

FAST CASSETTE INTERFACE DESCRIPTION

The February 1977 issue of BYTE magazine (page 82) contained an interesting article on a minimum hardware cassette interface. I have used this technique to develop a cassette I/O arrangement which records and loads via tape at over 1600 baud. Because I do not unpack bytes for recording, the effective data rate is over 160 bytes/second. The accompanying software listing for 6502 systems provides a record start sequence which requires at least ten 16 bytes followed with an OF byte to be inputted in succession before loading can commence. At end of loading, a two byte checksum is used for detection of errors. The hardware consists of a direct connection from a one-bit output port to the microphone input and a non-inverting hysteresis circuit incorporating an LM339 comparator as the playback electronics. Actually, I've used a direct connection for the playback with success but some cassette decks won't work unless the comparator is used. My General Electric and two Sankyo tape decks work very well without the comparator but the Realistic deck will not operate at all without the comparator.

An interesting note is that some tape decks put the signal on the barrel of the record and play jacks instead of on the inner tip. Also, some tape decks invert the signal on playback. This inversion can be compensated by inserting an inverter (7400 or equiv.) between the LM339 and the input port.

To use this software, enter data in memory locations 0123-0127 as follows:

0123	=	LOAD/NO
0124-0125	=	START ADDRESS
0126-0127	=	END ADDRESS

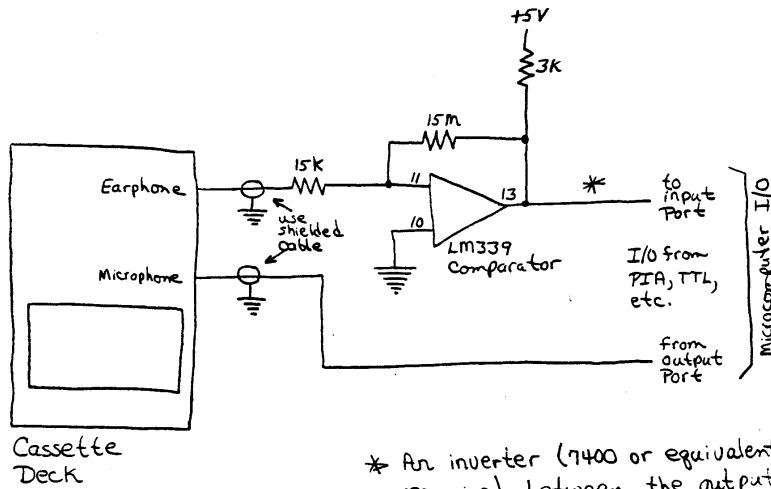
The record routine will record from START ADDRESS to END ADDRESS. LOAD/NO has no meaning to the record software.

The load routine will load from START ADDRESS to END ADDRESS but store data in memory only if LOAD/NO does not equal 0. When equal 0, LOAD/NO can be used for verifying and conditionally selecting modules on tape.

The load and record routines have callable entry points at C/WRITE (4000) and C/READ (40A5), and non-callable entry points at LOAD.ENTRY (4141) and RECORD.ENT (4152). If C/READ is called, the Z-bit in the PSR will be true on return if no error was detected and false if errors occurred. If execution is at the non-callable entry LOAD.ENTRY, a break (via BRK instruction) will be executed at end of loading and register A will indicate if the data was loaded correctly: R(A)=00 for good load, and EE for error.

To sum up, this has been a very reliable scheme and works error-free with the cheapest tapes (even Concert tapes which can be bought at many department stores at 3 for \$1.00).

FAST CASSETTE INTERFACE CIRCUITRY



LM339 Power	
	Pin
+5V	3
GND	12

If you ordered the KIM version of ASSM/TED, the cassette I/O is preconfigured for the following connections:

	Function	Pin number on Application Connector
tape deck Ø	REMOTE	Motor Control Ø
	Microphone	Cassette record
tape deck 1	REMOTE	Motor Control 1
	EARPHONE	Cassette playback

XASSEMBLE LIST *Change Underlined Portion Per Your System Requirements*

```

0010          .BA $4000
0020          .DS
0030 ;
0040 ;***** FAST CASSETTE INTERFACE *****
0045 ;      (CONFIGURED FOR KIM)
0050 ;
0060 ;
0070 ;== INPUT/OUTPUT ==
0080 C/PORT   .DE $1702   ;CASSETTE I/O
0090 C/PORTD  .DE $1703   ;PORT DIRECTION REG.
0100 ;BIT 3 = WRITE TO CASSETTE; BIT 2 = READ FROM CASSETTE
0120 ;
0130 ;
0140 ;== VARIABLES ==
0150 CHECKSUM .DE $B2 TWO BYTE CHECKSUM
0160 COUNT    .DE ADDR5
0170 FORM+BYTE .DE $B4
0180 SYNC+COUNT .DE FORM+BYTE
0190 BIT.COUNT .DE $B5
0200 ADDR5    .DE $B6
0210 ;
0220 ;INPUT PARMS
0230 LOAD/NO  .DE $0123   ;00=NO LOAD; 01=LOAD
0240 START    .DE $0124   ;START ADDRESS
0250 END      .DE $0126   ;END ADDRESS +1
0260 ;
0270 ;
0280 ;
0290 ;C/WRITE: WRITE TO TAPE FROM (START) TO (END)
0300 ;
4000- AD 03 17 0310 C/WRITE  LDA C/PORTD
4003- 03 08 0320          ORA %00001000   ;BIT 3 = CASSETTE OUT
4005- 8D 03 17 0330          STA C/PORTD
0340 ;THE ABOVE INITIALIZES BIT 3 FOR OUTPUT ON PIA
0350 ;
4008- A9 20 0360          LDA #$20 32 TIMES
400A- 85 B6 0370          STA +COUNT
400C- A9 16 0380 LOOP/RECST LDA #$16 SYNC CHAR.
400E- 20 41 40 0390          JSR WRITE/BYTE
0400 ;
4011- A9 10 0410          LDA #$10
4013- 85 B4 0420          STA +SYNC+COUNT
4015- 20 5D 40 0430 LOOP/DELSY JSR OUT:ZERO
4018- C6 B4 0440          DEC +SYNC+COUNT
401A- D0 F9 0450          BNE LOOP/DELSY
0460 ;DELAY TIME FOR SYNC
0470 ;
401C- C6 B6 0480          DEC +COUNT
401E- D0 EC 0490          BNE LOOP/RECST
0500 ;
4020- 20 84 40 0510          JSR MOVE+ST/AD START > ADDR5 (2)
0520 ;
4023- A9 0F 0530          LDA #$0F RECORD START CHAR.
4025- 20 41 40 0540          JSR WRITE/BYTE
0550 ;
4028- A2 00 0560          LDX #$00

```

```

402A- 06 B2    0570          STX  *CHECKSUM CLEAR CHECKSUM
402C- 06 B3    0571          STX  *CHECKSUM+$01
                                0580 ;
402E- A1 B6    0590 LOOP/DATA LDA (ADDRS,X) LOAD DATA
4030- 20 41 40 0600          JSR  WRITE/BYTE
4033- 20 8F 40 0610          JSR  INC/COMP
4036- 90 F6    0615          BCC  LOOP/DATA
                                0619 ;
                                0620 ;
4038- A5 B3    0621          LDA  *CHECKSUM+$01
403A- 48      0622          PHA  SAVE HI CKSUM
403B- A5 B2    0630          LDA  *CHECKSUM
403D- 20 41 40 0631          JSR  WRITE/BYTE WRITE LD CKSUM FIRST
4040- 68      0632          PLA  HI CKSUM NEXT
                                0640 ;THE ABOVE WRITES BOTH CHECKSUM BYTES
                                0650 ;
                                0660 ;
                                0670 ;ROUTINE TO WRITE A BYTE TO TAPE
                                0680 ;
4041- 85 B4    0690 WRITE/BYTE STA *FORM+BYTE
4043- 20 34 41 0691          JSR  CKSUM+ADD UPDATE CHECKSUM COUNTER
4046- 20 7C 40 0700          JSR  OUT:ONE START BIT
4049- A9 08    0710          LDA  #$08 8 BITS
404B- 85 B5    0720          STA  *BIT.COUNT
404D- 06 B4    0730 DATA/LOOP ASL  *FORM+BYTE SHIFT LEFT INTO CARRY
404F- 90 05    0740          BCC  ZERO.BIT
4051- 20 7C 40 0760 ONE.BIT JSR  OUT:ONE
4054- F0 03    0770          BEQ  CK+END+BY
4056- 20 5D 40 0790 ZERO.BIT JSR  OUT:ZERO
4059- C6 B5    0800 CK+END+BY DEC  *BIT.COUNT
405B- D0 F0    0810          BNE  DATA/LOOP
                                0820 ;NOW OUTPUT 1 STOP BIT
                                0830 ;
                                0840 ;ROUTINE OUTPUT A ZERO TO TAPE
                                0850 ;
405D- A9 20    0860 OUT:ZERO LDA  #$20 /0/ DELAY CONSTANT
                                0870 ;
                                0880 ;
                                0890 ;ROUTINE WRITE TO TAPE
                                0900 ;
405F- 48      0910 WRITE   PHA  SAVE DELAY CONSTANT
4060- AD 02 17 0920          LDA  C/PORT
4063- 03 08    0930          ORA  #%00001000      ;OUT A '1' ON BIT 3
4065- 8D 02 17 0940          STA  C/PORT
4068- 68      0950          PLA
4069- 48      0960          PHA
406A- AA      0970          TAX  DELAY CONSTANT
406B- 20 78 40 0980          JSR  LOOPD
406E- AD 02 17 0990          LDA  C/PORT
4071- 29 F7    1000          AND  #%11110111      ;OUT A '0' ON BIT 3
4073- 8D 02 17 1010          STA  C/PORT
4076- 68      1020 X        PLA
4077- AA      1030          TAX  DELAY CONSTANT
4078- CA      1040 LOOPD    DEX
4079- D0 FD    1050          BNE  LOOPD
407B- 60      1060          RTS
                                1070 ;
                                1080 ;

```

```

1090 ;
1100 ;ROUTINE OUTPUT A ONE TO TAPE
1110 ;
407C- A9 50 1120 OUT:ONE LDA #50 /1/ DELAY CONSTANT
407E- D0 DF 1130 BNE WRITE
1140 ;
1150 ;
1160 ;DELAY FOR /0/ TIME FOR READ
1170 ;
4080- A2 30 1180 READ.DELAY LDX #30
4082- D0 F4 1190 BNE LOOPD
1200 ;
1210 ;
1220 ;
1230 ;ROUTINE MOVE FROM START TO ADDR
1240 ;
4084- AD 24 01 1250 MOVE+ST/AD LDA START
4087- 85 B6 1260 STA +ADDRS
4089- AD 25 01 1270 LDA START+$01
408C- 85 B7 1280 STA +ADDRS+$01
408E- 60 1290 RTS
1300 ;
1310 ;
1320 ;
1330 ;ROUTINE INCREMENT AND COMPARE
1340 ;
408F- E6 B6 1350 INC/COMP INC +ADDRS
4091- D0 02 1360 BNE SKIP/INC
4093- E6 B7 1370 INC +ADDRS+$01
4095- A5 B7 1380 SKIP/INC LDA +ADDRS+$01
4097- CD 27 01 1390 CMP END+$01
409A- 90 08 1400 BCC NOT/END
409C- A5 B6 1410 LDA +ADDRS
409E- CD 26 01 1420 CMP END
40A1- 90 01 1430 BCC NOT/END
40A3- 38 1440 SEC
40A4- 60 1450 NOT/END RTS
1460 ;ON RETURN, C=CLEAR: NOT END; C=SET: END REACHED
1470 ;
1480 ;
1490 ;
1500 ;
1510 ;C/READ: READ FROM TAPE TO (START) TO (END)
1520 ;
40A5- A2 00 1530 C/READ LDX #500
40A7- 86 B6 1540 STX +COUNT
40A9- 20 EF 40 1550 LOOP/LOAD JSR READ/BYTE
40AC- C9 16 1560 CMP #516 SYNC
40AE- D0 04 1570 BNE SKIP/1
40B0- E6 B6 1580 INC +COUNT
40B2- D0 F5 1590 BNE LOOP/LOAD
1600 ;
40B4- A4 B6 1610 SKIP/1 LDY +COUNT
40B6- C0 0A 1620 CPY #50A MUST BE > = 10 SYNC'S
40B8- 90 EB 1630 BCC C/READ
40BA- C9 0F 1640 CMP #50F RECORD START
40BC- D0 E7 1650 BNE C/READ
1660 ;

```

```

40BE- A0 00      1670      LDY #S00
40C0- 84 B2      1680      STY +CHECKSUM
40C2- 84 B3      1681      STY +CHECKSUM+S01 CLEAR CHECKSUM LOCATIONS
40C4- 20 84 40   1690      JSR MOVE+ST/AD START > ADDR< (2)
1700 ;
1710 ;NOW LOAD DATA
40C7- 20 EF 40   1720 LOOP/69 JSR READ/BYTE
40CA- AC 23 01   1730      LDY LOAD/NO CKG. IF TO STORE
40CD- F0 02      1740      BEQ SKIP/STORE
40CF- 81 B6      1750      STA (ADDRS,X)
40D1- 20 8F 40   1760 SKIP/STORE JSR INC/COMP
40D4- 90 F1      1770      BCC LOOP/69
40D6- A5 B3      1771      LDA +CHECKSUM+S01
40D8- 48         1772      PHA SAVE CHSUM HI
40D9- A5 B2      1780      LDA +CHECKSUM
40DB- 48         1790      PHA SAVE CHECKSUM LO
40DC- 20 EF 40   1800      JSR READ/BYTE
40DF- 68         1810      PLA
40E0- C5 B4      1820      CMP +FORM+BYTE CHECK CHECKSUM LO
40E2- D0 07      1821      BNE RETURN
40E4- 20 EF 40   1822      JSR READ/BYTE
40E7- 68         1823      PLA
40E8- C5 B4      1824      CMP +FORM+BYTE CHECK CHECKSUM HI
40EA- 60         1830      RTS
40EB- 68         1831 RETURN PLA
40EC- A9 FF      1832      LDA #$FF CLEAR Z-BIT
40EE- 60         1833      RTS
1840 ;ON RETURN Z-BIT=TRUE:GOOD LOAD; Z-BIT==FALSE:ERROR
1850 ;
1860 ;
1870 ;ROUTINE READ A BYTE FROM TAPE
1880 ;
40EF- 20 2E 41   1890 READ/BYTE JSR IN/PORT
40F2- D0 FB      1900      BNE READ/BYTE LOOP UNTIL 0
1910 ;
40F4- 20 2E 41   1920 WAIT+FOR+1 JSR IN/PORT
40F7- F0 FB      1930      BEQ WAIT+FOR+1 LOOP UNTIL 1
1940 ;
40F9- 20 80 40   1950      JSR READ.DELAY
40FC- 20 2E 41   1960      JSR IN/PORT
40FF- F0 F3      1970      BEQ WAIT+FOR+1 IF ZERO
1980 ;
4101- 20 2E 41   1990 WAIT+FOR+0 JSR IN/PORT
4104- D0 FB      2000      BNE WAIT+FOR+0 WAIT TIL END OF START BIT
2010 ;
4106- A9 08      2020      LDA #S08
4108- 85 B5      2030      STA +BIT.COUNT
2040 ;
410A- 20 2E 41   2050 WAIT+TD+CN JSR IN/PORT
410D- F0 FB      2060      BEQ WAIT+TD+CN LOOP UNTIL '1'
410F- 20 80 40   2070      JSR READ.DELAY
4112- 20 2E 41   2080      JSR IN/PORT
4115- F0 08      2090      BEQ PROCESS+0 IF '0' THEN ZERO, ELSE ONE
4117- 20 2E 41   2110 PROCESS+1 JSR IN/PORT
411A- D0 FB      2120      BNE PROCESS+1 LOOP UNTIL '0'
411C- 38         2130      SEC
411D- B0 01      2140      BCS ROTATE+IN
411F- 18         2160 PROCESS+0 CLC

```

```

4120- 26 B4    2170 ROTATE+IN  ROL  +FORM+BYTE ROTATE CARRY
4122- C6 B5    2180          DEC  +BIT.COUNT
4124- D0 E4    2190          BNE  WAIT+TD+CM
4126- A5 B4    2200          LDA  +FORM+BYTE
4128- 20 34 41 2201          JSR  CKSUM+ADD UPDATE CHECKSUM
412B- A5 B4    2202          LDA  +FORM+BYTE
412D- 60      2210          RTS
                2220 ;
                2230 ;
                2240 ;INPUT FROM TAPE
                2250 ;
412E- AD 02 17 2260 IN/PORT  LDA  C/PORT
4131- 29 04    2270          AND  #%00000100      ;MASK OUT ALL BUT BIT 2
4133- 60      2280          RTS
                2281 ;
                2282 ;
                2283 ;
                2284 ;UPDATE CHECKSUM COUNTERS
                2285 ;
4134- 18      2286 CKSUM+ADD  CLC
4135- D8      2287          CLD
4136- 65 B2   2288          ADC  +CHECKSUM+$00 ADD R(A) TO CKSUM LD
4138- 85 B2   2289          STA  +CHECKSUM+$00
413A- A9 00   2290          LDA  #$00
413C- 65 B3   2291          ADC  +CHECKSUM+$01 ADD 00 TO CKSUM HI
413E- 85 B3   2292          STA  +CHECKSUM+$01
4140- 60      2293          RTS
                2294 ;
                2300 ;
                2310 ;
4141- 20 A5 40 2320 LOAD.ENTRY JSR  C/READ
4144- D0 08   2330          BNE  BAD
4146- A9 00   2340          LDA  #$00 INDICATE GOOD LOAD BY R(A)=00
4148- 00     2350 B          BRK
4149- EA     2360          NOP
414A- EA     2370          NOP
414B- 4C 41 41 2380          JMP  LOAD.ENTRY
414E- A9 EE   2390 BAD      LDA  #$EE INDICATE BAD LOAD BY R(A)=EE
4150- D0 F6   2400          BNE  B
                2410 ;
4152- 20 00 40 2420 RECORD.ENT JSR  C/WRITE
4155- 00     2430          BRK
4156- EA     2440          NOP
4157- EA     2450          NOP
4158- 4C 52 41 2460          JMP  RECORD.ENT
                2470 ;
                2480 END+OF+P6M .EN

```

LABEL FILE: [/ = EXTERNAL]

```

/C/PORT=1702
/COUNT=00B6
/BIT.COUNT=00B5
/ART=0124
LOOP/RECST=400C
WRITE/BYTE=4041

```

```

/C/PORTD=1703
/FORM+BYTE=00B4
/ADRS=00B6
/END=0126
LOOP/DELSY=4015
DATA/LOOP=404D

```

```

/CHECKSUM=00B2
/SYNC+COUNT=00B4
/LOAD/NO=0123
C/WRITE=4000
LOOP/DATA=402E
ONE.BIT=4051

```

ZERO.BIT=4056
WRITE=405F
OUT:ONE=407C
INC/COMP=408F
C/READ=40A5
LOOP/69=40C7
READ/BYTE=40EF
WAIT+TD+CN=410A
ROTATE+IN=4120
LOAD.ENTRY=4141
RECORD.ENT=4152
//0000,415B,415B
>

CK+END+BY=4059
X=4076
READ.DELAY=4080
SKIP/INC=4095
LOOP/LOAD=40A9
SKIP/STORE=40D1
WAIT+FOR+1=40F4
PROCESS+1=4117
IN/PORT=412E
B=4148
END+DF+PGM=415B

OUT:ZERO=405D
LOOPD=4078
MOVE+ST/AD=4084
NOT/END=40A4
SKIP/1=40B4
RETURN=40EB
WAIT+FOR+0=4101
PROCESS+0=411F
CKSUM+ADD=4134
BAD=414E