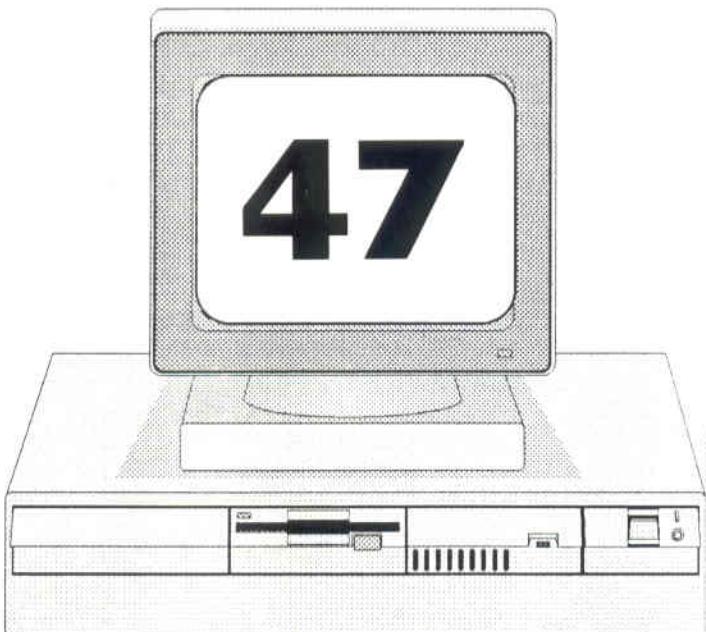
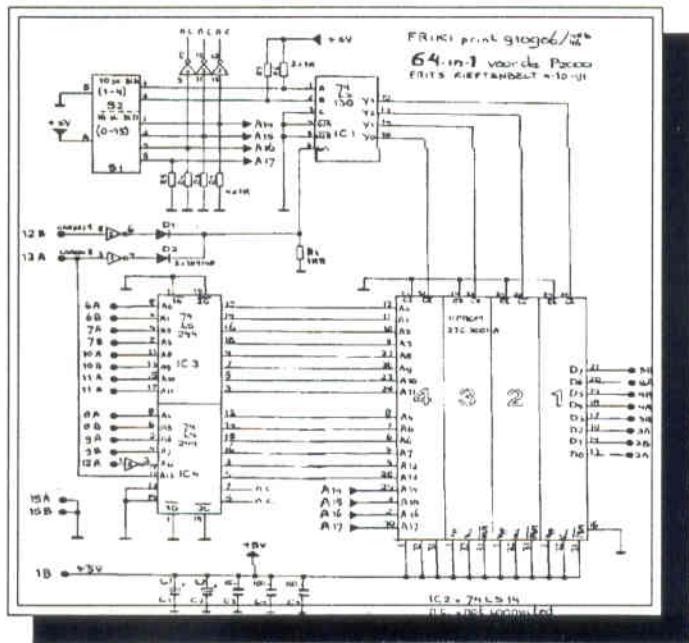


tron



o.a. in dit nummer:

¼ é è f
ö â è ¾

Heb je dat
ook wel
eens? (2)

**EMM386
en Extra
Upper
memory**

PIEP!
Videotext op de
kabel in Den Haag
Nieuwe operating
systems

**GGPC
TRON**





is het officiële orgaan van de stichting GGPC

COLOFON

Uitgever: Stichting GGPC

Redactie adres: Postbus 7268 2701 AG Zoetermeer

Database: TRON-VIEWTEKST 079 - 310.166
(24 uur per dag, 7 dagen per week multi speed)

Hoofdredacteur	Albert C. Veldhuis 079 - 316.915
Hoofdredacteur a.i.	Jeroen Wortelboer
Eindredacteur	Jo C. Garnier
Revisor	Wilfred Korrelboom
Vormgeving	Robert Vroegop
Omslag	Robert Vroegop en Jeroen Wortelboer
P2 + PC - publicist	Roeland van Zeijst
Algemene Zaken	Jannie Aalderink-Bosveld
Druk	CONTEXT Rijswijk

Medewerkers aan dit blad:

ap veldhuis, Andries Hofstra,
Peter Greve, Hans Kusters,
Guido Klemans, Roeland van Zeijst,
Peter de groot, Guido Klemans,
Karin, Andor Vierbergen, Albert Bee-
vendorp, Robert Vroegop, Jeroen
Hoppenbrouwers en Frits Kieftenhelt.

Advertentietarief: OP AANVRAAG.

Copyright (c)

De inhoud van dit blad mag niet gereproduceerd worden in welke vorm dan ook, zonder voorafgaande schriftelijke toestemming van de uitgever. De aansprakelijkheid uit hoofde van auteursrechten van ingezonden kopij ligt bij de inzender.

Abonnementen:

Deelnemers van de **GGPC** krijgen het blad gratis toegestuurd.

Doelstelling:

De Stichting GGPC stelt zich ten doel het gebruik van computers in de ruimste zin te bevorderen. Deelname aan de Stichting wordt aangegaan voor ten minste één (1) kalenderjaar en geldt tot schriftelijke wederopzegging. Het deelnemerschap bedraagt f 45,00 per jaar, voor individuele personen, bedrijven en instellingen met toezending van 1 gratis TRON. Alleen een abonnement op TRON kost f 30,00 per zes (6) nummers. Opgave voor het deelnemerschap dienen te worden gericht aan:

Het secretariaat van de Stichting GGPC, Wielingenplein 17, 3522 PC Utrecht. Telefoon: 030 - 881.087

Betaalwijze: Binnen 14 dagen na aanmelding of direct op gironummer: 240.800 t.n.v. Penningmeester Stichting GGPC te Utrecht.

INHOUD

Redactieel	3
Nieuws uit de Afdelingen	
Noord, Den Haag	4
P2000	
Heb je dat ook wel eens?	17
De 64-in-1-dos	22
MSX	
MSX-Spotlight (2)	25
PC	
Karakters & Accenten	
Teleac cursussen	
Turbo Pascal Routines (8)	
Een enkeltje Pascal alstublieft (18)	8
Flexibele Produktie Automatisering (6)	15
EMM386 en Extra Upper Memory	16
Siepiejoes	18
Digital Dancing	19
Nieuwe operating systems	19
GGPC-Net	
Piep!	7
Algemeen	
Videotext nu ook in Den haag op de kabel	5
Beursen	
Computer Show te Den Bosch	18
Boekbesprekingen	
Handboek voor 80386 programmeurs	28
De DOS expert t/m MS-DOS 5 en DrDOS 6	28
Geheugenbeheer met MSDOS 5	29
Werken met Turbo Pascal 6.0	30
Using Turbo Pascal 6.0	30
Boeken over Quattro Pro 4.0	31
Boeken over Coreldraw 2.0	33



tron

is het officiële orgaan van de stichting GGPC



Redactieel

Aan FLAARDEN

De zomervakantie zit er weer op en ‘het extra dikke zomernummer’ van TRON heeft u natuurlijk weer aan flarden gelezen!

Plaatjes en plaatjesmakers

Als ik het over het zomernummer heb, dan denk ik toch steeds, met een beetje weemoed terug aan de tijd dat wij daarop, net als op het kerstnummer, nog “eigen gemaakte” plaatjes zetten.

Onze “Plaatjesmakers” uit die dagen, als **Ronald Zeelenberg**, **Henry Aalderink**, de gebroeders **S&S**, en **QMO** wisten voor die gelegenheden toch verbluffende resultaten te bereiken op hun P2000-tje! Sla er de oude nummers nog maar eens op na. De drukkwaliteit was in die tijd niet om over naar huis te schrijven, omdat het vele jaren een kwestie van plakken geweest is, van artikelen die op matrixprinters van uiteenlopende kwaliteit werden uitgetikt. (*Vandaar dat wij onze layout man toen ook ‘plakker’ noemden*).

Maar kijk nu toch voor de aardigheid eens naar die fraaie plaatjes van KUIFJE (TRON 10), de Kerstkaars (TRON 12), het PALMENSTRAND (TRON 16) en de PAKJESAVOND (TRON 18) van RoZee!. Toen kwamen de werken van S&S en QMO: PALMEILAND (TRON 21), het logo van databank VREDENBORCH (TRON 26) en tenslotte het ZEILSCHIP BIJ ONDERGAANDE ZON (TRON 34, van augustus 1990).

Daarna bleven S&S en QMO nog wel actief in het maken van plaatjes en logo’s voor databanken en kerstwensen op TROS Teletekst pagina, maar deze verschenen niet meer in de immiddels op de PC bewerkte- en op de laserprinter geprinte TRON. De reden: de kwaliteit van een, zwart-wit gedrukt, “teletekst-plaatje” valt in het niet bij een hoogwaardig grafisch plaatje dat ergens uit een of ander plaatjes-programma kan worden gehaald. Toch jammer!

Achterstand ingehaald

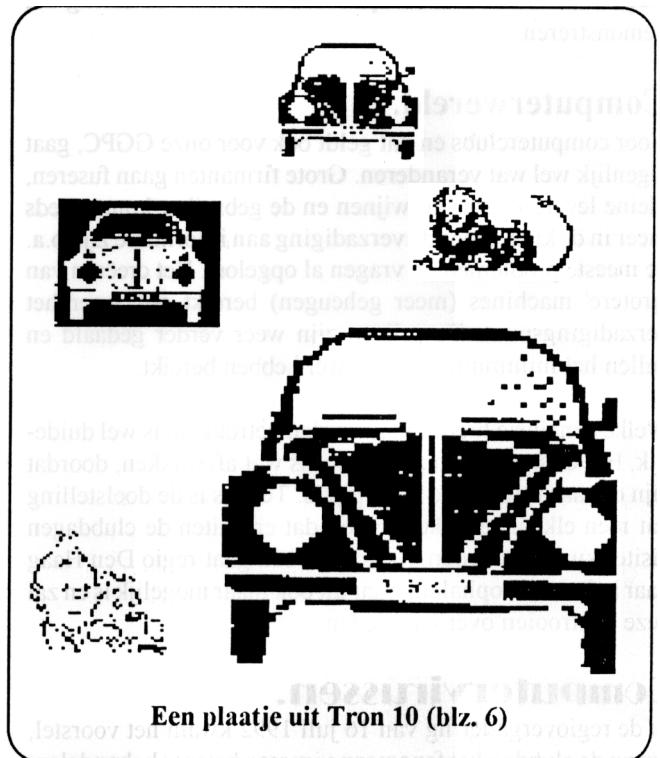
Twee nummers lang is TRON veeeeél te laat in uw brievenbus terechtgekomen. Wat blijkt het moeilijk te zijn om zo’n achterstand weer in te halen! Het gaat er niet alleen om de kopij tijdig binnen te krijgen, maar de volgende stappen moeten dan ook maar weer precies aansluitend gezet kunnen

worden (door de amateurs die wij tenslotte toch zijn en in de vrije tijd die wij daarvoor ter beschikking stellen).

U kunt zich voorstellen dat het in de zomervakantie wel heel erg makkelijk is om in de fout te gaan. En dat gebeurde dan ook!

Maar enfin, het is ons dan nu toch gelukt om u weer op tijd, vele interessante artikelen te presenteren. Ook u heeft daar door iets **MEER TIJD** gekregen **OM UW BIJDRAGE** te leveren. Doen!

ap veldhuis



Een plaatje uit Tron 10 (blz. 6)

UITERSTE DATUM VOOR HET INZENDEN VAN UW KOPIJ VOOR TRON 48:
zaterdag 31 oktober 1992

Turbo Pascal Routine's deel 8

door Guido Klemans

Om gegevens over de Virtual DMA Specification (VDS) op te vragen, moet het AX register worden geladen met de hexadecimale waarde 8102, het DX register met 0 en moet de CARRY-FLAG worden gezet. Als de VDS geïnstalleerd is, zal de CARRY-FLAG zijn gewist en bevatten de registers de gevraagde informatie. De VDS is een IBM/Microsoft lapmiddel om problemen met DMA in protected mode te omzeilen en kan alleen op systemen met een 80386 of 80486 processor worden gebruikt. EMM386, QEMM386 en dergelijke installeren de VDS.

Wat heeft dit met Turbo Pascal te maken?

Niks eigenlijk, ware het niet dat via de in de DOS unit aanwezige register variabelen en de INTR-instructie, deze gegevens zijn te achterhalen, indien men daarin is geïnteresseerd. En dat was ik. Ik keek dan ook vreemd op, toen mijn programma mij meldde dat op een 80286 systeem de VDS geïnstalleerd was. De bijbehorende gegevens sloegen natuurlijk nergens op!

PIEP!

een rubriek over datacommunicatie
door Roeland van Zeijst

Datacommunicatie-land is groot, maar tegelijk ook PIEP!-klein. Dat merk je in de vakanties! Totaal geen nieuws van het datafront dus... Nou ja, geeft ook niet, want afwisseling moet er zijn en een beetje minder dan in TRON 46 kan ook weinig kwaad. Verder heb ik de afgelopen nummers het één en ander aan "zware" (?) kost behandeld. Ik krijg de indruk dat de meeste lezers dat erg op prijs stellen, maar ik brul ook nu weer: geef eens een leuke suggestie voor de volgende TRON(nen)!

Nog een paar hele kleine feitjes dan:

De GGPC heeft in PTC-Net weer haar postbus POST GGPC#. Op 5 september werd TV-Videotex definitief opengesteld. De GGPC-dienst in ESCAPE en op VideotexNet (06-7400 ESCAPE#, 37.5 c/min) wordt steeds leuker!

Inmiddels zijn PIEP! en Virus Journaal in VIP-Net ook geheel opengesteld voor publiek (055-226116, normaal tarief).

Tot slot vindt u natuurlijk meer datacommunicatie-informatie in Hoppie's Net-werk elders in deze TRON (hoop ik).

In september, oktober zal "het wereldje" weer op gang komen, dus in de volgende TRON ben ik weer bij u terug, met weer een echte ouderwetse lading PIEP!-informatie.

RovaZ

STOP DE PERSEN!

Of, ach nee, laat ook maar doorrollen eigenlijk.

Toen ik geen fout in mijn programma kon vinden en in DEBUG op hetzelfde systeem geen VDS kon vinden; CARRY nog steeds gezet, stond ik voor een raadsel.

Ergens werd de CARRY gewist. Niet door de VDS, want die was er echt niet en niet door mijzelf, waarom zou ik. Zoals Sherlock Holmes altijd zei: **als alle andere mogelijkheden zijn geëlimineerd, wat dan nog overblijft, hoe onwaarschijnlijk ook, moet de waarheid zijn;** Turbo Pascal zat aan de CARRY!

Als voorbeeld het volgende deelprogramma:

```
var regs : registers;
rfs : word;
{maak een procedure aan, die via een interrupt kan worden
aangeroepen; de procedure doet niets, zeker niet met CARRY}
{$F+}
procedure int60; interrupt;
begin
asm
nop
end;
end;
{$F-}

begin
clrscr;
SetIntVec($60,Addr(int60));
regs.flags := 1;
intr($60,regs);
writeln(bin(regs.flags));
asm
stc
int $60
pushf
pop ax
mov rfs,ax
end;
writeln(bin(rfs));
end.
```

De interrupt-procedure wordt aan INT 60h verbonden, de flags-variabele wordt op 1 gezet: CARRY is bit 0 van het flags-register. De interrupt wordt aangeroepen en de flags worden geprint.

Daarna hetzelfde met de inline-assembler: zet CARRY, voer de interrupt uit en print de flags.

Dit zou hetzelfde resultaat moeten opleveren, tenminste voor CARRY. Echter, uit de eerste aanroep komt een GEWISTE CARRY, de tweede keer is CARRY GEZET, zoals het zou moeten zijn.

Is dit een bug in Turbo Pascal 6.0?????

Het is zeker niet zoals het zou moeten....

Een Enkeltje Pascal alstublieft (18)

Een cursus Turbo Pascal

't Is weer voorbij...

Die mooie zomer... En voor zover ik nu (augustus) kan bepalen zal datzelfde in oktober ook wel gelden voor het publieke omroep-bestel. Maar niet getreurd, want natuurlijk zijn er nog altijd de diverse commerciële kanalen en dat leuke station dat u alleen kunt ontvangen op uw PC-monitor: Turbo Pascal! (Hee, station, ook als in "trein"... Je zou haast denken dat...)

't Is weer voorbij...

De beurzen-maand september... Met natuurlijk de in deze TRON besproken beurs **ComputerShow Den Bosch '92/GAMES '92** in Den Bosch en de nog in TRON 48 te bespreken **Benelux Computer '92** en de **Firato**. Helaas, ook beurzen hebben zich te houden aan de deadline van ons blad... Overigens: Komt allen naar beurzen waar de GGPC staat, want (en dat moet gezegd) de PromoTeam-leden uit het Noorden des lands brengen iedere keer weer een weergaloze P2000-verrassing mee!

't Is weer voorbij...

De Kleine Leuke Zomerpuzzel... Een oud liedje klinkt in mijn hoofd terwijl ik bovenstaande kop intik, nl. "Kleine Blonde Mariandel" (voor de jongelui: één van de vele voorlopers van "Poing!"), met Annie de Reuver geloof ik... (Oei, en jullie altijd maar denken dat ik echt 17 jaar was...).

Zometeen (in het Pascal-gedeelte) volgt de listing van de slimme oplossing, maar nu alvast **The Credits**: De allereerste persoon van wie een volledige en intelligente listing binnengewaagd was **Kees Verhoog** (21) uit Amsterdam. Kees is geen GGPC'er (Kees toch!), maar wel een goede programmeur!

't Is weer voorbij...

TRON 46... Met natuurlijk onze kersverse nieuwe MSX-medewerker **Albert Beevendorp**, die bij deze is uitgenodigd om eens wat MSX-igs te komen laten zien op een beurs! Leuke boeken, een mooie gok-wedstrijd, een boeiend P2000-artikel en niet te vergeten al die series die we maar al te vaak als vanzelfsprekend beschouwen, zoals die van Guido, Hoppie en Robert Vroegop (helaas de laatste). Lekker leesbaar en - volgens mij- nóg iets beter dan de TRONnen daarvoor (op die 12 pagina's RovaZ-junk na dan)!

En onze hoofdredacteur natuurlijk... Even optimistisch als altijd wijst hij ons erop dat het computerhobbyisme dreigt uit te sterven. Nou wil ik mijn artikeltjes altijd een beetje vrolijk houden, maar daar spreekt Ap toch wel een waar woord. Tijd dus voor de GGPC om zich te gaan bezinnen op een (iets)

door Roeland van Zeijst

andere invulling van haar club en haar taken, om ook de huidige computer-gebruikers blijvend van dienst te kunnen zijn.

't Is weer voorbij...

Deze inleiding!

Vandaag...

Zoals beloofd: Deoplossing van de Leuke Kleine Zomerpuzzel 1992, we gaan een TSR maken en uitvinden wat Compiler Directives zijn.

Leuke Kleine Zomerpuzzel 1992

Okee, De Oplossing: Geen ingewikkelde algoritmes, zoals beloofd, maar gewoon een simpel, flauw truukje: de datum, waarvan we de weekdag willen bepalen, stoppen de even in de interne klok van MS-DOS en vragen dan de datum weer op, waarbij MS-DOS automatisch de weekdag meegeeft! Zo simpel is het dus!

Onderstaand een "kadertje" van hoe het programma zou kunnen worden. Er zitten geen mooie tests in of de datum wel "kan", maandnamen worden niet bepaald en het programma stopt alleen met Ctrl-Break. Kortom: net niet helemaal afgeremd, maar het gaat om het idee en die afwerking is nou juist weer net iets voor... u?

```
PROGRAM ZomerPuzzel92;
USES Crt, Dos;
CONST DagNaam : Array[0..6] of string
      = ('zondag','maandag','dinsdag',
        'woensdag','donderdag',
        'vrijdag','zaterdag');
```

```
VAR Dag,
     Maand,
     Jaar,
     WeekDagNr,
     ReserveDag,
     ReserveMaand,
     ReserveJaar : Word;
```

```
Procedure VraagDatum;
Begin
  write(' Dagnummer: ');
  read(Dag);
  write('Maandnummer: ');
  read(Maand);
  write(' Jaarnummer: ');
  read(Jaar);
End;
```

```

Procedure BepaalWeekDag;
Begin
  { bewaar oude datum }
  GetDate(ReserveJaar,ReserveMaand,
    ReserveDag,WeekDagNr);
  { stop test-datum in klok }
  SetDate(Jaar,Maand,Dag);
  { haal weekdagnummer op }
  GetDate(Jaar,Maand,Dag,WeekDagNr);
  { herstel oude datum }
  SetDate(ReserveJaar,ReserveMaand,
    ReserveDag);
End;

Procedure ToonDatum;
Begin
  WriteLn;
  WriteLn('Bovenstaande datum ',
    'valt op een ',
    DagNaam[WeekDagNr],':');
  WriteLn;
End;

Begin
  ClrScr;
  repeat
    VraagDatum;
    BepaalWeekDag;
    ToonDatum;
  until false;
End.

```

Les 13C : Onze eigen TSR

Er zijn allerhande leuke truukjes mogelijk met TSR's (dat wist u al), maar het Enkeltje is helaas een snelle, doch smalle trein, zodat er elke twee maanden toch niet meer stof in kan dan noodzakelijk. Vandaar dat we een nogal saai voorbeeld gaan nemen voor onze TSR, namelijk een klein programmaatje dat toch best handig kan zijn.

Misschien bent u net als ik één van die weinigen die geen printer (meer) heeft, of u hebt hem bijna nooit aanstaan. En dan druk je prongluk op Shift-PrtSc... Koffie halen!

We zullen het even dom houden: we gaan géén mooie routine maken die veel sneller dan BIO BIOS test of er wel of niet een printer on-line is, maar gewoon een "dom" programmaatje dat bij Shift-PrtSc "PIEP!" zegt en daarna meteen teruggaat (dus niet uren blijft hangen). Simpel? Nou...

Op zich is het principe verrie iezie: we maken een interruptroutine, zorgen dat ons programma in het geheugen blijft staan als we het verlaten en zorgen zo dat elke interrupt-aanroep direct wordt doorgesluisd naar ons stukje programma dat nog gewoon in het geheugen staat!

Hoe houden we het programma vast?

Het moge duidelijk zijn dat het geheugen waarin een programma zich bevindt na afloop van dat programma weer wordt vrijgegeven. Dat doet DOS automatisch voor ons en dat is nou juist het probleem. Stel, ik heb op adres X een interruptroutine neergezet. Mijn programma (met daarin de interruptroutine) wordt verlaten, het geheugen wordt vrijgegeven, de gebruiker laadt een nieuw programma in en er volgt toevallig een interrupt. **HANGEN!** zegt de computer dan... (of erger)...

Dus niet. Wat dan? Het programma niet verlaten, zodat de TSR blijft staan? Nee, want dan is het geen **Terminate and Stay Resident** programma meer, omdat het niet zichzelf beëindigt en dan resident blijft, want het blijft gewoon op de voorgrond draaien...

Een aparte oplossing

Er is binnen MS-DOS een aparte functie die zorgt, dat het huidige programma wordt beëindigd, maar dat het gereserveerde geheugen niet automatisch wordt vrijgegeven. In Turbo Pascal heet die functie "**Keep**" met een parameter erachter (een word), die een Exit-code voor DOS aangeeft. Normaliter zeggen we **Keep(3)**, omdat 3 de DOS-exitcode is voor "correct beëindigd, maar resident gemaakt". **Keep(0)** mag natuurlijk ook, alleen zijn we dan niet eerlijk tegen de andere programma's, want die weten alleen wat ze via de exitcode te zien krijgen.

Let op!

Keep is niets anders dan een "**Halt met resident blijven**", oftewel: **Keep** beëindigt direct het programma! Dus altijd eerst alles klaarmaken en dan pas **Keep** aanroepen.

Geheugenruimte

Eigenlijk komt nu een stukje theorie aan bod wat we verderop bij de Compiler Directives weer tegenkomen; pak dat er dus eventueel alvast bij.

Het is namelijk zo, dat DOS niet "zomaar" weet hoeveel geheugenruimte ons programma nodig heeft. Die informatie staat opgeslagen in de .EXE-file.

Wat we aan Turbo Pascal kunnen doorgeven is hoe groot de stack (met de lokale variabelen erop en ook de adressen waarnaar teruggesprongen moet worden als een aangeroepen procedure of functie is beëindigd) maximaal wordt, hoeveel ruimte de dynamische variabelen (heap) samen minimaal zullen innemen en hoeveel ruimte ze maximaal kunnen innemen.

Bij elk normaal programma komt de stack niet boven de 16 Kilobyte uit. De minimale heap-waarde is doorgaans 0 (als er nog geen dynamische variabelen zijn geïnitialiseerd) en bij een normaal programma is de maximale heap-waarde meestal 655360, oftewel: al het vrije, direct adresseerbare geheugen zou eventueel gebruikt mogen worden.

Bovenstaande getallen leveren problemen op bij Keep, want de stack-grootte en de maximale heap-grootte moeten minimaal gemaakt worden. MS-DOS durft namelijk niet te tornen aan de door ons opgegeven waarden, dus kan het voorkomen dat er veel teveel geheugen wordt gereserveerd.

De maximale heap-grootte uitrekenen is gewoon een kwestie van "ruw" optellen en vermenigvuldigen, bijv. een Word = 2 bytes, dus 100 dynamische Words kosten 200 bytes geheugen).

Het uitrekenen van de stack-grootte kan via de "trial and error" methode: compileer het programma met `{$S+}` (zie ook verderop) en maak de stacksize bij elke compilatie wat kleiner, tot het programma op een gegeven moment tijdens het draaien afbreekt met een Stack Error. Dan is er te weinig stack-ruimte gereserveerd.

Om de waarden door te geven aan Turbo Pascal moet de volgende compiler directive (zie verder) worden neergezet, liefst onder het PROGRAM statement:

`{$M stack,heapmin,heapmax}`

en alle waarden worden uitgedrukt in bytes. Bij een programma zonder dynamische variabelen dus bijvoorbeeld:

`{$M 16384,0,0}`

Overigens wordt de `{$M}` directive volledig genegeerd in units, dus tellen ze alleen bij het hoofd-programma. Verder mogen de aantallen bytes ook worden weergegeven in hexadecimale notatie, dus bijv. `$01E3` oid. En let op: op de heap staan dus alleen dynamische variabelen (New, GetMem); de gewone (statische) lokale variabelen komen op de stack te staan en de gewone globale variabelen staan gewoon in het

programma en daarvoor is dus al door het inladen van het programma ruimte gereserveerd!

Goed, het programma zal dus op zijn geheugen-gebruik moeten letten, `Keep(3)` moeten gebruiken en de PrtSc-interrupt moeten omleiden. Een kleine blik in een willekeurige interrupt-listing (of weet u het uit uw hoofd?) leert ons, dat het om interrupt \$05 gaat. Overigens een beetje vreemde notatie, want \$05 = hex 05 = 5, maar het is nu eenmaal gebruikelijk interrupt-vectoren in het hexadecimale stelsel af te beelden.

PROGRAM NoPrtSc;

`{$M 16384,0,0}`

USES Crt, Dos;

Procedure IntHandler05; interrupt;

Begin

Sound(1000);

Delay(50);

NoSound;

End;

Begin

SetIntVec(\$05,@IntHandler05);

WriteLn;

WriteLn('NOPRTSC 1.0 (c) GGPC 1992');

WriteLn('PrtSc-Blocker installed.');

Keep(3);

End.

En inderdaad, lekker kort hè! Dus zelfs "geavanceerde" technieken als TSR's kunnen zo eenvoudig zijn...

Voor de geïnteresseerden: de eerste "WriteLn" is toegevoegd omdat MS-DOS wel na, maar niet voor een klein stukje tekst een blanco regel wenst toe te voegen. Rare jongens, die Amerikanen. Maar goed, een simpel layout-technisch dingetje dus.

Natuurlijk kunt u met de in TRON 45 besproken printer-test een veel mooiere versie ontwikkelen, die eerst zelf snel kijkt (dus niet zo tergend langzaam of vaak als de oude PrtSc-interrupt) of de printer online is en zo ja, dan gewoon verdergaat. Maar... Dat vergt wel het één en ander aan inventiviteit en misschien wel assembly-kunsten, want... Misschien te gevaarlijk, of juist een grote uitdaging? Ga uw gang...

Overigens mag dat programma dan ook wel in het Nederlands hoor!

12D : Het Turbo Pascal Compiler Directives ABC

Dit is nou precies zo'n onderwerp waarvan je zegt: mag dit wel een ABC heten? Er zit immers een flink stuk theorie aan vast! Omdat de hoeveelheid Directives toch groter is dan de hoeveelheid benodigde uitleg, noem ik het toch maar een ABC.

Compiler Directives?

Jep. Hamvraag: Wat zijn dat? Compiler Directives zijn opdrachten voor de compiler. Oftewel: in een stuk programmacode willen we de compiler een opdracht geven. Verwar dit dus niet met gewone programma-code! Dat is heel iets anders... Verder kan een groot aantal opdrachten die via Compiler Directives gegeven kunnen worden ook via diverse parameter-menu's in de Turbo Pascal omgeving worden neergezet. Voordeel van Directives is dat ze in het programma staan. Zo kunnen diverse parameters in uw TP-omgeving in de loop der tijden veranderen, maar het programma wordt niet opeens anders gecompileerd.

Hoe zien ze eruit?

Een Compiler Directive heeft de vorm:

`{$x<parameter,<parameter,<...>>>}`

Bijvoorbeeld de Compiler Directive voor Boolean Evaluation (zie ook ietsje verderop):

`{$B+}`

of de memory-directive van hiervoor:

`{$M 16384,0,0}`

Drie soorten

Er zijn drie soorten directives, namelijk de schakelaars, de parameter-directives en conditionele directives.

Een schakelaar-directive zegt heel eenvoudig: deze optie moet aan of uit. Bijv. `{$B-}` (Boolean Evaluation uit) en `{$B+}` (Boolean Evaluation aan).

De meeste schakelaars kunnen op ieder moment in een programma worden omgezet. Zo kan dus bijv. één procedure op de `{$B-}` manier worden gecompileerd en de rest op de `{$B+}` manier, simpelweg door voor de procedure `{$B-}` neer te zetten en erachter `{$B+}`. We zouden zelfs middenin een procedure één opdracht tussen `{$B-}` en `{$B+}` neer kunnen zetten!

Sommige schakelaars, echter, mogen alleen aan het begin van een programma, dus vlak onder **PROGRAM** of **UNIT**, worden neergezet en gelden dan ook voor het complete programma, meestal inclusief evt. al gecompileerde units.

Denk ook niet dat u alle schakelaartjes steeds zult moeten aan-en uitzetten; Turbo Pascal zet ze allemaal vooraf in een standaard-stand (doorgaans de beste).

Bij **parameter-directives**, de naam zegt het al, moeten diverse extra gegevens worden meegegeven. Kijk maar naar bijv. de `{$M}` directive zoals hiervoor gebruikt, die heeft een aantal parameters nodig.

Conditionele directives worden verderop besproken, om verwarring te voorkomen. Nu eerst het ABC, met daarin **schakelaar-directives** en **parameter-directives** (let maar op de vorm):

`{$A+/-}`

Word Align Data

Globaal, standaard `{$A+}`

`{$A+}` zorgt dat alle variabelen op even geheugenadressen beginnen. Bij `{$A-}` kunnen variabelen ook op oneven adressen staan. Bij elke 80x86-processor levert `{$A+}` tijdwinst op, omdat de processor altijd een even geheugenadres adresseert en daar dan bij `{$A-}` nog 1 van af moet trekken of bij op moet tellen.

`{$B+/-}`

Boolean Evaluatie

Lokaal, standaard `{$B-}`

Bij `{$B-}` zal het programma sneller een booleaanse expressie "berekenen". Wanneer ergens staat "if (x and y)" en x is false, dan kan (x and y) niet meer true worden. Het programma beschouwt dan de booleaanse expressie direct als false, terwijl het bij `{$B+}` de hele expressie (dus ook y) zou zijn gaan uitrekenen. Een kwestie van tijdwinst dus.

`{$D+/-}`

Debug Informatie

Globaal, standaard `{$D+}`

Voegt met `{$D+}` extra informatie aan het gecompileerde programma toe, zodat het te debuggen valt met de ingebouwde debugger, Turbo Debugger of een andere debugger. `{$D-}` maakt debuggen moeilijker tot onmogelijk en spaart ruimte uit. Zie ook `{$L+/-}`.

{\$E+/-}

Coprocessor Emulatie

Globaal, standaard {\$E+}

{\$E+} werkt alleen in combinatie met {\$N+}, zie ook daar. Wanneer {\$E+} is ingeschakeld, zal Turbo Pascal, wanneer er een co-processor benodigd is voor zwaar rekenwerk ({\$N+} dus) deze indien afwezig zelf emuleren (nadoen). Met {\$E-} wordt het programma kleiner, maar werkt het (als {\$N+} was ingeschakeld) niet op een computer zonder 80x87.

{\$F+/-}

Far calls optie

Lokaal, standaard {\$F-}

Turbo Pascal kiest meestal de kortste weg voor het aanroepen van een procedure. Dat wil zeggen: als die procedure in het huidige segment zit, dan springt het programma naar "offset xxxx in het huidige segment" (opdracht 1). Zit die procedure in een ander segment, dan springt-ie naar "segment yyyy met offset xxxx" (opdracht 2). Opdracht 1 is korter, maar mag soms niet gebruikt worden. Bijvoorbeeld wanneer er overlays (zie volgende TRON) in het spel zijn, kan het voorkomen dat een procedure eerst moet worden ingeladen, en zich op offset xxxx in het huidige segment iets heel anders bevindt! {\$F-} genereert "als het effe kan" Opdracht 1 (sneller/minder ruimte) en {\$F+} genereert altijd Opdracht 2 (speciale gevallen). Bij dingen als interrupts, units e.d. zorgt Turbo Pascal er zelf al voor dat {\$F+} aanstaat. Wanneer we deze optie nodig hebben in een aflevering, zal ik dat duidelijk aangeven.

{\$G+/-}

Genereren 80286-code

Lokaal, standaard {\$G-}

De 80286-processor (AT) heeft meer instrukties dan de 8086 (XT) en sommige van die nieuwe instrukties zijn krachtiger varianten van die van de 8086. Met {\$G+} worden dan in plaats van de (mindere) 8086-equivalenten speciale 80286-instrukties geproduceerd. Voordeel: het loopt nog wat lekkerder. **Enorm nadeel:** een dergelijk programma werkt niet op een XT! En denk nou niet "die zijn er toch niet meer", want die worden nog steeds veel gebruikt. {\$G-} gebruiken dus, tenzij er echt speciaal voor een 286-machine geschreven wordt.

{\$I+/-}

I/O Controle

Lokaal, standaard {\$I+}

Wanneer er een I/O-fout (bijv. "disk vol") optreedt, zal met {\$I+} het programma direct worden beëindigd. Erg lelijk dus! Met {\$I-} komt een evt. I/O-fout gewoon als waarde terecht in de variabele **IOResult** (een oude bekende!) en

kunnen we deze fout dus zelf afvangen.

{\$I filenaam}

Include

Lokaal

Dit is een (ouderwetse) manier om stukken Pascal-source die in andere bestanden op disk staan op deze positie in het programma in te voegen. Let op: Dit mag niet binnen een **Begin...End** constructie gebeuren, dus altijd tussen verschillende procedures of aan het begin van een programma. Een geïncludeerd stuk programma (filenaam geeft de naam aan waarmee het programma op disk staat, de extensie ".PAS" mag achterwege worden gelaten) wordt door Turbo Pascal gewoon ingelezen (op het moment dat-ie de {\$I xxx} tegenkomt) en dan gecompileerd alsof de opdrachten op de huidige plek stonden. Een soort units dus, maar onhandiger, omdat hier de source wordt ingeladen en gecompileerd (iedere keer weer!), terwijl units kant-en-klaar zijn. Deze methode wordt eigenlijk zelden meer gebruikt, maar het kan nooit kwaad om te weten wat het betekent, want misschien komt u het eens tegen in een in Turbo Pascal 3.01A geschreven programma. Overigens mogen er meerdere {\$I xxx} directives worden gebruikt, dus meer dan 1 include-bestand.

{\$L+/-}

Lokale symbolen-informatie

Globaal, standaard {\$L+}

Om bijv. tijdens het debuggen ook de namen van lokale variabelen e.d. te kunnen koppelen aan een waarde, bewaart Turbo Pascal speciale informatie over al die variabelen in het programma met {\$L+}. Dit maakt o.a. het debuggen veel gemakkelijker (zie dus ook {\$D+/-}). Met {\$L-} wordt deze informatie niet bewaard, is het programma dus moeilijker te debuggen en wordt het (soms een stuk) kleiner.

{\$L filenaam}

Link object-file

Lokaal

Hangt een .OBJ-file (machinetaal) aan procedures die "external" zijn gemaakt, oftewel: een procedure die in machinetaal geschreven is wordt via **external** aan een TP-procedure gehangen en met {\$L xxx} vertellen we in welke .OBJ-file de gezochte procedure zit. Hier komen we in de verre toekomst wellicht weer op terug. Zie het voorlopig maar als een soort {\$I xxx} voor pure machinetaal. Overigens mogen ook meerdere {\$L xxx} directives worden opgegeven.

{\$M stack,heapmin,heapmax}

Geheugen-reserveringsgrootte

Globaal, standaard {\$M 16384,0,655360}

{\$M} geeft de hoeveelheid te reserveren geheugenruimte

voor stack en heap (dynamische variabelen) aan, maar alleen in een programma en werkt dus niet in een unit (vanuit het hoofdprogramma werkt een evt. {\$M} nl. door in alle units tijdens de compilatie). {\$M} is voornamelijk belangrijk wanneer het gaat om een TSR-programma of een programma dat zelf weer een ander programma aanroept; we moeten dan het geheugengebruik van het huidige programma beteugelen. Zie ook hierboven, les 13C.

{\$N+/-}

Numeriek

Globaal, standaard {\$N-}

Met {\$N-} zal Turbo Pascal alle opdrachten waarbij dat nodig is ("floating point"- of "drijvende komma"-opdrachten) zelf uitvoeren. Met {\$N+} gebruikt-ie daarvoor de 80x87 co-processor of een stukje eigen, extra code (zie {\$E+/-});

{\$O+/-}

Overlays

Globaal, standaard {\$O-}

Met {\$O+} maakt de compiler de unit of het programma gereed om eventueel overlaid te worden (dit is dus niet meteen verplicht, het is alleen nu pas mogelijk!). Bij {\$O+} dient bijna altijd ook {\$F+} in combinatie te worden gebruikt, zie dus ook aldaar. Bij {\$O-} kan het programma of de unit niet meteen overlaid worden indien nodig, daarvoor moet het dus eerst met {\$O+} gecompileerd worden. Voordeel van {\$O-} is dat het minder ruimte kost en dus normaliter de beste keus is. In de volgende TRON gaan we (echt!) overlays gebruiken...

{\$O unit-naam}

Overlay-bestand

Lokaal

Kan alleen gebruikt worden in het hoofdprogramma. De units die genoemd worden, zullen worden gestopt in een .OVR-bestand, zodat ze geoverlaid zullen worden. Dit kan natuurlijk alleen als zo'n unit was gecompileerd in {\$O+} mode. In de volgende TRON alles over lays...

{\$R+/-}

Range controle

Lokaal, standaard {\$R-}

Met {\$R+} wordt tijdens het draaien van het programma continu gecontroleerd of opgegeven string-waarden niet te lang zijn (dus bijv. "LangeString[100] := 'a';" als LangeString een String[5] is, mag niet) en of alle arraybewerkingen en alle getallen wel binnen het bereik van hun variabele-typen blijven. Constateert het programma tijdens het draaien zo'n fout, dan wordt het meteen beëindigd. {\$R-} verdient de voorkeur, want het neemt minder geheugenruimte

in beslag (geen) en het maakt het programma ook niet langzamer door al die controles. {\$R+} dus eigenlijk alleen gebruiken wanneer het programma nog niet af is (testen/debuggen).

{\$S+/-}

Stack controle

Lokaal, standaard {\$S+}

Bij {\$S+} wordt binnentussen het programma continu gekeken (bij elke procedure of functie die wordt aangeroepen) of er wel voldoende ruimte beschikbaar is voor o.a. de benodigde lokale variabelen. Lukt dat niet, dan stopt het programma tijdens uitvoering. Maak dan de stack wat groter, zie ook {\$M}. Bij {\$S-} heb je zo'n controle niet, het is ietsjes sneller en kost ietsjes minder ruimte, maar meestal veroorzaakt zo'n stackprobleem dan wel dat de computer hangt... Niet zo leuk natuurlijk!

{\$V+/-}

Var-string controle

Lokaal, standaard {\$V+}

Wanneer een string die wordt doorgegeven als parameter een andere lengte heeft dan staat aangegeven (dus bijv. lengte 73 i.p.v. 52) dan valt Turbo Pascal daarover tijdens het compileren als {\$V+} is ingesteld. Bij {\$V-} zal het TP worst wezen wat we met onze strings uitvoeren...

{\$X+/-}

Extra functie-optie

Globaal, standaard {\$X-}

Via {\$X+} kan een te compileren functie (dus geen functie van Turbo Pascal) worden aangeroepen zonder de uitkomst op te vragen, dus niet "xxx := Functie(...);" maar gewoon "Functie(...);". Let op: Natuurlijk blijft de oude aanroep-manier ook geldig! {\$X+} kan soms handig zijn, wanneer een functie bij een bepaalde waarde onzin produceert, maar probeer het normaal gesproken niet te doen, want het is wel een erg opvallende ontduiking van de Pascal-syntaxis! {\$X-} zorgt er natuurlijk voor dat een functie alleen maar als functie kan worden aangeroepen.

Conditionele Compiler Directives

Met C.C.D.'s is "conditionele compilatie" mogelijk. Oftewel: een stuk programma wordt alleen gecompileerd ALS... Een voorbeeld: u hebt een programma dat op een gegeven moment uw naam laat zien als die van de auteur, maar nu bent u erg bescheiden en wilt u die naam liever niet laten zien als het programma bij u thuis wordt gedraaid (voor als er vrienden op bezoek komen ofzo). In zo'n geval kunt u aangeven of het programma wel of niet voor uzelf bestemd is en dan in de

source aangeven of een stuk source wel of niet gecompileerd moet worden. Een beetje abstract? Kijk dan even heel goed naar dit voorbeeld:

```
{$DEFINE VoorAnderen}
```

Begin

```
 {$IFDEF VoorAnderen}
 writeln('Geschreven door MIJ!');
 {$ENDIF}
```

End;

Met {\$DEFINE} wordt een "label" (symbool) aangemaakt met de naam VoorAnderen. De compiler komt nu een CCD-tegen ({\$IFDEF}) en kijkt of VoorAnderen bestaat. Zo ja, dan compileert hij de regel, zo nee, dan dus niet. **Let er op** dat we altijd moeten aangeven wanneer de CCD niet meer geldt! Dat gebeurt dus met {\$ENDIF}. **Onthou goed:** CCD's hebben **direct** invloed op wat er gecompileerd moet worden! Als VoorAnderen niet is gedefinieerd, **wordt de regel met uw naam helemaal niet mee-gecompileerd!** Overigens kunnen CCD's ook bepalen of bepaalde compiler directives wel of niet worden gebruikt, zie ook verderop onder **CPU87**.

Dan de mogelijke opdrachten:

{\$DEFINE naam}

Definieert een bepaald symbool. Wordt deze CCD niet gebruikt, dan bestaat het symbool **niet**.

{\$UNDEF naam}

On-definieert een bepaald symbool, oftewel: vanaf nu bestaat het niet meer.

{\$IFDEF naam}

Alle volgende source (tot {\$ELSE} of {\$ENDIF}) wordt alleen gecompileerd als **naam** bestaat.

{\$IFNDEF naam}

Alle volgende source (tot {\$ELSE} of {\$ENDIF}) wordt alleen gecompileerd als **naam** niet bestaat.

{\$IFOPT schakelaar}

Alle volgende source (tot {\$ELSE} of {\$ENDIF}) wordt alleen gecompileerd als **schakelaar** in de aangegeven stand staat. Dus bijv. "{\$IFOPT L+}" kijkt of {\$L+/-} op {\$L+} staat.

{\$ELSE}

Compileert source als de vorige {\$IF...} CCD de source erachter niet heeft gecompileerd.

{\$SENDIF}

Beëindigt een {\$IF...} lus.

Standaard-symbolen

Er zijn in Turbo Pascal 6.0 alvast een paar symbolen voor-gedefinieerd, die het één en ander aangeven. Het betreft de volgende:

VER60

Gedefinieerd als/omdat de huidige TP-versie 6.0 is. In oudere versies dus **VER40**, **VER55** etc.

MSDOS

Gedefinieerd als/omdat Turbo Pascal onder MS-DOS draait. Oudere versies, zoals TP 2, werkten ook onder CP/M etc, zodat we via {\$IFDEF MSDOS} zouden kunnen afleiden of we wel of niet onder MS-DOS werken. De nieuwere versies werken natuurlijk allemaal onder MS-DOS.

CPU86

Gedefinieerd als we met een 80x86-processor werken (8088 ook).

CPU87

Gedefinieerd als we met een 80x87-coprocessor werken. Bijvoorbeeld:

```
 {$IFDEF CPU87} {$E-} {$ELSE} {$E+} {$ENDIF}
```

Let op!

Let erop dat het hier telkens gaat om de computer waarop wordt gecompileerd, niet degene waarop het programma wordt gedraaid!

't Is weer voorbij...

Voor vandaag... Jammer, we zijn inderdaad niet aan overlay's toegekomen, maar we hebben toch drie interessante facetten van Turbo Pascal kunnen belichten, dacht ik.

Over twee maanden **natuurlijk** die overlays en, zoals beloofd, een dubbel-aflevering van het Enkeltje Pascal...

Vragen? Problemen? Suggesties?

Roeland van Zeijst
Kerklaan 49
7311 AD Apeldoorn
Sprak 055-213013
Videotex 055-226116
Tot TRON!

Flexibele Produktie Automatisering (6)

door Peter de Groot

In deel 5 schreef ik over de IBM-Robot, die alle fout gelegde kogels moest herplaatsen. Als alle kogels herplaatst zijn, gaat de pallet voor de tweede keer naar het vision systeem, hier wordt gekeken of de kogels nu wel goed zijn geplaatst. De pallets gaan, na de controle, naar de Cloos-Knikarm-Robot. Als de pallets zijn afgekeurd, moet de Cloos-Robot de pallets gewoon voorbij laten gaan. De goedgekeurde pallets moeten echter door de robot van de band worden gehaald.

Dinsdag - De Cloos-Knikarm-Robot

Het vorige deel over FPA ging over m'n bezigheden op Goede Vrijdag. Maandag hadden we vrij in verband met 2e Paasdag. Dus beginnen we nu met de dinsdag na Pasen. Voor mij staat op het programma de Cloos-Knikarm-Robot. Deze robot kent 6 vrijheidsgraden en is dus erg nauwkeurig.

Zie bijlage 1.

Toch is het werkgebied kleiner dan die van de ASEA-robot (Tron 44) omdat de armen van de Cloos kleiner zijn. De bediening is vrij simpel. Vanuit de besturingskast geven we aan de robot door dat we hem gaan besturen met het bedieningspaneel. We moeten het werkgebied van de robot zien als een x, y, z opstelling. We geven op het bedieningspaneel commando's als X-, X+, Y-, Y+, Z- en Z+. Zie foto 1.

Zo bewegen we de Tool Centre Point (TCP - 'Hand') naar verschillende posities. We kunnen ook elke as afzonderlijk bedienen. Als we een belangrijke positie hebben bereikt, geven we een commando waarna de besturingskast de positie opslaat. Zo gaan we alle belangrijke posities langs. Dat is een heel karwei want de robot mag de pallets niet gaan stapelen, maar moet ze netjes naast elkaar rangschikken, zodat inpakkers de pallets zo van de tafel af in de doos kunnen leggen.

Als alle posities zijn opgeslagen, lopen we terug naar de besturingskast en geven het commando: **SAVE PROJ1 TO E:**

PROJ1 is een bestandsnaam en mag ook een andere max. 8 bytes lange naam zijn. Maar kijk nu eens goed naar het commando. Heb ik dat niet al eens eerder gezien?



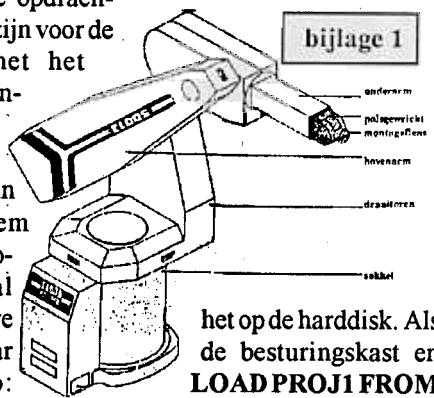
Foto 1: Chris beweegt de Robot m.b.v. het bedieningspaneel.

*Ja, inderdaad in Tron 43 over netwerken door Jeroen Hoppenbrouwers. Hier is dus sprake van een heel klein netwerkje tussen de besturingskast van de robot en de PC. Dus lopen we naar de desbetreffende PC en laden een programma, waarmee we PROJ1 kunnen wijzigen en verder af kunnen programmeren. We vullen nu ook commando's in als **GRIJP VAST** en **LAAT LOS**.*

Ook verwerken we opdrachten, die belangrijk zijn voor de communicatie met het vision systeem; tenslotte moet de robot straks als slaaf (Slave) werken van het vision systeem (Master). Als het programma helemaal klaar is, bewaren we het op de harddisk. Als de besturingskast en het commando:

LOAD PROJ1 FROM E: geven, wordt het gewijzigde programma in de besturingskast geladen. Als we het programma laten draaien, hebben we nog steeds de nood-stop bij de hand, want de robot ziet niet, wat wij wel zien. Zie foto 2.

Pas als het hele programma zonder problemen is doorlopen, kan het systeem zelfstandig werken (foto 3).



De praktijk

De Cloos-Robot gaat klaarstaan om de pallets van de band te halen. Een pallet wordt bij station 1 op de lopende band geplaatst. Bij station 2 wordt een figuur met behulp van kogels op de pallet gelegd. Bij het vision systeem wordt gekeken of de figuur wordt herkend. Als het niet wordt herkend, mag de Cloos-Robot de pallet niet van de band halen en wordt doorgezonden naar de fouten-analyse en de herstel (IBM) robot. Wordt de figuur wel herkend, dan bedient het vision-systeem een stop op de lopende band, waardoor de pallet en de palletdrager worden tegengehouden. Nu krijgt de Cloos-Robot een signaal, waarop de robot de pallet van de lopende band haalt.

Hierna stuurt de Robot een signaal terug, zodat het vision-systeem de stop kan laten vervallen. Terwijl het vision-systeem de volgende pallet keurt, legt de robot de vorige pallet op z'n plaats en gaat weer klaar staan voor de volgende. Dan geeft de robot nog een signaal, als teken dat hij klaar is voor de volgende pallet.

Zoals je ziet, is het draaiend maken van een robot niet erg moeilijk. Het wordt pas moeilijk als het gaat om massa produktie en communicatie. Op die communicatie kom ik

EMM386 en Extra Upper Memory

door Guido Klemans



Foto 2 : Ik kijk of de robot het programma goed doorloopt.

later nog weer terug, maar bij massa produktie is het bovenstaande systeem niet voldoende.

Dan gaan we gebruik maken van zogeheten schuifregisters; deze zijn erg belangrijk, want bij massaproductie kan het vision systeem echt niet wachten, tot de robot klaar is met het plaatsen van de pallets. Dus moet de robot gaan tellen pallet 1, 3, 4, 5 en 7 zijn goedgekeurd en pallets 2 en 6 niet enz. Dit tellen wordt gedaan in die schuifregisters.

Woensdag wordt het een drukke dag. Voor mij staat een lezing over PC's, verschillende computers en de geschiedenis daarvan op het programma. Ook ga ik mij bezig houden met de fouten-analyse, een apparaat dat met behulp van de PLC verkeerd gelegde figuren onderzoekt, zodat de IBM-robot de fouten kan herstellen.

Graag tot de volgende keer.

Peter de Groot
Postbus 63
9800 AB Zuidhorn
05940-5207



Foto 3:
Terwijl Chris kijkt
naar de communicatie op de PC van
het vision systeem,
doet de robot het
werk helemaal
zelfstandig.

Bij MS-DOS 5.0 en MS-Windows 3.x treft u het programma EMM386.EXE aan. Zoals de naam al doet vermoeden kan dit programma enkel worden gebruikt met een 80386 (of 80486) processor. EMM386 heeft meer functies: het is een expanded memory emulator, volgens de EMS 4.0 specificatie en het is een driver voor Upper Memory Blocks (UMBs). UMB's zijn blokken RAM geheugen boven de 640Kb grens en onder de 1Mb grens. Waarom Microsoft er voor gekozen heeft om deze twee compleet verschillende zaken in een programma te stoppen is een mij een raadsel. UMB's vallen onder de extended memory specification (XMS) en zouden dus eerder bij HIMEM.SYS thuis horen, EMM386 heeft daarom ook HIMEM nodig om te kunnen werken.

EMM386 zoekt zelf uit, welk geheugen als UMB in aanmerking komt. Dit verschilt (uiteindelijk) per computer. Helaas is EMM386's detectie niet erg efficient. Er wordt nogal eens wat geheugen overgeslagen en dat is niet de bedoeling. Gelukkig is er de RAM=xxxx-yyyy optie, waarmee kan worden aangegeven, waar beschikbaar geheugen is te vinden. Op de plaats van xxxx moet het segmentadres van het begin van het geheugenblok worden ingevuld, voor yyyy het segmentadres van het einde van het blok. Deze optie heeft echter een nadeel: EMM386 controleert blijkbaar niet (goed) of er wel RAM beschikbaar is! Op een systeem met maar 128Kb beschikbaar voor UMB's, maakt EMM386 even vrolijk 200Kb UMB's aan! Op de één of andere manier lijkt het nog zo dat er geheugen aanwezig is ook. Waarschijnlijk worden stukken geheugen dubbel gebruikt en dat gaat niet lang goed!

Een gewaarschuwd mens telt voor twee,
EMM386's UMB's ook.

UMB'S

Als u een PC met een 80386 of 80486 processor en een BIOS van American Megatrends Inc. (AMI) heeft, dan is het mogelijk een extra UMB van maximaal 32Kb aan te maken. Het BIOS ROM van de PC is een blok van 64Kb en begint op adres F000:0000. Bij de meeste AMI-BIOSsen wordt slechts de laatste 32Kb (vanaf F800:0000) gebruikt. De eerste 32Kb is alleen nodig tijdens het booten en verzorgt o.a. het CMOS-setup programma; dit blok wordt daarom ook wel de bootpage genoemd. Als u het BIOS in shadow-RAM hebt staan, (dit kan worden ingesteld met de setup) dan kan waarschijnlijk de bootpage, of een deel daarvan, naar UMB worden omgezet. Bij gebruik van EMM386 gaat dit bijvoorbeeld door de parameter RAM=F000-F7FF toe te voegen.

Als sommige programma's nadien niet meer werken, dan zit er iets niet lekker en moet het UMB worden verkleind. Experimenteer! Onder DOS is ieder bit de moeite waard.

Heeft u dat ook wel eens? (2)

Peter de Groot

In de Tron nummer 45 had ik het volgende stukje geschreven: U kijkt naar een wedstrijd golf en de bal gaat recht op de hole af, maar mist toch net 2 centimeter. U tilt het televisietoestel aan een kant een beetje op en...ja hoor de bal blijft netjes liggen waar hij ligt. Dit zal het nodige geschater teweeg brengen. Maar wat nu, als de bal er wel in was gegaan, door uw trucje met het televisietoestel?

Met de opdracht om dit eens voor u zelf uit te denken, heb ik het stukje afgesloten. U heeft er nu een half jaar de tijd voor gehad, om uw versie te bedenken. In dit artikel zal ik vertellen hoe de hardware hiervoor eruit ziet.

Het grote geheim

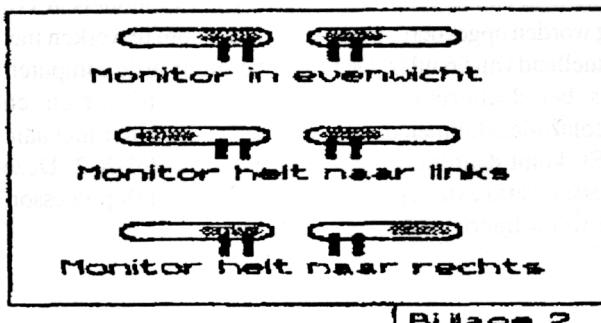
Het grote geheim zit in twee, in de monitor ingebouwde, kwikschakelaars. Deze kwikschakelaars zijn, schaars, verkrijgbaar in elektronica winkels voor rond de 10 gulden per stuk. Pas wel op: **KWIK IS SCHADELIJK VOOR GEZONDHEID EN MILIEU!**

Verder heb je nog nodig ongeveer 1 à 1,5 meter afgeschermd 2-adrig kabel en een RS-232 aansluiting.

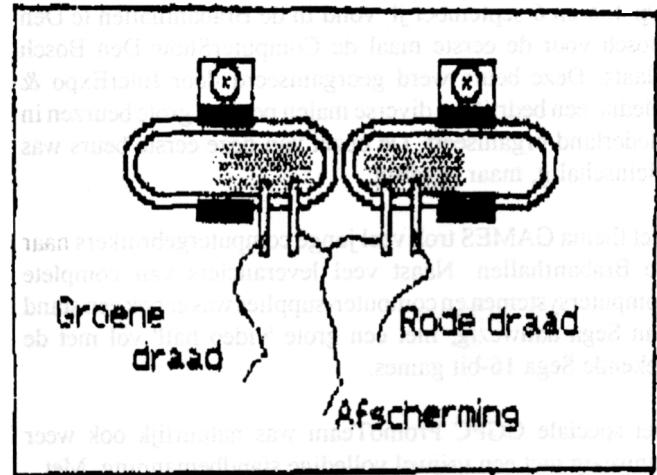
De kwikschakelaars worden op een lege plaats in de monitor aangebracht. Pas ook hier goed op: **OOK ALS DE STEKER UIT HET STOPCONTACT IS KUNNEN ER NOG LEVENSGEVAARLIJKE SPANNINGEN HEERSEN IN DE MONITOR!!**

De schakelaars moeten (bijna) horizontaal worden aangebracht en zodanig dat als de monitor waterpas staat, in beide schakelaars het kwik de aansluitpunten doorverbindt.

Als de monitor naar links wordt gekanteld, moet bij schakelaar 1 het kwik zodanig bewegen dat de aansluitpunten niet meer doorverbonden zijn, terwijl dat bij schakelaar 2 nog wel het geval is.



Bijlage 2



Bijlage 1

Wordt de monitor naar rechts gekanteld, dan moet het kwik in schakelaar 2 zodanig bewegen zijn, dat de doorverbinding vervalt terwijl schakelaar 1 doorverbonden blijft.

Zie bijlage 2.

Nu wordt de groene draad van de kabel aangesloten op één van de aansluitingen van schakelaar 1, de rode draad wordt aangesloten op één van de aansluitingen van schakelaar 2.

De afscherming wordt op de beide overgebleven aansluitingen aangesloten. Zie bijlage 1.

Nu kan de monitor weer dicht. We moeten alleen de RS-232 connector aansluiten op de kabel. De groene draad gaat naar aansluitpool nr. 20, de rode draad gaat naar aansluitpool nr. 2 en de afscherming gaat naar aansluitpool nr. 5. Somige mensen vinden het eng om iets nieuws aan te sluiten op hun dure computer.

Deze mensen raad ik aan om tussen de groene draad een weerstand aan te sluiten van ongeveer 1 kOhm en evenzo voor de rode draad. Maar in principe kan het geen kwaad om de weerstanden weg te laten.

Nu is de hardware klaar, test het nog maar even goed uit. De volgende keer behandel ik de software.

Graag tot de volgende keer!

Peter de Groot
Postbus 63
9800 AB Zuidhorn

ComputerShow Den Bosch

Op 4, 5 en 6 september jl. vond in de Brabanthallen te Den Bosch voor de eerste maal de ComputerShow Den Bosch plaats. Deze beurs werd georganiseerd door InterExpo & Media, een bedrijf dat diverse malen per jaar grote beurzen in Nederland organiseert. De opzet van deze eerste beurs was kleinschalig, maar gezellig.

Het thema GAMES trok veel jonge computergebruikers naar de Brabanthallen. Naast veel leveranciers van complete computersystemen en computer-supplies was er ook een stand van Sega aanwezig, met een grote 'video hall' vol met de bekende Sega 16-bit games.

Het speciale GGPC PromoTeam was natuurlijk ook weer aanwezig met een vrijwel volledige standbemanning. Met behulp van de Covox interface schalde naast aandacht trekende muziek, het zeer toepasselijke Veronica motto 'Jong snel en wild ...' door onze stand.

Roeland en Andor verzorgden met de telefooncentrale van de Databrothers een demonstratie van Vredenborch, werkend op het videotex host systeem QDR/ProTel.

Diederik van Bochove toonde de nieuwste versie van DiedjeTel. DiedjeTel is een multi-tasking videotex host voor, jawel, de P2000!

De meeste aandacht kregen echter de Databrothers met een simpel, doch zeer leuk spel op de P2000. Op een computerscherm stond cowboy "Billy the Kid" die zijn pistool trok. De bedoeling was dat de gebruiker Billy the Kid te snel af was en deze neerschoot met, een aan de P2000 verbonden pistool. Per keer dat er raak werd geschoten kon men punten verdienen en na afloop kreeg iedereen een kaartje mee met daarop de hoogst behaalde score.

Na afloop van de, naar mijn mening, gezellig en goed georganiseerde beurs waren we enkele deelnemers rijker. Al met al een redelijk geslaagd begin van een nieuw beursseizoen.

Op 25, 26 en 27 september wordt de Benelux Computer 1992 georganiseerd in het Beursgebouw in Eindhoven. Een verslag van deze beurs zult u in de volgende TRON aantreffen.

Andor Vierbergen

SiePieJoes

door Andor Vierbergen

Een zwart oppervlakje van enkele vierkante centimeters is het eigenlijke hart van een computer. Dit kleine doch ingewikkelde onderdeel bestuurt vrijwel de gehele computer. Er zijn op de wereld talloze soorten processors. De meest bekende zijn heden ten dage wel de processors van INTEL. Deze werden pas echt populair doordat IBM ze ging gebruiken in haar personal computers aan het begin van de jaren tachtig.

Bij de aanschaf van een nieuwe computer is het van belang te letten op de kwaliteiten en eigenschappen van de verschillende processors. De kenmerken die gebruikt worden door leveranciers zijn echter niet altijd even duidelijk. SiePieJoes zal dan ook gaan over de soorten processors die er zijn en hoe een processor intern in elkaar zit.

Om te beginnen wil ik het dit keer hebben over de nieuwste processor van INTEL, de i486. vergeleken bij de processors die we nu kennen, erg snel en geavanceerd. Van de i486 zijn er natuurlijk ook weer veel verschillende types op de markt.

i486SX

Zo is er de i486 met als type aanduiding SX. SX wil zeggen dat de processor geen co-processor bevat. De coprocessor, een mathematische, die er normaal gesproken inzit, is buiten werking gesteld. De prijs van deze processor is daarom ook duidelijk lager dan de normale processor types. Wanneer de computer veel wiskundige berekeningen moet uitvoeren, bijvoorbeeld wanneer er gewerkt wordt met uitgebreide tekenapplicaties, zoals een CAD/CAM programma, dan zal de computer hier een vertraging in ondervinden. De i486SX is verkrijgbaar in de snelheden 25 mHz, 33 mHz en 50 mHz.

i486DX

Dit type is de eigenlijke i486, met alles erop en eraan. De kloksnelheid van de processor, de snelheid waarin de opdrachten worden verwerkt varieert, evenals de SX, van 25 mHz, 33 mHz tot 50 mHz.

i486DX-2

De snelheid van de huidige processors kan inmiddels al vrij hoog worden opgevoerd. De i486 is in staat om te werken met een snelheid van 66 mHz. De randapparatuur in de computer, zoals beeldschermadapters, communicatiekaarten en diskcontrollers kunnen deze hoge snelheden echter niet aan. INTEL komt daarom op de markt met de i486DX-2. Deze processor werkt extern precies als een normale DX processor, maar werkt intern op dubbele snelheid.

Voordeel is dat berekeningen die binnen de processor worden uitgevoerd veel minder tijd kosten. De i486DX-2 is verkrijgbaar in de snelheden 25/50 mHz en 33/66 mHz.

i486DX Overdrive

De aansluitpinnen van een i486DX-2 verschillen echter in vergelijking met de normale i486DX. Gewoon een i486DX-2 processor aanschaffen en in de oude processor voet plaatsen is er dus niet bij. Om te voorkomen dat niet iedereen een nieuwe computer aan hoeft te schaffen brengt INTEL ook de i486DX Overdrive op de markt. Dit is een zelfde soort processor als de i486DX-2, en werkt dus ook op 25/50 mHz of 33/66 mHz. De i486DX Overdrive wordt echter in de co-processor voet geplaatst en de oude processor kan gewoon blijven zitten. De nieuwe processor schakelt deze geheel uit en neemt alle functies over. Zo kan er toch gebruik worden gemaakt van de nieuwe snelle processors.

i586

Volgens diverse berichten zal de i586 de laatste processor in haar soort van INTEL zijn. INTEL gaat dan over op het maken van zogenaamde RISC processoren, een 'nieuwe' technologie in het ontwerpen van processors. Meer over de huidige CISC en de 'nieuwe' RISC processoren in een volgende aflevering van SiePieJoes.

Digital Dancing, door Andor Vierbergen

In september introduceerde Philips haar nieuwste vindingen, die ervoor moeten zorgen dat één van de belangrijkste onderdelen van het bedrijf, de "Consumenten Electronica", weer winstgevend wordt. De hoop is daarbij gevestigd op onder andere DCC.

DCC

DCC staat voor Digital Compact Cassette en is dus een digitale audio cassette. Met deze cassettes is het mogelijk om digitaal geluid op te nemen zonder kwaliteitsverlies. Een opname van een Compact Disc zal een nagenoeg zelfde kwaliteit geven, zeker in vergelijking met de huidige cassettes. Het voordeel van de DCC is dat alle oude muziekcassettes af te spelen zijn in de DCC spelers. Daarmee is alles wel zo'n beetje gezegd, want in Japan komt Sony op de markt met de MiniDisk, afgekort tot MD. Deze diskette, die kleiner is dan de bestaande computer diskettes, is net als de DCC bedoeld voor muziek. Voordeel van de MD is dat het zoeken naar

bepaalde muziekstukken veel sneller gaat. Het is in principe hetzelfde als een normale computer diskette. Er hoeft dus niet eerst gespoeld te worden voordat een stuk is opgezocht. Verder heeft de afmeting van de MD als voordeel dat er veel compactere apparatuur voor te maken is. Dit is voor Sony, de uitvinder van de Walkman, erg belangrijk.

Volgens experts zal het Philips daarom niet lukken de Japanse markt te overwinnen. De verwachting is dat het zeker nog een jaar of twee à drie duurt voor het grote geld binnen gaat komen. Vandaar dat veel grote elektronica bedrijven rustig de kat uit de boom kijken alvorens zich op een bepaald systeem vast te leggen. Van beide systemen zullen nog voor het einde van het jaar meer dan 5000 muziekstukken verkrijgbaar zijn.

Andor

Nieuwe operating systems

door Jeroen Hoppenbrouwers

Al bijna drie jaar loop ik het rond te bazuinen en nu is het ogenblik dan gekomen: DOS heeft nu werkelijk een serieuze concurrent, en binnen korte tijd zelfs twee. Wat is hier voor de gemiddelde gebruiker nu eigenlijk van belang?

Iets anders

Om te beginnen wil ik even niet de fout begaan, die zowat alle andere computerbladen wel maken. Ik ga géén uitgebreide vergelijking aan tussen de grafische DOS-schil Windows 3.1 en de schil van OS/2 2.0 (de Workplace Shell). Wanneer je beweert dat OS/2 niet nodig is omdat de grafische mogelijkheden lijken op die van Windows, heb je een klein beetje gelijk en schiet je een vreselijke bok op hetzelfde moment.

Windows is een grafische schil om DOS heen. Onderin de computer ligt een bodem van DOS, daar overheen ligt Windows, en daar weer overheen lopen de programma's zoals Excel en Word. Windows doet zijn uiterste best om van DOS nog wat te maken, en met ondersteuning van een 386 of grotere processor lukt dat vrij aardig, maar het blijft puur gebroddel. Windows is en blijft een bibliotheek van grafische procedures, die een grafische user interface mogelijk maken, meer niet. DOS is en blijft een single-tasking interrupthandler voor disks.

OS/2 is een compleet nieuw besturingssysteem dat net zoals

DOS onderin de computer ligt en DOS dus niet aanvult, maar vervangt. Onderin de computer handelt het alle taken af, die met het beheer van de machine te maken hebben. Waar onder DOS programma's overal bij kunnen (alle geheugenadressen en I/O-poorten zijn vrij toegankelijk), sluit OS/2 deze toegang af. Een programma kan en mag alleen bij de hardware, die het ook werkelijk nodig heeft en meestal is dat beperkt tot een klein stukje werkgeheugen. De rest doet OS/2 zelf op verzoek. Op deze manier wordt effectief voorkomen dat een verkeerd programma de hele machine lamlegt of laat crashen. Verder biedt OS/2 ook echte multitasking en is het volledig vergelijkbaar met grotere systemen zoals UNIX.

Wanneer je Windows dus vergelijkt met OS/2, dan vergelijk je een fraai menuprogramma met een compleet besturingssysteem dat onder andere ook een fraai menuprogramma aan boord heeft. Zo'n vergelijking moet wel mank gaan.

Natuurlijk zal voor de gemiddelde gebruiker het onderscheid veel minder duidelijk zijn, want wanneer je alleen een Windows-beeldscherm of een OS/2 Workplace Shell (WPS)-beeldscherm voor je ziet, lijken de twee systemen nogal op elkaar. Natuurlijk, het ene menuprogramma lijkt op het andere menuprogramma. Maar dan ben je Windows aan het vergelijken met de Workplace Shell. De gerenommeerde computerbladen zouden eens Windows en de WPS moeten schrappen en alleen het kale DOS met OS/2 moeten vergelijken. Dat is eerlijk.

Is er leven na DOS?

Omdat zo langzamerhand de problemen die het oude DOS veroorzaakt ook aan de gemiddelde gebruiker duidelijk beginnen te worden (gedonder met memory managers, onoverzichtelijke configuratie, programma's die niet meer in het geheugen passen, slechte ondersteuning van achtergrondverwerking met zogenaamde TSR-programma's, ronduit slechte harddiskbehandeling en de gevoeligheid voor crashes), is de computerwereld echt hard bezig een nette vervanger voor DOS te zoeken. In principe is dat helemaal niet moeilijk, want er bestaan al jaren uitstekende besturingssystemen, die prima op een PC draaien. UNIX is er zoéén.

Helaas wil de wereld al de oude DOS-programma's niet kwijt, zodat een nieuw systeem de DOS-programmatuur zonder problemen moet kunnen draaien. Een systeem dat dit niet kan, zoals UNIX nu, heeft geen schijn van kans.

Zo blijven er twee kandidaten voor de opvolging van DOS over: OS/2 van IBM en het nog niet verkrijgbare Windows NT (New Technology) van Microsoft. OS/2 is een systeem dat behoorlijk op UNIX lijkt, maar beter op de één-persoon-per-computer PC-wereld is toegesneden

(UNIX is niet alleen multi-tasking, maar ook multi-user). OS/2 bevat naast de normale zaken ook een speciale DOS-omgeving, waarin een aantal bestaande DOS-programma's naast elkaar kunnen draaien, zonder elkaar in de weg te zitten. Verder heeft OS/2 een grafische schil, de Workplace Shell (WPS) en staat het bestaande Windows-programma's toe, om direct binnen deze WPS actief te zijn.

Windows NT is een systeem dat behoorlijk op UNIX lijkt maar beter op de één-persoon-per-computer PC-wereld is toegesneden. NT bevat naast de normale zaken, ook een speciale DOS-omgeving waarin een aantal bestaande DOS-programma's naast elkaar kunnen draaien. Verder heeft NT een grafische schil en staat het bestaande Windows-programma's toe, om direct binnen deze schil actief te zijn.

Hé, dat lijkt toch wel erg veel op elkaar. Inderdaad, OS/2 en NT zijn gewoon allebei moderne besturingssystemen die dus hetzelfde presteren. En reken maar dat wanneer de één een nieuw foefje introduceert, de ander er binnen een maand óók mee is uitgerust. Ze zullen zelfs elkaars programma's begrijpen. De vraag "wordt het Windows 3.1, OS/2, of Windows NT", is dus eigenlijk onzin. Allereerst moet Windows 3.1 er uit, omdat dat, zoals gezegd, geen besturingssysteem is -- de vraag wordt dan al "wordt het DOS, OS/2, of Windows NT?". Het verschil tussen OS/2 en Windows NT is op het moment erg eenvoudig: OS/2 is in de winkel te halen, Windows NT bestaat alleen nog maar op de computers van Microsoft en in een kreupele developers release. Dan hebben we de vraag "wordt het DOS of wordt het OS/2?". Maar laten we Microsoft nog één keer het voordeel van de twijfel gunnen en de vraag definitief stellen op "blijft het DOS of wordt het OS/NT?".

Voor elck wat wils

Er is een groep mensen die voldoende heeft aan DOS, net zoals er een groep mensen is die voldoende heeft aan een P2000 of aan een schrijfmachine. Wanneer je je PC alleen gebruikt als uitgebreide typemachine, calculator of kaartentbak en je niet teveel eisen stelt aan prestaties of gebruikersvriendelijkheid, dan is DOS een zeer goedkope en tamelijk betrouwbare keuze. DOS loopt uitstekend op een 8088 XT met monochroom tekstscherm en een disk van 10 Mb, en dat is ook voldoende voor al de bovengenoemde toepassingen. Zo'n machine kost tegenwoordig minder dan duizend gulden.

Wil je wat meer mogelijkheden, zoals een tekstverwerker die meteen laat zien hoe het er later op papier uit zal zien en ben je op zoek naar een manier om te kunnen schakelen tussen een paar DOS-programma's, dan is DOS met Windows 3.1 een goede keuze. Ook al zit DOS je hier behoorlijk in de weg, het werkt allemaal aardig en zolang je niets bijzonders doet en de machine elke dag een keer opnieuw start is er weinig aan de hand.

Windows 3.1 heeft toch echt wel een 386-machine nodig, met echt minstens 4 MB RAM en minstens 60 MB disk en een SVGA kleurenmonitor. Een duizend of drie.

Mensen die echt betrouwbaar meer programma's tegelijk willen laten lopen en die behoefte hebben aan een stabiel platform, waar gewoon alles op draait en dat dan ook nog behoorlijk presteert komen uit bij OS/NT. Deze systemen hebben een 386 nodig met 8 Mb RAM en 80 Mb disk (OS/2) of een 486 met 16 Mb RAM, 200 Mb disk en een CD-ROM drive (Windows NT). Vier tot zevenduizend gulden.

Omdat een groter systeem ook altijd de kleinere systemen omvat (Windows omvat DOS, OS/2 omvat Windows en DOS), zou je kunnen stellen dat het hele verhaal simpel neerkomt op de vraag "wat wil je met je nieuwe computer gaan

doen?". Is dat WordPerfect, koop dan de goedkoopste XT of AT die je tegenkomt en plant daar DOS op. Wil je meer, zoek dan de software uit (bijvoorbeeld Word for Windows) en kies de bijbehorende machine.

Draai het niet om. Koop niet een of andere machine en ga dan kijken wat je wilt. Dit kan wel, maar is meestal onverstandig. Vaak kun je met minder toe dan de reclamewereld je wil doen geloven. Heb je echter het budget voor een grotere machine, of staat die er domweg al, snuffel dan eens aan de bijpassende software. Ga niet DOS draaien op een 486/33 met 8 Mb RAM en 100 Mb disk, want dat staat gelijk met boodschappen doen in een Ferrari Testarossa. Iedereen gaapt je na, maar eigenlijk sta je voor gek.

Jeroen Hoppenbrouwers
hoppie@kub.nl

- Advertentie -



- De Toekomst Van Videotex -

UTRECHT - Tienduizenden gebruikers werken inmiddels met videotex. Waarom? Videotex lijkt sterk op wat wij kennen als teletekst, alleen werkt videotex via de telefoonlijn. Met videotex kan men op eenvoudige wijze informatie raadplegen. En videotex werkt net zo gemakkelijk als de telefoon.

Iedereen heeft wel eens gehoord over telebankieren. Maar ook begrippen als teleleren, telewerken, teleconverseren en natuurlijk ook telepost en telewinkelen komen steeds vaker in het nieuws.

Grote bedrijven beschikken over vaak dure systemen om de videotex gebruikers van de juiste informatie te voorzien.

Met **QDR/ProTel** wordt uw computer een volwaardig videotex systeem.

Alle videotex-gebruikers die er in Nederland zijn kunnen via de telefoonlijn contact met uw pc opnemen. Simpel en geheel automatisch.

U bepaalt zelf wie er wel en wie er geen toegang tot uw systeem krijgt. Iedere gebruiker kunt u een eigen klantnummer en een pincode geven. En ... videotex is absoluut virus-vrij!

QDR/ProTel is met een prijs van f 99,- vele malen goedkoper dan vergelijkbare pakketten, maar u als TRON abonnee kunt nu voor slechts f 84,- het complete pakket QDR/ProTel 1.0 in huis halen!

Voor meer informatie over onze videotex-systemen, maar ook over de vele andere produkten en diensten waarmee wij u van dienst kunnen zijn, zijn wij te bereiken via ons telefoonnummer 030-881087.

Wielingenplein 17
3522 PC Utrecht
Telefoon 030-881087
Videotex 030-888943



DE 64-IN-1 DOOS

door Frits Kieftenbelt

De 64-in-1 doos heb ik ontworpen, omdat ik mijn beide 8-in-1 dozen vol had en ook het steeds wisselen van die twee beu was. Er kwam een idee in me op, om een 32-in-1 doos te maken en spoedig lag het proto-type op tafel. Al snel bleek dat er van een 32-in-1 doos met wat aanpassingen eenvoudig een 64-in-1 doos was te maken en aangezien er bij onze P2000-stand van de PTC-afd. Zwolle in Den Bosch nogal wat belangstelling voor was, besloot ik om de print als zelfbouw-ontwerp in TRON te plaatsen.

Weet u het niet meer?

Voor diegenen die niet weten wat een 64-in-1 doos (of 8-in-1 doos) precies is nog even in het kort: Zoals je weet, zijn er voor de P2000 diverse insteek-dozen voor sleuf 1. Zo heb je een doos voor BASIC-NL, een doos voor het Familie-Geheugen, enz. Een 64-in-1 doos kan al deze programma's in 1 doos hebben en door middel van schakelaars bovenop de doos kan het gewenste programma worden gedraaid. Zo is de P2000 in bijv. stand 1 een Elektronische Kaartenbak, in stand 2 een BASIC-computer, in stand 3 een PASCAL-computer, een Textbewerker of wat er zoal is. Een 8-in-1 doos kan maximaal 8 verschillende programma's hebben, terwijl een 64-in-1 doos dus... je raad het al. Je hoeft dus nooit meer sleuf 1 van doos te verwisselen, het is nu een kwestie van een schakelaar in de goede stand zetten.

Compleet te koop

De doos is ook compleet gesoldeerd te bestellen, de prijs is dan f 175,- exclusief EPROM's. Tel daarbij op de prijs van de EPROM's (f 65,- per stuk medio 1991) en je bladert meteen verder in TRON. Maar misschien valt het mee als je de doos dus zelf bouwt en begint met niet 4 maar met maar 1 EPROM. Je hebt dan geen 64-in-1 doos maar een 16-in-1 doos. Als je 16 programma's in de doos hebt (de EPROM is dan vol) dan koop je er een EPROM bij en dan heb je al een 32-in-1 doos. Bovendien is de prijs f 65,- nog van 1991 en aangezien dit een vrij nieuw type EPROM is zal de prijs al snel dalen. Het kan dus zijn dat het nu al een stuk goedkoper is. Bij aankoop van een EPROM is deze natuurlijk leeg. Tegen geringe vergoeding wil ik de programma's wel in de gewenste volgorde in de EPROM programmeren, maar als je zelf een programmer bij je P2000 hebt, wacht dan tot de volgende TRON! Daarin staat een bouwontwerp van een adapter-print die het zo maakt dat je deze grote EPROM's met een kleine programmer kunt programmeren.

Kosten

Kosten compleet gebouwd:

$$16\text{-in-1} = f\ 175 + (1 \times f\ 65) = f\ 240,00$$

$$32\text{-in-1} = f\ 175 + (2 \times f\ 65) = f\ 305,00$$

$$48\text{-in-1} = f\ 175 + (3 \times f\ 65) = f\ 370,00$$

$$64\text{-in-1} = f\ 175 + (4 \times f\ 65) = f\ 435,00$$

Solderen:

Nadat de dubbelzijdige print op de fotografische wijze uit TRON is overgenomen en geboord, moet voordat de componenten erop worden gesoldeerd de print worden doorgemetaliseerd.

Dit betekent dat we verbindingen tussen de boven- en de onderkant van de print moeten maken. Op de tekening van de componenten-opstelling zien we zo'n 26 kleine kruisjes (x) staan. Steek hier een draadje (met harde kern) door en soldeer het aan BEIDE zijden van de print vast. Doe dit bij alle 26. Als dit is gebeurd, kun je de diodes en weerstanden op de print solderen. Alle componenten dienen aan BEIDE zijden te worden gesoldeerd! Dan de vier 32-pins IC-voeten; deze moeten van het professionele type zijn (gedraaide bus-contacten), omdat de voeten aan zowel de printzijde als de componentenzijde worden gesoldeerd.

Bij low-cost IC-voeten is het solderen aan de componentenzijde niet mogelijk, daarom dus de professionele IC-voeten. Overigens vereist het solderen van de IC-voeten een dunne soldeerpoint, omdat het anders vrij moeilijk is om tussen 2 IC-voeten aan de componentenzijde te solderen. Maar het valt mee als je alleen DIE eilandjes soldeert, waaraan een baantje zit en door die eilandjes eerst te vertinnen voor de IC-voeten erop worden gezet, vloeit de tin makkelijk weg.

Dan solderen we de condensatoren en de 4 IC's erop (let op de + bij de tantaal-elco's en op het nokje bij de IC's). Tot slot moeten de duimwielschakelaars S1 en S2 nog met de print worden verbonden. De nummers 1,2,4 en 8 komen aan S1, waarop deze cijfers ook staan vermeld en de letter A van S1 komt aan de plus (+), de aansluiting hiervoor bevindt zich vlak boven C1. S2 heeft 3 draden, de nr's 1 en 2 op S2 komen aan resp. 1 en 2 op de print en nr. 5 op S2 komt aan de min (-), ook die aansluiting bevindt zich boven C1. De print is nu klaar. Let op, als de print zonder doos wordt getest, zorg er dan voor dat je de print in sleuf 1 steekt, met de componenten naar **ACHTEREN!**

Print in doos bouwen:

Als de aangegeven type duimwelschakelaars zijn gekocht, dan kunnen de volgende maten worden aangehouden, ervan uitgaand dat de lege insteekdoos, de doos is met een vlakke bovenkant en niet schuin zoals de PHILIPS of Miniware insteek-dozen. De breedte van de doos is 93 mm. Houdt de doos nu zo voor je dat de schroeven naar achteren zitten. Meet vanaf rechts 10 mm en zaag daar de doos in. Dan 30 mm verder nog een zaagsnede en dan kan dit gedeelte er met een kopkniptang worden uitgeknipt. In de 30 mm brede uitsparing komen de beide duimwelschakelaars te zitten. Schakelaar S2 (BCD 10 standen) kan voorzichtig worden open gemaakt en een vergrendeling worden aangebracht, zodat de schakelaar alleen de standen 1 t/m 4 kan bereiken.

Reeds gebouwde exemplaren zijn te zien op clubavonden van de GGPC afd. Noord en de PTC afd. Zwolle. Als je zo'n doos wilt bouwen, kun je dus het best eerst bij een van deze afdelingen gaan kijken.

Werking:

Met S2 (1-4) zet je 1 van de 4 EPROM's in werking en met S1 (0-15) selecteer je, welk programma er van die EPROM moet draaien.

Onderdelenlijst:

R1 = 1K8
R2 t/m R7 = 1K
C1,C2 = 1 uF tantaal(!)
C3,C4,C5 = 100n Sibatit

D1,D2 = 1N4148

IC1 = 74LS138

IC2 = 74LS14

IC3,IC4 = 74LS244

S1 = Bin 16 st.(0-15) 03.45.1627

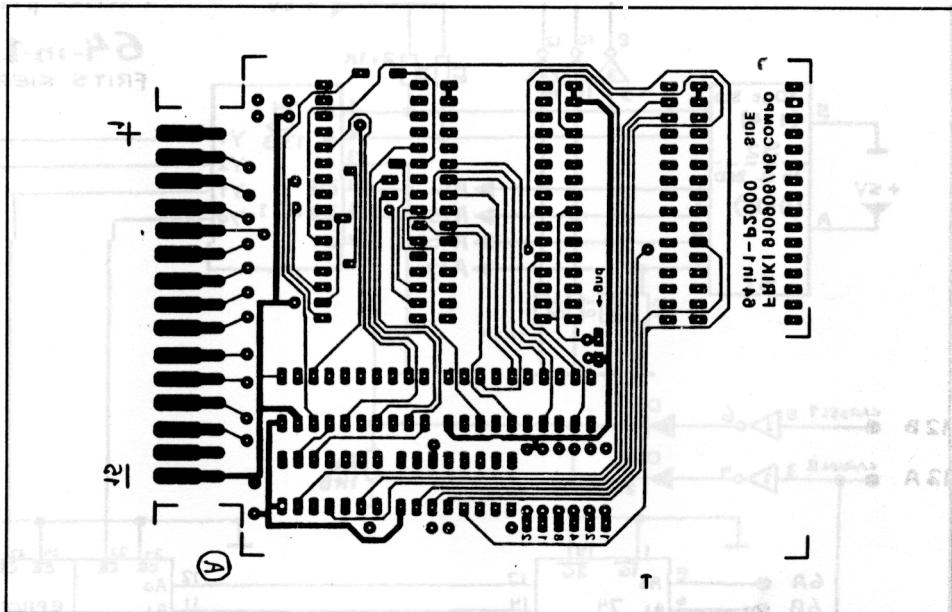
S2 = BCD 10 st.(0-9) 03.45.1731

4x 32 pins prof.IC-voeten (zie text)

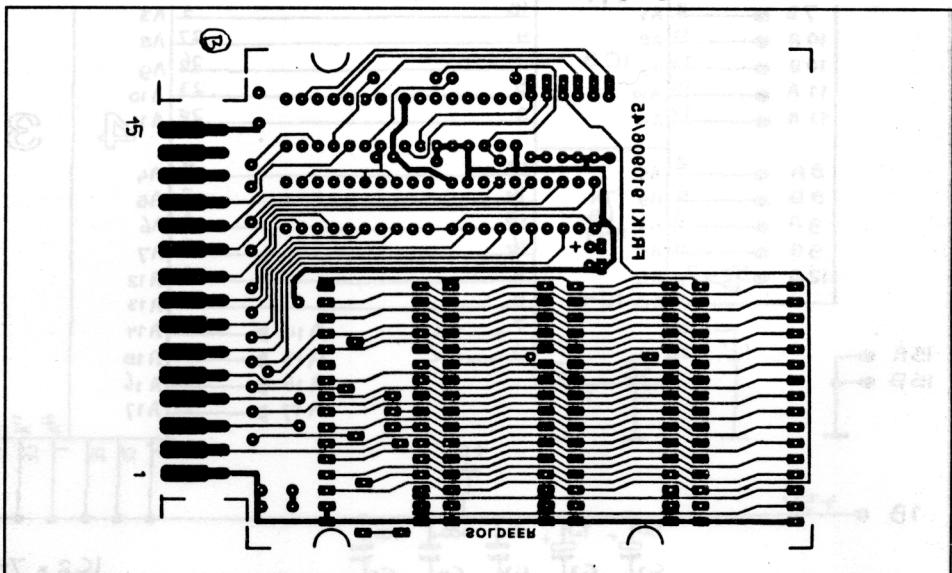
EPROM(s) type:27C2001A

1x lege kunststof doos voor sleuf 1

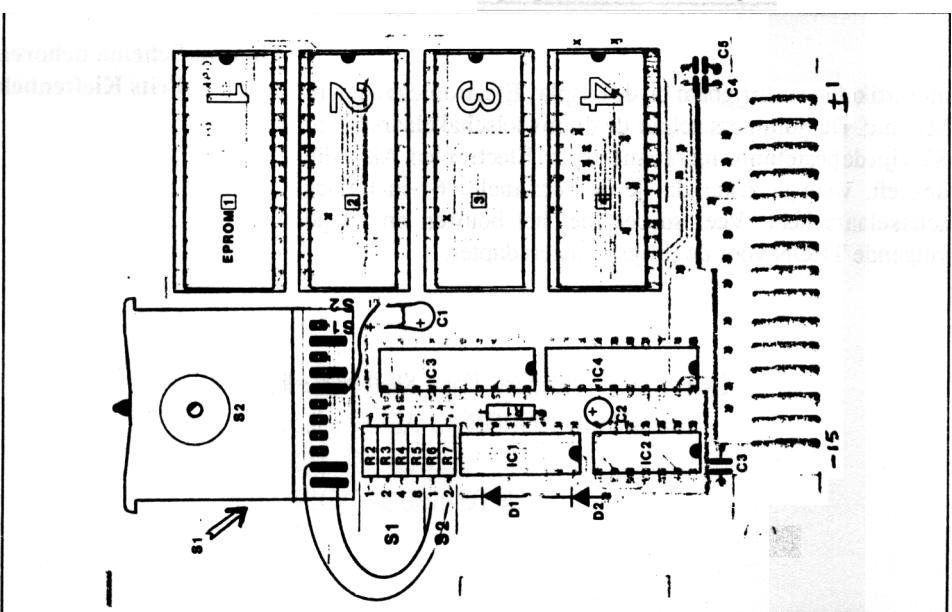
Op de lege kunststof doos na, zijn



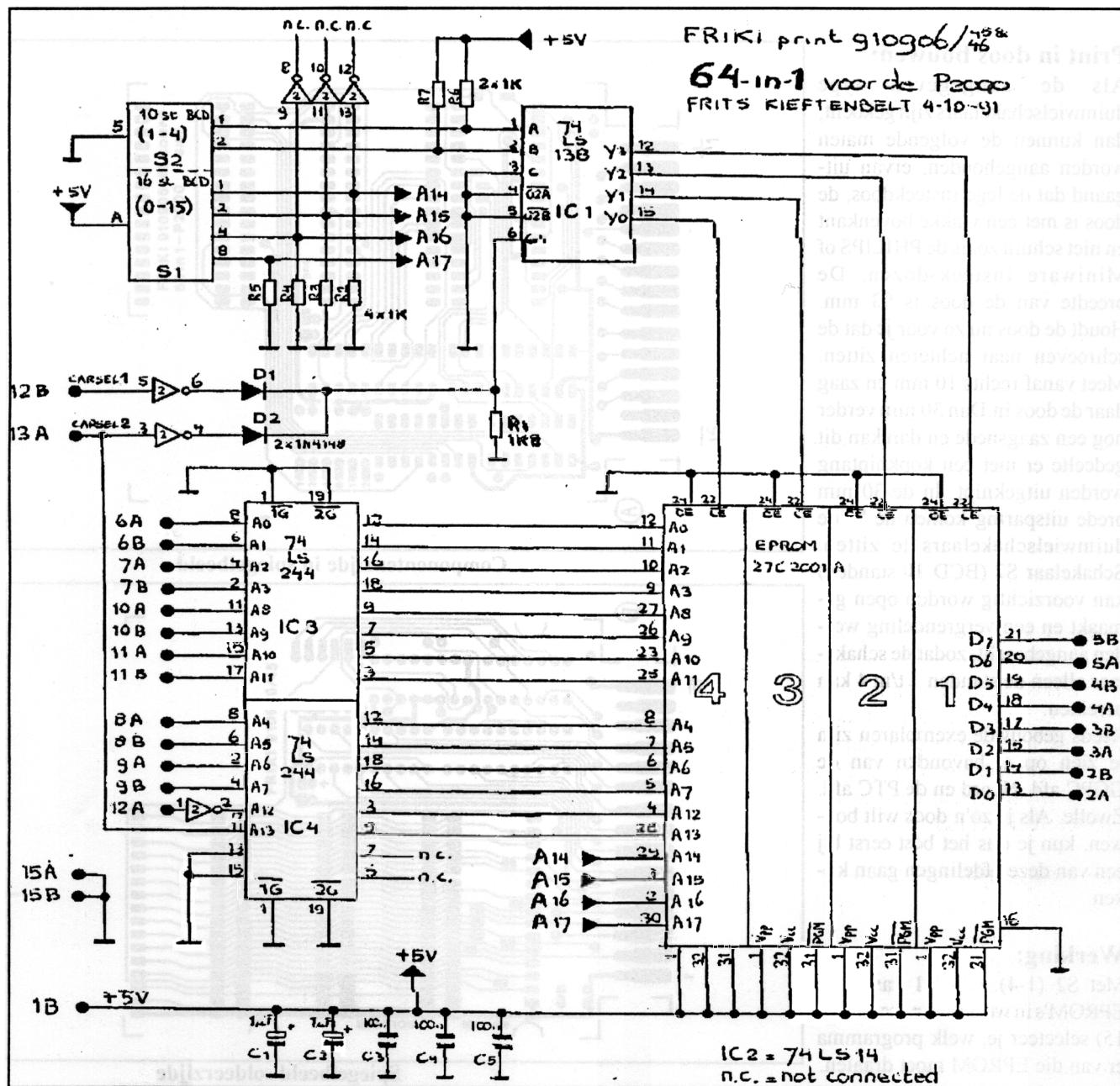
Componentenzijde in spiegelbeeld



Spiegelbeeld soldeerzijde



Componenten opstelling 64-in-1



MSX-SPOTLIGHT (2)

door Albert Beevendorp

Welkom bij de tweede aflevering van de TRON MSX-Spotlight.

Ik zal proberen om buiten de CV Waterland Spotlight om, apart een tekst samen te stellen voor TRON. De vorige aflevering werd aan de hand van diverse afleveringen uit Spotlights uit het CV Waterland Maandblad opgesteld. Toen werd er beloofd dat er de lijst met Fray haar DATA-disk zou komen. Nu zet ik deze lijst in deze aflevering van TRON. Deze lijst is overigens identiek aan de lijst die in het CV Waterland Maandblad is te vinden.

De vorige keer behandelde ik de DATA-disk van SD-Snatcher. Bij de Weapons-list stond een kolommetje (die ik apart had gezet) met SKILL per weapon.

Nu had ik daaronder (bij Adres 3:) \$40 gezet voor 100% Skill. Reken je terug van hexadecimaal naar decimaal met &H, dan kwam je uit op 64. Dat is dus géén 100%. Door nu voor 100% \$64 te schrijven op dat specifieke adresje, dan klopt het weer wel (&H64=100).

Verder kon je bij de OPTION/DRUG-list bij de DRUG een \$02 plaatsen. Dat klopt niet. Voor de DRUG mag je NOOIT (ik herhaal NOOIT) een \$02 zetten op Adres 1. Dat mag alleen \$00 of \$01 bevatten (niet hebbend of wel hebbend) en Adres 2 het aantal. Bij de OPTION mag het natuurlijk wel, daar deze in de wapenkuis in JUNKER kan worden verwisseld.

Wanneer je met de main-snatcher (het allerlaatste monster dat zich transformeert in een super-robot) wil vechten en daar heb je andere wapens voor nodig, dan zul je die hele route (inclusief het spiegelpaleis op de kermis) terug moeten lopen, om naar JUNKER terug te kunnen. Om daar af te saven en dan op de DATA-disk de wapens willen veranderen, dat lukt je niet. Deze truuk dacht ik zelf even te doen om met verschillende wapens de robot af te maken (klinkt vreemd). Maar ik kreeg steeds weer hetzelfde wapenpakket.

Now for thiz timez Spotlight:

1. De beloofde FRAY DATA-disk lijst.
2. BBS Noord-Holland
3. Easy
4. Diskette-opbouw
5. Software recensies.

*** DATA-disk voor FRAY ***

Beloof maakt schuld. Hierbij de lijst met FRAY-codes. De begin-sectoren voor de DATA-saves geef ik niet op, daar ik deze in mei '91 al had geplaatst en ik niet zo'n iemand ben, die dingen twee keer plaatst (tenzij er verbeteringen moeten worden gemaakt).

Gold wordt op adres \$029/\$02A bijgehouden.

<u>Equipment:</u>	<u>Adres:Item:</u>	<u>Adres:</u>
Magical rod 1	\$035 Bread	\$042
Magical rod 2	\$036 Meat	\$043
Magical rod 3	\$037 Bottle	\$044
Magical rod 4	\$038 Scroll1	\$045
Magical rod 5	\$03 Scroll 2	\$046
Magical rod 6	\$03A Scroll 3	\$047
Magical rod 7	\$03B Scroll 4	\$048
Magical rod 8	\$03C Scroll 5	\$04
	Scroll 6	\$04A
S ield 1	\$03D Scroll 7	\$04B
S ield 2	\$03E Scroll 8	\$04C
S ield 3	\$03F Scroll	\$04D
S ield 4	\$040 Scroll 10	\$04E
S ield 5	\$041 Scroll 11	\$04F
	Blue Bottle	\$054
nvullen	\$00 Niet in bezit	Bird ()\$055
	\$01 el in bezit	S ere \$056
ey		\$057
	nvullen aantal maximaal 55 (decimaal)	

*** Software recensies ***

Wat ik (nog) niet in het CV Waterland Maandblad doe, ga ik hier wel mee beginnen. Het MSX Software Team (dat door MSX Computer Magazine is opgezet) heeft diverse Software op de markt gebracht. Er zijn twee produkten van hen in het PD-circuit beland met de wil om het als een gek te verspreiden. Nu ben ik zo iemand, die daar absoluut voor is en daar hard aan meewerkt.

Ik spreek nu (uiteerdaard) over MemMan versies 1.0, 1.1, 1.2, 2.0, 2.1, 2.2, 2.3 en 2.31 en debijgeleverde bestandskopieerde genaamd BK versies 1.00 en 1.02. Alle versies van MemMan lager dan 2.0 zijn bedoeld om het Memory te Managen. De versies 2.0, 2.1 en 2.2 maken het mogelijk om zg. TSR's in te laden. Dit kon echter alleen nog onder MSX-DOS. Met versie 2.3 werd dit ook verleden tijd. Hiermee kunnen we ook TSR's in Basic inladen. Daar bleek echter ook nog een foutje in te schuilen, zodat versie 2.31 al spoedig een feit werd en ook als laatste standaard werd gehouden.

--- Wat is een TSR? ---

Voor iedereen die niet weten wat een TSR is, volgt hier de uitleg. TSR is de afkorting voor Terminate and Stay Resident. In het Nederlands zou dat zoëts zijn als: Beëindig en Blijf Aanwezig. TSR's zijn programma's, die op de achtergrond van het systeem werken en met een bepaalde toets-combinatie of commando een bepaalde functie starten, of juist met een menu'tje op het scherm komen. De meeste TSR's worden aan de zg. HOOK's gehangen.

HOOK's zijn stukjes geheugen, die in het Systeem-RAM aanwezig zijn. Al deze stukjes geheugen zijn 5 bytes lang en worden op een bepaald moment door de BIOS aangeroepen. Voordat een BIOS-call met zijn werk verder gaat, wordt er een sprong naar het Systeem-RAM gemaakt. Welk stukje er wordt aangeroepen, hangt af van de aangeroepen BIOS-call. Er zijn ook HOOK's, die door de Basic-interpreter worden aangeroepen en door het Operating System.

Een veel gebruikte HOOK is de TIMINT-HOOK. Deze HOOK 50 (of 60 in Japan) keert aangeroepen. Hierop zit op een Philips MSX2 met diskdrive bijvoorbeeld de sprong naar de routine, om de disk-drive motor uit te zetten. Ook wordt hierop een routine voor de achtergrondmuziek gezet, omdat muziek nu eenmaal op de maat moet blijven lopen.

Een paar goede voorbeelden van TSR's zijn: RAMdisks, printerbuffers, Screensavers, Alarm Systemen.

BK is een bestands-kopiëer-programma, dat programma's van verschillende diskettes in het geheugen opslaat en deze vervolgens naar een nieuwe disk kan kopiëren. Het programma werkt onder zowel MSX-DOS versie 1 als 2. Onder DOS2 wordt door het programma van de diskette in de huidige drive een TREE van de directories gemaakt, om ook in Subdirectories te kunnen laden en saven. In de volgende TRON MSX-Spotlight worden o.a. Improve (een disk(ette) optimizer naar een idee van ondergetekende), Tracer TSR en jANSI TSR besproken. Nu bezit ik deze programma's nog niet, maar daar komt op de beurs in Zandvoort een einde aan.

*** BBS Noord-Holland ***

Momenteel kunnen er nog geen MSX-Computers op 1200/75 in BBS Noord-Holland inloggen. Wanneer dit mogelijk is, is mij nog een raadsel. Voorlopig zal men op 300/300 moeten

inloggen. Tijdens de introductie werd 1 nummer voor het BBS Noord-Holland opgegeven. Ik had het nummer gegeven dat het makkelijkst was te onthouden. Het andere nummer om in het BBS Noord-Holland te komen is (02990-21899). De modem aan dit nummer heeft nogal wat kuren, wat betreft inloggen. Vandaar dat ik deze destijds niet had opgegeven. Verwacht wordt, binnenkort een modem met 1200/75 op het nummer 02990-21899 te hebben (speciaal voor de MSX-ers) dat uiteraard ook op 2400/2400 kan draaien voor de MS-DOS gebruikers. Het nummer 02990-71007 zal meer modems krijgen en alleen XX/XX inlogbaar zijn. Zelf log ik altijd op MS-DOS PC in en dan ga ik de MSX-files plaatsen (compatibiliteit ten top). Nogmaals, wanneer en op welke nummer het mogelijk is om op 1200/75 in te loggen, dat weet ik nu (nog) niet.

*** Easy ***

Wie het programmapakket 'Easy' bij MK (of HSH) heeft besteld en ontvangen, kent het wel. Deze aanbieding geldt alleen voor gebruikers met de ORIGINELE Easy-schijf. In dit pakket heb ik twee foutjes gevonden.

1. De Calendar werkt niet goed op MSX2-computers met de volgende datum-formatering:
MM-DD-YY en YY-MM-DD.
2. In de File-Manager werkt de Edit optie niet.

Gebruikers van Easy, die de werkende versies willen, dienen hun ORIGINELE diskette naar mij te sturen, met een aan zichzelf geadresseerde en voldoende gefrankeerde enveloppe. Ik zal de diskette kosteloos patchen en terugsturen. Ik zal MK op de hoogte stellen van de gemaakte veranderingen, met het verzoek om deze fouten bij iedere gebruiker te verbeteren.

*** De diskette-opbouw ***

WAARSCHUWING!

Alles wat er op een diskette wordt veranderd, is op eigen risico. U mag mij niet aansprakelijk stellen wanneer een diskette met belangrijke programma's niet meer kan worden gelezen. Aangeraden wordt een sector-kopie van de te editten disk te maken. Mocht het fout gaan, dan is er altijd nog een kopie beschikbaar die weer teruggekopieerd kan worden.

De meeste diskette-gebruikers weten wel dat een diskette in sporen (tracks) en sectoren (sectors) wordt ingedeeld. Wat deze gebruikers wellicht nog niet weten is, hoe de computer die diskettes weet te lezen.

We gaan uit van een 3.5" 2DD diskette:

Voordat de computer een diskette kan lezen en/of schrijven, dient die diskette te worden voorzien van een indeling. Dit noemen we FORMATTEREN. Of een diskette wordt ingedeeld van buiten af of van binnen uit, zal ik maar buiten beschouwing laten. Eerst wordt de diskette boven, daarna aan de onderzijde ingedeeld. Deze techniek levert de grootste tijdsvermindering op. Wanneer eerst boven en dan onder zou worden gedaan, zouden we als gebruiker knap gek worden van het lange wachten om zo'n diskette voor ons systeem (MSX(X)-

DOS compatible) leesbaar te maken. Na het indelen wordt de diskette voorzien van een sector met formateergegevens. In die gegevens staan:

De MEDIA (formateer) code: \$F9 (80 tracks, 9 sectoren)

Aantal sectoren	:	1440
Aantal bytes per sector	:	512
Aantal sectoren/track	:	9
Aantal sectoren/cluster	:	2
Aantal FATs	:	2
Aantal sectoren per FAT	:	3
Aantal koppen	:	2
Aantal entry's in de dir	:	112
Zg. OEM naam	:	NMS 2.0

Het onderste gegeven zal de naam van de ROM bevatten waar die disk mee is geformatteerd. In dit geval op een NMS 8250 van Philips. In het rijtje zijn overigens nog gegevens weggeleggen die wel degelijk op een disk aanwezig zijn, bijv. 'Aantal hidden sectors'.

De volgende sectoren (1 t/m 3) bevatten de 'File Allocation Table'. In deze sectoren wordt het huishoudboekje van de diskette bijgehouden. Wanneer de FAT 'leeg' is, wordt de disk als leeg beschouwd. Als er op die disk bestanden stonden, zijn deze niet meer te lezen. De FAT is een hoofdstuk apart en zal in een volgende aflevering worden besproken.

Na de FAT zitten we in de 'Directory' (7 t/m 13). Hier komen de namen van de bestanden, met hun specifieke gegevens, te staan: **Naam Extensie Lengte Datum Tijd Attributen Beg.cluster**

De naam, extensie en lengte spreken voor zich; de datum en tijd zijn de gegevens wanneer en hoe laat die file is weggeschreven. De attributen worden onder DOS2 (en de misschien latere versies) gebruikt. Hier worden de Hidden, Read-Only, System, Volume name, Sub-directory en Archive bijgehouden. De diverse attributen worden in een BIT (1/0) aangegeven. Dit zal ik, net als de FAT, in een volgende aflevering bespreken. Het allerbelangrijkste is, in welk cluster het bestand begint. Daarmee wordt in de FAT het aantal 'clusters' gezocht en het bestand ingeladen.

Bestanden kunnen ook worden verwijderd. Hiertoe komt op het eerste adres de code \$E5 te staan. De FAT wordt aangepast door de specifieke clusters weer vrij te maken. Het einde van de lijst wordt met \$00 aangegeven, of als alle plaatsen in de directory bezet zijn.

Na de directory (vanaf sector 14) komen we in het DATA-gebied.

Editten van sectoren

Om op een diskette te kunnen editten, hebben we een diskmonitor/editor nodig. Zelf gebruik ik DISKVIEW uit MCM/MST of een zeer uitgebreide diskdictor om in de FAT te veranderen. Laten we eens kijken naar Xak-serie (xak1, Xak2, Fray, Xak3, allen voor MSX2):

Xak1 zet zijn spel-data vanaf sector 1424. Vanaf daar steeds 1 cluster (=2 sectoren) verder de volgende data-entry. Voor de andere zal ik het geheel hier ook plaatsen:

Naam:	Data sector:							
	1	2	3	4	5	6	7	8
Xak1	1424	1426	1428	1430	1432	1434	1436	1438
Xak2	1400	1404	1408	1412				
Fray	1400	1402	1404	1408	1410	1412	1414	1416
Xak3	1400	1404	1408	1412				

Door nu op verschillende data-codes de spel-situatie weg te schrijven en voor dat spel die data-code aan te houden, kunnen we op slimme wijze de spel-situatie van de hele serie op 1 data-disk zetten en bovendien nog programma's ook!

Zelf heb ik Xak1 op data-code 4 en 5 gesaved, Xak2 op data-code 1, Fray op data-code 3, Xak3 op data-code 3, dan zal het geheel netjes op verschillende sectoren worden gesaved. De slimmerik zal hebben gezien dat alle data-codes voor Xak1 kunnen worden gebruikt.

Door naar bovenstaande sectoren te gaan, kun je met de diskmonitor de waarden aanpassen. Op die manier kun je zelf een data-disk aanmaken.

Voor Xak1 zit dat zo in elkaar:

\$000	bevat de code voor disk 1 (\$00) of 2 (\$05)
\$001-\$008	bevat de naam van de speler in ASCII code
\$013-\$048	wapens (0, niet in bezit, 1, wel in bezit)
\$04F-\$050	items (0, niet in bezit, elke andere waarde geeft aantal)
\$05D	level (0 tot 25)
\$062-\$063	gold in hexadecimale code
\$07F-\$080	exp in hexadecimale code

Nog 1 dingetje. BBS Noord-Holland is voorzien van een MSX-gebied en raad eens wie daar het beheer over heeft?

Tot de volgende keer, jullie mede-MSXer,

Albert Beevendorp
Maastricht 90
1442 RW Purmerend