

5 of 10

ATARI
PRELIMINARY
CONFIDENTIAL

C I O

13-NOV-79

COLLEEN OPERATING SYSTEM EQUATE FILE

MODULE ORIGIN TABLE

E000	CHORQ	= \$E000 CHARACTER SET
E400	VECTBL	= \$E400 VECTOR TABLE
E480	VCTABL	= \$E480 RAM VECTOR INITIAL VALUE TABLE
E4A6	CIOORQ	= \$E4A6 CENTRAL I/O HANDLER
E6D5	INTORQ	= \$E6D5 INTERRUPT HANDLER
E944	SIOORQ	= \$E944 SERIAL I/O DRIVER
EDEA	DSKORQ	= \$EDEA DISK HANDLER
EE7B	PRNORQ	= \$EE7B PRINTER HANDLER
EF41	CASORQ	= \$EF41 CASSETTE HANDLER
FOE3	MONORQ	= \$FOE3 MONITOR/POWER UP MODULE
F3E4	KBDORQ	= \$F3E4 KEYBOARD/DISPLAY HANDLER

VECTOR TABLE

HANDLER ENTRY POINTS ARE CALLED OUT IN THE FOLLOWING VECTOR
TABLE. THESE ARE THE ADDRESSES MINUS ONE.

EXAMPLE FOR EDITOR

E400	OPEN
2	CLOSE
4	GET
6	PUT
8	STATUS
A	SPECIAL
C	JUMP TO POWER ON INITIALIZATION ROUTINE
F	NOT USED

E400	EDITRV	= \$E400 EDITOR
E410	BCRENV	= \$E410 TELEVISION SCREEN
E420	KEYBDV	= \$E420 KEYBOARD
E430	PRINTV	= \$E430 PRINTER
E440	CASERV	= \$E440 CASSETTE

JUMP VECTOR TABLE

THE FOLLOWING IS A TABLE OF JUMP INSTRUCTIONS
TO VARIOUS ENTRY POINTS IN THE OPERATING SYSTEM.

E450	DISKIV	= \$E450 DISK INITIALIZATION
E453	DSKINV	= \$E453 DISK INTERFACE
E456	CIOV	= \$E456 CENTRAL INPUT OUTPUT ROUTINE
E459	SIOV	= \$E459 SERIAL INPUT OUTPUT ROUTINE
E45C	SETVBV	= \$E45C SET SYSTEM TIMERS ROUTINE
E45F	SVSVBV	= \$E45F SYSTEM VERTICAL BLANK CALCULATIONS
E462	XITVBV	= \$E462 EXIT VERTICAL BLANK CALCULATIONS

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

E465	SIOINV	=	*E465	SERIAL INPUT OUTPUT INITIALIZATION
E468	SENDEV	=	*E468	SEND ENABLE ROUTINE
E46B	INTINV	=	*E46B	INTERRUPT HANDLER INITIALIZATION
E46E	CIOINV	=	*E46E	CENTRAL INPUT OUTPUT INITIALIZATION
E471	BLKBDV	=	*E471	BLACKBOARD MODE
E474	WARMSV	=	*E474	WARM START ENTRY POINT
E477	COLDSV	=	*E477	COLD START ENTRY POINT
E47A	RBLOKV	=	*E47A	CASSETTE READ BLOCK ENTRY POINT VECTOR
E47D	CSOPIV	=	*E47D	CASSETTE OPEN FOR INPUT VECTOR
E480	VCTABL	=	*E480	

; OPERATING SYSTEM EQUATES

; COMMAND CODES FOR IOCB

0003	OPEN	=	3	OPEN FOR INPUT/OUTPUT
0005	GETREC	=	5	GET RECORD (TEXT)
0007	GETCHR	=	7	GET CHARACTER(S)
0009	PUTREC	=	9	PUT RECORD (TEXT)
000B	PUTCHR	=	0B	PUT CHARACTER(S)
000C	CLOSE	=	0C	CLOSE DEVICE
000D	STATIS	=	0D	STATUS REQUEST
000E	SPECIL	=	0E	BEGINNING OF SPECIAL ENTRY COMMANDS

; SPECIAL ENTRY COMMANDS

0011	DRAWLN	=	11	DRAW LINE
0012	FILLIN	=	12	DRAW LINE WITH RIGHT FILL
0020	RENAME	=	20	RENAME DISK FILE
0021	DELETE	=	21	DELETE DISK FILE
0022	FORMAT	=	22	FORMAT
0023	LOCKFL	=	23	LOCK FILE TO READ ONLY
0024	UNLOCK	=	24	UNLOCK LOCKED FILE
0025	POINT	=	25	POINT SECTOR
0026	NOTE	=	26	NOTE SECTOR
00FF	IOCFRE	=	FF	IOCB "FREE"

; AUX1 EQUATES

; () INDICATES WHICH DEVICES USE BIT

0001	APPEND	=	01	OPEN FOR WRITE APPEND (D), OR SCREEN READ (E)
0002	DIRECT	=	02	OPEN FOR DIRECTORY ACCESS (D)
0004	OPNIN	=	04	OPEN FOR INPUT (ALL DEVICES)
0008	OPNOT	=	08	OPEN FOR OUTPUT (ALL DEVICES)
000C	OPNINO	=	OPNIN+OPNOT	OPEN FOR INPUT AND OUTPUT (ALL DEVICES)
0010	HXDMOD	=	10	OPEN FOR FIXED MODE (E, S)
0020	INBCLR	=	20	OPEN WITHOUT CLEARING SCREEN (E, S)

; DEVICE NAMES

0045	SCREDIT	=	'E	SCREEN EDITOR (R/W)
0048	KBD	=	'K	KEYBOARD (R ONLY)
0053	DISPLY	=	'B	SCREEN DISPLY (R/W)
0050	PRINTR	=	'P	PRINTER (W ONLY)
0043	CASSET	=	'C	CASSETTE
004D	MODEM	=	'M	MODEM
0044	DISK	=	'D	DISK (R/W)

; SYSTEM EOL (CARRIAGE RETURN)

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

OPERATING SYSTEM STATUS CODES

0001	SUCCESS	= \$01	SUCCESSFUL OPERATION
0080	BRKABT	= \$80	BREAK KEY ABORT
0081	PRVOPN	= \$81	IOCB ALREADY OPEN
0082	NONDEV	= \$82	NON-EXISTANT DEVICE
0083	WRONLY	= \$83	IOCB OPENED FOR WRITE ONLY
0084	NVALID	= \$84	INVALID COMMAND
0085	NOTOPN	= \$85	DEVICE OR FILE NOT OPEN
0086	BADIOC	= \$86	INVALID IOCB NUMBER
0087	RDOONLY	= \$87	IOCB OPENED FOR READ ONLY
0088	EOFERR	= \$88	END OF FILE
0089	TRNRCD	= \$89	TRUNCATED RECORD
008A	TIMOUT	= \$8A	PERIPHERAL DEVICE TIME OUT
008B	DNACK	= \$8B	DEVICE DOES NOT ACKNOWLEDGE COMMAND
008C	FRMERR	= \$8C	SERIAL BUS FRAMING ERROR
008D	CHSROR	= \$8D	CURSOR OVERRANGE
008E	OVRUN	= \$8E	SERIAL BUS DATA OVERRUN
008F	CHKERR	= \$8F	SERIAL BUS CHECKSUM ERROR
0090	DERROR	= \$90	PERIPHERAL DEVICE ERROR (OPERATION NOT COMPLETED)
0091	BADMOD	= \$91	BAD SCREEN MODE NUMBER
0092	FNCTNOT	= \$92	FUNCTION NOT IMPLEMENTED IN HANDLER
0093	SCRMEM	= \$93	INSUFICIENT MEMORY FOR SCREEN MODE

PAGE ZERO RAM ASSIGNMENTS

0000	LINZBS	**\$0000	LINBUO RAM (WILL BE REPLACED BY MONITOR RAM)
------	--------	----------	--

THESE LOCATIONS ARE NOT CLEARED

0002	CASINI	**\$+2	CASSETTE INIT LOCATION
0004	RAPLO	**\$+2	RAM POINTER FOR MEMORY TEST
0006	TRAMSZ	**\$+1	TEMPORARY REGISTER FOR RAM SIZE
0007	TSTDAT	**\$+1	RAM TEST DATA REGISTER

CLEARED ON COLDSTART ONLY

0008	WARMST	**\$+1	WARM START FLAG
0009	BOOT?	**\$+1	SUCCESSFUL BOOT FLAG
000A	DOSVEC	**\$+2	DISK SOFTWARE START VECTOR
000C	DOSINI	**\$+2	DISK SOFTWARE INIT ADDRESS
000E	APPMHI	**\$+2	APPLICATIONS MEMORY HI LIMIT

CLEARED ON COLD OR WARM START

0010	INTZBS	**	INTERRUPT HANDLER
0010	POKMSK	**\$+1	SYSTEM MASK FOR POKEY IRQ ENABLE
0011	BRKKEY	**\$+1	BREAK KEY FLAG
0012	RYCLOW	**\$+3	REAL TIME CLOCK (IN 16 MSEC UNITS)
0015	BUFADR	**\$+2	INDIRECT BUFFER ADDRESS REGISTER
0017	ICCOMT	**\$+1	COMMAND FOR VECTOR
0018	DSKFMS	**\$+2	DISK FILE MANAGER POINTER

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

001A	DSKUTL	***+2	DISK UTILITIES POINTER
001C	PTIMOT	***+1	PRINTER TIME OUT REGISTER
001D	PBPNT	***+1	PRINT BUFFER POINTER
001E	PBUFSZ	***+1	PRINT BUFFER SIZE
001F	PTEMP	***+1	TEMPORARY REGISTER
0020	ZIOCB	==	ZERO PAGE I/O CONTROL BLOCK
0010	IOCBSZ	= 16	NUMBER OF BYTES PER IOCB
0080	MAXIOC	= 8*IOCBSZ	LENGTH OF THE IOCB AREA
0020	IOCBAS	==	
0020	ICHIDZ	***+1	HANDLER INDEX NUMBER (FF = IOCB FREE)
0021	ICDNOZ	***+1	DEVICE NUMBER (DRIVE NUMBER)
0022	ICCOMZ	***+1	COMMAND CODE
0023	ICSTAZ	***+1	STATUS OF LAST IOCB ACTION
0024	ICBALZ	***+1	BUFFER ADDRESS LOW BYTE
0025	ICBAHZ	***+1	
0026	ICPTLZ	***+1	PUT BYTE ROUTINE ADDRESS - 1
0027	ICPTHZ	***+1	
0028	ICBLZ	***+1	BUFFER LENGTH LOW BYTE
0029	ICBLHZ	***+1	
002A	ICAX1Z	***+1	AUXILIARY INFORMATION FIRST BYTE
002B	ICAX2Z	***+1	
002C	ICBPRZ	***+4	TWO SPARE BYTES (CIO LOCAL USE)
002E	ICIDNO	= ICBPRZ+2	IOCB NUMBER X 16
002F	CIOCHR	= ICBPRZ+3	CHARACTER BYTE FOR CURRENT OPERATION
0030	STATUS	***+1	INTERNAL STATUS STORAGE
0031	CHKSUM	***+1	CHECKSUM (SINGLE BYTE SUM WITH CARRY)
0032	BUFRLO	***+1	POINTER TO DATA BUFFER (LO BYTE)
0033	BUFRHI	***+1	POINTER TO DATA BUFFER (HI BYTE)
0034	BFENLO	***+1	NEXT BYTE PAST END OF THE DATA BUFFER (LO BYTE)
0035	BFENHI	***+1	NEXT BYTE PAST END OF THE DATA BUFFER (HI BYTE)
0036	CRETRY	***+1	NUMBER OF COMMAND FRAME RETRIES
0037	DRETRY	***+1	NUMBER OF DEVICE RETRIES
0038	BUFRFL	***+1	DATA BUFFER FULL FLAG
0039	RECDN	***+1	RECEIVE DONE FLAG
003A	XMTDN	***+1	TRANSMISSION DONE FLAG
003B	CHKSENT	***+1	CHECKSUM SENT FLAG
003C	NOCKSM	***+1	NO CHECKSUM FOLLOWS DATA FLAG
003D	BPTR	***+1	
003E	FTYPE	***+1	
003F	FEDF	***+1	
0040	FREQ	***+1	
0041	BOUNDR	***+1	NOISY I/O FLAG. (ZERO IS QUIET)
0042	CRITIC	***+1	DEFINES CRITICAL SECTION (CRITICAL IF NON-ZERO)
0043	FMSZPO	***+7	DISK FILE MANAGER SYSTEM ZERO PAGE
004A	CKEY	***+1	FLAG SET WHEN GAME START PRESSED
004B	CASSET	***+1	CASSETTE BOOT FLAG
004C	DSTAT	***+1	DISPLAY STATUS
004D	ATRACT	***+1	ATRACT FLAG
004E	DRKMSK	***+1	DARK ATRACT MASK
004F	COLORSH	***+1	ATRACT COLOR SHIFTER (EOR'ED WITH PLAYFIELD COLORS)

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

0002	LEDGE	= 2	LMARON'S VALUE AT COLD START
0027	REDGE	= 39	RMARON'S VALUE AT COLD START
0050	THPCR	==+1	
0051	HOLD1	==+1	
0052	LMARON	==+1	LEFT MARGIN (SET TO 1 AT POWER ON)
0053	RMARON	==+1	RIGHT MARGIN (SET TO 38 AT POWER ON)
0054	ROWCR8	==+1	CURSOR COUNTERS
0055	COLCR8	==+2	
0057	DINDEX	==+1	
0058	SAVMS	==+2	
005A	OLDROW	==+1	
005B	OLDCOL	==+2	
005D	OLDCHR	==+1	DATA UNDER CURSOR
005E	OLDADR	==+2	
0060	NEWROW	==+1	POINT DRAW GOES TO
0061	NEWCOL	==+2	
0063	LOGCOL	==+1	POINTS AT COLUMN IN LOGICAL LINE
0064	ADRESS	==+2	
0066	MLTTP	==+2	
0066	OPNTMP	= MLTTP	FIRST BYTE IS USED IN OPEN AS TEMP
0068	SAVADR	==+2	
006A	RAMTOP	==+1	RAM SIZE DEFINED BY POWER ON LOGIC
006B	BUFCNT	==+1	BUFFER COUNT
006C	BUFBTR	==+2	EDITOR GETCH POINTER
006E	BITMSK	==+1	BIT MASK
006F	SHFAMT	==+1	
0070	ROWAC	==+2	
0072	COLAC	==+2	
0074	ENDPT	==+2	
0076	DELTAR	==+1	
0077	DELTAC	==+2	
0079	ROWINC	==+1	
007A	COLINC	==+1	
007B	SWPFL0	==+1	NON-0 IF TXT AND REGULAR RAM IS SWAPPED
007C	HOLDCH	==+1	CH IS MOVED HERE IN KETCH BEFORE CNTL & SHIFT PROC
007D	INSDAT	==+1	
007E	COUNTR	==+2	

80 - FF ARE RESERVED FOR USER APPLICATIONS

NOTE : SEE FLOATING POINT SUBROUTINE AREA FOR ZERO PAGE CELLS

PAGE 1 - STACK

PAGE TWO RAM ASSIGNMENTS

0200	INTABS	==	INTERRUPT RAM
0200	VDBLBT	==+2	DISPLAY LIST NMI VECTOR

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

0202	VPRCED	***+2	PROCEED LINE IRQ VECTOR
0204	VINTER	***+2	INTERRUPT LINE IRQ VECTOR
0206	VBREAK	***+2	SOFTWARE BREAK (00) INSTRUCTION IRQ VECTOR
0208	VKEYBD	***+2	POKEY KEYBOARD IRQ VECTOR
020A	VBERIN	***+2	POKEY SERIAL INPUT READY IRQ
020C	VBEROR	***+2	POKEY SERIAL OUTPUT READY IRQ
020E	VSEIOC	***+2	POKEY SERIAL OUTPUT COMPLETE IRQ
0210	VTIMR1	***+2	POKEY TIMER 1 IRQ
0212	VTIMR2	***+2	POKEY TIMER 2 IRQ
0214	VTIMR4	***+2	POKEY TIMER 4 IRQ
0216	VIMIRG	***+2	IMMEDIATE IRQ VECTOR
0218	CDTMV1	***+2	COUNT DOWN TIMER 1
021A	CDTMV2	***+2	COUNT DOWN TIMER 2
021C	CDTMV3	***+2	COUNT DOWN TIMER 3
021E	CDTMV4	***+2	COUNT DOWN TIMER 4
0220	CDTMV5	***+2	COUNT DOWN TIMER 5
0222	VVBLKI	***+2	IMMEDIATE VERTICAL BLANK NMI VECTOR
0224	VVBLKD	***+2	DEFERRED VERTICAL BLANK NMI VECTOR
0226	CDTMA1	***+2	COUNT DOWN TIMER 1 JBR ADDRESS
0228	CDTMA2	***+2	COUNT DOWN TIMER 2 JBR ADDRESS
022A	CDTMF3	***+1	COUNT DOWN TIMER 3 FLAG
022B	BRTMR	***+1	SOFTWARE REPEAT TIMER
022C	CDTMF4	***+1	COUNT DOWN TIMER 4 FLAG
022D	INTMP	***+1	IAN'S TEMP (RENAMED FROM T1 BY POPULAR DEMAND)
022E	CDTMF5	***+1	COUNT DOWN TIMER FLAG 5
022F	SDMCTL	***+1	SAVE DMACTL REGISTER
0230	BDLSTL	***+1	SAVE DISPLAY LIST LOW BYTE
	BDLBTH	***+1	SAVE DISPLAY LIST HI BYTE
	BSKCTL	***+1	SKCTL REGISTER RAM
		***+1	
	LPENH	***+1	LIGHT PEN HORIZONTAL VALUE
	LPENV	***+1	LIGHT PEN VERTICAL VALUE
		***+2	SPARE BYTES
		***+2	SPARE
023A	CDEVIC	***+1	COMMAND FRAME BUFFER - DEVICE
023B	CCOMND	***+1	COMMAND
023C	CAUX1	***+1	COMMAND AUX BYTE 1
023D	CAUX2	***+1	COMMAND AUX BYTE 2
023E	TEMP	***+1	NOTE: MAY NOT BE THE LAST WORD ON A PAGE TEMPORARY RAM CELL NOTE: MAY NOT BE THE LAST WORD ON A PAGE
023F	ERRFLO	***+1	ERROR FLAG - ANY DEVICE ERROR EXCEPT TIME OUT
0240	DFLAG8	***+1	DISK FLAG8 FROM SECTOR ONE
0241	DBBECT	***+1	NUMBER OF DISK BOOT SECTORS
0242	BOOTAD	***+2	ADDRESS WHERE DISK BOOT LOADER WILL BE PUT
0244	COLDST	***+1	COLDSTART FLAG (1=IN MIDDLE OF COLDSTART)
		***+1	SPARE
0246	DSKTIH	***+1	DISK TIME OUT REGISTER
0247	LINBUF	***+40	CHAR LINE BUFFER
026F	OPRIOR	***+1	GLOBAL PRIORITY CELL
0270	PADDLO	***+1	POTENTIOMETER 0 RAM CELL

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

0271	PADDL1	****1	
0272	PADDL2	****1	
0273	PADDL3	****1	
0274	PADDL4	****1	
0275	PADDL5	****1	
0276	PADDL6	****1	
0277	PADDL7	****1	
0278	STICK0	****1	JOYSTICK 0 RAM CELL
0279	STICK1	****1	
027A	STICK2	****1	
027B	STICK3	****1	

027C	PTRIO0	****1	PADDLE TRIGGER 0
027D	PTRIO1	****1	
027E	PTRIO2	****1	
027F	PTRIO3	****1	
0280	PTRIO4	****1	
0281	PTRIO5	****1	

0282	PTRIO6	****1	
0283	PTRIO7	****1	
0284	STRIO0	****1	JOYSTICK TRIGGER 0
0285	STRIO1	****1	
0286	STRIO2	****1	
0287	STRIO3	****1	

0288	CBSTAT	****1	
0289	WMODE	****1	
028A	BLIM	****1	

****5 SPARE

0290	TXTR0W	****1	TEXT ROWCRB
0291	TXTCOL	****2	TEXT COLCRB
0293	TINDEX	****1	TEXT INDEX
0294	TXTMBC	****2	FOOLS CONVRT INTO NEW MBC
0296	TXTOLD	****6	OLDROW & OLD COL FOR TEXT (AND THEN SOME)

029C	TMPIX1	****1	
029D	HOLD3	****1	
029E	SUBTMP	****1	

029F	HOLD2	****1	
02A0	DMASK	****1	
02A1	TMPLBT	****1	
02A2	ESCFLO	****1	ESCAPE FLAG

02A3	TABMAP	****15	
02B2	LOOMAP	****4	LOGICAL LINE START BIT MAP
02B6	INVFL0	****1	INVERSE VIDED FLAG (TOGGLED BY ATARI KEY)
02B7	FILFLO	****1	RIGHT FILL FLAG FOR DRAW

02B8	TMPCOL	****2	
02B8	SCRFL0	****1	SET IF SCROLL OCCURS
02BC	HOLD4	****1	TEMP CELL USED IN DRAW ONLY
02BD	HOLD5	****1	DITTO

02BE	SHFLOK	****1	
02BF	BOTSCR	****1	BOTTOM OF SCREEN : 24 NORM 4 SPLIT

02C0	PCOLR0	****1	P0 COLOR
02C1	PCOLR1	****1	P1 COLOR

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

02C2	PCOLR2	***+1	P2 COLOR
02C3	PCOLR3	***+1	P3 COLOR
02C4	COLOR0	***+1	COLOR 0
02C5	COLOR1	***+1	
02C6	COLOR2	***+1	
02C7	COLOR3	***+1	
02C8	COLOR4	***+1	

***+23 SPARE

02E0	GLDABS	**	GLOBAL VARIABLES
------	--------	----	------------------

***+4 SPARE

02E4	RAMBIZ	***+1	RAM SIZE (HI BYTE ONLY)
02E5	MEMTOP	***+2	TOP OF AVAILABLE USER MEMORY
02E7	MEMLO	***+2	BOTTOM OF AVAILABLE USER MEMORY

02EA	DVSTAT	***+4	STATUS BUFFER
02EE	CBAUDL	***+1	CASSETTE BAUD RATE LOW BYTE
02EF	CBAUDH	***+1	

02F0	CRBINH	***+1	CURSOR INHIBIT (00 = CURSOR ON)
02F1	KEYDEL	***+1	KEY DELAY
02F2	CHI	***+1	

02F3	CHACT	***+1	CHACTL REGISTER RAM
02F4	CHBAS	***+1	CHBAS REGISTER RAM

***+5 SPARE BYTES

02FA	CHAR	***+1	
02FB	ATACHR	***+1	ATASCII CHARACTER
02FC	CH	***+1	GLOBAL VARIABLE FOR KEYBOARD
02FD	FILDAT	***+1	RIGHT FILL DATA (DRAW)
02FE	DSPFLO	***+1	DISPLAY FLAG: DISPLAY CNTLS IF NON-ZERO
02FF	BSFLAQ	***+1	START/STOP FLAG FOR PAGING (CNTL 1). CLEARED BY BREAK

PAGE THREE RAM ASSIGNMENTS

0300	DCB	**	DEVICE CONTROL BLOCK
0300	DDEVIC	***+1	PERIPHERAL UNIT 1 BUS I.D. NUMBER
0301	DUNIT	***+1	UNIT NUMBER
0302	DCOMND	***+1	BUS COMMAND
0303	DBTATS	***+1	COMMAND TYPE/STATUS RETURN
0304	DBUFLO	***+1	DATA BUFFER POINTER LOW BYTE
0305	DBUFHI	***+1	
0306	DTIMLO	***+1	DEVICE TIME OUT IN 1 SECOND UNITS
0307	DUNUSE	***+1	UNUSED BYTE
0308	DBYTLO	***+1	NUMBER OF BYTES TO BE TRANSFERRED LOW BYTE
0309	DBYTHI	***+1	

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

030A	DAUX1	***1	COMMAND AUXILIARY BYTE 1
030B	DAUX2	***1	
030C	TIMER1	***2	INITIAL TIMER VALUE
030E	ADDCOR	***1	ADDITION CORRECTION
030F	CASFL0	***1	CASSETTE MODE WHEN SET
0310	TIMER2	***2	FINAL TIMER VALUE. THESE TWO TIMER VALUES ARE USED TO COMPUTE INTERVAL FOR BAUD RATE
0312	TEMP1	***2	TEMPORARY STORAGE REGISTER
0314	TEMP2	***1	TEMPORARY STORAGE REGISTER
0315	TEMP3	***1	TEMPORARY STORAGE REGISTER
0316	SAVIO	***1	SAVE SERIAL IN DATA PORT
0317	TIMFLO	***1	TIME OUT FLAG FOR BAUD RATE CORRECTION
0318	STACKP	***1	810 STACK POINTER SAVE CELL
0319	TSTAT	***1	TEMPORARY STATUS HOLDER

031A	HATABS	***38	HANDLER ADDRESS TABLE
0021	MAXDEV	= *-HATABS-5	MAXIMUM HANDLER ADDRESS INDEX

NOTE : THE ENTIRE IOCB DEFINITIONS HAVE BEEN MODIFIED

0340	IOCB	***	I/O CONTROL BLOCKS
0340	ICHID	***1	HANDLER INDEX NUMBER (FF = IOCB FREE)
0341	ICDNO	***1	DEVICE NUMBER (DRIVE NUMBER)
0342	ICCOM	***1	COMMAND CODE
0343	ICSTA	***1	STATUS OF LAST IOCB ACTION
0344	ICBAL	***1	BUFFER ADDRESS LOW BYTE
0345	ICBAH	***1	
0346	ICPTL	***1	PUT BYTE ROUTINE ADDRESS - 1
0347	ICPTH	***1	
0348	ICBLL	***1	BUFFER LENGTH LOW BYTE
0349	ICBLH	***1	
034A	ICAX1	***1	AUXILIARY INFORMATION FIRST BYTE
034B	ICAX2	***1	
034C	ICBPR	***4	FOUR SPARE BYTES
		***MAXIOC-IOCB92	

03C0	PRNBUF	***40	PRINTER BUFFER
------	--------	-------	----------------

***21 SPARE BYTES

PAGE FOUR RAM ASSIGNMENTS

03FD	CASBUF	***131	CASSETTE BUFFER
------	--------	--------	-----------------

0480	USAREA	***128	SPARE
------	--------	--------	-------

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

0006	FPREC	=	6	! FLOATING PT PRECISION (# OF BYTES),
				IF CARRY USED THEN CARRY CLEAR => NO ERROR, CARRY SET => ERROR
D800	AFP	=	\$D800	! ASCII->FLOATING POINT (FP)
				INBUFF+CIX -> FRO, CIX, CARRY
D8E6	FASC	=	\$D8E6	! FP -> ASCII FRO-> LBUFF (INBUFF)
D9AA	IFP	=	\$D9AA	! INTEGER -> FP
				0-FFFF (LBB,MSB) IN FRO, FRO+1->FRO
D9D2	FFI	=	\$D9D2	! FP -> INTEGER FRO -> FRO, FRO+1, CARRY
DA60	FBUB	=	\$DA60	! FRO <- FRO - FRI, CARRY
DA66	FADD	=	\$DA66	! FRO <- FRO + FRI, CARRY
DADB	FMUL	=	\$DADB	! FRO <- FRO * FRI, CARRY
DB28	FDIV	=	\$DB28	! FRO <- FRO / FRI, CARRY
DD89	FLDOR	=	\$DD89	! FLOATING LOAD REG0 FRO <- (X,Y)
DD8D	FLDOP	=	\$DD8D	! " " " FRO <- (FLPTR)
DD98	FLDIR	=	\$DD98	! " " " REG1 FRI <- (X,Y)
DD9C	FLDIP	=	\$DD9C	! " " " FRI <- (FLPTR)
DDA7	FSTOR	=	\$DDA7	! FLOATING STORE REG0 (X,Y) <- FRO
DDAB	FSTOP	=	\$DDAB	! " " " (FLPTR) <- FRO
DD86	FMOVE	=	\$DD86	! FRI <- FRO
DD40	PLYEVL	=	\$DD40	! FRO <- P(Z) = SUM(I=N TO 0) (A(I)*Z**I) CARRY
				INPUT: (X,Y) = A(N), A(N-1)...A(0) -> PLYARG
				ACC = # OF COEFFICIENTS = DEGREE+1
				FRO = Z
DDC0	EXP	=	\$DDC0	! FRO <- E**FRO = EXP10(FRO * LOG10(E)) CARRY
DDCC	EXP10	=	\$DDCC	! FRO <- 10**FRO CARRY
DECD	LD0	=	\$DECD	! FRO <- LN(FRO) = LOG10(FRO)/LOG10(E) CARRY
DED1	LOG10	=	\$DED1	! FRO <- LOG10(FRO) CARRY

THE FOLLOWING ARE IN BASIC CARTRIDGE:

BD81	SIN	=	\$BD81	! FRO <- SIN(FRO) DEG/100 => RAD, DEG => DEG. CARRY
BD73	COS	=	\$BD73	! FRO <- COS(FRO) CARRY
BE43	ATAN	=	\$BE43	! FRO <- ATAN(FRO) CARRY
BE81	SQR	=	\$BE81	! FRO <- SQUAREROOT(FRO) CARRY

! FLOATING POINT ROUTINES ZERO PAGE (NEEDED ONLY IF F.P. ROUTINES ARE CALLED)

00D4	FRO	== \$D4	! FP REG0
00DA	FRE	== \$D4	! FP REG0
00E0	FRI	== \$D4	! FP REG1
00E6	FR2	== \$D4	! FP REG1
00EC	FRX	== \$D4	! FP SPARE

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

00ED	EEXP	+++1	VALUE OF E
00EE	NBION	+++1	SION OF #
00EF	EBION	+++1	SION OF EXPONENT
00F0	FCHRFLO	+++1	1ST CHAR FLAG
00F1	DICRT	+++1	# OF DIGITS RIGHT OF DECIMAL
00F2	CIX	+++1	CURRENT INPUT INDEX
00F3	INBUFF	+++2	POINTS TO USER'S LINE INPUT BUFFER
00F5	ZTEMP1	+++2	
00F7	ZTEMP4	+++2	
00F9	ZTEMP3	+++2	
00FB	DEOFLO		
00FB	RADFLO	+++1	0=RADIANS, 5=DEGREES
0000	RADON	= 0	INDICATES RADIANS
0006	DEON	= 6	INDICATES DEGREES
00FC	FLPTR	+++2	POINTS TO USER'S FLOATING PT NUMBER
00FE	FPTR2	+++2	

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

		==057E	
057E	LBPR1	+++1	LBUFF PREFIX 1
057F	LBPR2	+++1	LBUFF PREFIX 2
0580	LBUFF	+++128	LINE BUFFER
05E0	PLYAR0	=	LBUFF+060 POLYNOMIAL ARGUMENTS
05E5	FPBCR	=	PLYAR0+FPREC
05EC	FPBCR1	=	FPBCR+FPREC
05E6	FBCR	=	FPBCR
05EC	FBCR1	=	FPBCR1
05FF	LBFEND	=	-1 END OF LBUFF

COLLEEN MNEMONICS

			POKEY	=	#D200	VBLANK ACTION:	DESCRIPTION:
D200	POTO	=	POKEY+0		POTO-->PADDL0	0-227 IN RAM CELL	
D201	POT1	=	POKEY+1		POT1-->PADDL1	0-227 IN RAM CELL	
D202	POT2	=	POKEY+2		POT2-->PADDL2	0-227 IN RAM CELL	
D203	POT3	=	POKEY+3		POT3-->PADDL3	0-227 IN RAM CELL	
D204	POT4	=	POKEY+4		POT4-->PADDL4	0-227 IN RAM CELL	
D205	POT5	=	POKEY+5		POT5-->PADDL5	0-227 IN RAM CELL	
D206	POT6	=	POKEY+6		POT6-->PADDL6	0-227 IN RAM CELL	
D207	POT7	=	POKEY+7		POT7-->PADDL7	0-227 IN RAM CELL	
D208	ALLPOT	=	POKEY+8		???		
D209	KBCODE	=	POKEY+9				
D20A	RANDOM	=	POKEY+10				
D20B	POT00	=	POKEY+11		STROBED		
D20C	SECTN	=	POKEY+12				

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

D205	AUDC3	=	POKEY+5		
D206	AUDF4	=	POKEY+6		
D207	AUDC4	=	POKEY+7		
D208	AUDCTL	=	POKEY+8	NONE	AUDCTL<--(BIO)
D209	STIMER	=	POKEY+9		
D20A	SKREB	=	POKEY+10	NONE	SKREB<--(BIO)
D20D	SEROUT	=	POKEY+13	NONE	SEROUT<--(BIO)
D20E	IRGEN	=	POKEY+14	POKMBK-->IRGEN	(AFFECTED BY OPEN B: OR E:)
D20F	SKCTL	=	POKEY+15	BBKCTL-->SKCTL	BSKCTL<--(BIO)

			\$D000	VBLANK ACTION:	DESCRIPTION:
--	--	--	--------	----------------	--------------

D000	HPOSP0	=	CTIA+0		
D001	HPOBP1	=	CTIA+1		
D002	HPOSP2	=	CTIA+2		
D003	HPOSP3	=	CTIA+3		
D004	HPOSM0	=	CTIA+4		
D005	HPOSH1	=	CTIA+5		
D006	HPOSM2	=	CTIA+6		
D007	HPOBM3	=	CTIA+7		
D008	SIZEP0	=	CTIA+8		
D009	SIZEP1	=	CTIA+9		
D00A	SIZEP2	=	CTIA+10		
D00B	SIZEP3	=	CTIA+11		
D00C	SIZEH	=	CTIA+12		
D00D	GRAFF0	=	CTIA+13		
D00E	GRAFF1	=	CTIA+14		
D00F	GRAFF2	=	CTIA+15		
D010	GRAFF3	=	CTIA+16		
D011	GRAFH	=	CTIA+17		

D012	COLPH0	=	CTIA+18	PCOLR0-->COLPH0	WITH ATTRACT MODE
D013	COLPH1	=	CTIA+19	PCOLR1-->COLPH1	WITH ATTRACT MODE
D014	COLPH2	=	CTIA+20	PCOLR2-->COLPH2	WITH ATTRACT MODE
D015	COLPH3	=	CTIA+21	PCOLR3-->COLPH3	WITH ATTRACT MODE
D016	COLPF0	=	CTIA+22	COLOR0-->COLPF0	WITH ATTRACT MODE
D017	COLPF1	=	CTIA+23	COLOR1-->COLPF1	WITH ATTRACT MODE
D018	COLPF2	=	CTIA+24	COLOR2-->COLPF2	WITH ATTRACT MODE
D019	COLPF3	=	CTIA+25	COLOR3-->COLPF3	WITH ATTRACT MODE
D01A	COLBK	=	CTIA+26	COLOR4-->COLBK	WITH ATTRACT MODE
D01B	PRIOR	=	CTIA+27	(ON OPEN B: OR E:)	OPRIOR-->PRIOR

D01C	VDELAY	=	CTIA+28		
D01D	ORACTL	=	CTIA+29		
D01E	INTCLR	=	CTIA+30		
D01F	CUNSCN	=	CTIA+31	*OR-->CONSOL	TURN OFF SPEAKER

D020	HOPE	=	CTIA+0		
D021	H1PF	=	CTIA+1		
D022	H2PF	=	CTIA+2		
D023	H3PF	=	CTIA+3		
D024	POPF	=	CTIA+4		
D025	P1PF	=	CTIA+5		
D026	P2PF	=	CTIA+6		
D027	P3PF	=	CTIA+7		
D028	NOPL	=	CTIA+8		
D029	H1PL	=	CTIA+9		
D02A	H2PL	=	CTIA+10		
D02B	H3PL	=	CTIA+11		
D02C	P1PL	=	CTIA+12		
D02D	P2PL	=	CTIA+13		
D02E	P3PL	=	CTIA+14		
D02F	INTG	=	CTIA+15		

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

D011	TRI01	=	CTIA+17	TRI01-->BTRI01	
D012	TRI02	=	CTIA+18	TRI02-->BTRI02	
D013	TRI03	=	CTIA+19	TRI03-->BTRI03	
D400	ANTIC	=	#D400	VBLANK ACTION	DESCRIPTION
D400	DMACTL	=	ANTIC+0	DMACTL<--BDMCTL	ON OPEN 8: OR E:
D401	CHACTL	=	ANTIC+1	CHACTL<--CHACT	ON OPEN 8: OR E:
D402	DLISTL	=	ANTIC+2	DLISTL<--BDLSTL	ON OPEN 8: OR E:
D403	DLISTH	=	ANTIC+3	DLISTH<--BDLSTH	ON OPEN 8: OR E:
D404	HSCROL	=	ANTIC+4		
D405	VSCROL	=	ANTIC+5		
D407	PHBASE	=	ANTIC+7	CHBASE<--CHBAS	ON OPEN 8: OR E:
D409	CHBASE	=	ANTIC+9		
D40A	WSYNC	=	ANTIC+10		
D40B	VCOUNT	=	ANTIC+11		
D40C	PENH	=	ANTIC+12		
D40D	PENV	=	ANTIC+13		
D40E	NMIEN	=	ANTIC+14	NMIEN<--40	POWER ON AND (BETVBV)
D40F	NMIRES	=	ANTIC+15	STROBED	
D40F	NMIST	=	ANTIC+15		
D300	PIA	=	#D300	VBLANK ACTION	DESCRIPTION
D300	PORTA	=	PIA+0	PORTA-->BTICK0.1	X-Y CONTROLLERS
D301	PORTB	=	PIA+1	PORTB-->BTICK2.3	X-Y CONTROLLERS
D302	PACTL	=	PIA+2	NONE	PACTL<--3C [INIT]
D303	PBCTL	=	PIA+3	NONE	PBCTL<--3C [INIT]

END

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

TITLE "CENTRAL INPUT/OUTPUT (CIO) 2-7-79"
UPDATED BY AL MILLER 3-9-79

0030
003A
009B

ABCZER = '0
COLON = \$3A
EOL = \$9B

ASCII ZERO
ASCII COLON
END OF RECORD

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

; CIO JUMP VECTOR FOR USERS

**=CIOV

E456 4C C4 E4

JMP

CIO

GO TO CIO

; CIO INIT JUMP VECTOR FOR POWER UP

**=CIOINV

E46E 4C A6 E4

JMP

CIOINT

GO TO INIT

; ERROR ROUTINE ADDRESS EQUATE

00E4

ERRTNH=ERRTN/256

00C0

ERRTNL=-ERRTNH*256+ERRTN

**=CIOOR0

; CIO INITIALIZATION (CALLED BY MONITOR AT POWER UP)

E4A6 A2 00

CIOINT

LDX

#0

E4A8 A9 FF

CIOI1

LDA

#IOCFRE

SET ALL IOCB'S TO FREE

E4AA 9D 40 03

STA

ICID,X

BY SETTING HANDLER ID'S=FF

E4AD A9 C0

LDA

#ERRTNL

E4AF 9D 46 03

STA

ICPTL,X

POINT PUT TO ERROR ROUTINE

E4B2 A9 E4

LDA

#ERRTNH

E4B4 9D 47 03

STA

ICPTH,X

E4B7 8A

TXA

E4B8 1B

CLC

E4B9 69 10

ADC

#IOCBSZ

BUMP INDEX BY SIZE

E4BB AA

TAX

E4BC C9 B0

CMP

#MAXIOC

DONE?

E4BE 90 E8

BCC

CIOI1

NO

E4C0 60

RTS

YES, RETURN

; ERROR ROUTINE FOR ILLEGAL PUT

ERRTN=-1

E4C0

E4C1 AD B5

LDY

#NOTOPN

IOCB NOT OPEN

E4C3 60

RTS

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

002C

CIO LOCAL RAM (USES SPARE BYTES IN ZERO PAGE IOCB)
ENTVEC = ICBPRZ

CIO MAIN ROUTINE

E4C4 85 2F

CIO INTERFACES BETWEEN USER AND INPUT/OUTPUT DE
CIO BTA CIOCHR SAVE POSSIBLE OUTPUT CHARACTER
E4C6 86 2E BTX ICIDNO SAVE IOCB NUMBER * N

CHECK FOR LEGAL IOCB

E4C8 8A

TXA

E4C9 29 0F

AND

#6F

IS IOCB MULTIPLE OF 16?

E4CB D0 04

BNE

CIERR1

NO, ERROR

E4CD E0 80

CPX

#MAXIOC

IS INDEX TOO LARGE?

E4CF 90 05

BCC

IOC1

NO

INVALID IOCB NUMBER -- RETURN ERROR

E4D1 A0 B6

CIERR1

LDY

#BADIOC

ERROR CODE

E4D3 4C 1B E6

JMP

CIRTN1

RETURN

MOVE USER IOCB TO ZERO PAGE

E4D6 A0 00

IOC1

LDY

#0

E4D8 BD 40 03

IOC1A

LDA

IOCB, X

USER IOCB

E4DB 99 20 00

STA

IOCBAB, Y

TO ZERO PAGE

E4DE E8

INX

E4DF CB

INY

E4E0 C0 0C

CPY

#12

12 BYTES

E4E2 90 F4

BCC

IOC1A

COMPUTE CIO INTERNAL VECTOR FOR COMMAND

E4E4 A0 B4

LDY

#INVALID

ASSUME INVALID CODE

E4E6 A5 22

LDA

ICCDM2

COMMAND CODE TO INDEX

E4E8 C9 03

CMP

#OPEN

IS COMMAND LEGAL?

E4EA 90 25

BCC

CIERR4

NO

E4EC AB

TAY

MOVE COMMAND TO ZERO BASE FOR INDEX

E4ED C0 0E

CPY

#SPECIL

IS COMMAND SPECIAL?

E4EF 90 02

BCC

IOC2

NO

E4F1 A0 0E

LDY

#SPECIL

YES, SET SPECIAL OFFSET INDEX

E4F3 84 17

IOC2

STY

ICCOMT

SAVE COMMAND FOR VECTOR

E4F5 B9 C6 E6

LDA

COMTAB-3, Y

GET VECTOR OFFSET FROM TABLE

E4F8 F0 0F

BEQ

CIOPEN

00 IF OPEN COMMAND

E4FA C9 02

CMP

#2

IS IT CLOSE?

E4FC F0 39

BEQ

CICLOS

YES

E4FE C9 08

CMP

#8

IS IT STATUS OR SPECIAL?

E500 B0 4C

BCB

CIBTSP

YES

E502 C9 04

CMP

#4

IS IT READ?

E504 F0 63

BEQ

CIREAD

YES

E506 4C C9 E5

JMP

CIWRIT

ELSE, MUST BE WRITE

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

OPEN COMMAND

FIND DEVICE HANDLER IN HANDLER ADDRESS TABLE

E509 A5 20	CIOPEN	LDA	ICHIDZ	GET HANDLER ID
E50B C9 FF		CHP	#IOC FRE	IS THIS IOCB CLOSED?
E50D F0 05		BEQ	IOC6	YES

ERROR -- IOCB ALREADY OPEN

E50F A0 B1	CIERR3	LDY	#PRVOPN	ERROR CODE
E511 4C 1B E6	CIERR4	JMP	CIRTN1	RETURN

GO FIND DEVICE

E514 20 9E E6	IOC6	JBR	DEVBRG	CALL DEVICE SEARCH
E517 B0 FB		BCB	CIERR4	GO IF DEVICE NOT FOUND

DEVICE FOUND, INITIALIZE IOCB FOR OPEN

COMPUTE HANDLER ENTRY POINT

E519 20 3D E6	IOC7	JBR	COMENT	
E51C B0 F3		BCB	CIERR4	GO IF ERROR IN COMPUTE

GO TO HANDLER FOR INITIALIZATION

E51E 20 B9 E6		JBR	GORAND	USE INDIRECT JUMP
---------------	--	-----	--------	-------------------

STORE PUT BYTE ADDRESS-1 INTO IOCB

E521 A9 0B		LDA	#PUTCHR	SIMULATE PUT CHARACTER
E523 B5 17		STA	ICCOMT	
E525 20 3D E6		JBR	COMENT	COMPUTE ENTRY POINT
E528 A5 2C		LDA	ICSPRZ	MOVE COMPUTED VALUE
E52A B5 26		STA	ICPTLZ	TO PUT BYTE ADDRESS
E52C A5 2D		LDA	ICSPRZ+1	
E52E B5 27		STA	ICPTHZ	
E530 4C 1D E6		JMP	CIRTN2	RETURN TO USER

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

```

      ; CLOSE COMMAND
E533 A0 01      C1CLOS LDY      #SUCCEB      ASSUME GOOD CLOSE
E535 B4 23      BTY      ICSTAZ
      ; E6
      ; 03
      ; B9 E6
      ; 07 FF      C1CLOS LDA      #IOCFRE
E541 B5 20      STA      ICHIDZ      SET HANDLER ID
E543 A9 E4      LDA      #ERRTNH
E545 B5 27      STA      ICPHIZ      SET PUT BYTE TO POINT TO ERROR
E547 A9 C0      LDA      #ERRTNL
E549 B5 26      STA      ICPTLZ
E54B 4C 1D E6    JMP      C1RTN2      RETURN

```

```

      ; STATUS AND SPECIAL REQUESTS
      ; DO IMPLIED OPEN IF NECESSARY AND GO TO DEVICE
E54E A5 20      C1STSP LDA      ICHIDZ      IS THERE A HANDLER ID?
E550 C9 FF      CMP      #IOCFRE
E552 D0 05      BNE      C1BT1      YES

```

```

      ; IOCB IS FREE, DO IMPLIED OPEN
E554 20 9E E6    JBR      DEVBRG      FIND DEVICE IN TABLE
E557 B0 BB      BCS      CIERR4      GO IF ERROR IN COMPUTE

```

```

      ; COMPUTE AND GO TO ENTRY POINT IN HANDLER
E559 20 3D E6    C1BT1 JBR      COMENT      COMPUTER HANDLER ENTRY VECTOR
E55C 20 B9 E6    JBR      ODHAND      GO TO HANDLER

```

```

      ; RESTORE HANDLER INDEX (DO IMPLIED CLOSE)
E55F A6 2E      LDX      ICIDNO      IOCB INDEX
E561 BD 40 03    LDA      ICHID.X     GET ORIGINAL HANDLER ID
E564 B5 20      STA      ICHIDZ      RESTORE ZERO PAGE
E566 4C 1D E6    JMP      C1RTN2      RETURN

```

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

E569 A5 22	READ -- DO GET COMMANDS		
E56B 25 2A	CIREAD LDA	ICCOMZ	GET COMMAND BYTE
E56D 00 05	AND	ICAXIZ	IS THIS READ LEGAL?
	BNE	RCI1A	YES

E56F A0 B3	ILLEGAL READ -- IOCB OPENED FOR WRITE ONLY		
E571 4C 1B E6	LDY	#WRONLY	ERROR CODE
	RCI1B JMP	CIRTNI	RETURN

E574 20 3D E6	COMPUTE AND CHECK ENTRY POINT		
E577 B0 F8	RCI1A JSR	COMENT	COMPUTE ENTRY POINT
	BCB	RCI1B	GO IF ERROR IN COMPUTE

E579 A5 28	GET RECORD OR CHARACTER		
E57B 05 29	LDA	ICBLLZ	
E57D 00 08	ORA	ICBLLZ+1	IS BUFFER LENGTH ZERO?
E57F 20 B9 E6	BNE	RCI3	NO
E582 B5 2F	JSR	QOHAND	
E584 4C 1D E6	BTA	CIOCHR	
	JMP	CIRTN2	

E587 20 B9 E6	LOOP TO FILL BUFFER OR END RECORD		
E58A B5 2F	RCI3 JSR	QOHAND	GO TO HANDLER TO GET BYTE
E58C 30 33	BTA	CIOCHR	SAVE BYTE
E58E A0 00	BMI	RCI4	END TRANSFER IF ERROR
E590 91 24	LDY	#0	
E592 20 70 E6	BTA	(ICBALZ),Y	PUT BYTE IN USER BUFFER
E595 A5 22	JSR	INCBFP	INCREMENT BUFFER POINTER
E597 29 02	LDA	ICCOMZ	GET COMMAND CODE
E599 00 0C	AND	#2	IS IT GET RECORD?
	BNE	RCI1	NO

E59B A5 2F	CHECK FOR EOL ON TEXT RECORDS		
E59D C9 9B	LDA	CIOCHR	GET BYTE
E59F 00 06	CMF	#EOL	IS IT AN EOL?
E5A1 20 63 E6	BNE	RCI1	NO
E5A4 4C C3 E5	JSR	DECBFL	YES, DECREMENT BUFFER LENGTH
	JMP	RCI4	END TRANSFER

E5A7 20 63 E6	CHECK BUFFER FULL		
E5AA 00 DB	RCI1 JSR	DECBFL	DECREMENT BUFFER LENGTH
	BNE	RCI3	CONTINUE IF NON ZERO

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

```

; BUFFER FULL. RECORD NOT ENDED
; DISCARD BYTES UNTIL END OF RECORD
E5AC A5 22      RC12  LDA  ICCOMZ  GET COMMAND BYTE
E5AE 29 02      AND   #2          IS IT GET CHARACTER?
E5B0 D0 11      BNE   RC14        YES, END TRANSFER

```

```

; LOOP TO WAIT FOR EOL
E5B2 20 89 E6   RC16  JSR   QDHAND  GET BYTE FROM HANDLER
E5B5 85 2F      BTI   CIOCHR      SAVE CHARACTER
E5B7 30 0A      BMI   RC14        GO IF ERROR

```

```

; TEXT RECORD. WAIT FOR EOL
E5B9 A5 2F      LDA   CIOCHR      GET QOT BYTE
E5BB C9 9B      CMP   #EOL        IS IT EOL?
E5BD D0 F3      BNE   RC16        NO, CONTINUE

```

```

; END OF RECORD. BUFFER FULL -- SEND TRUNCATED RECORD MESSAGE
E5BF A9 89      RC111 LDA  #TRNRCD ERROR CODE
E5C1 85 23      BTI   ICSTAZ      STORE IN IOCB

```

```

; TRANSFER DONE
E5C3 20 77 E6   RC14  JSR   BUDBFL  SET FINAL BUFFER LENGTH
E5C6 4C 1D E6   JRP   CIRTN2      RETURN

```

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

```

WRITE -- DO PUT COMMANDS
E5C9 A5 22      CIOWRIT LDA      ICCOMZ      GET COMMAND BYTE
E5CB 25 2A      AND      ICAX1Z      IS THIS WRITE LEGAL?
E5CD D0 05      BNE      WCI1A      YES

```

```

ILLEGAL WRITE -- DEVICE OPENED FOR READ ONLY
E5CF A0 B7      LDY      WRDONLY      ERROR CODE
E5D1 4C 1B E6   WCI1B JMP      CIRTN1      RETURN

```

```

COMPUTE AND CHECK ENTRY POINT
E5D4 20 5D E6   WCI1A JBR      COMENT      COMPUTE HANDLER ENTRY POINT
E5D7 B0 F8      BCB      WCI1B      GO IF ERROR IN COMPUTE

```

```

PUT RECORD OR CHARACTERS
E5D9 A5 2B      LDA      ICBLLZ
E5DB 05 29      ORA      ICBLLZ+1      IS BUFFER LENGTH ZERO?
E5DD D0 06      BNE      WCI13      NO
E5DF A5 2F      LDA      CIOCHR      GET CHARACTER
E5E1 E6 2B      INC      ICBLLZ      GET BUFFER LENGTH-1
E5E3 D0 06      BNE      WCI14      THEN JUST TRANSFER ONE BYTE

```

```

LOOP TO TRANSFER BYTES FROM BUFFER TO HANDLER
E5E5 A0 00      WCI13 LDY      #0
E5E7 B1 24      LDA      (ICBALZ),Y      GET BYTE FROM BUFFER
E5E9 B5 2F      STA      CIOCHR      SAVE
E5EB 20 B9 E6   WCI14 JBR      GOHAND      DO PUT BYTE
E5EE 30 25      BMI      WCI15      END IF ERROR
E5F0 20 70 E6   JBR      INCBFP      INCREMENT BUFFER POINTER

```

```

CHECK FOR TEXT RECORD
E5F3 A5 22      LDA      ICCOMZ      GET COMMAND BYTE
E5F5 29 02      AND      #2      IS IT PUT RECORD?
E5F7 D0 0C      BNE      WCI11      NO

```

```

TEXT RECORD -- CHECK FOR EOL TRANSFER
E5F9 A5 2F      LDA      CIOCHR      GET LAST CHARACTER
E5FB C9 9B      CMP      #EOL      IS IT AN EOL?
E5FD D0 06      BNE      WCI11      NO
E5FF 20 63 E6   JBR      DECBFL      DECREMENT BUFFER LENGTH
E602 4C 15 E6   JMP      WCI15      END TRANSFER

```

```

CHECK FOR BUFFER EMPTY
E603 20 63 E6   WCI11 JBR      DECBFL      DECREMENT BUFFER LENGTH
E60B D0 D8      BNE      WCI13      CONTINUE IF NON ZERO

```

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

1 BUFFER EMPTY, RECORD NOT FILLED

1 CHECK TYPE OF TRANSFER

GET COMMAND CODE

IS IT PUT CHARACTER?

E60A A5 22

WC12

LDA

ICCOMZ

E60C 29 02

AND

#2

YES, END TRANSFER

E60E D0 05

BNE

WC15

1 PUT RECORD (TEXT), BUFFER EMPTY, SEND EOL

E610 A9 98

LDA

WEOL

GO TO HANDLER

E612 20 89 E6

JBR

GDHAND

1 END PUT TRANSFER

SET ACTUAL PUT BUFFER LENGTH
RETURN

E615 20 77 E6

WC15

JBR

SUBBFL

E618 4C 1D E6

JMP

CIRTN2

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

; CIO RETURNS			
; RETURNS WITH Y=STATUS			
E61B 84 23	CIRTN1	STY	ICSTAZ
			SAVE STATUS
; RETURNS WITH STATUS STORED IN ICSTAZ			
; MOVE IOCB IN ZERO PAGE BACK TO USER AREA			
E61D A4 2E	CIRTN2	LDY	ICIDNO
			GET IOCB INDEX
E61F B9 44 03		LDA	ICBAL,Y
E622 85 24		STA	ICBAL,Z
			RESTORE USER BUFFER POINTER
E624 B9 45 03		LDA	ICBAH,Y
E627 B5 25		STA	ICBAH,Z
E629 A2 00		LDX	#0
			LOOP COUNT AND INDEX
E62B B5 20	CIRT3	LDA	IOCBAS,X
			ZERO PAGE
E62D 99 40 03		STA	IOCB,Y
			TO USER AREA
E630 EB		INX	
E631 CB		INY	
E632 E0 0C		CPX	#12
			12 BYTES
E634 90 F9		BCC	CIRT3
; RESTORE A,X, & Y			
E636 A5 2F		LDA	CIOCHR
			GET LAST CHARACTER
E638 A6 2E		LDX	ICIDNO
			IOCB INDEX
E63A A4 23		LDY	ICSTAZ
			GET STATUS AND SET FLAGS
E63C 60		RTB	RETURN TO USER

CIO SUBROUTINES

COMENT -- CHECK AND COMPUTE HANDLER ENTRY POINT

E63D A4 20	COMENT	LDY	ICHDZ	GET HANDLER INDEX
E63F C0 22		CPY	#MAXDEV+1	IS IT A LEGAL INDEX?
E641 90 04		BCC	COM1	YES

ILLEGAL HANDLER INDEX MEANS DEVICE NOT OPEN FOR OPERATION

E643 A0 B5		LDY	#NOTOPN	ERROR CODE
E645 B0 1B		BCB	COM2	RETURN

USE HANDLER ADDRESS TABLE AND COMMAND TABLE TO GET VECTOR

E647 B9 1B 03	COM1	LDA	HATABB+1,Y	GET LOW BYTE OF ADDRESS
E64A B5 2C		STA	ICSPRZ	AND SAVE IN POINTER
E64C B9 1C 03		LDA	HATABB+2,Y	GET HI BYTE OF ADDRESS
E64F B5 2D		STA	ICSPRZ+1	
E651 A4 17		LDY	ICCOMT	GET COMMAND CODE
E653 B9 C6 E6		LDA	COMTAB-3,Y	GET COMMAND OFFSET
E656 A8		TAY		
E657 B1 2C		LDA	(ICSPRZ),Y	GET LOW BYTE OF VECTOR FROM
E659 AA		TAX		HANDLER ITSELF AND SAVE
E65A C8		INY		
E65B B1 2C		LDA	(ICSPRZ),Y	GET HI BYTE OF VECTOR
E65D B5 2D		STA	ICSPRZ+1	
E65F B6 2C		BTX	ICSPRZ	SET LO BYTE
E661 1B		CLC		SHOW NO ERROR
E662 60	COM2	RTS		

DECBFL -- DECREMENT BUFFER LENGTH DOUBLE BYTE

Z FLAG = 0 ON RETURN IF LENGTH = 0 AFTER DECREMENT

E663 C6 28	DECBFL	DEC	ICBL LZ	DECREMENT LOW BYTE
E665 A5 2B		LDA	ICBL LZ	CHECK IT
E667 C9 FF		CMF	#FF	DID IT GO BELOW?
E669 D0 02		BNE	DECBF1	NO
E66B C6 29		DEC	ICBL LZ+1	DECREMENT HI BYTE
E66D 05 29	DECBF1	ORA	ICBL LZ+1	SET Z IF BOTH ARE ZERO
E66F 60		RTB		

INCBFP -- INCREMENT WORKING BUFFER POINTER

E670 E6 24	INCBFP	INC	ICBALZ	BUMP LOW BYTE
E672 D0 02		BNE	INCBF1	GO IF NOT ZERO
E674 E6 25		INC	ICBALZ+1	ELSE, BUMP HI BYTE
E676 60	INCBF1	RTB		

BUBBFL -- SET BUFFER LENGTH = BUFFER LENGTH - WORKING BYTE COUNT

E677 A6 2E	BUBBFL	LDX	ICIDND	GET IDCB INDEX
E679 3B		SEC		
E67A BD 4B 03		LDA	ICBL LZ	GET LOW BYTE OF INITIAL LENGTH
E67D E5 2B		SBC	ICBL LZ	SUBTRACT FINAL LOW BYTE
E67F B5 2B		STA	ICBL LZ	AND SAVE BACK
E681 BD 49 03		LDA	ICBLH,X	GET HI BYTE
E684 E5 29		SBC	ICBL LZ+1	
E686 B5 29		STA	ICBLH Z	
E688 60		RTB		

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

GOHAND -- GO INDIRECT TO A DEVICE HANDLER

Y= STATUS ON RETURN, N FLAG=1 IF ERROR ON RETURN

E689 A0 92	GOHAND LDY	MFNCNDT	PREPARE NO FUNCTION STATUS FOR HANDLER RTB
E68B 20 93 E6	JBR	CIJUMP	USE THE INDIRECT JUMP
E68E B4 23	BTY	ICBTAT	SAVE STATUS
E690 C0 00	CPY	#0	AND SET N FLAG
E692 60	RTB		

INDIRECT JUMP TO HANDLER BY PAUL'S METHOD

E693 AA	CIJUMP TAX		SAVE A
E694 A5 2D	LDA	ICBPRZ+1	GET JUMP ADDRESS HI BYTE
E696 4B	PHA		PUT ON STACK
E697 A5 2C	LDA	ICBPRZ	GET JUMP ADDRESS LO BYTE
E699 4B	PHA		PUT ON STACK
E69A BA	TXA		RESTORE A
E69B A6 2E	LDX	ICIDNQ	GET IOCB INDEX
E69D 60	RTB		GO TO HANDLER INDIRECTLY

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

DEVBRG -- DEVICE SEARCH. FIND DEVICE IN HANDLER ADDRESS TABLE

LOOP TO FIND DEVICE

E69E A0 00	DEVBRG	LDY	#0	
E6A0 B1 24		LDA	(ICBALZ),Y	GET DEVICE NAME FROM UBER
E6A2 F0 0C		BEQ	CIERR2	
E6A4 A0 21		LDY	#MAXDEV	INITIAL COMPARE INDEX
E6A6 D9 1A 03	DEV81	CMF	HATAB5,Y	IS THIS THE DEVICE?
E6A9 F0 0A		BEQ	DEV82	YES
E6AB 88		DEY		
E6AC 88		BEY		ELSE, POINT TO NEXT DEVICE NAME
E6AD 88		DEY		
E6AE 10 F6		BPL	DEV81	CONTINUE FOR ALL DEVICES

NO DEVICE FOUND. DECLARE NON-EXISTENT DEVICE ERROR

E6B0 A0 82	CIERR2	LDY	#NONDEV	ERROR CODE
E6B2 38		SEC		SHOW ERROR
E6B3 B0 13		BCS	DEV84	AND RETURN

FOUND DEVICE. SET ICHID,ICDND, AND INIT DEVICE

E6B5 98	DEV82	TYA		
E6B6 B5 20		BTA	ICHIDZ	SAVE HANDLER INDEX
E6B8 38		SEC		
E6B9 A0 01		LDY	#1	
E6BB B1 24		LDA	(ICBALZ),Y	GET DEVICE NUMBER (DRIVE NUMBER)
E6BD E9 30		BBC	#ABCZER	SUBTRACT ABCZY ZERO
E6BF C9 0A		CMF	#9A	IS NUMBER IN RANGE?
E6C1 90 02		BCC	DEV83	YES
E6C3 A9 01		LDA	#1	NO, DEFAULT TO ONE
E6C5 B5 21	DEV83	BTA	ICDNDZ	SAVE DEVICE NUMBER
E6C7 18		CLC		SHOW NO ERROR

RETURN

E6CB 60	DEV84	RTB		
---------	-------	-----	--	--

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

CIO ROM TABLES

COMMAND TABLE

MAPS EACH COMMAND TO OFFSET FOR APPROPRIATE VECTOR IN HANDLER

COMTAB . BYTE 0, 4, 4, 4, 4, 6, 6, 6, 2, 8, 10

E6C9 00 04 04

E6CC 04 04 06

E6CF 05 05 05

E6D2 02 08 0A

022F

LENGTH=*-CIOINT

E6D5

CRNTPC =*

*=*14

0014 00

CIOBPR . BYTE INTORG-CRNTPC CIOI 18 TOO LONG

END

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

SYMBOL TABLE

ADDCOR	030E	ADREBB	0064	AFP	D800	ALLPOT	D208
ANTIC	D400	APPEND	D001	APPMH1	000E	ASCZER	0030
ATACHR	02FB	ATAN	BE43	ATTRACT	004D	AUDC1	D201
AUDC2	D203	AUDC3	D205	AUDC4	D207	AUDCTL	D208
AUDF1	D200	AUDF2	D202	AUDF3	D204	AUDF4	D206
BADIQC	0086	BADMOD	0091	BFENH1	0035	BFENLO	0034
BITMSK	006E	BLIM	028A	BLKBDV	E471	BOOT?	0009
BOOTAD	0242	BOTSCR	02BF	BPTR	003D	BRKABT	0080
BRKKEY	0011	BUFADR	0015	BUFCNT	006B	BUFRFL	0038
BUFRH1	0033	BUFRLO	0032	BUFSTR	006C	CABBUF	03FD
CASERV	E440	CABFLO	030F	CASINI	0002	CASORG	EF41
CASSBT	004B	CASSET	0043	CAUX1	023C	CAUX2	023D
CBAUDH	02EF	CBAUDL	02EE	CCOMND	023B	CDEVIC	023A
CDTMA1	0226	CDTMA2	0228	CDTMF3	022A	CDTMF4	022C
CDTMF3	022E	CDTMV1	0218	CDTMV2	021A	CDTMV3	021C
CDTMV4	021E	CDTMV5	0220	CH	02FC	CH1	02F2
CHACT	02F3	CHACTL	D401	CHAR	02FA	CHBAS	02F4
CHBASE	D409	CHKERR	008F	CHKBNT	003B	CHKSUM	0031
CHRROR	E000	CICLO2	E53F	CICLOB	E533	CIERR1	E4D1
CIERR2	E680	CIERR3	E50F	CIERR4	E511	CIJUMP	E493
CIO	E4C4	CIOCHR	002F	CIOI1	E4AB	CIOINT	E4A6
CIOINV	E46E	CIOORQ	E4A6	CIOPEN	E509	CIOBPR	0014
CIOV	E456	CIREAD	E549	CIRT3	E62B	CIRTN1	E418
CIRTN2	E61D	CIST1	E559	CISTSP	E54E	CIWRIT	E5C9
CIX	00F2	CKEY	004A	CLOBE	000C	COLAC	0072
COLBK	D01A	COLCRB	0055	COLDST	0244	COLDSV	E477
COLINC	007A	COLON	003A	COLOR0	02C4	COLOR1	02C5
COLOR2	02C6	COLOR3	02C7	COLOR4	02C8	COLPFO	D016
COLPF1	D017	COLPF2	D018	COLPF3	D019	COLPH0	D012
COLPM1	D013	COLPM2	D014	COLPM3	D015	COLRSH	004F
COM1	E647	COM2	E662	COMENT	E63D	COMTAB	E6C9
CONBOL	D01F	COS	D073	COUNTR	007E	CR	0098
CRETRY	0036	CRITIC	0042	CRNTPC	E6D5	CRBINH	02F0
CRSROR	008D	CSOPIV	E47D	CBTAT	0288	CTIA	D000
DAUX1	030A	DAUX2	030B	DBSECT	0241	DBUFH1	0305
DBUFLO	0304	DBYTH1	0309	DBYTLO	0308	DCB	0300
DCOMND	0302	DDEVIC	0300	DECBF1	E66D	DECBFL	E663
DEGFLQ	00FB	DEGON	0006	DELETE	0021	DELTAC	0077
DELTAR	0076	DERROR	0090	DEV81	E6A6	DEV82	E6B5
DEV83	E6C5	DEV84	E6C8	DEV8RC	E69E	DFLA08	0240
DIGRT	00F1	DINDEX	0057	DIRECT	0002	DIBK	0044
DISKIV	E450	DISPLY	0053	DLIBTH	D403	DLIBTL	D402
DMACTL	D400	DMASK	02A0	DNACK	008B	DOSINI	000C
DOSVEC	000A	DRAWLN	0011	DRETRY	0037	DRKHSK	004E
DSKFMB	0018	DSKINV	E453	DSKORQ	EDEA	DSKTIM	0246
DSKUTL	001A	DSPFLO	02FE	DSTAT	004C	DSTAT8	0303
DTIMLO	0306	DUNIT	0301	DUNUBE	0307	DVSTAT	02EA
EDITRV	E400	EEXP	00ED	ENDPT	0074	ENTVEC	002C
EDFERR	0088	EOL	009B	ERRFLQ	023F	ERRTN	E4C0
ERRTNH	00E4	ERRTNL	00C0	ESCFLO	02A2	ESIGN	00EF
EXP	DDC0	EXP10	DDCC	FADD	DA66	FASC	D8E6
FCHREF	00F0	FDIV	DB2B	FEQF	003F	FILDAT	02FD
FILFLO	02B7	FILLIN	0012	FLDOP	DD8D	FLDOR	DD89
FLDIP	DD9C	FLD1R	DD98	FLPTR	00FC	FMOVE	DD86
FMSIPQ	0043	FMUL	DAD8	FNCNOT	0092	FORMAT	0022
FPI	D9D2	FPREC	0006	FPSCR	05E6	FPBCR1	05EC
FPTR2	00FE	FRO	00D4	FR1	00E0	FR2	00E6
FRE	00DA	FREQ	0040	FRMERR	00BC	FRX	00EC
FSCR	05E6	FSCR1	05EC	FSTOP	DDAB	FSTOR	DDA7

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

FSUB	DA60	FTYPE	003E	GETCHR	0007	GETREC	0005
GLDAB8	02E0	GOHAND	E689	GPRIOR	026F	GRACITL	D01D
GRAFM	D011	GRAFP0	D00D	GRAFP1	D00E	GRAFP2	D00F
GRAFP3	D010	HATAB8	031A	HITCLR	D01E	HOLD1	0051
HOLD2	029F	HOLD3	029D	HOLD4	028C	HOLD5	028D
HOLDCH	007C	HPOSM0	D004	HPOSM1	D005	HPOSM2	D006
HPOSM3	D007	HPOSP0	D000	HPOSP1	D001	HPOSP2	D002
HPOSP3	D003	HSCROL	D404	ICAX1	034A	ICAX1Z	002A
ICAX2	034B	ICAX2Z	002B	ICBAH	0345	ICBAHZ	0025
ICBAL	0344	ICBALZ	0024	ICBLH	0349	ICBLHZ	0029
ICBL1	0348	ICBL1Z	0028	ICCOM	0342	ICCOMT	0017
ICCOMZ	0022	ICDND	0341	ICDNDZ	0021	ICID1	0340
ICIDZ	0020	ICIDND	002E	ICPTH	0347	ICPTHZ	0027
ICPTL	0346	ICPTLZ	0026	ICSPR	034C	ICSPRZ	002C
ICSTA	0343	ICSTAZ	0023	IFP	D9AA	INBUFF	00F3
INCBF1	E676	INCBFP	E670	INSCLR	0020	INSDAT	007D
INTAB8	0200	INTEMP	022D	INTINV	E468	INTORG	E6D5
INTZB8	0010	INVFL0	0286	IOC1	E4D6	IOC1A	E4D8
IOC2	E4F3	IOC6	E514	IOC7	E519	IOCB	0340
IOCBAS	0020	IOCB8Z	0010	IOCFRE	00FF	IRGEN	D20E
IRGST	D20E	RBCODE	D209	RBD	004B	RBDORG	F3E4
KEYBDV	E420	KEYDEL	02F1	LBFEND	05FF	LBPR1	057E
LBPR2	057F	LBUFF	0580	LEDGE	0002	LENGTH	022F
LINBUF	0247	LINZB8	0000	LMARON	0052	LOCKFL	0023
LOG	DECD	LOG10	DED1	LOGCOL	0063	LOGMAP	0282
LPENH	0234	LPENV	0235	MOPF	D000	MOPL	D008
MIPF	D001	M1PL	D009	M2PF	D002	M2PL	D00A
M3PF	D003	M3PL	D008	MAXDEV	0021	MAXIOC	0080
MEMLO	02E7	MENTOP	02E5	MLTTP	0066	MODEN	004D
MONDR0	F0E3	MXDMOD	0010	NEWCOL	0051	NEWROW	0060
NMIEN	D40E	NMIREB	D40F	NMIST	D40F	NOCK8H	003C
NONDEV	0082	NOTE	0026	NOTOPN	0085	NSION	00EE
NUALID	0084	OLDADR	005E	OLDCHR	005D	OLDCOL	005B
OLDROW	005A	OPEN	0003	OPNIN	0004	OPNINO	000C
OPNOT	000B	OPNTMP	0066	QVRRUN	008E	POPF	D004
POPL	D00C	PIPF	D005	PIPL	D00D	P2PF	D006
P2PL	D00E	P3PF	D007	P3PL	D00F	PACTL	D302
PADDLO	0270	PADDL1	0271	PADDL2	0272	PADDL3	0273
PADDL4	0274	PADDL5	0275	PADDL6	0276	PADDL7	0277
PBCTL	D303	PBPNT	001D	PBUFBZ	001E	PCOLR0	02C0
PCOLR1	02C1	PCOLR2	02C2	PCOLR3	02C3	PENH	D40C
PENV	D40D	PIA	D300	PLYARG	05E0	PLYEVL	D040
PMBASE	D407	POINT	0025	POKEY	D200	POKMBK	0010
PORTA	D300	PORTB	D301	POTO	D200	POT1	D201
POT2	D202	POT3	D203	POT4	D204	POT5	D205
POT6	D206	POT7	D207	POT80	D208	PRINTR	0050
PRINTV	E430	PRIOR	D018	PRNBUF	03C0	PRNORG	EE78
PRODPR	0081	PTERP	001F	PTINDT	001C	PTRIO0	027C
PTRIO1	027D	PTRIO2	027E	PTRIO3	027F	PTRIO4	0280
PTRIO5	0281	PTRIO6	0282	PTRIO7	0283	PUTCHR	000B
PUTREC	0009	RADFL0	00FB	RADON	0000	RAFLD	0004
RAMSI2	02E4	RAMTOP	006A	RANDOM	D20A	RBL0KV	E47A
RC11	E5A7	RC111	E58F	RC11A	E574	RC11B	E571
RC12	E5AC	RC13	E587	RC14	E5C3	RC15	E582
RONLY	0087	RECVDN	0039	REDGE	0027	RENAME	0020
RMARON	0053	ROWAC	0070	ROWCR8	0054	ROWINC	0079
RTCLOK	0012	SAVADR	006B	SAVIO	0316	SAVMBK	005B
SCREDT	0045	SCRENV	E410	SCRFL0	028B	SCRNEM	0093
SDL8TH	0231	SDL8TL	0230	SDMCTL	022F	SENDEV	E468
BERIN	D20D	BEROUT	D20D	SETV8V	E45C	SHFAHT	006F

CENTRAL INPUT/OUTPUT (C10) 2-7-79

SHFLOK	02BE	SIN	BDB1	SIOINV	E465	SIOORQ	E944
BIOV	E459	SIZEM	D00C	SIZEP0	D008	SIZEP1	D009
SIZEP2	D00A	SIZEP3	D00B	SKCTL	D20F	SKRES	D20A
SKSTAT	D20F	SOUNDR	0041	SPECIL	000E	SQR	8EB1
SRTIMR	022B	SSFLAQ	02FF	SSKCTL	0232	STACKP	0318
STATIS	000D	STATUS	0030	STICK0	027B	STICK1	0279
STICK2	027A	STICK3	027B	STIMER	D209	STRIO0	0284
STRIO1	0285	STRIO2	0286	STRIO3	0287	SUBBFL	E677
SUBTMP	029E	SUCCESS	0001	SWPFL0	0078	SVBVBV	E45F
TABMAP	02A3	TEMP	023E	TEMP1	0312	TEMP2	0314
TEMP3	0315	TIMER1	030C	TIMER2	0310	TIMFLO	0317
TIMOUT	008A	TINDEX	0293	TMPCHR	0056	TMPCOL	0289
TMPLBT	02A1	TMPROW	028B	TMPX1	029C	TRAMBZ	0006
TRIO0	D010	TRIO1	D011	TRIO2	D012	TRIO3	D013
TRNRCD	00B9	TSTAT	0319	TSTDAT	0007	TXTCOL	0291
TXTMSC	0294	TXTOLD	0296	TXTRDW	0290	UNLOCK	0024
USAREA	0480	VBREAK	0206	VCOUNT	D40B	VCTABL	E480
VDELAY	D01C	VDSLST	0200	VECTBL	E400	VIMIR0	0216
VINTER	0204	VKEYBD	0208	VPRCED	0202	VSCROL	D405
VSERIN	020A	VBEROC	020E	VBEROR	020C	VTIMR1	0210
VTIMR2	0212	VTIMR4	0214	VVBLKD	0224	VVBLK1	0222
WARMST	0008	WARMSV	E474	WC11	E605	WC11A	E5D4
WC11B	E5D1	WC12	E60A	WC13	E5E5	WC14	E5EB
WC15	E615	WMODE	0289	WRONLY	0083	WSYNC	D40A
XITVBV	E462	XMTDDN	003A	ZIOCB	0020	ZTEMP1	00F5
ZTEMP3	00F9	ZTEMP4	00F7				

INTERRUPT HANDLER

.TITLE 'INTERRUPT HANDLER'

.LIVES ON DK1: INTHV.SRC

0005 BRTIM2 = 6 SECOND REPEAT INTERVAL

==SETVBV

E45C 4C 12 E9 JMP SETVBL

E45F 4C D1 E7 JMP SYBVBL

E462 4C 3E E9 JMP XITVBL

==INTINV

E46D 4C D5 E6 JMP IHINIT

* = VCTABL+INTABS-VDSLST

E480 B3 E7 .WORD SYRTI VDSLST

E482 B2 E7 .WORD SYIRGB VPRCED

E484 B2 E7 .WORD SYIRGB VINTER

E486 B2 E7 .WORD SYIRGB VBREAK

* = #18

E490 B2 E7 .WORD SYIRGB VTIMR1

E492 B2 E7 .WORD SYIRGB VTIMR2

E494 B2 E7 .WORD SYIRGB VTIMR4

E496 F6 E6 .WORD SYIRG VIMIRG

E498 00 00 00 .WORD 0,0,0,0,0 CDTMV1-4

E49A 00 00 00

E49E 00 00 00

E4A1 00

E4A2 D1 E7 .WORD SYBVBL VBLR1

E4A4 3E E9 .WORD XITVBL VBLKD

*=\$900C

900C A9 E6 LDA #PIRGH SET UP RAM VECTORS FOR LINUX VERSION

900E BD F9 FF STA \$FFF9

9011 A9 F3 LDA #PIRGL

9013 BD FB FF STA \$FFF8

9016 A9 E7 LDA #PNMIH

9018 BD FB FF STA \$FFFB

901B A9 B4 LDA #PNMIL

901D BD FA FF STA \$FFFA

9020 60 RTS

INTERRUPT HANDLER

IRQ HANDLER

JUMP THRU IMMEDIATE IRQ VECTOR, WHICH ORDINARILY POINTS TO
SYSTEM IRQ: DETERMINE & CLEAR CAUSE, JUMP THRU SOFTWARE VECTOR.

==INTORG

E6D5 A9 40	IHINIT	LDA	##40	VBL ON BUF DLIST OFF***FOR NOW***
E6D7 8D 0E D4		STA	NMIEN	ENABLE DISPLAY LIST, VERTICAL BLANK
E6DA A9 38		LDA	##38	LOOK AT DATA DIRECTION REGISTER8 IN PIA
E6DC 8D 02 D3		STA	PACTL	
E6DF 8D 03 D3		STA	PBCTL	
E6E2 A9 00		LDA	#0	MAKE ALL INPUTS
E6E4 8D 00 D3		STA	PORTA	
E6E7 8D 01 D3		STA	PORTB	
E6EA A9 3C		LDA	##3C	BACK TO PORTS
E6EC 8D 02 D3		STA	PACTL	
E6EF 8D 03 D3		STA	PBCTL	
E6F2 60		RTS		
E6F3 6C 16 02	PIRG	JMP	(VIMIRG)	
E6F6 48	SVIRG	PHA		SAVE A ON STACK
E6F7 A9 10		LDA	##10	SERIAL OUT READ BIT
E6F9 2C 0E D2		BIT	IRGST	SEE IF SER OUT RDY
E6FC D0 0D		BNE	SVIRG1	JUMP IF NOT
E6FE A9 EF		LDA	##EF	
E700 8D 0E D2		STA	IRGEN	
E703 A5 10		LDA	POKMSK	RESET ENABLE BYTE
E705 8D 0E D2		STA	IRGEN	
E708 6C 0C 02		JMP	(VSEOR)	JUMP THRU VECTOR
E70B A9 20	SVIRG1	LDA	##20	SERIAL IN READY BIT
E70D 2C 0E D2		BIT	IRGST	
E710 D0 0D		BNE	SVIRG2	
E712 A9 DF		LDA	##DF	
E714 8D 0E D2		STA	IRGEN	
E717 A5 10		LDA	POKMSK	
E719 8D 0E D2		STA	IRGEN	
E71C 6C 0A 02		JMP	(VBERIN)	
E71F A9 08	SVIRG2	LDA	##08	SERIAL OUT COMPLETE
E721 24 10		BIT	POKMSK	ONLY CHECK IF ENABLED
E723 F0 12		BEQ	SVIRG3	SINCE THIS IS A WIERD CASE IN THE
E725 2C 0E D2		BIT	IRGEN	HARDWARE
E728 D0 0D		BNE	SVIRG3	
E72A A9 F7		LDA	##F7	
E72C 8D 0E D2		STA	IRGEN	
E72F A5 10		LDA	POKMSK	
E731 8D 0E D2		STA	IRGEN	
E734 6C 0E 02		JMP	(VSEROC)	
E737 AD 0E D2	SVIRG3	LDA	IRGEN	TO CHECK TIMERS
E73A 6A		ROR	A	
E73B 80 0D		BCB	SVIRG4	TIMER1
E73D A9 FE		LDA	##FE	
E73F 8D 0E D2		STA	IRGEN	
E742 A5 10		LDA	POKMSK	
E744 8D 0E D2		STA	IRGEN	
E747 6C 10 02		JMP	(VTIMR1)	
E74A 6A	SVIRG4	ROR	A	
E74B 80 0D		BCB	SVIRG5	
E74D A9 FD		LDA	##FD	
E74F 8D 0E D2		STA	IRGEN	
E752 A5 10		LDA	POKMSK	

INTERRUPT HANDLER

E754 8D 0E D2	STA	IRGEN	
E757 6C 12 02	JMP	(VTIMR2)	
E75A 6A	BYIRG5	ROR	A
E75B 8D 0A	BCS	SYIRG6	
E75D A9 FB	LDA	##FB	
E75F 8D 0E D2	STA	IRGEN	
E762 A5 10	LDA	POKMBK	
E764 8D 0E D2	STA	IRGEN	
	JMP	(VTIMR4)	
E767 2C 0E D2	BYIRG6	BIT	IRGEN CHECK BREAK KEY, KEYBOARD INTERRUPTS
E76A 70 0D	BVS	SYIRG7	IF KEYBOARD
E76C A9 BF	LDA	##BF	
E76E 8D 0E D2	STA	IRGEN	
E771 A5 10	LDA	POKMBK	
E773 8D 0E D2	STA	IRGEN	
E776 6C 08 02	JMP	(VKEYBD)	
E779 30 18	BYIRG7	BMI	SYIRG8 BREAK KEY
E77B A9 7F	LDA	##7F	
E77D 8D 0E D2	STA	IRGEN	
E780 A5 10	LDA	POKMBK	
E782 8D 0E D2	STA	IRGEN	
E785 A9 00	LDA	#0	
E787 85 11	STA	BRKKEY	SET BREAK KEY FLAG
E789 8D FF 02	STA	SBFLAG	START/STOP FLAG
E78C 8D F0 02	STA	CRBINH	CURSOR INHIBIT
E78F 85 4D	STA	ATRACT	TURN OFF ATRACT MODE
E791 68	PLA		
E792 40	RTI		EXIT FROM INT
E793 2C 02 D3	BYIRG8	BIT	PACTL PROCEED ***I GUESS***
E796 10 06	BPL	BYIRG9	
E798 AD 00 D3	LDA	PORTA	CLEAR INT STATUS BIT
E79B 6C 02 02	JMP	(VPRCD)	
E79E 2C 03 D3	BYIRG9	BIT	PBCTL INTERRUPT ***I GUESS***
E7A1 10 06	BPL	BYIRGA	
E7A3 AD 01 D3	LDA	PORTB	CLEAR INT STATUS
E7A6 6C 04 02	JMP	(VINTER)	
E7A9 08	BYIRGA	PHP	
E7AA 68	PLA		SEE IF SOFTWARE BREAK
E7AB 29 10	AND	##10	B BIT OF P REGISTER
E7AD F0 03	BEG	BYIRGB	
E7AF 6C 06 02	JMP	(VBREAK)	
E7B2 68	BYIRGB	PLA	
E7B3 40	BYRTI	RTI	UNIDENTIFIED INTERRUPT, JUST RETURN.

INTERRUPT HANDLER

NMI HANDLER

DETERMINE CAUSE AND JUMP THRU VECTOR

E7D4 2C 0F D4	PNMI	BIT	NMIST	
E7B7 10 03		BPL	PNM11	SEE IF DISPLAY LIST
E7B9 6C 00 02		JMP	(VDSLBT)	
E7DC 4B	PNM11	PHA		
E7BD AD 0F D4		LDA	NMIST	
E7C0 29 20		AND	#20	SEE IF RESET
E7C2 F0 03		BEQ	#+5	
E7C4 4C 74 E4		JMP	WARMBV	GO THRU WARM START JUMP
E7C7 8A		TXA		SAVE REGISTERS
E7CB 4B		PHA		
E7C9 9B		TYA		
E7CA 4B		PHA		
E7CB 8D 0F D4		BTA	NMIREB	RESET INTERRUPT STATUS
E7CE 6C 22 02		JMP	(VVBLKI)	JUMP THRU VECTOR

INTERRUPT HANDLER

SYSTEM VBLANK ROUTINE

INC FRAME COUNTER. PROCESS COUNTDOWN TIMERS. EXIT IF I WAS SET. CLEAR I.
SET DL1STL, DL1STH, DMACTL FROM RAM CELLS. DO SOFTWARE REPEAT.

E7D1 E6 14	BYSVBL	INC	RTCLK+2	INC FRAME COUNTER
E7D3 D0 08		BNE	BYSVB1	
E7D5 E6 4D		INC	ATRACT	INCREMENT ATRACT (CAUSES ATRACT WHEN MINUS)
E7D7 E6 13		INC	RTCLK+1	
E7D9 D0 02		BNE	BYSVB1	
E7DB E6 12		INC	RTCLK	
E7DD A9 FE	BYSVB1	LDA	##FE	{ATRACT} SET DARK MASK TO NORMAL
E7DF A2 00		LDX	#0	SET COLRSH TO NORMAL
E7E1 A4 4D		LDY	ATRACT	TEST ATRACT FOR NEGATIVE
E7E3 10 06		BPL	VBATRA	WHILE POSITIVE, DONT GO INTO ATRACT
E7E5 B5 4D		STA	ATRACT	IN ATRACT, SO STAY BY STA #FE
E7E7 A6 13		LDX	RTCLK+1	COLOR SHIFT FOLLOWS RTCLK+1
E7E9 A9 F6		LDA	##F6	SET DARK MASK TO DARK
E7EB B5 4E	VBATRA	STA	DRKMSK	
E7ED B6 4F		BTX	COLRSH	
E7EF A2 00		LDX	#0	POINT TO TIMER1
E7F1 20 F5 E8		JSR	DCTMR	GO DECREMENT TIMER1
E7F4 D0 03		BNE	BYSVB2	BRANCH IF STILL COUNTING
E7F6 20 EF E8		JSR	JTMR1	GO JUMP TO ROUTINE
E7F9 A5 42	BYSVB2	LDA	CRITIC	
E7FB D0 08		BNE	XXIT	GO IF CRITICAL SET
E7FD BA		TBX		SEE IF I WAS SET
E7FE BD 04 01		LDA	#104, X	GET STACKED P
E801 29 04		AND	#04	1 BIT
E803 F0 03		BEQ	BYSVB3	BRANCH IF OK
E805 4C 3E E9	XXIT	JMP	XITVBL	I WAS SET, EXIT
E808 58	BYSVB3	CLI		
E809 AD 0D D4		LDA	PENV	
E80C BD 35 02		STA	LPENV	
E80F AD 0C D4		LDA	PENV	
E812 BD 34 02		STA	LPENV	
E815 AD 31 02		LDA	BDL8TH	
E818 BD 03 D4		STA	DL1STH	
E81B AD 30 02		LDA	BDL8TL	
E81E BD 02 D4		STA	DL1STL	
E821 AD 2F 02		LDA	BDHCTL	
E824 BD 00 D4		STA	DMACTL	
E827 AD 6F 02		LDA	OPROR	GLOBAL PRIOR
E82A BD 1B D0		STA	PRIOR	
E82D A9 08		LDA	#08	TURN OFF KEYBOARD SPEAKER
E82F BD 1F D0		STA	CONSOL	
E832 A2 08		LDX	#8	
E834 BD C0 02	SCOLLF	LDA	PCOLOR, X	LOAD COLOR REGISTERS FROM RAM
E837 45 4F		EOR	COLRSH	DO COLOR SHIFT
E839 25 4E		AND	DRKMSK	AND DARK ATRACT
E83B 9D 12 D0		STA	COLPMO, X	
E83E CA		DEX		
E83F 10 F3		BPL	SCOLLF	
E841 AD F4 02		LDA	CHBAS	
E844 BD 09 D4		STA	CHBASE	
E847 AD F3 02		LDA	CHACT	
E84A BD 01 D4		STA	CHACTL	
E84D A2 02		LDX	#2	POINT TO TIMER 2
E84F 20 F5 E8		JSR	DCTMR	

INTERRUPT HANDLER

```

EB52 D0 03      BNE    SYSVB4  IF DIDNT GO ZERO
EB54 20 F2 EB    JSR    JTIMR2  GO JUMP TO TIMER2 ROUTINE
EB57 A2 02      LDX    #2      RESTORE X
EB59 EB        SYVB84  LDX
EB5A EB        SYVB8B  INX
EB5B BD 18 02    LDA    CDTMV1,X
EB5E 1D 19 02    ORA    CDTMV1+1,X
EB61 F0 06      BEQ    SYSVB8A
EB63 20 F5 EB    JSR    DCTIMR  DECREMENT AND SET FLAG IF NONZERO
EB66 9D 26 02    STA    CDTMF3-4,X
EB69 E0 08      CPX    #8      SEE IF DONE ALL 3
EB6B D0 EC      BNE    SYVB8B  LOOP

```

```

; CHECK DEBOUNCE COUNTER
EB6D AD 0F D2    LDA    BKSTAT
EB70 29 04      AND    #04     KEY DOWN BIT
EB72 F0 0B      BEQ    SYVB6A  IF KEY DOWN

```

```

; KEY UP GO COUNT IT
EB74 AD F1 02    LDA    KEYDEL  KEY DELAY COUNTER
EB77 F0 03      BEQ    SYVB6A  IF COUNTED DOWN ALREADY
EB79 CE F1 02    DEC    KEYDEL  COUNT IT

```

```

; CHECK SOFTWARE REPEAT TIMER
EB7C AD 2B 02    SYVB6A  LDA    SRTIMR
EB7F F0 17      BEQ    SYVB87  DOESNT COUNT
EB81 AD 0F D2    LDA    BKSTAT
EB84 29 04      AND    #04     CHECK KEY DOWN BIT
EB86 D0 60      BNE    SYVB86  BRANCH IF NO LONGER DOWN
EB88 CE 2B 02    DEC    SRTIMR  COUNT FRAME OF KEY DOWN
EB8B D0 0B      BNE    SYVB87  BRANCH IF NOT RUN OUT

```

```

; TIMER RAN OUT - RESET AND SIMULATE KEYBOARD IRQ
EB8D A9 06      LDA    #SRTIMR  TIMER VALUE
EB8F BD 2B 02    STA    SRTIMR  SET TIMER
EB92 AD 09 D2    LDA    KBCODE  GET THE KEY
EB95 BD FC 02    STA    CH      PUT INTO CH

```

```

; READ GAME CONTROLLERS

```

```

EB9B A0 01      SYVB87  LDY    #1
EB9A A2 03      LDX    #3
EB9C B9 00 D3    BTLOOP  LDA    PORTA,Y
EB9F 4A          LSR    A
EBA0 4A          LSR    A
EBA1 4A          LSR    A
EBA2 4A          LSR    A

```

```

EBA3 9D 7B 02    STA    STICK0,X  STORE JOYSTICK
EBA6 CA          DEX
EBA7 B9 00 D3    LDA    PORTA,Y
EBA9 29 0F      AND    #0F
EBAE 9D 7B 02    STA    STICK0,X  STORE JOYSTICK
EBAF CA          DEX
EBB0 8B          DEY
EBB1 10 E9      BPL    BTLOOP

```

```

EBB3 A2 03      LDX    #3
EBB5 BD 10 D0    STRL   LDA    TRI00,X  MOVE JOYSTICK TRIGGERS
EBB8 9D B4 02    STA    STRI00,X
EBBB BD 00 D2    LDA    POT0,X  MOVE POT VALUES
EBBE 9D 70 02    STA    PADD0,X
EBC1 BD 04 D2    LDA    POT4,X
EBC4 9D 74 02    STA    PADD4,X
EBC7 CA          DEX
EBCB 10 EB      BPL    STRL
EBCA BD 0B D2    STA    POT0D  START POTS FOR NEXT TIME

```

INTERRUPT HANDLER

```

E8CD A2 06      LDX    #6
E8CF A0 03      LDY    #3
E8D1 B9 78 02   PTRLP  LDA    STICK0,Y      TRANSFER BITS FROM JOYSTICKS
E8D4 4A         LSR     A                  TO PADDLE TRIGGERS
E8D5 4A         LSR     A
E8D6 4A         LSR     A
E8D7 9D 7D 02   STA    PTR101,X
E8DA A9 00      LDA    #0
E8DC 2A         ROL     A
E8DD 9D 7C 02   STA    PTR100,X
E8E0 CA         DEX
E8E1 CA         DEX
E8E2 88         DEY
E8E3 10 EC      BPL     PTRLP

```

```

E8E5 6C 24 02   JMP     (VBLKD)          GO TO DEFERRED VBLANK ROUTINE
00EB           BV7H    =    BY8VB77256
009B           BV7L    =    -256*BV7H+BY8VB7
E8E8 A9 00      SY8VB6 LDA    #0
E8EA BD 28 02   STA    SRTIMR ZERO TIMER
E8ED F0 A9      BEQ     SY8VB7 UNCOND
E8EF 6C 26 02   JTIMR1 JMP     (CDTMA1)
E8F2 6C 28 02   JTIMR2 JMP     (CDTMA2)

```

; SUBROUTINE TO DECREMENT A COUNTDOWN TIMER
 ; ENTRY X=OFFSET FROM TIMER 1
 ; EXIT A.P=ZERO IF WENT ZERO, FF OTHERWISE

```

E8F5 BC 18 02   DCTIMR LDY    CDTMV1,X    LO BYTE
E8FB D0 08      BNE     DCTIM1 NONZERO, GO DEC IT
E8FA BC 19 02   LDY     CDTMV1+1,X        SEE IF BOTH ZERO
E8FD F0 10      BEQ     DCTXF YES, EXIT NONZERO
E8FF DE 19 02   DEC     CDTMV1+1,X        DEC HI BYTE
E902 DE 18 02   DCTIM1 DEC     CDTMV1,X    DEC LO BYTE
E905 D0 08      BNE     DCTXF
E907 BC 19 02   LDY     CDTMV1+1,X
E90A D0 03      BNE     DCTXF
E90C A9 00      LDA     #0                WENT ZERO, RETURN ZERO
E90E 60         RTS
E90F A9 FF      DCTXF  LDA     #FF        RETURN NONZERO
E911 60         RTS

```

INTERRUPT HANDLER

SUBROUTINE TO SET VERTICAL BLANK VECTORS AND TIMERS

ENTRY X=HI, Y=LO BYTE TO SET

A= 1-5 TIMERS 1-5

6 IMM VBLANK

7 DEF VBLANK

```

E912 0A      SETVBL ASL      A      MUL BY 2
E913 8D 2D 02 STA      INTEMP
E916 A9 00      LDA      #000      DLIST OFF BUT VBLANK OFF***FOR NOW***
E918 8D 0E D4      STA      NMHIEN
E919 8A      TXA
E91C AE 2D 02      LDX      INTEMP
E91F 9D 17 02      STA      CDTMV1-1, X
E922 98      TYA
E923 9D 16 02      STA      CDTMV1-2, X
E926 A9 40      LDA      #440      DLIST OFF***FOR NOW***, VBLANK ON
E928 8D 0E D4      STA      NMHIEN
E92B 2C 0F D4      BIT      NM1ST      SEE IF A VBLANK HAPPENED
E92E 50 0D      BVC      SETVB1      IF NOT, GO EXIT

```

SIMULATE VERTICAL BLANK NM1

```

E930 A9 E9      LDA      #BV1H
E932 4B      PHA      PUSH PC ON STACK AS IF INTERRUPT
E933 A9 3D      LDA      #BV1L
E935 4B      PHA
E936 0B      PHP      DUMMY P
E937 4B      PHA      DUMMY A
E938 4B      PHA      DUMMY X
E939 4B      PHA      DUMMY Y
E93A 6C 22 02      JMP      (VBLKT) GO THRU VBLANK VECTOR
E93D 60      SETVB1 RTB
00E9      BV1H =      SETVB1/256
003D      BV1L =      -256+BV1H+SETVB1

```

EXIT FROM VERTICAL BLANK

```

E93E 6B      XITVBL PLA      UNSTACK Y
E93F AB      TAY
E940 6B      PLA      UNSTACK X
E941 AA      TAX
E942 6B      PLA      UNSTACK A
E943 40      RTI      AND GO BACK FROM WHENCE.
00E6      PIRGH =      PIRG/256
00F3      PIRGL =      -256+PIRGH+PIRG
00E7      PNMIH =      PNMI/256
00B4      PNML =      -256+PNMIH+PNMI
SPARE BYTE OR MODULE TOO LONG FLAG
E944      CRNTPC =#
0014 00      INTSPR .BYTE #14      SIOORG-CRNTPC      INTNV IS TOO LONG
END

```

INTERRUPT HANDLER

SYMBOL TABLE							
ADDCOR	030E	ADRE88	0064	AFP	D800	ALLPOT	D208
ANTIC	D400	APPEND	0001	APPMHI	000E	ATACHR	02F8
ATAN	BE43	ATTRACT	004D	AUDC1	D201	AUDC2	D203
AUDC3	D205	AUDC4	D207	AUDCTL	D208	AUDF1	D200
AUDF2	D202	AUDF3	D204	AUDF4	D206	BADIOC	0086
BADMDO	0091	BFENHI	0035	BFENLO	0034	BITHSK	006E
BLIM	028A	BLKBDV	E471	BDOT7	0009	BOOTAD	0242
BOTSCR	028F	BPTR	003D	BRKABT	0080	BRKKEY	0011
BUFADR	0015	BUFCNT	006B	BUFRFL	0038	BUFRHI	0033
BUFRLO	0032	BUFBTR	006C	CASBUF	03FD	CABETV	E440
CASFL0	030F	CABINI	0002	CASOR0	EF41	CASSBT	004B
CASSET	0043	CAUX1	023C	CAUX2	023D	CBAUDH	02EF
CBAUDL	02EE	CCOMND	023B	CDEVIC	023A	CDTHA1	0226
CDTHA2	0228	CDTHF3	022A	CDTHF4	022C	CDTHF5	022E
CDTHV1	0218	CDTHV2	021A	CDTHV3	021C	CDTHV4	021E
CDTHV5	0220	CH	02FC	CH1	02F2	CHACT	02F3
CHACTL	D401	CHAR	02FA	CHBAS	02F4	CHBASE	D409
CHKERR	008F	CHKBNT	003B	CHKBUM	0031	CHRROR	E000
CIDCHR	002F	CIOINV	E46E	CIOOR0	E4A6	CIOV	E456
CIX	00F2	CKEY	004A	CLOBE	000C	COLAC	0072
COLBK	D01A	COLCR8	0055	COLD8T	0244	COLD8V	E477
COLINC	007A	COLOR0	02C4	COLOR1	02C5	COLOR2	02C6
COLOR3	02C7	COLOR4	02C8	COLPFO	D016	COLPF1	D017
COLPF2	D018	COLPF3	D019	COLPM0	D012	COLPM1	D013
COLPM2	D014	COLPM3	D015	COLRSH	004F	CONSOL	D01F
COS	BD73	COUNTR	007E	CR	009B	CRETRY	0036
CRITIC	0042	CRNTPC	E944	CRBINH	02F0	CRBROR	008D
CSOPIV	E47D	CSTAT	0288	CTIA	D000	DAUX1	030A
DAUX2	030B	DBSECT	0241	DBUFI	0305	DBUFLO	0304
DBYTHI	0309	DBYTLO	0308	DCB	0300	DCOMND	0302
DCTIMI	E902	DCTIMR	EBF5	DCTXF	E90F	DDEVIC	0300
DEGFL0	00FB	DEGON	0006	DELETE	0021	DELTAC	0077
DELTA	0076	DERROR	0090	DFLAG8	0240	DIORT	00F1
DINDEX	0057	DIRECT	0002	DISK	0044	DISKIV	E450
DISPLY	0053	DLTBTH	D403	DLIBTL	D402	DMACTL	D400
DMASK	02A0	DNACK	008B	DOSINI	000C	DOSVEC	000A
DRAWLN	0011	DRETRY	0037	DRKMSK	004E	DSKFMS	0018
DSKINV	E453	DSKOR0	EDEA	DSKTIM	0246	DSKUTL	001A
DSPFLO	02FE	DBTAT	004C	DBTAT8	0303	DTIMLO	0306
DUNIT	0301	DUNUSE	0307	DVSTAT	02EA	EDITRV	E400
EEXP	00ED	ENDPT	0074	EDFERR	008B	ERRFL0	023F
ESCFLO	02A2	EBION	00EF	EXP	DDC0	EXP10	DDCC
FADD	DA66	FABC	D8E6	FCHRFL	00F0	FDIV	D82B
FEOF	003F	FILDAT	02FD	FILFLO	02B7	FILLIN	0012
FLDOP	DD8D	FLDOR	DD89	FLD1P	DD9C	FLDIR	DD9B
FLPTR	00FC	FMOVE	DD86	FMSZP0	0043	FML	DADB
FNCNOT	0092	FORMAT	0022	FPI	D9D2	FPREC	0006
FPSCR	05E6	FPBCR1	05EC	FPTR2	00FE	FRO	00D4
FR1	00E0	FR2	00E6	FRE	00DA	FREG	0040
FRMERR	008C	FRX	00EC	FSCR	05E6	FBCR1	05EC
FSTOP	DDA8	FSTOR	DDA7	FSUB	DA60	FTYPE	003E
GETCHR	0007	GETREC	0005	GLBABB	02E0	GPRIOR	026F
GRACTL	D01D	GRAFH	D011	GRAFP0	D00D	GRAFP1	D00E
GRAFP2	D00F	GRAFP3	D010	HATABB	031A	HITCLR	D01E
HOLD1	0051	HOLD2	029F	HOLD3	029D	HOLD4	028C
HOLD3	028D	HOLDBH	007C	HPOBMO	D004	HPOBHI	D005
HPDSM2	D006	HPOBHI	D007	HPOBPO	D000	HPOBPI	D001
HPOBSP2	D002	HPOBSP3	D003	HBCROL	D404	ICAX1	034A
ICAX17	002A	ICAX2	034B	ICAX22	002B	ICBAH	0345

INTERRUPT HANDLER

ICBAHZ	0025	ICBAL	0344	ICBALZ	0024	ICBLH	0349
ICBLHZ	0029	ICBL	0348	ICBLZ	0028	ICCOM	0342
ICCOMT	0017	ICCOMZ	0022	ICDNO	0341	ICDNOZ	0021
ICHID	0340	ICHIDZ	0020	ICDNO	002E	ICPTH	0347
ICPTHZ	0027	ICPTL	0346	ICPTLZ	0026	ICBPR	034C
ICSPRZ	002C	ICBTA	0343	ICBTAZ	0023	IFF	D9AA
IHINIT	E6D5	INBUFF	00F3	INBCLR	0020	INSDAT	007D
INTAB9	0200	INTEMP	022D	INTINV	E46B	INTORG	E6D5
INTBPR	0014	INTZB8	0010	INVFL0	02B6	IOCB	0340
IOCBAS	0020	IOCB8Z	0010	IOCFRE	00FF	IRGEN	D20E
IRGST	D20E	JTIMR1	E8EF	JTIMR2	E8F2	KBCODE	D209
KBD	004B	KBDORG	F3E4	KEYBDV	E420	KEYDEL	02F1
LBFEND	05FF	LBPR1	057E	LBPR2	057F	LBUFF	0580
LEDGE	0002	LINBUF	0247	LINZB8	0000	LMARON	0052
LOCKFL	0023	LOO	DECD	LOQ10	DED1	LDGCOL	0063
LOOMAP	02B2	LPENH	0234	LPENV	0235	MOPF	D000
MOPL	D008	M1PF	D001	M1PL	D009	M2PF	D002
M2PL	D00A	M3PF	D003	M3PL	D00B	MAXDEV	0021
MAXIOC	0080	MEMLO	02E7	MEMTOP	02E5	MLTTMP	0066
MODEM	004D	MONORG	F0E3	MXDMOD	0010	NEWCOL	0061
NEWROW	0060	NMIEN	D40E	NMIREB	D40F	NH18T	D40F
NOCKBM	003C	NONDEV	0082	NOTE	0026	NOTOPN	0085
NSIGN	00EE	NVALID	0084	OLDADR	005E	OLDCHR	005D
OLDCOL	005B	OLDROW	005A	OPEN	0003	OPNIN	0004
OPNIND	000C	OPNOT	0008	OPNTMP	0066	OVRUN	008E
POPF	D004	POPL	D00C	P1PF	D005	P1PL	D00D
P2PF	D006	P2PL	D00E	P3PF	D007	P3PL	D00F
PACTL	D302	PADDL0	0270	PADDL1	0271	PADDL2	0272
PADDL3	0273	PADDL4	0274	PADDL5	0275	PADDL6	0276
PADDL7	0277	PBCTL	D303	PBPNT	001D	PBUF8Z	001E
PCOLRO	02C0	PCOLR1	02C1	PCOLR2	02C2	PCOLR3	02C3
PENH	D40C	PENV	D40D	PIA	D300	PIRQ	E6F3
PIRGH	00E6	PIRGL	00F3	PLYAR0	05E0	PLYEVL	D040
PMBASE	D407	PNMI	E7B4	PNMI1	E7BC	PNMIH	00E7
PNMIL	00B4	POINT	0025	POKEY	D200	POKMSK	0010
PORTA	D300	PORTB	D301	POT0	D200	POT1	D201
POT2	D202	POT3	D203	POT4	D204	POT5	D205
POT6	D206	POT7	D207	POTQ0	D208	PRINTR	0050
PRINTV	E430	PRIOR	D018	PRNBUF	03C0	PRNDRO	EE7B
PRVOPN	0081	PTMP	001F	PTIMOT	001C	PTRIQ0	027C
PTRIQ1	027D	PTRIQ2	027E	PTRIQ3	027F	PTRIQ4	0280
PTRIQ5	0281	PTRIQ6	0282	PTRIQ7	0283	PTRLP	E8D1
PUTCHR	000B	PUTREC	0009	RADFL0	00FB	RADON	0000
RAHLO	0004	RAMSIZ	02E4	RAHTOP	006A	RANDOM	D20A
RBLOKV	E47A	RDONLY	00B7	RECVDN	0039	REDGE	0027
RENAME	0020	RMARON	0053	ROWAC	0070	ROWCRB	0054
ROWINC	0079	RTCLOK	0012	SAVADR	006B	SAVID	0316
SAVHBC	0058	SCOLLP	E834	SCREDT	0045	SCRENV	E410
SCRFL0	02B8	SCRHEM	0093	SDL8TH	0231	SDL8TL	0230
SDMCTL	022F	BENDEV	E468	BERIN	D20D	BEROUT	D20D
SETVB1	E93D	SETVBL	E912	SETVBV	E45C	SHFAHT	006F
SHFLOK	02BE	SIN	BDB1	SIOINV	E465	SIOORG	E944
SIOV	E459	SIZEM	D00C	SIZEP0	D00B	SIZEP1	D009
SIZEP2	D00A	SIZEP3	D00B	SRCTL	D20F	SKRES	D20A
SKSTAT	D20F	SOUNDR	0041	SPECIL	000E	SGR	BEB1
SRTIM2	0006	SRTIMR	022B	SBFLA0	02FF	S8KCTL	0232
STACKP	0318	STATIB	000D	STATUS	0030	STICK0	027B
STICK1	0279	STICK2	027A	STICK3	027B	STIMER	D209
STLOOP	E89C	STRIQ0	0284	STRIQ1	0285	STRIQ2	0286
STRIQ3	0287	STRL	E8B5	SUBTMP	029E	SUCCESS	0001

INTERRUPT HANDLER

SV1H	00E9	SV1L	003D	SV7H	00EB	SV7L	0098
SWPFL0	0078	SVIR0	E6F6	SVIR01	E70B	SVIR02	E71F
SVIR03	E737	SVIR04	E74A	SVIR05	E75A	SVIR06	E767
SVIR07	E779	SVIR08	E793	SVIR09	E79E	SVIR0A	E7A9
SVIR0B	E782	SVRT1	E783	SVSVB1	E7DD	SVSVB2	E7F9
SVSVB3	E808	SVSVB4	E857	SVSVB5	E8EB	SVSVB7	E898
SVSVBA	E869	SVSVBB	E859	SVSVBL	E7D1	SVSVBV	E45F
SVVB6A	E87C	TABMAP	02A3	TEMP	023E	TEMP1	0312
TEMP2	0314	TEMP3	0315	TIMER1	030C	TIMER2	0310
TIMFLO	0317	TIMOUT	008A	TINDEX	0293	TMPCHR	0050
TMPCOL	0289	TMPLBT	02A1	TMPROW	0288	TMPIX1	029C
TRAMS2	0006	TRIG0	D010	TRIG1	D011	TRIG2	D012
TRIG3	D013	TRNRCD	0089	TBTAT	0319	TSTDAT	0007
TXTCOL	0291	TXTMSC	0294	TXTOLD	0296	TXTRON	0290
UNLOCK	0024	USAREA	0480	VBATRA	E7EB	VBREAK	0206
VCDUNT	D408	VCTABL	E480	VDELAY	D01C	VDSLST	0200
VECTBL	E400	VIMIR0	0216	VINTER	0204	VKEYBD	0208
VPRCED	0202	VBCROL	D405	VSERIN	020A	VBEROC	020E
VSEROR	020C	VTIMR1	0210	VTIMR2	0212	VTIMR4	0214
VVBLKD	0224	VVBLK1	0222	WARMST	000B	WARMBV	E474
WMODE	0289	WRONLY	0083	WBYNC	D40A	XITVBL	E93E
XITVBV	E462	XMTDDN	003A	XXIT	E805	ZI0CB	0020
ZTEMP1	00F5	ZTEMP3	00F9	ZTEMP4	00F7		

810 (SERIAL BUS INPUT/OUTPUT CONTROLLER)

810 (SERIAL BUS INPUT/OUTPUT CONTROLLER)

TITLE "810 (SERIAL BUS INPUT/OUTPUT CONTROLLER)"
COLLEEN OPERATING SYSTEM

810 (SERIAL BUS INPUT/OUTPUT CONTROLLER)
WITH SOFTWARE BAUD RATE CORRECTION ON CASSETTE

AL MILLER 3-APR-79

THIS MODULE HAS ONE ENTRY POINT. IT IS CALLED BY THE DEVICE
HANDLER. IT INTERPRETS A PREVIOUSLY ESTABLISHED DEVICE CONTROL
BLOCK (STORED IN GLOBAL RAM) TO ISSUE COMMANDS
TO THE SERIAL BUS TO CONTROL TRANSMITTING AND RECEIVING DATA.

SIO (SERIAL BUS INPUT/OUTPUT CONTROLLER)

EQUATES

DCD DEVICE BUS ID NUMBERS

0030 FLOPPY = \$30
0040 PRINTR = \$40
0060 CASSET = \$60

BUS COMMANDS

0052 READ = 'R'
0057 WRITE = 'W'
0053 STATUS = 'S'
0021 FORMAT = 'I'

COMMAND AUX BYTES

0053 BLDWAY = 'S' PRINT 16 CHARACTERS SIDEWAYS
004E NORMAL = 'N' PRINT 40 CHARACTERS NORMALLY
0044 DOUBLE = 'D' PRINT 20 CHARACTERS DOUBLE WIDE
0050 PLOT = 'P' PLOT MODE

BUS RESPONSES

0041 ACK = 'A' DEVICE ACKNOWLEDGES INFORMATION
004E NACK = 'N' DEVICE DID NOT UNDERSTAND
0043 COMPLT = 'C' DEVICE SUCCESSFULLY COMPLETED OPERATION
0045 ERROR = 'E' DEVICE INCURRED AN ERROR IN AN ATTEMPTED OPERATION

MISCELLANEOUS EQUATES

0028 B192LO = \$28 19200 BAUD RATE POKEY COUNTER VALUES (LO BYTE)
0000 B192HI = \$00 (HI BYTE)
00CC B600LO = \$CC 600 BAUD (LO BYTE)
0005 B600HI = \$05 (HI BYTE)
0005 HITONE = \$05 F8K HI FREQ POKEY COUNT VALUE (5326 HZ)
0007 LOTONE = \$07 F8K LO FREQ POKEY COUNTER VALUE (3995 HZ)

00B4 WIRLO = 180 WRITE INTER RECORD GAP (IN 1/60 SEC)
0000 WIRHI = 0
007B RIRLO = 120 READ INTER RECORD GAP (IN 1/60 SEC)
0000 RIRHI = 0
000F WSIRO = 15 SHORT WRITE INTER RECORD GAP
000A RSIRO = 10 SHORT READ INTER RECORD GAP

0034 NCOMLO = \$34 PIA COMMAND TO LOWER NOT COMMAND LINE
003C NCOMHI = \$3C PIA COMMAND TO RAISE NOT COMMAND LINE
0034 MOTRGO = \$34 PIA COMMAND TO TURN ON CASSETTE MOTOR
003C MOTRST = \$3C PIA COMMAND TO TURN OFF MOTOR

0002 TEMPHI = TEMP/256 ADDRESS OF TEMP CELL (HI BYTE)
003E TEMPLO = -256*TEMPHI+TEMP (LO BYTE)
0002 CBUFHI = CDEVIC/256 ADDRESS OF COMMAND BUFFER (HI BYTE)
003A CBUFLO = -256*CBUFHI+CDEVIC (LO BYTE)

000D CRETRI = 13 NUMBER OF COMMAND FRAME RETRIES
0001 DRETRI = 1 NUMBER OF DEVICE RETRIES

SIO (SERIAL BUS INPUT/OUTPUT CONTROLLER)

0002
0000

CTIMLO = 2 COMMAND FRAME ACK TIME OUT (LO BYTE)
CTIMHI = 0 COMMAND FRAME ACK TIME OUT (HI BYTE)

00EB
00F0

JTADRH = JTIMER/256 HI BYTE OF JUMP TIMER ROUTINE ADDR
JTADRL = -256*JTADRH+JTIMER

SIO (SERIAL BUS INPUT/OUTPUT CONTROLLER)

SIO

E459 4C 59 E9 * = SIOV
JMP SIO SIO ENTRY POINT

E465 4C 44 E9 * = SIOINV
JMP SIOINT SIO INITIALIZATION ENTRY POINT

E46B 4C F6 EB * = SENDEV
JMP SENDEN SEND ENABLE ENTRY POINT

==VCTABL-INTAB8+VBERIN

E4BA 11 EB .WORD ISR5IR VSERIN
E4BC 90 EA .WORD ISR0DN VBEROR
E4BE D1 EA .WORD ISR7D VSEROC

==SIOORG

SIO INITIALIZATION SUBROUTINE

E944 A9 3C SIOINT LDA #MOTRST
E946 BD 02 D3 STA PACTL TURN OFF MOTOR

E949 A9 3C LDA #NCOMHI
E94B BD 03 D3 STA PBCTL RAISE NOT COMMAND LINE

E94E A9 03 LDA #3
E950 BD 32 02 STA SBKCTL GET POKEY OUT OF INITIALIZE MODE
E953 B5 41 STA SOUNDR INIT POKE ADDRESS FOR QUIET I/O
E955 BD 0F D2 STA SBKCTL

E95B 60 RTS RETURN

E959 BA SIO TSX
E95A BE 18 03 BTX STACKP SAVE STACK POINTER
E95D A9 01 LDA #1
E95F B5 42 STA CRITIC

E961 AD 00 03 LDA DDEVIC
E964 C9 60 CMP #CASSET
E966 D0 03 BNE NOTCST BRANCH IF NOT CASSETTE
E96B 4C B4 EB JMP CASSET OTHERWISE JUMP TO CASSETTE ENTER

ALL DEVICES EXCEPT CASSETTE ARE INTELLIGENT

E96B A9 00 NOTCST LDA #0
E96D BD 0F 03 STA CASFLG INIT CASSETTE FLAG TO NO CASSETTE

E970 A9 01 LDA #DRETRI SET NUMBER OF DEVICE RETRIES

B10 (SERIAL BUS INPUT/OUTPUT CONTROLLER)

E972 B5 37		BTA	DRETRY	
E974 A9 0D	COMMND	LDA	#CRETR1	SET NUMBER OF COMMAND FRAME RETRIES
E976 B5 36		BTA	CRETRY	

SEND A COMMAND FRAME

E978 A9 28	COMFRM	LDA	#B192LO	SET BAUD RATE TO 19200
E97A 8D 04 D2		BTA	AUDF3	
E97D A9 00		LDA	#B192HI	
E97F 8D 06 D2		BTA	AUDF4	

E982 18		CLC		SET UP COMMAND BUFFER
E983 AD 00 03		LDA	DDEVIC	
E986 6D 01 03		ADC	DUNIT	
E989 6F FF		ADC	#FF	SUBTRACT 1
E98B 8D 3A 02		BTA	CDEVIC	SET BUS ID NUMBER

E98E AD 02 03		LDA	DCOMND	
E991 8D 3B 02		BTA	CCOMND	SET BUS COMMAND

E994 AD 0A 03		LDA	DAUX1	STORE COMMAND FRAME AUX BYTES 1 AND 2
E997 8D 3C 02		BTA	CAUX1	
E99A AD 0B 03		LDA	DAUX2	
E99D 8D 3D 02		BTA	CAUX2	DONE BETTING UP COMMAND BUFFER

E9A0 18		CLC		SET BUFFER POINTER TO COMMAND FRAME BUFFER
E9A1 A9 3A		LDA	#CBUFLO	
E9A3 B5 32		BTA	BUFRLO	AND BUFFER END ADDRESS
E9A5 69 04		ADC	#4	
E9A7 B5 34		BTA	BFENLO	
E9A9 A9 02		LDA	#CBUFHI	
E9AB B5 33		BTA	BUFRHI	
E9AD B5 35		BTA	BFENHI	DONE BETTING UP BUFFER POINTER

E9AF A9 34		LDA	#NCOMLO	LOWER NOT COMMAND LINE
E9B1 8D 03 D3		BTA	PBCTL	

E9B4 20 BE EC		JBR	SENDIN	SEND THE COMMAND FRAME TO A SMART DEVICE
---------------	--	-----	--------	--

E9B7 AD 3F 02		LDA	ERRFLQ	
E9BA D0 03		BNE	BADCOM	BRANCH IF AN ERROR RECEIVED

E9BC 98		TYA		
E9BD D0 07		BNE	ACKREC	BRANCH IF ACK RECEIVED

E9BF C6 36	BADCOM	DEC	CRETRY	A NACK OR TIME OUT OCCURED
E9C1 10 B5		BPL	COMFRM	BO BRANCH IF ANY RETRIES LEFT

E9C3 4C 06 EA		JMP	DERR1	OTHERWISE, JUMP TO RETURN SECTION
---------------	--	-----	-------	-----------------------------------

E9C6 AD 03 03	ACKREC	LDA	DBTATS	ACK WAS RECEIVED
E9C9 10 0C		BPL	WATCOM	BRANCH TO WAIT FOR COMPLETE , IF THERE IS NO DATA TO BE SENT

SEND A DATA FRAME TO PERIPHERAL

SIO (SERIAL BUS INPUT/OUTPUT CONTROLLER)

E9CB A9 0D LDA #CRETRI SET NUMBER OF RETRIES
E9CD B5 36 STA CRETRY

E9CF 20 6E EB JSR LDPNTR LOAD BUFFER POINTER WITH DGB INFORMATION

E9D2 20 8E EC JSR SENDIN GO SEND THE DATA FRAME TO A SMART DEVICE

E9D5 F0 EB BEQ BADCOM BRANCH IF BAD

WAIT FOR COMPLETE SIGNAL FROM PERIPHERAL

E9D7 20 79 EC WATCOM JSR BTMOT SET DDEVICE TIME OUT VALUES IN Y,X

E9DA A9 00 LDA #000
E9DC 8D 3F 02 STA ERRFLO CLEAR ERROR FLAG

E9DF 20 9F EC JSR WAITER SET UP TIMER AND WAIT
E9E2 F0 12 BEQ DERR BRANCH IF TIME OUT

DEVICE DID NOT TIME OUT

E9E4 2C 03 03 BIT DBSTATS
E9E7 70 07 BVS MODATA BRANCH IF MORE DATA FOLLOWS

E9E9 AD 3F 02 LDA ERRFLO
E9EC D0 1B BNE DERR1 BRANCH IF AN ERROR OCCURRED
E9EE F0 1D BEQ RETURN OTHERWISE RETURN

RECEIVE A DATA FRAME FROM PERIPHERAL

E9F0 20 6E EB MODATA JSR LDPNTR LOAD BUFFER POINTER WITH DGB INFORMATION

E9F3 20 E2 EA JSR RECEIV GO RECEIVE A DATA FRAME

E9F6 AD 3F 02 DERR LDA ERRFLO
E9F9 F0 05 BEQ NOTERR BRANCH IF NO ERROR PRECEDED DATA

E9FB AD 19 03 LDA TSTAT GET TEMP STATUS
E9FE B5 30 STA STATUS STORE IN REAL STATUS

EA00 A5 30 NOTERR LDA STATUS
EA02 C9 01 CMP #SUCCEB
EA04 F0 07 BEQ RETURN BRANCH IF COMPLETELY SUCCESSFUL

EA06 C6 37 DERR1 DEC DRETRY
EA0B 30 03 BMI RETURN BRANCH IF OUT OF DEVICE RETRIES

EA0A 4C 74 E9 JMP COMND OTHERWISE, ONE MORE TIME

EA0D 20 63 EC RETURN JSR SENDDB DISABLE POKEY INTERRUPTS

BIO (SERIAL BUS INPUT/OUTPUT CONTROLLER)

EA10 A9 00	LDA	#0	
EA12 B5 42	STA	CRITIC	
EA14 A4 30	LDV	STATUS	RETURN STATUS IN Y
EA16 BC 03 03	STY	DSTATS	AND THE DCB STATUS WORD
EA19 60	RTS	RETURN	

WAIT SUBROUTINE

WAITS FOR COMPLETE OR ACK
RETURNS Y=0FF IF SUCCESSFUL, Y=000 IF NOT

EA1A A9 00	WAIT	LDA	#000	
EA1C BD 3F 02		STA	ERRFLO	CLEAR ERROR FLAG

EA1F 18	CLC		LOAD BUFFER POINTER WITH ADDRESS
EA20 A9 3E	LDA	#TEMPLO	OF TEMPORARY RAM CELL
EA22 B5 32	STA	BUFRLO	
EA24 69 01	ADC	#1	
EA26 B5 34	STA	BFENLO	ALSO SET BUFFER END +1 ADDRESS
EA28 A9 02	LDA	#TEMPHI	
EA2A B5 33	STA	BUFRHI	
EA2C B5 35	STA	BFENHI	DONE LOADING POINTER

EA2E A9 FF	LDA	#0FF	
EA30 B5 3C	STA	NOCKSM	SET NO CHECKSUM FOLLOWS DATA FLAG

EA32 20 E2 EA	JBR	RECEIV	00 RECEIVE A BYTE
---------------	-----	--------	-------------------

EA35 A0 FF	LDY	#0FF	ASSUME SUCCESS
EA37 A5 30	LDA	STATUS	
EA39 C9 01	CMF	#SUCCEB	
EA3B D0 19	BNE	NWOK	BRANCH IF IT DID NOT WORK OK

EA3D AD 3E 02	WOK	LDA	TEMP	MAKE SURE THE BYTE SUCCESSFULLY RECEIVED
EA40 C9 41		CMF	#ACK	HAS ACTUALLY AN ACK OR COMPLETE

EA42 F0 21	BEG	GOOD	
EA44 C9 43	CMF	#COMPLT	
EA46 F0 1D	BEG	GOOD	

EA48 C9 45	CMF	#ERROR	
EA4A D0 06	BNE	NOTDER	BRANCH IF DEVICE DID NOT SEND BACK A DEVICE ERROR CODE

EA4C A9 90	LDA	#DERROR	
EA4E B5 30	STA	STATUS	SET DEVICE ERROR STATUS
EA50 D0 04	BNE	NWOK	

EA52 A9 8B	NOTDER	LDA	#DNACK	OTHERWISE SET NACK STATUS
EA54 B5 30		STA	STATUS	

EA56 A5 30	NWOK	LDA	STATUS	
EA58 C9 8A		CMF	#TIMOUT	
EA5A F0 07		BEG	BAD	BRANCH IF TIME OUT

EA5C A9 FF	LDA	#0FF	
------------	-----	------	--

810 (SERIAL BUS INPUT/OUTPUT CONTROLLER)

EA5E 8D 3F 02
EA61 D0 02

STA
BNE

ERRFLO
GOOD

SET SOME ERROR FLAG
RETURN WITH OUT SETTING Y = 0

EA63 A0 00

BAD

LDY

#0

EA65 A5 30
EA67 8D 19 03
EA6A 60

GOOD

LDA
STA
RTB

STATUS
TSTAT

RETURN

SEND SUBROUTINE

SENDS A BUFFER OF BYTES OUT OVER THE SERIAL BUS

EA6B A9 01
EA6D 85 30

SEND

LDA
STA

#SUCCEB ASSUME SUCCEB
STATUS

EA6F 20 F6 EB

JBR

BENDEN ENABLE SENDING

EA72 A0 00
EA74 84 31
EA76 84 3B
EA78 84 3A

LDY
STY
BTY
BTY

#0
CHKSUM
CHKSENT
XMTDON

CLEAR CHECK SUM
CHECKSUM SENT FLAG
TRANSMISSION DONE FLAG

EA7A B1 32
EA7C 8D 0D D2

LDA
STA

(BUFRLD),Y
SEROUT

PUT FIRST BYTE FROM BUFFER
INTO THE SERIAL OUTPUT REGISTER

EA7F 85 31

STA

CHKSUM PUT IT IN CHECKSUM

EA81 A5 11
EA83 D0 03
EA85 4C A4 ED

NOTDON

LDA
BNE
JMP

BRKKEY
NTBRKO
BROKE

JUMP IF BREAK KEY PRESSED

EA8B A5 3A
EA8A F0 F5

NTBRKO

LDA
BEQ

XMTDON
NOTDON

LOOP UNTIL TRANSMISSION IS DONE

EA8C 20 63 EC

JBR

SENDD8 DISABLE SENDING

EA8F 60

RTB

RETURN

OUTPUT DATA NEEDED INTERRUPT SERVICE ROUTINE

EA90 98
EA91 48

ISRDON

TYA
PHA

SAVE Y REG ON STACK

EA92 E6 32
EA94 D0 02
EA96 E6 33

INC
BNE
INC

BUFRLD
NOWRPO
BUFRLH

INCREMENT BUFFER POINTER

SIO (SERIAL BUS INPUT/OUTPUT CONTROLLER)

EA98 A5 33	NOWRPO	LDA	BUFRHI	CHECK IF PAST END OF BUFFER
EA9A C5 35		CMP	BFENHI	
EA9C 90 22		BCC	NOTEND	
EA9E A5 32		LDA	BUFRLO	
EAA0 C5 34		CMP	BFENLO	
EAA2 90 1C		BCC	NOTEND	BRANCH IF NOT PAST END OF BUFFER

EAA4 A5 38		LDA	CHKSENT	
EAA6 D0 08		BNE	RELDONE	BRANCH IF CHECKSUM ALREADY SENT

EAA8 A5 31		LDA	CHKSUM	
EAAA 8D 0D D2		BTA	SEROUT	SEND CHECK SUM
EAAD A9 FF		LDA	#\$FF	
EAAF 85 38		BTA	CHKSENT	SET CHECKSUM SENT FLAG
EAB1 D0 09		BNE	CHKDON	

EAB3 A5 10	RELDONE	LDA	POKMSK	ENABLE TRANSMIT DONE INTERRUPT
EAB5 09 08		ORA	#\$08	
EAB7 85 10		BTA	POKMSK	
EAB9 8D 0E D2		BTA	IRGEN	

EABC 68	CHKDON	PLA		
EABD A8		TAY		RESTORE Y REG
EABE 68		PLA		RETURN FROM INTERRUPT
EABF 40		RTI		

EAC0 A0 00	NOTEND	LDY	#0	
EAC2 B1 32		LDA	(BUFRLO), Y	PUT NEXT BYTE FROM BUFFER
EAC4 8D 0D D2		BTA	SEROUT	INTO THE SERIAL OUTPUT REGISTER

EAC7 18		CLC		ADD IT TO CHECKSUM
EAC8 65 31		ADC	CHKSUM	
EACA 69 00		ADC	#0	
EACC 85 31		BTA	CHKSUM	

EACE 4C BC EA		JMP	CHKDON	GO RETURN
---------------	--	-----	--------	-----------

TRANSMIT DONE INTERRUPT SERVICE ROUTINE

EAD1 A5 38	ISRTD	LDA	CHKSENT	
EAD3 F0 08		BEG	FOOEY	BRANCH IF CHECKSUM NOT YET SENT

EAD5 85 3A		BTA	XMTDON	OTHERWISE SET TRANSMISSION DONE FLAG
------------	--	-----	--------	--------------------------------------

EAD7 A5 10		LDA	POKMSK	DISABLE TRANSMIT DONE INTERRUPT
EAD9 29 F7		AND	#\$F7	
EADB 85 10		BTA	POKMSK	
EADD 8D 0E D2		BTA	IRGEN	

EAE0 68	FOOEY	PLA		RETURN FROM INTERRUPT
EAE1 40		RTI		

SIO (SERIAL BUS INPUT/OUTPUT CONTROLLER)

RECEIVE SUBROUTINE

EAE2 A9 00	RECEIV	LDA	NO	
EAE4 AC 0F 03	LDY	CASFLO		
EAE7 D0 02	BNE	NOCLR	BRANCH IF CASSETTE	
EAE9 85 31	STA	CHKBUM	CLEAR	CHKBUM
EAE8 85 38	NOCLR	STA	BUFRL	BUFFER FULL FLAG
EAE8 85 39	STA	RECVDN		RECEIVE DONE FLAG

EAEF A9 01	LDA	#SUCCEB		
EAF1 85 30	STA	STATUS	SET GOOD STATUS FOR DEFAULT CASE.	
EAF3 20 1F EC	JBR	RECVEN	DO RECEIVE ENABLE	
EAF6 A9 3C	LDA	#NCOMHI	COMMAND FRAME HI COMMAND	
EAF8 8D 03 D3	STA	PBCTL	STORE IN PIA	
EAFB A5 11	LDA	BRKKEY		
EAFD D0 03	BNE	NTBRK1		
EAFF 4C A4 ED	JMP	BROKE	JUMP IF BREAK KEY PRESSED	

EB02 AD 17 03	NTBRK1	LDA	TIMFLO	NO.
EB05 F0 05	BEG	TOUT	IF TIMEOUT, GO SET ERROR STATUS	
EB07 A5 39	LDA	RECVDN		
EB09 F0 F0	BEG	CHMTIM	DONE ?	
EB0B 60	GOBACK	RTS		
EB0C A9 8A	TOUT	LDA	STINDUT	YES.
EB0E 85 30	STA	STATUS	SET TIMEOUT STATUS	

EB10 60	RRETRN	RTS	RETURN
---------	--------	-----	--------

SERIAL INPUT READY INTERRUPT SERVICE ROUTINE

EB11 98	ISRBR	TYA	
EB12 48	PHA		SAVE Y REG ON STACK

EB13 AD 0F D2	LDA	SKSTAT	
EB16 8D 0A D2	STA	SKRES	RESET STATUS REGISTER

***** THIS MAY NOT BE THE PLACE TO DO IT *****

EB19 30 04	BMI	NTFRAN	BRANCH IF NO FRAMING ERROR
------------	-----	--------	----------------------------

SIO (SERIAL BUS INPUT/OUTPUT CONTROLLER)

EB1B A0 8C LDY #FRMERR
EB1D B4 30 STY STATUS SET FRAME ERRORR STATUS

EB1F 29 20 NTFRAM AND #*20
EB21 D0 04 BNE NTOVRN BRANCH IF NO OVERRUN ERROR

EB23 A0 8E LDY #OVRUN
EB25 B4 30 STY STATUS SET OVERRUN ERROR STATUS

EB27 A5 38 NTOVRN LDA BUFRFL
EB29 F0 13 BEQ NOTYET BRANCH IF BUFFER WAS NOT YET FILLED

EB2B AD 0D D2 LDA SERIN THIS INPUT BYTE IS THE CHECKSUM
EB2E C5 31 CMP CHKSUM
EB30 F0 04 BEQ BRETRN BRANCH IF CHECKSUMS MATCH

EB32 A0 8F LDY #CHKERR
EB34 B4 30 STY STATUS SET CHECKSUM ERROR STATUS

EB36 A9 FF BRETRN LDA #*FF SET RECEIVE DONE FLAG
EB38 B5 39 STA RECVDN

EB3A 68 SUBUAL PLA
EB3B A8 TAY RESTORE Y REG
EB3C 68 PLA RETURN FROM INTERRUPT
EB3D 40 RTI

EB3E AD 0D D2 NOTYET LDA SERIN
EB41 A0 00 LDY #0
EB43 91 32 STA (BUFRLO),Y STORE INPUT REGISTER INTO BUFFER

EB45 18 CLC ADD IT TO CHECKSUM

EB46 65 31 ADC CHKSUM
EB48 69 00 ADC #0
EB4A B5 31 STA CHKSUM

EB4C E6 32 INC BUFRLO INCREMENT BUFFER POINTER
EB4E D0 02 BNE NTWRP1
EB50 E6 33 INC BUFRHI

EB52 A5 33 NTWRP1 LDA BUFRHI
EB54 C5 35 CMP BFENHI
EB56 90 E2 BCC SUBUAL
EB58 A5 32 LDA BUFRLO
EB5A C5 34 CMP BFENLO
EB5C 90 DC BCC SUBUAL BRANCH IF NEW BUFFER ADDRESS IS IN BUFFER LIMITS

EB5E A5 3C LDA NOCKSM
EB60 F0 06 BEQ QODN BRANCH IF A CHECKSUM WILL FOLLOW DATA

EB62 A9 00 LDA #0
EB64 B5 3C STA NOCKSM CLEAR NO CHECKSUM FLAG

EB66 F0 CE BEQ BRETRN GO RETURN AND SET RECEIVE DONE FLAG

EB68 A9 FF QODN LDA #*FF

BIO (SERIAL BUS INPUT/OUTPUT CONTROLLER)

EB6A 85 38 STA BUFRFL SET BUFFER FULL FLAG

EB6C D0 CC BNE SUBUAL GO RETURN

LOAD BUFFER POINTER SUBROUTINE

LOAD BUFFER POINTER WITH DCB BUFFER INFORMATION

LDPNTR

CLC

DBUFLO

LDA

BUFRLO

STA

DBYTLO

ADC

BFENLO

ALSO SET BUFFER END + 1 ADDRESS

EB74 6D 08 03

EB77 85 34

EB79 AD 05 03

EB7C 85 33

EB7E 6D 09 03

EB81 85 35

LDA

DBUFHI

STA

BUFRHI

ADC

DBYTHI

STA

BFENHI

EB83 60

RTS

RETURN

CASSETTE HANDLING CODE

EB84 AD 03 03

EB87 10 2E

CASENT

LDA

DBSTAT

BPL

CASRED

BRANCH IF INPUT FROM CASSETTE

WRITE A RECORD

EB89 A9 CC

EB8B 8D 04 D2

EB8E A9 05

EB90 8D 06 D2

LDA

#B600L0 SET BAUD RATE TO 600

STA

AUDF3

LDA

#B600H1

STA

AUDF4

EB93 20 F6 EB

JBR

BENDEN TURN ON POKEY MARK TONE

EB96 A0 0F

EB98 AD 08 03

EB9B 30 02

LDY

#WBIR0 LOAD SHORT WRITE INTER RECORD GAP TIME

LDA

DAUX2

BMI

SRTIRO BRANCH IF SHORT GAP IS DESIRED

EB9D A0 B4

EB9F A2 00

EBA1 20 BD ED

SRTIRO

LDY

#WIR0L0 SET WRITE IRO TIME

LDX

#WIRCHI

JBR

SETVBX

EBA4 A9 34

EBA6 8D 02 D3

LDA

#MOTRO0

STA

PACTL

TURN ON MOTOR

S10 (SERIAL BUS INPUT/OUTPUT CONTROLLER)

EBA9 AD 17 03	TIMIT	LDA	TIMFLO	LOOP UNTIL DONE
EBAC D0 FB		BNE	TIMIT	
EBAE 20 6E EB		JBR	LDPNTR	LOAD BUFFER POINTER WITH DCB INFORMATION
EBB1 20 6B EA		JBR	SEND	SEND A BUFFER
EBB4 4C E3 EB		JMP	CRETRN	GO RETURN

RECEIVE A RECORD

EBB7 A9 FF	CASRED	LDA	#OFF	
EBB9 8D 0F 03		STA	CASFLO	SET SET CASSETTE FLAG
EBBC A0 0A		LDY	#R8IR0	LOAD SHORT READ INTER RECORD GAP TIME
EBDE AD 0B 03		LDA	DAUX2	
EBC1 30 02		BMI	SRTIR1	BRANCH IF SHORT GAP IS DESIRED
EBC3 A0 7B		LDY	#R1R0L0	SET TIME OUT FOR READ IRQ
EBC5 A2 00	SRTIR1	LDX	#R1R0H1	
EBC7 20 8D ED		JBR	SETVBX	

EBCA A9 34		LDA	#MOTR00	
EBCC 8D 02 D3		STA	PACTL	TURN ON MOTOR

EBCF AD 17 03	TIMIT1	LDA	TIMFLO	LOOP UNTIL DONE
EBD2 D0 FB		BNE	TIMIT1	

EBD4 20 6E EB		JBR	LDPNTR	LOAD BUFFER POINTER WITH DCB INFORMATION
---------------	--	-----	--------	--

EBD7 20 79 EC		JBR	BTMT01	SET DEVICE TIME OUT IN Y.X
EBDA 20 8D ED		JBR	SETVBX	

EBDD 20 14 ED		JBR	BEGIN	SET INITIAL BAUD RATE
---------------	--	-----	-------	-----------------------

EBE0 20 E2 EA		JBR	RECEIV	GO RECEIVE A BLOCK
---------------	--	-----	--------	--------------------

EBE3 AD 0B 03	CRETRN	LDA	DAUX2	
EBE6 30 05		BMI	SRTIR2	BRANCH IF DOING SHORT INTER RECORD GAPS

EBE8 A9 3C		LDA	#MOTRST	DON'T TURN OFF CASSETTE MOTOR
EBEA 8D 02 D3		STA	PACTL	TURN OFF MOTOR

EBED 4C 0D EA	SRTIR2	JMP	RETURN	GO RETURN
---------------	--------	-----	--------	-----------

EBF0 A9 00	UTIMER	LDA	#000	
EBF2 8D 17 03		STA	TIMFLO	SET TIME OUT FLAG
EBF5 60		RTS		

SIO (SERIAL BUS INPUT/OUTPUT CONTROLLER)

SEND ENABLE SUBROUTINE

EBF6 A9 07	SENDEN	LDA	##07	MASK OFF PREVIOUS SERIAL BUS CONTROL BITS
EBF8 2D 32 02		AND	SSKCTL	
EBF8 09 20		ORA	##20	SET TRANSMIT MODE

EBFD AC 00 03		LDY	DDEVIC	
EC00 C0 60		CPY	#CABSET	
EC02 D0 0C		BNE	NOTCAS	BRANCH IF NOT CASSETTE

EC04 09 08		ORA	##08	SET THE FSK OUTPUT BIT
------------	--	-----	------	------------------------

EC06 A0 07		LDY	#LOTONE	SET FSK TONE FREQUENCIES
EC08 8C 02 D2		BTY	AUDF2	
EC08 A0 05		LDY	#HITONE	
EC0D 8C 00 D2		BTY	AUDF1	

EC10 8D 32 02	NOTCAS	BTA	SSKCTL	STORE NEW VALUE TO SYSTEM MASK
EC13 8D 0F D2		BTA	SKCTL	STORE TO ACTUAL REGISTER

EC16 A9 C7		LDA	##C7	MASK OFF PREVIOUS SERIAL BUS INTERRUPT BITS
EC18 25 10		AND	POKMSK	
EC1A 09 10		ORA	##10	ENABLE OUTPUT DATA NEEDED INTERRUPT

EC1C 4C 35 EC		JMP	CONTIN	GO CONTINUE IN RECEIVE ENABLE SUBROUTINE
---------------	--	-----	--------	--

RECEIVE ENABLE SUBROUTINE

EC1F A9 07	RECVEN	LDA	##07	MASK OFF PREVIOUS SERIAL BUS CONTROL BITS
EC21 2D 32 02		AND	SSKCTL	
EC24 09 10		ORA	##10	SET RECEIVE MODE ASYNCH.

EC26 8D 32 02		BTA	SSKCTL	STORE NEW VALUE TO SYSTEM MASK
EC29 8D 0F D2		BTA	SKCTL	STORE TO ACTUAL REGISTER

EC2C 8D 0A D2		BTA	SKRES	RESET SERIAL PORT/KEYBOARD STATUS REGISTER
---------------	--	-----	-------	--

EC2F A9 C7		LDA	##C7	MASK OFF PREVIOUS SERIAL BUS INTERRUPT BITS
EC31 25 10		AND	POKMSK	
EC33 09 20		ORA	##20	ENABLE RECEIVE INTERRUPT
EC35 85 10	CONTIN	BTA	POKMSK	STORE NEW VALUE TO SYSTEM MASK
EC37 8D 0E D2		BTA	IRGEN	STORE TO ACTUAL REGISTER

EC3A A9 28		LDA	##28	CLOCK CH. 3 WITH 1.79 MHZ
EC3C 8D 08 D2		BTA	AUDCTL	CLOCK CH. 4 WITH CH. 3

EC3F A2 06		LDX	#6	SET PURE TONES, NO VOLUME
EC41 A7 A8		LDA	##A8	
EC43 A4 41		LDY	SOUNDR	TEST QUIET I/O FLAG
EC45 D0 02		BNE	NOISE1	NE IS NORMAL (NOISY)

810 (SERIAL BUS INPUT/OUTPUT CONTROLLER)

EC47 A9 A0	LDA	#9A0	
EC49 9D 01 D2	STA	AUDC1, X	
EC4C CA	DEX		
EC4D CA	DEX		
EC4E 10 F9	BPL	NOISE1	

EC50 A9 A0	LDA	#9A0	
EC52 BD 03 D2	STA	AUDC3	TURN OFF SOUND ON CHANNEL 3
EC55 AC 00 03	LDY	DDEVIC	
EC58 C0 60	CPY	#CASSET	
EC5A F0 06	BEG	CAS31	BRANCH IF CASSETTE IS DESIRED
EC5C BD 01 D2	STA	AUDC1	OTHERWISE TURN OFF CHANNELS 1 AND 2
EC5F BD 03 D2	STA	AUDC2	

EC62 60	CAS31	RTS	RETURN
---------	-------	-----	--------

DISABLE SEND AND DISABLE RECEIVE SUBROUTINES

EC63 EA	BENDDS	NOP	
EC64 A9 C7	RECVDS	LDA	#C7 MASK OFF SERIAL BUS INTERRUPTS
EC66 25 10		AND	POKMBK
EC6B B5 10		STA	POKMBK STORE NEW VALUE TO SYSTEM MASK
EC6A BD 0E D2		STA	IRGEN STORE TO ACTUAL REGISTER

EC6D A2 06	LDX	#6	
EC6F A9 00	LDA	#0	
EC71 9D 01 D2	STA	AUDC1, X	
EC74 CA	DEX		
EC75 CA	DEX		
EC76 10 F9	BPL	ZERIT	TURN OFF AUDIO VOLUME

EC7B 60	RTS	RETURN
---------	-----	--------

GET DDEVICE TIME OUT VALUES IN Y, X SUBROUTINE

EC79 AD 06 03	BTTMOT	LDA	DTIMLO	GET DEVICE TIME OUT IN 1 SECOND INCR
EC7C 6A		ROR	A	PUT 6 HI BITS IN X, LO 2 BITS IN Y
EC7D 6A		ROR	A	
EC7E AB		TAY		TEMP BAVE
EC7F 29 3F		AND	#3F	MASK OFF 2 HI BITS
EC81 AA		TAX		THIS IS HI BYTE OF TIME OUT

SIO (SERIAL BUS INPUT/OUTPUT CONTROLLER)

ECB2 98	TYA	RESTORE
ECB3 6A	ROR A	
ECB4 29 C0	AND #C0	MASK OFF ALL BUT 2 HI BITS
ECB6 A8	TAY	THIS IS LO BYTE OF TIME OUT

ECB7 60	RTS
---------	-----

ECB8 11 EB	INTTBL	WORD	IBRBIR	SERIAL INPUT READY
ECB8 90 EA		WORD	IBRODN	OUTPUT DATA NEEDED
ECBC D1 EA		WORD	IBRTD	TRANSMISSION DONE

00EB	BIRHI	=	IBRBIR/256	SERIAL INPUT READY I8R ADDRESS
0011	BIRLO	=	-256*BIRHI+IBRBIR	
00EA	ODNHI	=	IBRODN/256	OUTPUT DATA NEEDED I8R ADDRESS
0090	ODNLO	=	-256*ODNHI+IBRODN	
00EA	TDHI	=	IBRTD/256	TRANSMISSION DONE I8R ADDRESS
00D1	TDLO	=	-256*TDHI+IBRTD	

SEND A DATA FRAME TO AN INTELLIGENT PERIPHERAL SUBROUTINE

ECBE A2 01	SENDIN	LDX	#01
EC90 A0 FF	DELAY0	LDY	#FF
EC92 88	DELAY1	DEY	
EC93 D0 FD		BNE	DELAY1
EC95 CA		DEX	
EC96 D0 F8		BNE	DELAY0

EC98 20 6B EA	JBR	SEND	GO SEND THE DATA FRAME
EC98 A0 02	LDY	#CTIMLO	SET ACK TIME OUT
EC9D A2 00	LDX	#CTIMHI	
EC9F 20 BD ED	WAITER	JBR	SETV8X

ECA2 20 1A EA	JBR	WAIT	WAIT FOR ACK
ECA3 98	TYA		IF Y=0, A TIME OUT OR NACK OCCURED
ECA6 60	RTS		RETURN

810 (SERIAL BUS INPUT/OUTPUT CONTROLLER)

COMPUTE VALUE FOR POKEY FREQ REGS FOR THE BAUD RATE AS
MEASURED BY AN INTERVAL OF THE 'VCOUNT' TIMER.

ECA7 8D 10 03	COMPUT	STA	TIMER2	
ECAA 8C 11 03		STY	TIMER2+1	SAVE FINAL TIMER VALUE
ECAD 20 08 ED		JSR	ADJUST	ADJUST VCOUNT VALUE
ECB0 8D 10 03		STA	TIMER2	SAVE ADJUSTED VALUE
ECB3 AD 0C 03		LDA	TIMER1	
ECB6 20 08 ED		JBR	ADJUST	ADJUST
ECB9 8D 0C 03		STA	TIMER1	SAVE ADJUSTED TIMER1 VALUE
ECBC AD 10 03		LDA	TIMER2	
ECBF 38		SEC		
ECC0 ED 0C 03		SBC	TIMER1	
ECC3 8D 12 03		STA	TEMP1	FIND VCOUNT DIFFERENCE
ECC6 AD 11 03		LDA	TIMER2+1	
ECC9 38		SEC		
ECCA ED 0D 03		SBC	TIMER1+1	
ECCD AB		TAY		FIND VBLANK COUNT DIFFERENCE
ECCE A9 7D		LDA	#-#B3	
ECD0 18	HITMR	CLC		
ECD1 69 B3		ADC	#B3	ACCUMULATE MULTIPLICATION
ECD3 88		DEY		
ECD4 10 FA		BPL	HITMR	DONE?
ECD6 18		CLC		
ECD7 6D 12 03		ADC	TEMP1	TOTAL VCOUNT DIFFERENCE
ECDA AB	FINDX	TAY		SAVE ACCUM
ECDB 4A		LSR	A	
ECDC 4A		LSR	A	
ECDD 4A		LSR	A	
ECDE 0A		ASL	A	
ECDF 38		SEC		
ECE0 E9 16		SBC	#22	ADJUST TABLE INDEX
ECE2 AA		TAX		DIVIDE INTERVAL BY 4 TO GET TABLE INDEX
ECE3 98		TYA		RESTORE ACCUM
ECE4 29 07		AND	#7	
ECE6 AB		TAY		PULL OFF 3 LB BITS OF INTERVAL
ECE7 A9 F5		LDA	#-11	
ECE9 18	DOINTP	CLC		
ECEA 69 08		ADC	#11	ACCUMULATE INTERPOLATION CONSTANT
ECEC 88		DEY		
ECED 10 FA		BPL	DOINTP	INTERPOLATION CONSTANT COMPUTATION DONE?
ECF7 10 03		BPL	PLUS	
ECF9 CE 0E 03		DEC	ADDCOR	
ECFC 18	PLUS	CLC		
ECFD 7D D2 ED		ADC	POKTAB, X	ADD CONSTANT TO LB BYTE TABLE VALUE
ED00 AB		TAY		LB BYTE POKEY FREQ VALUE
ED01 AD 0E 03		LDA	ADDCOR	
ED04 7D D3 ED		ADC	POKTAB+1, X	ADD CARRY TO HI BYTE TABLE VALUE
				HI BYTE POKEY FREQ VALUE

ED07 60

RTS

810 (SERIAL BUS INPUT/OUTPUT CONTROLLER)

ROUTINE TO ADJUST VCOUNT VALUE

ED08 C9 7C	ADJUST	CMP	#97C	
ED0A 30 04		BMI	ADJ1	LARGER THAN '7C' ?
ED0C 3B		BEC		YES,
ED0D E9 7C		BBC	#97C	
ED0F 60		RTS		
ED10 1B	ADJ1	CLC		
ED11 69 07		ADC	#97	
ED13 60		RTS		

INITIAL BAUD RATE MEASUREMENT -- USED TO SET THE
BAUD RATE AT THE START OF A RECORD.

IT IS ASSUMED THAT THE FIRST TWO BYTES OF EVERY
RECORD ARE 'AA' HEX.

ED14 A5 11	BEGIN	LDA	BRKKEY	
ED16 D0 03		BNE	NTBRK2	
ED18 4C A4 ED		JMP	BROKE	JUMP IF BREAK KEY PRESSED
ED1B 7B	NTBRK2	SEI		

ED1C AD 17 03		LDA	TIMFLO	
ED1F D0 02		BNE	OKTIM1	BRANCH IF NOT TIMED OUT
ED21 F0 25		BEG	TOUT1	BRANCH IF TIME OUT

ED23 AD 0F D2	OKTIM1	LDA	SKSTAT	
ED26 29 10		AND	#510	READ SERIAL PORT
ED28 D0 EA		BNE	BEGIN	START BIT?
ED2A BD 16 03		BTA	SAVIO	SAVE SER. DATA IN
ED2D AE 0B D4		LDX	VCOUNT	READ VERTICAL LINE COUNTER
ED30 A4 14		LDY	RTCLOCK+2	READ LO BYTE OF VBLANK CLOCK
ED32 BE 0C 03		STX	TIMER1	
ED35 BC 0D 03		STY	TIMER1+1	SAVE INITIAL TIMER VALUE

ED38 A2 01		LDX	#1	SET MODE FLAG
ED3A BE 15 03		BTX	TEMP3	
ED3D A0 0A		LDY	#10	SET BIT COUNTER FOR 10 BITS
ED3F A5 11	COUNT	LDA	BRKKEY	
ED41 F0 61		BEG	BROKE	BRANCH IF BREAK KEY PRESSED

ED43 AD 17 03		LDA	TIMFLO	
ED46 D0 04		BNE	OKTIMR	BRANCH IF NOT TIMED OUT
ED48 5B	TOUT1	CLI		
ED49 4C 0C EB		JMP	TOUT	BRANCH IF TIME OUT

ED4C AD 0F D2	OKTIMR	LDA	SKSTAT	
ED4F 29 10		AND	#510	READ SERIAL PORT
ED51 CD 16 03		CMP	SAVIO	DATA IN CHANGED YET?
ED54 F0 E9		BEG	COUNT	
ED56 BD 16 03		BTA	SAVIO	YES, SAVE SER. DATA IN
ED59 BB		DEY		DECR. BIT COUNTER

SIO (SERIAL BUS INPUT/OUTPUT CONTROLLER)

ED5A D0 E3	BNE	COUNT	DONE?
ED5C CE 15 03	DEC	TEMP3	YES.
ED5F 30 12	DMI	QOREAD	DONE WITH BOTH MODES?
ED61 AD 08 D4	LDA	VCOUNT	
ED64 A4 14	LDY	RTCLK42	READ TIMER LO & HI BYTES
ED66 20 A7 EC	JBR	COMPUT	NO, COMPUTE BAUD RATE
ED69 BC EE 02	BTY	CBAUDL	
ED6C BD EF 02	STA	CBAUDH	SET BAUD RATE INTO RAM CELLS
ED6F A0 09	LDY	#9	SET BIT COUNTER FOR 9 BITS
ED71 D0 CC	BNE	COUNT	

ED73 AD EE 02	QOREAD	LDA	CBAUDL	
ED76 BD 04 D2		STA	AUDF3	
ED79 AD EF 02		LDA	CBAUDH	
ED7C BD 06 D2		STA	AUDF4	SET POKEY FREQ REGS FOR BAUD RATE
ED7F A9 00		LDA	#0	

ED81 BD 0F D2		STA	BKSTAT	
ED84 AD 32 02		LDA	BKCTL	
ED87 BD 0F D2		STA	BKSTAT	INIT. POKEY SERIAL PORT
ED8A A9 55		LDA	#55	
ED8C 91 32		STA	(BUFRLO),Y	STORE '55' AS FIRST RCV. BUFFER
ED8E CB		INY		

ED8F 91 32		STA	(BUFRLO),Y	
ED91 A9 AA		LDA	#AA	
ED93 B5 31		STA	CHKSUM	STORE CHECKSUM FOR 2 BYTES OF 'AA'
ED95 18		CLC		

ED96 A5 32		LDA	BUFRLO	
ED98 69 02		ADC	#2	
ED9A B5 32		STA	BUFRLO	
ED9C A5 33		LDA	BUFRHI	
ED9E 69 00		ADC	#0	
EDA0 B5 33		STA	BUFRHI	INCR. BUFFER POINTER BY 1
EDA2 58		CLI		
EDA3 60		RTS		

EDA4 20 63 EC	BROKE	JBR	BENDDS	BREAK KEY WAS PRESSED, SO PREPARE
EDA7 A9 3C		LDA	#MOTRST	TO RETURN
EDA9 BD 02 D3		STA	PACTL	TURN OFF MOTOR
EDAC BD 03 D3		STA	PBCTL	RAISE NOT COMMAND LINE

EDAF A9 80		LDA	#BRKABT	
EDB1 B5 30		STA	STATUS	STORE BREAK ABORT STATUS CODE

EDB3 AE 1B 03		LDX	STACKP	
EDB6 9A		TXS		RESTORE STACK POINTER

EDB7 C6 11		DEC	BRKKEY	SET BREAK KEY FLAG TO NONZERO
EDB9 58		CLI		ALLOW IRQ'S

EDBA 4C 0D EA		JMP	RETURN	GO RETURN
---------------	--	-----	--------	-----------

EDBD A9 F0	SETVBX	LDA	#JTADRL	STORE TIME OUT ROUTINE ADDRESS
EDBF BD 26 02		STA	CDTHA1	

SIO (SERIAL BUS INPUT/OUTPUT CONTROLLER)

EDC2 A9 EB
EDC4 BD 27 02

LDA #JTADRH
STA CDTMA1+1

EDC7 A9 01
EDC9 BD 17 03

LDA #1 SET FOR TIMER 1
STA TIMFLO SET FLAG TO NOT TIMED OUT

EDCC 78
EDCD 20 5C E4
EDD0 58
EDD1 60

SEI THE SETVBL ROUTINE NEEDS THIS TO CUT SHORT
JSR SETVBL ANY VBLANKS THAT OCCUR
CLI
RTS

'VCOUNT' INTERVAL TIMER MEASUREMENT -- TO -- POKEY FREQ REG VALUE CONVERSION TABLE

THE VALUES STORED IN THE TABLE ARE 'AUDF+7'.

THE FOLLOWING FORMULAS WERE USED TO DETERMINE THE TABLE VALUES:

$F_{OUT} = F_{IN} / (2 * (AUDF + M))$, WHERE $F_{IN} = 1.78979 \text{ MHz}$, & $M = 7$

FROM THIS HAS DERIVED THE FORMULA USED TO COMPUTE THE
TABLE VALUES BASED ON A MEASUREMENT OF THE PERIOD BY
AN INTERVAL OF THE 'VCOUNT' TIMER.

$AUDF + 7 = (11.365167) * T_{OUT}$, WHERE T_{OUT} = # OF COUNTS (127 USEC.
RESOLUTION) OF 'VCOUNT' FOR 1
CHARACTER TIME (10 BIT TIMES).

		AUDF+7	BAUD RATE	VCOUNT INTERVAL
	WORD	\$27C	1407	56
	WORD	\$2D7	1231	64
	WORD	\$332	1094	72
	WORD	\$38D	985	80
	WORD	\$3E8	895	88
EDD2 EB 03	POK TAB	\$443	820	96
EDD4 43 04	WORD	\$49E	757	104
EDD6 9E 04	WORD	\$4F9	703	112
EDD8 F9 04	WORD	\$554	656	120
EDDA 54 05	WORD	\$5AF	615	128
EDDC AF 05	WORD	\$60A	579	136
EDDE 0A 06	WORD	\$665	547	144
EDE0 65 06	WORD	\$6C0	518	152
EDE2 C0 06	WORD	\$71A	492	160
EDE4 1A 07	WORD	\$775	469	168
EDE6 75 07	WORD	\$7D0	447	176
EDE8 D0 07	WORD	\$82B	428	184
	WORD	\$886	410	192
	WORD	\$8E1	394	200
	WORD	\$93C	379	208

910 (SERIAL BUS INPUT/OUTPUT CONTROLLER)

WORD	\$997	1365	216
WORD	\$9F2	1352	224
WORD	\$A4D	1339	232
WORD	\$AAB	1328	240
WORD	\$B03	1318	248

EDEA

CRNTPC

#814

0014 00

BIOSPR . BYTE DSKORG-CRNTPC 910L IS TOO LONG
.END

SIO (SERIAL BUS INPUT/OUTPUT CONTROLLER)

SYMBOL TABLE

ACK	0041	ACKREC	E7C6	ADDCOR	030E	ADJ1	ED10
ADJUST	ED00	ADDRESS	0064	AFP	DB00	ALLPOT	D208
ANTIC	D400	APPEND	0001	APPMHI	000E	ATACIR	02FB
ATAN	BE43	ATTRACT	004D	AUDC1	D201	AUDC2	D203
AUDC3	D205	AUDC4	D207	AUDCTL	D208	AUDF1	D200
AUDF2	D202	AUDF3	D204	AUDF4	D206	B192HI	0000
B192LO	D028	B600HI	0005	B600LO	00CC	BAD	EA63
BADCOM	E9BF	BADIOC	00B6	BADMOD	0091	BEGIN	ED14
BFENHI	0035	BFENLO	0034	BITMSK	004E	BLIM	028A
BLKBDV	E471	BOOT?	0009	BOOTAD	0242	BOTSCR	02BF
BPTR	003D	BRKABT	0080	BRKKEY	0011	BROKE	EDA4
BUFADR	0015	BUFCNT	006B	BUFRFL	0038	BUFRHI	0033
BUFRLO	0032	BUFSTR	006C	CAS31	EC62	CASBUF	03FD
CASENT	E884	CASETV	E440	CASFL0	030F	CASINI	0002
CASORG	EF41	CASRED	E887	CASBBT	004B	CASSET	0060
CAUX1	023C	CAUX2	023D	CBAUDH	02EF	CBAUDL	02EE
CBUFHI	0002	CBUFL0	003A	CCDMND	0238	CDEVIC	023A
CDTMA1	0226	CDTMA2	0228	CDTMF3	022A	CDTMF4	022C
CDTMF5	022E	CDTMV1	0218	CDTMV2	021A	CDTMV3	021C
CDTMV4	021E	CDTMV5	0220	CH	02FC	CHI	02F2
CHACT	02F3	CHACTL	D401	CHAR	02FA	CHBAS	02F4
CHBASE	D409	CHKDON	EABC	CHKERR	008F	CHSNT	003B
CHKSUM	0031	CHKTIM	EAF8	CHRDOR	E000	CIOCHR	002F
CIOINV	E46E	CIOORO	E4A6	CIOV	E436	CIX	00F2
CKEY	004A	CLOSE	000C	COLAC	0072	COLBK	D01A
COLCRB	0055	COLDST	0244	COLDVS	E477	COLINC	007A
COLOR0	02C4	COLOR1	02C5	COLOR2	02C6	COLOR3	02C7
COLOR4	02C8	COLPFO	D016	COLPF1	D017	COLPF2	D018
COLPF3	D019	COLPM0	D012	COLPM1	D013	COLPM2	D014
COLPM3	D015	COLRSH	004F	CONFRM	E978	CONMND	E974
COMPLT	0043	COMPUT	ECA7	CONBOL	D01F	CONTIN	EC35
COS	BD73	COUNT	ED3F	COUNTR	007E	CR	0098
CRETRI	000D	CRETRN	EBC3	CRETRY	0036	CRITIC	0042
CRNTPC	EDEA	CRBINH	02F0	CRBRDR	008D	CSDPIV	E47D
CBTAT	0288	CTIA	D000	CTIMHI	0000	CTIFLO	0002
DAUX1	030A	DAUX2	0308	DBSECT	0241	DBUFHI	0305
DBUFL0	0304	DBYTHI	0309	DBYTLO	0308	DCB	0300
DCDMND	0302	DDEVIC	0300	DEGFL0	00FB	DECON	0006
DELAY0	EC90	DELAY1	EC92	DELETE	0021	DELTAC	0077
DELTAR	0076	DERR	E9F6	DERR1	EA06	DERROR	0090
DFLAGS	0240	DIGRT	00P1	DIRDEX	0057	DIRECT	0002
DISK	0044	DISKIV	E450	DISPLY	0053	DL18TH	D403
DL18TL	D402	DMACTL	D400	DMABK	02A0	DNACK	008B
DOINTP	ECE9	DOBINI	000C	DOSVEC	000A	DOUBLE	0044
DRAWLN	0011	DRETRI	0001	DRETRY	0037	DRKMSK	004E
DSKFMS	0018	DSKINV	E453	DSKORO	EDEA	DSKTIM	0246
DSKUTL	001A	DSPFLO	02FE	DSTAT	004C	DSTATB	0303
DTIMLO	0306	DUNIT	0301	DUNUSE	0307	DVSTAT	02EA
EDITRV	E400	EEXP	00ED	ENDPT	0074	ENINTP	ECEF
EOFERR	008B	ERRFLO	023F	ERROR	0045	ESCFL0	02A2
ESIGN	00EF	EXP	DDC0	EXP10	DDCC	FADD	DA66
FASC	D8E6	FCHRF1	00F0	FDIV	DB28	FEOF	003F
FILDAT	02FD	FILFLO	02B7	FILLIN	0012	FINDX	ECDA
FLDOP	DD8D	FLDOR	DD89	FLD1P	DD9C	FLD1R	DD98
FLOPPY	0030	FLPTR	00FC	FMOVE	DD86	FMSZP0	0043
FMUL	DADB	FNCNOT	0092	FOOEY	EAE0	FORMAT	0021
FPI	D9D2	FPREC	0006	FPSCR	05E6	FPSCR1	05EC
FPTR2	00FE	FRO	00D4	FR1	00E0	FR2	00E6
FRE	00DA	FREQ	0040	FRMERR	00BC	FRX	00EC

SID (SERIAL BUS INPUT/OUTPUT CONTROLLER)

FSCR	05E6	FSCR1	05EC	FSTOP	DDA8	FSTOR	DDA7
FSUB	DA60	FTYPE	003E	GETCHR	0007	GETREC	0005
GLBABS	02E0	GOBACK	EB08	GOOD	EA65	GOODN	EB68
GOREAD	ED73	GPRIOR	026F	GRACITL	D01D	GRAFM	D011
GRAFF0	D00D	GRAFF1	D00E	GRAFF2	D00F	GRAFF3	D010
HATABS	031A	HITCLR	D01E	HITIMR	EC00	HITDNE	0005
HOLD1	0051	HOLD2	029F	HOLD3	029D	HOLD4	028C
HOLD5	028D	HOLDCH	007C	HPOSM0	D004	HPOSM1	D005
HPOSM2	D006	HPOSM3	D007	HPOSP0	D000	HPOSP1	D001
HPOSP2	D002	HPOSP3	D003	HSCROL	D404	ICAX1	034A
ICAX1Z	002A	ICAX2	034B	ICAX2Z	002B	ICBAH	0345
ICBAHZ	0025	ICBAL	0344	ICBALZ	0024	ICBLH	0349
ICBLHZ	0029	ICBLI	034B	ICBLIZ	002B	ICCOM	0342
ICCOMT	0017	ICCOMZ	0022	ICDNO	0341	ICDNOZ	0021
ICHID	0340	ICHIDZ	0020	ICIDNO	002E	ICPTH	0347
ICPTHZ	0027	ICPTL	0346	ICPTLZ	0026	ICSPR	034C
ICSPRZ	002C	ICSTA	0343	ICSTAZ	0023	IFF	D9AA
INBUFF	00F3	INBCLR	0020	INSDAT	007D	INTAB8	0200
INTEMP	022D	INTINV	E46B	INTORG	E6D5	INTTBL	EC88
INTZBS	001D	INVLQ	0286	IOCB	0340	IOCBAB	0020
IOCB5Z	0010	IOCFRE	00FF	IRGEN	D20E	IRGST	D20E
ISRODN	EA90	ISRSIR	EB11	ISRTD	EAD1	JTADRH	00EB
JTADRL	00F0	JTINER	EBF0	KBCODE	D209	KBD	0048
KBDORG	F3E4	KEYBDV	E420	KEYDEL	02F1	LBFEND	05FF
LBPR1	057E	LBPR2	057F	LBUFF	0580	LDPNTR	EB6E
LEDGE	0002	LINBUF	0247	LINZBB	0000	LMARON	0052
LOCKFL	0023	LOG	DECD	LOG10	DED1	LOGCOL	0053
LOGMAP	02B2	LOTONE	0007	LPENH	0234	LPEV	0235
MOPF	D000	MOPL	D008	M1PF	D001	M1PL	D009
M2PF	D002	M2PL	D00A	M3PF	D003	M3PL	D00B
MAXDEV	0021	MAXIOC	00B0	MEMLO	02E7	MEMTOP	02E5
MLTTMP	0066	MODATA	E9F0	MODEM	004D	MONORG	F0E3
MOTRGD	0034	MOTRBT	003C	HXBHOD	0010	NACK	004E
NCOMH1	003C	NCOMLO	0034	NEWCOL	0061	NEWROW	0060
NMIEN	D40E	NMIREB	D40F	NMIST	D40F	NOCK8M	003C
NOCLR	EAEB	NOISEI	EC49	NONDEV	00B2	NORMAL	004E
NOTCAS	EC10	NOTCBT	E96B	NOTDER	EA52	NOTDON	EA81
NOTE	0026	NOTEND	EAC0	NOTERR	EA00	NOTOPN	00B5
NOTYET	EB3E	NOWRPO	EA98	NSIGN	00EE	NTBRKO	EA8B
NTBRK1	EB02	NTBRK2	ED18	NTFRAM	EB1F	NTDVRN	EB27
NTWRP1	EB52	NVALID	00B4	NWOK	EA56	ODNHI	00EA
ODNLO	0090	OKTIMR	ED23	OLDROW	005A	OLDADR	005E
OLDCHR	005D	OLDCOL	0058	OLDROW	005A	OPEN	0003
OPNIN	0004	OPNINO	000C	OPNOT	0008	OPNTMP	0066
QVRRUN	008E	POPF	D004	POPL	D00C	P1PF	D005
P1PL	D00D	P2PF	D006	P2PL	D00E	P3PF	D007
P3PL	D00F	PACTL	D302	PADDLO	0270	PADDL1	0271
PADDL2	0272	PADDL3	0273	PADDL4	0274	PADDL5	0275
PADDL6	0276	PADDL7	0277	PBCTL	D303	PBPNT	001D
PBUF9Z	001E	PCOLR0	02C0	PCOLR1	02C1	PCOLR2	02C2
PCOLR3	02C3	PENH	D40C	PENV	D40D	PIA	D300
PLOT	0050	PLUS	ECFC	PLYAR0	05E0	PLYEVL	DD40
PMBASE	D407	POINT	0025	POKEY	D200	POKMSK	0010
POKTAB	EDD2	PORTA	D300	PORTB	D301	POTO	D200
POT1	D201	POT2	D202	POT3	D203	POT4	D204
POT5	D205	POT6	D206	POT7	D207	POT8	D208
PRINTR	0040	PRINTV	E430	PRIOR	001B	PRNBUF	03C0
PRNORG	EE78	PRVOPN	00B1	PTMP	001F	PTIMOT	001C
PTRIO0	027C	PTRIO1	027D	PTRIO2	027E	PTRIO3	027F
PTRIO4	0280	PTRIO5	0281	PTRIO6	0282	PTRIO7	0283

SIO (SERIAL BUS INPUT/OUTPUT CONTROLLER)

PUTCHR	000B	PUTREC	0009	RADFLG	00FB	RADON	0000
RAMLO	0004	RAMSIZ	02E4	RAMTOP	006A	RANDOM	D20A
RBLOKV	E47A	RDOPLY	00B7	READ	0052	RECEIV	EAE2
RECVDN	0039	RECVDS	EC44	RECVEN	EC1F	REDGE	0027
RELONE	EA83	RENAME	0020	RETURN	EA0D	RIRQHI	0000
RIRGLO	007B	RMARGN	0053	ROWAC	0070	ROWCRS	0054
ROWINC	0079	RRETRN	EB10	RSIRO	000A	RTCLOK	0012
SAVADR	006B	SAVIO	0314	SAVMSC	005B	SCREDT	0045
SCRENV	E410	SCRFLG	02BB	SCRMEM	0093	SDL8TH	0231
SDLSTL	0230	SDMCTL	022F	SEND	EA6B	SENDD8	EC43
SENDEN	EBF4	SENDEV	E448	SENDIN	EC8E	SERIN	D20D
SEROUT	D20D	SETVBV	E45C	SETVBX	ED8D	SHFANT	006F
SHFLOK	02BE	SIDWAY	0053	SIN	BDB1	SIO	E959
SIOINT	E944	SIOINV	E465	SIOORQ	E944	SIO8PR	0014
SIOV	E459	SIRHI	00EB	SIRLO	0011	SIZEN	D00C
SIZEP0	D00B	SIZEP1	D009	SIZEP2	D00A	SIZEP3	D00B
SKCTL	D20F	SKRES	D20A	SKSTAT	D20F	SOUNDR	0041
SPECIL	000E	SQR	BEB1	SRETRN	EB36	SRTIMR	022B
SRTIRO	EB9F	SRTIR1	EB05	SRTIR2	EBED	SSFLAQ	02FF
SSKCTL	0232	STACKP	031B	STATIS	0053	STATUS	0030
STICK0	027B	STICK1	0279	STICK2	027A	STICK3	027B
STIMER	D209	STRIG0	02B4	STRIG1	02B5	STRIG2	02B4
STRIG3	02B7	STIMOT	EC79	SUBTMP	029E	SUCCEB	0001
SUSUAL	EB3A	SWPFLG	007B	SYSVBV	E45F	TABMAP	02A3
TDHI	00EA	TDLO	00D1	TEMP	023E	TEMP1	0312
TEMP2	0314	TEMP3	0315	TEMPHI	0002	TEMPLO	003E
TIMER1	030C	TIMER2	0310	TIMFLO	0317	TIMIT	EBA9
TIMIT1	EBCF	TIMOUT	00BA	TINDEX	0293	TMPCHR	0050
TMPCOL	02B9	TMPLBT	02A1	TMPROW	02BB	TMPIX1	029C
TOUT	EB0C	TOUT1	ED4B	TRANSZ	0005	TRIG0	D010
TRIG1	D011	TRIG2	D012	TRIG3	D013	TRNRCD	00B9
TSTAT	0319	TSTDAT	0007	TXTCOL	0291	TXTMSC	0294
TXTOLD	0296	TXTR0W	0290	UNLOCK	0024	USAREA	04B0
VBREAK	0206	VCOUNT	D40B	VCTABL	E4B0	VDELAY	D01C
VDBLST	0200	VECTBL	E400	VIMIRO	0216	VINTER	0204
VKEYBD	020B	VPRCED	0202	VSCROL	D405	VSERIN	020A
VBEROC	020E	VBERDR	020C	VTIMR1	0210	VTIMR2	0212
VTIMR4	0214	VVBLKD	0224	VVBLK1	0222	WAIT	EA1A
WAITER	EC9F	WARMST	000B	WARM5V	E474	WATCOM	E907
WIRCHI	0000	WIRGLO	00B4	WMODE	02B9	WOK	EA3D
WRITE	0057	WROPLY	00B3	WSIRO	000F	WSYNC	D40A
XITVBV	E462	XMTDON	003A	ZERIT	EC71	ZTCB	0020
ZTEMP1	00F5	ZTEMP3	00F9	ZTEMP4	00F7		

DISK ***** DISKP.SRC ***** 3/9/79 ***** 4:00:00 P.M.

TITLE "DISK ***** DISKP.SRC ***** 3/9/79 ***** 4:00:00 P.M."

0002 STATVH = DVSTAT/256
00EA STATVL = -256*STATVH+DVSTAT STATUS POINTER

CONSTANT EQUATES

0031	DISKID	=	\$31	SERIAL BUS DISK I.D.
0050	PUTSEC	=	\$50	DISK PUT SECTOR DCB COMMAND
0052	READ	=	\$52	DISK GET SECTOR DCB COMMAND
0057	WRITE	=	\$57	DISK PUT SECTOR WITH READ CHECK DCB COMMAND
0053	STATC	=	\$53	DISK STATUS DCB COMMAND
0021	FORMAT	=	\$21	DISK FORMAT DCB COMMAND
0000	NODAT	=	0	BIO COMMAND FOR "NO DATA" OPERATION
0040	GETDAT	=	\$40	BIO COMMAND FOR "DATA FROM DEVICE"
0080	PUTDAT	=	\$80	BIO COMMAND FOR "DATA TO DEVICE"

VECTORS

* = \$E450

E450 4C EA ED	JMP	DINIT	DISK INIT. VECTOR
E453 4C F0 ED	JMP	DBKIF	DISK INTERFACE ENTRY POINT

CONSTANTS

* = DBKORU

DISK INTERFACE ROUTINE STARTS HERE

DISK INTERFACE INITIALIZATION ROUTINE

EDEA A9 A0	DINIT	LDA	#160	
EDEC BD 46 02		STA	DSKTIM	SET INITIAL DISK TIMEOUT TO 160 SEC
EDEF 60		RTS		

DISK INTERFACE ENTRY POINT

EDF0 A9 31	DSKIF	LDA	#DISKID	
EDF2 BD 00 03		STA	DDEVIC	SET SERIAL BUS I.D IN DCB
EDF3 AD 46 02		LDA	DSKTIM	
EDFB AE 02 03		LDX	DCOMND	
EDFB E0 21		CPX	#FORMAT	IS COMMAND A FORMAT COMMAND?
EDFD F0 02		BEG	PUTDTC	
EDFF A9 07		LDA	#7	NO, SET TIMEOUT TO 7 SECS.
EE01 BD 06 03	PUTDTC	STA	DTIMLO	PUT DISK TIMEOUT IN DCB
EE04 A2 40		LDX	#OETDAT	SET "OET DATA" COMMAND FOR BIO
EE06 A0 80		LDY	#80	SET BYTE COUNT TO 128
EE08 AD 02 03		LDA	DCOMND	READ COMMAND IN DCB
EE08 C9 57		CMF	#WRITE	IS COMMAND A "PUT SECTOR" COMMAND?
EE0D D0 02		BNE	CKBTC	
EE0F A2 80		LDX	#PUTDAT	YES, SET "PUT DATA" COMMAND FOR BIO
EE11 C9 53	CKBTC	CMF	#BTATC	IS COMMAND A STATUS COMMAND?
EE13 D0 0C		BNE	PUTCNT	
EE15 A9 EA		LDA	#STATVL	
EE17 BD 04 03		STA	DBUFLO	
EE1A A9 02		LDA	#STATVH	
EE1C BD 05 03		STA	DBUFHI	SET BUFFER ADDR TO GLOBAL STATUS BUFFER
EE1F A0 04		LDY	#4	YES, SET BYTE COUNT TO 4
EE21 BE 03 03	PUTCNT	STX	DBSTATS	PUT STATUS COMMAND FOR BIO IN DCB
EE24 BC 08 03		STY	DBYTLO	
EE27 A9 00		LDA	#0	
EE29 BD 09 03		STA	DBYTHI	PUT BYTE COUNT IN DCB
EE2C 20 59 E4		JBR	BIOV	CALL SERIAL I/O.
EE2F 10 01		BPL	GOODST	NO ERROR
EE31 60		RTS		NO, GO BACK
EE32 AD 02 03	GOODST	LDA	DCOMND	READ THE COMMAND
EE35 C9 53		CMF	#BTATC	WAS IT A STATUS COMMAND?
EE37 D0 0A		BNE	PUTBC	
EE39 20 6D EE		JBR	PUTADR	PUT BUFFER ADDR IN TEMP REG.
EE3C A0 02		LDY	#2	
EE3E B1 15		LDA	(BUFADR),Y	READ DISK TIMEOUT VALUE BYTE OF STATUS
EE40 BD 46 02		STA	DSKTIM	PUT IT IN DISK TIMEOUT REG.
EE43 AD 02 03	PUTBC	LDA	DCOMND	
EE46 C9 21		CMF	#FORMAT	WAS COMMAND A FORMAT COMMAND?
EE48 D0 1F		BNE	ENDDIF	
EE4A 20 6D EE	FMTD	JBR	PUTADR	YES, PUT BUFFER ADDR INTO TEMP REG
EE4D A0 FE		LDY	#FE	SET BUFFER POINTER
EE4F CB	TWICE	INY		
EE50 CB		INY		INCR BUFFER POINTER BY 2
EE51 B1 15	RDBAD	LDA	(BUFADR),Y	READ LO BYTE BAD SECTOR DATA
EE53 C9 FF		CMF	#FF	
EE55 D0 FB		BNE	TWICE	IS IT "FF" ?
EE57 CB		INY		YES,
EE5B B1 15		LDA	(BUFADR),Y	READ HI BYTE BAD SECTOR DATA
EE5A CB		INY		
EE5B C9 FF		CMF	#FF	
EE5D D0 F2		BNE	RDBAD	IS IT "FF" ?
EE5F BB		DEY		
EE60 BB		DEY		YES,

DISK ***** DISKP. SRC ***** 3/9/79 ***** 4:00:00 P.M.

EE61 8C 08 03		BTY	DBYTLO	PUT BAD SECTOR BYTE COUNT INTO DCB
EE64 A9 00		LDA	#0	
EE66 8D 09 03		STA	DBYTHI	
EE69 AC 03 03	ENDDIF	LDY	DSTATB	
EE6C 60		RTB		

SUBROUTINES

PUT BUFFER ADDR FROM DCB INTO TEMP REG

EE6D AD 04 03	PUTADR	LDA	DBUFLO	
EE70 B5 15		STA	BUFADR	
EE72 AD 05 03		LDA	DBUFHI	
EE75 B5 16		STA	BUFADR+1	PUT BUFFER ADDR IN TEMP REG
EE77 60		RTB		

SPARE BYTE OR MODULE TOO LONG FLAG

EE7B	CRNTPC	=	*	
		* = \$14		
0014 00	DSKBPR	BYTE	PRNDRG-CRNTPC	DISKP TOO LONG

END

SYMBOL TABLE

ADDCOR	030E	ADREBB	0064	AFP	D800	ALLPOT	D208
ANTIC	D400	APPEND	0001	APPMHI	000E	ATACHR	02FB
ATAN	BE43	ATTRACT	004D	AUDC1	D201	AUDC2	D203
AUDC3	D205	AUDC4	D207	AUDCTL	D208	AUDF1	D200
AUDF2	D202	AUDF3	D204	AUDF4	D206	BADIOC	0086
BADMOD	0091	BFENHI	0035	BFENLO	0034	BITHBK	006E
BLIM	028A	BLKBDV	E471	BOOT?	0009	BOOTAD	0242
BOTSCR	02BF	BPTR	003D	BRKABT	0080	BRKKEY	0011
BUFADR	0015	BUFCNT	006B	BUFRFL	0038	BUFRHI	0033
BUFRLO	0032	BUFBTR	006C	CASBUF	03FD	CASETV	E440
CABFL0	030F	CABINI	0002	CASOR0	EF41	CASBBT	0048
CASSET	0043	CAUX1	023C	CAUX2	023D	CBAUDH	02EF
CBAUDL	02EE	CCOMND	023B	CDEVIC	023A	CDTMA1	0226
CDTMA2	0228	CDTMF3	022A	CDTMF4	022C	CDTMF5	022E
CDTMV1	0218	CDTMV2	021A	CDTMV3	021C	CDTMV4	021E
CDTMV5	0220	CH	02FC	CHI	02F2	CHACT	02F3
CHACTL	D401	CHAR	02FA	CHBAS	02F4	CHBABE	D409
CHKERR	008F	CHKBNT	003B	CHKSUM	0031	CHRDOR	E000
CIOCHR	002F	CIOINV	E46E	CIDOR0	E4A6	CIOV	E456
CIX	00F2	CKEY	004A	CKSTC	EE11	CLOSE	000C
COLAC	0072	COLBK	D01A	COLCR8	0055	COLDST	0244
COLD5V	E477	COLINC	007A	COLOR0	02C4	COLOR1	02C5
COLOR2	02C6	COLOR3	02C7	COLOR4	02C8	COLPFO	D016
COLPF1	D017	COLPF2	D018	COLPF3	D019	COLPM0	D012
COLPM1	D013	COLPM2	D014	COLPM3	D015	COLRSH	004F
CONSOL	D01F	COB	BD73	COUNTR	007E	CR	0098
CRETRY	0036	CRITIC	0042	CRNTPC	EE78	CRBINH	02F0
CRBRDR	008D	CRPIV	E47D	CSTAT	0288	CTIA	D000
DAUX1	030A	DAUX2	0308	DBBECT	0241	DBUFHI	0305
DBUFLO	0304	DBYTHI	0309	DBYTLO	0308	DCB	0300
DCOMND	0302	DDEVIC	0300	DEQFLO	00FB	DEQDN	0006
DELETE	0021	DELTAC	0077	DELTAR	0076	DERROR	0090
DFLA08	0240	DIORT	00F1	DINDEX	0057	DINIT	EDEA
DIRECT	0002	DISK	0044	DISKID	0031	DISKIV	E450
DISPLY	0053	DLISTH	D403	DLISTL	D402	DHACTL	D400
DMASK	02A0	DNACK	008B	DOBINI	000C	DOBVEC	000A
DRAWLN	0011	DRETRY	0037	DRKMBK	004E	DBKFMS	0018
DSKIF	EDF0	DSKINV	E453	DBKOR0	EDEA	DBKSPR	0014
DSKTIM	0246	DSKUTL	001A	DSPFLO	02FE	DSTAT	004C
DSTAT8	0303	DTIMLO	0306	DUNIT	0301	DUNUSE	0307
DVBSTAT	02EA	EDITRV	E400	EEXP	00ED	ENDDIF	EE69
ENDPT	0074	EDFERR	008B	ERRFLO	023F	EBCFLO	02A2
ESIGN	00EF	EXP	DDC0	EXP10	DDCC	FADD	DA66

DISK ***** DISKP. SRC ***** 3/9/79 ***** 4:00:00 P. M.

HSCROL	D404	ICAX1	034A	ICAX1Z	002A	ICAX2	034B
ICAX2Z	002B	ICBAH	0345	ICBAHZ	0025	ICBAL	0344
ICBALZ	0024	ICBLH	0349	ICBLH2	0029	ICBLL	0348
ICBLLZ	0028	ICCOM	0342	ICCOMT	0017	ICCOMZ	0022
ICDNO	0341	ICDNOZ	0021	ICHID	0340	ICHIDZ	0020
ICIDNO	002E	ICPTH	0347	ICPTH2	0027	ICPTL	0346
ICPTLZ	0026	ICBPR	034C	ICBPRZ	002C	ICBTA	0343
ICSTAZ	0023	IFP	D9AA	INBUFF	00F3	INBCLR	0020
INSDAT	007D	INTABS	0200	INTEMP	022D	INTINV	E46B
INTORG	E6D5	INTZBB	0010	INVFLQ	02B6	IOCB	0340
IOCBAS	0020	IOCBGZ	0010	IOCFRE	00FF	IRGEN	D20E
IRGBY	D20E	KBCODE	D209	KBD	004B	KBDORG	F3E4
KEYBDV	E420	KEYDEL	02F1	LBFEND	05FF	LBPR1	057E
LBPR2	057F	LBUFF	0580	LEDGE	0002	LINBUF	0247
LINZBS	0000	LHARGN	0052	LOCKFL	0023	LDD	DECD
LO010	DED1	LOGCOL	0063	LOGMAP	02B2	LPENH	0234
LPENV	0235	MOPF	D000	MOPL	D00B	MIPF	D001
M1PL	D009	M2PF	D002	M2PL	D00A	M3PF	D003
M3PL	D00B	MAXDEV	0021	MAXIOC	00B0	MEMLO	02E7
MEMTOP	02E5	MLTTP	0066	MODEM	0040	MONORG	FOE3
HXDMOD	0010	NEWCOL	0061	NEWROW	0060	NIEN	D40E
NMIREB	D40F	NMIST	D40F	NOCKBM	003C	NODAT	0000
NONDEV	00B2	NOTE	0026	NOTOPN	00B5	NSION	00EE
NVALID	00B4	OLDADR	005E	OLDCHR	005D	OLDCOL	005B
OLDROW	005A	OPEN	0003	OPNIN	0004	OPNINO	000C
OPNOT	000B	OPNTMP	0066	OVRUN	00BE	POPF	D004
POPL	D00C	P1PF	D005	PIPL	D00D	P2PF	D006
P2PL	D00E	P3PF	D007	P3PL	D00F	PACTL	D302
PADDLO	0270	PADDL1	0271	PADDL2	0272	PADDL3	0273
PADDL4	0274	PADDL5	0275	PADDL6	0276	PADDL7	0277
PBCTL	D303	PBPNT	001D	PBUFZ	001E	PCOLRO	02C0
PCOLR1	02C1	PCOLR2	02C2	PCOLR3	02C3	PENH	D40C
PENV	D40D	PIA	D300	PLYARG	05E0	PLYEVL	DD40
PMBASE	D407	POINT	0025	POKEY	D200	POKMSK	0010
PORTA	D300	PORTB	D301	POTO	D200	POT1	D201
POT2	D202	POT3	D203	POT4	D204	POT5	D205
POT6	D206	POT7	D207	POT00	D20B	PRINTR	0050
PRINTV	E430	PRIOR	D01B	PRNBUF	03C0	PRNORG	EE7B
PRVOPN	00B1	PTEMP	001F	PTIHOT	001C	PTRIO0	027C
PTRIO1	027D	PTRIO2	027E	PTRIO3	027F	PTRIO4	0280
PTRIO5	0281	PTRIO6	0282	PTRIO7	0283	PUTADR	EE6D
PUTBC	EE43	PUTCHR	000B	PUTCNT	EE21	PUTDAT	00B0
PUTDT0	EE01	PUTREC	0009	PUTSEC	0050	RADFLQ	00FB
RADON	0000	RAMLQ	0004	RAMBIZ	02E4	RAMTOP	006A
RANDON	D20A	RBLQKV	E47A	RDBAD	EE51	RONLY	00B7
READ	0052	RECVDM	0039	REDGE	0027	RENAME	0020
RHARGN	0053	ROWAC	0070	ROWCRS	0054	ROWINC	0079
RTCLOCK	0012	SAVADR	006B	SAVID	0316	SAVMSK	005B
SCREDIT	0045	SCREENV	E410	SCRFLQ	02B8	SCRMEM	0093
SDLBTH	0231	SDLBTL	0230	SDMCTL	022F	SENDEV	E46B
SERIN	D20D	SEROUT	D20D	SETVBV	E45C	SHFANT	006F
SHFLOK	02BE	SIN	BDB1	SIOINV	E465	SIDORG	E944
SIDV	E459	SIZEM	D00C	SIZEPO	D00B	SIZEP1	D009
SIZEP2	D00A	SIZEP3	D00B	SKCTL	D20F	SKRES	D20A
SKSTAT	D20F	SOUNDR	0041	SPECIL	000E	SGR	BEB1
SRTIMR	022B	SSFLAQ	02FF	SBKCTL	0232	STACKP	031B
STATC	0053	STATIS	000D	STATUS	0030	STATVH	0002
STATVL	00EA	STICK0	027B	STICK1	0279	STICK2	027A
STICK3	027B	STIMER	D209	STRIO0	02B4	STRIO1	02B5
STRIO2	02B4	STRIO3	02B7	SUBTMP	029E	SUCCEB	0001

DISK ***** DISKP.SRC ***** 3/9/79 ***** 4:00:00 P.M.

SWPFLO	007B	BYBVBV	E45F	TABMAP	02A3	TEMP	023E
TEMP1	0312	TEMP2	0314	TEMP3	0315	TIMER1	030C
TIMER2	0310	TIMFLO	0317	TIMOUT	008A	TINDEX	0293
TMPCHR	0050	TMPCOL	02B9	TMPLBT	02A1	TMPROW	02B8
TMPX1	029C	TRANSZ	0006	TRI00	D010	TRI01	D011
TRI02	D012	TRI03	D013	TRNRCD	0089	TBTAT	0319
TSTDAT	0007	TWICE	EE4F	TXTCOL	0291	TXTHSC	0294
TXTOLD	0296	TXTROW	0290	UNLOCK	0024	USAREA	0480
VBREAK	0206	VCDUNT	D40B	VCTABL	E480	VDELAY	D01C
VDSLST	0200	VECTBL	E400	VIMIRQ	0216	VINTER	0204
VKEYBD	0208	VPRCED	0202	VSCROL	D405	VSERIN	020A
VSEROC	020E	VSEROR	020C	VTIMR1	0210	VTIMR2	0212
VTIMR4	0214	VVBLKD	0224	VVBLKI	0222	WARMST	0008
WARMSV	E474	WMODE	0289	WRITE	0057	WRONLY	0083
WSYNC	D40A	XITVBV	E462	XMTDON	003A	ZIOCB	0020
ZTEMP1	00F5	ZTEMP3	00F9	ZTEMP4	00F7		

TITLE "PRINTER ***** PRINTP.SRC ***** 3/9/79 ***** 4:00:00 P.M."

DEVICE NUMBER OR CODE EQUATES

0002	OPNOUT	=	\$2	IOCB OPEN FOR OUTPUT COMMAND
002B	NBUF8Z	=	40	PRINT NORMAL BUFFER SIZE
0014	DBUF8Z	=	20	PRINT DOUBLE BUFFFER SIZE
001D	SBUF8Z	=	29	PRINT SIDEWAYS BUFFER SIZE
0040	PDEVN	=	\$40	PRINTER DEVICE NUMBER
0053	STATC	=	\$53	DCB STATUS COMMAND CODE
0057	WRITEC	=	\$57	DCB WRITE COMMAND
009B	CR	=	\$9B	ASCII CARRIAGE RETURN
0020	SPACE	=	\$20	ASCII SPACE CHAR.
004E	N	=	\$4E	ASCII "N" CHAR.
0044	D	=	\$44	ASCII "D" CHAR.
0053	B	=	\$53	ASCII "B" CHAR.

PRINTER HANDLER ENTRY POINTS

* = \$E430

E430 9E EE	WORD	PHOPEN-1	PRINTER HANDLER OPEN
E432 DB EE	WORD	PHCLOS-1	PH CLOSE
E434 9D EE	WORD	BADBT-1 PH READ	
E436 A6 EE	WORD	PHWRIT-1	PH WRITE
E438 80 EE	WORD	PHSTAT-1	PH STATUS
E43A 9D EE	WORD	BADBT-1 PH SPECIAL	
E43C 4C 7B EE	JMP	PHINIT	PH INIT.
E43F 00	BYTE	0	ROM FILLER

* = PRNDRO

PRINTER HANDLER CONSTANTS

EE7D EA 02	PHBTLO	WORD	DVSTAT	STATUS BUFFER POINTER
EE7F C0 03	PHCHLO	WORD	PRNBUF	CHAR. BUFFER POINTER

PRINTER HANDLER ROUTINES

PRINTER HANDLER STATUS ROUTINE

EE81 A9 04	PHSTAT	LDA	#4	
EE83 85 1E		STA	PBUF8Z	SET BUFFER SIZE TO 4 BYTES
EE85 AE 7D EE		LDX	PHBTLO	
EE88 AC 7E EE		LDY	PHBTLO+1	SET POINTER TO STATUS BUFFER
EE8B A9 53		LDA	#STATC	SET COMMAND TO "STATUS"
EE8D 8D 02 03		STA	DCOMND	SET STATUS COMMAND
EE90 8D 0A 03		STA	DAUX1	
EE93 20 E6 EE		JBR	SETDCB	GO SETUP DCB
EE96 20 59 E4		JBR	BIOV	SEND STATUS COMMAND
EE99 30 03		BMI	BADST	GO IF ERROR
EE9B 20 14 EF		JBR	PHPUT	YES, PUT STATUS INTO GLOBAL BUFFER.
EE9E 60	BADST	RTB		

PRINTER HANDLER OPEN ROUTINE

EE9F 20 81 EE	PHOPEN	JBR	PHSTAT	DO STATUS COMMAND TO BIO
EEA2 A9 00		LDA	#0	
EEA4 85 1D		STA	PBPNT	CLEAR PRINT BUFFER POINTER
EEA6 60		RTB		

PRINTER HANDLER WRITE ROUTINE

EEA7 85 1F	PHWRIT	STA	PTEMP	SAVE ACCUM
EEA9 20 1A EF		JBR	PRMODE	GO DETERMINE PRINT MODE
EEAC A6 1D		LDX	PBPNT	
EEAE A5 1F		LDA	PTEMP	GET CHAR. SENT BY CIO
EEB0 9D C0 03		STA	PRNBUF, X	PUT CHAR. IN PRINT BUFFER
EEB3 E8		INX		INCR. BUFFER POINTER
EEB4 E4 1E		CPX	PBUF8Z	BUFFER POINTER-BUFFER SIZE?
EEB6 F0 13		BEG	BUFFUL	
EEB8 86 1D		STX	PBPNT	SAVE BUFFER POINTER
EEBA C9 9B		CMP	#CR	IS CHAR. = EOL ?
EEBC F0 03		BEG	BLFILL	IF YES, GO DO BLANK FILL.
EEBE A0 01		LDY	#SUCCES	PUT GOOD STATUS IN Y REG FOR CIO.
EEC0 60		RTS		
EEC1 A9 20	BLFILL	LDA	#SPACE	PUT BLANK IN ACCUM.

EEC3 9D C0 03	FILLBF	BTA	PRNBUF,X	STORE IT IN PRINT BUFFER.
EEC6 EB		INX		
EEC7 E4 1E		CPX	PBUF82	
EEC9 D0 F8		BNE	FILLBF	BUFFER BLANK FILLED?
EECB A9 00	BUFFUL	LDA	#0	
EECD B5 1D		BTA	PBPNT	CLEAR PRINT BUFFER POINTER
EECF AE 7F EE		LDX	PHCHLO	
EED2 AC 80 EE		LDY	PHCHLO+1	SET POINTER TO PRINT BUFFER
EED5 20 E6 EE		JBR	SETDCB	GO SETUP DCB
EED8 20 59 E4		JBR	SIOV	SEND PRINT COMMAND
EEDB 60		RTB		YES.

PRINTER HANDLER CLOSE ROUTINE

EEDC 20 1A EF	PHCLOS	JBR	PRMODE	GO DETERMINE PRINT MODE
EEDF A6 1D		LDX	PBPNT	
EEE1 D0 DE		BNE	BLFILL	
EEE3 A0 01		LDY	#SUCCES	
EEE5 60		RTB		

SUBROUTINES

SET UP DCB TO CALL SIO

03	SETDCB	STX	DBUFLO	
03		BTY	DBUFHI	SET BUFFER POINTER
40		LDA	#PDEVN	
00 03		BTA	DDEVICE	SET PRINTER BUS I.D. FOR DCB
A9 01		LDA	#1	
BD 01 03		BTA	DUNIT	SET UNIT NUMBER TO 1
A9 80		LDA	#80	DEVICE WILL EXPECT DATA
AE 02 03		LDX	DCOMND	
E0 53		CPX	#STATC	STATUS COMMAND?
D0 02		BNE	PBIOC	
A9 40		LDA	#40	EXPECT DATA FROM DEVICE
BD 03 03	PBIOC	BTA	DBSTATS	SET SIO MODE COMMAND.
A5 1E		LDA	PBUF82	
BD 08 03		BTA	DBYTLO	SET LO BYTE COUNT
A9 00		LDA	#0	
BD 09 03		BTA	DBYTHI	SET HI BYTE COUNT
A5 1C		LDA	PTIMOT	
BD 06 03		BTA	DTIMLO	SET DEVICE TIMEOUT COUNT
EF13 60		RTB		

GET DEVICE TIMEOUT FROM STATUS & SAVE IT

EF14 AD EC 02	PHPUT	LDA	DVSTAT+2	
EF17 85 1C		STA	PTIMOT	SAVE DEVICE TIMEOUT
EF19 60		RTS		

DETERMINE PRINT MODE & SETUP PRINT BUFFER SIZE, DCB PRINT
COMMAND, & DCB AUX1 FOR PRINT MODE

EF1A A0 97	PRMODE	LDY	#WRITEC	PUT WRITE COMMAND IN Y REG
EF1C A5 28		LDA	ICAX27	READ PRINT MODE
EF1E C9 4E	CMODE	CMF	#N	
EF20 D0 04		BNE	CDUBL	PRINT NORMAL ?
EF22 A2 28		LDX	#NBUFFZ	YES, SET NORMAL CHAR. BUFFER SIZE
EF24 D0 0E		BNE	SETBSZ	
EF26 C9 44	CDUBL	CMF	#D	
EF28 D0 04		BNE	CSIDE	PRINT DOUBLE?
EF2A A2 14		LDX	#DBUFFZ	YES, SET DOUBLE CHAR. BUFFER SIZE
EF2C D0 06		BNE	SETBSZ	
EF2E C9 53	CSIDE	CMF	#S	PRINT SIDEWAYS ?
EF30 D0 08		BNE	QOERR	IF NOT, GO TO ERROR ROUTINE
EF32 A2 1D		LDX	#SBUFFZ	YES, SET SIDEWAYS BUFFER SIZE
EF34 86 1E	SETBSZ	STX	PBUFFZ	STORE PRINT BUFFER SIZE
EF36 8C 02 03		STY	DCOMND	STORE DCB COMMAND
EF39 8D 0A 03		STA	DAUX1	STORE DCB AUX1 PRINT MODE
EF3C 60		RTS		
EF3D A9 4E	QOERR	LDA	#N	SET DEFAULT PRINT MODE TO NORMAL
EF3F D0 DD		BNE	CMODE	

SPARE BYTE OR MODULE TOO LONG FLAG

EF41	CRNTPC	=	*
		* = \$14	

0014 00	PRNBPR	BYTE	CASORG-CRNTPC	PRINTP TOO LONG
				END

SYMBOL TABLE							
ADDCOR	030E	ADREBB	0064	AFP	D800	ALLPOT	D208
ANTIC	D400	APPEND	0001	APPMHI	000E	ATACHR	02FB
ATAN	BE43	ATTRACT	004D	AUDC1	D201	AUDC2	D203
AUDC3	D205	AUDC4	D207	AUDCTL	D208	AUDF1	D200
AUDF2	D202	AUDF3	D204	AUDF4	D206	BADI0C	00B6
BADMOD	0091	BADBT	EE9E	BFENHI	0035	BFENLO	0034
BITMSK	006E	BLFILL	EEC1	BLIM	02BA	BLKBDV	E471
BOOT?	0009	BOOTAD	0242	BOTSCR	02BF	BPTR	003D
BRKABT	0080	BRKKEY	0011	BUFADR	0015	BUFCNT	0068
BUFFUL	EECB	BUFRFL	0038	BUFRHI	0033	BUFRLO	0032
BUFBTR	006C	CASBUF	03FD	CASEYV	E440	CABFLO	030F
CASINI	0002	CASORQ	EF41	CASBT	004B	CASSET	0043
CAUX1	023C	CAUX2	023D	CBAUDH	02EF	CBAUDL	02EE
CCOMND	023B	CDEVIC	023A	CDTHA1	0226	CDTHA2	0228
CDTMF3	022A	CDTMF4	022C	CDTMF5	022E	CDTHV1	0218
CDTMV2	021A	CDTMV3	021C	CDTMV4	021E	CDTMV5	0220
CDUBL	EF26	CH	02FC	CHI	02F2	CHACT	02F3
CHACTL	D401	CHAR	02FA	CHBAS	02F4	CHBASE	D409
CHKERR	00BF	CHKENT	003B	CHKSUM	0031	CHRRQ	E000
CIOCHR	002F	CIOINV	E46E	CIDORG	E4A6	CIOV	E456
CIX	00F2	CKEY	004A	CLOSE	000C	CMODE	EF1E
COLAC	0072	COLBK	D01A	COLCRS	0055	COLDST	0244
COLDSV	E477	COLINC	007A	COLOR0	02C4	COLOR1	02C5
COLOR2	02C6	COLOR3	02C7	COLOR4	02C8	COLPFO	D016
COLPF1	D017	COLPF2	D018	COLPF3	D019	COLPMO	D012
COLPM1	D013	COLPM2	D014	COLPM3	D015	COLRSH	004F
CONSOL	D01F	COS	B073	COUNTR	007E	CR	009B
CRETRY	0036	CRITIC	0042	CRNTPC	EF41	CRBINH	02F0
CRBROR	008D	CRIDE	EF2E	CRDPIV	E47D	CSTAT	0288
CTIA	D000	D	0044	DAUX1	030A	DAUX2	030B
DBSECT	0241	DBUFHI	0305	DBUFLO	0304	DBUFSZ	0014
DBYTHI	0309	DBYTLO	0308	DCB	0300	DCOMND	0302
DDEVIC	0300	DEQFLO	00FB	DEQON	0006	DELETE	0021
DELTAC	0077	DELTAR	0076	DERROR	0090	DFLAGB	0240
DIGRT	00F1	DINDEX	0057	DIRECT	0002	DISK	0044
DISKIV	E450	DISPLY	0053	DL18TH	D403	DL18TL	D402
DMACTL	D400	DMASK	02A0	DNACK	008B	DOSINI	000C
DOSVEC	000A	DRAWLN	0011	DRETRY	0037	DRKMSK	004E
DSKFMS	001B	DSKINV	E453	DSKORG	EDEA	DSKTIM	0246
DSKUTL	001A	DSPFLO	02FE	DSTAT	004C	DSTATS	0303
DTIMLO	0306	DUNIT	0301	DUNUBE	0307	DVBSTAT	02EA
EDITRV	E400	EEXP	00ED	ENDPT	0074	EOFERR	008B
ERRFLO	023F	ESCFLO	02A2	ESIGN	00EF	EXP	DDC0
EXPIO	DDCC	FADD	DA66	FASC	D8E6	FCHRFI	00F0
FDIV	DB2B	FEOF	003F	FILDAT	02FD	FILFLO	02B7
FILLBF	EEC3	FILLIN	0012	FLDOP	DD8D	FLDOR	DD89
FLDIP	DD9C	FLDIR	DD9B	FLPTR	00FC	FMOVE	DD86
FMSZPO	0043	FMUL	DADB	FNCNOT	0092	FORMAT	0022
FPI	D9D2	FPREC	0006	FPSCR	05E6	FPSCR1	05EC
FPTR2	00FE	FRO	00D4	FR1	00E0	FR2	00E6
FRE	00DA	FREQ	0040	FRMERR	00BC	FRX	00EC
FSCR	05E6	FSCR1	05EC	FSTOP	DDAB	FSTOR	DDA7
FSUB	DA30	FTYPE	003E	GETCHR	0007	GETREC	0005
QLBABB	02E0	QERR	EF3D	QPRIOR	026F	QACTL	D01D
GRAFM	D011	GRAFP0	D00D	GRAFP1	D00E	GRAFP2	D00F
GRAFP3	D010	HATABB	031A	HITCLR	D01E	HOLD1	0051
HOLD2	029F	HOLD3	029D	HOLD4	02BC	HOLD5	02BD
HOLDCH	007C	HPOBMO	D004	HPOBM1	D005	HPOBM2	D006
HPOBM3	D007	HPOBPO	D000	HPOBP1	D001	HPOBP2	D002

HP08P3	D003	HBCROL	D404	ICAX1	034A	ICAX1Z	002A
ICAX2	034B	ICAX2Z	002B	ICBAH	0345	ICBAHZ	0025
ICBAL	0344	ICBALZ	0024	ICBLH	0349	ICBLHZ	0029
ICBLL	0348	ICBLLZ	0028	ICCOM	0342	ICCOMT	0017
ICCOMZ	0022	ICDNO	0341	ICDNOZ	0021	ICDID	0340
ICDIDZ	0020	ICIDNO	002E	ICPTH	0347	ICPTHZ	0027
ICPTL	0346	ICPTLZ	0026	ICSPR	034C	ICSPRZ	002C
ICSTA	0343	ICSTAZ	0023	IFP	D9AA	INBUFF	00F3
INSCLR	0020	INSDAT	007D	INTAB5	0200	INTENP	022D
INTINV	E46B	INTOR0	E6D5	INTZ88	0010	INVFL0	02B6
IOCB	0340	IOCBAS	0020	IOCB8Z	0010	IOCFRE	00FF
IRGEN	D20E	IRGBT	D20E	KBCODE	D209	KBD	004B
KBDOR0	F3E4	KEYBDV	E420	KEYDEL	02F1	LBFEED	03FF
LBPR1	057E	LBPR2	037F	LBUFF	0580	LEDGE	0002
LINBUF	0247	LINZ88	0000	LMARON	0052	LOCKFL	0023
LOG	DECD	LOG10	DED1	LOGCOL	0063	LOGMAP	02B2
LPENH	0234	LPENV	0235	MOPF	D000	MOPL	D00B
M1PF	D001	M1PL	D009	M2PF	D002	M2PL	D00A
M3PF	D003	M3PL	D00B	MAXDEV	0021	MAXIOC	0080
MEMLO	02E7	MEMTOP	02E5	MLTTP	0066	MODEM	004D
MONOR0	FOE3	MXDMOD	0010	N	004E	NBUF8Z	002B
NEWCOL	0061	NEWROW	0060	NMIEN	D40E	NMIREB	D40F
NMIST	D40F	NOCKSM	003C	NONDEV	00B2	NOTE	0026
NOTOPN	0085	NSIGN	00EE	NVALID	0084	OLDADR	005E
OLDCHR	005D	OLDCOL	005B	OLDROW	005A	OPEN	0003
OPNIN	0004	OPNINO	000C	OPNOT	000B	OPNOUT	0002
OPNTHP	0066	QVRRUN	008E	P0PF	D004	P0PL	D00C
P1PF	D003	P1PL	D00D	P2PF	D006	P2PL	D00E
P3PF	D007	P3PL	D00F	PACTL	D302	PADDL0	0270
PADDL1	0271	PADDL2	0272	PADDL3	0273	PADDL4	0274
PADDL5	0275	PADDL6	0276	PADDL7	0277	PBCTL	D303
PBPNT	001D	PBUF8Z	001E	PCOLR0	02C0	PCOLR1	02C1
PCOLR2	02C2	PCOLR3	02C3	PDEVN	0040	PENH	D40C
PENV	D40D	PHCHLO	EE7F	PHCLOB	EEDC	PHINIT	EE7B
PHOPEN	EE9F	PHPUT	EF14	PHSTAT	EE81	PHSTLO	EE7D
PHWRIT	EEA7	PIA	D300	PLYAR0	05E0	PLYEVL	DD40
PMBASE	D407	POINT	0025	POKEY	D200	POKMBK	0010
PORTA	D300	PORTB	D301	POT0	D200	POT1	D201
POT2	D202	POT3	D203	POT4	D204	POT5	D205
POT6	D206	POT7	D207	POT00	D20B	PRINTR	0050
PRINTV	E430	PRIOR	D01B	PRMODE	EF1A	PRNBUF	03C0
PRNOR0	EE7B	PRNSPR	0014	PRVOPN	0081	PSIOC	EF01
PTEMP	001F	PTIMOT	001C	PTR100	027C	PTR101	027D
PTR102	027E	PTR103	027F	PTR104	0280	PTR105	0281
PTR106	0282	PTR107	0283	PUTCHR	000B	PYREC	0009
RADFL0	00FB	RADON	0000	RAMLO	0004	RAMBIZ	02E4
RAMTOP	006A	RANDOM	D20A	RBLOKV	E47A	RONLY	0087
RECVDN	0039	REDGE	0027	RENAME	0020	RMARON	0053
ROWAC	0070	ROWCR8	0034	ROWINC	0079	RTCLOK	0012
S	0053	SAVADR	006B	SAVID	0316	SAVMSC	005B
SBUF8Z	001D	SCREDIT	0045	SCRENV	E410	SCRFL0	02BB
SCRMEM	0093	SDLSTH	0231	SDLSTL	0230	SDMCTL	022F
SENDEV	E46B	BERIN	D20D	SEROUT	D20D	SETBSZ	EF34
SETDCB	EEE6	SETUBV	E45C	SHFANT	006F	SHFLOW	02BE
SIN	BDB1	SIOINV	E465	SIOOR0	E944	SIOV	E459
SIZEM	D00C	SIZEP0	D00B	SIZEP1	D009	SIZEP2	D00A
SIZEP3	D00B	SKCTL	D20F	SKREB	D20A	SKSTAT	D20F
SOUNDR	0041	SPACE	0020	SPECIL	000E	SQR	BEB1
SRTIMR	022B	SSFLA0	02FF	SSKCTL	0232	STACKP	031B
STATC	0053	STATIS	000D	STATUS	0030	STICK0	027B

PRINTER ***** PRINTP. SRC ***** 3/9/79 ***** 4:00:00 P. M.

BTICK1	0279	BTICK2	027A	BTICK3	027B	BTIMER	D209
BTRIO0	0284	BTRIO1	0285	BTRIO2	0286	BTRIO3	0287
SUBTMP	029E	SUCCEB	0001	BWPFL0	007B	BYBV8V	E45F
TABMAP	02A3	TEMP	023E	TEMP1	0312	TEMP2	0314
TEMP3	0315	TIMER1	030C	TIMER2	0310	TIMFL0	0317
TIMOUT	008A	TINDEX	0293	TMPCHR	0050	TMPCOL	02B9
TMPLBT	02A1	TMPROW	0288	TMPX1	029C	TRAMBZ	0006
TRIO0	D010	TRIO1	D011	TRIO2	D012	TRIO3	D013
TRNRCD	0089	TSTAT	0319	TSTDAT	0007	TXTCOL	0291
TXTMSC	0294	TXTOLD	0296	TXTROW	0290	UNLOCK	0024
USAREA	0480	VBREAK	0206	VCOUNT	D40B	VCTABL	E480
VDELAY	D01C	VDBLBT	0200	VECTBL	E400	VIMING	0216
VINTER	0204	VKEYBD	0208	VPRCD	0202	VSCROL	D405
VBERIN	020A	VSEROC	020E	VBEROR	020C	VTIMR1	0210
VTIMR2	0212	VTIMR4	0214	VVBLKD	0224	VVBLK1	0222
WARMST	0008	WARMSV	E474	WMODE	0289	WRITEC	0057
WRONLY	0083	WBYNC	D40A	XITVBV	E462	XMTDON	003A
ZIOCB	0020	ZTEMP1	00F5	ZTEMP3	00F9	ZTEMP4	00F7

CASSET HANDLER 3/12 (DK1: CASCV)

		TITLE	'CASSET HANDLER 3/12 (DK1: CASCV)'
0003	CBUFH	=	CABBUF/256
00FD	CBUFL	=	-256*CBUFH+CABBUF
0040	BRSTA	=	\$40 BIO READ STATUS
0080	BWSTA	=	\$80 BIO WRITE STATUS
0034	MOTRQD	=	\$34
003C	MOTRST	=	\$3C

00FC	DTA	=	\$FC DATA RECORD TYPE BYTE
00FA	DT1	=	\$FA LAST DATA RECORD
00FE	EOT	=	\$FE END OF TAPE
00FB	HDR	=	\$FB HEADER
0002	TONE1	=	2 CHANGE TO RECORD MODE TONE
0001	TONE2	=	1 PRESS PLAY TONE

E440 4B EF 2A		*-CABETV	
E443 F0 D5 EF		WORD	OPENC-1, CLOSEC-1, QBYTE-1, PBYTE-1, STATU-1, SPECIAL-1
E446 0F F0 27			
E449 F0 4A EF			
E44C 4C 41 EF	JMP	INIT	
E44F 00	BYTE	0	ROM FILLER BYTE

USED IN MONITP FOR CASSETTE BOOT

E47A 4C E9 EF	*-RBLOKV	
	JMP	RBLOK

E47D 4C 5D EF	*-CSOPIV	
	JMP	OPINP

*-CASDR0

INIT ROUTINE

EF41 A9 CC	INIT	LDA	\$*CC	
EF43 8D EE 02		STA	CBAUDL	
EF46 A9 05		LDA	\$*05	
EF48 8D EF 02		STA	CBAUDH	SET CASSET BAUD RATE TO 600
EF4B 60	SPECIAL	RTS		THAT'S ALL FOLKS

OPEN FUNCTION - WITH NO TIMING ADJUST

EF4C A5 2B	OPENC	LDA	ICAX2Z	GET AX2
EF4E B5 3E		STA	FTYPE	SAVE IT FOR FUTURE REFERENCE
EF50 A5 2A		LDA	ICAX1Z	
EF52 29 0C		AND	##0C	IN AND OUT BITS
EF54 C9 04		CMP	##04	
EF56 F0 05		BEG	OPINP	
EF58 C9 08		CMP	##08	SEE IF OPEN FOR OUTPUT
EF5A F0 39		BEG	OPOUT	
EF5C 60		RTS		IF ALREADY OPEN, RETURN LEAVING STATUS=##84
EF5D A9 00	OPINP	LDA	#0	
EF5F BD B9 02		STA	WMODE	SET READ MODE
EF62 B5 3F		STA	FEUF	NO EOF YET
EF64 A9 01	BFH	LDA	#TONE2	TONE FOR PRESS PLAY
EF66 20 58 F0		JBR	BEEP	GO BEEP
EF68 30 24		BHI	OPNRTN	IF ERROR DURING BEEP
EF6B A9 34		LDA	#MOTR00	
EF6D BD 02 D3		STA	PACTL	TURN MOTOR ON
EF70 A0 40		LDY	##40	5-31-79 9 SEC READ LEADER
EF72 A2 02		LDX	#2	
EF74 A9 03		LDA	#3	
EF76 BD 2A 02		STA	CDTMF3	
EF79 20 5C E4		JBR	SETVBV	SET UP VBLANK TIMER
EF7C AD 2A 02	WAITTH	LDA	CDTMF3	
EF7F D0 FB		BNE	WAITTH	WAIT FOR MOTOR TO COME UP TO SPEED
EF81 A9 80		LDA	##80	NEXT BYTE=NO BYTES IN BUFFER
EF83 B5 3D		STA	BPTR	
EF85 BD BA 02		STA	BLIN	
EF88 4C D3 EF		JMP	DPOK	OPEN OK

OPEN FOR OUTPUT

EF8B A0 80	PBRK	LDY	#BRKABT	BREAK KEY ABORT STATUS
EF8D C6 11		DEC	BRKKEY	RESET BREAK KEY
EF8F A9 00	OPNRTN	LDA	#0	CLEAR WRITE MODE FLAG
EF91 BD B9 02		STA	WMODE	
EF94 60		RTS		AND EXIT.
EF95 A9 80	OPOUT	LDA	##80	
EF97 BD B9 02		STA	WMODE	SET WRITE MODE
EF9A A9 02		LDA	#TONE1	TELL USER TO TURN ON RECORD MODE
EF9C 20 58 F0		JBR	BEEP	
EF9F 30 EE		BHI	OPNRTN	IF ERROR DURING BEEP
EFA1 A9 CC		LDA	##CC	SET BAUD RATE
EFA3 BD 04 D2		STA	AUDF3	WHICH SEEMS TO BE NESSECARY
EFA6 A9 05		LDA	##05	FOR SOME OBSCURE REASON
EFA8 BD 06 D2		STA	AUDF4	
EFA8 A9 60		LDA	##60	
EFA0 BD 00 03		STA	DDEVIC	
EFB0 20 68 E4		JBR	SENDEV	TELL POKEY TO WRITE MARKS
EFB3 A9 34		LDA	#MOTR00	WRITE 5 SEC BLANK TAPE
EFB5 BD 02 D3		STA	PACTL	
EFB8 A7 03		LDA	#3	
EFBA A2 04		LDX	#4	5/30/79 20 SEC LEADER
EFBC A0 80		LDY	##80	
EFBE 20 5C E4		JBR	SETVBV	
EFC1 A9 FF		LDA	##FF	
EFC3 BD 2A 02		STA	CDTMF3	

CASSET HANDLER 3/12 (DK1: CASCV)

EFC6 A5 11	WDLR	LDA	BRKKEY	
EFC8 F0 C1		BEG	PBRK	IF BREAK DURING WRITE LEADER
EFC9 AD 2A 02		LDA	CDTMF3	
EFCD D0 F7		BNE	WDLR	
EFCF A9 00		LDA	#0	INIT BUFFER POINTER
EFD1 85 3D		STA	BPTR	
EFD3 A0 01	OPDK	LDY	#SUCCEB	
EFD5 60		RTB		

CASSET HANDLER 3/12 (DK1: CASCV)

GET BYTE

EFD6 A5 3F	QBYTE	LDA	FE0F	IF AT EOF ALREADY
EFDB 30 33		BMI	ISEOF	RETURN EOF STATUS
EFDA A6 3D		LDX	BPTR	BUFFER POINTER
EFDC EC 8A 02		CPX	BLIM	IF END OF BUFFER
EFDF F0 08		BEQ	RBLOK	READ ANOTHER BLOCK
EFE1 BD 00 04		LDA	CASBUF+3,X	GET NEXT BYTE
EFE4 E6 3D		INC	BPTR	BUMP POINTER
EFE6 A0 01		LDY	#SUCCEB	OK STATUS
EFE8 60	QBX	RTS		
EFE9 A9 52	RBLOK	LDA	#'R	READ OP CODE
EFEB 20 75 F0		JSR	\$IO\$B	\$IO ON SYS BUF
EFEF 98		TYA		
EFEE 30 F7		BMI	QBX	IF \$IO ERRORS, RETURN
EFF1 A9 00		LDA	#0	
EFF3 B5 3D		STA	BPTR	RESET POINTER
EFF5 A2 80		LDX	#80	DEFAULT # BYTES
EFF7 AD FF 03		LDA	CASBUF+2	
EFFA C9 FE		CHP	#EOT	
EFFC F0 0D		BEQ	ATEOF	IF HEADER, GO READ AGAIN
EFDE C9 FA		CHP	#DT1	IF LAST DATA REC
F000 D0 03		BNE	NLR	
F002 AE 7F 04		LDX	CASBUF+130	LAST DATA RECORD, GET # BYTES
F003 BE 8A 02	NLR	STX	BLIM	
F00B 4C D6 EF		JMP	QBYTE	GET NEXT BYTE
F00B C6 3F	ATEOF	DEC	FE0F	SET FE0F
F00D A0 8B	ISEOF	LDY	#EOFERR	ENDFILE STATUS
F00F 60		RTS		

PUT BYTE TO BUFFER

F010 A6 3D	PBYTE	LDX	BPTR	BUFFER POINTER
F012 9D 00 04		STA	CASBUF+3, X	STORE CHAR AWAY
F015 E6 3D		INC	BPTR	BUMP POINTER
F017 A0 01		LDY	#SUCCEB	OK STATUS
F019 E0 7F		CPX	#127	IF BUFFER FULL
F01B F0 01		BEG	++3	
F01D 60		RTS		

WRITE OUT THE BUFFER

F01E A9 FC		LDA	#DTA	RECORD TYPE = DATA
F020 20 D2 F0		JBR	WBIOBB	DO WRITE ON SYSTEM BUFFER
F023 A9 00		LDA	#0	
F025 85 3D		STA	BPTR	REBET BUFFER POINTER
F027 60		RTS		EXIT.

CASSET HANDLER 3/12 (DK1: CASCV)

STATUS - RETURN STATUS INFO THRU DVSTAT

F02B A0 01
F02A 60

STATU LDY
RTS

#SUCCEB

CLOSE

F02B AD 89 02	CLOSEC	LDA	WMODE	SEE IF WRITING
F02E 30 08		BMI	CLWRT	00 CLOSE FOR WRITE
CLOSE FOR READ - FLAG CLOSED				
F030 A0 01		LDY	#SUCCESS	SUCCESSFULL
F032 A9 3C	FCAX	LDA	#MOTRST	STOP THE MOTOR IN CASE WAS SHORT IRO MODE
F034 8D 02 D3		STA	PACTL	
F037 60		RTS		
F03B A6 3D	CLWRT	LDX	BPTR	BUFFER POINTER
F03A F0 0A		BEQ	WTLR	IF NO DATA BYTES IN BUFFER, NO DT1 REC
F03C 8E 7F 04		STX	CASBUF+130	WRITE TO LAST RECORD
F03F A9 FA		LDA	#DT1	REC TYPE
F041 20 D2 F0		JSR	WB108B	WRITE OUT USER BUFFER
F044 30 EC		BMI	FCAX	00 IF ERROR
F046 A2 7F	WTLR	LDX	#127	ZERO BUFFER
F04B A9 00		LDA	#0	
F04A 9D 00 04	ZTBUF	STA	CASBUF+3, X	
F04D CA		DEX		
F04E 10 FA		BPL	ZTBUF	
F050 A9 FE		LDA	#EOT	WRITE EOT RECORD
F052 20 D2 F0		JSR	WB108B	
F055 4C 32 F0	JMP	FCAX	FLAG CLOSED AND EXIT	

SUBROUTINES

BEEP - GENERATE TONE ON KEYBOARD SPEAKER
ON ENTRY A= FREQ

F05B 85 40	BEEP	STA	FREQ	
F05A A3 14	BEEP1	LDA	RTCLK+2	CURRENT CLOCK
F05C 1B		CLC		
F05D 69 1E		ADC	#30	1 SEC TONE
F05F AA		TAX		
F060 A9 FF	WFL	LDA	#FF	
F062 8D 1F D0		STA	CONSO	TURN ON SPEAKER
F063 A9 00		LDA	#0	
F067 A0 F0		LDY	#F0	
F069 8B		DEY		
F06A D0 FD		BNE	*-1	
F06C 8D 1F D0		STA	CONSO	TURN OFF SPEAKER
F06F A0 F0		LDY	#F0	
F071 8B		DEY		
F072 D0 FD		BNE	*-1	
F074 E4 14		CPX	RTCLK+2	SEE IF 1 SEC IS UP YET
F076 D0 EB		BNE	WFL	
F078 C6 40		DEC	FREQ	COUNT BEEPS
F07A F0 0B		BEG	WFAK	IF ALL DONE GO WAIT FOR KEY
F07C 9A		TXA		
F07D 1B		CLC		
F07E 69 0A		ADC	#10	
F080 AA		TAX		
F081 E4 14		CPX	RTCLK+2	
F083 D0 FC		BNE	*-2	
F085 F0 D3		BEG	BEEP1	UNCOND GO BEEP AGAIN
F087 20 8C F0	WFAK	JBR	WFAK1	USE SIMULATED "JMP (K0ETCH)"
F08A 9B		TYA		
F08B 60		RTS		
F08C AD 25 E4	WFAK1	LDA	KEYBDV+3	
F08F 4B		PHA		
F090 AD 24 E4		LDA	KEYBDV+4	SIMULATE "JMP (K0ETCH)"
F093 4B		PHA		
F094 60		RTS		

BIOBB - CALL BIO ON SYSTEM BUFFER

F095 8D 02 03	BIOBB	STA	DCOMND	SAVE COMMAND
F098 A9 00		LDA	#0	
F09A 8D 09 03		STA	DBYTHI	SET BUFFER LENGTH
F09D A9 83		LDA	#131	
F09F 8D 0B 03		STA	DBYTL0	
F0A2 A9 03		LDA	#CBUFH	
F0A4 8D 05 03		STA	DBUFHI	SET BUFFER ADDRESS
F0A7 A9 FD		LDA	#CBUFL	
F0A9 8D 04 03		STA	DBUFLO	
F0AC A9 60	CBIO	LDA	#60	CASSET PSEUDO DEVICE
F0AE 8D 00 03		STA	DDEVIC	
F0B1 A9 00		LDA	#0	
F0B3 8D 01 03		STA	DUNIT	
F0B6 A9 23		LDA	#35	DEVICE TIMEOUT (5/30/79)
F0B8 8D 04 03		STA	DTIMLO	
F0BB AD 02 03		LDA	DCOMND	GET COMMAND BACK
F0BE A0 40		LDY	#SRSTA	BIO READ STATUS COMMAND

CASSET HANDLER 3/12 (DK1: CASCV)

FOC0 C9 52	CMP	#'R	
FOC2 F0 02	BEG	##4	
FOC4 A0 80	LDY	#SWSTA	SIO WRITE STATUS COMMAND
FOC6 BC 03 03	STY	DSTAT8	SET STATUS FOR SIO
FOC9 A3 3E	LDA	FTYPE	
FOCB 8D 0B 03	STA	DAUX2	INDICATE IF SHORT IRO MODE
FOCE 20 59 E4	JBR	SIOV	GO CALL SIO
FOD1 60	RTS		

WSIOSB - WRITE SIO SYSTEM BUFFER

FOD2 8D FF 03	WSIOSB	STA	CASBUF+2	STORE TYPE BYTE
FOD5 A9 55		LDA	##55	
FOD7 8D FD 03		STA	CASBUF+0	
FODA 8D FE 03		STA	CASBUF+1	
FODD A9 57		LDA	#'W	WRITE
FODF 20 95 F0		JBR	SIOSB	CALL SIO ON SYSTEM BUFFER
FOE2 60		RTS		AND RETURN
FOE3	CRNTPC	==		

0014 00	CASBPR	. BYTE	MONDRO-CRNTPC	CASCV IS TOO LONG
		. END		

CASSET HANDLER 3/12 (DK1: CABCV)

SYMBOL TABLE

ADDCOR	030E	ADRESB	0D64	AFP	D800	ALLPOT	D2D8
ANTIC	D400	APPEND	0001	APPMHI	000E	ATACHR	02FB
ATAN	BE43	ATEOF	F00B	ATTRACT	D04D	AUDC1	D201
AUDC2	D203	AUDC3	D205	AUDC4	D207	AUDCTL	D208
AUDF1	D200	AUDF2	D202	AUDF3	D204	AUDF4	D206
BADIOC	0086	BADMOD	0091	BEEP	F05B	BEEP1	F05A
BFENHI	0035	BFENLO	0034	BITHSK	006E	BLIM	028A
BLKBDV	E471	BOOT7	0009	BOOTAD	0242	BOTSCR	02BF
BPTR	003D	BRKABT	0080	BRKKEY	0011	BUFADR	0015
BUFCNT	006B	BUFRFL	0038	BUFRHI	0033	BUFRLO	0032
BUFBTR	006C	CABBUF	03FD	CABETV	E440	CABFLO	030F
CASINI	0002	CASOR0	EF41	CASBBT	004B	CASSET	0043
CASBPR	0014	CAUX1	023C	CAUX2	023D	CBAUDH	D2EF
CBAUDL	02EE	CBUFR	0003	CBUFL	00FD	CCOMND	023B
CDEVIC	023A	CDTMA1	0226	CDTMA2	0228	CDTMF3	022A
CDTMF4	022C	CDTMF5	022E	CDTMV1	0218	CDTMV2	D21A
CDTMV3	021C	CDTMV4	021E	CDTMV5	0220	CH	02FC
CH1	02F2	CHACT	02F3	CHACTL	D401	CHAR	02FA
CHBAB	02F4	CHBASE	D409	CHKERR	008F	CHK8NT	003B
CHK8UM	0031	CHRDRO	E000	CIOCHR	002F	CIOINV	E46E
CIDDR0	E4A6	CIOV	E456	CIX	00F2	CKEY	004A
CLOSE	000C	CLOSEC	F02B	CLWRT	F03B	COLAC	0072
COLBK	D01A	COLCRB	0055	COLDBT	0244	COLD8V	E477
COLINC	007A	COLOR0	02C4	COLOR1	02C5	COLOR2	02C6
COLOR3	02C7	COLOR4	02C8	COLPFO	D016	COLPF1	D017
COLPF2	D018	COLPF3	D019	COLPH0	D012	COLPH1	D013
COLPM2	D014	COLPM3	D015	COLRSH	004F	CONSOL	D01F
COS	D073	COUNTR	007E	CR	009B	CRETRY	0036
CRITIC	0042	CRNTPC	F0E3	CRBINH	02F0	CRBROR	00BD
CSIO	F0AC	CSOPIV	E47D	CBTAT	02BB	CTIA	D000
DAUX1	030A	DAUX2	030B	DBSECT	D241	DBUFHI	0305
DBUFLO	0304	DBVTHI	0309	DBYTLO	0308	DCB	0300
DCOMND	0302	DDEVIC	0300	DEOFL0	00FB	DECON	00D6
DELETE	0021	DELTAC	0077	DELTAR	0076	DERROR	0090
DIAGN	0045	DIORT	00F1	DINDEX	0057	DIRECT	0002
		DISKIV	E450	DISPLY	0053	DL18TH	D4D3
		DMACTL	D400	DMA8K	02A0	DNACK	008B
		DOSVEC	006A	DRAWLN	0011	DRETRY	0037
DRMSK	004E	DSKFMB	0018	DSKINV	E453	DSKOR0	EDEA
DSKTIM	0246	DSKUTL	001A	DSPFLO	02FE	DSTAT	004C
DSTATB	0303	DTI	00FA	DTA	00FC	DTIMLO	0306
DUNIT	0301	DUNUBE	0307	DVBSTAT	02EA	EDITRV	E400
EEXP	00ED	ENDPT	0074	EOFERR	00BB	EOT	D0FE
ERRFLO	023F	EBCFLO	02A2	EBIGN	00EF	EXP	DDC0
EXP10	DDCC	FADD	DA66	FABC	D8E6	FCAX	F032
FCHRFL	00F0	FDIV	D828	FEOF	003F	FILDAT	02FD
FILFLO	02B7	FILLIN	0012	FLDOP	DD8D	FLDOR	DD89
FLDIP	DD9C	FLDIR	DD98	FLPTR	DOFC	FMOVE	DD86
FMBZPO	0043	FMUL	DAD8	FNCNOT	0092	FORMAT	DD22
FPI	D9D2	FPREC	0006	FPSCR	05E6	FPSCR1	05EC
FPTR2	00FE	FRO	D0D4	FR1	00E0	FR2	D0E6
FRE	00DA	FREQ	0040	FRMERR	008C	FRX	00EC
FSCR	05E6	FSCR1	05EC	FSTOP	DDAB	F8TOR	DDA7
FSUB	DA60	FTYPE	003E	QBX	EFE8	QBYTE	EFD6
GETCHR	0007	GETREC	0005	QLBABB	02E0	QPRIOR	026F
GRACTL	D01D	GRAFH	D011	GRAFFO	D00D	GRAFP1	D00E
GRAFP2	D00F	GRAFP3	D010	HATABB	031A	HDR	D0FB
HITCLR	D01E	HOLD1	0051	HOLD2	029F	HOLD3	029D
HOLD4	028C	HOLD5	028D	HOLDCH	007C	HP08M0	D004

CASSET HANDLER 3/12 (DK1: CABCV)

HPOSM1	D005	HPOBM2	D006	HPOBM3	D007	HPOSP0	D000
HPOBP1	D001	HPOBP2	D002	HPOBP3	D003	HSCROL	D404
ICAX1	034A	ICAX1Z	002A	ICAX2	034B	ICAX2Z	002B
ICBAH	0345	ICBAHZ	0025	ICBAL	0344	ICBALZ	0024
ICBLH	0349	ICBLHZ	0029	ICBLL	0348	ICBLLZ	0028
ICCOM	0342	ICCOMT	0017	ICCOMZ	0022	ICDND	0341
ICDNOZ	0021	ICHID	0340	ICHIDZ	0020	ICIDNO	002E
ICPTH	0347	ICPTHZ	0027	ICPTL	0346	ICPTLZ	0026
ICSPR	034C	ICSPRZ	002C	ICSTA	0343	ICSTAZ	0023
IFP	D7AA	INBUFF	00F3	INIT	EF41	INBCLR	0020
INSDAT	007D	INTAB8	0200	INTEMP	022D	INTINV	E46B
INTORG	E6D5	INTZ88	0010	INVFL0	0286	IUCB	0340
IOCBAB	0020	IOCB8Z	0010	IOCFRE	00FF	IRGEN	D20E
IRGST	D20E	ISEOF	F00D	KBCODE	D209	KRD	004B
KDDOR0	F3E4	KEYBDV	E420	KEYDEL	02F1	LBFEND	05FF
LBPR1	057E	LBPR2	057F	LBUFF	0580	LED0E	0002
LINBUF	0247	LINZ88	0000	LMARON	0052	LOCKFL	0023
LOG	DECD	LO010	DE01	LOGCOL	0063	LOGHAP	0282
LPENH	0234	LPENV	0235	MOPF	D000	MOPL	D00B
M1PF	D001	M1PL	D009	M2PF	D002	M2PL	D00A
F	D003	M3PL	D00B	MAXDEV	0021	MAXIOC	0080
LLO	02E7	MEMTOP	02E5	MLTTP	0066	MODEM	004D
MONOR0	F0E3	MOTRO0	0034	MOTRBT	003C	MXDMOD	0010
NEWCOL	0061	NEWROW	0060	NLR	F005	NTIEN	D40E
NMIRE5	D40F	NMIST	D40F	NOCKSM	003C	NONDEV	0082
NOTE	0026	NOTOPN	0085	NSIGN	00EE	NVALID	0084
OLDADR	005E	OLDCHR	005D	OLDCOL	005B	OLDROW	005A
OPEN	0003	OPENC	EF4C	OPINP	EF5D	OPNIN	0004
OPNIND	000C	OPNOT	0008	OPNRTN	EFBF	OPNTMP	0066
OPOK	EFD3	OPDUT	EF95	OPRNUN	008E	OPPF	D004
POPL	D00C	P1PF	D005	P1PL	D00D	P2PF	D006
P2PL	D00E	P3PF	D007	P3PL	D00F	PACTL	D302
PADDL0	0270	PADDL1	0271	PADDL2	0272	PADDL3	0273
PADDL4	0274	PADDL5	0275	PADDL6	0276	PADDL7	0277
PBCTL	D303	PBPNT	001D	PBRK	EF8B	PBUF8Z	001E
PBYTE	F010	PCOLR0	02C0	PCOLR1	02C1	PCOLR2	02C2
PCOLR3	02C3	PENH	D40C	PENV	D40D	PIA	D300
PLYAR0	05E0	PLYEVL	DD40	PMBASE	D407	POINT	0025
POKEY	D200	POKMSK	0010	PORTA	D300	PORTB	D301
POT0	D200	POT1	D201	POT2	D202	POT3	D203
POT4	D204	POT5	D205	POT6	D206	POT7	D207
POT00	D208	PRINTR	0050	PRINTV	E430	PRIOR	D01B
PRNBUF	03C0	PRNOR0	EE78	PRVOPN	0081	PTEMP	001F
PTIMOT	001C	PTRIG0	027C	PTRIG1	027D	PTRIG2	027E
PTRIG3	027F	PTRIG4	0280	PTRIG5	0281	PTRIG6	0282
PTRIG7	0283	PUTCHR	000B	PUTREC	0009	RADFL0	00F8
RADON	0000	RAMLO	0004	RAMBIZ	02E4	RAHTOP	006A
RANDOM	D20A	RBLOK	EFE9	RBLOKV	E47A	RDONLY	00B7
RECVDN	0039	RED0E	0027	RENAME	0020	RMARON	0033
ROWAC	0070	ROWCR8	0054	ROWINC	0079	RTCLOK	0012
SAVADR	0068	SAVID	0316	SAVMSC	005B	SCREDT	0045
SCREENV	E410	SCRFL0	02B8	SCRMEH	0093	SDL8TH	0231
SDL8TL	0230	SDMCTL	022F	SENDEV	E468	SERIN	D20D
SEROUT	D20D	SETV8V	E45C	SFH	EF64	SHFANT	006F
SHFL0K	02BE	SIN	B0B1	SIOINV	E465	SIOOR0	E944
SIO8B	F095	SIOV	E459	SIZEM	D00C	SIZEP0	D00B
SIZEP1	D009	SIZEP2	D00A	SIZEP3	D00B	SKCTL	D20F
SKRES	D20A	SKSTAT	D20F	BOUNDR	0041	SPECIA	EF4B
SPECIL	000E	SQR	BEB1	SRSTA	0040	SRTIMR	022B
BSFLAQ	02FF	SSKCTL	0232	STACKP	031B	STATIB	000D

CASSET HANDLER 3/12 (DK1: CASCV)

STATU	F028	STATUS	0030	BTICK0	0278	BTICK1	0279
BTICK2	027A	BTICK3	027B	BTIMER	0209	STR100	0284
STR101	0285	STR102	0286	STR103	0287	SUBTMP	029E
SUCCEB	0001	SWPFL0	0078	SWSTA	0080	SYSVBV	E45F
TABMAP	02A3	TEMP	023E	TEMP1	0312	TEMP2	0314
TEMP3	0315	TIMER1	030C	TIMER2	0310	TIMFL0	0317
TIMOUT	008A	TINDEX	0293	TMPCHR	0050	TMPCOL	0289
TMPLBT	02A1	TMPROW	0288	TMPI1	029C	TONE1	0002
TONE2	0001	TRAMSZ	0006	TR100	0010	TR101	0011
TR102	0012	TR103	0013	TRNRCD	0089	TSTAT	0319
TSTDAT	0007	TXTCOL	0291	TXTMBC	0294	TXTOLD	0296
TX1ROW	0290	UNLOCK	0024	UBAREA	0480	VBREAK	0206
VCOUNT	D408	VCTABL	E480	VDELAY	D01C	VDSLST	0200
VECTBL	E400	VIMIR0	0216	VINTER	0204	VKEYBD	0208
VPRCD	0202	VSCROL	D405	VSERIN	020A	VSEROC	020E
VSEROR	020C	VTIMR1	0210	VTIMR2	0212	VTIMR4	0214
VVBLKD	0224	VVBLKI	0222	WAITTH	EF7C	WARMST	0008
WARMST	E474	WDLR	EFC6	WFAK	F0B7	WFAK1	F0BC
WFL	F060	WMODE	0289	WONLY	0083	WB108B	F0D2
WSYNC	D40A	WTLR	F046	XITVBV	E462	XMTDON	003A
Z10CB	0020	ZTBUF	F04A	ZTEMP1	00F5	ZTEMP3	00F9
ZTEMP4	00F7						