

```

1586 ; TITLE 'SIO ( SERIAL BUS INPUT/OUTPUT CONTROLLER ) '
1587 ; COLLEEN OPERATING SYSTEM
1588 ;
1589 ; SID ( SERIAL BUS INPUT/OUTPUT CONTROLLER )
1590 ; WITH SOFTWARE BAUD RATE CORRECTION ON CASSETTE
1591 ;
1592 ;
1593 ; AL MILLER 3-APR-79
1594 ;
1595 ;
1596 ; THIS MODULE HAS ONE ENTRY POINT. IT IS CALLED BY THE DEVICE
1597 ; HANDLERS. IT INTERPRETS A PREVIOUSLY ESTABLISHED DEVICE CONTROL
1598 ; BLOCK (STORED IN GLOBAL RAM) TO ISSUE COMMANDS
1599 ; TO THE SERIAL BUS TO CONTROL TRANSMITTING AND RECEIVING DATA.
1600 ;
1601 ;
1602 ;
1603 ;
1604 ; .PAGE
1605 ; EQUATES
1606 ;
1607 ; DCD DEVICE BUS ID NUMBERS
1608 0030 FLOPPY = $30
1609 ; PRINTR = $40
1610 ; CASSET = $60 ; !!!!! *****
1611 0060 CASET = $60 ; !!!!! *****
1612 ;
1613 ;
1614 ; BUS COMMANDS
1615 ;
1616 0052 READ = 'R
1617 0057 WRITE = 'W
1618 ; STATIS = 'S
1619 ; FORMAT = '!'
1620 ;
1621 ;
1622 ; COMMAND AUX BYTES
1623 ;

```

```

1624 0053      SIDWAY      =      'S      ; PRINT 16 CHARACTERS SIDEWAYS
1625 004E      NORMAL      =      'N      ; PRINT 40 CHARACTERS NORMALLY
1626 0044      DOUBLE      =      'D      ; PRINT 20 CHARACTERS DOUBLE WIDE
1627 0050      PLOT        =      'P      ; PLOT MODE
1628           ;
1629           ;
1630           ; BUS RESPONSES
1631           ;
1632 0041      ACK          =      'A      ; DEVICE ACKNOWLEDGES INFORMATION
1633 004E      NACK         =      'N      ; DEVICE DID NOT UNDERSTAND
1634 0043      COMPLT       =      'C      ; DEVICE SUCCESSFULLY COMPLETED OPERATION
1635 0045      ERROR        =      'E      ; DEVICE INCURRED AN ERROR IN AN ATTEMPTED OP
1636           ;
1637           ;
1638           ; MISCELLANEOUS EQUATES
1639           ;
1640 0028      B192LO        =      $28      ; 19200 BAUD RATE POKEY COUNTER VALUES (LO BYTE)
1641 0000      B192HI        =      $00      ; 19200 BAUD (HI BYTE)
1642 00CC      B600LO        =      $CC      ; 600 BAUD (LO BYTE)
1643 0005      B600HI        =      $05      ; 600 BAUD (HI BYTE)
1644 0005      HITONE        =      $05      ; FSK HI FREQ POKEY COUNT VALUE (5326 HZ)
1645 0007      LOTONE        =      $07      ; FSK LO FREQ POKEY COUNTER VALUE (3995 HZ)
1646           ;
1647           .IF PALFLG
1648      WIRGLO      =      150      ; WRITE INTER RECORD GAP (IN 1/60 SEC)
1649      RIRGLO      =      100      ; READ INTER RECORD GAP (IN 1/60 SEC)
1650      WSIRG       =      13      ; SHORT WRITE INTER RECORD GAP
1651      RSIRG       =      8       ; SHORT READ INTER RECORD GAP
1652           .ENDIF
1653           .IF PALFLG-1
1654 00B4      WIRGLO      =      180      ; WRITE INTER RECORD GAP (IN 1/60 SEC)
1655 0078      RIRGLO      =      120      ; READ INTER RECORD GAP (IN 1/60 SEC)
1656 000F      WSIRG       =      15      ; SHORT WRITE INTER RECORD GAP
1657 000A      RSIRG       =      10      ; SHORT READ INTER RECORD GAP
1658           .ENDIF
1659 0000      WIRGHI        =      0
1660 0000      RIRGHI        =      0

```

```

1661
1662 0034 NCOMLO = $34 ; PIA COMMAND TO LOWER NOT COMMAND LINE
1663 003C NCOMHI = $3C ; PIA COMMAND TO RAISE NOT COMMAND LINE
1664 0034 MOTRGO = $34 ; PIA COMMAND TO TURN ON CASSETTE MOTOR
1665 003C MOTRST = $3C ; PIA COMMAND TO TURN OFF MOTOR
1666 ;
1667 0002 TEMPHI = TEMP/256 ; ADDRESS OF TEMP CELL (HI BYTE)
1668 003E TEMPLO = (-256)*TEMPHI+TEMP ; (LO BYTE)
1669 0002 CBUFHI = CDEVIC/256 ; ADDRESS OF COMMAND BUFFER (HI BYTE)
1670 003A CBUFLO = (-256)*CBUFHI+CDEVIC ; (LO BYTE)
1671 ;
1672 000D CRETRI = 13 ; NUMBER OF COMMAND FRAME RETRIES
1673 0001 DRETRI = 1 ; NUMBER OF DEVICE RETRIES
1674 0002 CTIMLO = 2 ; COMMAND FRAME ACK TIME OUT (LO BYTE)
1675 0000 CTIMHI = 0 ; COMMAND FRAME ACK TIME OUT (HI BYTE)
1676 ;
1677 ;
1678 ; JTADRH = JTIMER/256 ; HI BYTE OF JUMP TIMER ROUTINE ADDR "M
1679 ; JTADRL = (-256)*JTADRH+JTIMER ; "MOVED TO LINE 1428"
1680 ;
1681 .PAGE
1682 ; SIO
1683 ;
1684 ;
1685 *=SIOV
1686 E459 4C 59 E9 JMP SIO ; SIO ENTRY POINT
1687 ;
1688 *=SIOINV
1689 E465 4C 44 E9 JMP SIOINT ; SIO INITIALIZATION ENTRY POINT
1690 ;
1691 *=SENDEV
1692 E468 4C F2 EB JMP SENDEN ; SEND ENABLE ENTRY POINT
1693
1694 *=VCTABL-INTABS+VSERIN
1695
1696 E48A 0F EB .WORD ISRSIR ; VSERIN
1697 E48C 90 EA .WORD ISRODN ; VSEROR

```

```

1698 E48E CF EA          .WORD ISRTD          ; V SEROC
1699                      ;
1700                      ;
1701                      ;
1702                      *=SIOORG
1703                      ;
1704                      ; SIO INITIALIZATION SUBROUTINE
1705                      ;
1706 E944 A9 3C          SIOINT: LDA  #MOTRST
1707 E946 8D 02 D3       STA  PACTL          ; TURN OFF MOTOR
1708
1709 E949 A9 3C          LDA  #NCOMHI
1710 E94B BD 03 D3       STA  PBCTL :RAISE NOT COMMAND LINE
1711
1712
1713 E94E A9 03          LDA  #3
1714 E950 8D 32 02       STA  SSKCTL          ; GET POKEY OUT OF INITIALIZE MODE
1715 E953 85 41          STA  SOUND R          ; INIT POKE ADDRESS FOR QUIET I/O
1716 E955 8D 0F D2       STA  SKCTL
1717                      ;
1718                      ;
1719 E958 60             RTS   :RETURN
1720                      ;
1721                      ;
1722                      ;
1723                      ;
1724                      ;
1725                      ;
1726 E959 BA            SIO:   TSX
1727 E95A BE 18 03       STX  STACKP          ; SAVE STACK POINTER
1728 E95D A9 01          LDA  #1
1729 E95F 85 42          STA  CRITIC
1730                      ;
1731 E961 AD 00 03       LDA  DDEVIC
1732 E964 C9 60          CMP  #CASET
1733 E966 D0 03          BNE  NOTCST          ; BRANCH IF NOT CASSETTE
1734 E968 4C 80 EB       JMP  CASENT          ; OTHERWISE JUMP TO CASSETTE ENTER

```

```

1735 ;
1736 ; ALL DEVICES      EXCEPT CASSETTE ARE INTELLIGENT
1737 ;
1738 E96B A9 00      NOTCST:  LDA  40
1739 E96D 8D 0F 03      STA  CASFLG      ; INIT CASSETTE FLAG TO NO CASSETTE
1740 ;
1741 E970 A9 01      LDA  BDRETRI      ; SET NUMBER OF DEVICE RETRIES
1742 E972 85 37      STA  DRETRY
1743 E974 A9 0D      COMMND:  LDA  #CRETRI      ; SET NUMBER OF COMMAND FRAME RETRIES
1744 E976 85 36      STA  CRETRY
1745 ;
1746 ; SEND          A COMMAND FRAME
1747 ;
1748 E978 A9 28      COMFRM:  LDA  #B192LO      ; SET BAUD RATE TO 19200
1749 E97A 8D 04 D2      STA  AUDF3
1750 E97D A9 00      LDA  #B192HI
1751 E97F 8D 06 D2      STA  AUDF4
1752 ;
1753 E982 18          CLC          ; SET UP COMMAND BUFFER
1754 E983 AD 00 03      LDA  DDEVIC
1755 E986 6D 01 03      ADC  DUNIT
1756 E989 69 FF          ADC  4fff ; SUBTRACT 1
1757 E98B 8D 3A 02      STA  CDEVIC      ; SET BUS ID NUMBER
1758 ;
1759 E98E AD 02 03      LDA  DCOMND
1760 E991 8D 3B 02      STA  CCOMND      ; SET BUS COMMAND
1761 ;
1762 E994 AD 0A 03      LDA  DAUX1      ; STORE COMMAND FRAME AUX BYTES 1 AND 2
1763 E997 8D 3C 02      STA  CAUX1
1764 E99A AD 0B 03      LDA  DAUX2
1765 E99D 8D 3D 02      STA  CAUX2      ; DONE SETTING UP COMMAND BUFFER
1766 ;
1767 E9A0 18          CLC          ; SET BUFFER POINTER TO COMMAND FRAME BUFFER
1768 E9A1 A9 3A      LDA  #CBUFLO
1769 E9A3 85 32      STA  BUFRLO      ; AND BUFFER END ADDRESS
1770 E9A5 69 04      ADC  44
1771 E9A7 85 34      STA  BFENLO

```

ERR LINE ADDR B1 B2 B3

SIO (SERIAL INPUT/OUTPUT BUS CONTROLLER)

Page 6

```

1772 E9A9 A9 02          LDA   #CBUFHI
1773 E9AB 85 33          STA   BUFRHI
1774 E9AD 85 35          STA   BFENHI          ; DONE SETTING UP BUFFER POINTER
1775                                     ;
1776 E9AF A9 34          LDA   #NCOMLO          ; LOWER NOT COMMAND LINE
1777 E9B1 8D 03 D3       STA   PBCTL
1778                                     ;
1779 EqB4 20 BA EC       JSR   SENDIN          ; SEND THE COMMAND FRAME TO A SMART DEVICE
1780                                     ;
1781 E9B7 AD 3F 02       LDA   ERRFLG
1782 E9BA D0 03          BNE   BADCOM          ; BRANCH IF AN ERROR RECEIVED
1783                                     ;
1784 E9BC 98              TYA
1785 E9BD D0 07          BNE   ACKREC          ; BRANCH IF ACK RECEIVED
1786                                     ;
1787                                     ;
1788 E9BF C6 36          BADCOM: DEC   CRETRY          ; A NACK OR TIME OUT OCCURED
1789 E9C1 10 85          BPL   COMFRM          ; SO BRANCH IF ANY RETRIES LEFT
1790                                     ;
1791 E9C3 4C 06 EA       JMP   DERR1          ; OTHERWISE. JUMP TO RETURN SECTION
1792                                     ;
1793                                     ;
1794 E9C6 AD 03 03       ACKREC: LDA   DSTATS          ; ACK WAS RECEIVED
1795 E9C9 10 0C          BPL   WATCOM          ; BRANCH TO WAIT FOR COMPLETE .
1796                                     ; IF THERE IS NO DATA TO BE SENT
1797                                     ;
1798                                     ;
1799                                     ;
1800                                     ; SEND A DATA FRAME TO PERIPHERAL
1801                                     ;
1802 E9CB A9 0D          LDA   #CRETRI          ; SET NUMBER OF RETRIES
1803 E9CD 85 36          STA   CRETRY
1804                                     ;
1805 E9CF 20 6A EB       JSR   LDPNTR          ; LOAD BUFFER POINTER WITH DCB INFORMATION
1806                                     ;
1807 E9D2 20 BA EC       JSR   SENDIN          ; GO SEND THE DATA FRAME TO A SMART DEVICE
1808                                     ;

```

```

1809 E9D5 F0 E8          BEQ   BADCOM      ; BRANCH IF BAD
1810                      ;
1811                      ;
1812                      ;
1813                      ; WAIT FOR COMPLETE SIGNAL FROM PERIPHERAL
1814                      ;
1815 E9D7 20 75 EC      WATCOM:   JSR   STTMOT      ; SET DDEVICE TIME OUT VALUES IN Y~X
1816                      ;
1817 E9DA A9 00          LDA   #800
1818 E9DC BD 3F 02      STA   ERRFLG      ; CLEAR ERROR FLAG
1819                      ;
1820 E9DF 20 9B EC      JSR   WAITER      ; SET UP TIMER AND WAIT
1821 E9E2 F0 12          BEQ   DERR       ; BRANCH IF TIME OUT
1822                      ;
1823                      ;
1824                      ; DEVICE DID NOT TIME OUT
1825                      ;
1826 E9E4 2C 03 03      BIT   OSTATS
1827 E9E7 70 07          BVS   MODATA      ; BRANCH IF MORE DATA FOLLOWS
1828                      ;
1829 E9E9 AD 3F 02      LDA   ERRFLG
1830 E9EC D0 18          BNE   DERR1      ; BRANCH IF AN ERROR OCCURRED
1831 E9EE F0 1D          BEQ   RETURN     ; OTHERWISE RETURN
1832                      ;
1833                      ;
1834                      ;
1835                      ;
1836                      ; RECEIVE A DATA FRAME FROM PERIPHERAL
1837                      ;
1838 E9F0 20 6A EB      MODATA:   JSR   LDPNTR      ; LOAD BUFFER POINTER WITH DCB INFORMATION
1839                      ;
1840 E9F3 20 E0 EA      JSR   RECEIV      ; GO RECEIVE A DATA FRAME
1841                      ;
1842 E9F6 AD 3F 02      DERR:     LDA   ERRFLG
1843 E9F9 F0 05          BEQ   NOTERR      ; BRANCH IF NO ERROR PRECEDED DATA
1844                      ;
1845 E9FB AD 19 03      LDA   TSTAT      ; GET TEMP STATUS

```

```

1846 E9FE 85 30          STA  STATUS      ; STORE IN REAL STATUS
1847                      ;
1848                      ;
1849 EA00 A5 30 NOTERR:  LDA  STATUS
1850 EA02 C9 01          CMP  #SUCCE
1851 EA04 F0 07          BEQ  RETURN      ; BRANCH IF COMPLETELY SUCCESSFUL
1852                      ;
1853 EA06 C6 37 DERR1:   DEC  DRETRY
1854 EA08 30 03          BMI  RETURN      ; BRANCH If OUT OF DEVICE RETRIES
1855                      ;
1856 EA0A 4C 74 E9       JMP  COMMND      ; OTHERWISE. ONE MORE TIME
1857                      ;
1858                      ;
1859                      ;
1860                      ;
1861 EA0D 20 5F EC RETURN: JSR  SENDDS      ; DISABLE POKEY INTERRUPTS
1862 EA10 A9 00          LDA  *0
1863 EA12 85 42          STA  CRITIC
1864 EA14 A4 30          LDY  STATUS      ; RETURN STATUS IN Y
1865 EA16 8C 03 03       STY  DSTATS     ; AND THE DCB STATUS WORD
1866 EA19 60            RTS  RETURN
1867                      ;
1868                      ;
1869                      ;
1870                      ;
1871                      ; WAIT SUBROUTINE
1872                      ;
1873                      ; WAITS FOR COMPLETE OR ACK
1874                      ; RETURNS Y=$FF IF SUCCESSFUL Y=$00 IF NOT
1875                      ;
1876 EA1A A9 00 WAIT:    LDA  #$00
1877 EA1C 8D 3F 02       STA  ERRFLG      ; CLEAR ERROR FLAG
1878                      ;
1879 EA1F 18             CLC                ; LOAD BUFFER POINTER WITH ADDRESS
1880 EA20 A9 3E          LDA  #TEMPLO     ; OF TEMPORARY RAM CELL
1881 EA22 85 32          STA  BUFRLO
1882 EA24 69 01          ADC  #1

```



```

1883 EA26 85 34          STA  BFENLO      ; ALSO SET BUFFER END +1 ADDRESS
1884 EA28 A9 02          LDA  #TEMPHI
1885 EA2A 85 33          STA  BUFRHI
1886 EA2C 85 35          STA  BFENHI      ; DONE LOADING POINTER
1887                      ;
1888 EA2E A9 FF          LDA  #$FF
1889 EA30 85 3C          STA  NOCKSM      ; SET NO CHECKSUM FOLLOWS DATA FLAG
1890                      ;
1891 EA32 20 E0 EA        JSR  RECEIV      ; GO RECEIVE A BYTE
1892                      ;
1893 EA35 A0 FF          LDY  #$FF        ; ASSUME SUCCESS
1894 EA37 A5 30          LDA  STATUS
1895 EA39 C9 01          CMP  #SUCCE
1896 EA3B D0 19          BNE  NWOK        ; BRANCH IF IT DID NOT WORK OK
1897                      ;
1898                      ;
1899                      ;
1900                      ;
1901 EA3D AD 3E 02        WOK:  LDA  TEMP      ; MAKE SURE THE BYTE SUCCESSFULLY RECEIVED
1902 EA40 C9 41          CMP  #ACK        ; WAS ACTUALLY AN ACK OR COMPLETE
1903 EA42 F0 21          BEQ  GOOD
1904 EA44 C9 43          CMP  #COMPLT
1905 EA46 F0 1D          BEQ  GOOD
1906                      ;
1907 EA48 C9 45          CMP  #ERROR
1908 EA4A D0 06          BNE  NOTDER      ; BRANCH IF DEVICE DID NOT SEND BACK
1909                      ; A DEVICE ERROR CODE
1910 EA4C A9 90          LDA  #DERROR
1911 EA4E 85 30          STA  STATUS      ; SET DEVICE ERROR STATUS
1912 EA50 D0 04          BNE  NWOK
1913
1914 EA52 A9 8B          NOTDER: LDA  #DNACK    ; OTHERWISE SET NACK STATUS
1915 EA54 85 30          STA  STATUS
1916                      ;
1917 EA56 A5 30          NWOK:  LDA  STATUS
1918 EA58 C9 8A          CMP  #TIMOUT
1919 EA5A F0 07          BEQ  BAD          ; BRANCH IF TIME OUT

```

```

1920      ;
1921      EA5C  A9 FF      LDA    #$FF
1922      EA5E  8D 3F 02   STA    ERRFLG      ; SET SOME ERROR FLAG
1923      EA61  D0 02      BNE    GOOD      ; RETURN WITH OUT SETTING Y = 0
1924      ;
1925      EA63  A0 00      BAD:    LDY    #0
1926      ;
1927      EA65  A5 30      GOOD:   LDA    STATUS
1928      EA67  8D 19 03   STA    TSTAT
1929      EA6A  60          RTS          ; RETURN
1930      ;
1931      ;
1932      ;
1933      ;
1934      ;
1935      ; SEND          SUBROUT    INE
1936      ;
1937      ; SENDS        A BUFFER OF BYTES OUT OVER THE SERIAL BUS
1938      ;
1939      ;
1940      EA6B  A9 01      SEND:    LDA    #SUCCESS      ; ASSUME SUCCESS
1941      EA61) 85 30      STA    STATUS
1942      ;
1943      EA6F  20 F2 EB   JSR     SENDEN      ; ENABLE SENDING
1944      ;
1945      EA72  A0 00      LDY     #0
1946      EA74  84 31      STY     CHKSUM      ; CLEAR CHECK SUM
1947      EA76  84 3B      STY     CHKSNT      ; CHECKSUM SENT FLAG
1948      EA78  84 3A      STY     XMTDON      ;TRANSMISSION DONE FLAG
1949      ;
1950      ;
1951      EA7A  B1 32      LDA     (BUFRLO),Y      ; PUT FIRST BYTE FROM BUFFER
1952      EA7C  SD 0D D2   STA     SEROUT      ; INTO THE SERIAL OUTPUT REGISTER
1953      ;
1954      ;
1955      EA7F  85 31      STA     CHKSUM      ;PUT IT IN CHECKSUM
1956      ;

```

```

1957 EA81 A5 11      NOTDON:   LDA    BRKKEY
j1958 EA83 D0 03      BNE     NTBRKO
1959 EA85 4C A0 ED      JMP     BROKE      ; JUMP IF BREAK KEY PRESSED
1960                      ;
1961 EA88 A5 3A      NTBRKO:   LDA    XMTDON      ; LOOP UNTIL TRANSMISSION IS DONE
1962 EASA F0 F5      BEQ     NOTDON
1963                      ;
1964 EASC 20 5F EC      JSR     SENDDS      ; DISABLE SENDING
1965                      ;
1966 EASF 60          RTS              ; RETURN
1967                      ;
1968                      ;
1969                      ;
1970                      ;
1971                      ;
1972                      ;
1973                      ; OUTPUT DATA NEEDED INTERRUPT SERVICE ROUTINE
1974                      ;
1975 EA90 98          ISRODN:   TYA
1976 EA91 48          PHA              ; SAVE Y REG ON STACK
1977                      ;
1978 EA92 E6 32      INC     BUFRLO      ; INCREMENT BUFFER POINTER
1979 EA94 D0 02      BNE     NOWRPO
1980 EA96 E6 33      INC     BUFRHI
1981                      ;
1982 EA98 A5 32      NOWRPO:   LDA    BUFRLO      ; CHECK IF PAST END OF BUFFER
1983 EA9A C5 34      CMP     BFENLO
1984 EA9C A5 33      LDA     BUFRHI      ; HIGH PART
1985 EA9E E5 35      SBC     BFENHI
1986 EAA0 90 1C      BCC     NOTEND      ; BRANCH IF NOT PAST END OF BUFFER
1987                      ;
1988 EAA2 A5 3B      LDA     CHKSNT
1989 EAA4 D0 0B      BNE     RELONE      ; BRANCH IF CHECKSUM ALREADY SENT
1990                      ;
1991 EAA6 A5 31      LDA     CHKSUM
1992 EAAS SD 0D D2    STA     SEROUT      ; SEND CHECK SUM
1993 EAAB A9 FF      LDA     #SFF

```

```

1994 EAAD 85 3B          STA  CHKSNT      ; SET CHECKSUM SENT FLAG
1995 EAAF D0 09          BNE  CHKDON
1996                      ;
1997 EAB1 A5 10          RELONE: LDA  POKMSK      ; ENABLE TRANSMIT DONE INTERRUPT
1998 EAB3 09 08          ORA   #$08
1999 EAB5 85 10          STA  POKMSK
2000 EAB7 SD 0E D2       STA  IRQEN
2001                      ;
2002 EABA 68             CHKDON: PLA
2003 EABB AS             TAY                ; RESTORE Y REG
2004 EABC 68             PLA                ; RETURN FROM INTERRUPT
2005 EABD 40             RTI
2006                      ;
2007                      ;
2008 EABE A0 00          NOTEND: LDY  #0
2009 EAC0 B1 32          LDA  (BUFRLO),Y    ; PUT NEXT BYTE FROM BUFFER
2010 EAC2 8D 0D D2       STA  SEROUT        ; INTO THE SERIAL OUTPUT REGISTER
2011                      ;
2012 EAC5 18             CLC                ; ADD IT TO CHECKSUM
2013 EAC6 65 31          ADC  CHKSUM
2014 EAC8 69 00          ADC  #0
2015 EACA 85 31          STA  CHKSUM
2016                      ;
2017 EACC 4C BA EA       JMP  CHKDON        ; GO RETURN
2018                      ;
2019                      ;
2020                      ;
2021                      ;
2022                      ;
2023                      ;
2024                      ; TRANSMIT DONE INTERRUPT SERVICE ROUTINE
2025                      ;
2026 EACF A5 3B          ISRTD: LDA  CHKSNT
2027 EAD1 F0 0B          BEQ  FOOEY        ; BRANCH IF CHECKSUM NOT YET SENT
2028                      ;
2029 EAD3 85 3A          STA  XMTDON        ; OTHERWISE SET TRANSMISSION DONE FLAG
2030                      ;

```

```

2031 EAD5 A5 10          LDA   POKMSK      ; DISABLE TRANSMIT DONE INTERRUPT
2032 EAD7 29 F7          AND    #$F7
2033 EAD9 85 10          STA   POKMSK
2034 EADB 8D 0E D2       STA   IRQEN
2035
2036 EADE 68             FOGY:   PLA          ; RETURN FROM INTERRUPT
2037 EADF 40             RTI
2038 ;
2039 ;
2040 ;
2041 ;
2042 ;
2043 ;
2044 ;
2045 ;
2046 ; RECEIVE SUBROUTINE
2047 ;
2048 EAE0 A9 00       RECEIV:  LDA   #0
2049 ;
2050 EAE2 AC 0F 03          LDY   CASFLG
2051 EAE5 D0 02          BNE   NOCLR      ; BRANCH IF CASSETTE
2052 ;
2053 EAE7 85 31          STA   CHKSUM      ; CLEAR CHKSUM
2054 EAE9 85 38       NOCLR:  STA   BUFRFL      ; BUFFER FULL FLAG
2055 EAEB 85 39          STA   RECVDN      ; RECEIVE DONE FLAG
2056 ;
2057 ;
2058 ;
2059 EAED A9 01          LDA   #SUCCES
2060 EAEF 85 30          STA   STATUS      ; SET GOOD STATUS FOR DEFAULT CASE.
2061 EAF1 20 1B EC       JSR   RECVEN      ; DO RECEIVE ENABLE
2062 EAF4 A9 3C          LDA   ANCOMHI      ; COMMAND FRAME HI COMMAND
2063 EAF6 8D 03 D3       STA   PBCTL      ; STORE IN PIA
2064 EAF9 A5 11       CHKTIM:  LDA   BRKKEY
2065 EAFB D0 03          BNE   NTBRL1
2066 EAFD 4C A0 ED       JMP   BROKE      ; JUMP IF BREAK KEY PRESSED
2067 ;

```

```

2068 EB00 AD 17 03 NTBRKI: LDA TIMFLG ; NO.
2069 EB03 F0 05 BEQ TOUT ; IF TIMEOUT~ GO SET ERROR STATUS
2070 EB05 A5 39 LDA RECVDN
2071 EB07 F0 F0 BEQ CHKTIM ; DONE ?
2072 EB09 60 GOBACK: RTS
2073 EB0A A9 8A TOUT: LDA #TIMOUT ; YES.
2074 EB0C 85 30 STA STATUS ; SET TIMEOUT STATUS
2075 ;
2076 ;
2077 ;
2078 ;
2079 ;
2080 ;
2081 EB0E 60 RRETRN: RTS ; RETURN
2082 ;
2083 ;
2084 ;
2085 ;
2086 ;
2087 ;
2088 ;
2089 ; SERIAL INP UT READY INTERRUPT SERVICE ROUTINE
2090 ;
2091 EB0F 98 ISRSIR: TYA
2092 EB10 48 PHA ; SAVE Y REG ON STACK
2093 ;
2094 ;
2095 ;
2096 EB11 AD 0F D2 LDA SKSTAT
2097 EB14 8D 0A D2 STA SKRES ; RESET STATUS REGISTER
2098 ; ***** THIS MAY NOT BE THE PLACE TO DO IT *****
2099 ;
2100 EB17 30 04 BMI NTFRAM ; BRANCH IF NO FRAMING ERROR
2101 ;
2102 EB19 A0 8C LDY #FRMERR
2103 EB1B 84 30 STY STATUS ; SET FRAME ERRORR STATUS
2104 ;

```

```

2105 EB1D 29 20      NTFRAM:   AND    #x20
2106 EB1F D0 04      BNE    NTOVRN      ; BRANCH IF NO OVERRUN ERROR
2107                ;
2108 EB21 A0 8E      LDY    #OVRUN
2109 EB23 84 30      STY    STATUS      ; SET OVERRUN ERROR STATUS
2110                ;
2111 EB25 A5 38      NTOVRN:   LDA    BUFRFL
2112 EB27 F0 13      BEQ    NOTYET      ; BRANCH IF BUFFER WAS NOT YET FILLED
2113                ;
2114 EB29 AD 0D D2      LDA    SERIN      ; THIS INPUT BYTE IS THE CHECKSUM
2115 EB2C C5 31      CMP    CHKSUM
2116 EB2E F0 04      BEQ    SRETRN      ; BRANCH IF CHECKSUMS MATCH
2117                ;
2118 EB30 A0 8F      LDY    *CHKERR
2119 EB32 84 30      STY    STATUS      ; SET CHECKSUM ERROR STATUS
2120                ;
2121 EB34 A9 FF      SRETRN:   LDA    #$FF      ; SET RECEIVE DONE FLAG
2122 EB36 85 39      STA    RECVDN
2123                ;
2124 EB38 68      SUSUAL:   PLA
2125 EB39 A8      TAY      ; RESTORE Y REG
2126 EB3A 68      PLA      ; RETURN FROM INTERRUPT
2127 E83B 40      RTI
2128                ;
2129                ;
2130                ;
2131 EB3C AD 0D D2      NOTYET:   LDA    SERIN
2132 EB3F A0 00      LDY    #0
2133 E841 91 32      STA    (BUFRLO),Y      ; STORE INPUT REGISTER INTO BUFFER
2134                ;
2135 EB43 18      CLC      ; ADD IT TO CHECKSUM
2136 EB44 65 31      ADC    CHKSUM
2137 EB46 69 00      ADC    #0
2138 EB48 85 31      STA    CHKSUM
2139                ;
2140 EB4A E6 32      INC    BUFRLO      ; INCREMENT BUFFER POINTER
2141 E84C D0 02      BNE    NTWRP1

```

```

2142 EB4E E6 33          INC    BUFRHI
2143                      ;
2144 EB50 A5 32          NTRPI:  LDA    BUFRLO
2145 EB52 C5 34          CMP    BFENLO
2146 EB54 A5 33          LDA    BUFRHI
2147 EB56 E5 35          SBC    BFENHI
2148 EB58 90 DE          BCC    SUSUAL      ; BRANCH IF NEW BUFFER ADDRESS IS IN BUFFER L
2149                      ;
2150 EB5A A5 3C          LDA    NOCKSM
2151 EB5C F0 06          BEQ    GOON      ; BRANCH IF A CHECKSUM WILL FOLLOW DATA
2152                      ;
2153 EB5E A9 00          LDA    110
2154 EB60 85 3C          STA    NOCKSM      ; CLEAR NO CHECKSUM FLAG
2155                      ;
2156 EB62 F0 D0          BEQ    SRETRN      ; GO RETURN AND SET RECEIVE DONE FLAG
2157                      ;
2158                      ;
2159 EB64 A9 FF          GOON:   LDA    #$FF
2160 EB66 85 38          STA    BUFRFL      ; SET BUFFER FULL FLAG
2161                      ;
2162 EB68 D0 CE          BNE    SUSUAL      ; GO RETURN
2163                      ;
2164                      ;
2165                      ;
2166                      ;
2167                      ;
2168                      ;
2169                      ;
2170                      ;
2171                      ; LOAD BUFFER POINTER SUBROUTINE
2172                      ;
2173                      ; LOAD BUFFER POINTER WITH DCB BUFFER INFORMATION
2174                      ;
2175 EB6A 18             LDPNTR:  CLC
2176 EB6B AD 04 03      LDA    DBUFLO
2177 EB6E 85 32          STA    BUFRLO
2178 EB70 6D 08 03      ADC    DBYTLO

```



```

2179 EB73 85 34          STA  BFENLO          ; ALSO SET BUFFER END + 1 ADDRESS
2180                      ;
2181 EB75 AD 05 03        LDA  DBUFHI
2182 EB78 85 33          STA  BUFRHI
2183 EB7A 6D 09 03        ADC  DBYTHI
2184 EB7D 85 35          STA  BFENHI
2185                      ;
2186 EB7F 60              RTS                  ; RETURN
2187                      ;
2188                      ;
2189                      ;
2190                      ;
2191                      ;
2192                      ;
2193                      ;
2194                      ;
2195                      ; CASSETTE HANDLING CODE
2196                      ;
2197 EB80 AD 03 03        CASENT: LDA  DSTATS
2198 EB83 10 2E           BPL   CASRED          ; BRANCH IF INPUT FROM CASSETTE
2199                      ;
2200                      ; WRITE      A RECORD
2201                      ;
2202 EB85 A9 CC           LDA  #B600LO          ; SET BAUD RATE TO 600
2203 EB87 8D 04 D2        STA  AUDF3
2204 EB8A A9 05           LDA  #B600HI
2205 EB8C 8D 06 D2        STA  AUDF4
2206                      ;
2207 EBBF 20 F2 EB        JSR   SENDEN          ; TURN ON POKEY MARK TONE
2208                      ;
2209 EB92 A0 0F           LDY   #WSIRG          ; LOAD SHORT WRITE INTER RECORD GAP TIME
2210 EB94 AD 0B 03        LDA  DAUX2
2211 EB97 30 02           BMI   SRTIRO          ; BRANCH IF SHORT GAP IS DESIRED
2212                      ;
2213 EB99 A0 B4           LDY   #WIRGLO          ; SET WRITE IRG TIME
2214 EB9B A2 00          SRTIRO: LDX   #WIRGHI
2215 EB9D 20 B9 ED        JSR   SETVBX

```

```

2216 ;
2217 EBA0 A9 34 LDA #MOTRGO
2218 EBA2 8D 02 D3 STA PACTL ; TURN ON MOTOR
2219 ;
2220 EBA5 AD 17 03 TIMIT: LDA TIMFLG ; LOOP UNTIL DONE
2221 EBAB D0 FB BNE TIMIT
2222 ;
2223 EBAA 20 6A EB JSR LDPNTR ; LOAD BUFFER POINTER WITH DCB INFORMATION
2224 ;
2225 EBAD 20 6B EA JSR SEND ; SEND A BUFFER
2226 ;
2227 EBB0 4C DF EB JMP CRETRN ; GO RETURN
2228 ;
2229 ;
2230 ;
2231 ; RECEIVE A RECORD
2232 ;
2233 EBB3 A9 FF CASRED: LDA #$FF
2234 EBB5 8D 0F 03 STA CASFLG ; SET SET CASSETTE FLAG
2235 ;
2236 EBBS A0 0A LDY #RSIRG ;LOAD SHORT READ INTER RECORD GAP TIME
2237 EBBA AD 0B 03 LDA DAUX2
2238 EBBD 30 02 BMI SRTIRI ; BRANCH IF SHORT GAP IS DESIRED
2239 ;
2240 EBBF A0 78 LDY #RIRGLO ; SET TIME OUT FOR READ IRG
2241 EBC1 A2 00 SRTIRI: LDX #RIRGHI
2242 EBC3 20 B9 ED JSR SETVBX
2243 ;
2244 EBC6 A9 34 LDA #MOTRGO
2245 EBCB 8D 02 D3 STA PACTL ; TURN ON MOTOR
2246 ;
2247 EBCB AD 17 03 TIMIT1: LDA TIMFLG ; LOOP UNTIL DONE
2248 EBCE D0 FB BNE TIMIT1
2249 ;
2250 EBD0 20 6A EB JSR LDPNTR ; LOAD BUFFER POINTER WITH DCB INFORMATION
2251 ;
2252 EBD3 20 75 EC JSR STTMOT ; SET DEVICE TIME OUT IN Y,X

```

```

2253 EBD6 20 B9 ED      JSR   SETVBX
2254                      ;
2255 EBD9 20 10 ED      JSR   BEGIN      ; SET INITIAL BAUD RATE
2256                      ;
2257 EBD6 20 B9 ED      JSR   SETVBX
2258                      ;
2259 EBD6 20 B9 ED      JSR   SETVBX
2260 EBD9 20 10 ED      JSR   BEGIN      ; SET INITIAL BAUD RATE
2261                      ;
2262 EBD6 20 B9 ED      JSR   SETVBX
2263 EBD9 20 10 ED      JSR   BEGIN      ; SET INITIAL BAUD RATE
2264                      ;
2265 EBD6 20 B9 ED      JSR   SETVBX
2266 EBD9 20 10 ED      JSR   BEGIN      ; SET INITIAL BAUD RATE
2267                      ;
2268 EBD6 20 B9 ED      JSR   SETVBX
2269 EBD9 20 10 ED      JSR   BEGIN      ; SET INITIAL BAUD RATE
2270                      ;
2271 EBEC A9 00      JTIMER: LDA   #$00
2272 00EB      JTADRH   =      JTIMER/256 ; HI BYTE OF JUMP TIMER ROUTINE ADDR
2273 00EC      JTADRL   =      (-256)*JTA DRH+JTIMER
2274 EBEE BD 17 03      STA   TIMFLG ; SET TIME OUT FLAG
2275 EBF1 60      RTS
2276                      ;
2277                      ;
2278                      ;
2279                      ;
2280                      ;
2281                      ;
2282                      ; SEND   ENABLE      SUBROUTINE
2283                      ;
2284 EBF2 A9 07      SENDEN: LDA   #$07 ; MASK OFF PREVIOUS SERIAL BUS CONTROL BITS
2285 EBF4 2D 32 02      AND   SSKCTL
2286 EBF7 09 20      ORA   #$20 ; SET TRANSMIT MODE
2287                      ;
2288 EBF9 AC 00 03      LDY   DDEVIC

```

```

2289 EBFC C0 60          CPY    #CASET
2290 EBFE D0 0C          BNE    NOTCAS      ; BRANCH IF NOT CASSETTE
2291                      ;
2292 EC00 09 08          ORA    #$08      ; SET THE FSK OUTPUT BIT
2293                      ;
2294 EC02 A0 07          LDY    #LOTONE    ; SET FSK TONE FREQUENCIES
2295 EC04 8C 02 D2      STY    AUDF2
2296 EC07 A0 05          LDY    #HITONE
2297 EC09 SC 00 D2      STY    AUDF1
2298                      ;
2299 EC0C 8D 32 02      NOTCAS: STA    SSKCTL      ; STORE NEW VALUE TO SYSTEM MASK
2300 EC0F 8D 0F D2      STA    SKCTL      ; STORE TO ACTUAL REGISTER
2301
2302 EC12 A9 C7          LDA    #$C7      ; MASK OFF PREVIOUS SERIAL BUS INTERRUPT BITS
2303 EC14 25 10          AND    POKMSK
2304 EC16 09 10          ORA    #$10      ; ENABLE OUTPUT DATA NEEDED INTERRUPT
2305                      ;
2306                      ;
2307 EC18 4C 31 EC          JMP    CONTIN      ; GO CONTINUE IN RECEIVE ENABLE SUBROUTINE
2308                      ;
2309                      ;
2310                      ;
2311                      ;
2312                      ;
2313                      ;
2314                      ;
2315                      ;
2316                      ;
2317                      ;
2318                      ; RECEIVE ENABLE SUBROUTINE
2319                      ;
2320 EC1B A9 07      RECVEN: LDA    #$07      ; MASK OFF PREVIOUS SERIAL BUS CONTROL BITS
2321 EC1D 2D 32 02      AND    SSKCTL
2322 EC20 09 10          ORA    #$10      ; SET RECEIVE MODE ASYNCH.
2323 EC22 8D 32 02      STA    SSKCTL      ; STORE NEW VALUE TO SYSTEM MASK
2324 EC25 8D 0F D2      STA    SKCTL      ; STORE TO ACTUAL REGISTER
2325                      ;

```

```

2326 EC28 8D 0A D2          STA    SKRES          ; RESET SERIAL PORT/KEYBOARD STATUS REGISTER
2327                          ;
2328 EC2B A9 C7              LDA    #$C7          ; MASK OFF PREVIOUS SERIAL BUS INTERRUPT BITS
2329 EC2D 25 10              AND     POKMSK
2330 EC2F 09 20              ORA     #$20          ; ENABLE RECEIVE INTERRUPT
2331 EC31 85 10              CONTIN: STA    POKMSK      ; STORE NEW VALUE TO SYSTEM MASK
2332 EC33 BD 0E D2          STA    IRQEN          ; STORE TO ACTUAL REGISTER
2333                          ;
2334                          ;
2335 EC36 A9 28              LDA    #$28          ; CLOCK CH.3 WITH 1.79 MHZ
2336 EC38 8D 08 D2          STA    AUDCTL        ; CLOCK CH.4 WITH CH. 3
2337                          ;
2338 EC3B A2 06              LDX     #6           ; SET PURE TONES. NO VOLUME
2339 EC3D A9 A8              LDA    #$A8
2340 EC3F A4 41              LDY     SOUNDNR      ; TEST QUIET I/O FLAG
2341 EC41 D0 02              BNE     NOISE1        ; NE IS NORMAL (NOISY)
2342 EC43 A9 A0              LDA    #$A0
2343 EC45 9D 01 D2          NOISE1: STA    AUDC1,X
2344 EC48 CA                DEX
2345 EC49 CA                DEX
2346 EC4A 10 F9              BPL     NOISE1
2347                          ;
2348 EC4C A9 A0              LDA    #$A0
2349 EC4E 8D 05 D2          STA    AUDC3          ; TURN OFF SOUND ON CHANNEL 3
2350 EC51 AC 00 03          LDY     DDEVIC
2351 EC54 C0 60              CPY     #CASET
2352 EC56 F0 06              BEQ     CAS31        ; BRANCH IF CASSETTE IS DESIRED
2353 EC58 8D 01 D2          STA    AUDC1          ; OTHERWISE TURN OFF CHANNELS 1 AND 2
2354 EC5B 8D 03 D2          STA    AUDC2
2355                          ;
2356                          ;
2357 EC5E 60              CAS31:  RTS            ; RETURN
2358
2359
2360
2361
2362

```

```

2363      ;
2364      ;
2365      ;
2366      ;
2367      ;
2368      ; DISABLE SEND AND DISABLE RECEIVE SUBROUTINES
2369      ;
2370      EC5F  EA      SENDDS:      NOP
2371      EC60  A9 C7      RECVDS:      LDA    #$C7      ; MASK OFF SERIAL BUS INTERRUPTS
2372      EC62  25 10      AND    POKMSK
2373      EC64  85 10      STA    POKMSK      ; STORE NEW VALUE TO SYSTEM MASK
2374      EC66  8D 0E D2      STA    IRQEN      ; STORE TO ACTUAL REGISTER
2375      ;
2376      EC69  A2 06      LDX    #6
2377      EC6B  A9 00      LDA    #0
2378      EC6D  9D 01 D2      ZERIT:      STA    AUDC1,X
2379      EC70  CA      DEX
2380      EC71  CA      DEX
2381      EC72  10 F9      BPL    ZERIT      ; TURN OFF AUDIO VOLUME
2382      ;
2383      EC74  60      RTS      ;RETURN
2384      ;
2385      ;
2386      ;
2387      ;
2388      ;
2389      ;
2390      ;
2391      ;
2392      ;
2393      ;
2394      ; SET DDEVICE      TIME OUT VALUES IN Y,X SUBROUTINE
2395      ;
2396      EC75  AD 06 03      STTMOT:      LDA    DTIMLO      ; GET DEVICE TIME OUT IN 1 SECOND INCR
2397      EC78  6A      ROR    A      ; PUT 6 HI BITS IN X. LO 2 BITS IN Y
2398      EC79  6A      ROR    A
2399      EC7A  A8      TAY      ; TEMP SAVE

```

```

2400 EC7B 29 3F      AND    #$3F      ; MASK OFF 2 HI BITS
2401 EC7D AA        TAX              ; THIS IS HI BYTE OF TIME OUT
2402                ;
2403 EC7E 98        TYA              ; RESTORE
2404 EC7F 6A        ROR    A
2405 EC80 29 C0      AND    #$C0      ; MASK OFF ALL BUT 2 HI BITS
2406 EC82 A8        TAY              ; THIS IS LO BYTE OF TIME OUT
2407                ;
2408 EC83 60        RTS
2409                ;
2410                ;
2411                ;
2412                ;
2413                ;
2414                ;
2415                ;
2416                ;
2417                ;
2418                ;
2419 EC84 0F EB      INTTBL: .WORD ISRSIR      ; SERIAL INPUT READY
2420 EC86 90 EA      .WORD ISRODN      ; OUTPUT DATA NEEDED
2421 EC88 CF EA      .WORD ISRTD      ; TRANSMISSION DONE
2422                ;
2423 00EB            SIRHI      =    ISRSIR/256 ; SERIAL INPUT READY    ISR ADDRESS
2424 000F            SIRLO      =    (-256)*SIRHI+ISRSIR
2425 00EA            ODNHI      =    ISRODN/256 ; OUTPUT DATA NEEDED    ISR ADDRESS
2426 0090            ODNLO      =    (-256)*ODNHI+ISRODN
2427 00EA            TDHI       =    ISRTD/256 ; TRANSMISSION DONE I    SR ADDRESS
2428 00CF            TDLO       =    (-256)*TDHI+ISRTD
2429                ;
2430                ;
2431                ;
2432                ;
2433                ; SEND A    DATA  FRAME TO AN INTELLIGENT PERIPHERAL  SUBROUTINE
2434                ;
2435                ;
2436 ECBA A2 01      SENDIN:    LDX    4301

```

```

2437 ECBC A0 FF      DELAY0:    LDY    #$FF
2438 ECBE 88        DELAY1:    DEY
2439 ECBF D0 FD            BNE    DELAY1
2440 EC91 CA            DEX
2441 EC92 D0 FS            BNE    DELAY0
2442                ;
2443 EC94 20 6B EA      JSR    SEND        ; GO SEND THE DATA FRAME
2444                ;
2445 EC97 A0 02            LDY    #CTIMLO    ; SET ACK TIME OUT
2446 EC99 A2 00            LDX    #CTIMHI
2447 EC9B 20 B9 ED      WAITER:  JSR    SETVBX
2448                ;
2449 EC9E 20 1A EA      JSR    WAIT    ; WAIT FOR ACK
2450                ;
2451 ECA1 98            TYA                ; IF Y=0, A TIME OUT OR NACK OCCURED
2452                ;
2453 ECA2 60            RTS                ; RETURN
2454                ;
2455                ;
2456                ;
2457                ;
2458                ;
2459                ;
2460                ;
2461                ;
2462                ;
2463                ;
2464                ;
2465                ; COMPUTE VALUE FOR POKEY FREQ REGS FOR THE BAUD RATE AS
2466                ; MEASURED BY AN INTERVAL OF THE 'VCOUNT' TIMER.
2467                ;
2468 ECA3 8D 10 03      COMPUT:    STA    TIMER2
2469 ECA6 8C 11 03            STY    TIMER2+1    ; SAVE FINAL TIMER VALUE
2470 ECA9 20 04 ED      JSR    ADJUST    ; ADJUST VCOUNT VALUE
2471 ECAC 8D 10 03            STA    TIMER2    ; SAVE ADJUSTED VALUE
2472 ECAF AD 0C 03            LDA    TIMER1
2473 ECB2 20 04 ED      JSR    ADJUST    ; ADJUST

```



```

2474 ECB5 8D 0C 03          STA  TIMER1          ; SAVE ADJUSTED TIMER1 VALUE
2475 ECBB AD 10 03          LDA  TIMER2
2476 ECBB 38                SEC
2477 ECBC ED 0C 03          SBC  TIMER1
2478 ECBF 8D 12 03          STA  TEMP1          ; FIND VCOUNT DIFFERENCE
2479 ECC2 AD 11 03          LDA  TIMER2+1
2480 ECC5 38                SEC
2481 ECC6 ED 0D 03          SBC  TIMER1+1
2482 ECC9 AS                TAY          ; FIND VBLANK COUNT DIFFERENCE
2483                      .IF  PALFLG
2484                      LDA  #-$9C
2485          HITIMR:         CLC
2486                      ADC  #-$9C
2487                      .ENDIF
2488                      .IF  PALFLG-1
2489          ECCA  A9 7D          LDA  #-$83
2490          ECCC 18          HITIMR:         CLC
2491          ECCD 69      83          ADC  #-$83          ; ACCUMULATE MULTIPLICATION
2492                      .ENDIF
2493          ECCF 88          DEY
2494          ECD0 10 FA          BPL  HITIMR          ; DONE?
2495          EC D2 18          CLC
2496          ECD3 6D 12 03          ADC  TEMPI          ; TOTAL VCOUNT DIFFERENCE
2497          ECD6 A8          FINDX:         TAY          ; SAVE ACCUM
2498          ECD7 4A          LSR  A
2499          ECDB 4A          LSR  A
2500          ECD9 4A          LSR  A
2501          ECDA 0A          ASL  A
2502          ECDB 38          SEC
2503          EDCD E9 16          SBC  #22          ; ADJUST TABLE INDEX
2504          ECDE AA          TAX          ; DIVIDE INTERVAL BY 4 TO GET TABLE INDEX
2505          ECDF 98          TYA          ; RESTORE PCCUM
2506          ECE0 29 07          AND  #7
2507          ECE2 A8          TAY          ; PULL OFF 3 LO BITS OF INTERVAL
2508          ECE3 A9 F5          LDA  #-11
2509          ECE5 18          DOINTP:        CLC
2510          ECE6 69 0B          ADC  #11          ; ACCUMULATE INTERPOLATION CONSTANT

```

```

2511 ECE8 88          DEY
2512 ECE9 10 FA      BPL   DOINTP      ; INTERPOLATION CONSTANT COMPUTATION DONE?
2513                ;
2514 ECEB A0 00      ENINTP: LDY   #0
2515 ECED 8C 0E 03    STY   ADDCOR      ; CLEAR ADDITION CORRECTION FLAG
2516 ECF0 38          SEC
2517 ECF1 E9 07      SBC   #7          ; ADJUST INTERPOLATION CONSTANT
2518 ECF3 10 03      BPL   PLUS
2519 ECF5 CE 0E 03    DEC   ADDCOR
2520 ECF8 18          PLUS:  CLC
2521 ECF9 7D D0 ED    ADC   POKTAB,X    ; ADD CONSTANT TO LO BYTE TABLE VALUE
2522 ECFC A8          TAY              ; LO BYTE POKEY FREQ VALUE
2523 ECFD AD 0E 03    LDA   ADDCOR
2524 ED00 7D D1 ED    ADC   POKTAB+1,X  ; ADD CARRY TO HI BYTE TABLE VALUE
2525                ; HI BYTE POKEY FREQ VALUE
2526 ED03 60          RTS
2527                ;
2528                ;
2529
2530                ; ROUTINE TO ADJUST VCOUNT VALUE
2531                ;
2532 ED04 C9 7C      ADJUST:  CMP   #$7C
2533 ED06 30 04      BMI   ADJ1      ; LARGER THAN '7C' ?
2534 ED0B 38          SEC          ; YES,
2535 ED09 E9 7C      SBC   #$7C
2536 ED0B 60          RTS
2537 ED0C 18          ADJ1:  CLC
2538                .IF   PALFLG
2539                ADC   #$20
2540                .ENDIF
2541                .IF   PALFLG-1
2542 ED0D 69 07      ADC   #$7
2543                .ENDIF
2544 ED0F 60          RTS
2545                ;
2546                ;
2547                ;

```

```

2548 ;
2549 ;
2550 ;
2551 ;
2552 ; INITIAL BAUD RATE MEASUREMENT -- USED TO SET THE
2553 ; BAUD RATE AT THE START OF A RECORD.
2554 ;
2555 ; IT IS ASSUMED THAT THE FIRST TWO BYTES OF EVERY
2556 ; RECORD ARE 'AA' HEX.
2557 ;
2558 ED10 A5 11 BEGIN: LDA BRKKEY
2559 ED12 D0 03 BNE NTBRK2
2560 ED14 4C A0 ED JMP BROKE ; JUMP IF BREAK KEY PRESSED
2561 ;
2562 ED17 78 NTBRK2: SEI
2563 ;
2564 ED18 AD 17 03 LDA TIMFLG
2565 ED1B D0 02 BNE OKTIM1 ; BRANCH IF NOT TIMED OUT
2566 ED1D F0 25 BEQ TOUT1 ; BRANCH IF TIME OUT
2567 ;
2568 ED1F AD 0F D2 OKTIM1: LDA SKSTAT
2569 ED22 29 10 AND #$10 ; READ SERIAL PORT
2570 ED24 D0 EA BNE BEGIN ; START BIT?
2571 ED26 SD 16 03 STA SAVIO ; SAVE SER. DATA IN
2572 ED29 AE 0B D4 LDX VCOUNT ; READ VERTICAL LINE COUNTER
2573 ED2C A4 14 LDY RTCLOK+2 ; READ LO BYTE OF VBLANK CLOCK
2574 ED2E 8E 0C 03 STX TIMERI
2575 ED31 8C 0D 03 STY TIMER1+1 ; SAVE INITIAL TIMER VALUE
2576 ;
2577 ED34 A2 01 LDX #1 ; SET MODE FLAG
2578 ED36 8E 15 03 STX TEMP3
2579 ED39 A0 0A LDY #10 ; SET BIT COUNTER FOR 10 BITS
2580 ED3B A5 11 COUNT: LDA BRKKEY
2581 ED3D F0 61 BEQ BROKE ; BRANCH IF BREAK KEY PRESSED
2582 ;
2583 ED3F AD 17 03 LDA TIMFLG

```

```

2584 ED42 D0 04      BNE    OKTIMR      ; BRANCH IF NOT TIMED OUT
2585 ED44 58          TOUT1:    CLI          ;
2586 ED45 4C 0A EB    JMP     TOUT      ; BRANCH IF TIME OUT
2587
2588 ED48 AD 0F D2    OKTIMR:    LDA     SKSTAT
2589 ED4B 29 10        AND     #$10      ; READ SERIAL PORT
2590 ED4D CD 16 03    CMP     SAVIO      ; DATA IN CHANGED YET?
2591 ED50 F0 E9        BEQ     COUNT
2592 ED52 8D 16 03    STA     SAVIO      ; YES SAVE SER. DATA IN
2593 ED55 88          DEY          ; DECR. BIT COUNTER
2594 ED56 D0 E3      BNE     COUNT      ; DONE?
2595
2596 ED58 CE 15 03    ;
2597 ED5B 30 12        DEC     TEMP3      ; YES
2598 ED5D AD 0B D4    BMI     GOREAD      ; DONE WITH BOTH MODES?
2599 ED60 A4 14        LDA     VCOUNT
2600 ED62 20 A3 EC    LDY     RTCLOK+2    ; READ TIMER LO & HI BYTES
2601 ED65 8C EE 02    JSR     COMPUT      ; NO. COMPUTE BAUD RATE
2602 ED68 8D EF 02    STY     CBAUDL
2603 ED6B A0 09        STA     CBAUDH      ; SET BAUD RATE INTO RAM CELLS
2604 ED6D D0 CC      LDY     #9          ; SET BIT COUNTER FOR 9 BITS
2605
2606 ED6F AD EE 02    ;
2607 ED72 8D 04 D2    GOREAD:    LDA     CBAUDL
2608 ED75 AD EF 02    STA     AUDF3
2609 ED78 SD 06 D2    LDA     CBAUDH
2610 ED7B A9 00        STA     AUDF4      ; SET POKEY FREQ BEGS FOR BAUD RATE
2611 ED7D 8D 0F D2    LDA     #0
2612 ED80 AD 32 02    STA     SKSTAT
2613 ED83 8D 0F D2    LDA     SSKCTL
2614 ED86 A9 55        STA     SKSTAT      ; INIT. POKEY SERIAL PORT
2615 EDB8 91 32        LDA     #$55
2616 ED8A C8          STA     (BUFRLO),Y   ; STORE '$55' AS FIRST RCV. BUFFER
2617 ED8B 91 32        INY
2618 ED8D A9 AA        STA     (BUFRLO),Y
2619 EDBF 85 31        LDA     #$AA
2620 ED91 18          STA     CHKSUM      ; STORE CHECKSUM FOR 2 BYTES OF '$AA'
                CLC

```

```

2621 ED92 A5 32          LDA  BUFRLO
2622 ED94 69 02          ADC   #2
2623 ED96 85 32          STA  BUFRLO
2624 ED98 A5 33          LDA  BUFRHI
2625 ED9A 69 00          ADC   #0
2626 ED9C 85 33          STA  BUFRHI      ; INCR. BUFFER POINTER BY 1
2627 ED9E 58             CLI
2628 ED9F 60             RTS
2629                      ;
2630                      ;
2631                      ;
2632 EDA0 20 5F EC      BROKE: JSR  SENDDS      ; BREAK KEY WAS PRESSED. SO PREPARE
2633 EDA3 A9 3C          LDA  #MOTRST      ; TO RETURN
2634 EDA5 8D 02 D3      STA  PACTL ; TURN OFF MOTOR
2635 EDAB 8D 03 D3      STA  PBCTL ; RAISE NOT COMMAND LINE
2636                      ;
2637 EDAB A9 80          LDA  #BRKABT
2638 EDAD 85 30          STA  STATUS      ; STORE BREAK ABORT STATUS CODE
2639                      ;
2640 EDAF AE 18 03      LDX  STACKP
2641 EDB2 9A             TXS              ; RESTORE STACK POINTER
2642                      ;
2643 EDB3 C6 11          DEC  BRKKEY      ; SET BREAK KEY FLAG TO NONZERO
2644 EDB5 58             CLI              ; ALLOW IRQ'S
2645                      ;
2646 ED86 4C 0D EA      JMP  RETURN      ; GO RETURN
2647                      ;
2648                      ;
2649                      ;
2650                      ;
2651                      ;
2652 EDB9 A9 EC      SETVBX: LDA  #JTADRL      ; STORE TIME OUT ROUTINE ADDRESS
2653 EDBB 8D 26 02      STA  CDTMA1
2654 EDBE A9 EB          LDA  #JTADRH
2655 EDC0 8D 27 02      STA  CDTMA1+1
2656

```

```

2657 EDC3 A9 01          LDA    #1          ; SET FOR TIMER 1
2658
2659 EDC5 78             SEI              ; THE SETVBL ROUTINE NEEDS T HIS TO CUT SHORT
2660 EDC6 20 5C E4       JSR    SETVBV      ; ANY VBLANKS THAT OCCUR
2661 EDC9 A9 01          LDA    #1          ; SET FOR TIMER 1
2662 EDCB 8D 17 03       STA    TIMFLG      ; SET FLAG TO NOT TIMED OUT
2663 EDCE 58             CLI
2664 EDCF 60             RTS
2665 ;
2666 ;
2667 ;
2668 ;
2669 ;
2670 ;
2671 ;
2672 ; 'VCOUNT' INTERVAL TIMER MEASUREMENT -- TO -- POKEY FR      EQ REG VALUE
2673 ; CONVERSION TABLE
2674 ;
2675 ;
2676 ; THE VALUES STORED IN THE TABLE ARE 'AUDF+7 '.
2677 ;
2678 ; THE FOLLOWING FORMULAS WERE USED TO DETERMINE THE TABLE VALUES:
2679 ;
2680 ; F OUT= F IN/(2*(AUDF+M)) ~ WHERE F IN=1 .78979 MHZ. & M=7
2681 ;
2682 ; FROM THIS WAS DERIVED THE FORMULA USED TO COMP      UTE THE
2683 ; TABLE VALUES BASED ON A MEASUREMENT OF THE PER      IOD BY
2684 ; AN INTERVAL OF THE 'VCOUNT' TIMER.
2685 ;
2686 ; AUDF+7=(11.365167)*T OUT WHERE T OUT=# OF COUNTS
2687 ; (127 USEC.RESOLUTION) OF 'VCOUNT' FOR 1
2688 ; CHARACTER TIME (10 BIT TIMES).
2689 ;
2690 ;
2691 ;
2692 ;
2693 ;

```

```

2694      ;          AUDF+7      BAUD RATE      VCOUNT INTERVAL
2695      ;      .WORD      $27C      ; 1407      56
2696      ;      .WORD      $2D7      ; 1231      64
2697      ;      .WORD      $332      ; 1094      72
2698      ;      .WORD      $38D      ; 985      80
2699      EDD0      E8 03      POKTAB: .WORD      $3E8      ; 895      88
2700      EDD2      43 04      .WORD      $443      ; 820      96
2701      EDD4      9E 04      .WORD      $49E      ; 757      104
2702      EDD6      F9 04      .WORD      $4F9      ; 703      112
2703      Eddb      54 05      .WORD      $554      ; 656      120
2704      EDDA      AF 05      .WORD      $5AF      ; 615      128
2705      EDDC      0A 06      .WORD      $60A      ; 579      136
2706      EDDE      65 06      .WORD      $665      ; 547      144
2707      EDE0      C0 06      .WORD      $6C0      ; 518      152
2708      EDE2      IA 07      .WORD      $71A      ; 492      160
2709      EDE4      75 07      .WORD      $775      ; 469      168
2710      EDE6      D0 07      .WORD      $7D0      ; 447      176
2711      .WORD      $82B      ; 428      184
2712      .WORD      $886      ; 410      192
2713      .WORD      $8E1      ; 394      200
2714      .WORD      $93C      ; 379      208
2715      .WORD      $997      ; 365      216
2716      .WORD      $9F2      ; 352      224
2717      .WORD      $A4D      ; 339      232
2718      .WORD      $AA8      ; 328      240
2719      .WORD      $B03      ; 318      248
2720      ;
2721      ;
2722      ;
2723      ;
2724      ; *****
2725      EDEB      CRNTP3 =*
2726      *=$14
2727      0014      02      SIOSPR: .BYTE      DSKORG-CRNTP3      ; ^GSIOL IS TOO LONG
2728      ;
2729      .TITLE 'DISK ***** DISKP.SRC ***** 3/9/79 ***** 4:00:00 P. M.'
2730      ;

```

```

2731 ;
2732 ;
2733 ;
2734 ;
2735 ;
2736 0002 STATVH = DVSTAT/256
2737 00EA STATVL = (-256)*STATVH+DVSTAT ;STATUS POINTER
2738 ;
2739 ;
2740 ;
2741 ;
2742 ; CONSTANT EQUATES
2743 ;
2744 0031 DISKID = $31 ; SERIAL BUS DISK I. D.
2745 0050 PUTSEC = $50 ; DISK PUT SECTOR DCB COMMAND
2746 ; READ = 452 ; DISK GET SECTOR DCB COMMAND
2747 ; WRITE = 457 ; DISK PUT SECTOR WITH READ CHECK DCB COMMAND
2748 0053 STATC = $53 ; DISK STATUS DCB COMMAND
2749 0021 FOMAT = $21 ; DISK FORMAT DCB COMMAND !!!!! *****
2750 0000 NODAT = 0 ; SIO COMMAND FOR "NO DATA" OPERATION
2751 0040 GETDAT = $40 ; SIO COMMAND FOR "DATA FROM DEVICE"
2752 0080 PUTDAT = $80 ; SIO COMMAND FOR "DATA TO DEVICE"
2753 ;
2754 ;
2755 ; VECTORS
2756 ;
2757 ; *= $E450
2758 ;
2759 E450 4C EA ED JMP DINIT ; DISK INIT. VECTOR
2760 E453 4C F0 ED JMP DSKIF ; DISK INTERFACE ENTRY POINT
2761 ;
2762 ;
2763 ;
2764 ;
2765 ;
2766 ;
2767 ; CONSTANTS

```



```

2768 ;
2769         *=DSKORG
2770 ;
2771 ;
2772 ;
2773 ;
2774 ;
2775 ;
2776 ;
2777 ;
2778 ;
2779 ; *****
2780 ; DISK INTERFACE ROUTINE STARTS HERE
2781 ; *****
2782 ;
2783 ;
2784 ;
2785 ;
2786 ; DISK INTERFACE INITIALIZATION ROUTINE
2787 ;
2788 EDEA A9 A0 DINIT: LDA #160
2789 EDEC 8D 46 02 STA DSKTIM ; SET INITIAL DISK TIMEOUT TO 160 SEC
2790 EDEF 60 RTS
2791 ;
2792 ;
2793 ;
2794 ; DISK INTERFACE ENTRY POINT
2795 ;
2796 EDF0 A9 31 DSKIF: LDA #DISKID
2797 EDF2 8D 00 03 STA DDEVIC ; SET SERIAL BUS I.D IN DCB
2798 EDF5 AD 46 02 LDA DSKTIM
2799 EDFB AE 02 03 LDX DCOMND
2800 EDFB E0 21 CPX #FOMAT ; IS COMMAND A FORMAT COMMAND?
2801 EDFD F0 02 BEQ PUTDTO
2802 EDFF A9 07 LDA #7 ; NO, SET TIMEOUT TO 7 SECS.
2803 EE01 8D 06 03 PUTDTO: STA DTIMLO ; PUT DISK TIMEOUT IN DCB
2804 EE04 A2 40 LDX #GETDAT ; SET "GET DATA" COMMAND FOR SIO

```

```

2805 EE06 A0 80          LDY    #$80          ; SET BYTE COUNT TO 128
2806 EE08 AD 02 03      LDA    DCOMND        ; READ COMMAND IN DCB
2807 EE0B C9 57          CMP    #WRITE       ; IS COMMAND A "PUT SECTOR" COMMAND?
2808 EE0D D0 02          BNE    CKSTC
2809 EE0F A2 80          LDX    #PUTDAT      ; YES. SET "PUT DATA" COMMAND FOR SIO
2810 EE11 C9 53          CKSTC:  CMP    #STATC    ; IS COMMAND A STATUS COMMAND?
2811 EE13 D0 0C          BNE    PUTCNT
2812 EE15 A9 EA          LDA    #STATVL
2813 EE17 8D 04 03      STA    DBUFLO
2814 EE1A A9 02          LDA    #STATVH
2815 EE1C 8D 05 03      STA    DBUFHI        ; SET BUFFER ADDR TO GLOBAL STATUS BUFFER
2816 EE1F A0 04          LDY    #4          ; YES. SET BYTE COUNT TO 4
2817 EE21 8E 03 03      PUTCNT:  STX    DSTATS    ; PUT STATUS COMMAND FOR SIO IN DCB
2818 EE24 8C 08 03      STY    DBYTLO
2819 EE27 A9 00          LDA    #0
2820 EE29 8D 09 03      STA    DBYTHI        ; PUT BYTE COUNT IN DCB
2821 EE2C 20 59 E4      JSR    SIOV          ; CALL SERIAL I/O.
2822 EE2F 10 01          BPL    GOODST        ; NO ERROR
2823 EE31 60             RTS                ; NO, GO BACK
2824 EE32 AD 02 03      GOODST:  LDA    DCOMND        ; READ THE COMMAND
2825 EE35 C9 53          CMP    #STATC        ; WAS IT A STATUS COMMAND?
2826 EE37 D0 0A          BNE    PUTBC
2827 EE39 20 6D EE      JSR    PUTADR        ; PUT BUFFER ADDR IN TEMP REG.
2828 EE3C A0 02          LDY    #2
2829 EE3E B1 15          LDA    (BUFADR),Y    ; READ DISK TIMEOUT VALUE BYTE OF STATUS
2830 EE40 8D 46 02      STA    DSKTIM        ; PUT IT IN DISK TIMEOUT REG.
2831 EE43 AD 02 03      PUTBC:   LDA    DCOMND
2832 EE46 C9 21          CMP    #FOMAT        ; WAS COMMAND A FORMAT COMMAND?
2833 EE48 D0 1F          BNE    ENDDIF
2834 EE4A 20 6D EE      FMTD:   JSR    PUTADR        ; YES PUT BUFFER ADDR INTO TEMP REG
2835 EE4D A0 FE          LDY    #$FE        ; SET BUFFER POINTER
2836 EE4F CS            TWICE:   INY
2837 EE50 CS            INY          ; INCR BUFFER POINTER BY 2
2838 EE51 B1 15          RDBAD:  LDA    (BUFADR),Y    ; READ LO BYTE BAD SECTOR DATA
2839 EE53 C9 FF          CMP    #$FF
2840 EE55 D0 F8          BNE    TWICE        ; IS IT "FF" ?
2841 EE57 CS            INY          ; YES,

```

```

2842 EE58 B1 15 LDA (BUFADR),Y ; READ HI BYTE BAD SECTOR DATA
2843 EE5A CS INY
2844 EE5B C9 FF CMP #$FF
2845 EE51) D0 F2 BNE RDBAD ; IS IT "FF" ?
2846 EE5F 88 DEY
2847 EE60 88 DEY ; YES,
2848 EE61 8C 08 03 STY DBYTLO ; PUT BAD SECTOR BYTE COUNT INTO DCB
2849 EE64 A9 00 LDA #0
2850 EE66 8D 09 03 STA DBYTHI
2851 EE69 AC 03 03 ENDDIF: LDY DSTATS
2852 EE6C 60 RTS
2853 ;
2854 ;
2855 ;
2856 ;
2857 ; S U B R O U T I N E S
2858 ;
2859 ;
2860 ; PUT BUFFER ADDR FROM DCB INTO TEMP REG
2861 ;
2862 EE6D AD 04 03 PUTADR: LDA DBUFLO
2863 EE70 85 15 STA BUFADR
2864 EE72 AD 05 03 LDA DBUFHI
2865 EE75 85 16 STA BUFADR+1 ; PUT BUFFER ADDR IN TEMP REG
2866 EE77 60 RTS
2867 ; *****
2868 ;
2869 ;
2870 ; SPARE BYTE OR MODULE TOO LONG FLAG
2871 ;
2872 ;
2872 EE78 CRNTP4 = *
2873 ;
2874 *= $14
2875 0014 00 DSKSPR: .BYTE PRNORG-CRNTP4 ; ^GDISKP TOO LONG
2876 ;
2877 ; .PAGE

```

```

2878 .TITLE      'PRINTER      ***** PRINTP.SRC ***** 3/9/79 ***** 4:00:00 P
2879 ;
2880 ;
2881 ;
2882 ;
2883 ;
2884 ;
2885 ;
2886 ;
2887 ;
2888 ;
2889 ;
2890 ;      DEVICE NUMBER OR CODE EQUATES
2891 ;
2892 0002 OPNOUT      $2      ; IOCB OPEN FOR OUTPUT COMMAND
2893 0028 NBUFSZ      =      40      ; PRINT NORMAL BUFFER SIZE
2894 0014 DBUFSZ      =      20      ; PRINT DOUBLE BUFFFFER SIZE
2895 001D SBUFSZ      =      29      ; PRINT SIDEWAYS BUFFER SIZE
2896 0040 PDEVN      =      $40      ; PRINTER DEVICE NUMBER
2897 ; STATC      =      $53      ; DCB STATUS COMMAND CODE
2898 0057 WRITEC      =      $57      ; DCB WRITE COMMAND
2899 0020 SPACE      =      $20      ; ASCII SPACE CHAR.
2900 004E N          =      $4E      ; ASCII "N" CHAR.
2901 0044 D          =      $44      ; ASCII "D" CHAR.
2902 0053 S          =      $53      ; ASCII "S" CHAR.
2903 ;
2904 ;
2905 ;      PRINTER      HANDLER ENTRY POINTS
2906 ;
2907 ;
2908 ;
2909 ;
2910 ;      *=$E430
2911 ;
2912 E430 9E EE .WORD      PHOPEN-1      ; PRINTER HANDLER OPEN
2913 E432 DB EE .WORD      PHCLOS-1      ; PH CLOSE
2914 E434 9D EE .WORD      BADST-1      ; PH READ

```

```

2915 E436 A6 EE .WORD PHWRIT-1 ; PH WRITE
2916 E438 80 EE .WORD PHSTAT-1 ; PH STATUS
2917 E43A 9D EE .WORD BADST-1 ; PH SPECIAL
2918 E43C 4C 78 EE JMP PHINIT ; PH INIT.
2919 E43F 00 .BYTE 0 ; ROM FILLER
2920 ;
2921 ;
2922 ;
2923 ;
2924 ;
2925 ; *=PRNORG
2926 ;
2927 ;
2928 ;
2929 ;
2930 ; PRINTER HANDLER INITIALIZATION ROUTINE
2931 ;
2932 EE78 A9 1E PHINIT: LDA #30
2933 EE7A 85 1C STA PTIMOT ; SET UP INITIAL PRINTER TIMEOUT OF 30 SEC.
2934 EE7C 60 RTS
2935 ;
2936 ;
2937 ; PRINTER HANDLER CONSTANTS
2938 ;
2939 EE7D EA 02 PHSTLO: .WORD DVSTAT ; STATUS BUFFER POINTER
2940 EE7F C0 03 PHCHLO: .WORD PRNBUF ; CHAR. BUFFER POINTER
2941 ;
2942 ;
2943 ;
2944 ; *****
2945 ; PRINTER HANDLER ROUTINES
2946 ; *****
2947 ;
2948 ;
2949 ;
2950 ;
2951 ;

```

```

2952      ;      PRINTER HANDLER STATUS ROUTINE
2953      ;
2954  EE81  A9 04      PHSTAT:    LDA    114
2955  EE83  85 1E      STA    PBUFSZ      ; SET BUFFER SIZE TO 4 BYTES
2956  EE85  AE 7D EE   LDY    PHSTLO
2957  EE88  AC 7E EE   LDY    PHSTLO+1    ; SET POINTER TO STATUS BUFFER
2958  EEBB  A9 53      LDA    #STATC    ;SET COMMAND TO "STATUS"
2959  EEBD  8D 02 03   STA    DCOMND    ;SET STATUS COMMAND
2960  EE90  8D 0A 03   STA    DAUX1
2961  EE93  20 E6 EE   JSR    SETDCB      ; GO SETUP DCB
2962  EE96  20 59 E4   JSR    SIOV      ; SEND STATUS COMMAND
2963  EE99  30 03      BMI    BADST      ; GO IF ERROR
2964  EE9B  20 14 EF   JSR    PHPUT      ; YES. PUT STATUS INTO GLOBAL BUFFER.
2965  EE9E  60      BADST:    RTS
2966      ;
2967      ;
2968      ;
2969      ;
2970      ;      PRINTER      HANDLER      OPEN ROUTINE
2971      ;
2972  EE9F  20 81 EE   PHOPEN:    JSR    PHSTAT    ; DO STATUS COMMAND TO SIO
2973  EEA2  A9 00      LDA    #0
2974  EEA4  85 1D      STA    PBPNT      ; CLEAR PRINT BUFFER POINTER
2975  EEA6  60      RTS
2976      ;
2977      ;
2978      ;
2979      ;
2980      ;      PRINTER      HANDLER      WRITE ROUTINE
2981      ;
2982  EEA7  85 1F      PHWRIT:    STA    PTEMP      ; SAVE ACCUM
2983  EEA9  20 1A EF   JSR    PRMODE    ; GO DETERMINE PRINT MODE
2984  EEAC  A6 1D      LDY    PBPNT
2985  EEAE  A5 1F      LDA    PTEMP      ; GET CHAR. SENT BY CIO
2986  EEB0  9D C0 03   STA    PRNBUF,X    ; PUT CHAR. IN PRINT BUFFER
2987  EEB3  EB      INX      ; INCR. BUFFER POINTER
2988  EEB4  E4 1E      CPX    PBUFSZ      ; BUFFER POINTER=BUFFER SIZE?

```

```

2989 EEB6 F0 13      BEQ  BUFFUL
2990 EEBB 86 1D      STX  PBPNT      ; SAVE BUFFER POINTER
2991 EEBA C9 9B      CMP  #CR      ; IS CHAR. = EOL ?
2992 EEEBC F0 03      BEQ  BLFILL      ; IF YES, GO DO BLANK FILL.
2993 EEBE A0 01      LDY  #SUCCESS      ; PUT GOOD STATUS IN Y REG FOR CIO.
2994 EEC0 60          RTS
2995 EEC1 A9 20      BLFILL: LDA  #SPACE      ; PUT BLANK IN ACCUM.
2996 EEC3 9D C0 03    FILLBF: STA  PRNBUF,X      ; STORE IT IN PRINT BUFFER.
2997 EEC6 E8          INX
2998 EEC7 E4 1E      CPX  PBUFSZ
2999 EEC9 D0 F8      BNE  FILLBF      ; BUFFER BLANK FILLED?
3000 EECB A9 00      BUFFUL: LDA  #0
3001 EECD 85 ID      STA  PBPNT      ; CLEAR PRINT BUFFER POINTER
3002 EECF AE 7F EE      LDX  PHCHLO
3003 EED2 AC 80 EE      LDY  PHCHLO+1      ; SET POINTER TO PRINT BUFFER
3004 EED5 20 E6 EE      JSR  SETDCB      ; GO SETUP DCB
3005 EEDB 20 59 E4      JSR  SIOV      ; SEND PRINT COMMAND
3006 EEDB 60          RTS      ; YES.
3007                ;
3008                ;
3009                ;
3010                ;
3011                ;      PRINTER      HANDLER CLOSE ROUTINE
3012                ;
3013 EEDC 20 1A EF      PHCLOS: JSR  PRMODE      ; GO DETERMINE PRINT MODE
3014 EEDF A6 1D          LDX  PBPNT
3015 EEE1 D0 DE          BNE  BLFILL
3016 EEE3 A0 01          LDY  #SUCCESS
3017 EEE5 60          RTS
3018                ;
3019                ;
3020                ;
3021                ;
3022                ;
3023                ;
3024                ;
3025                ;

```

```

3026      ;      S U B R O U T I N E S
3027      ;
3028      ;
3029      ;
3030      ;
3031      ;
3032      ; SET UP DCB TO CALL SIO
3033      ;
3034  EEE6  BE 04 03  SETDCB:      STX      DBUFLO
3035  EEE9  BC 05 03      STY      DBUFHI      ; SET BUFFER POINTER
3036  EEEC  A9 40      LDA      #PDEVN
3037  EEEE  BD 00 03      STA      DDEVIC      ; SET PRINTER BUS I. D. FOR DCB
3038  EEF1  A9 01      LDA      #1
3039  EEF3  8D 01 03      STA      DUNIT      ; SET UNIT NUMBER TO 1
3040  EEF6  A9 80      LDA      #$80      ; DEVICE WILL EXPECT DATA
3041  EEF8  AE 02 03      LDX      DCOMND
3042  EEFB  E0 53      CPX      #STATC      ; STATUS COMMAND?
3043  EEFD  D0 02      BNE      PSIOC
3044  EEFF  A9 40      LDA      #$40      ; EXPECT DATA FROM DEVICE
3045  EF01  8D 03 03  PSIOC:      STA      DSTATS      ; SET SIO MODE COMMAND.
3046  EF04  A5 1E      LDA      PBUFSZ
3047  EF06  8D 08 03      STA      DBYTLO      ; SET LO BYTE COUNT
3048  EF09  A9 00      LDA      #0
3049  EF0B  8D 09 03      STA      DBYTHI      ; SET HI BYTE COUNT
3050  EF0E  A5 1C      LDA      PTIMOT
3051  EF10  SD 06 03      STA      DTIMLO      ; SET DEVICE TIMEOUT COUNT
3052  EF13  60      RTS
3053      ;
3054      ;
3055      ;
3056      ;
3057      ; GET DEVICE TIMEOUT FROM STATUS & SAVE IT
3058      ;
3059      ;
3059  EF14  AD EC 02  PHPUT:      LDA      DVSTAT+2
3060  EF17  85 1C      STA      PTIMOT      ; SAVE DEVICE TIMEOUT
3061  EF19  60      RTS

```



```

3062      ;
3063      ;
3064      ;
3065      ;
3066      ; DETERMINE PRINT MODE & SETUP PRINT BUFFER SIZE, DCB PRINT
3067      ;      COMMAND & DCB AUXI FOR PRINT MODE
3068      ;
3069      EF1A  A0 57      PRMODE:      LDY      #WRITEC      ; PUT WRITE COMMAND IN Y REG
3070      EF1C  A5 2B      LDA      ICAX2Z      ; READ PRINT MODE
3071      EF1E  C9 4E      CMODE:      CMP      #N
3072      EF20  D0 04      BNE      CDUBL      ; PRINT NORMAL ?
3073      EF22  A2 28      LDX      #NBUFSZ      ; YES SET NORMAL CHAR. BUFFER SIZE
3074      EF24  D0 0E      BNE      SETBSZ
3075      EF26  C9 44      CDUBL:      CMP      #D
3076      EF28  D0 04      BNE      CSIDE      ; PRINT DOUBLE?
3077      EF2A  A2 14      LDX      #DBUFSZ      ; YES SET DOUBLE CHAR. BUFFER SIZE
3078      EF2C  D0 06      BNE      SETBSZ
3079      EF2E  C9 53      CSIDE:      CMP      #S      ; PRINT SIDEWAYS ?
3080      EF30  D0 0B      BNE      GOERR      ; IF NOT GO TO ERROR ROUTINE
3081      EF32  A2 1D      LDX      #SBUFSZ      ; YES SET SIDEWAYS BUFFER SIZE
3082      EF34  86 1E      SETBSZ:      STX      PBUFSZ      ; STORE PRINT BUFFER SIZE
3083      EF36  8C 02 03      STY      DCOMND      ; STORE DCB COMMAND
3084      EF39  8D 0A 03      STA      DAUX1      ; STORE DCB AUXI PRINT MODE
3085      EF3C  60      RTS
3086      EF3D  A9 4E      GOERR:      LDA      #N      ; SET DEFAULT PRINT MODE TO NORMAL
3087      EF3F  D0 DD      BNE      CMODE
3088      ; *****
3089      ;
3090      ;
3091      ; SPARE BYTE OR MODULE TOO LONG FLAG
3092      ;
3093      EF41      CRNTP5      =      *
3094      3095      *= $14
3096      ;
3097      0014 00      PRNSPR: .BYTE CASORG-CRNTP5 : ^GPRINTP TOO LONG
3098      ;

```

```

3099 .PAGE
3100 .TITLE 'CASSET HANDLER 3/12 (DKI:CASCV) '
3101 0003 CBUFH = CASBUF/256
3102 00FD CBUFL = (-256)*CBUFH+CASBUF
3103 0040 SRSTA = $40 ; SIO READ STATUS
3104 0080 SWSTA = $80 ; SIO WRITE STATUS
3105 ; MOTRGO = $34
3106 ; MOTRST = $3C
3107 ;
3108 ;
3109 00FC DTA = $FC ; DATA RECORD TYPE BYTE
3110 00FA DTI = $FA ; LAST DATA RECORD
3111 00FE EOT = $FE ; END OF TAPE
3112 00FB HDR = $FB ; HEADER
3113 0002 TONE1 = 2 ; CHANGE TO RECORD MODE TONE
3114 0001 TONE2 = 1 ; PRESS PLAY TONE
3115 ;
3116 ;
3117 ;
3118 *=Casetv
3119 E440 48 EF 2A F0 .WORD OPENC-1.CLOSEC-1.GBYTE-1,PBYTE-1,STATU-1, SPECIAL-1
3120 E444 D5 EF 0F F0 ;
3121 E448 27 F0 4A EF ;
3122 E44C 4C 41 EF JMP INIT
3123 E44F 00 .BYTE 0 ; ROM FILLER BYTE
3124 ;
3125 ;
3126 ;
3127 ; USED IN MONITP FOR CASSETTE BOOT
3128 ;
3129 *=RBLOKV
3130 E47A 4C E9 EF JMP RBLOK
3131 ;
3132 *=CSOPIV
3133 E47D 4C 5D EF JMP OPINP
3134 ;
3135 ;

```

```

3136                                     *=CASORG
3137                                     ;
3138                                     ;
3139                                     ;
3139                                     ; INIT ROUTINE
3140                                     ;
3141 INIT:                                LDA    #$CC
3142 EF41 EF43 8D EE 02                  STA    CBAUDL
3143 EF46 A9 05                          LDA    #$05
3144 EF48 8D EF 02                      STA    CBAUDH      ; SET CASSET BAUD RATE TO 600
3145 SPECIAL:                          ; THATS ALL FOLKS
3146 EF4B 60                            RTS
3147                                     .PAGE
3148                                     ;
3149                                     ; OPEN FUNCTION - WITH NO TIMING ADJUST
3150                                     ;
3151 EF4C A5 2B                          OPENC:  LDA    ICAX2Z      ; GET AX2
3152 EF4E 85 3E                          STA    FTYPE      ; SAVE IT FOR FUTURE REFERENCE
3153 EF50 A5 2A                          LDA    ICAXIZ
3154 EF52 29 0C                          AND     #$0C      ; IN AND OUT BITS
3155 EF54 C9 04                          CMP     #$04
3156 EF56 F0 05                          BEQ     OPINP
3157 EF58 C9 05                          CMP     #$08      ; SEE IF OPEN FOR OUTPUT
3158 EF5A F0 39                          BEQ     OPOUT
3159 EF5C 60                            RTS              ; IF ALREADY OPEN. RETURN LEAVING STATUS=f84
3160 EF5D A9 00                          OPINP:  LDA    #0
3161 EF5F 8D 89 02                      STA    WMODE      ; SET READ MODE
3162 EF62 85 3F                          STA    FEOF      ; NO EOF YET
3163 EF64 A9 01                          SFH:   LDA    #TONE2    ; TONE FOR PRESS PLAY
3164 EF66 20 58 F0                      JSR     BEEP      ; GO BEEP
3165 EF69 30 24                          BMI     OPNRTN    ; IF ERROR DURING BEEP
3166 EF6B A9 34                          LDA    #MOTRGO
3167 EF6D 8D 02 D3                      STA    PACTL      ; TURN MOTOR ON
3168                                     .IF    PALFLG
3169                                     LDY    #$E0
3170                                     LDX    #1
3171                                     .ENDIF

```

```

3172      .IF      PALFLG-1
3173      EF70      A0 40      LDY      #$40      ; 5-31-79 9 SEC READ LEADER
3174      EF72      A2 02      LDX      #2
3175      .ENDIF
3176      EF74      A9 03      LDA      #3
3177      EF76      8D 2A 02      STA      CDTMF3
3178      EF79      20 5C E4      JSR      SETVBV      ; SET UP VBLANK TIMER
3179      EF7C      AD 2A 02      WAITTM: LDA      CDTMF3
3180      EF7F      D0 FB      BNE      WAITTM      ; WAIT FOR MOTOR TO COME UP TO SPEED
3181      EFB1      A9 80      LDA      #$80      ; NEXT BYTE=NO BYTES IN BUFFER
3182      EF83      85 3D      STA      BPTR
3183      EF85      8D 8A 02      STA      BLIM
3184      EF88      4C D3 EF      JMP      OPOK      ; OPEN OK
3185      ;
3186      ; OPEN FOR OUTPUT
3187      ;
3188      EF8B      A0 80      PBRK:      LDY      #BRKABT      ; BREAK KEY ABORT STATUS
3189      EF8D      C6 11      DEC      BRKKEY      ; RESET BREAK KEY
3190      EFBF      A9 00      OPNRTN:   LDA      #0      ; CLEAR WRITE MODE FLAG
3191      EF91      8D 89 02      STA      WMODE
3192      EF94      60      RTS      ; AND EXIT.
3193      ;
3194      EF95      A9 80      OPOUT:   LDA      #$80
3195      EF97      8D 89 02      STA      WMODE      ; SET WRITE MODE
3196      EF9A      A9 02      LDA      #TONE1      ; TELL USER TO TURN ON RECORD MODE
3197      EF9C      20 58 F0      JSR      BEEP
3198      EF9F      30 EE      BMI      OPNRTN      ; IF ERROR DURING BEEP
3199      EFA1      A9 CC      LDA      #$CC      ; SET BAUD RATE
3200      EFA3      8D 04 D2      STA      AUDF3      ; WHICH SEEMS TO BE NESSECARY
3201      EFA6      A9 05      LDA      #$05      ; FOR SOME OBSCURE REASON
3202      EFAB      8D 06 D2      STA      AUDF4
3203      EFAB      A9 60      LDA      #$60
3204      EFAD      8D 00 03      STA      DDEVIC
3205      EFB0      20 68 E4      JSR      SENDEV      ; TELL POKEY TO WRITE MARKS
3206      EFB3      A9 34      LDA      #MOTRGO      ; WRITE 5 SEC BLANK TAPE
3207      EFB5      8D 02 D3      STA      PACTL
3208      EFBB      A9 03      LDA      #3

```

```

3209      .IF    PALFLG
3210              LDX    #$3
3211              LDY    #$C0
3212      .ENDIF
3213      .IF    PALFLG-1
3214      EFBA  A2 04              LDX    #4          ; 5/30/79 20 SEC LEADER
3215      EFBC  A0 80              LDY    #$80
3216      .ENDIF
3217      EFBE  20 5C E4          JSR    SETVBV
3218      EFC1  A9 FF              LDA    #$FF
3219      EFC3  8D 2A 02          STA    CDTMF3
3220      EFC6  A5 11      WDLR:  LDA    BRKKEY
3221      EFC8  F0 C1              BEQ    PERK          ; IF BREAK DURING WRITE LEADER
3222      EFCA  AD 2A 02          LDA    CDTMF3
3223      EFCD  D0 F7              BNE    WDLR
3224      EFCF  A9 00              LDA    #0          ; INIT BUFFER POINTER
3225      EFD1  85 3D              STA    BPTR
3226      EFD3  A0 01      OPOK: LDY    #SUCCES
3227      EFD5  60              RTS
3228              .PAGE
3229      ;
3230      ; GET BYTE
3231      ;
3232      EFD6  A5 3F      GBYTE:  LDA    FEOF          ; IF AT EOF ALREADY
3233      EFDB  30 33              BMI    ISEOF          ; RETURN EOF STATUS
3234      EFDA  A6 3D              LDX    BPTR          ; BUFFER POINTER
3235      EFDC  EC SA 02          CPX    BLIM          ; IF END OF BUFFER
3236      EFDF  F0 08              BEQ    RBLOK          ; READ ANOTHER BLOCK
3237      EFE1  BD 00 04          LDA    CASBUF+3,X      ; GET NEXT BYTE
3238      EFE4  E6 3D              INC    BPTR          ;BUMP POINTER
3239      EFE6  A0 01              LDY    #SUCCES          ; OK STATUS
3240      EFEB  60      GBX:      RTS
3241      EFE9  A9 52      RBLOK:  LDA    #'R          ; READ OPCGDE
3242      EFEB  2095 F0          JSR    SIOSB          ; SIO ON SYS BUF
3243      EFEE  98              TYA
3244      EFEF  30 F7              BMI    GBX          ; IF SIO ERRORS. RETURN
3245      EFF1  A9 00              LDA    #0

```

```

3246 EFF3 85 3D          STA  BPTR          ; RESET POINTER
3247 EFF5 A2 80          LDX  #$80          ; DEFAULT # BYTES
3248 EFF7 AD FF 03        LDA  CASBUF+2
3249 EFFA C9 FE          CMP  #EDT
3250 EFFC F0 0D          BEQ  ATEOF          ; IF HEADER, GO READ AGAIN
3251 EFFE C9 FA          CMP  #DT1          ; IF LAST DATA REC
3252 F000 D0 03          BNE  NLR
3253 F002 AE 7F 04        LDX  CASBUF+130    ; LAST DATA RECORD, GET # BYTES
3254 F005 8E 8A 02        NLR:             STX  BLIM
3255 F008 4C D6 EF        JMP  GBYTE          ; GET NEXT BYTE
3256 F00B C6 3F          ATEOF:             DEC  FEOF          ; SET FEOF
3257 F00D A0 88          ISEOF:            LDY  #EOFERR        ; ENDFILE STATUS
3258 F00F 60             RTS
3259                     .PAGE
3260                     ;
3261                     ; PUT BYTE TO BUFFER
3262                     ;
3263 F010 A6 3D          PBYTE:             LDX  BPTR          ; BUFFER POINTER
3264 F012 9D 00 04        STA  CASBUF+3,X    ; STORE CHAR AWAY
3265 F015 E6 3D          INC  BPTR          ; BUMP POINTER
3266 F017 A0 01          LDY  #SUCCESS      ; OK STATUS
3267 F019 E0 7F          CPX  #127          ; IF BUFFER FULL
3268 F01B F0 01          BEQ  *+3
3269 F01D 60             RTS
3270                     ; WRITE OUT THE  BUFFER
3271 F01E A9 FC          LDA  #DTA          ; RECORD TYPE = DATA
3272 F020 20 D2 F0        JSR  WSIOB        ; DO WRITE ON SYSTEM BUFFER
3273 F023 A9 00          LDA  #0
3274 F025 85 3D          STA  BPTR          ; RESET BUFFER POINTER
3275 F027 60             RTS                ; EXIT.
3276                     .PAGE
3277                     ;
3278                     ; STATUS - RETURN STATUS INFO THRU DVSTAT
3279                     ;
3280 F028 A0 01          STATU:             LDY  #SUCCESS
3281 F02A 60             RTS
3282                     .PAGE

```

```

3283      ;
3284      ; CLOSE
3285      ;
3286 F02B AD 89 02 CLOSEC: LDA WMODE      ; SEE IF WRITING
3287 F02E 30 08      BMI CLWRT      ; GO CLOSE FOR WRITE
3288      ; CLOSE FOR READ - FLAG CLOSED
3289 F030 A0 01      LDY #SUCCES      ; SUCCESSFULL
3290 F032 A9 3C FCAX: LDA *MOTRST      ; STOP THE MOTOR IN CASE WAS SHORT IRG MODE
3291 F034 8D 02 D3      STA PACTL
3292 F037 60      RTS
3293 F038 A6 3D CLWRT: LDX BPTR      ; BUFFER POINTER
3294 F03A F0 0A      BEQ WTLR      ; IF NO DATA BYTES IN BUFFER NO DTI REC
3295 F03C 8E 7F 04      STX CASBUF+130 ; WRITE TO LAST RECORD
3296 F03F A9 FA      LDA #DT1      ; REC TYPE
3297 F041 20 D2 F0      JSR WSIO SB      ; WRITE OUT USER BUFFER
3298 F044 30 EC      BMI FCAX      ; GO IF ERROR
3299 F046 A2 7F WTLR: LDX #127      ; ZERO BUFFER
3300 F048 A9 00      LDA #0
3301 F04A 9D 00 04 ZTBUF: STA CASBUF+3,X
3302 F04D CA      DEX
3303 F04E 10 FA      BPL ZTBUF
3304 F050 A9 FE      LDA #EOT      ; WRITE EOT RECORD
3305 F052 20 D2 F0      JSR WSIO SB
3306 F055 4C 32 F0      JMP FCAX      ; FLAG CLOSED AND EXIT
3307      .PAGE
3308      ;
3309      ; SUBROUTINES
3310      ;
3311      ; BEEP - GENERATE TONE ON KEYBOARD SPEAKER
3312      ; ON ENTRY A = FREQ
3313      ;
3314 F058 85 40 BEEP: STA FREQ
3315 F05A A5 14 BEEP1: LDA RTCLOK+2      ; CURRENT CLOCK
3316 F05C 18      CLC
3317      .IF PALFLG
3318      ADC #25
3319      .ENDIF

```

```

3320          .IF    PALFLG-1
3321 F05D  69 1E          ADC    #30          ; 1 SEC TONE
3322          .ENDIF
3323 F05F  AA          TAX
3324 F060  A9 FF      WFL:    LDA    #xFF
3325 F062  8D 1F D0      STA    CONSOL      ; TURN ON SPEAKER
3326 F065  A9 00          LDA    #0
3327 F067  A0 F0          LDY    #$F0
3328 F069  88          DEY
3329 F06A  D0 FD          BNE    #-1
3330 F06C  8D IF D0      STA    CONSOL      ; TURN OFF SPEAKER
3331 F06F  A0 F0          LDY    #$F0
3332 F071  88          DEY
3333 F072  D0 FD          BNE    #-1
3334 F074  E414          CPX    RTCLOK+2      ; SEE IF 1 SEC IS UP YET
3335 F076  D0 ES          BNE    WFL
3336 F078  C6 40          DEC    FREQ          ; COUNT BEEPS
3337 F07A  F0 0B          BEQ    WFAK          ; IF ALL DONE GO WAIT FOR KEY
3338 F07C  8A          TXA
3339 F07D  18          CLC
3340          .IF    PALFLG
3341          ADC    #8
3342          .ENDIF
3343          .IF    PALFLG-1
3344 F07E  69 0A          ADC    #10
3345          .ENDIF
3346 F080  AA          TAX
3347 F081  E4 14          CPX    RTCLOK+2
3348 F083  D0 FC          BNE    *-2
3349 F085  F0 D3          BEQ    BEEP1          ; UNCOND GO BEEP AGIN
3350 F087  20 8C F0      WFAK:    JSR    WFAK1      ; USE SIMULATED "JMP (KGETCH)"
3351 F08A  98          TYA
3352 F08B  60          RTS
3353 F08C  AD 25 E4      WFAK1:    LDA    KEYBDV+5
3354 F08F  48          PHA
3355 F090  AD 24 E4          LDA    KEYBDV+4      ; SIMULATE "JMP (KGETCH)"
3356 F093  48          PHA

```



```

3357 F094 60                      RTS
3358                               ;
3359                               ; SIOSB - CALL SIO ON SYSTEM BUFFER
3360                               ;
3361 F095 8D 02 03      SIOSB:     STA     DCOMND      ; SAVE COMMAND
3362 F098 A9 00          LDA     #0
3363 F09A 8D 09 03      STA     DBYTHI      ; SET BUFFER LENGTH
3364 F09D A9 83          LDA     *131
3365 F09F 8D 08 03      STA     DBYTLO
3366 F0A2 A9 03          LDA     #CBUFH
3367 F0A4 8D 05 03      STA     DBUFHI      ; SET BUFFER ADDRESS
3368 F0A7 A9 FD          LDA     #CBUFL
3369 F0A9 8D 04 03      STA     DBUFLO
3370 F0AC A9 60          CSIO:     LDA     #$60      ; CASSET PSEUDO DEVICE
3371 F0AE 8D 00 03      STA     DDEVIC
3372 F0B1 A9 00          LDA     #0
3373 F0B5 8D 01 03      STA     DUNIT
3374 F0B6 A9 23          LDA     #35      ; DEVICE TIMEOUT (5/30/79)
3375 F0B8 8D 06 03      STA     DTIMLO
3376 F0B8 AD 02 03      LDA     DCOMND      ; GET COMMAND BACK
3377 F0BE A0 40          LDY     #SRSTA      ; SIO READ STATUS COMMAND
3378 F0C0 C9 52          CMP     #'R
3379 F0C2 F0 02          BEQ     *+4
3380 F0C4 A0 80          LDY     #SWSTA      ; SIO WRITE STATUS COMMAND
3381 F0C6 8C 03 03      STY     DSTATS      ; SET STATUS FOR SIO
3382 F0C9 A5 3E          LDA     FTYPE
3383 F0CB 8D 0B 03      STA     DAUX2      ; INDICATE IF SHORT IRG MODE
3384 F0CE 20 59 E4      JSR     SIOV      ; GO CALL SIO
3385 F0D1 60                      RTS
3386                               ;
3387                               ; WSIOB - WRITE SIO SYSTEM  BUFFER
3388                               ;
3389 F0D2 8D FF 03      WSIOB:     STA     CASBUF+2      ; STORE TYPE BYTE
3390 F0D5 A9 55          LDA     #455
3391 F0D7 8D FD 03      STA     CASBUF+0
3392 F0DA 8D FE 03      STA     CASBUF+1
3393 F0DD A9 57          LDA     #'W      ; WRITE

```

ERR LINE ADDR B1 B2 B3

SIO (SERIAL INPUT/OUTPUT BUS CONTROLLER)

Page 50

3394	F0DF	20 95 F0		JSR	SIOSB	; CALL SIO ON SYSTEM BUFFER
3395	F0E2	60		RTS	AND	; RETURN
3396	F0E3		CRNTP6	=*		
3397				*=\$14		
3398	0014	00	CASSPR:	.BYTE	MONORG-CRNTP6	; ^GCASCV IS TOO LONG
3399				;		