

Load And Save KIM BASIC Programs On Your SYM

George Wells
LaVerne, CA

The SYM and KIM microcomputers are close cousins. Their hardware and tape interface compatibilities are well known. Not so well known is the fact that the BASIC interpreters on the two systems use the same tokens for their reserved keywords which makes transfer of BASIC programs between the SYM and KIM almost trivial.

Mike Hanna, a friend of mine who has had a KIM with BASIC for much longer than I have had my SYM with BASIC, has offered to share his library of BASIC programs with me. We had considered implementing a telephone/modem interface to accomplish this transfer but after comparing the disassembly listings of the two interpreters we decided a tape transfer would be easier. The scheme we finally settled on allows the SYM to create and read tapes in the original low speed KIM format since the SYM does not support any of the faster versions. Going from the SYM to the KIM is particularly simple; going the other way requires a short BASIC program (see listing).

SYM To KIM Transfer

STEP 1: Load the BASIC program to be transferred into the SYM.

STEP 2: Exit BASIC and return to the Monitor (by way of reset, for example).

STEP 3: Determine the end of the BASIC program by examining the two-byte pointer stored at \$7D/\$7E by entering .V 7D-7E. The SYM will respond with:

```
007D uv wx,yz
```

where wxuv is the end of the program (qrst) plus one. The monitor will calculate qrst for you if you can't do it in your head by entering .C wxuv-1.

STEP 4: Save the program on tape in KIM format by entering:

```
.S1 1,201-qrst
```

where qrst is the value from STEP 3.

STEP 5: Load the program into KIM BASIC in the normal manner.

KIM To SYM Transfer

In order to load KIM formatted BASIC programs into your SYM you will need to have a copy of the KIM BASIC PROGRAM LOADER listed with this article. Save this program on tape (in high speed format, of course) so that you will have it whenever you need it. NOTE: This program will not work with Monitor Version 1.0 which has an error in the KIM Load routine.

```
LIST: KEM      KIM BASIC PROGRAM LOADER
```

```
100 A=USR("&"@B86"+0)
110 A=42572: POKE A,1: POKE A+1,2: POKE A+2,255
120 FOR I=0 TO 29
130 POKE 300+I, PEEK(35960+I)
140 NEXT I
150 POKE 330,96
160 PRINT "AFTER 'LOADED' MESSAGE, ENTER:"
170 PRINT "POKE 125,PEEK(254): POKE 126,PEEK(255): CLEAR"
180 PRINT USR(300+"&"@C05"+&"@C0C"+0)
OK
```

STEP 1: On the KIM, save the program to be transferred in the normal manner; but make sure it is saved at the original tape low speed.

STEP 2: Initialize BASIC on your SYM and LOAD and RUN the KIM BASIC PROGRAM LOADER.

STEP 3: Play the tape with the KIM program in your recorder. If you have implemented a second cassette control for your read-only recorder you will have to over-ride it since this program will only activate the original cassette control.

STEP 4: After the LOADED message, enter the command printed by the program and then SAVE a copy of the KIM program in high speed format. In case you get a BAD LOAD message, start over again at STEP 2.

SYM/KIM BASIC Incompatibilities

The obvious hardware related incompatibilities due to different address availability in the two systems require careful use of the PEEK, POKE and USR commands. Of course, different terminals may also have special requirements for cursor controls or graphics capabilities. Not so obvious are the following additional potential problem areas.

GO: SYM treats GO as a reserved word so don't enter GOTO as two words. Also make sure that GO does not appear in any variable names such as DRAGON.

GET: SYM does not implement this function but it does reserve the same token as KIM. (See MICRO 24:15 if you want to implement GET on your SYM.)

USR: The multiple parameter versions of USR will not work on the KIM. The single parameter version will require a different set of POKE commands prior to the USR but otherwise it works the same in both systems.

& "ABCD": KIM does not support hexadecimal notation.

Theory Of Operation

The key to the success of this scheme is in the fact that the Microsoft BASIC interpreters automatically recalculate the internal line pointers whenever a BASIC program is loaded. Theoretically, this means that a program that was originally saved at one address could be loaded into a different address if the tape load routine is given the ID value of \$FF and the address where the program is to be loaded. The problem is that in some versions of KIM BASIC the ID value of \$FF is used to save programs which means that since the IDs match when loading, the programs get loaded into their original address instead of the new specified address. There are two ways to fix this problem. First, if you have access to the KIM you can change location \$2744 from \$FF to \$01 before the program is saved. This is part of the sequence LDA *\$FF, STA \$17F9.

The second method is the one the KIM BASIC PROGRAM LOADER uses which will work with any ID. It requires making a copy of the beginning of the SYM Monitor Load routine on page one of the SYM up to the point where the ID test is made. The FOR/NEXT loop in the LOADER program copies the code between address \$8C78 and \$8C95 and then an RTS instruction is attached to the end (\$60 = decimal 96).

The jump to continue into the Monitor Load routine is performed by an interesting technique which Hans W. Gschwind of West Germany wrote about in SYM-PHYSICS 4-20. It involves using the multi-parameter version of the USR function by pushing two return addresses on the stack so that when the first subroutine finishes it returns to the address equal to the third parameter of the USR command plus one which is the continuation point in the Monitor Load routine. The next RTS instruction encountered returns to the address of the second parameter plus one which is the normal return point for BASIC high-speed tape loads.

With this background in mind it is possible to understand the following line by line explanation of the KIM BASIC PROGRAM LOADER.

LINE 100: Calls the Monitor ACCESS routine to allow passing of tape parameters to System Ram.

LINE 110: Passes tape start address of \$201 and ID of \$FF to tape parameters.

LINES 120 to 140: Copies first part of Monitor Tape Load routine to page one.

LINE 150: Ends page one copy with an RTS.

LINES 160 and 170: Prints message to be entered after a good load. The command must be entered manually since the KIM BASIC program will overwrite the LOADER program.

LINE 180: Jumps to address 300 (first parameter) with Y index register equal to zero (fourth parameter) indicating KIM tape format. The RTS at address 330 jumps to address \$8CAD (third

parameter plus one). The RTS at the end of the Monitor Tape Load routine jumps to address \$C6C6 (second parameter plus one) in the BASIC interpreter which modifies the line pointers to fit the new location in the SYM.

Conclusion

Hopefully this scheme can be used to advantage by anyone having access to both a SYM and a KIM. If you find that it just doesn't work for you, try a different tape recorder. Mike and I spent many frustrating days trying to get the SYM to KIM transfer to work and it wasn't until I used a different recorder with my SYM before we finally did have success! Now we are able to transfer our BASIC programs with ease.

©