065 8704
Ottawa

The Ottawa T.I.99/4A Users' Group

NEWSLETTER

VOLUME 6 NUMBER 04.........APRIL 1987

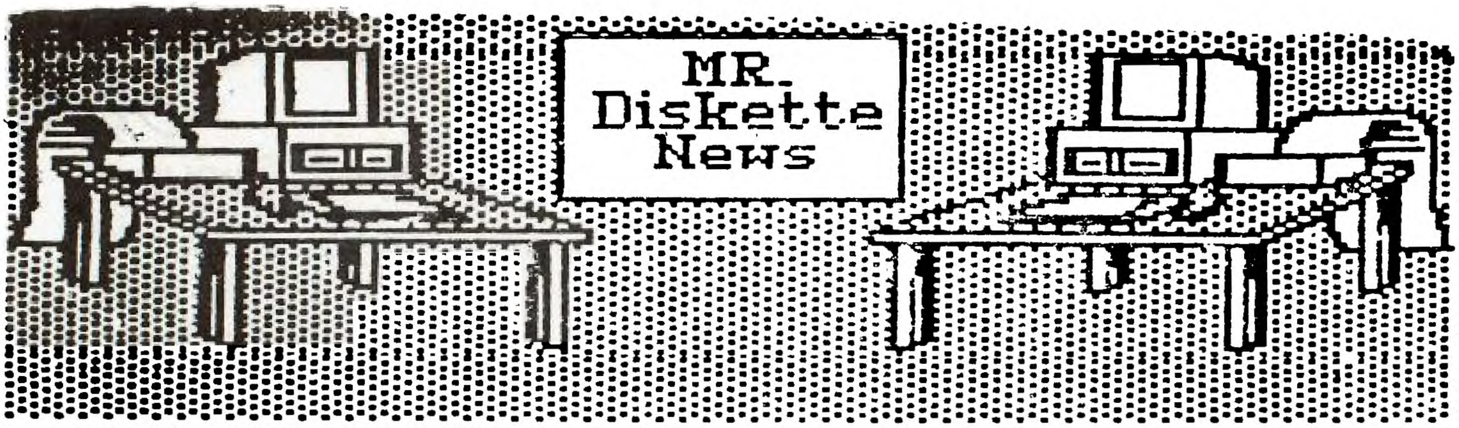
TI-FEST

9.00 AM to 5.30 PM
MERIVALE HIGH SCHOOL
1755 MERIVALE RD
NEPEAN, ONTARIO, CANADA

SATURDAY
MAY 16, 1987

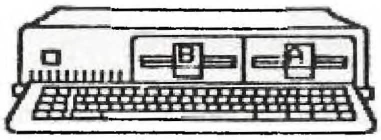
A FESTIVAL CELEBRATING
THE TI-99/4A

P.O. BOX 2144, STATION D, OTTAWA
*** ONTARIO, CANADA K1P 5W3 ***

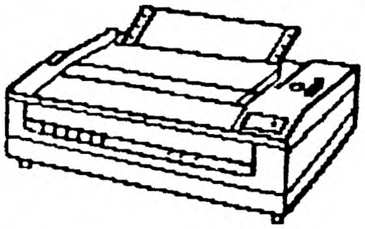


Need a second Disk Drive? We have Panosonics on at a special price of:

$169.00



Looking For A Printer?



The Siekosha SP-1000 A is a Near Letter Quality printer that can't be beat. It sports a draft mode at 100 cps and a NLQ mode of 22 cps. With both tractor and friction feed the price can't ba beat at

$369.00

Bulk DS/DD Diskettes

.79 EACH

DD-100L Diskette Box

$14.95

HEAD OFFICE. 105 O Connor St
Ottawa Ontario
K1P 5M8
(613) 232-5203

1600 Merivale Rd.
Nepean Ontario
K2J 3K3
(613) 727-0180

EDITOR'S NOTES

by Marg O'Connor

A problem that I must bring to your notice is that we will publish want adds for club members, BUT only for themselves. We can not put adds for companies in for free, the price for them is $40. a page, $20. a half page and $10. a quarter page. It is not fair to ask some dealers to pay and not others.
In going over some of my disks I came across this bit of information from Sheven Shaw.
It seems that people are having problems with module and peripheral connections. He has suggested the use of cotton buds to clean the contacts, but two new products which give a more satisfactory solution have benn brought to his attention, and are readily available by mail order from MAPLIN whose catalogue is sold at most branches of W H SMITH. Check the TOOLS section.
a. First cleanse the edge connectors using a special moist tissue described as "SAFEPADS" (ref FM81C). DO NOT use "CLEANING STRIPS". The moist tissue can be wrapped round an old sucker stick or even a cotton bud if you have problems getting at the edge connectors!
b. Now use CONTACT CLEANER LUBRICANT PEN ("SWITCH CLEANER PEN"), Soak a clean cotton bud with the lubricant and wipe it over both sides of the edge connector. Don't be shy!
This should solve most problems!
By EDGE CONNECTOR I am referring to the PCB in the modules, and in the peripheral sockets.

THE CHAIRMAN'S TWO CENTS WORTH

by Berry Minuk

The column this month will be a little unusual since it is being written before the April meeting instead of after it. That's because I am going on my holidays very soon.

I will therefore be writing to you about Fairware. This is partly because that is the topic for the May meeting and partly because of a letter we recently received from a Fairware author named Mark Beck. On many occasions in the past I have exhorted our members to support Fairware authors. The response has not always lived up to expectations.

Mark's letter is quite lengthy and I am writing him to obtain permission to edit it or receive it on disk. His program known as Creative Filing System was donated by him to his Users Group to help them raise funds for their activities. Our group has some knowledge of this method since we distribute several programs donated by our members Bruce Caron (updates by Ralph Romans), Jean-Pierre Morin, Art Green, Mauro Tomietto and Steve McWattie. Although DM-1000 has been very well received most of the others have had very little response.

I have done some thinking about the reasons why some fairware programs seem to generate much more response than others and there are some very obvious answers such as quality of the program and broadness of use. For example DM-1000 is a professional quality program which a user will use almost every time that user turns on their computer. This constant use of a high quality program results in a constant prodding of the user to send in his or her donation.

By contrast Mauro's program Sideways Print which is of equally high quality is not used very often. Firstly, its main use is to print Multiplan or TI-Writer files at a 90 degree angle. This is a tremendous idea but is not something that the average user will use quite often. I leave it you to see the consequences.

But there are other factors involved, some dependant on the previous and some not. It occurs to me that the more often used program spreads further and faster via the fairware route, i.e.

BBS's, User Groups, magazines and so forth. The less used program even though just as useful does not.

Another factor which I think has a strong effect is uniqueness. When DM-1000 first came on the scene there was no program on any market which did what it does. The closest was the Cor-Comp Disk Manager but that required their controller and was slower. Sideways on the other hand has to compete with at least one other fairware product and I believe one commercial product.

I do know one thing for certain even if most of the above article is speculative and that is that if we do not support fairware authors we can not expect them to continue supporting us. If every time you made major use of a fairware program you were to put some money in a bowl, you would soon have enough to send the author a donation.

Enough philosophizing I'll see you all at the May meeting and next month this column returns to its regular format.

BROWSING THE LIBRARY

--with STEPHEN BRIDGETT

The fest is fast approaching and your librarian is frantically trying to put the software catalogue together on time. Unfortunately it was not available for the April meeting. There seems to be a little confusion about the catalogue and the cataloger program. The program, which was offered as Disk of the Month for March and April allows the catalogue data files to be created and read. It is not the catalogue itself, but you must have this program in order to access the catalogue files. If you do not have the E/A module, you will also want to get one of the assemblers available, ie on : Funnelwriter.

In order to have a useful catalogue, we must also have documentation, that is a short explanation of what a program does. I think we all agree on that point. The problem has always proved to be to massive for any one or even several people. I would like to start a documentation process and this is my plan. The library disks will be available at the regular cost of $3.00. Members can order them by previewing their catalogue (to be released in May), and ordering them from the library either at the meeting or by telephone. With the disk I will be distributing a form, which I am asking the users to fill in and return. It asks such basic questions as we would want answered before purchasing a disk. The forms when returned can be exchanged for $1.00 or credit on another purchase. Now initially there will be little documentation and I'm asking users to get involved. Look at your catalogue and if there is something of interest to you, order the disk. If it turns out that the disk is of no interest to you then the library will refund your $3.00. If you fill in the form and return it you wil have done your library a great service and you'll get $1.00 in return. By making the cost and risk so low we all benefit. The software will reach the members and the documertation will reach the library. The docs will be on display at the meetings, for 'shopping' purposes.

If the catalcgue is ready for the May meeting, it will be the Disk of the Month, otherwise there will be back issues of disks and the catalogue will be ready for the Fest.

A special note to out of towners; I have several requests for software to be mailed. Although I have not had a chance to answer all letters promptly, they will be answered after the catalogue has been issued. Your patience will be rewarded with a realistic appreciation of what is available through your library.

Lets all get interested in making the library documentation a success.

See you in May
.....Stephen

SOFTWARE.......STEPHEN...521-3631

CASSETTE BASED SOFTWARE...JACK...225-6989

PRIOR PATTERNS APRIL 1987

by **Bob Boone**

What follows is strictly editorial and does not in any way, shape or form necessarily represent views of the Ottawa TI User Group or its executive.

Its getting out of hand, to the detrement of the TI community at large. Call it debate, discussion, arguement or mud-slinging...whatever; it has, long ago, gone too far. Very talented dedicated people are polarizing, one against the other on an international scale and from where I'm sitting there is no GOOD reason for it! If you haven't guessed yet the topic that has gotten so many of us, so hot under the collar is **PIRACY** and the strong commitments of some TI software distributors to continue to protect their rights or at least try to do so. Quite frankly, from my point of view, its turning into somewhat of a barroom brawl. I know and/or have personal dealings with people on both sides (pro and con protection) and have met or corresponded with most of them. They are all good, friendly people; each one is unquestionably dedicated to supporting the 99/4A. The biggest problems have surfaced since the great debate went public, first in newsletters then quickly spreading like cancer into the milieu of such as Compuserve and the Source. The arguement was fuelled and raged out of control when a trackcopy utility was uploaded to one of these databases for general public distribution.

I believe it would be contrary to my purpose in writing this to lay blame at anyones feet. North America, and in fact, most of the known TI Worldwide community is not dictatorship. Let each hold and freely express their opinion then act according to your own convictions. If you have strong feelings against a distributor protecting his software rights the solution is simple, don't buy his software. The only one suffering then is you, if the package is a good one. If you have a strong aversion against trackcopiers and their ilk (which I feel also have a place in the scheme of things), the solution there is simple too; resist their strong appeal!

My personal experience as a TI Distributor here in Canada is, on the whole a very positive one. I KNOW piracy is rampant EVERYWHERE yet I'm still selling software, both protected and unprotected. If its good and reasonably priced there will always be buyers. In fact many orders result from people seeing how well a pirated copy works, inspiring the honest users out there to get one too.

Stop reacting like wounded animals with one another. Tearing at each others throats does no one any good. Work together and not at odds with one another. Take a lesson about misplaced enthusiasm from the trojan horse article elsewhere in this issue and for gosh sakes stop taking each word uttered as a personal affront! You folks with your dedication, talent, enthusiasm and energy are too sorely needed by us all.

Terrie Masters, Mike Ballman, Craig Miller, Tony McGovern, Howie Rosenburg, Jim Horn, Ron Albright, etc, etc, etc, shake hands and please DON'T come out fighting!

NUAC NEWS APRIL 1987

by Bob Boone

Some of you renewed without prompting, many of you have not renewed at all! We will remain an Association in name only with no OFFICIAL existance for as long as you, its members, keep coming back. I very much appeciate the support I got in our attempt to accredit

NUAC, unfortunately it was too little to carry it through to reality. I've chosen our unofficial emblem from among those sent to us by Steve Andrews from North Bay, Ontario. Fine job Steve! The winning emblem will be shown at our Faire next month and the 50 disk prize will be presented to the North Bay User Group in Steve's name at that time as well.

An astounding 20 Ottawa User Group members went down to Boston last weekend and we had a ball! We were met there by 4 Nova Scotians and 6 avid TIers from SHER-TI in Quebec. It was a good show on our part and the Bostonians put on a great show for us! Hope we can equal or surpass that effort at our show next month. So far its looking good! More on ours later, here's the scoop on Boston's faire.....

MYARC, RAVE 99, Disk Only Software, Genial Computerware, ASGARD Software and TAPE(all the way from California! Good show!) were there. I represented Computer Download Unlimited and sold some Horizon Ramdisks down there and Art Green represented his company RAG Software. No doubt Jane, too, undertook some hard-nosed negotiating on behalf of Laflamme and Wrigley Wholesale. The trip down and back was made infinitly more enjoyable for all thanks to generosity of Dave Morrison and Ken McKenzie who respectively loaned Jane and I CB radios. The air was filled with "White Rabbits(me); Red Foxes(Jane) and Stubble jumpers(Lloyd)" quite gleefully all the way home! In addition to the afore-mentioned reprobates, we enjoyed the company of the Galenzoskis, Bentleys, Lanoys, Mssrs Taylor, Kuen and Earl and good ol' Ruth 'blueberry tea' O'Neill!

We collectively represented the Ottawa User Group at our table next to the Boston Computer Society's. Art Green, who's prowess with the 99/4A is making waves around the world was, in fact, a major attraction for many there during the day. Tom Bentley was invited to join Clint Pulley in the auditorium during Clint's presentation on c99. We met, for the first time, Mike Dodd, a young, well known and liked member of a user group in Kentucky and author of XB-Basher, a new release from Genial Computerware. Barry Traver was there, Peter Hoddie, Paul Charleton, Jim Horn and Jeff Guide, Chris Bobbit, Tom Freeman and even Terrie Mästers was there all the way from Los Angeles! (no doubt she was there trying to steal support from our stable(read FAIRE) to hers!) Just kidding Terrie; you know we luv ya! Our club's Vice presented Terrie with a desk pen set appropriately inscibed to acknowledge Terrie's efforts on behalf of all TIers. I think we actually caught her a little off guard, and I for one, always thought that was very close to impossible!!

User groups were represented from all over the eastern seaboard of the USA and several companys, who's names I'm unfortunely unable to remember, were there selling hardware for our machines. One last high note was the meeting of, and visit with, Albert and Joyce Visser who had come all the way from HOLLAND to attend Boston's faire!

Bouquet and brick time.... I call for thunderous applause to those companies, groups and individuals who, at considerable personal expense(both time and money), support these regional faires and fests! In the long, and even the short, run they go a long way toward keeping interest in the 99/4A high and growing higher all the time. Chris Bobbitt of Asgard, and Franz Wagenbach of TAPE from Ontario, California, for instance attend I think as much for the fun as for profit. Bricks should drop on the toes of companies such as Quality 99 software who, ruled by the almighty dollar, and placing little value on good will, sourly say; 'It was hardly worth my time and effort last year!'. They, by the way, are the only company I approached, of many, that replied in such an abrupt and surly manner to a request to attend our faire. Its very much a shame; he has many good programs to offer TI users. I'm sure if he was a lot more flexible in his price-setting for his products and in his dealer/distributor policy(or noted lack of), he'd have a smile on his face a lot more!

<u>OUR FAIRE MAY 16th:</u>

Disk Only Software, Not Polyoptics, Great Lakes Software and Horizon Computer Ltd. will be represented, if not by a member of their organization, then by Computer Download Unlimited. Each is still trying to find a way around a personal commitment that weekend so that they can attend our faire. Genial Computerware will be represented in the person of Barry Traver. He will also be doing a presentation on interfacing assembly language subroutines into a basic or extended basic program. Lou Phillips will be representing MYARC for the day and informing us all of Geneve's capabilities and status. RYTE DATA, I am told will be coming, though I haven't yet had first hand confirmation from Bruce Ryan. GENIE, represented by Scott Darling (from Washington State!) will be in Ottawa for the day. Chris Bobitt of Asgard software was hopeful but non-commital during Boston's faire. Local companies that will be taking booths include, so far, G-Plus and Mr Diskette, both of which are long-time loyal TI supporters. Guy Gourney will be selling his one-of-a-kind Maximem here again this year too.

In addition to Barry Traver and Lou Phillips we will also have Clint Pulley as a guest speaker in the amphitheater. Corcomp, Miller Graphics, MicroPendium, Dheins Home Homputer and Microsphere have contributed items for prizes during the day. More of everything is expected to commit before the big day next month.

We will have a classroom on that day dedicated to selling of used TI equipment and two other rooms for conducting of tutorials or workshops throughout the day. We have moved the 'rummage sale' to a classroom rather than in the cafeteria mainly because of the enormous amount of turnover they experieced last year. After the dust settled we found we'd lost two fairly expensive items. We'd like for that not to happen this year, thus the added security of a relatively closed environment. This year we will somehow mark each item sold as you pay for it. We also ask that if you are bringing items for resale at this booth that you supply us with a list of your items and the minimum price acceptable for each thing you bring. This year we will be withholding 5% of the purchase price for items over $100 and 10% of the selling price for items under $100 so price your goods accordingly.

We will make an attempt to entertain your children for you should you decide to bring them along. We'll be running some kind of contest with winners hourly, at a booth in the cafeteria for their amusement. We'll also have a refreshment booth available where snacks and drinks will be available during the show. We expect to have lots of literature available (ie: catalogs, brochures, software lists and newsletters) up for grabs on a first come first served basis. We're also gonna attempt to spread some fairware around Canada, a bit better than we did last year. Our fairware booth will be more visible this year and we'll put more effort into showing you what's available this year as well. I really hope we will be able to show fairware authors a lot ,better support for their effort than we did last year.

There will be two major highlights at our show this year. We will, at the very least, be a PRODUCTION MODEL GENEVE on display that day! There might even be some for sale...though my advice is; 'don't hold your breath!'. See this demo! It blew the socks off of a lot of sceptics in Boston and wowwed a lot of believers as well. Its impressive to say the least! The second major attraction will be a 3-way INTERCONTINENTAL TELECONFERENCE between Canada(us), Los Angeles and England. All three venues are holding a faire that day and we are all going to try to meet, at least once during the day on GENIE. If we succeed, and plans are going well, history will be made that day! Watch for it!

White Rabbit here; over and out!

FORTH TO YOU, TOO! SESSION 6

My original intention had been to write a few tutorials for our local 99ers to get them on their way with Forth. That seemed easier than to explain the basics over and over again. And besides, what I had been explaining had already appeared in condensed form in Millers's "The Smart Programmer". If some questions came up , they were easily resolved at our meetings or by a local phone call. Now that hese tutorials have been posted on

COMPU-SERVE I receive calls from all over and it looks as if there are a number of details yet to cover.
The question being posed most frequently indicates that some of you are not sure about Forth screens and programs. So let's clear this up:

Think of a screen simply as a means to record programs,which are, however, not limited to a single screen but may occupy as many screens as necessary. As an example, let's assume the following: You have made a copy of the TI-Forth disk, booted -EDITOR, -COPY, -PRINT and -BSAVE because these will be the only ones needed for the program you are going to write. You BSAVEd your autoboot starting on screen 22. The autoboot occupies screens 22 through 31. Screens 32 to 89 are now available for your program screens. (Remember, if you follow the' TASK 22 BSAVE with . (dot) Forth tells you the next available screen after the BSAVE.) The idea is to prepare a working Forthdisk with needed load options and then put one or more progams on it. We make the assumption that none of the unbooted load options will ever be needed for the programs on this disk, and therefore we can utilize the creens they occupy. Some people seem to have the idea that one Forth disk can do it all. That simply is impossible. You can have a collection of short routines on a disk but sooner or later you will run out of room if you try to maintain the original load option screens, unless you have two disk drives and put all your routines on a disk in drive 2.

The screens of a program are linked with the --> (load nextscreen) word which is placed at the very end of the screen. In this manner only nn LOAD is necessary to load an entire program (nn = number of the beginning screen). You will not find --> in Brodie's STARTING FORTH, apparently it is something TI added to their version of fig-Forth.

The line numbers are for reference only. They are not to be equated with BASIC line numbers. (Line numbers are superfluous in Forth because there is no GOTO.) However, words are compiled sequentially, i.e. starting with the first word of he first program screen and continuing down each screen line-by-line to the very last word of the last screen. Each word is added to the dictionary provided that any words within its definition can be found there. For example, : INVENTORY IN-STORE IN-WAREHOUSE + ; will not compile unless both IN-STORE and IN-WAREHOUSE have been compiled previously. (For advanced users who should not be reading this: Yes , this is not quite true, but remember that this is for beginners.)

For reasons which I don't understand it seems to be an obsession with some Forth programmers to cram their screens with utter disregard for legibility and clarity. If a program might take 9 screens they use every means to condense it to one less.I can see placing two short words on one line if (and that is a capital IF) one additional line would make it go into the next screen. But otherwise it is not good Forth style and it certainly does not make it easy for a beginning student to understand the program. In general, make it a practice to start words at the beginning of a line, indent the following lines if it takes more than one line for the definition. In long programs I even place the words being defined on each screen within the parentheses on line 0 so that I can find them easily with INDEX .

The normal number base (the one you're put in once Forth is booted) is DECIMA L. Invoke HEX and you can enter your parameters in hexadecimal numbers, but do not use > to designate them as such. You can also use binary numbers, simply put your system in that base with 2 BASE ! or better yet, define a short word like: BIN 2 BASE ! ; , I also define : DEC 10 BASE ! ; so I can go from base to base by entering only 3-letter words. In this manner you can use Forth as a handy-dandy conversion calculator.nnn HEX . will display a decimal number converted to hex. Just don't forget to reset the base with either DECIMAL (or DEC as above). In the same manner you can use any number base (Octalby 8 BASE ! etc), Forth will do the rest. Note: HEX is usually invoked within programs when putting parameters for character or sprite definitions on the stack.

Some of you report encountering problems with the words AT and TOP.

They are not standard TI-Forth words, instead John J.Volk, THE elder statesman of TI-Forth whose Data Disks have been distributed nation-wide, originated them to save wear and tear on his typing fingers (just kidding, John). They should be part of every one's dictionary:

```
: AT GOTOXY ; (AT is a lot shorter than GOTOXY)

: TOP CLS 0 0 AT ; (same as Miller's PAGE)
```

Time for one more question: How do I get out of Forth? Well,that depends. If you are through for the day, just pull out your disk (you might enter FLUSH first to make sure there are no loose ends in the buffers) and turn off your system. If you are going to continue, enter MON. This will return you to the TI color bar screen.

Lutz Winkler 619 277-4437

TI BASIC

by Steven Shaw

COLOUR AND SOUND

The TI99/4A allows you to set the screen, and the foreground and background colours of the characters, to any of fifteen colours, plus transparent. The transparent colour allows the screen colour to show behind a character.

CALL SCREEN is used to set the screen colour, and
CALL COLOR is used to set the character colours.
Note that COLOR is spelt the American way.

To Try:

```
100 FOR N=1 to 8
110 CALL HCHAR(N,1,24+N*8,32)
120 NEXT N
130 FOR N=1 TO 16
140 CALL SCREEN(N)
150 FOR DELAY=1 TO 300
160 NEXT DELAY
170 CALL COLOR(N,N,1)
180 NEXT N
190 END
```

In BASIC, characters are referred to by standard codes referred to as ASCII CODES. The ASCII code for the capital letter A is 65 for instance.These codes may simply be referred to as CHARACTER CODES.

TI BASIC allows you to define characters with the ASCII codes 32 to 159, and for colour purposes these are divided into sixteen sets. You may define different colours for each set, but all of the characters in that set must be the same colour.

Characters are defined with CALL CHAR(CODE,STRING$),where STRING$ is a string or string variable made up of HEXADECIMAL characters (0 to 9 plus A to F).

As each character occupies a grid of 8 x 8 dots, it can be defined by splitting it down the middle to form 16 rows of 4 dots.

Each possible combination of ON and OFF dots in a row of four can be defined in terms of one of the sixteen hexadecimal characters.

A character with one dot ON in the top right corner is defined as: "0100000000000000".

The definition is by row, first the left side then the right.

Each row of 4 dots can be considered a row of binary switches. The right switch is 1, the next 2, the next 4 and the leftmost switch 8. When a dot is ON, add its value to the others which are ON in the same row. You will obtain a unique number, from 0 to 15.

From this decimal number you change to a single hexadecimal digit (hexadecimal numbers have a 'base' of 16). The number 15 for instance is hexadecimal F.

You may purchase the EXTENDED BASIC module at some future date. As TI BASIC programs run faster in EXTENDED BASIC, it is worth noting that that language has only 14 character sets.

Therefore,if you use characters coded 144 to 159 in your TI BASIC programs, the programs will not run in Extended Basic.

There are several versions of the console around, and there have been slight changes in the relative shades of the colours used. If you purchase a program and the colours seem awful, it was probably written on a 99/4 or an NTSC 99/4A...you should be able to change the colours to something more suitable, and it will give you good programming exercise looking for the lines to amend.

If you wish to place a single character on the screen, use the CALL HCHAR command, it is slightly faster than CALL VCHAR.

Sound is produced with CALL SOUND(TIME,F1,V1,F2,V2,F3,V3,N,V4) where TIME is in milliseconds, F1,F2 and F3 are FREQUENCY in Hertz (cycles per second) and V1,V2 and V3 are volume (0 loudest, 30 quietist). N is a noise generator which provides some sound effects.

A CALL SOUND may use only one frequency if you wish: the second and third frequencies are optional and may be omitted. The Noise is also optional.

A CALL SOUND will occupy the computer for about 50 milliseconds, and then, even if the sound is still continuing, it will proceed with the next instruction. If it comes to another CALL SOUND, the computer will wait until the first has finished, unless the second CALL SOUND has a negative time, in which case the first CALL SOUND will immediately be terminated and the second CALL SOUND begin.

TI state that you can only have tones down to 110 Hz on your computer, but that is not quite the case:

Try:

```
  100 INPUT A
  110 IF A37 THEN 100
  120 CALL SOUND(2000,200,30,200,30,A*3,30,-4,0)
  130 CALL SOUND(500,200,30)
  140 GOTO 100
```

It would appear that the console can at least appear to go well below 110Hz. Try an input of say 50 or 60. If you find the sound interesting, try changing the -4 to -8.

Keep in mind that the computer takes about 40 milliseconds to process a CALL SOUND command.It is not possible to use this command to change the TYPE of sound produced.You cannot produce a piano or harpsichord sound for instance.

CALL KEY

Call Key is used to sense the use of a key on the keyboard and permits data to be entered without scrolling the screen (The INPUT command causes the screen to scroll). An ACCEPT AT routine using CALL KEY can be found later in this book. If you wish the computer to assume the ALPHA LOCK key is down, while a program is running, you can instruct it using CALL KEY, and avoid having to request the program user to ensure the key is down.

Use CALL KEY(3,K,S) in your program, and as soon as the program passes over it (no key has to be pressed) the computer will consider the alphalock key to be down, whether it is or is not. You resume normal operation with CALL KEY(5,K,S). These switching calls can use any variables you wish, and may be dummy calls or you may actually use them to obtain a key response.

In the appendix section you will find a list of the key codes which are available from the keyboard using the CTRL and FCTN keys.The normal keyboard, with and without using SHIFT will allow you access to characters 32 to 127 (a few of these codes do require the FCTN key).

IMPORTANT: If you use the split keyboard, CALL KEY(1.. and (2.. the value returned for keys X and M is only approximately zero, and although it prints on screen as zero will not equate with it. Instead of using IF KEY=0 THEN, you are forced to use IF KEY+1=1 THEN...

CALL JOYST

A program illustrating the use of this command is developed in a later section of this book. Please note that when using the joysticks, the alphalock key must be in the up position. If alphalock is down, the computer will not be able to sense when the joystick is pushed upwards. Using Call Key(3...) does not affect joystick operation, but you should not use CALL JOYST(3...) as this may prevent correct operation of the joystick.

ATN

ATN is a trigonometrical function which you may not need to use often, but in TI BASIC it may be used to obtain an accurate value

for the mathematical constant PI: PI=4*ATN(1)
  This and the other trigonometric functions provided work in radians. You may convert radians to degrees by using ATN:

```
      DEGREES=RADIANSO/4*ATN(1), or more simply
      DEGREES=RADIANS*45*ATN(1)
```

INT

  INTEGER is a numeric function you will use quite often. It removes the fraction from a number, so that 2.3 for instance becomes 2. It is frequently used with the RND function, and can also be used to round decimal numbers.
  For example, if B is a decimal number to 13 places, and you wish to print only the first two places, you could use:

```
  PRINT  INT(BO)/100
```

  If you wanted the last decimal to be 'rounded', the alternative is

```
  PRINT INT(BO+.5)/100
```

RANDOM NUMBERS

  Random numbers are useful in any program which you need to follow an unpredictable path. A program to display dice would be an example.
  The 99/4A generates pseudo random numbers. If you have a short program:

```
  100 FOR I=1 TO 10
  110 PRINT RND
  120 NEXT RND
```

  Every time you run the program the SAME 'random' numbers will be printed. This can sometimes be of value if you wish to be certain of the effect but still have the appearance of randomness.
  You may instruct the computer to start the list of 'random' numbers somewhere else, by adding the line:

```
    90 RANDOMIZE N    where N is a numeric variable.
                      The value of N determines where the random
                      numbers begin.
```

  Merely adding 90 RANDOMIZE, without 'seeding' the function with a value, will cause the computer to start at a truly random number, and if you run the program several times, different values will be printed each time.
  RND takes a value between 0 and 1, and is usually used in the format: NUMBER= INT(RND*MAXIMUM)+1
            The variable NUMBER will then take any integer value from 1 to MAXIMUM, including maximum. The maximum value of RND is the odd 1 to enable you to actually reach the maximum you want.
  You may prefer to have at the start of your program:

```
  DEF RAN(X)=INT(RND*X)+1
```

  Then when you want a random number up to say 12, you may enter in your program: NUMBER=RAN(12). You have created your own random function. Also see DEF.

SQR

  SQR is used to obtain the SQuaRe root of a number:
  A=SQR(4)   Many computers will not equate SQR(4)=2, or fail on some other comparison,due to internal rounding of numbers. Your 99/4A will equate all ten squares up to 100. Try it on a friends computer. You may not need to use this very often, but it is an indication of the numeracy of the 99/4A. Try:

```
  100 FOR I=1 TO 100
  110 A=SQR(I)
  120 IF I=A*A THEN 130 ELSE 140
  130 PRINT I;"PASSED THE TEST CORRECTLY"
  140 PRINT "NEXT VALUE OF I"
  150 NEXT I
```

NOTE that 1,2,4,9,16,25,36,49,64,81 and 100 pass the test OK. The failures are due to internal rounding, which still exists, but it is not quite so marked on the 99/4A as on other computers . If your program needs to make a comparison such as this, use the INT function to remove any (unprinted) fraction.
  In the program above for instance,there is some improvement by using INT(A*A) in line 120.
  There is a bigger improvement using
  110 A=INT(SQR(I))   which will remove from A any invisible fraction.

STRING EXPRESSIONS

A STRING is a non-numeric value or variable.A letter of the alphabet or a word or group of words may form a string.
NB: A NUMBER may also be a string:
2 is a number, "2" is a string.
A number (no quotation marks) must be used for mathematical operations. A string expression is therefore identified by quotation marks, or if represented by a variable, by a dollar sign after the variable name:
MESSAGE$="I WIN"

POS

The POS function is rarely used on other computers, but enables you to program very concisely on your 99/4a.
In the following example,CALL KEY is used to detect whether keys A B C or D are pressed, and control is passed accordingly. First,without using POS:

```
  100 CALL KEY(0,K,S)
  110 IF K=65 THEN 200
  120 IF K=66 THEN 250
  130 IF K=67 THEN 300
  140 IF K=68 THEN 350
  150 GOTO 400
```

In this case, the keys have adjacent ASCII codes, and it would be possible to use:

```
  110 IF S=0 THEN 400
  120 IF (K65)+(K>68) then 400
  130 ON K-64 goto 200,250,300,350
```

omitting 140 150.
However, in many games you may wish to test for keys which are well spread, such as AKESDXQP. The POS function can then offer the solution. Still using ABCD:

```
  100 CALL KEY(0,K,S)
  110 IF S=0 THEN 400
  120 ON POS("ABCD",CHR$(K),1)+1 GOTO 400,200,250,300,350
```

omit 130-150
If the key pressed is not in the string used in POS, then the expression has a value of zero, so one is added to enable us to use ON..GOTO,and the first transfer occurs if an unwanted key is pressed. In this case 'only' three lines have been saved, but if you wish to use more valid keys, you still only need to use three lines. This can be very useful in a program.
Although you may use a string up to 255 characters long, the POS function is unreliable for strings longer than 127 characters.

SEG$

SEG$ is used when you wish to print a SEGment of a string, or remove a part of a string.
TI BASIC uses only one command to segment strings, SEG$. Other computers use LEFT$, RIGHT$, and MID$, but you only really need the one.
It is used for instance in this DISPLAY AT routine taken from THE TEXAS PROGRAM BOOK.

```
100 REM
110 REM PRINT AT X,Y,M$
120 REM ROUTINE
130 REM
140 FOR J=1 TO LEN(P$)
150 IF Y32 THEN 180
160 Y=3
170 X=X+1
180 IF X24 THEN 200
190 X=1
200 CH=ASC(SEG$(P$,J,1))
210 CALL HCHAR(X,Y,CH)
220 Y=Y+1
230 NEXT J
```

(Set X and Y to the start position of your word, placed in M$. Then GOSUB this routine, and remember to add RETURN at the end to go back to the place in your program you left).

VAL

VAL is intended to make a number contained in a string available as a number, for use in mathematical operations. It changes "2" into 2, and may be used NO=VAL("2").. It is the opposite of STR$, used to change a number into a string:

A$=STR$(2)

The TI VAL function will change a string such as "2" to the numeric variable 2.
For instance: A=VAL("123")
This is of great importance when memory space is short, as a string variable representing "2" uses less memory than a numeric variable representing 2. This is explained in the section on Advanced Programming.
Please note that the TI VAL will only work if the string contains numbers only. It will not function for numeric expressions such as "2*3+8" nor if alphabetical letters are used such as "12 APPLES".

DEF
DEF is used to DEFine your own functions. TI Basic gives you great freedom in using this function- you are not restricted for instance to using a definition commencing FN as on some computers. DEF may also be used to create string functions.
Below are some examples of DEFs you may wish to use:
To print a random number from 1 to X.
At the beginning of your program type:

```
     DEF RAN(X)=INT(RND*X+1)
```

Then in your program, when you wish to use such a number, you type (for example)

```
     A=RAN(7) or
     A=RAN(LEVEL)
```

where LEVEL is a numeric variable.
Note that although X is used in the defining statement, you may use any number or variable when you use the new function. Although the function has here been called RAN, you may use any name you wish so long as it is not a reserved word(see TI manual for list).
The DEF statement is best used at the beginning of your program.
DEF PI=4*ATN(1)
Then when you wish to use PI in your program, just use the variable PI. NB: This has the same effect if you omit the DEF function. There is no advantage in using DEF instead of a simple LET.
Further uses of DEF may be found in the Advanced Programming Section.

ARRAYS

Arrays in TI Basic may have 1 2 or 3 dimensions.
A one dimensioned array may be thought of as a tower of building blocks. Each block has written on it a value or message. To find out what the message on block 3 is, we count upwards to the third block...
In TI Basic, normally the 'bottom' block is considered to be Block 0 (zero).
A tower of blocks may have blocks up to Block Number 10 without having to tell the computer how many blocks there are, but for a larger array, the computer has to be advised to reserve memory by using the DIM statement.
If you want to use an array of 16 values, before you use any of the values you must have the line DIM NAME(16) where NAME is the numeric variable for the tower of blocks.
We can place values in the array elements as follows:

```
100 DIM NAME(16)
110 FOR N=0 TO 16
120 NAME(N)=N*2
130 NEXT N
```

Now the variable NAME(3) has a value of 6. The variable is referred to in this way, with the 'level' after the variable name, and in brackets.
TI Basic differs from some other basics in the way it handles arrays in some important ways:
You must use a number in the DIM statement. A variable is not accepted.
Once an array has been DIMensioned, you may not alter the size of the array.
If an array variable occurs in your program before a DIM statement, the array is automatically dimensioned as 10. You cannot then use the DIM statement for that array.
Both numeric and string variables may be arrayed.
TI Basic does NOT permit you to use the same name for an arrayed variable and a simple variable. You cannot use both NAME as a

variable and NAME(3) say. Arrays reserve sections of memory, and you should not use a larger array than is needed. A numeric array will use 8 bytes for each element, regardless of the value in the element.

A string array initially occupies 2 bytes per element, but as you place strings in the elements,the space for each element will depend on the string you place in it (memory used= number of characters plus 1 byte).

If you can use string arrays instead of numeric arrays, you will usually save a great deal of memory.

A string array can be converted to a number by using the VAL function: eg A=VAL(SCORE$(2))

IF YOU DO THIS,ensure that you place the character "0" (zero) in any empty sectors to prevent possible program crashes.

As stated above, the TI 99/4A will assume the bottom sector is called Number Zero. Often you will find that having the base called Number One is quite adequate. If you do not use the 'zero' element,it is a waste of memory to have that element reserved, and TI allow you to instruct the computer to use a Base of 1.

The instruction to do this is OPTION BASE 1 (Note:The number is NOT in brackets!).

You can use the base 1 for all your arrays or none of them.Once set, you cannot reset the base, and as it is automatically set by any reference to an array before the computer finds the OPTION BASE statement, it MUST occur before any such reference.

In TI Basic therefore, FIRST: If you wish to set the base to 1, use OPTION BASE 1. Then, if your array is to have more than 10 elements, use DIM ARRAYNAME(NUMBER). Then you may use the array!

You may become more familiar with arrays by studying printed programs. (NB:Some other computers use the DIM statement to reserve memory for simple strings. This does not apply to the TI99/A.).

Two dimensional arrays are similar. It may help if you consider the first element name as a hotel floor, and the second as a hotel room. For instance: DIM NAME(FLOOR,ROOM).

Use of multidimensional arrays should be cautious, and careful consideration given to the use of OPTION BASE 1.

For example, a numeric array having dimensions (20,20) totals 21x21, or 441 elements,at 8 bytes each, that is 3.5k of memory used with just one DIM line! Use OPTION BASE 1 and the number of elements drops to 20x20=400. Times 8=3.2k, saving 300 bytes.

SUBROUTINES : GO SUB-RETURN

Whenever you use the same routine several times in a program (for instance the PRINT AT routine in THE TEXAS PROGRAM BOOK, reprinted in this chapter under SEG$), you may use GOTO and ON FLAG GOTO, but it is easier to use GOSUB to enter the routine, and when you have finished, RETURN will send you back to the program line immediately following the GOSUB with which you entered the subroutine.

Note: The computer stores the line number from which you GOSUB. This memory is only freed when you RETURN. If you GOSUB, ensure that you RETURN or you will quickly find a MEMORY FULL message appearing.

It IS possible to jump into a subroutine with GOTO provided you jump out with another GOTO, but that type of programming can lead to errors if you are not very careful, and should only be resorted to when you have absolutely no option. This should be rare. An ideal example of this can be found in a short routine to find out how much free memory there is after you have typed a program in. Add on to the end:

```
  10000 A=A+8
  10010 GOSUB 10000
```

Now type in RUN 10000. This little program will now run, and in due course the MEMORY FULL message will appear. Now type PRINT A, and press ENTER. The value of A is approximately equal to the memory space remaining.

(Your program may still not run even if there is a lot of space remaining: memory is still required to handle the values of the variables, the return addresses of GOSUBs and so on).

You MAY use GOSUB to enter a subroutine at any stage, you do not have to GOSUB to the beginning of a routine.

Within the limitations of memory, you MAY GOSUB from a subroutine, but keep an eye on the number of RETURNS !

The ON...GOSUB command is similar to the ON...GOTO command (which see), except that instead of a simple line transfer, the computer remembers where it has jumped from, and a RETURN will send it back

to the line immediately following the ON...GOSUB.
REMEMBER to watch your RETURNs.

THE DISK CONTROLS part 3

By **Michael Ballmann**

A while ago someone ask me the size the files in the disk directory. I have now come across the answer. The files are (decimal) 38 bytes long. Any other size will give an error. I once asked what all the names and uses of subprograms in the disk service routine. I now have the answer. If you want to know just ask.

Now this program needs:

1) Error checking so you can't try to read past the last track.
2) Some way to quit the program.
3) A display routine.
4) A way to switch to the second side.

I feel there is some merit in typing in programs so the only way this program is available from me will be on paper. It's less than 200 lines long so it should be easy to input it in with one sitting.

I have not used some of the commands available on the FDC (WD-1771). Here's how they work:

SEEK: The track register at >5FF2 must have the current track number, load the data register at >5FFE with the desired track, then send the SEEK command. The FDC will then issue the necessary step signals to arrive at the requested track. If the 'v' bit is set in the command the FDC will then check of the track number on the disk.

STEP: This command just steps the head the same direction as the last head move.

READ: Position the head on the correct track, load the sector register at >5FF4 with the desired sector, then send the read command. The track register and the track data on the disk must agree! Data will be at >5FF6. If there is a CRC error in the ID field this command will abort.

WRITE: This command is like the READ command except data is put at >5FFE to be written to the disk.

READ ADDRESS: When sent this command the FDC will read the next ID data field. The data will be at >5FF6 and will be six bytes long. The bytes will be:
1) Track number as it appears on the disk
2) Side number. Not use by the FDC.
3) Sector number.
4) Sector length.
5) First byte of the CRC error checking data.
6) Second byte of the CRC data.
If there is a CRC error; this command, or any which check the CRC, sets the status bit number three. Remember these bits are not numbered backwards like the bits on the TI.

WRITE TRACK: Just like the READ TRACK except it writes data instead of reads it. Don't forget the different addresses for read and write.

FORCE INTERRUPT: When this command is received the FDC stops whatever it's doing and goes not busy.

Now the status bits at >5FF0:
7) All commands- NOT READY
6) Write and type I commands- WRITE PROTECTED
   Read- 1st bit for data ID byte decode

```
5) Type I- Head engaged, Write- Write fault.
   Read- 2nd bit for data ID byte decode
4) Type I- Seek error
   Others- Requested data not found
3) All commands- CRC error
2) Type I- Track zero, Others- Lost data
1) Type I- Index (Begining of the track)
   Others- Requesting service of data register
0) All commands- BUSY

*************************************************
* SET THE  DRIVE TO TRACK ZERO                  *
*************************************************
       LIMI >0000        DISABLE INTERUPTS
       BL   @SENDC      SEND INSTRUCTION
       DATA >F700        SEEK TRACK ZERO
       BL   @CBUSY      WAIT UNTIL DONE
       CLR  R7           POINTER TO CURRENT TRACK
*************************************************
* MOVE THE ABOVE SECTION OF CODE TO THE PLACE   *
* RESERVED FOR THE ZERO TRACK ROUTINE AND DELETE*
* THE '*' AT THE BEGINNING OF THE THREE LINES   *
* MARKED FOR FUTURE USE.  DELETE THE STOP LINE  *
*************************************************
SETTRK MOV  R11,R13      SAVE RETURN
       MOV @TRACK#,R1   GET DESIRED TRACK NUMB.
       C    R1,R7        COMPARE DESIRED / ACTUAL
       JLT  SOUT         IF LESS THEN STEP OUT
       JEQ  RET13        EQUAL THEN RETURN
SIN    BL   @CBUSY      CHECK BUSY BIT
       BL   @SENDC      SEND COMMAND
       DATA >BD00        SELECT STEP IN CMD
       INC  R7           ADJUST TRACK POINTER
       C    R1,R7        IS DRIVE AT WANTED TRACK
       JNE  SIN          NO.
       JMP  RET13        RETURN TO MAIN PROGRAM
SOUT   BL   @CBUSY      CHECK BUSY BIT
       BL   @SENDC      SEND COMMAND
       DATA >9D00        SELECT STEP OUT CMD
       DEC  R7           ADJUST TRACK POINTER
       C    R1,R7        IS DRIVE AT WANTED TRACK
       JNE  SOUT         NO.
RET13  B    *R13         RETURN
                 * CHECK FOR DRIVES BUSY
CBUSY  MOVB @>5FF0,R0    GET STATUS
       SLA  R0,8         FIND BUSY FLAG
       JNC  CBUSY        IS IT SET? N
       B    *R11         RETURN
       END
```

Merge these three segments together in the order presented, then make the move and changes noted in this last segment, then assemble the result with the 'R' option. If when you assemble this code you want a printed copy; type in the correct print device. If you have PIO include a '.' after the name.

John Stocks has kindly sent me a source listing of his Mini-Memory graphics program mentioned in the last issue - it uses bit map graphics and QUICKLY produces a large number of odd designs! - due to the use of a look-up table to the trig, and the length of the source (5 pages without the tables!) I cant list it here unfortunately, but John is most happy to send a recording of the program to you - just send him a tape and return postage. The program plots 216 patterns, each with 1256 pixels. And the speed is splendid. For mini memory only. Send TAPES only.

John Stocks, 11 Stonehill Rd., Roxwell, Chelmsford, CM1 4PF

MYARC EXTEMDED BASIC Vn 2.11 continued from march
by **Steven Shaw**

If you have copied the disk to ramdisk, the module will load direct from the ramdisk, whatever it is called. The search appears to be: Is the file in RAMDISK? If yes then load, if no then what's on DISK

1.... and so on.
Once the system is loaded from any drive, it takes a look at DSK1 for a file called LOAD. You can bypass this (undocumented) by holding down FCTN and 4 (CLEAR) until the READY message appears.

MACHINE CODE at last...

Firstly, just as with TI XB, your m/c routines have a little less than 8k to fit into. You can enlarge the area slightly by typing in:
CALL INIT :: CALL LOAD(8194,32,130)

This is about the only time you MUST use call init! If you use CALL LOAD to load a machine code program, it checks to see if CALL INIT has been used and if not, does one for you.

Naturally if you wish to move the m/c boundary back a little, you must first use call load, or else when you load your file, the system will see no call init has been done, do one, and reset the boundary!
Now then...
1. Programs in machine code written for TI XB loading and running MAY not function.
If you have any XB LOADing programs they will
not operate correctly (eg the hidden code LOAD program with FUNLWRITER- and similar programs created by SYSTEX or ACE)
2. In general, machine code programs written for editor/assembler will work with the exceptions found below.
DF80 files may be too long to load in directly. Longer files can however be loaded using the UTILITY option of FUNLWRITER. (the error message MEMORY OVERFLOW tells you to try Funlwriter!).
MEMORY IMAGE files can usually be loaded using OLD or RUN.
Given that several machine code files in DF80 format are likely not to load into Myarc XB, for a whole variety of reasons, you can use them to create a memory image file which will run directly from Myarc XB.
By way of example, lets look at a game file which we shall call SNEGGIT, in DF80 format, which refused to load with Myarc XB. We then load Funlwriter and use the LOAD AND RUN loader option to load a little utility called SUPERSAVE (available from me) and use this to transfer SNEGGIT to memory image, choosing to carry over NO utilities. Yep, the new memory image file will now load directly with Myarc XB using a nice simple command like: RUN SNEGGIT.
You will find that some machine code programs - in either format - make assumptions regarding the operating environment which do not apply. Two areas of difficulty:
Comment on Version 2.1: -1. GRAPHICS: If the screen is blank or the graphics are not what they should be, try loading the program with the FUNLWRITER Utility option this restores the environment. I have only found CUBIT to fail, and even then the failure -2. SPRITES? The question mark indicates I have not traced the problem area, and have not cured it. Although many programs operate quite happily when loaded with Myarc ExBas (with or without Funlwriter) there are a few which have disconcerting side effects, apparently connected with sprite motion.
For Version 2.11, Myarc have provided an addition DF80 utility program called TIVDP which sets the VDP registers to correspond to TI XB, this apparently being the cause of the problem. It seems to cure almost all the faults I found, but I still found a black screen when loading MGR3 from the Funlwriter disk (cure: use CALL GRAPHICS(2) first- MGR3 file assumes a 40 column set up) , and FROG HAVEN by Fully Assembled Software had invisible logs for the frog to hop over - not easy!
The format is: CALL INIT :: CALL LOAD("DSK1.TIVDP") :: CALL INIT :: CALL LOAD("DSK1.PROGRAM")
and it is NOT a mistake to show CALL INIT twice above!!! In the event of a LOAD error you will have a blank screen, and need to type NEW to recover the standard MYARC VDP register values.
If using TIVDP fails, then try switching to one of the other graphics modes (2 or 3) before loading the machine code program. The DM1000 file MGR3 on the Funlwriter disk is an example: try to load it in GRAPHICS MODE(1) and you meet a blank screen - although it will function correctly! To see what you are doing you need to type in: CALL GRAPHICS(2) :: RUN "DSK1.MGR3" and off you go.
3. FORTH: Do not use the original TI loader for Ed/AS, it will not function. Instead use the later modification for mini/memory (the Universal loader).

4. FUNLWRITER:
The LOAD program with Funlwriter will not work. UTIL1 seems to use an interrupt routine and crashes if you try to run it directly. The easiest way into FUNLWRITER is with:
CALL LOAD("DSK1.LDFW") - the file LDFW is part of the Funlwriter package.
Funlwriter remembers its source drive number, and resets various things to allow you access to otherwise difficult programs which may be anticipaing an environment not matched by Myarc XB.
5. Any program with odd graphics or no graphics - try setting GRAPHICS mode to 2 (CALL GRAPHICS(2)) or loading TIVDP as above, or loading with FUNLWRITER loaders. Some programs make massive assumptions about the state of the console, when perhaps they shouldn't.
6. NEW HORIZON UTILITIES: If you have the source code for the several XB utilities which the New Horizon group have given us, you need to change any EQUates for:
PAD, GPLWS, SOUND, VDPRD, VDPSTA, VDPWD, VDPWA, SPCHRD, SPCHWT, GRMRD, GRMRA, GRMWA, SCAN, XMLLNK, KSCAN, VSBW, VSBR, VMBW, VMBR, VWTR, DSRLNK, LOADER, NUMASG, NUMREF, STRASG, and STRREF to external REFerences.
For example, change:

```
NUMASG EQU >2008
STRREF EQU >2014
```

to:

```
    REF NUMASG,STRREF
```

Then reassemble. ... in case the normal labels are not used, just watch out for any EQUates to >20NN. You will then need to look up the name!
7. INTERRUPT ROUTINES - as are used in graphics dumps such as DUMP for instance, or clock routines - do not eem to be available. There is only one vector available for interrupt routines and it looks as though Myarc XB is using it. It may be possible to adapt the environment so that interrupt routines can be used - my knowledge is insufficient to comment on this properly.
-Interrupt driven clocks: I have been able to load and run such things with Myarc XB, both from TI Forth and from XB itself, but in both cases with bad side effects. In Forth, using MON caused a lock up instead of a graceful return to the title screen! From XB, using GCHAR returned values 96 (>60) greater than the true value and NEW tried to load LOAD from DSK1... my observation therefore is that any program using the interrupt vector >83C4 is more than likely to misbehave or totally crash the system. A few programs use this address as an alternative to TI's standard auto-start.
8. CRU ADDRESS COLLISIONS: The Myarc ram card (needed for Myarc XB) has its own DSR in EPROM and some programs may disable it! - for instance, the disk copying program MASS COPY must be amended to run with the Myarc Ram Card with either version of Extended Basic!
9. The Myarc ExBas does a great deal of memory paging, and it is possible that a machine code program could interfere with this - may be the sprite problem listed above?
--- the Scratch Pad Ram (256 bytes) is used differently to TI XB but no data has been published as yet. Any machine code program using this area MAY have problems.
Cassette handling has not been included - possibly as a result of differing VDP mapping- or the thought that as few purchasers would want it, it was not worth the trouble of writing the code!

CONCLUSION:

Although there will be some programs that you cannot run with Myarc XB, this is inevitable with any upgrade - in some cases you can get round the problem with just a small bit of a rewrite. We are not looking at a Myarc failure here, but at programmers who have taken short cuts- in some cases taking advantages of bugs TI left in but Myarc have removed! In some cases a particuler upgrade demands that a degree of incompatability is introduced. Life is never easy! But by and large, I have had to use alternative modules to run very few programs.

Exceltronix

April Specials

Star NX-10........$349.

Pan. JU455........$135.

EXCELTRONIX COMPONENTS AND COMPUTING INC.
217 Bank Street, Ottawa, Ontario K2P 1W9 (613) 230-9000

```
*****************************************************************
*                      TRADING POST                             *
*                      ------------                             *
*                        FOR SALE                               *
*  BROTHER HR-5 THERMAL TRANSFER PRINTER....................    *
*  Prints on thermal and regular paper, 40-132 Column,......    *
*  Parallel Interface, Compatable with TI-ARTIST, TI-WRITER     *
*  etc.......$150.00                                            *
*  Contact....Lee Van Biesbrouck....................521-9290    *
*                                                               *
*  32K MEMORY EXPANSION CARD.............MAKE AN OFFER......    *
*  CONTANT.....DAVID MAJAURY..................990-1922          *
*                                                               *
*****************************************************************
```

momin PRINT

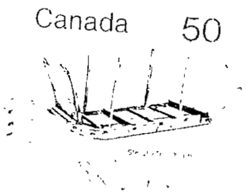
OFFSET PRINTING • LAYOUT • PHOTOCOPYING

• Quality printing at low prices
• Special prices on photocopies

OPEN
MON
TO
SAT
9:00 A.M
9:00 P.M.

ORLEANS
837-1844

2451 St-Joseph Bl Orleans
Lower Level in Le Centre
K1C 1E7



FROM

P.O. BOX 2144, STATION D, OTTAWA
*** ONTARIO, CANADA K1P 5W3 ***

EDMONTON 99er USER'S GROUP
P.O. BOX 11983
EDMONTON, ALBERTA
T5J 3L1