UPSTATE UPSTATE UPSTATE UPSTATE UPSTATE UPSTATE UPSTATE UPSTATE UPSTATE UPSTATE

OUR NEXT MEETING will be on Thursday
MARCH 20, 1986 at 7:30 pm

THE APRIL Meeting will be
APRIL 17, 1986 at 7:30 pm

PLACE: CAPITAL DISTRICT PSYCHIATRIC CENTER

New Scotland Ave. Next to Albany Medical Center

The program for the MARCH meeting is as follows:
Bob Burgess will give a demonstration on Miniwriter III.
A presentation of sort algorithms using TI Basic.
Chuck Eacy will demo a new utility for PRBASE.
The Software Library Club will return.

A NOTE to other Users Groups: The articles printed in the Upstate Newsletter may be reprinted if proper credit is given to the author and to the Upstate New York 99/4 Users Group.

ALBANY TEXAS INSTRUMENTS BULLETIN BOARD SERVICE (TIBBS)-out of service

UPSTATE 99/4A USERS GROUP
P.O. BOX 13522
ALBANY, N.Y. 12212

VOL IV, NO. 2 MARCH 1986 Allison Smith, EDITOR 439-4860

*** Extended Basic Auto-Boot (DSK1.LDAD) Bypass Patch ***

* First LDAD Extended Basic into the Gram Kracker.
* From the Gram Kracker menu select 5 Memory Editor. Then press FCTN = for HEX, FCTN 1 for the Gram Memory Window and then press FCTN 5 for SEARCH.
* Type in >6300 for the START address and >6400 for the FINISH address. Press FCTN 9 to put the cursor in the Search String Input area and type in 86 A3 71 and then press FCTN S (left arrow) to put the cursor on the last byte to search for. Next press ENTER to start the Search.
* For most Extended Basic modules this Hex string will be found at >63CD. We'll call that "address A". Now press FCTN 5 to leave SEARCH and then press FCTN 9 to put the cursor in the Memory Window. Turn off the Write Protect (turn it to Bank 1). Now change the first two bytes (86 A3) to 58 00. This is a BRANCH ON RESET to >7800 instruction.
* Press FCTN 9 and change the Memory Window to g7800. You will see garbage here (UNLESS YOU HAVE PREVIOUSLY PUT SOMETHING IN THIS SPACE!!). The GROMs are only 6K in length so the bytes in the last 2K are "garbage wrap around" read by the Gram Kracker Save routine. So, it's a good area for adding routines to your modules.
* Press FCTN 9 to put the cursor in the Memory Window and at the g7800 memory location, put in the following code:

```
86 A3 71          CLR V>371       Clear Auto Load needed flag
03                SCAN             Scan the Keyboard
D6 75 20          CEQ >20,>0375   Is the Space Bar pressed
[Now take your "address A" and add 6 to it ]
[>63CD + 6 = >63D3                           ]
63D3              BS "address A" plus 6 bytes  YES! (Branch on Set)
[Take your "address A", add 3 to it and replace the first digit with 4]
[>63CD + 3 = 63D0 ..... change it to 43D0                             ]
43D0              BR "address A" plus 3 bytes   NO!  (Branch on Reset)
```

* For a module with a >63CD "address A" your memory window should now look like this:

```
                  g7800
                  ========================================
                  86 A3 71 03 D6 75 20 63 D3 43 D0 xx
                  xx xx xx xx xx xx xx xx xx xx xx xx
                  xx = don't care
```

* Now restore the Write Protect, return to the Gram Kracker menu and resave your module.
* Now when you select EXTENDED BASIC you can bypass the auto-load command by holding down the space bar!! (No more DSK1.LDAD search)

QUIT KEY OFF with the NOAUTO.GKT PATCH

If you have installed the NOAUTO.GKT patch in your Extended Basic here is a slight modification to it. This modification AUTOMATICALLY turns off the QUIT key when you enter Extended Basic.

1. With the modified Extended Basic (NOAUTO.GKT) Loaded into the Gram Kracker select 5 Memory Editor from the Gram Kracker Menu.

2. Press FCTN = for HEX, FCTN 1 for Gram Memory Window and change the address to g7800e this: (for most XB's - don't forget about the "address A" changes)

```
                  g7800
                  ========================================
                  86 A3 71 03 D6 75 20 63 D3 43 D0 xx
```

4.~To make this patch we will ADD the following four bytes in front of the NOAUTO patch: BE 80 C2 10. This is the OPCODE for STORE >10 **>83C2**. So with the change in place your memory window should look like this:
(Hint - retype your existing NOAUTO patch starting at the fifth byte in the Memory Window.

```
                    g7800
                    ==============================
                    BE 80 C2 10 86 A3 71 03 D6 75 20 63
                    D3 43 D0 xx xx xx xx xx xx xx xx xx
                    xx = don't care
```

Also don't forget about the "address A" changes.

This is a text file discussion of the ROM cartridge port for the TI-99/4A. It represents information I have been able to obtain from various references. Cartridge programs must operate from >6000 to >7FFF. When the computer is RESET or turned on, the power up routine looks for a Header or Control block at location >6000 in the cartridge port. This contol block establishes the linkage into your cartridge program and allows you to have multiple entry points. Here is an example contol block used to provide one entry point;

```
0000 AA01        DATA    >AA01     6000      ID FOR BOOT
0002 0000        DATA    >0000     6002
0004 0000        DATA    >0000     6004
0006 000C        DATA    CHAIN     6006      ADDRESS OF MENU LIST
0008 0000        DATA    >0000     6008
000A 0000        DATA    >0000     600A
000C 0000 CHAIN  DATA    >0000     6010      CHAIN POINTER
000E 0020        DATA    SLOAD     6012      ENTRY POINT
0010   0F        BYTE    SLOAD-$-1 6014      LENGTH OF MENU TEXT
0011   54        TEXT    'CARTRIDGE NAME'
0020 0460 SLOAD  B       START
0022 092E
```

Let's examine the control block. If the TI operating system finds >AA at >6000 it knows a cartride is plugged in the port. The next byte must be a >01 at location >6001. This informs the operating system that the code in the cartridge is executable machine language. Other codes are used for GROM, but that's another discussion. The data at location >6002 - >6005 is zero. Location >6006 must contain a word pointer to a list which identifies the menu text and associated entry point when that item is selected. This location usually contains a >600C. Locations >6008 - >600B must be zero. The chain list at >600C contains the following:

Bytes 1 2 = chain pointer to the next menu list - or 0000 is this is the last list in the chain.

Bytes 3 4 = entry point associated with this menu selection.

Byte 5 = length of the menu text.

Bytes 6 - N = Menu Text - this is displayed on main menu. Craig Miller's newsletter has additional information on the power up routine for the computer. Remember all dynamic data must be in RAM usually in the >8300 area. This area is used for registers plus VDP RAM is used for variable storage. Cartridges cannot REFerence any lable or routine outside the cartridge. This means the cartridge program must provide it's own VSBW, VSBR, VMBW, and VMBR routines which are normally loaded from the Editor Assembler cartridge. Examples of what these routines look like may be found in the Tombstone City game or Craig Millers newsletter. Armed with this information, it possible to disassemble code to see how the program works. Hope you find this information useful.

PRESIDENTS'S COMPUTER CORNER

During last month's meeting , there was a presentation of FORM1040 and SCHEDULE-A templates with MULTIPLAN . If you missed receiving a copy of these templates, bring a diskette to the next meeting to receive a copy. Listed below are the instructions for using the templates:

(1) Use the TRANSFER LOAD command to bring FORM1040 .
(2) Fill out the line numbers thru line 32 (L32) using GOTO NAME command. (ie, L6a, 17, L32, etc.).
(3) Use the TRANSFER SAVE command to save FORM1040 to disk.
(4) Use the TRANSFER LOAD command to bring in SCHEDULE-A.
(5) Complete SCHEDULE-A , the use TRANSFER SAVE to put SCHEDULE-A to disk.
(6) Bring in FORM1040 and complete the form, then save it for posterity.
(7) REMEMBER to set RECALC to NO while entering your data.

If you still need to acquire MULTIPLAN , call me at 370-5215.

One of the functions that a computer can be called on to do is SORT. However , there are several methods or algoriths that could be implemented on a computer to perform a sort. Starting with the March meeting and ending with the April meeting, there will be a demonstration using TI Basic of 5 different sort techniques:

(1) Selection Sort - March
(2) Bubble Sort - March
(3) Heap Sort - March
(4) Shell sort - April
(5) Quick Sort - April

At the April meeting, all 5 sorts will be compared for speed relative to the number of records to be sorted.

Also for March, Chuck will demonstr ate a new utility to be used with PRBASE (bring an initialized diskette) . As a remainder, PRBASE IS FAIRWARE, if you use it, pay the author. This is one way to encourage software development for the TI. The Software Library club returns from vacations. Bob Burgess will also be on hand conducting a demonstration of MINIWRITER III from Data Biotics.

In this newsletter you will notice a new feature, TIPS FROM THE TIGERCLUB by Jim Peterson of Colombus, Ohio. His newsletter contains many nice features so if you like it let me know. If the response is favorable , it will be include in future newsletters. NOTE - support Jim by ordering some of his programs.

FOR SALE:

TI Expansion Box with 32K and ss/sd disk drive - $285
2nd Disk drive with power supply in case - $85
Many TI modules - $5 and up

Call Nick at 372-1178 or see Art at meeting

Arthur F. Payeur