SPECTRAVIDEO™

AUSTRALASIAN USERS GROUP

News Letter

REGISTERED BY AUSTRALIA POST PUBLICATION NO. TBH 0917 CATEGORY "B"

ISSUE No.
2 - 5

DATE
FEB 1985

ALL CORRESPONDENCE TO:

S.A.U.G.,
P.O. BOX 191,
SOUTH LAUNCESTON,
TASMANIA, 7249.

(003) 312648

ANNUAL SUBSCRIPTION

AUSTRALIA $15.00
OVERSEAS $20.00
OVERSEAS AIRMAIL .. $25.50

SPECTRAVIDEO™

AUSTRALASIAN USERS GROUP

INTRODUCTION

These short months can be a real worry. So if this newsletter is a bit late please forgive.

We are pleased that Mr. Dunning has again given us a great bit of information. It appears that he has been ill and I hope all is now well.

It is always hard to give everyone something every month and so we here at head office try to make the newsletter as varied as possible with each month being a little bit different.

This month is a funny one, normally I sit down and I know what will be in that months issue. But this one was different, I really did'nt know what it would look like. But I started typing and somehow it fell together.

Before I go into what we have this month I would like to introduce you all to the Wellington Spectravideo Users Group. This is the first group (apart from ours) that has a Newsletter that I was impressed with. So for further information about them you should contact Mr. Don Stanley, C/- P.O. Box 7057, Wellington South, New Zealand. PH. 896379.

This months newsletter has the following :

EXPLORING BASIC PT-9 Which is the first part of a Machine code routine discussion and how to use them with Basic.

LIBRARY NOTES again speak for themselves. But I would like to add that the LIBRARY ORDER FORM you find enclosed with this newsletter is designed to make our life easier. We ask that you use this when ordering and we will replace it when we return your order.

SOUND FACTORY This program was a puzzle to me when I first ran it but after playing with it I must admit I actually learned something. The program was designed to show you how the Sound Generator works, I know you will admit that the SOUND command is very much a mystery to us all. So I thought it good enough to make it the PROGRAM of the month. There is no information with it except that found at the end of the program in some REM statements. So it's all up to you to learn by your mistakes. Have fun...

DATA 70 is a useful bit of software if you like the idea of fancy text. It explains itself when you run it.

Well that is all from me for another month but remember we need articles for the newsletter and the quality of our newsletter depends on the quality of your (the users) input. Frankly the response to last months request for Articles was less than wonderful. In fact it was down right lousy.

SPECTRAVIDEO™
AUSTRALASIAN USERS GROUP

EXPLORING BASIC PT-9

By L.A. Dunning

This installment is the first part of a discussion about machine code routines and how to use them.

WHY?

Why use machine code routines at all? Basic is perfectly fine for most jobs you might like to give it, but sooner or later you will want to do things either quicker than basic or impossible using basic. Machine code routines allow easy access to special routines which, if planned carefully will make the program run more efficiently. There are two preconditions needed before you should meddle in machine code: knowledge of the Z80 instruction set and patience.

There are a number of books that deal with the Z80 codes. A full examination of a local technical bookshop should provide several books of this nature, although most of these are written in a machine specific manner (e.g TRS-80 or Spectrum) however there are books that deal only with the Z80, assuming you know enough about your machine to use the infomation. The discussion below deals with making those codes work on the Spectravideo. It is assumed that you know something of Z80 codes, but not necessarily a great deal. You will require patience because such machine code routines (abbreviated to MCR hereafter) have a habit of hatching bugs when run, either due to bad planning or simple errors, requiring the user to sit down and figure out what went wrong.

THE ROUTINE

For introductory purposes all examples initially given will assume that you wish to load and run the MCR shown in diagram 1. This is a listing from an EDITOR/ASSEMBLER. This is a program used to construct assembly instructions and encode those instructions into machine code. The two columns under "Op codes" are the assembler instructions as designed by the user. The hexadecimal numbers listed under code represent the results produced by encoding.

```
Bytes Code   Lines      OP Codes              Comments
0000         000        ORG0000H
0000 3A3AFE  010 START   LD   A,(0FE3AH) ;GET SCREEN STATUS
0003 FE00    020         CP   0          ;IS IT TEXT MODE?
0005 C0      030         RET  NZ         ; IF NOT,RETURN TO BASIC
0006 210000  040         LD   HL,0000
0009 CD3C37  050         CALL  373CH     ;SET UP VRAM ADDR
000C CDA91A  060         CALL  1AA9H     ;PUT USR IN A
000F 57      070         LD    D,A       ;SAVE A
0010 01C003  080         LD    BS,03C0H  ;#BYTES IN SCREEN 0
0013 7A      090 LOOP    LD    A,D       ;RESTORE VALUE
0014 D380    100         OUT   (80),A    ;DUMP VALUE TO SCREEN
0016 0B      110         DEC   BC        ;ONE MORE DONE
0017 78      120         LD    A,B       ;IS BC
0018 B1      130         OR    C         ;ZERO?
0019 20F8    140         JR    NZ,LOOP   ;IF NOT, JUMP TO LOOP
001B C9      150         RET             ;RETURN TO BASIC
0000         160         END
```

```
LOOP  0013
START 0000
```

This routine is only 28 bytes long. It grabs a value between 0-255 as input by the user and then dumps this value to all locations on the TEXT screen, like a CLS statement gone wrong. There is a safety catch which prevents a dump when the screen isn't in TEXT mode. The routine is relocatable - this means it can be placed anywhere (within reason) in RAM and still work. Some routines are designed to fit only in one spot and can't be relocated.

In this case the entry point of the MCR - the point at which the routine starts executing code - is at the start of the MCR, however in many cases the entry point of a routine need not be at the start and can be anywhere in that routine. There is therefore a need to distinguish between the start and entry of an MCR. In all the listings that follow, the start of the MCR is indicated by the variable MS!, the length of the routine by ML% and the entry point by EP! Both start and entry points use Single Precision variables for convenience as it is harder to cause an error this way.

One other point, this routine is designed to be used with the statement J=USR(Q). If you enter the routine directly from a BLOAD you will get an illegal function call. In case you were wondering, the EDITOR / ASSMBLER was one used on my System-80 Computer. As yet I don't know of any assemblers that will run on the SV's other than with its CP/M system.

WHERE, HOW & WHICH

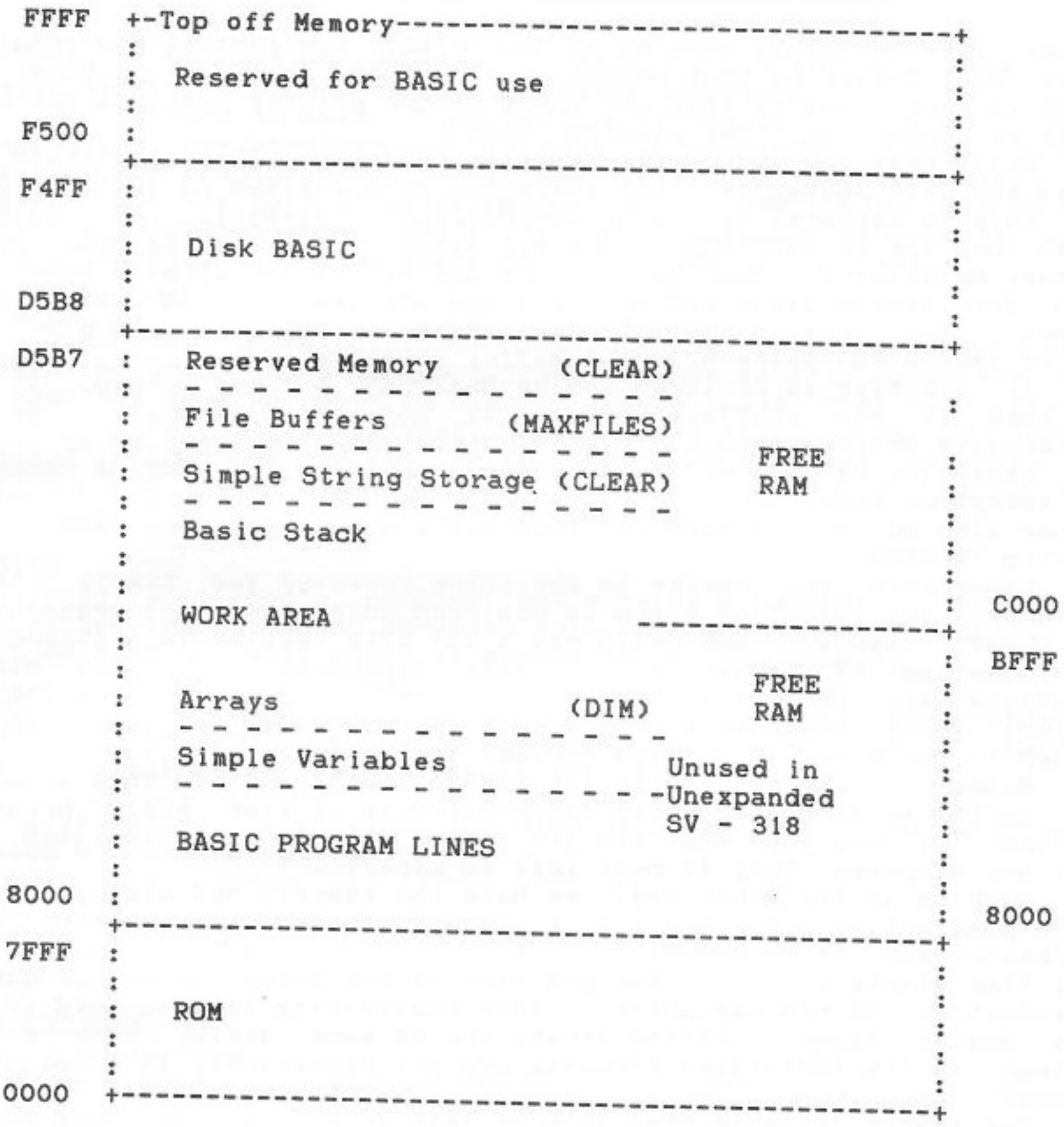
Assuming you have a handy MCR you wish to run, such as listed, there are three things to note about it. MCR's must reside in RAM, must be loaded into that RAM and must be accessed in some manner; thus we have three problems of MCR's-

1) WHERE TO PUT IT
2) HOW TO PUT IT THERE
3) HOW TO ACCESS IT

LOCATIONS

To figure out where to put the machine code routine, we have to know a little about the memory map of the SV and Diagram 2 gives a look at the general layout.

The top portion of the diagram shows an area reserved by basic as a general workspace, vectors and pointers so this is not a usable area to put MCR's. Just under this is the added coding for DISK basic. If you aren't using a disk system, then the area used for DISK basic is free for the areas beneath it. Just as with the area above it, this is not a sensible area at put an MCR because sooner or later the routines you've overwritten will be called and the system will crash.



The area designated as 'Reserved Memory' is the portion of protected memory delinated by the second argument in CLEAR, which is the lowest memory location of that segment. As this area will be just under the vectors or DISK basic, it will be left undisturbed and so makes a good place to put your MCR.

Basic will not let you reserve space above F500H for tape system and it is unwise to reserve above D5B8H for a disk system as this would overwrite the vectors or disk basic. If you know what type of system the MCR is to run on, it is a simple matter to reserve space to a fixed location. This assumes that there is no other MCR already in that space. This might not be the case.

If you have a relocatable routine and you wish to insert it underneath other possible MCR's, it is possible to reserve space that will start from the current reserved location, other than a fixed one. Memory locations FDE6H and FDE7H are pointers that hold the current value for the reserved location. You can determine this in a standard

manner (see below) by peeking at the value, subtracting the required bytes and declaring that amount on the CLEAR statement. Once the CLEAR is done, you've lost the value of the pointer but this can once again be peeked at. See Listing 'CLEAR'.

Below reserved memory are the file buffers. There is always at least one file buffer (#0) and there can be a maximum of 16 (#0 - #15) and this is declared by use of the MAXFILES statement. It should be noted that use of MAXFILES clears all variables, since the space below it must be altered. Each buffer uses 265 bytes; the first 9 bytes are used for status flags and counters and the last 256 is the buffer proper. The location of the first byte in each buffer is given by use of VARPTR(#N) where N is the buffer number.

If a buffer is declared (using MAXFILES) but not opened, it can be used to put routines in and will not be disturbed. Use of consecutive buffers enable routines longer than 256 bytes to be used. The exception here is buffer #0 which is cleared for routine purposes and therefore unsuitable. An open buffer can become corrupted however it can also be used to save and load MCR's of 255 bytes or less. See listing 'BUFFER'.

Underneath the buffer is the space reserved for simple string variables, the amount of which is declared using the first argument in the CLEAR statment. You could put a 255 byte routine in a string but the location of a string will vary depending upon what string functions are performed and if the garbage collector routine is called. This makes this area generally unsuitable for use. It is easier to use a buffer in such a case.

Under the strings starts the basic stack. Avoid this area at all costs as it will destroy the continuity of your basic program. However, if you know what you are doing, it can be manipulated for your own purposes. This is best left to experts.

Working up the other way, we have the numeric and string arrays. It is possible to put a complex routine into a numeric integer array and manipulate it by changing its elements. The difficulty here is that like simple strings, the position of the array can change due to introduction of new variables. This possibility is discussed in the next months issue. String arrays are no more useful than simple strings, as its individual elements are not neccesarily in consecutive memory.

The simple variable area is best left untouched. Routines can be put directly into the basic program. String literals or data statements or other lines can be modified by use of pokes to put the routine directly into the basic program. The disadvantages of such practices are that you have to avoid zero bytes (they can cause all sorts of upsets in the wrong location) and that the program itself can't be saved in ASC mode without losing the routine. One advantage of using this method is that once placed, the routine becomes an integral part of the program and more transportable. This technique is used on other computers where it was impossible to directly declare reserved space using the CLEAR statement. This possibility will be discussed in a later issue.

Finally it is possible to put a routine just below the basic program. Locations F54AH and F54BH point to the start of the basic program. This will normally be either 8001H (for SV328's) or C001H (for SV318's) and it points to the lowest significant byte in the 'next line pointer' of the first line. By altering this correctly, whole basic lines can be swallowed up and the freed area used for

routines. See Listing 'SNEAK' for details. So far as I know, this pointer is never changed until the system is reset.

The drawback is that if the pointer is not set correctly , weird basic lines are produced and the basic program will appear ruined. This can be restored by poking the correct values back into the pointer. You can make whole programs disappear by moving this pointer to the terminating zero bytes of the program. The program is still there of course, it's just that you can't LIST or RUN it!

PROGRAMS

There are three listings included in this part - 'CLEAR', 'BUFFER' & 'SNEAK'. Each demonstrates how to reserve space for MCRs and uses the routine shown in diagram 1 as a basis. As such they are fairly useless by themselves. Next month there will be further listings which load and run the routine. To run them correctly you will need to type in/load this months listings and save them on tape or disk, as they are used with the new listings.

POINTER VALUES

Where BASIC has a location that acts as a pointer the first byte acts as the low value and the next byte acts as the high value in the pair. To find out what value is stored there :the formula is:

Value = 1st byte + (2nd Byte x 256)

which gives a value in the range of 0 - 65535. This might exceed an INTEGER value so use a single precision variable instead. Such a value is still valid for process et cetera.

Next month, I continue the discussion on MCRs, and talk about loading and executing them.

BUFFER

by : L.A. Dunning

This Program may be entered using the 'INPUT' program from Newsletter 2 - 2 (NOV 84). Send $1 to S.A.U.G. for printout

```
HL      10 REM  BUFFER
EI      15 REM Demonstrates how to set up a              buffer for an MCR
BP      20 REM MS!= Start  of MCR                        ML%= Length of MCR
                           EP!= Entry  of MCR
AH      25 REM FOUR file buffers are assumed             to be used
HB      30 MAXFILES=3:ML%=28
EM      35 MS!=VARPTR(#2)+9:FIELD#2,ML%ASMC$,255-ML%ASDU$:EP!=MS!
EL      40 PRINT"The MCR starts at "HEX$(MS!)
CM      45 ' Those without disks REM next line
CK      50 OPEN"1:MCR.lib"AS#2
END
```

SPECTRAVIDEO™
AUSTRALASIAN USERS GROUP

SNEAK

by : L.A. Dunning

This Program may be entered using the 'INPUT' program from Newsletter 2 - 2 (NOV 84), send $1 to S.A.U.G. for printout.

```
JI      10 REM Line is M1%-4 bytes to:
FG      15 REM Type line 10 exactly as shown
AI      20 REM SNEAK
DH      25 REM Demonstrates how to reserve                space just below a BA
           SIC Program
CA      30 REM MS!= Start  of MCR                         ML%= Length of MCR
                           EP!= Entry  of MCR
EC      35 REM DO NOT RUN THIS PROGRAM TWICE              WITHOUT REBOOTING THE
            SV
EH      40 PRINT"'BEFORE'":GOSUB70
DE      45 PRINT:PRINT"How many lines to gobble <1-5>":INPUTN:IFN<1ORN>5GOT
           045
AH      50 MK=&HF54A:GOSUB60:MS!=M3:FORR=1TON:MK=M3:GOSUB60:MK=&HF54A:GOSUB
           65
EI      55 NEXT:GOTO75
IF      60 M1=PEEK(MK):M2=PEEK(MK+1):M3=M1+M2*256:RETURN
DG      65 POKEMK,M1:POKEMK+1,M2:RETURN
BN      70 PRINT"BASIC Program starts at ";:MK=&HF54A:GOSUB60:PRINTHEX$(M3)
           :BS=M3:PRINTSPC(14)"ends   at ";:MK=&HF7EE:GOSUB60:PRINTHEX$(M3)
           :RETURN
DC      75 PRINT"'AFTER'":GOSUB70:PRINT"The MCR starts at       ";HEX$(MS!)
           :ML%=BS-MS!-2:PRINT"          can use "ML%" bytes":EP!=MS!
EM      80 PRINT:LIST-35
FC      85 REM Delete line 80 when used with              another program
END
```

CLEAR

by : L.A. Dunning

This Program may be entered using the 'INPUT' program from Newsletter 2 - 2 (NOV 84), send $1 to S.A.U.G. for printout.

```
AO      10 REM CLEAR
JI      15 REM Demonstrates how to reserve                memory using CLEAR
BP      20 REM MS!= Start  of MCR                         ML%= Length of MCR
                           EP!= Entry  of MCR
CA      25 ML%=28
CE      30 CLS:PRINT"1: Manual Reserve":PRINT"2: Auto reserve":INPUTT:IFT<1
           ORT>2GOTO30
CM      35 ONTGOTO45,65
AC      40 REM Version 1:reserves space under             standard location
GB      45 PRINT"Current Top of Memory is either:":PRINT"1: F500H for a tap
           e system   or":PRINT"2: D5B8H for a disk system":INPUT"Input whi
           ch ";T:IFT<1ORT>2GOTO45
KE      50 IFT=1THEN MS!=&HF500 ELSE MS!=&HD5B8
AB      55 GOTO75
AN      60 REM Version 2:top of memory is                 unknown
AJ      65 GOSUB70:GOTO75
BC      70 MS!=PEEK(&HFDE6)+PEEK(&HFDE7)*256:RETURN
JO      75 MS!=MS!-ML%:CLEAR1000,MS!:GOSUB 70:PRINT"Memory is reserved from
            "HEX$(MS!):PRINT
DB      80 ML%=28:EP!=MS!
END
```

SPECTRAVIDEO™
AUSTRALASIAN USERS GROUP

LIBRARY NOTES

By J. Collins

A busy month for the library again with most members now aware of the way in which charges are levied and who has to pay for what to whom. Still the odd trickle of members writing in for Mysterious Manor and Count Dracular to be supplied on disk. To those members I can only repeat "PLEASE READ YOUR NEWSLETTER CAREFULLY". In fact the author of Mysterious Manor has come up with a partial answer to the problem and anyone who wants the program can now ask for it on disk and this can be done but you still can not run it from disk. You will still have to download from the disk to cassette in two seperate loads and then using the MERGE command and one of the halves as an ASCII file join them back together again to be run from Cassette Basic. Quite frankly I would not recommend this procedure to anyone not fully familiar with his machine and the uses of ASCII files. However the facility is there for those who may wish to try it, but please be quite specific about what you want. Count Dracular has not been altered and is still only available on cassette for 318 with extra memory or 328s.

Which brings me to the next point.....When ordering library software please state clearly whether it is required on cassette or disk, and it wouldn't hurt to mention which computer you have.....318 or 328. These may seem like minor points but if you give full information first time then your order will not be delayed while I write back requesting clarification.

At the end of this article there is an update to the price/program list. Please note that they are all CP/M programs and as such may only be used by disk users who have the CP/M operating system. Those of you who are experienced hackers will know what they are and how to use them. Please write in for any further information or assistance necessary. In the very near future we will start a series on CP/M wherein we will attempt to explain what it is and how it works. Of necessity this will be confined to simple explanations of the fundamental parts of CP/M and will be aimed at stimulating interest among those members who have it, don't really understand it, and would like to expand their knowledge. There are probably 3000 or more programs of all types available to run under CP/M. WORDSTAR is a good example of this and is the word-processing program the Editor and I use to prepare the newsletter.

New progams available :-

CP/M 2.22 (with a complete documented and commented disassembly of the BIOS) This source listing has got to be worth its' weight in gold. You wont believe the information which has been collated and added to this disassembly.

CP/M MASTER CATALOG SYSTEM. (this allows you to number your

CONTINUED PAGE 11

SPECTRAVIDEO™
AUSTRALASIAN USERS GROUP

DATA 70

by : M. Bailey & P.Gee

This Program may be entered using the 'INPUT' program from Newsletter 2 - 2 (NOV 84).
Send $1 to S.A.U.G. for printout of article.

```
CN      10 FORT=2176 TO 2256
CJ      20 READA$
EH      30 A=VAL("&H"+A$)
BG      40 VPOKET,A
BC      50 NEXT
DI      60 FORY=2312 TO 2516
CP      70 READB$
EM      80 B=VAL("&H"+B$)
BH      90 VPOKEY,B
DF     100 NEXTY
AM     110 RESTORE 280
BF     120 FOR I=2568TO2768
AF     130 READC$
BH     140 C=VAL("&H"+C$)
DN     150 VPOKEI,C
CL     160 NEXTI
DE     170  DATA F8,88,88,88,98,98,F8,00
BP     180  DATA 20,20,20,20,60,60,60,00
CL     190  DATA F8,88,08,30,C0,C0,F8,00
BJ     200  DATA F8,88,08,30,18,98,F8,00
BI     210  DATA 80,80,B0,F8,30,30,30,00
DD     220  DATA F8,80,F8,18,18,98,F8,00
CJ     230  DATA F8,88,80,F8,98,98,F8,00
CN     240  DATA F8,08,10,10,30,30,30,00
CL     250  DATA F0,90,90,F8,98,98,F8,00
DP     260  DATA F8,88,F8,18,18,98,F8,00
BF     270  DATA 00
BM     280  DATA 70,50,50,f8,c8,c8,c8,00
EP     290  DATA f0,90,90,f8,c8,c8,f8,00
HD     300  DATA f8,88,80,80,c0,c8,f8,00
EB     310  DATA f8,88,88,c8,c8,c8,f8,00
FH     320  DATA f8,80,80,F0,C0,C0,f8,00
DN     330  DATA f8,80,80,f0,c0,c0,c0,00
CL     340  DATA F8,88,80,d8,c8,c8,f8,00
BB     350  DATA 88,88,88,f8,c8,c8,c8,00
BJ     360  DATA 40,40,40,60,60,60,60,00
EM     370  DATA 10,10,10,18,98,98,f8,00
BN     380  DATA 90,90,90,f8,c8,c8,c8,00
BL     390  DATA 80,80,80,c0,c0,c0,f8,00
DM     400  DATA f8,a8,a8,a8,a8,a8,a8,00
DN     410  DATA f8,88,88,c8,c8,c8,c8,00
EL     420  DATA F8,98,98,98,88,88,f8,00
EE     430  DATA f8,88,88,f8,c0,c0,c0,00
EE     440  DATA f8,88,88,88,88,b8,f8,00
CG     450  DATA F0,90,90,F8,C8,C8,C8,00
EC     460  DATA f8,88,80,f8,18,98,f8,00
ED     470  DATA f8,20,20,30,30,30,30,00
BJ     480  DATA 88,88,88,c8,c8,c8,f8,00
BO     490  DATA c8,c8,c8,c8,50,50,70,00
DL     500  DATA a8,a8,a8,a8,a8,a8,f8,00
DK     510  DATA 88,88,88,78,c8,c8,c8,00
```

```
BN      520  DATA 88,88,88,f8,20,20,20,00
GB      530  DATA f8,88,08,30,c8,c8,f8,00
END
```

DATA 70 is a small program that changes the S.V. character set to that found on the bottom of cheques. It should be a useful bit of software if you want some nice text in games e.t.c.

CONTINUED FROM PAGE 9

disks, give them a name if you wish, create a catalog, and if you have a printer you can print out pages of info on your disk collection) You can see your catalog on screen of course. If you're into keeping track of your tracks this is the program for you.

Both the above are offered as Public Domain Software and we ask only the nominal $3..00 to cover cost of copying and handling. If you want us to provide the disk then add $6..00 to make $9..00 total. Supply your own disk and pay only $3..00.

One very important point which should have been mentioned before concerns the CP/M system you purchased when you bought your disk drive/s. Each copy of CP/M sold has a unique serial number and is registered to the purchaser. If you are supplying blank disks to purchase CP/M programs they will be returned to you correctly formatted for CP/M but they will not be system disks.....you will have to SYSGEN your own system from your original Master Disk. Of course you can format them and make them system disks yourself before you send them to me.

That's all for this month.....>

SPECTRAVIDEO™
AUSTRALASIAN USERS GROUP

SOUND FACTORY

by : F.R. Brown

This Program may be entered using the 'INPUT' program from Newsletter 2 - 2 (NOV 84).
Send $1 to S.A.U.G. for printout of article.

```
DG      10 GOSUB1240
AJ      20 GOSUB840:GOSUB1110
DN      25 REM  Get input from the keyboard
AI      30 I$=INKEY$:IFI$=""GOTO30ELSEA=ASC(I$)
GD      35 REM  To speed things up a little
FB      40 IFA>54THENIFA>105GOTO260ELSEGOTO150
FI      50 IFA=27THENGOSUB1210:GOTO20'esc key
EH      60 IFA=48THENES=ES+1:IFES=10THENES=0:GOTO340:ELSEGOTO340'0 key
AK      70 IFA=45THENEP=1/1.1*EP-1:IFEP<Y4THENEP=Y4
DL      80 IFA=45THEN470'- key
OH      90 IFA=49THENT1=-T1:GOTO340'1 key
LE     100 IFA=50THENT2=-T2:GOTO340'2 key
LB     110 IFA=51THENT3=-T3:GOTO340'3 key
KI     120 IFA=52THENN1=-N1:GOTO340'4 key
KF     130 IFA=53THENN2=-N2:GOTO340'5 key
KC     140 IFA=54THENN3=-N3:GOTO340'6 key
JD     150 IFA=55THENE1=-E1:GOTO340'7 key
JA     160 IFA=56THENE2=-E2:GOTO340'8 key
IN     170 IFA=57THENE3=-E3:GOTO340'9 key
CC     180 IFA=58THENEP=1.1*EP+1:IFEP>Y8THENEP=Y8
GA     190 IFA=58THEN470': key
DG     200 IFA=100ANDE2=-1THEN590'd key
AM     210 IFA=101ANDE2=-1THEN570
DA     220 IFA=102ANDE3=-1THEN630'f key
DH     230 IFA=103ANDE1=-1THEN670'g key
DB     240 IFA=104ANDE2=-1THEN720'h key
BC     250 IFA=105THEN800'i key
DH     260 IFA=106ANDE3=-1THEN770'j key
BG     270 IFA=107THEN820'k key
DA     280 IFA=115ANDE1=-1THEN550's key
CK     290 IFA=114ANDE3=-1THEN610'r key
CE     300 IFA=116ANDE3=-1THEN650't key
CE     310 IFA=117ANDE3=-1THEN750'u key
CH     320 IFA=119ANDE1=-1THEN530'w key
BH     330 IFA=121ANDE2=-1THEN700'y key
FJ     340 REM Enter data into registers
BJ     350 X(12)=INT(EP/Y1)
AH     360 X7=192
DF     365 REM  Tone off
AL     370 IFT1=-1THENX7=X7+1
AM     380 IFT2=-1THENX7=X7+2
AO     390 IFT3=-1THENX7=X7+4
GA     395 REM  Noise off
CA     400 IFN1=-1THENX7=X7+8
BN     410 IFN2=-1THENX7=X7+16
BH     420 IFN3=-1THENX7=X7+32
CJ     430 X(7)=X7
BJ     435 REM  Give volume control to envelope
KJ     440 X(8)=A1:IFE1=1THENX(8)=16
KM     450 X(9)=A2:IFE2=1THENX(9)=16
AF     460 X(10)=A3:IFE3=1THENX(10)=16
```

```
BI    470 X(12)=INT(EP/Y1)
EG    480 X(11)=INT(EP-X(12)*Y1)
AM    490 X(13)=NV(ES)
BH    500 FORL=ØTO13:SOUNDL,X(L):NEXTL
DF    510 GOTO 2Ø
AB    520 REM Amplitude & period controls
GJ    525 REM  A1,2,3 channel1,2,3 amplitude
CI    530 A1=A1+1:IFA1>Y7THENA1=Y7
AF    540 GOTO 56Ø
CK    550 A1=A1-1:IFA1<Y4THENA1=Y4
DE    560 X(8)=A1: SOUND8,X(8):GOTO2Ø
CE    570 A2=A2+1:IFA2>Y7THENA2=Y7
AG    580 GOTO 6ØØ
CG    590 A2=A2-1:IFA2<Y4THENA2=Y4
DL    600 X(9)=A2: SOUND9,X(9):GOTO2Ø
CL    610 A3=A3+1:IFA3>Y7THENA3=Y7
AF    620 GOTO 64Ø
CN    630 A3=A3-1:IFA3<Y4THENA3=Y4
AH    635 REM  P1,2,3 channel 1,2,3 period (1/frequency)
AO    640 X(1Ø)=A3:SOUND1Ø,X(1Ø):GOTO2Ø
DL    650 P1=INT(P1*Q1+1):IFP1>Y5THENP1=Y5
AF    660 GOTO 68Ø
DO    670 P1=INT(P1/Q1-1):IFP1<Y4THENP1=Y4
EL    680 X(1)=INT(P1/Y1):X(Ø)=P1-X(1)*Y1
HM    690 SOUNDØ,X(Ø):SOUND1,X(1): GOTO 2Ø
EB    700 P2=INT(P2*Q1+1):IFP2>Y5THENP2=Y5
AF    710 GOTO 73Ø
EE    720 P2=INT(P2/Q1-1):IFP2<Y4THENP2=Y4
EN    730 X(3)=INT(P2/Y1):X(2)=P2-X(3)*Y1
IK    740 SOUND2,X(2):SOUND3,X(3): GOTO 2Ø
DM    750 P3=INT(P3*Q1+1):IFP3>Y5THENP3=Y5
AF    760 GOTO 78Ø
DP    770 P3=INT(P3/Q1-1):IFP3<Y4THENP3=Y4
FK    780 X(5)=INT(P3/Y1):X(4)=P3-X(5)*Y1
IN    790 SOUND4,X(4):SOUND5,X(5): GOTO 2Ø
CO    800 NP=NP+1:IFNP>Y6THENNP=Y6
AF    810 GOTO 83Ø
DA    820 NP=NP-1:IFNP<Y4THENNP=Y4
CG    830 X(6)=NP:SOUND6,X(6):GOTO2Ø
EP    840 REM reg print out
AD    850 PRINTCHR$(11)
IA    860 FORL=ØTO4:PRINT" R"+RIGHT$(STR$(L),1)+"=";:PRINTUSING"###";X(L);
          :NEXT
BM    870 PRINT
IC    880 FORL=5TO9:PRINT" R"+RIGHT$(STR$(L),1)+"=";:PRINTUSING"###";X(L);
          :NEXT
BD    890 PRINT
GN    900 FORL=1ØTO13:PRINT" R"+RIGHT$(STR$(L),2)+"=";:PRINTUSING"###";X(L
          );:NEXT
BF    910 PRINT
BN    920 PRINT" env sh="ES;NV$(ES)
CG    930 RETURN
IO    935 REM  Print out block diag. of sound chip
AB    940  PRINT"              CH 1      CH 2      CH 3"
IA    950 PRINT"_______   ________  ________  ________"
FM    960 PRINT"| NOISE |   | TONE |  | TONE |  | TONE |"
GN    970 PRINT"---┌---   ----┌---  ----┌---  ----┌---"
```

```
LP    980 PRINT"    |
KA    990 PRINT"    |_________|_________|______   |"
MP   1000 PRINT"              |         |         |"
EP   1010 PRINT"           _|__|___  _|__|___  _|__|___"
DA   1020 PRINT"          |MIXER  |  |MIXER  |  |MIXER  |"
GD   1030 PRINT"          ----|---  ----|---  ----|---"
FC   1040 PRINT"__________"
CH   1050 PRINT"|ENVELOPE |___|_|________|_|_______|  |"
ML   1060 PRINT"|  SHAPE  |     |         |         |"
GH   1070 PRINT"|   AND   |  __|_________|_________|"
BF   1080 PRINT"|  PERIOD |__|      OUTPUT MIXER      |"
FB   1090 PRINT"---------  ----------|---------- "
MN   1100 PRINT"                     OUTPUT |"
CK   1105 REM  control of switches on block diag.
MK   1110 IFT1=1THENLOCATE14,10:PRINT"|" ELSELOCATE14,10:PRINT"\"
MM   1120 IFT2=1THENLOCATE24,10:PRINT"|" ELSELOCATE24,10:PRINT"\"
MO   1130 IFT3=1THENLOCATE34,10:PRINT"|" ELSELOCATE34,10:PRINT"\"
NB   1140 IFN1=1THENLOCATE11,12:PRINT"|" ELSELOCATE11,12:PRINT"\"
ND   1150 IFN2=1THENLOCATE21,12:PRINT"|" ELSELOCATE21,12:PRINT"\"
NF   1160 IFN3=1THENLOCATE31,12:PRINT"|" ELSELOCATE31,12:PRINT"\"
MG   1170 IFE1=1THENLOCATE13,16:PRINT"/ " ELSELOCATE13,16:PRINT" \"
MI   1180 IFE2=1THENLOCATE23,16:PRINT"/ " ELSELOCATE23,16:PRINT" \"
MK   1190 IFE3=1THENLOCATE33,16:PRINT"/ " ELSELOCATE33,16:PRINT" \"
AG   1200 PRINTCHR$(11): RETURN
ID   1205 REM  Esc (reset) routine
FG   1210 FORL=ØTO13:SOUNDL,Ø:NEXT:ES=Ø
EF   1220 FORL=ØTO13:X(L)=Ø:NEXT
LF   1221 T1=-1:T2=-1:T3=-1
KC   1222 N1=-1:N2=-1:N3=-1
IG   1223 E1=-1:E2=-1:E3=-1
BC   1230 RETURN
BE   1240 CLICKOFF:SCREEN2,Ø
FF   1250 CLS:PRINT:PRINT" SOUND":PRINT" FACTORY":FORZ=1TO15ØØ:NEXT
BA   1255 REM  The poke turns off the caps lock regardles of what the ligh
          t says
AH   1260 Y1=256:FORI=ØTO13:SOUNDI,Ø:NEXT:POKE&HFE38,Ø
AB   1270 DP=4:DIMX(13):Y4=Ø:Y5=4Ø95:Y6=31
JJ   1280 Q1=1.Ø1:Y7=15:Y8=65534!
BA   1290 FORL=ØTO9:READNV(L):NEXT
GI   1300 DATAØ,4,8,9,1Ø,11,12,13,14,15
GL   1310 T1=-1:T2=-1:T3=-1:N1=-1:N2=-1:N3=-1:E1=-1:E2=-1:E3=-1
BH   1320 NV$(Ø)="\_________"
FL   1330 NV$(1)="◢_________"
DM   1340 NV$(2)="◣◣◣◣◣◣◣◣◣"
BB   1350 NV$(3)="\_________"
FO   1360 NV$(4)="◥◤◥◤◥◤◥◤◥"
GE   1370 NV$(5)="◥---------"
EE   1380 NV$(6)="◢◢◢◢◢◢◢◢◢"
DJ   1390 NV$(7)="/---------"
BN   1400 NV$(8)="◣◢◣◢◣◢◣◢◣"
GA   1410 NV$(9)="◢_________"
DG   1420 GOSUB144Ø
AC   1430 GOSUB84Ø:PRINT:GOSUB94Ø:RETURN
AO   1435 REM  Print controls
AG   1440 SCREENØ,Ø:CLS:FX$="INST"
BG   1445 POKE&HFAØ5,Ø:REM cursor off
EG   1450 INPUT"    WOULD YOU LIKE A PRINTED COPY                OF THE CONTRO
          L KEYS";IP$:CLS
```

```
DL   1460 IFASC(IP$)=121THENFX$="LPT:"+FX$:ELSEFX$="CRT:"+FX$
FO   1470 OPEN FX$ FOR OUTPUT AS #1
DK   1480 IFASC(IP$)=121THENINPUT"IS THE PRINTER ON LINE";IP$:PRINT#1,STRI
          NG$(79,42)
FH   1490 PRINT#1,SPC(10)"CONTROL KEYS":PRINT
BC   1500 PRINT#1,"[7]E1  [8]E2  [9]E3    <TOGGLE"
CD   1510 PRINT#1,"[4]N1  [5]N2  [6]N3    <SWITCHES"
CP   1520 PRINT#1,"[1]T1  [2]T2  [3]T3    <ON/OFF":PRINT#1,
AH   1530 PRINT#1,"          I-AMPLITUDE-I I----PERIOD---I"
CP   1540 PRINT#1,
BL   1550 PRINT#1,"INCREASE [W] [E] [R]   [T] [Y] [U] [I]"
BC   1560 PRINT#1,"DECREASE [S] [D] [F]   [G] [H] [J] [K]"
CJ   1570 PRINT#1,"CHANNEL   1   2   3     1   2   3 NOISE"
FC   1580 PRINT#1,:PRINT#1,"   ENVELOPE CONTROL":PRINT#1,
DE   1590 PRINT#1,"[0]INCREMENTS THE ENVELOPE SHAPE"
BN   1600 PRINT#1,"[:]INCREASE ENVELOPE PERIOD"
AK   1610 PRINT#1,"[-]DECREASE ENVELOPE PERIOD":PRINT#1," "
CO   1615 PRINT#1,"[esc]RESET ALL REGISTERS":PRINT
EG   1620 IFASC(IP$)=121THENPRINT#1,STRING$(79,42)
HC   1630 PRINT:PRINT"    PRESS ANY KEY TO CONTINUE"
ED   1640 IP$=INKEY$:IFIP$=""THEN1640 ELSECLS:RETURN
CI   1990 'A little about this program
KM   2000 'The controlling sections and basic idea for this program where
          written by P.Beckett of Blackpool U.K. Conversion to the SPECTRA
          VIDEO Computer and graphics sequences by F.Brown S.A.U.G. in Jan
          uary 1985.
BH   2010 'This program was first published in August 1982 in Practical El
          ectronics magazine. It was designed for use with a UK101 or OSI
          Superboard, (Does any one remember Ohio Scientific?), fitted wit
          h a Practical Electronics Sound Board.
BD   2020 'To save on typing and for a slight increase in speed please del
          ete all REM's.
EK   2030 'Generally tone and noise periods cannot be altered unless that
          channel has been selected. Also when altering the envelope perio
          d the true sound cannot be heard until the key is released as th
          e program will be updating the registers and each update
JK   2040 'will restart the envelope cycle. This can be used to hear the s
          ound of one cycle envelopes,press the envelope eneble key for th
          e channel you are using twice, the first press turns the envelop
          e off the second restarts the envelope cycle.
END
```

SPECTRAVIDEO™

AUSTRALASIAN USERS GROUP

buy sell or trade

FOR SALE

SPECTRAVIDEO ROM BASIC EXPLAINED. Book still available from Users Group for $20.00 includes P & P.

NOTE : Stocks are very low and a reprint is now planned for some time.

SEND TO:

S.A.U.G. (book),
P.O. Box 191,
South Launceston,
Tasmania. 7249.

FOR SALE

1 Only Centronics Interface for the Expander.

Contact:

Call 44 6865

FOR SALE

3 SPECTRA HOHE ECONOMIST PACKS

In original packing. Never used. Sale price $30. Asking price $5 $2 P&P each.

Contact:

T. Kermanidis,
15 George St.,
Pennington,
South Australia. 5013