HIGH SPEED CASSETTE I/O
FOR THE KIM-1

BY THEODORE E. BRIDGE
54 Williamsburg Dr.
Springfield, MA 01108

This high speed cassette tape handler will load or dump at 12 times the speed of KIM-1 or twice the speed of the MICRO-ADE[1] Editor-Assembler. It replaces and overlays the editor routines CREAD and CWRITE at the end of the third 4K block of memory (2EA9-3000).

If you are not now using the MICRO-ADE[1] Editor-Assembler, you should consider it. I think it is well worth the $25 I paid for it.

Stand-alone operation is permitted with this high speed tape handler. You may call it from any program as follows:

20	A9 2E	JMS CREAD	Cassette Read
20	54 2F	JMS CWRITE	Cassette Write

If your program is to call CWRITE, be sure it first loads the starting and ending addresses as given on page 47 of the KIM-1 User Manual. This is just as if you were dumping with KIM-1 before starting at 1800.

You can use these high speed routines to dump from the keypad. Load the addresses as with KIM-1, but instead of starting at address 1800, start at address 2F54. You can also load from keypad control. Instead of starting at 1873 with KIM-1, start at 2EA9. This loader will save starting and ending addresses ready for you to copy the file on cassette by starting at 2F54.

Trade-offs

You pay a price for the increased speed. When you use the MICRO-ADE editor, with its HYPERTAPE[2] tape dump, you need a better cassette recorder than you would with KIM-1. I have had good results with my Realistic[3] seventy dollar recorder.

When you try to double the speed using these routines, you lose compatibility with KIM-1 and you lose the search capability of the monitor. You always load the next program on the tape rather than hunt for the (ID) you want. You need a meter on your recorder so that you can position the read head just in front of the program you want to load. Also, you must keep a journal showing the cassette number and the meter reading for all data you record on tape. You really ought to do this anyway.

Loss of the search capability is not altogether a disadvantage. If you have only one copy of a program on tape, a blip on the tape may prevent your loading with the MICRO-ADE loader. But with this high speed loader, even if you flag a checksum error, most of the program will probably load properly. I have found it helpful. Either way, you really need two cassette recorders: one for input, and another for output.

You pay yet another price when you use these high speed routines during an assembly. When you load the object tape, it will reside at address 0200 where it was assembled instead of at the ORG address where it belongs. After loading, you must use the BLOCK MOVE command to put it in its proper location. After all units of your program have been properly relocated, you must perform a SAVE operation to get a usable object tape.

Apart from making a quicker compilation when using these routines (even after relocating as described above), you can also recover more quickly from checksum errors encountered while reading the source tape during compilation. I have found them very convenient for general housekeeping as well as for compilation.

Description

These routines are similar to the MICRO-ADE HYPER-TAPE[2] routines that they replace. They simply gulp the entire 8 bit byte at once instead of taking it in two nybbles. Thus, they go twice as fast. The CREAD routine cannot look for an end-of-data symbol ("/" = 2F) to terminate the load as this might appear anywhere in the data being loaded. Instead, it reads the start and end addresses at the beginning of the tape. It terminates the load when the loading address (VEB+1, 2) equals or exceeds the end address. At the end, the display is lit; so that you can see if a checksum error has occurred; and if you have the right ID.

Since we are not loading ASCII characters, we lose the validity check on each character as is made in the HYPERTAPE program. We must depend on the checksum error to flag all errors. This has not been a problem for me. Actually, I think it is easier to recover from a checksum error when using these

high speed routines than when using the HYPERTAPE routines.

At the end of a tape load, this high speed loader will halt with the LED display showing:

    00F9 (ID)   if there was no checksum error, and
    17F9 (ID)   if there was a checksum error.

You look to see if the ID is the one you want. If you depress (C = Continue) you will return to the calling program. If you depress (B = Bad tape) you will read the next program on tape. Of course, you can rewind to try again on the same data if you wish. The VU-TAPE[2] featured by Jim Butterfield is incorporated in this high speed loader. With it you can see the loader lock on to the sync pulses when it starts to load.

At the end of a tape dump, you will halt with the LED display lit. If you depress (C = Continue), you will return to the calling program. If you depress (B = Begin) you will load the next program on the input tape. If you depress (RS, AD, 2F54, GO), you will dump another copy of the data to the output cassette.

Remote program control of the cassette recorders is provided for in these high speed routines. I use the HELP RELAY PACKAGE[4]. Application pin 11 activates the input cassette, and pin 12 activates the output cassette. If you are not now using automatic tape control, you should consider it. Later, we will describe two patches: one to permit manual tape control, and another to slow down the speed if your cassette recorder can't handle the high speed tapes.

Tape problems are not to be taken lightly. I have adopted the following procedures to try to minimize these problems.

1. Always rewind before ejecting a tape.
2. Stop the recorder before removing remote control plug.
3. Use short tapes. Short tapes can be obtained from PYRAMID DATA SYSTEMS[5] – five 10 minute tapes for $3.00, or five 20 minute tapes for $3.25.
4. You need a medium quality cassette recorder. Robert Tripp[4] reports good success with Panasonic RQ212 DAS, and Radio Shack Realistic CTR 39 model 14-819.

To load these high speed routines from cassette, you use the KIM-1 tape loader at 1873. Therefore, you must keep a copy on cassette in HYPERTAPE or KIM-1 format. I can furnish such a cassette for $4.00 or a paper tape for $2.00.

Patches

This patch for the Micro-Ade Assembler-Editor will link with the high speed routines:

    2E97 4C A9 2E   JMP CREAD
    2E9A 4C 4F 2F   JMP CWRITE

This manual control patch will permit you to operate without relays for program control of the cassette recorder:

    2EA9 A9 1C   LDAIM $1C
    2EAB 85 FB   STA POINTH
    2EAD 85 FA EA   STA POINTL
    2EB0 20 24 2F   JSR DISP

On entering CREAD, this patch will display 1C1CXX on the LEDS to remind you to start the input cassette immediate-

ly after you depress C to continue with the reading of the tape.

At the end of the load, the CREAD routine will display: 00F9 (ID) on the LEDs if there was no checksum error, and 17F9 (ID) if there was a checksum error. At this time, you must remember to stop the input cassette before depressing C to continue, or B to reread the tape.

On entering CWRITE, this patch will display OCOCXX to remind you to start the output cassette and to wait a few seconds before depressing C to continue with the dump:

    2F56 85 FB   STA POINTH
    2F58 85 FA EA   STA POINTL
    2F5B 20 24 2F   JSR DISP

Before leaving the CWRITE routine, this patch will display COFF on the LEDs to remind you to turn off the output cassette beofre depressing C to return to the calling program:

    2FA4 A9 CO   LDAIM $CO
    2FA6 85 FB   STA POINTH
    2FA8 A9 FF   LDAIM $FF
    2FAA 85 FA   STA POINTL
    2FAC 4C 24 2F   JMP DISP

A patch to lower the recording speed is given at the end of the listing. This patch is necessary only if your cassette recorder cannot handle the higher speed.

Conclusion

I think it is very worthwhile to speed up the tape operations by use of these routines. I recently wrote a program to compare the dental record of a corpse with up to 60,000 missing person records to seek a match. In writing this program, I was very conscious of the need for faster cassette reading. With these routines, it would take 4 hours to read 60,000 records. With HYPERTAPE it would take eight hours. With the KIM-1 loader it would take 48 hours.

In writing a program like this HIGH SPEED I/O, with only two overlays, the assembler reads the source tape twice to edit and twice to compile. Also it spends the same time to save a corrected copy, being equivalent to six reads. If we make ten corrections during the debugging, we would spend one hour for I/O using HYPERTAPE.

Furthermore, it is very convenient to use these routines Stand-Alone. It is nice to do a little housekeeping without needing to fire up the teletype and the Assembler-Editor.

Notes

1. MICRO-ADE[TM] for the 6502—Assembler, Disassembler, Editor, by Peter F. Jennings. Micro-Ware Ltd., 27 Firstbrooke Road, Toronto, Ontario, Canada, M4E 2L2.
2. *The First Book of KIM*, Hayden Book Co., Essex St., Rochelle Park, N.J. 07662. Cat. 5119-0. $9.00.
3. Realistic[TM] CTR-39, Model 14-819 by Radio Shack.
4. HELP RELAY PACKAGE, the Computerist, P.O. Box 3, S. Chelmsford, MA 01824.
5. Pyramid Data Systems, 6 Terrace Ave., New Egypt, N.J., 08533.

MICRO-MATE ASSEMBLY CSSM-1.0 PAGE 01   9-0-78

These high speed I/O routines are patterned after the low speed
(hypertape) routines that they replace in the MICRO-MATE assembler,
Editor by Peter Jennings.

```
                COMPATIBLEST HALF-TELEX FORMAT
                NON COMPATIBLE
                CASSETTE I/O

        OLTCH   *
        RETRY   *
        OTLF    *
        CHKL    *
        CHKH    *
        CHKT    *
        CRUC    *
        TIC     *
        COUNT   *
        ACC     *
        CKT     *
        TRIL    *
        POINTL  *
        POINTH  *
        PACKT   *
        SED     *
        TIBIT   *
        SAL     *
        SAH     *
        EAL     *
        EAH     *
        ID      *
        INIT    *
        INTVEB  *
        INCVEB  *
        VEL     *
CREAD   LDAIM   $CC
        STA     $17C3
        ANDIM   $FE           SET BIT 2 = 0, TO TURN ON
        ORA     CRUC           CASSETTE
        STA     CRUC
        STA     $17A0
        CMPIM   $16           PB2 APPLICATION PIN 11
        BNE     SYNC
        JSR     RDCHT
        STA     $17A0
        CMPIM   $2A           IF INPUT, DISPLAY 17
        BNE     TST
        STA     RDCHT          TURN ON KIM DISPLAY
        JSR     ID
        STA     $F9

SYNC    LDAIM   $13           TURN ON INPUT PORT
        STA     SEL
        JSR     RDBIT          READ A BIT AT A TIME
        LSR     CRUC           SHIFT INTO SYNC
        ORA     CRUC
        STA     CRUC
        STA     $17A0
        CMPIM   $16           PUT ON LED
SYNC                          IS IT A SYNC ?
                              KEEP TRYING
TST     JSR     RDCHT          IN SYNC, READ CHARAC
        STA     $17A0          DISPLAY ON LED
        CMPIM   $2A *          START OF DATA * ?
        BNE     TST            KEEP TRYING
        JSR     RDCHT          READ ID
        STA     ID             SAVE IT
        STAZ    $F9
```

```
0570: 2EE8 A2 03           LDXIM $03       GET 4 ADDRESSES
0580: 2EEA 20 48 2F LOADAD JSR   RDCHT
0580: 2EED 9D F5 17        STAAX SAL
0590:              CA       DEX
0600: 2EF0 CA              BPL   LOADAD
0610: 2EF1 10 F7           LDAIM $8D       SET UP VEB
0620:      A9 8D           STA   VEB
0630: 2EF5 8D EC 17        JSR   INTVEB
0640: 2EF8 20 32 19        JSR   RDCHT
0650:              2F READIT JSR  CHKT     ADD TO CHECKSUM
0660: 2EFE 20 4C 19        JSR   VEB       STORE IT
0670: 2F01 20 EC 17        JSR   INCVE     INCREMENT ADDR.
0680: 2F04 20 38 2F        BCC   READIT    GET NEXT BYTE
0690: 2F07 90 F2   ENDRD   JSR   RDCHT     GET CHECKSUM
0700: 2F09 20 48 2F        CMP   CHKL      AND CHECK IT
0710: 2F0C CD E7 17        BNE   AOK
0720: 2F0F D0 0B           JSR   RDCHT
0730: 2F11 20 48 2F        SEC
0740: 2F14 38              SBC   CHKH
0750: 2F15 ED E8 17 AOK    BNE   AOK
0760: 2F18 D0 02           STA   POINTH
0770: 2F1A 85 FB           LDA   $1702
0780: 2F1C AD 02 17 AOK    ORAIM $04      SET BIT 2 =1, TO
0790: 2F1F 09 04           STA   $1702     STOP CASSETTE
0800: 2F21 8D 02 17        JSR   $1F19     SCAND
0810: 2F24 20 19 1F DISP   JSR   $1F6A     GETKEY
0820: 2F27 20 6A 1F        CMPIM $0C
0830: 2F2A C9 0C           BEQ   DW
0840: 2F2C F0 07           CMPIM $0E       CONTINUE
0850: 2F2E C9 0E           BNE   DISP
0860: 2F30 D0 F2   DW      JMP   CREAD     EAD TAPE
0870: 2F32 4C A9 2E DWN    JMP   INIT
0880: 2F35 4C 8C 1E INCVE  JSR   INCVEB    POINT TO NEXT BYTE
0890: 2F38 20 EA 19        LDA   VEB       +01 CHECK FOR END
0900: 2F35 AD ED 17        CMP   EAL
0910: 2F3E CD F7 17        LDA   VEB       +02
0920: 2F41 AD EE 17        SEC
0930: 2F44 ED F8 17        RTS
0940: 2F47 60      RDCHT   JSR   $1A24
0950: 2F48 20 24 1A        LDA   $17EA
0960: 2F4B AD EA 17        RTS
0970: 2F4E 60      CRITE   LDAZ  $62
0980: 2F4F A5 62           STA   ID
0990: 2F51 8D F9 17
ID=31

0010:              THIS OUTPUT ROUTINE WAS ADAPTED FROM HYPERTAPE
0020:              BY JIM BUTERFIELD. SEE FIRST BOOK OF KIM
0070: 2F54 A9 0C   CWRITE LDAIM $0C        SET BIT 3 = 0, TO
0070: 2F56 8D 03 17        STA   $1703     TURN ON CASSETTE
0080: 2F59 29 F7           ANDIM $F7
0090: 2F5B 8D 02 17        STA   $1702     PB3 APPLICATION PIN 12
0100: 2F5E A9 AD           LDAIM $AD       SET UP VEB
0110: 2F60 8D EC 17        STA   VEB
0120: 2F63 20 32 19        JSR   INTVEB
0130: 2F66 A9 27           LDAIM $27       SET FLAG FOR
0140: 2F68 85 F0           STA   GANG      SED TOGGLE
0150: 2F6A A9 EF           LDAIM $EF       TURN ON
0160: 2F6C 8D 43 17        STA   $1743     OUTPUT
```