

# SIMSCAN

oftewel een verslag hoe een printer leerde lezen.

MSX CLUB MAGAZINE 24

Fred Wezenaar

*Scanned, ocr'ed, and converted to PDF by HansO, 2001*

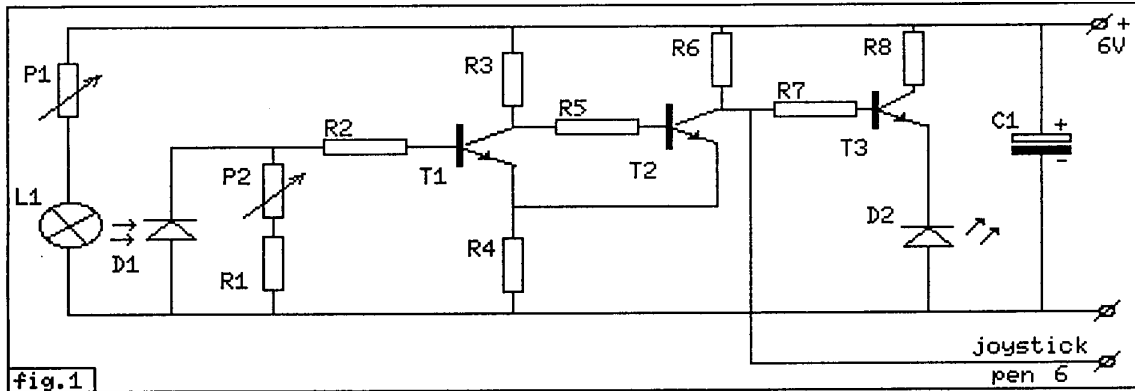
Het begon allemaal zo onschuldig... "Fred wat is een fotoscanner?" Toevallig had ik zo'n ding wel eens gezien : een prent op een wals, deze draait rond en de tekening wordt lijn voor lijn uitgelezen. De signalen van het "elektrieke oog" kunnen overal ingestopt worden: zetmachines, zenders, computers, telefoonlijnen, etc. Typisch voorbeeld van een parallel naar serie-omzetting!

"Te gek, zoiets zouden wij ook moeten hebben!" Zeker, zeker, maar voor die prijs eet ik een jaar boterhammen. Tja... "Zou zo iets niet zelf te maken zijn?" (verwijzing naar mijn knutseldrift.) Het leek mij niet zo simpel, de mechanische konstruktie is me te lastig. "Kun je daar dan geen printer voor gebruiken?" Nee oen, die is gemaakt om te schrijven, lezen is een ander verhaal!

Thuis gekomen keek ik mijn printer (een HR-5 van Brother) ineens met hele andere ogen aan. Het zou toch moeten kunnen..."maar ja dat 'busy'-signaal.en die linefeed...", malen dus. Het idee liet me niet los, dus de volgende dag met veel sterke koffie en een schrijfblok achter de toetsen van de getrouwe Goldstar.

Op de printkop een leeskop monteren bestaande uit een lampje en een fotodiode. Verlicht het lampje een donker deel van de prent dan valt er weinig licht op de diode en geeft deze een lage spanning af (een 0), omgekeerd wit of lichtgrijs weerkaatst veel licht dus dan een hoge spanning (een 1). Nee, dat was een probleem van de tweede orde. Belangrijker was: hoe krijg ik die printer zo gek dat hij als maar heen en weer langs de prent gaat? Lprint spc(80)? of lprint string\$(" ",80)? Helaas lege regels printen,daar beginnen ze bij Brother niet aan... Dus werd het lprint string\$(" ",80), daarbij een regelteller en toen zag het oer-programma' er zo uit:

```
10 y=0
20 lprint string$(".",80)
30 y=y+1 :if y=192 then end
40 goto20
```

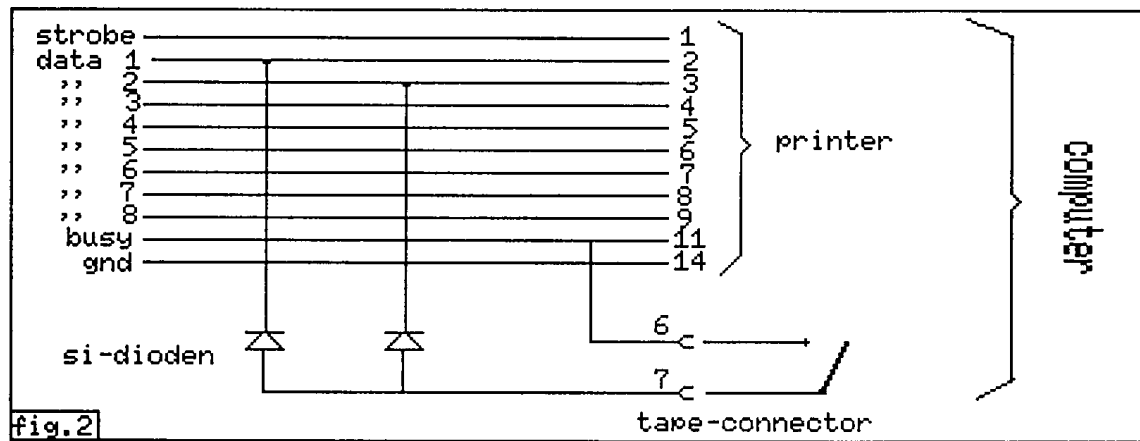


Mooi, dat werkte. Maar, ingevolge de wet van "Murphy" (ook wel "de wet tot behoud van ellende genoemd") dook een volgend en lastiger probleem op: zolang de printer printte, weigerde de computer enig bevel uit te voeren. Oorzaak bleek het 'Busy'-signaal te zijn dat de printer tijdens zijn werkzaamheden naar de computer stuurt. Deze wacht daardoor braaf totdat de printer klaar is met zijn taak.

Er moest dus, via een kunstgreep, een nep 'not-busy'-signaal gemaakt worden.

Ik probeerde dit eerst op te lossen via het programma (softwarematig dus) maar er bleek een elegantere en simpelere methode te zijn door het monteren van twee dioden in de printconnector. (fig. 2)

De Goldstar trapte er in en het volgende probleem kon te lijf gegaan worden.



### Het eigenlijke uitlezen.

Dat had ik mij als volgt voorgesteld: De enen en nullen van de leeskop worden via de joystickaansluiting aan de computer aangeboden. Is, op het moment dat het programma naar de joystickpoort "kijkt", deze 0, dan wordt op het grafische scherm de opdracht  $pset(x,y),1$  (zwarte punt) uitgevoerd en, omgekeerd, bij een "1",  $pset(x,y),15$  (witte punt). En dat uiteraard 256 maal per regel. Tot op dat moment zag het programma er ongeveer zo uit:

```

10 screen2: y=0: x=0
20 lprint string$(".",80)
30 if strig(1) then pset (x,y),1 else pset(x,y),15
40 x=x+1: if x>255 then x=0: y=y+1: if y>191 then 50 else goto20
50 goto 50
60 end

```

Dat betekende dus dat het programma 256 maal per (print)regel naar de JS-poort moest kijken. En helaas, de pset-opdracht is dermate traag dat in de 2,5 seconden die de Brother nodig heeft om 1 regel te printen slechts 60 pset-opdrachten konden worden uitgevoerd. Wat nu?

Machinetaal sprak ik nog niet, dus moest er binnen basic een andere oplossing gevonden worden. Het bleek dat de 'poke'- opdracht iets vlugger was. Daarmee zette ik de data van de leeskop tijdelijk in een stukje gereserveerd geheugen. Pas als alle nullen en enen gelezen waren werden ze middels onderstaand routinetje op het scherm gezet.

```

100 for a= adres to adres+191
110 if peek(a) then pset(x,y),1 else pset(x,y),15
120 next a

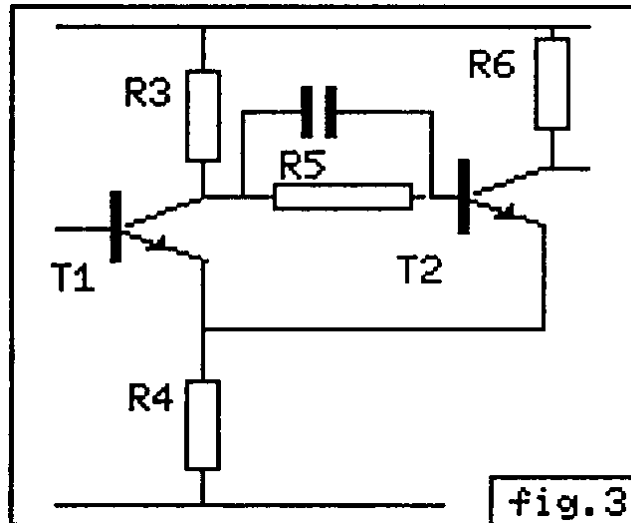
```

Een laatste moeilijkheid bleek de "timing" (zo je wilt: de synchronisatie) te zijn. Als het basicprogramma eenmaal gestart was dan rende het alsmaar door. Of de printkop op de juiste plaats stond of niet, na de print-opdracht werd meteen met het uitlezen begonnen. Vandaar dat regel 460 er in gekomen is. Na de printopdracht wacht deze regel op het eerste "witsignaal". Als we er nu voor zorgen dat er aan de linkerkant van de uit te lezen prent een witte band van ongeveer 5 mm. breed is, dan start het programma daarop en is een redelijke gelijkloop tussen de regels verkregen.

Uiteindelijk is dat de basis gebleven voor het definitieve programma. De wijzigingen daarna waren meer stroomlijningen, zoals het toevoegen van de 'motor on' en 'motor off' opdrachten. De printkabel met zijn extra dioden bleek onbruikbaar te zijn voor het normale printwerk. En aangezien "ons zuinig is..." wou ik geen nieuwe (andere) printerkabel maken, dus met de "motor"-opdrachten schakelde ik de dioden.

## **De bouw.**

Het uitgangspunt van deze scanner was, dat het te maken moest zijn met de spullen die elke rechtgeaarde knutselaar ergens heeft liggen, dan wel bij een standaard electronica winkel te koop zijn. De leeskop is gemaakt van een klein stukje montaprint van 4 banen breed. De lintkabel kan vast gemonteerd worden of via een (zelfgemaakte) miniconnector. De voorversterker is weinig kritisch en kan gevoed worden uit de 6 volts adapter van de Brother.



Voor het schakelen van de printerkabel nam ik een cassetterecorderkabel waarvan alleen de "remote"-plug gebruikt wordt.

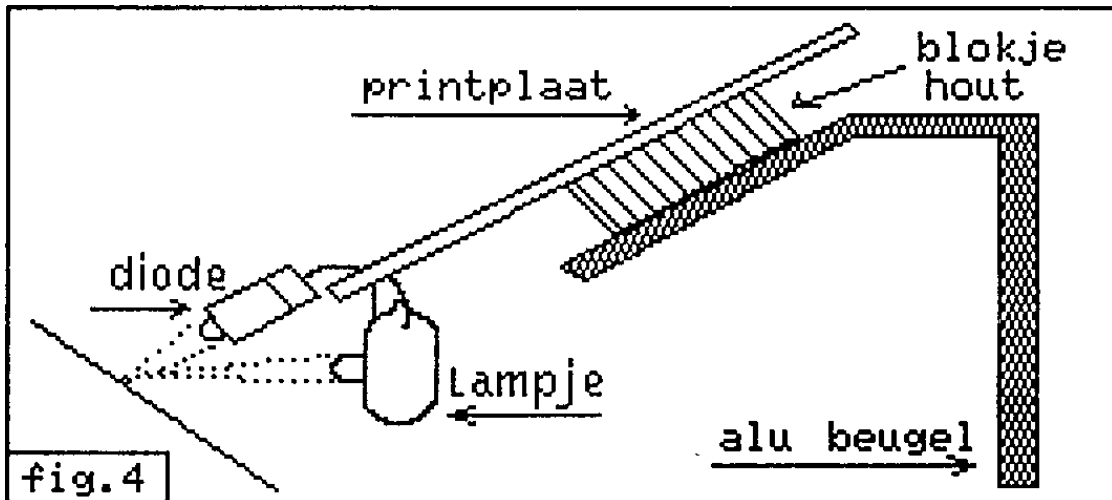
Wel kritisch is de plaats van de leeskop t.o.v. de prent. Een millimeter verschil in afstand kan een wereld van verschil uitmaken. Hoewel de derde transistor en de bijbehorende Led voor het scannen niet strikt nodig zijn, is het een welkome hulp bij het afregelen. Als dat moment aangebroken is, kan dat als volgt: duwdeprint/leeskopnaarvoren tot hij vastklikt, als alles goed gegaan is werpt het lampje een mooi rond lichtvlekje op het papier (neem hier als test een maagdelijk schoon vel voor). Met behulp van een pincet (en, indien nodig een leesbril...) wordt de lichtdiode zodanig verbogen dat hij midden in het lichtvlekje "kijkt". Als de gevoeligheidsregelaar (P2) maximaal (d.i. rechtsom) staat en die lampspanning op ongeveer 3/4, dan moet de Led oplichten. Draai nu aan de gevoeligheidsregelaar tot de Led net uitgaat, probeer door de afstand van de leeskop te variëren of de Led weer aan te krijgen is. Herhaal dit net zo lang totdat de grootste gevoeligheid bereikt is. Vergeet niet de hoek waaronder de fotodiode staat aan te passen met de veranderde afstand leeskop/papier.

### Over het lampje:

Terwille van de compactheid nam ik een fittingloos type, dan is het richten ook wat makkelijker. Deze kon ik echter niet met ingebouwde lens krijgen, die heb ik van een ander lampje gesloopt en met kit op het treinlampje (12 volt) gelijkmd. Of het lensje strikt nodig is betwijfel ik nu een beetje, vermoedelijk heeft de fototransistor een scherp gerichte bundel.

Soms wil het gebeuren dat de scanner een "rafelig" beeld geeft, hoe dat komt weet ik niet maar een condensator van 1000pF over R5 hielp...

Zwart schilderen van lampje en fototransistor heeft wel zin. Het hele gebeuren is zeer gevoelig voor vals licht, zodat een afschermkapje van zwart papier noodzakelijk bleek.



Veel knutselplezier en... scan ze.!

Materiaallijst bij tekening 1:

- D1.... fotodiode BPX 81
- D2.... LED (kleur naar keuze...)
- P1.... Potentiometer 30 ohm, draadgewonden
- P2.... potentiometer 10 Kohm lineair
- RL... weerstand 2,2 Kohm
- R2.... " 10 Kohm
- R3.... " 10 Kohm
- R4.... " 47 ohm
- R5.... " 10 Kohm
- R6.... " 2,2 Kohm
- R7.... " 10 Kohm
- R8.... " 220 ohm
- CL... Elco1000uF/16volt
- LL... Treinlampje 12v/0,1 A
- T1 -3.. Transistor BC 547b

```

10 ' SCANNER-PROGRAMMA.
20 '-----
30 'Fred Wezenaar, november 1986.
40 '
50 ' initialisatie*****
60 F=DSKF(0):IFF<30THENCLS:LOCATE0,10:PRINT"Te weinig disk-ruimte!!!":END
70 CLEAR100,&H8FFF
80 ONSTOPGOSUB640:STOPON
90 COLOR1,5,5:SCREEN2:DEFINTA-
Z:DEFUSR0=&HA5:DEFUSR1=342:DEFUSR2=&H9000:P=&H9000:R=-1
100 FORA1=0TO6:READA2:POKEP+A1,A2:NEXTA1
110 DATA205,168,0,50,248,247,201
120 OPEN"GRP:"AS1
130 ' INTROSCHEM.*****
140 LINE(6,2)-(250,10),15,BF
150 PRESET(10,3),15:PRINT#1,"***** SCANNER-PROGRAMMA. *****"
160 LINE(6,78)-(249,98),10,BF
170 IFUSR2(0)=0THENPRESET(48,85):PRINT#1,"PRINTER NIET AAN ! !":BEEP:GOTO170
180 LINE(6,78)-(249,98),10,BF
190 LPRINTCHR$(27);"A";CHR$(1);
200 LPRINTCHR$(27);"U";CHR$(1);
210 PRESET(8,80):PRINT#1,"Onder welke naam moet
het":PRESET(8,90):PRINT#1,"beeld weggeschreven worden?"
220 LINE(38,118)-(112,128),15,BF
230 PRESET(40,120),15:PRINT#1,N$+CHR$(&HDB)
240 A$=INKEY$:IFA$=""THEN240
250 IFA$=CHR$(13)THEN310
260 IFA$=CHR$(8)ANDLEN(N$)>0THENN$=LEFT$(N$,LEN(N$)-1):GOTO220
270 N$=N$+A$
280 LINE(38,118)-(112,128),15,BF
290 PRESET(40,120),15:PRINT#1,N$+CHR$(&HDB)
300 A$="":GOTO240
310 N$=LEFT$(N$,8)
320 LINE(38,118)-(112,128),5,BF
330 COLOR1,14,5 :CLS:MOTORON
340 LINE(0,0)-(47,191),11,BF:PRESET(2,18),11:PRINT#1,"R.nr:"
350 PRESET(2,80),11:PRINT#1,"NAAM:"
360 FORD=1TOLEN(N$)
370 PRESET(D*6-5,96),11:PRINT#1,MID$(N$,D,1)
380 NEXTD
400 ' regelnummer ophogen*****
410 R=R+1:K=0
420 IFR=192THEN610
430 'dummy-regel printen.*****
440
FORW=1TO76:OUT&H91,44:OUT&H90,0:A=USR(0):NEXTW:OUT&H91,10:OUT&H90,0:A=USR(0)
450 ' poken.*****
460 IFSTRIG(1)=-1THEN460
470 IFSTRIG(1)=-1THENPOKEP+K,1
480 IFSTRIG(1)=0THENPOKEP+K,0
490 K=K+1:IFK=216THEN510
500 GOTO470
510 LINE(20,28)-(47,38),11,BF:PRESET(16,30),11:PRINT#1,USING"###";R
520 GOSUB540
530 GOTO430
540 'op scherm printen.*****
550 X=46
560 FORA=&H9000TO&H90D7
570 IFPEEK(A)=1THENPSET(X,Y),1
580 X=X+1:POKEA,0:NEXTA
590 Y=Y+1:P=&H9000
600 RETURN410

```

```
610 'van scherm naar schijf.*****
620 BEEP:BEEP
630 LINE(0,0)-(47,40),11,BF
640 ONERRORGOTO620
650 BSAVEN$,0,&H3FFF,S
660 MOTOROFF:ONERRORGOTO0
670 ' programma beindigen.*****
680 MOTOROFF:END
```