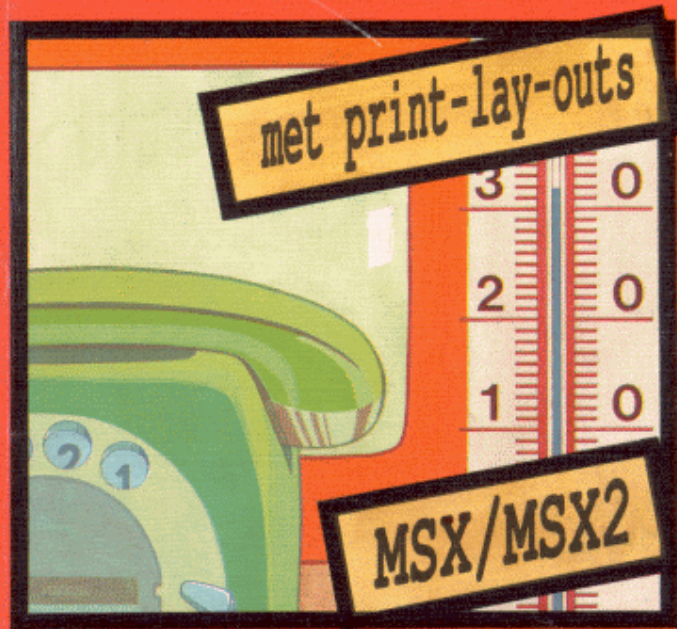


MSX computers en de buitenwereld



G. van der Sel

Scanned and converted to PDF by HansO, 2003

MSX computers en de buitenwereld

G. van der Sel

 **uitgeverij STARK-TEXEL b.v.**
postbus 302 1794 ZG Oosterend tel. 02223 661

Inhoud

1. Eenvoudig beginnen	7
1.1 Inleiding	7
1.2 De PDL interface	7
1.3 Teken en met twee PDL's	10
2. IO poort met de 8255	13
2.1 De IO poort	13
2.2 Bespreking schema	15
2.3 De bouw	16
2.4 Het testen	18
2.5 Uitbreiding IO poorten	20
3. Het 8255 IC	22
3.1 De buitenkant	22
3.2 De stuurlogika	23
3.3 Mode 0	24
3.4 Mode 1	25
3.5 Mode 2	25
3.6 Kanaalsturing poort C	25
4. Parallel output	27
4.1 Aansturing van de poort	27
4.2 Looplicht van 220 V lampen	28
4.3 Telefoon-interface	31
5. Parallel input	38
5.1 Inlezen van de poort	38
5.2 Lichtsluisteller	40
5.3 De schakeling	40
5.4 De bouw en test	41
5.5 Het programma	42

6.	Analoge uitgang (D/A omzetter)	43
6.1	Principe analoge uitgang	43
6.2	Schema voor analoge uitgang	43
6.3	De bouw van de print	45
6.4	De afregeling van de analoge uitgang	46
6.5	Modeltrein besturing	48
7.	Analoge ingang (A/D omzetter)	50
7.1	Het principe van de analoge ingang	50
7.2	De analoog/digitaalozetter	51
7.3	Temperatuurmeting	53
8.	EPRoMs programmeren	59
8.1	Het EPROM	59
8.2	Het programmeerprogramma	60
8.3	De programmeerschakeling	68
8.4	Uitbreiding aan de computer	71
9.	Extra voedingen	72
9.1	Algemeen	72
9.2	Positieve voeding	72
9.3	Negatieve voeding	74
	Verklarende woordenlijst	77
	Print-lay-outs	81

1. Eenvoudig beginnen

1.1 Inleiding

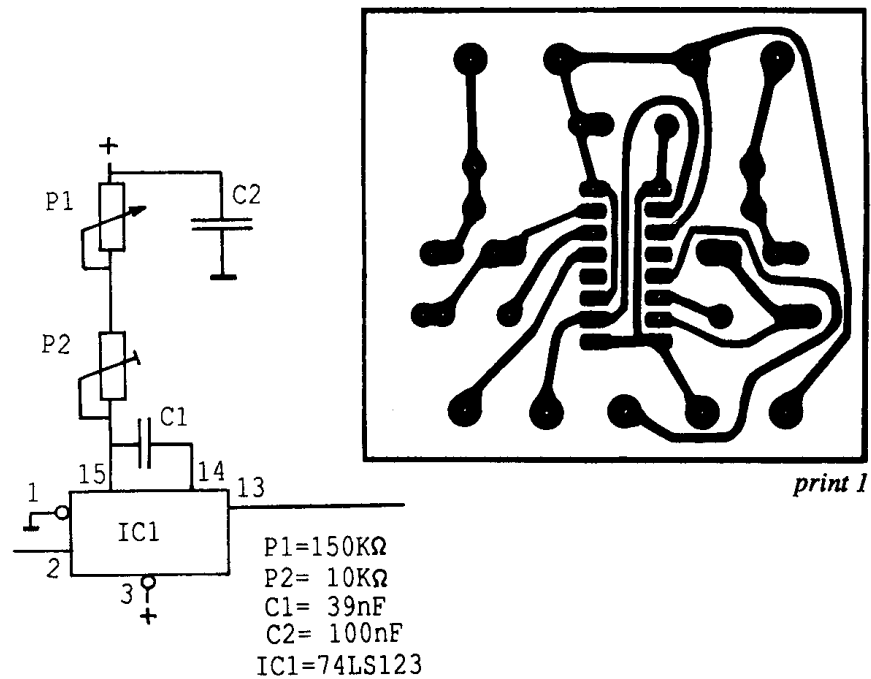
Met de schakelingen uit dit boekje kan de MSX computer reageren op signalen die niet van het toetsenbord of joystick komen, en tevens signalen naar de buitenwereld sturen anders dan de tv of monitor. De schakelingen zijn, op een enkele na, allemaal op print gezet, waarbij de printtekeningen achter in het boek te vinden zijn. De elektronikahandelaar, waar de onderdelen gekocht worden, zal graag uitleggen hoe hier printen van gemaakt kunnen worden.

Naast deze printen en de onderdelen is een goede multimeter (voor het meten van spanningen) en klein gereedschap nodig. Het kleine gereedschap bestaat onder andere uit een kniptang en een soldeerbout van maximaal 25 Watt. Met de schakelingen kan nog verder geëxperimenteerd worden. In dit hoofdstuk beginnen we met een voorbeeld dat op de joystick ingang kan worden aangesloten. Er kunnen nu op eenvoudige wijze tekeningen worden gemaakt.

1.2 De PDL interface

Met het Basic kommando: "A=PDL(0)" kan één van de twaalf paddles ingelezen worden. Met deze instructie krijgt A een waarde tussen 0 en 255. Hiermee kan net als met de joystick een punt over het beeldscherm bewogen worden. Op deze manier kunnen ook tekeningen gemaakt worden. De interface bestaat uit een IC dat twee PDL-ingangen kan besturen. Het IC is een 74LS123 die uit twee gelijke schakelingen bestaat. Voor de werking zijn alleen nog een weerstand en een condensator nodig. Door de weerstandswaarde te variëren zal het getal variëren van 0 tot 255.

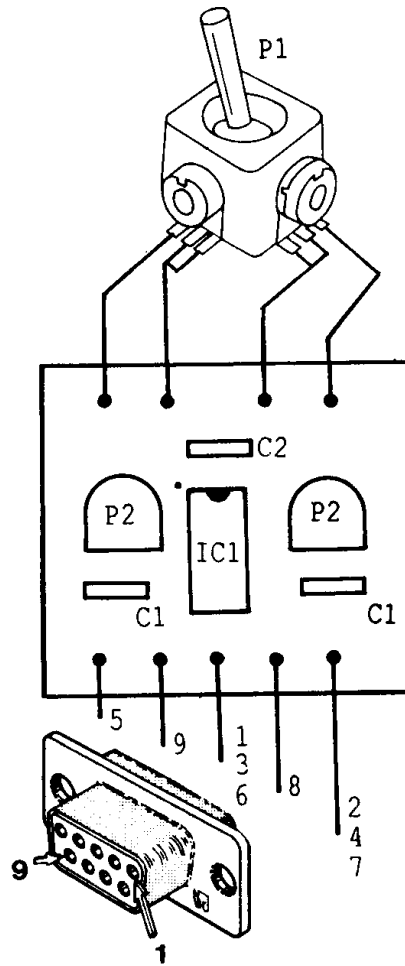
De bouw en afregeling



Als eerste wordt de print gecontroleerd op kortsluitingen en onderbrekingen met een weerstands meting. Als de print in orde is solderen we de onderdelen op de print. We beginnen met negen IP2 pennen. Hierna het IC-voetje en de condensatoren. Als laatste solderen we de twee instelbare weerstanden.

De print wordt op de hieronder aangegeven wijze aangesloten op de computer, op joystickpoort 1. Nu wordt het volgende testprogramma geRUND.

```
10 PRINT PDL(1):GOTO 10
```



We brengen de PDL naar de maximale stand en regelen met de instelbare weerstand de waarde af op 255. Bij het naar de minimale stand toe brengen zal de uitlezing naar 0 gaan. Door nu de regel te veranderen in

```
10 PRINT PDL(3):GOTO 10
```

kan de andere PDL afgeregeld worden. De PDL's zijn nu gereed voor gebruik.

1.3 Tekenen met twee PDL's

Met het onderstaande Basic programma kan op eenvoudige wijze met twee PDL's een tekening gemaakt worden. Voor het tekenen worden twee PDL's gebruikt.

```
10 REM *****
20 REM *           TEKENEN           *
30 REM *           -----           *
40 REM * (C)1987 UITGEVERIJ STARK-TEXEL BV *
50 REM *****
60 REM
70 REM *****
80 REM *           TEKEN SCHERM       *
90 REM *****
100 CLEAR200, (PEEK (&HFC4A)+256*(PEEK (&HFC4B)) -16396)
110 A=PEEK (&HFC4A)+256*PEEK (&HFC4B)
120 DEFUSR=A+16384
130 SCREEN 2
140 COLOR 15,4,4
150 OPEN "GRP:" AS 1
160 LINE (10,10)-(240,180),,B
170 REM *****
180 REM *     KEUZE AFDRUKKEN         *
190 REM *     EN TESTEN              *
200 REM *****
210 FOR I=0 TO 7
220 LINE (0,184+I)-STEP(255,0),4
230 NEXT I
240 PRESET (10,184)
250 PRINT #1,"keuze:Teken,Gum,Save,Load "
260 K$=INKEY$
270 IF K$="T" OR K$="t" THEN F=1 :GOSUB 510
280 IF K$="G" OR K$="g" THEN F=0 :GOSUB 510
290 IF K$="S" OR K$="s" THEN GOTO 590
300 IF K$="L" OR K$="l" THEN GOTO 730
310 GOTO 260
320 REM *****
330 REM *     NAAM TEKENING           *
340 REM *     OPVRAGEN               *
350 REM *****
```



```

360 N$=""
370 FOR I=0 TO 7
380 LINE (0,184+I)-STEP(255,0),4
390 NEXT I
400 PRESET (15,185)
410 PRINT #1,"NAAM :";N$;
420 I$=INKEY$:IF I$="" THEN 420
430 IF LEN(N$)>6 THEN RETURN
440 IF ASC(I$)=8 THEN IF LEN(N$)>0 THEN 470 ELSE 400
450 IF ASC(I$)=13 THEN RETURN
460 N$=N$+I$ :GOTO 400
470 N$=LEFT$(N$,LEN(N$)-1):GOTO 370
480 REM *****
490 REM * PRINT OP PDL KOORDINATEN *
500 REM *****
510 X=PDL(1)*230/255
520 Y=PDL(3)*170/255
530 IF F=1 THEN PSET(X,Y) ELSE PRESET(X,Y)
540 IF INKEY$<>"" THEN RETURN
550 GOTO 510
560 REM *****
570 REM * SAVE TEKENING *
580 REM *****
590 GOSUB 360
600 RESTORE 690
610 FOR I=A+16384 TO A+16396
620 READ D: POKE I,D
630 NEXT I
640 POKE A+16385,PEEK(&HFC4A)
650 POKE A+16386,PEEK(&HFC4B)
660 X=USR(0)
670 BSAVE N$,A,A+16384
680 GOTO 210
690 DATA 17,0,0,33,0,0,1,0,64,195,89,0
700 REM *****
710 REM * LOAD TEKENING *
720 REM *****
730 GOSUB 360
740 BLOAD N$
750 RESTORE 830
760 FOR I=A+16384 TO A+16396

```

```
770 READ D: POKE I,D
780 NEXT I
790 POKE A+16385,PEEK(&HFC4A)
800 POKE A+16386,PEEK(&HFC4B)
810 X=USR(0)
820 GOTO 210
830 DATA 33,0,0,17,0,0,1,0,64,195,92,0
```

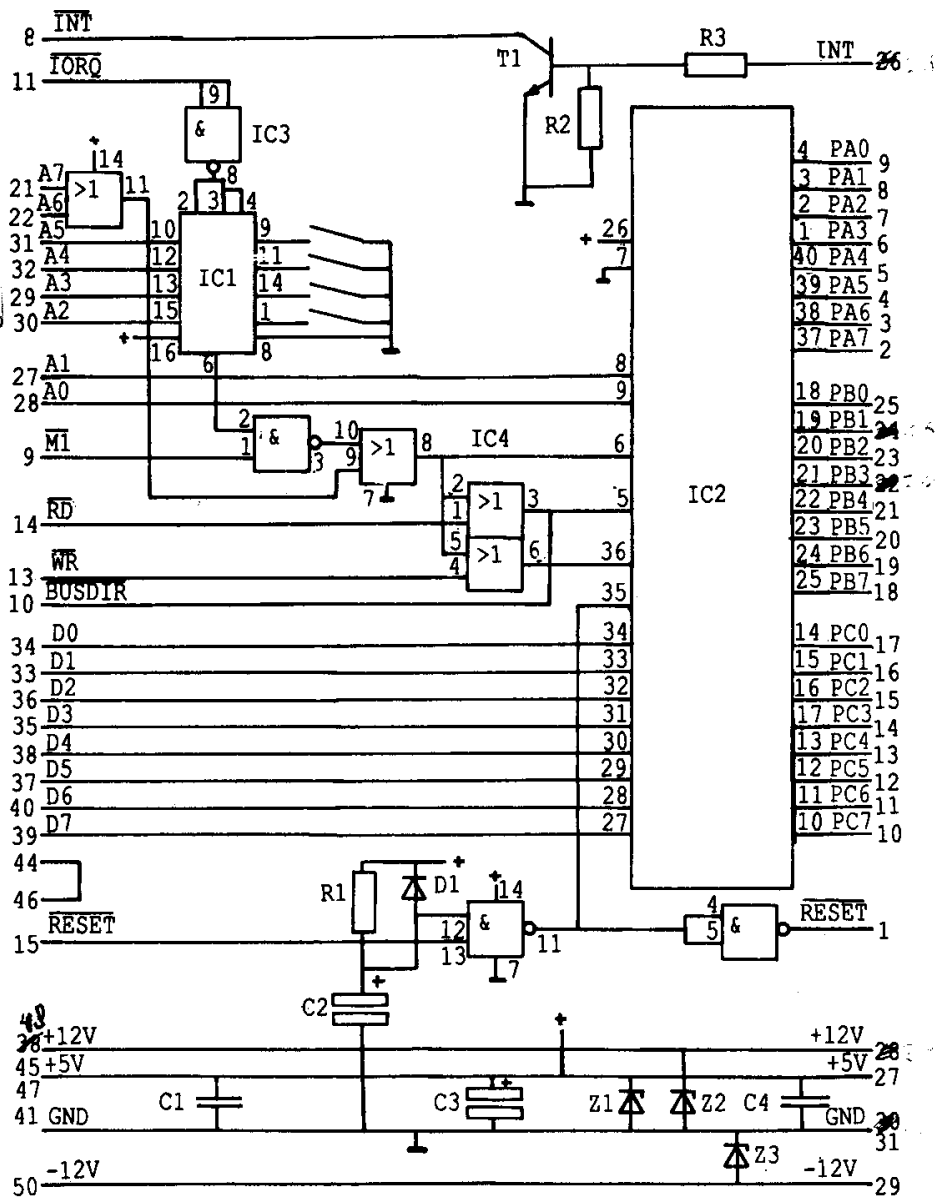
2. IO poort met de 8255

2.1 De IO poort

Naast de normale input en output mogelijkheden heeft de MSX computer ook nog zijn twee slotkonnektoren. In deze konnektoren ('slots' genaamd) kunnen naast de bekende ROM spelletjes of disk-drive ook een geheugenuitbreiding of de hier besproken IO poort gestoken worden. Deze IO poort heeft 24 lijnen die naar keuze als inputpoort (signalen van buiten naar de computer) of outputpoort (signalen van de computer naar de buitenwereld) kunnen zijn geschakeld.

Het besturen van de IO poort gebeurt vanuit Basic met de "Out" en "Inp" instructie, zodat de besturing eenvoudig is. Alle IO functies worden door een IC gedaan. Dit is de 8255 van Intel. Vele andere fabrikanten maken onder hun eigen naam ook dit IC, die alle gelijk zijn aan die van Intel. Dit IC bevat naast alle logika om met de computer te communiceren, tevens drie 8 bits IO poorten. In de output mode zullen de poorten de naar buiten gestuurde data bufferen, zodat de computer met het programma door kan gaan. In de input mode zal de data uit de poort als de computer er om vraagt op de databus gezet worden.

Door andere kommando's naar het IC te sturen kan gekozen worden voor gebufferde input of output en input of output onder interrupt sturing (wordt in dit boek niet toegepast). De IO poort is behalve op de databus ook op de adresbus aangesloten. Iedere poort in het IC heeft zijn eigen adres. Het adres van de poorten kan door middel van vier draadbruggen of *dip*-schakelaars ingesteld worden en loopt van &H00 tot en met &H3F. De adressen &H40 tot en met &HFF worden door de MSX computer voor onder andere het beeldscherm en het toetsenbord zelf gebruikt.



- | | |
|-----------|--------------|
| R1= 10KΩ | T1=BC547 |
| R2= 10KΩ | Z1=Zener 5V |
| R3= 47KΩ | Z2=Zener 12V |
| C1=100nF | Z3=Zener 12V |
| C2= 1μF | IC1=74LS85 |
| C3=100μF | IC2=8255 |
| C4=100nF | IC3=74LS00 |
| D1=1N4148 | IC4=74LS32 |

De IO poort bezet zelf vier adressen, zodat het adres van de IO poort vier keer het ingestelde getal is (bijvoorbeeld 2 ingesteld, dan is het adres &H08). Op de volgende drie adressen bevindt zich een achtkanaals brede data poort: in het voorbeeld op &H08, &H09 en &H0A, terwijl op het vierde adres (in het voorbeeld &H0B) een stuurregister aanwezig is, waarmee het IC bestuurd wordt. Dit moet gebeuren voordat het IC door de computer gebruikt wordt voor IO handelingen.

2.2 *Bespreking schema*

In het schema is links de MSX slotkonnektor getekend. Om de IO poort te kunnen gebruiken wordt dit met de computer verbonden. Aan de rechter zijde zit de 8255 met de IO konnektor. De IO konnektor is tevens voorzien van alle voedingsspanningen en een gebufferde Reset uitgang. Links boven is rond de 74LS85 de adresdekodering gemaakt. Tijdens een IO actie gaat de lijn IORQ (11) naar "0", zodat de 74LS85 begint met het vergelijken van A2 tot en met A5 met het ingestelde getal. Als deze gelijk zijn zal pen 6 van de 74LS85 naar "1" gaan. Als nu tevens de pen M1 (9) "1" is (er vindt geen interrupt bevestiging plaats), dan zal pen 3 van de Nand naar "0" gaan.

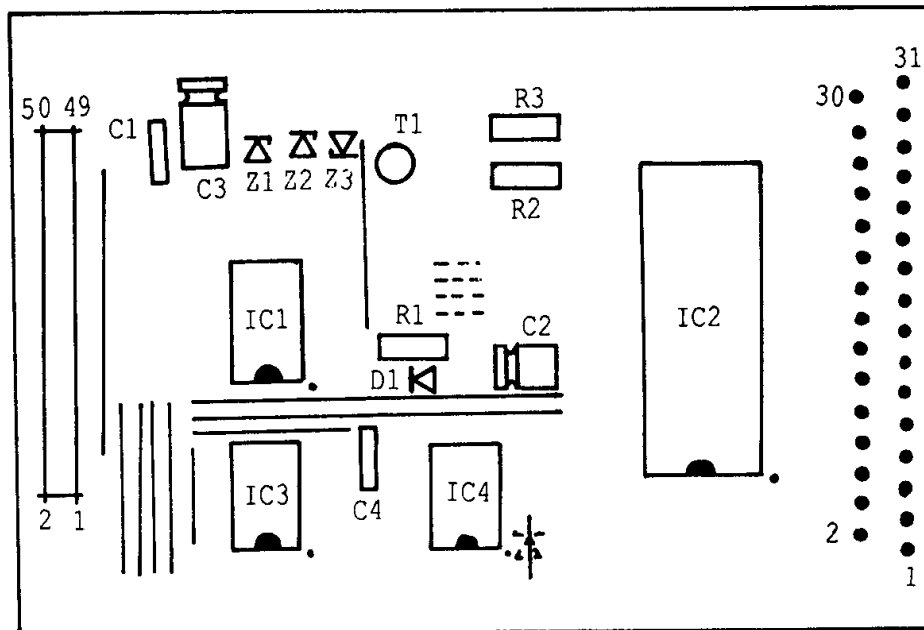
Volgens de MSX norm moeten de lijnen A6 (22) en A7 (21) ook "0" zijn, zodat uitgang 11 en dus ook ingang 9 "0" zijn. Hierdoor zal aan beide ingangen van de OF poort een "0" aangeboden worden, waardoor de uitgang hiervan ook naar "0" gaat. Hierdoor zal de CS (*chipselect*) ingang van de 8255 naar "0" gaan, waardoor de 8255 weet, dat afhankelijk van de toestand van de adres ingangen (A0 en A1) en de RD en WR ingang er een actie op de databus moet plaatsvinden.

Gaat RD naar "0" dan zal de 8255 de acht kanalen uit de IO poort (aangegeven met A0 en A1) op de databus plaatsen. Gaat de WR lijn naar "0" dan zal de 8255 de acht bits van de data bus in de IO poort of het stuurregister (aangegeven door A0 en A1) plaatsen. Omdat de databus buffer in de MSX computer alleen reageert als er data weggeschreven wordt, moet de databusbuffer tijdens leesacties door de IO poort bestuurd worden. Dit gebeurt met de lijn Busdir (10).

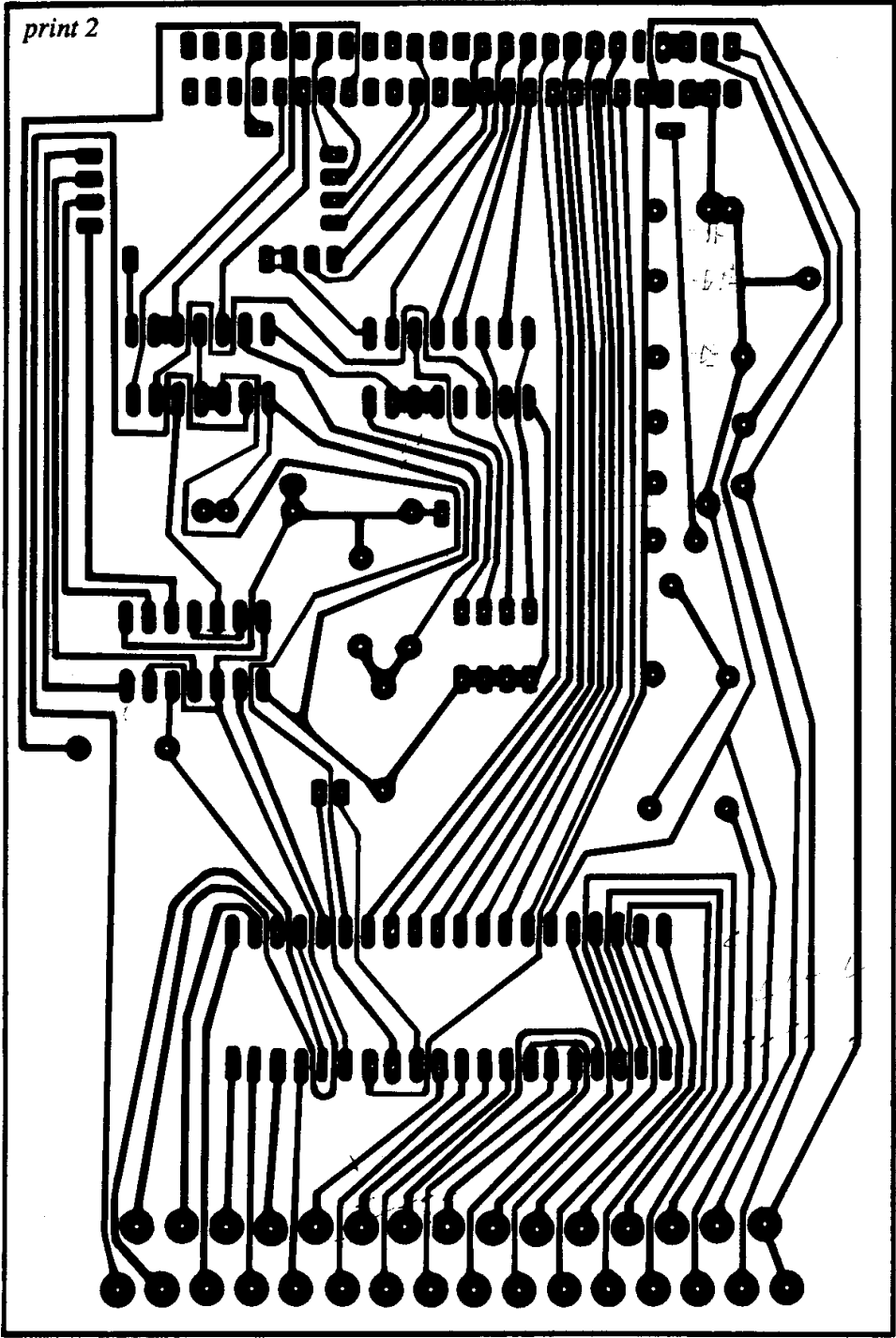
Door deze lijn tijdens het lezen "0" te maken zal de databusbuffer omschakelen van richting en de gegevens doorgeven aan de computer. In dit stuursignaal kan een germaniumdiode opgenomen worden om meerdere IO poorten op de computer aan te sluiten. De voedingslijnen zijn door middel van zenerdioden beveiligd tegen te hoge voedingsspanningen, en om stoorsignalen op de 5 V voeding te voorkomen zijn hier enkele condensatoren aangebracht.

Bij het aanzetten of resetten van de computer zal op de Reset lijn (15) een "0" komen. Deze nul komt via twee Nand poorten tevens op de IO konnektor. De 8255 wordt nu ook geReset, waardoor alle IO poorten als ingang worden geschakeld. Hierdoor wordt beschadiging van het IC voorkomen tijdens het inschakelen van de computer. Als laatste zijn in het schema nog twee weerstanden en een transistor opgenomen, waarmee interrupts ter verwerking aan de computer kunnen worden gegeven.

2.3 De bouw



Na de print goed gecontroleerd te hebben op onderbrekingen of kortsluitingen kan begonnen worden met de bouw van de IO poort.

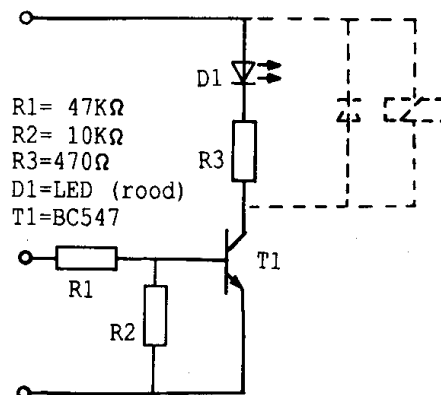


Eerst solderen we de elf draadbruggen. Dan maken we een 60 centimeter lange 50-aderige "flatcable" aan de IO print, met aan de andere zijde een dubbelpolige konnektor print. Dan solderen we de 31 aansluitpennen (dit zijn IP2 pennen).

Nu, na alle draad-aansluitingen, kunnen we met de componenten beginnen. Eerst de vier IC voeten. In de voetjes zitten nokjes voor de pen 1 aansluiting. Let goed op de plaatsing van de componenten voor deze pen 1! Soldeer ze er zodanig op, dat de IC's er goed in gedrukt kunnen worden. Hierna solderen we de transistor en tot slot de weerstanden en condensatoren. Als dit gedaan is, blijven er acht gaten open, deze zijn om het adres van de print in te stellen. In het boek zal in alle programma's het in het voorbeeld aangenomen adressenbereik worden aangehouden, dus het getal 2.

We solderen nu nog de drie extra draadbruggen voor het adres. De print kan nu eventueel in een kastje worden gebouwd.

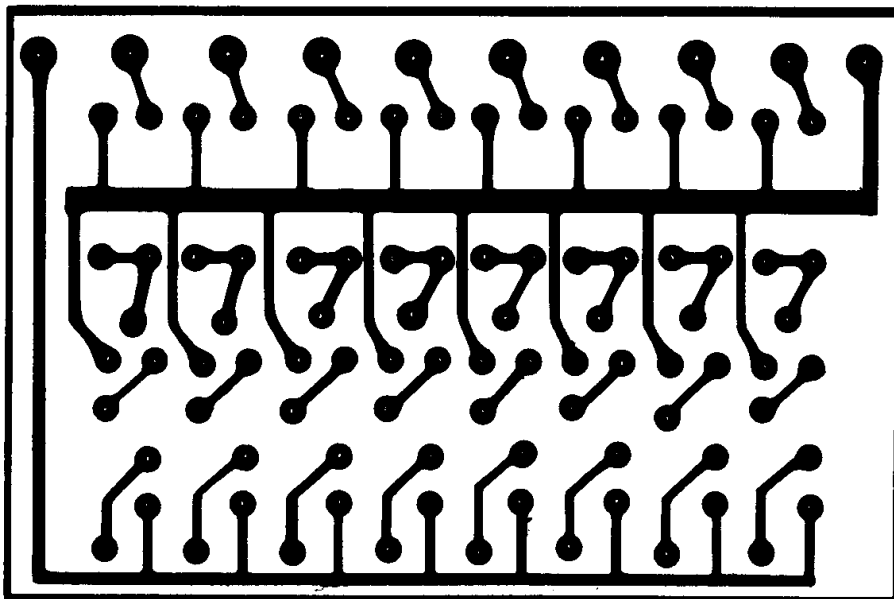
2.4 Het testen



Nadat de print op de computer is aangesloten, kunnen we hem testen. Dit doen we met behulp van een print met hierop acht LEDs. We testen poort voor poort met het test programma. In alle drie de gevallen moet een looplicht gaan draaien.

De print bestaat uit acht gelijke schakelingen die de LEDs doen oplichten. De print wordt gecontroleerd en van componenten voor-

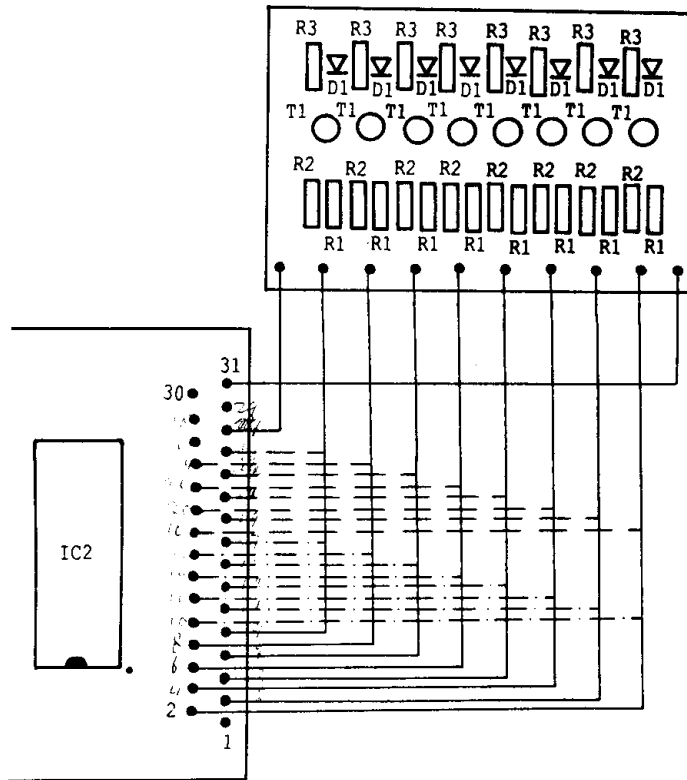
zien, en dan volgens tekening aangesloten op de IO print. We RUN-
nen nu het onderstaande programma en kijken of er een looplicht
draait. Als dit zo is, dan testen we de volgende poort. De print is
goed als in alle drie de gevallen hetzelfde patroon ontstaat.



print 3

```
10 OUT&H0C, &H80
20 FOR I=0 TO 7
30 OUT&H08, 2^I      :REM poort A
40 OUT&H09, 2^I      :REM poort B
50 OUT&H0A, 2^I      :REM poort C
60 FOR J=0 TO 1000 :NEXT J
70 NEXT I
80 GOTO 20
```

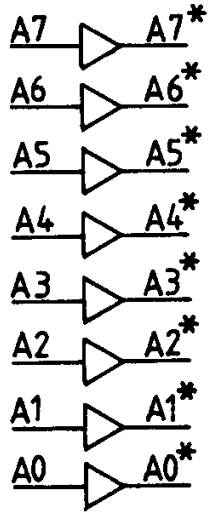
In de volgende hoofdstukken gaan we met de IO print als basis ver-
dere schakelingen aan de computer koppelen.



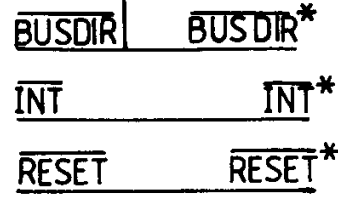
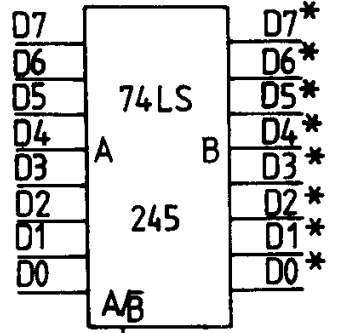
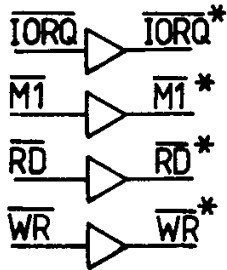
2.5 Uitbreiding IO poorten

Als er na enige tijd meerdere IO poorten nodig zijn, dus meer dan de huidige 24 lijnen, dan kan door het bouwen van meerdere printen het aantal IO poorten uitgebreid worden. Deze printen hoeven niet ieder in hun eigen slot, maar kunnen met maximaal drie in hetzelfde slot, door ze parallel te schakelen. Er kan nu echter geen gebruik meer worden gemaakt van de voedings-uitgangen, doordat de computer niet meer stroom kan leveren.

Bij uitbreiding tot het maximum van zestien IO printen, moet de MSX slotkonnektor gebufferd worden, terwijl ook nu een extra voeding noodzakelijk is. Dit gebeurt door onderstaand schema tussen de computer en de IO printen op te nemen. Tevens moet nu op alle printen de draadbrug (zie schema IO print) vervangen worden door een germaniumdiode.



2x 74LS07



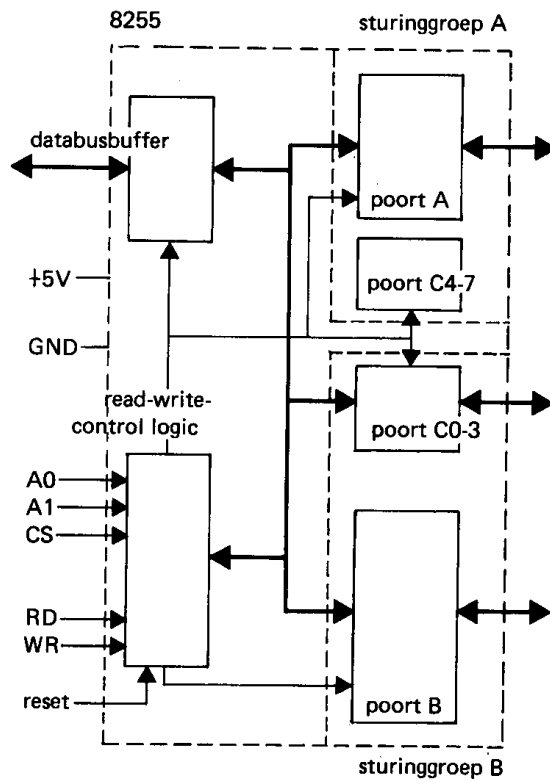
*= NAAR I/O POORT

3 + 10 + 18
 24 + 11 + 11
 10 + 11 + 11
 10 + 11 + 11
 10 + 11 + 11

3. Het 8255 IC

3.1 De buitenkant

De 8255 is een IC dat voor meerdere doeleinden gebruikt kan worden. Door stuurwoorden naar het IC te sturen zullen de IO in- en uitgangen (dit zijn de pennen op het IC) van het IC op verschillende manieren gebruikt kunnen worden.



De 8255 heeft drie verschillende I/O poorten. Deze poorten hebben de volgende namen: eerste poort A, tweede poort B en als derde poort C.

Van de poorten A en B kan de hele poort als ingang of uitgang geschakeld worden.

Poort C kan in twee helften gedeeld worden. Hierdoor ontstaat de mogelijkheid om twee groepen van vier bits te definiëren. Tevens kunnen, als deze poort als uitgang is geschakeld, de uitgangen stuk voor stuk "1" of "0" gemaakt worden met speciale kommando's.

3.2 De stuurlogika

Om met de I/O poort te kunnen werken, zullen de juiste kommando's naar de poort gestuurd moeten worden. Er wordt pas naar alle stuursignalen van de Z80 (het hart van uw MSX computer) gekeken, als de pen CS "0" is. Hierdoor weet de 8255 dat er een uitwisseling van gegevens met de Z80 gaat plaatsvinden.

Het blok *Read-Write-Control-Logic* zal afhankelijk van de toestand van de lijnen A0, A1, WR, en RD gegevens via de databusbuffer naar de blokken 1 tot en met 4 sturen, of vanuit de blokken naar de computer. De keuze welk blok de gegevens krijgt of verstuurt wordt aangegeven met de lijnen A0 en A1.

Als beide "0" zijn wordt poort A bedoeld; A0="1" en A1="0", dan poort B, A0="0" en A1="1" dan poort C en als A0 en A1 beide "1" dan wordt het stuurregister bedoeld.

Het krijgen van gegevens wordt aangegeven met de WR lijn, deze gaat dan naar "0". Het versturen van gegevens mag pas als de RD lijn naar "0" gaat. Door een Reset signaal zullen alle inwendige buffers naar "0" gaan en zal voor enkelvoudige input gekozen worden (Mode 0). Door het schrijven van een stuurwoord naar het stuurregister zal van input naar output geschakeld kunnen worden, terwijl tevens van mode gewisseld kan worden.

A1	A0	RD	WR	CS	AKTIE
0	0	0	1	0	INPUT A
0	1	0	1	0	INPUT B
1	0	0	1	0	INPUT C
1	1	0	1	0	VERBODEN
0	0	1	0	0	OUTPUT A
0	1	1	0	0	OUTPUT B
1	0	1	0	0	OUTPUT C
1	1	1	0	0	STUURWOORD
X	X	1	1	0	GEEN
X	X	X	X	1	GEEN

Er zijn drie verschillende gebruiksvormen, namelijk Mode 0, Mode 1 en Mode 2.

3.3 Mode 0

Mode 0 is de simpele input/output mode, waarbij poort A en B als achtkanaals poort gebruikt worden. Poort C wordt in deze mode in twee groepen van vier kanalen verdeeld, die naar keuze als input of output geschakeld kunnen worden.

De stuurwoorden voor deze mode zijn als volgt:

Stuurwoord	Poort A	Poort B	Poort C(0-3)	Poort C(4-7)
<i>&H9B</i>	<i>in</i>	<i>in</i>	<i>in</i>	<i>in</i>
<i>&H93</i>	<i>in</i>	<i>in</i>	<i>in</i>	<i>uit</i>
<i>&H9A</i>	<i>in</i>	<i>in</i>	<i>uit</i>	<i>in</i>
<i>&H92</i>	<i>in</i>	<i>in</i>	<i>uit</i>	<i>uit</i>
<i>&H99</i>	<i>in</i>	<i>uit</i>	<i>in</i>	<i>in</i>
<i>&H91</i>	<i>in</i>	<i>uit</i>	<i>in</i>	<i>uit</i>
<i>&H98</i>	<i>in</i>	<i>uit</i>	<i>uit</i>	<i>in</i>
<i>&H90</i>	<i>in</i>	<i>uit</i>	<i>uit</i>	<i>uit</i>
<i>&H8B</i>	<i>uit</i>	<i>in</i>	<i>in</i>	<i>in</i>
<i>&H83</i>	<i>uit</i>	<i>in</i>	<i>in</i>	<i>uit</i>
<i>&H8A</i>	<i>uit</i>	<i>in</i>	<i>uit</i>	<i>in</i>
<i>&H82</i>	<i>uit</i>	<i>in</i>	<i>uit</i>	<i>uit</i>
<i>&H89</i>	<i>uit</i>	<i>uit</i>	<i>in</i>	<i>in</i>
<i>&H81</i>	<i>uit</i>	<i>uit</i>	<i>in</i>	<i>uit</i>

<code>&H88</code>	<code>uit</code>	<code>uit</code>	<code>uit</code>	<code>in</code>
<code>&H80</code>	<code>uit</code>	<code>uit</code>	<code>uit</code>	<code>uit</code>

Om nu gegevens van en naar de poort te krijgen, hoeft alleen naar de desbetreffende poort geschreven of uitgelezen te worden.

3.4 Mode 1

Mode 1 is een input/output mode die uitsluitend met de poorten A en B werkt. Als voor deze mode wordt gekozen, worden van poort C een aantal kanalen gebruikt om kommando's aan het te besturen apparaat te geven en daarvan te ontvangen.

De 8255 zal nu interrupts generen om de computer te attenderen op het feit dat de opdracht is uitgevoerd.

Door nu poort C uit te lezen kan gekeken worden hoe ver de opdracht is uitgevoerd.

Deze mode wordt in dit boek niet gebruikt.

3.5 Mode 2

Mode 2 is een mode waarin poort A onder besturing van een extern apparaat van ingang of uitgang kan wisselen. Er worden ook weer signalen naar het te besturen apparaat gezonden en ontvangen. Hier-voor worden vijf kanalen van poort C gebruikt.

Deze mode wordt in dit boek niet gebruikt.

3.6 Kanaal sturing poort C

De kanalen van poort C kunnen in de output mode afzonderlijk ge-stuurd worden. Dit gebeurt door stuurwoorden naar het stuurregister te sturen.

Deze stuurwoorden zijn:

	Om "1" te maken	Om "0" te maken
Kanaal 1	&H01	&H00
Kanaal 2	&H03	&H02
Kanaal 3	&H05	&H04
Kanaal 4	&H07	&H06
Kanaal 5	&H09	&H08
Kanaal 6	&H0B	&H0A
Kanaal 7	&H0D	&H0C
Kanaal 8	&H0F	&H0E

Met deze woorden worden de andere kanalen niet beïnvloed.

10

4. Parallel output

4.1 Aansturing van de poort

Bij parallel aansturing kan de poort gebruikt worden om bijvoorbeeld relais of lampen aan en uit te schakelen. De poort wordt eerst in de output-mode geschakeld vanuit Basic. Hierna kan de poort vanuit Basic bestuurd worden. Ieder kanaal kan apart vanuit Basic met een kommando naar 5 V ("1") getrokken worden, of naar 0 V ("0").

Voor het naar 1 brengen van poort A:

Met de instructie

```
OUT(&H08), INP (&H08) OR 2^I
```

wordt kanaal I naar 1 getrokken en worden de andere kanalen niet veranderd. I is het kanaalnummer en loopt van 0 tot en met 7. Hiervoor moet wel eerst de oude stand met Inp(&H08) ingelezen worden. Met de logische Or-instructie wordt het kanaal op 1 gezet zonder dat de andere kanalen veranderd worden.

Voor het naar 0 brengen van poort A:

Met de instructie

```
OUT(&H08), INP (&H08) AND (255-2^I)
```

wordt kanaal I naar 0 getrokken en worden de andere kanalen niet veranderd. I is het kanaalnummer en loopt van 0 tot en met 7. Hiervoor moet wel eerst de oude stand met Inp(&H08) ingelezen worden. Met de logische And1-instructie wordt het kanaal op 0 gezet zonder dat de andere kanalen veranderd worden.

Door de getallen achter de And- of Or-funktie zelf uit te rekenen kunnen zelfs meerdere kanalen tegelijk op 1 of 0 gezet worden.

De beide voorgaande voorbeelden kunnen ook in machinetaal uitgevoerd worden.

Voor het "1" maken:

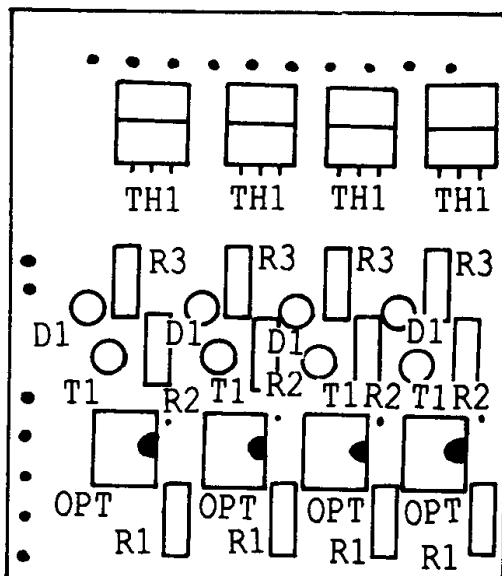
```
IN   A, (#08) ;Lees oude waarde in.  
SET  3,A      ;Maak kanaal 3 "1".  
OUT  (#08),A  ;Schrijf nieuwe waarde naar poort.
```

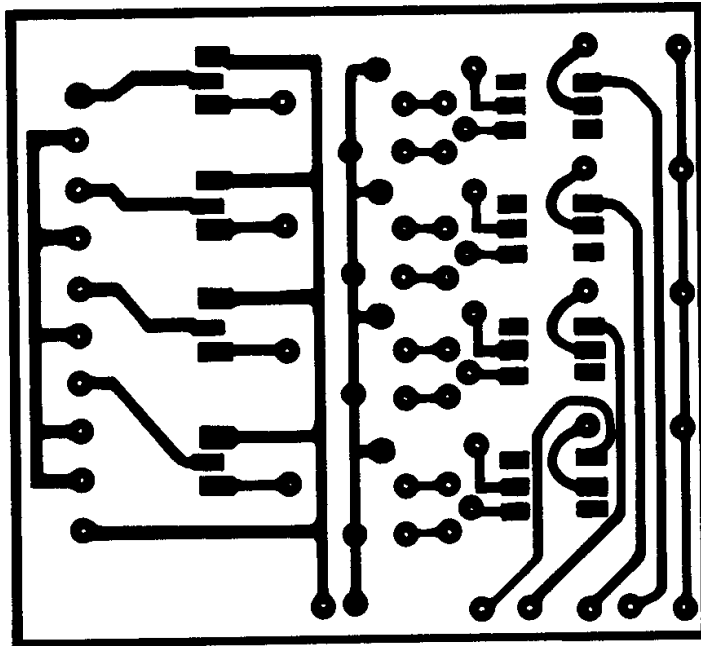
Voor het "0" maken:

```
IN   A, (#08) ;Lees oude waarde in.  
RES  5,A      ;Maak kanaal 5 "0".  
OUT  (#08),A  ;Schrijf nieuwe waarde naar poort.
```

Door de getallen 3 en 5 in de Set- of Res-instructie te veranderen kunnen andere kanalen op 0 of 1 gezet worden.

4.2 Looplicht van 220 V lampen



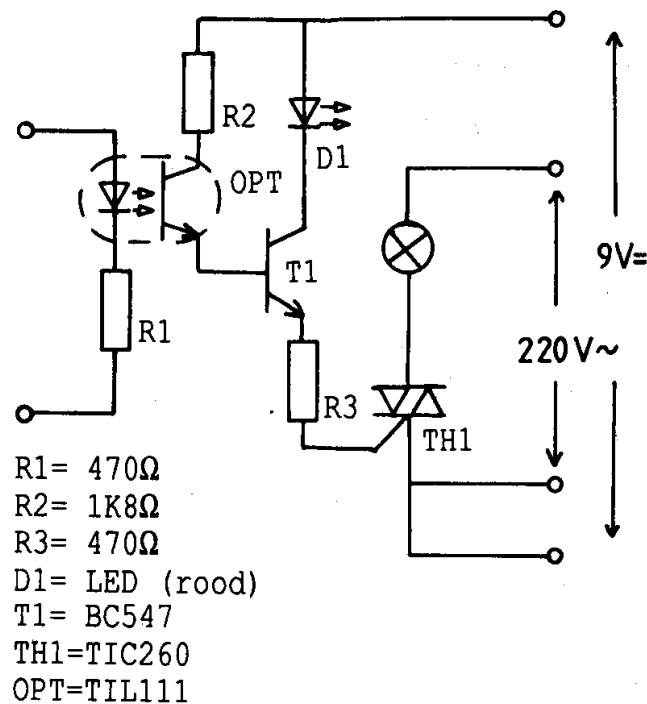


print 4

Om apparaten die uit het lichtnet gevoed worden (220 V) aan te sturen met de MSX computer moet de 220 V gescheiden worden van de computer. Dit wordt gedaan met een OPTO-coupler.

Een OPTO-coupler kan echter geen stroom groter dan enkele milliampères leveren, zodat de uitgangsstroom eerst door een transistor versterkt moeten worden. Daarna wordt deze stroom toegevoerd aan een TRIAC. Door deze TRIAC is het mogelijk om per kanaal 1000 W te schakelen (de TRIAC moet dan wel gekoeld worden).

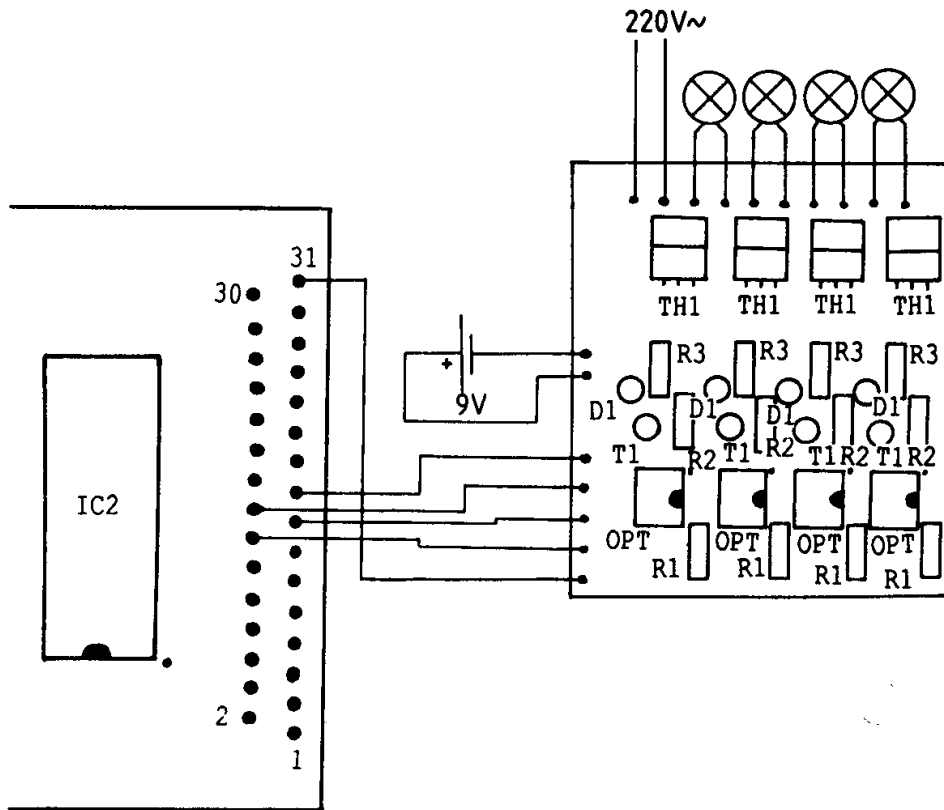
In het onderstaande schema is de OPTO-coupler links getekend. De poort zal met een 1 aan de uitgang de LED in de OPTO-coupler laten oplichten. Hierdoor zal de transistor in de OPTO-coupler gaan geleiden en via de weerstand R1 de stroom naar T1 sturen. Hierdoor gaat deze geleiden, waardoor via LED D1 en de weerstand R2 de TRIAC TH1 ontstoken wordt. Doordat TH1 ontstoken wordt kan de lamp gaan branden. Als de lamp brandt, zal ter controle tevens de LED D1 oplichten. In plaats van een lamp kan uiteraard ook bijvoorbeeld een koffiezetapparaat aan- en uitgeschakeld worden.



Op de print kunnen vier 220 V interfaces opgebouwd worden. Na de print goed gecontroleerd te hebben op onderbrekingen en kortsluitingen monteren we eerst de zeventien printpennen (IP2) waarop de aansluitingen komen. Hierna solderen we de voetjes voor de OPTO-couplers, dan de weerstanden. Als deze goed gesoldeerd zijn, dan als laatste de LEDs, transistors en TRIACs.

Tenslotte drukken we de OPTO-couplers in hun voetjes. We bedraden nu de print volgens de tekening. We sluiten een 9 V batterij op de voedingslijnen aan, maar steken de steker nog niet in het stopkontakt. Nu RUNnen we het onderstaande Basic programma en kijken of de LEDs één voor één oplichten. Als dit ook werkt kan de steker in het stopkontakt gestoken worden en het programma voor de tweede geRUNd worden.

Nu zullen de lampen en de LEDs tegelijk oplichten. Als dit ook werkt kan door veranderen van het programma de volgorde van het oplichten van de lampen veranderd worden. Er kan ook voor een volledig zelf ontworpen patroon gekozen worden.



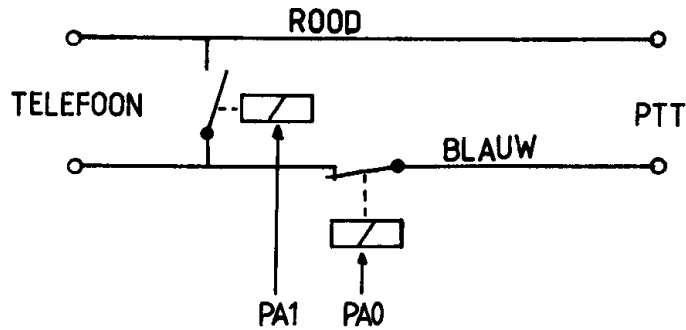
```

10 REM test 220V interface
20 OUT &HB,&H9A      :REM alleen C0-3 uitgang
30 FOR I=0 TO 3     :REM teller voor alle lampen
40 OUT(&HA),2^I     :REM nieuw patroon, linksom draaien
50 FOR J=0 TO 200:NEXTJ:REM wachtlus
60 NEXT I           :REM volgende lamp
70 GOTO 30         :REM stop bij Ctrl-Stop

```

4.3 Telefoon-interface

Met het hier beschreven programma en de testschakeling uit hoofdstuk 2 is het mogelijk om de computer een telefoonnummer te laten draaien. Het gesprek kan daarna gewoon gevoerd worden via de telefoon. Bij het buiten de stad bellen wacht de computer ongeveer een seconde om de centrale de gelegenheid te geven de kies-toon weer op de lijn te zetten.



Van de testschakeling worden twee LEDs en weerstanden vervangen door twee relais. De overige componenten mogen worden weggehaald. Het ene relais moet een maakkontakt hebben, terwijl het andere relais een verbreekkontakt moet hebben. De beide relais mogen ook wisselcontacten hebben. Parallel aan de relais worden dioden aangesloten om de transistor te beschermen. De contacten worden aangesloten zoals aangegeven in het schema. De leidingen A en B wordt in de steker uitgenomen en met de relais verbonden, waarna de nieuwe A en B aangesloten worden in de steker.

Als nu de voeding niet aan staat moet de telefoon normaal werken. De voedingsspanning is afhankelijk van het soort relais. Bij 5 V relais kunnen beide relais en de IO-print uit de computer gevoed worden. Voor andere spanningen kan een batterij gebruikt worden.

Nadat onderstaande programma is ingetikt en geSAVEd kan het voor de eerste maal geRUNd worden. Hierna verschijnt een menu op het scherm en kan uit de volgende mogelijkheden gekozen worden:

- F1 = Zelf nummer intoetsen
- F2 = Zoeken volgens een lijst
- F3 = Herhalen van laatste nummer
- F4 = Toevoegen/vervangen namen
- F5 = Laden van lijst
- F10= Saven van lijst

Met F1 wordt een nummer gevraagd, dat hierna door de computer "gedraaid" wordt.

Met F2 worden de eerste twintig namen uit de lijst getoond. Met de V toets kan naar het andere blok van twintig gegaan worden. Wordt er een nummer uit de lijst gekozen dan zal het bijbehorende telefoonnummer door de computer "gedraaid" worden.

Met F3 kan het laatst "gedraaide" nummer herhaald worden. Dit is te gebruiken als er niet opgenomen werd.

Met F4 kunnen namen aan de lijst worden toegevoegd of vervangen door andere namen.

Met F5 kan de lijst vanaf kassette of disk geladen worden.

Met F10 kan de lijst naar kassette of disk weggeschreven worden.

```
10 REM *****
20 REM *      TELEFOONNUMMER-DRAAIER      *
30 REM * ----- *
40 REM * (C)1987 UITGEVERIJ STARK-TEXEL BV *
50 REM *****
60 REM
70 REM *****
80 REM *      INITIALISTIE + MENU      *
90 REM *****
100 DIM NM$(40),NR$(40): OUT&HB,&H8B
110 ON KEY GOSUB 290,420,640,740,880,,,,,1050
120 KEY(1) ON:KEY(2) ON:KEY(3) ON:KEY(4) ON
125 KEY(5) ON:KEY(10) ON :KEY OFF
130 CLS: LOCATE 0,4
140 PRINT"  F1 = Zelf nummer intoetsen"
150 PRINT"  F2 = Zoeken volgens lijst"
160 PRINT"  F3 = Herhalen laatste nummer"
170 PRINT"  F4 = Toevoegen/vervangen nummers"
180 PRINT"  F5 = Laden van lijst"
190 PRINT"  F10= Saven van lijst"
200 LOCATE6,14: PRINT"DRUK OP FUNKTIE TOETS"
210 GOTO 210
220 REM *****
230 REM *
240 REM *      FUNKTIE 1      *
250 REM *
```

```

260 REM *   Direkt nummer kiezen   *
270 REM *
280 REM *****
290 CLS:LOCATE 0,7: NR$(0)=" "
300 PRINT "GEEF TE DRAAIEN NUMMER:( Q is stop )"
310 LOCATE6,8:LINEINPUT NR$(0)
320 IF NR$(0)="Q" OR NR$(0)="q" THEN RETURN 130
330 GOSUB 1230
340 RETURN 130
350 REM *****
360 REM *
370 REM *   FUNKTIE 2   *
380 REM *
390 REM *   Zoeken in lijst   *
400 REM *
410 REM *****
420 CLS:O=0
425 F$="##\           \           \"
430 PRINT"NR.   NAAM           TELEFOONNUMMER"
440 FOR I=1+O*20 TO 20+O*20
450 PRINTUSING F$;I,NM$(I),NR$(I)
460 NEXT I
470 PRINT" V voor volgende/vorige; Q is stop "
480 PRINT"           Wie wilt u bellen           ";
490 LOCATE23,22:INPUT I$
500 IF I$="V" OR I$="v" THEN O=1-O:GOTO 430
510 IF I$="Q" OR I$="q" THEN 560
520 IF I$<"0" OR I$>"9" THEN 490
530 I=VAL(I$)
540 NR$(0)=NR$(I)
550 GOSUB 1230
560 RETURN 130
570 REM *****
580 REM *
590 REM *   FUNKTIE 3   *
600 REM *
610 REM *   Herhalen laatste nummer *
620 REM *
630 REM *****
640 GOSUB 1230
650 RETURN 130

```



```

660 REM *****
670 REM *
680 REM *          FUNKTIE 4          *
690 REM *
700 REM *    Nummer toevoegen/      *
710 REM *          veranderen      *
720 REM *
730 REM *****
740 CLS:LOCATE5,8
750 PRINT"Welk nummer wilt u toevoegen"
755 LOCATE 6,9
756 PRINT "/veranderen ? (Q is STOP)"
760 PRINT "Nummer:";LINEINPUT I$
770 IF I$="Q" OR I$="q" THEN RETURN 130
780 LINEINPUT"Naam:";NM$(VAL(I$))
790 LINEINPUT"Telefoonnummer:";NR$(VAL(I$))
800 RETURN 130
810 REM *****
820 REM *
830 REM *          FUNKTIE 5          *
840 REM *
850 REM *    Laden van de lijst      *
860 REM *
870 REM *****
880 LOCATE0,21
890 PRINT"Zet kassetterekorder op afspelen, "
895 PRINT"en druk op een toets. (Q is STOP)"
900 I$=INKEY$:IF I$="" THEN 900
910 IF I$="Q" OR I$="q" THEN RETURN 130
920 OPEN "NAMTEL" FOR INPUT AS #1
930 FOR I=1 TO 40
940 INPUT#1,NM$(I),NR$(I)
950 NEXT I
960 CLOSE #1
970 RETURN 130
980 REM *****
990 REM *
1000 REM *          FUNKTIE 10         *
1010 REM *
1020 REM *    Saven van de lijst      *
1030 REM *

```

```

1040 REM *****
1050 LOCATE0,21
1060 PRINT"Zet kassetterekorder op opnemen,"
1065 PRINT"en druk op een toets. (Q is STOP)"
1070 I$=INKEY$:IF I$="" THEN 1070
1080 IF I$="Q" OR I$="q" THEN RETURN 130
1090 OPEN "NAMTEL" FOR OUTPUT AS #1
1100 FOR I=1 TO 40
1110 PRINT#1,NM$(I):PRINT#1,NR$(I)
1120 NEXT I
1130 CLOSE #1
1140 RETURN 130
1150 REM *****
1160 REM *   Computer gestuurde   *
1170 REM *   telefoonschijf     *
1180 REM *                       *
1190 REM *   Nummer in NR$(0)    *
1200 REM *   Laatste regel:     *
1210 REM *   te draaien cijfer  *
1220 REM *****
1230 OUT&H8,2:'kortsluiten A en B
1240 LOCATE0,23:PRINT"Nummer:";
1250 REM Wachten op kiestoon
1260 FOR V=1 TO 1000: NEXT V
1270 FOR L=1 TO LEN(NR$(0))
1280 IF MID$(NR$(0),L,1)="-" THEN 1290
1285 IF MID$(NR$(0),L,1)="_" THEN 1290 ELSE 1320
1290 PRINT"-";
1300 FOR V=1 TO 1000: NEXT V: ' wacht op kiestoon
1310 GOTO 1420
1320 IF MID$(NR$(0),L,1)<"0" THEN 1420
1325 IF MID$(NR$(0),L,1)>"9" THEN 1420
1330 PRINT MID$(NR$(0),L,1);
1340 V=VAL(MID$(NR$(0),L,1)): IF V=0 THEN V=10
1350 FOR C=1 TO V
1360 OUT&H8,3: REM Kortsluiten A en B
1370 FOR PA=0 TO 10: NEXT PA
1380 OUT&H8,2: REM Puls A en B
1390 FOR PU=0 TO 10: NEXT PU
1400 FOR P=0 TO 200
1410 NEXT C

```

```
1420 NEXT L
1430 OUT&H8,0: REM Weer normale toestand
1440 RETURN
```

Met de subroutine vanaf regel 1230 wordt het nummer door de computer "gedraaid". Het te "draaien" nummer staat in NR\$(0). Om buiten de stad te bellen moet een streepje in het nummer worden opgenomen, zodat de computer hier even zal wachten alvorens de rest van het nummer te "draaien".

De waarden van PU en PA in de For-Next lus kunnen eventueel iets aangepast worden als er problemen zijn.

5. Parallel input

5.1 Inlezen van de poort

Bij parallelle input wordt de poort gebruikt om signalen van buitenaf in de computer in te lezen. Dit gebeurt door alle acht de kanalen in één keer in te lezen. Vervolgens kan eventueel door de And functie één kanaal geselecteerd worden.

In Basic zijn twee instructies om een I/O poort in te lezen. Dit zijn de instructies "Inp(poort)" en "Waitpoort,getal1,getal2". Met $A = \text{Inp(poort)}$ wordt aan A de waarde van alle acht de kanalen van de poort in de variabele A gedaan.

Bij de instructie "Waitpoort,getal1,getal2" worden ook alle acht kanalen ingelezen, maar nu wordt het gevonden getal eerst geXored met getal2 en dan ge-And met getal1. Getal2 hoeft niet opgegeven te worden, en wordt dan gelijk aan 0 gemaakt. Als aan het einde van deze bewerkingen de uitkomst niet 0 is wordt doorgegaan met het programma. Als wel een 0 gevonden wordt, wordt de Wait-instructie nogmaals uitgevoerd.

Met deze laatste instructie kan met een instructie een alarmschakeling met acht controlepunten maken. De computer is de rest van de tijd dan met het bewaken bezig, en kan nergens anders meer voor gebruikt worden.

Om een kanaal in te lezen, moeten we alle acht kanalen inlezen en dan met de And-functie het ene kanaal opzoeken. Voor kanaal 4 gebeurt dat met de volgende instructie:

```
I=4:A=INP (&H08) AND (2^(I-1))
```

A is "0" als kanaal 4 "0" was, anders varieert de waarde van A met het kanaalnummer. Een programma voorbeeld waarbij en "0" of een "1" wordt geprint, afhankelijk van de toestand van kanaal 4.

```
10 I=4 : REM kanaal nummer 4
20 IF (INP(&H08)AND(2^(I-1)))=0 THEN 30 ELSE 50
30 PRINT "0";
40 GOTO 60
50 PRINT "1";
60 END
```

Dit kan ook in machinetaal gedaan worden, waarbij we hetzelfde kanaal testen:

```
IN    A, (#08)
BIT   3,A      ;test de waarde van het kanaal
JR    Z,NUL
LD    A,"1"    ;vul AKKU met een 1
JR    PRINT   ;print waarde in AKKU
NUL:  LD    A,"0" ;vul akku met een 0
PRINT:CALL #00A2 ;print waarde AKKU -
      RET     ;terug naar Basic
```

In machinetaal kunnen we de kanalen afzonderlijk testen met de instructie

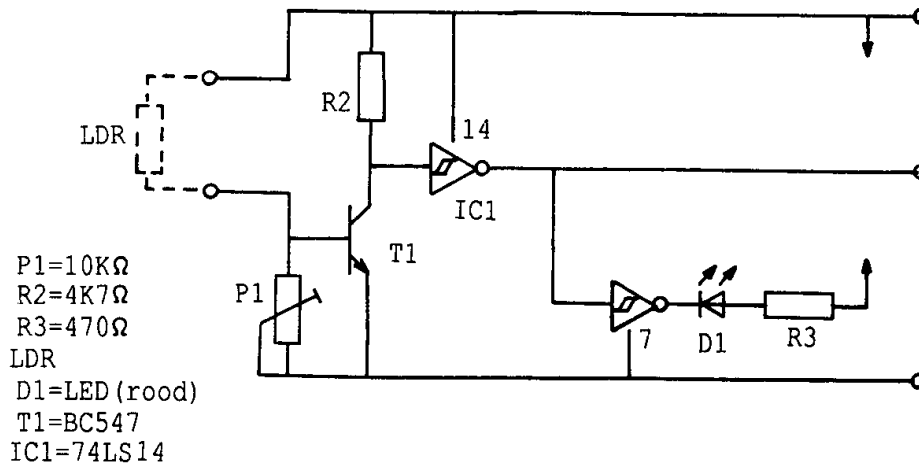
BIT *b*, *n*

b is het kanaalnummer (hier 3) en *n* is het register, in ons geval dus A. De waarde van het kanaal komt omgekeerd in de Z-flag terecht. Door deze te testen kan de Z80 "beslissen" wat er gebeuren moet. Hierna wordt teruggekeerd naar Basic.

5.2 Lichtsluis teller

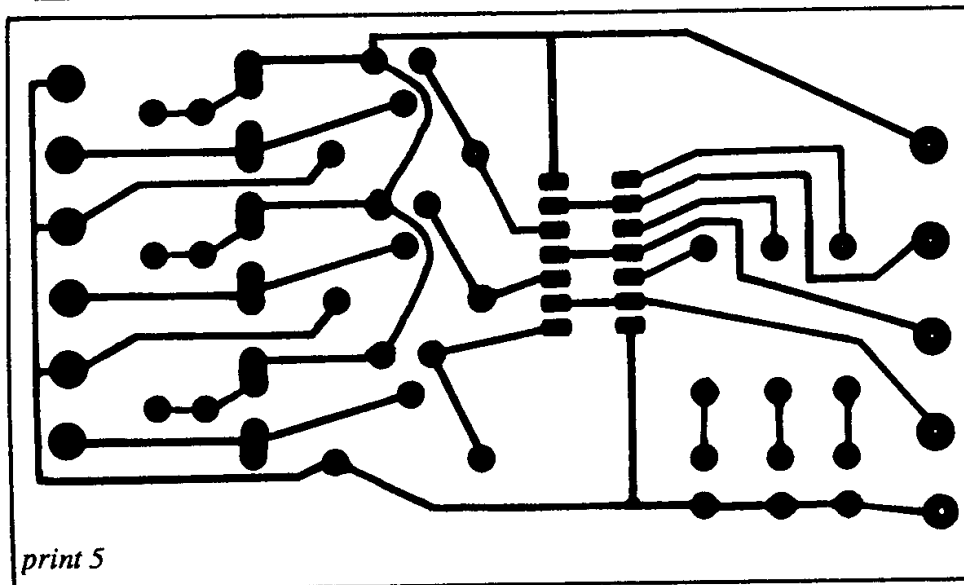
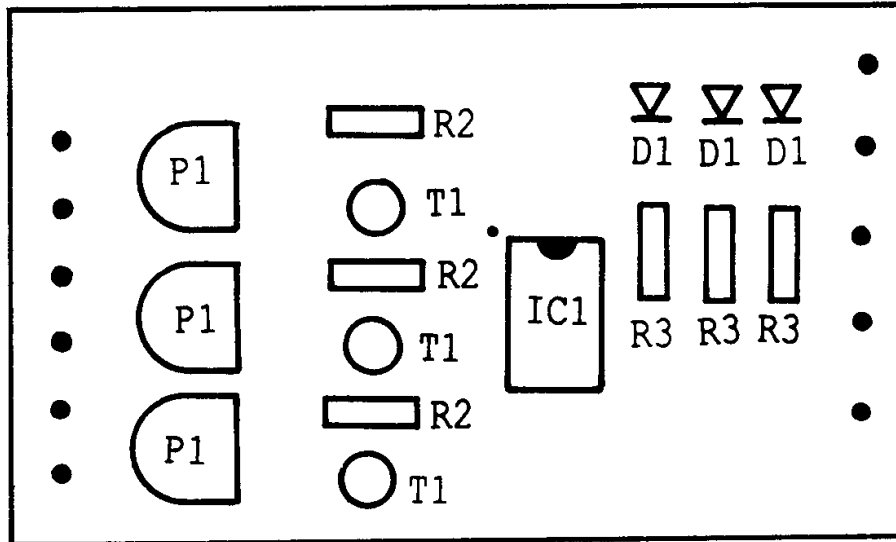
Met de hier beschreven lichtsluis en het programma kan het aantal voorwerpen dat de lichtsluis passeert, geteld worden. De lichtsluis bestaat uit een lamp die een lichtgevoelige weerstand beschijnt. Door de lichtstraal te onderbreken zal de computer het totaal met 1 ophogen.

5.3 De schakeling



Doordat het licht langzaam en gelijkmatig verandert, moet dit omgezet worden in een snel digitaal signaal. De lichtgevoelige weerstand (LDR) zal bij meer licht een lagere weerstand hebben. Hierdoor zal de spanning aan de basis van transistor T1 langzaam stijgen. Op een gegeven moment gaat de transistor in geleiding, waardoor hij de spanning aan de ingang van de Schmitt-trigger laag maakt. De Schmitt-trigger zorgt voor het omzetten van het analoge signaal naar het digitale signaal. Door aan weerstand P1 te draaien kan het lichtniveau, waarop de schakeling reageert, ingesteld worden.

5.4 De bouw en test



print 5

Eerst controleren we weer de print op kortsluitingen en onderbrekingen. Dan solderen we de IP2 pennen in de gaten en het IC voetje. Hierna solderen we de weerstanden R1 en R2, en tenslotte de transistor T1 en LED D1. We sluiten de LDR op één van de drie schakelingen aan. We belichten de LDR en gaan met de hand door de lichtstraal; het LEDje moet nu gaan knipperen.

5.5 *Het programma*

Het programma heeft twee lichtdetektoren nodig, zodat een ronde teller voor een autobaan ontstaat. De tellers tellen van 0 tot 99 en stoppen dan. De print wordt volgens de tekening aangesloten, en het programma geRUNd.

```
10 DIM A(2),F(2),T(2)
20 SCREEN2
30 OPEN "GRP:" AS 1
40 I=1
50 A=INP(&H8)
60 GOSUB 200
70 IF LEN(I$)<2 THEN P$=" "+I$ ELSE P$=I$
80 GOSUB 200
90 IF LEN(I$)<2 THEN P$=P$+" "+I$ ELSE P$=P$+I$
100 PSET(0,0):PRINT#1,P$
110 IF INKEY$="" THEN 50
120 CLOSE #1
130 END
200 F(I)=(AANDI)/I
210 IF F(I)=0 THEN A(I)=0:GOTO 250
220 IF F(I)=1 AND A(I)=1 THEN GOTO 250
230 T(I)=T(I)+1
240 A(I)=1
250 I$=STR$(T(I))
260 I=3-I
270 RETURN
```


6. Analoge uitgang (D/A omzetter)

6.1 Principe analoge uitgang

Als we met de computer op een andere manier iets willen regelen, zoals een apparaat aan of uit zetten, moeten we overgaan op het combineren van meerdere uitgangen tot één uitgang. Dit wordt dan een analoge uitgang genoemd. De uitgangsspanning wordt nu gevormd door de digitale uitgangen gewogen op te tellen. Dit gewogen optellen houdt in dat iedere uitgang een bepaald gedeelte van de voedingsspanning bij de andere gedeelten op te tellen. In ons geval zal de ingang A0 $1/256$ deel van de voedingsspanning bij het totaal optellen, en de ingang A7 $1/2$ deel van de voedingsspanning bij het totaal optellen.

De uitgangsspanning zal hierdoor 256 verschillende niveaus hebben. De uitgangsspanning zal als steeds grotere getallen naar de analoge uitgang worden gestuurd trapvormig stijgen. De ideale lijn (een rechte lijn van de ondergrens naar de bovengrens) wordt bereikt als er vele honderden digitale uitgangen gebruikt worden. Doordat we hier maar acht kanalen combineren tot een analoge uitgang, krijgen we stapjes van 9,4 millivolt als de analoge uitgang van 0 V naar 5 V gaat.

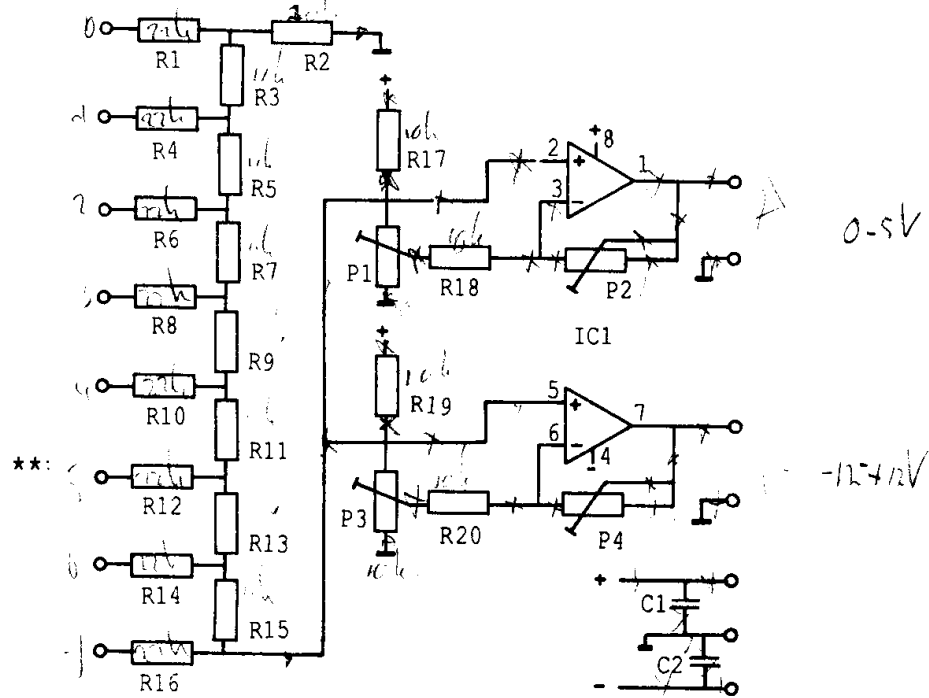
6.2 Schema voor analoge uitgang

Links zijn de aansluitingen naar de IO print getekend. Voor alle acht kanalen een aansluiting. De poort wordt geschakeld als uitgang en stuurt direkt een netwerk van zestien weerstanden aan (R1 tot en met R16). Deze weerstanden zorgen voor het omzetten van de "digitale" spanning (een getal tussen 0 en 255) in een analoge "trap" spanning. Deze spanning mag echter niet belast worden, en loopt van 0,4 V tot 2,4 V. Om dit in een meer bruikbare waarde te veran-

deren, wordt deze spanning met behulp van de OPAMPs A1 en A2 veranderd in:

A1 een spanning van 0 tot 5 V

A2 een spanning van -12 tot +12 V



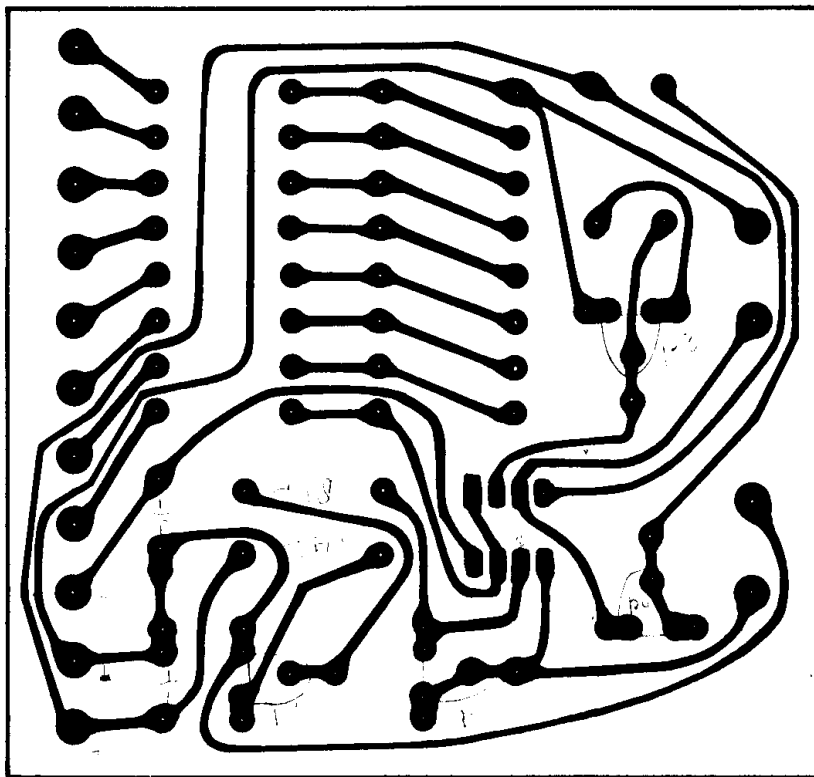
- R1= R4=R6=R8=R10=R12=R14=R16=22K Ω
- R2= 30K Ω
- R3= R5=R7=R9=R11=R13=R15=11k Ω
- R17=R18=R19=R20=10K Ω
- P1= 10K Ω
- P2= 100K Ω
- P3= 10K Ω
- P4= 100K Ω
- C1= 100nF
- C2= 100nF
- IC1=LM358

De weerstanden R1 tot en met R16 moeten nauwkeurig van waarde zijn, zodat de trap die ontstaat even hoge traptreden heeft. Om dit te bereiken gebruiken we weerstanden met een grotere nauwkeurigheid, hiervoor dus 1% weerstanden gebruiken. De beide

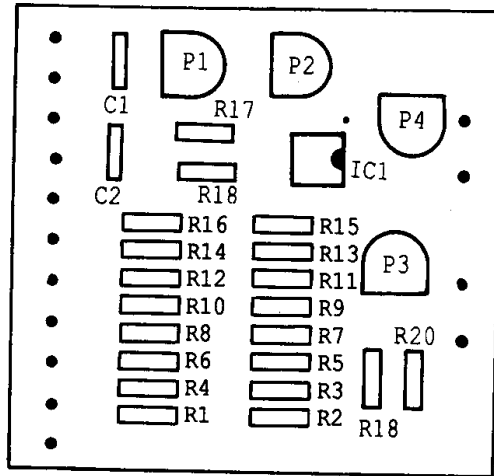
OPAMPs zijn zo geschakeld dat ze de aangeboden ingangsspanning (aan de + ingang) niet belasten, maar wel omzetten. Dit gebeurt door de met P1 en P3 ingestelde spanning er van af te trekken en dit signaal te vermenigvuldigen met een door P2 en P4 ingestelde waarde.

Door het verschillend instellen van P1, P2, P3 en P4 kan een ander uitgangstrajekt (van de laagste tot de hoogste spanning) gekozen worden.

6.3 De bouw van de print



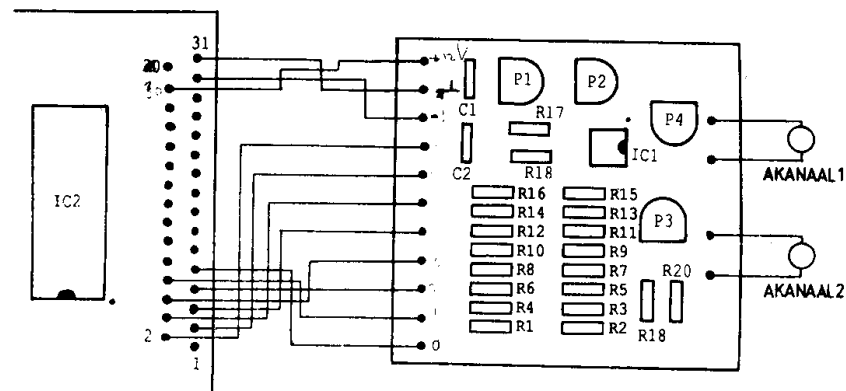
print 6



Eerst controleren we weer of de printbanen niet onderbroken zijn, of kortsluiting met elkaar maken. Als de print in orde is, solderen we de IP2 pennen in de gaten. Dan solderen we het IC voetje op zijn plaats. Vervolgens solderen we de weerstanden en condensatoren. Tot slot solderen we de instelbare weerstanden P1 tot en met P4.

De print is na het insteken van het IC in het voetje en het afregelen klaar voor gebruik.

6.4 De afregeling van de analoge uitgang



We hebben twee kanalen tot onze beschikking die afhankelijk van elkaar een uitgangsspanning geven. We regelen eerst het eerste kanaal af op 0 tot 5 V. Draai P2 geheel met de klok mee en P2 geheel tegen de klok in.

Sluit de print aan op de IO print volgens bovenstaande tekening en tik op de computer in:

```
OUT&H0B, &H8B:OUT&H08, 0<R>
```

Sluit een voltmeter aan op de uitgang A-kanaal1. Draai P1 met de klok mee tot de voltmeter 0 V aanwijst in het kleinste bereik. Tik nu op de computer in:

```
OUT&H08, 255<R>
```

Regel met P2 de uitgangsspanning van A-kanaal1 naar 5 V. De uitgang A-kanaal1 is nu afgeregeld. Hetzelfde doen we nu met A-kanaal2, echter omdat de 0 V in het midden tussen de -12 en +12 V ligt, doen we eerst op de computer:

```
OUT&H08, 128<R>
```

We verbinden de voltmeter nu met A-kanaal2 en draaien beide potentiometers geheel tegen de klok in. Door P3 met de klok mee te draaien kan de uitgangsspanning op 0 V in het kleinste bereik afgeregeld worden. Hierna tikken we op de computer in:

```
OUT&H08, 255<R>
```

en draaien met P4 de uitgangsspanning naar 12 V. Ter controle doen we nu nog een

```
OUT&H08, 0<R>
```

waarbij we op A-kanaal2 -12 V moeten meten. Als dit klopt, dan is de uitgang A-kanaal2 afgeregeld. Door de instelling van de regelbare weerstanden te veranderen kunnen andere uitgangsspanningen gekozen worden.

6.5 Modeltrein besturing

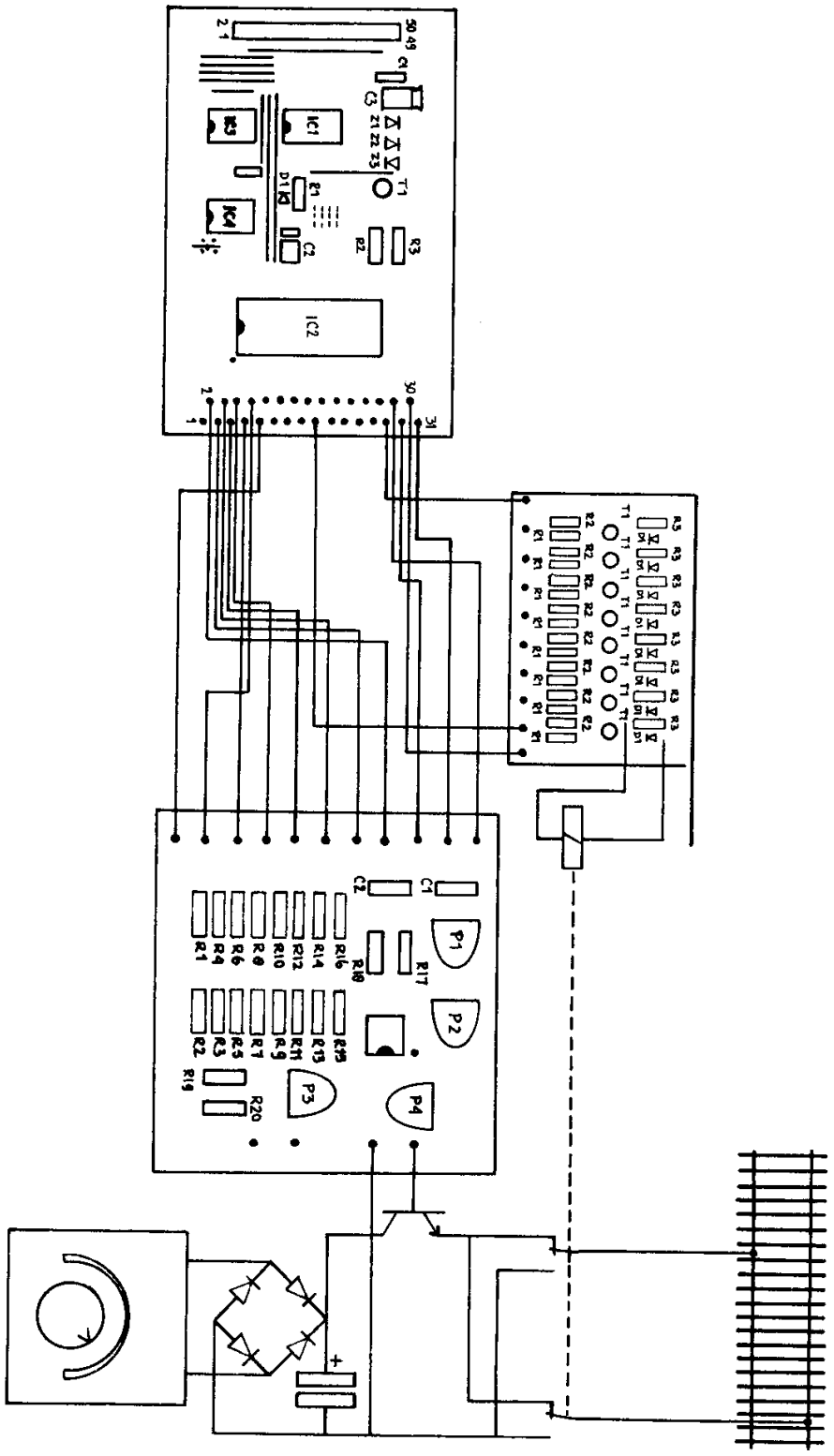
Met het hier getoonde programma en de digitaal-analoog omzetter zullen we de motor van een modeltrein vooruit en achteruit laten draaien, en tevens de snelheid variëren.

De snelheid verhogen gebeurt met de joystick in de stand *up* (1), verlagen gebeurt met de joystick in de stand *down* (5). De rijrichting wordt veranderd als op de vuurknop wordt gedrukt en de trein stil staat.

De trein wordt volgens de tekening op de volgende pagina aangesloten, waarbij gebruik wordt gemaakt van de test-print en de D/A print. De treinfrafo-spanning wordt afgeregeld op 14 V.

```
10 OUT&HB,&H8A
20 ON INTERVAL=150 GOSUB 90:INTERVAL ON
30 SN=0:REM snelheid is 0
40 RR=0:REM rijrichting vooruit
50 IF STRIG(1) THEN IF SN=0 THEN RR=1-RR
60 OUT&H8,SN:REM snelheid naar D/A omzetter
70 OUT&HA,RR:REM rijrichting naar relais
80 GOTO 50
90 IF STICK(1)=1 THEN SN=SN+15 : IF SN>255 THEN SN=255
100 IF STICK(1)=5 THEN SN=SN-15 : IF SN<0 THEN SN=0
110 RETURN
```

Door het programma te RUNnen kan de trein via de joystick gestuurd worden. Als verbetering van de besturing kan een PDL uit het eerste hoofdstuk gebruikt worden, waardoor een nettere regeling mogelijk is.



7. Analoge ingang (A/D omzetter)

7.1 *Het principe van de analoge ingang*

Een analoge ingang wordt gemaakt door een analoge spanning te "vertalen" naar een getal waar door de computer mee gerekend kan worden. Dit wordt bereikt door een aantal vast ingestelde analoge spanningen bij elkaar op te tellen, totdat de te meten spanning gevonden is. Dit is de reden, dat bij een analoog/digitaal omzetter altijd een synchronisatie-ingang en uitgang aanwezig is.

De ingang vertelt de computer dat de omzetting voltooid is. Met de uitgang geeft de computer een startsignaal, waardoor de omzetter begint met omzetten.

Meestal bestaat de omzetter uit een gewone teller die bij 0 begint en dan telt tot de waarde gevonden is. Bij te hoge analoge spanningen zal de teller tot het maximum tellen en zo de te hoge waarde aan de computer kenbaar maken.

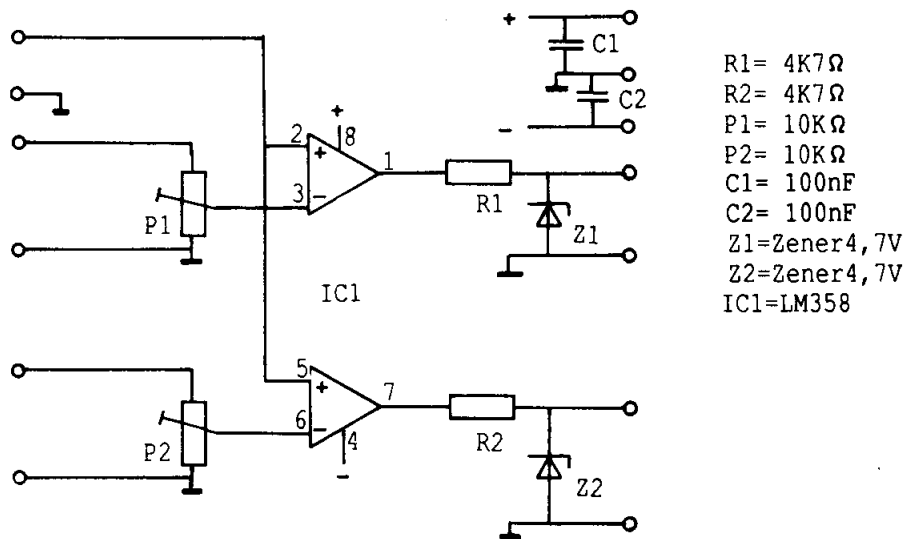
Deze vorm van omzetten heeft als nadeel dat lage spanningen snel gevonden worden, terwijl voor hoge spanningen de omzetting lang tot zeer lang kan zijn. De analoge spanningswaarde wordt gevonden door deze te vergelijken met een door de teller opgewekte spanning. Deze spanning is trapvormig van vorm, zoals de uitgangsspanning van de digitaal/anoalog omzetter uit het vorige hoofdstuk. Als de trapvormige spanning groter wordt dan de te meten analoge spanning dan zal de teller stoppen, zodat de trapvormige spanning niet verder stijgt. Er wordt nu tevens gemeld dat het omzetten voltooid is en de omgezette digitale waarde op de uitgang van de teller staat.

Tijdens het omzetten moet de computer wachten tot de omzetter klaar is, daarom laten we de computer de functie van de teller overnemen. Hierdoor kan de startuitgang komen te vervallen. De com-

puter maakt nu via de digitaal/analoo omzetter de trapvormige spanning, en via een ingang hoeft nu alleen nog maar getest te worden of de waarde al gevonden is of niet. De teller wordt met een eenvoudige For-Next lus gemaakt, waarbij een variabele aan het einde de omgezette waarde bevat.

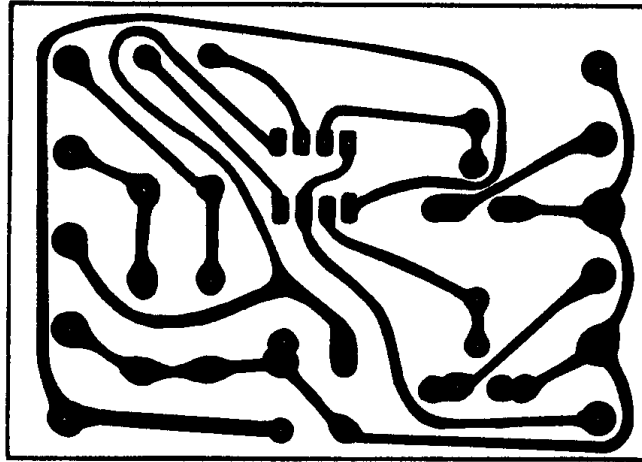
7.2 De analoog/digitaal omzetter

Doordat de computer samen met de digitaal/analoo omzetter uit het vorige hoofdstuk het meeste werk doet, kan de analoog/digitaal omzetter erg eenvoudig zijn. Uit het schema blijkt al dat op een print twee omzetter zijn ondergebracht. De omzetter bestaat uit een OPAMP die als vergelijker is geschakeld.

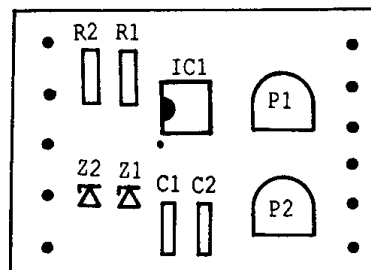


De uitgang van de OPAMP wordt met een weerstand en een zenerdiode begrensd op voor de I/O poort veilige spanning. Een ingang gaat rechtstreeks naar de digitaal/analoo omzetter, terwijl de andere ingang via een regelbare weerstand naar de te meten analoge spanning gaat. Met deze regelbare weerstand kan het bereik van de omzetter vergroot worden.

De print



print 7



We controleren eerst de print op onderbrekingen in de sporen en kortsluiting tussen de sporen. Dan solderen we de negen IP2 pennen in de daarvoor bestemde gaten en het IC voetje. Hierna solderen we de twee regelbare weerstanden en de gewone weerstanden. Tenslotte solderen we de twee zenerdioden (let op het streepje op het huis van de zenerdiode, dit is de pijlpunt). We drukken nu het IC in het voetje, met de uitsparing naar P2. We sluiten nu de digitaal/analoomzetter en de analoog/digitaalozetter aan op de I/O print volgens de tekening en kunnen nu spanningen meten van 0 tot 5 V. Dit doen we met het volgende programma:

```

10 OUT&H0C,&H8B
20 GOSUB 1000
30 LOCATE 0,0:PRINT"De spanning is:";I*5/255;" Volt."
40 GOTO 20
1000 I=0
1010 OUT &H08,I
1020 IF INP(&H0B) AND &H01 = 0 THEN I=I+1: GOTO 1010
1030 RETURN

```

Deze routine heeft tijdens het omzetten een van de te meten spanning afhankelijke tijdsduur. Bij hogere spanningen zal de tijd langer duren. Om deze tijd konstant te houden wordt de subroutine gewijzigd in de hieronder gegeven snelzoek routine, die in acht stappen de analoge waarde omzet.

```

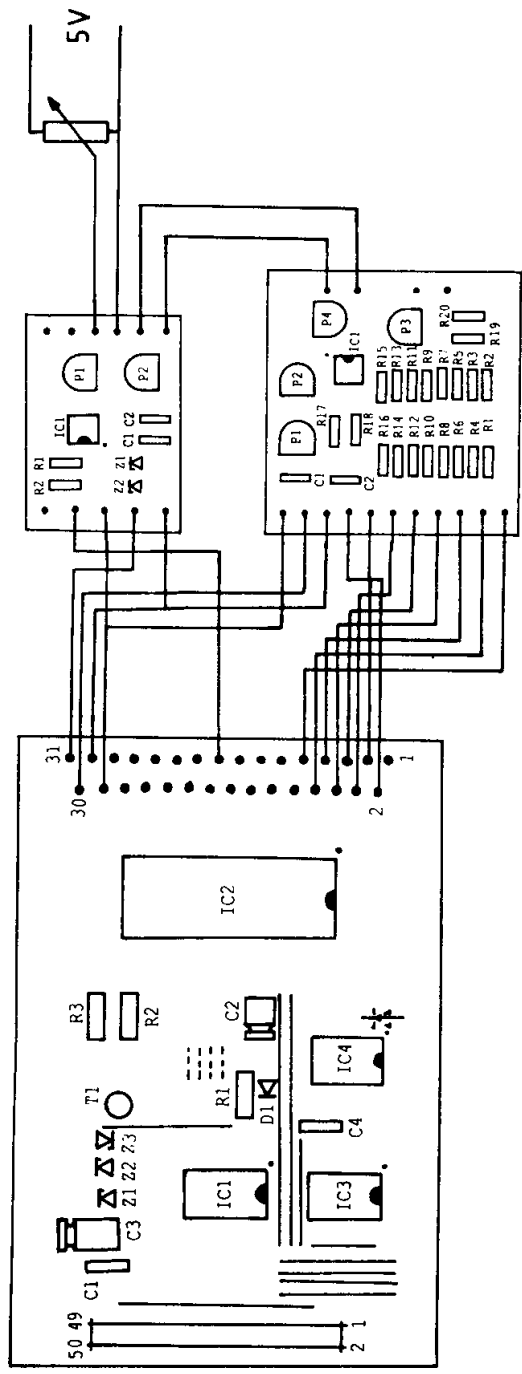
1000 OUT &H08,0
1010 FOR I=7 TO 0 STEP -1
1020 OUT &H08,INP(&H08)+2^I
1030 IF INP(&H0B) AND &H01 = 0 THEN O
UT &H08,INP(&H08)-2^I
1040 NEXT I
1050 I=INP(&H08)
1060 RETURN

```

Deze routine telt, iedere keer dat de For-Next lus uitgevoerd wordt, de helft van de vorige referentiespanning bij de al aanwezige spanning op. Als de waarde nu te groot is, wordt dit er weer van afgetrokken, waarna de volgende test met weer de helft wordt uitgevoerd. In acht keer is de waarde nu gevonden en wordt teruggekeerd naar het hoofdprogramma.

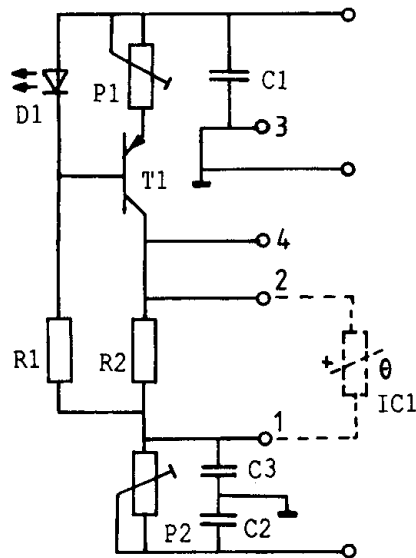
7.3 *Temperatuurmeting*

Met deze toepassing laten we de computer de temperatuur meten. We maken gebruik van een IC'tje dat zich gedraagt als een temperatuursafhankelijke weerstand. Bij hoger wordende temperaturen neemt de weerstand toe, en omgekeerd neemt deze af. Om nu uit deze weerstandsverandering een spanningsverandering te krijgen, sturen we door de sensor een konstante stroom, die we met behulp van T1 en wat extra componenten maken. Hierdoor zal nu de span-



ning veranderen met de temperatuur. Deze spanning wordt gemeten met de A/D omzetter.

Het schema

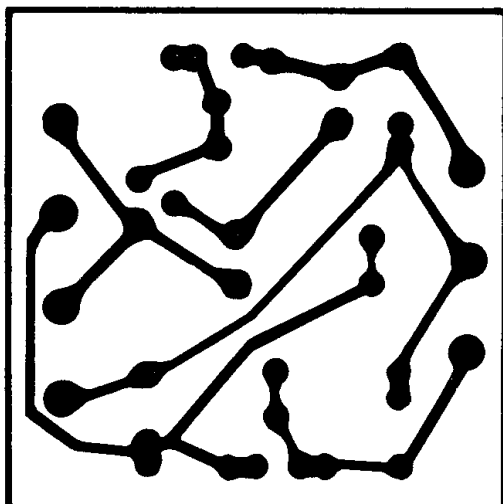


Uitlezing	Temperatuur
255	
228	+150 °C
178	+100 °C
100	+22 °C
78	0 °C
28	-50 °C
0	

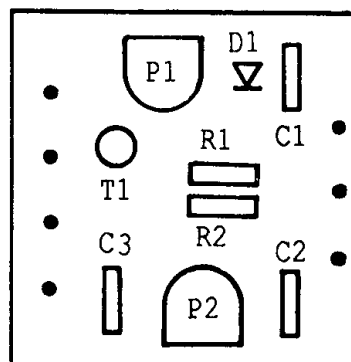
R1= 2K7Ω
 R2= 6K8Ω(1%)
 P1= 2K5Ω
 P2= 2K5Ω
 C1= 100nF
 C2= 100nF
 C3= 100nF
 D1=LED(rood)
 T1=BC557
 IC1=KTY10C

Door middel van R2, P1, T1 en D1 wordt de stroombron opgebouwd. D1 is een rode LED, dit is om een goede instelling voor de transistor T1 te krijgen en tevens een aanwijzing dat de stroombron stroom levert. Met P1 is de stroom instelbaar die door R1 en het IC lopen. Tenslotte wordt met P2 het nulpunt (bij 0°C) vastgelegd op 0 V. Hierdoor zullen temperaturen onder nul een negatieve spanning veroorzaken. De gehele temperatuurvoeler kan via de IO print gevoed worden uit de computer.

De bouw



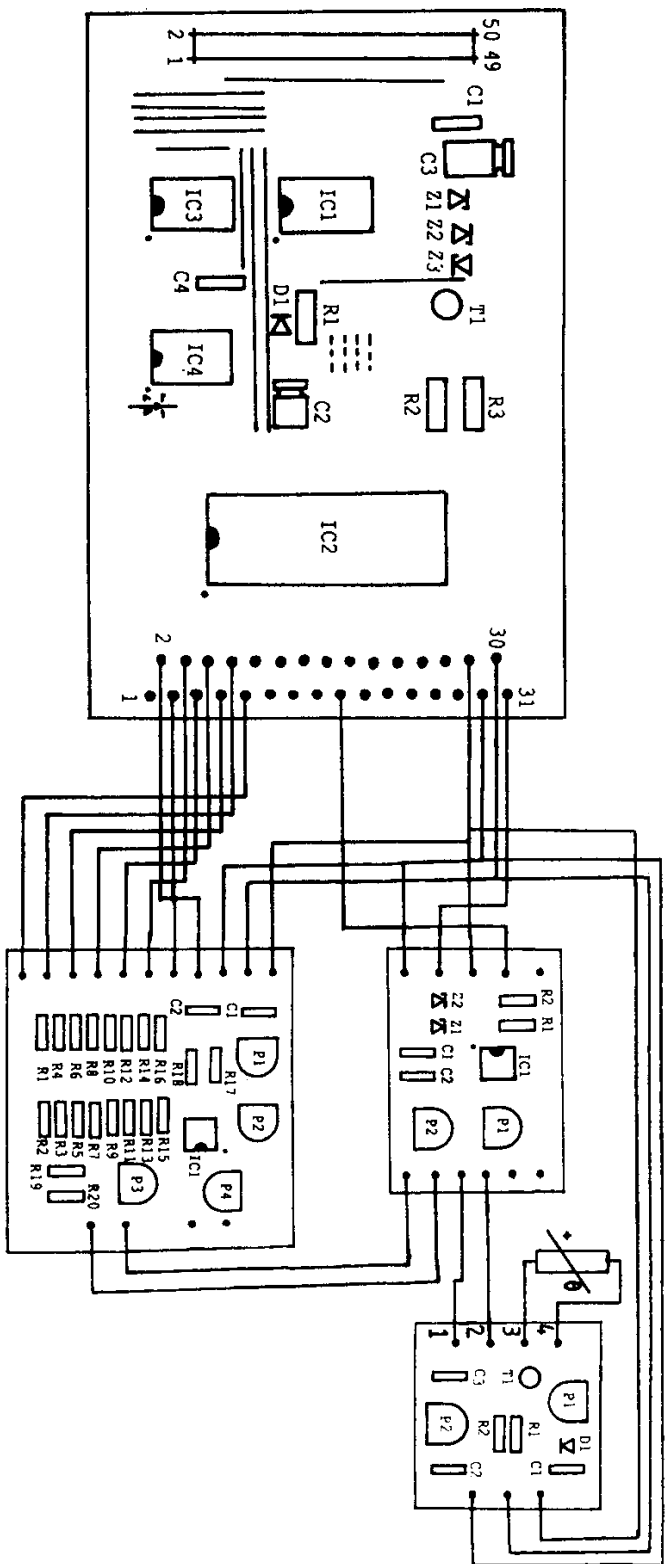
print 8



Na de print gecontroleerd te hebben op kortsluitingen en onderbrekingen kunnen we beginnen met de opbouw van de print. Als eerste solderen we de zeven IP2 pennen en de weerstanden. Hierna solderen we de regelbare weerstanden en de condensatoren. Als laatste solderen we de rode LED en de transistor. De print is nu gereed om afgeregeld te worden. De temperatuursensor solderen we aan een maximaal tien meter lange afgeschermd kabel, en zorgen er voor dat er geen vocht bij de aansluitingen kan komen, door de sensor in hars in te gieten. De afscherming van de kabel moet altijd aan pen 4 zijn bevestigd.

De afregeling

Om de temperatuursensor af te regelen hebben we naast de universeelmeter ditmaal ook nog een glas smeltend ijs en een pannetje kokend water nodig. Dit zijn namelijk twee temperaturen die konstant op een vaste waarde blijven, onafhankelijk van de omgevingstemperatuur. Bij het afregelen zullen we de temperatuursensor iedere keer ongeveer een minuut de tijd moeten gunnen om de omgevingstemperatuur aan te nemen.



Als eerste gaan we de D/A omzetter aanpassen aan de voor ons doel benodigde bereik. We draaien P3 en P4 geheel tegen de klok in. Nu tikken we op de computer het volgende kommando in:

```
OUT&H0B, &H8B:OUT&H08, 78<R>
```

We draaien aan P3 tot de uitgang 0 V aanwijst in het kleinste bereik. Het nulpunt is nu ingesteld. Nu gaan we het nulpunt van de temperatuursensor instellen. Dit doen we door de spanning tussen de punten 3 en 4 te meten, zonder dat de temperatuursensor is aangesloten. We verdraaien P1 net zolang, totdat we 6,8 V meten. Nu sluiten we de temperatuursensor aan en hangen deze in het glas met smeltend ijswater. We wachten enige tijd om de sensor de gelegenheid te geven zich aan te passen, waarna we de spanning tussen de punten 1 en 2 meten.

We draaien nu aan P2, tot da de spanning in het kleinste bereik 0 V is. Om de schakeling verder af te regelen hangen we de sensor in het kokende water. We typen het volgende programma in:

```
10 OUT&H0B, &H8B
20 FOR I=0 TO 255
30 OUT&H08, I:J=I-1
40 IF INP(&H0A)AND&H01 THEN I=255
50 NEXT I
60 PRINT J
70 GOTO 20
```

en RUNnen dit. We draaien nu aan P4 (op de A/D print), totdat het getal 178 verschijnt. De temperatuursensor is nu afgeregeld. Door regel 60 te veranderen in:

```
60 LOCATE0,0:PRINTJ-78, "graden Celsius"
```

krijgen we een temperatuursaanwijzing door de computer.

8. EPROMs programmeren

8.1 *Het EPROM*

Een EPROM (= Erasable Programmable Read Only Memory) is voor de computer hetzelfde als een ROM (= Read Only Memory). Voor de computerbezitter is er echter een groot verschil. Van een EPROM kunnen we zelf bepalen wat voor programma er in komt, na enige tijd kunnen we het programma weer wissen en veranderen door andere gegevens. Het schrijven van het programma naar EPROM doen we met behulp van de computer en de IO poort. Het wissen gebeurt met ultraviolet licht, dat via het venster in het IC op de chip geschreven wordt.

Een "leeg" EPROM bevat op iedere geheugenplaats een "1". Na het wissen moet dit altijd gecontroleerd worden, omdat de wistijd (ongeveer twintig minuten) toeneemt met de leeftijd van het EPROM, en het aantal malen dat het EPROM is gewist. Het programmeren kost vijftig duizendste seconde per geheugenplaats, dus in totaal duurt het programmeren twee minuten. Voor het programmeren van het EPROM hebben we een speciale programmeerspanning nodig. Als het programmeren gedaan is, wordt de EPROM op de computer aangesloten, waarna het programma uitgevoerd kan worden.

Bij een MSX computer kunnen we de volgende soorten programma's opslaan:

1. Machinetaalprogramma's die automatisch starten. De programma's moeten eindigen op een Ret (&HC9) instructie.
2. Machinetaalprogramma's die door de Basic Call-instructie worden gestart.
3. Machinetaalprogramma's die een randapparaat (printer, disk)

besturen.

4. Basic programma's die automatisch starten.

Door het slotmechanisme van de MSX computer kan het EPROM het hele geheugen door op de computer aangesloten worden. Het gebied van &H4000 tot &H4010 bevat getallen die het bovenstaande aan de computer vertellen.

8.2 *Het programmeerprogramma*

Het programma is alleen geschikt voor het programmeren van het EPROM type 2716. Het bestaat uit twee gedeelten, te weten een stuk Basic en een stuk machinetaal. Het stuk machinetaal zorgt voor het programmeren, het controleren en het kopiëren van het EPROM. Dit is gedaan, omdat de gegevens die van en naar het EPROM gebracht worden, in Basic teveel tijd zouden kosten.

Tijdens het programmeren van het EPROM zijn de tijden zo kritisch dat dit wel in machinetaal moet. De gegevens die naar het EPROM gaan komen uit een tabel in het geheugen. Deze tabel is 2048 bytes lang en staat direct voor het machinetaal-gedeelte en wordt tijdens het starten gevuld met de waarde &HFF (leeg EPROM). Hierna wordt om het startadres van het EPROM gevraagd, zodat de gegevens altijd op de juiste plaats terecht komen. De tabel kan vanuit Basic "gelist" worden, waarna de gegevens met de hand veranderd kunnen worden.

Ook kan de tabel vanuit het EPROM gevuld worden, zodat kopieën van de inhoud gemaakt kunnen worden. Het machinetaalgedeelte bestaat uit vier verschillende subroutines die met A=USR1(A) tot en met A=USR4(A) gestart worden. De routines worden hieronder gelist en uitgelegd.

Listing A=USR1(A), PROGRAM (Programmeren EPROM)

```

F3      PROGRAM:DI      ;geen interrupts meer!
2A0000      LD      HL, (START) ;laad start van tabel.
ED5B0400    LD      DE, (PRMST) ;laad start EPROM.
3E80       LD      A, #80      ;programmeer 8255 met
D30B       OUT     (#0B),A     ;alle kanalen uitgang.
3E20       LD      A,%00100000 ;programmeerspanning op
D309       OUT     (#09),A     ;5 V.
7B         PRGM05:LD     A,E     ;breng EPROM-adres naar
D30A       OUT     (#0A),A     ;EPROM. Alle tien de
7A         LD      A,D         ;adresbits. Bewaar tij-
E607       AND     #07        ;delijk in C register.
4F         LD      C,A
F620       OR      %01000000
D309       OUT     (#09),A
7E         LD      A, (HL)     ;Haal gegevens uit tabel
D308       OUT     (#08),A     ;en breng naar EPROM.
79         LD      A,C         ;Start programmeren van
D309       OUT     (#09),A     ;lokatie
01E31A     LD      BC, TIME
0B         PRG10: DEC    BC     ;Wacht 20 milliseconde
79         LD      A,C
B0         OR      B
20FB       JR      NZ, PRG10
7A         LD      A,D
E607       AND     #07        ;Haal programmeerpuls
F6A0       OR      %10100000   ;weg
D309       OUT     (#09),A
ED4B0200   LD      BC, (EIND)  ;Hele buffer gehad
ED42       SBC    HL, BC
09         ADD    HL, BC
23         INC    HL
13         INC    DE          ;Volgende lokatie
38D5       JR      C, PRG05    ;Nee - terug naar begin
3E9B       LD      A, #9B
D30B       OUT     (#0B),A
210000     LD      HL, #0000
22F8F7     LD      (#F7F8),HL  ;Test of goed
3E02       LD      A, #02
3263F6     LD      (#F663),A
FB         EI                ;TIJD NIET MEER KRITIEK

```

Listing A=USR2(A), EPTST (Testen EPROM)

```
3A63F6   EPTST: LD   A, (#F663)
FE02                CP   #02
C0                RET  NZ
3AF8F7                LD   A, (#F7F8)   ;Test of kontrole flag
47                LD   B,A
2A0000                LD   HL, (START)
ED5B0400            LD   DE, (PRMST)
3E90                LD   A, #90
D30B                OUT  (#0B), A
3E20                LD   A, %0010000
D309                OUT  (#09), A
7B                EPT05: LD  A, E
D30A                OUT  (#0A), A
7A                LD   A, D           ;Adres EPROM naar EPROM
E607                AND  #07
F6E0                OR   %11100000
D309                OUT  (#09), A
DB08                IN   A, (#08)       ;Lees lokatie
CB08                RRC  B           ;Test flag
3803                JR   C, EPT10
BE                CP   (HL)         ;Gelijk ?
1802                JR   EPT15
2F                EPT10: CPL                ;Geen 0 waar 1 moet
A6                AND  (HL)         ;komen
2011                EPT15: JR  NZ, EPT20
78                LD   A, B           ;Bewaar flag
B7                OR   A
ED4B0200            LD   BC, (EIND)       ;Eind bereikt
ED42                SBC  HL, BC
09                ADD  HL, BC
23                INC  HL
13                INC  DE
47                LD   B, A           ;Zet flag terug
38DB                JR   C, EPT05
210000                LD   HL, #0000       ;Geen fout
22F8F7            EPT20: LD   (#F7F8), HL
3E9B                LD   A, #9B
D30B                OUT  (#0B), A
C9                RET
```

Listing A=USR3(A) EPRD (EPROM read)

```
2A0000  EPRD: LD HL, (START)
ED5B0400 LD DE, (PRMST)
3E90 LD A, #90
D30B OUT (#0B), A
3E20 LD A, %00100000
D309 OUT (#09), A
7B . EPR05: LD A, E ;EPROM adres naar EPROM
D30A OUT (#0A), A
7A LD A, D
E607 AND #07
F6E0 OR %11100000
D309 OUT (#09), A
DB08 IN A, (#08) ;Lees lokatie
77 LD (HL), A ;Zet in tabel
ED4B0200 LD BC, (EIND)
B7 OR A ;Eind bereikt
ED42 SBC HL, BC
09 ADD HL, BC
23 INC HL
13 INC DE
38E7 JR C, EPR05
3E9B LD A, #9B
D30B OUT (#0B), A
C9 RET
```

Listing A=USR4(A) BFFLL (Buffer vullen)

```
3A63F6 BFFLL: LD A, (#F663)
FE02 CP #02
C0 RET NZ
3AF8F7 LD A, (#F7F8) ;Laad te vullen waarde
2A0200 LD HL, (EIND)
ED5B0000 LD DE, (START)
ED52 SBC HL, DE ;Bereken lengte
4D LD C, L
44 LD B, H
6B LD L, E
62 LD H, D
13 INC DE
77 LD (HL), A ;Waarde in buffer
```

```

EDB0          LDIR          ;Vul hele buffer
C9            RET

```

Deze subroutines worden door het programma onder HIMEM in het geheugen geplaatst, waarna de buffer gevuld wordt. HIMEM (&HFC4A) wordt nu onder het begin van de buffer gezet, zodat machinetaal en buffer beschermd zijn.

```

10 REM *****
20 REM *          EPROM- PROGRAMMER          *
30 REM *          -----                    *
40 REM * (C)1987 UITGEVERIJ STARK-TEXEL BV *
50 REM *****
60 REM *          MACHINETAAL IN GEHEUGEN    *
70 REM *****
80 AD=PEEK(&HFC4A)+256*PEEK(&HFC4B)
90 CLEAR 200,AD-225
100 DATA 00,00,00,00,00,00,*1
110 REM Listing A=USR1(A), PROGRM (Programmeren EPROM)
120 DATA F3,2A,**,00,ED,5B,**,04,3E,80,D3,0B,3E,20,D3,09
130 DATA 7B,D3,0A,7A,E6,07,4F,F6,20,D3,09,7E,D3,08,79,D3
140 DATA 09,01,E3,1A,0B,79,B0,20,FB,7A,E6,07,F6,A0,D3,09
150 DATA ED,4B,**,02,ED,42,09,23,13,38,D5,3E,9B,D3,0B,21
160 DATA **,00,22,F8,F7,3E,02,32,63,F6,FB,*2
170 REM Listing A=USR2(A), EPTST (Testen EPROM)
180 DATA 3A,63,F6,FE,02,C0,3A,F8,F7,47,2A,**,00,ED,5B,**
190 DATA 04,3E,90,D3,0B,3E,20,D3,09,7B,D3,0A,7A,E6,07,F6
200 DATA E0,D3,09,DB,08,CB,08,38,03,BE,18,02,2F,A6,20,11
210 DATA 78,B7,ED,4B,**,02,ED,42,09,23,13,47,38,DB,21,00
220 DATA 00,22,F8,F7,3E,9B,D3,0B,C9,*3
230 REM Listing A=USR3(A) EPRD (EPROM read)
240 DATA 2A,**,00,ED,5B,**,04,3E,90,D3,0B,3E,20,D3,09,7B
250 DATA D3,0A,7A,E6,07,F6,E0,D3,09,DB,08,77,ED,4B,**,02
260 DATA B7,ED,42,09,23,13,38,E7,3E,9B,D3,0B,C9,*4
270 REM Listing A=USR4(A) BFFLL (Buffer vullen)
280 DATA 3A,63,F6,FE,02,C0,3A,F8,F7,2A,**,02,ED,5B,**,00
290 DATA ED,52,4D,44,6B,62,13,77,ED,B0,C9
300 AD=PEEK(&HFC4A)+256*PEEK(&HFC4B)
310 CLS: LOCATE10,10
320 PRINT"MACHINECODE LADEN"

```

```

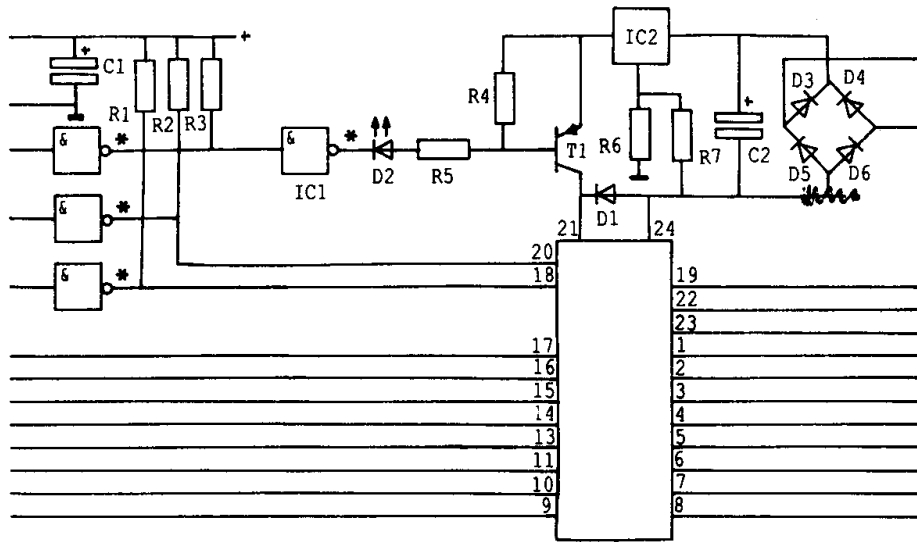
330 RESTORE 100
340 FOR I=AD TO AD+225
350 READ D$
360 IF LEFT$(D$,1)<>"*" THEN 480
370 IF MID$(D$,2,1)="*" THEN 440
380 R=VAL(MID$(D$,2,1))
390 ON R GOTO 400,410,420,430
400 DEFUSR1=I:I=I-1:GOTO 490
410 DEFUSR2=I:I=I-1:GOTO 490
420 DEFUSR3=I:I=I-1:GOTO 490
430 DEFUSR4=I:I=I-1:GOTO 490
440 READ D$
450 R=AD+VAL("&H"+D$)
460 POKEI,(R-256*INT(R/256)): I=I+1
470 POKEI,INT(R/256):GOTO 490
480 POKE I,VAL("&H"+D$)
490 NEXT I
500 REM *****
510 REM *      POKE BUFFER ADRES      *
520 REM *****
530 CLEAR 200,AD-2048
540 LOCATE10,11
550 INPUT"START ADRES EPROM: ";D$
560 EA=VAL("&H"+D$)
570 AD=PEEK(&HFC4A)+256*PEEK(&HFC4B)
580 POKEAD+2048,(AD-256*INT(AD/256))
590 POKEAD+2049,INT(AD/256)
600 POKEAD+2050,((AD+2047)-256*INT((AD+2047)/256))
610 POKEAD+2051,INT((AD+2047)/256)
620 POKEAD+2052,(EA-256*INT(EA/256))
630 POKEAD+2053,(INT(EA/256)AND&HFF)
640 REM *****
650 REM *      VUL BUFFER MET &HFF      *
660 REM *****
670 DEFINT B :REM variable tussen BASIC en ML routines
680 LOCATE10,12: PRINT" BUFFER VULLEN"
690 B=255: B=USR4(B)
700 REM *****
710 REM *      DISPLAY MENU      *
720 REM *****
730 CLS: LOCATE8,12

```

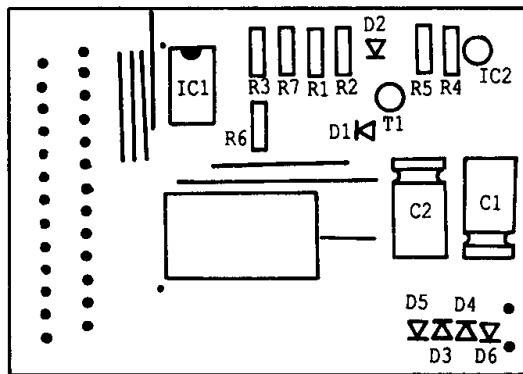
```

740 PRINT"M= Verander geheugen":LOCATE8,13
750 PRINT"R= Lees EPROM in geheugen": LOCATE8,14
760 PRINT"P= Programmmeer EPROM": LOCATE8,15
770 PRINT"C= Controleer EPROM": LOCATE8,16
780 PRINT"V= Vul geheugen";
790 I$=INKEY$: B=0
800 IF I$="M" OR I$="m" THEN 890
810 IF I$="R" OR I$="r" THEN 1130
820 IF I$="P" OR I$="p" THEN 1180
830 IF I$="C" OR I$="c" THEN 1260
840 IF I$="V" OR I$="v" THEN 1350
850 GOTO 790
860 REM *****
870 REM * KOMMANDO VERANDER GEHEUGEN *
880 REM *****
890 CLS
900 FOR I=AD TO AD+2047
910 PRINTRIGHT$("0000"+HEX$(I-AD+EA),4);
920 D$=RIGHT$("00"+HEX$(PEEK(I)),2)
930 PRINT " ";D$;
940 I$=INKEY$: IF I$="" THEN 940
943 AT=AD+2046
945 AB=AD-1
950 IF I$="+" THEN IF I>AT THEN I=AT:GOTO 1080 ELSE 1080
960 IF I$="=" THEN IF I>AT THEN I=AT:GOTO 1080 ELSE 1080
970 IF I$="-" THEN IF I<AB THEN I=AB:GOTO 1080 ELSE 1080
980 IF I$=" " THEN IF I<AB THEN I=AB:GOTO 1080 ELSE 1080
990 IF I$="S" OR I$="s" THEN I=AD+2047: GOTO 1080
1000 PRINT I$;
1010 D$=RIGHT$(D$+I$,2)
1020 I$=INKEY$: IF I$="" THEN 1020
1030 IF ASC(I$)=13 THEN 1070
1040 PRINT I$;
1050 D$=RIGHT$(D$+I$,2)
1060 GOTO 1020
1070 POKE I,VAL("&H"+D$)
1080 PRINT: NEXT I
1090 GOTO 730
1100 REM *****
1110 REM * KOMMANDO READ EPROM *
1120 REM *****

```

- 12L
- | | |
|----------|---------------|
| R1=390Ω | D1=1N4148 |
| R2=390Ω | D2=LED (rood) |
| R3=390Ω | D3=1N4148 |
| R4= 10KΩ | D4=1N4148 |
| R5= 1KΩ | D5=1N4148 |
| R6=100Ω | D6=1N4148 |
| R7=390Ω | T1=BC557 |
| C1=100μF | IC1=74LS26 |
| C2=100μF | IC2=78L24 |



```

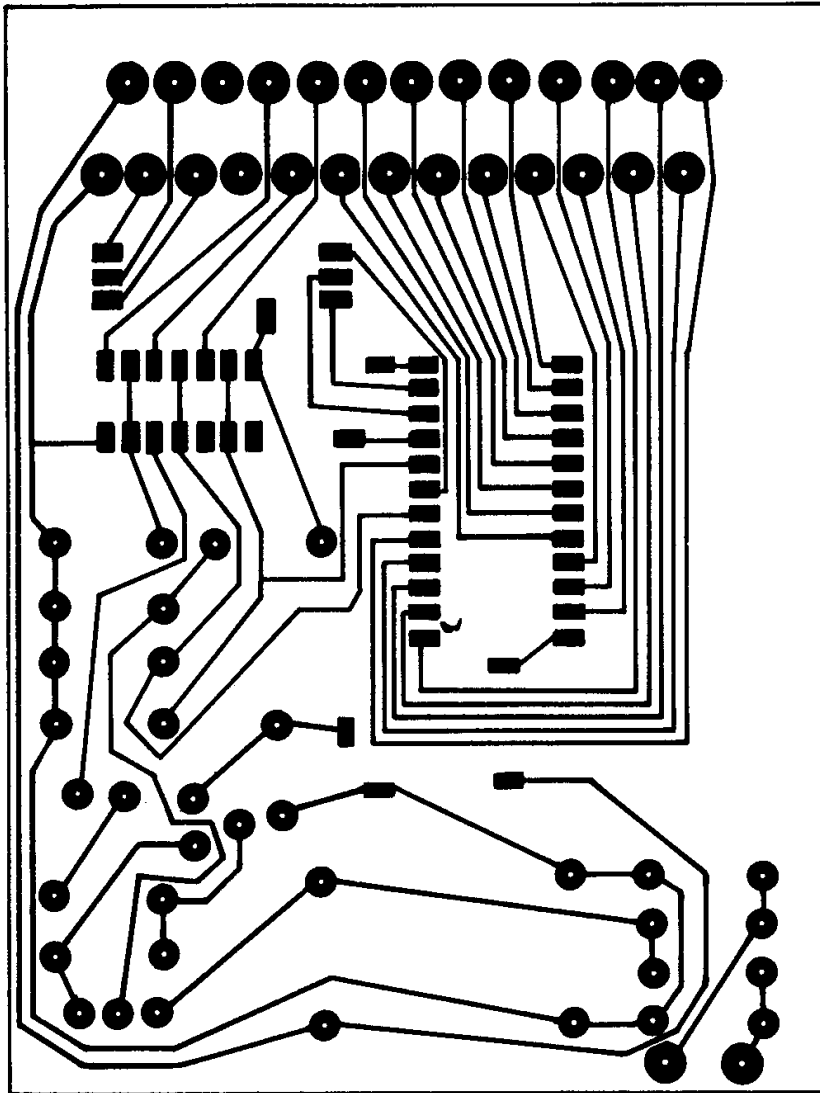
1130 B=USR3(B)
1140 GOTO 730
1150 REM *****
1160 REM * KOMMANDO PROGRAMMEER EPROM *
1170 REM *****
1180 LOCATE 0,23 : PRINT " 2 MINUTEN GEDULD"
1190 B=USR1(B)
1200 IF B=0 THEN 730
1210 LOCATE 0,23: PRINT" PROGRAMMEREN FOUT GEGAAN!"
1220 FOR I=1 TO 1000: NEXT I:GOTO 730
1230 REM *****
1240 REM * KOMMANDO KONTROLEER EPROM *
1250 REM *****
1260 B=USR2(B)
1270 IF B=0 THEN 1300
1280 LOCATE 0,23: PRINT" EPROM inhoud ongelijk buffer"
1290 : GOTO 1310
1300 LOCATE 0,23: PRINT" EPROM inhoud gelijk buffer"
1310 FOR I=1 TO 1000: NEXT I: GOTO 730
1320 REM *****
1330 REM *      KOMMANDO VUL BUFFER      *
1340 REM *****
1350 INPUT" met: ";H$
1360 B=VAL("&H"+H$)
1370 B=USR4(B)
1380 GOTO 730

```

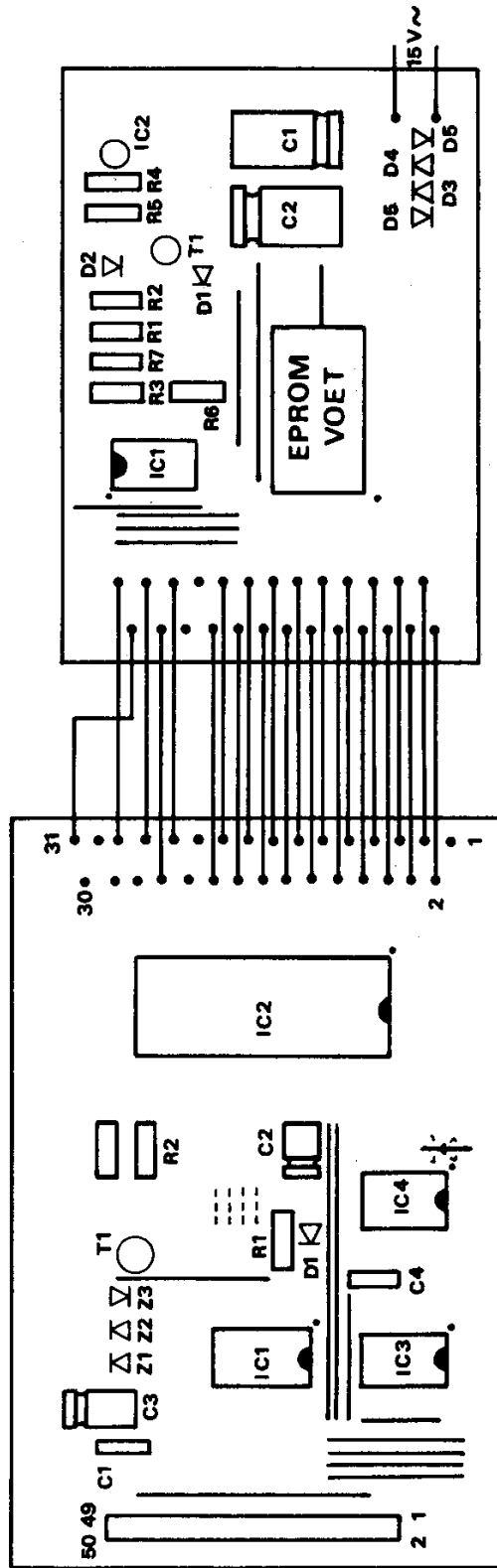
8.3 De programmeerschakeling

Samen met deze schakeling en het programma kunnen de EPROMs geprogrammeerd worden. De IO print doet het meeste werk voor ons, zodat de extra benodigde "hardware" klein is. Op de programmeerprint is een extra voeding aanwezig in verband met de hoge programmeerspanning (25 V). De stuurlijnen voor het programmeren zijn gebufferd met een IC om de hoge spanningen bij de computer weg te houden.

De opbouw is zoals bij alle schakelingen. Eerst de pennen en tot slot de componenten (zie voor plaatsing onderstaande tekening). Daarna wordt de print aangesloten en kan na het intikken van het program-



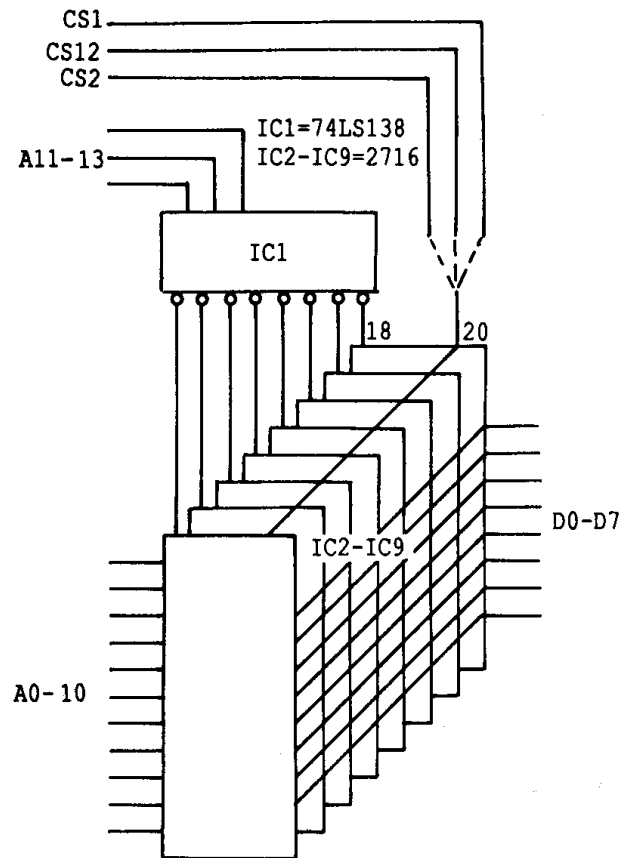
print 9



ma begonnen worden met programmeren.

8.4 *Uitbreiding aan de computer*

Met al de programma's in EPROM, kunnen we deze nog niet RUN-nen. De EPROMs moeten eerst met de computer verbonden worden. Dit gebeurt met onderstaand schema. Hiermee kunnen acht EPROMs van het type 2716 op de computer aangesloten worden. Met de draadbrug kan voor drie verschillende adresbereiken gekozen worden.



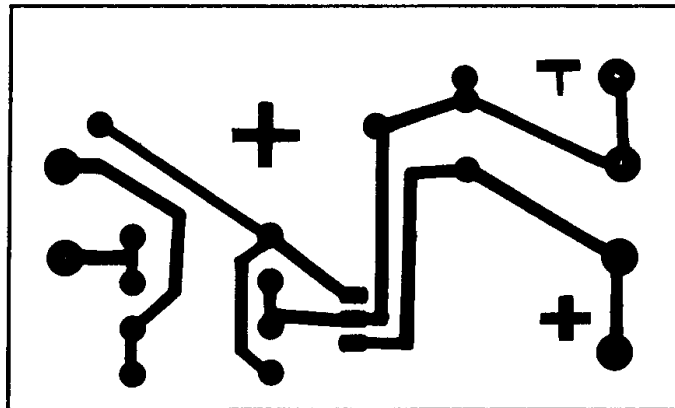
9. Extra voedingen

9.1 Algemeen

Bij alle in dit boek gebruikte schakelingen wordt de computervoeding gebruikt als voeding voor de schakelingen. De voeding van de computer heeft echter een beperkte capaciteit. Bij gebruik van een groot aantal extra schakelingen zal de voeding te klein zijn. Dit blijkt als de computer niet meer wil starten of fouten gaat maken.

Om hier een einde aan te maken kunnen we zelf voedingen maken die op het stopcontact worden aangesloten. Ze zijn gemakkelijk van een klein aantal componenten te maken. Door deze voedingen te gebruiken hoeven we de computervoeding helemaal niet meer te gebruiken. Iedere voeding levert een vaste spanning. Door andere IC's te kopen kan de spanning veranderd worden. De voedingen kunnen zowel positieve als negatieve spanningen leveren, en worden alle met de 0 aan de computer (IO print) verbonden.

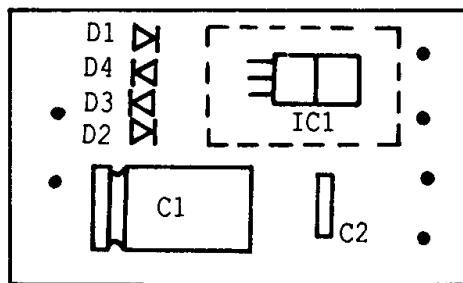
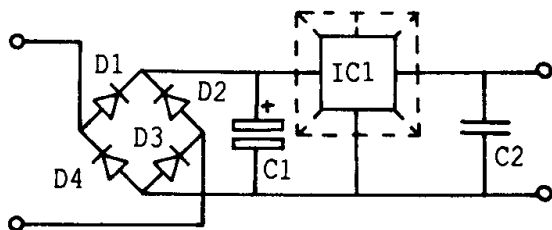
9.2 Positieve voeding



Met deze print kan zowel de spanning van 5 V als die van 12 V gemaakt worden. De 9 V voeding voor de 220 V interface kan ook met deze schakeling gemaakt worden. De voeding heeft als extra (en niet op de print aanwezig) een transformator nodig die 1 ampère kan leveren. De afgegeven spanning van deze transformator moet gelijk zijn aan de spanning die de voeding moet leveren.

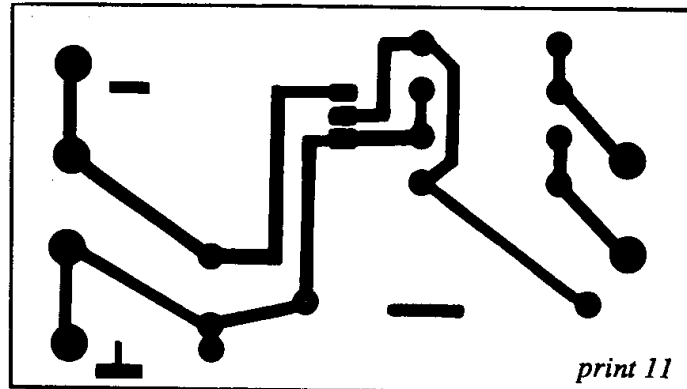
De spanning van de transformator wordt door de vier diodes (D1 - D4) gelijkgericht en door de condensator afgevlakt. Het IC 78XX zorgt er voor dat de spanning afgevlakt en kortsluitvast wordt. De 78XX bepaalt tevens welke spanning door de schakeling wordt afgegeven. Bij 5 V moet XX gelijk zijn aan 05, terwijl bij 12 V de XX gelijk moet zijn aan 12.

Om de warmte die dit IC opwekt af te voeren, wordt het IC op een koelplaat gemonteerd. De condensator zorgt er tenslotte voor dat kleine storingen weggewerkt worden.



D1=1N4001
D2=1N4001
D3=1N4001
D4=1N4001
C1=2200 μ F
C2=100nF
IC1=78XX (zie tekst)

9.3 Negatieve voeding

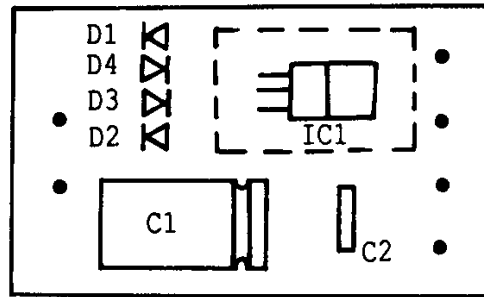
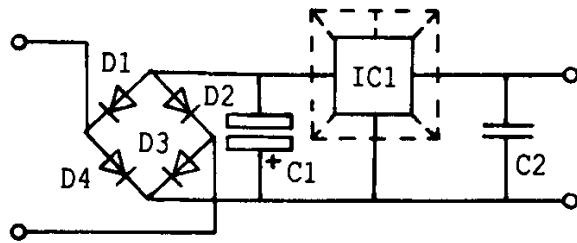


Met deze print kan zowel de spanning van 12 V gemaakt worden.

De voeding heeft als extra (en niet op de print aanwezig) een transformator nodig die 1 ampère kan leveren. De afgegeven spanning van deze transformator moet gelijk zijn aan de spanning die de voeding moet leveren. De spanning van de transformator wordt door de vier diodes (D1 - D4) gelijkgericht en door de condensator afgevlakt. Het IC 79XX zorgt ervoor dat de spanning afgevlakt en kortsluitvast wordt. De 79XX bepaalt tevens welke spanning door de schakeling wordt afgegeven. Bij 12 V moet de XX gelijk zijn aan 12.

Om de warmte, die dit IC levert, af te voeren, wordt het IC op een koelplaat gemonteerd.

De condensator zorgt er tenslotte voor, dat kleine storingen weggevoerd worden.



D1=1N4001
 D2=1N4001
 D3=1N4001
 D4=1N4001
 C1=2200 μ F
 C2=100nF
 IC1=79XX (zie tekst)

Verklarende woordenlijst

A/D converter

Elektronische schakeling die een analoog signaal omzet in een digitaal signaal.

adresbus

Elektronisch pad waarmee de Z80 aangeeft waar er gelezen of geschreven gaat worden.

analoog signaal

Signaal dat een oneindig groot aantal waarden kan aannemen.

CPU (centrale verwerkingseenheid)

De Z80 in de MSX computer. Digitale schakeling die via opdrachten, gegeven via de databus, in staat is om berekeningen en handelingen uit te voeren.

controlbus (stuurbus)

Elektrisch pad waarmee de Z80 aangeeft wat er met het geheugen of input/output-poorten gedaan gaat worden.

CS (chipselect)

Digitaal signaal waarmee aan digitale schakelingen de toegang tot de databus wordt vrijgegeven.

D/A converter

Elektronische schakeling die meerdere digitale signalen omzet in een analoog signaal.

databus (gegevensbus)

Elektrisch pad waarlangs de CPU (centrale verwerkingseenheid) gegevens verplaatst.

digitaal signaal

Signaal dat twee verschillende waarden kan aannemen, dat wil zeggen dat de waarde gelijk aan 0 of gelijk aan 1 is.

EPRM (eraseble programmable read only memory)

Geheugen-bouwsteen dat door middel van ultraviolet licht gewist kan worden. Hierna kan het geheugen opnieuw met hoge spanningen geprogrammeerd worden. Bij het uitschakelen van de computer blijft het programma in het EPROM aanwezig.

IC (intergrated circuit)

Gekompliceerde elektronische schakeling die zich in een gesloten behuizing bevindt.

INT (interrupt aanvraag)

Signaal bedoeld voor de Z80, waarmee een lopend programma kan worden onderbroken om tijdelijk een ander kort programma uit te voeren.

IORQ (in/output request)

Signaal waarmee de Z80 aangeeft dat er een input of een output actie gaat volgen.

MREQ (geheugentoeegang request)

Signaal waarmee de Z80 aangeeft dat er en geheugenactie gaat volgen.

OPTO-coupler (lichtkoppeling)

IC dat twee elektrische signalen door middel van licht met elkaar verbindt.

RD (leesstobe)

Signaal waarmee de CPU, in de MSX computer de Z80 mikroprocessor, aangeeft dat er data (gegevens) uit het geheugen of een input-poort gelezen worden.

reset

Signaal dat de Z80 mikroprocessor in een gedefinieerde begin-toestand brengt.

stuurwoord

Opdracht om de 8255 processor in een bepaalde toestand te zetten.

WR (schrijfstroke)

Signaal waarmee de Z80 aangeeft dat er data (gegevens) naar het geheugen of een outputpoort gestuurd worden.

Met de schakelingen uit dit boekje kan de MSX computer reageren op signalen die niet van het toetsenbord of van de joystick komen, en tevens signalen sturen naar andere apparaten dan printer of monitor. Bijzonder interessante, maar ook nuttige toepassingen worden in dit boek gepresenteerd. Stelt u zich voor: een modeltrein wordt bestuurd vanuit uw computer, of u meet temperaturen met uw MSX. En een toepassing van een lichtsluisteller kan iedereen wel bedenken. Zo zijn er meerdere boeiende en leerzame thema's opgenomen in dit boek, waarin u de computer laat communiceren met de buitenwereld!

De benodigde schakelingen zijn, op een enkele na, allemaal op print gezet en de printtekeningen zijn achter in het boek keurig op een rij gezet. Als u het al niet weet, zal de elektronika-handelaar waar u de onderdelen koopt, graag uitleggen hoe u hiervan prints kunt produceren.

Naast de prints en de componenten heeft u voor het maken van de schakelingen uit dit boek een goede multimeter (voor het afregelen) en klein gereedschap nodig. Met de schakelingen kunt u zelf nog eindeloos verder experimenteren.

Een greep uit de behandelde onderwerpen:

- Tekenen met twee PDL's
- IO poort met het 8255 IC
- Parallel output/input
- Telefoon-interface
- Lichtsluisteller
- Looplicht van 220 V lampen
- Analooq/digitaal omzetter en omgekeerd
- EPROMs programmeren
- Temperatuurmeting
- Modeltreinbesturing
- Extra positieve/negatieve voedingen

ISBN 90 6398 740 4

uitgeverij Stark-Textel b.v.