

# ATARI® 400/800™

---

## ATARI® HOME COMPUTER SYSTEM

---

# OPERATING SYSTEM SOURCE LISTING

---



A Warner Communications Company 

ATARI HOME COMPUTER SYSTEM

OPERATING SYSTEM  
SOURCE LISTING

COPYRIGHT 1982, ATARI, INC.  
ALL RIGHTS RESERVED

**TO ALL PERSONS RECEIVING THIS DOCUMENT**

Reproduction is forbidden without the specific written permission of ATARI, INC. Sunnyvale, CA 94086. No right to reproduce this document, nor the subject matter thereof, is granted unless by written agreement with, or written permission from the Corporation.

Every effort has been made to ensure that this manual accurately documents this product of the ATARI Home Computer Division. However, due to the ongoing improvement and update of the computer software and hardware, ATARI, INC. cannot guarantee the accuracy of printed material after the date of publication and disclaims liability for changes, errors, or omissions.



1  
2  
3  
4  
5  
6  
7  
8  
9

LIST X

THIS IS THE MODIFIED SEPTEMBER ATARI 400/800 COMPUTER OPERATING SYSTEM LISTING, MODIFIED TO ASSEMBLE ON THE MICROTEC CROSS ASSEMBLER.

THIS VERSION IS THE ONE WHICH WAS BURNED INTO ROM. THERE IS A RESIDUAL PIECE OF CODE WHICH IS FOR LNBUG. THIS IS AT LOCATION \$9000 WHICH IS NOT IN ROM.

THIS IS THE REVISION B EPROM VERSION



```

64 ; TO VARIOUS ENTRY POINTS IN THE OPERATING SYSTEM.
65 ;
66 DISKIV = $E450 ; DISK INITIALIZATION
67 DSKINV = $E453 ; DISK INTERFACE
68 CIOV = $E456 ; CENTRAL INPUT OUTPUT ROUTINE
69 SIOV = $E459 ; SERIAL INPUT OUTPUT ROUTINE
70 SETVBV = $E45C ; SET SYSTEM TIMERS ROUTINE
71 SYSVBV = $E45F ; SYSTEM VERTICAL BLANK CALCULATIONS
72 XITVBV = $E462 ; EXIT VERTICAL BLANK CALCULATIONS
73 SIOINV = $E465 ; SERIAL INPUT OUTPUT INITIALIZATION
74 SENDEV = $E468 ; SEND ENABLE ROUTINE
75 INTINV = $E46B ; INTERRUPT HANDLER INITIALIZATION
76 CIOINV = $E46E ; CENTRAL INPUT OUTPUT INITIALIZATION
77 BLKBDV = $E471 ; BLACKBOARD MODE
78 WARMVS = $E474 ; WARM START ENTRY POINT
79 COLDSV = $E477 ; COLD START ENTRY POINT
80 RBLOKV = $E47A ; CASSETTE READ BLOCK ENTRY POINT VECTOR
81 CSOPIV = $E47D ; CASSETTE OPEN FOR INPUT VECTOR
82 ; VCTABL = $E480
83 ;
84 ;
85 ;
86 ;
87 ; OPERATING SYSTEM EQUATES
88 ;
89 ; COMMAND CODES FOR IOCB
90 OPEN = 3
91 GETREC = 5
92 GETCHR = 7
93 PUTREC = 9
94 PUTCHR = $B
95 CLOSE = $C
96 STATIS = $D
97 SPECIL = $E
98 ;
99 ; SPECIAL ENTRY COMMANDS
100 DRAWLN = $11
101 FILLIN = $12
102 RENAME = $20
103 DELETE = $21
104 FORMAT = $22
105 LOCKFL = $23
106 UNLOCK = $24
107 POINT = $25
108 NOTE = $26
109 IOCFRE = $FF
110 ;
111 ; AUX1 EQUATES WHICH DEVICES USE BIT
112 ; ( ) INDICATES WHICH DEVICES USE BIT
113 APPEND = $1 ; OPEN FOR WRITE APPEND (D), OR SCREEN READ (
114 DIRECT = $2 ; OPEN FOR DIRECTORY ACCESS (D)
115 OPNIN = $4 ; OPEN FOR INPUT (ALL DEVICES)
116 OPNOT = $8 ; OPEN FOR OUTPUT (ALL DEVICES)
117 OPNINO = $10 ; OPEN FOR INPUT AND OUTPUT (ALL DEVICES)
118 MXDMOD = $10 ; OPEN FOR MIXED MODE (E,S)
119 INSLR = $20 ; OPEN WITHOUT CLEARING SCREEN (E,S)

```

```

118 ;
119 ;
120 ; DEVICE NAMES
121 SCREDIT = 'E
122 KBD = 'K
123 DISPLY = 'S
124 PRINTR = 'P
125 CASSET = 'C
126 MODEM = 'M
127 DISK = 'D
128 ;
129 ; SYSTEM EOL (CARRIAGE RETURN)
130 CR = $9B
131 ;
132 ;
133 ; OPERATING SYSTEM STATUS CODES
134 ;
135 ; SUCCES = $01 ; SUCCESSFUL OPERATION
136 ; BRKABT = $80 ; BREAK KEY ADRT
137 ; PRVOPN = $81 ; IOC8 ALREADY OPEN
138 ; NONDEV = $82 ; NON-EXISTANT DEVICE
139 ; WRONLY = $83 ; IOC8 OPENED FOR WRITE ONLY
140 ; NVALID = $84 ; INVALID COMMAND
141 ; NOTOPN = $85 ; DEVICE OR FILE NOT OPEN
142 ; BADIOC = $86 ; INVALID IOC8 NUMBER
143 ; RDONLY = $87 ; IOC8 OPENED FOR READ ONLY
144 ; EOFERR = $88 ; END OF FILE
145 ; TRNRCD = $89 ; TRUNCATED RECORD
146 ; TIMOUT = $8A ; PERIPHERAL DEVICE TIME OUT
147 ; DNACK = $8B ; DEVICE DOES NOT ACKNOWLEDGE COMMAND
148 ; FRMERR = $8C ; SERIAL BUS FRAMING ERROR
149 ; CRSROR = $8D ; CURSOR OVERRANGE
150 ; OVRRUN = $8E ; SERIAL BUS DATA OVERRUN
151 ; CHKERR = $8F ; SERIAL BUS CHECKSUM ERROR
152 ;
153 ; DERROR = $90 ; PERIPHERAL DEVICE ERROR (OPERATION NOT COMP
154 ; BADMOD = $91 ; BAD SCREEN MODE NUMBER
155 ; FNCNOT = $92 ; FUNCTION NOT IMPLEMENTED IN HANDLER
156 ; SCRMEM = $93 ; INSUFFICIENT MEMORY FOR SCREEN MODE
157 ;
158 ;
159 ;
160 ;
161 ;
162 ;
163 ;
164 ;
165 ; PAGE ZERO RAM ASSIGNMENTS
166 ; *=$0000
167 ; LINZBS: .RES 2 ; LINBUG RAM (WILL BE REPLACED BY MONITOR RAM
168 ;
169 ; THESE LOCATIONS ARE NOT CLEARED
170 CASINI: .RES 2 ; CASSETTE INIT LOCATION
171 RAMLO: .RES 2 ; RAM POINTER FOR MEMORY TEST
172 TRAMSZ: .RES 1 ; TEMPORARY REGISTER FOR RAM SIZE

```

```

172 0007 TSTDAT: .RES 1 ; RAM TEST DATA REGISTER
173 ;
174 ; CLEARED ON COLDSTART ONLY
175 WARMST: .RES 1 ; WARM START FLAG
176 BOOT?: .RES 1 ; SUCCESSFUL BOOT FLAG
177 DOSVEC: .RES 2 ; DISK SOFTWARE START VECTOR
178 DOSINI: .RES 2 ; DISK SOFTWARE INIT ADDRESS
179 APPMHI: .RES 2 ; APPLICATIONS MEMORY HI LIMIT
180 ;
181 ; CLEARED ON COLD OR WARM START
182 INTZBS =* ; INTERRUPT HANDLER
183 POKMSK: .RES 1 ; SYSTEM MASK FOR POKEY IRQ ENABLE
184 BRKKEY: .RES 1 ; BREAK KEY FLAG
185 RTCLOCK: .RES 3 ; REAL TIME CLOCK (IN 16 MSEC UNITS)
186 ;
187 BUFADR: .RES 2 ; INDIRECT BUFFER ADDRESS REGISTER
188 ;
189 ICCOMT: .RES 1 ; COMMAND FOR VECTOR
190 ;
191 DSKFMS: .RES 2 ; DISK FILE MANAGER POINTER
192 DSKUTL: .RES 2 ; DISK UTILITIES POINTER
193 ;
194 PTIMOT: .RES 1 ; PRINTER TIME OUT REGISTER
195 PBPNT: .RES 1 ; PRINT BUFFER POINTER
196 PBUFSZ: .RES 1 ; PRINT BUFFER SIZE
197 PTEMP: .RES 1 ; TEMPORARY REGISTER
198 ;
199 ZIOCB =* ; ZERO PAGE I/O CONTROL BLOCK
200 IOCBSZ = 16 ; NUMBER OF BYTES PER IOCB
201 MAXIOC = B*IOCBSZ ; LENGTH OF THE IOCB AREA
202 IOCBAS =* ;
203 ICHIDZ: .RES 1 ; HANDLER INDEX NUMBER (FF = IOCB FREE)
204 ICNDNZ: .RES 1 ; DEVICE NUMBER (DRIVE NUMBER)
205 ICCOMZ: .RES 1 ; COMMAND CODE
206 ICSTAZ: .RES 1 ; STATUS OF LAST IOCB ACTION
207 ICBALZ: .RES 1 ; BUFFER ADDRESS LOW BYTE
208 ICBAHZ: .RES 1 ; PUT BYTE ROUTINE ADDRESS - 1
209 ICPTLZ: .RES 1 ; BUFFER LENGTH LOW BYTE
210 ICPTHZ: .RES 1 ;
211 ICBLHZ: .RES 1 ; AUXILIARY INFORMATION FIRST BYTE
212 ICBLHZ: .RES 1 ;
213 ICAX1Z: .RES 1 ;
214 ICAX2Z: .RES 1 ; TWO SPARE BYTES (CIO LOCAL USE)
215 ICSPRZ: .RES 4 ; IOCB NUMBER X 16
216 ICIDND = ICSPRZ+2 ; CHARACTER BYTE FOR CURRENT OPERATION
217 CIOCHR = ICSPRZ+3 ;
218 ;
219 STATUS: .RES 1 ; INTERNAL STATUS STORAGE
220 CHKSUM: .RES 1 ; CHECKSUM (SINGLE BYTE SUM WITH CARRY)
221 BUFRLO: .RES 1 ; POINTER TO DATA BUFFER (LO BYTE)
222 BUFRHI: .RES 1 ; POINTER TO DATA BUFFER (HI BYTE)
223 BFENLO: .RES 1 ; NEXT BYTE PAST END OF THE DATA BUFFER (LO B
224 BFENHI: .RES 1 ; NEXT BYTE PAST END OF THE DATA BUFFER (HI B
225 CRETRY: .RES 1 ; NUMBER OF COMMAND FRAME RETRIES

```

```

226 0037 DRETRY: RES 1 ; NUMBER OF DEVICE RETRIES
227 0038 BUFRFL: RES 1 ; DATA BUFFER FULL FLAG
228 0039 RECVDN: RES 1 ; RECEIVE DONE FLAG
229 003A XMTDON: RES 1 ; TRANSMISSION DONE FLAG
230 003B CHKSNT: RES 1 ; CHECKSUM SENT FLAG
231 003C NOCKSM: RES 1 ; NO CHECKSUM FOLLOWS DATA FLAG
;
;
233
234 003D BPTR: RES 1 ;
235 003E FTYPE: RES 1 ;
236 003F FEDEF: RES 1 ;
237 0040 FREG: RES 1 ;
238 0041 SOUNDR: RES 1 ; NOISY I/O FLAG. (ZERO IS QUIET)
239 0042 CRITIC: RES 1 ; DEFINES CRITICAL SECTION (CRITICAL IF NON-Z
;
;
241 0043 FMSZPG: RES 7 ; DISK FILE MANAGER SYSTEM ZERO PAGE
;
;
242
243
244 004A CKEY: RES 1 ; FLAG SET WHEN GAME START PRESSED
245 004B CASSBT: RES 1 ; CASSETTE BOOT FLAG
246 004C DSTAT: RES 1 ; DISPLAY STATUS
;
248 004D ATRACT: RES 1 ; ATRACT FLAG
249 004E DRKMSK: RES 1 ; DARK ATRACT MASK
250 004F COLRSH: RES 1 ; ATRACT COLOR SHIFTER (EOR'ED WITH PLAYFIELD
;
251
252 LEDGE = 2 ; LMARGN'S VALUE AT COLD START
253 REDGE = 39 ; RMARGN'S VALUE AT COLD START
254 0050 TMPCHR: RES 1 ;
255 0051 HOLD1: RES 1 ;
256 0052 LMARGN: RES 1 ; LEFT MARGIN (SET TO 1 AT POWER ON)
257 0053 RMARGN: RES 1 ; RIGHT MARGIN (SET TO 38 AT POWER ON)
258 0054 ROWCRS: RES 1 ; CURSOR COUNTERS
259 0055 COLCRS: RES 2 ;
260 0057 DINDEX: RES 1 ;
261 0058 SAVMSC: RES 2 ;
262 005A OLDRON: RES 1 ;
263 005B OLDCOL: RES 2 ;
264 005D OLDCHR: RES 1 ; DATA UNDER CURSOR
265 005E OLDADR: RES 2 ;
266 0060 NEWROM: RES 1 ; POINT DRAW GOES TO
267 0061 NEWCOL: RES 2 ;
268 0063 LOGCOL: RES 1 ; POINTS AT COLUMN IN LOGICAL LINE
269 0064 ADDRESS: RES 2 ;
270 0066 MLTtmp: RES 2 ;
271 0066 DPNTMP = MLTtmp ; FIRST BYTE IS USED IN OPEN AS TEMP
272 0068 SAVADR: RES 2 ;
273 006A RAMTOP: RES 1 ; RAM SIZE DEFINED BY POWER ON LOGIC
274 006B BUFCNT: RES 1 ; BUFFER COUNT
275 006C BUFSTR: RES 2 ; EDITOR GETCH POINTER
276 006E BITMSK: RES 1 ; BIT MASK
277 006F SHFAMT: RES 1 ;
278 0070 ROWAC: RES 2 ;
279 0072 COLAC: RES 2 ;

```





```

334 022C CDTMF4: RES 1 ;COUNT DOWN TIMER 4 FLAG
335 022D INTMP: RES 1 ;IAN'S TEMP (RENAMED FROM T1 BY POPULAR DEMA
336 022E CDTMF5: RES 1 ;COUNT DOWN TIMER FLAG 5
337 022F SDMCTL: RES 1 ;SAVE DMACTL REGISTER
338 0230 SDLSTL: RES 1 ;SAVE DISPLAY LIST LOW BYTE
339 0231 SDLSTH: RES 1 ;SAVE DISPLAY LIST HI BYTE
340 0232 SSKCTL: RES 1 ;SKCTL REGISTER RAM
341 0233 RES 1 ;
342 ;
343 0234 LPENH: RES 1 ;LIGHT PEN HORIZONTAL VALUE
344 0235 LPENV: RES 1 ;LIGHT PEN VERTICAL VALUE
345 0236 BRKKY: RES 2 ;BREAK KEY VECTOR
346 ;
347 0238 RES 2 ;SPARE
348 ;
349 023A CDEVIC: RES 1 ;COMMAND FRAME BUFFER - DEVICE
350 023B CCOMND: RES 1 ;COMMAND
351 023C CAUX1: RES 1 ;COMMAND AUX BYTE 1
352 023D CAUX2: RES 1 ;COMMAND AUX BYTE 2
353 023E ; NOTE MAY NOT BE THE LAST WORD ON A PAGE
354 ; TEMP: RES 1 ;TEMPORARY RAM CELL
355 ; NOTE MAY NOT BE THE LAST WORD ON A PAGE
356 023F ERRFLG: RES 1 ;ERROR FLAG - ANY DEVICE ERROR EXCEPT TIME 0
357 ;
358 0240 DFLAGS: RES 1 ;DISK FLAGS FROM SECTOR ONE
359 0241 DBSECT: RES 1 ;NUMBER OF DISK BOOT SECTORS
360 0242 BOOTAD: RES 2 ;ADDRESS WHERE DISK BOOT LOADER WILL BE PUT
361 0244 COLDST: RES 1 ;COLDSTART FLAG (1=IN MIDDLE OF COLDSTART)
362 ;
363 0245 RES 1 ;SPARE
364 ;
365 0246 DSKTIM: RES 1 ;DISK TIME OUT REGISTER
366 ;
367 0247 LINBUF: RES 40 ;CHAR LINE BUFFER
368 ;
369 026F GPRIOR: RES 1 ;GLOBAL PRIORITY CELL
370 ;
371 0270 PADDLO: RES 1 ;POTENTIOMETER 0 RAM CELL
372 0271 PADDL1: RES 1 ;
373 0272 PADDL2: RES 1 ;
374 0273 PADDL3: RES 1 ;
375 0274 PADDL4: RES 1 ;
376 0275 PADDL5: RES 1 ;
377 0276 PADDL6: RES 1 ;
378 0277 PADDL7: RES 1 ;
379 0278 STICKO: RES 1 ;JOYSTICK 0 RAM CELL
380 0279 STICK1: RES 1 ;
381 027A STICK2: RES 1 ;
382 027B STICK3: RES 1 ;
383 027C PTRIGO: RES 1 ;
384 027D PTRIG1: RES 1 ;
385 027E PTRIG2: RES 1 ;
386 027F PTRIG3: RES 1 ;
387 0280 PTRIG4: RES 1 ;PADDLE TRIGGER 0

```



```

442      ;
443      ;
444      ;
445      ;
446      ;
447      ;
448      ;
449      ;
450      ;
451      ;
452      ;
453      ;
454      ;
455      ;
456      ;
457      ;
458      ;
459      ;
460      ;
461      ;
462      ;
463      ;
464      ;
465      ;
466      ;
467      ;
468      ;
469      ;
470      ;
471      ;
472      ;
473      ;
474      ;
475      ;
476      ;
477      ;
478      ;
479      ;
480      ;
481      ;
482      ;
483      ;
484      ;
485      ;
486      ;
487      ;
488      ;
489      ;
490      ;
491      ;
492      ;
493      ;
494      ;
495      ;

```

02C9 ; SPARE  
 02E0 ; SPARE  
 02E4 ; RAM SIZE (HI BYTE ONLY)  
 02E5 ; TOP OF AVAILABLE USER MEMORY  
 02E7 ; BOTTOM OF AVAILABLE USER MEMORY  
 02E9 ; SPARE  
 02EA ; STATUS BUFFER  
 02EE ; CASSETTE BAUD RATE LOW BYTE  
 02EF ;  
 02F0 ; CURSOR INHIBIT (00 = CURSOR ON)  
 02F1 ; KEY DELAY  
 02F2 ;  
 02F3 ; CHACTL REGISTER RAM  
 02F4 ; CHBAS REGISTER RAM  
 02F5 ; SPARE BYTES  
 02FA ;  
 02FB ; ATASCII CHARACTER  
 02FC ; GLOBAL VARIABLE FOR KEYBOARD  
 02FD ; RIGHT FILL DATA (DRAW)  
 02FE ; DISPLAY FLAG : DISPLAY CNTLS IF NON-ZERO  
 02FF ; START/STOP FLAG FOR PAGING (CNTL 1). CLEAR

PAGE THREE RAM ASSIGNMENTS

```

DCB      ;
DDEVIC:  .RES 1 ; DEVICE CONTROL BLOCK
DUNIT:   .RES 1 ; PERIPHERAL UNIT 1 BUS I. D. NUMBER
DCOMND:  .RES 1 ; UNIT NUMBER
DSTATS:  .RES 1 ; BUS COMMAND
DUFLO:   .RES 1 ; COMMAND TYPE/STATUS RETURN
DUFHI:   .RES 1 ; DATA BUFFER POINTER LOW BYTE
DTIMLO:  .RES 1 ; DEVICE TIME OUT IN 1 SECOND UNITS
DUNUSE:  .RES 1 ; UNUSED BYTE
DBYTLO:  .RES 1 ; NUMBER OF BYTES TO BE TRANSFERRED LOW BYTE
DBYTHI:  .RES 1 ;
DAUX1:   .RES 1 ; COMMAND AUXILIARY BYTE 1
DAUX2:   .RES 1 ;

```



550  
551  
552  
553  
554  
555  
556  
557  
558  
559  
560  
561  
562  
563  
564  
565  
566  
567  
568  
569  
570  
571  
572  
573  
574  
575  
576  
577  
578  
579  
580  
581  
582  
583  
584  
585  
586  
587  
588  
589  
590  
591  
592  
593  
594  
595  
596  
597  
598  
599  
600  
601  
602  
603

PAGE FIVE RAM ASSIGNMENTS  
PAGE FIVE IS RESERVED AS A USER WORK SPACE  
NOTE: SEE FLOATING POINT SUBROUTINE AREA FOR PAGE FIVE CELLS

PAGE SIX RAM ASSIGNMENTS  
PAGE SIX IS RESERVED AS A USER'S USER WORK SPACE

FLOATING POINT SUBROUTINES

```

FPREC = 6 ; FLOATING PT PRECISION (# OF BYTES)
; IF CARRY USED THEN CARRY CLEAR => NO ERROR, CARR
AFP = $D800 ; ASCII->FLOATING POINT (FP)
;
FASC = $DBE6 ; FP -> ASCII FRO-> LBUFF (INBUFF)
IFP = $D9AA ; INTEGER -> FP
;
FPI = $D9D2 ; FP -> INTEGER FRO -> FRO, FRO+1, CARRY
FSUB = $DA60 ; FRO <- FRO - FRI, CARRY
FADD = $DA66 ; FRO <- FRO + FRI, CARRY
FMUL = $DADB ; FRO <- FRO * FRI, CARRY
FDIV = $DB28 ; FRO <- FRO / FRI, CARRY
FLDOR = $DD89 ; FLOATING LOAD REGO FRO <- (X, Y)
FLDOP = $DD8D ; " " " " (FLPTR)
FLDIR = $DD98 ; " " REG1 FRI <- (X, Y)
FLDIP = $DD9C ; " " " " (FLPTR)
FSTOP = $DDA7 ; FLOATING STORE REGO (X, Y) <- FRO
FSTOP = $DDAB ; " " " (FLPTR) <- FRO
FMOVE = $DDB6 ; FRI <- FRO
PLYEVL = $DD40 ; FRO <- P(Z) = SUM(I=1 TO 0) (A(I)*Z**I) CAR
; INPUT: (X, Y) = A(N), A(N-1), ... A(0) -> PLYARG
; ACC = # OF COEFFICIENTS = DEGREE+1
; FRO = Z
;
EXP = $DDCO ; FRO <- E**FRO = EXP10(FRO * LOG10(E)) CARRY
EXPI0 = $DDCC ; FRO <- 10**FRO CARRY
LOG = $DECD ; FRO <- LN(FRO) = LOG10(FRO)/LOG10(E) CARRY
LOG10 = $DED1 ; FRO <- LOG10 (FRO) CARRY
; THE FOLLOWING ARE IN BASIC CARTRIDGE:
SIN = $BD81 ; FRO <- SIN(FRO) DEGFLG=0 =>RADS, 6=>DEG. CA
COS = $BD73 ; FRO <- COS(FRO) CARRY
ATAN = $BE43 ; FRO <- ATAN(FRO) CARRY
SGR = $BE81 ; FRO <- SQUAREROOT(FRO) CARRY
    
```

; FLOATING POINT ROUTINES ZERO PAGE (NEEDED ONLY IF F.P. ROUTINES ARE CA

```

*=#D4
FRO: .RES FPREC ;FP REGO
FRE: .RES FPREC ;FP REG1
FR1: .RES FPREC
FR2: .RES FPREC
FRX: .RES 1 ;FP SPARE
EEXP: .RES 1 ;VALUE OF E
NSIGN: .RES 1 ;SIGN OF #
ESIGN: .RES 1 ;SIGN OF EXPONENT
FCHRFLG: .RES 1 ;1ST CHAR FLAG
DIGRT: .RES 1 ;# OF DIGITS RIGHT OF DECIMAL
CIX: .RES 1 ;CURRENT INPUT INDEX
INBUFF: .RES 2 ;POINTS TO USER'S LINE INPUT BUFFER
ZTEMP1: .RES 2
ZTEMP4: .RES 2
ZTEMP3: .RES 2
DEGFLG .RES 1 ;0=RADIANS, 6=DEGREES
RADON = 0 ;INDICATES RADIAN
DEGON = 6 ;INDICATES DEGREE
FLPTR: .RES 2 ;POINTS TO USER'S FLOATING PT NUMBER
FPTR2: .RES 2

```

; FLOATING PT ROUTINES' NON-ZERO PAGE RAM (NEEDED ONLY IF F.P. ROUTINES CALLED)

```

*=#57E
LBPR1: .RES 1 ;LBUFF PREFIX 1
LBPR2: .RES 1 ;LBUFF PREFIX 2
LBUFF: .RES 128 ;LINE BUFFER
PLYARG = LBUFF+*60 ;POLYNOMIAL ARGUMENTS
FPSCR = PLYARG+FPREC
FPSCR1 = FPSCR
FSCR1 = FPSCR1
LBFEND = *-1 ;END OF LBUFF

```

COLLEEN MNEMONICS

```

POKEY = $D200 ;VBLANK ACTION:
POTO = POKEY+0 ;POTO-->PADDL0
POT1 = POKEY+1 ;POT1-->PADDL1
POT2 = POKEY+2 ;POT2-->PADDL2
POT3 = POKEY+3 ;POT3-->PADDL3
POT4 = POKEY+4 ;POT4-->PADDL4
POT5 = POKEY+5 ;POT5-->PADDL5
POT6 = POKEY+6 ;POT6-->PADDL6

```

DESCRIPTION: 0-227 IN RAM CELL 0-227 IN RAM CELL 0-227 IN RAM CELL 0-227 IN RAM CELL 0-227 IN RAM CELL 0-227 IN RAM CELL 0-227 IN RAM CELL

604  
605  
606 00D4  
607 00DA  
608 00E0  
609 00E6  
610 00EC  
611 00ED  
612 00EE  
613 00EF  
614 00FO  
615 00F1  
616 00F2  
617 00F3  
618 00F5  
619 00F7  
620 00F9  
621 00FB  
622 00FB  
623 0000  
624 0006  
625 00FC  
626 00FE  
627  
628  
629  
630 057E  
631 057F  
632 0580  
633 05E0  
634 05E6  
635 05EC  
636 05E6  
637 05EC  
638 05FF  
639  
640  
641  
642  
643  
644  
645  
646  
647  
648  
649  
650 D200  
651 D200  
652 D201  
653 D202  
654 D203  
655 D204  
656 D205  
657 D206





|     |      |        |   |          |                   |
|-----|------|--------|---|----------|-------------------|
| 712 | D01E | HITCLR | = | CTIA+30  |                   |
| 713 | D01F | CONSOL | = | CTIA+31  | TURN OFF SPEAKER  |
| 714 | D000 | MOPF   | = | CTIA+0   |                   |
| 715 | D001 | M1PF   | = | CTIA+1   |                   |
| 716 | D002 | M2PF   | = | CTIA+2   |                   |
| 717 | D003 | M3PF   | = | CTIA+3   |                   |
| 718 | D004 | P0PF   | = | CTIA+4   |                   |
| 719 | D005 | P1PF   | = | CTIA+5   |                   |
| 720 | D006 | P2PF   | = | CTIA+6   |                   |
| 721 | D007 | P3PF   | = | CTIA+7   |                   |
| 722 | D008 | MORL   | = | CTIA+8   |                   |
| 723 | D009 | M1PL   | = | CTIA+9   |                   |
| 724 | D00A | M2PL   | = | CTIA+10  |                   |
| 725 | D00B | M3PL   | = | CTIA+11  |                   |
| 726 | D00C | POPL   | = | CTIA+12  |                   |
| 727 | D00D | P1PL   | = | CTIA+13  |                   |
| 728 | D00E | P2PL   | = | CTIA+14  |                   |
| 729 | D00F | P3PL   | = | CTIA+15  |                   |
| 730 | D010 | TRIG0  | = | CTIA+16  |                   |
| 731 | D011 | TRIG1  | = | CTIA+17  |                   |
| 732 | D012 | TRIG2  | = | CTIA+18  |                   |
| 733 | D013 | TRIG3  | = | CTIA+19  |                   |
| 734 |      |        | = |          |                   |
| 735 | D400 | ANTIC  | = | \$D400   |                   |
| 736 | D400 | DMACTL | = | ANTIC+0  | DESCRIPTION       |
| 737 | D401 | CHACTL | = | ANTIC+1  | ON OPEN S: OR E:  |
| 738 | D402 | DLISTL | = | ANTIC+2  | ON OPEN S: OR E:  |
| 739 | D403 | DLISTH | = | ANTIC+3  | ON OPEN S: OR E:  |
| 740 | D404 | HSCRDL | = | ANTIC+4  | ON OPEN S: OR E:  |
| 741 | D405 | VSCRBL | = | ANTIC+5  |                   |
| 742 | D407 | PMBASE | = | ANTIC+7  |                   |
| 743 | D409 | CHBASE | = | ANTIC+9  | ON OPEN S: OR E:  |
| 744 | D40A | WSYNC  | = | ANTIC+10 |                   |
| 745 | D40B | VCOUNT | = | ANTIC+11 |                   |
| 746 | D40C | PENH   | = | ANTIC+12 |                   |
| 747 | D40D | PENV   | = | ANTIC+13 |                   |
| 748 | D40E | NMIEN  | = | ANTIC+14 | ON AND [SETVBV]   |
| 749 | D40F | NMIRE  | = | ANTIC+15 |                   |
| 750 | D40F | NMIST  | = | ANTIC+15 |                   |
| 751 | D300 | PIA    | = | \$D300   |                   |
| 752 | D300 | PORTA  | = | PIA+0    | DESCRIPTION       |
| 753 | D301 | PORTB  | = | PIA+1    | X-Y CONTROLLERS   |
| 754 | D302 | PACTL  | = | PIA+2    | X-Y CONTROLLERS   |
| 755 | D303 | PBCTL  | = | PIA+3    | PACTL<--3C [INIT] |
| 756 |      |        | = |          | PBCTL<--3C [INIT] |
| 757 |      |        | = |          |                   |
| 758 |      |        | = |          |                   |
| 759 |      |        | = |          |                   |

760  
 761  
 762  
 763  
 764 0030  
 765 003A  
 766 009B

.PAGE  
 LIST

S 'CENTRAL INPUT/OUTPUT (CIO) 2-7-79'  
 UPDATED BY AL MILLER 3-9-79  
 ;  
 ; ASCII ZERO  
 ; ASCII COLON  
 ; END OF RECORD

ASCZER  
 COLON  
 EOL

CENTRAL INPUT/OUTPUT (CIO) 2-7-79

ERR LINE ADDR B1 B2 B3 B4

```

767 . PAGE
768 ;
769 ; CIO JUMP VECTOR FOR USERS
770 *CIOV
771 JMP CIO ;GO TO CIO
772
773 ; CIO INIT JUMP VECTOR FOR POWER UP
774 *CIOINV
775 JMP CIOINT ;GO TO INIT
776
777 ;
778 ; ERROR ROUTINE ADDRESS EQUATE
779 ERRTNH =ERRTN/256 "MOVED TO LINE 788"
780 ERRTNL =-ERRTNH*256+ERRTN "MOVED TO LINE 789"
781
782 ;
783 *CIOORG
784
785 ; CIO INITIALIZATION (CALLED BY MONITOR AT POWER UP)
786 CIOINT: LDX #0
787 CIOI1: LDA #IOCFRE ;SET ALL IOCB'S TO FREE
788 STA ICHID,X ;BY SETTING HANDLER ID'S=0FF
789 LDA #ERRTNL ;POINT PUT TO ERROR ROUTINE
790 STA ICPTL,X
791 LDA #ERRTNH
792 STA ICPTH,X
793 TXA
794 CLC
795 ADC #IOCBSZ ;BUMP INDEX BY SIZE
796 TAX
797 CMP #MAXIOC ;DONE?
798 BCC CIOI1 ;NO
799 RTS ;YES, RETURN
800
801 ; ERROR ROUTINE FOR ILLEGAL PUT
802 ERRTN =*-1
803 ERRTNH =ERRTN/256
804 ERRTNL =(-ERRTNH)*256+ERRTN
805 E4C1 A0 B5
806 E4C3 60

```

```

807 . PAGE
808 ; CIO LOCAL RAM (USES SPARE BYTES IN ZERO PAGE IOCB)
809 ENTVEC = ICSPRZ
810 ;
811 ; CIO MAIN ROUTINE
812 ;
813 ;
814 ; CIO INTERFACES BETWEEN USER AND INPUT/OUTPUT DE
815 CIO: STA CIOCHR ;SAVE POSSIBLE OUTPUT CHARACTER
816 STX ICIDND ;SAVE IOCB NUMBER * N
817 ;
818 ; CHECK FOR LEGAL IOCB
819 TXA
820 AND ##F ;IS IOCB MULTIPLE OF 16?
821 BNE CIERR1 ;NO, ERROR
822 CPX #MAXIOC ;IS INDEX TOO LARGE?
823 BCC IOC1 ;NO
824 ;
825 ; INVALID IOCB NUMBER -- RETURN ERROR
826 CIERR1: LDY #BADIOC ;ERROR CODE
827 JMP CIRTN1 ;RETURN
828 ;
829 ; MOVE USER IOCB TO ZERO PAGE
830 IOC1: LDY #0
831 IOC1A: LDA IOCB,X ;USER IOCB
832 STA IOCBAS,Y ;TO ZERO PAGE
833 INX
834 INY
835 CPY #12 ;12 BYTES
836 BCC IOC1A
837 ;
838 ; COMPUTE CIO INTERNAL VECTOR FOR COMMAND
839 LDY #INVALID ;ASSUME INVALID CODE
840 LDA ICCOMZ ;COMMAND CODE TO INDEX
841 CMP #OPEN ;IS COMMAND LEGAL?
842 BCC CIERR4 ;NO
843 TAY
844 ;
845 ; MOVE COMMAND TO ZERO BASE FOR INDEX
846 CPY #SPECIL ;IS COMMAND SPECIAL?
847 BCC IOC2 ;NO
848 LDY #SPECIL ;YES, SET SPECIAL OFFSET INDEX
849 STY ICCOMT ;SAVE COMMAND FOR VECTOR
850 LDA COMTAB-3,Y ;GET VECTOR OFFSET FROM TABLE
851 BEG CIOOPEN ;GO IF OPEN COMMAND
852 CMP #2 ;IS IT CLOSE?
853 BEG CIOCLOSE ;YES
854 CMP #8 ;IS IT STATUS OR SPECIAL?
855 BCS CIOSTSP ;YES
856 CMP #4 ;IS IT READ?
857 BEG CIOREAD ;YES
858 JMP CIWRIT ;ELSE, MUST BE WRITE

```

```

859 .PAGE
860 ;
861 ; OPEN COMMAND
862 ;
863 ; FIND DEVICE HANDLER IN HANDLER ADDRESS TABLE
864 CIOPEN: LDA ICHIDZ ;GET HANDLER ID
865 CMP #IOCFRE ;IS THIS IOCB CLOSED?
866 BEQ IOC6 ;YES
867 ;
868 ; ERROR -- IOCB ALREADY OPEN
869 CIERR3: LDY #PRVOPN ;ERROR CODE
870 CIERR4: JMP CIRTN1 ;RETURN
871 ;
872 ; GO FIND DEVICE
873 IOC6: JSR DEVSRC ;CALL DEVICE SEARCH
874 BCS CIERR4 ;GO IF DEVICE NOT FOUND
875 ;
876 ; DEVICE FOUND, INITIALIZE IOCB FOR OPEN
877 ;
878 ; COMPUTE HANDLER ENTRY POINT
879 IOC7: JSR COMENT
880 BCS CIERR4 ;GO IF ERROR IN COMPUTE
881 ;
882 ; GO TO HANDLER FOR INITIALIZATION
883 JSR GOHAND ;USE INDIRECT JUMP
884 ;
885 ; STORE PUT BYTE ADDRESS-1 INTO IOCB
886 LDA #PUTCHR ;SIMULATE PUT CHARACTER
887 STA ICCOMT
888 JSR COMENT ;COMPUTE ENTRY POINT
889 LDA ICSPRZ ;MOVE COMPUTED VALUE
890 STA ICPTLZ ;TO PUT BYTE ADDRESS
891 LDA ICSPRZ+1
892 STA ICPTHZ
893 JMP CIRTN2 ;RETURN TO USER

```

```

894 . PAGE
895 ;
896 ;
897 ;
898 ; CLOSE COMMAND
899 ; C1CLOS: LDY #SUCCES ; ASSUME GOOD CLOSE
900 STY ICSTAZ ; COMPUTE HANDLER ENTRY POINT
901 JSR COMENT ; GO IF ERROR IN COMPUTE
902 BCS C1CLO2 ; GO TO HANDLER TO CLOSE DEVICE
903 JSR GOHAND ; GET IOCB "FREE" VALUE
904 LDA #IOCFRE ; SET HANDLER ID
905 STA ICHIDZ
906 LDA #HERRTNH ; SET PUT BYTE TO POINT TO ERROR
907 STA ICPTHZ
908 LDA #HERRTNL
909 STA ICPTLZ
910 JMP CIRTN2 ; RETURN
911 ;
912 ;
913 ; STATUS AND SPECIAL REQUESTS
914 ; DO IMPLIED OPEN IF NECESSARY AND GO TO DEVICE
915 C1STSP: LDA ICHIDZ ; IS THERE A HANDLER ID?
916 CMP #IOCFRE
917 BNE CIST1 ; YES
918 ;
919 ; IOCB IS FREE, DO IMPLIED OPEN
920 JSR DEVSRC ; FIND DEVICE IN TABLE
921 BCS CIERR4 ; GO IF ERROR IN COMPUTE
922 ;
923 ; COMPUTE AND GO TO ENTRY POINT IN HANDLER
924 CIST1: JSR COMENT ; COMPUTER HANDLER ENTRY VECTOR
925 JSR GOHAND ; GO TO HANDLER
926 ;
927 ; RESTORE HANDLER INDEX (DO IMPLIED CLOSE)
928 LDX ICIDNO ; IOCB INDEX
929 LDA ICHID,X ; GET ORIGINAL HANDLER ID
930 STA ICHIDZ ; RESTORE ZERO PAGE
931 JMP CIRTN2 ; RETURN

```

```

931 . PAGE
932 ;
933 ; READ -- DO GET COMMANDS
934 CIREAD: LDA ICCOMZ ;GET COMMAND BYTE
935 AND ICAX1Z ;IS THIS READ LEGAL?
936 BNE RC11A ;YES
937 ;
938 ; ILLEGAL READ -- IOCB OPENED FOR WRITE ONLY
939 LDY #WRONLY ;ERROR CODE
940 JMP CIRTN1 ;RETURN
941 ;
942 ; COMPUTE AND CHECK ENTRY POINT
943 RC11A: JSR COMENT ;COMPUTE ENTRY POINT
944 BCS RC11B ;GO IF ERROR IN COMPUTE
945 ;
946 ; GET RECORD OR CHARACTERS
947 LDA ICBL LZ
948 ORA ICBL LZ+1 ;IS BUFFER LENGTH ZERO?
949 BNE RC13 ;NO
950 JSR GOHAND
951 STA CIOCHR
952 JMP CIRTN2
953 ;
954 ; LOOP TO FILL BUFFER OR END RECORD
955 RC13: JSR GOHAND ;GO TO HANDLER TO GET BYTE
956 STA CIOCHR ;SAVE BYTE
957 BMI RC14 ;END TRANSFER IF ERROR
958 LDY #0
959 STA (ICBALZ),Y ;PUT BYTE IN USER BUFFER
960 JSR INCBFF ;INCREMENT BUFFER POINTER
961 LDA ICCOMZ ;GET COMMAND CODE
962 AND #2 ;IS IT GET RECORD?
963 BNE RC11 ;NO
964 ;
965 ; CHECK FOR EOL ON TEXT RECORDS
966 LDA CIOCHR ;GET BYTE
967 CMP #EOL ;IS IT AN EOL?
968 BNE RC11 ;NO
969 JSR DECBFL ;YES, DECREMENT BUFFER LENGTH
970 JMP RC14 ;END TRANSFER
971 ;
972 ; CHECK BUFFER FULL
973 RC11: JSR DECBFL ;DECREMENT BUFFER LENGTH
974 BNE RC13 ;CONTINUE IF NON ZERO

```

```

975 . PAGE
976
977 ; BUFFER FULL, RECORD NOT ENDED
978 ; DISCARD BYTES UNTIL END OF RECORD
979 RC12: LDA ICCOMZ ;GET COMMAND BYTE
980 AND #2 ;IS IT GET CHARACTER?
981 BNE RC14 ;YES, END TRANSFER
982
983 ; LOOP TO WAIT FOR EOL
984 RC14: JSR GOHAND ;GET BYTE FROM HANDLER
985 STA CIOCHR ;SAVE CHARACTER
986 BMI RC14 ;GO IF ERROR
987
988 ; TEXT RECORD, WAIT FOR EOL
989 LDA CIOCHR ;GET GOT BYTE
990 CMP #EOL ;IS IT EOL?
991 BNE RC16 ;NO, CONTINUE
992
993 ; END OF RECORD: BUFFER FULL -- SEND TRUNCATED RECORD MESSAGE
994 RC11: LDA #TRNRCD ;ERROR CODE
995 STA ICSTAZ ;STORE IN IOCB
996
997 ; TRANSFER DONE
998 RC14: JSR SUBBFL ;SET FINAL BUFFER LENGTH
999 JMP CIRTN2 ;RETURN

```



```

1000 . PAGE
1001 ;
1002 ; WRITE -- DO PUT COMMANDS
1003 C1WRIT: LDA ICCOMZ ;GET COMMAND BYTE
1004 AND ICAX1Z ;IS THIS WRITE LEGAL?
1005 BNE WC11A ;YES
1006 ;
1007 ; ILLEGAL WRITE -- DEVICE OPENED FOR READ ONLY
1008 LDY #RDONLY ;ERROR CODE
1009 JSR CIRTN1 ;RETURN
1010 WC11B: JMP
1011 ;
1012 ; COMPUTE AND CHECK ENTRY POINT
1013 WC11A: JSR COMENT ;COMPUTE HANDLER ENTRY POINT
1014 BCS WC11B ;GO IF ERROR IN COMPUTE
1015 ;
1016 ; PUT RECORD OR CHARACTERS
1017 LDA ICBL LZ
1018 ORA ICBL LZ+1 ;IS BUFFER LENGTH ZERO?
1019 BNE WC13 ;NO
1020 LDA CIOCHR ;GET CHARACTER
1021 INC ICBL LZ ;SET BUFFER LENGTH=1
1022 BNE WC14 ;THEN JUST TRANSFER ONE BYTE
1023 ;
1024 ; LOOP TO TRANSFER BYTES FROM BUFFER TO HANDLER
1025 WC13: LDY #0
1026 LDA (ICBALZ),Y ;GET BYTE FROM BUFFER
1027 STA CIOCHR ;SAVE
1028 JSR GOHAND ;GO PUT BYTE
1029 BMI WC15 ;END IF ERROR
1030 JSR INCBFP ;INCREMENT BUFFER POINTER
1031 ;
1032 ; CHECK FOR TEXT RECORD
1033 LDA ICCOMZ ;GET COMMAND BYTE
1034 AND #2 ;IS IT PUT RECORD?
1035 BNE WC11 ;NO
1036 ;
1037 ; TEXT RECORD --- CHECK FOR EOL TRANSFER
1038 LDA CIOCHR ;GET LAST CHARACTER
1039 CMP #EOL ;IS IT AN EOL?
1040 BNE WC11 ;NO
1041 JSR DECBFL ;DECREMENT BUFFER LENGTH
1042 JMP WC15 ;END TRANSFER
1043 ;
1044 ; CHECK FOR BUFFER EMPTY
1045 WC11: JSR DECBFL ;DECREMENT BUFFER LENGTH
1046 BNE WC13 ;CONTINUE IF NON ZERO

```

1000

1001

1002

1003

1004

1005

1006

1007

1008

1009

1010

1011

1012

1013

1014

1015

1016

1017

1018

1019

1020

1021

1022

1023

1024

1025

1026

1027

1028

1029

1030

1031

1032

1033

1034

1035

1036

1037

1038

1039

1040

1041

1042

1043

1044

1045

E5C9 A5 22

E5CB 25 2A

E5CD D0 05

E5CF A0 B7

E5D1 4C 1B E6

E5D4 20 3D E6

E5D7 B0 FB

E5D9 A5 28

E5DB 05 29

E5DD D0 06

E5DF A5 2F

E5E1 E6 28

E5E3 D0 06

E5E5 A0 00

E5E7 B1 24

E5E9 85 2F

E5EB 20 89 E6

E5EE 30 25

E5F0 20 70 E6

E5F3 A5 22

E5F5 29 02

E5F7 D0 0C

E5F9 A5 2F

E5FB C9 98

E5FD D0 06

E5FF 20 63 E6

E602 4C 15 E6

E605 20 63 E6

E608 D0 D8

```

1046          PAGE
1047          ; BUFFER EMPTY, RECORD NOT FILLED
1048          ; CHECK TYPE OF TRANSFER
1049          WC12: LDA ICCOMZ      ;GET COMMAND CODE
1050                AND #2        ;IS IT PUT CHARACTER?
1051                BNE WC15      ;YES, END TRANSFER
1052
1053          ; PUT RECORD (TEXT), BUFFER EMPTY, SEND EOL
1054          LDA #EOL            ;GO TO HANDLER
1055          JSR GOHAND
1056
1057          ; END PUT TRANSFER
1058          WC15: JSR SUBBFL     ;SET ACTUAL PUT BUFFER LENGTH
1059                JMP CIRTN2    ;RETURN
1060

```