



HOOSIER USERS GROUP HOOSIER USERS GROUP HOOSIER USERS GROUP HOOSIER USERS GROUP HOOSIER USERS GROUP
 USERS GROUP HOOSIER USERS GROUP HOOSIER USERS GROUP HOOSIER USERS GROUP HOOSIER USERS GROUP
 GROUP HOOSIER USERS HOOSIER USERS GROUP HOOSIER USERS GROUP HOOSIER USERS GROUP HOOSIER USERS GROUP
THE HUGgers
HOOSIER USERS GROUP
People Helping People
 HOOSIER USERS GROUP HOOSIER USERS GROUP HOOSIER USERS GROUP HOOSIER USERS GROUP HOOSIER USERS GROUP
 USERS GROUP HOOSIER USERS GROUP HOOSIER USERS GROUP HOOSIER USERS GROUP HOOSIER USERS GROUP
 GROUP HOOSIER USERS HOOSIER USERS GROUP HOOSIER USERS GROUP HOOSIER USERS GROUP HOOSIER USERS GROUP

NOVEMBER, 1985

THE HUGgers NEWSLETTER

VOLUME 3, NUMBER 8

THE OFFICER'S CORNER

The TI Computer Faire will be held this Saturday. The guest speakers will be Craig Miller of Millers Graphics; Lou Phillips of Myarc; and John Clulow who designed the Super Cartridge. (Editor Assembler with memory). There will be raffles and contests and all TI'ers are invited. I plan to go and will give a report at our November Meeting.

The response of the Myarc 128K card raffle has been dissappointing. To date, only 44 tickets have been sold. This is far short from the goal of \$250 in tickets, and the interest shown when the idea was first brought up at the August meeting. A decision will be made by November 30 to continue until the minimum is met, or cancell it and refund the money collected so far. Ultimately the decision is in the members hands because it is they're participation which will determine the success or failure of this venture.

Although the response of the Myarc 128K card raffle has been underwhelming; the response for a mass order of 4464's to upgrade Myarc's Ram Disk has been overwhelming. We need about 25 more memory chips ordered to meet our goal of 100. This offer is open to the general public, as well as our group members. For more information, see page 2. If you have any questions or would like to place an order, contact Bill Lucid through our Post Office Box. (Or if you want a quick reply to a question, leave a message for Bill on the HUGbbs.

This closes another Officer's Corner. See you at the November 10th meeting! J.Steven Sims.

THE NEXT MONTHLY MEETING WILL BE
NOVEMBER 10, 1985
 STARTING AT 2:00 PM AT CREATIVE LOGIC.
 SEE YOU THERE!

WORKSHOPS: BITMAC (see article, page 2.)
 NEWS FROM THE COMPUTER FAIRE

South Regional Meeting
 November 13th at 7:30 pm.
 See Page 2 for details.

DECEMBER NEWSLETTER DEADLINE
NOVEMBER 21, 1985

MYARC RAFFLE RULES

As you can see, we're going to have a raffle. The raffle is for a Myarc 128K Memory Expansion Card. Rules are as follows:

1. You must be an active member of H.U.G. whose dues are current to participate.
2. Tickets are \$1 each or 6 for \$5. There is no limit on the amount of tickets purchased.
3. Tickets may be purchased by mail. Please enclose a self addressed, stamped envelope with your order.

The drawing will be held at the December, 1985 Monthly Meeting if sufficient tickets are sold. \$250 in tickets must be collected before drawing will be held. If the goal of \$250 is not met, the drawing date will be extended or money refunded, depending on how many tickets are sold by November 30. The ticket order form is to the right. You may copy it if you don't want to cut up your Newsletter. There will also blank order forms at the meetings. (The Officers of H.U.G. and their families are not eligible to purchase tickets.)

| | |
|---|--|
| \$1 EACH RAFFLE TICKET ORDER FORM 6 FOR \$5 | |
| NAME _____ | |
| ADDRESS _____ | |
| CITY _____ STATE _____ ZIP _____ | |
| TICKETS ORDERED _____ AMOUNT ENCLOSED\$ _____ | |

BITMAC
by Vic Nelson

[This article isn't really a review, but just a description of the BITMAC program. I'm not a reviewer, just a user.]

I've been looking for a good drawing program for my TI for quite a while (ever since my first 5 minutes on a Macintosh, really). I think drawing is a REAL GOOD application for my home machine, so when I saw the BITMAC ad I sent off a check. I haven't bought the Navarone Paint program, even though it's pretty good, because I have a Gemini printer, and I'd have to pay an extra \$20 to use it. Ugh. BITMAC handles my Gemini just fine.

BITMAC is delivered on a copy-protected disk, with a 29-page looseleaf manual. The authors have obviously spent quite a bit of time on the manual - it is pretty readable, and describes the commands well. I found myself a bit lost at times when I first started, but practice helped a lot.

One of the biggest problems I had with BITMAC wasn't BITMAC's fault at all - those lousy TI joysticks. If you want to use the program at all, get some good, responsive joysticks. It is especially important that the "fire" button is quick, because the "click" function in the program depends on the speed of your use of the button.

BITMAC has a good set of drawing tools, a "pencil"-type mode, a line drawing mode, rectangles, circles, arcs, and a neat "game of life" enhancement feature (more about this later. I did, however, miss the "brush shape" feature, but the BITMAC "brush size" will do okay.

BITMAC, as I said, has a nifty enhancement feature, based on the "game of life" simulation (I think). It enhances the image on the screen by analyzing the image. I don't know EXACTLY how it works, but it's a pretty neat feature.

One of the real deficiencies of BITMAC is the fill command. BITMAC's ability to fill in irregularly shaped areas (particularly triangles) isn't very good. Strangely enough, it does okay on circles... Don't ask me why.

Colors are rather easy to use, but due to the way that hi-res colors work on the TI, actually drawing color pictures is kind of a pain. I pretty much stick to monochrome, because I like to make hardcopy, and my printer is black-and-white.

BITMAC has some facilities for uploading pictures from other machines, and, even though I haven't tried it out yet, I think this might be useful. Particularly, it makes me believe that a Prowriter or other printer dump program would be pretty easy to write.

In general, BITMAC is a reasonably useful drawing program for the TI. It is pretty easy to use, and has some neat features that I've never seen before. With a good (preferably analog) joystick, variable brush shapes, a good fill routine, (and while I'm wishing, a faster computer), it would be better.

HAPPY BIRTHDAY!

A Happy HUGger Birthday to these members who joined the Hoosier Users Group in October, 1985! Michael Lentz, Don Zimmerman, Darrell McConnell, George Forest, Robert Stahlhut, Steve Seiler, George Gordon, Bill Lucid, Johnny Powell, Bill Jones of Indy, Barb Uhrig, Carolyn Weintraut, and Steve and Pam Sims. It's time to renew.

WELCOME!

The Hoosier Users Group would like to welcome Jeffrey York, who joined the group in the past month.

WELCOME BACK!

We would also like to welcome back these renewing HUGgers: Steve Wilkins, Jack Peters, Howard Miller, Phil Dubbs, Kirk Mangold, Bill Overton, Paul Hubbard, Phil Caldwell, Ed Ferguson, Jim Acord, Roy Delong and Earl Brandhoefer.

SOUTH REGIONAL NEWS

Date: WEDNESDAY, November 13, 1985

Time: 7:30 p.m.

Location: 4582 Moccasin Place, Greenwood

Directions/Information: 881-5918

All members are welcome to attend. An updated version of the group's library is available for your use. Meeting topics are generally determined by the interests of those present. Usually members will bring programs they have written, typed, or acquired (non-copyright), and share them with the group.

4464 GROUP ORDER

Anyone interested in the USER GROUP making a group purchase of 4464 chips? The 4464 chip is almost a "new" product and a lot of distributors do not have stock as of yet. If a bulk order could be made, the savings to YOU could be significant! My source for the chips has some on-shelve now and if they should have to order from factory are looking at about 6 weeks. These chips are used to expand MYARC 32/128 memory expansion, I understand chips can be added in banks of 4, with a total 16 chips used to expand the card to 512 K.

Concerning 4464 chip pricing... based on quote 9/24/85 of \$5.10 per chip when purchased in quantity of 100 to 999 pieces. (Price break 1-24pcs \$8.60; 24-99 pcs \$7.70; 100-999pcs \$5.10 (each chip.)) Orders would need to be prepaid. Price may change as reflected by reps cost. A 100 pc or more order would come from the factory and take about 6 weeks or more for delivery. I would like to see the USER GROUP get a donation of \$5.00 per 4 chips purchased, money going to the HOOSIER USER GROUP Treasury.

Cost of 4pcs. 4464 \$5.10 would be \$20.40 plus tax, shipping and \$5.00 donation. Cost of 16pcs. 4464 \$5.10 would be \$81.60, plus tax, shipping and \$20.00 donation.

I would like to get membership input concerning bulk purchase of 4464's through the group's Post Office Box, the HUGbbs or at this upcoming meeting.

CHICAGO-AREA TI-99/4A USER'S GROUP

**P.O. Box 578341
Chicago, Illinois
60657**

August 6, 1985

Dear TI-Friends

The Chicago-Area TI99/4A User's Group cordially invites your group members, and any other interested persons, to attend their third annual TI-Computer Faire. The Faire committee currently has plans to have guest speakers, game contests, seminars on TI-computer related topics, drawings, and door prizes. The most important feature of our Faire, as always, are the vendors. The vendors that display at our Faire have the latest hardware and software available for the TI99/4A, as well as the old favorites. Bargain hunters will enjoy the wide selection of competitively priced equipment and programs that can be used to increase their computing capabilities on the 99/4A. Last year the attendance at our Faire exceeded 1500 visitors, some from as far away as Canada.

Come and visit our Faire. The Faire will be held in the Ironwood Room at Triton College 2000 North Fifth avenue in River Grove, Illinois on November 2, 1985 between the hours of 10:00 A.M. and 5:00 P.M. Just 1/4 mile north of the Maywood Park race track. Bring the family for a day of fun and mind expanding information on your favorite computer. Admission to the Faire to bona fide members of TI User's groups is \$1.00 with proof of membership. General admission to the Faire is \$2.00 per person at the door. There will be a table set up for visiting users groups to promote their own organization and recruit members, a condition of this promotion is that no hardware is to be sold by the groups. Take care you don't miss the Faire. That date again is Saturday November 2nd, 10 to 5, see you there.

Thank You,

Sandra Bartels

Group Secretary

P.S. If you need more information about our Faire please write to the group P.O. box, or if you prefer you may call our 24 hour BBS for the latest Faire updates at (312) 966-2342.

SB/jb

Programming SUPER CARTridge gives users new opportunities

[Editor's Note: For information on creating the SUPER CARTridge, see John Clulow's article in our June issue.]

By DAVID R. ROMER

You can alter assembly language software so that it may be run out of your RAM chip enhanced Editor/Assembler cartridge (SUPER CART) as a powerup menu selection.

Using information in this article, in essence you will be able to create your own cartridge, containing your own programs, runnable at the press of a single key.

The basic steps to convert software

for SUPER CART use are:

1. Move DEFinition statements to the beginning of the program.
2. Absolute ORGin the program at >6000.
3. Substitute stand-alone routines for external REFerences to cartridge utilities.
4. Add the CARTRIDGE HEADER to the source code so the powerup routine will place your program on the menu screen.

To take a closer look at these steps let's examine the following example program called TEST. If you enter, assemble and load program TEST into

your SUPER CART, selection 3 on the menu screen should say "MENU TEST". When 3 is pressed, the screen will clear and the text "MENU TEST PROGRAM" will be displayed briefly, followed by a return to the color bar screen.

Line 1 is the usual DEF statement setting the entry point or points to the program. DEF statements do not have to be at the beginning of a program as long as they precede the first reference to the DEF label. However, since the first reference to the entry point in our SUPER CART software will be in the CARTRIDGE HEADER, line 13 in TEST, it will be necessary to place the DEF statement first in the program.

Any external REFerences to utility routines must be removed from the program. Since there is no CALL IN-IT sequence prior to executing the cartridge program, utilities such as VMBW, KSCAN, DSRLNK, etc., are not available to your SUPER CART program. In a third article I will describe a routine that will make the cartridge utilities available to your SUPER CART program. Likewise, a program originally written to run from Mini-Memory will reference those utilities through EQU statements. Those EQUates must be removed from the source code. Since we are not in the Mini-Memory environment these routines do not exist at the EQU addresses. If the utilities are required in your SUPER CART program then stand-alone routines must be added to the source code. In our example program lines 30 to 39 are a stand-alone Multiple Byte Write routine used to display the text line 16 on the screen.

The 8K RAM chip you added to your E/A cartridge uses memory addresses >6000 to >7FFF. While a cartridge program could be located

```

*****
* SUPER CART TEST PROGRAM *
*****
1      DEF TEST
2      ORG >6000
3      DATA >AA01
4      DATA 0
5      DATA 0
6      DATA MNULNK
7      DATA 0
8      DATA 0
9      MNULNK DATA 0
10     DATA ENTRY pointer to
11     BYTE 07H
12     TEXT 'MENU TEST'
13     ENTRY 3 @TEST
*****
14     WS BSS 32
15     MASK DATA 0000
16     MSG TEXT 'MENU TEST PROGRAM'
17     TEST LWPI WS
18     LI R0,263
19     LI R1,MSG
20     LI R2,17
21     BL @VMBW
*****
22     LI R1,>0A
23     LOOP LI R0,>FFFF
24     LOOP1 DEC R0
25     JNE LOOP1
26     DEC R1
27     JNE LOOP
*****
28     LWPI >B7E0
29     BLWP @>0000
*****
30     VMBW SUC @MASK,R0
31     SWPB R0
32     MOVB R0,@>BC02
33     SWPB R0
34     MOVB R0,@>BC02
35     NOP
36     MLP MOVB @R1+,@>BC00
37     DEC R2
38     JNE MLP
39     B @R11
      END

```

define label TEST as program name
absolute code starting at >6000
start of CARTRIDGE HEADER,value >AA01
must be at address >6000

pointer to first menu table entry

1st menu table entry, DATA 0 = single entry
start of program
length of text to be displayed on menu scrn
actual menu text to be displayed
branch to start of program

screen location for display
pointer to text to be displayed
length of text
go display it

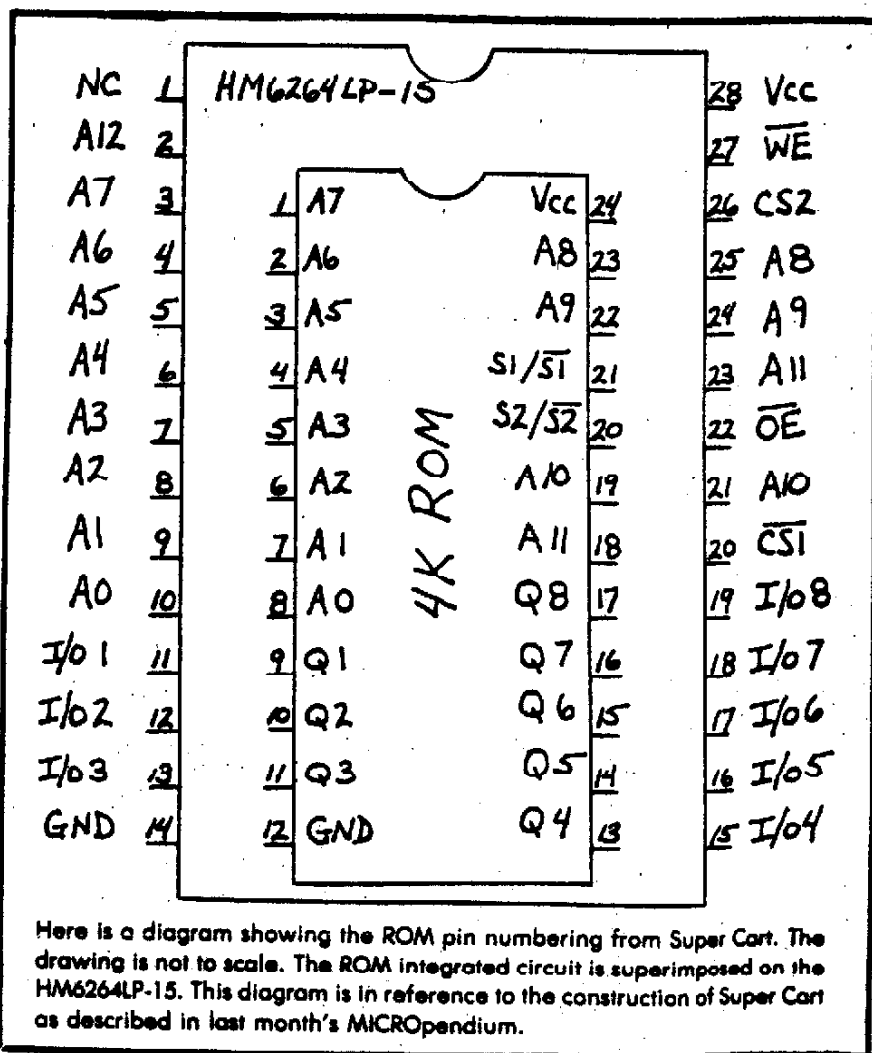
delay loop while text is on screen

load GPL workspace
branch to color bar screen

stand alone VOP multiple byte write routine

anywhere within those addresses, the CARTRIDGE HEADER must start at >6000. Since we are adding the HEADER to existing source code, an AORG >6000 statement, as in line 2 in TEST, will cause the LOADER program to place your program at that address so that it may be executed from the menu screen. All other AORG statements should be removed from the source code. Do not assume that DEF, REF, EQU and AORG statements occur only at the beginning of source code. You can save yourself some assembly time if you use the FIND function in the EDITOR to search all of the source code for occurrences of those statements.

Lines 3 to 13, in program TEST, make up the CARTRIDGE HEADER code that enables the powerup routine to include and execute your program from the menu screen. Lines 3 to 13 MUST load starting at >6000. The only source code statements that can precede the CARTRIDGE HEADER are DEF, AORG or EQU statements. In line 3 the value XAA will load at 6000 and tells the powerup routine that a valid cartridge is in the cartridge slot. The value >0! indicates that the cartridge contains executable machine code. Lines 4 and 5 set the next two memory words to 0. Line 6 can be any valid label and is a pointer to the first



SUPER CART PARTS LIST

| Parts | Qty | Source | Price |
|---|-----|----------------------|--------|
| HM6264LP-15 8K CMOS RAM | (1) | Microproc. Unlimited | \$8.85 |
| TI Game Module. Must have metal contacts on BOTH SIDES of connector; e.g. Munchman, TI Invaders | | | |
| 1N914 Signal Diode | (2) | Rad Shk 276-1122 | .20 |
| 1K Resistor—1/4 Watt | (3) | Rad Shk 271-1321 | .24 |
| 2.2 uF Tantalum Capacitor | (1) | Rad Shk 272-1435 | .59 |
| Red LED | (1) | Rad Shk 276-041 | .35 |
| Wire-wrap Wire | (1) | Rad Shk 276-503 | 2.39 |
| Lithium Cell | (1) | Rad Shk 23-160 | 1.79 |
| Coin-type Cell Holder | (1) | Digi-Key BH906-ND | 1.25 |

Tools

| Tools | Source |
|-------------------------|-----------------|
| 25 Watt Pencil Iron | Rad Shk 64-2070 |
| Vacuum Desoldering tool | Rad Shk 64-2070 |
| Resin Core Solder | Rad Shk 64-005 |
| Electrical tape | |

Digi-Key
P.O. Box 677
Thief River Falls, MN 56701
(800) 344-4539

Microprocessors Unlimited
24000 S. Peoria Ave.
Boggs, OK 74421
(918) 267-4961

entry in the menu table. Line 9 is the menu table entry pointed to by line 6. In program TEST it is set to 0 as there is only one entry point into TEST. If there were additional entry points or multiple programs, line 9 would point to the next menu item (e.g. MNULNK DATA NXTLNK). The menu screen would appear to have room for seven or eight selections. Line 10 is a pointer to the start of the program. Line 11 is a byte value that is the length of the text to be displayed on the menu screen. Line 12 is the text to be displayed. Finally, line 13 is the Branch to the start of the program.

Once you have made these changes to your source code, assemble it, then load it into your SUPER CART with Opt. 3 LOAD and RUN on the Editor/Assembler menu.

You may also load your SUPER CART from TI-BASIC with CALL INIT then CALL LOAD("DSK1.YOURPROG"). Press Function Quit to return to the color bar screen, press any key and your program should appear as selection 3 on the menu. For those of you who maybe using the Cor-Comp disk controller

(Insert Strips - Continued)

As many of you know who have used these programs, the reference strip is not a substitute for reading the manual or the instructions. It will keep you out of trouble a little longer but is most effectively used as a reminder after you have read the documentation.

You will find that the strips, if trimmed properly will slide in the slot of the earlier black and silver machines although if you are using a widget that has ceased to be practical. At any rate they will drop into the 'indentation' of any model easily.

*** ASSEMBLY INSTRUCTIONS ***

The strips may be cut out and used as they are though they will probably not last long. To make them permanent mount the sheet on a dense board such as a piece of black accopress such as that used in report binders. If not available 4-5 ply poster board will work. Use a glue which does not have water as the solvent such as rubber cement. Water relaxes the paper fibers and when the paste dries, the board will warp. While the adhesive is still wet, rub out the air bubbles and excess adhesive with a smooth hard object, working from the center outward. Let dry.

Next, color the circles according to the key shown on the diagram either red for CONTROL or grey for FUNCTION. There is one odd ball in the music box for SHIFT which I colored yellow but you may pick any color to remind you that it is 'different'.

seal the surface hold a piece of Scotch tape longer than the strip tightly and apply from one end to the other avoiding wrinkles or air bubbles. Rub firmly with a smooth object to remove all of the air.

The strips may be cut out with sharp scissors but I don't recommend it. It is better to use a paper cutter. If you want the strips to fit into the slot cut at the black and white boundary. If you cut a 1/16th inch border all around, they may be more attractive and are easier to handle but will not slide into the slot. For the purist - use a black marker to darken the edges. For the ultra-purist and for a permanence probably longer than the program they apply to - 3 coats of spray lacquer on the back and edges. I hope that you find them as useful as I have.

There are always additional programs coming along which would lend themselves to similar command strips and I will be looking for additional ideas to publish. If you have an idea for a strip layout which can be used, submit it and when we have 6-8 I will publish them. Any funds received from sales to non-members will be used for acquiring additional materials for the library. Editor

SUPER CART

and SUPER CART with a program having multiple entry points, you will find that the Cor-Comp screen will display only the first entry in the menu table. However, pressing the space bar twice will display the TI screen that will show all of the menu selections you have added to your SUPER CART. The best part of it is that your program will still be there ready to go until either the battery quits or you change the program.

Those are the basic steps required to alter software to run out of your SUPER CART. Some programs may require only the addition of the CARTRIDGE HEADER and others may

require extensive revision to substitute stand-alone utility routines. The possibilities are limited only to one's imagination and 8K of memory. By the way, this article was written with a SUPER CART containing the TK-Writer loader for TI-Writer. The menu screen has five selections:

1. TI-BASIC
2. Editor/Assembler
3. TK-Editor
4. TK-Format
5. TK-Utility

TK-Writer is a very easy conversion. You only have to add AORG >6000 and the CARTRIDGE HEADER.

A DISKO FIX

DISKO is a program that runs on the Editor/Assembler which will allow you to look at (as HEX or ASCII) and change any sector of a diskette. The menu that comes with the program is humorous. Only the first two choices are functional. I have it on good authority that the original DISKO that was submitted to TI was written by Guy Stefan Romano. The program as it first came from TI would allow you to address any sector on a double sided diskette (or single sided double density). However when stepping through the sectors in a forward direction there was a barrier in the program that would not let you go past 360 (the limit of a SS/DD diskette). Despite this "barrier" it is still possible to address specifically a sector up to 720 or step backwards if the starting sector was higher than 360.

Earl Hall posted a "fix" on Compuserve to allow this barrier in the program to be removed. Dick Vandenburg wrote an article explaining how to implement the fix. Bob Willis implemented the fix for the club's copy of DISKO.

This article is largely copied from Dick Vandenburg.

Load the code file into the ED/ASSEMBLER Editor and go to record #97. The numbers following the 3rd "B" tag should be >0167. Change these to >02cf (720 base 10) or >059F (1440 for you DS/DD folks). Then change the "7" tag (the 6th from the last non-blank character on that line) to an "B" so that the checksum won't be checked. Select the SAVE option of the Editor, respond N to the "Variable-Length?" prompt and save the file (as a different [file] than the original!!!). You should be able to load that file and access, through the PROCEED Fctn. key, the entire disk.

(This article is extracted from the Atlanta U/G, A9CUG CALL NEWSLETTER, Jun 85.)



You mean that after unpacking the styrofoam padding, the cushion-layering, the thermodynamic seal, and the plastic wrapper, that's what was left?"

SIMPLE GROUND RULES FOR LOADING NEW PROGRAMS ON THE TI99/4A COMPUTER

by Bruce Carson

Editor's Note: The following article is printed in the HUGger Newsletter via the October, 1985 issue of "The ROM Newsletter," Newsletter of the Users Group of Orange County, Fountain Valley, California.

Catalog the Disk

II. Look at titles, and if name is:
UTIL1 - Use Editor Assembler Option 5 or TI-Writer Option 3. Ignore file name request and just press enter. The program will autoload (from drive 1 only) and this is the fastest loader available on TI99/4A.)

A QUICK FIX FOR S-BUG BUG

by Tom Knight

Jacksonville, Florida

Editor's Note: The following article is printed in the HUGger Newsletter via the August, 1985 issue of Penn Ohio Users Group Newsletter, Youngstown, Ohio.

When TI finally released Super Bugger it had a "bug" in that it is supposed to be able to disassemble or dump memory to a disk and will not properly do this. (In my opinion this was a TI-induced bug.)

I have been working on this problem and have found a solution that, so far, seems to work fine.

With no other program in memory, "S-Bug" loads from >A000 to >B96A and I will be referencing memory with this assumption.

| Memory Location | Contains | Change To |
|-----------------|----------|-----------|
| A15A | 3F20 | 101F |
| B2DE | 7F00 | 0FFF |
| B2F2 | 3F09 | 1009 |
| B32A | 7F20 | 101F |
| B342 | 7F05 | 1005 |
| B356 | 7F00 | 0FFF |
| B366 | 3F09 | 1009 |
| B37A | 7F00 | 0FFF |
| B382 | 3F09 | 1009 |

These locations are all references to either the PAB or the data buffer which is used by DSRLINK which, by the way, is included in Super Bugger as are the other utilities used by the program. It is completely stand-alone. All of the utilities are very similar to the ones that come with the Editor/Assembler Cartridge.

There are three ways to make these changes:

1. Each time you load the program you can make the changes while the program is running.

2. The regular version (uncompressed) can be changed using the "Editor" or with TI-Writer. Be sure that on each line that you change you also change the "checksum" flag to an 8 (it is normally 3 7.)

3. To change the compressed version you need Disk Fixer or something similar. You actually change the disk information. If you are familiar with the use of Disk Fixer you should have no problem, otherwise it could get very hairy.

III. Look at program types:

DIS/FIX - Editor Assembler #3, DSK1.NAME, enter.

DIS/VAR - TI-Writer.

PROGRAM - Extended Basic (or) Basic (or) Editor Assembler #5 (or) TI-Writer #3 (try them in that order.)

INT/FIX - This is usually a file loaded from an Extended Basic program.

FORTH CORRECTIONS

PART I

Editor's Note: The following article is printed, in part, in the HUGger Newsletter via the August, 1985 issue of MicroPendium. These corrections were so extensive and due to space limitations, one or two corrections will appear each month.

By TOM FREEMAN

CORRECTIONS TO THE FORTH SYSTEM DISK

I have found the following errors in the system disk as I received it (even the version with screens 58-59 dated OCT83).

The first is simple. Line 1 in screens 53, 54, and 55 contains the word VDPSET2. This should be SETVDP2.

Second, lines 9 and 10 in screen 58 should be switched, and the new line 9 should read:

```
VDPMD @ 4 < IF SMTN 80 0 VFILL 300 ' SATR ! ENDIF
```

If (INIT ALL SPRITES) is on this line, it may be deleted as it is not compiled anyway. Note the ' before SATR not !, which was in the "corrected" version labelled 20OCT83 LAO on line 0. You may add 1NOV84 TSF to this if you wish.

Third, in line 9, screen 59, between >R and SP@ should read:

```
8 SLA SWAP 00FE AND OR
```

Line 0 on this screen should read 20OCT83 LCT).

CORRECTIONS TO DOUBLE FORTH

Thanks to Jim Vincent for publishing the Double-Sided Forth information, and to others for reprinting it.

Unfortunately the disk with screens only on it does not in fact copy with the TI Disk Manager. The problem is in bytes 15-17 (the sector access chain) and possibly in byte 10, which contains the total number of records, in this case twice the number of sectors.

Thus in screen 40 make the following changes/additions to Jim's note:

line 11 change CA02 to 9A05 for 180 SCR, 3A0B for 360 SCR

line 12 change 2250 to 22D0 for 180 SCR and 360 SCR

line 13 change 1403 to 2A03 for 180 SCR, 5703 for 360 SCR (retain previous change)

For DD-Forth make the following addition between DUP E etc. and DUP IC etc:

```
DUP 12 + j SWAP ! (j=4005 for 180 SCR, E00A for 360 SCR)
```

And the following changes:

```
DUP 1E etc. (h=5505 for 360 SCR)
```

```
20 + etc. (i=F056 for 360 SCR)
```

I do not own a double-density card, so I could not test these changes, but I am reasonably sure they work. I also decided to retain the original word DISK-HEAD and made a new one as follows:

```
: 2DISK-HEAD DISK-HEAD 0 BLOCK DUP A + 2D0 SWAP ! DUP 10 + 2028  
SWAP ! DUP 12 + 201 SWAP ! 200 + DUP E + 2CD SWAP ! DUP 12 +  
9A05 SWAP ! DUP 1C + 22D0 SWAP ! DUP 1E + 2A03 SWAP ! 20 + C02C  
SWAP ! UPDATE FLUSH :
```

These values are for DSSD drives. You can substitute the appropriate values for SDDD or DSDD.

For what it's worth, NONE of these is really necessary if you use Forth itself to copy the disks (or any mass copier) since it doesn't make use of the disk map. But it's nice to be complete! And you might give a disk to a friend who doesn't have Forth set up yet.

TI LOGO VARIABLES

by George Paschetto

The following article comes to the HUGger Newsletter from the "Computer Bridge," Newsletter of the St. Louis Users Group via the LA 99'ers TopIcs, Gardena, California.

To make your BOX procedure run a different size box each time, you can use a variable for the side. Type TO BOX and press enter ONCE. In the EDIT mode, the cursor appears after the procedure name at the screen's top for adding variables, so type the variable's name (S for SIDE) here. Now to use the variable, type :S (the colon is required when the variable is used instead of a number) where the 50 was. It should look like this:

```
TO BOX S
REPEAT 4 [FD :S RT 90]   FD :S will be the box's side.
END                       fctn 9 <back>
```

Now type BOX # (some number) and you'll get a box the size specified. If you type BOX and no number, or try to use the STAR procedure, the computer will come back with TELL ME MORE. BOX needs a number now in order to run, and any procedure that uses BOX now has to provide one. To use STAR now, you'll have to add.....

```
.....a variable
TO STAR S
REPEAT 6(BOX :S RT 60)
END
```

```
.....or a number.
TO STAR
REPEAT 6(BOX 50 RT 60)
END
```

fctn 9 <back> gets you back to the turtle.

Add this to change the number of boxes in the star:

```
TO STAR S TURNS
REPEAT :TURNS[BOX :S RT 360/:TURNS]
END
```

In turtle mode, type STAR # # 'enter'; the first number will be the box size, the second number will be how many boxes there are in the star. To see your procedures type 'NOTURTLE' enter, then PA (print all). when done type "TELL TURTLE" again to return to the turtle.

If you have a particular interest or question concerning LOGO please write to George Paschetto, Apt. #5 255 University, Radcliff, KY 40160.

TK-WRITER REVISIONS

From the Jackson County 99'ers, come these tips on modifying Tom Knight's Extended Basic load program for TI-Writer.

Apparently, when going from the Editor to the Formatter, the LOAD program reloads the assembly program, not checking to see if it is still in memory. The resulting wait can be avoided by making the following modifications to the LOAD program:

```
100 CALL CLEAR :: CALL INIT :: CALL PEEK(-2043,A,B) ::
    IF A<>84 OR B<>75 THEN 108
102 CALL LOAD(16360,85,84,73,76,73,84,250,212,70,79,82,
    77,65,84,250,132,69,68,73,84,79,82,250,22)
104 CALL LOAD(8196,63,232) :: GOTO 110
108 CALL LOAD("DSK1.WRITER")
```

The second tip deals with the potential problem of selecting "SD" for Show Directory and losing your current work. If you are aware of this problem you can probably avoid it, however, there is a fix. Find the third sector of the EDITA1 program (using a utility program such as DISKO), and make the following modification:

ORIGINAL FORM

```
2D 54 53 48 3E OF 2D 54 52 45 3E 2C 0C
2F CA 4D 20 3E B4 2F 42 53 44 3C D4 18
2E 4C 53 46 00 00 2E 8A 4C 46 00 00
```

ALTERED FORM

```
2D 54 53 48 3E OF 2D 54 52 45 3E 2C 0C
2F CA 4D 20 3E B4 2F 42 20 44 3C D4 18
2E 4C 53 46 00 00 2E 8A 4C 46 00 00
```

This change in the program will return you to the command line should you happen to press "SD".

TIPS FROM THE TIGERCUB

925

Copyright 1985

TIGERCUB SOFTWARE
156 Collingwood Ave.
Columbus, OH 43213

Distributed by Tigercub Software to TI-99/4A Users Groups for promotional purposes and in exchange for their newsletters. May be reprinted by non-profit users groups, with credit to Tigercub Software.

The entire contents of Tips from the Tigercub Nos. 1 through 14, with more added, are now available as a full disk of 50 programs, routines and files for just \$15.00 postpaid!

Nuts & Bolts is a diskfull of 100 (that's right, 100!) XBasic utility subprograms in MERGE format, ready for you to merge into your own programs. Contents include 13 type fonts, 14 text display routines, 12 sorts and shuffles, 9 data saving and reading routines, 9 wipes, 8 pauses, 6 music, 2 protection, etc., and now also a tutorial on using subprograms, all for just \$19.95 postpaid!

And I have about 140 other absolutely original programs in Basic and XBasic at only \$3.00 each! (plus \$1.50 per order for cassette, packing and postage, or \$3.00 for diskette, PPM) I will send you my descriptive catalog for a dollar, which you can then deduct from your first order.

Many of the users groups are taking a summer break, so I thought I would do the same. I'm going to mail out the July and August issues of the Tips in June (imagine, a TI publication

AHEAD of schedule!!) and then go fishing. However, if anyone should by any chance decide to send me an order during the summer, they will still get my same-day service.

It seems that I had better clear up a few misunderstandings. The "freeware" offers I have mentioned in past Tips are NOT available from me - send your disk and returnable mailer AND RETURN POSTAGE to the author of the program.

And, my copyrighted Tigercub Software programs are NOT freeware. They can only be legally obtained by mail order from me - if you copy them from anyone else, you are stealing!

As for the programs which I write and publish or distribute without copyright, they are also not Freeware, they are FREE. I don't want to be paid for them, and I don't think anyone else should be paid for them.

Some users groups are putting my copyrighted programs, and those of other programmers, in their software library, "for use but not copying" or "for review and evaluation only". Who do you think you're kidding? I know I won't sell any software to members of pirate clubs, so why should I support them?

If you didn't solve the Long Division Puzzle in Tips #24, try dividing 230709 by 833. As for the solution to the Tigercub Challenge, it was right on the same page! Try creating those DATA statements with the LINewriter routine. I don't know why it works, but it does.

I've been asked to print more information on the "program that writes a

program". I don't have room for a detailed account, but here are the basics. If you tried my TOKENLIST routine in Tips #23 you already have a list of the token codes you will need.

I won't go into the way that the computer squishes a program line number into only two characters, but you can accomplish it with DEF L\$=CHR\$(INT(LN/256))&CHR\$(LN-256*INT(LN/256)), where LN has been predefined as the value of the line number.

If you need to refer to a program line in a statement, as in GOTO 500, use DEF R\$=CHR\$(Z01)&CHR\$(INT(RN/256))&CHR\$(RN-256*INT(RN/256)), RN being the line number.

To print a statement or command, simply print its token character. For instance, the token for DATA is 147, so you would print CHR\$(147). Note that all the punctuation marks used in programming, such as (and +, are also represented by token codes which are NOT the same as their keyboard ASCII value.

To print a variable name, either numeric or string, just enclose it in quotes, "A" or "A\$".

To print a value, or an unquoted string (as in a DATA statement), or the word which follows a CALL, you must print CHR\$(200) followed by a token giving the number of characters to follow, such as CHR\$(5) for a 5-character word such as CLEAR, then the value in quotes. For instance, the token for CALL is 157, so CALL CLEAR is CHR\$(157)&CHR\$(200)&CHR\$(5)&"CLEAR".

You can simplify that by predefineding DEF U\$(V\$)=CHR\$(200)&CHR\$(LEN(V\$))&V\$, and then simply print CHR\$(157)&U\$("CLEAR").

A quoted string is handled in the same way

except that it is preceded by token 199 instead of 200, so you can predefined it as DEF Q\$(V\$)=CHR\$(199)&CHR\$(LEN(V\$))&V\$ - the computer will take care of the quote marks.

Each program line must end with CHR\$(0), and the last record you print must be CHR\$(255)&CHR\$(255).

A MERGE format file is D/V 163, so open the file with OPEN #1:"DSK1.MERGEFILE", VARIABLE 163.

Don't print more than 163 characters in a record or the computer will blow its mind! You can print multiple-statement XBasic lines, but be sure to use the double-colon token CHR\$(130) as the separator, not two of the CHR\$(181) colon tokens.

Any errors you make will usually not show up until you try to MERGE or use the program you have created. I/O ERROR 25 means that you forgot the final 255 & 255; DATA ERROR or SYNTAX ERROR probably means that you left off a CHR\$(0) or gave the wrong count of characters after CHR\$(200).

Here's a bit of psychedelic blues - -

```
100 REM - FRANKIE & JOHNNIE
      by Jim Peterson
110 DIM S(12)
120 CALL SCREEN(2)
130 FOR R=1 TO 12
140 CALL COLOR(R+1,1,1)
150 FOR T=R TO 25-R
160 CALL HCHAR(T,R,32+R*8,34-2*R)
170 NEXT T
180 NEXT R
190 DATA 262,294,311,330,349
      ,392,440,494,523,587,40000
200 FOR N=1 TO 11
210 READ S(N)
220 NEXT N
230 FOR J=1 TO 110 STEP 2
240 CALL COLOR(A+1,1,1)
250 READ T,A
260 CALL COLOR(A+1,A+2,A+2)
```

```

270 FOR TT=1 TO 1
280 CALL SOUND(-999,S(A),0)
290 NEXT TT
300 NEXT J
310 RESTORE 330
320 GOTO 230
330 DATA 2,1,2,2,2,4,2,7,1,1
1,1,7,2,6,4,4,2,1,1,11,13,1
340 DATA 2,1,2,2,2,4,2,7,1,1
1,1,7,2,6,4,4,12,1
350 DATA 1,11,3,1,2,5,2,6,2,
7,2,9,1,11,1,9,2,10,4,7,1,9,
1,11,7,9
360 DATA 4,7,2,8,2,9,1,11,3,
9,1,11,1,9,4,8,2,7,6,6
370 DATA 4,4,1,11,3,4,4,3,16
,2,1,11,4,7,2,6,4,7,4,6,20,1
,8,11

```

You can too have a blank space in your disk filenames! Just use FCTN V for the blank, instead of the space bar. You can even have a diskfull of 10 programs with invisible filenames consisting of 1 to 10 of those FCTN V's.

However, those invisible characters can do strange things when you list your disk catalog to a printer.

If you want to INPUT a string with leading and/or trailing blanks, just enclose the whole works in quotation marks. Try this -

```

100 INPUT AS !type TEST
110 PRINT AS;LEN(AS)
120 INPUT AS !type " TEST "
130 PRINT AS;LEN(AS)
140 GOTO 100 !you can even
input a blank string of 136
characters

```

I really shouldn't tell you this, but if you want to make it difficult for someone to LIST your program, just insert a garbage line, every 5th line or so until you run out of memory, consisting of REM followed by 4 or 5 lines of random characters typed with the CTRL key held down.

Here's a program that

```

can actually read your mind!
100 CALL CLEAR
110 PRINT "TIGERCUB MIND REA
DER PROGRAM":
120 PRINT "I'll bet you a do
llar I can guess what you ar
e thinking.":
130 GOSUB 440
140 PRINT "And I'll bet ano
ther dollar I can tell if wh
at you are thinking is cor
rect.":
150 GOSUB 440
160 PRINT "And I'll bet anot
her dollar I'm right BOTH ti
mes.":
170 GOSUB 440
180 PRINT "And I'll bet one
more dollar I can guess what
you'll be thinking a minute
from now.":
190 GOSUB 440
200 PRINT "OK....":
210 GOSUB 480
220 PRINT "You're thinking t
hat a compu-ter can't possib
ly know what you are thin
king.....right?":
230 GOSUB 480
240 PRINT "So I told you wha
t you were":"thinking.....
right?":
250 GOSUB 480
260 PRINT "You owe me a buck
.":
270 GOSUB 480
280 PRINT "And you're absolu
tely right..I can't re
ad your mind.":
290 GOSUB 480
300 PRINT "So I told you cor
rectly that":"what you were
thinking was":"correct.....
right?":
310 GOSUB 480
320 PRINT "You owe me anothe
r buck.":
330 GOSUB 480
340 PRINT "So I was right BO
TH times...right?":
350 GOSUB 480
360 PRINT "That makes three
bucks you owe me.":
370 GOSUB 480
380 PRINT "And now it's a mi
nute later":"and you're thin
king you've":"been played fo
r a sucker....":"...right?":

```

```

390 GOSUB 480
400 PRINT "...so you owe me
four bucks.":
410 GOSUB 480
420 PRINT "NEVER NEVER bet a
gainst a computer!! "
430 END
440 PRINT "Want to bet? Type
Y(yes)":
450 CALL KEY(3,K,ST)
460 IF (ST=0)+(K>89)THEN 45
0
470 RETURN
480 FOR D=1 TO 800
490 NEXT D
500 RETURN

```

Since the manual doesn't mention it, some folks don't know that you can use IMAGE and PRINT USING for output to the printer. Try this -

```

100 OPEN #1:"PIO"
110 INPUT "NAME? ":N$
120 INPUT "AMOUNT? ":A
130 PRINT #1,USING "#####
#####"
.##":N$,A
GOTO 110

```

Of course, you could also add a line -

```

105 IMAGE "#####
#####"

```

And change line 130 to

```

130 PRINT #1,USING 105:N$,A

```

John Taylor has written the most complete and versatile SPRITE BUILDER utility program that I have ever seen. It has 22 different options available with a single key press, including rotation and animation. And along with it comes a diskfull of preprogrammed sprites designed by a professional artist. This is being distributed as Freeware. Send two single-sided or one double-sided disks to John Taylor, 2170 Estaline Drive, Florence AL 35630, in a returnable mailer WITH RETURN POSTAGE, at least - and I hope you'll also include something more!

Attention, assembly programmers! Fred Hawkins of the Lehigh UG is trying to coordinate a project of documenting the operating system by breaking the console ROM down to pages of 256 bytes so that each individual or group can work on just one page. Only those who participate will share in the results! All this is far beyond me, but if you want in, send an SASE and a SSSD disk with return postage and mailer to Fred Hawkins, 1020 N 6th St, Allentown PA 18102 - soon!

If you have a program on disk which is so long that you must type CALL FILES(1) before you can load it, add several program lines to it consisting of REM and any key you want to hold down for 5 lines. Then SAVE it back to the disk; it will now be in INT/VAR 254 format and will load without CALL FILES(1). If you then need sometime to make a cassette copy, just delete those lines and SAVE it back to disk again.

If a program loads, but gives you a MEMORY FULL IN LINE ... when you try to run it, it has used up all available memory while reading DATA into arrays or performing other internal calculations. If it runs for some time and then gives you the MEMORY FULL message, it is because you have repeatedly jumped out of a FOR...NEXT loop with an IF...THEN...GOTO before the loop is completed. This rarely happen but it can, especially when you repeatedly jump out of the innermost of several nested loops.

MEMORY FULL

Jim Peterson