volume three/ June 1st, 1984 number 14

bi-monthly periodical of the Exidy Sorcerer Gebruikers Groep

a translation in English of the original Dutch version

The L O G I C partner to a Sorcerer

Subscriptions : per annum f. 27,50 (Europe)
f. 32,50 (other countries)

(more information: page 2)

Send your copy to : redaktie ESGG
p/a postbus 510
1000 AM AMSTERDAM Holland

Coordinator foreign relations: C. Boone
Stationsplein 26
B-9100 LOKEREN Belgium

****

CONTENTS OF THIS ISSUE

****

EDITORIAL STAFF

chief editor : Welmoed J. Jonker.
hardware-editor : Aad van Duijvenbode.
software-editor : Kees van Duijvenbode.
general editor : Don Siahaya.

****

SUBSCRIPTION.

You become a subcriber when remitting the fee due to postal check account number 5368539 of ESGG at Lopik, The Netherlands, referring to 'subscription periodical'. Subscriptions run from June 1st of the current volume.

SUBSCRIPTION-ADMINISTRATION.

Send your change of address and complaints about the delivery to:
Administratie ESGG periodiek
Prins Hendrikstraat 3d
3071 LG ROTTERDAM Netherlands

****

ADVERTISEMENTS.

macro's: exclusively for business:
acquisition: M. Sanders
Richard Wagnerlaan 25
2253 CB VOORSCHOTEN.

micro's: Only open to private persons.

Size : -a line of text consists of 66 characters or spaces.
-there is a maximum of six lines for each advertisement.
Cost : the cost per two lines of text is Dfl. 3,00.
Entering: send your advertisement to the editor before the first day of each odd month. Mention also the total of 66 character lines and your account nr.
Payment : Remit the amount due to postal check account number 5368539 of ESGG at Lopik, referring to 'micro's'.

If your remittance is not received before the first day of an odd month insertion in that number is no longer possible!

****

****

SOFTWARE COLLECTOR.

If you like to offer yourself developed so-called public domain software to your fellow members, send it on cassette to:
Wim Warning
Vogelweide 83
3815 HE AMERSFOORT The Netherlands

****

ESGG-SERVICE

Prices are including postal rates for second class mail all over the world. The language of explanations in the software is mainly Dutch! Because of transport risks no mail-order possible for diskettes! Advance payment and collect at Sorcerer Days possible!

Ordering : only by means of postal check to account number 5368539 of ESGG at LOPIK, referring to: ESGG-service.

in your order state name and quantity of the desired article.

you do not receive an acknowledgement of your order.

if the article is not in stock or no longer available THEN you will be informed so!

Ordering diskettes: see restriction above.
Catalog obtainable from CP/M-users group Netherlands.

Available formats are 77 tracks hard- and softsectored, 40 and 30 tracks softsectored. The last-named two formats are 2 and 3 disks respectively. We always send the mentioned quantity of diskettes to you (possibly only formatted).

Non ESGG members and non subscribers pay Dfl. 10,= extra per vol.

Guarantee: Electronic articles from ESGG are subject to guarantee for proper operation. ESGG is not liable for damage caused by incorrect installation by others than the official technicians at Sorcerer days.

Below is a list of articles available at this very moment:

name article (prices a piece!)	Sorcererday-collect	by mail
1. Collect-cassettes with various programs (nrs. 1 t/m 15).....	Dfl. 7,50	Dfl. 15,00
2. Collect-disks *) with various programs, per volume: 77 HS/SS.....	Dfl.25,00	
40 SS	Dfl.30,00	
30 SS	Dfl.40,00	
*) see: ordering)		
3. ESGG diskettes nrs 1-3	as in pt. 2	
4. Eprom Basic EXTension (version 8) with description of installation and manual		Dfl. 35,00
5. Manual Bext		Dfl. 5,00
6. Invers video print (assembled).....		Dfl. 20,00

****

INPUT
a column to ask questions and also to give your opinion or comment.

If you have a problem, describe it as clear as possible and send it in a post-paid envelope to the editor. Our team then will try to find a solution. We claim the right of publication for question and solution in our magazine.

****

A NEW SPRING....

At the beginning of a new volume it is like leaving winter behind and entering spring. That is, in this country one does not observe much difference between winter and spring or other seasons (***although...***). However, you do not have to blame us for having bad weather!

A Dutch proverb says as much as "A new spring brings a new sound". We do not issue a spoken magazine, but you may have noticed in the time passed, that we are experimenting eagerly with embellishments within the range of the possibilities of the equipment at our disposal! Showing off with a green "ESGG" logo we entered prudently into spring and now we carefully take another step: The "bold" print of the Epson MX-80 has been replaced by the delicate but more clear types of the JP-80A.

To you judging this being an improvement or preferring the bolder characters of the Epson. We feel that the advantage of the more clear character is to be preferred over the disadvantage of the type being a little smaller.
With the purchase of this printer with a range of possibilities, and having an improved wp-program to work with, we are able to offer to you a text that is better readable.
And with your help, by sending in articles and programs, the editor's team wil' **take care** that this periodical remains at the level you are used to.

As has been mentioned before, the production of the Sorcerer is to be terminated this year. Is this the end of our group? We guess it is not! We have a users group that is still growing steadily, either in number of members as well as in number of subscribers, even from abroad! Let us all use the knowledge that is gathered by Exidy users and have as much hobbyists as possible profit by that knowledge! In being active and engaged in maintaining the ESGG you will find that we are able to continue endlessly: For you and by you!

Consider that things that can be defect, are not necessarily lost! If something inside your Sorcerer is damaged, you can be sure it can be cured, especially concerning the electronics! Inside the users group, commercially as well as privately there is enough knowledge to help you into the saddle in no time. Use that knowledge and do not throw your Sorcerer away!

Dear Sorcerers, show your best side in this new year of the periodical and share your knowledge with us! I am sure that you will discover eager apprentices in us!

Welmoed Jonker.

****

INFO.

* *Dear Readers, up to now nobody reported to the secretary to replace Jan van Dijk, or to assist the editor's team!*
  Is it that there is nobody, having time available as well as experience and willing to contribute to the useful work that is done by board and editor for you all? I say no, am I right?
  If you do not know whether you are able, well, make this not the reason for not calling! Give it a chance.... we, and the other members and subscribers too will be grateful to you, really!

* Now that we are discussing help for the board: As you probably have noticed, there is hardly ESGG news in the yellow pages of HCCN. The explanation is that there is also no correspondent for HCCN. If you do not like to be a genuine board-member, or when not wishing to belong to

the editor's team, maybe a task like this is something for you? Do call Charles Netteler on the subject!

* Hello out there: Some-one there being a hardware specialist enough to be called to the task of co-ordinator hardware of ESGG? It is -not regarding the matter- really not that hard and difficult a job!! Who is going to be ...?

* Spending an entire page reporting on the Sorcerer Day is getting useless now: Most of you have been there anyway! This time some 1300 crowded our half yearly event.
We thank the technical schools of Tilburg and Alkmaar for their contributions, the automated warehouse and the harbour crane. We expect to be their host in future again.

* At this very Sorcerer Day we welcomed our 1024th ESGG member! The lucky one, mr. J.J.M. Mesman from Breda received from our chairman Floor Vogelaar a matrix printer, donated by Caicom from Gouda.

* Our Wim Warning, the collector of the programs, is going to move to a better site in our small country: Amersfoort! In his column at page 2 you will find his new address. Again he is waiting eagerly for new programs!!! Please show to him some of you(r efforts)! Thanks a lot, all of you!

* At the CP/M day of April 7th the ESGG again released a diskette. This diskette has been named ESGG volume 3. This diskette can be ordered for in the usual manner.

* From mr. De Kort, who implemented ZCPR2 to our Exidy in versions for 48, 52 and 56K, we got word that the 48K version as being on ESGG volume 3 has some bugs thus not allowing you to work with it. Those among you that need the 48K version, are invited to return the diskette to the secretary of the ESGG Foundation, whereupon a diskette holding the correct version will be returned.

* There is a possibility that you, user of 77 tracks softsectored/512 bytes disksystems, may have purchased a not so well readable disk at the CP/M day of April 7th. If so, please return that very disk to Charles Netteler and you will receive in return a correct one!

* You do not have to be an author, to appreciate the hint of one of our members: When writing an article for this here periodical about changes made in a certain (system) program, then do mention in **what** version these changes were made! This is guarding us from a lot of trouble and puzzling work!

* And then there is something about our repeting award: Sending in his article about the disasters CP/M is able to cause, the author

A.W. van de Ven
Roer 85
9733 AJ GRONINGEN,

truly earned our award for the ..... article from number 13! Congrats to you, for your contribution, Antoine.

* From Charles Boone is the message that the publisher of the ISIS magazine asks Dfl. 65,00 for a year's subsription of 12 numbers by air. Now you know!

* **DID YOU PAY FOR YOUR VOLUME 3 SUBSCRIPTION YET? No?... Then do it NOW! Or do you wish to be cut off of all coming Exidy Sorcerer information?**

FROM OTHER MAGAZINES.

* This column is a little behind, but we will keep you informed as well as is possible about things published in other magazines.

* Databus nr. 5: Several articles about disks and drives, hard as well as floppies. The return of magnetic tape as a medium for storage is also in this issue. Are you considering RAM-disk? There is an article about it too.

* PCM nr. 3: Some old and new about the inkjet printers from Tandy and Siemens; the Juki 6100 daisywheel printer. Working with calculation programs (spread-sheets.... yech!! who invented that name; makes one think of sandwich covers) are reviewed, although not deeply, and so they go over to the little brother of DBASEII. PCM is not PCM when not having something on hard disks!

* PCM nr. 4: Tests of two little printing machines, the CE50-BT by Micro Plus and one by Trend: JP80-A. Readers that do not own a printer will find a helping hand in making up one's mind. Also an article about another storage medium: the Borsu 10x10. Making window programs is in nowadays; this article shows the inside of VisiON. Finally a piece on the Klokhuis (Core) Foundation that make believe this form of governed organization for a computer club is their private invention!

* Elektuur 247, May 1984 has a good looking design for a switching power supply with a maximum current of 5 Amps. Also they show a simple diagram for testing diskdrives. This test is a manually step by step search for a suspected malfunctioning drive.

****

INPUT.

* Mr. Gobets from Uithoorn purchased from an office inventory a printer. This printer has been manufactured by Axiom Corp. from Glendale U.S.A. and bears the identification IMP2AA2, IMP miniprinter. The printer is capable of handling A4 paper; the transport method is frictionfeed. As this machine has been transferred without manuals, he likes to know if there is someone among the members who has the missing information of this printer. If you are able to help, his address is Colijnlaan 138.

* Our very own Hermine Bakker also got a problem. Her old printer faded away definitely and incurably. As a successor she purchased an Epson FX-80. She now explains her trouble:
With my FX-80 printer I now try out various graphic possibilities. This next program:

```
10 (L)PRINT CHR$(27);":";CHR$(0);CHR$(0);CHR$(0);
15 (L)PRINT CHR$(27);"%";CHR$(1);CHR$(0);
20 (L)PRINT CHR$(27);"&";CHR$(0);
25 (L)PRINT "@@";
30 (L)PRINT CHR$(139);
35 (L)PRINT CHR$(255);CHR$(0);CHR$(129);CHR$(0);
40 (L)PRINT CHR$(129);CHR$(0);CHR$(129);CHR$(0);
45 (L)PRINT CHR$(129);CHR$(0);CHR$(255);
50 (L)PRINT "This is a square ---> @"
```

one should have a nice square.
Now is strange thing: with Standard Basic through parallel driver it is successful, but NOT in EXBASIC; you then find a 6-dot-high filled block. When I call the parallel driver from EXBASIC the eighth bit draws a fine line above this block, and the text output now turns into italics. I guess this is caused by EXBASIC. Who is the brains who knows

the inside of EXBASIC so well that he (she?) can solve this mistery for me? Are there other FX-80 users who have found solutions to this?

* Mr. J.E.A. van der Heijden from Best likes to lay his hands on hardware drawings from the model I Sorcerer. He already has the hardware manual but the drawings were not in it. The well-known companies had (the usual) deaf ear, so.... If you are able to help out him as well as us, please inform the editor. We thank you!

****

BASICODE AND WORDPROCESSOR.

People have a habit of saying "Want something done? Well there is always someone or someway to do it". Computer hobbyists and especially those that also develop programs, usually create a solution in case a certain program does not have such an option. So does mr. T. Huisman from The Hague. He thought a Basicode listing is easier to be read and changed from a word-processor file. Basicode itself does not supply that option, so he 'simply' created such a routine:

With Basicode one can go many a way when considering that a file in Basicode is nothing more than a string of characters with a beginning and an end, marked respectively by control characters 02H (STX=start of text) and 03H (ETX=end of text). In what way such a file has been created is of less importance. Usually the write routine works in such a way that all that is being recorded on tape in Basicode, also is being copied to the screen; or said in a different way: all video output is directed to the Basicode file when the write routine is called. That video output not necessarily has to be a listing. Another thing is that the read routine does put the file read properly in memory, however it also passes it directly to the Basic interpreter. Now when the interpreter does not get a Basic program, the screen is rapidly filled with syntax errors.

Therefore lines too long are not allowed in a listing and this is the reason that the recently broadcasted Tiny Pascal programs only can be read by Tiny Pascal.
To avoid these problems I happened to have created programs like BCLST (Basicode lister) and BCL48 (a 48K autostart version of BCLST) to get the file received in an unmutilated way to the screen, or to the printer (if I am right, they ought to be on tape ESGG nr. 13). Making some minor alterations one can make the read routine in this programs stop as soon as the entire file is read, allowing an inspection of the file present in memory. The address of that file (from 1000H onwards) is more or less randomly chosen. A smart ID (you have to wait a little longer for those) is to choose 080EH as a starting address. This happens to be the address where the files for the WORDPROCESSOR start and whereelse a string of ASCIIs do belong, do they not?

Saying so is doing so and the result is the program BC2WP, that however only works when having loaded the Basicode routines **first** from FE00H to FFFFH. When having connected the Basicode interface and the cassette recorder 1 with the Basicode file is READY, one simply starts by hitting the RETURN key to start reading. Next the file is displayed on screen. This is an 'extra' but one has to search the entire file for the end, so...
To write a Wordprocessor file to cassette you have to mention the number of the recorder to which this file is to be written (when using two recorders a simple RETURN is sufficient). To use this tape in the Wordprocessor you return to monitor by command X and then read the tape with >LO. PP is

returning you into the wordprocessor and.... there it is!
You are now able to edit as you please, print and ehhh.. d***! returning to Basic happens to be a problem yet, that we have to think over some time! Well, the name of such a program is going to be WP2BC, of course, so we will have a start!
The program looks like this:

```
  0 REM Load with >LOG
100 :
110 M=256*PEEK(-4095)+PEEK(-4096)-111
120 IF M>32767 THEN M=M-65536: REM     Monitor inputbuffer
130 :
140 C$="": R$=" 2": V$="": I=2062: REM 2062=080EH, start wpf
150 :
160 POKE -282,8: POKE -283,14: POKE -422,201: REM Modify
170 POKE 260,0: POKE 261,254: REM                  and activate
180 :
190 PRINT CHR$(12): PRINT "From Basicode to Wordprocessor": PRINT
200 PRINT "Read Basicode using recorder 1"
210 INPUT "Start with <RETURN> please! ";V$: A=USR(A)
220 :
230 POKE -282,4: POKE -283,0: POKE -422,17: REM   Correction
240 :
250 A=PEEK(I): IF A=13 THEN PRINT: I=I+1: GOTO 250: REM Show now
260 IF A<>3 THEN PRINT CHR$(A);: I=I+1: GOTO 250: REM and find EOF
270 :
280 PRINT: PRINT: PRINT "Write wordprocessor file"
290 INPUT "Start with recorder number and <RETURN> please ";V$
300 IF V$="1" THEN R$=" 1"
310 :
320 J=I AND 15: I=INT(I/16):                        REM from decimal
330 C$=CHR$(J+48-7*(J>9))+C$: IF I>0 THEN 310: REM to hexadecimal
340 C$="SA WPFBC 080E "+C$+R$+CHR$(13):             REM monitor command
350 FOR I=1 TO LEN(C$):                             REM put into buffer
360 POKE M+I,ASC(MID$(C$,I,1)): NEXT I:             REM and
370 POKE 260,56: POKE 261,230: A=USR(A): END:       REM execute.
380 :
390 REM T.E. Huisman  Ieplaan 48   2565 LN Den Haag.
```

****

INPUT EXTRA.

The INPUT in our previous number about the problems mr. Timmerman had with his JRT Pascal did trigger you writers. From all sides has been responded with suggestions and agreements as to the experienced problems. Now a selection of the received mail is given, together with a possible solution from the original sender. He is the last to conclude.

* Wim van Egmond from Enschede writes: "I possess this Pascal for some time now, but it works well, when using the CP/M program ED for writing programs. The diskversion of the Wordprocessor creates probles as the compiling is cut off for several reasons." Wim also says he works with a 56K version and CP/M 2.2. According to the JRT Pascal handbook 56K is the minimum required."

* Mr. Hiddink from Huizen: "I suppose the source file has been written to disk without the extension PAS. The command mentioned in INPUT is correct in this way. JRTPAS2 now is looking for EXP.PAS. This is also mentioned in JRT Pascal User's Guide by CP/M group."

* Dany Rosseel from Westende (Belgium): "I experienced exactly the same troubles! After digging a little I found JRT Pascal not to run with CP/M version 1.42/1 and probably not on CP/M version 1.42/3 too.
JRT Pascal does not find the file to be compiled as the name is already erased at the moment the file name has been read (in CP/M 1.4). The name is put into the diskbuffer (80..FFH) and CP/M 1.4 uses tha area also for directory purposes. I got round it by swapping two CALLs in JRTPAS2 .COM. The situation now is:
02B9 CALL 3F70
02BC CALL 412A
This problem now had been solved. All did not go well yet. When compiling I sometimes discovered garbage on screen and CTRL-N and CTRL-A do not work. I have not found a solution to that yet. Compiling however is going without errors regardless of the garbage on screen. I also discovered that files having been compiled errorless, do not always run so well! This may be caused by some 'peculiarities' of JRT Pascal. Do more readers have this experience?"

* Mr. M. Timmerman hemself: "The problem of the 'source file not found' has been solved after thorough searching. Two errors happened to be in the programs (I guess they were inserted on purpose!). The first error has been found in JRTPAS2.COM and the second in PASCAL1.INT. Both programs use a FCB (File Control Block) to open Pascal library files: PASCAL0.INT to PASCAL4.INT. To do so the filename is copied into this FCB, whereupon the file is opened, however....
The FCB is initiated using a filetype .PAS, thus starting a search for files PASCAL0.PAS to PASCAL4.PAS and this is it..."Source file not found"! I tried to solve this by a simple RENAME, but found out that the Pascal program needed the INT-files in other places. Using the debugger SID or DDT one can change addresses. In JRTPAS2.COM this address is 4E2D, where INT is to be instead of PAS, as well as at address 101D in PASCAL1.INT."

We thank all contributors for their response and the trouble taken to do so!

****

SOMETHING ABOUT ESGG-DISKS.

The next article from Hermine Bakker is somewhat contemplative and concerns our ESGG volumes for the CP/Mers among the Exidy users.

Yes, why is ESGG issueing volumes herself; is she not bugging the CP/M users group doing so? Regarding the last remark: no, and that is just because of the first. On those ESGG-volumes are only specific EXIDY-programs or EXIDY-implemented ones which cannot be used on other computers without further changes. In such a case it would be senseless to hand them over to the CP/M-user group. We consider this a task for the ESGG, not only for those of the board, that shows a good example by issueing in no time a volume 3 for reasons of importance of the offered software, but for us all, since we all together are the ESGG!
I cannot imagine that a users group with 1K of members and a large amount of know-how, would stop at only 3 volumes. There is already something for the next volume, but .... before we can release until sufficient supply is received. Time passes rapidly for a computerer and before one realizes the next Sorcerer Day is there. Would be nice then to have that fourth volume ready for issueing. As you know we can store some 300K per volume. My question to you now is: Where do I get that. The circumstances made I now function a sort of disk jockey (a sloppy but interesting job). I consider it cheap to grab some cassette based programs from the previously released

cassettes and put that on the disk to fill a volume (this does not mean programs very much disk like are never copied on disk volumes; that is the far future).

I do not beat around the bush: Do look into your program library whatever turns up for sharing whit us for future disk volumes. Preferably with some sort of manual or explanation. That is where most programs fail: It takes time to work out such a thing. Allow me to share a secret with you? I had to provide almost all of the DOCs for volume 2 myself! So if 'you have a program for us, but no time to provide a USER.DOC, just make some notes on a piece of paper, also adding your phone number, whereupon yours truly is going to create that DOC file. But do give the time to have it ready in time. Last minute work is not the best kind, you know! When you use 40 track softsectored, your programs canbe sent directly to me. Different formats are best sent to the secretary who takes care of the conversion. You also can put the programs on cassette using SID or DDT (you do know that GE003 takes you from SID or DDT to monitor and then you just perform a simple SA YRPRG 100 ('NEXT' address)?) to cross the format gap or, when you do not think your country's MAIL is relialble transport for disks.

Dear folks, I wish my mail-box (Falklanddreef 18, 3563 AC Utrecht, The Netherlands) will be flooded in the next days. Please do not postpone your contribution too long. I, together with your fellow-members, thank you in advance for it. Let's make something nice out of this, shall we?

Hermine Bakker.

****

CASSETTE SPEEDSEARCH PROGRAM (2).

In ESGG number 11 there was an article about the speedsearch program by Rob de Beer. This program is also on ESGG software cassette number 13. The program also had been placed first on cassette to show the operation for try-outs. If you had done so, you should have noticed that it did not work as described in the article. Rob's program did not cause this, but while copying the tape a couple of programs 'missed the boat' causing the timing to be incorrect. Hereafter Rob will tell you how to fix it.

In the DATA-list of SNSPL there are at line 204 the names INFO and COLOR, but these programs are not on the tape. Instead the program MAZE is recorded twice. You can try out the speedsearch anyway, but do not go past the program PREM. You have to adjust the parameter Z to the program ESGG and the parameter C to program PREM. This is the fastest way to get an idea about the way of operation.

To speedsearch to the programs following PREM, the DATA-list has to be renewed, this can be done with TES51. With this you are generating a new program SNSPL. I already have done so and I obtained the DATA-list shown hereafter. The run-time from the mechanical stop at the tape-start till the program ESGG was 82 seconds. I think that this time will be about the same for other recorders. If that is correct, you can use this DATA-list as it is.

Also change at line 10 N=37 in 36. I adjusted the parameters to Z=-55 and C=1.017, but these values may probably differ for your recorder.

```
200 DATA 255,ESGG,346,index,679,WPFF,867,DATAC,963,P,993
202 DATA TES51,1336,SNSPT,1519,SNSPL,1607,BIKOL,2312,TES55,2375
204 DATA TES56,2427,TES57,2504,PREM,2710,DEMO,2752,DPLOT,2791
206 DATA MEMOR,2986,TGAME,3078,FIDRV,3265,BOOL,3392,BRIDG,3792
208 DATA CHASE,3889,CONC,4086,DSASM,4294,FXEX,4375,GFISH,4599
```

```
210 DATA INVER,4664,KING,4877,LAIR,5065,LASER,5246,LINE,5292
212 DATA MAZE,5442,MAZE,5589,MORSE,5660,PLFUN,5704,QUEST,5901
214 DATA REVER,6234
```

****

CREATING DATABASES (4).

With this part of the story by Frans Cieremans we almost come to the end of this series. This part shows an example of a sorting routine.

Example of the structure in BASIC of a file aiming at the completion of a key for a SUBMIT operation. The following is a key for sorting with the program SUPERSORT. (Remark: if the sort to do is only a single one, QUICK-SORT as described above, is equally fast, when not compiled.)

```
6800 FNCURS$(A,B)"CREATE SORT-KEY"
6820 CLOSE #4:OPEN "O",#4,"DATABASE.SUB":T$=STR$(D%(2)):
     T$=RIGHT$(T$,LEN(T$)-1):REM put number of rec. in STRING$
6840 PRINT #4,"SORT I=125;S=B:ADDRESS.FIN(1,"+T$+");O=A:
     SORT.FIN,KR;K=1,4;G":REM make key-file of 5 letters
6850 PRINT #4,"SORT I=125;S=B:ADDRESS.FIN(1,"+T$+");O=A:
     SORTKEY,KR;K=5,9;G":REM make key-file of next 4
6860 PRINT #4,"SORT I=125;S=B:ADDRESS.FIN(1,"+T$+");O=A:
     SORT.POS,KR;K=85,90;G:REM post(=ZIP, or MAIL-)code is at pos.85 / 90
6870 PRINT#4,"ERA A:*.SOR":PRINT#4,"ERA B:*.SOR":REM era dd.old
6890 PRINT #4,"SAVE 0 A:"+LEFT$(Z$,6)+RIGHT$(Z$,2)+".SOR"
     Put date of sort in index of A:-and B: disk.
     Z$=date, day/month/year (02/03/1983).
6900 PRINT #4,"SAVE 0 B:"+LEFT$(Z$,6)+RIGHT$(Z$,2)+".SOR"
6910 PRINT #4,MBASIC /F:9 /S:300:REM start MBASIC after sort
6920 PRINT TAB(28)"Type in SORT-DATABASE":REM users-instr.
6930 CLOSE:RESET:SYSTEM:  REM return to CP/M-level.
```

The name SUBMIT.COM is changed into SORT.COM, so the user has to type-in a meaningfull text. The key-file made by SUPERSORT is referred to below.

```
 1e field:SLEUTEL          (SLEUTEL =Dutch for KEY =English)
 2e field:DUMMY
 3e field:CVI(  ), binary record-nr. where the information is.
```

```
420 OPEN "R",#3,"A:RAPPORT.FIN",76          (RAPPORT=Engl.REPORT)
440 OPEN "R",#5,"A:SORT.FIN",7
450 OPEN "R",#6,"A:SORT.KEY",8
460 OPEN "R",#7,"A:SORT.POS",10
520 FIELD #3,26 AS I0$,25 AS I1$,25 AS I2$
530 FIELD #5,4 AS E0$,1 AS E1$,2 AS E2$
550 FIELD #6,5 AS F0$,1 AS F1$,2 AS F2$
560 FIELD #7,7 AS G0$,1 AS G1$,2 AS G2$
```

(to be continued)

****

CHIPTIPS.

It does not stop, those hints that come in!! This part is a continuation to the part published in the previous number by mr. De Witte. If you like to get more information of this kind, be patient, there is some more in stock!

1. Certain programs are specifically for machines with a small RAM area; If you use POKE and PEEK-addresses this may cause trouble when the program is used on an Exidy with more 'K'. It is possible to move both stack and MWA to an address where the program in the smaller machine was meant to run. Sometimes the entire memory (48K) is needed for a program; the Stack and MWA are then to be moved thus:

```
        EN 0 CR
-----0000:21 FF 7F C3 06 E0/ CR (7FFF=32K)-----
        GO 0 CR
```

   At other addresses then '7FFF' is also possible, of course.

2. When testing a program it is always possible that by mistake something goes wrong. The Exidy then TILTS and only a double reset will help out; to prevent this as far as possible, you can ENter the following:

```
        EN 0038 CR
-----0038:C3 03 E0/ CR-----
```

   Fill the whole memory with 'FF' and load/write the program to be tested; if something goes wrong now, the program will find 'somewhere' a 'FF', and will see this as a RESTART-0038. That is the address of the 'warm'-start of the monitor. If you like to see the address from where had been jumped erroneously, you have to ENter something at address 0:

```
        EN 0038 CR
-----0038:CD 0F E2 CD DD E0 C3 03 E0/ CR--------
```

3. Those of you that sometimes use RS-232, know there is a bug in the monitor 1.0. That's the reason the RS-232 is disabled by the KEYBOARD-routine. To turn it on, type in the following:

```
        EN BFCE CR
        BFCE:C0/ CR (is normally '40')
```

   Thereupon you are able to 'LO' and 'SA' via RS-232.

4. When NOT using some kind of TOOL-KIT together with BASIC, it happens now and then that writing of programs fouls up and then you have to RESET. Your program then is lost. In fact, the program is not lost at all, only a some pointers have been set again and the start of the program is mutilated. To be able to continue upon a RESET, you have to do next. Let us suppose the program runs from 1D5 till FFF:

```
        BYE CR
-----M0 100 FFF 5100-----
```

   NOW IF SOMETHING GOES WRONG...*then*...RESET.....*and*

```
        M0 5100 5FFF 100 CR
        PP CR
```

   and the program works again without reloading. There is also another way too; once the program has been M0ved, you can return to Basic with 'PP'. Now you can use POKE's to look into memory. Together with a TOOL-KIT this gives many possibilities, and goes as follows:

```
-----POKE 330,81 CR-----
```

   The (decimal) value '81' represents the HEX-value '51' of the address 5100. You also could load at different addresses different Basic programs and call with POKE in the *direct mode*. This statement could also be inserted in the program, allowing several programs to work together; be aware that a specific program does not overwrite another programs by its string space. You can solve this too, but we handle this later on. To return to the normal BASIC address 01D5, another POKE, but now '01'; this also is achieved with a Basic line.

   E.g. first move the program and look in it with POKE:

```
IX=Beginn-address origin   (0100H)
IY=  "       "  target-area (5100H)
X =Number of bytes of prog (see 1B7-1B8)
-----FOR I=0 TO X:POKE IY+I,PEEK(IX+I):NEXT I-----
```

5. When a program has been loaded from tape and after having used it, you like to see the program's FILE HEADER again then:
   Eventually first to the MONITOR (BYE CR)
   -----GO E6DE CR-----

6. When the programhas been moved, the string-stack-area has to be fixed to prevent programs from mutilating each other while running. This can be arranged using the next statement:
   -----CLEAR 50,2000-----
   The value following the comma is the end address of the stack; this address is to be determined by looking into memory using DUmp to find out how far beyond the program RAM has been used. A couple of bytes further the address has been converted from HEX into DEC and that number now is the value following the comma.

****

The CP/M DRAMA (2).

As has been said by Antoine van de Ven in the first part of this article, the continuation of his story has to be written by you readers. Even if we (the editor's team) would like to, we would not have the necessary time and manpower to continue an investigation like this one. Therefore it has to come really from You!
There is nothing left for me but to establish that more and more users of hardsectored systems provide themselves with a controller for softsectored and get rid of the S100 unit! To those users suddenly a world of programs that run without problems, is opening up!

3 The BIOS-entry.

The linking of CP/M and the hardware (excluding the disk units) is done in the (C)BIOS. As all of the computers working with CP/M are of a different design, we find in the BIOS most of the differences responsible for malfunctioning programs. We can find the start of the BIOS by looking into the addresses 0001 and 0002. Here is the address of the jump at a warm CP/M-start: We find B303. This means that the BIOS starts at address B300; we there find the jump-table. This table is up for discussion in the next paragraph; first you find some data of three CP/M-versions.

	CP/M 2.2 SS CBIOS 1.92 CD 1-3-82 48K	CP/M 1.4 HS 1.42/3 Exidy 1979 47K	CP/M 2.2 HS CBIOS 1.23 CD 25-6-82 48K
BDOS-entry:	A506	A406	A506
BIOS-entry:	B300	B100	B300
Nesting-area:	9D00 - BDA2	9C00 - BAFF	9D00 - BB1A
TPA-overwriting:	no	100 - 4FF (at cold-start)	100 - 4FF
Cassetteproblems:	no	no	yes, monitor cold start is necessary.

4 The BIOS jumptable.

Address *)	2.2S	1.4H	2.2H		
B300:	B303	????	B303	BOOT	cold start.
B302:	0000	????	????		P.S.: cold start=warm start
B303:	B3ED	B7F2	B3A2	WBOOT	warm start.
B306:	B498	B8AA	B405	CONST	console ready?
B309:	B489	B8E0	B3F6	CONIN	read from console.
B30C:	B492	B8E8	B3FF	CONOUT	write to console.
B30F:	B48F	B911	B3FC	LIST	write to LIST-device.
B312:	B48C	B916	B3F9	PUNCH	write to PUNCH-device.
B315:	B486	E00F	B3F3	READER	read from READER-device.
B318:	B637	B5B6	B4EE	HOME	move head to track 0.
B31B:	B64D	B57A	B4DA	SELDSK	select drive.
B31E:	B639	B1A6	B4F0	SETTRK	set tracknumber.
B321:	B63E	B1BA	B4F5	SETSEC	set sectornumber.
B324:	B643	B1C2	B4FA	SETDMA	set DMA-address.
B327:	B823	B1C7	B4FF	READ	read selected sector.
B32A:	B822	B23A	B510	WRITE	write selected sector.
B32D:	B47F	????	B3EC	LISTST	get liststatus.
B330:	B610	????	B4D4	SECTRAN	sector translate-routine.

**) BIOS-address for CP/M 1.4 (subtract 200H).*

In columns 2, 3 and 4 are the addresses to where is being jumped. What is next, is about CP/M version 2.2 SS. Disassembling leads us into a puzzle. I now will place some pieces on the table but then there is the moment you come in (remember our deal, part 2 is written by you!). We now take B315: jump to B486. The disassembling gives us the following:

5 From READER: read from READER-device (I).

```
B486:  061F    LD B,1F
B488:  3A0621  LD A,(2106)
B48B:  3A0615  LD A,(1506)
B48E:  3A060B  LD A,(0B06)
B491:  3A0601  LD A,(0106)
B494:  3A063F  LD A,(3F06)
B497:  3A0631  LD A,(3106)
B49A:  ......  ..........
```

At first look this seems nonsense. Six successive loads of the ACCU! Obviously they are dummy instructions. Let us analyse first another jump:

6 From CONIN: read from CONSOLE (I).

```
B489:  0621    LD B,21
B48B:  3A0615  LD A,(1506)
B48E:  3A060B  LD A,(0B06)
B491:  3A0601  LD A,(0106)
B494:  3A063F  LD A,(3F06)
B497:  3A0631  LD A,(3106)
B49A:  ......  ..........
```

Now compare these addresses with the ones from the previous paragraph. The LD A,(..)-instructions are obviously meant to omit a number of addresses. The only thing that counts is the loading of B.

7 The magic box.

```
B494:  21B6B4  LD HL,B4B6  Start address of the address-table.
B49D:  78      LD A,B      Load function number.
B49E:  E638    AND 38      Erase bit 0,1,2,6 and 7.
```

```
B4A0:  5F       LD E,A       Save.
B4A1:  A8       XOR B        Only the unequal bits become 1.
B4A2:  47       LD B,A       Save.
B4A3:  3A0300   LA A,(3)     Get the I/O byte.
B4A6:  07       RLCA         Rotate left, copy the last bit
                             into the carry-flag
B4A7:  10FD     DJNZ B4A6    Repeat till register B is empty.
B4A9:  E606     AND 6        Erase the bits 0 and 3 till 7.
B4AB:  50       LD D,B       Empty D.
B4AC:  19       ADD HL,DE    Add displacement to the start address.
B4AD:  EB       EX DE,HL     Address in DE, displacement in HL.
B4AE:  6F       LD L,A       Second displacement in L.
B4AF:  19       ADD HL,DE    We found the spot in the address-table!
                             There is the jump destination address.
B4B0:  7E       LD A,(HL)    Get the low byte of the address.
B4B1:  23       INC HL       Next address.
B4B2:  66       LD H,(HL)    Get high byte of the address.
B4B3:  6F       LD L,A       Low byte in L; HL has now a valid address
B4B4:  79       LD A,C       ?? now comes in A. +)
B4B5:  E9       JP(HL)       Ok.! We jump!
```

+) *This happens only if it is output and if the contents of reg. C is not known! (Ed.)*

Although this magic box seems to be a hopeless affair, one can figure it out. We need: a function number, the I/O-byte, a sheet of paper and a pencil. We take for example funktion-number 1F, 'read from READER'. Our I/O-byte turns out to be 0.
We get the following:

Addrs:	ACCU:	B:	D:	E:	H:	L:
B49A		1F			B4B6	
B49D	1F	1F			B4B6	
B49E	18	1F			B4B6	
B4A0	18	1F		18	B4B6	
B4A1	07	1F		18	B4B4	
B4A2	07	07		18	B4B6	
B4A3	00	07		18	B4B6	
B4A6/7	00	07/00		18	B4B6	
B4A9	00	00		18	B4B6	
B4AB	00	00	00	18	B4B6	
B4AC	00	00	0018		B4CE	
B4AD	00	00	B4CE		0018	
B4AE	00	00	B4CE		00	00
B4AF	00	00	B4CE		B4CE	
B4B0	6C*)	00	B4CE		B4CE	
B4B1	6C	00	B4CE		B4CF	
B4B2	6C	00	B4CE		B5*)	D5
B4B3	6C	00	B4CE		B5	6C
B4B4	??	00	B4CE		B56C	
B4B5	j u m p	t o	:		B56C	

*) from address-table.

(to be continued by YOU?).

****

EXIDY 30 TRACKS MPI DISK CONTROLLER (3).

In the introduction to the second part of the 'wanderings' of Henk Warnitz among the troubles with his disk-units we did refer to previous parts regarding his telephone number. Unfortunately that number never had been

given, therefore we now publish it: 03429-3181.

At this point you ought to get the story on the troubles with the BASF-drives, as had been listed in the first part of this article. Henk, however, has made some changes meanwhile and also requested to insert the modifications of the controllerboard first as this is a more logic step in the story. Of course we are glad to fullfill such a request.

Before dishing up these modifications, we first publish the marginal notes that have been put by Henk in the first two parts of his series.

Regarding part 1: the S100 cable.

It is better to terminate the datalines with 220 ohm, regarding the load of the buffer IC's.

Regarding part 2: the data-separator and no write precompensation.

Contrary to what I have written before, 125 ns for BASF 6106 drives works just fine with me too. I only did forget to connect the unused inputs of IC 74LS166 to ground and TTL looks upon unused inputs as being logically a 1; At least almost all times, but non connected inputs are not stable.

used	shift	connect to ground
C/E	(125/250ns)	A/B/F/G
B/F	(250/500ns)	A/C/E/G
A/G	(375/750ns)	B/C/E/F

If the write-precompensation is used on all the 40 tracks then the shift has to be as narrow as possible.

4. the modification of the controller board.

Check all the directions in the diagram and follow the print tracks. With me everything worked at once. However do not count on it that it will be the same with you, when following my directions gratuitously. Cutting a print track at the wrong place is easily done!

Making the changes at the controller board.

Cut the following tracks in two places, close to another, using a sharp knife. Heaten this piece of track with a soldering-iron. Now you find it will peel-off without much difficulty.

Remove IC 5C.

TO BE CUT:	**SOLDER A WIRE:**
1C-37 / R10	1C-37 / GND
5C-3 / 5C-4	3C-12 / 5C-1
5C-13 / 5C-12	6C-14 / 5C-3
5C-12 / 5C-10	R10 / 5C-5
5C-13 / 1C-37	5C-6 / 5C-4 / GND
5C-1 / 5C-11	5C-7 / 1C-27
5D-2 / 5C-2	5C-2 / 6A-10
	6B-6 / 2D-13
1C-33 / 2D-13	2D-12 / 6A-9
7B-5 / 1C-26	6A-8 / 1C-26
1C-30 / 6B-4	1C-30 / 1D-9 / 1D-10
	1D-8 / 6B-4
	5D-2 / 6C-2

Exchange the 10K resistor R10 with one from 2K2.
Install an IC-socket after having cut the tracks. This will facilitate the (future) exchange of the IC.

Remove IC 7c:

Cut at *components*-side:

7C-4 / 5C-6
7C-5 / 7C-10
7C-9: cut the groundtrack at both sides of this IC-pen. Connect both ends again using a THICK wire **around** 7C-9.

Cut at *bottom*

7C-15 / 7C-16
7C-5 / 7B-4
7C-7 / 7B-3
7C-10 / 7C-15
7C-11 / 1C-27 / 5B-4
7C-14 / 6B-6
1C-31 / 2A-12

Install an IC-socket.

Solder a wire (this is best done at the bottom):

1C-31 / 2D-11
2D-10 / 7C-15
1C-18 / 4C-2 125 ns shift @ 8Mhz.
4C-2 / 7C-4 ,, ,,
4C-1 / 7C-5
1C-17 / 4C-3 EARLY 125 ns @ 8Mhz.
4C-13 / 7C-10 ,, ,,
7C-14 / 6C-8
7C-9 / 1A-1
7C-7 / 5D-2 8Mhz
7C-1 / 7C-6
7C-1 / 7C-8
7C-13 / 2A-12

(to be continued)

****

THE ESGG AND THE CLOCK.

It is quite some now since we heard from Wim de Kreuk, to wit by the co-production together with Floot Vogelaar and others in BEXT.
At the Sorcerer Day he purchased a clock and wrote a program to it.

With many other members I purchased the realtime clock at the latest Sorcerer Day. After some soldering, where I found the yellow wire (DB1) to be a blue and white one, one likes to know if it works. The program KLOK on cassette number 15 turned out not to be the one meant for this clock and typing all of the demo programs was not to my liking. Therefore I dug into the programs and I happen to find a way to a reasonably short program. It is true that this program is not provided with all of the safeties of the demo program but on the other hand this program is capable of handling data changes during reading. I think that this could go wrong with the slow Basic program. When the clock functions well, one only need the TIMESET

program once. The date/time group is being typed-in from ADRES 0016. Just GO 0 and RETURN at the time set and your clock keeps good time. GO 20 returns the time at any request. Time is displayed in the right upper corner of the screen in tenth sharp.

First the HEX dump of the program:

```
ADDR     0  1  2  3     4  5  6  7     8  9  A  B     C  D  E  F

0000:    21 16 00 01    4E 0B AF D3    40 ED 79 7E    0D 23 10 F9
0010:    3E 01 D3 4E    18 0A 08 00    04 06 00 06    02 01 03 00
0020:    21 FD F0 01    41 0C ED 78    E6 0F FE 0F    28 F2 C6 30
0030:    0C 2B 77 10    F1 C9
```

The second program has more options in it. This programs places the date/time group in the variable DT$ and the time -hour, minute, second- in TD$. These variables are created by the subroutine itself. In a same manner any substring may be derived from the date/time group. The Basic example program shows in what ways the routine can be used.

```
1 POKE 260,32: POKE 261,0: T=USR(0)
2 PRINT DT$,TD$
```

```
0000   21 16 00   SETTIME   LD      HL,DATA    ;Address date + time
0003   01 4E 0B              LD      BC,0B4E    ;C=gate; B=counter
0006   AF                   XOR     A
0007   D3 40                OUT     (040H),A   ;TEST MODE off
0009   ED 79       LOOP1    OUT     (C),A      ;SET clockregisters
000B   7E                   LD      A,(HL)
000C   0D                   DEC     C
000D   23                   INC     HL
000E   10 F9                DJNZ    LOOP1-$
0010   3E 01                LD      A,001H     ;Start clock
0012   D3 4E                OUT     (04EH),A   ;
0014   18 0A                JR      READTIME-$;Jump over data

0016   08           DATA    DEFB    00008H     ;leap-year? 8 4 2 of 1
0017   00 04                DEFW    00400H     ;April = 4
0019   06                   DEFB    00006H     ;Saturday = 6
001A   00 06                DEFW    00600H     ;day
001C   02 01                DEFW    00102H     ;hour
001E   03 00                DEFW    00003H     ;minute

0020   21 FD F0   READTIME  LD      HL,SCREEN
0023   01 41 0C             LD      BC,0C41H   ;C=gate, B=counter
0026   ED 78       LOOP2    IN      A,(C)
0028   E6 0F                AND     00FH
002A   FE 0F                CP      00FH
002C   28 F2                JR      Z,READTIME-$;is there a change?
002E   C6 30                ADD     A,030H     ;ASCII
0030   0C                   INC     C
0031   2B                   DEC     HL
0032   77                   LD      (HL),A
0033   10 F1                DJNZ    L0026-$

0035   C9                   RET                ;end of base program.
                                                ;see HEX dump
```

```
                        SCREEN    EQU     OFOFDH    ;display (Right up)
                        DATABL    EQU     OF054H    ;shielded RAM for
                                                    ;our data
                        VARPTR    EQU     OCDAOH    ;find the address of
                                                    ;the variable pointer
                                                    ;and put that into DE
                        MVPTR     EQU     OD652H    ;move 4 bytes (str. length
                                                    ;address) from DE to HL
0035    11 54 FO                  LD      DE,DATABL ;to shielded address
0038    OE OC                     LD      C,OOCH
003A    ED BO                     LDIR
003C    11 51 OO                  LD      DE,VARTBL
003F    01 D4 44                  LD      BC,"DT$"  ;string var: C OR 80H
0042    CD 48 OO                  CALL    PTRSET
0045    01 C4 54                  LD      BC,"TD$
0048    D5              PTRSET    PUSH    DE
0049    CD AO CD                  CALL    VARPTR    ;put VARPTR in DE
004C    EB                        EX      DE,HL     ;now in HL
004D    D1                        POP     DE
004E    C3 52 D6                  JP      MVPTR     ;MOVE 4 Bytes and RET
0051    05 OO           VARTBL    DEFW    00005H
0053    54 FO                     DEFW    OF054H
0055    06 OO                     DEFW    00006H
0057    59 FO                     DEFW    OF059H
```

This last part of the program (from 35H up) is of course only to be used together with ROMBasic. It shows also in what way one can take data to Basic in a USR routine. The number of variables is here basically unlimited!

So far for Wim de Kreuk. The next part again point towards a genuine co-production: The complementary to the afore mentioned is from the hand of Floor Vogelaar, who elucidates with his explanations the matter for the beginners.

The first program (0000-0035, see HEX dump) is fully relocatable. You may store this anywhere in your (computer's) memory. In the second program only the values for PTRSET (0048) and VARTBL (0051) need to be modified.

Storing String variables in (Standard) Basic.

For a better understanding of the part of the ESGG clock program that takes care of the moving of the clock data to the two strings (DT$ and TD$) the following is of importance:
Basic keeps a list of string variables up to date, which can be found in the 'BASIC STRING SPACE'. For this purpose 50 bytes are automatically reserved upon starting the Basic pack. By use of the statement CLEAR NNNN this space may be increased or reduced (CLEAR 10 ⇒ 10 bytes; CLEAR 2000 ⇒ 2000 bytes). To be able to find the variables stored in this space, there also is kept a list of the names of these variables. The first address of this list can be found at 01B7 and 01B8 and the last address in 01B9 and 01BA (in reverse order of succession). Ordinary (floating point) and string variables are placed randomly. The string variables can be identified by the highest bit of the variable name being set. A 'D' that is usually the ASCII value 44 hex (0100 0100 bin) now is C4 hex (1100 0100 bin). Basic tests this; one needs to realize that it is important to be aware of the significance of only the first two characters of the name. A variable INPUT$ is really named as IN$. Only the *first two* characters are to be inserted in the list, in a **reverse order of sequence (first the N and then the I)**. The ordinary (floating point) variables are not taken into consideration. The structure of the string variable pointer in the table is as **follows:**

+0	variable name
+2	lenght of the string (maximum 255)
+3	always 00
+4	two bytes containing the address itself (again in reverse order of sequence).
+6	the next variable name; etc..

The total use by each pointer is 6 bytes.

Where now is this string address pointing to? This may be your program, or to the BASIC STRING SPACE. That is, whenever you leave the handling of the strings to Basic. If your program has a line line this: 10 LET AA$="TEST", then the string pointer will lead to the position that holds "TEST" in your program. An advantage of this is that no space will be taken from the STRING SPACE. When defining a string by an 'INPUT' or by a 'READ DATA' statement, then the string is to be placed in the STRING SPACE.

Now by changing the address of the variable pointer, one can assign a string of a maximum lenght of 255 bytes, being in memory anywhere, to a string variable.
Now this is exactly what happens in the clock program of ESGG by Wim de Kreuk. Back to that program: At address 0035 date and time, fetched from the clock, are being placed at memory addresses F054 hex and up. The routine next to the PTRSET label first looks for the address of the relevant variable pointer and next places the 0 byte, the string length and the string address from the table VARTBL at the right spot. From this table we learn that the first variable is 5 bytes in length and starts at address F054 and the second variable is 6 bytes in length and starts at address F059.

****

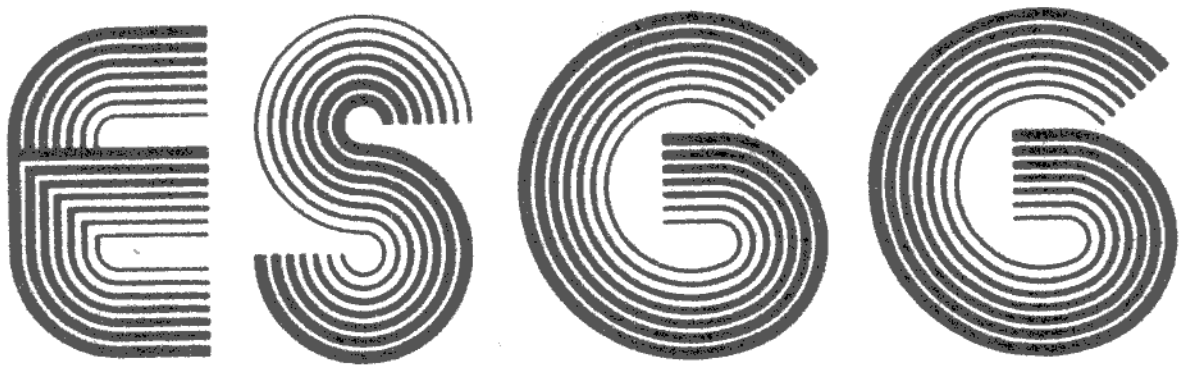
Dear Foreign Readers,

The ESGG always trys to make the English version a genuine copy of the periodical issued in the Dutch language. This takes up quite a lot of our time. Even then we realize that we never can have such a good translation. This is mainly because languages do have their own peculiarities and ways. Also certain proverbs or words are not always translateable. For this we ask your understanding.
As the magazine ESC from the European Sorcerer Club ceased to exist June 1984, we consider our duty is to be the 'voice' of a growing number of Sorcerer owners throughout the world. However, we only can do this with the help of many, especially from other countries.

Since we started with the periodical in English, the major part of the articles and programs is coming from Dutch sources. Is this because you do not have the possibilities and knowledge? We suppose this is not the reason. There is quite a lot of knowledge in various fields of the Sorcerer hidden abroad. Therefore I like to invite you all to join the contributing members of the ESGG and send your programs, articles, reviews and so on, to me.

We all have to realize that when nobody is responding, the ESGG soon will fade away for lack of interest. As long as there are subscribers and members willing to contribute the existence is secured. Especially the users in other countries are often shut off of information about their system and its possibilities. Sorcerer users therefore should stick together and help things keep on going in their best interest.
May I count on you, to help us grow a strong users group?

Welmoed Jonker.



The L O G I C partner to a Sorcerer

For whom is the ESGG?
For anyone being interested in the use and the possibilities of the Exidy Sorcerer.

Why the ESGG?
Because the ESGG trys to give as much knowledge to the possibilities of the Exidy Sorcerer and especially to the possibilities in the use of the Exidy Sorcerer, outside as well as inside the Hobby Computer Club of the Netherlands.

What does the ESGG?
Software distribution: We only supply software that is free of COPYRIGHT (so-called Public Domain Software) on collect cassette and on diskette.

For Exidy Standard Basic we supply a Basic EXTension in EPROM, that expands the possibilities of the Basic Pack very much.

Hardware development: Non commercial designs, that is designed by members and put at the disposal of the other members, are judged by ESGG and -if of importance to others- produced (e.g. video inverter).

Sorcerer days: Twice a year (usually in March and September) ESGG organizes these meeting places of many a Sorcerer user. These days meanwhile have become well-known.

Publications: Our bi-monthly issued ESGG-periodical, full of things worth knowing about the Sorcerer and alike matters. For only DF1. 27,50 (in Europe) or DF1. 32,50 for non European countries your can insure yourself of the most recent information concerning your Sorcerer! (see page 2).
In the running subscription year we supply all issued numbers!

Subscribe?... Just remit the fee for the subscription either by money order or by postal check to postal check account number 5368539 of ESGG, Herman de Manpark 41, 3411 ZN LOPIK, the Netherlands, referring to "subscription to ESGG periodical".

You like to know more about the ESGG? Apply to the sectretary, mr. Charles Netteler, Pr. Hendrikstraat 3d, 3071 LG Rotterdam.