

:ASM

```
1010          .OR $0000
1020          .TA $0800
1030 *****
1040 *
1050 * ROBOT MONITOR
1060 * VERSION 1.1 REV. D
1070 * COPYRIGHT 1984, ARCTEC SYSTEMS INC.
1080 *
1090 * WRITTEN BY: MICHAEL W. FOWLER
1100 * DATE: 11/19/84
1110 *****
1120          .IN ROMON.V1.1.PART1
1130 *-----
1140 * MEMORY DECODE MAP
1150 *      SELECT0= $0000-1FFF RAM 1 (BATTERY BACKED)
1160 *      SELECT1= $2000-3FFF RAM 2 (BATTERY BACKED)
1170 *      SELECT2= $4000-5FFF RAM 3 (BATTERY BACKED)
1180 *      SELECT3= $6000-7FFF RAM 4 (BATTERY BACKED)
1190 *      SELECT4= $8000-9FFF RAM 5 (BATTERY BACKED)
1200 *      SELECT5= $A000-BFFF RAM 6 (BATTERY BACKED)
1210 *      SELECT6= $C000-DFFF RAM 7 (BATT BACKED)/ ROMS 1-7
1220 *      SELECT7= $E000-FFFF I/O AND MONITOR SPACE
1230 *-----
1240 * I/O DECODING
1250 *      $E000-E00F= LCD DISPLAY
1260 *      $E010-E017= REAL TIME CLOCK
1270 *      $E018-E01F= TIMER
1280 *      $E020-E02F= SERIAL PORTS
1290 *      $E030-E03F= PARALLEL PORTS
1300 *      $E040-E04F= A/D CONVERTER
1310 *      $E050-E05F= BIT INPUT PORTS
1320 *      $E060-E06F= BIT OUTPUT PORTS
1330 *      $E070-E073= RANDOM NUMBER GENERATOR
1340 *      $E074-E077= I/R KEYBOARD
1350 *      $E078-E07B= BELL
1360 *      $E07C-E07F= NMI RESET
1370 *      $E080-E09F= SLOT 1 REGISTER
1380 *      $E0A0-E0BF= SLOT 2 REGISTER
1390 *      $E0C0-E0DF= SLOT 3 REGISTER
1400 *      $E0E0-E0FF= SLOT 4 REGISTER
1410 *      $E100-E17F= SLOT 1 ADDRESS SPACE
1420 *      $E180-E1FF= SLOT 2 ADDRESS SPACE
1430 *      $E200-E27F= SLOT 3 ADDRESS SPACE
1440 *      $E280-E2FF= SLOT 4 ADDRESS SPACE
1450 *      $E300-E37F= NOT USED
1460 *      $E380-E3FF= NOT USED
1470 *-----
1480 * SERIAL PORT ASSIGNMENTS ($E020-E02F)
1490 *      PORT 0= PROPULSION CONTROLLER
1500 *      PORT 1= VOICE I/O AND SOUND
1510 *      PORT 2= RF LINK
1520 *      PORT 3= EXTERNAL RS-232
1530 *      PORT 4= EXPANSION 1
1540 *      PORT 5= EXPANSION 2
1550 *-----
1560 * BIT PORT ASSIGNMENTS
1570 *      OUTPUT BIT PORT ($E060-E06F)
```

```

1450 *PA0    BIT0= PROCON ENABLED=1
1460 *PA1    BIT1= VIDS ENABLED=1
1470 *PA2    BIT2= BATT GND CONNECTED=0
1480 *PA3    BIT3= SONAR ENABLED=1
1490 *PA4    BIT4= DOOR DETECT ENABLED=0
1500 *PA5    BIT5= SONAR VSW ON=0
1510 *PA6    BIT6= BEACON DETECT ENABLED=0
1520 *PA7    BIT7= RF MODEM ENABLED=1
1530 *PC7    BIT8= PHASE0 STEPPER MTR
1540 *PC6    BIT9= PHASE1
1550 *PC5    BIT10=PHASE2
1560 *PC4    BIT11=PHASE3
1570 *PC3    BIT12=SONAR XDUCER SELECTA
1580 *PC2    BIT13=SONAR XDUCER SELECTB
1590 *PC1    BIT14=SONAR XDUCER SELECTC
1600 *PC0    BIT15=SONAR XDUCER SELECTD
1610 *PB7    BIT16=CPU CLK SPEED SELECT
1620 *PB6    BIT17=SERIAL SPARE 1 EN
1630 *PB5    BIT18=SERIAL SPARE 2 EN
1640 *PB4    BIT19=SLOT BIT SELECT1
1650 *PB3    BIT20=SLOT BIT SELECT2
1660 *PB2    BIT21=SLOT BIT SELECT3
1670 *PB1    BIT22=SLOT BIT OUTPUT1
1680 *PB0    BIT23=SLOT BIT OUTPUT2
1690 *-----
1700 *  INPUT BIT PORT ($E050-E05F)
1710 *PA0    BIT0= ON CHARGE=0
1720 *PA1    BIT1= IR BEACON DETECTED=1
1730 *PA2    BIT2= DOOR REFLECTOR DETECTED=1
1740 *PA3    BIT3= SMOKE DETECTED=0
1750 *PA4    BIT4= MOTION DETECTED=0
1760 *PA5    BIT5= RF CARRIER DETECTED=0
1770 *PA6    BIT6= RIGHT HEAD STOP HIT=0
1780 *PA7    BIT7= LEFT HEAD STOP HIT=0
1790 *PC7    BIT8= FUNCTION KEY 1
1800 *PC6    BIT9= FUNCTION KEY 2
1810 *PC5    BIT10=FUNCTION KEY 3
1820 *PC4    BIT11=FUNCTION KEY 4
1830 *PC3    BIT12=FUNCTION KEY 5
1840 *PC2    BIT13=SPARE TO SSS
1850 *PC1    BIT14=SPARE TO SSS
1860 *PC0    BIT15=SPARE TO SSS
1870 *PB7    BIT16=EXTERNAL SERIAL BAUD RATE SWITCH1
1880 *PB6    BIT17=EXTERNAL SERIAL BAUD RATE SWITCH2
1890 *PB5    BIT18=EXTERNAL SERIAL BAUD RATE SWITCH3
1900 *PB4    BIT19=INTERNAL SERIAL BAUD RATE SWITCH
1910 *PB3    BIT20=NU
1920 *PB2    BIT21=NU
1930 *PB1    BIT22=SLOT BIT INPUT1
1940 *PB0    BIT23=SLOT BIT INPUT2
1950 *-----
1960 *  A/D CONVERTER PORTS
1970 *-----
1980 *  INPUT 0= SLOT 1
1990 *  INPUT 1= SLOT 2
2000 *  INPUT 2= SLOT 3
2010 *  INPUT 3= SLOT 4
2020 *  INPUT 4= SPARE 1
2030 *  INPUT 5= SPARE 2
2040 *  INPUT 6= SPARE 3

```

```

2050 * INPUT 7= SPARE 4
2060 * INPUT 8= BATTERY #1, LEFT MOTOR
2070 * INPUT 9= BATTERY #2, RIGHT MOTOR
2080 * INPUT A= BATTERY #3, LOGIC
2090 * INPUT B= SOUND LEVEL
2100 * INPUT C= BAROMETER
2110 * INPUT D= LIGHT LEVEL
2120 * INPUT E= TEMP LEVEL
2130 * INPUT F= MEMORY BATTERY LEVEL
2140 *-----
2150 * ZERO PAGE MEMORY LOCATIONS
2160 *-----
0000- 2170 ZERO .EQ $00 NUMBER 0 AND ADDR 0
0000- 2180 FRESFC .BS $10 FREE SPACE
0010- 2190 HIMEM .BS $02 HIGH MEMORY LOC
0012- 2200 LOMEM .BS $02 LOW MEMORY LOC
0014- 2210 IRQLOC .BS $02 IRQ INDIRECT ADDR
0016- 2220 NMILOC .BS $02 NMI INDIRECT ADDR
0018- 2230 ALMIRO .BS $02 ALARM INDIRECT ADDR
001A- 2240 CLKIRQ .BS $02 CLOCK IRQ INDIRECT ADDR
001C- 2250 CHROUT .BS $02 CHARACTER OUT INDIRECT ADDR
001E- 2260 CHARIN .BS $02 CHARACTER IN INDIRECT ADDR
0020- 2270 CHARPL .BS $02 CHARACTER POLL INDIRECT ADDR
0022- 2280 USRMON .BS $02 USER MONITOR COMMAND INDIRECT ADDR
0024- 2290 COLDAD .BS $01 COLD START MARKER
0025- 2300 TIMBUF .BS $05 REAL TIME CLOCK TIME BUFFER
002A- 2310 YEARBF .BS $02 YEAR BUFFER
002C- 2320 DOWBUF .BS $01 DAY OF WEEK
002D- 2330 CLKMOD .BS $01 CLOCK MODE
002E- 2340 BUFcnt .BS $01 LINE BUFFER COUNTER
002F- 2350 ROWCNT .BS $01 ROW COUNTER FOR LCD
0030- 2360 CURADD .BS $01 CURRENT ADDRESS DISPLAYED BY MONITOR
0031- 2370 TEMP00 .BS $01
0032- 2380 TEMP01 .BS $01
0033- 2390 TEMP02 .BS $01
0034- 2400 TEMP03 .BS $01
0035- 2410 TEMP04 .BS $01
0036- 2420 TEMP05 .BS $01
0037- 2430 INDEX1 .BS $02
0039- 2440 ADRPTR .BS $02 ADDRESS POINTER FOR BANK #0
003B- 2450 DSPIMG .BS $02 DISPLAY IMAGE POINTER
003D- 2460 ENDIMG .BS $02 END OF DISPLAY IMAGE POINTER
003F- 2470 ADDR01 .BS $02 FIRST ADDRESS
0041- 2480 ADDR02 .BS $02
0043- 2490 ADDR03 .BS $02
0045- 2500 ADDR04 .BS $02 ADDRESS TABLE POINTER
0047- 2510 PCL .BS $02 PROGRAM COUNTER LOW
0049- 2520 PCH .BS $02 PROGRAM COUNTER HIGH
004B- 2530 REGA .BS $01
004C- 2540 REGX .BS $01
004D- 2550 REGY .BS $01
004E- 2560 REGP .BS $01
004F- 2570 REGS .BS $01
0050- 2580 HDSTAT .BS $01 HEAD STATUS BIT FLAGS
0051- 2590 CPUFLG .BS $01 CPU STATUS FLAG
2600 * BIT7= CPU SPEED, LO= LOW SPEED
2610 * BIT6= BATTERY GROUND, LO= NOT CONNECTED
2620 * BIT5= TIME OUT INDICATOR
2630 * BIT4= NEW YEAR BIT
2640 * BIT3= ALARM ON

```

	2650 *		BIT0= ERROR INDICATOR
0052-	2660 MSGNUM .BS #02		MESSAGE OUT POINTER
0054-	2670 QCL .BS #01		MATH PAC VARIABLES
0055-	2680 QCM .BS #01		
0056-	2690 QCH .BS #01		
0057-	2700 TCL .BS #01		
0058-	2710 TCM .BS #01		
0059-	2720 TCH .BS #01		
005A-	2730 FLG .BS #01		
005B-	2740 FLH .BS #01		
005C-	2750 RCL .BS #01		
005D-	2760 RCM .BS #01		
005E-	2770 RCH .BS #01		
005F-	2780 TMP .BS #01		
0060-	2790 CNT .BS #01		
0061-	2800 ACX .BS #01		
0062-	2810 ACL .BS #01		
0063-	2820 ACM .BS #01		
0064-	2830 ACH .BS #01		
0065-	2840 SCL .BS #01		
0066-	2850 SCM .BS #01		
0067-	2860 SCH .BS #01		
0068-	2870 SPARES .BS #05		SPARE MONITOR LOCATIONS
006D-	2880 USRLIV .BS #02		USER LIVING LOOP LOC
006F-	2890 LIFEAD .BS #02		LIFE LOOP INDIRECT LOC
0071-	2900 USRVOC .BS #02		USER VOCOL LOC
0073-	2910 DEMST .BS #01		
0074-	2920 ROSTAT .BS #01		
0075-	2930 STAFLG .BS #01		
0076-	2940 TRNST .BS #01		
0077-	2950 MSGPTR .BS #02		
	2960 *-----		
	2970 * NON ZERO PAGE MEMORY LOCATIONS		
	2980 *-----		
	2990 .OR \$0540		
	3000 .TA \$0940		
0540-	3010 BUFPTR .BS #07		UNPACK/MSGOUT LOCS
0547-	3020 CATN .BS #02		
0549-	3030 PACKBF .BS #03		
054C-	3040 UPACBF .BS #04		
0550-	3050 REG .BS #01		
0551-	3060 BANKAD .BS #02		ADDRESS OF ROUTINE IN NEW BANK
0553-	3070 CURBNK .BS #01		CURRENT BANK OF ROM SELECTED
0554-	3080 KEYCHR .BS #01		LAST KEYBOARD CHAR READ
0555-	3090 LINCNT .BS #01		POSITION COUNT OF CURRENT LINE
0556-	3100 SAVEA .BS #01		'A' REGISTER TEMP STORAGE
0557-	3110 SAVEX .BS #01		'X' REGISTER TEMP STORAGE
0558-	3120 SAVEY .BS #01		'Y' REGISTER TEMP STORAGE
0559-	3130 ADMODE .BS #01		FORMAT CODE FOR DISASSEM
055A-	3140 BYTCNT .BS #01		BYTE COUNT FOR DISASSEMBLER
055B-	3150 PROMPT .BS #01		PROMPT CHARACTER
055C-	3160 VTPOS .BS #01		VERTICAL POS OF LCD
055D-	3170 HZPOS .BS #01		HORIZONTAL POSITION OF LCD
055E-	3180 HEXCHR .BS #01		HEX CHAR TO CONVERT
055F-	3190 RDMSK .BS #01		READ DATA MASK
0560-	3200 CTSMSK .BS #01		CTS MASK
0561-	3210 RTSLO .BS #01		RTS LOW MASK
0562-	3220 TDLO .BS #01		TD LOW MASK
0563-	3230 MASK .BS #01		SERIAL MASK
0564-	3240 CHARIO .BS #01		CHARACTER I/O

0565-	3250	BDRATE	.BS	\$01	BAUD RATE
0566-	3260	SPRTNM	.BS	\$01	CURRENT SERIAL PORT NUMBER
0567-	3270	MSPSTP	.BS	\$01	STEPPER MOTOR MS/STEP
0568-	3280	STEPS	.BS	\$01	#STEPS TO TAKE
0569-	3290	STEPNO	.BS	\$01	#STEPS TAKEN
056A-	3300	CURHD	.BS	\$01	CURRENT HEAD POSITION
056B-	3310	SERPRT	.BS	\$01	CURRENT SERIAL PORT IN USE
056C-	3320	PORTNM	.BS	\$01	UP/DOWN LOAD PORT
056D-	3330	AVGLO	.BS	\$03	RANGER AVERAGE
	3340	*-----			
	3350	* MATH PAC WORK SPACE			
	3360	*-----			
	3370		.OR	\$0600	
	3380		.TA	\$0800	
0600-	3390	VAR	.BS	\$0100	
	3400	*-----			
	3410	* CONSTANTS			
	3420	*-----			
0007-	3430	BIT07	.EQ	\$07	
0006-	3440	BIT06	.EQ	\$06	
0005-	3450	BIT05	.EQ	\$05	
0004-	3460	BIT04	.EQ	\$04	
0003-	3470	BIT03	.EQ	\$03	
0002-	3480	BIT02	.EQ	\$02	
0001-	3490	BIT01	.EQ	\$01	
0000-	3500	BIT00	.EQ	\$00	
0080-	3510	BIT7	.EQ	\$80	BIT 7
0040-	3520	BIT6	.EQ	\$40	BIT 6
0020-	3530	BIT5	.EQ	\$20	BIT 5
0010-	3540	BIT4	.EQ	\$10	BIT 4
0008-	3550	BIT3	.EQ	\$08	BIT 3
0004-	3560	BIT2	.EQ	\$04	BIT 2
0002-	3570	BIT1	.EQ	\$02	BIT 1
0001-	3580	BIT0	.EQ	\$01	BIT 0
0001-	3590	RDRF	.EQ	\$01	ACIA RECV DATA REG FULL
00E0-	3600	BRMASK	.EQ	\$E0	EXTERNAL BAUD SWITCH MASK
0010-	3610	BRMSK1	.EQ	\$10	INTERNAL BAUD SWITCH MASK
0011-	3620	ACIASU	.EQ	\$11	ACIA SETUP CONSTANT
0013-	3630	ACIARS	.EQ	\$13	ACIA MASTER RESET
0080-	3640	SUBOP	.EQ	\$80	SETUP FOR BIT OUTPUT PORTS
0098-	3650	SUSERP	.EQ	\$98	SETUP FOR BIT SERIAL PORTS
0005-	3660	SUBP0	.EQ	\$05	SET UP BIT PORT #0
0080-	3670	SUBP1	.EQ	\$80	SETUP BIT PORT #1
00F0-	3680	SUBP2	.EQ	\$F0	SETUP BIT PORT #2
0001-	3690	DIREC	.EQ	\$01	HEAD DIRECTION BIT
0004-	3700	BECREF	.EQ	\$04	ROOM BEACON DETECTED FLAG BIT
0005-	3710	DORREF	.EQ	\$05	DOOR REFLECTOR FLAG BIT
0000-	3720	HDINIT	.EQ	\$00	HEAD INITIALIZED BIT
0002-	3730	BECBIT	.EQ	\$02	BEACON BIT MASK
0004-	3740	DORBIT	.EQ	\$04	DOOR BIT MASK
0080-	3750	LFTSPB	.EQ	\$80	HEAD LEFT STOP BIT MASK
0040-	3760	RGTSPB	.EQ	\$40	HEAD RIGHT STOP BIT MASK
0000-	3770	HDSO NR	.EQ	\$00	HEAD SONAR NUMBER
0001-	3780	RTFSO N	.EQ	\$01	RIGHT TOP FRONT SONAR
0002-	3790	RBFSON	.EQ	\$02	RIGHT BOTTOM FRONT SONAR
0003-	3800	LTFSON	.EQ	\$03	LEFT TOP FRONT SONAR
0004-	3810	LBFSO N	.EQ	\$04	LEFT BOTTOM FRONT SONAR
0005-	3820	RSDSON	.EQ	\$05	RIGHT SIDE SONAR
0006-	3830	LSDSON	.EQ	\$06	LEFT SIDE SONAR
0007-	3840	FCTSON	.EQ	\$07	FRONT CENTER SONAR

```

0008-      3850 BCKSON .EQ $08      BACK SONAR
0009-      3860 SPRSON .EQ $09      SPARE SONAR
000A-      3870 XDUCRS .EQ $0A      NUMBER OF VALID SONARS
000B-      3880 ENBSON .EQ $08      ENABLE SONARS
00F7-      3890 SONOFF .EQ $F7      SONARS OFF
0020-      3900 VSWON .EQ $20      VSW ON
00DF-      3910 VSWOFF .EQ $DF      VSW OFF
00EF-      3920 RGTSTP .EQ $EF      RIGHT HEAD STOP
0003-      3930 ETX .EQ $03      END OF TRANSMISSION
0000-      3940 SPKFLG .EQ $00
0001-      3950 DSPFLG .EQ $01
0005-      3960 WATFLG .EQ $05
0006-      3970 SPCFLG .EQ $06
0007-      3980 CDSFLG .EQ $07
0063-      3990 RANGE .EQ ACM
DFFD-      4000 RFCHOT .EQ $DFFD      RF CHAR OUT POINTER
DFFA-      4010 DATIN .EQ $DFFA      RF CHAR IN POINTER
      4020 *-----
      4030 * MEMORY EQUATES
      4040 *-----
0200-      4050 LINBUF .EQ $0200      INPUT LINE BUFFER
0300-      4060 MSGBUF .EQ $0300      MESSAGE BUFFER
0400-      4070 SCNIMG .EQ $0400      BEGINNING OF LCD IMAGE
053F-      4080 SCNEND .EQ $053F      END OF LCD IMAGE
C000-      4090 CATLOG .EQ $C000
E000-      4100 SCNADD .EQ $E000      LCD SCREEN BASE ADDR
E011-      4110 RTCFRE .EQ $E011      REAL TIME CLOCK FREEZE ADDR
E012-      4120 RTCBAS .EQ $E012      REAL TIME CLOCK BASE ADDR
E017-      4130 RTCCSR .EQ $E017      REAL TIME CLOCK COMMAND AND STATUS
E018-      4140 RNGADD .EQ $E018      TIMER/RANGER BASE ADDR
E020-      4150 SERADD .EQ $E020      SERIAL PORTS BASE ADDR
E030-      4160 PARADD .EQ $E030      PARALLEL PORT BASE ADDR
E040-      4170 ADCADD .EQ $E040      A/D CONVERTER BASE ADDR
E050-      4180 BITINP .EQ $E050      BIT INPUT PORT BASE ADDR
E060-      4190 BITOUT .EQ $E060      BIT OUTPUT PORT BASE ADDR
E070-      4200 RNDGEN .EQ $E070      RANDOM NUMBER GENERATOR ADDRESS
E074-      4210 KBDADD .EQ $E074      KEYBOARD BASE ADDR
E074-      4220 ACRSR .EQ $E074      ACIA CNTRL/STATUS REG
E075-      4230 ARDR .EQ $E075      ACIA RECV DATA REG
E078-      4240 BELLOC .EQ $E078      BELL LOCATION
E07C-      4250 NMIRST .EQ $E07C      NMI RESET ADDR
E080-      4260 SLOT1R .EQ $E080      SLOT 1 REGISTERS
E0A0-      4270 SLOT2R .EQ $E0A0      SLOT 2 REGISTERS
E0C0-      4280 SLOT3R .EQ $E0C0      SLOT 3 REGISTERS
E0E0-      4290 SLOT4R .EQ $E0E0      SLOT 4 REGISTERS
E100-      4300 SLOT1A .EQ $E100      SLOT 1 ADDRESS SPACE
E180-      4310 SLOT2A .EQ $E180      SLOT 2 ADDRESS SPACE
E200-      4320 SLOT3A .EQ $E200      SLOT 3 ADDRESS SPACE
E280-      4330 SLOT4A .EQ $E280      SLOT 4 ADDRESS SPACE
E300-      4340 IOSPC3 .EQ $E300      EXPANSION ADDR
E380-      4350 IOSPC4 .EQ $E380      EXPANSION ADDR
      4360 *-----
      4370 * MACROS
      4380 *-----
      4390 * INCREMENT 16 BIT NUMBER
      4400          .MA INCD
      4410          INC J1
      4420          BNE :1
      4430          INC J1+1
      4440 :1

```

```
4450      .EM
4460 * DECREMENT 16 BIT NUMBER
4470      .MA DECD
4480      LDX #$FF
4490      DEC J1
4500      CPX J1
4510      BNE :1
4520      DEC J1+1
4530 :1
4540      .EM
4550 * COMPARE TWO 16 BIT NUMBERS
4560      .MA CMFD
4570      LDA J1+1
4580      CMP J2+1
4590      BNE :1
4600      LDA J1
4610      CMP J2
4620 :1
4630      .EM
```

```

1140          .IN ROMON.V1.1.PART2
1000          .OR $E400
1010          .TF TEMP.MON.OBJ
1020 *-----
1030 * MONITOR ENTRY POINTS
1040 *-----
1050 * WARM/COLD START ENTRY
1060 *-----
E400- 4C 49 F8 1070 MONITR JMP MNTR0
1080 *-----
1090 * GET MONITOR COMMAND ENTRY
1100 *-----
E403- 4C 35 FA 1110 MONCOM JMP GETCMD
1120 *-----
1130 * RETURN FROM INTERRUPT ENTRY
1140 *-----
E406- 4C C7 FF 1150 INTRET JMP RETINT
1160 *-----
1170 * MOVE A BLOCK OF MEMORY
1180 * TARGET ADDRESS= $45
1190 * START ADDRESS= $49
1200 * END ADDRESS= $47
1210 *-----
E409- 4C FE EF 1220 MOVBLK JMP MEMMOV
1230 *-----
1240 * DIASSEMBLER INDIRECT ENTRY
1250 * FIRST ADDR TO BE DISASSEMBLED
1260 * IN $45
1270 * PLACE # OF LINES TO BE DISASSEMBLED
1280 * IN 'X' REGISTER AND JSR HERE
1290 *-----
E40C- 4C 12 EA 1300 DISASM JMP DASMBL
1310 *-----
1320 * CLEAR LCD DISPLAY AND IMAGE
1330 *-----
E40F- 20 01 FA 1340 CLDISP JSR CLRIMG
E412- 4C C1 F9 1350          JMP CLRDSP
1360 *-----
1370 * CLEAR LINE BUFFER
1380 *-----
E415- 4C 61 FA 1390 CLBUFR JMP CLRBUF
1400 *-----
1410 * GET A LINE FROM INPUT DEVICE
1420 *-----
E418- 4C 43 FA 1430 GTLINE JMP GETLIN
1440 *-----
1450 * OUTPUT CHAR TO CURRENT DEVICE
1460 *-----
E41B- 4C A0 FA 1470 CHAOUT JMP OUTCHR
1480 *-----
1490 * READ CHAR FROM INPUT DEVICE
1500 *-----
E41E- 4C BC FA 1510 CHAIN  JMP RDCHAR
1520 *-----
1530 * POLL INPUT DEVICE FOR CHAR
1540 *-----
E421- 4C BF FA 1550 CHAPOL JMP INCHAR
1560 *-----
1570 * TURN OFF ALL SERIAL PORTS

```



```

1580 *-----
E424- 4C E9 FC 1590 OFFSRL JMP SEROFF
1600 *-----
1610 * OPEN SERIAL PORT IN 'A'
1620 *-----
E427- 4C 13 FD 1630 OPNSRL JMP OPENP
1640 *-----
1650 * INPUT CHAR FROM OPEN SERIAL PRT
1660 *-----
E42A- 4C 64 FD 1670 SINCH JMP INCH
1680 *-----
1690 * SEND CHAR THRU OPEN SERIAL PORT
1700 *-----
E42D- 4C B4 FD 1710 SOUTCH JMP OUTCH
1720 *-----
1730 * INITIALIZE PARALLEL COMMUNICATIONS PORT
1740 *-----
E430- 4C 50 F1 1750 INTCOM JMP INTAPL
1760 *-----
1770 * SEND CHAR TO PARALLEL COMMUNICATOR
1780 *-----
E433- 4C 6B F1 1790 SNDCOM JMP SNDCHR
1800 *-----
1810 * RECV CHAR FROM PARALLEL COMM
1820 *-----
E436- 4C 77 F1 1830 RCVCOM JMP RECKEY
1840 *-----
1850 * GET FUNCTION KEYS IN 'A'
1860 *-----
E439- 4C 47 FE 1870 FNCKEY JMP GETFNC
1880 *-----
1890 * UNPACK MESSAGE INTO BUFFER
1900 *-----
E43C- 4C 1E F7 1910 UPCMSG JMP GETMSG
1920 *-----
1930 * SEND MESSAGE TO VIOS AND/OR
1940 * TO CURRENT OUPUT DEVICE
1950 *-----
E43F- 4C ED F7 1960 OUTMSG JMP MSGXMT
1970 *-----
1980 * SET REAL TIME CLOCK FROM MEMORY
1990 *-----
E442- 4C 6C FE 2000 SETRTC JMP SETCLK
2010 *-----
2020 * GET RTC INTO MEMORY
2030 *     SEC=TIMBUF
2040 *     MIN=TIMBUF+1
2050 *     HRS=TIMBUF+2
2060 *     DAY=TIMBUF+3
2070 *     MON=TIMBUF+4
2080 *-----
E445- 4C 89 FE 2090 GETRTC JMP GETCLK
2100 *-----
2110 * SET RTC ALARM FROM MEM
2120 *-----
E448- 4C A6 FE 2130 RTCALM JMP SETALM
2140 *-----
2150 * TURN OFF RTC ALARM
2160 *-----
E44B- 4C C5 FE 2170 RTCOFF JMP ALMOFF

```

```

2180 *-----
2190 * DISPLAY HEX WORD
2200 * WORD TO BE DISPLAYED IN 'A' AND 'X' REGISTERS
2210 * 'A'=HIBYTE 'X'=LOWBYTE
2220 *-----
E44E- 4C F5 F3 2230 DSPWRD JMP PRNTAX
2240 *-----
2250 * DISPLAY HEX BYTE
2260 * BYTE IN 'A' REGISTER
2270 *-----
E451- 4C B9 F3 2280 DSPBYT JMP HEXOUT
2290 *-----
2300 * ASCII TO HEX CONVERSION
2310 * ASCII CHARS IN 'A' AND 'X'
2320 * 'A'=HIBYTE 'X'=LOWBYTE
2330 * HEX CHAR RETURNED IN 'A'
2340 *-----
E454- 4C E0 F3 2350 ASTOHX JMP ASCHEX
2360 *-----
2370 * HEX TO ASCII, HEX NYBBLE IN 'A'
2380 *-----
E457- 4C D5 F3 2390 HXTOAS JMP HEXASC
2400 *-----
2410 * BCD TO HEX CONVERSION
2420 * BCD NUMBER IN 'A'
2430 * HEX CHAR RETURNED IN 'A' REGISTER
2440 *-----
E45A- 4C 84 F3 2450 BCTOBN JMP BCDBIN
2460 *-----
2470 * SEND CHAR TO LCD DISPLAY
2480 * CHAR IN 'A' REG
2490 *-----
E45D- 4C C2 FA 2500 LCDDSP JMP LCDOUT
2510 *-----
2520 * GET A KEYBOARD CHARACTER
2530 * CHAR RETURNED IN 'A'
2540 *-----
E460- 4C 03 FC 2550 KBDCHR JMP RDKEY
2560 *-----
2570 * GET KEYBOARD CHAR IF KEY PRESSED
2580 * CHAR IN 'A' OR ZERO IF NO KEY
2590 *-----
E463- 4C 09 FC 2600 GETAKY JMP INKEY
2610 *-----
2620 * PAUSE FOR 'Y' MILLI SECONDS
2630 *-----
E466- 4C 48 FC 2640 PAUSEY JMP PAUSE
2650 *-----
2660 * POLL SERIAL PORT # IN 'A'
2670 * CHAR RETURNED IN 'A' OR ZERO IF NO CHAR
2680 *-----
E469- 4C 99 FC 2690 PSERPT JMP POLPRT
2700 *-----
2710 * SEND CHAR TO PARALLEL PRINTER PORT
2720 *-----
E46C- 4C 29 FE 2730 PRINIT JMP PRTOUT
2740 *-----
2750 * INIT HEAD
2760 *-----
E46F- 4C 62 F6 2770 INITHD JMP INIT.HEAD

```

```

2780 *-----
2790 * FIND LEFT HEAD STOP
2800 *-----
E472- 4C 67 F6 2810 LFTHDS JMP FIND.LEFT.STOP
2820 *-----
2830 * TURN OFF STEPPER MOTOR WINDINGS
2840 *-----
E475- 4C 89 F6 2850 STPOFF JMP DS.HDSTPR
2860 *-----
2870 * TURN HEAD ONE STEP
2880 *-----
E478- 4C BC F6 2890 ONESTP JMP ONE.STEP
2900 *-----
2910 * STEP STEPPER MOTOR TO POSN 'A'
2920 *-----
E47B- 4C 94 F6 2930 STEPPR JMP POSHD
2940 *-----
2950 * TURN HEAD 1.5 DEGREES
2960 *-----
E47E- 4C E5 F6 2970 TURNHD JMP STEP1.5DEGRS
2980 *-----
2990 * READ A/D CONVERTER CHANNEL IN 'Y'
3000 * VALUE RETURNED IN 'A'
3010 *-----
E481- 4C D2 FE 3020 CONVRT JMP ADCONV
3030 *-----
3040 * READ SONAR RANGERS
3050 * SONAR # TO BE READ IN 'X'
3060 *-----
E484- 4C 97 F5 3070 RDRANG JMP GET.SON.RNG
3080 *-----
3090 * ENABLE SONAR IN 'X'
3100 *-----
E487- 4C A0 F5 3110 ENBSNR JMP EN.COLSON
3120 *-----
3130 * DISABLE SONARS
3140 *-----
E48A- 4C C9 F5 3150 DISONR JMP DS.COLSON
3160 *-----
3170 * GET RANGE ON ENABLED SONAR
3180 *-----
E48D- 4C D2 F5 3190 GETRNG JMP GET.COL.RNG
3200 *-----
3210 * GET AVERAGE RANGE ON ENAB SONAR
3220 *-----
E490- 4C 15 F6 3230 AVGRNG JMP GET.AVG.RNG
3240 *-----
3250 * CARRIAGE RETURN OUT TO OUT DEV
3260 *-----
E493- 4C 67 FC 3270 CRTOUT JMP CROUT
3280 *-----
3290 * SET REAL TIME CLOCK
3300 *-----
E496- 4C D3 F2 3310 SETTIM JMP SETIM0
3320 *-----
3330 * GET TIME FROM REAL TIME CLOCK
3340 *-----
E499- 4C 00 F2 3350 GETTIM JMP GETIM0
3360 *-----
3370 * SPACE RIGHT NUMBER IN 'Y' REG

```

		3380	*-----
E49C-	4C 6E FC	3390	SPCRGT JMP SPCOUT
		3400	*-----
		3410	* TAB TO COLUMN 20 OF DISPLAY
		3420	*-----
E49F-	4C 77 FC	3430	TABRGT JMP TABOUT
		3440	*-----
		3450	* SAVE REGISTERS
		3460	*-----
E4A2-	8D 56 05	3470	SAVALL STA SAVEA
E4A5-	8E 57 05	3480	STX SAVEX
E4A8-	8C 58 05	3490	STY SAVEY
E4AB-	60	3500	RTS
		3510	*-----
		3520	* RETRIEVE ALL REGISTERS
		3530	*-----
E4AC-	AC 58 05	3540	GETALL LDY SAVEY
E4AF-	AE 57 05	3550	LDX SAVEX
E4B2-	AD 56 05	3560	LDA SAVEA
E4B5-	60	3570	RTS
		3580	*-----
		3590	* SEND CHAR TO SERIAL PORT
		3600	*-----
E4B6-	48	3610	SNDSER PHA
E4B7-	AD 6B 05	3620	LDA SERPRT
E4BA-	20 13 FD	3630	JSR OPENP
E4BD-	68	3640	PLA
E4BE-	4C B4 FD	3650	JMP OUTCH
		3660	*-----
		3670	* RECIEVE CHAR FROM SERIAL PORT
		3680	*-----
E4C1-	AD 6B 05	3690	RCVSER LDA SERPRT
E4C4-	20 13 FD	3700	JSR OPENP
E4C7-	4C 64 FD	3710	JMP INCH
		3720	*-----
		3730	* SEND CHARACTER TO PROPULSION
		3740	* CONTROLLER
		3750	*-----
E4CA-	48	3760	PRCOUT PHA
E4CB-	A9 00	3770	LDA #\$00
E4CD-	8D 6B 05	3780	STA SERPRT
E4D0-	68	3790	PLA
E4D1-	4C B6 E4	3800	JMP SNDSER
		3810	*-----
		3820	* RECIEVE CHAR FROM PROCON
		3830	*-----
E4D4-	A9 00	3840	PRCIN LDA #\$00
E4D6-	8D 6B 05	3850	STA SERPRT
E4D9-	4C C1 E4	3860	JMP RCVSER
		3870	*-----
		3880	* RECIEVE CHAR FROM VIOS
		3890	*-----
E4DC-	48	3900	VIOSOT PHA
E4DD-	A9 01	3910	LDA #\$01
E4DF-	8D 6B 05	3920	STA SERPRT
E4E2-	68	3930	PLA
E4E3-	4C B6 E4	3940	JMP SNDSER
		3950	*-----
		3960	* SEND CHAR TO VIOS
		3970	*-----

```

E4E6- A9 01      3980 VIOSIN LDA #$01
E4E8- 8D 6B 05   3990          STA SERPRT
E4EB- 4C C1 E4   4000          JMP RCVSER
                     4010 *-----
                     4020 * SEND CHAR TO RF MODEM
                     4030 *-----
E4EE- 20 A2 E4   4040 RFMOUT JSR SAVALL
E4F1- 8D 6B 05   4050          STA SERPRT
E4F4- A9 01      4060          LDA #$01
E4F6- A2 DF      4070          LDX /RFCHOT
E4F8- A0 FD      4080          LDY #RFCHOT
E4FA- 20 49 E5   4090          JSR BANKSW
E4FD- 4C AC E4   4100          JMP BETALL LEDOFF ?
                     4110 *-----
                     4120 * RECIEVE CHAR FROM RF MODEM
                     4130 *-----
E500- 20 A2 E4   4140 RFMIN  JSR SAVALL
E503- A9 01      4150          LDA #$01
E505- A2 DF      4160          LDX /DATIN
E507- A0 FA      4170          LDY #DATIN
E509- 20 49 E5   4180          JSR BANKSW
E50C- 20 AC E4   4190          JSR GETALL
E50F- AD 6B 05   4200          LDA SERPRT
E512- 60         4210          RTS
                     4220 *-----
                     4230 * SEND CHAR TO EXTERNAL SERIAL
                     4240 *-----
E513- 48         4250 EXTOUT PHA
E514- A9 03      4260          LDA #$03
E516- 8D 6B 05   4270          STA SERPRT
E519- 68         4280          PLA
E51A- 4C B6 E4   4290          JMP SNDSER
                     4300 *-----
                     4310 * RECV CHAR FROM EXTERNAL PORT
                     4320 *-----
E51D- A9 03      4330 EXTIN  LDA #$03
E51F- 8D 6B 05   4340          STA SERPRT
E522- 4C C1 E4   4350          JMP RCVSER
                     4360 *-----
                     4370 * SEND CHAR TO EXPANSION SERIAL 1
                     4380 *-----
E525- 48         4390 EXP1OT PHA
E526- A9 04      4400          LDA #$04
E528- 8D 6B 05   4410          STA SERPRT
E52B- 68         4420          PLA
E52C- 4C B6 E4   4430          JMP SNDSER
                     4440 *-----
                     4450 * RECIEVE CHAR FROM EXP PORT 1
                     4460 *-----
E52F- A9 04      4470 EXP1IN LDA #$04
E531- 8D 6B 05   4480          STA SERPRT
E534- 4C C1 E4   4490          JMP RCVSER
                     4500 *-----
                     4510 * SEND CHAR TO EXP PORT 2
                     4520 *-----
E537- 48         4530 EXP2OT PHA
E538- A9 05      4540          LDA #$05
E53A- 8D 6B 05   4550          STA SERPRT
E53D- 68         4560          PLA
E53E- 4C B6 E4   4570          JMP SNDSER

```

```

4580 *-----
4590 * RECIEVE CHAR FROM EXP PORT 2
4600 *-----
E541- A9 05 4610 EXP2IN LDA #05
E543- 8D 6B 05 4620          STA SERPRT
E546- 4C C1 E4 4630          JMP RCVSER
4640 *-----
4650 * BANK SWITCHING ROUTINES
4660 * TO JUMP TO A SUBROUTINE IN BANK # X;
4670 * PUT LOW BYTE OF SUBROUTINE IN 'Y'
4680 * PUT HI BYTE OF SUBROUTINE IN 'X'
4690 * PUT TARGET BANK NUMBER IN 'A'
4700 * AND JSR TO BANKSW.
4710 * AFTER SUBROUTINE COMPLETED PROGRAM
4720 * FLOW RETURNS TO CORRECT BANK.
4730 * 'A','X',AND 'Y' REGISTERS DESTROYED
4740 * TO PASS PARAMETERS USE MEMORY INSTEAD OF THE STACK.
4750 *-----
E549- 85 31 4760 BANKSW STA TEMP00    SAVE TARGET BANK
E54B- AD 53 05 4770          LDA CURBNK    SAVE CURRENT BANK
E54E- 48      4780          PHA          ON STACK
E54F- A5 31 4790          LDA TEMP00    GET TARGET BANK
E551- 0A      4800          ASL
E552- 0A      4810          ASL
E553- 29 1C 4820          AND #1C      MASK UNWANTED BITS
E555- 8D 53 05 4830          STA CURBNK    MAKE IT CURRENT BANK
E558- AD 61 E0 4840          LDA BITOUT+1
E55B- 29 E3 4850          AND #E3
E55D- 0D 53 05 4860          ORA CURBNK
E560- 8D 61 E0 4870          STA BITOUT+1 PHYSICALLY SWITCH ROMS
E563- A9 80 4880          LDA #SWBACK SET UP RETURN ADDRESS
E565- 3A      4890          DEC
E566- 85 31 4900          STA TEMP00
E568- C9 FF 4910          CMP #FF
E56A- D0 05 4920          BNE BNKSW1
E56C- A9 E5 4930          LDA /SWBACK
E56E- 3A      4940          DEC
E56F- 80 02 4950          BRA BNKSW2
E571- A9 E5 4960 BNKSW1 LDA /SWBACK
E573- 48      4970 BNKSW2 PHA
E574- A5 31 4980          LDA TEMP00
E576- 48      4990          PHA
E577- 88      5000          DEY
E578- C0 FF 5010          CFY #FF
E57A- D0 01 5020          BNE BNKSW3
E57C- CA      5030          DEX
E57D- DA      5040 BNKSW3 PHX
E57E- 5A      5050          PHY
E57F- 60      5060          RTS
E580- 68      5070 SWBACK PLA      GO TO ROUTINE IN TARGET BANK
                                         AFTERWARDS RESTORE CURRENT BANK
E581- 8D 53 05 5080          STA CURBNK
E584- AD 61 E0 5090          LDA BITOUT+1
E587- 29 E3 5100          AND #E3
E589- 0D 53 05 5110          ORA CURBNK
E58C- 8D 61 E0 5120          STA BITOUT+1 AND SWITCH ROMS BACK
E58F- 60      5130          RTS      THEN RETURN
5140 *-----
5150 * TO JUMP INTO A ROUTINE IN ANOTHER BANK
5160 * PUT BANK # IN 'A'
5170 * PUT LO ADDRESS IN 'Y'

```

```

5180 * PUT HI ADDRESS IN 'X'
5190 * AND JMP SWBANK.
5200 *-----
E590- 0A      5210 SWBANK ASL
E591- 0A      5220      ASL
E592- 29 1C    5230      AND ##1C
E594- 8D 53 05 5240      STA CURBNK      SET CURRENT BANK TO NEW VALUE
E597- AD 61 E0 5250      LDA BITOUT+1
E59A- 29 E3    5260      AND ##E3
E59C- 0D 53 05 5270      ORA CURBNK
E59F- 8D 61 E0 5280      STA BITOUT+1 AND PHYSICALLY SWITCH ROMS
E5A2- 8C 51 05 5290      STY BANKAD
E5A5- 8E 52 05 5300      STX BANKAD+1
E5A8- 6C 51 05 5310      JMP (BANKAD)
5320 *-----
5330 * TO SAVE A VALUE IN BANK 'X' RAM
5340 * VALUE IN 'A' ADDRESS IN ADRPTR
5350 * AND ADRPTR+1
5360 *-----
E5AB- DA      5370 SAVBNK PHX      SAVE VALUE IN BANK 0
E5AC- A2 00    5380      LDX ##00
E5AE- 20 B3 E5 5390      JSR SAVXBK
E5B1- FA      5400      PLX
E5B2- 60      5410      RTS
E5B3- 48      5420 SAVXBK PHA      SAVE VALUE
E5B4- AD 61 E0 5430      LDA BITOUT+1
E5B7- 29 E3    5440      AND ##E3
E5B9- 85 31    5450      STA TEMP00
E5BB- 8A      5460      TXA
E5BC- 0A      5470      ASL
E5BD- 0A      5480      ASL
E5BE- 05 31    5490      ORA TEMP00
E5C0- 8D 61 E0 5500      STA BITOUT+1
E5C3- 68      5510      PLA
E5C4- 92 39    5520      STA (ADRPTR)
E5C6- 48      5530      PHA
E5C7- AD 61 E0 5540      LDA BITOUT+1
E5CA- 29 E3    5550      AND ##E3
E5CC- 0D 53 05 5560      ORA CURBNK
E5CF- 8D 61 E0 5570      STA BITOUT+1
E5D2- 68      5580      PLA
E5D3- 60      5590      RTS
5600 *-----
5610 * LOAD VALUE FROM BANK #0 RAM
5620 * ADDRESS IN ADRPTR AND ADRPTR+1
5630 *-----
E5D4- DA      5640 LODBNK PHX
E5D5- A2 00    5650      LDX ##00
E5D7- 20 DE E5 5660      JSR LODXBK
E5DA- FA      5670      PLX
E5DB- 48      5680      PHA
E5DC- 68      5690      PLA
E5DD- 60      5700      RTS
E5DE- AD 61 E0 5710 LODXBK LDA BITOUT+1
E5E1- 29 E3    5720      AND ##E3
E5E3- 85 31    5730      STA TEMP00
E5E5- 8A      5740      TXA
E5E6- 0A      5750      ASL
E5E7- 0A      5760      ASL
E5E8- 05 31    5770      ORA TEMP00

```

E5EA-	8D 61 E0	5780	STA BITOUT+1	
E5ED-	B2 39	5790	LDA (ADRPTR)	
E5EF-	48	5800	PHA	
E5F0-	AD 61 E0	5810	LDA BITOUT+1	
E5F3-	29 E3	5820	AND #E3	
E5F5-	0D 53 05	5830	ORA CURBNK	
E5F8-	8D 61 E0	5840	STA BITOUT+1	
E5FB-	68	5850	PLA	
E5FC-	60	5860	RTS	
		5870	*-----	
		5880	* CLOCK SPEED SELECTION ROUTINES	
		5890	* SELECTS HIGH CLOCK SPEED OF	
		5900	* 2.4576 MHZ AND LOW CLOCK	
		5910	* SPEED OF 1.2288 MHZ	
		5920	*	
		5930	* FOLLOWING DEVICES MUST BE	
		5940	* ACCESSED AT LOW SPEED	
		5950	* REAL TIME CLOCK	
		5960	* COUNTER/TIMER	
		5970	* ANALOG/DIGITAL CONVERTER	
		5980	* ACIA	
		5990	*-----	
E5FD-	48	6000	SETLSP PHA	SET CLOCK TO LOW SPEED
E5FE-	77 51	6010	RMB 7,CPUFLG	CLEAR THE SPEED FLAG
E600-	A9 80	6020	LDA #BIT7	
E602-	1C 61 E0	6030	TRB BITOUT+1	CHANGE SPEED
E605-	68	6040	PLA	
E606-	60	6050	RTS	
E607-	48	6060	SETHSP PHA	SET CLOCK TO HIGH SPEED
E608-	F7 51	6070	SMB 7,CPUFLG	CHANGE THE FLAG
E60A-	A9 80	6080	LDA #BIT7	
E60C-	0C 61 E0	6090	TSB BITOUT+1	
E60F-	68	6100	PLA	
E610-	60	6110	RTS	
E611-	48	6120	LOSPD PHA	
E612-	A9 80	6130	LDA #BIT7	
E614-	1C 61 E0	6140	TRB BITOUT+1	
E617-	68	6150	PLA	
E618-	60	6160	RTS	
E619-	48	6170	HISPD PHA	
E61A-	A9 80	6180	LDA #BIT7	
E61C-	0C 61 E0	6190	TSB BITOUT+1	
E61F-	68	6200	PLA	
E620-	60	6210	RTS	
E621-	48	6220	RESMSP PHA	
E622-	24 51	6230	BIT CPUFLG	
E624-	30 07	6240	BMI RSMSP1	
E626-	A9 80	6250	LDA #BIT7	
E628-	1C 61 E0	6260	TRB BITOUT+1	
E62B-	68	6270	PLA	
E62C-	60	6280	RTS	
E62D-	A9 80	6290	RSMSP1 LDA #BIT7	
E62F-	0C 61 E0	6300	TSB BITOUT+1	
E632-	68	6310	PLA	
E633-	60	6320	RTS	


```

1160          .IN ROMON.V1.1.PART3
1000 *-----
1010 *
1020 * SINE TABLE
1030 *

E634- 00 04 08
E637- 0D 12 16
E63A- 1B 1F 24
E63D- 28          1040 TRG      .HS 0004080D12161B1F2428
E63E- 2C 31 35
E641- 3A 3E 42
E644- 47 4B 4F
E647- 53          1050          .HS 2C31353A3E42474B4F53
E648- 58 5C 60
E64B- 64 68 6C
E64E- 70 74 78
E651- 7C          1060          .HS 585C6064686C7074787C
E652- 80 84 88
E655- 8B 8F 93
E658- 96 9A 9E
E65B- A1          1070          .HS 8084888B8F93969A9EA1
E65C- A5 A0 AB
E65F- AF B2 B5
E662- B8 BB BE
E665- C1          1080          .HS A5A0ABAFB2B5B8BBBEC1
E666- C4 C7 CA
E669- CC CF D2
E66C- D4 D7 D9
E66F- DB          1090          .HS C4C7CACC CFD2D4D7D9DB
E670- DE E0 E2
E673- E4 E6 E8
E676- EA EC ED
E679- EF          1100          .HS DEE0E2E4E6E8EAEC EDEF
E67A- F1 F2 F3
E67D- F5 F6 F7
E680- F8 F9 FA
E683- FB          1110          .HS F1F2F3F5F6F7F8F9FAFB
E684- FC FD FE
E687- FE FF FF
E68A- FF FF FF
E68D- FF          1120          .HS FCFDFE FEFFFFFFFFFFFFFFFF
1130 *-----
1140 * CLEAR: AC=0
1150 *-----
E68E- A0 00          1160 CLR      LDY #000
E690- 84 62          1170          STY ACL
E692- 84 63          1180          STY ACM
E694- 84 64          1190          STY ACH
E696- 60          1200          RTS
1210 *-----
1220 * INCREMENT: AC=AC+1
1230 *-----
E697- 20 90 E9      1240 INC.1   JSR TMA
E69A- E6 63          1250 INC.AC  INC ACM
E69C- D0 02          1260          BNE ICO
E69E- E6 64          1270          INC ACH
E6A0- 60          1280 ICO     RTS
1290 *-----
1300 * DECREMENT: AC=AC-1

```

			1310	*-----	
E6A1-	20	90	E9	1320	DEC.1 JSR TMA
E6A4-	38			1330	DEC.AC SEC
E6A5-	A5	63		1340	LDA ACM
E6A7-	E9	01		1350	SBC ##01
E6A9-	85	63		1360	STA ACM
E6AB-	B0	02		1370	BCS DCO
E6AD-	C6	64		1380	DEC ACH
E6AF-	60			1390	DCO RTS
				1400	*-----
				1410	* INTEGER: AC=INT(AC+.5)
				1420	*-----
				1430	*
				1440	* ROUNDS INSTEAD OF TRUNCATES
				1450	*
E6B0-	20	90	E9	1460	INT.1 JSR TMA
E6B3-	24	62		1470	INT BIT ACL
E6B5-	10	03		1480	BPL IND
E6B7-	20	9A	E6	1490	TNO JSR INC.AC
E6BA-	A2	00		1500	IND LDX ##00
E6BC-	86	62		1510	STX ACL
E6BE-	60			1520	TOK RTS
				1530	*-----
				1540	* TRUNCATE: AC=INT(AC)
				1550	*-----
				1560	*
				1570	* TRUNCATES AS PER BASIC "INT"
				1580	*
E6BF-	20	90	E9	1590	TNC.1 JSR TMA
E6C2-	A5	62		1600	TNC LDA ACL
E6C4-	F0	F8		1610	BEQ TOK
E6C6-	24	64		1620	BIT ACH
E6C8-	10	F0		1630	BPL IND
E6CA-	30	EB		1640	BMI TNO
				1650	*-----
				1660	* ADD: AC=AC+RC
				1670	*-----
E6CC-	20	90	E9	1680	ADD.2 JSR TMA
E6CF-	20	A0	E9	1690	ADD.1 JSR TMR
E6D2-	A2	FD		1700	ADD LDX ##FD
E6D4-	18			1710	CLC
E6D5-	B5	65		1720	ADL LDA SCL,X
E6D7-	75	5F		1730	ADC TMP,X
E6D9-	95	65		1740	STA SCL,X
E6DB-	E8			1750	INX
E6DC-	D0	F7		1760	BNE ADL
E6DE-	60			1770	RTS
				1780	*-----
				1790	* SUBTRACT: AC=AC-RC
				1800	*-----
E6DF-	20	90	E9	1810	SUB.2 JSR TMA
E6E2-	20	A0	E9	1820	SUB.1 JSR TMR
E6E5-	A2	FD		1830	SUB LDX ##FD
E6E7-	38			1840	SEC
E6E8-	B5	65		1850	SBL LDA SCL,X
E6EA-	F5	5F		1860	SBC TMP,X
E6EC-	95	65		1870	STA SCL,X
E6EE-	E8			1880	INX
E6EF-	D0	F7		1890	BNE SBL
E6F1-	60			1900	RTS

```

1910 *-----
1920 * SIGN: A REG = SGN(AC)
1930 *-----
1940 *
1950 * A REG CONDITIONED BY SIGN:
1960 *
1970 * A=00 FOR AC=0
1980 * A=01 FOR AC>0
1990 * A=FF FOR AC<0
2000 *
E6F2- 20 90 E9 2010 SGN.1 JSR TMA
E6F5- A5 64 2020 SGN LDA ACH
E6F7- 30 33 2030 BMI CMI
E6F9- D0 08 2040 BNE CPL
E6FB- A5 63 2050 LDA ACM
E6FD- D0 04 2060 BNE CPL
E6FF- A5 62 2070 LDA ACL
E701- F0 24 2080 BEQ CEQ
E703- A9 01 2090 CPL LDA #$01
E705- 60 2100 RTS
2110 *-----
2120 * COMPARE: A REG = (AC)CMP(RC)
2130 *-----
2140 *
2150 * A REG CONDITIONED BY COMPARE:
2160 *
2170 * A=00 FOR AC=RC
2180 * A=01 FOR AC>RC
2190 * A=FF FOR AC<RC
2200 *
E706- 20 90 E9 2210 CMP.2 JSR TMA
E709- 20 A0 E9 2220 CMP.1 JSR TMR
E70C- A2 02 2230 CMP.AC LDX #$02
E70E- A5 64 2240 LDA ACH
E710- 30 06 2250 BMI CMX
E712- A4 5E 2260 LDY RCH
E714- 10 0A 2270 BPL CLQ
E716- 30 EB 2280 BMI CPL
E718- A4 5E 2290 CMX LDY RCH
E71A- 30 04 2300 BMI CLQ
E71C- 10 0E 2310 BPL CMI
E71E- B5 62 2320 CLF LDA ACL,X
E720- D5 5C 2330 CLQ CMP RCL,X
E722- D0 06 2340 BNE CNE
E724- CA 2350 DEX
E725- 10 F7 2360 BPL CLF
E727- A9 00 2370 CEQ LDA #$00
E729- 60 2380 RTS
2390 *
E72A- B0 D7 2400 CNE BCS CPL
E72C- A9 FF 2410 CMI LDA #$FF
E72E- 60 2420 RTS
2430 *-----
2440 * ABSOLUTE VALUE: AC=ABS(AC)
2450 * CHANGE SIGN: AC=-AC
2460 *-----
E72F- 20 90 E9 2470 CHG.1 JSR TMA
E732- 4C 3C E7 2480 JMP CHG
2490 *
2500 * ABSOLUTE VALUE

```

```

2510 *
E735- 20 90 E9 2520 ABS.1 JSR TMA
E738- 24 64 2530 ABS BIT ACH
E73A- 10 0E 2540 BPL CHO
2550 *
2560 * CHANGE SIGN
2570 *
E73C- A2 FD 2580 CHG LDX ##FD
E73E- 38 2590 SEC
E73F- B5 65 2600 CHL LDA SCL,X
E741- 49 FF 2610 EOR ##FF
E743- 69 00 2620 ADC ##00
E745- 95 65 2630 STA SCL,X
E747- EB 2640 INX
E748- D0 F5 2650 BNE CHL
E74A- 60 2660 CHO RTS
2670 *-----
2680 * MULTIPLY: AC=AC*RC
2690 *-----
E74B- 20 90 E9 2700 MUL.2 JSR TMA
E74E- 20 A0 E9 2710 MUL.1 JSR TMR
E751- 20 34 E8 2720 MUL JSR CKS
E754- A5 62 2730 MUL1 LDA ACL
E756- 85 61 2740 STA ACX
E758- A5 63 2750 LDA ACM
E75A- 85 62 2760 STA ACL
E75C- A5 64 2770 LDA ACH
E75E- 85 63 2780 STA ACM
E760- A0 18 2790 LDY ##18
E762- A9 00 2800 LDA ##00
E764- 85 64 2810 STA ACH
E766- 85 65 2820 STA SCL
E768- 85 66 2830 STA SCM
E76A- A5 61 2840 MSHL LDA ACX
E76C- 4A 2850 LSR
E76D- 90 13 2860 BCC MROR
E76F- 18 2870 CLC
E770- A5 64 2880 LDA ACH
E772- 65 5C 2890 ADC RCL
E774- 85 64 2900 STA ACH
E776- A5 65 2910 LDA SCL
E778- 65 5D 2920 ADC RCM
E77A- 85 65 2930 STA SCL
E77C- A5 66 2940 LDA SCM
E77E- 65 5E 2950 ADC RCH
E780- 85 66 2960 STA SCM
E782- 66 66 2970 MROR ROR SCM
E784- 66 65 2980 ROR SCL
E786- 66 64 2990 ROR ACH
E788- 66 63 3000 ROR ACM
E78A- 66 62 3010 ROR ACL
E78C- 66 61 3020 ROR ACX
E78E- 88 3030 DEY
E78F- D0 D9 3040 BNE MSHL
E791- A6 5A 3050 LDX FLG
E793- F0 03 3060 BEQ MLO
E795- 20 3C E7 3070 JSR CHG
E798- 60 3080 MLO RTS
3090 *-----
3100 * DIVIDE: AC=AC/RC

```

```

3110 *-----
E799- 20 90 E9 3120 DIV.2 JSR TMA
E79C- 20 A0 E9 3130 DIV.1 JSR TMR
E79F- 20 34 E8 3140 DIV JSR CKS
E7A2- A5 64 3150 DIV1 LDA ACH
E7A4- 85 65 3160 STA SCL
E7A6- A5 63 3170 LDA ACM
E7A8- 85 64 3180 STA ACH
E7AA- A5 62 3190 LDA ACL
E7AC- 85 63 3200 STA ACM
E7AE- A0 18 3210 DIV2 LDY ##18
E7B0- 84 60 3220 STY CNT
E7B2- A9 00 3230 LDA ##00
E7B4- 85 62 3240 STA ACL
E7B6- 85 66 3250 STA SCM
E7B8- 85 67 3260 STA SCH
E7BA- 06 62 3270 DIVL ASL ACL
E7BC- 26 63 3280 ROL ACM
E7BE- 26 64 3290 ROL ACH
E7C0- 26 65 3300 ROL SCL
E7C2- 26 66 3310 ROL SCM
E7C4- 26 67 3320 ROL SCH
E7C6- 38 3330 SEC
E7C7- A5 65 3340 LDA SCL
E7C9- E5 5C 3350 SBC RCL
E7CB- AA 3360 TAX
E7CC- A5 66 3370 LDA SCM
E7CE- E5 5D 3380 SBC RCM
E7D0- A8 3390 TAY
E7D1- A5 67 3400 LDA SCH
E7D3- E5 5E 3410 SBC RCH
E7D5- 90 08 3420 BCC DIV5
E7D7- 86 65 3430 STX SCL
E7D9- 84 66 3440 STY SCM
E7DB- 85 67 3450 STA SCH
E7DD- E6 62 3460 INC ACL
E7DF- C6 60 3470 DIV5 DEC CNT
E7E1- D0 D7 3480 BNE DIVL
E7E3- A6 5A 3490 LDX FLG
E7E5- F0 03 3500 BEQ DVO
E7E7- 20 3C E7 3510 JSR CHG
E7EA- 60 3520 DVO RTS
3530 *-----
3540 * INVERT: AC=1/AC
3550 *-----
E7EB- 20 90 E9 3560 INV.1 JSR TMA
E7EE- 20 BA E9 3570 INV JSR TAR
E7F1- A9 00 3580 LDA ##00
E7F3- 85 63 3590 STA ACM
E7F5- 85 65 3600 STA SCL
E7F7- A9 01 3610 LDA ##01
E7F9- 85 64 3620 STA ACH
E7FB- 20 34 E8 3630 JSR CKS
E7FE- 4C AE E7 3640 JMP DIV2
3650 *-----
3660 * SQUARE ROOT: AC=SQR(AC)
3670 *-----
3680 *
3690 * NEWTON-RAPHSON SQUARE ROOT
3700 *

```

E801-	20	90	E9	3710	SQR.1	JSR TMA
E804-	20	C4	E9	3720	SQR	JSR TAT
E807-	A9	10		3730		LDA ##10
E809-	85	5F		3740		STA TMP
				3750	*	
				3760	*	INITIAL GUESS = 0
				3770	*	
E80B-	A9	00		3780		LDA ##00
E80D-	85	5A		3790		STA FLG
E80F-	85	5C		3800		STA RCL
E811-	85	5E		3810		STA RCH
E813-	85	5D		3820		STA RCM
				3830	*	
				3840	*	ADD GUESS TO ARG/GUESS
				3850	*	AND DIVIDE BY TWO
				3860	*	
E815-	20	D2	E6	3870	SQL	JSR ADD
E818-	18			3880		CLC
E819-	66	64		3890		ROR ACH
E81B-	66	63		3900		ROR ACM
E81D-	66	62		3910		ROR ACL
				3920	*	
				3930	*	GUESS=OLD GUESS?
				3940	*	
E81F-	20	0C	E7	3950		JSR CMP.AC
E822-	F0	0F		3960		BEQ SQ0
E824-	C6	5F		3970		DEC TMP
E826-	F0	0B		3980		BEQ SQ0
				3990	*	
				4000	*	STORE NEW GUESS
				4010	*	
E828-	20	BA	E9	4020		JSR TAR
				4030	*	
				4040	*	FETCH ARGUMENT AND
				4050	*	DIVIDE BY GUESS
				4060	*	
E82B-	20	D8	E9	4070		JSR TTA
E82E-	20	A2	E7	4080		JSR DIV1
E831-	F0	E2		4090		BEQ SQL
E833-	60			4100	SQ0	RTS
				4110	*	-----
				4120	*	CHECK SIGN ROUTINE
				4130	*	FOR MULTIPLY AND DIVIDE
				4140	*	-----
E834-	A0	00		4150	CKS	LDY ##00
E836-	24	64		4160		BIT ACH
E838-	10	04		4170		BPL CKA
E83A-	20	3C	E7	4180		JSR CHG
E83D-	88			4190		DEY
E83E-	24	5E		4200	CKA	BIT RCH
E840-	10	0F		4210		BPL CKB
E842-	C8			4220		INY
E843-	A2	FD		4230	CGR	LDX ##FD
E845-	38			4240		SEC
E846-	B5	5F		4250	CKL	LDA TMP,X
E848-	49	FF		4260		EOR ##FF
E84A-	69	00		4270		ADC ##00
E84C-	95	5F		4280		STA TMP,X
E84E-	E8			4290		INX
E84F-	D0	F5		4300		BNE CKL

E851-	84	5A	4310	CKB	STY	FLG	
E853-	60		4320		RTS		
			4330	*-----			
			4340	* FIX ANGLE ROUTINE			
			4350	* INSURES 0 <= AC < 360			
			4360	*-----			
			4370	*			
			4380	* RC=360			
			4390	*			
E854-	A9	00	4400	FXA	LDA	##00	
E856-	A2	68	4410		LDX	##68	
E858-	A0	01	4420		LDY	##01	
E85A-	85	5C	4430		STA	RCL	
E85C-	86	5D	4440		STX	RCM	
E85E-	84	5E	4450		STY	RCH	
			4460	*			
			4470	* ANGLE NEGATIVE?			
			4480	*			
E860-	A5	64	4490		LDA	ACH	
E862-	10	0A	4500		BPL	FXB	
			4510	*			
			4520	* FIX NEG ANGLE, ADD 360 UNTIL AC>=0			
			4530	*			
E864-	20	D2 E6	4540	FXN	JSR	ADD	
E867-	AA		4550		TAX		
E868-	30	FA	4560		BMI	FXN	
E86A-	60		4570		RTS		
			4580	*			
			4590	* FIX POS ANGLE, SUB 360 UNTIL AC<360			
			4600	*			
E86B-	20	E5 E6	4610	FXP	JSR	SUB	
E86E-	C9	01	4620	FXB	CMP	##01	
E870-	90	08	4630		BCC	FXD	
E872-	D0	F7	4640		BNE	FXP	
E874-	A5	63	4650		LDA	ACM	
E876-	C9	68	4660		CMP	##68	
E878-	B0	F1	4670		BCS	FXP	
			4680	*			
			4690	* ANGLE OK			
			4700	*			
E87A-	60		4710	FXD	RTS		
			4720	*-----			
			4730	* SINE: AC=SIN(AC)			
			4740	*-----			
E87B-	20	90 E9	4750	SIN.1	JSR	TMA	
E87E-	20	54 E8	4760	SIN	JSR	FXA	
E881-	20	B3 E6	4770		JSR	INT	
E884-	A0	00	4780		LDY	##00	
E886-	84	5A	4790		STY	FLG	SIGN +
			4800	*			
			4810	* REDUCE ANGLE TO <= 90			
			4820	*			
E888-	38		4830		SEC		
E889-	A5	64	4840		LDA	ACH	
E88B-	D0	06	4850		BNE	SNA	AC>255
E88D-	A5	63	4860		LDA	ACM	
E88F-	C9	B5	4870		CMP	##B5	
E891-	90	08	4880		BCC	SNB	AC<=180
E893-	A9	68	4890	SNA	LDA	##68	
E895-	E5	63	4900		SBC	ACM	

E897-	85	63		4910	STA	ACM	
E899-	C6	5A		4920	DEC	FLG	SIGN -
E89B-	C9	5B		4930	SNB	CMP	##5B
E89D-	90	04		4940		BCC	SNG AC<=90
E89F-	A9	B4		4950		LDA	##B4
E8A1-	E5	63		4960		SBC	ACM
				4970	*		
				4980	*	ANGLE IN A, GET	AC=SIN(A)
				4990	*		
E8A3-	20	8E	E6	5000	SNG	JSR	CLR
E8A6-	C9	57		5010		CMP	##57
E8AB-	90	04		5020		BCC	SNT
E8AA-	E6	63		5030		INC	ACM A>86
E8AC-	10	06		5040		BPL	SNS
E8AE-	AA			5050	SNT	TAX	
E8AF-	BD	34	E6	5060		LDA	TRG,X
E8B2-	85	62		5070		STA	ACL
				5080	*		
				5090	*	NEG	VALUE?
				5100	*		
E8B4-	24	5A		5110	SNS	BIT	FLG
E8B6-	10	03		5120		BPL	SNO
E8B8-	20	3C	E7	5130		JSR	CHG
E8BB-	60			5140	SNO	RTS	
				5150	*	-----	
				5160	*	COSINE: AC=	COS(AC)
				5170	*	-----	
E8BC-	20	90	E9	5180	COS.1	JSR	TMA
E8BF-	20	21	E9	5190	COS	JSR	CFA 90-A
E8C2-	4C	7E	E8	5200		JMP	SIN
				5210	*	-----	
				5220	*	TANGENT: AC=	TAN(AC)
				5230	*	-----	
E8C5-	20	90	E9	5240	TAN.1	JSR	TMA
E8C8-	20	C4	E9	5250	TAN	JSR	TAT
E8CB-	20	BF	E8	5260		JSR	COS
E8CE-	20	F5	E6	5270		JSR	SGN
E8D1-	D0	04		5280		BNE	TN0
E8D3-	A9	02		5290		LDA	##02
E8D5-	85	62		5300		STA	ACL
E8D7-	20	CE	E9	5310	TN0	JSR	TA0
E8DA-	20	D8	E9	5320		JSR	TTA
E8DD-	20	7E	E8	5330		JSR	SIN
E8E0-	20	F6	E9	5340		JSR	TQR
E8E3-	20	9F	E7	5350		JSR	DIV
E8E6-	60			5360		RTS	
				5370	*	-----	
				5380	*	ARCSINE: AC=	ASN(AC)
				5390	*	-----	
E8E7-	20	90	E9	5400	ASN.1	JSR	TMA
E8EA-	A0	00		5410	ASN	LDY	##00
E8EC-	B4	5A		5420		STY	FLG
E8EE-	A6	64		5430	ASN1	LDX	ACH
E8F0-	10	06		5440		BPL	ASC
E8F2-	20	3C	E7	5450		JSR	CHG
E8F5-	C6	5A		5460		DEC	FLG
E8F7-	AA			5470		TAX	
E8F8-	D0	04		5480	ASC	BNE	AOV
E8FA-	C6	63		5490		DEC	ACM
E8FC-	30	04		5500		BMI	ASG

E8FE-	A2	5A		5510	ADV	LDX	##5A	
E900-	10	0C		5520		BPL	ASF	
E902-	A2	57		5530	ASG	LDX	##57	
E904-	A5	62		5540		LDA	ACL	
E906-	CA			5550	ASL1	DEX		
E907-	F0	05		5560		BEQ	ASF	
E909-	DD	34	E6	5570		CMF	TRG,X	
E90C-	90	F8		5580		BCC	ASL1	
E90E-	20	8E	E6	5590	ASF	JSR	CLR	
E911-	86	63		5600		STX	ACM	
E913-	24	5A		5610		BIT	FLG	
E915-	10	03		5620		BPL	ASO	
E917-	20	3C	E7	5630		JSR	CHG	
E91A-	60			5640	ASO	RTS		
				5650	*-----			
				5660	* ARCCOSINE: AC=ACS(AC)			
				5670	*-----			
E91B-	20	90	E9	5680	ACS.1	JSR	TMA	
E91E-	20	EA	E8	5690	ACS	JSR	ASN	AC=ASN
E921-	38			5700	CPA	SEC		
E922-	A9	00		5710		LDA	##00	
E924-	E5	62		5720		SBC	ACL	
E926-	85	62		5730		STA	ACL	
E928-	A9	5A		5740		LDA	##5A	
E92A-	E5	63		5750		SBC	ACM	
E92C-	85	63		5760		STA	ACM	
E92E-	A9	00		5770		LDA	##00	
E930-	E5	64		5780		SBC	ACH	
E932-	85	64		5790		STA	ACH	
E934-	60			5800		RTS		
				5810	*-----			
				5820	* ARCTANGENT: AC=ATN(AC)			
				5830	*-----			
E935-	20	90	E9	5840	ATN.1	JSR	TMA	
E938-	20	F5	E6	5850	ATN	JSR	SGN	
E93B-	85	5B		5860		STA	FLH	
E93D-	85	5A		5870		STA	FLG	
E93F-	10	03		5880		BPL	ATP	
E941-	20	3C	E7	5890		JSR	CHG	
E944-	A5	64		5900	ATP	LDA	ACH	
E946-	D0	B6		5910		BNE	ADV	
E948-	A5	63		5920		LDA	ACM	
E94A-	30	B2		5930		BMI	ADV	
E94C-	20	CE	E9	5940		JSR	TAQ	
E94F-	20	BA	E9	5950		JSR	TAR	
E952-	A2	00		5960		LDX	##00	
E954-	86	5A		5970		STX	FLG	
E956-	20	54	E7	5980		JSR	MUL1	
E959-	20	9A	E6	5990		JSR	INC.AC	
E95C-	20	04	E8	6000		JSR	SQR	
E95F-	A5	55		6010		LDA	QCM	
E961-	D0	0E		6020		BNE	ATM	
E963-	20	BA	E9	6030		JSR	TAR	
E966-	20	EC	E9	6040		JSR	TQA	
E969-	20	A2	E7	6050		JSR	DIV1	
E96C-	20	EE	E8	6060		JSR	ASN1	
E96F-	10	06		6070		BPL	ATF	
E971-	20	EE	E7	6080	ATM	JSR	INV	
E974-	20	1E	E9	6090		JSR	ACS	
E977-	24	5B		6100	ATF	BIT	FLH	

E979-	10 03	6110	BPL ATO
E97B-	20 3C E7	6120	JSR CHG
E97E-	60	6130	ATO RTS
		6140	*-----
		6150	* LET: VAR1=VAR2
		6160	*-----
		6170	*
		6180	* ENTRY: POINTER TO VAR#1 IN Y AND POINTER TO VAR#2 IN X
		6190	*
E97F-	A9 03	6200	LET.2 LDA #03
E981-	85 60	6210	STA CNT
E983-	BD 00 06	6220	LTL LDA VAR,X
E986-	99 00 06	6230	STA VAR,Y
E989-	E8	6240	INX
E98A-	C8	6250	INY
E98B-	C6 60	6260	DEC CNT
E98D-	D0 F4	6270	BNE LTL
E98F-	60	6280	RTS
		6290	*-----
		6300	* FETCH AC: AC=VARIABLE
		6310	*-----
		6320	*
		6330	* TRANSFERS VARIABLE TO AC
		6340	* ENTER WITH VAR PTR IN Y
		6350	* PROTECT PTR IN X
		6360	*
E990-	86 5F	6370	TMA STX TMP
E992-	A2 FD	6380	LDX #FD
E994-	B9 00 06	6390	MPL LDA VAR,Y
E997-	95 65	6400	STA SCL,X
E999-	C8	6410	INY
E99A-	E8	6420	INX
E99B-	D0 F7	6430	BNE MPL
E99D-	A6 5F	6440	LDX TMP
E99F-	60	6450	RTS
		6460	*-----
		6470	* FETCH RC: RC=VARIABLE
		6480	*-----
		6490	*
		6500	* TRANSFERS VARIABLE TO RC
		6510	* ENTER WITH VAR PTR IN X
		6520	*
E9A0-	8A	6530	TMR TXA
E9A1-	A8	6540	TAY
E9A2-	A2 FD	6550	LDX #FD
E9A4-	B9 00 06	6560	MRL LDA VAR,Y
E9A7-	95 5F	6570	STA TMF,X
E9A9-	C8	6580	INY
E9AA-	E8	6590	INX
E9AB-	D0 F7	6600	BNE MRL
E9AD-	60	6610	RTS
		6620	*-----
		6630	* STORE: VARIABLE=AC
		6640	*-----
		6650	*
		6660	* STORES A RESULT FROM AC INTO A VARIABLE LOCATION
		6670	*
		6680	* ENTER WITH VAR PTR IN Y
E9AE-	A2 FD	6690	STR LDX #FD
E9B0-	B5 65	6700	STL LDA SCL,X

E9B2-	99	00	06	6710	STA	VAR,Y
E9B5-	C8			6720	INY	
E9B6-	E8			6730	INX	
E9B7-	D0	F7		6740	BNE	STL
E9B9-	60			6750	RTS	
				6760	*-----	
				6770	* TRANSFER AC TO RC	
				6780	*-----	
E9BA-	A2	02		6790	TAR	LDX ##02
E9BC-	B5	62		6800	TRL	LDA ACL,X
E9BE-	95	5C		6810	STA	RCL,X
E9C0-	CA			6820	DEX	
E9C1-	10	F9		6830	BPL	TRL
E9C3-	60			6840	RTS	
				6850	*-----	
				6860	* TRANSFER AC TO TC	
				6870	*-----	
E9C4-	A2	02		6880	TAT	LDX ##02
E9C6-	B5	62		6890	TAL	LDA ACL,X
E9C8-	95	57		6900	STA	TCL,X
E9CA-	CA			6910	DEX	
E9CB-	10	F9		6920	BPL	TAL
E9CD-	60			6930	RTS	
				6940	*-----	
				6950	* TRANSFER AC TO QC	
				6960	*-----	
E9CE-	A2	02		6970	TAQ	LDX ##02
E9D0-	B5	62		6980	AQL	LDA ACL,X
E9D2-	95	54		6990	STA	QCL,X
E9D4-	CA			7000	DEX	
E9D5-	10	F9		7010	BPL	AQL
E9D7-	60			7020	RTS	
				7030	*-----	
				7040	* TRANSFER TC TO AC	
				7050	*-----	
E9D8-	A2	02		7060	TTA	LDX ##02
E9DA-	B5	57		7070	TTL	LDA TCL,X
E9DC-	95	62		7080	STA	ACL,X
E9DE-	CA			7090	DEX	
E9DF-	10	F9		7100	BPL	TTL
E9E1-	60			7110	RTS	
				7120	*-----	
				7130	* TRANSFER TC TO RC	
				7140	*-----	
E9E2-	A2	02		7150	TTR	LDX ##02
E9E4-	B5	57		7160	TXL	LDA TCL,X
E9E6-	95	5C		7170	STA	RCL,X
E9E8-	CA			7180	DEX	
E9E9-	10	F9		7190	BPL	TXL
E9EB-	60			7200	RTS	
				7210	*-----	
				7220	* TRANSFER QC TO AC	
				7230	*-----	
E9EC-	A2	02		7240	TQA	LDX ##02
E9EE-	B5	54		7250	QAL	LDA QCL,X
E9F0-	95	62		7260	STA	ACL,X
E9F2-	CA			7270	DEX	
E9F3-	10	F9		7280	BPL	QAL
E9F5-	60			7290	RTS	
				7300	*-----	

	7310	* TRANSFER QC TO RC	
	7320	*-----	
E9F6- A2 02	7330	TQR	LDX ##02
E9F8- B5 54	7340	QRL	LDA QCL,X
E9FA- 95 5C	7350		STA RCL,X
E9FC- CA	7360		DEX
E9FD- 10 F9	7370		BPL QRL
E9FF- 60	7380		RTS
	7390	*-----	
	7400	* TRANSFER RC TO AC	
	7410	*-----	
EA00- A2 02	7420	TRA	LDX ##02
EA02- B5 5C	7430	TRA.1	LDA RCL,X
EA04- 95 62	7440		STA ACL,X
EA06- CA	7450		DEX
EA07- 10 F9	7460		BPL TRA.1
EA09- 60	7470		RTS
	7480	*-----	

```

1180          .IN ROMON.V1.1.PART4
1000 *-----
1010 * 65C02 DISASSEMBLER
1020 * WRITTEN 09/16/84
1030 * BY MICHAEL W. FOWLER
1040 * DISASSEMBLES CODE FOR THESE
1050 * MICROPROCESSORS;
1060 *      6502
1070 *      65SC02      GTE,NCR
1080 *      R65C02      ROCKWELL
1090 *-----
1100 * DISASSEMBLE 'X' LINES
1110 *-----
EA0A- A2 06      1120 DASMSX LDX #$06
EA0C- 20 12 EA   1130          JSR DASMBL      DISASSEMBLE 6 LINES
EA0F- 4C 38 FA   1140          JMP GETCD1
EA12- DA         1150 DASMBL PHX
EA13- 20 25 EA   1160          JSR DASMON
EA16- A0 32      1170          LDY #$32
EA18- 20 48 FC   1180          JSR PAUSE
EA1B- 20 37 FC   1190          JSR WAITKY
EA1E- C9 0D      1200          CMP #$0D
EA20- FA         1210          PLX
EA21- CA         1220          DEX
EA22- D0 EE      1230          BNE DASMBL
EA24- 60         1240 DASMB1 RTS
1250 *-----
1260 * DISASSEMBLE ONE LINE
1270 *-----
EA25- 20 FC F3   1280 DASMON JSR DSPADD      DISPLAY ADDRESS
EA28- B2 3F      1290          LDA (ADDR01)
EA2A- A8         1300          TAY
EA2B- B9 44 EB   1310          LDA NUMBYT,Y GET NUMBER OF BYTE FOR OPCODE
EA2E- AA         1320          TAX
EA2F- 29 07      1330          AND #$07      MASK OUT ADDR MODE
EA31- 8D 5A 05   1340          STA BYTCNT      SAVE BYTE COUNT
EA34- 8A         1350          TXA
EA35- 4A         1360          LSR          GET ADDRESS MODE
EA36- 4A         1370          LSR
EA37- 4A         1380          LSR
EA38- 4A         1390          LSR
EA39- 8D 59 05   1400          STA ADMODE      AND SAVE IT
EA3C- 64 33      1410          STZ TEMP02
EA3E- A4 33      1420          LDY TEMP02
EA40- 5A         1430 SHOWBT PHX
EA41- A0 01      1440          LDY #$01
EA43- 20 6E FC   1450          JSR SPCOUT
EA46- 7A         1460          PLY
EA47- B1 3F      1470          LDA (ADDR01),Y
EA49- 20 B9 F3   1480          JSR HEXOUT
EA4C- E6 33      1490          INC TEMP02
EA4E- A4 33      1500          LDY TEMP02
EA50- CC 5A 05   1510          CPY BYTCNT
EA53- D0 EB      1520          BNE SHOWBT
EA55- 20 77 FC   1530          JSR TABOUT
EA58- B2 3F      1540          LDA (ADDR01) GET OPCODE
EA5A- A8         1550          TAY
EA5B- B9 44 EC   1560          LDA OPTABL,Y
EA5E- A8         1570          TAY

```

EA5F-	A2 03	1580	LDX ##03
EA61-	B9 44 ED	1590	DISPOP LDA OP CODE,Y
EA64-	20 A0 FA	1600	JSR OUTCHR
EA67-	C8	1610	INY
EA68-	CA	1620	DEX
EA69-	D0 F6	1630	BNE DISPOP
EA6B-	B2 3F	1640	LDA (ADDR01)
EA6D-	AA	1650	TAX
EA6E-	29 07	1660	AND ##07
EA70-	C9 07	1670	CMP ##07
EA72-	D0 0E	1680	BNE DSPMOD
EA74-	8A	1690	TXA
EA75-	4A	1700	LSR
EA76-	4A	1710	LSR
EA77-	4A	1720	LSR
EA78-	4A	1730	LSR
EA79-	29 07	1740	AND ##07
EA7B-	09 30	1750	ORA ##30
EA7D-	20 A0 FA	1760	JSR OUTCHR
EA80-	80 05	1770	BRA DSPMD1
EA82-	A0 01	1780	DSPMOD LDY ##01
EA84-	20 6E FC	1790	JSR SPCOUT
EA87-	AC 59 05	1800	DSPMD1 LDY ADMODE
EA8A-	D0 03	1810	BNE DSPMD2
EA8C-	4C 34 EB	1820	JMP ENDLIN
EA8F-	A2 00	1830	DSPMD2 LDX ##00
EA91-	88	1840	MODEL P DEY
EA92-	F0 0B	1850	BEQ PRTPRF
EA94-	BD 13 EE	1860	MODLP2 LDA PREFIX,X
EA97-	F0 03	1870	BEQ MODLP3
EA99-	E8	1880	INX
EA9A-	80 F8	1890	BRA MODLP2
EA9C-	EB	1900	MODLP3 INX
EA9D-	80 F2	1910	BRA MODEL P
EA9F-	BD 13 EE	1920	PRTPRF LDA PREFIX,X
EAA2-	F0 06	1930	BEQ PRTADD
EAA4-	20 A0 FA	1940	JSR OUTCHR
EAA7-	E8	1950	INX
EAA8-	80 F5	1960	BRA PRTPRF
EAAA-	AC 59 05	1970	PRTADD LDY ADMODE
EAAD-	C0 01	1980	CPY ##01
EAAF-	F0 41	1990	BEQ RELADD
EAB1-	C0 0D	2000	CPY ##0D
EAB3-	D0 0C	2010	BNE PRTAD2
EAB5-	A0 01	2020	LDY ##01
EAB7-	B1 3F	2030	LDA (ADDR01),Y
EAB9-	20 B9 F3	2040	JSR HEXOUT
EABC-	20 D3 EA	2050	JSR PRTPFX
EABF-	80 31	2060	BRA RELADD
EAC1-	AC 5A 05	2070	PRTAD2 LDY BYCNT
EAC4-	88	2080	PRTAD1 DEY
EAC5-	F0 07	2090	BEQ PSTFIX
EAC7-	B1 3F	2100	LDA (ADDR01),Y
EAC9-	20 B9 F3	2110	JSR HEXOUT
EACC-	80 F6	2120	BRA PRTAD1
EACE-	20 D3 EA	2130	PSTFIX JSR PRTPFX
EAD1-	80 61	2140	BRA ENDLIN
EAD3-	A2 00	2150	PRTPFX LDX ##00
EAD5-	AC 59 05	2160	LDY ADMODE
EAD8-	88	2170	PSTFX1 DEY

EAD9-	F0 0B		2180	BEQ PSTFX4
EADB-	BD 3F EE		2190	PSTFX2 LDA POSTFX,X
EADE-	F0 03		2200	BEQ PSTFX3
EAE0-	E8		2210	INX
EAE1-	80 F8		2220	BRA PSTFX2
EAE3-	E8		2230	PSTFX3 INX
EAE4-	80 F2		2240	BRA PSTFX1
EAE6-	BD 3F EE		2250	PSTFX4 LDA POSTFX,X
EAE9-	D0 01		2260	BNE PSTFX5
EAEB-	60		2270	RTS
EAEC-	20 A0 FA		2280	PSTFX5 JSR OUTCHR
EAEF-	E8		2290	INX
EAF0-	80 F4		2300	BRA PSTFX4
EAF2-	A4 40		2310	RELADD LDY ADDR01+1
EAF4-	A5 3F		2320	LDA ADDR01
EAF6-	1A		2330	INC
EAF7-	D0 01		2340	BNE RELAD1
EAF9-	C8		2350	INY
EAFA-	1A		2360	RELAD1 INC
EAFB-	D0 01		2370	BNE RELAD2
EAFD-	C8		2380	INY
EAFE-	AE 59 05		2390	RELAD2 LDX ADMODE
EB01-	E0 0D		2400	CPX #\$0D
EB03-	D0 04		2410	BNE RELAD3
EB05-	1A		2420	INC
EB06-	D0 01		2430	BNE RELAD3
EB08-	C8		2440	INY
EB09-	5A		2450	RELAD3 PHY
EB0A-	48		2460	PHA
EB0B-	E0 0D		2470	CPX #\$0D
EB0D-	D0 04		2480	BNE RELAD6
EB0F-	A0 02		2490	LDY #\$02
EB11-	80 02		2500	BRA RELAD7
EB13-	A0 01		2510	RELAD6 LDY #\$01
EB15-	B1 3F		2520	RELAD7 LDA (ADDR01),Y
EB17-	85 32		2530	STA TEMP01
EB19-	30 0F		2540	BMI RELAD5
EB1B-	68		2550	PLA
EB1C-	7A		2560	PLY
EB1D-	18		2570	CLC
EB1E-	65 32		2580	ADC TEMP01
EB20-	90 01		2590	BCC RELAD4
EB22-	C8		2600	INY
EB23-	AA		2610	RELAD4 TAX
EB24-	98		2620	TYA
EB25-	20 F5 F3		2630	JSR PRNTAX
EB28-	80 0A		2640	BRA ENDLIN
EB2A-	68		2650	RELAD5 PLA
EB2B-	7A		2660	PLY
EB2C-	18		2670	CLC
EB2D-	65 32		2680	ADC TEMP01
EB2F-	B0 F2		2690	BCS RELAD4
EB31-	88		2700	DEY
EB32-	80 EF		2710	BRA RELAD4
EB34-	20 67 FC		2720	ENDLIN JSR CROUT
EB37-	AE 5A 05		2730	LDX BYTCNT
EB3A-	E6 3F		2740	ADRINC INC ADDR01
EB3C-	D0 02		2750	BNE INCADR
EB3E-	E6 40		2760	INC ADDR01+1
EB40-	CA		2770	INCADR DEX

EB41-	D0	F7	2780	BNE	ADRINC	
EB43-	60		2790	RTS		
			2800	*-----		
			2810	* NUMBER OF INSTRUCTION BYTES		
			2820	* AND ADDRESSING MODE		
			2830	* ADDR MODE = UPPER NIBBLE		
			2840	* # BYTES = LOWER NIBBLE		
			2850	* ACCUM/IMPLIED=0		
			2860	RELATIVE	=1	
			2870	IMMEDIATE	=2	
			2880	ZERO PAGE	=3	
			2890	ZP,X	=4	
			2900	ZP,Y	=5	
			2910	ABSOLUTE	=6	
			2920	ABS,X	=7	
			2930	ABS,Y	=8	
			2940	INDIRECT	=9	
			2950	IND,X	=A	
			2960	IND,Y	=B	
			2970	IND,X ABS	=C	
			2980	BIT,REL	=D	
			2990	*-----		
EB44-	01	A2 01				
EB47-	01	32 32				
EB4A-	32	32 01				
EB4D-	22	01 01				
EB50-	63	63 63				
EB53-	D3		3000	NUMBYT	.HS 01A201013232323201220101636363D3	
EB54-	12	B2 92				
EB57-	01	32 42				
EB5A-	42	32 01				
EB5D-	83	01 01				
EB60-	63	73 73				
EB63-	D3		3010		.HS 12B292013242423201830101637373D3	
EB64-	63	A2 01				
EB67-	01	32 32				
EB6A-	32	32 01				
EB6D-	22	01 01				
EB70-	63	63 63				
EB73-	D3		3020		.HS 63A201013232323201220101636363D3	
EB74-	12	B2 92				
EB77-	01	42 42				
EB7A-	42	32 01				
EB7D-	83	01 01				
EB80-	73	73 73				
EB83-	D3		3030		.HS 12B292014242423201830101737373D3	
EB84-	01	A2 01				
EB87-	01	01 32				
EB8A-	32	32 01				
EB8D-	22	01 01				
EB90-	63	63 63				
EB93-	D3		3040		.HS 01A201010132323201220101636363D3	
EB94-	12	B2 92				
EB97-	01	01 42				
EB9A-	42	32 01				
EB9D-	83	01 01				
EBA0-	01	73 73				
EBA3-	D3		3050		.HS 12B292010142423201830101017373D3	
EBA4-	01	A2 01				
EBA7-	01	32 32				

EBAA-	32	32	01		
EBAD-	22	01	01		
EBB0-	93	63	63		
EBB3-	D3			3060	.HS 01A201013232323201220101936363D3
EBB4-	12	B2	92		
EBB7-	01	42	42		
EBBA-	42	32	01		
EBBD-	83	01	01		
EBC0-	C3	73	73		
EBC3-	D3			3070	.HS 12B292014242423201830101C37373D3
EBC4-	12	A2	01		
EBC7-	01	32	32		
EBCA-	32	32	01		
EBCD-	22	01	01		
EBD0-	63	63	63		
EBD3-	D3			3080	.HS 12A201013232323201220101636363D3
EBD4-	12	B2	92		
EBD7-	01	42	42		
EBDA-	52	32	01		
EBDD-	83	01	01		
EBE0-	63	73	73		
EBE3-	D3			3090	.HS 12B292014242523201830101637373D3
EBE4-	22	A2	22		
EBE7-	01	32	32		
EBEA-	32	32	01		
EBED-	22	01	01		
EBF0-	63	63	63		
EBF3-	D3			3100	.HS 22A222013232323201220101636363D3
EBF4-	12	B2	92		
EBF7-	01	42	42		
EBFA-	52	32	01		
EBFD-	83	01	01		
EC00-	73	73	83		
EC03-	D3			3110	.HS 12B292014242523201830101737383D3
EC04-	22	A2	01		
EC07-	01	32	32		
EC0A-	32	32	01		
EC0D-	22	01	01		
EC10-	63	63	63		
EC13-	D3			3120	.HS 22A201013232323201220101636363D3
EC14-	12	B2	92		
EC17-	01	01	42		
EC1A-	42	32	01		
EC1D-	83	01	01		
EC20-	01	73	73		
EC23-	D3			3130	.HS 12B292010142423201830101017373D3
EC24-	22	A2	01		
EC27-	01	32	32		
EC2A-	32	32	01		
EC2D-	22	01	01		
EC30-	63	63	63		
EC33-	D3			3140	.HS 22A201013232323201220101636363D3
EC34-	12	B2	92		
EC37-	01	01	42		
EC3A-	42	32	01		
EC3D-	83	01	01		
EC40-	01	73	73		
EC43-	D3			3150	.HS 12B292010142423201830101017373D3

3160 *-----

3170 * OP CODE POINTER TABLE

```

3180 *-----
EC44- 27 6F CC
EC47- CC BD 6F
EC4A- 06 8A 75
EC4D- 6F 06 CC
EC50- BD 6F 06
EC53- 09      3190 OPTABL .HS 276FCCCCBD6F068A756F06CCBD6F0609
EC54- 21 6F 6F
EC57- CC BA 6F
EC5A- 06 8A 30
EC5D- 6F 51 CC
EC60- BA 6F 06
EC63- 09      3200      .HS 216F6FCCBA6F068A306F51CCBA6F0609
EC64- 5D 03 CC
EC67- CC 18 03
EC6A- 8D 8A 81
EC6D- 03 8D CC
EC70- 18 03 8D
EC73- 09      3210      .HS 5D03CCCC18038D8A81038DCC18038D09
EC74- 1B 03 03
EC77- CC 18 03
EC7A- 8D 8A 9C
EC7D- 03 45 CC
EC80- 18 03 8D
EC83- 09      3220      .HS 1B0303CC18038D8A9C0345CC18038D09
EC84- 93 4E CC
EC87- CC CC 4E
EC8A- 69 8A 72
EC8D- 4E 69 CC
EC90- 5A 4E 69
EC93- 09      3230      .HS 934ECCCCCC4E698A724E69CC5A4E6909
EC94- 2A 4E 4E
EC97- CC CC 4E
EC9A- 69 8A 36
EC9D- 4E 7B CC
ECA0- CC 4E 69
ECA3- 09      3240      .HS 2A4E4ECCCC4E698A364E7BCCCC4E6909
ECA4- 96 00 CC
ECA7- CC B1 00
ECAA- 90 8A 7E
ECAD- 00 90 CC
ECB0- 5A 00 90
ECB3- 09      3250      .HS 9600CCCCB100908A7E0090CC5A009009
ECB4- 2D 00 00
ECB7- CC B1 00
ECBA- 90 8A A2
ECBD- 00 87 CC
ECC0- 5A 00 90
ECC3- 09      3260      .HS 2D0000CCB100908AA20087CC5A009009
ECC4- 24 A8 CC
ECC7- CC AE A8
ECCA- AB A5 4B
ECCD- 18 C3 CC
ECD0- AE A8 AB
ECD3- 0C      3270      .HS 24A8CCCCAEABABA54B18C3CCAEABAB0C
ECD4- 0F A8 A8
ECD7- CC AE A8
ECDA- AB A5 C9
ECDD- A8 C6 CC
ECE0- B1 A8 B1

```

ECE3-	0C		3280	.HS	0FABA8CCAEBABAB5C9A8C6CCB1A8B10C
ECE4-	66	60	63		
ECE7-	CC	66	60		
ECEA-	63	A5	B7		
ECED-	60	B4	CC		
ECF0-	66	60	63		
ECF3-	0C		3290	.HS	666063CC666063A5B760B4CC6660630C
ECF4-	12	60	60		
ECF7-	CC	66	60		
ECFA-	63	A5	39		
ECFD-	60	C0	CC		
ED00-	66	60	63		
ED03-	0C		3300	.HS	126060CC666063A53960C0CC6660630C
ED04-	42	3C	CC		
ED07-	CC	42	3C		
ED0A-	45	A5	57		
ED0D-	3C	48	CC		
ED10-	42	3C	45		
ED13-	0C		3310	.HS	423CCCCC423C45A5573C48CC423C450C
ED14-	1E	3C	3C		
ED17-	CC	CC	3C		
ED1A-	45	A5	33		
ED1D-	3C	78	CC		
ED20-	CC	3C	45		
ED23-	0C		3320	.HS	1E3C3CCCCC3C45A5333C78CCCC3C450C
ED24-	3F	99	CC		
ED27-	CC	3F	99		
ED2A-	51	A5	54		
ED2D-	99	6C	CC		
ED30-	3F	99	51		
ED33-	0C		3330	.HS	3F99CCCC3F9951A554996CCC3F99510C
ED34-	15	99	99		
ED37-	CC	CC	99		
ED3A-	51	A5	9F		
ED3D-	99	84	CC		
ED40-	CC	99	51		
ED43-	0C		3340	.HS	159999CCCC9951A59F9984CCCC99510C

3350 *-----

3360 * OP CODE TABLE

3370 *-----

ED44-	41	44	43	3380	OPCODE .AS /ADC/
ED47-	41	4E	44	3390	.AS /AND/
ED4A-	41	53	4C	3400	.AS /ASL/
ED4D-	42	42	52	3410	.AS /BBR/
ED50-	42	42	53	3420	.AS /BBS/
ED53-	42	43	43	3430	.AS /BCC/
ED56-	42	43	53	3440	.AS /BCS/
ED59-	42	45	51	3450	.AS /BEQ/
ED5C-	42	49	54	3460	.AS /BIT/
ED5F-	42	4D	49	3470	.AS /BMI/
ED62-	42	4E	45	3480	.AS /BNE/
ED65-	42	50	4C	3490	.AS /BPL/
ED68-	42	52	41	3500	.AS /BRA/
ED6B-	42	52	4B	3510	.AS /BRK/
ED6E-	42	56	43	3520	.AS /BVC/
ED71-	42	56	53	3530	.AS /BVS/
ED74-	43	4C	43	3540	.AS /CLC/
ED77-	43	4C	44	3550	.AS /CLD/
ED7A-	43	4C	49	3560	.AS /CLI/
ED7D-	43	4C	56	3570	.AS /CLV/

ED80-	43	4D	50	3580	.AS	/CMP/
ED83-	43	50	58	3590	.AS	/CPX/
ED86-	43	50	59	3600	.AS	/CPY/
ED89-	44	45	43	3610	.AS	/DEC/
ED8C-	44	45	58	3620	.AS	/DEX/
ED8F-	44	45	59	3630	.AS	/DEY/
ED92-	45	4F	52	3640	.AS	/EOR/
ED95-	49	4E	43	3650	.AS	/INC/
ED98-	49	4E	58	3660	.AS	/INX/
ED9B-	49	4E	59	3670	.AS	/INY/
ED9E-	4A	4D	50	3680	.AS	/JMP/
EDA1-	4A	53	52	3690	.AS	/JSR/
EDA4-	4C	44	41	3700	.AS	/LDA/
EDA7-	4C	44	58	3710	.AS	/LDX/
EDAA-	4C	44	59	3720	.AS	/LDY/
EDAD-	4C	53	52	3730	.AS	/LSR/
EDB0-	4E	4F	50	3740	.AS	/NOP/
EDB3-	4F	52	41	3750	.AS	/ORA/
EDB6-	50	48	41	3760	.AS	/PHA/
EDB9-	50	48	50	3770	.AS	/PHP/
EDBC-	50	48	58	3780	.AS	/PHX/
EDBF-	50	48	59	3790	.AS	/PHY/
EDC2-	50	4C	41	3800	.AS	/PLA/
EDC5-	50	4C	50	3810	.AS	/PLP/
EDC8-	50	4C	58	3820	.AS	/PLX/
EDCB-	50	4C	59	3830	.AS	/PLY/
EDCE-	52	4D	42	3840	.AS	/RMB/
EDD1-	52	4F	4C	3850	.AS	/ROL/
EDD4-	52	4F	52	3860	.AS	/ROR/
EDD7-	52	54	49	3870	.AS	/RTI/
EDDA-	52	54	53	3880	.AS	/RTS/
EDDD-	53	42	43	3890	.AS	/SBC/
EDE0-	53	45	43	3900	.AS	/SEC/
EDE3-	53	45	44	3910	.AS	/SED/
EDE6-	53	45	49	3920	.AS	/SEI/
EDE9-	53	4D	42	3930	.AS	/SMB/
EDEC-	53	54	41	3940	.AS	/STA/
EDEF-	53	54	58	3950	.AS	/STX/
EDF2-	53	54	59	3960	.AS	/STY/
EDF5-	53	54	5A	3970	.AS	/STZ/
EDF8-	54	41	58	3980	.AS	/TAX/
EDFB-	54	41	59	3990	.AS	/TAY/
EDFE-	54	52	42	4000	.AS	/TRB/
EE01-	54	53	42	4010	.AS	/TSB/
EE04-	54	53	58	4020	.AS	/TSX/
EE07-	54	58	41	4030	.AS	/TXA/
EE0A-	54	58	53	4040	.AS	/TXS/
EE0D-	54	59	41	4050	.AS	/TYA/
EE10-	3F	3F	3F	4060	.AS	/???/
				4070	*-----	
				4080	* ADDRESS MODE PREFIXES	
				4090	*-----	
EE13-	20	24		4100	PREFIX	.AS / \$/
EE15-	00			4110		.HS 00
EE16-	20	23	24	4120		.AS / ##/
EE19-	00			4130		.HS 00
EE1A-	20	24		4140		.AS / \$/
EE1C-	00			4150		.HS 00
EE1D-	20	24		4160		.AS / \$/
EE1F-	00			4170		.HS 00

EE20-	20 24	4180	.AS / \$/
EE22-	00	4190	.HS 00
EE23-	20 24	4200	.AS / \$/
EE25-	00	4210	.HS 00
EE26-	20 24	4220	.AS / \$/
EE28-	00	4230	.HS 00
EE29-	20 24	4240	.AS / \$/
EE2B-	00	4250	.HS 00
EE2C-	20 28 24	4260	.AS / (\$/
EE2F-	00	4270	.HS 00
EE30-	20 28 24	4280	.AS / (\$/
EE33-	00	4290	.HS 00
EE34-	20 28 24	4300	.AS / (\$/
EE37-	00	4310	.HS 00
EE38-	20 28 24	4320	.AS / (\$/
EE3B-	00	4330	.HS 00
EE3C-	20 24	4340	.AS / \$/
EE3E-	00	4350	.HS 00
		4360	*-----
		4370	* ADDRESS MODE POSTFIXES
		4380	*-----
EE3F-	20	4390	POSTFX .AS / /
EE40-	00	4400	.HS 00
EE41-	20	4410	.AS / /
EE42-	00	4420	.HS 00
EE43-	20	4430	.AS / /
EE44-	00	4440	.HS 00
EE45-	2C 58	4450	.AS / ,X/
EE47-	00	4460	.HS 00
EE48-	2C 59	4470	.AS / ,Y/
EE4A-	00	4480	.HS 00
EE4B-	20	4490	.AS / /
EE4C-	00	4500	.HS 00
EE4D-	2C 58	4510	.AS / ,X/
EE4F-	00	4520	.HS 00
EE50-	2C 59	4530	.AS / ,Y/
EE52-	00	4540	.HS 00
EE53-	29	4550	.AS /) /
EE54-	00	4560	.HS 00
EE55-	2C 58 29	4570	.AS / ,X) /
EE58-	00	4580	.HS 00
EE59-	29 2C 59	4590	.AS /) ,Y/
EE5C-	00	4600	.HS 00
EE5D-	29 2C 58	4610	.AS /) ,X/
EE60-	00	4620	.HS 00
EE61-	2C 24	4630	.AS / ,\$/
EE63-	00	4640	.HS 00

```

1200          .IN ROMON.V1.1.PART5
1000 *-----
1010 * MONITOR ROUTINES BEGINNING
1020 * AT $F000
1030 *-----
1040 * DO COMMAND IN LINE BUFFER
1050 *-----
EE64- 64 2E    1060 DOCMD  STZ BUFCNT    START AT BEGINNING OF LINE BUFFER
EE66- A4 2E    1070 DOCMD1 LDY BUFCNT    GET POINTER TO LINE BUFFER CHAR
EE68- B9 00 02 1080          LDA LINBUF,Y NOW GET A CHARACTER
EE68- A2 00    1090          LDX #ZERO    GET COMMAND TABLE POINTER
EE6D- DD 8D EE 1100 DOCMD2 CMP COMTAB,X IS THIS THE COMMAND?
EE70- F0 09    1110          BEQ DOCMD3    YES, NOW GO JUMP TO COMMAND ROUTINE
EE72- E8      1120          INX          NO, LOOK AGAIN
EE73- E8      1130          INX
EE74- E8      1140          INX
EE75- E0 36    1150          CPX #36      END OF TABLE?
EE77- 10 11    1160          BPL DOCMD4    YES, SEND ERROR MESSAGE
EE79- 80 F2    1170          BRA DOCMD2    NO LOOK SOME MORE
EE7B- E8      1180 DOCMD3 INX          GET THE ADDRESS OF THE ROUTINE
EE7C- BD 8D EE 1190          LDA COMTAB,X
EE7F- 85 45    1200          STA ADDR7B
EE81- E8      1210          INX
EE82- BD 8D EE 1220          LDA COMTAB,X
EE85- 85 46    1230          STA ADDR7B+1
EE87- 6C 45 00 1240          JMP (ADDR7B) AND JUMP TO IT
EE8A- 4C 10 F4 1250 DOCMD4 JMP ERRORX
1260 *-----
1270 * COMMAND TABLE
1280 *-----
EE8D- 24      1290 COMTAB .DA #'$
EE8E- F4 EE    1300          .DA GETAD1    GET ADDRESS #1
EE90- 2E      1310          .DA #'.
EE91- FA EE    1320          .DA GETAD2    GET ADDRESS #2
EE93- 3C      1330          .DA #'<
EE94- 00 EF    1340          .DA GETAD3    GET ADDRESS #3
EE96- 3A      1350          .DA #':
EE97- 88 EF    1360          .DA GETVAL    GET A HEX VALUE
EE99- 20      1370          .DA #'
EE9A- 88 EF    1380          .DA GETVAL    GET SOME MORE HEX VALUES
EE9C- 44      1390          .DA #'D
EE9D- 61 EF    1400          .DA LINDAT    DISPLAY A LINE OF DATA
EE9F- 4D      1410          .DA #'M
EEA0- F6 EF    1420          .DA MOVMEM    MOVE SOME MEMORY
EEA2- 56      1430          .DA #'V
EEA3- 6D F0    1440          .DA MEMCMP    COMPARE TWO BLOCKS OF MEMORY
EEA5- 47      1450          .DA #'G
EEA6- C6 EE    1460          .DA GOJUMP    JUMP TO AN ADDRESS
EEA8- 0D      1470          .DA #00D
EEA9- C4 F0    1480          .DA NXTLIN    NEXT LINE
EEAB- 4C      1490          .DA #'L
EEAC- 0A EA    1500          .DA DASMSX    DISASSEMBLE SOME MEMORY
EEAE- 53      1510          .DA #'S
EEAF- CD F2    1520          .DA SETIME    SET THE CLOCK
EEB1- 54      1530          .DA #'T
EEB2- FA F1    1540          .DA GETIME    DISPLAY THE TIME
EEB4- 58      1550          .DA #'X
EEB5- DB F0    1560          .DA SLAVE    TRANSFER DATA TO/FROM PARALLEL PORT
EEB7- 43      1570          .DA #'C

```

EEB8-	3F F4	1580	.DA SERCOM	TRANSFER DATA TO/FROM SERIAL PORT
EEBA-	4B	1590	.DA #'K	
EEBB-	38 F4	1600	.DA RMTCOM	TRANSFER DATA VIA RF MODEM
EEBD-	55	1610	.DA #'U	
EEBE-	C3 EE	1620	.DA GODOIT	USER MONITOR COMMAND
EEC0-	45	1630	.DA #'E	
EEC1-	D0 EE	1640	.DA ENTLIF	ENTER LIFE ROM IF AVAILABLE
		1650	*-----	
		1660	* USER MONITOR COMMAND	
		1670	*-----	
EEC3-	6C 22 00	1680	GODOIT JMP	(USRMON)
		1690	*-----	
		1700	* GO JUMP TO ADDRESS	
		1710	*-----	
EEC6-	A9 FA	1720	GOJUMP LDA /GETCMD	SET UP A RETURN ADDRESS
EEC8-	48	1730	PHA	
EEC9-	A9 35	1740	LDA #GETCMD	
EECB-	3A	1750	DEC	
EECC-	48	1760	PHA	
EECD-	6C 3F 00	1770	JMP (ADDR01)	AND JUMP TO THE ROUTINE
		1780	*-----	
		1790	* ENTER LIFE LOOP	
		1800	*-----	
EED0-	AD 61 E0	1810	ENTLIF LDA BITOUT+1	
EED3-	29 E3	1820	AND #\$E3	
EED5-	09 04	1830	ORA #\$04	
EED7-	8D 61 E0	1840	STA BITOUT+1	
EEDA-	AD 00 C0	1850	LDA \$C000	
EEDD-	C9 4C	1860	CMP #\$4C	
EEDF-	D0 08	1870	BNE ETLIF1	
EEE1-	A9 04	1880	LDA #\$04	
EEE3-	8D 53 05	1890	STA CURBNK	
EEE6-	6C 6F 00	1900	JMP (LIFEAD)	
EEE9-	AD 61 E0	1910	ETLIF1 LDA BITOUT+1	
EEEC-	29 E3	1920	AND #\$E3	
EEEE-	8D 61 E0	1930	STA BITOUT+1	
EEF1-	4C 35 FA	1940	JMP GETCMD	
		1950	*-----	
		1960	* GET ADDRESS	
		1970	*-----	
EEF4-	A2 01	1980	GETAD1 LDX #\$01	ADDR01
EEF6-	86 30	1990	STX CURADD	
EEF8-	80 0A	2000	BRA GETADD	
EEFA-	A2 03	2010	GETAD2 LDX #\$03	ADDR02
EEFC-	86 30	2020	STX CURADD	
EEFE-	80 04	2030	BRA GETADD	
EF00-	A2 05	2040	GETAD3 LDX #\$05	ADDR03
EF02-	86 30	2050	STX CURADD	
EF04-	E6 2E	2060	GETADD INC BUFCNT	
EF06-	A4 2E	2070	GADD01 LDY BUFCNT	
EF08-	B9 00 02	2080	LDA LINBUF,Y	
EF0B-	85 35	2090	STA TEMP04	
EF0D-	E6 2E	2100	INC BUFCNT	
EF0F-	A4 2E	2110	LDY BUFCNT	
EF11-	B9 00 02	2120	LDA LINBUF,Y	
EF14-	AA	2130	TAX	
EF15-	A5 35	2140	LDA TEMP04	
EF17-	20 E0 F3	2150	JSR ASCHEX	
EF1A-	A6 30	2160	LDX CURADD	
EF1C-	95 3F	2170	STA ADDR01,X	

EF1E-	8A		2180	TXA	
EF1F-	29 01		2190	AND #\$01	
EF21-	F0 07		2200	BEQ GADD02	
EF23-	E6 2E		2210	INC BUFCNT	
EF25-	C6 30		2220	DEC CURADD	
EF27-	4C 06 EF		2230	JMP GADD01	
EF2A-	A5 30		2240	GADD02 LDA CURADD	
EF2C-	C9 02		2250	CMP #\$02	
EF2E-	10 0F		2260	BPL GADD03	
EF30-	18		2270	CLC	
EF31-	69 03		2280	ADC #\$03	
EF33-	85 30		2290	STA CURADD	
EF35-	A5 2E		2300	LDA BUFCNT	
EF37-	38		2310	SEC	
EF38-	E9 03		2320	SBC #\$03	
EF3A-	85 2E		2330	STA BUFCNT	
EF3C-	4C 06 EF		2340	JMP GADD01	
EF3F-	E6 2E		2350	GADD03 INC BUFCNT	
EF41-	4C 66 EE		2360	JMP DOCMD1	
			2370	*-----	
			2380	* DISPLAY LINE OF HEX DATA	
			2390	*-----	
EF44-	20 67 FC		2400	NWLNDT JSR CROUT	NEW LINE
EF47-	A0 A0		2410	LDY #\$A0	
EF49-	20 48 FC		2420	JSR PAUSE	
EF4C-			2430	>CMPD ADDR01,ZERONM	
EF4C-	A5 40		0000>	LDA ADDR01+1	
EF4E-	CD D2 FF		0000>	CMP ZERONM+1	
EF51-	D0 05		0000>	BNE :1	
EF53-	A5 3F		0000>	LDA ADDR01	
EF55-	CD D1 FF		0000>	CMP ZERONM	
			0000>	:1	
EF58-	F0 29		2440	BEQ LNDT02	
EF5A-	20 37 FC		2450	JSR WAITKY	IF KEY PRESSED WAIT UNTIL KEY PRESSED A GAIN
EF5D-	C9 0D		2460	CMP #\$0D	
EF5F-	F0 22		2470	BEQ LNDT02	
EF61-	20 FC F3		2480	LINDAT JSR DSPADD	DISPLAY AN ADDRESS
EF64-	64 37		2490	STZ INDEX1	SET UP BYTE COUNTER
EF66-	20 B2 F3		2500	LNDT01 JSR OUTBYT	DISPLAY A BYTE
EF69-			2510	>INCD ADDR01 INCREMENT THE ADDRESS POINTER	
EF69-	E6 3F		0000>	INC ADDR01	
EF6B-	D0 02		0000>	BNE :1	
EF6D-	E6 40		0000>	INC ADDR01+1	
			0000>	:1	
EF6F-	E6 37		2520	INC INDEX1	AND THE BYTE COUNTER
EF71-	A9 07		2530	LDA #\$07	8 BYTES DISPLAYED ON THIS LINE?
EF73-	C5 37		2540	CMP INDEX1	
EF75-	10 EF		2550	BPL LNDT01	NO, DO SOME MORE
EF77-			2560	>CMPD ADDR01,ADDR02 ARE WE PAST LAST ADDRESS TO BE DISPLAYED?	
EF77-	A5 40		0000>	LDA ADDR01+1	
EF79-	C5 42		0000>	CMP ADDR02+1	
EF7B-	D0 04		0000>	BNE :1	
EF7D-	A5 3F		0000>	LDA ADDR01	
EF7F-	C5 41		0000>	CMP ADDR02	
			0000>	:1	
EF81-	90 C1		2570	BCC NWLNDT	NO, DO SOME MORE
EF83-	E6 2E		2580	LNDT02 INC BUFCNT	YES, GO GET NEXT COMMAND
EF85-	4C 66 EE		2590	JMP DOCMD1	


```

2600 *-----
2610 * GET HEX VALUE AND STORE IT
2620 *-----
EF88- 2630 GETVAL >CMPD ADDR01,ADDR02
EF88- A5 40 0000> LDA ADDR01+1
EF8A- C5 42 0000> CMP ADDR02+1
EF8C- D0 04 0000> BNE :1
EF8E- A5 3F 0000> LDA ADDR01
EF90- C5 41 0000> CMP ADDR02
0000> :1
EF92- 90 23 2640 BCC GETVL1
EF94- E6 2E 2650 INC BUFCNT
EF96- A4 2E 2660 LDY BUFCNT
EF98- B9 00 02 2670 LDA LINBUF,Y
EF9B- 85 36 2680 STA TEMP05
EF9D- E6 2E 2690 INC BUFCNT
EF9F- A4 2E 2700 LDY BUFCNT
EFA1- B9 00 02 2710 LDA LINBUF,Y
EFA4- AA 2720 TAX
EFA5- A5 36 2730 LDA TEMP05
EFA7- 20 E0 F3 2740 JSR ASCHEX
EFAA- 92 3F 2750 STA (ADDR01)
EFAC- 2760 >INCD ADDR01
EFAC- E6 3F 0000> INC ADDR01
EFAE- D0 02 0000> BNE :1
EFB0- E6 40 0000> INC ADDR01+1
0000> :1
EFB2- E6 2E 2770 INC BUFCNT
EFB4- 4C 66 EE 2780 JMP DOCMD1
EFB7- E6 2E 2790 GETVL1 INC BUFCNT
EFB9- A4 2E 2800 LDY BUFCNT
EFBB- B9 00 02 2810 LDA LINBUF,Y
EFBE- 85 36 2820 STA TEMP05
EFC0- E6 2E 2830 INC BUFCNT
EFC2- A4 2E 2840 LDY BUFCNT
EFC4- B9 00 02 2850 LDA LINBUF,Y
EFC7- AA 2860 TAX
EFC8- A5 36 2870 LDA TEMP05
EFCA- 20 E0 F3 2880 JSR ASCHEX
EFCD- 85 36 2890 STA TEMP05
EFCF- A5 36 2900 GETVL2 LDA TEMP05
EFD1- 92 3F 2910 STA (ADDR01)
EFD3- 2920 >INCD ADDR01
EFD3- E6 3F 0000> INC ADDR01
EFD5- D0 02 0000> BNE :1
EFD7- E6 40 0000> INC ADDR01+1
0000> :1
EFD9- 2930 >CMPD ADDR01,ADDR02
EFD9- A5 40 0000> LDA ADDR01+1
EFD8- C5 42 0000> CMP ADDR02+1
EFD0- D0 04 0000> BNE :1
EFD5- A5 3F 0000> LDA ADDR01
EFE1- C5 41 0000> CMP ADDR02
0000> :1
EFE3- F0 02 2940 BEQ GETVL3
EFE5- 80 E8 2950 BRA GETVL2
EFE7- A5 36 2960 GETVL3 LDA TEMP05
EFE9- 92 3F 2970 STA (ADDR01)
EFEB- 2980 >INCD ADDR01
EFEB- E6 3F 0000> INC ADDR01

```

```

EFED- D0 02      0000>          BNE :1
EFEF- E6 40      0000>          INC ADDR01+1
                                0000> :1
EFF1- E6 2E      2990          INC BUFCNT
EFF3- 4C 66 EE   3000          JMP DOCMD1
                                3010 *-----
                                3020 * MOVE BLOCK OF MEMORY
                                3030 * $DEST<STRT.ENDM
                                3040 *-----
EFF6- 20 FE EF   3050 MOVMEM JSR MEMMOV
EFF9- E6 2E      3060          INC BUFCNT
EFFB- 4C 66 EE   3070          JMP DOCMD1
EFFE-            3080 MEMMOV >CMPD ADDR01,ADDR03
EFFE- A5 40      0000>          LDA ADDR01+1
F000- C5 44      0000>          CMP ADDR03+1
F002- D0 04      0000>          BNE :1
F004- A5 3F      0000>          LDA ADDR01
F006- C5 43      0000>          CMP ADDR03
                                0000> :1
F008- F0 04      3090          BEQ MEMOV1
F00A- B0 21      3100          BCS MEMOV3
F00C- 80 01      3110          BRA MEMOV2
F00E- 60         3120 MEMOV1 RTS
F00F- B2 43      3130 MEMOV2 LDA (ADDR03)
F011- 92 3F      3140          STA (ADDR01)
F013-            3150          >CMPD ADDR03,ADDR02
F013- A5 44      0000>          LDA ADDR03+1
F015- C5 42      0000>          CMP ADDR02+1
F017- D0 04      0000>          BNE :1
F019- A5 43      0000>          LDA ADDR03
F01B- C5 41      0000>          CMP ADDR02
                                0000> :1
F01D- F0 EF      3160          BEQ MEMOV1
F01F-            3170          >INCD ADDR03
F01F- E6 43      0000>          INC ADDR03
F021- D0 02      0000>          BNE :1
F023- E6 44      0000>          INC ADDR03+1
                                0000> :1
F025-            3180          >INCD ADDR01
F025- E6 3F      0000>          INC ADDR01
F027- D0 02      0000>          BNE :1
F029- E6 40      0000>          INC ADDR01+1
                                0000> :1
F02B- 80 E2      3190          BRA MEMOV2
F02D- 38         3200 MEMOV3 SEC
F02E- A5 41      3210          LDA ADDR02
F030- E5 43      3220          SBC ADDR03
F032- 85 33      3230          STA TEMP02
F034- A5 42      3240          LDA ADDR02+1
F036- E5 44      3250          SBC ADDR03+1
F038- 85 34      3260          STA TEMP02+1
F03A- 18         3270          CLC
F03B- A5 33      3280          LDA TEMP02
F03D- 65 3F      3290          ADC ADDR01
F03F- 85 3F      3300          STA ADDR01
F041- A5 34      3310          LDA TEMP02+1
F043- 65 40      3320          ADC ADDR01+1
F045- 85 40      3330          STA ADDR01+1
F047- B2 41      3340 MEMOV4 LDA (ADDR02)
F049- 92 3F      3350          STA (ADDR01)

```

F04B-		3360	>DECD ADDR02
F04B-	A2 FF	0000>	LDX ##FF
F04D-	C6 41	0000>	DEC ADDR02
F04F-	E4 41	0000>	CPX ADDR02
F051-	D0 02	0000>	BNE :1
F053-	C6 42	0000>	DEC ADDR02+1
		0000>	:1
F055-		3370	>DECD ADDR01
F055-	A2 FF	0000>	LDX ##FF
F057-	C6 3F	0000>	DEC ADDR01
F059-	E4 3F	0000>	CPX ADDR01
F05B-	D0 02	0000>	BNE :1
F05D-	C6 40	0000>	DEC ADDR01+1
		0000>	:1
F05F-		3380	>CMPD ADDR02,ADDR03
F05F-	A5 42	0000>	LDA ADDR02+1
F061-	C5 44	0000>	CMP ADDR03+1
F063-	D0 04	0000>	BNE :1
F065-	A5 41	0000>	LDA ADDR02
F067-	C5 43	0000>	CMP ADDR03
		0000>	:1
F069-	B0 DC	3390	BCS MEMOV4
F06B-	B0 A1	3400	BRA MEMOV1
		3410	*-----
		3420	* COMPARE A BLOCK OF MEMORY
		3430	*-----
F06D-	B2 43	3440	MEMCMP LDA (ADDR03)
F06F-	D2 3F	3450	CMP (ADDR01)
F071-	F0 03	3460	BEQ MCMP01
F073-	20 95 F0	3470	JSR DSPMEM
F076-		3480	MCMP01 >CMPD ADDR02,ADDR03
F076-	A5 42	0000>	LDA ADDR02+1
F078-	C5 44	0000>	CMP ADDR03+1
F07A-	D0 04	0000>	BNE :1
F07C-	A5 41	0000>	LDA ADDR02
F07E-	C5 43	0000>	CMP ADDR03
		0000>	:1
F080-	D0 05	3490	BNE MCMP03
F082-	E6 2E	3500	INC BUFCNT
F084-	4C 66 EE	3510	JMP DDCMD1
F087-		3520	MCMP03 >INCD ADDR03
F087-	E6 43	0000>	INC ADDR03
F089-	D0 02	0000>	BNE :1
F08B-	E6 44	0000>	INC ADDR03+1
		0000>	:1
F08D-		3530	>INCD ADDR01
F08D-	E6 3F	0000>	INC ADDR01
F08F-	D0 02	0000>	BNE :1
F091-	E6 40	0000>	INC ADDR01+1
		0000>	:1
F093-	B0 D8	3540	BRA MEMCMP
F095-	20 FC F3	3550	DSPMEM JSR DSPADD
F098-	20 B2 F3	3560	JSR OUTBYT
F09B-	20 6C FC	3570	JSR SPOUT
F09E-	A9 28	3580	LDA ##28
F0A0-	20 A0 FA	3590	JSR OUTCHR
F0A3-	B2 43	3600	LDA (ADDR03)
F0A5-	20 B9 F3	3610	JSR HEXOUT
F0A8-	A9 29	3620	LDA ##29
F0AA-	20 A0 FA	3630	JSR OUTCHR

```

F0AD- 20 67 FC 3640 JSR CROUT
F0B0- A0 A0 3650 LDY #A0
F0B2- 20 48 FC 3660 JSR PAUSE
F0B5- 20 37 FC 3670 JSR WAITKY
F0B8- C9 0D 3680 CMP #0D
F0BA- D0 07 3690 BNE MCMP02
F0BC- 68 3700 PLA
F0BD- 68 3710 PLA
F0BE- E6 2E 3720 INC BUFCNT
F0C0- 4C 66 EE 3730 JMP DOCMD1
F0C3- 60 3740 MCMP02 RTS
3750 *-----
3760 * GET NEXT LINE
3770 *-----
F0C4- A5 2E 3780 NXTLIN LDA BUFCNT
F0C6- C9 00 3790 CMP #00
F0C8- F0 02 3800 BEQ NXTLN1
F0CA- 80 0C 3810 BRA NXTLN2
F0CC- 20 FC F3 3820 NXTLN1 JSR DSPADD
F0CF- 20 B2 F3 3830 JSR OUTBYT
F0D2- 3840 >INCD ADDR01
F0D2- E6 3F 0000> INC ADDR01
F0D4- D0 02 0000> BNE :1
F0D6- E6 40 0000> INC ADDR01+1
0000> :1
F0D8- 4C 35 FA 3850 NXTLN2 JMP GETCMD
3860 *-----
3870 * SLAVE TO REMOTE COMPUTER
3880 * FOR UP/DOWNLOADING VIA PARALLEL
3890 *-----
F0DB- 20 11 E6 3900 SLAVE JSR LOSPD
F0DE- 20 50 F1 3910 JSR INTAPL
F0E1- 64 37 3920 SLAV10 STZ INDEX1
F0E3- 20 07 F1 3930 JSR MSGOUT
F0E6- 20 8F F1 3940 JSR RECCMD
F0E9- AD 00 03 3950 LDA MSGBUF
F0EC- C9 D2 3960 CMP #'R+$80
F0EE- D0 03 3970 BNE SLAV20
F0F0- 4C BC F1 3980 JMP HANDLR
F0F3- C9 D3 3990 SLAV20 CMP #'S+$80
F0F5- D0 03 4000 BNE ERRXXX
F0F7- 4C DB F1 4010 JMP HANDLS
F0FA- A0 1A 4020 ERRXXX LDY #1A
F0FC- 84 37 4030 STY INDEX1
F0FE- 20 07 F1 4040 JSR MSGOUT
F101- 20 54 F1 4050 JSR PIAINP
F104- 4C 10 F4 4060 JMP ERRORX
F107- A4 37 4070 MSGOUT LDY INDEX1
F109- B9 16 F1 4080 LDA MSG100,Y
F10C- F0 07 4090 BEQ MSGEND
F10E- 20 6B F1 4100 JSR SNDCHR
F111- E6 37 4110 INC INDEX1
F113- 80 F2 4120 BRA MSGOUT
F115- 60 4130 MSGEND RTS
F116- A0 AA D2
F119- CF C2 CF
F11C- D4 A0 D2
F11F- C5 C1 C4
F122- D9 A0 C6
F125- CF D2 A0

```

F12B-	C3	CF	CD		
F12B-	CD	C1	CE		
F12E-	C4			4140	MSG100 .AS -/ *ROBOT READY FOR COMMAND/
F12F-	8D	00		4150	.HS 8D00
F131-	A0	D3	C5		
F134-	D4	A0	D4		
F137-	CF	A0	D2		
F13A-	C5	C3	C5		
F13D-	C9	D6	C5	4160	SETRCV .AS -/ SET TO RECEIVE/
F140-	8D	00		4170	.HS 8D00
F142-	A0	D3	C5		
F145-	D4	A0	D4		
F148-	CF	A0	D3		
F14B-	C5	CE	C4	4180	SETSND .AS -/ SET TO SEND/
F14E-	8D	00		4190	.HS 8D00
				4200	*-----
				4210	* PARALLEL COMMUNICATOR ROUTINES
				4220	*-----
F150-	A9	FF		4230	INTAPL LDA #\$FF
F152-	80	02		4240	BRA INTPIA
F154-	A9	00		4250	PIAINP LDA #\$00
F156-	9C	31	E0	4260	INTPIA STZ PARADD+1
F159-	9C	33	E0	4270	STZ PARADD+3
F15C-	9C	30	E0	4280	STZ PARADD
F15F-	8D	32	E0	4290	STA PARADD+2
F162-	A9	2C		4300	LDA #\$2C
F164-	8D	31	E0	4310	STA PARADD+1
F167-	8D	33	E0	4320	STA PARADD+3
F16A-	60			4330	RTS
				4340	*-----
				4350	* SEND CHARACTER TO PARAL
				4360	*-----
F16B-	8D	32	E0	4370	SNDCHR STA PARADD+2 SEND CHAR TO PORT A
F16E-	2C	33	E0	4380	SNDWAT BIT PARADD+3 SEE IF CHAR TAKEN
F171-	10	FB		4390	BPL SNDWAT
F173-	AD	32	E0	4400	LDA PARADD+2
F176-	60			4410	RTS
				4420	*-----
				4430	* RECEIVE CHARACTER FROM PARAL
				4440	*-----
F177-	20	BF	FA	4450	RECKEY JSR INCHAR
F17A-	F0	0A		4460	BEQ RECCHR
F17C-	C9	03		4470	CMP #\$03
F17E-	D0	06		4480	BNE RECCHR
F180-	20	54	F1	4490	JSR PIAINP
F183-	4C	13	F4	4500	JMP ERRORZ
F186-	2C	31	E0	4510	RECCHR BIT PARADD+1 DATA READY?
F189-	10	EC		4520	BPL RECKEY
F18B-	AD	30	E0	4530	LDA PARADD GET DATA
F18E-	60			4540	RTS
				4550	*-----
				4560	* RECEIVE COMMANDS FROM PARAL
				4570	*-----
F18F-	A0	00		4580	RECCMD LDY #\$00
F191-	20	86	F1	4590	NXTCMD JSR RECCHR
F194-	99	00	03	4600	STA MSGBUF,Y
F197-	C8			4610	INY
F198-	C0	05		4620	CPY #\$05
F19A-	D0	F5		4630	BNE NXTCMD
F19C-	60			4640	RTS

```

4650 *-----
4660 * GET TRANSFER ADDRESSES
4670 *-----
F19D- 84 37 4680 XFRADD STY INDEX1
F19F- 20 07 F1 4690 JSR MSGOUT
F1A2- AD 01 03 4700 LDA MSGBUF+1
F1A5- 85 40 4710 STA ADDR01+1
F1A7- AD 02 03 4720 LDA MSGBUF+2
F1AA- 85 3F 4730 STA ADDR01
F1AC- 18 4740 CLC
F1AD- AD 04 03 4750 LDA MSGBUF+4
F1B0- 65 3F 4760 ADC ADDR01
F1B2- 85 41 4770 STA ADDR02
F1B4- AD 03 03 4780 LDA MSGBUF+3
F1B7- 65 40 4790 ADC ADDR01+1
F1B9- 85 42 4800 STA ADDR02+1
F1BB- 60 4810 RTS
4820 *-----
4830 * RECEIVE BLOCK FROM PARAL
4840 *-----
F1BC- A0 1B 4850 HANDLR LDY #1B
F1BE- 20 9D F1 4860 JSR XFRADD
F1C1- 20 86 F1 4870 HNDLR1 JSR RECCHR
F1C4- 92 3F 4880 STA (ADDR01)
F1C6- 4890 >INCD ADDR01
F1C6- E6 3F 0000> INC ADDR01
F1C8- D0 02 0000> BNE :1
F1CA- E6 40 0000> INC ADDR01+1
0000> :1
F1CC- 4900 >CMPD ADDR01,ADDR02
F1CC- A5 40 0000> LDA ADDR01+1
F1CE- C5 42 0000> CMP ADDR02+1
F1D0- D0 04 0000> BNE :1
F1D2- A5 3F 0000> LDA ADDR01
F1D4- C5 41 0000> CMP ADDR02
0000> :1
F1D6- D0 E9 4910 BNE HNDLR1
F1D8- 4C E1 F0 4920 JMP SLAV10
4930 *-----
4940 * SEND BLOCK TO PARAL
4950 *-----
F1DB- A0 2C 4960 HANDLS LDY #2C
F1DD- 20 9D F1 4970 JSR XFRADD
F1E0- B2 3F 4980 HNDLS1 LDA (ADDR01)
F1E2- 20 6B F1 4990 JSR SNDCHR
F1E5- 5000 >INCD ADDR01
F1E5- E6 3F 0000> INC ADDR01
F1E7- D0 02 0000> BNE :1
F1E9- E6 40 0000> INC ADDR01+1
0000> :1
F1EB- 5010 >CMPD ADDR01,ADDR02
F1EB- A5 40 0000> LDA ADDR01+1
F1ED- C5 42 0000> CMP ADDR02+1
F1EF- D0 04 0000> BNE :1
F1F1- A5 3F 0000> LDA ADDR01
F1F3- C5 41 0000> CMP ADDR02
0000> :1
F1F5- D0 E9 5020 BNE HNDLS1
F1F7- 4C E1 F0 5030 JMP SLAV10
5040 *-----

```

```

5050 * GET TIME ROUTINE
5060 *-----
F1FA- 20 00 F2 5070 GETIME JSR GETIM0
F1FD- 4C 35 FA 5080      JMP GETCMD
F200- 20 54 F3 5090 GETIM0 JSR NWYEAR
F203- 20 6C FC 5100      JSR SPOUT
F206- A5 28      5110      LDA TIMBUF+3
F208- 29 3F      5120      AND #$3F
F20A- 20 B9 F3 5130      JSR HEXOUT
F20D- A5 29      5140      LDA TIMBUF+4
F20F- 29 1F      5150      AND #$1F
F211- 20 84 F3 5160      JSR BCDBIN
F214- AA          5170      TAX
F215- CA          5180      DEX
F216- A0 00      5190      LDY #$00
F218- B9 54 F2 5200 GETIM1 LDA MONDAT,Y
F21B- F0 0A      5210      BEQ GETIM3
F21D- C8          5220      INY
F21E- E0 00      5230      CPX #$00
F220- D0 F6      5240      BNE GETIM1
F222- 20 A0 FA 5250 GETIM2 JSR OUTCHR
F225- 80 F1      5260      BRA GETIM1
F227- C8          5270 GETIM3 INY
F228- CA          5280      DEX
F229- 30 02      5290      BMI GETIM4
F22B- 80 EB      5300      BRA GETIM1
F22D- A5 2B      5310 GETIM4 LDA YEARBF+1
F22F- 20 B9 F3 5320      JSR HEXOUT
F232- A5 2A      5330      LDA YEARBF
F234- 20 B9 F3 5340      JSR HEXOUT
F237- 20 6C FC 5350      JSR SPOUT
F23A- A2 02      5360      LDX #$02
F23C- B5 25      5370 GETIM5 LDA TIMBUF,X
F23E- E0 02      5380      CPX #$02
F240- D0 02      5390      BNE GETIM6
F242- 29 3F      5400      AND #$3F
F244- CA          5410 GETIM6 DEX
F245- 30 0A      5420      BMI GETIM7
F247- 20 B9 F3 5430      JSR HEXOUT
F24A- A9 3A      5440      LDA #'
F24C- 20 A0 FA 5450      JSR OUTCHR
F24F- 80 EB      5460      BRA GETIM5
F251- 4C B9 F3 5470 GETIM7 JMP HEXOUT
F254- 20 4A 41
F257- 4E 55 41
F25A- 52 59 2C
F25D- 20          5480 MONDAT .AS / JANUARY, /
F25E- 00          5490      .DA #$00
F25F- 20 46 45
F262- 42 55 41
F265- 52 59 2C
F268- 20          5500      .AS / FEBUARY, /
F269- 00          5510      .DA #$00
F26A- 20 4D 41
F26D- 52 43 48
F270- 2C 20      5520      .AS / MARCH, /
F272- 00          5530      .DA #$00
F273- 20 41 50
F276- 52 49 4C
F279- 2C 20      5540      .AS / APRIL, /

```

F27B-	00		5550	.DA ##00
F27C-	20	4D 41		
F27F-	59	2C 20	5560	.AS / MAY, /
F282-	00		5570	.DA ##00
F283-	20	4A 55		
F286-	4E	45 2C		
F289-	20		5580	.AS / JUNE, /
F28A-	00		5590	.DA ##00
F28B-	20	4A 55		
F28E-	4C	59 2C		
F291-	20		5600	.AS / JULY, /
F292-	00		5610	.DA ##00
F293-	20	41 55		
F296-	47	55 53		
F299-	54	2C 20	5620	.AS / AUGUST, /
F29C-	00		5630	.DA ##00
F29D-	20	53 45		
F2A0-	50	54 45		
F2A3-	4D	42 45		
F2A6-	52	2C 20	5640	.AS / SEPTEMBER, /
F2A9-	00		5650	.DA ##00
F2AA-	20	4F 43		
F2AD-	54	4F 42		
F2B0-	45	52 2C		
F2B3-	20		5660	.AS / OCTOBER, /
F2B4-	00		5670	.DA ##00
F2B5-	20	4E 4F		
F2B8-	56	45 4D		
F2BB-	42	45 52		
F2BE-	2C	20	5680	.AS / NOVEMBER, /
F2C0-	00		5690	.DA ##00
F2C1-	20	44 45		
F2C4-	43	45 4D		
F2C7-	42	45 52		
F2CA-	2C	20	5700	.AS / DECEMBER, /
F2CC-	00		5710	.DA ##00
			5720	*-----
			5730	* SET CLOCK ROUTINE
			5740	*-----
F2CD-	20	D3 F2	5750	SETIME JSR SETIM0
F2D0-	4C	35 FA	5760	JMP GETCMD
F2D3-	A9	1F	5770	SETIM0 LDA #CLKFMT
F2D5-	85	52	5780	STA MSGNUM
F2D7-	A9	F3	5790	LDA /CLKFMT
F2D9-	85	53	5800	STA MSGNUM+1
F2DB-	20	87 F5	5810	JSR MSGSND
F2DE-	A9	3F	5820	SETIM2 LDA #'?
F2E0-	8D	5B 05	5830	STA PROMPT
F2E3-	20	43 FA	5840	JSR GETLIN
F2E6-	64	2E	5850	STZ BUFCNT
F2E8-	A2	06	5860	LDX ##06
F2EA-	A4	2E	5870	LDY BUFCNT
F2EC-	DA		5880	SETIM3 PHX
F2ED-	B9	00 02	5890	SETIM4 LDA LINBUF,Y
F2F0-	85	35	5900	STA TEMP04
F2F2-	C8		5910	INY
F2F3-	F0	24	5920	BEG SETIM7
F2F5-	20	78 F3	5930	JSR VALBCD
F2F8-	F0	F3	5940	BEG SETIM4
F2FA-	B9	00 02	5950	SETIM5 LDA LINBUF,Y

F2FD-	C8		5960	INY
F2FE-	F0	19	5970	BEQ SETIM7
F300-	20	78 F3	5980	JSR VALBCD
F303-	F0	F5	5990	BEQ SETIM5
F305-	AA		6000	TAX
F306-	A5	35	6010	LDA TEMP04
F308-	20	E0 F3	6020	JSR ASCHEX
F30B-	FA		6030	PLX
F30C-	95	25	6040	STA TIMBUF,X
F30E-	CA		6050	DEX
F30F-	10	DB	6060	BPL SETIM3
F311-	A9	3E	6070	LDA #'>
F313-	8D	5B 05	6080	STA PROMPT
F316-	4C	6C FE	6090	JMP SETCLK
F319-	FA		6100	SETIM7 PLX
F31A-	20	1C F4	6110	JSR ERROR
F31D-	80	B4	6120	BRA SETIM0
F31F-	53	45 54		
F322-	20	43 4C		
F325-	4F	43 4B		
F328-	20	49 4E		
F32B-	20	46 4F		
F32E-	4C	4C 4F		
F331-	57	49 4E		
F334-	47	20 46		
F337-	4F	52 4D		
F33A-	41	54 3A	6130	CLKFMT .AS /SET CLOCK IN FOLLOWING FORMAT:/
F33D-	0D		6140	.HS 0D
F33E-	20	59 59		
F341-	59	59 2D		
F344-	4D	4D 2D		
F347-	44	44 2D		
F34A-	48	48 3A		
F34D-	4D	4D 3A		
F350-	53	53	6150	.AS / YYYY-MM-DD HH:MM:SS/
F352-	0D	0D	6160	.HS 0D0D
			6170	*-----
			6180	* CHECK FOR NEW YEARS AND INC
			6190	* YEAR BUFFER
			6200	*-----
F354-	20	89 FE	6210	NWYEAR JSR GETCLK
F357-	A5	29	6220	LDA TIMBUF+4
F359-	29	1F	6230	AND ##1F
F35B-	C9	12	6240	CMP ##12
F35D-	D0	03	6250	BNE NWYR1
F35F-	C7	51	6260	SMB BIT04,CPUFLG
F361-	60		6270	RTS
F362-	4F	51 11	6280	NWYR1 BBR BIT04,CPUFLG,NWYR2
F365-	47	51	6290	RMB BIT04,CPUFLG
F367-	F8		6300	SED
F368-	18		6310	CLC
F369-	A9	01	6320	LDA ##01
F36B-	65	2A	6330	ADC YEARBF
F36D-	85	2A	6340	STA YEARBF
F36F-	D0	05	6350	BNE NWYR2
F371-	18		6360	CLC
F372-	A9	01	6370	LDA ##01
F374-	65	2B	6380	ADC YEARBF+1
F376-	D8		6390	NWYR2 CLD
F377-	60		6400	RTS

```

6410 *-----
6420 * VALID BCD # IN ASCII?
6430 * RETURN 0 IN 'A' IF NOT
6440 *-----
F378- C9 30 6450 VALBCD CMP ##30
F37A- 30 05 6460 BMI VBCD2
F37C- C9 3A 6470 CMP ##3A
F37E- 10 01 6480 BPL VBCD2
F380- 60 6490 RTS
F381- A9 00 6500 VBCD2 LDA ##00
F383- 60 6510 RTS
6520 *-----
6530 * CONVERT BCD TO BINARY
6540 * BCD NUMBER IN 'A'
6550 * BINARY NUMBER RETURNED IN 'A'
6560 *-----
F384- A8 6570 BCDBIN TAY
F385- 29 F0 6580 AND ##F0
F387- 4A 6590 LSR
F388- 85 31 6600 STA TEMP00
F38A- 4A 6610 LSR
F38B- 4A 6620 LSR
F38C- 18 6630 CLC
F38D- 65 31 6640 ADC TEMP00
F38F- 85 31 6650 STA TEMP00
F391- 98 6660 TYA
F392- 29 0F 6670 AND ##0F
F394- 18 6680 CLC
F395- 65 31 6690 ADC TEMP00
F397- 60 6700 RTS
6710 *-----
6720 * RANGE CHECK
6730 * SEE IF VALID ASCII-HEX CHAR
6740 * AND CONVERT IT
6750 *-----
F398- C9 30 6760 RNGCHK CMP ##30
F39A- 30 13 6770 BMI HEXERR
F39C- C9 3A 6780 CMP ##3A
F39E- 10 03 6790 BPL RCHK01
F3A0- 29 0F 6800 AND ##0F
F3A2- 60 6810 RTS
F3A3- C9 41 6820 RCHK01 CMP ##41
F3A5- 30 08 6830 BMI HEXERR
F3A7- C9 47 6840 CMP ##47
F3A9- 10 04 6850 BPL HEXERR
F3AB- 38 6860 SEC
F3AC- E9 37 6870 SBC ##37
F3AE- 60 6880 RTS
F3AF- 4C 10 F4 6890 HEXERR JMP ERRORX
6900 *-----
6910 * OUTPUT HEX BYTE AT ADDR01
6920 *-----
F3B2- A0 01 6930 OUTBYT LDY #1
F3B4- 20 6E FC 6940 JSR SPCOUT
F3B7- B2 3F 6950 LDA (ADDR01) GET A HEX BYTE
6960 *-----
6970 * OUTPUT HEX CHARACTER TO LCD
6980 * HEX CHAR IN 'A'
6990 *-----
F3B9- 8D 5E 05 7000 HEXOUT STA HEXCHR

```

F3BC-	4A		7010	LSR
F3BD-	4A		7020	LSR
F3BE-	4A		7030	LSR
F3BF-	4A		7040	LSR
F3C0-	20	D5 F3	7050	JSR HEXASC
F3C3-	20	A0 FA	7060	JSR OUTCHR
F3C6-	AD	5E 05	7070	LDA HEXCHR
F3C9-	29	0F	7080	AND #\$0F
F3CB-	20	D5 F3	7090	JSR HEXASC
F3CE-	20	A0 FA	7100	JSR OUTCHR
F3D1-	AD	5E 05	7110	LDA HEXCHR
F3D4-	60		7120	RTS
			7130	*-----
			7140	* HEX TO ASCII CONVERSION
			7150	* HEX NIBBLE IN 'A'
			7160	* ASCII CHAR RETURNED IN 'A'
			7170	*-----
F3D5-	18		7180	HEXASC CLC
F3D6-	69	30	7190	ADC #\$30
F3D8-	C9	3A	7200	CMP #\$3A
F3DA-	30	03	7210	BMI HEXAS1
F3DC-	18		7220	CLC
F3DD-	69	07	7230	ADC #\$07
F3DF-	60		7240	HEXAS1 RTS
			7250	*-----
			7260	* ASCII TO HEX CONVERSION
			7270	* ASCII CHARS IN 'A' AND 'X'
			7280	* HEX BYTE RETURNED IN 'A'
			7290	*-----
F3E0-	20	98 F3	7300	ASCHEX JSR RNGCHK
F3E3-	0A		7310	ASL
F3E4-	0A		7320	ASL
F3E5-	0A		7330	ASL
F3E6-	0A		7340	ASL
F3E7-	8D	5E 05	7350	STA HEXCHR
F3EA-	8A		7360	TXA
F3EB-	20	98 F3	7370	JSR RNGCHK
F3EE-	0D	5E 05	7380	ORA HEXCHR
F3F1-	8D	5E 05	7390	STA HEXCHR
F3F4-	60		7400	RTS
			7410	*-----
			7420	* OUTPUT HEX VALUES IN 'A' & 'X'
			7430	*-----
F3F5-	20	B9 F3	7440	PRNTAX JSR HEXOUT
F3F8-	8A		7450	TXA
F3F9-	4C	B9 F3	7460	JMP HEXOUT
			7470	*-----
			7480	* DISPLAY CURRENT ADDRESS
			7490	*-----
F3FC-	A9	24	7500	DSPADD LDA #'\$
F3FE-	20	A0 FA	7510	JSR OUTCHR
F401-	A5	40	7520	LDA ADDR01+1
F403-	A6	3F	7530	LDX ADDR01
F405-	20	F5 F3	7540	JSR PRNTAX
F408-	A9	20	7550	LDA #\$20
F40A-	4C	A0 FA	7560	JMP OUTCHR
F40D-	4C	35 FA	7570	JMP GETCMD
			7580	*-----
			7590	* ERROR ROUTINE
			7600	*-----

F410-	20	1C	F4	7610	ERRORX	JSR	ERROR	
F413-	20	21	E6	7620	ERRORZ	JSR	RESMSP	
F416-	A2	FF		7630		LDX	##FF	
F418-	9A			7640		TXS		
F419-	4C	35	FA	7650		JMP	GETCMD	
F41C-	A9	27		7660	ERROR	LDA	#ERRMSG	
F41E-	85	52		7670		STA	MSGNUM	
F420-	A9	F4		7680		LDA	/ERRMSG	
F422-	85	53		7690		STA	MSGNUM+1	
F424-	4C	87	F5	7700		JMP	MSGEND	
F427-	20	2A	2A					
F42A-	2A	20	45					
F42D-	52	52	4F					
F430-	52	20	2A					
F433-	2A	2A		7710	ERRMSG	.AS	/ *** ERROR ***/	
F435-	07	0D	00	7720		.HS	070D00	
				1210		.IN	ROMON.V1.1.PART6	
				1000	*-----			
				1010	* SERIAL COMMUNICATIONS			
				1020	* UPLOAD/DOWNLOAD ROUTINES			
				1030	*-----			
F438-	A9	02		1040	RMTCOM	LDA	##02	SET UP FOR XFERS VIA RF MODEM
F43A-	8D	6C	05	1050		STA	PORTNM	
F43D-	80	07		1060		BRA	SERXFR	
F43F-	A9	03		1070	SERCOM	LDA	##03	SET UP FOR XFERS VIA EXTERNAL SERIAL
F441-	8D	6C	05	1080		STA	PORTNM	
F444-	80	00		1090		BRA	SERXFR	
F446-	64	36		1100	SERXFR	STZ	TEMP05	SEND INITIAL MESSAGE
F448-	20	74	F4	1110		JSR	XFROUT	SEND IT
F44B-	A2	0B		1120	SERXF1	LDX	##0B	GET 10 CHARS
F44D-	86	36		1130	SERXF2	STX	TEMP05	
F44F-	20	20	F5	1140		JSR	SERPIN	GET ONE CHAR AT A TIME
F452-	A6	36		1150		LDX	TEMP05	
F454-	9D	00	03	1160		STA	MSGBUF,X	AND SAVE IT
F457-	CA			1170		DEX		
F458-	D0	F3		1180		BNE	SERXF2	ALL 10 GOT?
F45A-	20	E9	F4	1190		JSR	GETXFA	CONVERT ADDRESSES
F45D-	AD	0B	03	1200		LDA	MSGBUF+##0B	SEE WHAT COMMAND WAS
F460-	C9	53		1210		CMP	##'S	WAS IT SEND
F462-	F0	1F		1220		BEQ	SERSND	
F464-	C9	52		1230		CMP	##'R	WAS IT RECEIVE?
F466-	F0	54		1240		BEQ	SERRCV	
F468-	C9	45		1250		CMP	##'E	END LINK?
F46A-	D0	03		1260		BNE	SERXF3	
F46C-	4C	35	FA	1270		JMP	GETCMD	
F46F-	20	1C	F4	1280	SERXF3	JSR	ERROR	THEN IT WAS ERROR
F472-	80	D2		1290		BRA	SERXFR	DO IT ALL OVER
F474-	A6	36		1300	XFROUT	LDX	TEMP05	
F476-	BD	3F	F5	1310		LDA	SERMSG,X	SEND MESSAGE OUT SERIAL PORT
F479-	F0	07		1320		BEQ	XFR0T1	MESSAGE DONE?
F47B-	20	2D	F5	1330		JSR	SERPOT	SEND A CHAR
F47E-	E6	36		1340		INC	TEMP05	
F480-	80	F2		1350		BRA	XFROUT	
F482-	60			1360	XFR0T1	RTS		
F483-	A2	34		1370	SERSND	LDX	##34	SEND READY MSG
F485-	86	36		1380		STX	TEMP05	
F487-	20	74	F4	1390		JSR	XFROUT	
F48A-	20	20	F5	1400		JSR	SERPIN	GET CR FROM HOST
F48D-	B2	3F		1410	SERSN1	LDA	(ADDR01)	GET DATA TO SEND
F48F-	8D	5E	05	1420		STA	HEXCHR	MAKE TWO ASCII CHARS OUT OF EACH BYTE

F492-	4A	1430	LSR	
F493-	4A	1440	LSR	
F494-	4A	1450	LSR	
F495-	4A	1460	LSR	
F496-	20 D5 F3	1470	JSR HEXASC	CONVERT TO ASCII
F499-	20 2D F5	1480	JSR SERPOT	SEND IT
F49C-	AD 5E 05	1490	LDA HEXCHR	GET LOW NIBBLE
F49F-	29 0F	1500	AND #\$0F	
F4A1-	20 D5 F3	1510	JSR HEXASC	
F4A4-	20 2D F5	1520	JSR SERPOT	AND SEND IT
F4A7-		1530	>INCD ADDR01	NEXT ADDRESS
F4A7-	E6 3F	0000>	INC ADDR01	
F4A9-	D0 02	0000>	BNE :1	
F4AB-	E6 40	0000>	INC ADDR01+1	
		0000>	:1	
F4AD-		1540	>CMPD ADDR01,ADDR02	
F4AD-	A5 40	0000>	LDA ADDR01+1	
F4AF-	C5 42	0000>	CMP ADDR02+1	
F4B1-	D0 04	0000>	BNE :1	
F4B3-	A5 3F	0000>	LDA ADDR01	
F4B5-	C5 41	0000>	CMP ADDR02	
		0000>	:1	
F4B7-	D0 D4	1550	BNE SERSN1	KEEP GOING IF NOT DONE
F4B9-	4C 46 F4	1560	JMP SERXFR	ELSE START OVER
F4BC-	A2 20	1570	SERRCV LDX #\$20	SEND READY MSG
F4BE-	86 36	1580	STX TEMP05	
F4C0-	20 74 F4	1590	JSR XFRUT	
F4C3-	20 20 F5	1600	JSR SERPIN	GET CR FROM HOST
F4C6-	20 20 F5	1610	SERRC1 JSR SERPIN	GET A CHAR
F4C9-	48	1620	PHA	
F4CA-	20 20 F5	1630	JSR SERPIN	GET LOW NIBBLE
F4CD-	AA	1640	TAX	
F4CE-	68	1650	PLA	
F4CF-	20 E0 F3	1660	JSR ASCHEX	CONVERT
F4D2-	92 3F	1670	STA (ADDR01)	AND SAVE
F4D4-		1680	>INCD ADDR01	NEXT ADDRESS
F4D4-	E6 3F	0000>	INC ADDR01	
F4D6-	D0 02	0000>	BNE :1	
F4D8-	E6 40	0000>	INC ADDR01+1	
		0000>	:1	
F4DA-		1690	>CMPD ADDR01,ADDR02	
F4DA-	A5 40	0000>	LDA ADDR01+1	
F4DC-	C5 42	0000>	CMP ADDR02+1	
F4DE-	D0 04	0000>	BNE :1	
F4E0-	A5 3F	0000>	LDA ADDR01	
F4E2-	C5 41	0000>	CMP ADDR02	
		0000>	:1	
F4E4-	D0 E0	1700	BNE SERRC1	LOOP TIL DONE
F4E6-	4C 46 F4	1710	JMP SERXFR	THEN START OVER
F4E9-	A0 0A	1720	GETXFA LDY #\$0A	
F4EB-	20 11 F5	1730	JSR FTCHBT	GET 2 CHARS AND CONVERT TO BYTE
F4EE-	85 40	1740	STA ADDR01+1	USE AS ADDRESS
F4F0-	88	1750	DEY	
F4F1-	20 11 F5	1760	JSR FTCHBT	
F4F4-	85 3F	1770	STA ADDR01	
F4F6-	A0 05	1780	LDY #\$05	
F4F8-	20 11 F5	1790	JSR FTCHBT	
F4FB-	85 42	1800	STA ADDR02+1	
F4FD-	88	1810	DEY	
F4FE-	20 11 F5	1820	JSR FTCHBT	

F501-	85	41	1830	STA	ADDR02	
F503-	18		1840	CLC		
F504-	A5	41	1850	LDA	ADDR02	
F506-	65	3F	1860	ADC	ADDR01	ADD LENGHT TO START ADDRESS
F508-	85	41	1870	STA	ADDR02	
F50A-	A5	42	1880	LDA	ADDR02+1	
F50C-	65	40	1890	ADC	ADDR01+1	
F50E-	85	42	1900	STA	ADDR02+1	
F510-	60		1910	RTS		
F511-	B9	00 03	1920	FTCHBT	LDA MSGBUF,Y	
F514-	85	36	1930	STA	TEMP05	
F516-	88		1940	DEY		
F517-	B9	00 03	1950	LDA	MSGBUF,Y	
F51A-	AA		1960	TAX		
F51B-	A5	36	1970	LDA	TEMP05	
F51D-	4C	E0 F3	1980	JMP	ASCHEX	
F520-	AD	6C 05	1990	SERPIN	LDA PORTNM	GET THE PORT
F523-	C9	02	2000	CMP	#\$02	RF MODEM?
F525-	F0	03	2010	BEQ	SERPN1	YES GET CHAR
F527-	4C	1D E5	2020	JMP	EXTIN	
F52A-	4C	00 E5	2030	SERPN1	JMP RFMIN	
F52D-	AE	6C 05	2040	SERPOT	LDX PORTNM	GET PORT
F530-	E0	02	2050	CPX	#\$02	RF MODEM?
F532-	F0	08	2060	BEQ	SERPT1	
F534-	C9	03	2070	CMP	#\$03	IS CHAR ETX
F536-	D0	01	2080	BNE	SERPT2	THEN DON'T SEND IT
F538-	60		2090	RTS		
F539-	4C	13 E5	2100	SERPT2	JMP EXTOUT	SEND CHAR TO EXTERNAL SERIAL
F53C-	4C	EE E4	2110	SERPT1	JMP RFMOUT	SEND CHAR TO REMOTE
F53F-	0D	0A	2120	SERMSG	.HS 0D0A	
F541-	20	2A 52				
F544-	4F	42 4F				
F547-	54	20 52				
F54A-	45	41 44				
F54D-	59	20 46				
F550-	4F	52 20				
F553-	43	4F 4D				
F556-	4D	41 4E				
F559-	44	2E	2130	.AS	/ *ROBOT READY FOR COMMAND./	
F55B-	0D	0A 2B				
F55E-	03	00	2140	.HS	0D0A2B0300	
F560-	20	52 45				
F563-	41	44 59				
F566-	20	54 4F				
F569-	20	52 45				
F56C-	43	45 49				
F56F-	56	45	2150	.AS	/ READY TO RECEIVE/	
F571-	0D	0A 03				
F574-	00		2160	.HS	0D0A0300	
F575-	20	52 45				
F578-	41	44 59				
F57B-	20	54 4F				
F57E-	20	53 45				
F581-	4E	44	2170	.AS	/ READY TO SEND/	
F583-	0D	0A 03				
F586-	00		2180	.HS	0D0A0300	
			2190	*-----		
			2200	* MESSAGE OUTPUT ROUTINE		
			2210	* SENDS MESSAGE STRING POINTED		
			2220	* AT BY MSGNUM TO CURRENT OUTPUT		

```

2230 * DEVICE. #00 TERMINATES MESSAGE.
2240 *-----
F587- B2 52 2250 MSGSND LDA (MSGNUM)
F589- F0 0B 2260      BEQ MSGEXT
F58B- 20 A0 FA 2270      JSR OUTCHR
F58E- 2280      >INCD MSGNUM
F58E- E6 52 0000>      INC MSGNUM
F590- D0 02 0000>      BNE :1
F592- E6 53 0000>      INC MSGNUM+1
0000> :1
F594- 80 F1 2290      BRA MSGSND
F596- 60 2300 MSGEXT RTS
2310 *-----
2320 * GET.SON.RNG
2330 *-----
2340 GET.SON.RNG
F597- 20 A0 F5 2350      JSR EN.COLSON
F59A- 20 D2 F5 2360      JSR GET.COL.RNG
F59D- 4C C9 F5 2370      JMP DS.COLSON
2380 *-----
2390 * SUBROUTINE EN.COLSON
2400 *
2410 * ON ENTRY: (1) X=0 HEAD SONAR
2420 *              (2) X=1 RIGHT TOP FRONT
2430 *              (3) X=2 RIGHT BOTTOM FRONT
2440 *              (4) X=3 LEFT TOP FRONT
2450 *              (5) X=4 LEFT BOTTOM FRONT
2460 *              (6) X=5 RIGHT SIDE
2470 *              (7) X=6 LEFT SIDE
2480 *              (8) X=7 FRONT CENTER
2490 *              (9) X=8 BACK
2500 *              (10) X=9 SPARE
2510 * ON EXIT: (1) C=0 IF SONAR ENABLED
2520 *              (2) C=1 IF SONAR INVALID
2530 *-----
2540 EN.COLSON
F5A0- E0 0A 2550      CPX #XDU CRS      CK FOR VALID COL SON NO
F5A2- 90 01 2560      BCC .1
F5A4- 60 2570      RTS
F5A5- 8D BF F5 2580 .1      LDA SONARS,X GET BIT PATTERN FOR THIS SONAR
F5A8- 8D 62 E0 2590      STA BITOUT+2
F5AB- A9 04 2600      LDA #BIT2      DISCONNECT GROUNDS
F5AD- 0C 60 E0 2610      TSB BITOUT
F5B0- A9 08 2620      LDA #ENBSON      TRN ON SON, VSW OFF
F5B2- 0D 60 E0 2630      ORA BITOUT
F5B5- 8D 60 E0 2640      STA BITOUT
F5B8- A0 0F 2650      LDY #15
F5BA- 20 48 FC 2660      JSR PAUSE
F5BD- 18 2670      CLC      RTN W C=0
F5BE- 60 2680      RTS
F5BF- F0 F3 F5
F5C2- F4 F6 F2
F5C5- F1 F7 F8
F5C8- F9 2690 SONARS .HS F0F3F5F4F6F2F1F7F8F9
2700 *-----
2710 * SUBROUTINE DS.COLSON
2720 *
2730 * ON EXIT: (1) COL SONARS OFF
2740 *-----
2750 DS.COLSON

```

```

F5C9- A9 F7      2760      LDA #SONOFF
F5CB- 2D 60 E0    2770      AND BITOUT
F5CE- 8D 60 E0    2780      STA BITOUT
F5D1- 60          2790 .1    RTS
2800 *-----
2810 * SUBROUTINE GET.COL.RNG
2820 *
2830 * ON ENTRY: (1) COL SON DESIRED IS ENABLED THRU CALL
2840 *              TO EN.COLSON
2850 * ON EXIT: (1) COL RNG IN AC
2860 *              (2) GET.COL.RNG CAN BE IMMEDIATELY CALLED AGAIN
2870 *-----
2880 GET.COL.RNG
F5D2- 20 11 E6    2890      JSR LOSPD
F5D5- A9 FF      2900      LDA #FF      SET TIMER
F5D7- 8D 1A E0    2910      STA RNGADD+2
F5DA- 8D 1E E0    2920      STA RNGADD+6
F5DD- A9 A9      2930      LDA #A9
F5DF- 8D 1C E0    2940      STA RNGADD+4
F5E2- A9 20      2950      LDA #VSWON    VSW ON
F5E4- 0D 60 E0    2960      ORA BITOUT
F5E7- 8D 60 E0    2970      STA BITOUT
F5EA- A0 64      2980      LDY #100      FOR 100 MS
F5EC- 20 48 FC    2990      JSR PAUSE
F5EF- 20 11 E6    3000      JSR LOSPD
F5F2- A9 49      3010      LDA #49      READ CNTRS
F5F4- 8D 1C E0    3020      STA RNGADD+4
F5F7- 38          3030      SEC
F5F8- A9 FF      3040      LDA #FF
F5FA- ED 1A E0    3050      SBC RNGADD+2
F5FD- 85 63      3060      STA ACM
F5FF- A9 FF      3070      LDA #FF
F601- ED 1E E0    3080      SBC RNGADD+6
F604- 85 64      3090      STA ACH
F606- 64 62      3100      STZ ACL
F608- A9 DF      3110      LDA #VSWOFF
F60A- 2D 60 E0    3120      AND BITOUT
F60D- 8D 60 E0    3130      STA BITOUT
F610- A0 64      3140      LDY #100      PAUSE FOR 100 MS
F612- 4C 48 FC    3150      JMP PAUSE
3160 *-----
3170 * SUBROUTINE GET.AVG.RNG
3180 *-----
3190 GET.AVG.RNG
F615- 9C 6D 05    3200      STZ AVGL0      ZERO SUM
F618- 9C 6E 05    3210      STZ AVGL0+1
F61B- 9C 6F 05    3220      STZ AVGL0+2
F61E- A9 08      3230      LDA #8      TAKE 8 RANGES
F620- 48          3240 .1    PHA
F621- A0 64      3250      LDY #100
F623- 20 48 FC    3260      JSR PAUSE
F626- 20 D2 F5    3270      JSR GET.COL.RNG
F629- 18          3280      CLC      ADD RNG TO SUM
F62A- A5 63      3290      LDA ACM
F62C- 6D 6D 05    3300      ADC AVGL0
F62F- 8D 6D 05    3310      STA AVGL0
F632- A5 64      3320      LDA ACH
F634- 6D 6E 05    3330      ADC AVGL0+1
F637- 8D 6E 05    3340      STA AVGL0+1
F63A- A9 00      3350      LDA #0

```



```

F63C- 6D 6F 05 3360      ADC AVGL0+2
F63F- 8D 6F 05 3370      STA AVGL0+2
F642- 68          3380      PLA
F643- 3A          3390      DEC
F644- D0 DA       3400      BNE .1
F646- A2 03       3410      LDX #3          DIV BY 8
F648- 18          3420      CLC
F649- 6E 6F 05 3430      ROR AVGL0+2
F64C- 6E 6E 05 3440      ROR AVGL0+1
F64F- 6E 6D 05 3450      ROR AVGL0
F652- CA          3460      DEX
F653- D0 F3       3470      BNE .2
F655- 64 62       3480      STZ ACL          AVG TO AC
F657- AD 6D 05 3490      LDA AVGL0
F65A- 85 63       3500      STA ACM
F65C- AD 6E 05 3510      LDA AVGL0+1
F65F- 85 64       3520      STA ACH
F661- 60          3530      RTS

3540 *-----
3550 * SUBROUTINE INIT.HEAD
3560 *
3570 * PURPOSE: MOVE HEAD TO LEFT STOP
3580 *
3590 * ENTRY: HEAD MOTOR SPEED (MSPSTP) SET AS FOLLOWS:
3600 *      MSPSTP  TIME(SECS) STOP TO STOP
3610 *      -----
3620 *           08          10
3630 *           09          11
3640 *           0A          12
3650 *           0B          13.5
3660 *           0C          14.5
3670 *           0D          16
3680 *           0E          17
3690 *           0F          18
3700 *           10          19.5
3710 *           11          20.5
3720 *
3730 * EXIT: (1) INITS CURHD TO ZERO
3740 *        (2) HDINT BIT IS SET
3750 *        (3) HEAD STEPPER MOTOR TURNED OFF
3760 *-----
3770 INIT.HEAD
F662- 87 50       3780      SMB HDINIT,HDSTAT  FLG HEAD INITIALIZED
F664- 9C 6A 05 3790      STZ CURHD          INIT CURHD & FALL THRU
3800 *-----
3810 * SUBROUTINE FIND.LEFT.STOP
3820 *
3830 * PURPOSE: POSITION HEAD AT LEFT STOP
3840 *
3850 * ENTRY: (1) MSPSTP SET TO DESIRED VALUE
3860 *
3870 * EXIT:  (1) HEAD AT LEFT STOP
3880 *        (2) STEPPER MOTOR TURNED OFF
3890 *-----
3900 FIND.LEFT.STOP
F667- 9C 69 05 3910      STZ STEPNO  INIT STEP NO.
F66A- 97 50       3920      SMB DIREC,HDSTAT  SET CCW ROTATION
F66C- AD 67 05 3930      LDA MSPSTP  GET MOTOR SPEED
F66F- 48          3940      PHA          SAVE AS A TIMER
F670- AD 50 E0 3950      LDA BITINP  CK IF AT LEFT STOP

```

```

F673- 29 80      3960      AND #LFTSPB
F675- F0 11      3970      BEQ .3          BR IF SO
F677- A0 01      3980      LDY #1          ELSE PAUSE FOR 1 MS
F679- 20 48 FC   3990      JSR PAUSE
F67C- 68         4000      PLA              CHECK IF DONE WITH THIS STEP
F67D- 3A         4010      DEC
F67E- D0 EF      4020      BNE .1          BR IF NO
F680- 20 BC F6   4030      JSR ONE.STEP    ELSE TAKE A STEP
F683- AD 67 05   4040      LDA MSPSTP     RELOAD TIMER
F686- 80 E7      4050      BRA .1          THEN REPEAT
F688- 68         4060      .3 PLA          FIX STK & FALL THRU TO DS.HDSTPR
4070 *-----*
4080 * SUBROUTINE DS.HDSTPR
4090 *
4100 * PURPOSE: TURN OFF ALL WINDINGS IN STEPPER MOTOR
4110 *
4120 * ENTRY: NO REQ'MNTS
4130 *
4140 * EXIT: (1) BITOUT UPPER NIBBLE SET HIGH
4150 *-----*
4160 DS.HDSTPR
F689- AD 62 E0   4170      LDA BITOUT+2
F68C- 29 0F      4180      AND #$0F
F68E- 09 F0      4190      ORA #$F0
F690- 8D 62 E0   4200      STA BITOUT+2
F693- 60         4210      RTS
4220 *-----*
4230 * SUBROUTINE POSHD
4240 *
4250 * PURPOSE: MOVE HEAD TO POSITION IN A-REG
4260 *
4270 * ENTRY: (1) A=DESIRED HD POS ($00-$EF)
4280 * EXIT: (1) C=0 IF ALL'S WELL
4290 *        (2) C=1 IF ASKING FOR HD POSN $F0-$FF
4300 *
4310 * HD WILL BE INITIALIZED IF NOT DONE PREVIOUSLY
4320 *-----*
F694- 8F 50 05   4330 POSHD  BBS HDINIT,HDSTAT,.1 BR IF HD INITIALIZED
F697- 48         4340      PHA              ELSE INIT HEAD
F698- 20 62 F6   4350      JSR INIT.HEAD
F69B- 68         4360      PLA
F69C- C9 F0      4370      .1 CMP #RGTSPT+1 CK IF ASKING FOR $F0-$FF
F69E- 90 01      4380      BCC .2          BR IF NO
F6A0- 60         4390      RTS              ELSE RTN W C=1
F6A1- 17 50      4400      .2 RMB DIREC,HDSTAT SET CW
F6A3- 38         4410      SEC              GET DESIRED-CURHD
F6A4- ED 6A 05   4420      SBC CURHD
F6A7- B0 05      4430      BCS .3          BR IF ZERO OR POS
F6A9- 49 FF      4440      EOR #$FF       ELSE NEGATE
F6AB- 1A         4450      INC
F6AC- 97 50      4460      SMB DIREC,HDSTAT AND SET CCW
F6AE- D0 02      4470      .3 BNE .4          BR IF STEPS REQ'D
F6B0- 18         4480      CLC              ELSE RTN W C=0
F6B1- 60         4490      RTS
F6B2- 48         4500      .4 PHA              SAVE #OF STEPS REQUIRED
F6B3- 20 E5 F6   4510      JSR STEP1.5DEGRS
F6B6- 68         4520      PLA
F6B7- 3A         4530      DEC              CK IF DONE
F6B8- D0 F8      4540      BNE .4          BR IF NO
F6BA- 80 CD      4550      BRA DS.HDSTPR  ELSE RTN VIA DS.HDSTPR

```

```

4560 *-----
4570 * SUBROUTINE ONE.STEP
4580 *
4590 * PURPOSE: STEP HEAD ONE STEP IN
4600 *           DIRECTION SPECIFIED BY
4610 *           'DIREC' BIT (0=CW,1=CCW)
4620 *
4630 * ENTRY: (1) STEPNO SETUP ON FIRST CALL TO ROUTINE
4640 *
4650 * EXIT:  (1) HEAD MOVED ONE PHYSICAL STEP
4660 *         (2) STEPNO IS UPDATED
4670 *
4680 *-----
4690 ONE.STEP
F6BC- AE 69 05 4700      LDX STEPNO    GET LAST STEP
F6BF- 1F 50 09 4710      BBR DIREC,HDSTAT,.1 BR IF CW DIRECTION
F6C2- E8      4720      INX          ELSE, INC TO NEXT STEP
F6C3- E0 04   4730      CPX #4      CK IF TOO FAR
F6C5- D0 0B   4740      BNE .2      BR IF NO
F6C7- A2 00   4750      LDX #0      ELSE RESET STEP
F6C9- 80 07   4760      BRA .2
F6CB- CA      4770 .1    DEX          DEC STEP
F6CC- E0 FF   4780      CPX #$FF    CK IF TOO FAR
F6CE- D0 02   4790      BNE .2      BR IF NO
F6D0- A2 03   4800      LDX #3      ELSE RESET STEP
F6D2- 8E 69 05 4810 .2    STX STEPNO  UPDATE LAST STEP
F6D5- AD 62 E0 4820      LDA BITOUT+2 GET PORT IMAGE
F6D8- 29 0F   4830      AND #$0F    STRIP OFF HI NIBBLE
F6DA- 1D E1 F6 4840      ORA STPSEQ,X ADD IN NEW STEP
F6DD- 8D 62 E0 4850      STA BITOUT+2
F6E0- 60      4860      RTS
F6E1- 50 90 A0
F6E4- 60      4870 STPSEQ .HS 5090A060
4880 *-----
4890 * SUBROUTINE STEP1.5DEGRS
4900 *
4910 * PURPOSE: STEP HEAD 5 PHYSICAL STEPS (ONE LOGICAL STEP)
4920 *           AND LOOK FOR BEACONS AND DOOR EDGES
4930 *           WHILE STEPPING.
4940 *
4950 * ENTRY: (1) HEAD INIT'D
4960 *         (2) DIREC TO STEP SPECIFIED BY 'DIREC'
4970 * EXIT:  (1) HEAD MOVED 5 PHYSICAL STEP IN DIRECTION
4980 *           SPECIFIED BY DIREC BIT
4990 *         (2) DORREF BIT OF HDSTAT
5000 *           0 = NO DOOR REFLECTOR FOUND
5010 *           1 = DOOR REFLECTOR FOUND
5020 *         (3) BECREF BIT OF HDSTAT
5030 *           0 = NO BEACON FOUND
5040 *           1 = BEACON FOUND
5050 *         (4) HEAD STOPS
5060 *           CARRY SET IF HIT
5070 *           CARRY CLEAR IF OK
5080 *-----
5090 STEP1.5DEGRS
F6E5- 47 50   5100      RMB BECREF,HDSTAT NO BEACON DETECTED THIS STEP
F6E7- 57 50   5110      RMB DORREF,HDSTAT NO DOOR REFLECT THIS STEP
F6E9- A9 05   5120      LDA #5      TAKE 5 PHYS STEPS
F6EB- 48      5130 .1    PHA
F6EC- 20 BC F6 5140      JSR ONE.STEP TAKE A PHYS STEP

```

F6EF-	CF 50 09	5150	BBS BECREF,HDSTAT,.2	
F6F2-	AD 50 E0	5160	LDA BITINP	CHK IF BEACON DET
F6F5-	29 02	5170	AND #BECBIT	
F6F7-	D0 02	5180	BNE .2	BR IF NO
F6F9-	C7 50	5190	SMB BECREF,HDSTAT	ELSE FLAG BEACON
F6FB-	DF 50 09	5200	.2 BBS DORREF,HDSTAT,.3	
F6FE-	AD 50 E0	5210	LDA BITINP	CHK IF DOOR REFLECT
F701-	29 04	5220	AND #DORBIT	
F703-	D0 02	5230	BNE .3	BR IF NO
F705-	D7 50	5240	SMB DORREF,HDSTAT	FLAG DOOR
F707-	AC 67 05	5250	.3 LDY MSPSTP	DELY 'MSPSTP' MS
F70A-	20 48 FC	5260	JSR PAUSE	
F70D-	68	5270	PLA	
F70E-	3A	5280	DEC	
F70F-	D0 DA	5290	BNE .1	
F711-	9F 50 05	5300	BBS DIREC,HDSTAT,.4	
F714-	EE 6A 05	5310	INC CURHD	
F717-	80 03	5320	BRA .5	
F719-	CE 6A 05	5330	.4 DEC CURHD	
F71C-	18	5340	.5 CLC	
F71D-	60	5350	RTS	
		5360	*-----	
F71E-	AD 61 E0	5370	GETMSG LDA BITOUT+1	SAVE CUR BNK
F721-	48	5380	PHA	
F722-	29 E3	5390	AND #E3	
F724-	09 18	5400	ORA #18	SWITCH TO BANK 6
F726-	8D 61 E0	5410	STA BITOUT+1	
F729-	DA	5420	PHX	SAVE MSG#
F72A-	AD 47 05	5430	LDA CATN	GET CATALOG#
F72D-	0A	5440	ASL	USE AS INDEX TO GET BASE ADR
F72E-	A8	5450	TAY	OF MSGS
F72F-	B9 00 C0	5460	LDA CATLOG,Y	
F732-	85 77	5470	STA MSGPTR	
F734-	C8	5480	INY	
F735-	B9 00 C0	5490	LDA CATLOG,Y	
F738-	85 78	5500	STA MSGPTR+1	
F73A-	68	5510	PLA	GET BACK MSG#
F73B-	0A	5520	ASL	USE AS INDEX TO GET MSG ADR
F73C-	A8	5530	TAY	
F73D-	B1 77	5540	LDA (MSGPTR),Y	
F73F-	48	5550	PHA	
F740-	C8	5560	INY	
F741-	B1 77	5570	LDA (MSGPTR),Y	
F743-	85 78	5580	STA MSGPTR+1	
F745-	68	5590	PLA	
F746-	85 77	5600	STA MSGPTR	
		5610	*	
		5620	* GET MSG, UNPACK AND PUT IN SPEECH BUF	
		5630	*	
F748-	A0 00	5640	UP1 LDY #0	GET 3 BYTES OF PACKED DATA
F74A-	B2 77	5650	UP2 LDA (MSGPTR)	
F74C-	99 49 05	5660	STA PACKBF,Y	
F74F-	E6 77	5670	INC MSGPTR	
F751-	D0 02	5680	BNE UP3	
F753-	E6 78	5690	INC MSGPTR+1	
F755-	C8	5700	UP3 INY	
F756-	C0 03	5710	CPY #3	
F758-	D0 F0	5720	BNE UP2	FALL THRU W Y=3
F75A-	9C 50 05	5730	UP4 STZ REG	UNPACK THE 3 BYTES IN PACKBF
F75D-	A2 06	5740	LDX #6	

F75F-	4E 49 05	5750	UP5	LSR	PACKBF	
F762-	6E 4A 05	5760		ROR	PACKBF+1	
F765-	6E 4B 05	5770		ROR	PACKBF+2	
F768-	6E 50 05	5780		ROR	REG	
F76B-	CA	5790		DEX		
F76C-	D0 F1	5800		BNE	UP5	
F76E-	AD 50 05	5810		LDA	REG	
F771-	4A	5820		LSR		
F772-	4A	5830		LSR		
F773-	99 4C 05	5840		STA	UPACBF,Y	AND PUT INTO THE 4 BYTE UNPACK BUF
F776-	88	5850		DEY		
F777-	10 E1	5860		BPL	UP4	
		5870	*			
		5880	*	CONVERT UNPACKED BUFFER TO ASCII AND PUT IN SPEECH BUF		
		5890	*			
F779-	C8	5900		INY		Y NOW 0
F77A-	B9 4C 05	5910	UP6	LDA	UPACBF,Y	GET PACKER CODE
F77D-	AA	5920		TAX		USE AS INDEX
F77E-	BD AD F7	5930		LDA	ASCII,X	TO GET ASCII CHAR
F781-	30 10	5940		BMI	UP7	BR IF LAST CHAR IN MSG
F783-	AE 40 05	5950		LDX	BUFPTR	
F786-	9D 00 03	5960		STA	MSGBUF,X	
F789-	EE 40 05	5970		INC	BUFPTR	
F78C-	C8	5980		INY		
F78D-	C0 04	5990		CPY	#4	
F78F-	D0 E9	6000		BNE	UP6	
F791-	80 B5	6010		BRA	UP1	UNPACK SOME MORE
		6020	*			
		6030	*	FINISH UP		
		6040	*			
F793-	29 7F	6050	UP7	AND	#7F	STRIP OFF HI BIT
F795-	AE 40 05	6060		LDX	BUFPTR	
F798-	9D 00 03	6070		STA	MSGBUF,X	
F79B-	E8	6080		INX		
F79C-	6F 73 06	6090		BBR	SFCFLG,DEMST,UP8	BR IF NO SPACE DESIRED
F79F-	A9 20	6100		LDA	#	
F7A1-	9D 00 03	6110		STA	MSGBUF,X	
F7A4-	E8	6120		INX		
F7A5-	8E 40 05	6130	UP8	STX	BUFPTR	
F7A8-	68	6140		PLA		RESTORE BANK
F7A9-	8D 61 E0	6150		STA	BITOUT+1	
F7AC-	60	6160		RTS		
F7AD-	0D 20 21					
F7B0-	2C 2D 2E					
F7B3-	2F 30 31					
F7B6-	32 33 34					
F7B9-	35 36 37					
F7BC-	38	6170	ASCII	.HS	0D20212C2D2E2F303132333435363738	
F7BD-	39 3F 41					
F7C0-	42 43 44					
F7C3-	45 46 47					
F7C6-	48 49 4A					
F7C9-	4B 4C 4D					
F7CC-	4E	6180		.HS	393F4142434445464748494A4B4C4D4E	
F7CD-	4F 50 51					
F7D0-	52 53 54					
F7D3-	55 56 57					
F7D6-	58 59 5A					
F7D9-	C1 C2 C3					
F7DC-	C4	6190		.HS	4F505152535455565758595A5B5C5D5E5F6061626364	

```

F7DD- C5 C6 C7
F7E0- C8 C9 CB
F7E3- CC CD CE
F7E6- CF D0 D2
F7E9- D3 D4 D7
F7EC- D9          6200      .HS C5C6C7C8C9CBCCCCDCECFD0D2D3D4D7D9
6210 *-----*
F7ED- AE 40 05 6220 MSGXMT LDX BUFPTR          PUT CR AND ETX IN SPEECH BUF
F7F0- A9 0D      6230      LDA #0D
F7F2- 9D 00 03 6240      STA MSGBUF,X
F7F5- E8          6250      INX
F7F6- A9 03      6260      LDA #ETX
F7F8- 9D 00 03 6270      STA MSGBUF,X
F7FB- 9C 40 05 6280      STZ BUFPTR          INIT SPEECH BUF PTR
F7FE- 7F 73 03 6290      BBR CDSFLG,DEMST,.1 DONT CLR SCRIN IF CDSFLG=0
F801- 20 0F E4 6300      JSR CLDISP
F804- 0F 73 05 6310 .1    BBR SPKFLG,DEMST,.2 BR IF NO SPEECH
F807- A9 31      6320      LDA #'1          SET VIOS FOR TXT-TO-SPCH
F809- 20 DC E4 6330      JSR VIOSOT
F80C- AE 40 05 6340 .2    LDX BUFPTR          GET CHAR FM SPEECH BUF
F80F- BD 00 03 6350      LDA MSGBUF,X
F812- C9 03      6360      CMP #ETX          CK IF AT END
F814- F0 15      6370      BEQ .5          BR IF SO
F816- 1F 73 07 6380      BBR DSPFLG,DEMST,.3 BR IF NO SHOW
F819- C9 2F      6390      CMP #'/'          DONT OUTPUT STRESS MARKS
F81B- F0 03      6400      BEQ .3
F81D- 20 A0 FA 6410      JSR OUTCHR          PASS TO CUR OUTPUT DEV
F820- 0F 73 03 6420 .3    BBR SPKFLG,DEMST,.4 BR IF NO SPEECH
F823- 20 DC E4 6430      JSR VIOSOT
F826- EE 40 05 6440 .4    INC BUFPTR          NXT CHAR
F829- 80 E1      6450      BRA .2
F82B- 1F 73 03 6460 .5    BBR DSPFLG,DEMST,.6 BR IF NO DISPLAY
F82E- 20 A0 FA 6470      JSR OUTCHR
F831- 0F 73 06 6480 .6    BBR SPKFLG,DEMST,.7 BR IF NO SPEECH
F834- 5F 73 03 6490      BBR WATFLG,DEMST,.7 BR IF NO WAIT
F837- 20 E6 E4 6500      JSR VIOSIN          WAIT HERE 'TIL MSG SPOKEN
F83A- 60          6510 .7    RTS

```

1230 .IN ROMON.V1.1.PART7

1000 *-----

1010 * MAIN ROUTINE

1020 * INITIALIZE RAM AND VECTORS

1030 * FIND HI MEMORY AND ENABLE

1040 * INTERRUPTS, AND DISPLAY

1050 * INITIAL PROMPT.

1060 *-----

F83B-	78	1070	RESENT SEI	NO INTERRUPTS
F83C-	D8	1080	CLD	
F83D-	A2 FF	1090	LDX #\$FF	
F83F-	9A	1100	TXS	
F840-	20 47 FE	1110	JSR GETFNC	GET FUNCTION KEY
F843-	C9 05	1120	CMP #\$05	WAS IT FUNCTION KEY 5?
F845-	D0 02	1130	BNE MNTR0	NO, DO NORMAL RESET
F847-	64 24	1140	STZ COLDAD	YES, DO COLD START
F849-	A5 24	1150	MNTR0 LDA COLDAD	COLD START?
F84B-	C9 5A	1160	CMP #\$5A	
F84D-	F0 3E	1170	BEQ MNTR10	NO, JUMP TO INIT VECTORS
F84F-	A2 00	1180	LDX #ZERO	YES, DO MEMORY TEST
F851-	74 00	1190	STZ ZERO,X	
F853-	E8	1200	INX	
F854-	74 00	1210	MNTR1 STZ ZERO,X	
F856-	E8	1220	INX	
F857-	F0 02	1230	BEQ MNTR2	
F859-	80 F9	1240	BRA MNTR1	
F85B-	E6 01	1250	MNTR2 INC ZERO+1	
F85D-	A9 5A	1260	MNTR3 LDA #\$5A	
F85F-	92 00	1270	STA (ZERO)	
F861-	EA	1280	NOF	
F862-	EA	1290	NOF	
F863-	B2 00	1300	LDA (ZERO)	
F865-	C9 5A	1310	CMP #\$5A	
F867-	D0 0A	1320	BNE MNTR5	
F869-	A9 00	1330	LDA #\$00	
F86B-	92 00	1340	STA (ZERO)	
F86D-	E6 00	1350	INC ZERO	
F86F-	F0 EA	1360	BEQ MNTR2	
F871-	80 EA	1370	BRA MNTR3	
F873-	A5 01	1380	MNTR5 LDA ZERO+1	
F875-	85 11	1390	STA HIMEM+1	
F877-	86 10	1400	STX HIMEM	
F879-		1410	>DECD HIMEM	SET HIGH MEMORY
F879-	A2 FF	0000>	LDX #\$FF	
F87B-	C6 10	0000>	DEC HIMEM	
F87D-	E4 10	0000>	CPX HIMEM	
F87F-	D0 02	0000>	BNE :1	
F881-	C6 11	0000>	DEC HIMEM+1	
		0000>	:1	
F883-	64 12	1420	STZ LOMEM	
F885-	A9 08	1430	LDA #\$08	
F887-	85 13	1440	STA LOMEM+1	
F889-	A9 5A	1450	LDA #\$5A	
F88B-	85 24	1460	STA COLDAD	SET COLD START INDICATOR
F88D-	A9 C7	1470	MNTR10 LDA #RETINT	SET UP NMI VECTORS
F88F-	85 16	1480	STA NMILOC	
F891-	85 14	1490	STA IRQLOC	
F893-	85 18	1500	STA ALMIRO	
F895-	85 1A	1510	STA CLKIRQ	

F897-	A9	FF	1520	LDA	/RETINT	
F899-	85	17	1530	STA	NMILOC+1	
F89B-	85	15	1540	STA	IRQLOC+1	
F89D-	85	19	1550	STA	ALMIRO+1	
F89F-	85	1B	1560	STA	CLKIRQ+1	
F8A1-	A9	C2	1570	LDA	#LCDOUT	
F8A3-	85	1C	1580	STA	CHROUT	
F8A5-	A9	FA	1590	LDA	/LCDOUT	
F8A7-	85	1D	1600	STA	CHROUT+1	
F8A9-	A9	03	1610	LDA	#RDKEY	
F8AB-	85	1E	1620	STA	CHARIN	
F8AD-	A9	FC	1630	LDA	/RDKEY	
F8AF-	85	1F	1640	STA	CHARIN+1	
F8B1-	A9	09	1650	LDA	#INKEY	
F8B3-	85	20	1660	STA	CHARPL	
F8B5-	A9	FC	1670	LDA	/INKEY	
F8B7-	85	21	1680	STA	CHARPL+1	
F8B9-	A9	35	1690	LDA	#GETCMD	
F8BB-	85	22	1700	STA	USRMON	
F8BD-	A9	FA	1710	LDA	/GETCMD	
F8BF-	85	23	1720	STA	USRMON+1	
F8C1-	A9	00	1730	LDA	#\$00	
F8C3-	85	6F	1740	STA	LIFEAD	
F8C5-	A9	C0	1750	LDA	#\$C0	
F8C7-	85	70	1760	STA	LIFEAD+1	
F8C9-	20	C1 F9	1770	JSR	CLRDSF	CLEAR THE DISPLAY
F8CC-	20	01 FA	1780	JSR	CLRIMG	CLEAR THE MEMORY IMAGE
F8CF-	20	11 E6	1790	JSR	LOSPD	
F8D2-	A2	10	1800	LDX	#\$10	DELAY FOR PPI CHIPS
F8D4-	A0	FF	1810	LDY	#\$FF	
F8D6-	88		1820	DEY		
F8D7-	D0	FD	1830	BNE	DLAYY	
F8D9-	CA		1840	DEX		
F8DA-	D0	F8	1850	BNE	DLY	
F8DC-	A9	80	1860	LDA	#SUBOP	SETUP BIT OUTPUT PORT
F8DE-	8D	63 E0	1870	STA	BITOUT+3	
F8E1-	A9	05	1880	LDA	#SUBP0	
F8E3-	8D	60 E0	1890	STA	BITOUT	
F8E6-	A9	80	1900	LDA	#SUBP1	
F8E8-	8D	61 E0	1910	STA	BITOUT+1	
F8EB-	A9	F0	1920	LDA	#SUBP2	
F8ED-	8D	62 E0	1930	STA	BITOUT+2	
F8F0-	A9	98	1940	LDA	#SUSERP	SETUP BIT SERIAL PORTS
F8F2-	8D	23 E0	1950	STA	SERADD+3	
F8F5-	A9	13	1960	LDA	#ACIARS	RESET ACIA
F8F7-	8D	74 E0	1970	STA	ACRSR	
F8FA-	A9	11	1980	LDA	#ACIASU	SET UP ACIA
F8FC-	8D	74 E0	1990	STA	ACRSR	
F8FF-	20	54 F1	2000	JSR	PIAINP	SET UP PIA
F902-	20	07 E6	2010	JSR	SETHSP	
F905-	A9	50	2020	LDA	#MNMSG	
F907-	85	52	2030	STA	MSGNUM	
F909-	A9	F9	2040	LDA	/MNMSG	
F90B-	85	53	2050	STA	MSGNUM+1	
F90D-	20	87 F5	2060	JSR	MSGEND	
F910-	A2	FF	2070	LDX	#\$FF	
F912-	8E	21 E0	2080	STX	SERADD+1	TURN OFF ALL SERIAL PORTS
F915-	A2	0F	2090	LDX	#\$0F	
F917-	8E	22 E0	2100	STX	SERADD+2	
F91A-	9C	55 05	2110	STZ	LINCNT	SET DISPLAY LINE COUNT

F91D-	A5	29	2120	LDA	TIMBUF+4	
F91F-	D0	04	2130	BNE	MNTR70	
F921-	A9	06	2140	LDA	##06	SETUP CLOCK MODE
F923-	85	2D	2150	STA	CLKMOD	
F925-	AD	61 E0	2160	MNTR70	LDA	BITOUT+1
F928-	29	E3	2170	AND	##E3	
F92A-	09	04	2180	ORA	##04	
F92C-	8D	61 E0	2190	STA	BITOUT+1	
F92F-	AD	00 C0	2200	LDA	\$C000	
F932-	C9	4C	2210	CMF	##4C	
F934-	D0	08	2220	BNE	MNTR80	
F936-	A9	04	2230	LDA	##04	
F938-	8D	53 05	2240	STA	CURBNK	
F93B-	4C	00 C0	2250	JMP	\$C000	
F93E-	AD	61 E0	2260	MNTR80	LDA	BITOUT+1
F941-	29	E3	2270	AND	##E3	
F943-	8D	61 E0	2280	STA	BITOUT+1	
F946-	A5	29	2290	LDA	TIMBUF+4	
F948-	D0	03	2300	BNE	MNTR90	
F94A-	20	D3 F2	2310	JSR	SETIM0	
F94D-	4C	35 FA	2320	MNTR90	JMP	GETCMD
F950-	0D		2330	MNMSG	.HS	0D
F951-	20	47 45				
F954-	4D	49 4E				
F957-	49	20 52				
F95A-	4F	42 4F				
F95D-	54	20 4D				
F960-	4F	4E 49				
F963-	54	4F 52				
F966-	20	56 31				
F969-	2E	31	2340	.AS	/ GEMINI ROBOT MONITOR V1.1/	
F96B-	0D		2350	.HS	0D	
F96C-	20	20 20				
F96F-	20	43 4F				
F972-	50	59 52				
F975-	49	47 48				
F978-	54	20 31				
F97B-	39	38 34				
F97E-	2C	20 41				
F981-	52	43 54				
F984-	45	43 20				
F987-	53	59 53				
F98A-	54	45 4D				
F98D-	53	20 49				
F990-	4E	43 2E	2360	.AS	/	COPYRIGHT 1984, ARCTEC SYSTEMS INC./
F993-	0D		2370	.HS	0D	
F994-	20	20 20				
F997-	20	41 4C				
F99A-	4C	20 52				
F99D-	49	47 48				
F9A0-	54	53 20				
F9A3-	52	45 53				
F9A6-	45	52 56				
F9A9-	45	44 2E	2380	.AS	/	ALL RIGHTS RESERVED./
F9AC-	0D	0D 00	2390	.HS	0D0D00	
F9AF-	00	20 4D				
F9B2-	49	43 48				
F9B5-	41	45 4C				
F9B8-	20	46 4F				
F9BB-	57	4C 45				

```

F9BE- 52 0D 00 2400 .HS 00204D49434841454C20464F574C45520D00
2410 *-----
2420 * INITIALIZE LCD DISPLAY
2430 *-----
F9C1- A9 80 2440 CLRDSF LDA #$80 SET HORIZONTAL POSITION
F9C3- 8D 5D 05 2450 STA HZPOS TO BEGINNING OF FIRST LINE
F9C6- A9 10 2460 LDA #$10 LCD SYSTEM RESET COMMAND
F9C8- 20 E4 F9 2470 JSR ALLSEG SEND TO ALL SEGMENTS
F9CB- A9 0D 2480 LDA #$0D TURN DISPLAY ON
F9CD- 20 E4 F9 2490 JSR ALLSEG SEND TO ALL SEGMENTS
F9D0- A9 0E 2500 LDA #$0E TURN OFF CURSOR
F9D2- 20 E4 F9 2510 JSR ALLSEG SEND TO ALL SEGMENTS
F9D5- A9 01 2520 LDA #$01 CLEAR AND HOME LCD DISPLAYS
F9D7- 20 E4 F9 2530 JSR ALLSEG SEND TO ALL SEGMENTS
F9DA- A9 09 2540 LDA #$09 CURSOR FONT TO 5X7
F9DC- 20 E4 F9 2550 JSR ALLSEG SEND TO ALL SEGMENTS
F9DF- A9 0F 2560 LDA #$0F TURN ON CURSOR IN FIRST SEGMENT
F9E1- 4C 96 FB 2570 JMP COMDAT
F9E4- 9C 5C 05 2580 ALLSEG STZ VTPOS SET VERT POS TO SEGMENT 0
F9E7- 20 96 FB 2590 ALSEG01 JSR COMDAT SEND COMMAND TO LCD
F9EA- EE 5C 05 2600 INC VTPOS INC LCD ADDRESS
F9ED- EE 5C 05 2610 INC VTPOS FOUR TIMES TO GET
F9F0- EE 5C 05 2620 INC VTPOS TO THE NEXT
F9F3- EE 5C 05 2630 INC VTPOS SEGMENT
F9F6- AC 5C 05 2640 LDY VTPOS
F9F9- C0 11 2650 CPY #$11 ALL SEGMENTS DONE?
F9FB- 30 EA 2660 BMI ALSEG01 NO, REPEAT TILL THEY ARE
F9FD- 9C 5C 05 2670 STZ VTPOS YES, SET UP FOR SEGMENT 0
FA00- 60 2680 RTS AND RETURN
2690 *-----
2700 * CLEAR SCREEN IMAGE
2710 * AT LOCATIONS $400 THRU $540.
2720 *-----
FA01- A0 04 2730 CLRIMG LDY #$04 SET POINTER TO START
FA03- 84 3C 2740 STY DSPIMG+1 OF DISPLAY IMAGE
FA05- 64 3B 2750 STZ DSPIMG
FA07- A0 05 2760 LDY #$05 SAVE END OF DISPLAY
FA09- 84 3E 2770 STY ENDIMG+1 ADDRESS
FA0B- A0 3F 2780 LDY #$3F
FA0D- 84 3D 2790 STY ENDIMG
FA0F- A9 20 2800 LDA #$20 FILL IMAGE WITH SPACES
FA11- 85 32 2810 STA TEMP01
FA13- A5 32 2820 FILIMG LDA TEMP01 FILL IMAGE WITH CHAR
FA15- 92 3B 2830 STA (DSPIMG) STORE IT IN DISP IMAGE
FA17- 2840 >INCD DSPIMG AND INC POINTER
FA17- E6 3B 0000> INC DSPIMG
FA19- D0 02 0000> BNE :1
FA1B- E6 3C 0000> INC DSPIMG+1
0000> :1
FA1D- 2850 >CMPD DSPIMG,ENDIMG
FA1D- A5 3C 0000> LDA DSPIMG+1
FA1F- C5 3E 0000> CMP ENDIMG+1
FA21- D0 04 0000> BNE :1
FA23- A5 3B 0000> LDA DSPIMG
FA25- C5 3D 0000> CMP ENDIMG
0000> :1
FA27- D0 EA 2860 BNE FILIMG IF NOT DONE, LOOP
FA29- A0 04 2870 LDY #$04 POINT TO BEGINNING
FA2B- 84 3C 2880 STY DSPIMG+1 OF IMAGE
FA2D- 64 3B 2890 STZ DSPIMG

```

FA2F-	64	2F	2900	STZ ROWCNT	
FA31-	9C	55 05	2910	STZ LINCNT	
FA34-	60		2920	RTS	AND RETURN
			2930	*-----	
			2940	* GET COMMAND LINE	
			2950	* MAIN ENTRY POINT INTO MONITOR.	
			2960	*-----	
FA35-	20	67 FC	2970	GETCMD JSR CROUT	SEND A CARRIAGE RETURN
FA38-	A9	3E	2980	GETCD1 LDA #\$3E	SET PROMPT TO '>'
FA3A-	8D	5B 05	2990	STA PROMPT	AND SAVE IT
FA3D-	20	43 FA	3000	JSR GETLIN	THEN GET A LINE
FA40-	4C	64 EE	3010	JMP DOCMD	AND DO THE COMMAND
			3020	*-----	
			3030	* GET LINE AND STORE IN LINBUF	
			3040	* HANDLE CONTROL CHARACTERS	
			3050	*-----	
FA43-	20	61 FA	3060	GETLIN JSR CLRBUF	CLEAR THE LINE BUFFER
FA46-	20	98 FA	3070	JSR XPRMPT	SEND A PROMPT
FA49-	20	BC FA	3080	GETKEY JSR RDCHAR	GET A CHAR
FA4C-	20	A0 FA	3090	GETKY1 JSR OUTCHR	DISPLAY IT
FA4F-	C9	20	3100	CMF #\$20	WAS IT A CONTROL CHAR?
FA51-	B0	02	3110	BCS STRLIN	NO, STICK IT IN BUFFER
FA53-	80	19	3120	BRA CONTRL	YES, HANDLE IT
FA55-	A4	2E	3130	STRLIN LDY BUFCNT	GET BUFFER POINTER
FA57-	99	00 02	3140	STA LINBUF,Y	STORE CHAR THERE
FA5A-	E6	2E	3150	INC BUFCNT	MOVE POINTER UP
FA5C-	F0	02	3160	BEQ FULLIN	IF BUFFER FULL RETURN
FA5E-	80	E9	3170	BRA GETKEY	OTHERWISE GET NEXT CHAR
FA60-	60		3180	FULLIN RTS	
			3190	*-----	
			3200	* CLEAR LINE BUFFER	
			3210	*-----	
FA61-	A0	00	3220	CLRBUF LDY #ZERO	FILL LINE BUFFER WITH SPACES
FA63-	A9	20	3230	LDA #\$20	
FA65-	99	00 02	3240	FILBUF STA LINBUF,Y	
FA68-	C8		3250	INY	
FA69-	D0	FA	3260	BNE FILBUF	
FA6B-	64	2E	3270	STZ BUFCNT	
FA6D-	60		3280	RTS	
			3290	*-----	
			3300	* CONTROL CHARACTERS	
			3310	*-----	
FA6E-	C9	08	3320	CONTRL CMP #\$08	BACKSPACE?
FA70-	D0	08	3330	BNE CNTRL1	NO, LOOK AGAIN
FA72-	A5	2E	3340	LDA BUFCNT	YES, GET POINTER
FA74-	F0	D3	3350	BEQ GETKEY	IF BEGINNING OF LINE
FA76-	C6	2E	3360	DEC BUFCNT	ELSE BACKUP
FA78-	80	CF	3370	BRA GETKEY	AND GET NEXT CHAR
FA7A-	C9	0D	3380	CNTRL1 CMP #\$0D	CR?
FA7C-	D0	0E	3390	BNE CNTRL5	NO, LOOK AGAIN
FA7E-	A4	2E	3400	LDY BUFCNT	GET POINTER
FA80-	99	00 02	3410	STA LINBUF,Y	STORE CR IN BUFFER
FA83-	60		3420	RTS	AND RETURN
FA84-	C9	09	3430	CNTRL2 CMP #\$09	RIGHT ARROW?
FA86-	D0	04	3440	BNE CNTRL5	NO LOOK AGAIN
FA88-	E6	2E	3450	INC BUFCNT	
FA8A-	80	BD	3460	BRA GETKEY	
FA8C-	A4	2E	3470	CNTRL5 LDY BUFCNT	NONE OF THESE
FA8E-	99	00 02	3480	STA LINBUF,Y	THEN STORE IT IN BUFFER
FA91-	E6	2E	3490	INC BUFCNT	MOVE POINTER

FA93-	D0 01	3500	BNE CNTRL9	AND RETURN
FA95-	60	3510	RTS	
FA96-	80 B1	3520	CNTRL9 BRA GETKEY	GET NEXT CHAR
		3530	*-----*	
		3540	* PUT PROMPT TO OUTPUT DEVICE	
		3550	*-----*	
FA98-	AD 5B 05	3560	XPRMPT LDA PROMPT	GET FROMPT CHAR AND DISPLAY
FA9B-	20 A0 FA	3570	JSR OUTCHR	
FA9E-	A9 03	3580	LDA #\$03	SEND ETX
		3590	*-----*	
		3600	* CHARACTER OUT TO	
		3610	* CURRENT OUTPUT DEVICE	
		3620	*-----*	
FAA0-	C9 0D	3630	OUTCHR CMP #\$0D	CR?
FAA2-	F0 12	3640	BEQ OTCHR1	
FAA4-	C9 08	3650	CMP #\$08	BACKSPACE?
FAA6-	D0 05	3660	BNE OTCHR0	
FAAB-	CE 55 05	3670	DEC LINCNT	
FAAB-	80 0C	3680	BRA OTCHR2	
FAAD-	C9 0C	3690	OTCHR0 CMP #\$0C	FORM FEED?
FAAF-	F0 05	3700	BEQ OTCHR1	
FAB1-	EE 55 05	3710	INC LINCNT	
FAB4-	80 03	3720	BRA OTCHR2	
FAB6-	9C 55 05	3730	OTCHR1 STZ LINCNT	
FAB9-	6C 1C 00	3740	OTCHR2 JMP (CHROUT)	
		3750	*-----*	
		3760	* CHARACTER IN FROM CURRENT	
		3770	* INPUT DEVICE	
		3780	*-----*	
FABC-	6C 1E 00	3790	RDCHAR JMP (CHARIN)	
		3800	*-----*	
		3810	* POLL CURRENT INPUT DEVICE FOR	
		3820	* CHARACTER	
		3830	*-----*	
FABF-	6C 20 00	3840	INCHAR JMP (CHARPL)	
		3850	*-----*	
		3860	* CHARACTER OUT TO LCD	
		3870	* MAIN ROUTINE TO DISPLAY	
		3880	* CHARACTERS ON LCD.	
		3890	*-----*	
FAC2-	20 A2 E4	3900	LCDOUT JSR SAVALL	SAVE THE REGISTERS
FAC5-	C9 20	3910	LCDOT0 CMP #\$20	IS IT CONTROL CHAR?
FAC7-	30 19	3920	BMI LCDOT3	IF YES GO THERE
FAC9-	92 3B	3930	LCDOT1 STA (DSPIMG)	OTHERWISE DISPLAY IT
FACB-		3940	>INCD DSPIMG AND INC THE POINTER	
FACB-	E6 3B	0000>	INC DSPIMG	
FACD-	D0 02	0000>	BNE :1	
FACF-	E6 3C	0000>	INC DSPIMG+1	
		0000>	:1	
FAD1-	20 A9 FB	3950	JSR CHRDAT	THEN SEND CHAR TO LCD
FAD4-	E6 2F	3960	INC ROWCNT	AND INC THE ROW COUNT
FAD6-	A5 2F	3970	LDA ROWCNT	
FAD8-	C9 28	3980	CMP #\$28	SEE IF END OF LINE
FADA-	30 03	3990	BMI LCDOT2	NO, RETURN
FADC-	20 42 FB	4000	JSR CRTN	YES, DO A CARRIAGE RETURN
FADF-	4C AC E4	4010	LCDOT2 JMP GETALL	RESTORE REGISTERS
FAE2-	C9 08	4020	LCDOT3 CMP #\$08	BACK SPACE?
FAE4-	D0 17	4030	BNE LCDOT4	NO, KEEP LOOKING
FAE6-	A6 2F	4040	LDX ROWCNT	GET ROW COUNT
FAE8-	F0 55	4050	BEQ LCDOT2	IF START OF LINE RETURN

FAEA-	C6	2F	4060	DEC ROWCNT	OTHERWISE BACKUP
FAEC-			4070	>DECD DSPIMG	MOVE IMAGE POINTER BACK
FAEC-	A2	FF	0000>	LDX #\$FF	
FAEE-	C6	3B	0000>	DEC DSPIMG	
FAF0-	E4	3B	0000>	CPX DSPIMG	
FAF2-	D0	02	0000>	BNE :1	
FAF4-	C6	3C	0000>	DEC DSPIMG+1	
			0000> :1		
FAF6-	A9	07	4080	LDA #\$07	AND BACK UP LCD CURSOR
FAF8-	20	96 FB	4090	JSR COMDAT	
FAF8-	80	42	4100	BRA LCDOTZ	THEN RETURN
FAFD-	C9	07	4110 LCDOT4	CMP #\$07	BELL?
FAFF-	D0	05	4120	BNE LCDOT5	
FB01-	20	86 FC	4130	JSR BEEP	
FB04-	80	39	4140	BRA LCDOTZ	
FB06-	C9	0D	4150 LCDOT5	CMP #\$0D	CR?
FB08-	D0	05	4160	BNE LCDOT6	NO, KEEP LOOKING
FB0A-	20	42 FB	4170	JSR CRTN	YES, DO A CR
FB0D-	80	30	4180	BRA LCDOTZ	THEN RETURN
FB0F-	C9	09	4190 LCDOT6	CMP #\$09	RIGHT ARROW?
FB11-	D0	18	4200	BNE LCDOT7	NO, LOOK AGAIN
FB13-			4210	>INCD DSPIMG	YES, INC POINTER
FB13-	E6	3B	0000>	INC DSPIMG	
FB15-	D0	02	0000>	BNE :1	
FB17-	E6	3C	0000>	INC DSPIMG+1	
			0000> :1		
FB19-	A9	06	4220	LDA #\$06	AND LCD
FB1B-	20	96 FB	4230	JSR COMDAT	
FB1E-	E6	2F	4240	INC ROWCNT	AND ROW COUNT
FB20-	A5	2F	4250	LDA ROWCNT	
FB22-	C9	28	4260	CMP #\$28	END OF LINE?
FB24-	30	19	4270	BMI LCDOTZ	NO, RETURN
FB26-	20	42 FB	4280	JSR CRTN	YES, DO CR
FB29-	80	14	4290	BRA LCDOTZ	THEN RETURN
FB2B-	C9	0C	4300 LCDOT7	CMP #\$0C	FORM FEED?
FB2D-	D0	0A	4310	BNE LCDOT8	
FB2F-	20	C1 F9	4320	JSR CLRDISP	
FB32-	20	01 FA	4330	JSR CLRIMG	
FB35-	64	2F	4340	STZ ROWCNT	
FB37-	80	06	4350	BRA LCDOTZ	
FB39-	C9	03	4360 LCDOT8	CMP #\$03	^C OR ETX?
FB3B-	F0	02	4370	BEQ LCDOTZ	YES, IGNORE IT.
FB3D-	80	8A	4380 LCDOT9	BRA LCDOT1	NONE OF THESE, DISPLAY IT
FB3F-	4C	AC E4	4390 LCDOTZ	JMP GETALL	RESTORE REGS
			4400	*-----	
			4410	* CARRIAGE RETURN LCD	
			4420	*-----	
FB42-	20	7F FB	4430 CRTN	JSR FILINE	FILL REST OF LINE WITH SPACES
FB45-	AE	5D 05	4440 CRTN1	LDX HZPOS	GET HORIZONTAL POSITION
FB48-	E0	C0	4450	CPX #\$C0	SEE IF SECOND LINE
FB4A-	B0	08	4460	BCS CRTN5	YES
FB4C-	A2	C0	4470	LDX #\$C0	NO, THEN SETUP FOR SECOND LINE
FB4E-	8E	5D 05	4480	STX HZPOS	AND SAVE IT
FB51-	8A		4490	TXA	
FB52-	80	42	4500	BRA COMDAT	AND TELL LCD
FB54-	AD	5C 05	4510 CRTN5	LDA VTPOS	SWITCH TO NEXT SEGMENT
FB57-	C9	0C	4520	CMP #\$0C	LAST SEGMENT?
FB59-	30	02	4530	BMI CRTN9	NO, CONTINUE
FB5B-	80	5F	4540	BRA SCROLL	YES, SCROLL SCREEN
FB5D-	A9	0E	4550 CRTN9	LDA #\$0E	TURN OFF CURSOR IN LAST SEGMENT

```

FB5F- 20 96 FB 4560      JSR COMDAT
FB62- EE 5C 05 4570      INC VTPOS      GO TO NEXT SEGMENT
FB65- EE 5C 05 4580      INC VTPOS
FB68- EE 5C 05 4590      INC VTPOS
FB6B- EE 5C 05 4600      INC VTPOS
FB6E- A9 80      4610      LDA #$80      SET HZPOS TO START OF FIRST LINE
FB70- 8D 5D 05 4620      STA HZPOS
FB73- 20 96 FB 4630      JSR COMDAT      TELL LCD
FB76- A9 09      4640      LDA #$09      CURSOR FONT 5X7
FB78- 20 96 FB 4650      JSR COMDAT
FB7B- A9 0F      4660      LDA #$0F      TURN ON CURSOR
FB7D- 80 17      4670      BRA COMDAT
FB7F- A5 2F      4680 FILINE LDA ROWCNT      GET ROW COUNT
FB81- C9 28      4690      CMP #$28      END OF LINE?
FB83- B0 0E      4700      BCS FILIN1      YES RETURN
FB85- A9 20      4710      LDA #$20      NO, FILL WITH SPACES
FB87- 92 3B      4720      STA (DSPIMG) IN IMAGE
FB89-      4730      >INCD DSPIMG
FB89- E6 3B      0000>      INC DSPIMG
FB8B- D0 02      0000>      BNE :1
FB8D- E6 3C      0000>      INC DSPIMG+1
      0000> :1
FB8F- E6 2F      4740      INC ROWCNT      AND LINE POINTER
FB91- 80 EC      4750      BRA FILINE      TILL END OF LINE
FB93- 64 2F      4760 FILIN1 STZ ROWCNT
FB95- 60      4770      RTS
      4780 *-----
      4790 * COMMANDS TO LCD
      4800 *-----
FB96- 48      4810 COMDAT PHA      SAVE COMMAND
FB97- AD 00 E0 4820 COMDT1 LDA SCNADD      TEST LCD
FB9A- 29 80      4830      AND #$80      IF READY FOR COMMAND
FB9C- D0 F9      4840      BNE COMDT1      LOOP TIL IT IS
FB9E- AD 5C 05 4850      LDA VTPOS      GET SEGMENT NUMBER
FBA1- 29 FE      4860      AND #$FE      MAKE ADDRESS EVEN
FBA3- A8      4870      TAY
FBA4- 68      4880      PLA      GET COMMAND
FBA5- 99 00 E0 4890      STA SCNADD,Y AND SEND IT
FBA8- 60      4900      RTS
      4910 *-----
      4920 * CHARACTERS TO LCD
      4930 *-----
FBA9- 48      4940 CHRDAT PHA      SAVE CHAR
FBAA- AD 00 E0 4950 CHRDT1 LDA SCNADD      WAIT TIL LCD NOT BUSY
FBAD- 29 80      4960      AND #$80
FBAF- D0 F9      4970      BNE CHRDT1
FBB1- AD 5C 05 4980      LDA VTPOS      GET SEGMENT
FBB4- 09 01      4990      ORA #$01      MAKE ADDRESS ODD
FBB6- A8      5000      TAY
FBB7- 68      5010      PLA      GET CHAR
FBB8- 99 00 E0 5020      STA SCNADD,Y AND SEND IT
FBBB- 60      5030      RTS
      5040 *-----
      5050 * SCROLL ROUTINE
      5060 *-----
FBBC- 20 C1 F9 5070 SCROLL JSR CLRDSP      CLEAR THE DISPLAY
FBBF- A9 18      5080      LDA #$18      SET END OF DISPLAY
FBC1- 85 3D      5090      STA ENDIMG      TO NEXT TO LAST LINE
FBC3- 64 2F      5100      STZ ROWCNT
FBC5- 64 3B      5110      STZ DSPIMG

```

```

FBC7- A9 04      5120      LDA #04      SET POINTER TO START OF IMAGE
FBC9- 85 3C      5130      STA DSPIMG+1
FBCB- A0 28      5140 SCROL1 LDY #28      SET INDEX AHEAD A LINE
FBCD- B1 3B      5150      LDA (DSPIMG),Y
FBCF- 92 3B      5160      STA (DSPIMG) MOVE CHAR UP ONE LINE
FBD1- 20 A9 FB   5170      JSR CHRDAT  AND COPY TO LCD
FBD4- E6 2F      5180      INC ROWCNT
FBD6-             5190      >INCD DSPIMG
FBD6- E6 3B      0000>      INC DSPIMG
FBD8- D0 02      0000>      BNE :1
FBD A- E6 3C      0000>      INC DSPIMG+1
                        0000> :1
FBD C-             5200      >CMPD DSPIMG,ENDING
FBD C- A5 3C      0000>      LDA DSPIMG+1
FBD E- C5 3E      0000>      CMP ENDIMG+1
FBE0- D0 04      0000>      BNE :1
FBE2- A5 3B      0000>      LDA DSPIMG
FBE4- C5 3D      0000>      CMP ENDIMG
                        0000> :1
FBE6- 30 0E      5210      BMI SCROL2  IF NOT DONE GO THERE
FBE8- 64 2F      5220      STZ ROWCNT  POINT TO START OF LINE
FBEA- A0 0C      5230      LDY #0C      SET LCD TO LAST SEGMENT
FBE C- BC 5C 05   5240      STY VTPOS
FBEF- A9 C0      5250      LDA #C0      AND LAST LINE
FBF1- 8D 5D 05   5260      STA HZPOS
FBF4- 80 A0      5270      BRA COMDAT
FBF6- A5 2F      5280 SCROL2 LDA ROWCNT
FBF8- C9 28      5290      CMP #28      END OF LINE?
FBFA- 30 CF      5300      BMI SCROL1  NO, LOOP
FBFC- 20 45 FB   5310      JSR CRTN1  YES, DO CARRIAGE RETURN
FBFF- 64 2F      5320      STZ ROWCNT  AND POINT TO START OF LINE
FC01- 80 C8      5330 SCROL3 BRA SCROL1  AND LOOP
                        5340 *-----
                        5350 * READ KEYBOARD ROUTINES
                        5360 * RDKEY WAITS FOR KEY AND
                        5370 * RETURNS IT.
                        5380 * INKEY READS KEYBOARD AND
                        5390 * IF NO KEY PRESSED RETURNS
                        5400 * ZERO OTHERWISE KEY RETURNED.
                        5410 *-----
FC03- 20 09 FC   5420 RDKEY JSR INKEY
FC06- F0 FB      5430      BEQ RDKEY
FC08- 60          5440      RTS
                        5450 *-----
FC09- DA          5460 INKEY PHX
FC0A- 20 11 E6   5470      JSR LOSPD
FC0D- AD 74 E0   5480      LDA ACRSR      GET ACIA CMD/STAT REG
FC10- AA          5490      TAX
FC11- 29 01      5500      AND #RDRF  RECV REG FULL?
FC13- F0 16      5510      BEQ INKEY1  NO, EXIT
FC15- 8A          5520      TXA
FC16- 29 10      5530      AND #10      FRAMING ERROR?
FC18- D0 0E      5540      BNE INKEY4
FC1A- 8A          5550      TXA
FC1B- 29 28      5560      AND #40      PARITY ERROR?
FC1D- D0 09      5570      BNE INKEY4
FC1F- AD 75 E0   5580      LDA ARDR
FC22- C9 80      5590      CMP #80      VALID CHAR?
FC24- 10 05      5600      BPL INKEY1
FC26- 80 05      5610      BRA INKEY2

```

```

FC28- AD 75 E0 5620 INKEY4 LDA ARDR
FC2B- A9 00 5630 INKEY1 LDA ##00
FC2D- 8D 54 05 5640 INKEY2 STA KEYCHR
FC30- FA 5650 PLX
FC31- AD 54 05 5660 LDA KEYCHR
FC34- 4C 21 E6 5670 JMP RESMSP
5680 *-----
5690 * WAIT IF KEY PRESSED
5700 *-----
FC37- 20 09 FC 5710 WAITKY JSR INKEY
FC3A- AD 54 05 5720 LDA KEYCHR
FC3D- F0 08 5730 BEQ WTKEY1
FC3F- A0 A0 5740 LDY ##A0
FC41- 20 48 FC 5750 JSR PAUSE
FC44- 20 03 FC 5760 JSR RDKEY
FC47- 60 5770 WTKEY1 RTS
5780 *-----
5790 * PAUSE 1 MILLISEC * 'Y'
5800 *-----
FC48- 20 11 E6 5810 PAUSE JSR LOSPD
FC4B- C0 00 5820 CPY ##00
FC4D- F0 15 5830 BEQ PAUS5
FC4F- EA 5840 NOP
FC50- C0 01 5850 CPY ##01
FC52- D0 02 5860 BNE PAUS0
FC54- 80 09 5870 BRA PAUS3
FC56- 88 5880 PAUS0 DEY
FC57- A2 C6 5890 PAUS1 LDX ##C6
FC59- CA 5900 PAUS2 DEX
FC5A- D0 FD 5910 BNE PAUS2
FC5C- 88 5920 DEY
FC5D- D0 F8 5930 BNE PAUS1
FC5F- A2 C3 5940 PAUS3 LDX ##C3
FC61- CA 5950 PAUS4 DEX
FC62- D0 FD 5960 BNE PAUS4
FC64- 4C 21 E6 5970 PAUS5 JMP RESMSP
5980 *-----
5990 * CARRIAGE RETURN OUT
6000 * GENERATES CARRIAGE RETURN ON
6010 * LCD.
6020 *-----
FC67- A9 0D 6030 CROUT LDA ##0D
FC69- 4C A0 FA 6040 JMP OUTCHR
6050 *-----
6060 * SPACES OUT
6070 * SPOUT MOVES THREE SPACES
6080 * RIGHT ON LCD.
6090 * SPCOUT MOVES NUMBER OF SPACES
6100 * IN 'Y' REG TO RIGHT ON LCD
6110 * UP TO A MAX OF 255 SPACES.
6120 *-----
FC6C- A0 03 6130 SPOUT LDY ##03
FC6E- A9 20 6140 SPCOUT LDA ##20
FC70- 20 A0 FA 6150 JSR OUTCHR
FC73- 88 6160 DEY
FC74- D0 F8 6170 BNE SPCOUT
FC76- 60 6180 RTS
6190 *-----
6200 * TAB TO COLUMN 20 OF DISPLAY.
6210 *-----

```


FC77-	AD 55 05	6220	TABOUT	LDA	LINCNT
FC7A-	C9 14	6230		CMF	##14
FC7C-	B0 07	6240		BCS	TABOUT1
FC7E-	A0 01	6250		LDY	##01
FC80-	20 6E FC	6260		JSR	SPCOUT
FC83-	B0 F2	6270		BRA	TABOUT
FC85-	60	6280	TABOUT1	RTS	
		6290	*-----		
		6300	* BELL ROUTINE		
		6310	*-----		
FC86-	20 11 E6	6320	BEEP	JSR	LOSPD
FC89-	A0 3F	6330		LDY	##3F
FC8B-	A2 FF	6340	BEEP1	LDX	##FF
FC8D-	2C 78 E0	6350	BEEP2	BIT	BELLOC
FC90-	CA	6360		DEX	
FC91-	D0 FA	6370		BNE	BEEP2
FC93-	88	6380		DEY	
FC94-	D0 F5	6390		BNE	BEEP1
FC96-	4C 21 E6	6400		JMP	RESMSP
		1240			. IN ROMON.V1.1.PARTB
		1000	*-----		
		1010	* POLL SERIAL PORTS		
		1020	* PORT TO BE SCANNED IN 'A'		
		1030	* IF NO CHAR READY THEN		
		1040	* ZERO RETURNED IN 'A' ELSE		
		1050	* CHAR IN 'A'.		
		1060	*-----		
FC99-	08	1070	POLPRT	PHP	
FC9A-	78	1080		SEI	
FC9B-	8D 66 05	1090		STA	SPRTNM
FC9E-	A9 02	1100		LDA	##02
FCA0-	85 32	1110		STA	TEMP01
FCA2-	64 31	1120		STZ	TEMP00
FCA4-	AD 66 05	1130		LDA	SPRTNM
FCA7-	20 13 FD	1140		JSR	OPENP
FCAA-	20 11 E6	1150		JSR	LOSPD
FCAD-	20 12 FE	1160		JSR	BITWAT
FCB0-	A2 01	1170		LDX	##01
FCB2-	AD 66 05	1180		LDA	SPRTNM
FCB5-	29 0E	1190		AND	##0E
FCB7-	F0 02	1200		BEQ	POLPT1
FCB9-	A2 02	1210		LDX	##02
FCBB-	20 12 FE	1220	POLPT1	JSR	BITWAT
FCBE-	AD 61 05	1230		LDA	RTSLO
FCC1-	9D 20 E0	1240		STA	SERADD,X
FCC4-	AD 20 E0	1250	POLPT2	LDA	SERADD
FCC7-	2D 5F 05	1260		AND	RDMSK
FCCA-	F0 12	1270		BEQ	POLPT3
FCCC-	E6 31	1280		INC	TEMP00
FCCE-	D0 F4	1290		BNE	POLPT2
FCD0-	C6 32	1300		DEC	TEMP01
FCD2-	D0 F0	1310		BNE	POLPT2
FCD4-	20 E9 FC	1320		JSR	SEROFF
FCD7-	20 21 E6	1330		JSR	RESMSP
FCD A-	28	1340		PLP	
FCD B-	A9 00	1350		LDA	##00
FCD D-	60	1360		RTS	
FCDE-	20 93 FD	1370	POLPT3	JSR	RECVCH
FCE1-	20 E9 FC	1380		JSR	SEROFF
FCE4-	20 21 E6	1390		JSR	RESMSP

```

FCE7- 28      1400      PLP
FCE8- 60      1410      RTS
                1420 *-----
                1430 * TURN ALL SERIAL PORTS OFF
                1440 *-----
FCE9- 48      1450 SEROFF PHA
FCEA- A9 FF    1460      LDA #$FF
FCEC- 8D 21 E0 1470      STA SERADD+1
FCEF- A9 0E    1480      LDA #$0E
FCF1- 0D 22 E0 1490      ORA SERADD+2
FCF4- 8D 22 E0 1500      STA SERADD+2
FCF7- 68      1510      PLA
FCF8- 60      1520      RTS
                1530 *-----
                1540 * SET BAUD RATE FOR EXTERNAL
                1550 * SERIAL PORT ACCORDING TO IT'S
                1560 * SWITCHES.  BAUD RATES ARE:
                1570 * 0- 110 BAUD = #$D7
                1580 * 1- 150 BAUD = #$9D
                1590 * 2- 300 BAUD = #$4D
                1600 * 3- 600 BAUD = #$25
                1610 * 4- 1200 BAUD = #$11
                1620 * 5- 2400 BAUD = #$07
                1630 * 6- 4800 BAUD = #$02
                1640 * 7- 7200 BAUD = #$01
                1650 *-----
FCF9- AD 51 E0 1660 STBAUD LDA BITINP+1
FCFC- 29 E0    1670      AND #BRMASK
FCFE- 4A      1680      LSR
FCFF- 4A      1690      LSR
FD00- 4A      1700      LSR
FD01- 4A      1710      LSR
FD02- 4A      1720      LSR
FD03- A8      1730      TAY
FD04- B9 0B FD 1740      LDA BAUDRT,Y
FD07- 8D 65 05 1750      STA BDRATE
FD0A- 60      1760      RTS
FD0B- D7 9D 4D
FD0E- 25 11 07
FD11- 02 01    1770 BAUDRT .HS D79D4D2511070201
                1780 *-----
                1790 * OPEN SERIAL PORT IN 'A'
                1800 * AND SET BAUD RATE FOR EITHER 300 OR 1200
                1810 *-----
FD13- 8D 66 05 1820 OPENP STA SPRTNM  SAVE SERIAL PORT NUMBER
FD16- 0A      1830      ASL
FD17- 0A      1840      ASL          MULTIPLY BY FOUR
FD18- AA      1850      TAX          AND USE AS AN INDEX
FD19- A0 00    1860      LDY #$00
FD1B- BD 4C FD 1870 OPENP1 LDA MSKDTA,X GET FIRST MASK
FD1E- 99 5F 05 1880      STA RDMSK,Y AND PUT IT IN MEM
FD21- E8      1890      INX          NEXT MASK
FD22- C8      1900      INY          NEXT MEM
FD23- C0 04    1910      CPY #$04    DONE?
FD25- D0 F4    1920      BNE OPENP1  NO, LOOP
FD27- AD 66 05 1930      LDA SPRTNM
FD2A- C9 02    1940      CMP #$02    IS RF MODEM?
FD2C- D0 05    1950      BNE OPENP4  NO
FD2E- AD 0D FD 1960      LDA BAUDRT+2 YES SET BAUD FOR 300
FD31- 80 15    1970      BRA OPENP3

```

```

FD33- C9 03      1980 OPENP4 CMP  #03      IS EXTERNAL SERIAL PORT?
FD35- D0 02      1990          BNE OPENP5      NO
FD37- 80 C0      2000          BRA STBAUD      YES THEN READ BAUD RATE SWITCHES
FD39- AD 51 E0    2010 OPENP5 LDA  BITINP+1  GET BAUD RATE SWITCH
FD3C- 29 10      2020          AND  #BRMSK1    SEPARATE GARBAGE
FD3E- F0 05      2030          BEQ OPENP2
FD40- AD 0D FD    2040          LDA BAUDRT+2    SET FOR 300 BAUD
FD43- 80 03      2050          BRA OPENP3
FD45- AD 0F FD    2060 OPENP2 LDA  BAUDRT+4    SET FOR 1200 BAUD
FD48- 8D 65 05    2070 OPENP3 STA  BDRATE      SAVE BAUD RATE
FD4B- 60          2080          RTS

FD4C- 01 40 BF
FD4F- FE 02 80
FD52- 7F FD      2090 MSKDTA .HS 0140BFFE02807FFD
FD54- 04 80 FF
FD57- FB 08 40
FD5A- FD F7      2100          .HS 0480FFFFB0840FDF7
FD5C- 10 20 FB
FD5F- EF 20 10
FD62- F7 DF      2110          .HS 1020FB EF2010F7DF
2120 *-----
2130 * GET SERIAL CHAR FROM OPEN PORT
2140 *-----
FD64- 08          2150 INCH   PHP
FD65- 78          2160          SEI
FD66- 20 11 E6    2170          JSR LOSPD      CLOCK TO LO SPEED
FD69- A2 01      2180          LDX  #01      SET UP FOR PORT 1 OR 0
FD6B- AD 66 05    2190          LDA SPRTNM     GET SERIAL PORT NUMBER
FD6E- 29 0E      2200          AND  #0E      PORT 1 OR 0?
FD70- F0 02      2210          BEQ INCH0
FD72- A2 02      2220          LDX  #02
FD74- 20 12 FE    2230 INCH0  JSR BITWAT      WAIT ONE BIT TIME
FD77- AD 61 05    2240          LDA RTSLO      SET RTS LOW
FD7A- 9D 20 E0    2250          STA SERADD,X
FD7D- AD 20 E0    2260 INCH1  LDA SERADD      GET RD
FD80- 2D 5F 05    2270          AND  RDMSK     IS IT START BIT
FD83- D0 F8      2280          BNE INCH1     NO, WAIT TIL IT IS
FD85- 20 93 FD    2290          JSR RECVCH     GO GET CHAR
FD88- 20 E9 FC    2300          JSR SEROFF     TURN OFF PORT
FD8B- 20 21 E6    2310          JSR RESMSP     RESUME OLD CLOCK SPEED
FD8E- 28          2320          PLP
FD8F- AD 64 05    2330          LDA CHARIO
FD92- 60          2340          RTS
FD93- 20 15 FE    2350 RECVCH JSR BITHAF      WAIT TIL MIDDLE OF BIT
FD96- A0 08      2360          LDY  #08      SET UP FOR 8 BITS
FD98- A9 00      2370          LDA  #00
FD9A- 8D 64 05    2380          STA CHARIO     CLEAR CHARIO
FD9D- 20 12 FE    2390 NXTRBT JSR BITWAT      WAIT FOR CENTER OF NEXT BIT
FDA0- 18          2400          CLC          SET BIT MARKER LOW
FDA1- AD 20 E0    2410          LDA SERADD     GET BIT
FDA4- 2D 5F 05    2420          AND  RDMSK     SEPARATE IT FROM GARBAGE
FDA7- F0 01      2430          BEQ LOWBIT     IF LOW PUT IT IN CHARIO
FDA9- 38          2440          SEC          OTHERWISE SET BIT MARKER
FDAA- 6E 64 05    2450 LOWBIT ROR CHARIO     AND SHIFT IT INTO CHARIO
FDAD- 88          2460          DEY          DEC BIT COUNT
FDAE- D0 ED      2470          BNE NXTRBT     AND GET NEXT BIT
FDB0- AD 64 05    2480          LDA CHARIO     GET CHARACTER
FDB3- 60          2490          RTS
2500 *-----
2510 * SEND SERIAL CHAR TO OPEN PORT

```

```

2520 *-----
FDB4- 08      2530 OUTCH  PHP
FDB5- 78      2540      SEI
FDB6- 48      2550      PHA          SAVE CHARACTER
FDB7- 20 11 E6 2560      JSR LOSPD    CLOCK TO LO SPEED
FDBA- A2 00      2570      LDX #00     SETUP FOR PORT 1 OR 0
FDBC- AD 66 05 2580      LDA SPRTNM  GET PORT #
FDBF- 29 0E      2590      AND #0E    IS IT 0 OR 1?
FDC1- F0 02      2600      BEQ OUTCH1
FDC3- A2 02      2610      LDX #02
FDC5- BD 20 E0 2620 OUTCH1 LDA SERADD,X GET CTS
FDC8- 2D 60 05 2630      AND CTSMK   SEPARATE IT FROM GARBAGE
FDCB- D0 F8      2640      BNE OUTCH1  LOOP TIL ITS LOW
FDCD- 68      2650      PLA          RETRIEVE CHAR
FDCE- 8D 64 05 2660      STA CHARIO  AND SAVE IT
FDD1- AD 62 05 2670      LDA TDLO    SEND START BIT
FDD4- 8D 21 E0 2680      STA SERADD+1
FDD7- 20 12 FE 2690      JSR BITWAT
FDDA- A0 08      2700      LDY #08    SET UP FOR 8 BITS
FDDC- A9 01      2710      LDA #01    SET BIT MARKER HIGH
FDDE- 8D 63 05 2720      STA MASK
FDE1- AD 64 05 2730 NXTSBT LDA CHARIO  GET A COPY OF CHAR
FDE4- 2D 63 05 2740      AND MASK    GET A BIT
FDE7- D0 05      2750      BNE HIBIT   IF HI SENDIT
FDE9- AD 62 05 2760      LDA TDLO    IF LOW SEND IT
FDEC- D0 02      2770      BNE OUTDTA
FDEE- A9 FF      2780 HIBIT  LDA #FF
FDF0- 8D 21 E0 2790 OUTDTA STA SERADD+1
FDF3- 20 12 FE 2800      JSR BITWAT
FDF6- 0E 63 05 2810      ASL MASK    THEN GO FOR NEXT BIT
FDF9- 88      2820      DEY          DEC BIT COUNT
FDFA- D0 E5      2830      BNE NXTSBT   IF NOT DONE LOOP
FDFC- A9 FF      2840      LDA #FF    TD BIT HIGH FOR STOP BITS
FDFF- 8D 21 E0 2850      STA SERADD+1
FE01- 20 12 FE 2860      JSR BITWAT  STOP BIT
FE04- 20 12 FE 2870      JSR BITWAT  STOP BIT
FE07- 20 E9 FC 2880      JSR SEROFF
FE0A- 20 21 E6 2890      JSR RESMSP  RESUME OLD CLOCK SPEED
FE0D- 28      2900      PLP
FE0E- AD 64 05 2910      LDA CHARIO  RETRIEVE CHAR
FE11- 60      2920      RTS
2930 *-----
FE12- 20 15 FE 2940 BITWAT JSR BITHAF
FE15- 08      2950 BITHAF PHP
FE16- 48      2960      PHA
FE17- DA      2970      PHX
FE18- 5A      2980      PHY
FE19- AE 65 05 2990      LDX BDRATE
FE1C- A0 03      3000 DLYX  LDY #03
FE1E- 88      3010 DLYY  DEY
FE1F- D0 FD      3020      BNE DLYY
FE21- CA      3030      DEX
FE22- D0 F8      3040      BNE DLYX
FE24- 7A      3050      PLY
FE25- FA      3060      PLX
FE26- 68      3070      PLA
FE27- 28      3080      PLP
FE28- 60      3090      RTS
3100 *-----
3110 * SEND PARALLEL CHAR TO PRINTER

```

```

3120 *-----
FE29- C9 03 3130 PRTOUT CMP #03
FE2B- D0 01 3140 BNE PRTOT0
FE2D- 60 3150 RTS
FE2E- 48 3160 PRTOT0 PHA SAVE CHAR
FE2F- 20 11 E6 3170 JSR LOSPD
FE32- 20 50 F1 3180 JSR INTAPL SET UP PIA FOR OUTPUT
FE35- 68 3190 PLA GET CHAR
FE36- 8D 32 E0 3200 STA PARADD+2 SENDIT
FE39- 2C 33 E0 3210 PRTOT1 BIT PARADD+3 SEE IF ACK LOW
FE3C- 10 FB 3220 BPL PRTOT1 WAIT FOR IT
FE3E- AD 32 E0 3230 LDA PARADD+2 CLEAR STROBE
FE41- 20 54 F1 3240 JSR PIAINP
FE44- 4C 21 E6 3250 JMP RESMSP
3260 *-----
3270 * GET FUNCTION KEYS
3280 *-----
FE47- AD 52 E0 3290 GETFNC LDA BITINP+2
FE4A- 09 07 3300 ORA #07
FE4C- 49 FF 3310 EOR #FF
FE4E- D0 01 3320 BNE GETFK2
FE50- 60 3330 RTS
FE51- A0 14 3340 GETFK2 LDY #14
FE53- 20 48 FC 3350 JSR PAUSE
FE56- A0 01 3360 LDY #01
FE58- 0A 3370 ASL
FE59- B0 0D 3380 BCS GETFK1
FE5B- C8 3390 INY
FE5C- 0A 3400 ASL
FE5D- B0 09 3410 BCS GETFK1
FE5F- C8 3420 INY
FE60- 0A 3430 ASL
FE61- B0 05 3440 BCS GETFK1
FE63- C8 3450 INY
FE64- 0A 3460 ASL
FE65- B0 01 3470 BCS GETFK1
FE67- C8 3480 INY
FE68- 98 3490 GETFK1 TYA
FE69- 4C 86 FC 3500 JMP BEEP
3510 *-----
3520 * REAL TIME CLOCK ROUTINES
3530 * IRQ INDICATORS
3540 * STATUS BIT 7= CLOCK OUT IRQ (HI)
3550 * STATUS BIT 6= ALARM IRQ (HI)
3560 * COMMAND REGISTER
3570 * BIT0&1= OSC TO 2.09MHZ 0=LOW 1=HI
3580 * BIT2= COUNTERS 0=DISABLE 1=ENABLE
3590 * BIT3= ALARM LATCHES 0=DISABLE
3600 * BIT4-7= COUNTER OUT (1101=1/MIN)
3610 *-----
FE6C- 20 11 E6 3620 SETCLK JSR LOSPD CLOCK TO LOW SPEED
FE6F- 8D 11 E0 3630 STA RTCFRE FREEZE COUNTERS IN CLOCK
FE72- A9 02 3640 LDA #02 DISABLE CLOCK
FE74- 8D 17 E0 3650 STA RTCCSR
FE77- A2 04 3660 LDX #04 START AT MONTHS
FE79- B5 25 3670 SETCL1 LDA TIMBUF,X GET TIME FROM MEM
FE7B- 9D 12 E0 3680 STA RTCBAS,X AND PUT IN CLOCK
FE7E- CA 3690 DEX NEXT REG
FE7F- 10 FB 3700 BPL SETCL1 LOOP TIL DONE
FE81- A5 2D 3710 LDA CLKMOD GET MODE

```

```

FE83- 8D 17 E0 3720          STA RTCCSR      AND TURN ON CLOCK
FE86- 4C 21 E6 3730          JMP RESMSP
FE89- 20 11 E6 3740 GETCLK JSR LOSPD
FE8C- 8D 11 E0 3750          STA RTCFRE      FREEZE THE COUNTERS
FE8F- A9 02      3760          LDA #02        TURN OFF ALARM
FE91- 8D 17 E0 3770          STA RTCCSR
FE94- A2 04      3780          LDX #04        START WITH MONTH REG
FE96- BD 12 E0 3790 GETCL1 LDA RTCBAS,X GET REG
FE99- 95 25      3800          STA TIMBUF,X STORE IN MEM
FE9B- CA      3810          DEX
FE9C- 10 F8      3820          BPL GETCL1
FE9E- A5 2D      3830          LDA CLKMOD      GET MODE
FEA0- 8D 17 E0 3840          STA RTCCSR      TURN ON CLOCK
FEA3- 4C 21 E6 3850          JMP RESMSP
FEA6- 20 11 E6 3860 SETALM JSR LOSPD
FEA9- 8D 11 E0 3870          STA RTCFRE      FREEZE COUNTERS
FEAC- A9 0A      3880          LDA #0A        STOP CLOCK
FEAE- 8D 17 E0 3890          STA RTCCSR
FEB1- A2 02      3900          LDX #02        START WITH HOURS
FEB3- B5 25      3910 SETAL1 LDA TIMBUF,X GET TIME FROM MEM
FEB5- 9D 12 E0 3920          STA RTCBAS,X STORE IN ALARM REGS
FEB8- CA      3930          DEX
FEB9- 10 F8      3940          BPL SETAL1
FEBB- A9 0E      3950          LDA #0E        START CLOCK AND ENABLE ALARM
FEBD- 85 2D      3960          STA CLKMOD
FEBF- 8D 17 E0 3970          STA RTCCSR
FEC2- 4C 21 E6 3980          JMP RESMSP
FEC5- 20 11 E6 3990 ALMOFF JSR LOSPD
FEC8- A9 06      4000          LDA #06        TURN OFF ALARM
FECA- 85 2D      4010          STA CLKMOD
FECC- 8D 17 E0 4020          STA RTCCSR
FECF- 4C 21 E6 4030          JMP RESMSP
4040 *-----
4050 * READ ANALOG TO DIGITAL
4060 * CONVERTER
4070 * 'Y' REG HAS PORT # TO READ
4080 * VALUE IS RETURNED IN 'A' REG
4090 * 'X' IS DESTROYED
4100 *-----
FED2- 20 11 E6 4110 ADCONV JSR LOSPD      SET CLOCK TO LOW SPEED
FED5- 64 32      4120          STZ TEMP01      SET ANSWER LOC TO ZERO
FED7- 64 33      4130          STZ TEMP02
FED9- A9 08      4140          LDA #08        SET UP FOR EIGHT REPS
FEDB- 85 31      4150          STA TEMP00
FEDD- A9 00      4160          LDA #0
FEDF- 99 40 E0 4170 RDPORT STA ADCADD,Y      PULSE ALE, SELECT CHANNEL
FEE2- B9 40 E0 4180          LDA ADCADD,Y      PULSE SC/DE, START CONVERSION
FEE5- A2 0E      4190          LDX #0E        WAIT FOR CONVERSION
FEE7- CA      4200 .1          DEX
FEE8- D0 FD      4210          BNE .1
FEEA- B9 40 E0 4220          LDA ADCADD,Y      GET RESULTS
FEED- 18      4230          CLC
FEEE- 65 32      4240          ADC TEMP01      ADD EM UP
FEF0- 85 32      4250          STA TEMP01
FEF2- 90 06      4260          BCC DONE8
FEF4- A9 00      4270          LDA #0
FEF6- 65 33      4280          ADC TEMP02      IF CARRY INC TEMP2
FEF8- 85 33      4290          STA TEMP02
FEFA- C6 31      4300 DONE8 DEC TEMP00      READ PORT 8 TIMES?
FEFC- D0 E1      4310          BNE RDPORT      NO, LOOP

```

FEFE-	46	33	4320	LSR TEMP02	DIVIDE BY 8 FOR AVERAGE
FF00-	66	32	4330	ROR TEMP01	ROTATE TEMP1
FF02-	46	33	4340	LSR TEMP02	CARRY INTO TEMP2
FF04-	66	32	4350	ROR TEMP01	
FF06-	46	33	4360	LSR TEMP02	
FF08-	66	32	4370	ROR TEMP01	
FF0A-	20	21	E6 4380	JSR RESMSP	RESUME OLD CLOCK SPEED
FF0D-	A5	32	4390	LDA TEMP01	PUT ANSWER IN 'A'
FF0F-	60		4400	RTS	
			4410	*-----	
			4420	* DETERMINE IRQ OR BREAK	
			4430	*-----	
FF10-	85	4B	4440	BRKIRQ STA REGA	SAVE REGISTER 'A'
FF12-	68		4450	PLA	
FF13-	48		4460	PHA	TRANSFER 'P' TO 'A'
FF14-	29	10	4470	AND #\$10	
FF16-	D0	15	4480	BNE BREAK	WAS BREAK BIT SET?
			4490	*-----	
			4500	* IRQ HANDLER	
			4510	*-----	
FF18-	A5	4B	4520	IRQHND LDA REGA	
FF1A-	5A		4530	PHY	
FF1B-	DA		4540	PHX	
FF1C-	48		4550	PHA	
FF1D-	2C	17	E0 4560	BIT RTCCSR	WAS IT CLOCK?
FF20-	70	05	4570	BVS IRQHD1	YES
FF22-	30	06	4580	BMI IRQHD2	WAS IT ALARM?
FF24-	6C	14	00 4590	JMP (IRQLOC)	
FF27-	6C	1A	00 4600	IRQHD1 JMP (CLKIRQ)	
FF2A-	6C	18	00 4610	IRQHD2 JMP (ALMIRO)	
			4620	*-----	
			4630	* BREAK HANDLER	
			4640	*-----	
FF2D-	68		4650	BREAK PLA	YES, RESTORE 'P'
FF2E-	20	B9	FF 4660	JSR SAVREG	AND SAVE ALL REGISTERS
FF31-	68		4670	PLA	
FF32-	85	47	4680	STA PCL	STORE PROGRAM COUNTER L
FF34-	68		4690	PLA	
FF35-	85	49	4700	STA PCH	STORE PROGRAM COUNTER HI
FF37-	64	37	4710	STZ INDEX1	INDEX TO ZERO
FF39-	20	67	FC 4720	JSR CROUT	SEND CR TO DISPLAY
FF3C-	A4	37	4730	REGLIN LDY INDEX1	GET INDEX
FF3E-	B9	4A	FF 4740	LDA REGTAB,Y	GET CHAR FROM MESSAGE ARRAY
FF41-	F0	26	4750	BEQ RGLN10	NULL MEANS ALL DONE
FF43-	20	A0	FA 4760	JSR OUTCHR	SEND CHAR TO DISPLAY
FF46-	E6	37	4770	INC INDEX1	NEXT CHAR
FF48-	80	F2	4780	BRA REGLIN	
FF4A-	20	20	50		
FF4D-	43	20	20		
FF50-	20	20	20		
FF53-	41	20	20		
FF56-	20	20	58		
FF59-	20	20	20		
FF5C-	20	59	20		
FF5F-	20	20	20		
FF62-	50	20	20		
FF65-	20	20	53	4790 REGTAB .AS / PC	A X Y P S/
FF68-	00		4800	.DA #\$00	
FF69-	20	67	FC 4810	RGLN10 JSR CROUT	SEND CR TO DISPLAY
FF6C-	A5	47	4820	LDA PCL	DISPLAY PROGRAM COUNTER

FF6E-	85 3F	4830	STA ADDR01	
FF70-	A5 49	4840	LDA PCH	
FF72-	85 40	4850	STA ADDR01+1	
FF74-		4860	>DECD ADDR01	
FF74-	A2 FF	0000>	LDX #\$FF	
FF76-	C6 3F	0000>	DEC ADDR01	
FF78-	E4 3F	0000>	CPX ADDR01	
FF7A-	D0 02	0000>	BNE :1	
FF7C-	C6 40	0000>	DEC ADDR01+1	
		0000> :1		
FF7E-		4870	>DECD ADDR01	
FF7E-	A2 FF	0000>	LDX #\$FF	
FF80-	C6 3F	0000>	DEC ADDR01	
FF82-	E4 3F	0000>	CPX ADDR01	
FF84-	D0 02	0000>	BNE :1	
FF86-	C6 40	0000>	DEC ADDR01+1	
		0000> :1		
FF88-	20 FC F3	4880	JSR DSPADD	
FF8B-	20 6C FC	4890	JSR SPOUT	
FF8E-	A5 4B	4900	LDA REGA	DISPLAY REG 'A'
FF90-	20 B9 F3	4910	JSR HEXOUT	
FF93-	20 6C FC	4920	JSR SPOUT	
FF96-	A5 4C	4930	LDA REGX	DISPLAY REG 'X'
FF98-	20 B9 F3	4940	JSR HEXOUT	
FF9B-	20 6C FC	4950	JSR SPOUT	
FF9E-	A5 4D	4960	LDA REGY	DISPLAY REG 'Y'
FFA0-	20 B9 F3	4970	JSR HEXOUT	
FFA3-	20 6C FC	4980	JSR SPOUT	
FFA6-	A5 4E	4990	LDA REGP	DISPLAY CONDITION CODE REG
FFA8-	20 B9 F3	5000	JSR HEXOUT	
FFAB-	20 6C FC	5010	JSR SPOUT	
FFAE-	A5 4F	5020	LDA REGS	DISPLAY STACK POINTER
FFB0-	20 B9 F3	5030	JSR HEXOUT	
FFB3-	A2 FF	5040	LDX #\$FF	
FFB5-	9A	5050	TXS	
FFB6-	4C 35 FA	5060	JMP GETCMD	GO TO THE MONITOR
		5070	*-----	
		5080	* SAVE ALL REGISTERS	
		5090	*-----	
FFB9-	85 4E	5100	SAVREG STA REGP	
FFBB-	86 4C	5110	STX REGX	
FFBD-	84 4D	5120	STY REGY	
FFBF-	BA	5130	TSX	
FFC0-	86 4F	5140	STX REGS	
FFC2-	A6 4C	5150	LDX REGX	
FFC4-	A5 4B	5160	LDA REGA	
FFC6-	60	5170	RTS	
		5180	*-----	
		5190	* RETURN FROM INTERRUPT	
		5200	*-----	
FFC7-	68	5210	RETINT PLA	
FFC8-	FA	5220	PLX	
FFC9-	7A	5230	PLY	
FFCA-	40	5240	RTI	
		5250	*-----	
		5260	* NMI HANDLER	
		5270	*-----	
FFCB-	5A	5280	NMIHND PHY	
FFCC-	DA	5290	PHX	
FFCD-	4B	5300	PHA	

FFCE-	6C 16 00	5310	JMP (NMILOC)
FFD1-	00 00 00		
FFD4-	00	5320	ZERONM .HS 00000000
FFD5-	EA	5330	ZZEND NOP
		5340	*-----
		5350	* 65C02 VECTORS
		5360	*-----
		5370	.OR \$FFFA
		5380	.TA \$3FFA
FFFA-	CB FF	5390	NMI .DA NMIHND
FFFC-	3B FB	5400	RESET .DA RESENT
FFFE-	10 FF	5410	IRQ .DA BRKIRQ

SYMBOL TABLE

E738- ABS
 E735- ABS.1
 0064- ACH
 0013- ACIARS
 0011- ACIASU
 0062- ACL
 0063- ACM
 E074- ACRSR
 E91E- ACS
 E91B- ACS.1
 0061- ACX
 E040- ADCADD
 FED2- ADCONV
 E6D2- ADD
 E6CF- ADD.1
 E6CC- ADD.2
 003F- ADDR01
 0041- ADDR02
 0043- ADDR03
 0045- ADDR0B
 E6D5- ADL
 0559- ADMODE
 EB3A- ADRINC
 0039- ADRPTR
 F9E4- ALLSEG
 0018- ALMIRQ
 FEC5- ALMOFF
 F9E7- ALSG01
 E8FE- AOV
 E9D0- AOL
 E075- ARDR
 E8F8- ASC
 F3E0- ASCHEX
 F7AD- ASCII
 E90E- ASF
 E902- ASG
 E906- ASL1
 E8EA- ASN
 E8E7- ASN.1
 E8EE- ASN1
 E91A- ASD
 E454- ASTOHX
 E977- ATF
 E971- ATM

E938- ATN
E935- ATN.1
E97E- ATQ
E944- ATP
Q56D- AVGLD
E49Q- AVGRNG
Q551- BANKAD
E549- BANKSW
FDQB- BAUDRT
F384- BCDBIN
Q0Q8- BCKSON
E45A- BCTOBN
Q565- BDRATE
Q0Q2- BECBIT
Q0Q4- BECREF
FC86- BEEP
FC8B- BEEP1
FC8D- BEEP2
E078- BELLOC
Q0Q1- BITQ
Q0Q0- BITQ0
Q0Q1- BITQ1
Q0Q2- BITQ2
Q0Q3- BITQ3
Q0Q4- BITQ4
Q0Q5- BITQ5
Q0Q6- BITQ6
Q0Q7- BITQ7
Q0Q2- BIT1
Q0Q4- BIT2
Q0Q8- BIT3
Q01Q- BIT4
Q02Q- BIT5
Q04Q- BIT6
Q08Q- BIT7
FE15- BITHAF
E05Q- BITINP
E06Q- BITOUT
FE12- BITWAT
E571- BNKSW1
E573- BNKSW2
E57D- BNKSW3
FF2D- BREAK
FF1Q- BRKIRQ
Q0EQ- BRMASK
Q01Q- BRMSK1
Q02E- BUFCNT
Q54Q- BUFPTR
Q55A- BYTCNT
C0Q0- CATLOG
Q547- CATN
Q0Q7- CDSFLG
E727- CEO
E843- CGR
E41E- CHAIN
E41B- CHAOUT
E421- CHAPOL
Q01E- CHARIN
Q564- CHARIO
Q02Q- CHARPL

E73C- CHG
E72F- CHG.1
E73F- CHL
E74A- CHO
FBA9- CHRDAT
FBAA- CHRDT1
001C- CHROUT
E83E- CKA
E851- CKB
E846- CKL
E834- CKS
E415- CLBUFR
E40F- CLDISP
F31F- CLKFMT
001A- CLKIRQ
002D- CLKMOD
E71E- CLP
E720- CLQ
E68E- CLR
FA61- CLRBUF
F9C1- CLRDSP
FA01- CLRIMG
E72C- CMI
E709- CMP.1
E706- CMP.2
E70C- CMP.AC
E718- CMX
E72A- CNE
0060- CNT
FA7A- CNTRL1
FAB4- CNTRL2
FABC- CNTRL5
FA96- CNTRL9
0024- COLDAD
FB96- COMDAT
FB97- COMDT1
EE8D- COMTAB
FA6E- CONTRL
E481- CONVRT
E8BF- COS
E8BC- COS.1
E921- CPA
E703- CPL
0051- CPUFLG
FC67- CROUT
FB42- CRTN
FB45- CRTN1
FB54- CRTN5
FB5D- CRTN9
E493- CRTOUT
0560- CTSMSK
0030- CURADD
0553- CURBNK
056A- CURHD
EA24- DASMB1
EA12- DASMBL
EA25- DASMON
EA0A- DASMSX
DFFA- DATIN
E6AF- DCO

E6A1- DEC.1
E6A4- DEC.AC
0073- DEMST
0001- DIREC
E40C- DISASM
E48A- DISONR
EA61- DISPOF
E79F- DIV
E79C- DIV.1
E799- DIV.2
E7A2- DIV1
E7AE- DIV2
E7DF- DIV5
E7BA- DIVL
F8D6- DLAYY
F8D4- DLY
FE1C- DLYX
FE1E- DLYY
EE64- DOCMD
EE66- DOCMD1
EE6D- DOCMD2
EE7B- DOCMD3
EE8A- DOCMD4
FEFA- DONE8
0004- DORBIT
0005- DORREF
002C- DOWBUF
F5C9- DS.COLSON
.01=F5D1
F689- DS.HDSTPR
F3FC- DSPADD
E451- DSPBYT
0001- DSPFLG
003B- DSPIMG
EA87- DSPMD1
EA8F- DSPMD2
F095- DSPMEM
EA82- DSPMOD
E44E- DSPWRD
E7EA- DVO
F5A0- EN.COLSON
.01=F5A5
E487- ENBSNR
0008- ENBSON
003D- ENDIMG
EB34- ENDLIN
EED0- ENTLIF
F427- ERRMSG
F41C- ERROR
F410- ERRORX
F413- ERRORZ
F0FA- ERRXXX
EEE9- ETLIF1
0003- ETX
E52F- EXP1IN
E525- EXP1OT
E541- EXP2IN
E537- EXP2OT
E51D- EXTIN
E513- EXTOUT

0007- FCTSON
FA65- FILBUF
FA13- FILIMG
FB93- FILIN1
FB7F- FILINE
F667- FIND.LEFT.STOP
.01=F66F, .03=F688
005A- FLG
005B- FLH
E439- FNCKEY
0000- FRESPO
F511- FTCHBT
FA60- FULLIN
E854- FXA
E86E- FXB
E864- FXN
E87A- FXO
E86B- FXP
EF06- GADD01
EF2A- GADD02
EF3F- GADD03
F615- GET.AVG.RNG
.01=F620, .02=F648
F5D2- GET.COL.RNG
F597- GET.SON.RNG
EEF4- GETAD1
EEFA- GETAD2
EF00- GETAD3
EF04- GETADD
E463- GETAKY
E4AC- GETALL
FA38- GETCD1
FE96- GETCL1
FE89- GETCLK
FA35- GETCMD
FE68- GETFK1
FE51- GETFK2
FE47- GETFNC
F200- GETIM0
F218- GETIM1
F222- GETIM2
F227- GETIM3
F22D- GETIM4
F23C- GETIM5
F244- GETIM6
F251- GETIM7
F1FA- GETIME
FA49- GETKEY
FA4C- GETKY1
FA43- GETLIN
F71E- GETMSG
E48D- GETRNG
E445- GETRTC
E499- GETTIM
EF88- GETVAL
EFB7- GETVL1
EFCF- GETVL2
EFE7- GETVL3
F4E9- GETXFA
EEC3- GODOIT

EEC6- GOJUMP
E418- GTLINE
F1BC- HANDLR
F1DB- HANDLS
0000- HDINIT
0000- HDSONR
0050- HDSTAT
F3DF- HEXAS1
F3D5- HEXASC
055E- HEXCHR
F3AF- HEXERR
F3B9- HEXOUT
FDEE- HIBIT
0010- HIMEM
E619- HISPD
F1C1- HNDLR1
F1E0- HNDLS1
E457- HXTOAS
055D- HZPOS
E6A0- ICO
E697- INC.1
E69A- INC.AC
EB40- INCADR
FD64- INCH
FD74- INCH0
FD7D- INCH1
FABF- INCHAR
0037- INDEX1
F662- INIT.HEAD
E46F- INITHD
FC09- INKEY
FC2B- INKEY1
FC2D- INKEY2
FC2B- INKEY4
E6BA- INQ
E6B3- INT
E6B0- INT.1
F150- INTAPL
E430- INTCOM
F156- INTPIA
E406- INTRET
E7EE- INV
E7EB- INV.1
E300- IOSPC3
E380- IOSPC4
FFFE- IR0
FF27- IRQHD1
FF2A- IRQHD2
FF18- IRQHND
0014- IRQLOC
E074- KBDADD
E460- KBDCHR
0554- KEYCHR
0004- LBFSON
E45D- LCDDSP
FAC9- LCDOT1
FADF- LCDOT2
FAE2- LCDOT3
FAFD- LCDOT4
FB06- LCDOT5

FB0F- LCDOT6
FB2B- LCDOT7
FB39- LCDOT8
FB3D- LCDOT9
FB3F- LCDOTZ
FAC2- LCDOUT
E97F- LET.2
E472- LFTHDS
0080- LFTSPB
006F- LIFEAD
0200- LINBUF
0555- LINCNT
EF61- LINDAT
EF66- LNDT01
EF83- LNDT02
E5D4- LODBNK
E5DE- LODXBK
0012- LOMEM
E611- LOSPD
FDAA- LOWBIT
0006- LSDSON
0003- LTFSON
E983- LTL
0563- MASK
F076- MCMP01
F0C3- MCMP02
F087- MCMP03
F06D- MEMCMP
EFFE- MEMMOV
F00E- MEMOV1
F00F- MEMOV2
F02D- MEMOV3
F047- MEMOV4
E798- MLO
F950- MNMSG
FB49- MNTR0
FB54- MNTR1
FB8D- MNTR10
FB5B- MNTR2
F905- MNTR20
FB5D- MNTR3
FB73- MNTR5
F910- MNTR50
F925- MNTR70
F93E- MNTR80
F94D- MNTR90
EA91- MODEL1
EA94- MODLP2
EA9C- MODLP3
E403- MONCOM
F254- MONDAT
E400- MONITR
E409- MOVBLK
EFF6- MOVMEM
E994- MPL
E9A4- MRL
E782- MROR
F116- MSG100
0300- MSGBUF
F115- MSGEND

F596- MSGEXT
0052- MSGNUM
F107- MSGOUT
0077- MSGPTR
F587- MSGSND
F7ED- MSGXMT
.01=F804, .02=F80C, .03=F820, .04=F826
.05=F82B, .06=F831, .07=F83A
E76A- MSHL
FD4C- MSKDTA
0567- MSPSTP
E751- MUL
E74E- MUL.1
E74B- MUL.2
E754- MUL1
FFFA- NMI
FFCB- NMIHND
0016- NMILOC
E07C- NMIRST
EB44- NUMBYT
EF44- NWLNDT
F354- NWYEAR
F362- NWYR1
F376- NWYR2
F191- NXTCMD
F0C4- NXTLIN
F0CC- NXTLN1
F0D8- NXTLN2
FD9D- NXTRBT
FDE1- NXTSBT
E424- OFFSRL
F6BC- ONE.STEP
.01=F6CB, .02=F6D2
E478- ONESTP
ED44- OFCODE
FD13- OPENP
FD1B- OPENP1
FD45- OPENP2
FD48- OPENP3
FD33- OPENP4
FD39- OPENP5
E427- OPNSRL
EC44- OPTABL
FAAD- OTCHR0
FAB6- OTCHR1
FAB9- OTCHR2
F3B2- OUTBYT
FDB4- OUTCH
FDC5- OUTCH1
FAA0- OUTCHR
FDF0- OUTDTA
E43F- OUTMSG
0549- PACKBF
E030- PARADD
FC56- PAUS0
FC57- PAUS1
FC59- PAUS2
FC5F- PAUS3
FC61- PAUS4
FC64- PAUS5

FC48- PAUSE
E466- PAUSEY
0049- PCH
0047- PCL
F154- PIAINP
FC99- POLPRT
FCBB- POLPT1
FCC4- POLPT2
FCDE- POLPT3
056C- PORTNM
F694- POSHD
.01=F69C, .02=F6A1, .03=F6AE, .04=F6B2
EE3F- POSTFX
E4D4- PRCIN
E4CA- PRCOUT
EE13- PREFIX
E46C- PRINIT
F3F5- PRNTAX
055B- PROMPT
EAC4- PRTAD1
EAC1- PRTAD2
EAAA- PRTADD
FE2E- PRTOT0
FE39- PRTOT1
FE29- PRTOUT
EAD3- PRTPFX
EA9F- PRTPRF
E469- PSERPT
EACE- PSTFIX
EADB- PSTFX1
EADB- PSTFX2
EAE3- PSTFX3
EAE6- PSTFX4
EAEC- PSTFX5
E9EE- QAL
0056- QCH
0054- QCL
0055- QCM
E9F8- QRL
0063- RANGE
0002- RBFSON
005E- RCH
F3A3- RCHK01
005C- RCL
005D- RCM
E436- RCVCOM
E4C1- RCVSER
FABC- RDCHAR
FC03- RDKEY
055F- RDMSK
FEDF- RDPORT
.01=FEE7
E484- RDRANG
0001- RDRF
F186- RECCHR
F18F- RECCMD
F177- RECKEY
FD93- RECVCH
0550- REG
004B- REGA

FF3C- REGLIN
004E- REGP
004F- REGS
FF4A- REGTAB
004C- REGX
004D- REGY
EAFA- RELAD1
EAFE- RELAD2
EB09- RELAD3
EB23- RELAD4
EB2A- RELAD5
EB13- RELAD6
EB15- RELAD7
EAF2- RELADD
F83B- RESENT
FFFC- RESET
E621- RESMSP
FFC7- RETINT
DFFD- RFCHOT
E500- RFMIN
E4EE- RFMOUT
FF69- RGLN10
0040- RGTSPB
00EF- RGTSTP
F438- RMTCOM
E070- RNDGEN
E018- RNGADD
F398- RNGCHK
0074- ROSTAT
002F- ROWCNT
0005- RSDSON
E62D- RSMSP1
E448- RTCALM
E012- RTCBAS
E017- RTCCSR
E011- RTCFRE
E44B- RTCOFF
0001- RTFSON
0561- RTSLO
E4A2- SAVALL
E5AB- SAVBNK
0556- SAVEA
0557- SAVEX
0558- SAVEY
FFB9- SAVREG
E5B3- SAVXBK
E6E8- SBL
0067- SCH
0065- SCL
0066- SCM
E000- SCNADD
053F- SCNEND
0400- SCNIMG
FBCB- SCROL1
FBF6- SCROL2
FC01- SCROL3
FBBC- SCROLL
E020- SERADD
F43F- SERCOM
F53F- SERMSG

FCE9- SEROFF
F520- SERPIN
F52A- SERPN1
F52D- SERPOT
056B- SERPRT
F53C- SERPT1
F539- SERPT2
F4C6- SERRC1
F4BC- SERRCV
F48D- SERSN1
F483- SERSND
F44B- SERXF1
F44D- SERXF2
F46F- SERXF3
F446- SERXFR
FEB3- SETAL1
FEA6- SETALM
FE79- SETCL1
FE6C- SETCLK
E607- SETHSP
F2D3- SETIM0
F2DE- SETIM2
F2EC- SETIM3
F2ED- SETIM4
F2FA- SETIM5
F319- SETIM7
F2CD- SETIME
E5FD- SETLSP
F131- SETRCV
E442- SETRTC
F142- SETSND
E496- SETTIM
E6F5- SGN
E6F2- SGN.1
EA40- SHOWBT
E87E- SIN
E87B- SIN.1
E42A- SINCH
F0E1- SLAV10
F0F3- SLAV20
F0DB- SLAVE
E100- SLOT1A
E080- SLOT1R
E180- SLOT2A
E0A0- SLOT2R
E200- SLOT3A
E0C0- SLOT3R
E280- SLOT4A
E0E0- SLOT4R
E893- SNA
E89B- SNB
F16B- SNDCHR
E433- SNDCOM
E4B6- SNDSER
F16E- SNDWAT
E8A3- SNG
E8BB- SNO
E8B4- SNS
E8AE- SNT
F5BF- SONARS

00F7- SONOFF
E42D- SOUTCH
0068- SPARES
0006- SPCFLG
FC6E- SPCOUT
E49C- SPCRGT
0000- SPKFLG
FC6C- SPOUT
0009- SPRSON
0566- SPRTNM
E815- SQL
E833- SQO
E804- SQR
E801- SQR.1
0075- STAFLG
FCF9- STBAUD
F6E5- STEP1.5DEGRS
.01=F6EB, .02=F6FB, .03=F707, .04=F719
.05=F71C
0569- STEPNO
E47B- STEPPR
0568- STEPS
E9B0- STL
E475- STPOFF
F6E1- STPSEQ
E9AE- STR
FA55- STRLIN
E6E5- SUB
E6E2- SUB.1
E6DF- SUB.2
0080- SUBOP
0005- SUBP0
0080- SUBP1
00F0- SUBP2
0098- SUSERP
E580- SWBACK
E590- SWBANK
FC85- TABOT1
FC77- TABOUT
E49F- TABRGT
E9C6- TAL
E8C8- TAN
E8C5- TAN.1
E9CE- TAQ
E9BA- TAR
E9C4- TAT
0059- TCH
0057- TCL
0058- TCM
0562- TDLO
0031- TEMP00
0032- TEMP01
0033- TEMP02
0034- TEMP03
0035- TEMP04
0036- TEMP05
0025- TIMBUF
E990- TMA
005F- TMP
E9A0- TMR

E6C2- TNC
E6BF- TNC.1
E6B7- TNO
EBD7- TNO
E6BE- TOK
E9EC- TQA
E9F6- TQR
EA00- TRA
EA02- TRA.1
E634- TRG
E9BC- TRL
0076- TRNST
E9D8- TTA
E9DA- TTL
E9E2- TTR
E47E- TURNHD
E9E4- TXL
F748- UP1
F74A- UP2
F755- UP3
F75A- UP4
F75F- UP5
F77A- UP6
F793- UP7
F7A5- UP8
054C- UPACBF
E43C- UPCMSG
006D- USRLIV
0022- USRMON
0071- USRVOC
F378- VALBCD
0600- VAR
F381- VBCD2
E4E6- VIOSIN
E4DC- VIOSOT
00DF- VSWOFF
0020- VSWON
055C- VTPOS
FC37- WAITKY
0005- WATFLG
FC47- WTKEY1
000A- XDUCRS
F19D- XFRADD
F482- XFRQT1
F474- XFRQUT
FA9B- XPRMPT
002A- YEARBF
0000- ZERO
FFD1- ZERONM
FFD5- ZZEND

0000 ERRORS IN ASSEMBLY

: