

# Monitor program listing

SCIOS

TITLE SCIOS

```
DEVELOPED FROM SCMPKB MONITOR
BY D.J.D.
TAPE ROUTINES BY N.J.T.
```

```
0F00 RAM      =    0F00
0D00 DISP     =    0D00
; RAM OFF-SET
0000 DL       =    0      ;SEGMENT FOR DIGIT 1
0001 DH       =    1      ;SEGMENT FOR DIGIT 2
0002 D3       =    2      ;SEGMENT FOR DIGIT 3
0003 D4       =    3      ;SEGMENT FOR DIGIT 4
0004 ADLL     =    4      ;SEGMENT FOR DIGIT 5
0005 ADLH     =    5      ;SEGMENT FOR DIGIT 6
0006 ADHL     =    6      ;SEGMENT FOR DIGIT 7
0007 ADHH     =    7      ;SEGMENT FOR DIGIT 8
0008 D9       =    8      ;SEGMENT FOR DIGIT 9
0009 CNT      =    9      ;COUNTER.
000A PUSHED   =    10     ;KEY PUSHED.
000B CHAR     =    11     ;CHAR READ.
000C ADL      =    12     ;MEMORY ADDRESS LOW
000D WORD     =    13     ;MEMORY WORD.
000E ADH      =    14     ;MEMORY ADDRESS HI.
000F DDTA     =    15     ;FIRST FLAG.
0010 ROW      =    16     ;ROW COUNTER.
0011 NEXT     =    17     ;FLAG FOR NOW DATA.
```

RAM POINTERS USED BY SCIOS, P3 IS SAVED ELSEWHERE

```
OFF9 P1H      =    OFF9
OFFA P1L      =    OFFA
OFFB P2H      =    OFFB
OFFC P2L      =    OFFC
OFFD A        =    OFFD
OFFE E        =    OFFE
OFFF S        =    OFFF
```

```

; MONITOR OPERATION SUMMARY
;
; INITIALLY IN 'ADDRESS ENTRY' MODE
;
; TERM
; CHANGE TO 'DATA-ENTRY' MODE
;
; MEM:
; INCREMENT MEMORY ADDRESS
;
; ABORT:
; CHANGE TO 'ADDRESS ENTRY' MODE
;
; GO:
;
; THE REGISTERS ARE LOADED FROM RAM AND PROGRAM
; IS TRANSFERRED USING XPPC P3.
; TO GET BACK DO A XPPC P3.
;
; MONITOR LISTING

```

```

0000 00          HALT          ;ZEROS DISPLAYED ON RESET
0001 CFFF INIT:  ST @-1(3)    ;SO P3 = -1

0003 901E          JMP START

0005
;              DEBUG EXIT
;              RESTORE ENVIRONMENT
GOOUT:
0005 37          XPAH 3
0006 C20C        LD  ADL(2)
0008 33          XPAL 3
0009 C7FF        LD  @-1(3)    ;FIX GO ADDRESS.
000B C0F2        LD  E          ;RESTORE REGISTERS.
000D 01          XAE
000E C0EB        LD  P1L
0010 31          XPAL 1
0011 C0E7        LD  P1H
0013 35          XPAH 1
0014 C0E7        LD  P2L
0016 32          XPAL 2
0017 C0E3        LD  P2H
0019 36          XPAH 2
001A C0E4        LD  S

```

```

001C 00          HALT          ;RESET SINGLE-STEP
001D 07          CAS
001E CODE        LD      A
0020 08          NOP
0021 05          IEN
0022 3F          XPPC  3
                  ENTRY POINT FOR DEBUG
;
0023          ; START:
0023 C8D9        ST      A
0025 40          LDE
0026 C8D7        ST      E
0028 06          CSA
0029 C8D5        ST      S
002B 35          XPAH  1
002C C8CC        ST      P1H
002E 31          XPAL  1
002F C8CA        ST      P1L
0031 C40F        LDI     H(RAM) ;POINT P2 TO RAM
0033 36          XPAH  2
0034 C8C6        ST      P2H
0036 C400        LDI     L(RAM)
0038 32          XPAL  2
0039 C8C2        ST      P2L
003B C701        LD      @ 1 (3) ;BUMP P3 FOR RETURN
003D 33          XPAL  3 ;SAVE P3
003E CA0C        ST      ADL(2)
0040 37          XPAH  3
0041 CA0E        ST      ADH(2)
0043 C400        LDI     0
0045 CA02        ST      D3(2)
0047 CA03        ST      D4(2)
0049 C401        LDI     1
004B 37          XPAH  3
004C          ABORT:
004C 906D        JMP     MEM
004E          GONOW:
004E C20E        LD      ADH(2)
0050 90B3        JMP     GOOUT
001C 00
;
;
;          TAPE INTERFACE ROUTINES
;
00D5          COUNT = 0D5
00D6          LEN   = 0D6
;
;          STORE TO TAPE = 0052
;
0052 C501        TOTAPE: LD1(1)
0054 01          XAE
0055 C401        LDI 1
0057 CBD5        NEXT:  ST COUNT(3)
0059 C401        LDI 1

```

```

005B 07          CAS
005C 8F08       DLY 8
005E C3D5       LDCOUNT(3)
0060 50         ANE
0061 9807       JZ ZERO
0063 8F18       DLY 018
0065 C400       LDI 0
0067 07         CAS
0068 9005       JMP DONE
006A C400       ZERO: LDI 0
006C 07         CAS
006D 8F18       DLY 018
006F 8F20       DONE: DLY 020
0071 C3D5       LDCOUNT(3)
0073 F3D5       ADDCOUNT(3)
0075 9CE0       JNZ NEXT
0077 BBD6       DLD LEN(3)
0079 9CD7       JNZ TOTAPE
007B 3F         XPPC 3

;
;          LOAD FROM TAPE = 007C
;
007C C408       FRTAPE: LDI 8
007E CBD5       STCOUNT(3)
0080 06         LOOP:  CSA
0081 D420       ANI 20
0083 98FB       JZ LOOP
0085 8F1C       DLY 01C
0087 19         S10
0088 8F1C       DLY 01C
008A BBD5       DLD COUNT(3)
008C 9CF2       JNZ LOOP
008E 40         LDE
008F CD01       ST@1 (1)
0091 90E9       JMP FRTAPE

;
;          OFFSET CALCULATION = 0093
;
0093          OFFSET:
0093 C6FE       LD@-2 (2) ;Subtract 2 from
;destination address
0095 32         XPAL 2 ;Put low byte in AC
0096 03         SCL ;Set carry for subtraction
0097 FBD8       CAD OD8(3) ;Subtract low byte of jump
;instruction address
0099 C901       ST + 1 (1) ;Put in jump operand
009B 3F         XPPC 3 ;Return to monitor

009C 08         NOP

009D          DTACK:
009D AAOE       ILD ADH(2)
009F 9036       JMP DATA

```

```

00A1          MEMDN:
00A1 C20E    LD   ADH(2)   ;PUT WORD IN MEM.
00A3 35      XPAH 1
00A4 C20C    LD   ADL(2)
00A6 31      XPAL 1
00A7 C20D    LD   WORD(2)
00A9 C900    ST   (1)
00AB 9034    JMP  DATAK

00AD          MEMCK:
00AD E406    XRI  06           ;CHECK FOR GO.
00AF 989D    JZ                ;GONOW
00B1 E405    XRI  05           ;CHECK FOR TERM.
00B3 9822    JZ   DATA       ;CHECK IF DONE.
00B5 AA0C    ILD  ADL(2)   ;UPDATE ADDRESS LOW.
00B7 9C1E    JNZ  DATA
00B9 90E2    JMP  DTACK
;
; MEM KEY PUSHED
00BB          MEM:
00BB C4FF    LDI  -1           ;SET FIRST FLAG.
00BD CA11    ST   NEXT(2)   ;SET FLAG FOR ADDRESS NOW.
00BF CA0F    ST   DDTA(2)
00C1          MEML:
00C1 C20E    LD   ADH(2)
00C3 35      XPAH 1           ;SET P1 FOR MEM ADDRESS.
00C4 C20C    LD   ADL(2)
00C6 31      XPAL 1
00C7 C100    LD   (1)
00C9 CA0D    ST   WORD(2)   ;SAVE MEM DATA.
00CB C43F    LDI  L(DISP D)-1 ;FIX DATA SEG.
00CD 33      XPAL 3
00CE 3F      XPPC 3           ;GO TO DISPD SET SEG FOR DATA.
00CF 90DC    JMP  MEMCK       ;COMMAND RETURN.
00D1 C41A    LDI  L(ADR)-1   ;MAKE ADDRESS.
00D3 33      XPAL 3
00D4 3F      XPPC 3
00D5 90EA    JMP  MEML        ;GET NEXT CHAR.
00D7          DATA:
00D7 C4FF    LDI  -1           ;SET FIRST FLAG.
00D9 CA0F    ST   DDTA(2)
00DB C20E    LD   ADH(2)   ;SET P1 TO MEMORY ADDRESS
00DD 35      XPAH 1
00DE C20C    LD   ADL(2)
00E0 31      XPAL 1
00E1 C100    LD   (1)         ;READ DATA WORD.
00E3 CA0D    ST   WORD(2)   ;SAVE FOR DISPLAY.

.PAGE
00E5          DATAL:
00E5 C43F    LDI  L(DISP D)-1 ;FIX DATA SEG.
00E7 33      XPAL 3
00E8 3F      XPPC 3           ;FIX DATA SEG-GO TO DISPD.

```



```

011B C404      LDI 4      ;SET NUMBER OF SHIFTS.
011D CA09      ST CNT(2)
011F AA0F      ILD DDTA(2) ;CHECK IF FIRST.
0121 9C06      JNZ NOTFST ;JMP IF NO.
0123 C400      LDI 0      ;ZERO ADDRESS.
0125 CA0E      ST ADH(2)
0127 CA0C      ST ADL(2)
0129          NOTFST:
0129 02        CCL          ;CLEAR LINK.
012A C20C      LD ADL(2)    ;SHIFT ADDRESS LEFT 4 TIMES.
012C F20C      ADD ADL(2)
012E CA0C      ST ADL(2)    ;SAVE IT.
0130 C20E      LD ADH(2)    ;NOW SHIFT HIGH.
0132 F20E      ADD ADH(2)
0134 CA0E      ST ADH(2)
0136 BA09      DLD CNT(2)    ;CHECK IF SHIFTED 4 TIMES.
0138 9CEF      JNZ NOTFST ;JMP IF NOT DONE.
013A C20C      LD ADL(2)    ;NOW ADD NEW NUMBER.
013C 58        ORE
013D CA0C      ST ADL(2)    ;NUMBER IS NOW UP DATED.
013F 3F        XPPC 3
                .PAGE 'DATA TO SEGMENTS'

```

```

;          CONVERT HEX DATA TO SEGMENTS.
;          P2 POINTS TO RAM.
;          DROPS THRU TO HEX ADDRESS CONVERSION.

```

```

0140          DISPD:
0140 C401      LDI H(CROM) ;SET ADDRESS OF TABLE.
0142 35        XPAH 1
0143 C40B      LDI L(CROM)
0145 31        XPAL 1
0146 C20D      LD WORD(2) ;GET MEMORY WORD.
0148 D40F      ANI OF
014A 01        XAE
014B C180      LD -128(1) ;GET SEGMENT DISP.
014D CA00      ST DL(2)    ;SAVE AT DATA LOW.
014F C20D      LD WORD(2) ;FIX HI.
0151 1C        SR          ;SHIFT HI TO LOW.
0152 1C        SR
0153 1C        SR
0154 1C        SR
0155 01        XAE
0156 C180      LD -128(1) ;GET SEGMENTS.
0158 CA01      ST DH(2)    ;SAVE IN DATA HI.

```

```

PAGE          ADDRESS TO SEGMENTS

```

```

;          CONVERT HEX ADDRESS TO SEGMENTS.
;          P2 POINTS TO RAM.
;          DROPS THRU TO KEYBOARD AND DISPLAY.

015A      DISPA:
015A 03      SCL
015B C401    LDI    H(CROM) ;SET ADDRESS OF TABLE.
015D 35      XPAH  1
015E C40B    LDI    L(CROM)
0160 31      XPAL  1
0161      LOOPD:
0161 C20C    LD     ADL(2) ;GET ADDRESS.
0163 D40F    ANI    OF
0165 01      XAE
0166 C180    LD     -128(1) ;GET SEGMENTS.
0168 CA04    ST     ADLL(2) ;SAVE SEG OF ADR LL.
016A C20C    LD     ADL(2)
016C 1C      SR
;SHIFT HI DIGIT TO LOW.
016D 1C      SR
016E 1C      SR
016F 1C      SR
0170 01      XAE
0171 C180    LD     -128(1) ;GET SEGMENTS.
0173 CA05    ST     ADLH(2)
0175 06      CSA     ;CHECK IF DONE.
0176 D480    ANI    080
0178 9809    JZ     DONE
017A 02      CCL     ;CLEAR FLAG.
017B C400    LDI    0
017D CA03    ST     D4(2) ;ZERO DIGIT 4.
017F C602    LD     @2(2) ;FIX P2 FOR NEXT LOOP.
0181 90DE    JMP    LOOPD
0183      DONE:
0183 C6FE    LD     @-2(2) ;FIX P2.
.PAGE 'DISPLAY AND KEYBOARD INPUT'

;          CALL XPPC 3
;
;          JMP COMMAND IN A GO = 6, MEM = 7, TERM = 3
;          IN E GO = 22, MEM = 23, TERM = 27.
;          NUMBER RETURN HEX NUMBER IN E REG.
;
;          ABORT KEY GOES TO ABORT.
;          ALL REGISTERS ARE USED.
;
;          P2 MUST POINT TO RAM. ADDRESS MUST BE XXX0.
;
;          TO RE-EXECUTE ROUTINE DO XPPC3.

0185      KYBD:
0185 C400    LDI    0 ;ZERO CHAR.

```



```

0187 CA0B      ST   CHAR(2)
0189 C40D      LDI   H(DISPLAY) ;SET DISPLAY ADDRESS.
018B 35        XPAH  1
018C          OFF:
018C C4FF      LDI   -1 ;SET ROW/DIGIT ADDRESS.
018E CA10      ST   ROW(2) ;SAVE ROW COUNTER.
0190 C40A      LDI   10 ;SET ROW COUNT.
0192 CA09      ST   CNT(2)
0194 C400      LDI   0
0196 CA0A      ST   PUSHED(2);ZERO KEYBOARD INPUT.
0198 31        XPAL  1 ;SET DISP ADDRESS LOW.
0199          LOOP:
0199 AA10      ILD   ROW(2) ;UP DATE ROW ADDRESS.
019B 01        XAE
019C C280      LD   -128(2) ;GET SEGMENT.
019E C980      ST   -128(1) ;SEND IT.
01A0 8F00      DLY   0 ;DELAY FOR DISPLAY.
01A2 C180      LD   -128(1) ;GET KEYBOARD INPUT.
01A4 E4FF      XRI   OFF ;CHECK IF PUSHED.
01A6 9C4C      JNZ   KEY ;JUMP IF PUSHED.
01A8          BACK:
01A8 BA09      DLD   CNT(2) ;CHECK IF DONE.
01AA 9CED      JNZ   LOOP ;NO IF JUMP.
01AC C20A      LD   PUSHED(2);CHECK IF KEY.
01AE 980A      JZ    CKMORE
01B0 C20B      LD   CHAR(2) ;WAS THERE A CHAR?
01B2 9CD8      JNZ   OFF ;YES WAIT FOR RELEASE.
01B4 C20A      LD   PUSHED(2);NO SET CHAR.
01B6 CA0B      ST   CHAR(2)
01B8 90D2      JMP   OFF
01BA          CKMORE:
01BA C20B      LD   CHAR(2) ;CHECK IF THERE WAS A CHAR.
01BC 98CE      JZ    OFF ;NO KEEP LOOKING.

```

.PAGE

; COMMAND KEY PROCESSING

```

01BE          COMMAND:
01BE 01        XAE ;SAVE CHAR.
01BF 40        LDE ;GET CHAR.
01C0 D420      ANI   020 ;CHECK FOR COMMAND.
01C2 9C28      JNZ   CMND ;JUMP IF COMMAND.
01C4 C480      LDI   080 ;FIND NUMBER.
01C6 50        ANE
01C7 9C1B      JNZ   LT7 ;0 TO 7.
01C9 C440      LDI   040
01CB 50        ANE
01CC 9C19      JNZ   N89 ;8 OR 9.
01CE C40F      LDI   OF
01D0 50        ANE
01D1 F407      ADI   7 ;MAKE OFF SET TO TABLE.
01D3 01        XAE ;PUT OFF SET AWAY.

```

```

01D4 C080      LD    -128(0)    ;GET NUMBER.
01D6          KEYRTN:
01D6 01        XAE                ;SAVE IN E.
01D7 C702      LD    @2(3)      ;FIX RETURN.
01D9 3F        XPPC 3          ;RETURN.
01DA 90A9      JMP    KYBD          ;ALLOWS XPPC P3 TO RETURN.

01DC 0A0B      .BYTE 0A,0B,0C,0D,0,0,0E,0F
01DE 0C0D
01E0 0000
01E2 0E0F
01E4          LT7:
01E4 60        XRE                ;KEEP LOW DIGIT.
01E5 90EF      JMP    KEYRTN
01E7          N89:
01E7 60        XRE                ;GET LOW.
01E8 F408      ADI    08          ;MAKE DIGIT 8 OR 9.
01EA 90EA      JMP    KEYRTN

          .PAGE
01EC          CMND:
01EC 60        XRE
01ED E404      XRI    04          ;CHECK IF ABORT.
01EF 9808      JZ     ABRT        ;ABORT.
01F1 3F        XPPC 3          ;IN E 23 = MEM,22 = GO,27 = TERM
          ;IN A 7 = MEM,6 = GO,3 = TERM.
01F2 9091      JMP    KYBD        ;ALLOWS JUST A XPPC P3 TO
          ;RETURN.

01F4          KEY:
01F4 58        ORE                ;MAKE CHAR.
01F5 CA0A      ST     PUSHED(2)    ;SAVE CHAR.
01F7 90AF      JMP    BACK

01F9          ABRT:
01F9 C400      LDI    H(ABORT)
01FB 37        XPAH 3
01FC C44B      LDI    L(ABORT)-1
01FE 33        XPAL 3
01FF 3F        XPPC 3          ;GO TO ABORT
          0000      .END

```