ADVERTISEMENTS

If you have something to sell please give details and your ad will be carried. Due to the tiny circulation at present, no charge will be made, and probably never for private members. If you want to do a full page ad, please submit it on the paper size that this newsletter is printed on (9.5" x 11.5") Please do the photocopying yourself and send me enough copies for the next issue. (Phone for an estimate). However you may wish to sell your product by writing an article about it, in which case it should be submitted in the normal way. You can also lend your wares to someone else, for the purpose of writing a review.

Inclusion of an article for sale in these pages does not mean endorsement by the Group, and all transactions to be conducted directly between vendor and purchaser (except where outsiders offer club discounts in which case special arrangements will be made).

HINTS & TIPS

This section is for small items discovered that can be useful in program writing and use. I will start the ball rolling this time with some of my own. Please send your own bits and pieces for this page.

*Many of your programs will display some text, and wait for the spacebar to be pressed before moving on. Instead of the typing: 100 G=GETC:IF G<>32 GOTO 100 just use CALLM #D6DA which has the same effect.

*If you want to disable the whole keyboard during a BASIC program, including the BREAK key, just POKE #5F,#84. The keyboard will re-enable at the program end, or when you POKE #5F,#C4.

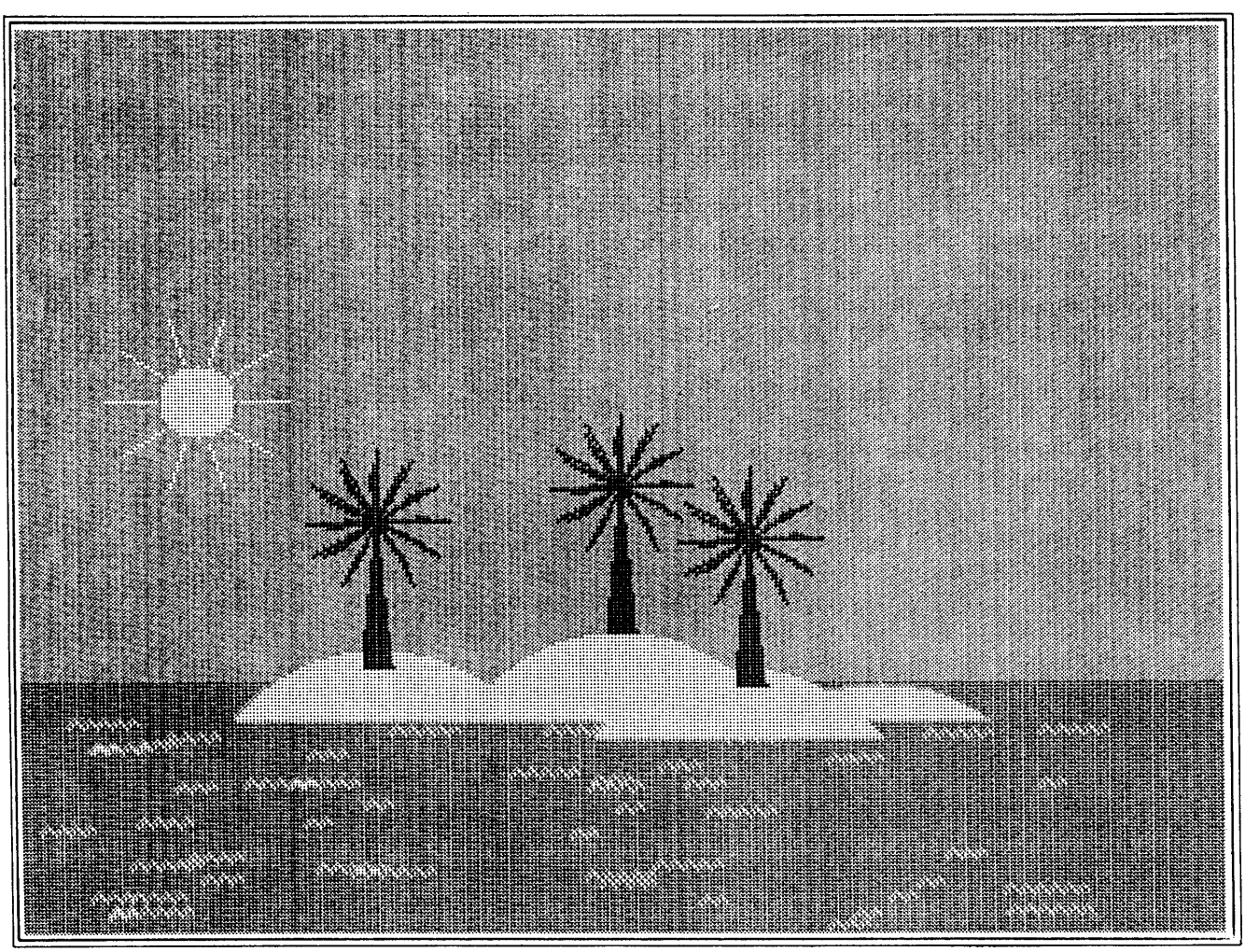
*A common feature of programs is to test GETC for a particular key but to disregard whether it is upper or lower case. Rather than G=GETC:IF G=#41 OR G=#61 GOTO ... to test for the 'A' key, just use G=GETC:IF G IOR #20=#61 GOTO ... The IOR converts the value to lower case if it is upper. (i.e. sets Bit 5).

*For a neater printed menu etc, blank out the screen i.e with COLORT 8 8 8 8 before your printing lines, then reset the colours i.e. COLORT 8 0 8 0 after the printing, for an apparently instantaneous image.

*You will know that POKE #75,32 blanks out the cursor by making it a SPACE character. However this is not always satisfactory as if the cursor slides over screen area that already contains characters, it can be seen momentarily blanking out the characters. To avoid this, ignore the number in #75 and POKE #74,0. Your 3rd and 4th COLORT must be the same as the first two. i.e. COLORT 8 0 8 0. The cursor will then truly disappear. To see the difference try sliding the cursor over a line in the editor in each mode.

```
100   REM *** GROETEN UIT HAWAI ALOHA OE *********
110   REM *** GESCHREVEN DOOR : DE BONT CORNEEL **
120   REM ********************** OOSTEINDE 10 *****
130   REM ***************** 2338 BAARLE-HERTOG ****
140   REM ********************** 28 - 1 - 1983 ****
150   REM ****************************************
160   GOTO 1000
200   REM *** HELE OF HALVE CIRKEL
210   FOR Z=D1 TO D2:S=SQR(D2*D2-Z*Z)
220   DRAW XC-S,YC+Z XC+S,YC+Z 22:NEXT:RETURN
300   REM *** GOLVEN
310   NG=INT(5.0*RND(1.0))+1.0
320   FOR X=0.0 TO NG:DOT XG,YG-1 22:XG=XG+1.0
330   DOT XG,YG+0 22:XG=XG+1.0:DOT XG,YG+1 22:XG=XG+1.0
340   DOT XG,YG+0 22:XG=XG+1.0:NEXT:RETURN
400   REM *** PALMBOOM
410   DI=8.0:FOR Y=YP TO YP+40.0:DRAW XP,Y XP+DI,Y 21
420   DI=DI-0.2:XP=XP+0.1:NEXT:ST=PI/60.0
430   FOR X=0.0 TO 2.0*PI STEP PI/6.0
440   CO=COS(X):SI=SIN(X):DRAW XP,Y XP+20*CO,Y+20*SI 21
450   CO=COS(X+ST):SI=SIN(X+ST)
460   DRAW XP+3.0*CO,Y+3.0*SI XP+17*CO,Y+17*SI 21
470   CO=COS(X+ST+ST):SI=SIN(X+ST+ST)
480   DRAW XP+6.0*CO,Y+6.0*SI XP+14*CO,Y+14*SI 21
490   NEXT:RETURN
1000  REM *** INLEZEN NOTENARRAY
1010  CLEAR 3000:DIM F(60.0)
1020  MODE 0:PRINT CHR$(12):POKE #75,32:RESTORE:SOUND OFF
1030  ENVELOPE 0 15
1040  ENVELOPE 1 12,5;1,10;0
1050  SOUND OFF :NOISE OFF :PRINT CHR$(12);:COLORT 9 14 9 9
1060  COLORG 12 5 14 9:MODE 6A:POKE #FF05,2
1070  POKE #7556,222:POKE #74D0,211
1080  POKE #744A,213:POKE #73C4,220
1090  FOR I=141.0 TO 211.0:J=#BFEF-I*90.0-2.0:POKE J,119
1100  POKE J-1,143:POKE J-86,255:POKE J-87,255:NEXT
1110  FILL 0,0 335,70 23:I=0.0
1120  CURSOR 11,3:PRINT "GROETEN UIT DE STILLE ZUIDZEE."
1130  XG=INT(300.0*RND(1.0)):YG=INT(60.0*RND(1.0))+2.0
1140  GOSUB 300:G=GETC:IF G<>0.0 THEN 1160
1150  I=I+1.0:IF I<50.0 THEN 1130
1160  CURSOR 16,2:PRINT "GROETEN VANUIT HAWAI"
1170  D1=25.0:D2=50.0:XC=167.0:YC=35.0:GOSUB 200
1180  D1=30.0:D2=50.0:XC=100.0:YC=30.0:GOSUB 200
1190  D1=30.0:D2=50.0:XC=200.0:YC=25.0:GOSUB 200
1200  D1=40.0:D2=50.0:XC=240.0:YC=20.0:GOSUB 200
1210  D1=-10.0:D2=10.0:XC=50.0:YC=150.0:GOSUB 200
1220  FOR X=0.0 TO 2.0*PI STEP PI/5.0:S=SIN(X):C=COS(X)
1230  DRAW 50+12*C,150+12*S 50+25*C,150+25*S 22
1240  NEXT
1250  XP=200.0:YP=70.0:GOSUB 400
1260  XP=164.0:YP=85.0:GOSUB 400
1270  XP=96.0:YP=75.0:GOSUB 400
1280  CURSOR 14,1:PRINT "OF ZOALS MEN DAAR ZEGT :"
1290  CURSOR 21,0:PRINT CHR$(34);"ALOHA OE";CHR$(34);
1300  SOUND OFF :NOISE OFF :POKE #FF05,255
1500  REM *** MUZIEK
1510  RESTORE:FOR I=0.0 TO 57.0:READ F(I):NEXT:TE=7.0
1520  FOR I=1.0 TO 84.0:READ T,W:S=F(T)
1530  SOUND 0 0 15 0 FREQ(S*4.0):SOUND 1 0 15 0 FREQ(S)
1540  SOUND 2 1 15 0 FREQ(S*2.0):WAIT TIME TE*W:NEXT
1550  SOUND OFF
1560  FOR I=0.0 TO 3.0:CURSOR 0,I:PRINT SPC(58);:NEXT
1570  POKE #FF05,2:CURSOR 27,0:PRINT "EINDE";:CURSOR 27,1
1580  PRINT "ALOHA":CURSOR 25,2:PRINT "TOT ZIENS"
1590  POKE #75,95:POKE #FF05,255:CURSOR 0,3
```

```
2000  REM *** NOTEN VAN 4 OKTAVEN F(57)
2010  DATA 20000,20000
2020  DATA 065,069,073,078,082,000,087
2030  DATA 092,098,104,110,116,123,000
2040  DATA 131,138,147,155,165,000,175
2050  DATA 185,196,208,220,233,247,000
2060  DATA 262,277,294,311,330,000,349
2070  DATA 370,392,415,440,466,494,000
2080  DATA 523,554,587,622,659,000,698
2090  DATA 740,784,831,880,932,988,000
2100  REM *** F,T (5 * PER REGEL)
2110  DATA 24,2,30,2,34,2,34,4,32,2
2120  DATA 30,3,28,1,30,2,26,2,24,8
2130  DATA 24,4,00,2,34,2,34,2,32,2
2140  DATA 32,4,31,2,32,2,30,2,36,3
2150  DATA 34,1,32,8,32,4,00,2,24,2
2160  DATA 30,2,34,2,34,4,32,2,30,3
2170  DATA 28,1,30,2,26,2,24,8,24,4
2180  DATA 00,2,30,2,28,2,26,4,32,2
2190  DATA 30,2,28,4,34,2,32,2,30,8
2200  DATA 30,6,00,2,24,2,26,4,30,4
2210  DATA 36,6,26,2,24,4,30,4,34,6
2220  DATA 00,2,30,2,28,3,26,1,28,2
2230  DATA 30,2,32,2,32,2,36,4,34,8
2240  DATA 30,6,00,2,24,2,26,4,30,4
2250  DATA 36,6,26,2,24,4,30,4,34,6
2260  DATA 00,2,30,2,28,6,30,2,34,2
2270  DATA 32,4,28,2,30,8,30,4,00,2
2280  REM --------- EINDE ---------
```



```
*****************************************
*                                       *
*  S P L : A NEW ASSEMBLER FOR THE DAI PC  *
*                                       *
*****************************************
```

As announced in DAInamic 13, workgroup SPHYNX (located in the Netherlands) has designed an assembler with many special features. SPL was created exclusive for the DAI pc with the intention to allow easy assembly of large machine language programs.

1) FEATURE LIST

1.1) MEMORY USAGE

Unlike DNA the SPL is a lot more economical with memory. This is accomplished by using a single buffer storage and a special format for source code store.
8080 instructions are stored as single bytes codes. Symbols are kept in a symboltable and in the sourceline only references into the symboltable are included (this is the same method as used by DAI basic). The only thing which takes as many bytes as characters shown on a listing are comments.
As illustration the whole source for SPL (which generate 12 kbyte code) can be loaded, editted and assembled with SPL.

1.2) CONDITIONEL MACRO ASSEMBLER

SPL is a conditionel macro assembler.
Conditional assembly means that some parts of the source code are ignored during assembly. Control of this is done by using IF ELSE ENDIF constructs (which may be nested). As condition any comparation may be used. Purpose of conditional assembly is to generate more than 1 program version with the same source (eg make a game with keyboard or paddle input, or include multi-language menu's in the source and select 1 language at assembly time).
Macro assembly can be used to declare a sequence of mlp instructions, which perform a given function, only 1 time in the source. Each location where the function is needed in a program, only a macro request is included and SPL will insert the sequence of instructions during assembly. An example could be the macro PSHALL, which references the sequence PUSH PSW, PUSH B, PUSH D, PUSH H.

1.3) SOURCE EDITOR AND HANDLER

Full-scale editor and source handler, with following commands :
- source code list from whole or part of the source, where selection can be done using linenumbers or/and labels
- full-screen editing (DAI editmode), selection idem as for list ; on exit of editmode a check is performed to catch typing and syntax errors
- MOVE, COPY and KILL commands for sourcelines
- replacing a symbol with a new symbolname
- find in which lines a label or a symbol is used
- READ, WRITE and MERGE commands
- configure memory-map (dimension and location of source-buffer)
- display memory-map to verify spare area source-buffer
- inline call from all dcr commands
- configure hardcopy output (lines/pages) and send control characters to a hardcopy device

1.4) ASSEMBLY COMMANDS

Assembly monitor with following commands :
- object list (hex + source), with LST UNL control
- poke object code (offset possible)
- generate an error list (eg not-defined symbols)
- symbol table list
- and other commands to help program design and test

Due to the way SPL stores his source, assembly runs very fast.

1.5) SPL LANGUAGE

- SPL uses the normal INTEL 8080 mnemonics for opcodes and pseudo-operations
- SPL implements a large subset of the MACRO-80 (Mikrosoft) or ISIS-II 8080 MACRO-ASSEMBLER (Intel) features
- operators for operands : add, subtract, dived, multiply, modulo, shift, and, or, xor
- operators for conditions : <, >, =, <>, or, and
- all operators are single characters (eg * " #)
- symbols may be 6 characters long and lower-case characters must be used for naming macros

2) REQUIREMENTS

2.1) MEMORY AND ROM VERSIONS

SPL as delivered will only run in DAI systems with 48k RAM.
Generic code of SPL starts at hex address B400 and ends nearly at begin of screenram. When using SPL switching to a graphic mode is not possible, as it would destroy the SPL code.
A version for 32k RAM will only be supplied if sufficient requests reach us.
SPL was tested with ROM V1.0 and V1.1.

2.2) PERIPHERALS

SPL is tested with DCR software DTOS1.2.
Due to location of SPL no memory overwrite problem should exist for using a floppy system which uses lower part of RAM to store the disk-operating system. However as no high-quality floppy system is available until now, Dainamic had not tested SPL-floppy interface.
As delivered SPL will use the RS232 output to drive a printer, but SP can be configured to drive a parallel printer interface (interface description in the manual).

3) EVALUATION

3.1) BY DAINAMIC MEMBERS

In the past 2 months SPL was distributed to a selected group of members, this to perform a pre-release test and evaluation.
Reactions were channeled to the designers and updates were made to code or/and manual.
In general the results of the evaluation were very good and we think SPL will became an important enhancement for machine language programming on the DAI pc.

3.2) COMPARE WITH MACRO-80

For those members who know the MACRO-80 assembler, an (incomplete) list of M80 features not implemented in SPL :

- ASEG, CSEG, DSEG location counters
- EXTRN, PUBLIC, COMMON, NAME , .REQUEST : SPL doesn't generate linkable object modules
- IFDEF, IFB special conditional tests
- IRP, IRPC repetition controls (normal REPT is implemented in SPL with ### pseudo)
- LOCAL labels in macros (SPL uses a kind of SET pseudo to avoid multi-defined errors, however this seems not to have some restrictions)
- parameters for macros cannot be register names

NOTE : MACRO-80 is an extremely powerfull assembler which requires a floppy-system (eg CP/M) and which is sold for about 10.000 Bfrs. Also MACRO-80 does only the assembly job. Sourcecode editting and objectcode linking must be done with other programs.
Most of the above mentioned differences can be overcome by using some other instructions. Problem of module linking is not so important, because most applications will fit in 1 SPL source file (see 1.1).

3.3) SOME POINTS ABOUT USE

Numeric constants must always be entered with a B, D or H appended, no default is provided. In a list appearing of these constants in decimal or hexadecimal format must be controlled by using PUT directives.
Due to single character commands (the whole alfabet including some lower case characters) some skill is needed to use them without looking back at the manual.
Same thing applies for single character operands and meaning of single or double character error reports.

3.4) CONCLUSIONS

SPL is a good compromise between a professional full-line assembler and a system environment as the DAI pc.
SPL don't require floppy's to retrieve records of source but allows that the source is stored in ram in a compacted format.
Used solutions to minimalise SPL generic code (such as single character commands and operators) are easy to live with.

4) PACKAGE

SPL is delivered with a dutch and english manual.
Included on tape are following source-programs :

- IMPLEM : to patch SPL (eg parallel printer, screen colors, lines/page)
- RESTORE XXXX : example SPL source
- DISPL : a very powerfull 8080 disassembler which directly generates SPL source

Planned to be included :

- CONVERT : convert DNA source to SPL format

It is also planned to make the SPL source listing available on request.

5) DELIVERY

PRICE : audio tape : 1.100 Bfrs
dcr tape : 1.250 Bfrs
SCHEDULE : delivered from march 83 on

SELLING RIGHTS AND ONLY PLACE FOR ORDERS :
DAInamic vzw Westmeerbeek Belgium

Geachte heren,

Met belangstelling heb ik het artikel FGT-DISK-PEEK-POKE in nummer dertien gelezen.De methode die de heer Gesp gebruikt vind ik nogal omslachtig en maakt geen gebruik van de mogelijkheden die het DOS ons bieden.

Wanneer we het FGT programma deel laten uitmaken van het DOS zal dit bij het inschakelen samen met het DOS ingeladen worden en zal er blijven zitten zolang we willen.Ook brengt het splitsen van het programma voordelen met zich mee.Het is immers enkel de tabel die veranderd als we een ander lettertype willen.Wanneer we van de tabellen D-files maken kan een programma over meer dan een lettertype beschikken.

Voor we zover zijn moeten we echter een beetje goochelen maar er zijn geen trucjes mee gemoeid.

Het nieuwe DOS bestaat uit :

1. Het originele DOS	Van 02E3 tot 19CB
2. Het FGT programma	Van 19D0 tot 1C05
3. De FGT tabel	Van 1C10 tot TBLEND
4. Vrije ruimte	Van TBLEND tot 27E0

TBLEND is het einde van een FGT tabel.
De vrije ruimte is nodig om ook de langste tabel in het DOS te laten passen.TBLEND van de langste tabel is dus 27E0.
Wanneer het nieuwe DOS geladen is zal de HEAP op 27E1 beginnen.Er is dus nog plaats voor een fors programma.
Op de systeem schijf staat buiten het DOS ook de verschillende tabellen zodat ze beschikbaar zijn voor de programmas die er gebruik van gaan maken.

Hoe gaan we nu te werk ?
Na HARD=RESET laden we de FGT met de langste tabel en gaan met UT kijken waar deze eindigd.Daarna laden we de FGT met standaard tabel en gaan nu het geheel van DOS en FGT op een systeem-schijf zetten waarvan $MSTRDOS is ge-deleted.
Tik in : DSAVE $MSTRDOS:0 2E3 27E0 13EC [RET]
en onze nieuwe DOS is klaar.Het adres 27E0 kan in uw geval anders zijn,het hangt van de lengte van de door u als langste tabel beschouwde tabel af en van de locatie van uw FGT.
We laden nu achtereenvolgens de programmas met de verschillende tabellen en DSAVEn enkel de tabellen op de schijf met een passelijke naam.

Hoe laad een programma zijn tabel ?
We gebruiken hiervoor het commando DLOAD maar plaatsen het tussen de gebruikelijke poke's.Voorbeeld :

```
100   POKE#131,3:PRINT "DLOAD MATH:0":POKE#131,1
```

Deze programma regel hoeft slechts eenmaal uitgevoerd en kan desgewenst in een zelfde programma herhaald worden om over een ander lettertype te kunnen beschikken.Het ganse programhoeft niet langer in een array geladen te worden en op z'n plaats gePOKEd te woden :tijdwinst en plaatswinst.

Om het geheel te proberen stel ik voor dat u nu een beetje speelt met de demo-programmas die aangepast zijn en eveneens op deze schijf staan.

Met de meeste hoogachting,

Couwberghs Frans
Boekdonkstraat 13
3980 Tessenderlo
013/666340

program identification

title	:	DAI-FLOPPIES (the "old" ones)
author	:	Mr. Baptiste
purpose	:	Explains how to solve some DOS-problems
comment	:	This information should be in the manuals !

APRES 8 MOIS DE TRAVAIL SUR DISK ET PLUS SPECIALEMENT EN TRAVAILLANT SUR DES FICHIERS, JE VOUS LIVRE MES CONCLUSIONS.

UNE DISQUETTE DOIT ETRE STOCKEE A L'ABRI DE LA POUSSIERE, DANS UNE BOITE FERMEE,PAS A PROXIMITE D'UN CHAMP MAGNETIQUE (ENREGISTREUR,HAUT PARLEUR,ARRIERE DE L'ORDINATEUR,SUR L'UNITE DE DISK CAR UN TRANSFO SE TROUVE A DROITE). J'AI PERSONNELLEMENT PERDU 1 DISQUETTE EN LA DEPOSANT SUR LE DISK DRIVE A L'ARRIERE DROIT OU SE TROUVE LE TRANSFORMATEUR.

QUAND LE DISK EST FROID, IL PEUT PROVOQUER DES RATES DE LECTURE .LES RATES SONT PLUS FREQUENTS DANS LES LECTURES DE FICHIERS QUE DANS LES LECTURES DE PROGRAMMES.
POUR EVITER LES ENNUIS,IL VAUT MIEUX CHAUFFER LES AIGUILLES EN EFFECTUANT UN VERIFY SUR LE DISK 0 ET DISK 1 AVANT DE TRAVAILLER.

L'ADRESSE #9B7 SIGNALE SI UNE ERREUR A ETE COMMISE SUR LE DISK DRIVE.MAIS AVANT TOUTE INSTRUCTION, ON DOIT METTRE LE BYTE #9B7 A 0;LE DOS NE LE FAIT PAS LUI-MEME.

LE NOM DES FICHIERS NE PEUT CONTENIR DE BLANCS, CAR LE DOS ARRETE SA LECTURE DES LE PREMIER ESPACE ET IGNORE LES RENSEIGNEMENTS SUIVANTS (SAUF POUR LES INSTRUCTIONS SAVE,SAVEA,LOAD,LOADA,QUI NE FONT PAS PARTIE DU DOS).

LE DOS EMPLOIE UNE TECHNIQUE QUI PEUT INTERESSER LES PERSONNES N'AYANT PAS DE DISK DRIVE:L'APPEL D'UN ASSEMBLEUR AVEC UNE INSTRUCTION :L'APPEL SE FAISANT EN DONNANT LE NOM DE L'INSTRUCTION AU CLAVIER OU PAR PROGRAMME PAR 10 POKE #131,3:PRINT "INSTRUCTION":POKE #131,1
POUR CELA IL FAUT CHANGER LE VECTEUR KEYBOARD PAR E5 2A 6E PAR C3 4A 0A
L'INITIALISATION DANS LE DOS SE FAIT PAR LA ROUTINE 13EC
L'ADRESSE DE LA TABLE SE MET A 297 (VOIR 13FC)

```
13FC LXI H(#1837)
     SHLD (#297)

0A4A PUSH H
0A4B LXI H((A51)
0A4E XTHL
0A50 PCHL
```

CETTE ROUTINE SERT A EXECUTER #A51 SI LE MOT ENCODE NE SE TROUVE PAS DANS LA TABLE DE ROM.
LA ROUTINE #A51 SERT A CONSULTER LA TABLE DU DOS.
LE VECTEUR 2DD (CHARACTER OUT) EST ARME ET CONTIENT C3 CF 0A.
SI ON LE COUPE,L'EXECUTION D'UNE INSTRUCTION DU DOS PAR PROGRAMME EST IGNOREE.(JE CROIS QU'IL Y AURAIT INTERET A BIEN COMPRENDRE LES ROUTINES A4A-ADA ET 1746-1834 POUR PERMETTRE AUX UTILISATEURS SANS DISK D'AVOIR LA FACULTE D'APPEL D'UNE INSTRUCTION PAR CETTE TECHNIQUE.

POUR LA PARTIE PROGRAMMATION,LES VECTEURS 2C5 A 2E0 SONT CHANGES
2C5 JUMP 6A7
2C8 JUMP 6D7
2CB JUMP 751
2CE JUMP 865
2D1 JUMP 879
2D4 JUMP 963
CETTE TABLE SE TROUVE EN C6E-C7D ET SE TRANFERE PAR TRANFERT BLOCK DE LA ROM PAR 32A JUMP (DE4F) (REF DAINAMIC 82/15)
LES VECTEURS CASSETTE SWITCH SONT RAPPELLES EN 32D-33C

L'INSTRUCTION CREATE QUI N'A PAS ETE BIEN EXPLIQUEE EST LIBELLEE COMME SUIT :
CREATE FILEMANE.STR:0 7 OU CREATE FILENAME.STR:0 D (D=VARIABLE)
OU POKE #131,3:PRINT "CREATE "+D$+" "+D AVEC D$="FILENAME.STR:0"
CECI EXECUTE L'OUVERTURE D'UN FICHIER STR DE NOM FILENAME ET DE 7 SECTEURS DE LONGUEUR.

POUR SAUVER UN FICHIER,PLUSIEURS ENNUIS PEUVENT SURVENIR.
-LE FICHIER EST RATE A L'ECRITURE PARFOIS PAR UNE TETE DE LECTURE TROP FROIDE OU TROP CHAUDE OU UNE CAUSE INDETERMINEE (TRAVAIL INTENSE ET RAPIDE MAIS RARE;JE SUIS PARVENU A TROP CHAUFFER LA TETE PAR UN PROGRAMME DE TRANSFERT RAPIDE D'UN DISK VERS L'AUTRE EN PASSANT PAR LE DIRECTORY DU DISK 1 ET AVEC UNE LONGUEUR DE FICHIER DE 1 SECTEUR: LA PREMIERE FOIS JE SUIS PARVENU A FAIRE PASSER 80 SECTEURS PUIS UN ARRET ;JE L'AI RELANCE AUSSITOT,IL N A FAIT QUE 10 SECTEURS ET AU TROISIEME ESSAI IL N'EN A MEME PAS FAIT UN).LE DIRECTORY DANS CE CAS N'EST PAS TOUCHE.
-LE FICHIER EST CREE MAIS IL Y A UN FICHIER VIDE DE LONGUEUR 1 AVEC LE NOM INSCRIT DANS LE DIRECTORY ET UN VERIFY EXACT.
-LE FICHIER EST CREE ET LE NOMBRE DE SECTEURS EST EXACT MAIS LE FICHIER NE PEUT ETRE LU (PEUT ETRE CONTRE PAR UN VERIFY)
-SI UN FICHIER A UNE LONGUEUR EGALE A UN MULTIPLE DE 128 BYTES, IL Y A DANGER DE BLOQUAGE.LE DOS OUVRE UN SECTEUR DE TROP OU IL SE TROUVE DES #55 .A LA LECTURE,CE SECTEUR EST LU COMME UN FICHIER ET IL APPARAIT DES "UUUUU".
MAIS PARFOIS LA LECTURE EST IMPOSSIBLE;LE FICHIER EST PERDU.
POUR RETABLIR LA LECTURE,IL FAUT REMPLIR CE SECTEUR DE CODES 00 (VOIR PLUS LOIN POUR LA TECHNIQUE)
POUR S'ASSURER QU'UN FICHIER A ETE BIEN ENREGISTRE,LE MEILLEUR MOYEN QUE JE VOIS EST DE CREER UN AUTRE FICHIER, DE FAIRE LOADA ET DE COMPARER LES DEUX FICHIERS. CETTE TECHNIQUE NE PEUT ETRE FAITE QU'AVEC DE PETITS FICHIERS.

SI ON SAUVE UN FICHIER QUI EST DEJA SUR LA DISQUETTE ET QUE LE FICHIER QUE L'ON SAUVE EST PLUS GRAND QUE LE FICHIER SUR DISQUETTE,IL Y A APPARITION DU FATIDIQUE "END OF FILE" QUI SIGNIFIE ICI FICHIER DETRUIT.POUR EVITER CECI,ON DOIT FAIRE UN "RENAME" DU FICHIER SUR DISQUE,PUIS "DELETE" ET ENFIN SAUVER LE NOUVEAU FICHIER.
REMARQUE:LE NOM D'UN FICHIER DETRUIT PAR "DELETE" RESTE DANS LE $DKDIR ET EST RECONNU LORS D'UN SAVEA.

UN AUTRE PROBLEME,LES MESSAGES D'ERREUR NE BLOQUENT PAS LE PROGRAMME BASIC. AVEC LES INSTRUCTIONS DE LA ROM,UN NUMBER OUT OF RANGE,... ETC ARRETE L'EXECUTION DU BASIC. ICI PAS, LE PROGRAMME PASSE A L'INSTRUCTION SUIVANTE.

POUR LIRE OU ECRIRE UN SECTEUR,ON EXECUTE CE QUI SUIT SOIT EN UTILITY SOIT EN ASSEMBLEUR:
A=NUMERO DU DISK DRIVE #30 OU #31 ("0" OU "1" EN ASCII)
B=NUMERO DE PISTE #0 A #23 (35 DECIMAL)
C=NUMERO DE SECTEUR #1 A #12
HL=ADRESSE OU ON DESIRE PLACER LE SECTEUR (LE DOS LE PLACE EN #1945 (ref #D70)POUR SON BACKUP OU COPY .LE NUMERO DU DISK DRIVE EN #19C5 (ref #D6D);MAIS L'ADRESSE POUR CETTE ROUTINE EST LAISSEE AU BON VOULOIR DU PROGRAMMEUR.

POUR LIRE UN SECTEUR,EXECUTER #A1C
POUR ECRIRE UN SECTEUR,EXECUTER #A32
EN CAS D'ERREUR ON AURA AU RETOUR LE REGISTRE A<>0
UNE AUTRE ZONE MEMOIRE SE TROUVE EN #977- #9B9 AVEC
EN 992-99C LE NOM EN 8 LETTRES D'UN FICHIER OU D'UN PROGRAMME
SI - DE 8 LETTRES,LE DOS COMPLETE AVEC DES BLANCS (#20)
SUIVIS DE BAS,BIN,STR,INT,FPT(CES NOMS SE TROUVENT
EN #761-#76F DANS LE DOS)
EN 9A7 NUMERO DU DISK EN ASCII
EN 9A8 TYPE DE FICHIER QUE L'ON A :
0 BAS
1 BIN
2 FPT
3 INT
4 STR
EN 9A9 LONGUEUR REELLE DU NOM SANS LES ESPACES
EN 9AC-9AD : COPIE DE 2A3-2A4
EN 9B1 NOMBRE DE BITS A CONSIDERER POUR ARRETER LA LECTURE/ECRITURE
DES MOTS D'UN FICHIER (POUR UN FICHIER STR,#9B1=#FF;POUR UN
FICHIER BAS,#9B1=#0F (JE N'AI PAS COMPRIS POURQUOI #0F)).
EN 9B7 SI <> 0 ERREUR
LA ROUTINE #14EF SERT A ALLER CHERCHER UN PROGRAMME SUR
DISQUETTE ET A LE LANCER.
CALLM#300,N$:AVEC N$=NOM DU PROGRAMME DEMANDE
HL POINTE VERS LE NOM DU PROGRAMME
EN 300 JUMP (14EF)
LA ROUTINE #5CE SERT A IMPRIMER LES MESSAGES D'ERREUR
ELLE SE TROUVE A LA FIN DE TOUTES LES INSTRUCTIONS ET EST
APPELEE PAR CNZ(5CE)
LES ROUTINES DE DISCUSSION AVEC LE DISK DRIVE SONT
AU NOMBRE DE 5.
387 SE TROUVE EN PREMIER LIEU DANS LES INSTRUCTIONS
PROBABLEMENT UNE ROUTINE D'INITIALISATION
3B3 ENVOI D'UN CARACTERE VERS LE DISK DRIVE
3D6 RECEPTION D'UN CARACTERE DU DISK DRIVE
4AA NON COMPRISE
3A5 NON COMPRISE
PAR LA ROUTINE 3B3,LE DOS ENVOIE UN CODE CORRESPONDANT A CERTAINES
INSTRUCTIONS:CELA SE FAIT PAR LA ROUTINE 402 AVEC UNE SEQUENCE
DE CODES STOCKEE EN 980 981 982 ...ETC
5A EXECUTE LE RESETD
49 A :A=NUMERO DU DISK DRIVE:IDISK(LE IDISK PEUT AUSSI ETRE
EXECUTE PAR 9C7 AVEC A=NUMERO DU DISK
4E A B C :A=NUMERO DU DISK
B C NOMBRE DE SECTEURS CREES
CELA SEMBLE ETRE UNE SORTE DE CREATE
53 LECTURE
54 ECRITURE
4F A SUIVI DE SOIT 49 SOIT 4F
52
43
4B
JE NE COMPRENDS PAS ENCORE BIEN CES DIFFERENTS CODES ET LEUR
FONCTION.
CONCERNANT LE PROBLEME DU FICHAGE, DEUX SOLUTIONS
SE PRESENTENT:
SI L'ON A UN FICHIER AVEC UN NOM ET DES VALEURS
SATELLITES IMPORTANTES

EXEMPLE:
BAPTISTE / alain/ 6 PLACE DE DINANT/ 1000/ BRUXELLES
NOM *********VALEUR SATELLITE*********
ET QUE CES VALEURS ET LE NOM FONT PLUS DE 612/107=+-6 SECTEURS
ON CREE UN FICHIER A$(4)
AVEC A$(0)="BAPTISTE"
A$(4)="BRUXELLES"
ET ON FAIT SAVEA A$ "BAPTISTE". ATTENTION,LE NOM <= 8 LETTRES
SI PAR CONTRE,LA TAILLE DU FICHIER EST < 6*128 BYTES ET SI ON A
PLUS DE 107 NOMS,ON OUVRE UN FICHIER A$(10,4) PAR EXEMPLE AVEC 11 NOMS
ET ON SAUVE EN BLOCS DE PLUSIEURS NOMS.
SI L'ON A QUE DES NOMS,ON OUVRE UN FICHIER A$(255)
OU A$(10) ET ON SAUVE EN AYANT EU SOIN DE NE SAUVER QUE LES
MEMOIRES OCCUPEES.SI ON A A$(255) AVEC SEULEMENT 20 NOMS ON
PERD (255-20) = 235 BYTES OU 2 SECTEURS QUI SONT POURTANT VIDES.
(FAUTE VUE DANS LE WORD PROCESSOR)
POUR EVITER CELA,ON PEUT SOIT CREER UN FICHIER DE LONGUEUR A$(19)
ET TRANSFERER LES DONNEES DANS CE FICHIER SOIT CHANGER LES
CARACTERISTIQUES DU FICHIER PAR DES POKES SUR VARPTR(A$(0))-3 -2 ET -1
ET LES REMETTRE EN PLACE APRES L'AVOIR SAUVE.
SI UN FICHIER DOIT ETRE LU,CORRIGE PUIS ETRE REPLACE
SUR LA MEME DISQUETTE, IL EST SAGE DE CALCULER AU MOMENT
DE LA LECTURE LE NOMBRE DE SECTEURS QU'IL PREND SUR LA DISQUETTE
POUR CELA ON FAIT :

```
30000 LONG=2:FOR I=0 TO 255
30010 IF A$(I)="" THEN 30040
30020 LONG=LONG+1+LEN(A$(I))
30030 NEXT I
30040 SECTOR=(LONG+1)/128
```

AVANT DE SAUVER,ON REFAIT LA MEME CHOSE AVEC LE FICHIER
A SAUVER. SI LONG = UN MULTIPLE EXACT DE 128,ON DOIT RAJOUTER
UN NOM DE PASSE SOUS PEINE DE PERDRE LE FICHIER.PUIS ON CALCULE
LE NOMBRE DE SECTEUR DU NOUVEAU FICHIER A SAUVER.SI CE NOMBRE
EST <= A L'ANCIEN,PAS DE PROBLEME.PAR CONTRE,S'IL
EST SUPERIEUR ON "RENAME" L'ANCIEN PUIS "DELETE" ET ENFIN
ON SAUVE LE NOUVEAU SUR DISQUETTE.CE FICHIER SE METTRA A LA
FIN DE LA DISQUETTE.POUR RECUPERER LA PLACE PERDUE ON EXECUTERA UN
COMPACT A LA FIN DU TRAVAIL.
EN CONCLUSION,LE DISK A DES AVANTAGES ET DES
INCONVENIENTS PAR RAPPORT AUX AUTRES DOS DES MARQUES CONCURRENTES.
AVANTAGES

ON N'A PAS BESOIN DE DONNER UNE LONGUEUR FIXE PAR DONNEE,
CE QUI FAIT QU'ON NE PERD PAS DE PLACE SUR LA DISQUETTE.
CECI EST UN GROS AVANTAGE QUI DOIT AU MOINS FAIRE GAGNER
UNE QUARANTAINE DE K PAR DISQUETTE (JE DIS BIEN 40000 BYTES
SUR LES 78000 QUE PEUT CONTENIR UNE DISQUETTE !).
INCONVENIENTS

LE DIRECTORY EST LIMITE A 107 NOMS.
LE SAVEA/LOADA TRANSFERE LE FICHIER DANS LE BUFFER APRES LE
PROGRAMME AVANT DE LE SAUVER/LIRE,CE QUI ENTRAINE UNE PERTE
DE PLACE ET DE TEMPS.CECI ETAIT INDISPENSABLE POUR LA CASSETTE,
CAR LE FLOT DE DONNEES DOIT ETRE REGULIER,MAIS ICI LE PROCESSEUR
DU DISK EST ESCLAVE ET PEUT ETRE LEGEREMENT FREINE.
MAIS AU TOTAL L'INSTRUCTION SERAIT PLUS RAPIDE.
POUR L'INSTANT UN FICHIER DE 6 SECTEURS NECESSITE 10 SECONDES,
CE QUI EST BEAUCOUP TROP LENT (J'AI CREE UNE ROUTINE EN ASSEMBLEUR
QUI NE NECESSITE QUE 4" ET IL Y A MOYEN DE FAIRE BEAUCOUP MIEUX).
LA DISKETTE,QUAND ELLE TRAVAILLE,BLOQUE L'UNITE CENTRALE
CONTRAIREMENT A CE QUE RENSEIGNE DAI DANS SA PUBLICITE.
LA DISQUETTE POURRAIT EFFECTIVEMENT LE FAIRE MAIS LE DOS N'A PAS
ETE PREVU POUR CELA.

CHER MR HERMANS

JE VOUS ENVOIE ENCORE QUELQUES TRUCS CONCERNANT PARTICULIEREMENT LES DISK I/O.

LE DSAVE OU LE SAUVETAGE DES PROGRAMMES BINAIRES EST LIBELLE COMME SUIT:EXEMPLE DSAVE ESSAI.BIN:0 2000 2EFF 2345
2000 DEBUT DU PROGRAMME.
2EFF FIN DU PROGRAMME.
2345 POINT D ENTREE DE LA ROUTINE D INITIALISATION.
IL EST PREFERABLE DE TOUJOURS INDIQUER FILENAME.BAS:1, CE QU INDIQUE QUEL FICHIER VOUS SAUVEZ (BAS,BIN,STR,FPT,INT) ET DE PREFERENCE LE NUMERO DU DISK CAR IL PEUT Y AVOIR DES PROBLEMES.
SI L ON SAUVE UN FICHIER BINAIRE SOUS LE NOM $USER SUR UNE DISQUETTE CONTENANT UN DOS, A L' INITIALISATION, LE COMPUTER LIRA LE $USER.BIN ET EXECUTERA LA ROUTINE D INITIALISATION:DANS CETTE ROUTINE ON CHANGERA LE 29B-29C.LE CHANGEMENT DES AUTRES POINTEURS SE FERA PAR LA ROUTINE D INITIALISATION DU DOS ET NE DOIT DONC PAS ETRE CHANGEE ICI.ON VEILLERA A TOUJOURS AVOIR LE BOOTSTRAP ET LE MSTRDOS AVEC LE MEME NOMBRE DE SECTEURS QUE SUR LA DISQUETTE NORMALE.(PAS DE SECTEUR VIDE)
EN METTANT UN $USER.BIN ET UN $USER.BAS EN MEME TANT QUE LE DOS ON PEUT A L INITIALISATION CHARGER UN BINAIRE, METTRE LES POINTEURS EN PLACE SANS MANIPULATION, ENSUITE L'ORDINATEUR IRA CHERCHER LE BASIC ET LE LANCERA.CELUI-CI A SON TOUR POURRA ALLER CHERCHER DES FICHIERS FPT,INT OU STR.

LE NOM D'UN FICHIER NE DOIT PAS NECESSAIREMENT AVOIR 8 BYTES DE LONGUEURS . ON PEUT L'APPELER PAR EXEMPLE:LOAD "TAR.BAS:1" QUI VEUT DIRE PROGRAMME BASIC "TAR" SUR LE DISK 1.

L APPEL CALLM #300 NE FNCTIONNE QU'UNE FOIS.SI ON REFAIT UN AUTRE APPEL,IL Y A UN PLANTAGE MONUMENTAL.

IL EST TRES FACILE D'AJOUTER DES INSTRUCTIONS A DOS:
EXEMPLE:
CHANGER LE POINTEUR 0297-0298 PAR 00-20 (#2000) PUIS ENTRER LE PETIT PROGRAMME SUIVANT
2000 03 41 41 41 77 04 00 37 18
PUIS REVENER EN BASIC ET TAPER AU CLAVIER AAA PUIS RETURN
LE RESETD S'EXECUTE.
ON PEUT AUSSI L'APPELER PAR PROGRAMME:
10 POKE #131,3:PRINT "AAA":POKE #131,1
EXPLICATION: LA TABLE DU DOS EST DEVIEE VERS 2000
3=LA LONGUEUR DU NOM DE L'INSTRUCTION.
41 41 41 CODE # DE A
77 04 =0477 ADRESSE DE L'INSTRUCTION RESETD
00 ARRET DE CETTE TABLE
37 18 ADRESSE DE LA TABLE DU DOS(#1837)

LE BACKUP ET LE COMPACT EXECUTENT LE IDISK AVANT D EFFECTUER LE RECOPIAGE.

LE COPY D'UN FICHIER OU D'UN PROGRAMME AJOUTE UN SECTEUR VIDE A LA FIN.POUR L EVITER ON FAIT UN CREATE AVANT LE COPY
EXEMPLE
FILENAME.BAS:0 A RECOPIER SUR DISK 1.IL A 50 SECTEURS
SI ON FAIT COPY FILENAME.BAS:0 ON AURA SUR LE DISK 1
51 SECTEURS POUR FILENAME.BAS
PAR CONTRE SI ON FAIT:
CREATE FILENAME.BAS:1 50
COPY FILENAME.BAS:0
ON N'AURA QUE 50 SECTEURS SUR LE DISK 1

LE BYTE #9B7, QUI EST LE BYTE D'ERREUR POUR LES INSTRUCTIONS DU DOS, PEUT SERVIR D UNE SORTE DE IF ERROR.

EXEMPLE:

```
5 DIM A$(255)
10 POKE #9B7,0:POKE #131,3:PRINT "VERIFY FILENAME.STR:0":POKE#131,1
15 IF PEEK(#9B7)=0 THEN 50
20 PRINT "LE FICHIER FILENAME NE SE TROUVE PAS SUR LE DISK,VEUILLEZ CHANGER LE
ISK"
30 RJ%=GETC:IF RJ%=0 THEN 30:GOTO 10
50 LOADA A$ "FILENAME.STR:0"
```

STOCKAGE DES DISQUETTES: ON VEILLERA A TOUJOURS COLLER SUR LE COTE LA PETITE ETIQUETTE POUR BLOQUER L'ECRITURE ;UN IDISK EST SI VITE ARRIVE.

L'INSTRUCTION RESETD SERT A REINITIALISER LE PROCESSEUR DE LA DISQUETTE :IL SERA UTILISE QUAND ON UTILISE LES FICHIERS ASCII ,QU' ON A LA DONNEE QUE L ON VEUT ET QUE L ON VEUT ARRETER LA LECTURE.

SI ON FAIT UN BREAK DANS UNE INSTRUCTION DU DOS OU DANS UN LOADA,SAVEA,LOAD,SAVE ,IL VAUT MIEUX,DES QU'ON A LE CONTROLE, EXECUTER UN RESETD SINON LA PROCHAINE INSTRUCTION DU DISK RISQUE D'ETRE FAUTIVE (SURTOUT POUR LE DIR QUI N'EFFECTUE PAS UN RESETD).

SI VOUS AVEZ UN FICHIER QUE VOUS MODIFIE REGULIEREMENT (FICHIER CORRESPONDANCE,STOCK,COMMANDE),VOUS DEVEZ TROUVER UNE TECHNIQUE POUR NE PAS RISQUER DE LE PERDRE.

LE SAUVETAGE TRIANGULAIRE

SOIT DISK "A" LE FICHIER FAIT LE 1 NOVEMBRE
VOUS FAITES UN BACKUP SUR LE DISK "B"
VOUS MODIFIER LE DISK "A" LE 5 NOVEMBRE
VOUS FAITES UN BACKUP DE "A" SUR "C"
VOUS AVEZ DONC
"B" DISK 1 NOVEMBRE
"A"-"C" DISK 5 NOVEMBRE
VOUS MODIFIEZ LE DISK "A" LE 10 NOVEMBRE
VOUS FAITES UN BACKUP DE "A" VERS "D"
VOUS AVEZ DONC
"B" DISK 1 NOVEMBRE
"C" DISK 5 NOVEMBRE
"A"-"D" DISK 10 NOVEMBRE
QUANT VOUS ETES CERTAIN QU'IL N Y A PAS EU D'ERREUR SUR "A"-"D" VOUS DETRUISE LE DISK "B".ON DOIT EN FAIT VEILLEZ A TOUJOURS AVOIR LE FICHIER, SA COPIE ET SA VERSION PRECEDENTE.
CECI EST UN MINIMUM.

POUR CEUX QUI FONT DES MODIFICATIONS QUOTIDIENNEMEN, IL VAUT MIEUX ADOPTER UN CYCLE SUR 7 JOURS. VOUS DETRUISEZ LE FICHIER DU JOUR DE LA SEMAINE PRECEDENTE ET GARDEZ DES LORS A TOUT MOMENT 6 VERSIONS.
LE LUNDI, ON DETRUIT LA VERSION DU LUNDI PRECEDENT
D'AUTRE PART, UNE COPIE NE SE STOCKE PAS DANS LA MEME BOITE QUE L ORIGINAL ET DE PREFERENCE PAS DANS LA MEME PIECE.CECI POUR EVITER DES ACCIDENTS (VOLS,CHAMP MAGNETIQUE ACCIDENTEL,ETC..)

CONCERNANT L'IMPRIMANTE (EPSON MX80 F/T),IL FAUT EVITER DE LUI ENVOYER TROP SOUVENT DES CARACTERES DE CONTROLE, CAR ELLE NE FAIT DES LORS PLUS QUE L'ECRITURE DE GAUCHE A DROITE ET EST DE CE FAIT LORS PLUS LENTE.

EXEMPLE:

```
10 FOR I=0 TO 200
20 PRINT CHR$(15);A$(I)
30 NEXT I
```

LE CHR$(15) EST RAPPELE A CHAQUE LIGNE.IL EST NON SEULEMENT INUTILE MAIS L'IMPRIMANTE NE TRAVAILLERA QUE DE GAUCHE A DROITE
IL VAUT MIEUX FAIRE

```
10 PRINT CHR$(15);:FOR I=0 TO 200
20 PRINT A$(I)
30 NEXT I
```

CONCERNANT LE PERSONAL COMPUTER,PLUSIEURS REMARQUES:
L INSTRUCTION FRE EST LE NOMBRE DE BYTES ENTRE LA FIN DE
LA TABLE DES SYMBOLES ET LE DEBUT DE LA ZONE D'ECRAN
ELLE FAIT LA DIFFERENCE ENTRE LA VALEUR DE (A5-A6) ET (A3-A4)
ELLE NE REPRESENTE EN RIEN LE RESTANT DE MEMOIRE.
SACHANT QUE L'INSTRUCTION SAVEA LOADA UTILISE CE BUFFER
POUR SAUVER SES FICHIERS, ON DOIT AVOIR LE FRE QUI DOIT
ETRE EGAL OU SUPERIEUR A LA DIMENSION DU FICHIER.
POUR LES FICHIERS STRINGS, ON CALCULE LA DIMENSION
DE LA FACON SUIVANTE:EXEMPLE: DIM A$(20,10)
1 BYTE POUR LE NOMBRE DE DIMENSION
2 BYTE POUR LES CARACTERISTIQUES (20,10)
(20+1)*(10+1)*2 POUR LE TABLEAU DE POINTEURS
1 BYTE DE LONGUEUR DE STRING+ LA VALEUR PAR MEMOIRE OUVERTE
SI CE NOMBRE EST PLUS GRAND QUE FRE, IL APPARAIT UN LOADING ERROR 1
ON PEUT A LA RIGUEUR,SI L ON N A PAS BESOIN DE L'ECRAN
A CE MOMENT CHANGER A3-A4 PAR #BFFF,PUIS LOADA ET REMETTRE
A3-A4 EN PLACE APRES

```
5 DIM A$(255)
10 AZ=PEEK(#A3):BZ=PEEK(#A4)
20 POKE #A3,#FF:POKE #A4,#BF
30 LOADA A$ "FICHIER"
40 POKE #A3,AZ:POKE #A4,BZ
50 MODE0:PRINT CHR$(12)
```

L'ECRAN PENDANT LE LOADA SE REMPLIT DE DONNEES CODEES,IL
FAUT L EFFACER PAR LA LIGNE 50.
AVEC CE SYSTEME, ON PEUT AVOIR UNE HEAP PLUS GRANDE MAIS
IL NE FAUT PAS AVOIR BESOIN DE L'ECRAN A CE MOMENT LA.

CONCERNANT L'ORGANISATION DES PROGRAMMES BASIC,
JE RESERVE DES ADRESSES POUR CERTAINES FONCTIONS,MEME
SI JE N EN AI PAS BESOIN
1-99 UNIQUEMENT L'INITIALISATION
100-199 CONTROLEUR GENERAL DOUT TOUT PART ET REVIENT
200-999 ZONES DE TRAVAIL OU LES TACHES SONT SEPAREES TOUS LES 100
TACHE 1 DE 200 A 299
TACHE 2 DE 300 A 399 ETC
1000 EDITEUR DE VALEURS
30000 LOADA DE FICHIERS
31000 SAVEA DE FICHIERS
32000 LECTURE DU DIRECTORY DU DISK
10000 ROUTINE D IMPRESSION SUR L'IMPRIMANTE
CECI PERMET DE CONSTRUIRE DE NOUVEAUX PROGRAMMES SANS
RELOCATER CERTAINES ZONES ET D'UTILISER LE MERGE POUR
FAIRE UN PROGRAMME. C'EST LE PRINCIPE DU MODULE.
JE RESERVE EGALEMENT CERTAINS NOMS POUR DES PROBLEMES
PARTICULIERS.
RJZ TOUJOURS POUR LE GETC
Z,ZZ,ZZ,ZZZ,Z$,ZZ$ TOUJOURS POUR DES CALCULS INTERMEDIAIRES
QUI VONT ETRE UTILISES TRES RAPIDEMENT DANS
LE PROGRAMME OU DES DONNEES TRANSMISES A UNE
SOUS- ROUTINE
EXEMPLE Z=3:GOSUB 2000:PRINT A$(Z) LE Z N'ETANT PLUS
UTILISE PAR APRES.CECI PERMET DE LIMITER LE NOMBRE DE
VARIABLES

I,J,K,IZ,JZ,KZ UNIQUEMENT POUR LES BOUCLES FOR NEXT
DI$( ) UNIQUEMENT POUR LE COPIAGE DU DIRECTORY($DKDIR)
AVEC DI$(0) OU DI$(0,0) ET DI$(0,1) LE NOMBRE DE FICHIERS OUVERTS
A$(),B$(),C$() A() AZ() ETC POUR LES FICHIERS
AVEC CETTE ORGANISATION, LA LISIBILITE DES
PROGRAMMES EST PLUS GRANDE.
JE VOUDRAIT SAVOIR SI VOUS AVEZ UN ASSEMBLEUR
POUR UTILISER DISK,ET CONTENANT EGALEMENT DES MACROS

A BAPTISTE PHARMACIEN

didaisoft

Nieuwe onderwijssoftware ... met het diDAIsoft-kwaliteitslabel.

1. diDAIsoft-talentape

a. Test your tenses level 1

5 lijsten van telkens 10 onregelmatige engelse werkwoorden worden ter studie aangeboden. Per lijst van 10 worden de hoofdtijden opgevraagd. Foutieve antwoorden worden onthouden , verbeterd en opnieuw aangeboden. Basiskennis.

b. Test your tenses level 2

6 lijsten van telkens 10 onregelmatige engelse werkwoorden worden op dezelfde wijze als in level 1 aangeboden en verwerkt. Uitbreiding van de basiskennis.

c. Test your tenses level 3

De 110 werkwoorden uit level 1 en level 2 worden at random opgevraagd.

d. Test your structures

De leerlingen moeten een correcte engelse zin vormen met een gegeven verb, subject, object en time adjunct. De aard van de gewenste zin (question, statement...) wordt eveneens opgegeven. Na twee pogingen wordt het correcte antwoord aangeboden.

e. Test your words: animals

De computer presenteert woorden en uitdrukkingen in verband met:
1. animal sounds
2. young animals
3. male ... and... female
4. groups of animals
5. animals we eat

Na een studietijd worden deze uitdrukkingen ondervraagd. Correctie van foute antwoorden.

f. Test your words: quantities and qualities

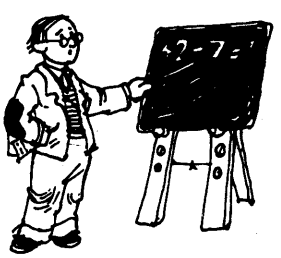
Analoog als bij animals. De woorden en uitdrukkingen hebben betrekking op quantities and qualities: synoniemen , tegengestelde begrippen....

g. synoniemen en antoniemen

De computer presenteert een woord tesamen met een ganse reeks andere. De leerling moet het synoniem of het antoniem uit de lijst aanstippen. Vlot te gebruiken. Eenvoudig aan te passen naar andere toepassingen.

h. Nederlandse items

Gebruiken jou leerlingen ook zo vaak foutieve uitdrukkingen zoals "wedden voor 20 frank" in plaats van "wedden om 20 frank" ? Laat ze dan dit diDAIsoft-programma uittesten. Dit programma hamert 40 analoge uitdrukkingen op een correcte wijze vast in hun taalgebruik.



2. Het diDAIsoft-familiebudget

Voortaan geen problemen meer met uw familiebudget. De DAI-personal computer houdt al uw inkomsten en uitgaven (zeer sterk gerubriceerd) bij, maakt alle gewenste totalen en overzichten. Het geeft u een duidelijk beeld (grafisch) van de diversiviteit van uw jaarlijkse uitgaven en berekent vanaf een gewenste maand het cumulatieve verschil van uw uitgaven en inkomsten, zodat u zelf uw eigen budget kritisch kan bekijken (grafisch) en beslissingen treffen in verband met uw maandelijkse spaaractiviteiten. Alle bedragen worden opgeslagen in arrays die op informatiedrager worden opgeslagen.
In het onderwijs als toepassing mogelijk in de lessen economie, handel... en in de lessen informatica waar bepaalde routines die in het programma uitgewerkt zijn kunnen bestudeerd worden.

3. Grafische hulp

Dit diDAIsoft-programma laat toe op een eenvoudige wijze tekeningen te maken en deze tekeningen tesamen met de tekst die erbij werd geschreven in files weg te schrijven en snel op te roepen. Op deze wijze kan men gemakkelijk de schermlayout van educatieve programma's verzorgen.
Geleverd met een uitgebreide handleiding (20 blz.)

4. diDAIsoft-wiskunde tape

a. Studie van de neperiaanse logaritmische functie

Zeer didactische grafische presentatie van de definitie en het verloop van de neperiaanse logaritmische functie.

b. Stelling van Pythagoras

Zeer duidelijke grafische illustratie bij het bewijs van de stelling van Pythagoras.

c. Methode van Gauss : oplossen van nxn stelsels

Dit programma lost nxn stelsels op met de methode van Gauss en laat toe voor n kleiner dan 6 deze methode stap per stap te volgen.

d. Grootste gemene deler van 2 natuurlijke getallen

Dit programma toont duidelijk het algoritme van de opeenvolgende delingen om de GGD te berekenen. Het laat bovendien toe dit algoritme zelf ook toe te passen.

e. Priemgetallen : Zeef van Eratosthenes

Dit programma bestaat uit 3 delen:
1. demo: het zeefalgoritme wordt toegelicht voor de priemgetallen kleiner dan 200
2. m.b.v. deze methode kan de leerling de priemgetallen kleiner dan 5000 bepalen deze worden genummerd, zodat kan gevraagd worden naar priemgetal nummer n.
3. onderzoeken of een getal kleiner dan 25000000 een priemgetal is en zo nee een priemdeler laten afdrukken

f. Oplossen van willekeurige en rechthoekige driehoeken (2 programma's)

Dit programma laat toe met concrete gegevens de driehoek volledig op te lossen. Het behandelt meer dan de vertrouwde hoofdgevallen.

didaisoft

5 diDAIsoft-fysica_tape

a. Veldlijnen A1

Dit programma laat toe de veldlijnen uit te tekenen veroorzaakt door twee ladingen. Doordat het programma aan de gebruiker toelaat waarden voor parameters in te voeren is het programma voldoende flexibel om alle aspecten van deze theorie toe te lichten. Een uitgebreide handleiding wordt bijgeleverd.

b. Tijd-weg

Het doel van dit programma is:
1. meten van tijden tussen N opeenvolgende sluitingen van de event ingang op paddle 1
2. in de onderstelling dat dit sluiten gebeurt door de beweging van een voorwerp (wagentje) voorbij kontakten op onderling regelmatige afstanden A langs een baan, uitzetten van de beweging in weg-tijd-grafiek
3. uitzetten in grafiek van gemiddelde snelheid en gemiddelde versnelling in elk interval. Met handleiding, o.a. t.b.v. het practisch uitvoeren van de kontakten.

c. Stopwatch (kort)

Dit programma laat toe de DAI p.c. als chronometer te gebruiken en een aflezing te doen met grote cijfers. Paddle noodzakelijk.

d. Stopwatch (lang)

Zoals stopwatch kort maar met mogelijkheid om het tijdsverloop te volgen.

e. Hydraulische pers

Principe en werking van de hydraulische pers wordt grafisch toegelicht.

f. U-vormige buis

Enkele concrete voorbeelden (water - kwik en water-olie) worden gebruikt om op een zeer mooie grafische wijze de wetmatigheden die gelden bij een U-vormige buis te illustreren.

Wat is diDAIsoft ?

diDAIsoft is een softwaregroep die werkt binnen DAInamic en bestaat uit leerkrachten uit alle niveaus van het Belgisch onderwijs (lager, secundair, hoger). De groep heeft als doelstelling voor alle vakken van het (hoofdzakelijk secundair) onderwijs didactische software te ontwikkelen. Op dit ogenblik bestaat de groep uit 27 leden, die tweemaandelijks samenkomen in het CMO-vormingscentrum te Haasrode Geldenaaksebaan 327, telkens op zaterdag van 10u tot 16u. Volgende bijeenkomst 26-2-1983. Bent U leraar en heeft U interesse, kom eens kijken. Vooraf toch graag een telefoontje aan Bruno Van Rompaey (016/461085). Waar de groep geïnteresseerde collega's te woord zal staan is op zaterdag 9 april 1983 op onze DAInamic dag in Westerlo-Tongerlo. Kom daar zeker eens kennis maken. We hebben ook uw bijdrage nodig om van de diDAIsoft-programma's kwaliteitssoftware te maken. We verwachten U, collega.

program identification

title	:	"KEN-DOS", a new floppy-standard for DAIpc
author	:	Kenneth Gooswit
purpose	:	
comment	:	We will bring test results in the next edition

Dear User,

About a year ago I was annoyed by the fact that there was no fast memory-unit available for the DAI-PC. Then Dai came with a twin-drive floppyunit, but in my opinion this unit doesn't meet all the needs of the common user. In fact, this unit is very slow.
To overcome this problem I then started to do some research.
After several months of hard working and talking to the right people, I managed to construct a controller-board that made it possible to operate four 80-track floppy drives.
At that moment only in single-density.
The problem was that I couldn't adapt any software to my system and I was forced to write mine own operating system.
Now most of the work is done, and I can proudly announce the birth of 'Ken-Dos'. I have contacted a few firms and it is now possible to supply a complete unit with fast accestime and a lot of storage capabilities. Up to 3,2 megbyte. A back-up is done within one minute and fifteen seconds (400kbyte)

THE HARDWARE

The system consists of a controller-PCB, powersupply with ringcore transformer,enclosure for two drives, one or two drives and an eprom-PC-Board with 14 banks of 2kbyte that has to be connected to the X-bus; thus 28kbyte extra memory.
The software resides totally in eprom (12kbyte) and there is an option of using 4kbyte of static rams (6116), which is then used as an buffer for the directory and the sectormap. Normally 2560 bytes of the DAI-ramspace from #A950 to #B350 is used for this purpose. This buffer moves automatically with the screenbuffer, so saving a picture won't be a problem. There are programs however, like the new assembler/editor, which use those addresses and in that case the user has the option of using the extra 4kbyte buffer. To do so however, the stack-interrupt circuit has to be disabled by putting the base of transistor T1 to ground.
The user has 16kbyte extra romspace at his disposal, so it is possible to put several programs in eprom.
Since Ken-Dos is totally in eprom, any program ever written for the DAI-PC can be loaded and run without relocation.
The controller PC-Board can handle four mini 80-trackdrives (double sided/double density). That means that the user can put 3,2megbyte on line.
To use the double density mode, the user has to make a minor hardware modification. Pin 49 of the X-bus has to be connected to pin 4 of the DCE-bus.
The powersupply can service two drives and is automatically switched on and off by the DAI-PC.
The controller PC-Board, powersupply and drives are horizontally mounted in a cabinet. On the rear is an extra connector for a printer or DCR.

THE SOFTWARE

The software can handle sequential- and randomacces-files.
In single density the user can store 200kbytes formatted and in double density
400kbytes on a single side of a diskette (DS/DD).
Since Ken-Dos is entirely in eprom, there is no need for a systemdisk so
all memoryspace on the diskette is at the users disposal.
After coldstart or reset Ken-Dos will be in search of com-files on the disk re
sided in drive-0. The names of those files are placed in the extended command-
table and are then handled as commands. This means that those files can be
loaded and eventually executed by just entering the name of the file.

After this search another search takes place. Ken-Dos will look for a file
with an unique mark telling that this file has to be loaded and eventually
executed. If no such file is found it returns by printing name of the inserted
disk,date, free sectors, status of protection and density on top of the screen.
Ken-Dos will then ask for date and password. To keep the password secret, it
is scrambled on the screen. If the user doesn't know the password, he will not
have acces to protected disks or files and utilities.
With Ken-Dos it is possible to link a basicprogram to a ML-part.On loading
the basicprogram the ML-program will automatically be loaded. First Ken-Dos
loads the ML-part and then the basicprogram. Pointers are corrected.
When loading a file, the user could enter the full name of the file, but
it is also possible to enter just a part of the name. Ken-Dos will find and
load the desired file. If an autostartmark is placed before the name (+),
the file will be executed.
Every file is saved with date of creation and date of the latest modifi-
cation.
The disk has a name, a date of formatting, and can be software protected
against writing or reading. Every disk has a directory of the created files.
In single density the user can create 64 files; in double density 128 files.
After entering the command 'DIR1', Ken-Dos will display the directory of the
disk in drive-1. The name, date etc. of the disk will be printed on top of the
screen followed by the filenames. On hitting the spacebar the filenames will
scroll on the screen. The diskname etc. will remain stable.
To display all files ,including deleted files, just type 'DIRi1'.

The following COMMANDS are available:

'DIR0"	: display directory on screen(drive-0 is selected)
'LIB1'	: all commands are displayed on screen
'LIB2'	: all extentioncommands (com-files) are displayed
'HELP'	: information about using Ken-Dos is given on the screen
'LOAD".."'	: normal Dai-load
'SAVE".."'	: normal Dai-save
'DLOAD".."'	: any file or a part can be loaded (ML-files etc.)
'DSAVE".."'	: any file or a part can be saved
'DISK1'	: assign system to disk (drive-1 is selected)
'CAS'	: assign system to casset
'RDCAS'	: read-only of casset
'WRCAS'	: write-only
'DCR'	: assign DCE-bus to DCR
'LPRINT'	: assign DCE-bus to printer
'PASS'	: to enter the password
'DATE'	: to enter the date
'BASIC'	: basic pointers are shown and can be modified
'COM1'	: com-files are retrieved from disk (drive-1 is selected)
'BANK5'	: to select an eprombank (bank-5 is selected)

The following commands are only available to the user who knows the right password:

```
'RENAME".."': to change the filename
'DNAME1'    : to change the diskname(drive-1 is selected)
'LOCK".."'  : to protect file
'UNLOCK".."':
'DELETE".."': to delete a file
'RESTORE"."':
'KILL".."'  : to destroy a file
'COMPACT"."': to copy all files (except deleted files)
'COPY".."'  : copy a file
'BACKUP".."': total copy of the disk
'OPEN'".."' : open a file for writing
'CLOSE".."' : close it
'VERIFY".."': verify a file and print informations about lenght,startaddress,
'FORMAT".."': format a disk                                              etc...
'MANUAL'    : To read or write any sector on the disk
```

It is possible to link most commands to basicprograms. A basic-line with 'CALLM#F000:REM ...(command)' will do the job.
There are more features, but explaining them all goes beyond the purpose of this article. For detailed information I refer to the USERS-MANUAL

The whole system complete with one or two double-sided drives (resp. 800kb or 1,6 megbyte of memoryspace) and the USERS-MANUAL can be ordered.
Just write to:

Ken-dos dev.
P.O.B. 40
1616 ZG HOOGKARSPEL
HOLLAND
TEL: 02285/1 28 93

or:

MIKRO SHOP
Bennenbergweg 1
3221 NIEUWRODE (Bij AARSCHOT)
BELGIUM
TEL: 016/56 87 70

It will be possible to adapt the system to other minidrives or microdrives.
For further information write to us.
There are also entrypoints to use with modified CPM
A utility program to read standard DAI-disks will be available.
Ken-Dos is written in a way that it can be extended. All extentions will be available at minor costs

Kenneth Gooswit

KEN-DOS

AUDIO CASSETTE INTERFACE

PART 1: WRITING TO TAPE

1. FORMAT:

1.1. DATA FORMAT:

Many personal and hobby computers use the so-called Kansas-City format for writing data to an audio cassette tape. This is a FSK (frequency shift keying) method, using 1200 Hz and 2400 Hz tones to write '0' and '1' data bits to the tape.
The DAI uses a different format. For each bit written to tape, it uses 2 impulses of different duration, a short one and a long one. The signal send to tape for a '0' and for a '1' bit are given in figure 1.

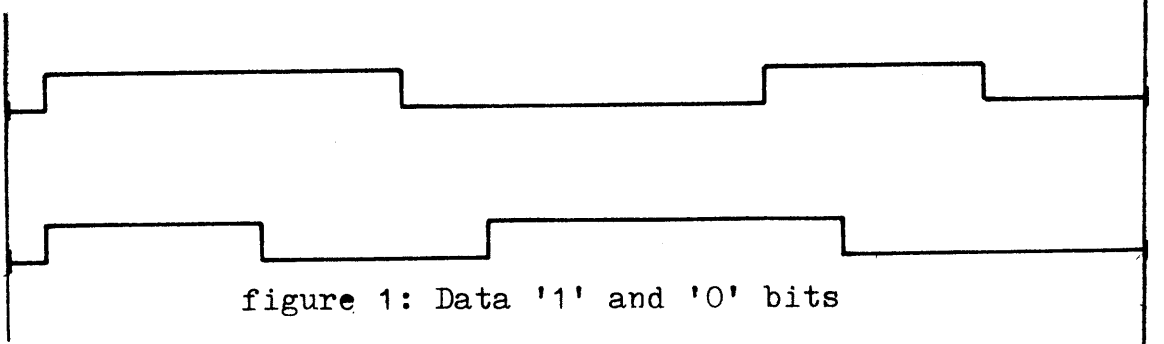


figure 1: Data '1' and '0' bits

1.2. LEADER FORMAT:

At the beginning of a program (a 'file'), written on tape, a 'leader' tone is written in order to enable synchronisation when the file has to be read from the tape. The leader consists of a great number of bits, for which the duration of the 'long' and the 'short' impulse are equal (see figure 2). This results in a constant tone of about 890 Hz for ca. 2.3 secs.

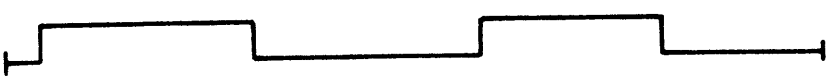


figure 2: a leader bit

At the end of the constant tone, one '1' bit is written to tape to indicate the end of the leader tone.

1.3. TRAILER FORMAT:

Figure 3 shows the format of a bit of the trailer tone, which is written after a file on tape to indicate its end.
Although the DAI writes a trailer tone on the tape, it does not use it when reading the file from tape.

figure 3: a trailer bit

A data '1' bit ends the trailer tone.

1.4. TAPE SPEED DATA:

The information for the impulse duration of the impulses to be written on tape is moved from ROM (D7C5-D7CA) into RAM 02E6-02EB during system reset. These constants are:

02E6/7	24/24	Impulse,duration leader bits
02E8/9	24/3C	Impulse duration data bits
02EA/B	24/18	Impulse duration trailer bits

2. AUDIO CASSETTE RECORDER CONNECTIONS:

Two audio cassette recorders can be connected. Their motors can be switched on/off under program control. The audio input and output channels of both recorders are parallel connected.

Output DAI to recorders:	Port FD06, bit 0.
Input recorders to DAI:	Port FD00, bit 7.
Motorcontrol rec. 1:	Port FD06, bit 4.
Motorcontrol rec. 2:	Port FD06, bit 5.

A copy of the data send to port FD06 is kept in RAM (#013D), because port FD06 may only be written to.

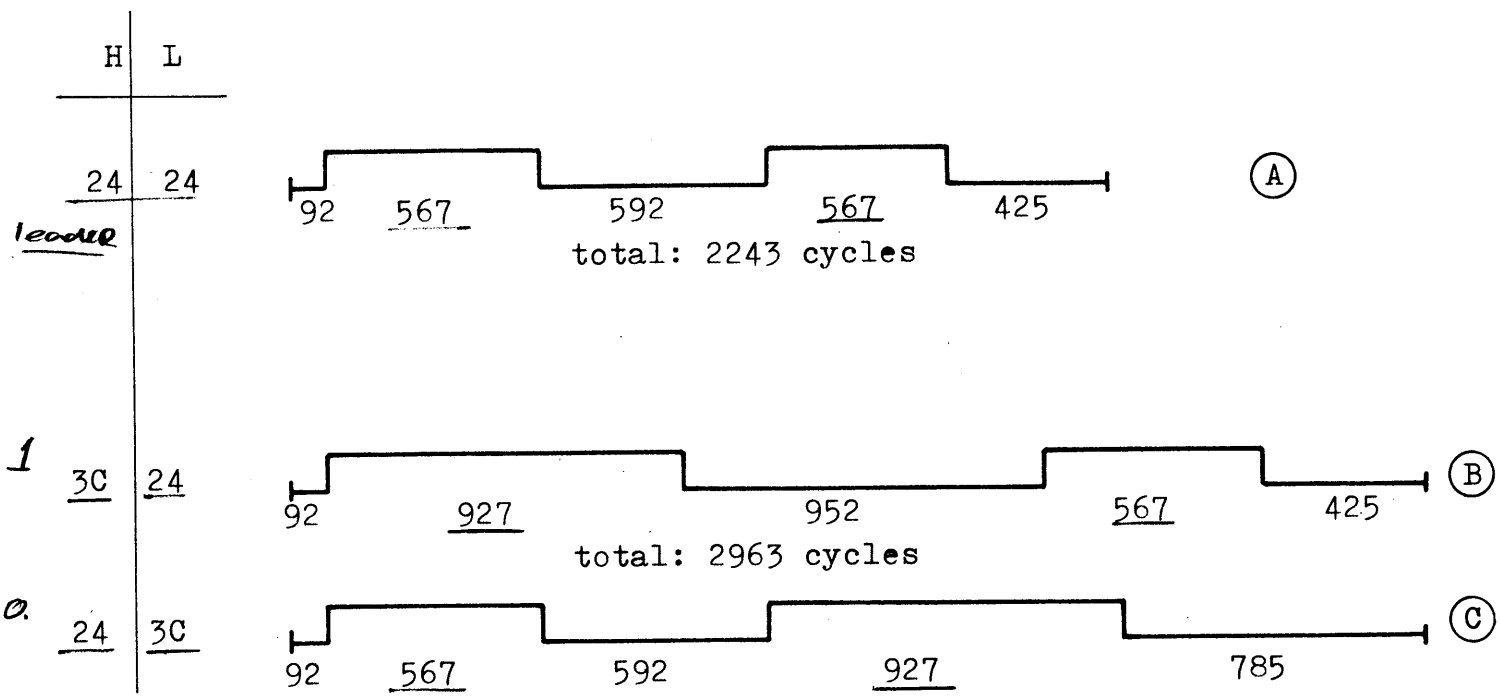
See the DAI schematics (sheet 5) for details on the hardware.
Details on the software can be found in the 'DAI pc FIRMWARE MANUAL'.

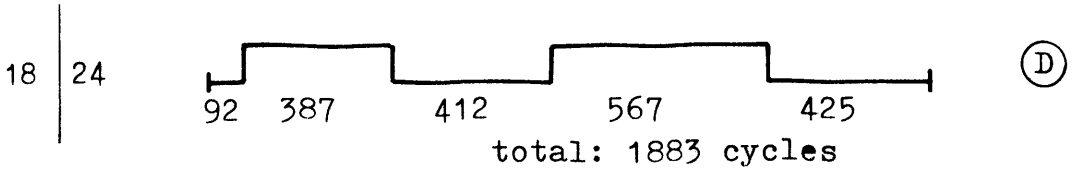
3. SUBROUTINES FOR WRITING TO TAPE:

3.1. WRITE A BIT - WBIT/D524:

On entry, the HL-registers contain the relevant impulse duration information from 02E6-02EB. One cycle (= 1 impulse = high/low) is written to port FD06/bit 0 for the duration of the value loaded into H. Then the value in L is moved into H, and the second cycle (with other impulse duration) is written to tape (see figure 1).

This routine is used for writing data bits, leader bits and trailer bits to tape.





total: 1883 cycles

Duration given in machine cycles (1 cycle = 0,5 usec)

Tape speed data:

24/24 - Used for the leader.

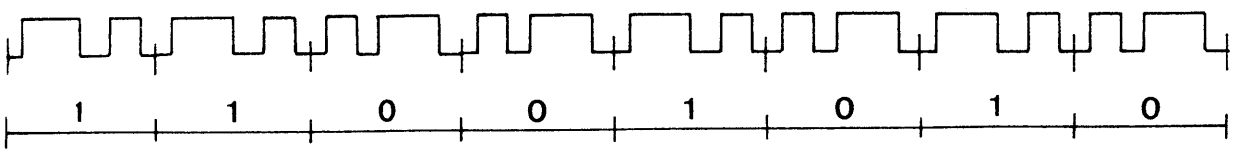
3C/24, 24/3C - Used for data bytes: '1', '0'

18/24 - Used for the trailer.

Figure 4: Bit patterns and timing

3.2. WRITE A BYTE - WBYTE/D509:

On entry, the byte to be written is in the accumulator.The HL registers contain the impulse duration values, and the DE registers are loaded with the inversed values.
To write a byte (= 8 bits), the highest bit is moved into the CY-flag. If the bit is '1', fig.4b is written to tape; if the bit is '0', fig.4c is written to tape. This sequence is done 8 times.



data byte 11001010 = CA

figure 5: A data byte pattern

3.3. WRITE A LEADER - WLEAD/D4ED:

The HL registers are loaded with the tape speed data 24/24. In BC, the number of leader bits to be written are loaded (#07E8 = 2024). Now 2024 times 24/24 impulses are written via WBIT. When all leader bits are written, HL is loaded with the tape speed data for a data bit, and a '1' is written, indicating the end of the leader.

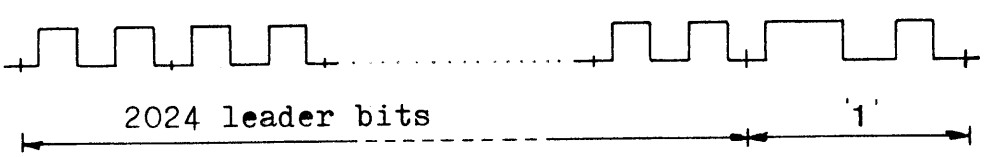


figure 6: Leader pattern

3.4. WRITE A TRAILER - WTRLX/D550:

On entry, BC = #0142 + length of the program name. Via D4F6 (WLD10), a number of trailer bits (18/24), equal to the contents of the C-register is written. At the end, a '1' bit indicates the end of the trailer.

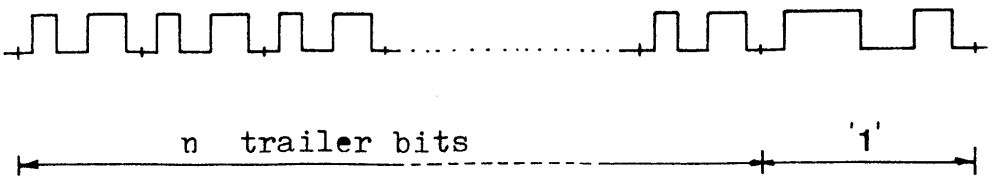


figure 7: Trailer bit pattern

4. FILE HANDLING:

Now all subroutines are known, the file set-up will be explained. (A program written to tape can be considered as a file). Normally, a file consists of one or more blocks of data, which are written to tape sequentially. At the beginning of the file on tape, a kind of a 'header' is written, containing the leader tone and some information on the type of file. The end of the file is marked with a trailer tone.

For audio cassette operation, subroutines are available in the resident software to open a tape file (CWOPEN), write data blocks to tape (CWBLK) and to close a tape file (CWCLOSE).
The startaddresses of these audio tape file handling routines are moved from ROM (D7A4-D7AC) into RAM 02C5-02CD during system reset. These RAM pointers WOPEN, WBLK, WCLOSE can be loaded with other - user defined - pointers in order to address write routines for other mass storage devices, like floppy disk or digital cassette recorder (like the Memocom MDCR).

The default values (audio cassette recorder) are:

```
WOPEN     02C5 JMP #D2B8
WBLK      02C8 JMP #D2F1
WCLOSE    02CB JMP #D427
```

4.1. OPEN A TAPE FILE - CWOPEN/D2B8:

- Evaluate if writing to tape during a program run or in direct mode (MPT21/D720).
- If in direct mode: Print 'SET RECORD,START TAPE,TYPE SPACE' and wait for the spacebar to be pressed.
- Start selected cassette motor(s) (CASST/D42E).
- Write leader tone (WLEAD/D4ED).
- Write a 'flag byte' #55 to tape. This is the first byte after the leader tone.
- Write a 'file type byte':
  #30 : Basic program.
  #31 : A hex file.
  #32 : An array.
- Write name length and name of the file (if a name is given) to tape via a CWBLK routine (via MPT22/D7F8).

4.2. WRITE A DATA BLOCK TO TAPE - CWBLK/D2F1:

- Write the length of the block, followed by its checksum.
- Write the contents of the block, plus its checksum.

The checksum-byte is a control byte on all transmitted data bytes. It is intially set to #56, and then EXOR-ed with the data byte, followed by a RLC (rotate left). This is done again and again for all data bytes transmitted. The final checksum byte is written on tape after the last data byte.

4.3. CLOSE A TAPE FILE - CWCLOSE/D427:

- Write trailer bits (WTRLX/D550).
- Switches off cassette motors (CASSP/D445).

5. RESIDENT DATA FILING ROUTINES:

The DAI firmware knows 3 data filing routines:

'SAVE'	file type 0	Save Basic programs.
'UT/write'	file type 1	Save Hex files
'SAVEA'	file type 2	Save Arrays

The use of the routines WOPEN, WBLK and WCLOSE is slightly different for each of the 3 methods.

5.1. BASICCOMMAND 'SAVE' - RSAVE/D23D:

- D23D: Clear all variables, calculate length of textbuffer and symboltable, evaluate a (evt) given name.
- D263: WOPEN (see 4.1).
- D268: WBLK: write textbuffer length + checksum.
  idem for textbuffer contents.
- D26C: WBLK: write symboltable length + checksum.
  idem for symboltable contents.
  WCLOSE (see 4.3).

5.2. UTILITY COMMAND 'W(rite)' - 3EEE4:

- EEE4: Get low- and highaddress on stack.
- EEED: Get an evt. name in the encoded input buffer.
- EEF8: WOPEN.
- EEFE: WBLK: write startaddress (length + checksum, contents + checksum).
- EF08: WBLK: write hex block (length + checksum, contents + checksum)
- EF0B: WCLOSE.

5.3. BASICCOMMAND 'SAVEA' - RSAVA/D81D:

- D81D: Test type of array.
- D824: If stringarray: Arrange stringelements in the free RAM space (see previous article on SAVEA/LOADA).
- D82F: WOPEN.
- D837: WBLK: write array type (length + checksum, byte + checksum).
  WBLK: write array contents (length + checksum, contents + checksum).
  WCLOSE.

program identification	
title	3-D pictures (with colored glasses)
author	J.Roelants
purpose	
comment	you need glasses with one red and one green filter.

Als waarnemer geplaatst in de ruimte ontvangen we twee verschillende beelden via ons linker-en rechter oog . Deze twee beelden versmelten in onze hersenen tot één enkel beeld zodanig dat er een diepte-indruk ontstaat, m.a.w. we verkrijgen informatie omtrent de drie dimensionele ruimte die ons omringt .
Willen we een 3D-ruimte suggereren op een plat vlak dan kan dit in principe gebeuren door twee perspectievisch verschoven beelden te creëren . Het probleem stelt zich nu dat er door één oog slechts één beeld mag worden waargenomen . We rekenen er op dat onze hersenen deze twee beelden integreren tot één enkel 3D-beeld .
Een eerste bekende methode bestaat er uit een stereo-kijker te ontwerpen die er voor zorgt dat elk oog slechts één van de voorgestelde figuren waarneemt . Het voordeel van deze methode bestaat erin dat de voorgestelde figuren allerlei kleuren kunnen bevatten .

Een tweede methode bestaat erin de twee perspectievisch verschoven figuren over elkaar te plaatsen . Beide figuren bestaan uit één kleur , bijvoorbeeld rood en groen . Daar waar de gemeenschappelijke delen elkaar overlappen gebruiken we bijvoorbeeld geel .
Plaatsen we nu over het rechter oog een rode filter dan zal het rechter oog de groene figuur niet waarnemen .
Plaatsen we over het linker oog een groene filter dan wordt de rode figuur niet waargenomen .
Zijn de twee figuren op de juiste manier perspectievisch t.o.v. elkaar verschoven dan ontstaat een 3D-figuur .
Het voordeel van deze methode bestaat erin dat we gans het beeldscherm kunnen benutten . We dienen echter over een bril, voorzien van een rood en groen glas, te beschikken .

Er bestaan ondertussen technieken die toelaten drie dimensionele beelden op te wekken door het beeld te onderwerpen aan vibraties , hierbij kan het 3D-effect worden waargenomen zonder enig hulpmiddel .

Laten we de tweede methode nader toelichten .
We dienen te beschikken over 4 kleuren .
Namelijk zwart,rood,groen en geel .
Dit kan bij de DAI-gebeuren met COLORG 0 3 5 14 .
Hierin is kleur 3 (rood) voor het rechterbeeld .
kleur 5 (groen) voor het linkerbeeld
kleur 14 (geel) voor de gemeenschappelijke delen .
Elk beeldelement dient te beschikken over twee bits die we afzondelijk moeten kunnen beïnvloeden .
Het rechterbeeld onstaat door de eerste bit te setten , het linkerbeeld door de tweede bit te setten .
Bij de gemeenschappelijke delen is nu zowel de eerste als de tweede bit op 1 gebracht .

	X1	X0	COLORG	
0	0	0	achtergrond	0
1	0	1	rechterbeeld in rood	3
2	1	0	linkerbeeld in groen	5
3	1	1	gemeenschappelijk in geel	14

Bij DAI onstaat het rechterbeeld door kleur 17 te gebruiken, hierbij wordt de eerste bit geset en behoudt de tweede zijn oorspronkelijke inhoud .
Het linkerbeeld onstaat door kleur 19 te gebruiken, waarbij de laatste bit wordt geset .

Het laatste probleem dat ons rest is de coördinaten te bepalen van het linker-en rechter beeldpunt XEL,YE en XER,YE . Om deze te kunnen vinden dienen de coördinaten van het weer te geven punt P te zijn gegeven (XP,YP,ZP) en de positie van de twee waarnemingspunten .
Het tafereel bevindt zich in het xy-vlak en de waarnemingspunten in het xz-vlak symmetrisch t.o.v. het yz vlak . De coördinaten van het rechterwaarnemingspunt zijn R(L,0,-V) , deze van het linkerwaarnemingspunt L(-L,0,-V) .

Het snijpunt van de rechte PR met het tafereel kunnen we vinden door de vergelijking van deze rechte op te stellen in de ruimte en de z-waarde gelijk aan nul te stellen .

$$\frac{x_{E_R} - x_R}{x_P - x_R} = \frac{y_{E_R} - y_R}{y_P - y_R} = \frac{z_E - z_R}{z_P - z_R}$$

$$\frac{x_{E_R} - L}{x_P - L} = \frac{y_E}{y_P} = \frac{+V}{z_P + V}$$

of

$$x_{ER} = \frac{V}{ZP + V} \cdot (x_P - L) + L \qquad y_E = \frac{V}{ZP+V} \cdot y_P$$

stellen we $\boxed{K = \frac{V}{ZP + V}}$ dan :

$$x_{ER} = (1 - K) . L + K . x_P \qquad y_E = K . y_P$$

stellen we $\boxed{D = (1 - K) . L}$ dan :

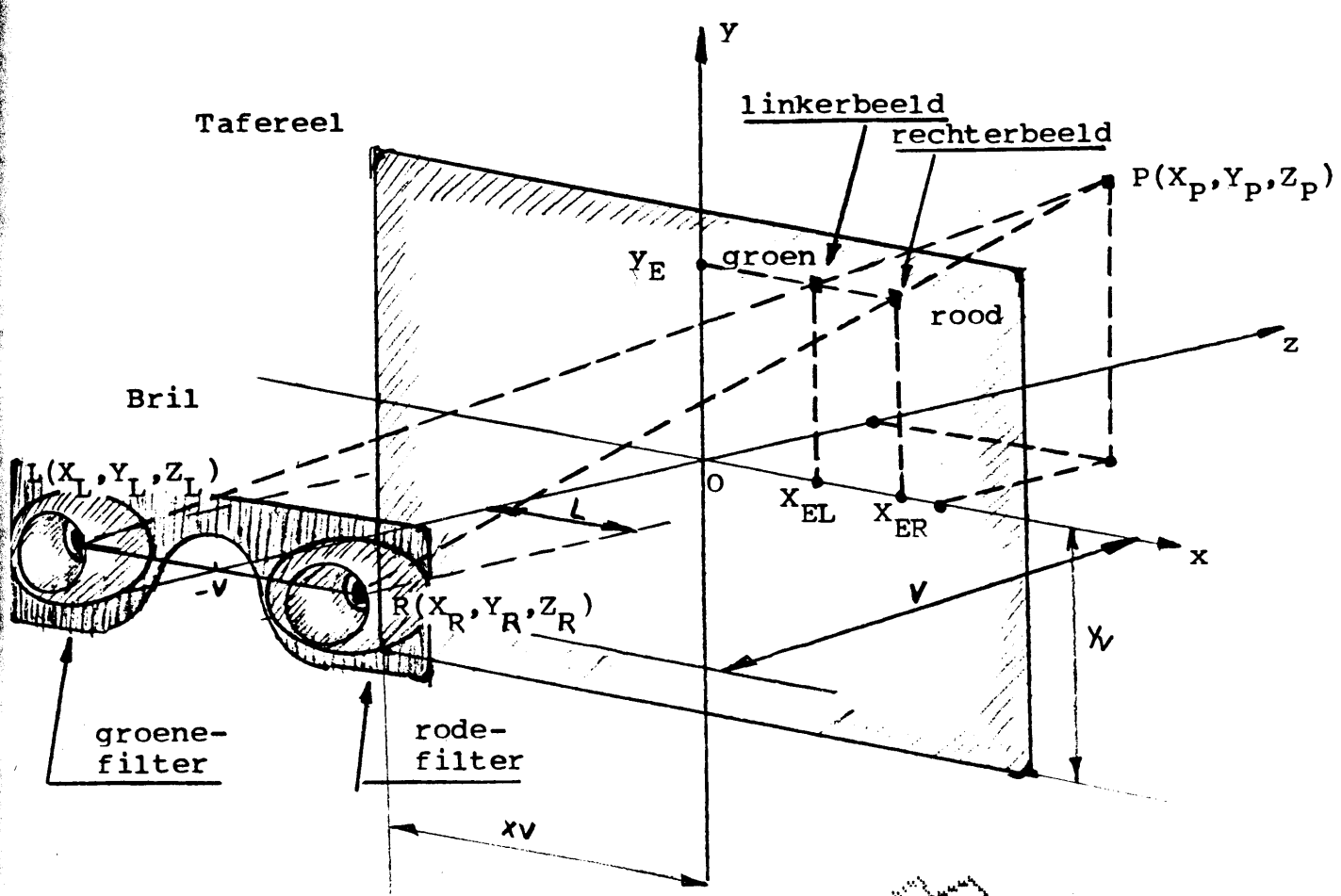
$x_{ER} = K . x_P + D$	$y_E = K . y_P$

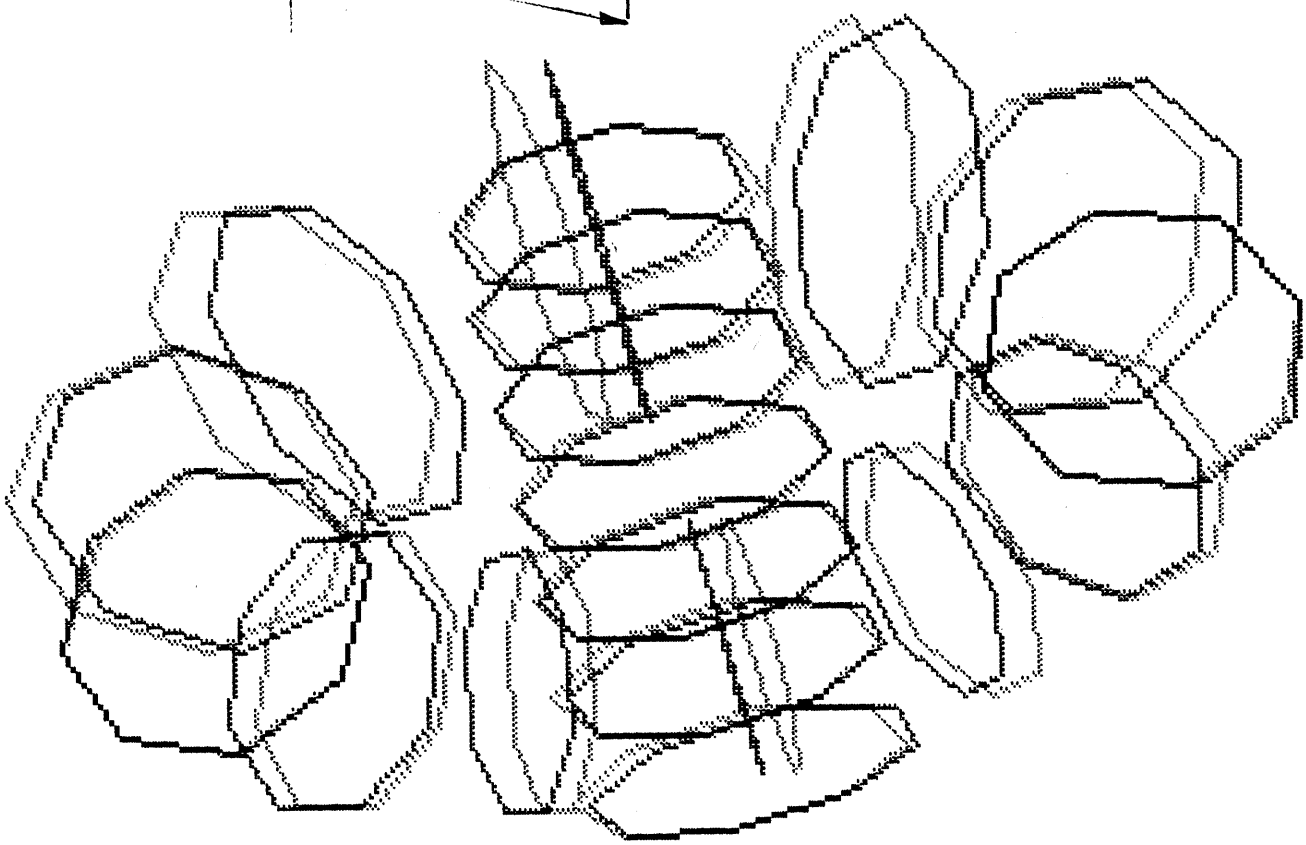
Voor het linkerbeeldpunt :

$x_{EL} = K . x_P - D$	$y_E = K . y_P$

Daar de oorsprong van ons assenkruis in het centrum van ons beeld is gelegen dienen we aan de gevonden waarden een translatie toe te voegen van XMAX/2 voor de x-waarden en YMAX/2 voor de y-waarden .

Nu dienen we de coördinaten van de gewenste figuur te genereren deze indien gewenst te onderwerpen aan de nodige translatie en rotatiebewegingen om uiteindelijk de waarden van XP,YP en ZP vast te leggen .
Daarna bepalen we de coördinaten van het linker-en het rechter beeldpunt zoals hierboven is beschreven . Passen eventueel een window toe indien bepaalde gedeelten van de figuur buiten het toegestane gebied vallen en wachten naar het resultaat dat we bekijken met onze bril .
De waarden van de afstand V tot het tafereel bedraagt 500 en deze van L = 25 indien XMAX=336 en YMAX=256 in MODE 6 .





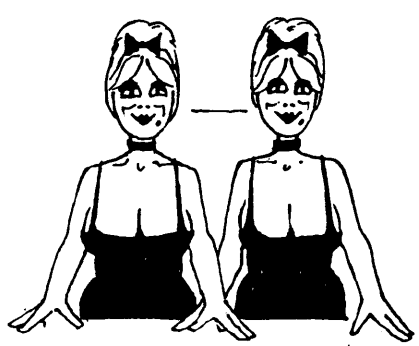
YELLOW

RED

GREEN

```
50    MODE 0
60    PRINT CHR$(12)
70    CURSOR 5,14:PRINT " D R I E   D I M E N S I O N E E L   T E K E N E N "
80    CURSOR 40,5:PRINT " J.ROELANTS":PRINT
100   V=500.0:L=25.0
600   INPUT " VERDRAAIING ROND X,Y,Z AS -> ";A,B,C:PRINT
620   A=A/57.296:B=B/57.296:C=C/57.296
640   GOSUB 4000
1000  MODE 6:MODE 6
1020  COLORG 0 3 5 14
1040  R1=90.0:R2=30.0:REM FIGUUR
1060  FOR II=0.0 TO 330.0 STEP 30.0
1070  I=II*PI/180.0
1080  FOR JJ=0.0 TO 360.0 STEP 45.0
1090  J=JJ*PI/180.0
1100  X=R2*COS(J)+R1:Y=R2*SIN(J):Z=0.0
1110  GOSUB 2500
1140  GOSUB 3000:GOSUB 2000
1180  NEXT JJ:NEXT II
1185  FOR YS=-60.0 TO 60.0 STEP 20.0
1190  FOR JJ=0.0 TO 360.0 STEP 45.0
1200  J=JJ*PI/180.0
1210  XS=R2*COS(J):ZS=R2*SIN(J)
1240  GOSUB 3000:GOSUB 2000
1250  NEXT JJ:NEXT YS
1400  GET%=GETC:GET%=GETC:GET%=GETC
1420  GET%=GETC:IF GET%=0 THEN 1420
1500  GOTO 50
2000  REM BEELDPUNTEN
2020  K=V/(ZP+V)
2040  D=(1.0-K)*L:KX=K*XP+168.5
2060  XER=KX+D:YE=K*YP+128.5
2080  XEL=KX-D
2100  IF J=0.0 THEN 2200
2120  DRAW XER,YE XERM,YEM 17:DRAW XEL,YE XELM,YEM 19
2200  XERM=XER:XELM=XEL:YEM=YE
2220  RETURN
2500  XS=X*COS(I)-Z*SIN(I)
2520  YS=Y
2540  ZS=X*SIN(I)+Z*COS(I)
2580  RETURN
3000  REM ROTATIE
3010  XP=AX*XS+BX*YS+CX*ZS
3020  YP=AY*XS+BY*YS+CY*ZS
3040  ZP=AZ*XS+BZ*YS+CZ*ZS
3200  RETURN
4000  AX=COS(B)*COS(C)
4020  BX=SIN(A)*SIN(B)*COS(C)-COS(A)*SIN(C)
4040  CX=COS(A)*SIN(B)*COS(C)+SIN(A)*SIN(C)
4060  AY=COS(B)*SIN(C)
4070  BY=SIN(A)*SIN(B)*SIN(C)+COS(A)*COS(C)
4080  CY=SIN(C)*SIN(B)*COS(A)-COS(C)*SIN(A)
4090  AZ=(-1.0)*SIN(B)
4100  BZ=SIN(A)*COS(B)
4120  CZ=COS(A)*COS(B)
4200  RETURN
```

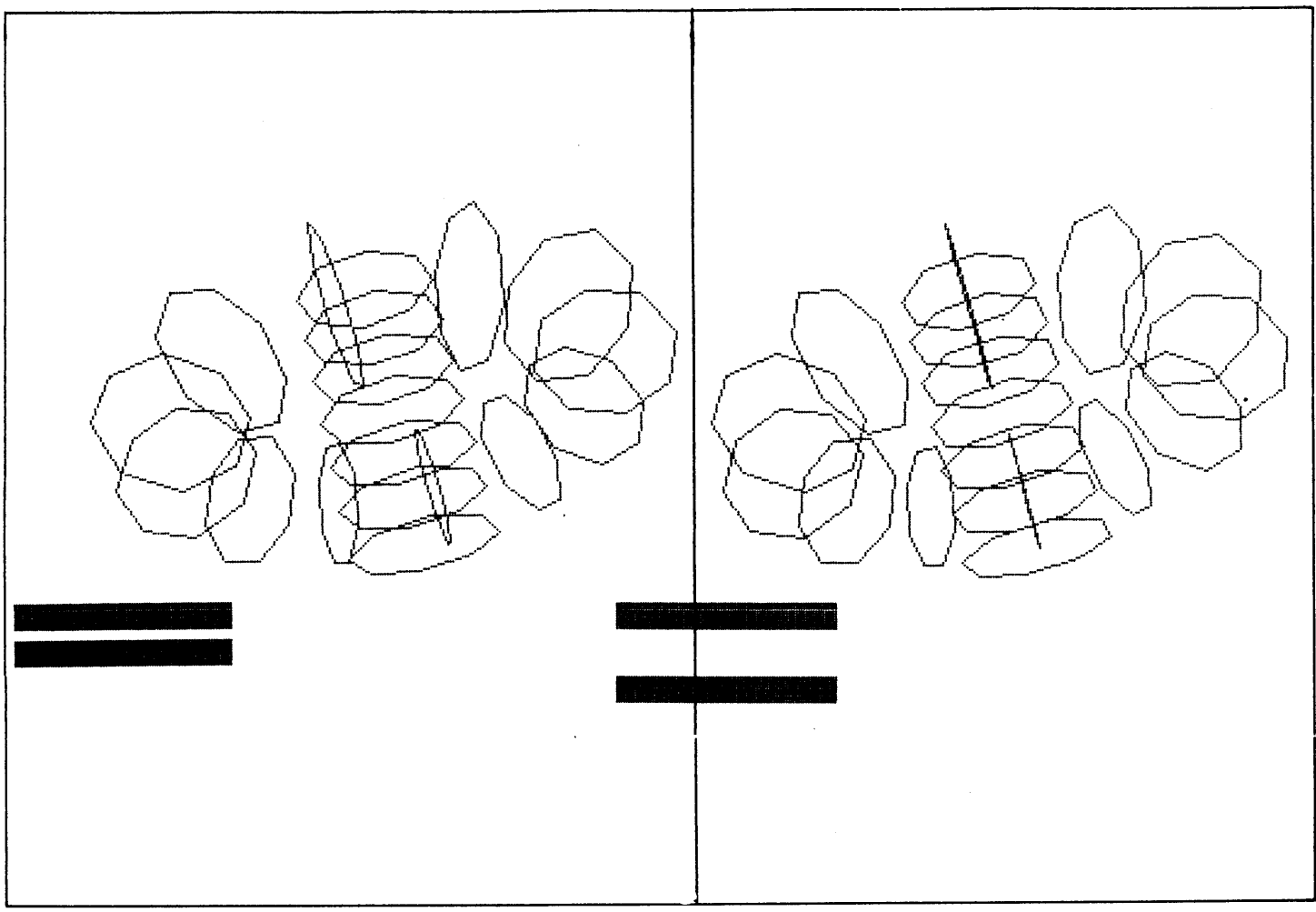
e.g. 30,30,30



Bij het waarnemen van het beeld dient men de helderheid van het beeld zo hoog mogelijk te nemen zodanig dat de groene en rode kleur even sterk doorkomen .

Men kan het beeld bewaren door een foto te nemen van het scherm of door twee screen coppy's te nemen .
Bij de eerste screen coppy drukt men enkel de tweede en vierde kleur af , bij de tweede de derde en de vierde kleur .
Nu bekomt men twee figuren waargenomen door het linker - en het rechter oog . Bekijkt men deze twee beelden door een stereo-kijker dan onstaat het drie dimensionele beeld opnieuw .

Veel genoegen met het waarnemen van de drie-dimensionele beelden .



program identification

title	:	PAINT
author	:	Frank Druijff
purpose	:	To fill a shape with a specified color
comment	:	a machine-language version will be available on TOOLKIT 5

* * * * p a i n t * * * * p a i n t * * * * p a i n t * * * *

De instructie PAINT is een grafische instructie die de DAI moest ontberen. De instructie die wel in de standaard instructieset van sommige andere machines zit kleurt een willekeurig vlak in de door de gebruiker opgegeven kleur. Wij wilden die mogelijkheid ook graag voor de DAI realiseren en dat is gelukt. Na wat gefilosofeer over een mogelijk methode kwam Hans Peters met de gouden tip. Het schrijven van een programma om deze tip te gebruiken kostte toen ook nog het nodig denkwerk maar was uit te voeren. Toen de basicversie goed bleek te werken (nog geeneens zo erg traag) kwam de volgende stap: het programma in machinetaal schrijven. Er was een aanzienlijke snelheidswinst maar toch bleef het verlangen naar meer snelheid. Die was mogelijk omdat mijn machinetaalversie gebruik maakte van de SCRN-routine in rom. Dat betekende dat de DAI bij elke aanroep van deze routine de x en y waarde controleerde op grootte en dan elke keer opnieuw de byte uitrekende waarin het gewenste punt stond. Dit is maar een keer in het begin nodig. Ik heb toen Nico Looije om hulp gevraagd en die heeft mijn versie grondig onder handen genomen. Een gemiddelde snelheidverbetering van 4 a 5 maal. Samen nog even de laatste hand aan de routine gelegd en de DAI heeft nu ook een PAINT.
Voor DCR en floppy bezitters moet het mogelijk zijn de routine zo aan te passen dat PAINT direct gebruikt kan worden of in een programma op de manier zoals DCR en floppy de 'nieuwe' instructies gebruiken.
De routine zal opgenomen worden in een nieuwe TOOLKIT-collectie.

Hoe werkt de paint ?

1) we kiezen een punt in de figuur die we wensen te kleuren.
2) we geven de volgende gegevens door aan de routine: de x en y van ons punt de randkleur van de op te vullen figuur en de gewenste vulkleur.
3) de routine begint naar de linkerrand te zoeken.
4) dan wordt vanuit dit punt de rechterrand gezocht.
5) er wordt een lijn getrokken in opvulkleur van linkerrand naar rechterrand.
6) er wordt een volgend punt op de rand gezocht vanwaar naar rechts een lijn (evt met lengte 1) getrokken kan worden.
7) als we nog niet aangekomen zijn bij ons uitgangspunt bij de rand dan worden de punten 5 en 6 herhaald totdat dit wel zo is en dan zij we klaar

Vooral punt 6 heeft het nodige denkwerk gekost. Hoe vindt je het volgende punt en hoe ga je de figuur rond. Het bleek achteraf erg simpel te zijn. Mensen die er zelf over willen nadenken moeten dan nu niet verder lezen.
We gaan als volgt rond (rechtsom): kijk of je naar boven kan zo ja doe dat en kijk dan of je naar links kan zo ja doe dat en kijk dan of je naar beneden kan en zo ja doe dat dan en kijk dan of je naar rechts kan en zo ja doe dat dan en kijk dan weer of je naar boven kan enz.. Kan je niet naar boven gaan dan naar rechts gaan, kan je niet naar rechts dan naar beneden, kan je niet naar beneden dan naar links. En ..... kun je niet naar links dan heb je een punt van waaruit je een lijn naar rechts kunt trekken. Goed geprogrammeerd zijn dit vier regels. (20,30,40 en 50) In regel 60 wordt gecontroleerd of we al klaar zijn en regel 70 wordt de lijn getrokken.

In het basicprogramma zit echter nog een belangwekkend deel: regel 100 t/m 170 Het is een zeer simpele, maar wel fraaie methode om een punt op het beeld te laten bewegen. Met de cursortoetsen gewoon een naar links,rechts,boven of onder maar met cursortoetsen en ingedrukte shift met stappen van tien. Dit aantal kan ook anders gemaakt worden. Dit programmadeel loont de moeite om op te nemen in een set standaardroutines.
Het basicprogramma voorziet verder in een mogelijkheid om zelf uw figuur te tekenen of de standaardfiguur te kiezen.(uit de DATA lijnen)

Nog wat laatste opmerkingen over de twee machinetaal routines. Eerste versie werkt in alle modes,maar controleert niets. De tweede versie is sneller en controleert op juiste gebruik kleuren en geeft OFF SCREEN als de beeldrand bereikt wordt. Het nadeel dat deze versie alleen in vier-kleuren modes werkt is niet een echte beperking, door kleurconflicten is het programma nauwelijks zinvol te gebruiken in de zestien-kleuren modes. Probeer de basicversie maar eens in een zestien-kleuren mode.

Al de oplossingen hebben echter een nadeel; als in de figuur nog een andere figuur ligt dan werpt deze een 'schaduw' naar rechts. Dit is nauwelijks op te lossen, tenminste niet op een zinvolle manier. Mogelijkheden zijn een tweede keer linksom rond lopen en dan juist naar links een lijn trekken maar dan is gedeelte tussen twee binnenfiguren weer ongevuld en het programma wordt met een factor twee vertraagd. Andere mogelijkheden als het 'oversteken' naar zo'n binnenfiguur leveren grote problemen op bij het bijhouden van terugkeer adressen. Een paar manieren om het probleem op te lossen.

1) trek een lijn tussen buitenrand en binnenfiguur. Alles wordt dan goed gevuld en trek dan de verbindingslijn opnieuw in de vulkleur.
2) teken buitenfiguur, kleur die, teken binnenfiguur, kleur met andere kleur.
3) Kies nieuw startpunt in het nog niet gevulde gedeelte en dan wordt tekening daar alsnog gevuld. (de binnen figuur is dan rand !)

Hierna het basic-programma en als u het intikt vergeet u dan niet eerst IMPINT te tikken ?

```
10    CLEAR 256:GOTO 400:REM PAINT / F.H. DRUIJFF 12/82
20    IF SCRN(X,Y+1)<>G THEN Y=Y+1:GOTO 50:REM BOVEN
30    IF SCRN(X+1,Y)<>G THEN X=X+1:GOTO 20:REM RECHTS
40    IF SCRN(X,Y-1)<>G THEN Y=Y-1:GOTO 30:REM BENEDEN
50    IF SCRN(X-1,Y)<>G THEN X=X-1:GOTO 40:REM LINKS
60    K=X-1:IF X=BX THEN IF Y=BY THEN IF F=1 GOTO 80:F=1
70    K=K+1:IF SCRN(K,Y)<>G GOTO 70:DRAW X,Y K-1,Y W:GOTO 20
80    END
100   C=SCRN(X,Y):DOT X,Y 23:H=GETC:DOT X,Y C:IF H=0 GOTO 100
110   IF H<16 GOTO 170:IF H>23 GOTO 170:V=H/20*9+1
120   ON H MOD 4 GOTO 140,150,160
130   Y=Y+V:IF Y<YMAX GOTO 170:Y=YMAX:GOTO 170
140   Y=Y-V:IF Y>0 GOTO 170:Y=0:GOTO 170
150   X=X-V:IF X>0 GOTO 170:X=0:GOTO 170
160   X=X+V:IF X<XMAX GOTO 170:X=XMAX
170   RETURN
300   PRINT "Move blinking dot. If inside field to color press space ."
310   GOSUB 100:IF H<>32 GOTO 310:X=X+1
320   X=X-1:IF SCRN(X,Y)<>G GOTO 320:X=X+1:BX=X:BY=Y:F=0:GOTO 60
400   G=14:W=1:COLORG 0 W G 15:MODE 4A:REM INITIALISATION
410   INPUT "Standard/Own drawing ";VE$:PRINT
420   X=33:Y=33:IF VE$="O" GOTO 500
430   READ P,Q:IF P+Q=0 GOTO 300:DRAW P,Q X,Y 22:X=P:Y=Q:GOTO 430
500   GOSUB 100:IF H=13 GOTO 300:IF H<>32 GOTO 500
510   IF F=0 THEN DOT X,Y 22:F=1:P=X:Q=Y
620   DRAW X,Y P,Q 22:P=X:Q=Y:GOTO 500
900   DATA 147,12,147,45,90,45,80,31,80,60,120,70
910   DATA 85,70,75,60,66,45,45,55,33,33,0,0
```

veel plezier
Frank H. Druijff

UNIVERSELE PC. DCE-BUS INTERFACEKAART (GICC

Deze kaart heeft hetzelfde doel en dezelfde logische opbouw als de DCE-F module van DAI, met dit verschil dat bij deze kaart alle aansluitingen in volgorde liggen zoals bij de personal computer.(de industriele kaarten hebben een afwijkende volgorde.)
Het gevolg hiervan is dat deze "GICC" (General Interface Control Card) met een kaartadapter rechtstreeks op de DCE-bus van DAIpc kan worden aangesloten door middel van een gewone bandkabel als het een enkele kaart betreft.
Voor het aansluiten van meerdere kaarten doen we een beroep op de busprint van Elektuur (EPS80024).
Iedere busprint biedt plaats voor max 5 interface kaarten, terwijl meerdere busprinten plaats bieden voor max. 15 kaarten volledig in BASIC programmeerbaar met de OUT I,J en de INP(I) instructies.
Voor de niet-ingewijden in de DCE-BUS techniek verwijzen we naar DAInamic No 1 p 3,4,5 en 6 , of naar "THE BEST OF DAInamic" p 4,5,6 en 7.
De GICC is een eurokaart met een 64-polige DIN 41612 connector een volledige adresdecodering, 3 8-bits I/O poorten met een command register (8255A), een wrapping area van 110mm x 110mm en alle DCE-BUS signalen.
Het kaartadres is door middel van 4 DIP-switches door de gebruiker zelf instelbaar (1 tot 15).
Van de 64-polige DIN connector zijn er nog een aanzienlijk aantal lijnen onbezet, die we kunnen gebruiken om onze bus-structuur tussen de verschillende interfacekaarten verder uit te bouwen.(oa bijkomende voedingsspanningen +/- 15V)

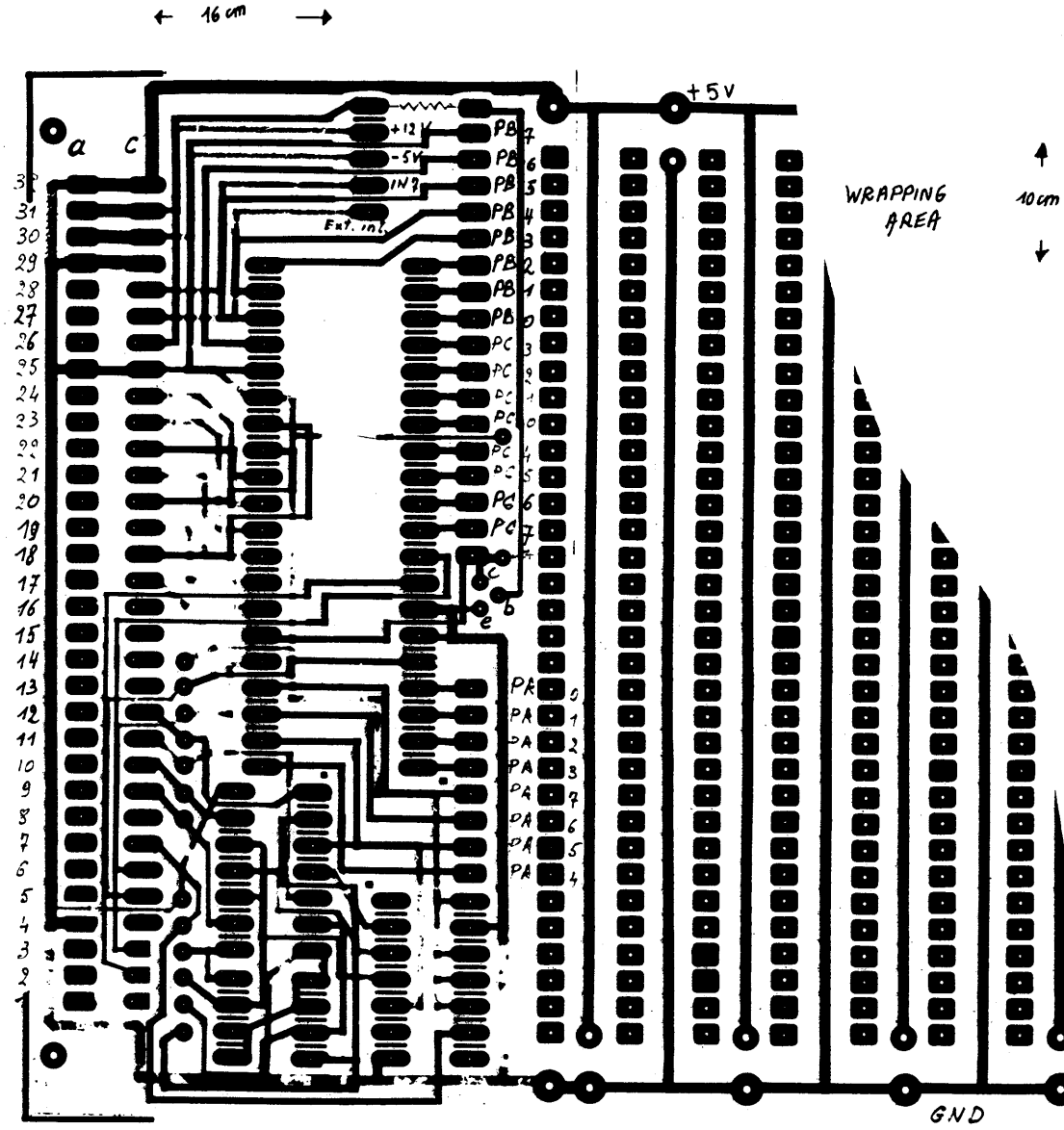
THE DCE-CONCEPT

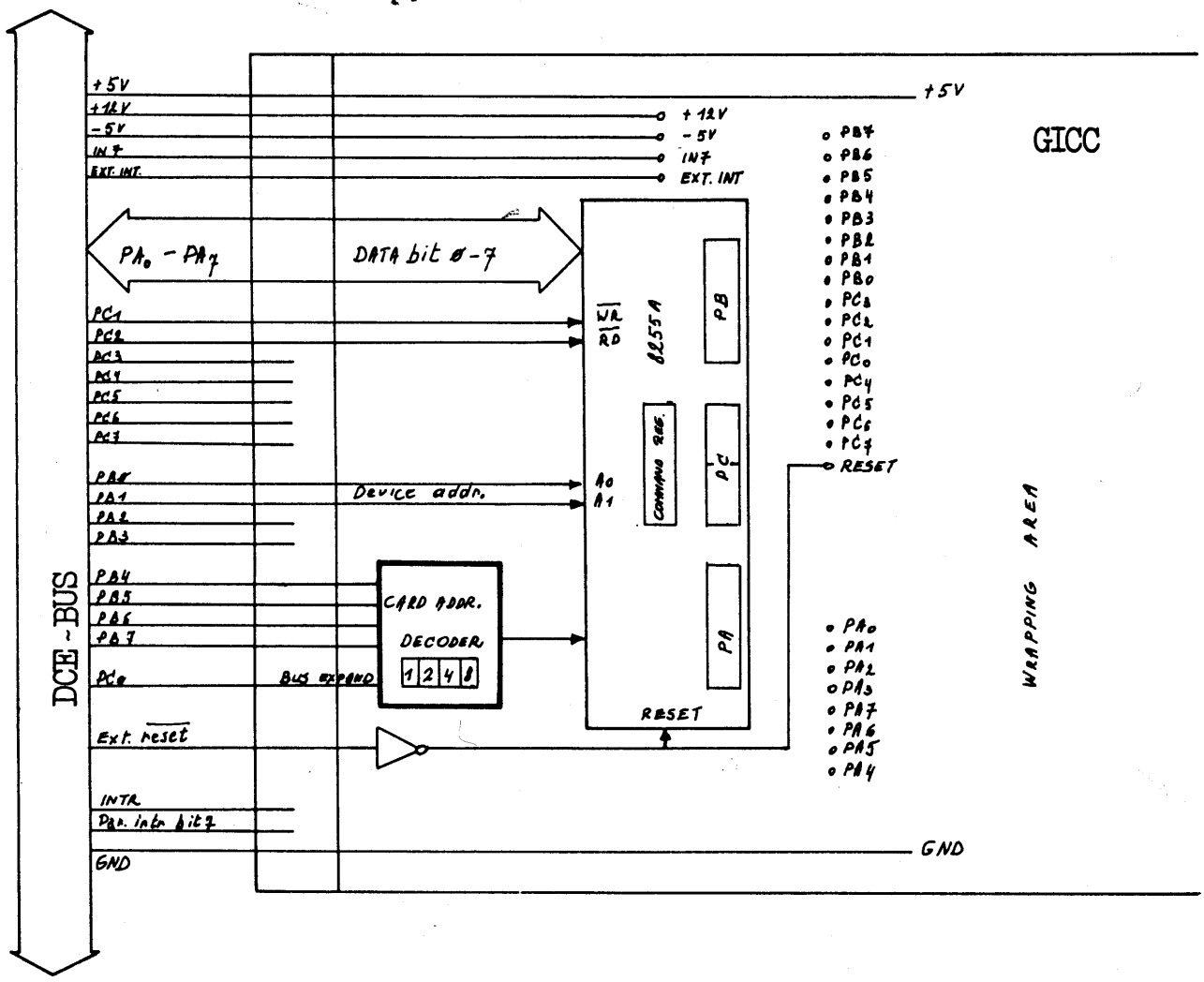
The DCE-BUS provides an easy way for the exchange of data and control information between the DCE-BUS of the personal computer and the interface card(s).
The cards are driven by the GIC (8255) on the personal computer, configured to provide 8-bit input/output transfer, read & write signals, two external interrupts requests and eight address control lines for selective access to the cards on the bus.
Each card is plugged into the DCE-BUS (through Elektuur 80024 motherboard) and given a unique address via 4 dipswitches on the card.
Read and Write subroutines (IN/OUT) are provided (in DAI-BASIC) to simplify the tranfer of data between the personal computer and the interface cards.
GIC PORT 0 is used as a bidirectional data path for data transfer between the computer and the cards.
GIC PORT 1 is used to specify the address of the connecting card.
GIC PORT 2 issues the control signals to complete the transfer logic.
PORT 0 must be alternated between input and output by software command.

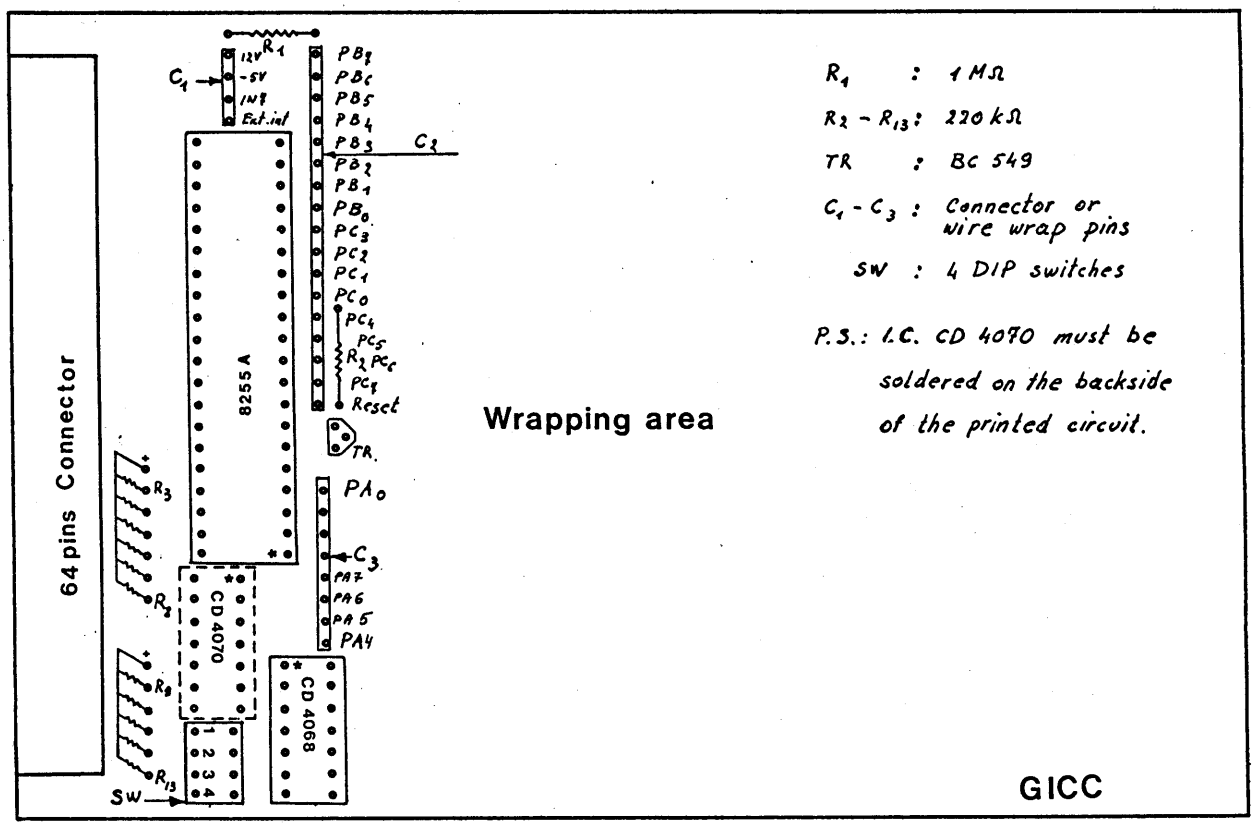
For more information, please consult :

" DCE MICROCOMPUTER SYSTEM DESIGNER'S HANDBOOK"

PRINT EPS80024 (Elektuur)		pin	DAIpc Output	DCE-BUS Function	CONNECTOR
c	a				
1	1	1	+5 V	+5V	32
2	2	2	+12 V	+12 V	31
3	3	3	-5 V	-5 V	30
4	4	4	GND	GND	29
	5	5	IN7	IN7	28
	6	6	EXT.INTR	EXT.INTR.	27
	7	7	EXT.RES.	EXT.$\overline{\text{RES}}$.	26
8	8	8	N.C.	+5 V	25
	9	9	PA4	data 4	24
	10	10	PA3	data 3	23
	11	11	PA5	data 5	22
	12	12	PA2	data 2	21
	13	13	PA6	data 6	20
	14	14	PA1	data 1	19
	15	15	PA7	data 7	18
	16	16	PA0	data 0	17
	17	17	PC7	N.U.	16
	18	18	PC6	N.U.	15
	19	19	PC5	N.U.	14
	20	20	PC4	N.U.	13
	21	21	PB7	CARD AD.3	12
	22	22	PB6	CARD AD.2	11
	23	23	PB5	CARD AD.1	10
	24	24	PB4	CARD AD.0	9
	25	25	PB3	DEV. AD 3	8
	26	26	PC0	BUS EXP.	7
	27	27	PC1	$\overline{\text{WR}}$	6
	28	28	PC2	$\overline{\text{RD}}$	5
GND	29	29	PC3	N.U.	4
INTR.	30	30	PB0	DEV. AD 0	3
P.INTR.7	31	31	PB1	DEV. AD 1	2
32	32	32	PB2	DEV. AD 2	1
		33	INTR.	N.EXIST (3a)	
		34	PAR.INTR 7	N.EXIST (2a)	




GICC
DCE-BUS
+5V
+12V
-5V
INT
EXT. INT.
PA0 - PA7
DATA bit 0-7
PC1
PC2
PC3
PC4
PC5
PC6
PC7
PA0
PA1
PA2
PA3
PB4
PB5
PB6
PB7
PC0
Ext. reset
INTR.
Per. int. bit
GND
Device addr.
BUS expand
CARD ADDR. DECODER
1 2 4 8
WR
RD
8255A
PB
PC
PA
COMMAND REG.
A0
A1
RESET
+5V
+12V
-5V
INT
EXT. INT
GND
PB7
PB6
PB5
PB4
PB3
PB2
PB1
PB0
PC3
PC2
PC1
PC0
PC4
PC5
PC6
PC7
RESET
PA0
PA1
PA2
PA3
PA7
PA6
PA5
PA4
WRAPPING AREA


64 pins Connector
C1
12V
-5V
INT
Ext.inf
R1
PB7
PB6
PB5
PB4
PB3
C2
PB2
PB1
PB0
PC3
PC2
PC1
PC0
PC4
PC5
PC6
PC7
Reset
TR.
8255A
R2
R3
R13
SW
1 2 3 4
CD 4070
CD 4068
PA0
C3
PA7
PA6
PA5
PA4
Wrapping area
R1 : 1 MΩ
R2 - R13 : 220 kΩ
TR : Bc 549
C1 - C3 : Connector or wire wrap pins
SW : 4 DIP switches
P.S.: I.C. CD 4070 must be soldered on the backside of the printed circuit.
GICC

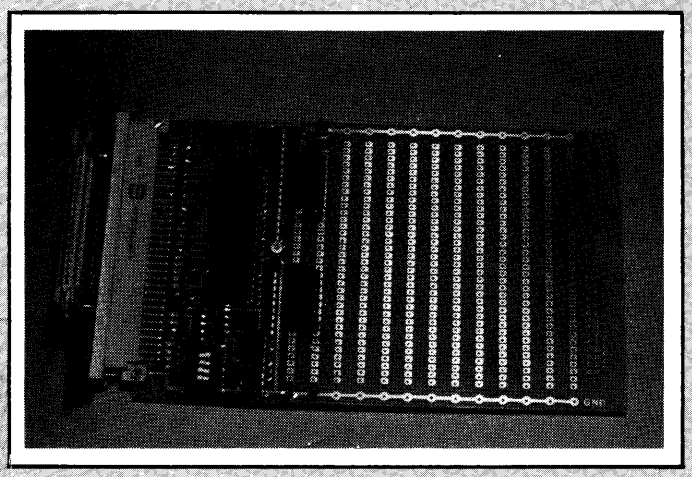


photo 1 Assembled interface-card

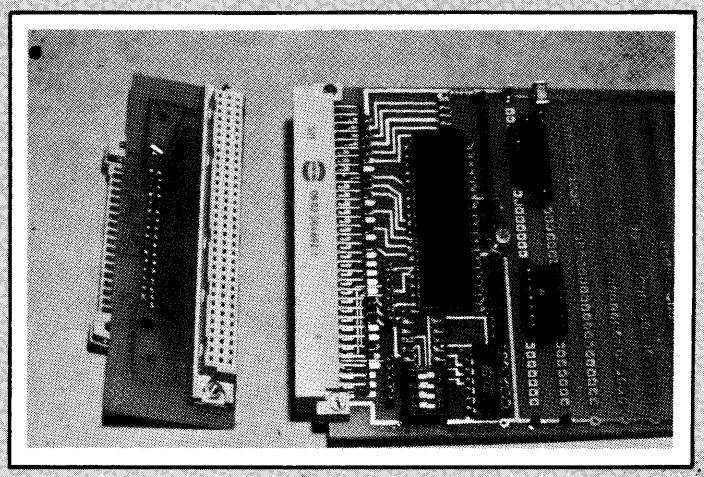


photo 2 Assembled interface-card + adapter card

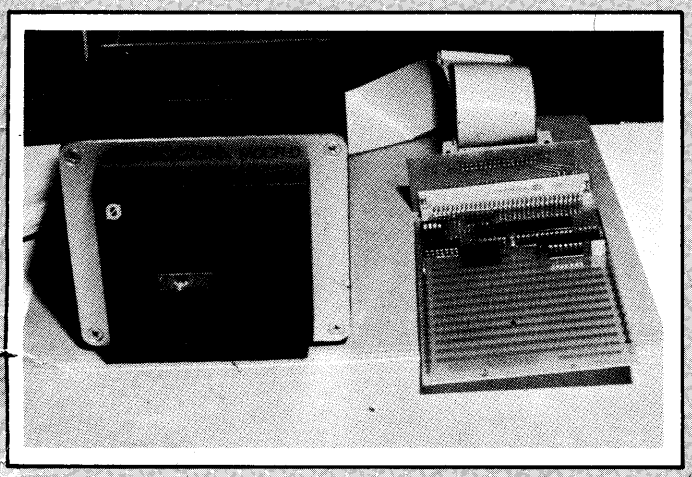


photo 3 MDCR + interface-card

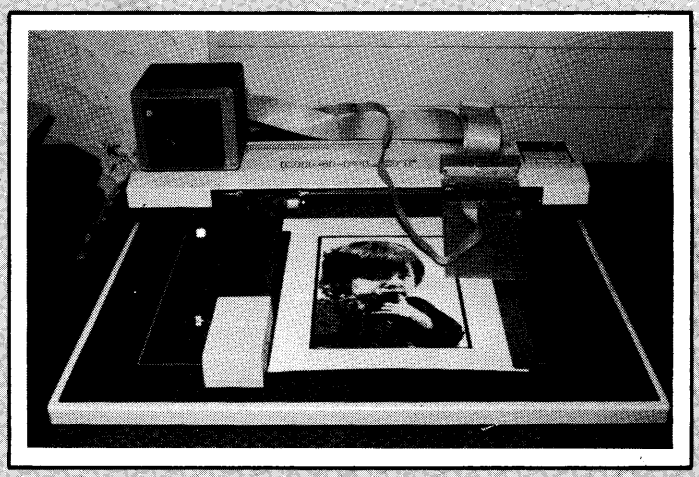


photo 4 MDCR + plotter, together on the DCE-bus

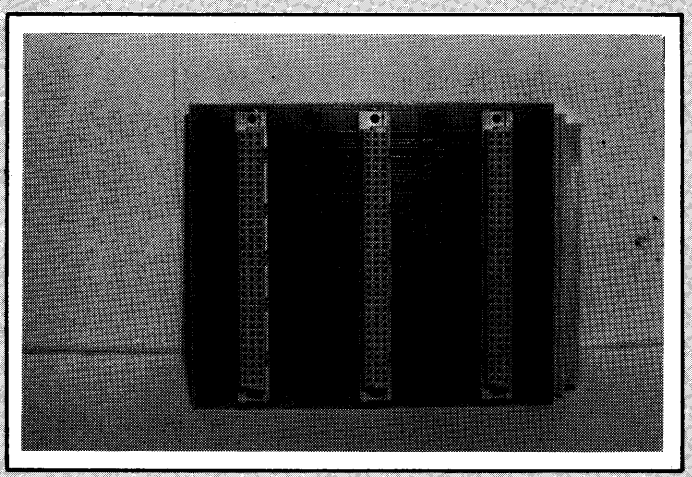


photo 5 Elektuur EPS80024,
motherboard for interface-cards

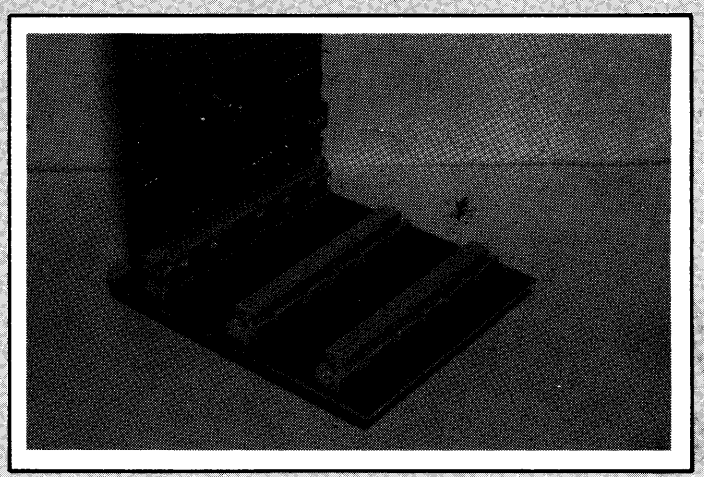
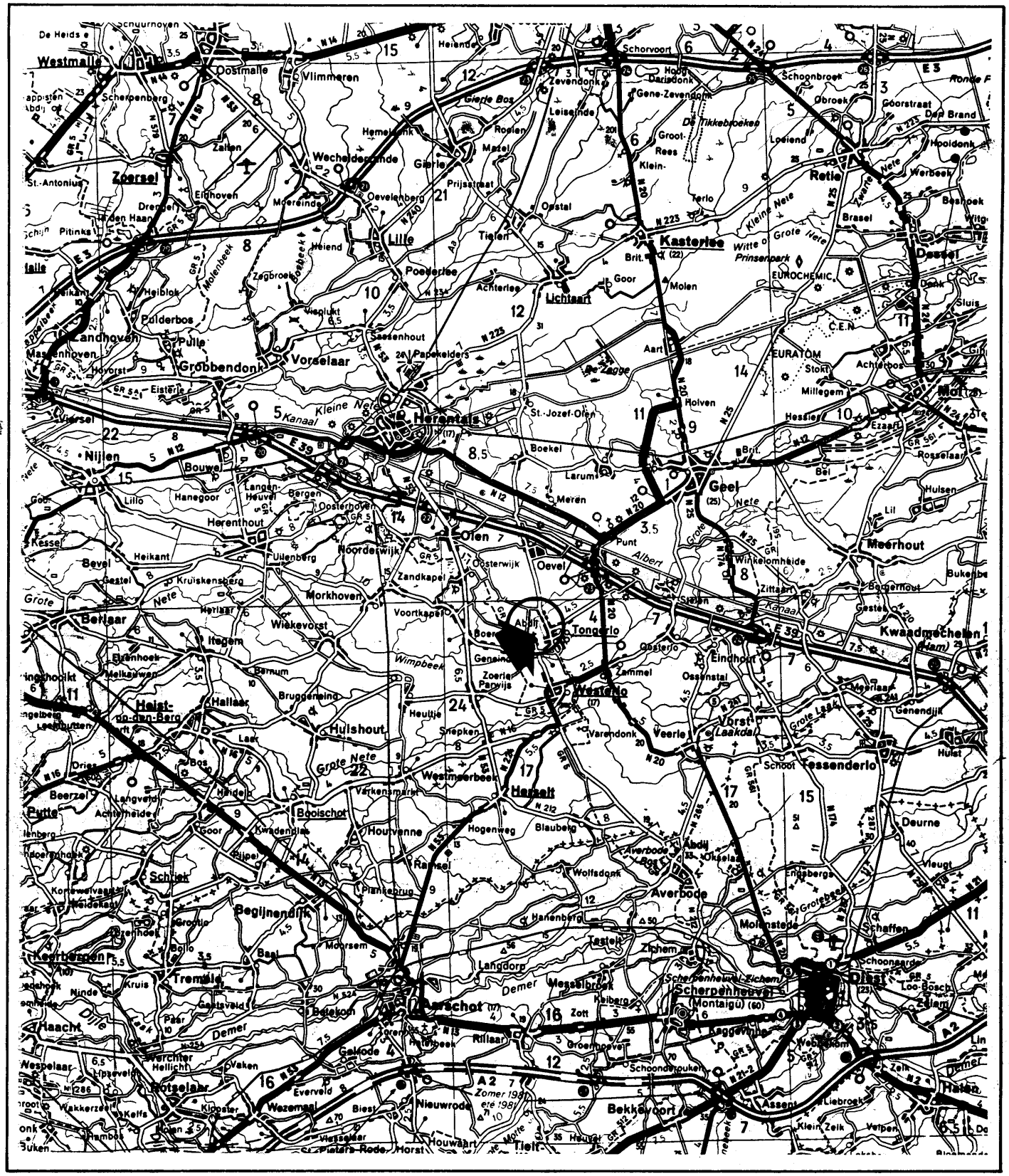


photo 6 Elektuur EPS80024 with 1 interface-card



DAInamic MEETING :

on 9th April 1983 10.00 - 18.00 Hr
in TONGELSBOS Bosstr. 2 3180 WESTERLO