

# KIM Tidbits

## Expanding Your System

Harvey B. Herman  
Department of Chemistry  
University of North Carolina  
at Greensboro  
Greensboro, North Carolina 27412

The January 1981 issue of **COMPUTE!** carried a brief review by me, of a disk operating system for an expanded KIM-1. I have been asked (can you be locked up for talking to yourself?) to supply details on how to expand KIM into a complete system including a disk drive. Even if you have reservations about this, as I am sure some readers do, read on and see how I did it. I am quite pleased with the finished project and feel it may make sense for other KIM owners to follow my lead.

### First Steps

My work with microcomputers began in the summer of 1976 when I purchased a KIM-1, a small power supply and teletype (KSR-33). My justification was to have these serve as both an educational tool for me and my students and as an instrument controller. Our first project, a success, was to develop a machine language program, hand assembled by Jeffrey Schmoyer, to control a syringe pump. Hand assembly is a drag and the next natural step was addition of extra memory in order to run an assembler. An article in 6502 Users Notes (November 1976) encouraged me to order a 4K \$100 memory board (S.D. Systems, Dallas, TX 75228) and attach it directly to the KIM bus.

At this point I was able to run a small assembler and tiny BASIC. I quickly saw the value of hybrid programs which mixed experimental control (in machine language) and data analysis (in BASIC). However, in general, the later required a floating point BASIC. Microsoft 8K BASIC was, in late 1977, just being offered by Johnson Computer (Medina, OH 44256). This program needed additional memory and I felt compelled to take the next step and add a mother board, KIMS I (Forethought Products, Eugene, OR 97402), which could support \$100 8K static memory add-ons, e.g., Econoram II (Godbout Electronics, Oakland Airport, CA 94614).

### Willi's Advice

Each step in the expansion of KIM this far was relatively painless with no problems and did not appear overly expensive. Now I was able to run a very nice enhanced BASIC (see previous KIM Tidbits) and a more comprehensive assembler.

The only frustration was loading from cassette tape (even at 6x normal speed using Butterfield hypertape). Obviously a conversion to floppy disk was the way to go but none of the commercial systems was in my budget range. I did have an 8" disk drive and homemade power supply left over from an aborted project. Willi Kusche (Wilserv Industries, Haddonfield, NJ 08033) noted my dilemma and constraints on a visit here in the later part of 1979. He informed me that with the addition of a disk controller, cable, and software, which he sold, I could get a disk operating system (DOS) up on my KIM. This idea appealed to me and even though the going was not perfectly smooth I'm glad I followed his suggestion and began to put the system together.

### Adding The Disk

The first step was to construct the disk controller board (Versafloppy I, S.D. Systems). Table I shows the jumpers we made on the board. The kit was not too difficult to make but one chip was bad and I clumsily blew out several more before I got the board working. After that auspicious start we (Leon Stokes is we, I is me) put together a cable from the controller to the drive. The disk drive was old enough to have a non-standard connector and we could not purchase a ready-made cable. Table II shows the interconnections that were necessary, between the

controller and the drive. (Innovex 220M, now Innotronics, Lincoln, MA 01773). Sad to tell on power up the head loaded but nothing else happened. We subsequently found that the original lubricant (not used now) had congealed because the drive had not been used for four years. Our man in the machine shop (Henry Teague) freed things up for us and now everything worked. I was able to load the disk Willi had sent and made a back-up copy with his utility program.

### Observations

My students and I have been using this system for slightly over a year. We originally loaded a short disk boot program from tape but are now using a 16K EPROM board (Digital Research Computer, Garland, TX 75040) with just one 2708 programmed by Willi for us. We make extensive use of disk data files which are then used as input to separate data analysis programs. A simple editor program (written by Becky Efird) prepares the data for storage on disk. A listing of the very similar PET version of the program is shown as illustration of the features of the DOS. The KIM version of this program uses the USR function for monitor calls as opposed to SYS for the PET. Otherwise the programs are almost identical.

Disk data files have the decided advantage of allowing recalculation by alternate models without rekeying of data. About 128K bytes of data can be stored on each 8" disk. While some of the space is taken up by the disk directory and DOS the latter can be optionally omitted on initial formatting. Backup of data files is done with a copy utility and a second diskette or could even be done on tape. As further insurance I am considering a second system to protect against hardware failure.

### Conclusions

Would I recommend this approach to others? Yes and no. The additional cost to me for disk operation was nominal as I already had the mother board, extra memory, disk drive and power supply. However, if I was starting from scratch, a complete system based on an Apple or PET would probably have been cheaper. Furthermore, there is a potential noise problem with an exposed single board computer which other commercial expansions might minimize. I have had no problems with this but it should be mentioned.

Working within a limited budget where only a few things could be purchased at a time, I was able to assemble a very nice system. I did have some difficulties but was able to overcome. Get a commercial system if you have the money. If not, consider occasional add-ons with the object of reaching a disk based system at some future point. It really is great bringing up BASIC in less than 4 seconds. You will never be satisfied with less.

Table I

Versafloppy Jumpers for 8"		Technical Function
Innovex 220M Drive		
E1 - E2		Port E3/3 to DRVSEL 4
E4 - E5		Port E3/2 to DRVSEL 3
E7 - E8		HLA to HEAD LOAD
E10 - E11		Port E3/3 to DRVSEL 4
E15 - E17		HLT(1771) to Q(U15)
E21 - E22		4MHz clock (8" drive)
E24 - E25		pull up on pin 2 of U2
E32 - E33		Ports E3 to E7 (locations F0E3 to F0E7 on KIMSI)

Note -

Some jumpers are etched on boards (rev A) and may need to be cut. Most jumpers are same as given for Altair mainframe with Shugart SA800 disk drive.

Table II

Controller	8" Disc Drive	Function
Versafloppy I	Innovex 220 M	
36	L6	step
34	L15	direction
26	L13	device select
20	L5	index
42	L12	track zero
6	L21	write current select
18	L18	head load
40	L7	write gate
22	L8	ready
46	L17	separated data
38	L10	write data
44	L16	write protect

NOTE: Power - +5, -5, +24V, and AC made to drive.

Hard sector option on drive was disabled.

Unseparated data sent to controller on separated data line.

©