

```

0036: 0002 91 CSWL DS 2 ;hook for output routine
0038: 0037 92 CSWH EQU CSWL+1
0038: 0002 93 KSWL DS 2 ;hook for input routine
003A: 0039 94 KSWH EQU KSWL+1
003C: 003C 95 ORG $3C
003C: 0002 96 A1L DS 2 ;Monitor temps for MOVE
003E: 003D 97 A1H EQU A1L+1
003E: 0002 98 A2L DS 2
0040: 003F 99 A2H EQU A2L+1
0040: 0002 100 DS 2 ;A3 NOT USED
0042: 0002 101 A4L DS 2
0044: 0043 102 A4H EQU A4L+1
0044: 0001 103 MACSTAT DS 1 ;machine state on breaks
004E: 004E 104 ORG $4E
004E: 0002 105 RNDL DS 2 ;random number seed
0050: 004F 106 RNDH EQU RNDL+1
0000: 107 DEND
0000: 108 *
0000: 0200 109 BUF EQU $200 ;input buffer
0000: 110 * Permanent data in screenholes
0000: 111 *
0000: 112 * Note: these screenholes are only used by
0000: 113 * the 80 column firmware if an 80 column card
0000: 114 * is detected or if the user explicitly activates
0000: 115 * the firmware. If the 80 column card is not
0000: 116 * present, only MODE is trashed on RESET.
0000: 117 *
0000: 118 * The success of these routines rely on the
0000: 119 * fact that if 80 column store is on (as it
0000: 120 * normally is during 80 column operation), that
0000: 121 * text page 1 is switched in. Do not call the
0000: 122 * video firmware if video page 2 is switched in!!
0000: 123 *
0000: 07F8 124 MSL0T EQU $7F8 ;=$Cn ;n=slot using $C800
0000: 125 *
0000: 047B 126 OLDCH EQU $478+3 ;LAST CH used by video firmware
0000: 04FB 127 MODE EQU $4F8+3 ;video firmware operating mode
0000: 057B 128 OURCH EQU $578+3 ;80 column CH
0000: 05FB 129 OURCV EQU $5F8+3 ;80 column CV
0000: 067B 130 CHAR EQU $678+3 ;character to be printed/read
0000: 06FB 131 XCOORD EQU $6F8+3 ;GOTOXY X-coord (pascal only)
0000: 077B 132 TEMP1 EQU $778+3 ;temp
0000: 077B 133 OLDBASL EQU $778+3 ;last BASL (pascal only)
0000: 07FB 134 TEMP2 EQU $7F8+3 ;temp
0000: 07FB 135 OLDBASH EQU $7F8+3 ;last BASH (pascal only)
0000: 136 *
0000: 137 * BASIC MODE BITS
0000: 138 *
0000: 139 * 0..... - BASIC active
0000: 140 * 1..... - Pascal active
0000: 141 * .0..... -
0000: 142 * .1..... -
0000: 143 * ..0..... - Print control characters
0000: 144 * ...1..... - Don't print ctrl chars.

```

```

0000: 145 * ...0.... -
0000: 146 * ...1.... -
0000: 147 * ....0.... - Print control characters
0000: 148 * .....1.... - Don't print next ctrl char
0000: 149 * .....0.... -
0000: 150 * .....1.... -
0000: 151 * .....0.... -
0000: 152 * .....1.... -
0000: 153 * .....0.... - Mouse text inactive
0000: 154 * .....1.... - Mouse text active
0000: 155 *
0000: 0040 156 M.6 EQU $40
0000: 0020 157 M.CTL2 EQU $20 ;Don't print controls
0000: 0010 158 M.4 EQU $10
0000: 0008 159 M.CTL EQU $08 ;Temp ctrl disable
0000: 0004 160 M.2 EQU $04
0000: 0002 161 M.1 EQU $02
0000: 0001 162 M.MOUSE EQU $01
0000: 163 *
0000: 164 * Pascal Mode Bits
0000: 165 *
0000: 166 * 0..... - BASIC active
0000: 167 * 1..... - Pascal active
0000: 168 * .0..... -
0000: 169 * .1..... -
0000: 170 * ..0..... -
0000: 171 * ..1..... -
0000: 172 * ...0.... - Cursor always on
0000: 173 * ...1.... - Cursor always off
0000: 174 * ....0.... - GOTOXY n/a
0000: 175 * ....1.... - GOTOXY in progress
0000: 176 * .....0.... - Normal Video
0000: 177 * .....1.... - Inverse Video
0000: 178 * .....0.... - PASCAL 1.1 F/W ACTIVE
0000: 179 * .....1.... - PASCAL 1.0 INTERFACE
0000: 180 * .....0.... - Mouse text inactive
0000: 181 * .....1.... - Mouse text active
0000: 182 *
0000: 0080 183 M.PASCAL EQU $80 ;Pascal active
0000: 0010 184 M.CURSOR EQU $10 ;Don't print cursor
0000: 0008 185 M.GOXY EQU $08 ;GOTOXY IN PROGRESS
0000: 0004 186 M.VMODE EQU $04 ;PASCAL VIDEO MODE
0000: 0002 187 M.PAS1.0 EQU $02 ;PASCAL 1.0 MODE
0000: 188 *
0000: 189 * F8 ROM entries
0000: 190 *
0000: FA47 191 NEWBREAK EQU F8ORG+$247
0000: FC74 192 IROUSER EQU F8ORG+$474
0000: FC7A 193 IRQDONE2 EQU F8ORG+$47A
0000: F8B7 194 TSTROM EQU F8ORG+$B7
0000: 18 INCLUDE BFUNC
----- NEXT OBJECT FILE NAME IS REFLIST.0
C100: C100 1 ORG C1ORG
C100: C100 2 BFUNCPG EQU *

```