SVI & MSX

— SPECTRAVIDEO —

NEWSLETTER

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

ISSUE NO.
2 - 7

ANNUAL SUBSCRIPTION	
AUSTRALIA	$20.00
OVERSEAS	$25.00
OVERSEAS AIRMAIL ...	$30.00
YEAR BOOK 83/84	$15.00

DATE
APRIL - 1985

CONTENTS

NEWSLETTER CORRESPONDENCE

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

(003) 312648

LIBRARY CORRESPONDENCE

S.A.U.G. LIBRARY,
1 CONRAD AVENUE,
GEORGE TOWN,
TASMANIA, 7253.

(003) 822919

INTRODUCTION

A few changes have occured over the past weeks. That is one reason why this Newsletter is a bit late.
The Users Group has now been running for some 18 months. In that time much has happened, one of the biggest problems we are now running into is the cost of living, in the past 18 months the cost of postage, printing, paper, power, phone calls e.t.c, e.t.c. (the list goes on).

I am sorry but we are forced to increase the membership price for this newsletter from $15 to $20. At the same time we are also increasing the size of the average newsletter from 16 to 20 pages.

We have also decided to include M.S.X. computer users in the club. Spectravideo has the SV-728,929 & EXPRESS which are all M.S.X. computers. But we are not limiting ourselves to just these, we are going to include ALL M.S.X. machines.

I have been avoiding M.S.X. for some months now as I have not had an M.S.X. computer to play with. But this changed this month when I was loaned the YAMAHA M.S.X.. My review of this computer and M.S.X. Basic can be found in the center 4 pages.

The rest of the newsletter needs no further comment. Just read and enjoy.

S.A.U.G. 1983/84 YEAR BOOK

WELL FOLKS YOU ASKED FOR IT, AND I PRODUCED IT. THE CLUB SPENT OVER $1,000 PRINTING IT. WE ARE SELLING IT FOR NO PROFIT, THE $15 PAYS FOR PRINTING, POSTING E.T.C.

SO WHY ARE YOU NOT BUYING IT ?????

IF YOU THE MEMBERS DON'T HELP US, WE WONT PRODUCE A1 GRADE PUBLICATIONS.

MOST USER GROUPS JUST PHOTOSTAT THEIR NEWSLETTER E.T.C., WE PAY FOR AN OFFSET PRINTER WHO GIVES US A TOP QUALITY ARTICLE.

ADDRESS: S.A.U.G. YEAR BOOK,
P.O. BOX 191,
LAUNCESTON SOUTH,
TAS. 7249.

COST: $15.00 (cheap)

P.S. SPECTRAVIDEO R.O.M. BASIC EXPLAINED BOOKS ARE ALSO STILL AVAILABLE. FROM US FOR $20.00.

EXPLORING BASIC PT-11

By L.A. Dunning

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

SAVING & LOADING

Assuming you have in some manner managed to put your MCR in memory, how do you go about saving it in a more permanent manner? Listing 'SAVE' demonstrates three techniques that can be used to do this.

A 'BLOCK' save is one using the BASIC command BSAVE. This command is designed to save any portion of memory. This is the simplest method to use. The result is a binary file on tape/disk that can be reloaded using BLOAD.

The format is:

```
     BSAVE filename, START, END, (ENTRY)
```

most parameters are obvious, however ENTRY (which is optional) refers to the entry point of a routine. If this is ommitted, the START parameter is assumed to be the ENTRY point. When reloading a binary file the format for BLOAD is:

```
     BLOAD filename ( ,R ) ( ,OFFSET )
```

'R' is a literal R. If this is used, BASIC jumps to the saved ENTRY point to execute the routine. OFFSET indicates at what location you want the binary file to load. If this is ommitted, the file is reloaded at the point it was saved from. If you use a variable for OFFSET, do not use one that starts with the letter 'R', as this will create either a syntax error or cause the file to be executed. Hopefully, if the file is reloaded at a different offset than saved and is executed, the ENTRY point also changes. So if a routine is relocatable then BSAVE/BLOAD can be used.

The 'DATA' save is just saving the routine as a serial file. START, LENGTH and ENTRY points are saved first so they can be recalled later. LENGTH is then used to reload the routine. While taking longer than the BLOCK method to save and reload, the DATA method retains each byte individually, enabling the ASC file produced to be manipulated easier than a binary file, which can't be changed easily.

The 'STRING' method is one used to save a routine of 255 bytes or less. The method creates a dummy string variable. The length of the MCR is dumped into VARPTR (dummy string) and the start of the MCR is copied from the DEFUSR0 pointer to VARPTR (dummy variable) +1 and +2. This points the string directly to the routine in question. It's then a simple matter to save and reload the string. Upon reloading, the dummy string's pointers are redirected once again to the right positon and the string is LSET to it so as not to change the position of the destination (x$=z$ will put the routine somewhere else). This is one way to save a short routine when it isn't sitting in a buffer.

If you've already put the routine into a normal string you can skip part of the process described above. Using such string can lead to trouble when you try to access it, because BASIC shuffles around strings during string operations to create space. This means that the routine has to be relocatable (see below) and before every access you need to determine the start of the routine by using:

```
        PEEK(VARPTR(dummy)+1) + PEEK(VARPTR(dummy)+2)
```

which gives the start of the string. This can also be used in literal strings within a basic program.

RELOCATABLE ROUTINES

Just what does the above title mean? A relocatable routine is one that can be used regardless of where it is dumped. Some routines aren't relocatable. For example, part of a routine is:

```
D500            100             ORG             D500H
D500   DB81     110 LOOP        IN              A,(81H)
D502   FE42     120             CP              42H
D504   C8       130             RET             Z
D505   C300D5   140             JP              LOOP
                150             END
```

- This subroutine reads data from port 81H, checks if it is 80H and if so returns to the calling routine, or else does the check again. Normally you assume an assembly to 0000H or some other location, such as D500H above to make assembly easier. The trouble is, the rootne as written above is not directly relocatable because of the last instruction. This jumps to D500H. If the routine has been dumped elsewhere then it won't work because the first time it's used, it will jump to whatever is at D500H and not to 'LOOP'. There are three ways around this.

First, you can make sure the routine always goes to the same place. This can be awkward if you don't know what the setup is or know something else is already there.

Secondly, you can rewrite the routine. Instead of using 'JP LOOP' you could use 'JR LOOP'. This assembles to:

```
D505   18F9        140            JR              LOOP
```

- which will work regardless of where the routine is. Relative jumps however only have a range of +/-127 from the current flow of the program.

Thirdly, you can arrange the loading and DATA statements to take into account where the program is, Assuming the original starts at D500H but that the load could be anywhere (you are using data statements to load the routine) you could do the following

```
      100 M1=&H##:M2=&H##:J1=M1+0:J2=M2+0:MS!=M1+M2*256:ML%=8
      110 FOR A=MS!TOTMS!+ML%-1:READ V$:V=VAL("&H"+V$)
      120 IF V$="XL" THEN V=J1
      130 IF V$="XH" THEN V=J2
      140 POKE A,V:NEXT
      150 DATA DB,81,FE,42,C8,C3,XL,XH
```

where M1 is the low value of the start byte, M2 is the high value, both in hexadecimal. Here when the appropriate DATA statement is read, the program substitutes the correct value. You will need to calculate the start of the program and the difference between start and where your jump is going to. This can be tedious however in this for you can correct it after discovering any errors.

The best way is plan ahead. Then, whatever method you use you still know what you are doing.

EMBEDDED CODE

It is possible to physically insert and MCR directly into a BASIC program. This requires the user to determine the location of a complete or part of a line. In Part 1, I demonstrated a technique of POKEing values into literal strings; in part 8 I showed a similar technique using the RESTORE statement. Rather than reiterate both here and now, I will assume you know just how to locate a suitable point using these techniques and poke in the routine. Just a reminder, a basic line consists of:

```
NLP:LN#: Statements:0
```

where NLP is the next line pointer (2 bytes), LN# is the line number (2 bytes) and 0 the terminating byte.

With literal strings you can't have a routine longer than 246 bytes because the line pointers + 'A$="' parts of the line take up 9 bytes (you don't need the terminating "). With a whole line, you have 249 bytes: the line pointers + 'REM' take up 6 bytes (REM in memory). You don't need the REM statement but if the line is ever executed you are bound to get a sytax error or worse.

In either case, you can't save the program in ASC format, alter the line by editing or get a decent listing. Also, you need to avoid zero bytes. These are used by BASIC to indicate the end of a line. If you have one in the routine the line ends prematurely and the next line is rubbish. It normally becomes a game for the user to develop routines without zero bytes in them - using 'XOR A' instead of 'LD A,0' for example. NOPs can't be used at all. The advantage of the technique is that the routine sits directly in the program Once in, you don't have to reload it and you can delete any data loading routines. When the program is saved, the routine is saved with it.

PROGRAMS

This part features only two programs. 'SAVE' should be typed in/loaded and saved to tape/disk in ASC format. It should be merged with previous programs in this series. 'MON' is a simple MONITOR program you can use to inspect/modify/save memory with. Instructons are included with the listing.

Note however that in modify mode, a simple ' will skip a byte and 'MESSAGE will put the message in memory.

next month, I conclude this discussion about Machine code routines and include two new routines for the experimental at heart.

MON

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 of article.

```
AP       10 REM MON
HK       30 WIDTH39:SCREENØ,Ø:LOCATEØ,Ø,1:CLEAR 1ØØØ:STOPON:ONSTOPGOSUB82Ø
HA       40 CS$=".+=_-DdGgMmSsLlhH":DEF FNA$(N)=STRING$(4-LEN(HEX$(N)),"Ø")+
            HEX$(N):DEF FNB$(N)=STRING$(2-LEN(HEX$(N)),"Ø")+HEX$(N)
BP       50 PRINT"MON VERSION 1.1":PRINT"DESIGNED BY ROBERT BRINKWORTH":PRIN
            T"Revised by L.A. Dunnning":PRINT"PRESS H FOR HELP"
BA      100 PRINTCHR$(212);
AP      110 LINE INPUTA$:A1$=LEFT$(A$,1):IFA$=""GOTO1ØØ
HO      120 ON INSTR(CS$,A1$)\2 GOTO 22Ø,14Ø,2ØØ,3ØØ,4ØØ,5ØØ,6ØØ,7ØØ,82Ø
CN      130 PRINTA$;" ?":PRINT:GOTO1ØØ
AB      140 AD=AD-16Ø:GOTO22Ø
GO      200 ' Display memory
DA      210 GOSUB81Ø
CI      220 FORJ=1 TO 1Ø:PRINTFNA$(AD);:FORI=AD TO AD+7:P=PEEK(I):PRINT":"FN
            B$(P);:NEXT
DF      230 PRINT"|";:FORR1=AD TO AD+7
IL      240 R2=PEEK(R1):IFR2<32ORR>127THENR2=46
EN      250 PRINTCHR$(R2);:NEXT R1:PRINT:AD=I
BP      260 NEXT J:GOTO1ØØ
AF      300 ' Go to specified address
CL      310 GOSUB81Ø:DEFUSRØ=AD:A=USRØ(Ø):GOTO1ØØ
CO      400 ' Modify memory
DC      410 GOSUB81Ø
BP      420 PRINTFNA$(AD)":";:P=PEEK(AD):R3=P:IFR3<32ORR3>127THENR3=46
FL      430 PRINTFNB$(P);"|";CHR$(R3);"|>";:LINEINPUTQ$
FP      440 IFQ$=""THEN1ØØELSEIFASC(Q$)=39THEN47Ø
EI      450 IFLEN(Q$)>2ORLEFT$(Q$,1)>"F"THENBEEP:GOTO42Ø
CM      460 MV=VAL("&H"+Q$):POKEAD,MV:AD=AD+1:GOTO42Ø
HF      470 K=LEN(Q$):IFK=1THENAD=AD+1:GOTO42ØELSEQ$=RIGHT$(Q$,K-1)
AE      480 FORJ=1TOK-1:POKEAD,ASC(MID$(Q$,J)):AD=AD+1:NEXT:GOTO42Ø
DB      500 ' SAVE
BI      510 CLS:PRINT"MEMORY SAVING SEQUENCE":GOSUB8ØØ:LINEINPUT"Filename >"
            ;IS$:GOSUB79Ø:GOSUB74Ø:IFTSGOTO51ØELSEFS$=IS$
EM      520 LOCATEØ,4:INPUT"Starting location:";IS$:GOSUB79Ø:IFLEN(IS$)>4GOT
            O52ØELSES1$=IS$
DP      530 LOCATE2,6:INPUT"Ending location:";IS$:GOSUB79Ø:IFLEN(IS$)>4GOTO5
            3ØE:ELSES2$=IS$
LN      540 IS$="":LOCATE3,8:INPUT"Entry Point    :";IS$:GOSUB79Ø:IFLEN(IS$)>
            4GOTO54ØELSES3$=IS$
FF      550 S1=VAL("&h"+S1$):S2=VAL("&h"+S2$):S3=VAL("&h"+S3$):S1$=FNA$(S1):
            S2$=FNA$(S2):IFS1$>=S2$GOTO52Ø
BJ      560 IFS3$=""THENS3=S1
BG      570 LOCATEØ,12:BSAVE FS$,S1,S2,S3:GOTO1ØØ
AJ      600 'LOAD
IE      610 CLS:PRINT"MEMORY LOADING SEQUENCE":GOSUB8ØØ:LINEINPUT"Filename >
            ";IS$:GOSUB79Ø:GOSUB74Ø:IFTSGOTO61ØELSEFS$=IS$
IK      620 IS$="":LOCATEØ,5:INPUT"Offset              ";IS$:GOSUB79Ø:IFLEN(IS$)>4G
            OTO62ØELSE:OS%=VAL("&h"+IS$)
AO      630 LOCATEØ,7:E$="":INPUT"Execute code ";E$::E$=LEFT$(E$,1):GOSUB79Ø
BF      640 R1=1-(INSTR(".YyRr",E$)>1)-(IS$<>"")*2:LOCATEØ,1Ø:ONR1GOSUB65Ø,6
            6Ø,67Ø,68Ø:GOTO1ØØ
```

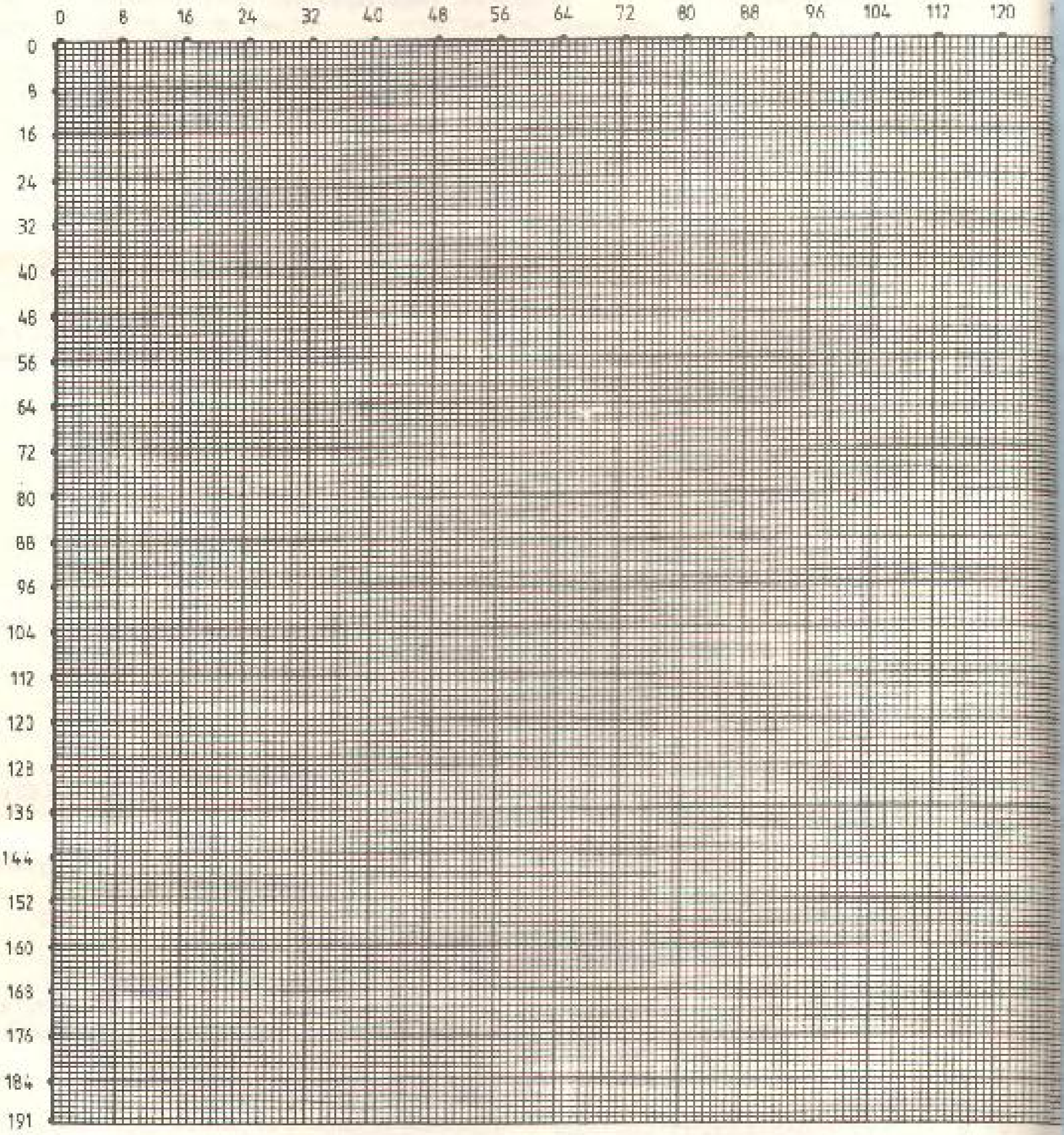
```
CK      650 BLOAD FS$:RETURN
AF      660 BLOAD FS$,R:RETURN
DH      670 BLOAD FS$,OS%:RETURN
BC      680 BLOAD FS$,R,OS%:RETURN
LJ      700 CLS:PRINT"H/h       Produces Instructions":PRINT:PRINT"D/d #### G
            ives Dump of Address":PRINT"=/+       Gives dump of next segment"
            :PRINT"-/_       Gives dump of previous segment"
MN      710 PRINT"M/m #### Modifies Memory Address"
HN      720 PRINT"G/g #### Execute MCR at Address":PRINT"S/s       Initiate S
            AVE routine":PRINT"L/l       Initiate LOAD routine":PRINT"^STOP
              Exit Program"
BK      730 PRINTSTRING$(39,172);:GOTO1ØØ
DK      740 TS=Ø:K=INSTR(IS$,":"):IFK=ØORK=LEN(IS$)GOTC78Ø
EE      750 F1$=LEFT$(IS$,K):IFINSTR("1:2:CAS:cas:",F1$)=ØGOTO78Ø
GE      760 F2$=MID$(IS$,K+1):IFF2$<"A"GOTO78Ø
CA      770 RETURN
AI      780 TS=1:RETURN
GF      790 IFIS$="."THENRETURN1ØØELSERETURN
KC      800 PRINT"Enter '.' as an input to exit from      sequence":RETURN
DL      810 IFLEN(A$)>6THENRETURN13ØELSEAD=VAL("&H"+MID$(A$,3,4)):RETURN
HP      820 SCREEN,1:PRINT"MON Exited"
END
```

SAVE

by : L.A. Dunning

```
AE      300 REM SAVE
IG      305 REM Demstrates saving & loading of          MCRs
AI      310 LINEINPUT"Input DEV:Filename >";FS$
CD      315 DIM X$
DF      320 PRINT"1: Block save":PRINT"2: Data Save":PRINT"3: String Save -
            MCRs of 255 bytes or      less"
CI      325 INPUTST:IFST<1ORST>3THEN325
CP      330 ON ST GOSUB 335,345,355:GOTO385
EI      335 REM block
DM      340 BSAVE FS$,MS!,MS!+ML%-1,EP!:RETURN
BA      345 REM data
CG      350 GOSUB37Ø:FORA=MS!TOMS!+MS!+ML%-1:V=PEEK(A):PRINT#1,V;:NEXT:CLOSE
            :RETURN
JI      355 GOSUB37Ø:DEFUSRØ=MS!:GOSUB36Ø:PRINT#1,X$:CLOSE:RETURN
CB      360 J=VARPTR(X$):POKEJ,ML%:POKEJ+1,PEEK(&HF52B):POKEJ+2,PEEK(&HF52C)
            :RETURN
CC      365 RETURN
DF      370 OPENFS$FOROUTPUTAS#1:PRINT#1,MS!;ML%;EP!:RETURN
DM      375 OPENFS$FORINPUTAS#1:INPUT#1,MS!,ML%,EP!:RETURN
BL      380 RETURN
CK      385 PRINT"Save Complete":PRINT"Press Key to continue":A$=INPUT$(1)
IL      390 IFINSTR(FS$,":")=ØORLEFT$(FS$,4)="CAS:"ORLEFT$(FS$,4)="cas:"THEN
            PRINT"REWIND TAPE":PRINT"Press key when ready":A$=INPUT$(1)
HN      395 PRINT"Now Reloading MCR":ONSTGOSUB4ØØ,42Ø,43Ø:GOTO 445
EH      400 REM block
DD      405 BLOAD FS$,MS!:RETURN
AB      410 REM BLOAD FS$ will reload where                 saved
DO      415 REM BLOAD FS$,R will execute from               entry point EP!
BC      420 REM data
LO      425 GOSUB 375:FORA=MS!TOMS!+ML%-1:INPUT#1,V:POKEA,V:NEXT:CLOSE:RETUR
            N
BK      430 REM string
KE      435 GOSUB 375:DEFUSRØ=MS!:GOSUB36Ø:INPUT#1,Z$:LSETX$=Z$:CLOSE:RETURN
CA      440 RETURN
CF      445 REM
END
```

SCREEN 1


0
8
16
24
32
40
48
56
64
72
80
88
96
104
112
120
0
8
16
24
32
40
48
56
64
72
80
88
96
104
112
120
128
136
144
152
160
168
176
184
191

MSX BASIC & YAMAHA CX5M (REVIEW)

By P. W. Deckert.

It's 5pm on Saturday and my wife wakes me with a cup of coffee (I'm working nightshift). I ask "What's new?" and she replies "Steve Yarrow from Rose Music, dropped an MSX Computer off for you this morning."
Before my poor wife could utter another word, I had jumped out of bed and raced down the stairs.

There I was, standing in the middle of the lounge, wearing nothing but a pair of underpants and a big smile. Thank God we didn't have any visitors. Back up the stairs and into the Computer room, where I pushed my poor SV-328 out of the way and placed the MSX Computer, a Yamaha CX5M, on the table. I plugged in the Transformer that came with it and moved the Colour Monitor into place. Would you believe it, the cables I use to plug the SV into the monitor wouldn't fit the Yamaha. Back down the stairs and into the hall cupboard. Let it be known that finding anything in this cupboard is an art in itself and usually requires the removal of many boxes. But this time I was in luck, the video dubbing cables I was searching for were near the top and only required the removal of two boxes. Up the stairs again, thank God it's only one floor. I connected computer to monitor with the patch cords and was ready for the big moment. I turned on and to my pleasure it all fired up. The screen looked much the same as the SV except for the substitution of the word Spectravideo with the word MSX.

For example where Spectravideo begins with:

SV extended BASIC version 1.1

Yamaha M.S.X. begins with:

M.S.X. extended BASIC version 1.0

The user definable key prompts on the bottom line are not in inverse video but did have exactly the same commands as Spectravideo.

Well the only way to try out a computer is to write a program. So I thought something simple to start with might be in order.

```
5 SCREEN 1
10 LINE (0,0)-(100,100)
20 GOTO 20
RUN

SYNTAX ERROR IN LINE 10
```

Well I thought it too good to be true. So I fell back on the old philosophy, which says, when all else fails read the manual. So I looked inside the box the Yamaha came in and found not one but two manuals.

1) The Yamaha Music Computer CX5M Owner's Manual.
2) Yamaha M.S.X. Reference Manual.

The first shows how to put the computer together and get started. The second is a thick 354 page manual that tells you all you want to know about the M.S.X. Basic. Why doesn't Spectravideo have something like this.

Well back to my little program, I looked up the LINE command in the reference manual and as far as I could make out all was OK?? It was now about 30 minutes since I woke up and I was still wearing nothing but my underpants and a slightly less vigorous smile. I decided to turn off the computer, have a shower, get dressed and find something to eat.

It was in the shower that it hit me, not the soap, (I do some of my best thinking in the shower). The syntax error in line 10 was not caused by the LINE command, but by the SCREEN command in line 5. I suddenly remembered that our SCREEN 1 is their SCREEN 2. At least it nearly is.

Later that evening I finally got back to the computer and tried my little program using SCREEN 2 instead of SCREEN 1. Hey presto!! It worked. After a few more LINEs, some BOXES, a CUBE and three CIRCLES, I decided that it worked much the same as our Spectravideo.

In fact apart from a few differences the Yamaha M.S.X. Basic and Spectravideo Basic are in the same family.

Some of the differences I found were:

1) The SCREEN command has five parameters
SCREEN [mode][,sprite size][,key click][,tape speed][,printer]

Mode: four screen modes instead of three
Sprite size: same as SV
Key click: used instead of SV command CLICK ON/OFF
Tape speed: 1200 or 2400 baud. SV set to 1800 baud
Printer: can it handle M.S.X. graphics

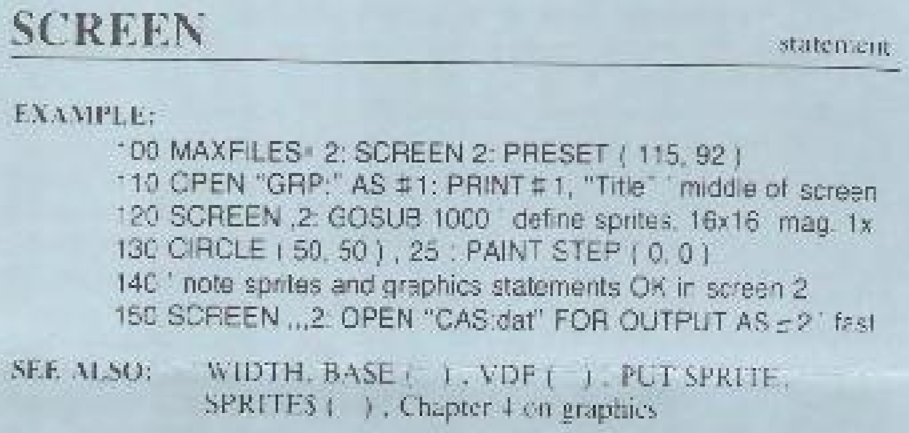
2) M.S.X. Basic has a CALL command that allows it to access other ROM routines or cartridges. EG. in the Yamaha, CALL MUSIC will access a program that handles the FM Music Synthesizer.

3) M.S.X. Basic lets you at the VDP Registers and the Pattern, Sprite and Colour Tables by giving you two special variables, VDP and BASE.

4) Would you believe it Spectravideo Basic has something M.S.X. Basic doesn't. The GET and PUT commands used with graphics in the Spectravideo are not present in M.S.X. Basic.

But all in all the similarities far outweigh the differences. Also I noticed the bug in the DRAW command has been fixed.

Before I look at the Yamaha as a computer, there is one thing I would like to show you. The program below comes from the M.S.X. Reference Manual. It doesn't work!! So the SV Manual is not the only one with errors.



SCREEN

statement

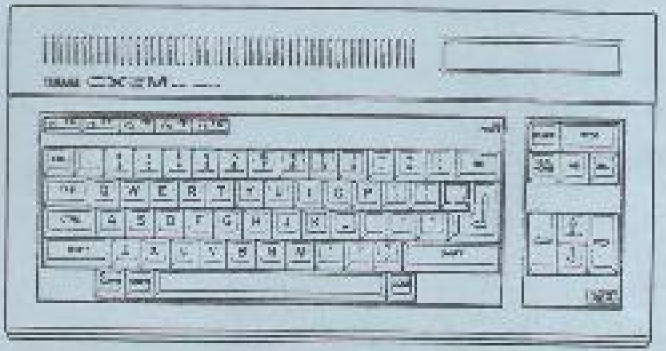
EXAMPLE:

```
100 MAXFILES= 2: SCREEN 2: PRESET ( 115, 92 )
110 OPEN "GRP:" AS #1: PRINT #1, "Title" ' middle of screen
120 SCREEN ,2: GOSUB 1000 ' define sprites: 16x16  mag. 1x
130 CIRCLE ( 50, 50 ) , 25 : PAINT STEP ( 0, 0 )
140 ' note sprites and graphics statements OK in screen 2
150 SCREEN ,,,2: OPEN "CAS:dat" FOR OUTPUT AS #2 ' fast
```

SEE ALSO: WIDTH, BASE ( ), VDP ( ), PUT SPRITE, SPRITES ( ), Chapter 4 on graphics

It's now Sunday night and I am taking time to have a good look at the Yamaha and the way it is presented.

You may be wondering what I am doing with a Yamaha Computer anyway. Well I have been pestering Steve Yarrow of Rose Music for some time to get me an M.S.X. Computer to look at. Spectravideo have produced two M.S.X. computers. The SV-728 and the SV-EXPRESS but these are still not available in Australia. Luckily Rose Music also market another computer, the Yamaha M.S.X. Music Computer. So thanks very much to Steve Yarrow who went out of his way to get the computer to me so I could have a play with it and review M.S.X.



The Yamaha is much more than your M.S.X Computer, it is a full power musical instrument. As it also has an FM Sound Synthesizer unit built in. This unit is an eight note polyphonic synthesizer, capable of producing eight notes simultaneously. It also has 46 preprogrammed voices.

The whole atmosphere of the owner's manual is music not programming. This is enhanced by the two M.S.X. cartridges that came with the computer. The FM music composer and the Yamaha FM music macro.

To my sons disgust, Steve didn't get the music keyboard that plugs into the computer and allows you to play the synthesizer. So as far as that part goes we couldn't get any sound out of the FM synthesizer. I did plug both cartridges in, however and they seem to work, but no music.

The computer is basic black, black case with black keys and white letters. The keyboard had a different feel to the SV with the keys having a tendency to rock under my fingers. I tended to drop a lot of letters while typing and had to slow down. I guess one would get used to it with time. A normal cassette player can be plugged into the Yamaha for loading and saving of programs and files. I didn't bother hooking one up as it required making up a cable to hook it together. I did scare myself while playing with the computer, when I typed MOTOR ON and heard a loud CLUNK. The Yamaha uses a relay as motor control. There is a centronics plug at the back of the computer for a printer, also RF, Video and Audio sockets for a TV or monitor. Joystick ports look the same and are on the same side as the Spectravideo.

One thing I noticed in the owner's manual is that there is no mention of Disk Drives but as there is an expansion slot I guess any M.S.X. Disk Drive will plug in??

Well that's about all I can think of to say about the Yamaha. It sure is a nice computer and I'm only sorry I didn't get a chance to try out the synthesizer.

DOUBLE SIDED DISK DRIVES

382K DISK DRIVES ARE AVAILABLE FOR USE WITH THE SV601 EXPANDER AND THE SV 801 CONTROLLER. THE DISK DRIVES COME WITH A MODIFIED BIOS THAT ALLOWS READING OF 15 OTHER DISK FORMATS.

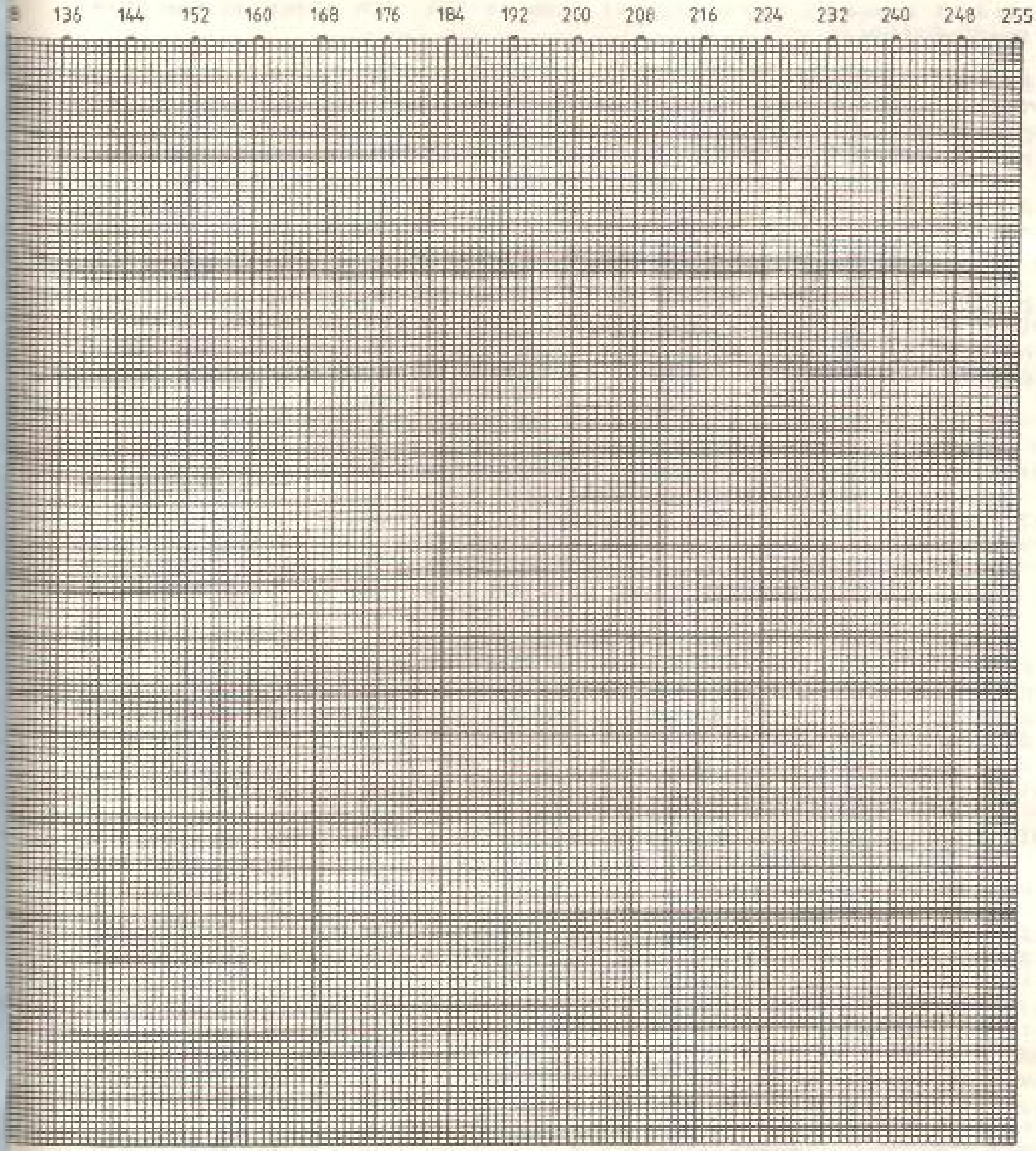
Costs:	1 DRIVE	$450	($500 non club members)
	2 DRIVES	$850	($900 non club members)

If you already own diskdrives a modified BIOS is also available.

Cost:	S/S BIOS $25	($35 non club members)

For further information call:

S.W. McNamee,
5/15 Stuckey Rd., CALL: 07-8358683 (working hours)
Clayfield 4011.


136
144
152
160
168
176
184
192
200
208
216
224
232
240
248
255

UNDERSTANDING CP/M Pt-2

By. S.W. McNamee.

As promised in part one, I turn this time to a discussion of the CPR resident commands and a look at some of the .COM files on the CP/M distribution disk.

CPR RESIDENT COMMANDS

These commands are called resident because they are built into the operating system and are executed immediately, without having to be first loaded from disk. A detailed discussion of each follows, but first a quick look at Wildcard filespecs. Whenever a filespec is called for in CP/M it can always be the normal unambiguous form mentioned in the previous article or it can often be an ambiguous form using combinations of two "Wildcard characters". These are "*" and "?". A "*" in any position within a filespec will match any letter in that position and all positions following in the field. A "?" in any position will match any character in that position only. When a command that recognises ambiguous filespecs sees these characters in a filespec it searchs for all files that match the wild card characters. Eg:

.	- matches all files on the disk.
*.COM	- matches all files with a .COM file type.
MODEM*.*	- matches all files with the first letters in the file name of "MODEM".
????????.???	- also matches all files on the disk.
ST????AB.*	- matches all files with the first two letters of the file name being "ST" and the last two being "AB".
*.??C	- matches all files with the last character of the file type being "C".

LOGGED DISK The first command is the one used to change the currently logged in drive. It has no name and is invoked by simply typing the letter of the desired drive followed by a colon and <CR>. CP/M drives are designated A: to P: but most systems will only have 1 or 2 drives A: and B:. The CPR prompt will change to the letter of the selected drive and all activity will default to the selected drive.

DIR This command allows you to check which files are on the disk. It can be used with ambiguous filespecs to search for any combinations of files desired. Eg:

DIR<CR>	- all files on the current disk.
DIR B:<CR>	- all files on the selected drive (in this case the B: drive).
DIR *.DOC<CR>	- all files with a filetype of .DOC.

ERA This command lets you erase a specified set of files from a disk. It accepts wild card filespecs and if "*.*" is specified will ask you if you are sure you want to erase everything. Eg:

ERA STAT.COM<CR>	- erase a single file.
ERA A:*.DOC<CR>	- erase all .DOC files from the specified drive A:.

REN This command allows you to rename a single file. It does not accept wild card filespecs. The form is :

REN newfile=oldfile<CR>. Eg:

```
REN SW.COM=SWEEP.COM<CR>
                  - renames SWEEP.COM to SW.COM on the current drive.
REN B:EDIT.COM=ED.COM<CR>
                  - renames ED.COM to EDIT.COM on the selected drive (B:).
```

SAVE Allows you to save sections of memory as a named disk file. The format is SAVE n filespec<CR> where n is the number of "pages" of memory to save. A page of memory is 256 bytes or one quarter of a "K". The memory is always saved starting at 100H. The H stands for Hexadecimal which means base 16. 100H = 256 decimal. Eg:

```
SAVE 4 B:TEST.COM<CR>
                         - saves 4 pages of memory to the file named and
                           the disk specified.
```

A short digression into the concept of Booting is in order at this point.

When you first turn on your computer with the CP/M master disk in the drive and get the sign on message you have done what is termed in the jargon as a "Cold Boot". This procedure starts from scratch and loads the whole system into the computer and initilises everything to a "Startup" status. Whenever most programmes have finished executing they do what is known as a "Warm Boot". This reloads the CPR and BDOS into memory and resets the disk system but does not change anything else. You can do a warm boot any time you like by simply pressing ^C immediately following the system prompt. (A ^ preceding any character means press that key while holding down the control key). When either a cold or warm boot is done CP/M builds a directory map of each drive that is logged in. It then checks the map each time you try to write to the drive and if the disk has been changed will abort with the error - Bdos Err on .. R/O . This is to protect against accidentally writing on the wrong disk. To avoid this happening you must do a warm boot each time you change a disk that you want to write to; this includes renaming, erasing or saving files.

TYPE Simply types out the file named on the con: (console) device-usually the TV screen. TYPE does not recognise wildcards and is rather liberal with what it will "type". It will happily try to type a '.COM' file with weird results - try it and see!

USER This command is one that is seldom used in a hobby environment. Syntax is USER n<CR> where n is a number from 0 to 15. User areas are a means of keeping unrelated files in a different location on the same disk. All files stored under a particular user area can only be accessed, used and changed from within that user area. This command is really only of use to systems with more than one user and/or with large capacity drives.

We come now to a short discussion of some of the utility programmes on the CP/M distribution disk, but first a quick look at the CP/M "Command Line". Any file with an extension of .COM (short for COMmand) can be loaded and run by typing its' name (without the .COM extension) followed by <CR>, after the CP/M prompt. Most application programmes will also expect one or more parameters to be passed to it from the

command line. These parameters can be of many forms but usually consist of one or two filespecs and/or option characters. Each parameter must be separated from the command name or the previous parameter by one or more spaces.
Eg.
    STAT DSK:<CR> will print out the characteristics of all disk drives that have been logged in since the last warm boot.

Some programmes will require a character other than a space to separate parameters. For example PIP uses the "=" sign to separate its' filespecs.
Eg. PIP filespec1=filespec2 copies filespec2 to filespec1.

STAT.COM This is the routine that allows you to check on various statistics related to the disk system and to change various parameters of files, disks and logical devices. You can set individual files or whole disks to "Read Only" status or make changes in the assignments of Logical to Physical devices. (More will be said on this in a later article). You can also get a listing of selected files and their characteristics. Full instructions appear in the CP/M manual.

PIP.COM This programme is used to copy files from one disk to another or to copy from or send files to logical devices. Its' biggest disadvantage is that it cannot copy files from one disk to another unless you have two or more drives. All is not lost however as the SAUG has a couple of programmes in its' library that allow single drive file copies, and the good news is that they are in the "Public Domain", which means that they are freely able to be copied. One in particular that springs to mind is called MFT.COM .

SYSGEN.COM is the utility that allows you to copy the system tracks from one disk to another. Its' use is self explanatory however one useful aspect may not be general knowledge. SYSGEN can also be used to copy the system tracks of a BASIC disk. To do this when the programme asks for the source disk just remove the CP/M disk and replace it with a BASIC system disk and so on. This method is at least 10 times faster than the BASIC programme that does the same thing.

ED.COM is the standard CP/M context editor. It allows you to create ascii files and edit them. Eg. letters, source files for a language translator school essays etc. I feel however that the only thing that ED is good for is to show how not to do it! A much better proposition is to get hold of a good full screen editor and use that. Since most commercial editors (Eg. WORDSTAR or PERFECT WRITER) are exorbitantly priced - WORDSTAR is about $400.00 - this can be a problem. There are cheaper editors around but I feel the best place to look is in the public domain. The SAUG should shortly have a quite reasonable screen editor in its' library and I strongly reccomend that you get hold of a copy.

The final part of this installment concerns applications software that can be run under CP/M.

The CP/M distribution disk by itself doesn't really do very much. You can create and edit text files using ED (if you are a masochist) or you can write programmes in 8080 assembly language, using ED to write the programme and ASM.COM to assemble it. Since most people are loath to touch assembly language and those that do would much rather use a good Z80 Macro assembler, this leaves the standard distribution disk pretty well useless.

All is not lost however. The list of applications programmes available to run under CP/M numbers literally in the several thousands. A lot of these are hideously expensive commercial programmes but the majority of them are on volumes from various user groups and the good news is that all of these are in the Public Domain. This means that they can be copied freely and no royalties need be paid for this privilege. Although it costs nothing to copy these programmes one first must find someone who has them. Some commercial software dealers maintain a library of public domain volumes but the catch is they will usually charge anywhere from $10.00 to $30.00 to copy each volume. (A volume is a standard 8" single density single sided floppy disk - about 245k bytes). I feel that this is a bit of a rip off but there is not much that can be done about it.

Alternatively many user groups maintain a library of Public Domain software and will generally charge only a nominal amount for copying. Our own SAUG has a small but continually growing library of CP/M utilities and this is a good place to start, - my apologies to the librarian if this article generates a stack of work for him. Another alternative is to join a user group catering excusively to CP/M users. These groups usually have fairly large libraries of software and you should find what you are after there. For people living in Brisbane a group called the "Queensland CP/M Users" meets at 12:30 PM on the last Sunday of each month at the Civil Engineering Building, Queensland University, St Lucia. Members in other cities will have to find their own local group. If you know of any in your area perhaps you could let the editor know so he can pass the information on.

The last option is to obtain an RS-232 card and a modem. This will allow you to access any of the numerous RCPM systems scattered round the country. RCPM stands for Remote CP/M. Then, using one of the available communications programmes, (YAM and MODEM7 are both available from the SAUG library) you can download software from the remote system to you own computer. Two problems become apparent with this method. The cost of STD calls lasting several hours can quickly force you to sell your computer to pay the phone bill and the time taken to transfer files at 300 baud can be very frustrating after a while. Still it can be heaps of fun. In the next news letter I shall be publishing the circuit diagram for an RS-232 card that I have designed, if any of you adventurous electronics types want to build yourselves one.

In the next installment I will go into some of the more technical details of the operating system and include a short discussion on assemblers and languages.

BRISBANE USERS MEETING

S.V. Users meetings held at 26 Gable Street, Stafford Heights. Held on the 1st Tuesday of each month. Please note correction to last months address. (sorry E.D.)

CASSETTE LABEL

by T. Davies.

Recently I typed in a Label making program on my schools computer. I was so pleased with the result that I decided to adapt the program for my Spectravideo. This was quite a hard task to attempt for me and seeing the SV doesn't have a 'lprint at' function for vertical tabbing, after several hours of work, I gave up. Then when I was rereading one of my Rainbow magazines I saw a program for a cassette label program of a very similar to the one I had tried to convert before. As the CoCo has the closest Basic to the SV I know of (except for the IBM PC) I decided to try to adapt it. After about an hour of debugging and mucking around with it I eventually had it running exactly as I wanted it. I modified a few features like the fact that the title was upside-down and wasn't even close to being centred and eventually had it running perfectly. Here is the listing, it isn't much but for my standards it is a great acomplishment.

```
----------------------------------------
SIDE A             :SIDE B
----------------------------------------
BORN IN THE USA    :NO SURRENDER
COVER ME           :BOBBY JEAN
DARLINGTON COUNTY  :I'M GOIN DOWN
WORKING ON THE     :GLORY DAYS
.HIGHWAY           :DANCING IN THE DARK
DOWN BOUND TRAIN   :MY HOMETOWN
I'M ON FIRE        :
                   :
                   :
                   :
                   :
                   :
----------------------------------------

   BORN  IN  THE  USA

----------------------------------------





----------------------------------------
```

```
10 CLEAR600:DIMA$(12),B$(12):CLS
20 INPUT"TITLE MAX 20 CHARS";GP$:IFLEN(GP$)>20THENGP$=LEFT$(GP$,20)
30 CLS:LOCATE10,0:PRINT"SIDE A"
40 FORI=1TO12
50 PRINTI;:INPUTA$(I)
60 IFA$(I)=""THEN70ELSENEXTI
70 CLS:LOCATE10,0:PRINT"SIDE B"
80 FORI=1TO12
90 PRINTI;:INPUTB$(I)
100 IFB$(I)=""THEN110ELSENEXTI
110 CLS:LOCATE10,0:PRINTGP$:LOCATE42,0:PRINTAN$:PRINT:PRINT"SIDE A","SIDE B"
120 FORI=1TO12
130 IFLEN(A$(I))>15THENPA$=LEFT$(A$(I),15)ELSEPA$=A$(I)
140 IFLEN(B$(I))>15THENPB$=LEFT$(B$(I),15)ELSEPB$=B$(I)
150 PRINTPA$,PB$:NEXTI
160 INPUT" PRESS ENTER TO PRINT A LABEL";N$
170 LPRINTSTRING$(40,"-")
180 LPRINT"SIDE A";TAB(19);":SIDE B":LPRINTSTRING$(40,"-")
190 FORI=1TO12:LPRINTA$(I);TAB(19);":";B$(I):NEXTI
200 LPRINTSTRING$(40,"_"):LPRINT:LPRINTCHR$(14);TAB(INT((20-LEN(GP$))/2));GP$
210 LPRINTSTRING$(40,"_"):FORI=1TO5:LPRINT:NEXTI
220 LPRINTSTRING$(40,"-")
```

CLARKE ELECTRONICS

55-57 Belair Road, Kingswood, S.A. 5062
Telephone (08) 271 1218

***** SPECIAL OFFER TO CLUB MEMBERS *****

	S.A.U.G. Price	Retail
DSDD 5.25" DISKS (FOR TEN)	28.00	39.00
DSQD 5.25" DISKS (FOR TEN)	47.30	55.40
PAPER TAMER (creates a tidy environment for computer printouts)	37.00	46.95
CENTRONICS PLUGS IDC.	9.95	16.75
CENTRONICS CABLE	27.50	39.50
PRINTERS SCM-1000B (2 only left at old pre-devaluation price)	345.00	481.00
VERBATIM HEAD CLEANING KIT	9.95	12.50
SVI 80 COLUMN CARD	229.00	249.00
ABC SWIVEL BASE VDU STAND	29.00	35.00
HIGH RES MITSUBISHI AMBER VDU 20MHz	199.00	232.50

TIP OF THE MONTH **CLARKE ELECTRONICS April 85**

Following last months comments on the PLAY command, I discovered **another form of the reserved word PLAY.** It seems that the standard play **command fills up a string** buffer with up to 25 notes for each channel. The **buffers can be interrogated** by using PLAY(C) where C is the channel number **being tested.** When the channel is empty the function PLAY(C) returns 0 **otherwise** -1. The following examples show the various syntax arrangements.

```
A=PLAY(1)       or      IF (PLAY(1))=0 THEN GOSUB 1000

FOR Z = 1 TO 3:IF (PLAY(Z))=0 THEN GOSUB 1000:NEXT Z

100 CLS
110 FOR I=1TO 2 STEP 0  'Loop forever
120 '
130 'If play queue empty gosub and refill
140 IF (PLAY(1))=0 THEN GOSUB 190
150 LOCATE 20,12:PRINT TIME        'This line to show CPU is busy
160 NEXT I
170 '
180 'Play subroutine called only when play queue empty
190 PLAY"s9m3000t255o4co3bagfedc":RETURN
```

Best Regards Eric S. Clarke Eric Clarke.

buy, trade & sell

FOR SALE

SV-328, 2 JOY STICKS,
CASSETTE & LOTS OF
SOFTWARE.

$390

CONTACT: S. W. McNAMEE
CALL: (07) 8358683
WORKING HOURS

OR:S.A.U.G.
(003) 312648

FOR SALE

1. JUST WRITE JNR.
2. FLIPPER SLIPPER.
3. FRANTIC FREDDY.
4. CROSS FORCE.
5. SECTOR ALPHA.
6. MUSIC MENTOR.

$25.00 EACH

H. ZUYDAM,
P.O. BOX 285,
FAIRFIELD,
N.S.W. 2165.

1983/84 YEARBOOK $15.00

WANTED

INFORMATION ON USE OF
SPECTRAFILER.

CONTACT:
V. RAUTER,
14 ADELAIDE ST,
GEORGE TOWN,
TAS. 7253.
CALL:(003) 822556

FOR SALE

SV-603 GAMES ADAPTER +
QUICKSHOT III, ZAXXON &
TURBO GAMES. WORTH $300
WILL SELL FOR $160.

B. MARKEY,
19 SHIEL STREET,
TOOWOOMBA,
QLD. 4350.

CALL:(076) 325148