal 0,5 A en maximaal 50 V verwerken.

De KEMO stappenmotor wordt via de ULN2803 aangesloten volgens onderstaande tabel:

	J1 pin	stappenmotor
P1.7	1	Oranje
+V-motor	2	Rood
P1.6	3	Zwart
P1.5	4	Geel
+V-motor	5	Rood
P1.4	6	Bruin

### Software

Met S1 (P3.0) wordt de stapsnelheid ingesteld: 0= staptijd 10 ms, 1= staptijd 20 ms.

Met S2 (P3.1) en S3 (3.2) wordt de wijze van aansturing gekozen: (binair, in de volgorde S3-S2!) 00= uit, 01= éénfase, 10= tweefasen, 11= halfstap.

Met S4 (P3.3) wordt de draairichting ingesteld.

 $\$  Sturing voor unipolaire stappenmotor. Lengte 205 bytes.

89C2051 TARGET	\ Compileer voor AT89C2051
\$90 SFR MOTOR \$80 SFR INGANG \$83 BIT-SFR RICHTING?	<pre>\ Stappenmotor aan uitgangen P1.4 - P1.7 \ Schakelaar aan ingangen P3.0 - P3.2 \ Vooruit/achteruitselectie P3.3</pre>
VARIABLE STAP VARIABLE STAP#	\ Volgend uitgangswoord \ Tabel masker

```
        $80 $40 $20 $10
        4 CONSTANTS 1FASE \ Enkelfase-aansturing (n - x)

        $90 $C0 $60 $30
        4 CONSTANTS 2FASE \ Dubbelfase-aansturing (n - x)

$90 $80 $C0 $40 $60 $20 $30 $10 8 CONSTANTS HALF \ Halvestap-aansturing ( n - x )
: EEN-FASE FROM STAP 1FASE TO MOTOR 3 TO STAP# ;
: TWEE-FASE FROM STAP 2FASE TO MOTOR 3 TO STAP# ;
: HALF-STAP FROM STAP HALF TO MOTOR 7 TO STAP# ;
: WACHT
               ( -- )
                                       \ Wacht 20 of 40 milliseconden
   INGANG INVERT 1 AND 1+ 20 * MS
    ;
: EEN-STAP
               ( -- )
    INGANG INVERT $06 AND
                                       \ Lees type sturing
    BEGIN-SELECT
                                        \ Selecteer sturing
       SELECT $02 EEN-FASE EXIT \ Sturing via éénfase
       SELECT $04 TWEE-FASE EXIT \ Sturing via tweefasen
       SELECT $06 HALF-STAP EXIT \ Sturing via halfstap
                  CLEAR MOTOR \ Geen motorsturing
    END-SELECT
    ;
: VOORUIT ( -- )
                                     \ Motor een stap vooruit
    FROM STAP 1+ FROM STAP# AND TO STAP
    EEN-STAP WACHT
    ;
: ACHTERUIT ( -- ) \
FROM STAP 1- FROM STAP# AND TO STAP
                                      \ Motor een stap achteruit
    EEN-STAP WACHT
    ;
: SETUP-STAP ( -- )
    3 TO STAP#
                                       \ Zet default tabelmasker
    CLEAR STAP
                                        \ Begin bij de eerste stap
    ;
: BEWEEG ( -- )
                                       \ Kies de draairichting van de motor
   RICHTING? IF VOORUIT ELSE ACHTERUIT THEN
    ;
                                      \ Bestuur een 4-fasen-stappenmotor
: STAP-MAAR
              ( -- )
    SETUP SETUP-STAP
    BEGIN BEWEEG AGAIN
    :
' STAP-MAAR RESET-VEC SET-VECTOR \ STAP-MAAR start op na een reset
```

# **Bipolaire stappenmotor**

#### Zie appendix Stappenmotoren

De hier toegepaste motor heeft twee stators met elk een wikkeling: de twee fasen. Er zijn vier aansluitdraden. De twee wikkelingen kunnen als volgt bekrachtigd worden:





De motor kan worden aangestuurd in:

hele-stapbedrijf:	eenfase-aansturing:			
	tweefasen-aansturing:			
halve-stapbedrijf:	een combinatie van deze twee.			

telkens 1 wikkeling bekrachtigd 2 wikkelingen bekrachtigd

Elk aansluitpunt wordt gecodeerd met een waarde:

poort	P1.4	P1.5	P1.6	P1.7	masker
waarde	\$10	\$20	\$40	\$80	
tweefasen-aansturing:	+	-	-	+	\$90
	-	-	+	+	\$C0
	-	+	+	-	\$60
	+	+	-	-	\$30

Om de stappenmotor van voldoende stroom te voorzien wordt een driver-IC van het type L293D toegepast. Dit IC is geschikt voor induktieve belastingen. Het bevat vier push-pulldrivers waardoor de schakeling eenvoudig kan blijven. Via de uitgang van elke driver wordt een aansluitpunt van de stappenmotor met de voedingsspanning of met massa verbonden.

Schema







## Onderdelenlijst

Referentie	Omschrijving
\$1,\$2,\$3,\$4, J1,	2-polige male headerconnector
J2, J3	
RA1	weerstandarray 8-voudig 4,7 k $\Omega$ met 1
	common
U1	L293D (maximaal 0,6A per uitgang)
-	IC-voet DIL-16
M1	bipolaire stappenmotor, bijvoorbeeld Kemo
	RDM37

Headerconnectors worden geleverd in strippen van 36 pinnen. Je kunt met een enkele strip volstaan en voor elke connector het gewenste aantal pinnen afbreken. De KEMO stappenmotor is verkrijgbaar bij alle elektronicawinkels die het KEMO assortiment voeren, waaronder Display Elektronika.

### Bouwaanwijzing

De bedrading onder U1 (L293D) behoeft extra aandacht. Monteer de volgende geïsoleerde doorverbindingen aan de componentenzijde:

- Q19 naar P19, P16 naar I16, M18 naar L18, M19 naar L19, A21 naar L21
- Plaats het voetje voor U1 met pin-1 op P17 zoals aangegeven
- Plaats J1 op de locaties M22 en N22
- Plaats J2 op de locaties K22 en L22
- Plaats het weerstandsnetwerk RA1 op de locatie O14 t/m G14
- Breng aan de soldeerzijde de volgende geïsoleerde doorverbindingen aan:
- N15 naar O17, M15 naar O20, L15 naar J17, K15 naar J20
- N17 naar N22, N20 naar M22, K17 naar L22, K20 naar L22

Plaats aan de linkerzijde de pinnen voor S1 t/m S4 en verbind deze aan de onderzijde van de print zoals aangegeven.

De L293D kan een stroom leveren van 0,6 A per uitgang en kan een spanning schakelen van maximaal 30 V. Hier geldt hetzelfde als voor de unipolaire stappenmotor: gebruik een robuuste voeding.

Dit project heeft erg veel doorverbindingen onder U1. Als je niet echt ervaren bent neem dan geïsoleerde draden van zo'n 3 cm lengte en ongeveer 0.7 mm<sup>2</sup> doorsnee om de langere doorverbindingen te maken. Vouw, als je helemaal klaar bent, de draden tegen de print om te voorkomen dat ze ergens achter blijven haken en afbreken. Soldeer een 16-pins IC-voetje op de plaats U1. Zet U1 (L293D) in het voetje. Soldeer nu het weerstandsnetwerk RA1 en de connectors J1 en J2. Leg vervolgens de doorverbindingen die steek 1 (0.1 inch) uit elkaar liggen en de oversteek van pin 5 naar pin 12 onder U1, evt. ongeïsoleerd. Breng de overige doorverbindingen aan met geïsoleerd draad. Tenslotte de schakelaars aan de linkerzijde monteren en met de processor verbinden zoals bij o.a. EHW-11.

Sluit de draden van de stappenmotor als volgt aan op de connectors J1 en J2: J1.1 rose, J1.2 geel, J2.1 blauw, J2.2 zwart

Op connector J3 kan een aparte voedingsspanning voor de motor worden aangesloten. Wilt je 5-Voltstappenmotoren gebruiken dan kun je van de 5-Volt-spanning op het breadboard gebruik maken mits de voeding constant 0.5 A kan leveren. De L293 kan maximaal 0.6 A per uitgang doorlaten. Bestudeer eventueel de bouwtekening van het AT51-breadboard in het ByteForthboek om een zwaardere spanningsstabilisatie aan te brengen, bijvoorbeeld 78S05 in plaats van 78L05 en 1N4002 in plaats van 1N4148.

#### Software

Met S1 (P3.0) wordt de stapsnelheid ingesteld: 0= staptijd 10 ms, 1= staptijd 20 ms. Met S2 (P3.1) en S3 (3.2) wordt de wijze van aansturing gekozen: (binair, in de volgorde S3-S2!) 00= uit, 01= éénfase, 10= tweefasen, 11= halfstap. Met S4 (P3.3) wordt de draairichting ingesteld.

\ Sturing voor bipolaire stappenmotor. Lengte 205 bytes.

```
89C2051 TARGET
                                      \ Compileer voor de AT89C2051
$90 SFR MOTOR
                                       \ Stappenmotor aan uitgangen P1.3 - P1.7
$B0 SFR INGANG
                                       \ Schakelaar aan ingangen P3.0 - P3.2
$B3 BIT-SFR RICHTING?
                                       \ Vooruit/achteruitselectie P3.3
VARIABLE STAP
                                       \ Volgend uitgangswoord
VARIABLE STAP#
                                       \ Tabelmasker
$DD $BB $EE $77
                               4 CONSTANTS 1FASE \ Enkelfase-aansturing
                               4 CONSTANTS 2FASE \ Dubbelfasen-aansturing
$99 $AA $66 $55
$DD $99 $BB $AA $EE $66 $77 $55 8 CONSTANTS HALF \ Halfstaps-aansturing
: EEN-FASE FROM STAP 1FASE TO MOTOR 3 TO STAP# ;
: TWEE-FASE FROM STAP 2FASE TO MOTOR 3 TO STAP# ;
: HALF-STAP FROM STAP HALF TO MOTOR 7 TO STAP# ;
                                       \ Wacht 10 of 20 milliseconden
: WACHT
               ( -- )
   INGANG INVERT 1 AND 1+ 10 * MS
    ;
: EEN-STAP
              ( -- )
   INGANG INVERT $06 AND
                                       \ Lees sturing type bits
   BEGIN-SELECT
                                       \ en selecteer sturing
       SELECT $02 EEN-FASE
                                      \ Sturing via éénfase
                              EXTT
       SELECT $04 TWEE-FASE EXIT \ Sturing via tweefasen
       SELECT $06 HALF-STAP EXIT \ Sturing via halfstap
                   CLEAR MOTOR
                                       \ Geen motorsturing
   END-SELECT
    ;
: VOORUIT
               ( -- )
                                       \ Motor een stap vooruit
   FROM STAP 1+ FROM STAP# AND TO STAP
   EEN-STAP WACHT
    ;
: ACHTERUIT
              ( -- )
                                       \ Motor een stap achteruit
   FROM STAP 1- FROM STAP# AND TO STAP
   EEN-STAP WACHT
    ;
: SETUP-STAP
               ( -- )
   3 TO STAP#
                                       \ Zet default tabelmasker
   CLEAR STAP
                                       \ Begin bij de eerste stap
    ;
: BEWEEG
                                       \ Kies draairichting van de motor
               ( -- )
   RICHTING? IF VOORUIT ELSE ACHTERUIT THEN
    ;
: STAP-MAAR
               ( -- )
                                       \ Bestuur een 2 fasen-stappenmotor
   SETUP SETUP-STAP
   BEGIN BEWEEG AGAIN
    ;
STAP-MAAR
             RESET-VEC
                                SET-VECTOR
                                            \ STAP-MAAR start op na een reset
```

# Frequentiemeting

Van een blokvorming signaal meten we de periodeduur. Uit deze tijd wordt de frequentie berekend. Het blokvormig signaal wordt hier gesimuleerd door schakelaar S2. De schakeling wordt vrijgegeven met schakelaar S1. De meting wordt gestart door het bedienen van schakelaar S2 en gestopt door S2 opnieuw te bedienen.



Van het ingangssignaal worden de neergaande flanken gebruikt omdat de OR-ingangen van de microcontroller dan rechtstreeks zijn aan te spreken.

De tijdsduur van een periode wordt bepaald door het aantal pulsen te tellen van een door timer-1 geleverde pulstrein. De pulstrein zelf heeft een periode van 1 ms. De teller is 16 bits breed zodat de langste te meten periode ruim 65 seconden bedraagt. De nauwkeurigheid wordt beperkt door twee factoren:

- 1. De resolutie van het timersignaal is 1 ms. Daardoor ontstaat aan het begin en aan het eind van de meting een fout. De invloed hiervan is maximaal +/-1 ms, dus relatief klein.
- Door denderonderdukking van de schakelaar gaat aan het begin van de meting 4 ms verloren. Denderonderdrukking is het gedurende enkele milliseconden filteren van snelle schakelpulsen, die ontstaan door het stuiteren van het schakelcontact. Vooral bij korte signalen is de invloed daarvan groot.

Het resultaat van de meting wordt via de RS232-aansluiting naar een computer verzonden.

## Schema

Voor deze toepassing wordt het printje EHW-06 van projecten 6 en 7 weer gebruikt. De uitvoer wordt ook hier door de LED's weergegeven.



## Onderdelenlijst

Referentie	Omschrijving
C1, C2, C3, C4	Elco 10 μF - 16 Volt
J1	3-polige contactstrip female
J2	9-polige D-connector female
J3	3-polige contactstrip male
U1	MAX232 buffer/omzetter
-	DIL-16 IC-voet
S1, S2	dubbelpolige headerstrip

#### Bouwaanwijzing

Soldeer de draadbruggen en de connector J3 op de print, breng daarna de IC-voet aan op de plaats van U1. Tenslotte C1 t/m C4 monteren; let op de + en -. Sluit J1 en J2 aan zoals in het bedradingsplan is aangegeven.

#### Software

Bij het assembleren wordt, alvorens de ByteForthwoordenlijst te doorzoeken, eerst nog een aparte woordenlijst doorzocht. Indien nodig kunnen we woorden uit deze lijst ook benaderen zonder dat wordt geassembleerd. ASSEMBLER - De woordenlijst ASSEMBLER wordt gebruikt WORK - Alleen de ByteForthwoordenlijst wordt geldig gemaakt

```
\ Tijdsduurmeting (TOERENTELLER, rondentijd, afstandsmeter etc). Lengte 758 bytes.
                                        \ Compileer voor de AT89C2051
89C2051 TARGET
\ RS232-I/O
$99 BIT-SFR RS232-EMIT?
                                        \ RS232-zendbuffer leeg ?
$8E BIT-SFR TIMER1
                                        \ Timer-1 start/stop
SAC BIT-SFR RS232-INTERRUPT
                                        \ Interrupt timer-1 aan/uit
ASSEMBLER
                                        \ Gebruik assemblerlabels
TMOD: SFR TIMER-MODE
                                        \ Timerinstelling
PCON: SFR POWER-MODE
                                        \ RS232-snelheidsverdubbeling
SCON: SFR RS232-MODE
                                        \ RS232-instelling
TH1: SFR TIMER1-HIGH
                                        \ Hoge byte van timer-1
SBUF: SFR RS232-BUFFER
                                        \ RS232-in-/uitvoerbuffer
WORK
: RS232-SETUP
    TIMER-MODE $20 OR TO TIMER-MODE
                                        \ Timer-1 mode-2 = 8-bit-autoreload
    243 TO TIMER1-HIGH
                                        \ Selecteer 1200/2400 baud
    SET TIMER1
                                        \ Start timer-1
    POWER-MODE $80 OR TO POWER-MODE
                                        \ Verdubbel RS232-snelheid (2400 baud)
    $52 TO RS232-MODE
                                        \ Seriële mode-1, ontvangst aan
                                        \ Zet zend-interruptvlag
    CLEAR RS232-INTERRUPT
                                        \ Zet RS232-interrupt uit
    ;
: RS232-EMIT
                    ( char -- )
                                        \ Een karakter naar RS232
    BEGIN RS232-EMIT? UNTIL
    TO RS232-BUFFER
    CLEAR RS232-EMIT?
    ;
: RS232-TYPE
                    ( $addr u -- )
                                        \ Een string uit ROM naar RS232
    ?DUP IF
        FOR
            COUNT RS232-EMIT
        NEXT
    THEN
    2DROP
    ;
: RS232-RTYPE
                    ( addr u -- )
                                       \ Een string uit RAM naar RS232
    ?DUP IF
        FOR
            @+ RS232-EMIT
        NEXT
    THEN
    DROP
    ;
           RS232-TYPE ;
: TYPE
                                                ( $addr u -- )
: RTYPE
           RS232-RTYPE ;
                                                ( addr u -- )
: CR
           ^M RS232-EMIT ^J RS232-EMIT ;
                                                ( -- )
```

```
: SPACE BL RS232-EMIT ;
                                             ( -- )
: PAGE
          ^L RS232-EMIT ;
                                              ( -- )
\ Einde RS232
\ Getal uitvoer routines
ATOM UM/MOD
                                      \ Importeer enkele macro's als woord
ATOM INLINE$
VARIABLE BASE
                                      \ Conversiegrondtal
VARIABLE HLD
                                      \ Werkadres in getalconversiebuffer
$80 CONSTANT NUM
                                      \ Startadres van getalconversiebuffer
                                      \ Alleen geldig voor de AT89C2051
: M/MOD
          ( d1 u1 -- u2 d2 )
                                     \ Deel dubbel met teken door single
   >R 0 R@ UM/MOD R> SWAP >R UM/MOD R> \ zonder teken.
    ;
\ Getaluitvoerprimitieven
\ Zet karakter char in de uitvoerbuffer
: HOLD DECR HLD FROM HLD ! ;
                                                      ( char -- )
\ Initialiseer de uitvoerbuffer
: <#
          NUM TO HLD ;
                                                      ( -- )
\ Beëindig getalconversie, geef addr en lengte u van de string in de buffer
           2DROP FROM HLD NUM FROM HLD - ;
: #>
                                                     ( d -- addr u )
\ Zet een cijfer van d1 in de uitvoerbuffer, d2 blijft over
           FROM BASE M/MOD ROT 9 OVER <
                                                      ( d1 -- d2 )
: #
           IF 7 + THEN [CHAR] 0 + HOLD ;
\ Verzamel alle cijfers in de cijferstring onder (NUM)
       BEGIN # 2DUP OR 0= UNTIL ;
                                                      ( d1 -- d2 )
: #S
\ Uitvoer van getallen
\ Toon ud in een veld met vrij formaat
          <# #S #> RTYPE SPACE ;
: DU.
                                                      (d -- )
\ Toon u in idem
: U. 0 DU.;
                                                      (u -- )
\ Zet grondtal decimaal
: DECIMAL #10 TO BASE ;
                                                      ( -- )
\ Einde getaluitvoer
\ Start tijdmeting routines
    $B2 BIT-SFR IN
                                      \ Ingang-P3.2 meet periode
    $B5 BIT-SFR STARTKNOP
                                      \ Start meting P3.5
    2VARIABLE TELLER
                                      \ Periodetijdteller
   2VARIABLE MEETTIJD
                                      \ Onthoud periodetijd
                                      \ Onthoud frequentie
    2VARIABLE FREQUENTIE
   FLAG DENDER?
                                      \ Denderonderdrukking
   FLAG METING?
                                      \ Complete meting gedaan
   FLAG KLAAR?
                                      \ Start een meetcyclus
```

```
\ Interruptroutine voor periodetijdmeter, de maximaal te meten
\ periodetijd is 65535 x 1 / 1000 = 65,5 sec. De kortste te meten periode
\ bedraagt in principe 4 millisec met een afwijking van maximaal 20%.
CODE TEL
                      ( -- )
                                       \ Wordt elke 1/1000 sec geactiveerd
   ACC: PUSH,
                                       \ Bewaar de gebruikte registers
   PSW: PUSH,
    TH0: -500. NIP # MOV,
                                       \ Zet negatieve 16-bits waarde in timer-0
    TL0: -500. DROP # MOV,
    ADR TELLER LOW INC,
                                       \ Verhoog lage byte van teller
    A: ADR TELLER LOW MOV,
   ZER IF,
                                       \ Overflow?
       ADR TELLER INC,
                                       \ Ja, verhoog hoge byte
    THEN,
   A: 4 # EQL IF,
                                       \ Denderonderdrukking van 4 millisec.
       ADR DENDER? CLR,
                                       \ Sta nieuw signaal toe
    THEN,
   PSW: POP,
                                       \ Herstel de gebruikte registers
   ACC: POP,
   RETI,
END-CODE
CODE NOTEER
                      ( -- )
                                       \ Noteer periodetijd
   ADR KLAAR? BS IF, RETI, THEN,
                                      \ Meting afgerond? doe dan niets!
   ADR DENDER? BC IF,
                                      \ Ingang vrijgegeven?
       ADR METING? BC IF,
                                       \ Begin een meting?
           ADR TELLER LOW 0 # MOV,
                                       \ Ja, reset teller
           ADR TELLER 0 # MOV,
           ADR METING? SETB,
        ELSE,
           ADR MEETTIJD LOW ADR TELLER LOW MOV, \ Nee, bewaar periodetijd
           ADR MEETTIJD ADR TELLER MOV,
           ADR METING? CLR,
                                       \ Meetcyclus afgerond
           ADR KLAAR? SETB,
       THEN,
    THEN,
    ADR DENDER? SETB,
                                       \ Activeer denderonderdrukking
    RETI,
END-CODE
CODE TIJDMETING-SETUP ( -- )
    TMOD: $01 # ORL,
                                       \ Timer-0 in mode-1, 16-bits-timer
    TH0: -100. NIP # MOV,
                                       \ Timer-0 start waarde
    TL0: -100. DROP # MOV,
   IP: $01 # MOV,
                                      \ Externe interrupt hoge prioriteit
    TCON: $11 # MOV,
                                      \ Start timer-0 & ing. flank getriggerd
   IE: $83 # MOV,
                                      \ Zet interrupts EX0 & ET0 aan
   ADR METING? CLR,
                                      \ Nog geen meetcyclus begonnen
    ADR DENDER? CLR,
                                       \ Denderonderdrukking niet actief
   ADR KLAAR? SETB,
                                       \ Meetcyclus niet actief
   RET,
END-CODE
: START-METING
                       ( -- )
   BEGIN 10 MS STARTKNOP 0= UNTIL
                                       \ Knop ingedrukt
   BEGIN 10 MS STARTKNOP UNTIL
                                      \ Knop losgelaten
    CLEAR KLAAR?
    ;
: BEREKEN-TOERENTAL ( -- ud ) \ Converteer periodetijd naar omw/min
    60000. 0. FROM MEETTIJD DUM/MOD 2NIP
    ;
```

```
: TOON-METING
                            ( -- )
    BEGIN KLAAR? UNTIL
                                                  \ Wacht tot meting is gedaan,
    CR ." Duur van een meting: "
    FROM MEETTIJD DU.
                                                  \ Print dan de periodetijd en
     ." millisec."
    CR ." Toerental: "
    BEREKEN-TOERENTAL DU.
                                                  \ Bereken en print het toerental
     ." omw/min. " 50 MS
     ;
\ Het meetbereik is in de huidige configuratie van 1 tot 15000 toeren.
: TIJDMETING (--)
    SETUP TIJDMETING-SETUP RS232-SETUP
    DECIMAL CLEAR MEETTIJD 100 MS
    PAGE CR ." 'Egel toerentalmeter" CR
    BEGIN
        START-METING TOON-METING
    AGAIN
    ;
' TEL
                TF0-VEC
                               SET-VECTOR
                                                  \ TEL is de periodetijdtimer

      ' TEL
      TF0-VEC
      SET-VECTOR
      \ TEL is de periodetijdtimer

      ' NOTEER
      IE0-VEC
      SET-VECTOR
      \ NOTEER meet de periodetijd

      ' TIJDMETING RESET-VEC
      SET-VECTOR
      \ TIJDMETING start op na een reset
```

```
\ End
```

# **Ontvanger Afstandsbediening**

Zie appendix RC5

In dit project wordt een ontvanger gebouwd voor infraroodsignalen van een tv-afstandsbediening volgens het RC5-protocol. Appendix RC5 beschrijft dit protocol. Met de aan-/uitknop van de afstandsbediening wordt via de ontvanger een solidstaterelais geschakeld. In de software hoeft alleen de waarde van de constante RC-TYPE veranderd te worden om de ontvanger "om te bouwen" tot een afstandsbediening voor bijvoorbeeld Video of CD-speler. Dit ontwerp werkt ook samen met de zender uit project ESW-15.

De drie niveaus van het ontvangen signaal worden als volgt verwerkt.

- 1. Het 36-kHz-basissignaal wordt door een IR-diode ontvangen en vervolgens hardwarematig gelijkgericht tot bitsignalen of pulsen. De tijdsduur hiervan wordt vooral bepaald door de waarden -430. en -215. in de listing bij LEES-BIT respectievelijk RC-START.
- 2. De bitsignalen worden ingelezen via poort-P3.2 Nadat de startbit gelezen is wordt gedurende iedere bit het signaal 2 maal bemonsterd. Op deze manier is het mogelijk een fout in de data-overdracht vast te stellen. De bitstroom die een TV bestuurt bevat 5 systeembits met de waarde 0 gevolgd door 6 commandobits. Voor de aan/uitknop hebben de commandobits een waarde van 12 decimaal:



De fieldbit (tweede startbit) is 1. Met de controlbit wordt geen rekening gehouden.

Bemonstering van de bitstroom (de curve is geïnverteerd):



Opmerking:

De ontvanger inverteert het ontvangen signaal. Van de startbit wordt de eerste helft niet verwerkt. De ontvanger komt in actie op de flank van de eerste puls die ontvangen wordt. Daarna worden van elke bit twee waarden bepaald: één op 1/4 van de bittijdsduur en één op 3/4 van de bittijdsduur. Deze twee waarden moeten voor een geldig signaal tegengesteld zijn.

3. De ontvanger ontvangt signalen en verwerkt deze zolang de toets op de afstandsbediening is ingedrukt.

# Schema







12 3 4 uit uit in+ in-

Referentie	Omschrijving
U1	infrarode detector SFH506-36 of TSOP1836 of
	TSOP4838
U2	solidstate relais S201S01 (max. stroom 1,5 A)
	of: S202SE1 (max. stroom 8 A)
	of: S202S01 (max. stroom 8 A)
D1	rode LED $\emptyset$ 5 mm
C1	Elco 22 μF 35V
R1	weerstand 150 $\Omega$
R2	weerstand 330 $\Omega$
R3	weerstand 330 $\Omega$
J1	2-polige printkroonsteen

### Bouwaanwijzing

We bouwen eerst de linkerzijde van het breadboard. Soldeer eerst de weerstanden, gevolgd door de LED, de elko en de RC5-ontvanger. Plaats aan de rechterzijde de weerstand, het solid-staterelais en het kroonsteentje. Leg aan de onderzijde de doorverbindingen zoals op het bedradingsplan staat aangegeven.

Er moet op gelet worden dat er geen contact ontstaat tussen het wisselspanningsgedeelte en de gelijkspanning. Het is en blijft wisselspanning en dat is gevaarlijk boven 24 Volt. Test eventueel met een beltrafo. Die heeft een gering vermogen en een lage uitgangsspanning.

Veiligheidsnormen schrijven voor dat er voldoende isolatie is tussen geleiders die netspanning voeren. Het is daarom raadzaam de ongebruikte kopereilandjes rondom de printkroonsteen en het solid-staterelais weg te krassen.

### Software

```
\ RC5-ontvanger, -decoder en lampaan-/uitsturing. Lengte 260 bytes.
```

```
89C2051 TARGET
                                         \ Compileer voor AT89C2051, xt = 6 MHz
#00 CONSTANT RC-TYPE
                                         \ RC5-zendertype
                                         \setminus TV = 0, CD = 20 en VCR = 5
VARIABLE RC-SYS
                                         \ Systeemdatabyte (8 bits)
VARIABLE RC-DATA
                                         \ Commandodatabyte (6 bits)
VARIABLE RC#
                                         \ Aantal ontvangen bits
\ Geef true als er een voor dit systeem geldig RC5-commando is.
FLAG RC-KEY? ( -- flag )
FLAG RC-CONTROLE
                                         \ Eerste halve bit, voor controle
                                         \ Tweede halve bit, voor controle
FLAG RC-BIT
$B2 BIT-SFR RC-IN
                                         \ Ingangsbit-P3.2
CODE BEWAAR-RC-BIT ( -- )
                                         \ Bewaar gedecodeerde bits
    A: 16 # ANL,
    ZER IF,
                                         \ Is het een systeembit?
       A: RC-SYS MOV,
                                         \ Ja, ...
        A: RLC,
        RC-SYS A: MOV,
    ELSE,
                                         \ Nee, het is een databit
        A: RC-DATA MOV,
        A: RLC,
        RC-DATA A: MOV,
    THEN,
    RC# INC,
                                         \ Verhoog de bitteller
    RET,
END-CODE
```

```
CODE LEES-BIT ( -- )
                                        \ Lees en decodeer een bit
   PSW: PUSH,
   ACC: PUSH,
    TH1: -430. NIP # MOV,
                                        \ Zet wachttijd tot volgende meting 860 us
   TL1: -430. DROP # MOV,
                                        \ plus 20 us looptijd = 880 us
   C: ADR RC-IN MOV,
                                        \ Lees RC5-bit
   ADR RC-BIT C: MOV,
                                        \ Bewaar voor patrooncontrole
   A: RC# MOV,
                                        \ Haal bitteller op
   ACC: .0 BS IF,
' BEWAAR-RC-BIT CALL,
                                        \ Verzamel elke even bit
                                        \ Bewaar bit
       A: CLR,
                                        \ Controleer bitpatroon....
        C: ADR RC-CONTROLE MOV,
                                        \ Lees vorige RC5-bit
        A: RLC,
                                        \ Zet op positie-1 in accu
        C: ADR RC-BIT MOV,
                                        \ Lees actuële RC5-bit
        A: 0 \# ADDC,
                                        \ Tel de bits bij elkaar op
        A: 1 # EQL IF,
                                        \ Geldig patroon 0/1 of 1/0 opgepikt?
           A: RC# MOV,
                                       \ Einde van de bitstroom
               $1C # EQL IF,\ Als er 14 bits opgepikt zijnRC-SYS $FF # XRL,\ Keer systembits omRC-DATA $FF # XRL,\ Keer databits omRC-DATA $3F # ANL,\ De 6 lage databits zijn geld
            A: $1C # EQL IF,
                                        \ De 6 lage databits zijn geldig
                A: RC-SYS MOV,
                A: $1F # ANL,
                                        \ Pik zender-ID op
                A: RC-TYPE # EQL IF, \ Is dat van het juiste type?
                    ADR RUMER IE: .0 SETB,
                                        \ EX0-interrupt weer aan
                                        \ Stop timer-1
                    TCON: .6 CLR,
                THEN,
           THEN,
        ELSE,
                                        \ Ongeldig patroon, stop decoder
           IE: .0 SETB,
                                        \ EX0-interrupt aan
            TCON: .6 CLR,
                                        \ Stop timer-1
        THEN,
    ELSE.
        ADR RC-CONTROLE C: MOV,
                                        \ Bewaar eerste halve bitwaarde
       RC# INC,
                                        \ Verhoog bitteller
    THEN.
   ACC: POP,
   PSW: POP,
   RETI,
END-CODE
                 ( -- )
CODE RC-START
                                        \ Flank gezien, start decoder
   PSW: PUSH,
   ADR RC-KEY? BC IF,
                                        \ Vorige data verwerkt?
        IE: .0 CLR,
                                        \ Ja, EX0-interrupt uit
        TH1: -215. NIP # MOV,
                                        \ Tijd tot eerste meting, 430 us
        TL1: -215. DROP # MOV,
                                        \ plus 24 us = 454 us
        TCON: .6 SETB,
                                        \ Start timer-1
       RC\# 1 \# MOV,
                                        \ Fake half ontvangen bit,
       ADR RC-CONTROLE SETB,
                                        \ dit fakebit is hoog!!!
    THEN,
   PSW: POP,
   RETI,
END-CODE
CODE RC-SETUP ( -- )
                                        \ Installeer decoderhardware
    TMOD: $10 # ORL,
                                        \ Zet Timer-1 in mode-1
    IE: .7 SETB,
                                        \ Zet interrupts aan
   TCON: .6 CLR,
                                        \ Stop timer-1
    IE: .3 SETB,
                                        \ Sta interrupts van timer-1 toe
    TCON: .0 SETB,
                                        \ EX0 actief op lage flanken
    IE: .0 SETB,
                                        \ EX0-interrupt aan
                                        \ Geen RC5-commando ontvangen
    ADR RC-KEY? CLR,
   ADR RC-IN SETB,
                                        \ P3.2 is de ingang
   RET,
END-CODE
```

```
\ Ontvang een RC5-databyte, x is een code uit de gedefinieerde
\ RC5-codecommandoset. Bekijk hiervoor de RC5-devicedocumentatie.
: RC-KEY ( -- x )
                                       \ Wacht op een commando-databyte
   BEGIN RC-KEY? UNTIL
                                        \ Wacht
   FROM RC-DATA
                                        \ Laat \ commando \ achter
   CLEAR RC-KEY?
                                        \ Sta volgend commando toe
    ;
$B5 BIT-SFR LED
                                        \ Een LED op P3.5
$B7 BIT-SFR LAMP
                                        \ Het solidstate relais op P3.7
: SECONDEN
                   (u--)
                                        \ Wacht u seconden
   FOR 4 FOR 250 MS NEXT NEXT
    ;
: RC-LAMP
                    ( -- )
                                        \ Bestuur een lamp via RC5
   SETUP RC-SETUP SET LAMP
    CLEAR LED 2 SECONDEN SET LED
                                        \ Toon opstarten
   BEGIN
       RC-KEY 12 = IF
                                        \ Aan-/uittoets ontvangen ?
            TOGGLE LED TOGGLE LAMP
                                        \ Ja, schakel lamp aan/uit
            250 MS RC-KEY? IF
                                        \ Wacht even en gooi tweede
                                        \ code weg indien ontvangen
              RC-KEY DROP
            THEN
       THEN
   AGAIN
    ;
\ Installeer interrupt routines
LEES-BITTF1-VECSET-VECTORRC-STARTIE0-VECSET-VECTORRC-LAMPRESET-VECSET-VECTOR
                                        \ LEES-BIT via interrupt timer-1
                                        \ RC-START geactiveerd door P3.2
                                       \ RC-LAMP start op na een reset
```

\ Einde

# Zender Afstandsbediening

#### Zie appendix RC5

In dit ontwerp wordt het RC5-protocol (Remote Control 5) van Philips toegepast. In de appendix betreffende RC5 wordt dit protocol beschreven.

De zender commandeert een ontvanger door middel van infraroodsignalen, zoals een TV of de ontvanger uit project ESW-14, met behulp van het RC5-protocol. Het signaal wordt uitgezonden door een IR-LED. De verzonden code is die van de aan-/uit- (stand-by-) knop.

De drie niveaus van het ontvangen signaal worden als volgt verwerkt.

- Het 36-kHz-basissignaal wordt door de hardware gegenereerd. Het wordt vooral beïnvloed door R3 in het schema.
- De bitstroom voor het bedienen van een TV bevat 5 systeembits met de waarde 0 gevolgd door 6 commandobits. De aan-/uitknop heeft de commandowaarde 12 decimaal:



De fieldbit (of tweede startbit) is 1. De controlbit wordt vast op 1 gesteld.

• De bitstroom wordt uitgezonden zolang de toets is ingedrukt. De tijd tussen twee bitstromen noemen we de herhalingstijd. Deze wordt bepaald door de routine **FLITS** en bedraagt ongeveer 1 seconde. De controlbit wordt niet veranderd.

#### Schema



### Bedradingsplan







Referentie	Omschrijving
R1,R2	weerstand 8,2 k $\Omega$ 1%
R3	weerstand 39,4 k $\Omega$ 1%
R4	weerstand 470 $\Omega$
R5	weerstand 1,2 $\Omega$
R6	weerstand 10 $\Omega$
R7	weerstand 330 $\Omega$
R8	weerstand 10 k $\Omega$
D1,D2	diode 1N4148
D3, D4	infrarode LED type LD274 Ø 5 mm
D5	rode LED
C1	condensator 2,2 nF MKT
C2	elco 1000 μF 16V radiaal
U1	IC CD4093 of equivalent
T1	transistor BC517
-	14-pins IC-voet
<b>S</b> 1	drukschakelaar

### Bouwaanwijzing

Soldeer eerst de weerstanden, de LED en de drukknop op de linkerzijde van de print. Leg aan de onderzijde de volgende verbindingen: O2 naar P2, M4 naar L6, J2 naar J8, I1 naar I4 en I6 naar I8.

Plaats aan de rechterzijde eerst de IC-voet voor de 4093. Leg dan een doorverbinding aan de componentenzijde van D17 naar A16. Plaats de volgende componenten liggend op de print: R3 (39,4 k $\Omega$ ), R4 (470  $\Omega$ ) en R5 (1,2  $\Omega$ ).

De weerstanden R1 en R2 (8,2 k $\Omega$ ) en de dioden D1 en D2 worden staand gemonteerd. Plaats vervolgens de transistor. De IR-LED's D3 en D4 worden liggend naar de buitenzijde gemonteerd. De weerstand R6 (10  $\Omega$ ) moet aan de onderzijde van de print gemonteerd worden, direct achter de diode van de 9-volts voeding op de AT51-print.

Aan de rechterzijde moeten nu de volgende doorverbindingen worden gelegd:

- F17 via E17 en C15 naar B15 met blank draad

- C17 naar C18
- C19 naar C20
- B21 via C21, D20, E20 en H20 naar I20. De verbinding tussen D20 en I20 isoleren
- F20 naar G20 en G21
- E21 naar E22
- D22 naar C23
- F22 naar H22
- I22 naar J22
- K22 naar I22
- O18 naar P21 en P22
- N18 naar N23
- als laatste de +5 volt geïsoleerd van O15 naar J20.

Opmerking: Ter controle van het functioneren van de IR-LED is voor D3 een rode LED gekozen. Deze geeft een zichtbare indicatie van het signaal. Als de schakeling goed werkt kan diode D3 vervangen worden door een infrarode LED zoals D4.

### Software

De software van de RC5-zender bestaat voornamelijk uit wachtlussen, afgewisseld door het hoog en laag zetten van de uitgang. Deze uitgang schakelt de met de 4093 opgebouwde 36-KHz-oscilator en daarmee de IR-led aan of uit. Omdat de timing zeer precies het voorgeschreven RC5-bitpatroon moet volgen is bij iedere regel genoteerd wat de uitvoeringstijd van de instructie(s) is. De routine **RC-EMIT** maakt een complete RC5-bitstroom met daarin eerst de gewenste ontvangercode en daarna het 6-bits RC5-commando.

**RC-ZENDER** zendt het commando '12' (aan/uit) en laat daarna gedurende een seconde een LED flitsen. Hierdoor wordt er maximaal 1 commando per seconde verstuurd, zoals opgegeven in de RC5-specificatie.

\ RC5-zender voor een 6-MHz-kristal. Lengte 154 bytes.

```
89C2051 TARGET
                                         \ Compileer voor AT89C2051
                                         \ RC5-uitgangsbit
$B7 BIT-SFR UITGANG
$B3 BIT-SFR INGANG
                                         \ RC5-schakelaaringang
$B2 BIT-SFR LED
                                          \ RC5-LED-uitgang
                                         \ RC5-zendertype (hier TV)
$E0 CONSTANT RC-TYPE
                     ( -- )
CODE HALVE-BIT
                                         \ Wacht 878 us
    R1: 2 # MOV,
    DO,
        R2: 107 # MOV,
        DO, R2: LOOP,
    R1: LOOP,
    RET,
END-CODE
: EEN
                     ( -- )
                                          \ Zend hoge bit 1768 us
    CLEAR UITGANG
                                          \ 4 us
    HALVE-BIT
                                          \ 878 us
    SET UITGANG
                                          \ 4 us
    HALVE-BIT
                                          \ 878 us
    ;
                                          \ 4 us
: NUL
                     ( -- )
                                         \ Zend lage bit 1752 us
    SET UITGANG
    HALVE-BIT
    CLEAR UITGANG
    HALVE-BIT
    ;
```

```
CODE ZEND-RC ( -- )
                                   \ Zend R3 bits uit accu: R3x1786/1782 us
   DO,
       A: RLC,
                                    \ 2 us
       CS IF,
' EEN CALL,
                                    \ 4 us
                                    \ 1768 us
       ELSE,
                                   \ 4 us
         ' NUL CALL,
                                    \ 1768 us
       THEN,
   R3: LOOP,
                                    \ 4 us
   RET,
                                    \ 8 us
END-CODE
CODE RC-EMIT
               ( x -- )
                                    \ Zend 'x' naar een tv of videorecorder
   A: @R0 MOV,
                                    \ Het commando
   A: RL,
                                    \ Sla twee hoge bits over
   A: RL,
   R4: A: MOV,
                                    \ Bewaar commando
   R0: INC,
   A: RC-TYPE # MOV,
                                    \ Systeemdata (zender type)
   R3: 8 # MOV,
                                    \ Zend ook de startbits
   ' ZEND-RC CALL,
                                    \ 1786x8+8 us
   A: R4: MOV,
   R3: 6 # MOV,
                                    \ Zend een 6-bit RC5-commando
   ' ZEND-RC CALL,
                                    \ 1786x6+8+10 us
   ADR UITGANG CLR,
                                    \ Zet de 36-kHz-oscillator uit
   RET,
                                    \ 8 us
END-CODE
         (u--)
: 10MS
                                   \ Wacht u x10 millisec.
  FOR 10 MS NEXT
   ;
: FLITS
                 ( -- )
                                    \ Laat LED flitsen
  CLEAR LED 50 10MS
   SET LED 50 10MS
   ;
: RC-ZENDER (--)
                               \ Zend een RC-startcode
   SETUP CLEAR UITGANG
                                    \ Uitgang uit
   BEGIN
      FLITS
                                    \ Toon activiteit
      BEGIN INGANG 0= UNTIL
                                    \ Wacht op toetsaanslag
      12 RC-EMIT
                                    \ Zend aan-/uitcode
   AGAIN
   ;
' RC-ZENDER RESET-VEC SET-VECTOR \ RC-ZENDER start op na een reset
```

# 7-Segmentdisplays

Een viertal 7-segment-LED-displays is geplaatst op een afzonderlijk printje: EHW-16D. De displays worden elk aangestuurd via een schuifregister met slechts drie draden. Voor de tekens die naar het display worden gestuurd vindt de decodering in de software plaats. De tekens worden per display naar de gewenste combinatie van LED's omgezet met behulp van de tabel **CLJFERS**. Met de constanten uit deze tabel kunnen waarden in alle talstelsels, zowel binair als decimaal en hexadecimaal, worden gemaakt. De decimale punt is wel in de software opgenomen maar wordt in het voorbeeldprogramma niet gebruikt.

### Schema



Dit is het schema van het deel op de AT51-print. Dat wordt aangesloten op EHW-16D, de print met de 7-segmentdisplays. Zie de appendix over EHW-16D. De aansluitingen zijn als volgt:

poortbit	J1 pin	naam
-	1	GND
P1.7	2	STR
P1.6	3	DATA
P1.5	4	CLK
P1.4	5	LED

## Bedradingsplan



#### Onderdelenlijst

Referentie	Omschrijving
R1	weerstand 4,7 k $\Omega$
T1	transistor BC559B
J1	5-polige male headerconnector
B1	EHW16D-print (zie bijlage)

Voor dit project is een printplaat gemaakt met ruimte voor vier displays. Verander zelf de software als je meer of minder displays wilt gebruiken. Eventueel kunnen meer printplaten aan elkaar worden gekoppeld.

### Bouwaanwijzing

Soldeer de volgende componenten op de print:

- Connector J1 op K17 .. O17
- Weerstand R1 op K15; steek de andere aansluiting door G16
- Plaats transistor T1 in F17, G17 en H17; soldeer de weerstand aan G17; soldeer de collector van T1 aan K17
- Bedraad nu aan de soldeerzijde van de print: L15 aan L17; M15 aan M17; N15 aan N17; O17 aan O23;
- Steek een stukje geïsoleerd draad aan de componentzijde door F19 en soldeer dat aan de emitter van T1; soldeer het andere eind in Q19.

EHW-16A is nu klaar om te verbinden met displayprint EHW-16D. Dit gebeurt het best door een 5-polige female headerconnector te solderen aan een stukje 5-aderig flatcable en die te verbinden met de EHW-16D-print. Let goed op aan welke kant pin-1 zit; dit is de massa-aansluiting!

### Software

Bij toepassing van een ander aantal displays behoeft alleen de constante **#DIGITS** aangepast te worden. Kiezen tussen het decimaal en het hexadecimaal talstelsel gebeurt met de colondefinities **DECIMAAL** en **HEX**. In het voorbeeldprogramma worden deze aangeroepen in de definitie **WIJZIG-BASE**.

\ Compileer voor AT89C2051

```
\ Aansturing van vier 7-segmentdisplays, codelengte 487 bytes.
```

```
89C2051 TARGET
```

\ u\*8 bits uitvoer gebruikt altijd 3 I/O-bits

```
ASSEMBLER
P1: .4 BIT-SFR LED
                                        \ Leds tussen display aan/uit
P1: .5 BIT-SFR KLOK
                                        \ Klok
                                                           (13=Clock )
P1: .6 BIT-SFR DATA
                                        \ Data bitstroom
                                                           (14=Data )
P1: .7 BIT-SFR KLAAR
                                        \ Data geldig
                                                           (15=Strobe)
WORK
: SCHRIJF-BIT
                    ( flag -- )
                                        \ Schrijf een bit naar de uitgang
    TO DATA CLEAR KLOK SET KLOK
    ;
: TOON
                    ( -- )
                                        \ Zet data op de uitgangen
    SET KLAAR CLEAR KLAAR
    ;
: >DRIVER
                    (x -- )
                                        \ Schrijf x naar display
    8 FOR
                                        \ display driver
       DUP 128 AND SCHRIJF-BIT 2*
    NEXT
    DROP
    ;
```

```
\ Maak cijfers voor een 7-segmentdisplay via de 74HC4094 0 t/m F
\ De digits staan in omgekeerde volgorde op de stack bij CONSTANTS !!!
\ Layout van 7-segmentdisplay +-A-+
                             F B
\
\ Uitsturing 74HC4094
                             +-G-+
\ PBCAFDGE
                             E C
\ Q87654321
                             +-D-+
                                                \ opbouw van tekens
                       %01100111
  %00011011
             %00011111
                                    %00011101
                                                \ F , E , d , C
 %00101111 %01111011 %01111110 %01111111
                                               \b,A,9,8
  %01110000 %00111111
                         %00111110 %01101010
                                              \7,6,5,4
  %01110110 %01010111 %01100000 %01111101
                                              \3,2,1,0
 16 CONSTANTS CIJFERS
VARIABLE PUNT
                                      \ Onthoud decimale punt
: ZET-PUNT
                  (x1u--x2)
   FROM PUNT = $80 AND OR
   ;
· >DTSPLAY
                   ( adr u -- )
                                      \ Stuur u getal digits van RAM-adres
                                      \ naar het 7-segment display
   FOR
       @+ 16 UMIN CIJFERS
                                      \ Van binair naar 7-segment
       I' ZET-PUNT >DRIVER
   NEXT
   DROP TOON
                                      \ en laat ze zien !
    ;
4 CONSTANT #DIGITS
                                      \ Aantal gebruikte 7-segmentdisplays
                                      \ en 74HC4094 chips.
                                      \ Type getal op 7-segmentdisplays
: RTYPE
                   (adr u -- )
   #DIGITS OVER U> IF
                                      \ Getal is minder dan #DIGITS lang ?
       #DIGITS OVER - FOR
                                      \ Ja, vul aan met lege digits
          0 OVER I' + ZET-PUNT >DRIVER
       NEXT
       DUP 0= IF 2DROP EXIT THEN
                                     \ Stop als u nul is !!!
   THEN
    #DIGITS UMIN >DISPLAY
                                      \ Type getal van max. #DIGITS
    ;
: 10MS FOR 10 MS NEXT ; ( u -- )
                                     \ Wacht u maal 10 ms
\ Getal uitvoer routines
                                      \ Importeer enkele macro's als woord
ATOM UM/MOD
VARIABLE BASE
                                      \ Basis voor getalconversie
VARTABLE HLD
                                      \ Werkadres in getalconversiebuffer
$80 CONSTANT NUM
                                      \ Startadres van getalconversiebuffer
                                      \ Alleen geldig voor de AT89C2051
          ( d1 u1 -- u2 d2 )
                                      \ Deel dubbel met teken door single
: M/MOD
   >R 0 R@ UM/MOD R> SWAP >R UM/MOD R> \ zonder teken.
\ Primitieven voor getaluitvoer
\ Zet digit in de uitvoer buffer
: HOLD
          DECR HLD FROM HLD ! ;
                                                     ( digit -- )
\ Initialiseer de uitvoer buffer
: <#
          NUM TO HLD ;
                                                      ( -- )
\ Beëindig getalconversie, geef addr en lengte u van de string in de buffer
           2DROP FROM HLD NUM FROM HLD - ;
                                                     ( d -- addr u )
: #>
\ Zet een cijfer van d1 in de uitvoerbuffer, d2 blijft over
                                                     ( d1 -- d2 )
: # FROM BASE M/MOD ROT HOLD ;
```

```
\ Verzamel alle cijfers in de cijferstring onder (NUM)
                                                       (d1 -- d2)
: #S BEGIN # 2DUP OR 0= UNTIL ;
\ Woorden voor getaluitvoer:
\ Toon ud in een veld van vrij formaat.
: DU. <# #S #> RTYPE ;
                                                       ( ud -- )
\ Toon u in een vrij formaat veld.
          0 DU.;
                                                       (u -- )
: U.
\ Decimale getalconversie
: DECIMAL #10 TO BASE ;
                                                       ( -- )
\ Hexadecimale getalconversie
: HEX
         #16 TO BASE ;
                                                       ( -- )
\ Einde getaluitvoer
\ Voorbeeld een teller met drie 7-segmentdisplays
: WIJZIG-BASE ( -- )
   FROM BASE #10 = IF
                                       \ Conversiebasis decimaal ?
                                       \ Ja, maak hem hexadecimaal
       HEX
   ELSE
       DECIMAL
                                       \ Nee, maak hem decimaal
   THEN
   ;
: KNIPPER
              ( -- )
                                       \ Toon opstarten
   3333. DU. 50 10MS
4444. DU. 50 10MS
    ;
                                       \ Binaire teller op 7-segmentdisplays
: TELLER
               ( -- )
   SETUP DECIMAL TOON KNIPPER
                                       \ Initialiseer voorbeeld
   1 TO PUNT
                                       \ Stel decimale punt in
   0
                                       \ Zet teller op de stack
   BEGIN
       BEGIN
           1+ DUP U.
                                       \ Verhoog teller en toon getal
           20 10MS
                                       \ Wacht 200 ms
       DUP 0= UNTIL
                                       \ Klokje rond ?
       WIJZIG-BASE
                                       \ Ja, ga van decimaal naar hex en andersom
   AGAIN
                                       \ Begin opnieuw
    ;
' TELLER
           RESET-VEC SET-VECTOR
                                       \ TELLER start op na een reset
```

\ Einde

# **LCD-aansturing**

Hardware EHW-17 Software ESW-17

Zie appendix LCD

In dit project wordt een LCD (Liquid Crystal Display) met twee regels van 16 tekens toegepast. Het LCD heeft een Hitachi-chip aan boord voor de aansturing. Kenmerkend voor deze chip is dat die intern een buffer van 40 tekens per regel heeft, ook al is het display minder dan 40 tekens breed. Om van regel te wisselen wordt de routine **LCD-CR** gebruikt.

### Schema



De aansluitingen van een karakter LCD :		
pinnr	naam	functie
1	VSS	massa
2	VDD	+5V-voedingsspanning
3	V0	contrastspanning 05 V (soms -5 5 V)
4	RS	registerselectie
5	R/W	lees/schrijf
6	Е	enable
7	DB0	databit-0
8	DB1	databit-1
9	DB2	databit-2
10	DB3	databit-3
11	DB4	databit-4
12	DB5	databit-5
13	DB6	databit-6
14	DB7	databit-7

# Bedradingsplan



P1	Potmeter 10 k $\Omega$ (Piher 10mm liggend)
J1	male headerconnector
J2	female header (optioneel voor montage op LCD)
U1	LCD met of zonder achtergrondverlichting
R1, R2	weerstand 22 k $\Omega$

## Bouwaanwijzing

Soldeer de potmeter en de 14-polige connector op de rechterzijde van de print. Leg de doorverbindingen aan de onderzijde. Monteer een stukje lintkabel met een 14-polige female connector aan het LCD.

Na het aanzetten verschijnt op de eerste regel even een zwarte balk. De contrastspanning kan met de potmeter zo worden ingesteld dat de karakters duidelijk afsteken tegen de achtergrond.
### Software

In het voorbeeldprogramma wordt een stilstaande tekst getoond. Het display wordt met 4 databits aangestuurd. Om de chip op deze 4-bitmode in te stellen moet de routine **LCD-SETUP** strikt worden aangehouden.

\ LCD-uitvoer met vier databits. Lengte 422 bytes.

```
89C2051 TARGET
                                       \ Compileer voor AT89C2051
ATOM MS
                                       \ Importeer de routine MS
40 CONSTANT B/L
                                       \ Regelbufferlengte van het display
16 CONSTANT C/L
                                       \ Regellengte van LCD
2 CONSTANT L/S
                                       \ Regels per LCD
VARIABLE UIT
                                       \ Huidige cursorpositie
FLAG DATA?
                                       \ LC datatype, true is een karakter
ASSEMBLER
P1: SFR POORT
                                       \ I/O-poort met het LCD
P1: .4 BIT-SFR GELDIG
                                       \ Data-geldig bit
P1: .5 BIT-SFR SELECTEER
                                       \ Data is karakter of instructiebit
WORK
              ( nibble -- )
: >LCD
    $F AND TO POORT DATA? TO SELECTEER \ Stuur nibble op, karakter of data
   SET GELDIG CLEAR GELDIG
                                      \ Zet data op LCD
    ;
: LCD-CHAR
              ( kar -- )
                                      \ Stuur kar naar LCD
   SPLIT >LCD >LCD
    ;
: LCD-INSTR ( byte -- )
CLEAR DATA? LCD-CHAR SET DATA?
                                       \ Stuur data naar LCD
    ;
: LCD-PAGE
              ( -- )
                                       \ Maak LCD schoon en zet de
   1 LCD-INSTR CLEAR UIT 2 MS
                                       \ cursor in de linker bovenhoek.
    ;
: LCD-SETUP
              ( -- )
   CLEAR DATA? 16 MS
                                       \ Naar instructiemode, wacht en
    3 FOR %0011 >LCD 5 MS NEXT
                                       \ Stel de 8-bits-interfacemode in
    %0010 >LCD
                                       \ Tenslotte de 4-bits-interfacemode
    $28 LCD-INSTR
                                       \ 4-bits-interf, 2 regels, 5x7 bits kar.
    $08 LCD-INSTR LCD-PAGE
                                       \ Display uit en schoon
    $06 LCD-INSTR
                                       \ Displayrichting naar rechts
    $0C LCD-INSTR
                                       \ Display aan & cursor uit
    ;
: LCD-EMIT
              ( kar -- )
                                       \ Stuur een karakter naar het LCD
    INCR UIT LCD-CHAR
    ;
: LCD-TYPE ( $addr u -- )
                                       \ Stuur u karakters vanaf het ROM-adres
   FOR COUNT LCD-EMIT NEXT 2DROP
                                       \ $addr naar het LCD.
    ;
: LCD-SPACE
              ( -- )
                                       \ Stuur een spatie naar het LCD.
   BL LCD-EMIT
    ;
: LCD-SPACES ( u -- )
                                      \ Stuur u spaties naar het LCD.
   FOR LCD-SPACE NEXT
    ;
```

```
: LCD-CR ( -- )
C/L FROM UIT - 0< IF
                                    \ Stuur CR naar het LCD.
     $80 LCD-INSTR CLEAR UIT
                                    \ Staat cursor op regel-2?
                                     \ Ja, naar regel-1
   ELSE
    $C0 LCD-INSTR C/L TO UIT \ Nee, naar regel-2
   THEN
   ;
\ Het LCD-voorbeeld
ATOM INLINE$
                                     \ Maak ." bruikbaar
: TYPE LCD-TYPE ;
: 100MS FOR 100 MS NEXT ; ( u -- ) \ Wacht u x100 millisec
: REGEL-VOL? ( -- )
FROM UIT 16 =
                                     \ Ga indien nodig naar de volgende regel
                                     \  \  Regel-1 of regel-2
   FROM UIT 32 = OR IF
                                     \ gevuld?
     LCD-CR
                                     \ Ja, naar andere regel
   THEN
   ;
: LCD-DEMO
             (n--)
                                     \ Een voorbeeld
   SETUP CLEAR MASSA CLEAR LED
   100 MS SET LED LCD-SETUP
   BEGIN
       ." 'Egel LCD demo" LCD-CR
                                     \ Titel
       ." (C) Forth-gg"
       30 100MS LCD-PAGE
                                     \ Wacht en maak het scherm schoon
       0 0 DO
                                     \ Toon de hele karakterset
          I LCD-EMIT 100 MS
          REGEL-VOL?
                                     \ Ga indien nodig naar nieuwe regel
       LOOP
       10 100MS LCD-PAGE
                                     \ Wacht en maak scherm schoon
   AGAIN
   ;
' LCD-DEMO RESET-VEC SET-VECTOR \ LCD-DEMO start op na een reset
```

## I<sup>2</sup>C-inleiding

Zie appendix I<sup>2</sup>C

٠	I <sup>2</sup> C-UIT	EHW18A/I	ESW18
•	I <sup>2</sup> C-I/O	EHW18A/I	ESW19
•	I <sup>2</sup> C-EEPROM	EHW18A/I	ESW20
•	I <sup>2</sup> C-KLOK	EHW18A/I	ESW21

I<sup>2</sup>C (I-kwadraat-C) is een methode voor dataoverdracht tussen I<sup>2</sup>C-componenten. Alle componenten worden voor de dataoverdracht onderling met twee draden verbonden, de SDA-lijn voor de data en de SCL-lijn voor de klokpulsen. Nadere toelichting in appendix I<sup>2</sup>C.

De in deze vier projecten toegepaste I<sup>2</sup>C-componenten zijn:

- PCF8574; een I/O-expander voor projecten I<sup>2</sup>C-IN (ESW-18) en I<sup>2</sup>C-I/O (ESW-19)
- 24C02; een 8-pins-EEPROM voor project I<sup>2</sup>C-EPROM (ESW-20)
- PCF8583; een klok/wekker/kalender voor project I<sup>2</sup>C-KLOK (ESW-21)

Voor deze projecten is de printplaat EHW-18I ontworpen waarop ze alle vier gebouwd kunnen worden. Op deze print kunnen een extra PCF8574 (I/O-expander) en een PCF8591 (ADC/DAC-chip) nog een plaatsje vinden. Het schema, de onderdelenlijst en het bedradingsschema (layout) worden beschreven in de documentatie bij de printplaat. De projecten kunnen natuurlijk ook afzonderlijk worden gebouwd.

De printplaat EHW-18I kan via een kabeltje gekoppeld worden met een AT51-printje. Hieronder wordt dit printje EHW-18A beschreven. De vijf draden van het kabeltje zijn als volgt ingedeeld:

1	+ 5V
2	massa
3	SDA (data)
4	SCL (klok)
5	/INT (interrupt)

### Schema



## Bedradingsplan



## Bouwaanwijzing

J1

Soldeer connector J1 op de printplaat. De verbindingskabel naar de EHW-18I-print kan direct gesoldeerd worden, maar hier worden female connectors gebruikt zodat de verbinding makkelijk kan worden losgenomen.

In de volgende projecten over I<sup>2</sup>C wordt steeds terugverwezen naar dit bedradingsplan.

EHW-18I print (zie bijlage)

## I<sup>2</sup>C-UIT

#### Zie I<sup>2</sup>C-inleiding

Door de microcontroller wordt een databyte verzonden naar een I<sup>2</sup>C-component. De SDA-lijn voor de dataoverdracht is aangesloten op poort-P3.4 en de SCL-lijn op P3.5. De gebruikte PCF8574 (I/O-expander) ontvangt de 8 bits van de databyte serieel en zet ze op zijn parallelle poort (P0 t/m P7). Deze poort wordt dus als uitgang gebruikt. Op vier pinnen zijn LED's aangesloten. Het programma toont hierop een looplicht.

Het basisadres van de I/O-expander PCF8574 is \$40. De laagste drie bits van het adres worden ingesteld met de 3 ingangen A2, A1 en A0 op de PCF8574. Op elk van de drie adresingangen A2, A1 en A0 wordt een jumper geplaatst. Binair geschreven zijn voor het chipadres (=\$40) de vier hoogste bits %0100 (binair). De drie volgende bits zijn A2=0, A1=0 en A0=0 (jumpers geplaatst). Het 7-bits-chipadres wordt daarmee %0100 000. De laagste bit (voor "schrijven naar") R/W=0 wordt toegevoegd. Daarmee ontstaat het te verzenden adresbyte %0100 0000.

Let op: Het is ook mogelijk om i.p.v. de PCF8574 de PCF8574A toe te passen. Beide chips zijn op het basisadres na gelijk aan elkaar. Het basisadres van de PCF8574A is \$70. Bepaal zelf wat er aan de jumpers op de adresingangen en in de software moet worden gewijzigd.

De vier LED's worden bestuurd via de vier hoogste bits in de databyte. De vier laagste bits worden alle hoog gezet.

Het volgende diagram toont de communicatie op de I<sup>2</sup>C-bus tussen master en slave. De (re-)actie van de slave staat in de grijze blokken. Voor R/W lees "Read, Not Write".

Start	Slave-adres	R/W	Ack.bit	Databyte	Ack.bit	Stop
S	a6 a5 a4 a3 a2 a1 a0	0	0	b7 b6 b5 b4 b3 b2 b1 b0	0	Р

gezonden door master

gezonden door slave

### Schema

Het EHW-18A-deel is hetzelfde als beschreven in de I<sup>2</sup>C-inleiding. Op de EHW-18I-print is de I/O-expander PCF8574 geplaatst.



### Bedradingsplan, onderdelenlijst & bouwaanwijzing

EHW-18A: zie I<sup>2</sup>C-inleiding. Voor de EHW-18I: zie appendix.

### Software

De I<sup>2</sup>C-primitieven zijn in de listing overgenomen uit de bibliotheek in de subdirectory LIB. \ I<sup>2</sup>C-uitvoer met een PCF8574. Codelengte 234 bytes. 89C2051 TARGET \ Code draait op een 89C2051 \ I<sup>2</sup>C-primitieven voor een 6-MHz-kristal ASSEMBLER P3: .4 BIT-SFR SDA \ Seriële datalijn P3: .5 BIT-SFR SCL \ Seriële kloklijn WORK \ Device-adres-0 van gebruikte chip \$40 CONSTANT I/O-ID \ Hoe zat dat ook weer voor een PCF8574A? \ Algemene I<sup>2</sup>C-primitieven : SETUP-I2C ( -- ) \ Setup I<sup>2</sup>C-I/O-bits SET SDA SET SCL ; : START-BIT \ Genereer startbit (startconditie) ( -- ) SET SDA SET SCL CLEAR SDA CLEAR SCL ; : STOP-BIT ( -- ) \ Genereer stopbit (stopconditie) CLEAR SDA SET SCL SET SDA ; : ACK? ( -- vlag ) \ Test binnenkomend ack.bit. De vlag is SET SDA SET SCL SDA 0= CLEAR SCL \ true als er een ack is ontvangen ; : BYTE-UIT ( byte -- ) \ Stuur byte en test ack 8 FOR DUP 128 AND TO SDA 2\* \ **#128=\$80** SET SCL CLEAR SCL NEXT DROP ACK? DROP \ Geen ack, genereer dan een stopbit ; \ Uitvoerroutine voor PCF8574(A) chips : !BYTE ( byte chip -- ) START-BIT

```
$FE AND BYTE-UIT
                                     \ Stuur schrijfadres naar chip
BYTE-UIT
                                     \ Daarna de databyte
STOP-BIT
                                     \ Klaar
;
```

 $\ \$  Onze eerste I<sup>2</sup>C-toepassing: een looplicht

```
\ Stuur data naar PCF8574 op adres-0
: NAAR-LEDS ( byte -- )
   AAR-LEDS ( byte -- )
INVERT $0F OR I/O-ID !BYTE
                                        \ De lage 4 bits zijn altijd hoog (!)
    ;
: KNIPPER ( -- )
255 NAAR-LEDS 250 MS
0 NAAR-LEDS 250 MS
                                        \ Toon het opstarten
    ;
: LOOPLICHT ( -- )
                                        \ Toon een looplicht op de LED's
    SETUP SETUP-12C KNIPPER
    BEGIN
        4 0 DO
           1 I 4 + LSHIFT NAAR-LEDS \ Zet een LED aan
           100 MS
                                        \ Wacht 100 millisec
       LOOP
    AGAIN
    ;
' LOOPLICHT RESET-VEC SET-VECTOR \ LOOPLICHT start op na een reset
```

## I<sup>2</sup>C-I/O

#### Zie I<sup>2</sup>C-inleiding

In dit project wordt dezelfde I<sup>2</sup>C-component PCF8574 (I/O-expander) gebruikt als in het voorgaande project ESW-18. De vier hoogste bits van P0 t/m P7 sturen weer de LED's aan, dus er wordt naar het IC geschreven. De vier laagste bits zijn aangesloten op schakelaars. De stand daarvan wordt via de I<sup>2</sup>C-bus gelezen door de microcontroller.

Het basisadres is \$40 (\$70 voor de PCF8574A) en er zijn drie jumpers geplaatst. Voor het schrijven: R/W=0; adresbyte %0100 0000 (\$40) of %0111 0000 (\$70) Voor het lezen: R/W=1; adresbyte %0100 0001 (\$41) of %0111 0001 (\$71)

Voor het schrijven naar de LED's, de uitvoer dus, is dezelfde figuur geldig als in project ESW-18.

Start	Slave-adres	R/W	Ack	Databyte	Ack	Stop
S	a6 a5 a4 a3 a2 a1 a0	0	0	b7 b6 b5 b4 b3 b2 b1 b0	0	Р

gezonden door master

gezonden door slave

Voor het lezen van de stand van de schakelaars, de invoer dus, is de figuur als volgt:

Start	Slave-adres	R/W	Ack	Databyte	Nack	Stop
S	a6 a5 a4 a3 a2 a1 a0	1	0	b7 b6 b5 b4 b3 b2 b1 b0	0	Р

De databyte wordt gelezen door de master. Die geeft normaal gesproken een Acknowledge-bit terug, maar omdat dit het laatst opgevraagde databyte is (er is er maar een) geeft de master een NoAcknowledge- (of Negative acknowledge-)bit op de SDA-lijn.



### Schema



### Software

De I<sup>2</sup>C-primitieven zijn ook hier weer in de listing opgenomen.

```
\ I<sup>2</sup>C-in- en uitvoer met PCF8574, totale codelengte 304 bytes
89C2051 TARGET
                                         \ Code draait op een 89C2051
\ I<sup>2</sup>C-primitieven voor een 6-MHz-kristal
ASSEMBLER
P3: .4 BIT-SFR SDA
                                         \ Seriële datalijn
P3: .5 BIT-SFR SCL
                                         \ Seriële kloklijn
WORK
$40 CONSTANT I/O-ID
                                         \ Device-adres-0 van gebruikte chip
\ Algemene I<sup>2</sup>C-primitieven
: SETUP-I2C
               ( -- )
                                         \ Setup I<sup>2</sup>C-I/O-bits
    SET SDA SET SCL
    ;
: START-BIT
                                         \ Genereer startbit (startconditie)
               ( -- )
    SET SDA SET SCL CLEAR SDA CLEAR SCL
    ;
: STOP-BIT
                                         \ Genereer stopbit (stopconditie)
               ( -- )
    CLEAR SDA SET SCL SET SDA
    ;
: ACK?
                                         \ Test binnenkomende ack.bit. De vlag is
                ( -- vlag )
    SET SDA SET SCL SDA 0= CLEAR SCL \ true als er een ack ontvangen is
    ;
: BYTE-UIT
                                         \ Stuur byte en test ack
                ( byte -- )
    8 FOR
        DUP 128 AND TO SDA 2*
        SET SCL CLEAR SCL
    NEXT
    DROP ACK? 0= IF STOP-BIT THEN \ Geen ack, genereer dan een stopbit
    ;
```

```
: BYTE-IN ( -- byte )
                                   \ Ontvang byte
   0 8 FOR
     SET SCL 2* SDA 1 AND OR
      CLEAR SCL
   NEXT
   SET SDA SET SCL CLEAR SCL \ Geef no-ack
   ;
\ Uitvoerroutine voor PCF8574(A) chips
: !BYTE
              ( byte chip -- )
   START-BIT
   $FE AND BYTE-UIT
                                     \ Stuur schrijfadres naar chip
   BYTE-UIT
                                     \ Daarna de databyte
   STOP-BIT
                                     \ Klaar
   ;
\ Lees routine voor PCF8574(A) chips
: @BYTE
              ( chip -- byte )
   START-BIT
   1 OR BYTE-UIT
                                     \ Stuur leesadres naar chip
   BYTE-IN
                                     \ Lees databyte van ingang
   STOP-BIT
   ;
\ De tweede I<sup>2</sup>C-applicatie is een looplicht met variabele snelheid
: 10MS
             (u--)
                                     \ Wacht u x10 millisec.
   FOR 10 MS NEXT
   ;
: INGANG
             ( -- +n )
                                     \ Lees data van PCF8574 op adres-0
   I/O-ID @BYTE INVERT $F AND 1+ 2* \ +n is een getal tussen 2 en 32
   ;
: NAAR-LEDS
             ( byte -- )
                                     \ Stuur data naar PCF8574 op adres-0
   INVERT $0F OR I/O-ID !BYTE
                                     \ De lage 4 bits zijn ingangen
   ;
: KNIPPER
            ( -- )
                                     \ Signaleer opstarten
   -1 NAAR-LEDS 25 10MS
0 NAAR-LEDS 25 10MS
   ;
: LOOPLICHT ( -- )
                                     \ Toon een looplicht op de LED's
   SETUP SETUP-12C KNIPPER
   BEGIN
      4 0 DO
          1 I 4 + LSHIFT NAAR-LEDS
                                     \ Zet een LED aan
          INGANG 10MS
                                     \ Wacht 20 tot 320 millisec
      LOOP
   AGAIN
   ;
' LOOPLICHT
                RESET-VEC
                              SET-VECTOR \ LOOPLICHT start op na een reset
```

## I<sup>2</sup>C-EEPROM

#### Zie I<sup>2</sup>C-inleiding

In dit project worden de I<sup>2</sup>C-componenten 24C02 (EEPROM met 2 kbits = 256 bytes) en PCF8574 (I/O-expander) gebruikt. Voor de gegevens van de I/O-expander verwijzen we naar de eerder behandelde projecten ESW-18 en ESW-19.

Bij het schrijven naar de EEPROM wordt de inhoud daarvan overschreven met de stand van de vier schakelaars; telkens 4 bits dus. De sampletijd is 10 seconden. Daarna wordt de inhoud van de EEPROM getoond. De lengte van het zo gemaakte "bestand" is opgeslagen op geheugenadres-0.

Het basisadres van de EEPROM is \$A0. Met jumpers of dipswitches aangeschakeld op A2 en A0 wordt het adres:

voor het schrijven naar de EEPROM:	%10100100 (\$A4).
voor het lezen van de EEPROM:	%10100101 (\$A5).

De woorden **INGANG** en **NAAR-LEDS** komen in voorgaande listings reeds voor. De waarden die **INGANG** leest worden naar een geheugenadres in het EEPROM-IC geschreven. Dat adres in de EEPROM wordt in plaats van het eerste databyte verzonden: het 'word address' of 'register address', hier afgekort tot reg.adr.

#### Schrijven met **SCHRIJF-BYTE**:

Start	Slave-adres	R/W	Ack	Registeradres	Ack	Databyte	Ack	Stop
S	a6 a5 a4 a3 a2 a1 a0	0	0	a7 a6 a5 a4 a3 a2 a1 a0		b7 b0	0	Р

gezonden door master

gezonden door slave

Aan het begin van deze procedure wordt eerst in een lus getest of de slave een Acknowledge teruggeeft. Een EEPROM heeft tijd nodig om een databyte te verwerken en er kan dus nog een vorig byte in behandeling zijn.

Als er uit het slave-geheugen gelezen moet worden dan moet na het schrijven van het registeradres en eventuele databytes met R/W=0 meteen een doorstart of een zogenaamde 'Repeated-start' gegeven worden, gevolgd door het slave-adres met R/W=1 (lezen). De Repeated-start is gelijk aan een gewone starttoestand.

Het lezen met LEES-BYTE :

Start	Slave-adres	R/W	Ack	Reg.adres	Ack	Rep Start	Slave-adres	R/W	Databyte	NAck	Stop
S	a6 a0	0	0		0	S	a6 a0	1	b7 b0	1	Р

Het laatste databyte wordt gegeven door de slave. Aansluitend wordt de (aller-)laatste Ack als NoACK door de master gegeven.

### Schema

Dit schema toont het deel met de EEPROM op de EHW-18I-print.



### Software

In de volgende listing ontbreken de routines voor de primitieven. We moeten deze zelf uit de bibliotheek in de subdirectory **LIB**\ ophalen om de listing compleet te maken. Ook de onderstaande broncode lezen we van schijf in. Het bestand vinden we in de subdirectory **'EGEL**\ of in **WORK**\. Desnoods even zoeken. We laden achtereenvolgens de bestanden:

I2C-PRIM.FRT I2C-8574.FRT I2C24C02.FRT ESW-20.FRT

```
\ I<sup>2</sup>C-voorbeeld met PCF8574 en 24C02. Lengte 403 bytes.
\ Type voor het laden van het bestand: 89C2051 TARGET
\ Laad dan de bibliotheekbestanden: I2C-PRIM.FRT, I2C-8574.FRT en I2C24C02.FRT.
\ Laad tenslotte dit bestand:
: NAAR-LEDS
                                         \ Stuur data naar de PCF8574 op adres-0
                ( byte -- )
    4 LSHIFT INVERT I/O-ID1 !BYTE
    ;
: INGANG
                ( -- byte )
                                         \ Lees data van PCF8574 op adres-0
   I/O-ID1 @BYTE INVERT $F AND
    ;
: VUL-EEPROM
                ( -- )
                                         \ Vul EEPROM met monsters vd schakelaars
                                         \ Eerste databyte is adres-1
   1
   100 FOR
                                         \ Neem 100 monsters (totaal 10 sec)
        INGANG OVER SCHRIJF-BYTE
                                         \ Zet de stand vd schakelaars in EEPROM
        INGANG NAAR-LEDS 1+ 100 MS
                                         \ Verhoog adresteller en wacht
   NEXT
   0 SCHRIJF-BYTE
                                         \ Bewaar de datalengte op adres-0
    ;
: TOON-EEPROM
                ( -- )
                                         \ Toon de datamonsters uit de EEPROM
    0 LEES-BYTE 1 DO
                                         \ Haal de lengte op
        I LEES-BYTE NAAR-LEDS
                                         \ Druk de data byte voor byte af
        100 MS
                                         \ Wacht 100 millisec
   LOOP
    ;
```

```
: KNIPPER ( -- ) \ Signaleer opstarten
5 FOR \ Knipper 5 maal
$FF NAAR-LEDS 250 MS
$00 NAAR-LEDS 250 MS
NEXT
;
: I2C-EEPROM ( -- ) \ Voorbeeld van gebruik I<sup>2</sup>C-EEPROM
SETUP SETUP-I2C
KNIPPER VUL-EEPROM KNIPPER \ Toon opstarten en vul EEPROM
BEGIN
TOON-EEPROM
AGAIN
;
' I2C-EEPROM RESET-VEC SET-VECTOR \ I<sup>2</sup>C-EEPROM start op na een reset
```

## I<sup>2</sup>C-Klok

#### Zie I<sup>2</sup>C-inleiding

In dit project worden de I<sup>2</sup>C-componenten PCF8583 (een klok met wekker, kalender en circa 220 bytes extra RAM) en een PCF8574 (I/O-expander) gebruikt. Voor de gegevens van de I/O-expander wordt verwezen naar de projecten ESW-18 en ESW-19.

Het basisadres van het klok-IC is \$A0. Dit is gelijk aan het basisadres van de EEPROM! Er wordt een jumper geplaatst op de enig aanwezige externe adresbit A0. Voor het schrijven: %1010 0010 (\$A2). Voor het lezen: %1010 0011 (\$A3).

Vergelijk dit met de adreskeuze voor de EEPROM in project ESW-20.

De waarden voor jaar, datum, uur, enzovoorts staan in het geheugen van het klok-IC. Het adres van de geheugenlokatie moet op de plaats van de eerste databyte worden opgegeven: het 'word address' of 'register address', hier afgekort tot reg.adr.

Het instellen van de klok door **!KLOK** :

Start	Slave-adres	R/W	Ack	Reg.adres	Ack	Databyte	Ack	Stop
S	a6 a5 a4 a3 a2 a1 a0	0	0	a7 a0		b7b0	0	Р

gezonden door master

gezonden door slave

Als er uit het slave-geheugen gelezen moet worden dan moet na het "schrijven" van het reg.adr. en eventuele databytes met R/W=0 een "Repeated-start" of "Doorstart" gegeven worden, gevolgd door het slave-adres met R/W=1 (lezen). De "Repeated-start" is gelijk aan een gewone startconditie.

Het lezen van de klok door @KLOK :

Start	Slave-adres	R/W	Ack	Reg.adres	Ack	Rep Start	Slave adres	R/W	Databyte	NAck	Stop
S	a6 a0	0	0	a7 a0	0	S	a6 a0	1	b7 b0	1	Р

De laatste databyte wordt verzonden door de slave. Vervolgens wordt de Negative Acknowledge door de master gegeven.

### Schema

Dit schema toont het klok-IC-gedeelte op de EHW-18I-print.



## Software

Laad achtereenvolgens de bestanden

I2C-PRIM.FRT uit LIB\, I2C-8583.FRT uit LIB\ en ESW-21.FRT uit 'EGEL\ of uit WORK\

In de listing staat vermeld hoe **ZET-KLOK** en **ZET-DATUM** gebruikt worden.

In de colon-definitie **WEK-KLOK** staat: **WEK KLOK DAGELIJKSE WEKKER**. Vervang dit door: **NORMALE KLOK GEEN WEKKER** of eenvoudiger nog door **SETUP-KLOK** en ga na wat de gevolgen zijn voor de werking van de klok.

```
\ PCF8583 of PCF8593, Een proef met klok en wekker, lengte 813 bytes.
\ Type voor het laden van het bestand: 890C2051 TARGET
\ Laad dan de bibliotheekbestanden: I2C-PRIM.FRT en I2C-8583.FRT
\ En laad tenslotte dit bestand:
$40 CONSTANT I/O-ID
                                       \ Device-adres-0 van gebruikte chip
\ Uitvoerroutine voor PCF8574-chips
: !BYTE
              ( byte chip -- )
   START-BIT
   $FE AND BYTE-UIT
                                       \ Stuur schrijfadres naar chip en
   BYTE-UIT
                                       \ daarna de databyte
   STOP-BIT
                                       \ Klaar
    ;
: NAAR-LEDS ( byte -- )
                                       \ Stuur data naar PCF8574 op adres-0
    INVERT $0F OR I/O-ID !BYTE
    ;
: KNIPPER
              ( -- )
                                       \ Visualiseer opstarten
   255 NAAR-LEDS 50 MS
    0 NAAR-LEDS 50 MS
    ;
: VOLGENDE-WEK-TIJD ( -- )
                                       \ Zet wektijd 10 sec na nu !!!
   WEK KLOK
                                       \ Reset wekker
   LEES-KLOK ROT 10 + >R
                                       \ Maak nieuwe wektijd over 10 sec.
   R@ 59 U> IF
                                       \ Meer dan 59 sec. ?
       R> 60 - >R
                                       \ Ja, corrigeer seconden
        SWAP 1+ >R R@ 60 = IF
R> DROP 0 >R 1+
                                       \ Verhoog min, meer dan 59 min. ?
                                       \ Ja, corrigeer minuten en uren
           DUP 24 = IF DROP 0 THEN
                                       \ Dag verstreken, dan uren op nul
        THEN
       R> SWAP
                                       \ Herstel minuten
   THEN
    R> -ROT ZET-WEK-TIJD
                                       \ Herstel seconden
    ;
VARIABLE SEC
                                       \ Onthoud aantal seconden
                   ( -- )
: TIK
                                       \ Laat elke sec. een LED flitsen
   LEES-KLOK 2DROP DUP FROM SEC <> IF \ Seconde verstreken ?
      128 NAAR-LEDS
                                       \ Ja, laad hoogste LED flitsen
       DUP TO SEC
                                       \ Bewaar aantal seconden
       25 MS 0 NAAR-LEDS
    THEN
   DROP
    ;
                   ( -- )
: WEK-KLOK
                                       \ Toon klok en wekker
   SETUP KNIPPER
   WEK KLOK DAGELIJKSE WEKKER
                                       \ Initialiseer klok
   BEGIN
       VOLGENDE-WEK-TIJD
                                       \ Zet wekker vooruit
       BEGIN TIK WEKKER? UNTIL
                                       \ Wacht tot wekker afloopt en flits
       KNIPPER
                                       \ Na het aflopen alle LED's even aan
   AGAIN
    ;
WEK-KLOK
             RESET-VEC SET-VECTOR \ WEK-KLOK start op na een reset
\ Einde
```

## Appendix 'Input/Output'

#### De poorten van de AT89Cx051

De poorten -1 en -3 van de AT89Cx051 worden gebruikt als In-poort en als Uit-poort. De poort is als geheel, dus alle 8 bits tegelijk, aan te spreken. De poortbits kunnen ook individueel benaderd worden. Een aantal bitpoorten wordt ook gebruikt als timer, counter, als seriële poort en voor interrupts. Deze komen in de volgende appendices aan de orde. Poort-3.6 wordt door de 2051 intern gebruikt en is dus niet beschikbaar. Zie ook het ByteForth Handboek en/of de SFR-tabel aan het eind van deze appendix.



Figuur 1 Pingebruik

### De uit-poorten

De uitgangen van de poorten gedragen zich als een geaarde-emitterschakeling met een pull-upweerstand van ongeveer 33K. Er zijn twee manieren om de uitvoerpoort te belasten.

#### • De uitgangen "logisch hoog" belasten.

Als een uitgang logisch '1' is dan spert de interne transistor (FET) van de microcontroller-poort. De stroom aan de uitgang wordt dan geleverd door de interne pull-upweerstand. Deze kan maximaal 0,08 mA leveren voor een logische 1 bij 2,4 V. Daarom is bij het relais in project-9 een NPN-Darlingtontransistor als driver toegepast. Het relais in ESW-09 is aan de voedingsspanning +V gelegd. Deze zou een hogere waarde mogen hebben dan 5 Volt, bijvoorbeeld 12 V, maar dan mag deze spanning alleen over het relais staan en moet de pull-upweerstand aan massa gelegd worden:



Figuur 2 Logisch hoog belasten

• De uitgangen "logisch laag" belasten.

Als een uitgang logisch '0' is dan geleidt de interne transistor. De stroom van de aangesloten belasting loopt nu door de emitter van de transistor. Deze stroom moet worden begrensd en mag voor een poortbit niet meer dan 20 mA bedragen. De som van de stromen door alle bits van alle poorten mag op eenzelfde moment niet hoger zijn dan 80 mA. De belasting wordt nu tussen de voedingsspanning van 5V en de poort aangesloten. Voor hogere belastingen kan een PNP-transistor worden gebruikt als driver. Voor het relais ziet dit er dan als volgt uit.



Figuur 3 Logisch laag belasten

Het relais werkt nu omgekeerd in vergelijking met het relais uit project ESW-09. Het kan daarom nodig zijn om de software aan te passen of om een breek- of omschakelrelais te gebruiken.

### De in-poorten

Bij het opstarten van de microcontroller worden alle poortbits hoog. Bij een reset blijft de toestand behouden. De poort of de poortbits kunnen in de hoog-toestand als ingang worden gebruikt. Zie project ESW-03.

Als er eerder een '0' in een poort geschreven werd dan moet er een SET-opdracht worden gegeven voordat de poort (weer) als ingang gebruikt kan worden. Bij het gebruik van een poort als uitgang is iets dergelijks niet van toepassing.





Figuur 4 Port 1 bit

Figuur 5 Port 3 bit

#### **DC Characteristics**

 $T_A=-40^\circ C$  to  $85^\circ C,~V_{CC}=2.0V$  to 6.0V (unless otherwise noted)

Symbol	Parameter	Condition	Min	Max	Units
VIL	Input Low Voltage		-0.5	0.2 V <sub>CC</sub> - 0.1	V
VIH	Input High Voltage	(Except XTAL1, RST)	0.2 V <sub>CC</sub> + 0.9	V <sub>CC</sub> + 0.5	V
V <sub>IH1</sub>	Input High Voltage	(XTAL1, RST)	0.7 V <sub>CC</sub>	V <sub>CC</sub> + 0.5	v
V <sub>OL</sub>	Output Low Voltage <sup>(1)</sup> (Ports 1, 3)	$I_{OL} = 20 \text{ mA}, V_{CC} = 5V$ $I_{OL} = 10 \text{ mA}, V_{CC} = 2.7V$		0.5	V
V <sub>OH</sub>	Output High Voltage	$I_{OH} = -80 \ \mu A, \ V_{CC} = 5V \pm 10\%$	2.4		V
	(Ports 1, 3)	I <sub>OH</sub> = -30 μA	0.75 V <sub>CC</sub>		V
		I <sub>OH</sub> = -12 μA	0.9 V <sub>CC</sub>		V
IIL	Logical 0 Input Current (Ports 1, 3)	$V_{IN} = 0.45V$		-50	μA
ITL	Logical 1 to 0 Transition Current (Ports 1, 3)	$V_{IN}=2V,V_{CC}=5V\pm10\%$		-750	μA
ILI	Input Leakage Current (Port P1.0, P1.1)	$0 < V_{IN} < V_{CC}$		±10	μA
Vos	Comparator Input Offset Voltage	$V_{CC} = 5V$		20	mV
V <sub>CM</sub>	Comparator Input Common Mode Voltage		0	V <sub>CC</sub>	V
RRST	Reset Pulldown Resistor		50	300	KΩ
CIO	Pin Capacitance	Test Freq. = 1 MHz, T <sub>A</sub> = 25°C		10	pF
Icc	Power Supply Current	Active Mode, 12 MHz, V <sub>CC</sub> = 6V/3V		15/5.5	mA
		Idle Mode, 12 MHz, $V_{CC} = 6V/3V$ P1.0 & P1.1 = 0V or $V_{CC}$		5/1	mA
	Power Down Mode <sup>(2)</sup>	V <sub>CC</sub> = 6V P1.0 & P1.1 = 0V or V <sub>CC</sub>		100	μA
		V <sub>CC</sub> = 3V P1.0 & P1.1 = 0V or V <sub>CC</sub>		20	μA

 Notes:
 1. Under steady state (non-transient) conditions, I<sub>OL</sub> must be externally limited as follows: Maximum I<sub>OL</sub> per port pin: 20 mA Maximum total I<sub>OL</sub> for all output pins: 80 mA If I<sub>OL</sub> exceeds the test condition, V<sub>OL</sub> may exceed the related specification. Pins are not guaranteed to sink current greater than the listed test conditions.

2. Minimum  $V_{CC}$  for Power Down is 2V.

# Speciale Functie Registers (SFR's)

adres +	+ 0	+ 1	+ 2	+ 3	+ 4	+ 5	+ 6	+ 7
\$F8								
\$F0	В							
	0000000							
\$E8								
\$E0	ACC							
	0000000							
\$D8								
\$D0	PSW							
	00000000							
\$C8								
\$C0								
\$B8	IP							
<b>#</b> D.0	xxx00000							
\$B0	P3:							
	11111111		<b>D</b> 0 0	<b>DO 0</b>	<b>D</b> 0 4	<b>D</b> 0 5	<b>D</b> 0 0	<b>D0</b> 7
<b>#</b> 4.0	P3: .0	P3: .1	P3: .2	P3: .3	P3: .4	P3: .5	P3: .6	P3: .7
\$A8	IE							
¢40	000000							
\$AU \$00	SCON.	ODUE.						
<b></b>	500N:	SBUF:						
\$00	D1.	*****						
ψ30	11111111							
	P1 · 0	P1· 1	P1 · 2	P1 · 3	P1· 4	P1: 5	P1.6	P1· 7
\$88			TI 0.	TI 1.	TH0.	TH1.	1 11 10	
ΨCC	00000000	00000000	00000000	00000000	00000000	00000000		
\$80		SP	DPL	DPH				PCON:
		00000111	00000000	0000000				00000000
adres +	+ 0	+ 1	+ 2	+ 3	+ 4	+ 5	+ 6	+ 7

## Appendix 'Timers/Counters'

#### De poorten van de AT89Cx051

De twee aanwezige timers/counters -0 en -1 worden gebruikt als klok of als teller. In het SFR (Special Function Register) TMOD wordt met de bit C/T de funktie 'timer' of 'counter' ingesteld. De vier mogelijke modes worden ingesteld met de bits -M0 en -M1.

	(MSB)							(LSB)
	GATE	C/T	M1	MO	GATE	C/T	M1	MO
		Ti	imer1	•		Tin	ner0	
GATE	Gating contro while INTx p cleared, Tim set.	ol when set. Tii in is high and T er x is enabled	mer/Counter x is TRx control pin is whenever TRx c	enabled only s set. When control bit is		Timer 0 gate	bit ter/timer select	bit
C/T	Timer or Counter Selector cleared for Timer operationTimer 0 M1 bit(input from internal system clock). Set for CounterTimer 0 M0 bitoperation (input from Tx input pin).Timer 0 M0 bit							
M1	Mode bit 1							
MO	Mode bit 0							
	M1	MO	Mode	Operating M	ode			
	0	0	0	13-bit Timer N 8-bit Timer/Co	Mode. ounter THz with	n TLx as 5-bit pr	escaler.	
	0	1	1	16-bit Timer Mode. 16-bit Timer/Counters THx and TLx are cascaded; there is				no prescaler.
	1	0	2	8-bit Auto Reload. 8-bit auto-reload Timer/Counter THx holds a value which is to be reloaded into TLx each time it overflows.				
	1	1	3	Split Timer M (Timer 0) TL0 0 control bits.	ode. ) is an 8-bit Tim TH0 is an 8-bi	er/Counter cont t timer only cont	rolled by the sta rolled by Timer	andard Timer 1 control bits.
	1	1	3	(Timer 1) Timer/Counter 1 stopped.				

#### Figuur 6 TMOD: Timer/Counter Mode Control Register

Timer SFR	Purpose	Address	Bit-Addressable
TCON	Control	88H	Yes
TMOD	Mode	89H	No
TLO	Timer 0 low-byte	8AH	No
TL1	Timer 1 low-byte	8BH	No
THO	Timer 0 high-byte	8CH	No
TH1	Timer 1 high-byte	8DH	No
T2CON <sup>(1)</sup>	Timer 2 control	С8Н	Yes
T2MOD <sup>(1)</sup>	Timer 2 Mode	С9Н	No
RCAP2L <sup>(1)</sup>	Timer 2 low-byte capture	САН	No
RCAP2H <sup>(1)</sup>	Timer 2 high-byte capture	СВН	No
TL2 <sup>(1)</sup>	Timer 2 low-byte	ССН	No
TH2 <sup>(1)</sup>	Timer 2 high byte	СDH	No

Note: 1. AT89C52 only.

#### Figuur 7 Registers die de timer/counters beïnvloeden

Poortbit-P3.4 (T0) is voor timer/counter-0 de verbinding met de buitenwereld en poortbit-P3.5 (T1) is dat voor timer/counter1.

### Gebruik als klok (timer).

Bij het gebruikte 6-MHz-kristal wordt de teller van de klok 500.000 keer per seconde verhoogd.

## Gebruik als teller (counter).

De teller reageert op een overgang van 1 naar 0 van P3.4 (T0) respectievelijk P3.5 (T1). Bij het gebruikte 6-MHz-kristal is de maximaal mogelijke telfrekwentie 250 kHz.

	(MSB)							(LSB)
	TF1	TR1	TF0	TR0	IE1	IT1	IE0	ITO
Symbol	Position	Name and Si	ignificance					
TF1	TCON.7	Timer 1 overfl vectors to inte	Timer 1 overflow flag. Set by hardware on Timer/Counter overflow. Cleared by hardware when processor vectors to interrupt routine.					
TR1	TCON.6	Timer 1 run c	ontrol bit. Set/c	leared by softw	are to turn Time	er/Counter on/of	if.	
TF0	TCON.5	Timer 0 overflow flag. Set by hardware on Timer/Counter overflow. Cleared by hardware when processor vectors to interrupt routine.						
TR0	TCON.4	Timer 0 run c	Timer 0 run control bit. Set/cleared by software to turn Timer/Counter on/off.					
IE1	TCON.3	Interrupt 1 ed processed.	Interrupt 1 edge flag. Set by hardware when external interrupt edge detected. Cleared when interrupt processed.					
IT1	TCON.2	Interrupt 1 type control bit. Set/cleared by software to specify falling edge/low level triggered external interrupts.						
IE0	TCON.1	Interrupt 0 edge flag. Set by hardware when external interrupt edge detected. Cleared when interrupt processed.						
ITO	TCON.0	Interrupt 0 type control bit. Set/cleared by software to specify falling edge/low level triggered external interrupts.						

We onderscheiden vier werkwijzen voor de timers/counters. Met timer wordt timer en/of counter bedoeld.

• Mode-0: instelling als timers/counter met prescaler.

Timer-0 en timer-1 gedragen zich als een 8-bit-timer (TH0 voor timer-0 en TH1 voor timer-1) met een 5-bit prescaler (TL0 resp. TL1). De prescaler deelt de klokfrekwentie door 32. Als alle bits van de teller naar nul gaan (overflow) dan wordt de interruptvlag van de timer (TF0 resp. TF1) geset. De ingang wordt weer vrijgegeven als TR0=1 resp. TR1=1 (in TCON) en GATE=0 (in TMOD) of /INT1=1. Als GATE=1 dan is het met behulp van /INT1 mogelijk de lengte van pulsen te meten.

• Mode-1: Instelling als timers/counters.

Het verschil met Mode-0 is dat er geen prescalers zijn en dat alle 16 bits als geheel timer-0 resp. timer-1 vormen.

• Mode-2: Instelling als timer/counter.

In TH0 resp. TH1 wordt een 8-bit-waarde verwacht. TL0 resp. TL1 werken als 8-bit-tellers. Als een teller 0 wordt (overflow) dan wordt de waarde van THx naar TLx gekopieerd. Bovendien wordt TFx geset.

• Mode-3: Instelling als split timer.

Timer-0 en timer-1 verschillen hier van gedrag.

Timer-0:

TL0 is een 8-bit-timer/counter die op de gebruikelijke wijze werkt. TH0 is een 8-bit-timer die door de controlbits van timer-1 wordt bediend. TH0 stuurt zelf de interrupts van timer-1 aan. Timer-0 bevat nu dus twee gescheiden tellers: T0 als timer/counter en T1 uitsluitend als timer.

Timer-1:

Deze gedraagt zich als timer/counter maar behoudt zijn telwaarde. De timer kan gebruikt worden voor toepassingen waarbij geen interrupts nodig zijn, bijvoorbeeld een baudrate-generator voor RS232.







Figuur 10 Timer/counter 0 mode 3: Two 8-Bit Counters

## Appendix 'Serieel / RS232'

#### De poorten van de AT89Cx051

Met de microcontroller is een full-duplexverbinding mogelijk. Het RxD-signaal is toegewezen aan poortbit-P3.0 en het TxD-signaal aan bit-P3.1.

Ook hier zijn vier werkwijzen mogelijk. Deze worden ingesteld met de bits-SM0 en -SM1 in (SFR) SCON.

	(MSB)							(LSB)
	SM0	SM1	SM2	REN	TB8	RB8	TI	RI
Symbol	Position	Name and S	ignificance					
SM0	SCON.7	Serial port m	ode bit 0 (see ta	able below).				
SM1	SCON.6	Serial port m	ode bit 1 (see ta	able below).				
SM2	SCON.5	Enables the multiprocessor communication feature in Modes 2 and 3. In Mode 2 or 3, if SM2 is set to 1, then RI will not be activated if the received 9th data bit (RB8) is 0. In Mode 1, if SM2 = 1, then RI will not be activated if a valid stop bit was not received. In Mode 0, SM2 should be 0.						
REN	SCON.4	Enables seria	al reception. Set	t by software to	enable receptio	on. Clear by sof	ware to disable	reception.
TB8	SCON.3	The 9th data	bit that will be t	ransmitted in Me	odes 2 and 3. S	Set or clear by s	oftware.	
RB8	SCON.2	In Modes 2 and 3, the 9th data bit that was received. In Mode 1, if SM2 = 0, RB8 is the stop bit that was received. In Mode 0, RB8 is not used.						
ТΙ	SCON.1	Transmit interrupt flag. Set by hardware at the end of the 8th bit time in Mode 0, or at the beginning of the stop bit in the other modes, in any serial transmission. Must be cleared by software.				ginning of the		
RI	SCON.0	Receive interrupt flag. Set by hardware at the end of the 8th bit time in Mode 0, or halfway through the stop bit time in the other modes, in any serial reception (except see SM2). Must be cleared by software.						

Where SM0, SM1 specify the serial port mode as follows:

SM0	SM1	Mode	Description	Baud Rate
0	0	0	Shift Register	fixed (f <sub>OSC</sub> ./12)
0	1	1	8-bit UART	variable (set by timer)
1	0	2	9-bit UART	fixed (f <sub>OSC</sub> ./64 or f <sub>OSC</sub> ./32)
1	1	3	9-bit UART	variable (set by timer)
1	1	3	9-bit UART	variable (set by timer)

### Figuur 11 SCON: Serial Port Control Register

Special Function Register SBUF is de zend/ontvangbuffer. Zie het ByteForth Handboek en/of de SFR-tabel in de appendix 'Input/Output'. Het verzenden van de data wordt gestart door naar SBUF te schrijven. Het ontvangen wordt in mode-0 gestart door de statusbits RI=0 en REN=1 te maken. In de andere modes is de voorwaarde REN=1 voldoende. Van de data wordt steeds het LSB als eerste verzonden.

#### Mode-0

Het dataformaat is 8 bits. De pin-RxD wordt gebruikt in twee richtingen. Op pin-TxD staat de "klok". De frequentie is gelijk is aan de frequentie van de systeemklok gedeeld door twaalf.

#### • Mode-1

TxD en RxD hebben hun gebruikelijke funktie. Het dataformaat is 1 startbit, 8 databits, 1 stopbit. Na ontvangst wordt de stopbit geplaatst in bit-RB8 van (SFR) SCON. De baudrate is variabel. De keuze wordt gemaakt met bit-SMOD in (SFR) PCON.

#### • Mode-2

TxD en RxD hebben hun gebruikelijke funktie. Het dataformaat is 1 startbit, 8 databits, 1 programmeerbaar bit en 1 stopbit. Na ontvangst wordt de 9-de bit geplaatst in bit-RB8 van (SFR) SCON. De 9-de bit wordt bij verzending gelezen uit bit-TB8 van (SFR) SCON en kan gebruikt worden als parity door de P-vlag van het Program Status Word (PSW) naar TB8 te kopiëren. De baudrate is de systeemklok gedeeld door 32 of 64.

#### • Mode-3

Deze mode is geheel vergelijkbaar met Mode-2. Alleen de baudrate is vrij programmeerbaar.

Baud Rate	fosc	SMOE		Timer 1	
			с/т	Mode	Reload Value
Mode 0 Max: 1 MHz	12 MHz	х	х	х	х
Mode 2 Max: 375K	12 MHz	1	х	х	х
Modes 1, 3: 62.5K	12 MHz	1	0	2	FFH
19.2K	11.059 MHz	1	0	2	FDH
9.6K	11.059 MHz	0	0	2	FDH
4.8K	11.059 MHz	0	0	2	FAH
2.4K	11.059 MHz	0	0	2	F4H
1.2K	11.059 MHz	0	0	2	E8H
137.5	11.986 MHz	0	0	2	1DH
110	6 MHz	0	0	2	72H
110	12 MHz	0	0	1	FEEBH

#### Figuur 12 Veel gebruikte baudsnelheden gegenereerd door timer-1

SMOD, in figuur-12 per abuis SMOE genoemd, is de hoogste bit op het adres van (SFR) PCON-\$87 en is niet bit-adresseerbaar. Zie het ByteForth Handboek en/of de SFR-tabel in de appendix 'Input/Output'. Voor toepassingen: zie projecten ESW-06, -07, -13 en de RS232-primitieven in de bibliotheek **LIB**. In de genoemde projecten wordt timer-1 voor de seriële poort gebruikt.

## Appendix 'Externe Interrupts'

De poorten van de AT89Cx051

Er zijn twee interrupt-bitpoorten: /INT0 op P3.2 en /INT1 op P3.3.

Bij externe interrupts wordt gereageerd op een neergaande flank (1/0) of op een logisch laag niveau. De keuze hiertussen wordt gemaakt met de bit-IT0 resp. -IT1 van (SFR) TCON. De interrupt op een neergaande flank wordt geactiveerd door de bits-IE0 resp. -IE1. De CPU reset deze bits als naar de serviceroutine gesprongen wordt. Bij interrupts door een logisch niveau moet in de serviceroutines de ingang hersteld worden omdat anders de interrupt-aanroep blijft bestaan.

	Source	Priority Within Level	Interrupt	Source	Vector Address
1.	IE0	(highest)	External 0	IE0	0003H
2	TEO		Timer 0	TF0	000BH
2.			External 1	IE1	0013H
3.	IE I		Timer 1	TF1	001BH
4.	TF1		Serial Port	BL or TI	0023H
5.	RI + TI		Timor 2	TE2 or EVE2	00284
6.	TF2 + EXF2	(lowest)	TITIET Z	TFZ OF EAFZ	00201
o. The PEALE		(1011000)	System Reset	RST	0000H

#### Figuur 13 Interruptbronnen

Figuur 14 Interruptvectoren



Figuur 15 8052 Interrupt Control System

In de listings worden interruptserviceroutines afgesloten met IRET.

## Appendix 'Stappenmotoren'

Een stappenmotor is een bijzondere vorm van een elektrische draaiveldmotor. De motor wordt gevoed met gelijkstroompulsen en beweegt met eindige stapjes. De stapgrootte kan uitgedrukt worden in graden, maar gewoonlijk wordt het aantal stappen per omwenteling opgegeven. Als de bekrachtiging niet verandert houdt de motor zijn stand vast. Bij stappenmotoren zijn de stators bewikkeld. De volgorde waarin een of meer wikkelingen aangesproken worden bepaalt de draairichting. De meest toegepaste stappenmotoren zijn de permanentmagneeten de hybride stappenmotoren. De permanentmagneetmotor is de goedkoopste en heeft een rotor die in asrichting om-en-om gepoolde magneetvelden heeft. Het aantal stappen per omwenteling bedraagt meestal 24 tot 48 in hele-stapbedrijf, zie de projecten ESW-11 en ESW-12.





Figuur 17

De rotor van de hybride stappenmotor bestaat uit een enkele cilindrische permanentmagneet waarvan het magneetveld ook in asrichting is gericht. Op elk uiteinde van de rotor zit een vertande bus van ferromagnetisch materiaal. De twee bussen zijn ten opzichte van elkaar een halve steek verschoven.



Bij een unipolaire stappenmotor wordt elke wikkeling slechts op een manier bekrachtigd voor wat betreft de stroomrichting. De polariteit van een wikkeling wordt dus niet omgekeerd: unipolair. Om de magnetisering van een stator wel te kunnen omkeren zijn er per stator twee wikkelingen aangebracht. Meestal zijn deze twee wikkelingen met elkaar verbonden zoals bij een "middenaftakking". Bij een 2-statormotor zijn er dan zes aansluitdraden. Bij sommige motoren zijn de "middenaftakkingen" van beide stators samengenomen en zijn er vijf aansluitdraden. Soms ook worden alle aansluitingen apart naar buiten gevoerd en zijn er acht aansluitdraden. In deze uitvoering kan ook de schakeling van de bipolaire motor toegepast worden zonder dat diens betere eigenschapen meegenomen worden.

Een bipolaire stappenmotor heeft op elke stator een enkele wikkeling die in beide mogelijke richtingen bekrachtigd kan worden: bipolair. Alle aansluitpunten worden gescheiden naar buiten gevoerd. Voor de 2-statormotor zijn er dan vier aansluitingen.

De in dit werkboek gebruikte motoren hebben de volgende specificaties.

#### Unipolaire stappenmotor:

5V, 250 mA, 18  $\Omega$  per wikkeling. De toegepaste driver is de ULN2803 (zie appendix), 48 stappen/omw, 6 aansluitdraden, kleuren:

poort	schema	kleur
P1.4	M1.1	bruin
P1.6	M1.4	zwart
+-motor	M1.2	rood
+-motor	M1.5	rood
P1.5	M1.3	geel
P1.7	M1.6	oranje

#### **Bipolaire stappenmotor:**

7,5Volt, 250 mA, 30  $\Omega$  per wikkeling. Toegepaste driver: zie appendix L293D. 40 stappen/omw, 4 aansluitdraden, kleuren:

poort	schema	kleur
P1.4	M1.1	zwart
P1.6	M1.3	geel
P1.5	M1.2	blauw
P1.7	M1.4	violet





BIPOLAR





Van boven naar beneden:

- helestapbedrijf, tweefasenaansturing

- halvestapbedrijf

- helestapbedrijf, eenfase-aansturing

## Appendix 'RC5'

#### **IR-Remote Control RC5**

Voor infrarood afstandsbedieningen voor TV, video, versterkers, CD-spelers e.d. heeft Philips een protocol met de naam RC5 ontwikkeld. Dit wordt toegepast bij de in Europa gangbare afstandsbedieningen. Er zijn ook andere systemen in gebruik.

Er zijn drie niveau's te onderscheiden.

Een basissignaal van 36 kHz. De periode van deze pulstrein is 1.000.000 / 36.000 = 27,7 (1rep) µsec. De hoogtijd van de puls is 6,94 (1rep) µsec. N.B. (1rep) betekent 1 laatste cijfer repetent. In de ontvanger ESW-14 wordt dit signaal eenvoudig gelijkgericht. Bij de zender ESW-15 wordt het signaal gemaakt met een oscillatorschakeling.



Figuur 20

• Het 36-kHz-signaal wordt "gemoduleerd" tot een bitstroom. Een bitstroom bestaat uit 14 bits. Op bitniveau wordt de waarde van een bit voorgesteld door:



Figuur 21

De halve bitlengte bevat 32 pulsen van het 36kHz-signaal. Een volle bitlengte is dus  $2x32x27.7 \ \mu sec = 1,7 \ (1rep)$  millisec.

In onze ontvanger wordt niet op een flank gedetecteerd maar wordt het signaal op 1/4 en op 3/4 van de puls gemeten, juist voor en na de flank. De bemonstering begint wel op de eerste flank van het geïnverteerde signaal. In de zender worden de bitsignalen volledig door de software gegenereerd. De 14 bits waaruit een enkelvoudige bitstroom bestaat nemen 24,8 (1rep) ms aan tijd in beslag.

Achtereenvolgens in de tijd is de betekenis van de bits:

- 1: Een startbit dat altijd 1 is.
- 2: Een fieldbit of tweede startbit. Als deze 1 is dan zijn de zes commandobits 0...63 en als deze 0 is dan zijn ze 64...127, dus: bitwaarde + 64. De ontvangersoftware negeert deze bit (cmd==cmd+64). Door de zender wordt altijd een 1 verzonden.
- 3: Een controlbit met een togglefunctie waarmee vastgesteld kan worden of op de zender een knop is losgelaten. Volgens het protocol is in de eerste bitstroom de waarde =1. In onze ontvanger wordt de controlbit genegeerd; bij de zender is de controlbit altijd 1.
- 4..8: Vijf systeembits die het te bedienen toestel (radio, TV enz.) beschrijven; afstandsbediening en toestel moeten natuurlijk dezelfde systeemcode hebben.
- 9..14: Zes commandobits die de ingedrukte toets, het commando dus, voorstellen.



Figuur 22

De stroom van 14 bits wordt steeds weer uitgezonden zolang de knop op de afstandsbediening blijft ingedrukt. De tijd tussen de groepen is bij de zender ongeveer 1 seconde. In het protocol wordt voor het herhalen een periode van minimaal 64x24,8 = 113,7 (1rep) ms voorgeschreven. Bij onze zender wordt bij het indrukken van een knop geen debounce- en scantijd in de software toegepast. Samen zouden deze 2 bittijden moeten bedragen, d.i. ca 3,5 ms. De tijd die nodig is voor de aansturing van de LED neemt deze taak ruimschoots in beslag.

Codes voor de toestel- / systeembits					
0	1	TV-1 respectievelijk TV-2			
2		Teletekst			
4		Laser Visionspeler			
5	6	Videorecorder (-1 resp2)			
8	10	Satelliet-TV-ontvanger (-1 resp2)			
12		CD-videospeler			
14		Foto-CD-speler			
16	19	(Voor-)versterker (-1 resp2)			
17		Afstemeenheid / ontvanger			
18	23	Magn.recorder resp. DAT / DDC			
20		CD-speler			
21		Grammofoon / audiorack			
22		Satelliet-audio-ontvanger			
26		Beschrijfbare CD			

De volgende codes zijn veelal gereserveerd voor toekomstige toepassingen en kunnen daarom voor experimenten worden gebruikt: 7, 11, 13, 15, 24, 25 en 27 t/m 31.

De commandobits zijn voor een deel voor alle toestellen gelijk. Een aantal geven we hieronder.

Universele commando's

09	selectie (preset)
12	stand-by
13	mute
16	volume+
17	volume-
26	balans rechts
27	balans links
64	functie-selectie

## Appendix 'LCD'

#### Liquid Crystal Displays

LCD's zijn er in karakter- en matrix- of dot-georiënteerde uitvoeringen. Het LCD in dit werkboek heeft een karakterset en kan twee regels van 16 tekens weergeven. Daarnaast kunnen 8 tekens door de gebruiker gedefinieerd worden. Op het printje van het LCD is een Hitachi-chip aanwezig voor de aansturing. Vrijwel alle LCD-aanstuur-IC's werken met hetzelfde protocol waarvoor Hitachi de standaard heeft gezet. Kenmerkend voor de chip is dat er intern een buffer van 40 tekens per regel is, ook al is het display minder dan 40 tekens breed.

Om van regel te wisselen wordt in project ESW-17 de routine LCD-CR gebruikt. Om de chip op de 4-bit-mode in te stellen moet de routine LCD-SETUP strikt worden aangehouden. De adrespositie linksboven op het scherm wordt aangeduid met 00. In het geheugen betreffen de adressen:

			databus									
kommando	RS	R/W	D7	D6	D5	D4	D3	D2	D2 D1 D0 omschrijving			
display wissen	o	o	0	o	o	0	0	0	0	1	Wist het geheugen, niet dat van de karakterge- nerator. Zet de cursor op home-positie, adres 00 (linksboven).	
cursor home	o	o	o	0	o	0	0	0	1	×	Zet de cursor op home-positie. Verschuivingen van de tekst worden ongedaan gemaakt (adres 00 is weer linksboven).	
verschuif tekst en/of cursor	о	0	ο	0	о	0	0	1 ID S			Geeft aan in welke richting de cursor verschuift (ID) en of de tekst gelijktijdig ook moet verschuiven (S).	
display aan/uit cursor aan/uit/ knipperen	0	o	o	о	0	o	1	D	с	в	Display aan/uit (D) Cursor aan/uit (C) cursor knipperen ja/nee (B)	
cursor of dis- play verschuiven	0	o	0	0	о	1	sc	RL	×	x	Verschuift cursor (SC = 0) of tekst (SC = 1) naar rechts/links (RL).	
initialisatie	0	0	ο	ο	1	DL	N	x x x		×	Breedte data-bus (DL) 1 of 2 regels gebruiken (N)	
karakter- generator adres	0	0	0	1	karakter rij			rij		Geeft aan welke rij (000111) van welk karakter (000111) gedefinieerd moet worden met het volgende data-byte.		
geheugenadres	0	0	1		adres						Zet de adresteller op ''adres''. De volgende data zijn ASCII-tekens.	
BUSY-FLAG, adres lezen	0	1	BF		adres						Leest de BUSY-FLAG en het cursor-adres.	
data schrijven	1	0				da	ta				Schrijf data.	
data lezen	1	1	data								Lees data.	

\$80 tot \$A7 de eerste regel van het display en \$C0 tot \$E7 de tweede regel.

x = don't care

Figuur 23 Instructieset van het LCD

Higher 4bit 4bit 4bit	0000	0010	0011	0100	0101	0110	0111	1010	1011	1100	1101	1110	1111
××××0000													
××××0001													
××××0010													
××××0011													
××××0100													
××××0101													
××××0110													
××××0111													
××××1000													
××××1001													
××××1010													
××××1011													
××××1100													
××××1101													
××××1110													
××××1111													

Figuur 24 Karaktertabel van het LCD

## Appendix 'l<sup>2</sup>C'

De I<sup>2</sup>C-bus (Inter Integrated Circuit Bus) van Philips is een tweedraads bidirectioneel systeem voor 8-bit dataoverdracht tussen electronische componenten of IC's. De eisen aan de data-transfersnelheid zijn daarbij meestal niet extreem hoog. Voor I<sup>2</sup>C zijn speciaal ontworpen componenten verkrijgbaar. Ook bestaan er microcontrollers die met I<sup>2</sup>C zijn uitgerust. In dit werkboek zorgt de AT89C2051 zelf voor het beheer van de bus. Het protocol is in de software vastgelegd.

De gegevens en de besturingscommando's worden serieel tussen verschillende I<sup>2</sup>C-componenten heen en weer gezonden over een enkele draad: de SDA- of Serial DAta-lijn. Een tweede draad is de kloklijn of SCL- (Serial CLock-)lijn. Deze lijn bepaalt de betekenis van de toestand op de SDA-lijn. Beide draden worden gezien ten opzichte van massa.



Figuur 25 Geldigheid van data

Alle componenten zijn verbonden met dezelfde SDA- en SCL-lijnen. Een derde lijn voor interrupts wordt in dit werkboek niet gebruikt. De start- en stopcondities zijn de voorwaarden waarmee een overdracht begint en eindigt.





Alle I<sup>2</sup>C-componenten hebben een adres, een device-adres, dat uit 7 bits bestaat. Afhankelijk van het IC-type zijn de vier tot zes hoogste bits in de chip zelf vastgelegd en zijn de overblijvende een tot drie bits uitwendig in te stellen, bijvoorbeeld met jumpers. Met de 1 tot 3 hardwarematig in te stellen adresbits zijn 2 tot 8 verschillende adressen voor IC's van hetzelfde type samen te stellen. Men kan dus 2 tot 8 IC's van een zelfde type in een ontwerp opnemen. Er bestaan address-extended chips waarmee een uitgebreide adressering mogelijk is, maar daar maken wij geen gebruik van.

Aan de 7 bits van het device-adres wordt als 8e, laagste, bit een R/W-bit toegevoegd. Daarmee wordt een lees-(read, R/W=0) of schrijf- (write, R/W=1) actie opgedragen aan de ontvanger. De te verzenden adresbyte is nu de vereiste 8 bits breed. Zowel adressen als data zijn 8 bits.

Elke chip of component kan een *master* of een *slave* zijn. De master regelt het verkeer op de bus (datalijn en kloklijn) en bepaalt ook de richting van het dataverkeer. De slave reageert op de master en kan, afhankelijk van de voorafgaande opdrachten, data ontvangen of verzenden. Een component kan zowel zender als ontvanger zijn.

Bovendien geeft de ontvanger, of deze nu master of slave is, een bevestigingsbit ACK (soms een NACK, Not ACKnowledge) terug op de datalijn. De ACKnowledge wordt door de master gelezen om na te gaan of de overdracht geslaagd is. Een adres- of data-actie beslaat dus 9 bits: adres+ACK of data+ACK.

In de werkboekprojecten wordt één master toegepast: de microcontroller. Deze is altijd de master en heeft zelf geen adres. Alle hier toegepaste I<sup>2</sup>C-chips zijn slaves. Het gebruik van twee of meer masters vereist een strategie voor conflictoplossing om het geheel goed te laten functioneren; dat is hier dus niet nodig.



Figuur 28 Een complete gegevensoverdracht

## Het I<sup>2</sup>C-protocol uitgelegd

In de volgende schema's staan op wit de acties van de master en op grijs de antwoorden van de slave. Voor R/W lees *Read*, *Not Write*.

## Een eenvoudige schrijfopdracht

De master zendt een databyte naar de slave. Eerst wordt het slave-adres verzonden, daarna de databyte:

Start	Slave-adres	R/W	Ack.bit	Databyte	Ack.bit	Stop
		-				
gezor	nden door master					

gezonden door slave

Als er meer databytes verzonden worden, bijvoorbeeld drie:



gezonden door master

gezonden door slave

## Schrijfopdracht met registeradressering

De master zendt een databyte naar de slave. Deze byte moet in een intern register van de slave worden geplaatst. Dit registeradres is ook 8 bits breed en wordt als tweede byte, direct na het slave-adres, verzonden. Het registeradres is dus eigenlijk de eerste databyte in deze transfer.



## Eenvoudige leesopdracht

De master vraagt een databyte op bij een slave:

De master zendt het chipadres met R/W=1, de slave reageert met ACK. Vervolgens zendt de slave de data, de master geeft een NACK.

Start	Slave-adres	R/W	Ack	Databyte	Nack	Stop
S	a6 a5 a4 a3 a2 a1 a0	1		b7 b6 b5 b4 b3 b2 b1 b0		

gezonden door master

gezonden door slave

Tenslotte wordt een stopconditie gegenereerd.

### Leesopdracht met registeradressering

De master vraagt een databyte op uit een register van een slave en ontvangt deze byte. Omdat er eerst een registeradres naar de slave gezonden wordt (R/W=0) en vervolgens een databyte gelezen wordt (R/W=1), moet na een herhaalde startconditie (repeated-start) het slave-adres opnieuw verzonden worden.

Start	Slave-a	adres R/W Ack			Registeradres			••••			
S	a6 a5 a4 a3	a2 a1 a0	0		a7	a6 a5 a4	4 a3 a2 a1				
	Rep Start	Slave-adres				R/W	Ack		databyte	NoAck	Stop
	S	a6 a5 a	a6 a5 a4 a3 a2 a1 a0			1		a7 a6	a5 a4 a3 a2 a1		

gezonden door master

gezonden door slave

De 'repeated start'-conditie is gelijk aan de gewone startconditie. Ook hier wordt na de laatste databyte die door de slave naar de master gezonden wordt, een NACK gegeven door de master.

### Overdracht van meer dan een byte

Met gebruik van de 'repeated start' kunnen aansluitend na elkaar meer bytes opgevraagd of verzonden worden. Wordt een reeks databytes door de slave naar de master verzonden dan geeft de master telkens een ACK terug, behalve na de laatste: dan geeft de master een NACK. Ook afwisselend ontvangen en verzenden is mogelijk. Dit wordt hier niet toegepast.


De maximale snelheid van de datastroom voor I<sup>2</sup>C is 100 kHz in de standaardmode. Tegenwoordig zijn er echter componenten die 400 kHz of zelfs een 1-MHz-datastroom aan kunnen. Daarom heeft Philips intussen een vernieuwde I<sup>2</sup>C-standaard gepresenteerd waarin die hogere snelheden beschreven worden.

De minimale duur van een hoge en een lage puls zijn respectievelijk 4 en 4,7 µsec. Voor de projecten in het werkboek is gekozen voor een kristal van 6 MHz op het AT51-printje. Door deze keuze hoeven we in de software geen vertragingen op te nemen. Bij een hogere kristalfrequentie kan dat wel nodig zijn!

Dan worden ook de stijg- en daaltijden mogelijk van belang. Deze kunnen kritisch worden.



Figuur 29 Verbinding met IC's met vaste ingangsniveaus



Figuur 30 Verbinding met IC's met meer voedingsspanningen

VDD2. 3 ARE DEVICE DEPENDENT (e.g., 12V)



Figuur 31 Verbindingen met ingangsniveaus gerelateerd aan VDD gemengd met verbindingen met vaste ingangsniveaus



Figuur 32 Serieweerstanden (Rs) voor bescherming tegen spanningspieken



Figuur 33 Verbindingen van I<sup>2</sup>C-interfaces aan de I<sup>2</sup>C-bus

Het aantal aan te sluiten componenten wordt mede bepaald door de capaciteit van de SDA- en de SCL-lijn. Deze bedraagt maximaal 400 pF. Elke pin die op een lijn wordt aangesloten geeft maximaal 10pF extra. Verder is er nog de eigen capaciteit van de lijn, bepaald door de lengte en de kwaliteit (prijs!) daarvan.

Het systeem is geschikt voor korte afstanden, tot ongeveer 1 meter. Met buffers is het bereik te vergroten tot een veelvoud van 1m.

Device	Size (Bytes)	Page Size (Bytes)	Max Per Bus	Addresses Used
AT24C01	1K	8	1	None
AT24C01A	1K	8	8	A0, A1, A2
AT24C02	2K	8	8	A0, A1, A2
AT24C04	4K	16	4	A1, A2
AT24C08	8K	16	2	A2
AT24C16	16K	16	1	None
AT24C164	16K	16	8	A0, A1, A2
AT24C32	32K	32	8	A0, A1, A2
AT24C64	64K	32	8	A0, A1, A2

Figuur 34 Atmel's tweedraads (I<sup>2</sup>C-) seriële EEPROM-familie

#### 15.0 ELECTRICAL SPECIFICATIONS AND TIMING FOR I/O STAGES AND BUS LINES

The I/O levels, I/O current, spike suppression, output slope control and pin capacitance for  ${\rm I}^2{\rm C}\mbox{-bus}$  devices are given in Table 3. The  $\rm l^2C$  bus timing is given in Table 4. Figure 34 shows the timing definitions for the I<sup>2</sup>C-bus.

The noise margin for HIGH and LOW levels on the bus lines for fast-mode devices are the same as those specified in Section 10.0 for standard-mode I<sup>2</sup>C-bus devices.

The minimum HIGH and LOW periods of the SCL clock specified in Table 4 determine the maximum bit transfer rates of 100 kbit/s for standard-mode devices and 400 kbit/s for fast mode devices. Standard-mode and fast-mode I<sup>2</sup>C-bus devices must be able to follow transfers at their own maximum bit rates, either by being able to transmit or receive at that speed or by applying the clock synchronization procedure described in Section 7.0 which will force the master into a wait state and stretch the LOW period of the SCL signal. Of course, in the latter case the bit transfer rate is reduced.

PARAMETER	SYMBOL	STANDA DEV	RD-MODE ICES	FAST- DEV	UNIT	
		Min.	Max.	Min.	Max.	
LOW level input voltage: fixed input levels V <sub>DD</sub> -related input levels	V <sub>IL</sub>	-0.5 -0.5	1.5 0.3V <sub>DD</sub>	-0.5 -0.5	1.5 0.3V <sub>DD</sub>	V
HIGH level input voltage: fixed input levels V <sub>DD</sub> -related input levels	V <sub>IH</sub>	3.0 0.7V <sub>DD</sub>	*1) *1)	3.0 0.7V <sub>DD</sub>	*1) *1)	V
Hysteresis of Schmitt trigger inputs: fixed input levels V <sub>DD</sub> -related input levels	V <sub>hys</sub>	n/a n/a	n/a n/a	0.2 0.05V <sub>DD</sub>		V
Pulse width of spikes which must be suppressed by the input filter	t <sub>SP</sub>	n/a	n/a	0	50	ns
LOW level output voltage (open drain or open collector): at 3 mA sink current at 6 mA sink current	V <sub>OL1</sub> V <sub>OL2</sub>	0 n/a	0.4 n/a	0 0	0.4 0.6	V
$\begin{array}{l} \label{eq:Vietarian} Output fall time from V_{IHmin} to V_{ILmax} with a bus capacitance from 10 \ pF to 400 \ pF; \\ with up to 3 \ mA sink current at V_{OL1} \\ with up to 6 \ mA sink current at V_{OL2} \end{array}$	t <sub>of</sub>	– n/a	250 <sup>3)</sup> n/a	$20 + 0.1C_b^{(2)}$ $20 + 0.1C_b^{(2)}$	250 250 <sup>3)</sup>	ns
Input current each I/O pin with an input voltage between 0.4 V and 0.9V <sub>DDmax</sub>	li	-10	10	-10 <sup>4)</sup>	10 <sup>4)</sup>	μΑ
Capacitance for each I/O pin	Ci	-	10	-	10	pF

#### Table 3. Characteristics of the SDA and SCL I/O stages for I<sup>2</sup>C-bus devices

n/a = not applicable

1. Maximum  $V_{IH} = V_{DDmax} + 0.5 V$ 2.  $C_b$  = capacitance of one bus line in pF.

3. The maximum tr for the SDA and SCL bus lines quoted in Table 4 (300 ns) is longer than the specified maximum tof for the output stages (250 ns). This allows series protection resistors (Rs) to be connected between the SDA/SCL pins and the SDA/SCL bus lines as shown in Figure 37 without exceeding the maximum specified tf.

4. I/O pins of fast-mode devices must not obstruct the SDA and SCL lines if  $V_{DD}$  is switched off.

ТҮРЕ		I <sup>2</sup> C SLAVE ADDRESSES <sup>(1)</sup>		I <sup>2</sup> C SLAVE ADDRESSES <sup>(1)</sup>				
NUMBER	DESCRIPTION	A6	<b>A</b> 5	A4	A3	A2	A1	A0
-	General call address	0	0	0	0	0	0	0
-	Reserved addresses	0	0	0	0	Х	Х	Х
-	Reserved addresses	1	1	1	1	Х	Х	Х
CCR921	RDS/RBDS decoder	0	0	1	0	0	А	Α
NE5751	Audio processor for RF communication	1	0	0	0	0	0	Α
PCA1070	Programmable speech transmission IC	0	1	0	0	0	0	Α
PCA8516	Stand-alone OSD IC	1	0	1	1	1	0	1
PCA8581/C	128 × 8-bit EEPROM	1	0	1	0	Α	Α	Α
PCB2421	1K dual mode serial EEPROM	1	0	1	0	0	0	0
PCD3311C	DTMF/modem/musical tone generator	0	1	0	0	1	0	А
PCD3312C	DTMF/modem/musical tone generator	0	1	0	0	1	0	Α
PCD4440	Voice scrambler/descrambler for mobile telephones	1	1	0	1	1	1	Α
PCD5002	Pager decoder	0	1	0	0	1	1	1
PCD5096	Universal codec	0	0	1	1	0	А	Α
PCE84C467/8	8-bit CMOS auto-sync monitor controller	0	1	1	0	0	1	1
PCE84C882	8-bit microcontroller for monitor applications	0	1	1	0	0	1	1
PCE84C886	8-bit microcontroller for monitor applications	0	1	1	0	0	1	1
PCF2116	LCD controller/driver	0	1	1	1	0	1	Α
PCF8522/4	512 × 8-bit CMOS EEPROM	1	0	1	0	A	Α	Α
PCF8566	96-segment LCD driver 1:1 - 1:4 Mux rates	0	1	1	1	1	1	Α
PCF8568	LCD row driver for dot matrix displays	0	1	1	1	1	0	Α
PCF8569	LCD column driver for dot matrix displays	0	1	1	1	1	0	Α
PCF8570	$256 \times 8$ -bit static RAM	1	0	1	0	Α	Α	Α
PCF8573	Clock/calendar	1	1	0	1	0	Α	Α
PCF8574	8-bit remote I/O port (I <sup>2</sup> C-bus to parallel converter)	0	1	0	0	Α	Α	Α
PCF8574A	8-bit remote I/O port (I <sup>2</sup> C-bus to parallel converter)	0	1	1	1	Α	А	Α
PCF8576C	16-segment LCD driver 1:1 - 1:4 Mux rates	0	1	1	1	0	0	Α
PCF8577C	32/64-segment LCD display driver	0	1	1	1	0	1	0
PCF8578/9	Row/column LCD dot matrix driver/display	0	1	1	1	1	0	А
PCF8582/A	256 × 8-bit EEPROM	1	0	1	0	Α	А	Α
PCF8583	$256 \times 8$ -bit RAM/clock/calendar	1	0	1	0	0	0	Α
PCF8584	I <sup>2</sup> C-bus controller	Х	Х	Х	Х	Х	Х	Х
PCF8591	4-channel, 8-bit Mux ADC and one DAC	1	0	0	1	Α	Α	Α
PCF8593	Low-power clock calender	1	0	1	0	0	0	1
PCX8594	512 × 8-bit CMOS EEPROM	1	0	1	0	A	Α	Р
PCX8598	1024 × 8-bit CMOS EEPROM	1	0	1	0	Α	Р	Р
PDIUSB11	Universal serial bus	0	0	1	1	0	1	1
SAA1064	4-digit LED driver	0	1	1	1	0	Α	Α

### ASSIGNED I<sup>2</sup>C-BUS ADDRESSES (IN ALPHANUMERIC ORDER OF TYPE NUMBER)

PARAMETER	SYMBOL STANDARD-MODE I <sup>2</sup> C-BUS			FAST- I <sup>2</sup> C-	UNIT	
		Min.	Max.	Min.	Max.	
SCL clock frequency	f <sub>SCL</sub>	0	100	0	400	kHz
Bus free time between a STOP and START condition	t <sub>BUF</sub>	4.7	_	1.3	-	μs
Hold time (repeated) START condition. After this period, the first clock pulse is generated	t <sub>HD;STA</sub>	4.0	_	0.6	-	μs
LOW period of the SCL clock	t <sub>LOW</sub>	4.7	-	1.3	-	μs
HIGH period of the SCL clock	t <sub>HIGH</sub>	4.0	-	0.6	-	μs
Set-up time for a repeated START condition	t <sub>SU;STA</sub>	4.7	-	0.6	-	μs
Data hold time: for CBUS compatible masters (see NOTE, Section 9.1.3)	t <sub>HD;DAT</sub>	5.0	-	-	-	μs
for I <sup>2</sup> C–bus devices		01)	-	01)	0.9 <sup>2)</sup>	μs
Data set-up time	t <sub>SU;DAT</sub>	250	-	100 <sup>3)</sup>	-	ns
Rise time of both SDA and SCL signals	t <sub>r</sub>	-	1000	$20 + 0.1 C_b^{4)}$	300	ns
Fall time of both SDA and SCL signals	t <sub>f</sub>	-	300	$20 + 0.1 C_b^{(4)}$	300	ns
Set-up time for STOP condition	t <sub>SU;STO</sub>	4.0	-	0.6	-	μs
Capacitive load for each bus line	Cb	-	400	-	400	pF

#### Table 4. Characteristics of the SDA and SCL bus lines for I<sup>2</sup>C-bus devices

All values referred to V<sub>IHmin</sub> and V<sub>ILmax</sub> levels (see Table 3).
 A device must internally provide a hold time of at least 300 ns for the SDA signal (referred to the V<sub>IHmin</sub> of the SCL signal) in order to bridge the undefined region of the falling edge of SCL.

The maximum t<sub>HD:DAT</sub> has only to be met if the device does not stretch the LOW period (t<sub>LOW</sub>) of the SCL signal.
 A fast-mode l<sup>2</sup>C-bus device can be used in a standard-mode l<sup>2</sup>C-bus system, but the requirement t<sub>SU;DAT</sub> ≥ 250 ns must then be met. This will automatically be the case if the device does not stretch the LOW period of the SCL signal. If such a device does stretch the LOW period of the SCL signal, it must output the next data bit to the SDA line t<sub>r max</sub> + t<sub>SU;DAT</sub> = 1000 + 250 = 1250 ns (according to the standard-mode l<sup>2</sup>C-bus specification) before the SCL line is released.

4. C<sub>b</sub> = total capacitance of one bus line in pF.

### Appendix 'Solderen'

Gebruik voor het solderen een geschikte soldeerbout met een dunne tip. Gebruik soldeertin van 0,8 mm diameter. De bout moet goed warm zijn voordat je begint met solderen. Er mag geen druppeltje soldeer aan hangen. Let er bij het solderen van componenten goed op dat ze niet te langdurig achtereen verwarmd worden, anders kunnen er beschadigingen optreden. Bepaalde typen IC's zijn hier gevoelig voor. Dit is een goede reden voor beginners om met voetjes te werken.

Pas ook op met statische elektriciteit: je moet op een geschikt vlak werken en je even "aarden" om de statische electriciteit af te laten vloeien. En bewaar de chip in of op geschikt materiaal. Niet elk plastic is geschikt!

In het algemeen is het het beste om de componenten in onderstaande volgorde aan te brengen:

- draadbruggen
- liggende componenten
- staande componenten (zeer grote componenten pas later)
- controleer of draadbruggen geen problemen op gaan leveren want nu zijn ze meestal nog zonder veel schade te veranderen
- stekers en connectors
- voetjes en de grote componenten

De basisregel is om draadbruggen aan de componentzijde aan te brengen.

- Aan de bovenzijde kan je draadrestjes gebruiken. Voor de beginner is het verstandig om geïsoleerd draad te kiezen.
- Aan de koperzijde is geïsoleerd draad beslist een vereiste. Houd rekening met nog aan te brengen componenten!

De methode: eerst aan een kant vastzetten, dan doorlussen en strak trekken. Vervolgens het tweede punt solderen. Tenslotte de draad op maat knippen.

Draadbruggen solderen aan pinnen van componenten: de draad door een naastliggend gaatje lussen en buigen. Het is practisch om ze ook in dat naastliggende gaatje te solderen. Ook hier komen dan de draadbruggen aan de componentenzijde van de print te liggen.

Liggende componenten: eerst plaatsen en solderen, dan knippen.



### Appendix 'Experimenteren'

De opbouw van alle programmalistings in dit werkboek is in grote lijnen als volgt. De "primitieve" routines vinden we uiteraard aan het begin van de listing. Ze zijn vanuit de library (LIB) gekopieerd naar de listing. Dan volgen de toepassing-specifieke routines die vaak de primitieven aanroepen. Aan het eind van de listing vinden we het demo-gedeelte dat de vooraf gedefinieerde routines aanroept en de funktie realiseert.

Je wordt na het bouwen en testen uitgenodigd te experimenteren om goed met ByteForth vertrouwd te raken. Verander de inhoud van het demogedeelte. Verander ook de waarden (constanten) in de rest van de software als je dat van pas komt. Als de wijzigingen doen wat je wilde bereiken dan ben je op de goede weg met Forth.

Verander nooit de bestanden in de bibliotheek (de library in de subdirectory LIB) want deze bestanden kunnen ook door andere toepassingen aangeroepen worden. Of later nog nodig zijn. Deze routines zijn volledig getest en algemeen toepasbaar. Dat moet natuurlijk zo blijven. Kopieer ze daarom voor je eigen programma's eerst naar jouw listing en verander ze in die listing zoveel als je nodig vindt.

### Zelf ontwikkelen van toepassingen

De steek van de gaatjes op de AT51-print bedraagt 0.1"= 2,54 mm. Voor dit ontwikkelsysteem is gekozen voor voetjes voor alle toegepaste IC's. We gebruiken jumpers in plaats van dipswitches omdat die goedkoper zijn. Voor connectors, stekers en jumpers worden SIL- (single-in-line) of DIL- (dual-in-line-) strips gebruikt, zowel male als female.

Neem als connector voor een voedingsaansluiting drie pinnen van een SIL-strip. Kies de middelste als massaaansluiting en een van de buitenste voor de +, zet een merkteken en gebruik de kleuren rood en zwart. Dit helpt tegen het verkeerd om plaatsen. En toch kan het nog fout gaan. Zet desnoods ook iets hoogs naast de maleconnector op het bordje, aan de goede kant. Een betere beveiliging tegen verkeerd gepoolde spanningen geeft een diode in het nieuwe ontwerp.

Als je regelmatig een nieuw project opzet: maak dat dan eerst op een Solderless Breadboard. Het kost een beetje maar het is erg handig. Gebruik geïsoleerde draadjes en houdt ze vrij kort.

Let op: Monteer een ontkoppelcondensator over de voedingspinnen van iedere chip!

### Appendix 'Programmeren'

Ben je onbekend met Forth en wil je je eigen toepassingen maken dan is de spoedcursus in het ByteForth Handboek een goed begin. De Forth-gg heeft een cursusboek "De programmeertaal FORTH (ANS Forth), Cursus & Systematisch Overzicht", in het Nederlands geschreven door Albert Nijhof.

Forth is een programmeertaal met een interactieve compiler. Als je op interpreter-niveau een routine (in Forth: een colondefinitie) invoert dan wordt deze direct gecompileerd. In feite voeg je een uitbreiding aan de programmeertaal zelf toe. De gecompileerde routines zijn na invoer onmiddellijk beschikbaar en kunnen dan een voor een getest worden, zonder dat je last hebt van de rest van het programma. De stacks zijn rechtstreeks te benaderen. Assemblercode wordt "inline" geschreven en dus direct in de listing opgenomen. Met Forth kun je de hardware direct benaderen. De taal wordt dan ook veel toegepast voor machinebesturingen en bij de toepassing van microcontrollers.

ByteForth is een 8-bit-Forth die ook 16-bit-waarden kent. De syntax ligt dicht tegen de ANSI-standaard aan. Voor onze specifieke ontwikkelomgeving is er een beperkt aantal woorden toegevoegd.

Een ander kenmerk van Forth is dat een programma opgebouwd kan worden uit korte tot zeer korte routines, de colondefinities. Het maken van alleen maar korte routines wordt warm aanbevolen. In elke toepassing die in het werkboek beschreven is vinden we de "primitieve" routines aan het begin van de listing.

Als je een eigen toepassing maakt "leen" dan zoveel mogelijk van de gegeven primitieven uit deze bibliotheek! Bestudeer de gegeven listings. Zoek in die listings naar de voor het nieuwe project bruikbare routines.

### **Bibliotheekfuncties**

Voor dit werkboek zijn de primitieven die in de listings worden gebruikt uit de bibliotheek gekopieerd. Dat is noodzakelijk als je de primitieven op maat wilt maken. Je hebt in een programma niet altijd alle primitieven uit zo'n LIB-bestand nodig, dat spaart geheugen. Of misschien wil je veranderingen in de primitieven aanbrengen omdat je het gewoon beter weet. Maar verander de bestanden in LIB zelf niet!

### Opstartindicatie

Het is heel nuttig om bij elk programma een "koude start" op te nemen. Dat is, aan het begin van de hoofdroutine, een actie die aangeeft dat het systeem reageert. Zie ESW-01 waarin de acht LED's allemaal eerst even aan- en dan weer uitgezet worden. Of een motortje maakt een stelbeweging. Je kunt zelfs alleen voor dit doel een LED in je ontwerp opnemen.

### Appendix 'Het laden van bestanden'

De algemene struktuur van een programma is:

De regels 1, 7 en 9 zijn de "verplichte figuren": { 89C2051 TARGET } { SETUP } { RESET-VEC SET-VECTOR }.

### De AF-versie

- 1. Voeg 89C2051 TARGET, de bibliotheekbestanden en de programma- / demolisting samen in een ASCII of .TXT bestand, sla dit op in de subdirectory WORK (en / of 'EGEL) en laad dit (projecten ESW-01 t/m -19).
- 2. Type **89C2051 TARGET** en laad achtereenvolgens de bestanden uit de subdirectories (zie projecten ESW-20 en -21).
- 3. Maak een scriptfile volgens het volgend model en bewaar dit in de subdirectory WORK (en / of 'EGEL).

```
SEND( AT89C2051)
ESEND {path}\LIBx.FRT
ESEND {path}\LIBy.FRT
ESEND {toepassing}.FRT
```

Het scriptfile wordt geladen door <F8> <filenaam> in te toetsen. Andere scriptfilecommando's vinden we in de Server, in het Help menu. Toets achtereenvolgens: <F10> H C .

### De PC-versie

In deze versie kunnen we de bibliotheekbestanden vanuit het toepassingsprogramma zelf automatisch laten laden door het woord **NEEDS** te gebruiken. **NEEDS** weet de subdirectory LIB te vinden.

```
\ Kop van het bestand TOEPASSING.FRT
```

AT98C2051 TARGET ..... NEEDS LIBX.FRT NEEDS LIBY.FRT ..... \ de rest van TOEPASSING ..... \ dus met SETUP in de hoofdroutine ..... \ en daarna RESET-VEC en SET-VECTOR

### Appendix 'Internet-adressen'

Hier volgen enkele websites waar datasheets of produktinformatie te vinden is. Datasheets staan gewoonlijk in het PDF-formaat waarvoor een bladerprogramma nodig is: de Adobe Acrobat Reader. Dit programma is gratis te downloaden van <u>http://www.adobe.com/Acrobat/Acrobat0.htm</u>, zowel voor de Macintosh als voor Windows 3.x/95/98.

Deze URLs zijn geldig op dit moment (30 okt 2004), maar door de veranderlijke aard van websites kunnen de gelinkte paginas ophouden te bestaan en wordt dit "dode" links. Het intikken van een zoekterm in een zoekmachine kan aanknopingspunten bieden.

Aansluitgegevens van alle denkbare chips alsmede URL's van zeer veel halfgeleiderfabrikanten zijn te vinden op: <u>http://www.xs4all.nl/~ganswijk/chipdir/index.htm</u> of zoek op <u>http://www.google.com</u> naar "GIICM ".

Informatie over de AT89Cx051 van de firma Atmel is te vinden op: http://www.atmel.com/atmel/acrobat/doc0368.pdf

Informatie en datasheets over andere 8051-afgeleiden: <u>http://www.keil.com/dd/list\_ca.asp</u> <u>http://www.infineon.com/cgi/ecrm.dll/ecrm/scripts/prod\_ov.jsp?oid=13738</u> <u>http://www.semiconductors.philips.com/markets/mms/products/microcontrollers/product\_catalog/8bit\_80c51/ind</u> <u>ex.html</u>

Alles over het Philips I<sup>2</sup>C-protocol : <u>http://www.semiconductors.philips.com/markets/mms/protocols/i2c/</u> of zoek op <u>http://www.semiconductors.philips.com</u> naar I2C. Enkele fraaie presentaties met duidelijke afbeeldingen over I2C zijn hier te vinden: <u>http://www.standardproducts.philips.com/products/collateral/i2c/pdf/presentation\_i2c.pdf</u> <u>http://www.standardproducts.philips.com/products/collateral/i2c/pdf/presentation\_i2c.pdf</u>

Over de opbouw van het RC5-protocol : <u>http://www.ustr.net/infrared/infrared1.shtml</u> <u>http://www.vishay.com/docs/fmod\_data\_formats.pdf</u>

Informatie over LCD's met een karaktergenerator volgens de Hitachi- (HD44780-) standaard, o.a. datasheets, instructies, aansluitingen en karaktertabellen: <u>http://www.doc.ic.ac.uk/~ih/doc/lcd/</u> <u>http://www.eio.com/lcdintro.htm</u> <u>http://www.eio.com/datashet.htm</u> <u>http://home.iae.nl/users/pouweha/lcd/lcd.shtml</u>

Over stappenmotoren en de principes van hun stuurelektronica : <u>http://eio.com/jasstep.htm</u>

Een korte Forthcursus: http://www.albany.net/~hello/inching.htm

Meer informatie over Forth vind je op http://www.forth.hccnet.nl/

### Appendix 'TLC 549'

A/D converter, ESW-04 -05 -07 -13



#### description

The TLC548 and TLC549 are CMOS analog-to-digital converter (ADC) integrated circuits built around an 8-bit switched-capacitor successive-approximation ADC. These devices are designed for serial interface with a microprocessor or peripheral through a 3-state data output and an analog input. The TLC548 and TLC549 use only the input/output clock (I/O CLOCK) input along with the chip select  $(\overline{CS})$  input for data control. The maximum I/O CLOCK input frequency of the TLC548 is 2.048 MHz, and the I/O CLOCK input frequency of the TLC549 is specified up to 1.1 MHz.

AVAILABLE OPTIONS						
	PACKAGE					
Τ <sub>Α</sub>	SMALL OUTLINE (D)	PLASTIC DIP (P)				
0°C to 70°C	TLC548CD TLC549CD	TLC548CP TLC549CP				
-40°C to 85°C	TLC548ID TLC549ID	TLC548IP TLC549IP				

#### operating sequence



The conversion cycle, which requires 36 internal system clock periods (17  $\mu$ s maximum), is initiated with the eighth I/O clock pulse trailing edge after  $\overline{CS}$  goes low for the channel whose address exists in memory at the time. The most significant bit (A7) is automatically placed on the DATA OUT bus after  $\overline{CS}$  is brought low. The remaining seven bits (A6–A0) NOTES: A.

В. are clocked out on the first seven I/O clock falling edges. B7-B0 follows in the same manner

### Appendix 'MAX 232'

**RS232**, ESW-06 -07 -13

### MAX232, MAX232I DUAL EIA-232 DRIVER/RECEIVER

SLLS047G - FEBRUARY 1989 - REVISED AUGUST 1998

- Operates With Single 5-V Power Supply
- LinBiCMOS<sup>™</sup> Process Technology
- Two Drivers and Two Receivers
- ±30-V Input Levels
- Low Supply Current . . . 8 mA Typical
- Meets or Exceeds TIA/EIA-232-F and ITU Recommendation V.28
- Designed to be Interchangeable With Maxim MAX232
- Applications
  - TIA/EIA-232-F Battery-Powered Systems Terminals Modems Computers
- ESD Protection Exceeds 2000 V Per MIL-STD-883, Method 3015
- Package Options Include Plastic Small-Outline (D, DW) Packages and Standard Plastic (N) DIPs

#### description

The MAX232 device is a dual driver/receiver that includes a capacitive voltage generator to supply EIA-232 voltage levels from a single 5-V supply. Each receiver converts EIA-232 inputs to 5-V TTL/CMOS levels. These receivers have a typical threshold of 1.3 V and a typical hysteresis of 0.5 V, and can accept  $\pm$ 30-V inputs. Each driver converts TTL/CMOS input levels into EIA-232 levels. The driver, receiver, and voltage-generator functions are available as cells in the Texas Instruments LinASIC<sup>TM</sup> library.

The MAX232 is characterized for operation from 0°C to 70°C. The MAX232I is characterized for operation from -40°C to 85°C.



D, DW, OR N PACKAGE



<sup>†</sup> This symbol is in accordance with ANSI/IEEE Std 91-1984 and IEC Publication 617-12.

AVAILABLE OPTIONS					
	PACKAGED DEVICES				
TA	SMALL SMALL OUTLINE OUTLINE (D) (DW)		PLASTIC DIP (N)		
0°C to 70°C	MAX232D <sup>‡</sup>	MAX232DW <sup>‡</sup>	MAX232N		
-40°C to 85°C	MAX232ID‡	MAX232IDW <sup>‡</sup>	MAX232IN		

<sup>‡</sup> This device is available taped and reeled by adding an R to the part number (i.e., MAX232DR).



Please be aware that an important notice concerning availability, standard warranty, and use in critical applications of Texas Instruments semiconductor products and disclaimers thereto appears at the end of this data sheet.

### recommended operating conditions

		MIN	NOM	MAX	UNIT
Supply voltage, V <sub>CC</sub>		4.5	5	5.5	V
High-level input voltage, V <sub>IH</sub> (T1IN,T2IN)		2			V
Low-level input voltage, VIL (T1IN, T2IN)				0.8	V
Receiver input voltage, R1IN, R2IN				±30	V
Operating free air temperature T	MAX232	0		70	ŝ
Operating nee-air temperature, ra	MAX232I -40	85	°С		

### electrical characteristics over recommended ranges of supply voltage and operating free-air temperature range (unless otherwise noted)

	PARAMETER		TEST CO	ONDITIONS	MIN	TYP†	МАХ	UNIT
Valu	High lovel output veltage	T1OUT, T2OUT	$R_L = 3 k\Omega$ to GND		5	7		м
⊻он	High-level output voltage	R1OUT, R2OUT	I <sub>OH</sub> = -1 mA		3.5			v
Vei	1 t t	T1OUT, T2OUT	$R_L = 3 k\Omega$ to GND			-7	-5	v
VOL	Low-level output voltage+	R1OUT, R2OUT	I <sub>OL</sub> = 3.2 mA				0.4	v
V <sub>IT+</sub>	Receiver positive-going input threshold voltage	R1IN, R2IN	$V_{CC} = 5 V,$	$T_A = 25^{\circ}C$		1.7	2.4	v
VIT-	Receiver negative-going input threshold voltage	R1IN, R2IN	$V_{CC} = 5 V,$	$T_A = 25^{\circ}C$	0.8	1.2		v
V <sub>hys</sub>	Input hysteresis voltage	R1IN, R2IN	$V_{CC} = 5 V$		0.2	0.5	1	V
rj	Receiver input resistance	R1IN, R2IN	V <sub>CC</sub> = 5,	$T_A = 25^{\circ}C$	3	5	7	kΩ
r <sub>o</sub>	Output resistance	T1OUT, T2OUT	$V_{S+} = V_{S-} = 0,$	$V_{O} = \pm 2 V$	300			Ω
los§	Short-circuit output current	T1OUT, T2OUT	V <sub>CC</sub> = 5.5 V,	$V_{O} = 0$		±10		mA
IIS	Short-circuit input current	T1IN, T2IN	$V_{\parallel} = 0$				200	μA
Icc	Supply current		$V_{CC} = 5.5 \text{ V},$ $T_A = 25^{\circ}\text{C}$	All outputs open,		8	10	mA

 <sup>†</sup> All typical values are at V<sub>CC</sub> = 5 V, T<sub>A</sub> = 25°C.
 <sup>‡</sup> The algebraic convention, in which the least positive (most negative) value is designated minimum, is used in this data sheet for logic voltage levels only.

§ Not more than one output should be shorted at a time.

### switching characteristics, $V_{CC}$ = 5 V, $T_A$ = 25°C

	PARAMETER	TEST CONDITIONS	ΜΙΝ ΤΥΡ ΜΑΧ	UNIT
tPLH(R)	Receiver propagation delay time, low- to high-level output	See Figure 1	500	ns
tPHL(R)	Receiver propagation delay time, high- to low-level output	See Figure 1	500	ns
SR	Driver slew rate	$R_L = 3 k\Omega$ to 7 kΩ, See Figure 2	30	V/µs
SR(tr)	Driver transition region slew rate	See Figure 3	3	V/µs

### **APPLICATION INFORMATION**



**Typical Operating Circuit** 

### Appendix 'BUK100-50GL'

FET. ESW-08 **Philips Semiconductors** 

### **PowerMOS transistor** Logic level TOPFET

#### DESCRIPTION

Monolithic temperature and overload protected logic level power MOSFET in a 3 pin plastic envelope, intended as a general purpose switch for automotive systems and other applications.

### **APPLICATIONS**

General controller for driving

- lamps ۰
- motors ٠
- ٠ solenoids ۰ heaters

### **FEATURES**

- . Vertical power DMOS output stage
- Low on-state resistance Overload protection against
- over temperature
- Overload protection against short circuit load Latched overload protection
- reset by input
- 5 V logic compatible input level
- Control of power MOSFET and supply of overload protection circuits derived from input Low operating input current ESD protection on input pin

- Overvoltage clamping for turn off of inductive loads

### **PINNING - TO220AB**

PIN	DESCRIPTION
1	input
2	drain
3	source
tab	drain

### QUICK REFERENCE DATA

SYMBOL	PARAMETER	MAX.	UNIT
$\begin{array}{c} V_{DS} \\ I_{D} \\ P_{D} \\ T_{j} \\ R_{DS(ON)} \end{array}$	Continuous drain source voltage Continuous drain current Total power dissipation Continuous junction temperature Drain-source on-state resistance $V_{IS} = 5 V$	50 13.5 40 150 125	V Α Ψ°C mΩ

### FUNCTIONAL BLOCK DIAGRAM



### **PIN CONFIGURATION**

tab  $\bigcirc$ 





### **Product specification**

### **BUK100-50GL**

PowerMOS transistor	
Logic level TOPFET	

### BUK100-50GL

### LIMITING VALUES

Limiting values in accordance with the Absolute Maximum Rating System (IEC 134)

SYMBOL	PARAMETER	CONDITIONS	MIN.	MAX.	UNIT
V <sub>DSS</sub>	Continuous off-state drain source	$V_{IS} = 0 V$	-	50	V
	voltage <sup>1</sup>				
V <sub>IS</sub>	Continuous input voltage	-	0	6	V
I <sub>D</sub>	Continuous drain current	$T_{mb} \le 25 \text{ °C}; V_{IS} = 5 \text{ V}$	-	13.5	А
I <sub>D</sub>	Continuous drain current	$T_{mb} \le 100 \text{ °C}; V_{IS} = 5 \text{ V}$	-	8.5	А
I <sub>DRM</sub>	Repetitive peak on-state drain current	$T_{mb} \le 25 \ C; \ V_{IS} = 5 \ V$	-	54	А
PD	Total power dissipation	T <sub>mb</sub> ≤ 25 °C	-	40	W
T <sub>sta</sub>	Storage temperature	-	-55	150	°C
Tj	Continuous junction temperature <sup>2</sup>	normal operation	-	150	°C
T <sub>sold</sub>	Lead temperature	during soldering	-	250	°C

### **OVERLOAD PROTECTION LIMITING VALUES**

With the protection supply provided via the input pin, TOPFET can protect itself from two types of overload.

SYMBOL	PARAMETER	CONDITIONS	MIN.	MAX.	UNIT
VISP	Protection supply voltage <sup>3</sup>	for valid protection	4	-	V
	Over temperature protection				
V <sub>DDP(T)</sub>	Protected drain source supply voltage	$V_{IS} = 5 V$	-	50	V
V <sub>DDP(P)</sub> P <sub>DSM</sub>	Short circuit load protection Protected drain source supply voltage <sup>4</sup> Instantaneous overload dissipation	$V_{IS} = 5 V$ $T_{mb} = 25 °C$	-	35 0.6	V kW

### **OVERVOLTAGE CLAMPING LIMITING VALUES**

At a drain source voltage above 50 V the power MOSFET is actively turned on to clamp overvoltage transients.

SYMBOL	PARAMETER	CONDITIONS	MIN.	MAX.	UNIT
I <sub>DROM</sub>	Repetitive peak clamping current	$V_{IS} = 0 V$	-	15	А
E <sub>DSM</sub>	Non-repetitive clamping energy	$T_{mb} \le 25 \ ^{\circ}C; I_{DM} = 15 \ A;$	-	200	mJ
		$V_{DD} \leq 20 \text{ V}$ ; inductive load			
EDRM	Repetitive clamping energy	$T_{mb} \le 95 \ ^{\circ}C; I_{DM} = 4 A;$	-	20	mJ
		$V_{\text{DD}} \leq 20 \text{ V}; \text{ f} = 250 \text{ Hz}$			

### ESD LIMITING VALUE

SYMBOL	PARAMETER	CONDITIONS	MIN.	MAX.	UNIT
Vc	Electrostatic discharge capacitor voltage	Human body model; C = 250 pF; R = 1.5 k $\Omega$	-	2	kV

<sup>1</sup> Prior to the onset of overvoltage clamping. For voltages above this value, safe operation is limited by the overvoltage clamping energy.

<sup>2</sup> A higher T<sub>j</sub> is allowed as an overload condition but at the threshold T<sub>j(TO)</sub> the over temperature trip operates to protect the switch.

<sup>3</sup> The input voltage for which the overload protection circuits are functional.

<sup>4</sup> The device is able to self-protect against a short circuit load providing the drain-source supply voltage does not exceed V<sub>DDP(P)</sub> maximum. For further information, refer to OVERLOAD PROTECTION CHARACTERISTICS.

### Appendix 'BC517'

darlington, ESW-09-15

### NPN Silicon Darlington Transistor

- High current gain
- High collector current
- Complementary type: BC 516 (PNP)



BC 517

Туре	Marking	Ordering Code	Pin Configuration		Package <sup>1)</sup>	
	_		1	2	3	_
BC 517	-	Q62702-C825	С	В	Е	TO-92

### **Maximum Ratings**

Parameter	Symbol	Values	Unit
Collector-emitter voltage	VCEO	30	V
Collector-base voltage	Vсво	40	
Emitter-base voltage	VEBO	10	
Collector current	<i>I</i> c	500	mA
Peak collector current	Ісм	800	
Base current	Ів	100	
Peak base current	Івм	200	
Total power dissipation, $T_{\rm C}$ = 66 °C	Ptot	625	mW
Junction temperature	Tj	150	°C
Storage temperature range	$T_{ m stg}$	- 65 + 150	)

### **Thermal Resistance**

Junction - ambient	$R_{ m th}$ JA	≤ <b>200</b>	K/W
Junction - case <sup>2)</sup>	$R_{ m th}$ JC	≤ <b>135</b>	

### **Electrical Characteristics**

at  $T_A = 25$  °C, unless otherwise specified.

Parameter	Symbol	Values			Unit	
		min.	typ.	max.	1	
DC characteristics		1				
Collector-emitter breakdown voltage Ic = 10 mA	$V_{(BR)CE0}$	30	-	-	V	
Collector-base breakdown voltage $I_{\rm C} = 100 \ \mu A$	$V_{(BR)CB0}$	40	-	-	_	
Emitter-base breakdown voltage $I_E = 10 \ \mu A$	$V_{(BR)EB0}$	10	_	-	_	
Collector cutoff current $V_{CB} = 30 \text{ V}$ $V_{CB} = 30 \text{ V}, T_A = 150 \text{ °C}$	Ісво			100 10	nA μA	
Emitter cutoff current $V_{\text{EB}} = 4 \text{ V}$	Іево	-	-	100	nA	
DC current gain $I_{\rm C} = 20 \text{ mA}; V_{\rm CE} = 2 \text{ V}^{1)}$	hfe	30 000	-	-	-	
Collector-emitter saturation voltage <sup>1)</sup> $I_{\rm C} = 100$ mA; $I_{\rm B} = 0.1$ mA	VCEsat	-	-	1	V	
Base-emitter voltage <sup>1)</sup> $I_{C} = 10 \text{ mA}; V_{CE} = 5 \text{ V}$	VBE	-	-	1.4		

### Appendix 'ULN2803'

driver, ESW-11

# MOTOROLA

### **Octal High Voltage, High Current Darlington Transistor Arrays**

The eight NPN Darlington connected transistors in this family of arrays are ideally suited for interfacing between low logic level digital circuitry (such as TTL, CMOS or PMOS/NMOS) and the higher current/voltage requirements of lamps, relays, printer hammers or other similar loads for a broad range of computer, industrial, and consumer applications. All devices feature open-collector outputs and free wheeling clamp diodes for transient suppression.

The ULN2803 is designed to be compatible with standard TTL families while the ULN2804 is optimized for 6 to 15 volt high level CMOS or PMOS.

package, unless otherwise noted.)	0,		
Rating	Symbol	Value	Unit
Output Voltage	Vo	50	V
Input Voltage (Except ULN2801)	VI	30	V
Collector Current – Continuous	lc	500	mA
Base Current – Continuous	ΙB	25	mA
Operating Ambient Temperature Range	TA	0 to +70	°C
Storage Temperature Range	T <sub>stg</sub>	-55 to +150	°C
Junction Temperature	Тј	125	°C

**MAXIMUM RATINGS** (T<sub>A</sub> = 25°C and rating apply to any one device in the

 $R_{\theta JA} = 55^{\circ}C/W$ Do not exceed maximum current limit per driver.

#### **ORDERING INFORMATION**

	(		
Device	Input Compatibility	V <sub>CE</sub> (Max)/I <sub>C</sub> (Max)	Operating Temperature Range
ULN2803A ULN2804A	TTL, 5.0 V CMOS 6 to 15 V CMOS, PMOS	50 V/500 mA	$T_A = 0$ to + 70°C

Order this document by ULN2803/D



### **OCTAL PERIPHERAL DRIVER ARRAYS**

SEMICONDUCTOR **TECHNICAL DATA** 



#### **PIN CONNECTIONS**



### ULN2803 ULN2804

Characteristic		Symbol	Min	Тур	Max	Unit
$\begin{array}{l} \mbox{Output Leakage Current (Figure 1)} \\ (V_O = 50 \ V, \ T_A = +70^{\circ} \ C) \\ (V_O = 50 \ V, \ T_A = +25^{\circ} \ C) \\ (V_O = 50 \ V, \ T_A = +70^{\circ} \ C, \ V_I = 6.0 \ V) \\ (V_O = 50 \ V, \ T_A = +70^{\circ} \ C, \ V_I = 1.0 \ V) \end{array}$	All Types All Types ULN2802 ULN2804	ICEX	- - - -		100 50 500 500	μA
$\label{eq:constraint} \begin{array}{l} \mbox{Collector-Emitter Saturation Voltage (Figure 2)} \\ (I_C = 350 \mbox{ mA}, I_B = 500 \mbox{ \muA}) \\ (I_C = 200 \mbox{ mA}, I_B = 350 \mbox{ \muA}) \\ (I_C = 100 \mbox{ mA}, I_B = 250 \mbox{ \muA}) \end{array}$	All Types All Types All Types	V <sub>CE(sat)</sub>		1.1 0.95 0.85	1.6 1.3 1.1	V
Input Current – On Condition (Figure 4) (V <sub>I</sub> = 17 V) (V <sub>I</sub> = 3.85 V) (V <sub>I</sub> = 5.0 V) (V <sub>I</sub> = 12 V)	ULN2802 ULN2803 ULN2804 ULN2804 ULN2804	l(on)		0.82 0.93 0.35 1.0	1.25 1.35 0.5 1.45	mA
Input Voltage – On Condition (Figure 5) $(V_{CE} = 2.0 V, I_C = 300 mA)$ $(V_{CE} = 2.0 V, I_C = 200 mA)$ $(V_{CE} = 2.0 V, I_C = 250 mA)$ $(V_{CE} = 2.0 V, I_C = 300 mA)$ $(V_{CE} = 2.0 V, I_C = 125 mA)$ $(V_{CE} = 2.0 V, I_C = 200 mA)$ $(V_{CE} = 2.0 V, I_C = 275 mA)$ $(V_{CE} = 2.0 V, I_C = 350 mA)$	ULN2802 ULN2803 ULN2803 ULN2803 ULN2804 ULN2804 ULN2804 ULN2804 ULN2804	VI(on)	- - - - - -	- - - - - -	13 2.4 2.7 3.0 5.0 6.0 7.0 8.0	V
Input Current – Off Condition (Figure 3) (I <sub>C</sub> = 500 $\mu$ A, T <sub>A</sub> = +70°C)	All Types	I(off)	50	100	-	μΑ
DC Current Gain (Figure 2) (V <sub>CE</sub> = 2.0 V, I <sub>C</sub> = 350 mA)	ULN2801	hFE	1000	-	-	-
Input Capacitance		CI	-	15	25	pF
Turn–On Delay Time (50% E <sub>I</sub> to 50% E <sub>O</sub> )		ton	-	0.25	1.0	μs
Turn–Off Delay Time (50% El to 50% EO)		toff	-	0.25	1.0	μs
Clamp Diode Leakage Current (Figure 6) (V <sub>R</sub> = 50 V)	$T_A = +25^{\circ}C$ $T_A = +70^{\circ}C$	IR	-	-	50 100	μA
Clamp Diode Forward Voltage (Figure 7) (I <sub>F</sub> = 350 mA)		VF	-	1.5	2.0	V

**ELECTRICAL CHARACTERISTICS** (T<sub>A</sub> = 25°C, unless otherwise noted)

### Figure 12. Representative Schematic Diagrams





### Appendix 'L293D'

### driver, ESW-12



## L293D L293DD

### PUSH-PULL FOUR CHANNEL DRIVER WITH DIODES

- 600mA OUTPUT CURRENT CAPABILITY PER CHANNEL
- 1.2A PEAK OUTPUT CURRENT (non repetitive) PER CHANNEL
- ENABLE FACILITY
- OVERTEMPERATURE PROTECTION
- LOGICAL "0" INPUT VOLTAGE UP TO 1.5 V (HIGH NOISE IMMUNITY)
- INTERNAL CLAMP DIODES

### DESCRIPTION

The Device is a monolithic integrated high voltage, high current four channel driver designed to accept standard DTL or TTL logic levels and drive inductive loads (such as relays solenoides, DC and stepping motors) and switching power transistors.

To simplify use as two bridges each pair of channels is equipped with an enable input. A separate supply input is provided for the logic, allowing operation at a lower voltage and internal clamp diodes are included.

This device is suitable for use in switching applications at frequencies up to 5 kHz.

#### **BLOCK DIAGRAM**



The L293D is assembled in a 16 lead plastic packaage which has 4 center pins connected together and used for heatsinking

The L293DD is assembled in a 20 lead surface mount which has 8 center pins connected together and used for heatsinking.



PIN CONNECTIONS (Top view)



**ELECTRICAL CHARACTERISTICS** (for each channel,  $V_S$  = 24 V,  $V_{SS}$  = 5 V,  $T_{amb}$  = 25 °C, unless otherwise specified)

Symbol	Parameter	Test Conditions	Min.	Тур.	Max.	Unit
Vs	Supply Voltage (pin 10)		V <sub>SS</sub>		36	V
V <sub>SS</sub>	Logic Supply Voltage (pin 20)		4.5		36	V
Is	Total Quiescent Supply Current	$V_i=L\ ;\ I_O=0\ ;\ V_{en}=H$		2	6	mA
	(pin 10)	$V_i = H$ ; $I_O = 0$ ; $V_{en} = H$		16	24	mA
		$V_{en} = L$			4	mA
I <sub>SS</sub>	Total Quiescent Logic Supply	$V_i = L \ ; \ I_O = 0 \ ; \ V_{en} = H$		44	60	mA
	Current (pin 20)	$V_i = H \ ; \ I_O = 0 \ ; \ V_{en} = H$		16	22	mA
		V <sub>en</sub> = L		16	24	mA
VIL	Input Low Voltage (pin 2, 9, 12, 19)		- 0.3		1.5	V
VIH	Input High Voltage (pin 2, 9,	$V_{SS} \le 7 V$	2.3		$V_{SS}$	V
	12, 19)	V <sub>SS</sub> > 7 V	2.3		7	V
IIL	Low Voltage Input Current (pin 2, 9, 12, 19)	$V_{IL} = 1.5 V$			- 10	μA
Цн	High Voltage Input Current (pin 2, 9, 12, 19)	$2.3~V \leq V_{IH} \leq V_{SS} - 0.6~V$		30	100	μA
V <sub>en L</sub>	Enable Low Voltage (pin 1, 11)		- 0.3		1.5	V
V <sub>en H</sub>	Enable High Voltage	$V_{SS} \le 7 V$	2.3		V <sub>SS</sub>	V
	(pin 1, 11)	V <sub>SS</sub> > 7 V	2.3		7	V
l <sub>en L</sub>	Low Voltage Enable Current (pin 1, 11)	V <sub>en L</sub> = 1.5 V		- 30	- 100	μA
l <sub>en H</sub>	High Voltage Enable Current (pin 1, 11)	$2.3~V \leq V_{en~H} \leq V_{SS} - 0.6~V$			± 10	μA
V <sub>CE(sat)H</sub>	Source Output Saturation Voltage (pins 3, 8, 13, 18)	I <sub>O</sub> = - 0.6 A		1.4	1.8	V
V <sub>CE(sat)L</sub>	Sink Output Saturation Voltage (pins 3, 8, 13, 18)	I <sub>O</sub> = + 0.6 A		1.2	1.8	V
VF	Clamp Diode Forward Voltage	$I_0 = 600 nA$		1.3		V
tr	Rise Time (*)	0.1 to 0.9 V <sub>O</sub>		250		ns
t <sub>f</sub>	Fall Time (*)	0.9 to 0.1 V <sub>O</sub>		250		ns
t <sub>on</sub>	Turn-on Delay (*)	0.5 V <sub>i</sub> to 0.5 V <sub>O</sub>		750		ns
t <sub>off</sub>	Turn-off Delay (*)	0.5 V <sub>i</sub> to 0.5 V <sub>O</sub>		200		ns

(\*) See fig. 1.

### Appendix 'SFH506'

### IR-ontvanger/demodulator, ESW-14

De IR-ontvanger SFH506-36 van Siemens kan worden vervangen door IR-ontvanger TSOP1836 en TSOP4836 van Vishay. Deze componenten zijn nagenoeg exact gelijk aan de SFH506-36.

### IR-Empfänger/Demodulator-Baustein IR-Receiver/Demodulator Device

### SFH 506



Maße in mm, wenn nicht anders angegeben/Dimensions in mm, unless otherwise specified.

### Wesentliche Merkmale

- Fotodiode mit integriertem Verstärker
- Angepaßt an verschiedene Trägerfrequenzen
- Gehäuse schwarz eingefärbt: Verguß optimiert für eine Wellenlänge von 950 nm
- Hohe Störsicherheit
- Geringe Stromaufnahme
- 5 V Betriebsspannung
- Hohe Empfindlichkeit
- TTL und CMOS kompatibel
- Verwendbar bis zu einem Tastverhältnis ≤ 40 %

### Anwendungen

• Empfänger für IR-Fernsteuerungen

### Features

- Photodiode with hybride integrated circuit
- Available for several carrier frequencies
- Black epoxy resin, daylight filter optimized for 950 nm
- High immunity against ambient light
- Low power consumption
- 5 V supply voltage
- High sensitivity (internal shield case)
- TTL and CMOS compatibility
- Continuous transmission possible  $(t_{p}/T \le 0.4)$

### Applications

• IR-remote control preamplifier modules

Тур	Trägerfrequ.	Bestellnr.	Тур	Trägerfrequ.	Bestellnr.
Туре	Carrier Frequency kHz	Ordering Code	Туре	Carrier Frequency kHz	Ordering Code
SFH 506-30	30	Q62702-P1196	SFH 506-38	38	Q62702-P1199
SFH 506-33	33	Q62702-P1197	SFH 506-40	40	Q62702-P1200
SFH 506-36	36	Q62702-P1198	SFH 506-56	56	Q62702-P1201

#### Grenzwerte Maximum Ratings

Bezeichnung Description		Symbol Symbol	Wert Value	Einheit Unit
Betriebs- und Lagertemperatu Operation and storage temper	$T_{\rm A},  T_{\rm stg}$	- 25 + 85	°C	
Sperrschichttemperatur Junction temperature range		T <sub>j</sub>	100	°C
Löttemperatur Lötstelle 2 mm vom Gehäuse; Soldering temperature soldering joint $\ge 2$ mm distance package, soldering time $t \le 5$ s	Lötzeit $t \le 5$ s se from	T <sub>S</sub>	260	°C
Betriebsspannung Supply voltage	Pin 2	Vs	- 0.3 + 6.0	V
Betriebsstrom Supply current	Pin 2	I <sub>CC</sub>	5	mA
Ausgangsspannung Output voltage	Pin 3	V <sub>OUT</sub>	- 0.3 + 6.0	V
Ausgangsstrom Output current	Pin 3	I <sub>OUT</sub>	5	mA
Verlustleistung Total power dissipation $T_A \le 85 \ ^{\circ}C$		P <sub>tot</sub>	50	mW







Blockschaltbild Block Diagram

### Appendix 'S101S101'

Solid state relais, ESW-14

#### S102S01/S102S02/S202S01/S202S02

# S102S01/S102S02 S202S01/S202S02

### Features

- 1. High radiation resin mold package
- 2. RMS ON-state current I<sub>T</sub>: 8 Arms at T<sub>C</sub> <= 80°C (With heat sink)
- 3. Built-in zero-cross circuit (\$102\$02/\$202\$02)
- 4. High repetitive peak OFF-state voltage S102S01/S102S02 V<sub>DRM</sub>: MIN. 400V S202S01/S202S02 V<sub>DRM</sub>: MIN. 600V
- 5. Isolation voltage between input and output (Viso: 4 000V ms)
- 6. Approved by CSA, No. LR63705 Recognized by UL, file No. E94758

### Applications

- 1. Automatic vending machines, programmable controllers
- 2. Amusement equipment

#### Model Line-ups

	For 100V lines	For 200V lines
For phase control No built-in zero-cross circuit	S102S01	S202S01
Built-in zero-cross circuit	S102S02	S202S02

### Absolute Maximum Ratings

### SIP Type SSR for Medium Power Control

Outline Dimensions

(Unit : mm)



#### $(Ta = 25^{\circ}C)$

	-					-	
Parameter		Symbol	Rat \$102\$01 \$102\$02	s202801 \$202802	Unit	*1 $T_C \le 80^{\circ}C$ *2 50Hz sine wave, $T_j = 25^{\circ}C$	
Lunat	Forward current	IF	50		mA	*3 60Hz AC for 1 minute,	
Input	Reverse voltage	V <sub>R</sub>	6		v	40 to 60% RH, Apply voltages	
	*1RMS ON-state current	IT	8		A rms	<ul> <li>between input and output, by the dielectric withstand</li> </ul>	
	*2Peak one cycle surge current	I surge	80		Α	voltage tester with zero-	
0	Repetitive peak OFF-state voltage	VDRM	400	600	V	cross circuit.	
Output	Non-repetitive peak OFF-state voltage	V <sub>DSM</sub>	400	600	v	<ul> <li>(Input and output shall be shorted respectively.)</li> </ul>	
	Critical rate of rise of ON-state current	dI/dt	50		A/μ s	(Note)	
	Operating frequency	f	45 to 65		Hz	When the isolation voltage	
*3 Isolation voltage		Viso	4 000		V rms	is necessary at using external heat sink, please use the insulation sheet. *4 For 10 seconds	
Operating temperature		T opr	- 25 to + 100		°C		
Storage temperature		T stg	- 30 to + 125		°C		
*4Solderin	g temperature	T sol	26	0	°C	-	

" In the absence of confirmation by device specification sheets, SHARP takes no responsibility for any defects that occur in equipment using any of SHARP's devices, shown in catalogs, data books, etc. Contact SHARP in order to obtain the latest version of the device specification sheets before using any SHARP's device."

### Electro-optical Characteristics

#### $(Ta = 25^{\circ}C)$

Parameter			Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Input	Forward voltage		V <sub>F</sub>	$I_F = 20 m A$	-	1.2	1.4	V
mput	Reverse current		$I_R$	$V_R = 3V$	-	-	10.4	А
	Repetitive peak OFF-sta	ate current	I drm	$V_D = V_{DRM}$	-	-	10.4	А
	ON-state voltage		V <sub>T</sub>	Resistance load $I_F = 20mA, I_T = 2Arms$	-	-	1.5	V rms
	Holding current		I <sub>H</sub>	-	-	-	50	mA
Output	Critical rate of rise of OFF-state voltage		dV/dt	$V_D = 2/3 \bullet V_{DRM}$	30	-	-	V/µ s
	Critical rate of rise of commutating OFF-state voltage		(dV/dt) <sub>C</sub>	$T_j = 125^{\circ}C, dI_T/dt = -4.0 A/ms, V_D = 400 V$	5	-	-	V/μ s
	Zero-cross voltage	S102S02 S202S02	V ox	$I_F = 8mA$	-	-	35	V
	Minimum \$102\$01 \$202\$01		т	$V_D = 12V, R_L = 30\Omega$	-	-	8	mA
	current	S102S02 S202S02	1 FT	$V_D = 6V, R_L = 30\Omega$	-	-	8	mA
Transfer	Isolation resistance		R ISO	DC500V, 40 to 60 % RH	1010	-	-	Ω
teristics	Turn-on \$102\$01 \$202\$01			A.C. 5011	-	-	1	ms
tensues	time	time \$102\$02 \$202\$02		AC 50Hz	-	-	10	ms
	Turn-off time		t <sub>off</sub>	-	-	-	10	ms
Thermal resistance (Between junction and case)			$R_{th(j\ -\ c)}$	-	-	4.5	-	°C/W
Thermal resistance (Between junction and ambience)			$R_{th(j-a)}$		-	40	-	°C/W

#### Fig. 1 RMS ON-state Current vs. Ambient Temperature



- (1) With infinite heat sink
  (2) With heat sink (200 x 200 x 2 mm Al plate)
  (3) With heat sink (100 x 100 x 2 mm Al plate)
  (4) With heat sink (75 x 75 x 2 mm Al plate)
  (5) With heat sink (50 x 50 x 2 mm Al plate)
  (6) With heat sink

- (5) With heat sink (50 x 50 x 2 mm AI plate)
  (6) Without heat sink
  (Note) With the AI heat sink set up vertically,tighten the device at the center of the AI heat sink with a torque of 0.4N m and apply thermal conductive silicone grease on the heat sink mounting plate. Forcible cooling shall not be carried out carried out.

### Appendix 'LTS547AR'

7-segment-LED-display, ESW-16



Figuur 35 Interne aansluitingen van het display

### 14.2 mm (0.56 inch) **Seven Segment Displays**

### **Technical Data**

HDSP-530X Series HDSP-532X Series HDSP-550X Series HDSP-552X Series **HDSP-560X Series** HDSP-562X Series HDSP-570X Series HDSP-572X Series **HDSP-H15X Series** 

#### Features

- Industry Standard Size
- Industry Standard Pinout 15.24 mm (0.6 in.) DIP Leads on 2.54 mm (0.1 in.) Centers
- Choice of Colors Red, AlGaAs Red, High Efficiency Red, Yellow, Green
- Excellent Appearance Evenly Lighted Segments Mitered Corners on Segments Gray Package Gives Optimum Contrast
- $\pm\,50^\circ\,\mathrm{Viewing}$  Angle • Design Flexibility Common Anode or Common Cathode Single and Dual Digits Right Hand Decimal Point  $\pm$  1. Overflow Character
- Categorized for Luminous Intensity Yellow and Green Categorized for Color Use of Like Categories Yields a
- Uniform Display
- High Light Output • High Peak Current
- Excellent for Long Digit
- String Multiplexing • Intensity and Color Selection Option See Intensity and Color
- Selected Displays Data Sheet • Sunlight Viewable AlGaAs

#### Description

The 14.2 mm (0.56 inch) LED seven segment displays are designed for viewing distances up to 7 metres (23 feet). These devices use an industry standard size package and pinout. Both the numeric and  $\pm 1$  overflow devices feature a right hand decimal point. All devices are available as either common anode or common

cathode.

Devices

Red HDSP-	AlGaAs Red HDSP-[1]	HER HDSP-[1]	Yellow HDSP-	Green HDSP-	Description	Package Drawing
5301	H151	5501	5701	5601	Common Anode Right Hand Decimal	А
5303	H153	5503	5703	5603	Common Cathode Right Hand Decimal	В
5307	H157	5507	5707	5607	Common Anode ± 1. Overflow	С
5308	H158	5508	5708	5608	Common Cathode $\pm 1$ . Overflow	D
5321		5521	5721	5621	Two Digit Common Anode Right Hand Decimal	Е
5323		5523	5723	5623	Two Digit Common Cathode Right Hand Decimal	F

Note: 1. These displays are recommended for high ambient light operation. Please refer to the HDSP-H10X/K12X AlGaAs and HDSP-555X HER data sheet for low current operation



Figuur	36	Aangezicht	van	het	displ	ay
<b>—</b> • • •						· · •/

	FUNCTION								
PIN	А	В	С	D	E	F			
1	CATHODE e	ANODE e	CATHODE c	ANODE c	E CATHODE NO. 1	E ANODE NO. 1			
2	CATHODE d	ANODE d	ANODE c, d	CATHODE c, d	D CATHODE NO. 1	D ANODE NO. 1			
3	ANODE <sup>[3]</sup>	CATHODE <sup>[4]</sup>	CATHODE b	ANODE b	C CATHODE NO. 1	C ANODE NO. 1			
4	CATHODE c	ANODE c	ANODE a, b, DP	CATHODE a, b, DP	DP CATHODE NO. 1	DP ANODE NO. 1			
5	CATHODE DP	ANODE DP	CATHOPDE DP	ANODE DE	E CATHODE NO. 1	E ANODE NO. 2			
6	CATHODE b	ANODE b	CATHODE a	ANODE a	D CATHODE NO. 2	D ANODE NO. 2			
7	CATHODE a	ANODE a	ANODE a, b, DP	CATHODE a, b, DP	G CATHODE NO. 2	G ANODE NO. 2			
8	ANODE <sup>[3]</sup>	CATHODE <sup>[4]</sup>	ANODE c, d	CATHODE c, d	C CATHODE NO. 2	C ANODE NO. 2			
9	CATHODE f	ANODE f	CATHODE d	ANODE d	DP CATHODE NO. 2	DP ANODE NO. 2			
10	CATHODE g	ANODE g	NO PIN	NO PIN	B CATHODE NO. 2	B ANODE NO. 2			
11					A CATHODE NO. 2	A ANODE NO. 2			
12					F CATHODE NO. 2	F ANODE NO. 2			
13					DIGIT NO. 2 ANODE	DIGIT NO. 2 CATHODE			
14					DIGIT NO. 1 ANODE	DIGIT NO. 1 CATHODE			
15					B CATHODE NO. 1	B ANODE NO. 1			
16					A CATHODE NO. 1	A ANODE NO. 1			
17					G CATHODE NO. 1	G ANODE NO. 1			
18					F CATHODE NO. 1	F ANODE NO. 1			

NOTES:

1. ALL DIMENSIONS IN MILLIMETRES (INCHES).

2. ALL UNTOLERANCED DIMENSIONS ARE FOR REFERENCE ONLY.

3. REDUNDANT ANODES.

4. REDUNDANT CATHODES.

5. FOR HDSP-5600/-5700 SERIES PRODUCT ONLY.

Figuur 37 Aansluitingen van het display



Appendix 'Schema van EHW16-D'

Figuur 38 Schema van EHW18-D

Appendix 'I<sup>2</sup>C-print'



Figuur 39 schema van de I2C-print



Figuur 40 Layout van de I2C-print

### Appendix 'Bouwtips'

Als er jumpers voorgeschreven zijn, plaats dan bij het testen van de schakeling alleen de voorgeschreven jumpers! De overige moeten er uit. Als je dipswitches gebruikt: zet dan de overige op 'OFF'.

Plaats een merkteken op het AT51-printje bij de plaats waar de U-vormige uitsparing van de chip hoort. Het voetje moet ook in die richting wijzen. Er bestaat dan weinig kans op verkeerd plaatsen. Daar kan de AT89Cx051 nu eenmaal niet tegen.

De ZIF- (Zero Insertion Force-) voet van Textool op de FLAP is geplaatst op twee gestapelde standaard DI(L)Pvoetjes. Dat is nodig in verband met de hoogte. Het stapelen gaat soms wat lastig. Eenmaal geplaatst kun je ze beter zo laten zitten. De ZIF-voet is vrij prijzig maar hij spaart de controllerpootjes.

Bij het overplaatsen van de controller van de ZIF-voet naar het voetje op de AT51 moet je altijd de pootjes controleren en zonodig rechtzetten! En bij het plaatsen de chip losjes opzetten, controleren en met gevoel in de voet duwen.

#### De voeding:

Voor de AF en de FLAP: 15V 400 mA, eenvoudig gestabiliseerd. Een FAT-15V-printje is leverbaar

Het AF-bord kan samen met FLAP en voeding op een kunststof plaat gemonteerd worden, model BaBoPlex.

De resetknop voor het AF-bord heeft een plaatsje gevonden op de print van de voeding.

Het AF-bord kan van een NiCad -backupbatterij voorzien worden. Een NiCad-lader is voorzien op het AF-bord. De nominale accuspanning moet tenminste 3 Volt bedragen en maximaal 5 Volt. Het F+-bord is standaard voorzien van een Lithium batterij. Daarmee hou je de Crosscompiler-software in RAM wekenlang vast en hoef je de crosscompiler niet elke keer in het AF-bord te laden.

Bij gebruik van gewone penlights is er een extra diode nodig om te voorkomen dat ze vanaf het AF-bord voedingsspanning krijgen. Een beetje voorzichtig daarmee dus!

De FET BUK-100-50 in project ESW-08 is relatief duur, maar heeft veel meer mogelijkheden dan hier benut worden, bijvoorbeeld thermische-, kortsluit- en overspanningsbeveilingen. Je kunt dergelijke componenten ook in een passend voetje stoppen. Het kan nuttig zijn om van transistoren die direct op het bordje worden gesoldeerd de pootjes lang te houden. Dan zijn ze later, na het uitsolderen, nog bruikbaar.

Bij het hoofdstuk over RS232 is gekozen voor een baudrate van 2400, maar de software geldt alleen voor een 6-MHz-kristal

### Achterflap / Colophon

Het 'Egelproject is ontstaan naar aanleiding van vragen en ruime belangstelling op afgelopen HCCdagen waar de Forth-gg een aantal van de ontwerpen van Willem Ouwerkerk, die ook in het werkboek voorkomen, op de balie had liggen. Daarbij putte de werkgroep inspiratie uit de 'Basic Stamp' van de Amerikaanse fabrikant Parallax. Zie de publikaties daarover in Elektuur. Natuurlijk geven we bij het programmeren de voorkeur aan Forth boven Basic. Stamp vertalen we met 'postzegel'. De benaming ontwikkelde zich van Postzegel via (Postz)egel naar 'Egel. Het teken ' (tick) heeft in de programmeertaal Forth de betekenis van "adres". Overigens zeggen wij gewoon "Egel".

De projecten zijn door de werkgroep ontworpen, gebouwd en getest. De werkwijze van de groep was erop gericht goed geteste en gedocumenteerde toepassingen samen te stellen. Daarom was er:

- Maandelijks een bijeenkomst
- Eerst een ontwerp, dat vervolgens werd gebouwd en er werd software geschreven
- Daarna een beschrijving gemaakt
- Het eerste commentaar verwerkt
- Door een daarvoor aangestelde tester is de verkrijgbaarheid van onderdelen, de bouwprocedure, de software, de documentatie en wat al niet meer uitgeprobeerd.
- Diens commentaar verwerkt en zonodig het project opnieuw getest, enzovoorts

Samenstelling van de oorspronkelijke Egel-werkgroep:

Willem Ouwerkerk	Arnhem
Bas Boetekees	Maarssen
Paul Wiegmans	Alkmaar
Dick Willemse	Amsterdam
At van Wijk	Hoorn
Leendert van den Heuvel	Velp

Taken van de werkgroepleden bij de eerste druk:

Willem leverde de projecten, maakte de eerste opstelling en bleef de software aanpassen en documenteren. Bas testte zowel de hardware als de software, voelde de teksten aan de tand, ontwierp ook nog printplaten en etste die. Paul ontwierp de figuren, verzorgde de opmaak en maakte het werkboek produktierijp. Leendert verzorgde de teksten en allen droegen zorg voor de broodnodige kritische begeleiding.

Bij de tweede druk was Leendert helaas niet meer beschikbaar. Zijn taak als redacteur is overgenomen door Ben Koehorst. In deze druk is de feedback van de gebruikers van het eerste uur zo goed mogelijk verwerkt. Verder hebben de werkgroepleden aan de hand van eigen ervaringen en overleg, zowel onderling als met gebruikers, een aantal wijzigingen voorgesteld.

Derde druk is herzien en bijgewerkt.

De werkgroep gaat door en zal voor een tweede werkboek een aantal meer samengestelde projecten uitwerken, bijvoorbeeld een robotarm, een PIR/sensor, satellietschotelbesturing. Als U ideeën heeft voor interessante onderwerpen dan kunt U die ook aan ons kwijt:

Willem Ouwerkerk e-mail: voorz@forth-gg.hobby.nl Boulevard Heuvelink 126 6828 KW Arnhem telefoon: 026 443 1305

HCC Forth g.g. op het internet http://www.forth.hccnet.nl/

3<sup>e</sup> druk najaar 2004 © 1998-2004 HCC Forth gebruikersgroep.