Sorcerer Users' Group (Toronto) Newsletter
200 Balsam Ave. Toronto Ont. M4E 3C3
Vol. 1 No. 4
June 1980

EDITOR'S TURN

There is definitely a growing discontent amongst microcomputer users towards dealers and stores selling microcomputers and microcomputer software. From personal experience and from talking with others, I can honestly say that the microcomputer user and hobbyist is being treated very shoddily by these businesses. In the beginning of this micro revolution, dealers would welcome with open arms, anyone walking through their doors itching to spend up to $2000.00 on these new toys. As the market expands and these micros are upgraded to handle small business applications, the hobbyist is being chucked aside for the more profitable businessman who has up to $15,000.00 to spend. However, because of a large misconception, both on the part of the dealer and the buyer, a slingshot effect is beginning to occur that I feel is going to kill this whole micro thing altogether. As a result of this effect, the dealer finds that he must get deeper and deeper into deception in order to sell the product. It really bugs me to hear a salesman telling Mr. Small Businessman of all the wonderful ways micros can help to run his business, how cost effective it is, how easy it is to operate and that his business will never be able to compete with others unless he computerizes now. If micros are so wonderful, then why are most of these shops doing the major part of all their operations manually?

Whoever is behind the marketting and running of these businesses needs to try one more time and then quit. They seem to think they can run a computer sales outlet with the same business consciousness of the hi-fi/appliance dealers. This whole idea of being able to purchase a computer, take it home, plug it in and voila - instant everything is a lot of BULL****. Computers, especially micros, are complex pieces of electronic machinery that are just as capable, if not properly programmed, of spitting out nothing but useless trash as they are of handling your accounts and providing your new digital entertainment. You can't convince me that the shop-keepers were not aware of this when they sold you the machine. So why do they look so surprised and treat you like you had the plague when you show up at the shop a few days later asking how to use the damn thing.

The most frustrating thing to a new owner of a microcomputer has got to be realizing that the salesman who sold you the thing knows less about it than you do. The second most frustrating thing is finding out that all the other shops are exactly the same. The next time you visit the shop, you find out that they are pulling out of the micro market. They give lame excuses like "poor delivery" and "no dealer support" as the reasons. Who do they think they are

kidding? I get fast delivery from manufacturers when I order direct. I can't believe that a shop that orders in large quantities can't get that kind of service. The real reason for their downfall is poor marketting. They shouldn't have gotten into the business in the first place. They can't handle it or service it.

The truth is, that I wouldn't even consider getting into this market. The average buyer doesn't know how to use the product and will be back with lots of questions. That means that I would have to pay experienced, educated staff to answer them day in, day out. Furthermore, while Discount Dan is down the street making all the sales for dirt cheap, his customers are in my shop looking for free support.

Let's face it. Only a very small portion of those people who presently own micros know how to use them and those who can manage, just don't have the time to write programmes. The only other alternative is to spend hundreds of dollars for commercially produced software. For this reason, the microcomputer will fail as a personal product and disappear from the home. What will come, though, is a communication centre whereby people will communicate with the outside world (in a computerized way) through a black box, a T.V. and a cable or telephone line. The user will be able to respond to programmes, take courses, get the daily news, shop from catalogues, etc. from his/her favourite chair without effort. I'm totally against this, but that's another editorial.

So what's the solution for those who already have a computer and are in this boat? Tough question. The first thing I would do is make the additional investment to standardize my system to the average level of sophistication that most people have in order to keep communication lines open. As far as I can see that would mean an S-100 box, a floppy disk unit, perhaps a modem and probably the CP/M operating system. The second thing I would do is join any groups I could where I could interact with other users and share in some of their ideas and programmes. The third thing I would do is take advantage of any educational programmes (no plug intended) to learn how to use this equipment effectively. There are two other solutions; be satisfied to spend hundreds of dollars for playtime software or SELL.

Sorry for the pessimism again this month but we're only fooling ourselves if we are convinced that we have broken down communication barriers and extended our minds simply because we own a microcomputer. There's a lot more to it than that. The fact that this is our fourth issue and in that time I've only had two articles submitted is a clear indication to me that nothing is happening in personal computers or in personal computer users and that is why nothing is happening in shops that sell personal computers. Very soon this bomb is going to land right back into the laps of the manufacturers. Look at all the large eletronics manufacturers who know the market; Texas Instruments - video games with optional programming; Atari - video games with optional programming; Hewlett Packard - sophisticated usage, labs and engineering only. If there was a good, solid market in the personal computer area then you would definitely see IBM, DEC, Honeywell, Burroughs, etc. right in there with the best machines available.

This whole thing will soon come full circle. There will be a few winners and many losers. Those winners will be the ones who can programme and make full use of their system and the Tandy Corporation because they have successfully shoved their product down the throats of America past the bottom line already.

LIBRARY

Here is a barebones list of the titles that are currently in the cassette library. At present, I only have the titles and the list is two weeks old. For a better and more complete description, call the librarian, David LeBlanc at 925-9917.

LIFE, QUEST, CHECK, USWAR, HANGM, KINGD, CHESS, RTREK, G, TREK, SHOOT, TYPE, LINKR, GETT, WP8K, AMBSH, LRGLT, SPACE, WSHOT, ESP, BIO, PUZZL, BLBOX, INTRP, GET, BRK3, ROOT, FGTR, SCRAL, MORTG, DUMP

We certainly hope that the library will grow and contain some good programmes that we can all share and enjoy. If you have something that you can donate, please call David. The present method of getting library programmes is to sign out one of the tapes at a club meeting or to call David to make other arrangements. The borrowing period is from one meeting to the next or 1 month if you cannot attend the meetings. We ask that you return the tapes on time so that others may borrow and that you take the responsibility not to distribute copies of these programmes to anyone. Remember, you have signed the agreement form to this effect.

SOFTWARE

Last issue I gave you a very powerful routine that made it possible to perform flexible, easily formatted screen output. In this issue and the next, I will show how to do some equally powerful and flexible input routines. The first routine below simulates a BASIC "GET" command and the second routine allows you to perform real-time input without tying up the processor.

The first one is very simple to implement. It seems that Microsoft forgot(?) to include one of the most important and versatile commands into their ROMPAC, the GET command. The GET command allows you to request some input and programme your own inputting conditions. Presently, when using an INPUT statement, your programme does not get control back until a terminator is input. That terminator in BASIC is a carriage return. The GET command requests a single pass of the keyboard and a return of control without a terminator. This allows you to programme your conditions as to how to handle the input. The machine language subroutine looks like:

```
21 10 00    LD   HL,CHARHOLD     ;POINT TO WHERE CHAR WILL BE STORED
36 00       LD   (HL),0          ;CLEAR IT FIRST
CD 09 E0    CALL EXIDYINPUT      ;GET CHAR - 0 IF NONE
77          LD   (HL),A          ;STORE CHAR FOR NOW
CD 0C E0    CALL EXIDYOUTPUT     ;AND ECHO IT TO THE SCREEN
C9          RET                  ;AND BACK TO CALLING PROG.
```

In BASIC it looks like:

```
XX00 REM: THE GET ROUTINE
XX10 ML = 0:CHAR = 16
XX20 RESTORE XX99
XX30 READ CODE:IF CODE = -1 THEN XX50
XX40 POKE ML,CODE:ML = ML + 1:GOTO XX30
XX50 POKE 260,0:POKE 261,0:M = USR(0)
XX60 M$ = CHR$(PEEK(CHAR))
XX70 (whatever action you wish to take)
XX90 RETURN
XX99 DATA 33,16,0,54,0,205,9,224,119,205,12,224,201,-1
```

line 10 ML = the address where the routine begins. CHAR = the address where the input character will be stored.
line 20 sets the data pointer to this subroutine in the data statement
line 30 pokes the code into memory starting at location 0
line 50 calls the M/L subroutine at location 0
line 60 converts the character value stored at CHAR into its alpha equivalent and assigns to M$
line 70 you now have a character - take whatever action you wish
line 99 the equivalent decimal code as the above M/L code

Note: if nothing is input then a 0 is returned and M$=CHR$(0). You must also deal with this case in whatever way you wish.

The second routine is a little more complex. The Exidy monitor provides a necessary software function when normally inputting characters. Once it detects that a key has been pressed, it will stay in a loop until the key is released. This is because at processor speeds, it could read the same key dozens of times in the time required for an operator to press and release the key. However, it does have one drawback and that is that there is not a true real-time input (even with the above GET function), since the processor is stopped as long as a key is held. (This could result in cheating in games.) It can be overcome by simply changing the software. The problem is, though, that it is in a ROM and cannot be changed, so we have to relocate it in RAM first and then make the changes. The following code accomplishes this initialization:

```
XX00 REM: REAL-TIME INPUT INITIALIZATION ROUTINE
XX05 LOC = (wherever you have space for the routine - 258 bytes)
XX10 IX = LOC:IY = -5348
XX15 FOR I=0 TO 258:POKE IX+I,PEEK(IY+I):NEXT I
XX20 FOR I=IX+190 TO IX+194:POKE I,0:NEXT I
XX25 POKE IX+58,INT((IX+198)/256)
XX30 POKE IX+57,(IX+198)-INT((IX+198)/256)*256
XX35 POKE IX+75,INT((IX+198)/256)
XX40 POKE IX+74,(IX+198)-INT((IX+198)/256)*256
XX45 POKE IX+213,INT((IX+53)/256)
XX50 POKE IX+212,(IX+53)-INT((IX+53)/256)*256
XX55 POKE IX+226,INT((IX+49)/256)
XX60 POKE IX+225,(IX+49)-INT((IX+49)/256)*256
XX65 POKE IX+243,INT((IX+40)/256)
XX70 POKE IX+242,(IX+40)-INT((IX+40)/256)*256
XX75 RETURN
```

The above routine is the initialization process as follows:

line 05 initialize the addresses to move from and to
line 15 is a BASIC block move routine which relocates the monitor input routine from IY to IX
line 20 removes the routine that waits for the key to be released
line 25 adjusts the 5 jump addresses in the routine for the new address.

Now the new routine can be used in conjunction with the previous GET function for real-time input which will not stop the processor with the inclusion of the following lines in the GET routine:

```
XX42 MWA=256*PEEK(-4095)+PEEK(-4096)
XX44 IF MWA<32767 THEN MWA=MWA-65536
XX46 MWA=MWA-110
XX48 POKE MWA+66,INT(IX/256):POKE MWA+65,IX-INT(IX/256)*256
XX85 POKE MWA+65,28:POKE MWA+66,236
```

Line XX42 finds the MWA area in any size machine where the input and output vectors are located.
Line XX48 changes the input address from the old input routine to the new one we just created. IX is the same value we used in the preceeding initialization routine.
Line XX85 puts the old routine address back for regular usage.

Now when you call GET it will call the new input routine for its character. If you want to hold a key down for continuous input such as moving a video image fluidly across the screen, the processor will not be stopped and the user will not have to keep toggling the key. If, for some reason, you want discreet input using this routine, you have to insert a timing loop to give the user time to release the key. Experiment with this to suit your taste and needs.

Next month, I'll show you how to test for particular keys (best suited for games) without using any machine language subroutines and directly from Basic using only a couple of lines of code. It's extremely fast, efficient and very simple to use.

I just recieved a very useful and powerful software submission from Tony Bagshaw at North American Software which is appended on a separate sheet.

HARDWARE

I have had no hardware submissions and since I'm not very skilled in hardware this column will remain vacant this time around.

EQUIPMENT REVIEWS

I would like to start a new category that may be of interest to others who are looking for equipment called "EQUIPMENT REVIEWS". If anybody is purchasing any new items, we'd all like to hear what you think about it.

Because of the success with my present course, I am planning to offer a series of courses over the next year. Due to my present situation (unemployed), this will not be done as volunteer work for SUGT, as the Z-80 Assembly Language course was. However, I will offer a discount for SUGT members (which obviously means that the courses are open to anyone and everyone). The courses will each be 36 hrs. - that is, one lesson/wk. at 3 hrs./lesson, 12 lessons/term. The fee for these courses will be $100.00 for non-members and $85.00 for SUGT members. In most cases, you will leave the course with a good working knowledge of the subject and with some good software developed in the class.

Fall Term (Sept.-Dec.) "Languages"

1) BASIC - not as simple as you might think. Although we will start from the very beginning, the course is designed to build a good, solid foundation of style and a thorough understanding of programming techniques and algorithmic construction.

2) 8080/Z-80 Assembly Language - the course offers instruction in the machine coding language, number systems, Boolean Algebra, logic, assembly language programming technique, data structures and some database management, etc.

3) The "C" Programming Language - this language is quickly becoming one of the forerunners in structured programming and artificial intelligence. It soon will become one of the most popular languages in microcomputer usage. The language itself is quite simple to learn, easy to code and reaches far beyond BASIC in flexibility and power.

Winter Term (Jan.-Apr.) "Development"

1) Advanced BASIC - this second course teaches some fairly advanced BASIC applications. It delves into algorithmic design and forward flow. In the course, we will extend the ROMPAC command set to include a versatile video package and input routines, new string handling capabilities and tape drivers. A good working knowledge of BASIC is a pre-requisite.

2) Z-80 Programming Techniques - database management, sorting, hashing, vectored I/O and piping, algorithm design and analysis in Z-80 Assembly Language. Pre-requisite - Z-80 Assembly Language

3) Structured Programming - using structures and unions, single-thread, forward-flow programming, binary trees and tree pruning, hashing, look-up tables, recursion, etc. using "C". Pre-requisite - The "C" Programming Language.

Spring Term (May-Aug.) "Specialties"

1) CP/M - how to use CP/M and DOS, writing programmes for disk, how to use ED, ASM, PIP, DDT, SUBMIT, STAT, DISKCOPY, DUMP, FILECOPY, LOAD, MOVECPM, SYSGEN, FORMAT and others. How to understand system interfacing and alterations.

2) Z-80 Programming Team Effort - a team effort by everyone in the class in completing a project. Guidelines are set down for a project and each student is responsible for the design of a portion of it according to specifications. Class instruction includes a detailed dissection of the EXIDY monitor, programming techniques and discipline and algorithmic design. At the end of the course, each person will own a (hopefully) good piece of software. Some present suggestions are a record retrieval system, a video package or a text editor. Pre-requisite - a good understanding of Z-80 Assembly Language

3) Special Interest - not available until 1981. Some present ideas are LISP, FORTRAN or PASCAL.

The CP/M course and the Z-80 Team Effort are planned for this summer and will run from sometime in June through the summer. It is imperative that you call me as soon as possible if you are interested so that proper arrangements can be made for the courses.

GENERAL NEWS

No news yet on the development of the modems. At last check, there were a few bugs yet to be ironed out. Perhaps there will be a breakthrough by the next meeting.

You may have noticed the improved print quality of this newsletter. Thanks to Willie Fahnestock, the newsletter is now being done on a XEROX Diablo printer and editted with WORDSTAR. What a treat for me.

Memberships are $10.00 per year and run from Jan. to Dec. If you would like to become a member, call the president, Duncan Lang, at 461-2224 and he will send you out a membership agreement form to sign.

We are presently in need of a Secretary/Treasurer for the group. If anyone is interested in volunteering for this esteemed position, please call Duncan and we will put it before the membership at one of the meetings. Your support would be appreciated.

Next meeting is Thursday June 12 1980 at 7:00pm in the cafeteria of Jesse Ketchum Public School.

BASIC TEXT-EDITOR

The following is a listing of an Editor for Standard Exidy Basic. It must be loaded from Address ØØØØ Hex in Monitor and must be checked thoroughly before using it, as most Assembler Programs destroy the contents of your Memory if they contain any errors. Once you have entered the Program, go into Basic and enter the following:

```
Ø     GOTO 1Ø
1     POKE 26Ø,Ø  :  POKE 261,Ø  :  X=USR(X)
```

Then start writing your Program in the usual manner, beginning from LINE 1Ø. So far, when you keyed in errors in your Basic Program, you had to change the whole line. From now on, Load the Program before writing and correct errors in the following manner:

List the Basic Program and stop the listing when the line containing the error appears on the screen. Then type GOSUB 1 and hit (Return). The cursor should now be flashing. If this is not the case, you went wrong somewhere. Move the cursor to the line on the screen containing the error by pressing Control Key with the 8,4,6 & 2 for the direction required. Once at that line (arrows on Numeric Pad will get you there) go to the position where the error is. Overwrite the error. Should you want to delete a character at the cursor-position, press CTRL D (Control and D). If you want to insert a character press CTRL I. When finished you must hit (Return) twice. It is vital that you hit (Return) twice! The editing functions on one line only. That line is where the cursor is at the time you hit (Return). Whatever is on that line is accepted as input. i.e. If a line goes over 64 characters, you lose the rest of the second line. Onestroke instructions are accepted. List the line and check that the correction was done properly. For the next line, again key in GOSUB 1.

SCREEN-EDITOR FOR EXIDY BASIC
COPYRIGHT (C) 1980 BY C. ALMER
**** VERSION 2 FEB. 1980 ****

ADDR	Ø	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
ØØØØ	CD	F1	ØØ	CD	C6	ØØ	28	FB	FE	17	28	12	FE	1A	28	ØE
ØØ1Ø	FE	11	28	ØA	FE	Ø1	28	Ø6	FE	13	28	Ø2	18	Ø5	CD	FØ
ØØ2Ø	E9	18	EØ	FE	Ø4	28	38	FE	Ø9	28	1F	FE	ØD	28	58	CD
ØØ3Ø	7C	ØØ	CD	ØC	EØ	18	CC	CD	D6	E9	E5	21	4Ø	ØØ	37	3F
ØØ4Ø	ED	52	E5	C1	E1	Ø9	2B	E5	D1	C9	CD	37	ØØ	E5	D5	C5
ØØ5Ø	CD	E8	E9	C1	D1	E1	2B	ØB	ED	B8	CD	CC	E9	18	A4	CD
ØØ6Ø	37	ØØ	C5	D5	E5	CD	E8	E9	E1	D1	C1	37	3F	ED	42	23
ØØ7Ø	23	E5	D1	1B	ØB	ED	BØ	CD	CC	E9	18	87	FE	1B	CA	FA
ØØ8Ø	DF	FE	7F	CØ	3E	Ø8	C9	CD	E8	E9	CD	D6	E9	37	3F	ED
ØØ9Ø	52	11	4C	Ø1	Ø1	4Ø	ØØ	ED	BØ	EB	2B	3E	2Ø	Ø6	3F	BE
ØØAØ	2Ø	Ø6	36	ØØ	2B	Ø5	18	F7	23	E5	21	AØ	Ø1	36	ØØ	CD
ØØBØ	A2	E1	3E	ØØ	CD	E6	ØØ	ØØ	ØØ	E1	3E	ØD	11	6F	C3	D5
ØØCØ	C3	3F	C5	ØØ	ØØ	ØØ	Ø6	1Ø	CD	E8	E9	CD	Ø9	EØ	2Ø	1Ø
ØØDØ	1Ø	F9	Ø6	1Ø	CD	CC	E9	CD	Ø9	EØ	2Ø	Ø4	1Ø	F9	18	E6
ØØEØ	F5	CD	E8	E9	F1	C9	D9	FD	73	68	FD	72	69	FD	74	67
ØØFØ	C9	CD	A2	E1	FD	5E	68	FD	56	69	FD	66	67	D9	C9	ØØ

DUNCAN LANG and JOHN ROY would like to announce the formation of their new company MICRO DIGITAL SYSTEMS with the following programmes:

MDS/BIOS -is a programme which provides enhancements to Sorcerer owners running the CP/M operating system by providing customized I/O processing allowing normal rub-out capability, screen wrap-around, dynamic status testing, key conversions, variable speed outputs, etc.
-speeds up normal operations
-provides vectored I/O for keyboard, screen, parallel and serial ports
-links the entire EXIDY monitor system directly into CP/M
-does not alter normal operations of CP/M

programme and manual...$75.00/manual only...$10.00

KWIKBASE -an indexed record retrieval system in 8080 machine code providing fast, easy access to information, and user defined field names.
-options allowed include ADD, CHANGE, DELETE, SEARCH, SORTED LISTING, FILE DUMPS, PACKING(compressing), and MERGING FILES.
-records always sorted on input
-provides a unique security system to prevent tampering with files.

programme and manual...$75.00/manual only...$10.00

TAPECOPY -for Sorcerer Users running CP/M
-allows buffered I/O of files between disk and cassette tape
-can save files which are larger than the machines memory size, backup disks onto tape automatically under a SUBMIT file at 300 or 1200 baud.

programme and manual...$40.00/manual only...$5.00

Programmes presently available on Micropolis Mod I and Mod II disks and on Thinker Toys Single and Dual density disks.

KWIKBASE and TAPECOPY also available on cassette for Sorcerer owners with other disk systems.

How to order: Manuals may be ordered separately and will be deducted from the price of the programme if ordered later. If ordering more than one programme, save $4.00 on each additional by requesting all on one disk. Deduct $4.00 for each disk saved. Unless specified, all programmes will be shipped on separate disks. If ordering more than one programme, deduct a further 5% off the total.

EXAMPLE: All 3 programmes.....	$190.00
less disks (2*$4.00).....	$ 8.00
Sub-total.....	$182.00
less 5%.....	$ 9.10
TOTAL.....	$172.90

Note: Ontario residents include 7% sales tax. Certified Checks or Money Orders only. Chargex and Mastercharge available soon.

Send orders to:

Micro Digital Systems
762 Gerrard St E
Toronto, Ontario
M4M 1Y6