

SYM-1 USERS' GROUP
P. O. BOX 315
CHICO, CA 95927
(916) 895-8751

RAE NOTES - ISSUE NO. 2

So many people have written and called asking about the disk linkages in RAE that we decided to discuss this topic even before publishing the list of page zero and page one assignments.

Meanwhile, here is a list of six "absolutely safe" page zero addresses which are never used by RAE: \$CA, \$CB, \$CC, \$E8, \$EB, \$DE.

Until you add a disk system, all five disk vectors from \$EC through \$F7 (but not the two flags at \$EE and \$EF) are safe, unless you use the .CT patch described in Section 10.0 of the RAE-1 Reference Manual. \$B7 and \$B8 are also "safe", unless you use them for a printer vector as part of the HARDCOPY command. More on this in the next issue of RAE Notes. Our approach to the disk system integration leaves \$EE, \$EF, and \$F4-\$F7 unused, but the .CT patch uses \$EE, \$F6, and \$F7.

RAE-1 has three specific disk commands. These are >DC (for Disk Command), >EN (for ENTER), and >LO (for LOad, or LOokup). These are vectored through the six page zero bytes at \$EC/\$ED, \$F0/\$F1, and \$F2/\$F3, respectively. Try filling these bytes with USRENT (\$8035) before entering RAE-1 at the cold start, and see what happens when you issue the commands >DC, >EN, and >LO. The use of these three vectors is illustrated in the RAE/FODS patch listed below. >EN and >LO support named files, and >DC provides access to all of the commands supported by the disk system.

An alternate approach, not used by us, is provided by two additional vectors at \$F4/\$F5 and \$F6/\$F7, and a pair of flags at \$EE and \$EF. \$EE and \$EF are initialized to 0 by RAE. If these are set to 1 by a patch (or in any fashion), >PU and >GE will be vectored through \$F4/\$F5, and \$F6/\$F7 respectively; your disk vectors should be placed in these positions. Note that the flag at \$EE corresponds with the vector at \$F6/\$F7 while the flag at \$EF corresponds to the vector at \$F4/\$F5. We obviously prefer the first approach, and until we can implement a .CD (Continue on Disk) pseudo-op, we need \$F6/\$F7 to correct the .CT bus in RAE-1!

Label File continued from Page 6

/DSKRW=767A	/CMDINT=79CE	/USRENT=8035
/NIBASC=8309	/ACCESS=8B86	/UBRKVC=A676
/RAE.COLD=B000	/RAE.WARM=B003	/ERROR=B00E
/RAE.HOT=B05E	INIT=6E00	WARM=6E03
COLD=6E09	ENTER=6E0F	EN1=6E18
EN2=6E1C	EN3=6E27	EN4=6E36
LOAD=6E3F	LO1=6E48	LO2=6E4C
LO3=6E57	LO4=6E64	LO5=6E67
LO6=6E86	DISK.CMD=6E8E	DC1=6E97
DC2=6E99	DC3=6EB4	^C.CHK=6EB7
SET.VEC=6EC5	RST.DSK=6EEE	HEX.ASC=6EF7
NAM.CMD=6FOC	MOV.NAM=6F17	RAE.STR=6F2C
RAE.END=6F39	SAVE=6F44	LOD=6F4A

//0000,6F50,6F50

```

0010
0020 ; RAE-FODS linkage program
0030 ; written by Thomas Gettys
0040
0050
0060 ; (.DS goes here)
0070         .BA $6E00
0080
0090 ; PAGE 0 AND 1 USAGE
0100
0110 TPRES         .DE $D3           !PRESENT END OF RAE TEXT FILE
0120 EOT          .DE $A8           !END OF FODS FILE
0130 DISCC.VEC    .DE $EC           !DISK COMMAND VECTOR
0140 DISCI        .DE $EE           !RAE TAPE/DISK INPUT FLAG
0150 DISCO        .DE $EF           !RAE TAPE/DISK OUTPUT FLAG
0160 DISC1        .DE $F0           !RAE DISK OUTPUT VECTOR
0170 DISC2        .DE $F2           !RAE DISK INPUT VECTOR
0180 TXST         .DE $100          !START OF RAE FILE ADDRESS
0190 TXEN         .DE $102          !END OF RAE FILE ADDRESS
0200 CRT          .DE $135          !RAE INPUT BUFFER
0210
0220 ; FODS ADDRESSES
0230
0240 TXBUF        .DE $7280         !FODS TEXT BUFFER
0250 SET2         .DE $7651         !SELECT DRIVE 2
0260 DSKRW        .DE $767A         !DRIVE CONTROL ROUTINE
0270 CMDINT       .DE $79CE         !FODS COMMAND INTERPRETER
0280
0290 ; SUPERMON ADDRESSES
0300
0310 USRENT       .DE $8035         !PSEUDO-INTERRUPT ENTRY
0320 NIBASC       .DE $8309         !CONVERT NIBBLE IN A TO ASCII
0330 ACCESS       .DE $8B86         !UNWRITE PROTECT SYS. RAM
0340
0350 ; SYM SYSTEM RAM USAGE
0360
0370 UBRKVC       .DE $A676         !USER BREAK VECTOR
0380
0390 ; RAE ADDRESSES
0400
0410 RAE.COLD     .DE $B000         !RAE COLD START
0420 RAE.WARM     .DE $B003         !RAE WARM START
0430 ERROR        .DE $B00E         !RAE ERROR ROUTINE VECTOR
0440 RAE.HOT      .DE $B05E         !RAE WARM W/NO MESSAGE
0450
0460
0470
0480 ; INIT is the cold entry point, and WARM (INIT+3) is th
0490 ; warm entry point. The only reason this is set up thi
0500 ; way is because it is easy to remember.
0510
0520 INIT         JMP COLD           !THIS IS THE COLD ENTRY POINT
0530 WARM         JSR SET.VEC        !THIS IS THE WARM ENTRY POINT
0540
0550
0560 COLD         JSR SET.VEC
0570
06E00- 4C 09 6E 0520 INIT         JMP COLD           !THIS IS THE COLD ENTRY POINT
06E03- 20 C5 6E 0530 WARM         JSR SET.VEC        !THIS IS THE WARM ENTRY POINT
06E06- 4C 03 B0 0540
06E09- 20 C5 6E 0560 COLD         JSR SET.VEC
06E0C- 4C 00 B0 0570

```

```

0580
0590
0600 ; ENTER is the entry point on an ENTER command. All RAE
0610 ; disk files are prefaced with an ':'. An error code of
0620 ; 30 means no filename was supplied.
0630
6E0F- C0 50 0640 ENTER      CPY ##50      !IF Y=50 FILENAME IS MISSING
6E11- D0 05 0650          BNE EN1       !Y POINTS AT FILENAME
6E13- A2 30 0660          LDX ##30     !ERR 30=NO FILENAME
6E15- 6C 0E B0 0670          JMP (ERROR) !PRINT ERROR MESSAGE
0680
6E18- A2 00 0690 EN1       LDX #0
6E1A- 86 EF 0700          STX *DISCO   !RE-ENABLE TAPE OUTPUT
0710
0720 ; Build command string: ENT $s.adr$e.adr=:name
0730 ; Put it in FODS buffer and let FODS do the work.
0740
6E1C- BD 44 6F 0750 EN2     LDA SAVE,X   !GET NEXT CMD CHARACTER
6E1F- F0 06 0760          BEQ EN3     !ARE WE DONE?
6E21- 9D 80 72 0770          STA TXBUF,X !MOVE TO FODS BUFFER
6E24- E8 0780          INX
6E25- D0 F5 0790          BNE EN2
6E27- 20 2C 6F 0800 EN3     JSR RAE.STR  !TELL FODS WHERE FILE STARTS
6E2A- A9 24 0810          LDA #'$
6E2C- 9D 80 72 0820          STA TXBUF,X
6E2F- E8 0830          INX
6E30- 20 39 6F 0840          JSR RAE.END !TELL FODS WHERE FILE ENDS
6E33- 20 0C 6F 0850          JSR NAM.CMD !GO PUT NAME IN AND DO CMD
0860
6E36- 20 EE 6E 0870 EN4     JSR RST.DSK
6E39- A2 FF 0880          LDX ##FF   !RESET STACK IN CASE WE
6E3B- 9A 0890          TXS       !GOT A BRK FROM FODS
6E3C- 4C 5E B0 0900          JMP RAE.HOT !AND GO BACK TO RAE
0910
0920
0930 ; LOAD is the entry point on a LOAD command. The file
0940 ; specified is down-loaded into the current text file.
0950 ; If the file exceeds the current text file boundry an
0960 ; error 31 results. The entire file has been brought
0970 ; in, but the upper limit should be set to accommodate
0980 ; the whole file before preceding. If the filename is
0990 ; preceded with a '+' the file will be appended to the
1000 ; current file. If no filename is specified, an error
1010 ; 30 results.
1020
6E3F- C0 50 1030 LOAD      CPY ##50
6E41- D0 05 1040          BNE L01     !Y POINTS AT FILENAME
6E43- A2 30 1050          LDX ##30     !ERR 30=NO FILENAME
6E45- 6C 0E B0 1060          JMP (ERROR) !PRINT ERROR MESSAGE
1070
6E48- A2 00 1080 L01      LDX #0
6E4A- 86 EE 1090          STX *DISCI   !RE-ENABLE TAPE INPUT
1100
1110 ; Build command string: LOD $s.adr=:name
1120 ; Put it in FODS buffer and let FODS do the work.
1130
6E4C- BD 4A 6F 1140 L02     LDA LOD,X   !GET NEXT CMD CHARACTER
6E4F- F0 06 1150          BEQ L03     !ARE WE DONE?

```

```

6E51- 9D 80 72 1160 STA TXBUF,X !MOVE TO FODS BUFFER
6E54- E8 1170 INX
6E55- D0 F5 1180 BNE LO2
6E57- B9 35 01 1190 LO3 LDA CRT,Y !GET 1ST CHARACTER IN NAME S...IN
6E5A- C9 2B 1200 CMP #'+' !IS IT TO BE APPENDED?
6E5C- D0 06 1210 BNE LO4 !IF NOT, SKIP
6E5E- C8 1220 INY !BYPASS THE '+'
6E5F- 20 39 6F 1230 JSR RAE.END !TELL FODS WHERE ATTACH POINT IS
6E62- D0 03 1240 BNE LO5
6E64- 20 2C 6F 1250 LO4 JSR RAE.STR !TELL FODS WHERE TO BEGIN FILE
6E67- 20 0C 6F 1260 LO5 JSR NAM.CMD !GO PUT IN NAME AND DO CMD
1270
1280 ; Set EOT pointer and test for overflow
1290
6E6A- A9 00 1300 LDA #0
6E6C- A0 02 1310 LDY #2
6E6E- 91 A8 1320 STA (EOT),Y !FLAG EOT FOR RAE
6E70- A5 A8 1330 LDA *EOT !TELL RAE
6E72- 85 D3 1340 STA *TPRES !WHERE THE
6E74- A5 A9 1350 LDA *EOT+1 !END OF THE
6E76- 85 D4 1360 STA *TPRES+1 !FILE IS
1370
6E78- CD 03 01 1380 CMP TXEN+1 !TEST FOR
6E7B- 90 B9 1390 BCC EN4 !TEXT FILE
6E7D- D0 07 1400 BNE LO6 !OVERFLOW
6E7F- AD 02 01 1410 LDA TXEN !HIGH BYTES EQUAL,
6E82- C5 D3 1420 CMP *TPRES !SO MUST TEST
6E84- B0 B0 1430 BCS EN4 !LOW BYTES
1440
6E86- 20 EE 6E 1450 LO6 JSR RST.DSK
6E89- A2 31 1460 LDX ##31 !ERR 31=OVERFLOW OF THE
6EBB- 6C 0E B0 1470 JMP (ERROR) !TEXT FILE ON DISK LOAD
1480
1490
1500
1510 ; DISK.CMD is the entry point on a DC command. If no
1520 ; command string is supplied an error 32 results.
1530
6E8E- C0 50 1540 DISK.CMD CPY ##50
6E90- D0 05 1550 BNE DC1 !Y POINTS AT COMMAND STRING
6E92- A2 32 1560 LDX ##32 !ERR 32=NO COMMAND STRING
6E94- 6C 0E B0 1570 JMP (ERROR) !PRINT ERROR MESSAGE
1580
6E97- A2 00 1590 DC1 LDX #0
6E99- B9 35 01 1600 DC2 LDA CRT,Y !GET NEXT CMD CHARACTER
6E9C- 9D 80 72 1610 STA TXBUF,X !PUT IT IN FODS BUFFER
6E9F- E8 1620 INX
6EA0- C8 1630 INY
6EA1- C9 20 1640 CMP ##20 !IS IT A BLANK?
6EA3- D0 F4 1650 BNE DC2 !NO, KEEP GOING
6EA5- B9 35 01 1660 LDA CRT,Y !GET CHARACTER AFTER SPACE
6EA8- C9 20 1670 CMP ##20 !ANOTHER ONE?
6EAA- D0 ED 1680 BNE DC2 !NOPE, STILL MORE
6EAC- A9 0D 1690 LDA ##D !PUT IN THE <CR>
6EAE- 9D 7F 72 1700 STA TXBUF-1,X
6EB1- 20 CE 79 1710 JSR CMDINT !LET FODS DEAL WITH IT
6EB4- 4C 36 6E 1720 DC3 JMP EN4 !GO FINISH UP
1730

```

```

1740
1750 ;  ^C.CHK is entered whenever a BRK instruction occurs.
1760 ;  If the break was due to a CTRL-C in RAE we exit to
1770 ;  SUPERMON, otherwise it must have come from FODS, so
1780 ;  we stay in RAE.
1790
6EB7- BA      1800 ^C.CHK      TSX
6EB8- BD 03 01 1810      LDA $103,X      !GET PC,HIGH
6EBB- C9 B0      1820      CMP ##B0        !BRK FROM RAE?
6EBD- D0 F5      1830      BNE DC3        !NO, SO STAY IN RAE
6EBF- 20 35 80  1840      JSR USRENT     !ELSE GOTO SUPERMON
6EC2- 4C 03 6E  1850      JMP WARM       !DO A WARM START
1860
1870
1880 ;  *** SUPPORT SUBROUTINES ***
1890
1900
1910 ;  SETUP ALL NECESSARY VECTORS FOR FODS AND RAE
1920
6EC5- 20 EE 6E  1930 SET.VEC      JSR RST.DSK
6EC8- A9 0F      1940      LDA #L,ENTER
6ECA- 85 F0      1950      STA *DISC1
6ECC- A9 6E      1960      LDA #H,ENTER
6ECE- 85 F1      1970      STA *DISC1+1
6ED0- A9 3F      1980      LDA #L,LOAD
6ED2- 85 F2      1990      STA *DISC2
6ED4- A9 6E      2000      LDA #H,LOAD
6ED6- 85 F3      2010      STA *DISC2+1
6ED8- A9 8E      2020      LDA #L,DISK.CMD
6EDA- 85 EC      2030      STA *DISCC.VEC
6EDC- A9 6E      2040      LDA #H,DISK.CMD
6EDE- 85 ED      2050      STA *DISCC.VEC+1
6EE0- 20 86 8B  2060      JSR ACCESS
6EE3- A9 B7      2070      LDA #L,^C.CHK
6EE5- 8D 76 A6  2080      STA UBRKVC
6EE8- A9 6E      2090      LDA #H,^C.CHK
6EEA- 8D 77 A6  2100      STA UBRKVC+1
6EED- 60        2110      RTS
2120
2130
2140 ;  RESET DRIVES OFF AND DEFAULT TO DRIVE 2
2150
6EEE- A9 10      2160 RST.DSK      LDA ##10        !DRIVES OFF CODE
6EF0- 20 7A 76  2170      JSR DSKRW     !GO TURN 'EM OFF
6EF3- 20 51 76  2180      JSR SET2     !DEFAULT IS DRIVE 2
6EF6- 60        2190      RTS
2200
2210
2220 ;  CONVERT A TO 2 HEX DIGITS & PUT 'EM IN TXBUF
2230
6EF7- 48        2240 HEX.ASC      PHA
6EF8- 4A        2250      LSR A
6EF9- 4A        2260      LSR A
6EFA- 4A        2270      LSR A
6EFB- 4A        2280      LSR A
6EFC- 20 09 83  2290      JSR NIBASC
6EFF- 9D 80 72  2300      STA TXBUF,X
6F02- E8        2310      INX

```

```

6F03- 68          2320          PLA
6F04- 20 09 83   2330          JSR NIBASC
6F07- 9D 80 72   2340          STA TXBUF,X
6F0A- E8          2350          INX
6F0B- 60          2360          RTS
                2370
                2380
                2390 ; PUT NAME IN FODS BUFFER & PASS COMMAND TO FODS
                2400
6F0C- A9 3D      2410 NAM.CMD    LDA #'=
6F0E- 9D 80 72   2420          STA TXBUF,X
6F11- E8          2430          INX
6F12- A9 3A      2440          LDA #' :      !ALL RAE FILES BEGIN WITH ':'
6F14- 9D 80 72   2450          STA TXBUF,X
6F17- B9 35 01   2460 MOV.NAM    LDA CRT,Y      !MOVE FILENAME INTO FODS BUFFER
6F1A- E8          2470          INX
6F1B- C8          2480          INY
6F1C- 9D 80 72   2490          STA TXBUF,X
6F1F- C9 20      2500          CMP ##20
6F21- D0 F4      2510          BNE MOV.NAM
6F23- A9 0D      2520          LDA ##D
6F25- 9D 80 72   2530          STA TXBUF,X  !PUT IN A <CR>
6F28- 20 CE 79   2540          JSR CMDINT    !AND LET FODS FIGURE IT OUT
6F2B- 60          2550          RTS
                2560
                2570
                2580 ; TELL FODS WHERE RAE TEXT AREA BEGINS
                2590
6F2C- AD 01 01   2600 RAE.STR    LDA TXST+1     !CONVERT RAE
6F2F- 20 F7 6E   2610          JSR HEX.ASC   !START ADDRESS
6F32- AD 00 01   2620          LDA TXST      !TO ASCII
6F35- 20 F7 6E   2630          JSR HEX.ASC   !AND PUT IN FODS BUFFER
6F38- 60          2640          RTS
                2650
                2660
                2670 ; TELL FODS WHERE RAE TEXT AREA ENDS
                2680
6F39- A5 D4      2690 RAE.END    LDA *TPRES+1  !CONVERT RAE
6F3B- 20 F7 6E   2700          JSR HEX.ASC   !END ADDRESS
6F3E- A5 D3      2710          LDA *TPRES    !TO ASCII
6F40- 20 F7 6E   2720          JSR HEX.ASC   !AND PUT IN FODS BUFFER
6F43- 60          2730          RTS
                2740
6F44- 45 4E 54   2750 SAVE      .BY 'ENT #' 0
6F47- 20 24 00   2760 LOD        .BY 'LOD #' 0
6F4A- 4C 4F 44   2770
6F4D- 20 24 00   2780          .EN

```

LABEL FILE: [/ = EXTERNAL]

```

/TPRES=00D3          /EOT=00AB          /DISCC.VEC=00EC
/DISCI=00EE         /DISCO=00EF        /DISC1=00F0
/DISC2=00F2         /TXST=0100         /TXEN=0102
/CRT=0135           /TXBUF=7280        /SET2=7651

```

```
0005 ;USER PATCH FOR RAE-1.0      J.CYR,MAR1980
0010 ;
0015 ;FUNCTIONS:
0020 ;-CLEANER LOOKING EDIT ON TERMINALS THAT RECOGNIZE
0025 ; DEL(RUBOUT). TRANSLATE DEL TO BS ON INPUT, AND
0030 ; BS TO DEL ON OUTPUT.
0035 ;-HARDCOPY. PROVIDE INTERFACE TO OUTPUT ONLY TTY.
0040 ;-READABILITY. ECHO INPUT CHARACTERS TO HARDCOPY
0045 ; DEVICE WHEN .HA IS SET (RAE DOES NOT!).
0050 ;-USEFULNESS. PROVIDE USER COMMAND TO SORT LABEL
0055 ; FILE ALPHABETICALLY.
0060 ;-USEFULNESS. SET RAE BOUNDS TO BETTER USE AVAILABLE
0065 ; MEMORY.
0070 ;
0075 ;ASSEMBLY PARAMETER(S)
0080 ;
0085 LBSIZ      .DE $500      ;LABEL FILE SIZE
0090 ;CALCULATED PARAMETER(S)
0095 BUFADR     .DE INITIALIZE-$100 ;ADDR OF BUFFER
0100 LBLHADR    .DE BUFADR-4     ;END OF LBL FILE
0105 LBLADR     .DE LBLHADR-LBSIZ+4 START OF LBL FILE
0110 TXTHADR    .DE LBLADR-4     ;END OF TEXT FILE
0115 ;
0120 ;MON-1.1 VARIABLE(S)
0125 ;
0130 TOUTFL     .DE $A654      ;TERMINAL OUTPUT FLAG
0135 CRTI       .DE %10000000 CRT IN BIT
0140 TTYI       .DE %01000000 TTY IN BIT
0145 TTYO       .DE %00100000 TTY OUT BIT
0150 CRTO       .DE %00010000 CRT OUT BIT
0155 ;
0160 SDBYT      .DE $A651      ;SPEED BYTE
0165 B110      .DE $D5        ;110 BAUD CONSTANT
0170 B300      .DE $4C        ;300 BAUD CONSTANT
0175 ;
0180 ;RAE-1.0 VARIABLE(S)
0185 ;
0190 HAFLAG     .DE $011F      ;HARDCOPY FLAG
0195 LBLHIGH    .DE $0106      ;END OF LABEL FILE
0200 LBL        .DE $0104      ;LABEL FILE ADDRESS
0205 TXTHIGH    .DE $0102      ;END OF TEXT FILE
0210 BUF        .DE $00C8      ;BUFFER ADDRESS
0215 ;
0220 ;MON-1.1 ROUTINE(S)
0225 ;
0230 TOUT        .DE $8AA0      ;OUTPUT BYTE TO TERMINAL
0235 INTCHR     .DE $8A58      ;INPUT BYTE FROM TERMINAL
0240 ;
0245 ;RAE-1.0 ROUTINE(S)
0250 ;
0255 RAEWARM    .DE $B003      ;RAE WARM ENTRY POINT
0260 ;
0265 ;MON-1.1 LINK(S)
0270 ;
0275 OUTVEC     .DE $A663      ;TERMINAL OUTPUT LINK
0280 INVEC      .DE $A660      ;TERMINAL INPUT LINK
0285 ;
```

```

0290 ;RAE-1.0 LINK(S)
0295 ;
0300 PRIVEC      .DE $00B6      ;PRINTER IO LINK
0305 USERVEC    .DE $0003      ;USER COMMAND LINK
0310 ;
0315 ;PAGE ZERO SCRATCH STORAGE (USED BY SORT)
0320 ;
0325 SCRN        .DE $FE        ;ADDRESS OF NEXT LABEL
0330 SCRC        .DE $FC        ;ADDRESS OF CURRENT LABEL
0335 ;
0340 ;DUMMY VARIABLE FOR MACRO EXPANSION
0345 ;
0350 DUMMY       .DE 0
0355 ;
0360 ;CONTROL CHAR(S)
0365 ;
0370 DEL         .DE $7F        ;DEL(RUBOUT) CHARACTER
0375 BS         .DE $08        ;BACKSPACE CHARACTER
0380 ;
0385 ;MACRO DEFINITION(S)
0390 ;
0395 !!!SL      .MD (ROUTINE LINK) ;SET LINK
0400             LDA #ROUTINE
0405             STORE (LINK+1)
0410             LDA #H,ROUTINE
0415             STORE (LINK+2)
0420             .ME
0425 ;
0430 !!!SB      .MD (OLDCHAR NEWCHAR) ;SUBSTITUTE CHARAC
0435             CMP #OLDCHAR
0440             BNE =+3
0445             LDA #NEWCHAR
0450             .ME
0455 ;
0460 !!!MW      .MD (FROM TO) ;MOVE WORD
0465             LOAD (FROM)
0470             STORE (TO)
0475             LOAD (FROM+1)
0480             STORE (TO+1)
0485             .ME
0490 ;
0495 !!!MT      .MD (FROM TO) ;MOVE TEXT
0500             LDY #0
0505 ...MT1     LDA (FROM),Y
0510             STA (TO),Y
0515             BMI ...MT3
0520 ...MT2     INY
0525             BNE ...MT1
0530 ...MT3     CPY #2
0535             BCC ...MT2
0540             .ME
0545 ;
0550 !!!STORE    .MD (ADR)      ;STORE ACCUMULATOR
0555             SET DUMMY = ADR
0560             IFM DUMMY
0565             SET DUMMY = $100
0570             ***
0575             IFP $FF-DUMMY

```



```

0580      STA *ADR
0585      ***
0590      IFP DUMMY-$100
0595      STA ADR
0600      ***
0605      .ME
0610 ;
0615 !!!LOAD      .MD (ADR)      ;LOAD ACCUMULATOR
0620      SET DUMMY=ADR
0625      IFM DUMMY
0630      SET DUMMY=$100
0635      ***
0640      IFP SFF-DUMMY
0645      LDA *ADR
0650      ***
0655      IFP DUMMY-$100
0660      LDA ADR
0665      ***
0670      .ME
0675 ;
0680 ;SET ASM
0685 ;
0690      .EC
0695      .BA $1EE8
0700      .OS
0705 ;
0710 ;INITIALIZE PATCH
0715 ;
1EER- A9 4C      0720 INITIALIZE LDA #$4C      ;JMP INSTR.
1EEA- 85 B6      0725      STA *PRIVEC ;INIT PRINTER LINK
1EEC- 85 03      0730      STA *USERVEC INIT USER LINK
0735      SL (TTYOUT PRIVEC);SET PRINTER LINK
0740      SL (CRTIN INVEC) SET TERMINAL INPUT LINK
0745      SL (CRTOUT OUTVEC) SET TERMINAL OUTPUT LINK
0750      SL (SORTLBS USERVEC) SET USER COMMAND LINK
0755      SL (BUFADR BUF-1) SET RAE BOUNDS
0760      SL (LBLHADR LBLHIGH-1)
0765      SL (LBLADR LBL-1)
0770      SL (TXTHADR TXTHIGH-1)
1F3R- A9 90      0775      LDA #CRTI+CRTO SET FOR TERMINAL ONLY
1F3A- 8D 54 A6   0780      STA TOUTFL
1F3D- 4C 03 B0   0785 USEREXIT  JMP RAEWARM ;RETURN TO RAE-1.0
0790 ;
0795 ;OUTPUT TO TTY
0800 ;
1F40- 48          0805 TTYOUT   PHA ;      SAVE ACCUMULATOR
1F41- A9 20      0810      LDA #TTY0   ;SET FOR TTY OUTPUT
1F43- 8D 54 A6   0815      STA TOUTFL
1F46- A9 D5      0820      LDA #B110   ;SET AT 110 BAUD
1F48- 8D 51 A6   0825      STA SDBYT
1F4B- 68          0830      PLA ;      RESTORE ACC
1F4C- 48          0835      PHA
1F4D- 20 A0 8A   0840      JSR TOUT   ;OUTPUT CHARACTER
1F50- A9 90      0845      LDA #CRTI+CRTO SET FOR TERMINAL I/O
1F52- 8D 54 A6   0850      STA TOUTFL
1F55- A9 4C      0855      LDA #B300   ;SET FOR 300 BAUD
1F57- 8D 51 A6   0860      STA SDBYT
1F5A- 68          0865      PLA ;      RESTORE ACCUMULATOR

```

```

1F5B- 60      0870      RTS ; RETURN TO RAE-1.0
              0875 ;
              0880 ; INPUT FROM KEYBOARD
              0885 ;
1F5C- 20 58 8A 0890 CRTIN JSR INTCHR ; GET CHARACTER
              0895 SB (DEL BS) ; SUBSTITUTE BS FOR DELETE
1F65- AE 1F 01 0900 LDX HAFLAG ; TEST HARDCOPY OPTION
1F68- D0 D6 0905 BNE TTYOUT ; OUTPUT IF SET
1F6A- 60 0910 RTS ; RETURN TO RAE-1.0
              0915 ;
              0920 ; OUTPUT TO SCREEN
              0925 ;
              0930 CRTOUT SB (BS DEL) ; SUBSTITUTE DEL FOR BS
1F71- 4C A0 8A 0935 JMP TOUT ; OUTPUT AND RETURN TO RAE-1.0
              0940 ;
              0945 ; EXCHANGE SORT LABEL FILE
              0950 ;
              0955 SORTLBLS MW (LBL SCRN) SET NEXT TO 1ST LABEL
              0960 NEXTLBL MW (SCRN SCRC) NEXT BECOMES CURRENT
1F86- A0 02 0965 LDY #2 ; FIND NEXT LABEL
1F88- B1 FC 0970 NEXTCHAR LDA (SCRC),Y
1F8A- 30 03 0975 BMI COMPSTRING
1F8C- C8 0980 INY
1F8D- D0 F9 0985 BNE NEXTCHAR
1F8F- 20 F1 1F 0990 COMPSTRING JSR ADRNEXT ; COMPARE LABEL IF PRESENT
1F92- A0 02 0995 LDY #2
1F94- B1 FE 1000 LDA (SCRN),Y
1F96- F0 A5 1005 BEQ USEREXIT EXIT IF END OF FILE
1F98- B1 FC 1010 COMPCHAR LDA (SCRC),Y TEST FOR LAST CHAR
1F9A- 51 FE 1015 EOR (SCRN),Y
1F9C- 30 0B 1020 BMI EOS ; TO END OF STRING IF SUCH
1F9E- B1 FE 1025 LDA (SCRN),Y COMPARE CHARACTERS
1FA0- D1 FC 1030 CMP (SCRC),Y
1FA2- 90 1B 1035 BCC XCHANGE ; EXCHANGE LABELS IF BACKWARDS
1FA4- D0 D8 1040 BNE NEXTLBL ; TO NEXT LABEL IF FORWARDS
1FA6- C8 1045 INY ; TO NEXT CHARACTER IF EQUAL
1FA7- D0 EF 1050 BNE COMPCHAR
1FA9- B1 FE 1055 EOS LDA (SCRN),Y SPECIAL CASE CODE FOR
1FAB- 10 0A 1060 BPL EOSC ; END OF STRING
1FAD- 29 7F 1065 AND #$7F
1FAF- D1 FC 1070 CMP (SCRC),Y WHEN STRINGS EQUAL
1FB1- F0 0C 1075 BEQ XCHANGE ; CONSIDER SHORT LOW
1FB3- 90 0A 1080 HIGHLOW BCC XCHANGE
1FB5- B0 C7 1085 BCS NEXTLBL
1FB7- 09 80 1090 EOSC ORA #$80
1FB9- D1 FC 1095 CMP (SCRC),Y
1FBB- F0 C1 1100 BEQ NEXTLBL
1FBD- D0 F4 1105 BNE HIGHLOW
              1110 XCHANGE MT (SCRC BUF) ; COPY CURRENT LABEL TO BUFFB
              1115 MT (SCRN SCRC) COPY NEXT TO CURRENT
1FDD- 20 F1 1F 1120 JSR ADRNEXT
              1125 MT (BUF SCRN) ; COPY BUFFER TO NEXT
1FEF- B0 83 1130 BCS SORTLBLS START OVER
              1135 ;
1FF1- 98 1140 ADRNEXT TYA ; CALCULATE ADDRESS OF
1FF2- 38 1145 SEC ; NEXT LABEL
1FF3- 65 FC 1150 ADC *SCRC
1FF5- 85 FE 1155 STA *SCRN ; STORE INTO SCRN

```

```

1FF7- A5 FD      1160      LDA *SCRC+1
1FF9- 69 00      1165      ADC #0.
1FFB- 85 FF      1170      STA *SCRN+1
1FFD- 60         1175      RTS ;
                  1180      RETURN
                  1185      .EN

```

LABEL FILE: [/ = EXTERNAL]

```

/LBLSIZ=0500      /BUFADR=1DE8      /LBLHADR=1DE4
/LBLLADR=18E8     /TXTHADR=18E4     /TOUTFL=A654
/CRTI=0080        /TTYI=0040        /ITYO=0020
/CRTO=0010        /SDBYT=A651       /B110=00D5
/B300=004C        /HAFLAG=011F     /LBLHIGH=0106
/LBL=0104         /TXTHIGH=0102    /BUF=00C8
/TOUT=8AA0        /INTCHR=8A58     /RAEWARM=B003
/OUTVEC=A663      /INVEC=A660      /PRTVEC=00B6
/USERVEC=0003     /SCRN=00FE       /SCRC=00FC
/DUMMY=00FD       /DEL=007F        /BS=0008
INITIALIZE=1EE8   ROUTINE=18E4     LINK=0101
ADR=00FD          USEREXIT=1F3D   TTYOUT=1F40
CRTIN=1F5C        OLDCHAR=0008     NEWCHAR=007F
CRTOUT=1F6B       SORTLBS=1F74    FROM=00C8
TO=00FE           NEXTLBL=1F7E     NEXTCHAR=1F88
COMPSTRING=1F8F   COMPCHAR=1F98   EOS=1FA9
HIGHLOW=1FB3     EO SC=1FB7      XCHANGE=1FBF
ADRNEXT=1FF1
//0000,1FFE,1FFE
>RUN INITIALIZE

```

```

0200-18E4 18E8-1DE4 1DE8
15F2 1B08
//

```

>USER

```

0200-18E4 18E8-1DE4 1DE8
15F2 1B08
//

```

>LABELS

LABEL FILE: [/ = EXTERNAL]

```

/B110=00D5      /B300=004C      /BS=0008
/BUF=00C8        /BUFADR=1DE8     /CRTI=0080
/CRTO=0010        /DEL=007F        /DUMMY=00FD
/HAFLAG=011F     /INTCHR=8A58     /INVEC=A660
/LBL=0104         /LBLHADR=1DE4    /LBLHIGH=0106
/LBLLADR=18E8     /LBLSIZ=0500    /OUTVEC=A663
/PRTVEC=00B6     /RAEWARM=B003   /SCRC=00FC
/SCRN=00FE       /SDBYT=A651     /TOUT=8AA0
/TOUTFL=A654     /TTYI=0040      /ITYO=0020

```

/TXTHADR=18E4
 ADR=00FD
 COMPSTRING=1F8F
 EOS=1FA9
 HIGHLOW=1FB3
 NEWCHAR=007F
 OLDCHAR=0008
 TO=00FE
 XCHANGE=1FBF
 //0000,1FFE,1FFE
 >HARDCOPY CLEAR
 P

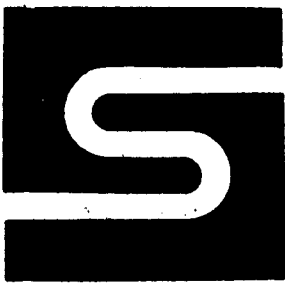
/TXTHIGH=0102
 ADRNEXT=1FF1
 CRTIN=1F5C
 EOSC=1FB7
 INITIALIZE=1EE8
 NEXTCHAR=1F88
 ROUTINE=18E4
 TTYOUT=1F40

/USERVEC=0003
 COMPCHAR=1F98
 CRTOUT=1F6B
 FROM=00C8
 LINK=0101
 NEXTLBL=1F7E
 SORTLBS=1F74
 USEREXIT=1F3D

.V 1EE8,1FFF

1EE8 A9 4C 85 B6 85 03 A9 40,A1
 1EF0 85 B7 A9 1F 85 B8 A9 5C,E7
 1EF8 8D 61 A6 A9 1F 8D 62 A6,D8
 1F00 A9 6B 8D 64 A6 A9 1F 8D,D8
 1F08 65 A6 A9 74 85 04 A9 1F,51
 1F10 85 05 A9 E8 85 C8 A9 1D,7F
 1F18 85 C9 A9 E4 8D 06 01 A9,97
 1F20 1D 8D 07 01 A9 E8 8D 04,6B
 1F28 01 A9 18 8D 05 01 A9 E4,4D
 1F30 8D 02 01 A9 18 8D 03 01,2F
 1F38 A9 90 8D 54 A6 4C 03 B0,EE
 1F40 48 A9 20 8D 54 A6 A9 D5,04
 1F48 8D 51 A6 68 48 20 A0 8A,82
 1F50 A9 90 8D 54 A6 A9 4C 8D,C4
 1F58 51 A6 68 60 20 58 8A C9,4E
 1F60 7F D0 02 A9 08 AE 1F 01,1E
 1F68 D0 D6 60 C9 08 D0 02 A9,70
 1F70 7F 4C A0 8A AD 04 01 85,9C
 1F78 FE AD 05 01 85 FF A5 FE,74
 1F80 85 FC A5 FF 85 FD A0 02,BD
 1F88 B1 FC 30 03 C8 D0 F9 20,4E
 1F90 F1 1F A0 02 B1 FE F0 A5,44
 1F98 B1 FC 51 FE 30 0B B1 FE,2A
 1FA0 D1 FC 90 1B D0 D8 C8 D0,E2
 1FA8 EF B1 FE 10 0A 29 7F D1,13
 1FB0 FC F0 0C 90 0A B0 C7 09,25
 1FB8 80 D1 FC F0 C1 D0 F4 A0,87
 1FC0 00 B1 FC 91 C8 30 03 C8,88
 1FC8 D0 F7 C0 02 90 F9 A0 00,3A
 1FD0 B1 FE 91 FC 30 03 C8 D0,41
 1FD8 F7 C0 02 90 F9 20 F1 1F,B3
 1FE0 A0 00 B1 C8 91 FE 30 03,8E
 1FE8 C8 D0 F7 C0 02 90 F9 B0,18
 1FF0 83 98 38 65 FC 85 FE A5,F4
 1FF8 FD 69 00 85 FF 60 AA AA,92

9292



Synertek Systems

CORPORATION

150 S. WOLFE RD • SUNNYVALE, CA 94086

TELEPHONE (408) 988-5689

TECHNICAL
NOTE

101SSC

February 1980

ADDING MOTOR CONTROL FOR A SECOND CASSETTE RECORDER TO SYM-1

This reprint distributed with RAE NOTES NO. 2 By SYM-1 USERS' GROUP

The Resident Assembler/Editor (RAE-1) has provision to control two tape recorders allowing a user to assemble large programs from small source modules on one cassette unit and output the object code to a second cassette unit. To fully utilize this feature of RAE with the SYM-1 computer, some additional circuitry must be added external to the SYM board. Also, three jumpers must be installed on the SYM board itself. Note: The MDT 1000 series of development tools have provisions for two cassette recorders and do not need any changes.

Figure 1 gives a list of parts needed and Figure 2 shows the required hookup. In addition, add the following jumpers to your SYM board:

T5 to T17
T7 to T15
▼ to 23

To use the dual recorder hookup, follow the instructions in the RAE-1 Reference Manual (see pages 2-3, 2-14, 3-2, and 3-4). To move the tape in fast forward or reverse modes, you may use either the ON/OFF commands in RAE or the optional push-buttons after selecting the desired function on the cassette unit.

ADDED COMMENTS BY SYM-1 USERS' GROUP

The use of the two relays as shown here is probably more convenient than adapting the circuit shown in Figure 4-4 of the SYM-1 Reference Manual to control the second recorder, especially if you will be using recorders which have different types of remote control connections.

Also incidentally, it is considered good practice to shunt the relay inputs with back-biased diodes (1N914's are fine) to prevent inductive kickbacks from injuring the drivers (see Figure 4-4). Diodes should also be placed across the relay contacts to prevent kickback from the motor inductance from burning out the contacts, prematurely, like within a few hundred closures. The arrangement of the diodes will unfortunately depend on the polarity of the remote control circuit. An expensive way to protect the contacts is a pair of 6 volt Zener diodes back-to-back across them (this is polarity independent); an inexpensive way is a 0.1 ufd capacitor. Let us know about any relay life problems you have.

<u>QUANTITY</u>	<u>REFERENCE</u>	<u>ITEM</u>
2	K1, K2	Relay, SPDT; Radio Shack 275-004
1	U1	7407 Hex Buffer IC
2	P1, P2	Subminiature Phone Plug: Radio Shack 274-289
1	P3	Connector, Dual 22/44; Microplastic 15622DPIC
2	S1, S2	Pushbutton Switch (optional): Radio Shack 275-1549

Figure 1. Parts List. (Equivalent parts may be substituted.)

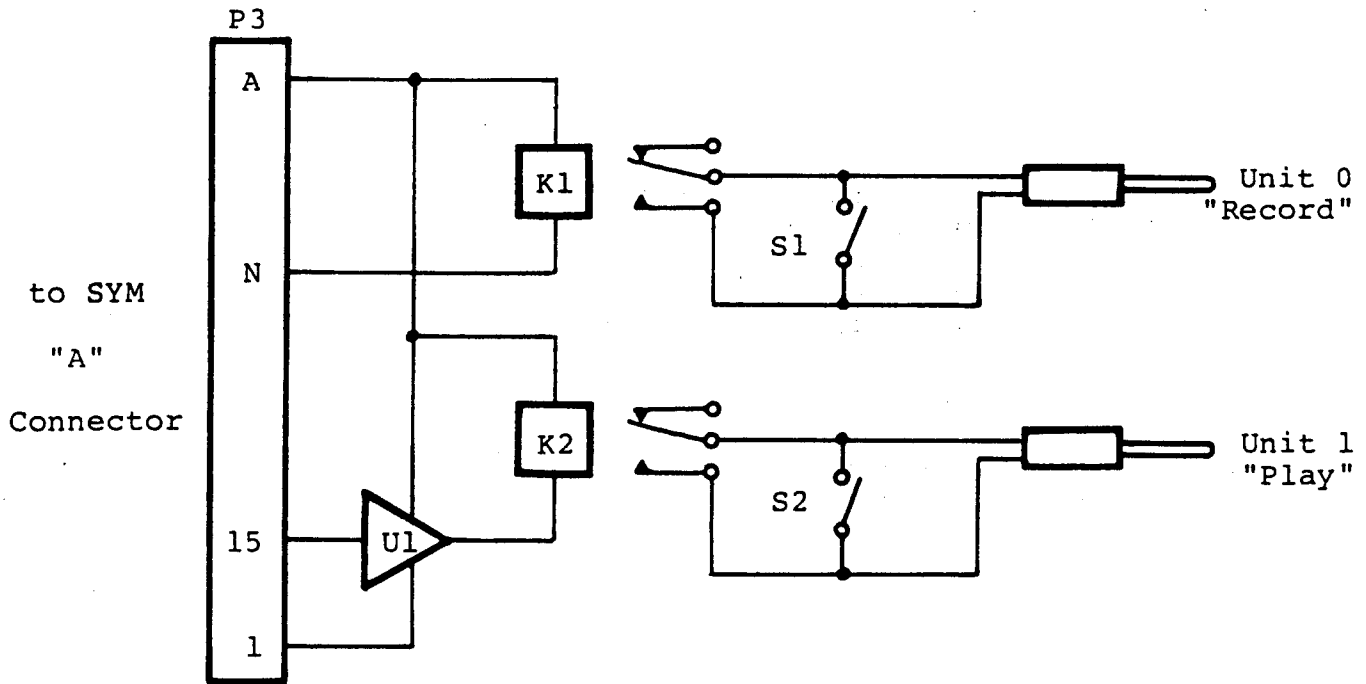


Figure 2. Hookup