

```

; *****
;
;                               *
;                               *
;                               *
;                               *
;                               *
; *****

```

; Constants

```

0800      TLINE  EQU  4000Q      ; Start of display memory (line 0)
0032      TSTA   EQU  62Q        ; Top line normal status
                                ; Normal - white on black
                                ; Special - bright white on black
0022      TSTB   EQU  42Q        ; Top line complement status (for special)
0020      STN    EQU  40Q        ; Normal display status -
                                ; black on white/black on bright white
0018      NLINE  EQU  24         ; Number of lines on display
0050      NCHAR  EQU  80         ; Number of characters displayed per line
                                ; (+1 for status)
0852      DBGIN  EQU  4122Q      ; Address of 1st character for line 1
0FE8      DEND   EQU  7750Q      ; Address of last character in last line
0039      ONCUR  EQU  71Q        ; Cursor on pulse
003A      OFCUR  EQU  72Q        ; Cursor off pulse
003B      ABELL  EQU  73Q        ; Pulse for audio indicator
003C      PICU   EQU  74Q        ; Priority interrupt decoder port address
003D      KEYBD  EQU  75Q        ; Keyboard port address
003E      UPORT  EQU  76Q        ; Serial line interface data port address
003F      USTAT  EQU  77Q        ; Serial line interface status port address
1FFF      STACK  EQU  8191      ; Top of stack memory
0006      ACK    EQU  6Q         ; Ascii acknowledgement character (^F)
0015      NAK    EQU  25Q        ; Ascii no acknowledgement character (^U)
0007      LOGO   EQU  7         ; The following are the offsets from the
                                ; left edge of the top line at which
                                ; various messages are displayed

0012      DUP0   EQU  18
001D      MOD0   EQU  29
002D      ROLL0  EQU  45
003A      IDD0   EQU  58
0045      DL0    EQU  69

```

; Interrupt Locations

```
0000          ORG  0
0000 C35E00    JMP  DM          ; Begin program past interrupt locations
0008          ORG  100        ; Receive usart interrupt
0008 CD1001    CALL RXSV
000B C9       RET
0038          ORG  700        ; Receive keyboard interrupt
0038 CDBE00    CALL KBDSV
003B C9       RET
```

```
; *****
;
;           Table of addresses of useful routines          *
;           Do not change the order in this table         *
;           New addresses must be added at the end        *
;
; *****
```

```
0040          ORG  1000
0040 AC00     RCTBL: DW  TERM
0042 F202     DW  INIT
0044 5E00     DW  DM
0046 2901     DW  PCHAR
0048 3704     DW  MULT
004A B304     DW  IST
004C C204     DW  LSTAT
004E 2403     DW  FULL
0050 1503     DW  HALF
0052 0401     DW  TXCHR
0054 CF04     DW  PQUE
0056 DA04     DW  GQUE
0058 E104     DW  GQ
005A 3303     DW  LOCAL
005C 4203     DW  LINE
```

```

; *****
;
;           Begin Main Program
;
; *****

```

```

005E 31FF1F   DM:   LXI   SP, STACK           ; Initialize stack pointer
0061 21FC13           LXI   H, QTOP                ; Initialize queue pointers
0064 221010           SHLD  QHEAD
0067 221210           SHLD  QTAIL
006A 210008           LXI   H, TLINE              ; Blank top line of display
006D 3632            MVI   M, TSTA              ; Set top line status
006F 23             INX   H
0070 3E00            MVI   A, 0
0072 CDA103           CALL  BFILL
0075 110708           LXI   D, TLINE+LOGO        ; Display logo
0078 212505           LXI   H, LMESS
007B CD1B05           CALL  SDISP
007E CDE302           CALL  ROLL                 ; Put the terminal in roll mode
0081 CD1503           CALL  HALF                 ; Half duplex
0084 CD3303           CALL  LOCAL                ; Local mode (characters not
                                ; transmitted on line)
0087 CDB304           CALL  IST                  ; Initialize screen status words
008A CDF202           CALL  INIT                 ; Initialize variables and screen

                                ; Initialize interrupts

008D 3E4F            MVI   A, 4FH               ; Set mode of usart -
                                ; 1 stop bit, no parity,
                                ; 8 bit chars,
                                ; baud rate factor = 64x

008F D33F            OUT   USTAT
0091 3E37            MVI   A, 37H              ; Output command word
                                ; rts, error reset, receive enable,
                                ; dtr, transmit enable

0093 D33F            OUT   USTAT
0095 3E0F            MVI   A, 170              ; Initialize priority int. decoder
0097 D33C            OUT   PICU
0099 FB             EI                          ; Enable interrupts
009A CDDA04           CALL  GQUE                 ; Get first character
009D CDAB03           CALL  FRSTC                ; And accept or reject defaults
00A0 D339            OUT   ONCUR               ; Turn on the cursor
00A2 2A0210           LHLD  CURSE                ; Initialize hardware cursor register
00A5 7E             MOV   A, M

```

```
; *****  
;  
; Chief Wait Loop Of The Terminal *  
;  
; *****
```

```
00A6 CDAC00 WAIT: CALL TERM  
00A9 C3A600 JMP WAIT
```

```
; *****  
;  
; Main terminal routine *  
;  
; *****
```

```
; This routine processes all characters found in the queue, and  
; returns when complete. It can thus be used by any local program  
; to do terminal processing on a stream of characters.
```

```
00AC CDE104 TERM: CALL GQ ; Get character from the queue  
00AF D0 RNC ; Queue empty, return  
00B0 D33A OUT OFCUR ; Turn off the cursor  
00B2 CD2901 CALL PCHAR ; Process character  
00B5 D339 OUT ONCUR ; Turn on the cursor  
00B7 2A0210 LHLD CURSE ; Refresh hardware cursor register  
00BA 7E MOV A, M ; (Insert INX H for TMM board)  
00BB C3AC00 JMP TERM ; Next character
```

```

; *****
;
; Keyboard interrupt server
;
; *****

```

```

; ^U is an 'escape' character--it must be typed
; twice to get one transmitted. It is used to call
; the downloader and the local monitor. The
; downloader is called with the sequence ^U ^D.
; The local monitor is called with the sequence
; ^U ^V.
; If ^U is typed and followed by any character
; other than ^D or ^V, then only the second
; character is transmitted,

```

```

00BE F5      KBDSV;  PUSH  PSW
00BF C5      PUSH  B
00C0 D5      PUSH  D
00C1 E5      PUSH  H
00C2 DB3D    IN     KEYBD      ; Read input character
00C4 47      MOV    B, A        ; Save character in B Register

00C5 3A0910  LDA    LCMOD      ; Don't bother about ^U check
00C8 FE00    CPI    0        ; in local mode (the normal
                        ; character processor will
                        ; catch it)

00CA C2E300  JNZ    KTST
00CD 3A0E10  LDA    UFLG      ; Was last char ^U?
00D0 FE00    CPI    0
00D2 C2F600  JNZ    CTRLU    ; yes--jump out
00D5 3E15    MVI    A, 21     ; No
00D7 B8      CMP    B        ; Is this char a ^U?
00D8 C2E300  JNZ    KTST     ; No--process as usual
00DB 3EFF    MVI    A, 377Q
00DD 320E10  STA    UFLG     ; yes, set flag--and ignore
                        ; character

00E0 C31F01  JMP    RTRN1

00E3 3A0A10  KTST:  LDA    HDMOD ; Test for half duplex mode
00E6 FE00    CPI    0
00E8 C4CF04  CNZ    PQUE     ; If half duplex, then queue the
                        ; character for processing

00EB 3A0910  LDA    LCMOD     ; Test for local mode
00EE FE00    CPI    0
00F0 CC0401  CZ     TXCHR    ; If not in local mode, then
                        ; transmit character
00F3 C31F01  JMP    RTRN1    ; Initialize interrupts and
                        ; return from routine

```



```

; *****
;
;           Transmit Character Server
;
; *****

```

```

; B Register contains character to transmit
0104 F5 TXCHR: PUSH PSW
0105 DB3F POLL: IN USTAT ; Read usart status
0107 1F RAR
0108 D20501 JNC POLL ; Wait if tx port not available
010B 78 MOV A, B
010C D33E OUT UPORT ; Transmit character
010E F1 POP PSW
010F C9 RET

```

```

; *****
;
;           USART Receive Server
;
; *****

```

```

0110 F5 RXSV: PUSH PSW
0111 C5 PUSH B
0112 D5 PUSH D
0113 E5 PUSH H
0114 DB3E IN UPORT ; Get character
0116 47 MOV B, A ; Mask parity bit with PMASK
0117 3A0F10 LDA PMASK
011A A0 ANA B
011B 47 MOV B, A ; Move character for call to PQUE
011C CDCF04 CALL PQUE ; Queue character for processing

011F 3E0F RTRN1: MVI A, 170 ; Initialize priority interrupt decoder
0121 D33C OUT PICU
0123 FB EI

0124 E1 RETRN: POP H ; **** standard return routine ****
0125 D1 POP D
0126 C1 POP B
0127 F1 POP PSW
0128 C9 NRET: RET

```

```
; *****  
;  
; Character Processor  
;  
; *****
```

```
0129 FE7F PCHAR: CPI 177Q ; A Register contains input character  
012B C8 RZ ; Do not display rubouts  
012C FE20 CPI 40Q ; If this is displayable character then  
012E D25103 JNC DCHAR ; display it--else it is a ctrl character  
; use ctrl char branch table address plus  
; offset to get address of processor for  
; the received ctrl char  
  
0131 213E01 LXI H, CCTBL  
0134 4F MOV C, A  
0135 0600 MVI B, 0  
0137 09 DAD B ; Add offset to branch address  
0138 09 DAD B  
0139 5E MOV E, M ; Load routine address  
013A 23 INX H  
013B 56 MOV D, M  
013C EB XCHG  
013D E9 PCHL ; Jump to processor for the received  
; control character
```



```

; *****
;
;           Control Character Branch Table
;
; *****

```

```

013E 2801      CCTBL:  DW  NRET          ; ^@ (NULL)--Don't display
0140 2801      DW  NRET          ; ^A--SU-AI uses this for fill char--
                                ; don't display
0142 7E01      DW  HOME          ; ^B--Home cursor and clear xy
                                ; addressing mode
0144 9E01      DW  HUH           ; ^C
0146 9E01      DW  HUH           ; ^D
0148 9E01      DW  HUH           ; ^E
014A 9E01      DW  HUH           ; ^F
014C A101      DW  BELL          ; ^G--Ring bell
014E B201      DW  LEFT          ; ^H--LEFT/DC
0150 9E01      DW  HUH           ; ^I
0152 CC01      DW  LF            ; ^J--DOWN/IL
0154 9E01      DW  HUH           ; ^K
0156 0F02      DW  LCA           ; ^L--Load cursor address
0158 5602      DW  CR            ; ^M (CR)
015A 6202      DW  SB            ; ^N--Start blink
015C 9E01      DW  HUH           ; ^O
015E 6802      DW  DLE           ; ^P--Start I/D mode
0160 9E01      DW  HUH           ; ^Q
0162 9E01      DW  HUH           ; ^R
0164 9E01      DW  HUH           ; ^S
0166 9E01      DW  HUH           ; ^T
0168 7702      DW  CTUR          ; ^U (Escape Character)--Check for calls
016A 9E01      DW  HUH           ; ^V
016C 8C02      DW  EEOL          ; ^W--Erase to end of line
016E 9502      DW  UNLOK         ; ^X--End blink, roll, I/D, & xy addressing
                                ; modes
0170 9E01      DW  HUH           ; ^Y
0172 A502      DW  UP            ; ^Z--Up/delete line
0174 9E01      DW  HUH           ; Escape
0176 C902      DW  RIGHT         ; ^,--Right/insert space
0178 E302      DW  ROLL          ; ^J--Start roll mode
017A F202      DW  INIT          ; ^.--Clear memory, blink, and I/D modes
017C FF02      DW  BLANK         ; ^_--Blank screen

```

```

; *****
;
;           Control Character Processing Routines
;
; *****

```

; ^B Processor

```

017E 3E00   HOME:  MVI  A, 0
0180 320010 STA  XCURS           ; Home the cursor
0183 320110 STA  YCURS
0186 215208 LXI  H, DBGIN
0189 220410 SHLD CLINE          ; Address of column 0 of current

```

```

; line
018C 220210 SHLD CURSE          ; Absolute cursor address
018F 320710 MODES: STA  BKMOD          ; Turn off blink and I/D modes
0192 320610 STA  IDMOD
0195 213A08 LXI  H, TLINE+IDD0 ; Clear I/D mode display
0198 3E39   MVI  A, IDD0-1
019A CDA103 CALL BFill
019D C9     RET

```

; Processor For Control Characters Not Normally  
; transmitted by TENEX/TOPS-20

```

019E C35103 HUH:  JMP  DCHAR           ; Display character

```

; ^G Processor

```

01A1 D33B   BELL:  OUT  ABELL           ; Send pulse to audio indicator
01A3 210008 LXI  H, TLINE          ; Complement logo
01A6 7E     MOV  A, M
01A7 FE32   CPI  TSTA
01A9 C2AF01 JNZ  BEL1
01AC 3622   MVI  M, TSTB
01AE C9     RET
01AF 3632   BEL1: MVI  M, TSTA
01B1 C9     RET

```

; ^H Processor

```

01B2 3A0610    LEFT:  LDA  IDMOD
01B5 FE00      CPI   0           ; Check I/D mode
01B7 C2B403    JNZ  DLETE       ; If in I/D mode--delete character
01BA 3A0010    LDA  XCURS       ; Else--backspace cursor
01BD FE00      CPI   0
01BF C8        RZ           ; Cannot backspace off screen
01C0 3D        DCR  A
01C1 320010    STA  XCURS       ; Update column count
01C4 2A0210    LHL  CURSE
01C7 2B        DCX  H
01C8 220210    SHLD CURSE      ; Update cursor address
01CB C9        RET

```

; ^J (Line Feed) Processor

```

01CC 3A0610    LF:   LDA  IDMOD       ; Check I/D mode
01CF FE00      CPI   0
01D1 C20104    JNZ  AROW        ; If in I/D mode--add a row
01D4 3A0110    LDA  YCURS      ; Else--move the cursor
01D7 FE17      CPI  NLINE-1   ; Last row?
01D9 C2F901    JNZ  LF1        ; No
01DC 3A0810    LDA  RLMOD      ; Yes, check roll mode
01DF FE00      CPI   0
01E1 C27E03    JNZ  SCROL      ; If roll mode--scroll and return
01E4 215208    LXI  H, DBGIN  ; Else--move to line 0
01E7 220410    SHLD CLINE
01EA 3A0010    LDA  XCURS
01ED 5F        MOV  E, A
01EE 3E00      MVI  A, 0
01F0 320110    STA  YCURS
01F3 57        MOV  D, A
01F4 19        DAD  D
01F5 220210    SHLD CURSE
01F8 C9        RET
01F9 3C        LF1:  INR  A           ; Update cursor row
01FA 320110    STA  YCURS
01FD 2A0210    LHL  CURSE      ; Add NCHAR+1 to cursor address
0200 115100    LXI  D, NCHAR+1
0203 19        DAD  D
0204 220210    SHLD CURSE
0207 2A0410    LHL  CLINE      ; and to the current line indicator
020A 19        DAD  D
020B 220410    SHLD CLINE
020E C9        RET

```

; ^L (Cursor Addressing) Processor

```

020F CDDA04      LCA:  CALL  GQUE          ; Treat first char as x cursor
                                ; address
0212 CD4B02      CALL  CANCL         ; Can always abort with special char
0215 EE60        LCX:  XRI   140Q         ; Bits 5 and 6 must be
                                ; complemented!!!%*@@!!$<!!
0217 FE50        CPI   NCHAR         ; If column > NCHAR-1 then set to 0
0219 DA1E02      JC    XST
021C 3E00        MVI   A, 0
021E 320010      XST:  STA  XCURS         ; Save column number
0221 CDDA04      CALL  GQUE          ; Treat next char as y cursor address
0224 CD4B02      CALL  CANCL
0227 E61F        LCY:  ANI   37Q         ; Row address is in lower 5 bits
0229 FE18        CPI   NLINE         ; If row > NLINE-1 then set to 0
022B DA3002      JC    YST
022E 3E00        MVI   A, 0
0230 320110      YST:  STA  YCURS
0233 4F          MOV   C, A             ; Set up registers for call to
                                ; multiply routine
0234 1651        MVI   D, NCHAR+1
0236 CD3704      CALL  MULT           ; Mult row by NCHAR+1, result
                                ; to B, C
0239 215208      LXI   H, DBGIN
023C 09          DAD   B               ; Add video memory offset
023D 220410      SHLD  CLINE          ; Store start of current line
0240 3A0010      LDA   XCURS         ; D, E get column number
0243 5F          MOV   E, A
0244 1600        MVI   D, 0
0246 19          DAD   D               ; Add row address to column number
0247 220210      SHLD  CURSE         ; and store cursor
024A C9          RET
024B FE18        CANCL; CPI  30Q         ; ^X or ^B can cancel
024D CA9502      JZ    UNLOK         ; xy addressing mode
0250 FE02        CPI   2
0252 CA7E01      JZ    HOME
0255 C9          RET

```

; ^M (Carriage Return) Processor

```

0256 3E00        CR:   MVI   A, 0             ; Carriage return without line feed
0258 320010      STA  XCURS
025B 2A0410      LHLD  CLINE          ; Return cursor to begin of current
                                ; line
025E 220210      SHLD  CURSE
0261 C9          RET

```

```

; ^N Processor
0262 3E80      SB:  MVI  A, 2000      ; Start blink mode
0264 320710    STA  BKMOD
0267 C9       RET

; ^P Processor
0268 3EFF      DLE:  MVI  A, 3770      ; Mark I/D mode active
026A 320610    STA  IDMOD
026D 113A08    LXI  D, TLINE+IDD0    ; Display mode on top line
0270 214905    LXI  H, IMESS
0273 CD1B05    CALL SDISP
0276 C9       RET

; ^U Processor
0277 CDDA04    CTUR: CALL  GQUE      ; Get char following ^U
027A FE04      CPI   4              ; ^D?
027C CA4A04    JZ   DNLD      ; Yes, jump to downloader
027F FE16      CPI   22             ; ^V?
0281 CAFD13    JZ   MON       ; Yes, jump to local monitor
0284 FE15      CPI   21             ; ^U--display it
0286 C35103    JMP  DCHAR      ; DCHAR will return out of this
; routine
0289 C32901    JMP  PCHAR      ; Not a magic character--ignore
; the ^U and treat the following
; character In the usual way

; ^W Processor
028C 3A0010    EEOL: LDA  XCURS      ; Erase to end of line
028F 2A0210    LHL  CURSE
0292 C3A103    JMP  BFILL

; ^X Processor
0295 212D08    UNLOK: LXI  H, TLINE+ROLL0    ; Clear roll and I/D mode displays
0298 3E2C      MVI  A, ROLL0-1
029A CDA103    CALL BFILL
029D 3E00      MVI  A, 0
029F 320810    STA  RLMOD      ; Reset roll mode
02A2 C38F01    JMP  MODES      ; Reset blink and I/D modes

```

; ^Z Processor

```

02A5 3A0610    UP:   LDA   IDMOD           ; Check I/D mode
02A8 FE00      CPI   0
02AA C2E703    JNZ   DROW           ; If in I/D mode--delete a row
02AD 3A0110    LDA   YCURS         ; Else--move the cursor
02B0 FE00      CPI   0           ; Top row?
02B2 C8        RZ           ; Yes, do nothing
02B3 3D        DCR   A           ; No, decrement row counter
02B4 320110    STA   YCURS
02B7 2A0210    LHLD  CURSE         ; Subtract NCHAR+1 from the cursor
                                ; address

02BA 11AFFF    LXI   D, -(NCHAR+1)
02BD 19        DAD   D
02BE 220210    SHLD  CURSE
02C1 2A0410    LHLD  CLINE         ; also from the current line
                                ; indicator

02C4 19        DAD   D
02C5 220410    SHLD  CLINE
02C8 C9        RET

```

; ^, Processor

```

02C9 3A0610    RIGHT: LDA  IDMOD           ; Cursor right/insert blank
02CC FE00      CPI   0           ; Check I/D mode
02CE CAD702    JZ    ROK           ; If not in I/D mode--Space ahead
02D1 0620      MVI   B, ' '       ; Else--Insert a blank
02D3 C0CE03    CALL  ADDC
02D6 C9        RET
02D7 210010    ROK:  LXI   H, XCURS
02DA 34        INR   M           ; Increment column count
02DB 2A0210    LHLD  CURSE
02DE 23        INX   H           ; And increment cursor addr
02DF 220210    SHLD  CURSE
02E2 C9        RET

```

; ^J Processor

```

02E3 3EFF      ROLL:  MVI   A, 3770           ; Turn on roll mode
02E5 320810    STA   RLMOD
02E8 112D08    LXI   D, TLINE+ROLL0 ; Display roll mode on top line
02EB 214405    LXI   H, RMESS
02EE CD1B05    CALL  SDISP
02F1 C9        RET

```

; ^. Processor

```
02F2 3E00      INIT:  MVI  A, 0           ; Clear ^U flag
02F4 320E10    STA  UFLG
02F7 3E7F      MVI  A, 177Q          ; Set parity mask
02F9 320F10    STA  PMASK
02FC CDFF02    CALL BLANK           ; Clear screen
```

; ^\_ Processor--Blank Screen  
; (destroys machine state)

```
02FF 1618      BLANK: MVI  D, NLINE
0301 1E50      MVI  E, NCHAR
0303 215208    LXI  H, DBGIN        ; Start blanking here
0306 3620      WBLNK: MVI  M, ' '
0308 23        INX  H           ; Advance address pointer
0309 1D        DCR  E
030A C20603    JNZ  WBLNK          ; Iterate
030D 1E50      MVI  E, NCHAR        ; Refresh character counter
030F 23        INX  H           ; Skip status word
0310 15        DCR  D
0311 C20603    JNZ  WBLNK
0314 C9        RET
```

```
; *****  
;  
; Utility Routines  
;  
; *****
```

; Set terminal to half duplex mode

```
0315 3EFF HALF: MVI A, 377Q  
0317 320A10 STA HDMOD  
031A 111208 LXI D, TLINE+DUP0 ; Put up message  
031D 212E05 LXI H, HMESS  
0320 CD1B05 CALL SDISP  
0323 C9 RET
```

; Set terminal to full duplex mode

```
0324 3E00 FULL: MVI A, 0  
0326 320A10 STA HDMOD  
0329 111208 LXI D, TLINE+DUP0 ; Put up message  
032C 213105 LXI H, FMESS  
032F CD1B05 CALL SDISP  
0332 C9 RET
```

; Put terminal in local mode

```
0333 3EFF LOCAL: MVI A, 377Q  
0335 320910 STA LCMOD  
0338 111D08 LXI D, TLINE+MOD0  
033B 213405 LXI H, LOMES  
033E CD1B05 CALL SDISP  
0341 C9 RET
```

; Put terminal in on-line mode

```
0342 3E00 LINE: MVI A, 0  
0344 320910 STA LCMOD  
0347 111D08 LXI D, TLINE+MOD0  
034A 213C05 LXI H, LIMES  
034D CD1B05 CALL SDISP  
0350 C9 RET
```



```

; Character Display Routine
; A Register contains character to display

0351 47          DCHAR: MOV  B, A          ; Transfer char temporarily
0352 3A0010      LDA  XCURS              ; Ignore characters after NCHARth
                                           ; on a line

0355 FE50          CPI  NCHAR
0357 D0          RNC
0358 3A0710      LDA  BKMOD              ; BKMOD on/off = 200Q/0Q
035B B0          ORA  B                  ; OR character in with high bit
035C 47          MOV  B, A
035D 3A0610      LDA  IDMOD              ; Check for I/D mode
0360 FE00          CPI  0
0362 CA6E03      JZ   DOK
0365 CDCE03      CALL ADDC              ; I/D mode - insert character
0368 2A0210      LHL  CURSE
036B C37203      JMP  DFIN
036E 2A0210      DOK:  LHL  CURSE
0371 70          MOV  M, B              ; Normal mode - display character
0372 23          DFIN: INX  H
0373 220210      SHLD CURSE
0376 3A0010      LDA  XCURS              ; Update column count
0379 3C          INR  A
037A 320010      STA  XCURS
037D C9          RET

```

; Scroll routine

```

037E 21A308   SCROL: LXI  H, DBGIN+NCHAR+1   ; Initialize two pointers
0381 115208   LXI  D, DBGIN                   ; for transferring Chars
                                           ; (DBGIN+NCHAR+1 is address
                                           ; of 1st character on line 2)
0384 0617     MVI  B, NLINE-1                 ; Number of lines to be
                                           ; scrolled
0386 0E50     MVI  C, NCHAR                   ; Number of characters per
                                           ; line
0388 7E       SLOOP: MOV  A, M                 ; Load character
0389 12       STAX  D                 ; and store it
038A 13       INX  D                 ; Increment pointers
038B 23       INX  H
038C 0D       DCR  C                 ; Check count
038D C28803   JNZ  SLOOP
0390 0E50     MVI  C, NCHAR         ; Refresh character counter
0392 23       INX  H                 ; Jump over status word
                                           ; (so status is 'sticky')
0393 13       INX  D
0394 05       DCR  B
0395 C28803   JNZ  SLOOP
0398 21990F   SLAST: LXI  H, DEND-(NCHAR-1) ; Load address of 1st char
                                           ; on last line
039B 3E00     MVI  A, 0                   ; and blank the line
039D CDA103   CALL BFILL
03A0 C9       RET

```

; Blank line routine

```

; fills in remainder of line with blanks
; H, L contains starting address for blanking
; A contains starting column for blanking
; Exits with H, L set to beginning of next line
; Note columns are numbered from -1 (status) to NCHAR-1
; (last character)
; Must not be rewritten so as to clobber B Register

```

```

03A1 D650     BFILL: SUI  NCHAR                 ; Reformat counter
03A3 3620     BLOOP: MVI  M, ' '           ; Blank this column
03A5 23       INX  H                 ; Bump pointers
03A6 3C       INR  A
03A7 C2A303   JNZ  BLOOP                 ; End of line?
03AA C9       RET

```

; Routine to handle first character typed after a RESET

```

03AB FE4C      FRSTC: CPI   'L'           ; It first char typed is 'L'--set
                                ; to local mode
03AD C8        RZ
03AE CD2403    CALL  FULL           ; else--accept defaults
                                ; (full duplex, on-line)
03B1 C34203    JMP   LINE

```

; Routine to delete a character at cursor position

```

03B4 2A0410    DLETE: LHL D  CLINE
03B7 114F00    LXI  D, NCHAR-1
03BA 19        DAD  D           ; H gets address of last column
03BB 7D        MOV  A, L           ; Save in A for loop termination
                                ; check
03BC 2A0210    LHL D  CURSE
03BF BD        DLOOP: CMP  L           ; Finished?
03C0 CACB03    JZ   LAST          ; Yes - do last column
03C3 23        INX  H
03C4 46        MOV  B, M           ; Get char from cursor+1
03C5 2B        DCX  H
03C6 70        MOV  M, B           ; and write it in cursor position
03C7 23        INX  H
03C8 C3BF03    JMP  DLOOP          ; Continue
03CB 3620     LAST: MVI  M, ' '       ; Put a blank in the last column
03CD C9        RET

```

; The following routine inserts a character on a line at the cursor  
; Position by pushing all other characters on the line one place to  
; The right - the character to be displayed should be in the B  
; Register.

```

03CE 2A0210    ADDC: LHL D  CURSE
03D1 7D        MOV  A, L           ; Save low order bits for loop termination check
03D2 2A0410    LHL D  CLINE
03D5 114F00    LXI  D, NCHAR-1
03D8 19        DAD  D           ; Address of last column
03D9 BD        ALOOP: CMP  L           ; Finished?
03DA CAE503    JZ   ALAST          ; Yes - do last character
03DD 2B        DCX  H
03DE 4E        MOV  C, M           ; Get char from current pos- 1
03DF 23        INX  H
03E0 71        MOV  M, C           ; and write it at current pos
03E1 2B        DCX  H           ; Move left for next char
03E2 C3D903    JMP  ALOOP          ; Continue
03E5 70     ALAST: MOV  M, B           ; Insert the new character
03E6 C9        RET

```

; Routine to delete a line--used by ^Z Processor

```

03E7 3A0110   DROW:  LDA  YCURS           ; If last line then simply blank it
03EA FE17     CPI  NLINE-1
03EC CA9803   JZ   SLAST
03EF 2A0410   LHLD CLINE           ; Delete current line and scroll up
03F2 EB       XCHG          ; Save CLINE in D, E
03F3 47       MOV  B, A
03F4 3E17     MVI  A, NLINE-1
03F6 90       SUB  B
03F7 47       MOV  B, A
03F8 0E50     MVI  C, NCHAR
03FA 215100   LXI  H, NCHAR+1
03FD 19       DAD  D           ; H, L get CLINE+NCHAR+1
03FE C38803   JMP  SLOOP          ; Scroll, then return
    
```

; Routine to add a row--used by ^J processor

```

0401 3A0110   AROW:  LDA  YCURS           ; If last line, then blank it
0404 FE17     CPI  NLINE-1
0406 CA2604   JZ   AFIN
0409 47       MOV  B, A
040A 3E17     MVI  A, NLINE-1
040C 90       SUB  B
040D 67       MOV  H, A
040E 2E50     MVI  L, NCHAR
0410 01E80F   LXI  B, DEND          ; B, D are pointers for
                                ; transferring characters
0413 11970F   LXI  D, DEND-(NCHAR+1)
0416 1A       DSLUP: LDAX D           ; Move character to new line
0417 02       STAX B
0418 1B       DCX  D           ; Decrement pointers and counter
0419 0B       DCX  B
041A 2D       DCR  L
041B C21604   JNZ  DSLUP           ; Repeat loop
041E 2E50     MVI  L, NCHAR       ; Refresh character counter
0420 1B       DCX  D           ; Skip status word
0421 0B       DCX  B
0422 25       DCR  H
0423 C21604   JNZ  DSLUP
0426 2A0410   AFIN:  LHLD CLINE          ; Set up parameters for blanking line
0429 3E00     MVI  A, 0
042B CDA103   CALL BFILL          ; Fill current line with blanks
042E C9       RET
    
```

; Double register negate (2's complement)

```

042F 7C      DNG:  MOV  A, H
0430 2F      CMA                      ; Input in H, L. Output in B, C
0431 47      MOV  B, A
0432 7D      MOV  A, L                      ; Destroys A register
0433 2F      CMA
0434 4F      MOV  C, A
0435 03      INX  B
0436 C9      RET

```

; Multiply C register times D register. Result in B, C  
; See 8080 assembler language manual for explanation

```

0437 0600    MULT:  MVI  B, 0                      ; initialize most significant byte
                                           ; of result
0439 1E09    MVI  E, 9                      ; Bit counter
043B 79      MULT0: MOV  A, C                      ; Rotate least significant bit of
                                           ; multiplier to carry and shift

043C 1F      RAR
043D 4F      MOV  C, A                      ; Low order byte of result
043E 1D      DCR  E
043F C8      RZ                          ; Exit if complete
0440 78      MOV  A, B
0441 D24504  JNC  MULT1
0444 82      ADD  D                      ; Add multiplicand to high order
                                           ; byte of result if bit was 1
0445 1F      MULT1: RAR                      ; Carry=0 here--shift high order
                                           ; byte of result

0446 47      MOV  B, A
0447 C33B04  JMP  MULT0

```

```

; *****
;
;           Download Routine
;
; *****

```

```

; Download data, check checksum, transmit confirmation char

```

```

; Format for downloaded file:
; Byte 1 & 2   low & high order bytes of beginning address
; Byte 3 & 4   low & high order bytes of byte count
; Byte 5       control byte: if 0 then load & continue,
;               else load & go
;
; Byte 6       checksum byte--negated sum of downloaded data bytes
; The rest     download data

```

```

044A 3EFF      DNLD:  MVI  A, 3770          ; Alter parity bit mask to accept
044C 320F10    STA  PMASK          ; 8 bit data
044F 114508    LXI  D, TLINE+DL0
0452 214D05    LXI  H, DMESS       ; Display downloading message
0455 CD1B05    CALL SDISP
0458 CDDA04    CALL GQUE          ; Accept starting address for data
045B 6F        MOV  L, A
045C CDDA04    CALL GQUE
045F 67        MOV  H, A
0460 220C10    SHLD BEGAD        ; Save beginning address
0463 EB        XCHG          ; Shift address to D, E Registers
0464 CDDA04    CALL GQUE          ; Place byte count in H, L
0467 6F        MOV  L, A
0468 CDDA04    CALL GQUE
046B 67        MOV  H, A
046C 19        DAD  D          ; Final address + 1 in H, L
046D CDDA04    CALL GQUE
0470 320B10    STA  CNBYT        ; Save control byte
0473 CDDA04    CALL GQUE
0476 47        MOV  B, A          ; Save checksum in B
0477 7B        DNLD1: MOV  A, E          ; Finished loading data?
0478 BD        CMP  L          ; Check lower order byte first
0479 C28104    JNZ  MORE         ; No
047C 7A        MOV  A, D
047D BC        CMP  H          ; Check higher order byte
047E CA8B04    JZ   CHECK        ; Finished - compare checksums
0481 CDDA04    MORE:  CALL GQUE        ; Get next byte
0484 12        STAX D          ; And store it
0485 80        ADD  B
0486 47        MOV  B, A          ; Update checksum
0487 13        INX  D          ; Increment pointer
0488 C37704    JMP  DNLD1

```

; Checksum Comparison Routine--used in downloader

```

048B 3E7F      CHECK: MVI  A, 1770      ; Restore parity bit mask
048D 320F10    STA  PMASK
0490 214508    LXI  H, TLINE+DL0
0493 3E44      MVI  A, DL0-1          ; Clear message
0495 CDA103    CALL BFILL
0498 78        MOV  A, B              ; Checksum Register (B) should be
                                ; zero now
0499 0606      MVI  B, ACK
049B FE00      CPI  0
049D CAA204    JZ   CNFRM            ; Transmit ^F (ACK) if checksums
                                ; agree
                                ; Else transmit ^U (NAK)
04A0 0615      MVI  B, NAK
04A2 CD0401    CNFRM: CALL TXCHR
04A5 78        MOV  A, B
04A6 FE15      CPI  NAK
04A8 C8        RZ              ; If checksum does not agree,
                                ; then return
04A9 3A0B10    LDA  CNBYT
04AC FE00      CPI  0              ; Else check control byte to
                                ; determine next action
04AE C8        RZ              ; If 0, then return
04AF 2A0C10    LHLD BEGAD           ; Else jump to downloaded program
04B2 E9        PCHL

```

; This routine initializes the status words for a normal display

```

04B3 0E00      IST:  MVI  C, 0
04B5 0620      MVI  B, STN          ; B reg contains normal status word
04B7 CDC204    IST1: CALL LSTAT      ; LSTAT sets status on one line
04BA 3E18      MVI  A, NLINE
04BC B9        CMP  C              ; Finished?
04BD C8        RZ
04BE 0C        INR  C
04BF C3B704    JMP  IST1           ; No - continue

```

; Routine to set the status on the line number contained in  
; The C register to the value contained in the B register

```

04C2 C5        LSTAT: PUSH B
04C3 1651      MVI  D, NCHAR+1
04C5 CD3704    CALL MULT
04C8 215108    LXI  H, DBGIN-1     ; Address of 1st status word
04CB 09        DAD  B              ; (ignoring top line)
04CC C1        POP  B
04CD 70        MOV  M, B
04CE C9        RET

```

```
; *****  
;  
; Character Queue Routines  
;  
; *****
```

```
; PQUE  
; Adds character in B Register to queue  
; Destroys H and L, no check for queue overflow
```

```
04CF 2A1010 PQUE: LHL D QHEAD ; Get head of queue  
04D2 70 MOV M, B ; And store character  
04D3 CD0605 CALL MVPTR ; Bump ptr in H, L  
04D6 221010 SHLD QHEAD ; Save pointer  
04D9 C9 RET
```

```
; GQUE  
; Attempts to remove the first character in the queue and  
; Return it in the A register, if the queue is empty,  
; then GQUE waits for a character to become available
```

```
04DA CDE104 GQUE: CALL GQ  
04DD D8 RC  
04DE C3DA04 JMP GQUE
```



```

; GQ
; Returns the first character in the queue in the A
; Register, If the queue is empty, then GQ returns with
; the carry = 0, and the A Register contents are to be
; ignored. If the character is valid, then GQ returns with
; the carry = 1.

```

```

04E1 C5      GQ:   PUSH  B
04E2 D5      PUSH  D
04E3 E5      PUSH  H
04E4 2A1210  GQUE1: LHLD  QTAIL
04E7 3A1010  LDA   QHEAD           ; Get low order address bits of
                                ; queue head and compare with low
                                ; order address bits of qtail

04EA BD      CMP   L
04EB C2FA04  JNZ  NOEMP           ; If not equal, then queue is not
                                ; empty
04EE 3A1110  LDA   QHEAD+1       ; Get high order bits of qtail
04F1 BC      CMP   H           ; and compare
04F2 C2FA04  JNZ  NOEMP           ; Queue not empty
04F5 37      STC           ; Queue empty, clear carry
04F6 3F      CMC
04F7 C30205  JMP  PP
04FA 7E      NOEMP: MOV  A, M           ; If queue not empty, get character
04FB CD0605  CALL MVPTR         ; Bump ptr in H, L
04FE 221210  SHLD QTAIL         ; Save ptr
0501 37      STC           ; Set carry for normal return
0502 E1      PP:   POP  H
0503 D1      POP  D
0504 C1      POP  B
0505 C9      RET

```

```

; Moves queue pointer (in H, L)
; Watches for wrap-around

```

```

0506 F5      MVPTR: PUSH  PSW
0507 2B      DCX  H           ; Move pointer
0508 3A5905  LDA  QUEM1           ; Get low order bits of queue
                                ; bottom - 1
050B BD      CMP  L           ; and compare with ptr
050C C21905  JNZ  MRTRN         ; Return unless A = L
050F 3A5A05  LDA  QUEM1+1       ; Get high order bits of queue
                                ; bottom - 1
0512 BC      CMP  H
0513 C21905  JNZ  MRTRN         ; Return unless A = L
0516 21FC13  LXI  H, QTOP       ; wrap-around. Load top of queue
                                ; address

0519 F1      MRTRN: POP  PSW
051A C9      RET

```

```

; This routine displays a string consisting of a series of bytes
; Terminated by a null byte.
; Enter with the address of the first byte of the string in H, L
; and the address in video memory where the string is to start in
; D, E
; Destroys machine state

```

```

051B 7E      SDISP:  MOV  A, M          ; Get a string byte
051C FE00    CPI    0              ; End on null byte
051E C8      RZ
051F 23      INX   H              ; Update address for next string
                                ; byte
0520 12      STAX  D              ; Display the character
0521 13      INX   D              ; Update address for next display
                                ; byte
0522 C31B05  JMP   SDISP           ; Next byte

```

```

; The following bytes correspond to strings for top line
; Status display

```

```

0525 D3      LMESS:  DB    200Q+'S'      ; Logo message
0526 CD      DB    200Q+'M'
0527 C9      DB    200Q+'I'
0528 D4      DB    200Q+'T'
0529 C3      DB    200Q+'C'
052A C8      DB    200Q+'H'
052B AD      DB    200Q+'-'
052C B0      DB    200Q+'0'
052D 00      DB    0

```

```

; *****

```

```

052E 48      HMESS:  DB    'H'          ; Half duplex message
052F 44      DB    'D'
0530 00      DB    0

```

```

0531 46      FMESS:  DB    'F'          ; Full duplex message
0532 44      DB    'D'
0533 00      DB    0

```

```

0534 20      LOMES:  DB    ' '          ; Local message
0535 4C      DB    'L'
0536 4F      DB    'O'
0537 43      DB    'C'
0538 41      DB    'A'
0539 4C      DB    'L'
053A 20      DB    ' '
053B 00      DB    0

```

053C 4F	LIMES;	DB	'O'	; On-line message
053D 4E		DB	'N'	
053E 2D		DB	'-'	
053F 4C		DB	'L'	
0540 49		DB	'I'	
0541 4E		DB	'N'	
0542 45		DB	'E'	
0543 00		DB	0	
0544 52	RMES:	DB	'R'	; Roll mode message
0545 4F		DB	'O'	
0546 4C		DB	'L'	
0547 4C		DB	'L'	
0548 00		DB	0	
0549 49	IMES:	DB	'I'	; I/d mode message
054A 2F		DB	'/'	
054B 44		DB	'D'	
054C 00		DB	0	
054D C4	DMES:	DB	200Q+'D'	; Downloading message
054E CF		DB	200Q+'O'	
054F D7		DB	200Q+'W'	
0550 CE		DB	200Q+'N'	
0551 CC		DB	200Q+'L'	
0552 CF		DB	200Q+'O'	
0553 C1		DB	200Q+'A'	
0554 C4		DB	200Q+'D'	
0555 C9		DB	200Q+'I'	
0556 CE		DB	200Q+'N'	
0557 C7		DB	200Q+'G'	
0558 00		DB	0	
0559 1310	QUEM1:	DW	QUEUE-1	; Constant for queue wrap-around test

```

1000          ORG  4096          ; Variables stored in RAM
1000 00      XCURS: DB  0        ; Column counter (-1, 0 to NCHAR-1)
1001 00      YCURS: DB  0        ; Row counter (-1, 0 to NLINE-1)
1002 0000    CURSE: DW  0        ; Absolute cursor address
1004 0000    CLINE: DW  0        ; Address of column 0 of current
                                     ; cursor line
1006 00      IDMOD: DB  0        ; State of inset/delete mode
                                     ; on/off = 377Q/0Q
1007 00      BKMOD: DB  0        ; Blink mode flag. on/off = 200Q/0Q
1008 00      RLMOD: DB  0        ; State of roll mode.
                                     ; on/off = 377Q/0Q
1009 00      LCMOD: DB  0        ; Local mode flag. on/off = 377Q/0Q
100A 00      HDMOD: DB  0        ; Half duplex mode, on/off = 377Q/0Q
100B 00      CNBYT: DB  0        ; Control byte for downloading
100C 0000    BEGAD: DW  0        ; Beginning address of downloaded
                                     ; program
100E 00      UFLG:  DB  0        ; Flag set when ^U received
                                     ; on/off = 377Q/0Q
100F 00      PMASK: DB  0        ; Parity bit mask
1010 0000    QHEAD: DW  0        ; Pointer to head of queue
1012 0000    QTAIL: DW  0        ; Pointer to tail of queue
03E8         QUEUE: DS  1000     ; Maximum length of queue. If
                                     ; 1000 chars isn't enough,
                                     ; there's room for more
13FC 00      QTOP:  DB  0        ; Top byte of queue
13FD 00      MON:   DB  0        ; Address for downloaded local
                                     ; monitor routine

          END

```

NO PROGRAM ERRORS

ABELL	003B	HMESS	052E	QTAIL	1012
ACK	0006	HOME	017E	QTOP	13FC
ADDC	03CE	HUH	019E	QUEM1	0559
AFIN	0426	IDD0	003A	QUEUE	1014
ALAST	03E5	IDMOD	1006	RCTBL	0040
ALLOOP	03D9	IMESS	0549	RETRN	0124
AROW	0401	INIT	02F2	RIGHT	02C9
BEGAD	100C	IST	04B3	RLMOD	1008
BEL1	01AF	IST1	04B7	RMESS	0544
BELL	01A1	KBDSV	00BE	ROK	02D7
BFILL	03A1	KEYBD	003D	ROLL	02E3
BKMOD	1007	KTST	00E3	ROLL0	002D
BLANK	02FF	LAST	03CB	RTRN1	011F
BLOOP	03A3	LCA	020F	RXSV	0110
CANCL	024B	LCMOD	1009	SB	0262
CCTBL	013E	LCX	0215	SCR0L	037E
CHECK	048B	LCY	0227	SDISP	051B
CLINE	1004	LEFT	01B2	SLAST	0398
CNBYT	100B	LF	01CC	SLOOP	0388
CNFRM	04A2	LF1	01F9	STACK	1FFF
CR	0256	LIMES	053C	STN	0020
CTRLU	00F6	LINE	0342	TERM	00AC
CTUR	0277	LMESS	0525	TLINE	0800
CURSE	1002	LOCAL	0333	TSTA	0032
DBGIN	0852	LOGO	0007	TSTB	0022
DCHAR	0351	LOMES	0534	TXCHR	0104
DEND	0FE8	LSTAT	04C2	UFLG	100E
DFIN	0372	MOD0	001D	UNL0K	0295
DL0	0045	MODES	018F	UP	02A5
DLE	0268	MON	13FD	UPORT	003E
DLETE	03B4	MORE	0481	USTAT	003F
DLOOP	03BF	MRTRN	0519	WAIT	00A6
DM	005E	MULT	0437	WBLNK	0306
DMESS	054D	MULT0	043B	XCURS	1000
DNG	042F	MULT1	0445	XST	021E
DNLD	044A	MVPTR	0506	YCURS	1001
DNLD1	0477	NAK	0015	YST	0230
DOK	036E	NCHAR	0050		
DROW	03E7	NLINE	0018		
DSLUP	0416	NOEMP	04FA		
DUP0	0012	NRET	0128		
EE0L	028C	OFCUR	003A		
FMESS	0531	ONCUR	0039		
FRSTC	03AB	PCHAR	0129		
FULL	0324	PICU	003C		
GQ	04E1	PMASK	100F		
GQUE	04DA	POLL	0105		
GQUE1	04E4	PP	0502		
HALF	0315	PQUE	04CF		
HDMOD	100A	QHEAD	1010		