

(751) 8706  
Upstate

UPSTATE UPSTATE UPSTATE UPSTATE UPSTATE UPSTATE UPSTATE UPSTATE UPSTATE UPSTATE

OUR NEXT MEETING will be on Thursday  
JUNE 18 ,1987 AT 7:30 PM

THE JULY Meeting will be  
TO BE DISCUSSED

PLACE: CAPITAL DISTRICT PSYCHIATRIC CENTER  
New Scotland Ave. Next to Albany Medical Center

The program for the JUNE meeting is as follows  
A demonstration of Plato interpreter and course software.  
A presentation of the advanced features of PRBASE.  
Member's presentations of favorite software  
The Software Library Club will be present.

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.

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

---

---

---

VOL V, NO. 6

JUNE 1987

Arthur Payeur, Temp. Editor

(continued from the May newsletter)  
Editor/Assembler  
-----

The primary one is EDITOR/ASSEMBLER which is necessary for creation of relocatable object files (... well you could write one with a word processor but that would be masochism of a high order, exceeding even direct POKEing of machine code bytes). E/A will load any of the machine code file types mentioned above, from its menu screen. The LOAD AND RUN option handles both uncompressed and compressed tagged object files, and will resolve REFERENCES by name from one object file to another, or to standard system names. Uncompressed object files represent bytes in Hex notation, and take about twice as much disk space as compressed object files. Invocation of LOAD AND RUN re-initializes the memory pointers completely while loading the system utility routines such as VMBW from GROM storage, so if a sequence of file loads is interrupted by an error, it must be started all over again. E/A adds CALL LOAD and CALL LINK to console Basic to allow these same object files to be loaded and accessed from Basic. The standard utilities such as NUMREF for communicating with the Basic program must be loaded as a separate file BSCSUP. E/A will also load and run from the RUN PROGRAM FILE option, program files of machine code, prepared according to a specific recipe as SAVE files. The details of these will be a subject for HV99 News articles in the future. It will load them from cassette as well but I can't see anyone doing that in preference to using disks, unless perhaps they have installed the TIUP internal memory expansion mod in the spare console that gets taken away on holidays. No provision exists in Basic to load SAVED program files from Basic as they could overwrite part of the Basic program in the VDP on the way in. Extended Basic ----- The next module which can load assembly code is our old friend EXTENDED BASIC. This is much more limited than E/A in what it will handle. Firstly it recognizes only the 8K low memory area from >2000 to >4000 for loading relocatable object code. Absolute code can be loaded, or RAM buffer areas used in the lower part of high memory once it is known how far down this is filled by the XBasic program. The loader does not handle external REFERENCES, and the utilities loaded by CALL INIT in XB are missing the most useful one - DSRLNK, and GPLLNK as well. The Basic support utilities are loaded by CALL INIT from GROM. The assembly source code has to locate them with EQUates. A minor difference from E/A is that CALL LINK always hands over control in the GPL workspace. Programs written to LINK to E/A Basic will almost always need at least minor modifications to LINK to XB successfully. The operational hangup with XB is that the loader is written in GPL and is painfully slow. A long assembly routine, such as Text to Speech, may take several minutes to load (shades of the Commodore 64's disk system). The usual way round this is to load an assembly language loader which in turn does a faster load of the longer program. The great virtue of the XB module that sets it apart from the others is that it supports auto-RUNning from disk, as soon as the module is selected, of an XB program DSK1.LOAD which can then load further programs. The other reason for preferring XB is that it is a vastly more powerful language than the mildly enhanced console Basic offered by E/A. Unlike E/A it can never load machine code programs without Basic as an intermediary. Mini-Memory ----- This module has its own particular charm as the only one which allows access to machine code without the 32K memory expansion and using cassette storage. In this mode a LINE by LINE (or immediate input) Assembler allows standard TMS-9900 mnemonic assembly code to be entered in a restricted format. This is a descendant of TI's board level 990 evaluation systems. Only 700 odd bytes are available in the module's 4K of CMOS RAM after loading the assembler but I can't imagine anyone wanting to do programs longer than that with L by L. Still it's enough to do a pretty fair Game of Life program. MM also contains a full set of system utilities and Basic support routines in ROM and EASYBUG in GROM, a monitor program that is

useful but much less powerful than the E/A DEBUG. MM is even more useful in a fully expanded system. It does not provide the Editor and Assembler features of E/A, but offers more scope for loading and running programs. Firstly there is 5-6K more RAM available, 4K of CMOS RAM in the cartridge and the saving of space in RAM because of the utilities in cartridge ROM. Its principal deficiency as compared to E/A is the lack of a PROGRAM file loader, but this can be easily remedied by writing your own to reside in MM cartridge RAM. Even the L by L Assembler, as well as EASYBUG, remains useful for occasional little purposes anywhere in RAM, and I have prepared a disk based version for convenience. E/A object code, even compressed, is loaded successfully as long as REFs are used for system utilities and Basic support routines. EQUates as used for XB code will only get it right for one module. The loader has one more space, cartridge RAM, to place relocateable object code as a last resort. I have not yet experimented to see whether the loader will link object files with external references as the E/A loader does. The MM manual, never a fount of information, is silent on this point. MM does not erase its DEF table unless it is explicitly done by one of several means. The table survives a return to the title screen, and even switch-off if the internal battery is still alive. This is different from E/A's workings, and must be taken into account for better or for worse. Code in memory expansion does not survive switch-off even if its name lives on. TI-WRITER ----- Now we leave the modules which can load files under any name for one which loads program files with particular names. TI-WRITER tries to load an E/A type SAVE file from DSK1 under the name EDITAL (and successor filenames) when the EDITOR option is selected from the menu screen. If you have followed the E/A manual's advice on using the SAVE utility with TOMBSTONE CITY as the victim, take the file so generated and place it on a fresh disk under the name EDITAL and place in DSK1. Then fire up TI-WRITER, choose the Editor option, and see what happens. Extension to Formatter and Utility options are obvious. It may provide light relief to heavy TI-WRITER sessions. More seriously, short of writing a PROGRAM file loader to be loaded by XB using DSK1.LOAD, it is the nearest that the TI-99/4a comes to an auto-loader for machine code program files.

#### UPSTATE NEWS

Stan Garvey Writes:

After acquiring the Basic Loader I started to play around with Call Peek and Call Load with some measure of success. Previously I played with Minimem and some programs starting with the Sept 85 issues thanks to Mike Henry. They all worked. I started to convert these to XB and 32k and was not able to do so. I then put in lower case Feb 86 and it worked. Using this as a guide I converted CLEARS from hex to decimal object code with a minor change at the last line from B to B\*R11 and it worked. I then worked on count, typer, and aocol changing the MM operands to XB equates. (BLWP <6028 to BLWP <2024 , see appendix

ED. Asm manual). This is also done in lowercase. I found THE MM manual helpful. None of these conversions worked and I still do not know why? Can anybody help? My last success was a program from a book of TI Programs

at

the Clifton Park library (001.64d) called plot which does bit map graphics to draw a sine and cosine wave and a circle. This worked very nice and was very interesting when I experimented with the basic circle portion (call link specifically. I am writing this for help and your information. The article in May 87 got me motivated. Any comments Mike Henry? How about some followups from members. I notice Basic loader is being used a lot in Compute, Byte, and PC magazines.

PH. 355-3327.

I wish to thank Stan for the above letter and encourage other members to contribute to this newsletter.

Speaking of contributing : I can hardly wait to see what the members bring to the meeting to demo. BRING BLANK FORMATTED SS/SD DISKS if you wish to copy public domain software.

I still have not had any volunteers to bring equipment to the June meeting. Because of personal difficulties I am unable to bring a computer and PEB. So if we are to have a meeting with hardware, please call me. Well we can always sit around and stare at each other.

Rich Lane still wishes to acquire ss/sd disk drives cheap. See Rich at the meeting if you have one to sell.

Summer fast approaches and the subjects of meetings and newsletters will be discussed at the meeting. My gut feeling is to start back up in September. Be there if you wish to comment.

Arthur F. Payeur  
President