

# SINC-LINK

VOL. 8 NO. 6

## NOV - DEC '90

SINC-LINK IS A PUBLICATION OF THE TORONTO TIMEX-SINCLAIR USERS CLUB AND IS ISSUED 6 TIMES A YEAR. COPIES OF THE NEWSLETTER ARE \$1.50 EACH FOR NON-MEMBERS. CLUB MEMBERS RECEIVE FREE COPIES AS PART OF THE \$20.00 ANNUAL MEMBERSHIP FEE. A NEWSLETTER SUBSCRIPTION ONLY IS AVAILABLE FOR \$12.00.

NEWSLETTERS ARE EXCHANGED, FREE OF CHARGE, WITH OTHER TIMEX-SINCLAIR USERS GROUPS.

PLEASE CREDIT THIS PUBLICATION AND THE AUTHOR IF YOU COPY MATERIAL.

THE CLUB MEETS ON THE FIRST WEDNESDAY OF EACH MONTH AT FOREST HILLS COLLEGIATE INSTITUTE, 730 EGLINTON AVE. W., TORONTO. START TIME: 7:00 PM.

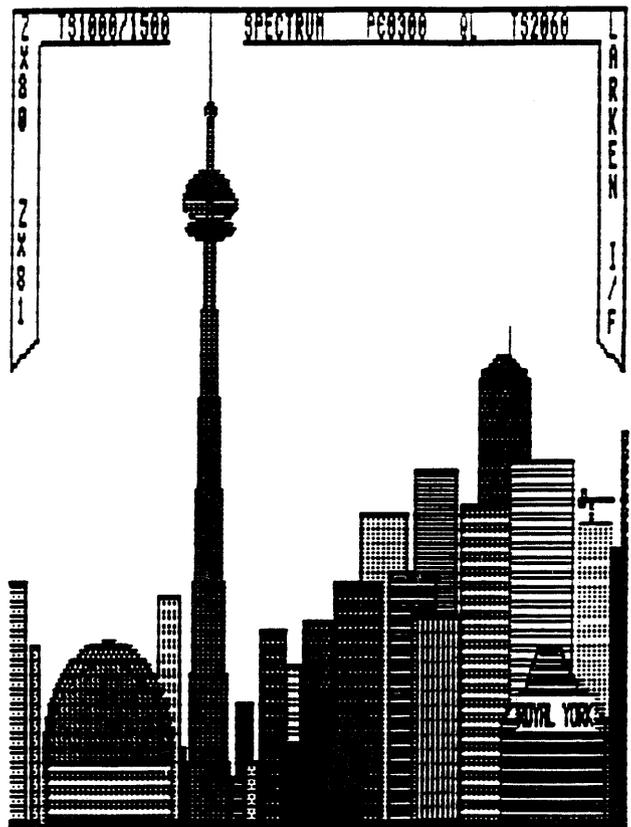
SEND CORRESPONDANCE TO:

Attention: SINC-LINK EDITOR  
TORONTO TIMEX-SINCLAIR USERS  
CLUB, 14 RICHOME COURT,  
SCARBOROUGH, ONTARIO, CANADA M1K 2Y1

EXECUTIVE OFFICERS:

PRESIDENT:  
TREASURER:  
SECRETARY:  
ACTIVITIES:  
TAPE LIBRARIAN ZX81:  
TAPE LIBRARIAN 2068:  
TAPE/DISK LIBRARIAN QL:  
PAPER LIBRARIAN:  
NEWSLETTER:  
LIAISON OFFICER:  
( Out-of-town members )

(Area Code 416)  
RENE BRUNEAU ( 531-9749 )  
BILL LAWSON ( 444-8772 )  
GEORGE CHAMBERS ( 751-7559 )  
RENE BRUNEAU ( 531-9749 )  
LYMAN PAQUETTE ( 482-4479 )  
RENATO ZANNESE ( 635-6536 )  
HUGH HOWIE ( 634-4929 )  
JEFF TAYLOR ( 244-8583 )  
JEFF TAYLOR ( 244-8583 )  
GEORGE CHAMBERS, 14 RICHOME COURT,  
SCARBOROUGH, ONTARIO, M1K 2Y1  
( 416-751-7559 )



TORONTO TIMEX-SINCLAIR  
USERS CLUB

**TORONTO TIMEX-SINCLAIR USERS CLUB**

## Editorial

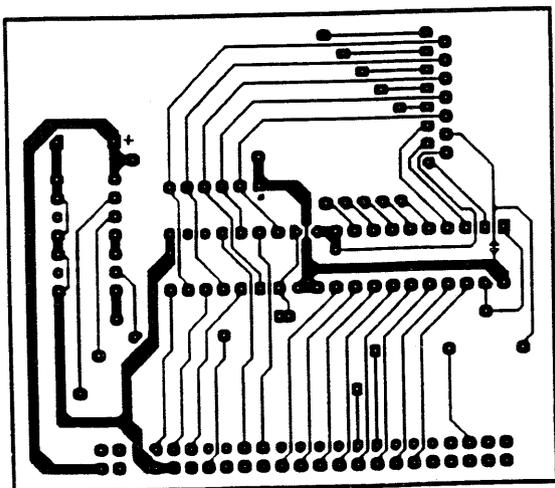
I recently came upon a happy discovery for club members with modems. A BBS devoted entirely to Timex-Sinclair 2068s and running on TS modem software! I'm talking about "SIR CLIVE'S CASTLE" running out of Ottawa, Ontario at (613)-745-8838. The system runs at 300 and 1200 baud and uses Larry Kenny's MAXCOM terminal software. You can call in using any of the modem programs available for the 2068. I believe this phone number is more current than the one listed in the article about this BBS. I got on using the number above the week of October 31.

I cannot stress how important it is to register and participate in making this BBS grow. It doesn't cost that much to call Ottawa, especially late at night. The BBS is free and runs 24 hours so what more do you want? As the saying goes, "If you don't use it, you lose it". Let's encourage the sysops.

Also, don't forget that there is a TS file area on the TIBB WIZARD BBS in Toronto at (416)-743-6703. Get Modeming.

That's all for now ...

J.T.



## Larken Disk Library by G. Chambers

Most of the disks in our Larken Disk Library have a HELP file. Usually this consists of a Tasword file which is displayed on screen in 64 column format.

This is a very handy way of providing documentation about the programs on the disk. Often though, it is desirable to print this out to hard copy rather than simply read it on-screen. We may sometime decide to put in a "print to paper" option in the disk menu. However, in the meantime here is an easy way to obtain printouts to a large printer.

Assuming that you have a large printer hooked up to your 2068 this is the procedure you may follow:

1. AUTOSTART the disk to bring up the menu.
2. Break out of the program and enter the command: PRINT USR 100: OPEN #4 3,"lp"  
(This opens a channel to the large printer, a channel which the command LPRINT can access)
3. LIST the program to find the LINE which prints out the "help" file. It will have a command very much like this PRINT #4 : PRINT "help.CT"  
or RANDOMIZE USR 100: PRINT "xxxxxx.CT"
4. Change the PRINT command to LPRINT.
4. Load your printer with paper, and activate it.
5. RUN the menu program and select the HELP file option. Printing should ensue.

That's all there is to it.  
\*\*\*\*\*

Does anyone have the documentation for WORD SYNC version 3?  
Please contact the Editor.

Last issue this PCB layout was not photo-reduced to the correct size. Here is the proper sized copy.

- Ed.

# 280 DICTIONARY of Instructions

Adapted from Toni Baker's book  
"Mastering Machine Code on the  
ZX81" by Bob Mitchell 900916.

## NOTE:

Throughout these instructions the operand <r> denotes a single register A,B,C,D,E,H or L, a numerical constant or an address pointed to by <HL>, <IX+d> or <IY+d>; the operand <s> denotes any register pair BC,DE,HL,SP,IX or IY.

**ADC** Two forms ADC A,r and ADC HL,s; ADC A,r is a single byte instruction which calculates the sum of A+r+carry flag and stores result in A. ADC HL,s is a two byte instruction which calculates HL+s+carry flag and stores result in HL. ADC A,A does the same thing as RLA but ADC alters all the flags.

**ADD** Similar to ADC except carry flag is not used in the initial calculation but is altered by the final result. Differences from ADC: ADD HL,s is a one byte instruction; ADD IX,s and ADD IY,s are allowed.

**AND** AND r is only form. The A register is altered one bit at a time. If the bit=0 it goes unaltered. If the bit=1, it takes on value of corresponding bit of r. AND FF leaves A unchanged; AND alters all flags with the carry flag reset to 0.

**BIT** Finds the value of a bit in a register. BIT 5,B returns a zero or non-zero which can be used to exploit using JR Z for example, or RET NZ. BIT does not alter the value of any of the registers nor change the carry flag.

**CALL** Like GO SUB. PUSHes return address onto the stack and JUMPs to the call address. The return address is used by the RET instruction. The subroutine must not alter the stack unless any PUSH in it is followed by a POP before the RET. CALL has several forms (eg, CALL Z,pq which means CALL pq (an absolute address) if the last calculation was zero otherwise continue with the next instruction.

**CCF** Complement Carry Flag. If zero then it becomes one and vice versa.

**CP** ComPare. CP r is the form and will calculate A-r; the result is not stored anywhere nor is A or r changed. It alters all the flags. CP r followed by JR Z will JUMP if A=r.

**CPD** This is like CP(HL)+DEC HL+DEC BC. Zero flag is changed as if a single CP(HL) had been executed. P/V flag is altered thus: If BC becomes zero then P/V=0; if BC not=0 then P/V=1. So JP PO will JUMP only if BC=0. JP PE will JUMP only if BC not=0. Carry Flag is not altered by CPD.

**CPDR** Same as CPD but with automatic loop. Stands for ComPare with Decrement and Repeat. Loop ends when A=HL(zero flag set) or BC reaches 0.

**CPI** As CPD but Incremented.

**CPIR** As CPDR but Incremented.

**CPL** ComPLeMent. A is altered bit by bit. 0 becomes 1 and vice versa. 11010101 becomes 00101010. No flags or other registers are altered.

**DAA** Decimal Adjust Accumulator. Changes hex number to decimal. Flags changed appropriately.

**DEC** Two forms. DEC r (single register), or DEC s (register pair). In DEC r, r is DECReased by one, carry flag is unchanged and zero flag is changed appropriately; in DEC s, the zero flag is not altered nor are any others. Thus DEC BC followed by JR NZ,-3 is either an infinite loop or has no effect.

**DI** Disable Interrupts.

**DJNZ** Decrement B and Jump Relative if Not Zero. If B=7, then it becomes 6; if 0 then it becomes FF. If B=0 then action JUMPs to new destination. Form is DJNZ e where e is a single byte. If e is between 00 and 7E, the JUMP is forwards, if between 80 and FE then backwards. Start the count from the next instruction so that DJNZ 00=DEC B, except that DJNZ does not alter any flags.

**EI** Enable Interrupts. See IM for more information.

**EX** EXchange values between register pairs. AF,AF', DE,HL (SP),HL (SP),IX (SP),IY. No flags are altered. With the last three, the old value of HL (or IX or IY) is pushed onto the stack and at the same time, the old value at the top of the stack is popped and loaded into HL. The position of the stack pointer is therefore unchanged. AF' called AF dash is a register pair distinct from the real AF and this is the only instruction that uses it.

**EXX** As well as AF' there are also BC', DE', and HL' which

are a set of six new registers or three register pairs. These can only be accessed by this one single instruction. Useful in preserving the values in the main registers while calling a ROM routine which relies on A but wipes out the other registers. The example is <EXX> <CALL ROM subroutine> <EXX>.

**IN** Three forms: IM 0 IM1 and IM2. If zero, the interrupt device itself must supply an instruction to be executed. If one, the instruction RST 38 is executed. If two, the interrupt device must supply one byte of data which is the low byte of an address. Register I is used to supply the high byte of the address. The computer then looks up this address and should find a second address store there and this is used as a subroutine call. But note: The TS2068 requires the use of a table of 257 bytes to do this job properly.

**IN** Input with two forms: IN A,(n) where n refers to an external device with a different n for each device. One byte is read from device n and loaded into A. No effect on flags. IN r,(C) where the C register specifies the device and the number input is loaded into r; the flags are altered.

**IND** Input with Decrement= IN (HL),(C) + DEC B + DEC HL. The carry flag is not altered but the zero flag is altered to show whether B has decremented to zero.

**INDR** like IND with Repeat until B reaches zero.

**INI** like IND except that HL is Incremented.

**INIR** like INDR except that HL is Incremented.

**INC** INC r INCREASES the value of r by one. All flags but the carry flag is altered. INC s where s is a register pair will not change any of the flags. If s is FFFF it will zoom around to 0000 and carry on. Do not use a check for zero after INC s as it will not work. INC HL + JR Z means jump if the instruction before INC HL came to zero, not if HL has reached zero. INC H + JR Z does work.

**JP** Jump is like GO TO but to an address. JP NZ, 4300 means jump to 4300 if NOT zero (ie the zero flag is not set). JP (HL) allows for variable destinations, but no conditions here. Destination variables can be in HL, IX and IY and these are powerful instructions for HL

can be the result of a calculation, even at random.

**JR** Jump Relative and one byte shorter than JP. Four conditions JR Z, JR NZ, JR C and JR NC. Write the instruction as JR e where e is a single byte, forward if e is 0-7F, backward if 80-FF.

**LD** Load. Transfers data from one place to another. Simplest of many forms is LD r1,r2 where r=register. LD A,(BC) and LD A,(DE) are legal and are one byte codes; these are reversible, eg, LD (BC),A. REMEMBER THAT THE BRACKETS MEAN THE CONTENTS OF THE ADDRESS BC (or DE). Two special registers R and I may be loaded to and from A as in LD A,I LD A,R LD I,A LD R,A (but only A). R is the memory refresh register used for outputting to the screen; I is used in IM. Register pairs may all be loaded with either numerical constants or the contents of absolute addresses eg, LD s,mn or LD s,(pq). LD (pq),s is like a double POKE in BASIC. The register pair SP, the stack pointer, can be loaded directly with either HL, IX or IY. You can say LD SP,HL but not LD HL,SP.

**LDD** Load with Decrement. LD(DE),(HL)+ DEC HL+ DEC DE + DEC BC all in one go. Carry flag, zero flag and sign flag are unaltered but P/V flag becomes zero if BC becomes zero, one otherwise. JP PO will jump only if BC is zero after the instruction.

**LDDR** Like LDD but Repeats until BC reaches zero.

**LDI** Like LDD but DE and HL are both Incremented.

**LDIR** Like LDDR but DE and HL are both Incremented.

**NEG** Alters the accumulator and all flags. It NEGates A. If A contains 1 then NEG will change it to minus 1 or FF. If A contains -6 (FA) then NEG will alter it to +6 (06). Same effect achieved using CPL + INC A but this does not affect the carry flag as NEG does. NEG is faster.

**NOP** No OPERATION. Wastes time. Like a REM statement in BASIC but you cannot put messages after it. Use it also to overwrite previous machine code when debugging.

**OR** OR r is opposite of AND r. If a bit is one then it will be unaltered but if zero then it will take on the value of the corresponding bit in r. If A contains 00 then OR r is the

same as LD A,r (except for the flags). If A contains FF then OR r will not change it. All the flags are changed and the carry flag is reset to zero.

**OUT** Like IN. Unlike OUT in BASIC. Instruction OUT(n),A where n is a one byte numerical constant, will transfer contents of A to external device n. OUT(C),r will transfer contents of r to the device pointed to by register C. OUT is used in ROM to SAVE things and has no effect at all on the flags.

**OUTD** Output with Decrement. Carry flag unchanged; zero flag depends on the final result of B. OUTD is equivalent to OUT(C),(HL) + DEC HL + DEC B.

**OTDR** Output with Decrement and Repeat. Like OUTD but with repeat until B=0.

**OUTI** Like OUTD but HL is Incremented.

**OTIR** Like OTDR but HL is Incremented.

**POP** Remove two bytes of data from stack top and load them into any register pair except SP. POP AF will act on the A register and the flags register. Low byte is popped first then the high. The computer remembers that the stack is now two bytes shorter and alters the value of SP automatically.

**PUSH** PUSH s is the opposite of POP s. It stores the contents of any register pair (except SP but including AF) at the stack top. It alters the stack pointer accordingly. The high byte is pushed first then the low so that the low part is at the top. After a push instruction SP will point to the address of this low part.

**RES** Alters individual bits of any register. Set means change to one; RESet means change to zero, eg, RES 3,D will Reset bit 3 of D to zero. RES has no effect on the flags.

**RET** RETURN from a subroutine. It pops an address from the stack top and jumps to that address. This address can be altered by changing the value at the stack top, eg, POP HL + INC HL + PUSH HL will increase the return address by one. You can store one byte of data right after the CALL instruction, then POP HL + LD A,(HL) + INC HL + PUSH HL to store that byte in A while at the same time ensuring that the subroutine will return to the address after that data. Another trick is to push an

artificial return address onto the stack and then JP or JR to a subroutine instead of calling it. Now it will "return" to wherever you want it to go! RET may be used with conditions if needed. It does not alter the flags.

**RETI** Used to end an Interrupt subroutine (see IM). Its function is the same as RET, but RETI must be used instead of RET because the chip does clever things if a 2nd interrupt occurs in the middle of an interrupt subroutine: a DI instruction is executed automatically when an interrupt subroutine is called, but there are such things as non maskable (super high-powered) interrupts that override even DI. These can cause confusion if you don't use RETI.

**RETN** Used to end an NMI. Its function is like RETI except that the IM (altered by the NMI in the first place) is also restored to its previous value.

**RLA** Rotate Left Accumulator. Each bit of A is moved one position to the left. The leftmost bit is moved into the carry and the rightmost bit takes on the previous value of the carry, eg, if A=10010101 & the carry was 0 then after RLA, A will be 00101010 and carry will be 1. Only the carry flag is altered by this instruction.

**RL** RL r (which may be RL A and be taken in error to be RLA). Like RLA but alters ALL of the flags; RLA will only alter the carry. RL may be applied to any register, not just A. RL A does exactly what ADC A,A does down to the last flag, except one that you cannot get at: the H flag. The only way to tell the difference is by following it with a DAA instruction. ADC A,A is twice as fast.

**RLCA** Almost like RLA but not quite. Each bit of A is moved one position to the left and the leftmost bit is moved BOTH into the carry AND into the rightmost position of A. If, as before, A started off with 10010101 and carry was zero, then after RLCA it will be 00101011. The carry will also be one. Only the carry flag is changed - the previous value of which is lost forever.

**RLC** RLC r will Rotate Left with Carry the register r in the same way that RLCA does with A. RLC A is a valid instruction which is not same as RLCA. RLC B is a valid instruction but there is no such instruction as RLCB. Always check the spacing. RLC r

will alter all of the flags.

**RLD** Rotate Left Decimal. Do not confuse with RL D. RLD works as follows: Write the value of A and the value of address (HL) in hex. The second hex digit of (HL) is shifted left so that it becomes the first digit. The first digit overwrites the second digit of A. The second digit of A moves to the second digit of (HL). Thus if A contains 25 (hex) and (HL) contains EB then after an RLD has been carried out A will contain 2E and (HL) will contain B5.

**RRA** Like RLA except that the bits are moved Right.

**RR** Like RL except that the bits are moved Right.

**RRCA** Like RLCA except that the bits are moved Right.

**RRC** Like RLC except that the bits are moved Right.

**RRD** The contents of (HL) are moved 1 hex digit to the right, the rightmost digit moving into the rightmost digit of A which in turn becomes the left digit of (HL). If A equals 25 hex and (HL) equals EB then after RRD, A will equal 2B and (HL) will equal 5E. Note that RRD twice is the same as RLD once and vice versa. All the flags except carry are altered.

**RST** ReStart: like CALL except that it is only one byte long. But conditions are not legal and only one of eight specific addresses may be called: RST 0 = power down and up; RST 8 = error number follows; RST 10 = print current character; RST 18 = get char; RST 20 = get next char; RST 28 = start Floating Point Calc; RST 30 = make a space; RST 38 = frames counter and keyboard scan.

**SBC** Like ADC with two forms: SBC A,r which will subtract r from A and then Subtract the Carry digit. SBC HL,s which will subtract both s and the carry flag from HL. SBC A,A is useful: if the carry is one then A will be reassigned FF and the carry will still be one.

**SET** The opposite of RES. SET 4,H will change the value of bit 4 of H to one. Any bit of any register may be set.

**SLA** Shift Left Arithmetic. The form is SLA r. It is similar to RL r except that the rightmost bit is automatically replaced by zero. It alters all the flags. Note that SLA A does the same thing as ADD A,A except that ADD

A,A is faster.

**SRA** Shift Right Arithmetic. Any register may be shifted right using the format SRA r. The rightmost bit falls into the carry but the leftmost bit remains unaltered. Thus after an SRA instruction bit 6 will always be the same as bit 7. The effect of SRA is to divide both positive and negative numbers by two. FC (minus four) become FE (minus two).

**SRL** Shift Right Logical. As SLA except that the bits are shifted right instead of left and the leftmost bit becomes zero.

**SUB** SUB r = SUB A,r. The value of r is SUBtracted from the A register. Note that unlike ADD, there is no corresponding instruction SUB HL,s. If you wish to do this you must first of all reset the carry flag (usually by use of AND A) and then use SBC HL,s.

**XOR** XOR r alters all of the flags, resetting the carry to zero and the A register alone. r is not altered; A is altered bit by bit like with AND and OR. If a bit is zero it takes on the value of the corresponding bit of r. If on the other hand a bit is one then its new value is the complement of the appropriate bit of r. XOR A is very useful since in one byte it zeroes both the accumulator and the carry flag. So does SUB A.

**HALT** This definition belongs up above. Halt is not the same as STOP. It means "do nothing" but keeps repeating NOP instructions until it gets an interrupt. Use EI before HALT but not DI before HALT.

Finally, I have checked this summary against the referenced book; even so, it would be wise to consult a good Z80 manual if in doubt about any definition.

## Report from QL Librarian 1990

Last year at this time the QL membership was about 20. Today, we have a list of 27 QL members, some of whom are admittedly only Newsletter members, but very important members at that. Some of our most active people are Newsletter\_only members. Some of our N/L only members are our most prolific contributors. Of those 27 members, we can count 8 as In-town members, that is those who attend the meetings on a reasonably regular basis; which leaves 19 as Out-of-Town members. I think the number of N/L members is 5, ( or 6) I am not too sure on that figure.

This leaves us with 19 O/T members. I have had contact by letter with no less than 11, with the occasional phone call thrown in. 11 out of 19 is I think, very gratifying, and indicates the amount of interest in the QL Library. The number of letters I have received is well in excess of fifty in this last twelve-month. Yes, I have them all on file. Yes, I have replied to them all. (This does not include internal club letters, or my personal letters)

At this time last year we had about 80 programs in the library. Our library to-day, consists of over 150 programs. The improvement is not only in the quantity, but a vast improvement in the Quality. I have sent out, on request, about 14 full disks, some of which are 80 track disks, and about a dozen plus cartridges. As a matter of interest, the 40 track disks are just a little more than the 80 track. With a few 3 1/2 tossed in. I will not try to count the number of programs concerned. I could, but the end result is not worth the toil of counting them. The most recent catalogue has generated a further number of requests already.

To go from nothing to what we have to-day is no mean accomplishment, and I would like to extend my thanks, the thanks of this club, to all those people who have contributed so much, in such a short time, to the benefit of this club. Thanks to those who send in material, and thanks to those who make use of the library..... That is my reward.

I am sometimes accused of being too serious, but if I were anything else, I would not be able to do a good job. As the foremost Sinclair Club in Canada, and one of the most powerful in North America, I believe that we must be forever trying to improve our position, and to do that means a constant effort on the part of every member, no matter who or where they may be.

In Jeff's editorial in the May issue, he stated we were arguably the strongest and richest club in North America. That is a very strong statement to make, but it is not quite correct. With apologies to Jeff, I would like to amend that statement. We are WITHOUT argument, the BEST in North America.

Let's keep it that way.

Hugh H. Howie.  
October 5/90.

## QLIPS

In the current issue of NESQLUG Newsletter, a Member Survey form was included, and in the process of filling it out, and adding a few remarks, I came up with an idea, and I don't know why someone has not tried it before, and if they had, why it did not work. Perhaps there is someone out there who can give me the answer. All comments will be welcome on this subject.

Ever since I started using the QL, I have been looking for some instruction on SuperBasic Programming. No one in my immediate circle would seem to have the necessary knowledge, and yet I know there is a thirst for this instruction, I know there is a desire for this instruction.

Here is my idea, I will throw it out to the winds and see what blows back. I will publish the results in this column.

Is there anyone out there willing to start a correspondence course in SuperBasic? Make a few bucks?

Each lesson or group of lessons would be accompanied by a test paper. Payment of say \$5 for each test? Too much? Too little? If you have an opinion write and let me know. If you would be willing to provide the service, write and let me know. I am willing to bet there are a number of takers out there, but if you do not tell us your opinion, then nothing will be done. This may be your last chance to get something going.

The idea may not be viable, but if you do not take this opportunity to make your views known, you may never get another chance. Anyone interested in the use and in providing this service, would you please write me giving me your opinion. Do you want to learn SuperBasic?

As I have said, the results WILL be published in the next issue of this Newsletter. ---Don't delay---Write today.

Hugh H. Howie  
QL Librarian  
586 Oneida Dr  
Burlington. Ont.

L7T 3V3

From the **QL LIBRARY**

There are many wonderful little things in our QL Library, and few are aware that they are there. Here is one of those gems, complete with the doc file which goes with it.

Hugh Howie.

"DISK\_RE\_NAME" by Real Gagnon.

### CHANGING THE NAME OF A DISK.

When you format a new disk, you can assign a name to the disk. But it is impossible to change this name after the disk has been formatted unless you reformat the disk and all the data on this disk will be lost.

The following program will correct this situation. You can now rename a disk without having to reformat this disk.

When you run the program, you will be asked for the number of the disk drive (e.g. 1 or 2 )." This program will only work with disk "" floppies "" and you can not rename microdrive. " The program will then display the name of the disk and will ask you for the proposed name The limit is 10 characters.

```
100 REMark Disk_RE_Name
110 REMark by Real Gagnon
120 :
130 REPEAT loop
140 PRINT #0,"Disk Drive Number :- " !;
145 d$=INKEY$(-1)
150 IF d$ INSTR '123456789':PRINT
#0,d$: EXIT loop
160 END REPEAT loop
165 :
170 OPEN#10,"flp"&d$&"_D2D"
180 :
190 GET#10\1,a$
200 PRINT#0,"Present name : "!a$(5 TO
14)
210 INPUT#0,"New name : "!n$
220 LET a$(5 TO 15)=n$
230 PUT#10\1,a$
240 :
250 CLOSE#10
```

This is a reprint from the last issue  
because this article was badly copied.

- Ed.

The C Page

By Timothy Swenson

Last month we looked at a simple program. This month we will look at how to do input and output in C.

C has no built in input or output commands. Functions are called that perform input and output. Last month we used the function printf. This month we will use the function getchar. Getchar gets one character from the standard input channel, the keyboard. The input from getchar is assigned to an integer variable.

The opposite function is putchar. Putchar puts one character to the standard output channel, the screen. Putchar takes in a single integer variable.

Take a look at program number 2. The program gets a character from the keyboard, then while the character is not a lower case q, the character is put to the screen and another character is read in.

This program takes characters entered from the keyboard and echos them to the screen. Getchar does not echo any characters by itself.

Any assignment can be used within an expression. Look at program number 3. With the assignment statement inside the logic of the while statement, one extra line can be left out. The assignment will be executed before the evaluation on it is performed. This shows you how C can become more compact (and more unreadable).

Since we are on the subject of input/output, let's take a look at string input/output. C does not have a string type. You have to use an array of characters. Since there are no string types, there are no string manipulation commands. Instead there are functions to do this.

Look at program number 4. The variable string is defined as an array of 20 characters. Now a call is made to the function gets. Gets is short for "get string." This function gets a string from the standard input and assigns it to the variable string.

Now in the while statement a comparison is made to see if the string is of length 0. Strlen is a function that will return the length of the string. If the length is greater than 0, the string is sent to the standard output with puts (put string) and another string is inputed.

A lot of necessary functions are included in the standard C library. Most commands in basic have an equal C function. Plus you may write your own functions and add them to the library. There has always been a problem that not all C libraries are equal and some programs may not be ported from compiler to compiler.

Next month we will look at how C handles files.

```
/* Program 2 */  
#include <stdio_h>
```

```
main()
{
    int c;

    c = getchar();
    while ( c != 'q' ) {
        putchar(c);
        c = getchar();
    }
}
```

```
/* Program 3 */
#include <stdio_h>
```

```
main()
{
    int c;

    while ( (c = getchar() ) != 'q' ) {
        putchar(c);
    }
}
```

```
/* Prog 4 */
#include <stdio_h>
```

```
main()
{
    char string[20];

    gets(string);

    while (strlen(string) != 0) {
        puts(string);
        gets(string);
    }
}
```

=====  
 A R T I C L E S  
 =====

 -----  
 LARKEN MAXCOM  
 -----

A review by Greg Popovich

It has finally happened! Fellow LKDOS users, it has finally come to pass that the 2868 has finally gotten not only a terminal program BUT a BBS that will bring us into the 90's!!! What would you say to a terminal program that will handle both the Z-SIO AND the 2050. Either receive or send a file up to 110K DIRECTLY TO DISK. Have a 64 column display that CAN handle 1200 baud without losing characters. Can move either headerless (TSXmodem/Mterm2) or header type (Specterm 64) files whether they be Basic or Code. With a choice of 128, 512 OR 1024 byte block protocol (Checksum). And, ALSO display IBM Hercules (Monochrome) graphics! AND has a BBS built right in! You'd say I must be dreaming! WRONG! It is TRUE! It is REAL! I couldn't believe it myself. Being a seasoned modemer and BBS sysop I've gotten used to using Telix on my Clone to get around the BBS world. I've used all of the terminal programs that are available for all of the Timex's, and IBM & Clones. My tastes for terminal programs are sweet. I judge a program not only on how it looks, but on how well it performs and ease of operation. MAXCOM has it ALL! MAXCOM consists of 2 large Basic programs (one for the 2050 and one for the Z-SIO) as well as a 7K code file. When loaded it will display a Main Menu. From there you can go to the Terminal Mode or run the BBS. Also from this menu you receive and send files, select Xmodem packet size, load the BBS message base, Catalog a drive, toggle duplex, set the modem parameters, close or open the buffer (write to disk also!), dial the modem, and hangup the phone. You can, in the terminal mode, send the full ASCII control character table to the other computer from ACK to NAK. The terminal mode has a 64 column display that you can read without eyestrain. You also have a status bar at the top of the screen telling you your setup and whether or not you are online. Also, an added feature is that you can access up to 4 MacroKeys that you can utilize.

The BBS.....

Being a Sysop of a BBS (I use RBBS on a Clone). I can safely say that this BBS is one of the most advanced BBS's for a Timex computer. I once ran one on my 2868 but soon found that 64K, 3 800K, 1 400K floppies and 256K of Ramdisk wasn't cutting it for me. I started with TinyBoard and worked my way up to Casboard. I also ran modified versions of Casboard by James Rodlin and myself (don't get me wrong, I still use my 2868! (Like typing in this review) Sometimes I cannot even get on my own BBS!). Well MAXCOM's BBS you can:

Telecon  
3-14-91  
still has 15  
Larken VFs @ 124.95

- Use 300/1200 baud operation
- Have Passwords, Security levels, and Time Limits
- Multiple Messagebases with search
- Downloads, and Uploads without memory constraints
- Help files and Bulletins
- Access to LKDOS from Remote
- Interrupt driven hangup and timeout watchdog
- A userlog of the BBS activity
- Dual Window monitor screen
- Can be easily customized and expanded

send  
(12) x 9 or 10  
envelopes  
for monthly updates  
(with 'F' stamp)

Some of the features until now you only saw on the Big Blue boards. Like the Watchdog and access to DOS. You can even use the IBM ASCII set to setup your opening screen and Bulletins! You do have to do some customizing to start it up, like let it know how many drives you have. But for the most part not much has to be done to be on the air in under an hour. To assist you in setting up the BBS, Larry included an Editor to setup your screens and a deluxe file manager that can MOVE individual files to different disks (a value in itself). The error trapping is the most advanced feature of the BBS. MAXCOM uses 2 interrupt traps to trap almost anything that a user can do to crash a BBS (like dropping carrier to falling asleep behind to console). If an error occurs the interrupt routines will correct it by entering RUN 9230 (just as if you typed it in yourself) to reboot the system. These routines can handle lost carrier, timeout, timeup and Basic crashes. The operation of the BBS is simple to both the Sysop AND the User. This BBS comes a long way from the days of 1 messagebase, 20 messages Board and 300 baud only operation. For the \$25.00 you get more than your moneys' worth with this program. This program will work with the straight LKDOS sytem or LKDOS compatable setups. I've thoroughly tested MAXCOM and found it to be 100% bugfree (but a few minor misspellings). The program is available from:

with 80.11  
4%  
m/c 5.00  
shp 18.00  
2 disks 29.95  
POST +5 25.00  
Mercon

Larken Electronics  
RR #2, Navan, Ontario, Canada K4B 1H9  
(613)835-2680

RMG Enterprises  
1419 1/2 7th Street, Oregon City, OR 97045 (503)655-7484 Noon-10PM  
Pacific Time \* Tue-Sat

Ed Grey  
P. O. Box 2186, Inglewood, CA 90305  
(213)759-7406

For more information, call or write your favorite dealer at the above numbers or addresses, or myself at:

Greg Popovich  
C/O The W.Y.S.I.W.Y.G. RBBS FidoNet 1:107/564  
147 N7th Street, Paterson, NJ 07522  
(201)956-1556 (Voice) (201)956-7703 (BBS 8-1-N 300/1200/2400 BPS)

3 1/2" cells  
used for RAMdisk

text87

## MY IMPRESSION

by hugh h. howie.

A couple of months ago I decided I would like to try this wonderful new word processor, text87. So off went my bucks, (dear dear) and back came the word processor.

Now I don't know whether to laugh or cry. This darned thing is so complex I hardly know where to start with my diatribe. It is so easy to get into a rut, and this is what happened to me, I got into the most goldarnedest rut you ever saw or imagined, and even now I can't get out of it.

This pesky text87 (no capitals please, we're modest) is the most habit forming thing I have seldom, rarely, ever come across. It gets into you, it gets under your skin, it bugs you. Night and day you think about it, until there is no other program on the QL you want to even look at. You waken up in the middle of the night and say to yourself, "I must try this in the morning," you look at the clock, and by golly it IS morning. (3.15AM) so as it is morning, you have to have a go at your new toy.

Type F3 and you have a whole string of commands at your fingertips, all operated by the initial letter, the reaction time is instant, no waiting. Each keypress opens up a new string of commands, and then another. You are gently led from one stage to the next.

To scroll from top to bottom or bottom to top, is adequately fast, giving you time to have a look at what passes on the screen. If you want to get there in a hurry, that is how it can be done, instantly. Select a page? Done. Select a line? Done. Fast is the word for text87.

Change from one fount to another in mid sentence is easy. To change an already completed document in part or in whole, from one fount to another, is no problem. Even a Block in the middle can be changed in like manner. You can type a document in Elite, and when ready to print it, you can do so in Pica Italic Double Width underlined if that is your thing. I hate

to think how it would look on paper, but it can be done.

To change from one page width to another is done with the Ruler command, this can be a trifle tricky but with a little effort the way to do it becomes second nature. Tabs and Margins are set up from the Ruler command, can be retained in memory, and switching from one to another in separate parts of a document is only a keypress away.

'Tis a pity that the Ruler and Layout commands are not closer together so that you could set them both up at the same time instead of two separate procedures.

Print-out can be obtained in up to four columns, I find that two is ideal for Newsletter work. If your text will not fit inside the Ruler you have set up, you are told so as soon as you ask for a print-out; then you have to adjust the Ruler width, (or layout width) but this is really no hardship in the general run of things. After changing the Ruler width you can Reformat the whole thing to ensure proper spacing is maintained.

On occasion you will find that a two column width does not go onto the paper completely, leaving an odd line to go to the next page, once again no problem. Go to the Layout command and alter the length of your page. text87 makes all other adjustments which are necessary.

There are so many Founts, and Options for the founts, there is no need for other founts to be brought in, but this can be done. Each printer Driver has its own set of founts, sufficient for most needs.

Quill files, ASCII files or practically anything can be imported without too much trouble, a little editing may be required, but so simple.

In the latest text87, the one I have, Qtyp Spelling Checker is included. Now I don't like to rely on spelling checkers as I like to think I do not need one, but when you type a document and edit it, correcting as you go, then use the Checker and find an error you have overlooked, then Qtyp is a very nice thing! Qtyp, as supplied with text87, will not correct an error, just points it out, but the correction is no real

hardship. You just stop, make the correction, re-activate the checker, and carry on: Simple. You can even make your own dictionary if you wish.

I can not cover all the points of this program, just how I see it, but there is one thing I have a problem with, that is the header and footer section of the Layout command. Perhaps I am a bit thick, but I can not surmount this problem. It just don't work for me, yet other folks have no problem. As soon as I write this the answer will probably come to me. I hope so.

It could be my printer, but I find that in many cases my printer head rattles and jumps all over the place, not in all cases, just in some, and this I am working on. The problem would appear to be when using right justification. With no justify the printing appears to be smoother. Perhaps I have not hit on the correct driver for my printer. This column is being printed with FXB0DRAFT and the head is bouncing back and forth, the wear must be excessive. There are ample drivers to select from. I try to use that which works for me. You see, I have a choice. To be perfectly fair, I oft-times think the problem is more me than thee.

The manual is not the best, but then again it is by no means the worst. I feel that a little time spent with actual screen-display-on-page would be advantageous. With a little more detail as to how to achieve a certain object/target. The Commands and secondary commands are not too easily separated on the page, so what I did was draw a line down through the Commands, with another line down through the secondary commands, then the third commands. Makes for faster and easier reading.

My next step was to use text87 to make up the commands in columns, First, second etc, and quoting the page number. Printed out in small print, mounted on cardboard similar in thickness to the manual cover, this can be but-mounted to the back cover of the manual using cloth tape, and be used as a foldout of the commands. The way I did it has three foldout pages, and it lies flat in front of me as I work.

Talking about the Manual. This is a wire

bound set up, which lies flat at all times, and does not close up, or the pages curl up in use. You see, Software 87 have realised that the manual will get a lot of use, and have used a VERY heavy paper, almost light cardboard. Really a great idea, as this manual is going to get a lot of use, by me anyway.

At one time if I had been asked if I would purchase this program again, I would have said probably not. Now that I have persevered with it, and am gaining confidence, I will state without any reservation, yes I would buy it. It is not the easiest to learn, but once you start to really get into it, you find that you like it more and more.

There is so much flexibility about text87, I cannot imagine where the variations stop. It is not an editor or DeskTop publisher, but it comes close. There is only one way to assess what it does, you must try it for yourself. Oh sure, you will have to work at it, but the reward is worthy of the toil. And the more you use it, the more inquisitive you become about what it can't do.

Most folks do a little Quill-bashing when discussing word processing, not me. I like Quill, it is simple to use, easy to learn. Not exciting, but does a job. Sure it does have problems, but I still like it because of its simplicity. But, if you like excitement and exploring, you must try text87. If you like wide open spaces, and room to move around, versatility, control of how you want something done. Give me text87.

The main thing is to get to UNDERSTAND the programme, and what it is trying to do.

Yes it is true, I find myself using text87 more and more and more.

Where did I get this text87? Why from EMSOFT of course, and Peter Hale has been most encouraging with help and advice. Like the saying goes:- "A farmer is a man OUT STANDING in his field"

So also, is text87. OUTSTANDING.

FROM MEMBERS' LETTERS

One of our members, Steven Gunhouse, writes to me with such interesting material in his letters that I am excerpting some of it for our newsletter.

I have been attempting to save NMI-type programs from disk to tape, and mentioned difficulties I was having. The following is Steven's suggestions on how this might be done. Some of his ideas may prove useful in other applications. G.F.C.

Since I happen to have it in front of me, here is a piece of code that might be useful for your NMI-Save to tape copy program. I had written it for other purposes - part of my work on transferring files from AERCO to LKDOS - but you will find it easy to use. Essentially it is just a modification on the first part of MOVE .C1 to drop the headers and copy to the right place.

```

00: F3          DI
01: CD 62 00    CALL 0062      (ENABLE CARTRIDGE)
04: 21 XX XX    LD HL, XXXX   (wherever you have the name)
07: 11 22 20    LD DE, 2022
0A: 01 0A 00    LD BC, 000A   (LENGTH OF NAME)
0D: ED B0       LDIR
0F: AF         XOR A        (SET A TO 00)
10: 32 20 20    LD (2020), A
13: CD 84 00    CALL 0084     (FIND DIRECTORY ENTRY)
16: 3A 20 20    LD A, 2020
19: FE 0A       CP #0A        (FILE NOT FOUND FLAG)
1B: C8         RET Z        (RETURN IF NOT FOUND)
1C: 3E 0B       LD A, #0B
1E: 32 02 20    LD (2002), A
21: CD C6 00    CALL 00C6     (GET DIRECTORY ENTRY)
24: 21 45 20    LD HL, 2045   (FIRST ADDR OF TRACKS)
27: 7E         LD A, (HL)
28: FE F9       CP #F9
2A: C8         RET Z        (FILE EMPTY!)
2C: 22 YY YY    LD (YYYY), HL (POINTER STORAGE)
2F: 32 1D 20    LD (201D), A (TRACK TO LOAD)
32: CD 7E 00    CALL 007E     (SEEK TRACK)
35: CD 7B 00    CALL 007B     (READ TRACK)
38: 21 72 20    LD HL, 2072   (START OF NAME IN HEADER)
3B: 11 ZZ ZZ    LD DE, ZZZZ   (HEADER STORAGE)
3E: 01 16 00    LD BC, 0016   (LENGTH OF HEADER)
41: ED B0       LDIR
43: 21 88 20    LD HL, 2088   (START OF BLOCK)
46: ED 5B 7C 20 LD DE, (207C) (START ADDRESS IN HEADER)
4A: ED 4B 7E 20 LD BC, (207E) (BLOCK LENGTH IN HEADER)
4E: ED B0       LDIR
50: 2A YY YY    LD HL, (YYYY) (TRACK POINTER)
53: 23         INC HL
54: 7E         LD A, (HL)    (NEXT TRACK)
55: FE F9       CP #F9        (END FLAG)
57: 28 17       JR Z, +17     (IF DONE, TO ??70)
59: 32 39 20    LD (2039), A (STORAGE)
5C: 22 YY YY    LD (YYYY), HL (RE-STORE POINTER)
5F: CD 81 00    CALL 0081     (STEP TO NEXT TRACK)
62: 3A 39 20    LD A, (2039) (GET TRACK AGAIN)

```

```

65: 21 1D 20      LD HL, 201D
68: BE           CP (HL)
69: 20 F4        JR NZ, -0C      (NOT CORRECT TRACK - LOOP)
6B: CD 00 7B     CALL 007B      (READ TRACK)
6E: 18 D3        JR -2B         (AND REPEAT UNTIL DONE)
70: etc.         etc.          (set up for tape save, etc.)

```

There are three addresses you will have to choose to be some appropriate safe position. These are - 10 bytes for the name, 22 bytes for the disk header info, and 2 bytes for a pointer. Of course, you could use the same 22 bytes for name and block header. I actually left out the first 2 bytes of the block header, since you won't need the FF or the track number. You will need the 2 bytes right after the name (the start address) and the last 2 bytes of the header (the total length).

Naturally all of this does not care where the stack is, or whatever, so it will work in either Spectrum or 2068 mode. You will have to move the stack before you get into this routine regardless, something I have completely ignored. Sorry, change all of those RET Z commands to some appropriate branch to restore the stack.

For your purposes, the whole point of this routine is that it will completely load a file without trying to run it. Therefore it defeats the autorun aspect of an NMI-Save, and let's you go do your tape routine or whatever. My notes might also be useful. I decoded some of the LKDOS routines, and used some sort of appropriate description above.

I mentioned last time that if you did complete your NMI-Save to tape copier, you would still need a special loader to autorun the tape load. I don't have my information on the load routines in the Spectrum at hand, but I can tell you what will be involved.

As with the copy to tape program, the first thing you will have to do is put the stack in a safe place. This could be a major difficulty, however, since if you use the standard tape routines it will print to the screen whatever it finds on the tape. As usual, you have 2 choices. You could render the printing ineffective by defining every character to be a blank and setting OVER 1. Or you could simply re-write whatever parts you need and leave out the printing to the screen. Probably the easiest way is to presume the NMI-Save is immediately after the loader, so you won't need to look for it, so just load the header and code block directly.

The hard part, so to speak, is that once you succeed in loading it, you will have to reverse all the register saves that were done in the original NMI-Save routine. This is what LKDOS does automatically whenever it finds a file of length A826 (at least in version 2, that's how it knows). The routine it executes then consists of the following:

```

ED 7B FB 57 LD SP, (57FB)
F1          POP AF
C1          POP BC
D1          POP DE
E1          POP HL
D9          EXX
F1          POP AF

```

```

C1      POP BC
D1      POP DE
E1      POP HL
DD E1   POP IX
FD E1   POP IY
ED 7B FD 57 LD SP, (57FD)
F5      PUSH AF
3A FA 57 LD A, (57FA)
ED 47   LD I, A
FE 3F   CP #3F
28 02   JR Z, +02
ED 5E   INTMD2
3A FF 57 LD A, (57FF)
FE FF   CP #FF
20 01   JR NZ, +01
FB      EI
F1      POP AF
C9      RET

```

This is the routine from LKDOS at address 01DD (hex) adjusted for the presumption that it will not be running in the cartridge.

As you should be able to tell - or presume - naturally the interrupts will need to be disabled before you load from the tape.

Turned out to be rather a long P.S., didn't it. I happened to be examining the cartridge ROM, and recognized this routine for what it was, and thought you might need it. Then again, you might have already known where it was - the 01DD is of course for this version, but it is probably pretty near there in yours as well.

**ONE NEWSLETTER BOUNDUP BY TOM SKAPINSKI**

Corporate Systems Center, 730 North Pastoria Ave., Sunnyvale CA 94086

Selling: 720 K 3.5" Drives for \$38. ea., \$28. ea. in Quant. of 10

**PROFILE+5!**, Profile update, New Features. Available from RMG ENTERPRISES

1419 1/2 7th St., Oregon City, OR 97045 (Phone 503/655-7484)

Users manual Available from Tom Woods \$10.00 (ask RMG for Toms' address.)

UPDATE Computer Systems, Quarterly Magazine for SINCLAIR computers, covers disk drives for our machines, ... and is \$18.00 P.O. Box 1095, Peru IN 46970, ... Subscribe now for the upcoming year (starts next year with the issue due out after this current one) confused? ... write them

**FIRWARE 747** Flight Simulator for TS-2068 by Derek Ashton available from Bob Swoger, 613 Parkside Circle, Streamwood IL 60107

COMMODORE 1351 Mouse may be used with TS-2068 if you have a LARKEN disk drive controller, use the Larken joystick port and hold the Right button while powering up the TS-2068, I also heard the CoCo mouse also works

Try it with Artworks and Art Studio programs, I believe the price for Commodore mouse is about \$37.00...check prices before buying.

**WORDMASTER and PIXEL PRINT PRO**

ABCDEFGHIJKLM &  
 NOPQRSTUVWXYZ &  
 abcdefghijklmnopqrstu

ABCDEFGFG  
 12345678

OUTLINE  
 aboutl. cs

JULIUS CAESAR said  
 'omnia gallia est in  
 tres partes divisa', i.e.,  
 in English: All Gaul is  
 divided into three  
 parts.

*JC* CAESAR. C\$  
 MCMXC 24SEP

*ABCDEFGHIJKLMN O PQRST*

*UVWXYZ 1234567890*

*ABCDEFGHIJKL  
 MNOPQRSTU*

DATA italics  
 abDATi. CS

Light ABCDEFGHIJ  
 KLMNOPQRSTUVWXYZ  
 YZ1234567890 '&'  
 abcdefghijklmnopqrst  
 uvwxyz. *ABCDEFGHIJ  
 KLMNOPQRSTU  
 VWXYZ1234567890*

ABCDEFGHIJ  
 KLMNOPQRS  
 TUVWXYZ  
 ABCDEFGHIJKLM  
 NOPQRSTUVWXYZ  
 1234567890  
 STADIUM

THE UNITED STATES



Some notes on using Wordmaster  
 for its excellent graphics.

The Smith-Corona Fastext 80 printer lacks several crucial features needed to make WORDMASTER work properly, eg, Margin set; Printer reset; inadequate bit image graphics; to name a few. If your printer has similar drawbacks, all is not lost. Graphics can be made and saved, then imported into Pixel Print Professional. The examples above show excellent large fonts that can be used to enhance desk top publishing columns. Outline, Roman, Data, Light and Stadium are shown; Bold and italicized versions are optional. The U. S. A. map is from Steve Spalding's PIX collection. (RHM 900925)

# Hints For Timex/Sinclair Users

by G.A. Smith

The following tips, hints and tricks are for your T/S 1000 & ZX81 and the T/S 2068 color computers.

## T/S 2068 Hints And Tips

The T/S 2068 can use the bottom 2 lines also. You use: PRINT #0;, or PRINT #1;, or PRINT #0; AT 0,7, or PRINT #0; TAB 5;. The # symbol must be used to print on lines 22 and/or 23. A PRINT #0; AT 1,5;, will print on line 23, five spaces over. To clear the bottom two lines only, use the command INPUT "" without the semi-colon and without the variable.

If you miss the good old automatic scrolling on your T/S 2068, then here is a solution to your problem. Use POKE 23692,-1 or POKE 23692,255 to get an automatic scroll feature. You POKE 23692,1 to turn it off.

Do you have a little machine code program you would like to type into the T/S 2068 for your use, but don't know where the first available byte is located in your RAM? If you have a line number, then REM, the first available byte after that (and your start location for your machine code program) is 26715.

On the T/S 2068 you can "copy protect" your program by:

- 1 GOTO 3
- 2 NEW
- 3 program starts here.

Start your program by using RUN 9999, with that line reading:

```
9999 ON ERR GO TO 2 : SAVE  
"program's name" LINE 1
```

Did you accidentally "copy protect" a program and for some reason want to look at the LISTing? Well, you can load and stop any BASIC program by typing: MERGE "program name." The program will load into memory, but won't run automatically.

On the T/S 2068, if you want to duplicate a line in order to slightly change it, or even have it as another line number too, then backspace to the left of the line number. Put a single quote mark " there. Move your cursor to the right of the number, then rub out the number and replace it with the new one. This technique works with line number and every command and statement except for the "THEN" command. "THEN" just won't rub out!

## T/S 1000 And ZX81 Hints, Helps, And Tricks

Have you got 16K RAM or more hooked up to your ZX81 or T/S 1000 and you're still using the byte saving techniques you used with only 1K or 2K RAM? Techniques like:

```
10 LET T = 25  
20 LET Z = T  
30 LET V = T
```

Well, normally you will have more than enough memory to run your program. This space saving technique, though necessary for computers with limited memory, is not needed for the 16K RAM and larger memory machines.

"If I can save a byte, why not?" you ask. Though you save a byte or two, you lose program execution speed! Your programs will run faster if you do not use the byte-saver techniques.

If you want to check to see how much RAM your program is using in your ZX81 or T/S 1000 then type:

```
PRINT (PEEK 16404 + 256*PEEK  
16405)-16509
```

Would you like to use the bottom two lines of your display? POKE 16418,0 allows you to use the bottom two lines. Be sure to POKE 16418,2 before you use INPUT or SCROLL. If you don't, then you'll have a nasty display CRASH, plus you will have to reboot!

If you have a line number 0 (zero) you cannot accidentally delete, erase, or replace it. You can have a line number zero by:

(A) enter your first line as usual, or load your program.

(B) POKE 16510,0. Your first line is now zero.

(You can still POKE into line number zero.)

Try this! Enter two or three lines, REMark lines will be alright. POKE 16509,n (let n = any number from 40 to 60). Now list your program. Look what happened to the last line you entered! Enter another line. List the program again. Your "funny" numbered line moved! This technique is great for putting information in your program that you don't want deleted or erased.

ZX81 and T/S 1000 users, if you want to show off for all of your friends, instead of using the COPY command, use

PRINT USR 2153 to get the same results!

## Screen Control For The T/S 1000 And ZX81

You want to control the screen for your program, right? In this section, you will learn how to put the computer into slow, clear the screen, then fill it with whatever character you want to display, all using super fast assembly language. Also, as part of screen control, you will have machine code routines to scroll the screen up, down, left, or right!

The T/S 1000 and ZX81 don't allow DATA statements. By carefully studying the machine code loader program you will learn how to simulate READ/DATA statements!

You'll need to make your first program line a REM statement line with as many characters in it as the assembly routine you plan to use. At the beginning of the program area is where machine code is stored for our purposes. This REM statement line will need 24 characters in it. Or you could use the formula PRINT LEN A\$/2 to find out how many to use.

First, the machine code program sets the computer to the "slow" mode, clears the screen, then fills the screen with inverse spaces. You may change this character and I'll show you how.

Let's get the "machine code loader program" into our computer. Your machine code loader program is as follows:

```
10 REM 12345678901234567890-  
123456789012  
20 LET A$ = "CD280FCD09863E-  
80061720D70DC2884005C28640C9"  
30 LET A = 16514  
40 FOR Q = 1 TO LEN A$-1 STEP 2  
50 POKE A, 16*CODE A$(Q) +  
CODE A$(Q+1)-476  
60 LET A = A + 1  
70 NEXT Q
```

Line 20 is your "DATA" statement line. As much as I hate HEX numbers, I've had to use them for this application. Some of the numbers are single, double, and even triple digits, POKE VAL(A\$(I)) can't be used.

The string assigned earlier to A\$ is the machine language routine. The CD280f calls the slow routine from the ROM. The CD0986 calls the "CLS" screen routine from the ROM. The rest of the routine fills the screen with the

inverse space. If you want to fill it with the period, POKE 16521, 23. On POKEs, use the decimal value, never HEX. POKE the code of the character you want to fill the screen into 16521 then "call" the routine. (Use any of the methods shown.)

In your programs use PRINT USR 16514, RANDOMIZE USR 16514, or LET A = USR 16514 to call the routine. All will give you the same results. Just one word of caution though, if you are using the variable A in your program, then one of the other methods may be better suited for your program since the value of A will be changed.

Would you like to know what character is at the next print position? Here is a routine that you can use to do just that. Let line number 20 be A\$ = "2A-0E404E06C9" and use the machine code loader program.

```
Have a program with lines:
PRINT AT X,Y;
A = USR 16514
```

[PRINT LEN A\$/2 will tell you how many characters this routine needs. You have to adjust the address (16514) if you use previous screen fill routine. If you did use it, then your REM statement would have to have 36 characters and use the following as A\$:

```
A$ = "CD280FCD09863E800617-
20D70DC2884005C28640C92A0E4-
04E06C9" you'd use USR 16533 to ac-
cess the next place routine.]
```

The PRINT AT X,Y; (use your variables for "X" and "Y") is needed. Be sure to put the semi-colon at the end of the "PRINT" statement! The variable "A" will be the CODE of the character at position X,Y on your screen. Your program will then decide if the next character is a "safe" spot, or whatever.

You've got this great idea for a game, right? The only thing holding you back from writing it, is the fact you can't move the screen sideways, or up, or down. Well, here are four programs to help with just that!

You'll need to make your first line a REM statement line with as many characters in it as the program you plan to use. The up-scroll uses 24 bytes, the down-scroll uses 29 bytes, the right-scroll uses 21 bytes, and the left-scroll uses 26 bytes. In case your solar powered abacus is on the fritz, that's a grand total of 100 characters. You could tell your computer to: PRINT 24 + 29 + 21 + 26 to get the same answer. Try it!

Use the previous machine code loader program to convert the REM line into assembly language.

I generally use PRINT USR to access the machine code routine. That doesn't mean that you cannot use LET A = USR 16514, or RANDOMIZE USR 16514. One works just as well as the other. However, when you use a routine that is to return a value to your BASIC program (in this case the variable A), such as the next print position routine, then you will have to use the form: LET A = USR 16514 to get the value into your variable so your BASIC program can work with it. Of course, this USR address "call" is not 16514; that address was used only as an example. Each step of the way I'll tell you the correct addresses to use for each assembly routine.

Scroll up:

```
A$ = "2A0C4023E511210019D101B5-
02EDB0EB062036002310FBC9"
```

Scroll down:

```
A$ = "2A1040114300ED52E511210-
0ED52D101B502EDB8EB06202B36-
0010FBC9"
```

Scroll right:

```
A$ = "2A0C400616C506203E00234-
F7E7110FA23C110F1C9"
```

Scroll left:

```
A$ = "2A1040114300ED520616C50-
6203E002B4F7E7110FA2BC110F1C9"
```

You can use any, or all, of these routines to move the screen as you wish. But remember to put enough characters in your REM line and to use the correct address for the routine you want. If you were to use all four, then:

PRINT USR 16514 would scroll the screen up one line.

PRINT USR 16538 would scroll the screen down one line.

PRINT USR 16567 would scroll the screen right one space.

PRINT USR 16588 would scroll the screen left one space.

If you have these four routines, then tack on the screen filler at the end and next print position routines, you'll have 136 characters in the REM statement line. The screen filler would be accessed by: PRINT USR 16614 and the next print position routine will be LET A = USR 16621.

With these programs in place, you should be able to do just about anything you want to with the screen on your T/S 1000 or ZX81. Now, can you come up with the applications to use them.

May you have many hours of happy computing with your T/S 1000 or ZX81. ●

Press Enter BBS SOFTWARE BY LARKEN ELECTRONICS

SIR CLIVE'S CASTLE BBS

TIMEX/SINCLAIR	
Q W E R T Y U I O P	LKDOS
A S D F G H J K L RET	
Z X C V B N M S SP	

NOTE: (613)-745-8838 - Ed.

NOW RUNNING AT 1200 AND 300 BAUD

Ctrl-S to Pause, Ctrl-Q to Restart, (Cap Shift) Z to Abort

W E L C O M E !

Special Welcome New Members: Rod Humphreys, Robert Cazares,  
Greg Popovich, Charlie Day,  
and Bruno Gagnon!

Sir Clive's Castle BBS is dedicated to all serious users of Sinclair computers and provides upload and download facilities for the Timex/Sinclair 2068, ZX Spectrum emulating Timex/Sinclair 2068 and the Timex/Sinclair 2068 running the Zebra OS-64 Column Board.

Your sysops are: Michael Dove -- BBS Hardware And David Solly -- SIG's & Library  
This BBS operates free of charge, however; complete access is restricted to registered and verified members only. Membership applications and enquiries should be made at the Sysop's Message Base or in writing to: Sysop, Sir Clive's Castle BBS, 1402-1545 Alta Vista Drive, Ottawa, Ontario, Canada. K1G 3P4

Hours of Operation Sir Clive's Castle BBS is in operation 24 Hours A Day

EBS Software: LARKEN MaxBBS for the Timex/Sinclair 2068.

Supplied by Larry Kenny of LARKEN Electronics

RR #2, Navan, Ontario

Canada K1B 1H9

Tel: (613)835-2680

20 June 1990

SYSTEM NEWS AND WHO FILES

Two information files are now available on-line and may be read from the Main Menu by using the "(P)rint" command or downloaded by using the (D)ownload command.

These files are: NEWS.CT: System news, Bulletins and Announcements;  
and, WHO.CT: The current Sir Clive's Castle BBS User/Membership list.  
OPEN ACCESS =====For a short time I shall be opening the system with a special log-on and password so that all callers may have access to the messages bases and the up and download area.

When asked for your name type: Special Guest

When asked for your password type: Clive

If you have already been assigned a log-on and password, please use it. If you like this BBS and plan to be a regular caller, please apply for an individual password which will allow me to identify callers and to keep the stats necessary for efficient running of the system.

How long this BBS remains open and whether this becomes a permanent set-up depends solely on the conduct of the callers -- I sincerely hope that that is enough said on the topic.

Meanwhile, Enjoy your visit. -- David Solly, Sysop 20 August 1990.

User list: BOULT, MARCEL\BRANDON, ALVARO\CAZARES, ROBERT\DAVIS, MIKE\DAY,  
CHARLES\DOVE, MICHAEL (SYSOP)\DOVE, SYDE\FOSS, LEO\GAGNON, BRUNO\GAGNON,  
REAL\HARMER, BILL\HILL, PAUL\HOLDEN, CLEM\HUMPHRES, ROD\KENNY, LARRY\KENNY,

ORVILLE\MATHEWSON, JOHN\MOORE, ALAN\MULDER, ROELOF\NATHANSON, MEL\PATERSON,  
SCOTT\POPOVITCH, GREG\RAYMON-JONES, COLIN\RIDGEWAY, CHUCK\SAMPSON, JOHN\SCHIEMAN,  
JOHN\SCHIMKE, ROBERT\SHADE, ROBERT\SOLLY, DAVID (SYSOP)\TEN HOLDER, IRIS\WARD, TIM  
L.\YEO, LORETTA

#### File Library Help

---

An option called "File (L)ibrary" has been added to the help files which may be reached by typing "H" at the Main Menu. This option prints a detailed listing of all the files available for downloading at the time of posting. The file itself may also be downloaded by requesting file: UD.CT and then printed using a word processor such as M-Script. The Files message base should still be consulted for the most recent uploads. An additional file, (Libr.Cv), is a Vu-File data file of all the public domain software available from Sir Clive's Castle BBS but may not be on the Upload/Download disk.

#### Pascal Programming SIG

---

David Solly, your friendly sysop at Sir Clive's Castle BBS, would like to create a Pascal Programming Special Interest Group for any of you who own HiSoft Pascal for the Timex/Sinclair 2068 or for the ZX Spectrum. The SIG will focus on how to best to exploit the HiSoft Pascal package, programming problems specific to HiSoft Pascal, and Pascal programming itself. Those of you who are interested may leave a message at the Sysop's Message Area or in writing to: Sysop, Sir Clive's Castle BBS, 1402-1545 Alta Vista Drive, Ottawa, Ontario, Canada, K1G 3P4

A reminder that there are a number of Pascal source code files available on the Up/Download disk. Pascal files can be recognized by their ".PS" extension.

#### TS 1000 File Transfers

---

The sysops would like to welcome any of you who have called the system using the Timex/Sinclair 1000 or ZX81. We regret that at the moment we can not supply you with file transfer facilities and would like to make a special request that you do not attempt to do file transfers using the unmodified version of Wymil Mini-X-Mode. We have tested the Wymil Mini-X-Mode package extensively on this BBS and have found that not only will it not transfer files but is more than likely to cause a system crash.

We have also had little luck with Fred Nachbaur's ZX-Term\*80 modem package which uses a high resolution SCRAM board. Although we have been able to get programs to upload using ZX-Term\*80, we have not been able to download anything. If any of you know of a fix for either one of the above packages or know of a new modem package for the TS 1000 we would like to hear from you.

#### UPDATE! MAGAZINE

Update! Magazine is still alive and well although Bill Jones has stepped down as publisher. The new publishers are Frank and Carol Davis.

Subscriptions to Update! are US\$18 and should be mailed to:

Update Magazine  
P.O. Box 1095  
Peru, In  
USA 46970

This is about the last of the great Timex/Sinclair magazines, folks! When it is gone there will be no more so show Frank and Carol your support by subscribing and mailing in those articles and programs.

---

*Taken from the Oct 1990 issue of SWYM, the newsletter of the  
Seattle Area T/S Users Group.*

# The Québec Link

by:  
Réal Gagnon  
8286 St-Hubert  
Montréal (Qué)  
CANADA H2P 1Z3

## Default device on the QL

Say you are writing a program and it is important to know from which device the user is running it. What can you do?

If the user's QL is equipped with TOOLKIT II, then it is possible to set the default devices for the program and the data and retrieve them with the functions PROCD\$ and DATAD\$. But if the user don't set default devices then we are out of luck.

One solution to this problem is to look in the system variable area to determine from which device the QL was booted. The following function, BOOT\_DEVICE\$, will retrieve that information.

```
DEFine FuNction BOOT_DEVICE$
  pointer1=PEEK_L(164096)+16
  pointer2=PEEK_L(pointer1)+36
  length=PEEK_W(pointer2)
  b$=""
  FOR i=0 to length
    b$b$&CHR$(PEEK(i+length+pointer2))
  END FOR
  RETURN b$&"1_"
END DEFine
```

The first variable, POINTER1, contains SU\_FSDEF which give us the address of the file system physical definition. From this address, we read the variable POINTER2, called FS.NMLEN in the QL TECHNICAL GUIDE. This word gives us the length of the boot device name. From that point, all we have to do is to read the device name with the PEEK function.

Type in the function BOOT\_DEVICE\$ and try the following command to test it:

```
PRINT "QL booted from ";BOOT_DEVICE$
```

For example, the BOOT\_DEVICE\$ function can be used in a menu program to detect from which device the QL was booted from.

## QL Tip

To load a machine code file in memory, we have to proceed in 3 steps:

1. reserve some RAM with RESPR
2. load the file with LBYTES
3. execute the machine code with CALL

For example, we have a game called YOYO, which represents 5000 bytes. To start the game, we type:

```
a=RESPR(5000)
LBYTES mdv1_YOYO,a
CALL a
```

It is possible to do the same thing with only 1 or 2 commands.

```
-with TOOLKIT II:
  LRESPR mdv1_YOYO
-without TKII:
  LBYTES mdv1_YOYO,RESPR(5000):
  CALL RESPR(0)
```

that way we don't have to declare any variable and it is faster to type.

## Special effects with CSIZE

The appearance is important, even with software. Let's see some techniques to make the look of our programs more pretty.

The command CSIZE is used to change the character size. The syntax is simple:

```
CSIZE [channel],width,height
```

The width parameter can be 0,1,2 or 3. The height parameter can be 0 or 1.

When used alone, CSIZE is boring but with a judicious combination of INK and PAPER the result can be quite interesting. Try this:

```
OPEN#5,scr_200x30a195x25
PAPER#5,0: INK#5,7: BORDER#5,1,7
CLS#5
CSIZE#5,2,1: INK#5,49
PRINT#5,"The Quebec Link"
```

Ok, now if we use the OVER function and the CURSOR command, the result is a totally unexpected and very good looking!

```
OPEN#5,scr_200x30a195x25
PAPER #5,2 : INK#5,7 : BORDER#5,1,7
CSIZE#5,2,1: OVER#5,1: CLS#5
test$="The Quebec Link"
CURSOR#5,1,1 : PRINT#5,test$;
CURSOR#5,2,2 : PRINT#5,test$;
OVER#5,-1
CURSOR#5,1,1 : PRINT#5,test$;
CURSOR#5,2,2 : PRINT#5,test$;
CSIZE#5,1,0
```

Try to experiment with different PAPER, INK and CSIZE values.

JSU SCREEN MODES

The screen problem reported by Robin Beaumont in the July/Aug '90 issue of Sinc-link is really due to deficiencies in the North American NTSC TV system. If you have ever visited Europe you will probably have noticed that the quality of the TV pictures on quite ordinary sets is as good as you get over here on a premium model like the Sony Trinitron series. This is due to better colour handling (PAL) and the fact that there are 625 instead of 525 lines per screen, and it is the latter that causes Robin's problem (and anyone else's who uses a QL with an NTSC TV!). The difference does carry over into the computer world too since the basic IBM screen is 200 lines against 256 for the QL or Acorn BBC - but most monitors can cope, with a bit of vertical size adjustment.

There are differences between the PAL TV and monitor mode, but they are minor compared with those on the JSU chip for the NTSC TV mode. (The monitor mode is the same on both.) In PAL TV mode the top 16 lines and 32 pixels down each side are omitted since they are not normally visible on a TV screen, but this is just done by choosing appropriate WINDOW sizes, and using larger characters (CSIZE 2,0). In line terms this means that there are 25 lines on a monitor (with six spare lines) and 24 in TV mode since the characters are 10 pixels high. The screen memory is mapped in the same way for MON and TV, and even if you have pressed F2 on boot up you can access the full screen using the WINDOW and CSIZE commands. Many commercial programs seem to use the full screen and try to avoid putting anything too important around the edges where TV users might not see it. (The small 0,0 characters must be reasonable legible on most PAL TVs.)

However, the situation is much more complicated with a JSU ROM. In NTSC TV mode there are only 192 lines available. So, so as to keep a 24 line screen the TV mode uses characters only 8 pixels high, omitting the two blank lines of pixels between characters that normally appear in monitor mode, making the text rather cramped. In fact the JSU character set is different from all the others to make this easier, descenders are only one pixel instead of two, and underlining runs over them. In addition the memory is mapped onto the screen differently in TV and monitor modes and you cannot restore the full monitor screen after pressing F2; you can get the full width, but not the height. This means that programs not specially written to take this into account behave in the way that Robin describes, only the top 192 lines are visible, the bottom 64 cannot be seen. In my experience the Psion programs that come with the machine are the

only commercial ones that allow for the peculiarities of the JSU ROM, although it is not that difficult to write programs that can adapt to all three formats. (All the major ones I have written do!)

As long as you are not in two screen mode with a MINERVA ROM then PEEK(163890) returns the value of a system variable SV.TVMOD that tells you the mode: 0 for monitor, 1 for PAL TV and 2 for NTSC TV, and you can make use of this. (Unfortunately there is a bug in the JM ROM which corrupts SV.TVMOD when the MODE command is used, but that's another story, and you can code around it.) The procedure in Listing 1 is an example of the sort of thing I mean. It defaults to the PAL TV mode and defines global variables mon, pal, ntsc, and hc% that can be used elsewhere in your program. (Make all variables LOCAL unless there is a good reason for not doing so!) A way to avoid the JM bug is also included. N.B. these are NOT Sir Clive's default windows.

The variables mon, pal ntsc and hc% are made global because they are sometimes useful elsewhere in a program.

There is another screen problem with the JSU ROM, the SCALEd graphics commands like LINE, CIRCLE etc. use a different vertical to horizontal ratio from all other ROMs. This means that graphics mixed with text, or commands like BLOCK which address pixels directly may not come out in the right place. (In case anyone needs to know the values are 0.62659 on the JSU and 0.7380 for the others, you can adapt programs to the JSU if you multiply all X parameters in the affected commands by the ratio of these two numbers - I forget which way up it has to be, but try 0.8490 it will soon become obvious if it should be the other way! Tedious, but it does work; I have converted a couple of programs.) Get in touch with me if you want more information.

Vertical hold and flicker problems are due to the 60/50 Hz difference. As far as I can see in monitor mode the JSU chip thinks it is in Europe and uses 50 Hz. My Amdek 300A monitor has a long persistence phosphor, and is rock steady, but cheaper monitors and TVs using video input may require vertical hold adjustment and may give an unpleasant 10 Hz flicker. Since TV mode is OK this must be 60 Hz.

Howard Clase, Tel: (709)-753-6415  
Box 9947, Station B,  
St John's,  
Newfoundland,  
CANADA, A1A 4L4.

Listing 1

```
10 REMark ~~~~~ h. j. clase 1986.10.11 ~~~~~
12 DEFine PROCedure ~~~~~ Uwindows
14 REMark Adjust windows for monitor or TV (European or N. American)
16 REMark 20 lines in windows #1 & #2. hc% = height of characters
18 LOCAL i,f,w%,x%,y%: mon=0: pal=0: ntsc=0: REMark GLOBAL VARIABLES
20 WINDOW 512,256,0,0: PAPER 0: CLS : REMark -----
22 f=PEEK (163890): SELEct ON f
24 =0: w%=512: hc%=10: x%= 0: y%= 0: mon=1
26 =2: w%=448: hc%= 8: x%=32: y%= 0: ntsc=1
28 =REMAINDER : w%=448: hc%=10: x%=32: y%=16: pal=1
30 END SELEct : Moad 8*(NOT mon)
32 WINDOW#2,w%,hc%*20+2*mon,x%,y%: PAPER#2,4: INK#2,0
34 WINDOW#0,w%,hc%*(5-(NOT mon)),x%,y%+hc%*20+6*mon: PAPER#0,2: INK#0,6
36 WINDOW#1,w%,hc%*20+2*mon,x%,y%: PAPER 0: INK 4:
38 FOR i=0,2,1: BORDER#i,mon*(NOT(NOT i)),0,6: CLS#i: CSIZE#i,0,0
40 END DEFine Uwindows
42 REMark ~~~~~
44 DEFine PROCedure ~~~~~ Moad(n%)
46 REMark A nasty pun to avoid the JM mode bug!
48 LOCAL v: v=PEEK(163890): MODE n%: POKE 163890,v
50 END DEFine Moad
```



# Pixel Print Plus! The T52068 Desktop Publisher.

<<<< MENU SELECTIONS >>>>		<<<< EDITOR FUNCTIONS >>>>						
(K) KEEP-Save/keep a doodle pad. (S) SAVE-PIXEL PRINT column. (L) LOAD-PIXEL PRINT column. (I) ICON-LOAD into a doodle pad. (G) GRAPHICS-LOAD/SAVE SCREENS. (W) UNDO-replace present doodle pad with last 'KEEP'. (W) WIDE-select character width. (H) HIGH-select character height. (C) CLS-Clear display/doodle pad. (M) MOVE-Scroll UP 'T' or DOWN 'B' in three speeds. Speeds 2 & 3 use CS or SS with 'T' or 'B' key. (P) PRINT-to 2040 or full size Printer.	(F) FONT-activates FONT MENU. Select T-TIMEX, B-BOLD, M-MODERN, I-ITALICS, S-SPACE, K-KERN, R-RESTORE, or L-LOAD FONTS. (B)-BLOCK FUNCTION MENU. C-COPY or E-ERASE a segment of a column. I-INSERT or D-DELETE pixel row at the top of the cursor. (X)-shifts to the RIGHT or LEFT column. (Q)-QUIT or terminates the Program.	(CURSOR CONTROL)"CAPS SHIFT"+5, 6, 7, or 8 keys to move CURSOR character position LEFT, DOWN, UP, & RIGHT respectively. (CURSOR SHIFT)"SYMBOL SHIFT"+Q(←), F(↑), G(→), E(↓) to SHIFT the CURSOR 1 pixel LEFT, DOWN, UP, and RIGHT respectively. (DELETE)"CAPS SHIFT"+0 step the the CURSOR backward, erasing the LAST character. (UNDERLINE)"SYMBOL SHIFT"+0 toggles the UNDERSCORE ON/OFF. (INVERSE)"CAPS SHIFT"+4 toggles inverse video ON/OFF.	(OVER)"CAPS SHIFT"+3 toggles the OVER type function ON/OFF. (SEGMENT SHIFT)"SYMBOL SHIFT"+Y(AND), D(DSTEP), S(NOT), or U(OR) shifts sections of the doodle pad (screen) LEFT, DOWN, UP, or RIGHT respectively. (CAPS ON/OFF)"CAPS SHIFT"+2 toggles upper & lower case characters. (EDIT EXIT)"CAPS SHIFT"+1 exits the EDIT mode & returns to the MAIN MENU.					
EDIT MODE EXIT CS+1	CAPS LOCK ON / OFF CS+2	OVER ON / OFF CS+3	INVERSE VIDEO ON / OFF CS+4	CURSOR CONTROL ← CS+5	CURSOR CONTROL ↓ CS+6	CURSOR CONTROL → CS+7	CURSOR CONTROL ↑ CS+8	DELETE UNDERLINE CS+0/SS+0
CUR SHIFT BY PIXEL+	CUR SHIFT BY PIXEL+	CUR SHIFT BY PIXEL+	CUR SHIFT BY PIXEL+	CUR SHIFT BY PIXEL+	CUR SHIFT BY PIXEL+	CUR SHIFT BY PIXEL+	CUR SHIFT BY PIXEL+	CUR SHIFT BY PIXEL+
SEG SHIFT BY PIXEL+	SEG SHIFT BY PIXEL+	SEG SHIFT BY PIXEL+	SEG SHIFT BY PIXEL+	SEG SHIFT BY PIXEL+	SEG SHIFT BY PIXEL+	SEG SHIFT BY PIXEL+	SEG SHIFT BY PIXEL+	SEG SHIFT BY PIXEL+
KEY B OVERLAY A R D								

needs 103%

Game Review - "King's Keep"

by Jeff Taylor

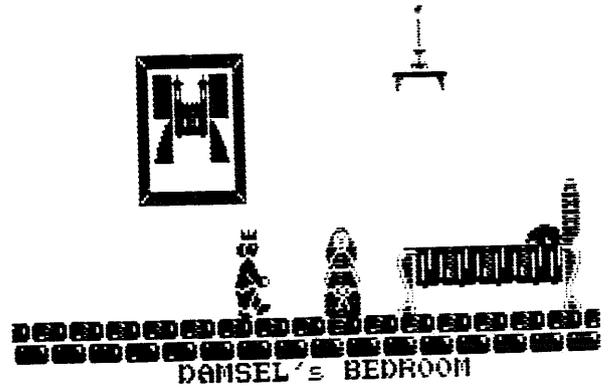
Most adventure games for the Spectrum involve looking at a non-moving picture and keying in words until you finally hit upon the one which will allow you to move to the next screen. This can lead to frustration or boredom very quickly. Now try to imagine an adventure game which allows you to move around freely using the joystick, picking up and using objects you encounter along the way. A much more enjoyable form of game.

*King's Keep* casts you as the young prince trying to escape from your father's castle. You are allowed to wander throughout the castle acquiring and using whatever you find in order to aid in your escape. Each room contains either an object or a person which will help you if you utilize them properly. Many of the objects' uses are not immediately evident so some thought is required, a feature which should appeal to the purists. There are four rooms to which entrance is forbidden until you cast a spell to unlock them. These rooms contain the final elements you need to effect your escape. Entering these rooms before you assemble the clues needed to cast the spell causes you to get caught and the game restarts, so don't try it!

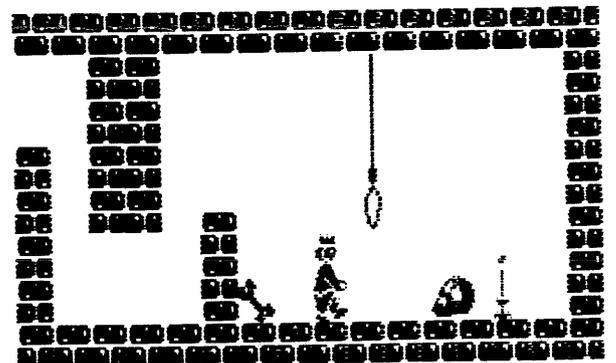
The game can be played using the keyboard exclusively or you have the option of using the joystick ( Kempston or Larken interface ) for movement and keys 1 through 0 for object manipulation. An interesting feature is the very effective use of drop-down windows to provide information.

TS2068 owners can play this game as well, however you will have to use the keyboard to move the prince unless you have a Larken disc driver with its built-in Kempston interface.

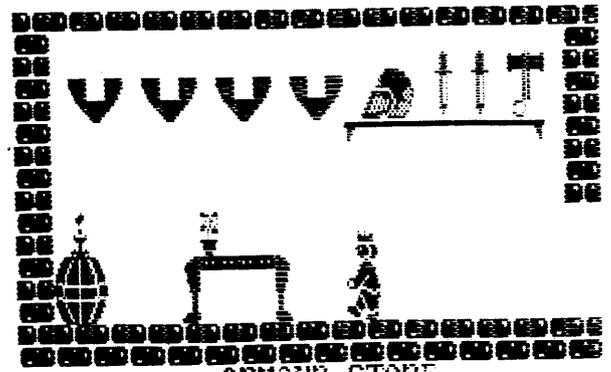
*King's Keep* is an enjoyable, sometimes infuriating, action adventure game which will keep your interest for days not just a few minutes. By the way, if you do get stuck, club secretary George Chambers has the solution to each clue but he will only give you a couple at a time. Just to keep you guessing.



The DAMSEL says -  
"Oh good! You have found my lucky charm! Have this shawl as a token of my gratitude."



The scroll has a message written on it - "To cast this sleep spell the user must be in possession of a ticklish object and a small coin."



Ouch! The heat from the candle is burning your feet!

As promised last month, I'll talk about how C accesses files.

The simplest way to discuss how C handles files is to take a look at a C program that uses it. Look at program number 5. This program asks the user for a file name. It then opens the file, reads in a character and prints it out the screen. This goes on for the entire file.

The program declares a character array of size 20 to be hold the file name. It then declares the integer fd to hold the file ID. When a file is opened, C assigns a number to it. This number is returned from the fopen command to the variable fd.

Note the syntax of the fopen command. The file name is sent into the function along with a string that specifies if the file is for reading or writing. If we wanted to open a file to write to, we would have used "w" instead of "r".

The next couple of lines checks to see if the file was opened sucessfully. If the file was not opened (there was some error like a bad file name) fd will be returned as NULL (a predefined constant).

Instead of using getch the function getc is used. Where getch defaults to the keyboard, getc needs a file ID to know where to get it's input. In our case we use fd.

We exit the while loop when the program reaches the end of the file (EOF). After the program exits the while loop, we use the fclose function to close the file. We must tell fclose which file to close. If we do not close the file, we may loose the file.

```
/* prog 5 */
#include <stdio_h>

main() {
    char file[20];
    int c, fd;

    printf("Enter file: ");
    gets(file);

    fd=fopen(file,"r");
    if (fd == NULL) {
        printf("File open Error");
        abort(1);
    }
    while ( ( c = getc(fd) ) != EOF) {
        putchar(c);
    }
    fclose(fd);
}
```

**BEGINNERS DON'T KNOW WHAT**  
---(it is)--- **THEY DON'T KNOW**

Recently I read a statement to the effect that "Beginners don't know what they don't know". And after I re-read the statement, I sat back and thought about it, and when I applied the statement to my own situation, I realised just how true it was, and is.

In my own case I started into computing fairly recently, and had the usual problems, both with the QL and information. When I asked for help, no one could help me. Why? because they did not know what I wanted to know. Why? because I did not know what I wanted to know; I did now know what to ask for, except HELP. So we were back where we started.

Then someone suggested to me that I start a QL library in the club, this I did.

The programmes started to come in, and I was flummoxed, bamboozled, dizzy, you name it, I was it. Gradually in the process of trying out all the stuff which was coming my way, I learned a lot.

I learned how to get 'into' the various types of programmes and utilities. What could and could not be accomplished once I was there; I learned that the way to computer literacy was not an easy one, or one which could be learned easily. Rather an ongoing process where one obstacle was overcome only to have another pop up right in front of you.

When I first entertained the idea of owning a computer, it was as a tool for record keeping, later I found that this was just one of the uses they had. I also found my interests were changing; that there were many more facets to be considered.

Because of the amount of material being passed to me, and the fact that I was DOING something, made me more critical of my actions and considerations. I had to collect the programmes, put them in

some sort of order, and if I were to make a catalogue, I had to know a little of what the programme did, so that a thumb-nail description could be made to tell others the intent of the programme.

Soon I realised I had to have some form of displaying the material in catalogue form. This made me look at the Psion set as supplied, and of course I was absolutely forced to work with them. I may not have chosen the best way to do it, but I formulated an acceptable form of presentation. To me at any rate.

The whole exercise meant that I had to work, and work hard at that. But the end result was I had a working knowledge of many things.

I also realise, now, that 'way back then as a beginner, I did not KNOW what I WANTED, or NEEDED to know.

Therefore, I would advise all beginners out there, to keep at it. Try all the wonderful stuff there is going around. Try each and every one of the Psion set. Learn by hands-on experience.

Get the whole darned library and do as I did, try to make sense out of it all. Apply it to your own uses. In the process you will find how to do things....and all the assistance you could dream of.....is there, just for the asking.

Subscribe to "UPDATE", probably the best magazine in North America, for information for beginners and experts alike. (Apart from our own NewsLetter of course) Learn from the experts.

I had a slight advantage I must admit, I had to go through a lot of stuff which was duplicated under a different title, was faulty in some form or another. In other words I had to be critical. I was, and I learned.

We are only beginners because we are beginning, and we "Don't know WHAT we don't know"

Hugh H. Howie.

Q L

## HARDWARE LIBRARY

### DO WE NEED A HARDWARE LIBRARY ?

#### IS THERE A DESIRE FOR ONE ?

For some time I have been pondering if there is enough interest to form a QL Hardware Library. If so, I would be happy to try and set one up. How do you folks out there feel about it?

My perception is that it could be set up in a similar manner to the present QL Software library, and would be part of that library.

All material submitted should be accompanied with adequate printed text, explaining the whys and wherefors. There would have to be a clear and precise legible diagram with a list of all parts used, and with values clearly indicated. Too often when I read some of those articles I find that the clarity is far from what it should be.

As I am no Hardware Hack, I would not be able to assess the value or accuracy of anything submitted. All inquiries for further information would have to be directed to the person submitting the article. This would mean that the documentation would have to contain the name and address of the donor/author.

Owing to the fact that there would be a considerable amount of Photo Copy involved, and travelling to and from such place, there would have to be a charge for this service. It must be remembered that the equipment we use is our own. This applies to our secretary and our Editor, to every one of the Committee, we have to look after it and are responsible for the repairs thereto ourselves. Our services are voluntary.

My suggestion is a charge of \$1 for this service. (per project) as the library would all be on paper.

Submissions would in some cases, who knows, perhaps in all cases, be used in the Newsletter, as and when space permitted.

As I stated earlier, the QL Hardware Library would become part and parcel of the present QL Software Library, and the contents would be listed in that catalogue.

There are a lot of projects out there, and I think it would be a good idea for a clearing house for them. Perhaps I am in error, but who is to know unless an attempt is made to gather them and make them available to all.

Comments (and projects) will be welcome.

Hugh H. Howie  
QL Librarian  
586 Oneida Drive  
Burlington. Ont.  
L7T 3V3

#### PEEKs \_\_\_\_\_ PEEKs

With TK2, if you want FREE\_MEM you type "Print Free\_mem" ( 85504 )

If you type "Print PEEK\_L(163856)-PEEK\_L(163852) you will get a different answer. ( 86016 )

BUT...Where did the 512 come from?

With 512K added the figures are 600064 & 600576. A difference of 512K.

With a program loaded the figures are 371712 & 372736. A difference of 1024 !

By the way, this is an easy way to find Free\_Mem if you do not have TK2.

H. H. H.



Instructions for sending in computer equipment:

1. For repairs, please use a separate sheet of paper to describe in detail the problem you are encountering, and whether or not the problem is intermittent. List any software or hardware that are associated with the problem. Also list any modifications that have been done to your equipment.
2. For upgrades, please enclose or specify the magazine article or other source of information for doing each upgrade.
3. You may include a check or money order as a deposit for repair costs. You will be notified if a balance is due. Over amounts will be refunded.
4. Carefully pack and ship your equipment to the address below via UPS or parcel post. UPS is usually cheaper, especially with heavy items.

DAN ELLIOTT  
RT 1, BOX 117  
CABOOL, MO 65689

Phone (314) 739-1712 evenings, Sunday through Thursday till 10 PM Central.  
Phone (417) 469-4571 Saturdays by chance till 11 PM Central time.  
VISA / MASTERCARD accepted with 4% surcharge.  
(Charge customers must provide Acct. #, expiration date, and name on card.)

WHERE CAN YOU GET TS1000 SUPPLIES?

Just recently a young lady from California sent an envelope to ISTUG wanting a TS1000 catalog. I had to let her know we are a user group, not a seller of software and hardware. I did send her a small list of places to look for these and hope this will be of help. What this brought to mind is that it would be a good idea to publish a short list of places to look for software and other items for the TS1000, ZX81 and the TS1500.

- (1) Charles Ridgeway -for software  
2816 Chestnut  
San Angelo, TX 76901
- (2) SMUG -has a fair amount of stuff from Zebras stock  
P.O. Box 101  
Butler, WI 53007
- (3) RMG -has most anything you could want  
1419 1/2 7th Street  
Oregon City, OR 97045
- (4) EMSOft -software and connectors  
P.O. Box 8763  
Boston, MA 02114

Let us know of any others out there that are still doing some business with the TS1000 and we will try to get the word out to the users.

Bob Mitchell  
20 Wild Briarway  
Willowdale Ontario  
M2J 2L2  
901117

Dear Les,

If you load the AUTOSTART on the enclosed disk and then read the help file you'll get the drift of what this is all about.

I call the disk Sideways Fonts and it's a follow up to your work on the overlays. Hope you find the main file (redef.Cc) and its loader, (redef.Bb) useful.

Re entering text into the overlays, I found it better to enter each horizontal line (or part line) by the rather tedious procedure of moving vertically. This rather than your option which became confusing when I altered the spacing. So you and I will differ there, but so what? Whatever is easier for the individual is what counts.

I have a copy of tsdb but lack the instructions so never explored it. George is getting me a copy of the ones you sent him. I have a variety of disassemblers and switch around amongst them as the occasion demands because they all have their strengths and weaknesses. I like to work in decimal so I gravitate to Spectramon which I call Monitor for the TS2068. It has the advantage of being usable with RND/SEQ filing since a CR is sent at the end of each line thus identifying each block. It doesn't decode Floating Point code but HOT Z or others will do that if needed. It doesn't have a search facility but my Editor file will do that (in hex). So there's usually a way.

George is looking into the schematic thing. But the chip number you wanted is 74HCTLS-32N. He knows what is in the forbidden area (96 to 112 hex) and says it is not particularly useful. But ask him for it; he said he could send it to Ken if it would be useful to him.

I'm providing the club library with a copy of the enclosed disk. If you find any bugs in it, let me know. By the way I fixed up your !sidew.Cf in a couple of spots where the characters were too thin (eg. the letter Y). My version is on the disk as !side>.Cf. I also made the m and the w narrower and more like the ones in Spectrum.

All the best to you. Keep in touch.



Bob.

Bob Mitchell  
20 Wild Briarway  
Willowdale Ontario  
M2J 2L2  
901020

Dear Les,

Thanks for sending along the overlays, a truly fine application of Pixel Print Professional. George gave me a copy of your most recent disk with the various columns so I am ready to do some of these myself. I have used my Exacto knife to cut out the rectangles. I really know Tasword and Mscript by heart now (almost) but now and then the overlays will come in handy for little used commands, etc. It's the seldom used programs where the overlays will be most appreciated. I'm thinking of Zeus for one; Profile and Vu-Calc are two others I'll have a go at. I did a modification of the Mscript overlay to include some new stuff (H for help; f for fullcat after pressing C for catalogue; M for move (alias Rename) and D-4 for Drive Select. This turned out OK although I must admit it's a bit hard on the neck leaning over to read the sideways characters. Just kidding!.

So once again, applause for a fantastic job to both you and Matt Kiddo!

I checked out the Monitor/Disassembler which is quite good considering how few bytes it takes up. Have you done a conversion to 2068 ROM by any chance? If not, I'll have a go at that as it would be a useful variation.

We had a fun time getting PPP into working shape; Larry Crawford (London Ont) did the essential work to make the bank switching work. George and I added some frills here and there. We were sorry to hear that Stan Lemke had abandoned the whole project and sold off all his Timex gear.

I have just finished doing some revisions to my Omnibus disks. One is for those club members who do not sport a RAMDISK and a QUAD drive but who have two drives one of which is DSDD. It has a neat trick built into it that stores the drive numbers of two drives in addresses 26666 and 26667 and so makes switching more automatic and, since these addresses are in the systems variables area, makes getting back to the Omnibus disk easier (ie, from other BASIC programs on the disk).

*and program*

I have not built this device into my Quad version of Omnibus and doubt that I shall. My latest version is 2.10 and has a lot of niceties and other user-friendly operations that make the whole disk easier to use. Many of these frills are included in the DSDD version.

Using the Larken Random/Sequential filing system (disk version) enabled me to do the 2068 ROM disassembly (55 tracks long) and the exrom. I put these on a Quad disk along with some tutorial material.

If any of these things interest you, let me know and I'll be happy to send you copies.

It's doubtful if my wife and I will be coming to Florida this winter but if we change our minds, it'll be for a month. Will let you know.

Meanwhile, best regards.

A handwritten signature in cursive script that reads "Bob".

Bob Mitchell

TORONTO TIMEX-SINCLAIR USERS CLUB  
October 20, 1990

14 Richome Court,  
Scarborough, Ont. M1K 2Y1

Les Cottrell  
108 River Heights Drive  
Cocoa, FL 32922

Dear Les,

Though I have had several letters from you recently, I have not replied. Partly because I have been pretty busy lately. But I see that I have neglected to send several disks off to you, that you asked for. Namely #25, ~~28~~, and 32 (the Bytepower disk).

*menu* *see end of letter*

I also have to thank you for the keyboard matrices that you sent me, Bob, Larry, and the club. Of course the club ones went like a snap the first meeting!

Alas, I cannot make proper use of them, since I have an auxiliary keyboard on my computer, and it sits right over top of the computer.

I think that Bob is going to write to you to inquire about the easiest way to use the sidewise character set. He is coping, but is sure there is a trick of to to using it.

Actually I have not loaded your matrix programs into PPPr. and used them. I simply made a copy and sent it to Bob. I think I also sent a copy to Larry Crawford, also. I have made a new club library disk number, #37, which will hold all the data files which are common to PP+ and PPPr. Your disk, and the 'fonts' disk that is in PP+ #10.

You asked about the computer. Well, as I said in the newsletter, I was going to reduce the price of things that hadn't sold, by 25% in the next newsletter. So I think you can pick one up, one of these days.

For some reason both computers do not output to a monitor, i.e. the monitor jack seemed non-functional. But the TV output is good, though I am spoiled with an RGB monitor, and am not particularly enchanted with the colors!! Both computers are equipped with a reset switch which is mounted at the rear of the computer.

\$60-45

The best computer comes in it's original shipping case, and seems otherwise perfect. I tried them both out with large programs, and they seemed to work properly. Tried them out with a Larken system, and with tape recorder.

The best computer comes in it's original shipping case, and seems otherwise perfect. no sign of keybard wear, and all keys function. Has all the accessories including manual, power supply, and cords.

\$50-37<sup>50</sup>

The second computer (the one I originally priced at \$50) has some cosmetic flaws. The keyboard shows no sign of wear, the legends are in perfect shape. But there is a hole in the upper case to provide access to one of the screen adjusting pots. The bottom case also is cut away right at the rear of the RF output box. Not visible from the front or top, though. There's a manual to go with it, and the power supply. I think there are no cords for the tape recorder or TV, though there is a TV/Computer switcher.

It does not have the original shipping box.

I think I have described them fairly critically. Take 25% off the original price, with each successive newsletter.

I may have overpriced the computers. They have not sold. I thought the time the prices were good. However, in reducing the price in stages we'll find out what the going price is.

The 2050 modem had been modified to give it a Serial port, per some mods that were published in some past newsletters. I turned it over to Rene Bruneau to see what he could make of it. I have not spoken to him about it. I was not sure that it was still working as a modem, and I was not going to get into experimenting with it.

Shall close off now and get the disks off to you.

Sincerely,

George Chambers

On reading at an earlier letter from you I see I have another comment. The "Thor 4" font was sent to me by Steve Spalding for the club some time ago, so it is in the public domain.

Bob is revising the pull-down menu Omnibus disk to incorporate Larry Crawford's idea for customizing drives. I shall wait until I get that upgraded disk from him - Ask for it again (Well, I don't happen to have a spare copy at the moment either, & I'd sooner not make up another, than have to re-do the disk).

I also rec'd the disks etc back from you, re Ken Silverberger

September/October 1990

September 8, 1990

Dear Out-of-town members,

(28) All you Larken RAMdisk owners may be interested to know that we have completed the modifications to the PIXEL PRINT PROFESSIONAL program to make it compatible with the Larken RAMdisk. This work has been done through the combined efforts of members Larry Crawford, Bob Mitchell, and myself. It makes use of the first chip on the RAMdisk.

The program is on disk #28 in our Larken disk library. The program is "Fairware". That is you are welcome to use it; if you like it you can send Stan Lemke a recommended contribution. So by all means ask for it and try it out.

In connection with this, another of our members, Les Cottrell, has sent me a keyboard template to be used with PPPProf. Well, actually the template pattern can be used with any program you wish. You take the pattern, xerox any number of copies, and customise each of them to suit each program you choose. By customise I mean mark in the function of each key used by the program. Ask for a copy (copies).

The Indiana T/S user group have created a PP+ file that prints out this template. I am hoping to get a copy of it, in the near future.

I'm not sure if I mentioned it in the last newsletter. Stan Lemke, of PIXEL PRINT fame, has sold off all his Timex stuff, and has gone onto other things.

We received an "offer to sell" in the mail a couple of days ago. It was from Paul Bingham, and he was offering to sell a lot of stuff, including Timex gear. You may remember Paul Bingham as the author of a series of articles in Time Designs Magazine, called "Classy Front End". I think I remember him in the old SYNC magazine, before that. Well, it looks to me as though he is giving up the candle, also.

(29) We have another disk in our library. It is from BYTEPOWER, and the programs on it are written by Kristian Boisvert. He has placed them in the public domain. The programs are a suite of Larken disk utilities. Quite some novel ideas. Do try this disk out.

Then there's another disk, #31, which contains a suite of programs for doing various electronics calculations. The programs for this disk were supplied by club member Ronald Ginardi.

(32) And another disk which contains a TS2068 ROM disassembly, prepared by club member Bob Mitchell. It makes use of the Larken Sequential/Random Access program, so you will need to be a bona fide owner of the Larken LKSR LBASE code to make use of this disk.

We also have a disk, #33, which contains a "pull-down" menu approach to the use of the OMNIBUS program suite of disk #2. Another work by Bob Mitchell. This "pull-down menu" program has also been placed on our "MENUS" disk #25.

I shall write about these new disks in our next newsletter, but this will keep you posted in the meantime.

I have been reading a book by Scott Mueller called Upgrading and Repairing Your PC. A very readable book. It costs \$35; I borrowed a copy from the library. I was most interested in the section about disk drives, but I found the whole book very enlightening. The author heads up a service company, and he writes in a down to earth manner.

I have quite a bit of written material about disk drives if you are interested; ask me for some of it!

Our treasurer and I went out to visit a member who suffered a stroke some time ago. Consider yourself lucky, believe me.

Anyway, Bill and I brought most all of his Timex equipment back with us to dispose of. I have put an advert in the newsletter. If any of you wish to add to your Timex equipment, this would be a good opportunity to do so. I have put some suggested prices in the advert. If it does not sell, I shall lower the price in subsequent newsletters by 25% until it goes. I think that some of the prices are attractive. Particularly the drives. They seem as new.

The 2 computers seem to be in working order. They both have reset buttons installed. No, both of the computer have the TV output working, but the Monitor output does not seem to work on one of them.

And there's a load of other smaller items that I have not been able to list. If you are interested in adding to your collection, ask me about them! Please do, I don't want to clutter up this column with the details.

This may be your last chance, so don't say I didn't warn you. I shall quote from the most recent C/CATS newsletter, the PLOTTER":

LARKEN NEWS

Larry Kenny tells us that he will try to have another 35 cartridge boards made up. He said that, after almost two years, he hoped that the company that made up the PC boards would still be in business and would still have the artwork needed to make the boards. If they do, we will have 35 more systems available. Unless Larken and RMG Enterprises receive more than 50 orders within the next three months, there will be NO MORE LKDOS systems made by Larken. Maybe we can get someone else to take the product on? We can only hope. Order your system today, or you may be out of luck." End of quote.

Larken Electronics

RR #2

Navan Ont.

CANADA K4B 1H9

tel (613) 835 2680

RMG ENTERPRISES

1419 1/2 7th STR

Oregon City, OR 97045

tel (503) 655-7484

I have a back-up Larken LKDOS i/f. Being so involved with the club, I thought it would be a disaster if mine failed, so when a used one came up I grabbed it. Well, a couple of days ago my system stopped writing to disk. I have installed the second board, and am trying to figure out the problem. By chip interchange I've determined that it is not in the removeable chips. Even the big chip tests out OK. So there's where it stands at the moment. Any of you serious users might take note!

Another newsletter, from the Indiana S/T user Group, has a sheet listing Dan Elliot's repair pricelist, etc. We shall publish it in the next newsletter. Seems mostly to be clarifications re shipping charges & minimum charges; the repair prices are essentially the same, really a bargain!

I have picked up the August '90 issue of Vulcan's Computer Monthly magazine. It looks good. Though they seem to be having some teething problems. They have repeated a T/S article by Richard Matejovic that first appeared in the April issue. Then one month they put in an article by Bill Ferrabee, but omitted a listing that he made reference to. But Bill was not in evidence the following issue.

The Sinclair Milwaukee User Group newsletter, SMUG-bytes mentions that the first issue of the UPDATE magazine under

the new publisher, Frank Davis, has appeared. The newsletter continues with the comment, "It has just as much, if not more, interesting facts and art on all Sinclair computers".

The Time Designs saga continues.... A newsletter (PLOTTER) reports that Michael de Sos, author of TAKING THE QUANTUM LEAP, has filed suit in small claims court, to try to recover some royalties from the sales of his book, and as well, to try to recover control of the book so that he can market it again. And another subscriber has got a refund as well as the back issues that were owing him. He went to the Oregon Dept of Justice. I can give you more information if you wish to follow this up. The last issue that we know of was Vol 5, No. 2.

We look kindly upon the Clackamas County Area T/S Users group (C/CATS). They recently reprinted Jeff Taylor's Editorial in our May/June '90 issue!!

Anyone for Spectrum games!! Ask for my tapes/disk listings. I worked this listing up using Larry Crawford's InterBank DataBase program on disk #30.

Something else. Bob Mitchell has updated his Pro/File database of our newsletters. These Pro/File files now contain all the data for issues from 1985 to the present. I use them quite often, looking up material that club members inquire about. Ask me for a copy of the files.

Also, maybe I mentioned it before, maybe not. Another member (sorry, I can't credit him at this moment; my memory fails me) has supplied me with a Pro/File file which logs every episode of the STAR TREK series. Any StarTrek fan would find this fascinating. I find it staggering that we have a member who can go to the trouble of getting all this into a database!!

Then Larry Crawford says that he is going to update the list of movie titles on his Interbank Database file. When completed it will have over 2500 entries, he tells me! I understand that these are titles that he has viewed! How he finds time for the TS2068 is beyond me!

Please don't look at the expiry dates on the mailing labels - They are gold labels & the expiry dates are not valid. Thanks.

Sincerely George Chambers