

10  
2/2  
January 22, 1978

Dear Bob,

First off, I am writing to request the Executive Software information you mentioned in Arcadian#3; my check for \$1 is enclosed.

I am looking forward to receiving this info, because I am working on getting an Assembler to run in my Arcade, which requires, of course, I/O routines, which I would rather not have to write. Part of this project is a memory board, which I'll describe when (and if) I get it working.

On other matters, I haven't explored this fully, but my CALL statement requires that the DE registered pair be left untouched. There is no problem if the CALL is from the keyboard, but if the Call is from a Basic program, the Basic will not properly restart, and gives an error. The easiest way around this is to push DE onto the stack, then Popping it.

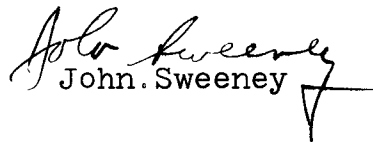
Also along these lines, I have found a convenient place to put machine language sub-routines for Basic programs. The problem, of course, is finding a place which the Basic won't write over the program when it issues an error, etc. My solution is to use the editor-buffer. This begins somewhere after "Z" (20128). I start at 20200, and have successfully run routines 30 bytes long. It looks like there might be space for 50 or 60. The advantage to this is that the Basic never uses this memory, except when statements are being entered from the keyboard. The choice of 20200 comes because this allows 30 or 40 bytes for the editor buffer, which means that statements and commands this long can be entered after the machine language routine.

I am surprised that no one has mentioned the output port &(10) in the newsletter. The value output here sets the number of lines which the video chip displays on the screen. This varies from 0 to about 204 or 205 (slightly higher values start re-scanning the screen with a  $\frac{1}{2}$ -line shift sideways). The Basic still alters the un-displayed memory, however. For values greater than 174, that part of the memory which the Basic reserves for its own use is displayed. Aside from "investigation" revealing this memory allows a quick visual "monitor" to check on the general progress of a program.

Finally, the Basic manual says that the language is a derivative

of Palo Alto Tiny Basic, and I have found a great deal of information in reading the sources of it. There are several articles on the subject, but the one I have access to is "Palo Alto Tiny BASIC Version Three" by Li-Chen Wang in PCC's Reference Book of Personal and Home Computing. This article defines the language, and virtually all of what it says is implemented in Bally Basic. There are several interesting logical features, including logical expressions ((A=3) may be used as a variable, and is assigned 0 or 1 depending on whether it is true or false). Also , GOTO, and GOSUB may be followed by expressions, which will be evaluated, and the transfer executed to the line number corresponding to the answer. There is much more, so I would encourage everyone to get hold of this book, or Doctor Dobb's Vol. 1, or one of the other articles on PATB version 3.

I'll be looking forward to receiving the Executive listing, and to further issues of Arcadian.

  
John Sweeney