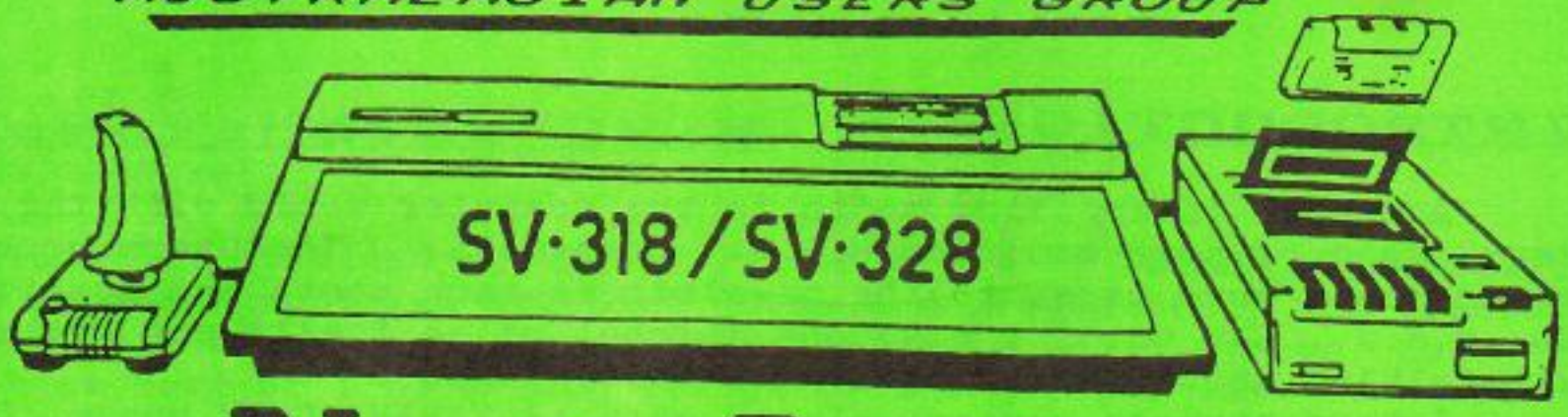
SPECTRAVIDEO™

AUSTRALASIAN USERS GROUP



News Letter

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

ISSUE No.
2 - 6

DATE
MAR - 1985

INTRODUCTION	2
UNDERSTANDING CP/M	3
TYPE (Program)	6
DOUBLE SIDED DISK DRIVES	7
TEXT (GRID)	8
EXPLORING BASIC PT-10	10
POKE	13
STRING	13
DATA	14
ARRAY	14
CLARKE ELECTRONICS	15
BUY, SELL OR TRADE	16

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

Another interesting month in the computer world with the release very soon of some exciting M.S.X. Computers. The Users group will be adding 4 more pages to the newsletters each month and devoting them to M.S.X. But until I have one infront of me I will not get my feet wet talking about something I know little about.

I have however already played with he YAMAHA CX5M Music Computer also sold by Rose Music (free plug). This is an M.S.X. computer and you will be pleased to note that is was'nt all that much different from a 328. I will list differences as soon as I get some reliable information about M.S.X.

I am not yet allowed to tell you about the SV-728 or the SV-EXPRESS which are to be released any time now. So next month all going well we will have a full review on these machines. So until then I did'nt tell you about them, DID I ! ! ! !

THE 1983/84 YEAR BOOK NOW AVAILABLE

So to all you people that have asked about back issues this is the book for you.

There were 15 Newsletters in 1983/84 and these are reprinted in this book. Also a little bit extra is added which will not appear in any newsletter to entice you to purchase it. For all you Exploring Basic fans parts 1 to 8 are in this book. Cost has been held down to a minimum for you and only $15 which includes post and pack.

In this months newsletter we have the following.

UNDERSTANDING CP/M for all you Disk Drive owners who have not as yet found the wonders of this magic operating system.

TYPE, a typing tutor which will give you details on SUCCESS RATE and TYPING RATE, which is in Characters per Minute.

DOUBLE SIDED DISK DRIVES is some information on the exciting new disk format that is available for Spectravideo Disk Owners. An upgrade to this format for ALL DISK OWNERS is a must.

The large Middle page grid is from Mike Johns and for the next few newsletters we will also have Grids for SCREEN 1, SCREEN 2 and SPRITES. We hope they will be of some help in Programming.

Thanks again to Mr. Dunning, his article speaks for itself.

Finally I would like to greet CLARKE ELECTRONICS to the S.V. Family and hope we can look forward to more tips as the year goes on.

Happy computing and don't eat too many Easter Eggs.

SPECTRAVIDEO™
AUSTRALASIAN USERS GROUP

Understanding CP/M

By. S.W. McNamee.

Since I obtained my SV-328 I have met many other Spectravideo owners and had great fun swapping ideas and programmes, and generally enjoying the computing power now available to me in my own home. In this time however I have been struck by the general lack of knowledge most people seem to have about the CP/M operating system available to disk drive owners. Indeed it seems that there is a definate reluctance to even try using this system and I have come to the conclusion that this is caused partly by the somewhat technical nature of most CP/M documentation and partly by ignorance of just what CP/M is. So I decided to attempt to impart a little bit of my hard won information and this series of articles is the result.

Well to start at the beginning as the saying goes, just what is CP/M ? Firstly it is NOT a language. I have lost track of the number of times I have heard people referring to something being "written in CP/M". CP/M is a combination Monitor and Disk Operating System (DOS) to control the operation of Z80 and 8080 based microcomputers. It provides routines that allow an operator to input information to the computer, (via the keyboard), see information coming out of the computer, (via the T.V. monitor - also known as the terminal) and to control the storage of information in a logical manner on disk drives. Basically this is ALL that CP/M does. What is really meant by saying something is written in CP/M, is that it is written in a form that is suitable for storing on disk, and being retrieved and run using the CP/M operating system. It is written in some language (usually assembler) FOR CP/M.

CP/M consists of four main parts.

1. The Command PRocessor or CPR
2. The Basic Disk Operating System or BDOS
3. The Basic Input Output System or BIOS
4. The programmes and files that can be loaded and run

The first three of these together comprise the machine instructions that are loaded into the computer when it is first turned on - sometimes called a "cold boot". The fourth part is made up of the files that come with the CP/M system disk and can be added to with any of the thousands of commercial and public domain programmes available for use under the CP/M operating system.

CPR

This is the part of the system that has control when you are entering commands into the computer. It can be recognised by the "System Prompt" either A> or A0>. The CPR takes information typed at the keyboard, formats it in certain ways, places parts of it in standard memory locations and then acts upon the commands given. It has several built in functions that enable the operator to do the following:

DIR	Display a directory of files on a disk
ERA	Erase specified files from a disk
REN	Rename a specified disk file
SAVE	Save a specified portion of memory to a file
TYPE	Type a disk file on the T.V. monitor
USER	Change the current user number (More will be said about this later)

If the command typed is not one of these the CPR will assume that it is the name of a file on disk and will attempt to load and then run this programme.

BDOS

This is the part of the system that manages the disk files. It Keeps track of where everything is on the disk, maintains a directory for that disk and allows programmes to be stored and retrieved from disk in a logical manner. It also allows programmes to take input from the keyboard and other devices, and send output to the T.V. monitor and other devices. Programmes can access the BDOS using a standard memory location and a standard protocol. The BDOS and CPR are exactly the same on every computer which runs CP/M which is why a programme written under CP/M can run on any computer that uses this operating system.

BIOS

This part of the system provides routines that do the actual disk reading and writing operations and control the information that is to be sent out or received in. This is the part that is different for each type of computer which runs CP/M and programmes that use parts of the BIOS directly will generally be unable to run on another brand of computer. This part of the system can often be modified by the knowledgable user to enhance the features available in it.

PROGRAMMES

These comprise all the different applications that can be loaded into the computer from disk using the operating system and run. They comprise Language translators - compilers and interpreters (BASIC,FORTRAN,PASCAL etc.); Text editors - wordprocessors etc.; Assemblers - allow programmes to be written using a text editor and "assembled" into machine code; Utilities - to copy files, copy disks, examine memory locations etc. ... the list is endless. So you can see the vast number of applications that are opened up to you when you become familiar with using CP/M. It should be noted here that all programmes that are to run under CP/M consist of Machine Code which is the pattern of numbers understood by the Microprocessor in the computer. It is the job of a language translator or assembler to produce the correct Machine Code from any programme written using a text editor.

The distribution disk that you get with your CP/M contains several of these types of programme. Later I will discuss each of them and explain how to use the more important ones.

We now come to a discussion of the CP/M file system, how it is organised, named and accessed.

Files are organised in BLOCKS of data of a certain size. A block is the smallest amount of storage space that can be allocated on the disk and file storage procedes in integral numbers of blocks. There are several possible block sizes ranging from 1k bytes up to 16k bytes.

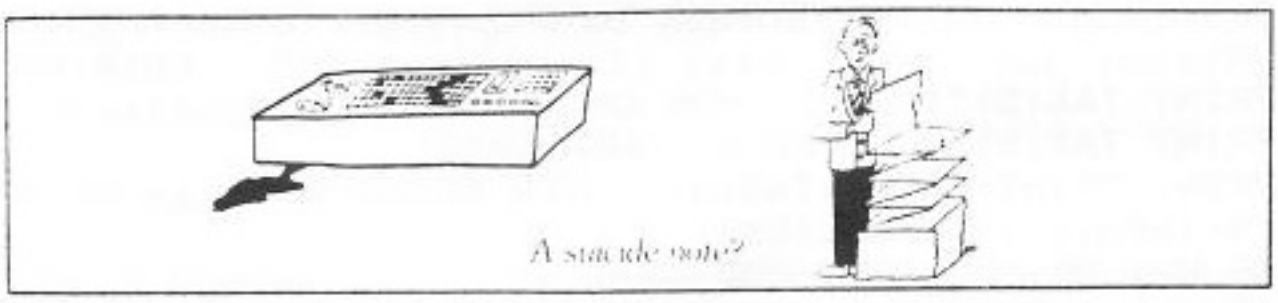
A full file specification (abreviated to filespec) consists of a drive designation letter (A to P), a colon ":", a file name of up to 8 characters, a period "." and a file extension of up to 3 characters. Eg. B:STAT.COM If the file name is shorter than 8 cahracters it is padded with spaces as is the extension if it is shorter than 3 characters long.

Any file that has an extension .COM can be loaded and run from the CPR by simply typing the file name (without the extension) followed by <CR> immediately after the system prompt (A> or B>). Eg. A>stat<CR> loads and runs the file STAT.COM

A file can also be created using the CPR command "SAVE". Files can also be accesed from within a programme using the BDOS functions but discussion of this must wait for a later installment.

N.B. In all these articles whenever you see "<CR>" it means press the "ENTER" key.

Next time I will discuss the CPR resident commands in detail and some of the COMmand files on the CP/M distribution disk, and how to run and use various application programmes. Later on I will talk about the more technical details of CP/M and how to write programmes that use its' features. Finally I will discuss the implementaion of CP/M as it is peculiar to the Spectravideo and examine the BIOS in full detail. If there is anything specific about the CP/M system that you would like explained in more detail please let the editor know and I will try to include it at the appropriate time.



BRISBANE MEMBERS PLEASE NOTE

A meeting of Spectravideo Users is being held at 7 Gables Street, Stafford Heights, on the 1st Tuesday of the Month. Next meeting will be 4-APRIL-1985. You are welcome to attend. We are hoping Ian Evans (computer sales Rose Music) will be able to attend, to give a small talk on Spectravideo and what is comming.

SPECTRAVIDEO™
AUSTRALASIAN USERS GROUP

TYPE

by : T. McGee

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.

```
BE      5 '** TYPE **
CD    100 LOCATE 0,0,0:SCREEN 0,0
AJ    120 GOSUB 600:GOSUB 620
EL    140 T=RND(-TIME):C=200:F$="####":G$="###.##":NM=-1
BL    160 LOCATE 0,0,0
CH    180 GOSUB 600
EM    200 PRINT:PRINT TAB(4);"  Press [ENTER] to end test. "
OO    220 FOR Y=8 TO 12:LOCATE 18,Y:PRINT"■■■■■":NEXT
FA    240 LOCATE 2,5:PRINT "  TIME";TAB(10)"NUMBER";TAB(18)"CORRECT";TAB(2
          8)"WRONG"
AM    260 TIME=0
AD    280 LOCATE 20,10:PRINT CHR$(32)
FI    300 A=INT(RND(1)*K+32):NM=NM+1:LOCATE 11,6:PRINT USING F$;NM
BP    320 LOCATE 20,10:PRINT CHR$(27)+"p";CHR$(A);CHR$(27)+"q"
BG    340 IF C<30 THEN C=30
AH    360 INTERVAL ON: ON INTERVAL=C GOSUB 480
HA    380 S=INT(TIME/50*100)/100:LOCATE 2,6:PRINT USING G$;S:Q$="":Q$=INKE
          Y$:IF Q$="" THEN 380
CJ    400 IF Q$=CHR$(13) THEN 500
AG    420 INTERVAL OFF:IF ASC(Q$)=A THEN C=C-5:SC=SC+1:LOCATE 19,6:PRINT U
          SING F$;SC:PLAY"O5V12T255G":GOTO 460
AD    440 C=C+10:W=W+1:PLAY"O2V12T225A":LOCATE 28,6:PRINT USING F$;W
AH    460 GOTO 280
CD    480 INTERVAL OFF:PLAY"O2V12T255A":C=C+10:W=W+1:LOCATE 28,6:PRINT USI
          NG F$;W:Q$="":GOTO 280
DN    500 LOCATE 12,14,1:PRINT"TEST TERMINATED":PRINT
BL    520 PRINT "SUCCESS RATE = ";USING G$;INT(SC/NM*10000)/100;:PRINT " %
          "
EK    540 PRINT" TYPING RATE = ";USING G$;INT(NM*60/S*100)/100;:PRINT " C.
          P.MINUTE"
CB    560 SCREEN ,1:END
BI    600 CLS:PRINT TAB(14)"TYPING TUTOR":PRINT TAB(14)STRING$(12,"¯"):RET
          URN
CP    620 PRINT TAB(5)"1 UPER CASE ONLY"
HJ    640 PRINT TAB(5)"2 UPPER & LOWER CASE"
BG    660 PRINT:PRINT:PRINT TAB(5)"ENTER SELECTION"
GN    680 K$=INPUT$(1):K=VAL(K$)
AB    700 IF K<1 OR K>2 THEN 680
GH    720 IF K=1 THEN K=64:POKE &HFE38,32:OUT &H88,15:OUT &H8C,255:GOTO 78
          0
CN    740 IF K=2 THEN K=96:POKE &HFE38,0:OUT &H88,15:OUT &H8C,223:GOTO 780
CP    780 ::
EM    800 FOR T=10 TO 0 STEP -1
AJ    820 LOCATE 18,10,0:PRINT USING"###";T
CP    840 FOR Z=1 TO 200:NEXT
CF    860 NEXT
CH    920 RETURN
END
```

SPECTRAVIDEO™

AUSTRALASIAN USERS GROUP

DOUBLE SIDED DISK DRIVES

Double sided 40 track disk drives are now available for use with super expander SV 601 and disk controller SV 801.
The drives come complete with modified CP/M BIOS to allow reading and writing of many different formats including:

Custom double sided format
Spectravideo single sided format
Spectravideo double sided format (CP/M Version 2.23)
Microbee D/S D/D format
Kaypro 2 S/S D/D format
Osbourne 1 S/S S/D format
and several others - the list is being constantly updated.
(24 Formats at time of printing E.D.)

BASIC disks can still be used but only as normal single sided disks.
Under the CP/M operating system each drive has a massive formatted capacity of 382k .
The drives need a small modification to the disk controller card and instructions for doing this are included. Alternatively the card may be sent with your order and the modification will be done free of charge.
Standard Spectravideo disk drives will still work with the modified controller card.
The disk drives come with the manufacturers guarantee and all other work is guaranteed for three (3) months from date of dispatch.

Prices: 1 Drive - $450.00
2 Drives - $850.00
Including post and pack.

If you Already Own a Disk

Also available is a modified BIOS to use with new Spectravideo double sided drives, and a BIOS for use with standard single sided drives giving 186K of storage and most of the features of the custom double sided BIOS. These also will read other disk formats. Call for further information.

Cost is $25.00 ea.

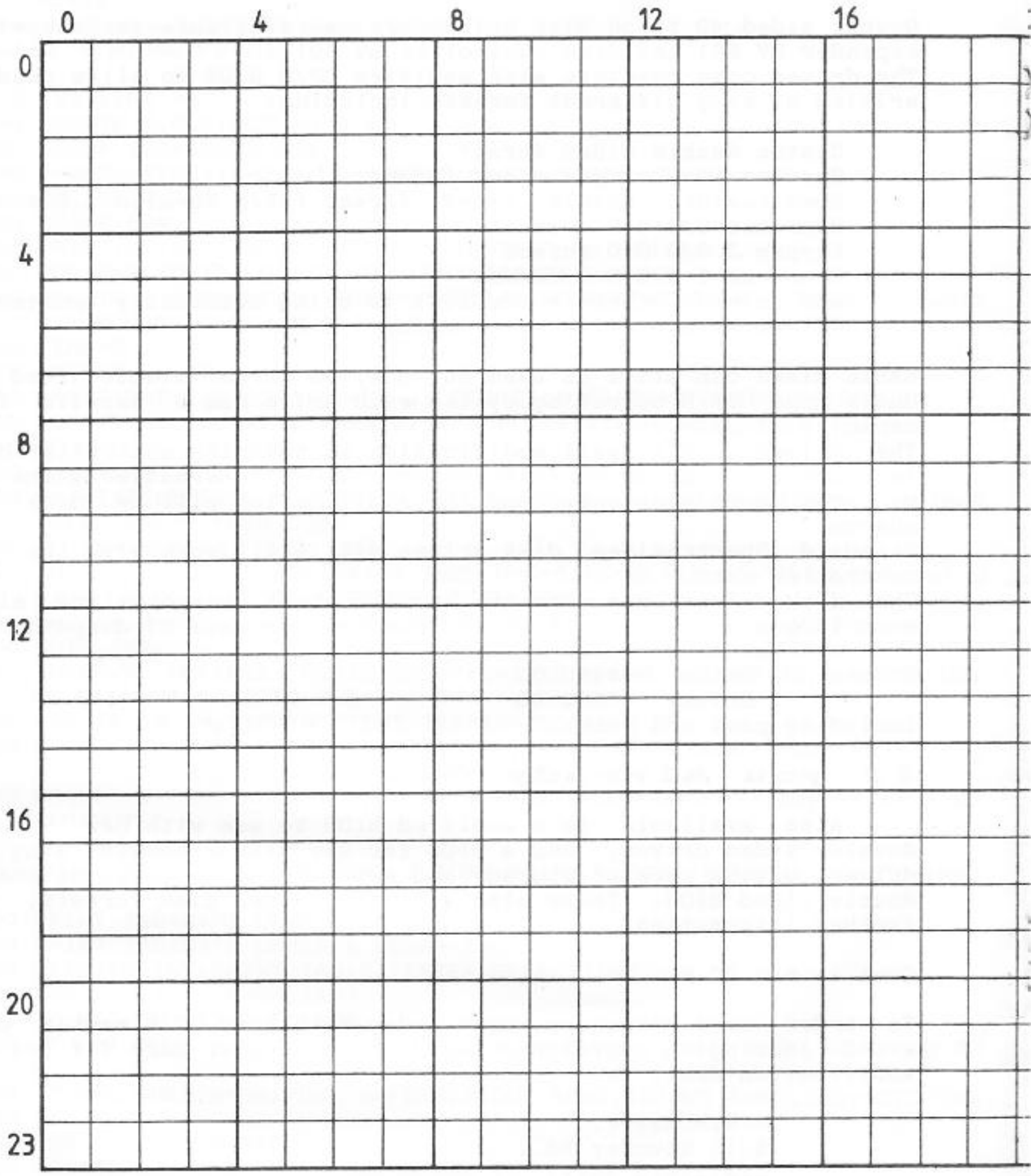
To order send cheque or money order with your CP/M master disk (to avoid infringing copyright) and disk controller card (if you want the modification done) to:

S.W.McNamee,
5/15 Stuckey Rd.,
Clayfield 4011.

If you would like further details phone 07-8358683 during working hours.

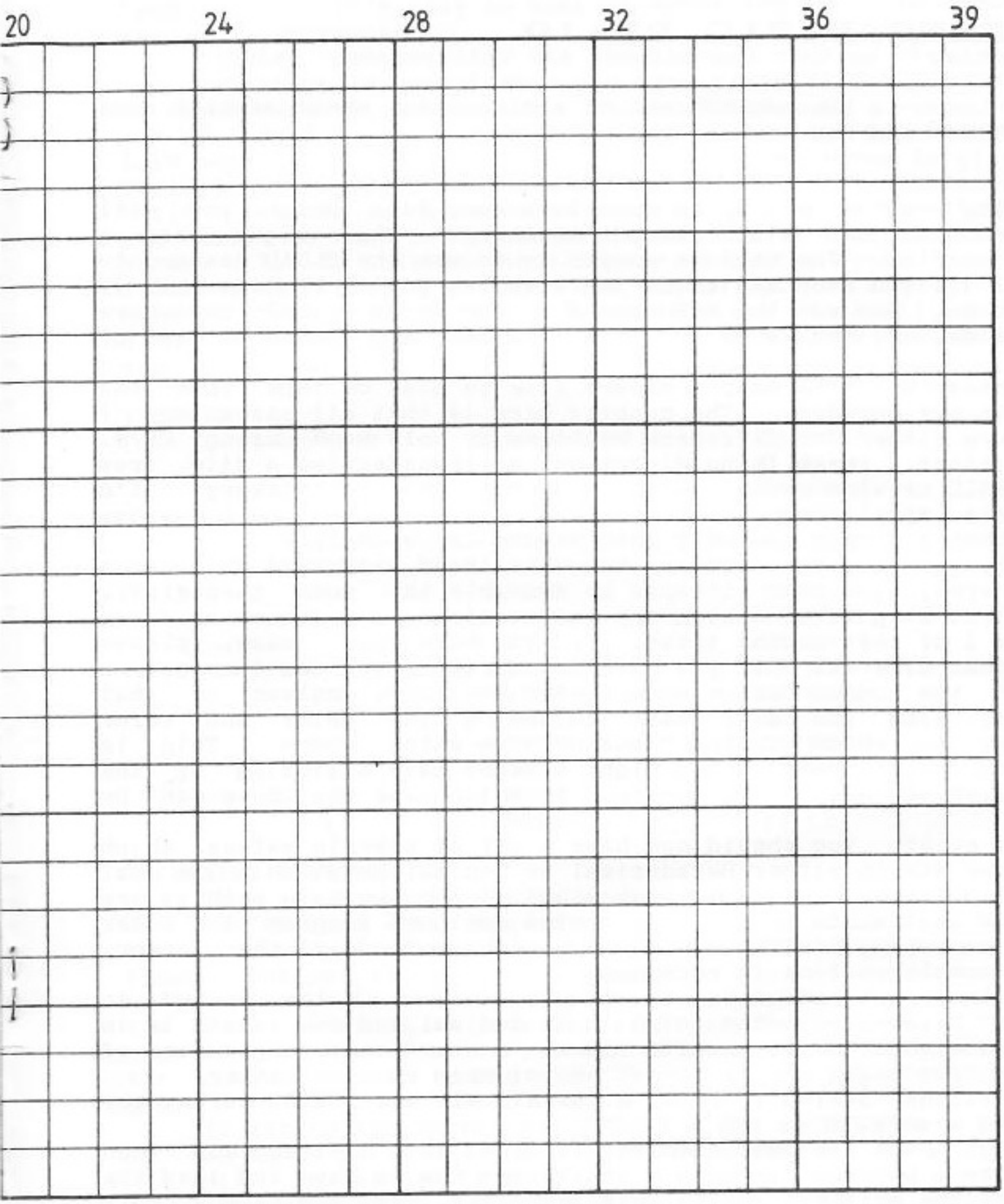
SPECTR
AUSTRALASIAN

TEXT



AVIDEO™

USERS GROUP



SPECTRAVIDEO™
AUSTRALASIAN USERS GROUP

EXPLORING BASIC PT-10

By L.A. Dunning

This issue is the second part of a discussion about machine code routines (MCRs) and how to use them.

LOADING?

Assuming you've set up an area in memory into which you will place your MCR, how will you put it there? This might seem a pointless question. The obvious answer is to use the BLOAD command to load it in. If you stop and think for a while, you'll realise this is not a solution. How was the MCR loaded in the first place? Somewhere along the line, it had to be put in memory and then BSAVEd to tape or disk.

An assembler will dump a binary file to disk or tape when the instructions are encoded. The problem here is that all assemblers I know of are either for different machines or only work using CP/M. With the latter, there is no direct way of transferring a file from CP/m to BASIC or vice versa, without a great deal of trickery. I'd like to see myself proven wrong here, however we shall assume therefore that a direct assembly into memory is impossible.

This leaves us with a process known as 'hand assembly' which is a process whereby the user attempts to assemble the code themselves.

The first step is to design the instructions in a manner as shown in diagram 1 of last months issue. If you have that issue, please refer to that diagram. If you have an assembler you can then get a listing of the codes as is shown under the code column of that diagram; or else you will have to use a Z80 guide and refer individually to which instructions produce which codes. This is tedious and error prone. You might already have a listing of the codes you wish to insert into memory, in which case the above can be bypassed.

In any event, you should now have a set of numeric values, which represent the MCR in either Hexadecimal or Decimal notation. The next step is to put these into data statements. the two main methods are to use DATA statements or to use a custom designed program to enter them into memory. Listing 'DATA' shows four methods of the former, using our example routine in each case.

The first set of data is in Hexadecimal, the second in 'compressed' hexadecimal, the third is in decimal and the fourth is in a form used with a 'magic array'. Each is shown on a single line of DATA, though you could easily format any of them to look better.

The listings 'STRING', 'POKE' & 'ARRAY' all use 'DATA' to put the routine into memory.

'STRING' puts the data (decimal is used) into a string and then LSETs it into a buffer. It also demonstrates how to save and load the routine from the buffer. Using this technique, you can save a whole library of short routines on disk and call them up when necessary.

'POKE' puts the data into memory by POKEing it there. This is a simple procedure. The three types of data can be used. Hexadecimal is handy when you find a bug and want to change it directly in the data statements, because the code is the same as that produced by the assembler. Compressed hexadecimal is a format I've invented to save space and is derived from a SPRITE pattern loader. Use this only if

you want to save space on long routines. Decimal format is OK, but on long routines it's easy to make an error typing it in, then harder to spot it afterwards.

'ARRAY' demonstrates the loading and running of a magic array. This technique is useful when you have parts of the MCR that you wish to alter directly from basic. Unlike POKEing in a value, you can just use US (##) = Value where ## is the element to be changed. I will talk about this later. The first byte in an array is given by VARPTR (##(0)) where ## is the name of the array. All numeric arrays have their elements in consecutive memory. In order to set up the array, you have to know how to convert the code in integer values for the array. To do this you should pair every two bytes in the routine, starting with the first two. The second byte of each pair is the high byte in the integer value. To cover, use ? &H2211 where 22 is the high value and 11 is the low. Table 1 shows how the example routine was converted.

Table 1

Array #	Integer Value	Codes L	Codes H
0	14906	3A	3A
1	-258	FE	FE
2	-16384	00	C0
3	33	21	00
4	-13056	00	CD
5	14140	3C	37
6	-22067	CD	A9
7	22298	1A	57
8	-16383	01	C0
9	31235	03	7A
10	-32557	D3	80
11	30731	0B	78
12	8369	B1	20
13	-13832	F8	C9

ACCESSING ROUTINES

Once you have the MCR in memory how do you access it? The first step is to know where the entry point is in memory. In our example, the entry point is also the start of the MCR, which is convenient. In other routines it could be anywhere in the routine or even somewhere in ROM! Assuming it is in the routine proper, you should calculate how many bytes it is from the start of the routine. Then, if the routine is relocatable, you can calculate the entry point by adding the correct number to the start location, which should be known to you.

To set up the entry, use DEFUSR = EP! or DEFUSR# = EP!, where EP! is the entry point value and # is the number of the routine (0 to 9, no value is assumed to be 0). The ten values are kept in pointers starting with F52BH & F52CH and ending at F53DH & F53EH. Normally however, you will not need to know this.

Thereafter, to execute the routine use JJ = USR(vv) or JJ = USR#(VV), where JJ is the 'return' variable, VV is the 'input' variable and # is the number of the routine called (no value is assumed to be 0). You probably know about both DEFUSR and USR(), but what you mightn't know is that a value can be passed to and from the MCR. The input variable is stored by BASIC at a location in memory

using parts of the INPUT routines.

Table 2

CALL 1AA9H	Passes input variable to register A this must be in a range of 0 - 255
CALL 1CC3H	Passes the input variable to register HL thus is in a range of 0 - 65535
CALL 183CH	Passes register A to the return variable
CALL 56C4H	Passes register HL to the return variable
CALL 55C9H	Passes register A to the return variable if register A is not zero, result is -1 otherwise result is 0

Table 2 lists CALLs that will pass the values to/from the routine and basic. If you examine diagram 1 from last month, you will see that this is how I get a value from BASIC to the routine, without pokes et cetera. There are other ways of doing this. If you have a fixed location to put values in, you can PEEK and POKE them in and out, however using CALLs seems much more convenient and uses no or little basic coding.

With magic arrays, you put the result in an array element however the result is limited to integers and you must set up the routine properly. For example the coding:

```
LD      A,(####H)
```

uses three bytes; 3AH followed by a low/high integer pair (eg. 3000H would be 00H followed by 30H) so the value is put in a single element, which means the 3AH is put in the high byte of the element before it. This is OK, except if the code should occupy the low byte in the pair. In this case you pad out the code with:

```
NOP
LD      A,(####H)
```

so that the first element contains 00 (NOP) and 3AH (LD A,) and the second has the value to be loaded. Using this method, you can put as many values as you like into the routine from basic, if they are set up properly.

PROGRAMS

Most programs in this issue are designed to be used with the listings in the previous part. Type in/load the programs and then save them on tape/disk in ASC format using the 'A' option.

'STRING' is designed to be used with 'BUFFER', merge it with that program.

'POKE' can be used with either 'BUFFER', 'CLEAR' or 'SNEAK'. Merge it with any of these.

'ARRAY' doesn't use any previous listings. It can be saved normally.

'DATA' is used with all of the above. It must be merged after the above actions have been done.

Next issue I'll talk about saving/reloading MCRs and some other aspects of machine code programming.

POKE

by : L.A. Dunning

```
CC    100 REM POKE
EJ    105 REM Demonstrates how to poke into                    memory an MCR from DA
          TA
EO    110 REM Requires a setup program from                    last article - either
           POKE,                BUFFER or SNEAK                          Merge w
          ith any of above
CH    115 REM Merge with DATA Listing before                   use
HP    120 PRINT"Which Method of Pokes to use?":PRINT"1: Hexadecimal":PRINT
          "2: Compressed Hexadecimal":PRINT"3: Decimal"
DL    125 INPUT T:IFT<1ORT>3GOTO125
BI    130 ONTGOSUB135,140,145:GOTO150
IF    135 RESTORE220:FORX=0TO27:READD$:VV=VAL("&h"+D$):POKEMS!+X,VV:NEXT:R
          ETURN
LK    140 RESTORE240:X=0:READD$:FORL=1TOLEN(D$)STEP2:VV=VAL("&h"+MID$(D$,L
          ,2)):POKEMS!+X,VV:X=X+1:NEXT:RETURN
BA    145 RESTORE260:FORX=0TO27:READVV:POKEMS!+X,VV:NEXT:RETURN
BE    150 DEFUSR=EP!
KH    155 LOCATE0,22:PRINTCHR$(27)"J";:INPUT"Screen Patten";Q:IFQ>-1THENQ=
          QMOD256:G=USR(Q):GOTO155
END
```

STRING

by : L.A. Dunning

```
BK    100 REM STRING
GO    105 REM Demontrates using MCRs in File                   Buffers - Must be mer
          ged with             BUFFER
CC    110 REM Also requres DATA
IC    115 RESTORE 260:Z$="":FORA=1TO28:READ VV:Z$=Z$+CHR$(VV):NEXT
BO    120 LSET MC$=Z$
CE    125 DEFUSR=EP!:BT=1
KK    130 LOCATE0,22:PRINTCHR$(27)"J";:INPUT"Screen Patten";Q:IFQ>-1THENQ=
          QMOD256:J=USR(Q):GOTO130
DO    135 REM Non Disk users delete following                  lines
FD    140 REM Shows saving & loading of MCR                    into buffer from disk
BF    145 CLS:GOSUB160
IG    150 PUT#2,1:LOCATE0,0:PRINT"Buffer saved to MCR.lib":LSETMC$="":PRIN
          T"Buffer cleared":GOSUB160
EP    155 LOCATE0,3:PRINT"Reloading buffer with MCR from   MCR.lib":GET#2,
          1:GOSUB160:GOTO165
CH    160 LOCATE0,14:PRINT"CONTENTS OF BUFFER #2":PRINTSTRING$(39,"-"):FO
          RA=0TO255:Q=PEEK(RS+A):VPOKE640+A,Q:NEXT:A$=INPUT$(1):RETURN
KK    165 LOCATE0,22:PRINTCHR$(27)"J";:INPUT"Screen Patten";Q:IFQ>-1THENQ=
          QMOD256:J=USR(Q):GOTO165
END
```

SPECTRAVIDEO™

AUSTRALASIAN USERS GROUP

DATA

by : L.A. Dunning

```
BC      200 REM DATA
EM      210 REM Hexadecimal Data
CE      220 DATA 3A,3A,FE,FE,00,C0,21,00,00,CD,3C,37,CD,A9,1A,57,01,C0,03,7A
            ,D3,80,0B,78,B1,20,F8,C9
CJ      230 REM Compressed Hexadecimal Data
AI      240 DATA 3A3AFEFE00C0210000CD3C37CDA91A5701C0037AD3800B78B120F8C9
CO      250 REM Decimal Data
FJ      260 DATA 58,58,254,254,0,192,33,0,0,205,60,55,205,169,26,87,1,192,3,
            122,211,128,11,120,177,32,248,201
CA      270 REM Magic Array Decimal Data
DD      280 DATA 14906,-258,-16384,33,-13056,14140,-22067,22298,-16383,31235
            ,-32557,30731,8369,-13832
END
```

ARRAY

by : L.A. Dunning

```
AC       10 REM ARRAY
DH       15 REM Illustrates Use of Magic Arrays          to load and run MCRs.
BP       20 REM MS!= Start  of MCR                       ML%= Length of MCR
                            EP!= Entry  of MCR
HG       25 REM Use only with INTEGER arrays
GH .     30 REM Merge with DATA listing before                running.
FF       35 DEFINTU:ML%=28:NE%=(ML%-1)\2:DIMUS(NE%)
GF       40 RESTORE280:FORA=0TONE%:READUS(A):NEXT
EE       45 MS!=VARPTR(US(0)):EP!=MS!:DEFUSR=EP!
KA       50 LOCATE0,22:PRINTCHR$(27)"J";:INPUT"Screen Patten";Q:IFQ>-1THENQ=
            QMOD256:J=USR(Q):GOTO50
END
```

SPECTRAVIDEO™

AUSTRALASIAN USERS GROUP

CLARKE ELECTRONICS

55-57 Belair Road, Kingswood, S.A. 5062
Telephone (08) 271 1216

Complete business system 328/605B includes 80 column card and green screen CP/M and micropro software.
$2399

Taxan colour monitors VISION EX gives bright clear colours on the spectravideo graphics $499

SP-1000 100CPS printer has graphics proportional printing forward and reverse paper feed and lots more
$380

NASHU DSDD 5.25" floppy disks 5 year warranty.
$2.99 1-9 $2.80 10-99

VERBATIM SSDD 5.25" floppy disks 1 year warranty.
$2.70 1-9 $2.60 10-99

As we are the SALES and SERVICE agents for SPECTRAVIDEO computers in South Australia we can deal directly with problems or modifications and upgrades.

P.S. The 16K to 64K conversion description in JANUARY 85 S.A.U.G. should include the removal of C53 to C60 which in the 16K version bypasses the supply rail. However in the 64K version pin 9 of the DRAMS is an address line and should not be bypassed.

TIP OF THE MONTH — **CLARKE ELECTRONICS** March 85

As we are also sales and service agents for electronic organs I soon noticed that the PLAY command in the spectravideo was 2 semitones flat, so that PLAYing the note "A" produced a "G". This would not normally cause problems unless you wish to use a piano or other fixed pitch instrument to play along with the spectravideo. The reason for the incorrect pitch is due to Spectravideo prototyping the software for the ROM basic with a 2 MHz clock feeding the AY-3-8910 sound generator chip. The frequency used in the production model is 1.789772MHz (a divided down NTSC Xtal 3.579545MHz). As the same oscillator supplies the cassette data rate you cannot just pop in a 4 MHz Xtal. However, all is not lost, below are 2 examples of a solution.

1) Modify your PLAY input strings: E.G. change "C" to "D" or "B" to "C#" etc.
Example: Original... 10 PLAY "CEFGGCEFGG"
Modified... 10 PLAY "DF#GAADF#GAA"

I am sure someone could even come up with a small utility to take original music strings and convert them to modified strings...

2) Add a 2 MHz clock on a little PC board inside the computer, disconnect the 1.789772 MHz from pin 22 of the AY-3-8910 and connect the 2 MHz clock output to pin 22.

Best Regards Eric Clarke.

SPECTRAVIDEO™

AUSTRALASIAN USERS GROUP

buy sell or trade

WANTED TO BUY

A Coleco games adaptor for as little as possible.

Please contact the Editor (Thats Me.)

S.A.U.G. or

Phone (003) 312648.

FOR SALE

A new SV 807 (64K Ram Expansion).

$220 o.n.o.

Please Contact:

David Martin,
20 Webb Street,
Henley Beach,
South Aust. 5022.
Or: (08) 356 4972.