

# SVI & MSX

SPECTRAVIDEO



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

ISSUE NO.

2 - 10

ANNUAL SUBSCRIPTION

AUSTRALIA .....	\$20.00
OVERSEAS .....	\$25.00
OVERSEAS AIRMAIL ...	\$30.00
YEAR BOOK 83/84 ....	\$20.00

DATE

JULY - 85

## CONTENTS

INTRODUCTION .....	2
EXPLORING BASIC PT-13 .....	3
MOVE BASIC TO RAM .....	8
CLARKE ELECTRONICS .....	9
UNDERSTANDING CP/M PT-3 .....	10
RS-232 INTERFACE .....	12
POPCORN (PROGRAM) .....	14
FORREST DATA SERVICES .....	16
DOUBLE SIDED DISK DRIVES .....	17
PROGRAM CHANGES "INPUT" .....	18
SOUND WITH A MONITOR .....	18
BUY, TRADE & SELL .....	19
BUY, TRADE & SELL .....	20

## 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***By. The Editor.*

We have another mixed bag of goodies this month to tempt you and amaze you. But first I have a very important problem I must talk to you about. That is the 1983/84 Year book. When we first printed the book, some months ago now, we priced the book according to the cost of printing. Thus for \$15 you all got a great bargain. Well the first 100 copies are now all gone, so we had a reprint. This is were the problem starts. I received the bill for the printing and if we continue to sell the Year Book for \$15 we will make a very large loss. As the group is totally non profit our bank balance cannot stand any kind of loss. We have honored the \$15 price for the last 2 weeks, but as of now the price will be \$20 and from now on any \$15 cheques will get a bill for \$5. sorry about that but we must break even.

```
*****  
*          *  
* 1983/84  YEAR BOOK  *  
*          *  
* NEW COST    $20.00  *  
*          *  
*****
```

Back to this newsletter.

**EXPLORING BASIC PT-13**

As usual Mr. Dunning is up to his usual programming magic.

**MOVING BASIC TO RAM.**

A great program for adventurous patchers.

**CLARKE ELECTRONICS**

Tip of the Month

**UNDERSTANDING CP/M PT-3**

For all you CP/M addicts.

**NOTE:** This article contains the design of an RS-232 interface for use with a modem. Don't call the Users group for information on the Circuit. please pass all requests to Mr. S. McNamee. Also a good Knowledge of Electronics is a must with this project.

**RS-232 INTERFACE LIST & CIRCUIT****POPCORN (PROGRAM)**

Another music program. O.K. for M.S.X.

**FOREST DATA SERVICES**

More addons available.

**DOUBLE SIDED DISK DRIVES** A repeat of an earlier article due to the interest generated.**PROGRAM CHANGES "INPUT"**

Another useful patch for INPUT.

**SOUND WITH A MONITOR**

A must for you 80 Column card owners.

**Exploring Basic Pt-13**

by L.A. Dunning

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

**ROM CALLS**

If you are programming in machine code you may wish to make use of the code already existing in the Spectravideo ROM. The advantage of doing this is twofold: you don't have to re-invent the wheel and you save memory since the routine is already written. Short of burning in a new ROM you can't change the ROM. You can cover it up by bank switching but the routines are still there! The disadvantages of using ROM calls are also twofold: you must pre and post package the data used by the routine so it will do what you want, and if the routines are changed by the manufacturer (as some are wont to do) your MCR will crash.

To use a ROM call effectively you need to know three things-

- A) What data is required
- B) What the routine does
- C) What data is produced

The data required is the information that is needed initially to make the routine work. This could be simple data in a memory location, data in Z80 registers, external conditions or something else. Knowing what a routine does is a bit trickier. In most cases, you need to know the function of a routine (or the functions of related routines) before you can see how it does it. A classic example of this on the SV is the use of characters 28 to 31. These are used to prefix constants and other variable but they are also used as cursor control codes. Thus, a routine that detects such a value might be related to an arithmetic routine, or it might deal with graphics!

What data is produced is vital knowledge if you want to get values out of the system. Also, it may be that a ROM routine might be useless because the state it returns from destroys vital data or jumps to another ROM routine. In this case you write your own routine. Another thing about ROM routines is that they are good demonstrations of how machine code is used by professional programmers. Discovering why the designer did something one way instead of another can be just as much fun as getting it to work.

**BASIC COMMANDS**

The following list indicates the starting or main entry points to major ROM routines based on BASIC commands. It does not give an explanation of how they work nor does it indicate how they should be entered or exited. Routines marked with '#' are not support directly in ROM and must be modified to work properly. Disk basic does this when booting up. Also, note that some commands are missing from the list. This is because they are "composite" statements. INTERVAL in memory is really INT:e:r:VAL and so its start is in the INT routine.

I lack the space and knowledge to give you detailed instructions on each routine. The best method is to use your disassembler to look at each routine, noting down points of information.

ADDR	COMMAND	ADDR	COMMAND	ADDR	COMMAND
66CF	END	0D65	FOR	6821	NEXT
109B	DATA	13D2	INPUT	6061	DIM
1405	READ	10C0	LET	1028	GOTO
OFE2	RUN	1225	IF	66AE	RESTORE
OFF6	GOSUB	1061	RETURN	109D	REM
66C8	STOP	1265	PRINT	67A6	CLEAR
1AB8	LIST	6556	NEW	1124	ON
1A52	WAIT	188A	DEF	1CAD	POKE
671B	CONT	1E15	CSAVE	1EAA	CLOAD
1A4C	OUT	125D	LPRINT	1AB3	LLIST
3777	CLS	1A6C	WIDTH	109D	ELSE
672F	TRON	6730	TROFF	3567	SWAP
676E	ERASE	11EA	ERROR	119D	RESUME
1C6C	DELETE	11F5	AUTO	1CF6	RENUM
OF5C	DEFSTR	0F5F	DEFINT	0F62	DEFSNG
OF65	DEFDBL	1374	LINE	7080	OPEN
6B66	LEFT\$	6B96	RIGHT\$	6B9F	MID\$
55C6	SGN	57F8	INT	55B1	ABS
5222	SQR	5306	RND	50D1	SIN
5197	LOG	526B	EXP	50B8	COS
5120	TAN	5139	ATN	6CF7	FRE
1A37	INP	1839	POS	6B04	LEN
6909	STR\$	6BC0	VAL	6B10	ASC
6B20	CHR\$	1CA6	PEEK	46F2	VPEEK
6B4D	SPACE\$	68FA	OCT\$	68FF	HEX\$
1834	LPOS	6904	BIN\$	56B5	CINT
56DD	CSNG	5765	CDBL	57E9	FIX
3206	STICK	3263	STRIG	3280	PDL
32BD	PAD	34C9	DSKF #	74C6	FPOS
732B	CVI	732E	CVS	7331	CVD
74B0	EOF	7484	LOC	749A	LOF
7312	MKI\$	7315	MKS\$	7318	MKD\$
72CD	FIELD	2FB4	GET	2FB1	PUT
7375	CLOSE	7121	LOAD	7122	MERGE
73B2	FILES #	7228	LSET	7227	RSET
7167	SAVE	73AD	LFILES #	2652	CIRCLE
4552	COLOR	29DA	DRAW	24FC	PAINT
40BE	BEEP	2C24	PLAY	232D	PSET
2328	PRESET	2BFD	SOUND	459A	SCREEN
46D8	VPOKE	3120	KEY	31AF	CLICK
337F	SWITCH	7CBA	MAX	7B44	MON #
2BE5	MOTOR	7684	BLOAD	7624	BSAVE
3036	MDM	79C2	DIAL #	34A6	DSKO\$
34AB	SET #	34B0	NAME #	34B5	KILL #
34BA	IPL #	34BF	COPY #	34C4	CMD #
2FD1	LOCATE				

**RESTARTS**

There are also the RST vectors which are used regularly by BASIC. A RST is a form of CALL that requires only one byte to use, but is limited to locations in the first 256 bytes of memory. Notice that 38H is used also for the Z80 interrupt, which interfaces devices in a real-time manner. The RST's are:

RST 00H	BOOT ROUTINE	Resets computer
RST 08H	SYNTAX CHECK	Checks byte after RST with contents of memory set by reg. HL. If not equal to byte it then jumps to SYNTAX error routine
RST 10H	END OF LINE ?	Check for end of line; return ZERO flag set if end of line (':' or 00H). Also loads reg. A with byte of line set by reg. HL
RST 18H	PRINT CHR\$	If contents of F542H = 1 then the CHR\$ in reg. A goes to centronics port else it goes to video screen
RST 20H	COMPARE HL/DE	Returns with ZERO flag set if HL = DE NC " " " HL > DE C " " " HL < DE
RST 28H	?	Unknown at this time.
RST 30H	?	Unknown at this time.
RST 38H	INTERRUPT	Executed every 1/50th second. Detects ^STOP, SPRITES, SHIFT & CAPS LOCK, MODEM, KEY LABELS & DISKS.

**VECTORS**

Now we come to the vectors. These are "links" between parts of routines and are present mostly to enable changes to the system. Disk basic alters some vectors to create new commands. If they can do it, so can you! Most vectors are in high memory, and typical vector is set to the instructions of:

RET                  or                  JP    OF9EH  
NOP  
NOP

-the first simply returns to the routine that called it, the second jumps to an illegal function call error routine (commands marked # above do this). Each uses 3 bytes. You can replace this with a JP or CALL to another location. Once you've diverted the vector and executed your new routine, it is your responsibility to redirect the flow back to a suitable exit, so that the BASIC program will continue as normal (unless you don't want to, of course). The listing OK as printed last issue is a good example of this.

If we examine a disassembly of the listing we can see how it works:

FFC0	CALL 6463H	;Check print status
FFC3	LD HL,FFDOH	;Set HL to start of MSSG
FFC6	CALL 1B05H	;Print MSSG
FFC9	POP HL	;Clear Stack
FFCA	LD HL,09C4	;Set HL to next routine
FFCD	PUSH HL	;Dump to Stack
FFCE	RET	;Jump to next routine

FFD0 MSSG.....

FFD0+ LINEFEED + CARRIAGE RETURN + 0 BYTE

FE8E JP FFC0 ;Vector jump to above routine

—here the vector at FE8EH was changed to point to the new routine at FFCOH. Normally it was a link between the last action of BASIC and general command level.

The routine checks if there is a clear line, then sets the print pointer to the start of the MSSG. There is a call to print the MSSG and then a new return value is put onto the stack to return to the next routine. The linefeed/return and zero bytes were needed to make a neat display and terminate the string printed.

The table below list most of the known vectors. The VECT column lists the first byte of the vector, the call column lists which routine called the vector (where known) and the ABBR column is a simple abbreviation of the routine known to use the vector. As you can see this information is incomplete. A good way to solve this is to get some practice with your disassembler by decoding the BASIC commands.

VECT	CALL	ABBR	VECT	CALL	ABBR	VECT	CALL	ABBR
FE79	3CC2	IN1	FEE5	7736		FF51		KEYON
FE7C	129F	PRINT	FEE8			FF54	OE3E	
FE7F			FEEB			FF57	OE88	
FE82			FEEE	1061	RETURN	FF5A		IN2
FE85			FEF1	737A		FF5D	7933	MDM
FE88			FEF4	1AB8	LIST	FF60	7939	MDM
FE8B			FEF7	656A	RUN	FF63	793E	MDM
FE8E	09B8	OK	FEFA	163F		FF66	7943	MDM
FE91	OBED		FEFD	1A31		FF69	7948	MDM
FE94	09C4	INPUT	FF00	12D3		FF6C	7940	MDM
FE97	7457		FF03	1CC9		FF6F		DIAL
FE9A		LOC	FF06	0A31		FF72	79D8	
FE9D			FF09	0018	RST 18	FF75	30BB	ON XX
FEAO	65C0		FF0C	0D2C		FF78	3FCB	
FEA3			FF0F	7AA8		FF7B	3FD8	
FEA6			FF12			FF7E	0180	NMI
FEA9			FF15	77A5		FF81	3EB5	
FEAC	OC16		FF18			FF84		MON
FEAF	67A9	CLEAR	FF1B	098D		FF87		
FEB2	74C6	FPOS	FF1E	13B3		FF8A		DSKO\$
FEB5		SAVE	FF21	0BC4		FF8D		SET
FEB8		FILES	FF24	0B4B		FF90		NAME
FEBB		LOF	FF27	1365		FF93		KILL
FEBE	150C		FF2A	14D3		FF96		IPL
FEC1	7728		FF2D	71A5		FF99		COPY
FEC4	247C		FF30	0AC9		FF9C		CMD
FEC7		LOAD	FF33	71E8		FF9F		DSKF
FECA		EOF	FF36	0AD9		FFA2		DSKI\$
FECD	606B		FF39	17A0		FFA5		ATTR\$
FED0	70BF		FF3C			FFA8	7B4A	
FED3			FF3F	WIDTH		FFAB	3584	
FED6	1BB7		FF42	6D13		FFAE	3958	
FED9	6474		FF45	6D26		FFB1	3771	
FEDC	178E		FF48	6D2F				

```
FEDF      : FF4B  6F95      ;
FEE2 657F : FF4E      KEYOFF;
```

**CP/M CALLS**

It is possible to call the ROM routines from CP/M. This is a hazardous affair since page 21 must be swapped for page 01, the routine operated and then the pages swapped back. To do this you must place your entry address on the HL register and then CALL either E651H (a vector) or E7FFH (the routine). I would advise calling the vector in case the routine has been shifted on your version of CP/M.

Depending upon what parameters are required, you should be able to set up information for any routine, provided you study both the routine above and the ROM routine called. In CP/M this is no problem. There are more than enough utilities to enable you to do this. A side benefit is that if you have assembled an MCR in CP/M, you can now save it to tape by using a modified BSAVE routine. This could then be reloaded while the SV is in BASIC mode. A nice way of transferring programs.

**THANKS**

I would like to thank members of the Perth users group who helped in the preparation of this series on machine code routines. In particular, I'd like to thank Robert Brinkworth, David Elson and Peter Fox who supplied detailed information about the ROM. Robert has written two of the listings and is responsible for the lists in this part and the USR handlers. I believe he is soon to complete work on a new MCR game for the SV.

**NEXT MONTH**

Getting away from the subject of MCRS, I will cover Disk files and disk access and the efficient use of these from BASIC.

**MOVE BASIC TO RAM**

By, R.J. Tudor

```
1000 ' ****      ****
1100 ' **** MOVE BASIC ROM TO RAM ****
1200 ' **** ====== ****
1300 ' ****      ****
1400 '           By R.J. TUDOR
1500 '
1600 'The data statements in this program constitute a machine code program
1700 'which move the BASIC ROM to the lower RAM on SV328's and expanded 318's
1800 '
1900 'The program pokes this code into ram and then executes it to move the
2000 'BASIC ROM. The BASIC interpreter now executes from RAM and may be
2100 'patched by the adventurous.
2200 '
2300 AD=0
2400 READ BY$
2500 IF BY$="end" THEN 3000
2600 BY=VAL("&h"+BY$)
2700 POKE &HC000+AD,BY
2800 AD=AD+1
2900 GOTO 2400
3000 DEFUSR0=&HC000
3100 X=USR0(0)
3200 END
3300 DATA f5,c5,d5,e5,f3,3e,0f,d3
3400 DATA 88,db,90,32,00,c1,e6,fd
3500 DATA 32,01,c1,3e,08,32,02,c1
3600 DATA 21,00,00,11,00,b0,01,00
3700 DATA 10,ed,b0,01,00,10,b7,ed
3800 DATA 42,3a,01,c1,d3,8c,11,00
3900 DATA b0,eb,ed,b0,eb,3a,02,c1
4000 DATA 3d,32,02,c1,ca,47,c0,3a
4100 DATA 00,c1,d3,8c,c3,1b,c0,e1
4200 DATA d1,c1,f1,fb,c9,end
```

**CLARKE ELECTRONICS**

55-57 Belair Road, Kingswood, S.A. 5062

Telephone (08) 271 1216

Spectravideo

SV318	includes cassette recorder & 30 programs	\$199.00
SV328	" " "	\$449.00
SV728	with cassette recorder	\$590.00
Complete business system as per newsletter March 85		\$2390.00

PX-120 NLQ 120cps (same as Logitek 5002) printer	\$495.00
Texas Instruments 810 160cps (tractor only) printer S/H	\$299.00
Perkin Elmer 80 by 24 (50 to 9600 baud) terminal S/H	\$299.00
DSDD 5.25" Disks (10 up price)	\$32.00
DX85 Disk file boxes (with lockable lid)	\$22.00
ABC Swivel base for your monitor	\$25.00

CLARKE PC-88 Fully IBM compatible 16 bit personal computer ideal for hobbyist or business.

**SPECIFICATIONS:**

CPU Intel 8088 @ 4.77Mhz. Colour Graphics Card with Light pen port RGB or Composite video output and resolution up to 640 x 200 pixels. Multifunction card which includes 1 x Serial port 1 x Parallel port 1 x Games adapter port, Battery backed clock and 384K of memory. 2 x Floppy drives with controller (capacity 360K each). 130 Watt power supply to support the 8 card slots available. Mother board with 256K memory makes a total of 640K and will support a hard disk as system emulates the PC XT.

PRICE which includes 3 months warranty \$2195 inc. TAX P & P \$20

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

If you have ever written a program in Mbasic (Microsoft basic interpreter) under the CP/M operating system, you may have experienced the fatal error of returning to CP/M without saving the program to disk. The following hint will solve this problem provided you have not run another program before realising your oversight.

After returning to CP/M from Mbasic at the A> prompt use the save command to create a dummy com file E.G.

A>SAVE 0 DUMMY.COM(cr)

A>DUMMY(cr)

OK

The above dummy program is a name only com file and has no substance. This in turn loads nothing from the disk and commences to execute code at 100H. As Mbasic is still in memory you will get an OK prompt with all your lines of code safe and sound.

Note: If you are running your CP/M with ZCPR attached as in Steve McNamee's Custom CP/M system, then it is only a matter of using the GO command at the A> prompt.

**UNDERSTANDING CP/M Pt-3**

By S.W. McNamee

This month I will talk about things of interest to all you budding assembly language programmers out there. I will be assuming that the reader has a basic knowledge of binary and hexadecimal notation, and the fundamentals of the Z80 instruction set and architecture. If you do not possess this knowledge the following will probably be a little hard to comprehend, however it is not my intention to cover this basic knowledge. It can be obtained by reading any good book on the Z80 such as "Z80 Assembly Language Programming" by Rodney Zaks. Perhaps somebody else may like to try his new word processor out and write an article on the "Fundamentals".

As you have realised (I hope!) from the previous articles, CP/M is largely unconcerned with what sort of machine it is running on. The only requirements are an 8080 compatible CPU, at least 20k of memory and at least one disk drive. This is achieved by keeping all machine dependent routines in a standardized area and interfacing to them through a standard set of jump vectors. This area as you all know by now is called the BIOS - Basic Input Output System. A page of memory is reserved from 0000H to 00FFH and initialised with some standard jump vectors and buffers. This allows any applications software to access all the capabilities of the BDOS by simply calling these jump vectors using a defined set of rules. The BDOS then calls the BIOS through its' standardised jump vectors to perform all the required machine specific jobs - such as console I/O and low level disk drive operations.

I guess this is a good point to explain the difference between low level and high level operations. High level operations are those that interface directly with the user or applications software. Such things as opening files, erasing files, getting a line of input from the user etc. are high level, are all done by the BDOS, and are identical for all CP/M systems. Low level operations are those that deal directly with the hardware in a machine. Putting a character on the screen, writing a sector to disk and reading from a serial port are all low level operations. They are done by the BIOS and are nearly always completely different from machine to machine.

Well, now that I've got that off my chest on with the nitty gritty. The BDOS has 39 different routines that can be used by a programme for I/O and file manipulation. These are accessed by loading the C register with the number of the desired routine, the DE registers with the parameter to be passed (if any), and doing a CALL to location 0005H. Any returned value will be in the A or HL registers. I do not intend to discuss these routines individually as they are all quite well covered in the CP/M manual, however I will discuss the file operation routines in general and a few of the others which I have found to have quirks in their operation.

All the BDOS file manipulation routines must be passed an address of a block of memory 36 bytes long. The address is passed in the DE registers and describes an area called the File Control Block (FCB for abbreviation). This block must first be initialised with the drive code and the file name and certain bytes set to 0 before a file can be opened or created. One of my most frequent mistakes in my early days of assembler programming was forgetting to set these few bytes to 0 causing all sorts of weird things to happen when trying to open a file. The bytes that have to be set to 0 are the 13th and 33rd. Always

double check that you have done this and you will save yourself a lot of frustrating debugging. The use of the FCB is covered on page 93 of the CP/M manual and further information can be found there. Function 23 will allow you to rename a file and it is possible to rename a file to have the same name as an existing file. This would cause the BDOS to be totally confused when accessing a file of more than one extent so always make sure to have your programme check for the existance of a file name before it is used in a rename call.

BDOS function 9 caused me a few headaches as well. It will print a string on the console, the address of which is passed in the DE registers. The string must be terminated with a '\$' character and forgetting to append this character can cause a very interesting display as the BDOS attempts to display all of your machine code.

The only other function I will discuss is 37 which lets a particular drive or drives to be reset according to a vector of bits in the DE registers. This works fine except for the A: drive. I have found that under some circumstances the only way to be able to write to the A: drive after changing the diskette is to do a full disk system reset by calling function 13.

The CPR has a couple of useful facilities included which make life a little easier for the programmer. Whenever a transient command is called the CPR examines the information typed after the command and constructs up to 2 FCB's in low memory starting at 05CH and 06CH. The one at 05CH can be used where it is but first make sure that if you need the one at 06CH that you move it somewhere else. This is necessary because as soon as the FCB at 05CH is used in an OPEN call the FCB at 06CH will be overwritten. The CPR also leaves a copy of the command line tail (what is typed after the command verb) at location 80H for use by a transient programme. A very tricky thing to watch is that all leading blanks are included in this, so that there will be at least one blank preceding the actual string you are interested in.

Well that's all I will say this time. Next month I will conclude this series with a discussion of the BIOS and how various parts of it are implemented for the Spectravideo.

As promised last time I have included in this issue a copy of the RS-232 circuit I designed for my computer. A couple of points should be noted:

1. The circuit is NOT compatible (for reasons of design ease and cost) with the Spectravideo one.
2. It does not have any handshake lines and as such is only of use with a modem, NOT printers etc.
3. It uses different port addressing to the SV.

The physical construction is left up to the reader but I suggest using one of the readily available double sided experimenter boards with 50 way edge connector fingers etched on. The components were all available at Jaycar at the time I built mine. The following port location information will be needed when modifying software for this circuit:

- 08H Status Register - When bits 7 and 0 are BOTH 1 the UART is ready for another transmit character. When bit 6 is 1 the UART has a received character ready.  
 09H Received Character Register - The received characters appear one by one in this register.  
 0AH Transmit Buffer Register - The characters to be transmitted are loaded one by one into this register.  
 0BH Control Register - Must be loaded with the correct bit pattern for the mode of operation desired:

Bit7 - 1=no parity 0=parity check

Bit6 - 1=2 stop bits 0=1 stop bit

Bit5,4 Control the number of bits per character as follows:

Bit4	Bit5	Bits/Character
0	0	5
0	1	6
1	0	7
1	1	8

Bit3 - 1=Even parity 0=Odd parity

For most Australian bulletin boards and RCPM's at 300 baud the following byte should be loaded into the control register - OB0H.

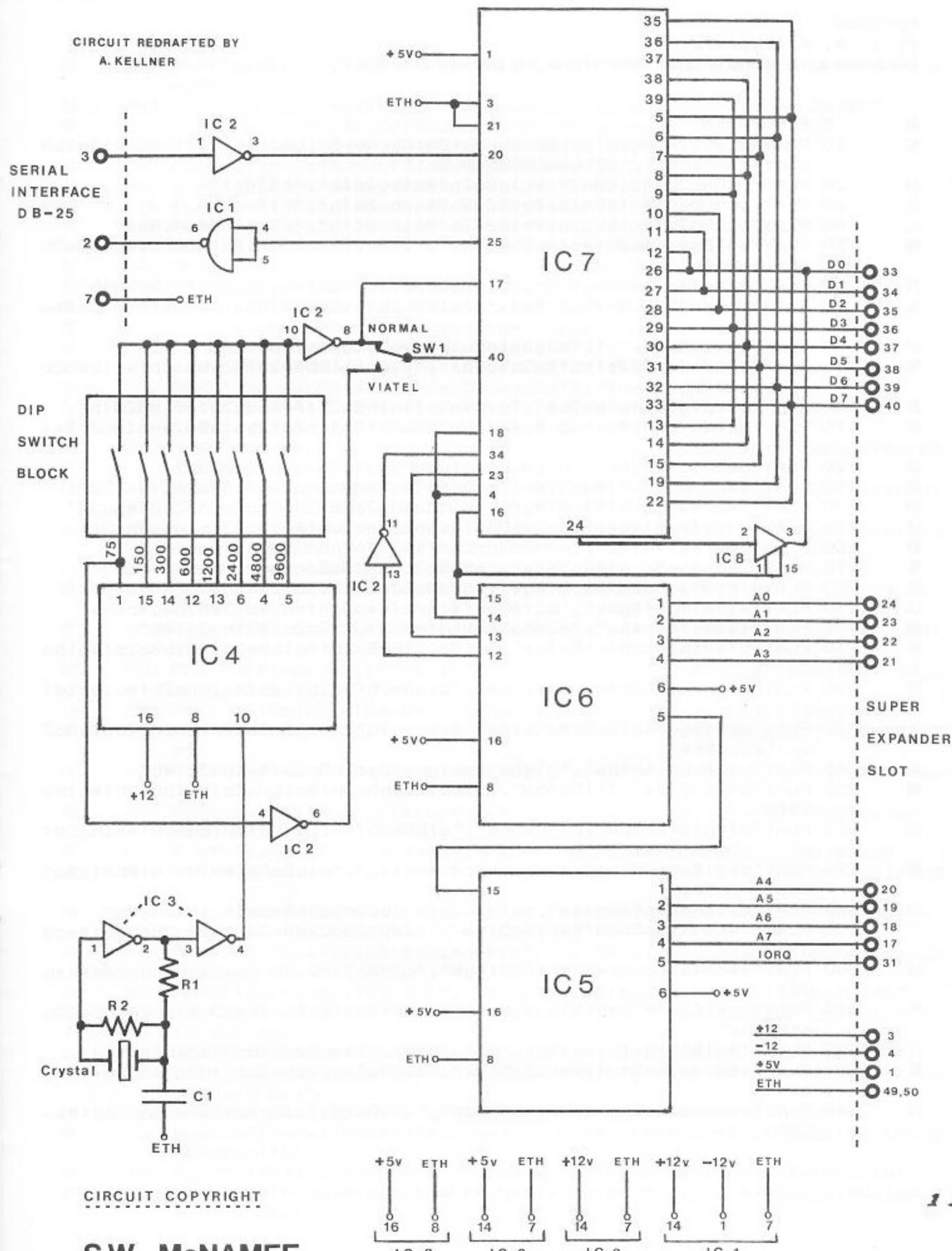
### RS-232 INTERFACE COMPONENT LIST

! COMPONENT !	! NO./ VALUE !	DESCRIPTION	! NO.OFF !	PRICE !
! IC 1	! 1488	! Quad line driver	! 1	! 2.00 !
! IC 2	! 1489	! Quad line receiver	! 1	! 2.00 !
! IC 3	! 4069	! Hex inverter	! 1	! 0.90 !
! IC 4	! 4020	! 14 stage binary counter	! 1	! 1.45 !
! IC 5 & 6	! 74LS138	! 3/8 decoder	! 2	@1.50 !
! IC 7	! AY-3-1015D	! UART	! 1	! 12.50 !
! IC 8	! 74LS365	! Tri state hex buffer	! 1	! 1.00 !
!	!	!	!	!
! R 1	! 2.2 K ohm	! Resistor 1/4 watt	! 1	! 0.06 !
! R 2	! 1.0 M ohm	! Resistor 1/4 watt	! 1	! 0.06 !
! C 1	! 10 pf	! Ceramic capacitor	! 1	! 0.15 !
!	!	!	!	!
! Crystal	! 4.1952 MHZ	!	! 1	! 6.50 !
!	!	!	!	!
! DIP SW.	!	! 8-Way dip switch	! 1	! 2.40 !
! SW 1	!	! SPST Instrument switch	! 1	! 1.25 !
! DB-25	!	! RS-232 male plug (25 pin)	! 1	! 6.50 !
! Vero Bd.	!	! Connector spacings 2.54mm	! 1	! 4.00 !

\*\*\*\*\* APPROXIMATE PROJECTED COST ONLY \*\*\*\*\*

TOTAL = 42.77

CIRCUIT REDRAFTED BY  
A. KELLNER



## POPCORN

by : M. Fitzgerald

This Program may be entered using the 'INPUT' program from Newsletter 2 - 2 (NOV 84).

```
GH      5 ' POPCORN
BC     10 PLAY"v10t12818o2br8br8br8br8br8br8br8br8o1bo2116bf#o1b8o2bf#o1b8
       o2bf#o1bo2dfb", "s0t64m65535s0o5b1"
DJ     20 PLAY"o1b8o2bf#o1b8o2bf#o1b8o2bf#o1b8o2dfb", "t128r1"
PL     30 PLAY"o1b8o2bf#o1b8o2bf#o1b8o2bf#o1b8o2dfb", "r1"
CL     40 PLAY"o1b8o2bf#o1b8o2bf#o1b8o2bf#o1b8o2dfb", "r2.m1000o418ba"
HG     50 PLAY"o418s0m1000bf#df#o3b4o4ba", "v10o1116b8o2bf#o1b8o2bf#o1b8o2b
       f#o1bo2dfb"
PO     60 PLAY"bf#df#o3b4o4bo5c#", "o1b8o2bf#o1b8o2bf#o1b8o2bf#o1bo2dfb"
OL     70 PLAY"dc#do4bo5c#o4bo5c#o4a", "o1b8o2bf#o1bo2df#bo1a8o2aeo1ao2c#ea
       "
DP     80 PLAY"babgb4ba", "o1g8o2gdo1gbo2dgo1b8o2bf#o1bo2df#b"
HC     90 PLAY"o418s0m1000bf#df#o3b4o4ba", "v10o1116b8o2bf#o1b8o2bf#o1b8o2b
       f#o1bo2dfb"
DF    100 PLAY"bf#df#o3b4o4bo5c#", "o1b8o2bf#o1b8o2bf#o1b8o2bf#o1bo2dfb"
CC    110 PLAY"dc#do4bo5c#o4bo5c#o4a", "o1b8o2bf#o1bo2df#bo1a8o2aeo1ao2c#ea
       "
CM    120 PLAY"bab05c#d4f#e", "o1g8o2gdo1gbo2dgo1g8o2gdo1gbo2dg"
O6    130 PLAY"f#do4ao5do4f#4o5f#e", "o2d8o3do2ad8o3do2ad8o3do2adf#ao3d"
B6    140 PLAY"f#do4ao5do4f#4o5f#g#", "o2d8o3do2ad8o3do2ad8o3do2adf#ao3d"
L6    150 PLAY"ag#af#g#f#g#e", "o2f#8o3f#c#o2f#ao3c#f#o2e8o3eo2beg#bo3e"
BN    160 PLAY"f#ef#df#4f#e", "o2d8o3do2adf#ao3do2d8o3do2adf#ao3d"
OC    170 PLAY"f#do4ao5do4f#4o5f#e", "o2d8o3do2ad8o3do2ad8o3do2adf#ao3d"
BK    180 PLAY"f#do4ao5do4f#4o5f#g#", "o2d8o3do2ad8o3do2ad8o3do2adf#ao3d"
LC    190 PLAY"ag#af#g#f#g#e", "o2f#8o3f#c#o2f#ao3c#f#o2e8o3eo2beg#bo3e"
CB    200 PLAY"f#edef#4o4ba", "o2d8o3do2adf#ao3do1b8o2bf#o1bo2df#b"
NA    210 PLAY"bf#d16f#o3b4r16o4ba", "v10o1116b8o2bf#o1b8o2bf#o1b8o2bf#o1bo
       2dfb"
FC    220 PLAY"bf#d16f#o3b4r16o4bo5c#", "o1b8o2bf#o1b8o2bf#o1b8o2bf#o1bo2df
       b"
DL    230 PLAY"dc#16ddo4b16o5c#o4b16o5c#c#o4a16", "o1b8o2bf#o1bo2df#bo1a8o2
       ae01ao2c#ea"
KN    240 PLAY"ba16bbg16b4ba", "o1g8o2gdo1gbo2dgo1b8o2bf#o1bo2df#b"
MM    250 PLAY"bf#d16f#o3b4r16o4ba", "v10o1116b8o2bf#o1b8o2bf#o1b8o2bf#o1bo
       2dfb"
EO    260 PLAY"bf#d16f#o3b4r16o4bo5c#", "o1b8o2bf#o1b8o2bf#o1b8o2bf#o1bo2df
       b"
DP    270 PLAY"dc#16ddo4b16o5c#o4b16o5c#c#o4a16", "o1b8o2bf#o1bo2df#bo1a8o2
       ae01ao2c#ea"
JI    280 PLAY"ba16bb05c#16d4de", "o1g8o2gdo1gbo2dgo1b8o2bf#o1bo2df#b"
EL    290 PLAY"f#do4a16o5do4f#4r16o5f#e", "o2d8o3do2ad8o3do2ad8o3do2adf#ao3
       d"
LK    300 PLAY"f#do4a16o5do4f#4r16o5f#g#", "o2d8o3do2ad8o3do2ad8o3do2adf#ao
       3d"
NF    310 PLAY"ag#16aaaf#16g#f#16g#g#e16", "o2f#8o3f#c#o2f#ao3c#f#o2e8o3eo2b
       eg#bo3e"
FL    320 PLAY"f#e16f#f#e16f#4f#e", "o2d8o3do2adf#ao3do2d8o3do2adf#ao3d"
EE    330 PLAY"f#do4a16o5do4f#4r16o5f#e", "o2d8o3do2ad8o3do2ad8o3do2adf#ao3
       d"
LG    340 PLAY"f#do4a16o5do4f#4r16o5f#g#", "o2d8o3do2ad8o3do2ad8o3do2adf#ao
       3d"
```

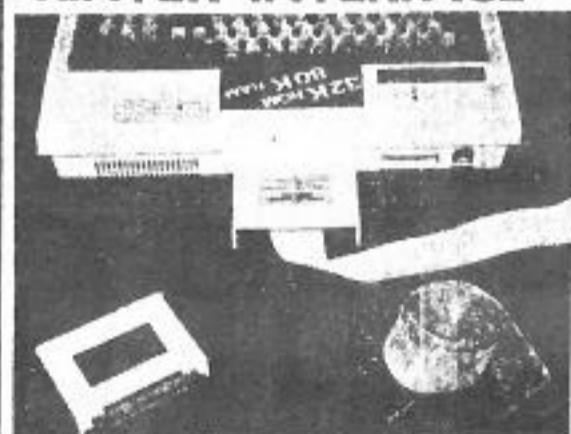
NJ 350 PLAY"ag#16aaaf#16g#f#16g#g#e16", "o2f#8o3f#c#o2f#ao3c#f#o2eBo3eo2b  
eg#bo3e"  
BG 360 PLAY"f#e16dd16ef#4r4", "o2d8o3do2adf#ao3do1b8o2bf#o1bo2df#b"  
DG 370 PLAY"v8l16o4bf#bo5df#do4bf#bf#bo5df#do4bf#", "v1011o3d"  
IM 380 PLAY"o4bf#bo5df#do4bf#bf#bo5df#do4bf#", "o2b2.o3d8e8"  
HJ 390 PLAY"bf#bo5df#do4bf#aeao5c#ec#o4ae", "o3f#2e2"  
DJ 400 PLAY"gdgb05do4bgdbf#bo5df#do4bf#", "d2o2b2"  
AG 410 PLAY"o4bf#bo5df#do4bf#bf#bo5df#do4bf#", "o3d"  
JD 420 PLAY"o4bf#bo5df#do4bf#bf#bo5df#do4bf#", "o2b2.o3d8e8"  
IA 430 PLAY"bf#bo5df#do4bf#aeao5c#ec#o4ae", "o3f#2e2"  
OF 440 PLAY"gdgb05do4bgdbf#bo5df#do4bf#", "d2o2b2"  
DE 450 PLAY"o5do4ao5df#af#do4ao5do4ao5df#af#do4a", "o3a"  
HG 460 PLAY"o5do4ao5df#af#do4ao5do4ao5df#af#do4a", "f#2.a8b8"  
CE 470 PLAY"o5c#o4ao5c#f#af#c#o4abg#bo5eg#eo4bg#", "o4c#2o3b2"  
AJ 480 PLAY"af#ao5df#do4af#af#ao5df#do4af#", "a"  
DI 490 PLAY"o5do4ao5df#af#do4ao5do4ao5df#af#do4a", "o3a"  
GP 500 PLAY"o5do4ao5df#af#do4ao5do4ao5df#af#do4a", "f#2.a8b8"  
CL 510 PLAY"o5c#o4ao5c#f#af#c#o4abg#bo5eg#eo4bg#", "o4c#2o3b2"  
AC 520 PLAY"af#ao5df#do4af#af#ao5df#do4af#", "a"  
AO 530 PLAY"s0m100018bf#d16f#o3b4r16o4ba", "v10o1116b8o2bf#o1b8o2bf#o1b8  
o2bf#o1bo2dfb"  
FD 540 PLAY"bf#d16f#o3b4r16o4bo5c#", "o1b8o2bf#o1b8o2bf#o1b8o2bf#o1bo2df  
b"  
DK 550 PLAY"dc#16ddo4b16o5c#o4b16o5c#c#o4a16", "o1b8o2bf#o1bo2df#bo1a8o2  
aeo1ao2c#ea"  
KO 560 PLAY"ba16bbg16b4ba", "o1g8o2gdo1gbo2dgo1b8o2bf#o1bo2df#b"  
MN 570 PLAY"bf#d16f#o3b4r16o4ba", "v10o1116b8o2bf#o1b8o2bf#o1b8o2bf#o1bo  
2dfb"  
EP 580 PLAY"bf#d16f#o3b4r16o4bo5c#", "o1b8o2bf#o1b8o2bf#o1b8o2bf#o1bo2df  
b"  
DO 590 PLAY"dc#16ddo4b16o5c#o4b16o5c#c#o4a16", "o1b8o2bf#o1bo2df#bo1a8o2  
aeo1ao2c#ea"  
IM 600 PLAY"ba16bb05c#16d4de", "o1g8o2gdo1gbo2dgo1b8o2bf#o1bo2df#b"  
DP 610 PLAY"f#do4a16o5do4f#4r16o5f#e", "o2d8o3do2ad8o3do2ad8o3do2adf#ao3  
d"  
LL 620 PLAY"f#do4a16o5do4f#4r16o5f#g#", "o2d8o3do2ad8o3do2ad8o3do2adf#ao  
3d"  
ME 630 PLAY"ag#16aaaf#16g#f#16g#g#e16", "o2f#8o3f#c#o2f#ao3c#f#o2eBo3eo2b  
eg#bo3e"  
FK 640 PLAY"f#e16f#f#e16f#4f#e", "o2d8o3do2adf#ao3do2d8o3do2adf#ao3d"  
ED 650 PLAY"f#do4a16o5do4f#4r16o5f#e", "o2d8o3do2ad8o3do2ad8o3do2adf#ao3  
d"  
LH 660 PLAY"f#do4a16o5do4f#4r16o5f#g#", "o2d8o3do2ad8o3do2ad8o3do2adf#ao  
3d"  
NI 670 PLAY"ag#16aaaf#16g#f#16g#g#e16", "o2f#8o3f#c#o2f#ao3c#f#o2eBo3eo2b  
eg#bo3e"  
BJ 680 PLAY"f#e16dd16ef#4o4ba", "o2d8o3do2adf#ao3do1b8o2bf#o1bo2df#b"  
KI 690 PLAY"o418s0m1000bf#df#o3b4o4ba", "v10o1116b8o2bf#o1b8o2bf#o1b8o2b  
f#o1bo2dfb"  
DL 700 PLAY"bf#df#o3b4o4bo5c#", "o1b8o2bf#o1b8o2bf#o1b8o2bf#o1bo2dfb"  
DE 710 PLAY"dc#16ddo4b16o5c#o4b16o5c#c#o4a16", "o1b8o2bf#o1bo2df#bo1a8o2  
aeo1ao2c#ea"  
KO 720 PLAY"ba16bb16b4ba", "o1g8o2gdo1gbo2dgo1b8o2bf#o1bo2df#b"  
KP 730 PLAY"o418s0m1000bf#df#o3b4o4ba", "v10o1116b8o2bf#o1b8o2bf#o1b8o2b  
f#o1bo2dfb"  
DH 740 PLAY"bf#df#o3b4o4bo5c#", "o1b8o2bf#o1b8o2bf#o1b8o2bf#o1bo2dfb"  
DI 750 PLAY"dc#16ddo4b16o5c#o4b16o5c#c#o4a16", "o1b8o2bf#o1bo2df#bo1a8o2  
aeo1ao2c#ea"

```

KA    760 PLAY"ba16bba16b4ba", "o1b8o2bf#o1bo2df#bo1b8o2bf#o1bo2df#b"
KH    770 PLAY"v10o5f#df#df#df#d", "o1b8o2bf#o1bo2df#bo1b8o2bf#o1bo2df#b"
BB    780 PLAY"o5f#df#df#df#d", "o1b8o2bf#o1bo2df#bo1b8o2bf#o1bo2df#b"
BC    790 PLAY"t120o4bo5do4bo5do4bo5do4bo5d", "t120o1b8o2bf#o1b8o2bf#o1b8o2
      bf#o1b8o2bf#"
BM    800 PLAY"o4bo5do4bo5do4bo5do4bo5d", "o1b8o2bf#o1b8o2bf#o1b8o2bf#o1b8o
      2bf#"
FL    810 PLAY"t100o4bo5do4bo5do4bo5do4bo5d", "t100o114br4br4"
FM    820 PLAY"o4bo5do4bo5do4bo5do4bo5d", "o114br4br4"
IA    830 PLAY"t32m65535o4b1", "t32s0o1b1"
AE    840 FORT=1T05000:NEXT
END

```

### SPI01 PARALLEL - SUITS SPECTRAVIDEO PRINTER INTERFACE SV318 — SV328



- NO MODIFICATIONS TO THE COMPUTER
- NO SEPARATE POWER SUPPLY REQUIRED
- ATTRACTIVE RIGID WHITE ACRYLIC CASE
- SUITS SV318 AND SV328 COMPUTERS
- ACCEPTS STANDARD CENTRONICS PRINTER CABLE
- DIMENSIONS 105W x 27H x 85D
- WEIGHT 100 GRAM
- CENTRONICS PRINTER CABLE AVAILABLE 11 metre

PRICE \$87

Your members \$78 inc P&P

### SP102 PARALLEL PRINTER INTERFACE

(FOR USE WITH THE SPECTRAVIDEO)

The SP102 Parallel Printer Interface is designed for use in the SV601 Super Expander or SV602 Mini Expander.

The SP102 Parallel Printer Card comes complete with its own inbuilt Centronics Printer Cable (just add a printer).

At an extremely competitive price of \$120 inc tax (plus \$3.00 P&P, why pay more?

YOUR MEMBERS \$108

incl tax & P&P

#### SP102 SPECIFICATIONS -

- 1.5 metre Centronics cable fitted
- Standard Centronics plug fitted
- High quality fibreglass Printed Circuit Board
- Gold plated edge connector for more reliable contact
- On-card power-on indicator fitted
- Computer reset facility (reset button optional)

### FDS FORREST DATA SERVICES

P.O. Box 71, Palmyra W.A. 6157  
201 Forrest Street, Palmyra  
Western Australia 6157

Telephone:  
(09) 339 5087

### the econo-buffer between computer & printer

PRICE \$330

Your Members  
\$297 inc P&P

#### SPACE COMPRESSION

This buffer has space-compression. In most printouts there will be empty or partly empty lines (such as between columns and paragraphs, in screen dumps also). These space codes will not be stored by just storing every character, but only the number of it.

This means an enormous saving in memory. The effective buffer capacity in practice will be more than 1: the 49152 characters for 48kB.

#### COPY-FUNCTION BUTTON

The contents of the buffer can be printed again by pressing the copy-button. Another extra from the Econo-buffer!

#### PAGE NUMBERING MODE

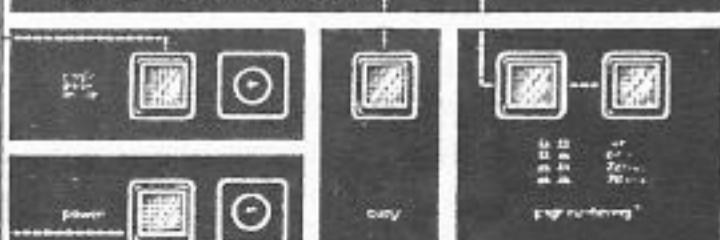
buttons for various page lengths

When making long listings it can be useful to have the printed pages numbered. The page length needed (66, 72 or 78 lines) can simply be chosen by a certain key combination. This function is also deselectable.

No complicated software commands... just a mere keypress.

#### ECONO-BUFFER

high speed parallel buffer with space compression



**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.

**PROGRAM CHANGES WITH "INPUT"***by Nick Booth, Sydney*

Like R. Duffy, I very much like the "INPUT" program, which makes life much easier when typing programs in, especially ones like "MAP", which has mostly a series of meaningless letters and numbers.

However, I find it's rather annoying that one cannot alter the material that's being typed in. It would be nice, for example, to be able to correct the odd spelling error that has crept into a PRINT statement. Or just to fiddle with the program a bit! All that is needed is to add one line:

```
225 IF RIGHT$(A$,1)=""" THEN A$=LEFT$(A$,LEN(A$)-1):GOTO360
```

When you alter a line as you enter it, hit the ~ symbol (just above the ENTER key) before you hit ENTER. Line 225 will recognise this symbol, remove it from the input line, and then by-pass the other checking procedures. (As there are no checks, type the line carefully!)

The reason for choosing the """ symbol is that it is highly unlikely that it will ever crop up at the end of a Basic program line.

**SOUND WITH A MONITOR***by Nick Booth, Sydney*

My article about Expander modifications in the November issue of the Newsletter has aroused some interest, and I have investigated more closely the possibility of a neater arrangement than the one I have installed on my SV (which rather resembles wire and plastic spaghetti).

I find that it is possible to build a small single-sided printed circuit board which will provide both an audio amplifier and the reset function. This board is not in a fancy plastic box like the genuine SV parts, but on the other hand it doesn't cost \$150!

Anyone who is interested can get this board by sending me \$15 and one of the push-out covers from an Expander. Allow a couple of weeks for delivery, and you'll receive a board to plug into the Expander with volume control and reset button fitted into the push-out cover.

You will have to provide a loudspeaker, but this is not critical: almost anything will do. If you don't have a spare small speaker on hand, your best bet might be to get one that is designed to screw onto a car parcel shelf (try Dick Smith/Tandy/Jaycar/etc.). Given the sorts of noises that the SV makes, there's little point in spending hundreds of dollars on a hi-fi speaker!

Anyone interested can contact me at 203 Bridge Road, Glebe, 2037; telephone (02) 660 0684 after 7.30 p.m. weekdays or after 11 a.m. weekends.

# buy, trade & sell

**FOR SALE**

ONE S/S SV-902 DISK DRIVE STILL IN  
ORIGINAL PACKING FOR ONLY \$350.00

ONE SVI-105 GRAPHICS TABLET ALSO IN  
ORIGINAL PACKING \$120.00

A MINI EXPANDER FOR ONLY \$20.00

**CONTACT:**

BRENDAN PRATT,  
26 OATBERRY CRESC.,  
SHAILER PARK.

**PHONE:**

(07) 2098718 HOME  
(07) 3437600 WORK

YEARBOOK NEW PRICE IS NOW \$20.00

M.S.X. CASSETTE CABLES \$15.00

M.S.X. PRINTER CABLES \$35.00

ROM BASIC EXPLAINED IS NO-LONGER  
AVAILABLE. REPRINTS ONLY IF ENOUGH  
INTEREST IS SHOWN BY 318/328 OWNERS.

# buy, trade & sell

---

**FOR SALE**

1 SPECTRAVIDEO SINGLE-SIDED DISK DRIVE.  
WITH SPARE DRIVE BELT. \$299.00

**CONTACT:**

GREG HASTINGS,  
20 SORELL ST.,  
GEORGE TOWN,  
TAS. 7253.

(003) 821035 A.H.

---

**FOR SALE**

COLECO GAMES CARTRIDGES, ELEVEN  
DIFFERENT TITLES, ALL IN NEW CONDITION &  
WITH FULL DOCUMENTATION. \$12.00 EACH OR  
ALL ELEVEN FOR \$110.00.

PHONE ENQUIRIES AFTER 5 P.M.  
TO (003) 822919.

---

**FOR SALE**

A NEW SV-807 (64K RAM EXPANSION)

**BEST OFFER****CONTACT:**

DAVID MARTIN,  
20 WEBB STREET,  
HENLEY BEACH,  
SOUTH AUST. 5022

---