

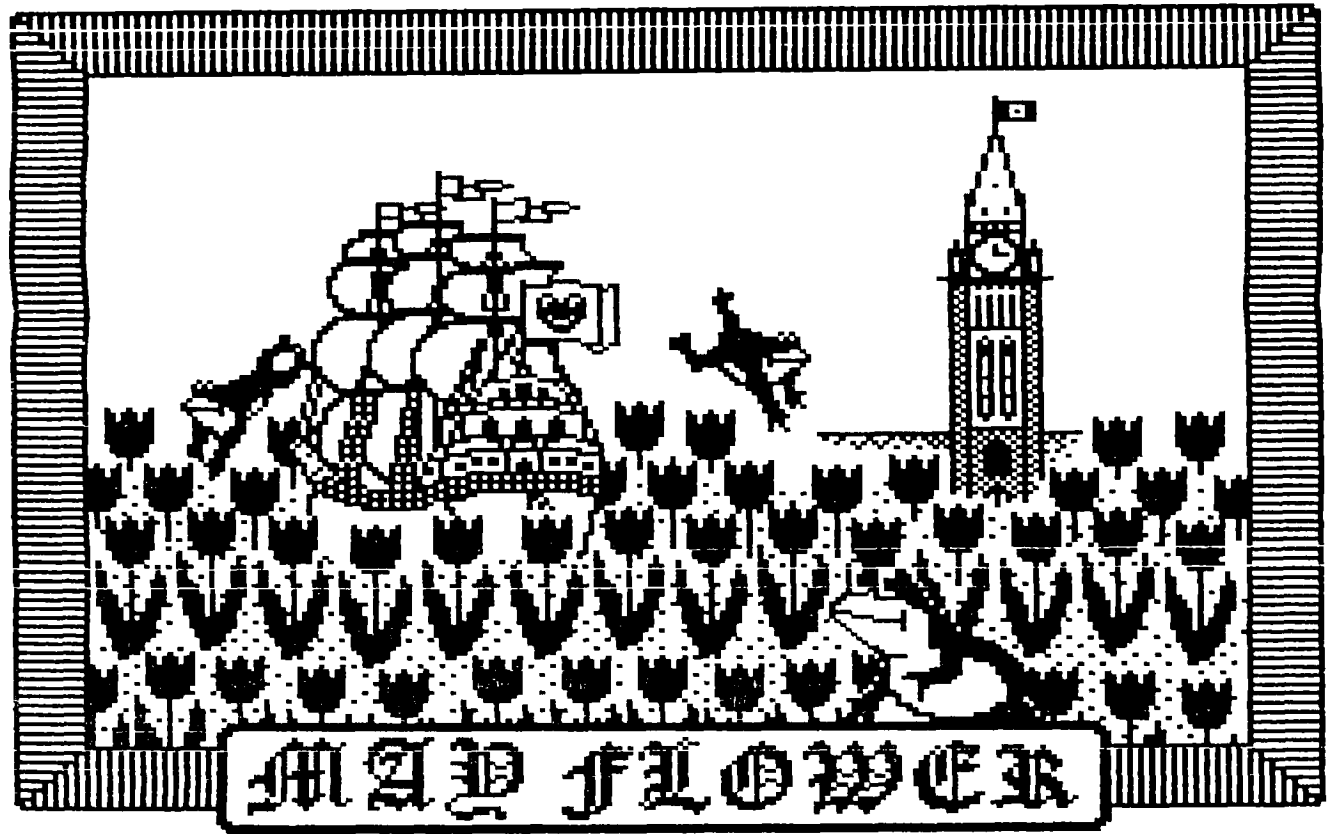
065
8805
ottawa



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



VOLUME 7 NUMBER 05.....MAY 1988



DON'T FORGET THE MEETING -- MAY 3, 1988

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

COMING EVENTS

- May Meeting: May 3, 1988 Merivale High School
7:30 p.m. Back in the Cafeteria
- June Meeting: June 7, 1988 Merivale High School
- Beginners Assembly: May 18, 1988 Dick Piché's home
7:30 p.m. Contact Bill Sponchia for
further information.
- Summer Barbecue: June 25, 1988 Marg & John O'Connor's
home. Bring all your own
everything. Watch for further
Details
- Newsletter Deadline: May 15, 1988

Mr. Diskette



INTRODUCES...

NX-1000 Multi-Font

The NEW and INNOVATIVE NX-1000 Multi-Font is a versatile and reliable printer for home, school or office.

\$349.

- 144 CPS Draft/36 CPS NLQ (18/23 Character Matrix) • Both IBM™ and Epson™ Compatible • Generous 4K Buffer
- Standard Parallel Interface • High Resolution Graphics • Time Saving Paper Parking Feature (Feed single sheet without removing sprocket paper.) • Easy-to-Use Membrane Control Panel • Push Button Control of 20 Type Styles • 1-Year Manufacturer's Warranty



star
micronics

DOWNTOWN
105 O'CONNOR ST.
232-5203

NEPEAN
1600 MERIVALE RD.
ACROSS FROM THE K-MART **727-0180**

EDITOR'S NOTES

from
Ruth O'Neill

Once again, the cover art comes to us courtesy of Lucie Dorais. Thanks, Lucie, for another job well done.

At the April meeting, Jane Laflamme read a letter to the club from the Hillel Academy, thanking us for our \$100.00 contribution in memory of Berry Minuk. Their computer department was able to purchase a number of software packages, and a copy of the letter has been forwarded to Berry's family.

The May meeting is back in our usual location, the cafeteria. Hope to see all of you there... and bring a friend!

During a recent Compuserve conference, the subject of hostility towards Geneves and their owners came up. It seems that in many clubs (including our own), there are some people who feel very strongly that Geneves should be banned from the club, or at the very least should not be welcomed. While this sentiment is not held by the majority (in our club, we amended our constitution to include Geneves), the fact that it exists at all puzzles me. What has created this? Why is it continuing? Is it spreading? I certainly hope not.

During the conference, I drew an analogy that seemed to be very well received. Most of the members in the Ottawa TI99/4A Users' Group now have at least one disk drive, but a few years ago, this was not the case. At that time, an expansion system and even 1 drive was very expensive, so many people were content (or at least had to be content) with cassette-based systems. Those who were fortunate enough to be able to upgrade to a diskette-based system were able to run programs that the cassette users could not, while still being able to run everything they could before. Did the cassette users express hostility towards the disk users over this? I don't imagine so, but I have no first-hand knowledge, since I am a relative newcomer to this group. Isn't the situation with Geneves very similar? A Geneve owner can run some new programs that someone with a 99/4A cannot, but is still very interested in /4A software and hardware. I think the most reasonable way to view the Geneve is as an upgrade to the /4A -- in fact, not even as expensive an upgrade as expanding to a disk-based system was a few years ago. We would like to bring more cassette users into the group to swell our ranks. How can we expect them to feel welcome and supported when they can do less than we can, if we are unwilling to share with those who can do as much or more?

Even if we don't care to think of the Geneve as an upgrade, why should we feel so antagonistic towards them? Are TIers jealous? It's a human emotion, but it would be a shame to allow it to undermine the unique sense of community that has developed in the TI world. Does the Geneve threaten the /4A? I don't think so. I believe that it has provided programmers with something new and exciting to work with, so they'll stay in the community longer than they might have otherwise. The trend right now is not towards programming for the Geneve exclusively (after all, the greater market is still the /4A), but towards new programs that will work on both machines, but that take advantage of the extra features of the Geneve. Even the programmers who have been devoting their time almost exclusively to writing for Myarc have not been lost to us nearly as much as they would have if they had gone to another family of computers. They have been given an opportunity to remain in the community while developing their skills professionally. I'm quite sure that this will show in programs they write for the /4A in the future. In any case, how can we feel threatened, knowing that over 7,000 /4A's were sold in the Ottawa area alone? Not even a quarter of that number of Geneve's have been sold yet in total.

I'm not advocating buying a Geneve, by the way. That's certainly something to be determined by your own needs and taste, and I don't have any plans to get one myself just now. I am advocating a greater degree of tolerance towards Geneves and their owners -- of welcome, in fact. After all, we all came from the same "family".

The President's Two Cents' Worth
by Michael Taylor

As I write this, I reflect that only a week ago nine of our group were hosting an informal reception in Bob Boone's motel room in Lexington, prior to the New England Fayuh. We repeated the exercise twenty-four hours later with an even larger group of TIers. I believe that these meetings were every bit as important to us as the show itself. We had fun! We had fun renewing old acquaintances; we had fun meeting new acquaintances. It was apparent that our guests had fun, too -- for the very same reasons.

Now, the talk at these functions was not confined to discussions on the TI computer, nor even on computers in general; but rather, discussion encompassed a wide range of topics -- such as one might expect at any reception or garden party.

This leads me back to a comment made over a week ago on our Texlink bulletin board, during heavy debate concerning the future direction our club should take. The comment, espoused first, I believe, by Lucie Dorais, was that the friendships we form through the users group are as important as, or even more important than, the technical things we learn about our computer. That friendships can survive past the "death" of the TI is evidenced by the many members who, having buried their 99/4A and taken up with a new mistress, continue to visit with us -- either in person or via the BBS.

As reported to the April general meeting, your executive addressed the question of opening up the club to other computers. The overwhelming consensus was that we should remain as we are -- a club devoted exclusively to the TI-99/4A (and compatibles, such as Geneve).

Since that time, your Vice-president and I have had the opportunity to talk to the President of a TI users group which did open its doors to other computers. At the New England Fayuh, we talked with Sandra Ross, of the North Eastern 99ers. Her group was faced with the same dilemma our group faces -- to remain exclusively TI, and so lose members, and friends, who have switched to another computer, or to welcome other computers into the group, and so risk overwhelming, and perhaps losing, the TI. Not without some trepidation, they chose the latter solution. Sandra says the move has proved successful; the group is alive and well, and the TI continues to thrive. Special interest groups (SIGs), each devoted to a particular computer, were formed within the main group. Certain restrictions are placed on these SIGs to ensure that they pull their own weight and do not bleed off the TI effort. For example: each SIG is responsible for its portion of the club's newsletter, and each is responsible for organizing its own demonstrations at meetings.

Perhaps, in light of the experience of the North Eastern 99ers, we have been too hasty in rejecting our friends who ask to have a niche for their computer in our group. Our old friends do miss us -- just look on our BBS at the flurry of bravado generated by former TI owners now lonely in the cold world of Big Blue and desperately seeking warmth amongst their former compatriots.

Speaking (writing?) of friends, do mark your calendar for June 25. That is the date selected for a club barbecue to be held at John and Margaret O'Connor's place in Cumberland. This will be in lieu of the annual picnic, previously held at Hog's Back, so plan to attend. There will be gas barbecues. Bring your own food to grill, your own beverages to drink, your own chair in which to sit; and, since the O'Connors have a pool, bring your own towel and swimsuit. Look for more details in the June newsletter and on the BBS.

BROWSING THE LIBRARY
--with DAVE MORRISON

As I (feebly) attempted to indicate at the April meeting, the Library is not a producer of software; it can only distribute material that has been donated by Members. At the moment, the Disk of the Month is mainly of interest to new Members who have little or no software. Another problem in the production of the Disk of the Month is that it should appeal to a majority of Members and for that reason I will endeavour to provide an Alternate Disk of the Month as often as possible.

Commencing in May or June, I intend to bring to the monthly meetings, a selection of Library disks complete with a printed disk catalogue and one printed copy of the documentation. At the "break", I hope that members will take a little time to examine the disk catalogue and the documentation to see if any of the programmes are of interest to them. As the Library has a limited number of blank disks, I shall only be offering three copies of each disk, so first come - first served!

Please remember that the Library is an important source (or potential source!) of revenue for the Group. When you support your Library, you support your Group.

For those of you that utilize our fine BBS (logging-on with the equally fine TELCO terminal emulator!), I read the Board several times a day, and for those without a MODEM, my telephone (737-4889) very rarely fails, so any questions you may have, or advice (badly needed) to offer, please contact me at any reasonable hour. The BBS can handle those who choose to pick an unreasonable hour!

Until the next meeting, or by BBS'ing, or by telephoning,

Dave

MAIL CALL ! ! !

As our club and our orphaned machine moves into the 90's, our out of town members will play a larger and larger role in comprising the total club membership. Already 'out of towners' comprise approximately one third of the club membership. It becomes increasingly important, therefore, that we hear from you, the 'out of towners' view, on the direction the club takes.

I invite all our out of town members to write to the club. We want to hear from you -- we need to hear from you to ensure the continued success of the club. We want to hear from you even if your letter is only 'chatty', informing us of your current equipment or equipment needs, the software you use or need. Or perhaps you belong to a small group. How many people do you know that have TI's, in your area?

TI clubs that are in existence now, more than at any time previous, are in the natural process of extending themselves to form one TI community at large. This is the strength, camaraderie, and legacy of the little orphaned computer.

The more the lines of communication are kept open, the more we will grow and the more we can serve you, the member, who may be feeling isolated from other TI'ers.

The mail is virtually caught up at his point. I have about 4 outstanding letters yet to answer. Whatever your TI needs, your club has the expertise to help. I encourage you to write and be an active long-distance member.

Stephen Bridgett

HINTS, TIPS & ANSWERS by Bill Sponchia

Hello again. As reported last month by the Editor, this column has had a name change. The reason was that questions were few and far between and therefore I would have been forced to "make-up" them if I wanted to continue writing a column. To avoid that (because I believe that it is unethical), I decided to change to this, thus giving me the freedom to give out what I believe may be useful information. Of course, if you have a question let's still have it. I will do my darndest to answer.

Please be advised that much of the information you will see in this column has appeared in newsletters of other User Groups. There is a lot of valuable information floating around. I say this so that you will not think that I am claiming to be the originator (although in some cases I am) of all the information. Having said that, I will now state that unless I am passing on a major excerpt from an article or I wish to let you know where further information is available, I will not be noting where the hint, tip or answer came from. The reason for this is twofold. The first is that many of the tips and hints have been around so long that the "original" originator's name has been lost; and the second is that most of the information is really available in manuals, etc., and the tip or hint has only been put into a more readable form. To all or any person whose ideas or thoughts are repeated here please accept my thanks and the thanks of all the other readers.

PRBASE:

Hint 1 - When doing a Selective Indexing, if the character combination you wish to search for contains a blank, you must insert a "?" in place of the blank. Otherwise the selection will only be on the character combination preceding the blank.

Hint 2 - I found that setting up the printer codes in the Report Section did not work; however, if the printer codes were set up in Section 6 "Set Printer Codes" and before printing out the Report you did a Command C to select the printer codes, then everything worked okay.

"DISPLAY AT" in Extended Basic:

Hint 3 - This statement will clear from the column number indicated to the end of the row. Therefore by entering DISPLAY AT(5,1):" will allow you to clear a given row. To avoid clearing the entire line by entering the size (eg SIZE (5)) will allow for the retaining the balance of information on the row.

MULTIPLAN - Printer Codes:

As promised a few months ago, I have finished off a Multiplan template to allow for the inputting of printer codes into your spreadsheets. It will be uploaded to the BBS in the near future. For readers who wish the template and don't have a modem or wish it before it is uploaded, please feel free to contact me to get your copy.

That's all for now folks... this is Bill signing off... No Carrier!!

RAIDERS OF THE LOST...

This month, we will try to get answers to the various questions asked in this column from the beginning of the year.

GAMEAUCRATS
Henri Monat

BACKGROUND

In my opinion, giving straight answers to those questions is as flat and boring as hiking in the desert. I therefore chose to find a cryptographic program i.e., a program that would cipher and decipher the answers. I came across many programs, but none of them were satisfactory, in that they were not responding in what I was looking for. Then our librarian (Dave Morrison) found this exceptional program lost in the 33 game diskettes that our library is keeping. The program is called BAZERIES and was made known to us by...

BRUCE CARON STRIKES AGAIN

Bruce Caron translated the Bazerieries' program originally written by Rinaldo F. Prisco for another computer into TI Basic. Here is how Bruce expressed himself in the June/July 1983 issue of the Ottawa Newsletter:

"For those wishing more details on this subject, we recommend that you purchase the June 1983 issue of Byte magazine. To encapsulate for those who can't get it, the Bazerieries Cylinder was developed by Commandant Bazerieries of the Black Chamber, the French Army's Cryptographic Department, in the early part of the nineteenth century. It basically consists of a cylinder divided into 20 disks (similar, if you want, to a date stamp). Each disk contains a combination of 20 letters. By rotating the cylinder, you make a pattern of 20 letters. In order to use it, one simply aligns the letters to make up the sentence to be encrypted. 19 other combinations emerge on the cylinder, any one of which can be used as the "ciphertext". To decipher, just reverse the procedure: enter the ciphertext and the original text will emerge on the cylinder. This of course will only work if both parties have the same settings."

This program is a beauty. We recognize the simplicity and the quality of product Bruce used to give us.

ENCODED ANSWERS

To decipher the following answers, you must use BAZERIES. You will be asked to enter a key word; type OTTAWA in capital letters. I doubt that any other TI program or key word would decode the following ciphertext. The same letters in the original text may be represented by different letters in the coded text.

Good luck and have fun.

TI-RUNNER (October 1987 issue):

Question: how many screens can be related to TI and which ones?

Answer: V D X H H K E J K Z L J G K T R P C F H
Z Z J C J M E C J D A J R H X Y S P M Z
M X G C R P

MINER 2049er (November 1987 issue):

Question a): who is James?

Answer: H V P T N J B P F F M X

Question b): how to make screen 008 by going on only 5 of the 6 guarded platforms?

Answer: T P R G J L L G O S O R R U Z Y U E Y E
F V Q E W V G J N S W F A I S R A O J V
A Y S N G H P O Q L I X N M M B N D H D
U V P G D O E Y V G Z X P N L W Y Z B R
N Y O U O D R T W M E Q W L A N B

Question c): how to do the same and to finish upstairs?

Answer: D N S C P N U N T N R Q B D Y U D M A Q
D Z J V I F W O Y L K J A I N N L D M P
D O K J B Y K I K M S S J J C W O K S C
D K A O X P V V Y P G M J P F X R R Y B
Q Q R R V R B C A X O J N U B R I E T C
T H Z S B N

BACK TO BOSTON!
by Ruth O'Neill

As our president noted earlier in this newsletter, 9 people from our group went down to Boston for the faire this year: Steven and Lorraine McWatty, Bob Boone, Jane Laflamme, Michael Taylor, Ken McKenzie, Charles Earl, Ralph Kuhn, and myself. The year the users' group shared a double booth with Computer Download Unlimited, and Charles Earl & TELCO. We all took turns minding the booth so that we could enjoy the faire and catch some of the presentations.

There were two kinds of presentations available -- Talks by some of the well-known people in our community, and more workshop like sessions on various subjects. I didn't get to any of the workshops, but I did manage to sit in on several talks -- Barry Traver's "We're out of the Orphanage", Jim Horn's "Black hats I have known and worn" (on his favorite subject, telecommunications). At the end of his talk, Jim had a draw for a Compuserve i.d. with a \$15.00 credit, and the winner was Ralph Kuhn. Unfortunately, I was minding the booth when Warren Agee gave his presentation on First Base, the new data base coming soon from Genial Computerware. I had planned my schedule to be able to attend, but things just didn't quite work out. Jane managed to catch it, however, so I'm sure she'd be glad to field questions about it. Jane?

There was also an impromptu panel discussion moderated in a very professional manner by J. Zittrain. Our Jane Laflamme took part along with Chris Bobbitt (Asgard), Lou Phillips (Myarc), Barry Traver (Genial TRAVELER), and a number of others, representing a good cross-section of users' groups, sysops, programmers, and companies. There was also a fair amount of audience participation in the discussion, which ranged from the future of the TI and the role of users' groups to such issues as copy protection and piracy.

While I don't think the fair was quite as exciting this year in terms of the number of vendors or the bargains available, the presentations made the show for me, along with the chance to meet people I'd only talked to on Compuserve, or spend more time with people I met at our Fest here in March. This is what the TI community is all about. After the fair, there was a dinner at a Chinese restaurant, quite well attended. Both Friday and Saturday evening, too, a number of people met in Bob's motel room to socialize.

The trips there and back went well, too -- there rather better than back. With the CB's, all three cars managed to stay pretty well together on the way down, but we managed to get separated on the way back. It was just as well, though, since Bob and Jane's cars spent a long time at customs (New fees schedules, so it took some time for the customs officers to figure out what duties to charge, I understand) and we in Ken's van just breezed through, to make it home just in time to log on to a Compuserve conference. It was a super weekend, but it's great to be back home, too. Congratulations to Peter Hoddie and the others involved for a fine "Fayuh".

EXPANSION PORT INTERFACING: Part 6.
By David Caron

This is the last of four articles on the power of the 9901 interface chip and also the final article on the subject of expansion port interfacing. You will need to refer back to the pinout diagram in the February 1988 newsletter, as well as the March 1988 newsletter in order to properly understand exactly how the CPU and the 9901 "talk" to each other. In the March newsletter, there is a diagram of Register Twelve. This is called the CRU base address and is used to define which 9901 is used in A3 to A7. A10 to A14 are usually left zero with the actual CRU bit being defined by the variable X in the five assembly instructions. For example, if X was 14, A10 to A14 would be set to 01110 + the bits 10 to 14 in register twelve; in this case they are zero, so A10 to A14 would be 01110. This is explained in the Editor-Assembler Manual.

The three main lines used are CRUIN, CRUOUT, and CRUCLK.

READING: (from 9901 to 9900 (CPU)) When the CPU wishes to read from a particular 9901, it first sends out the address. A3 - A7 will indicate which 9901 is to be accessed, and A10 - A14 will indicate which bit in that particular 9901 is to be set or reset. A decoder chip must be used to decode the A3 - A7 addresses, and the output would go to the CE (pin 5) on the 9901 to turn it on. This could be done with some NOT gates and a 5 input AND chip.

For example, when the CPU wishes to read something from the disk controller, it will set A3 - A7 to 10001 (disk controller cru assignment, refer to the Editor-Assembler page 407). This will turn on address decoding circuitry which will enable the disk controller 9901. That 9901 will AUTOMATICALLY put the data on the CRUIN line, depending on the addresses A10 - A14 which are set up by the CPU along with A3 - A7. After the CPU sends out the address, it will read the condition of the CRUIN line from the 9901. If the line is high, the CRU bit being accessed is high and vice versa. Notice that the CPU memory will remain inactive because DBIN (active low) is always high. However, when the CPU is really accessing memory, the particular 9901 will still turn on and set the CRUIN line high or low. There is nothing wrong with this, since the CPU will simply ignore the data on the CRUIN and read the data on the data lines instead. Therefore the CRUIN line may be almost always active.

WRITING: (From 9900 to 9901) When the CPU wishes to write data to a particular 9901, it sets the address lines to access a particular 9901 and the particular CRU bit in that 9901, like the procedure above. The 9901 will immediately set the CRUIN to reflect the condition of the bit accessed, but we are not interested in CRUIN, so the CPU ignores it and instead sends out the data on the CRUOUT line, which happens to be the same as A15. This does not matter, though, since the CPU memory is inactive because WE (Write enable, active low) stays high. In an case, after those two steps the CPU will set the CRUCLK (active high for 9901, but active low from CPU -- a NOT gate must be used in between line high) low (from the CPU) and the 9901 receives the data from the CRUOUT line and the CRU bit in the 9901 is set or reset depending on the state of CRUOUT.

EXAMPLE: Say you wish to find out whether the Alpha lock is down or up on the keyboard. In order to do this, CRU bit 21 in the console 9901 must first be set low to enable the alpha lock key. Then CRU bit 7 must be read. If that bit is low, the alpha lock is down (CRU bit 21, pin 5 is connected with CRU bit 7, pin 34). If that bit is high, the alpha lock is up. CRU bit 21 should be set high again to avoid problems with the KSCAN routine.

STEP 1: Set CRU bit 21 low.

-Since we must access the console 9901 then A3 - A7 should be set low (A3 must be low). Therefore register 12 should be set to:
0000 0000 0000 0000 or cleared. The instruction SBZ 21 must be executed to access bit 21. When the CPU performs this instruction, it will put the number 21 in A10 - A14 and add R12 so that the address lines become: 0000 0000 0010 1010 .

The CRUOUT line is set low since the SBZ instruction is used.

The CRUCLK line is enabled and the internal 9901's CRU bit 21 is reset (zero).

STEP 2: The instruction TB must be used to read the 9901. TB 7 is executed to read bit 7 (R12 must be cleared of course). The address lines become: 0000 0000 0000 1110 . The console 9901 immediately puts the condition of CRU bit 7 in the CRUIN line and the CPU reads it, setting the equal bit in the status register if CRUIN is high or resetting the equal bit if CRUIN is low. Therefore, if the equal bit is low or reset then the alpha lock is down.

STEP 3: CRU bit 21 must be set high again in order to allow the KSCAN routine to work properly.

Based on what you know now and your knowledge of simple gate chips, you should be able to successfully interface your TI computer to an external circuit fairly easily. If you wish to do it through the memory address system, I would suggest that you use addresses >4000 to >5FFF. These addresses are used to transfer data to and from the cards in the expansion box: Disk controller, Ramdisk, RS232, etc. However, they will not react to the addresses if their CRU switch has not been activated. The condition of this "card enable" switch can be discovered from the card L.E.D. If, on the other hand, you wish to use the CRU system, you may only need to use some simple gate chips which can detect the condition of address lines and whether the CRUCLK is enabled. Remember to design the circuitry to activate on an unused CRU assignment. (A3 - A&). Check p. 407 of the Editor Assembler for the various CRU Addresses or assignments. If you wish to obtain more than one output, then you need only simply connect to output of this address decoder circuit to CE of a 9901 (they can be bought at ACTIVE COMPONENTS for less than \$10.00) and then you would have 22 interface pins to play with for connecting to other things. All the connections you need can be found on the side port of the TI-99/4A. An extensive pinout can be found in the November 1987 Newsletter under the article: EXPANSION PORT INTERFACING: Part 1. In case you are still worried about this sort of thing actually working, let me tell you that I am currently building a 1536 message sign which uses 192 bytes of addresses out of the >4000 to >5FFF address block, and yes, it has been tested and does work. If you have any questions, call me at 745-4618 or leave a message on the BBS. Oh! by the way, the BBS thinks my name is DAVE CARON, not DAVID CARON.

Texlink BBS

Features:

Written in 100% TMS Assembly Language
Multiple Message Bases & File Areas
Word Wrap & Reply Options
Xmodem File Transfers
Supports MBP & Triple Tech Clocks
300/1200 Bps Operation
Hard Disk & Ramdisks Supported

Available for \$50.00 Canadian

For more information contact
The Ottawa TI-99/4A Users' Group
P.O. Box 2144 Station D
Ottawa, Ontario
K1P 5W3 Canada

Or Call the BBS (613)738-0617
(Guests must enter the part number
for the TI Extended Basic Module.)

FAST EXTENDED BASIC

LUCIE DORRIS

The purpose: a tutorial on the many options you can use with the DISPLAY AT and ACCEPT AT statements. The solution: a screen-input oriented data base. The challenge: a program small enough to fill just over a page. The result:

```
100 REM ** STAMP DBASE ** / L. Dorais / April 1988
110 REM
120 UN WARNING NEXT :: DIM S$(75)
130 CALL CLEAR :: CALL COLOR(12,16,1) :: ER$=RPT$(" ",140)
140 CALL CHAR(125,"000000FF",120,"0000000000183CFFFF3C18000000000000
1010307070301018080C0E0E0C08080") ! stamp design
150 GOTO 160 :: F,K,N,P,S,T,X,COL$,COU$,DAS$,FV$,NU$,PI$,SN$,SP$,VA$
:: CALL KEY :: !@@P-
160 A$="z~~~~{" :: DISPLAY AT(1,2):"xxxx":A$:TAB(15):"STAMP":A$:A$:
TAB(13);"DATA BASE":A$:A$: "yyyy" ! graphic & title
170 DISPLAY AT(11,1):"LOAD AN EXISTING FILE? Y" ::
CALL AS(11,25,1,A$,"YN")
180 IF ASC(A$)=78 THEN N,T=1 :: GOTO 200 ELSE F=1 ::
OPEN #1:"DSK1.STAMPDATA"
190 X=X+1 :: INPUT #1:S$(X) :: T=T+1 :: IF EOF(1)THEN CLOSE #1 ::
GOTO 200 ELSE 190 ! read a disk D/V 80 file
200 GOSUB 920 :: DISPLAY AT(2,10):"Record # " :: DISPLAY AT(4,10):
"C |":TAB(28);"|" :: DISPLAY AT(6,10):"D |" ! screen
210 DISPLAY AT(9,1):"Scott> # | | Pic | |":
"Face Value | |": " Color |":TAB(28);"|"
220 DISPLAY AT(15,1):" Used/New |U|": "Cat. Value | |":
RPT$("|" 28) :: IF F=1 THEN F=0 :: N=1 :: GOSUB 900 :: GOTO 260
230 DISPLAY AT(2,19):STR$(N) :: CALL AU(4,13,15,COU$) ::
CALL AN(6,13,7,DAS$) :: CALL AD(9,11,4,SN$) ::
CALL A(9,24,4,PI$) :: ! accept data
240 CALL AN(11,14,4,FV$) :: CALL AU(13,14,14,COL$) ::
CALL AS(15,14,1,NU$,"UN") :: CALL AN(17,14,4,VA$)
250 S$(N)=SEG$(STR$(N)&" ",1,3)&"|&COU$&DAS$&SN$&PI$&FV$&COL$
&NU$&VA$ ! make record string
260 DISPLAY AT(20,1)BEEP:" [A]dd a record [F]orward": " [D]el a
record [B]ackward": " [R]edo record [P]rint DB"
270 DISPLAY AT(23,1):" [G]et a record [S]lave DB": " [*]sort a
field [Q]uit"
280 CALL KEY(3,K,S) :: IF S=0 THEN 280
290 P=POS("ADRG*FBPSQ",CHR$(K),1) :: IF P=0 THEN 280 ELSE ON P
GOTO 300,340,310,350,390,320,330,360,360,850
300 GOSUB 870 :: IF T<75 THEN T=T+1 :: N=T :: GOTO 200 ELSE 260
! add a record
310 GOSUB 870 :: GOTO 230 ! redo record
320 IF N=T THEN 260 ELSE N=N+1 :: GOSUB 900 :: GOTO 260 ! forwrd
330 IF N=1 THEN 260 ELSE N=N-1 :: GOSUB 900 :: GOTO 260 ! bckwrd
340 A$="DEL" :: GOSUB 890 :: IF P THEN N=P :: S$(N)=SEG$(S$(N),1,4)
&"~~ del ~~" :: GOSUB 900 :: GOTO 260 ELSE 260 ! delete
350 A$="GET" :: GOSUB 890 :: IF P THEN N=P :: GOSUB 900 ::
GOTO 260 ELSE 260 ! get a record
360 IF P=8 THEN D$="PIO" :: SP$=" " ELSE D$="DSK1.STAMPDATA"
:: SP$="" ! print/save data base
370 OPEN #1:D$ :: IF P=8 THEN PRINT #1:SP$&"Rec| Country | da
te |Sc# |Pic |Face| Color |x|Val.|" :SP$&RPT$(" ",65)
380 FOR X=1 TO T :: PRINT #1:SP$&S$(X) :: NEXT X :: CLOSE #1 ::
GOTO 260 ! print a D/V 80 file
390 GOTO 260 ! **** SORT & PACK will be here ****
*** change line numbering! ***
850 GOSUB 870 :: DISPLAY AT(22,8):"QUIT (Y/N)? N" ::
CALL AS(22,21,1,A$,"YN") :: IF ASC(A$)=78 THEN 260 ELSE END
860 REM ** subs **
870 DISPLAY AT(20,1):ER$ :: RETURN ! erase
880 A$=A$&RPT$(" ",S-LEN(A$))&"|" :: RETURN ! pad with spaces & |
890 GOSUB 870 :: DISPLAY AT(21,3):A$&" RECORD # (or 0)" ::
ACCEPT AT(21,16)SIZE(2)BEEP:P :: IF P<0 OR P>T THEN 890 ELSE
RETURN ! get record no. for del/get
```

```

900 A$=S$(N) :: CALL DR(2,19,A$,1,3) :: CALL DR(4,13,A$,5,15) ::
CALL DR(6,13,A$,21,7) :: CALL DR(9,11,A$,29,4)! display record
910 CALL DR(9,24,A$,34,4) :: CALL DR(11,14,A$,39,4) ::
CALL DR(13,14,A$,44,14) :: CALL DR(15,14,A$,59,1) ::
CALL DR(17,14,A$,61,4) :: RETURN
920 DISPLAY AT(2,2)SIZE(3):STR$(T)&"~~" :: RETURN ! show total
930 !@@P+
940 SUB A(R,C,S,A$) :: ACCEPT AT(R,C)SIZE(-S):A$ :: GOSUB 880
:: SUBEND ! accept any character
950 SUB AU(R,C,S,A$) :: ACCEPT AT(R,C)VALIDATE(UALPHA)SIZE(-S):A$
:: GOSUB 880 :: SUBEND ! accept only uppercase
960 SUB AN(R,C,S,A$) :: ACCEPT AT(R,C)VALIDATE(NUMERIC)SIZE(-S):A$
:: GOSUB 880 :: SUBEND ! accept 0-9, plus ".+-E"
970 SUB AD(R,C,S,A$) :: ACCEPT AT(R,C)VALIDATE(DIGIT)SIZE(-S):A$
:: GOSUB 880 :: SUBEND ! accept digits 0-9 only
980 SUB AS(R,C,S,A$,B$) :: ACCEPT AT(R,C)VALIDATE(B$)SIZE(-S):A$
:: GOSUB 880 :: SUBEND ! accept only char. in string B$
990 SUB DR(R,C,A$,P,S) :: DISPLAY AT(R,C)SIZE(S):SEG$(A$,P,S)
:: SUBEND ! display a segment of a string

```

The drawbacks: the SORT function will be added next month. The data base can hold only a limited number of records: 75 (or 125 with memory expansion). And if you want to have different filenames, you will have to add a SUB to ask for it (lines 180 and 360). But that should be no problem, once you have studied and typed the program! And cassette users can type it too: just replace "DSK1.STAMPDATA" with "CS1" in those two lines.

I choose a stamp data base because it uses every kind of ACCEPT AT options. But since I have kept the ACCEPT and DISPLAY functions separate in user-defined subs, it is very easy to modify the program to do an address book, or any other kind of small data base that you might need.

The program is very fast because it is kept simple: the data is all in memory; as it is kept in a text file (DIS/VAR 80), it prints fast; saving it in the same format is not the best way (an INT/FIX format would be faster), but here we can use the same coding for both; as a bonus, you can use your file with TI-WRITER. And all the functions (see lines 260 and 270) are available at all times from the main screen.

The beginning of the program should by now be easy to understand; the graphics in line 140 are used to draw a stamp on the screen (line 160). We encounter our first ACCEPT sub in line 170: CALL AS; go to line 980 for the explanation: Tex will accept at row R=11, column C=25, with a size S=1, and validate only the string "YN" (it will not accept anything else); the result will be returned in the string A\$. This is the normal way to use ACCEPT with the VALIDATE(string) option; and normally we would check IF A\$="Y" or A\$="N", but here we have to use the ASC value of A\$ because, in this program, all ACCEPT subs have the statement "GOSUB 880". This sub pads each string input with spaces (depending on its length) and a delimiter "|", so that all our records are built the same way. This will become clear when we get into entering and using the data proper.

The SIZE, which tells Tex how many characters to accept, is made negative in the sub; this allows you to have a default value for the ACCEPT: the space allotted to the entry will not be erased, so whatever is on the screen will be kept or can be modified; here, our default is "Y"; in line 850 (QUIT), built the same way, the default will be "N". This program uses only negative SIZES, so that you can easily REDO your data entries.

The data for each stamp consists of eight ACCEPTable fields: Country, Date, Scott catalogue and "picture" numbers, Face Value (printed on the stamp), Color, whether you have a New or Used copy, and finally the Catalogue Value, which hopefully will go up over the years. The white "|" delimiters in the screen display are markers to show you the extent of the size of each field: yes, you can type CZECHOSLOVAKIA, but not DOMINICAN REPUBLIC.

At present, there is only one "default" value put in, for the New/Used, presuming the bulk of your collection is used stamps; if you collect only from one country, you can easily add its name to line 200: replace "C|" with "C|CANADA" for example. The variable F in line 220 is a flag to tell Tex if you have loaded a file or not.

Lines 230 and 240 call user-defined subs for almost the complete range of ACCEPT AT options; refer to lines 940-980 and their trailing remarks to understand what they do. (When you type the program, you can omit these remarks; most were added in the listing text file and might not fit your program line anyway.) Note the difference between DIGIT and NUMERIC: the catalogue numbers are only digits, while the values (face and catalogue) can be 50 (cents) or 2.00 (dollars). Better, type the program and try each one: even if you never collected stamps, you will quickly learn what kind of data is expected, because if you press the wrong key, Tex will beep and do nothing. This is the beauty of the VALIDATE option: get only what you need, and no need for IFs to check the errors! In all these statements, the parameters are passed in the same order: row, column, maximum size, return string; only CALL/SUB AS has one more, the validated string. As I told you, it is very easy to modify for your own needs, and the CALLS look much neater than a bunch of ACCEPT AT(R,C)VALIDATE(WHATEVER)SIZE(-S)!

All the (already padded) data is gathered in one string in line 250, together with the padded record number. The Menu is then displayed, with its inevitable CALL KEY. There are 10 options; that would need a lot of IF K=, with no easy checking for out of range keys, since there is no real range. So we check the position of the key value in a string (line 290), a much much more elegant and shorter way. Lines 300 to 390 are the action lines.

In the ADD line (300), change the maximum T to 125 if you have the memory expansion (and don't forget to re-DIM S\$(125) in line 120). When you DELETE a record (line 310), all you do is erase its content; the screen will show "~~del ~~" as the country name, and you can always re-use the record with REDO. As they say in the DBASE III PLUS manual, these records are only "marked for deletion". Deleting a record in the middle of a file takes a lot of time; to save it, we will actually delete these records each time a SORT will be done: this is called "packing" a file.

The same coding is used for printing or saving to disk, since both are D/V 80 files; these files are the default for Tex, so there is no need to qualify the statement with "VARIABLE,OUTPUT,etc". The SP\$ (spaces) will make a nice margin on paper only. And only the printed version will have a proper header (line 370; watch for the spaces and "|s"). Since both GET a record and DELETE it need a record number to start with ("0" is the escape), we use the same sub to get it. In the sub itself (line 890), we encounter a new ACCEPT AT: when you accept a numeric value, you cannot validate anything, of course, so we cannot use any of our CALLS here, only specify the row and column. Note the SIZE, though: it will take only 2 digits, without erasing the remainder of the screen line (change the size to 3 if you have the memory expansion).

The sub in lines 900-910 displays a record on the screen; again, I used a user-defined sub; for ease of typing, we read S\$(N), our complete record, into A\$; we pass the following parameters: row, column, string, position of segment to read from it, and number of characters to read. The SUB DR is in line 990. Now, you understand the importance of padding each field entry with trailing spaces! All records being built alike, it is an easy task for Tex to reconstruct each field for screen display (the "|s" in the records are not really needed, but when you print the data base, you get instant lines to separate each column). We could have used as many arrays as there are fields, but... what a job to write the program, let alone to type it and modify it!

Two final notes on DISPLAY AT: in line 920, the SIZE of 3 will not disrupt the rest of the line; depending on the length of STR\$(T), we will get "2~~" or "34~~" or "123"; the "~~s" are part of the stamp design. And, Henri, DISPLAY AT(R,C):"" (not used in this program) will simply erase the rest of a line by displaying nothing.

TI BASIC continued from April
by Steven Shaw

So far we are amending the variables ROW and COL without checking to see if they are valid. To use HCHAR etc they must be from 1 to 24 or 32 respectively. Anything else will produce an error message and halt the program.

```
It is possible to use lots of lines of coding in TI Basic:
200 IF ROW<1 THEN 210 ELSE 220
210 ROW=1
220 IF ROW>24 THEN 230 ELSE 240
230 ROW=24.....
      and so on.
```

It is easier however to add to the ROW incremental line a value check which will reverse the increment if it places the value outside the limits.

To do this we need to use the relational expressions discussed under "IF THEN" in the previous section.

If a relational expression is TRUE it has a value of -1
If a relational expression is FALSE it has a value of 0

Thus PRINT (2=3) will appear as 0, but
PRINT (2=2) will appear as -1.

Dealing with the ROW first, if the variable ROW starts with a value of 1, and the joystick is pushed up, we must reverse the reduction of ROW.

There are two expressions which must be true : if both ROW=1 and RR=4 then after we have added 1 to ROW we must deduct it, to leave it set to 1:

```
ROW=ROW-RR/4+(ROW=1)*(RR=4)
```

Now it is impossible for ROW to become less than 1.

This has been developed further in the sample program which you will find printed separately.

A typical use of the joystick is to move a character around the screen, and this is what the sample program will do. To give greater flexibility, this program checks both joysticks, and also checks the keyboard (keys WERSDZXC).

First the screen is cleared and the row and column variables are set to initial values. Our character is placed on screen and the joysticks and keyboard are scanned.

The next line checks to see if an input has been made: if neither joystick nor the keyboard has been used, the program will go back and look at the joysticks/keyboard again. The plus sign between the relational expressions serves as an 'OR'.

If the status of one keyboard unit is NOT zero, the program continues.

Now the program is divided into two. If the keyboard has been used, the variable ST will have a non-zero value which causes the program to branch to the keyboard section. Otherwise it continues with the joystick section.

The joystick section is a slight development of what has been discussed above. We are checking for both limits to the row variable.

The keyboard section uses similar principles, but the limit checks are a little different: If the ROW variable has a value of 1, it cannot be decreased as (RW<>1) takes a value 0 (false) and no change is made.

In the sample program a character is moved around the screen, but if you wish to leave a line of characters, just delete the line which places a blank (32) in the old position.

That was quite a complex program to develop, so check it over thoroughly. The use of relational expressions can become quite complex, but they can both speed up execution time and save memory usage.

(Look for the Program in your June newsletter)

HOTLINE NUMBERS

The executive has expressed a desire to assist all members should you have some problems or questions, want to do some library swapping or borrow a book. This will be the place to look. Listed here are the members of the executive, committee heads, and others in the group willing to help in their specialized areas. Of course, if you wish to be placed on the list, just give me a call. I know there is a lot of expertise within our Group, so I hope to add to this list. Please respect normal hours unless you specifically know that someone doesn't mind a call at 3am, or use the BBS to leave a message at 738-0617, 24 hours a day, 7 days a week.

MICHAEL TAYLOR....PRESIDENT.....831-0143
JANE LAFLAMME.....VICE-PRESIDENT.....(H) 837-1719 or (W) 745-2225
PETER ARPIN.....TREASURER AND SYSOP.....523-0017
JOHN O'CONNOR.....SECRETARY.....833-2626
BILL SPONCHIA.....PAST PRESIDENT, SOFTWARE CONTEST.....523-0878
RUTH O'NEILL.....NEWSLETTER EDITOR.....234-8050
TONY HOPKINS.....ADVERTISING.....746-4463
DAVE MORRISON.....LIBRARY CHAIRMAN.....737-4889
JACK McALLISTER...CASSETTE LIBRARY.....225-6989
HENRI MONAT.....ARCHIVES.....824-0941
LUCIE DORAIS.....MEMBERSHIPS.....232-0393
BOB BOONE.....HARDWARE/SOFTWARE.....623-7841
ART GREEN.....ASSEMBLY HELP.....837-1955
DICK PICHE.....TECH.....521-8667
CLUB BBS.....SET MODEM TO 8N1.....738-0617

LAFRAMME & WRIGLEY WHOL. CAN. DISTRIBUTOR

264 Weber St. W.
Kitchner, Ontario
N2H 4A6
(519) 578-3873

25 Ottawa St.
Amptior, Ontario
K7S 1N7
(613) 623-7841

2846 Göttingen St.
Halifax, N.S.
B3K 3E1
(902) 454-0232

UNLIMITED

CANARIA DATA INC.

COMPUTER DOWNLOAD

THE ORPHANAGE

FOR MORE INFORMATION

CONTACT A DEALER

NOW IN STOCK

GENEVE 9640



FROM

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

EDMONTON 99er USER'S GROUP
P.O. BOX 11993
EDMONTON, ALBERTA
T5T 3T1